Deformable Model with Adaptive Mesh and Automated Topology Changes

Jacques-Olivier Lachaud
Benjamin Taton

Laboratoire Bordelais de Recherche en Informatique (LaBRI)
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Motivations</td>
</tr>
<tr>
<td>2.</td>
<td>Description of the deformable model</td>
</tr>
<tr>
<td>2.1</td>
<td>Resolution adaptation by changing metrics</td>
</tr>
<tr>
<td>2.2</td>
<td>Topology adaptation</td>
</tr>
<tr>
<td>2.3</td>
<td>Dynamics</td>
</tr>
<tr>
<td>3.</td>
<td>Defining metrics with respect to images</td>
</tr>
<tr>
<td>3.1</td>
<td>Required properties</td>
</tr>
<tr>
<td>3.2</td>
<td>Building metrics from images</td>
</tr>
<tr>
<td>4.</td>
<td>Results</td>
</tr>
<tr>
<td>5.</td>
<td>Conclusion and perspectives</td>
</tr>
</tbody>
</table>
Motivations

Segmentation/Reconstruction of large 3D images.

- steady technical improvements of acquisition devices,
- increase of image resolution and hence of image size.
Motivations

Segmentation/Reconstruction of large 3D images.

Deformable templates, superquadrics, Fourier snakes...

- reduced set of shape parameters $\Rightarrow$ robust and efficient,
- lack of genericity: new problem $\Rightarrow$ new model.
**Motivations**

<table>
<thead>
<tr>
<th>Segmentation/Reconstruction of large 3D images.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deformable templates, superquadrics, Fourier snakes…</td>
</tr>
<tr>
<td>Fully generic models (T-Snakes, Simplex meshes, Level-sets…)</td>
</tr>
</tbody>
</table>

- very wide range of shapes,
- number of shape parameters directly determined by image resolution ⇒ heavy computational costs.
Motivations

Segmentation/Reconstruction of large 3D images.

<table>
<thead>
<tr>
<th>Deformable templates, superquadrics, Fourier snakes...</th>
<th>not generic enough</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully generic models (T-Snakes, Simplex meshes, Level-sets...)</td>
<td>computationally expensive</td>
</tr>
</tbody>
</table>

Objective

- To build a deformable model
  - that can recover objects with any topology,
  - with costs more independent from the size of input data.
Model Description

Explicit model

- Closed triangulated surface,
- Dynamics of a mass-spring system that undergoes
  - image forces,
  - regularizing internal forces,
  - any other additional force...
Explicit model

- Closed triangulated surface,
- Dynamics of a mass-spring system that undergoes
  - image forces,
  - regularizing internal forces,
  - any other additional force...

Regular sampling of the model mesh
Explicit model

- Closed triangulated surface,
- Dynamics of a mass-spring system that undergoes
  - image forces,
  - regularizing internal forces,
  - any other additional force...

Regular sampling of the model mesh

Transformed into adaptive sampling by changing metrics
Model Description

Explicit model

- Closed triangulated surface,
- Dynamics of a mass-spring system that undergoes
  - image forces,
  - regularizing internal forces,
  - any other additional force...

Regular sampling of the model mesh

Transformed into adaptive sampling by changing metrics

Automated topology changes
Regular sampling using distance constraints

\[ \delta \leq d_E(u, v) \leq \zeta \delta \]

Where

- \( u, v \) are neighbour vertices,
- \( d_E \) denotes the Euclidean distance,
- \( \delta \) determines the global resolution of the model,
- \( \zeta \) is the ratio between the lengths of the longest and smallest edge on the mesh.
Regular Sampling

Regular sampling using distance constraints

$$\delta \leq d_E(u, v) \leq \zeta \delta$$

Restoring constraints

Edge too short: contraction
(+ special case...)

Edge too long: split
Resolution Adaptation

Euclidean distance replaced by a Riemannian distance

\[ \delta \leq d_R(u, v) \leq \zeta \delta \]
Resolution Adaptation

Euclidean distance replaced by a Riemannian distance

\[ \delta \leq d_R(u, v) \leq \zeta \delta \]

If \( d_R \) underestimates distances

edge lengths fall under the \( \delta \) threshold

\( \Rightarrow \) edges contract and vertex density decreases.
Resolution Adaptation

Euclidean distance replaced by a Riemannian distance

\[ \delta \leq d_R(u, v) \leq \zeta \delta \]

If \( d_R \) underestimates distances

- edge lengths fall under the \( \delta \) threshold
  \[ \Rightarrow \text{edges contract and vertex density decreases.} \]

If \( d_R \) overestimates distances

- edge lengths exceed the \( \zeta \delta \) threshold
  \[ \Rightarrow \text{edges split and vertex density increases.} \]
Resolution Adaptation

