Medical Imaging : Connexity and Shape Constrained Image segmentation

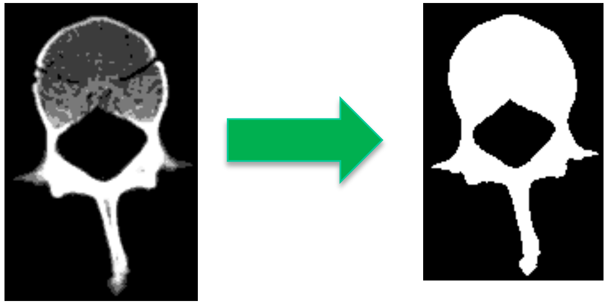
Hervé Delingette
Epione Team
Herve.Delingette@inria.fr

4. Connexity and Shape Constrained Image segmentation

- **4.1 Label Connexity Hypothesis : Markov Random Field**
  - **Definition of prior**
  - **Graph cut algorithm**
  - **Neighborhood EM**
  - **Grab Cut**
- 4.2 Introduction to shape and deformable Models
- 4.3 Snakes algorithm
- 4.4 Level Set Algorithm
- 4.5 Point Distribution Model
- 4.6 Multi-atlas Algorithm

Image Segmentation Approaches

Thresholding

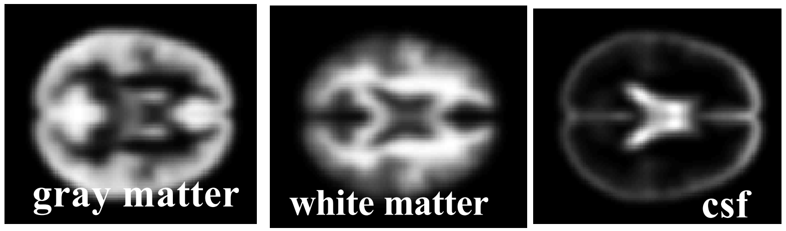


Intensity
Only

Contrast Agent
in CT

Lesions in CT / MR

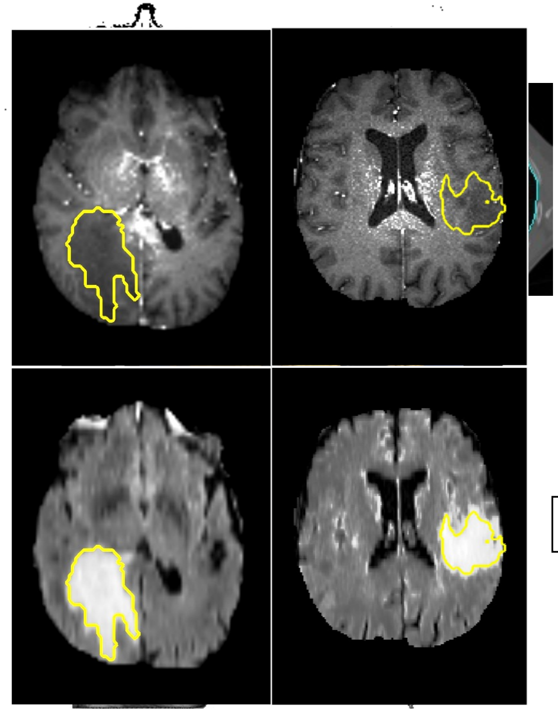
Intensity-only
Classification



No Typical Shape

Typical Shape

Image Segmentation Approaches



Intensity and connexity between regions

Mathematical Morphology

MRF (graph cuts, RW, watershed)

Machine Learning (RF, SVM, Boost,ML)

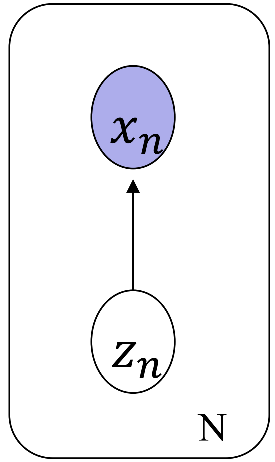
Vessels / tumors / bones /lesions

Grey / White matter in MR

No Typical Shape

Typical Shape

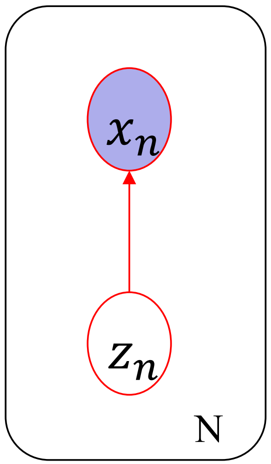
MoG Segmentation Hypothesis



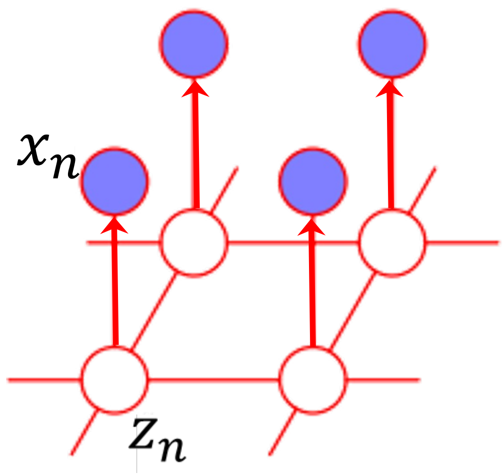
- So far considered independent voxels
  - Z_n variable specifying the class of voxel n
  - X_n variable representing the intensity
- Class membership only dependent on voxel intensity (thresholding)
- But may not be realistic in the presence of noise & partial volume effect

MRF Segmentation Hypothesis

- In Markov Random Fields :
  - Label variables z_n are no longer independent but depend on their neighbors
  - Intensity variables x_n only depends on the class label (variable z_n)



Mixture of Gaussian



Markov Random Field

Markov Random Field

- Intensity prior depends on neighboring values :

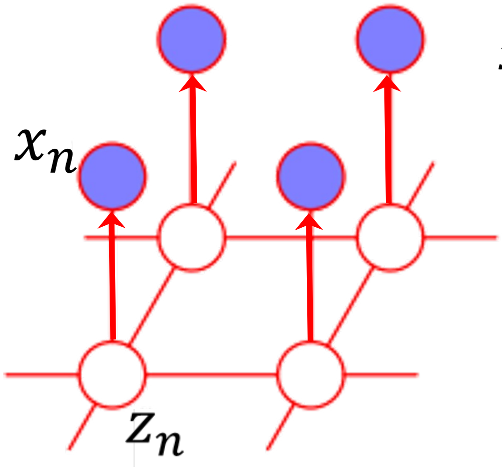
$$p(Z_n|Z_{-n}) = p(Z_n|Z_{N(n)})$$

Label at voxel n

Set of Labels of
all image voxels except
Voxel n

Labels of
Neighboring voxels
Of voxel n

- Graphical Model



x_n are independent only if z_n are known
(conditional independence)

$$p(X) \neq \prod_n p(x_n)$$

$$p(X|Z) = \prod_n p(x_n|z_n)$$

Challenges in MRF

- Posterior probability is no longer tractable

$$p(Z|X) = \frac{p(X|Z)p(Z)}{\sum_{Z'} p(X|Z')p(Z')}$$

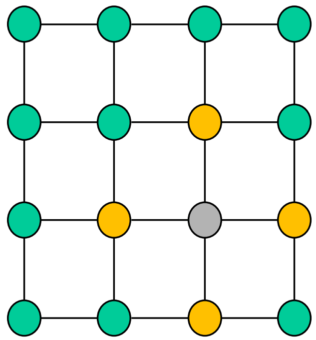
Intractable sum over 2^N terms

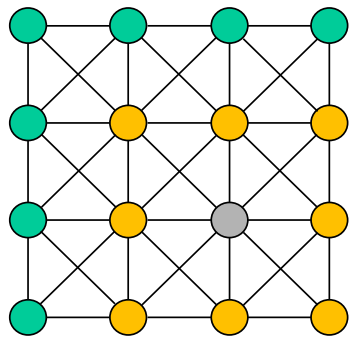
$$p(z_n|X) = \underbrace{\sum_{z_1}\sum_{z_2}\cdots\sum_{z_{n-1}}\sum_{z_{n+1}}\sum_{z_N}}_{\text{Intractable marginalization over N-1 term}} p(Z|X)$$

Definition of Label Prior in MRF

- Images seen as Graph

4 neighborhood





8- neighborhood

- Label Prior $p(Z)$ depends on neighborhood :
  - 2D images : 4 or 8 neighborhood
  - 3D images : 6, 18 or 26 neighborhood

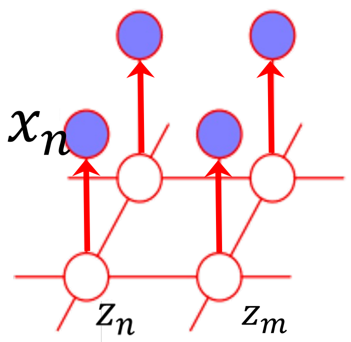
Definition of Label Prior in MRF

- Label prior $p(Z)$ is defined on a graph

  4 neighborhood : $p(Z_n|Z_{-n}) = f(Z_{n-1}, Z_{n+1}, Z_{n-R}, Z_{n+R})$

- Hammersley-Clifford theorem gives the expression of $p(Z)$:
  - There exists functions ψ and ϕ such that

$$\log p(Z|\theta) = \frac{-1}{T} \sum \underbrace{_{edges(n,m)} \psi(z_n, z_m, \theta)}_{\text{Binary term}} - \frac{1}{T^*} \underbrace{\sum_n \phi(z_n, \theta)}_{\text{Unary term}}$$



$\psi(z_n, z_m, \theta)$ is any function of 2 Binary vectors : it enforces how likely are two labels are different

$\phi(z_n, \theta) = \phi_n$

Gives how likely voxel n belongs to class k

Potts Model for Label Prior

- Idea : neighboring voxels should have similar labels.
- Definition Ising when K=2 :
  - One hot encoding : $Z_n = (Z_{n1}, Z_{n2} \dots Z_{nK})^T$
  - $\psi(z_n, z_m, \theta) = -\sum_{k=1}^{K} f_{nm} z_{nk}\, z_{mk}$,
  - In another words :
    - $\psi(z_n, z_m, \theta) = -f_{nm}$ if $Z_n = Z_m$ and $\psi(z_n, z_m, \theta) = 0$ if $Z_n \neq Z_m$,
- Alternative 1 : $\psi(z_n, z_m, \theta) = f_{nm} \| Z_n - Z_m \|^2$
- Coefficient definition : neighboring voxels having similar intensity should have the same labels.

$$f_{nm} = \exp -\beta (x_n - x_m)^2$$

Joint Probability in MRFs

- Definition of joint probability :
  - $p(X, Z|\theta) = p(Z)p(X|Z)$
- Log joint probability

$$\Lambda(Z,\theta) = \log p(X,Z|\theta) = \log p(Z|\theta) + \log p(X|Z,\theta)$$

Conditional independence ⇒

$$\Lambda(Z,\theta) = \log p(Z|\theta) + \sum_n \log p(x_n|z_n,\theta)$$

Categorical variable ⇒

$$\Lambda(Z,\theta) = \log p(Z|\theta) + \sum_n \sum_k z_{nk} \log p(x_n|z_{nk}=1,\theta)$$

Energy

$$-\Lambda(Z,\theta) = \underbrace{\frac{1}{T} \sum_{edges(n,m)} \psi(z_n, z_m, \theta)}_{\text{Binary term}} + \frac{1}{T^*} \underbrace{\sum_n \phi(z_n,\theta) - \sum_n \sum_k z_{nk} \log p(x_n|z_{nk}=1,\theta)}_{\text{Unary terms}}$$

Algorithms for solving MRF

- Many existing algorithms :
  - 1) **Graph cut Algorithm** :
    - Fast
    - solve for hard memberships z_{nk}
    - Unique solution for K=2 if some constraints on f_{nm} are met
    - Several extensions for K>2
  - 2) **Neighborhood EM**
    - solve for soft memberships $\mathrm{p}(z_n|x_n)$
    - Simple Extension of GMM
    - Fixed point Iterative method
  - 3) **Grab Cut**

Graph cuts

- Binary case & Ising model :
  - 2 labels case $y_i \in \{0,1\}$
  - Minimize energy :
    $E(Y) = \sum_{i,j} c_{ij} y_i (1 - y_j) + \sum_i d_i y_i$, with $d_i > 0$
  - Submodular constraint for unique solution

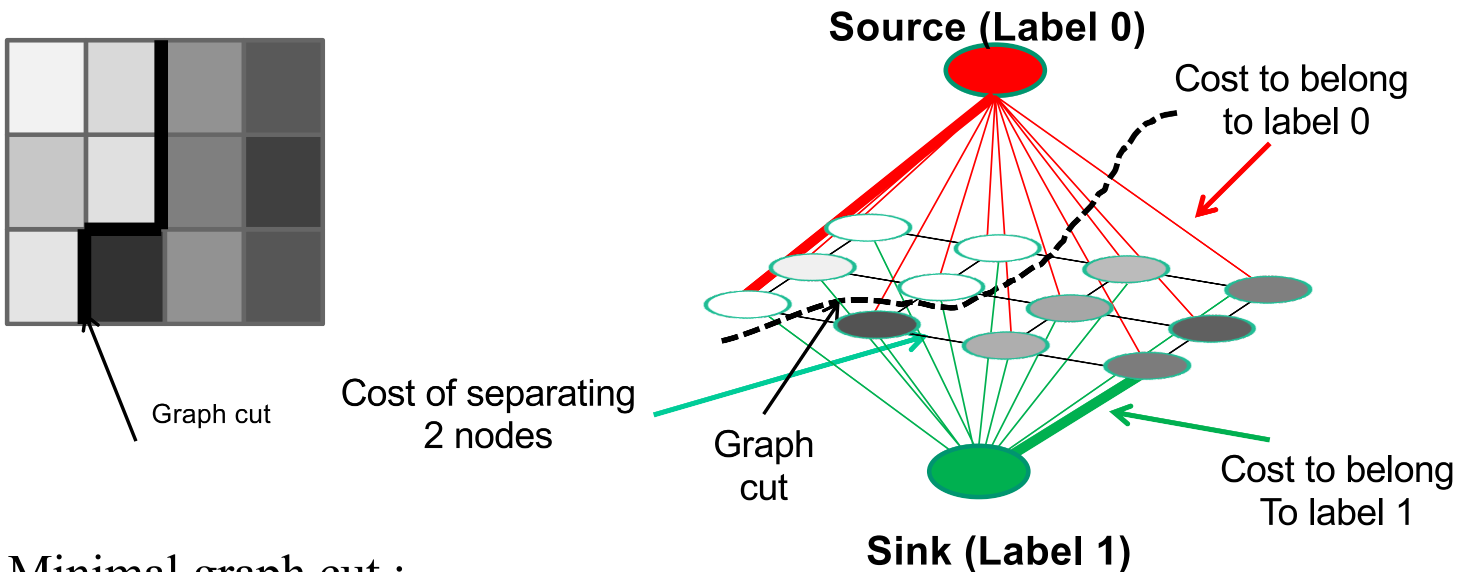
$$c_{ij} + c_{ji} \geq 0$$

  - Minimize $E(Y)$ $\Leftrightarrow$ Minimize a graph cut

Combinatorial problem

D.M. Greig, B.T. Porteous and A.H. Seheult (1989), *Exact maximum a posteriori estimation for binary images*, Journal of the Royal Statistical Society Series B, **51**, 271–279.

