Towards Semantic City Models

Stuttgart, sept 2013

Luc Van Gool, Andelo Martinovic, Markus Mathias

ETH Zurich, Computer Vision Lab

Un. Of Leuven, VISICS

The VarCity ERC Advanced Grant

erc
European Research Council
Established by
the European Commission

- VarCity = Variation & the City
- 5-yr project
- Goal: modeling existing cities in 4D, realistically, incl. the static and dynamic parts
- Emphasis is on adding and exploiting semantics

Envisaged pipeline

1. Camera-only mobile mapping -> 3D pt cloud/surfaces
   (+ additional internet-mined images for landmarks)
   this we have developed in the past, is operational

2. Recognition of urban object classes, e.g. people,
   cars, buildings, vegetation, traffic signs, …
   partially solved, we work on better scalability

3. Inverse procedural modeling
   our lab is experienced with forward procedural modeling
   issue: needs a style grammar, we focus on its automated learning

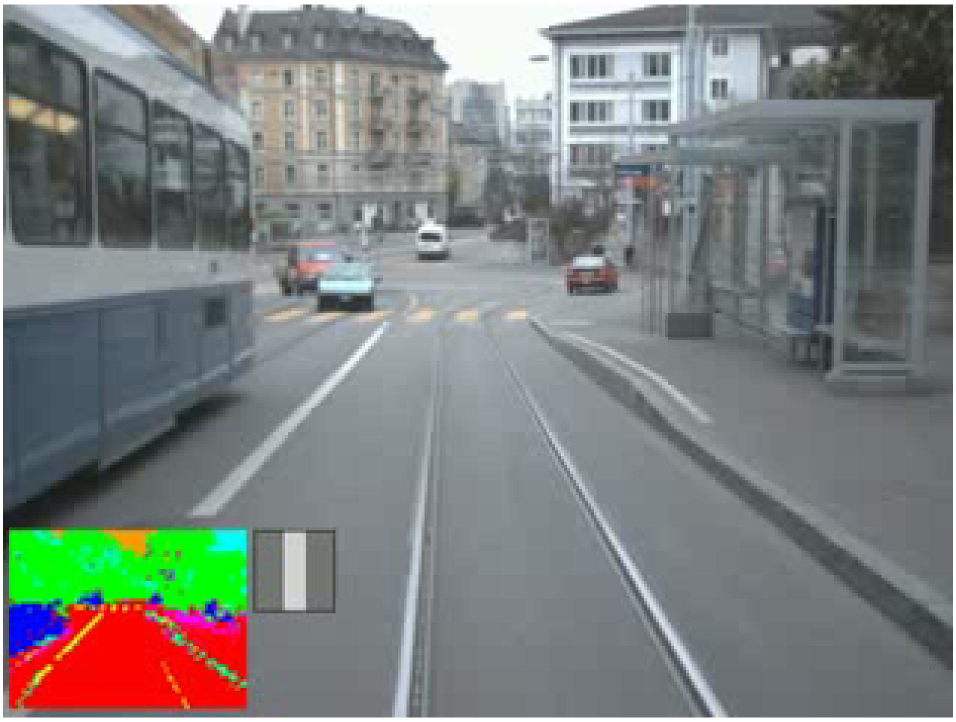
4. Adding dynamics: analyze traffic flows from intermittent data
   again recognition of cars, pedestrians, etc. is a key element

MOBILE MAPPING

Mobile mapping

RECOGNITION

semantic street scene analysis

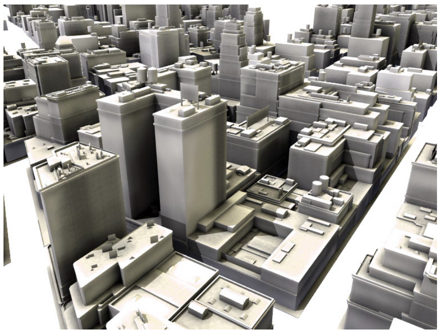


3D & recognition: mutual benefits

- A recent feat in computer vision: object class recognition, very useful *per se*…
- 3D information can strengthen such recognition.
- At least as importantly: recognition can support 3D acquisition.
- Here: 3D pt clouds, style grammars, & object detectors -> inverse procedural modeling

