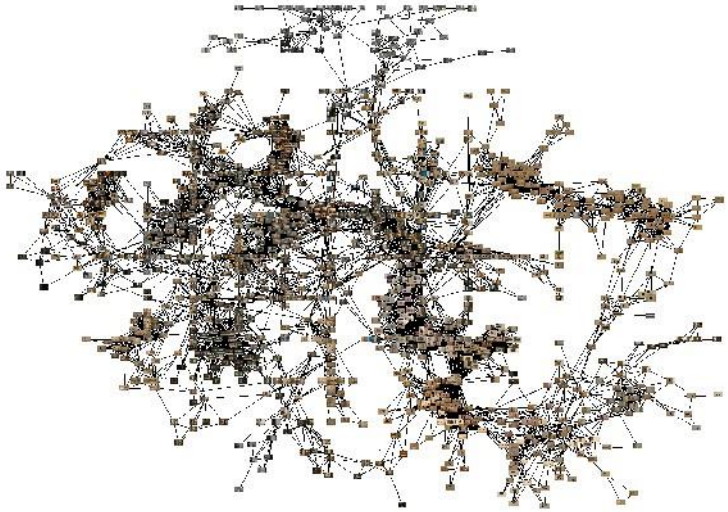
Image Webs

Computing and Exploiting Connectivity in Image Collections



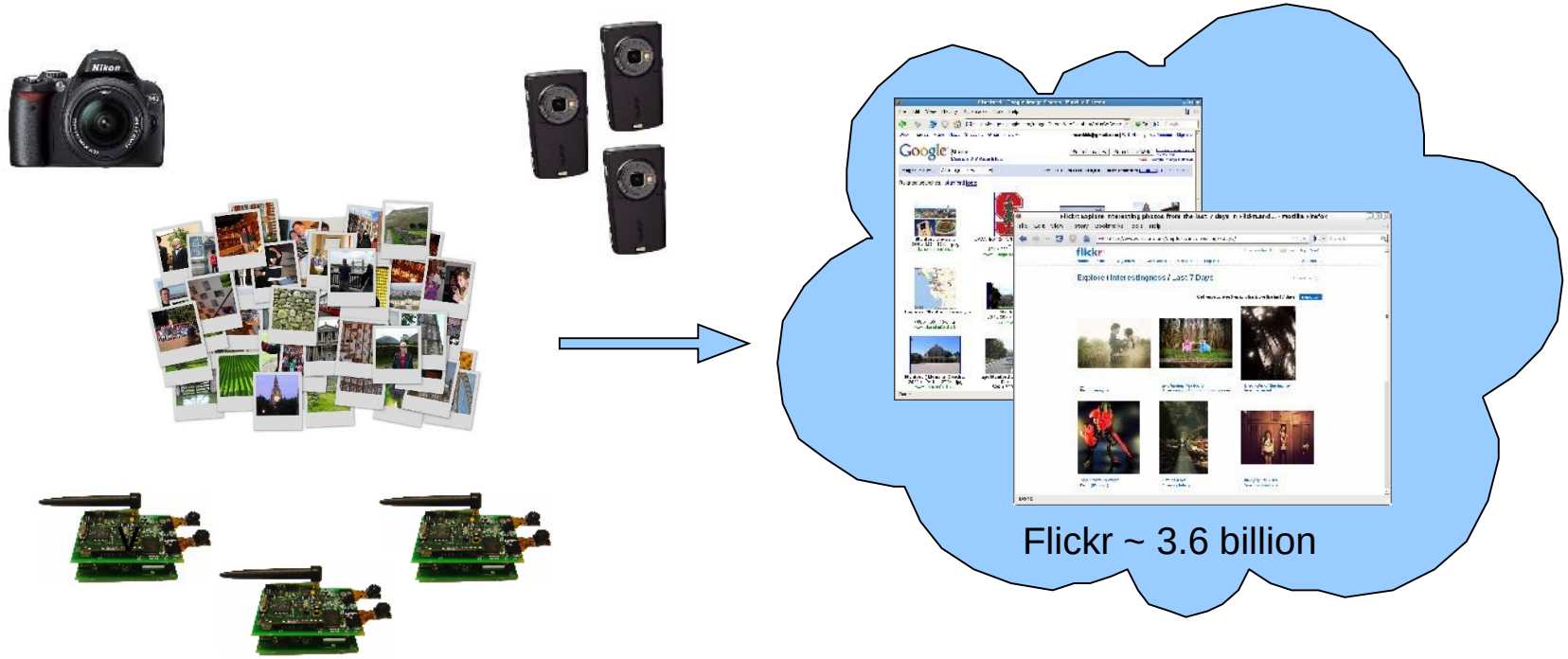
Kyle Heath, Natasha Gelfand, Maks Ovsjanikov, Mridul Aanjaneya, Leo Guibas