Graph Cut

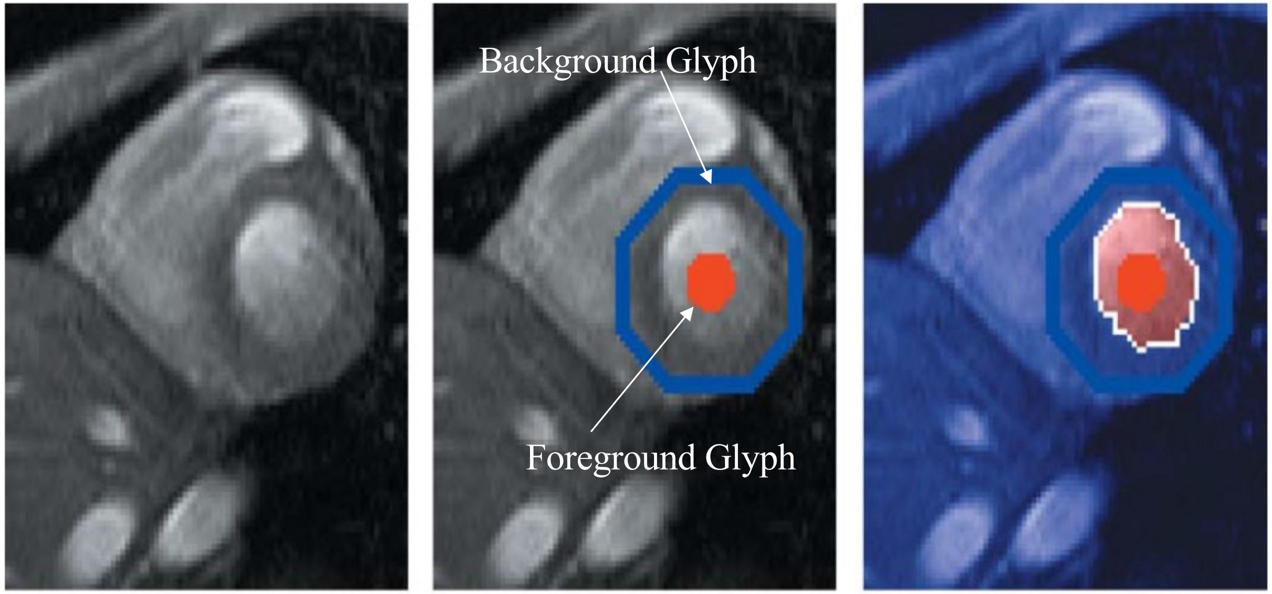


- Minimal graph cut :
  - Set of edges whose removal create several connected components:
  - Cost of a cut :

$$cut(A,B) = \sum_{p \in A, q \in B} c_{p,q}$$

Maximize the flux between the source and the sink nodes

Interactive Segmentation Algorithm



Source : Boykov & Gareth Funka-Lea Graph Cuts and Efficient N-D Image Segmentation

Manual glyph from user to guide segmentation

Graph cut Segmentation

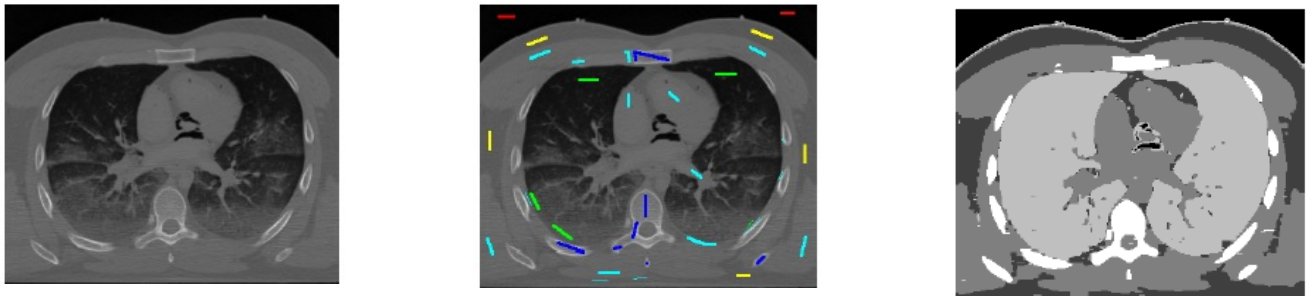
- Combinatorial algorithm for graph cut :

  Ford & Fulkerson Algorithm (1951)

  BoyKov & Kolmogorov Algorithm (2004)

  Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(9):1124–1137, September 2004.

- Multi Label Segmentation with α-expansion algorithm [Veksler 99] [Boykov 99]



R. Kéchichian, S. Valette, M. Desvignes, R. Prost: **Efficient multi-object segmentation of 3D medical images using clustering and graph cuts.** ICIP 2011

Neighborhood EM

- Hypothesis :
  - Posterior probability $p(z_n|X\quad)$ is intractable therefore estimate an approximation
  - Each tissue class is represented by a Gaussian distribution

$$p(x_n|z_{nk}=1)=\mathcal{N}(x_n|\theta_k)$$

  - The label prior is a Potts model and global prior per class

$$\log \mathrm{p(Z)} = -\frac{\beta}{2}\sum_{k}\sum_{edges(m,n)} c_{nm}\, z_{nk} z_{mk} + \sum_{n}\sum_{k} \pi_k z_{nk}$$

C. Ambroise , M. Dang , G. Govaert: Clustering of Spatial Data by the EM Algorithm. In geoENV I-Geostatistics for Environmental Applications (1997), pp. 493-504.

Mean Field approximation

- A.ka Variational Bayes approach
  - Look for an approximation of posterior parameters as product $q(Z) = \{q_n\}$ of factorized terms $p(Z = \{z_n\}|X) \approx \prod_n q_n(z_n)$
  - Therefore NK unknown q_{nk} s.t

$$\mathrm{q_n}(z_n) = \sum_k q_{nk} z_{nk} \;\; \& \sum_k q_{nk} = 1 = \sum_{z_n} q_n(z_n)$$

  - Find the set q which minimizes the Kullback Leibler divergence between q and true posterior $p(Z|X)$

Mean Field Criterion

- Reminder EM criterion for GMM :
  - Maximize =F(π, θ, u)

$$\mathrm{F}(\pi, \theta, u) = \mathrm{L}(\pi, \theta) - D_{KL}\big(u||p(z|x)\big) = \mathrm{Q}(\theta, u) + \mathrm{H}(u)$$

- Evidence Lower bound :

$$D_{KL}\big(q||p(Z|X)\big) = -\log p(X) - \mathbb{E}_q\,(\log p(X, Z)) - H(q)$$

- Neighborhood EM criterion same as GMM but with additional term $R(q)$

minimize $D_{KL}\big(q|p(Z|X)\big) = -\mathrm{H}(q) + R(q) - \mathrm{Q(q)} + \log p(X)$

  - Where $R(q) = \frac{\beta}{2} \sum_k \sum_{edges(n,m)} c_{nm}\, q_{nk} q_{mk}$

Neighborhood EM

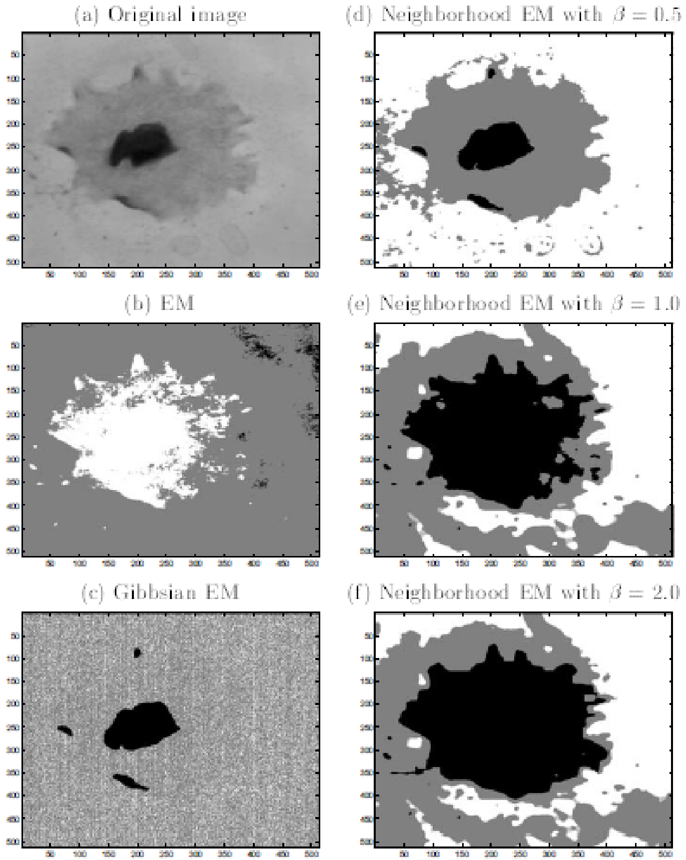
- Only E-step changed compared to regular EM for GMM
- New E-step :
  - Fixed point iteration

$$q_{nk} = \frac{\pi_k \mathcal{N}(x_n|\theta_k) \exp \beta \sum_m c_{mn} q_{nm}}{\sum_l \pi_l \mathcal{N}(x_n|\theta_l) \exp \beta \sum_m c_{mn} q_{nm}}$$

- Same M-step

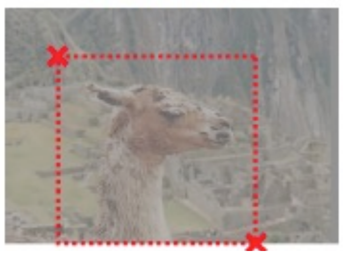
$$\mu_k = \frac{\sum_{n=1}^{N} q_{nk}\, x_n}{\sum_{n=1}^{N} q_{nk}} \qquad \Sigma_k = \frac{\sum_{n=1}^{N} q_{nk}\,(x_n - \mu_k)(x_n - \mu_k)^T}{\sum_{i=1}^{N} q_{nk}} \qquad \pi_k = \frac{1}{N}\sum_{n=1}^{N} q_{nk}$$

Neighborhood EM

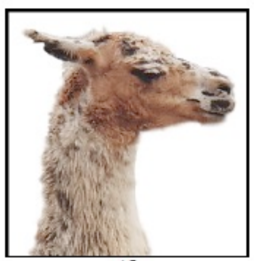


Grab Cut

- Algorithm combines :
  - Model intensity of foreground and background as mixture of Gaussians (vs one Gaussian for each class)
  - Iterate between :
    - hard segmentation using graph cuts
    - Estimation of Gaussian components

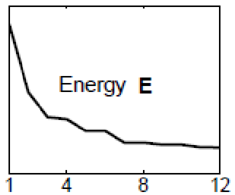


Input + bounding box

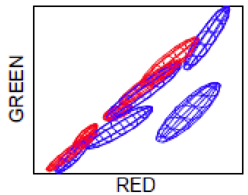


(f)

Segmentation

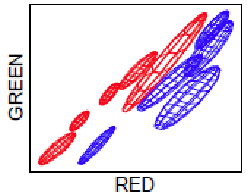


(a)



(b)

Initial GMM



(c)

Final GMM

Grab Cut Examples

Available in MS Office !!

