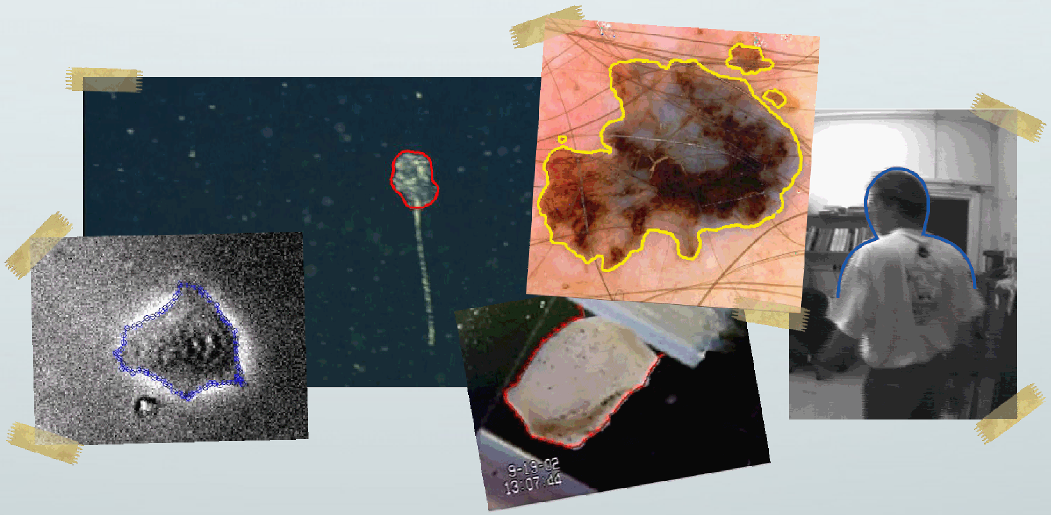
Dynamic shape detection and analysis of deformable structures in biomedical imaging

Dip. di Ing. Dell'Informazione
Dottorato in Scienza e Tecnologia dell'Informazione – XXII Ciclo

Ph.D. Advisor: Prof. Angelo Cenedese

Ph.D. Candidate: Alberto Silletti

School Coordinator: Prof. Matteo Bertocco



Outline

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

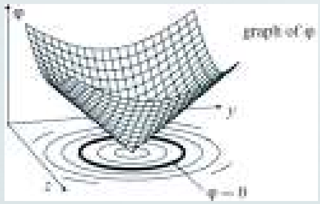
Introduction: What is a shape?

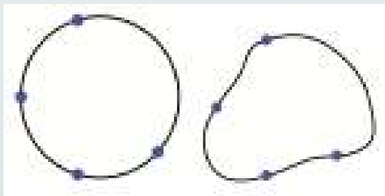
A fuzzy concept, many definitions occurs over scientific and common literature:

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

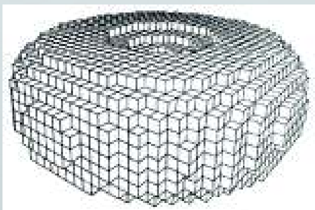
- *A shape is all the geometrical information that remains when location, scale and rotational effects are filtered out from an object* (George Kendall, mathematician)

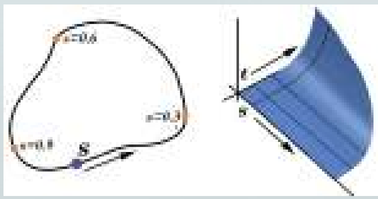
$$X \xrightarrow{sensors} I \xrightarrow{intuition} \Phi(X) \xrightarrow{representation} \mathcal{R}(\Phi)$$

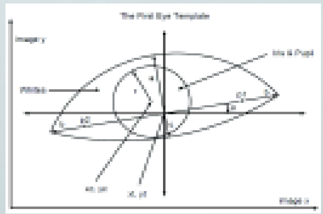

Level sets


Landmark


Polygons


Voxels or pixels


Spline


Deformable model

The shape detection problem

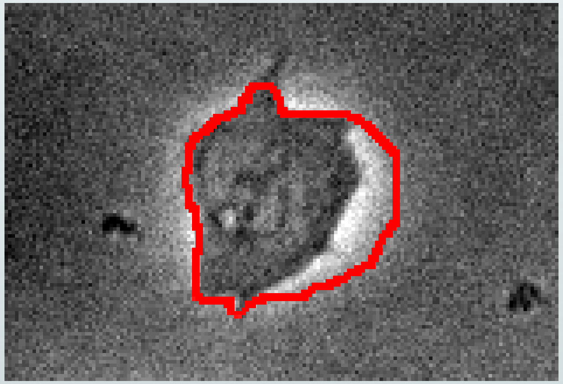
Given an image I (data), and choosen a shape representation model $\mathcal{R}(\Phi)$ the shape detection problem consist of:

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

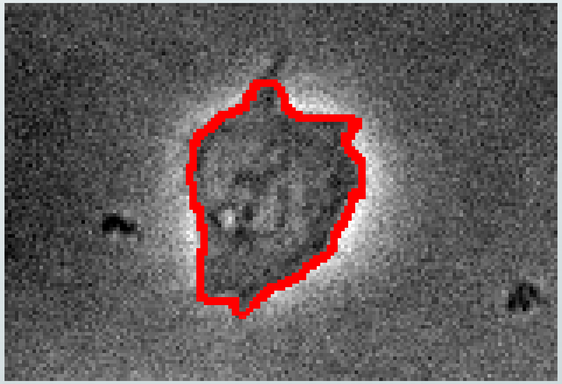
where $\mathcal{E}$ is a suitable energy function and $\mathcal{S}$ is the space of the possible representations.

