Feature Map Hashing: Sub-linear Indexing of Appearance and Global Geometry

Yannis Avrithis, Giorgos Tolias and Yannis Kalantidis

Image, Video and Multimedia Systems Laboratory
National Technical University of Athens

ACM Multimedia 2010, 25-29th October 2010, Firenze, Italy

Outline

Outline

Object retrieval

Problem description

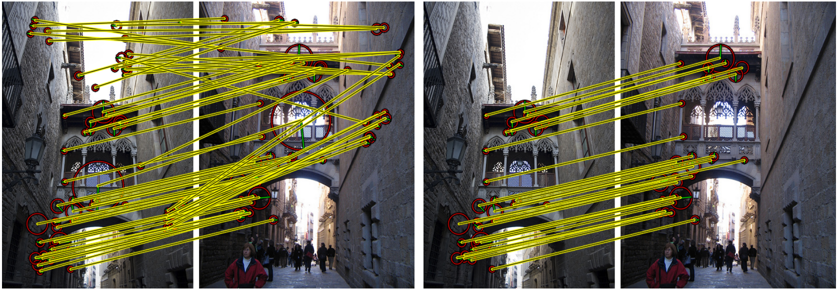
- Fast search in a large dataset of images
- Images depicting the same object
- Robustness against viewpoint change, photometric variations, occlusion and background clutter

Our goal

- Both appearance and geometry within the indexing process
- Fast search in all dataset images with geometric constraints

Background

- Extract local features and descriptors
- Create visual codebook using clustering/hashing techniques
- Map features to visual words with approximate nearest neighbor search
- Use visual words to find correspondences between features
- Find inliers with RANSAC or approximation



Appearance and geometry

Appearance only

- Discriminative local features and descriptors: an easy way to deal with view-point change and occlusion
- Bag-of-Words (BoW) in retrieval: good performance with low computational cost
- BoW discards spatial relations

Geometry

- Important in many problems of computer vision like feature correspondence, image registration, wide baseline stereo matching, object recognition, and retrieval
- Geometry essential to boost performance at large scale

State of the art limitations

Geometry for re-ranking

- Filtering stage: Based only on appearance [Sivic and Zisserman 2003]
- Re-ranking stage: Apply geometric or spatial constraints
- Geometric verification applied linearly only in the top ranking images [Philbin *et al.* 2007]

Indexing geometry

- Geometric hashing: only geometry, no appearance [Lamdan and Wolfson 1988][Chum and Matas 2006]
- Hough voting in transformation space: no feature quantization [Lowe 2004]
- Weak geometric information [Jegou *et al.* 2008]
- Geometric min-Hash: proximity constraints, small object discovery [Chum *et al.* 2009]

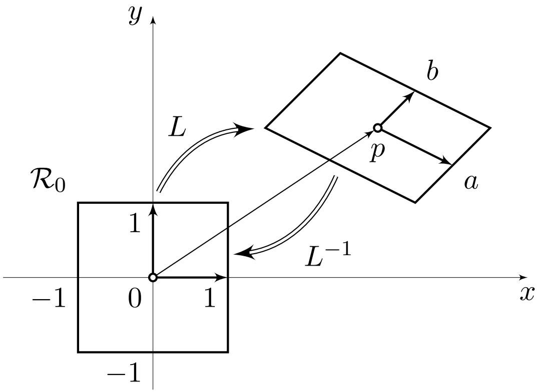
Overview of our approach

- Estimate image alignment via single correspondence
- For each feature construct a feature map encoding normalized positions and appearance of all remaining features
- An image is represented by a collection of such feature maps
- RANSAC-like matching is reduced to a number of set intersections
- Build inverted file of feature maps using min-wise independent permutations

Outline

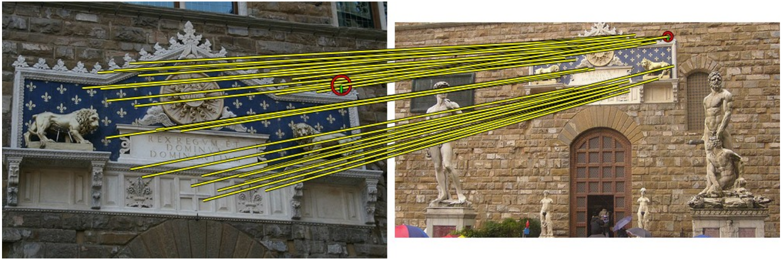
Local patches

- Each local feature is associated with an image patch L, which also represents an affine transform
- The rectified patch $\mathcal{R}_0$ is transformed to the patch via L
- The patch is rectified back to $\mathcal{R}_0$ via L^{-1}