Difficult Examples

Camouflage & Low Contrast | **Fine structure** | **Harder Case**

Initial Rectangle

Initial Result

Grabcut: Interactive foreground extraction using iterated graph cuts, Carsten Rother, V. Kolmogorov, Andrew Blake, Siggraph 2004

4. Connexity and Shape Constrained Image segmentation

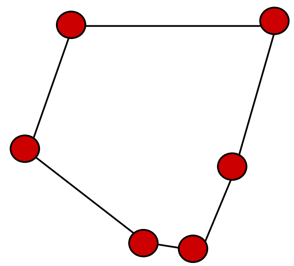
- 4.1 Label Connexity Hypothesis : Markov Random Field
- **4.2 Introduction to shape and deformable Models**
- 4.3 Snakes algorithm
- 4.4 Level Set Algorithm
- 4.5 Point Distribution Model
- 4.6 Multi-atlas Algorithm

Shape Constraints in Image Segmentation

- MRFs enforce connectivity between neighboring voxels : region approach
- Deformable shapes / models :
  - Work on boundaries between regions -> dual approach
  - Define constraints on the boundaries :
    - Minimize length
    - Minimize curvature
    - Shape constraints

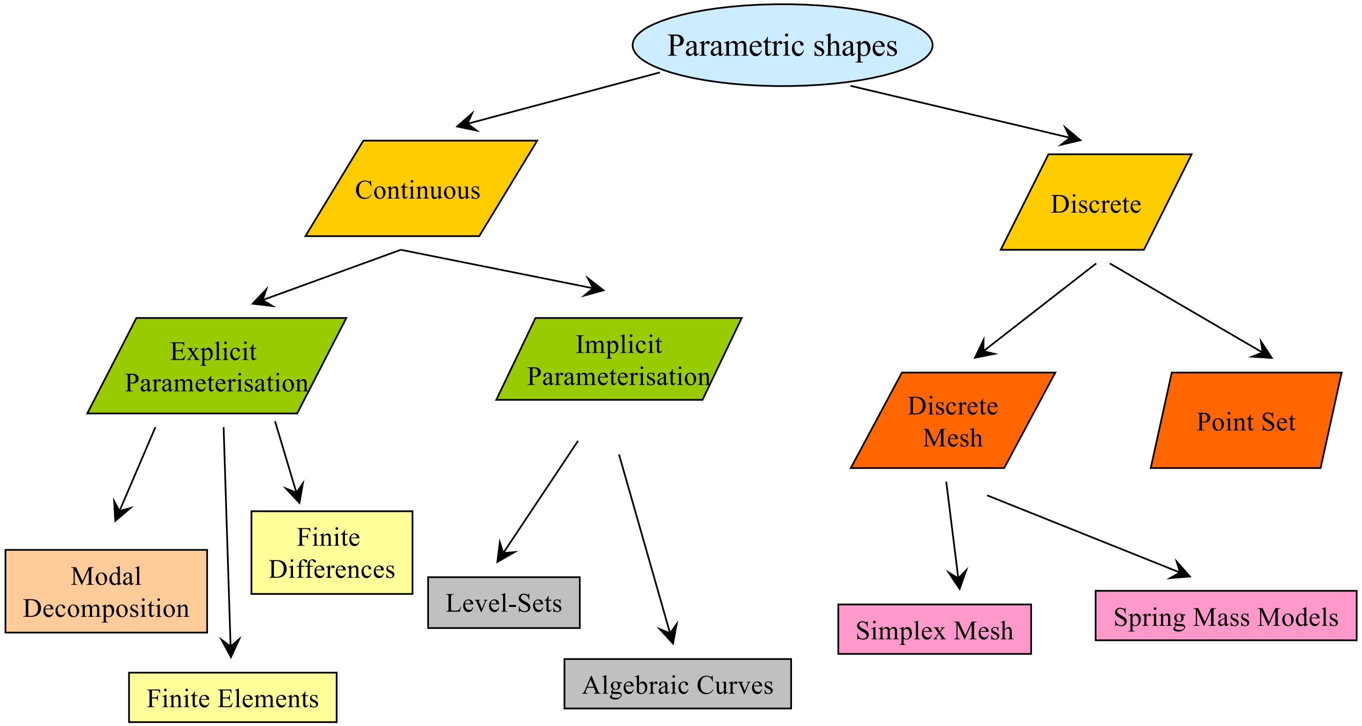
Parametric Shape representation

- Parametric representation of a shape :
  - Shape controlled by (intrinsic) parameters
- Examples :
  - Vertex position of a mesh
  - Scalar field for level sets
  - Fourier coefficients,…



Deformation in the object space

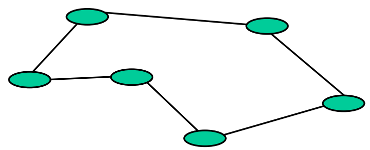
Shape representation

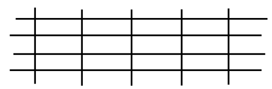


Shape representation As Template Transformation

- Template Transformation :
  - Define a single shape instance in $\mathbb{R}^n$ as template
  - Parameterise the deformation of the embedding space $\phi(x): \mathbb{R}^n \to \mathbb{R}^n$
- Examples :
  - Rigid Transformation (translation + rotation)
  - Affine Transformation (translation + linear transform)

Shape instance= template





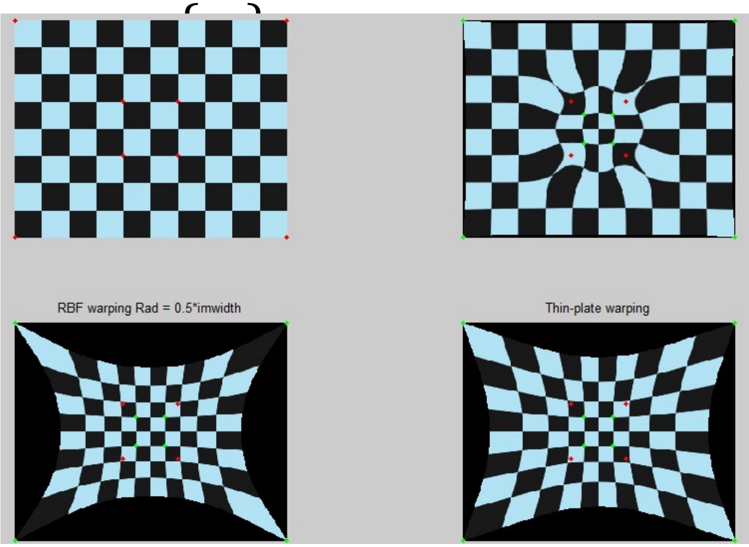
Define $\phi(x)$ as an affine transform

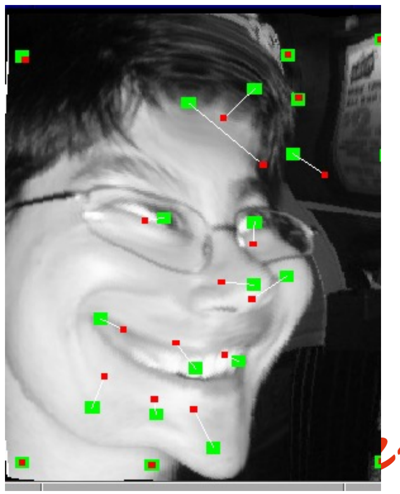
Simple Transformations

T_{reg}	**Description**	**Degrees of Freedom**
2D Rigid	Translation + Rotation	2+1=**3**
2D Similarity	Translation + Rotation + Scale	3+1=**4**
2D Affine	Translation + Linear	2+4=**6**
3D Rigid	Translation + Rotation	3+3=**6**
3D Similarity	Translation + Rotation + Scale	6+1=**7**
3D Affine	Translation + Linear	3+9=**12**

Complex Transformations

- Radial Basis functions :
  - Basis $\psi(x) = \psi(\|x\|)$ which only depend on distance : example : Gaussian, thin plate spline, B-spline
  - Define N control points x_i
  - Define $\phi(x)$ as $\phi(x) = \sum_i^N \psi(x - x_i) y_i$ parameterized by





Shape Optimization

- If $\{\theta\}$ are parameters in the shape space (parametric representation)

→ Framework of deformable templates

- If $\{\theta\}$ are parameters in the space of geometric transformations

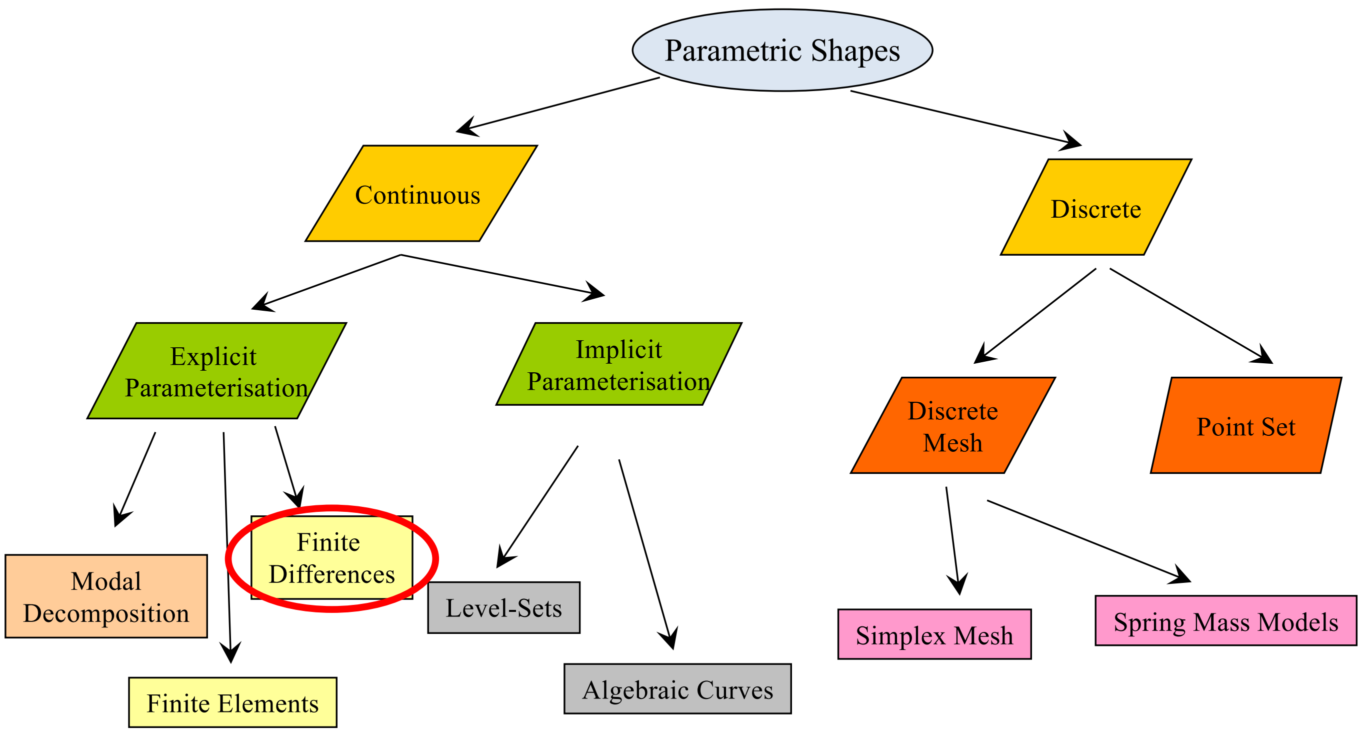
→ Framework of Image Registration

- Often includes both frameworks

4. Connexity and Shape Constrained Image segmentation

- 4.1 Label Connexity Hypothesis : Markov Random Field
- 4.2 Introduction to shape and deformable Models
- **4.3 Snakes algorithm**
- 4.4 Level Set Algorithm
- 4.5 Point Distribution Model
- 4.6 Multi-atlas Algorithm

Shape representation



Snake Algorithm

- Energy Definition :

$$E = E_{\text{int}} + E_{\text{ext}}$$

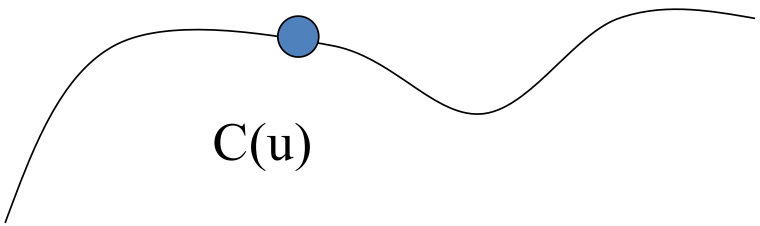
- E_{int} measures the contour smoothness
- E_{ext} measures the distance of the contour to the visible border of the object of interest
- Variational problem : minimize E

Contour Representation

- Explicit Representation of a contour

$$C : [a, b] \rightarrow \mathbb{R}^2$$

$$\boldsymbol{C}(u) = \begin{pmatrix} x(u) \\ y(u) \end{pmatrix}$$



Contour Representation(2)

- Geometry Reminder:
  - Tangent Vector :

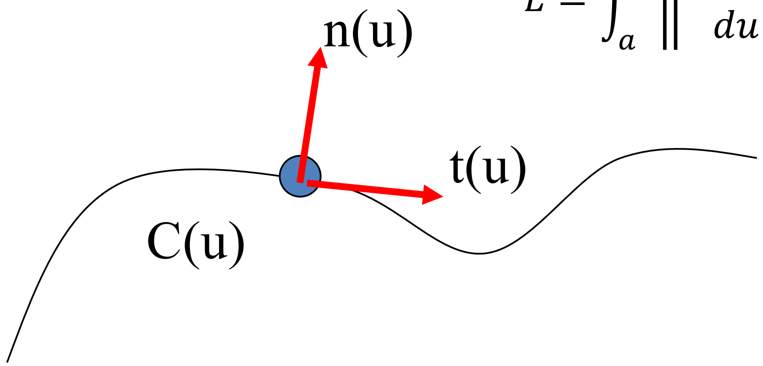
$$t(u) = \frac{\frac{d\boldsymbol{C}(u)}{du}}{\left\|\frac{dC(u)}{du}\right\|}$$

