Image matching and visual search

Local features and geometry

Yannis Avrithis

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

Bordeaux, March 2011

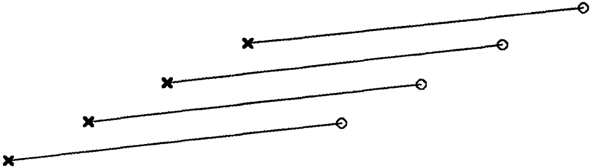
Outline

1. Visual search, local features and bag-of-words
2. Local features based on distance maps
3. Geometry indexing: feature map hashing
4. Relaxed spatial matching and re-ranking
5. Photo collections: view clustering and scene maps
6. Location and landmark recognition
7. Implementation: `ivl` library

Outline

Matching local feature points

[Scott and Longuet-Higgins 1991]



- given two sets of points $a_i, i = 1, \ldots, m$ and $b_j, j = 1, \ldots, n$ on the same plane, let d_{ij} be the distance between a_i and b_j
- following earlier theories of Ullman and Marr, the problem is to associate points a_i and b_j in a one-to-one correspondence such that the sum of squared distances between corresponding points is minimized

A spectral approach

1. construct the $m \times n$ proximity matrix G with elements

$$g_{ij} = \exp(-d_{ij}^2 / 2\sigma^2)$$

2. perform singular value decomposition of G

$$G = USV^{\mathrm{T}}$$

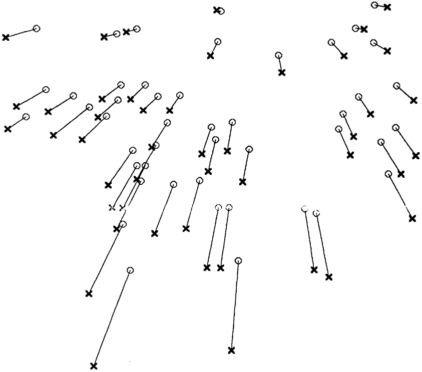
   where U, V are orthogonal matrices of dimension m, n and S is a non-negative diagonal $m \times n$ matrix

3. replace each diagonal element s_{ij} of S by 1 and reconstruct

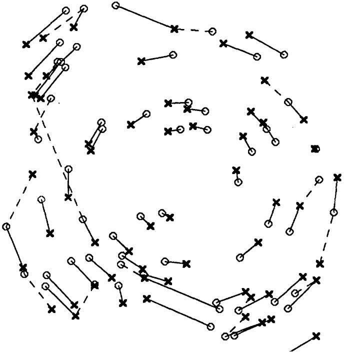
$$P = UEV^{\mathrm{T}}$$

4. finally, associate points a_i and b_j if element p_{ij} of P is the greatest element in its row and its column

A spectral approach



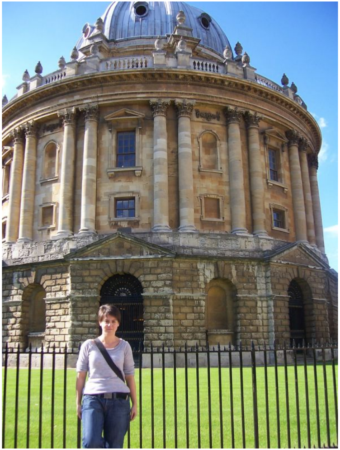
scale, translation



rotation

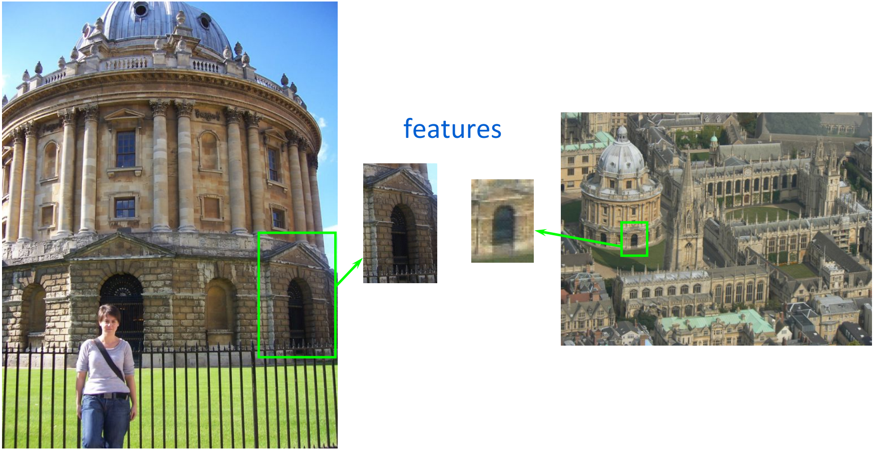
Matching discriminative local features

[Lowe 1999]



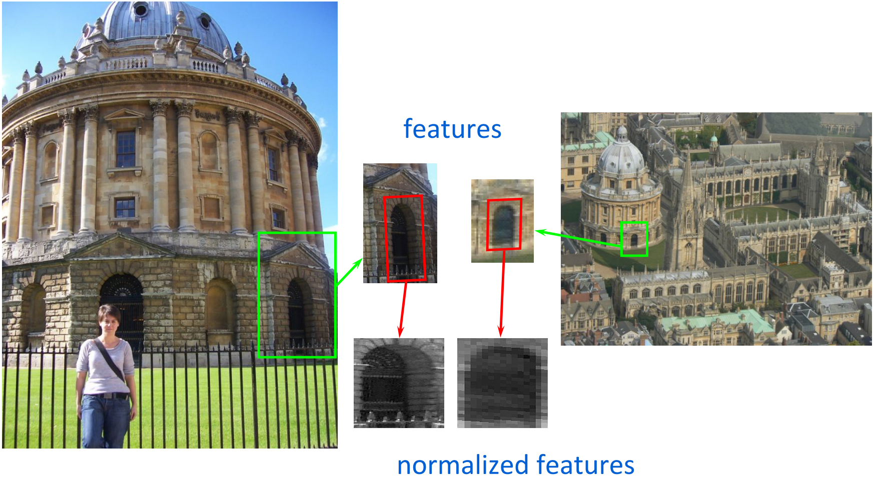
Matching discriminative local features

[Lowe 1999]



Matching discriminative local features

[Lowe 1999]

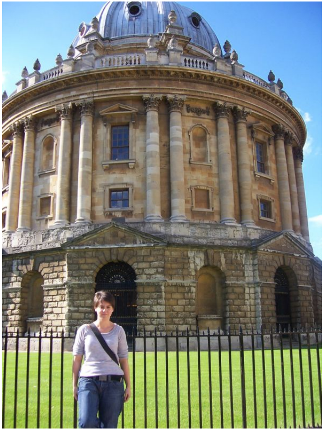


Forget about geometry: bag-of-words

[Sivic and Zisserman 2003]

Object → Bag of 'words'

