Urban Grammar

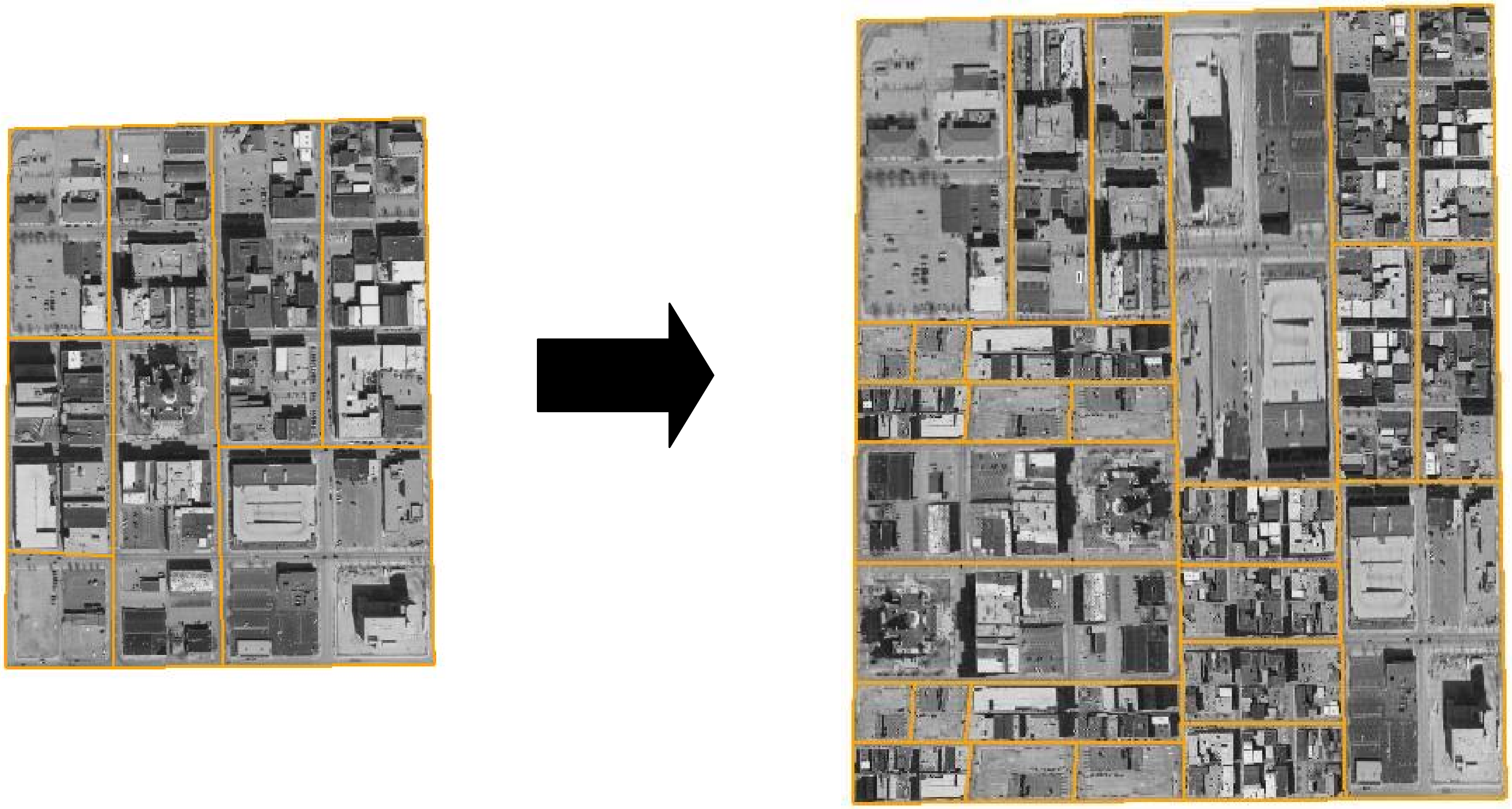
Nate Andrysco
Daniel Aliaga
Chris Hoffmann

Last Time…

- At the end of Spring Quarter Aliaga gave a talk about Urban Visualization.

- Urban Grammar is a way to visualize how a city might change.
  - What might a city look like as it expands?
  - What might a city look like if a completely different city's style was used?

Example: Stretch Lafayette



Aliaga’s Future Work

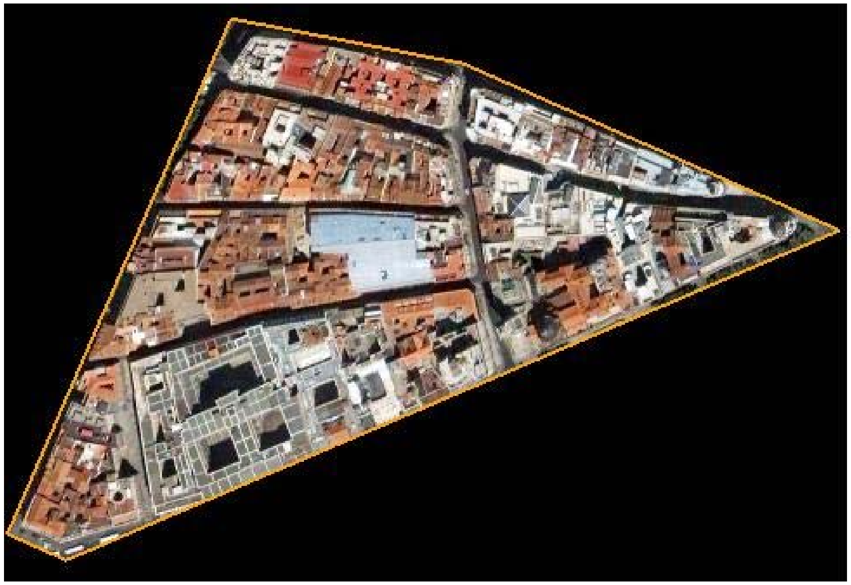
- Modify more interesting cities
- Procedural Simplification
- Full Inverse Modeling
  - Combine with Build-by-Numbers
- Applications
  - Road planning, growth algorithms, rapid prototyping

Data Representation

- Specification
  - An aerial view is marked up with lines.
  - The lines denote a road, land boundaries, or a building outline.
- 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

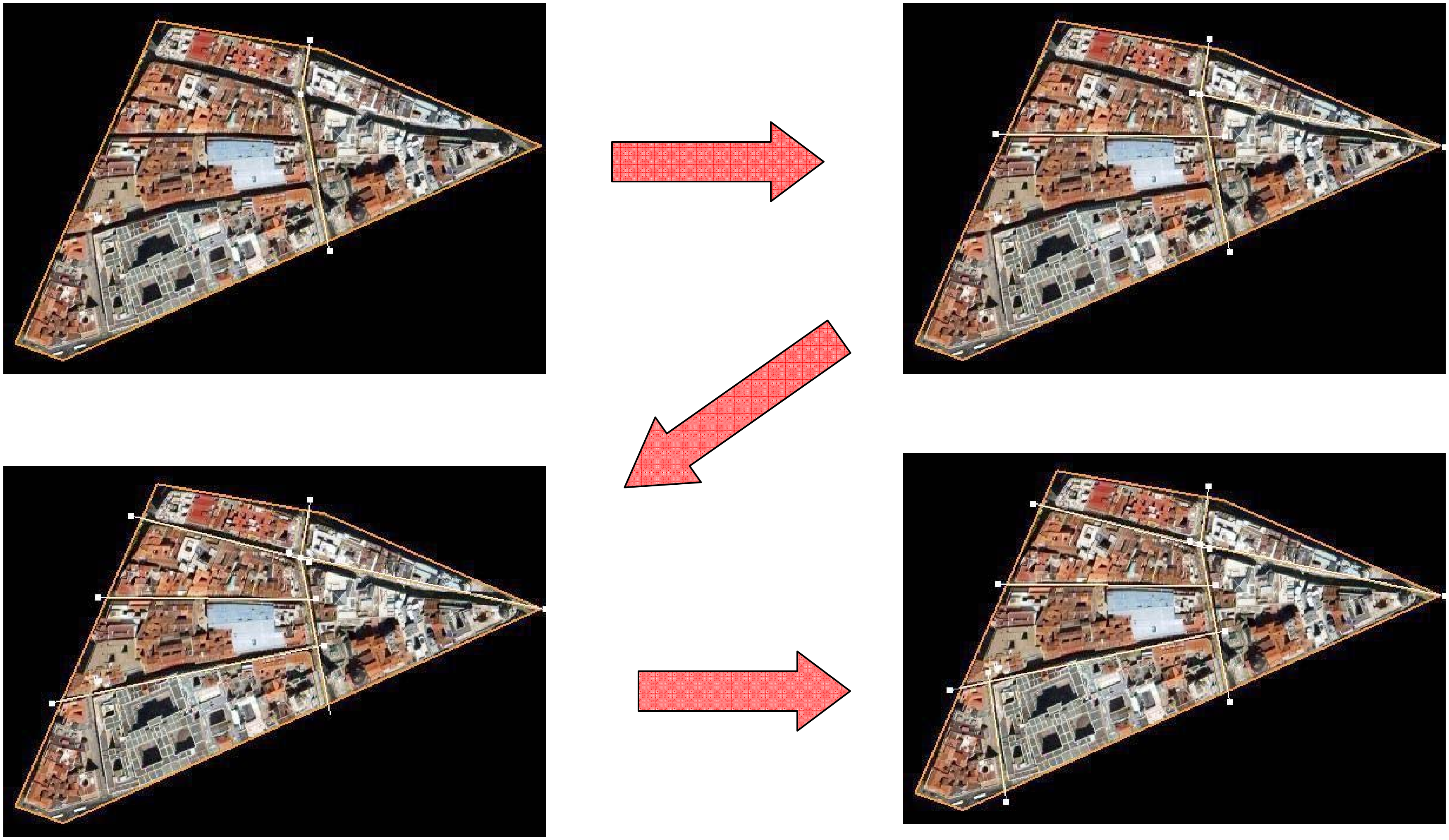
- Specify an initial region that encloses the buildings you wish to include.



Urban Specification

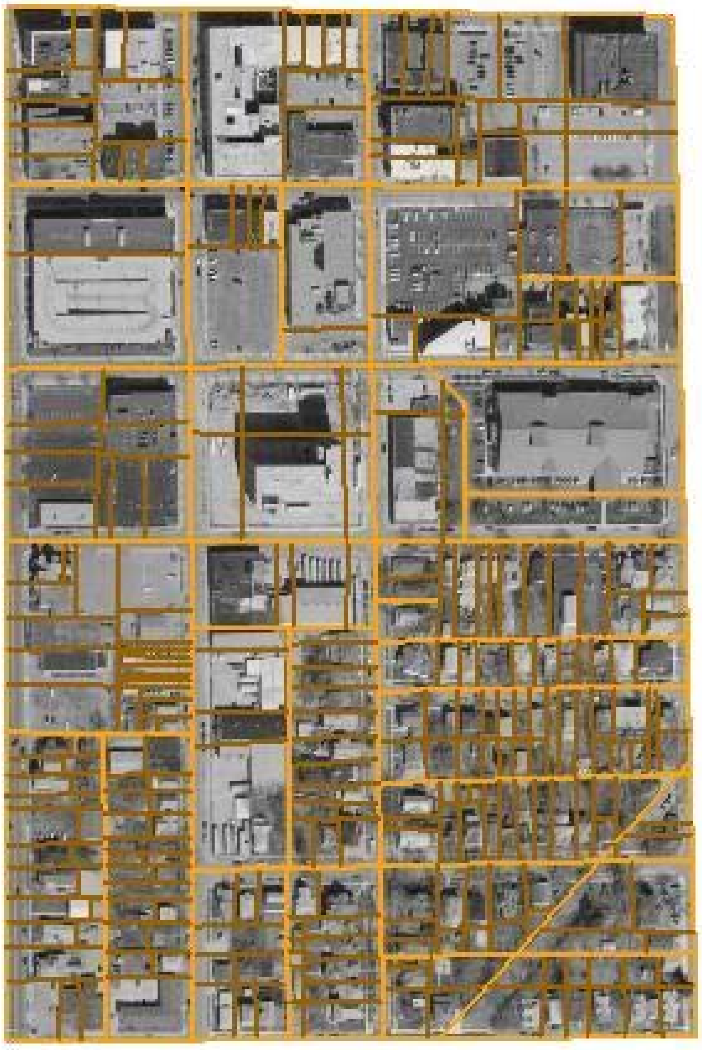
- Ideally, extract automatically from GIS database roads and other boundaries.
  - For now, we mark then manually in a top-down manor.
  - As you add edges you can see the tuples that are being created.

