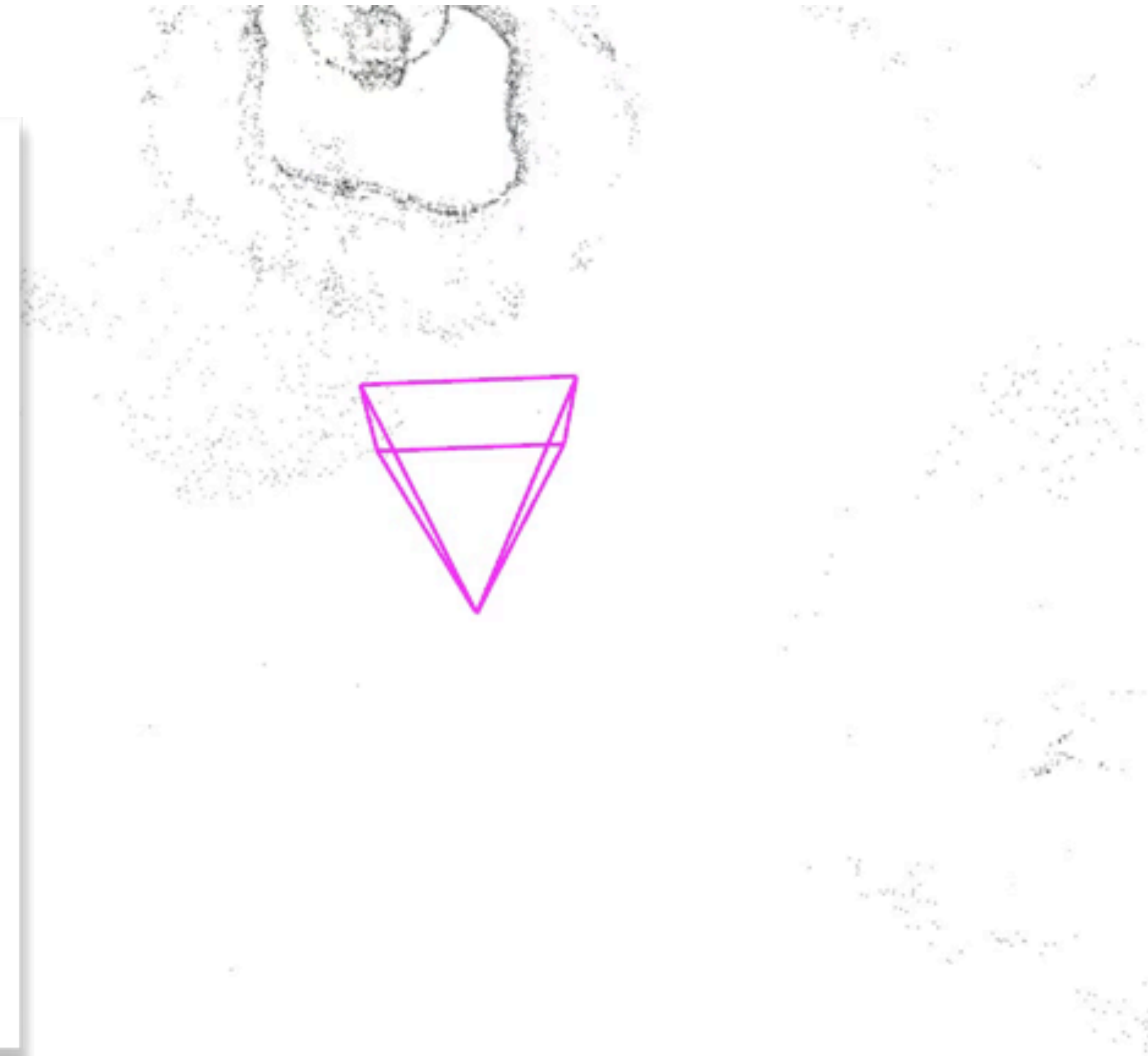
Has the (Large-Scale) Image-based Localization Problem been solved?

Torsten Sattler
Computer Vision & Geometry Lab
ETH Zurich

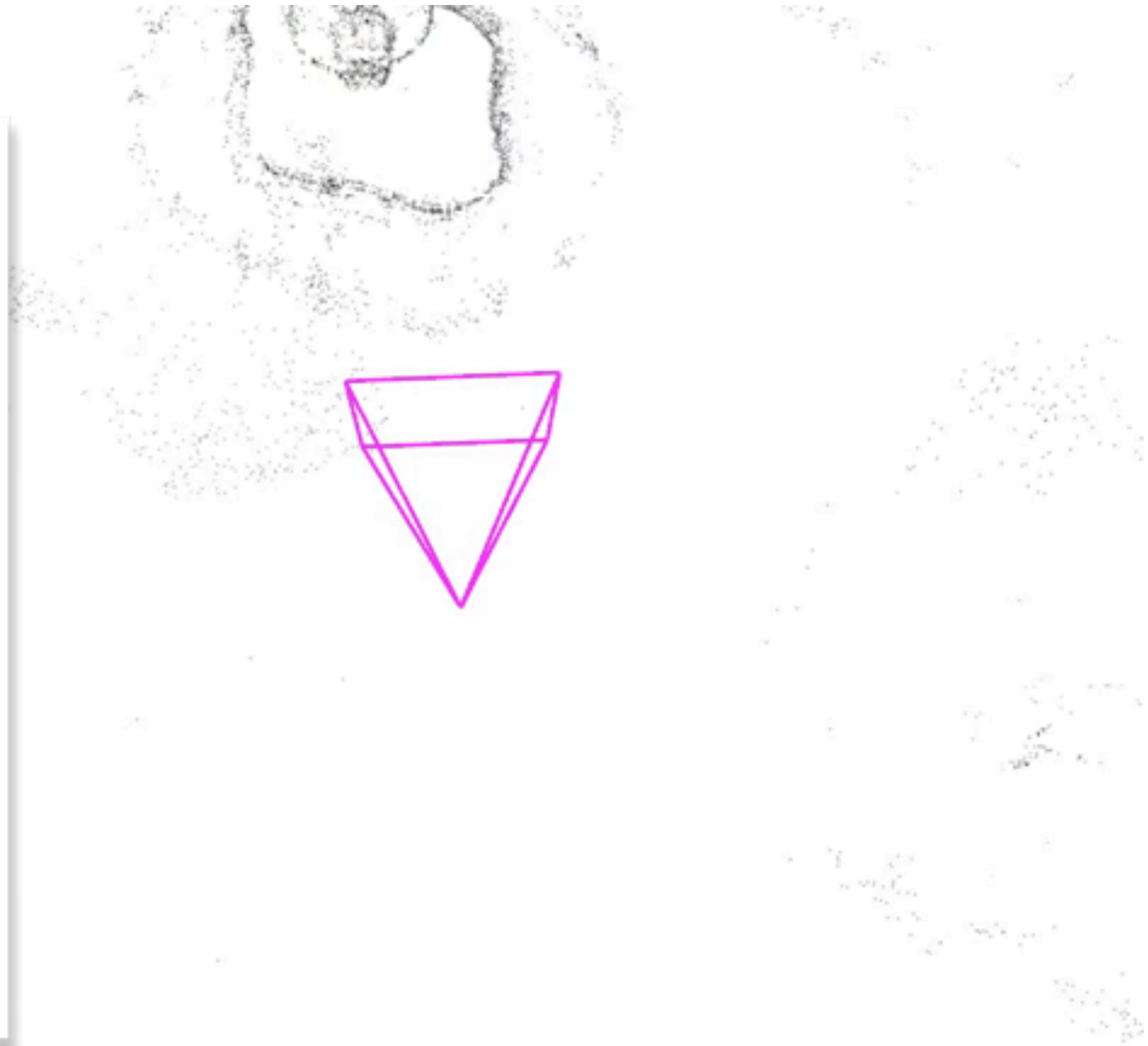
The Image-Based Localization Problem



Compute **exact position and orientation** of query image relative to 3D scene model.

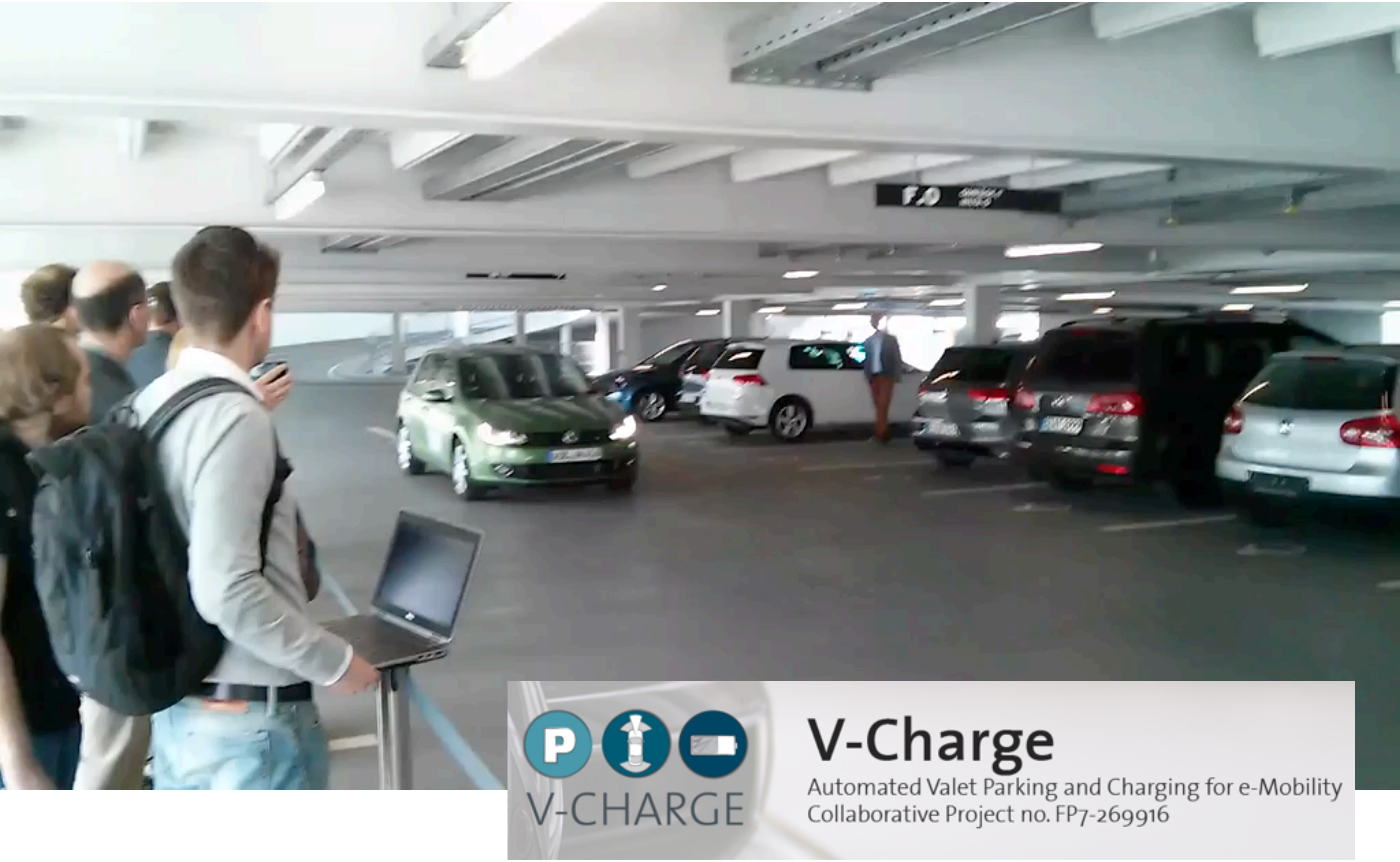
Computer Vision
and Geometry Group
ETH zürich

The Image-Based Localization Problem

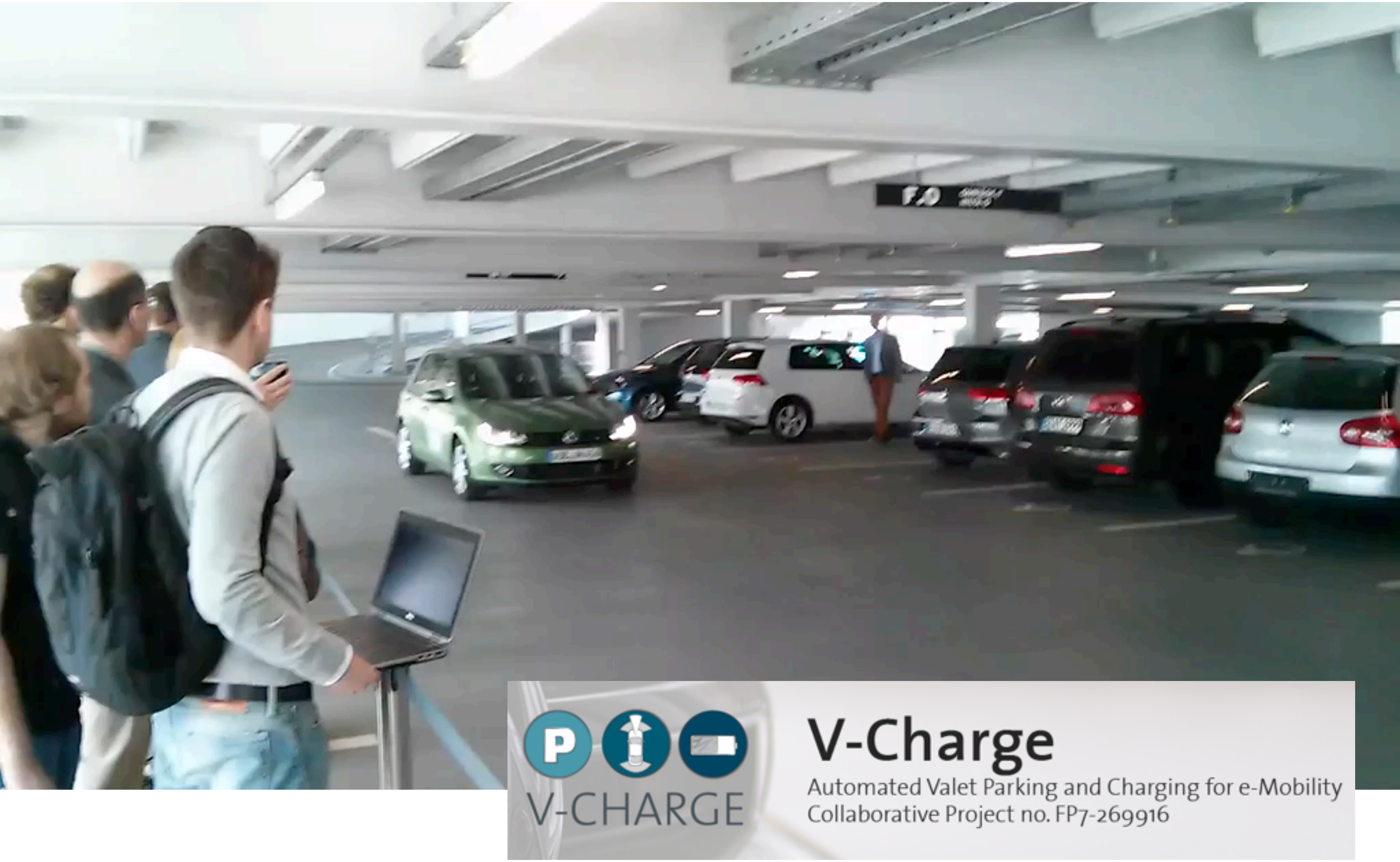


Compute **exact position and orientation** of query image relative to 3D scene model.

