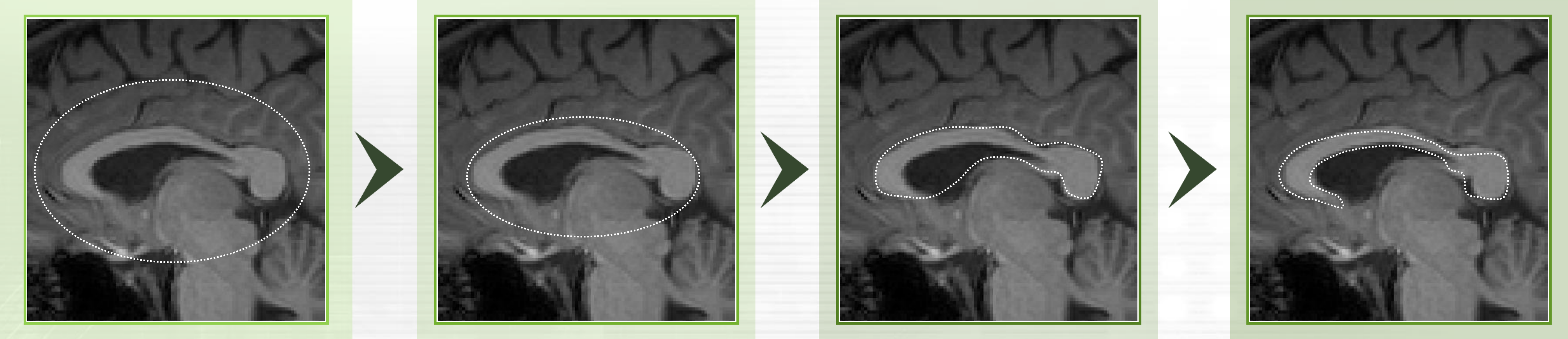
Active Contour Model

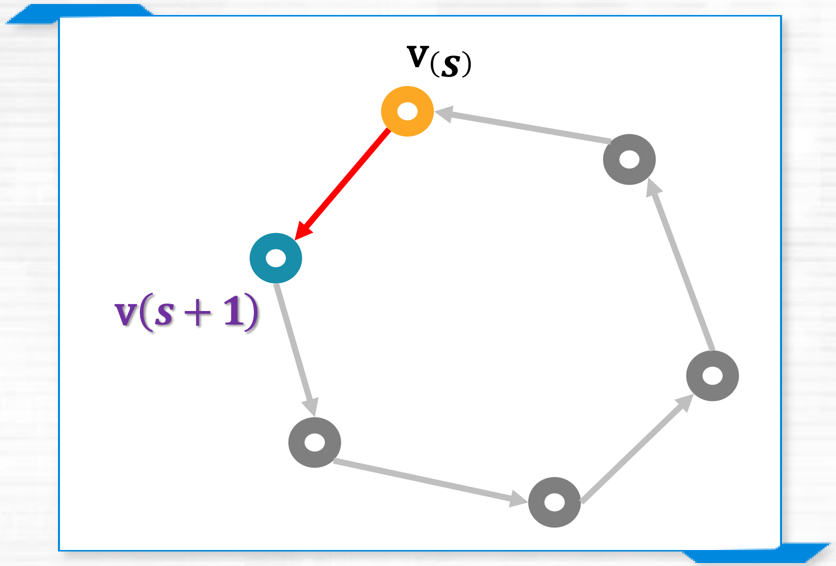
Active Contour Model

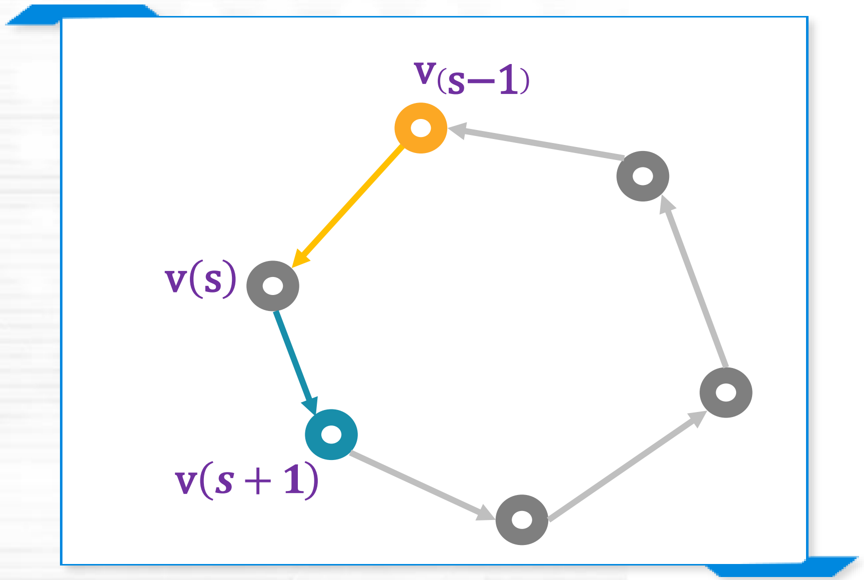


출처 http://adni.loni.usc.edu/data-samples/

External Energy

Internal Energy





Energy Equation

$$\mathbf{E}^{*}_{snake} = \int_s \mathbf{E}_{snake}(\mathbf{v}(s))ds$$

$$= \int_s \boxed{\mathbf{E}_{int}(\mathbf{v}(s))} + \boxed{\mathbf{E}_{ext}(\mathbf{v}(s))} \; ds$$

