Shape Representations for Retrieval and Matching

Thomas Funkhouser

COS526 Fall 2006
Princeton University

3D Representations

What properties are required for analysis and retrieval?

Property	Editing	Display	Analysis	Retrieval
Intuitive specification	Yes	No	No	No
Guaranteed continuity	Yes	No	No	No
Guaranteed validity	Yes	No	No	No
Efficient boolean operations	Yes	No	No	No
Efficient rendering	Yes	Yes	No	No
Accurate	Yes	Yes	?	?
Concise	?	?	?	Yes

3D Representations for Retrieval

Some desirable properties

- Quick to compute
- Efficient to match
- Concise to store
- Invariant to transforms
- Insensitive to noise
- Insensitive to topology
- Robust to degeneracies
- Discriminating

3D Representations for Retrieval

Statistical shape descriptors

- Voxels, moments, wavelets, …
- Attributes, histograms, ...

Structural shape descriptors

- Feature-based methods
- Part-based methods
- Skeletons

Statistical Shape Descriptors

Alignment-dependent

- Voxels
- Wavelets
- Moments
- Extended Gaussian Image
- Spherical Extent Function
- Spherical Attribute Image

Alignment-independent

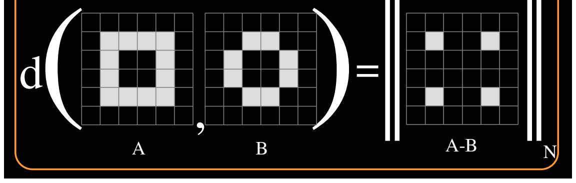
- Shape distributions
- Shape histograms
- Harmonic descriptor

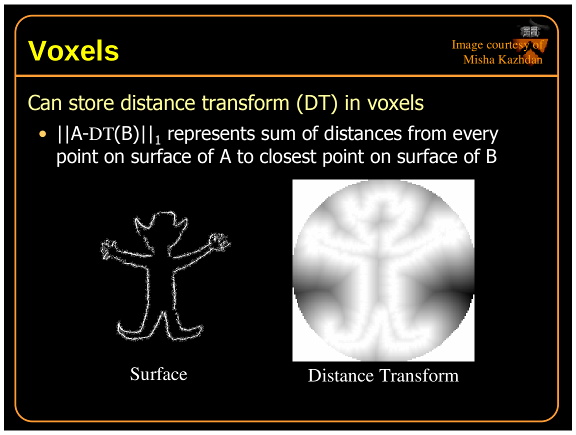
Voxels

Use voxel values as feature vector (shape descriptor)

- Feature space has N^3 dimensions (one dimension for each voxel)
- $d(A,B) = ||A-B||_N$

Example:



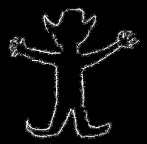


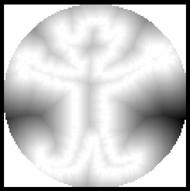
Voxels

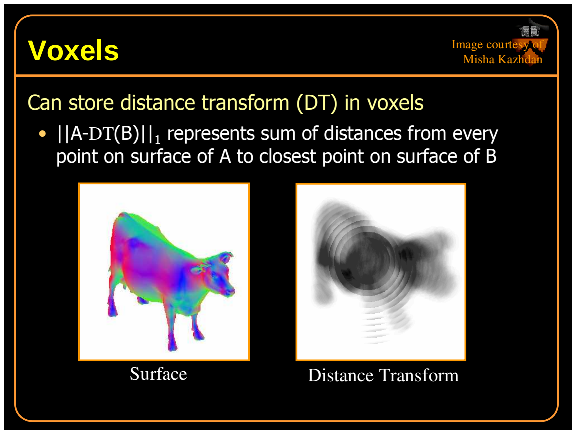
Image courtesy of Misha Kazhdan

Can store distance transform (DT) in voxels

- $||A\text{-}DT(B)||_1$ represents sum of distances from every point on surface of A to closest point on surface of B


Surface


Distance Transform

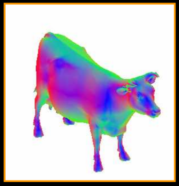


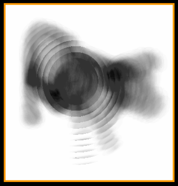
Voxels

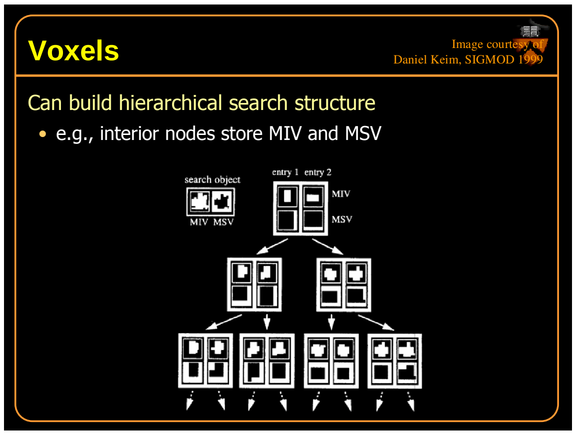
Image courtesy of Misha Kazhdan

Can store distance transform (DT) in voxels

- $||A\text{-}DT(B)||_1$ represents sum of distances from every point on surface of A to closest point on surface of B


Surface


Distance Transform

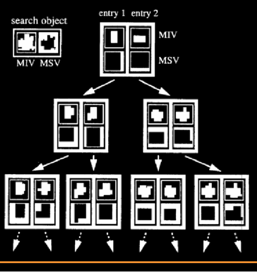


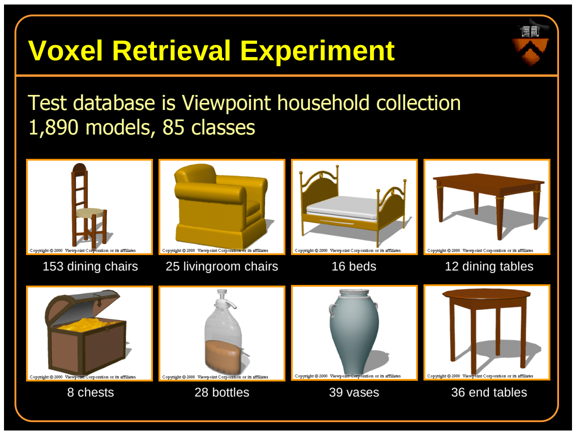
Voxels

Image courtesy of Daniel Keim, SIGMOD 1999

Can build hierarchical search structure

- e.g., interior nodes store MIV and MSV

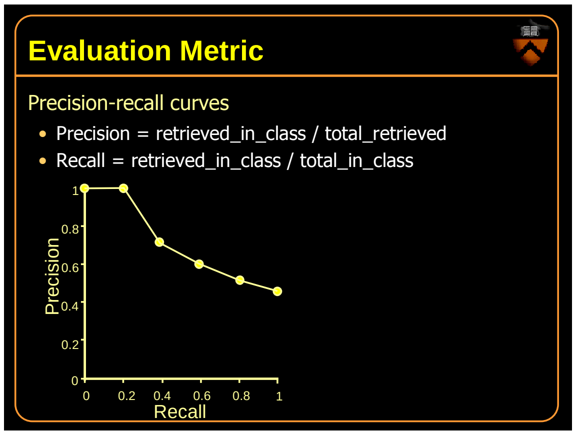




Voxel Retrieval Experiment

Test database is Viewpoint household collection
1,890 models, 85 classes

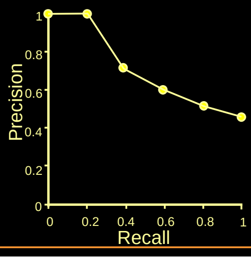
153 dining chairs
25 livingroom chairs
16 beds
12 dining tables
8 chests
28 bottles
39 vases
36 end tables

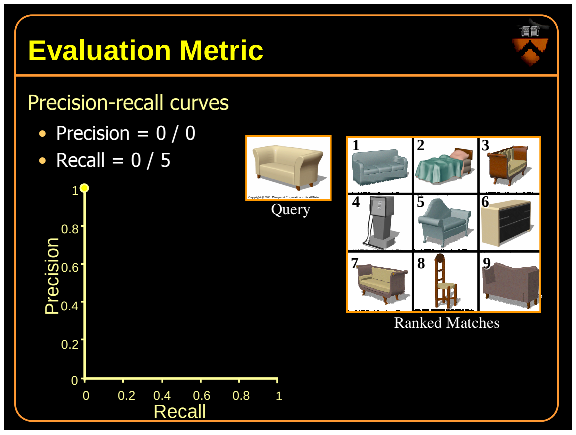


Evaluation Metric

Precision-recall curves

- Precision = retrieved_in_class / total_retrieved
- Recall = retrieved_in_class / total_in_class