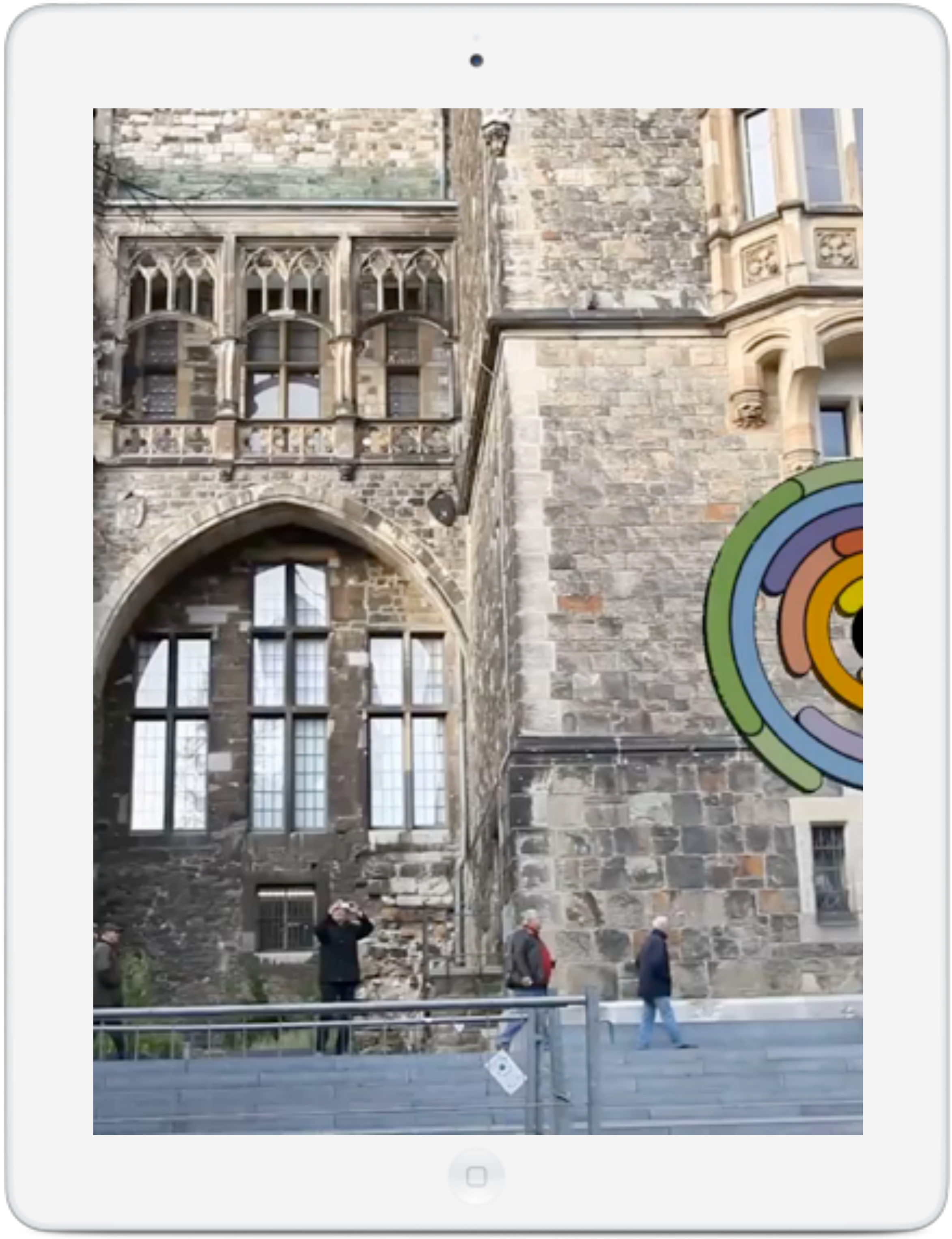
Why Image-Based Localization?



Why Image-Based Localization?

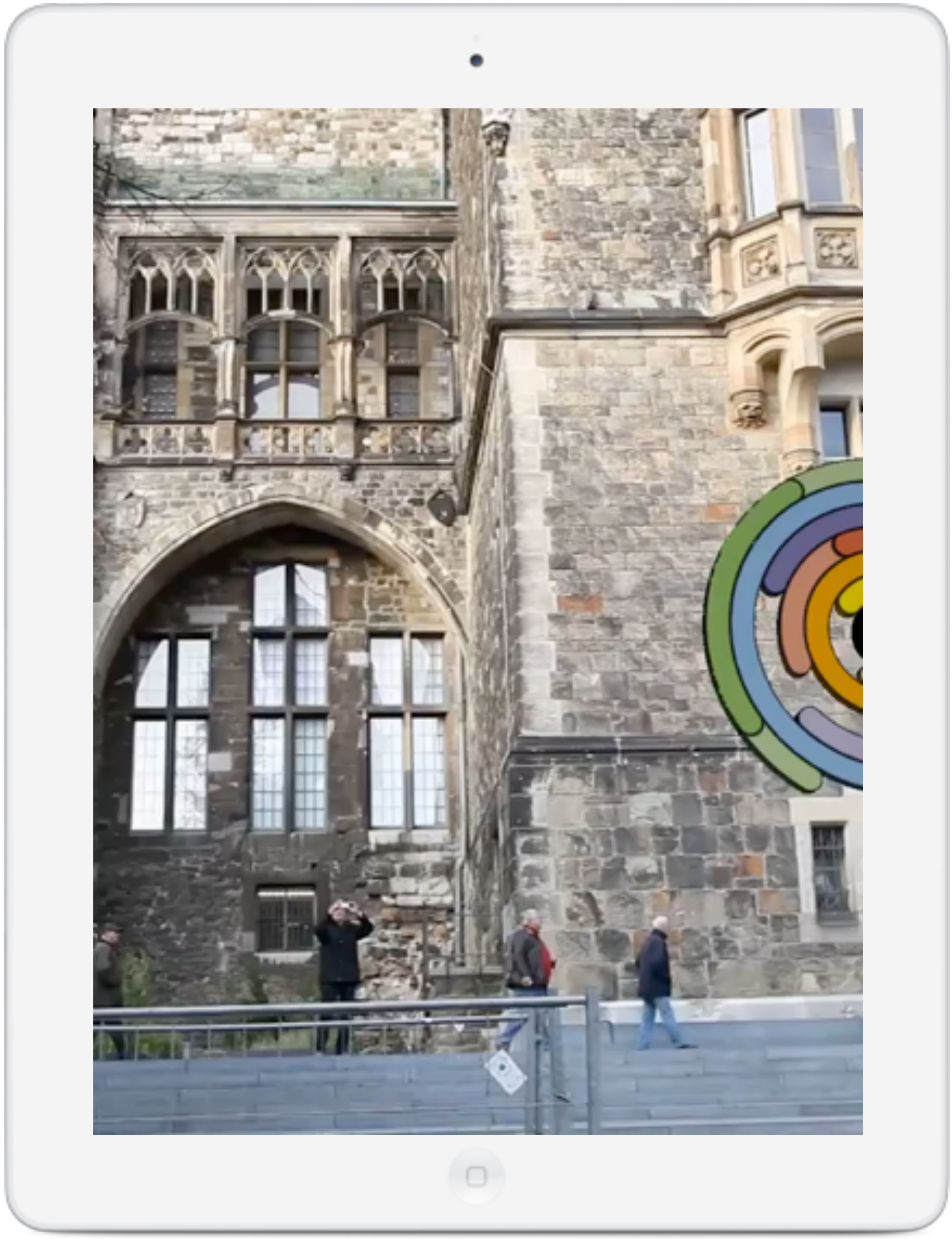


Why Image-Based Localization?



[Middelberg et al., ECCV'14]

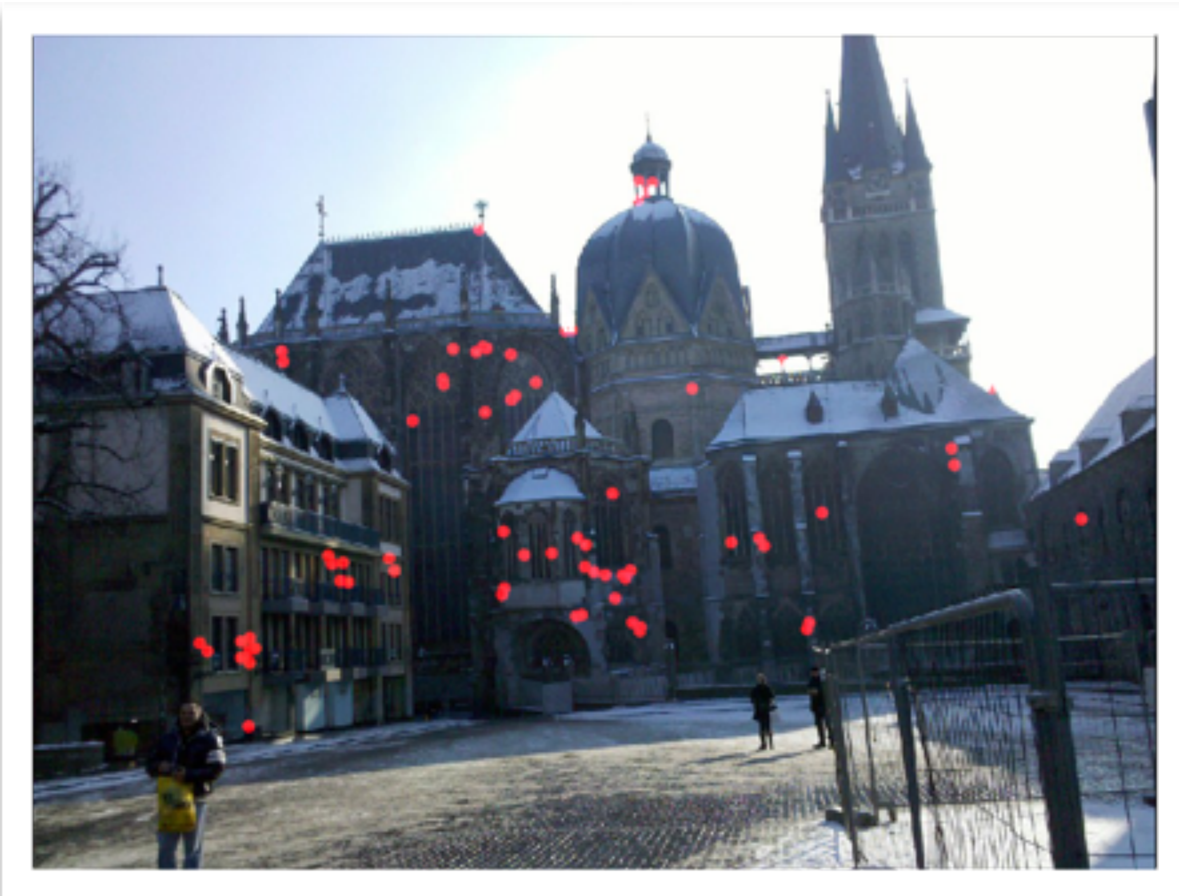
Why Image-Based Localization?