Stanford University , Nokia Research - Palo Alto

Presenter: Sugeerth Murugesan

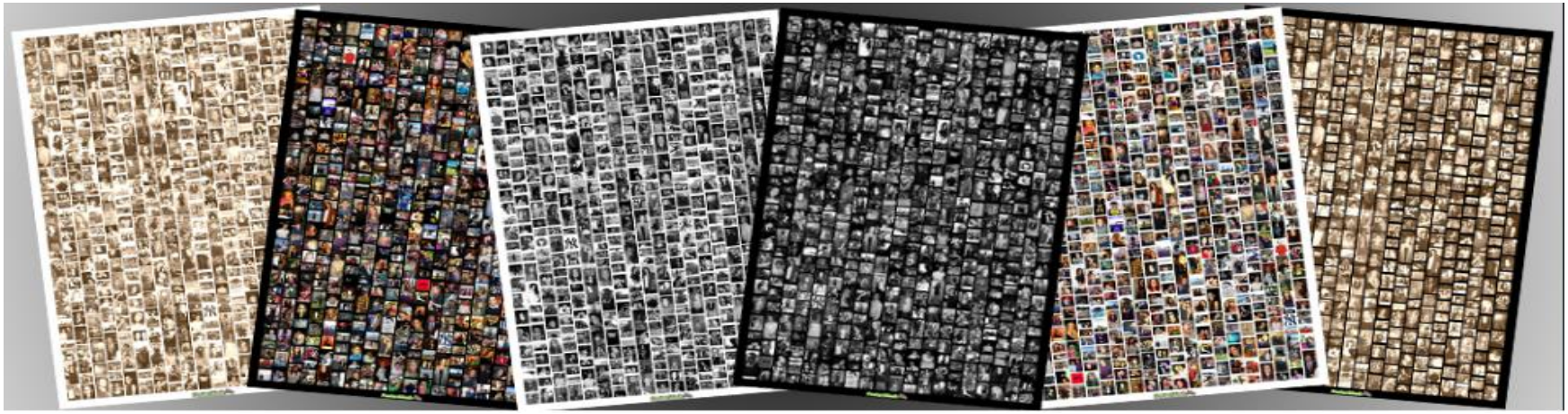
Large image collections are emerging

- Innovation in devices
  - Images are cheap and easy to create and store
- Availability of high-speed internet
  - Images are easy to share



Challenges in large image collections

- **Challenges**
  - Understanding large image collections
  - Retrieval of information you want



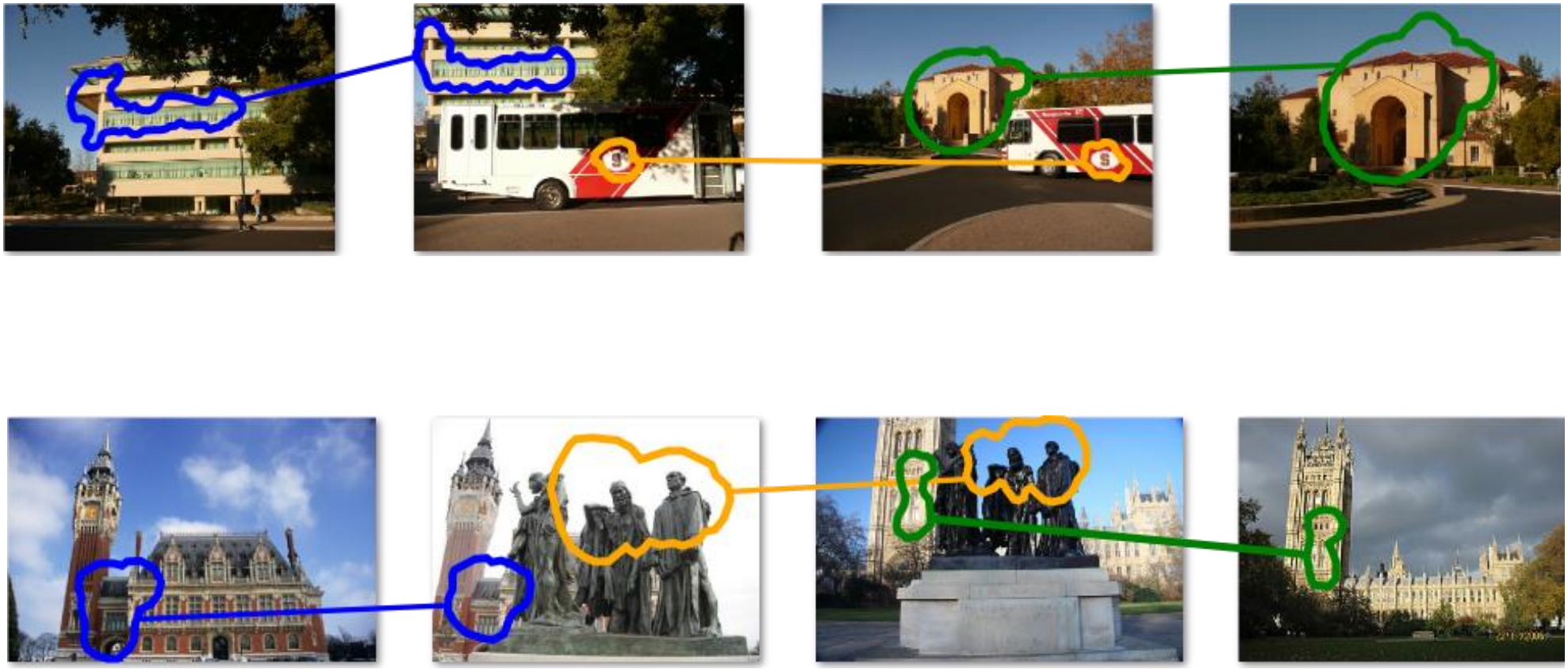
- **Opportunities**
  - Lots of visual information which is yet uneasy to access
  - Use data driven approach for hard problems like general object recognition