INVERSE PROCEDURAL MODELING

procedural modeling



Procedural modeling using rules

Id : predecessor : cond ⟶ successor : prob

- *id*: rule identifier
- *predecessor*: shape to be replaced (refined)
- *cond*: condition for the replacement to happen
- *successor*: new (refined) shape
- *prob*: probability of this rule to be applied

procedural modeling

```
building ⇢ SplitY{ columns | entablature | I(roof) }
sanctuary

columns ⇢ RepeatX{ column }
column ⇢ SplitY{ base | shaft | capital }
base ⇢ I(corinthian_base)
shaft ⇢ S I(corinthian_shaft)
capital ⇢ I(corinthian_capital)

entablature ⇢ SplitY{ architrave | frieze | cornice }
architrave ⇢ RepeatX{ I(architrave_tile) }
frieze ⇢ I(frieze)
cornice ⇢ RepeatX{ I(cornice_tile) }

sanctuary : orient == front ⇢ SplitXY{ wall | wall | wall ¦
                                       wall | I(door) | wall }
sanctuary ⇢ wall
wall ⇢ I(quad)
```

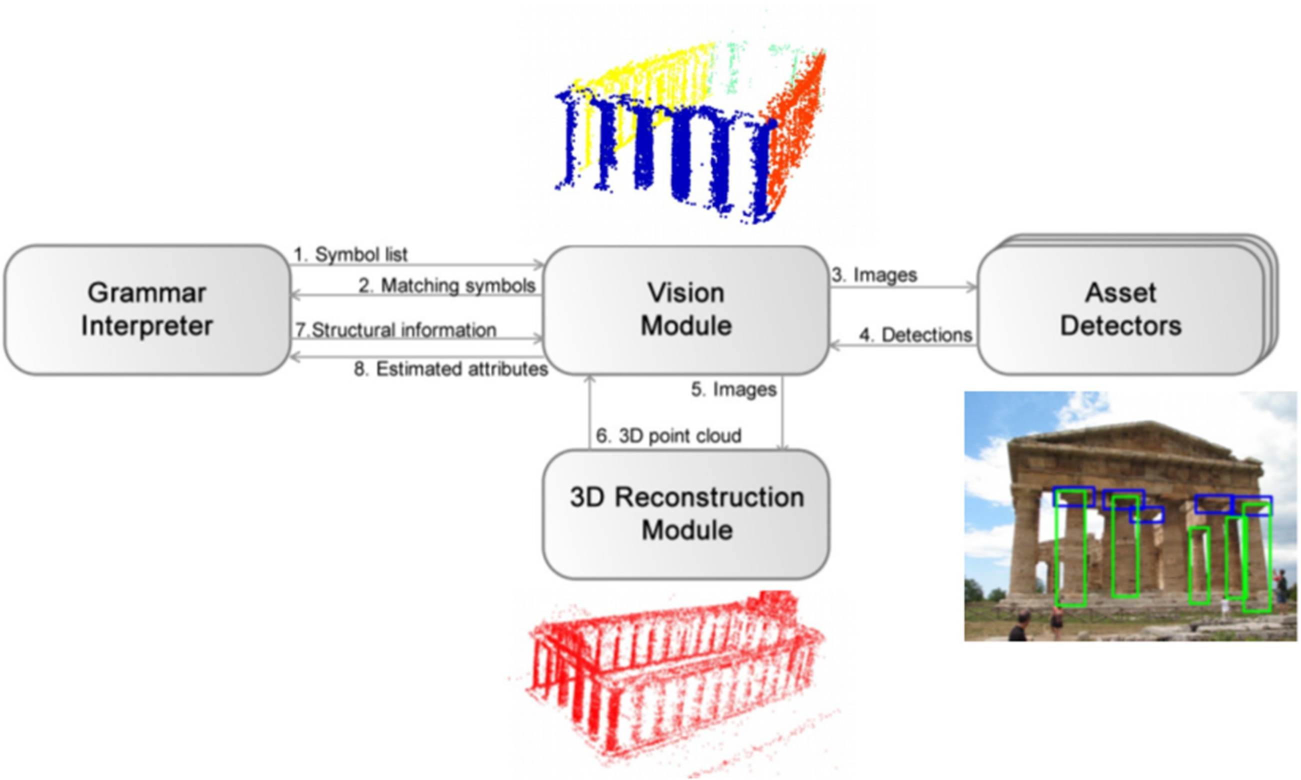
Procedural modeling

Inverse procedural modeling

Our automated pipeline contains:

- Structure-from-motion from images -> 3D pt cloud basically *bottom-up*
- Style-grammar interpreter basically *top-down*
- Building component detectors: this *mid-level* input avoids fragile bottom-up segmentation errors and is a catalyzer between bottom-up and top-down processes

Inverse procedural modeling



Inverse procedural modeling

Inverse procedural modeling

- Style grammars are non-trivial to produce
- and the resulting optimization spans a large space
- … so, is it wise to consider them in the first place?