[Middelberg et al., ECCV'14]

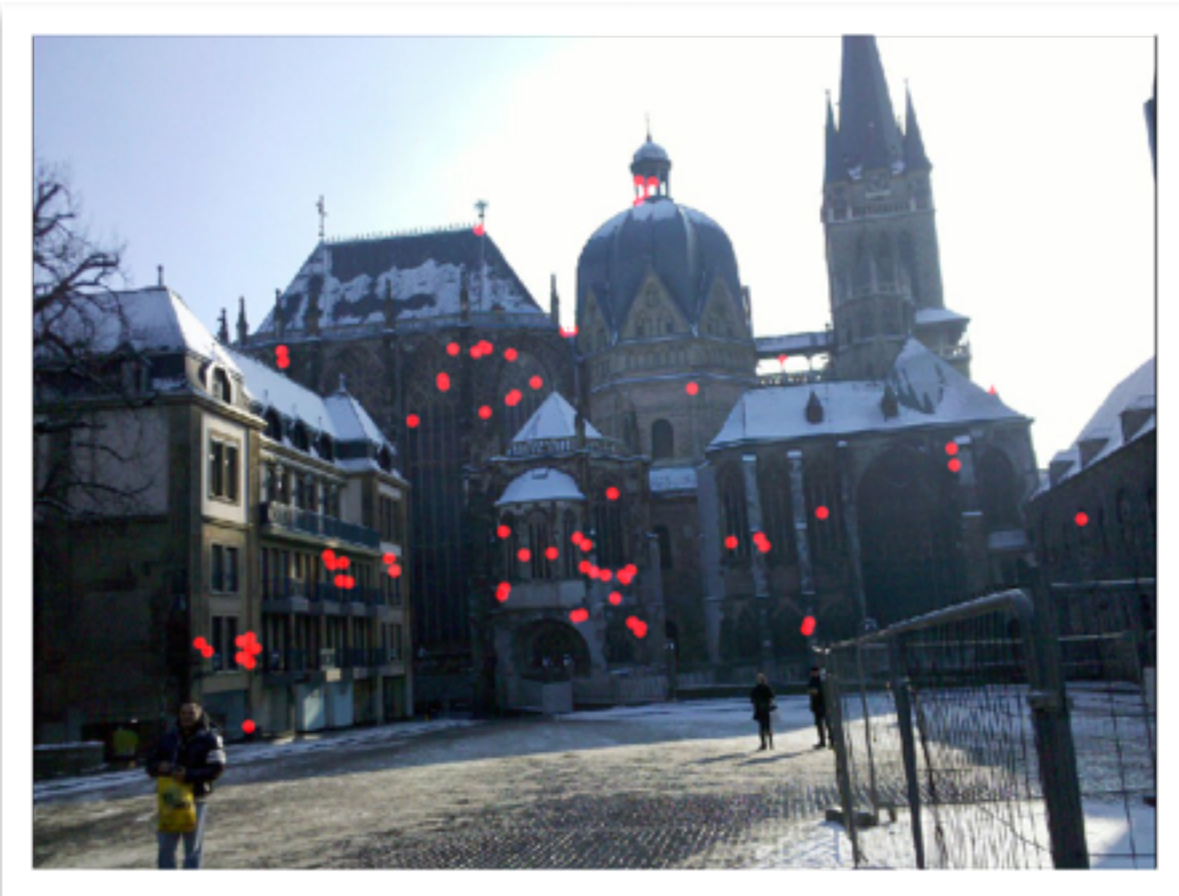
Image-Based Localization Pipeline

Image-Based Localization Pipeline



Extract Local Features

Image-Based Localization Pipeline



Extract Local Features

Establish 2D-3D Matches

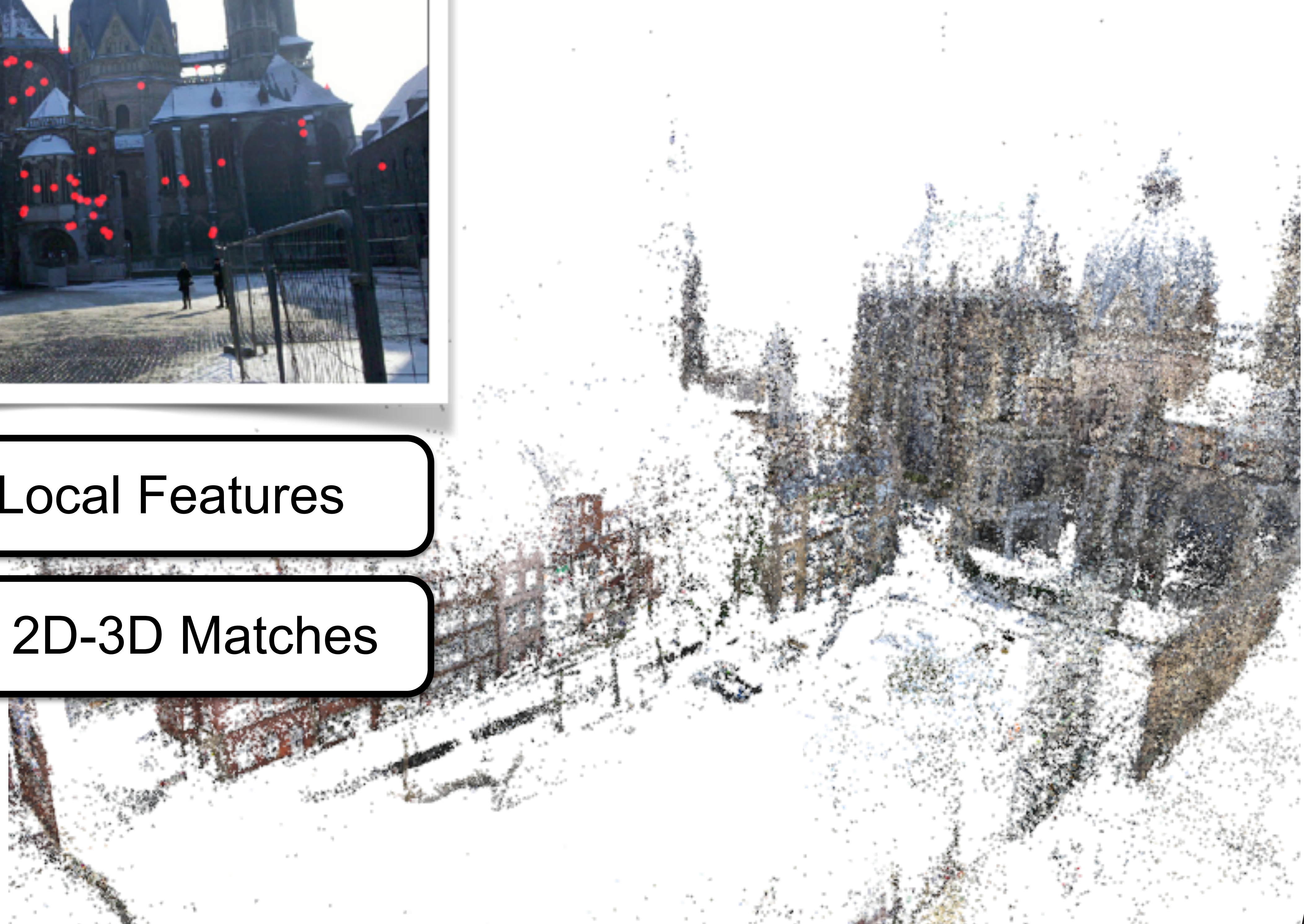
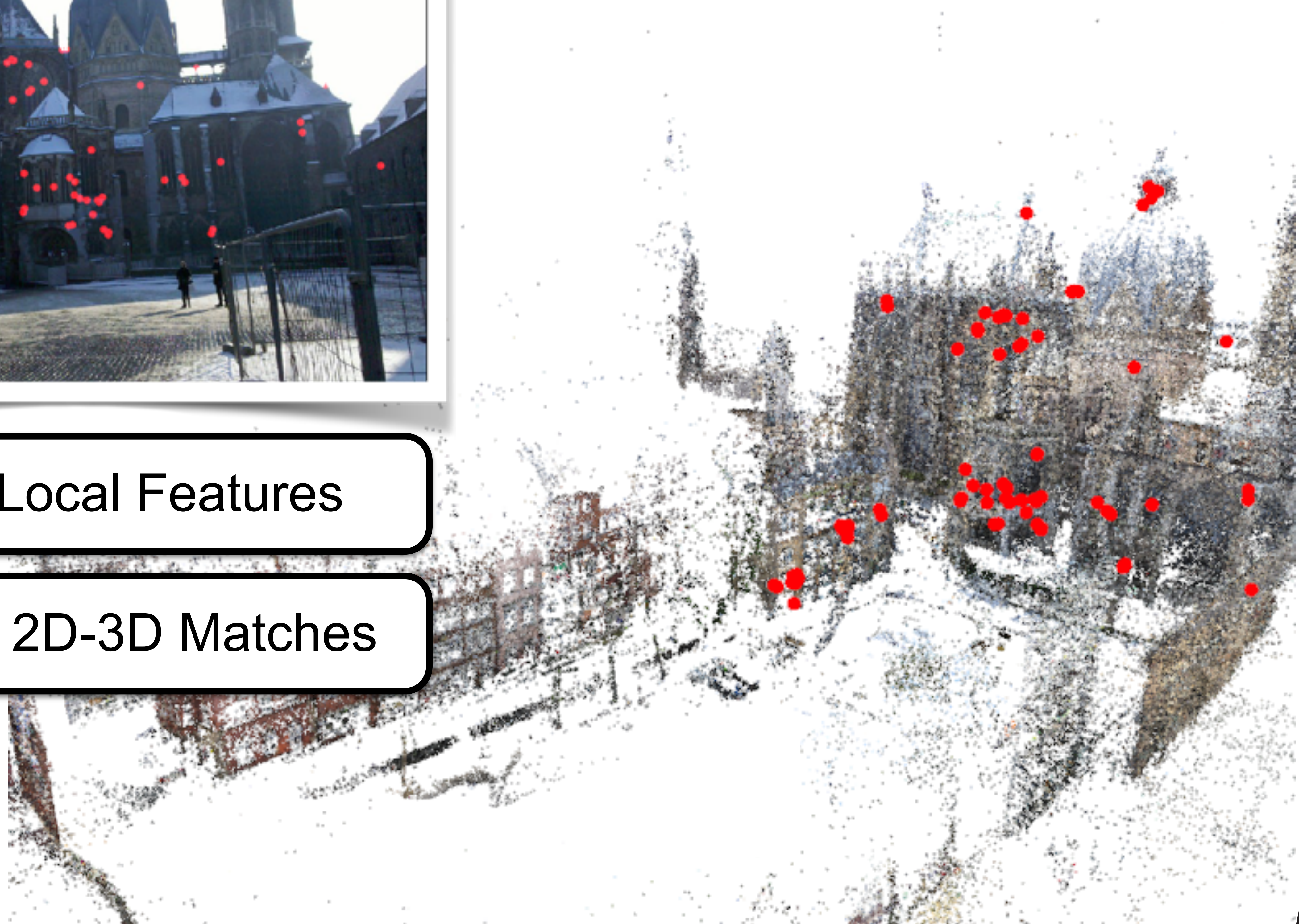


