



# A geometric alternative to computed tomography

Jean-Philippe Thirion

► **To cite this version:**

Jean-Philippe Thirion. A geometric alternative to computed tomography. [Research Report] RR-1463, INRIA. 1991, pp.34. <inria-00075098>

**HAL Id: inria-00075098**

**<https://hal.inria.fr/inria-00075098>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**UNITÉ DE RECHERCHE  
INRIA-ROCQUENCOURT**

**Institut National  
de Recherche  
en Informatique  
et en Automatique**

**Domaine de Voluceau  
Rocquencourt  
B.P. 105  
78153 Le Chesnay Cedex  
France  
Tél.:(1)39 63 55 11**

# Rapports de Recherche

**N°1463**

*Programme 4*

*Robotique, Image et Vision*

## **A GEOMETRIC ALTERNATIVE TO COMPUTED TOMOGRAPHY**

**Jean-Philippe THIRION**

**Juin 1991**

# A Geometric Alternative to Computed Tomography \*

**Jean-Philippe THIRION**  
INRIA, Epidaure †

June 10, 1991

## Abstract

This paper describes a totally new way to process Xray data in order to reconstruct the external and internal boundaries of objects, which do not involve Computed Tomography (CT). We show that the segmentation can be performed directly with the raw data, the sinogram produced with the scanner, and that those segmented shapes can be geometrically transformed into reconstructed shapes in the usual space. Thus, if we are interested in only the boundaries of the objects, our method eliminates the computationally expensive step of Computed Tomography. Experimental results are presented for both synthetic and real data, leading to subpixel positioning of the reconstructed boundaries. Our method gives its best results for sparse, high contrasted objects such as bones or blood vessels in angiograms. It can be adapted to any kind of scanner, including 3D scanners. At last, we present an extension of our method which allows “on the fly” processing of the data and real time tracking of the objects boundaries.

**Key words:** Xray scanner, Computed Tomography, segmentation, surface reconstruction, deformable model, Geometric Tomography.

---

\*This paper describes parts of a patent taken out by the INRIA

†<http://zenon.inria.fr:8003/Equipes/EPIDAURE-eng.html>

# Une alternative géométrique à la Tomographie Calculée <sup>1</sup>

**Jean-Philippe THIRION**

INRIA, Epidaure

## Résumé :

Cet article décrit une méthode entièrement nouvelle de traitement des données scanner, qui permet la reconstruction des surfaces internes et externes d'objets, et qui ne nécessite pas la mise en œuvre d'une méthode de Tomographie Calculée (Computed Tomography, CT). Nous montrons comment l'étape de segmentation peut être réalisée directement sur les données brutes, ou sinogrammes, produit par le scanner, et comment les informations géométriques obtenues peuvent être traduites en formes géométriques de l'espace usuel. Ainsi, dans le cas où l'on s'intéresse uniquement aux frontières des objets, notre méthode permet d'éviter de l'étape longue et coûteuse de la Tomographie Calculée. Nous présentons des résultats obtenus pour des données synthétiques et des données réelles, conduisant à une précision de reconstruction inférieure au pixel. Notre méthode donne ses meilleurs résultats dans le cas d'objets peu nombreux, bien répartis, et très contrastés, tels que les os, ou les vaisseaux sanguins dans les angiographies. La méthode peut être facilement adaptée à n'importe quel type de scanner, y compris les scanners 3D. Enfin, nous présentons une extension de cette méthode, qui permet le traitement en cours d'acquisition des données, et le suivi en temps réel des surfaces caractéristiques des objets.

**Mots clef:** scanner X, Tomographie Calculée, segmentation, reconstruction de surfaces, modèle déformable, Tomographie Géométrique.

---

<sup>1</sup>Cet article décrit en partie un brevet faisant l'objet d'un dépôt par l'INRIA

# 1 Introduction

Reconstructing the boundaries of the objects from medical images is a very active field that involves both geometric and signal processing knowledge. The principles of Computed Tomography (CT) are now well established. CT transforms several 1 dimensional projections of the objects into a 2D image, where each pixel contains the value of the absorption coefficient at this location, into a slice of the object. This transform is referred to as the *Radon transform* (see [17]). A straightforward extension is to transform several 2D projections into a 3D image (a set of voxels). Two alternative techniques are used to inverse the Radon transform : analytical solutions, based on the Fourier transform and/or the back projection (Bracewell and Riddle [5], Oldendorf [15], Cormack [7]), and algebraic solutions, like the ART algorithm (Algebraic Reconstruction technique, see for example Gordon et al. [9], Herman et al. [10] ). Then, starting with the image produced with CT techniques, other, independent methods are used to detect edges and to reconstruct and display external or internal surfaces. One alternative is to visualize those surfaces, directly from the voxel representation (see [13]), another one is to first reconstruct a geometrical representation of those surfaces, and then to display them with conventional Computer Graphics techniques [3]. Surface representation can be used for other purposes than visualization or diagnostic, like for example the measurement of volumes, surgical planning, prosthesis design, craniofacial reconstructions, . . . .

We propose in this paper a geometric method in order to reconstruct directly the boundaries of the objects from the set of 1D or 2D projections, without using an intermediate pixel or voxel representation of the objects in object space, and therefore without using the step of Computed Tomography. (see figure 1). The main idea is to detect the edges and to follow the external or internal boundaries of the object directly into the original set of projections (the sinogram).

Our study is a geometry based approach. Recent works in Computer Vision, gave a new insight into the geometrical problem of finding the shape of an object from its profiles. Those works were based on a continuous representation of the objects among a set of projections. Some concentrates on finding hidden surfaces and detecting occlusions between the objects, like [16], others tried to reconstruct surfaces from the set of different camera views ([4],[14],[8],[18], [1],[6]). Their main problem was to determine very

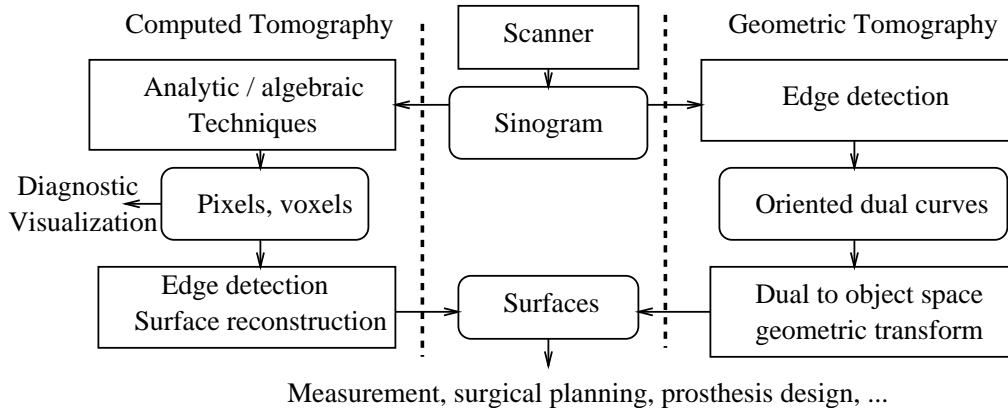


Figure 1: The geometric alternative to Computed Tomography

accurately the position of the camera along the time, any slight error leading to severe artefacts for the reconstructed shape of the object. The other problem was the occlusion of an object, which prevents from reconstructing some parts of it. For example, self occlusion prevents from reconstructing concave parts of the object. Because we are dealing with transmission tomography, many of those problems disappear: all the objects are transparents, and therefore there is never a complete occlusion of an object by another one. Furthermore, the trajectory of the scanning line, our “camera”, is very well defined, or can be precisely measured by the way of synthetic objects (phantoms). As we will see, some related works have already been performed in the field of tomography, which directly use the set of projections, mainly in order to compute the attenuation coefficients for PET or SPECT data (see [2], [11]). Their purpose was to roughly locate the external shape of a convex object, whereas our goal is to precisely determine the position of both internal and external boundaries, for (possibly) non-convex objects, and with (possibly) subpixel precision.

## 2 Definitions

We briefly recall the principle of a third generation scanner, with parallel beams, and we describe some geometrical properties of the projective space.

Then we explain our basic assumptions which is that the internal or external boundaries of the objects are responsible of discontinuities in the projections. Those discontinuities can be directly detected in the raw images generated by the scanner, and transformed into geometric informations in the object space.

## 2.1 A third generation scanner

Most of the data which are presented in this paper have been simulated with our own third generation scanner simulator. This simulator allows us to verify the assumption of subpixel positioning, because any geometric feature is known and well defined. We deal with transmission tomography, and we suppose that the X rays are all parallel. The third generation scanner produce a set of one dimensional projections of the objects, with a constant spacing between the angle of viewing (see figure 2). For our simulations, the X rays are linearly absorbed by the objects, and the noise in the projections is not modeled. Our scanner is characterized by the following parameters :

- r: number of detectors in a line.
- n: number of viewing angles, ranging from 0 to  $\text{PI}$ .
- l: length of the line of detectors.
- d: distance between the line of detectors and the center of the rotation.

