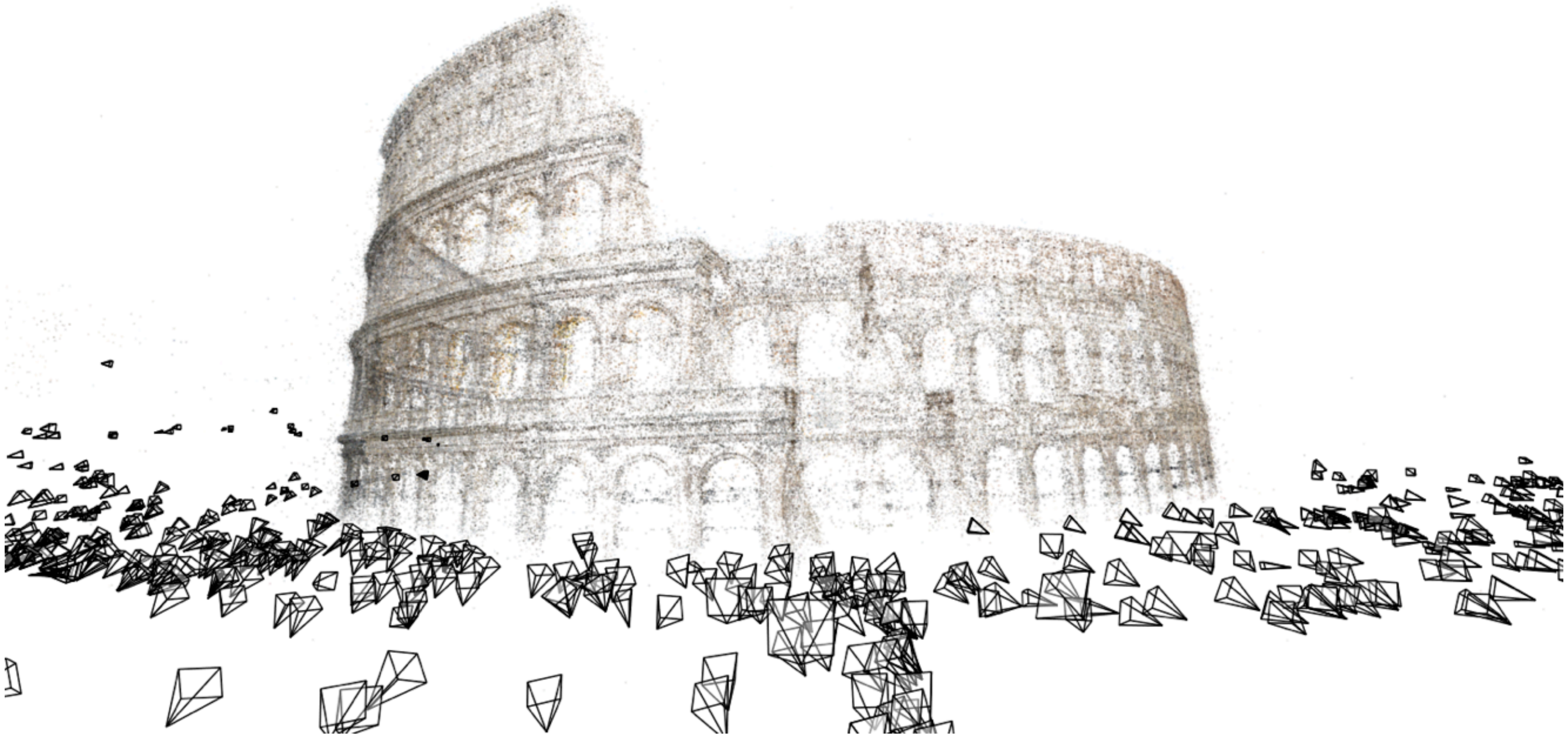
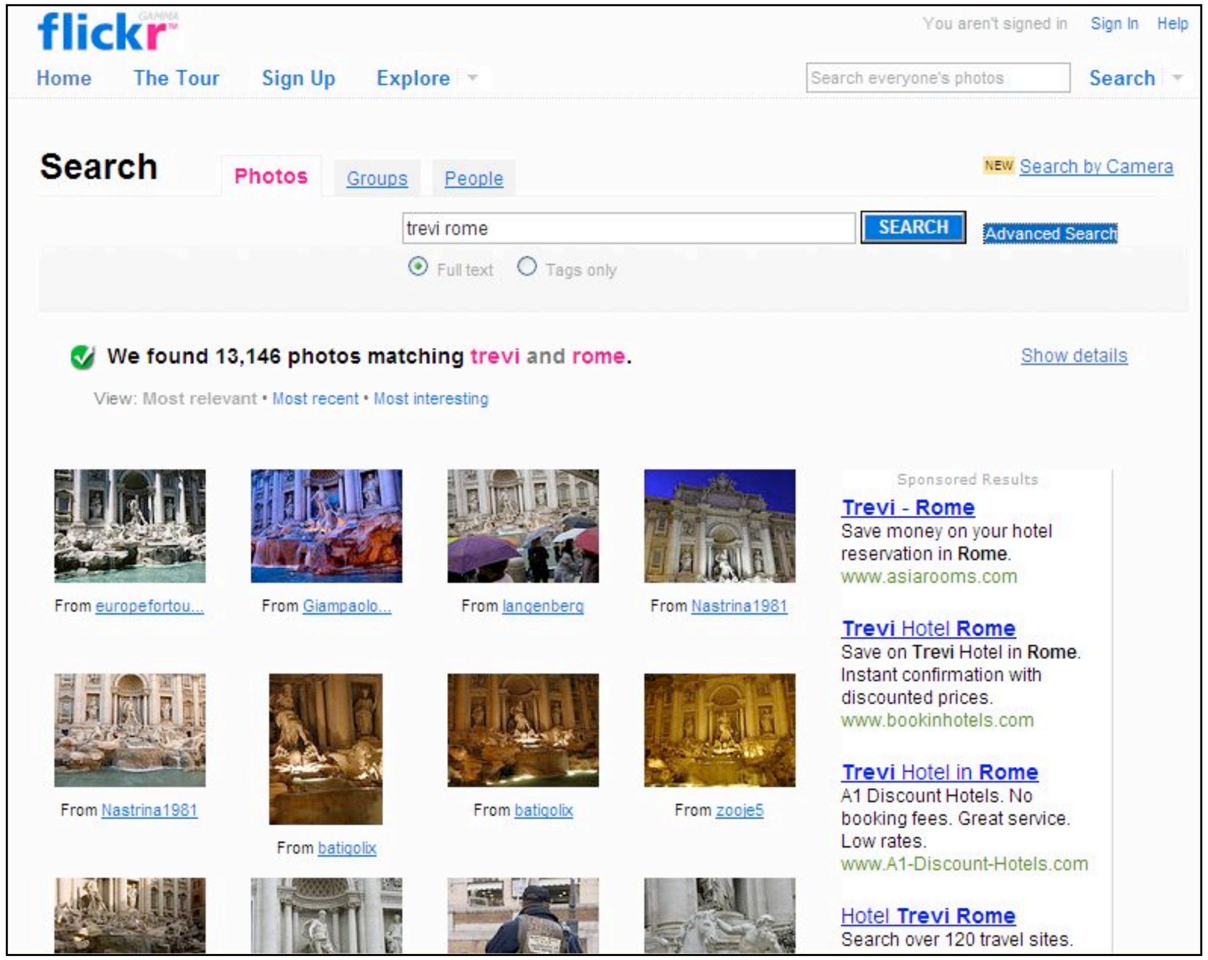


# Building Rome in a Day

| | |
|---|---|
| Sameer Agarwal | University of Washington |
| Noah Snavely | Cornell University |
| Ian Simon | University of Washington |
| Steven Seitz | University of Washington |
| Richard Szeliski | Microsoft Research |

# Photo Tourism



trevi rome

We found 13,146 photos matching trevi and rome.

From europefortou...

From Giampaolo...

From langenberg

From Nastrina1981

From Nastrina1981

From batigolix

From batigolix

From zooje5

Sponsored Results

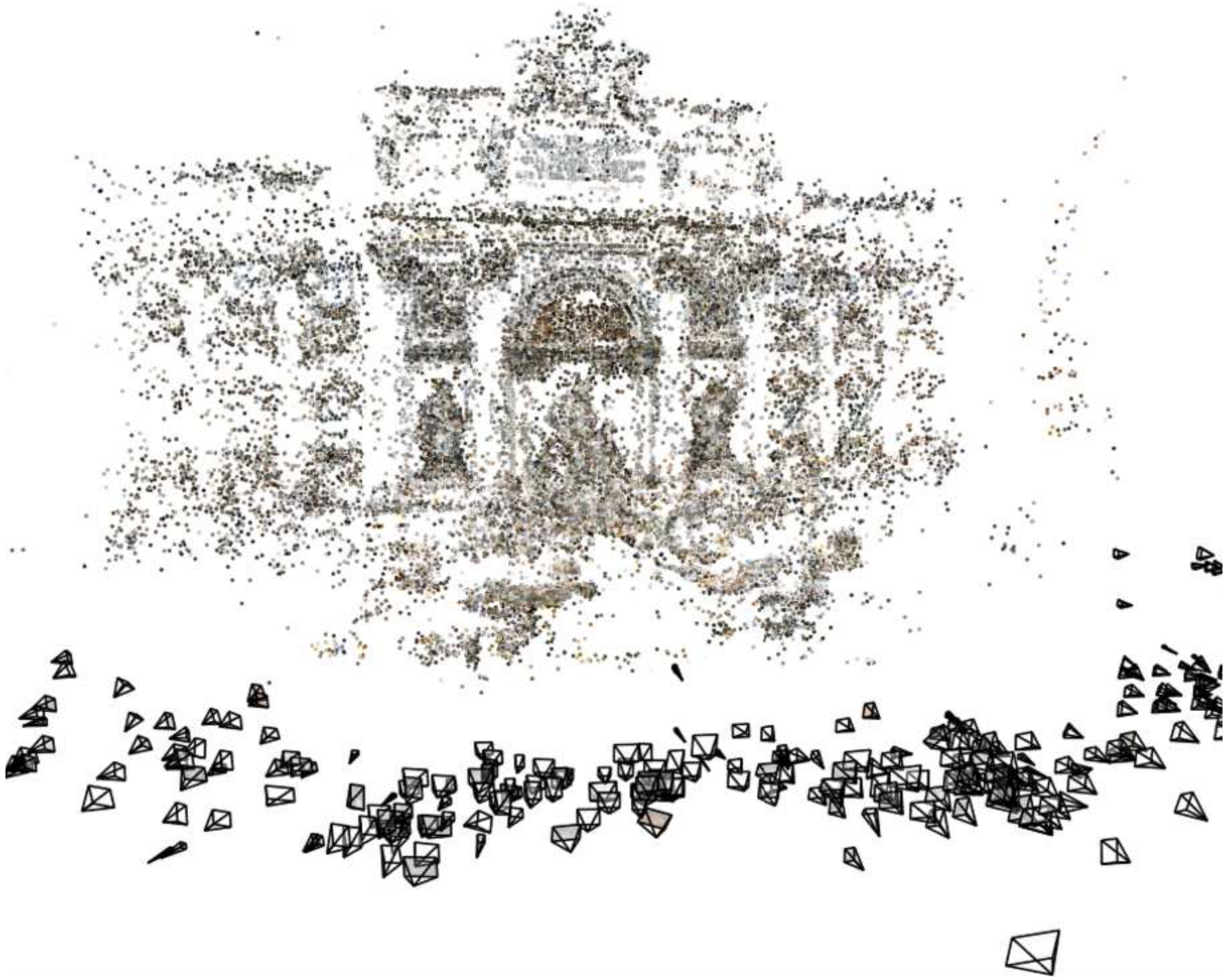
Trevi - Rome
Save money on your hotel reservation in Rome.
www.asiarooms.com

Trevi Hotel Rome
Save on Trevi Hotel in Rome. Instant confirmation with discounted prices.
www.bookinhotels.com

Trevi Hotel in Rome
A1 Discount Hotels. No booking fees. Great service. Low rates.
www.A1-Discount-Hotels.com

Hotel Trevi Rome
Search over 120 travel sites.