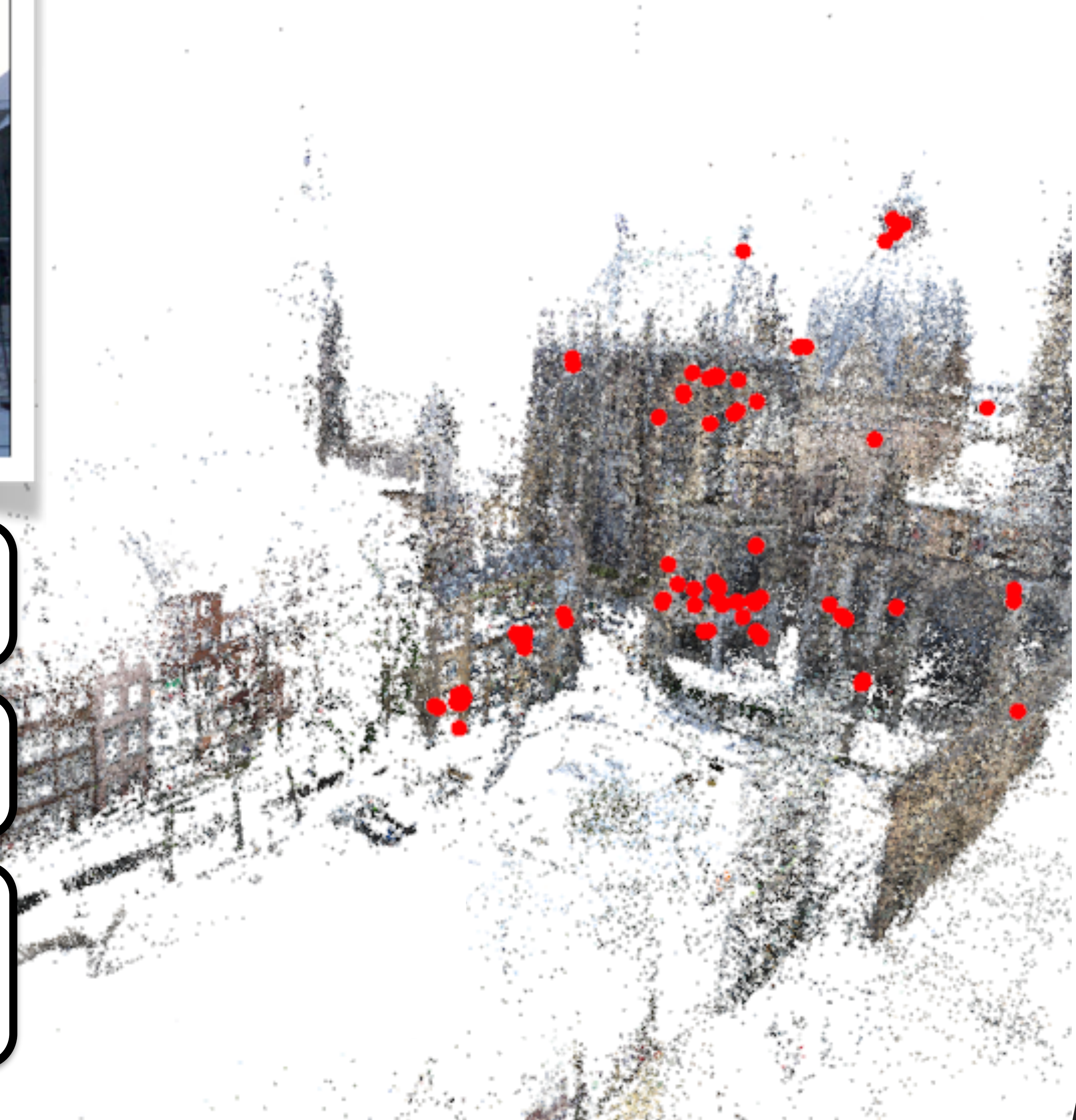
Image-Based Localization Pipeline



Extract Local Features

Establish 2D-3D Matches

Image-Based Localization Pipeline



Extract Local Features

Establish 2D-3D Matches

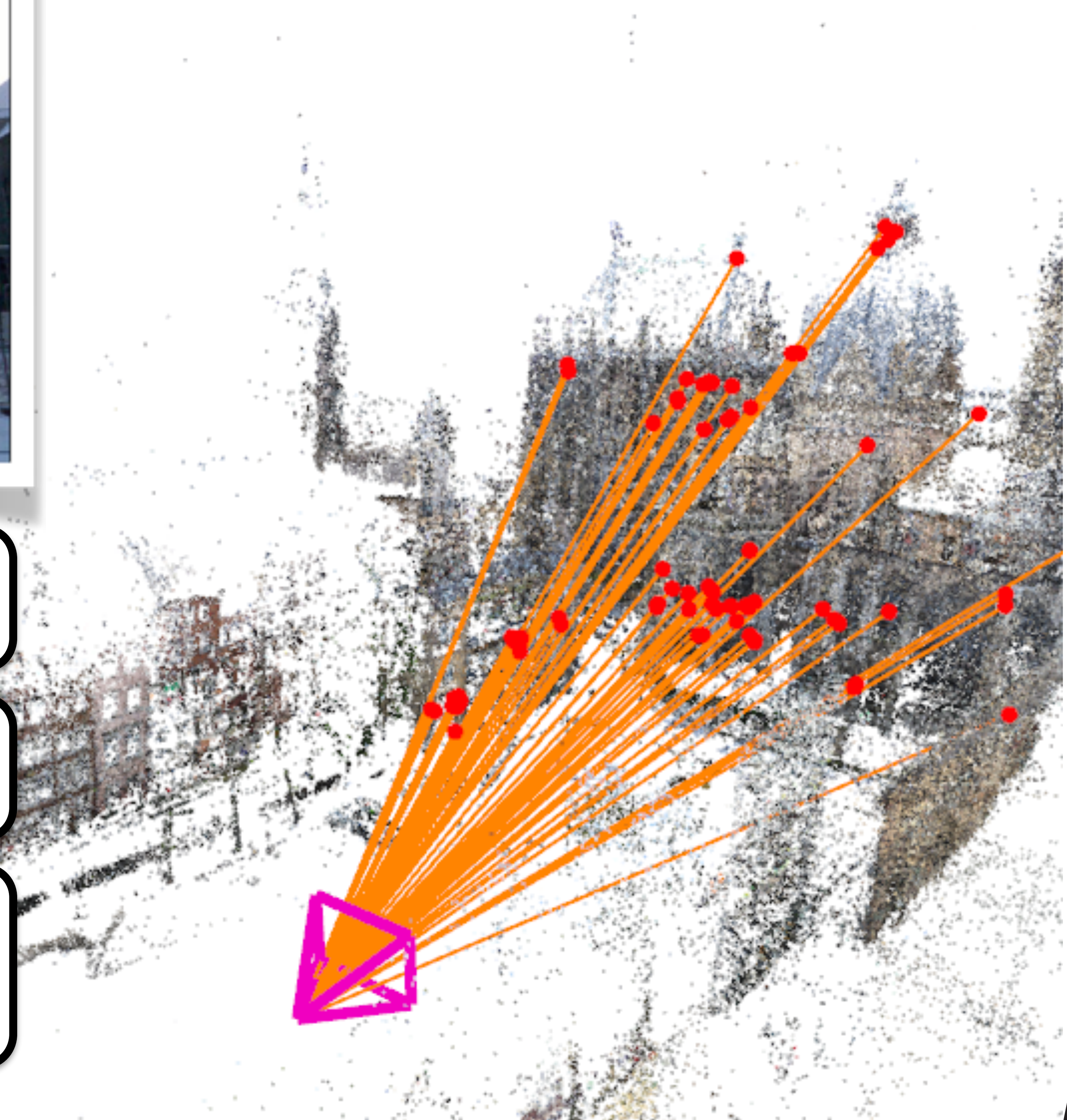
Camera Pose Estimation:
RANSAC + n-Point-Pose Algorithm