$$\boxed{\mathbf{E}_{elastic} + \mathbf{E}_{bending}} \qquad \boxed{\mathbf{E}_{line} + \mathbf{E}_{edge}}$$

$$\mathrm{E}_{elastic} = \frac{1}{2}\alpha(s)\,|v_s|^2$$

$$\mathbf{E}_{bending} = \frac{1}{2}\beta(s)\,|v_{ss}|^2$$

$$\mathrm{E}_{\mathrm{line}} = \mathrm{I}(x,y)$$

$$\mathrm{E}_{\mathrm{edge}} = -\,|\nabla\,\mathrm{I}(x,y)|^2$$

Energy Equation

$$\mathbf{E}^{*}_{snake} = \int_{s} \mathbf{E}_{snake}\left(\mathbf{v}(s)\right)ds$$

$$= \int_{s} \boxed{\mathbf{E}_{int}\left(\mathbf{v}(s)\right)} + \boxed{\mathbf{E}_{ext}\left(\mathbf{v}(s)\right)} \; ds$$

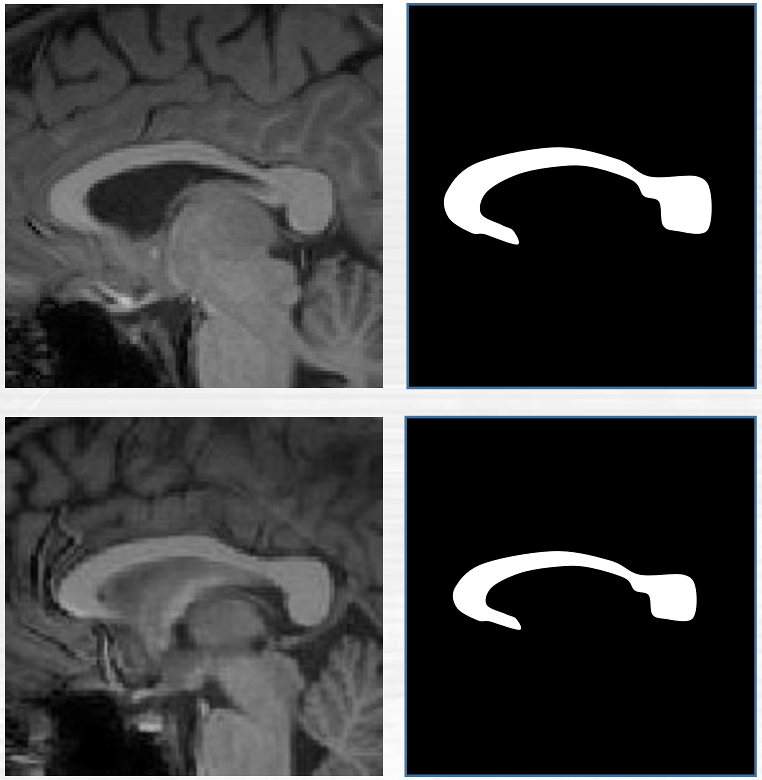
$$\mathbf{E}_{int}: \; \mathbf{E}_{\boldsymbol{elastic}} + \mathbf{E}_{\boldsymbol{bending}}$$

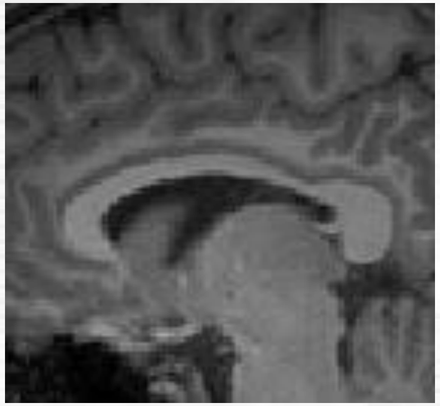
$$\mathbf{E}_{ext}: \; \mathbf{E}_{\boldsymbol{line}} + \mathbf{E}_{\boldsymbol{edge}}$$