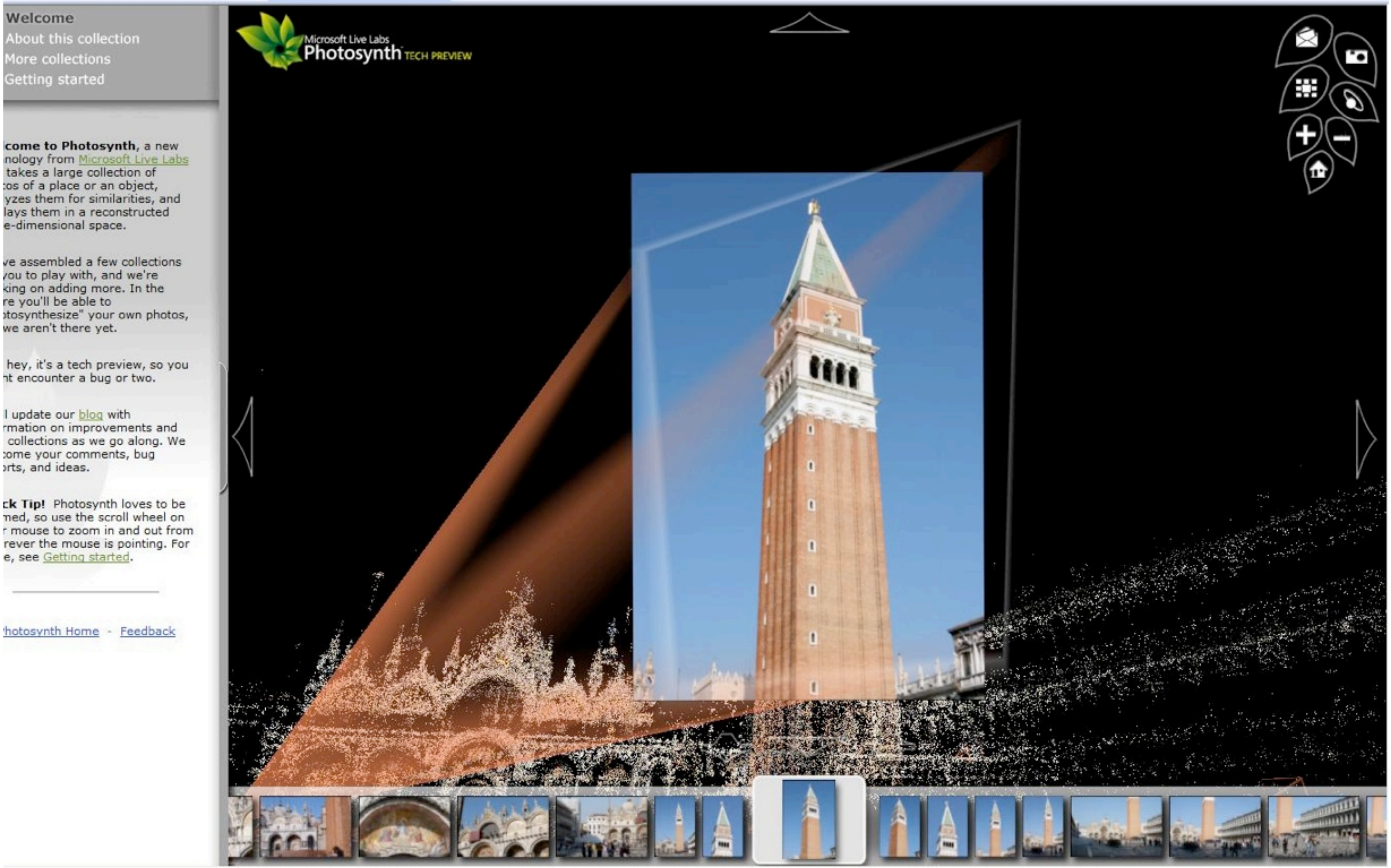
Images on the Internet



Computed 3D structure (360 views)

Snavely, Seitz, Szeliski, SIGGRAPH 2006

# Microsoft Photosynth



**lcome to Photosynth**, a new
nology from Microsoft Live Labs
takes a large collection of
tos of a place or an object,
yzes them for similarities, and
lays them in a reconstructed
e-dimensional space.

ve assembled a few collections
you to play with, and we're
king on adding more. In the
re you'll be able to
otosynthesize" your own photos,
we aren't there yet.

hey, it's a tech preview, so you
ht encounter a bug or two.

l update our blog with
rmation on improvements and
collections as we go along. We
come your comments, bug
orts, and ideas.

**ck Tip!** Photosynth loves to be
med, so use the scroll wheel on
r mouse to zoom in and out from
rever the mouse is pointing. For
e, see Getting started.

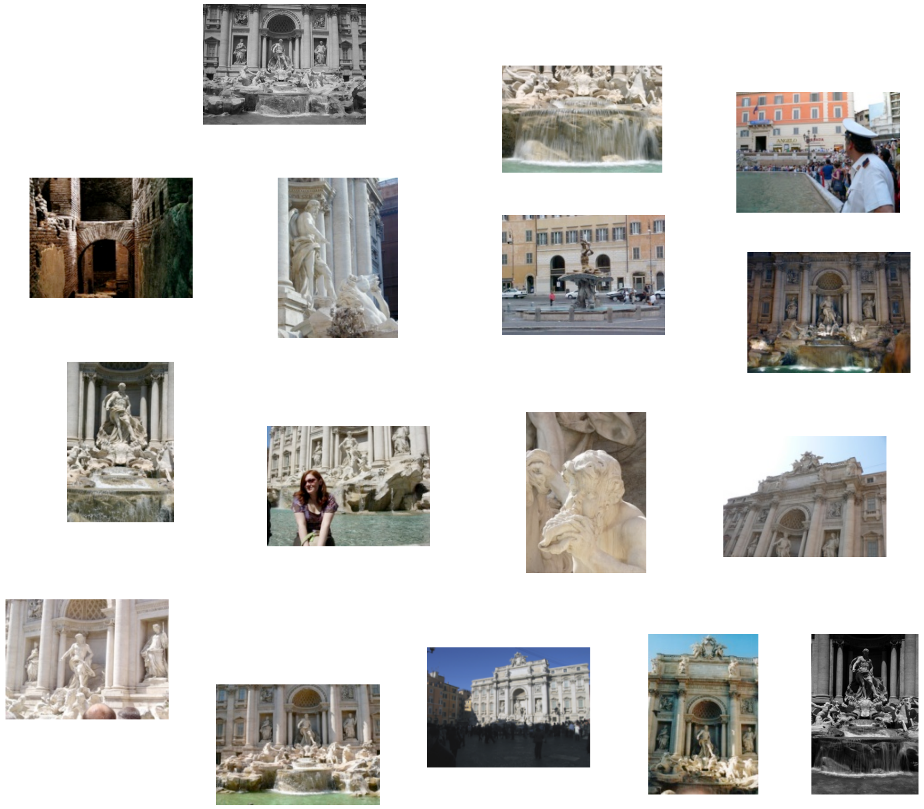
hotosynth Home - Feedback

# Cities on the web

| | Flickr | Picasa | images.google |
|---|---|---|---|
| **Venice** | 1.3M | 8M | 12M |
| **Rome** | 2.6M | 26M | 20M |
| **Tokyo** | 3.2M | 12M | 20M |
| **New York** | 6.5M | 41M | 290M |
| **London** | 7.2M | 41M | 89M |

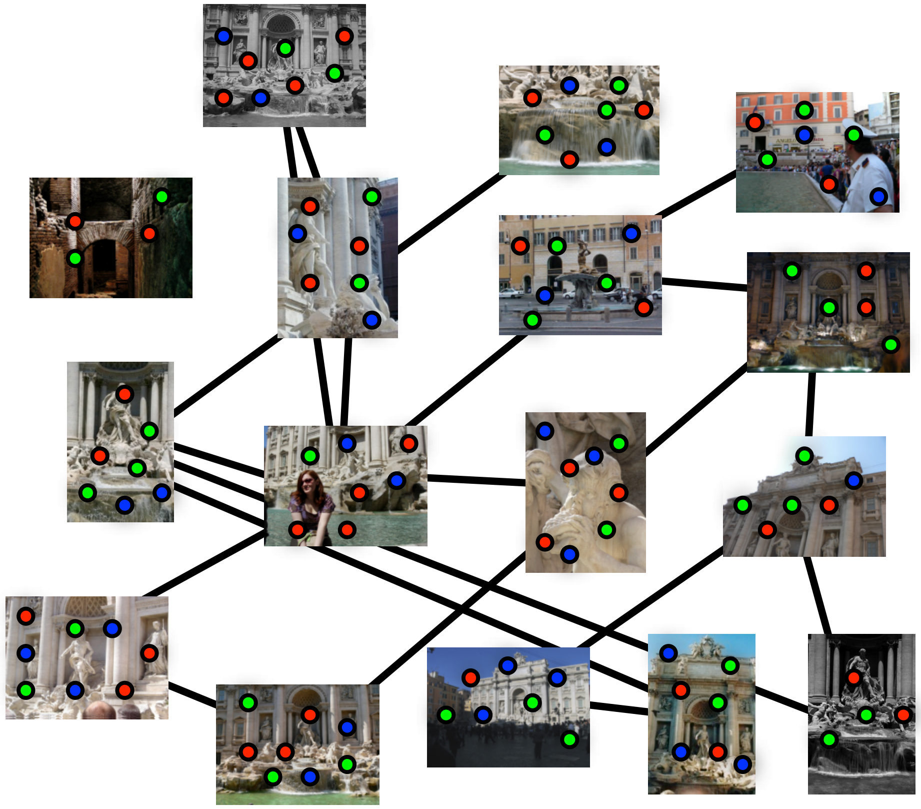
# The Task

- Download a million images of Rome
- Match the images
- Build a 3D model of the city



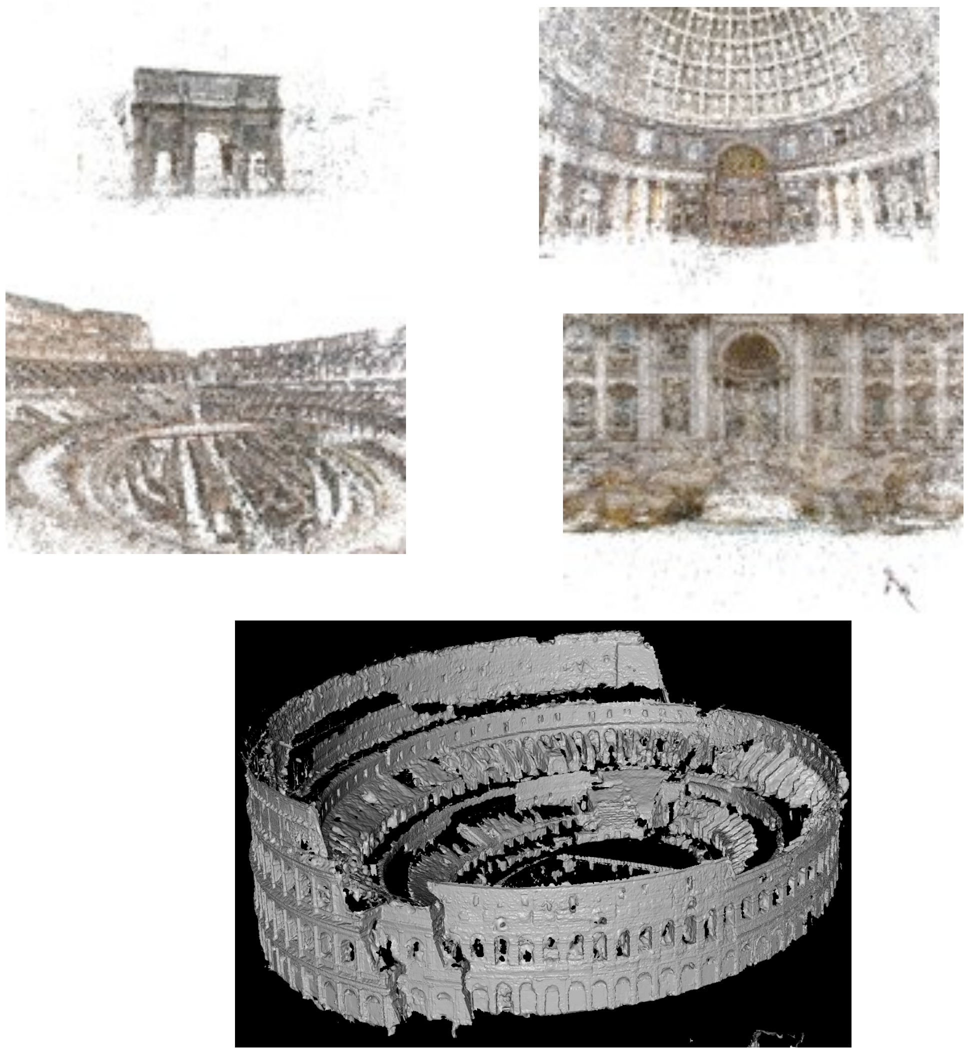
# The Task

- Download a million images of Rome
- Match the images
- Build a 3D model of the city



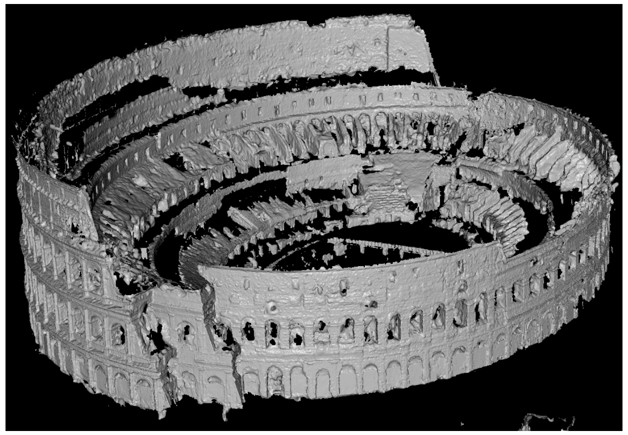
# The Task

- Download a million images of Rome
- Match the images
- Build a 3D model of the city



# The Task

- Download a million images of Rome
- Match the images
- Build a 3D model of the city



Do all of the above in a fully distributed manner on a 1000 node cluster in 24 hours.

# Why?

- Because we can.
- Interiors, high level of geometric detail, texture maps.
- Better models for Google/Virtual Earth, GPS, virtual sets for movie production.
- Historical preservation.
- Urban geography.
- Games set in the real world, Photocity, Grand Theft Auto “Roma”.

