1. Deformable and Functional Models In Medical Image Analysis

2. A Tensor Algebraic Framework for Image Synthesis, Analysis & Recognition

Demetri Terzopoulos

Computer Science Department

University of California, Los Angeles

Deformable Models

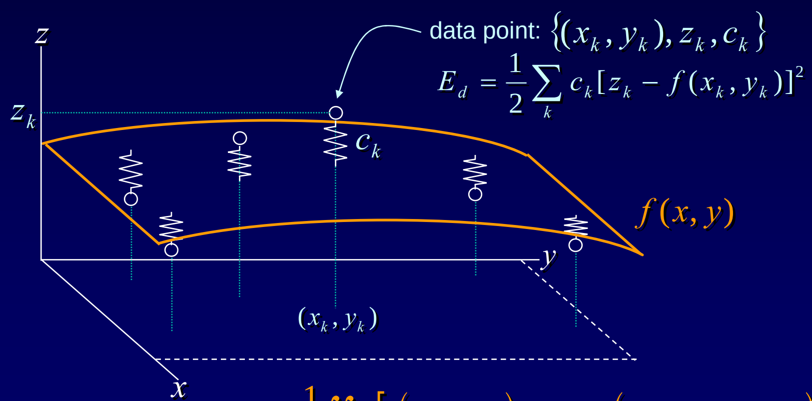
A powerful, model-based medical image analysis approach

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

Computing Visible Surfaces from Scattered Visual Data

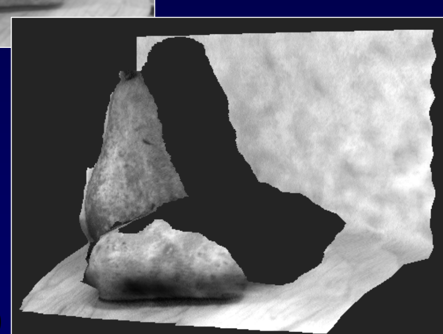
[Terzopoulos, 1984]

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

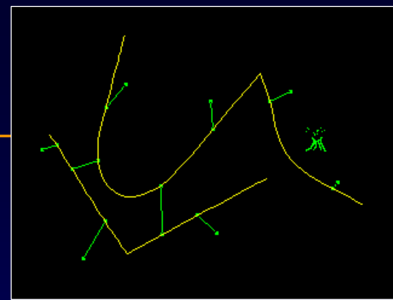


Make "rigidity" & "tension" functions of (x,y)

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

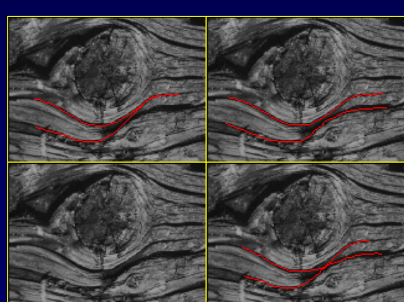
$$\mu \ddot{\mathbf{c}} + \gamma \dot{\mathbf{c}} - \frac{\partial}{\partial u}\left(w_1 \frac{\partial \mathbf{c}}{\partial u} \right) + \frac{\partial^2}{\partial u^2}\left(w_2 \frac{\partial \mathbf{c}^2}{\partial u^2} \right) = \mathbf{f}$$

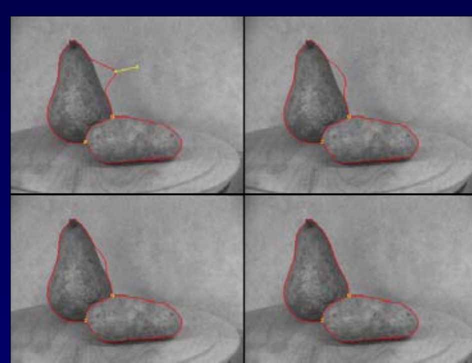
Image Analysis Using Snakes

External forces come from an image

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

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

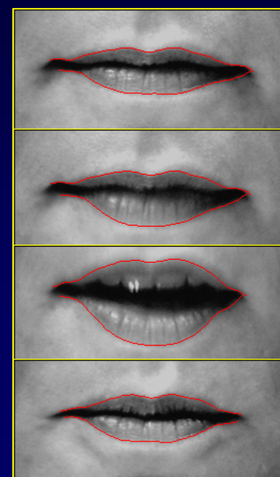




Motion Tracking in Video

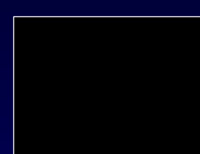
Time-varying image potential

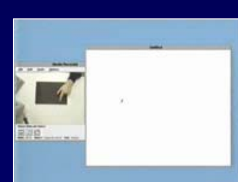
$P(x,y,t)$



Snake-Based Tracking

(Blake & Isard, Oxford University)





Discretization

- Continuous equations of motion

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

- Discrete equations of motion

$$\mathbf{M}\ddot{\mathbf{c}}+\mathbf{D}\dot{\mathbf{c}}+\mathbf{K}\mathbf{c}=\mathbf{f}$$

Mass matrix

Damping matrix

Stiffness matrix

External forces

Snake Stiffness Matrix

Finite differences:

$$\mathbf{c}_i=\mathbf{c}(ih);\quad i=0,\cdots,N-1$$

$$\frac{\partial\mathbf{c}}{\partial u}\approx\frac{\mathbf{c}_{i+1}-\mathbf{c}_i}{h}$$

$$\frac{\partial^2\mathbf{c}}{\partial u^2}\approx\frac{\mathbf{c}_{i+1}-2\mathbf{c}_i+\mathbf{c}_{i-1}}{h^2}$$

$$\mathbf{K}=\begin{bmatrix} a_0 & b_0 & c_0 & & & & c_{N-2} & b_{N-1} \\ b_0 & a_1 & b_1 & c_1 & & & & c_{N-1} \\ c_0 & b_1 & a_2 & b_2 & c_2 & & & \\ & c_1 & b_2 & a_3 & b_3 & c_3 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & c_{N-5} & b_{N-4} & a_{N-3} & b_{N-3} & c_{N-3} \\ c_{N-2} & & & & c_{N-4} & b_{N-3} & a_{N-2} & b_{N-2} \\ b_{N-1} & c_{N-1} & & & & c_{N-3} & b_{N-2} & a_{N-1} \end{bmatrix}$$

$$a_i=\frac{w_{1;i-1}+w_{1;i}}{h^2}+\frac{w_{2;i-1}+4w_{2;i}+w_{2;i+1}}{h^4}$$

$$b_i=-\frac{w_{1;i}}{h^2}-\frac{2w_{2;i}+2w_{2;i+1}}{h^4}$$

$$c_i=\frac{w_{2;i+1}}{h^4}$$

$$w_{1;i}=w_1(ih)$$

$$w_{2;i}=w_2(ih)$$

Stable, Implicit Euler Time-Integration Method

Solve linear system at each time step

$$\begin{cases} \mathbf{A}^{(t)}\,\dot{\mathbf{c}}^{(t+\delta t)} = \dot{\mathbf{c}}^{(t)} + \mathbf{g}^{(t)} \\ \mathbf{c}^{(t+\delta t)} = dt\,\dot{\mathbf{c}}^{(t+\delta t)} + \mathbf{c}^{(t)} \end{cases}$$

- Efficient skyline storage of $\mathbf{A}^{(t)}$
- LU factorization of $\mathbf{A}^{(t)}$
- Forward / Back substitution solves for $\dot{\mathbf{c}}^{(t+\delta t)}$

Deformable Surfaces

[Terzopoulos, 1986; Terzopoulos, Witkin, Kass, 1987]

- Surface representation:

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

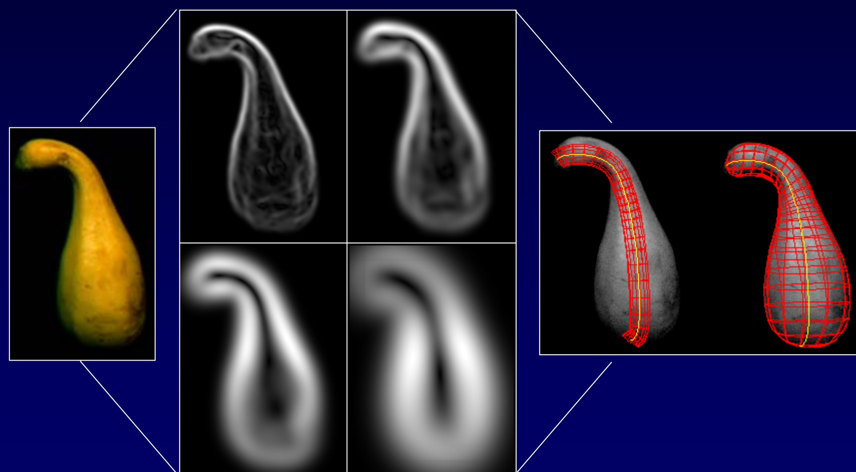
- Surface deformation energy:

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

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

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

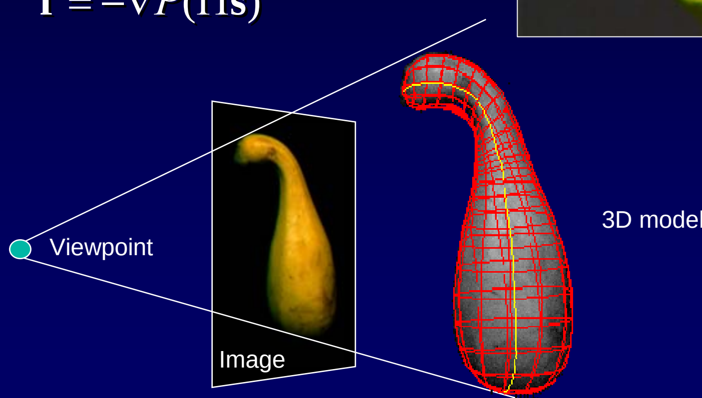
Deformable Model Reconstruction



Reconstruction

3D to 2D projection

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

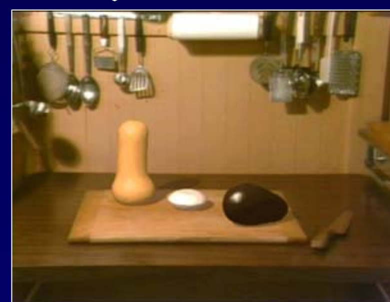


Graphics / Vision

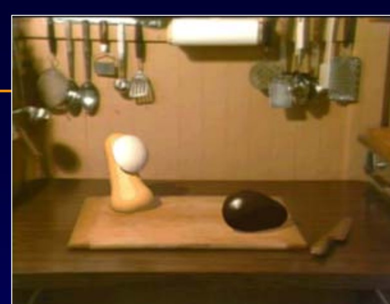
Converse problems

"Cooking with Kurt" (1987)

Image → CV → Reconstructed 3D Scene



Vegetable Model Animation



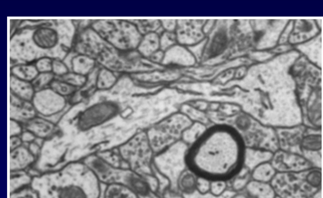
Medical Image Analysis Tasks

- Segmentation
- Shape modeling
- Matching
- Motion recovery and analysis
- Functional modeling

Interactive Medical Image Segmentation using Snakes

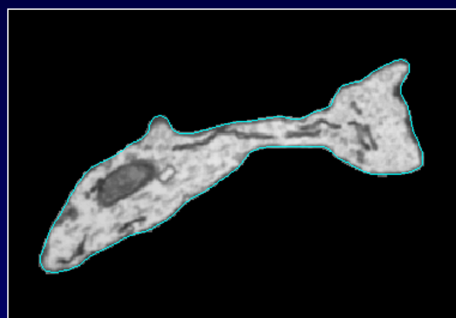
[Carlbom, Terzopoulos, Harris, 1994]

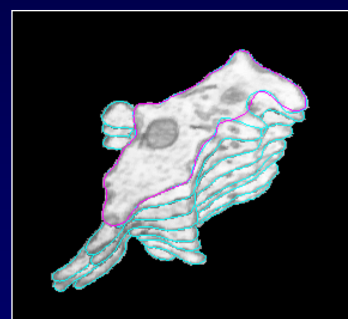
EM neuronal tissue sections



Reconstruction of Neuronal Dendrite

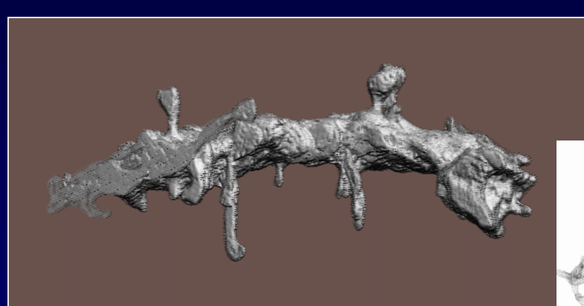
Cell interiors stacked in 3D

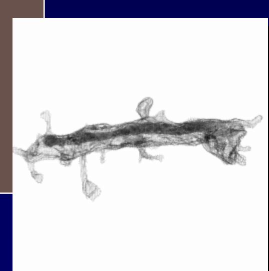




Visualization of Dendrite

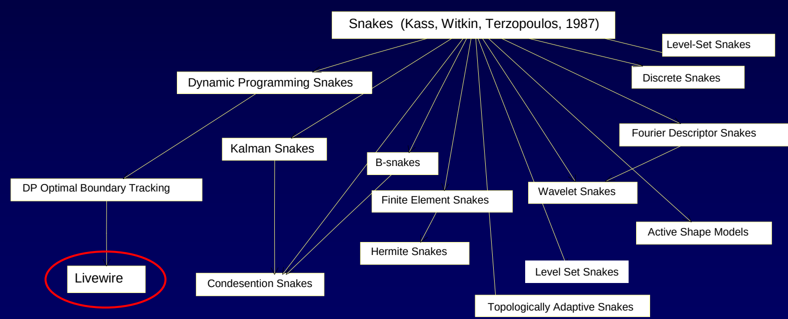
Ray-traced interpolated volume





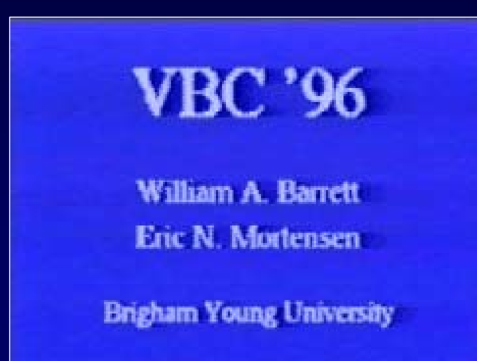
Family of Snakes

Many snakes variants



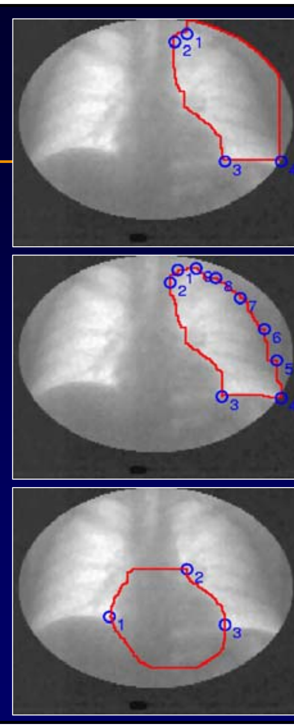
- "Livewire" or "Intelligent Scissors":
  An interactive boundary tracing tool