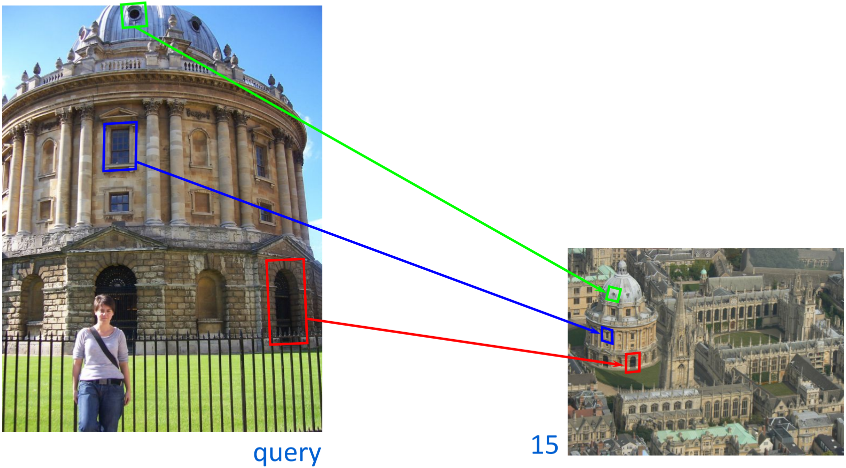
Vector quantization → visual words



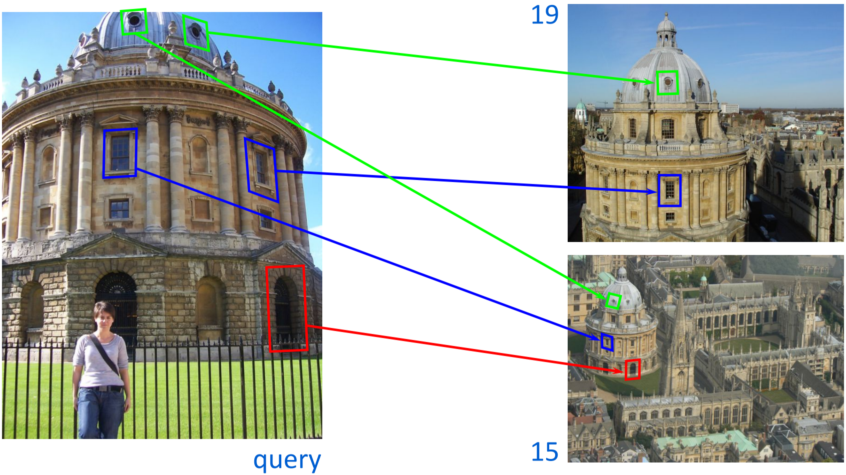
query

15

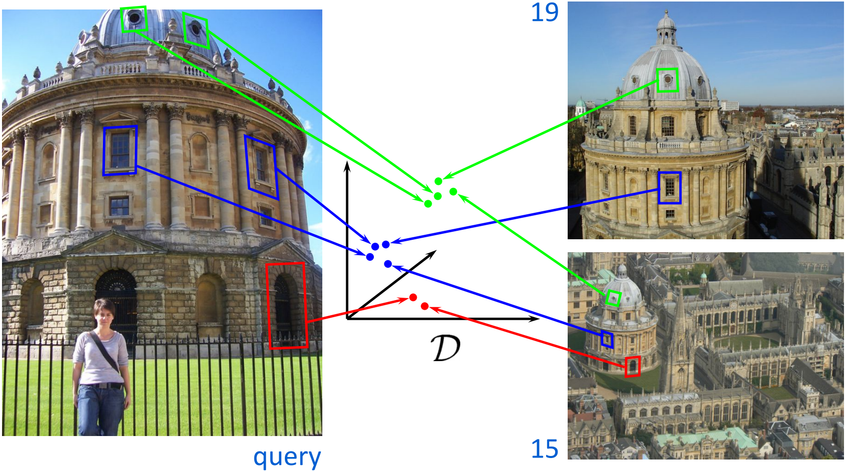
Vector quantization → visual words



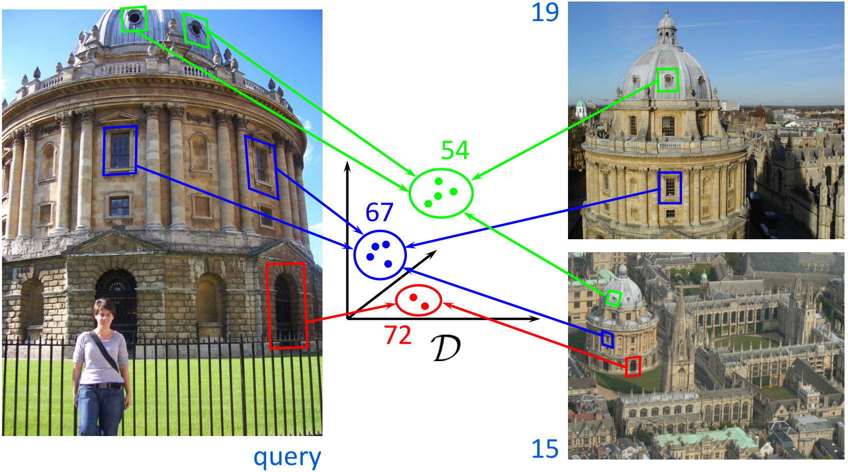
Vector quantization → visual words



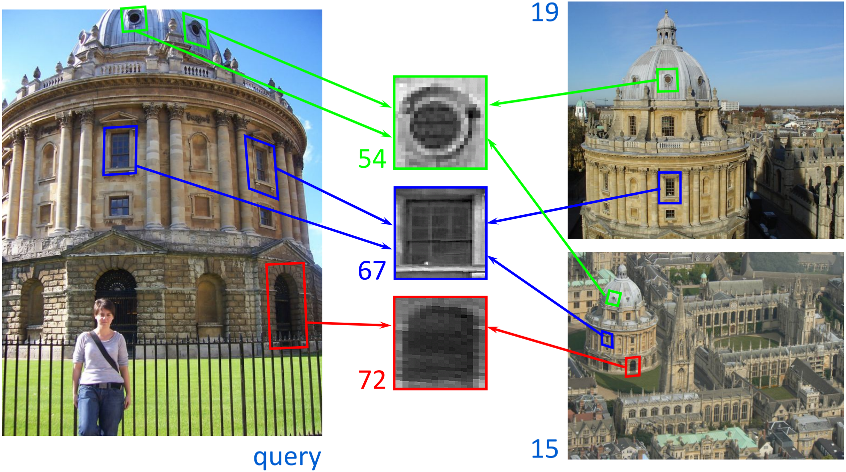
Vector quantization → visual words



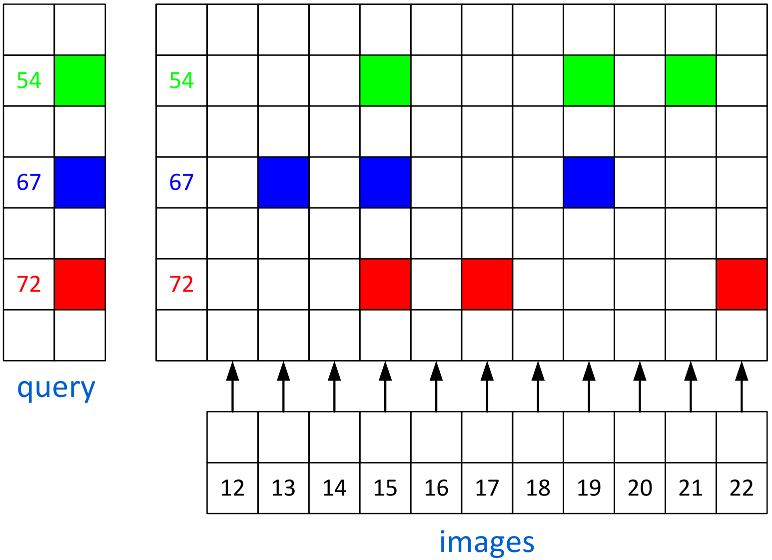
Vector quantization → visual words



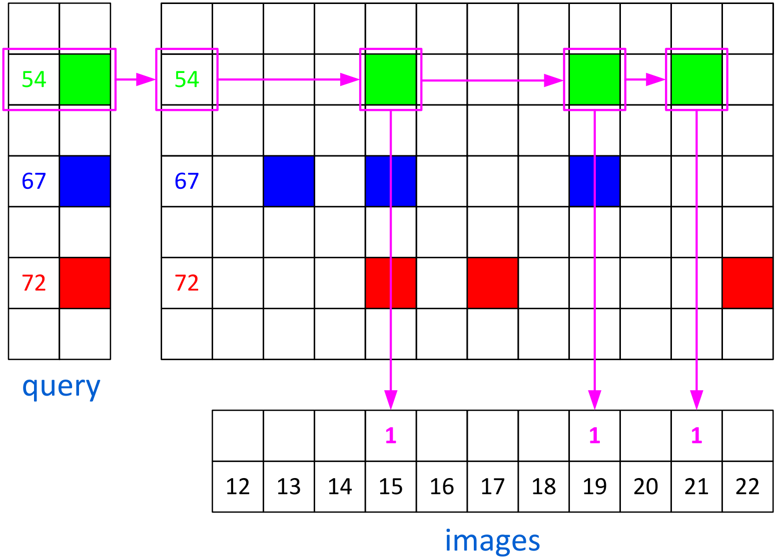
Vector quantization → visual words



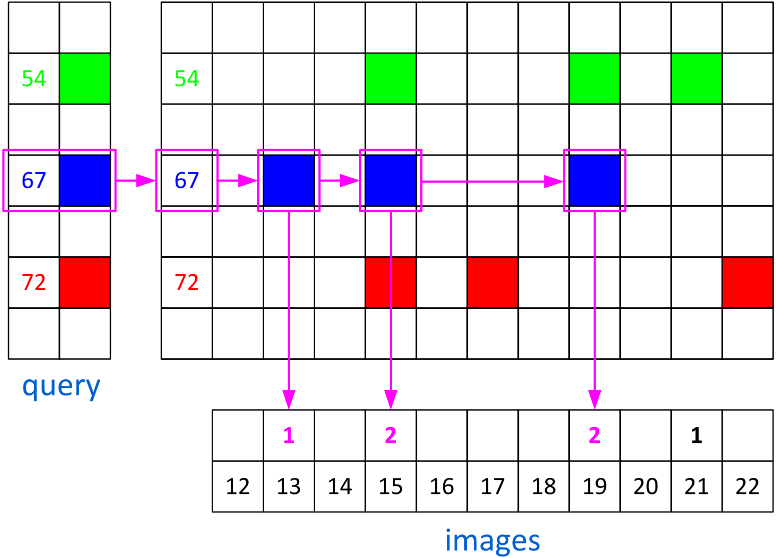
Inverted file indexing



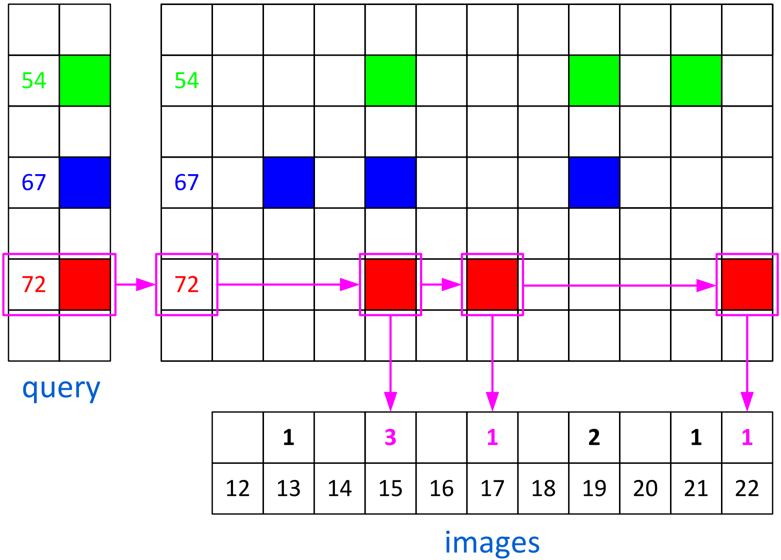
Inverted file indexing



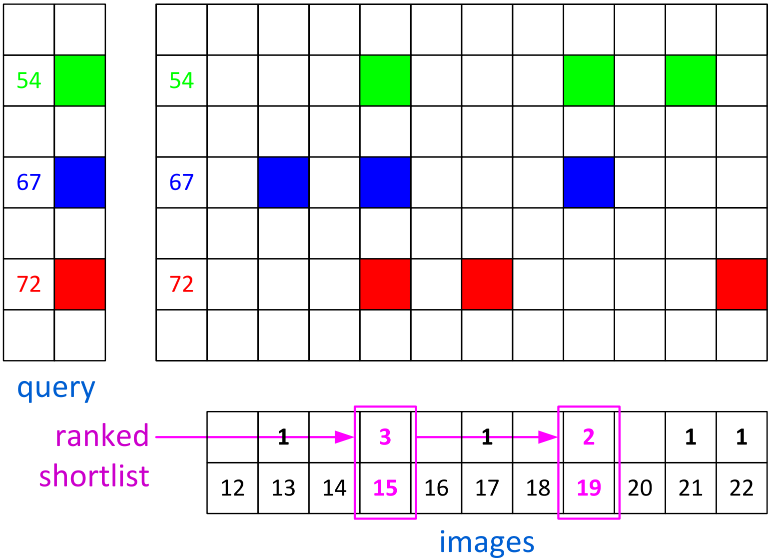
Inverted file indexing



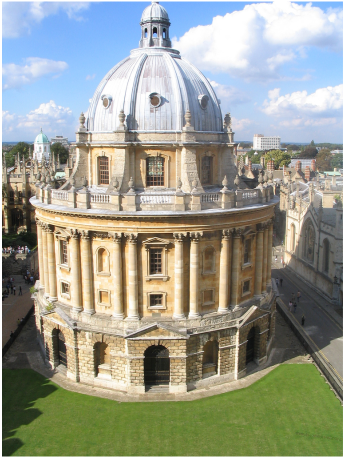
Inverted file indexing

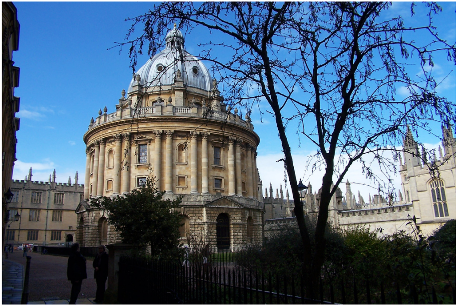


Inverted file indexing



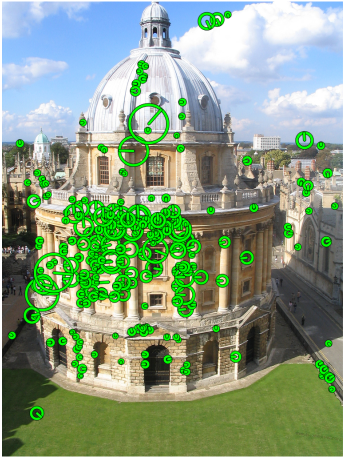
Back to geometry: re-ranking





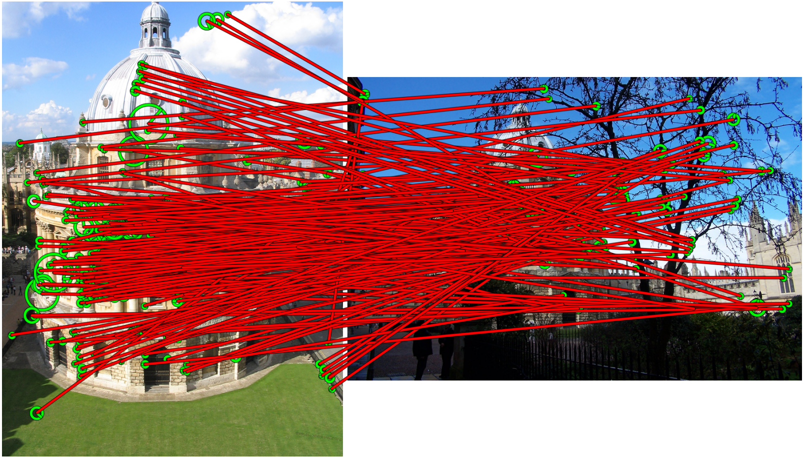
original images

Back to geometry: re-ranking



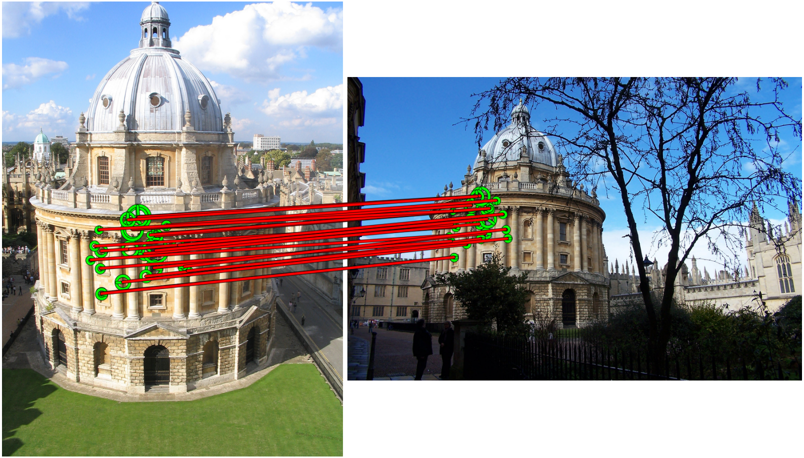

local features

Back to geometry: re-ranking



tentative correspondences

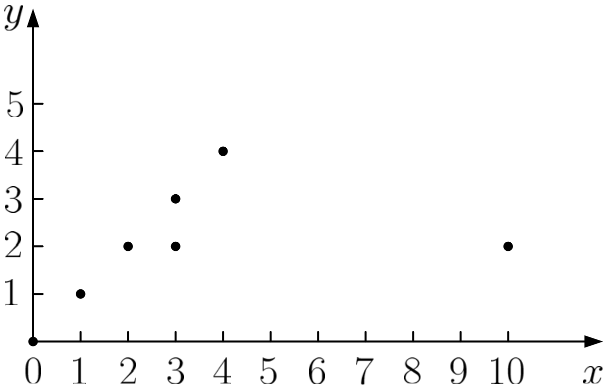
Back to geometry: re-ranking



RANSAC inliers

RANSAC

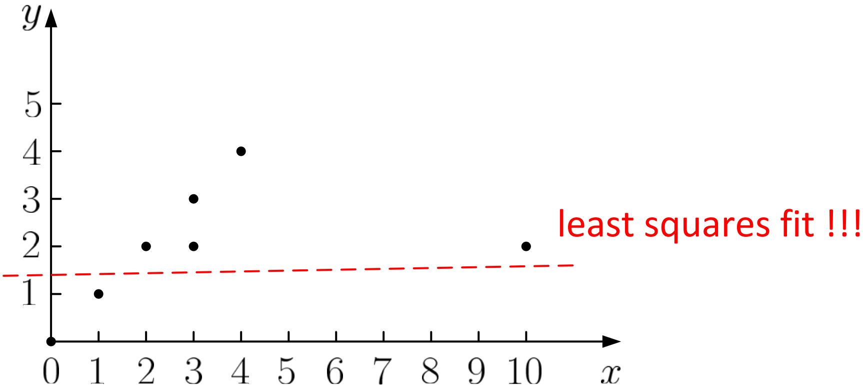
[Fischler and Bolles 1981]



problem: fit line to data

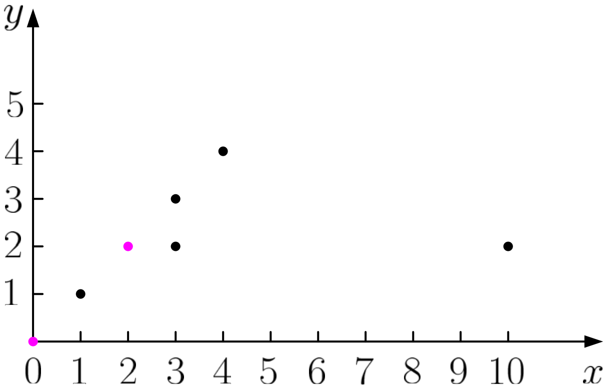
RANSAC

[Fischler and Bolles 1981]



RANSAC

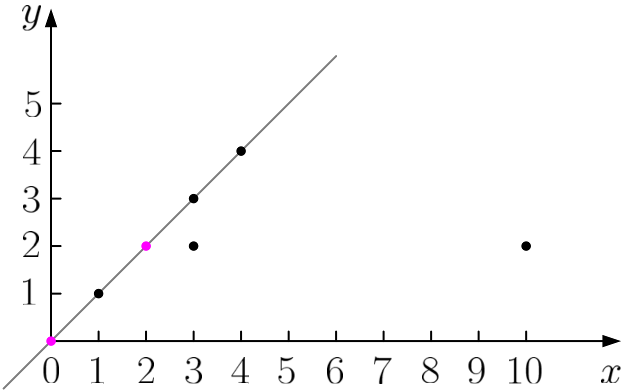
[Fischler and Bolles 1981]



solution: choose 2 random points ...

RANSAC

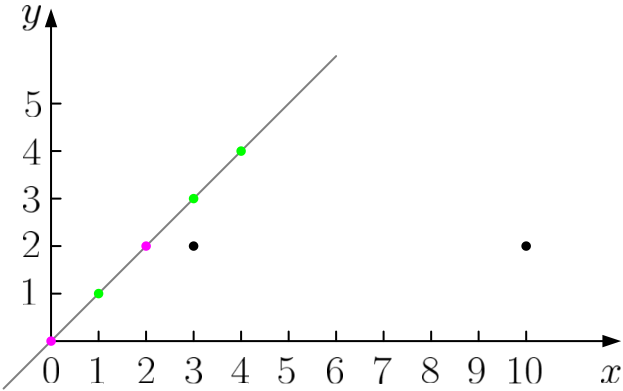
[Fischler and Bolles 1981]



... fit line to them ...

RANSAC

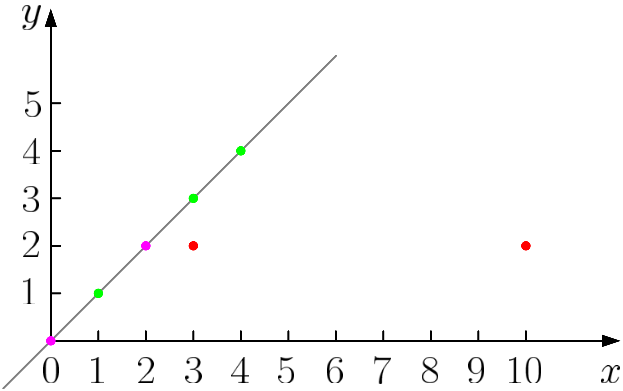
[Fischler and Bolles 1981]



... classify remaining points to inliers ...

RANSAC

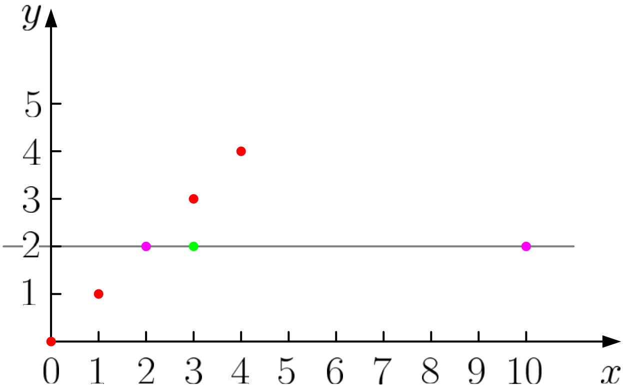
[Fischler and Bolles 1981]



... and outliers

RANSAC

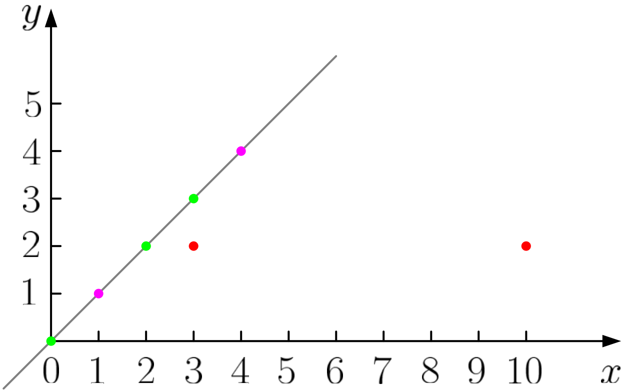
[Fischler and Bolles 1981]



repeat ...