  - Normal Vector :

$$n(u) = t(u)^{\perp}$$

  - Curve Length :

$$L = \int_a^b \left\|\frac{d\boldsymbol{C}(u)}{du}\right\| du$$



Internal Energy (1)

- Internal energy is the sum of 2 terms :
  - Stretching energy $E_{stretching}$ which measures the change of length of a curve
  - Bending energy $E_{bending}$ which measures the change of curvature along the curve
  - Use of Sobolev norms to simplify numerical solution

Stretching Energy

- Expression :

Dirichlet Energy

$$E_{\text{stretching}} = \alpha \int_a^b \left\| \frac{dC(u)}{du} \right\|^2 du$$

- Link with curve length :

$$L = \int_a^b \left\| \frac{d\boldsymbol{C}(u)}{du} \right\| du \leq \int_a^b \left\| \frac{d\boldsymbol{C}(u)}{du} \right\|^2 du$$

- Extension :

$$E_{\text{stretching}} = \alpha \int_a^b \left\| \frac{dC(u)}{du} \right\|^2 du$$

Bending Energy

- Expression :

$$E_{\text{bending}} = \beta \int_a^b \left\| \frac{d^2 C(u)}{du^2} \right\|^2 du$$

- Link with beam bending energy :

$$E_{Beam} = \int_a^b E(u) I(u) k^2(u) du$$

- Extension :

$$E_{\text{bending}} = \beta \int_a^b w_2(u) \left\| \frac{d^2 C(u)}{du^2} \right\|^2 du$$

$W_2(u)=1$ except at C^1 discontinuities

External Energy

- Main Idea : attract the contour towards high gradient voxels
- 2 formulations :
  - Local using gradient image
  - Global using contour points

$$E_{\text{ext}} = E_{\text{local}} + E_{\text{global}}$$

Local External Energy (1)

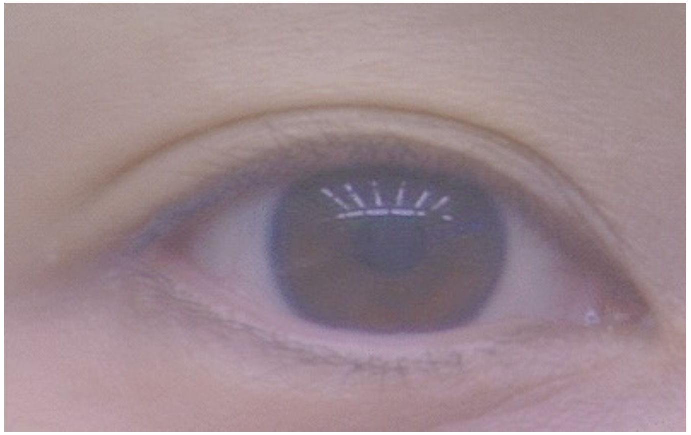
- Local Energy
  - Gradient Computation by convolving with the derivative of Gaussian $\mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix}; 0, \sigma\right)$

$$\nabla I(x,y) = \nabla\mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix}; 0, \sigma\right) \star I(x,y) = \iint \nabla\mathcal{N}\left(\begin{pmatrix} x \\ y \end{pmatrix}; 0, \sigma\right)(u-x, v-y)\, I(u,v)\, du\, dv$$

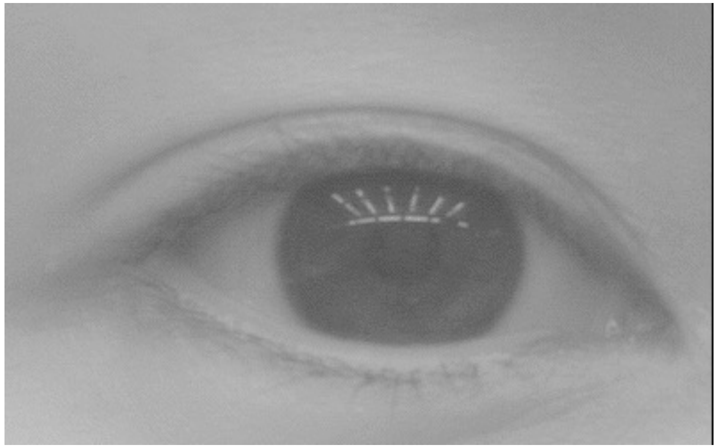
  - The standard deviation σ of Gaussian allows to control the smoothness

Local External Energy (2)

- Example



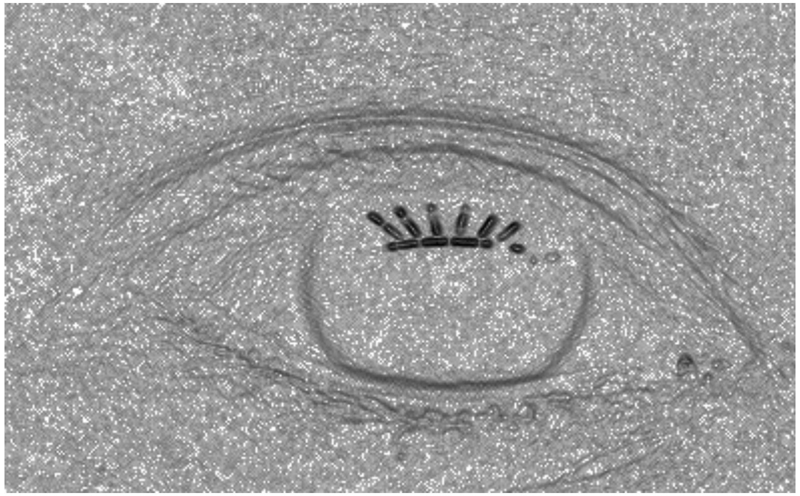
Original Image



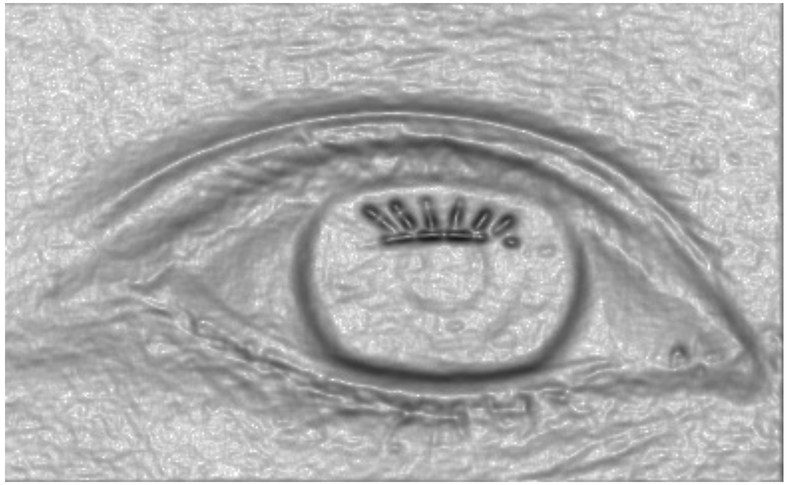
Red Band

Local External Energy(3)

- Computation of the gradient norm $-\|\nabla I(x,y)\|$



Low σ



High σ

Local External Energy (4)

- Definition of the local external energy

$$E_{local} = -\|\nabla I(x, y)\|^2$$

- The contour is driven towards minima of potential whose width is linked to σ

Global External Energy (1)

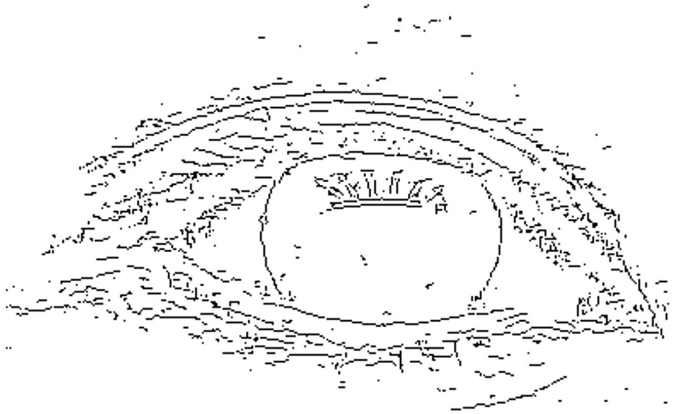
- Main Idea :
  - Select high gradient pixels which correspond to border between 2 regions
  - Define a potential field as a distance map from those pixels

Global External Energy (2)

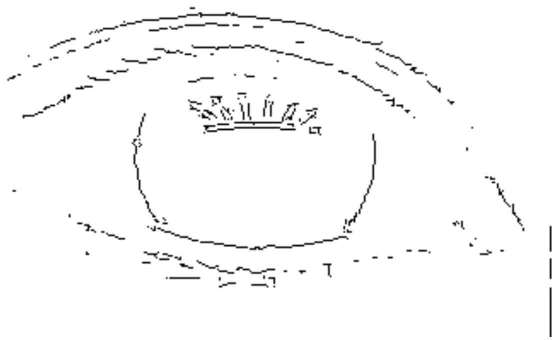
- How are contour points defined ?
- Contour point extraction algorithm :
  - Compute gradient $\nabla I(x,y)$ and its norm $\|\nabla I\|(x,y)$ at each voxel
  - Extract extrema of gradient in the direction of gradient
  - Threshold those extrema based on the gradient norm
  - Construction of a potential field E_{global}

Global External Energy (3)

- Example



Extrema of gradient



Threshold of Extrema of gradient

Global External Energy (4)

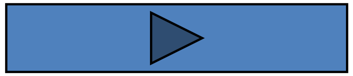
- Computation of potential field $E_{global}(x,y)$:
  - Use of chamfer distance which approximates the Euclidean distance

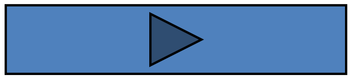
5	4	3	2	1	2	3
4	3	2	1	0	1	2
4	3	2	1	0	1	2
3	2	1	0	0	0	1
3	2	1	0	0	1	2
4	3	2	1	1	2	3
5	4	3	2	2	3	4

Example
in 4-connexity

Global External Energy(5)

Distance Map





Problem Position

- Variational Problem :

Find C(u) which
minimize :

$$\boxed{E(C(u)) = E_{\text{int}}(C) + E_{\text{ext}}(C)}$$

- Necessary condition for C(u) :

$$\delta E(C(u)) = 0$$

Numerical Approach

- Use calculus of variation to compute $F_{\text{int}}(C) + F_{\text{ext}}(C) = 0$
- Use Lagrangian Evolution :

$$\frac{\partial C}{\partial t} = F_{\text{int}} + F_{\text{ext}}$$

- Use Finite Difference discretization

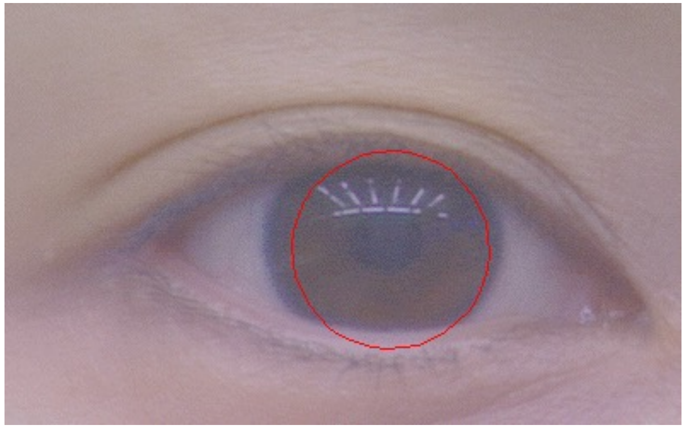
$$F_{\text{int}} = \alpha \frac{d^2 C(u)}{du^2} - \beta \frac{d^4 C(u)}{du^4} \Rightarrow F_{\text{int}}(P_i) = \alpha \frac{P_{i-1} - 2P_i + P_{i+1}}{h^2} - \beta \frac{(P_{i-2} - 4P_{i-1} + 6P_i - 4P_{i+1} + P_{i+2})}{h^4}$$

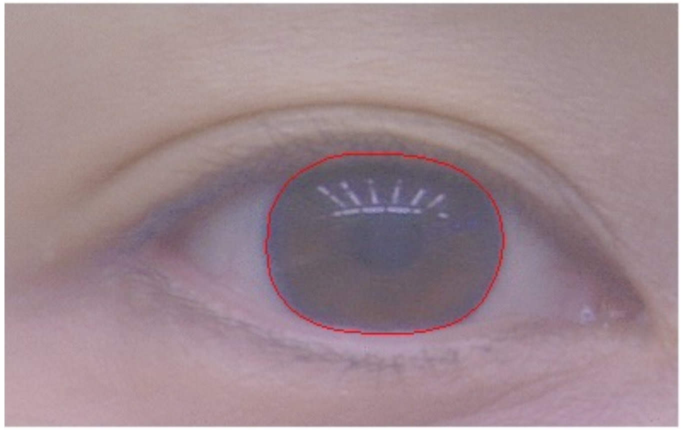
- Use semi-implicit time integration scheme