- Optimization using numerical method

Learning based Method

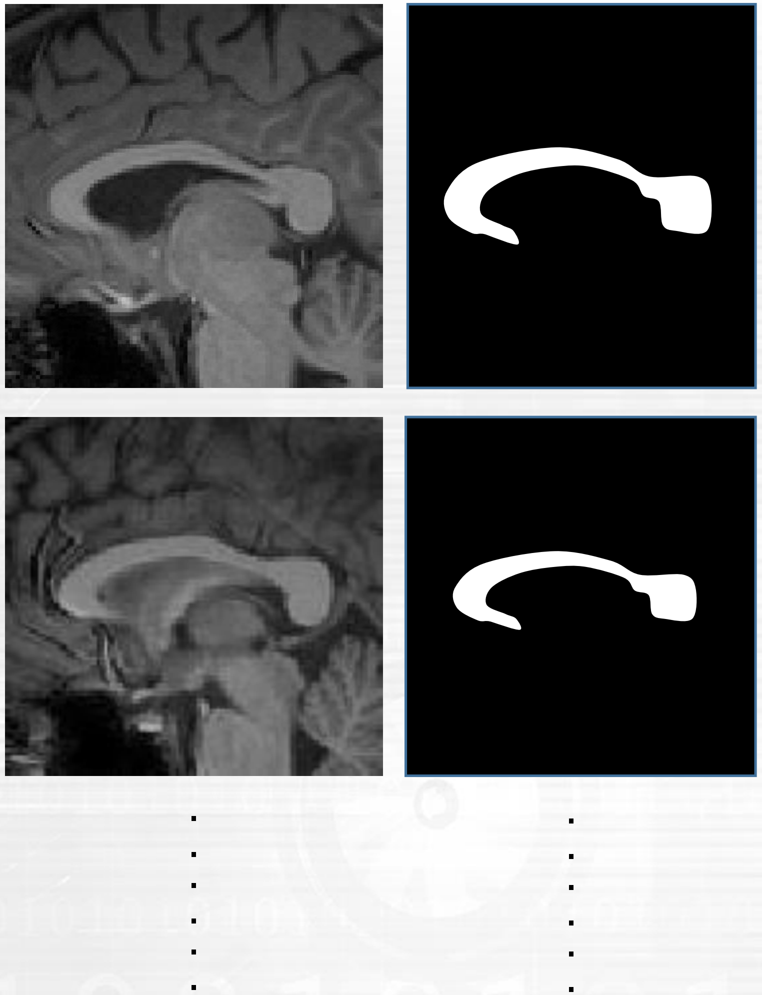
Atlas-based Label Fusion

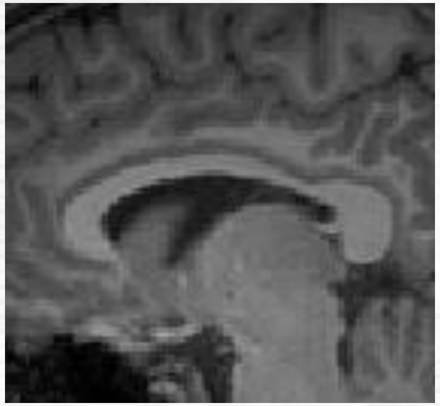




출처 http://adni.loni.usc.edu/data-samples/

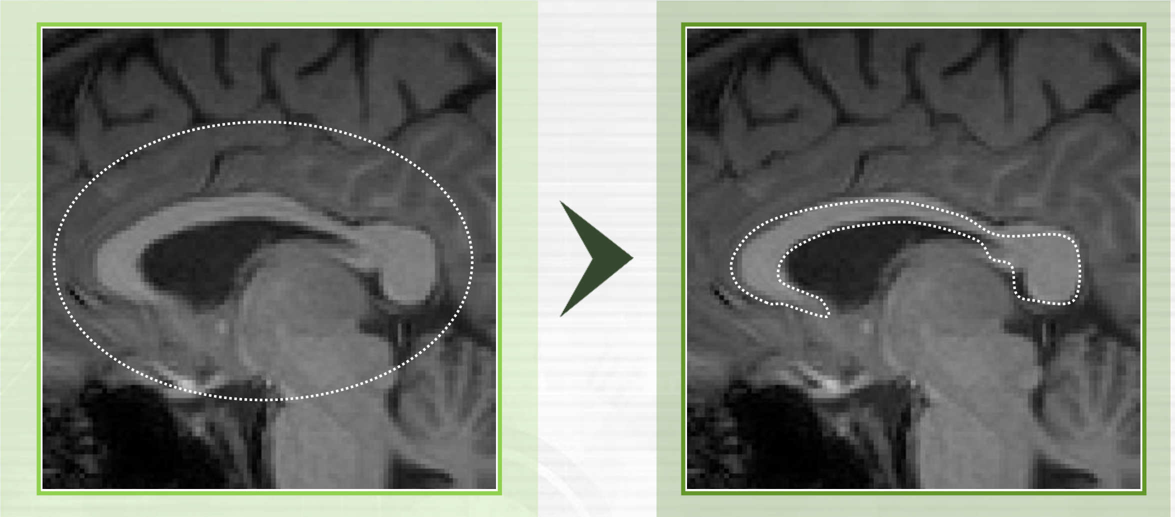
Patch-based Label Fusion

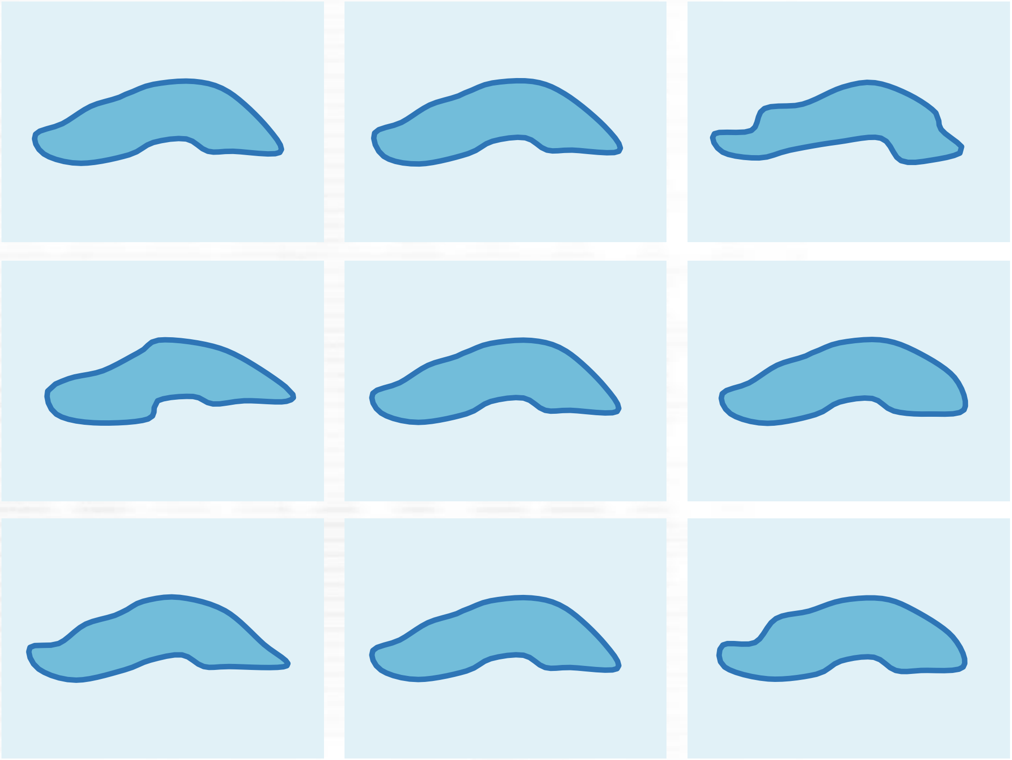




Active Shape Model

Active contour model





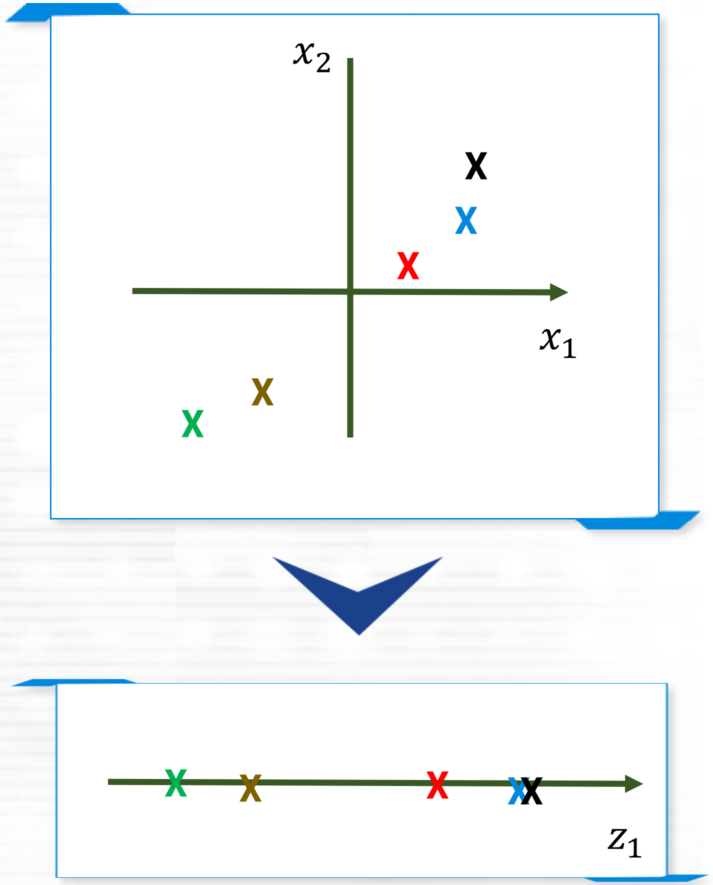
출처 http://adni.loni.usc.edu/data-samples/

Dimensionality Reduction

- Data compression

  - Reduce data from 2D to 1D

$$x^{(1)} \rightarrow z^{(1)}$$
$$x^{(2)} \rightarrow z^{(2)}$$
$$\vdots$$
$$x^{(m)} \rightarrow z^{(m)}$$



Principal Component Analysis (PCA)

$$Var[X\vec{e}] = \vec{e}^T C \vec{e}$$

$$Var[X\vec{e}] = \frac{1}{m-1}\sum_{i=1}^{m}[X\vec{e} - E(X\vec{e})]^2 = \frac{1}{m-1}\sum_{i=1}^{m}[X\vec{e} - E(X)\vec{e}]^2, \qquad (E(X) = 0)$$

$$= \frac{1}{m-1}\sum_{i=1}^{m}(X\vec{e})^2 \qquad = \frac{1}{m-1}(X\vec{e})^T(X\vec{e})$$

$$= \frac{1}{m-1}\vec{e}^T X^T X\vec{e} \qquad = \vec{e}^T\left(\frac{X^T X}{m-1}\right)\vec{e}, \qquad \left(\frac{X^T X}{m-1} = C\right)$$

$$= \vec{e}^T C \vec{e}$$

Principal Component Analysis (PCA)

