Interactive Sketching of Urban Procedural Models
Interactive Sketching of Urban Procedural Models

Gen Nishida\textsuperscript{1} \quad Ignacio Garcia-Dorado\textsuperscript{1} \quad Daniel G. Aliaga\textsuperscript{1} \\
Bedrich Benes\textsuperscript{1} \quad Adrien Bousseau\textsuperscript{2}

\textsuperscript{1}\textit{Purdue University} \quad \textsuperscript{2}\textit{Inria}
Outline

• **Motivation**
• Inverse Procedural Modeling
• Our Approach
  1) Offline: Grammar Snippets
     Training CNNs
  2) Online: Sketching
• Results
• Evaluation
• Conclusions
Motivation

3D modeling remains a difficult task
Previous Work

Procedural Modeling

• Generative 3D modeling
• Requires programming skills
• Hard to control the output

[Wonka et al. 2003]  [Muller et al. 2006]
Sketching

- Natural interface for humans
Previous Work

### Sketch-Based Modeling

- Teddy [Igarashi et al. 1999]
- Polyhedral scaffolds [Schmidt et al. 2009]
- Cross-section lines [Xu et al. 2014]
We Combine Sketching & Procedural Modeling

- Sketching: easy and intuitive
- Procedural modeling: detailed completion and reusability
Outline

- Motivation
- **Inverse Procedural Modeling**
- Our Approach
  1) Offline: Grammar Snippets
     Training CNNs
  2) Online: Sketching
- Results
- Evaluation
- Conclusions
Inverse Procedural Modeling

- Generate a grammar for a given input
- MCMC and SMC are too slow

We use the CNN to quickly solve the inverse problem

[Talton et al. 2011]
Outline

• Motivation
• Inverse Procedural Modeling
• **Our Approach**
  1) Offline: Grammar Snippets
     Training CNNs
  2) Online: Sketching
• Results
• Evaluation
• Conclusions
Our Approach

Offline Training

Grammar snippets

PM

Training images

Training

Recognition CNN

Mass

Roof

Parameter CNN

Mass 1

Mass 2

Root 1

Root 2

⋯
Our Approach

Offline Training

Grammar snippets → PM → Training images

Recognition CNN

Parameter CNN

Online Sketching

User sketch → Recognition snippet → Parameter Estimation

Snippet parameters

PM → 3D model → Generated grammar

Interactive Sketching of Urban Procedural Models
Our Approach

Off-line Training

Grammar snippets

PM

Training images

Recognition CNN

Mass

Roof

Parameter CNN

Mass 1

Roof 1

Mass 2

Roof 2

Online Sketching

User sketch

Recognized snippet

Parameter estimation

Snippet parameters

\( p1 = 0.1 \)

\( p2 = 0.6 \)

\( \ldots \)

PM

3D model

Refinement

\( \ldots \) Generated grammar

Interactive Sketching of Urban Procedural Models
Grammar Snippet

• We define the grammar snippets based on real buildings
Grammar Snippet

a) Building mass

b) Roof

c) Window

d) Ledge
Example Snippet Fragment

```xml
<param building_height="20"/>
...

<rule name="Start">
  ...
  ...
  <extrude height="building_height"/>
  <comp>
    <top name="TopFace"/>
    <side name="Façade"/>
  </comp>
</rule>
```
Example Snippet Fragment

```xml
<param building_height="20"/>
...

<rule name="Start">
  ...
  <extrude height="building_height"/>
  <comp>
    <top name="TopFace"/>
    <side name="Façade"/>
  </comp>
</rule>
```

Parameter
Example Snippet Fragment

```xml
<param building_height="20"/>
...
<rule name="Start">
  ...
  <extrude height="building_height"/>
  <comp>
    <top name="TopFace"/>
    <side name="Façade"/>
  </comp>
</rule>
```

Non-terminal
Example Snippet

```xml
<param slope="70"/>
<param roof_height="3"/>
<rule name="Start">
    <mansard slope="slope" height="roof_height"/>
</rule>
```
Example Snippet

```xml
<param slope="70"/>
<param roof_height="3"/>
<rule name="Start">
  <mansard slope="slope" height="roof_height"/>
</rule>
```

Example Snippet

Interactive Sketching of Urban Procedural Models
Example Snippet

```xml
<param slope="70"/>
<param roof_height="3"/>
<rule name="Start">
  <mansard slope="slope" height="roof_height"/>
</rule>
```
Our Approach

