Image Science Research

Demetri Terzopoulos

UCLA Computer Science Dept.

Introduction

- Computer vision (analysis)
- Computer graphics (synthesis)

} Modeling

Talk Overview

1. Statistical models
2. Physics-based models

Statistical Models

Appearance-based approach

- The appearance of objects in images

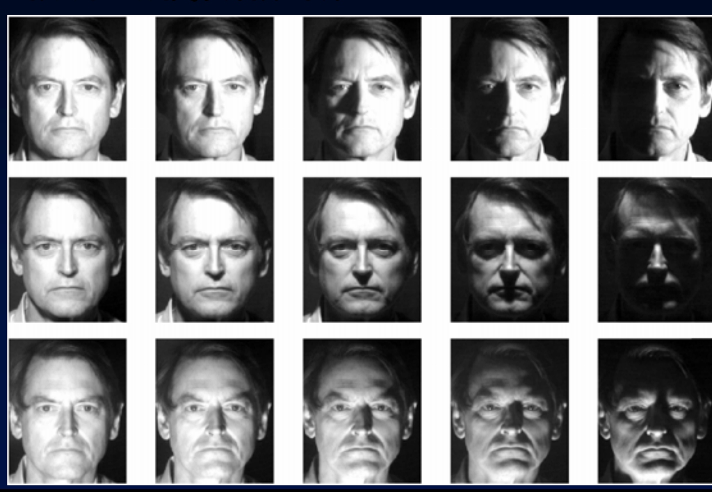
Objective

- A compact and descriptive representation for recognition

Turning on to tensor algebra !

Why is Face Recognition Difficult?

Variable illumination

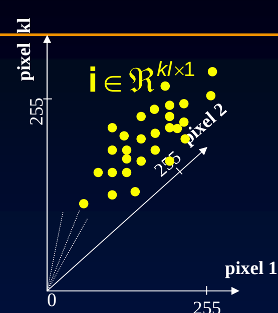


Why is Face Recognition Difficult?

Variable viewpoint

Images

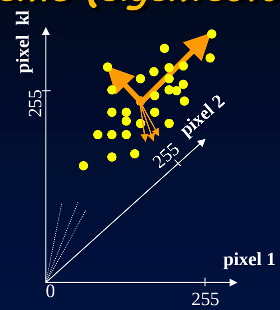
$I \in \Re^{k \times l}$



An image is a point in $\Re^{kl \times 1}$ dimensional space

Eigenimages

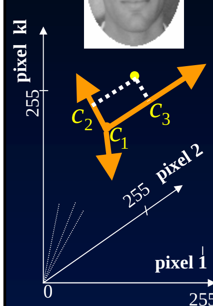
Principal components (eigenvectors) of image ensemble

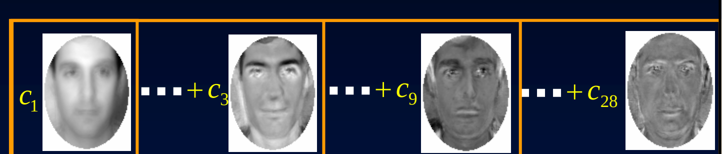


- Typically computed using the SVD algorithm

......

Linear Representation





Running Sum:

1 term 3 terms 9 terms 28 terms

Eigenfaces

- Facial images

- Eigenfaces basis vectors capture the variability in facial appearance

....

- Eigenfaces have been successful in simple facial recognition
  - *i.e., frontal images with fixed illumination*

The Problem with Linear (PCA) Appearance-Based Recognition Methods

Eigenimages work best for recognition when only a single factor – e.g., object identity – varies

- However, natural images result from ***multiple causal factors*** related to scene structure, illumination and imaging

......

- Distinguish between **observational** factor and **causal** factors

Our Approach

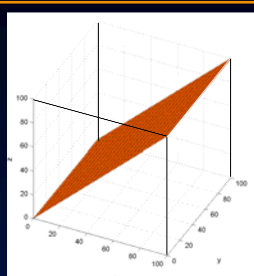
[Vasilescu & Terzopoulos, ECCV 02, ICPR 02, CVPR 03, CVPR 05, ICCV 07]

A nonlinear appearance-based technique

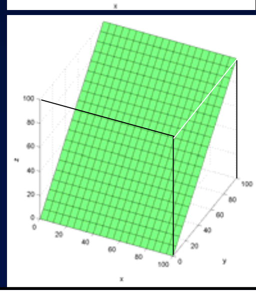
- Our appearance-based model ***explicitly accounts*** for each of the multiple causal factors inherent in image formation
- Multilinear algebra, the algebra of higher-order tensors
- Applied to facial images, we call our tensor technique "TensorFaces"

Linear versus Bilinear

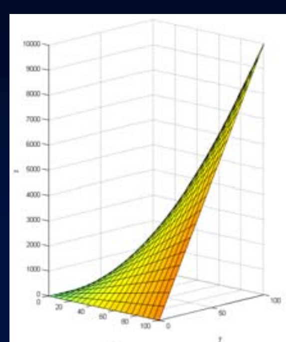
Z = X



Z = Y

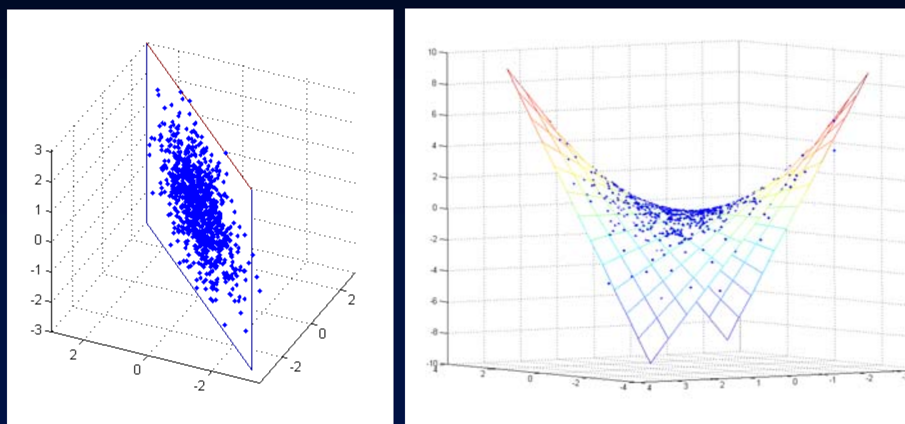


Z = XY



Linear vs Multilinear Manifolds

In low-dimensional space



PIE Database (Weizmann)