Furthermore, the line of detectors is centered around the closest point of the line to the center of the rotation. We use those assumptions in order to have a simpler model, but our method do not require all of them. The last part of this paper is about the generalization of our method. The set of 1D projections are collected in order to form a 2D image  $I(U, \theta)$  called *sinogram* (see figure 3). The axis  $U$  of this image corresponds to the position of the detector along the line of detectors, the axis  $\Theta$  corresponds to the angle of the parallel rays, with respect to a fixed direction.

## 2.2 Geometric properties

For a given hypersurface  $S$  of a multi-dimensional space  $E$ , and a given orientation  $\vec{d}$ , the *critical set* (see [8]) is the set of the points of  $S$  where the

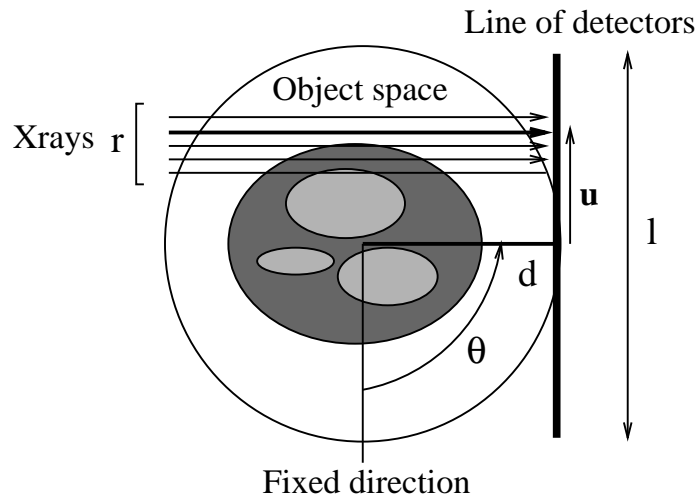


Figure 2: Model of a third generation scanner, with parallel beams

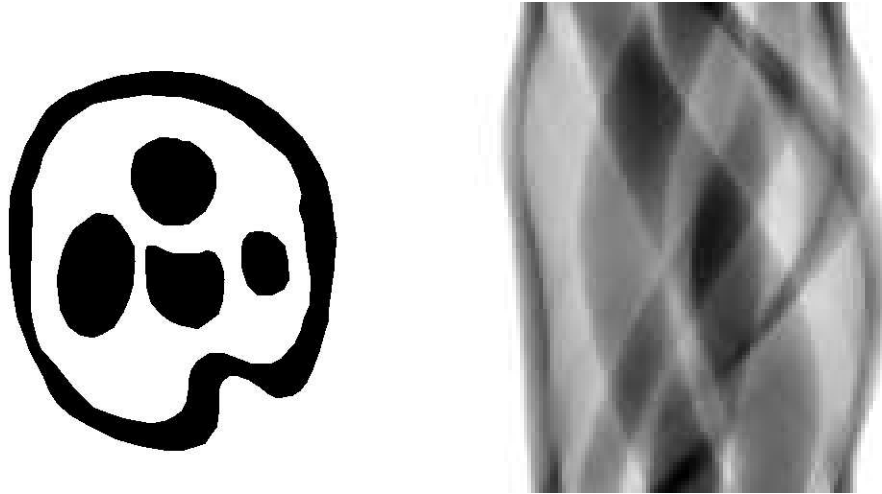


Figure 3: Synthetic object (left) and the corresponding sinogram  $I(u, \theta)$  with  $\theta \in [0, \Pi)$  (right)



tangent hyperplane to  $S$  contains the vector  $\vec{d}$ . The *profile* is the orthogonal projection of the critical set onto a hyperplane perpendicular to  $\vec{d}$ . Let us introduce now our notations. For a  $3^{rd}$  generation scanner,  $E$  has two dimensions  $E(X, Y)$ , and each projection  $I_\theta(U)$  of the object has one dimension  $U$ . Therefore the profile of a curve  $C(x, y) = 0$  is a set of points, where each point corresponds to a straight line tangent to  $C$ . The set of profile points  $C^*(u, \theta) = 0$  into the whole set of projections (the sinogram)  $I(U, \Theta)$ , will be called the *dual curve* associated to  $C$  (sometime it is also called the “pedal curve” of  $C$ ). Conversely, a curve defined with the set of its tangent lines is called the *envelope* of this family of lines.

The 2D space associated to the sinogram  $I(U, \Theta)$ , with  $\theta \in [0, \Pi)$ , will be called *the dual space*  $E^*(U, \Theta)$  associated to  $E(X, Y)$ .  $E^*$  is the parametric space of the lines of the object space  $E$  (each point in  $E^*$  corresponds to a line of  $E$ ).  $E^*$  is a moebius strip (see figure 3). Let us give an orientation to  $E$  and denote this:  $\vec{E}$ . We call *oriented dual space*  $\vec{E}^*(U, \Theta)$ , with  $\theta \in [0, 2 \cdot \Pi)$  the parametric space of the *oriented* lines of  $\vec{E}$ . The oriented dual space  $\vec{E}^*$  is a cylinder.

We define now the “inside”, and the “outside” of any oriented curve  $\vec{C}(x, y) = 0$  of the space  $\vec{E}$ . For example, the left-handed side of the curve is the inside, and the right-handed side is the outside. This allows us to define the *oriented dual curve*  $\vec{C}^*(u, \theta) = 0$  associated to any oriented curve  $\vec{C}$ . It is the points of the oriented dual space  $\vec{E}^*$  corresponding to the oriented lines of  $\vec{E}$  which are tangent to  $\vec{C}$ , and whose direction is locally the same as the curve  $\vec{C}$  (see figures 4,5).

If we assume that for a continuous closed curve  $C$  of  $E$ , a cusp corresponds to an infinite set of tangent lines, ranging between the two tangential directions defined at this point of  $C$ , then the dual curve  $C^*$  associated to  $C$  is also a continuous closed curve of the dual space  $E^*$ . “Closed” is then to be understood in the sense of a moebius (cylindric for  $\vec{E}^*$  space. Furthermore, if  $C$  is “smooth”, that is, without any cusp, then the corresponding dual curve  $C^*$  is also smooth, except for the points of  $C^*$  corresponding to the inflexion points of  $C$ . For a closed curve with a single concavity, the corresponding dual curve leads to a figure with two cusps and a crossing, that is generally referred to as a *swallow tail*, as described in figure 5.

### 2.3 Our basic assumptions

We are interested in the outside boundaries of the object, but also in the internal boundaries corresponding to a “significant” variation of the absorption coefficient inside the object. Those boundaries will be referred to simply as *the boundaries* of the object(s). If we deal with a third generation scanner, those boundaries are plane curves. We will assume that they are also closed and continuous. Therefore, we can give an orientation to the boundaries, according to the value of the absorption coefficient for each side of the boundary: the “inside” will be the side where this coefficient is locally the biggest.

Our basic assumption is that an *edge* of the intensity function in the sinogram corresponds to a line in the object space which is parallel to the axis of the projection and tangent to a boundary of the object. In short, a boundary in the object space can be detected as a continuous set of edges of the intensity function into the sinogram, matching the dual curve associated to this boundary.

To justify this assumption, let us remind that the value measured with a detector corresponds to the integration along the line in the object space of the density of the matter into the object. Therefore, for two parallel lines situated at each side of the point of a boundary, tangent to the viewing direction, and corresponding to a significant variation of the density inside the object, there will be also a swift change of the intensity function between the two measured values. This assumption can be proved for a single, convex and homogeneous object. It can't be proved in the general case, for example because, for some specific points of the sinogram, the corresponding line can be tangent to two boundaries, which compensate each others influence. Noise can also interfere with edges into the sinogram. Nevertheless, one can verify the validity of our assumption looking at the figure 3 (for example, try to find the swallow tail(s) corresponding the concavity into this image).

A second assumption is that for a point of an edge in the sinogram, corresponding to the dual curve associated to a boundary in object space, the sign of the gradient according to the direction  $U$  of the intensity function gives an orientation which allow us to determine the orientation of the dual curve. Once again, this can be proved only for a single convex homogeneous object: the integration of the density along a line for each side of the tangent point leads to a biggest value for the line which intersect the “inside” of

the boundary than for the line which do not intersect it. In short, if we select into a sinogram, ranging between 0 and  $2\Pi$ , the rising edges of the intensity function, we determine then the *oriented* dual curves associated to the boundaries.

To conclude, our approach tries to solve two different kinds of problems : the first one is to detect very accurately the location of the dual curves into the sinogram, in  $E^*$ . The second one is the geometric problem of reconstructing the curves of the object space  $E$  from the set of their tangent lines. We call our approach *Geometric Tomography* (or GT), and we distinguish between two major alternatives that we call *Direct Geometric Tomography* and *Inverse Geometric Tomography*. Roughly, direct GT consists in finding the (oriented) dual curves into the sinogram first, and then to translate them into (oriented) curves of the object space. Inverse GT is an iterative process which start with a first estimated curve in object space  $E$ , and try to minimize both the distance between the corresponding dual curve and the edges into the sinogram, in  $E^*$ , and the variation of the differential properties along the estimated curve in  $E$ .