Euclidean distance replaced by a Riemannian distance

\[ \delta \leq d_R(u, v) \leq \zeta \delta \]

The new distance \( d_R \) should

- overestimate distances in interesting parts of the image to increase accuracy,
- underestimate distances elsewhere to decrease accuracy.
Euclidean length of an elementary displacement $\vec{d}s$

\[ L_E(\vec{d}s) = \sqrt{\vec{d}s \times t \vec{d}s} \]
Riemannian length of an elementary displacement $\vec{d}s$

$$L_R(\vec{d}s) = \sqrt{\vec{d}s \times G(x_1, \ldots x_n) \times t\vec{d}s}$$

Where $G$ is a Riemannian metric, i.e.

- $G(x_1, \ldots x_n)$ is a dot product,
- $G$ is continuous.

Which means that $L_R(\vec{d}s)$ depends on both

- the displacement $\vec{d}s$,
- the origin $(x_1, \ldots x_n)$ of the displacement.
Riemannian length of an elementary displacement $\vec{d}s$

$$L_R(\vec{d}s) = \sqrt{\vec{d}s \times G(x_1, \ldots, x_n) \times t \vec{d}s}$$

Length of a path $\gamma$

$$L_R(\gamma) = \int_0^1 \sqrt{\vec{\gamma}(t) \times G(\gamma(t)) \times t \vec{\gamma}(t)} \, dt$$

Length of a path $\gamma$ $\sim$ Sum of the lengths of the elementary displacements it is composed of.
Riemannian Metrics

Riemannian length of an elementary displacement $\tilde{d}s$

$$L_R(\tilde{d}s) = \sqrt{\tilde{d}s \times G(x_1, \ldots, x_n) \times ^t\tilde{d}s}$$

Length of a path $\gamma$

$$L_R(\gamma) = \int_0^1 \sqrt{\tilde{\gamma}(t) \times G(\gamma(t)) \times ^t\tilde{\gamma}(t)} \, dt$$

Distance between two points $u$ and $v$

$$d_R(u, v) = \inf \{ L_R(\gamma) \mid \gamma(0) = u, \gamma(1) = v \}$$
Geometrical Interpretation of Metrics

Euclidean unit ball

Local Riemannian unit ball

\((\vec{v}_1, \mu_1), (\vec{v}_2, \mu_2)\) local eigen decomposition of the metric.
Geometrical Interpretation of Metrics

Euclidean unit ball

Local Riemannian unit ball

Changing the Euclidean metric with a Riemannian metric $G$

Locally expanding/contracting the space
Changing the Euclidean metric with a Riemannian metric $G$

Locally expanding/contracting the space along $\vec{v}_1$ with the ratio $\frac{1}{\sqrt{\mu_1}}$, 

\[
\frac{1}{\sqrt{\mu_2}}
\]
Changing the Euclidean metric with a Riemannian metric $G$

Locally expanding/contracting the space along $\vec{v}_1$ with the ratio $\frac{1}{\sqrt{\mu_1}}$, along $\vec{v}_2$ with the ratio $\frac{1}{\sqrt{\mu_2}}$, ...
Topology Changes

Two stages:

- detect self collisions of the model,
- perform appropriate local reconfigurations of the mesh.
Two stages:

- detect self collisions of the model,
- perform appropriate local reconfigurations of the mesh.
Two stages:

- detect self collisions of the model,
- perform appropriate local reconfigurations of the mesh.
Two stages:

- detect self collisions of the model,
- perform appropriate local reconfigurations of the mesh.
Topology Changes

Detection of self collisions

Vertex \( u \) crosses over the \((a, b, c)\) face \(\implies\) Vertex \( u \) enters one of the \(S_a, S_b, \text{ or } S_c\) spheres.
Detection of self collisions

- Collision if $u$ and $w$ are not neighbours and $d_I(u, w) \leq \lambda_I \zeta \delta$. 
Detection of self collisions

Collision if \( u \) and \( w \) are not neighbours and \( d_E(u, w) \leq \lambda_E \zeta \delta \).

Local reconfigurations

Collision between two parts of the mesh

Special case of edge contraction
Topology Changes

Detection of self collisions

- Collision if \( u \) and \( w \) are not neighbours and \( d_E(u, w) \leq \lambda_E \zeta \delta \).