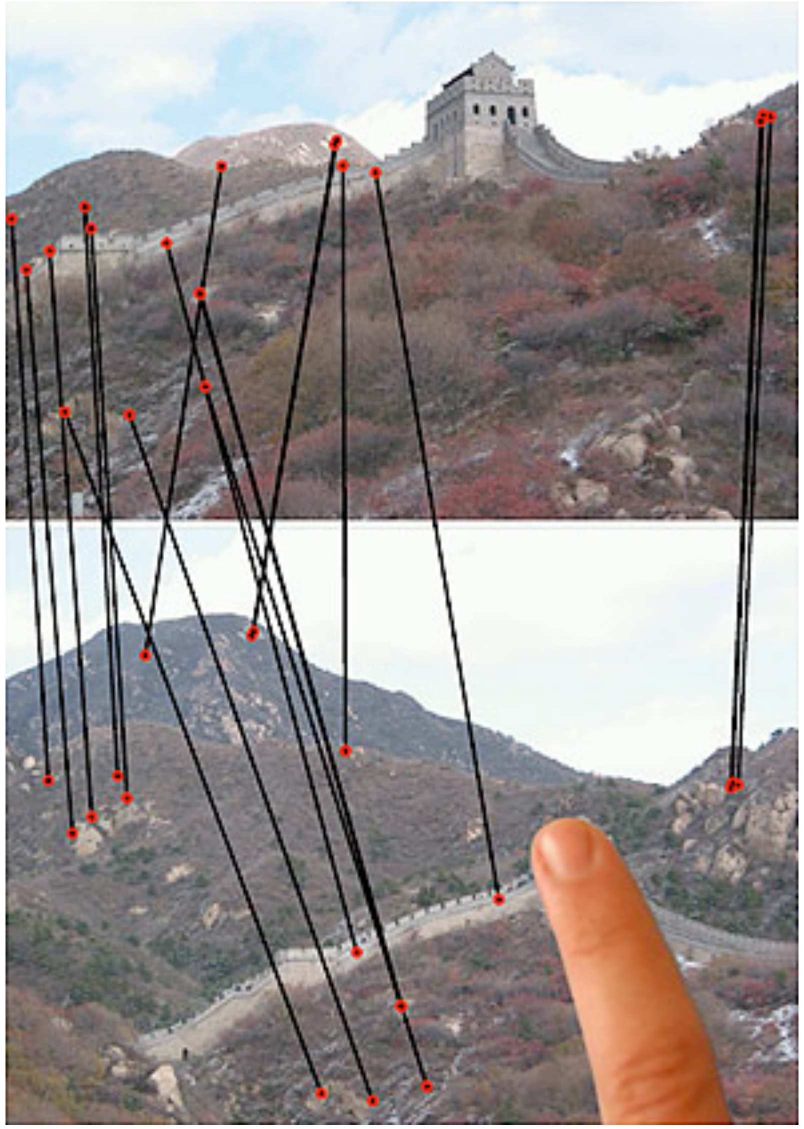
# Our Approach

1. Scrape images
2. Extract Features (SIFT)
3. Match Images
4. Reconstruct sparse image set (Skeletal Sets)
5. Reconstruct full image set

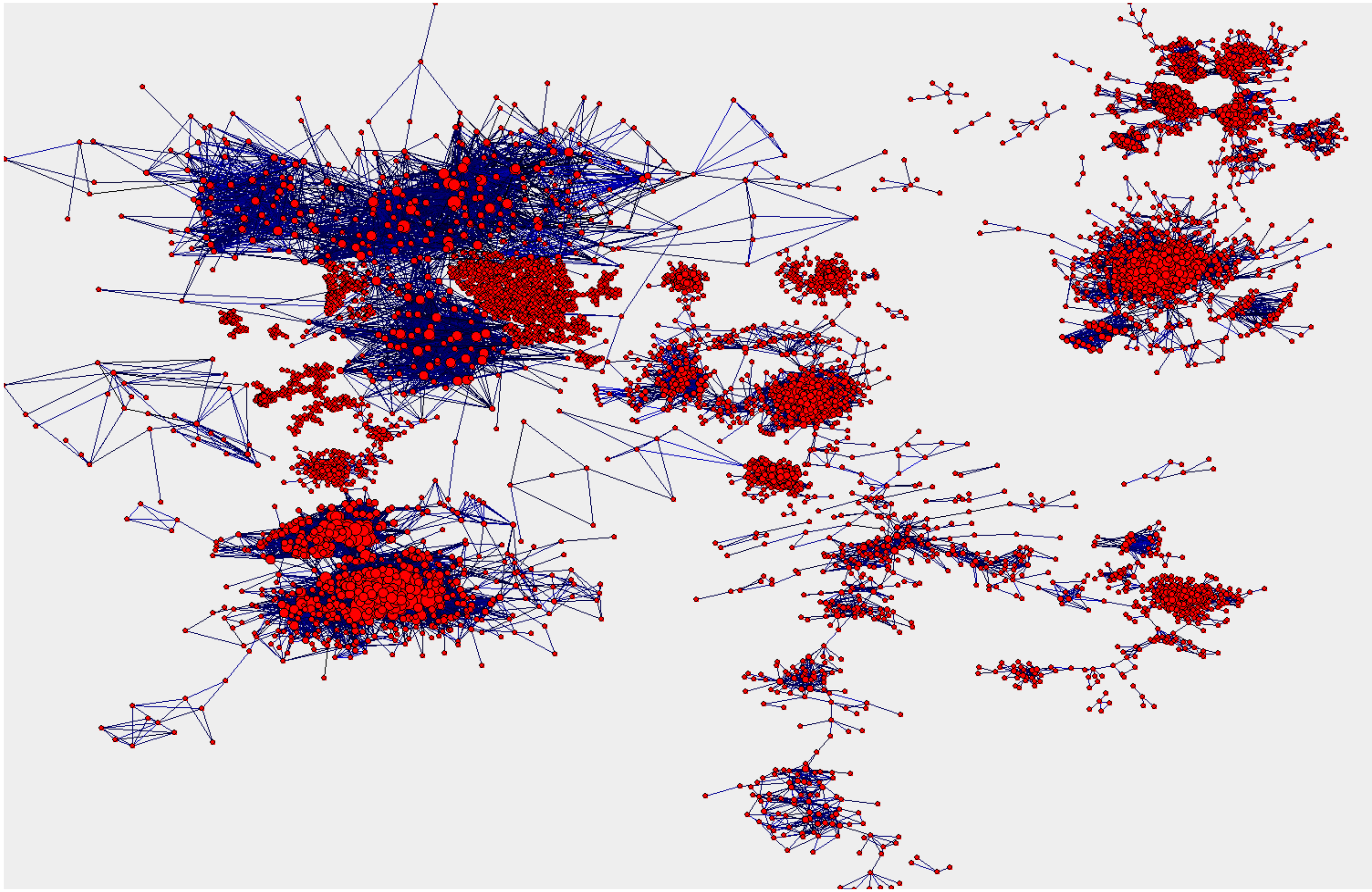
# Image Matching

Find points across images which correspond to the same point in the world.

- All pairs matching is data parallel, but expensive in CPU and network bandwidth (~10TB).
- 0.5 Trillion pairwise comparisons.
- 10k matches/sec = 1.5 years.

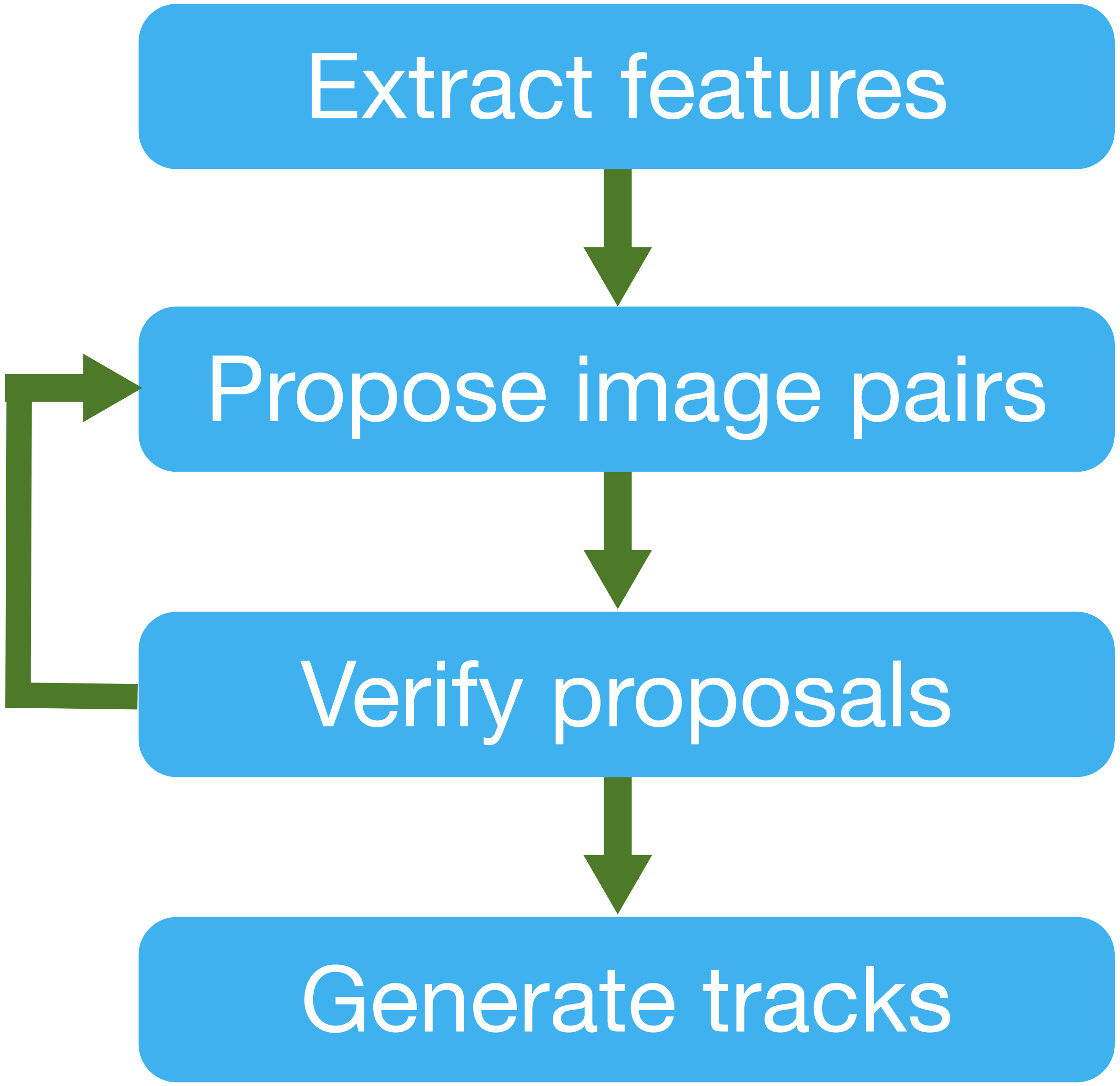


# Rome



# Matching Algorithm

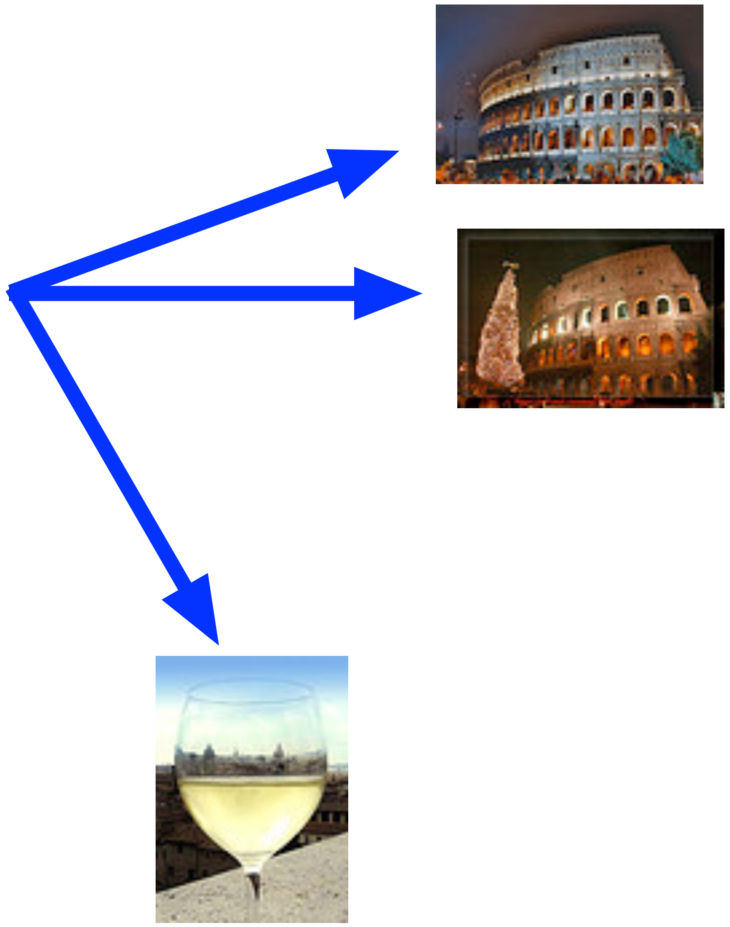
- Multi round propose and verify scheme
  - 2 rounds based on whole image similarity.
  - 4 rounds based on query expansion
- Verification = SIFT feature matching + RANSAC.



# Whole Image Similarity

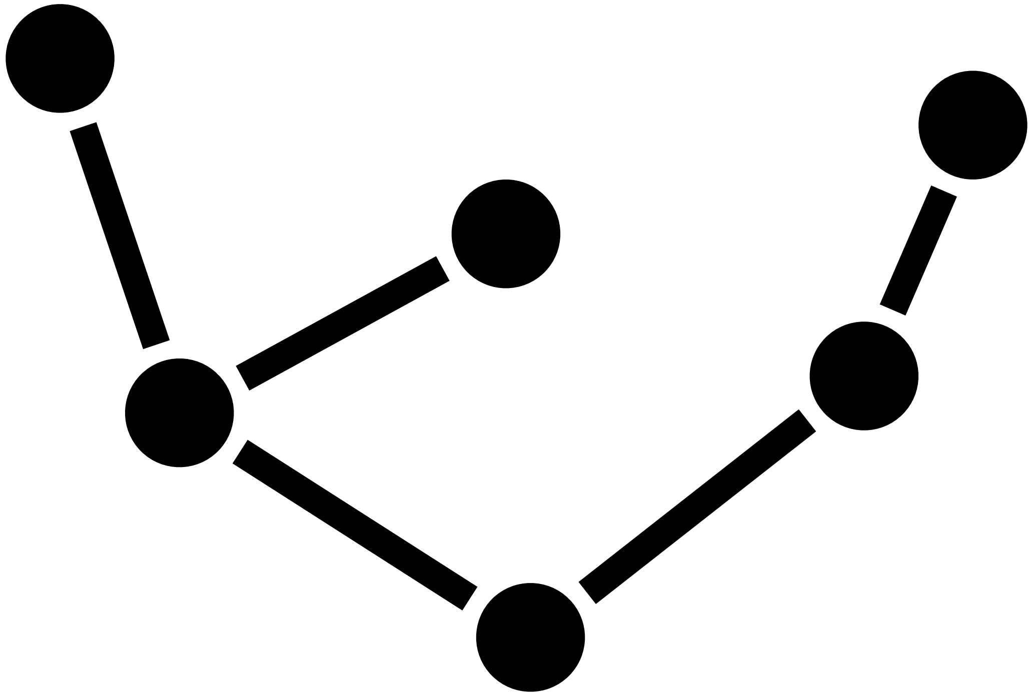
Text retrieval inspired approach.