Urban Specification



Urban Specification

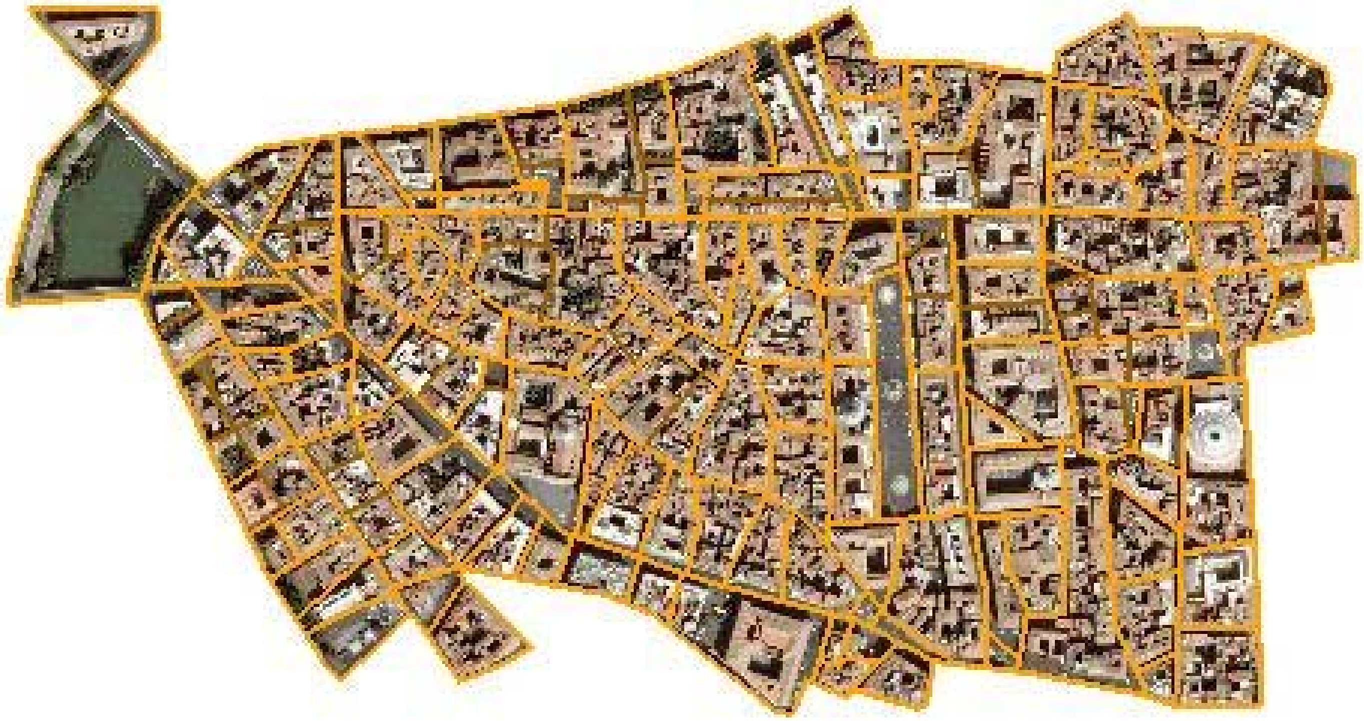
- Lafayette



Courtesy of Shweta Svaidya

Urban Specification

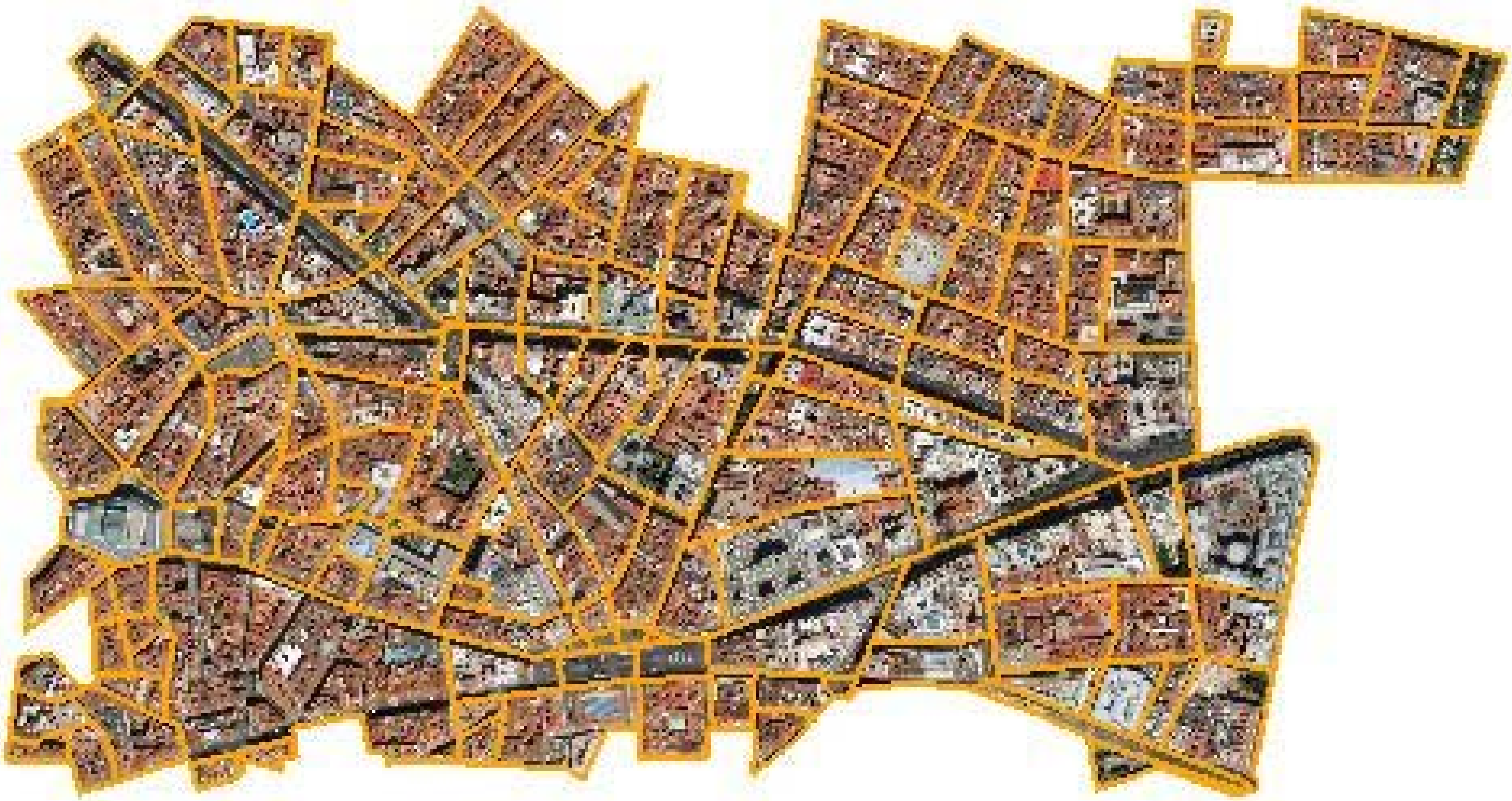
- Rome



Courtesy of Shweta Svaidya

Urban Specification

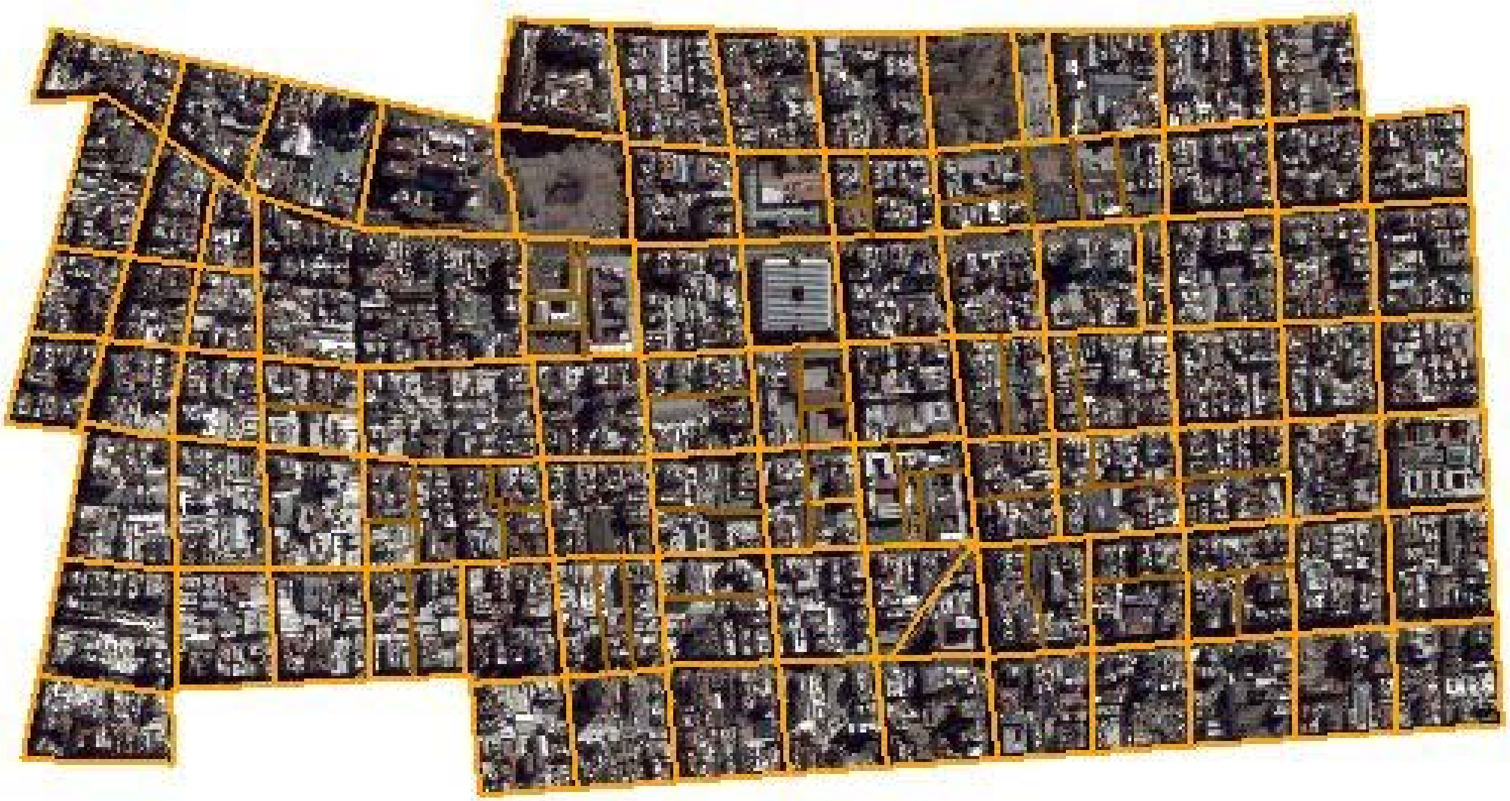
- Madrid



Courtesy of Shweta Svaidya

Urban Specification

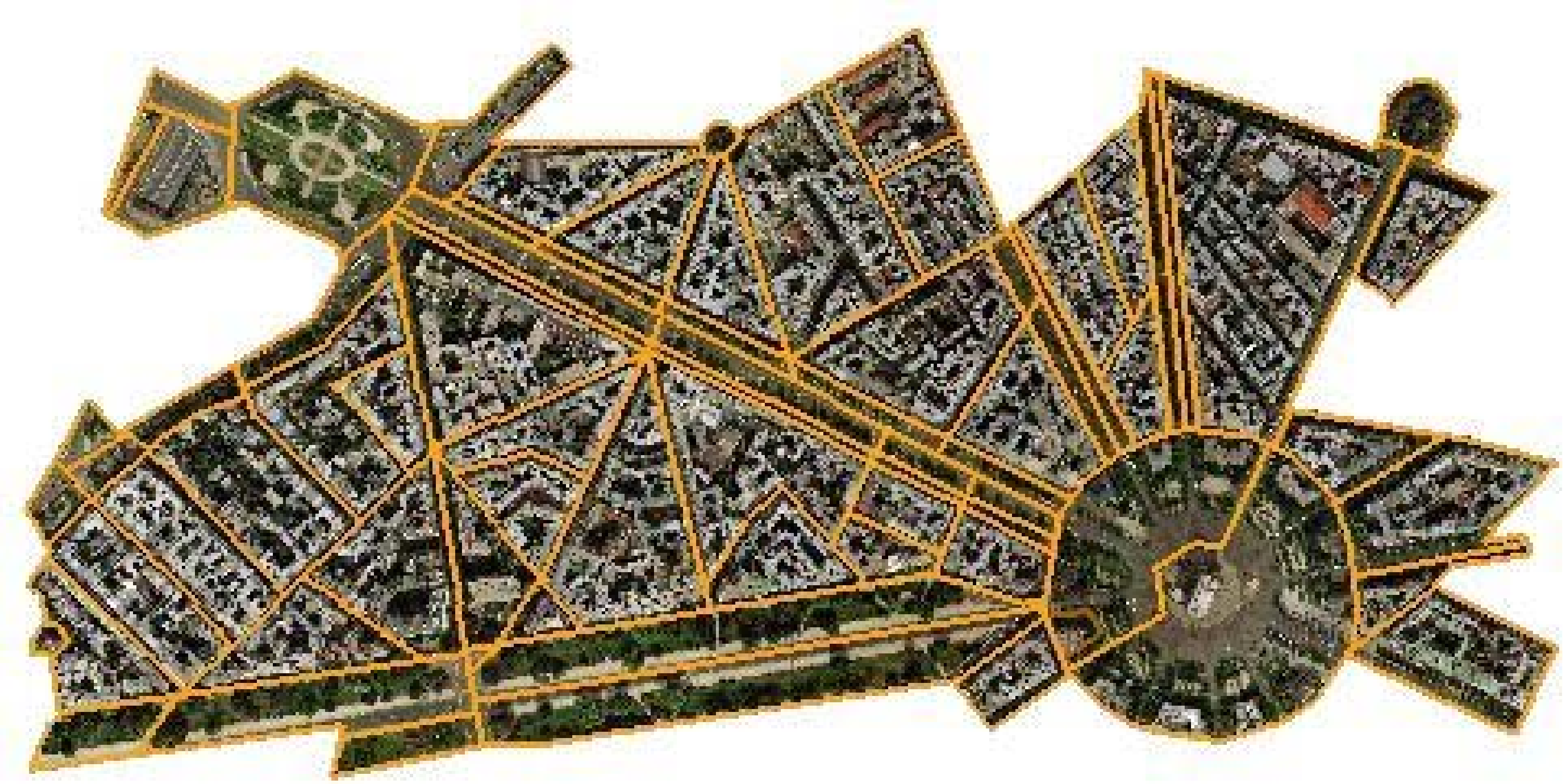
- Buenos Aires



Courtesy of Shweta Svaidya

Urban Specification

- Paris



Courtesy of Shweta Svaidya

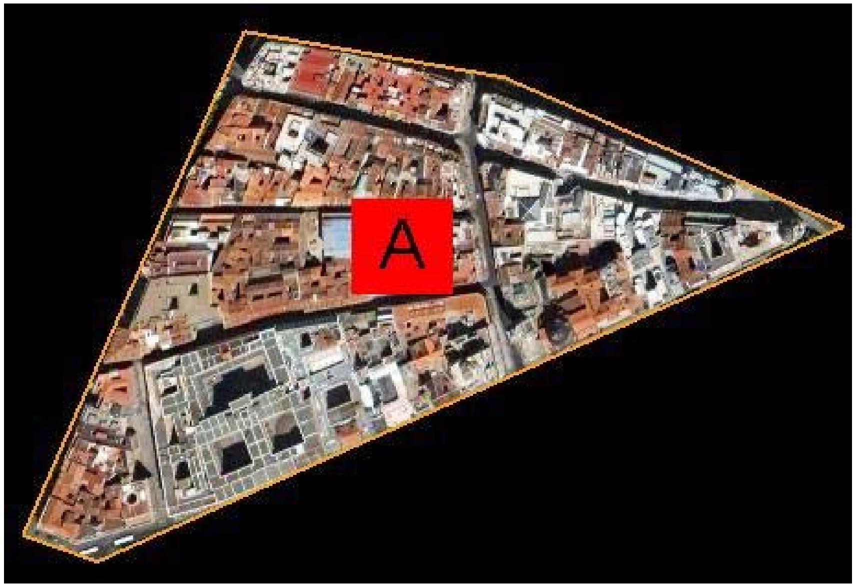
Parsing the City

- 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

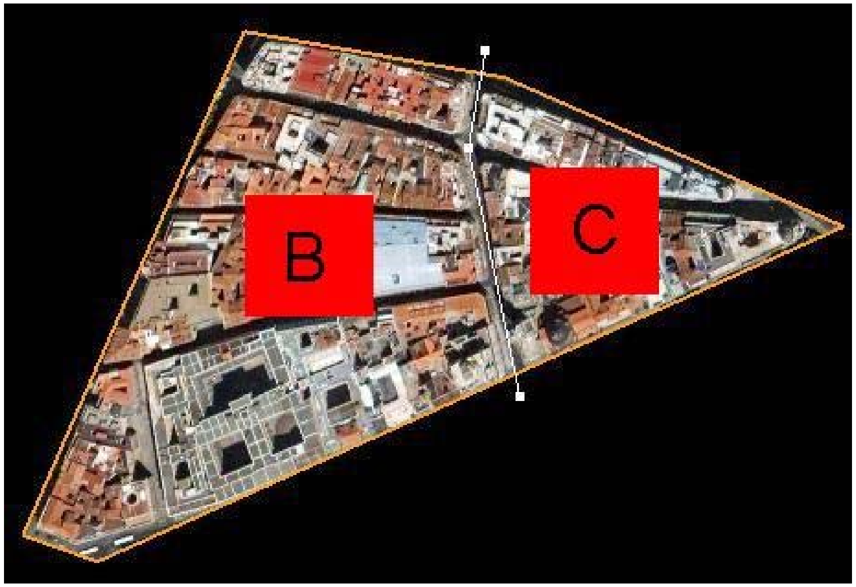
- When a tuple is divided, a rule is created.
  - The rule consists of the tuple’s geometry, its location, and the line (partition) that divided it.
- The rule has either zero or two children, creating a binary tree.
  - A rule has 0 children if it can not be divided any more (terminal)
  - Otherwise, its 2 children are either rules or terminals.