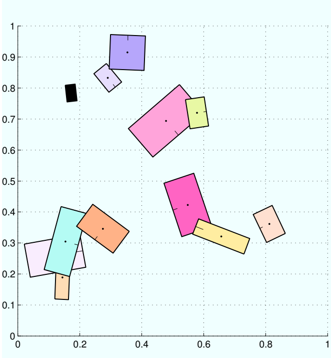
Single correspondence hypothesis

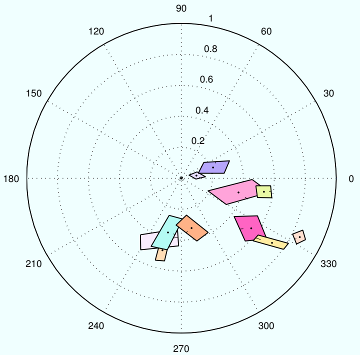
- A patch correspondence $L \leftrightarrow R$
- The transformation from one patch to the other is RL^{-1}
- Each correspondence provides a transformation hypothesis.
- Transformation hypotheses are now $O(n)$ and we can compute them all [Philbin *et al.* 2007]



Feature set rectification

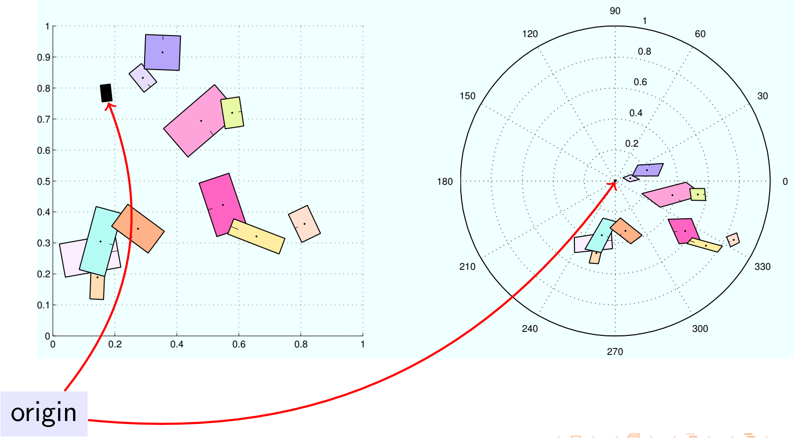
- Rectify both feature sets by transformations L^{-1} and R^{-1}, then compare
- Extrapolate each local transform to the entire image frame
- Rectify the entire set of features in advance





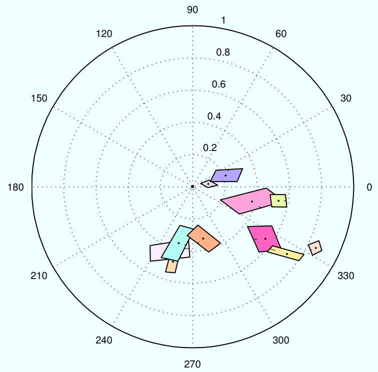
Feature set rectification

- Rectify both feature sets by transformations L^{-1} and R^{-1}, then compare
- Extrapolate each local transform to the entire image frame
- Rectify the entire set of features in advance



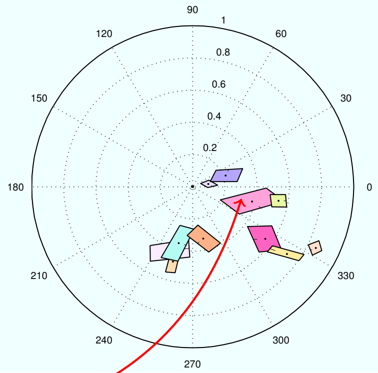
Spatial quantization

- Encode positions in polar coordinates (ρ, θ)
- Quantize positions in the rectified frames
- Define spatial codebook $\mathcal{U} \subseteq \mathbb{R}^2$ with $|\mathcal{U}| = k_\rho \times k_\theta = k_u$ bins



Spatial quantization

- Encode positions in polar coordinates (ρ, θ)
- Quantize positions in the rectified frames
- Define spatial codebook $\mathcal{U} \subseteq \mathbb{R}^2$ with $|\mathcal{U}| = k_\rho \times k_\theta = k_u$ bins



$\tilde{\rho} = 1, \tilde{\theta} = 11$

$k_\rho = 5, k_\theta = 12$

Feature maps

- An image is represented by a local feature set P
- Define the joint (visual-spatial) codebook $\mathcal{W} = \mathcal{V} \times \mathcal{U}$ with $|\mathcal{W}| = k_v k_u = k$ bins
- To construct a feature map we rectify a feature set and assign rectified features to spatial bins and visual words
- There is a different map for each origin; represent each image with a feature map collection F_P
- Can be seen as a local descriptor encoding the global feature set rectified in a local coordinate frame

$$f_P(\hat{x}) \ = \ h_{\mathcal{W}} \ \ (P^{(\hat{x})})$$

Feature maps

- An image is represented by a local feature set P
- Define the joint (visual-spatial) codebook $\mathcal{W} = \mathcal{V} \times \mathcal{U}$ with $|\mathcal{W}| = k_v k_u = k$ bins
- To construct a feature map we rectify a feature set and assign rectified features to spatial bins and visual words
- There is a different map for each origin; represent each image with a feature map collection F_P
- Can be seen as a local descriptor encoding the global feature set rectified in a local coordinate frame