- Represent images as high dimensional vectors using a vocabulary tree.
- Inner product between of vectors is the similarity between images.
- Top k scoring images are potential matches.



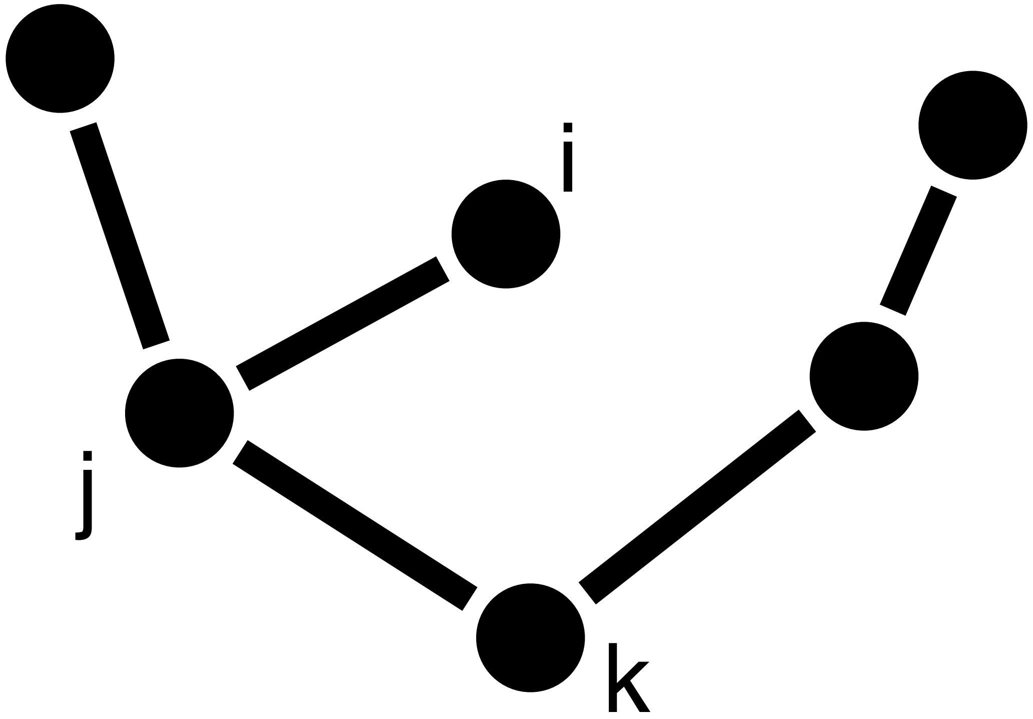
Nister and Stewenius, CVPR 2006

# Query Expansion



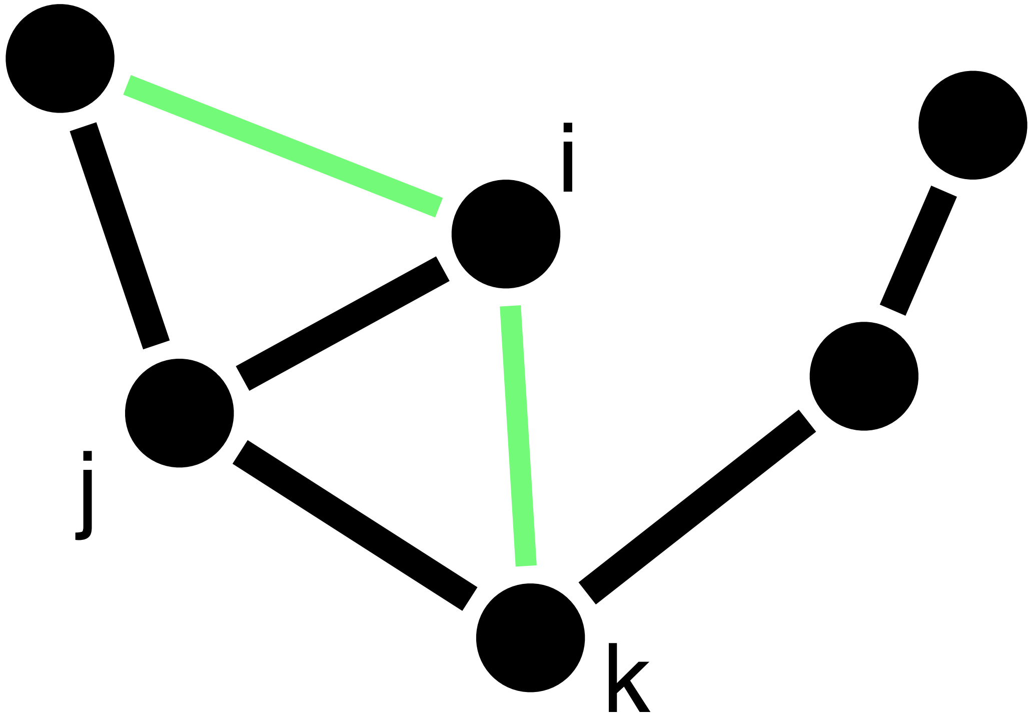
Chum et al, ICCV 2007

# Query Expansion



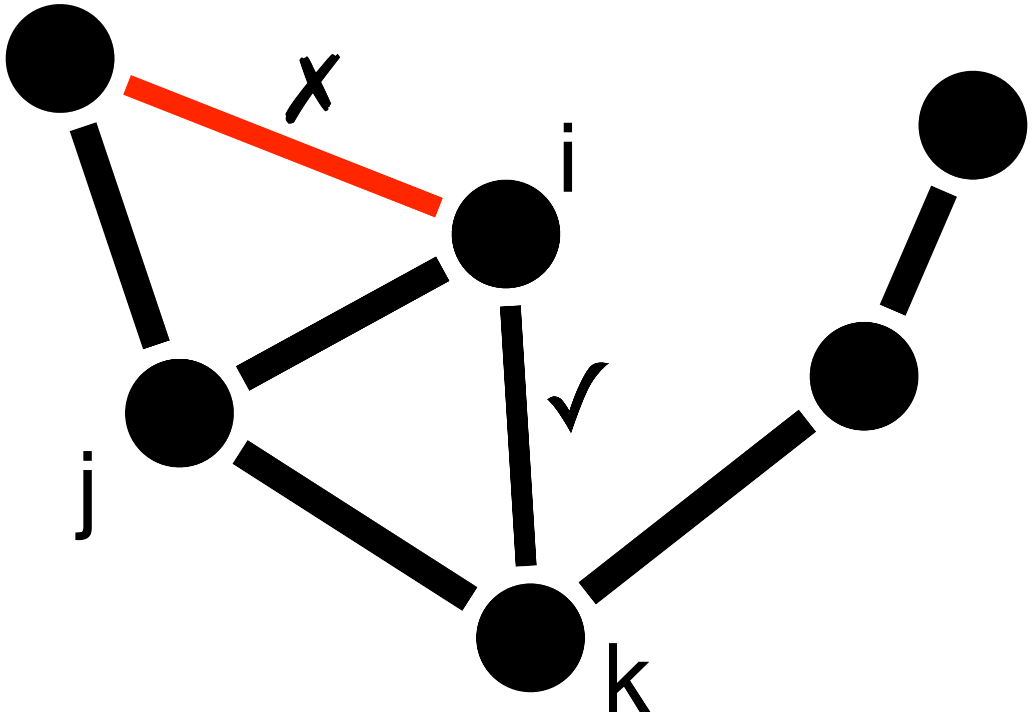
Chum et al, ICCV 2007

# Query Expansion



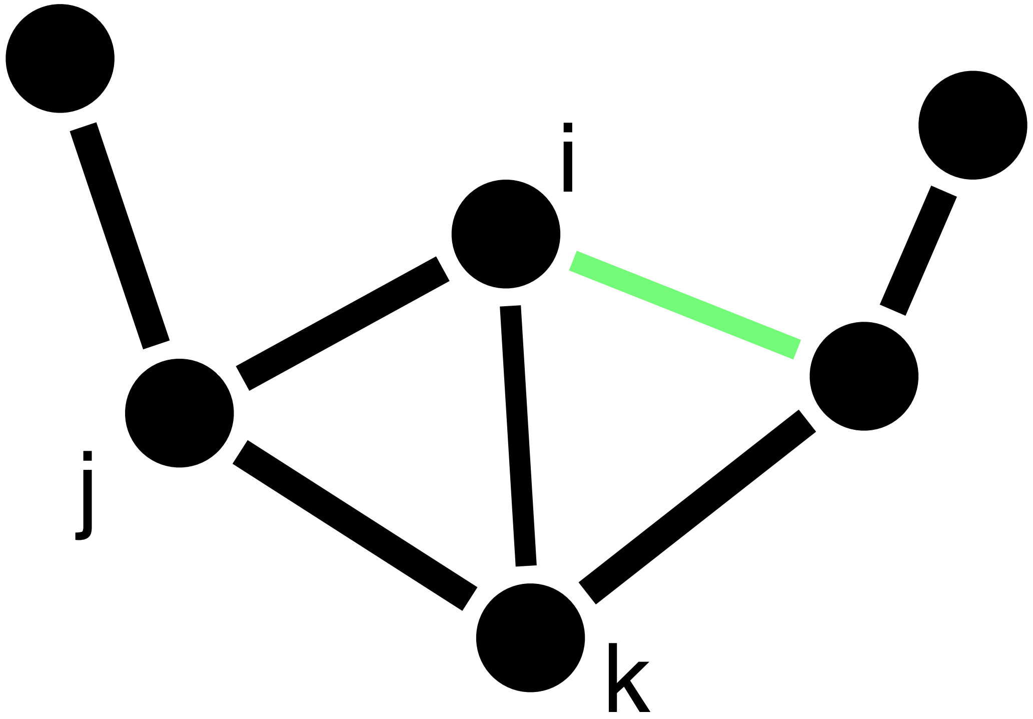
Chum et al, ICCV 2007

# Query Expansion



Chum et al, ICCV 2007

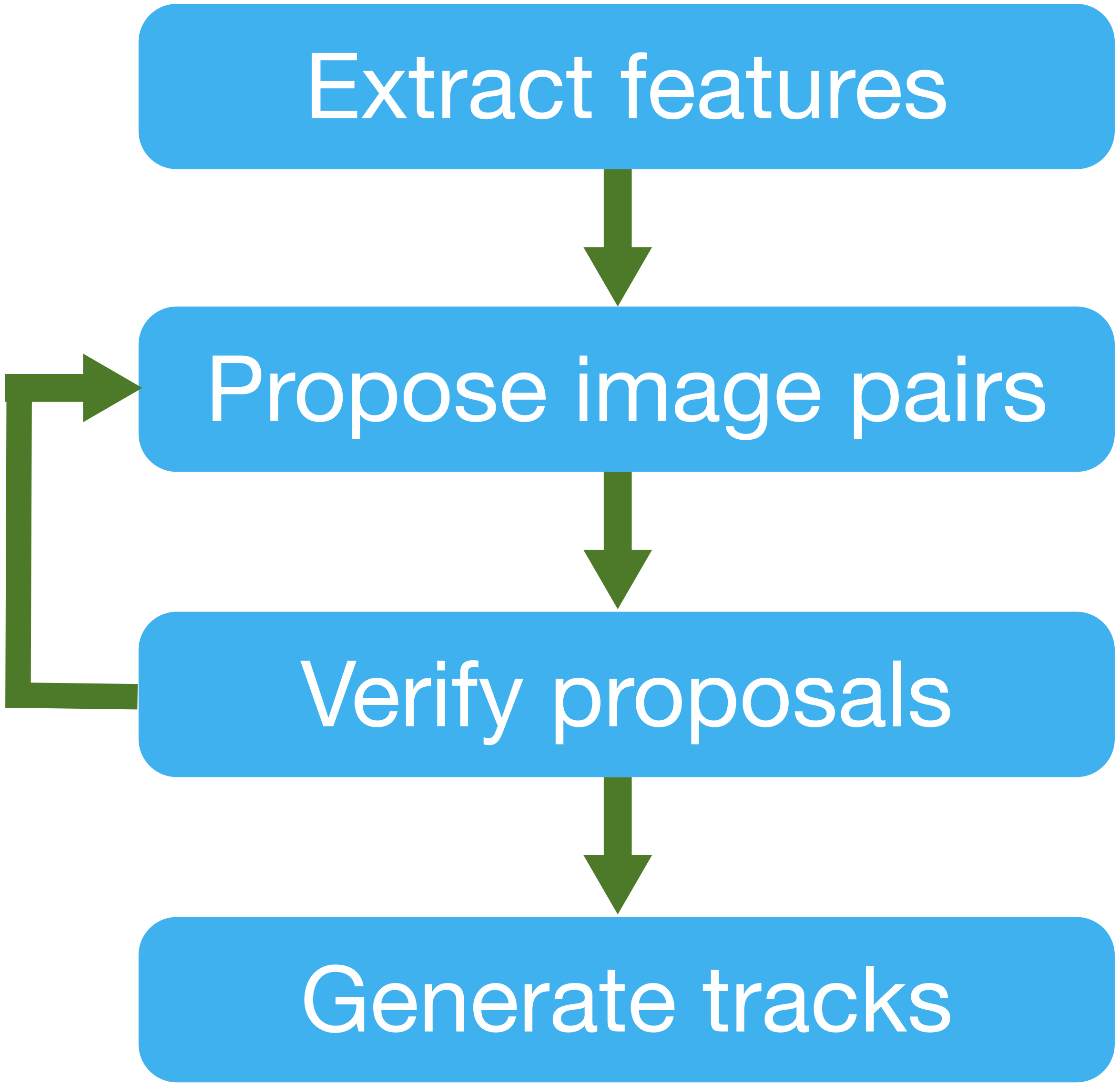
# Query Expansion



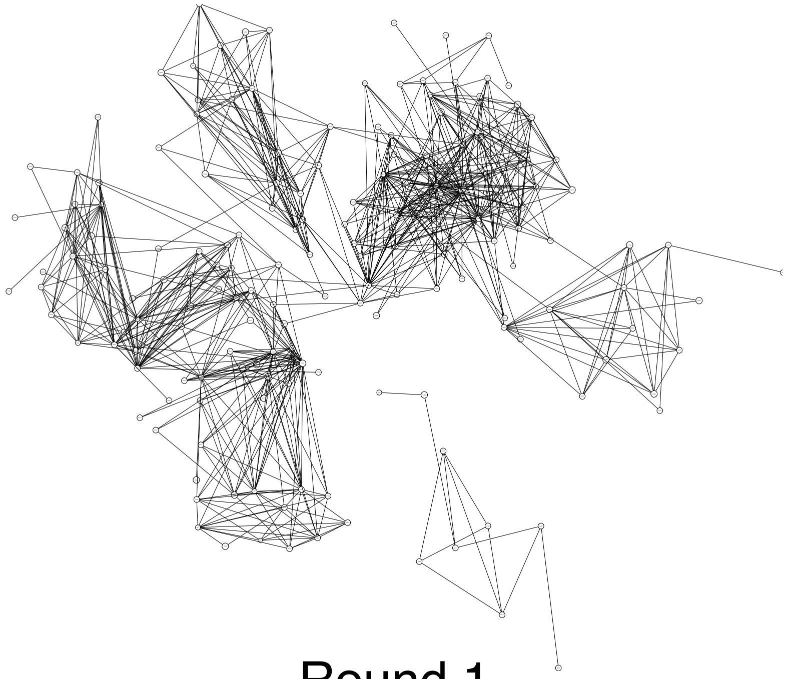
Chum et al, ICCV 2007

# Matching Algorithm

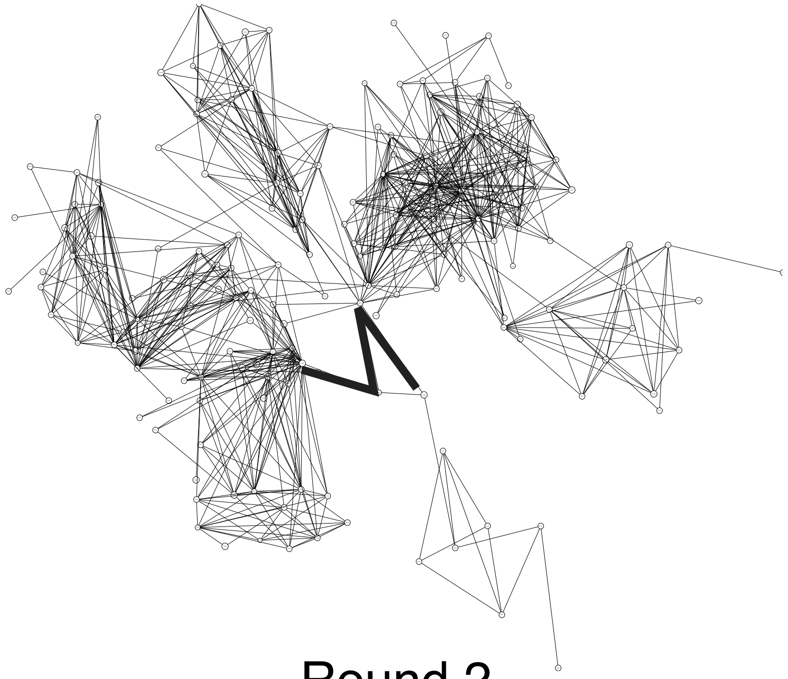
- Multi round propose and verify scheme
  - 2 rounds based on whole image similarity.