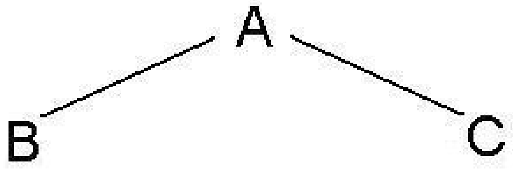
Parsing the City



A

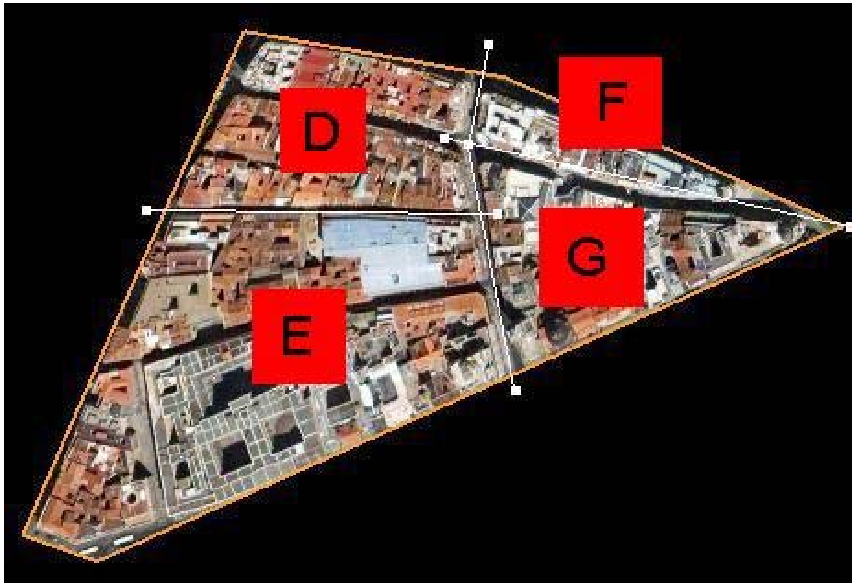
Parsing the City

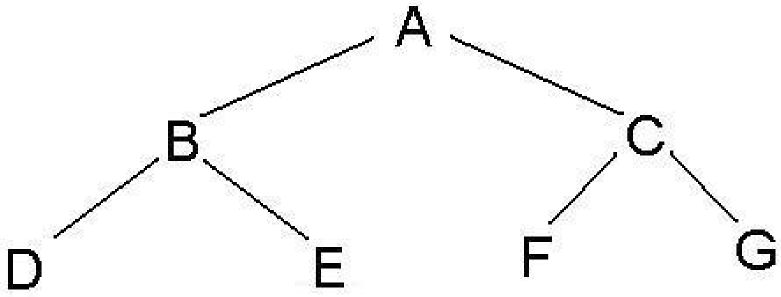




$A \rightarrow BC$

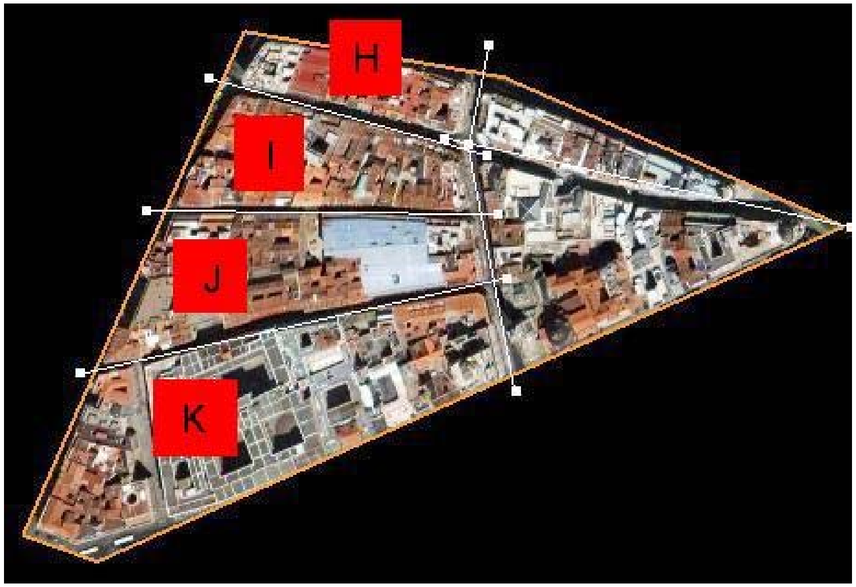
Parsing the City

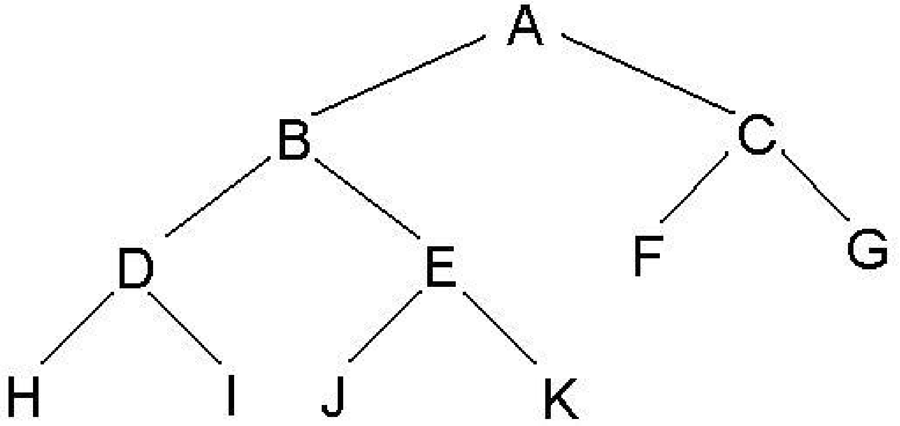




A → BC
B → DE
C → FG

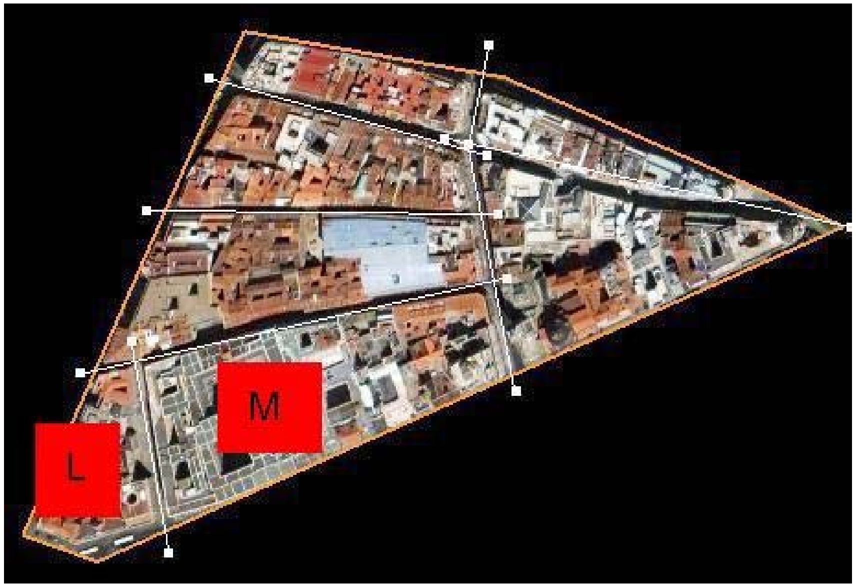
Parsing the City

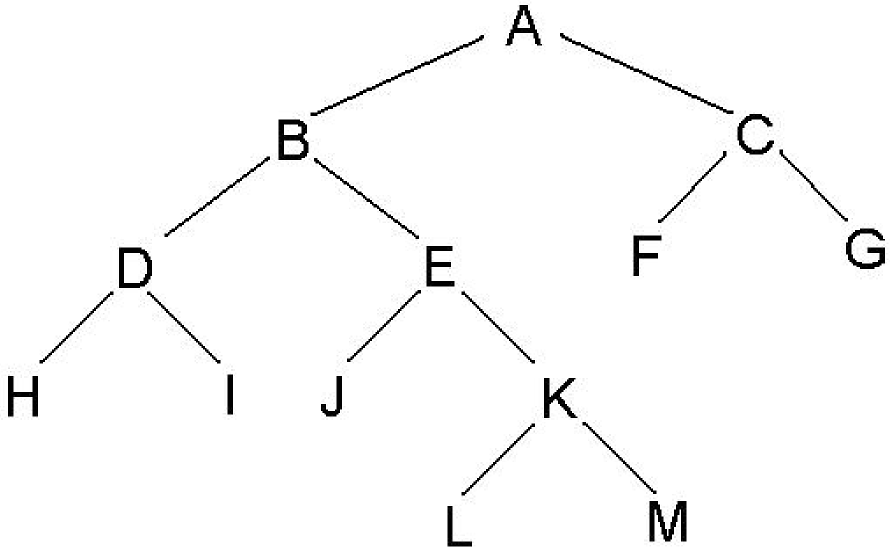




A → BC
B → DE
C → FG
D → H I
E → JK

Parsing the City



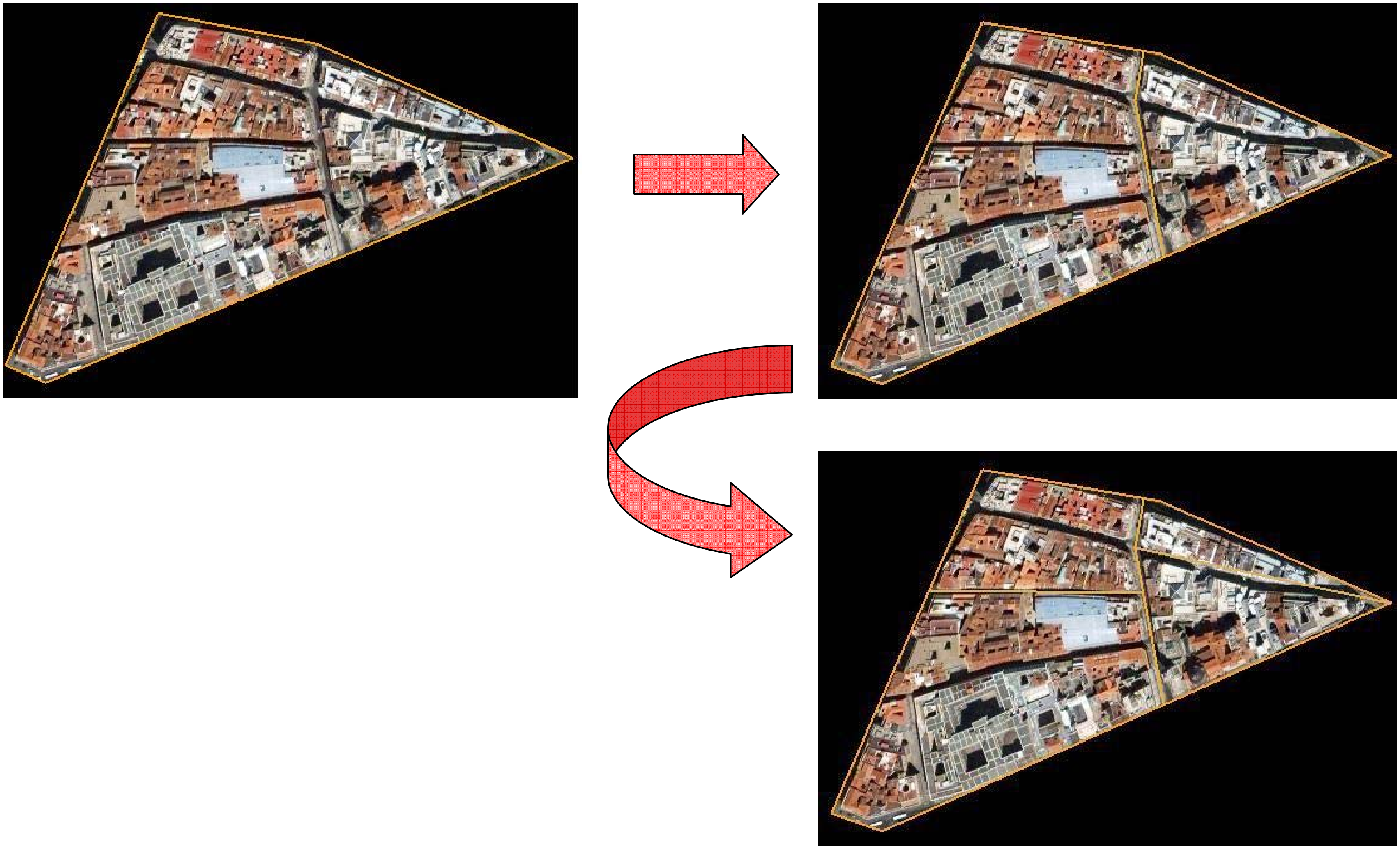


A → BC
B → DE
C → FG
D → HI
E → JK
K → LM

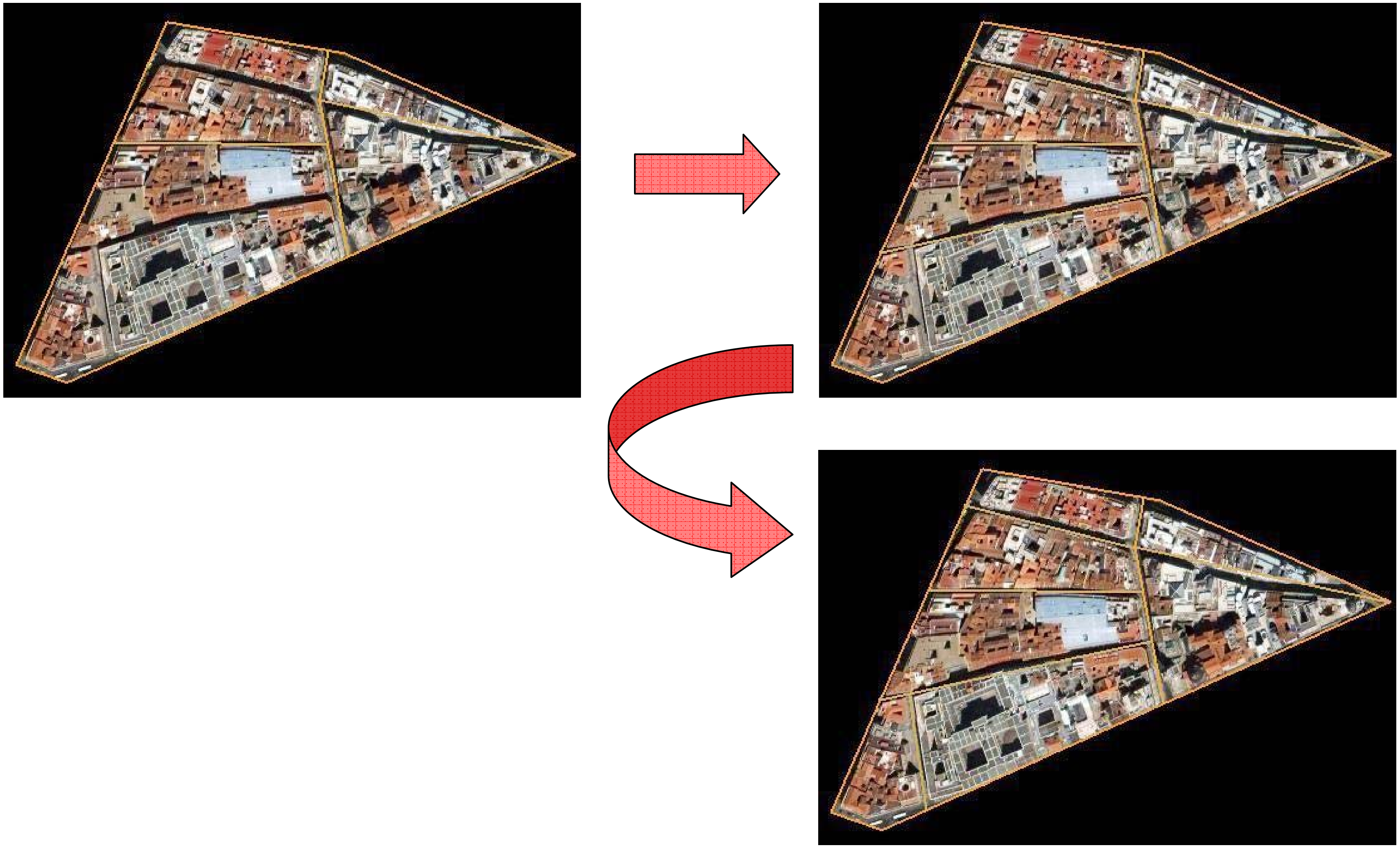
Deriving an Edited City

- If a region appears to have stretched or shrunk a significant amount, recursively re-apply or partially apply the best rule.
