Learning 3D Models for Scene Understanding

Thomas Funkhouser
Princeton University
Scene Understanding

Database of Example Scenes → Analysis (Learning) → Probabilistic Model of Scene → New Scenes

User Input → Output Segments & Labels

Output 3D Models

Recognition

Reconstruction

Processing

Synthesis
Traditional Computer Vision

Database of Example Images → Analysis (Learning) → Probabilistic Model of Images → New Images → Recognition → Output Segments & Labels

User Input → Reconstruction → Output 3D Models
Traditional Computer Vision

- Database of Example Images → Analysis (Learning) → Probabilistic Model of Images → New Images
- User Input

Output:
- Segments & Labels
- 3D Models

Objects:
- horse, tree, ...

Regions:
- sky, grass, horse, tree, human, saddle, ...

Text:
- [Li, Socher, Fei-Fei, 2009]
Why Learn Models of Images?

Database of Example Images → Analysis (Learning) → Probabilistic Model of Images → Recognition → Reconstruction → Processing → Synthesis → New Images → User Input → Output Segments & Labels → Output 3D Models
Why Learn Models of Images?

Reasons:
- The goal is to understand scenes from images … duh

Diagram:
- Database of Example Images
- Analysis (Learning)
- Probabilistic Model of Images
  - Recognition
  - Reconstruction
  - Processing
  - Synthesis
  - New Images
  - User Input
  - Output Segments & Labels
  - Output 3D Models
Why Learn Models of Images?

Reasons:

- The goal is to understand scenes from images … duh
- Some labeled examples, lots of unlabeled examples

LabelMe [Russell 2005]
Why Learn Models of Images?

Reasons:
- The goal is to understand scenes from images … duh
- Some labeled examples, lots of unlabeled examples

Problems:
- Shape
- Materials
- Lighting
- Viewpoint
- Perspective
- Occlusions
- Light transport
- Segmentation
- Noise
Shouldn’t We Learn Models of Scenes?

Database of Example Scenes → Analysis (Learning) → Probabilistic Model of Shapes, Materials, Lights, Cameras, Image Formation, etc. → Recognition → Reconstruction → Processing → Synthesis → Output Labels → Output 3D Models

Input Images → User Input

Probabilistic Model of Shapes, Materials, Lights, Cameras, Image Formation, etc.
Observation: databases of computer graphics (CG) models provide examples from which we can learn probabilistic models of scenes.
Why Learn from CG Models?

CG models provide …

- Shape
- Materials
- Lighting
- Viewpoint
- Perspective
- Occlusions
- Light transport
- Segmentation
- No noise
Why Learn from CG Models?

CG models provide …

- Shape
- Materials
- Lighting
- Viewpoint
- Perspective
- Occlusions
- Light transport
- Segmentation
- No noise

Issues:

- Enough examples?
- Quality?
Why Learn from CG Models?

CG models provide ...

- Shape
- Materials
- Lighting
- Viewpoint
- Perspective
- Occlusions
- Light transport
- Segmentation
- Noise

Issues:

- Enough examples?
- Quality?

Trimble 3D Warehouse
Why Learn from CG Models?

CG models provide …

- Shape
- Materials
- Lighting
- Viewpoint
- Perspective
- Occlusions
- Light transport
- Segmentation
- Noise

Issues:

- Enough examples?
- Quality?
Related Work

Using CG models for scene understanding
- Fitting CG models to images
  - Lai 2009, Xu 2011, Satkin 2013, Aubry 2014, etc.
- Fitting CG models to range scans
  - Nan 2012, Shen 2012, Kim 2012, Song 2014, etc.
- Using CG models to learn parameters
  - Zhao 2013, etc.

Analyzing databases of CG models
- Consistent segmentation, labeling, correspondence, …
  - Golovinskiy 2009, Sidi 2011, Kim 2013, Mitra 2013, etc.
- Learning probabilistic models
Focus of This Talk

This talk will focus on learning probabilistic models of shapes from databases of example CG models.
Outline of Talk

Introduction

Learning probabilistic models from CG collections
  ◦ Object templates
  ◦ Contextual model
  ◦ Hierarchical grammar

Conclusions
Outline of Talk

Introduction

Learning probabilistic models from CG collections

- Object templates
  - Generative model
  - Hierarchical grammar

Conclusions

Goal for This Project

Database of 3D meshes representing an object class

Consistent part segmentations, labels, and correspondences

Probabilistic Model of Shape
Goal for This Project

Database of 3D meshes representing an object class

Challenge

Need to discover segmentations, labels, correspondences, and deformation modes all together

Probabilistic Model of Shape

Consistent part segmentations, labels, and correspondences
Object Templates

Represent object class by part-based templates where each template has a set of parts, and each part has probability distributions for its shape, position, and anisotropic scales.
Template Learning and Fitting

Aim to learn a set of corresponding templates that provides a good fit to every mesh in the database
Template Learning and Fitting

Aim to learn a set of corresponding templates that provides a good fit to every mesh in the database.
Template Fitting Problem