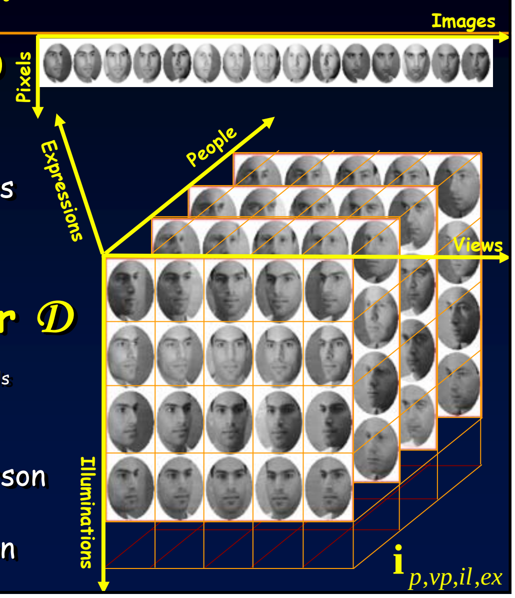
Data Organization

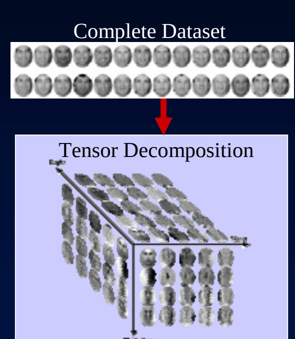
Linear/PCA: *Data Matrix* **D**

- $R^{\text{pixels x images}}$
- A matrix of image vectors

Multilinear: **Data Tensor** $\mathcal{D}$

- $R^{\text{people x views x illums x express x pixels}}$
- N-dimensional tensor
- 28 people, 45 images/person
- 5 views, 4 illuminations, 3 expressions per person





Learning Phase

(Tensor Decomposition)

Background on Tensor Decomposition

- Factor Analysis:
  - *Psychometrics, Econometrics, Chemometrics,…*
- SVD:
  - *[Eckart and Young, 1936]* (Psychometrika)

    "The approximation of one matrix by another of lower rank"
- 3-Way Factor Analysis:
  - *[Tucker,1966]* (Psychometrika)

    "Some mathematical notes on three mode factor analysis"
- N-Way Factor Analysis:
  - *[Harshman, 1970] – Parafac*
  - *[Carrol and Chang, 1970] – Candecomp*
  - *[Kruskal, 1977]*
  - *[Kroonenberg and De Leeuw, 1980]*
  - *[Kapteyn, Neudecker, and Wansbeek, 1986]*
  - *[Franc, 1992]*
  - *[de Lathauwer, 1997] [Kolda, 2001] …*

Matrix Decomposition - SVD

- A matrix $\mathbf{D} \in \Re^{I_1 x I_2}$ has a column and row space
- SVD orthogonalizes these spaces and decomposes **D**

  $$\mathbf{D} = \mathbf{U}_1 \mathbf{S} \mathbf{U}_2^T$$ ($\mathbf{U}_1$ contains the eigenfaces)
- Rewrite in terms of *mode-n products*

  $$\mathbf{D} = \mathbf{S} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2$$

Tensor Decomposition

$\mathcal{D}$ is a N-dimensional "matrix", with N spaces

- N-mode SVD is the natural generalization of SVD
- N-mode SVD orthogonalizes these spaces and decomposes $\mathcal{D}$ as the mode-n product of N-orthogonal spaces

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \cdots \times_n \mathbf{U}_n \cdots \times_N \mathbf{U}_N$$

- Mode-n matrix $\mathbf{U}_n$ spans the column space of $\mathbf{D}_{(n)}$
- Core tensor $\mathcal{Z}$ governs interaction between mode matrices

N-Mode SVD Algorithm

Two steps:

1. For n = 1,…,N, compute matrix $\mathbf{U}_n$ by computing the SVD of the flattened matrix $\mathbf{D}_{(n)}$ and setting $\mathbf{U}_n$ to be the left matrix of the SVD

2. Solve for the core tensor

$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$$

Multilinear PCA (MPCA) Algorithm
(Covariance Matrix Form)

1. Compute mode matrix $\mathbf{U}_n$ for each causal factor:

Compute the factor-n covariance matrix:

$$\left[\mathbf{D}_{[n]}\mathbf{D}_{[n]}^T\right]_{jk} = \sum_{f_1=1}^{F_1}\cdots\sum_{f_{n-1}=1}^{F_{n-1}}\sum_{f_{n+1}=1}^{F_{n+1}}\cdots\sum_{f_N=1}^{F_N}\mathbf{d}^T_{f_1\cdots f_{n-1}\,j\,f_{n+1}\cdots f_N}\,\mathbf{d}_{f_1\cdots f_{n-1}\,k\,f_{n+1}\cdots f_N}$$

Set $\mathbf{U}_n$:= left matrix of the SVD of the covariance matrix

2. Solve for the core tensor:

$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$$

Dimensionality Reduction

$$\mathbf{E} = \left\| \mathcal{D} - \mathcal{Z} \times_1 \mathbf{U}_1 \times \ldots \times_n \mathbf{U}_n \times \ldots \times_N \mathbf{U}_N \right\| + \sum_{n=1}^{N} \lambda_n \left\| \mathbf{U}_n \mathbf{U}_n^T - \mathbf{I} \right\|$$

Iterative data reduction approach:

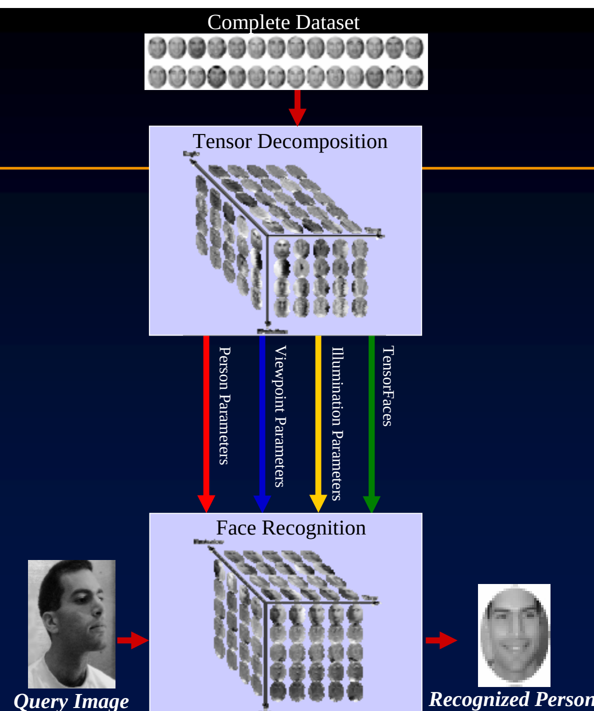
- Optimize mode per mode in an iterative way
- Alternating Least Squares (ALS) algorithm

Strategic Data Compression = Perceptual Quality

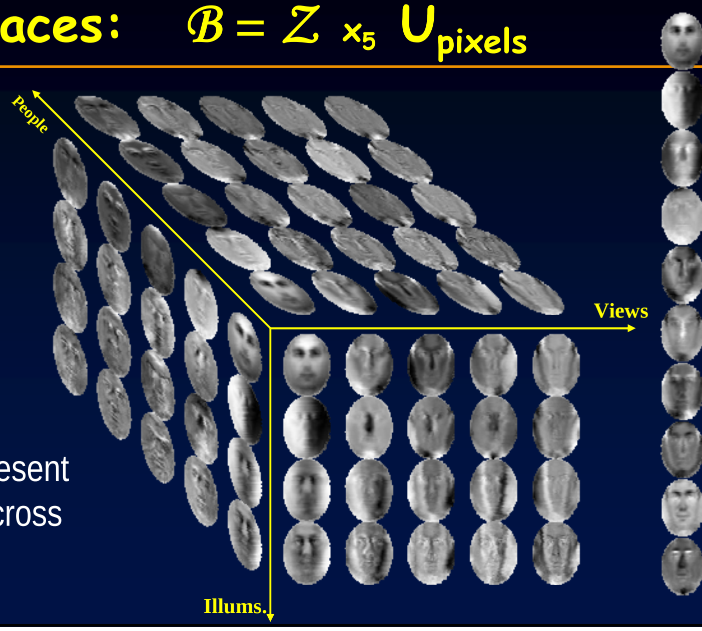
TensorFaces data reduction in illumination space primarily suppresses illumination effects (shadows, highlights)