  - Note: an unmodified city's derivation should be the same as the original specification.

Deriving a City



Deriving a City



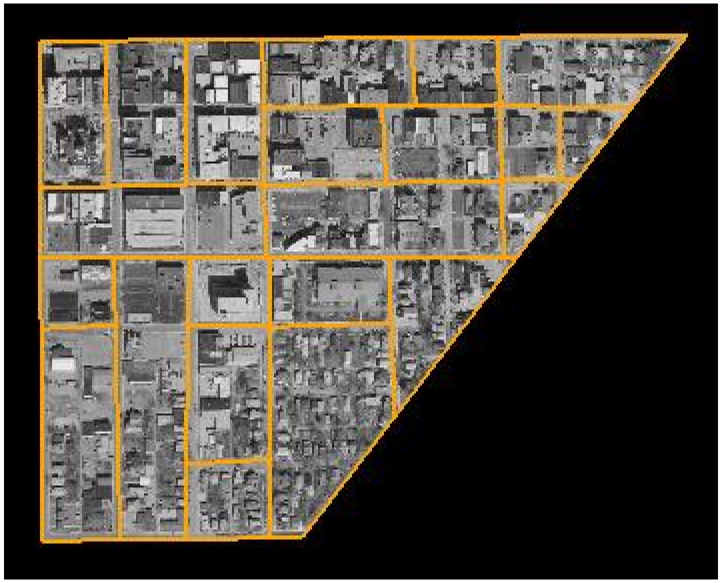
Urban Editing

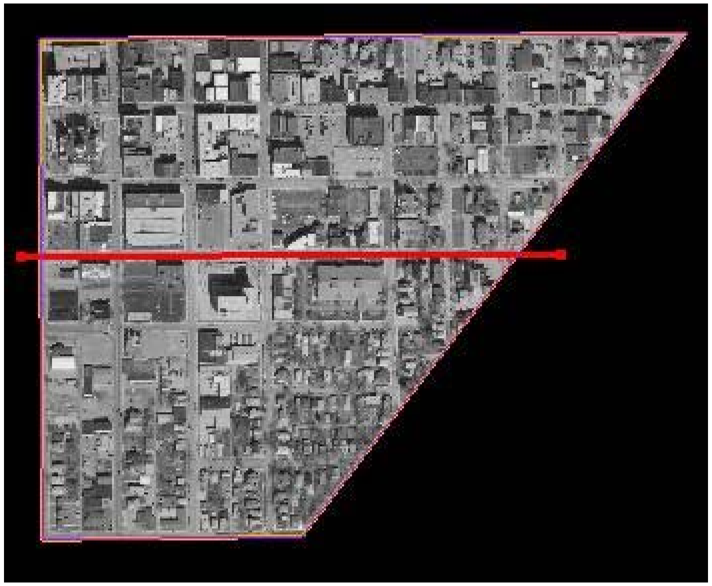
- What if a river was to erode away part of the city? What might the city look like?

- What if a major road was moved? What if a major road was added?

Urban Editing

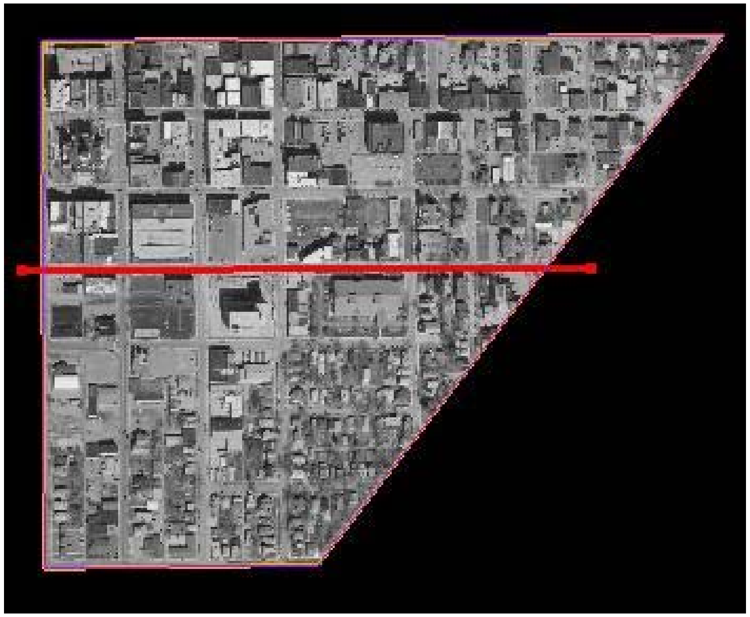
- Select an edge from the original specification.

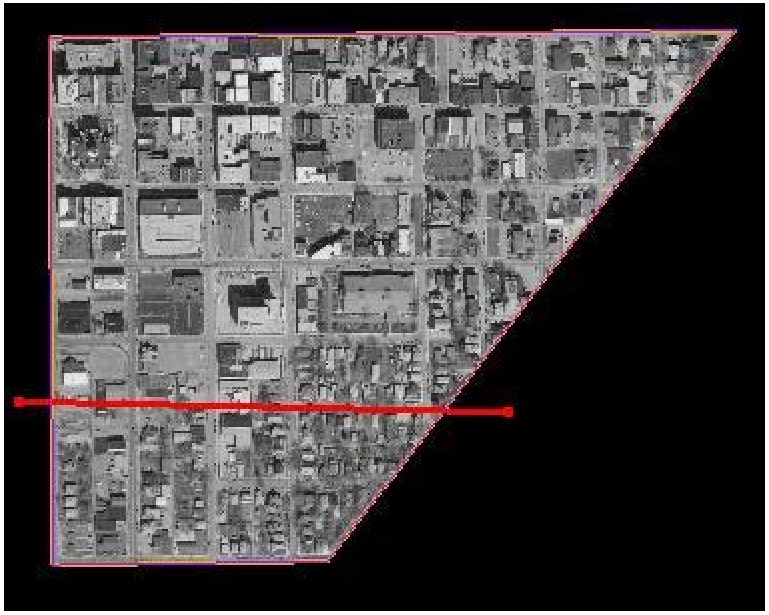




Urban Editing

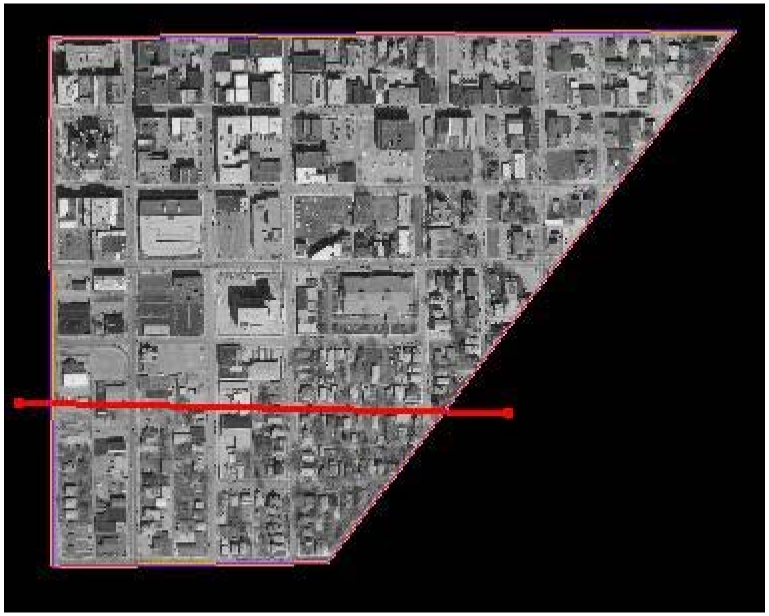
- Move the edge.





Urban Editing

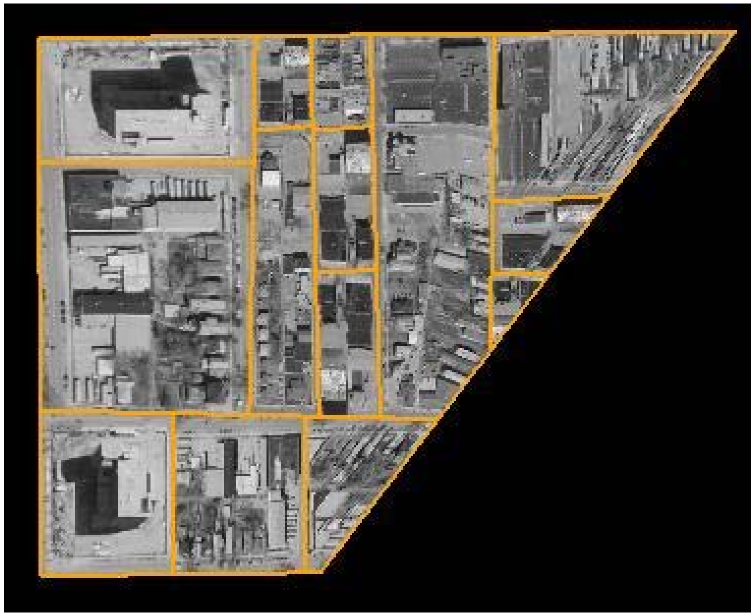
- Derive a new city.



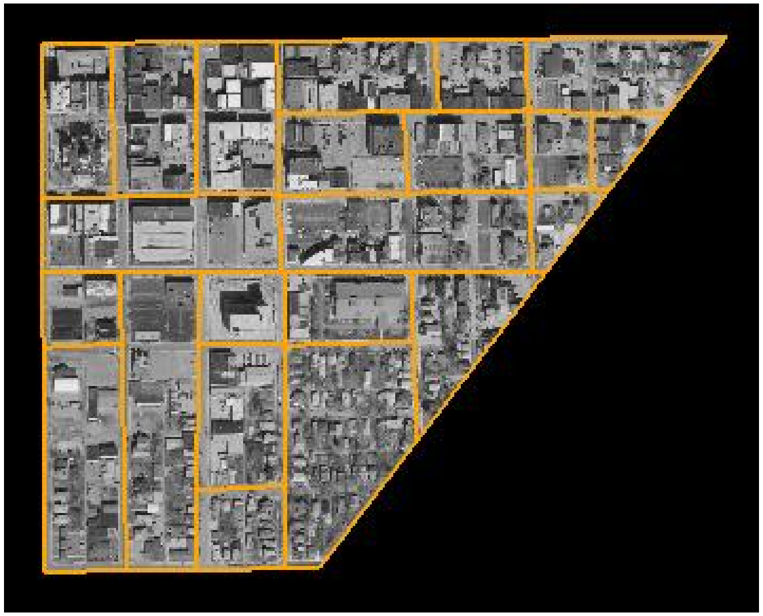
Urban Editing

- Derive a new city.