Image-Based Localization Pipeline

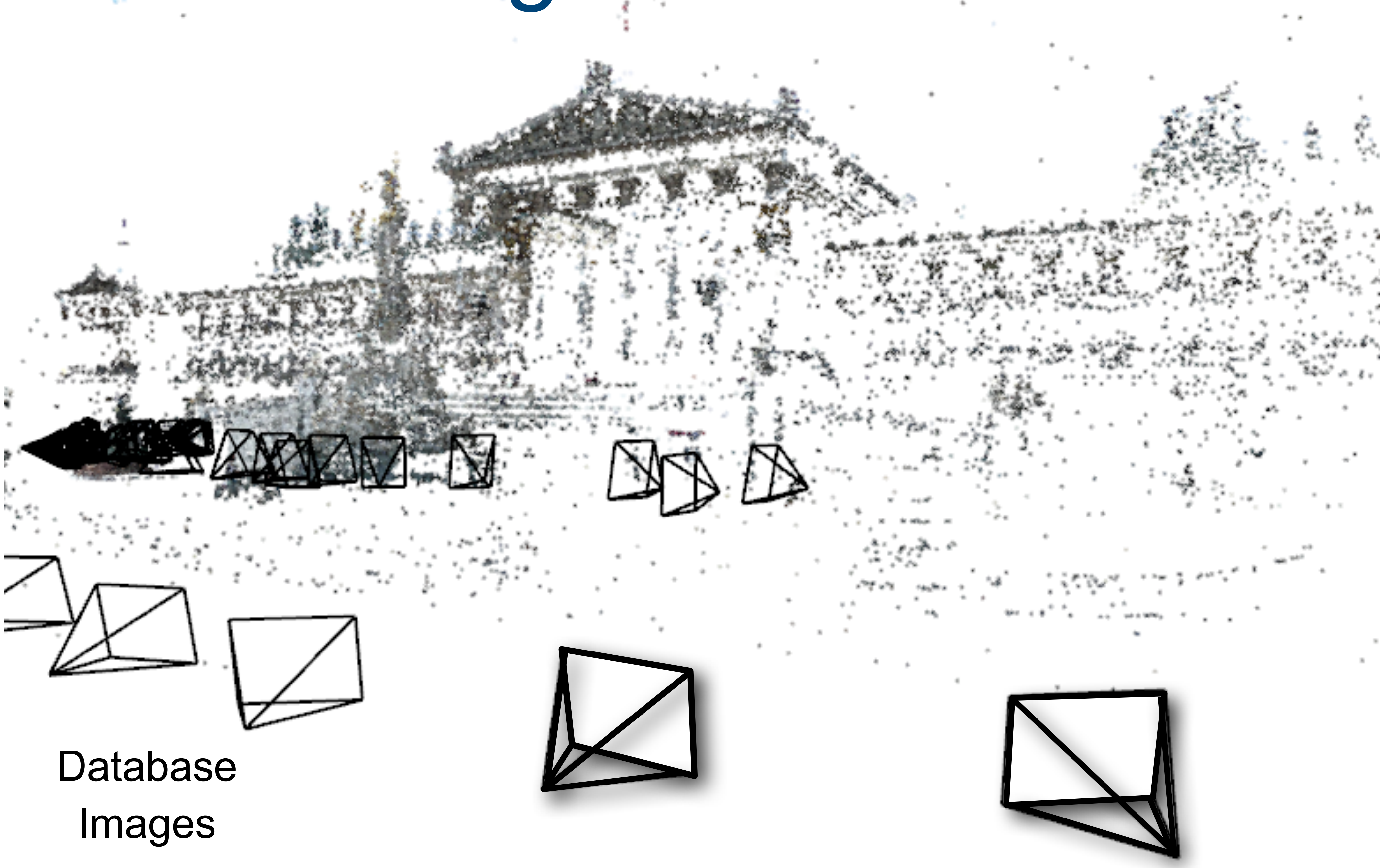
Extract Local Features

Establish 2D-3D Matches

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

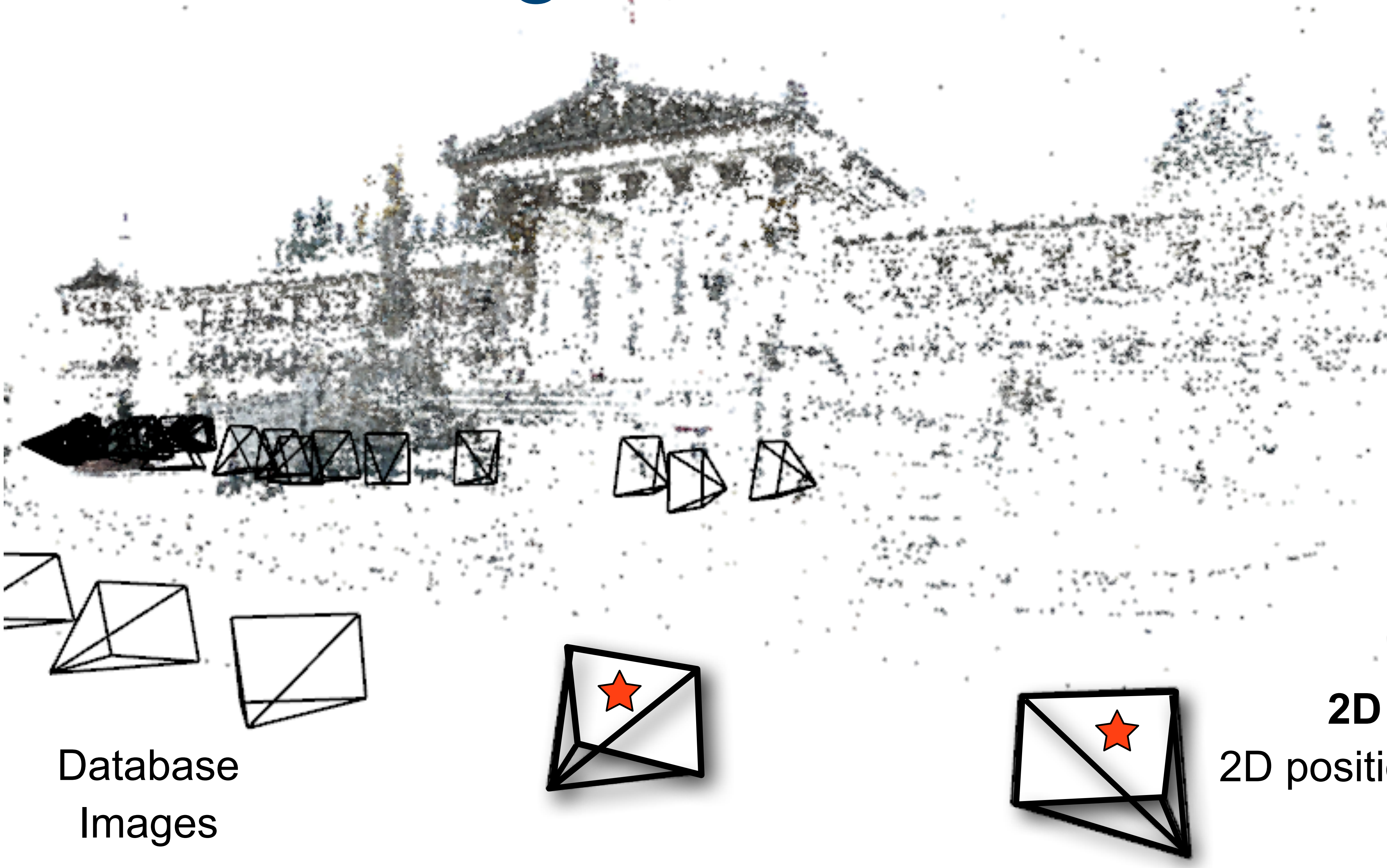


Establishing 2D-3D Matches



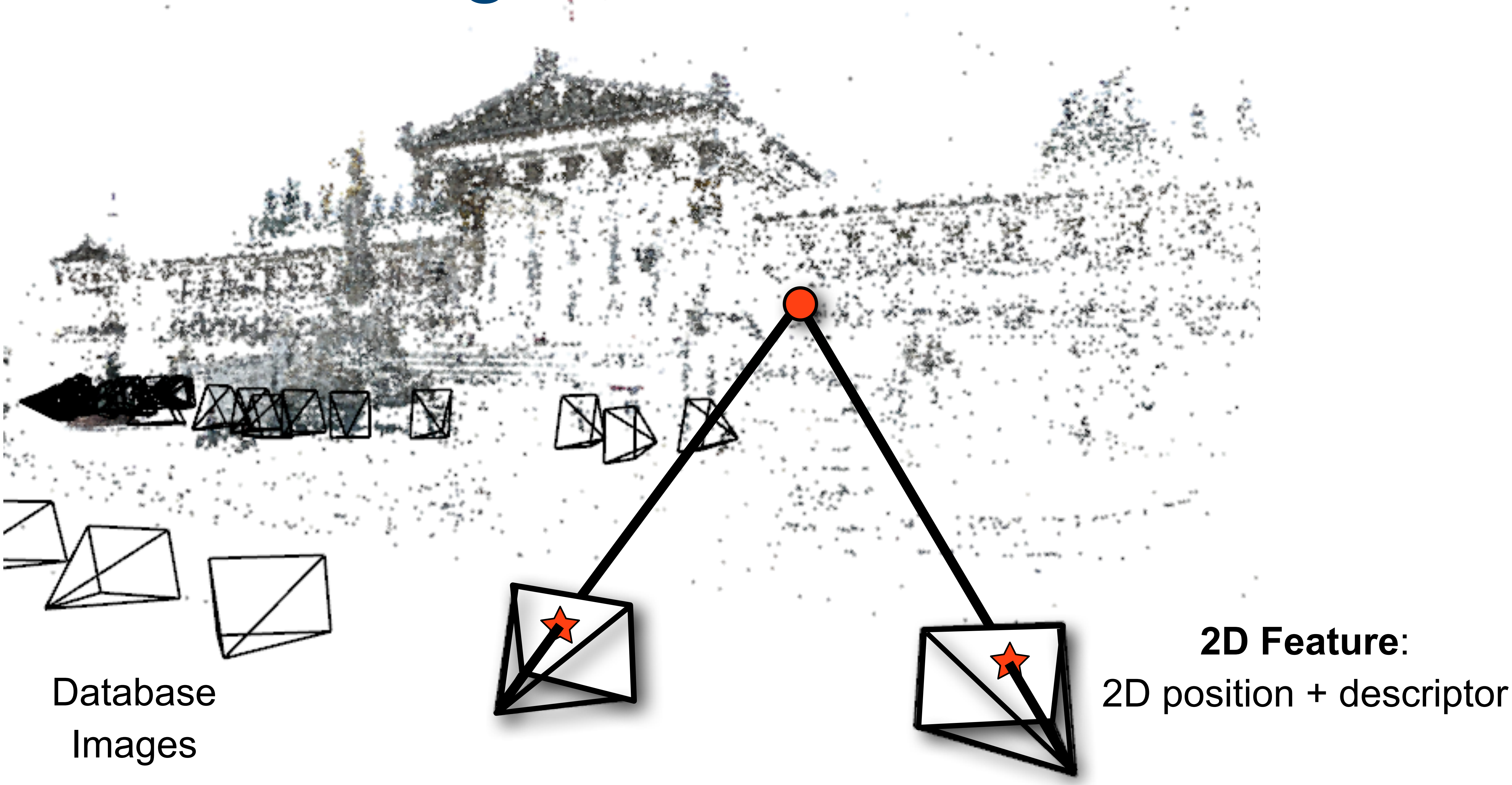
- 3D model from SfM

Establishing 2D-3D Matches



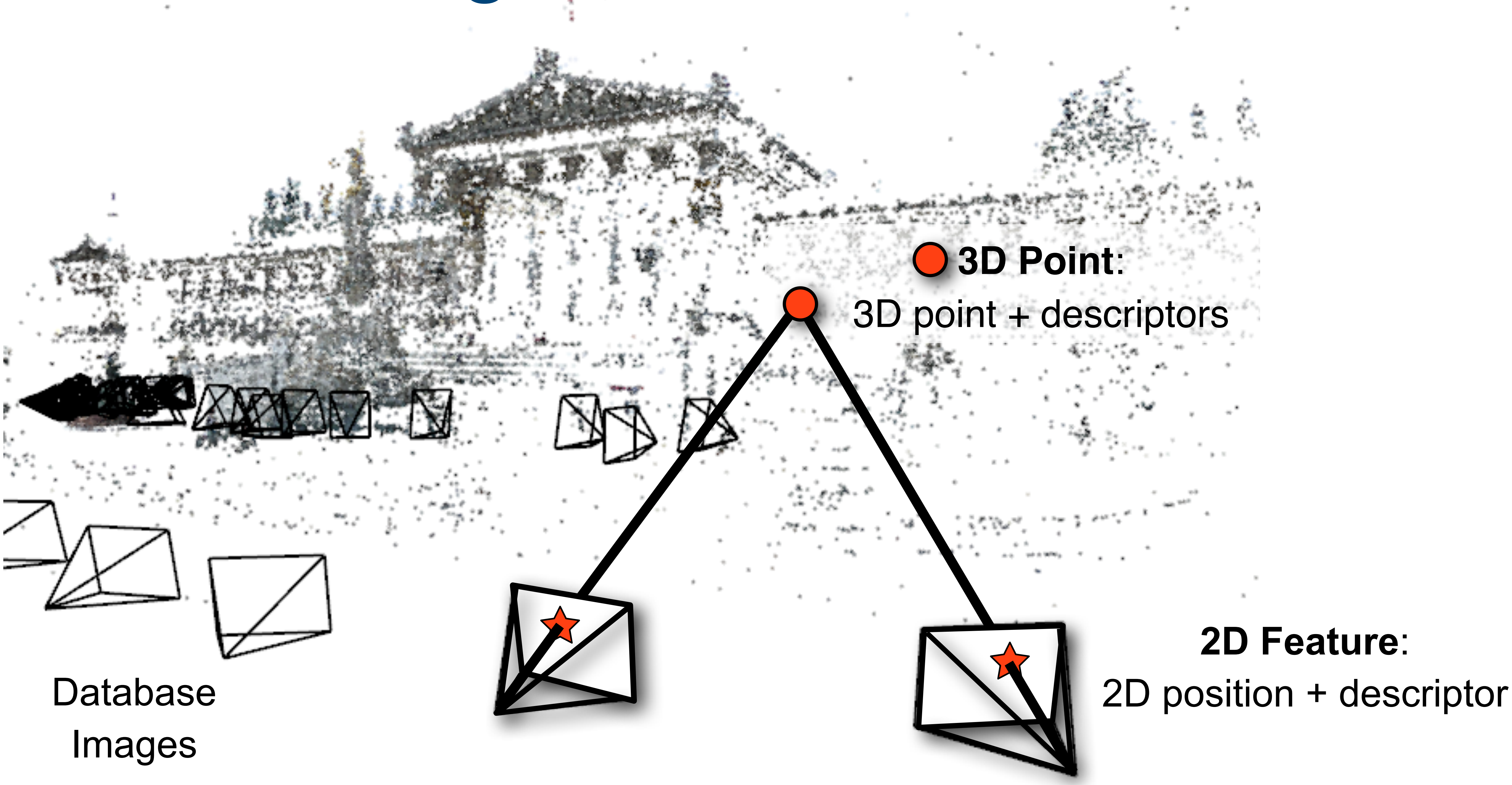
- 3D model from SfM

Establishing 2D-3D Matches



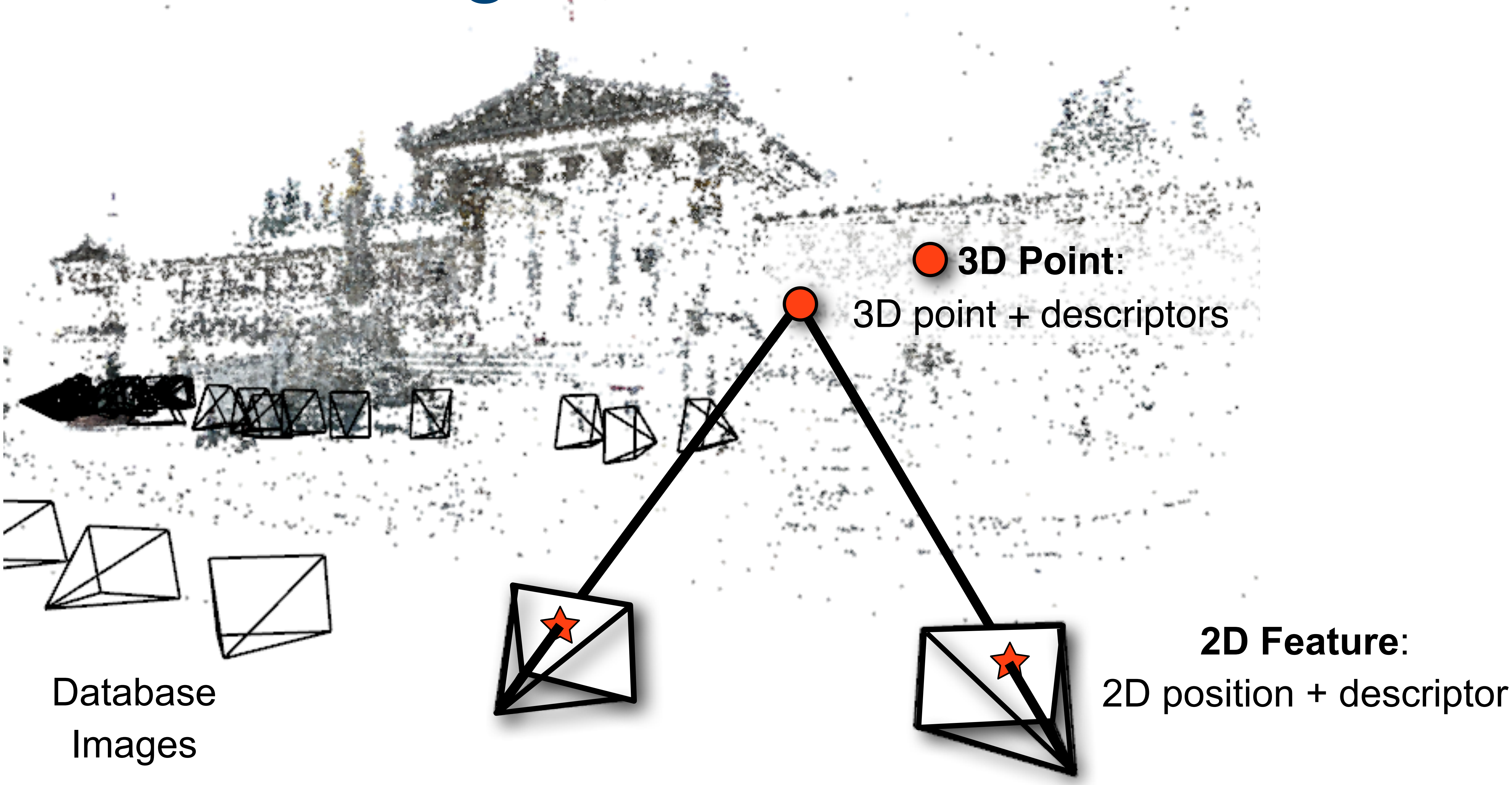
- 3D model from SfM

Establishing 2D-3D Matches



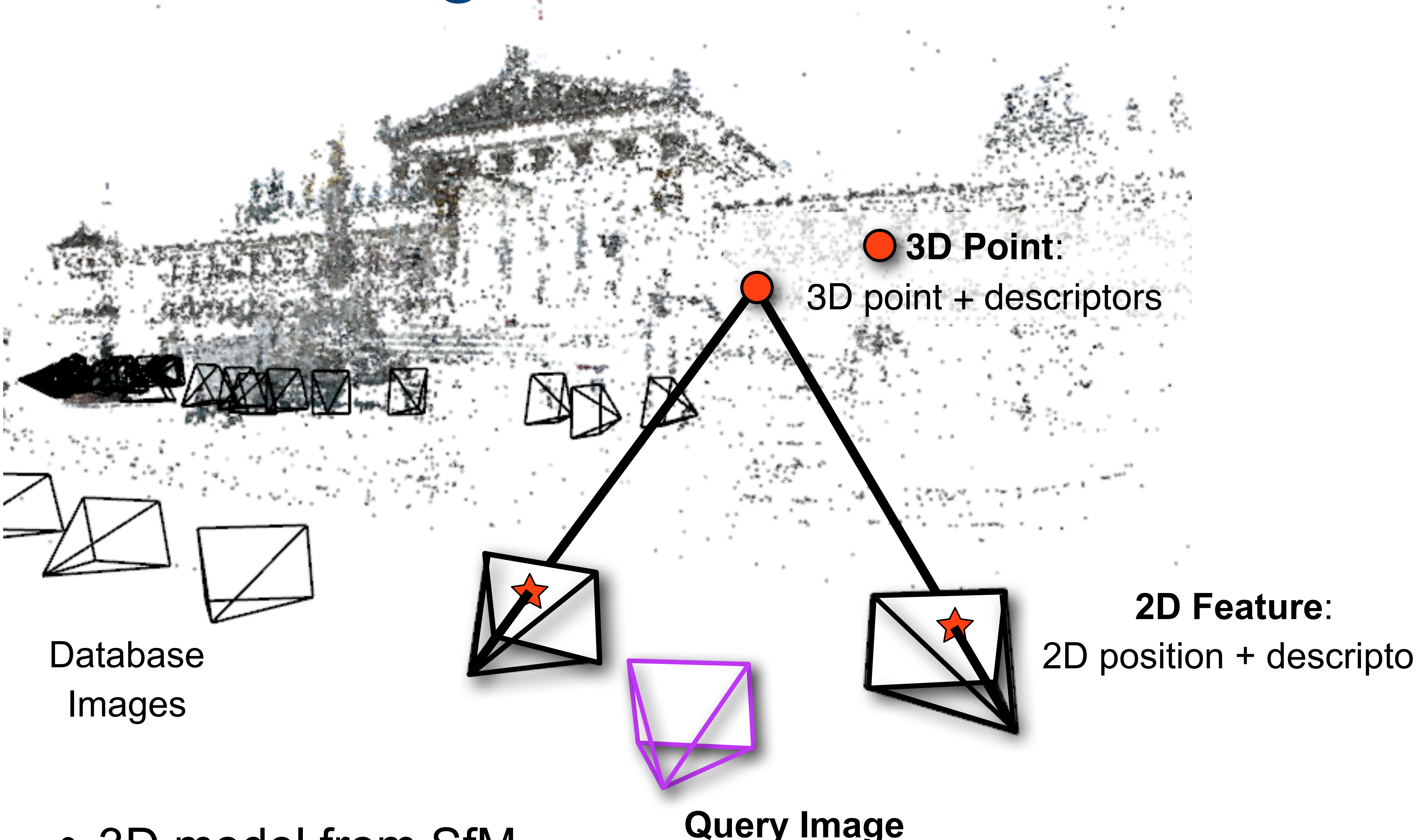
- 3D model from SfM

Establishing 2D-3D Matches



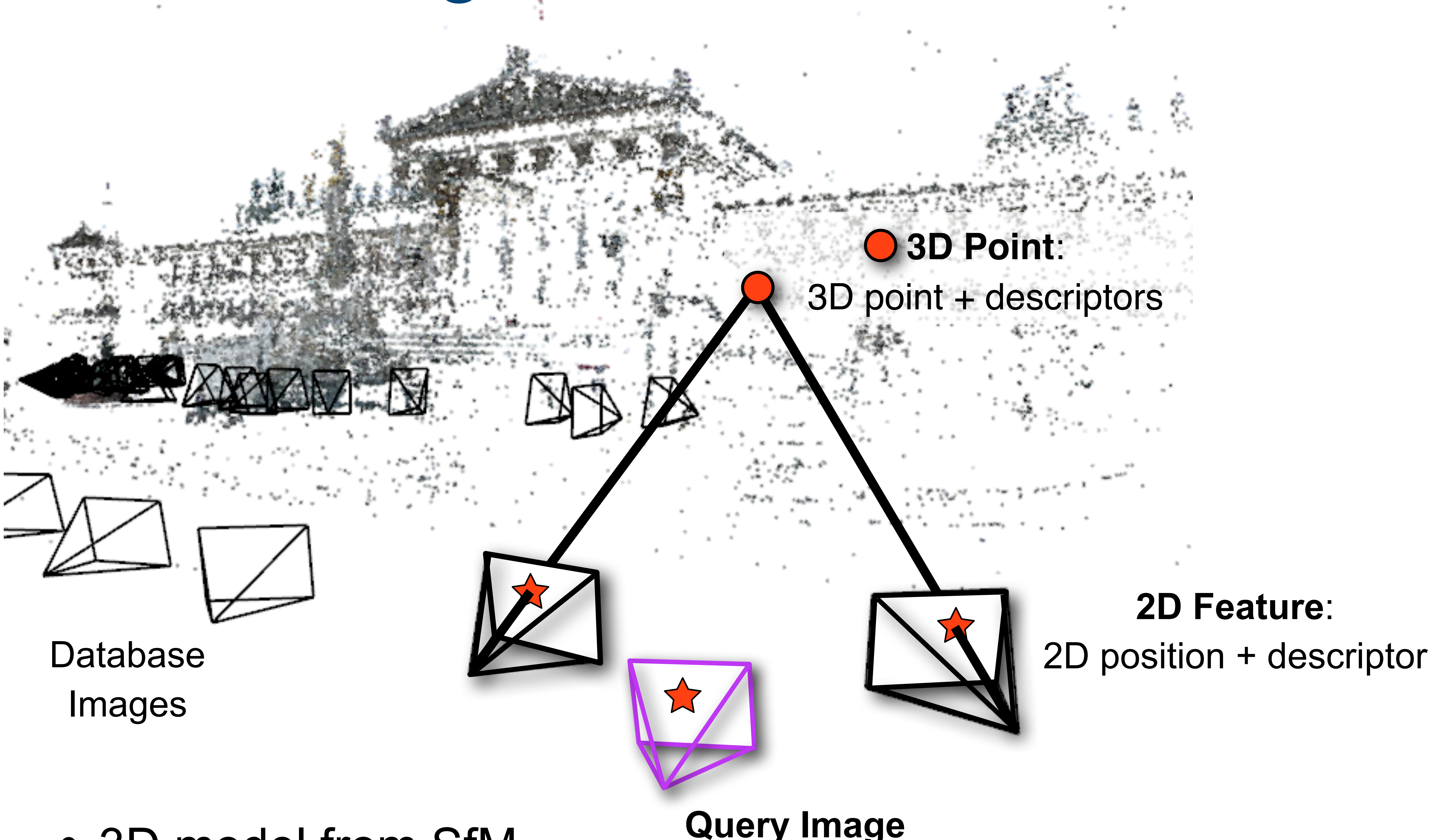
- 3D model from SfM
- 2D-3D correspondences from (SIFT) **descriptor matching**

Establishing 2D-3D Matches



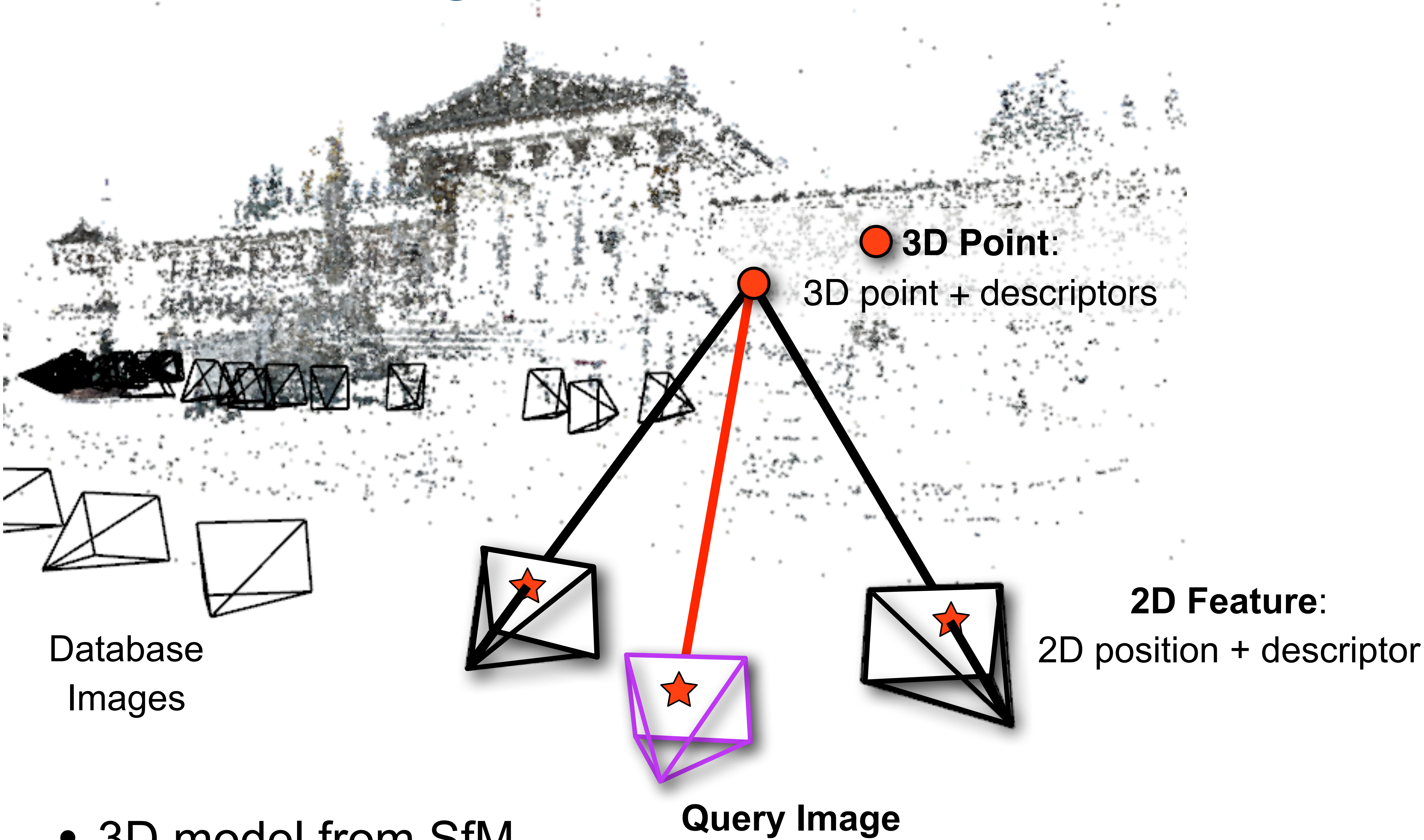
- 3D model from SfM
- 2D-3D correspondences from (SIFT) **descriptor matching**

Establishing 2D-3D Matches



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

Establishing 2D-3D Matches



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

Challenges