Original	TensorFaces	TensorFaces	PCA
	6 illum + 11 people param.	3 illum + 11 people param.	33 parameters
176 basis vectors	66 basis vectors	33 basis vectors	33 basis vectors

- PCA can have *lower mean square error, yet higher perceptual error*

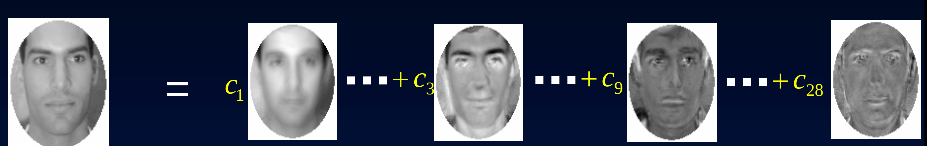


TensorFaces: $\mathcal{B} = \mathcal{Z} \times_5 \mathbf{U}_{pixels}$



TensorFaces:
explicitly represent covariance across factors

Linear Projection



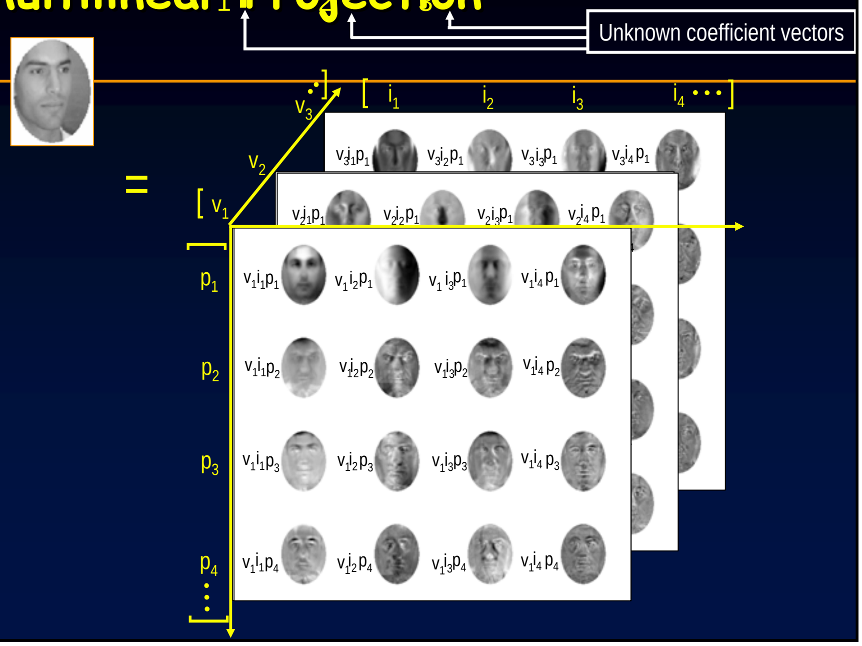
$$\mathbf{U}^T \mathbf{d}_i = \mathbf{U}\mathbf{c}_{ij}$$

Unknown coefficient vector

Projection Operator

[Vasilescu & Terzopoulos CVPR'05]

Multilinear Projection



Typical Facial Recognition Results

[Vasilescu & Terzopoulos, ICPR'02]

PIE Recognition Experiment	*PCA*	*TensorFaces*
Training: ***23 people, 3 viewpoints (0,+34,-34), 4 illuminations*** *Testing:* ***23 people, 2 viewpoints (+17,-17), 4 illuminations (center,left,right,left+right)***	*61%*	***80%***
Training: ***23 people, 5 viewpoints (0,+17, -17,+34,-34), 3 illuminations*** *Testing:* ***23 people, 5 viewpoints (0,+17, -17,+34,-34), 4th illumination***	*27%*	***88%***

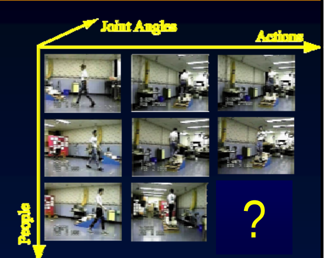
Perspective on Multilinear Models

	Linear Models	*Our Nonlinear (Multilinear) Models*
2nd - Order Statistics (covariance)	**PCA** Eigenfaces	**Multilinear PCA** TensorFaces
Higher- Order Statistics	**ICA**	**Multilinear ICA** Independent TensorFaces

[Vasilescu & Terzopoulos, Learning 2004]

Other Multilinear Applications

- Human Motion Signatures
  [*Vasilescu ICPR 02, CVPR 01, SIGGRAPH 01*]



- Image-Based Rendering
  [Vasilescu & Terzopoulos, SIGGRAPH 04]

BTF Texture Mapping

[Dana et al. 1999]

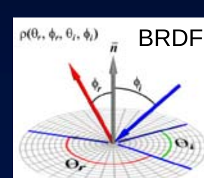
	Concrete	Pebbles	Plaster
Standard Texture Mapping			
BTF Texture Mapping			

Bidirectional Texture Function (BTF)

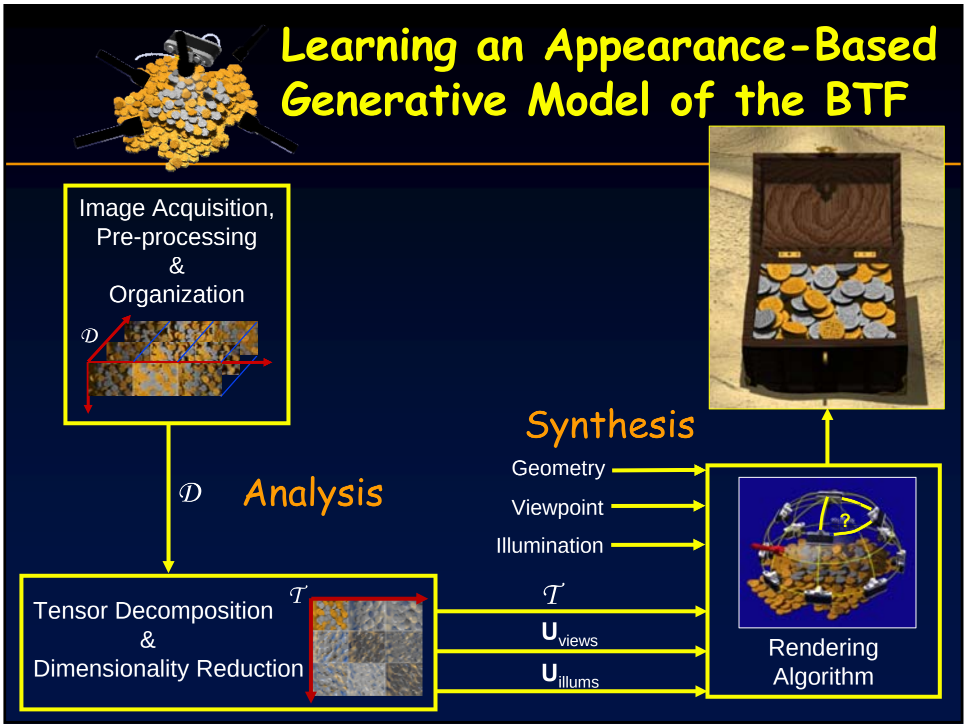
Surface reflectance as a function of position on surface, view direction, and illumination direction

