Retrieving Landmark and Non-Landmark Images from Community Photo Collections

Yannis Avrithis, Yannis Kalantidis, Giorgos Tolias and Evaggelos Spyrou




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

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

Community photo collections

clustering / landmark recognition

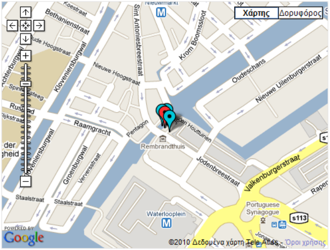
- focus on popular subsets
- applications: browsing, 3D reconstruction

[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

State-of-the-art limitations

location recognition

- city-scale, local features, inverted index [Schindler et al. 2007]
- im2gps: world scale, global features, low matching accuracy, geolocation probability map [Hayes and Efros 2008]

structure from motion / 3D reconstruction

- photo tourism: up to 10^3 images [Snavely et al. 2006]
- city-scale model reconstuction, 10^5 images [Agarwal et al. 2009]

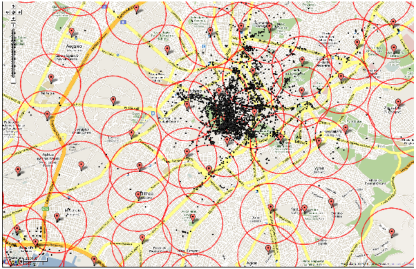
clustering / landmark recognition

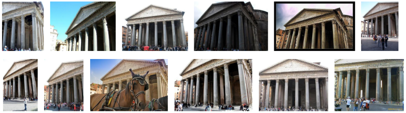
- web-scale clustering: no location data, popular locations [Chum and Matas 2010]
- overlaping tiles, pairwise homography estimation [Quack et al. 2008, Gammeter et al. 2009]
- tour the world: search by travel guides, parallel computing [Zheng et al. 2009]

An overview of our approach

View clustering

- 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

An overview of our approach

Scene maps

- align all images for each visual cluster to a reference image
- construct a 2D scene map by grouping similar local features
- extend index, retrieval, and spatial matching for scene maps

Kernel Vector Quantization

[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:

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

- obtain a sufficiently 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 $x \in Q(D)$