Livewire Demo



Limitations of Livewire

- No control of trace between seed points; only backtracking
- Many seed points needed for complex boundaries
- Nearby strong edges can capture trace (on-the-fly training)
- Fundamentally image-based
  - *cannot bridge gaps*
  - *smoothness not guaranteed*



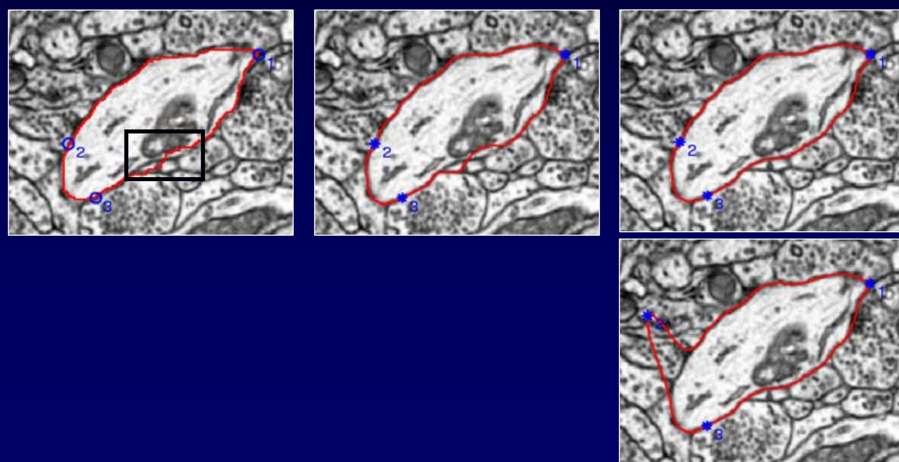
Combining Snakes and Livewire

“United Snakes”

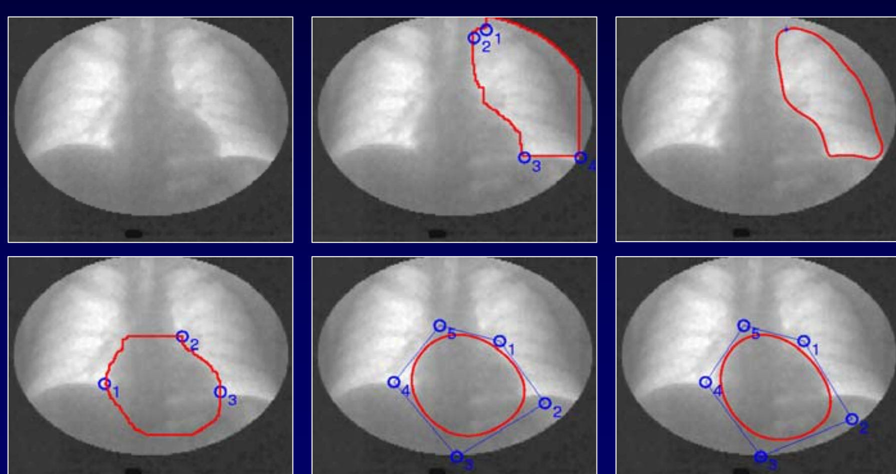
- Livewire serves for quick initialization of snakes
  - *typically requires fewer seed points*
- Livewire-initialized snakes quickly lock on boundaries
- Snakes enable adjustment of traces between seeds
  - *snake provides subpixel accuracy*
- Snake energy imposes smoothness and bridges gaps
- Livewire seed points capture user’s knowledge
  - *can serve as hard or soft constraints on snake*

Combining Snakes and Livewire

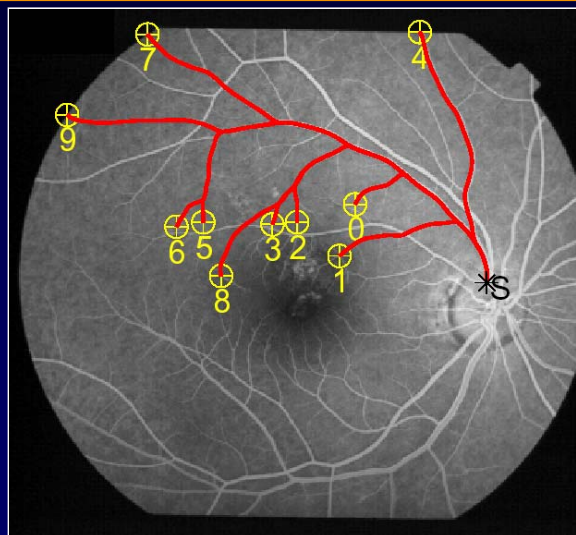
"United Snakes" accrue benefits of both



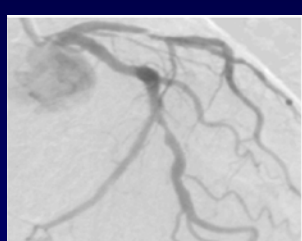
Dynamic Chest Image Analysis

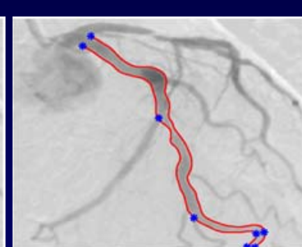


Vessel Segmentation



Vessel Segmentation



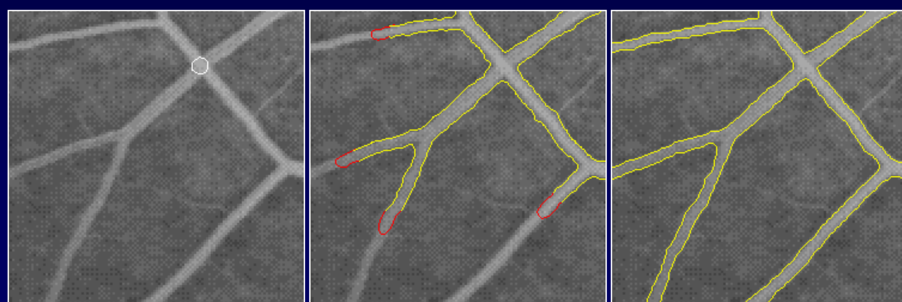


Topologically Adaptive Snakes

(McInerney & Terzopoulos, 1996)

Segmenting Retinal Angiogram

- T-snake flows and bifurcates

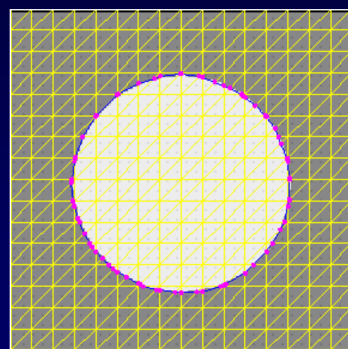


Initial Model | Flow | Segmented Angiogram

Retinal Angiogram Segmentation

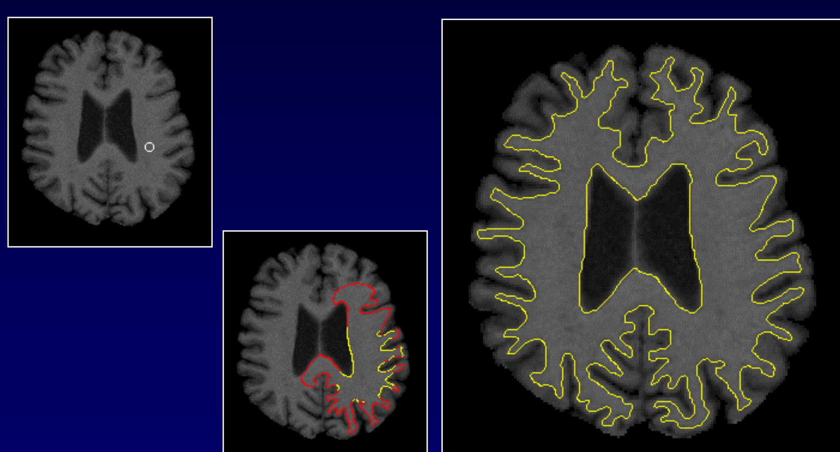
Affine Cell Image Decomposition

ACID makes snakes topologically flexible

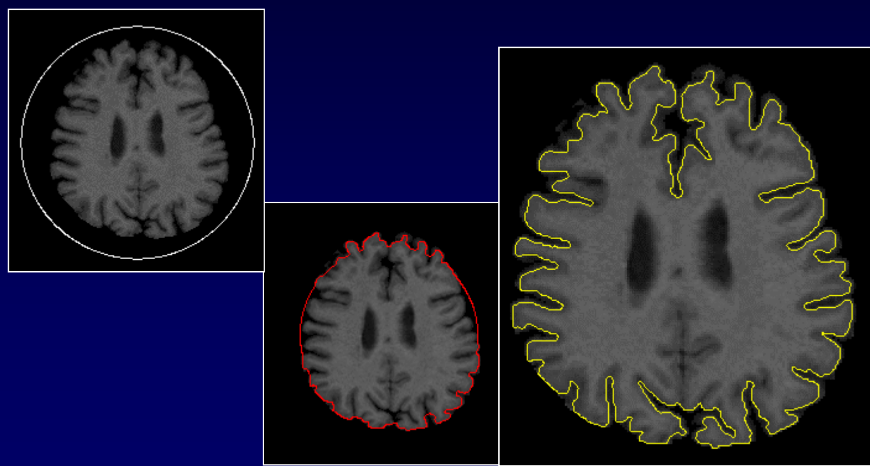


- ACID grid continually reparameterizes snake

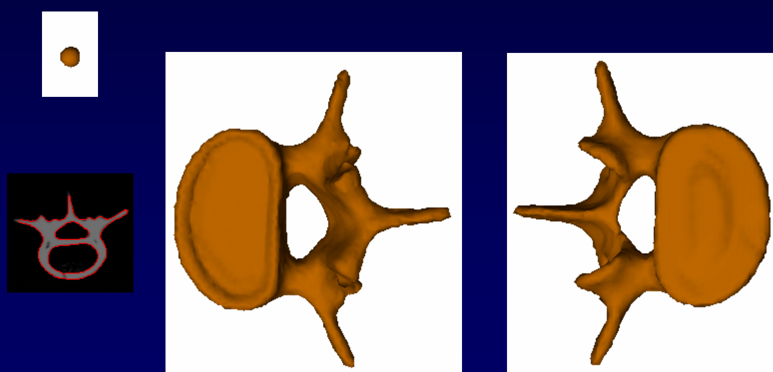
T-Snake Segmentation of Brain Image



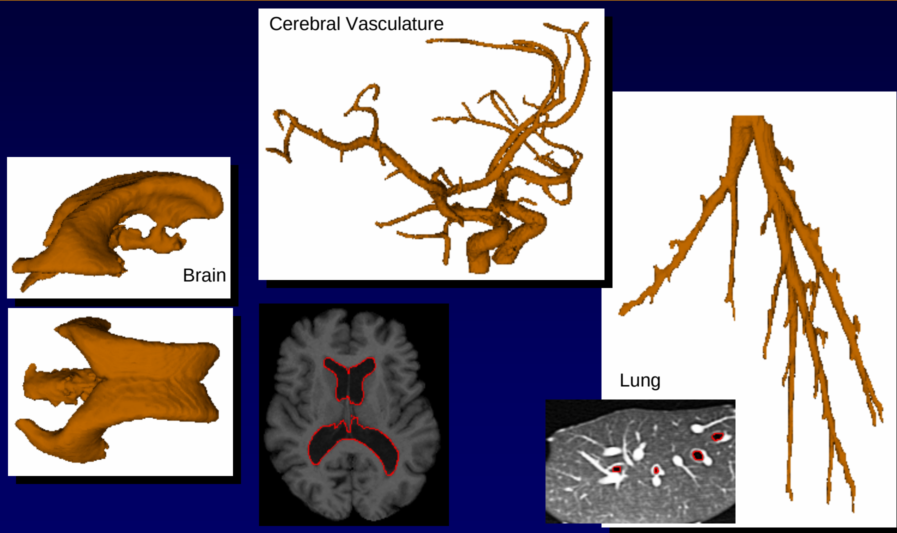
Shrink-Wrap Segmentation



Vertebra Reconstruction

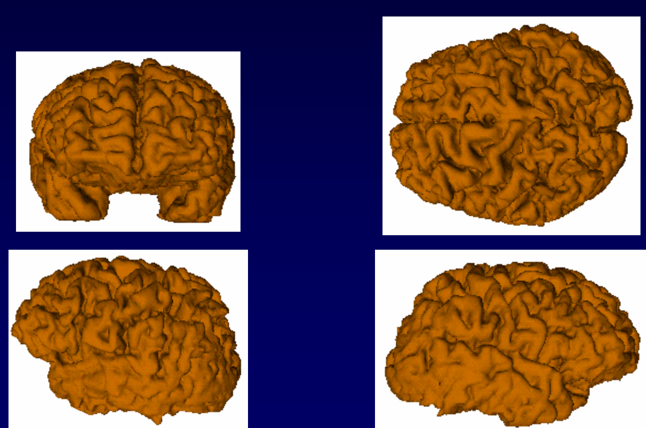


Complex Structure Extraction



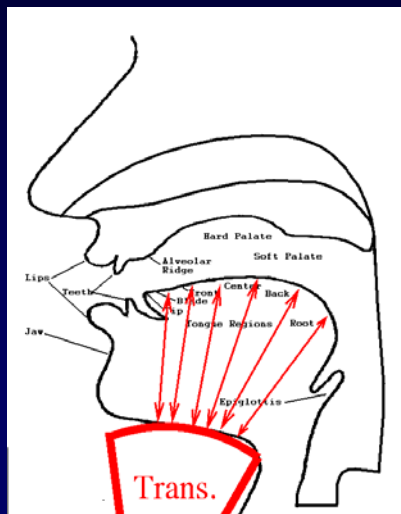
T-Surface Segmentation of Cortex

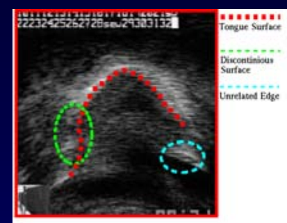
[McInerney & Terzopoulos, 1997]



