Dynamic shape detection and analysis of deformable structures in biomedical imaging

Ph.D. Candidate: Alberto Silletti

Ph.D. Advisor: Prof. Angelo Cenedese

School Coordinator: Prof. Matteo Bertocco

Alberto Silletti - 13th April 2010
Outline

• Introduction: what is a shape?
• The shape detection problem
  • generalized active contours
  • reticular shape detection
• Dynamic shape detection
• Analysis
• Synthesis
Introduction: What is a shape?

A fuzzy concept, many definitions occurs over scientific and common literature:

- The spatial arrangement of something as distinct from its substance; “geometry is the mathematical science of shape” (wordReference)

- A shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object (George Kendall, mathematician)

X \xrightarrow{sensors} I \xrightarrow{intuition} \Phi(X) \xrightarrow{representation} \mathcal{R}(\Phi)

Level sets  
Landmark  
Polygons  
Voxels or pixels  
Spline  
Deformable model
The shape detection problem

Given an image $I$ (data), and choosing a shape representation model $\mathcal{R}(\Phi)$, the shape detection problem consists of:

$$\mathcal{R}(\Phi) = \arg \min_{\mathcal{R}(\Phi) \in \mathcal{S}} \mathcal{E}(I, \Phi)$$

where $\mathcal{E}$ is a suitable energy function and $\mathcal{S}$ is the space of the possible representations.

Shape detection involves:
- the design of $\mathcal{E}$
- the procedure to minimize over $\mathcal{S}$
Generalized Active Contours 1/4

An Active Contour is a curve $C(s) \in \mathbb{R}^d$, $s \in [0,1]^d$
evolving in pseudo-time according to an associated energy:

$$\mathcal{E}(C) = S(C) + \mathcal{P}(C)$$

$$S(C) = \int_{\partial C} \alpha(s) \left| \frac{\partial C}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 C}{\partial s^2} \right|^2 \, ds$$

$$\mathcal{P}(C) = \int_{\partial C} \mathcal{Q} \left[ I \right] \, ds$$

Generalized Active Contours extends the framework:

$$\mathcal{E}_g(C) = S_g(C) + \mathcal{P}_g(C)$$

$$\int_{\partial C} \alpha(s) \left| \frac{\partial C}{\partial s} \right|^2 + \beta(s) \left| \frac{\partial^2 C}{\partial s^2} \right|^2 \, ds + \sum_{i=1}^{n_i} \gamma_i \mathcal{F}_i$$

$$\int_{\partial C} \mathcal{Q} \left[ I \right] \, ds + \sum_{j=1}^{n_g} \lambda_j \mathcal{G}_{ij}$$

where $\mathcal{F}_i$ and $\mathcal{G}_{ij}$ are suitable energy terms, each of those addressing a peculiar aspect of the shape (such as color, texture or central moments).
Generalized Active Contours build $\mathcal{F}_i$ and $\mathcal{G}_j$ energies as probability density functions. Let’s pretend you want to capture the dark protein appendixes:

1. Choose a metric to characterize them, such as color at boundary:
   \[
   \tau(C) = \int_{\partial C} I(s)ds
   \]

2. Infer the probability density $\Gamma_\tau$ - analytically, empirically or by training -

3. Build an energy term as:
   \[
   \mathcal{E}_\tau = -\Gamma_\tau(\tau_C)
   \]
Generalized Active Contours 3/4

The work provides a Toolbox with several build-in energy terms

Alberto Silletti - 13th April 2010
Assuming all the density function are Gaussian, we rewrite the energy terms as:

\[
\Gamma = \mathcal{N}(\bar{\mu}, \bar{\Sigma}) = \mathcal{N}
\begin{pmatrix}
\mu_1 \\
\mu_2 \\
\vdots \\
\mu_i \\
\vdots \\
\mu_{n_{y+n_{g}}}
\end{pmatrix}
\begin{pmatrix}
\Sigma_{11} & \Sigma_{12} & \Sigma_{13} & \cdots \\
\Sigma_{21} & \Sigma_{22} & \Sigma_{23} & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{i1} & \Sigma_{i2} & \Sigma_{i3} & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
\Sigma_{n_{y+n_{g}}+1, n_{y+n_{g}}} & \cdots & \cdots & \cdots & \Sigma_{n_{y+n_{g}}+n_{y+n_{g}}}
\end{pmatrix}
\]

We can assess the quality of our Snaking algorithm looking at the eigenvectors: if they are close to zero, then we have a robust algorithm.

We can also simplify the matrix over certain thresholds

\[
\begin{pmatrix}
\sigma_1 & 0 & 0 & \cdots \\
0 & \sigma_2 & 0 & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sigma_i \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \cdots & \sigma_k
\end{pmatrix}
\]

\[
T^T \cdot \begin{pmatrix}
\sigma_1 & 0 & 0 & \cdots \\
0 & \sigma_2 & 0 & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \sigma_i \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \cdots & \sigma_k
\end{pmatrix}
\]
Reticular shape detection

-The random walk agents approach-

Idea: many Random Walk Agents $A_i$ flood the frame, each of them finding a path. Each agent has a position $p$ and an energy $E$.