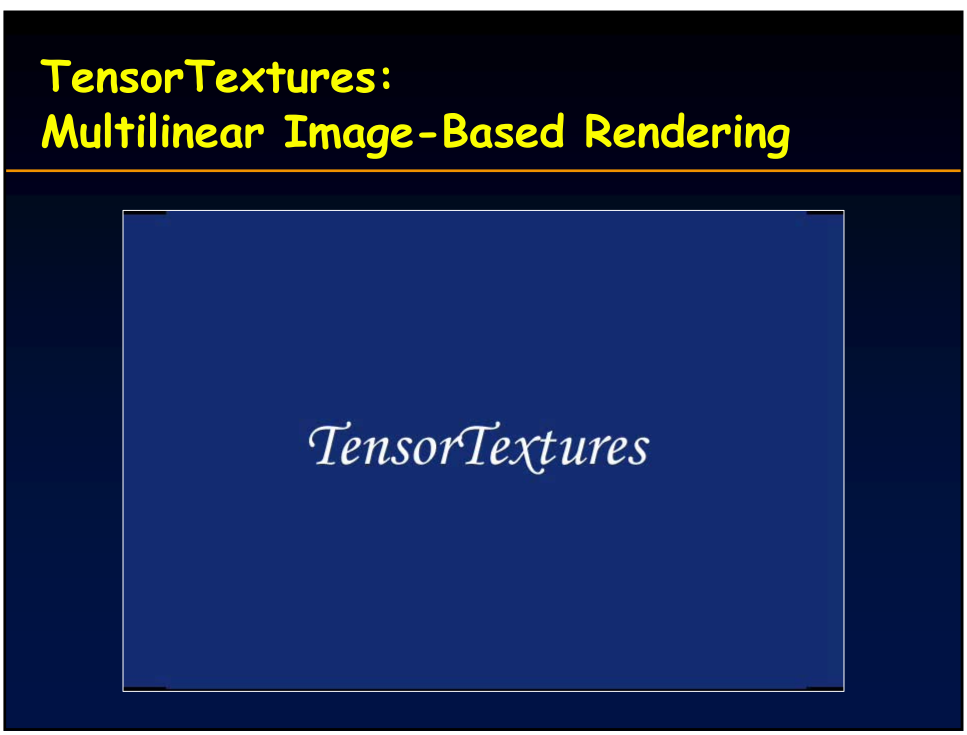
$$f_{BTF}(\underbrace{x, y}_{\text{position on surface (texel)}}, \underbrace{\theta_v, \phi_v}_{\text{view direction}}, \underbrace{\theta_i, \phi_i}_{\text{illumination direction}})$$

view direction + illumination direction: photometric angles



- The BTF captures shading and mesostructural self-shadowing, self-occlusion, interreflection, subsurface scattering


Learning an Appearance-Based
Generative Model of the BTF
Image Acquisition,
Pre-processing
&
Organization
D
D
Analysis
Synthesis
Geometry
Viewpoint
Illumination
?
Tensor Decomposition
&
Dimensionality Reduction
T
T
U views
U illums
Rendering
Algorithm


TensorTextures:
Multilinear Image-Based Rendering
TensorTextures

Rendered Texture for a Planar Surface

Generalizations: Kernelization

Factor-n Covariance Matrix:

$$\underbrace{\left[\mathbf{D}_{[n]}\mathbf{D}_{[n]}^T\right]_{jk}}_{\text{Covariance Matrix}} = \sum_{f_1=1}^{F_1}\cdots\sum_{f_{n-1}=1}^{F_{n-1}}\sum_{f_{n+1}=1}^{F_{n+1}}\cdots\sum_{f_N=1}^{F_N} \underbrace{\mathbf{d}^T_{f_1\cdots f_{n-1}\, j\, f_{n+1}\cdots f_N}\,\mathbf{d}_{f_1\cdots f_{n-1}\, k\, f_{n+1}\cdots f_N}}_{\text{Inner Product of Images}}$$

Factor-n Kernel Covariance Matrix:

$$\left[\mathbf{C}_K\right]_{jk} = \sum_{f_1=1}^{F_1}\cdots\sum_{f_{n-1}=1}^{F_{n-1}}\sum_{f_{n+1}=1}^{F_{n+1}}\cdots\sum_{f_N=1}^{F_N} K\left(\mathbf{d}_{f_1\cdots f_{n-1}\, j\, f_{n+1}\cdots f_N},\mathbf{d}_{f_1\cdots f_{n-1}\, k\, f_{n+1}\cdots f_N}\right)$$

Kernel MPCA (KMPCA) Algorithm

1. *Compute mode matrix* $\mathbf{U}_n$ *for each causal factor:*

Compute the factor-*n* covariance matrix:

$$[\mathbf{C}_K]_{jk} = \sum_{f_1=1}^{F_1} \cdots \sum_{f_{n-1}=1}^{F_{n-1}} \sum_{f_{n+1}=1}^{F_{n+1}} \cdots \sum_{f_N=1}^{F_N} K\left(\mathbf{d}_{f_1 \cdots f_{n-1}\, j\, f_{n+1} \cdots f_N}, \mathbf{d}_{f_1 \cdots f_{n-1}\, k\, f_{n+1} \cdots f_N}\right)$$

Set $\mathbf{U}_n$:= left matrix of the SVD of the covariance matrix

2. *Solve for the core tensor:*

$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$$

Kernels

- Linear kernel: $K(\mathbf{u},\mathbf{v}) = \mathbf{u}^T\mathbf{v} = \mathbf{u}\cdot\mathbf{v}$
- Polynomial kernel up to degree d: $K(\mathbf{u},\mathbf{v}) = \left(\mathbf{u}^T\mathbf{v}+1\right)^d$
- Sigmoidal kernel: $K(\mathbf{u},\mathbf{v}) = \tanh\left(\alpha\mathbf{u}^T\mathbf{v}+\beta\right)^d$
- Gaussian radial basis function (RBF) kernel: $K(\mathbf{u},\mathbf{v}) = e^{-\frac{\|\mathbf{u}-\mathbf{v}\|^2}{2\sigma}}$

Connection to manifold learning algorithms

- Locally Linear Embedding kernel: $K_{\mathrm{LLE}}(\mathbf{u},\mathbf{v})$
- Isomap kernel: $K_{\mathrm{IM}}(\mathbf{u},\mathbf{v})$
- Laplacian Eigenmap kernel: $K_{\mathrm{LE}}(\mathbf{u},\mathbf{v})$

Introduction

- Computer vision (analysis)
- Computer graphics (synthesis)

