Towards Image Parsing

Stuart Geman
Elie Bienenstock
Eran Borenstein
Ya Jin
Wei Zhang
I. Introduction to a Composition Machine
II. Formulation
III. Prototype
Architectural Principles: Hierarchy and Reusability

Increasing
- Invariance
- Selectivity

- e.g. discontinuities, gradient
- e.g. linelets, curvelets, T-junctions
- e.g. contours, intermediate objects
- e.g. animals, trees, rocks
Instantiation and Reusability

animal head instantiated by tiger head

animal head instantiated by bear head
Parsing (scene interpretation)
Parsing (scene interpretation)

interpretation

selected subgraph
Mathematical Characterization

Generative, Probabilistic, Bayesian

Non-Markovian (context/content sensitive)

Generates pixels (not features)

Photometric-invariant data model
I. Introduction to a Composition Machine
II. Formulation
III. Prototype
Architecture

Every brick is
• off, or
• on, and selects a
set of children
Image interpretation

Bricks

Image interpretation: a complete subgraph $I$
Notation

Bricks $B$

$x^\alpha \in \{0,1,2\}$

$x^\alpha = 1$

$x^\alpha = 2$  $x^\beta \in \{0,1,2,3\}$

Image

$y_1, y_2, \ldots, y_n$
Markov backbone

Bricks $B$

$\{3,2,1,0\} \ni x^\beta \in \{0,1,2,3\}$

$\varepsilon_0^\beta + \varepsilon_1^\beta + \varepsilon_2^\beta + \varepsilon_3^\beta = 1$

Image $\{y_1, y_2, ..., y_n\}$
Markov backbone

\[ P(I) = \frac{\prod_{\beta \in B} \varepsilon_{x^\beta}}{\prod_{\beta \in B(I)} (1 - \varepsilon_0^\beta)} \]

Is this sufficiently constrained?

\[ x^\beta \in \{0,1,2,3\} \]

\[ \varepsilon_0^\beta + \varepsilon_1^\beta + \varepsilon_2^\beta + \varepsilon_3^\beta = 1 \]
Beyond Markovian distributions

Compositions depend on instantiations…e.g. the positioning of parts
Compositional distribution – a perturbed Markov model

\[ P(I) = \frac{\prod_{\beta \in B} \mathcal{E}_x^\beta}{\prod_{\beta \in B(I)} (1 - \mathcal{E}_0^\beta)} \]

- Bricks \( B \)
- \( x^\beta \in \{0, 1, 2, 3\} \)
- \( \mathcal{E}_0^\beta + \mathcal{E}_1^\beta + \mathcal{E}_2^\beta + \mathcal{E}_3^\beta = 1 \)

Image \( y_1, y_2, \ldots, y_n \)
Compositional distribution – a perturbed Markov model

\[ P(I) = \frac{\prod_{\beta \in B} \varepsilon_0^\beta}{\prod_{\beta \in B(I)} (1 - \varepsilon_0^\beta)} \frac{\prod_{\beta \in A(I)} p_\beta^c (d_\beta^b (I))}{\prod_{\beta \in A(I)} p_\beta^o (d_\beta^b (I))} \]

\[ x^\beta \in \{0, 1, 2, 3\} \]

\[ \varepsilon_0^\beta + \varepsilon_1^\beta + \varepsilon_2^\beta + \varepsilon_3^\beta = 1 \]
$C = \text{Corr}(T_A, y_A)$

$p_A(C)$

$sufficiency assumption$

$p(y_A) = \frac{p_A(C)}{\#\{y_A' : \text{Corr}(T_A, y_A') = C\}}$

$P(\tilde{y} | I)$
Bayesian inference

- Operate on: \( P(I \mid \vec{y}) \propto P(\vec{y} \mid I)P(I) \)

- Computation:
  - full parse:
    - bottom-up \(\rightarrow\) candidate objects (trees)
    - top-down \(\rightarrow\) resolve ambiguity & conflict \(\rightarrow\) full parse
  - search for an object:
    - depth-first \(\rightarrow\) candidate objects (trees)
    - top-down \(\rightarrow\) resolve ambiguity & conflict \(\rightarrow\) full parse
  - bottom-up and depth-first are CTF
I. Introduction to a Composition Machine

II. Formulation

III. Prototype
Architecture

- license plates
- license numbers (3 digits + 3 letters, 4 digits + 2 letters)
- plate boundaries, strings (2 letters, 3 digits, 3 letters, 4 digits)
- generic letter, generic number, L-junctions of sides
- characters, plate sides
- parts of characters, parts of plate sides
Sampling the prior, data models

Random 4-digit strings:

Markov backbone

compositional distribution
Test set: 385 images, mostly from Logan Airport

Courtesy of Visics Corporation
Image interpretation

Original Image

Top object

Top 10 objects

Top 25 objects
Image interpretation

Test image

Top objects
Performance

• 385 images

• Six plates read with mistakes (>98%)

• Approx. 99.5% characters read correctly

• Zero false positives
Performance: errors
Performance: errors
Efficient discrimination: Markov versus Compositional dist.
Efficient discrimination: testing objects against their parts

Test image

9 active “8” bricks under whole model

1 active “8” brick under parts model
Efficient discrimination: testing objects against their parts

Test image

53 active “A” bricks under whole model   20 active “A” bricks under parts model
Efficient computation: depth-first search
Efficient computation: depth-first search

Number of visits to each pixel. Left: linear scale  Right: log scale
Efficient computation: depth-first search

Number of visits to each pixel. Left: linear scale  Right: log scale