CAP5415-Computer Vision
Lecture 17-Shape in Vision

Dr. Ulas Bagci
bagci@ucf.edu
Outline

• Shape in Vision
• Morphology Analysis and Shape

**Shape Context** – Shape Matching
• Shape Models - **Active Shape Model (ASM)**
• Gradient Descent Algorithm
• Active Contours and Snakes

**ShapeNET**: Deep Learning for Shape Representation
Shape in Vision

• More than 50% of human brain is involved in visual information analysis (directly and indirectly).
Shape in Vision

• More than 50% of human brain is involved in visual information analysis (directly and indirectly).

• Several important imaging concepts are closely related to biologic principles: edge detection, Gabor filtering, artificial neural networks, high curvature points in shape perception, etc.
Shape in Vision

• More than 50% of human brain is involved in visual information analysis (directly and indirectly).
• Several important imaging concepts are closely related to biologic principles: edge detection, Gabor filtering, artificial neural networks, high curvature points in shape perception, etc.
• Shape of the objects play a significant role in perception.
Shape in Vision

• More than 50% of human brain is involved in visual information analysis (directly and indirectly).

• Several important imaging concepts are closely related to biologic principles: edge detection, Gabor filtering, artificial neural networks, high curvature points in shape perception, etc.

• Shape of the objects play a significant role in perception.

• **Shape is any connected set of points!**
Shape in Vision

• Shape is any connected set of points!

Shape (S) Points
T: boundary
P: interior
Q: exterior
Example Applications

**NeuroScience**
- Morphological taxonomy of neural cells
- Interplay between form and function
- Comparisons between cells of different cortical areas
- Comparisons between cells of different species
  - ...

**Internet/Document Analysis**
- Content-based information retrieval
- Watermarking
- Graphic design
- WWW
- Optical character recognition
- Multimedia databases
  - ...

**Visual Arts**
- Video restoration
- Video tracking
- Special effects
- Games
- Computer graphics
- Image synthesis
- Visualizations
  - ...

**Imaging/Vision**
- Pathology detection/classification (spiculated and spherical nodules)
- 3D pose estimation
- Morphological operations
- Anatomy modeling
- Segmentation
- Registration
- Volumetry

Other areas: medicine, biology, engineering, physics, agriculture, security...
Ex: Actor Driven Face Animation
Ex: Audio-Visual Speech Analysis

- Lip reading can improve speech analysis
Ex: Hand-tracking for human robot interaction and animations

Thumb closed to “lock” object while hand returns to start.

Thumb open: object follows hand translating and rotating.
Most structures of clinical interest have a characteristic shape and anatomical location relative to other structures.

- Ventricles
- Caudate nucleus
- Lentiform nucleus
Ex. Anatomy Modeling

- Heart model with large vessels (multi-linear shape model)
Ex: Model Creation

• Require labeled training images
  – Landmarks represents correspondences

Match shape model to new image!
More reasons for Why study shape?

• Across the normal population, instances vary in size but also in shape, while retaining the “key features” of the shape
  – Shape varies statistically

• Abnormal shape variations often characterize disease
  – Learn the “normal” shape & variations on normal subjects, and be sensitive to clinically significant variants to these norms
Case Study: Morphology of Plant Leaves

A special problem where shape analysis usually comes into play is the classification of biological entities based on respective morphological information.
A special problem where shape analysis usually comes into play is the classification of biological entities based on respective morphological information.

4 different plant species/classes are given. Now consider that if a given unknown plant leaf, how do you classify based on what we have?
Case Study: Morphology of Plant Leaves

A special problem where shape analysis usually comes into play is the classification of biological entities based on respective morphological information.

4 different plant species/classes are given. Now consider that if a given unknown plant leaf, how do you classify based on what we have?

Consider two simple features-> Feature1: entropy, Feature2: perimeter of the plant leaf
Case Study: Morphology of Plant Leaves
Case Study: Morphology of Plant Leaves

- Belongs to class 3 by nearest neighborhood!
Case study 2: Shape Similarity

Which shape is more similar?