$$f_P(\hat{x}) = h_{\mathcal{W}} \; (P^{(\hat{x})})$$

feature map of P wrt origin $\hat{x}$

Feature maps

- An image is represented by a local feature set P
- Define the joint (visual-spatial) codebook $\mathcal{W} = \mathcal{V} \times \mathcal{U}$ with $|\mathcal{W}| = k_v k_u = k$ bins
- To construct a feature map we rectify a feature set and assign rectified features to spatial bins and visual words
- There is a different map for each origin; represent each image with a feature map collection F_P
- Can be seen as a local descriptor encoding the global feature set rectified in a local coordinate frame

$$f_P(\hat{x}) = h_{\mathcal{W}} \left(P^{(\hat{x})}\right)$$

feature map of P wrt origin $\hat{x}$

rectified feature set P wrt origin $\hat{x}$

Feature maps

- An image is represented by a local feature set P
- Define the joint (visual-spatial) codebook $\mathcal{W} = \mathcal{V} \times \mathcal{U}$ with $|\mathcal{W}| = k_v k_u = k$ bins
- To construct a feature map we rectify a feature set and assign rectified features to spatial bins and visual words
- There is a different map for each origin; represent each image with a feature map collection F_P
- Can be seen as a local descriptor encoding the global feature set rectified in a local coordinate frame

$$f_P(\hat{x}) \;=\; h_{\mathcal{W}} \;\; (P^{(\hat{x})})$$

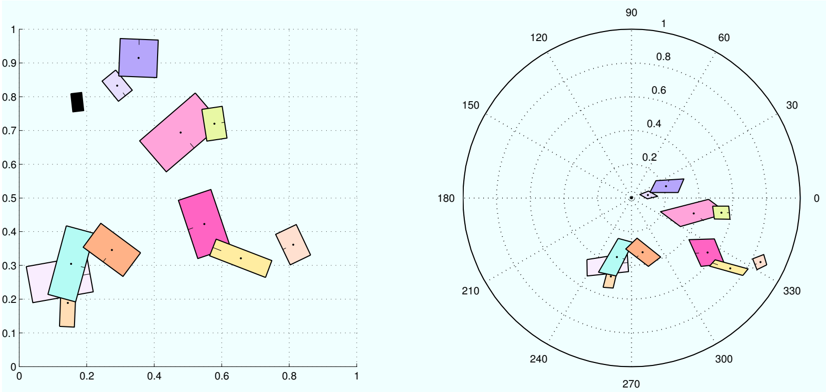
feature map of P wrt origin $\hat{x}$

histogram wrt joint codebook $\mathcal{W}$

rectified feature set P wrt origin $\hat{x}$

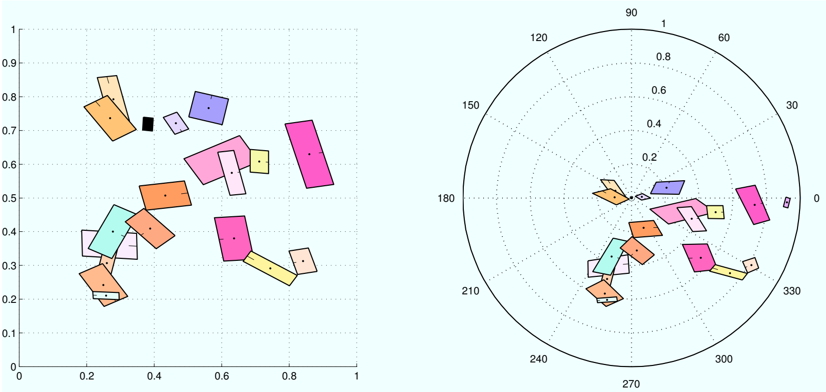
Feature maps – example

- Well aligned feature sets are likely to have maps with a high degree of overlap



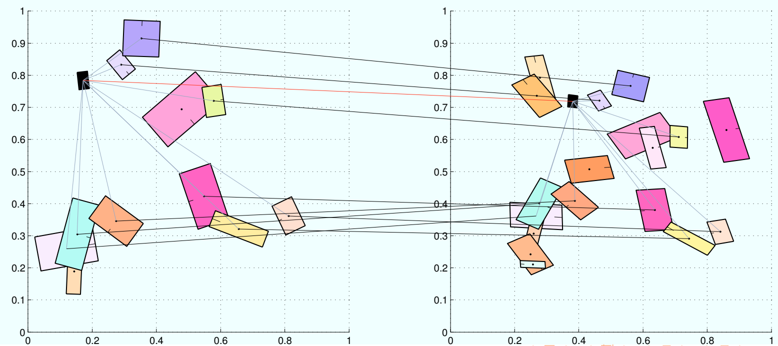
Feature maps – example

- Well aligned feature sets are likely to have maps with a high degree of overlap



Feature map similarity (FMS)