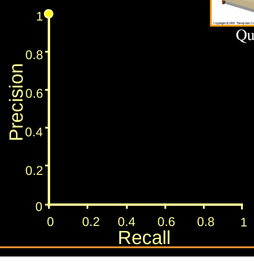


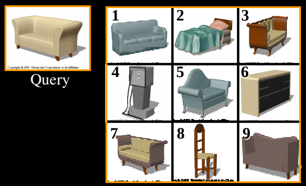


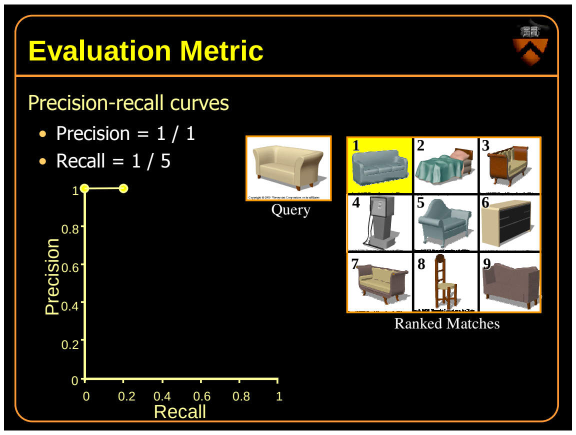
Evaluation Metric

Precision-recall curves

- Precision = 0 / 0
- Recall = 0 / 5



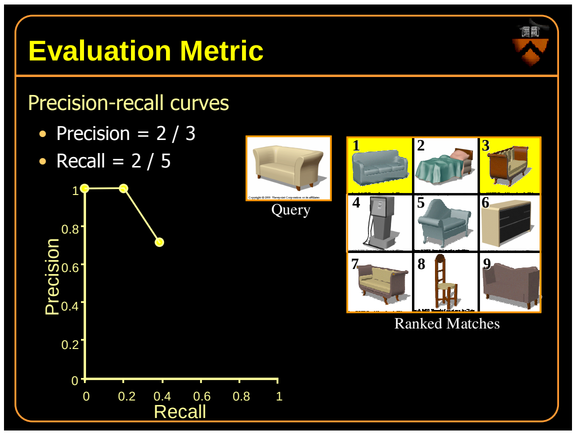




Evaluation Metric

Precision-recall curves

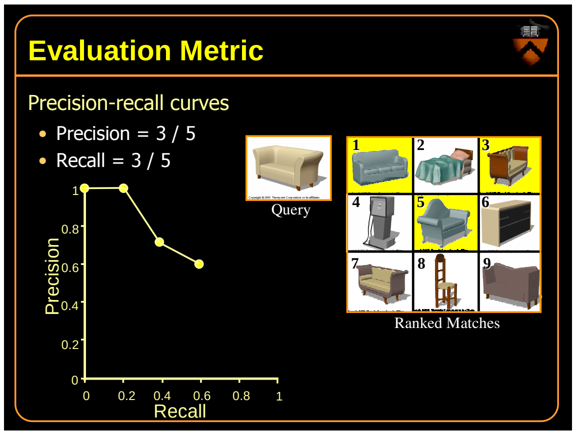
- Precision = 1 / 1
- Recall = 1 / 5



Evaluation Metric

Precision-recall curves

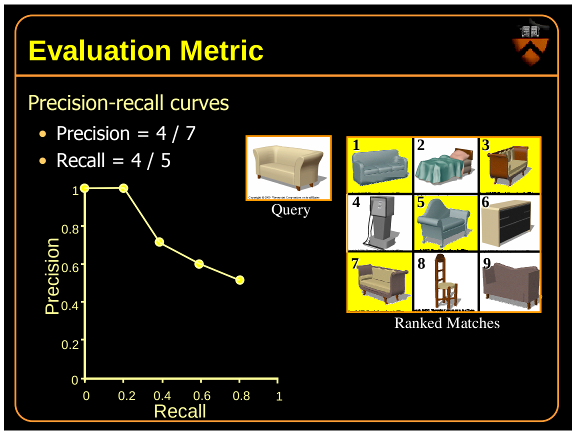
- Precision = 2 / 3
- Recall = 2 / 5



Evaluation Metric

Precision-recall curves

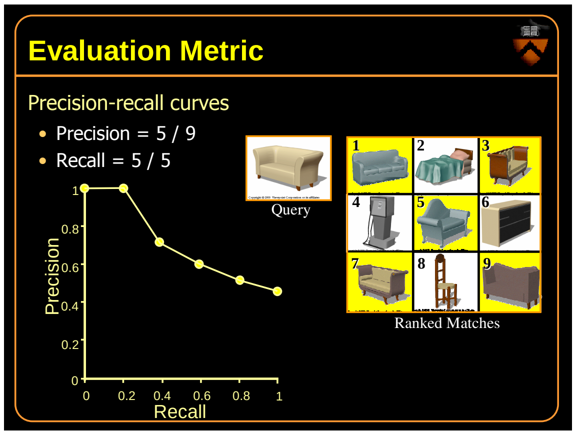
- Precision = 3 / 5
- Recall = 3 / 5



Evaluation Metric

Precision-recall curves

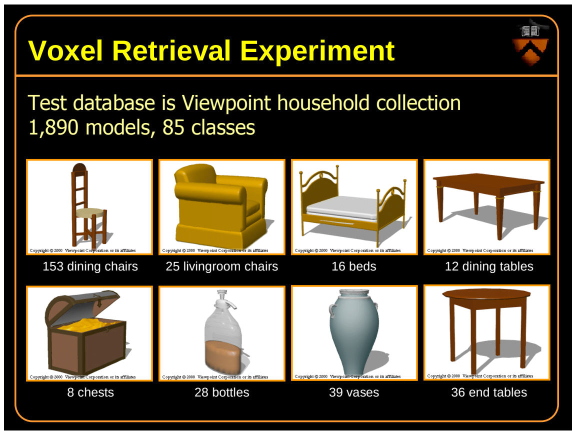
- Precision = 4 / 7
- Recall = 4 / 5



Evaluation Metric

Precision-recall curves

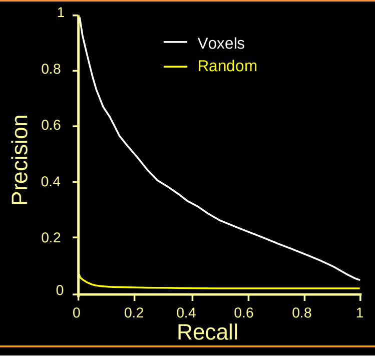
- Precision = 5 / 9
- Recall = 5 / 5



Voxel Retrieval Experiment

Test database is Viewpoint household collection
1,890 models, 85 classes

Voxel Retrieval Results



Voxels

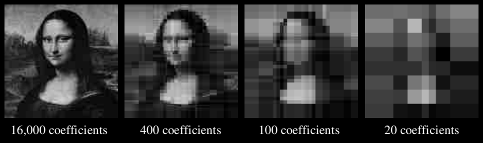
Properties

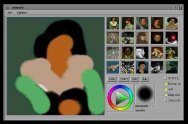
- ü Discriminating
- ü Insensitive to noise
- ü Insensitive to topology
- ü Robust to degeneracies
- ü Quick to compute
- • Efficient to match?
- X Concise to store
- X Invariant to transforms

Wavelets

Image courtesy of
Jacobs, Finkelstein, & Salesin

Define shape with wavelet coefficients



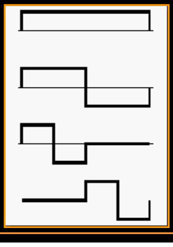


Wavelets

Jacobs, Finkelstein, & Salesin
SIGGRAPH 95

Descriptor 1:

- Given an NxNxN grid, generate an NxNxN array of the wavelet coefficients for the standard Haar basis functions



Wavelets

Jacobs, Finkelstein, & Salesin
SIGGRAPH 95

Descriptor 1:

- Given an NxNxN grid, generate an NxNxN array of the wavelet coefficients for the standard Haar basis functions

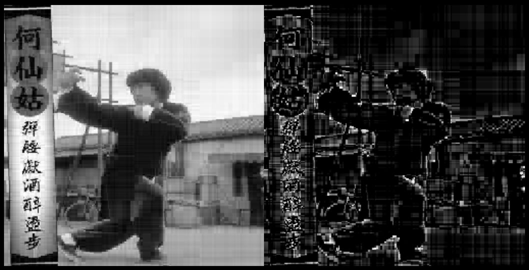
Descriptor 2:

- Truncate: Find the m largest coefficients and set all others equal to zero
- Quantize: Set the non-zero coefficients to +1 or –1 depending on their sign

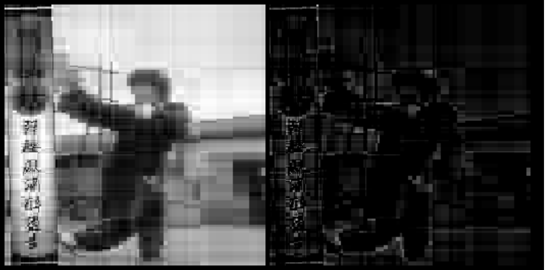
Jackie Chan Example

Original Image (256x256)

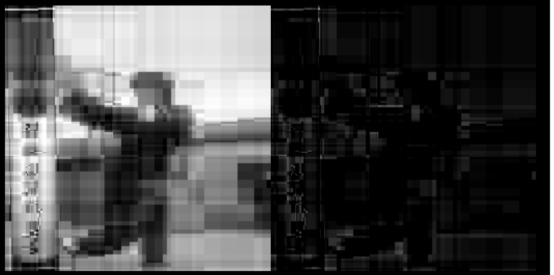
Truncated And Quantized to 5000



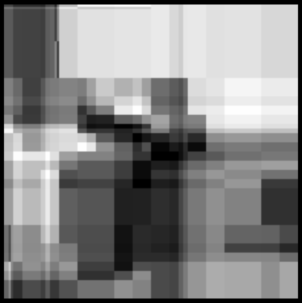
Truncated And Quantized to 1000



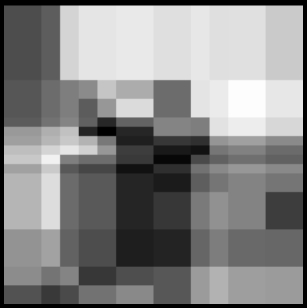
Truncated And Quantized to 500



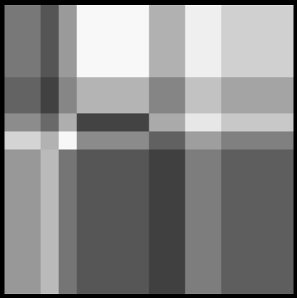
Truncated 100

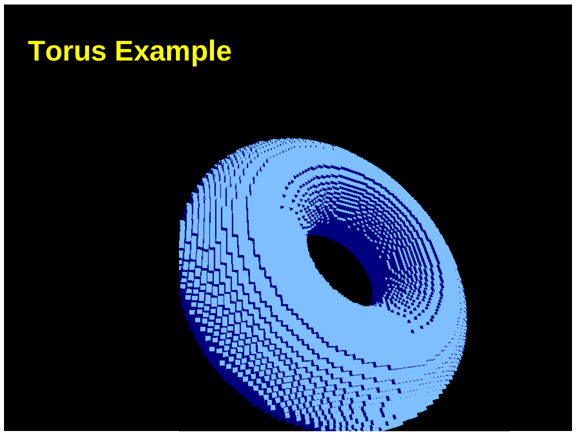


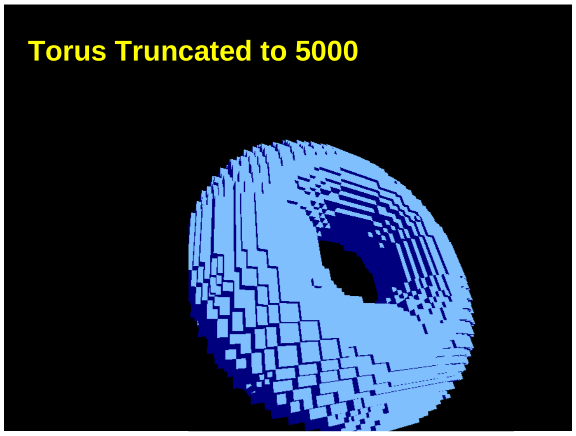
Truncated 50

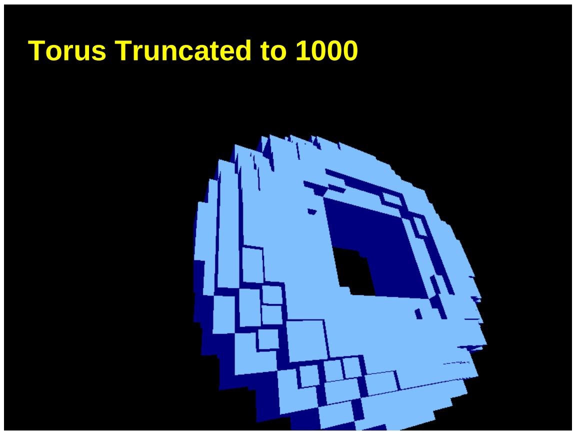


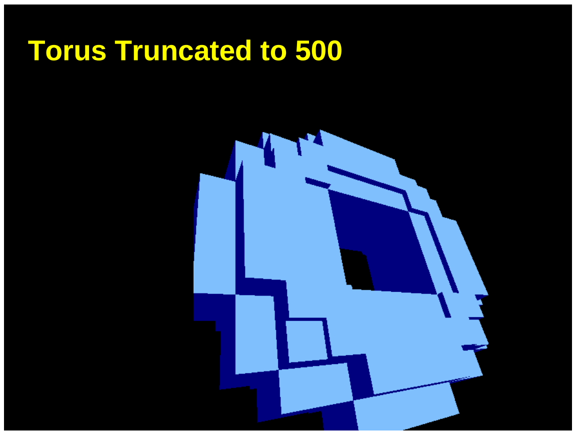
Truncated 10

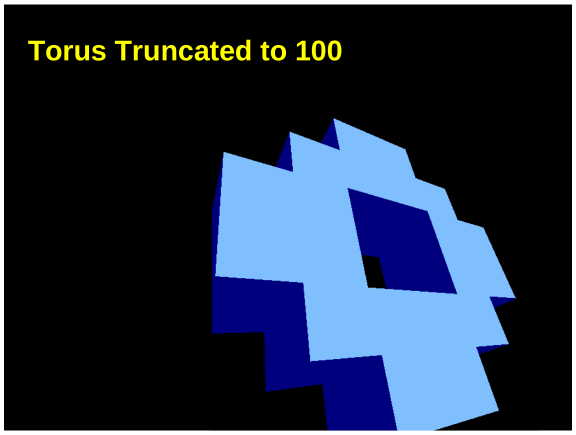


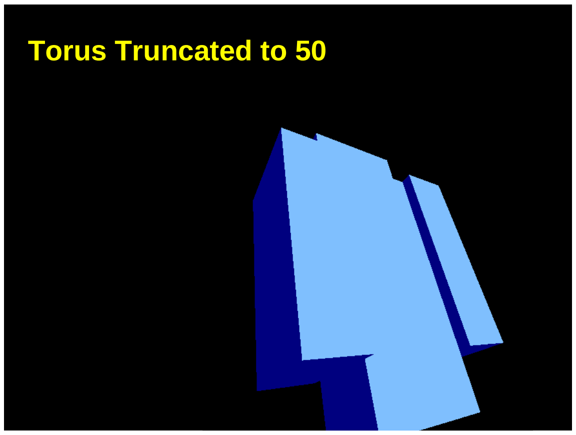

Torus Example


Torus Truncated to 5000


Torus Truncated to 1000


Torus Truncated to 500


Torus Truncated to 100


Torus Truncated to 50

Wavelets

Jacobs, Finkelstein, & Salesin
SIGGRAPH 95

Distance Function 1:

- The query metric is defined by:

$$d(A,B)=\sum_{i,j,k} w_{i,j,k}\left\| A[i,j,k]-B[i,j,k]\right\|$$

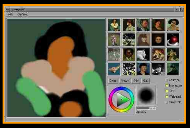
where A[i,j,k] and B[i,j,k] are the truncated and quantized coefficients and $w_{i,j,k}$ are weights, fine tuned to the database.