How can similarity be measured objectively?
Case Study 3: Shape Matching

- Shape matching can involve finding the correct corresponding points between a given shape A and a target shape B.
Shape Processing: Dilation

• Morphological operations are often done with a structural element “s” and a binary image “f”.
Shape Processing: Dilation

• Morphological operations are often done with a structural element “s” and a binary image “f”.
• Different structuring elements define different outputs for the same processed shape.
Shape Processing: Dilation

• Morphological operations are often done with a structural element “s” and a binary image “f”.
• Different structuring elements define different outputs for the same processed shape.
• Dilation is used to expand or dilate the shapes in an input image.
Shape Processing: Dilation

• Morphological operations are often done with a structural element “s” and a binary image “f”.
• Different structuring elements define different outputs for the same processed shape.
• Dilation is used to expand or dilate the shapes in an input image.

1. Shaded area -> effected area from dilation
2. Structural element has a reference point
3. Boundary is tracked
1. Shaded area (both internal and external of the object) is effected region by dilation.
2. Depending on the size of the structural element, holes can be diminished/eliminated (left)
3. Concave regions narrower than size of structural element are filled up (right).

**The result of image dilation** is the set of image pixels where the intersection between structuring element and object is not empty.

It is often used for noise reduction in shapes, gap and hole filling, etc.
Dilation: Example

Structuring element (g), centered at o.

Binary image (f)
Dilation: Example

Structuring element (g), centered at o.

Binary image (f)
Dilation: Example

Input Binary image (f)

Output Binary image (f)
Shape Processing: Erosion

- Opposite of dilation
- Practically it is used for object separation
Erosion: Example
Morphological Closing & Opening

• Please read
Shape Representation

• Boundary-based (contour) shape representation
  – Object shape is the set of points identified in the original image (outline, 1D approach)

• Region-based shape representation
  – Treat shape as regions (2D approach)
Shape Representations: Contour Based

- **Shape Representation**
  - Contours
  - Regions
  - Transforms

- **Parametric Contour**
  - Vectors $(x(t), y(t))$
  - Complex Signal $u(t) = x(t) + jy(t)$
  - Chain-Codes
  - Run-Lengths

- **Set of Contour Points**

- **Curve Approximation**
  - Polygonal Approximation
  - Circle Arcs
  - Elliptical Arcs
  - Syntactic Primitives
  - Autoregressive Models
  - B-Splines
  - Snakes/Active Contours
  - Multiscale Primitives
Shape Representations: Region Based

- **Shape Representation**
  - **Contours**
  - **Regions**
  - **Transforms**

**Region Decomposition**
- Polygons
- Voronoi / Delaunay
- Quadtrees
- Syntactic Primitives
- Morphological Decomposition
- Dendrograms

**Bounding Regions**
- Feret Box
- Minimum Enclosing Rectangle
- Convex Hull / Deficiency

**Internal Features**
- Skeletons
- Shape Matrix
- Distance Transform
- Run-Length
Shape Representations: Transforms

Typical Transform Domain Descriptors
✓ Transform Coefficients
✓ Transform Measures (e.g. Energy)
✓ Transform Statistics

Linear Transforms
- Monoscale
  ✓ Fourier
  ✓ Karhunen-Loève
  ✓ Sin
  ✓ Cosine
  ✓ Laplace
  ✓ Z

- Multiscale
  ✓ Short-Time Fourier Transform
  ✓ Haar
  ✓ Wavelets
  ✓ Gabor
  ✓ Scale-Space

Nonlinear Transforms
- Hough
- Time-Frequency Distributions
- Cohen Classes
- Wigner-Ville
- Mathematical Morphology
SHAPE CONTEXT
Matching with Shape Context