## 2.4 Some examples and properties of the oriented dual curves

Let us consider a disk of uniform density, centered at the center of the rotation of the scanner. The corresponding sinogram is a strip  $I(u, \theta) = \sqrt{1 + u^2}$ , constant all along the angular axis  $\Theta$ . The dual curve, defined in the previous section, is composed with two parallel lines. Each point of a line corresponds to a line of the object space tangent to the disk. For a sinogram where  $\theta$  is regularly sampled between 0 to  $2\Pi$ , those two lines will follow twice the same edge. The notion of *oriented* dual curve helps to avoid this repetition, and only the line corresponding to an edge with increasing intensity along the axis  $U$  is selected. A point of the oriented dual space  $\vec{E}^*$  corresponds to an oriented straight line of  $\vec{E}$ . In fact this oriented straight line cut the plane  $\vec{E}(X, Y)$  into two half planes. With our convention, the left handed half plane with respect to the orientation of the line is called *interior* half plan and contains the disk, and the right handed half plane is called *exterior* half plane. The disk, as any convex object, is the intersection of all the interior half planes corresponding to the whole set of points of its oriented dual curve.

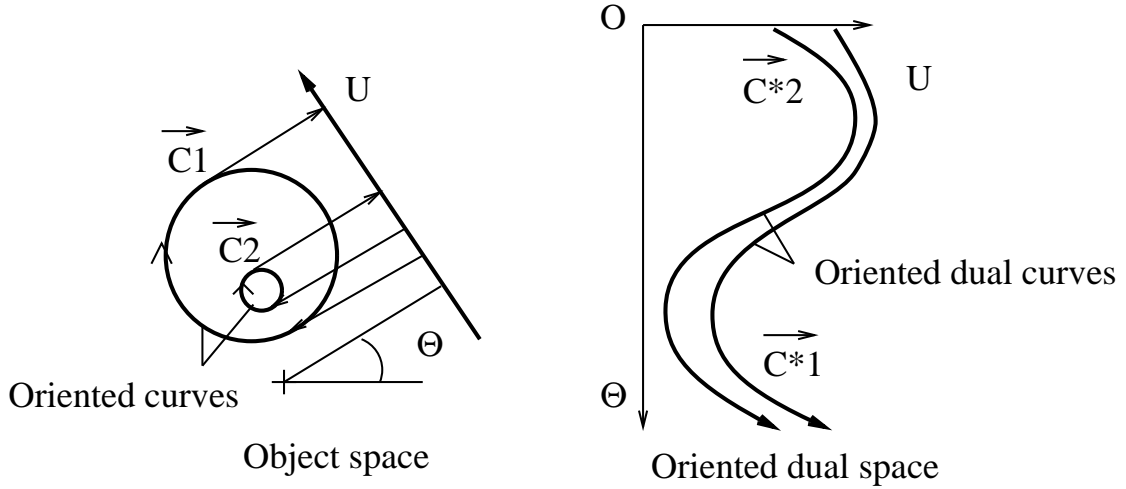


Figure 4:  $C_2$  is inside  $C_1$  because  $\vec{C}_2^*$  is always on the left side of  $\vec{C}_1^*$

If the disk is not centered, the corresponding oriented dual curve is a sinusoid  $u = a + b * \sin(\theta + \phi)$ , where  $a$  is the radius and  $b, \phi$  are the polar coordinates of the center of the disk. If the radius  $a$  tends toward 0, the dual curve is a sinusoid  $b * \sin(\theta + \phi)$ . Therefore, the dual curve of a cusp in object space  $\vec{E}$  is a piece of sinusoid  $b * \sin(\theta + \phi)$  in our oriented dual space  $\vec{E}^*$ , where  $(b, \phi)$  are the polar coordinates of the cusp in  $E$ .

We have also some very interesting topological properties such as (see figure 4):

If a closed oriented dual curve  $\vec{C}_1^*$  of an object  $C_1$  is always on the same side of an oriented dual curve  $\vec{C}_2^*$  of another object  $C_2$ , then  $C_1$  is entirely inside  $C_2$ , or  $C_2$  is entirely inside  $C_1$  depending on the relative position of  $\vec{C}_1^*$  and  $\vec{C}_2^*$  along the axis  $U$ , for only one line of the sinogram.

Let us consider now a non-convex object in the plane, with a single concavity. There are three characteristic points of view for this object (see figure 5). One direction corresponds to a line which is tangent to the curve for two distinct points (line  $A$ ). The two other directions correspond to the tangent of the object for the two inflection points (line  $B$  and  $C$ ). The dual curve overcross for the angle corresponding to line  $A$  (see figure 5). Our capacity to follow the edges of the objects in object space will depend on our ability

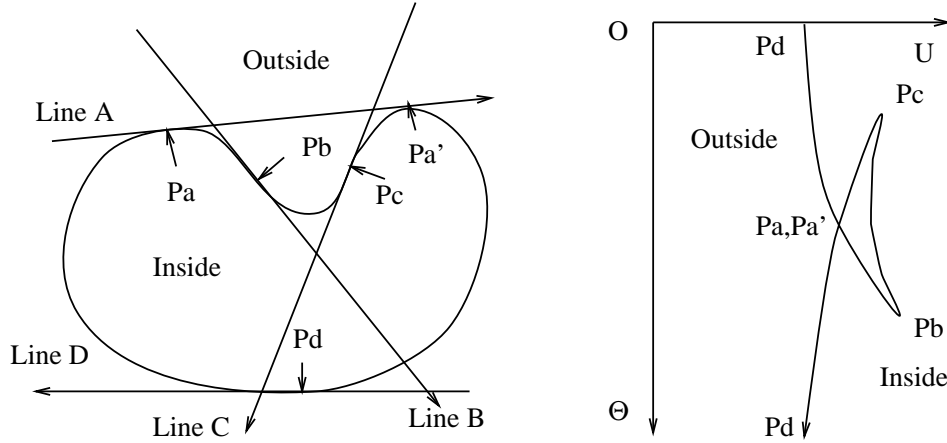


Figure 5: Concave object and the corresponding oriented dual curve. Starting from point  $Pd$ , it goes through  $Pa$ ,  $Pb$ ,  $Pc$ ,  $Pa'$ , and comes back to  $Pd$

to follow the dual curves in the sinogram. If we choose the wrong path in the crossing of a swallow tail, we only reconstruct the convex hull of the object in object space. If we decompose the oriented dual curve  $\vec{C}^*(u, \theta) = 0$  into monotonous parts  $u = f(\theta)$ , each of those parts corresponds to an entirely convex (respectively entirely concave) part of the reconstructed curve, which is defined with the intersection of the corresponding interior half planes (respectively exterior half planes). Thus we are also able to reconstruct the shape of a non-convex object from its oriented dual curve.

To conclude, the sinogram of a complex object is a complex mesh of curves. If we are not able to detect the “swallow tails” in the sinogram, then our method is to be restricted to the detection of only monotonous sections of oriented dual curves. A higher level procedure is then requested to gather the pieces of the boundaries which are not convex.

### 3 Direct Geometric Tomography

Direct GT consists in performing the following steps:

- Perform edge enhancement into the sinogram:  $I(U, \Theta) \Rightarrow I_e(U, \Theta)$ .

- Detect an oriented dual curve  $\vec{C}^*(u, \theta) = 0$  in  $I_e$ : only rising edges are selected, the result is a linked list of points, or a function in the dual space, corresponding to a continuous representation of an edge into the sinogram.
- Transform the oriented dual curve  $\vec{C}^*(u, \theta) = 0$  of  $\vec{E}^*$  into a reconstructed curve  $\vec{C}(x, y) = 0$  of  $\vec{E}$ , corresponding to a boundary of the object.

Let us see now each of those steps.

### 3.1 Edge enhancement

Conventional edge detectors can be used in order to detect the edges in the sinogram, but we have designed more specific edge detectors for our purpose. If we suppose that the boundaries of the objects are smooth, except for a few locations where singularities occurs, then the projection of the edges is closer to an arc of circle than to a perfect step. An edge is traditionally detected as a peak of the first derivative (gradient), or as a zero crossing of the second derivative (laplacian) of the image. This is true for a perfect step, but in the case of an arc of circle, the edge is more efficiently detected for example as an extremum of the laplacian. We have also considered other techniques, like matching straight lines or pieces of circles to the intensity function  $I(U, \Theta)$  into the sinogram, in order to detect a swift change of the slope of the signal. This last approach gave us the best results (see figure 6), with subpixel precision for the position of the edges. We call  $I_e(U, \Theta)$  the sinogram after edge enhancement.

