VSLAM on Phones
Closing Loops

Frank Dellaert

CVPR 2014 Visual SLAM Tutorial

With many slides/movies generously donated by **Torsten Sattler (!!!)**, Gim Hee Lee, Marc Pollefeys
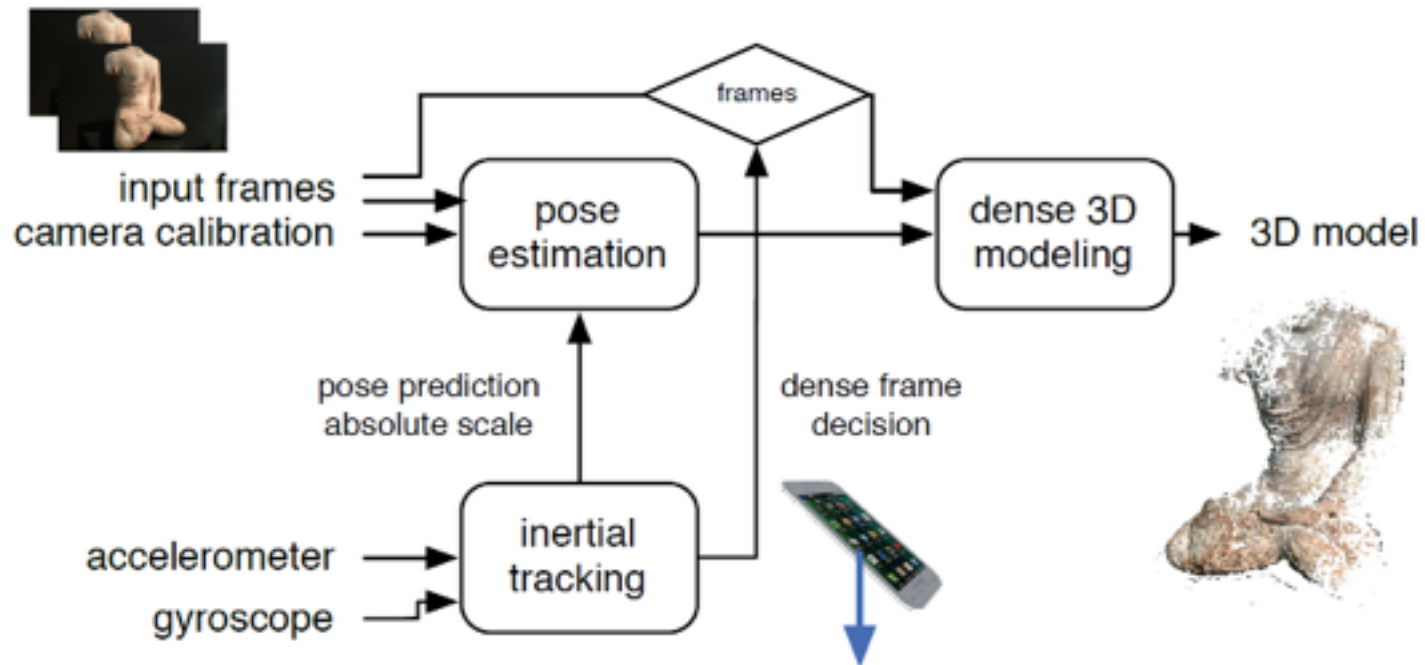
Outline

- VSLAM on the Phone
  - With live Metaio demo
- VSLAM in the Large
  - KITTI, ETH Multi-camera demo
- Direct
  - F2P, Prioritized (Sattler11iccv)
  - P2F, Prioritized (Li10eccv)
- Image Retrieval
  - Nistér06cvpr
  - Sattler12bmvc

Outline

- VSLAM on the Phone
  - With live Metaio demo
- VSLAM in the Large
  - KITTI, ETH Multi-camera demo
- Direct
  - F2P, Prioritized (Sattler11iccv)
  - P2F, Prioritized (Li10eccv)
- Image Retrieval
  - Nistér06cvpr
  - Sattler12bmvc

Turning Mobile Phones into 3D-Scanners

CVG Computer Vision and Geometry Lab   ETH zürich

Mobile 3D scanner: Pipeline

# Sparse SLAM Example

Metaio: Live Demo by Jürgen Sturm!

Outline

- VSLAM on the Phone
  - With live Metaio demo
- VSLAM in the Large
  - KITTI, ETH Multi-camera demo
- Direct
  - F2P, Prioritized (Sattler11iccv)
  - P2F, Prioritized (Li10eccv)
- Image Retrieval
  - Nistér06cvpr
  - Sattler12bmvc

Loop Closing Factor Graph

- In addition to projection factors, IMU factors, add long-range links that correct the graph
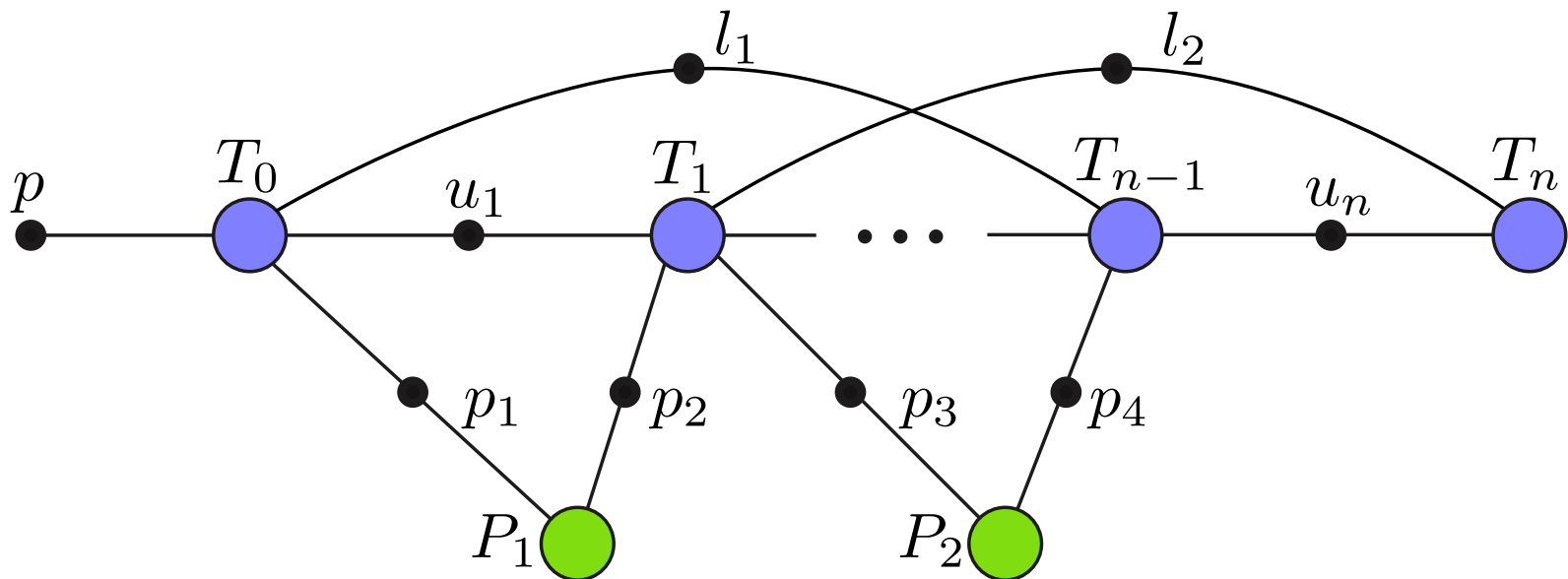
Real-time Large-scale VSLAM
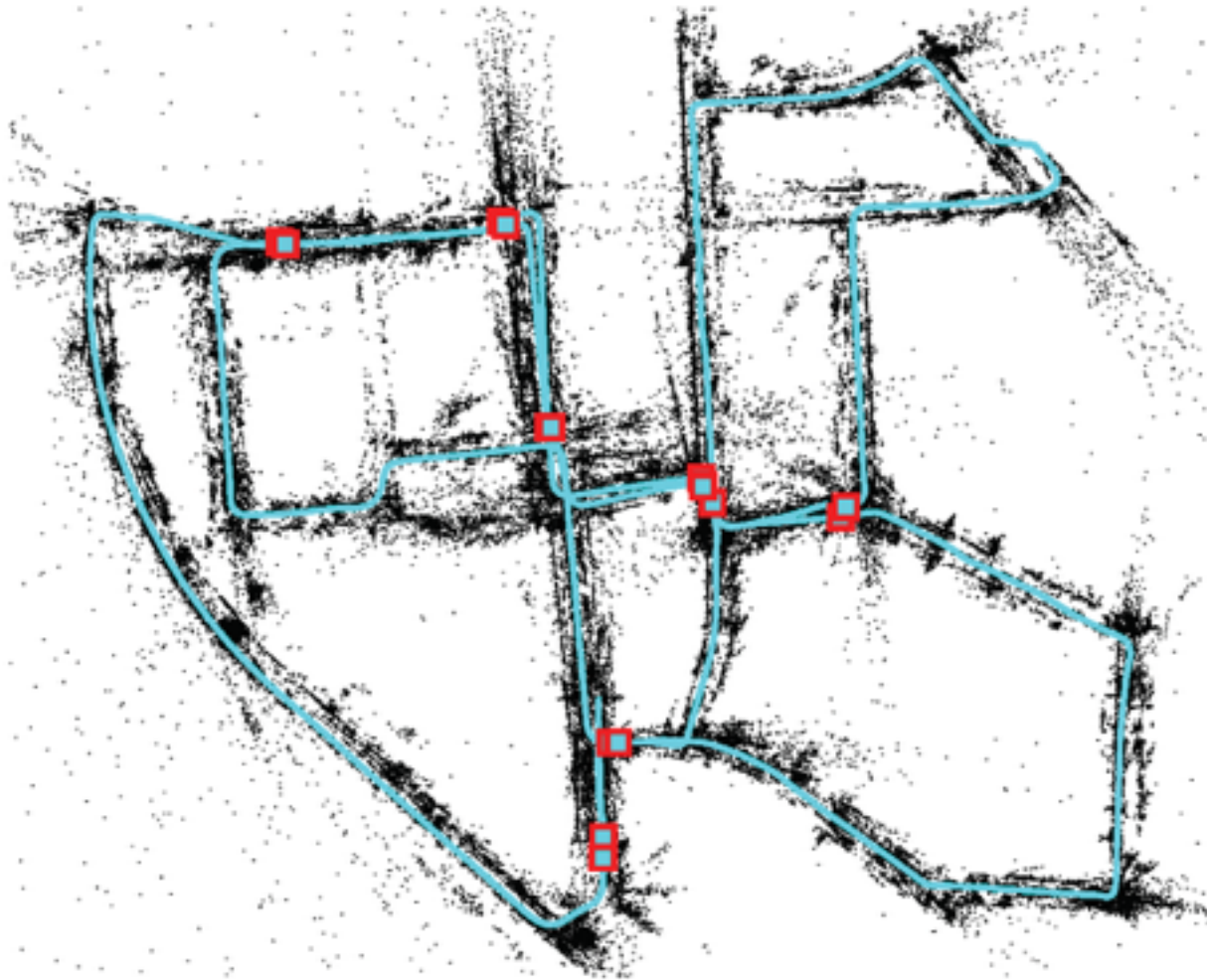# Lim et al, ICRA 2014