Wavelets

Jacobs, Finkelstein, & Salesin
SIGGRAPH 95

Properties

ü Insensitive to noise
ü Insensitive to topology
ü Robust to degeneracies
ü Quick to compute
ü Efficient to match
ü Concise to store
• Discriminating?
X Invariant to transforms



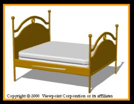
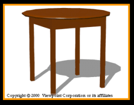
Moments

Define shape by moments of inertia:

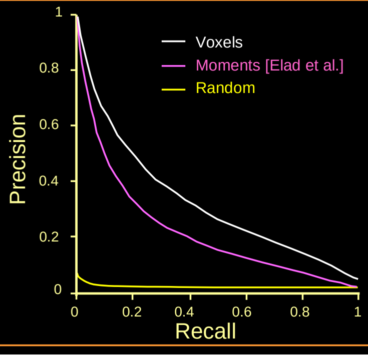
$$m_{pqr}=\int_{surface} x^p y^q z^r \, dxdydz$$

Moments Retrieval Experiment

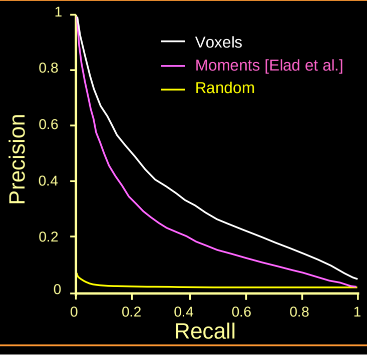
Test database is Viewpoint household collection
1,890 models, 85 classes

153 dining chairs	25 livingroom chairs	 16 beds	12 dining tables
8 chests	28 bottles	39 vases	 36 end tables

Moments Retrieval Results



Moments Retrieval Results



Moments

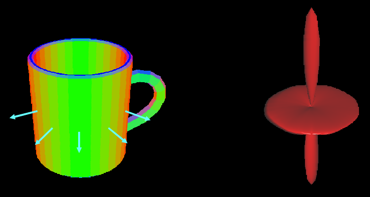
Properties

- ü Insensitive to topology
- ü Robust to degeneracies
- ü Quick to compute
- ü Efficient to match
- ü Concise to store
- X Insensitive to noise
- X Invariant to transforms
- X Discriminating

Extended Gaussian Image

Define shape with histogram of normal directions

- Invertible for convex objects
- Spherical function

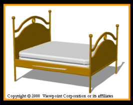


EGI Retrieval Experiment

Test database is Viewpoint household collection
1,890 models, 85 classes

153 dining chairs

25 livingroom chairs

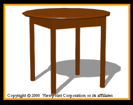

16 beds

12 dining tables

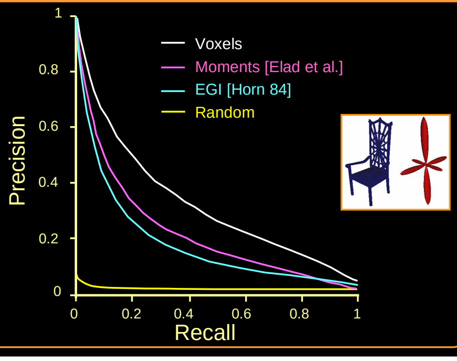
8 chests

28 bottles

39 vases


36 end tables

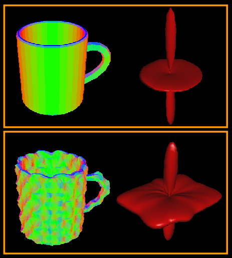
EGI Retrieval Results



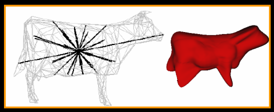
Extended Gaussian Images

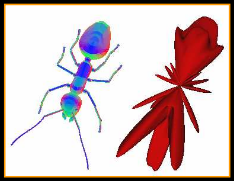
Properties

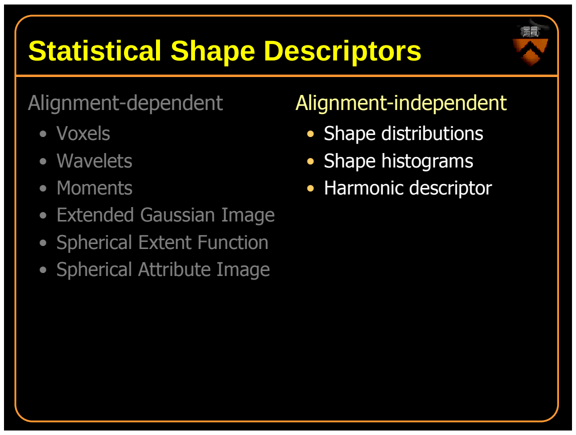
- ü Insensitive to topology
- ü Quick to compute
- ü Efficient to match
- ü Concise to store
- X Insensitve to noise
- X Robust to degeneracies
- X Invariant to transforms
- X Discriminating



Other Rotation-Dependent Descriptors


Spherical Extent Functions
(Vranic & Saupe, 2000)

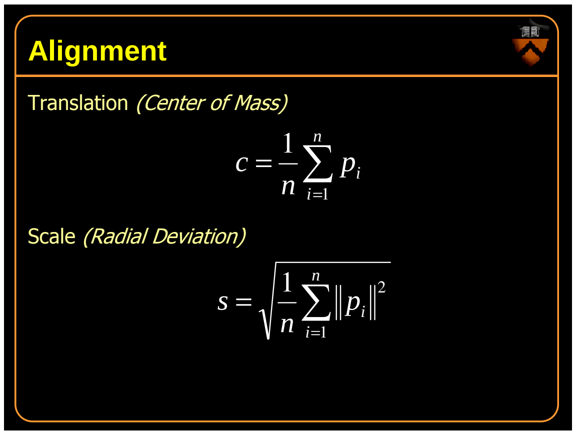

Shape Histograms (sectors)
(Ankherst, 1999)



Statistical Shape Descriptors

Alignment-dependent
- Voxels
- Wavelets
- Moments
- Extended Gaussian Image
- Spherical Extent Function
- Spherical Attribute Image

Alignment-independent
- Shape distributions
- Shape histograms
- Harmonic descriptor



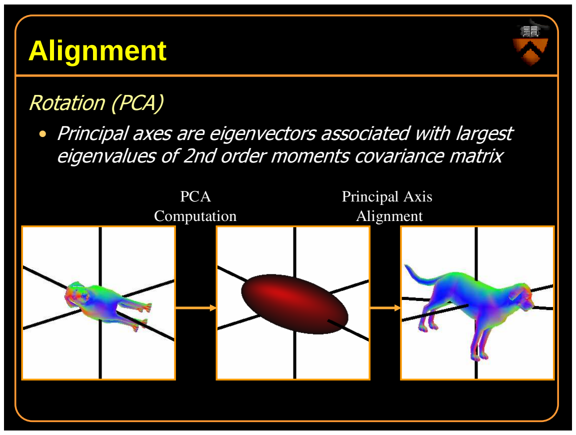
Alignment

Translation *(Center of Mass)*

$$c = \frac{1}{n}\sum_{i=1}^{n} p_i$$

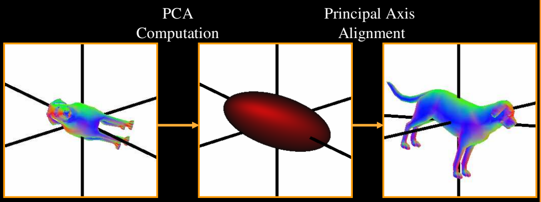
Scale *(Radial Deviation)*

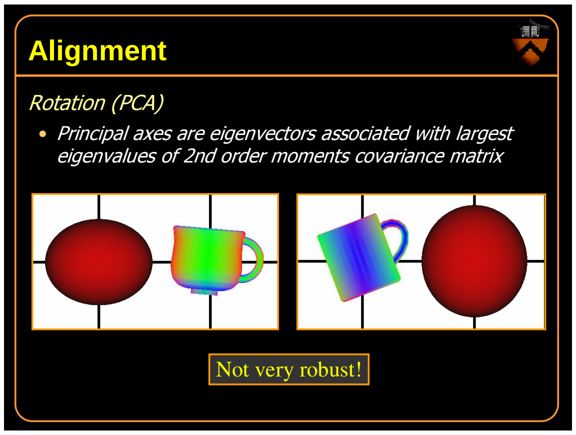
$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \|p_i\|^2}$$



Alignment

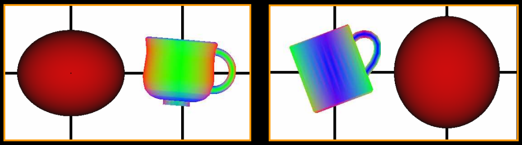
Rotation (PCA)
- *Principal axes are eigenvectors associated with largest eigenvalues of 2nd order moments covariance matrix*



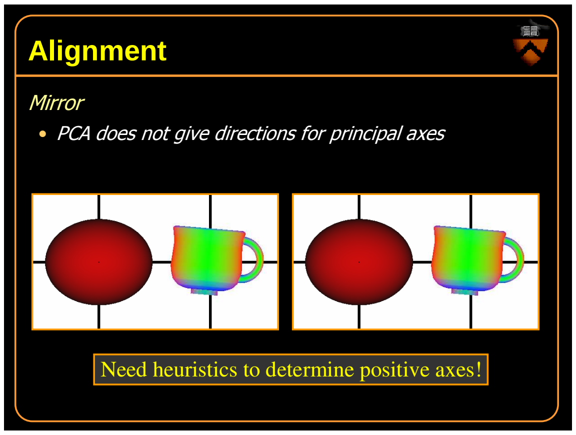


Alignment

Rotation (PCA)
- *Principal axes are eigenvectors associated with largest eigenvalues of 2nd order moments covariance matrix*

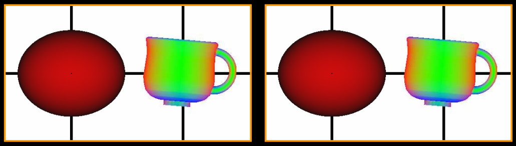


Not very robust!

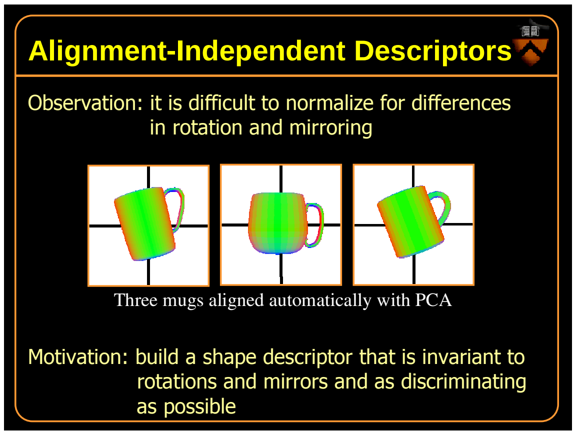


Alignment

Mirror
- *PCA does not give directions for principal axes*

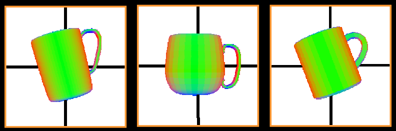


Need heuristics to determine positive axes!



