On the Completeness of Coding with Image Features
British Machine Vision Conference 2009

Wolfgang Förstner    Timo Dickscheid    Falko Schindler

Department of Photogrammetry
Institute of Geodesy and Geoinformation
University of Bonn

September 8th 2009
Local Feature Detectors

Feature detectors usually capture different image content.

Dimension 2
Laplacian Blobs
(Lowe 2004)

Dimension 1
Straight edges
(Förstner 1994)

Dimension 0
SFOP Junctions
(Förstner et al. 2009)

“Sextant” from Caltech 256 database
http://www.vision.caltech.edu/Image_Datasets/Caltech256/
Local Feature Detectors

Differences even within the same dimension.

- **Dimension 2**
  - **Laplacian Blobs**
    - (Lowe 2004)
  - **Harris affine blobs**
    - (Mikolajczyk/Schmid 2004)
  - **MSER blobs**
    - (Matas et al. 2004)

“Duck” from Caltech 256 database
http://www.vision.caltech.edu/Image_Datasets/Caltech256/
How to quantify the completeness of information preserved by features?

By feature detection, we attempt to

- gain robustness,
- decrease the data volume,
- **preserve important information.**

→ How to measure/quantify?

Interpret feature detection/description as coding scheme.

This requires
1. A representation for measuring relevant local image content
2. A representation for measuring local feature content
3. A distance measure between the two
How to quantify the completeness of information preserved by features?

By feature detection, we attempt to

▸ gain robustness,
▸ decrease the data volume,
▸ preserve important information.

→ How to measure/quantify?

Interpret feature detection/description as coding scheme.

This requires

1. A representation for measuring relevant local image content
2. A representation for measuring local feature content
3. A distance measure between the two
(1) Representation of “relevant local image content”

Derive an **entropy density** $p_H(x)$ from local image statistics.

- requires few bits in homogeneous areas
- requires many bits in busy areas

Image shown as shaded relief

Peak
Low
High

$\text{Image}$

$\text{Low}$

$\text{High}$

$x_1$

$x_2$

$p_H(x)$

shown as shaded relief
Derivation of entropy density $p_H(x)$

For all pixels $x \in I$

<table>
<thead>
<tr>
<th>For patch sizes $s = 3, 5, 9, 17, \ldots$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H(x, s) = \frac{1}{2} \sum_{uv \setminus {0}} \log 2\pi e^{\frac{\max(P_{x,s}(u,v) - \sigma_n^2,0)}{\sigma_n^2}}$</td>
</tr>
<tr>
<td>(#bits for coding the patch)</td>
</tr>
</tbody>
</table>

$H(x) = \sum_s H(x, s)$  
(#bits for coding the pixel over scales)

$p_H(x) = \frac{H(x)}{\sum_{y \in I} H(y)}$  
(Normalized entropy density)

- Entropy from the power spectrum $P_{x,s}(u, v)$
- Noise variance determines significance level
- Assumes image to be a sample of a Gaussian process
Derive a **Feature coding density** $p_c(x)$

- Represent the region covered by each feature with an anisotropic Gaussian
- Assume a certain number of bits per feature
- Spread the number of bits over each Gaussian area

The normalized sum of Gaussians gives $p_c(x)$.

---

**Image**

**Edge segments**

**SFOP junctions**
(3) Distance measure: Hellinger’s metric

\[ d(p_H(x), p_c(x)) = \sqrt{\frac{1}{2} \sum_x \left( \sqrt{p_H(x)} - \sqrt{p_c(x)} \right)^2} \]

Comparable to Kullback / Leibler divergence (which is not a metric!)

Sketch for an image with two pixels
Evaluation scheme for two sets of features

Feature set 1

Feature set 2
Evaluation scheme for two sets of features

- Feature set 1
  - Coding $p_{c1}(x)$
  - Reference $p_H(x)$

- Feature set 2
  - Coding $p_{c2}(x)$
Evaluation scheme for two sets of features

Feature set 1
- detect
- Coding $p_{c1}(x)$
- Reference $p_H(x)$
- Compute $d_1 = d(p_{c1}(x), p_H(x))$
- $d_2 < d_1$

Feature set 2
- detect
- Coding $p_{c2}(x)$
- Reference $p_H(x)$
- Compute $d_2 = d(p_{c2}(x), p_H(x))$
Results for separate detectors

“Fifteen Scene Categories” http://www-cvr.ai.uiuc.edu/ponce_grp/data/
“Brodatz textures” http://www.ux.uis.no/~tranden/brodatz.html

Average distance $d(p_H(x), p_C(x))$ for separate detectors per image category
Results for separate detectors

- Overall best results: MSER / SFOP junctions

Average distance $d(p_H(x), p_C(x))$ for separate detectors per image category
Results for separate detectors

- Overall best results: MSER / SFOP junctions
- Straight edges not useful for “Forest” and “Brodatz”

Average distance $d(p_H(x), p_c(x))$ for separate detectors per image category
Results for separate detectors

- Overall best results: MSER / SFOP junctions
- Straight edges not useful for “Forest” and “Brodatz”
- Harris affine rarely better than Harris Laplace

Average distance $d(p_H(x), p_c(x))$ for separate detectors per image category
How well do others complement the Lowe detector?

Average distance $d(p_H(x), p_C(x))$ for detector combinations
Results for multiple detectors

- Lowe most efficiently complemented by SFOP or MSER

Average distance $d(p_H(x), p_C(x))$ for detector combinations
Results for multiple detectors

- Lowe most efficiently complemented by SFOP or MSER
- Lowe/MSER/SFOP usually as good as taking all detectors
Results for multiple detectors

- Lowe most efficiently complemented by SFOP or MSER
- Lowe/MSER/SFOP usually as good as taking all detectors
- Harris affine complements Lowe no better than Hessian affine

Average distance $d(p_H(x), p_C(x))$ for detector combinations
Conclusion & Outlook

- By interpreting feature detection as image coding, "completeness" becomes quantifiable.
- Proposed approach helps on:
  - distinguishing detectors
  - choosing detector combinations
- What about "mapping the space of detectors"?
Thanks for your attention

Feature set 1

Feature set 2

Coding $p_c^1(x)$

Coding $p_c^2(x)$

Reference $p_H(x)$

Compute $d_1$

$d(p_c^1(x), p_H(x))$

$d_2 < d_1$

Compute $d_2$

$d(p_c^2(x), p_H(x))$
Thanks for your attention

Feature set 1

Feature set 2

Coding $p_{c_1}(x)$

Coding $p_{c_2}(x)$

Reference $p_H(x)$

Compute $d_1$

$d_1 = d(p_{c_1}(x), p_H(x))$

Compute $d_2$

$d_2 = d(p_{c_2}(x), p_H(x))$

$d_2 < d_1$