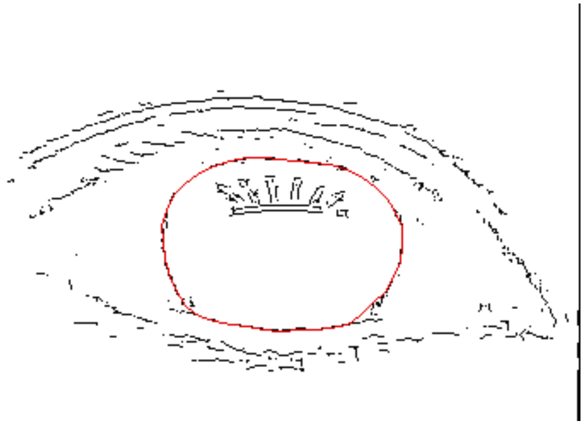
$$X^{t+1} = (I - \Delta t\, K)^{-1} \left(X^t + \Delta t\, F_{ext}^x (X^t, Y^t) \right)$$

$$Y^{t+1} = (I - \Delta t\, K)^{-1} \left(Y^t + \Delta t\, F_{ext}^y (X^t, Y^t) \right)$$

Result



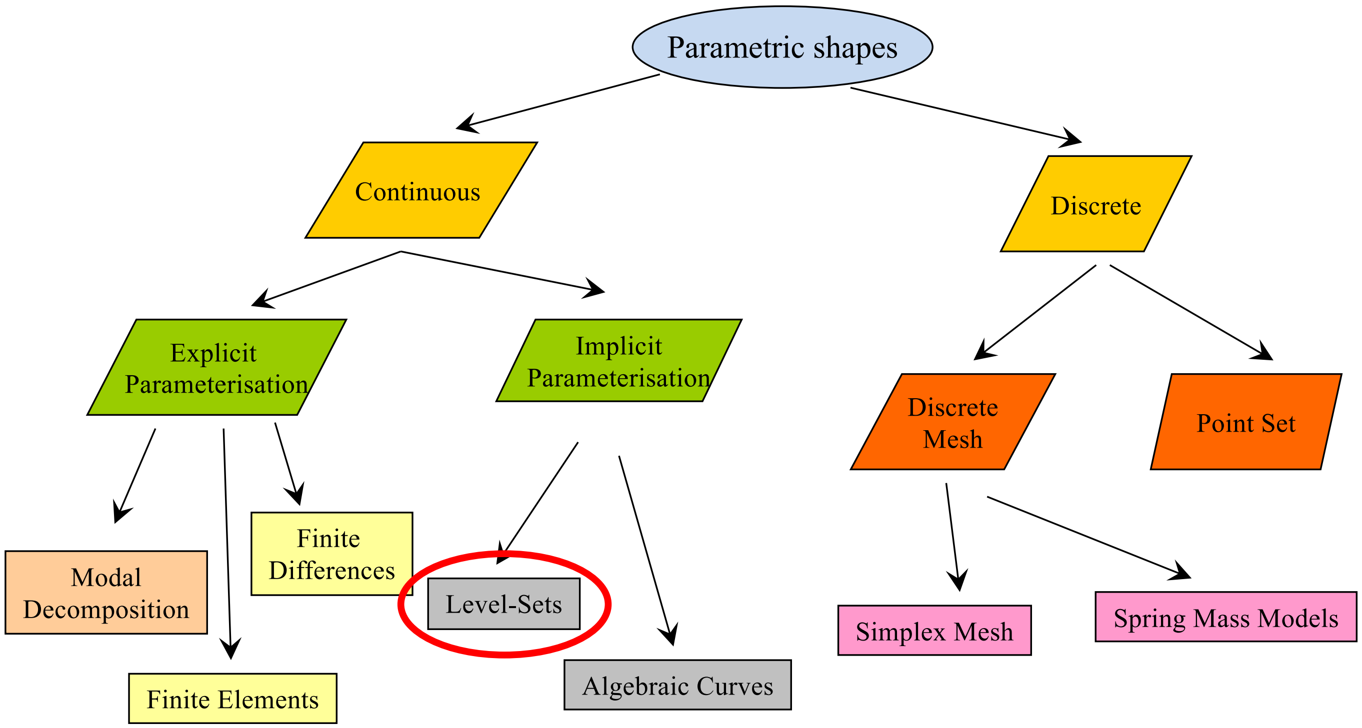




4. Connexity and Shape Constrained Image segmentation

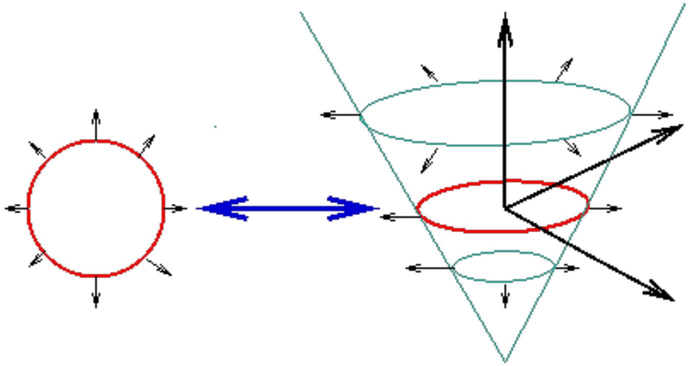
- 4.1 Label Connexity Hypothesis : Markov Random Field
- 4.2 Introduction to shape and deformable Models
- 4.3 Snakes algorithm
- **4.4 Level Set Algorithm**
- 4.5 Point Distribution Model
- 4.6 Multi-atlas Algorithm

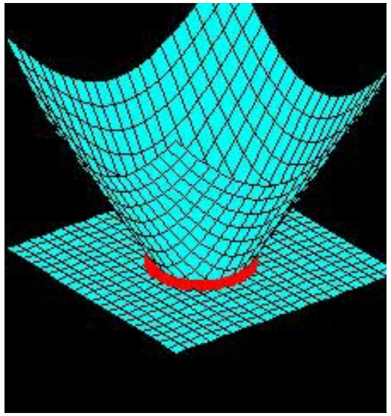
Shape representation



Level Sets

- How to define boundary curves :
  - Curve / Surface represented as the zero crossing of a scalar function : $\phi(x,t)=0$
  - The scalar field evolves over time

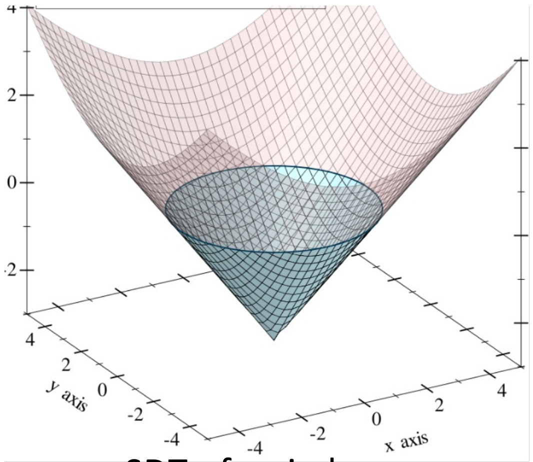




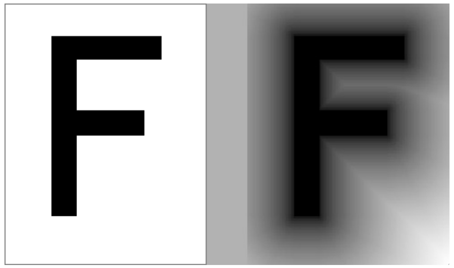
Source Fast Marching Methods and Level Set Methods, J.A. Sethian

How to define scalar field ?

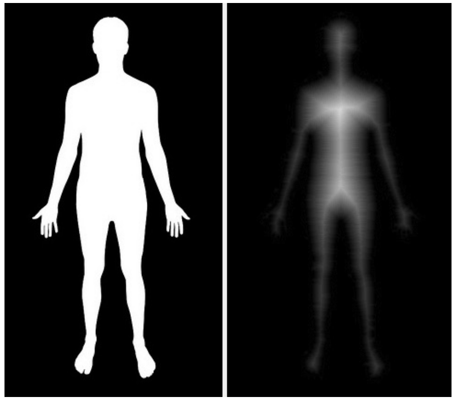
- Use regular grid of the input image
- Initialize $\phi(x, 0) = SDT(\mathcal{S}_0)$ as signed distance transform of an initial shape $\mathcal{S}_0$



SDT of a circle



Positive distance



Negative distance

Shape representation

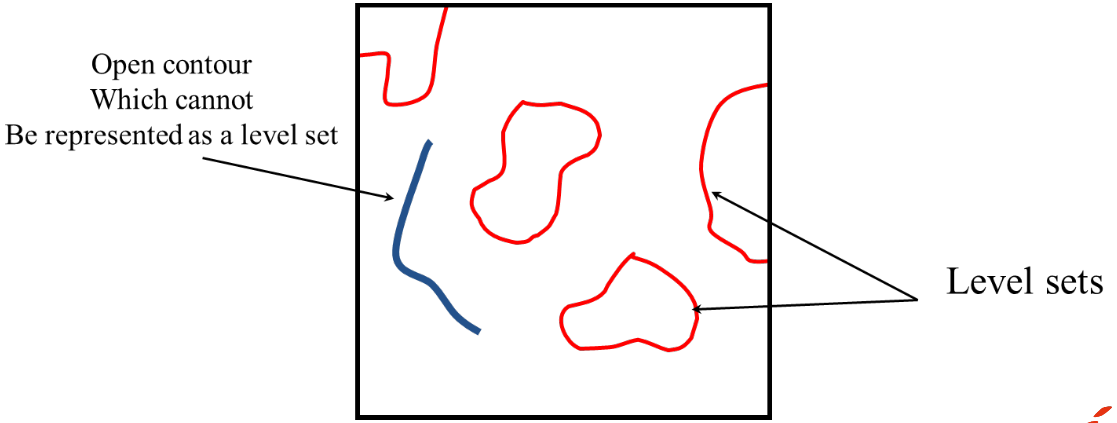
- Positive aspects :
  - Can represent all topologies (sphere, torus..)
  - Simple : No parameterization !!
  - Can define normal and curvature from derivatives of scalar field !!

$$n(x,t) = \frac{\nabla \phi(x,t)}{\left\|\nabla \phi(x,t)\right\|} \qquad \text{For } x \,/ \quad \phi(x,t) = 0$$

$$k(x,t) = -\frac{\nabla \phi^T H \nabla \phi}{\left\|\nabla \phi\right\|^2} = -\frac{\phi_{xx}\phi_y^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_x^2}{\left(\phi_x^2 + \phi_y^2\right)^{3/2}}$$

Limitations of level sets

- Topology restricted to closed contours (except at the image borders)
- Uniform discretisation that depends on a regular grid
- Difficult to handle manifold of co-dimension > 1



Evolution of level sets

- Contour defined as : $\phi(x,t) = \phi(C(u,t),t) = 0$

- Normal vector defined as : $\frac{d\phi(C(u,t),t)}{du} = 0 = \nabla\phi \cdot \frac{dC}{du}$

$$n(x,t) = \frac{\nabla\phi(x,t)}{\left\|\nabla\phi(x,t)\right\|} \qquad \text{For } x \,/\quad \phi(x,t) = 0$$

- Total derivation with t :

$$\frac{d\phi(x,t)}{dt} = \frac{d\phi(C(u,t),t)}{dt} = \nabla\phi \cdot \frac{\partial C}{\partial t} + \frac{\partial\phi}{\partial t} = 0$$

Evolution of level sets

- We only consider an evolution along the normal direction :
- Fundamental Equation : $$\frac{\partial \mathbf{C}(u,t)}{\partial t} = \beta(u,t)\,\mathbf{n}(u)$$

$$\boxed{\frac{\partial \phi(x,t)}{\partial t} = -\beta \left\| \nabla \phi \right\|}$$

<u>For all x</u>

Evolution of level sets

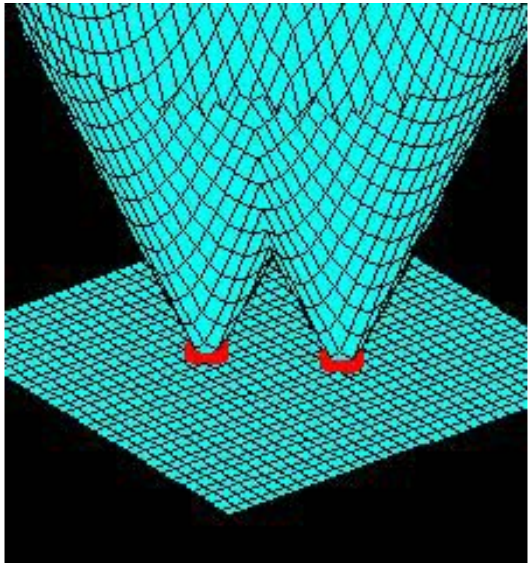
- No need for parameterisation :
  - Deformation invariant with change of parameterisation

No need to handle the number of points and their spacing along the contour

Easy and stable computation of intrinsic values (curvature)

Advantages of level sets (2)

- Allow to handle topological changes



Source : Fast Marching Methods and Level Set Methods, J.A. Sethian

Evolution of level sets

- Computationally expensive since $\phi(u,t)$ is 2D / 3D whereas contour /surface is 1D / 2D

Use of a narrow band around the contour

Stability and convergence issues linked to the narrow band (reinitialisation)

Spatial and temporal evolution

- Need to define $\beta(x, t)$ for LS evolution
- Temporal Discretisation :
  - Explicit Scheme :

$$\frac{\partial \phi(x,t)}{\partial t} \Rightarrow \frac{\phi^{t+1} - \phi^{t}}{\Delta t}$$

- Spatial discretisation :
  - Regular Grid (image)
  - Use centered finite differences except for « advection» term («upwind » scheme)