Real-Time 6-DOF Monocular Visual SLAM
in a Large-Scale Environment
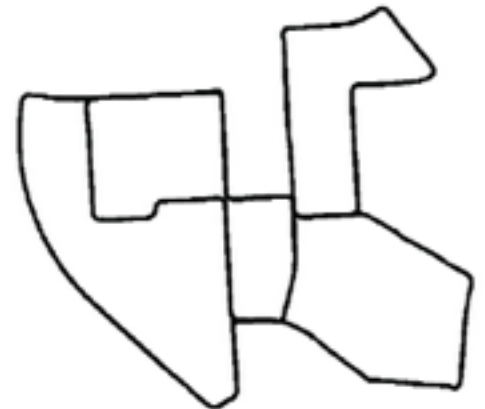
Hyon Lim, Jongwoo Lim, H. Jin Kim

ICRA 2014 Video
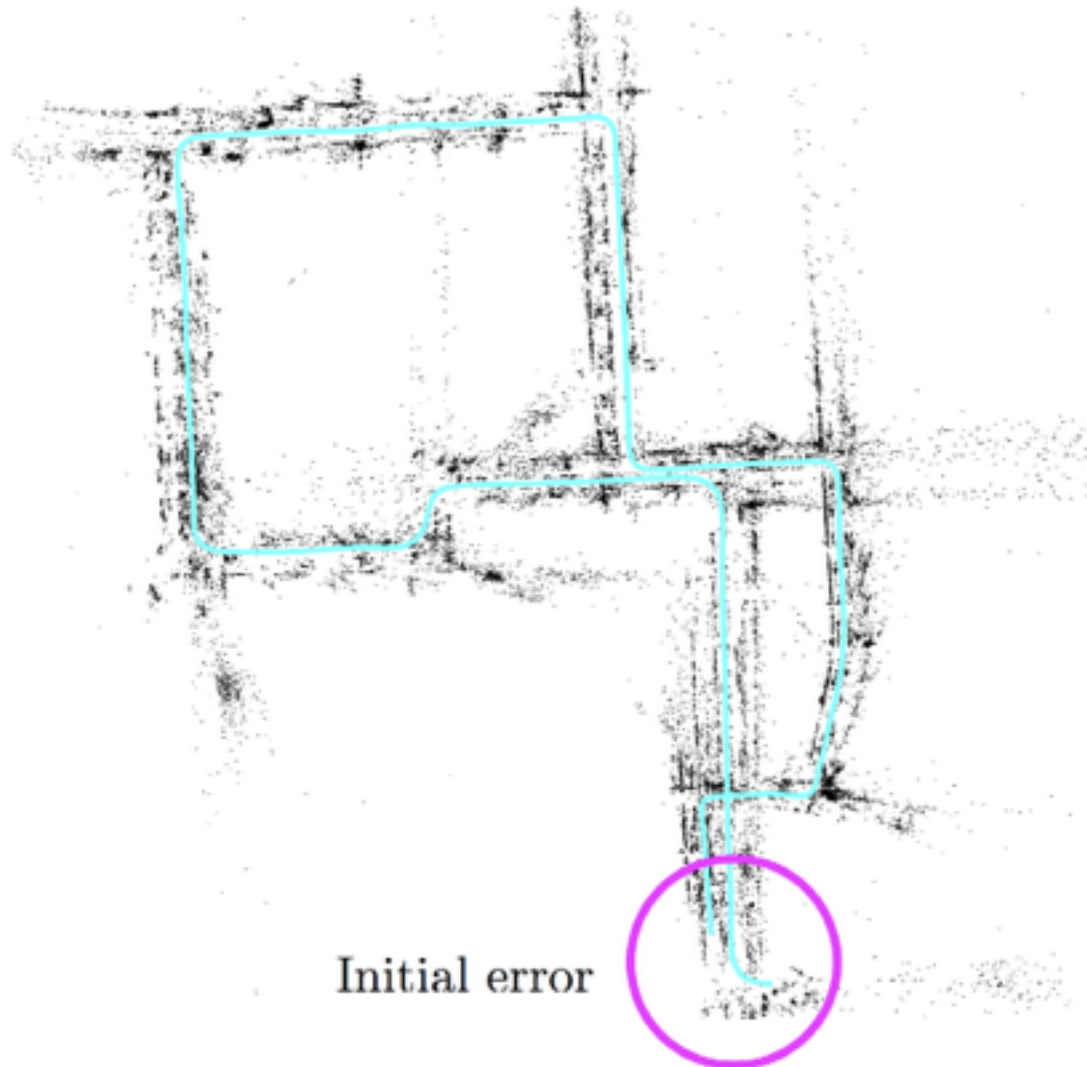
Resulting Map
Lim et al, ICRA 2014