Goal: Link images together like web documents

- Discover “visual hyperlinks” between images in the collection induced by shared objects

- Exploit these links to search, visualize, and mine data from large image collections

Goal: Link images together like web documents



Overview

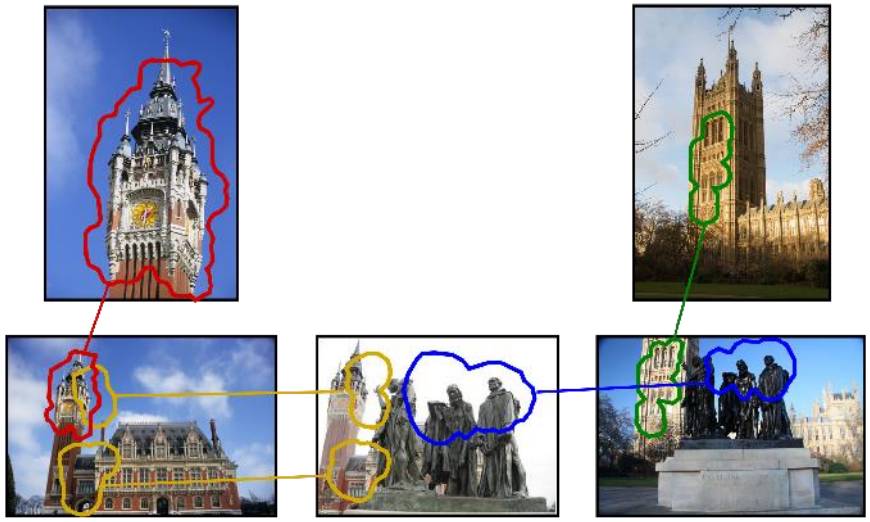
- What is an Image Web?
- Efficient construction
- Application demo

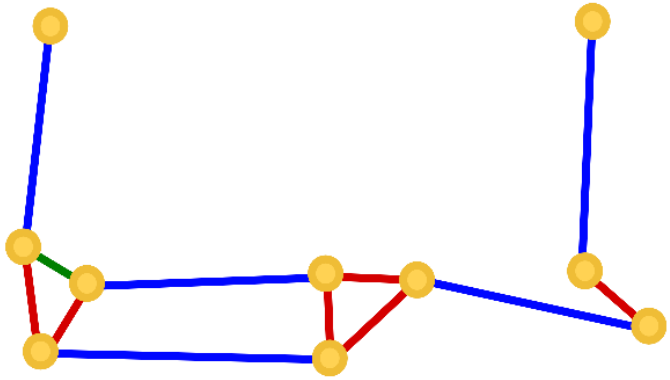
What is an Image Web?

- An **Image Web** is a graph generated by
  - Detecting corresponding regions in pairs of images
  - Forming links between these regions

Link types

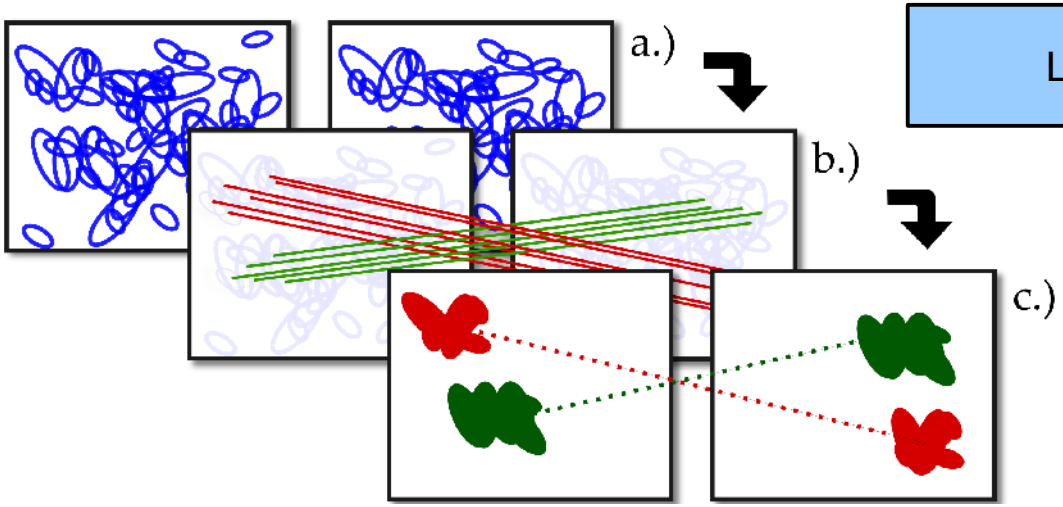
- Match (M)-links (density of matches)
- Overlap (O)-links (degree of overlap)
- Pivot (P)-links (fixed distance)