} Modeling

Talk Overview

1. Statistical models
2. Physics-based models

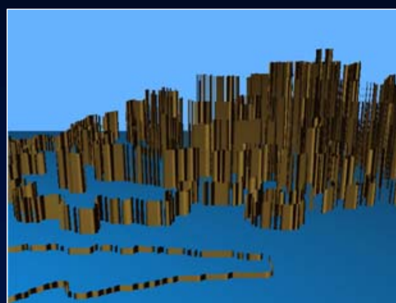
Deformable Models

A powerful, model-based image analysis approach

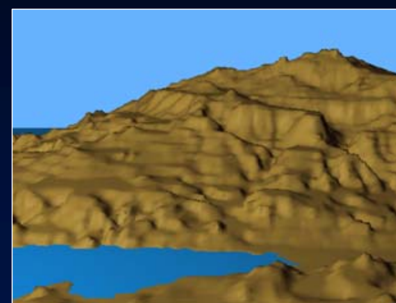
- Proposed in computer vision (and graphics)
- Actively explored in medical image analysis
- Combine bottom-up and top-down analysis
- Accommodate shape & motion constraints/variability
- Support intuitive interaction mechanisms
- Incorporate a priori anatomical knowledge

Example

Scattered data fitting



Input Data

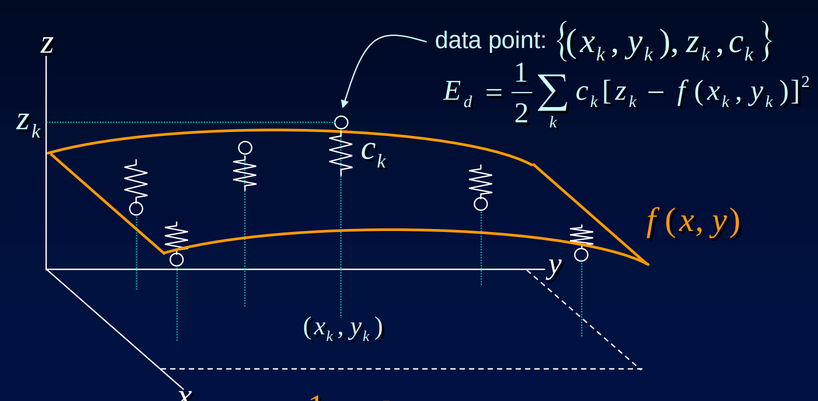


Desired Result

Visual Surface Reconstruction

[Terzopoulos, 1980-84]

Regularization: Thin-plate spline under tension



$$E(f) = \frac{1}{2}\iint \rho\left[\tau\left(f_x^2 + f_y^2\right) + (1-\tau)\left(f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2\right)\right] dx\,dy$$

Discontinuity-Preserving Surface Reconstruction

Make "rigidity" & "tension" functions of (x,y)

- Tangent discontinuities:

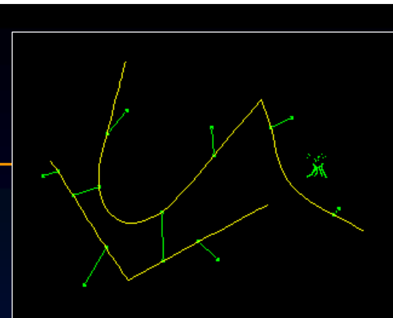
$$\tau(x, y) = 1$$

- Jump discontinuities:

$$\rho(x, y) = 0$$

Snakes: Active Contours

[Kass, Witkin, Terzopoulos, 1987]



- Curve representation:

$$\mathbf{c}(u,t) = \begin{bmatrix} x(u,t) \\ y(u,t) \end{bmatrix}; \quad u \in [0,1]$$

- Curve deformation energy: $E(\mathbf{c}) = \frac{1}{2}\int_0^1 w_1 \left|\frac{\partial \mathbf{c}}{\partial u}\right|^2 + w_2 \left|\frac{\partial^2 \mathbf{c}}{\partial u^2}\right|^2 du$

- Equations of motion: $\mu\ddot{\mathbf{c}} + \gamma\dot{\mathbf{c}} + \delta_{\mathbf{c}} E(\mathbf{c}) = \mathbf{f}$

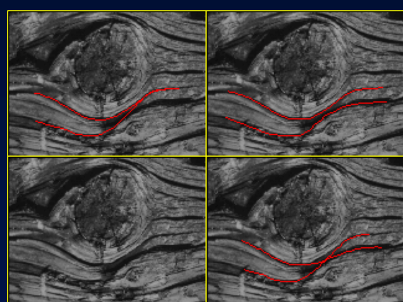
$$\mu\ddot{\mathbf{c}} + \gamma\dot{\mathbf{c}} - \frac{\partial}{\partial u}\left(w_1 \frac{\partial \mathbf{c}}{\partial u}\right) + \frac{\partial^2}{\partial u^2}\left(w_2 \frac{\partial \mathbf{c}^2}{\partial u^2}\right) = \mathbf{f}$$

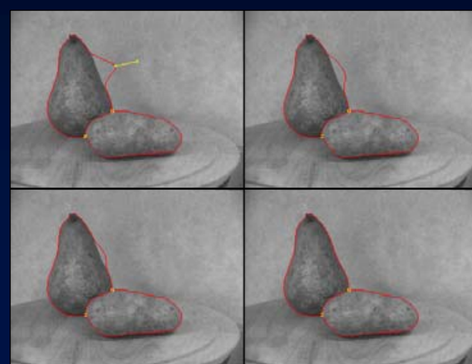
Image Analysis Using Snakes

External forces come from an image

$$\mathbf{f} = -\nabla P(\mathbf{c})$$

- Image potential: $P(x, y)$

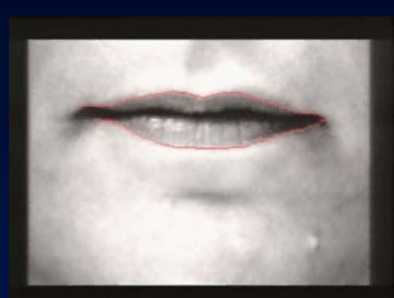


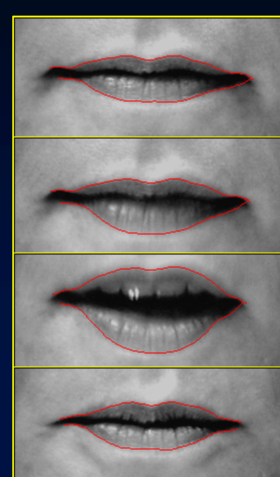


Motion Tracking in Video

Time-varying image potential

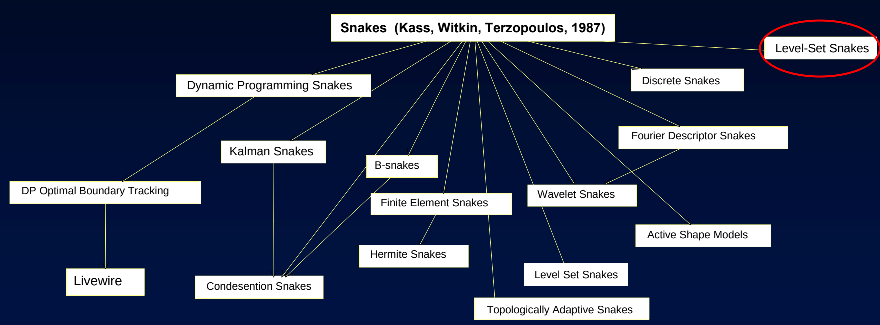
$P(x, y, t)$





Family of Active Contours

Numerous variants



- Level-set implementation of active contours
  Eulerian (implicit) vs Lagrangian (parametric)