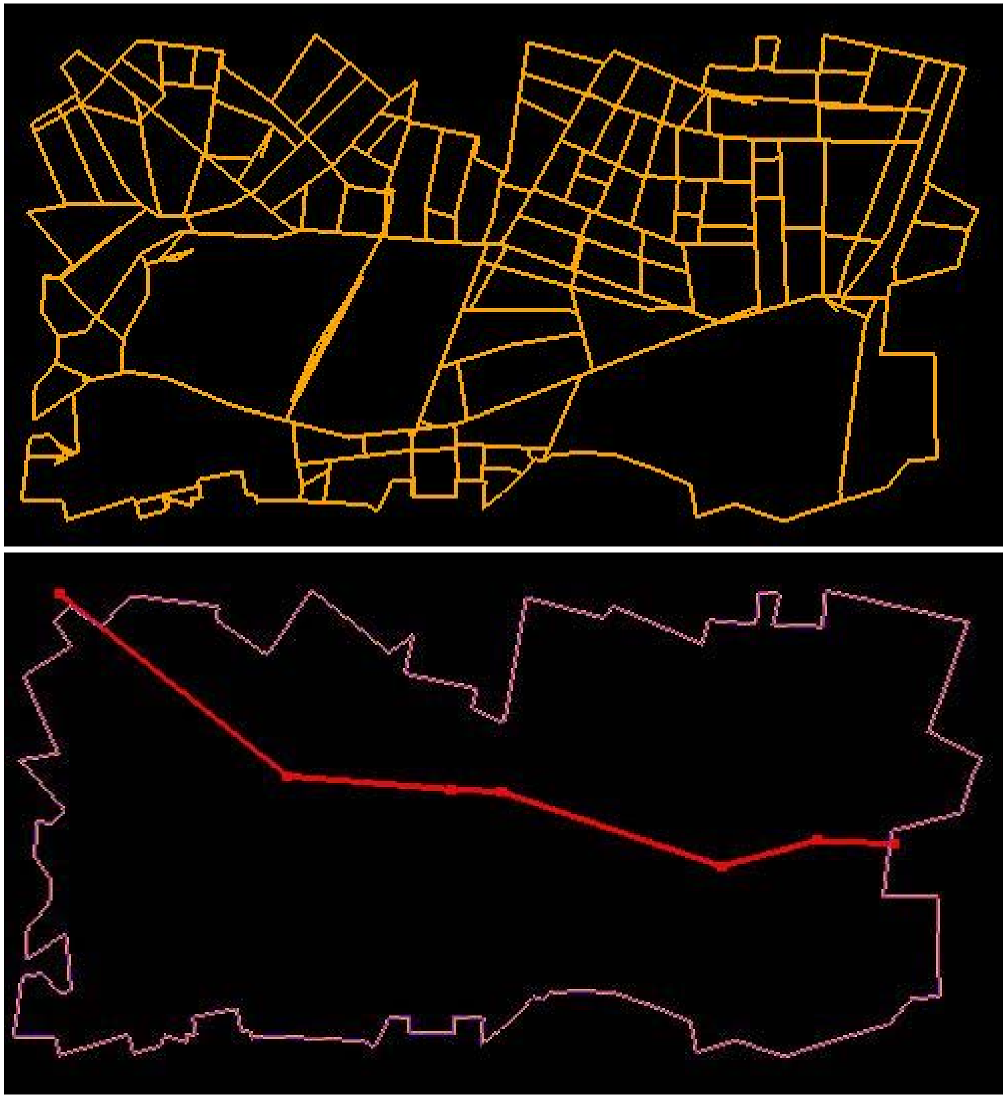
new



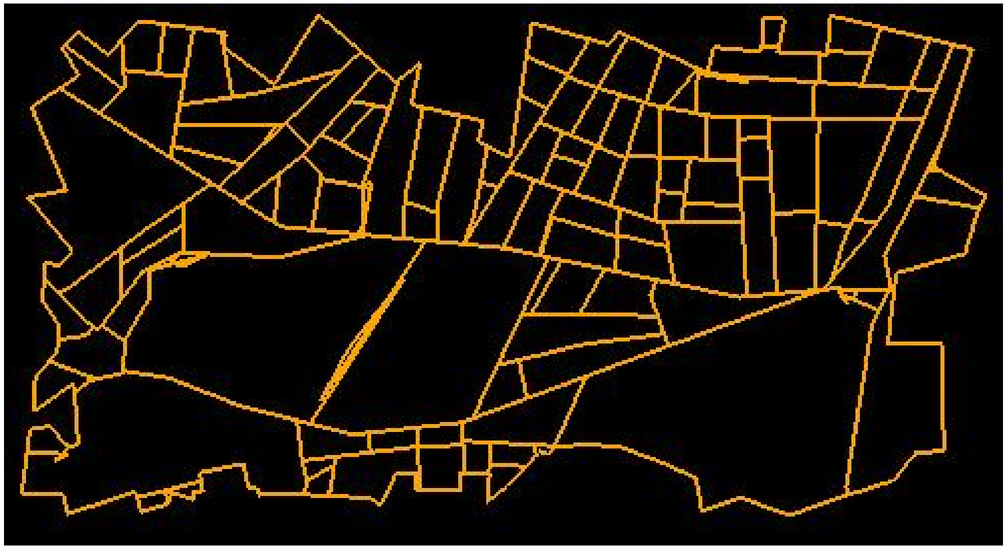
original

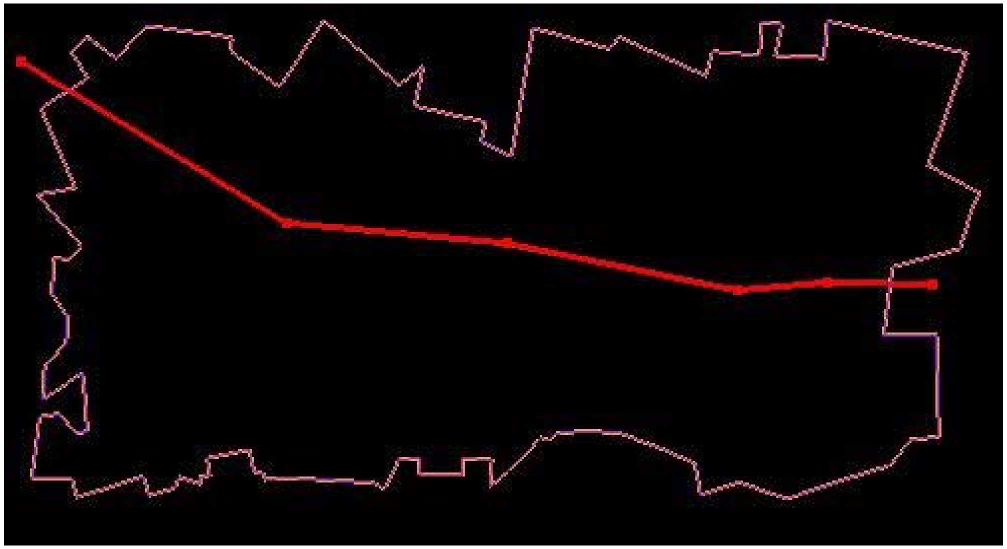


Urban Editing: Bigger City

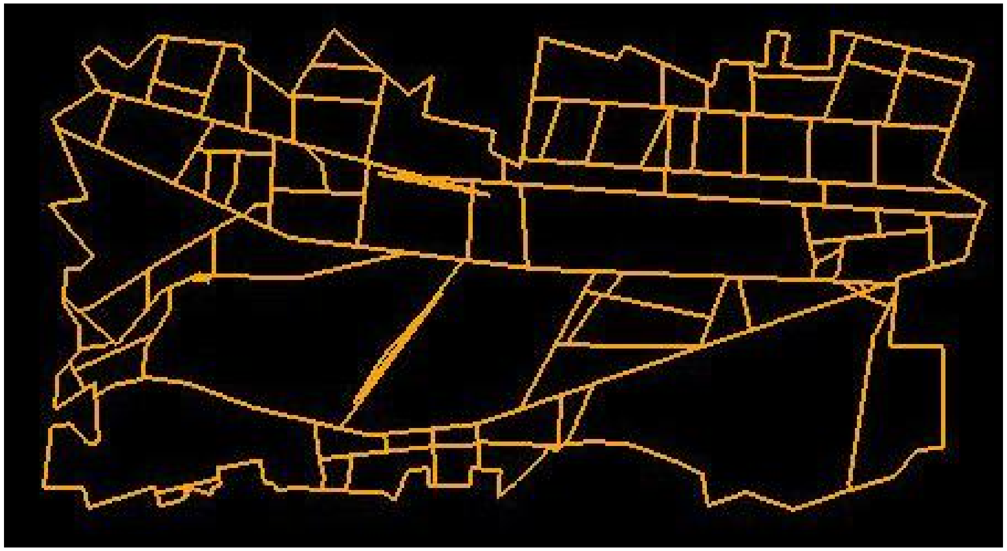


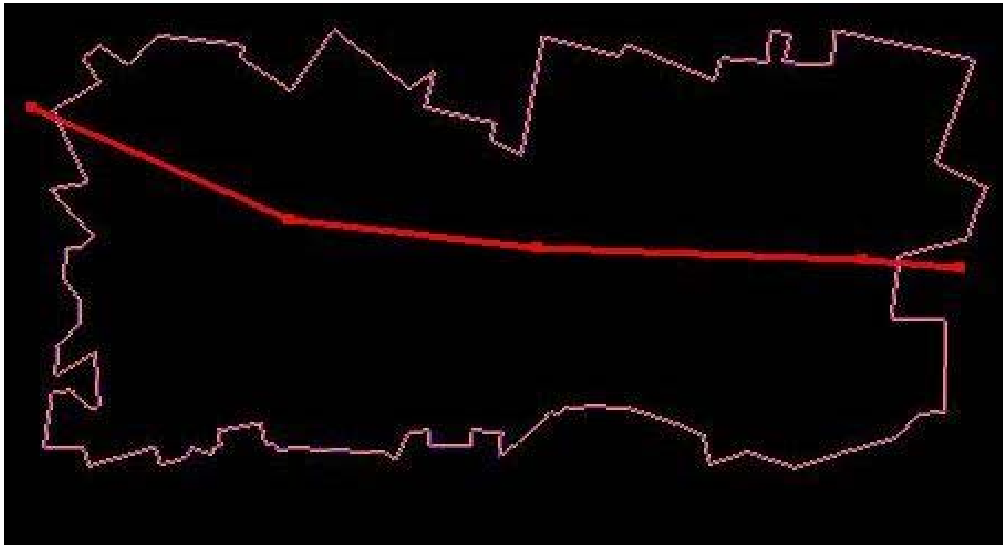
Urban Editing: Bigger City



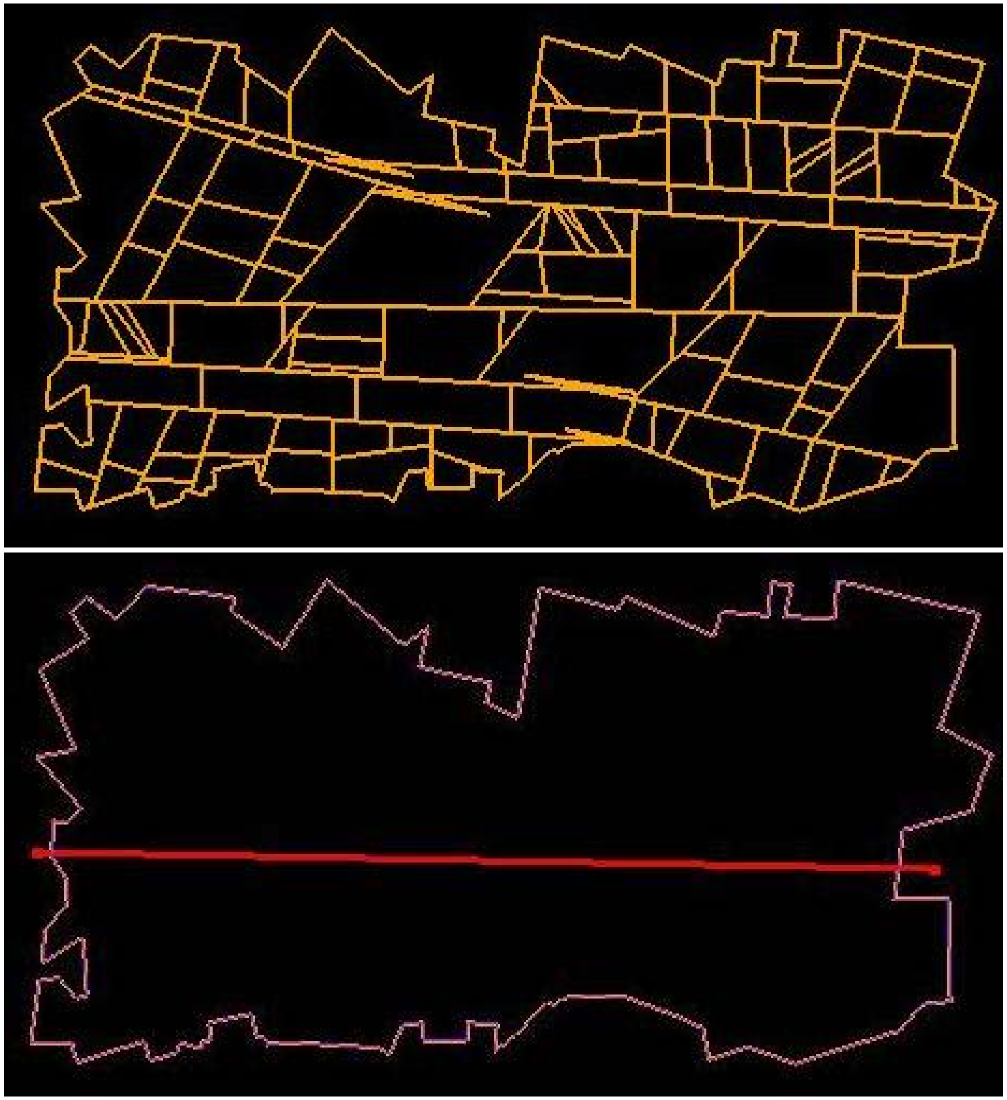


Urban Editing: Bigger City





Urban Editing: Bigger City



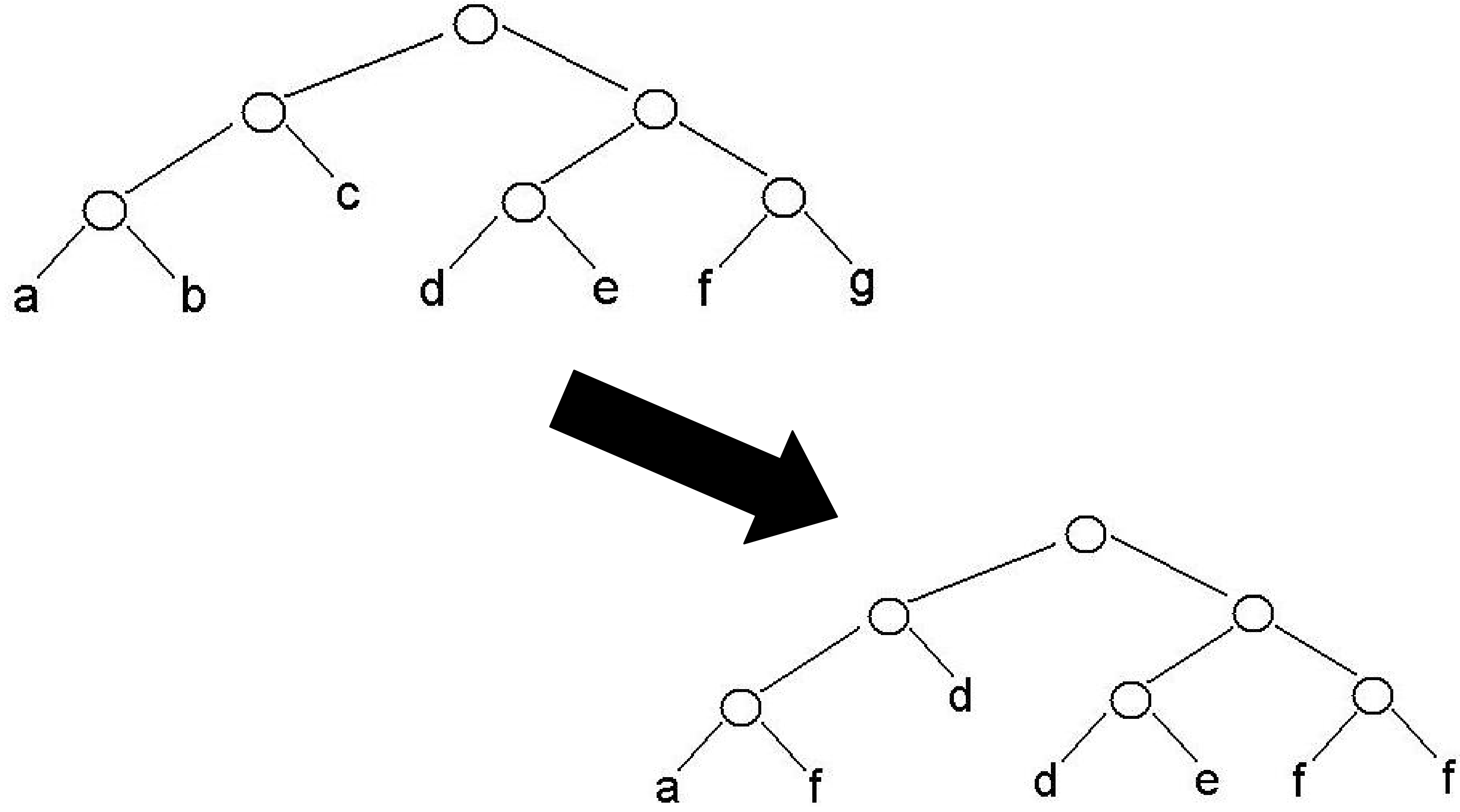
Urban Simplification

- Motivation
  - Want to have interactive rates.
    - For large cities we may have hundreds of thousands of terminals and hundreds of thousands of rules.
    - Finding a best matching rule for each iteration is time consuming.
    - 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.