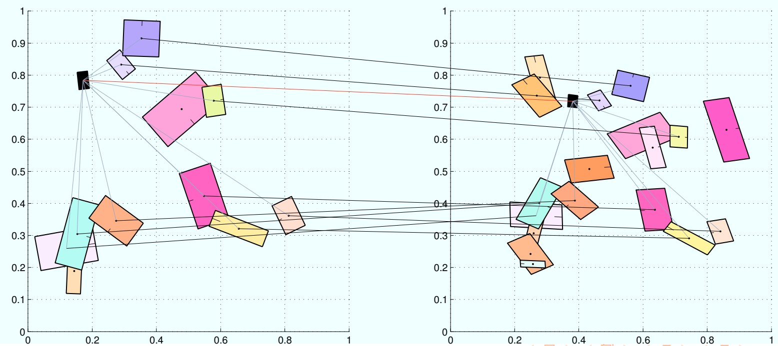
$$S_F(P,Q) = \max_{v \in V(P,Q)} \max_{\substack{\hat{x} \in H_v(P) \\ \hat{y} \in H_v(Q)}} f_P^{\mathrm{T}}(\hat{x}) \;\; f_Q(\hat{y})$$



Feature map similarity (FMS)

$$S_F(P,Q) = \max_{v \in V(P,Q)} \max_{\substack{\hat{x} \in H_v(P) \\ \hat{y} \in H_v(Q)}} f_P^{\mathrm{T}}(\hat{x}) \;\; f_Q(\hat{y})$$

feature map of image P wrt origin $\hat{x}$

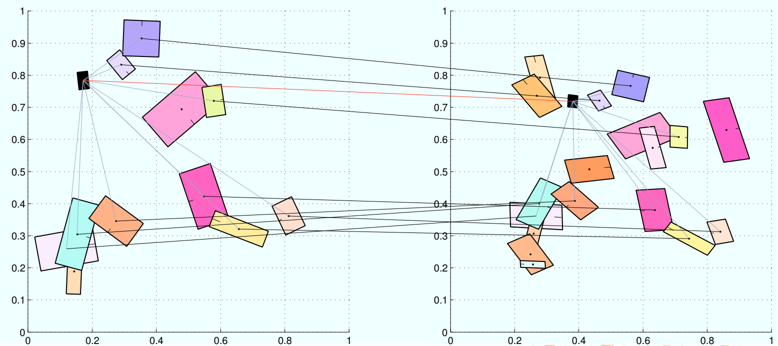


Feature map similarity (FMS)

$$S_F(P,Q) = \max_{v \in V(P,Q)} \max_{\substack{\hat{x} \in H_v(P) \\ \hat{y} \in H_v(Q)}} f_P^{\mathrm{T}}(\hat{x}) \;\; f_Q(\hat{y})$$

feature map of image P wrt origin $\hat{x}$

feature map of image Q wrt origin $\hat{y}$



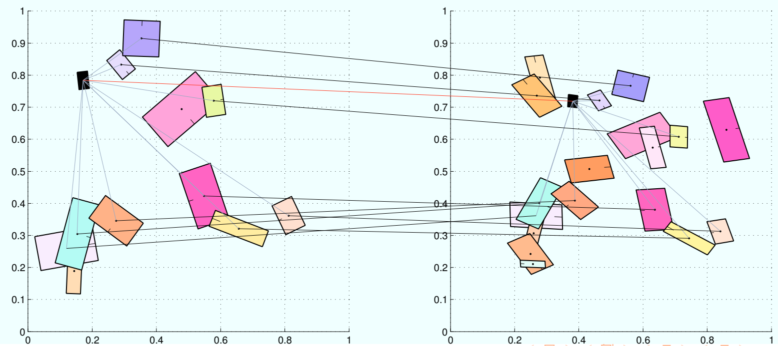
Feature map similarity (FMS)

for all origins mapped to visual word v

$$S_F(P,Q) = \max_{v \in V(P,Q)} \max_{\substack{\hat{x} \in H_v(P) \\ \hat{y} \in H_v(Q)}} f_P^{\mathrm{T}}(\hat{x}) \; f_Q(\hat{y})$$

feature map of image P wrt origin $\hat{x}$

feature map of image Q wrt origin $\hat{y}$



Feature map similarity (FMS)

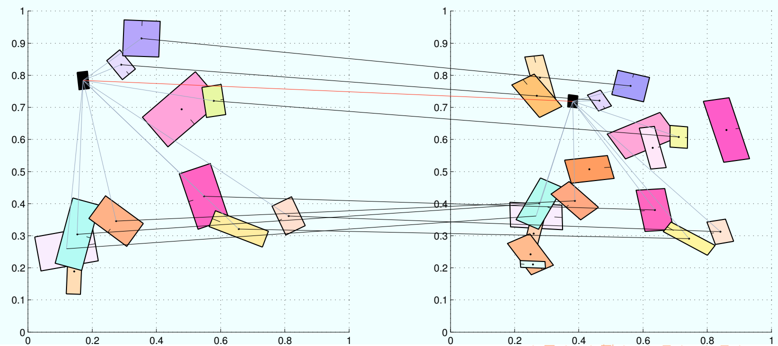
for all visual words that P, Q have in common

for all origins mapped to visual word v

$$S_F(P,Q) = \max_{v \in V(P,Q)} \max_{\substack{\hat{x} \in H_v(P) \\ \hat{y} \in H_v(Q)}} f_P^{\mathrm{T}}(\hat{x}) \;\; f_Q(\hat{y})$$