- Measuring similarity between shapes for object recognition
  - Solving point correspondences between two shapes
  - Using correspondences to estimate alignment of shapes
- **Shape Context** at a reference point captures the distribution of remaining points relative to it.
  - Globally discriminative descriptor

Example of two handwritten digits. In terms of pixel to pixel correspondences, these two shapes are quite different, but to human observers, they are similar!
Matching with Shape Context

• Object is represented with “n” points.
• For a point $p_i$, compute a coarse histogram $h_i$ of the relative coordinates of the “n-1” points.

$$h_i(k) = \# \{ q \neq p_i : (q - p_i) \in \text{bin}(k) \}$$

• Histogram is the shape context
  – Bin -> in polar coordinates (to make it more sensitive to position of the nearby points)
  • Linearly increased positional uncertainty in Cartesian coordinates

$$C_{ij} \equiv C(p_i, q_j) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)},$$

Cost of matching two points i and j
Appearance info can be added Too!
Matching with Shape Context

Two shapes to be matched!

Log polar bins (5,12)

(d) (e) (f)

(correspondences)
Matching with Shape Context
Matching with Shape Context

First column: query image
Remaining fives show closest objects found by using shape context!
SHAPE MODELS
Action Understanding: Key Components

**Image measurements**
- Foreground segmentation
- Image gradient
- Optical flow
- Local space-time features

**Prior knowledge**
- Deformable contour models
- 2D/3D body models
- Motion priors
- Background models
- Space-time templates
- SVM classifiers

**Association**
- (Semi-) Manual = training annotation
- Automatic = result
Shape Models

- **Point distribution, Active Shape**, Active Appearance models (Cootes & Taylor)
- Fourier Snakes (Szekely)
- Active Contours (Blake)
- Parametrically-deformable models (Staib & Duncan)
- ...

Active Shape Models (ASM)

• Search images for represented structures
• Automated
• Classify shapes
• Specific to ranges of variation
• Robust (noisy, cluttered, and occluded images)
• Deform to characteristics of the class represented
• Learn specific patterns of variability from a training set
Point Distribution Model (PDM)

• Represent a shape instance by a judiciously chosen set of points (features), each of which is a \textit{k-dim} vector. N feature points are stacked into a long vector of length \( kn \):

\[
q = [p_1, \ldots, p_n]^T
\]

• where

\[
p_i = (x_i, y_i), \text{ for } i = 1, \ldots, n
\]
Example Instances in PDM

- M=12 training instances
- One point feature, the 14th m is shown with correspondences on each instance
  \[ p_{14}^1, p_{14}^2, \ldots, p_{14}^{12}, \]
- Evidently, if the number of feature points n is large, and the training set size M is also large, this is going to be tedious unless it can be automated!
Aligning the shape instances

• The Procrustes Algorithm is used to that the sum of instances to the mean of each shape is minimized! Why?

\[ \sum_{i=1,\ldots,M} (q_i - \bar{q})^2 \]
Aligning the shape instances

• The Procrustes Algorithm is used to that the sum of instances to the mean of each shape is minimized! Why?
  – Variation should be calculated from shape only, not because of translation, scaling, and rotation!

\[
\sum_{i=1,\ldots,M} (q_i - \bar{q})^2
\]
Aligning the shape instances

• The Procrustes Algorithm is used to minimize the sum of instances to the mean of each shape:

\[ \sum_{i=1,\ldots,M} (q_i - \bar{q})^2 \]

Why?
– Variation should be calculated from shape only, not because of translation, scaling, and rotation!