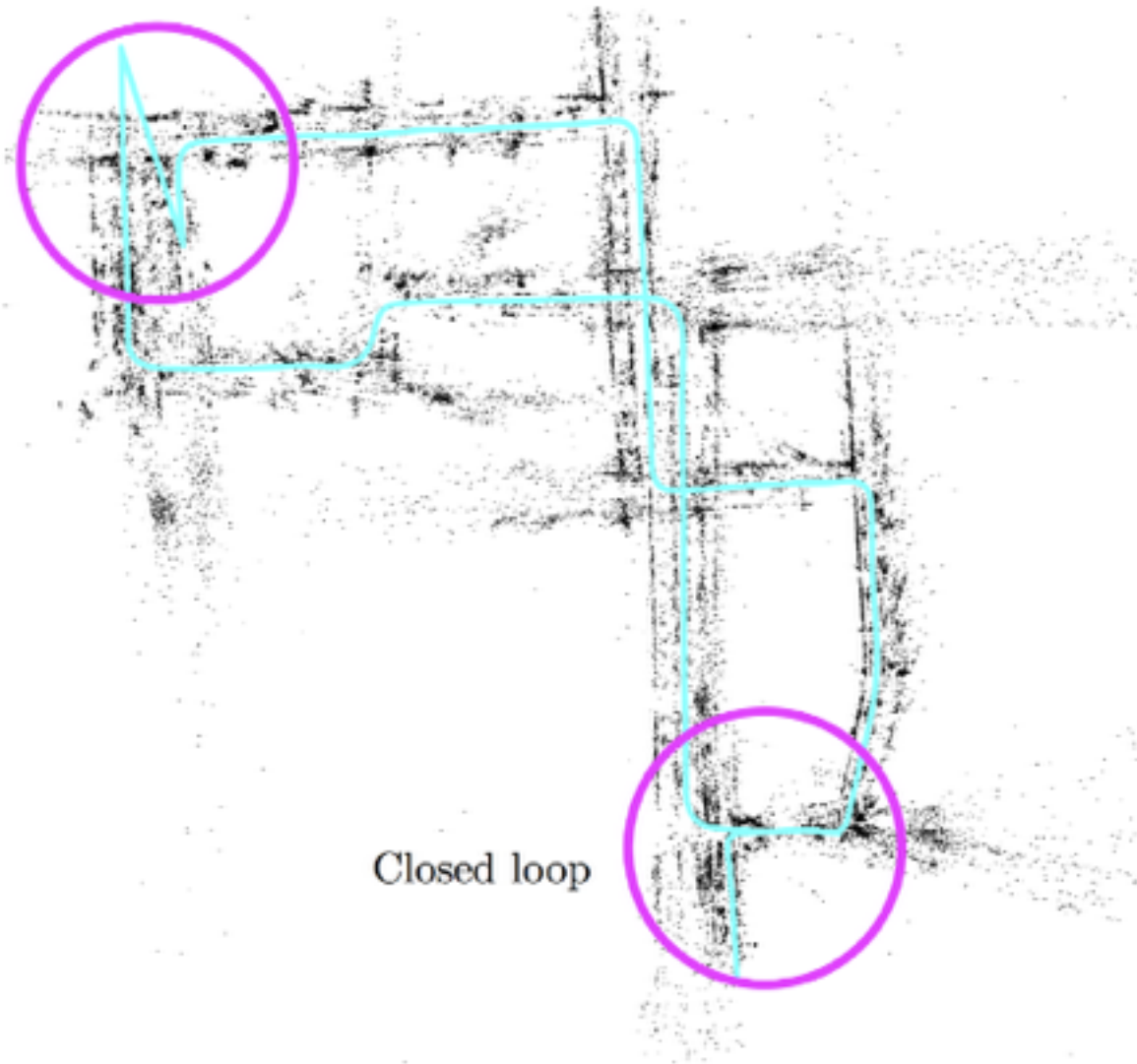

Ground truth

Red square = loop closure

Loop Closing
Lim et al, ICRA 2014

Initial error
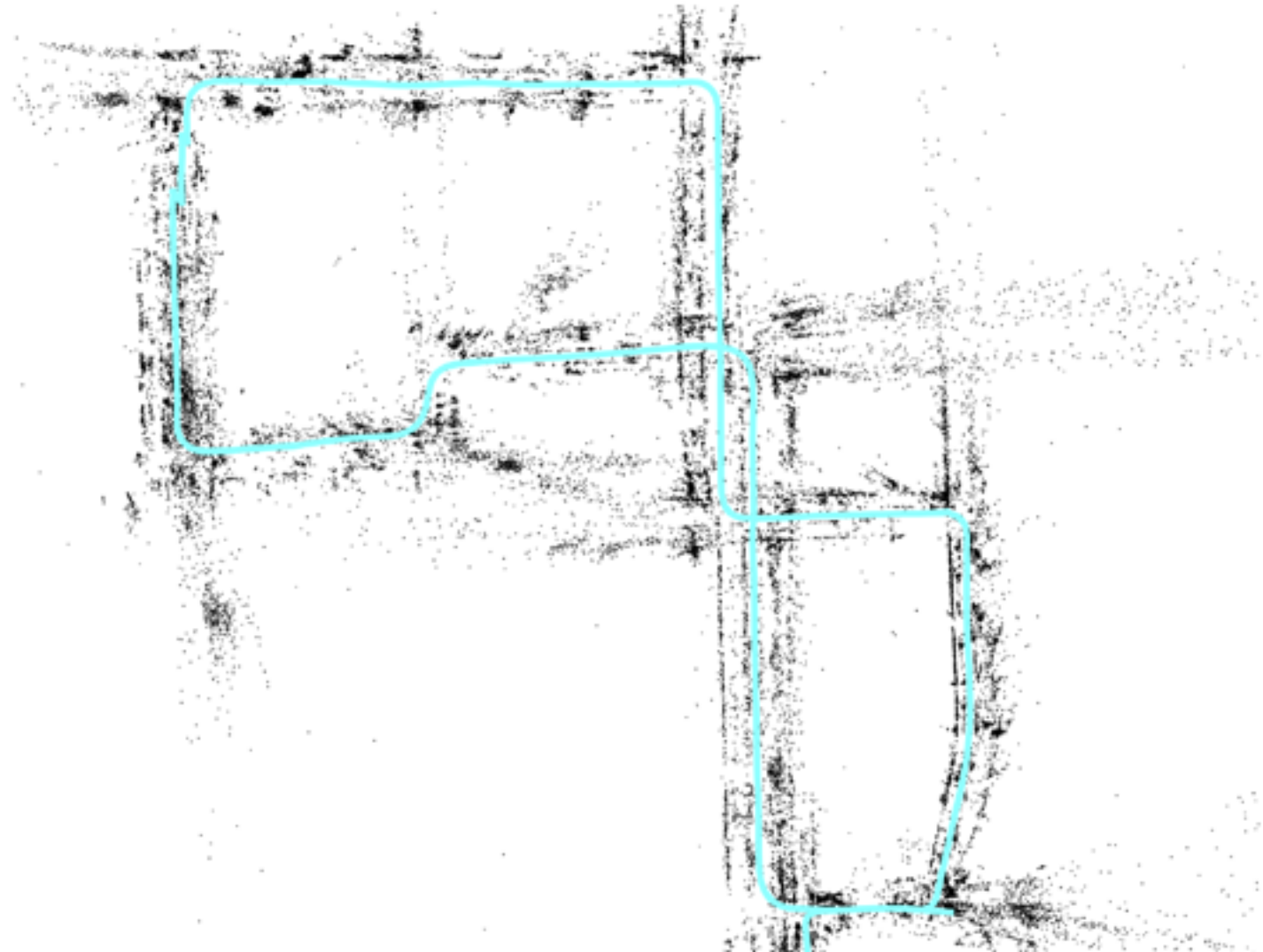
Loop Closing
Lim et al, ICRA 2014

Before pose-graph optimization

Closed loop

Loop Closing
Lim et al, ICRA 2014
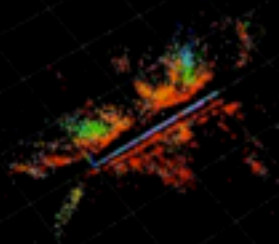
Multi-camera Visual SLAM
Gim Hee Lee et. al, ETH, 2014
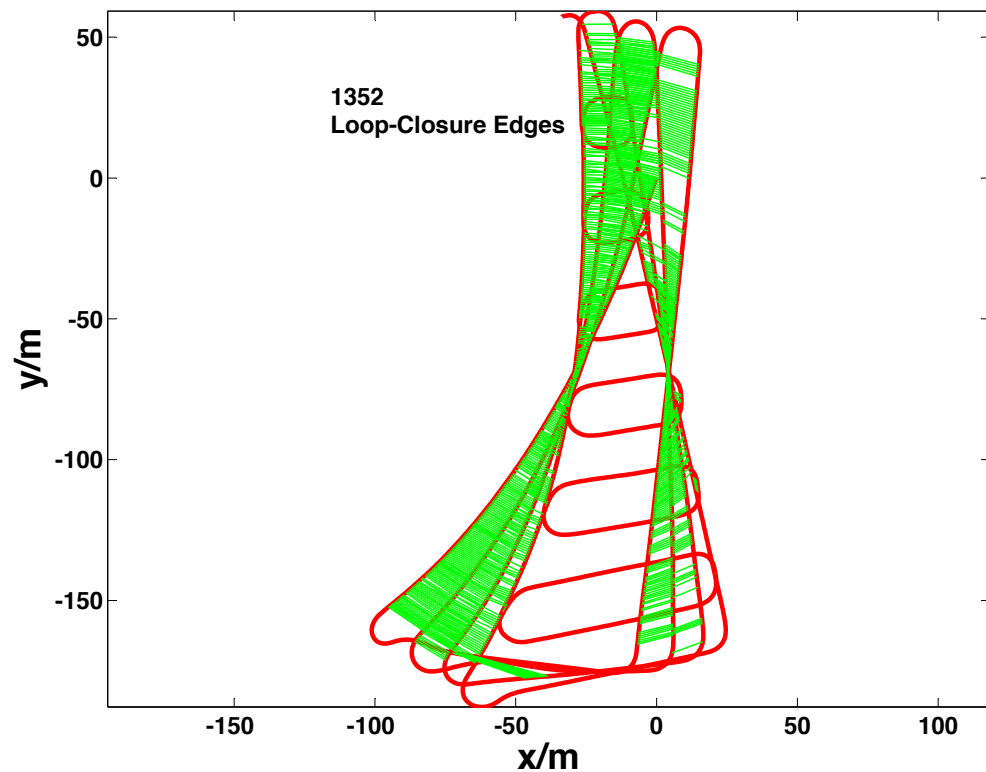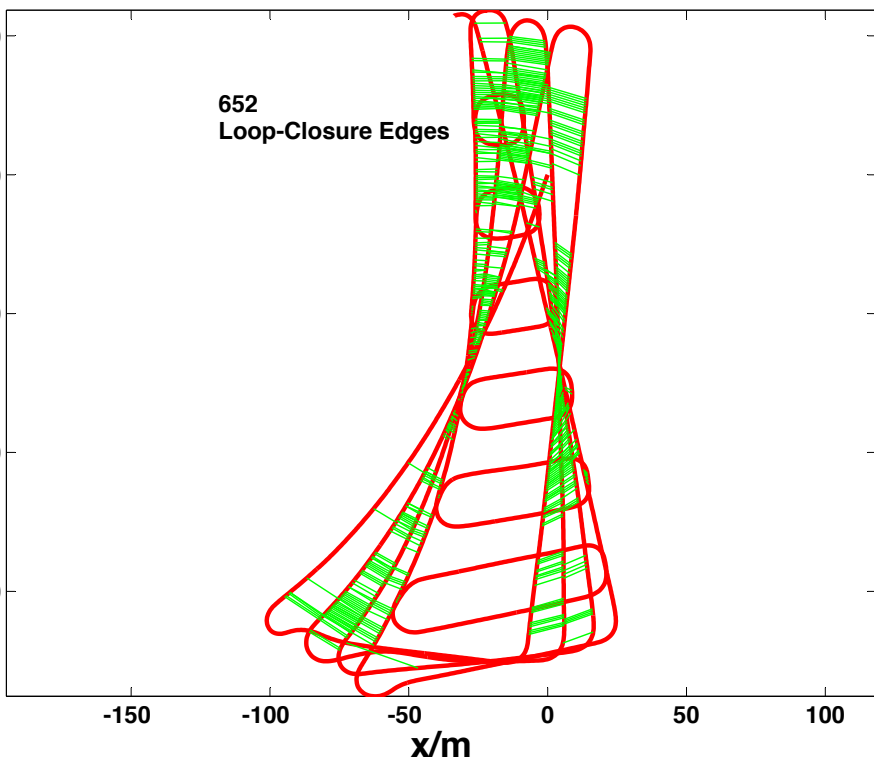
Multi-camera Visual SLAM
Gim Hee Lee et. al, ETH, 2014