RANSAC

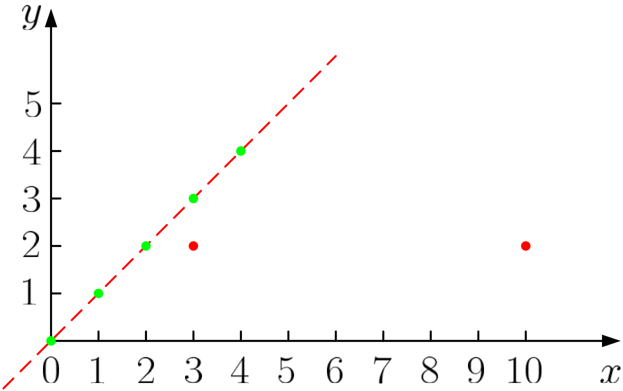
[Fischler and Bolles 1981]



... and repeat

RANSAC

[Fischler and Bolles 1981]



finally: maximum inliers

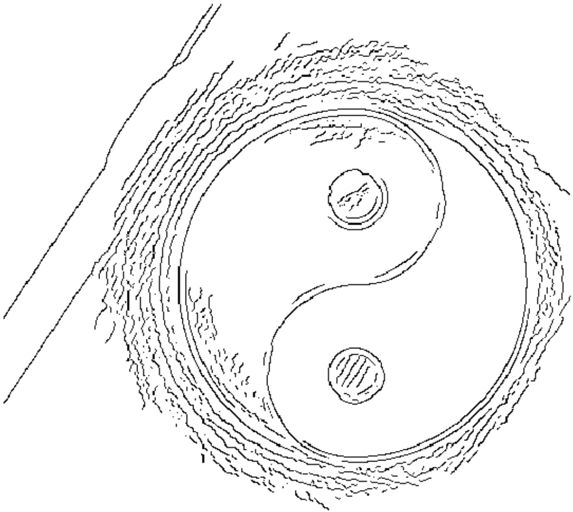
Outline

Edge-based feature detection
[Rapantzikos and Avrithis 2010]

- Blob-like regions starting from single-scale edges
- local maxima of Euclidean distance transform expected to lie in region interior or close to ridges
- greedily merge maxima guided by edge strength, to reproduce the effect of smoothing in scale-space evolution
- regions of arbitrary shape and scale, unaffected by spurious or disconnected edges

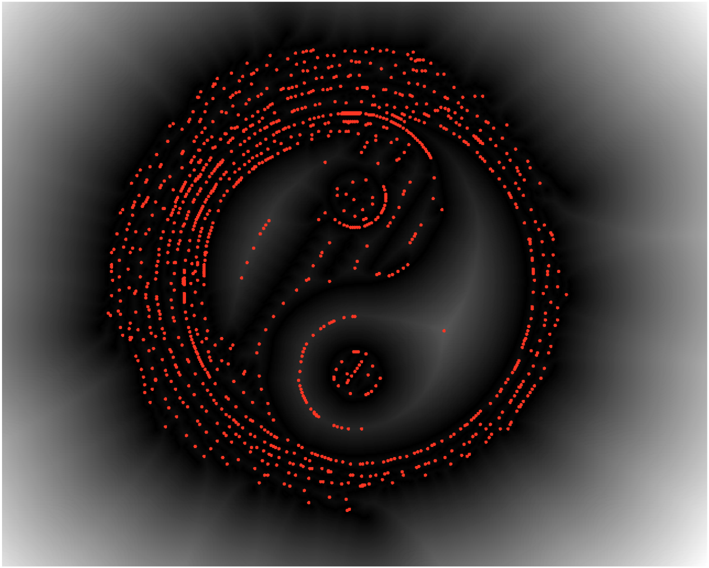
Original image

Binary edge map

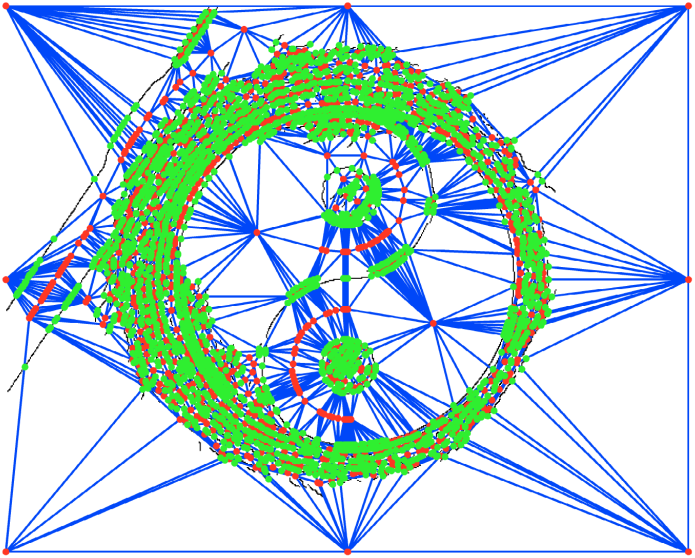


Binary distance map

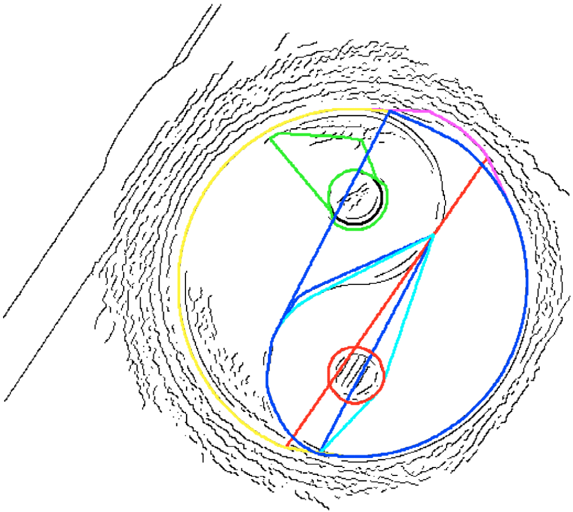
Distance map + local maxima



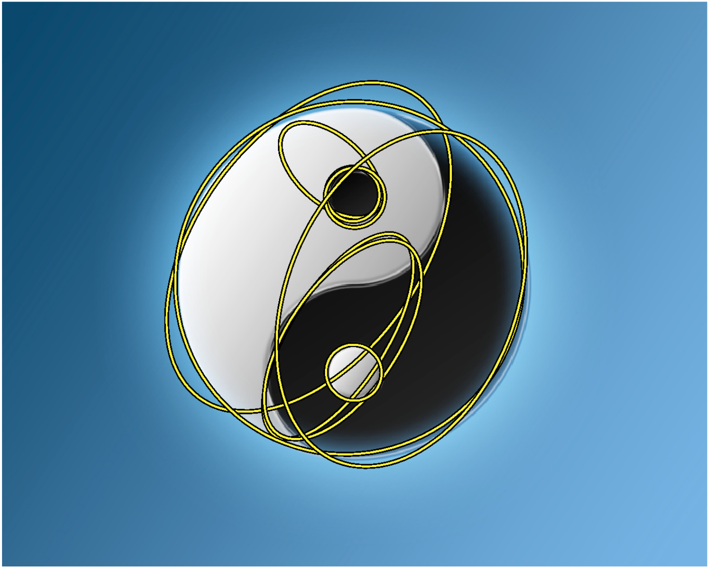
Delaunay triangulation



Convex hulls of selected regions



Original image + features

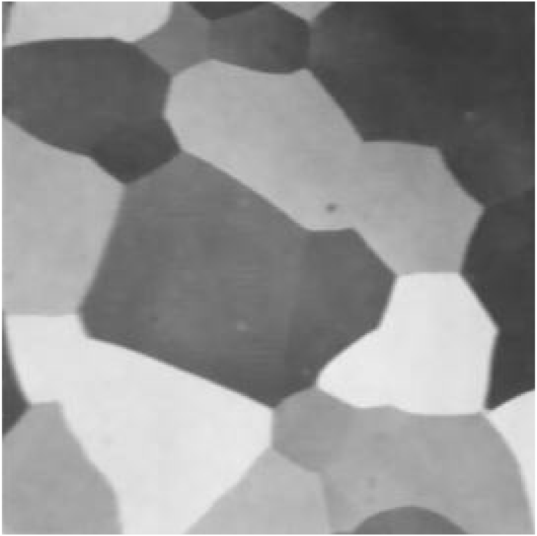


A weighted approach

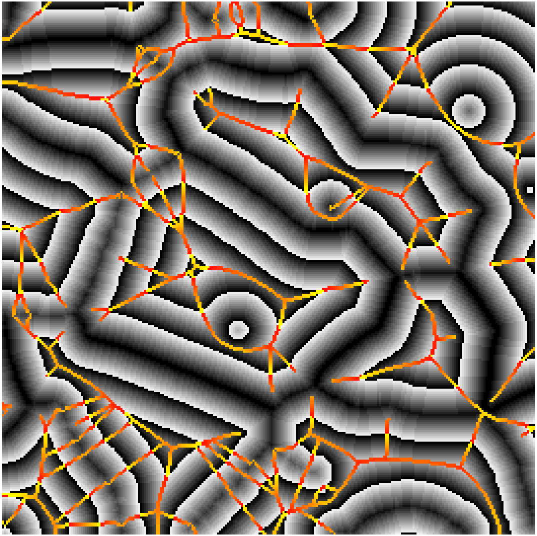
[Avrithis and Rapantzikos 2011, unpublished]

- Weighted distance map directly from image gradient
- Weighted medial capturing region structure and topology
- Very simple selection criterion: is a region well-enclosed by boundaries?
- Again, arbitrary shape and scale, without explicit scale-space construction
- Affordable speed—$1s$ for an 1Mpixel image, on average