$$maximize \quad \vec{e}^T C \vec{e}$$
$$s.t \quad \|\vec{e}\|^2 = 1$$

Lagrange Multiplier Method

$$L(\vec{e},\lambda) = \vec{e}^T C \vec{e} - \lambda(\vec{e}^T \vec{e} - 1)$$

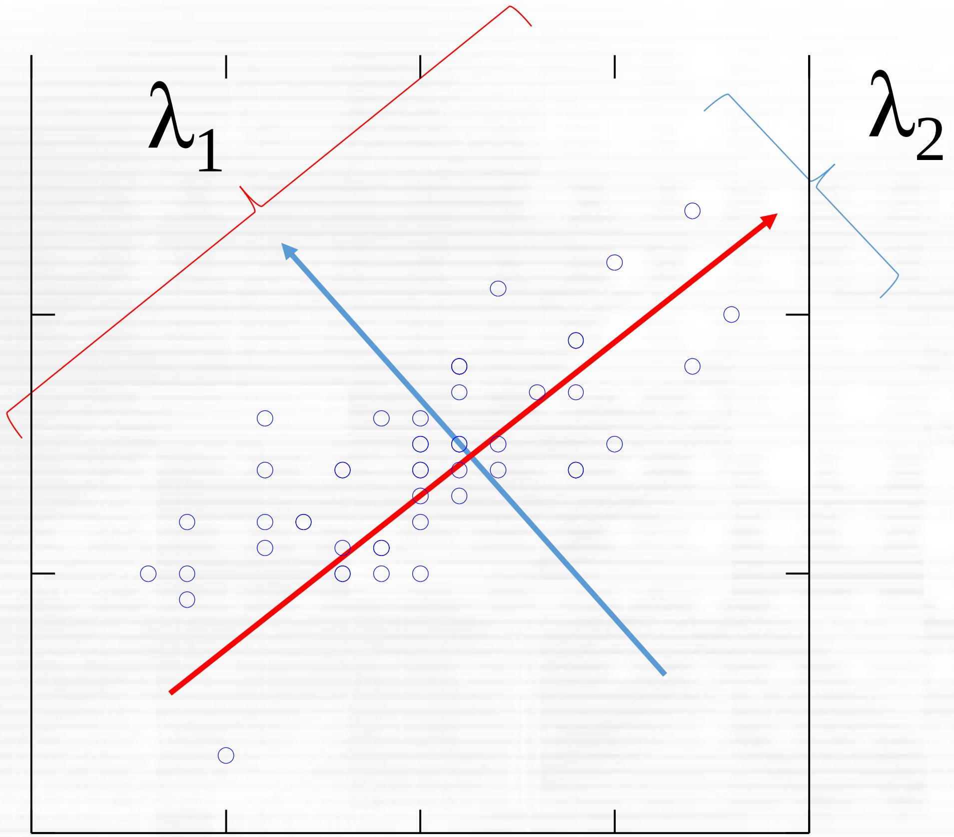
$$\frac{\partial L}{\partial \vec{e}} = (C + CT)\vec{e} - 2\lambda\,\vec{e}$$

$$= 2C\vec{e} - 2\lambda\,\vec{e} = 0$$

$$\therefore C\vec{e} = \lambda\,\vec{e}$$

$$\therefore C = \vec{e}\,\lambda\,\vec{e}$$

Principal Component Analysis (PCA)



PCA Implementation

- **Data Normalization**

  - Training set: $x^{(1)}, x^{(2)}, \dots \dots, x^{(m)}$
  - Preprocessing (feature scaling/mean normalization):

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$$

  - Replace each $x_j^{(i)}$ with $x_j - \mu_j$

PCA Implementation

- Compute “covariance matrix”:

$$C = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)})(x^{(i)})^T$$

PCA Implementation

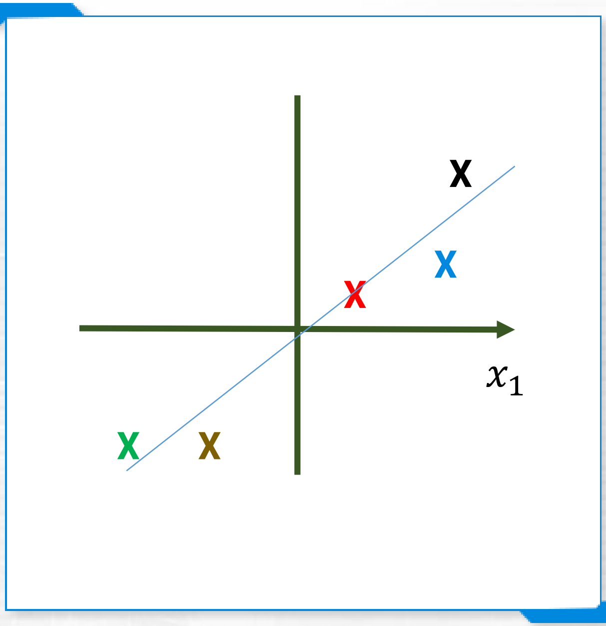
- Compute “eigenvectors” of matrix : [U,S,V] = svd(C);

$$U = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n}$$

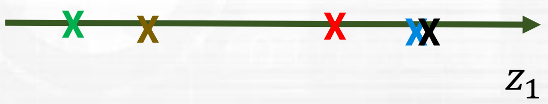
- Compute “Principal components”:

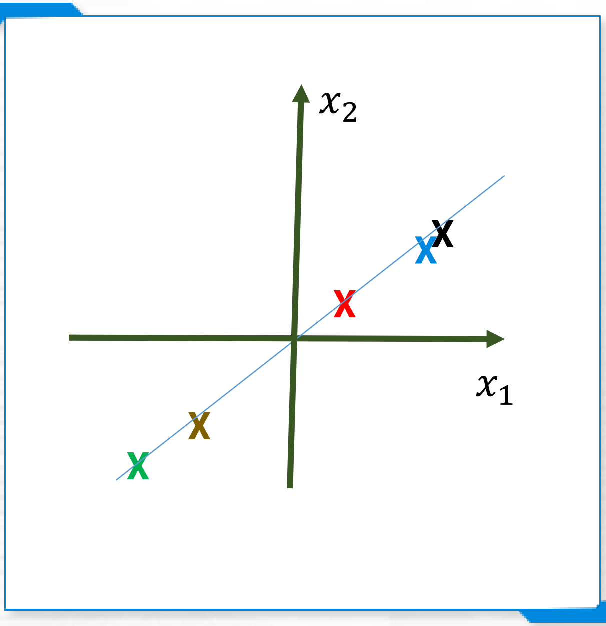
$$z = U_{reduce}^{T}\, x$$

PCA Reconstruction



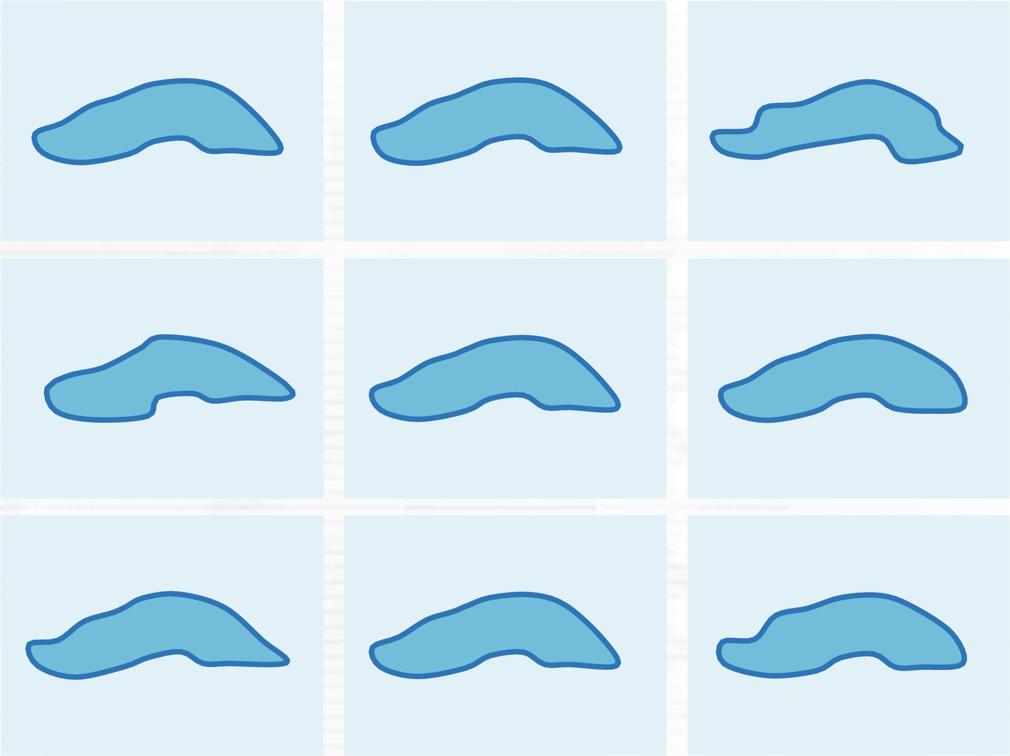
$z = U_{reduce}^{T}\ x$





$x_{approx} = U_{reduce} z$

Active Shape Model



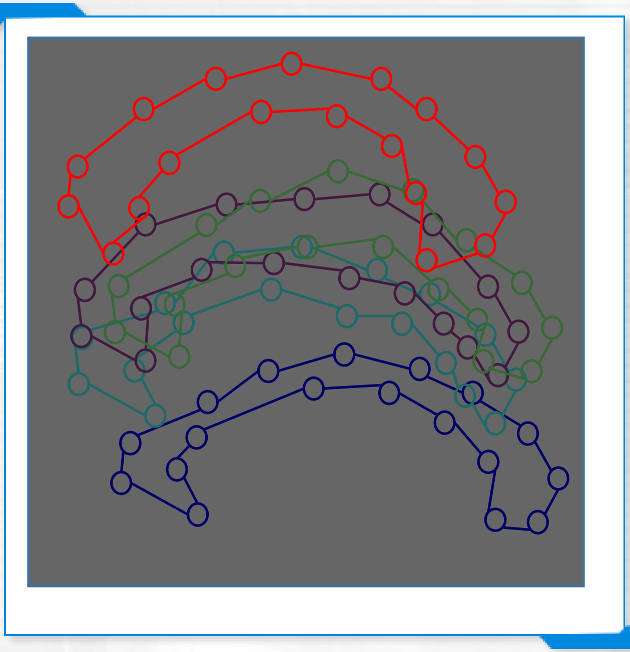
출처 http://adni.loni.usc.edu/data-samples/