### 3.2 Following edges into the sinogram

This problem is similar to following roads in airplane pictures, vessels in an angiogram, or the trajectories of particles in a particle system. However, we can limit our study to only entirely convex (or concave) parts of the reconstructed curve. Then, the oriented dual curve is defined with a function  $u = f(\theta)$  into the sinogram. Methods like dynamic programming, or even much more local algorithms can be efficiently applied. Furthermore, the slope of those functions is limited, which helps to select the good curve to follow,

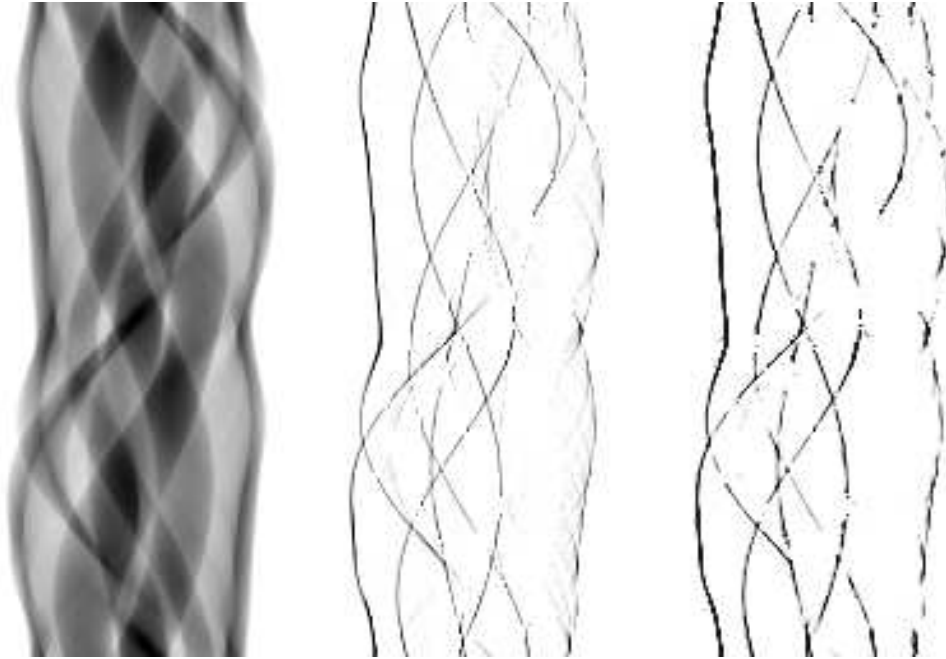


Figure 6: Sinogram (left), Laplacian (middle), Pattern matching (right). The resolution is 128x256

even when there is a crossing. If there are  $r$  detectors in the scanning line, and  $n$  (sampled between 0 and  $\Pi$ ) different viewing angles, the slope of a function  $u = f(\theta)$  corresponding to a dual curve is limited to  $\pm\Pi \cdot r/2 \cdot n$ .

The general approach is to follow the crest of local maximum in  $I_e(U, \Theta)$ . To go from one point to the next, we use two sometimes competing constraints : a small variation of the slope  $du/d\theta$  of the dual curve and a small distance between the dual curve and the local maxima of  $I_e$ . We reinforce the criterion of closeness to the local maximum when there is no ambiguity on its position, and we reinforce the criterion of small variation of the slope otherwise.

Global minimization techniques (like the use of “snakes” [12]) can help to improve the result. We match splines to the dual curve, in order to filter the noise, and to improve its location to subpixel values. Subpixel positioning can be achieved if the dynamic of the sinogram allows it (this is possible because we are looking for continuous edges, and not for pixels or voxels values. It is effective for sparse, high contrasted objects).

### 3.3 From dual curves to reconstructed curves

We have now a description of the oriented dual curves in  $\vec{E}^*$ . It can be linked lists of points, or splines of arbitrary degrees. In fact it can be any function  $\vec{C}^*(u, \theta) = 0$ . We have found several different ways to compute the corresponding reconstructed curves in the oriented object space  $\vec{E}$ , which characterize different implementations of direct GT. Some are purely geometric solutions, others are “signal processing” solutions.

#### 3.3.1 Related works

Recently, considerations about a continuous representation among successive camera views gave rise to new developments in the field of Computer Vision. Bolles and Baker [4] use a linear camera motion in order to determine the positions of the objects in the environment of a robot. Their results have been extended by Marimont in [14] to the case of arbitrary camera motions in the plane. Marimont shows that an image, called EPI (Epipolar-Plane Image, somehow equivalent to our sinograms) can be resampled into a “dual scene”, where stationary points in object space are transformed into straight lines and conic curves into other conics. A stationary point is a cusp, a



“wedge” in the object space, a location where a complete set of tangents can be defined (because the curvature is infinite). His method is suitable in the case of wedges, but gives poor results in the case of curved surfaces. Fortunately, wedges are generally the important information in the natural environment of a robot (which is composed with polygonal objects like chairs, tables, computers . . .). Informations coming from the reflectivity of the light on materials, like painted lines on the objects, can also be regarded and processed as wedges.

The need is different in the case of Xray tomography, because objects like bones or organs have smoothed, curved boundaries, with very few cusps, and there is no informations about the reflectivity of those surfaces. In our approach, we do not resample the sinogram but we directly track the edges of the objects into the original set of projections. Another major difference is that we reconstruct directly a linked list of points corresponding to the surface, instead of isolated, stationary points.

Nevertheless, the “philosophy” of the approach is the same, in that we are interested in a continuous representation of the objects along the pass of our “camera”. Another example is the work of H. Plantiga and C.R. Dyer in ([16]) which introduce the *aspects representation*, as a representation of all the possible views of the objects. The aspect representation is used to compile all the informations about occlusion, and can be used for object recognition.

An important related work as been presented by P. Giblin and R. Weiss in [8]. After a mathematical study of the profile of a surface for a single orthogonal projection, they show how to compute local characteristics of the surface with a sequence of views, such as the Gauss and mean curvature. They also point out that the profile of a non convex surface generates singularities, such as “swallow tails”, where the cusps corresponds to the inflection points of the original curve. Another important work has been performed by Regis Vaillant and Olivier Faugeras in [18]. They derive a local geometric transform from the *envelope theorem* which allow them to reconstruct 3D surfaces from 2D profiles. Their goal was to classify the different kinds of edges into the images (edge, wedge, profile), and to find back the shape and the differential properties of simple surfaces. Their main problem was the lack of precision on the registration of their cameras, which leads to severe artefact for the reconstructed shapes. Similar works can be found in [1],[6]. In the next section, we present a reconstruction method based also on a lo-

cal geometric transform. Fortunately, impreciseness are generally less severe for tomography than between different camera views. Furthermore, we show how a global minimization along the dual curve improves the results.

Some work has also already been performed in the field of Computed Tomography, about the direct use of the sinogram in order to compute the approximate position of a convex object. For example, Bergstrom et al. proposed a method in [2] to roughly locate the skull in order to compute a better approximation of the absorption coefficients for PET image reconstruction (Positron Emission Tomography). They determine the edges of the intensity function into the sinogram corresponding to the convex hull of the skull, and compute the corresponding back projected lines in the object space. Then, they determine points on the external boundary of the skull as the intersections of lines starting from the center of the rotation, with the back projected lines. We have classified this kind of approach in the next section “computational geometry methods”. We propose another solution, which lead to the exact, computationally optimal solution for the intersection of the back projected lines. Another method have been proposed by Hosoba et al., this time in order to compute the absorption coefficient for SPECT data (Single Photon Emission Computed Tomography) in [11], based on the accumulation of the back projected lines onto a regular grid in the object space. We present a similar approach in the section “back projection of the edges”.

### 3.3.2 Computational Geometry methods

Those methods will refer to the properties of the convex sets. As we saw in previous sections, a curve  $u = f(\theta)$  of the oriented dual space  $\vec{E}^*$  corresponds to an infinite set of half planes in  $\vec{E}$ , whose intersection corresponds to a convex (or concave) part of a boundary of the object. Those solutions are global, because the reconstructed curve depends on the entire set of points of the dual curve.

There is however a pitfall for those methods : we wish to have as many different projections as possible in order to reconstruct more accurately the object, and therefore we want to have a small step  $\delta\theta$  between consecutive views. But the smallest  $\delta\theta$  is, the more imprecise the intersection of the two lines will be. Taking successive samples of a dual curve and computing successively the intersection points leads to serious artefacts. The next two solutions answer to this problem. We already saw a first solution, proposed

by Bergstrom et al. in [2]. It is an approximate solution which allows to compute only points belonging to the polygon defined with the set of half planes. We propose a second method which is computationally optimal, and leads to the exact determination of the polygonal line. In both cases, the dual curve is sampled into a list of points in the oriented dual space  $\vec{E}^*$ , each point corresponding to an oriented line (or an half plane) in the object space  $\vec{E}$ .