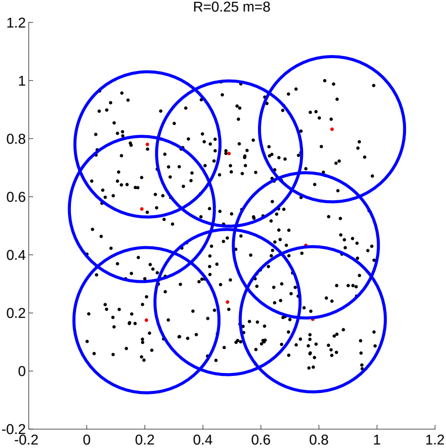
$$\max_{y \in C(x)} d(x, y) < r$$

- the cluster collection

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

is a cover for D

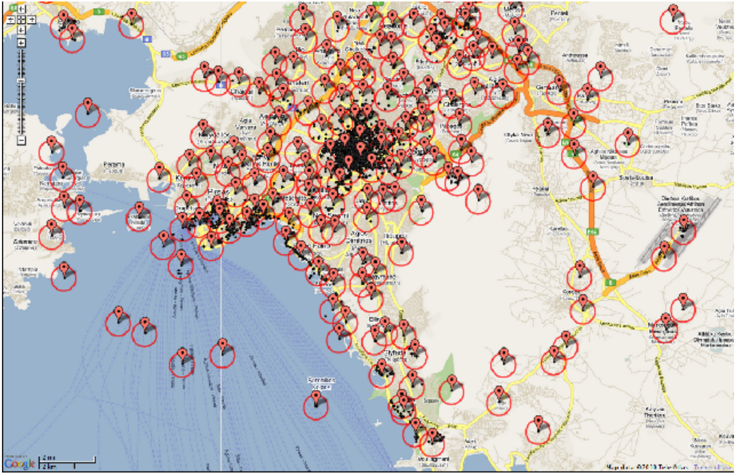
- clusters are overlapping



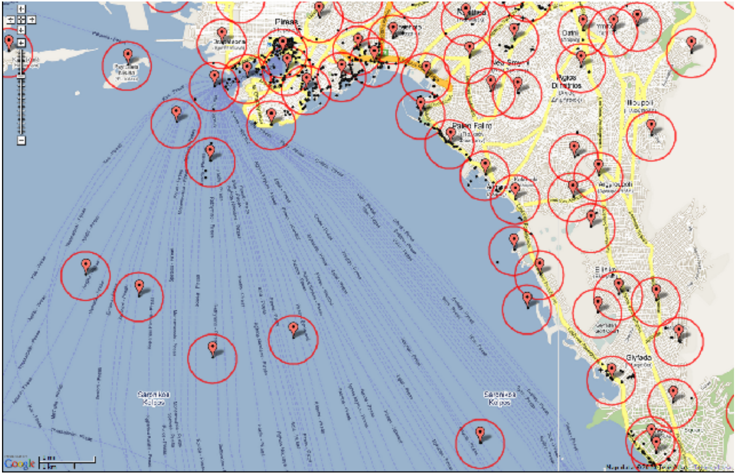
Geo clustering

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

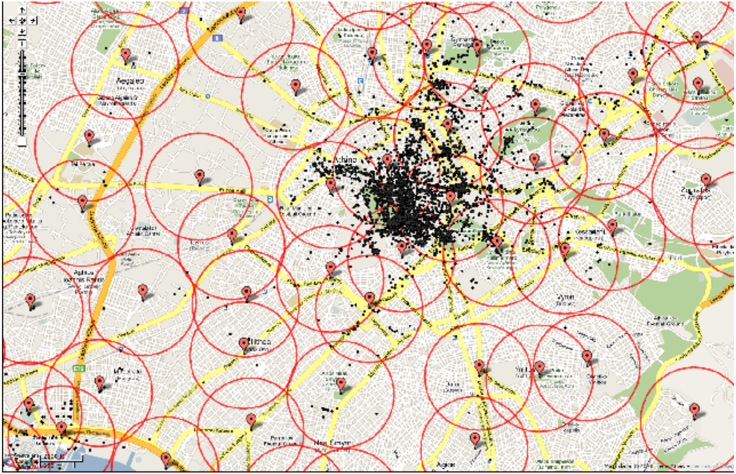
Geo clustering—example



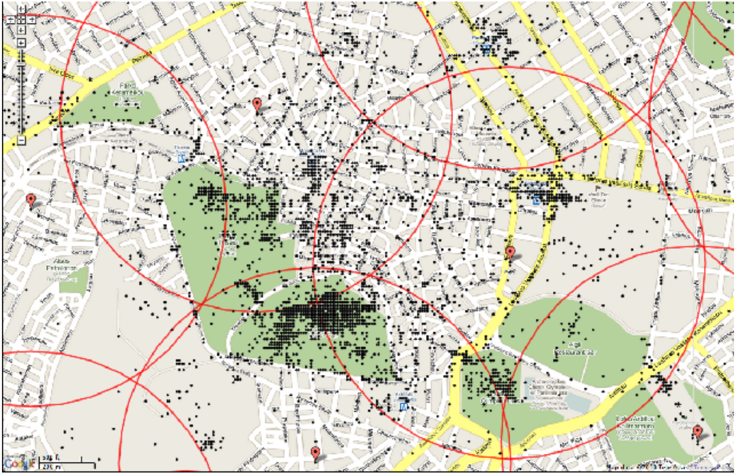
Geo clustering—example



Geo clustering—example



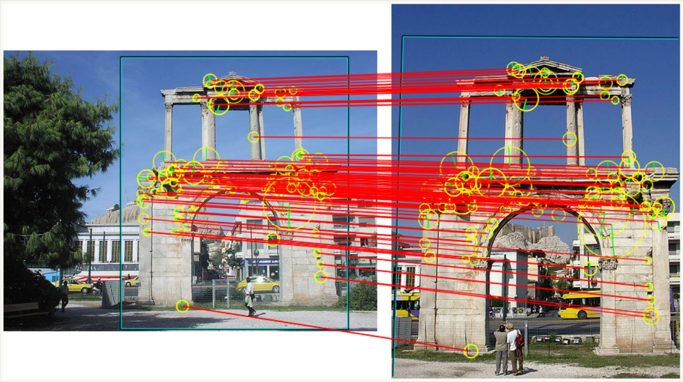
Geo clustering—example



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 $G \in \mathcal{C}_g(P)$, construct codebook $Q_v(G)$ by KVQ in space $(\mathcal{P}, d_v)$ with scale parameter r_v
- the exact formula of $d_v(p, q)$ is not important, the scale parameter specifies a threshold in the number of inliers
- visual cluster: $C_v(p) = \{q \in G : d_v(p, q) < r_v\}$
- visual cluster collection: $\mathcal{C}_v(G) = \{C_v(p) : p \in Q_v(G)\}$
- maximum distortion: equivalent to minimum number of inliers
- overlapping: support gradual transitions of views

Visual clustering