Loop Closing
Gim Hee Lee et. al, ETH, 2014

Final Result
Gim Hee Lee et. al, ETH, 2014

Outline

- VSLAM on the Phone
  - With live Metaio demo
- VSLAM in the Large
  - KITTI, ETH Multi-camera demo
- Direct
  - F2P, Prioritized (Sattler11iccv)
  - P2F, Prioritized (Li10eccv)
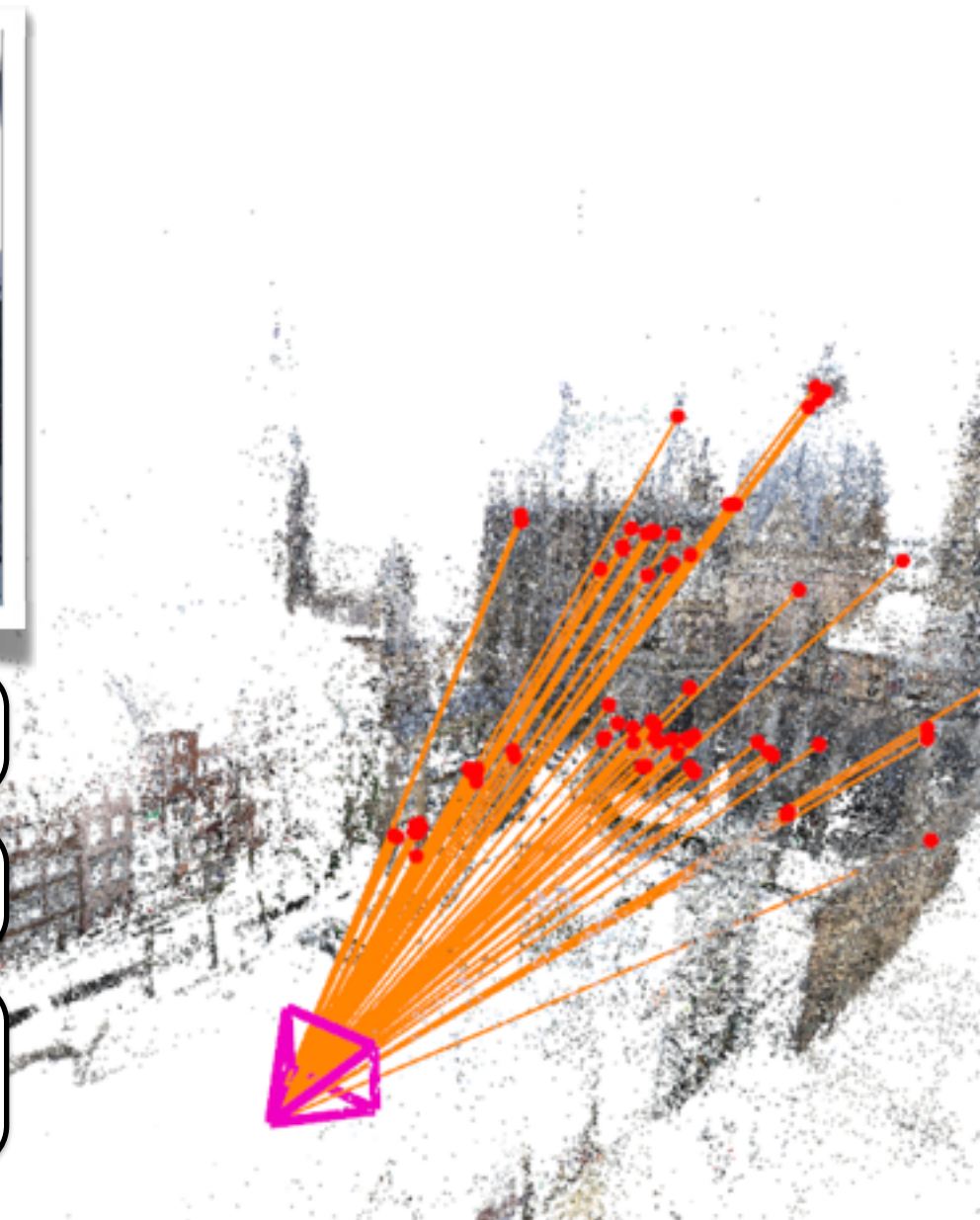- Image Retrieval
  - Nistér06cvpr
  - Sattler12bmvc

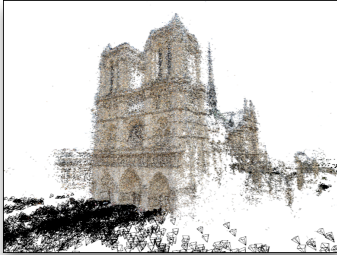# Image-Based Localization Pipeline

Extract Local Features

Establish 2D-3D Matches

Camera Pose Estimation:
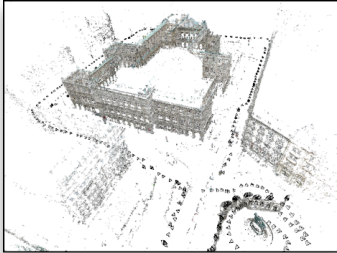RANSAC + n-Point-Pose Algorithm

What Situations Can Be Handled

Easy

Hard

- Database & query images from same source, e.g., Flickr
- 97% - 100% localization rates
- *Challenges*: Run-time & memory consumption for large scale

- Database & query images from different spatial distributions
- 70% - 90% localization rates
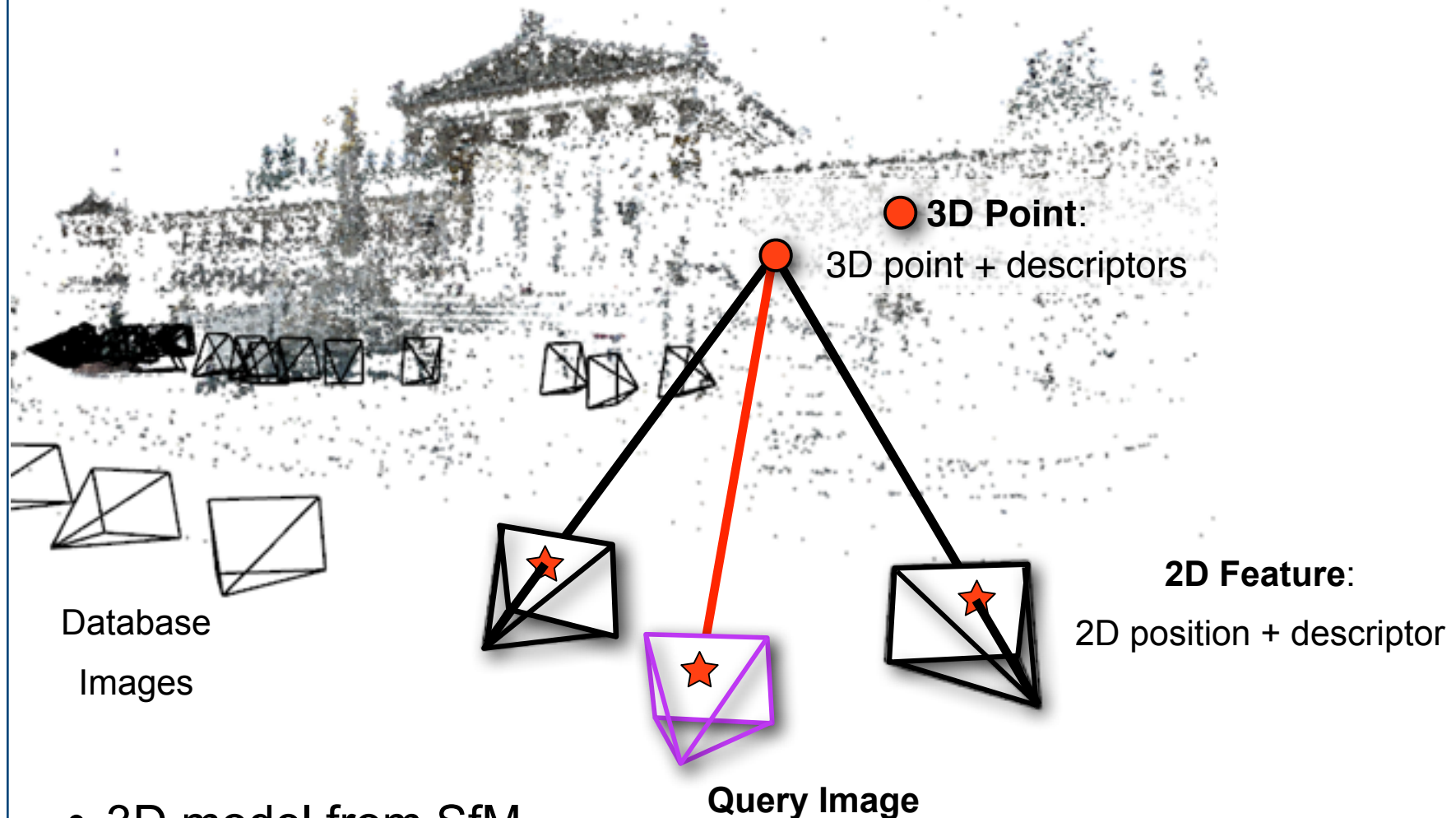- *Challenges*: Deal with larger variety in viewpoints

- Streetview imagery
- 50% - 65% localization rates
- *Challenges*: Repetitions, viewpoint variations, scale