Deformable Surfaces

- Surface representation:

$$\mathbf{s}(u,v,t) = \begin{bmatrix} x(u,v,t) \\ y(u,v,t) \\ z(u,v,t) \end{bmatrix}; \quad u,v \in [0,1]$$

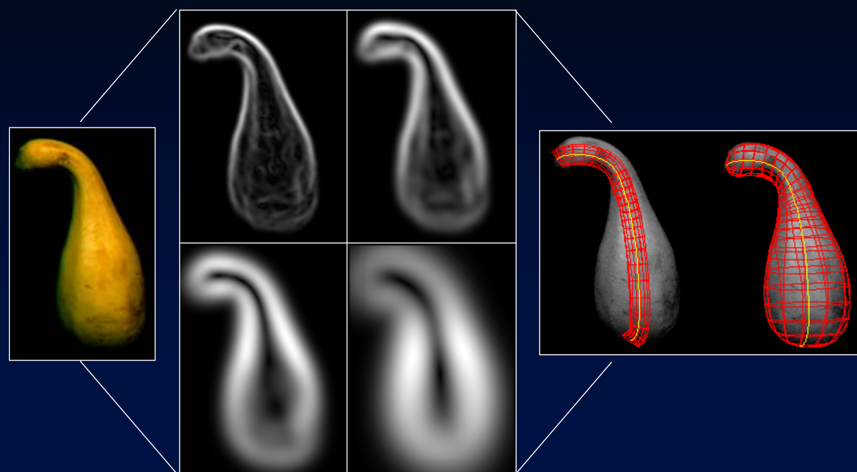
- Surface deformation energy:

$$E(\mathbf{s}) = \frac{1}{2}\int_0^1\int_0^1 w_{10}\left|\frac{\partial \mathbf{s}}{\partial u}\right|^2 + w_{01}\left|\frac{\partial \mathbf{s}}{\partial v}\right|^2 + w_{20}\left|\frac{\partial^2 \mathbf{s}}{\partial u^2}\right|^2 + 2w_{11}\left|\frac{\partial^2 \mathbf{s}}{\partial u \partial v}\right|^2 + w_{02}\left|\frac{\partial^2 \mathbf{s}}{\partial v^2}\right|^2 du\,dv$$

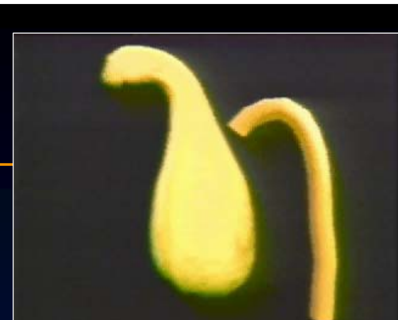
- Equations of motion: $\mu\ddot{\mathbf{s}} + \gamma\dot{\mathbf{s}} + \delta_{\mathbf{s}}E(\mathbf{s}) = \mathbf{f}$

$$\mu\ddot{\mathbf{s}} + \gamma\dot{\mathbf{s}} - \frac{\partial}{\partial u}\left(w_{10}\frac{\partial \mathbf{s}}{\partial u}\right) - \frac{\partial}{\partial v}\left(w_{01}\frac{\partial \mathbf{s}}{\partial v}\right) + \frac{\partial^2}{\partial u^2}\left(w_{20}\frac{\partial^2 \mathbf{s}}{\partial u^2}\right) + 2\frac{\partial^2}{\partial u \partial v}\left(w_{11}\frac{\partial^2 \mathbf{s}}{\partial u \partial v}\right) + \frac{\partial^2}{\partial v^2}\left(w_{02}\frac{\partial^2 \mathbf{s}}{\partial v^2}\right) = \mathbf{f}$$

Deformable Model Reconstruction

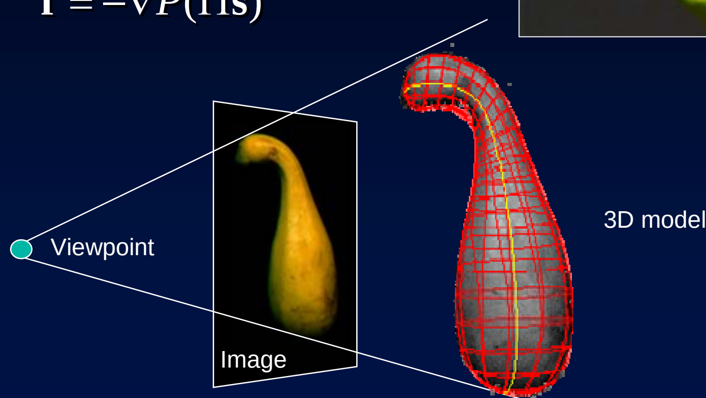


Reconstruction



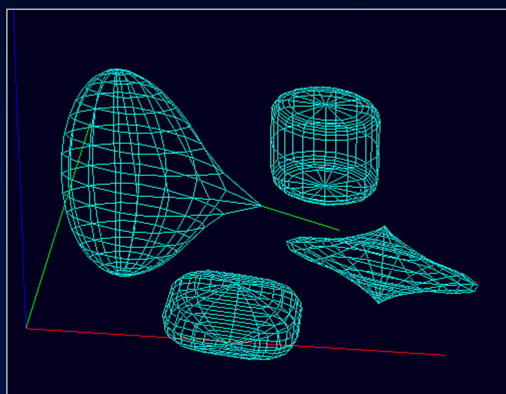
3D to 2D projection

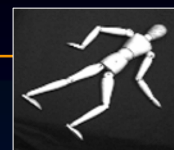
$$\mathbf{f} = -\nabla P(\Pi \mathbf{s})$$



Deformable Superquadrics

Parameterized models with local deformations





Deformable Models for Graphics: Deformation Strain Energies

Differential geometry: Fundamental theorems

- Curves

arc length — curvature — torsion

$$E(\mathbf{x}) = \int \alpha(s - s^0)^2 + \beta(\kappa - \kappa^0)^2 + \gamma(\tau - \tau^0)^2 \, du$$

- Surfaces

metric tensor — curvature tensor

$$E(\mathbf{x}) = \int \left\| \mathbf{G} - \mathbf{G}^0 \right\|_\alpha^2 + \left\| \mathbf{B} - \mathbf{B}^0 \right\|_\beta^2 \, du$$

- Solids

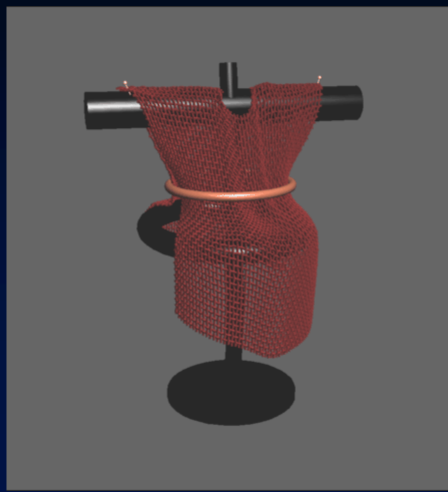
metric tensor

$$E(\mathbf{x}) = \int \left\| \mathbf{G} - \mathbf{G}^0 \right\|_\alpha^2 \, du$$

$$\mathbf{G}_{ij}(\mathbf{x}(u)) = \frac{\partial \mathbf{x}}{\partial u_i} \bullet \frac{\partial \mathbf{x}}{\partial u_j}$$

$$\mathbf{B}_{ij}(\mathbf{x}(u)) = \hat{\mathbf{n}} \bullet \frac{\partial^2 \mathbf{x}}{\partial u_i \partial u_j}$$

