Urban Grammar

Nate Andrysco
Daniel Aliaga
Chris Hoffmann
Urban Visualization

- Procedural creation of urban layouts for:
  - City planning.
  - Creation of virtual environments for games.
  - Emergency response training.
  - Fast prototyping.

- Urban Grammar deals with the modification of urban layouts.
Example: Stretch Lafayette
Data Representation

- Specification
  - An aerial view is marked up with lines denoting roads or boundaries.

- Parsing
  - The specification is then parsed to create a city grammar.

- Deriving
  - When the city is changed, the grammar is used to derive a new city image.
Urban Specification

- Specify an initial region that encloses the buildings you wish to include.
Urban Specification

- Ideally, extract automatically from GIS database roads and other boundaries.
  - For now, we mark them manually in a top-down manner.
  - As you add edges you can see the tuples that are being created.
Urban Specification
Urban Specification

- Lafayette

Courtesy of Shweta Svaidya
Urban Specification

- Rome

Courtesy of Shweta Svaidya
Urban Specification

- Madrid

Courtesy of Shweta Svaidya
Urban Specification

- Buenos Aires

Courtesy of Shweta Svaidya
Urban Specification

- Paris
Parsing the City

- The top-down approach of marking up the city is key to parsing the city.
- Start by looking at the initial region and find a markup edge that splits the region in (approximately) half.
- Recurse on each of these regions and find edges that split them.
- Do this until all edges have been used.
Parsing the City

- When a tuple is divided, a rule is created.
  - The rule consists of the tuple’s geometry, its location, and the line (partition) that divided it.
  - A rule has 2 children, either more rules or terminals.

- If a tuple cannot be divided anymore, it is a terminal (0 children).
Parsing the City
Parsing the City

A → BC

A

B

C
Parsing the City

A → BC
B → DE
C → FG
Parsing the City

A → BC
B → DE
C → FG
D → HI
E → JK
Parsing the City
Deriving an Edited City

- If a region appears to have stretched or shrunk a significant amount, find the number of times to apply a rule so that distortion is minimized.

  □ Note: an unmodified city’s derivation should be the same as the original specification.
Deriving an Unmodified City
Deriving an Unmodified City
Stretch in 1D

Original

Scaled width by 3
Stretch in 2D

Original

Scaled by 3
Stretching Rome

Scaled by 3
Reducing Distortion

- Granularity
  - If a region is stretched, we can choose to only apply the partition once.
  - This passes the work of further dividing the tuple onto the next rule in the tree.

- Terminal Matching
  - Find a terminal that best fits a given area.
Granularity Example

- Original Method

Original Region  Stretch  After application of first rule
Undivided Stretched Region

Result after application of next rules
Granularity Example

- Work pushed to terminal level

Original Region

Stretch

After application of first rule

Result after application of next rules

Comparison of previous method
Reducing Distortion

<table>
<thead>
<tr>
<th>Scale of Original Region</th>
<th>Average Tuple Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.05</td>
</tr>
<tr>
<td>1.1</td>
<td>0.1</td>
</tr>
<tr>
<td>1.21</td>
<td>0.15</td>
</tr>
<tr>
<td>1.331</td>
<td>0.2</td>
</tr>
<tr>
<td>1.4641</td>
<td>0.25</td>
</tr>
<tr>
<td>1.61051</td>
<td>0.3</td>
</tr>
<tr>
<td>1.77156</td>
<td>0.35</td>
</tr>
<tr>
<td>1.94872</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Legend:
- Green: Regular
- Pink: Auto Granularity
- Red: Terminal Best Fit
Urban Simplification

- Motivation
  - Want to have interactive rates.
    - For large cities we may have hundreds of thousands of terminals and hundreds of thousands of rules.
    - Displaying every unique terminal may tax the GPU.
  - Want to extrapolate interesting data from each city.

- Solution
  - These problems can be solved by simplifying the parse tree.
Terminal Simplification

- Group tuples that are similar to each other.
- Designate one (or more) tuples of the group to be used whenever a terminal is needed from the group.
Terminal Simplification

```
  c
 / \
/   \ 
/     \
/       \
/         \
/           \
/             \
/               \
/                 \
/                   \
/                     \
/                     \
a  b  d  e  f  g
```

```
  d
 / \
/   \ 
/     \
/       \
/         \
/           \
/             \
/               \
/                 \
/                   \
/                     \
/                     \
a  f  d  e  f  f
```
Terminal Simplification
Procedural Simplification

- Combine rules that are similar.
  - Are the tuples similar?
  - Are the partition lines similar?
Procedural Simplification

1000 Rules

\[
\begin{align*}
A & \rightarrow BC \\
B & \rightarrow DE \\
C & \rightarrow FG \\
\vdots & \\
X & \rightarrow YZ \\
\vdots &
\end{align*}
\]

5 Rules

\[
\begin{align*}
A & \rightarrow DN \\
D & \rightarrow LK \\
N & \rightarrow TX \\
L & \rightarrow QR \\
Q & \rightarrow YZ
\end{align*}
\]
Procedural Simplification

30 Rules

10 Rules

Rome
Procedural Simplification

Paris

77 Rules

19 Rules
Procedural Simplification

Buenos Ares

130 Rules

38 Rules
Tools

- Similarity Estimation
- N-gon mapping
Similarity Estimation

- Tuple similarity is a weighted combination of:
  - Shape/perimeter similarity.
  - Location similarity.
  - Size/radii similarity.

- Partition similarity is a weighted combination of:
  - Length similarity
  - Orientation similarity
Tuple Shape Similarity

- Use oriented bounding boxes.
  - Simplifies the computation to the comparison of two boxes.
  - Improved reliability over old method.
Oriented Bounding Box
Oriented Bounding Box
N-gon mapping

N-gon to M-gon mapping

- New tuples are derived that do not match the original tuples geometry.
- Can you map a hexagon to a square? Should this be allowed?
- Can we prevent tough cases by obtaining better derivations?
N-gon mapping

- Let $N > M$
  - Since the vertices of the $n$-gon contain the needed texture coordinates, only use those.
  - Map the $M$ vertices of the $m$-gon plus $(N-M)$ intermediate points.
  - Intermediate points are obtained by projecting the $n$-gon’s vertices onto the $m$-gon’s perimeter.
  - Rotate and scale to find a best fit.
Simple example
In the program
Stretch more
More Stretching
Problems

- For more complex scenes, the mapping looks even worse.
- Can still get seams in the texture.
Future Work

- Can you combine the layouts of two cities?
  - What would it look like if Lafayette wanted to incorporate the layout of Paris.

- Apply the framework to other images.
  - What might a famous painting look like if the artist had used a bigger canvas?

- Integrate with Build-by-Numbers to procedurally create full 3D cities.
Questions?