- Indoor scenarios
- *Challenges*: Identical structures, small distance to scene

Establishing 2D-3D Matches

3D Point:
3D point + descriptors

2D Feature:
2D position + descriptor

Database
Images

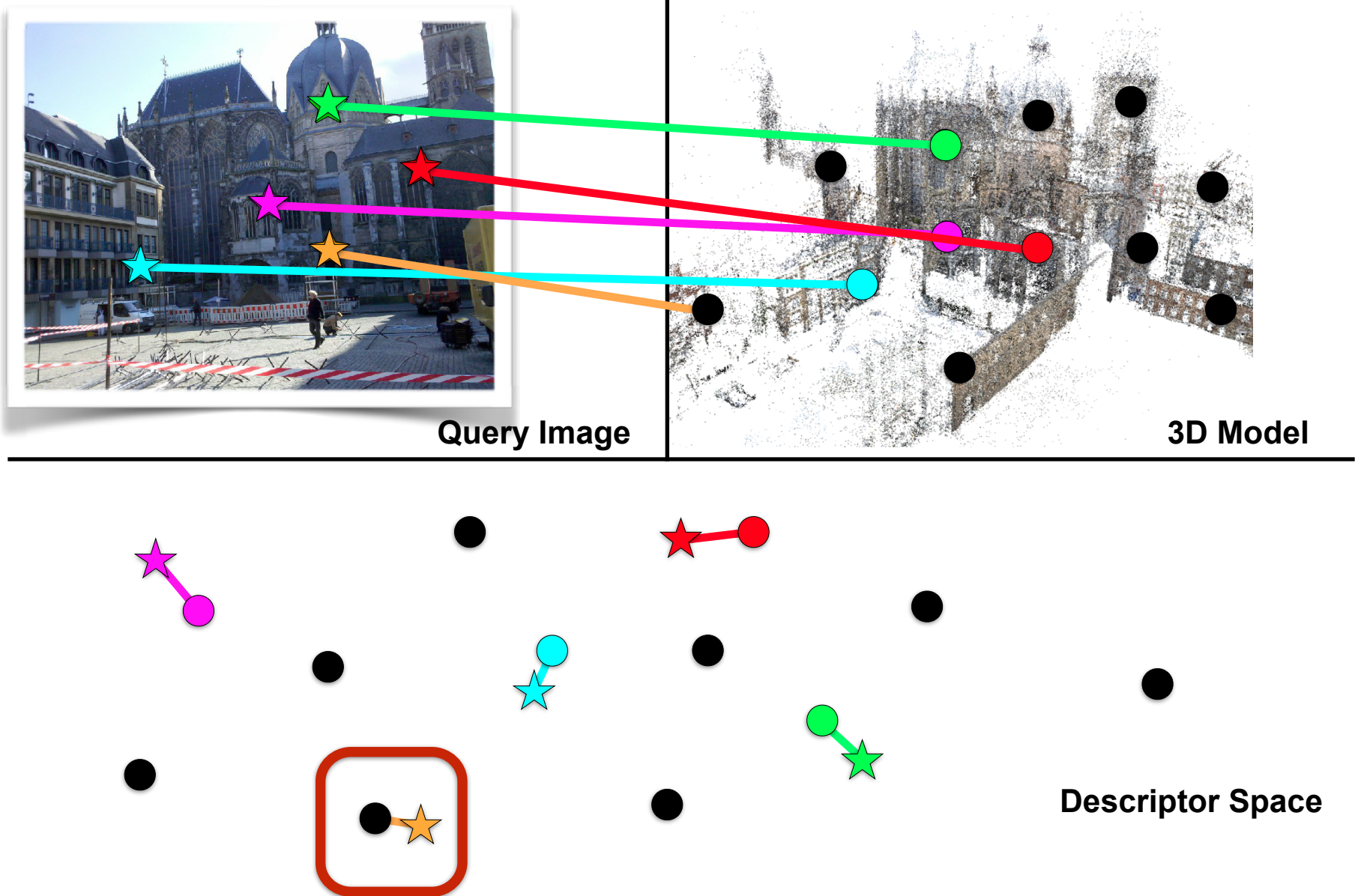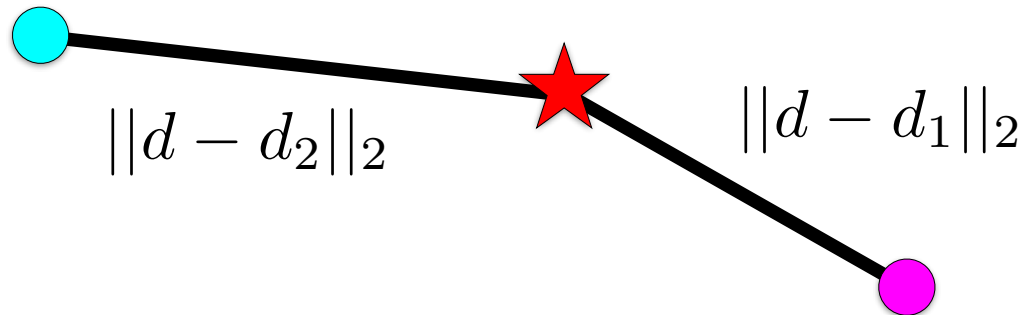**Query Image**

- 3D model from SfM
- 2D-3D correspondences from (SIFT) **descriptor matching**

Matching = Nearest Neighbor Search

Query Image

3D Model

Descriptor Space

Lowe's Ratio Test

- No every nearest neighbor is correct

- Use ratio test to reject wrong / ambiguous matches [Lowe, IJCV'04]

- Only accept match if

$$\frac{||d - d_1||_2}{||d - d_2||_2} < 0.8$$

Nearest Neighbor Search

- Typical datasets: `3-10k` features, `>1M` points
- ➡ Exhaustive (linear) nearest neighbor search is prohibitive
- **Curse of dimensionality**: No exact search method that is faster than linear search

- Multiple fast approximate nearest neighbor search methods:
  - **kd-trees** [Muja & Lowe, PAMI'14] [code]
  - **Hierarchical k-means trees** [Muja & Lowe, PAMI'14] [code]
  - Product quantization [Jégou et al., PAMI'11] (Orals 4A) [code] [Kalantidis & Avrithis, CVPR'14] (Posters P5)
 - Diverse hashing techniques
  - …

kd-tree Construction & Search
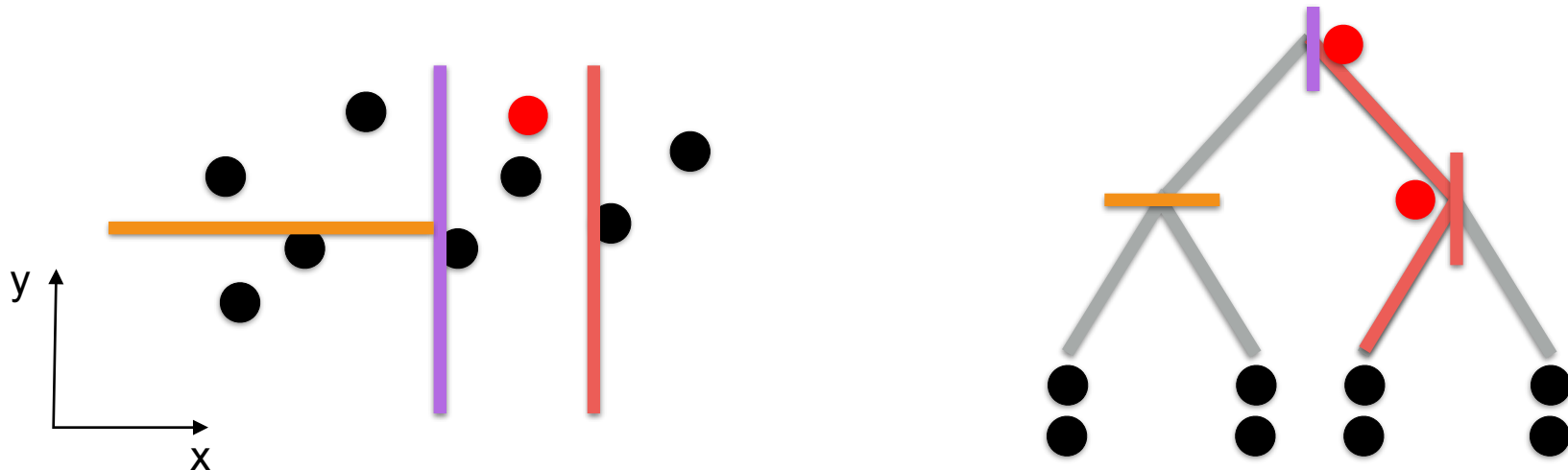
- Iteratively split dimension with largest variance at median

- Traversal based on side of split

- 1B SIFT descriptors (128D) → Only 30 dimensions considered!
  - Curse of dimensionality: Need to visit all leaves!
  - Approximate search: Visit $\mathbb{N}$ leaf nodes

Tree-Based Search Drawbacks

- Tree-based approach so slow because it
    - … tries to find all possible matches
        - **Don't need all of them!**
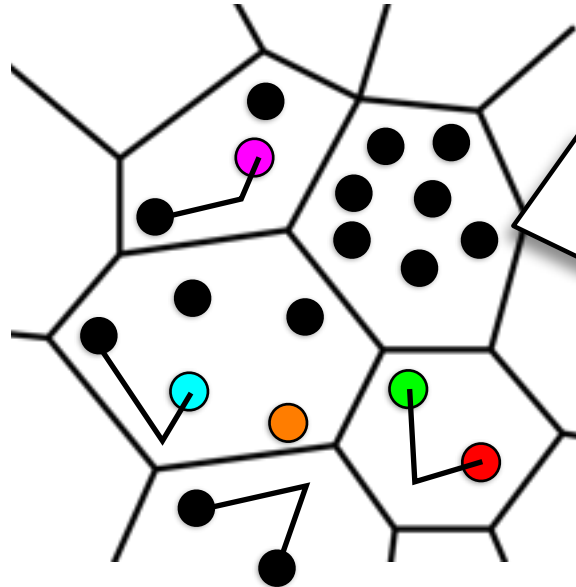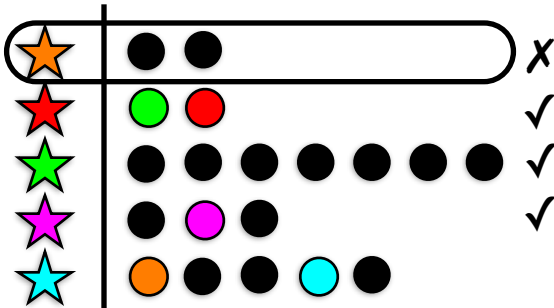   - … ignores dependencies between matches