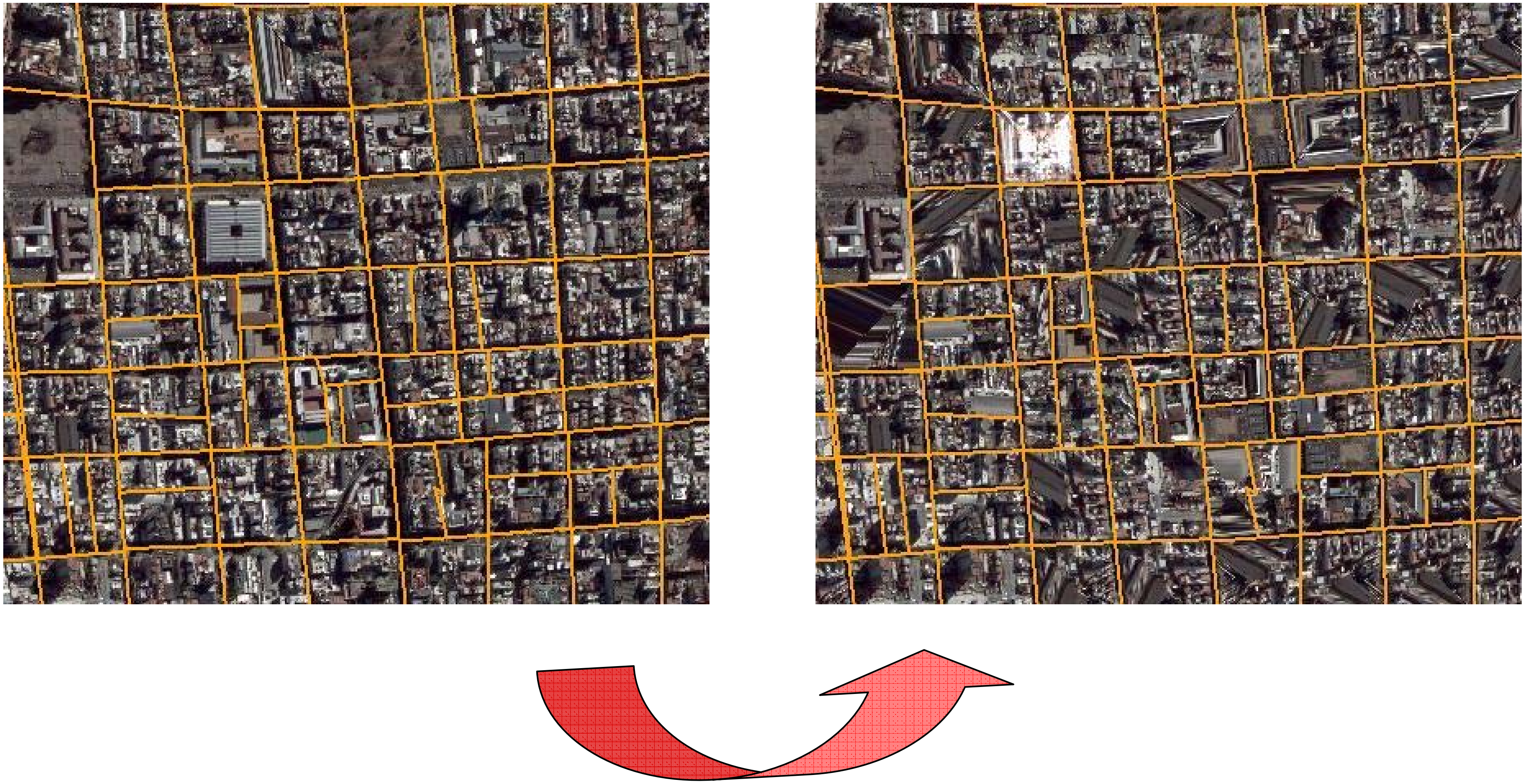
Terminal Simplification

- 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

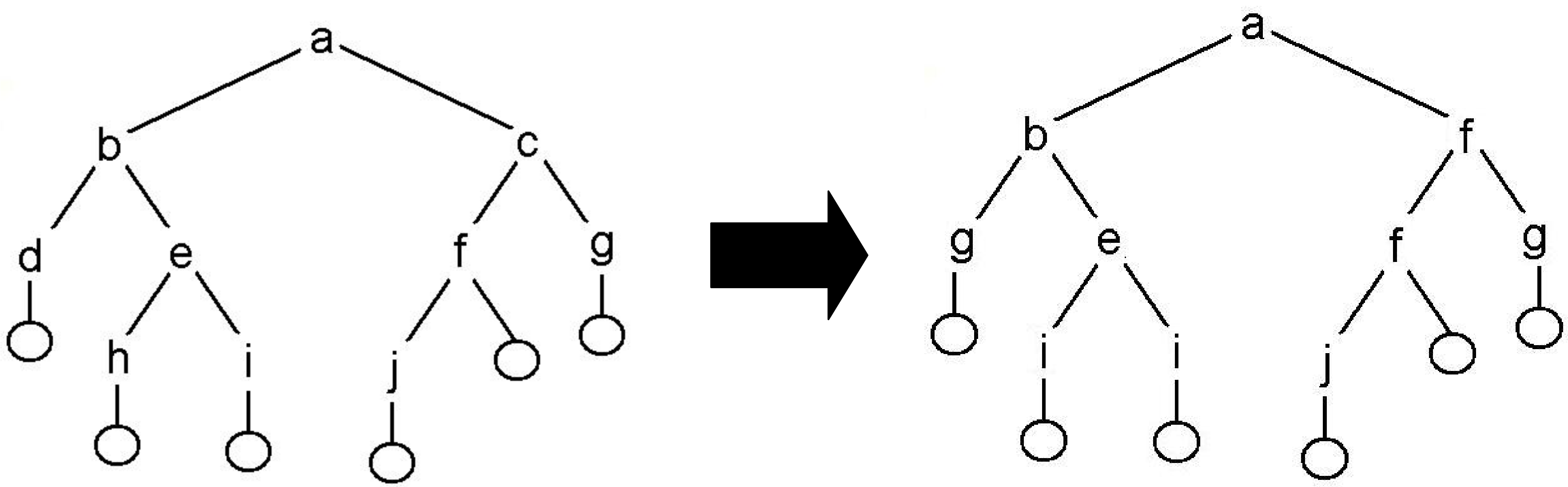


Terminal Simplification

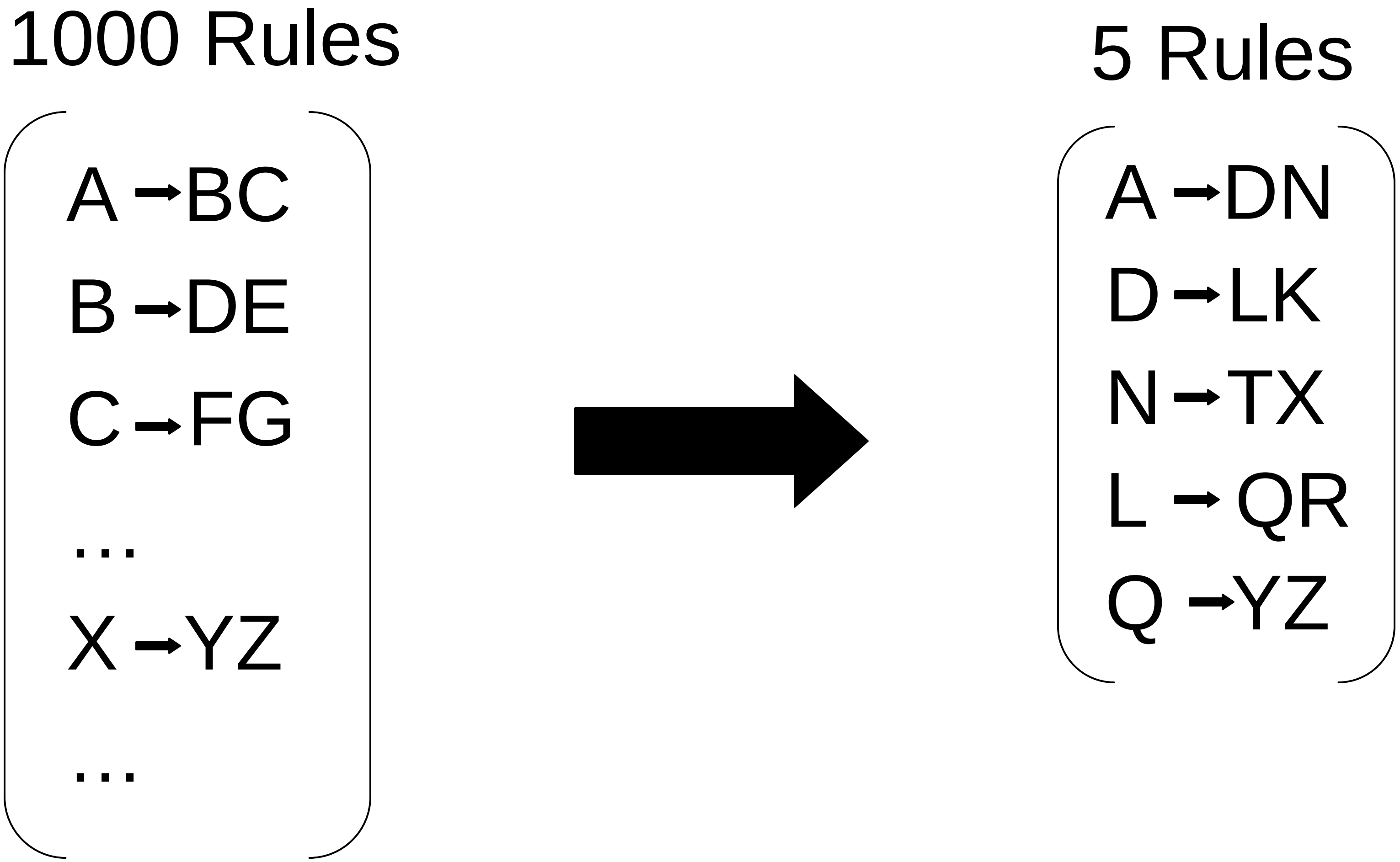


Procedural Simplification

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



Procedural Simplification



1000 Rules

A→BC

B→DE

C→FG

…

X→YZ

…

5 Rules

A→DN

D→LK

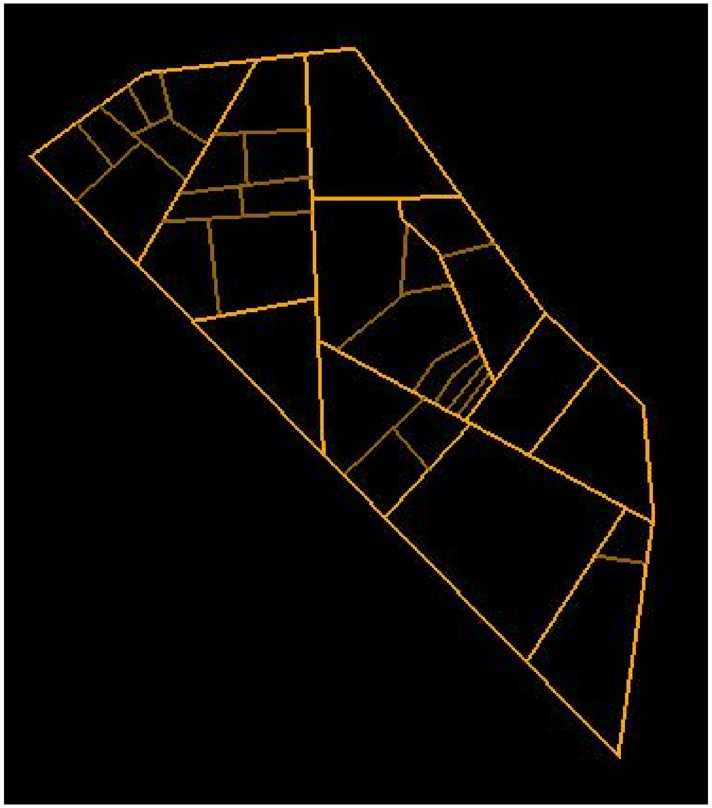
N→TX

L→QR

Q→YZ

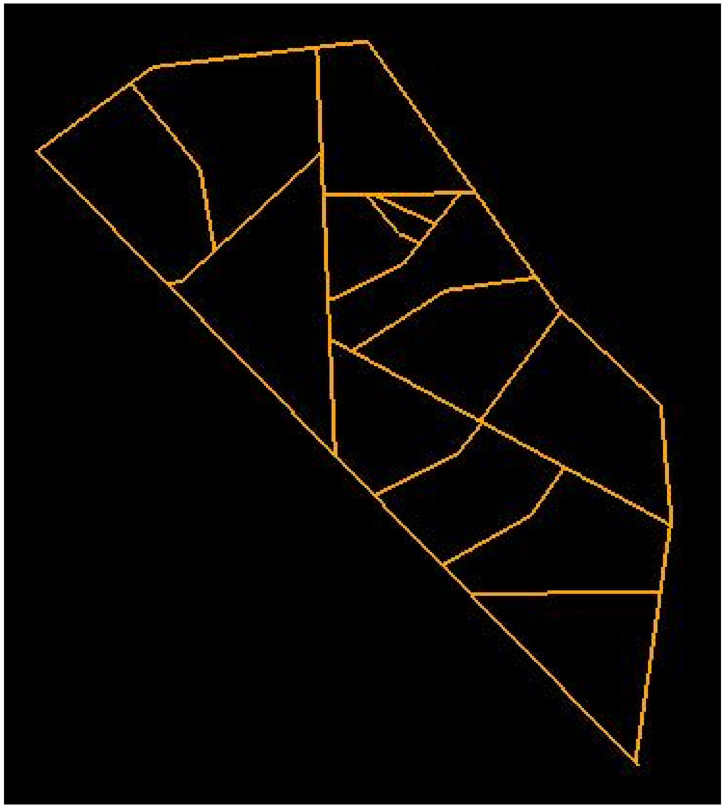
Procedural Simplification

30 Rules



Rome

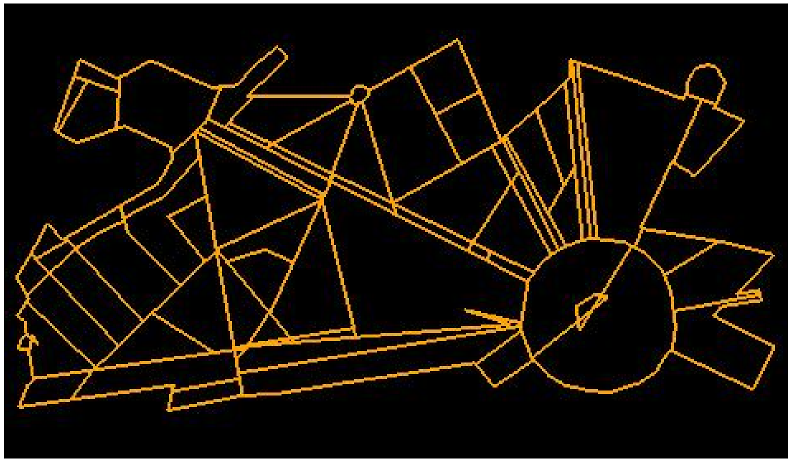
10 Rules



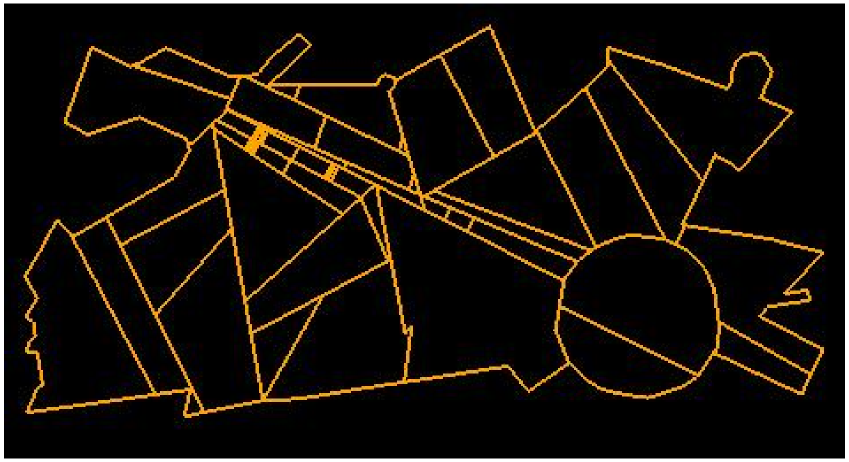
Procedural Simplification

Paris

77 Rules



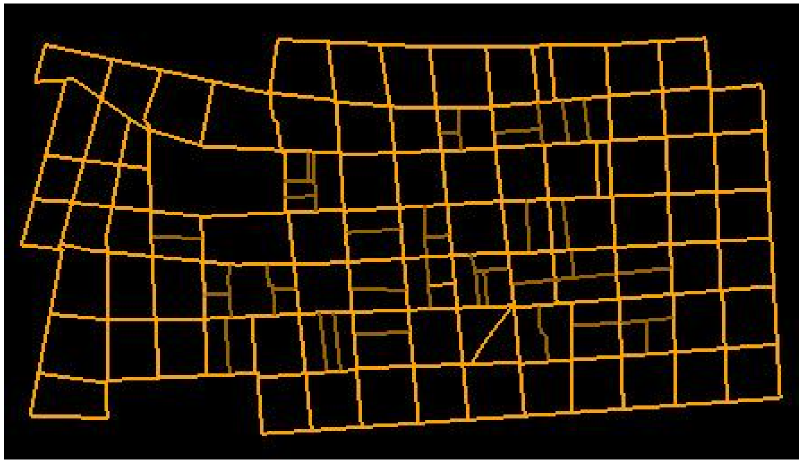
19 Rules



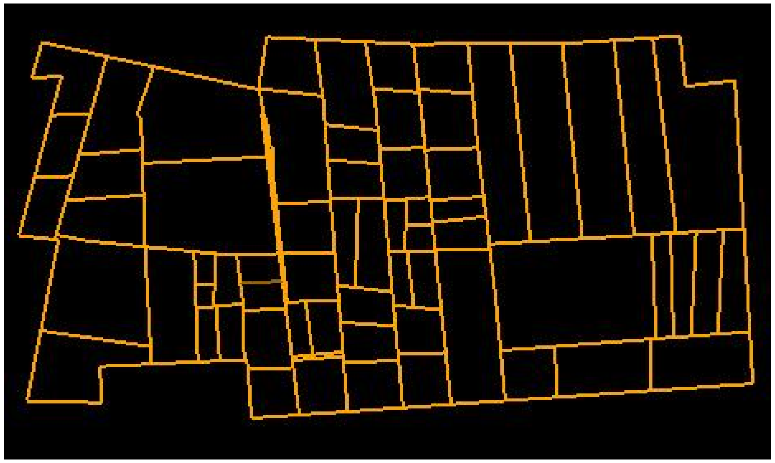