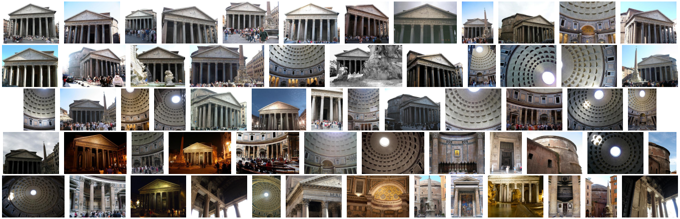
geo-cluster specific sub-linear indexing

- bottleneck: computation of pairwise distances, quadratic in $|G| \rightarrow$ 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 $|G|$

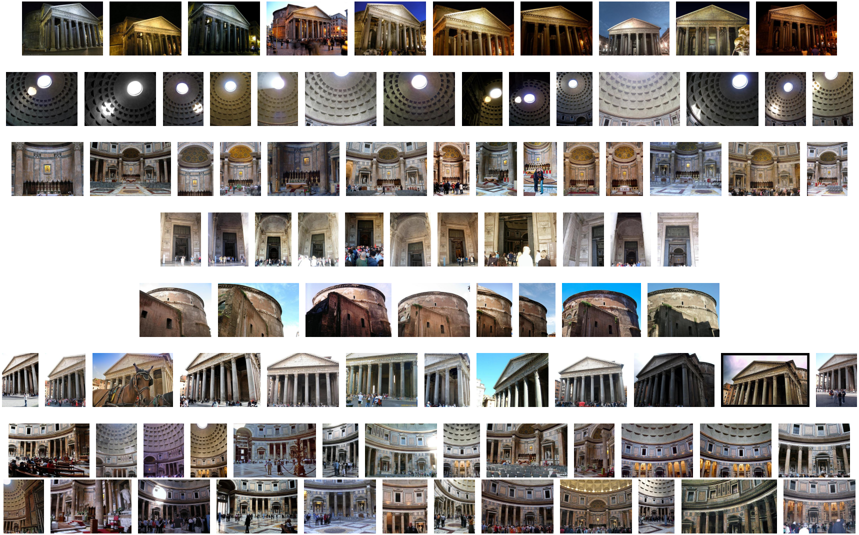
Visual clustering—example

$1,146$ 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



View cluster alignment

so far we know:

- the image associated to the center of a view cluster shares at least one rigid object with all other images in the cluster

alignment

- 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

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

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)

- construct a compact representation of $F(p) \rightarrow$ 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)

- construct a compact representation of $F(p) \rightarrow$ 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)

- construct a compact representation of $F(p) \rightarrow$ 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)

- construct a compact representation of $F(p) \rightarrow$ 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)

