Contour-based models for 3D binary reconstruction in X-ray tomography

Charles SOUSSEN and Ali MOHAMMAD–DJAFARI

Groupe Problèmes Inverses
Laboratoire des Signaux et Systèmes
École supérieure d'éléctricité (SUPÉLEC)
Gif–sur–Yvette, France

First part: General contour-based models in X-ray tomography

Ali MOHAMMAD–DJAFARI

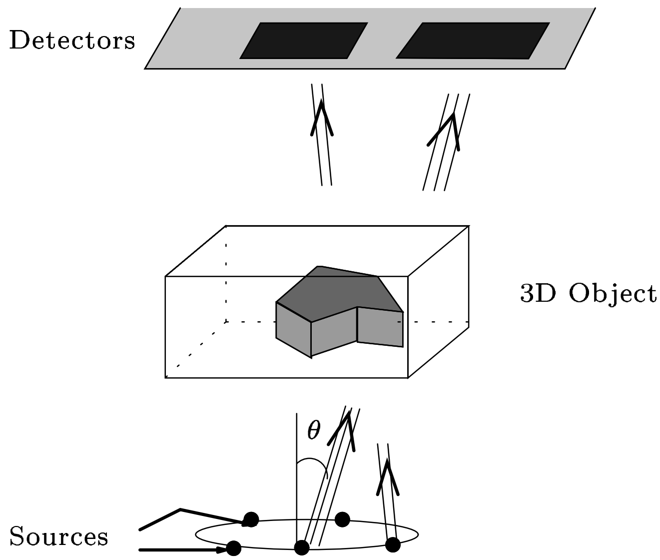
Discrete Tomography, Oct. 11-14, 2000, Siena, Italy

Contents

- Contexte and Application
- Data acquisition modeling
- Binary Voxel and Geometric Contour modeling
- Binary voxels modeling
- Contour based modeling
- Non parametric modeling:
  - Active contours or Snakes
  - Level set
- Parametric modeling:
  - Simple geometric shapes: ellipses, polygons, ellipsoides and polyedra
  - Affine transformation of simple geometric shapes
  - Harmonic modeling of contours (global basis functions)
  - Spline based modeling (local basis function)
  - Wavelet based modeling (Hybrid global and local basis functions)

Contexte and Application

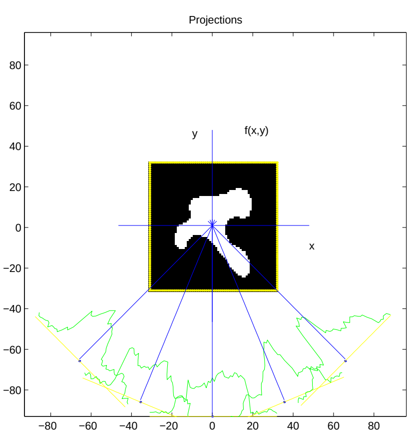
- Non Destructif Testing (NDT) ;
- Reconstruction of the shape of a defective area inside a metalic object.



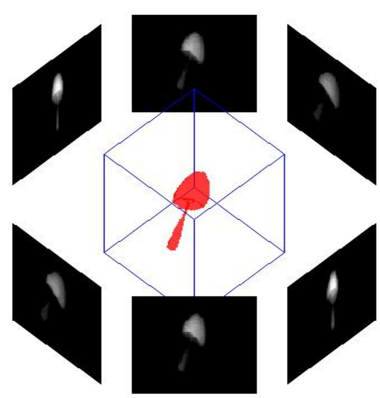
- Few radiographic data ($\leqslant 10$);
- Limited angle ($|\theta| \leqslant \pi/4$).

Data acquisition: 2D and 3D Tomography

2D



3D

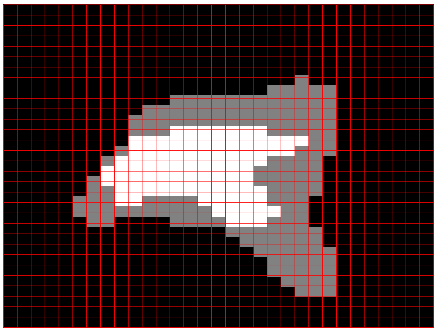


$$p_\theta(t) = \int_{\mathcal{L}_{t,\theta}} f(x,y)\,\mathrm{d}l \qquad p_{\theta,\phi}(t_1,t_2) = \int_{\mathcal{L}_{t_1,t_2,\theta,\phi}} f(x,y,z)\,\mathrm{d}l$$

