Learning bottom-up visual processes using automatically generated ground truth data

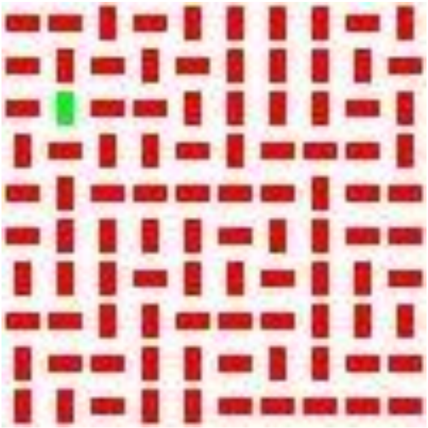
Kalle Åström, Yubin Kuang, Magnus Oskarsson, Lars Kopp an Martin Byröd

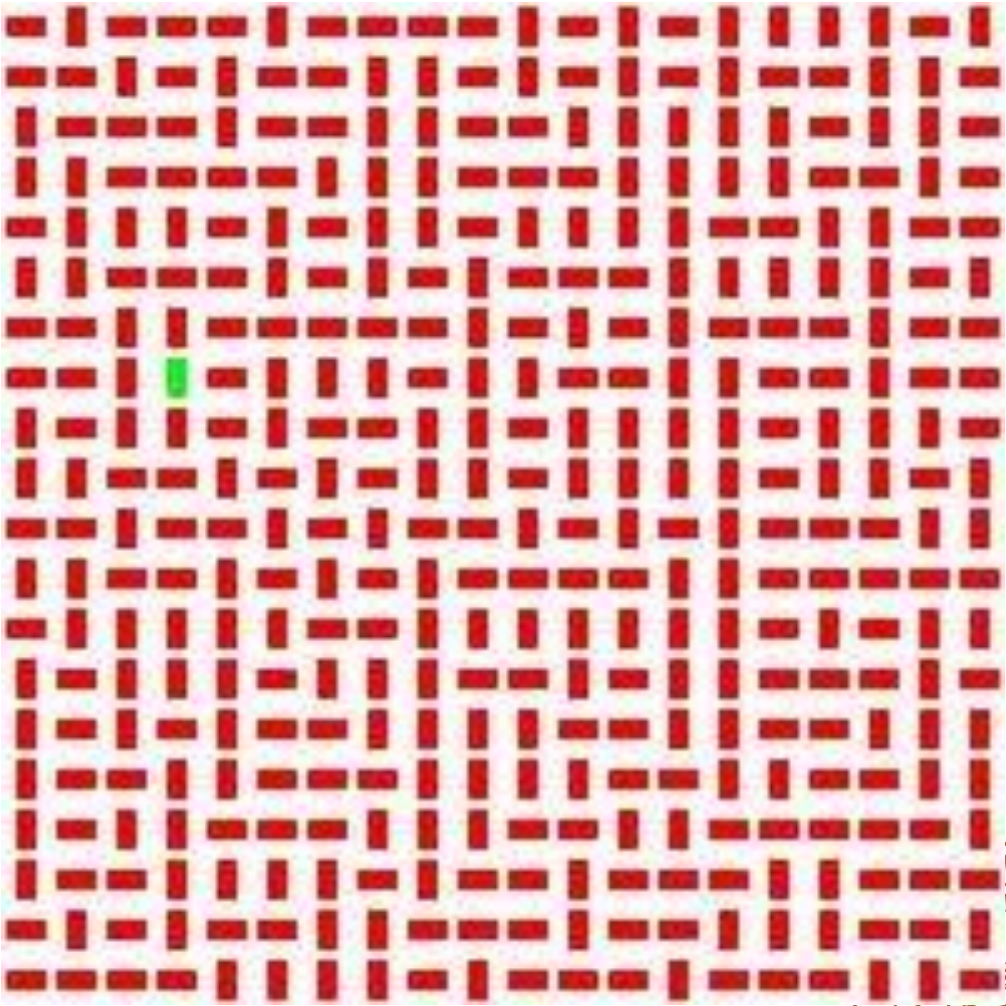
Mathematical Imaging Group

Center for Mathematical Sciences

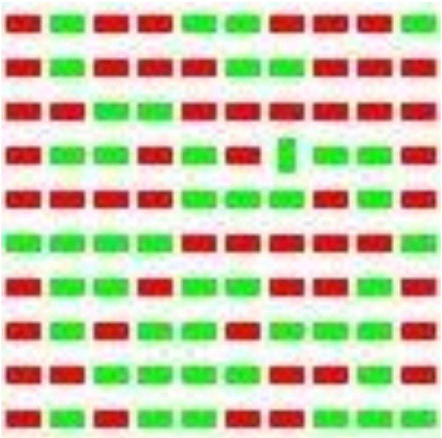
Lund University

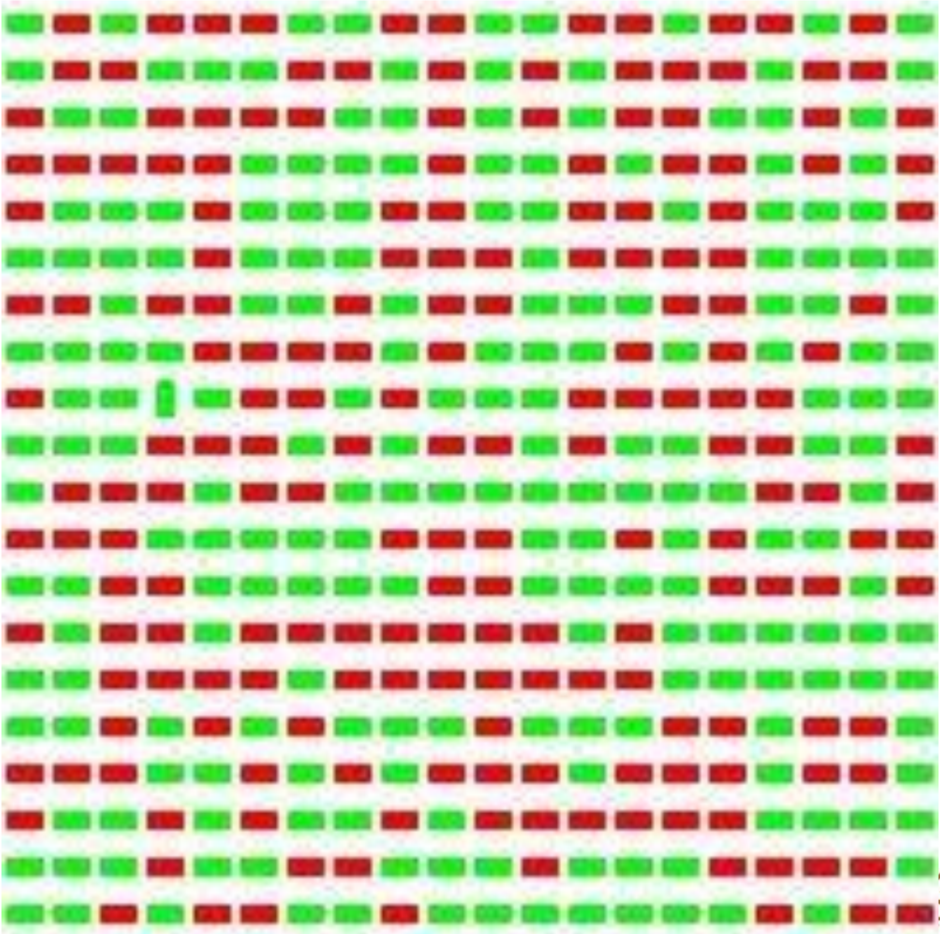
Fast (bottom-up) - some methods scale



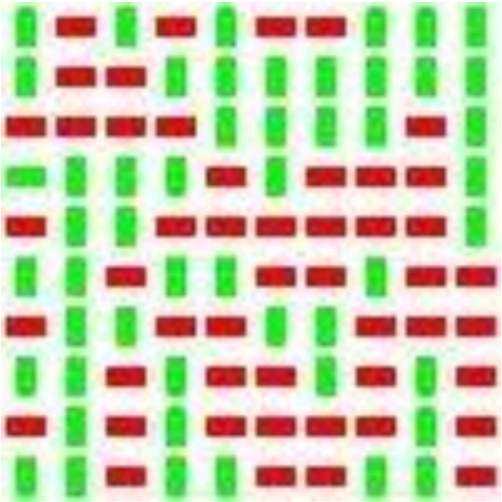


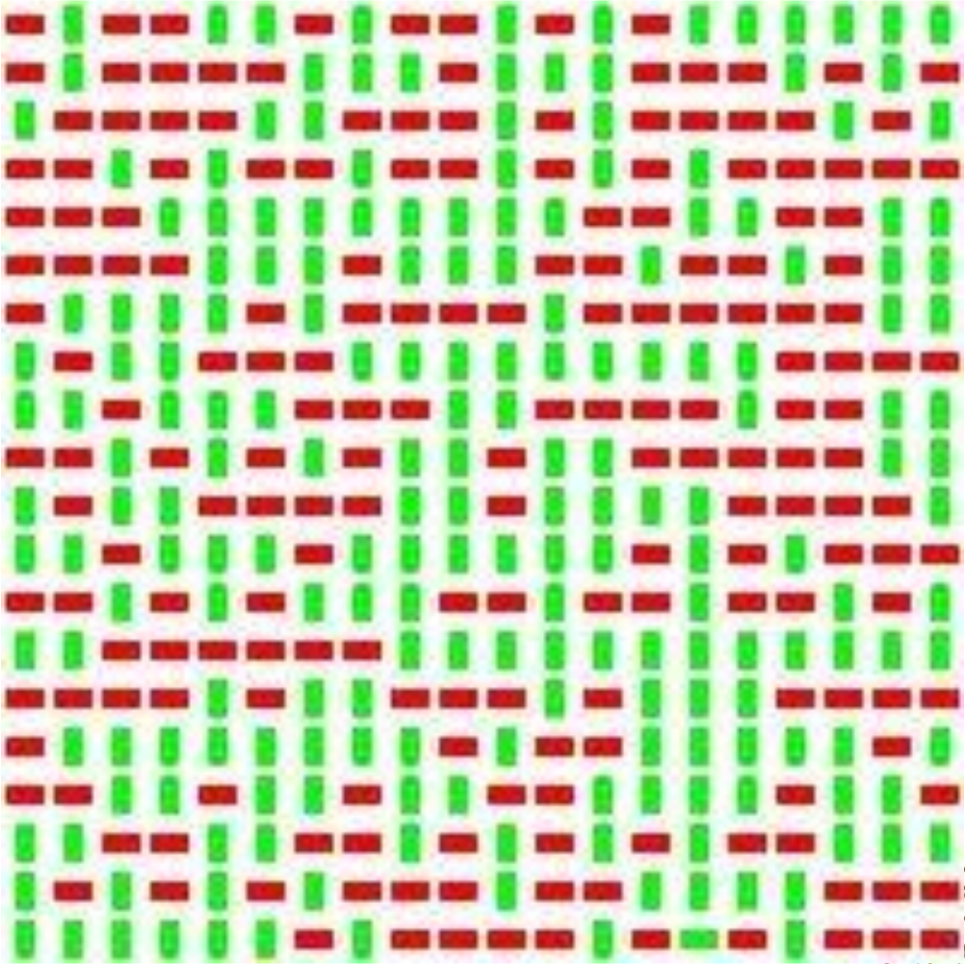
Fast (bottom-up) - some methods scale





Fast (bottom-up) - some don't





Task: Image Retrieval

- Oxford Building Data (Philbin et al. CVPR'07)

Query

Task: Image Retrieval

- Oxford Building Data (Philbin et al. CVPR'07)

Query

Task: Image Retrieval

- Oxford Building Data (Philbin et al. CVPR'07)

Query

Good match

Task: Image Retrieval

- Oxford Building Data (Philbin et al. CVPR'07)

Query

Good match

Matched?

Task: Image Retrieval

- Oxford Building Data (Philbin et al. CVPR'07)

Query

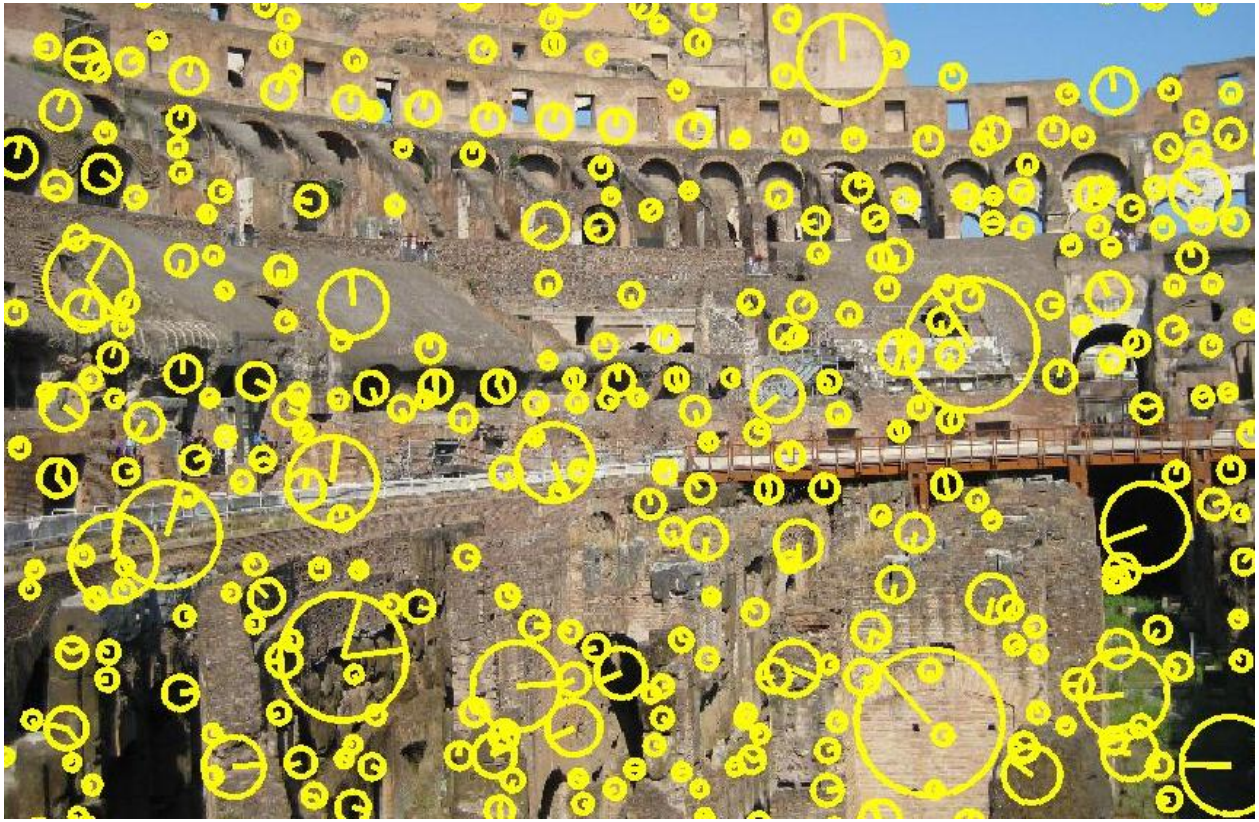
Good match

Matched?

Baseline System – Bags of Words

Baseline System – Bags of Words

- **Interest point detection (position, scale, orientation)**