Affine cosegmentation

Goal: Detect the shared region between a pair of images



Local Feature Extraction

Feature Matching & Verification

Shared Region Segmentation

Keypoints
1) Harris –affine
2) Hessian-affine
3) Maximal color stable regions with SIFT descriptors

Feature matching
Detecting affine consistant features

Shared region segmentation
Merging keypoint support regions

Efficient Image Web construction

- Ideally an Image Web would be built by cosegmenting all pairs of images
  - O(N^2) cosegmentation operations too expensive
- Instead, quickly recover the essential connectivity with a small number of cosegmentation attempts
  - **Phase 1: Discover connected components**
  - **Phase 2: Boost component connectivity**

Phase 1: Discover connected components

- Pre-processing
  - Each image contributes cosegmentation candidates by pairing it with its top *K* CBIR(Content based image retrieval) results
  - List of candidates from all images sorted by CBIR similarity score
- Online
  - Choose next valid candidate in list
    - Candidate pair valid if images are in different connected components of image-graph

Phase 2: Boost component connectivity

Use notion of connectivity from spectral graph theory

Algebraic connectivity

- Second smallest Eigenvalue λ_2 of the matrix graph Laplacian

$$L_{i,j} = \begin{cases} d(i) & \text{if } i = j \\ -1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

$d(i)$ Degree of vertex *i*

v_2 *Fiddler vector, eigen vector of second smallest Eigenvalue*

Phase 2: Boost component connectivity

- Want a fixed number of additional edges to maximize algebraic connectivity (NP-hard*)
- Instead, use a greedy strategy proposed by Wang and Van Mieghem **
  - Select next pair of vertices to connect according to largest absolute difference of corresponding entries in the Fiedler vector

$$e = (s, t) \qquad c_e = |v_2(s) - v_2(t)|$$

* D. Mosk-Aoyama "Maximum algebraic connectivity augmentation is NP-hard." Oper. Res. Lett., 2008.
** H. Wang and P. Van Mieghem. Algebraic connectivity optimization via link addition. In Proc. BIONETICS, 2008.

Phase 2: Boost component connectivity

EdgeRank candidate selection strategy:

1) Each image contributes cosegmentation candidate edges by pairing it with its top *K* CBIR results
(among images in same component)

2) Sort candidate edges in decreasing order of connectivity score $c_e = |v_2(s) - v_2(t)| \quad e = (s,t)$

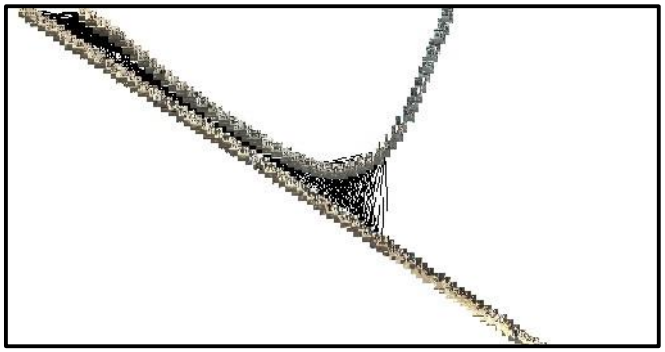
3) Attempt cosegmentation in this order...
If a cosegmentation attempt succeeds, update Laplacian matrix L and Fiedler vector v_2 and return to step 2