Example

- β=1-εk *:* combinaison of <u>hyperbolic</u> (β=1) with <u>parabolic</u> (β=- εk) terms

$$\frac{\partial \phi(x,t)}{\partial t} = \underbrace{-\|\nabla \phi\|}_{} + \underbrace{\varepsilon k \|\nabla \phi\|}_{}$$

Advection :
Use of an « upwind » scheme to discretize
$\|\nabla \phi\|$

Regularisation Term :
Use of finite differences to discretize k

Curvature Discretization

- Use Hessian and first derivative discretized with finite differences :

$$k = -\frac{\nabla\phi^T H \nabla\phi}{\left\|\nabla\phi\right\|^2} = -\frac{\phi_{xx}\phi_y^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}\phi_x^2}{\left(\phi_x^2 + \phi_y^2\right)^{3/2}}$$

Application to image segmentation (1)

- Several propagation terms:
  - $$\beta(C,x) = c(x)\left(k + \beta_0\right)$$

With $$c(x) = \frac{1}{1 + \left\|\nabla\left(G_\sigma(x) * I(x)\right)\right\|}$$

<u>Interpretation</u> : Contour propagates until it reaches
Voxels with large intensity gradient

Application to image segmentation (2)

- **Geodesic Active Contours**

$$\beta(C,x) = h(x)k - \nabla h \cdot \frac{\nabla \phi}{\|\nabla \phi\|}$$

With $$h(x) = \frac{1}{1 + \left\|\nabla\left(G_\sigma(x) * I(x)\right)\right\|^2}$$

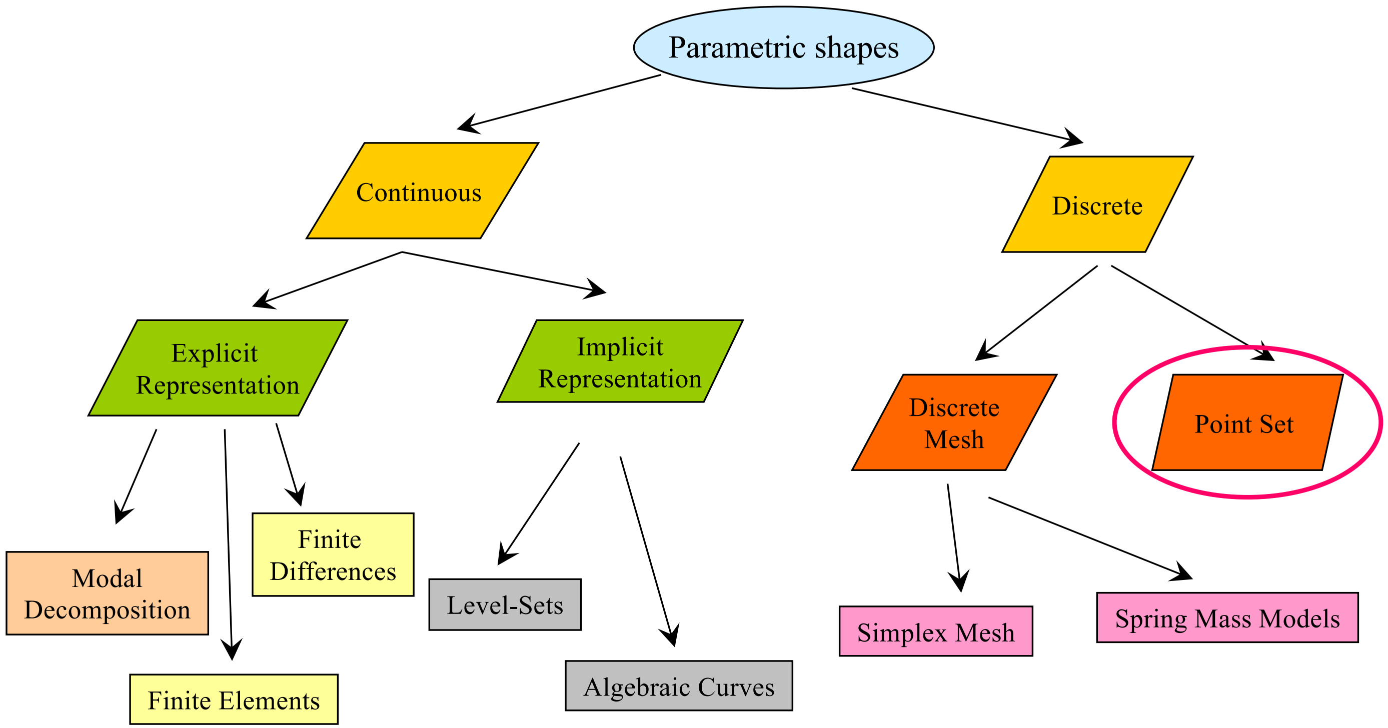
<u>Interpretation</u> : Minimizing the geodesic distance
Of a contour in a metric Riemanian space govern by metric h(x)

$$L^* = \int_a^b h\left(C(u)\right) \left\| \frac{d\mathbf{C}(u)}{du} \right\| du$$

4. Connexity and Shape Constrained Image segmentation

- 4.1 Label Connexity Hypothesis : Markov Random Field
- 4.2 Introduction to shape and deformable Models
- 4.3 Snakes algorithm
- 4.4 Level Set Algorithm
- **4.5 Point Distribution Model**
- 4.6 Multi-atlas Algorithm

Explicit vs Implicit
Shape representation



Point Distribution Model (PDM)

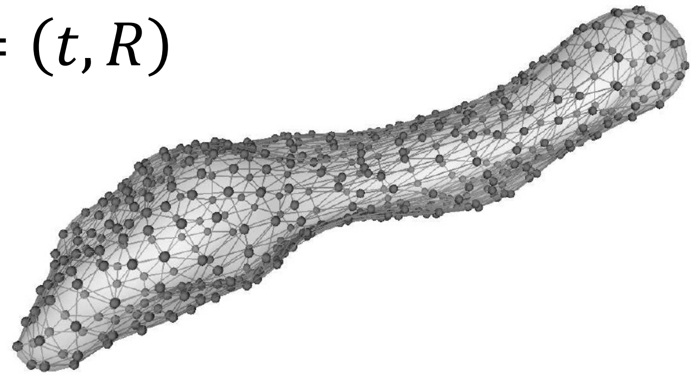
- Shape defined as a set of P points in $\mathbb{R}^d$ d=2 or 3

$$X = (x_1, , \ldots, x_P\ )^T \in \mathbb{R}^{3P} \qquad x_i = \begin{bmatrix} x_i^1 \\ x_i^2 \\ x_i^3 \end{bmatrix}$$

- Shape space is defined as Gaussian distributions :

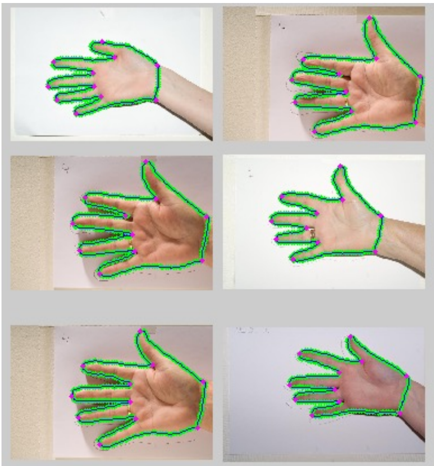
$$p(X) = \mathcal{N}(X; \mu^*, \Sigma^*)$$

  – shape preserving group is rigid transform $T = (t, R)$



- How to define μ^* and Σ^*?

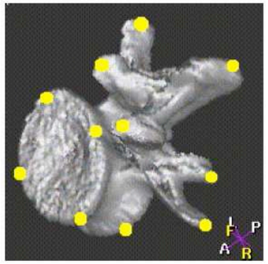
Input as collection of homologous point sets

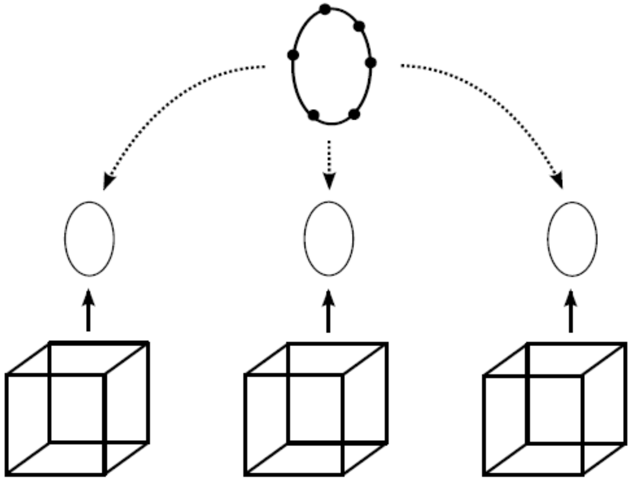


- Use supervision :
  - training set $\hat{X}$ of N sample shapes $\hat{X} = \{X_n\}\ 1 \leq n \leq N$
- Constraint :
  - All input shapes have the same number of P points
  - All points are homologous
- Create an allowed shape space based on collection :
  - Create All shapes
  - Register all input shapes rigidly
  - Create Mean shape $\bar{X} = \frac{1}{N}\sum_{n=1}^{N} X_n$
  - Estimate variability with the sample Covariance Matrix : $\Sigma = \frac{1}{N}\sum_{n=1}^{N}(X_n - \bar{X})(X_n - \bar{X})^T$

How to create input homologous point set ?

- Finding Point Correspondence between shapes in the training set is difficult :
  - Can be done manually for simple shapes
  - Can use template registration
  - Optimize point correspondence using minimum description length





Definition of shape space

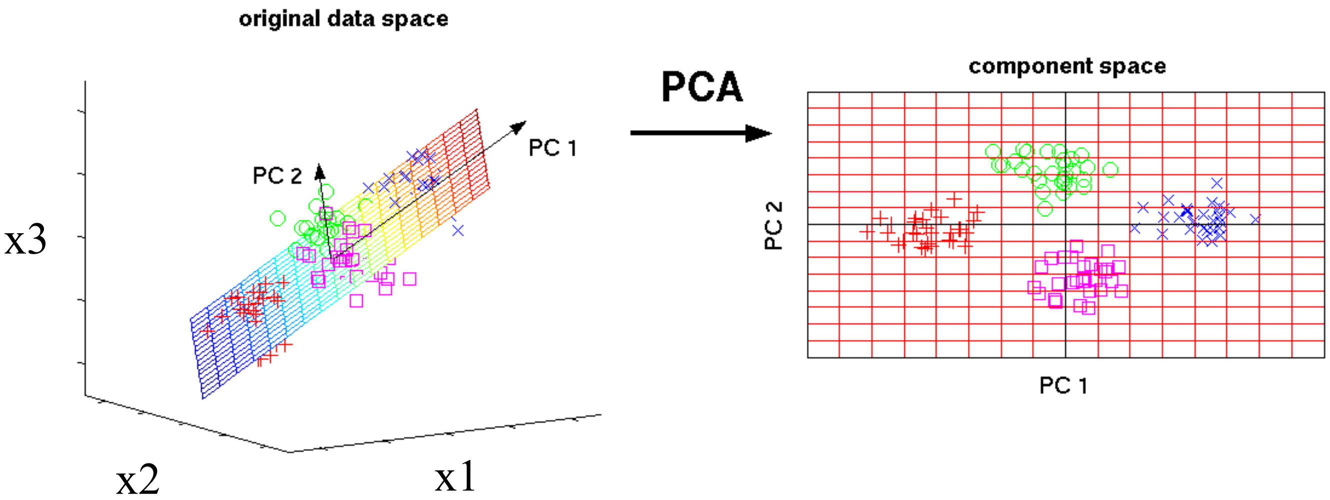
- Use sample mean $\bar{X}$ and sample covariance Σ for the Gaussian shape space ? $\mu^* = \bar{X}$ & $\Sigma^* = \Sigma$?
  - Not a good idea :
    - Size of training set is often much smaller than the dimension of X :
    - Noise may be present in $\bar{X}, \Sigma$
    - Covariance matrix may not be invertible

- Alternative : use **principal component analysis** to use low rank (rank M) representation of covariance matrix

Principal Component Analysis

- PCA solves 3 equivalent problems :
  - Pb 1 : Find a subset of M orthogonal directions u for which the projected variance $u^T \Sigma u$ is maximum

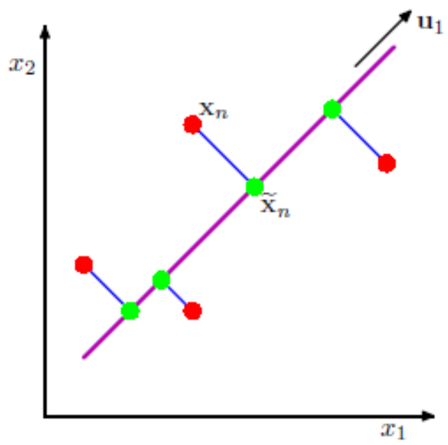
➡ Find eigenvectors of Σ associated with maximal eigenvalues



Principal Component Analysis

- PCA solves 3 equivalent problems :
  - Pb 2 : Find a set of M orthogonal directions u which minimize the average projection cost