  - Differences of Gaussian/Harris



Baseline System – Bags of Words

- **Interest point detection**
  - Differences of Gaussian/Harris
- **Feature extraction** (feature vector e g R^128)
  - SIFT/SURF/DAISY

Baseline System – Bags of Words

- **Interest point detection**

  - Differences of Gaussian/Harris

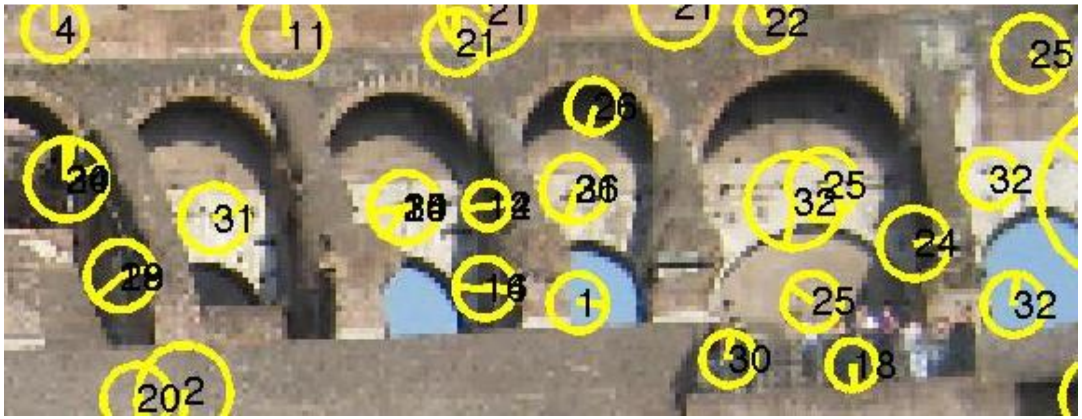
- **Feature extraction**

  - SIFT/SURF/DAISY

- **Generating vocabularies – quantization**

  - hierarchical k-means (Nister, Stewenius CVPR'06)

  - approximate k-means (Philbin et al. CVPR'08)



Baseline System – Bags of Words

- **Interest point detection**

  - Differences of Gaussian/Harris

- **Feature extraction**

  - SIFT/SURF/DAISY

- **Generating vocabularies – quantization**

  - hierarchical k-means (Nister, Stewenius CVPR'06)

  - approximate k-means (Philbin et al. CVPR'08)

- **Bags of words**

  - measure image similarities based on the histogram of words with L_1 or L_2 norm

Baseline System – Bags of Words

- **Interest point detection**

  - Differences of Gaussian/Harris
- **Feature extraction**

  - SIFT/SURF/DAISY
- **Generating vocabularies – quantization**

  Our Focus

  - hierarchical k-means (Nister, Stewenius CVPR'06)