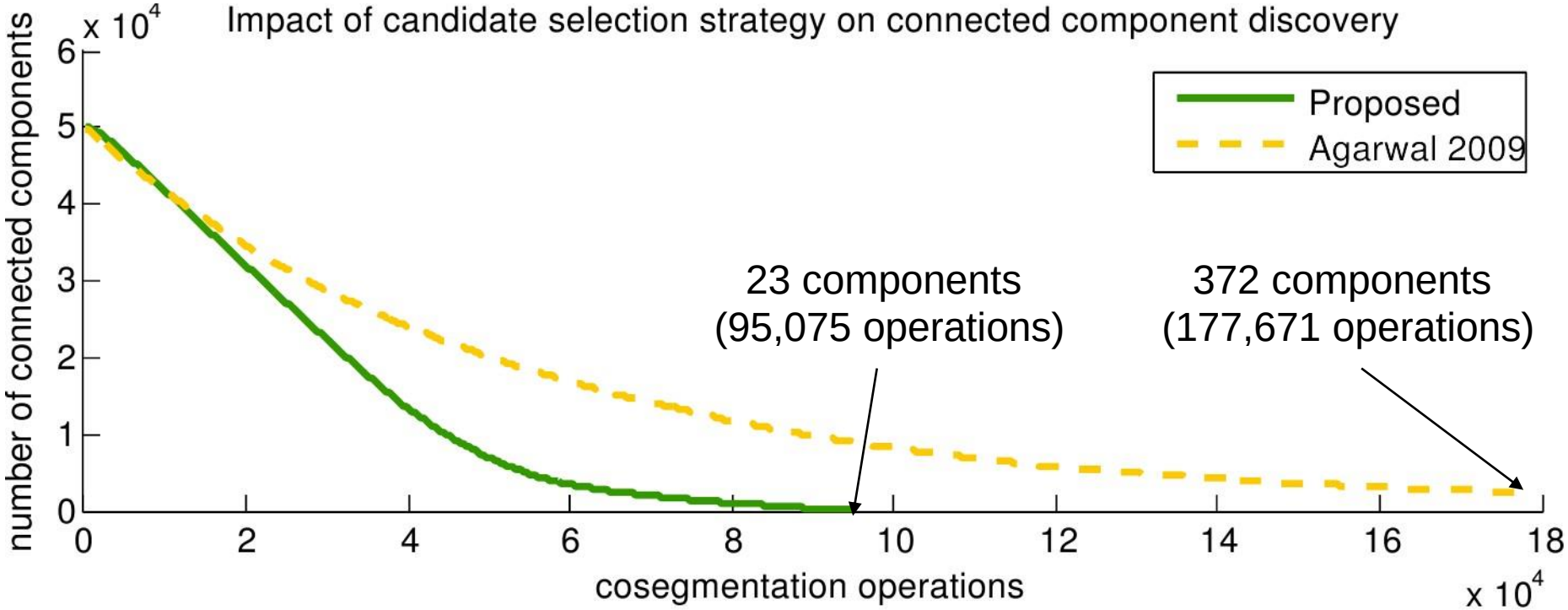
Power iteration method

$$v_2 = \frac{(2nI - L)v_2}{\|(2nI - L)v_2\|_2}, \quad v_2 = v_2 - \frac{1}{n}\sum_{j=1}^{n} v_2(j)$$

Evaluation: Phase 1 strategy

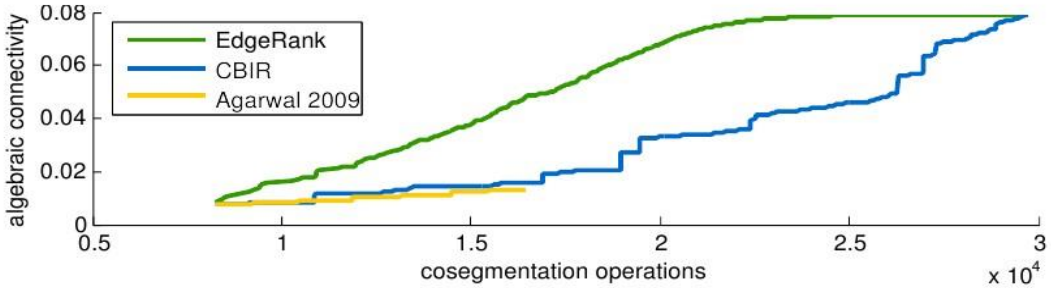
- Dataset: 50,000 StreetView images from Pittsburgh