Shape detection involves:

- *the design of $\mathcal{E}$*
- *the procedure to minimize over $\mathcal{S}$*



non optimal $\mathcal{E}$



optimal (minimum) $\mathcal{E}$

Generalized Active Contours 1/4

An Active Contour is a curve $\mathcal{C}(s) \in \mathbb{R}^d, \quad s \in [0,1]^d$

evolving in pseudo-time according to an associated energy:

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

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

$$\mathcal{P}(\mathcal{C}) = \int_{\partial C} \mathcal{D}[I] \, \mathrm{d}s$$

Generalized Active Contours extends the framework:

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

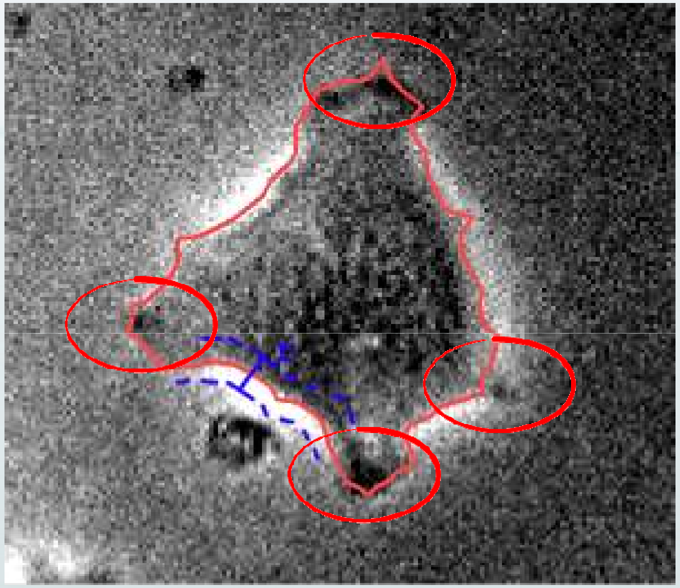
$$\int_{\partial C} \alpha(s) \left|\frac{\partial \mathcal{C}}{\partial s}\right|^2 + \beta(s) \left|\frac{\partial^2 \mathcal{C}}{\partial s^2}\right|^2 ds + \sum_{i=1}^{n_f} \gamma_i \mathcal{F}_i$$

$$\int_{\partial C} \mathcal{D}[I] \, ds + \sum_{j=1}^{n_g} \lambda_j \mathcal{G}_j$$

where $\mathcal{F}_i$ and $\mathcal{G}_j$ are suitable energy terms, each of those addressing a peculiar aspect of the shape (such as color, texture or central moments).

Generalized Active Contours 2/4

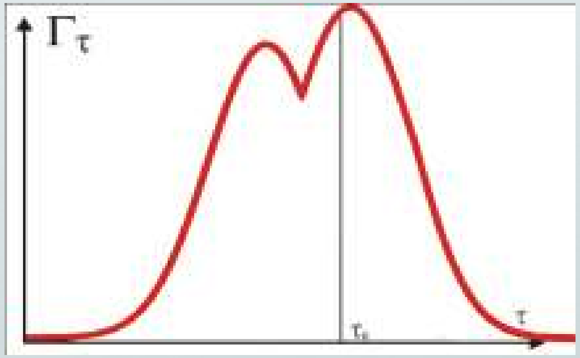
Generalized Active Contours build $\mathcal{F}_i$ and $\mathcal{G}_j$ energies *as probability density functions*.
Let's pretend you want to capture the dark protein appendixes:



1. Choose a metric to characterize them, such as color at boundary:

$$\tau(\mathcal{C}) = \int_{\partial\mathcal{C}\pm r} I(s)ds$$

2. Infer the probability density Γ_τ - analitically, empirically or by training -



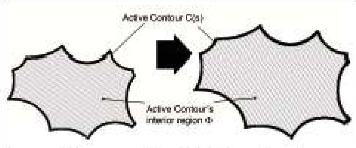
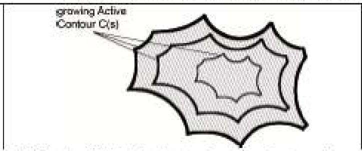
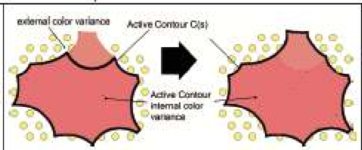
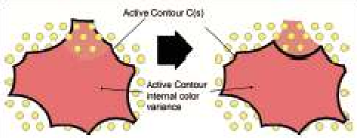
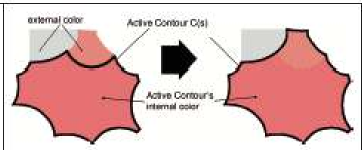
3. Build an energy term as:

$$\mathcal{E}_\tau = -\Gamma_\tau(\tau_c)$$

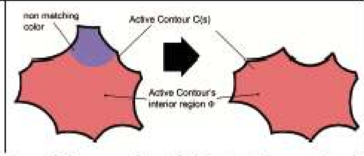
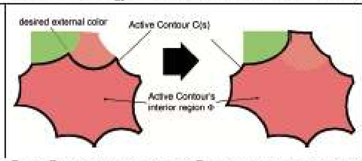
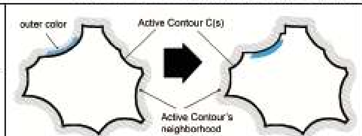
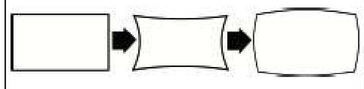
Generalized Active Contours 3/4