  - approximate k-means (Philbin et al. CVPR'08)
- **Bags of words**

  - measure image similarities based on the histogram of words with L_1 or L_2 norm

Quantization – State of the Art

- ~ 1M words

Quantization – State of the Art

- ~ 1M words
- Hierarchical K-means, Approximate K-means, Approximate K-means + soft-assignment

Quantization – State of the Art

- ~ 1M words
- Hierarchical K-means, Approximate K-means, Approximate K-means + soft-assignment
- **Advantage:**

  - efficient training

  - fast matching or retrieval using inverted files

Quantization – State of the Art

- ~ 1M words
- Hierarchical K-means, Approximate K-means, Approximate K-means + soft-assignment
- **Advantage:**

  - efficient training

  - fast matching or retrieval using inverted files
- **Disadvantage?**

  - Unsupervised: Features quantized to the same word do not usually correspond

Evaluation

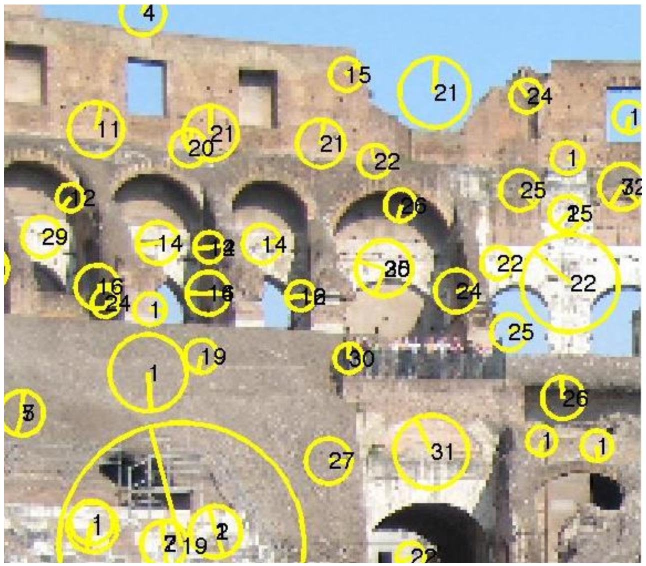
- Image query vs database (5000 images, 100 000 images)
  – mean average precision

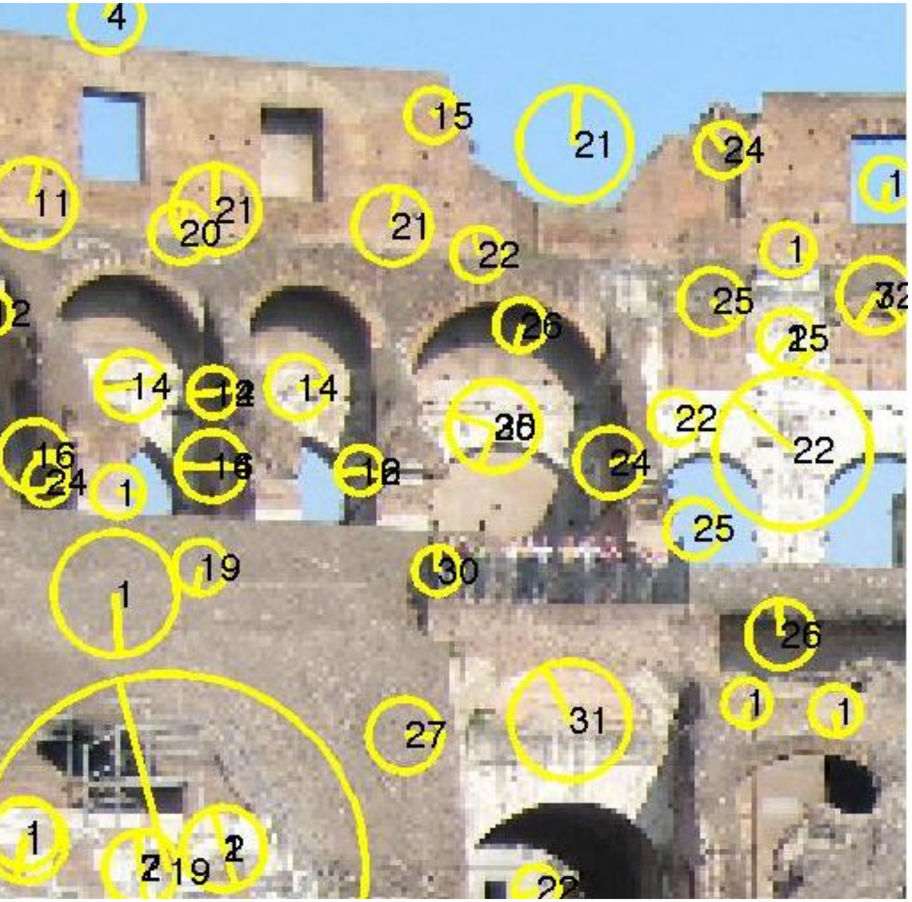
Evaluation

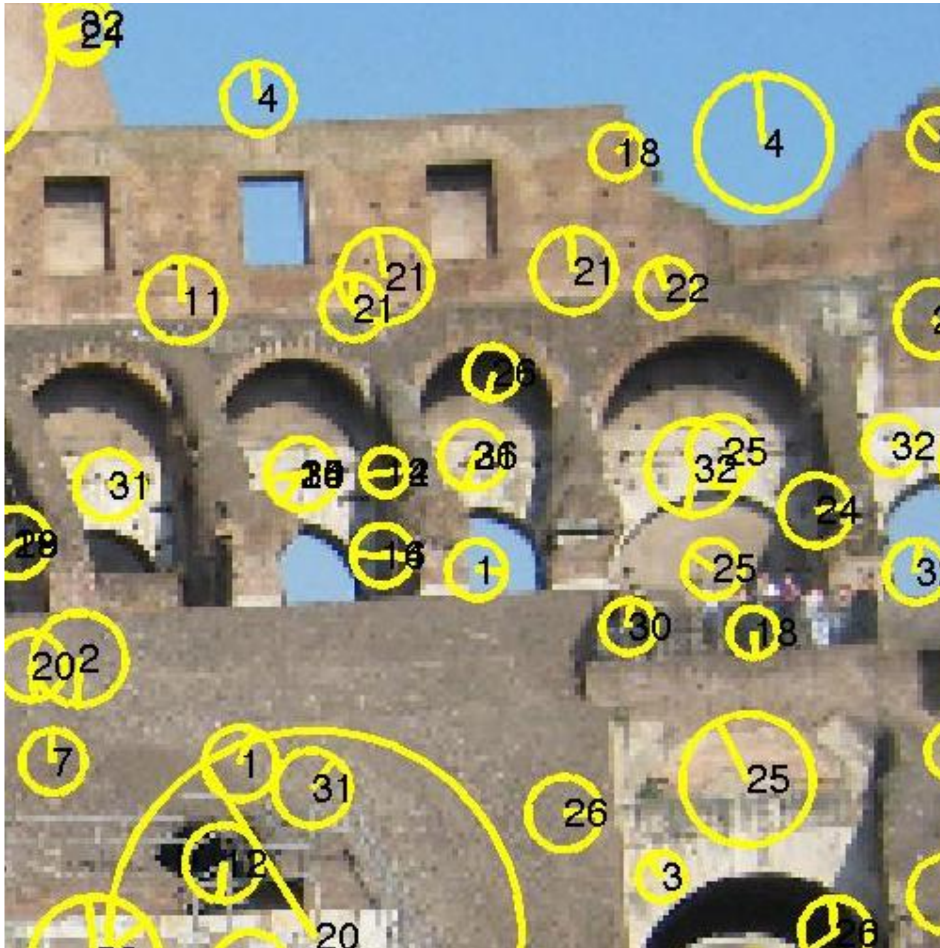
- Image query vs database (5000 images, 100 000 images)
  – mean average precision
- Image vs Image

Evaluation

- Image query vs database (5000 images, 100 000 images) – mean average precision
- Image vs Image
- Descriptor vs descriptor