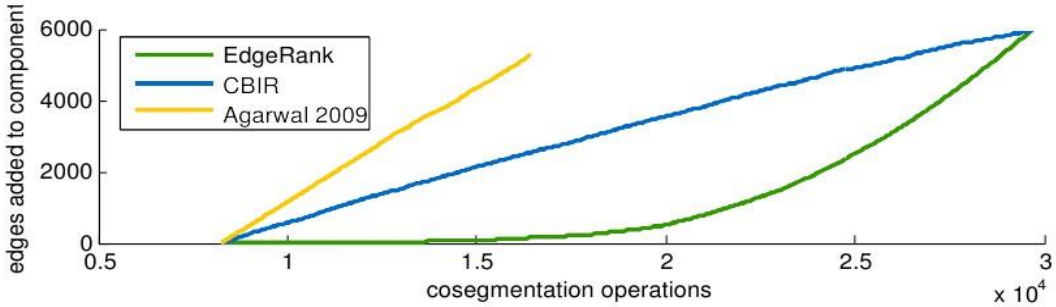


Evaluation: Phase 2 strategy

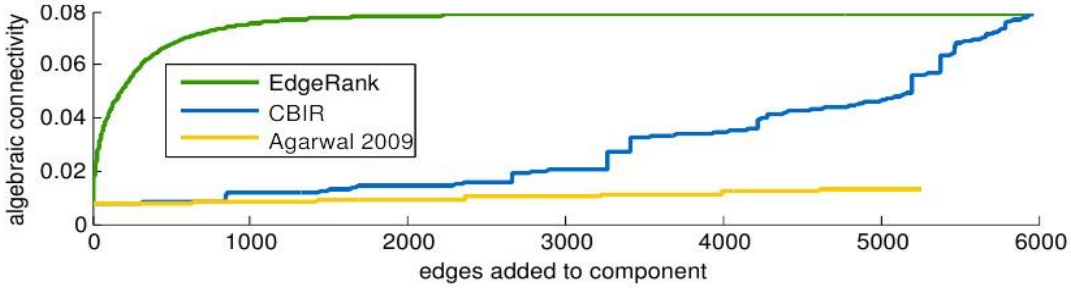
Dataset: 1,257 images in art museum



(a) Connectivity / Construction Time



(b) Edges / Construction Time



(c) Connectivity / Edges

Distributed implementation

- Construction pipeline easily distributed on a computer cluster
  - A manager node issues feature extraction, CBIR, and cosegmentation jobs to worker nodes
  - Communication using the Internet Communication Engine (ICE) middleware and a shared file system

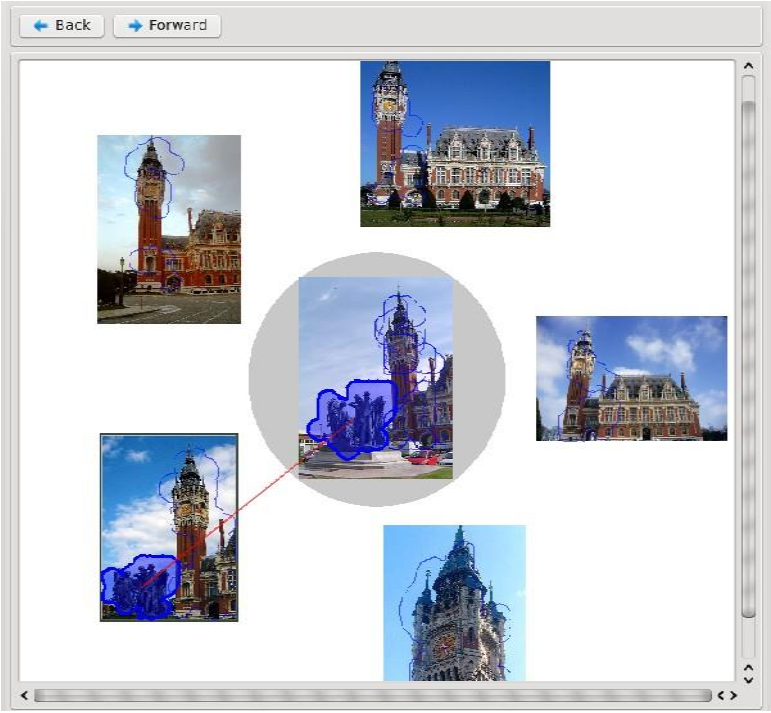
Image Web Construction Timings on Cluster with 500 Nodes

Collection Name (Source)	Images	Components (size > 1)	Largest Component	Construction Time (min) Phase 1	Phase 2	Total
Stanford (Flickr)	193,277	12,505	11,240	173	96	269
Pittsburgh (StreetView)	50,224	23	49,907	7.9	70	78
London (Panoramio)	17,925	902	4,617	7.7	5.9	14
Art Museum (created)	1,257	5	1,217	0.06	0.74	0.8

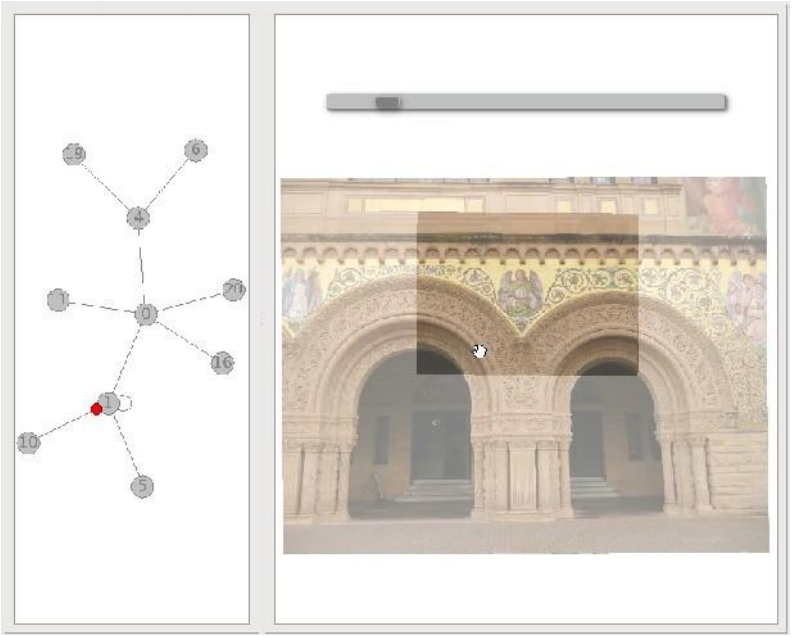
Applications

- Image collection exploration

Visual Hyperlink Browser
(low-level view)