1. Translate each shape instance \( q_i \) so that its center of mass is at the origin
2. Choose one example as the mean shape and rescale so that norm of the mean shape is 1
3. Record the first estimate as \( q_0 \) to define the default reference frame
4. Align all instances of the shape with the current estimate of the mean shape
5. Re-estimate the mean from the aligned shapes
6. Re-scale and re-iterate the process if necessary
Modeling Shape Variation

• PCA (principal component analysis) is used

1. Compute the mean of the data
   \[ \bar{q} = \frac{1}{M} \sum_{i=1}^{M} q_i \]

2. Compute the covariance of the data
   \[ S = \frac{1}{M-1} \sum_{i} (q_i - \bar{q})(q_i - \bar{q})^T \]

3. Compute the eigenvectors \( u_i \) and eigenvalues \( \lambda_i \) of the covariance matrix, sorted in decreasing order of eigenvalue size

4. Remove the small eigenvalues, retaining “most” (eg 98%) of the variation
   \[ \text{Choose } t \text{ so that } \sum_{i=1}^{t} \lambda_i \geq 0.98 \sum_{i=1}^{M} \lambda_i \]
We have the eigenvectors $\mathbf{u}_i$ sorted in order so that $\lambda_1 \geq \lambda_2 \ldots \geq \lambda_M$

and have chosen $t$ so that $\sum_{i=1}^{t} \lambda_i \geq 0.98 \sum_{i=1}^{M} \lambda_i$

Now define the matrix $\mathbf{U}$ from the top $t$ eigenvectors:

$$\mathbf{U} = [\mathbf{u}_1 | \mathbf{u}_2 | \ldots | \mathbf{u}_t]$$

Then we approximate any shape instance $\mathbf{x}$ by a $t$-dimensional vector $\mathbf{b}$

$$\mathbf{x} \approx \bar{\mathbf{x}} + \mathbf{U} \cdot \mathbf{b}$$

$$\mathbf{b} = \mathbf{U}^T \cdot (\mathbf{x} - \bar{\mathbf{x}})$$

$E = \sum_{i=1}^{M} \lambda_i$

$0.98E$

$t \ll M$
Active Shape Model (ASM): Approximation

• We now **approximate** any instance of the shape, including the training instances, by projecting onto the first $t$ eigenvectors:

$$q = \tilde{q} + \sum_{i=1}^{t} b_i u_i$$

• The weight vector $b$ is identified as the characteristic of this instance of the shape:

$$b = [b_1, \ldots, b_t]^T$$

• Varying the weights $b_i$ enables us to explore the allowable variations in the shape!
Placing the model

• The model points are defined in a model co-ordinate frame
• Must apply global transformation, $T$, to place in image

$x = \bar{x} + Pb$

$T(x; X_c, Y_c, s, \theta)$

$X = T(\bar{x} + Pb)$
ASM Search

Need to search for local match
For each point:
- strongest edge
- correlation
- statistical model of profile
Deeper in ASM Search

- Model boundary
- Model point \((X, Y)\)
- Profile normal to boundary
- Interpolate at these points
  \[
  (X, Y) + i(s_n n_x, s_n n_y)
  \]
  \[
  i = \ldots -2, -1, 0, 1, 2, \ldots
  \]
- Take steps of length \(s_n\) along \((n_x, n_y)\)
- Select point along profile at strongest edge

\[
\frac{dg(x)}{dx} = 0.5(g(x + 1) - g(x - 1))
\]

\[
\frac{dg(x)}{dx} = 0.5(g(x + 1) - g(x - 1))
\]

59
The PDM was learned from 18 shapes, each comprising 72 points, at finger tips, finger junctions, and equally spaced along the finger sides.
Ex: Face Shapes
Finding the model pose & parameters

• Suppose we have identified a set of points $\mathbf{Y}$ in the image. Evidently, we can seek to minimize the squared distance:

$$\| \mathbf{Y} - T(\bar{q} + \sum_{i=1}^{t} b_i \mathbf{u}_i) \|^2$$
Finding the model pose & parameters