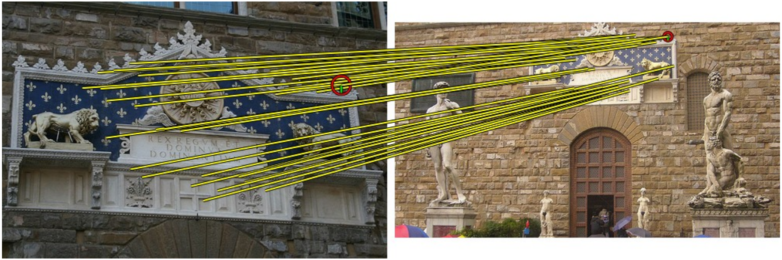
feature map of image P wrt origin $\hat{x}$

feature map of image Q wrt origin $\hat{y}$

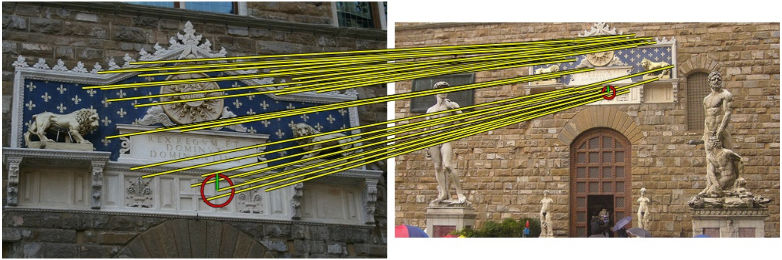


Feature map similarity - example

Inliers using fast spatial matching [FastSM - Philbin *et al.*] (35 inliers)

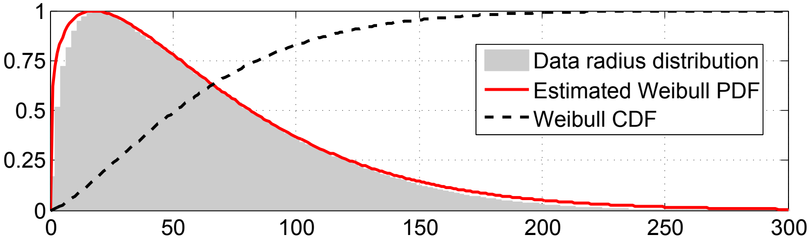


Inliers using feature map similarity (32 inliers)



Distribution of ρ

- Non-linear transformation using Weibull CDF
- Estimation of parameters via maximum likelihood
- Bins equally populated when distribution w.r.t. ρ is uniform



Memory savings – speed

Unique visual words

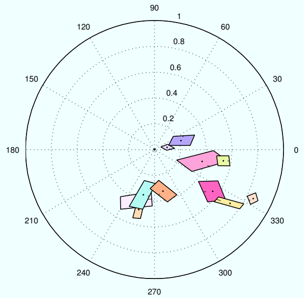
- Use as origins only features that map uniquely to visual words

Range parameter τ

- Add constraints on spatial proximity via range parameter τ
- $\tau \in [0, 1]$ controls the balance between local and global geometry

Origin selection

- Statistically measure which visual words get better aligned
- Select as origins only features mapped to those visual words



.

Memory savings – speed

Unique visual words

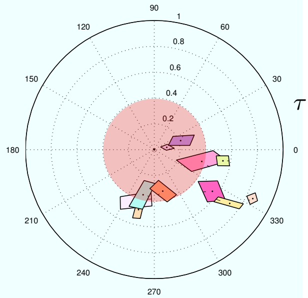
- Use as origins only features that map uniquely to visual words

Range parameter τ

- Add constraints on spatial proximity via range parameter τ
- $\tau \in [0, 1]$ controls the balance between local and global geometry

Origin selection

- Statistically measure which visual words get better aligned
- Select as origins only features mapped to those visual words



$\tau = 0.4$

.

Memory savings – speed

Unique visual words

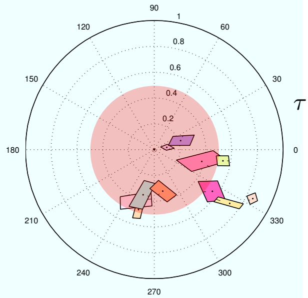
- Use as origins only features that map uniquely to visual words

Range parameter τ

- Add constraints on spatial proximity via range parameter τ
- $\tau \in [0, 1]$ controls the balance between local and global geometry

Origin selection

- Statistically measure which visual words get better aligned
- Select as origins only features mapped to those visual words



$\tau = 0.5$

.