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

Energy Toolbox II/II

Name	Figure	Definition
Area()	 Area energy drives the Contour towards an area size c	$E=-\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{-\frac{\mu^2}{2\sigma^2}}$ $\mu=\int_{C^-}\lvert area(C)-c\rvert\,d\omega$ c =desired area size
Enflate() Deflate()	 Enflate (or Deflate) the Active Contours forcing a larger (smaller) area. μ_{max} is an upper bound, possibly unreachable	$E=-\frac{min(\mu,\mu_{max})}{\mathcal{N}}$ $\mu=\int_{C^-}ds$ μ_{max} =max value for μ $\mathcal{N}$ =normalizing constant
Inter Class Color Variance()	 Inter Class Color Variance enhances the color variance between the interior and the exterior	$E=-\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{-\frac{\mu^2}{2\sigma^2}}$ $\mu=\lvert\Sigma(I(C^+))-\Sigma(I(C^-))\rvert$
Internal Minimal (Maximal) Color Variance()	 Internal Minimal (or Maximal) Variance favours a small (large) internal color variance	$E=-\frac{min(\mu,\mu_{max})}{\mathcal{N}}$ $\mu=\Sigma(I(C^-))$ μ_{max} =max value for μ $\mathcal{N}$ =normalizing constant
Inter Class Color()	 Inter Class Color enhances the color difference between the interior and the exterior	$E=-\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{-\frac{\mu^2}{2\sigma^2}}$ $\mu=\lvert\int_{C^-}I(C)d\omega+$ $-\int_{C^+}I(C)d\omega\rvert$

Energy Toolbox I/II

Name	Figure	Definition
Inner Color()	 Inner Color energy drives the Contour to have an internal color c	$E=-\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{-\frac{\mu^2}{2\sigma^2}}$ $\mu=\int_{C^-}\lvert I(\omega)-c\rvert\,d\omega$ c =desired inner color C^- = interior of C
Outer Color()	 Outer Color energy drives the Contour towards an external color c. The areas of c color remain out of the Contour	$E=-\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{-\frac{\mu^2}{2\sigma^2}}$ $\mu=\int_{C^+}\lvert I(\omega)-c\rvert\,d\omega$ c =desired inner color C^+ = exterior of C
Surrounding Inner Color()	 Surrounding Inner Color forces the Contour to have a small internal neighbor of color c	$E=-\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{-\frac{\mu^2}{2\sigma^2}}$ $\mu=\int_{C^+}\lvert I(\omega)-c\rvert\,d\omega$ c =desired outer color
Surrounding Outer Color()	 Surrounding Outer Color forces the Contour to have a small external neighbor of color c. The areas of c color remain out of the Contour	$E=-\frac{1}{\sigma\sqrt{2\pi}}\cdot e^{-\frac{\mu^2}{2\sigma^2}}$ $\mu=\int_{C^+}\lvert I(\omega)-c\rvert\,d\omega$ c =desired outer color
Spikeness()	 Depending on the sign, force a spiky appearance or a cloudy one c	$E=-\frac{min(\mu,\mu_{max})}{\mathcal{N}}$ $\mu=\int_{C}\kappa(s)ds$ $\kappa(s)=\frac{\partial^2 C(s)}{\partial s^2}$ μ_{max} =max value for μ $\mathcal{N}$ =normalizing constant

Generalized Active Contours 4/4

Assuming all the density function are Gaussian, we rewrite the energy terms as:

$$\Gamma = \mathcal{N}\left(\bar{\mu}, \bar{\Sigma}\right) = \mathcal{N}\left(\begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_i \\ \vdots \\ \mu_{n_f+n_g} \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} & \cdots & & \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} & \cdots & & \\ \Sigma_{31} & & \ddots & & & \\ \vdots & & & \Sigma_i & & \\ & & & & \ddots & \\ & & & & & \Sigma_{n_f+n_g, n_f+n_g} \end{bmatrix}\right)$$

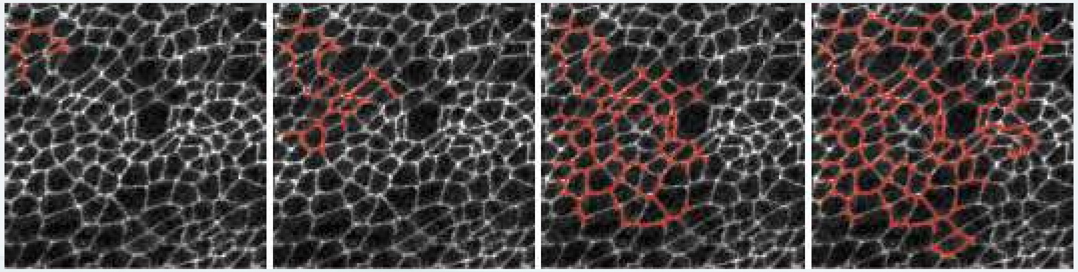
We can assess the quality of our Snaking algorithm looking at the eigenvectors: if they are close to zero, then we have a robust algorithm.