Source : C. Bishop

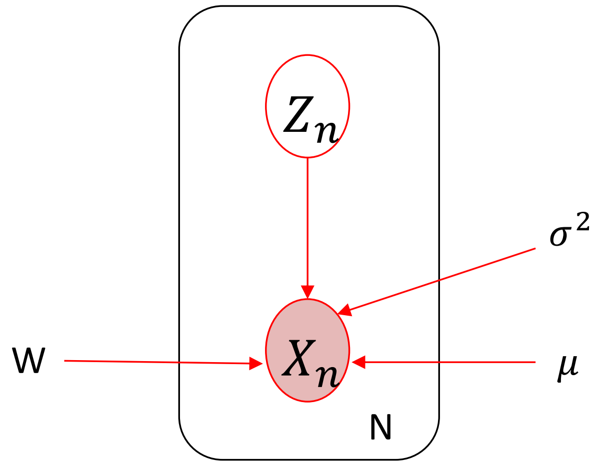


Probabilistic Principal Component Analysis

- Pb 3 : PPCA solves 3 equivalent problems :
  - X considered as an observed random variable (PPCA)
  - Existence of random latent variable Z of dimension M with $p(Z) = \mathcal{N}(0, I)$
  - X assumed to be generated by latent variable Z:

$$p(X|Z) = \mathcal{N}(X; WZ + \mu, \sigma^2 I)$$

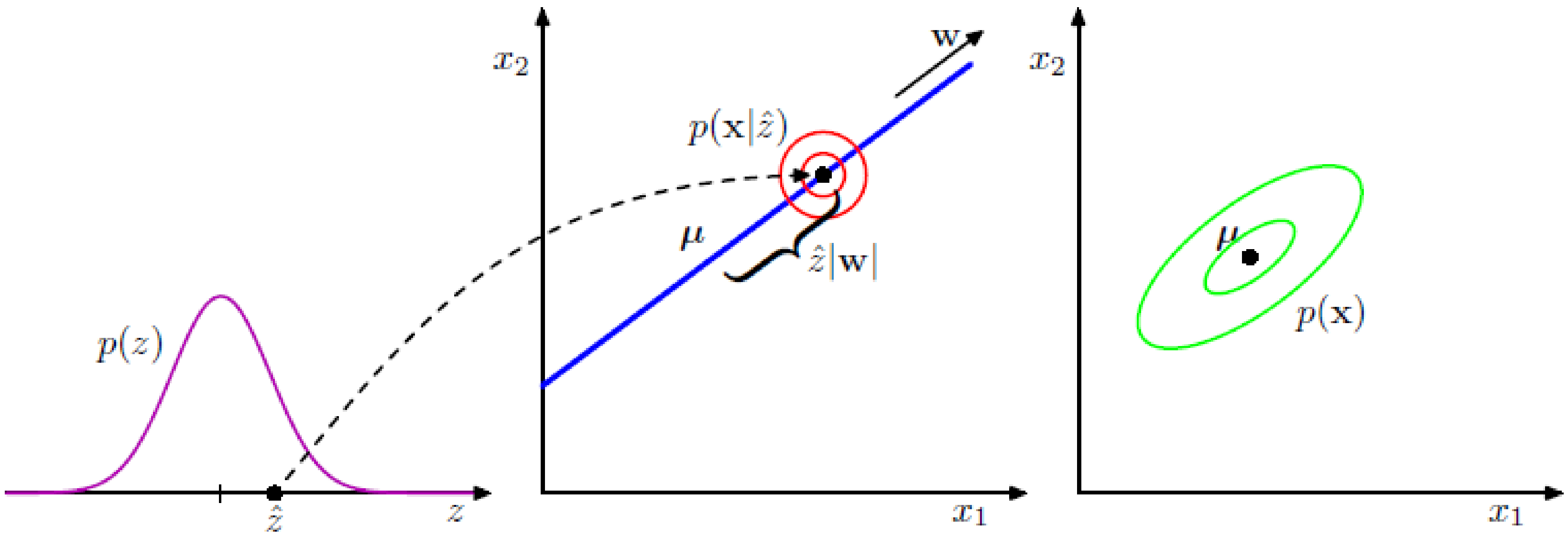
  - Likelihood Parameters :
    - W matrix $N \times M$
    - Mean value μ
    - Variance noise σ^2



Source : C. Bishop

Probabilistic Principal Component Analysis

– Equivalent to write $X \approx WZ + \mu + \epsilon$ where
  - $Z = \mathcal{N}(0, I)$ is a Gaussian random variable of 0 mean and dimension M.
  - $\epsilon = \mathcal{N}(0, \sigma^2 I)$ is a Gaussian random variable of 0 mean and dimension D



Probabilistic Principal Component Analysis

- Inference :
  - Marginal likelihood is also Gaussian as the product of 2 Gaussian distributions

$$p(X_n) = \int_{\mathbb{R}^M} p(X_n|Z_n)p(Z_n)dZ_n = \mathcal{N}(X_n|\mu, WW^T + \sigma^2 I)$$

  - Maximize log marginal likelihood $\log p(\hat{X}) = \sum_n^N \log p(X_n)$
    - Closed form solution :
      - $\mu = \bar{X}$
      - Sample Covariance matrix $\Sigma = U\Lambda U^T$
      - M largest eigenvalues: $\lambda_m, 1 \leq m \leq M$
      - Eigenvectors associated w_m with largest eigenvalues λ_m
      - $W = \omega_M (\Lambda_m - \sigma^2 I)^{\frac{1}{2}} R$

$$\Sigma = \frac{1}{N}\sum_{n=1}^{N}(X_n - \bar{X})(X_n - \bar{X})^T$$

Matrix NxM of Eigenvectors w_m

Diagonal Matrix MxM of Eigenvalues λ_m

Orthogonal Matrix MxM

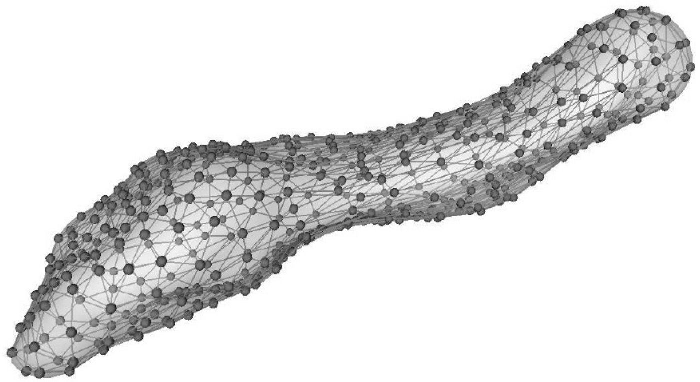
Point Distribution Model Shape Space

- Define shape space from $\bar{X}, W\ , \sigma^2$ as:

$$p(X) = \mathcal{N}(X; \bar{X}, WW^T + \sigma^2 I)$$

- Accounting for any rigid transformation:
  - Rotation R and translation t

$$p(X) = \mathcal{N}(X; R\bar{X} + t, RWW^T R^T + \sigma^2 I)$$

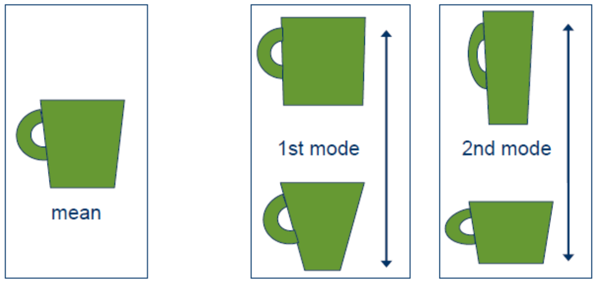


Fitting a PDM

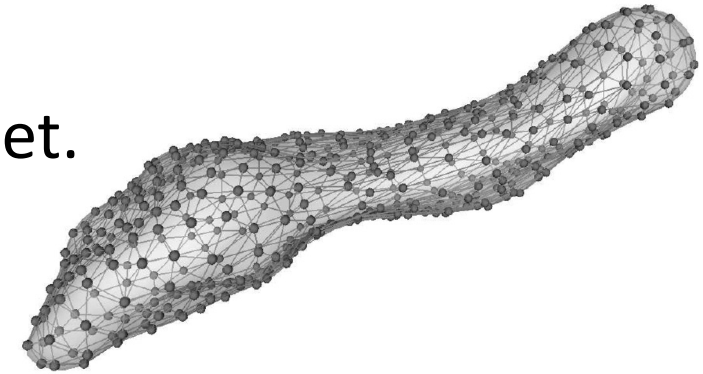
- Let Y be a set of points representing an instance of the structure
- How to project Y on the allowable shape space ?
- 3 steps :
  - Align with template
  - Project on shape space
  - Realign the projection

Restricted Shape Space

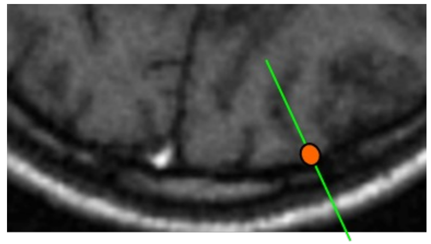
- Align Current Shape Y with mean template :
  - Find the rotation R & translation t which minimizes $||\bar{X} - RY - t||^2$ (closed form solution)
  - Center data $Y' = RY + t$
- Project centered data
  - $\phi_m = w_m^T(Y' - \bar{X})$ for $1 \leq m \leq M$
  - Bound projection : if $|\phi_m| < 3\lambda_m$ then $\psi_m = \phi_m$ else $\psi_m = 3\lambda_m sign(\phi_m)$
- Reconstruct data
  - $\hat{Y} = R^{-1}(\bar{X} + \sum_m \psi_m w_m - t)$



Active Shape Model in Medical Imaging

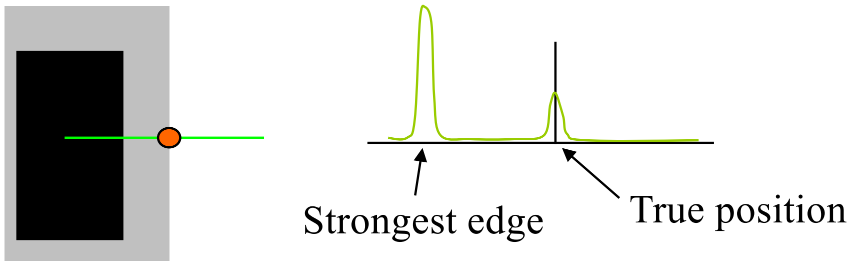


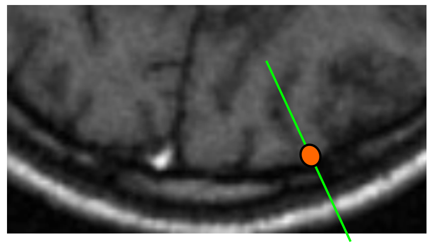
- Construct PDM shape space from training set.
- Iterate :
  - Estimate normal vector n_i at each vertex x_i
  - Find displacement s_i along normal n_i which minimizes local energy $s_i = \arg\min_s E_i(I, s)$
  - Update position $x_i \rightarrow x_i + s_i n_i$
  - Project current shape X on restricted shape space



Profile Models

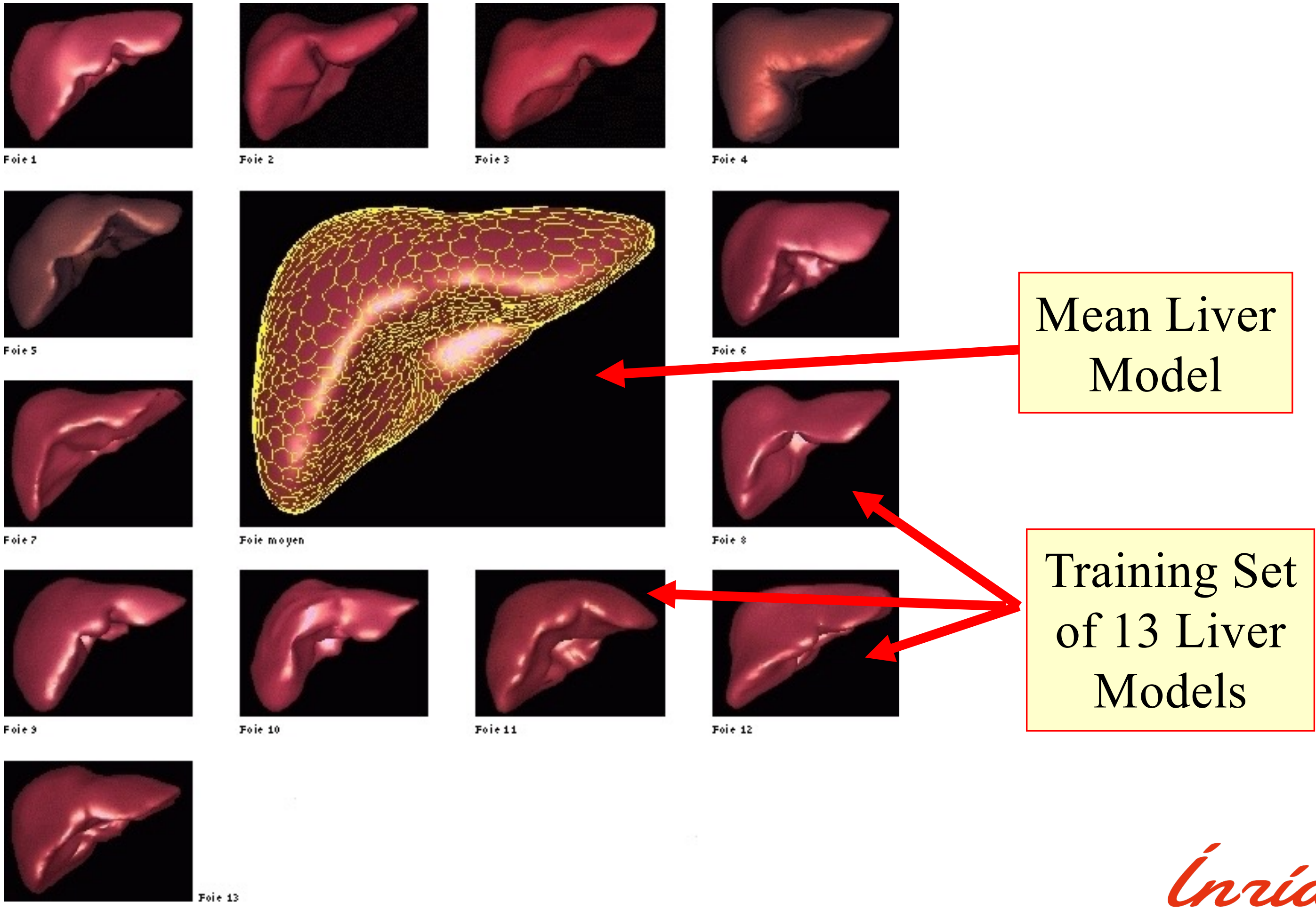
- Sometimes true point not on strongest edge