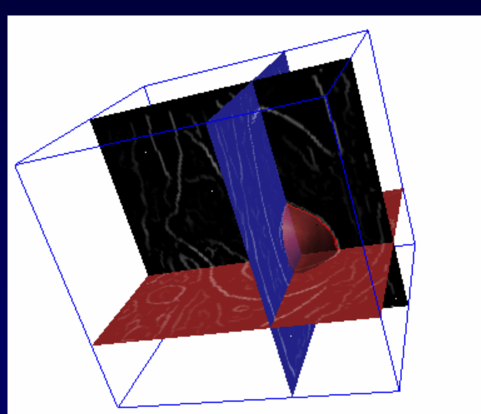
Tongue Tracking in Ultrasound

[Kambhamettu et al, 1999]

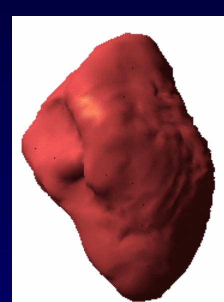




LV Reconstruction

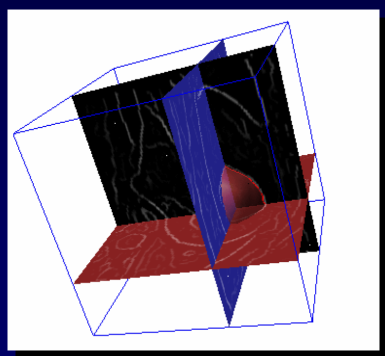


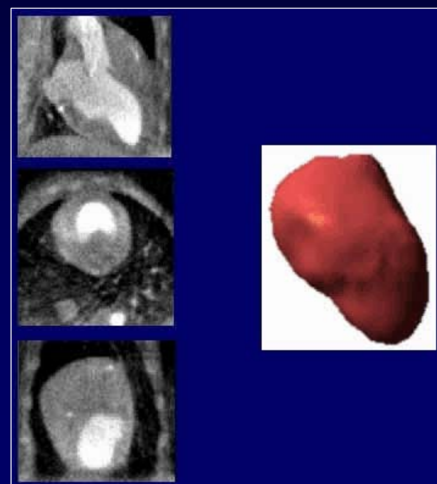
Deformable Balloon in Processed DSR Data



Reconstructed LV

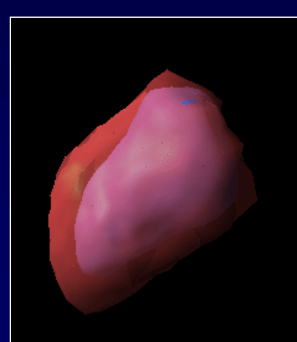
Cardiac LV Motion Tracking





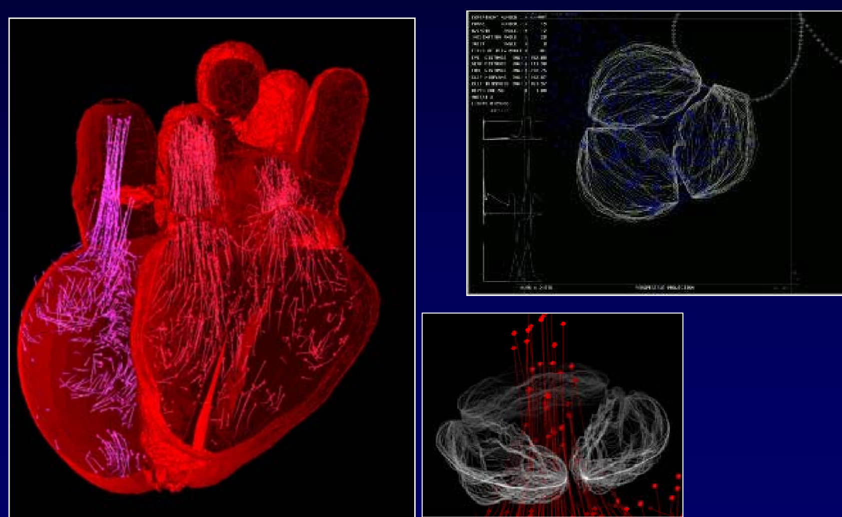
Systolic/Diastolic LV

Computing ejection fraction



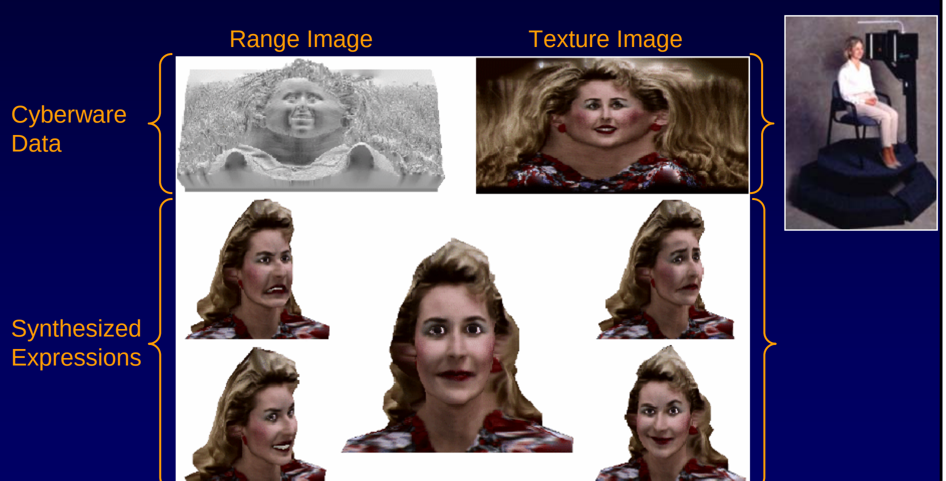
Functional Model of the Heart

[Peskin & McQueen]



Artificial Humans

Scanned Data ➡ Synthetic Faces



Fitting the Generic Mesh

Feature-based image matching algorithm

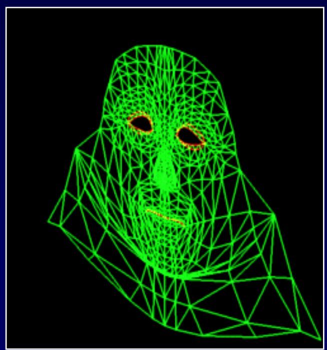
localizes facial features in:

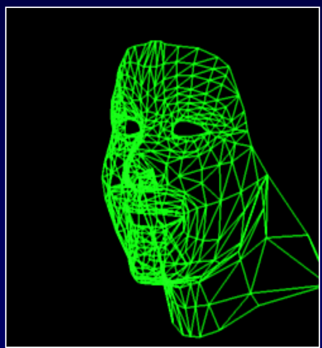
Processed range image

RGB texture image

Sampling Facial Shape

Fitted mesh nodes sample range data

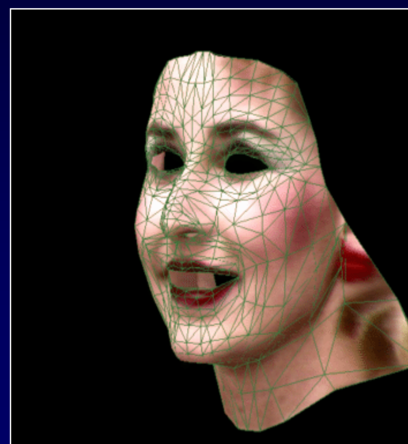




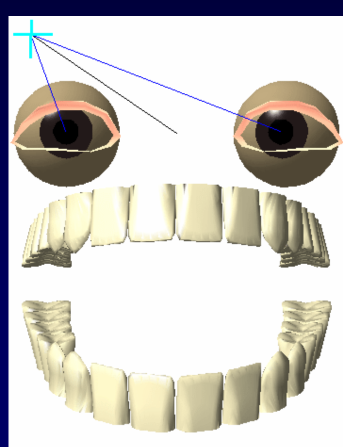
Textured 3D Geometric Model

Texture map coordinates

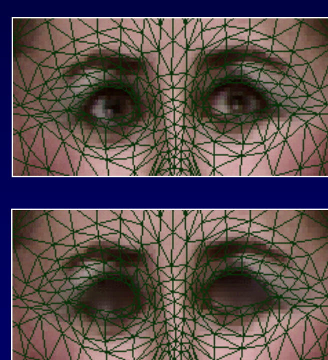
- Positions of fitted mesh nodes in RGB texture image