We can also simplify the matrix over certain thresholds

$$T \cdot \begin{bmatrix} \sigma_i & 0 & 0 & \cdots & & \\ 0 & \sigma_2 & 0 & \cdots & & \\ 0 & 0 & \ddots & & & \\ \vdots & & & \sigma_i & & \\ & & & & \ddots & \\ & & & & & \sigma_k \end{bmatrix} \cdot T^T$$

Reticular shape detection

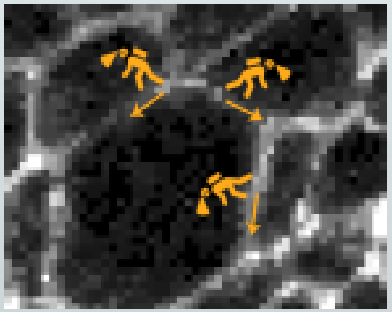
-The random walk agents approach-

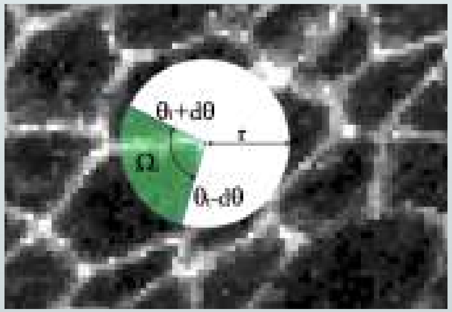


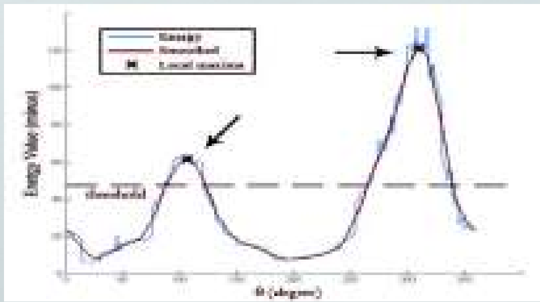
Idea: many Random Walk Agents $\mathcal{A}_i$ flood the frame, each of them finding a path. Each agent has a position $\mathbf{p}$ and an energy $\hat{\mathcal{E}}$.

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

$$\hat{\mathcal{E}}(\theta_i) = \frac{\int_{\Omega_i} \sqrt{(I(\omega)-1)^2}\, d\omega}{\int_{\Omega_i} d\omega} \qquad \mathbf{p}(t+1) = \mathbf{p}(t) + k \cdot g\left(\hat{\mathcal{E}}\right)$$







Reticular shape detection

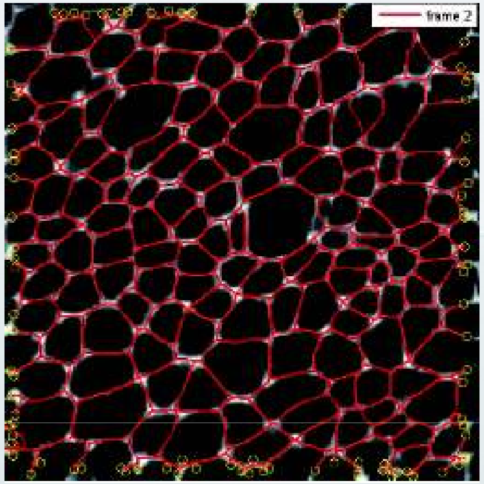
-The random walk agents approach-

Code snippet 4 - Randow Walk Agents main loop

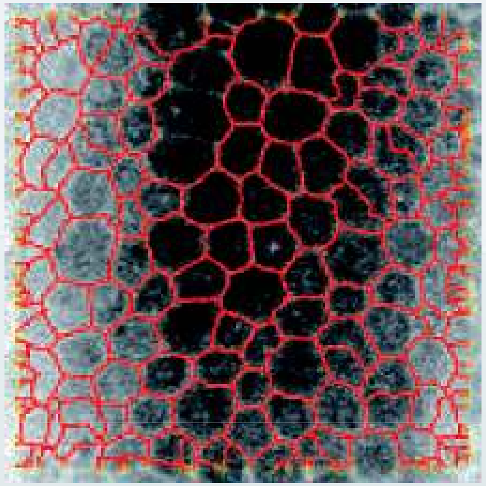
```
PQ ← initPriorityQueue()          //create a priority queue
G ← initGraph()                   //create an empty graph
while PQ not empty() {
    A ← dequeue(PQ)               //extract the best agent
    valid, border ← validateAgent(A,I,G)
    if valid
        G← add2Graph(G,A) //add to the graph
    if (valid and not(border)) {
        E ← computeEnergyFunction(I,A_i)
        D ← pickDirections(E)
        for k=1 to |D| {
            A_k ← moveAgent(A,D_k)
            PQ ← enqueue(A_k,PQ) }
    }
}
```