**Algorithm** (see figure 7):

- Step 1 (initialization): We have to process  $n$  lines  $l_i$ , with  $i \in [1, n]$ , defined with their starting points  $p_i$ , and their directions  $\theta_i$ . We set an index  $j$  to 0.  $j$  will be the number of nodes (0 at this step), of the reconstructed polygonal line. We set another index  $k$  to 2.  $k$  is the index of the next line to be processed (line  $l_2$  at this step).
- Step 2: We set the values of  $l_2$  with those of  $l_k$  (if  $k > 2$ ). We compute the intersection point  $P_1$  between the line  $l_1$  and the line  $l_2$ , and also the parameters  $\gamma_1 = |p_1 \vec{P}_1|$  and  $\gamma_2 = |p_2 \vec{P}_1|$ , of the intersection along those two lines.  $j$  and  $k$  are then incremented.
- Step 3: Suppose that we have processed the values  $(P_i, \gamma_i, \gamma_{i+1})$  for  $i \in [1, j]$  corresponding to the successive intersections between the lines  $l_i$  and  $l_{i+1}$ . If  $k > n$ , then the process is stopped, and the polygonal line is defined with the  $j$  points  $P_i$ , with  $i \in [1, j]$ . Else let us consider a new line  $l_k$ . The intersection of the line  $l_{j+1}$  with the line  $l_k$  leads to  $(P', \gamma', \gamma'')$ . We compare then  $\gamma_{j+1}$  with  $\gamma'$ .
- Step 4:
  - If  $\gamma_{j+1} < \gamma'$  (in fact, if there was no error on the accuracy of the position of the dual curve, only this branch of the algorithm is executed): We set the values of the line  $l_{j+2}$  with those of the line  $l_k$  (if  $k > j + 2$ ). We set also  $(P_{j+1}, \gamma_{j+1}, \gamma_{j+2})$  with  $(P', \gamma', \gamma'')$ , we increment  $k$  and  $j$ , and we are back to step 3.
  - Else  $\gamma_{j+1} \geq \gamma'$ . This means that  $l_{j+1}$  is outside the intersection of the two half planes defined with  $l_j$  and  $l_k$ , and therefore  $l_{j+1}$  must be discarded. If  $j = 1$ ,  $j$  is decremented, and we are back to step

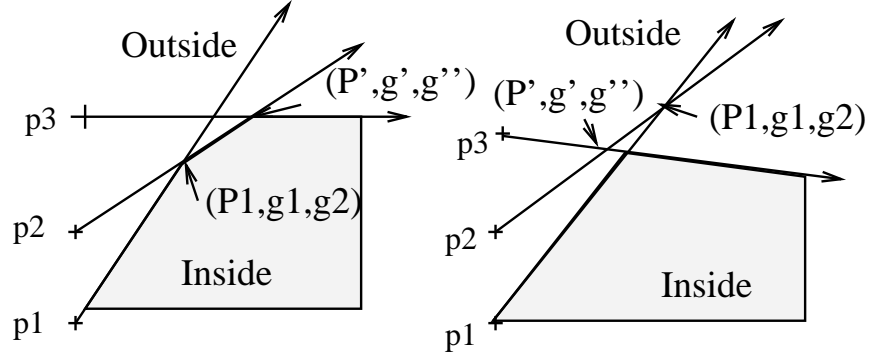


Figure 7: Intersection of three half planes

2. Else we compute the intersection between the line  $l_j$  and the line  $l_k$  which gives a new  $(P', \gamma', \gamma'')$ . We decrement  $j$ , and we are back to step 4.

This algorithm is optimal because it leads to the exact computation of the convex polygon defined as the intersection of the set of half planes. Each line can be considered at most twice: the first time it is processed (incrementation of  $k$ ), and once if it is back tracked and therefore eliminated (decrementation of  $j$ ). Thus this algorithm has a linear  $O(n)$  complexity. The complexity of the output is  $O(n)$ , so this algorithm is optimal. The key point is to notice that the lines are already sorted with respect to their directions, because the angle  $\theta$  always increases between one line and the next one. Any algorithm working with a randomly distributed set of half planes will have at least a  $O(n \log n)$  complexity, because the sorting step can't be avoided.

This method leads to subpixel positioning (see 8). The only problem is that any slight error on the position of a point of the dual curve “eats” a part of the reconstructed object. Therefore, the reconstructed curve tends to be inside the object for convex parts, and outside for concave parts. We reduce this phenomenon by the way of a global smoothing along the dual curve.

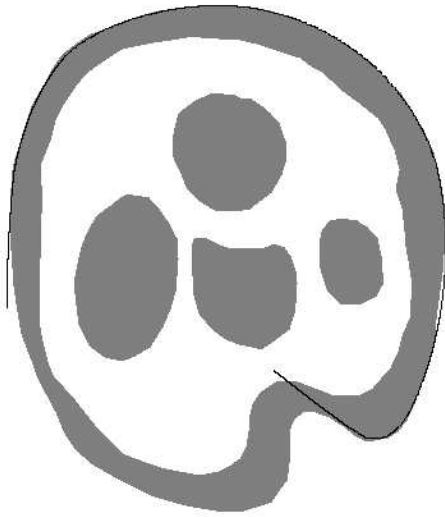
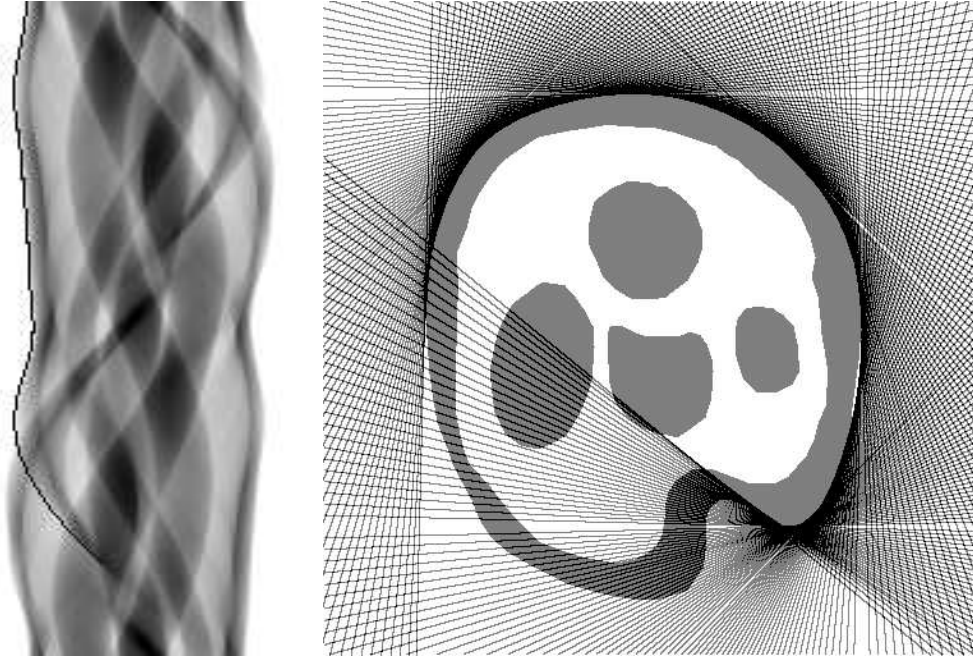


Figure 8: Computational geometry solution: sinogram (up left 128x256), half planes (up right 512x512) and reconstructed line (down 512x512)

### 3.3.3 Global geometric transform

The purely geometric solution is to find the implicit equation of the curve  $C(x, y) = 0$  in object space which corresponds to the implicit equation of the dual curve  $C^*(u, \theta) = 0$  in the dual space. Those couples of curves have been studied by the geometers for a long time, but formal solutions are sometimes difficult to implement on a computer. An intermediate solution is to match pieces of curves to the dual curves, with known corresponding representations in the object space, as for example pieces of sinusoids in dual space, which corresponds to pieces of circles in the object space. Until now, we have not fully investigated this field.

### 3.3.4 Local geometric transform

Any local geometric transform (point to point transform) between a dual curve in  $E^*$  and its reconstructed curve in  $E$ , has to take into account the differential characteristics of the curves. The relation between the dual curve in the dual space and the reconstructed curve in the object space can be written:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \sin(\theta) & -\cos(\theta) \\ \cos(\theta) & \sin(\theta) \end{pmatrix} \begin{pmatrix} u \\ du/d\theta \end{pmatrix} \quad (1)$$

Where  $(u, \theta)$  are the coordinates of a point  $P^*$  of the dual curve in the sinogram,  $du/d\theta$  is the slope of the curve according the axis  $\theta$ ,  $(x, y)$  are the coordinates of the corresponding point  $P$  in the object space, and  $\theta$  is the angle of the tangent to the curve at  $P$ . The formula for the 3D case can be found in [18]. As we can see, the accuracy of the reconstructed curve directly depends on the accuracy of the position of the curve into the sinogram, but also on the knowledge of the slope of the dual curve. The precision of the position is easy to settle, because it is related to the resolution of the sinogram. But the precision of the slope of the dual curve depends on the way it is computed. The local transform is in fact a 3D transform between  $P^* : (u, \theta, du/d\theta)$  and  $P : (x, y, \theta)$ .

Therefore, the crucial point in those methods is to determine the first differential characteristics of the curves. We already know that the slope of a dual curve is bounded with  $\pm\Pi \cdot r/2 \cdot n$  (see section 3.2). If  $r = n$  (the number of projection and the number of detectors of the scanning line is the

same), and if we want to reconstruct a curve with a precision of  $1/r$ , then the precision required for the slope is  $\pm\Pi/2 \cdot r$ . The slope can't be derived simply from two successive points of the dual curve, because, if the accuracy of the dual curve is  $\pm 1/2$  pixel of the sinogram, the difference along the axis  $U$  between two consecutive lines of the sinogram, for the position of a dual curve is of the order of  $\pm 1$  pixel, which prevents any direct use of a local geometric transform.

The direction of the gradient  $(dI(u, \theta)/du, dI(u, \theta)/d\theta)$ , where  $I(u, \theta)$  is the intensity function of the sinogram, can help to improve the local determination of the slope of the dual curve, but the precision on the direction of the gradient depends also on the dynamic of the sinogram. We found that the gradient information generally do not suffice to give the required precision. However, the norm of the gradient can be used as an estimate of the "confidence" we can have in this direction. As it can be expected, the precision of the reconstruction depends on the contrast between the object and the "background" in the sinogram. A global smoothing all along the dual curve is an absolute necessity, which helps to improve both the location of the curve and the value of the slope. This method gave us the best results in terms of accuracy.