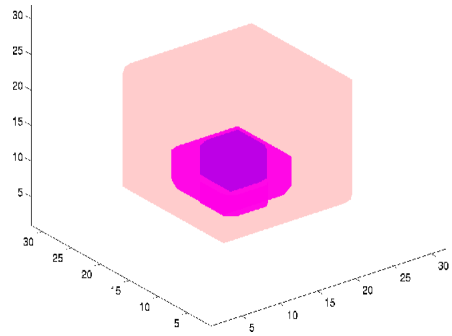
- Direct problem : Given $f(x,y,z)$ compute $p_{\boldsymbol{\theta}}(\boldsymbol{t})$
- Inverse problem : Given $p_{\boldsymbol{\theta}}(\boldsymbol{t})$ estimate $f(x,y,z)$

Discretized Tomography

2D



3D



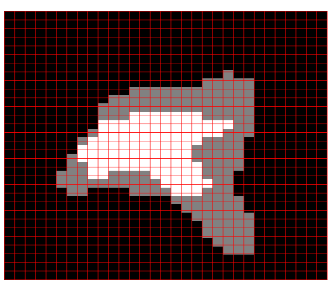
$$f(x,y,z) = \sum_{k,l,m} f_{(k,l,m)}\, b_{(k,l,m)}(x,y,z) \text{ with } b_{(k,l,m)}(x,y,z) = \begin{cases} 1 \text{ if } (x,y,z) \in \text{Voxel } (k,l,m) \\ 0 \text{ elsewhere} \end{cases}$$

$$p_i = \sum_j a_i(j)\, f(j), \qquad \text{with} \quad a_i(j) = \text{length of ray segment } i \text{ through voxel } (j)$$

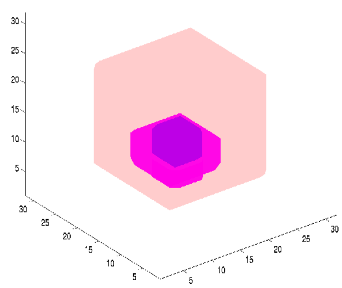
- Direct problem: Given $\boldsymbol{f}$ compute $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{f}$
- Inverse problem: Given $\boldsymbol{p}$ estimate $\boldsymbol{f}$

Discrete Tomography

2D



3D



$$f(x,y,z) = \sum_{k,l,m} f_{(k,l,m)}\, b_{(k,l,m)}(x,y,z) \text{ with } b_{(k,l,m)}(x,y,z) = \delta(x-k, y-l, z-m)$$

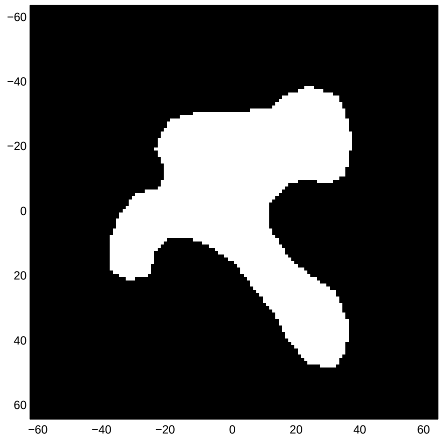
$$p_i = \sum_j a_i(j)\, f(j), \qquad \text{with} \quad a_i(j) = \begin{cases} 1 \text{ if ray } i \text{ goes through the voxel } j \\ 0 \text{ elsewhere} \end{cases}$$

$f(j)$ are also discrete valued variables

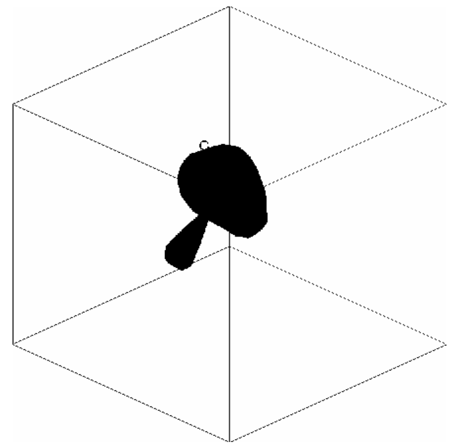
- Direct problem: Given $\boldsymbol{f}$ compute $\boldsymbol{p}$
- Inverse problem: Given $\boldsymbol{p}$ estimate $\boldsymbol{f}$

Compact object / Shape reconstruction

2D



3D



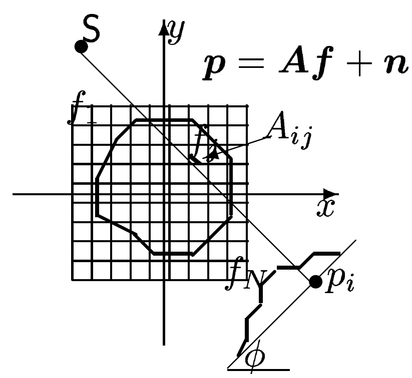
$$f(x,y,z) = \begin{cases} 1 & \text{if } (x,y,z) \in \mathcal{C}, \\ 0 & \text{elsewhere} \end{cases} \longrightarrow \quad p_{\boldsymbol{\theta}}(\boldsymbol{t}) = \int_{\mathcal{L}_{\boldsymbol{t},\boldsymbol{\theta}}} f(x,y,z)\,\mathrm{d}l = \int_{\mathcal{L}_{\boldsymbol{t},\boldsymbol{\theta}}} \mathrm{d}l$$