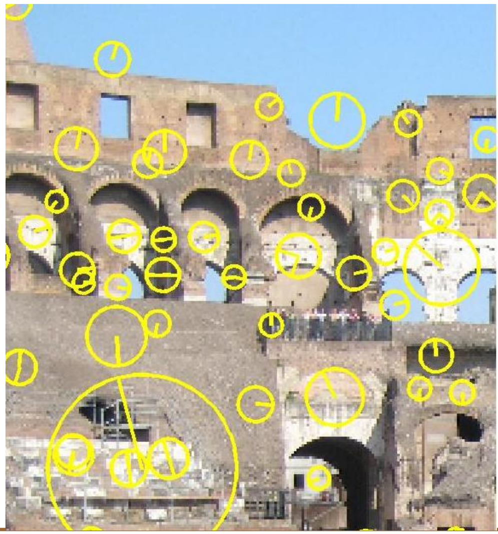


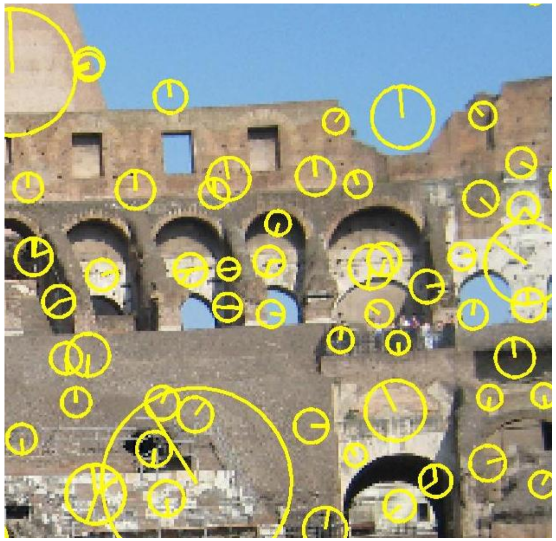




Evaluation

- Image query vs database (5000 images, 100 000 images) – mean average precision
- Image vs Image
- Descriptor vs descriptor
- Feature vs feature





Evaluation

- Image query vs database (5000 images, 100 000 images) – mean average precision
- Image vs Image
- **Descriptor vs descriptor**
- Feature vs feature

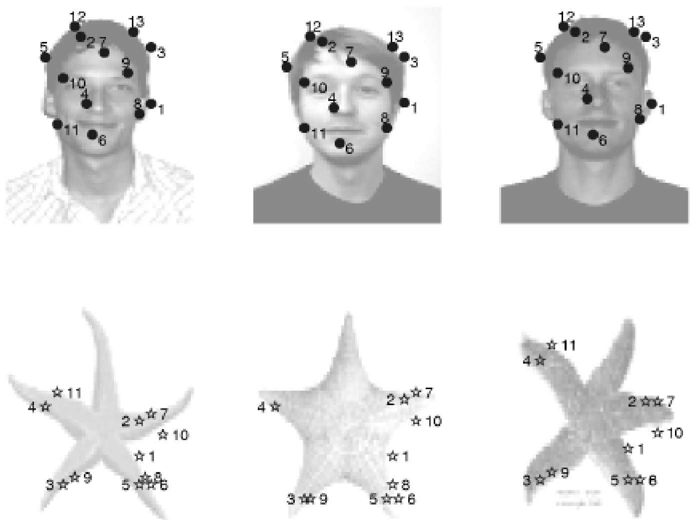
- Aim: Improve on features and descriptors by evaluating and learning already on lower levels
- For this we need **ground truth** correspondences
- We aim at fast bottom up processes
- We use heavier algorithms for generating ground truth
- **Feedback**

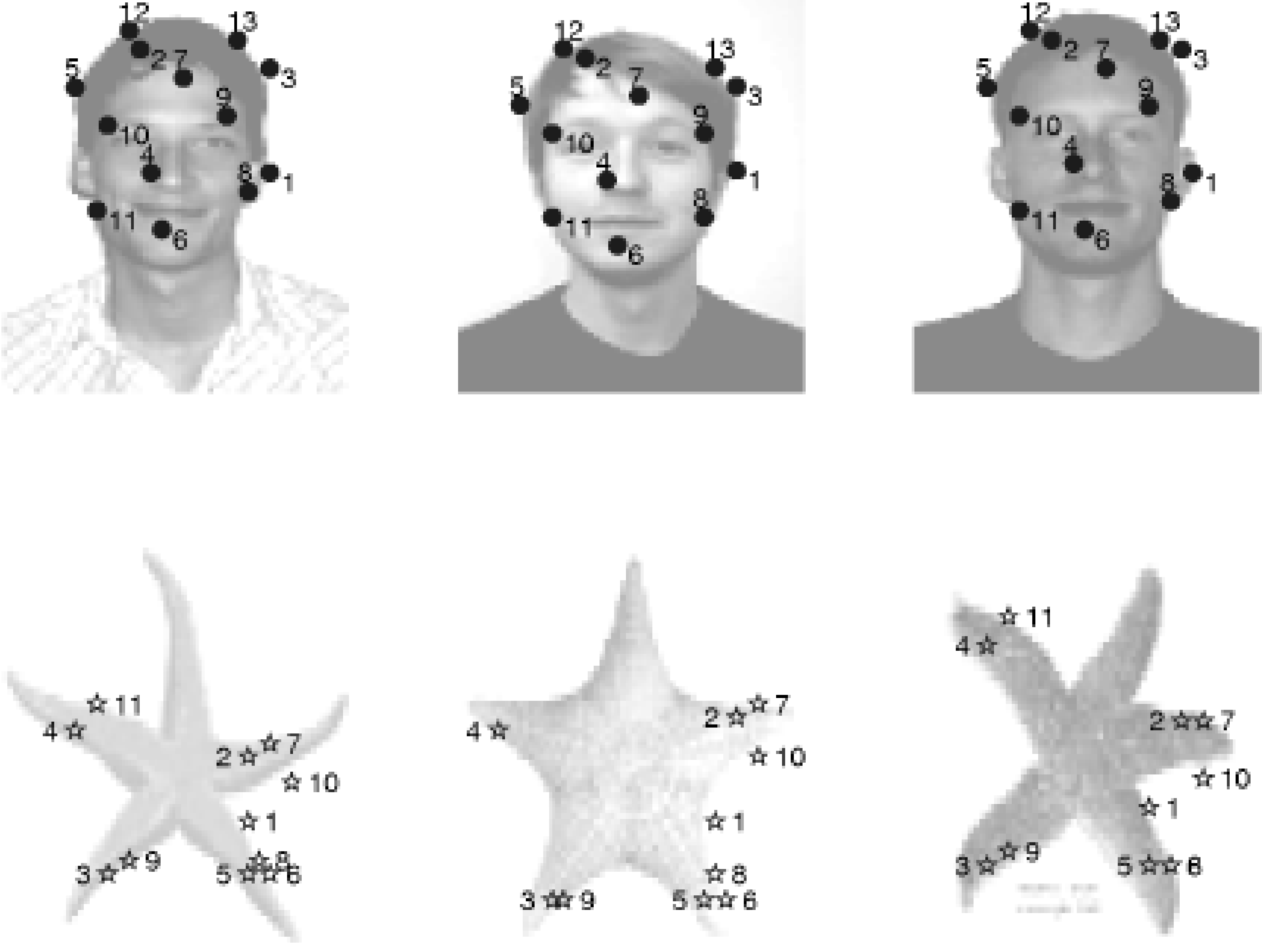
Obtaining ground truth data

- Static scenes

Obtaining ground truth data

- Static scenes
- Matches using deformable shape models (Karlsson-Åström ICPR 2008, CVPR 2008)