Agents locally minimize an energy term $\hat{E}$, a local approximation of $E$. $E$ is unknown and difficult to design, whereas $\hat{E}$ is

$$\hat{E}(\theta_i) = \frac{\int_{\Omega_i} \sqrt{(I(\omega) - 1)^2} d\omega}{\int_{\Omega} d\omega} \quad p(t+1) = p(t) + k \cdot g(E^\theta)$$
Reticular shape detection
-The random walk agents approach-

**Code snippet 4 - Random Walk Agents main loop**

```
1 PQ ← initPriorityQueue() // create a priority queue
2 G ← initGraph() // create an empty graph
3 while PQ not empty() {
4     A ← dequeue(PQ) // extract the best agent
5     valid, border ← validateAgent(A, I, G)
6     if valid
7         G ← add2Graph(G, A) // add to the graph
8         if (valid and not(border)) {
9             E ← computeEnergyFunction(I, A)
10            D ← pickDirections(E)
11            for k-1 to |D| {
12                A_k ← moveAgent(A, D_k)
13                PQ ← enqueue(A_k, PQ)
14            }
15         }
```
Reticular shape detection
-The random walk agents approach-

Drosophila wing  Corneal fundus  Retina

Drosophila developmental stage
Dynamic shape detection

Given a sequence of temporal coherent images \( \{I_1, \ldots, I_t, \ldots, I_N\} \), Dynamic shape detection is the problem of detecting the shape in each frame:

\[
\Phi_1 = \arg \min_{\Phi_1 \in S} E(I_1, \Phi_1) \\
\Phi_2 = \arg \min_{\Phi_2 \in S} E(I_2, \Phi_2) \\
\vdots \\
\Phi_N = \arg \min_{\Phi_N \in S} E(I_N, \Phi_N)
\]

Images temporal coherence: \( I_{t+1} = I_t + \Delta_{t+1} \)

Shape temporal coherence: \( \Phi_{t+1} = \Phi_t + \delta_{t+1} \)

The optimum \( \delta_{t+1} \) satisfies:

\[
\Phi_{t+1} = \Phi_t + \delta_{t+1} = \arg \min_{\Phi_t \in S} E(I_{t+1}, \Phi_t)
\]

We compute \( \delta_{t+1} \) using the \textit{J-maps} approach
Dynamic shape detection
-the J-maps approach-

Choose a location \( p_t = [x, y] \in I_t \), a \( J_{p_t}(\Delta x, \Delta y) \) map is scalar function whose values are (inversely) related to the probability of \( p_t \) being translated into \( p_{t+1} = [x + \Delta x, y + \Delta y] \in I_{t+1} \)

\[
J_{p_t}(\Delta x, \Delta y) = \int p \left| N_t(p_t) - N_{t+1}(p_{t+1}) \right| dp
\]

\[
\delta_{t+1}^i = \arg \min_{\Delta x, \Delta y} J_{p_t}
\]

where \( N_t \) is around \( p_t = [x, y] \in I_t \),
and \( N_{t+1} \) is around \( p_{t+1} = [x + \Delta x, y + \Delta y] \in I_{t+1} \)

- The J-maps describe the deformation of \( I_t \) into \( I_{t+1} \)
- The args-min of the J-maps is the Optical Flow Field
- The args-min computed in the location of points of \( \Phi(X) \) is \( \delta_{t+1}^i \)
Dynamic shape detection
-the J-maps approach-

Correction: we impose \( \frac{\partial \delta_{i+1}}{\partial p} \rightarrow 0 \)

The J-maps correction is an iterative steps. For each map \( J_p \) we consider a set of neighbor maps \( \{ J_{p_1}, J_{p_2}, ..., J_{p_n} \} \), and compute the global minima for each of them:
\[ \{(v_{x1}, v_{y1}), (v_{x2}, v_{y2}), ..., (v_{xn}, v_{yn})\} \]

We correct \( J_p \) according to: \( \tilde{J}_p = J_p \cdot (1 - \mathcal{G}([\vec{v}_x, \vec{v}_y], \Sigma_{xy})) \)

where \( \mathcal{G} \) is a Gaussian build on \( \{(v_{x1}, v_{y1}), (v_{x2}, v_{y2}), ..., (v_{xn}, v_{yn})\} \)
Dynamic shape detection

-the J-maps approach-
Shape analysis

Shape analysis is the process of extracting metrics and descriptors from shape or sequence of shapes.

\[ \{ R(\Phi_0), \ldots, R(\Phi_t), \ldots, R(\Phi_N) \} \xrightarrow{\text{analysis}} \mathbb{R}^k \]

The idea is to forget about the Representation Model and use only a reduced set of “numbers” to capture properties of the shape. These values are forces, lengths, labels, descriptors, medical diagnosis, classifications.
Shape synthesis

The synthesis step involves the creation of a model of the structure of interest, mechanical, chemical, statistical...

We build a mechanical model for the Drosophila epithelium.

The idea is to simulate what we see in the microscopes in “silico” (Matlab in our case).

Here we use a “FEM” simulator (work in progress)
-not mentioned in the slides-

Full Body Scanner project

Robust Registration of dermatoscopic images

Spectral methods for point set matching
Questions?