- construct a compact representation of $F(p) \rightarrow$ scene map $S(p)$

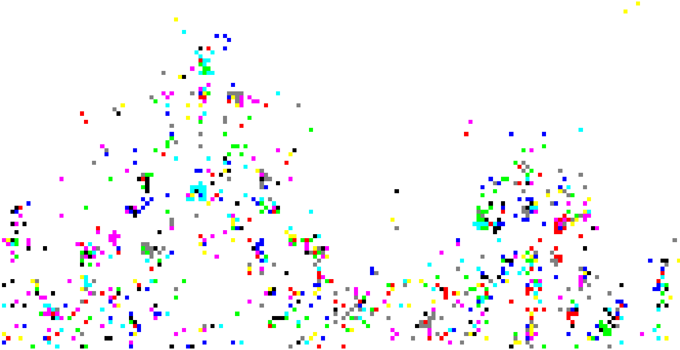
Scene map construction

- construct minimal $S(p) \subseteq F(p)$, such that no feature in $F(p)$ is too distant from its nearest neighbor in $S(p) \rightarrow$ vector quantization
- partition $F(p)$ into a number of disjoint sets, each corresponding to a visual word w and apply KVQ separately
- the scale parameter $r_x = \theta$, where θ is the error threshold used in spatial matching
- join the resulting codebooks into a single scene map

Scene map construction—example

visual cluster containing 30 images of Palau Nacional, Montjuic

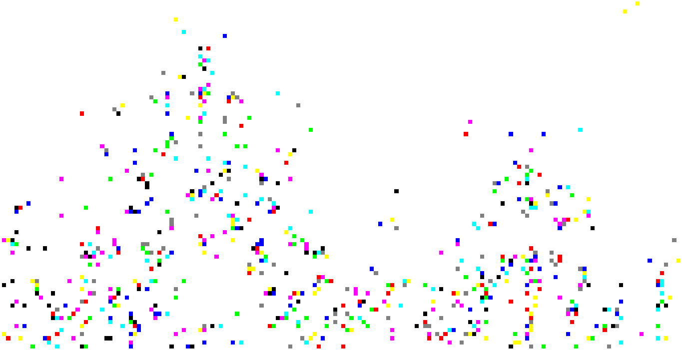
Scene map construction—example



before vector quantization

	before KVQ	after KVQ	compression rate
features	$11,623$	$9,924$	15%
inverted file entries	$11,165$	$8,616$	23%

Scene map construction—example



after vector quantization

	before KVQ	after KVQ	compression rate
features	$11,623$	$9,924$	15%
inverted file entries	$11,165$	$8,616$	23%

Scene map retrieval

index construction:

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

query:

- retrieve scene maps by histogram intersection and TF-IDF
- 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 considered relevant as well

European Cities 1M dataset (EC1M)

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

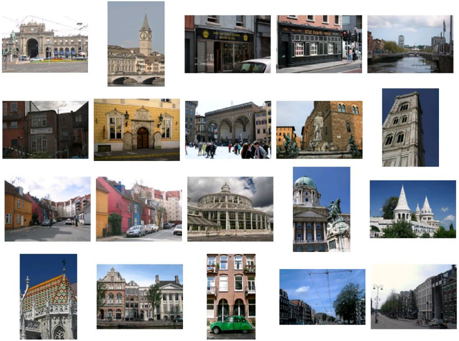
17 landmark groups

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—single machine

view clustering:

- geo clustering takes less than 5 minutes and generates $1,677$ geo-clusters
- visual similarities calculation takes approximately 52 hours
- visual clustering takes approximately 22 minutes and generates $493,693$ visual clusters
- single images are $351,391$ of the visual clusters

scene maps:

- scene map creation takes about 5 hours
- inverted index compression: 25% [1.2Gb]

Related mining statistics

- [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
- **scene maps**: 1M images, 58 hours, single machine (8GB RAM)