Auxiliary Geometric Models



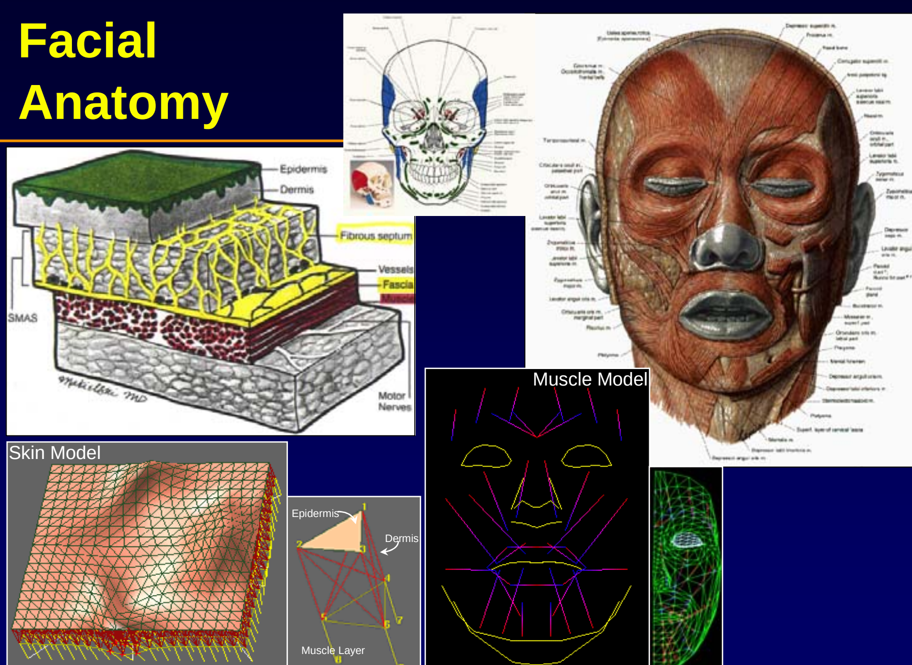
Eyelid Texture Interpolation



Complete Geometric Model

Neutral expression is estimated

Facial Anatomy



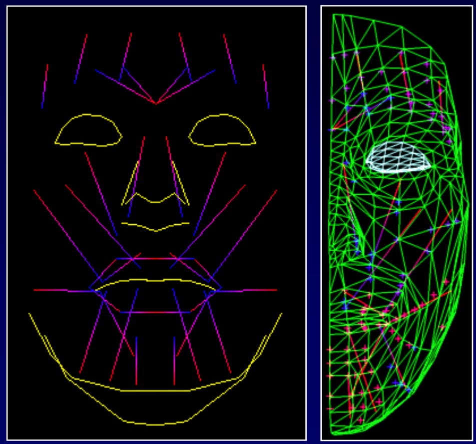
Facial Muscle Model Structure

35 Muscles

- Levator Oculii
- Corrugators
- Naso-Labial
- Zygomatics
- Obicularis Oris

plus

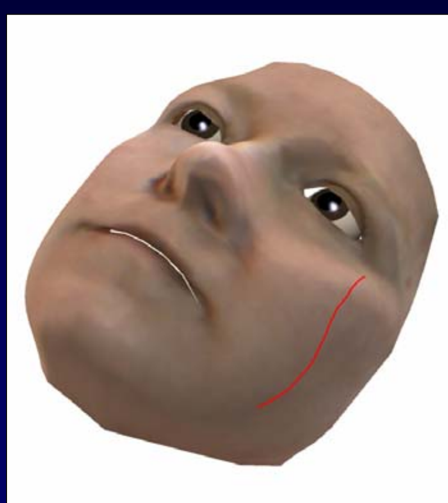
- Articulate Jaw
- Eyes/Eyelids

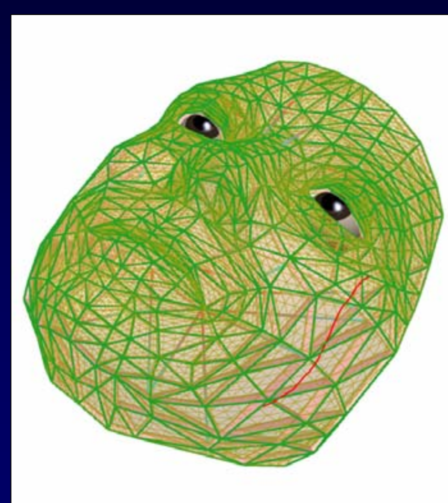


Synthetic Face Animation

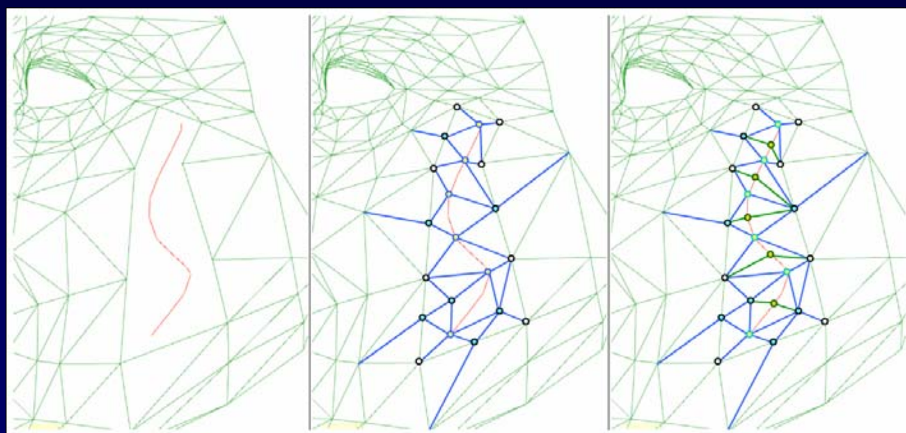
Real-Time Facial Simulation

Incision on Facial Mesh

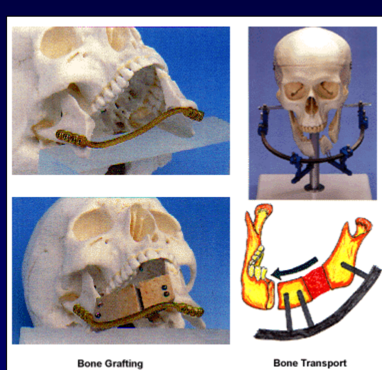


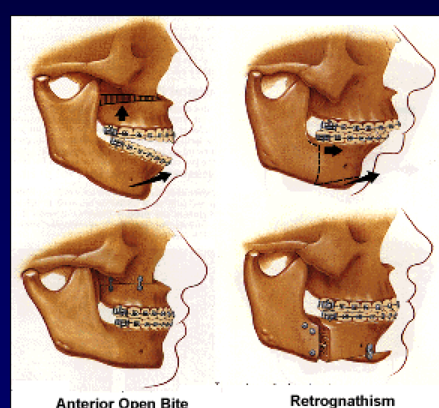


Retriangulation Around Incision



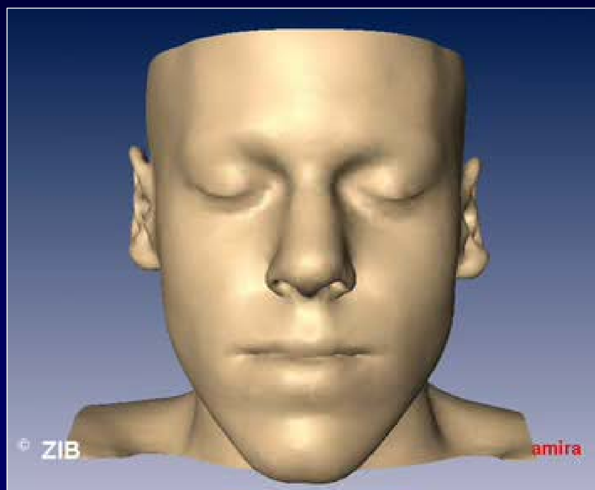
Maxillo Surgery



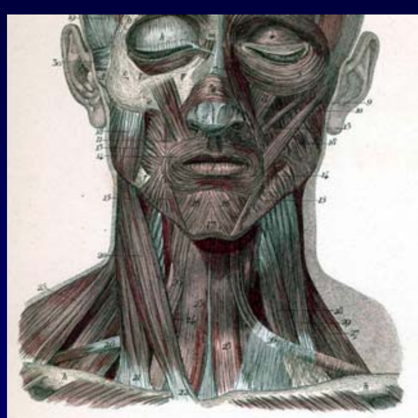


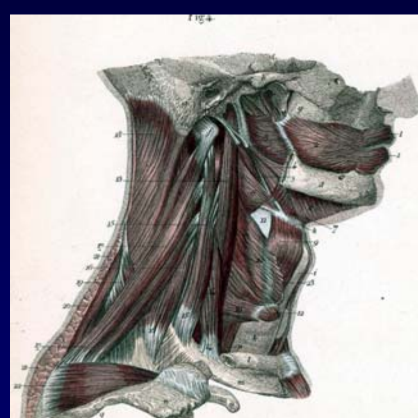
Craniofacial Surgery

[Gladalin, 2002]



Anatomical Structure of the Neck

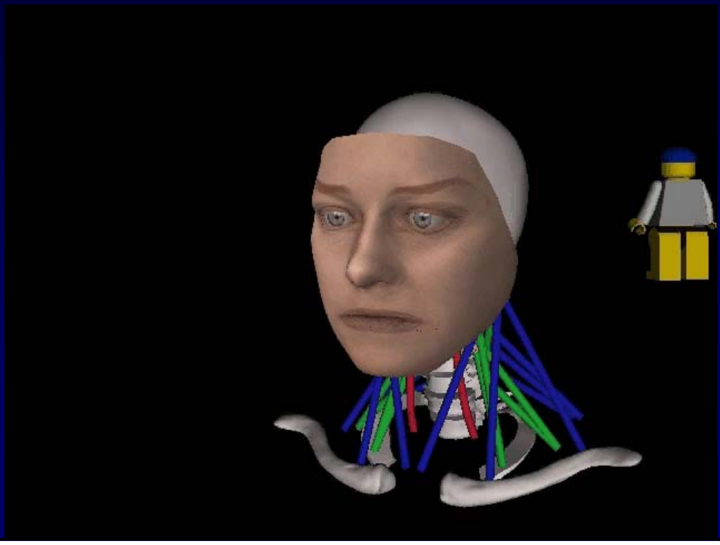




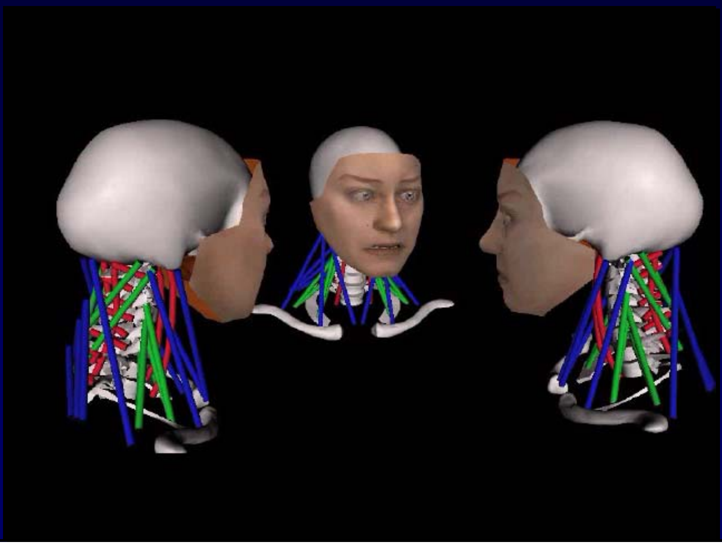
Biomechanical Modeling

What would Leonardo da Vinci Think of This?

Demo:
Gaze Behavior

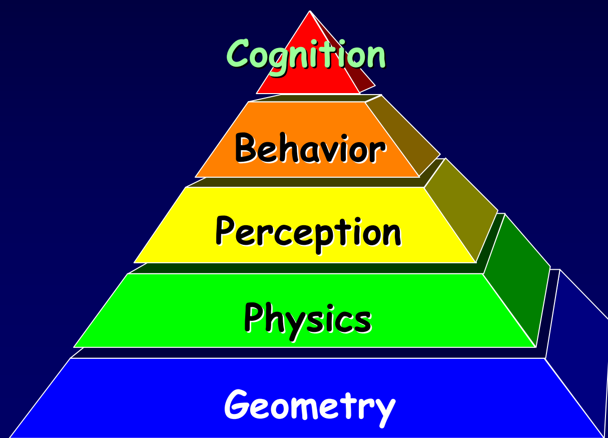


Demo: Autonomous Multi-Head Interaction



Artificial Life Modeling

From physics to intelligence

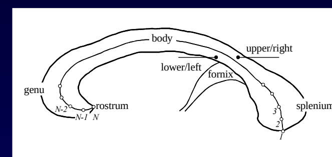


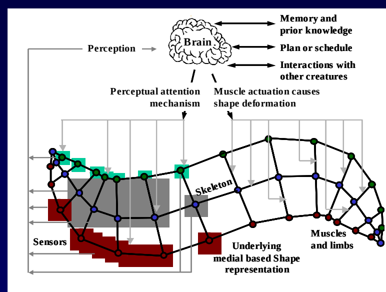
Artificial Fishes

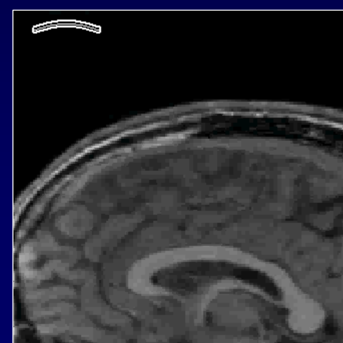
Deformable Organisms

[Hamarneh, McInerney, Terzopoulos, 2001]

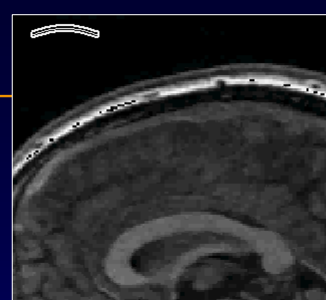
Corpus Callosum Organism

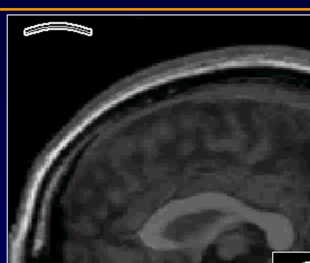


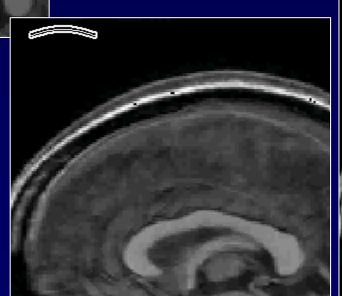


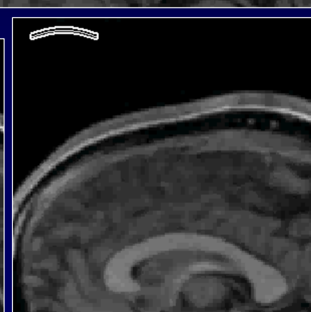


Deformable Organisms

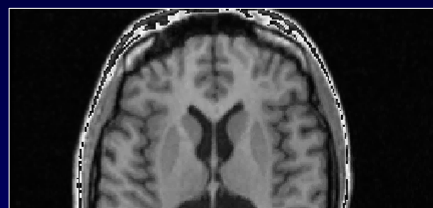








Deformable Organisms



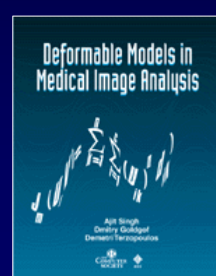
Conclusion

Deformable models

- Powerful technique for extracting geometric models of anatomical structures
- Functional models
- Development continues

“Deformable Models in Medical Image Analysis: A Survey”, *Medical Image Analysis*, **1**(2), 1997

See deformable.com



A Tensor Algebraic Framework for Image Synthesis, Analysis & Recognition

M. Alex O. Vasilescu

MIT Media Laboratory

Demetri Terzopoulos

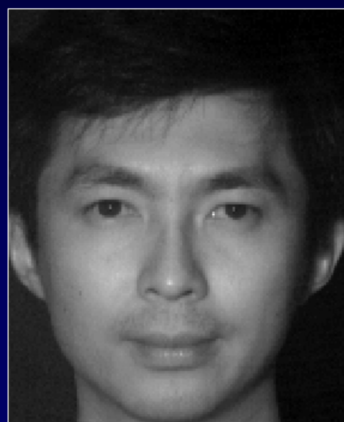
University of California, Los Angeles

Why is Face Recognition Difficult?

Viewpoint changes

Why is Face Recognition Difficult?