Alignment-Independent Descriptors

Observation: it is difficult to normalize for differences in rotation and mirroring



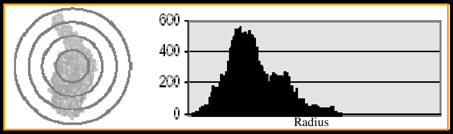
Three mugs aligned automatically with PCA

Motivation: build a shape descriptor that is invariant to rotations and mirrors and as discriminating as possible

Shape Histograms

Image courtesy of Ankerst et al, 1999

Shape descriptor stores histogram of how much surface resides at different radii from center of mass

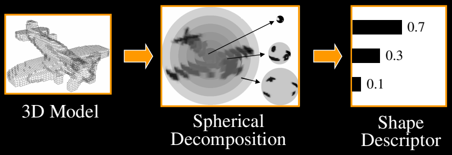


Shape Histograms (shells)
(Ankherst, 1999)

Shape Histograms

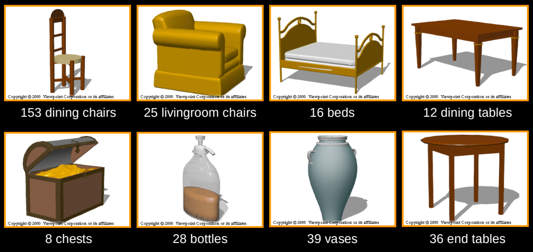
Image courtesy of Misha Kazhdan

Shape descriptor stores histogram of how much surface resides at different radii from center of mass



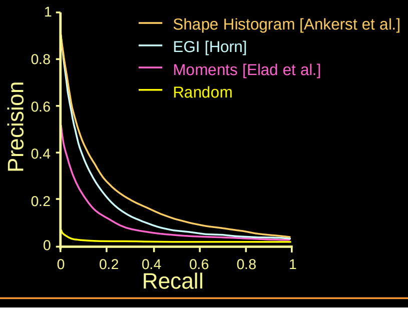
Shape Histogram Experiment

Test database is Viewpoint household collection
1,890 models, 85 classes



Shape Histogram Retrieval Results

Precision-recall curves (mean for all queries)

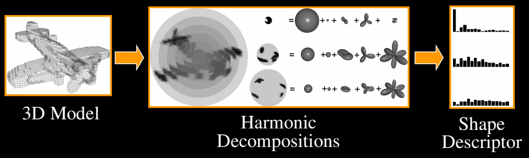


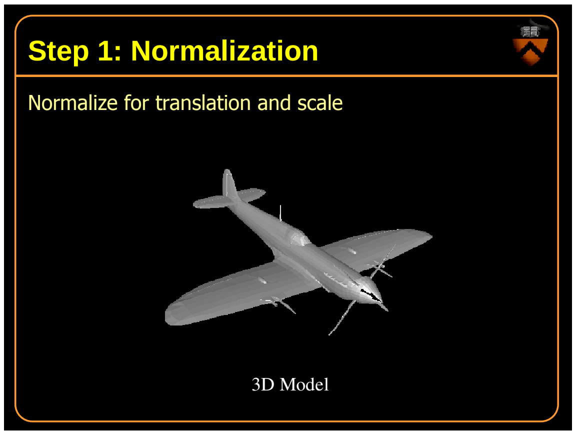
Shape Histograms

Properties
- ü Insensitive to noise
- ü Insensitive to topology
- ü Robust to degeneracies
- ü Quick to compute
- ü Efficient to match
- ü Concise to store
- ü Invariant to rotations
- Discriminating?

Harmonic Shape Descriptor

Key idea:
- Decompose each sphere into irreducible set of rotation independent components
- Store "how much" of the model resides in each component

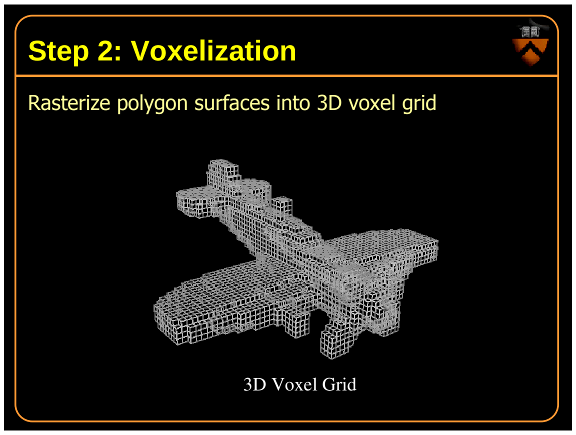




Step 1: Normalization

Normalize for translation and scale

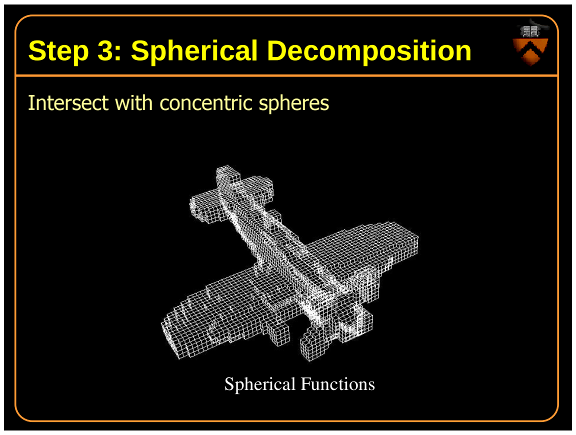
3D Model



Step 2: Voxelization

Rasterize polygon surfaces into 3D voxel grid

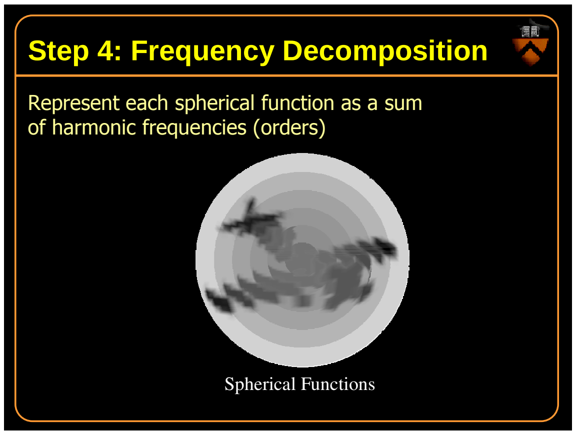
3D Voxel Grid



Step 3: Spherical Decomposition

Intersect with concentric spheres

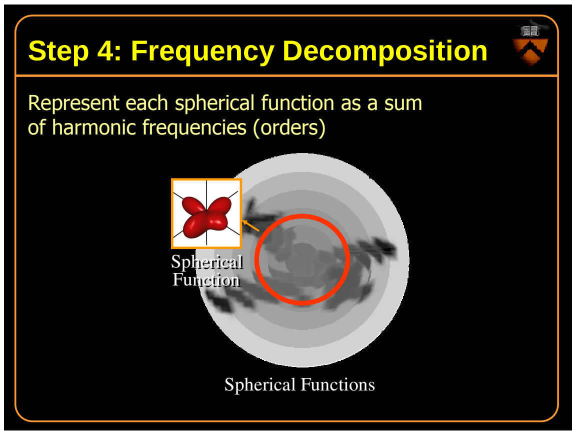
Spherical Functions



Step 4: Frequency Decomposition

Represent each spherical function as a sum of harmonic frequencies (orders)

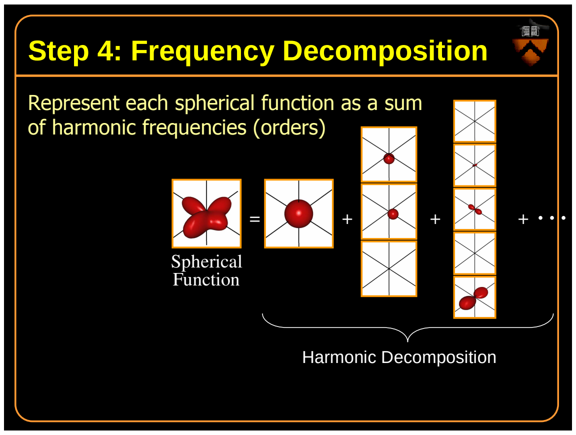
Spherical Functions



Step 4: Frequency Decomposition

Represent each spherical function as a sum of harmonic frequencies (orders)

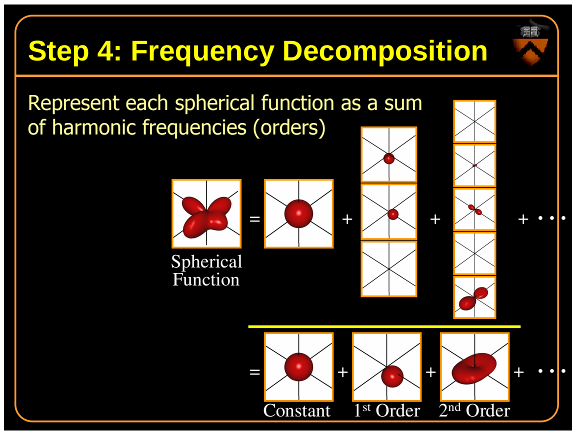
Spherical Functions



Step 4: Frequency Decomposition

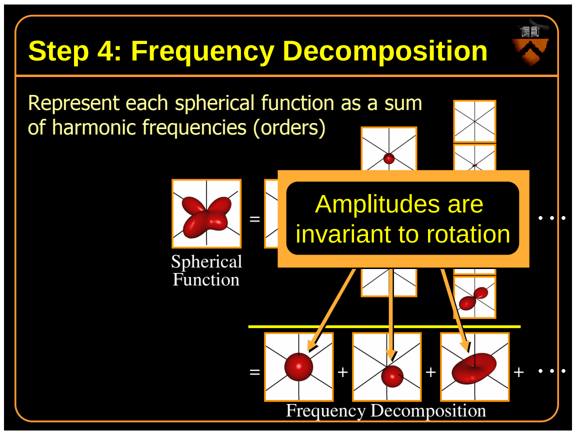
Represent each spherical function as a sum of harmonic frequencies (orders)

Harmonic Decomposition



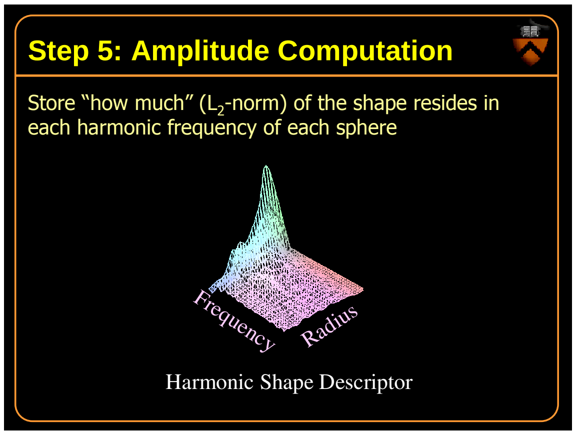
Step 4: Frequency Decomposition

Represent each spherical function as a sum of harmonic frequencies (orders)



Step 4: Frequency Decomposition

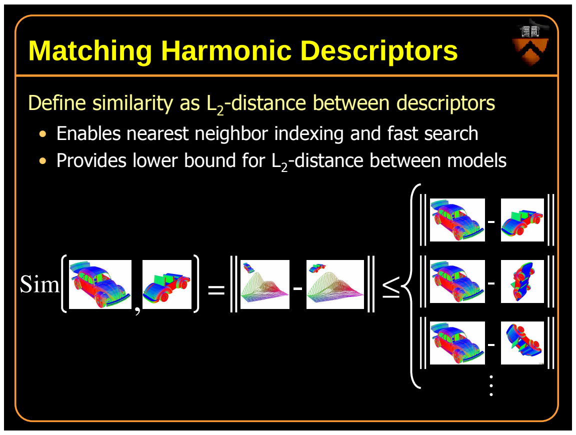
Represent each spherical function as a sum of harmonic frequencies (orders)