- Direct problem : Given $\mathcal{C}$ compute $p_{\boldsymbol{\theta}}(\boldsymbol{t})$
- Inverse problem : Given $p_{\boldsymbol{\theta}}(\boldsymbol{t})$ estimate $\mathcal{C}$

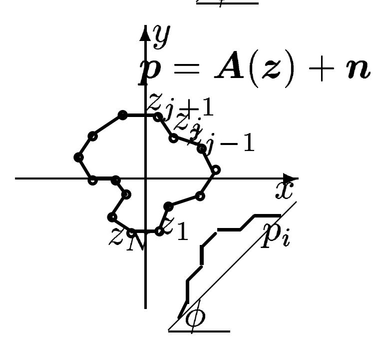
Binary Voxel and Geometric Contour modeling

- Two approches:

1. Binary Voxels modeling
   - (Gautier, Sauer, Dinten)
   - Linear direct problem: $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{f} + \boldsymbol{n}$
   - Inverse problem: Given $\boldsymbol{p}$ estimate $\boldsymbol{f}$
   - Number of parameters to estimate $= 1000^3$
   - Compactness of the defective area ?



2. Geometric modeling of the contour
   - (Battle, Chalmond, Garnero)
   - Non linear direct problem: $\boldsymbol{p} = \boldsymbol{A}(\boldsymbol{z}) + \boldsymbol{n}$
   - Inverse problem: Given $\boldsymbol{p}$ estimate $\boldsymbol{z}$
   - Number of parameters to estimate $\leqslant 500$



Voxels modeling

- ❑ Discretization of the object in N voxels; $\boldsymbol{f} = \{f_1, \ldots, f_N\}$.
- ❑ Linear direct problem: $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{f} + \boldsymbol{n}$.
  - Backprojection: $\widehat{\boldsymbol{f}} = \boldsymbol{A}^t \boldsymbol{p}$
  - minimize $\Omega(\boldsymbol{f}) = \|\boldsymbol{D}\boldsymbol{f}\|^2$ s.t. $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{f}$:

$$\widehat{\boldsymbol{f}} = \boldsymbol{A}^t \left(\boldsymbol{A}\boldsymbol{A}^t + \lambda \boldsymbol{D}\boldsymbol{D}^t\right)^{-1} \boldsymbol{p} \longrightarrow \text{Backprojection of filtered projections}$$

  - Quadratic regularization:
    $\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \{\mathcal{J}(\boldsymbol{f})\}$ with $\mathcal{J}(\boldsymbol{f}) = \|\boldsymbol{p} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda\|\boldsymbol{D}\boldsymbol{f}\|^2$

$$\longrightarrow \widehat{\boldsymbol{f}} = \left(\boldsymbol{A}^t\boldsymbol{A} + \lambda \boldsymbol{D}^t\boldsymbol{D}\right)^{-1} \boldsymbol{A}^t \boldsymbol{p}$$

  - More general regularization or Bayesian MAP estimation:
    $\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \{\mathcal{J}(\boldsymbol{f})\}$ with $\mathcal{J}(\boldsymbol{f}) = \|\boldsymbol{p} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda\Omega(\boldsymbol{f})$
    with appropriate $\Omega(\boldsymbol{f})$ corresponding to the energy of a Gibbsian or Markovian random field modeling.

Binary voxels modeling

- ❑ Discretization of the object in N **binary** voxels; $\boldsymbol{f} = \{f_1, \ldots, f_N\}$, $f_j \in \{0, 1\}$.
- ❑ Linear direct problem: $\boldsymbol{p} = \boldsymbol{A}\boldsymbol{f} + \boldsymbol{n}$.
  - Regularization: $\widehat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f} \in \{0,1\}^N} \{\mathcal{J}(\boldsymbol{f})\}$ with $\mathcal{J}(\boldsymbol{f}) = \|\boldsymbol{p} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda\Omega(\boldsymbol{f})$
  - Discrete optimization is too huge $\{0, 1\}^N$, $N = 1000^3$
  - Alternative:
    Choose appropriate $\Omega(\boldsymbol{f})$ using either a Gibbsian or Markovian random field modeling to enhance the binary caracteristics of the voxels
  - We propose a Markovian regularization + bacground pulling

$$\Omega(\boldsymbol{f}) = \lambda \sum_{i \sim j} \left|f_i - f_j\right|^\alpha + \mu \sum_i \left|f_i\right|$$