  - 4 rounds based on query expansion
- Verification = SIFT feature matching + RANSAC.



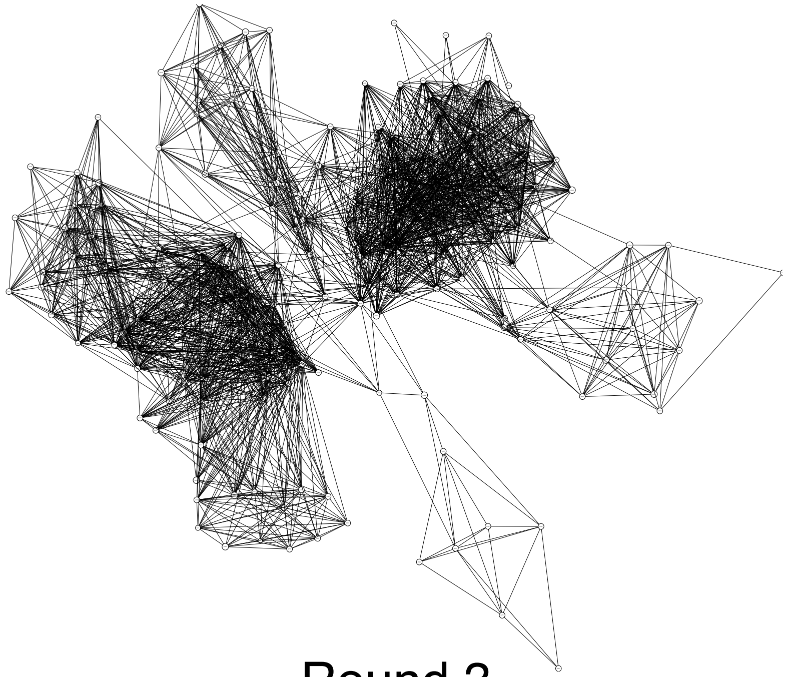
# Matching Progress



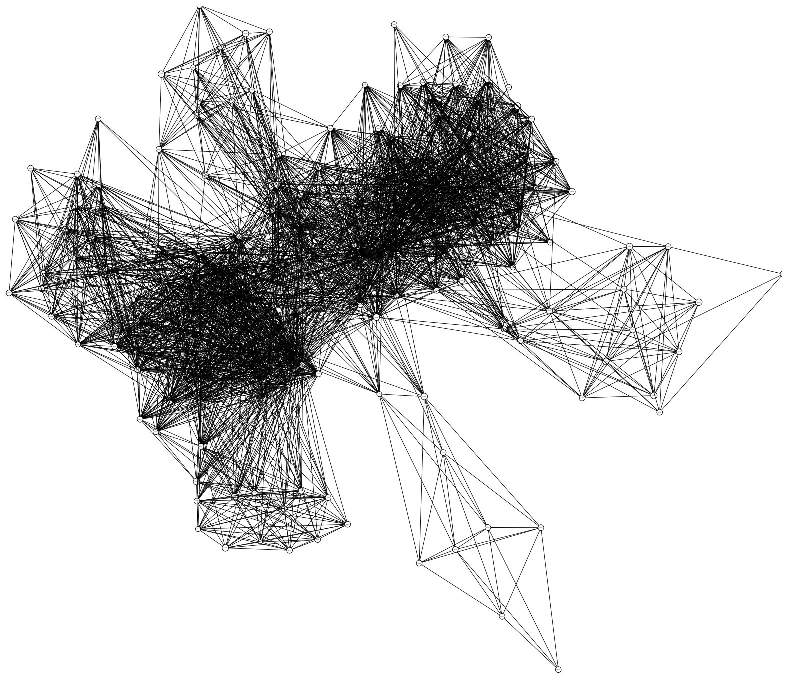
Round 1



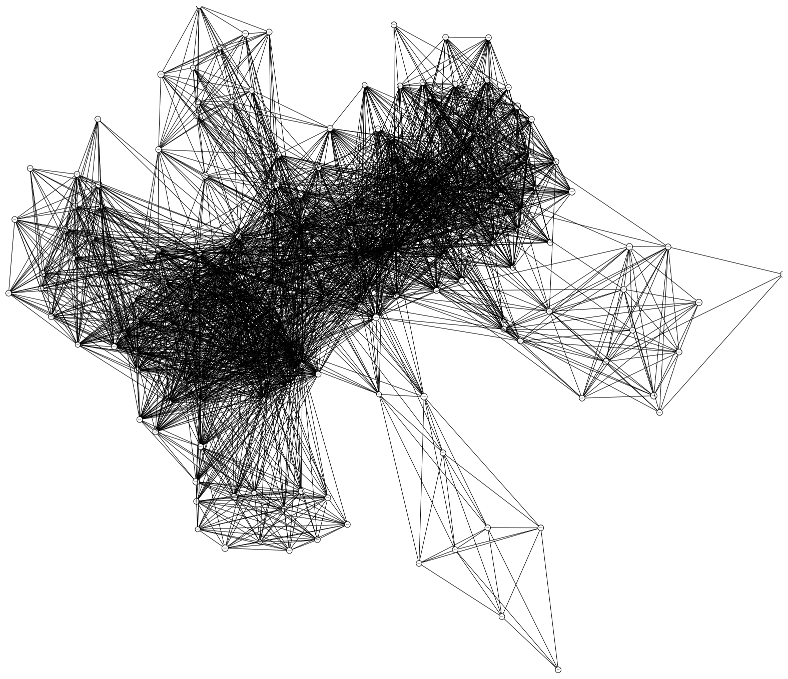
Round 2



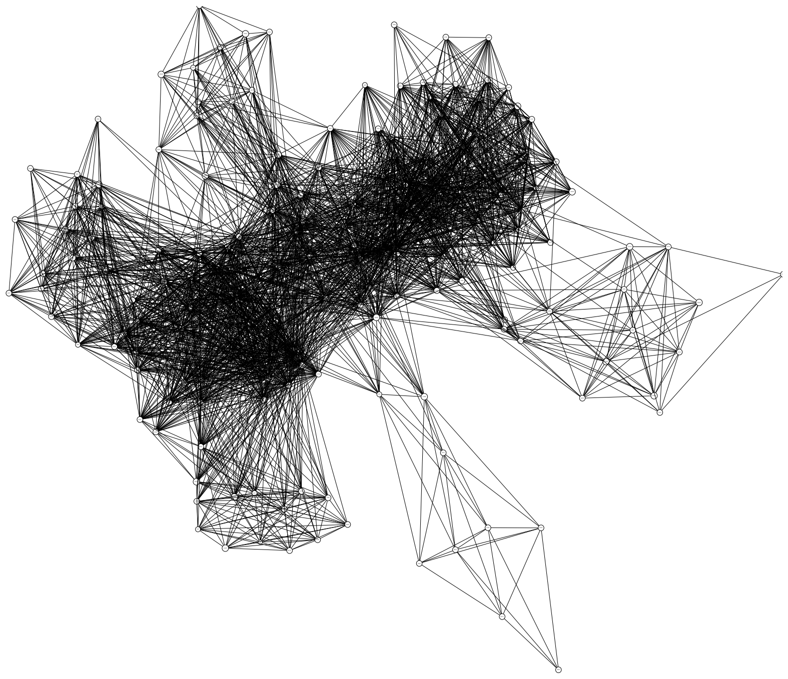
Round 3



Round 4



Round 5



Round 6

# Matching Results

| Dataset | Size | Matches possible | Matches Tried | Matches Found | Time |
|---|---|---|---|---|---|
| Rome | 150K | 11.2 Billon | 8.8M | 2.7M | 9 hrs |

# Matching Statistics

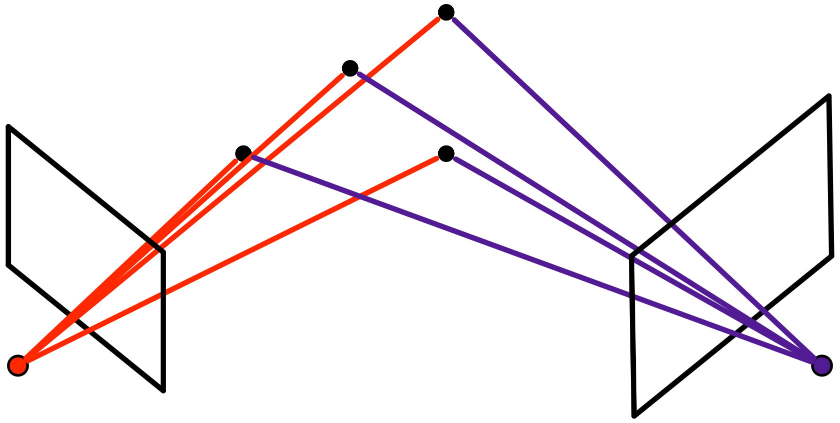
| Dataset | Size | Matches possible | Matches Tried | Matches Found | Time |
|---|---|---|---|---|---|
| Dubrovnik | 58K | 1.6 Billion | 2.6M | 0.5M | 5 hrs |
| Rome | 150K | 11.2 Billion | 8.8M | 2.7M | 13 hrs |
| Venice | 250K | 31.2 Billion | 35.5M | 6.2M | 27 hrs |