PCA Implementation

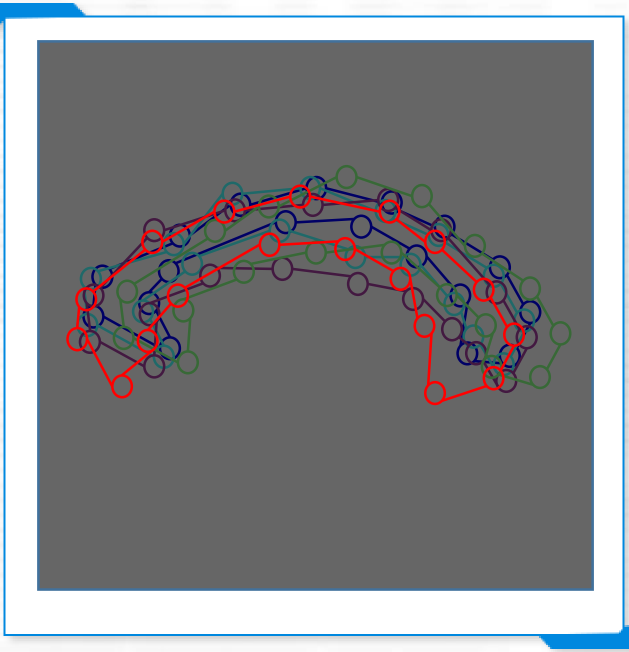
- Data normalization
- Compute “covariance matrix”
- Compute “eigenvectors” of matrix : [U,S,V] = svd(C)
- Compute “Principal components”

Model Construction

- Data normalization = Shape alignment



Before



After

Model Construction

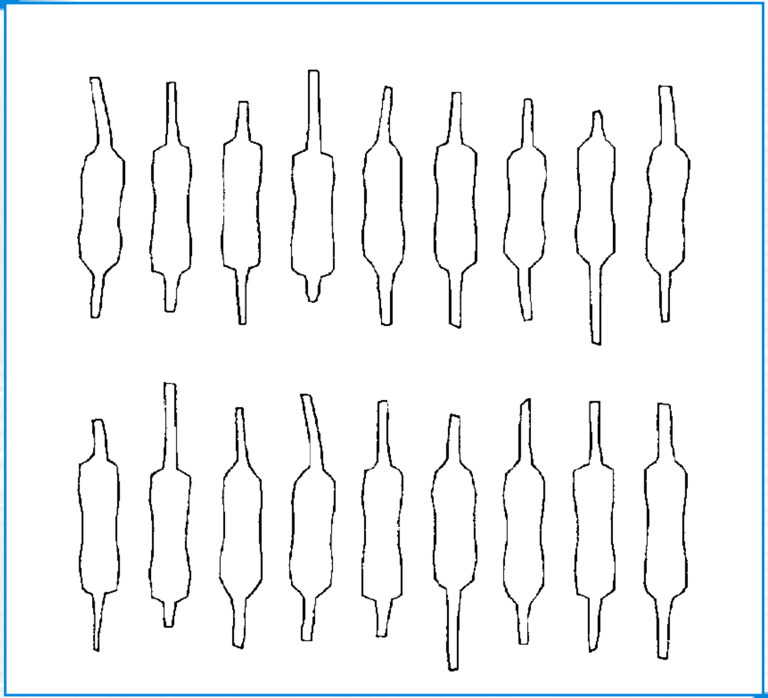
- $\bar{x} = \frac{1}{m}\sum_{i=1}^{m} x^i$
- $dx^i = x^i - \bar{x}$
- $C = \frac{1}{m}\sum_{i=1}^{m}(dxi)(dxi)^T$
- $C\vec{e} = \lambda\,\vec{e}$
- $U = [u^1 \ \ldots u^n]$
- $b^i = U_{reduce}^T(dx^i - \bar{x})$
- $dx^i = U_{reduce}b^i$
- $x^i = \bar{x} + U_{reduce}b^i$

Active Shape Model

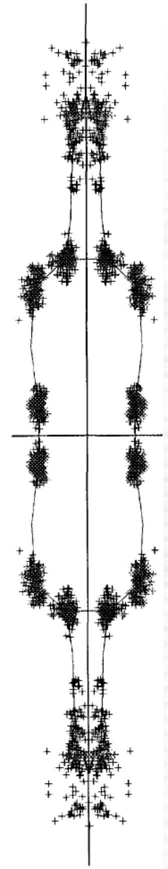
$$\boldsymbol{b} = U_{reduce}^{T}(\boldsymbol{y} - \overline{\boldsymbol{x}})$$