Offline Training
- Grammar snippets
- Mass
- Roof
- PM
- Training images
- Recognition CNN
- Mass
- Roof
- Parameter CNN
- Mass 1
- Mass 2
- Roof 1
- Roof 2

Online Sketching
- User sketch
- Recognition
- Recognized snippet
- Parameter Estimation
- Snippet parameters
- \( p1 = 0.1 \)
- \( p2 = 0.6 \)
- PM
- Refinement
- 3D model
- Generated grammar

Interactive Sketching of Urban Procedural Models
Training Images

- Custom sketch rendering
- Fixed camera position

Parameters

\[ p_1 = 0.85 \]
\[ p_2 = 0.74 \]

...
Our Approach

Offline Training

Grammar snippets

PM

Training images

Recognition CNN

Mass

Roof

Parameter CNN

Mass 1

Mass 2

Roof 1

Roof 2

Online Sketching

User sketch

Recognition

Parameter estimation

Snippet parameters

PM

3D model

... Generated grammar

Interactive Sketching of Urban Procedural Models
Recognition CNNs

• Find the best grammar snippet for the user sketch
• Based on AlexNet architecture [Krizhevsky et al. 2012]
• Output is a probability distribution over the snippets
Training Recognition CNNs

- Training from scratch
- Fine-tuning the pretrained weights took 1.5 hours

![Graph showing training from scratch with a loss of 0.78 and accuracy of 1.0](image1)

![Graph showing fine-tuning with a loss of 0.96 and accuracy of 1.0](image2)
Recognition CNNs

Input sketch → CNN →

<table>
<thead>
<tr>
<th>Probability</th>
<th>Snippets</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1: 0.999</td>
<td>![Snippet 1]</td>
</tr>
<tr>
<td>#2: 0.000</td>
<td>![Snippet 2]</td>
</tr>
<tr>
<td>#3: 0.000</td>
<td>![Snippet 3]</td>
</tr>
<tr>
<td>#4: 0.000</td>
<td>![Snippet 4]</td>
</tr>
<tr>
<td>#5: 0.000</td>
<td>![Snippet 5]</td>
</tr>
<tr>
<td>#6: 0.001</td>
<td>![Snippet 6]</td>
</tr>
</tbody>
</table>
Recognition CNNs

Input sketch → CNN → Probability

Snippets

<table>
<thead>
<tr>
<th>Snippet</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>0.000</td>
</tr>
<tr>
<td>#2</td>
<td>0.000</td>
</tr>
<tr>
<td>#3</td>
<td>0.000</td>
</tr>
<tr>
<td>#4</td>
<td>0.998</td>
</tr>
<tr>
<td>#5</td>
<td>0.000</td>
</tr>
<tr>
<td>#6</td>
<td>0.002</td>
</tr>
</tbody>
</table>
Recognition CNNs

Input sketch → CNN → Snippets

<table>
<thead>
<tr>
<th>Probability</th>
<th>Snippets</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1: 0.001</td>
<td><img src="Image" alt="Snippets" /></td>
</tr>
<tr>
<td>#2: 0.998</td>
<td><img src="Image" alt="Snippets" /></td>
</tr>
<tr>
<td>#3: 0.001</td>
<td><img src="Image" alt="Snippets" /></td>
</tr>
<tr>
<td>#4: 0.000</td>
<td><img src="Image" alt="Snippets" /></td>
</tr>
<tr>
<td>#5: 0.000</td>
<td><img src="Image" alt="Snippets" /></td>
</tr>
<tr>
<td>#6: 0.000</td>
<td><img src="Image" alt="Snippets" /></td>
</tr>
</tbody>
</table>
Recognition CNNs

- One recognition CNN per object type
- We use 4 recognition CNNs in total

Object types

a) Building mass 1 CNN
b) Roof 1 CNN
c) Window 1 CNN
d) Ledge 1 CNN
Our Approach

Offline Training

Grammar snippets → Training images

Training CNN

Recognition CNN

Parameter CNN

Online Sketching

User sketch → Recognized snippet → Snippet parameters

PM → 3D model

Parameter CNN

\( p_1 = 0.1 \)

\( p_2 = 0.6 \)