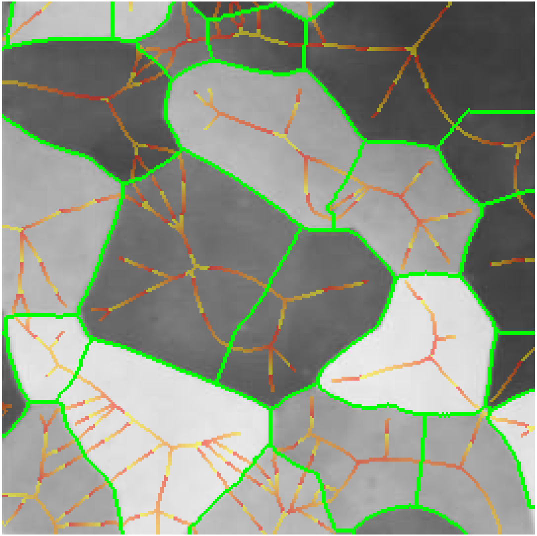
Original image



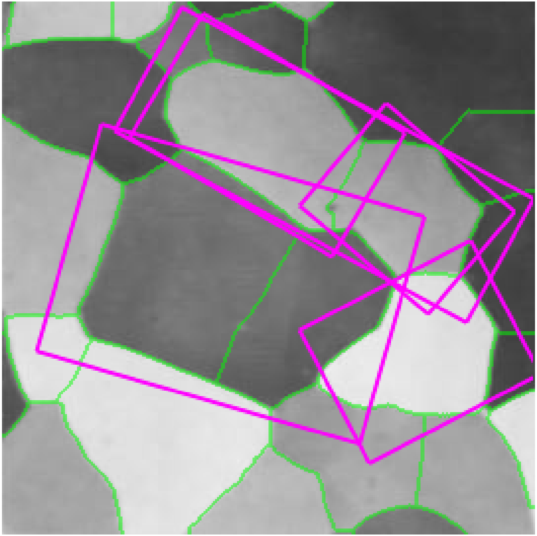
Weighted distance map and medial



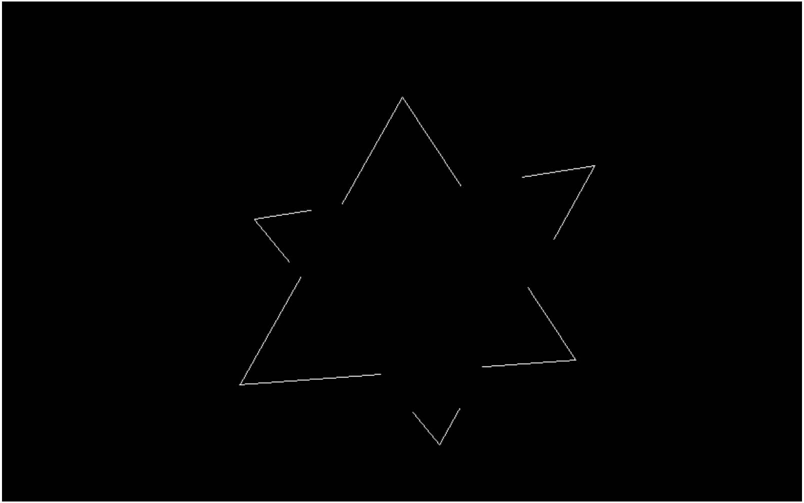
Region/boundary duality



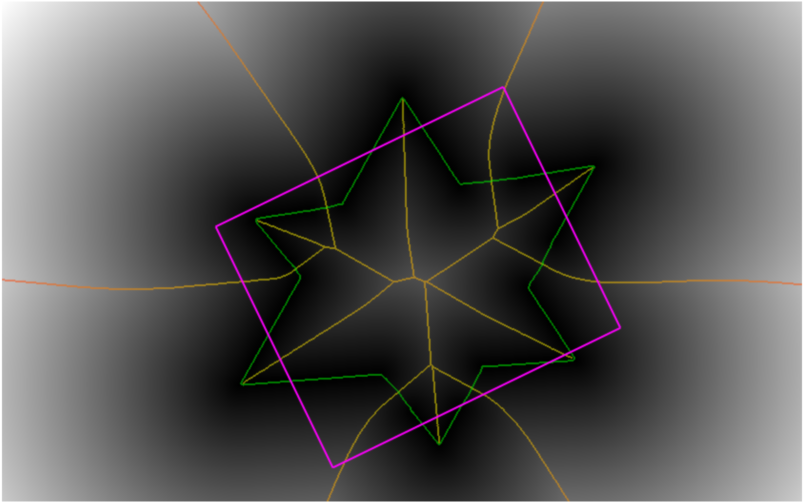
Original image + features



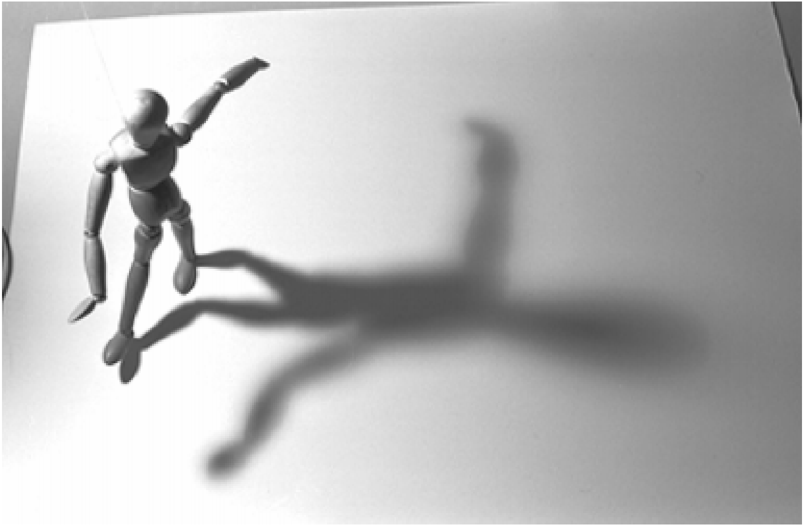
The challenge of shape



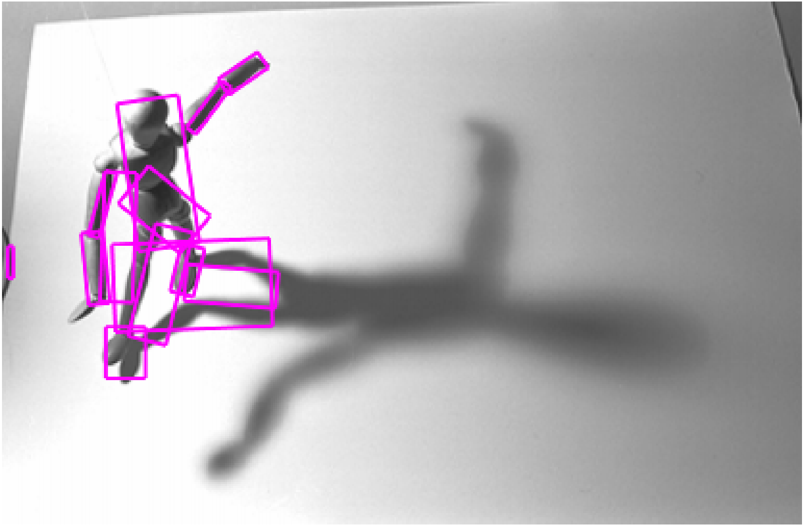
The challenge of shape



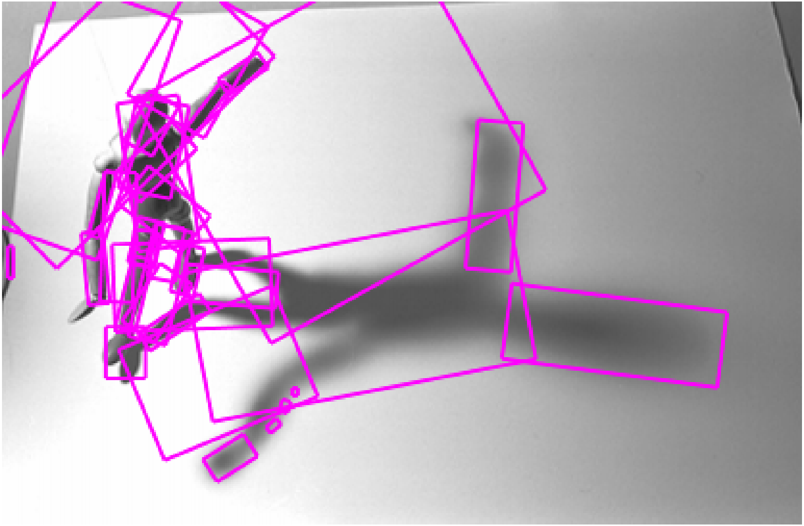
The challenge of scale



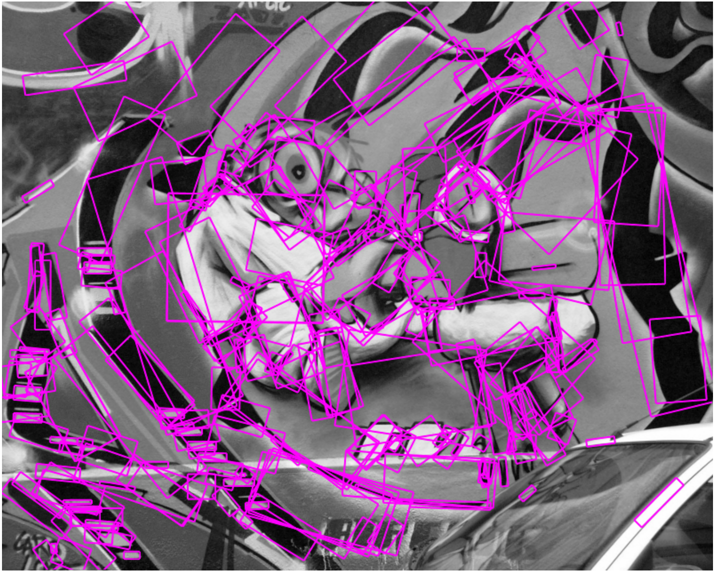
The challenge of scale



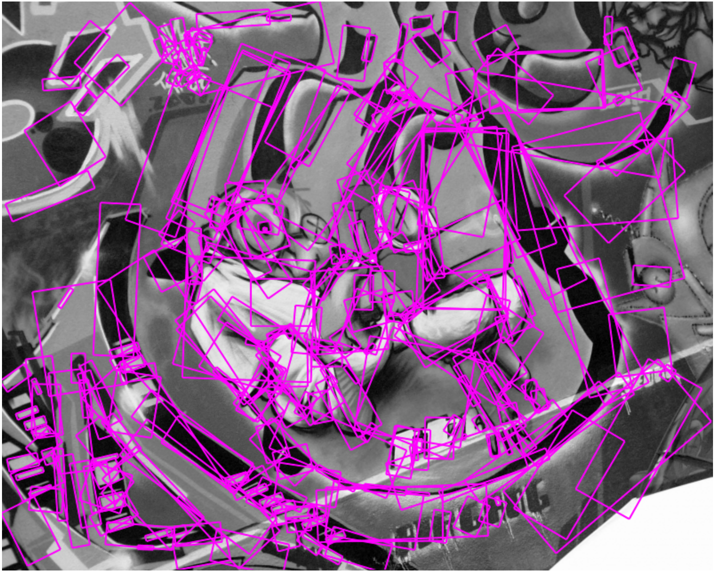
The challenge of scale



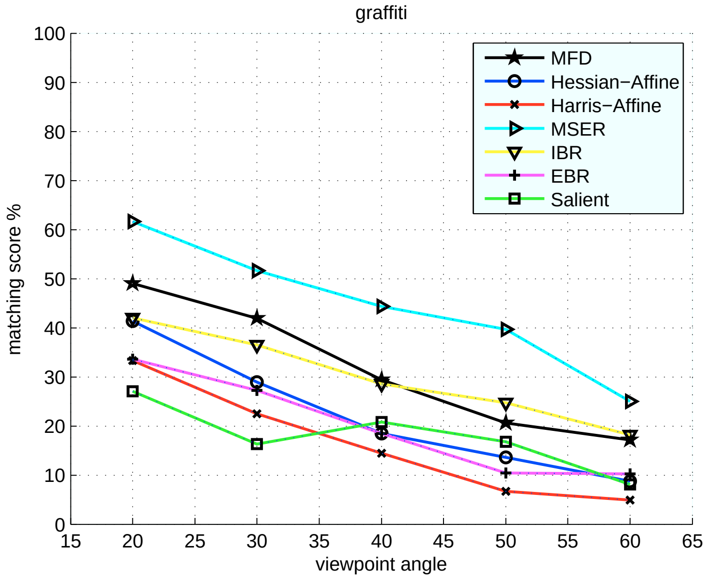
Viewpoint: `graffiti` scene



Viewpoint: `graffiti` scene



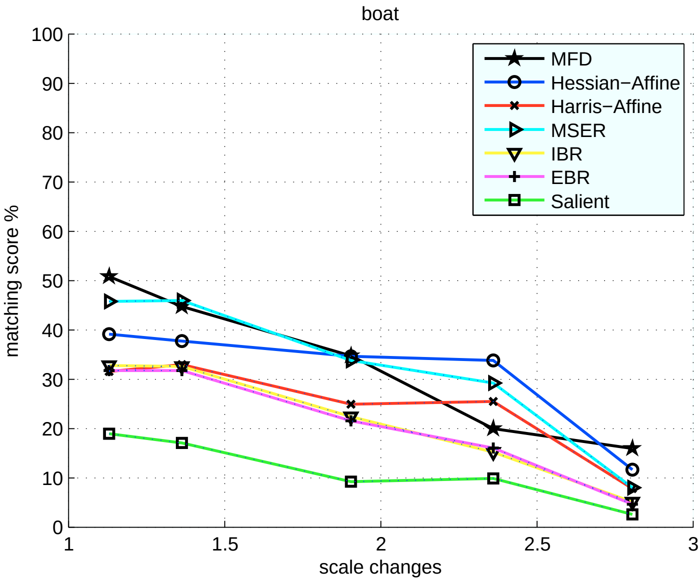
Viewpoint: `graffiti` scene



Scale + rotation: `boat` scene

Scale + rotation: `boat` scene

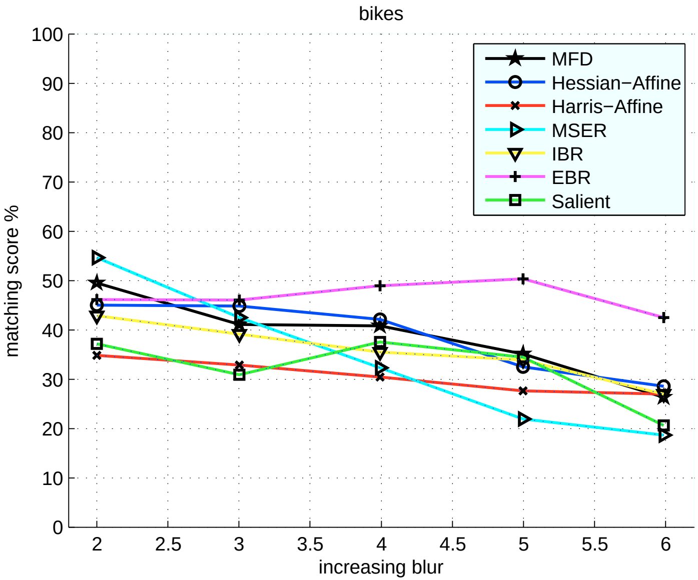
Scale + rotation: `boat` scene



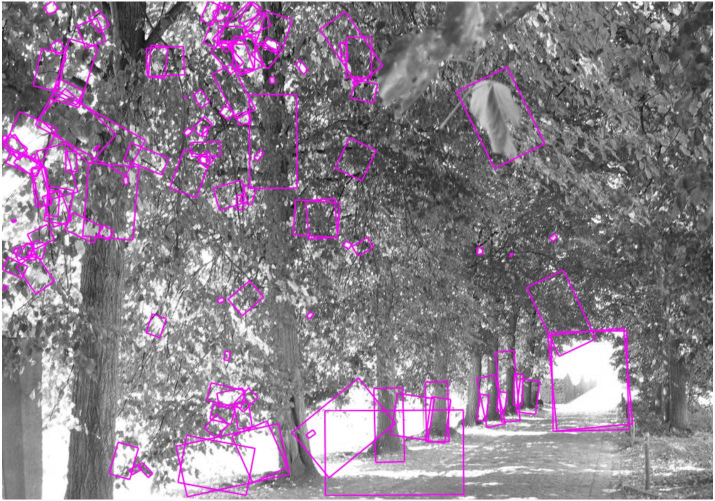
Blur: `bikes` scene

Blur: `bikes` scene

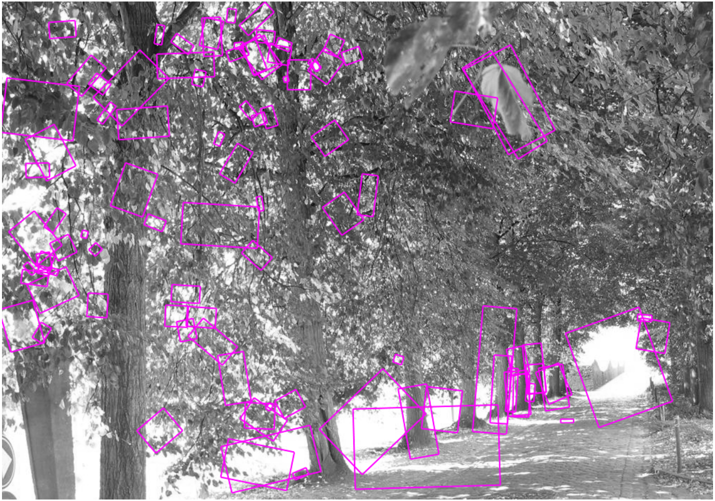
Blur: `bikes` scene



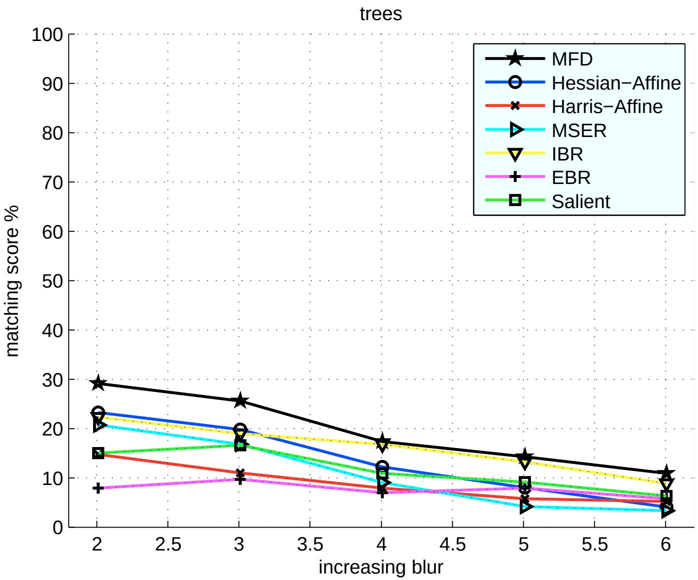
Texture + blur: `trees` scene



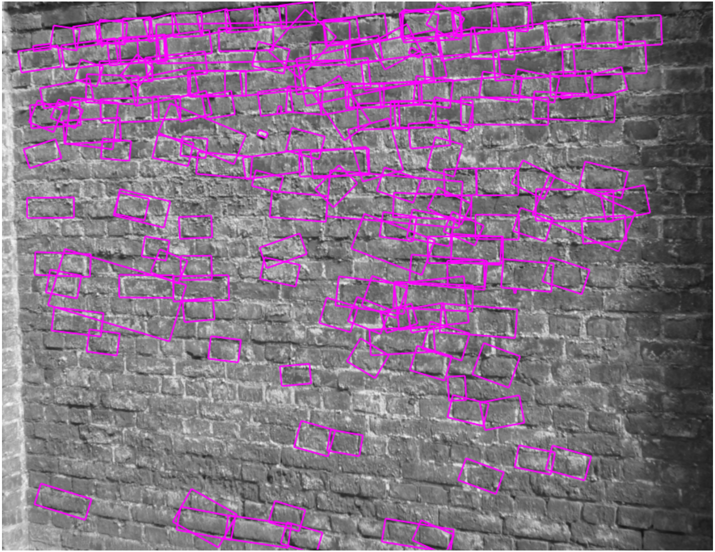
Texture + blur: `trees` scene



Texture + blur: `trees` scene

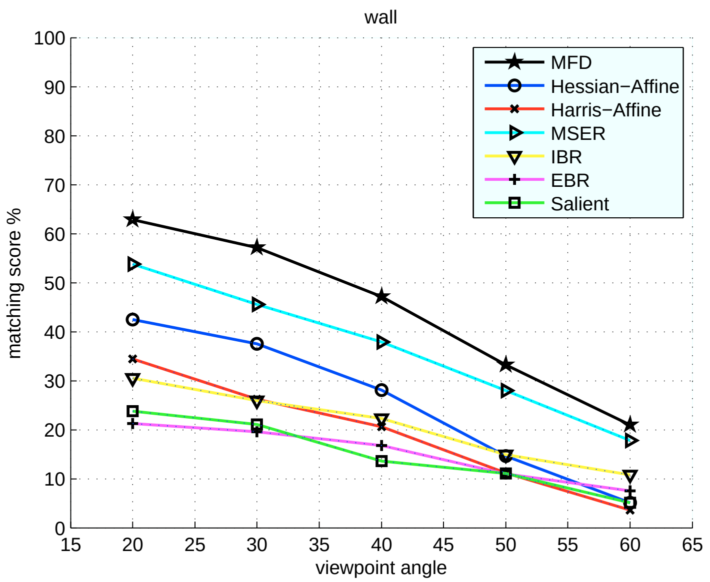


Viewpoint: `wall` scene



Viewpoint: `wall` **scene**

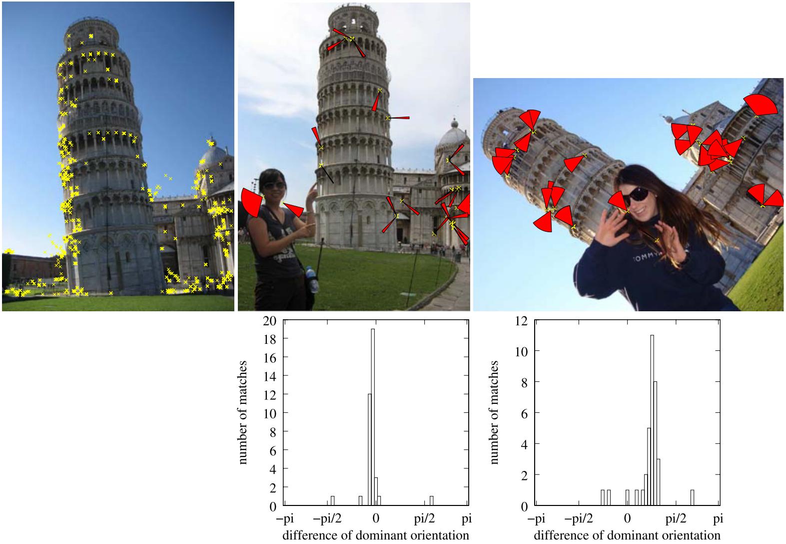
Viewpoint: `wall` scene



Outline

Weak geometric consistency (WGC)

[Jegou et al. 2008]



Weak geometric consistency

- when an image undergoes rotation or scaling, the orientation and scale of local features is consistently modified
- quantize orientation and scale differences between feature pairs
- maintain several scores for each image, one for each difference bin
- this is not enough to recover a full transformation, but does improve ranking

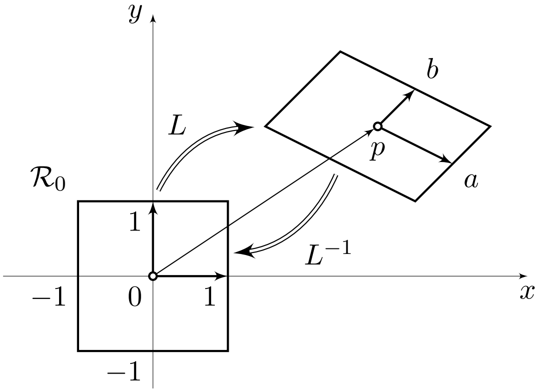
Feature map hashing

[Avrithis et al. 2010]

- estimate image alignment via single correspondence
- for each feature construct a feature map encoding normalized positions and appearance of all remaining features
- represent an image by a collection of such feature maps
- RANSAC-like matching is reduced to a number of set intersections

Local patches

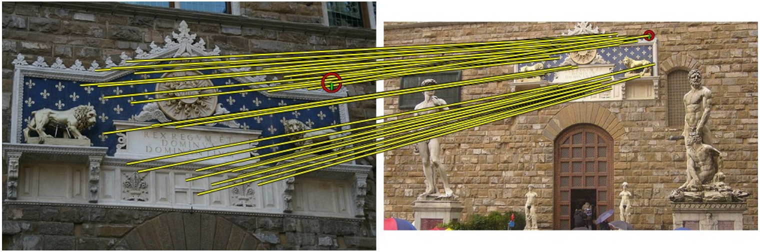
- each local feature is associated with an image patch L, which also represents an affine transform
- the rectified patch $\mathcal{R}_0$ is transformed to the patch via L
- the patch is rectified back to $\mathcal{R}_0$ via L^{-1}



Fast spatial matching (FSM)

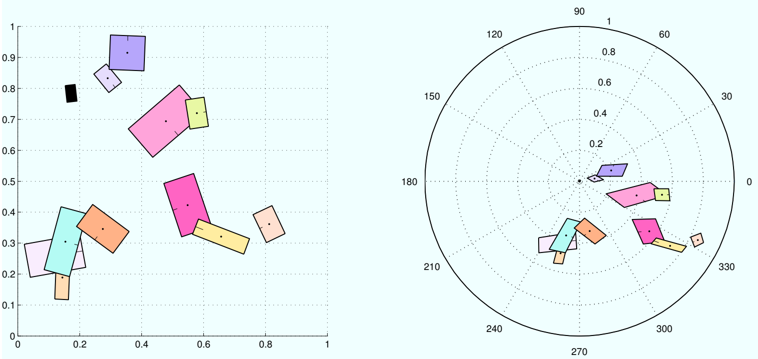
[Philbin et al. 2007]

- single 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 $O(n)$; we can compute them all