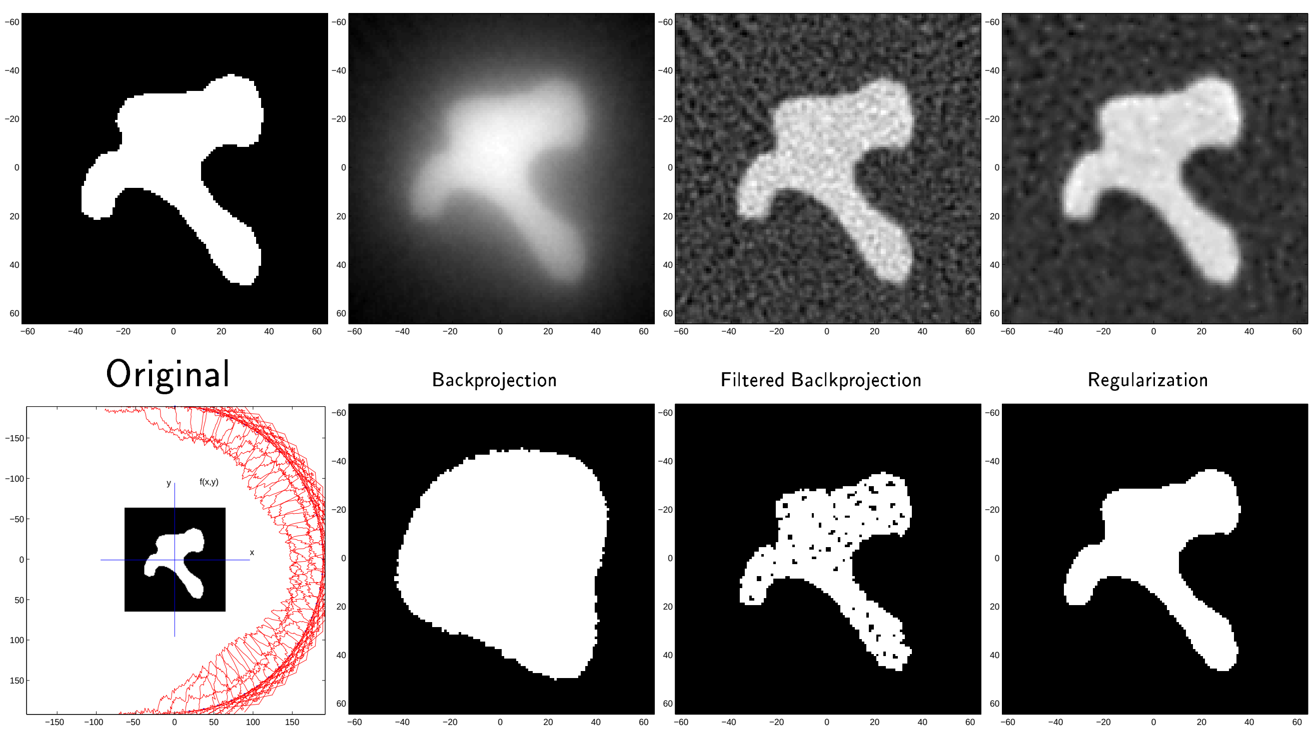
  - Optimization subject to positivity.
  - Compactness constraint $\longrightarrow$ too complex to account for.
  - Gradient based algorithem.

Voxel modeling: Multiresolution approach

- Mono-resolution:
  - Initialize $\boldsymbol{f}$ by $\boldsymbol{f}^{(0)}$
  - Optimize $\mathcal{J}(\boldsymbol{f}) = \|\boldsymbol{p} - \boldsymbol{A}\boldsymbol{f}\|^2 + \lambda\Omega(\boldsymbol{f})$
- Multiresolution:
  - Initialize with a very coarse grid $\boldsymbol{f}^{(0)}$ (Filtered backprojection), then $\forall r$,
  - $\boldsymbol{f}^{(r-1)} \longrightarrow \boldsymbol{f}_0^{(r)}$: 1 voxel $\Longrightarrow$ 8 sub-voxels with the same density.
  - $\boldsymbol{f}_0^{(r)} \longrightarrow$ Mono-resolution $\longrightarrow \boldsymbol{f}^{(r)}$
- Hyperparameters determination :

$$\mathcal{J}^{(r-1)}\left(\boldsymbol{f}^{(r-1)}\right) = \mathcal{J}^{(r)}\left(\boldsymbol{f}_0^{(r)}\right)$$