Cloth Simulation: Draped Robe

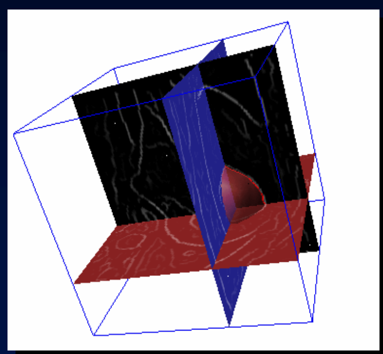


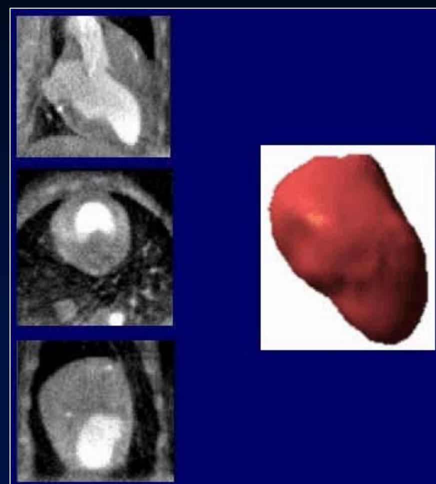
Deformable Models in Medical Image Analysis

- Segmentation
- Shape reconstruction and modeling
- Motion estimation and analysis
- Registration
- Functional modeling

Cardiac LV Motion Tracking

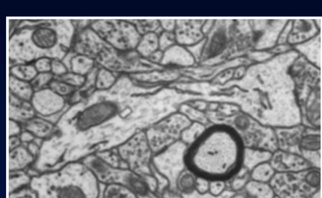
[McInerney & Terzopoulos, 1995]





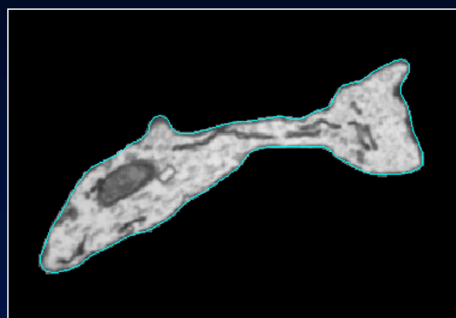
Interactive Medical Image Segmentation using Snakes