- **Efficiency**: Quickly localize query images

Challenges

- **Efficiency**: Quickly localize query images

- **Effectiveness**: Localize all query images

Challenges

- **Efficiency**: Quickly localize query images
- **Effectiveness**: Localize all query images
- **Accuracy**: Accurately recover camera pose

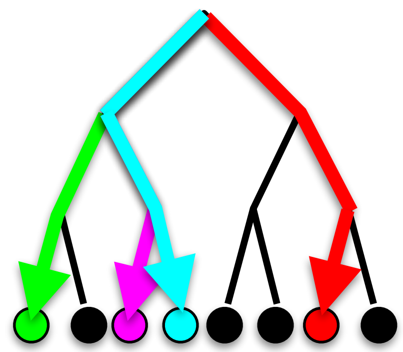
Overview

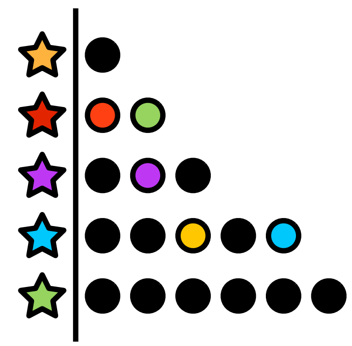
- Efficient & Effective Large-Scale Localization
- Real-Time Mobile Localization
- Open Challenges

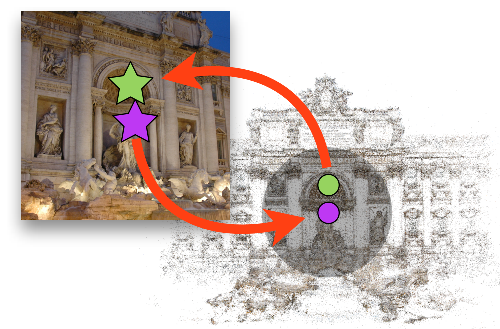
Overview

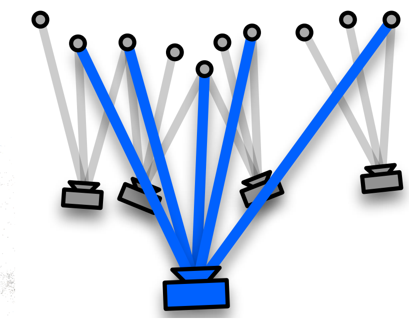
- Efficient & Effective Large-Scale Localization
- Real-Time Mobile Localization
- Open Challenges

Localization - Overview



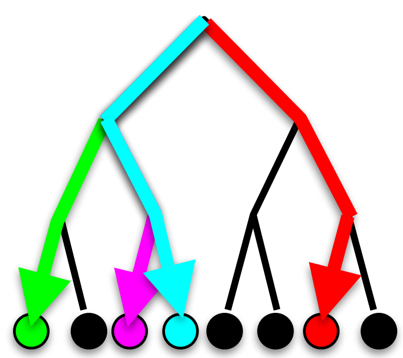






	Baseline: kd-tree search	**VPS**	**Active Search**	**+ Visibility Filtering**
	[Sattler et al., ICCV'11]	[Sattler et al., ICCV'11]	[Sattler et al., ECCV'12]	
effectiveness	✓	✗	✓✓	✓✓
efficiency	✗ ✗	✓	✗	✓

Localization - Overview



Baseline: kd-tree search

[Sattler et al., ICCV'11]

VPS

[Sattler et al., ICCV'11]