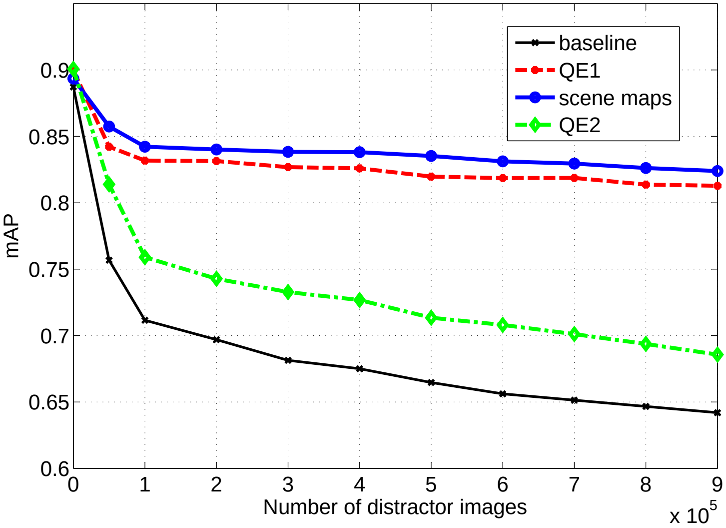
Comparisons

- 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



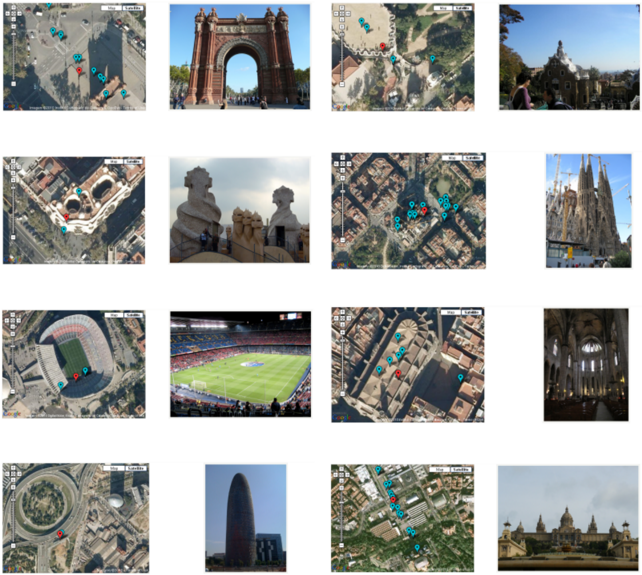
Location recognition

- Y. Kalantidis, G. Tolias, Y. Avrithis, M. Phinikettos, E. Spyrou, P. Mylonas, S. Kollias. VIRaL: Visual Image Retrieval and Localization. In *Multimedia Tools and Applications*, 2011 (in press).

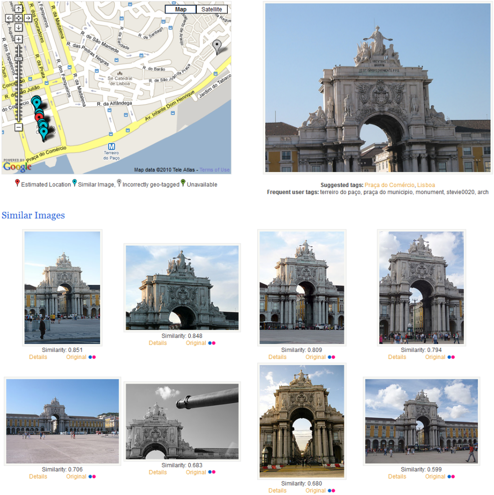
percentage of correctly localized queries:

Method	Distance threshold		
	$< 50m$	$< 100m$	$< 150m$
Baseline BoW	82.5%	91.6%	94.2%
QE1	86.3%	93.5%	96.2%
QE2	86.7%	93.3%	96.5%
Scene maps	**87.8%**	**94.2%**	**97.1%**

Location recognition examples



http://viral.image.ntua.gr



See us tomorrow at Multimedia Grand Challenge!

Discussion - future work

discussion

- geo-cluster specific indexing $\longrightarrow$ fast mining
- considerable increase in retrieval performance
- reduced memory requirements for the index
- can still retrieve any isolated image from the original database

future work

- perceptual summarization / browsing
- landmark recognition
- exact localization *i.e.* pose detection

project page

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

EC1M dataset

http://image.ntua.gr/iva/datasets/ec1m

VIRaL

http://viral.image.ntua.gr

Thank you!