# Reconstruction

1. Choose two images to seed the reconstruction.

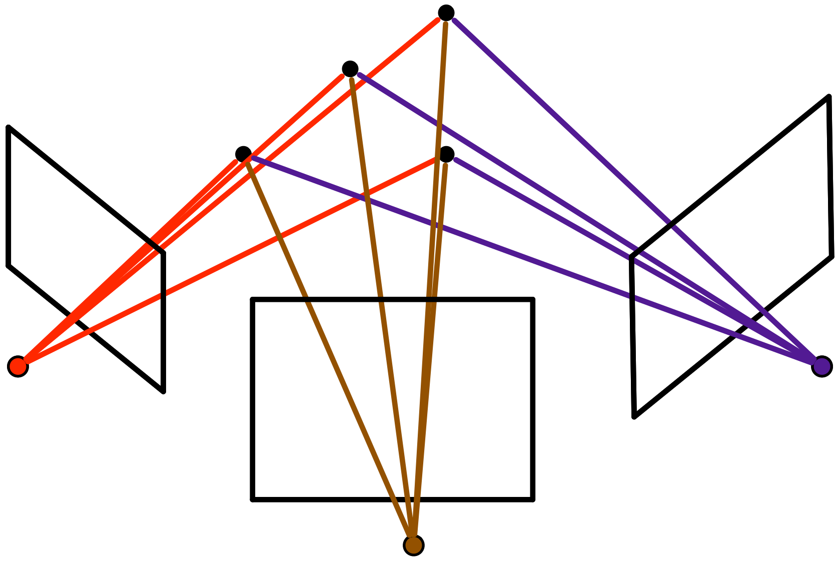
# Reconstruction

1. Choose two images to seed the reconstruction.



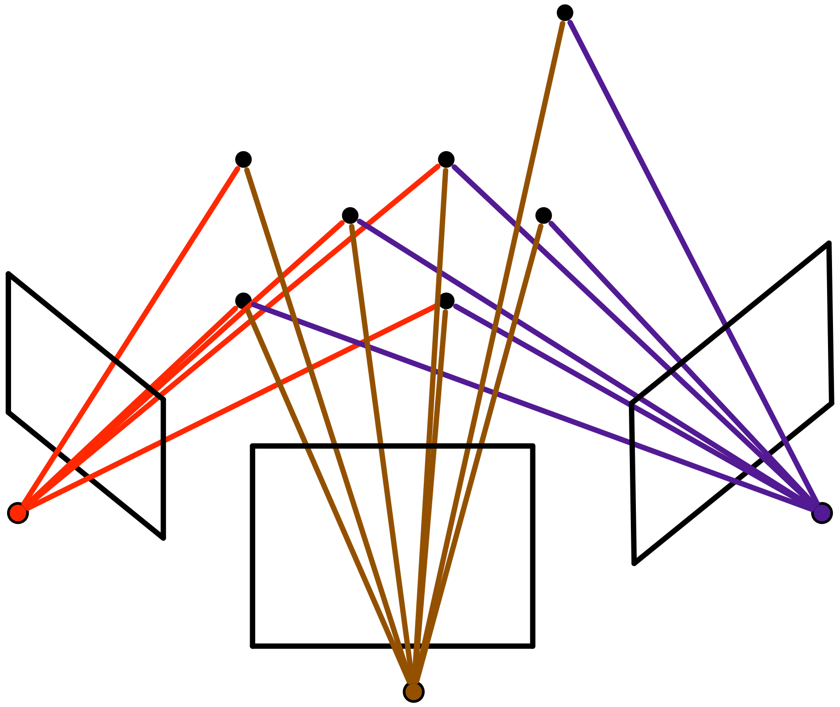
# Reconstruction

1. Choose two images to seed the reconstruction.
2. Add cameras using pose estimation.



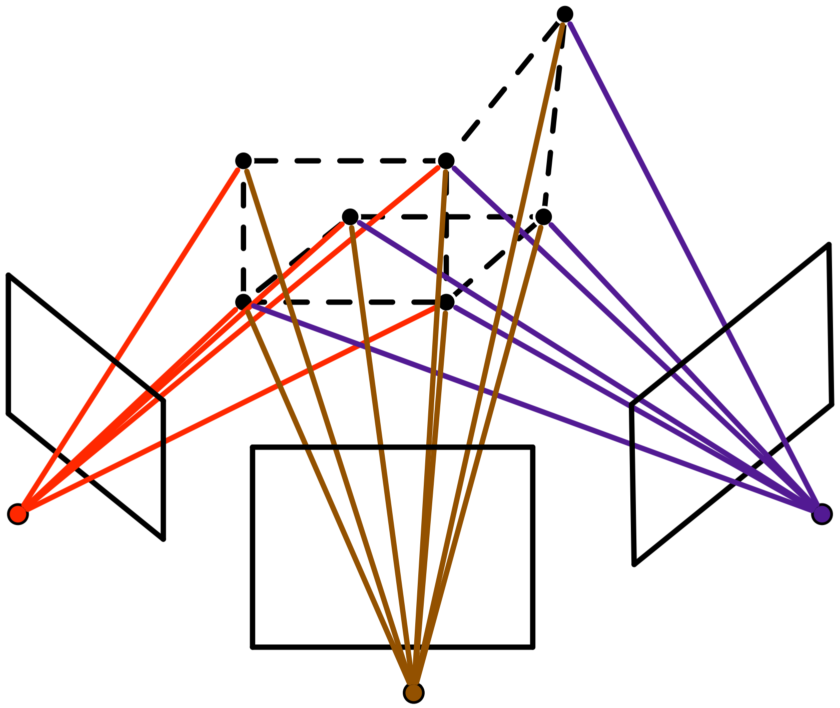
# Reconstruction

1. Choose two images to seed the reconstruction.
2. Add cameras using pose estimation.
3. Add 3d points via triangulation.



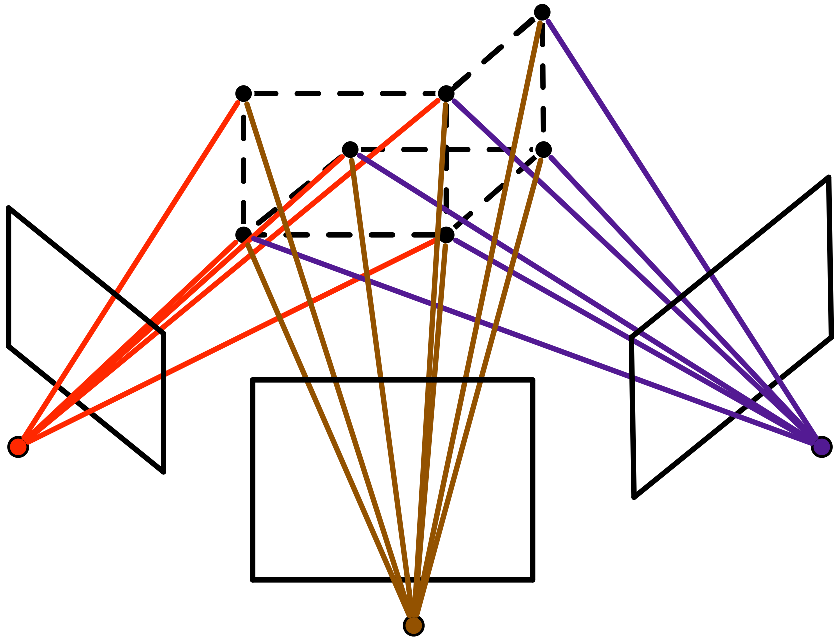
# Reconstruction

1. Choose two images to seed the reconstruction.
2. Add cameras using pose estimation.
3. Add 3d points via triangulation.



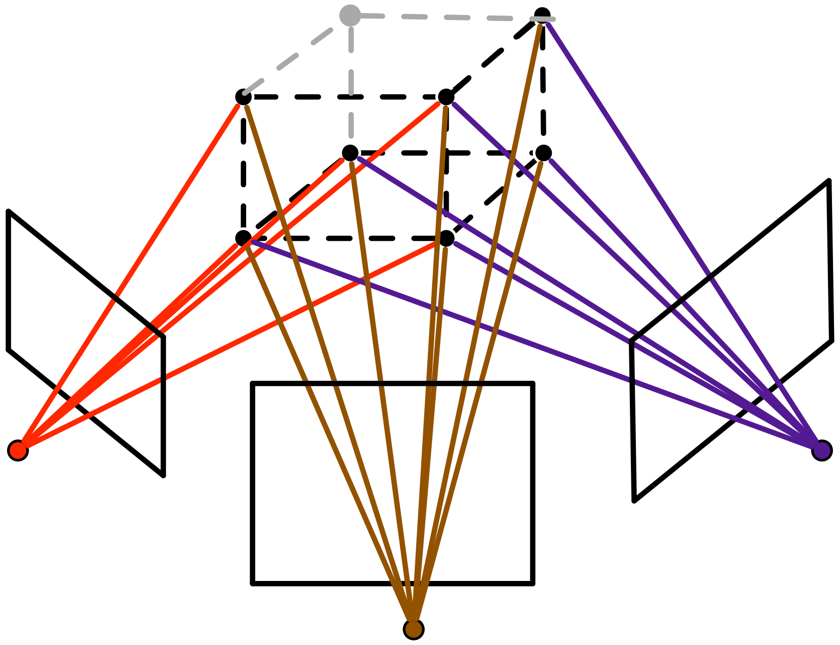
# Reconstruction

1. Choose two images to seed the reconstruction.
2. Add cameras using pose estimation.
3. Add 3d points via triangulation.
4. Non-linear refinement.



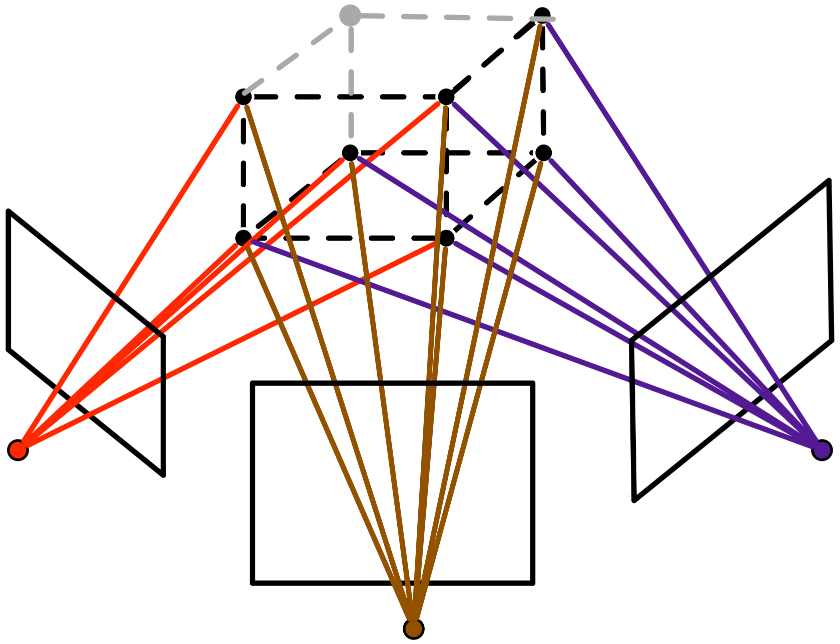
# Reconstruction

1. Choose two images to seed the reconstruction.
2. Add cameras using pose estimation.
3. Add 3d points via triangulation.
4. Non-linear refinement.
5. Goto step 2.

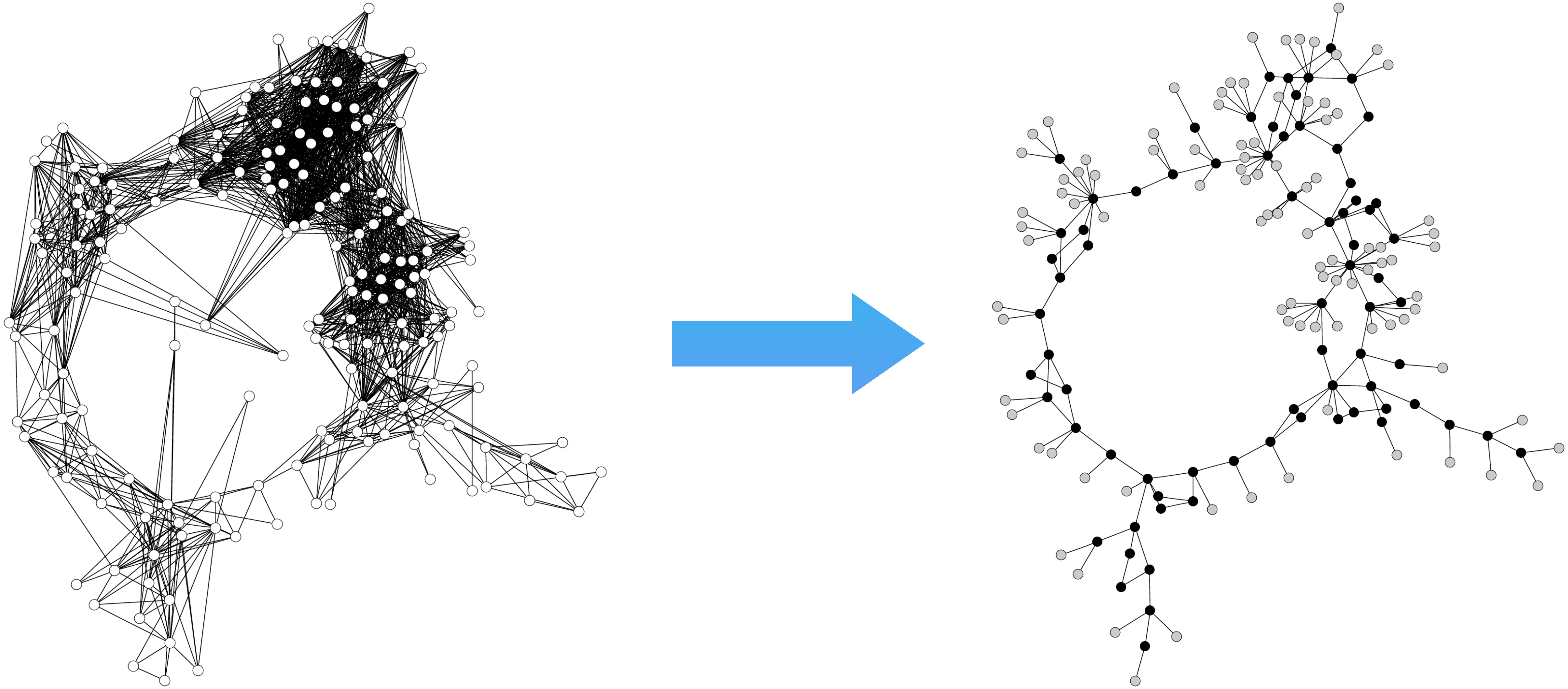


# Reconstruction

1. Choose two images to seed the reconstruction.
2. Add cameras using pose estimation.
3. Add 3d points via triangulation.
4. Non-linear refinement.
5. Goto step 2.