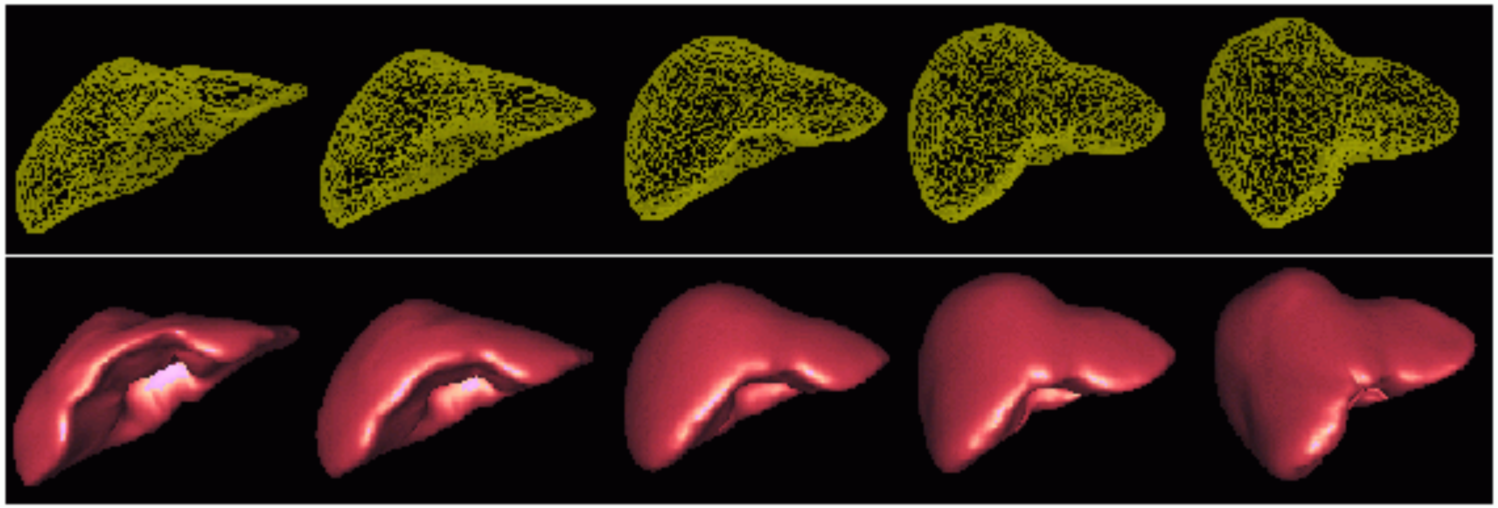
- Model local structure to help locate the point

Statistical Shape Model Of the Liver



Statistical Shape Model of the Liver

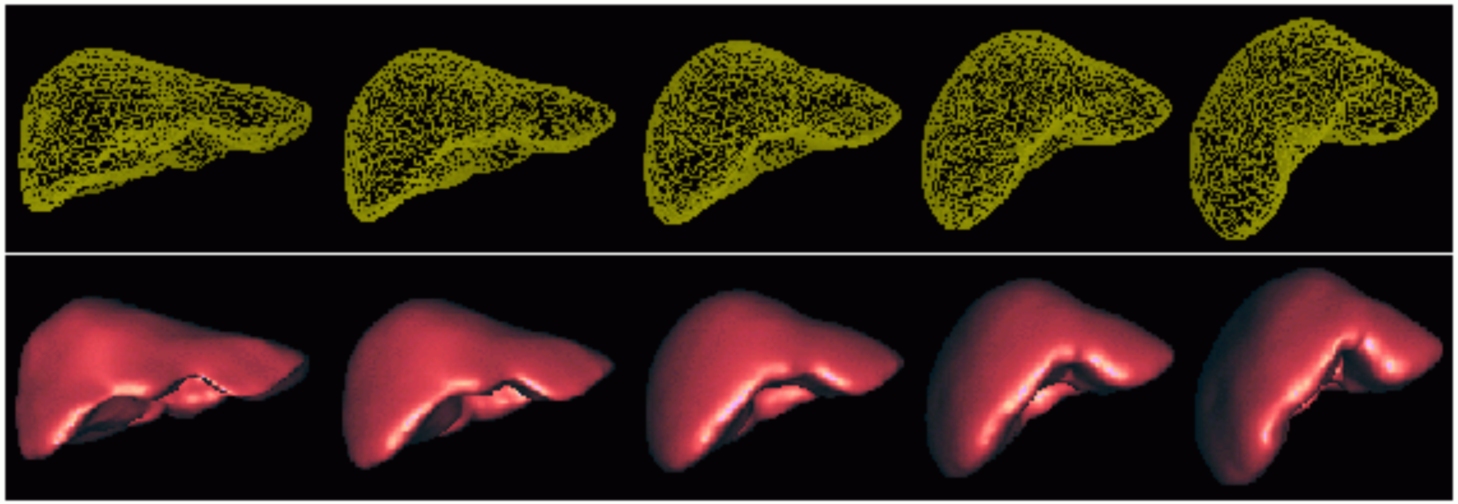
- Modes Of Variation



First Mode of Variation

Statistical Shape Model of the Liver

- Modes Of Variation

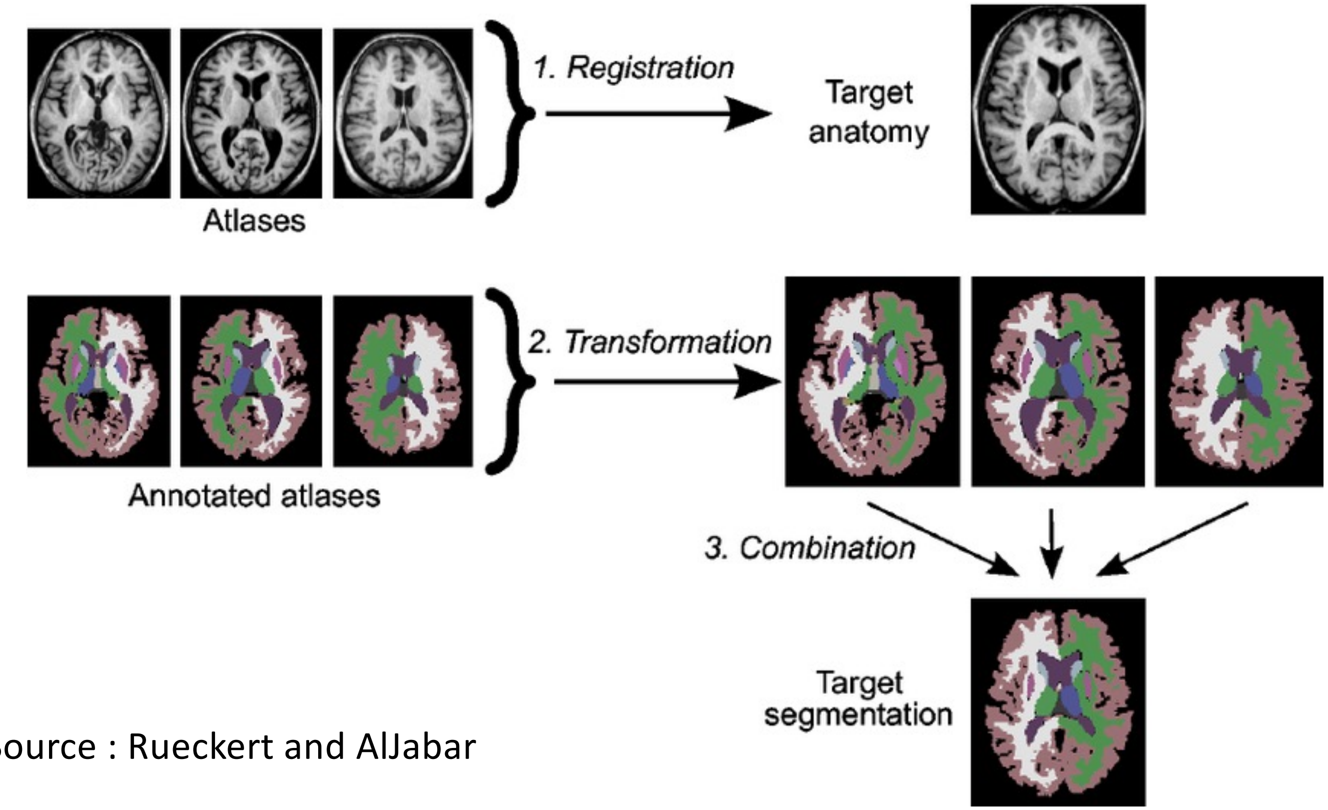


Second Mode of Variation

4. Connexity and Shape Constrained Image segmentation

- 4.1 Label Connexity Hypothesis : Markov Random Field
- 4.2 Introduction to shape and deformable Models
- 4.3 Snakes algorithm
- 4.4 Level Set Algorithm
- 4.5 Point Distribution Model
- **4.6 Multi-atlas Algorithm**

Use annotated images to segment new image



Source : Rueckert and AlJabar

Use annotated images to segment new image

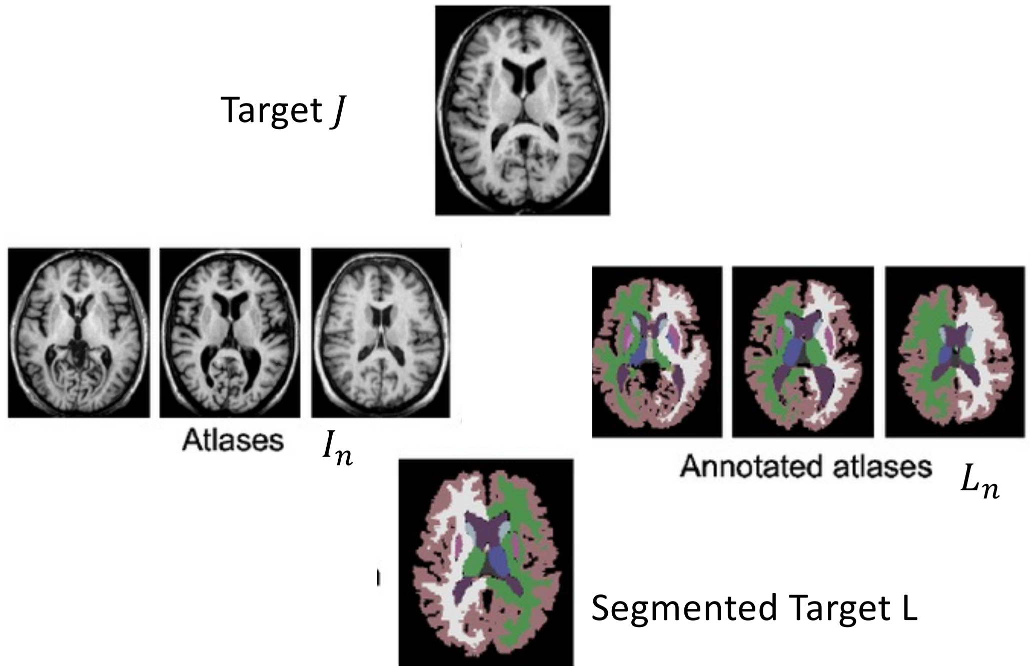
- Input data :
  - Set of representative images with their segmentation
- For all images in input set :
  - Register non-rigidly input image on target image
  - Apply deformation on corresponding label image
- Combine deformed label image into a classification or segmentation of target anatomy

Key steps

- Atlas selection :
  - Use only input images that are closest from the target images (metrics definition)
- Combination of registered labels
  - Majority voting
  - Weighted majority voting
  - STAPLE algorithm

Definitions, Notations

- Input :
  – Target image J
  – Atlas images I_n, $1 \leq n \leq N$
  – Segmented atlas images, L_n
- Output :
  – Segmented target image L



- Image patch : small rectangular piece of an image
- Define Image patch extraction operator S() : S(J,x) is an image path centered on x from image J

Membership function

- **<u>Hypothesis</u>** : every patch $S(J, x)$ originates from a patch $S(I_n, y)$ from the training database.
- Define membership function M :

$$M(x) = (n, y)$$

spatial position
In input image

atlas id
(training database)

spatial position
In atlas image

Probabilistic Framework for multi-atlas segmentation

- Knowing *M(x)* , probabilities of observing intensity *I(x)* and label *L(x)* are conditionally independent.
- $p(J(x)|M, I)$ often a Gaussian on mean $J(x) - I_n(y)$
- $p(L|M, L)$ often related to distance of x to the border of the structure
- $p(M)$ is a Potts model to enforce that neighboring voxels have similar memberships