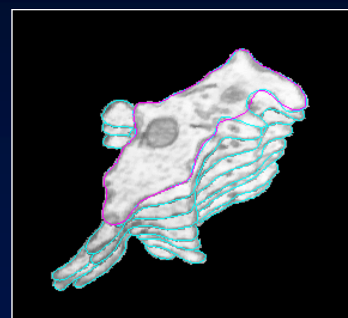
EM neuronal tissue sections



Reconstruction of Neuronal Dendrite

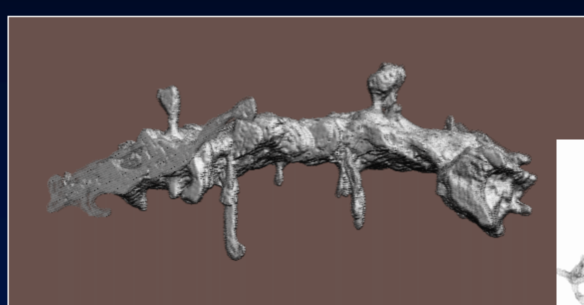
Cell interiors stacked in 3D

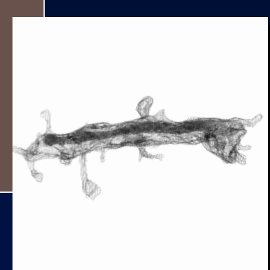




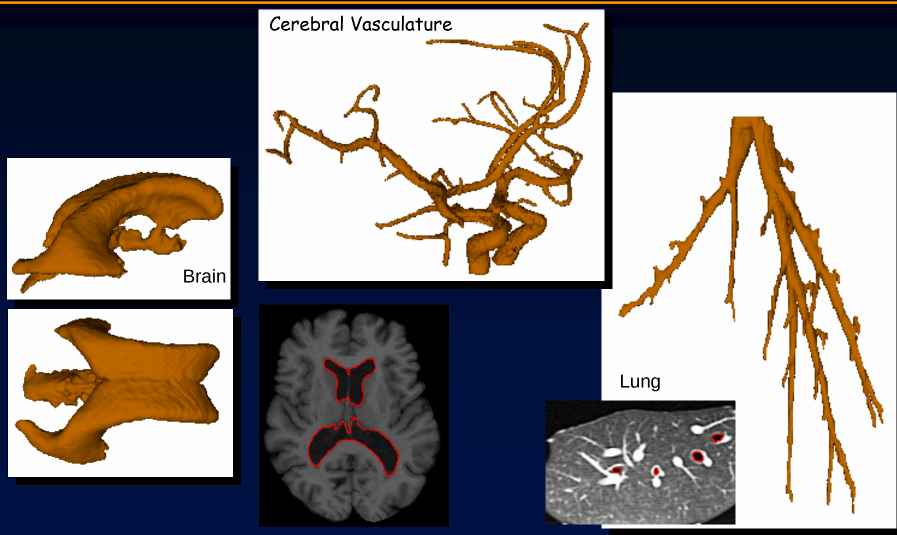
Visualization of Dendrite

Ray-traced interpolated volume

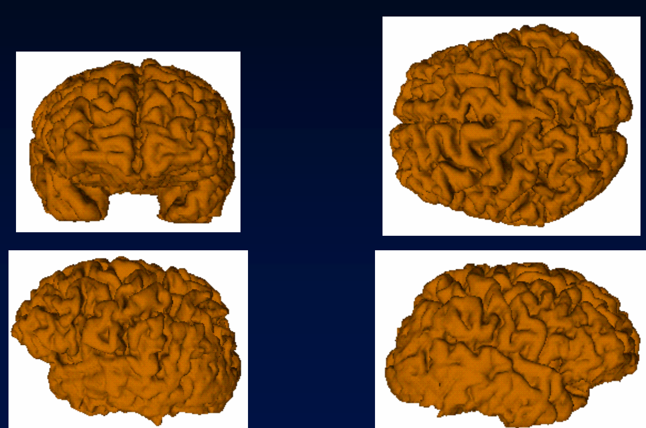




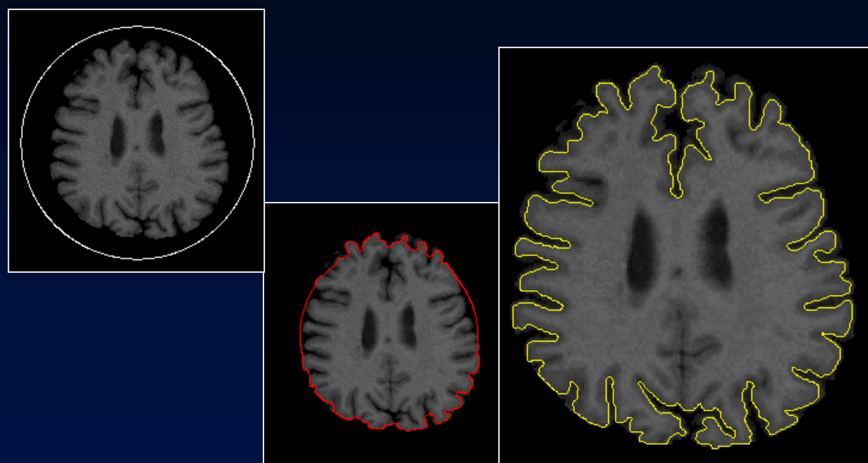
Complex Structure Extraction



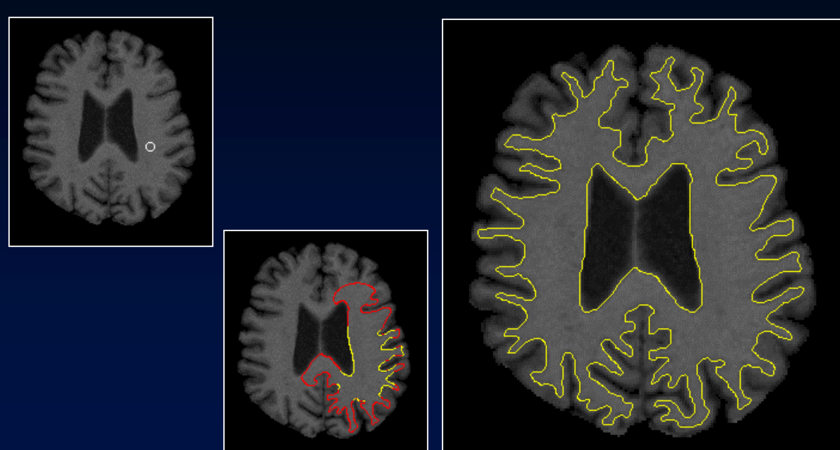
T-Surface Segmentation of Cortex



Shrink-Wrap Segmentation

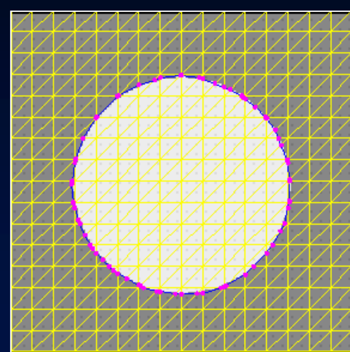


T-Snake Segmentation of Brain Image



Affine Cell Image Decomposition

ACID makes snakes topologically flexible



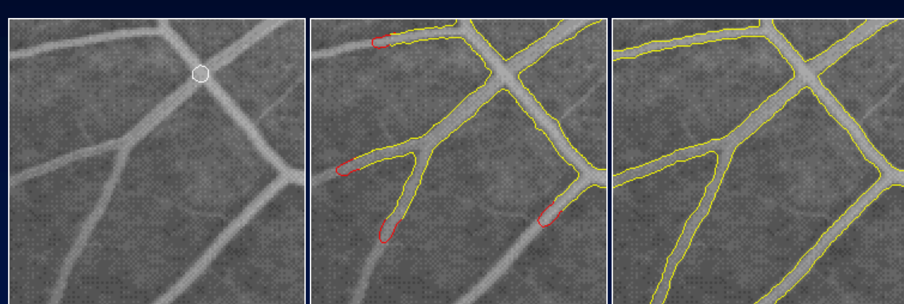
- ACID grid continually reparameterizes snake

Topologically Adaptive Snakes

[McInerney & Terzopoulos, 1996]

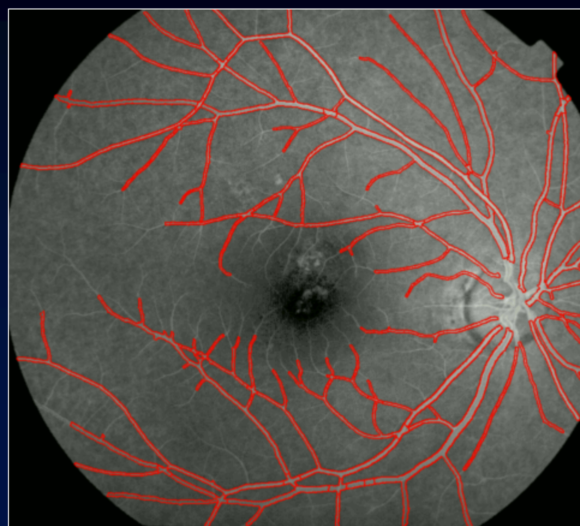
Segmenting Retinal Angiogram

- T-snake flows and bifurcates



Initial Model | Flow | Segmented Angiogram

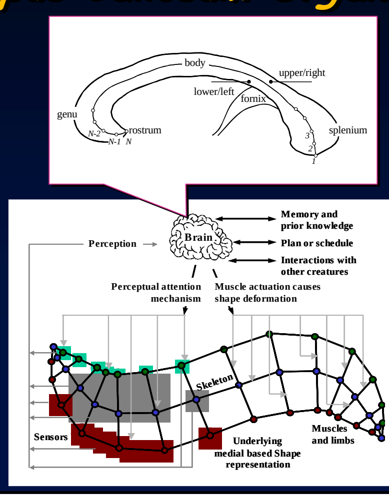
Retinal Angiogram Segmentation



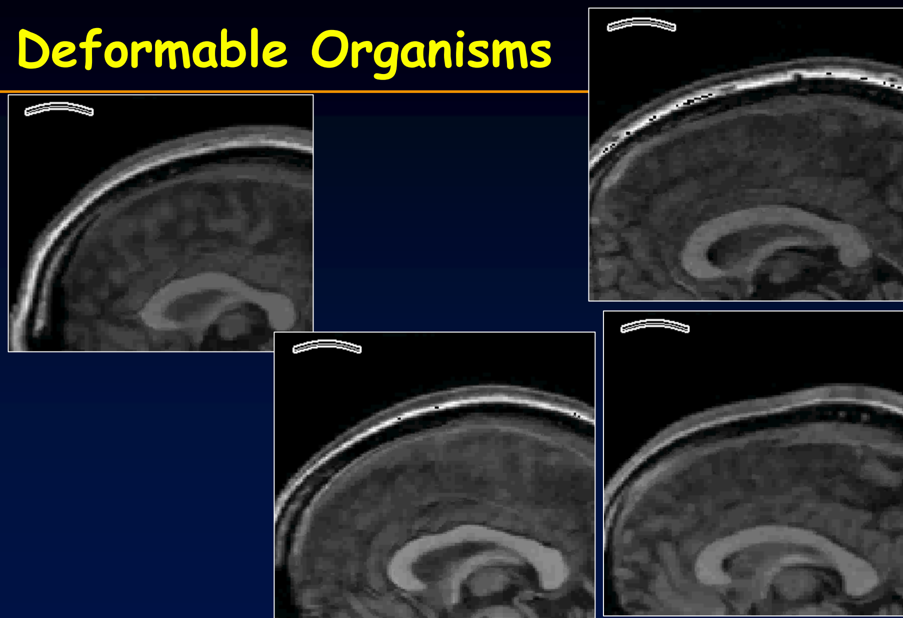
Deformable Organisms

[Hamarneh, McInerney, Terzopoulos, MICCAI'01, MIA'01]

Corpus Callosum Organism



Deformable Organisms



Deformable Organisms