Effective but very slow



Stonehenge

Skeletal Set

Snavely et al. CVPR 07

# Bundle Adjustment

- State of the art (SBA) is not fast or scalable enough for our needs.
- Introducing BAng!
    - Exact step algorithm solves the Schur complement  system using a Sparse Direct solver.
    - Inexact step algorithm solves the normal equations using a Preconditioned Conjugate gradient solver.
- 10x or more faster than SBA.
- Used to solve a problem with, 14M variables and 54M terms.

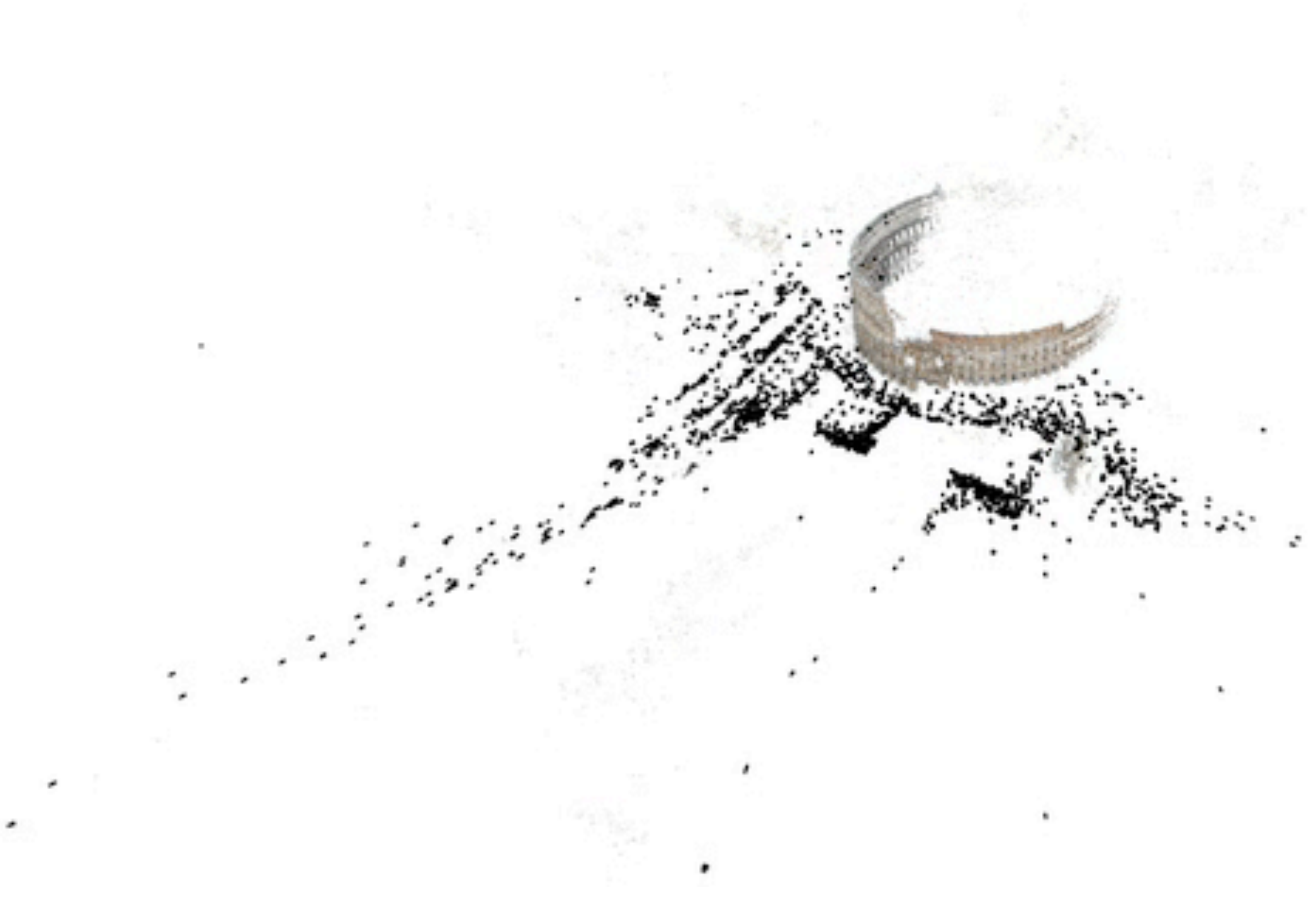
**Yours to use soon!**

# System architecture

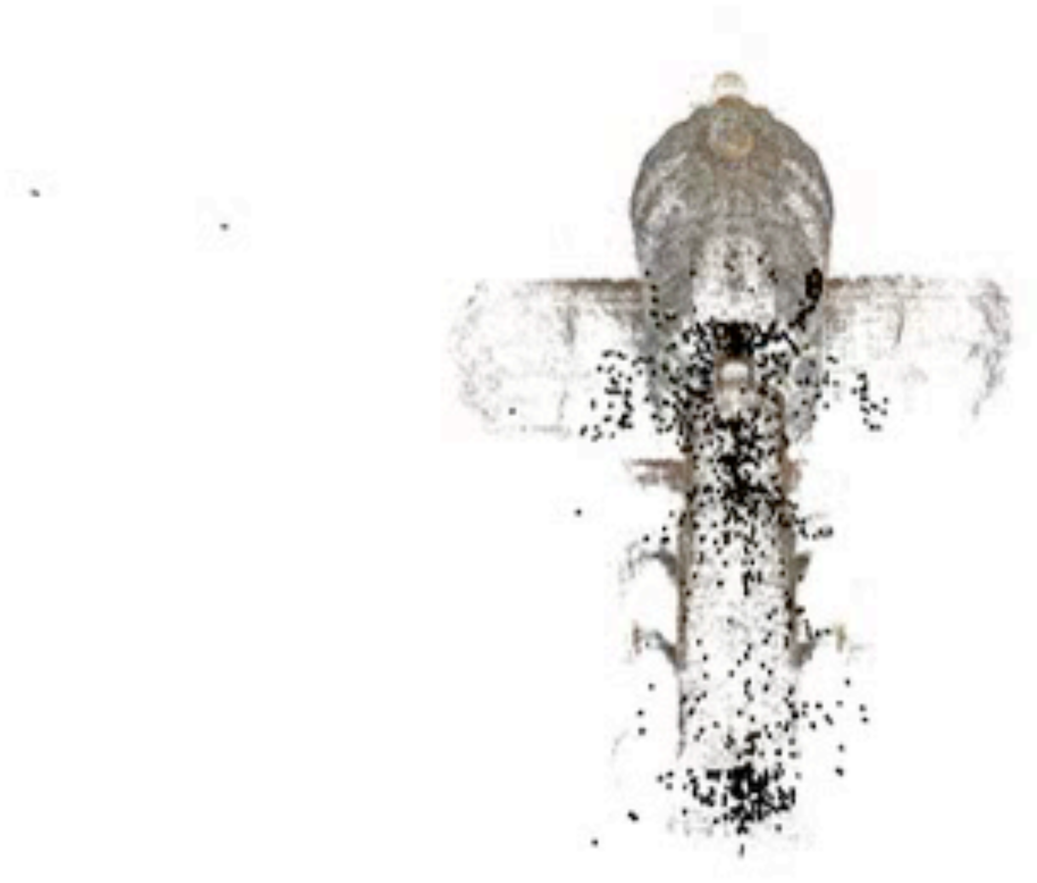
- Two layer system.
- Application agnostic, Python based distributed computing engine.
- Aimed at data intensive applications, with extensive support for caching.
- Small OS dependent core, easily portable.
- Matching system is an application written on top (Python & C++).