Simulated data: Easy and confortable 2D Case

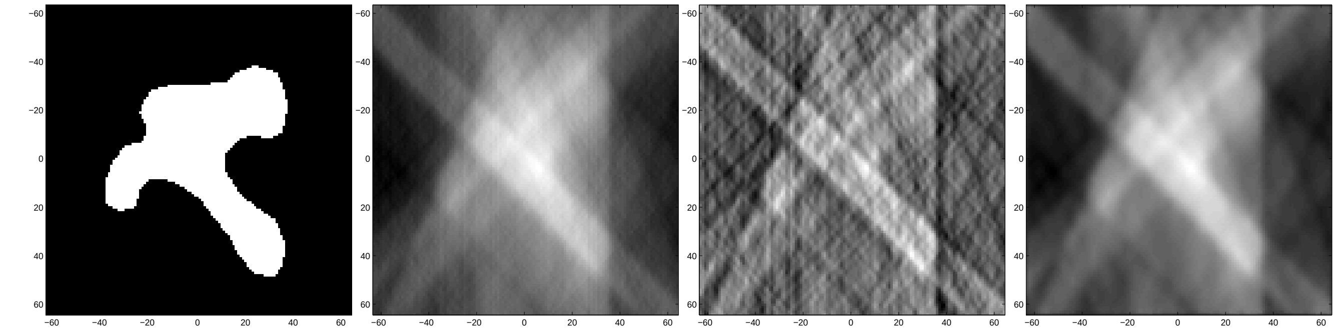


Original — Backprojection — Filtered Baclkprojection — Regularization

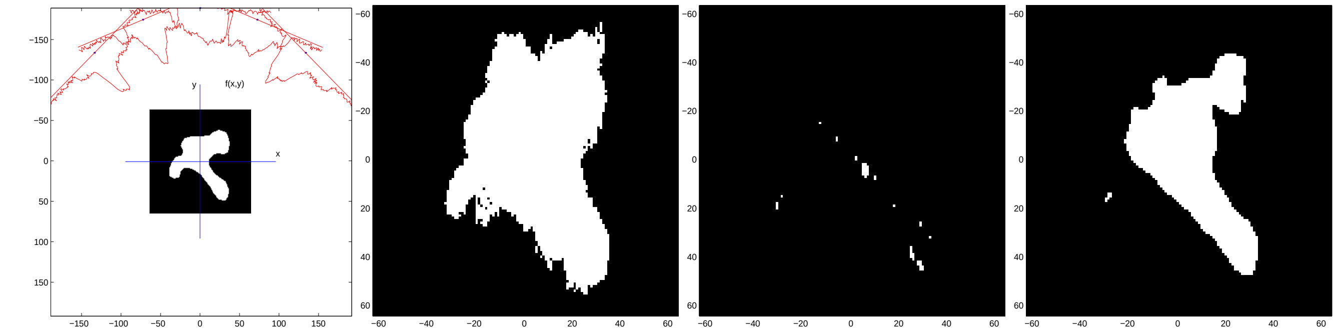
Data — Binary Backprojection — Binary Filtered Baclkprojection — Binary Regularization

Charles Soussen and Ali Mohammad-Djafari

Simulated data: Few and limited angle data 2D Case



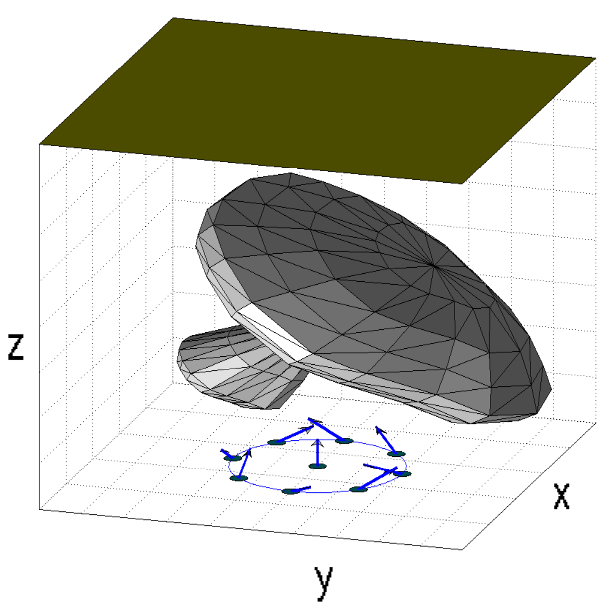
Original | Backprojection | Filtered Baclkprojection | Regularization

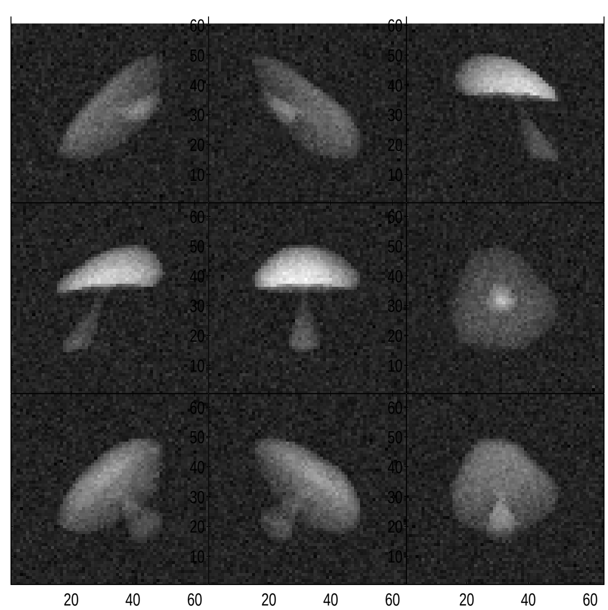


Data | Binary Backprojection | Binary Filtered Baclkprojection | Binary Regularization