• Suppose we have identified a set of points $Y$ in the image. Evidently, we can seek to minimize the squared distance:

$$\left| Y - T(\bar{q} + \sum_{i=1}^{t} b_i u_i) \right|^2$$

Algorithm

1. Initialize $b=0$
2. Generate initial model instance $q = (\bar{q} + \sum_{i=1}^{t} b_i u_i)$
3. Find $T$ that best aligns $q$ to $Y$ (e.g., similarity transform)
4. Invert pose parameters, to project $y = T^{-1}(Y)$ into model frame
5. Update the model parameters: $b = U^T(y - \bar{q})$ where $U = [u_1 | u_1 | ... | u_t]$
6. Repeat from step 2 until convergence
Ex: Fitting a cartilage model to a knee MR image
Ex: MR Brain structure segmentation
ASM Summary

Pros:

- Shape prior helps overcoming segmentation errors
- Fast optimization
- Can handle interior/exterior dynamics

Cons:

- Optimization gets trapped in local minima
- Re-initialization is problematic

Possible improvements:

Learn and apply specific motion priors for different actions
Active Appearance Models

• Please read

Active Contour Model (Snake)

- First introduced in 1987 by Kass et al, and gained popularity since then.

- Represents an object boundary or some other salient image feature as a \textit{parametric curve}.

- An energy functional $E$ is associated with the curve.

- The problem of finding object boundary is cast as an energy minimization problem.
Frameworks for Snakes

- A higher level process or a user initializes any curve close to the object boundary.
- The snake then starts deforming and moving towards the desired object boundary.
- In the end it completely “shrink-wraps” around the object.
Active Contour Modeling

• The contour is defined in the \((x, y)\) plane of an image as a parametric curve
  \[ v(s) = (x(s), y(s)) \quad 0 \leq s \leq 1 \]

• Contour is said to possess an energy \(E_{\text{snake}}\) which is defined as the sum of the three energy terms.

\[ E_{\text{snake}} = E_{\text{internal}} + E_{\text{external}} + E_{\text{constraint}} \]

• The energy terms are defined **cleverly** in a way such that the final position of the contour will have a minimum energy \((E_{\text{min}})\)

• Therefore our problem of detecting objects reduces to an energy minimization problem.

What are these energy terms which do the trick for us?


Internal Energy

- The smoothness energy at contour point $v(s)$ could be evaluated as

$$E_{in}(v(s)) = \alpha(s) \left| \frac{dv}{ds} \right|^2 + \beta(s) \left| \frac{d^2v}{d^2s} \right|^2$$

Elasticity/stretching  Stiffness/bending

Then, the interior energy (smoothness) of the whole snake

$$C = \{ v(s) | s \in [0,1] \}$$

$$E_{in} = \int_{0}^{1} E_{in}(v(s)) \, ds$$
Internal Energy

elastic energy (elasticity)

\[ \frac{d \nu}{ds} \approx \nu_{i+1} - \nu_i \]

\[ C = (\nu_0, \nu_1, \nu_2, ..., \nu_{n-1}) \in \mathbb{R}^{2n} \]

\[ \nu_i = (x_i, y_i) \]

bending energy (stiffness)

\[ \frac{d^2 \nu}{ds^2} \approx (\nu_{i+1} - \nu_i) - (\nu_i - \nu_{i-1}) = \nu_{i+1} - 2\nu_i + \nu_{i-1} \]
Internal Energy

\[ \mathbf{C} = (\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{n-1}) \in \mathbb{R}^{2n} \]

\[ \mathbf{v}_i = (x_i, y_i) \]

\[ \frac{d \mathbf{v}}{ds} \approx \mathbf{v}_{i+1} - \mathbf{v}_i \]

\[ \frac{d^2 \mathbf{v}}{ds^2} \approx (\mathbf{v}_{i+1} - \mathbf{v}_i) - (\mathbf{v}_i - \mathbf{v}_{i-1}) = \mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1} \]

\[ E_{in} = \sum_{i=0}^{n-1} \alpha \left| \mathbf{v}_{i+1} - \mathbf{v}_i \right|^2 + \beta \left| \mathbf{v}_{i+1} - 2\mathbf{v}_i + \mathbf{v}_{i-1} \right|^2 \]