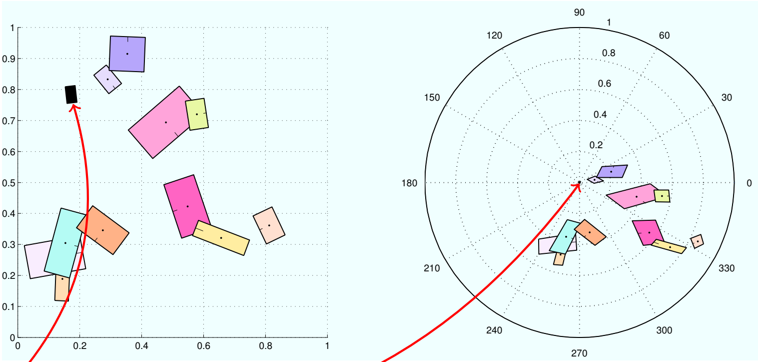
Feature set rectification

- rectify both feature sets by transformations L^{-1} and R^{-1}, then compare
- rectify the entire set of features in advance



Feature set rectification

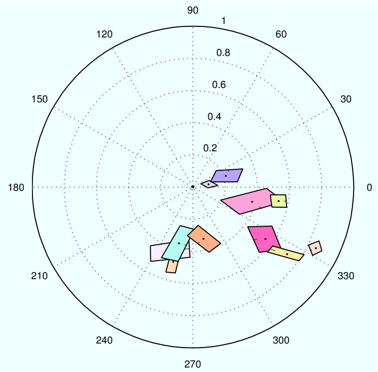
- rectify both feature sets by transformations L^{-1} and R^{-1}, then compare
- rectify the entire set of features in advance



origin

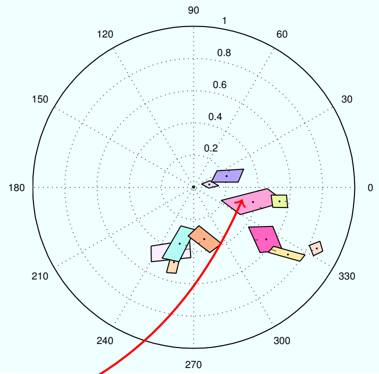
Spatial quantization

- 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

- represent an image 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

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

- 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

Feature maps

- represent an image 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

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

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

- 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

Feature maps

- represent an image 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

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

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

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

- 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

Feature maps

- represent an image 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

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

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

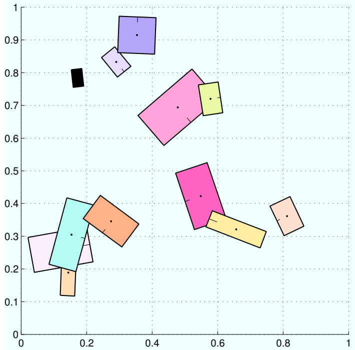
histogram wrt joint codebook $\mathcal{W}$

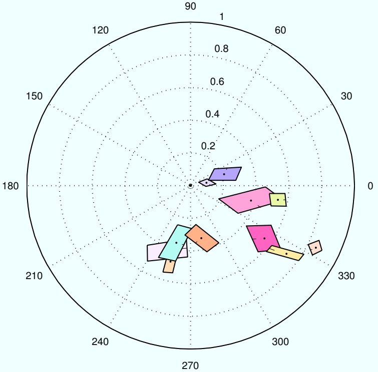
rectified feature set P wrt origin $\hat{x}$

- 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

Feature maps—example

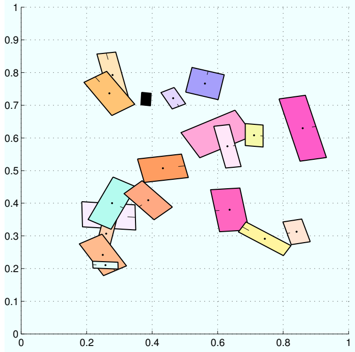
- well aligned feature sets are likely to have maps with a high degree of overlap

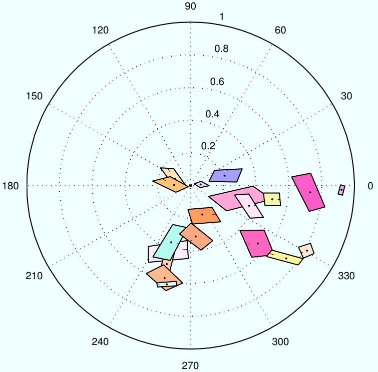




Feature maps—example

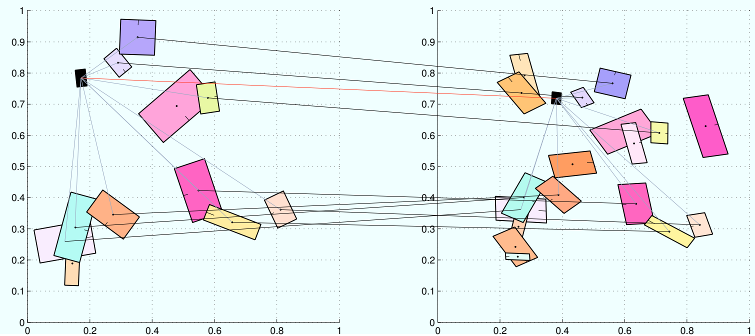
- well aligned feature sets are likely to have maps with a high degree of overlap





Feature map similarity

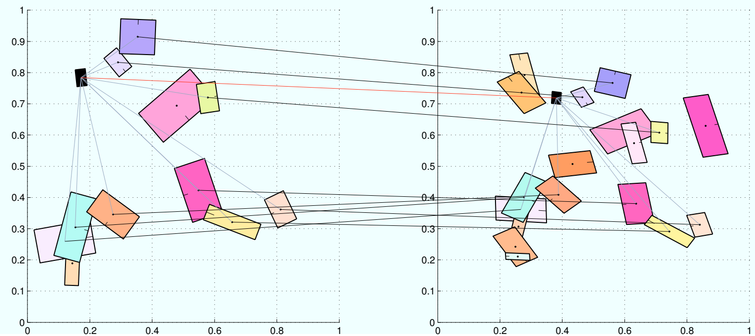
$$S_F(P,Q) = \max_{v \in V(P,Q)} \max_{\substack{\hat{x} \in H_v(P) \\ \hat{y} \in H_v(Q)}} f_P^{\mathrm{T}}(\hat{x}) \;\; f_Q(\hat{y})$$



Feature map similarity

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

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

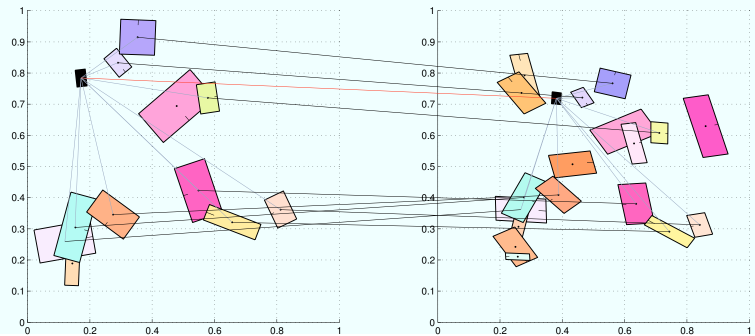


Feature map similarity

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

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

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



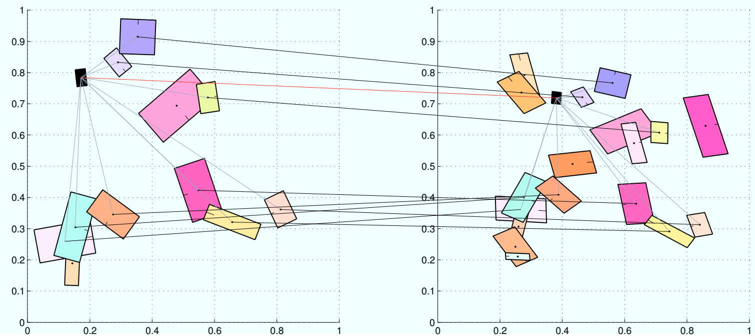
Feature map similarity

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

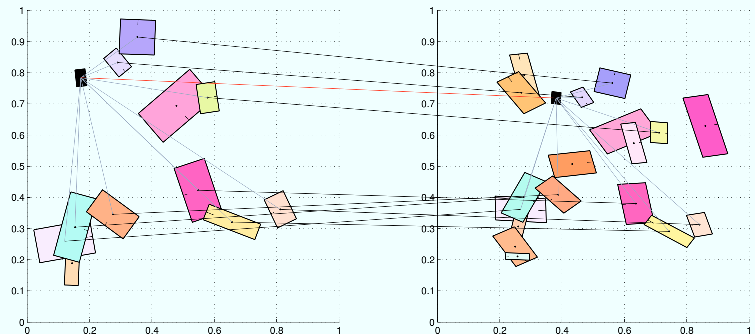
for all visual words that P, Q have in common

for all origins mapped to visual word v

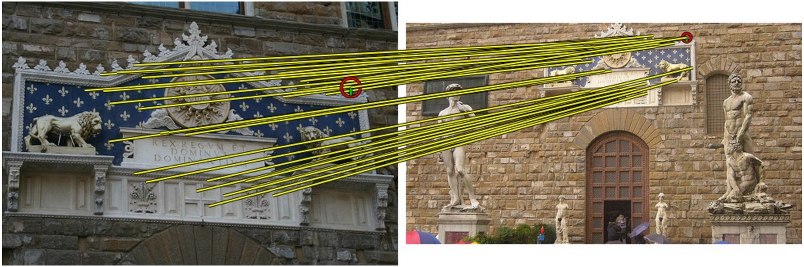
$$S_F(P,Q) = \max_{v \in V(P,Q)} \max_{\substack{\hat{x} \in H_v(P) \\ \hat{y} \in H_v(Q)}} f_P^{\mathrm{T}}(\hat{x}) \;\; f_Q(\hat{y})$$

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

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

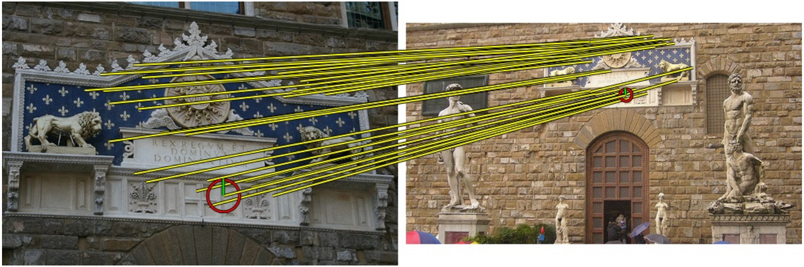


Feature map similarity—example



fast spatial matching [Philbin *et al.* 2007] (35 inliers)

Feature map similarity—example



feature map similarity (32 inliers)

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

a.k.a. min-hashing [Broder 2000]

- feature space $\mathbb{F} = \mathcal{P}(\mathcal{W})$, 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 $h(\bar{f}) = \min\{\pi(\bar{f})\}$
- two features maps are hashed to the same value with probability equal to their resemblance or Jaccard similarity coefficient

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

An example

[Chum et al. 2007]

a	b	c	d	e	f	$\{a, b, c\}$	$\{b, c, d\}$	$\{a, e, f\}$
permutations						hash values		
3	6	2	5	4	1	2	2	1
1	2	6	3	5	4	1	2	1
3	2	1	6	4	5	1	1	3
4	3	5	6	1	2	3	3	1

An example

[Chum et al. 2007]

a	b	c	d	e	f	$\{a,b,c\}$	$\{b,c,d\}$	$\{a,e,f\}$
permutations						hash values		
3	6	2	5	4	1	2	2	1
1	2	6	3	5	4	1	2	1
3	2	1	6	4	5	1	1	3
4	3	5	6	1	2	3	3	1

An example

[Chum et al. 2007]

a	b	c	d	e	f	$\{a, b, c\}$	$\{b, c, d\}$	$\{a, e, f\}$
permutations						hash values		
3	6	2	5	4	1	2	2	1
1	2	6	3	5	4	1	2	1
3	2	1	6	4	5	1	1	3
4	3	5	6	1	2	3	3	1

An example

[Chum et al. 2007]

a	b	c	d	e	f	$\{a,b,c\}$	$\{b,c,d\}$	$\{a,e,f\}$
permutations						hash values		
3	6	2	5	4	1	2	2	1
1	2	6	3	5	4	1	2	1
3	2	1	6	4	5	1	1	3
4	3	5	6	1	2	3	3	1

An example

[Chum et al. 2007]

a	b	c	d	e	f	$\{a, b, c\}$	$\{b, c, d\}$	$\{a, e, f\}$
permutations						hash values		
3	6	2	5	4	1	2	2	1
1	2	6	3	5	4	1	2	1
3	2	1	6	4	5	1	1	3
4	3	5	6	1	2	3	3	1

Map sketch

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

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

- 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

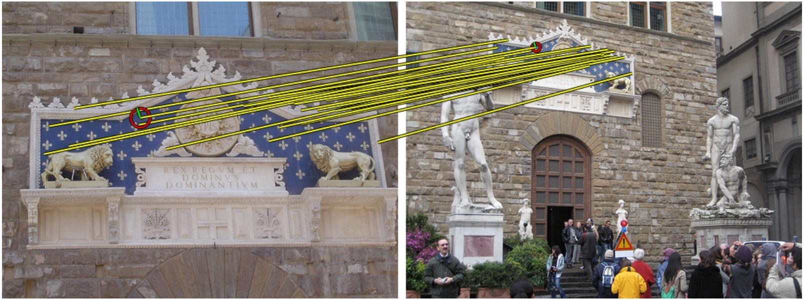
$$S_M(\mathbf{F}, \mathbf{G}) = \max_{\mathbf{f} \in \mathbf{F}} \max_{\mathbf{g} \in \mathbf{G}} s_K(\mathbf{f}, \mathbf{g})$$

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

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

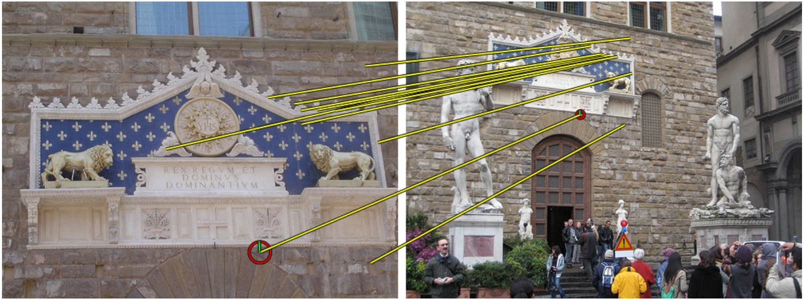
- constrain sketch origins to those mapping uniquely to the same visual words

Matching maps



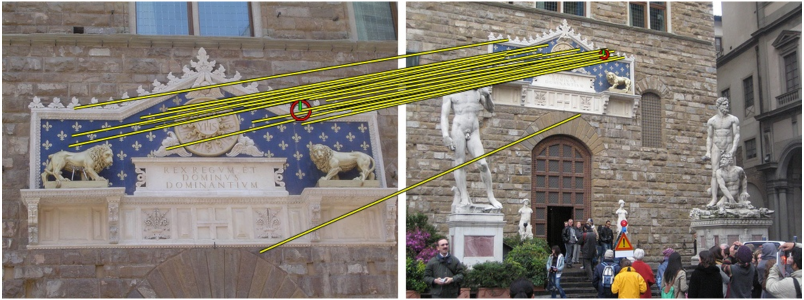
Multiple matching pairs of feature maps

Matching maps



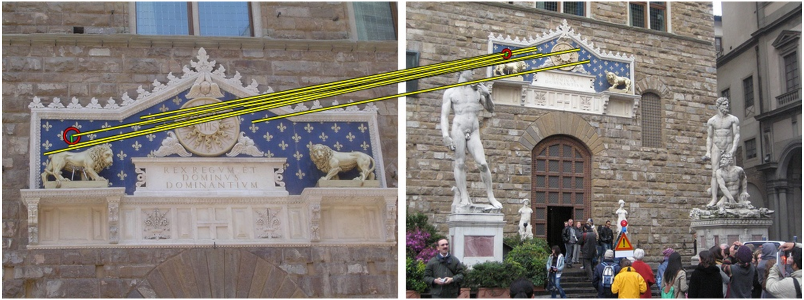
Multiple matching pairs of feature maps

Matching maps



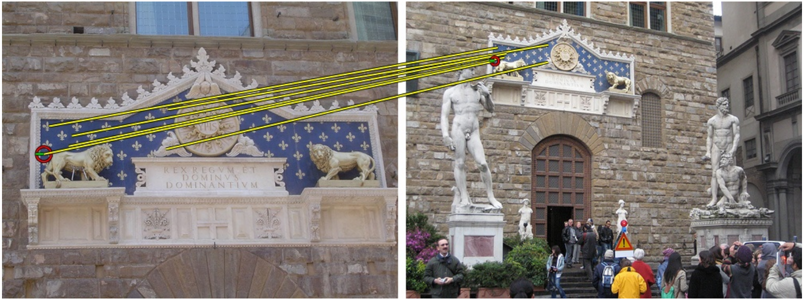
Multiple matching pairs of feature maps

Matching maps



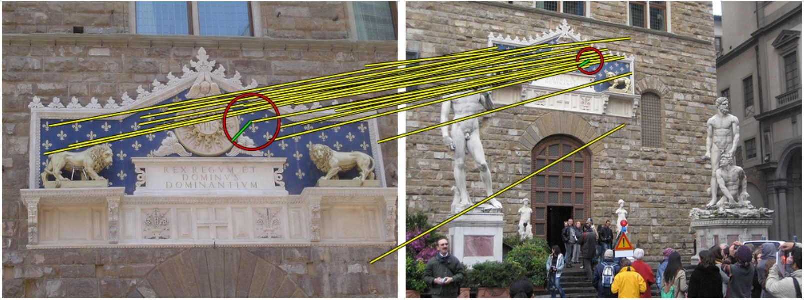
Multiple matching pairs of feature maps

Matching maps



Multiple matching pairs of feature maps

Matching maps



Multiple matching pairs of feature maps

Indexing

index construction

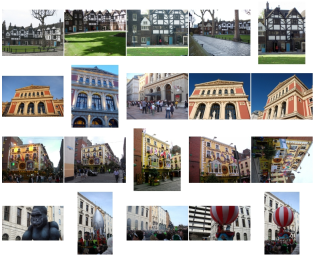
- construct inverted file of triplets $(\hat{v}, w, \pi)$ (origin, hash value, permutation)
- memory requirements $5\times$ a typical baseline system