Illumination Changes




Appearance-Based Recognition

Recognition of 3D objects (faces) directly from their appearance in ordinary images

- PCA / Eigenimages:
  - *[Sirovich & Kirby 1987]*
    - *"Low Dimensional Procedure for the Characterization of Human Faces"*
  - *[Turk & Pentland 1991]*
    - *"Face Recognition Using Eigenfaces"*
  - *[Murase & Nayar 1995]*
    - *"Visual learning and recognition of 3D objects from appearance"*

Linear Algebra

The algebra of vectors and matrices

- Traditionally of great value in image science
  - *Fourier transform*
  - *Karhunen-Loeve transform*
- Linear methods (PCA, FLD, ICA) model:
  - *Linear operators over a vector space*
  - *Single-factor variation in image formation*
  - *The linear combination of multiple sources*

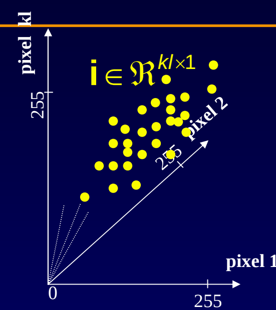
Multilinear Algebra

The algebra of higher-order (>2) tensors

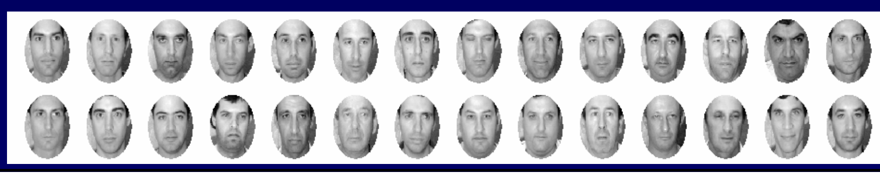
- Natural images result from the interaction of multiple factors related to
  - *scene geometry*
  - *Illumination*
  - *Imaging*
- Multilinear algebra can explicitly represent multifactor variation
  - *Multilinear operators over a* ***set*** *of vector spaces*
- Multilinear algebra subsumes linear algebra as a special case
- A unifying mathematical framework

Images

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

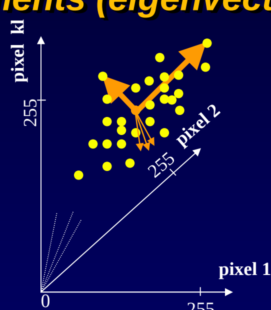


An image is a point in $\Re^{kl \times 1}$ dimensional space



Eigenimages

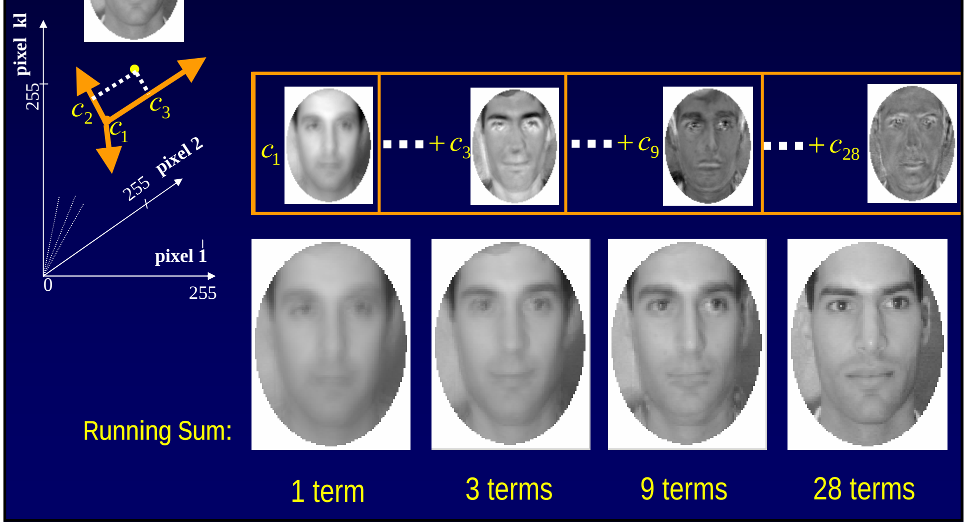
Principal components (eigenvectors) of image ensemble



- Typically computed using the SVD Algorithm

......

Linear Representation



Eigenfaces

- Facial images

- Eigenfaces basis vectors capture the variability in facial appearance

....

- Eigenfaces have been successful in simple facial recognition problem
  - *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 – is allowed to vary

- However, natural images are the consequence of *multiple factors* (or modes) related to scene structure, illumination and imaging

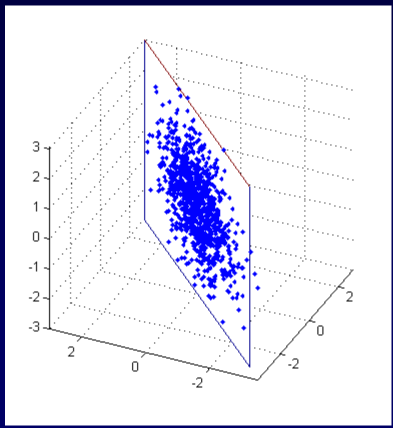
Our Approach

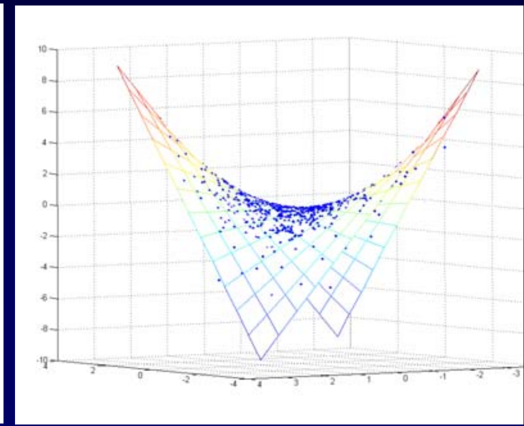
[Vasilescu & Terzopoulos, ECCV 02, ICPR 02, CVPR 03, CVPR 05]

A <u>nonlinear</u> appearance-based technique

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

Linear vs Multilinear Manifolds



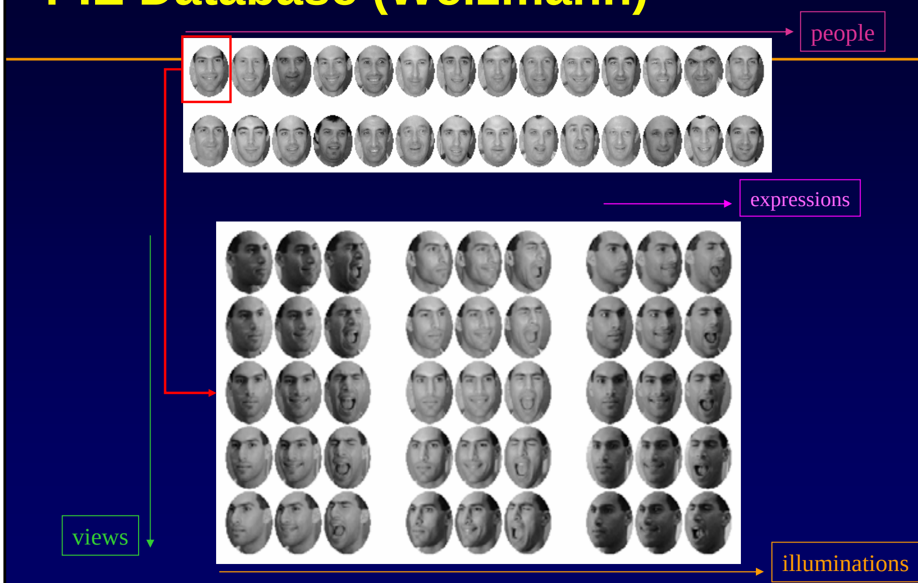


Preliminary Recognition Results

[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%*

PIE Database (Weizmann)



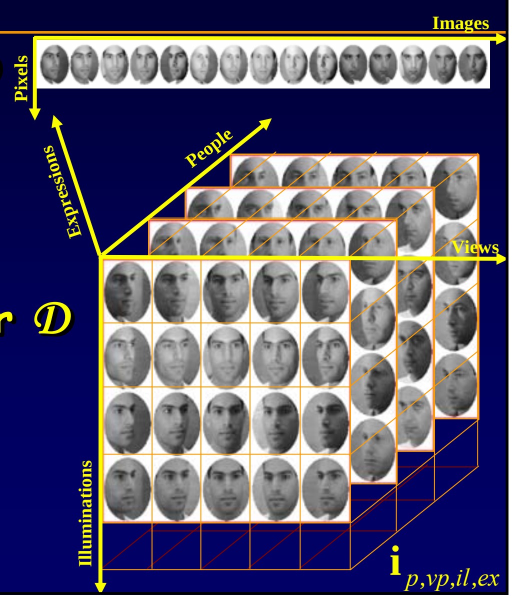
Data Organization

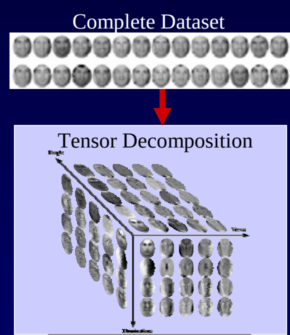
Linear/PCA: Data Matrix $\mathbf{D}$

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

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

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





Learning Stage

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]*

Matrix Decomposition - SVD

- A matrix $\mathbf{D} \in \Re^{I_1 \times I_2}$ 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

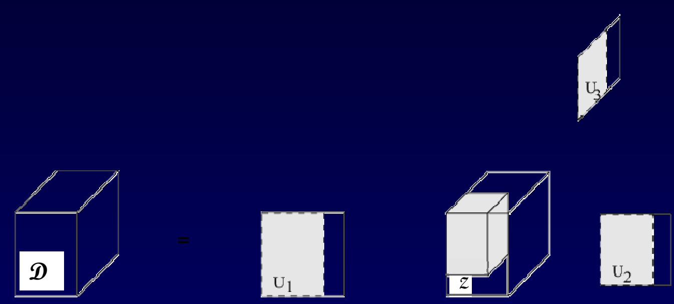
$\mathcal{D}$ is a N-dimensional "matrix", with N spaces

- 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

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

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

Tensor Decomposition



$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$$

$$= \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} \sigma_{r_1 r_2 r_3} \mathbf{u}_{1,r_1} \circ \mathbf{u}_{2,r_2} \circ \mathbf{u}_{3,r_3}$$

$$vec(\mathcal{D}) = (\mathbf{U}_3 \otimes \mathbf{U}_2 \otimes \mathbf{U}_1)\, vec(\mathcal{Z})$$

N-Mode SVD Algorithm

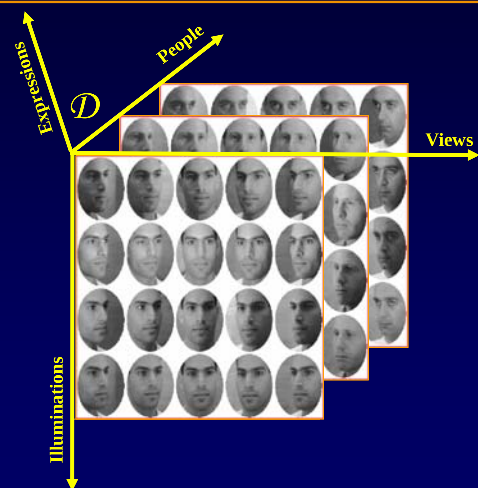
Two steps:

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 as follows

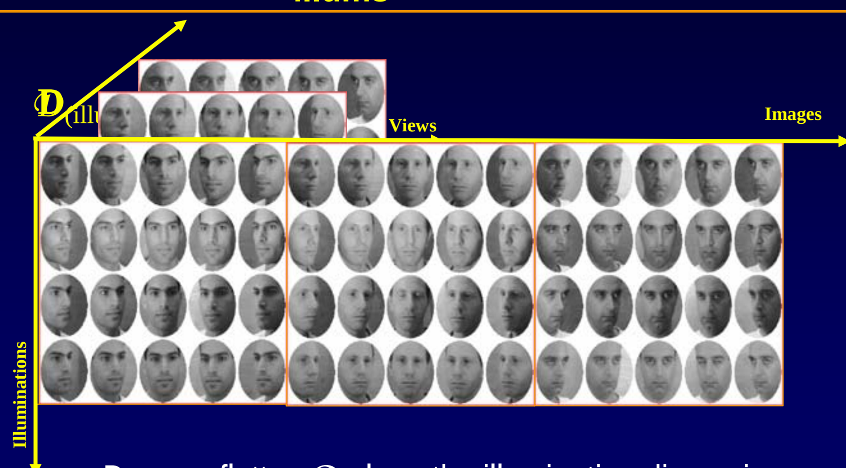
$$\mathcal{Z} = \mathcal{D} \times_1 \mathbf{U}_1^T \times_2 \mathbf{U}_2^T \cdots \times_n \mathbf{U}_n^T \cdots \times_N \mathbf{U}_N^T$$

Facial Data Tensor Decomposition



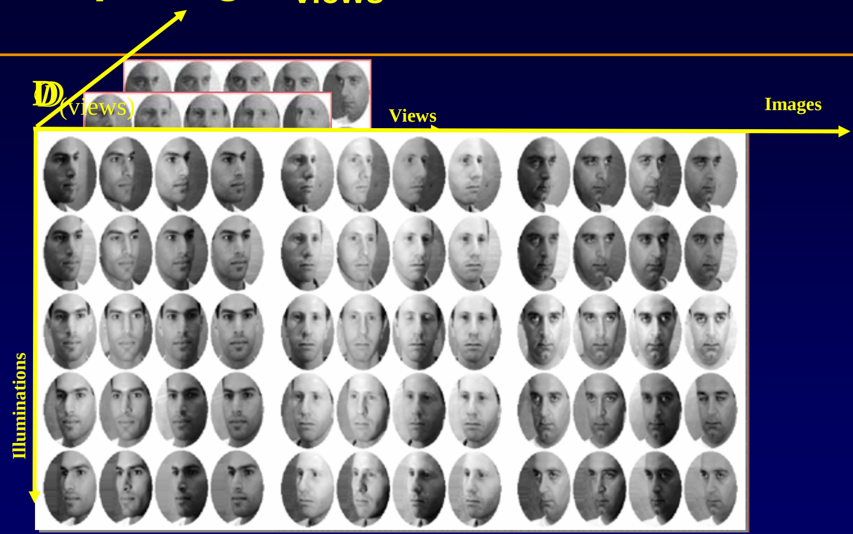
$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{people} \times_2 \mathbf{U}_{views} \times_3 \mathbf{U}_{illums.} \times_4 \mathbf{U}_{express} \times_5 \mathbf{U}_{pixels}$$

Computing U_{illums}



- $D_{(illums)}$ - flatten $\mathcal{D}$ along the illumination dimension
- U_{illums} – orthogonalizes the column space of $D_{(illums)}$

Computing U_{views}



- $D_{(views)}$ – flatten $\mathcal{D}$ along the viewpoint dimension
- U_{views} – orthogonalize the column space of $D_{(views)}$

Computing U_{pixels}

- $D_{(pixels)}$ – flatten $\mathcal{D}$ along the pixel dimension
- U_{pixels} – orthogonal column space of $D_{(pixels)}$
  – eigenimages

Multilinear (Tensor) Algebra

Nth-order tensor $\mathcal{A} \in \Re^{I_1 \times I_2 \times \cdots I_N}$

matrix (2nd-order tensor) $\mathbf{M} \in \Re^{J_n \times I_n}$

mode-*n* product:

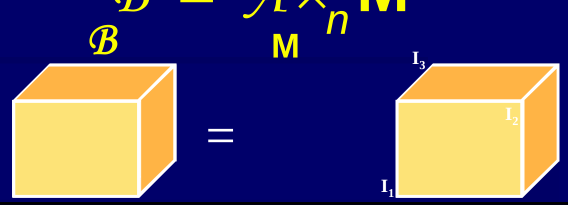
$$\mathcal{B} = \mathcal{A} \times_n \mathbf{M} \quad \text{where} \quad \mathbf{B}_{(n)} = \mathbf{M}\,\mathbf{A}_{(n)}$$

Mode-*n* Product

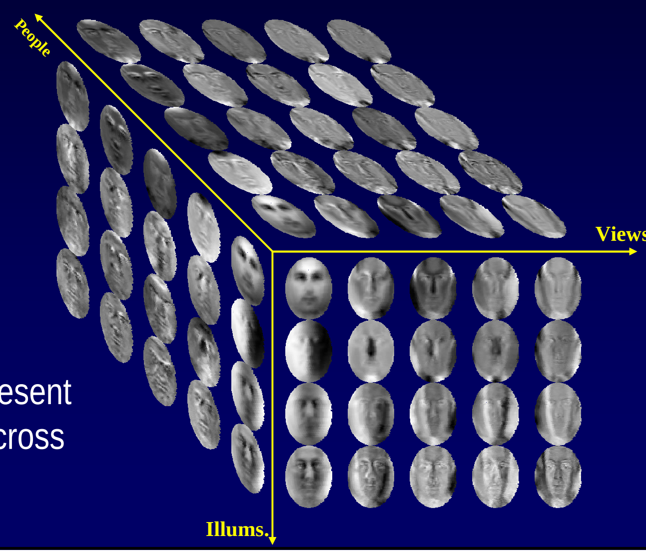
- The mode-n product is a generalization of the product of two matrices
- It is the product of a tensor with a matrix
- Mode-n product of $\mathcal{A} \in \Re^{I_1 \mathrm{x} \ldots \mathrm{x} I_n \mathrm{x} \ldots \mathrm{x} I_N}$ and $\mathbf{M} = \Re^{J_n \mathrm{x} I_n}$
  $\mathcal{B} \in \Re^{I_1 \mathrm{x} \ldots \mathrm{x} I_{n-1} \mathrm{x} J_n \mathrm{x} I_{n+1} \mathrm{x} \ldots \mathrm{x} I_N}$

$$\left(\mathcal{A} \times_n \mathbf{M}\right)_{i_1 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} m_{j_n i_n}$$

$$\mathcal{B} = \mathcal{A} \times_n \mathbf{M}$$



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



TensorFaces:
explicitly represent covariance across factors

TensorFaces Subsume Eigenfaces

Multilinear Analysis / TensorFaces:

$$\mathcal{D} = \mathcal{Z} \times_1 \mathbf{U}_{\text{people}} \times_2 \mathbf{U}_{\text{views}} \times_3 \mathbf{U}_{\text{illums.}} \times_4 \mathbf{U}_{\text{express}} \times_5 \mathbf{U}_{\text{pixels}}$$

Linear Analysis / Eigenfaces:

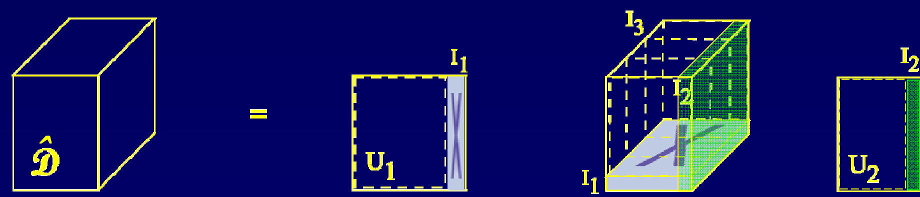
$$\underbrace{\mathbf{D}_{\text{(pixels)}}}_{\text{data matrix}} = \underbrace{\mathbf{U}_{\text{pixels}}}_{\text{basis matrix}} \underbrace{\mathbf{Z}_{\text{(pixels)}} \left(\mathbf{U}_{\text{express}} \otimes \mathbf{U}_{\text{illums.}} \otimes \mathbf{U}_{\text{views}} \otimes \mathbf{U}_{\text{people}} \right)^T}_{\text{coefficient matrix}}$$

Dimensionality Reduction

Iterative dimensionality reduction approach:

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

$$\left\| \mathcal{D} - \hat{\mathcal{D}} \right\|^2 \leq \sum_{i_1=R_1}^{I_1} \sigma_{i_1}^2 + \sum_{i_2=R_2}^{I_2} \sigma_{i_2}^2 \cdots + \sum_{i_N=R_N}^{I_N} \sigma_{i_N}^2$$

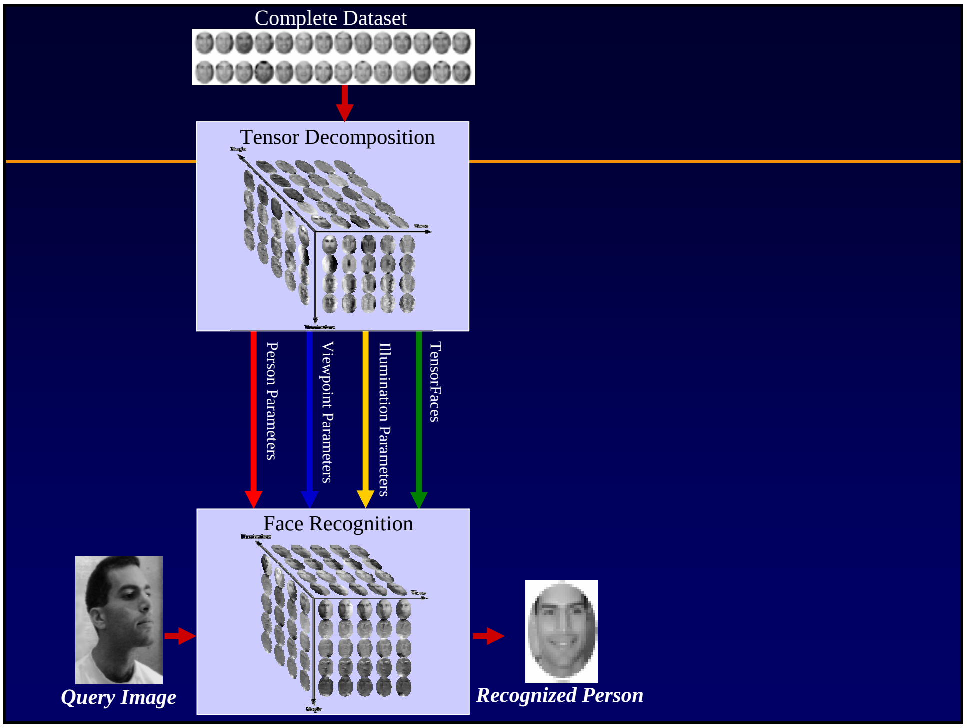


Strategic Data Compression = Perceptual Quality

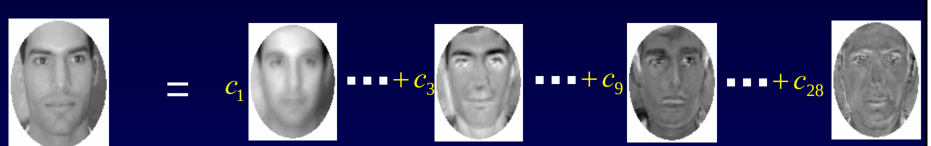
TensorFaces data reduction in illumination space primarily degrades illumination effects (cast shadows, highlights)

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 has *lower mean square error* but *higher perceptual error*



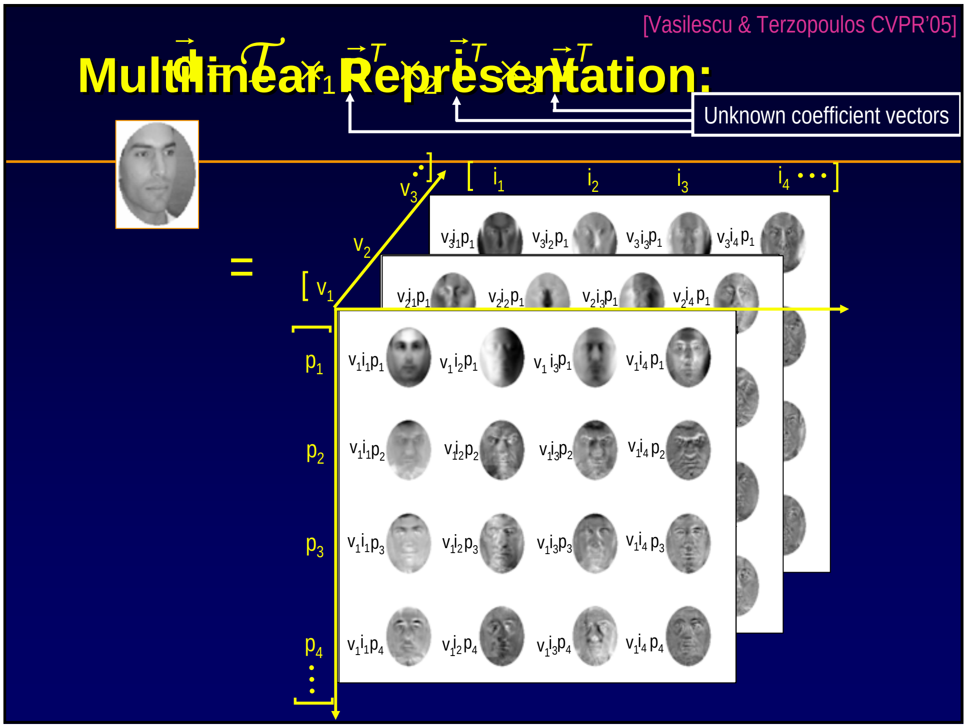
Linear Representation:

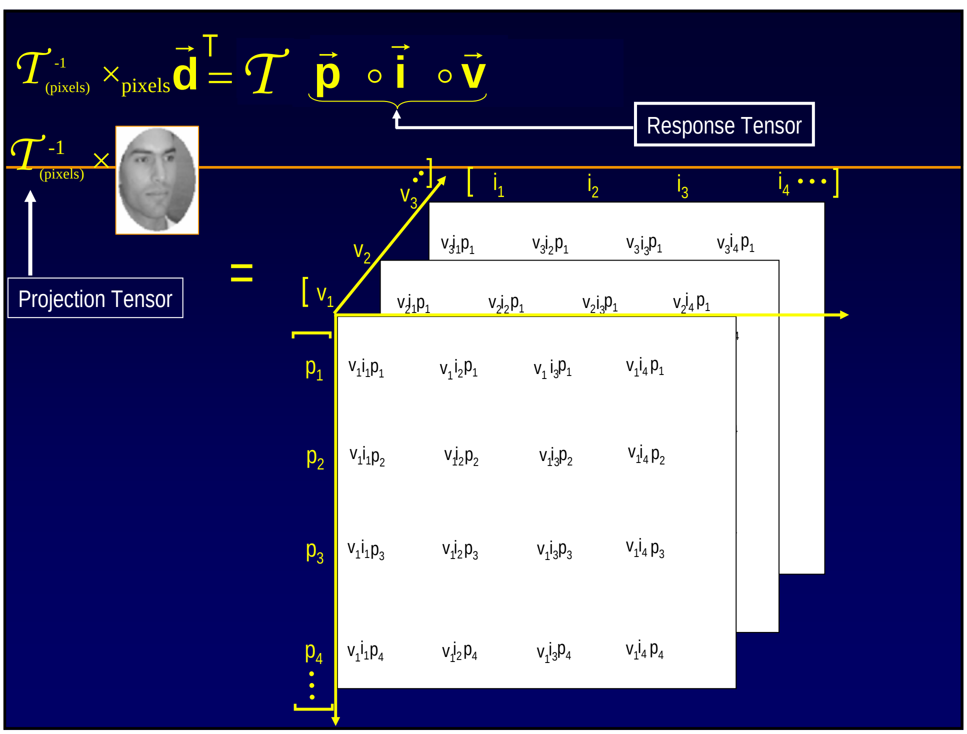


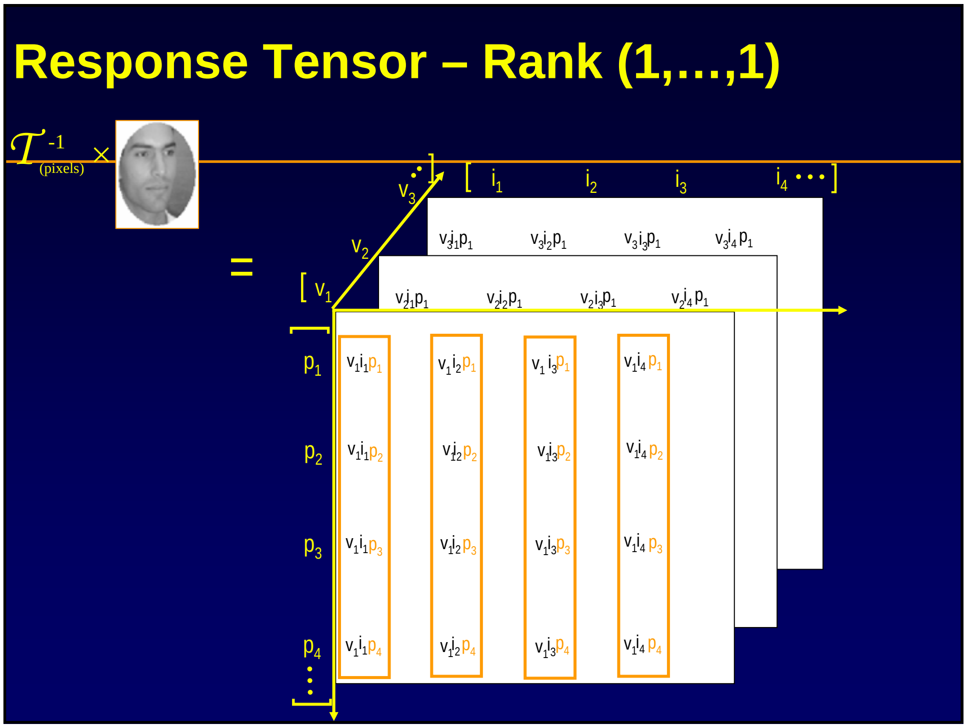
$$\mathbf{U}^T \mathbf{d}_i = \mathbf{U}\mathbf{c}_{ij}$$