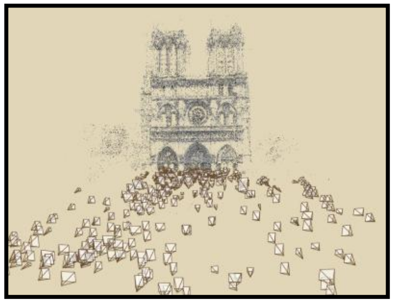
Obtaining ground truth data

- Static scenes
- Matches using deformable shape models
- Matches using geometry

Reconstruction pipeline

Input photographs

Scene reconstruction



Relative camera positions and orientations

Point cloud

Sparse correspondence

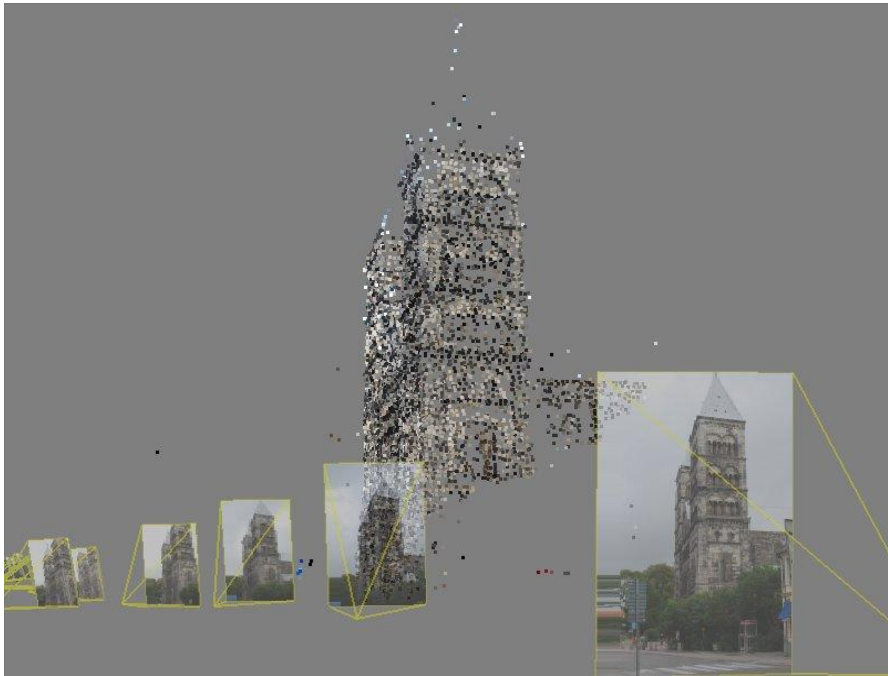
Photo Explorer

Slide courtesy of N. Snavely

Examples from Lund

Lundagård

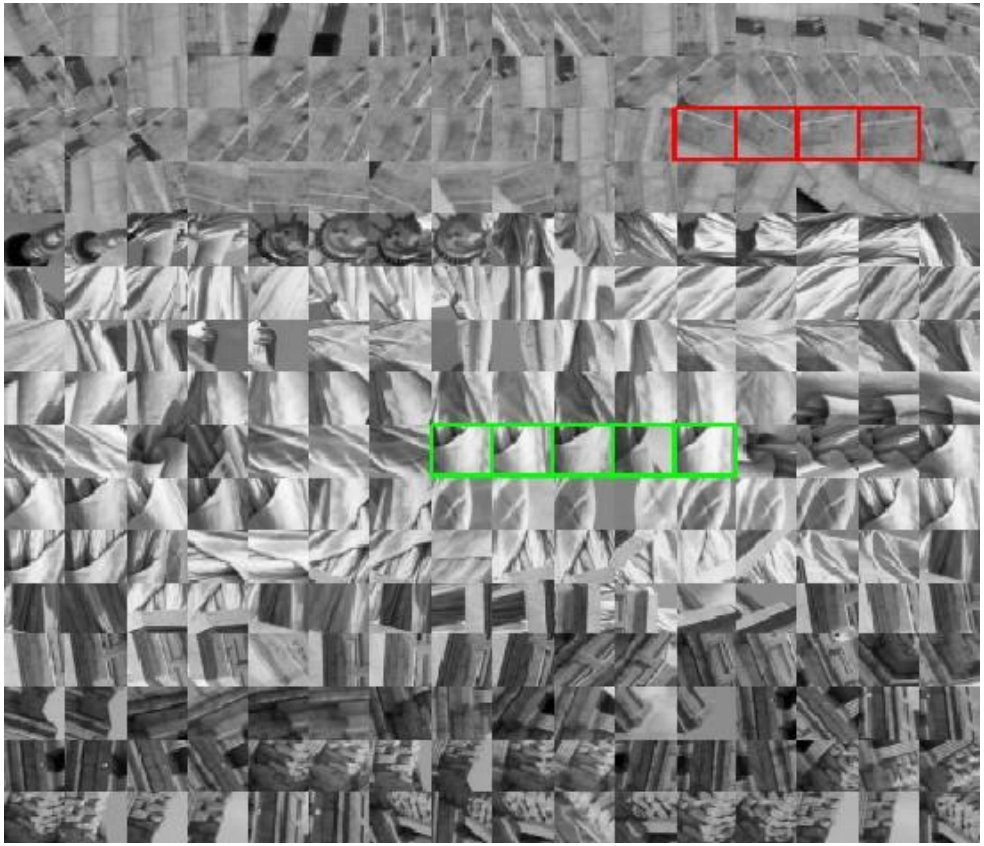
Domkyrkan



Ground-truth correspondences

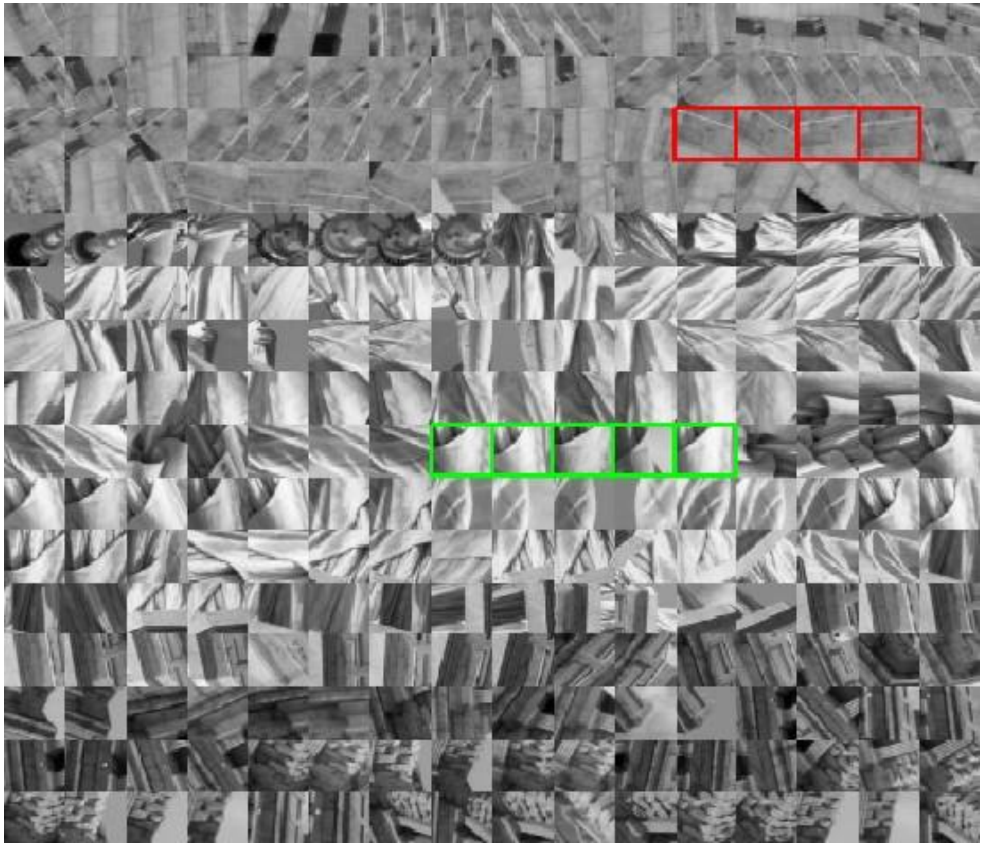
UBC Patch Data (Hua et al. CVPR'09)

Ground-truth correspondences



UBC Patch Data (Hua et al. CVPR'09)

Ground-truth correspondences

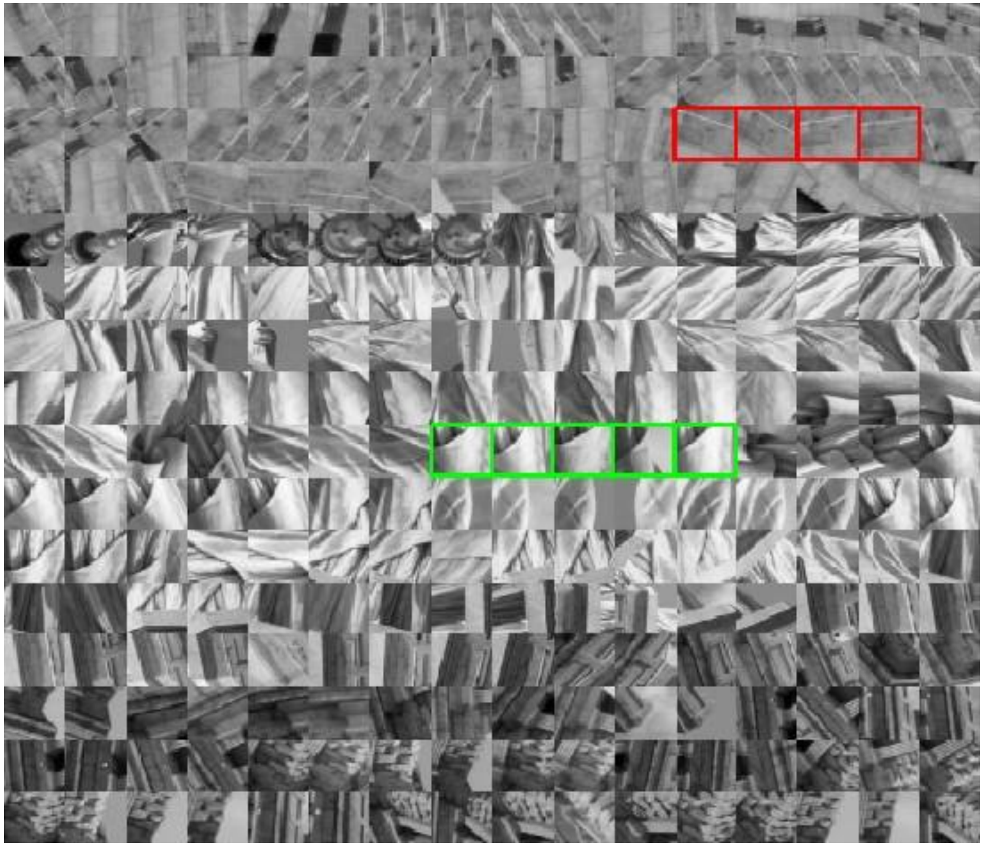


UBC Patch Data (Hua et al. CVPR'09)

- Patch correspondences obtained via 3D reconstructions

- Scale and orientation normalized

Ground-truth correspondences



UBC Patch Data (Hua et al. CVPR'09)