Simulated data

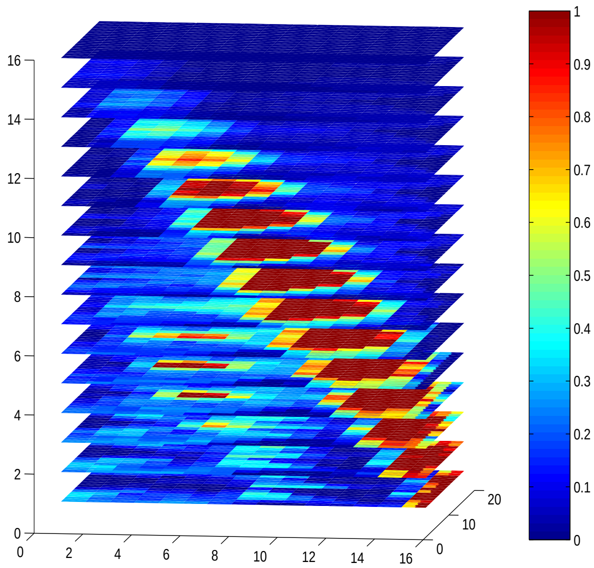
- 9 radiographies of 64×64 pixels.
- SNR = 10 dB.



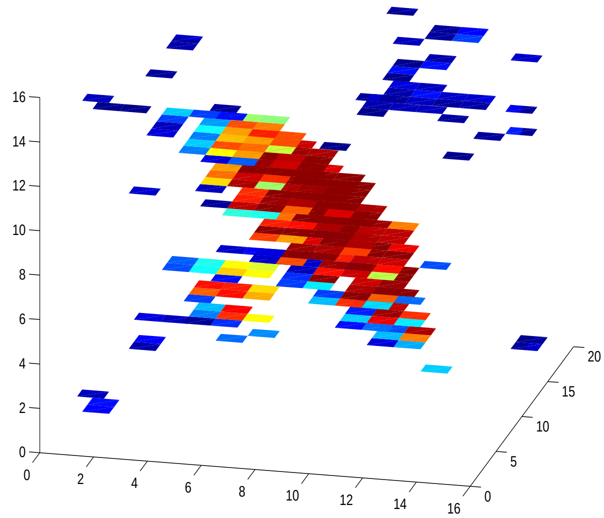


Voxel modeling Reconstructions

- 3 resolution levels $\Longrightarrow$ Object of 64^3 voxels.
- 15 iterations in each resolution.

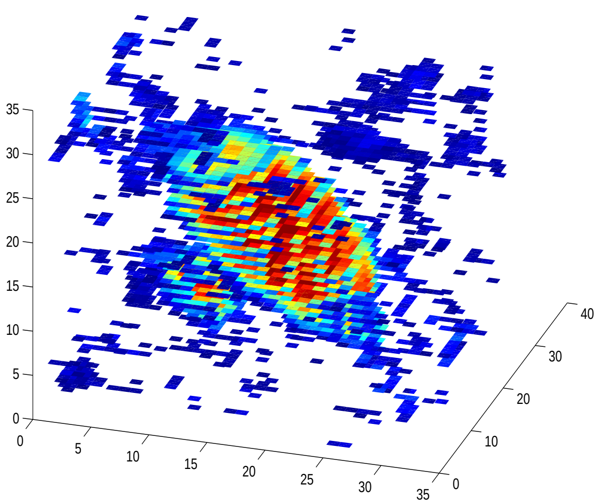


Initialization (16^3 voxels)

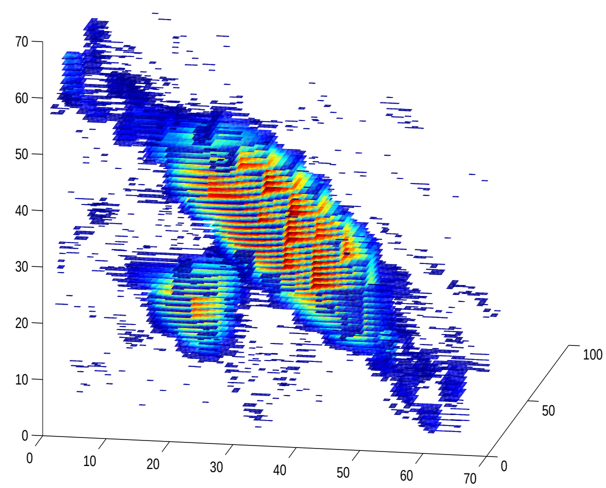


First reconstruction (172 sec. CPU).