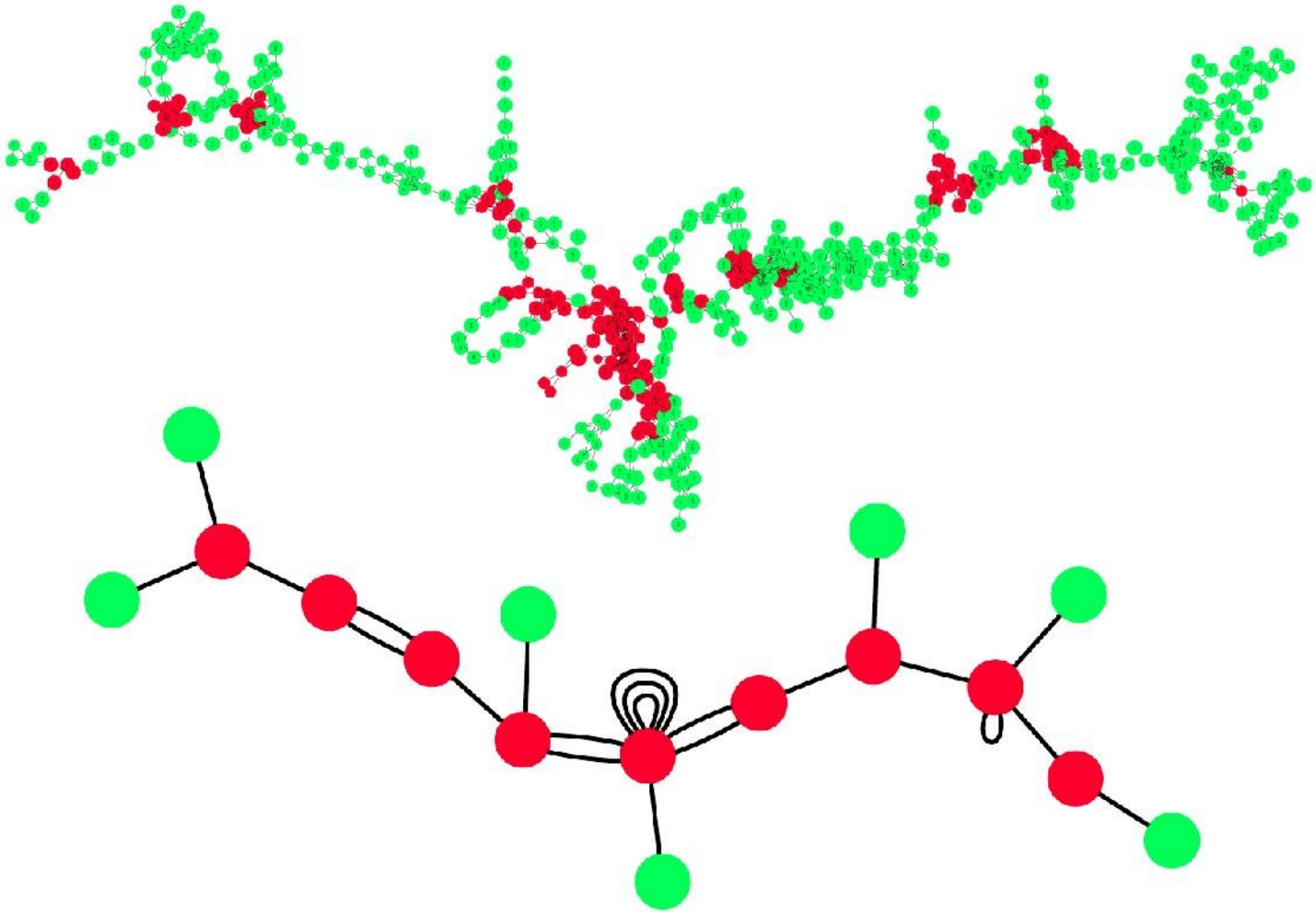
Summary Graph Browser
(high-level view)



Summary Graph

- **Problem:** Image Web can be large and complex, visualization may be difficult
- **Goal:** Create a simplified representation at a desired level of detail that captures the global structure
- **Approach:** Apply techniques from computational topology to detect structures that persist across scales

Summary Graph



Future Work

- Extract meaning “in words” of a path between two image.
- Analyzing sub graphs, does it provide interesting cues?
- Scalability to millions of nodes??

Why optimize connectivity?