Inverse procedural modeling

- Procedural models are compact
- and are semantic in nature
- and can produce more realistic visualizations
- and cut down on the pt cloud completeness needed
- and provide robustness to start from generic detectors and learn more dedicated ones on-the-go

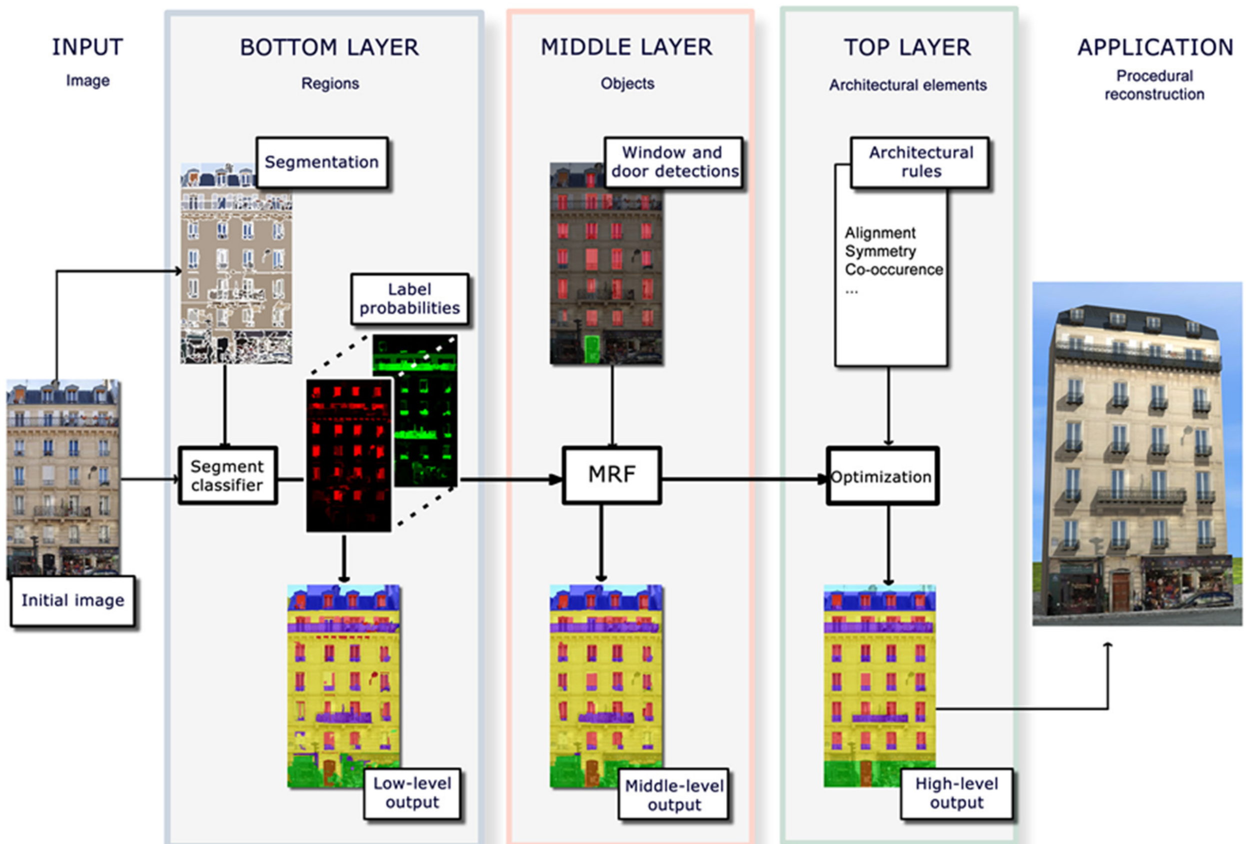
Inverse procedural modeling

- Note that we knew the style – hence appropriate grammar – in advance
- Not scalable if a human needs to tell the style of every single building, as with 3D city modeling
- Therefore we developed style classifiers (a current limitation is their focus on planar, i.e. façade oriented features; but expandable to 3D)

Inverse procedural modeling

- Also when styles are automatically recognized, creating many style grammars remains non-trivial
- And buildings may be of yet another style or a mixture of styles…
- Hence, we need to automatically learn style rules … or to at least start from more generic ones

inverse procedural modeling



inverse procedural modeling

Left: textured plane;
Right: semantic-guided visualisation, looks better and is more compact

Inverse procedural modeling

- User-generated grammar rules may only indirectly describe regularities of the *observed* structures.
- Our generic rules directly target observed structures though
- … and they are used as guidelines and not strict rules
- Our ongoing work extracts style-specific rules from such façade segmentations automatically, which can take the form of lists of rules & probabilities, which need not be human-readable