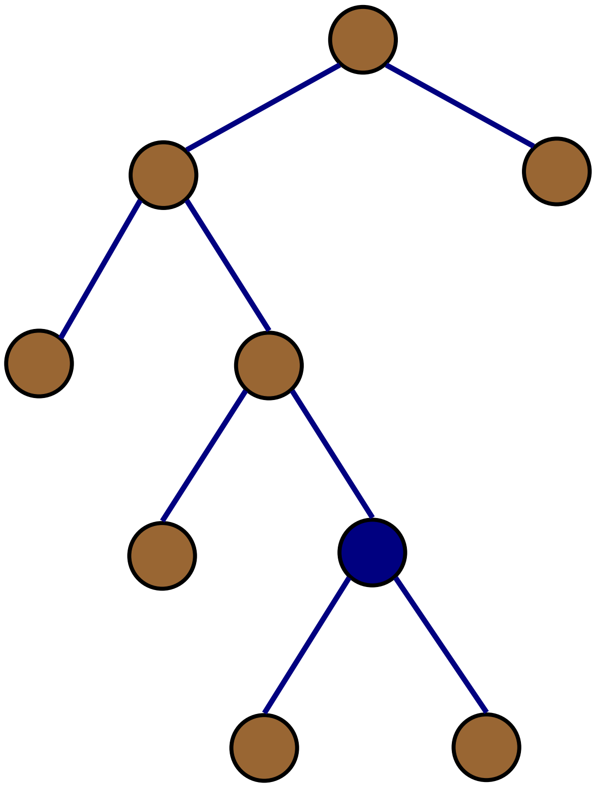
- Patch correspondences obtained via 3D reconstructions

- Scale and orientation normalized

We train the vocabulary in the manner that corresponding patches tend to fall in the same word (cluster)

Approximately 1.5 million patches in 0.6 million classes

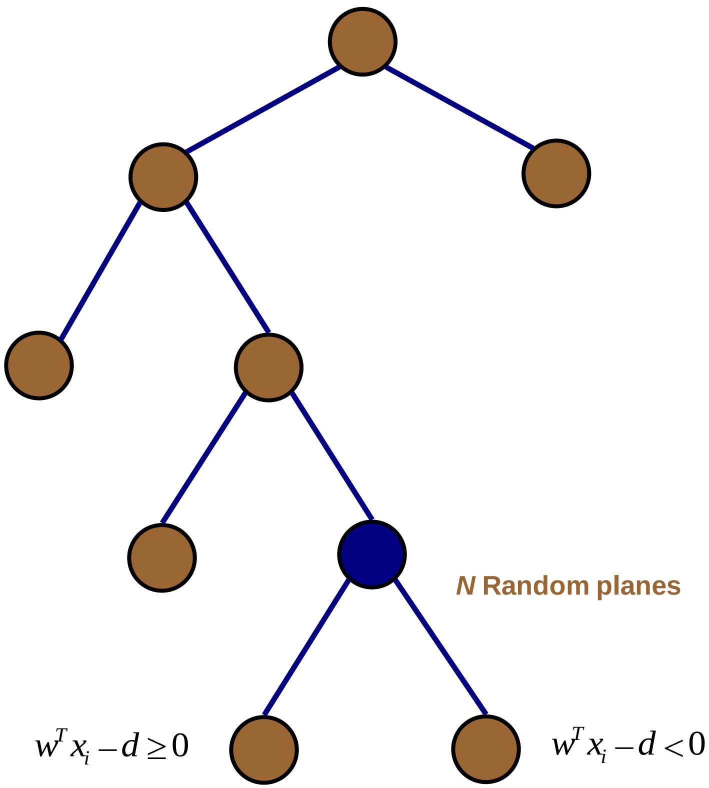
Our approach



- **Hierarchical splits**

  - At each level the features are split into 2 clusters

Our approach



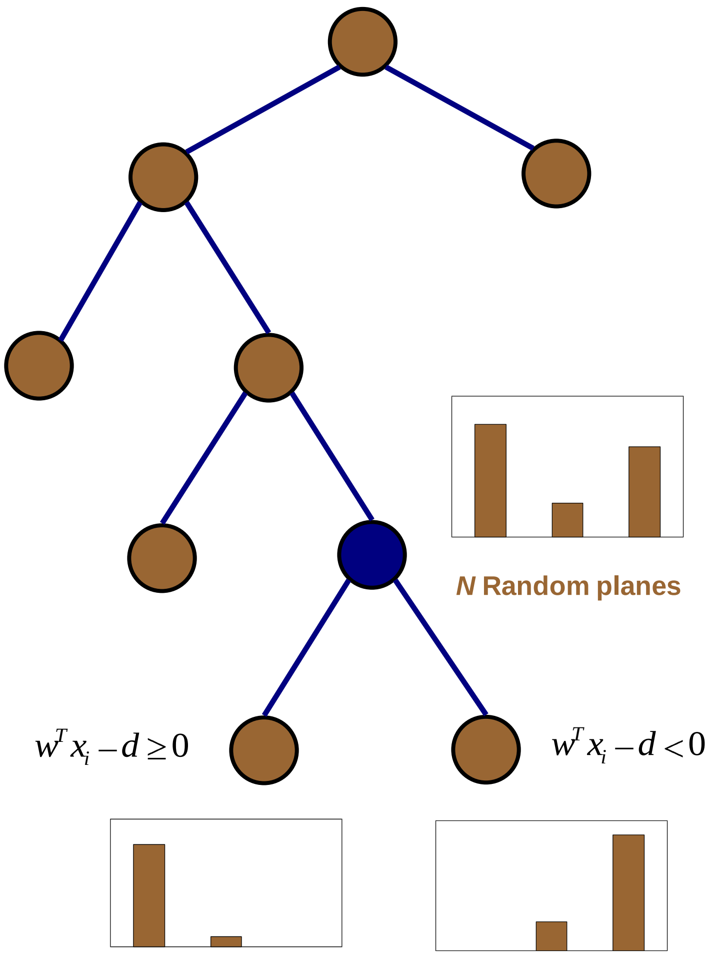
- **Hierarchical splits**

  - At each level the features are split into 2 clusters

- **Random plane**

  - At each split node, *N* (~1000) planes are randomly generated, and the one with lowest entropy is selected

Our approach



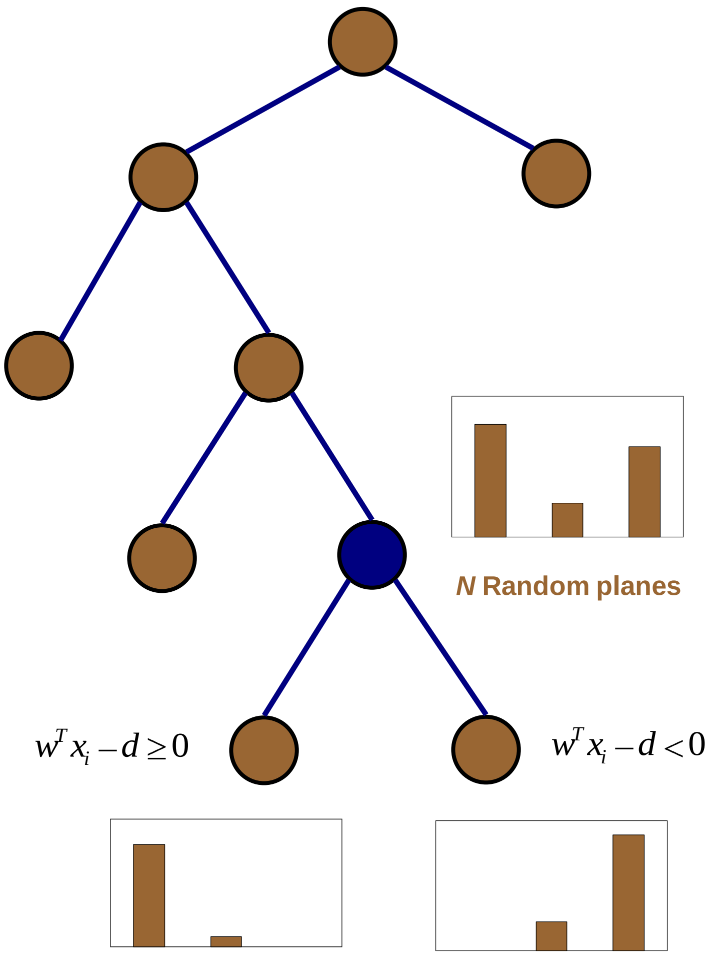
- **Hierarchical splits**

  - At each level the features are split into 2 clusters

- **Random plane**

  - At each split node, *N* (~1000) planes are randomly generated, and the one with **lowest entropy** is selected