# Results

# Rome: Colosseum



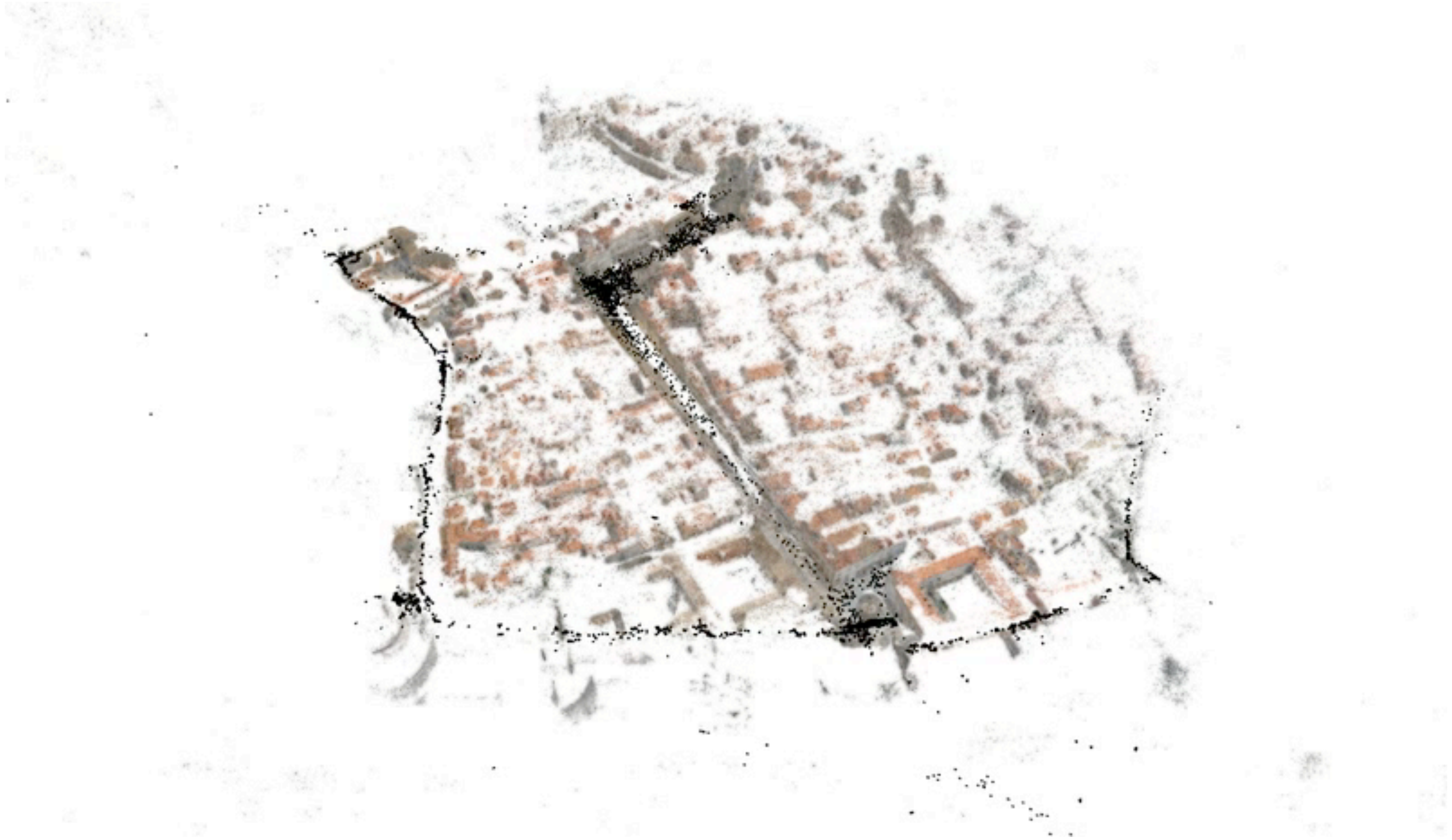
# Rome: St. Peters Basilica



# Venice: The Canal

# Venice: San Marco

# Dubrovnik

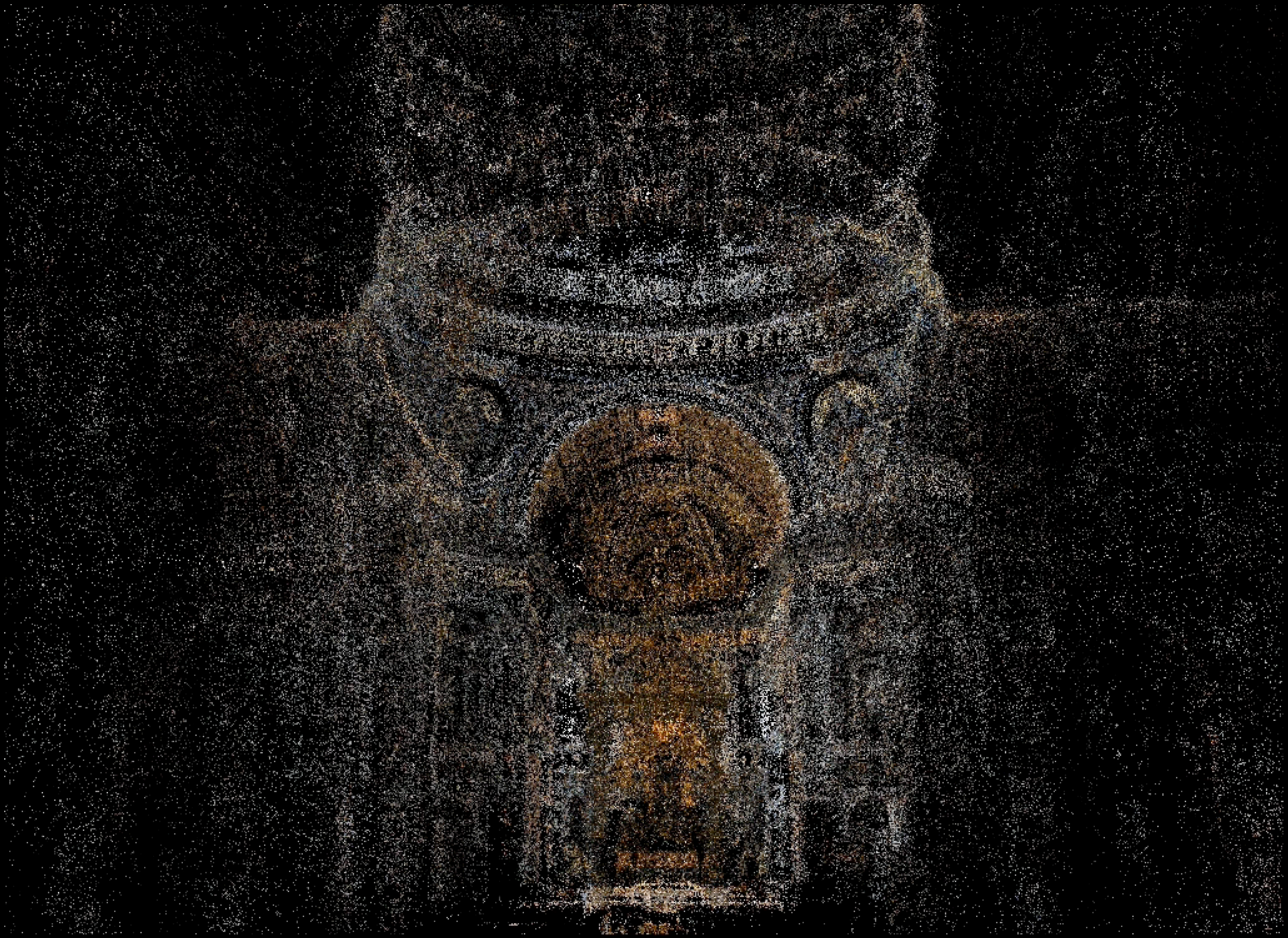


# Reconstruction Statistics

| Dataset | Images | Largest component | Skeletal Set | Time (hrs) |
|---|---|---|---|---|
| Dubrovnik | 58K | 4,619 | 977 | 18 |
| Rome | 150K | 2,106 | 254 | 8 |
| Venice | 250K | 14,079 | 1,801 | 38 |

# The road ahead...

- Integrate aerial and streetview imagery.
- Parallel, hierarchical reconstruction algorithm.
- New visualization tools for all this data.
- Dense models...



Furukawa et al., in preparation

# Acknowledgments

Microsoft Research

Office of Naval Research

Google

SPAWAR

National Science Foundation

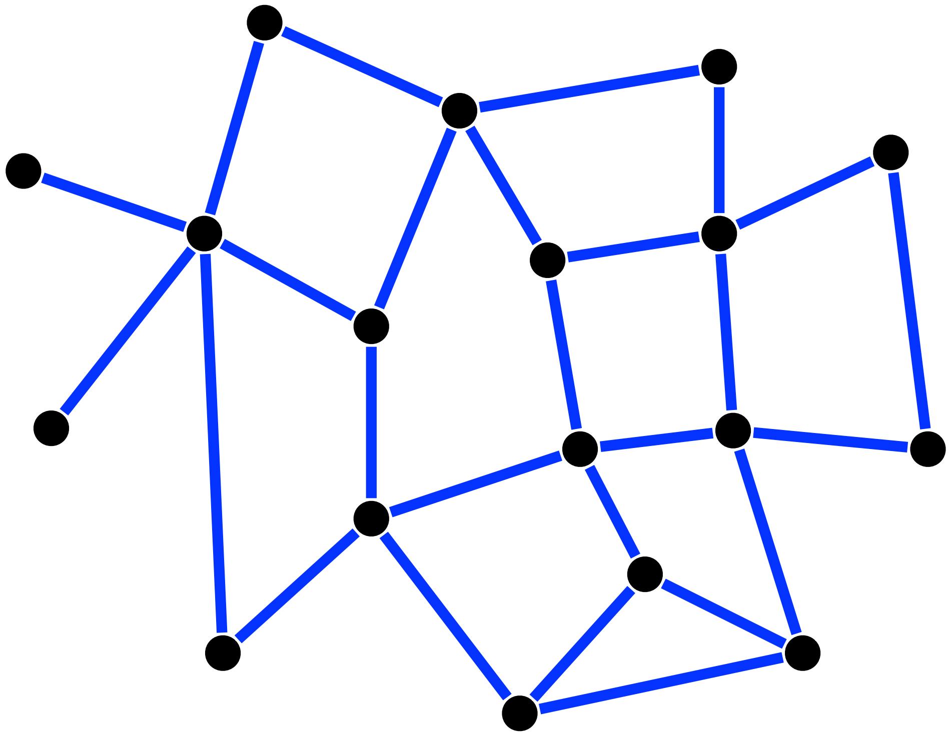
# Thank you

Please come to the poster session and explore the 3d models.

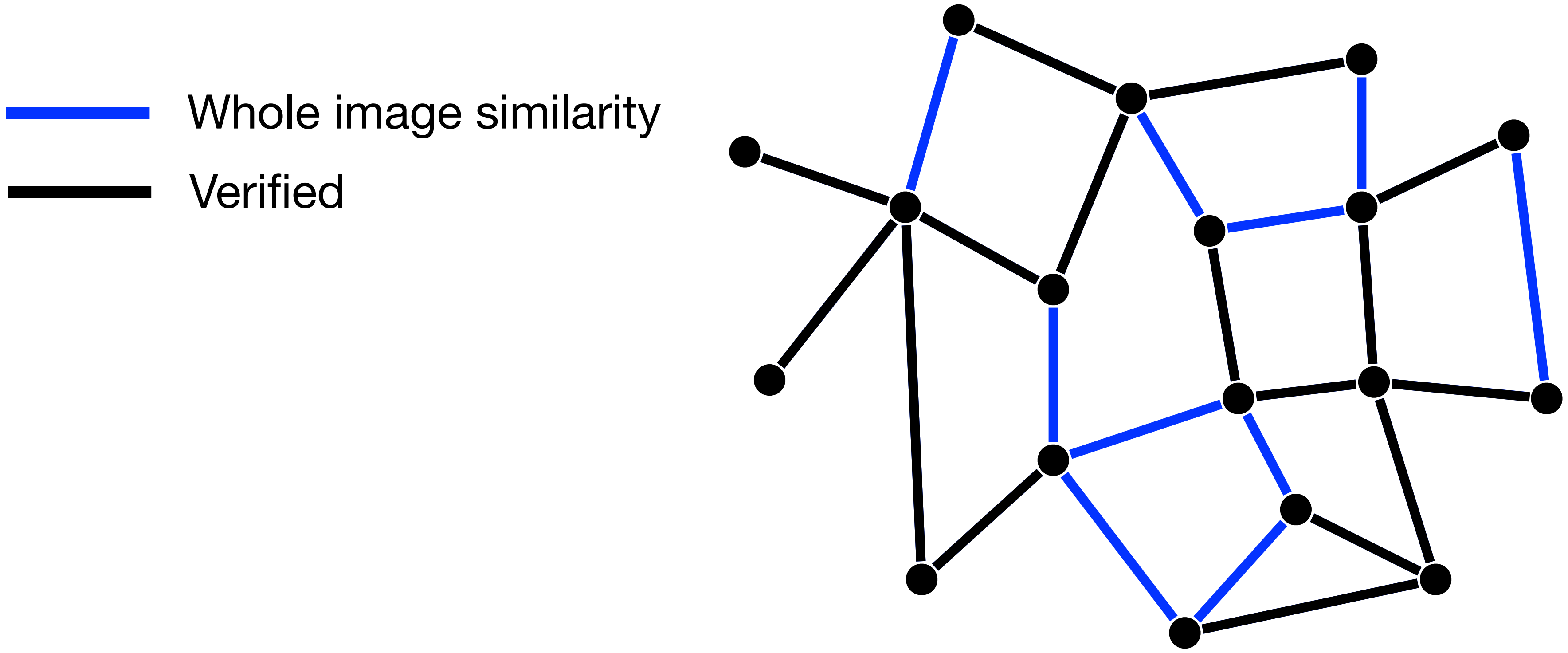
**http://grail.cs.washington.edu/rome**

# Matching Algorithm

━ Whole image similarity

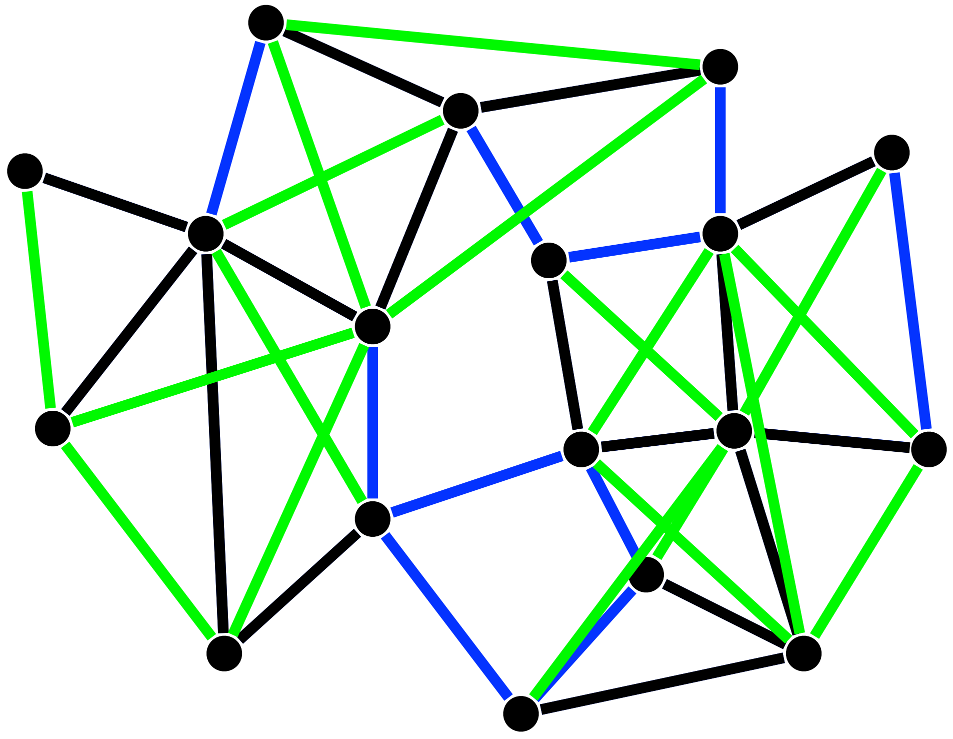


# Matching Algorithm

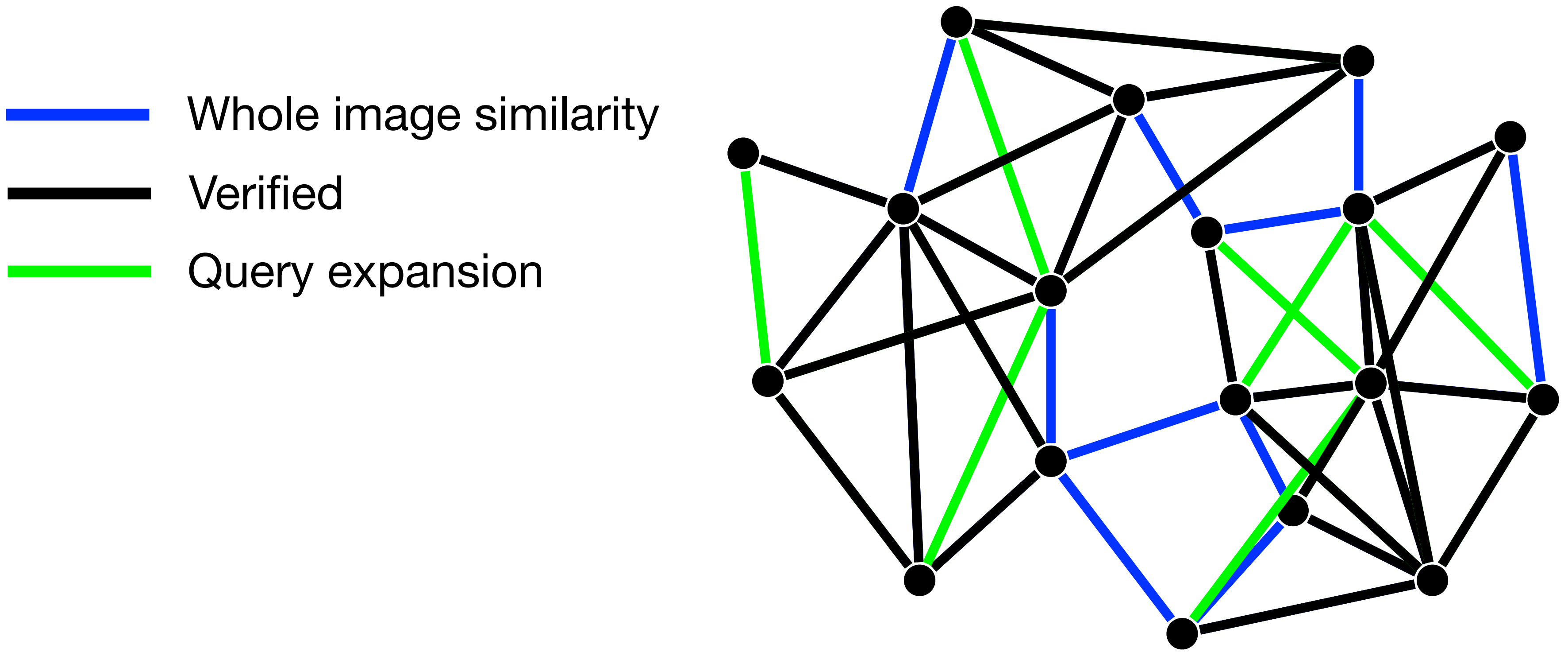


# Matching Algorithm

- 🟦 Whole image similarity
- ⬛ Verified
- 🟩 Query expansion



# Matching Algorithm



Whole image similarity

Verified

Query expansion

# Matching Algorithm