Intermediate and final reconstructions



32^3 voxels, 122 sec. CPU



64^3 voxels, 367 sec. CPU

Contour based modeling and reconstruction

- Non parametric modeling:
  - Active contours or Snakes
  - Level set
- Parametric modeling:
  - Simple geometric shapes: ellipses, polygons, ellipsoides and polyedra
  - Affine transformation of simple geometric shapes
  - Harmonic modeling of contours (global basis functions)
  - Spline based modeling (local basis function)
  - Wavelet based modeling (Hybrid global and local basis functions)

Non parametric modeling: Active contours or Snakes

- Used mainly in Image segmentation {Kass88, Malladi95, Caselles97, Cohen93}
- $\mathcal{C}$ is modeled as a function $u:(0,1)\rightarrow\mathbb{R}^2$ minimizing the energy

$$\mathcal{J}(u)=\mathcal{K}(u)+\mathcal{R}_1(u)+\mathcal{R}_2(u)$$

  - $\mathcal{K}(u)$ is an external energy (Data acting as an external force)
  - $\mathcal{R}_1(u)$ and $\mathcal{R}_2(u)$ two internal energies:

    $\mathcal{R}_1(u)=\int_0^1 \|u'(t)\|^2\,dt$ is related to local length of the contour (elasticity)

    $\mathcal{R}_2(u)=\int_0^1 \|u''(t)\|^2\,dt$ is related to local curvature of the contour (rigidity)
- $u(t)$ is updated via a gradient descent leading to a PDE eq. such as

$$v(t)\;=\;\lambda_1\,u''(t)-\lambda_2\,u^{(4)}(t)+\boldsymbol{H_I}\,\nabla\boldsymbol{I}(u(t))/\|\nabla\boldsymbol{I}(u(t))\|\quad=\quad 0$$

where $\boldsymbol{H_I}$ is the Hessian and $\nabla\boldsymbol{I}$ the gradient of the image.

Non parametric modeling: Level set

- Used mainly in Image segmentation but also in Inverse problems
  {Osher88, Malladi95, Santosa96, Litman97b, Whitaker98}
- $\mathcal{C}$ is modeled as the Level-Set of a function $\Phi(x,y): \mathbb{R}^2 \longrightarrow \mathbb{R}$

$$\mathcal{C} = \left\{M(x,y) \in \mathbb{R}^2,\ \Phi(x,y) = 0\right\},$$

- Time evolution $\Phi(x,y,T)$:

$$\partial\Phi(x,y,T)/\partial T + F\,\|\nabla\Phi(x,y,T)\| = 0$$

  - F is the speed of contour evolution which depends on the curvature of $\mathcal{C}_T$
  - Needs updates of $\Phi(x,y,T)$ then $\mathcal{C}_T$
- Updates of $\Phi(x,y,T)$ in 2D or $\Phi(x,y,z,T)$ in 3D needs pixel or voxel modeling.

Parametric modeling: ellipses, ellipsoides

- Ellipses:

$$\sum_{i=1}^{2} \frac{\left[\xi_i - \xi_i(G)\right]^2}{a_i^2} = 1 \longrightarrow \boldsymbol{x} = [\boldsymbol{G}, \theta, a_1, a_2]$$

- Ellipsoides:

$$\sum_{i=1}^{3} \frac{\left[\xi_i - \xi_i(G)\right]^2}{a_i^2} = 1 \longrightarrow \boldsymbol{x} = [\boldsymbol{G}, \theta, \phi, a_1, a_2, a_3]$$

- Parameter estimation:
  - Least squares (LS): $\widehat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \left\{\|\boldsymbol{p} - \boldsymbol{p}(\boldsymbol{x})\|^2\right\}$
  - Moments method
    - Estimate moments of object from moments of projections
    - Determine the geometrical parameters from these moments

Parametric modeling: superquadrics

❏ Super-ellipsoides:

$$\left\{\left[\frac{\xi_1-\xi_1(G)}{a_1}\right]^{2/\beta}+\left[\frac{\xi_2-\xi_2(G)}{a_2}\right]^{2/\beta}\right\}^{\beta/\alpha}+\left[\frac{\xi_3-\xi_3(G)}{a_3}\right]^{2/\alpha}=1$$

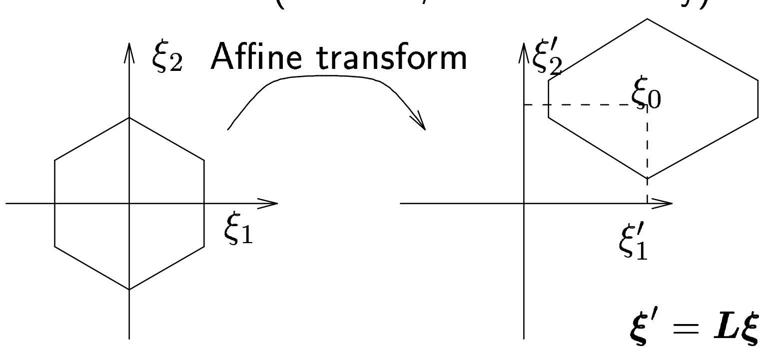
$$\boldsymbol{x}=[\boldsymbol{G},\theta,\phi,a_1,a_2,a_3,\alpha,\beta]$$