Local reconfigurations

- Collision between two parts of the mesh
- Special case of edge contraction

If the metric is changed

- \( \lambda_E \) replaced with a new constant \( \lambda_R \)
Motion equations with a Euclidean Metric

$$\forall k \in \{1, \ldots n\}, \quad m\ddot{x}_k = F_k$$
Dynamics

Motion equations with a Riemannian metric

\[ \forall k \in \{1, \ldots n\}, \quad m\ddot{x}_k = F_k - \sum_{i,j=1}^{n} \Gamma^k_{ij} \dot{x}_i \dot{x}_j \]

Addition of a **corrective term** that takes account of the metric

- \[ \Gamma^k_{ij} = \frac{1}{2} \sum_{l=1}^{n} g^{kl} \left( \frac{\partial g_{il}}{\partial x_j} + \frac{\partial g_{lj}}{\partial x_i} - \frac{\partial g_{ij}}{\partial x_l} \right) \] (Christoffel’s symbols),

- \( g_{ij} \) are the coefficients of the \( G(x_1, \ldots x_n) \),

- \( g^{kl} \) are the coefficients of \( G^{-1}(x_1, \ldots x_n) \),
Motion equations with a Riemannian metric

\[ \forall k \in \{1, \ldots n\}, \quad m\ddot{x}_k = F_k - \sum_{i,j=1}^{n} \Gamma^k_{ij} \dot{x}_i \dot{x}_j \]

Addition of a corrective term that takes account of the metric

- \[ \Gamma^k_{ij} = \frac{1}{2} \sum_{l=1}^{n} g^{kl} \left( \frac{\partial g_{il}}{\partial x_j} + \frac{\partial g_{lj}}{\partial x_i} - \frac{\partial g_{ij}}{\partial x_l} \right) \] (Christoffel’s symbols),

- \( g_{ij} \) are the coefficients of the \( G(x_1, \ldots x_n) \),

- \( g^{kl} \) are the coefficients of \( G^{-1}(x_1, \ldots x_n) \),

(corrective term neglected: second order in \( \dot{x} \) + no influence on the rest position)
### Summary

<table>
<thead>
<tr>
<th>Distance estimation</th>
<th>Euclidean Metric</th>
<th>Riemannian Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sqrt{\overrightarrow{uv} \times t\overrightarrow{uv}}$</td>
<td>$\inf_{\gamma} L_R(\gamma)$</td>
<td></td>
</tr>
<tr>
<td>Regular sampling</td>
<td>$\delta \leq d_E(u, v) \leq \zeta \delta$</td>
<td>$\delta \leq d_R(u, v) \leq \zeta \delta$</td>
</tr>
<tr>
<td>uniform resolution</td>
<td>adaptive resolution</td>
<td></td>
</tr>
<tr>
<td>Collision detection</td>
<td>$d_E(u, v) \leq \lambda_E \zeta \delta$</td>
<td>$d_R(u, v) \leq \lambda_R \zeta \delta$</td>
</tr>
<tr>
<td>Local reconfigurations</td>
<td>unchanged</td>
<td></td>
</tr>
<tr>
<td>Motion equations</td>
<td>$m\ddot{x} = F$</td>
<td>$m\ddot{x}<em>k = F_k - \sum</em>{i,j=1}^{n} \Gamma_{ij}^k \dot{x}_i \dot{x}_j$</td>
</tr>
</tbody>
</table>

Deformable Model with Adaptive Mesh and Automated Topology Changes -- p.11/21
Required Properties

Four kinds of situations

- no contour
- plane contour
- edge, tubular structure
- corner

Choice of the metric

- Eigenvectors should correspond to the normal and the principal directions of the contour,
- Eigenvalues should correspond to the strength and the principal curvatures of the contour.
### Required Properties

<table>
<thead>
<tr>
<th>Structure in the image</th>
<th>Expected resolution</th>
<th>Eigen structure of the metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>No contour</td>
<td>◇ low in all directions</td>
<td>$0 \preceq \mu_2 \preceq \mu_1 \preceq \mu_0$</td>
</tr>
</tbody>
</table>
| Flat contour           | ◇ low in the direction of the contour  
                         ◇ high in the orthogonal direction | $0 \preceq \mu_2 \preceq \mu_1 \ll \mu_0$ |
| Tubular structure      | ◇ low in the direction of the structure  
                         ◇ high in both orthogonal directions | $0 \preceq \mu_2 \ll \mu_1 \preceq \mu_0$ |
| Corner                 | ◇ high along all directions          | $0 \ll \mu_2 \preceq \mu_1 \preceq \mu_0$ |
**Structure Tensor (1/2)**

**Definition**

The matrix $J_{\rho,\sigma}$ that represents the mapping:

$$\vec{u} \rightarrow g_\rho \ast \left( \vec{\nabla} ( g_\sigma \ast I ) \cdot \vec{u} \right)^2$$

Which results in:

$$J_{\rho,\sigma} = g_\rho \ast \begin{pmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{pmatrix}$$
**Structure Tensor (1/2)**