Elasticity

Stiffness

Min energy when curve minimizes length of contour..........................is smooth
**External Energy**

- The external energy describes **how well the curve matches the image data locally**
- Numerous forms can be used, attracting the curve toward different image features
External (Image) Energy

• Suppose we have an image \( I(x,y) \)
• Can compute image gradient \( \nabla I = (I_x, I_y) \) at any point
• Edge strength at pixel \((x,y)\) is \( |\nabla I(x,y)|\)

• External energy of a contour point \( \mathbf{v}=(x,y) \) could be

\[
E_{ex}(\mathbf{v}) = -|\nabla I(\mathbf{v})|^2 = -|\nabla I(x,y)|^2
\]

- **External energy** term for the whole snake is

\[
E_{ex} = \int_{0}^{1} E_{ex}(\mathbf{v}(s)) \, ds \quad \text{continuous case} \quad C = \{ \mathbf{v}(s) | s \in [0,1] \}
\]

\[
E_{ex} = \sum_{i=0}^{n-1} E_{ex}(\mathbf{v}_i) \quad \text{discrete case} \quad C = \{ \mathbf{v}_i | 0 \leq i < n \}
\]
The total energy of a basic elastic snake is

\[ E = \alpha \cdot \int_{0}^{1} \left| \frac{dv}{ds} \right|^2 ds \quad - \quad \int_{0}^{1} \left| \nabla I(v(s)) \right|^2 ds \]

continuous case

\[ C = \{ v(s) | s \in [0,1] \} \]

\[ E = \alpha \cdot \sum_{i=0}^{n-1} \left| v_{i+1} - v_i \right|^2 \quad - \quad \sum_{i=0}^{n-1} \left| \nabla I(v_i) \right|^2 \]

discrete case

\[ C = \{ v_i | 0 \leq i < n \} \]

- elastic smoothness term (interior energy)
- image data term (exterior energy)
Basic Elastic Snake

\[ C = (v_i \mid 0 \leq i < n) = (x_0, y_0, x_1, y_1, \ldots, x_{n-1}, y_{n-1}) \]

\[ E_{in} = \alpha \cdot \sum_{i=0}^{n-1} L_i^2 \]

This can make a curve shrink (to a point)

\[ = \alpha \cdot \sum_{i=0}^{n-1} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 \]

\[ E_{ex} = - \sum_{i=0}^{n-1} |\nabla I(x_i, y_i)|^2 \]

\[ = - \sum_{i=0}^{n-1} |I_x(x_i, y_i)|^2 + |I_y(x_i, y_i)|^2 \]
Find Contour C that minimizes $E(C)$

$$E(C) = \alpha \cdot \sum_{i=0}^{n-1} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 - \sum_{i=0}^{n-1} |I_x(x_i, y_i)|^2 + |I_y(x_i, y_i)|^2$$

Optimization problem for function of $2n$ variables
- can compute local minima via gradient descent
- more robust option: dynamic programming
Constraint Forces ($E_{\text{constraints}}$)

- Initial snake result can be nudged where it goes wrong, simply add extra external energy terms to
  - Pull nearby points toward cursor, or
    \[
    E_{\text{pull}} = - \sum_{i=0}^{n-1} \frac{r^2}{|v_i - p|^2}
    \]
  - Push nearby points away from cursor
    \[
    E_{\text{push}} = + \sum_{i=0}^{n-1} \frac{r^2}{|v_i - p|^2}
    \]
Gradient Descent

- Example: minimization of functions of 2 variables

\[ -\nabla E = \begin{bmatrix} -\frac{\partial E}{\partial x} \\ -\frac{\partial E}{\partial y} \end{bmatrix} \]

**negative gradient** at point \((x, y)\) gives direction of the steepest descent towards lower values of function \(E\)
Gradient Descent