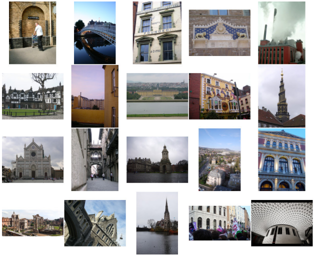
query

- retrieve images by triplets $(\hat{v}, w, \pi)$ of query image
- re-estimate transformation parameters using LO-RANSAC
- re-ranking is an order of magnitude faster than FastSM, because an initial estimate is already available

European Cities dataset 50K (EC50K)

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

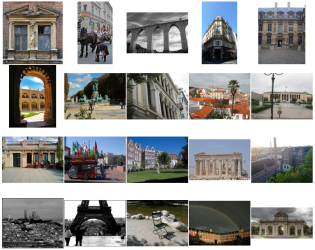


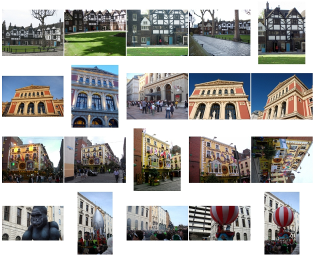


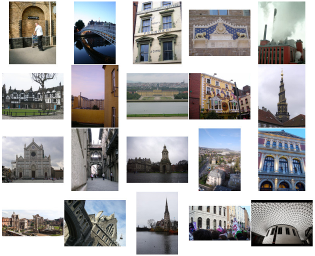
Publicly available: `http://image.ntua.gr/iva/datasets/ec50k`

European Cities dataset 50K (EC50K)

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

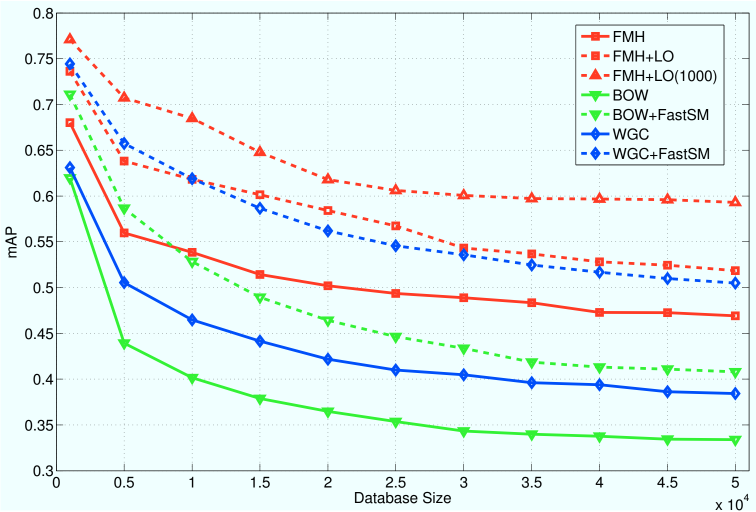






Publicly available: `http://image.ntua.gr/iva/datasets/ec50k`

Results EC50K



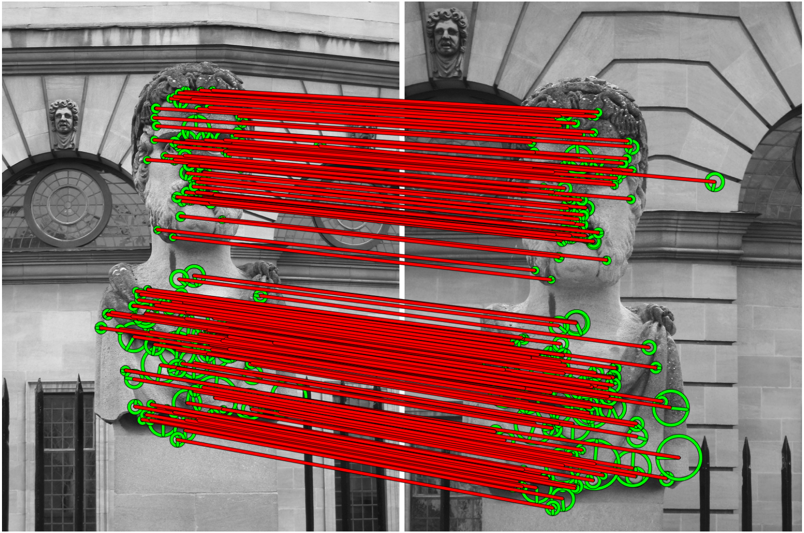
Outline

Relaxed spatial matching

[Tolias and Avrithis 2011, unpublished]

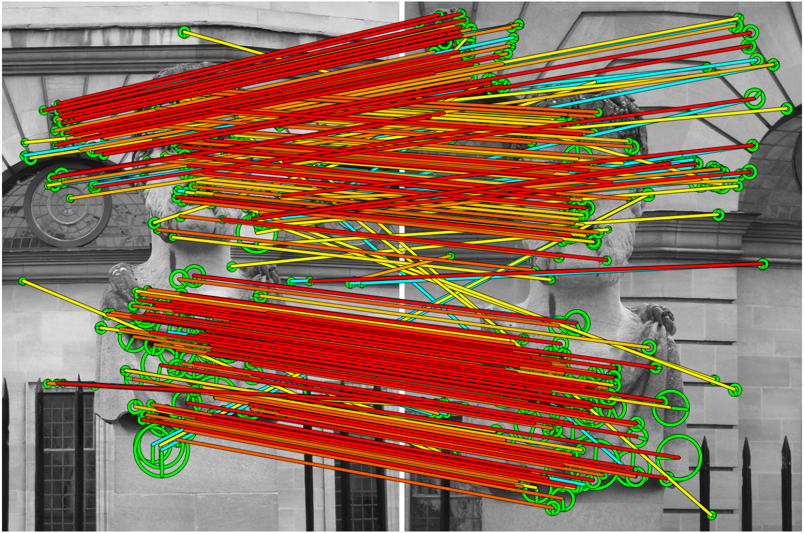
- invariant to similarity transformations
- flexible, allowing non-rigid motion and multiple matching surfaces or objects
- imposes one-to-one mapping
- non-iterative, and linear in the number of correspondences
- in a given query time, can re-rank one order of magnitude more images than the state of the art
- needs less than one millisecond to match a pair of images, on average

Relaxed matching—examples



fast spatial matching

Relaxed matching—examples



relaxed matching

Relaxed matching—examples

fast spatial matching

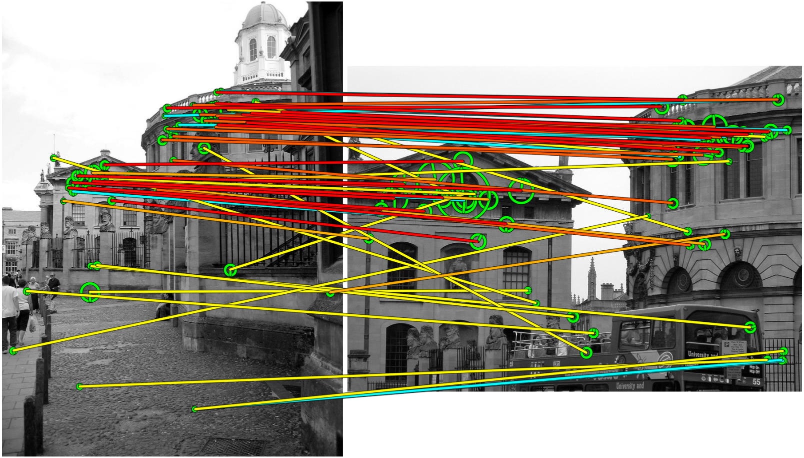
Relaxed matching—examples

relaxed matching

Relaxed matching—examples

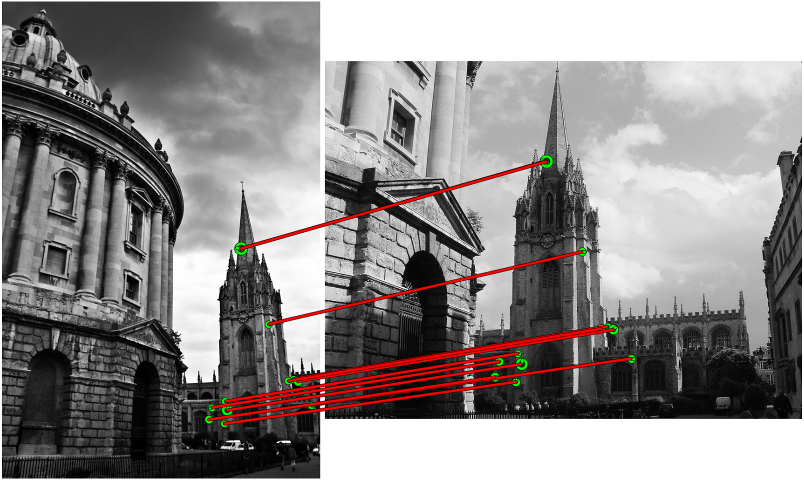
fast spatial matching

Relaxed matching—examples



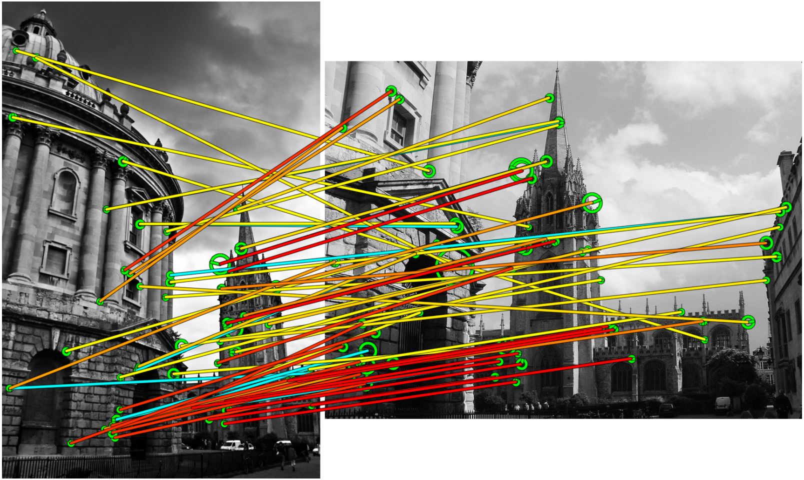
relaxed matching

Relaxed matching—examples



fast spatial matching

Relaxed matching—examples

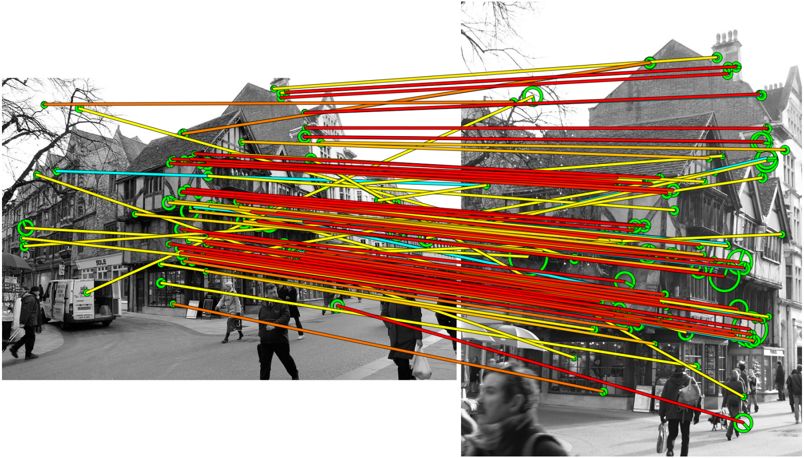


relaxed matching

Relaxed matching—examples

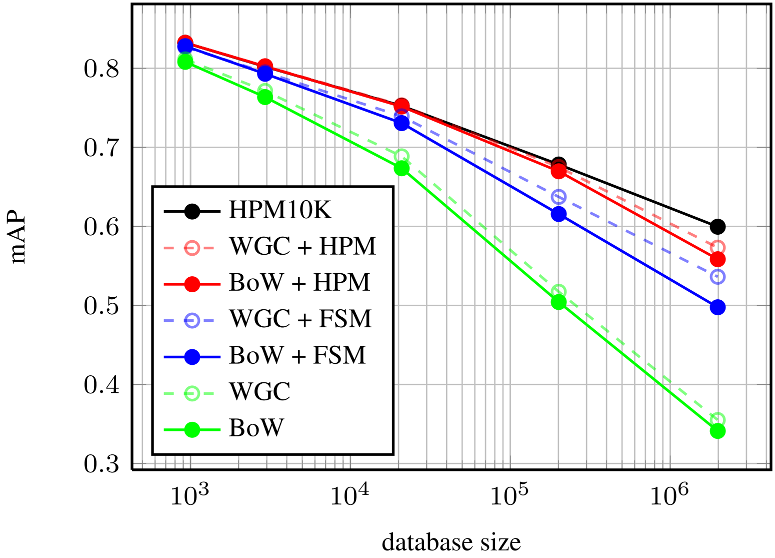
fast spatial matching

Relaxed matching—examples

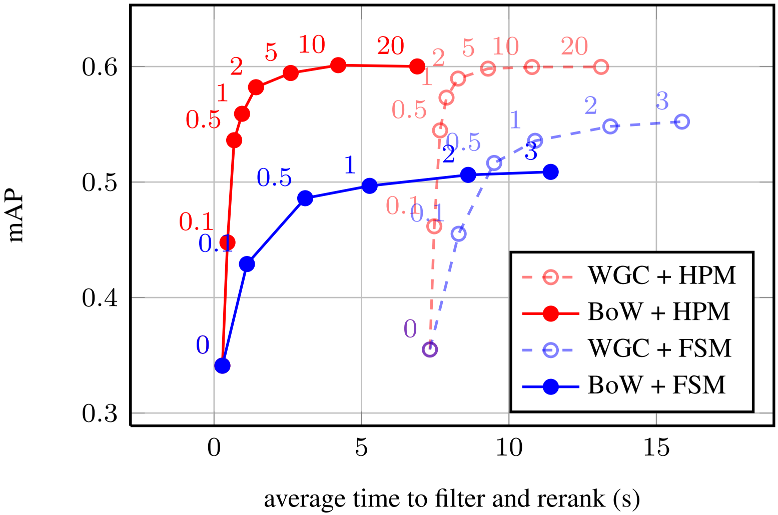


relaxed matching

Relaxed matching—statistics



Relaxed matching—timing



Outline

1. Visual search, local features and bag-of-words
2. Local features based on distance maps
3. Geometry indexing: feature map hashing
4. Relaxed spatial matching and re-ranking
5. **Photo collections: view clustering and scene maps**
6. Location and landmark recognition
7. Implementation: `ivl` library

Community photo collections

clustering / landmark recognition

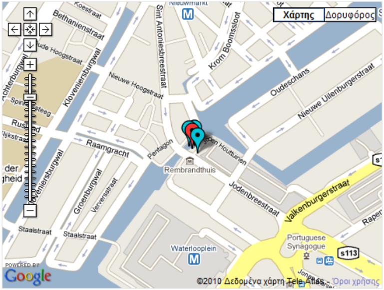
- focus on popular subsets
- applications: browsing, 3D reconstruction

[Crandall et al. 2009]

Community photo collections

retrieval / location recognition

- include all images, has not yet scaled enough
- applications: automatic geo-tagging, camera pose estimation