Unlikely to match

More likely to match
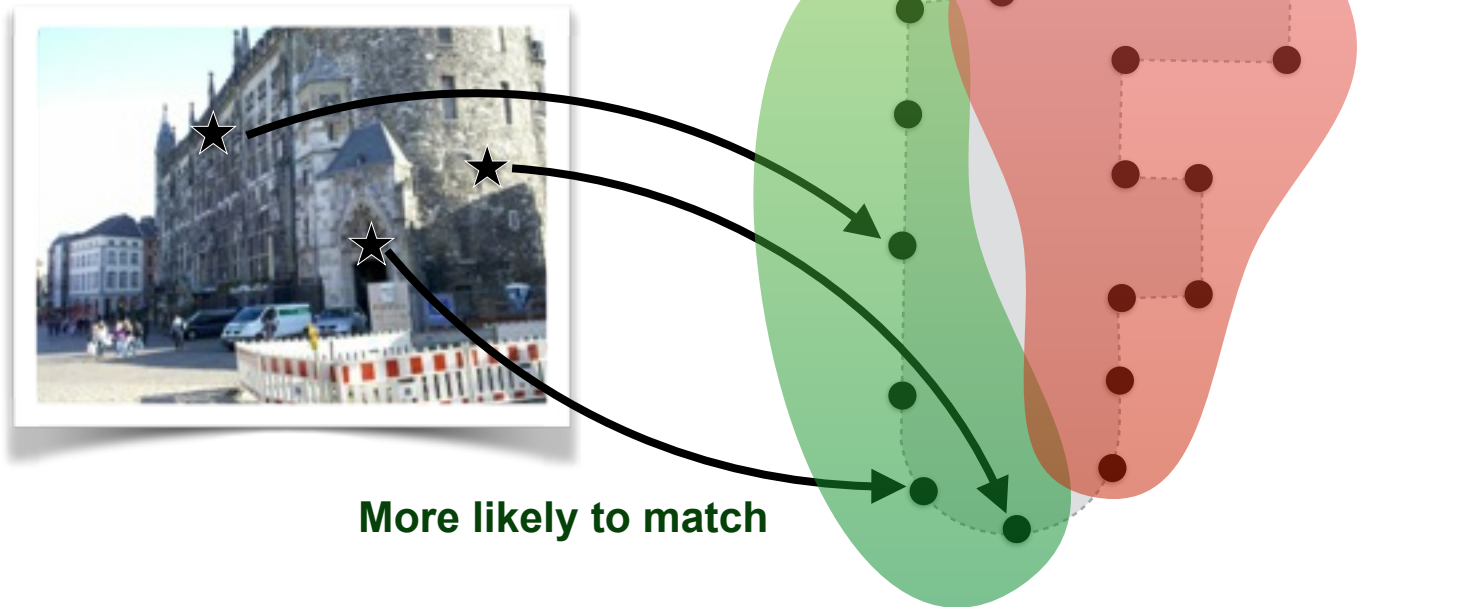
➡ **Exploit co-visibility information to guide matching!**

Vocabulary-Based Prioritized Search (VPS)

Query Image

3D Model

100k words
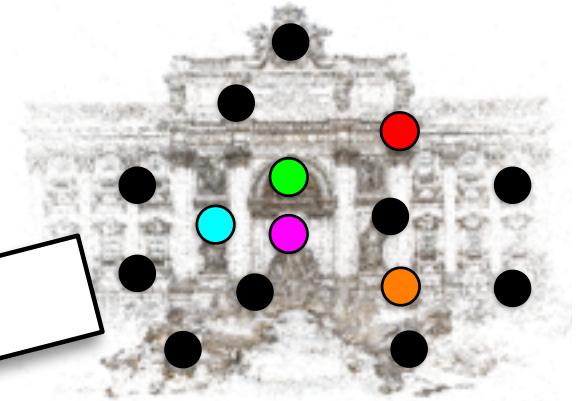
Assign features to words	Sort based on costs	Linear search through words	Stop after 100 matches	Pose estimation: RANSAC + p6p

[Sattler et al., ICCV'11] [code]
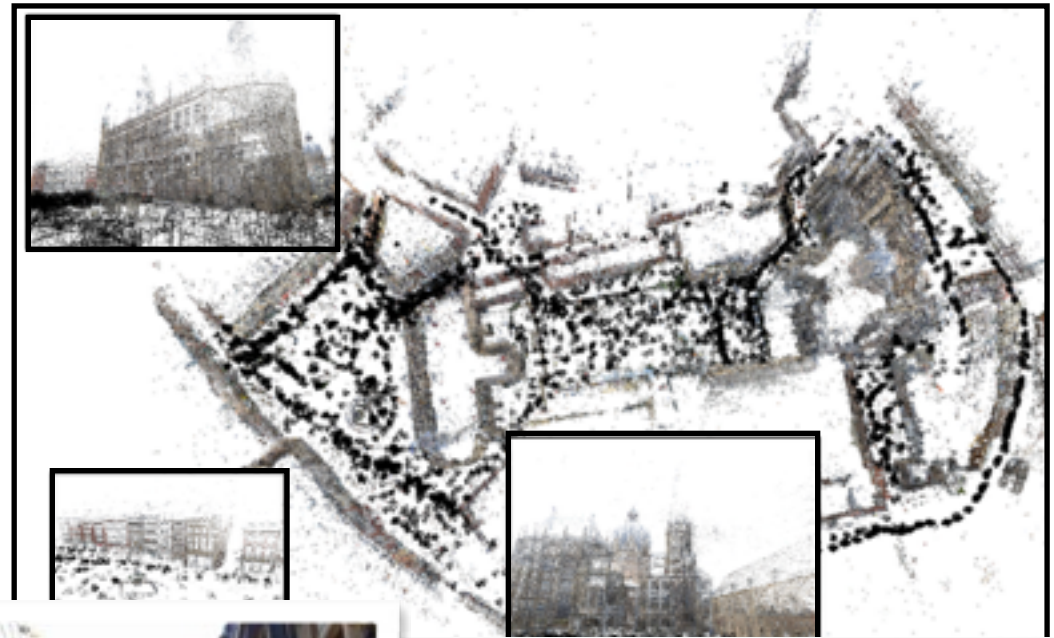
Vocabulary-Based Prioritized Search (VPS)
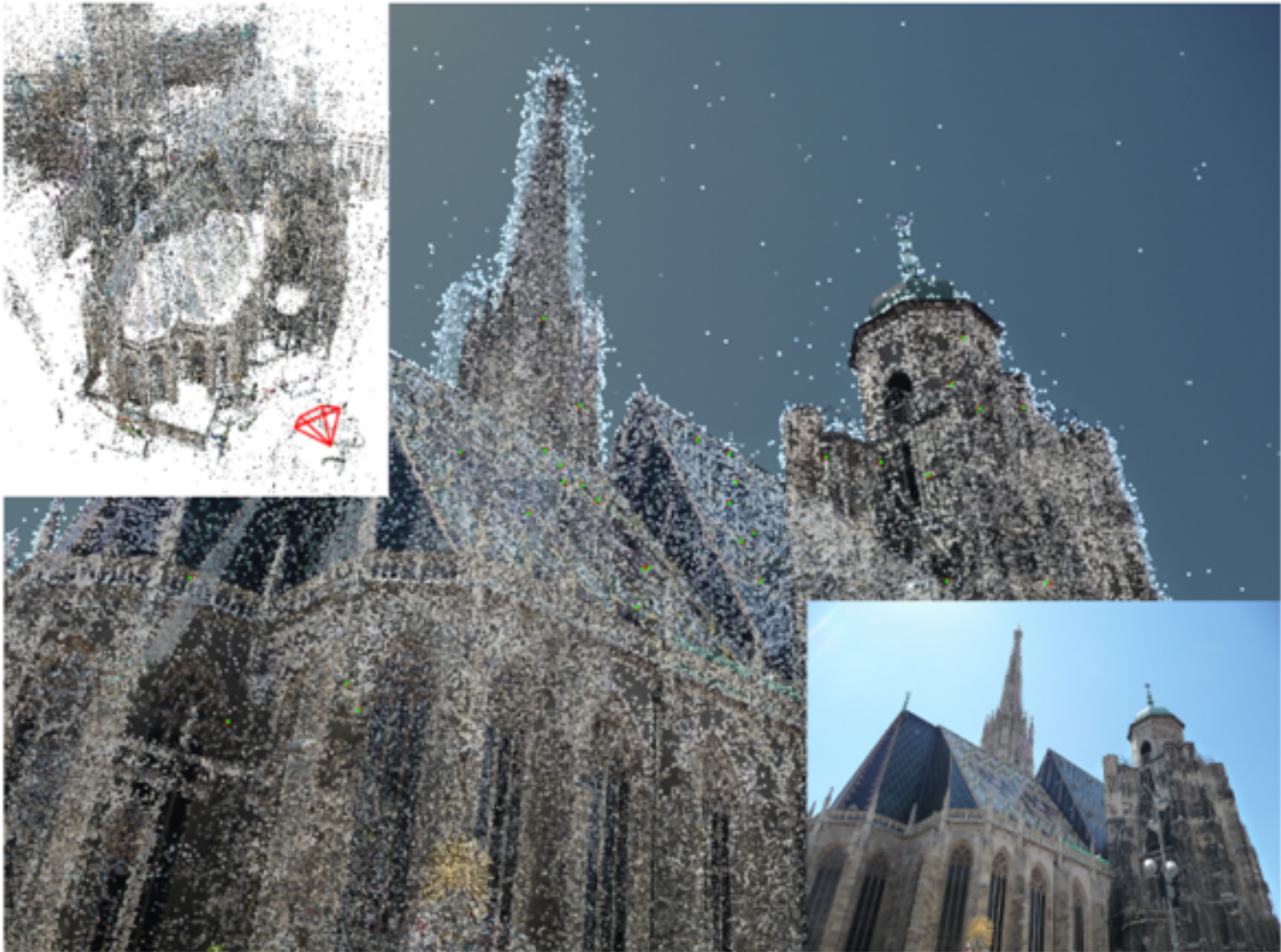
City-Scale Localization