Refinement

Generated grammar
Parameter Estimation CNNs

- Our custom network: 3 convolutional layers and 2 fully connected layers
- Outputs are normalized parameter values
Training Parameter Estimation CNNs

- No pretrained weights is available
- Training from scratch
- Takes 3 hours using 40,000 images
Parameter Estimation CNNs

Input sketch → CNN → Parameter values

x: 0.321
y: 0.273
w: 0.341
d: 0.335
h: 0.312
Parameter Estimation CNNs

Input sketch → CNN → Parameter values

- \( x: 0.402 \)
- \( y: 0.324 \)
- \( w: 0.138 \)
- \( d: 0.132 \)
- \( h: 0.243 \)
Parameter Estimation CNNs

Input sketch → CNN → Parameter values

(x, y) = (0.398, 0.294)
w = 0.140
d = 0.332
h = 0.226
Parameter Estimation CNNs

- One parameter estimation CNN per grammar snippet
- We use 26 parameter estimation CNNs in total

<table>
<thead>
<tr>
<th>Object types</th>
<th>CNNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Building mass</td>
<td>6 CNNs</td>
</tr>
<tr>
<td>b) Roof</td>
<td>7 CNNs</td>
</tr>
<tr>
<td>c) Window</td>
<td>9 CNNs</td>
</tr>
<tr>
<td>d) Ledge</td>
<td>4 CNNs</td>
</tr>
</tbody>
</table>
Outline

• Motivation
• Inverse Procedural Modeling
• Our Approach
  1) Offline: Grammar Snippets
     Training CNNs
  2) Online: Sketching
• Results
• Evaluation
• Conclusions
Our Approach

Offline Training

Grammar snippets

Training images

Recognition CNN

Parameter CNN

Online Sketching

User sketch

Recognized snippet

Snippet parameters

Parameter CNN

Recognition CNN

3D model

Generated grammar
1. Sketching – Building Mass

1. Select an anchor region
2. Draw the mass shape
3. Snippet and its parameters are detected
1. Sketching – Roof

1. Select an anchor region
2. Draw the roof shape
3. Snippet and its parameter values are detected
Optional Refinement by MCMC

- A few iterations of a MCMC engine for refinement
- Distance transform is used to define a score function

<table>
<thead>
<tr>
<th></th>
<th>CNN only</th>
<th>CNN + MCMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. RMSE</td>
<td>0.024</td>
<td>0.021</td>
</tr>
<tr>
<td></td>
<td>0.039</td>
<td>0.027</td>
</tr>
<tr>
<td></td>
<td>0.027</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>0.020</td>
<td>0.018</td>
</tr>
<tr>
<td></td>
<td>0.028</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>0.026</td>
<td>0.025</td>
</tr>
</tbody>
</table>

Interactive Sketching of Urban Procedural Models
Optional Refinement by MCMC

• A few iterations of a MCMC engine for refinement
• Distance transform is used to define a score function

<table>
<thead>
<tr>
<th></th>
<th>CNN only</th>
<th>CNN + MCMC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. RMSE</td>
<td>0.024</td>
<td>0.021</td>
</tr>
<tr>
<td>Avg. RMSE</td>
<td>0.039</td>
<td>0.027</td>
</tr>
<tr>
<td>Avg. RMSE</td>
<td>0.020</td>
<td>0.024</td>
</tr>
<tr>
<td>Avg. RMSE</td>
<td>0.028</td>
<td>0.018</td>
</tr>
<tr>
<td>Avg. RMSE</td>
<td>0.026</td>
<td>0.024</td>
</tr>
<tr>
<td>Avg. RMSE</td>
<td>0.026</td>
<td>0.025</td>
</tr>
</tbody>
</table>
2. Procedural Model Generation

```xml
<extrude height="building_height"/>
<comp>
  <top name="TopFace"/>
  <side name="Façade"/>
</comp>

<rule name="TopFace">
  <mansard slope="s" height="h"/>
</rule>
```
2. Procedural Model Generation

```xml
<extrude height="building_height"/>
<comp>
  <top name="TopFace"/>
  <side name="Façade"/>
</comp>

<rule name="Start">
  <mansard slope="s" height="h"/>
</rule>
```
2. Procedural Model Generation

```xml
<extrude height="building_height"/>
<comp>
  <top name="TopFace"/>
  <side name="Façade"/>
</comp>

<rule name="Start">
  <mansard slope="s" height="h"/>
</rule>
```
2. Procedural Model Generation

```
<extrude height="building_height"/>
<comp>
  <top name="TopFace"/>
  <side name="Façade"/>
</comp>

<rule name="Start">
  <mansard slope="s" height="h"/>
</rule>
```
2. Procedural Model Generation

```xml
<extrude height="building_height"/>
<comp>
  <top name="TopFace"/>
  <side name="Façade"/>
</comp>

<rule name="TopFace">
  <mansard slope="s" height="h"/>
</rule>
```
2. Procedural Model Generation

```xml
<extrude height="building_height"/>
<comp>
  <top name="TopFace"/>
  <side name="Façade"/>
</comp>

<rule name="TopFace">
  <mansard slope="s" height="h"/>
</rule>
```
2. Procedural Model Generation