Reticular shape detection

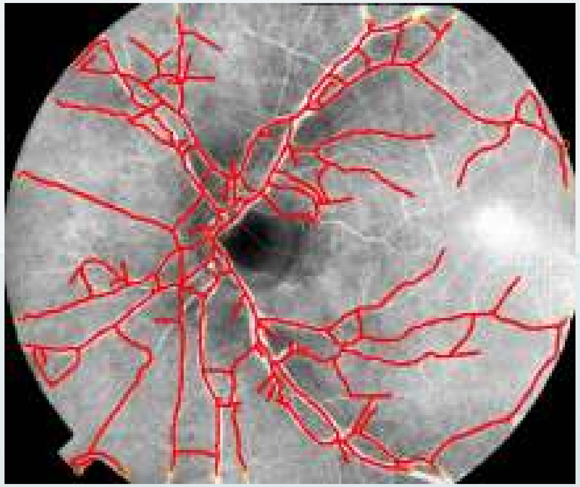
-The random walk agents approach-



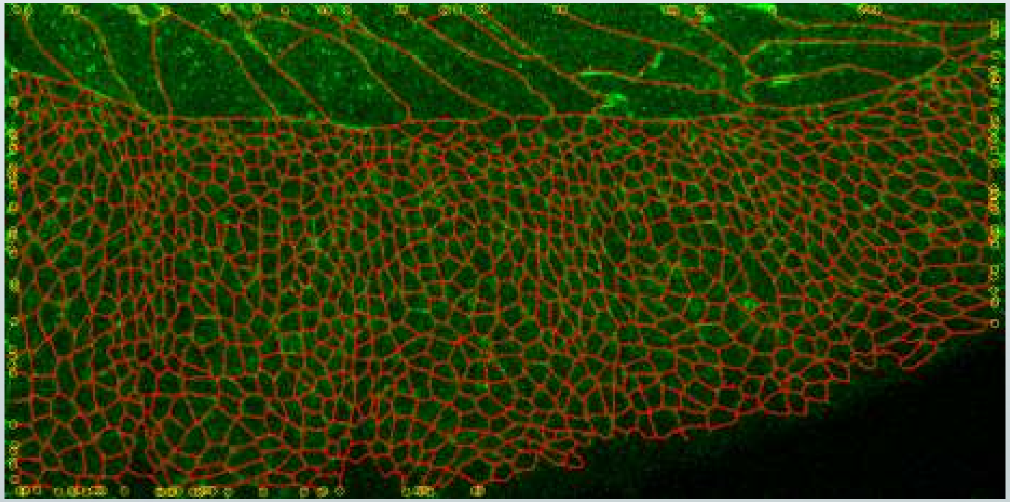
Drosophila wing



Corneal fundus



Retina



Drosophila developmental stage

Dynamic shape detection

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

$$
\begin{gathered}
\Phi_1 = \arg\min_{\Phi_c \in S} \mathcal{E}(I_1, \Phi_c) \\
\cdots \\
\cdots \\
\Phi_N = \arg\min_{\Phi_c \in S} \mathcal{E}(I_N, \Phi_c)
\end{gathered}
$$

Images temporal coherence: $I_{t+1} = I_t + \Delta_t^{t+1}$

Shape temporal coherence: $\Phi_{t+1} = \Phi_t + \delta_t^{t+1}$

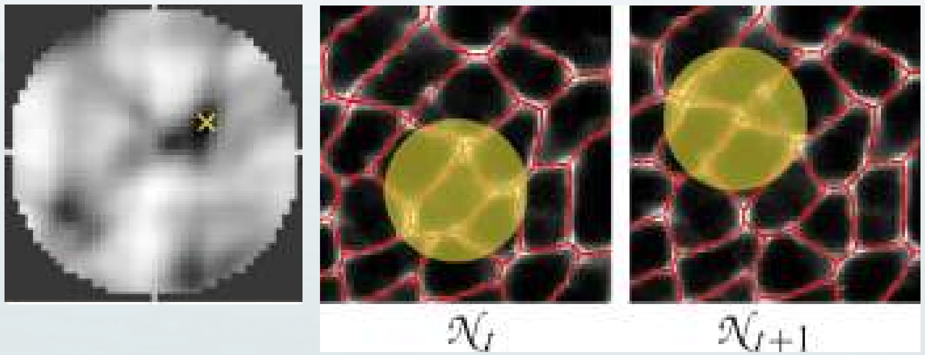
The optimum δ_t^{t+1} satisfies: $\Phi_{t+1} = \Phi_t + \delta_t^{t+1} = \arg\min_{\Phi_c \in S} \mathcal{E}(I_{t+1}, \Phi_c)$

We compute δ_t^{t+1} using the *J-maps* approach

Dynamic shape detection

-the J-maps approach-

Choosen 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 $p_{t+1} = [x + \Delta x, y + \Delta y] \in I_{t+1}$



$$J_{p_t}(\Delta x, \Delta y) = \int_P \left| \mathcal{N}_t(p_t) - \mathcal{N}_{t+1}(p_{t+1}) \right| dp$$

$$\delta_t^{t+1} = \underset{vx\ ,\ vy}{\arg\min}\ J_{p_t}$$

where $\mathcal{N}_t$ is around $p_t = [x\ ,\ y] \in I_t$,
and $\mathcal{N}_{t+1}$ 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 δ_t^{t+1}*