Nevertheless, the theoretical precision of the reconstructed curve is very difficult to settle. Three different factors contribute to give this precision. First, the step of edge enhancement can lead to subpixel positioning of the estimated dual curve into the sinogram. This precision directly depends on the contrast between the projection of the object and the background, and therefore it is not independent of a model of the objects. Second, the total amount of information is distributed all along the dual curve. The global minimization of the curvature modify the distribution of this information, in a way which is difficult to modelize. Third, the arc length of the dual curve directly depends on the number  $n$  of projections, whereas the arc length of the reconstructed curve depends on the size of the object. For example, if we set the scale of the object space in accordance with the resolution of the line of detectors, if  $r = n$ , and if the object is a disk of radius  $a$ , then its arc length is  $s = 2 \cdot \Pi \cdot a$ . The arc length  $s^*$  of the oriented dual curve do not depends on  $a$ , but only on the distance  $b$  between the center of the disk and the center of the rotation of the scanner:

$$s^* = (n/\Pi) \cdot \int_0^{2\Pi} (1 + b \cdot \cos(\theta))^{1/2} d\theta \geq 2 \cdot n \quad (2)$$

Therefore, if  $a < n/\Pi$ , the density of the information along the reconstructed curve is greater than along the oriented dual curve. This means that the boundary of a small, smooth, and high contrasted object can be reconstructed with an accuracy which can significantly exceed the resolution of the sinogram (furthermore, for regularly spaced detectors, this accuracy increases when the object goes away from the center of the rotation, which is unintuitive). This legitimates our search of subpixel positioning of the reconstructed curves. At last, the concavities of the object seriously increases the arc length of the oriented dual curve, and therefore the precision of the reconstructed curve increases with the number of concavity (but only if the projection of the object is sufficiently contrasted, so that it is possible to settle the position of the dual curve). Thus many factors contribute to the final precision of the reconstructed curves, such as the dynamic of the sinogram, the contrast, the shape, and the position of the object, and also the influence of other objects. Our method gives its best results for sparse, smooth, homogeneous and high contrasted objects, like bones for example.

### 3.3.5 Back projection of the edges

This approach is a signal processing approach, because, unlike previously described methods, it uses a discrete image representation in object space, in order to accumulate the result. Such a method is sometimes called a *Hough method*. Once again, we sample the curves  $\vec{C}^*(u, \theta) = 0$  of  $\vec{E}^*$  into a set of points and we back project those points into the object space. We accumulate for each pixel of a grid in object space the contributions of all the lines which cross this pixel. The contribution of a line depends both on the confidence coefficient computed at the step of edge following, and on the relative position between the line and the pixel (for example the distance from the line to its center). Then we follow once again a crest of local maxima, but this time directly into the object space  $\vec{E}$ . Our method is close to [11], but is not restrained to the computation of the external shape of the object, neither than to a convex object. Nevertheless, this approach has several disadvantages compared with previously described methods.



- This approach is computationally more expensive than previous ones, because, for each line of the dual curve, we have to process all the pixels which are intersected by the line, therefore, if  $r = n$ , the complexity of this method is  $O(n^2)$  for each dual curve, instead of  $O(n)$ .
- As for any other Hough methods, the problem is to determine the resolution of the grid in object space. As the resolution of the grid increases, the accuracy increases, but the curve in object space is much more difficult to reconstruct, because the probability of the accumulation of several ( $\geq 3$ ) lines for the same pixel decreases to 0. Conversely, if the resolution of the grid decreases, the curve is more efficiently detected, but the accuracy of the localization decrease.
- Two different curve following pathes have to be applied; one in dual space  $E^*$  and one in object space  $E$ , whereas previous methods require only on following path in  $E^*$ .

Figure 9 is the back projection of all the edges detected into the sinogram.

### 3.4 Conclusion for Direct Geometric Tomography

Direct GT can be applied without any a-priori information. The implementation based on the local geometric transform seems to lead to the best results, but the “Computational Geometry” solution seems also interesting. The major problem is topological errors, which lead sometimes to the reconstruction of the convex hull of non-convex objects: those problems have to be solved with a higher level procedure. Nevertheless, subpixel positioning can be achieved.

## 4 Inverse Geometric Tomography

### 4.1 General principle

The general principles of Inverse GT is to start with an a-priori estimate of the reconstructed curve  $C_0$  in object space, and to iteratively improve the location of this curve  $C$ , with the search of the best matching between the corresponding dual curve  $C^*$  and the real data into the sinogram  $I_e(U, \Theta)$ .

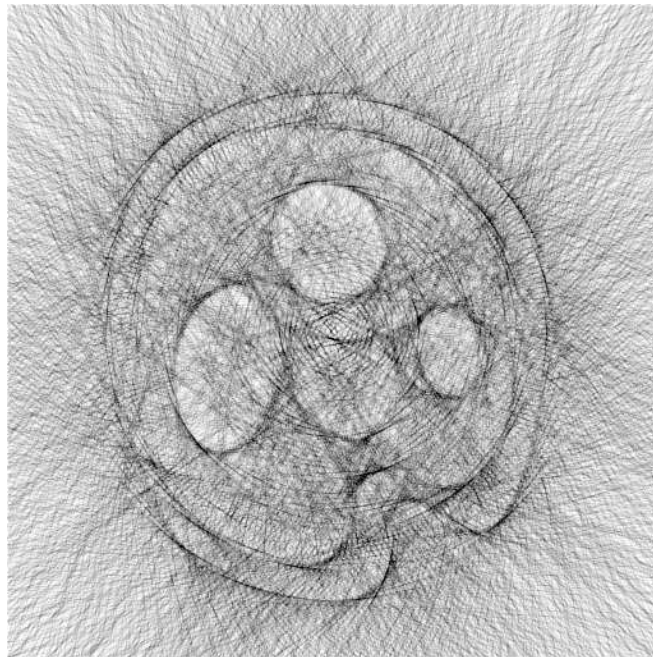


Figure 9: Back projected lines

This approach is a special kind of *deformable model*, as they are described for example in [12]. Deformable models are used to find a curve (or surface) which both minimize some kind of “distance” between the estimated curve, and the real data (called *external energy*), and minimize internal constraints (called *internal energy*) along the curve, such as the curvature. Inverse GT is an iterative process which try to minimize the external “energy” of the curve  $C^*$  in the dual space  $E^*$ , and to minimize the internal “energy” of the curve  $C$  in object space  $E$ .

## 4.2 Example of implementation and results

We present here an example of implementation. This is a simple iterative algorithm where we alternatively reduce the external “energy” in the dual space and the internal “energy” in the object space. An iterative algorithm minimizing both internal and external “energies” at the same time for each iteration is under current investigation.

We start with a first approximate curve  $C_0(x, y) = 0$  in object space of the curve. As we saw in previous sections, differential properties are needed in order to relate the dual and the object space. We regularly sample  $C_0$  into a linked list of points, where each point contains the position  $(x, y)$ , and the direction  $\theta$  of the tangent (similarly, a point of the dual curve is defined with its position  $(u, \theta)$ , and the angle of the tangent, which gives the slope  $du/d\theta$  of the curve). We have then the following algorithm:

- Step 1: Give a first curve  $C_0$ , index  $i$  is set to 0.
- Step 2: Transform  $C_i$  into its dual curve  $C_i^*$  (transforming a point  $P : (x, y, \theta)$  into a point  $P^* : (u, \theta, du/d\theta)$  is straightforward).
- Step 3: Find a new curve  $C_i'^*$  “closer” to the edges in  $I_e : (U, \Theta)$  than  $C_i^*$  (this reduce the external “energy”). A residual error can be computed at this step.
- Step 4: Transform back  $C_i'^*$  into a new curve  $C_i'$  in object space. This transform has been previously referred to as the local geometric transform.

- Step 5: Apply a filter to the curve  $C'_i$  in order to smooth high frequency variation of the curvature (which reduce the internal “energy”). This gives a new curve  $C_{i+1}$ .
- Step 6: The process is stopped if the residual error is bellow a given threshold, or if a maximum number of iteration is reached, else we are back to step 2.

This is a very fast method, few iterations are needed to converge toward a stable solution, and the operations to perform at each step are very simple (see results in figure 10 for synthetic data and figure 11 for real data). We give now a way to compute a point  $P'^* : (u', \theta', du'/d\theta)$  of  $C'^*_i$  from a point  $P^* : (u, \theta, du/d\theta)$  of  $C^*_i$ : we apply an edge detector at point  $P$  of the sinogram, according to the direction of the normal to  $C^*_i : (-1, du/d\theta)$ , which gives a shift  $(\delta u, \delta\theta)$ . We have no other information on the direction of the new curve, we make the approximation that the slope at point  $P'^*$  remains the same as the slope at point  $P^*$ , therefore  $P'^* : (u + \delta u, \theta + \delta\theta, du/d\theta)$ . Then  $P'^*$  is transformed back into a point of the object space  $P'_i : (x + \delta x, y + \delta y, \theta + \delta\theta)$ , with a local geometric transform of the shift:

$$\begin{aligned} \delta x &= y \cdot \delta\theta - \cos(\theta) \cdot \delta u \\ \delta y &= -x \cdot \delta\theta + \sin(\theta) \cdot \delta u \end{aligned} \tag{3}$$

Inverse GT gives the same precision than the local geometric transform, because it use the same edge detector as the first step of direct GT. However, Inverse GT start with a first approximation  $C_0$  of the curve. If a concavity exists in  $C_0$ , this indicates that a swallow tail is to be found into the sinogram, and gives an approximate location of this swallow tail. In short, Inverse GT prevents from performing some of the topological errors of direct GT.