Derivation Tree

- Building Mass
  - Façade
  - TopFace
  - Mansard Roof
2. Procedural Model Generation

Derivation Tree

- Building Mass
- Façade
- TopFace
- Mansard Roof
2. Procedural Model Generation

Derivation Tree

- Building Mass
- Façade
- TopFace
- Mansard Roof

Interactive Sketching of Urban Procedural Models
2. Procedural Model Generation

Derivation Tree
2. Procedural Model Generation

Derivation Tree

- Building Mass
  - Façade
  - TopFace
    - Mansard Roof
2. Procedural Model Generation

Derivation Tree

Building Mass

Façade

TopFace

Mansard Roof
3. Repetition

• Split rules for regular façade styles

```
<split axis="y">
  <param name="Floor" repeat/>
</split>
```

```
<split axis="y">
  <param name="GFloor"/>
  <param name="Ledge"/>
  <param name="Floor" repeat/>
</split>
```
3. Repetition

- Split rules for regular façade styles

```xml
<spli t axis="y">
  <param name="GFloor"/>
  <param name="Ledge"/>
  <param name="Floor"/>
  <param repeat>
    <param name="Ledge"/>
    <param name="Floor"/>
  </param repeat>
</split>
```
3. Repetition

• Split rules for regular façade styles

```xml
<rule name="Tile">
  <split axis="x">
    <param name="W"/>
    <param name="Tile" repeat>
      <param name="W"/>
    </param repeat>
  </split>
</rule>
```

...
3. Repetition

• Split rules for regular façade styles

```xml
<spl<i split axis="x">  
   <param name="W"/>
   <repeat>
      <param name="Tile1"/>
      <param name="Tile2"/>
   </repeat>
   <param name="W"/>
</split>
<rule name="Tile1">
   <split axis="x">
      ...
   </split>
```

4. Final Model Previewing

- Model preview
- Predefined default grammar snippets and parameter values are applied
5. Manual Grammar Editing

The user can directly change the parameter values

Grammar

```xml
...<param roof_slope="20"/>
...
<rule name="TopFace">
  <roofGable slope="roof_slope"/>
</rule>
```

5. Manual Grammar Editing

The user can directly change the parameter values

Grammar

```xml
...<param roof_slope="50"/>
...<rule name="TopFace">
    <roofGable slope="roof_slope"/>
</rule>
```
5. Manual Grammar Editing

The user can directly change the parameter values

Grammar

...  
<param panel_height="0.1"/>
...
<rule name="RoofPanel">
  <extrude height="panel_height"/>
</rule>
5. Manual Grammar Editing

The user can directly change the parameter values

**Grammar**

```xml
...<param panel_height="1.0"/>...
<rule name="RoofPanel">
  <extrude height="panel_height"/>
</rule>
```
6. Interactive Grammar Editing

Select, move, resize, delete, copy and paste, and undo functions

• Resize

• Copy and paste
Outline

• Motivation
• Inverse Procedural Modeling
• Our Approach
  1) Offline: Grammar Snippets
     Training CNNs
  2) Online: Sketching
• Results
• Evaluation
• Conclusions
Results
Results

Interactive Sketching of Urban Procedural Models
Results

• Creation time was around 5 minutes

• Offline training
  • 1.5 hours per recognition CNN
  • 3 hours per parameter estimation CNN
  • 3.5 days in total to train 30 CNNs using GPU, GeForce GTX 980
Expressivity