Step 5: Amplitude Computation

Store "how much" (L_2-norm) of the shape resides in each harmonic frequency of each sphere

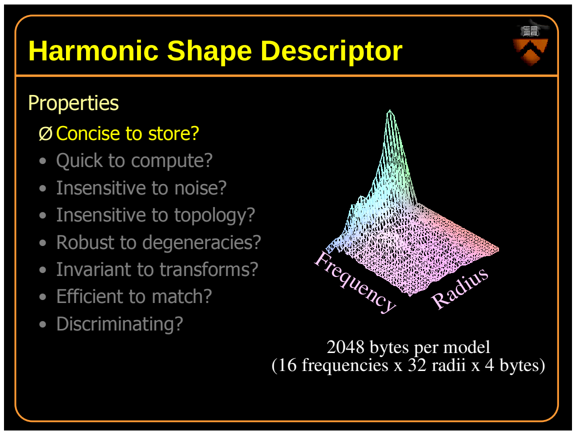
Harmonic Shape Descriptor



Matching Harmonic Descriptors

Define similarity as L_2-distance between descriptors

- Enables nearest neighbor indexing and fast search
- Provides lower bound for L_2-distance between models



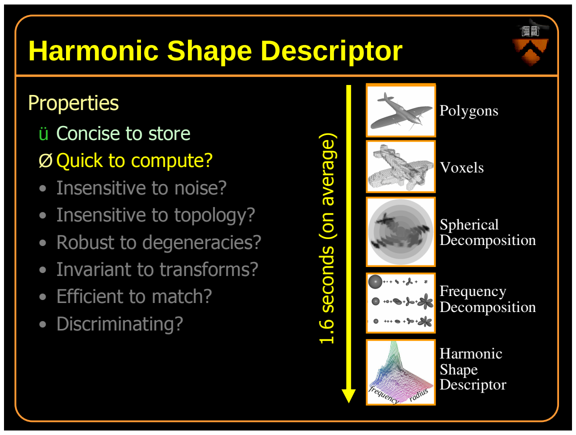
Harmonic Shape Descriptor

Properties

- Ø Concise to store?
- Quick to compute?
- Insensitive to noise?
- Insensitive to topology?
- Robust to degeneracies?
- Invariant to transforms?
- Efficient to match?
- Discriminating?

Frequency
Radius

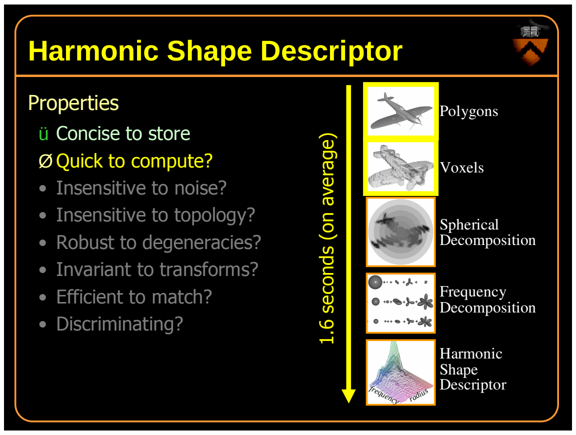
2048 bytes per model
(16 frequencies x 32 radii x 4 bytes)



Harmonic Shape Descriptor

Properties

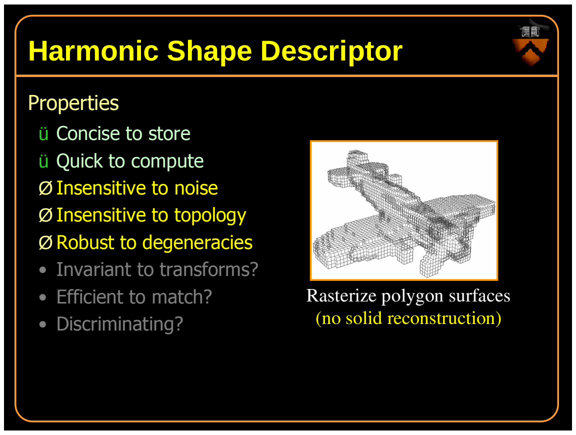
- ü Concise to store
- Ø Quick to compute?
- Insensitive to noise?
- Insensitive to topology?
- Robust to degeneracies?
- Invariant to transforms?
- Efficient to match?
- Discriminating?



Harmonic Shape Descriptor

Properties

- ü Concise to store
- Ø Quick to compute?
- Insensitive to noise?
- Insensitive to topology?
- Robust to degeneracies?
- Invariant to transforms?
- Efficient to match?
- Discriminating?

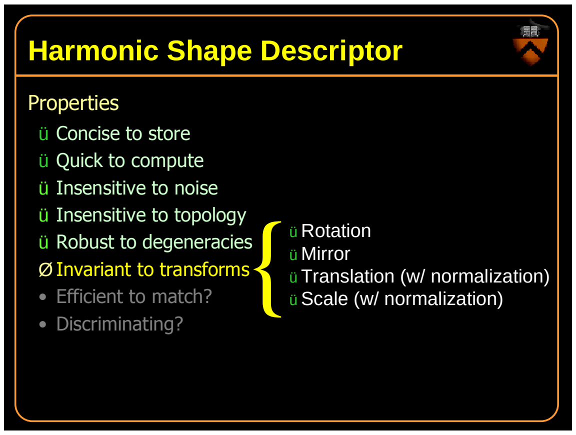


Harmonic Shape Descriptor

Properties

- ü Concise to store
- ü Quick to compute
- Ø Insensitive to noise
- Ø Insensitive to topology
- Ø Robust to degeneracies
- Invariant to transforms?
- Efficient to match?
- Discriminating?

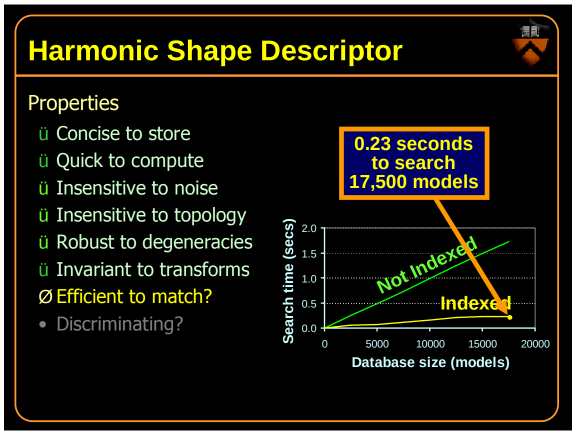
Rasterize polygon surfaces
(no solid reconstruction)



Harmonic Shape Descriptor

Properties

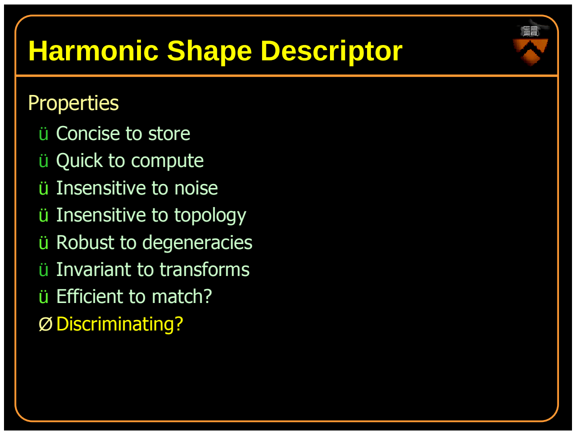
- ü Concise to store
- ü Quick to compute
- ü Insensitive to noise
- ü Insensitive to topology
- ü Robust to degeneracies
- Ø Invariant to transforms
  - ü Rotation
  - ü Mirror
  - ü Translation (w/ normalization)
  - ü Scale (w/ normalization)
- Efficient to match?
- Discriminating?



Harmonic Shape Descriptor

Properties

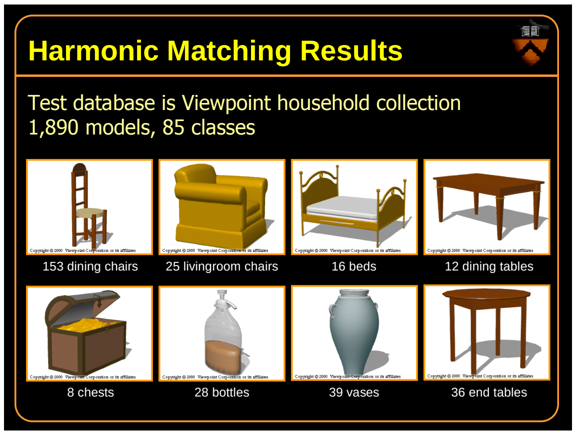
- ü Concise to store
- ü Quick to compute
- ü Insensitive to noise
- ü Insensitive to topology
- ü Robust to degeneracies
- ü Invariant to transforms
- Ø Efficient to match?
- Discriminating?



Harmonic Shape Descriptor

Properties

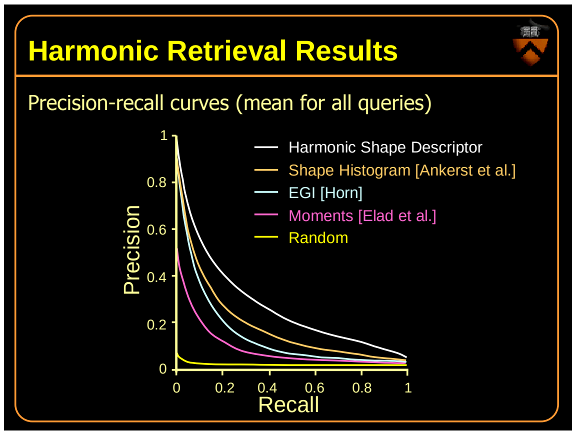
- ü Concise to store
- ü Quick to compute
- ü Insensitive to noise
- ü Insensitive to topology
- ü Robust to degeneracies
- ü Invariant to transforms
- ü Efficient to match?
- Ø Discriminating?



Harmonic Matching Results

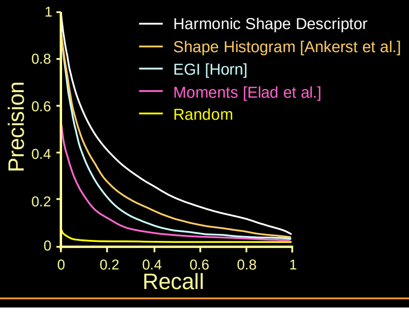
Test database is Viewpoint household collection
1,890 models, 85 classes

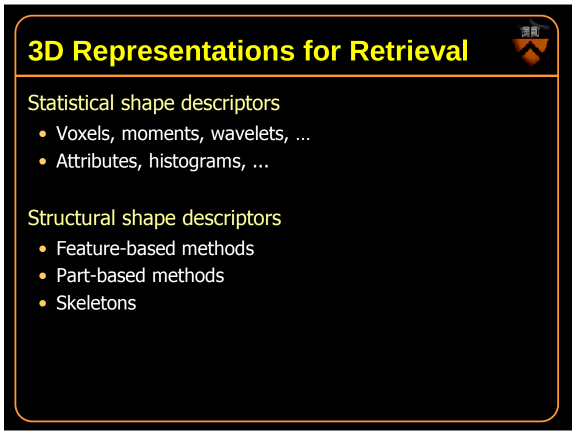
153 dining chairs	25 livingroom chairs	16 beds	12 dining tables
8 chests	28 bottles	39 vases	36 end tables



Harmonic Retrieval Results

Precision-recall curves (mean for all queries)