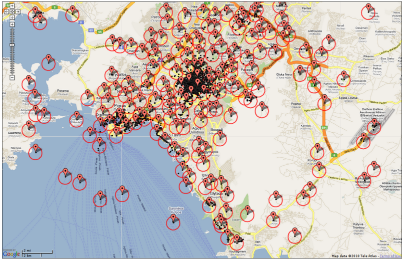


Estimated Location, Similar Image, Incorrectly geo-tagged, Unavailable

Suggested tags: Sint Antoniesbreestraat, Zwanenburgwal, Amsterdam
Frequent user tags: Anthoniesluis, sluijswacht, krom, stare, Skirt

View clustering
[Avrithis et al. 2010]

- identify images that potentially depict views of the same scene
- geo clustering: according to location
- visual clustering: according to visual similarity



- use sub-linear indexing in the clustering process

Kernel vector quantization (KVQ)

[Tipping and Schölkopf 2001]

- input dataset: $D \subseteq X$, where (X, d) is a metric space
- codebook: a small subset $Q(D)$ such that distortion is minimized
- for codebook vector $x \in Q(D)$, cluster $C(x)$ contains all points $y \in D$ within distance r:

$$C(x) = \{y \in D : d(x, y) < r\}$$

- sparse solution by solving a linear programming problem
- pairwise distance matrix: quadratic in the dataset size $|D|$

Kernel vector quantization

properties

- codebook vectors are points of the original dataset: $Q(D) \subseteq D$
- distortion upper bounded by r: for all $x \in Q(D)$

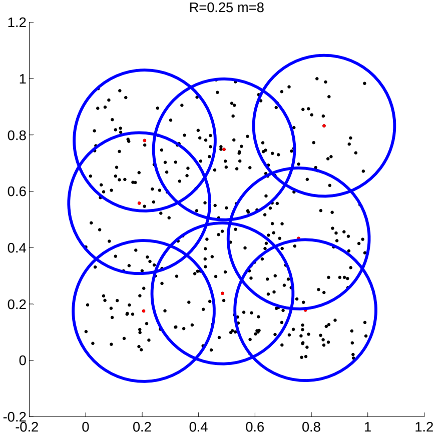
$$\max_{y \in C(x)} d(x, y) < r$$

- the cluster collection

$$\mathcal{C}(D) = \{C(x) : x \in Q(D)\}$$

is a cover for D

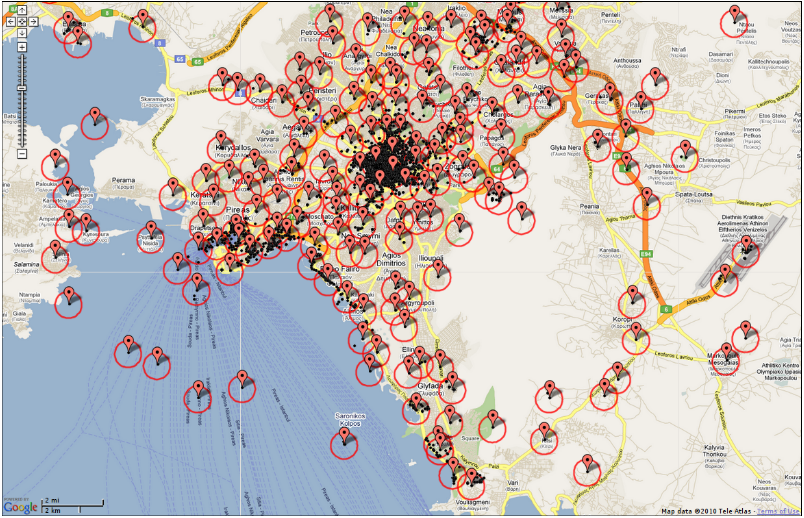
- clusters are overlapping



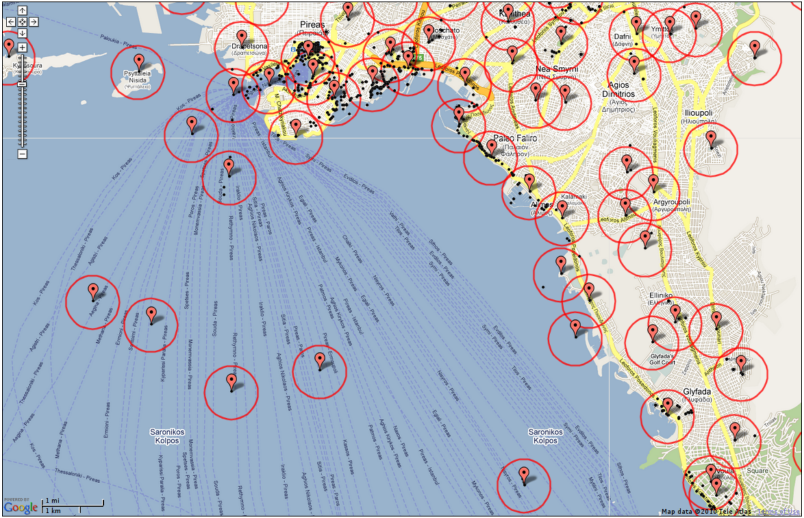
Geo clustering

- given set of photos $P \subseteq \mathcal{P}$ in metric space $(\mathcal{P}, d_g)$
- each photo $p \in P$ is represented by tuple (ℓ_p, F_p) (location, features)
- metric d_g: the great circle distance
- construct codebook $Q_g(P)$ by KVQ of P with scale parameter r_g
- maximum distortion: photos taken *e.g.* further than $2km$ apart are not likely to depict the same scene

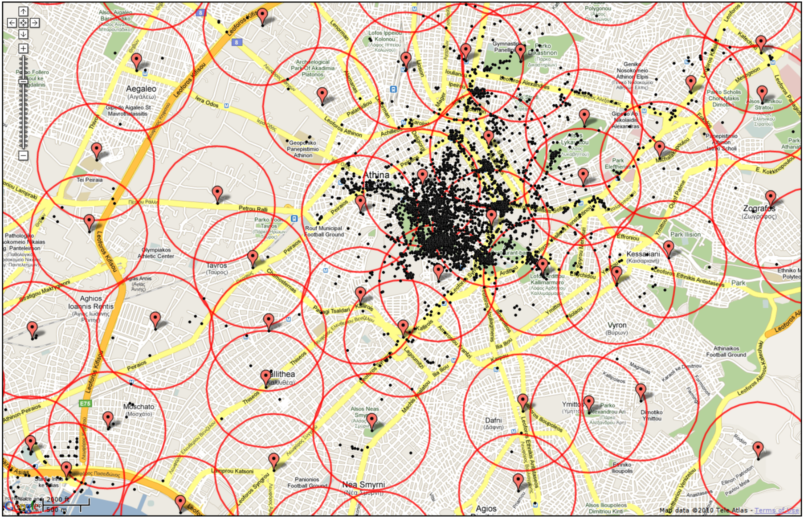
Geo clustering—example



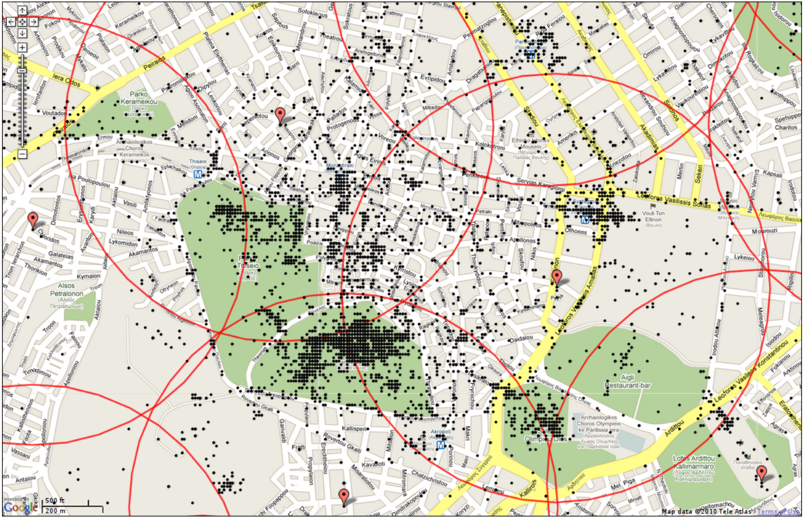
Geo clustering—example



Geo clustering—example



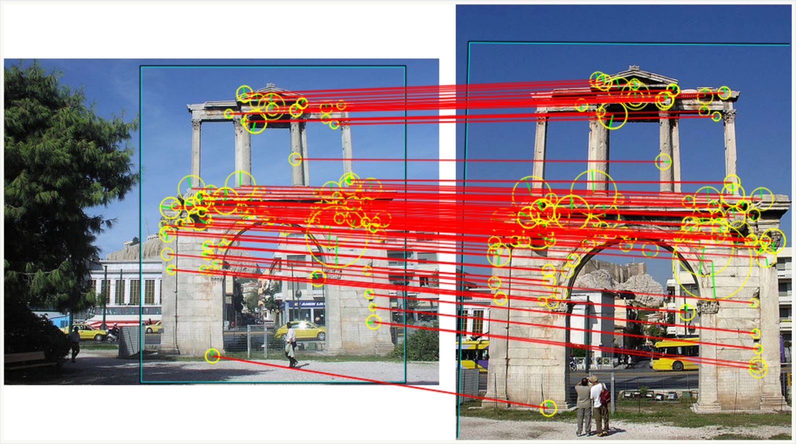
Geo clustering—example



Visual clustering

visual similarity measure

- $I(F_p, F_q)$: number of inliers between visual feature sets F_p, F_q of photos p, q respectively



Visual clustering

- for each geo-cluster G, construct codebook $Q_v(G)$ by KVQ in space $(\mathcal{P}, d_v)$ with scale parameter r_v
- metric $d_v(p, q)$ and scale parameter r_v are expressed in terms of number of inliers
- maximum distortion: equivalent to minimum number of inliers
- overlapping: support gradual transitions of views

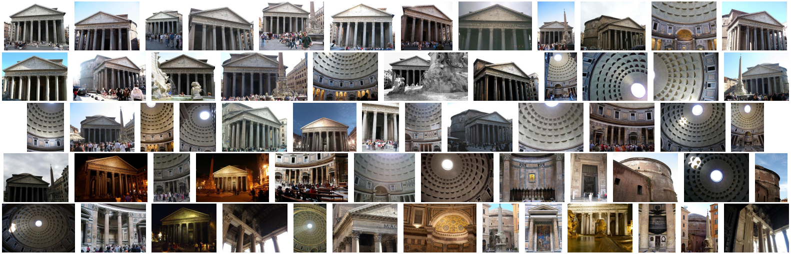
Visual clustering

- geo-cluster specific sub-linear indexing
- bottleneck: computation of pairwise distances, quadratic in $|G| \to$ inverted file indexed by both visual word and geo-cluster
- given a query image $q \in G$, find all matching images $p \in G$ with $I(F_p, F_q) > \tau$ in constant time, typically less than one second
- the entire computation is now linear in $|G|$

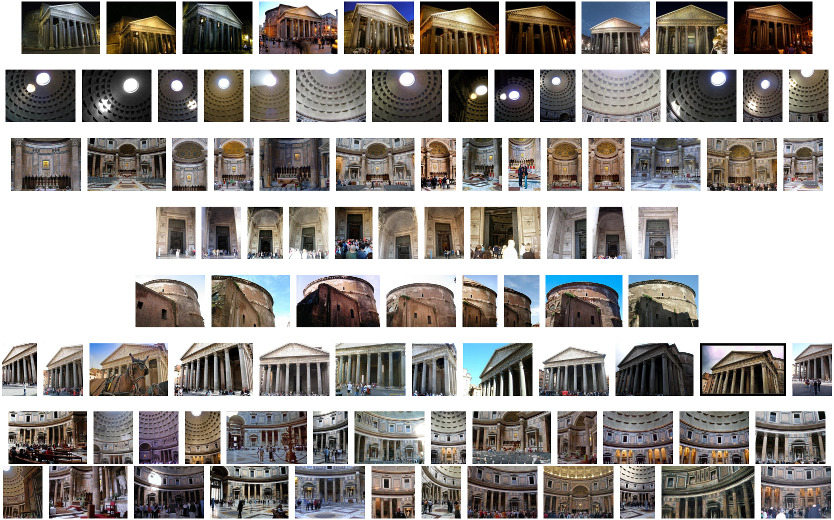
Visual clustering—example

$1,146$ geo-tagged Flickr images of Pantheon, Rome

- 258 resulting visual clusters
- 30 images at each visual cluster on average
- an image belongs to 4 visual clusters on average



Visual clustering—example



Scene maps

[Avrithis et al. 2010]

- the image associated to the center of a view cluster shares at least one rigid object with all other images in the cluster
- treat this image as a reference for the cluster and align all other images to it
- initial estimates available from the view clustering stage—only local optimization needed
- construct a 2D scene map by grouping similar local features
- extend index, retrieval, and spatial matching for scene maps

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—input images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images






View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

View cluster alignment—example

Palau Nacional, Montjuic, Barcelona—aligned images

Scene map construction

- $F(p)$: the union of features over all images in visual cluster $C_v(p)$ after alignment

$$F(p) = \bigcup_{q \in C_v(p)} \{ (\; H_{qp} x \; , w) : \; (x, w) \; \in \; F_q \; \}$$

- apply spatial KVQ separately to features mapped to each visual word
- join the resulting codebooks into a single scene map $S(p)$

Scene map construction

- $F(p)$: the union of features over all images in visual cluster $C_v(p)$ after alignment

feature set of photo q

$$F(p) = \bigcup_{q \in C_v(p)} \{ (H_{qp} x, w) : (x, w) \in F_q \}$$

- apply spatial KVQ separately to features mapped to each visual word
- join the resulting codebooks into a single scene map $S(p)$

Scene map construction

- $F(p)$: the union of features over all images in visual cluster $C_v(p)$ after alignment

feature set of photo q

$$F(p) = \bigcup_{q \in C_v(p)} \{ (H_{qp} x \; , w) : \; (x, w) \; \in \; F_q \; \}$$

(position, visual word)

- apply spatial KVQ separately to features mapped to each visual word
- join the resulting codebooks into a single scene map $S(p)$

Scene map construction

- $F(p)$: the union of features over all images in visual cluster $C_v(p)$ after alignment

position aligned to reference image p

feature set of photo q

$$F(p) = \bigcup_{q \in C_v(p)} \{ (H_{qp} x , w) : (x, w) \in F_q \}$$

(position, visual word)

- apply spatial KVQ separately to features mapped to each visual word
- join the resulting codebooks into a single scene map $S(p)$

Scene map construction

- $F(p)$: the union of features over all images in visual cluster $C_v(p)$ after alignment

position aligned to reference image p

feature set of photo q

$$F(p) = \bigcup_{q \in C_v(p)} \{ (H_{qp} x , w) : (x, w) \in F_q \}$$

union over all photos q of $C_v(p)$

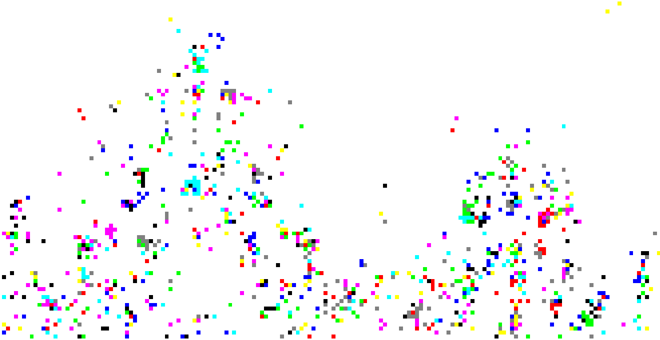
(position, visual word)

- apply spatial KVQ separately to features mapped to each visual word
- join the resulting codebooks into a single scene map $S(p)$

Scene map construction—example

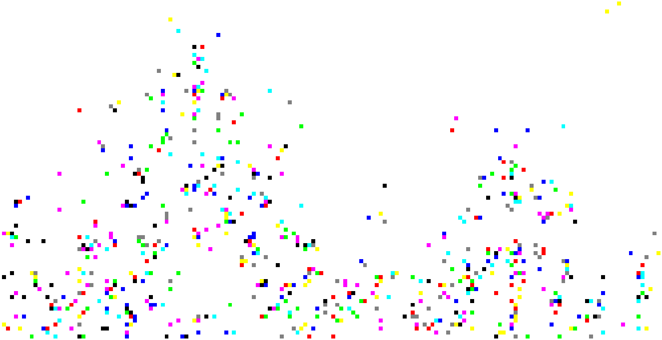
visual cluster containing 30 images of Palau Nacional, Montjuic

Scene map construction—example



before vector quantization

Scene map construction—example



after vector quantization

Scene map indexing

index construction

- scene maps and images have the same representation—sets of features
- index all scene maps by visual word in an inverted file

query

- re-rank using the single correspondence assumption [Philbin et al. 2007]
- whenever a scene map $S(p)$ is found relevant, all images $q \in C_v(p)$ are retrieved as well

European Cities 1M dataset (EC1M)