For a given template and mesh, aim to minimize:

\[ E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}} \]

- \( E_{\text{data}} \) (template ↔ shape distance + local shape features)
- \( E_{\text{deform}} \) (plausibility of template deformation)
- \( E_{\text{smooth}} \) (close & similar regions get same label)

Unknowns are:
- Point segmentations and labels
- Point correspondences
- Part center positions
- Part anisotropic scales
Template Fitting Algorithm

Solve by iteratively minimizing different energy terms:

- Segmentation and labeling
- Point correspondence
- Part-aware deformation

\[ E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}} \]
Template Fitting Algorithm

Solve by iteratively minimizing different energy terms:

- Segmentation and labeling
  - Point correspondence
  - Part-aware deformation

\[
E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}}
\]

Solve with graph cut [Boykov 2001]
Template Fitting Algorithm

Solve by iteratively minimizing different energy terms:

- Segmentation and labeling
- Point correspondence
- Part-aware deformation

\[ E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}} \]

Solve with part-aware closest points
Template Fitting Algorithm

Solve by iteratively minimizing different energy terms:

- Segmentation and labeling
- Point correspondence
- Part-aware deformation

\[ E = E_{\text{data}} + \gamma E_{\text{deform}} + \beta E_{\text{smooth}} \]

Solve for positions and scales of each part by setting partial derivatives to zero.
Template Learning Problem

Aim to **learn** a set of corresponding templates that provides a good fit to every mesh in the database.
Template Learning Algorithm

Iteratively grow a set of templates with each optimized to fit a different cluster of meshes

Template Initialization
Template Fitting
Template Refinement

repeat until convergence
Template Learning Example

Initial Part-Based Template
Template Learning Example

Updated Part-Based Template
Template Learning Example

New Part-Based Template
Template Learning Example

Updated Part-Based Template
Template Learning and Fitting Results

Data sets:
- Crawl SketchUp Warehouse for collections by keyword
- Eliminate outliers with Mechanical Turk
- Specify manual correspondences for subset of models

Experiments:
- Solve for part-based templates for collection
- Evaluate correspondences & segmentations
Template Learning and Fitting Results

2 Templates

1508

1605

3113 Airplanes
Template Learning and Fitting Results

2 Templates

1508

1605

3113 Airplanes
Template Learning and Fitting Results

2 Templates

378

63

441 Bikes
Template Learning Results

2 Templates

378 63

441 Bikes

Failures
Surface Correspondence Results

Correspondence benchmark (7442 seats)
# Surface Segmentation Results

Co-segmentation benchmark [Sidi et al, 2011]

<table>
<thead>
<tr>
<th>Class</th>
<th>Hu</th>
<th>Our</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chairs</td>
<td>89.6</td>
<td><strong>97.6</strong></td>
</tr>
<tr>
<td>Lamps</td>
<td>90.7</td>
<td><strong>95.2</strong></td>
</tr>
<tr>
<td>FourLegged</td>
<td>88.7</td>
<td>86.9</td>
</tr>
<tr>
<td>Goblets</td>
<td>99.2</td>
<td><strong>97.6</strong></td>
</tr>
<tr>
<td>Vase</td>
<td>80.2</td>
<td><strong>81.3</strong></td>
</tr>
<tr>
<td>Guitars</td>
<td>98.0</td>
<td>88.5</td>
</tr>
<tr>
<td>Candelaebra</td>
<td>93.9</td>
<td>82.4</td>
</tr>
</tbody>
</table>

*within 2% or ours is better*
Outline of Talk

Introduction

Learning probabilistic models from CG collections
  ◦ Objet templates
  ➢ Contextual model
  ◦ Hierarchical grammar

Conclusions
Goal for This Project

Exemplar scenes + Database of Scenes → Probabilistic Model of Shape → Synthesized novel scenes
Goal for This Project

Exemplar scenes + Database of Scenes → Probabilistic Model of Shape

Challenge
Need to learn a model with great generality from few examples

Synthesized novel scenes
Define categories of objects based on their contexts in a scene rather than basic functions

- Learned from examples by clustering of objects with similar spatial neighborhoods

Some Contextual Object Categories
Contextual Model

Represent the probability of a scene $S$ by a generative model based on category cardinalities ($c$), support hierarchy topology relationships ($t$), and spatial arrangement relationships ($a$)

$$P(S) = P(c, t, a) = P(a|t, c) P(t|c) P(c)$$

Exemplar scenes
Contextual Model Details

Category cardinalities: $P(c)$
- Represent with Bayesian network
- Boolean random variables (# desks > 1?)
- Add support surface constraints

Object frequencies in target scenes + support constraints

Bayesian network
Contextual Model Details

Support relationships: $P(t|c)$

- Boolean random variables (desk supports keyboard?)
- Learn frequencies for pairs of categories
- Total probability is product over all objects in scene

$$P(t|c) = \prod_{o} P(C(o), C(support(o)))$$
Contextual Model Details

Spatial arrangements: \( P(a|t,c) = R(a,t,c)S(a,t,c) \)
- Random variables for relative positions and orientations
- Pairwise distributions of spatial relationships