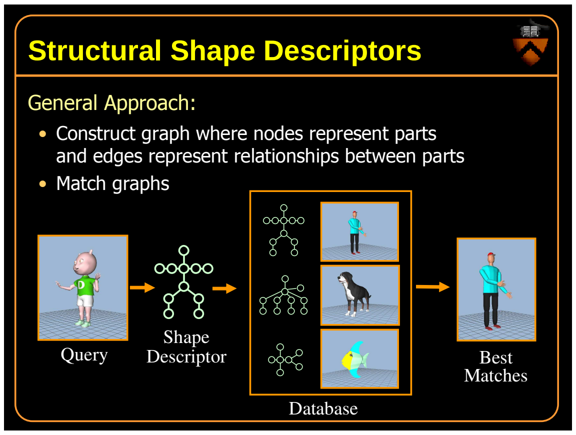
3D Representations for Retrieval

Statistical shape descriptors

- Voxels, moments, wavelets, …
- Attributes, histograms, …

Structural shape descriptors

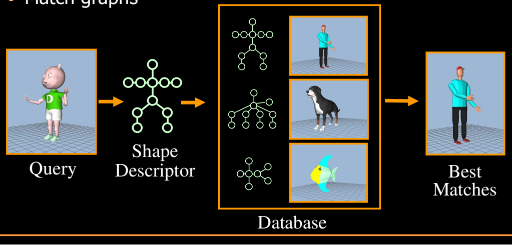
- Feature-based methods
- Part-based methods
- Skeletons

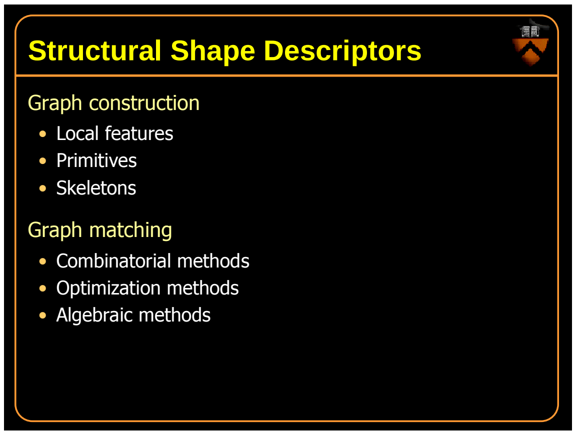


Structural Shape Descriptors

General Approach:

- Construct graph where nodes represent parts and edges represent relationships between parts
- Match graphs





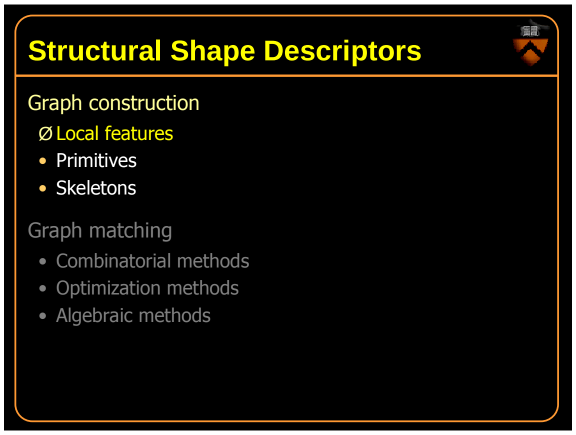
Structural Shape Descriptors

Graph construction

- Local features
- Primitives
- Skeletons

Graph matching

- Combinatorial methods
- Optimization methods
- Algebraic methods



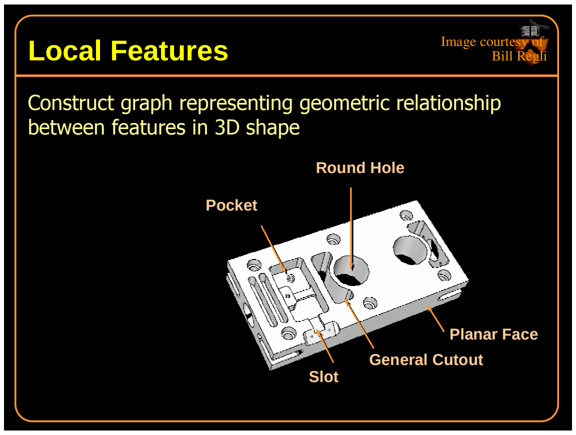
Structural Shape Descriptors

Graph construction

Ø Local features
- Primitives
- Skeletons

Graph matching

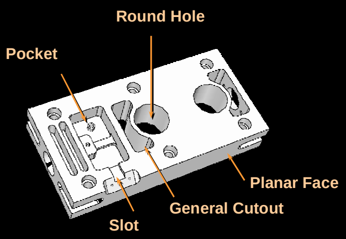
- Combinatorial methods
- Optimization methods
- Algebraic methods



Local Features

Image courtesy of Bill Regli

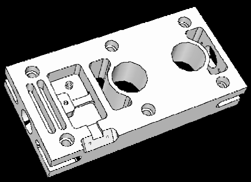
Construct graph representing geometric relationship between features in 3D shape



Local Features

Images courtesy of Bill Regli

General strategy
- Extract features
- Construct graph
- Match graphs

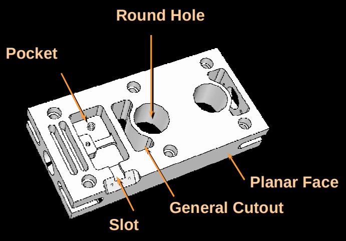


Local Features

Images courtesy of Bill Regli

Example 1

Ø Extract features
- Construct graph
- Match graphs

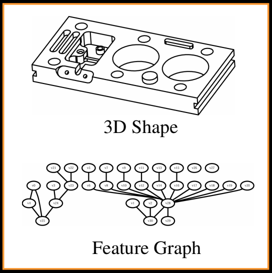


Local Features

Images courtesy of Bill Regli

Example 1
- Extract features

Ø Construct graph
- Match graphs

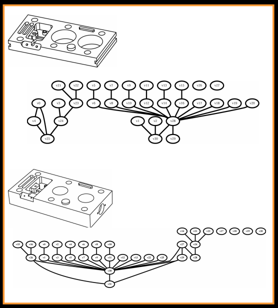


Local Features

Images courtesy of Bill Regli

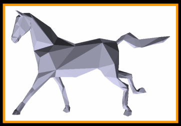
Example 1
- Extract features
- Construct graph

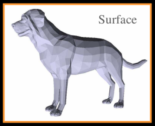
Ø Match graphs



Local Features

Example 2
- Extract features
- Construct graph
- Match graphs


A



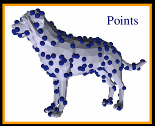
B

Local Features

Example 2

Ø Extract features
- Construct graph
- Match graphs

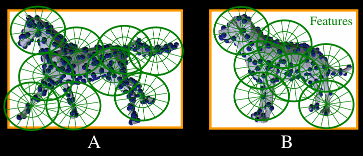
A



B

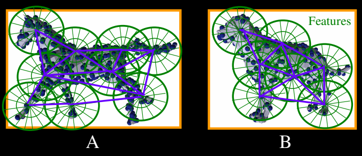
Local Features

Example 2
- Ø Extract features
- Construct graph
- Match graphs



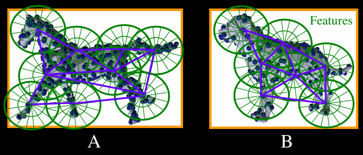
Local Features

Example 2
- Extract features
- Ø Construct graph
- Match graphs



Local Features

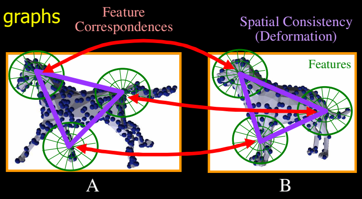
Example 2
- Extract features
- Construct graph
- Ø Match graphs



$$D(A,B) = \sum_{\text{Correspondences}} \Delta\text{FeatureShape} + \sum_{\substack{\text{Correspondence}\\\text{Pairs}}} \Delta\text{SpatialConsistency}$$

Local Features

Example 2
- Extract features
- Construct graph
- Ø Match graphs



$$D(A,B) = \sum_{\text{Correspondences}} \Delta\text{FeatureShape} + \sum_{\substack{\text{Correspondence}\\\text{Pairs}}} \Delta\text{SpatialConsistency}$$

Local Features

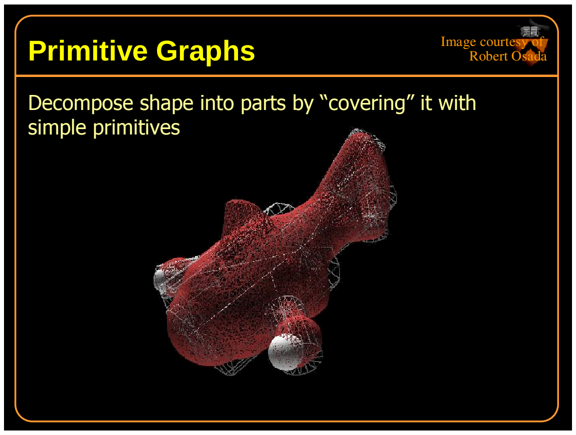
Properties
- Decomposes shape into graph based on features
- Features can be prioritized, pruned, and attributed
- Invariance to transformations

Limitations
- Robust to missing or extra parts
- Computationally expensive

Structural Shape Descriptors

Graph construction
- Local features
- Ø Primitives
- Skeletons

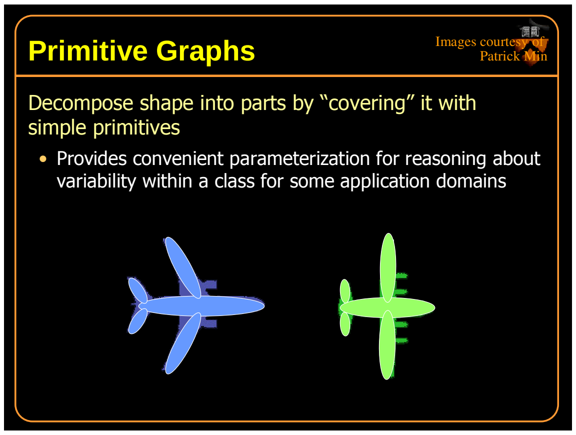
Graph matching
- Combinatorial methods
- Optimization methods
- Algebraic methods



Primitive Graphs

Image courtesy of Robert Osada

Decompose shape into parts by "covering" it with simple primitives

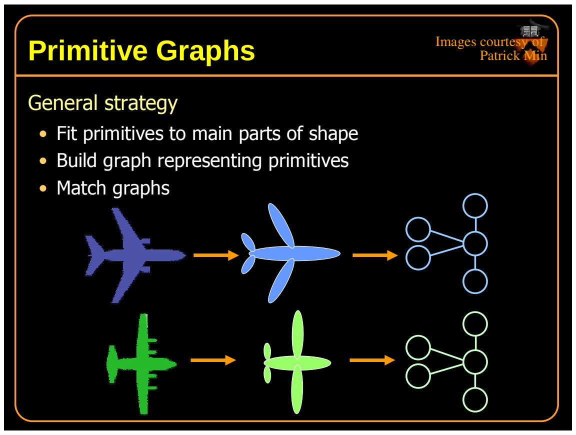


Primitive Graphs

Images courtesy of Patrick Min

Decompose shape into parts by "covering" it with simple primitives

- Provides convenient parameterization for reasoning about variability within a class for some application domains

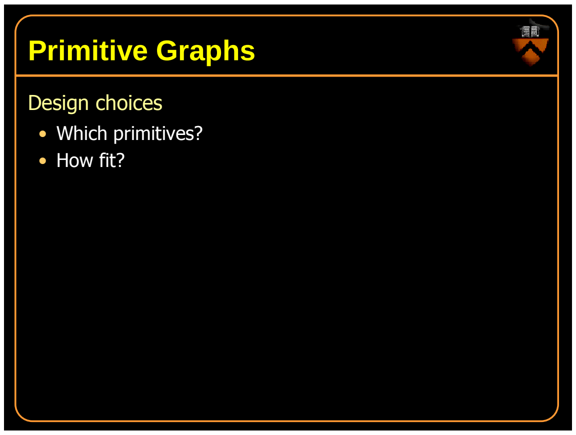


Primitive Graphs

Images courtesy of Patrick Min

General strategy

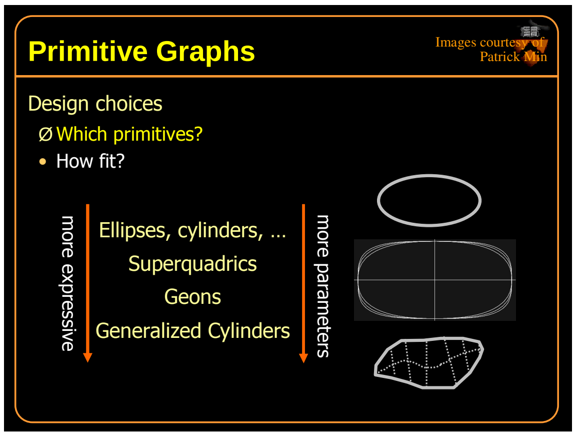
- Fit primitives to main parts of shape
- Build graph representing primitives
- Match graphs



Primitive Graphs

Design choices

- Which primitives?
- How fit?