$$\boldsymbol{x} = \overline{\boldsymbol{x}} + U_{reduce}\boldsymbol{b}$$

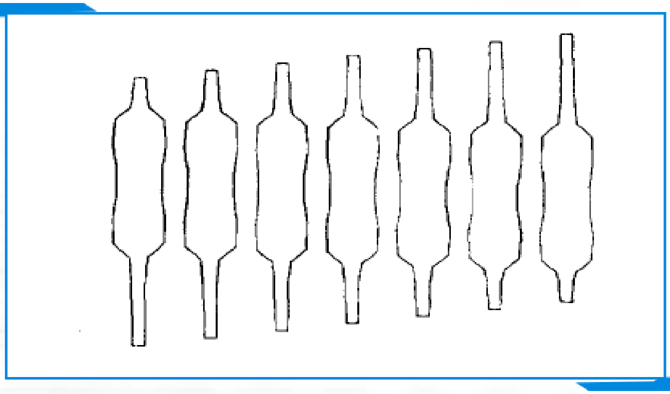
Active Shape Model



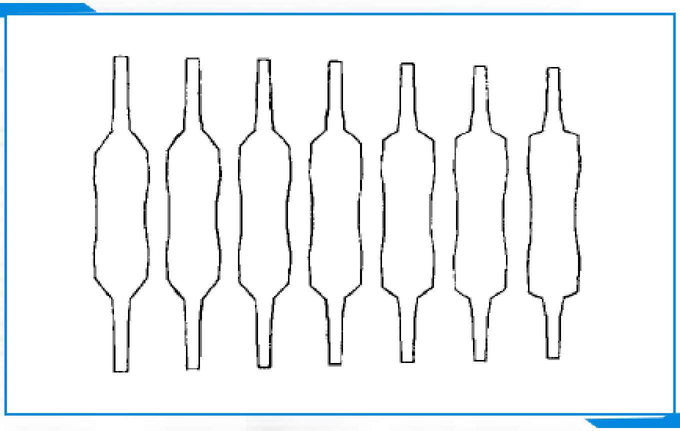
Training set



Statistical shape model



$-2\sqrt{\lambda_1} \leftarrow b_1 \rightarrow 2\sqrt{\lambda_1}$



$-2\sqrt{\lambda_2} \leftarrow b_2 \rightarrow 2\sqrt{\lambda_2}$

출처 *T.F. Cootes et al. "Active shape models - their training and application". Computer Vision and Image Understanding 1995*

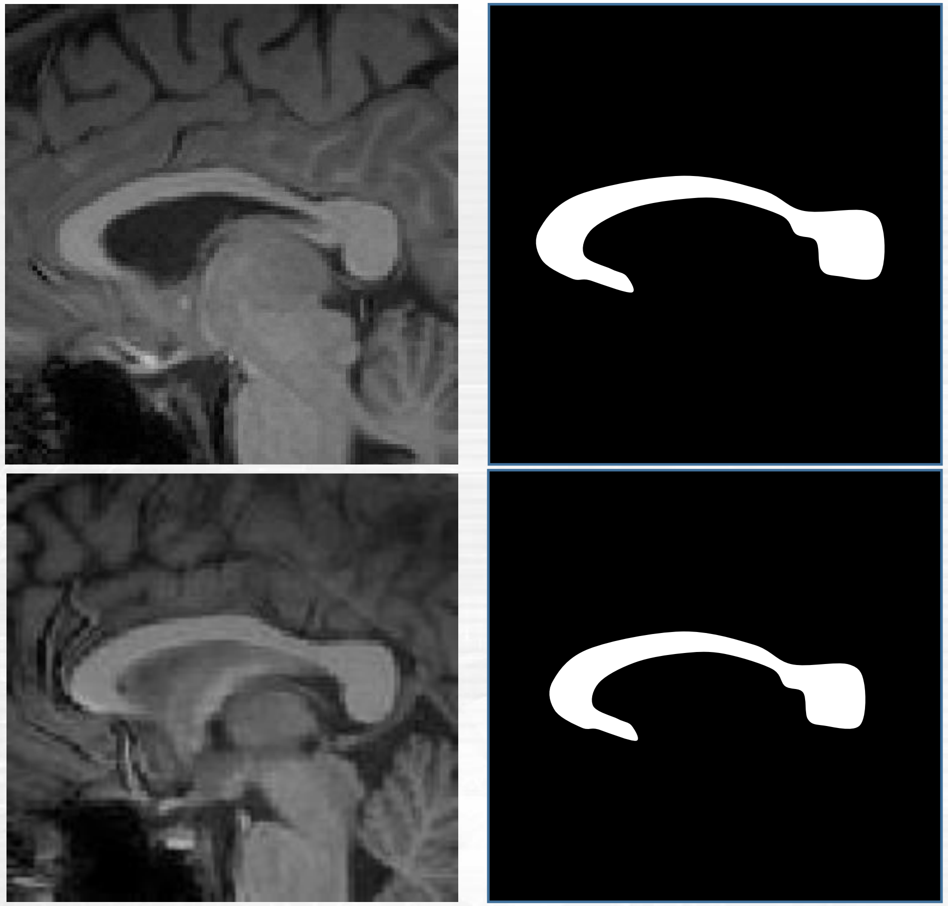
Segmentation based on Classification

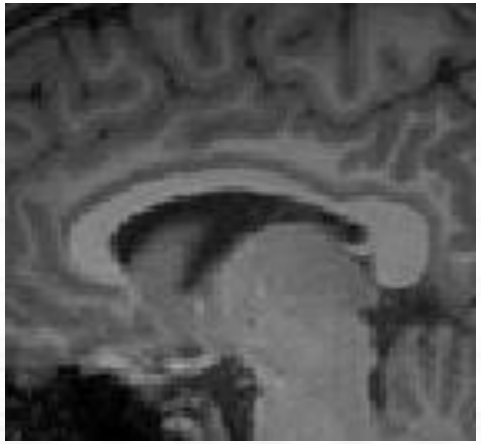
$$E(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N) = -\log P(x_1, x_2, \dots, x_N | z_1, z_2, \dots, z_N)$$

$$= -\log \prod_i P(z_i | x_i) \prod_{(i,j)} P(x_i, x_j)$$

$$= \underbrace{\sum_i \theta_i (z_i | x_i)}_{\text{Likelihood term}} + \underbrace{\sum_{(i,j)} \theta_{ij}(x_i, x_j)}_{\text{Prior term}}$$

Segmentation based on Classification





출처 http://adni.loni.usc.edu/data-samples/

Segmentation based on Classification