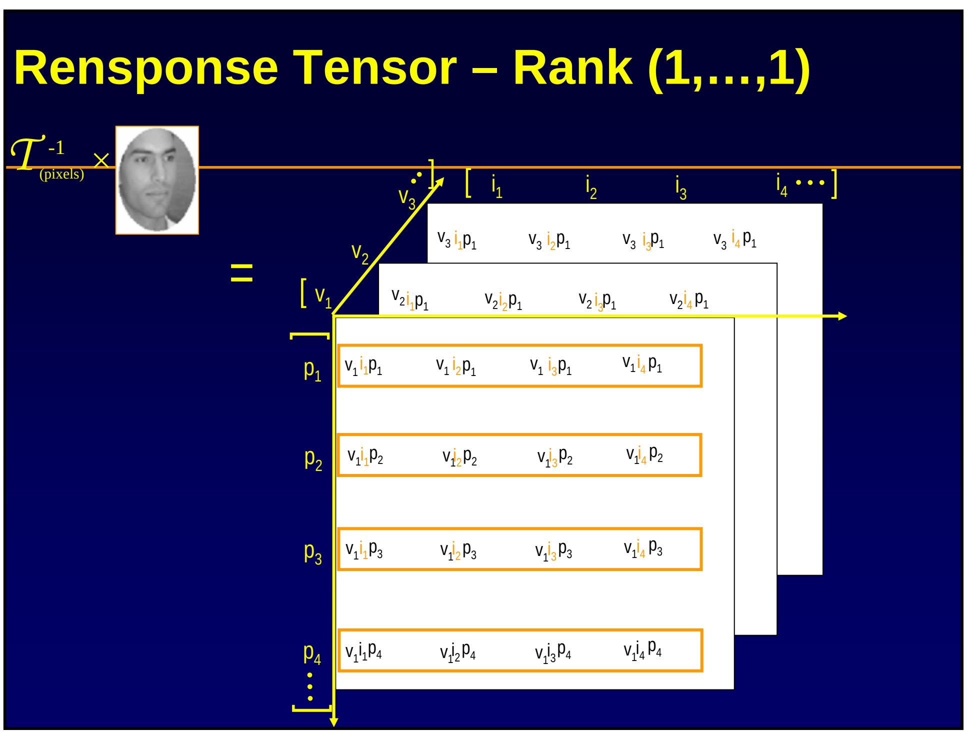
Unknown coefficient vector

Projection Operator

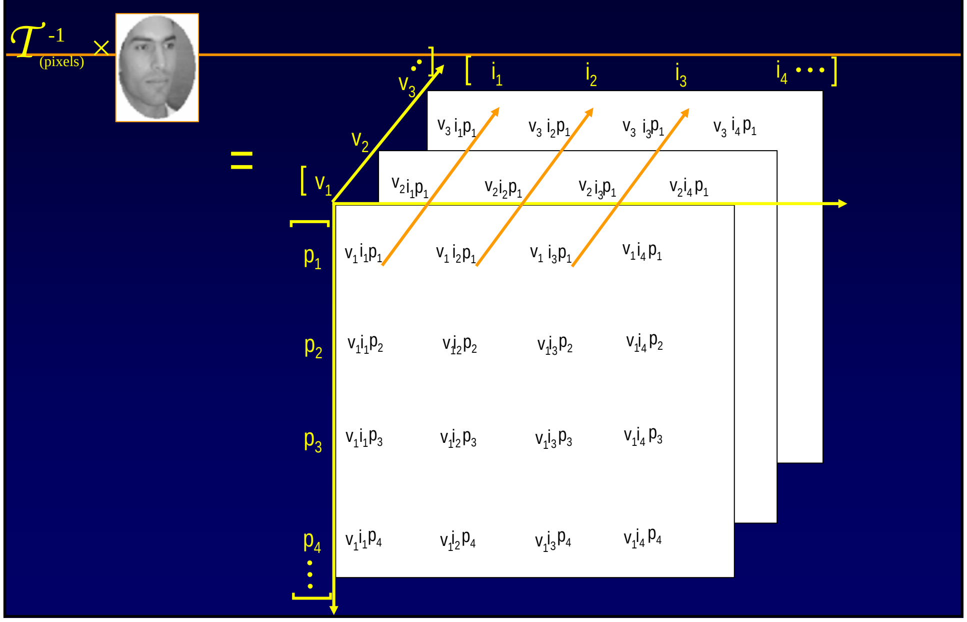

[Vasilescu & Terzopoulos CVPR'05]
Multilinear Representation:
Unknown coefficient vectors
=
[v1
v2
v3
[i1 i2 i3 i4 ...]
p1
p2
p3
p4
v1i1p1 v1i2p1 v1i3p1 v1i4p1
v1i1p2 v1i2p2 v1i3p2 v1i4p2
v1i1p3 v1i2p3 v1i3p3 v1i4p3
v1i1p4 v1i2p4 v1i3p4 v1i4p4
v2i1p1 v2i2p1 v2i3p1 v2i4p1
v3i1p1 v3i2p1 v3i3p1 v3i4p1


$\mathcal{T}^{-1}_{(pixels)} \times_{pixels} \vec{\mathbf{d}}^T = \mathcal{T} \; \vec{\mathbf{p}} \circ \vec{\mathbf{i}} \circ \vec{\mathbf{v}}$
Response Tensor
$\mathcal{T}^{-1}_{(pixels)} \times$
Projection Tensor
=
[v1
v2
v3
[i1 i2 i3 i4 ...]
p1
p2
p3
p4
v1i1p1 v1i2p1 v1i3p1 v1i4p1
v1i1p2 v1i2p2 v1i3p2 v1i4p2
v1i1p3 v1i2p3 v1i3p3 v1i4p3
v1i1p4 v1i2p4 v1i3p4 v1i4p4
v2i1p1 v2i2p1 v2i3p1 v2i4p1
v3i1p1 v3i2p1 v3i3p1 v3i4p1


Response Tensor – Rank (1,…,1)
-1
(pixels)
×
=
[v1
v2
v3
[i1
i2
i3
i4 …]
p1
p2
p3
p4


Rensponse Tensor – Rank (1,…,1)
-1
(pixels)
×
=
[v1
v2
v3
[i1
i2
i3
i4 …]
p1
p2
p3
p4

Response Tensor – Rank (1,…,1)



Multilinear Projection

1. Compute the Projection Tensor:

$$\mathbf{P}_{(mode)} = \mathbf{T}^{+}_{(mode)}$$

2. Compute the Response Tensor:

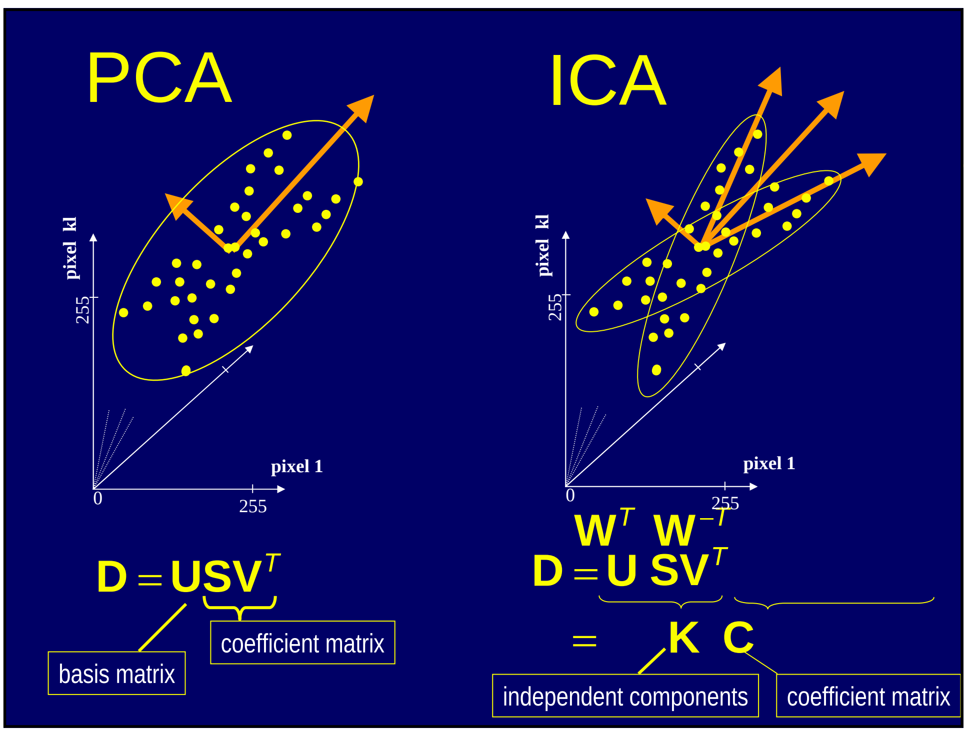
$$\mathcal{R} = \mathcal{P} \times_{mode} \mathbf{d}^T$$

3. Extract the coefficient vectors by factorizing the Response Tensor using the N-mode SVD algorithm

Perspective on Multilinear Models

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

[Vasilescu & Terzopoulos, Learning 2004]



N-Mode ICA

[Vasilescu & Terzopoulos, CVPR 2005]

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. Compute $\mathbf{W}_n^T$ using ICA. Our new mode matrix is $\mathbf{K}_n$

$$\mathbf{D}_{(n)} = \mathbf{U}_n \mathbf{Z}_{(n)} \mathbf{V}_n^T = \underbrace{\left(\mathbf{U}_n \mathbf{W}_n^T\right)}_{} \mathbf{W}_n^{-T} \mathbf{Z}_{(n)} \mathbf{V}_n^T$$

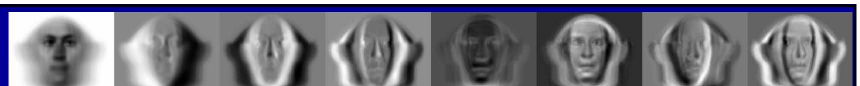
$$= \mathbf{K}_n \mathbf{W}_n^{-T} \mathbf{Z}_{(n)} \mathbf{V}_n^T$$

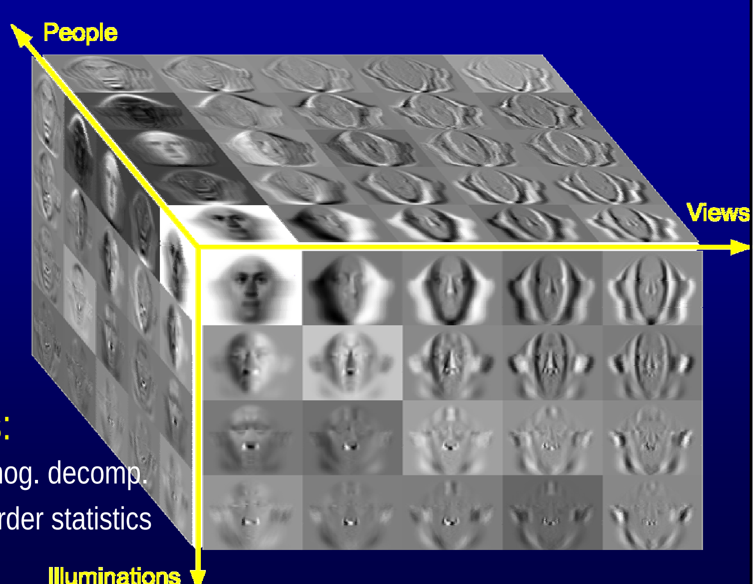
2. Solve for the core tensor as follows

$$\mathcal{S} = \mathcal{D} \times_1 \mathbf{K}_1^{-1} \times_2 \mathbf{K}_2^{-1} \times \cdots \times_n \mathbf{K}_n^{-1} \times \cdots \times_N \mathbf{K}_N^{-1}$$

$$\mathcal{S} = \mathcal{Z} \times_1 \mathbf{W}_1^{-T} \times_2 \mathbf{W}_2^{-T} \times \cdots \times_n \mathbf{W}_n^{-T} \times \cdots \times_N \mathbf{W}_N^{-T}$$

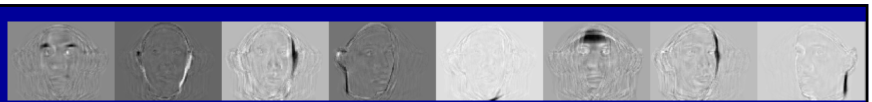
PCA:

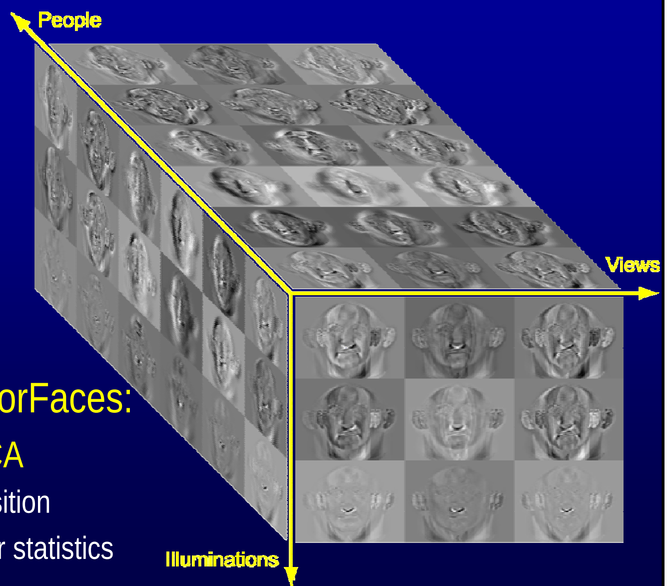




TensorFaces:

- Multilinear orthog. decomp.
- Encodes 2nd order statistics

ICA: 