Dynamic shape detection

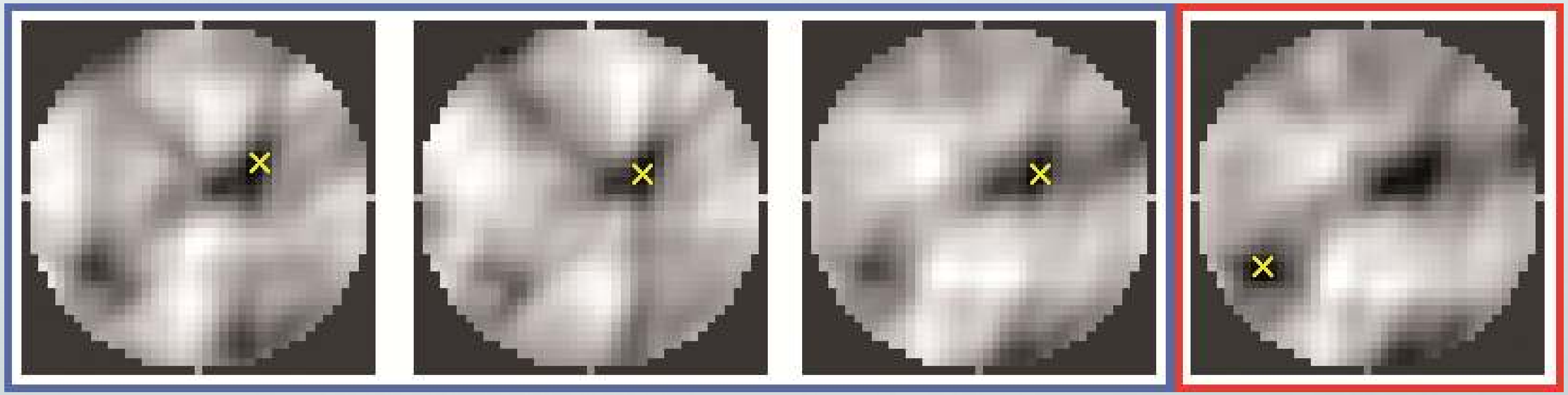
-the J-maps approach-

Correction: we impose $\frac{\partial \delta_t^{t+1}}{\partial p} \rightarrow 0$

The J-maps correction is an iterative steps. For each map J_p we consider a set of neighbor maps $\{J_{p1}, J_{p2} \ldots J_{p_n}\}$, and compute the global minima for each of them:
$\{(v_{x1}, v_{y1}), (v_{x2}, v_{x2}), \ldots (v_{xn}, v_{xn})\}$

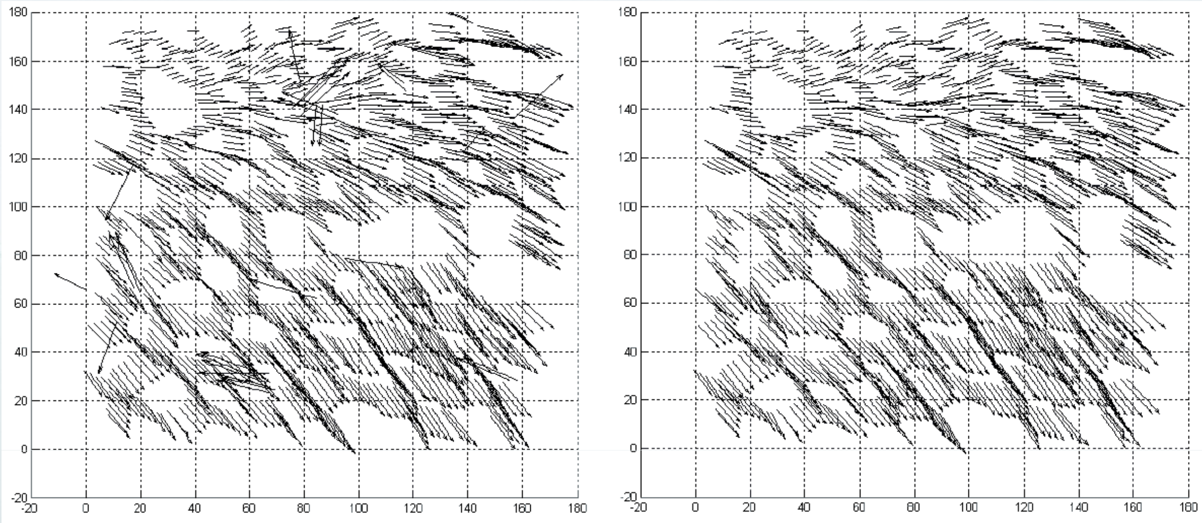
We correct J_p according to: $\bar{J}_p = J_p \cdot (1 - \mathcal{G}([\bar{v}_x, \bar{v}_y], \Sigma_{xy}))$