Our approach



- **Hierarchical splits**

  - At each level the features are split into 2 clusters

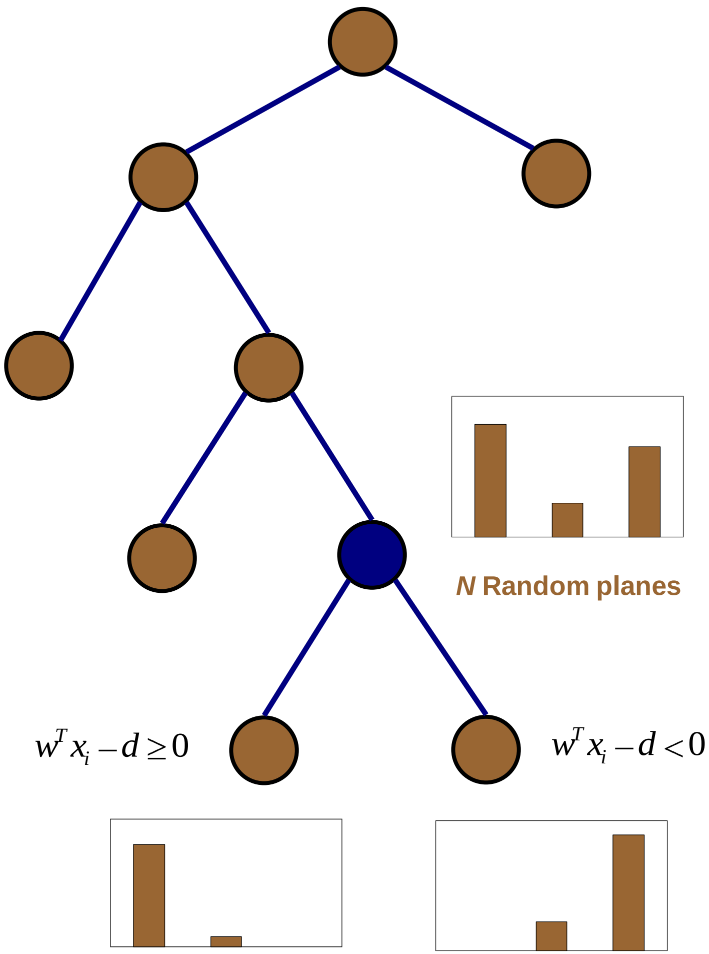
- **Random plane**

  - At each split node, *N* (~1000) planes are randomly generated, and the one with lowest entropy is selected

- **Local optimum**

  - Perturbing the selected plane to obtain a locally optimal plane w.r.t the entropies

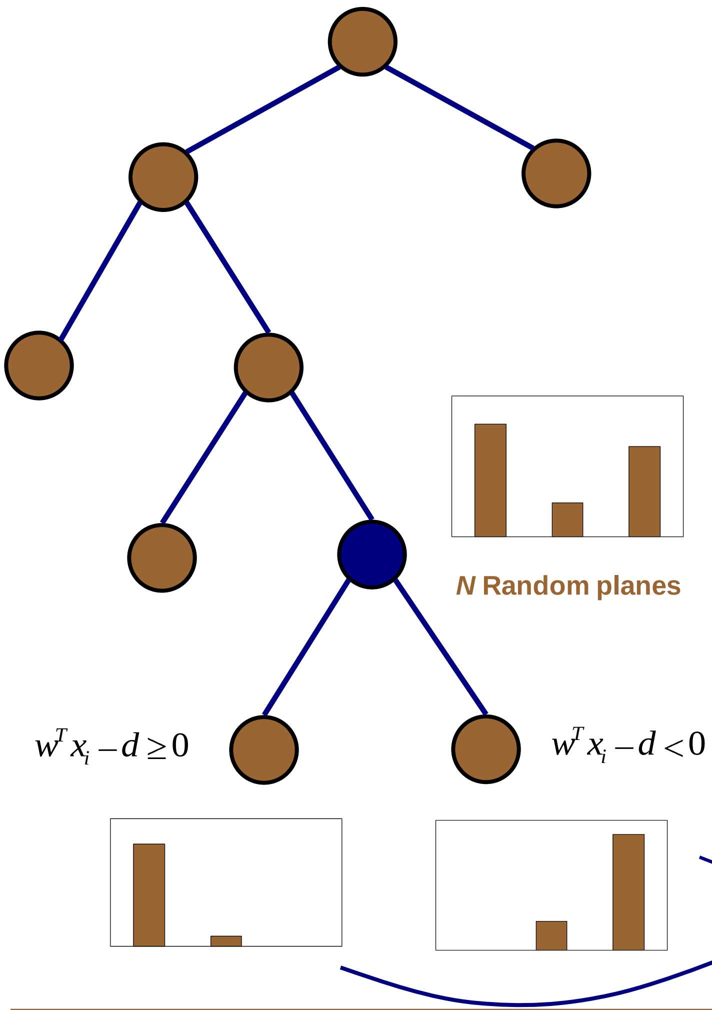
Our approach - Entropy



$$E_{l,r} = -\sum_{i=1}^{N} p_{l,r}(i)\log_2(p_{l,r}(i))$$

$$E_{split} = \frac{n_l}{n_{tot}} E_l + \frac{n_r}{n_{tot}} E_r$$

Our approach - Entropy



$$E_{l,r} = -\sum_{i=1}^{N} p_{l,r}(i)\log_2(p_{l,r}(i))$$

$$E_{split} = \frac{n_l}{n_{tot}} E_l + \frac{n_r}{n_{tot}} E_r$$

Our approach - Soft Assignment

- **Margin**

  - Instead of assigning each feature exactly to 1 cluster, we assign features smoothly to both clusters according to their **distances** to the best plane.

Our approach - Soft Assignment

- **Margin**

  - Instead of assigning each feature exactly to 1 cluster, we assign features smoothly to both clusters according to their **distances** to the best plane.

  - The smoothness is determined by the parameter *m – the margin.*

$$f(t) = \frac{1}{1+e^{-t/m}}$$

Our Approach - Soft Assignment

- **Margin**

  - Instead of assigning each feature exactly to 1 cluster, we assign features smoothly to both clusters according to their directional **distances** to the best plane.

  - The smoothness is determined by the parameter *m – the margin.*

$$t = w^T x_i - d \qquad f(t) = \frac{1}{1+e^{-t/m}}$$

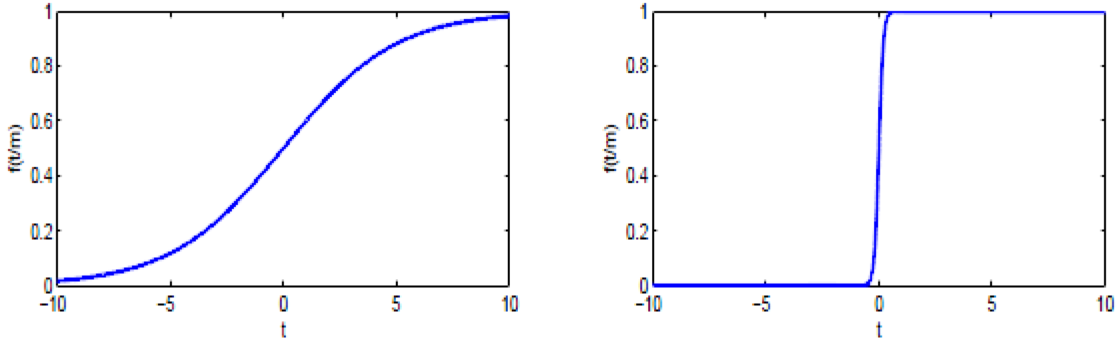


Fig. 1. Left: $f(t/m)$ for large m Right: $f(t/m)$ for small m

Entropy minimization - local search

- **Gradients derivation**

  - the gradients of the entropy w.r.t the random plane direction (*w*), offset(*d*) and the margin (*m*) are derived for optimization step

Entropy minimization - local search

- **Gradients derivation**

  - the gradients of the entropy w.r.t the random plane direction (w), offset(d) and the margin (m) are derived for optimization step

- **Optimization**

  - Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (Broyden, 1970) is used to find the local minimum

Experiments

- Extract SIFT features on the patches with default settings

- 20% of Statue of Liberty data for training
- 10% of non-overlapped Statue of Liberty data for testing