Memory savings – speed

Unique visual words

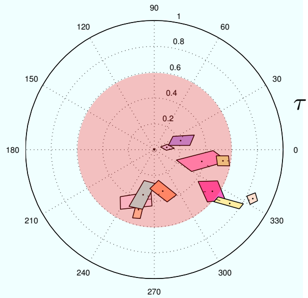
- Use as origins only features that map uniquely to visual words

Range parameter τ

- Add constraints on spatial proximity via range parameter τ
- $\tau \in [0, 1]$ controls the balance between local and global geometry

Origin selection

- Statistically measure which visual words get better aligned
- Select as origins only features mapped to those visual words



$\tau = 0.6$

.

Memory savings – speed

Unique visual words

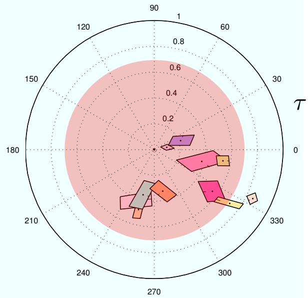
- Use as origins only features that map uniquely to visual words

Range parameter τ

- Add constraints on spatial proximity via range parameter τ
- $\tau \in [0, 1]$ controls the balance between local and global geometry

Origin selection

- Statistically measure which visual words get better aligned
- Select as origins only features mapped to those visual words



$\tau = 0.7$

.

Memory savings – speed

Unique visual words

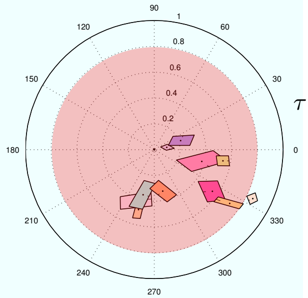
- Use as origins only features that map uniquely to visual words

Range parameter τ

- Add constraints on spatial proximity via range parameter τ
- $\tau \in [0, 1]$ controls the balance between local and global geometry

Origin selection

- Statistically measure which visual words get better aligned
- Select as origins only features mapped to those visual words



$\tau = 0.8$

.

Outline

Towards indexing

- FMS is a fast way of matching 2 images, but still not enough for indexing
- A feature map is an extremely sparse histogram; bin count typically takes values in $\{0, 1\}$
- Each feature map f is represented by set $\bar{f} \subset \mathcal{W}$ of non-empty bins

Min-wise independent permutations

- The feature space is now $\mathbb{F} = \mathcal{P}(\mathcal{W})$, the powerset of $\mathcal{W}$
- $h : \mathbb{F} \to \mathcal{W}$, hash function mapping objects back to $\mathcal{W}$
- $\pi : \mathbb{F} \to \mathbb{F}$, a random permutation
- Given a feature map $\bar{f} \subset \mathcal{W}$: compute a hash value $h(\bar{f}) = \min\{\pi(\bar{f})\}$

$$\Pr[\min\{\pi(\bar{f})\} = \min\{\pi(\bar{g})\}] = \frac{|\bar{f} \cap \bar{g}|}{|\bar{f} \cup \bar{g}|} = J(\bar{f}, \bar{g})$$

- Two features maps are hashed to the same value with probability equal to their resemblance or Jaccard similarity coefficient

Map sketch

- Construct a set $\Pi = \{\ \pi_i : i = 1, \ldots, m\}$ of m independent random permutations
- Represent each feature map $\bar{f}$ by map sketch $\mathbf{f} \in \mathcal{W}^m$,

$$\mathbf{f} = \mathbf{f}(\bar{f}) = [\min\{\pi_1(\bar{f})\}, \ldots, \min\{\pi_m(\bar{f})\}]^{\mathrm{T}}$$

- Sketch similarity, count number of elements that sketches $\mathbf{f}$, $\mathbf{g}$ have in common

$$s_K(\mathbf{f}, \mathbf{g}) = m - \|\mathbf{f} - \mathbf{g}\|_0$$

Feature map hashing (FMH)

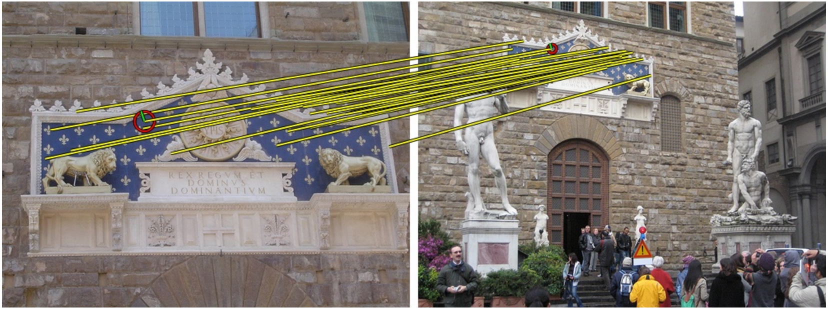
- Map sketch collection $\mathbf{F}$: set of all map sketches $\mathbf{f}$ of an image
- Image similarity reduces to sketch similarity

$$S_M(\mathbf{F}, \mathbf{G}) = \max_{\mathbf{f} \in \mathbf{F}} \max_{\mathbf{g} \in \mathbf{G}} s_K(\mathbf{f}, \mathbf{g})$$

- Collisions may appear for several pairs of maps; sum all sketch similarities instead of keeping the best one