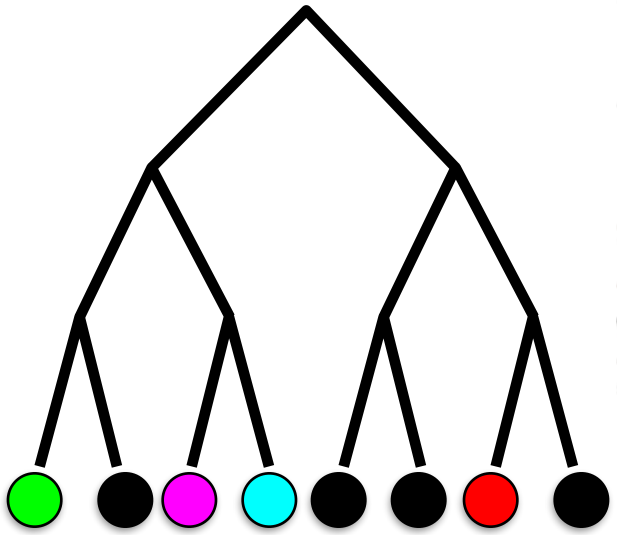
Active Search

+ Visibility Filtering

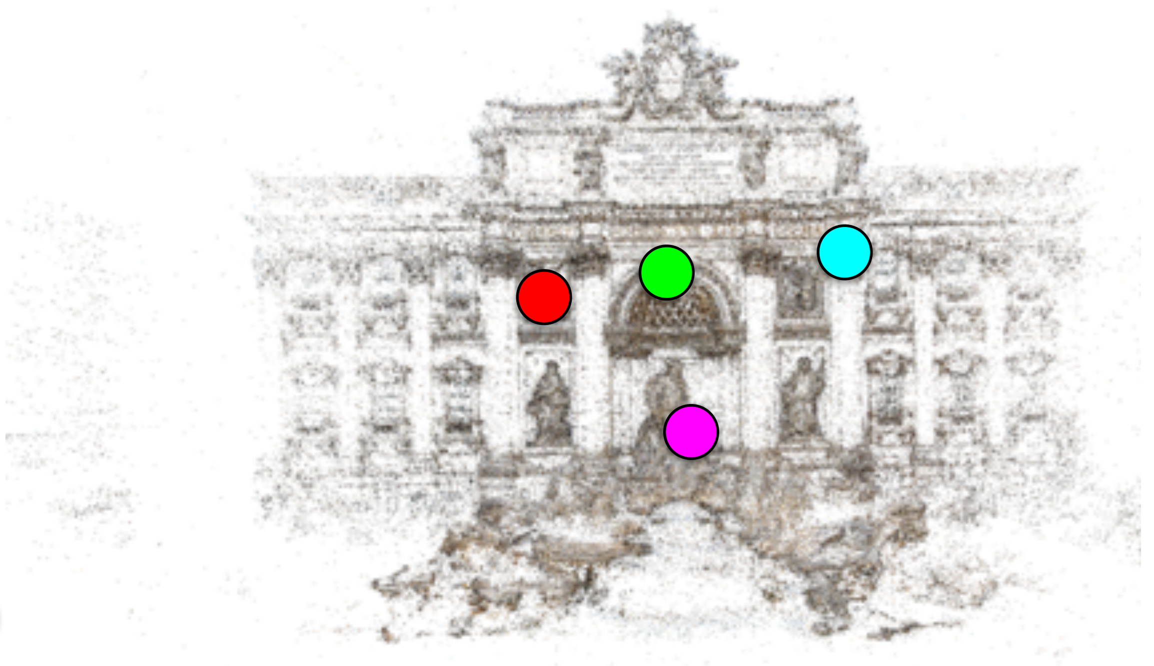
[Sattler et al., ECCV'12]

	Baseline: kd-tree search	VPS	Active Search	+ Visibility Filtering
effectiveness	✓	✗	✓✓	✓✓
efficiency	✗ ✗	✓	✗	✓

Baseline: Tree-Based Search



kd-tree

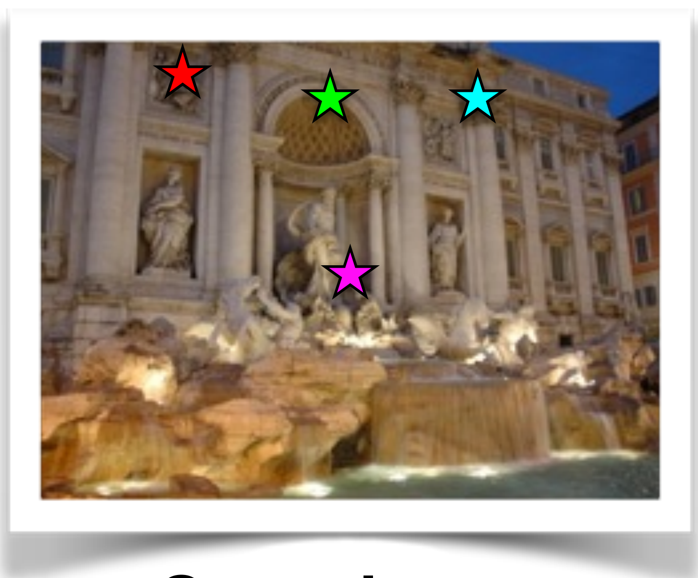


3D Model

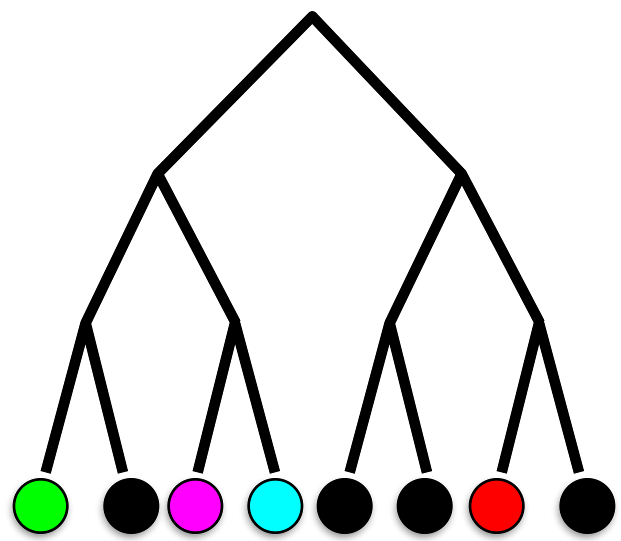
[Sattler et al., ICCV'11] [code]