Procedural Simplification

Buenos Ares

130 Rules



38 Rules



Tools

- Similarity Estimation
- N-gon mapping

Similarity Estimation

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

N-gon mapping

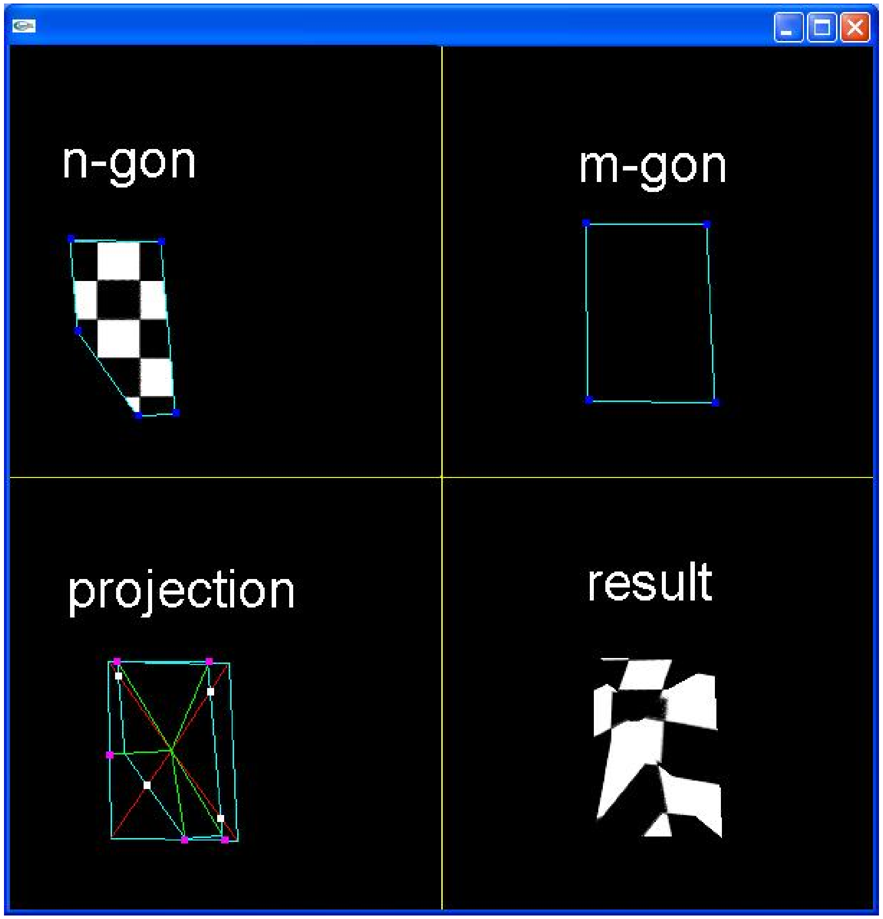
N-gon to M-gon mapping

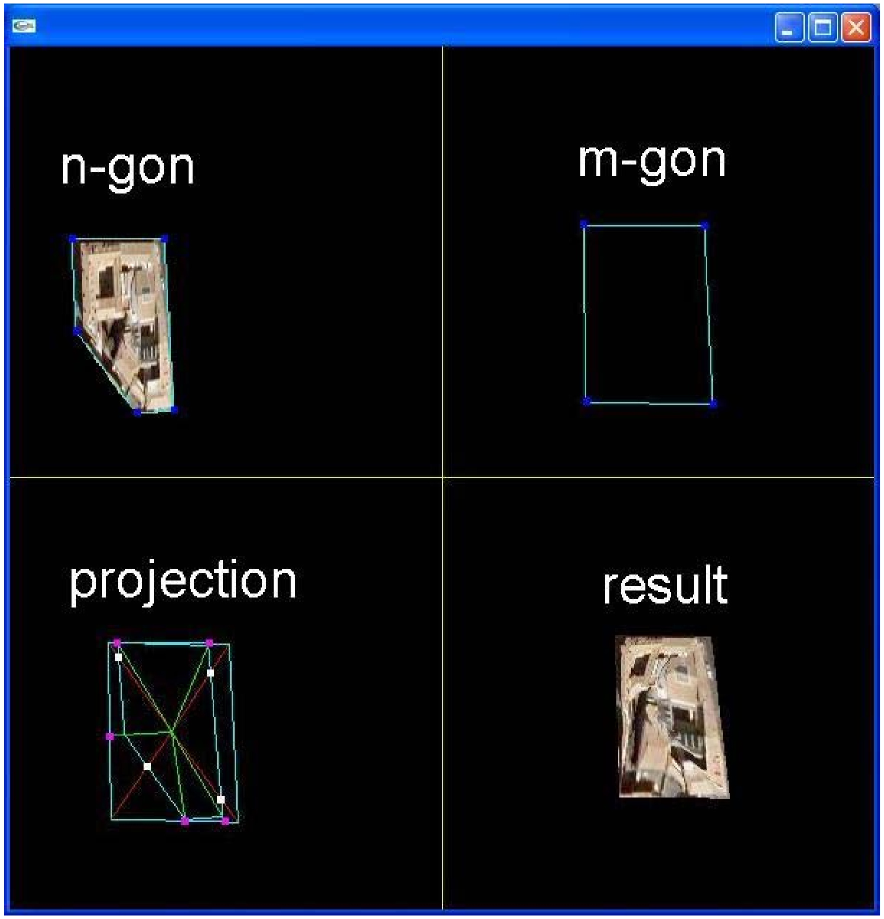
- □New tuples are derived that do not match the original tuples geometry.
- □Can you map a hexagon to a square? Should this be allowed?

Attempts at N-gon To M-gon Mapping

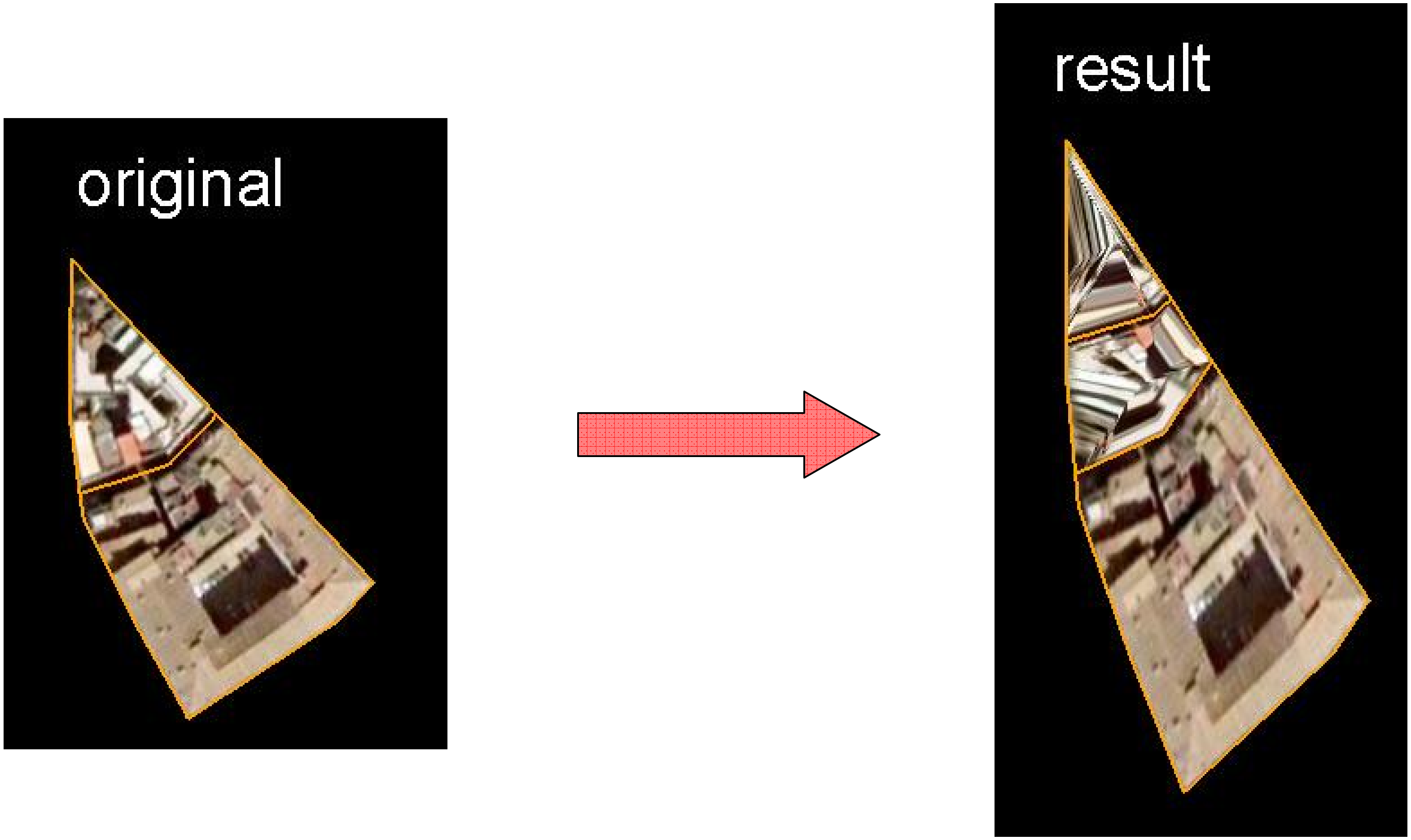
- Let N > M
- Attempt 1:
  - Project the vertices of each polygon onto each other. Now both polygons have the same number of points.
  - Determine the texture coordinates of the points on the perimeter of the n-gon. Use the coordinates for each corresponding point in the m-gon.