### 4.3 Tracking

But the most efficient application of Inverse GT is the tracking of the boundary of an object along the time. The straightforward implementation is to apply Inverse GT at time  $t + \delta t$ , and to use the curve reconstructed at time  $t$  as the new first approximate curve  $C_0$ .

Let us see now another, even more local way to compute the new curve, dedicated to the tracking. Suppose that we have already computed all the

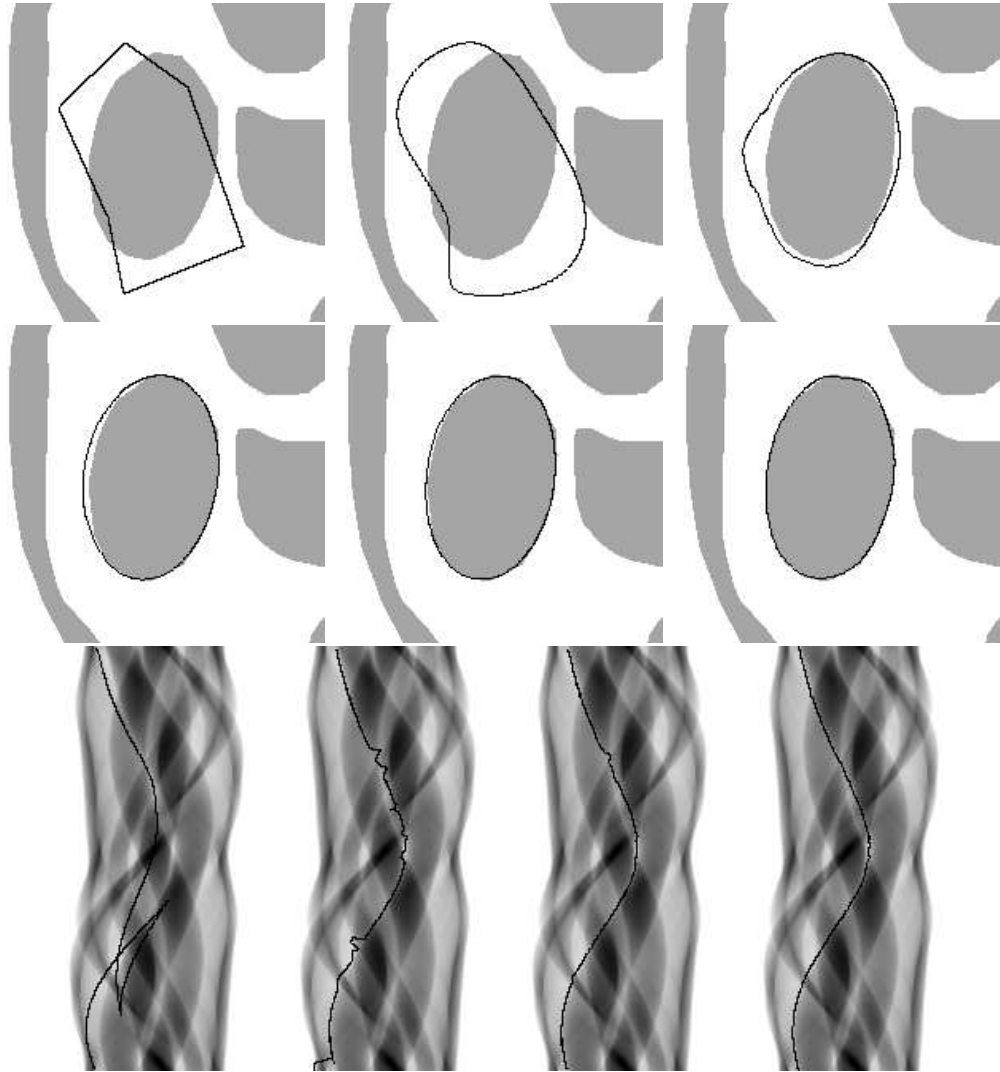


Figure 10: Original control points (given by the user), and iteration 1,2,3,4,10 in object space. Iteration 1,2,3,10 in dual space (left to right, up to down). Images in object space corresponds to a window 200x200 in the original 512x512 synthetic image, whereas sinograms have been computed with resolution 128x128, therefore one pixel precision in object space corresponds to  $1/4^{th}$  pixel precision with respect to the original data set.

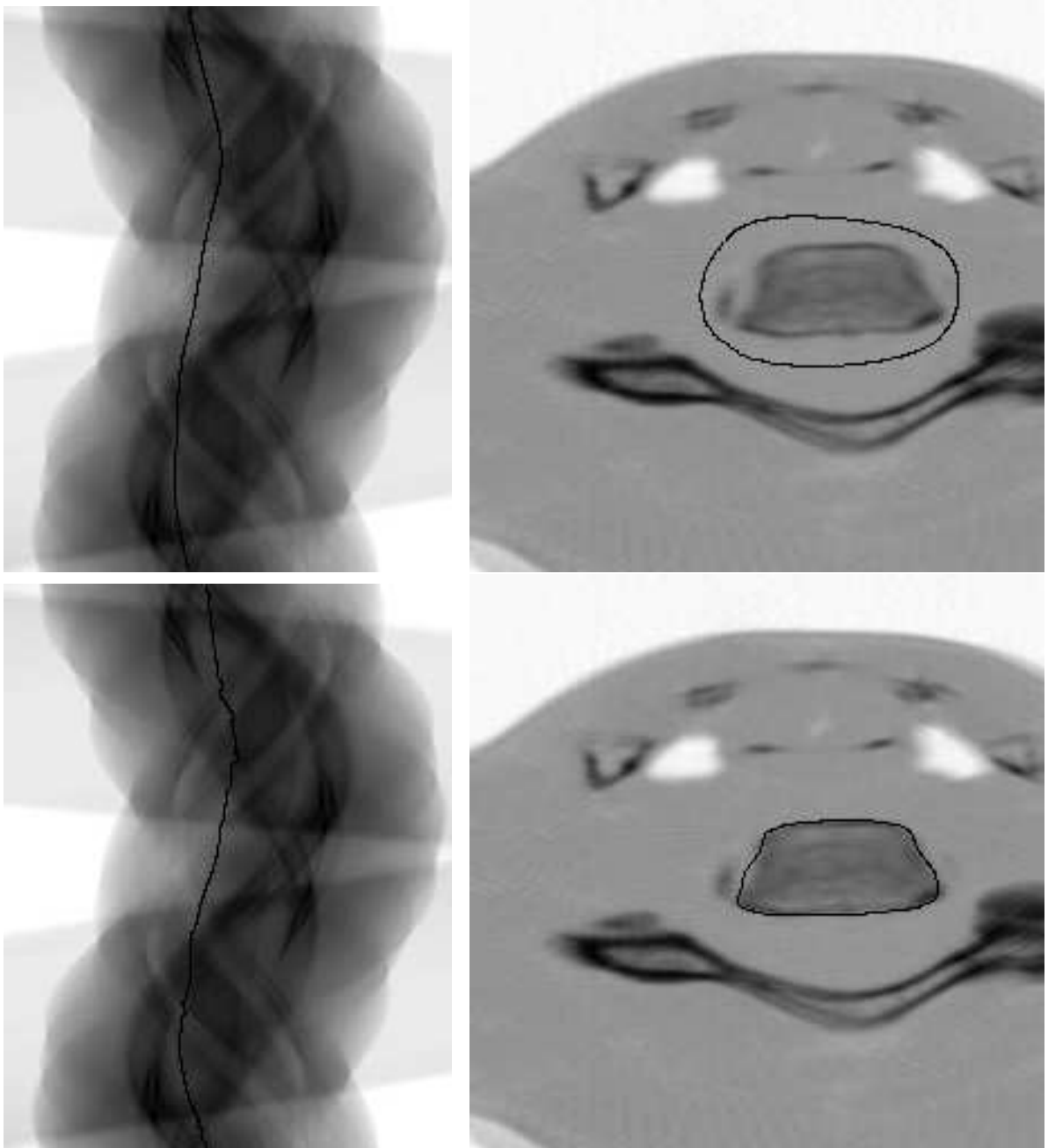


Figure 11: Real data: First estimate and result after 30 iterations, superimposed with a reconstructed image, and with the corresponding sinograms. The approximate diameter of the reconstructed object is 17mm, and the field of the scanner is about 500mm. A pixel in the reconstructed image (256x256) corresponds to about 0.2mm only ...

family of curves  $C_{t-T}, \dots, C_{t-\delta t}$  in  $E$ , representing the boundary of the object between  $t-T$  and  $t-\delta t$ , where  $T$  is the period of the rotation of the scanner, and  $\delta t$  is the interval of time between two consecutive views. Suppose that each of these curves is sampled with  $n$  points ( $\delta t = T/n$ ). We want to compute a new curve  $C_t$ . This new curve will have all the points  $P_i$  of the previous curve  $P_{t-\delta t}$  except point  $P_{t-T}$ , which is replaced with a new point  $P_t$ . We compute the dual point  $P_{t-T}^*$  in the sinogram, in  $E^*$ , corresponding to  $P_{t-T}$ , and we determine a new point  $P_{t-T}^{*}$  which locally reduce the external “energy” of the curve, as it was previously described for inverse GT. The local geometric transform of  $P_{t-T}^{*}$  gives a new point  $P_t'$  in the object space  $E$ . We determine also another point  $P_t''$  in  $E$  corresponding to the extrapolation of points of the curve  $C_{t-\delta t}$  around  $P_{t-T}$ , taking also into account some differential characteristics such as the tangent direction, or the curvature of the curve, this locally reduce the internal “energy”. The new point  $P_t$  is a tradeoff between  $P_t'$  and  $P_t''$ , the weights of each of those two points give a way to control the respective influence of the external and the internal “energy”.