• 20 building photos from ImageNet

Source image

Result
Expressivity

• 20 building photos from Flickr

Source image

Result
Variability

3D geometry can be modified by changing the grammar
Outline

• Motivation
• Inverse Procedural Modeling
• Our Approach
  1) Offline: Grammar Snippets
     Training CNNs
  2) Online: Sketching
• Results
• Evaluation
• Conclusions
Evaluation

Comparison to MCMC

<table>
<thead>
<tr>
<th>Our approach</th>
<th>Avg. time [sec]</th>
<th>Avg. RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.29</td>
<td>0.021</td>
</tr>
<tr>
<td></td>
<td>0.29</td>
<td>0.027</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0.018</td>
</tr>
<tr>
<td></td>
<td>0.32</td>
<td>0.024</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0.025</td>
</tr>
<tr>
<td>Avg.</td>
<td>0.30</td>
<td>0.023</td>
</tr>
</tbody>
</table>
### Evaluation

#### Comparison to MCMC

<table>
<thead>
<tr>
<th></th>
<th>Our approach</th>
<th>MCMC 100 iterations</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. time [sec]</td>
<td>0.29</td>
<td>2.81</td>
<td>2.87</td>
</tr>
<tr>
<td>Avg. RMSE</td>
<td>0.021</td>
<td>0.20</td>
<td>0.23</td>
</tr>
</tbody>
</table>

**Note:** MCMC results are averaged over 100 iterations.
# Evaluation

## Comparison to MCMC

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.29</td>
<td>0.021</td>
<td>2.81</td>
<td>0.20</td>
<td>27.93</td>
<td>0.15</td>
<td>30.11</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>0.29</td>
<td>0.027</td>
<td>2.78</td>
<td>0.27</td>
<td>28.84</td>
<td>0.13</td>
<td>29.38</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0.024</td>
<td>2.87</td>
<td>0.19</td>
<td>29.31</td>
<td>0.13</td>
<td>29.35</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0.018</td>
<td>2.85</td>
<td>0.20</td>
<td>30.11</td>
<td>0.11</td>
<td>29.35</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0.024</td>
<td>3.05</td>
<td>0.23</td>
<td>29.35</td>
<td>0.15</td>
<td>29.35</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>0.30</td>
<td>0.025</td>
<td>2.89</td>
<td>0.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MCMC 100 iterations</th>
<th>Avg. time [sec]</th>
<th>Avg. RMSE</th>
<th>MCMC 1000 iterations</th>
<th>Avg. time [sec]</th>
<th>Avg. RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.30</td>
<td>0.023</td>
<td></td>
<td>29.16</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Average time comparison for different methods and iterations, showing significant improvements with our approach compared to MCMC.
Robustness

• Our method is robust to varying input sketches
Robustness

• If an undefined shape is sketched, a similar one will be detected
User Study

Quantitative evaluation

- After a short tutorial, the participants had to reproduce two similar target buildings using our system

<table>
<thead>
<tr>
<th>Target building</th>
<th>#clicks</th>
<th>Creation time [min]</th>
<th>#rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>a)</td>
<td>39</td>
<td>6.7</td>
<td>94</td>
</tr>
<tr>
<td>b)</td>
<td>42</td>
<td>6.2</td>
<td>72</td>
</tr>
</tbody>
</table>
User Study

Qualitative evaluation

Target building

a) |

Buildings created by the users

b)
Outline

• Motivation
• Inverse Procedural Modeling
• Our Approach
  1) Offline: Grammar Snippets
     Training CNNs
  2) Online: Sketching
• Results
• Evaluation
• Conclusions
Contributions

- Combination of sketching and procedural modeling
  - Intuitive by sketching interface
  - Grammar snippets complete details
  - Requires no programming skills

- Grammar snippets to increase the variation in the output grammar

- Using CNNs to solve inverse procedural modeling
Limitations

Interactive Sketching of Urban Procedural Models
Future Work

• Automatic decomposition of existing grammars for generating grammar snippets

• Support arbitrary viewpoints

• Less GPU memory usage
  One parameter estimation CNN for all the snippets
Acknowledgement

• Anonymous reviewers
• Jennifer Neville and Hogun Park
• User study participants
• NSF CBET 1250232
• NSF IIS 1302172
• Google Research Award
• French ANR project SEMAPOLIS (ANR-13-CORD-0003)
Thanks

Web: http://cs.purdue.edu/cgvlab/urban/
Email: gnishida@purdue.edu