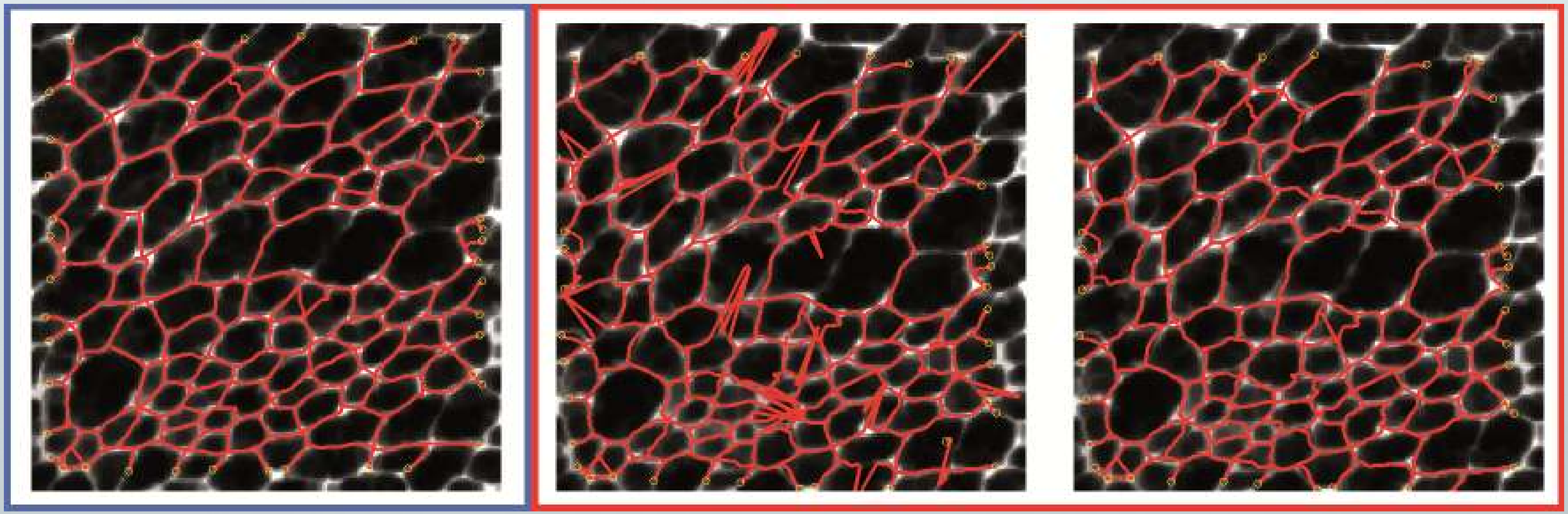
where $\mathcal{G}$ is a Gaussian build on $\{(v_{x1}, v_{y1}), (v_{x2}, v_{x2}), \ldots (v_{xn}, v_{xn})\}$



Dynamic shape detection

-the J-maps approach-



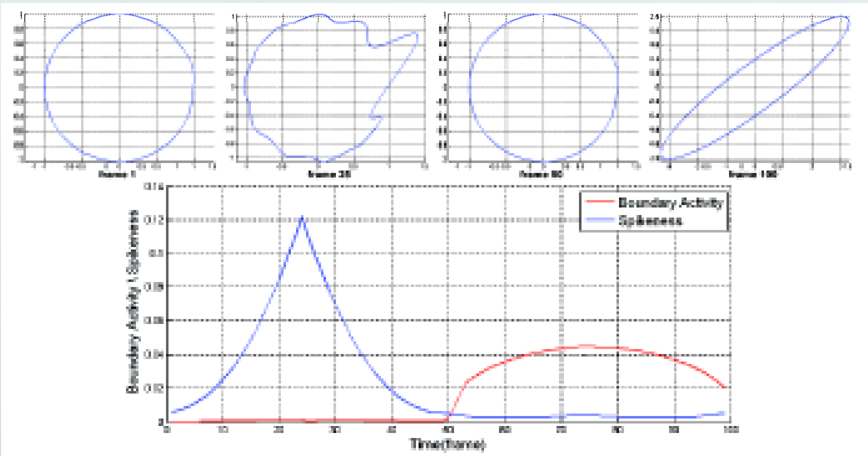


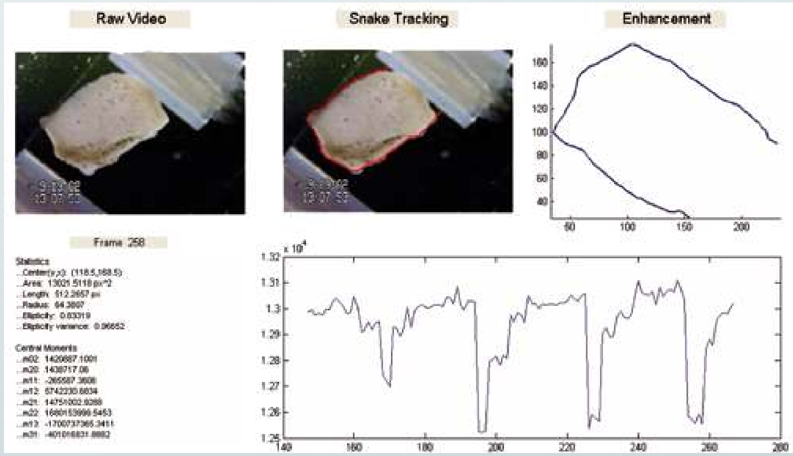
Shape analysis

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

$$\{\mathcal{R}(\Phi_0), \ldots, \mathcal{R}(\Phi_t), \ldots, \mathcal{R}(\Phi_N)\} \xrightarrow{analysis} \mathbb{R}^k$$

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.