**Definition**

The matrix $J_{\rho,\sigma}$ that represents the mapping:

$$\vec{u} \rightarrow \left( g_{\sigma} \ast I \right) \cdot \vec{u}^2$$

**Interpretation**

- smoothes the input image,
Structure Tensor (1/2)

Definition

The matrix $J_{\rho,\sigma}$ that represents the mapping:

\[ \vec{u} \rightarrow \nabla \left( (g_\sigma * I) \cdot \vec{u} \right)^2 \]

Interpretation

- smooths the input image,
- characterizes direction and orientation of image gradient,
Structure Tensor (1/2)

Definition

The matrix $J_{\rho,\sigma}$ that represents the mapping:

$$\vec{v} \rightarrow \left( \nabla \left( g_\sigma * I \right) \cdot \vec{v} \right)^2$$

Interpretation

- smoothes the input image,
- characterizes direction and orientation of image gradient,
- removes the orientation information,
**Structure Tensor (1/2)**

### Definition

The matrix $J_{\rho,\sigma}$ that represents the mapping:

$$\vec{v} \rightarrow g_\rho \* \left( \vec{\nabla} (g_\sigma \* I) \cdot \vec{v} \right)^2$$

### Interpretation

- smoothes the input image,
- characterizes direction and orientation of image gradient,
- removes the orientation information,
- integrates the direction information over a neighbourhood.
Properties of the eigenstructure of the structure tensor

In the neighbourhood of a contour

- $\vec{v}_1$ orthogonal to image contours, $\xi_1$ contour strength,
- $\vec{v}_2, \vec{v}_3$ principal directions of the contour, $\xi_2$ and $\xi_3$ qualitatively equivalent to principal curvatures.

<table>
<thead>
<tr>
<th>Condition</th>
<th>$\xi_1$</th>
<th>$\xi_2$</th>
<th>$\xi_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>no contour</td>
<td>$0 \approx \xi_3 \approx \xi_2 \approx \xi_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>flat contour</td>
<td>$0 \approx \xi_3 \approx \xi_2 \ll \xi_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sharp edge</td>
<td>$0 \approx \xi_3 \ll \xi_2 \approx \xi_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>corner</td>
<td>$0 \ll \xi_3 \approx \xi_2 \approx \xi_1$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Results (1/3)

Computer generated image

Object represented in the image

Slices extracted from the image
Eigen decomposition of the structure tensor

- Object represented in the image
- Isosurfaces of the second and third eigen values of the structure tensor of the image.
Results (1/3)

Segmentation/Reconstruction results

2163 vertices, $\delta = 2, \zeta = 2.5$

3497 vertices, $\delta = 3, \zeta = 2.5$

8904 vertices, $\delta = 1, \zeta = 2.5$

$1 \leq \sqrt{\mu_2} \leq \sqrt{\mu_1} \leq \sqrt{\mu_0} \leq 10$
Results (1/3)

One corner of the cube

Hole in the cube
Repartition of edge lengths

- fine model, • adaptive model, ● coarse model
Results (2/3)

Computer generated image

Object represented in the image

Slices extracted from the image
Eigen decomposition of the structure tensor

Object represented in the image

Isosurfaces of the second and third eigen values of the structure tensor of the image.
Segmentation/Reconstruction results

- **2107 vertices**
  \[ \delta = 2, \zeta = 2.5 \]

- **4832 vertices**
  \[ \delta = 3, \zeta = 2.5, \quad 1 \leq \sqrt{\mu_2} \leq \sqrt{\mu_1} \leq \sqrt{\mu_0} \leq 10 \]

- **8542 vertices**
  \[ \delta = 1, \zeta = 2.5 \]
Repartition of edge lengths

- fine model,
- adaptive model,
- coarse model
Results (3/3)

Biomedical image (Head CT-scan)
Eigen decomposition of the structure tensor

An isosurface of the 2nd eigenvalue.
An isosurface of the 3rd eigenvalue.
Object in the image.
Segmentation/Reconstruction results

10.970 vertices
$\delta = 0.012, \zeta = 2.5$

23.142 vertices
$\delta = 0.024, \zeta = 2.5,$
$1 \leq \sqrt{\mu_2} \leq \sqrt{\mu_1} \leq \sqrt{\mu_0} \leq 15$

46.590 vertices,
$\delta = 0.005, \zeta = 2.5$
Left ear

Orbit of the left eye viewed from behind left
Repartition of the edge length

- fine model,
- adaptive model,
- coarse model
Conclusion

- Deformable model that achieves both
  - adaptive topology,
  - adaptive resolution

Perspectives

- Initialization with adaptive resolution,
- Different ways of building metrics...