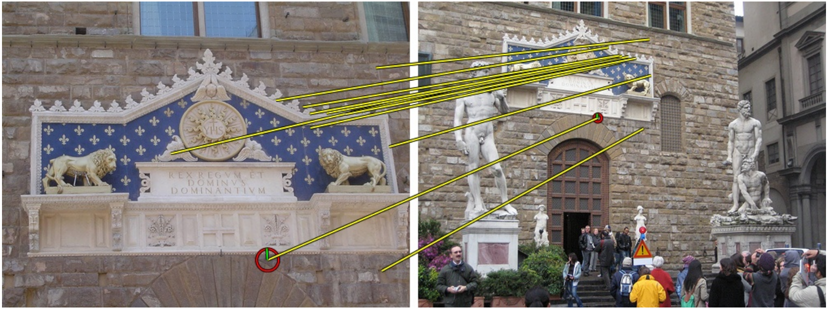
$$S_K(\mathbf{F}, \mathbf{G}) = \sum_{\mathbf{f} \in \mathbf{F}} \sum_{\mathbf{g} \in \mathbf{G}} s_K(\mathbf{f}, \mathbf{g})$$

Matching maps



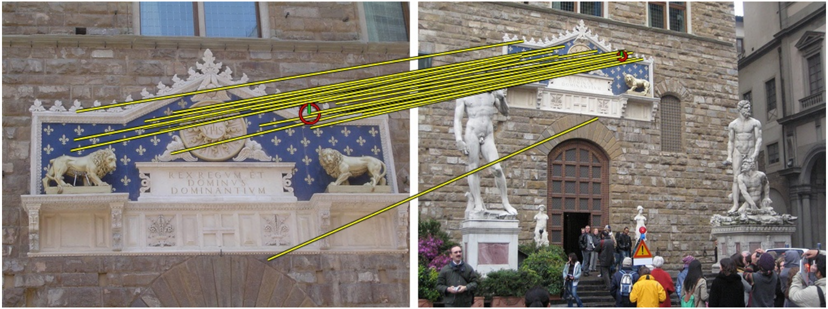
Multiple matching pairs of feature maps

Matching maps



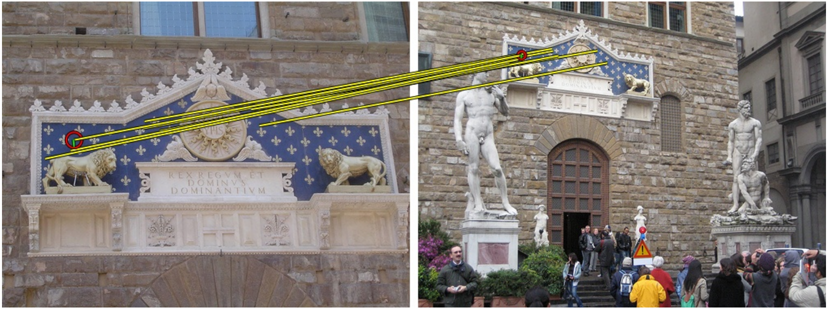
Multiple matching pairs of feature maps

Matching maps



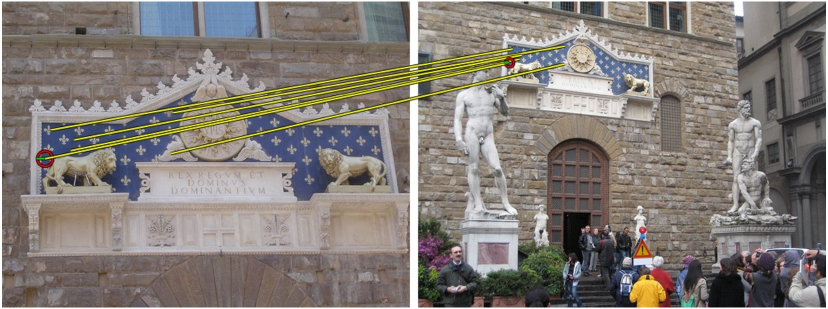
Multiple matching pairs of feature maps

Matching maps



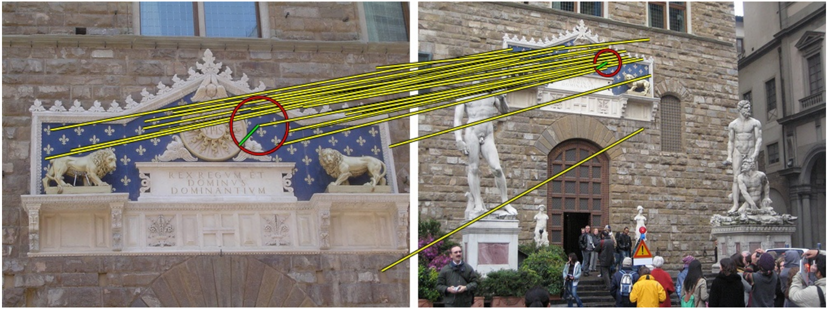
Multiple matching pairs of feature maps

Matching maps



Multiple matching pairs of feature maps

Matching maps



Multiple matching pairs of feature maps

Indexing

Index construction

- Represent the entire dataset by triplet $(\hat{v}, w, \pi)$ (origin, sketch element, permutation)
- Use an inverted file for sub-linear search
- Memory requirements $5\times$ a typical baseline system