From *Building Rome in a Day* - S. Agarwal, et al. ICCV 2009

Applications

Summary Graph

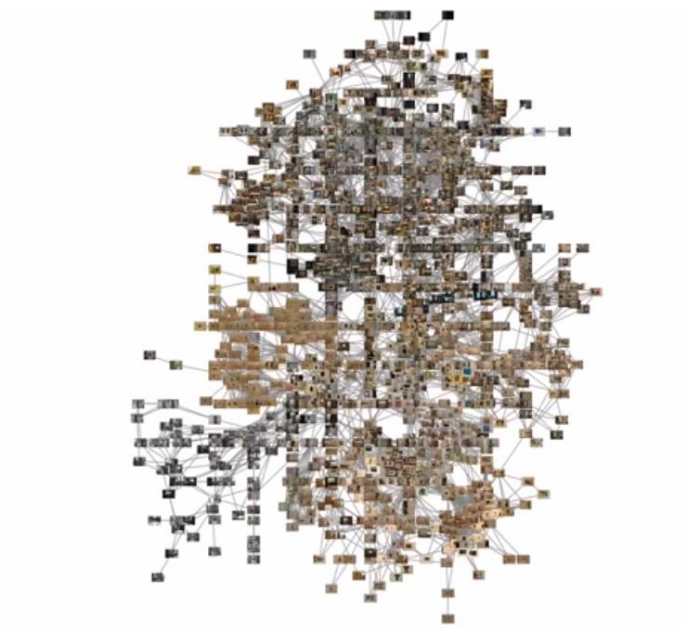
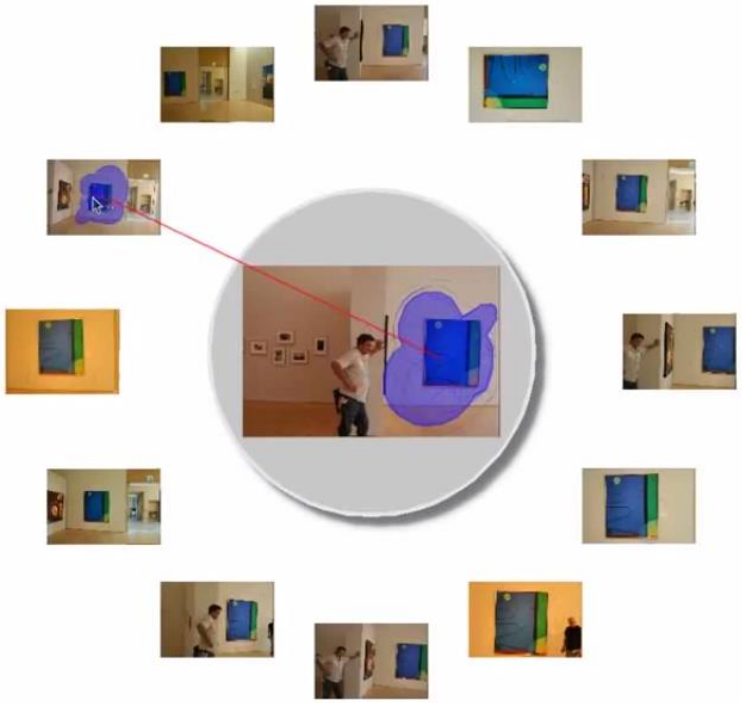
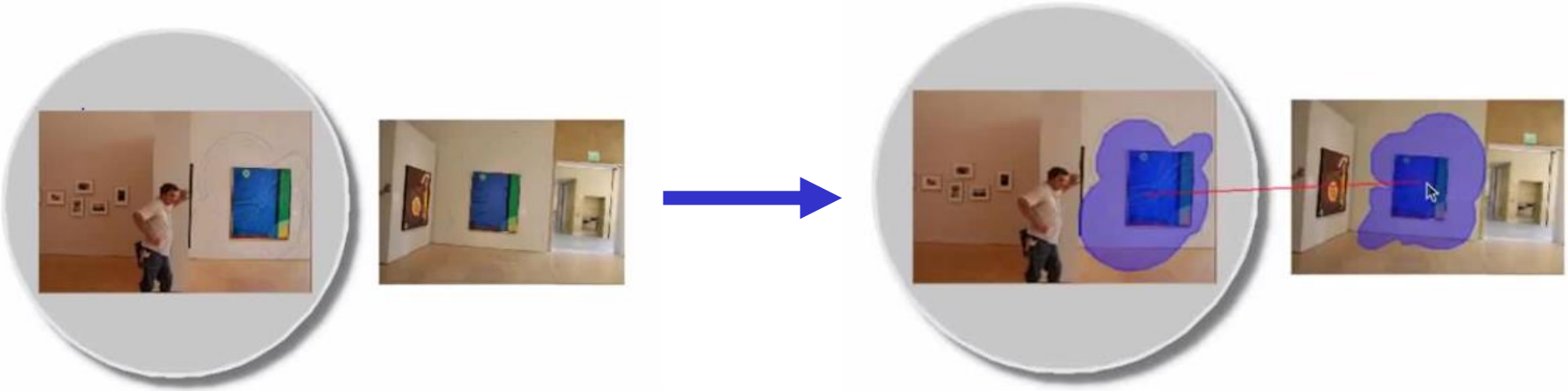


Image Collection Exploration





Content based image retrieval