Example: Aachen
~1.5M points
3047 database images
369 query images

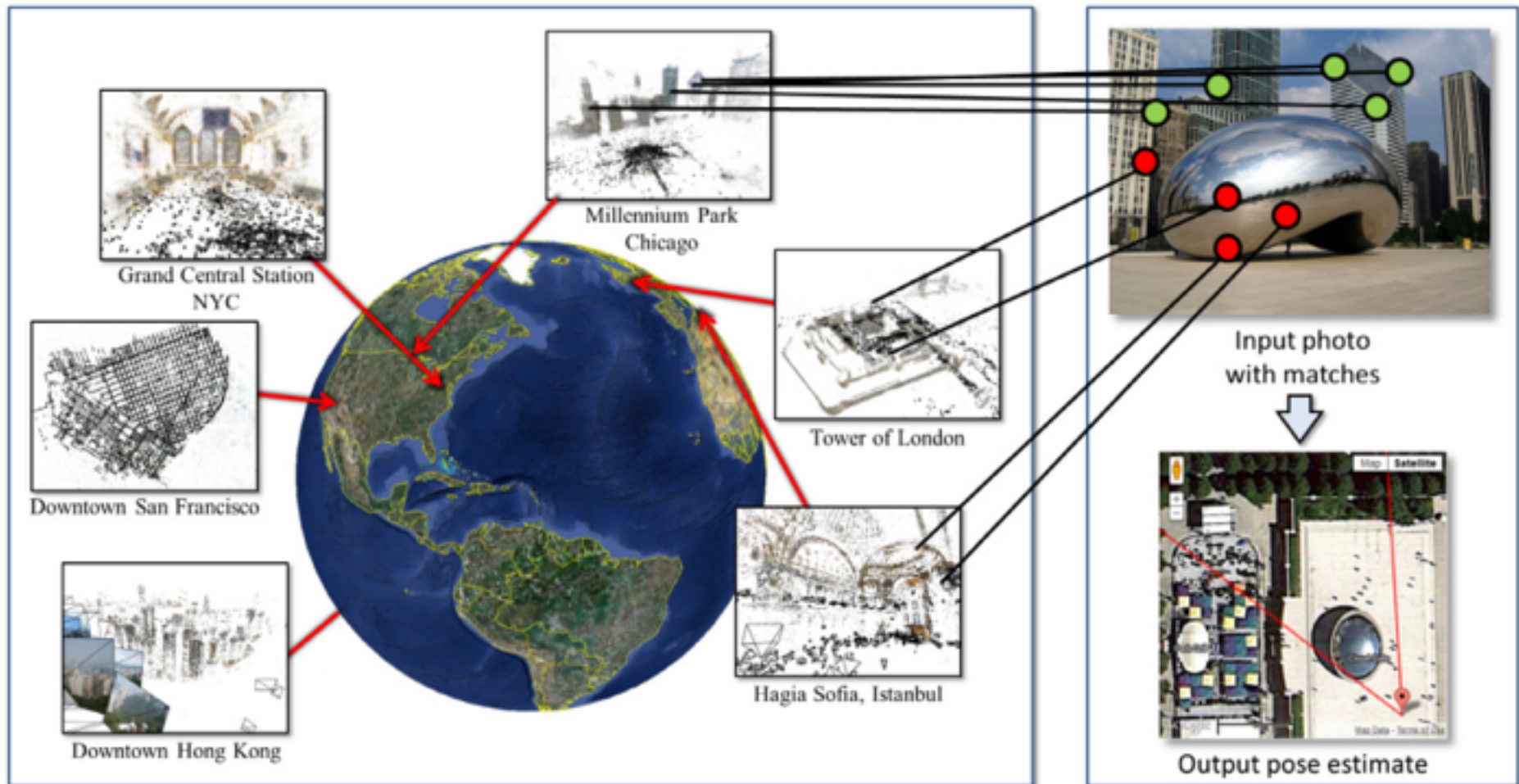# Vocabulary-Based Prioritized Search (VPS)

[Sattler et al., ICCV'11] [code]

Worldwide Pose Estimation using 3D Point Clouds

Grand Central Station NYC

Downtown San Francisco

Downtown Hong Kong

Millennium Park Chicago

Tower of London

Hagia Sofia, Istanbul

Input photo with matches

Output pose estimate

[Li et al., ECCV'10]

Prioritized Point-to-Feature Matching (P2F)

Idea: Use Visibility Graph to guide 3D-to-2D matching
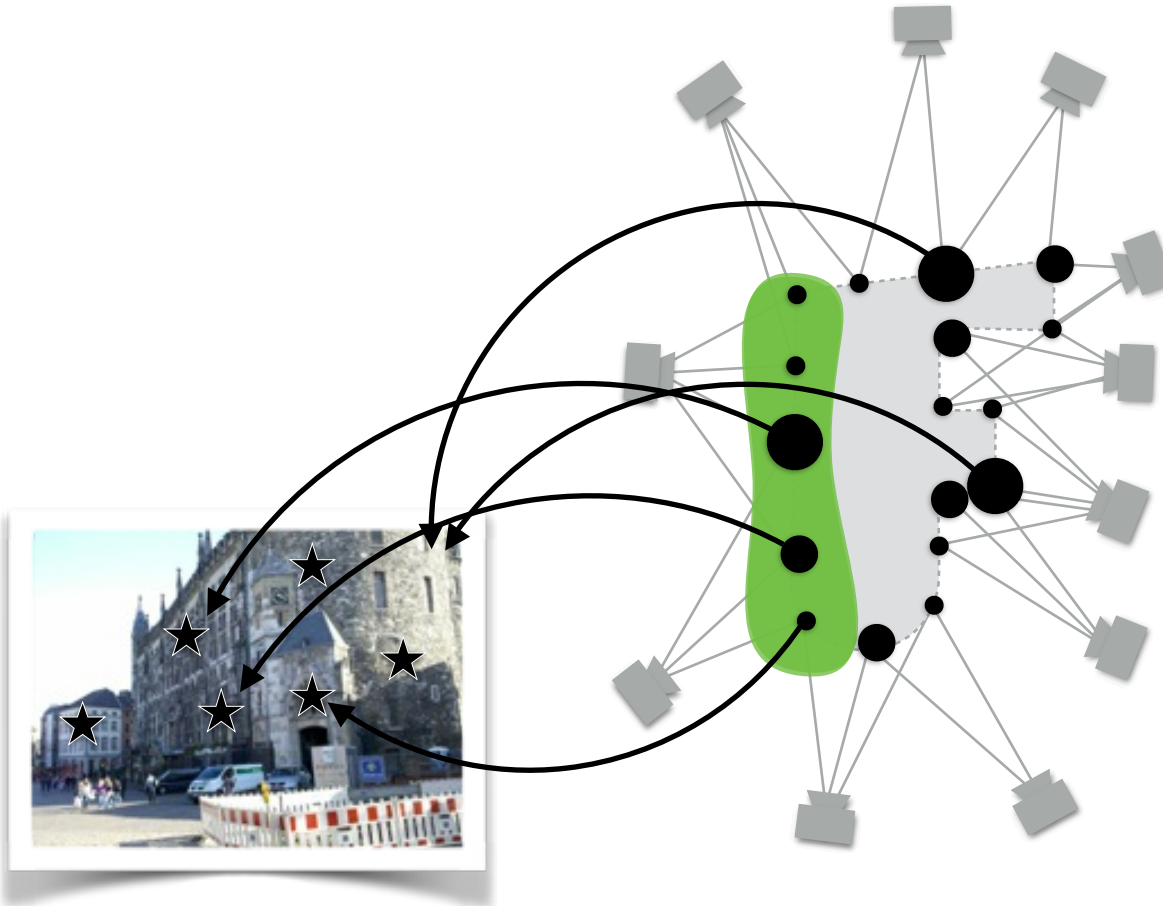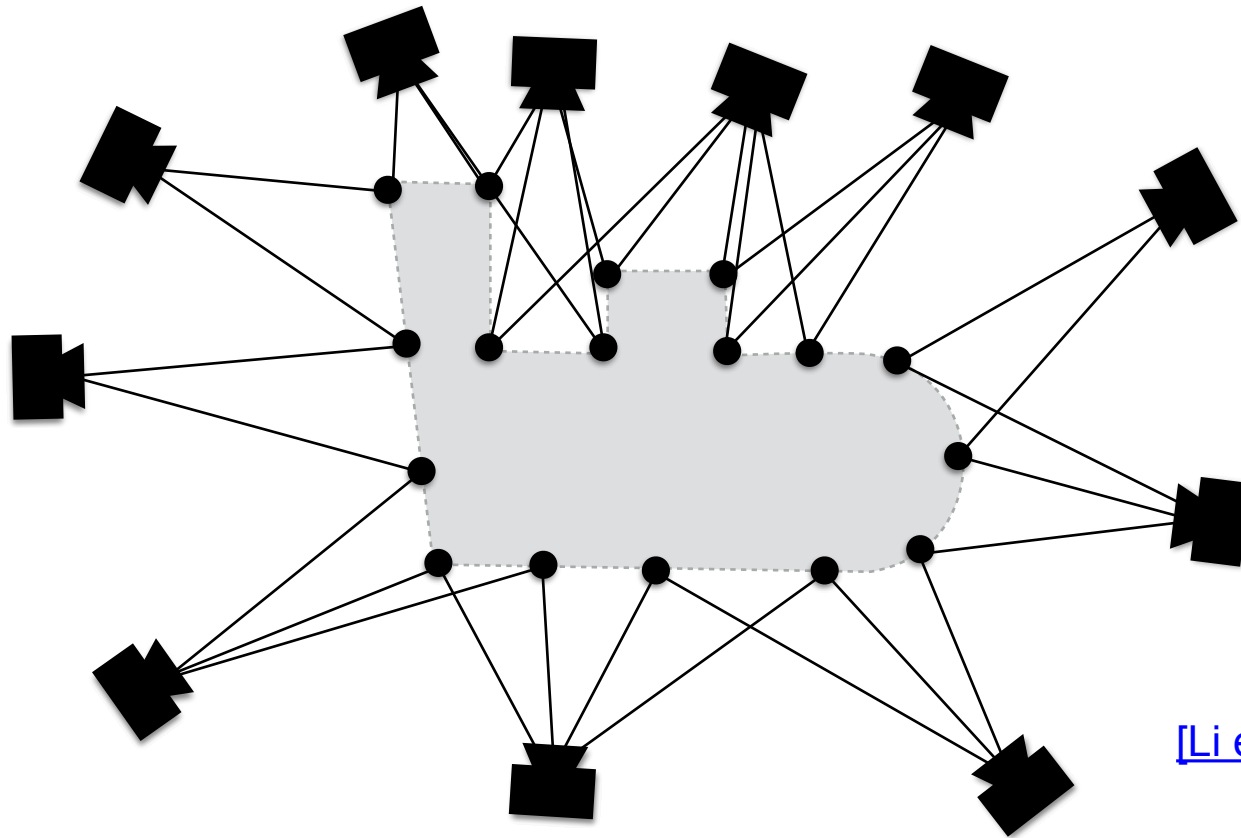