- Example: minimization of functions of 2 variables

\[
\begin{align*}
E(x, y) &= \nabla E \\
\begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} x \\ y \end{pmatrix} - \Delta t \cdot \begin{pmatrix} \frac{\partial E}{\partial x} \\ \frac{\partial E}{\partial y} \end{pmatrix} \\
\text{Stop at a local minima where} \quad \nabla E &= 0
\end{align*}
\]
Gradient Descent

- Example: minimization of functions of 2 variables

$E(x, y)$

High sensitivity with respect to the initialisation!!
Gradient Descent in Snakes

simple elastic snake energy

\[
E(x_0, \ldots, x_{n-1}, y_0, \ldots, y_{n-1}) = -\sum_{i=0}^{n-1} |I_x(x_i, y_i)|^2 + |I_y(x_i, y_i)|^2 + \alpha \cdot \sum_{i=0}^{n-1} (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2
\]

here, energy is a function of 2n variables

update equation for the whole snake

\[
C' = C - \nabla E \cdot \Delta t
\]
Dynamic Programming for Snakes

• Please Read
  – *Interactive Segmentation with Intelligent Scissors* by E. Mortensen and W. Barrett,
  – *Using Dynamic Programming for Solving variational Problems in Vision* by AA. Amini et al, where authors used dynamic programming for image segmentation tasks.
Ex: Corpus Collasum
Problems with Snakes

- Depends on number and spacing of control points
- Snake may over-smooth the boundary
- Initialization is crucial
- Not trivial to prevent curve self intersecting

- Can not follow topological changes of objects
ShapeNET: A Deep Representation of Volumetric Shapes (Z.Wu et al CVPR 15)

Motivation: 3D Shape Representation

Only in Theory
Hand Craft Design

Only in Theory
Hand Craft Design

No State-of-the-arts
Use 3D Shapes

Limited to
Instance Level Matching

Desirable Properties of a Good 3D Shape Representation:

- **Data-driven:** learn from data rather than predefined shape routines.
- **Generic:** any complex shapes rather than simple shape primitives.
- **Compositional:** compose complex shapes by assembling simple ones.
- **Versatile:** applicable to various vision tasks.
Princeton ModelNet Dataset

A large-scale CAD dataset of more than 150,000 models in 660 categories.

Object Categories

**Left:** word cloud visualization of the ModelNet dataset based on the number of 3D models in each category. Larger font size indicates more instances in the category. **Right:** Examples of 3D chair models.

The 3D ShapeNet paper represents a geometric 3D shape as a probability distribution of binary variables on a 3D voxel grid.
For each neuron, authors average the top 100 training examples with highest responses (>0.99).

The averaged result is visualized by transparency in 3D (Gray) and by the average surface obtained from zero-crossing (Red).
3D Feature Extraction

1. Convert 3D ShapeNets to a 3D CNN
2. Feature learning by back-propagation
3. Test for classification and retrieval

<table>
<thead>
<tr>
<th>10 classes</th>
<th>SPH [18]</th>
<th>LFD [8]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>classification</td>
<td>79.79%</td>
<td>79.87%</td>
<td><strong>83.54%</strong></td>
</tr>
<tr>
<td>retrieval AUC</td>
<td>45.97%</td>
<td>51.70%</td>
<td><strong>69.28%</strong></td>
</tr>
<tr>
<td>retrieval MAP</td>
<td>44.05%</td>
<td>49.82%</td>
<td><strong>68.26%</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>40 classes</th>
<th>SPH [18]</th>
<th>LFD [8]</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>classification</td>
<td>68.23%</td>
<td>75.47%</td>
<td><strong>77.32%</strong></td>
</tr>
<tr>
<td>retrieval AUC</td>
<td>34.47%</td>
<td>42.04%</td>
<td><strong>49.94%</strong></td>
</tr>
<tr>
<td>retrieval MAP</td>
<td>33.26%</td>
<td>40.91%</td>
<td><strong>49.23%</strong></td>
</tr>
</tbody>
</table>

Table 1: Shape Classification and Retrieval Results.