Primitive Graphs

Images courtesy of Patrick Min

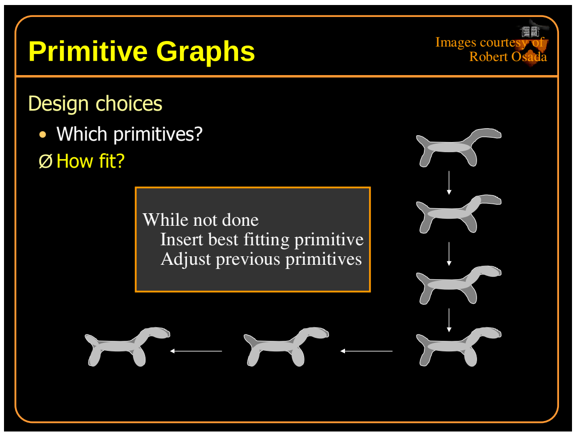
Design choices

Ø Which primitives?

- How fit?

Ellipses, cylinders, …
Superquadrics
Geons
Generalized Cylinders

more expressive
more parameters



Primitive Graphs

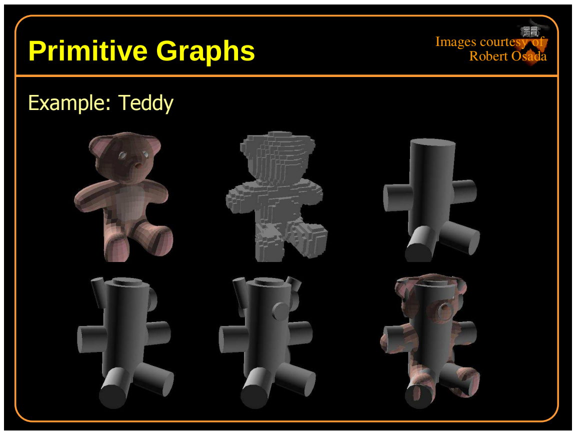
Images courtesy of Robert Osada

Design choices

- Which primitives?

Ø How fit?

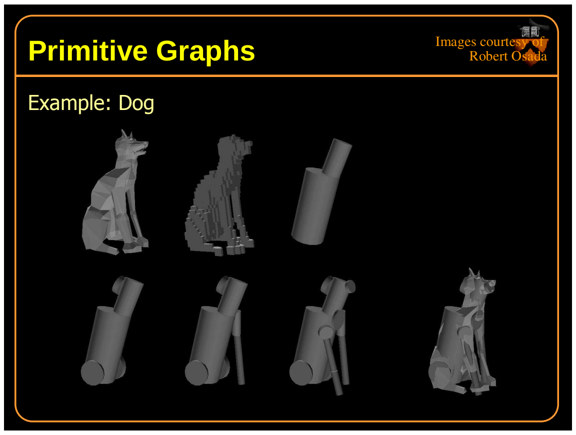
```
While not done
    Insert best fitting primitive
    Adjust previous primitives
```



Primitive Graphs

Images courtesy of Robert Osada

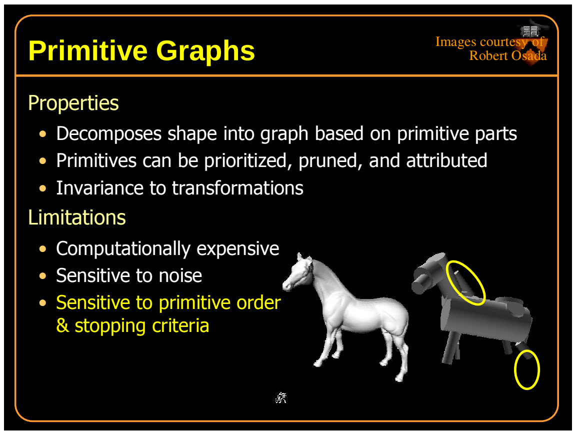
Example: Teddy



Primitive Graphs

Images courtesy of Robert Osada

Example: Dog

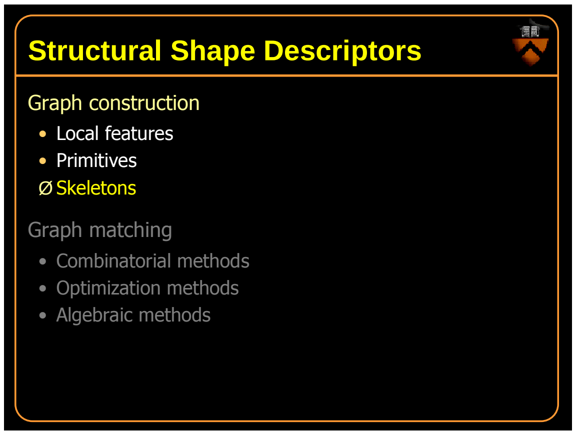


Primitive Graphs

Images courtesy of Robert Osada

Properties
- Decomposes shape into graph based on primitive parts
- Primitives can be prioritized, pruned, and attributed
- Invariance to transformations

Limitations
- Computationally expensive
- Sensitive to noise
- Sensitive to primitive order & stopping criteria

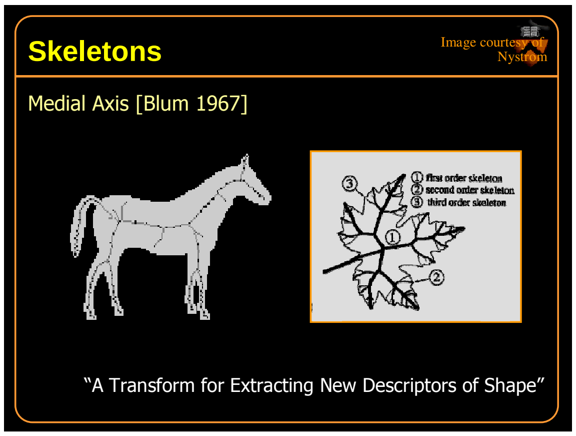


Structural Shape Descriptors

Graph construction
- Local features
- Primitives

Ø Skeletons

Graph matching
- Combinatorial methods
- Optimization methods
- Algebraic methods

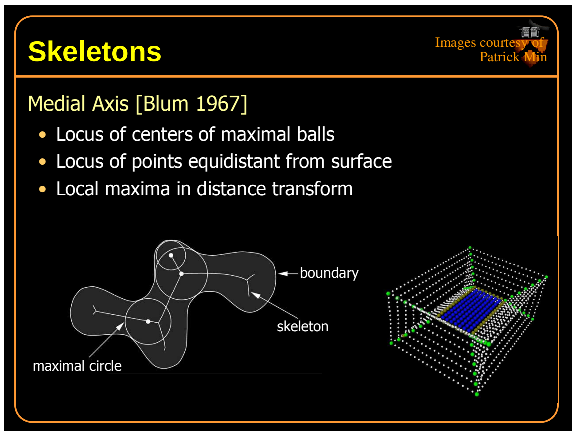


Skeletons

Image courtesy of Nystrom

Medial Axis [Blum 1967]

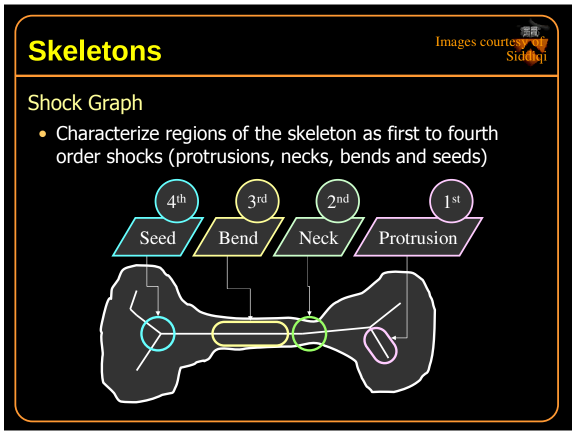
"A Transform for Extracting New Descriptors of Shape"



Skeletons

Images courtesy of Patrick Min

Medial Axis [Blum 1967]
- Locus of centers of maximal balls
- Locus of points equidistant from surface
- Local maxima in distance transform

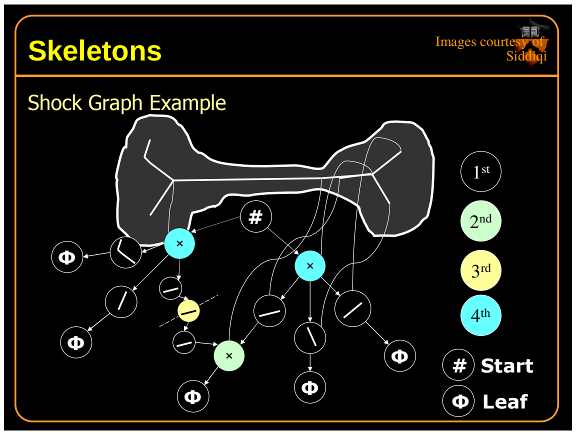


Skeletons

Images courtesy of Siddiqi

Shock Graph

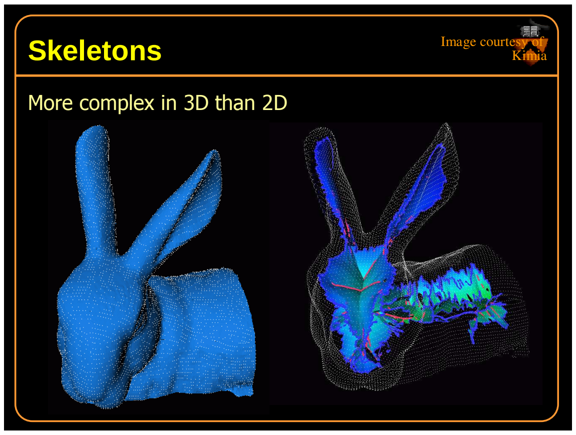
- Characterize regions of the skeleton as first to fourth order shocks (protrusions, necks, bends and seeds)