Attempt 1 Example





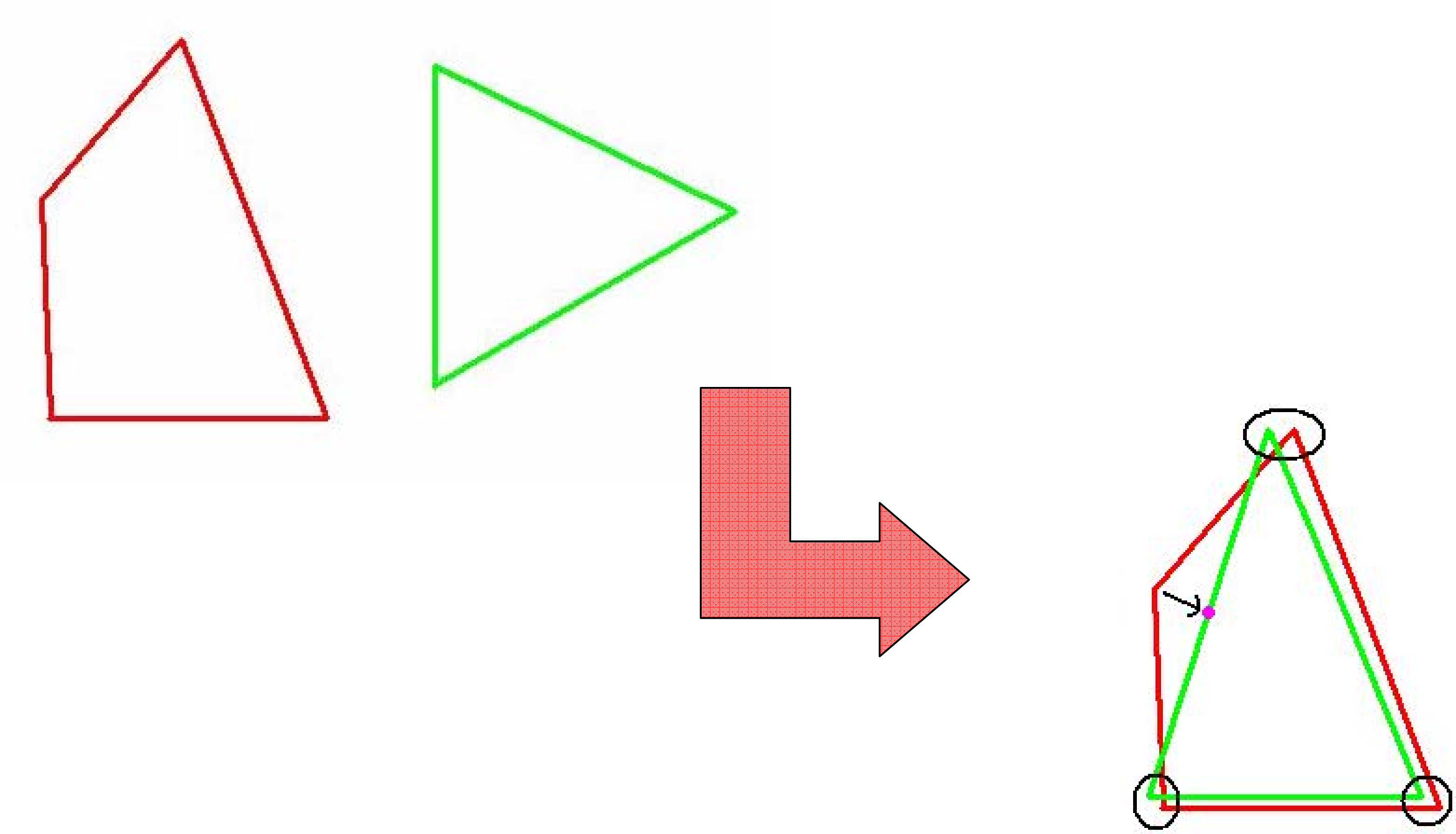
Attempt 1 in the program



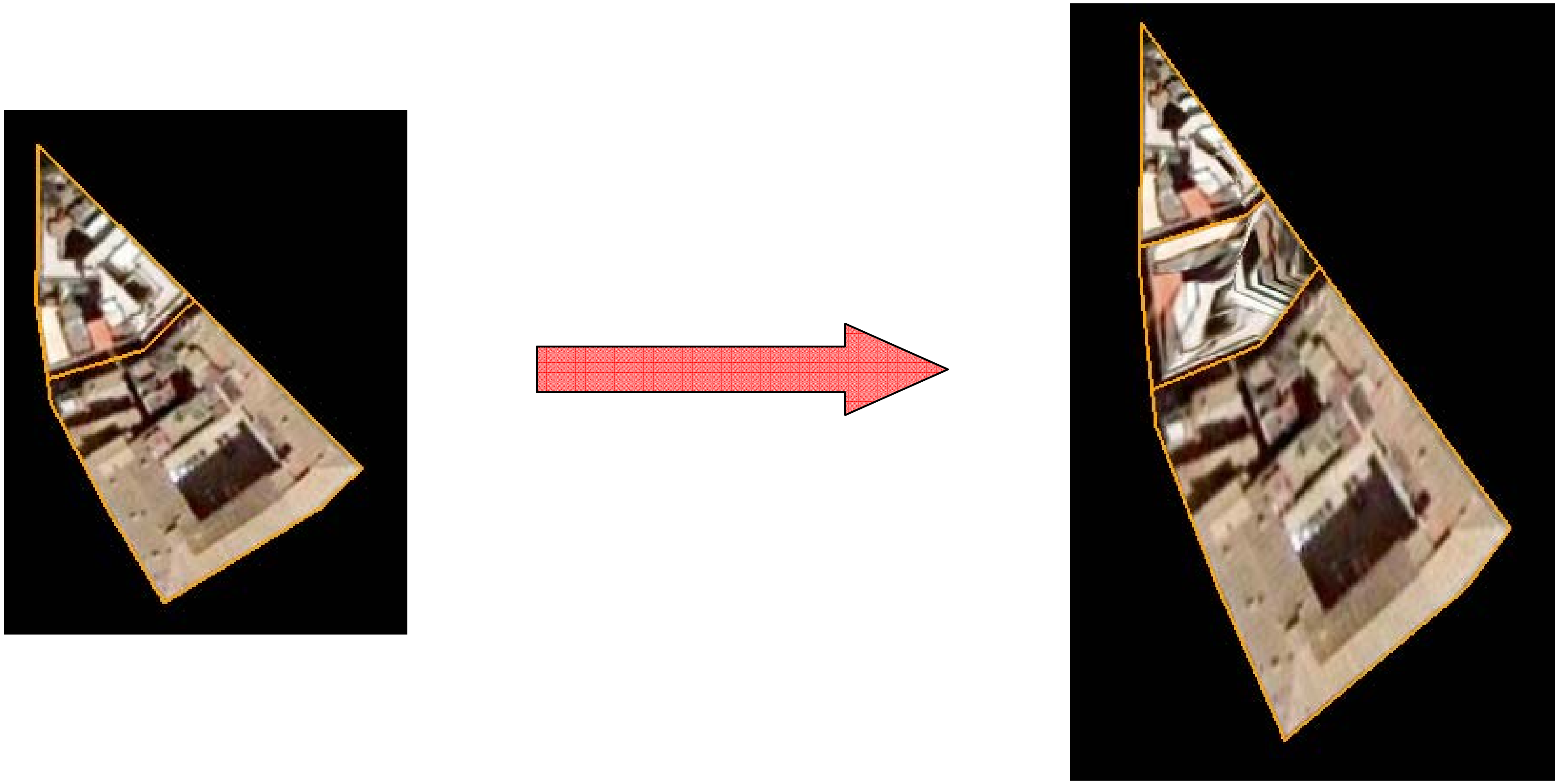
Attempts at N-gon To M-gon Mapping

- Let N > M
- Attempt 2:
  - 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.
  - For the intermediate points use the projected points of the n-gon onto the m-gon like before.
  - Rotate and scale to find a best fit.

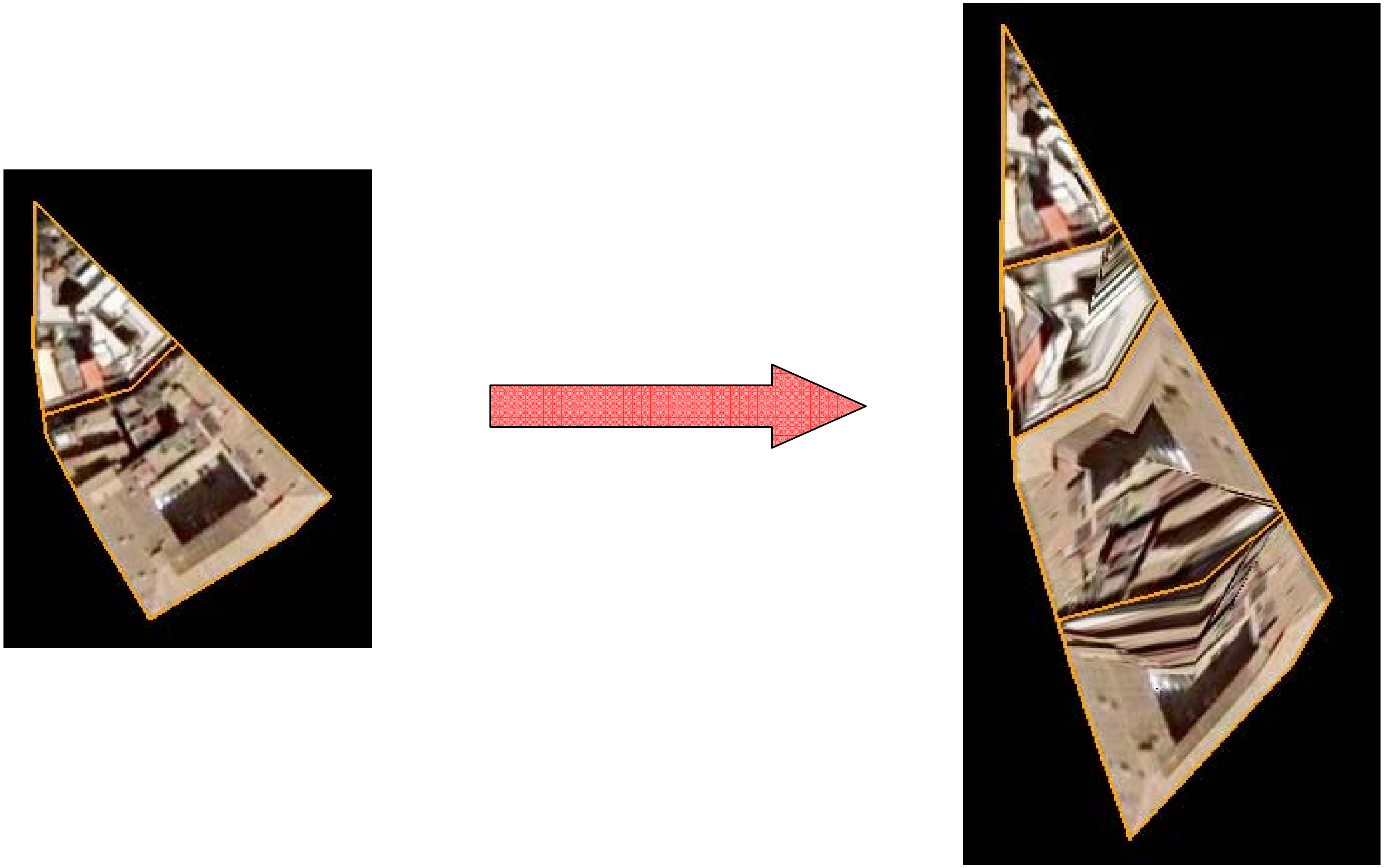
Attempt 2: Simple example



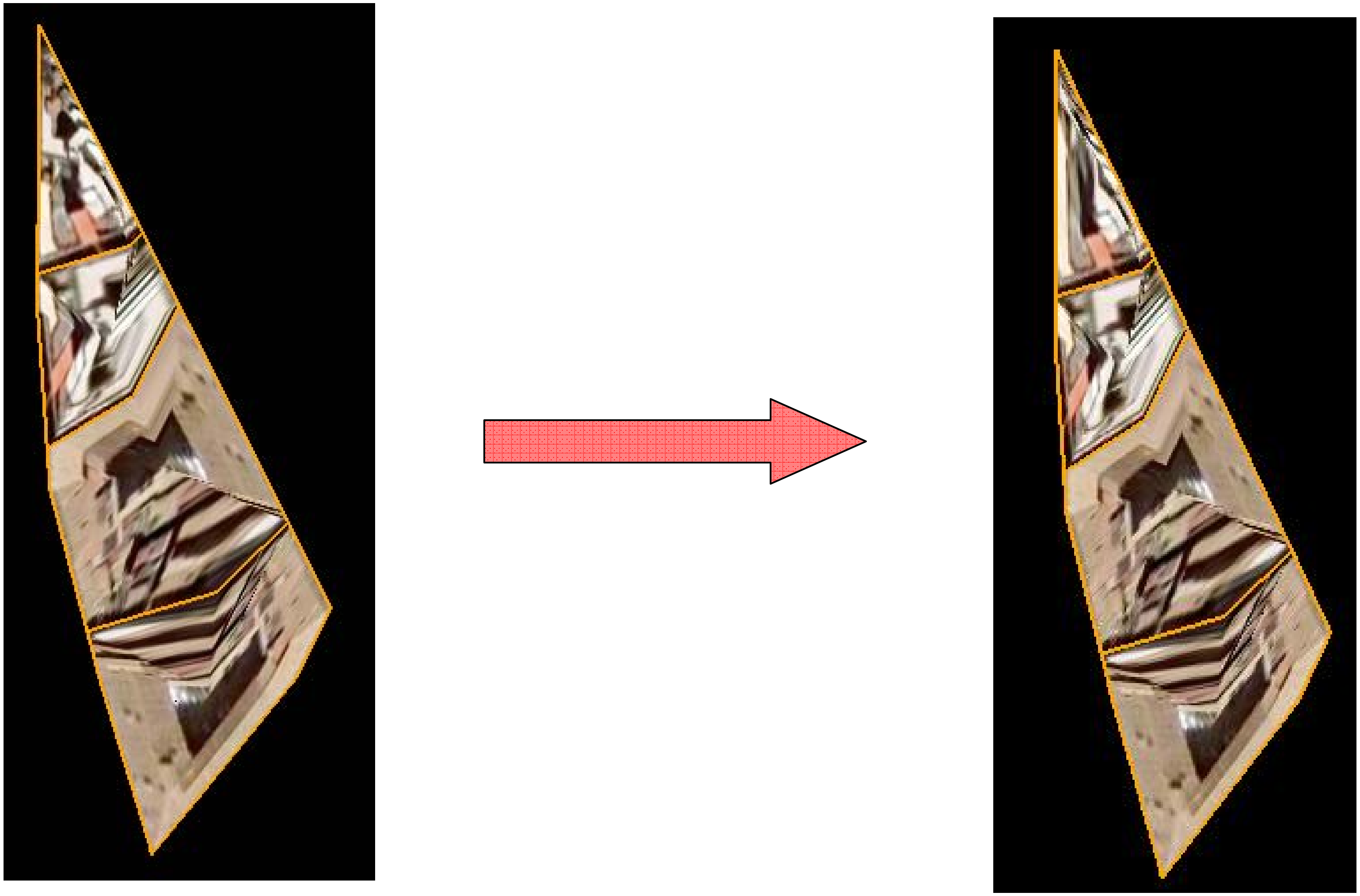
Attempt 2: In the program



Attempt 2: Stretch more



Attempt 2: More Stretching



Attempt 2 cont.

- For more complex scenes, the mapping looks worse.
- Vertices are not always mapped in a way that seems the most natural.
- Can still get seams in the texture.

- Maybe do not map based on point distance. Maybe base the mapping on vertex angles?

Future Work

- Improve N-gon to M-gon mapping
- Improve similarity metrics

- …Suggestions?

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?