Distributions of spatial relationships for pairs of object categories
Contextual Model Details

Spatial arrangements: \( P(a|t,c) = R(a,t,c)S(a,t,c) \)

- Random variables for relative positions and orientations
- Pairwise distributions of spatial relationships
- Feature distributions for positions on support surfaces

Distributions of geometric features of support surfaces
Scene Synthesis Results

Synthesized novel scenes
Scene Synthesis Results

User study suggests that people find our synthesized scenes almost as good as manually created ones.
Outline of Talk

Introduction

Learning probabilistic models from 3D collections
- Object templates
- Contextual model
- Hierarchical grammar

Conclusions
Goal for This Project

Probabilistic Model of Shape

Training set of labeled scene graphs
Goal for This Project

Probabilistic Model of Shape

Training set of labeled scene graphs

Unlabeled test scene
Goal for This Project

Training set of labeled scene graphs

Unlabeled test scene

Probabilistic Model of Shape

Labeled test scene graph
Goal for This Project

Probabilistic Model of Shape

Challenge

Scenes have a lot of variability in the types and spatial arrangements of objects

Training set of labeled scene graphs

Unlabeled test scene

Labeled test scene graph
Observation

Semantic and functional relationships are often more prominent within hierarchical contexts.
Hierarchical Grammar

We learn a hierarchical grammar from examples, and then use it to parse new test scenes

Diagram:
- Library
- Study area (X2)
- Meeting area (X5)
- Study desk
- Study chair (X2)
- Meeting table
- Meeting chair (X4)
Hierarchical Grammar

**Labels:** object group, object category, object part

- sleep area, bed, curtain piece

**Rules:** derivation from a label to a list of labels

- \( \text{bed} \rightarrow \text{bed frame} \quad \text{mattress} \)
Hierarchical Grammar

Probabilities:

Derivation: \( P_{nt}(rule \mid lhs) \)

\[
\text{bed} \rightarrow \text{frame} \quad \text{mattress} \\
P = 0.8
\]

Cardinality distribution: \( P_{\text{Card}}(\#, rhs \mid lhs) \)

\[
\text{sleep area} \rightarrow \text{bed} \quad \text{nightstand} \quad \text{rug} \quad \ldots
\]

<table>
<thead>
<tr>
<th>( P_{\text{card}}(* \mid \text{sleeparea}) )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4+</th>
</tr>
</thead>
<tbody>
<tr>
<td>bed</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>nightstand</td>
<td>0.3</td>
<td>0.3</td>
<td>0.4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>rug</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Hierarchical Grammar

Shape descriptor probability: $P_g(x \mid \text{label})$

$P_g(x \mid \text{bedframe}) > P_g(y \mid \text{bedframe})$

Spatial relationships: $P_g(v \mid \text{lhs, rhs1, rhs2})$

$P_s(x_1, x_2 \mid \text{sleeparea, bed, nightstand}) > P_s(x_1, x_3 \mid \text{sleeparea, bed, nightstand})$
Grammar Learning and Parsing

Learn

Probabilistic Hierarchical Grammar

Parse

Training set of labeled scene graphs

+ 

Unlabeled test scene

Labeled test scene graph
Hierarchical Grammar Results

Learned hierarchical probabilistic grammars from scenes in Trimble 3D Warehouse

77 bedrooms  30 classrooms  8 libraries

17 small bedrooms  8 small libraries
Hierarchical Grammar Results

Parsed left-out scenes with learned grammar

Shape Only
Flat Grammar
Our Hierarchical Grammar

Comparison of our parsing results to other methods
Hierarchical Grammar Results

Parsed left-out scenes with learned grammar

Shape Only

Flat Grammar

Our Hierarchical Grammar

Comparison of our parsing results to other methods
Hierarchical Grammar Results

Comparison of object classification

Impact of Individual Energy Terms
Outline of Talk

Introduction

Learning probabilistic models from 3D collections
- Part-based templates
- Generative model
- Hierarchical grammar

➢ Conclusions
Conclusions

Main result:
- Probabilistic models can be learned from collections of 3D meshes

Future work:
- Learn probabilistic models of lighting, materials, cameras
- Use these models for understanding scenes captured in scans and images
Conclusions

Database of Example CG Models → Analysis (Learning) → Probabilistic Model of Shapes, Materials, Lights, Cameras, Image Formation, etc. → Recognition → Reconstruction → Processing → Synthesis → Output Labels → Output 3D Models
Acknowledgments

People:
- Sid Chaudhuri, Steve Diverdi, Matthew Fisher, Pat Hanrahan, Qixing Huang, Vladimir Kim, Wilmot Li, Tianqiang Liu, Niloy Mitra, Daniel Ritchie, Manolis Savva

Data sets:
- Trimble 3D Warehouse

Funding:
- NSF, Intel, Google, Adobe

Thank You!