Build kd-tree for query image

Prioritized 3D-to-2D Matching

Stop after 100 **matches**

Camera Pose Estimation: RANSAC + P6P

[Li et al., ECCV'10]

The Visibility Graph

[Li et al., ECCV'10]

- Bipartite visibility graph G_V defined by SfM reconstruction

- Two points co-visible if share a common camera

Point Priorities

- Start with points that are …

    - stable under viewpoint changes

    - at more popular parts of the model

    ➡ Points with high degree in G_V

- **Initial priority** of point p_i: $S_i = d_i$ = degree in G_V

- **Update priorities**: $S_j = S_j + 10/d_i$ for co-visible points p_j

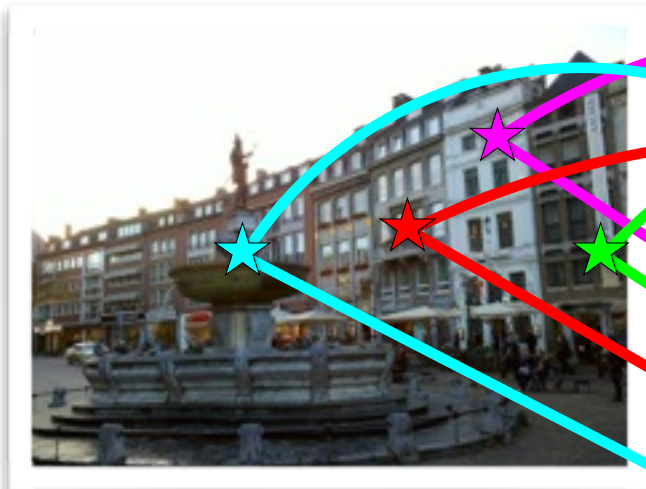[Li et al., ECCV'10]

Outline

- VSLAM on the Phone
  - With live Metaio demo
- VSLAM in the Large
  - KITTI, ETH Multi-camera demo
- Direct
  - F2P, Prioritized (Sattler11iccv)
  - P2F, Prioritized (Li10eccv)
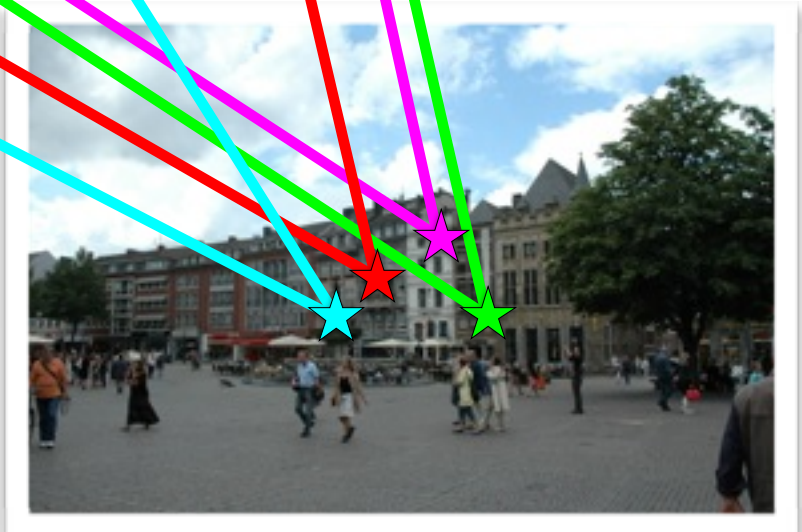- Image Retrieval
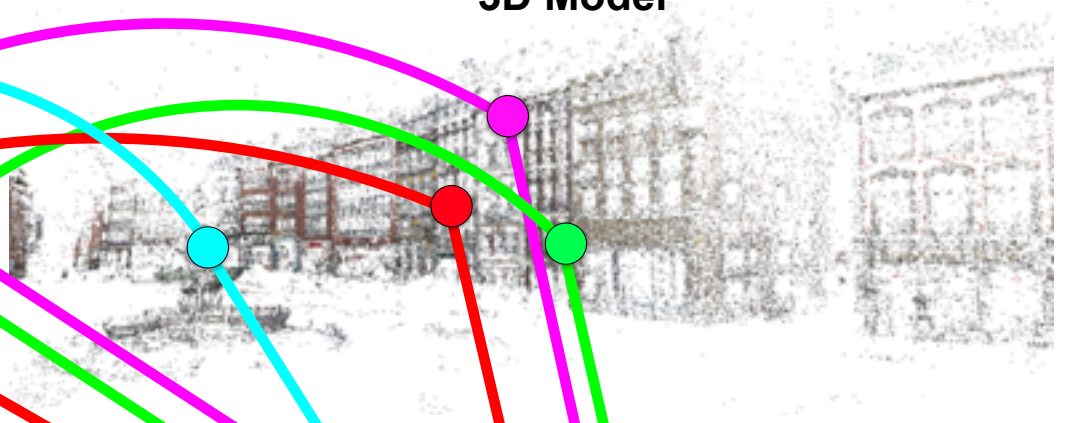 - Nistér06cvpr
  - Sattler12bmvc

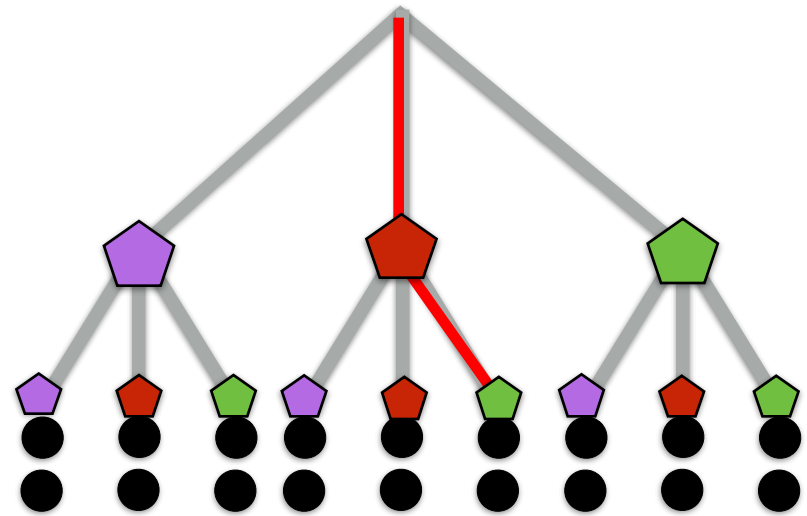# Image-Based Localization & Place Recognition

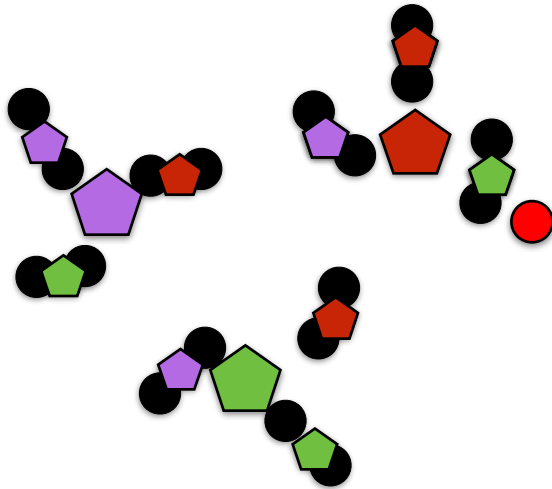**Direct Matching**
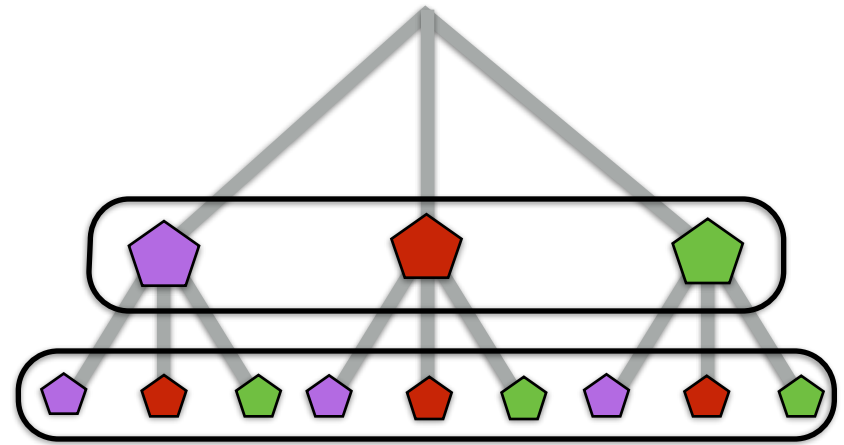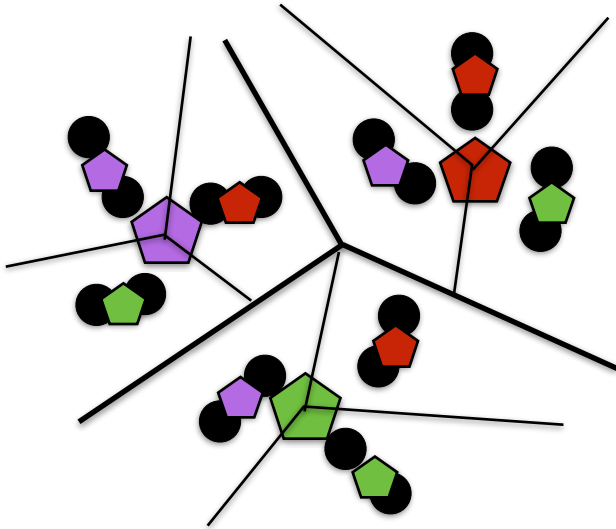
3D Model

Query Image

Retrieved Database Image

Hierarchical k-Means Tree

- Iteratively apply k-means clustering

- Traversal based on nearest neighboring cluster

- Approximate search: Visit $\mathbb{N}$ leaf nodes

- Performs (slightly) worse than kd-tree

Vocabulary Trees

- Each level in a hierarchical k-means tree defines a quantization of the descriptor space (**visual vocabulary**)

- Hierarchical k-means trees also known as **Vocabulary Trees** [Nister & Stewenius, CVPR'06]