- $1,081$ images from Barcelona annotated into 35 groups
- all geo-tagged Flickr images

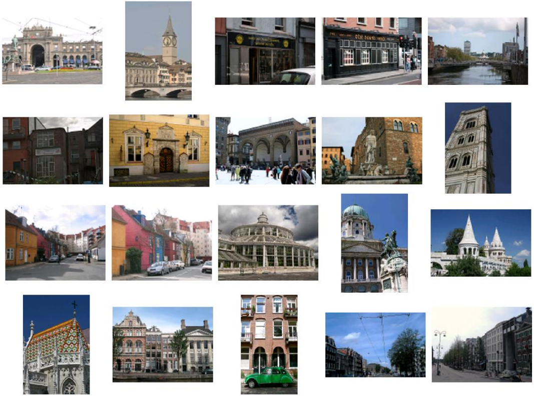
17 landmark groups

18 non-landmark groups

Publicly available: `http://image.ntua.gr/iva/datasets/ec1m/`

European Cities 1M dataset (EC1M)

- $908,859$ distractor images from 21 European cities, excluding Barcelona
- most depict urban scenery like the ground-truth



Publicly available: `http://image.ntua.gr/iva/datasets/ec1m/`

Mining statistics—scene maps

- 1M images, 58 hours, single machine (8GB RAM), landmarks and non-landmarks

Mining statistics—related work

- [Chum et al. 2009] web-scale clustering: 5M images, 28 hours, single machine (64GB RAM), popular subsets only
- [Agarwal et al. 2009] Rome in a day: 150K images, 24 hours, 500 cores
- [Frahm et al. 2010] Rome in a cloudless day: 3M images, 24 hours, GPU
- [Heath et al. 2010] image webs: 200K images, 4,5 hours, 500 cores

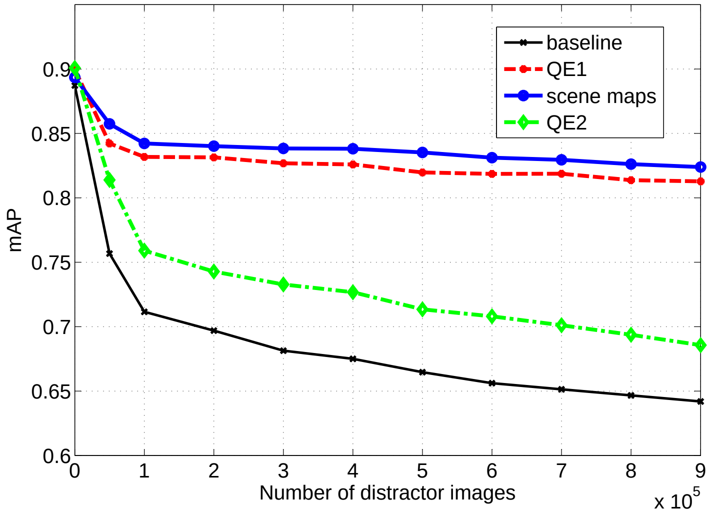
Retrieval comparisons

- baseline: bag-of-words with fast spatial matching [Philbin et al. 2007]
- QE1: iterative query expansion, re-query using the retrieved images and merge results, 3 times iteratively
- QE2: create a scene map using the initial query's result and re-query once
- both QE schemes similar to total recall [Chum et al. 2007]

query timing

Method	time	mAP
Baseline BoW	1.03s	0.642
QE1	20.30s	0.813
QE2	2.51s	0.686
Scene maps	1.29s	0.824

Retrieval statistics



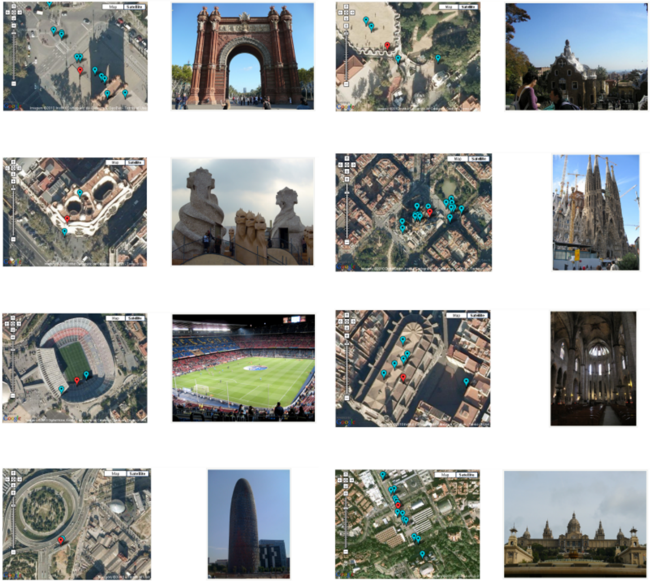
Outline

Location and landmark recognition

[Y. Kalantidis et al. 2011]

- assume that a subset of similar photos are correctly geo-tagged, and not too far apart
- recognize the location where the query photo is taken, as the centroid of the most populated spatial (geo) cluster
- cross-validate locations and text (title, tags) of similar images with Geonames entries and geo-referenced Wikipedia articles
- link to known landmarks or points of interest

Location recognition—examples

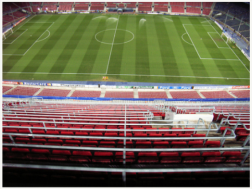


Landmark recognition—examples

Suggested tags: Park Güell, Barcelona
Frequent user tags: Best of, me, Palau Güell

Suggested tags: La Barceloneta, Barcelona
Frequent user tags: honeymoon, wedding, straße

Suggested tags: Triumphal arch, Arc de Triomf, Barcelona
Frequent user tags: Sant Pere, Santa Caterina i La Ribera, macba, Passeig de Lluís Companys, lluis companys, Sant Beltra



Suggested tags: FC Barcelona Museum, Camp Nou, Barcelona
Frequent user tags: champions league, vfb, vfb stuttgart, Zoo de Barcelona, Camp Nou

Suggested tags: Montjuïc circuit, Museu Nacional d'Art de Catalunya, Barcelona
Frequent user tags: Montjuic, castellers, Travelling Pooh, architecture, mnac

Suggested tags: Sagrada Familia, Sagrada Família, Barcelona
Frequent user tags: gaudi, Sagrada Familia, sagrada, familia, expiatorio

http://viral.image.ntua.gr

Query

Results

Estimated Location, Similar Image, Incorrectly geo-tagged, Unavailable

Suggested tags: Buxton Memorial Fountain, Victoria Tower Gardens, London
Frequent user tags: Victoria Tower Gardens, Buxton Memorial Fountain, Winchester Palace, Architecture, Victorian gothic

Similar Images

Similarity: 0.619
Details Original

Similarity: 0.491
Details Original

Similarity: 0.397
Details Original

Similarity: 0.385
Details Original

Similar of similar

Original

Original

Original

Original

Original

Original

Original

Original

Similar of similar

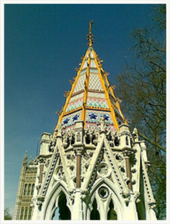

Original


Original


Original


Original


Original


Original


Original


Original

Similar of similar

Original

Original

Original

Original

Original

Original

Original

Original

Suggested tags

Suggested tags: Buxton Memorial Fountain, Victoria Tower Gardens, London
Frequent user tags: Victoria Tower Gardens, Buxton Memorial Fountain, Winchester Palace, Architecture, Victorian gothic

Related wikipedia articles

WIKIPEDIA
The Free Encyclopedia

- Main page
- Contents
- Featured content
- Current events
- Random article

Interaction
- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

Toolbox
- What links here
- Related changes
- Upload file
- Special pages
- Permanent link
- Cite this page

Print/export

New features Log in / create account

Article Discussion Read Edit View history Search

Buxton Memorial Fountain

From Wikipedia, the free encyclopedia

The **Buxton Memorial Fountain** is a memorial and drinking fountain in London, the United Kingdom, that commemorates the emancipation of slaves in the British Empire in 1834.

It was commissioned by Charles Buxton MP, and was dedicated to his father Thomas Fowell Buxton along with William Wilberforce, Thomas Clarkson, Thomas Babington Macaulay, Henry Brougham and Stephen Lushington, all of whom were involved in the abolition. It was designed by Gothic architect Samuel Sanders Teulon (1812–1873) in 1865 coincidently with the passing of the Thirteenth Amendment to the United States Constitution, which effectively ended the western slave-trade.[1]

It was originally constructed in Parliament Square, erected at a cost of £1,200. As part of the postwar redesign of the square it was removed in 1949 and not reinstated in its present position in Victoria Tower Gardens until 1957.[2] There were eight decorative figures of British rulers on it, but four were stolen in 1960 and four in 1971. They were replaced by fibreglass figures in 1980. By 2005 these were missing, and the fountain was no longer working. Between autumn 2006 and February 2007 restoration works were carried out. The restored fountain was unveiled on 27 March 2007 as part of the commemoration of the 200th anniversary of the act to abolish the slave trade [3].

A memorial plaque commemorating the 150th anniversary of the Anti-Slavery Society was added in 1989.

The Buxton Memorial Fountain, designed by Samuel Sanders Teulon, celebrating the emancipation of slaves in the British Empire in 1834, in Victoria Tower Gardens, Millbank,

Description [edit]

The base is octagonal, about twelve feet in diameter, having open arches on the eight sides, supported on clustered shafts of polished Devonshire marble around a large central shaft, with four massive granite basins. Surmounting the pinnacles at the angles of the octagon are eight figures of bronze, representing the different rulers of England; the Britons represented by Caractacus, the Romans by Constantine, the Danes by Canute, the Saxons by Alfred, the Normans by William the Conqueror, and so on, ending with Queen Victoria. The fountain bears an inscription to the effect that it is "intended as a memorial of those members of Parliament who, with Mr. Wilberforce, advocated the abolition of the British slave-trade, achieved in 1807; and of those members of Parliament who, with Sir T.

Related wikipedia articles

Wikipedia
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate

Interaction
About Wikipedia
Community portal
Recent changes
Contact Wikipedia
Help

Toolbox
What links here
Related changes
Upload file
Special pages
Permanent link
Cite this page

Print/export

New features Log in / create account

Article Discussion Read Edit View history Search

Victoria Tower Gardens

From Wikipedia, the free encyclopedia

Coordinates: 51°29′49.0″N 0°7′30.0″W

Victoria Tower Gardens is a public park along the north bank of the River Thames in London. As its name suggests, it is adjacent to the Victoria Tower, the south-western corner of the Palace of Westminster. The park, which extends southwards from the Palace to Lambeth Bridge, sandwiched between Millbank and the river, also forms part of the Thames Embankment.

Victoria Tower Gardens, 2005, with the Buxton Memorial Fountain at the front and the Palace of Westminster in the background

Contents [hide]

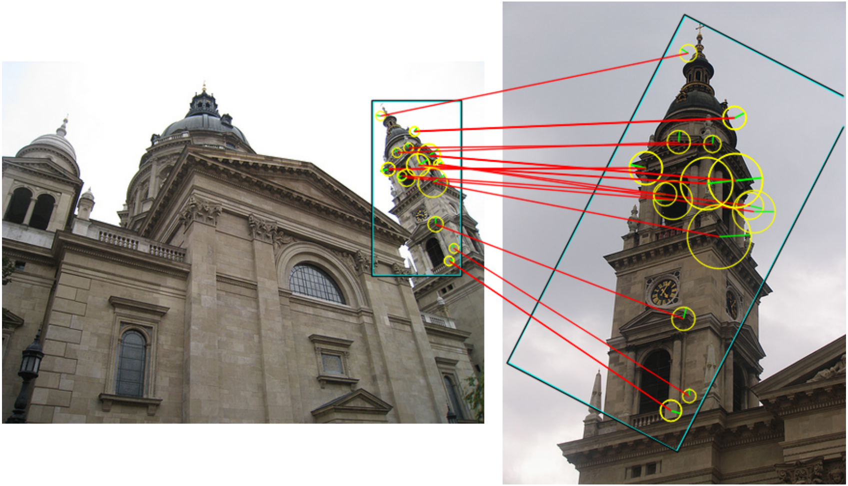
Features [edit]

The park features:

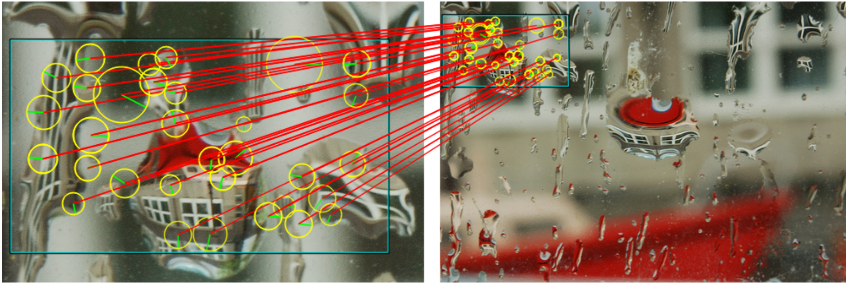
- A reproduction of the sculpture *The Burghers of Calais* by Auguste Rodin, purchased by the British Government in 1911 and positioned in the Gardens in 1915.
- A 1930 statue of the suffragette Emmeline Pankhurst, by A.G. Walker.
- The Buxton Memorial Fountain – originally constructed in Parliament Square, this was removed in 1940 and placed in its present position in 1957. It was commissioned by Charles Buxton MP to commemorate the emancipation of slaves in 1834, dedicated to his father Thomas Fowell Buxton, and designed by Gothic architect Samuel Sanders Teulon (1812–1873) in 1865.
- A stone wall with two modern-style goats with kids – situated at the southern end of the Gardens.

Transport [edit]

Visual similarity



Visual similarity



Outline

Recall feature point matching

1. construct the $m \times n$ proximity matrix G with elements

$$g_{ij} = \exp(-d_{ij}^2 / 2\sigma^2)$$

2. perform singular value decomposition of G

$$G = USV^{\mathrm{T}}$$

   where U, V are orthogonal matrices of dimension m, n and S is a non-negative diagonal $m \times n$ matrix

3. replace each diagonal element s_{ij} of S by 1 and reconstruct

$$P = UEV^{\mathrm{T}}$$

4. finally, associate points a_i and b_j if element p_{ij} of P is the greatest element in its row and its column

Matlab **code**

```
function [m1, m2] =
match(                  x1,                   y1,
                        x2,                   y2, F s)


    [Ax1, Ax2] = meshgrid  (x1, x2);
    [Ay1, Ay2] = meshgrid  (y1, y2);

                D = sqrt((Ax1 - Ax2)  .^ 2 + (Ay1 - Ay2)  .^ 2);
                G = exp(-D  .^ 2 ./ (2 *   s    ^ 2));

    [U, S, V] = svd  (G);
                E = S > 0;
                P = U   * E   * V'   ;


    [tmp, c] = max  (P, [], 2);
    [tmp, r] = max  (P, [], 1);

                match = r(c) ==    (1 :  length(c)   );
       m1 = find(match);
       m2 = c(match)';
```

ivl C++ code

```
template<class F>   ret<array<F>, array<F> >
match(const array<F>& x1, const array<F>& y1,
      const array<F>& x2, const array<F>& y2, F s)
{
   array_2d<F> Ax1, Ax2, Ay1, Ay2, U, S, V, tmp;
   _(Ax1, Ax2) = meshgrid++(x1, x2);
   _(Ay1, Ay2) = meshgrid++(y1, y2);

   array_2d<F> D = sqrt((Ax1 - Ax2) ->* 2 + (Ay1 - Ay2) ->* 2);
   array_2d<F> G = exp(-D ->* 2  / (2 * _[s] ->* 2));

   _(U, S, V) = svd++(G);
   array_2d<F> E = S > 0;
   array_2d<F> P = U ()* E ()* V(!_);

   array<int> c, r;
   _(tmp, c) = max++(P, _ , 2);
   _(tmp, r) = max++(P, _ , 1);

   array<bool> match = r[c] == rng(0, c.length() - 1);
   return _(find(match),
            c[match]);
}
```

`ivl` library

(Kontosis and Avrithis, expected 2011)

- `C++` template library, compatible to `STL`
- supports most types, syntax and built-in operations of `Matlab` language
- fully optimized: minimal overhead/temporaries/copying; all array expressions boil down to a single `for` loop
- uses multiple CPU cores
- integrated with basic image functionalities of `OpenCV`
- integrated with most common `LAPACK` routines

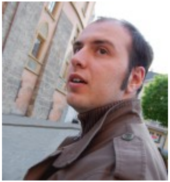
plans

- integration with `QT` to support visualization
- `CUDA` massively parallel implementation on GPU

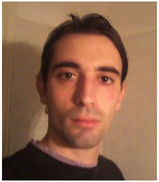
Credits

Yannis Kalantidis

Giorgos Tolias



Christos Varitimidis



Kimon Kontosis

Marios Phinikettos

Phivos Mylonas

Kostas Rapantzikos

Yannis Avrithis

project pages

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

VIRaL

http://viral.image.ntua.gr

datasets

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

thank you!