❏ Parameter estimation:

- Least squares (LS): $\widehat{\boldsymbol{x}}=\arg\min_{\boldsymbol{x}}\left\{\|\boldsymbol{p}-\boldsymbol{p}(\boldsymbol{x})\|^2\right\}$
- Needs polygonal approximation to compute projections
- Shapes too poor even if non convex shapes

Parametric modeling: Affine transformation of simple geometric shapes

❏ Polygon Affine transformation (Milanfar, Karl and Wilsky)



$$\boldsymbol{\xi}'=\boldsymbol{L}\boldsymbol{\xi}+\boldsymbol{\xi}_0$$

❏ Estimate $\boldsymbol{L}$ and $\boldsymbol{\xi}$ from the data by identifying

- $\boldsymbol{\xi}$ to center of gravity
- $\boldsymbol{L}\boldsymbol{L}^t$ to Inertia matrix

and using the **Moments method** to estimate them from the projections.

❏ Too poor shapes, arbitrary choice of number of vertices

Parametric modeling: Harmonic modeling

❏ Harmonic basis modeling: $\mathcal{C} = \begin{cases} (x(s), y(s)) & \text{or} \quad \rho(\theta) \\ (x(s), y(s), z(s)) & \text{or} \quad \rho(\theta, \phi) \end{cases}$

2D Example:

$$(x(s), y(s)) = \sum_{k=0}^{K} \left(a_k \cos(2\pi ks/S + \alpha_k),\ b_k \cos(2\pi ks/S + \beta_k)\right)$$

or

$$\rho(\theta) = \sum_{k=0}^{K} \rho_k \cos(k\theta) + \sum_{k=0}^{K} \rho_k \sin(k\theta)$$

❏ Estimate the parameters a_k, b_k, α_k, β_k, or ρ_k by LS.

❏ No analytical relation between the parameters and the data

- Polygonal approximation of the contour
- Results depend sensitively on K and on the number of approximated polygon vertices

Parametric modeling: Spline basis modeling

❏ Spline basis modeling [GARNERO,CHALMOND]

- 2D Example: Polygonal modeling

  $\{z_k = (x_k, y_k)\}$ vertices of the polygon.
- 3D Example: Polyedral modeling :

  $\{\text{vertices } [v_k = (x_k, y_k, z_k)] + \text{ faces}\}$

❏ Estimation of $\boldsymbol{z} = \{z_k\}$ or $\boldsymbol{v} = \{v_k\}$ with fixed faces

- Criterion: $\mathcal{J}(\boldsymbol{z}) = \|\boldsymbol{p} - \boldsymbol{p}(\boldsymbol{z})\|^2 + \lambda \sum_k \left\|z_k - \frac{1}{2}(z_{k-1} + z_{k+1})\right\|^2$
- Coordinate by coordinate updating takes advantage of local property of splines

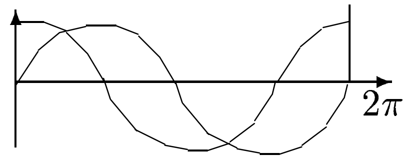
Parametric modeling: wavelet based modeling

❏ Combination of global Fourier basis and local spline basis modeling

- Fourier basis

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \sum_{k=0}^{K} \boldsymbol{A}_k \begin{bmatrix} \cos(2\pi ks/S + \alpha_k) \\ \sin(2\pi ks/S + \alpha_k) \end{bmatrix}$$

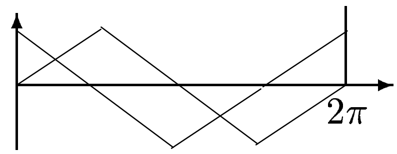
  - Smooth star shaped contours
  - Computation of projections needs polygonal approximation



- Haar basis

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \sum_{k=0}^{K} \boldsymbol{A}_k \begin{bmatrix} h(2\pi ks/S + \alpha_k) \\ h_{\perp}(2\pi ks/S + \alpha_k) \end{bmatrix}$$

  - Directly polygona contours
  - Exact computation of projections



Polygonal and polyedral modeling

❏ Motivations:

- Propose a practical X-ray tomography reconstruction method with reasonable cost of computation
- Computation of projections: intersection of a line–line or line–triangular surface
- Direct Computer Vision presentation
- Easy geometrical parameter computation for NDE decion makers

❏ See the second part of this presentation by Charles Soussen