Skeletons

Images courtesy of Siddiqi

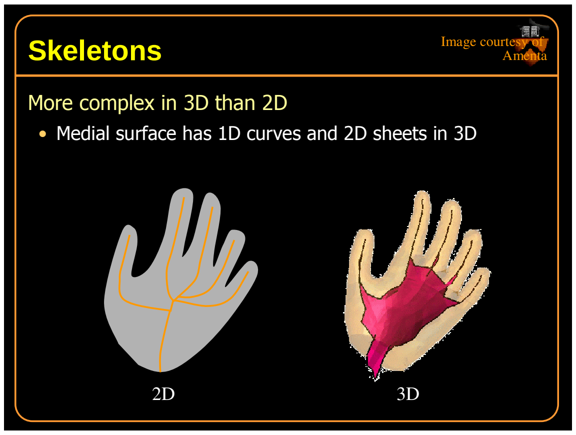
Shock Graph Example



Skeletons

Image courtesy of Kimia

More complex in 3D than 2D

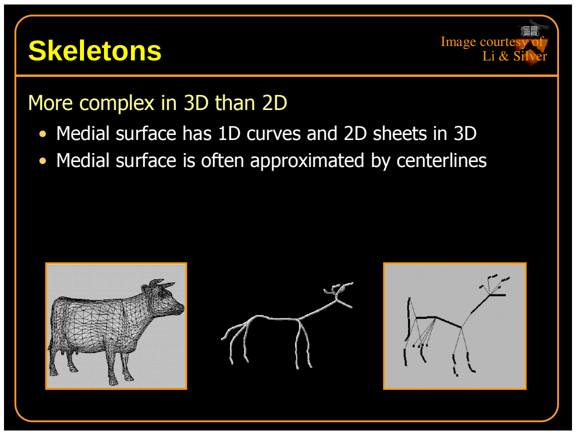


Skeletons

Image courtesy of Amenta

More complex in 3D than 2D

- Medial surface has 1D curves and 2D sheets in 3D

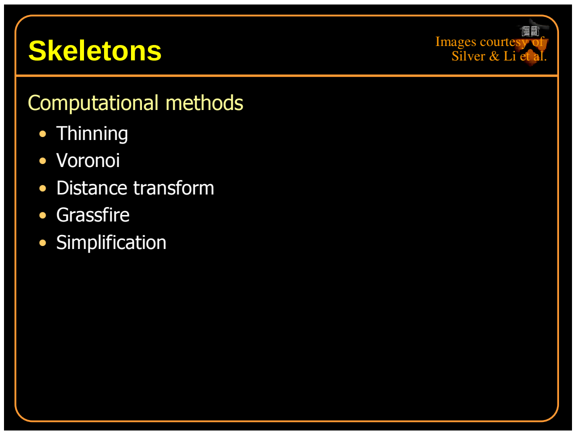


Skeletons

Image courtesy of Li & Silver

More complex in 3D than 2D

- Medial surface has 1D curves and 2D sheets in 3D
- Medial surface is often approximated by centerlines



Skeletons

Images courtesy of Silver & Li et al.

Computational methods

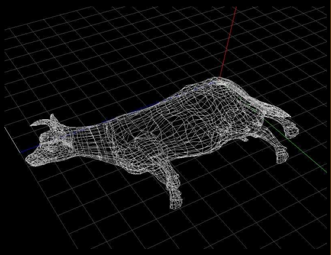
- Thinning
- Voronoi
- Distance transform
- Grassfire
- Simplification

Skeletons

Images courtesy of Patrick Min

Example: thinning

- Start with mesh
- Voxelize
- Thin voxels
- Make graph
- Simplify graph
- Assign attributes
- Match graphs

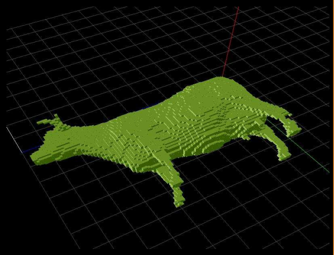


Skeletons

Images courtesy of Patrick Min

Example: topological thinning

- Start with mesh
- Voxelize
- Thin voxels
- Make graph
- Simplify graph
- Assign attributes
- Match graphs

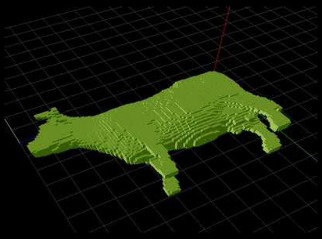


Skeletons

Images courtesy of Patrick Min

Example: topological thinning

- Start with mesh
- Voxelize
- Thin voxels
- Make graph
- Simplify graph
- Assign attributes
- Match graphs

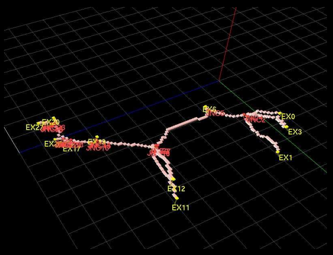


Skeletons

Images courtesy of Patrick Min

Example: topological thinning

- Start with mesh
- Voxelize
- Thin voxels
- Make graph
- Simplify graph
- Assign attributes
- Match graphs

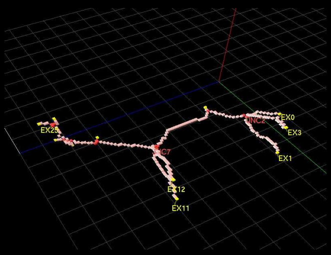


Skeletons

Images courtesy of Patrick Min

Example: topological thinning

- Start with mesh
- Voxelize
- Thin voxels
- Make graph
- Simplify graph
- Assign attributes
- Match graphs

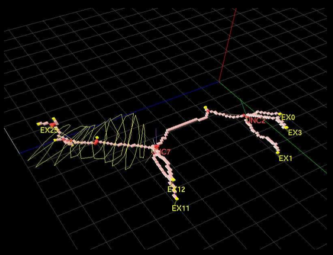


Skeletons

Images courtesy of Patrick Min

Example: topological thinning

- Start with mesh
- Voxelize
- Thin voxels
- Make graph
- Simplify graph
- Assign attributes
- Match graphs

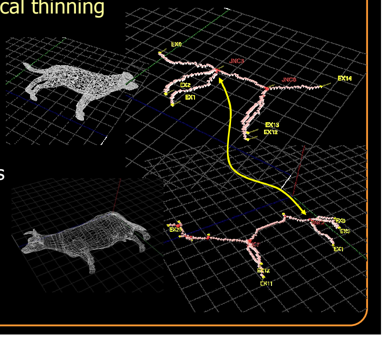


Skeletons

Images courtesy of Patrick Min

Example: topological thinning

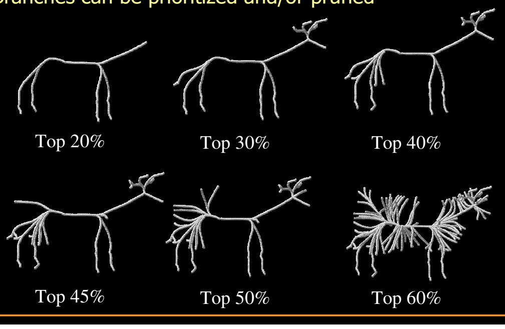
- Start with mesh
- Voxelize
- Thin voxels
- Make graph
- Simplify graph
- Assign attributes
- Match graphs



Skeletons

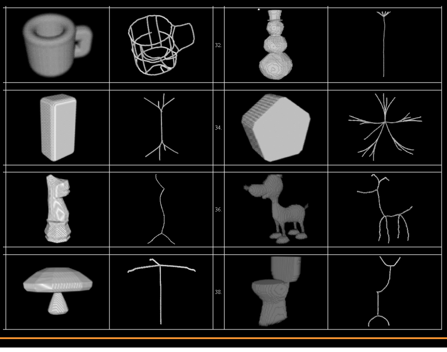
Images courtesy of Deborah Silver

Branches can be prioritized and/or pruned



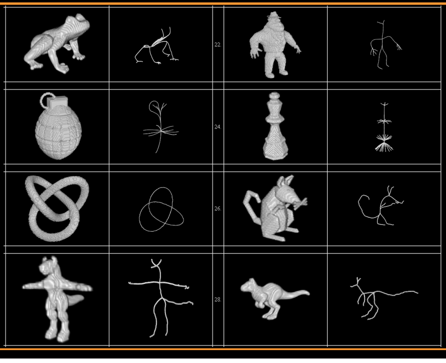
Skeletons

Images courtesy of Deborah Silver



Skeletons

Images courtesy of Deborah Silver



Skeletons

Properties

- Decomposes shape into graph based on local symmetries
- Branches can be prioritized, pruned, and attributed
- Invariance to transformations

Limitations

- Computationally expensive
- Sensitive to noise
- May yield complex structures

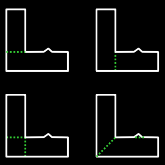
Graph Construction Summary

Construct graph where nodes represent parts and edges represent relationships between parts

- Features
- Primitives
- Skeletons

Issues

- Computationally expensive
- Sensitive to noise
- Sensitive to stopping criteria
- Often do not produce same topology for similar shapes



Structural Shape Descriptors

Graph construction
- Skeletons
- Primitive graphs
- Feature graphs
- Surface segmentation graphs

Graph matching
- Combinatorial methods
- Optimization methods
- Algebraic methods

Structural Shape Descriptors

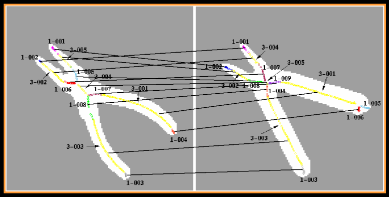
Graph construction
- Skeletons
- Primitive graphs
- Feature graphs
- Surface segmentation graphs

Graph matching
- Ø Combinatorial methods
- Optimization methods
- Algebraic methods

Combinatorial Graph Matching

Image courtesy of Siddiqi

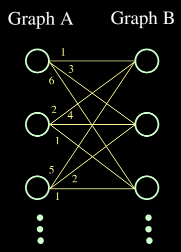
Find a matching (node correspondence) between two graphs with greatest total weight



Combinatorial Graph Matching

Image courtesy of Shokoufandeh

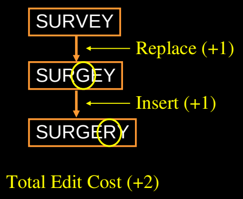
Find a matching (node correspondence) between two graphs with greatest total weight
- Ø Bipartite matching
- Graph edit distance



Combinatorial Graph Matching

Image courtesy of Shokoufandeh

Find a matching (node correspondence) between two graphs with greatest total weight
- Bipartite matching
- Ø Graph edit distance

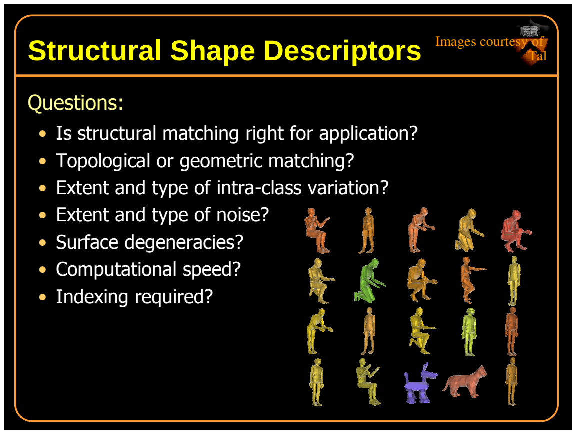


Graph Matching Summary

Combinatorial methods
- No indexing

Optimization methods
- No indexing

Algebraic methods (spectral methods)
- Limited many-to-many matching support

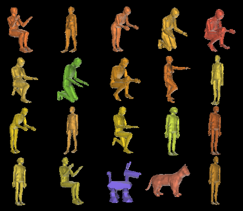


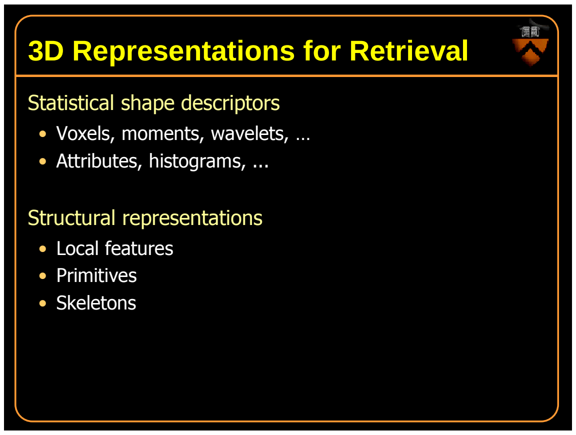
Structural Shape Descriptors

Images courtesy of Tal

Questions:

- Is structural matching right for application?
- Topological or geometric matching?
- Extent and type of intra-class variation?
- Extent and type of noise?
- Surface degeneracies?
- Computational speed?
- Indexing required?





3D Representations for Retrieval

Statistical shape descriptors

- Voxels, moments, wavelets, …
- Attributes, histograms, ...

Structural representations

- Local features
- Primitives
- Skeletons