Result Evaluation

For evaluation --

- **Matched pairs**

  - sets of pair-wise matching within each class
- **Non-matched pairs**

  - pick an unmatched randomly for each feature

Result Evaluation

For evaluation --

- **Matched pairs**

  - sets of pair-wise matching within each class

- **Non-matched pairs**

  - pick an unmatched randomly for each feature

- **TP ('True Positive')**

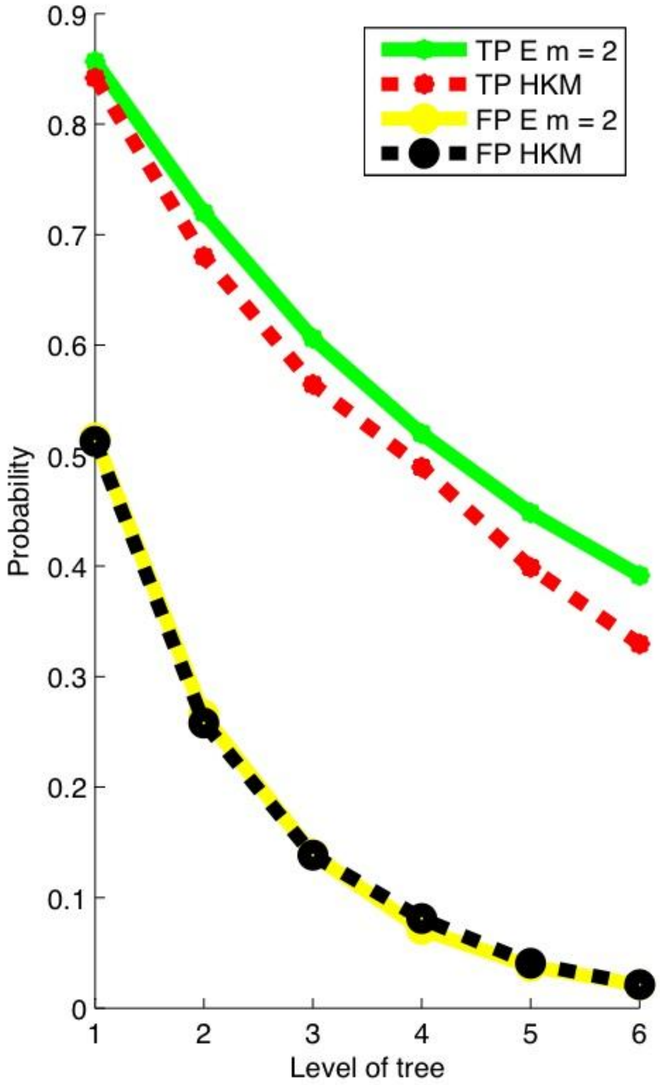
  - the percentage of matched pairs get the same word ID

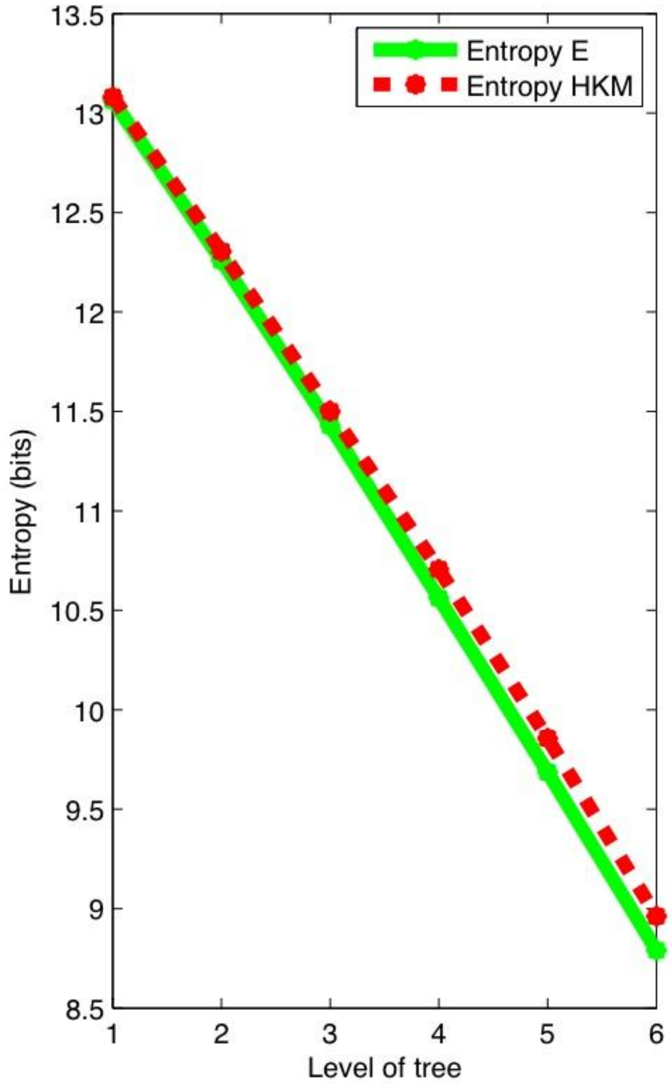
- **FP ('False Positive')**

  - the percentage of non-matched pairs get the same word ID

Results

- Compare with hierarchical kmeans with 2 splits





Results (cont.)

- Different margins

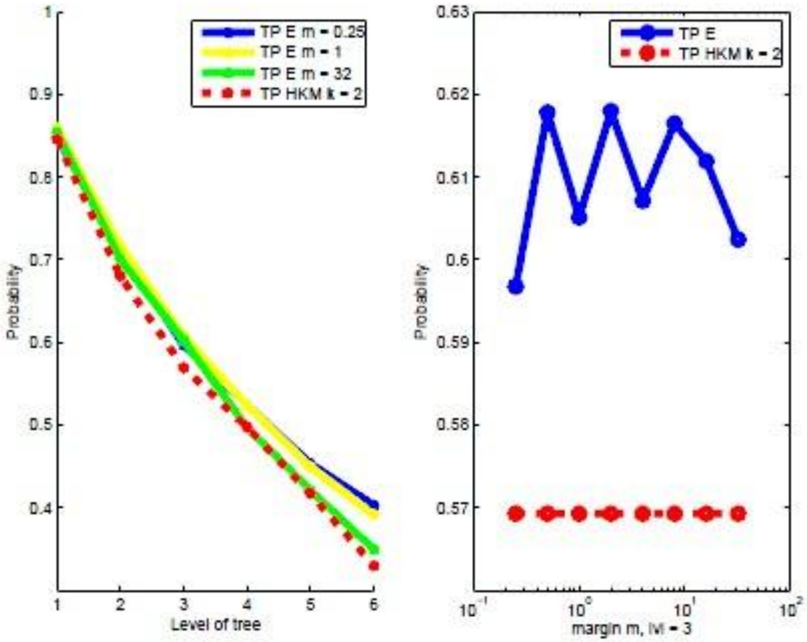


Fig. 4. Left: Estimated probability of two corresponding (TP) descriptors ending up in the same word as a function of tree depth for Hierarchical K-means (HKM) and for entropy optimization with three settings of margin m. Right: Estimated probability of two corresponding (TP) descriptors ending up in the same word as a function of margin m.

Future Work

- Generalize to K splits at each levels

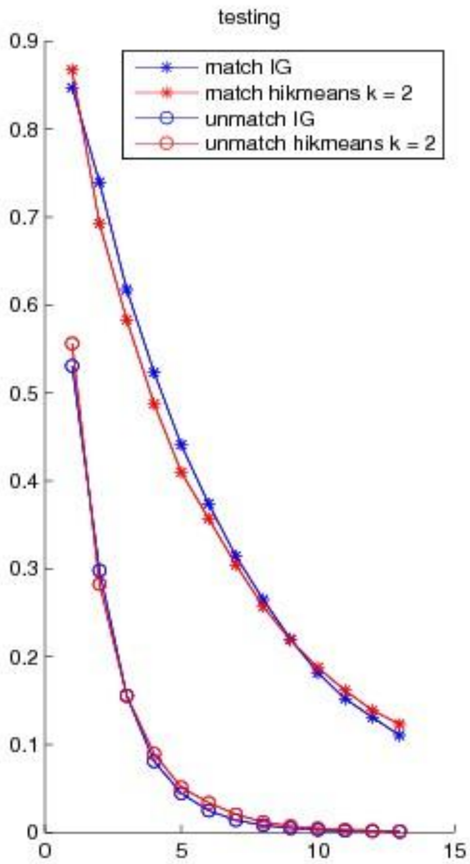
- Create additional large ground-truth dataset using geometry with images from Lund, Malmö, Stockholm

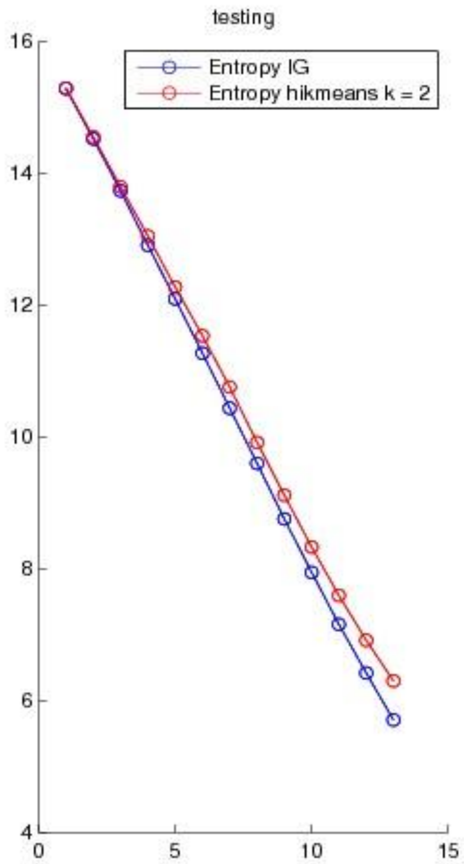
- Use soft margin

- Vocabulary for combinations of words

Thank you for your attention!

Further experiments





50% liberty for training

50% notredame for testing

Put in the Oxford pipeline

Train on 50% of the whole patch data (~750K features) with 65K words

HIK-2splits	Entropy-opt
0.1641	0.1956

State of the art….train on 5M features with 50K words (Philbin 2007)

kmeans	AKM
0.464	0.453