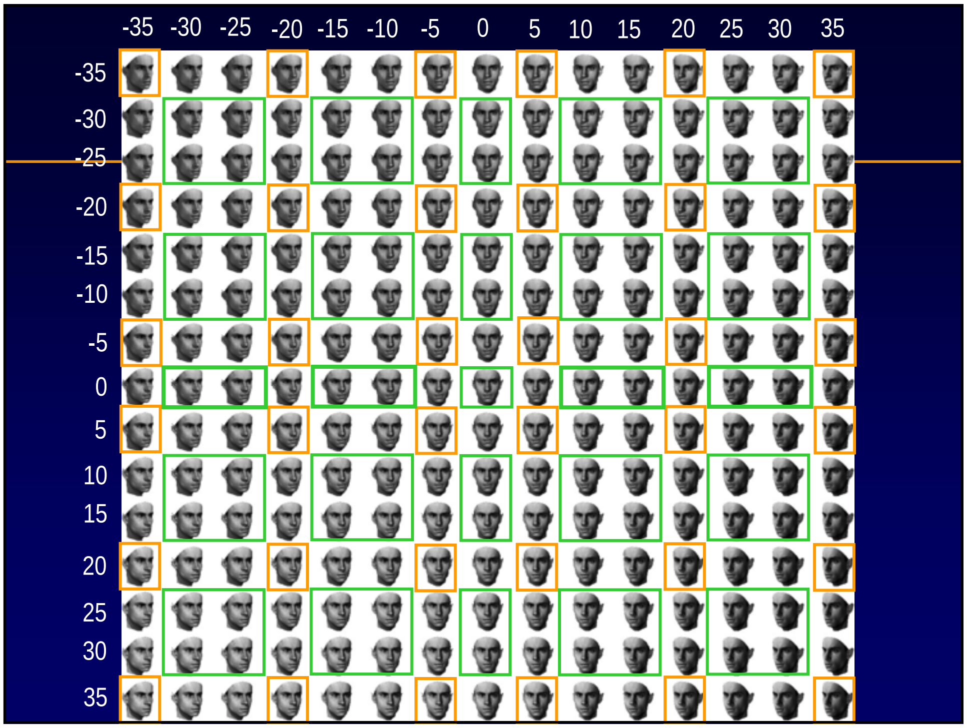


Independent TensorFaces:

Multilinear ICA

- Multilinear decomposition
- Encodes higher order statistics

Freiburg U. 3D-Morphable Data



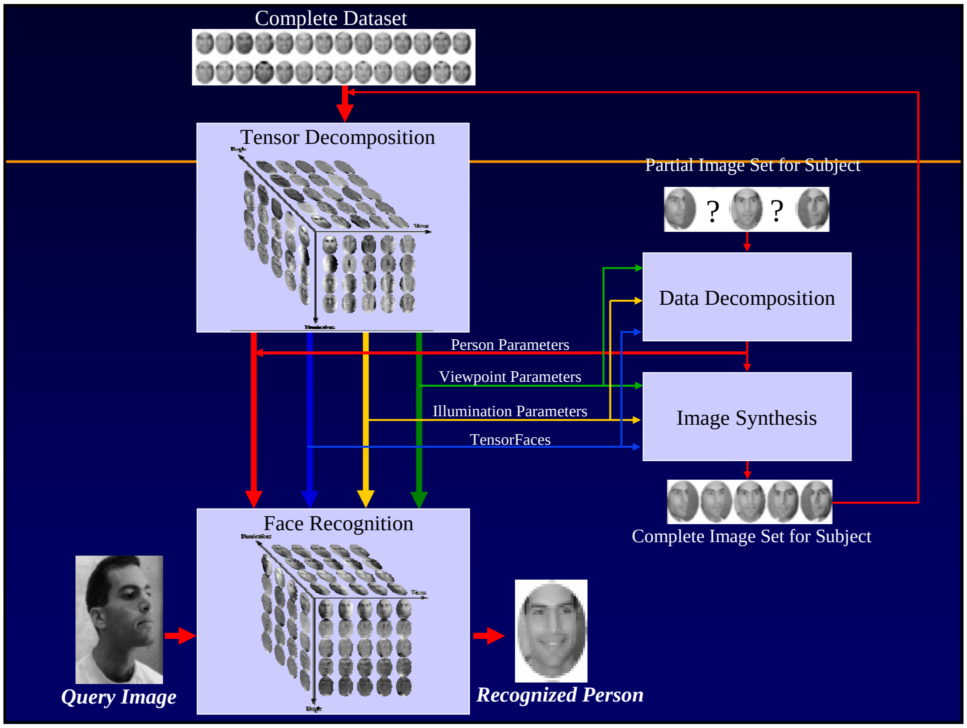
Results

Data Set - 16,875 images
- 75 people
- 15 viewpoints
- 15 illuminations

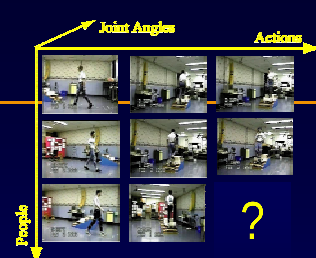
Training Images - 2,700
- 75 people
- 6 viewpoints
- 6 illuminations

Test Images:
- 75 people
- 9 viewpoints
- 9 illums

Linear Models		Multilinear Models	
PCA	ICA	TensorFaces	Independent TensorFaces
83%	89%	93%	97%



Other Applications



- Human Motion Signatures
  - *3-Mode Decomposition, Recognition, & Synthesis*

    [*Vasilescu ICPR 02, CVPR 01, SIGGRAPH 01*]

- Multilinear Image-Based Rendering

  [Vasilescu & Terzopoulos, SIGGRAPH 04]

Multilinear Image-Based Rendering

IBR: Rendering based on sparse samples of object appearance (images)

[Gortler et al. 1996, Levoy & Hanrahan 1996, …]

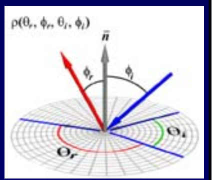
- Surface appearance is determined by the complex interaction of multiple factors:
  - *Scene geometry*
  - *Illumination*
  - *Imaging*

Bidirectional Texture Function

BTF: Captures the appearance of extended textured surfaces with

- *Spatially varying reflectance*
- *Surface mesostructure (3D texture)*
- *Subsurface scattering*
- *Etc.*

- Generalization of BRDF, which accounts only for surface microstructure at a point



BTF

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

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

position on surface (texel)

view direction

illumination direction

photometric angles

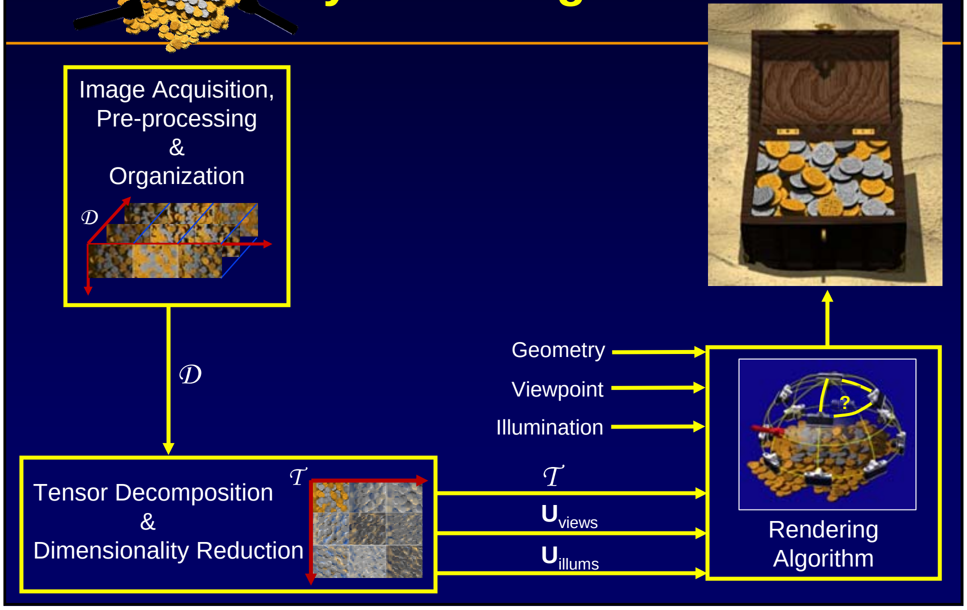
- The BTF captures shading and mesostructural self-shadowing, self-occlusion, interreflection, subsurface scattering

BTF Texture Mapping

[Dana et al. 1999]

	Concrete	*Pebbles*	*Plaster*
Standard Texture Mapping			
BTF Texture Mapping			

System Diagram



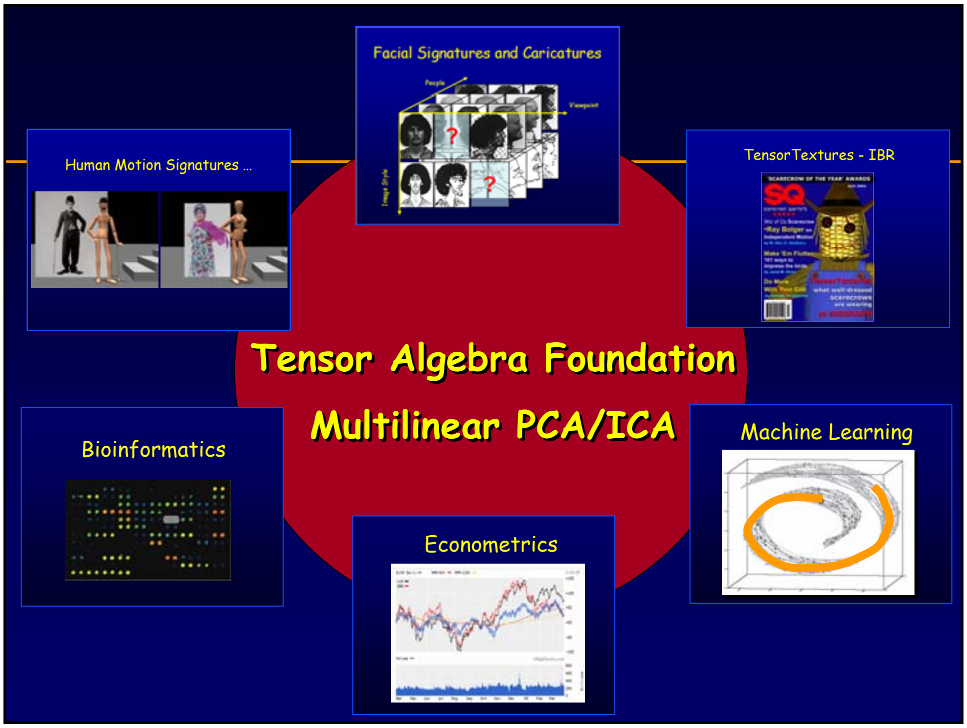
TensorTextures:
Multilinear Image-Based Rendering

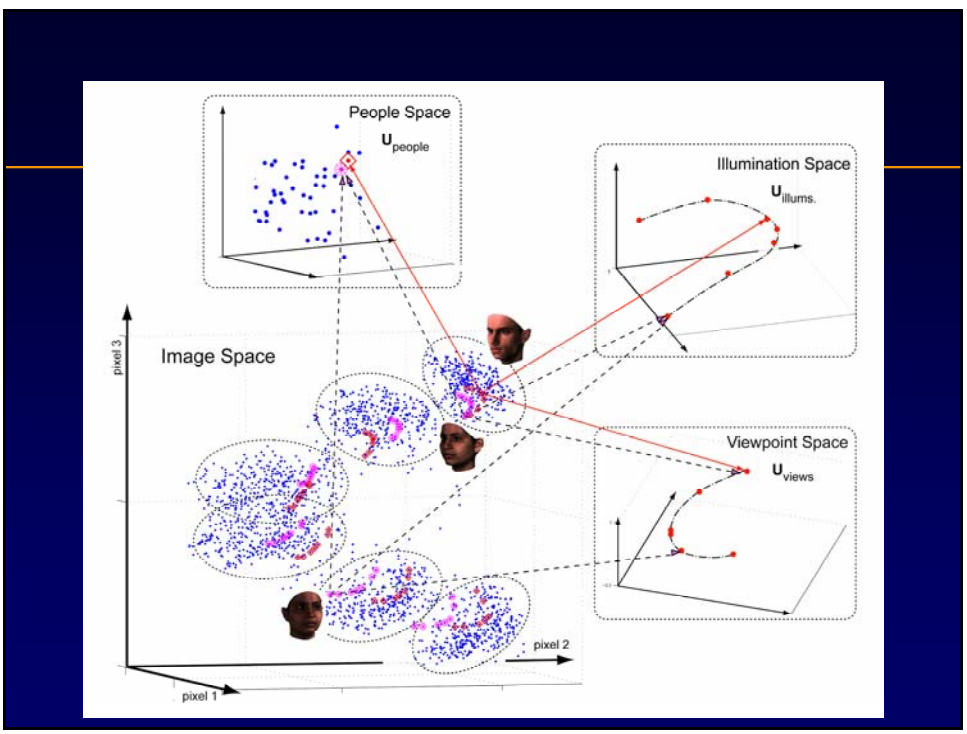
Rendered Texture for a Planar Surface

Conclusion

Multilinear algebraic framework for computer vision and computer graphics

- Tensor approach to the analysis and synthesis of image ensembles
  - *TensorFaces and TensorTextures*
  - *Multilinear PCA and ICA*
- Potentially of interest in ***all*** multifactor problems in vision and graphics to which PCA has been applied; e.g:
  - *Deformable models – Active appearance models [Cootes and Taylor]*
  - *Morphable face models [Blanz and Vetter]*
  - *Precomputed dynamics [James and Fatahalian]*
- Applications in many other fields of science


Facial Signatures and Caricatures
Human Motion Signatures ...
TensorTextures - IBR
Tensor Algebra Foundation
Multilinear PCA/ICA
Bioinformatics
Machine Learning
Econometrics


People Space
U_{people}
Illumination Space
$U_{illums.}$
Image Space
Viewpoint Space
U_{views}
pixel 3
pixel 2
pixel 1

Additional Information

www.media.mit.edu/~maov

terzopoulos.com

Acknowledgement:

Funded by the Technical Support Working Group (TSWG) of the US Department of Defense