This procedure is therefore a tremendous short cut between the acquisition of the data and the dynamic reconstruction of the shape. For example, if the period of rotation of the scanner is  $T = 10s$ , and the number of different views is  $n = 256$ , the curve is updated on  $1/256^{th}$  of its length in about  $1/25s^{-1}$ , which allows real time tracking of the boundaries of the objects. Inverse GT is a very local method which need only a few pixels into the sinogram (in  $E^*$ ) and around the previous position in order to update the curve. A similar method can be derived for traditional methods: the new position of the curve is computed with the reconstruction of a small window of the digital image in object space  $E$ , but each pixel of this local reconstruction needs a global integration along a sinusoid into the sinogram. Therefore, the complexity of the tracking with Inverse GT is at least one order less in  $n$  than traditional Computed Tomography methods, and do not require special hardware for real time implementation. Furthermore, the tracking of the curve is realized along a strip in the sinogram, which could be used to reduce the field of the beam of Xrays and therefore the total amount of radiation received during the tracking of a specific boundary.

## 5 Extensions of our method

For convenience, we described our method for a  $3^{rd}$  generation scanner, with parallel rays. The assumption of parallel rays can be replaced with an hypothesis of continuity between the ray directions and positions. Other formula of local geometric transform can be easily derived, depending on the exact geometry of the scanner, but the method remains the same.

Our results can also be extended to the case of a 3D scanner (sometime called  $5^{th}$  generation scanner). The sinogram is then a 3D digital image composed with the appending of 2D projections of the object. The principles of Direct and Inverse Geometric Tomography remain the same, except that we look for 3D oriented dual surfaces, and of course for the corresponding 3D surfaces in object space. For a point of a 2D projection corresponding to the profile of the object, the tangent of this profile in the 2D projection, together with the direction of the projection define a plane which is tangent to a boundary of the object. The reconstructed surface is then defined as the envelope of its tangent planes, a local geometric transform can be derived from the “envelope theorem” as it is shown in [18]. For “computational geometry” solutions, parabolic sections of the reconstructed surface are defined as the intersection of the corresponding 3D half spaces. Taking into account the coherence of the dual surfaces in a three (or four) dimensional dual space  $E^*$  may lead to a tremendous improvement of our method.

An application of our method is for surgery planning. As we saw, Geometric Tomography allows real time tracking of the boundary of the object. The associated oriented dual shapes can be directly used for radiotherapy planning and intervention: the beam used for radiotherapy can be approximated with a strait line, therefore, it can be represented as a point into the sinogram. Because any real beam is somewhat divergent, it is better approximated with a spot in the dual space  $E^*$ , that we call the *dual spot* associated to the beam. The relative position of the dual spot with respect to the oriented dual shapes states immediately what the objects intersected by the beam are.

Another extend of our method is its application to the registration of a predefined object with a real object. The goal is for example to automatically give labels to the objects, or to detect differences between a previous and a new scan of the same object. Instead of transforming the true signal into a digital image in the object space, and comparing this image with the generic



object, it is sometime possible to compute a synthetic image of the generic objects into the parametric space of the signal, and to directly compare the synthetic image with the true signal. If the predefined object and the real object are close to each other, Inverse GT can be used to find a point to point correspondence between the predefined and the real object. However, registration between distant objects could be difficult to solve in the dual space, because a translation in object space is not a translation in dual space. Perhaps the study of some characteristic points such as the direction of the tangents for inflexion points may help to find the transform between the two objects.

We proposed in this paper new methods to compute the boundaries directly from the original set of projections produced with the scanner, without any help of Computed Tomography. Our methods have been tested with both synthetic and real data, but for 2D sinograms only. We expect even better results with 3D sinograms. Another extension of our method, mixing both GT and CT techniques can also be foreseen.

## 6 Conclusion

We hope to have given an unexpected and original insight into some of the geometric aspects of tomography. Of course, “Geometric Tomography” is not to replace Computerized Tomography, because the first method gives back geometric surfaces, whereas the second one produce digital images. CT is a global method of reconstruction which has been optimized in order to give the most suitable images for diagnostic, whereas GT is a local method of reconstruction, which has been optimized in order to settle with few computation the external and internal boundaries of the objects. As we saw GT gives very good results for sparse objects of homogeneous, high density, like bones, or blood vessels in angiograms, and can be also efficiently used for tracking. It is a tremendous short cut between data acquisition and processing.

To conclude, there have been many works about tomography reconstruction, but mainly analytic and algebraic solutions were provided, because people were looking for digital images. If we are looking for the boundaries of the objects, and no more for digital images, then geometric solutions have also to be considered.

## Acknowledgment

I would like to thank Nicolas Ayache, Jean-Marie Rocchisani, and Donald Geman for stimulating discussions about this new approach to tomography. We also thank Siemens France who provided us with some real sinograms.

## References

- [1] Emmanuel Arbogast and Roger Mohr. Reconstruction de surfaces à partir des contours critiques, dans un contexte multi-images. *Rapport de recherche du LIFIA, Institut IMAG, Grenoble, France*, June 1989.
- [2] M. Bergstrom, J. Litton, L. Ericksson, C. Bohm, and G. Blomqvist. Determination of object contour from projections for attenuation correction in cranial positron emission tomography. *J Comput Assist Tomogr*, 6:365–372, 1982.
- [3] Jean-Daniel Boissonnat. Shape reconstruction from planar cross sections. *Computer Vision, Graphics, and Image Processing*, 44:1–29, 1988.
- [4] Robert C. Bolles and Harlyn H. Baker. Epipolar-plane image analysis: A technique for analyzing motion sequences. *Proceedings of the Third Workshop on Computer Vision: Representation and Control*, pages 168–178, October 1985. Bellaire, Mich.,.
- [5] R.N. Bracewell and A.C. Riddle. Strip integration in radio astronomy. *Australian J. of Physics*, 9:198–217, 1956.
- [6] Roberto Cipolla and Andrew Blake. The dynamic analysis of apparent contours. *IEEE third international conference on computer vision*, pages 616–623, December 1990.
- [7] A.M. Cormack. Representation of a function by its line integrals, with some radiological applications. *J. Appl. Phys.*, 34:2722–2727, 1963.
- [8] Peter Giblin. Reconstruction of surfaces from profiles. *Proc. IEEE First Int. Conf. on Computer Vision*, pages 136–144, 1987.
- [9] R. Gordon. A tutorial on art (algebraic reconstruction techniques). *IEEE Trans. Nucl. Sci.*, 21:78–93, 1974.
- [10] G.T. Herman, A. Lent, and S. Rowland. Art: Mathematics and applications: A report on the mathematical foundations and on applicability to real data of the algebraic reconstruction techniques. *J. Theor. Biol.*, 43:1–32, 1973.

- [11] Minoru Hosoba, Hidenobu Wani, Hinako Toyama, Hajime Murata, and Tanaka Eiichi. Automated body contour detection in spect: Effects on quantitative studies. *Journal of Nuclear Medecine*, pages 1184–1191, june 1986.
- [12] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.
- [13] W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.
- [14] David H. Marimont. Projective duality and the analysis of image sequences. *Proc. IEEE First Int. Conf. on Computer Vision*, pages 7–14, 1986.
- [15] W.H. Oldendorf. Isolated flying spot detection of radiodensity discontinuities displaying the internal structural pattern of a complex object. *IRE Trans. Biomed. Eng.*, BME-8:68–72, 1961.
- [16] Harry Plantinga and Charles R. Dyer. The asp: A continuous viewer-centered representation for 3d object recognition. *Proc. IEEE First Int. Conf. on Computer Vision*, pages 626–630, 1987.
- [17] Johann Radon. Uber die bestimmung von funktionen durch irhe integralwerte langs gewisser mannigfaltigkeiten. *Ber. Saechs. Akad. Wiss. Leipzig, Math.-Phys. Kl. 69*, pages 262–277, 1917.
- [18] R. Vaillant and O. Faugeras. Using occluding contours for 3d object modeling. *International Advanced Robotics Program*, october 1989.