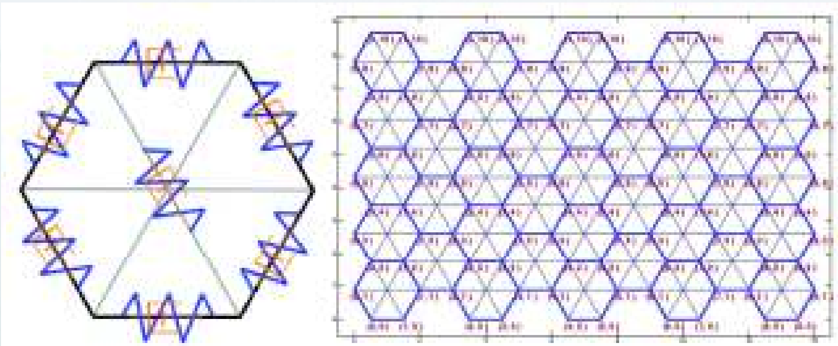


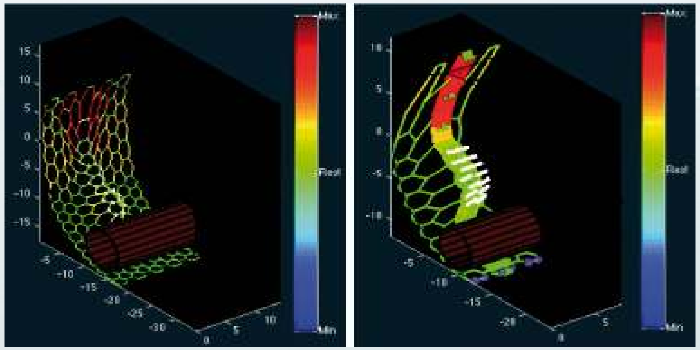


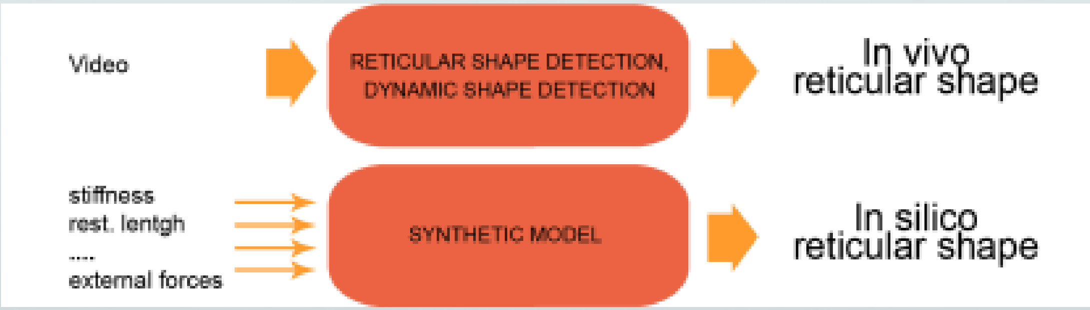
Shape synthesis

The synthesis step involves the creation of a model of the structure of interest., *mechanical, chemical, stastical…*

We build a *mechanical model* for the Drosophila epithelium.





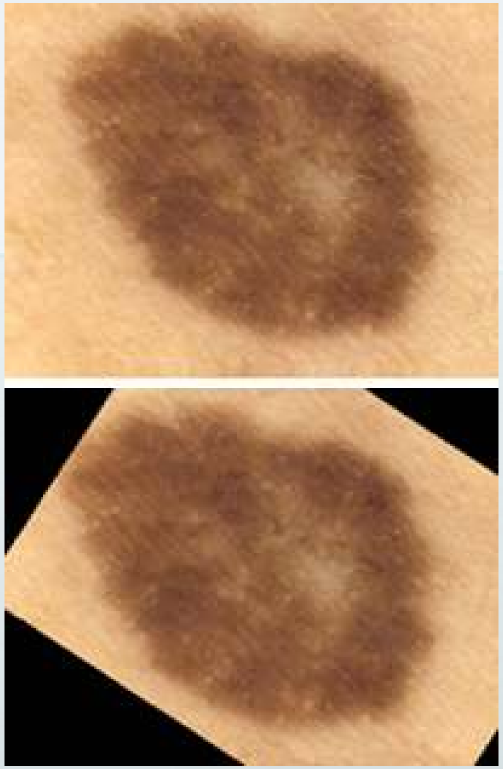


The idea is to simulate what we see in the microscopes in “silico” (Matlab in our case).

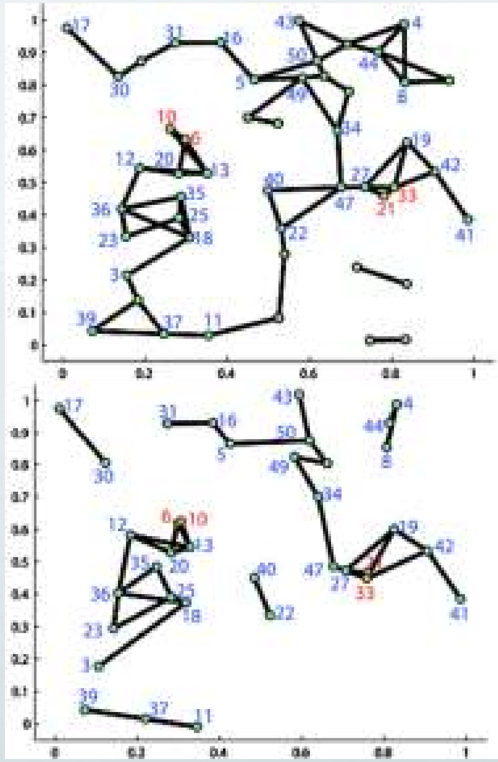
Here we use a “FEM” simulator (work in progress)

-not mentioned in the slides-

Full Body Scanner
project



Robust Registration of
dermatoscopic images



Spectral methods
for point set matching

Questions?