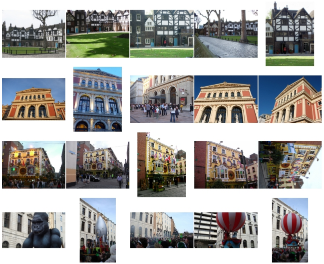
Query

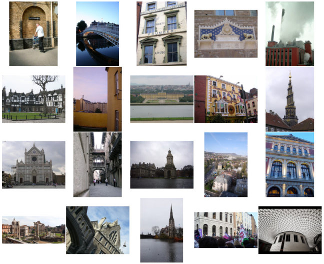
- Construct triplet $(\hat{v}, w, \pi)$ for query image
- Rank images with a voting process
- Re-estimate transformation parameters using LO-RANSAC
- Re-ranking is an order of magnitude faster than FastSM, because an initial estimate is already available

Outline

European Cities Dataset 50K (EC50K)

- 778 Annotated images
- 20 groups of photos
- 5 queries from each group

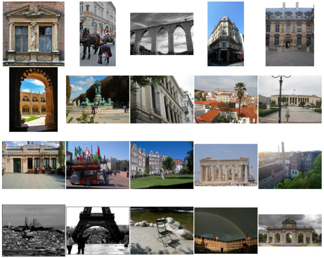


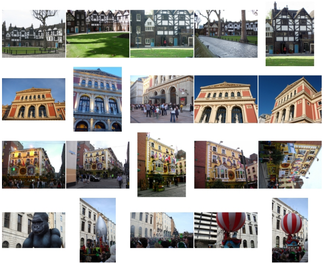


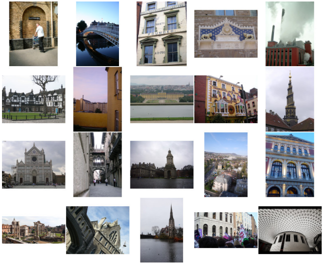
Publicly available: `http://image.ntua.gr/iva/datasets/ec50k`

European Cities Dataset 50K (EC50K)

- 778 Annotated images
- 20 groups of photos
- 5 queries from each group
- 50,000 distractor images

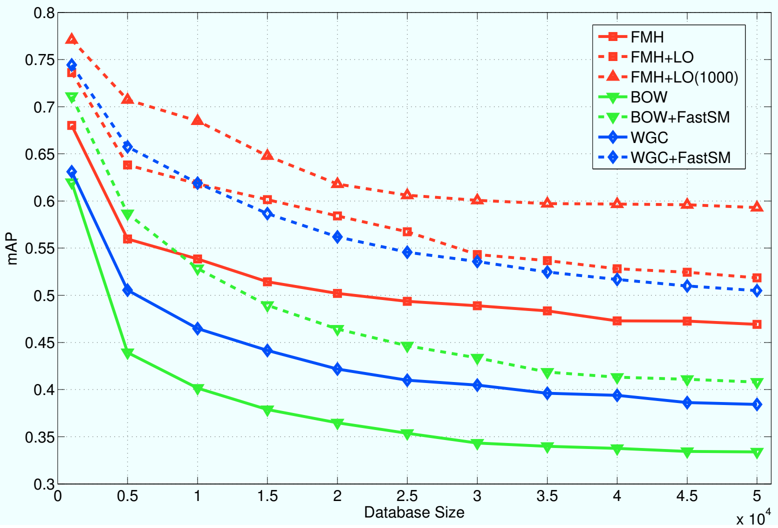






Publicly available: `http://image.ntua.gr/iva/datasets/ec50k`

Results EC50K



Results Oxford Buildings - Inria Holidays

Dataset	Holidays		Oxford	
Method	1.4K	51.4K	5K	55K
BOW	0.583	0.492	0.372	0.329
WGC	0.591	0.510	**0.375**	0.333
FMH	**0.610**	**0.542**	0.362	**0.362**
BOW+FastSM	0.622	0.537	0.421	0.356
WGC+FastSM	0.626	0.542	**0.436**	0.388
FMH+LO(100)	**0.639**	0.556	0.422	0.391
FMH+LO(1000)	-	**0.571**	0.431	**0.410**

Retrieval Examples

Outline

Discussion – future work

Discussion

- First work to integrate appearance and global geometry in sub-linear image indexing
- We make spatial matching work at large scale, and demonstrate how this keeps precision almost unaffected under a significant amount of distractors
- We see it as a challenge for future feature detectors to achieve better alignment

Future work

- Mine frequent feature maps from large image dataset
- Create codebook of feature maps

FMH page:

http://image.ntua.gr/iva/research/feature_map_hashing

EC50K dataset page:

http://image.ntua.gr/iva/datasets/ec50k

Thank you!