Precision = \( \frac{tp}{tp + fp} \)
Recall = \( \frac{tp}{tp + fn} \)

**Recall and Completion**

**Inference Process**
- Convert the depth map into volumetric representation
- Identify free space, observed surface, unknown space
- Infer unknown space and label jointly by gibbs sampling

**Shape Completion**

![Image of various 3D models and images with labels](image)

**Results on synthetic depth**

---

**Recognition and Reconstruction on NYU depth**

*Training only uses CAD models!*

---

**Table 2: Accuracy for View-based 2.5D Recognition on NYU dataset.**

<table>
<thead>
<tr>
<th></th>
<th>bathtub</th>
<th>bed</th>
<th>chair</th>
<th>desk</th>
<th>dresser</th>
<th>monitor</th>
<th>nightstand</th>
<th>sofa</th>
<th>table</th>
<th>toilet</th>
<th>all</th>
</tr>
</thead>
<tbody>
<tr>
<td>[29] Depth</td>
<td>0.000</td>
<td>0.729</td>
<td>0.806</td>
<td>0.100</td>
<td>0.466</td>
<td>0.222</td>
<td>0.343</td>
<td>0.481</td>
<td>0.415</td>
<td>0.200</td>
<td>0.376</td>
</tr>
<tr>
<td>NN</td>
<td>0.429</td>
<td>0.446</td>
<td>0.395</td>
<td>0.176</td>
<td>0.467</td>
<td>0.333</td>
<td>0.188</td>
<td>0.458</td>
<td>0.455</td>
<td>0.400</td>
<td>0.374</td>
</tr>
<tr>
<td>ICP</td>
<td>0.571</td>
<td>0.608</td>
<td>0.194</td>
<td>0.375</td>
<td>0.733</td>
<td>0.389</td>
<td>0.438</td>
<td>0.349</td>
<td>0.052</td>
<td>1.000</td>
<td>0.471</td>
</tr>
<tr>
<td>3D ShapeNets</td>
<td>0.142</td>
<td>0.500</td>
<td>0.685</td>
<td>0.100</td>
<td>0.366</td>
<td>0.500</td>
<td>0.719</td>
<td>0.277</td>
<td>0.377</td>
<td>0.700</td>
<td>0.437</td>
</tr>
<tr>
<td>3D ShapeNets fine-tuned</td>
<td>0.857</td>
<td>0.703</td>
<td>0.919</td>
<td>0.300</td>
<td>0.500</td>
<td>0.625</td>
<td>0.735</td>
<td>0.247</td>
<td>0.400</td>
<td>0.579</td>
<td>0.400</td>
</tr>
<tr>
<td>[29] RGB</td>
<td>0.142</td>
<td>0.743</td>
<td>0.766</td>
<td>0.150</td>
<td>0.266</td>
<td>0.166</td>
<td>0.218</td>
<td>0.313</td>
<td>0.376</td>
<td>0.200</td>
<td>0.334</td>
</tr>
<tr>
<td>[29] RGBD</td>
<td>0.000</td>
<td>0.743</td>
<td>0.693</td>
<td>0.175</td>
<td>0.466</td>
<td>0.388</td>
<td>0.468</td>
<td>0.602</td>
<td>0.441</td>
<td>0.500</td>
<td>0.448</td>
</tr>
</tbody>
</table>
Given a depth map of an object (like from kinect), authors convert it into a volumetric representation and identify the observed surface, free space and occluded space.  

**3D ShapeNets** can recognize object category, complete full 3D shape, and predict the next best view if the initial recognition is uncertain.  

Finally, **3D ShapeNets** can integrate new views to recognize object jointly with all views.
3D Shape Generation
References and Slice Credits

• Credits to: M.Brady and R.Szelisky
• Statistical Shape Analysis, I. Dryden and KV Mardia, 1998.
• A Survey of Shape Analysis Techniques, S. Loncaric, 1998.
• TF. Cootes et al. ASM and their training and applications, 1995.