photo credit: Ming Li

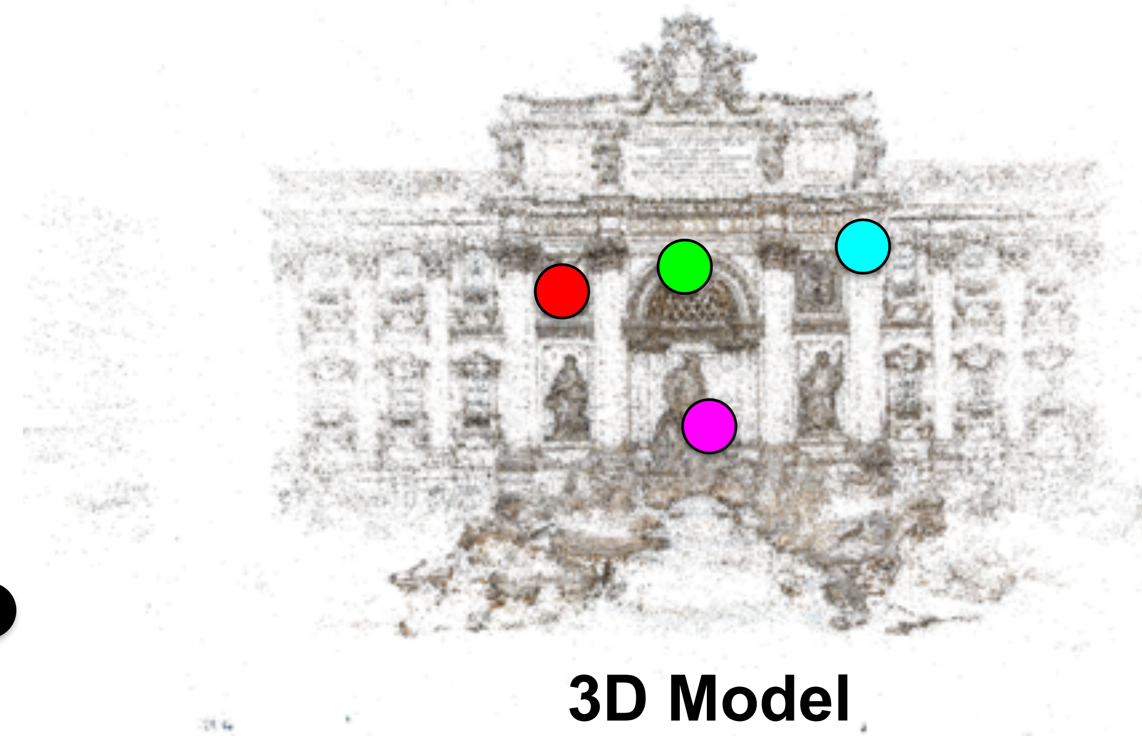
Baseline: Tree-Based Search



Query Image



kd-tree

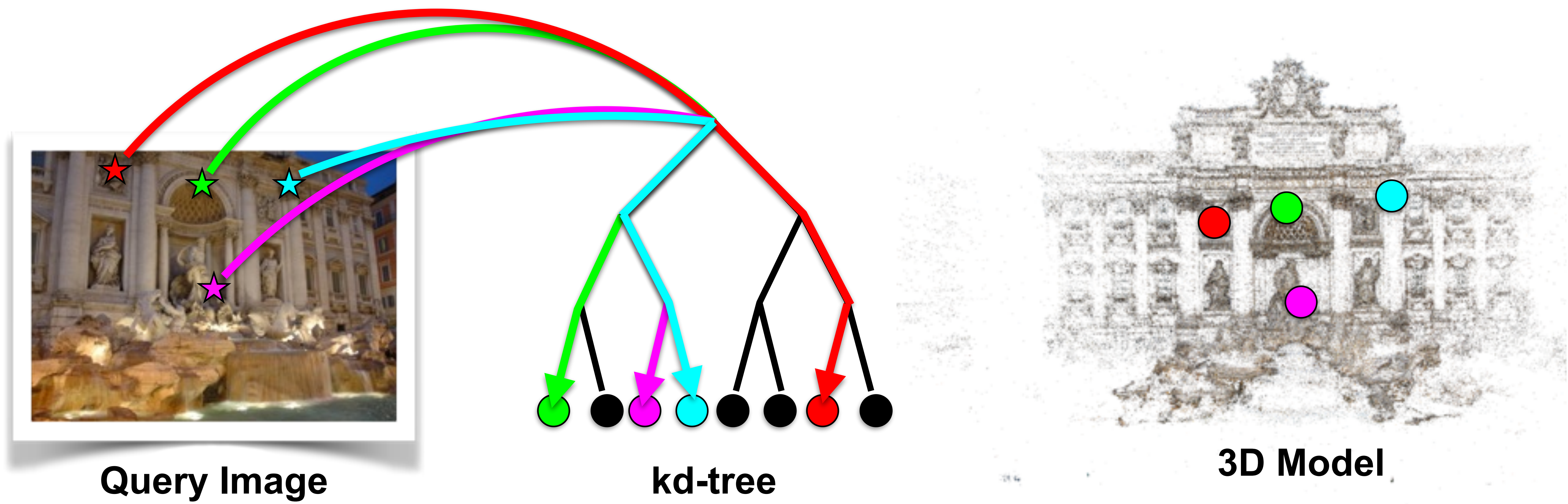


3D Model

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

photo credit: Ming Li

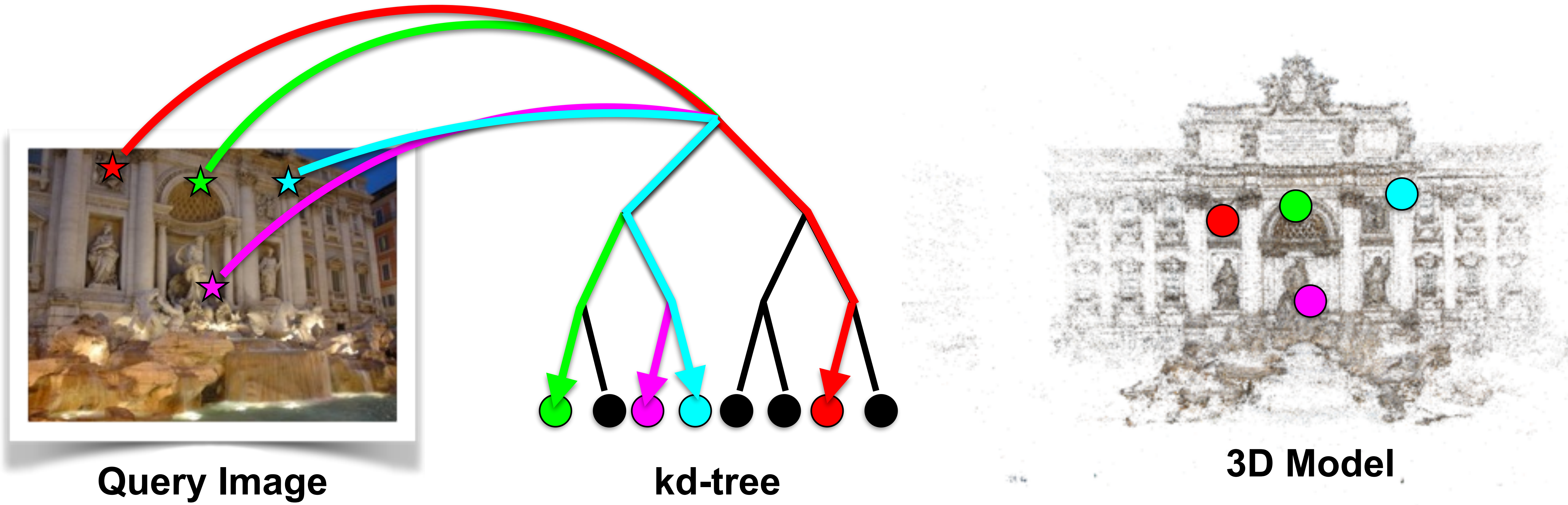
Baseline: Tree-Based Search



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

photo credit: Ming Li

Baseline: Tree-Based Search



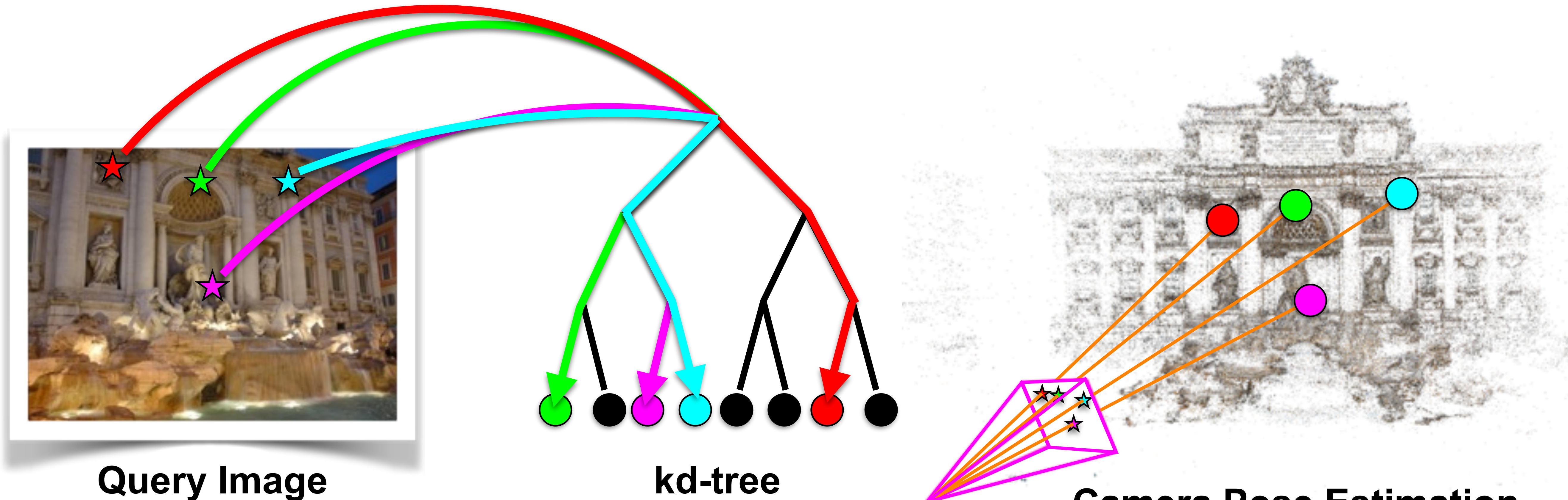
2-nn search & ratio test

$$\frac{||d - d_1||_2}{||d - d_2||_2} < 0.7$$

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

photo credit: Ming Li

Baseline: Tree-Based Search



2-nn search & ratio test

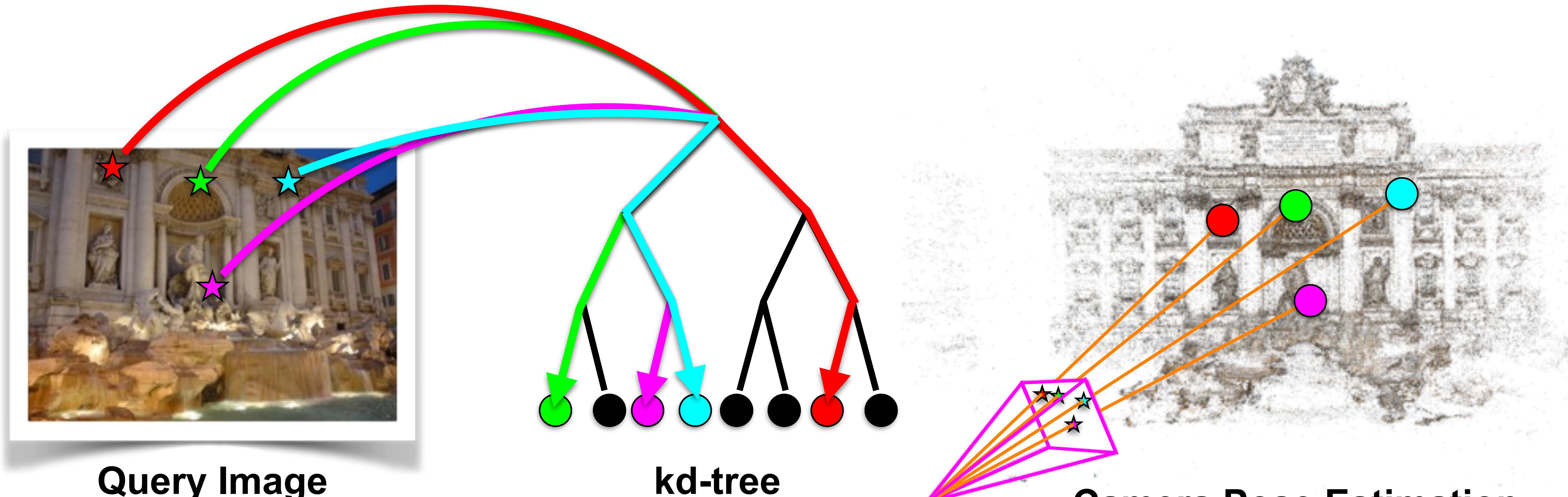
$$\frac{||d - d_1||_2}{||d - d_2||_2} < 0.7$$

LO-RANSAC + $T_{1,1}$ + p6p

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

photo credit: Ming Li

Baseline: Tree-Based Search



2-nn search & ratio test

$$\frac{||d - d_1||_2}{||d - d_2||_2} < 0.7$$

Camera Pose Estimation

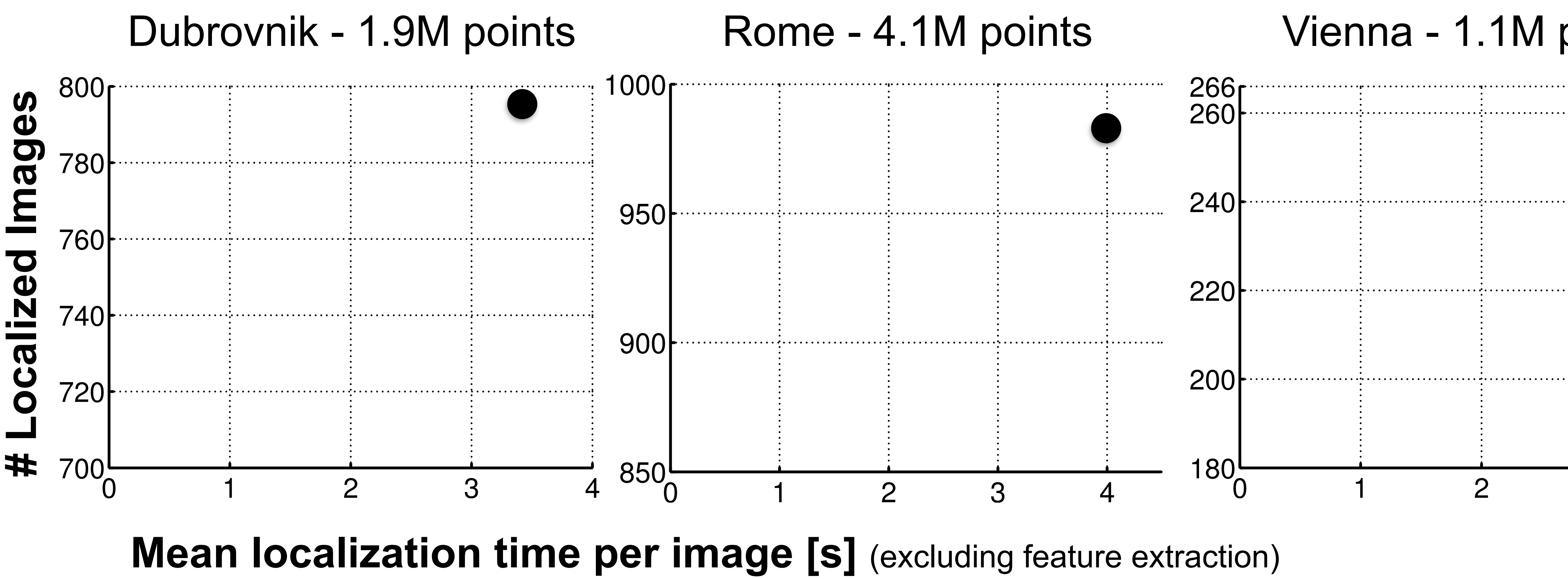
LO-RANSAC +$T_{1,1}$ + p6p

pose valid if ≥ 12 inliers

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

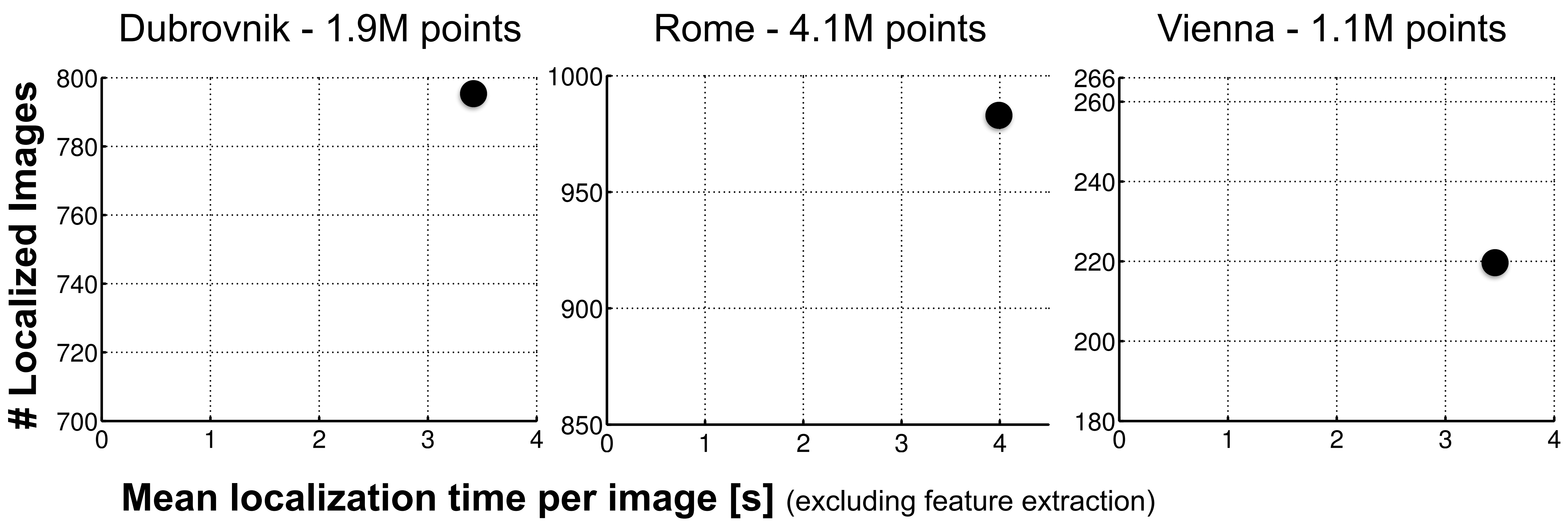
photo credit: Ming Li

Results



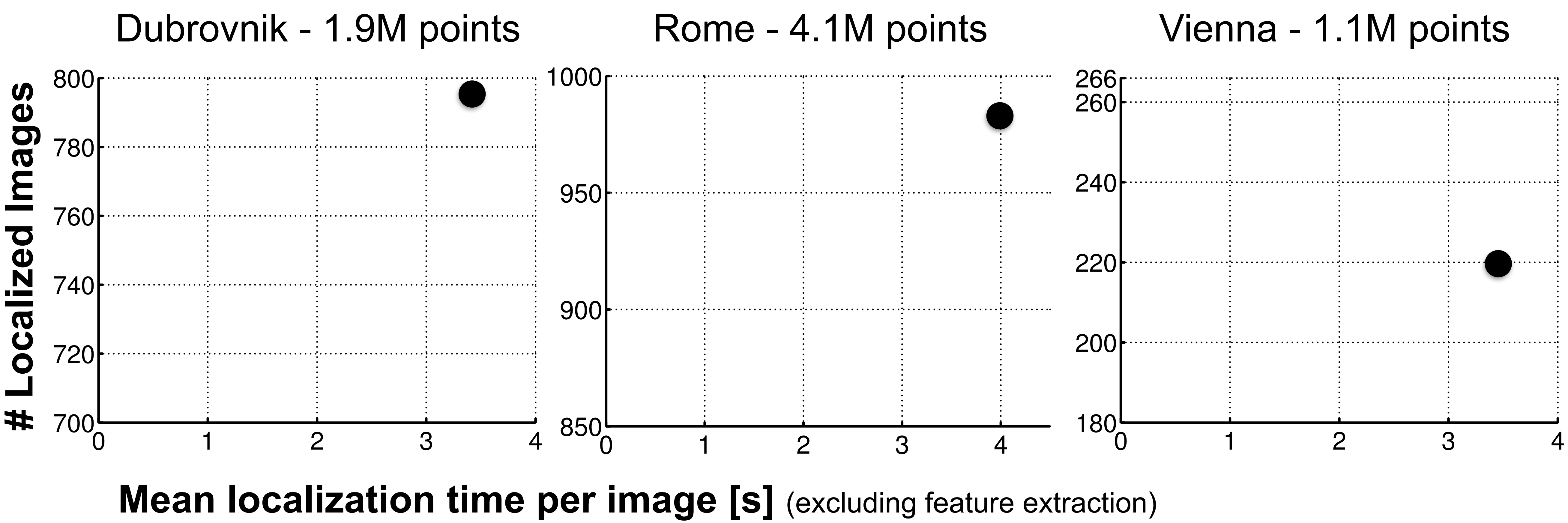
Dubrovnik - 1.9M points
Rome - 4.1M points
Vienna - 1.1M points
Localized Images
800
780
760
740
720
700
1000
950
900
850
266
260
240
220
200
180
0
1
2
3
4
Mean localization time per image [s] (excluding feature extraction)

Results



✓Excellent localization effectiveness…

Results



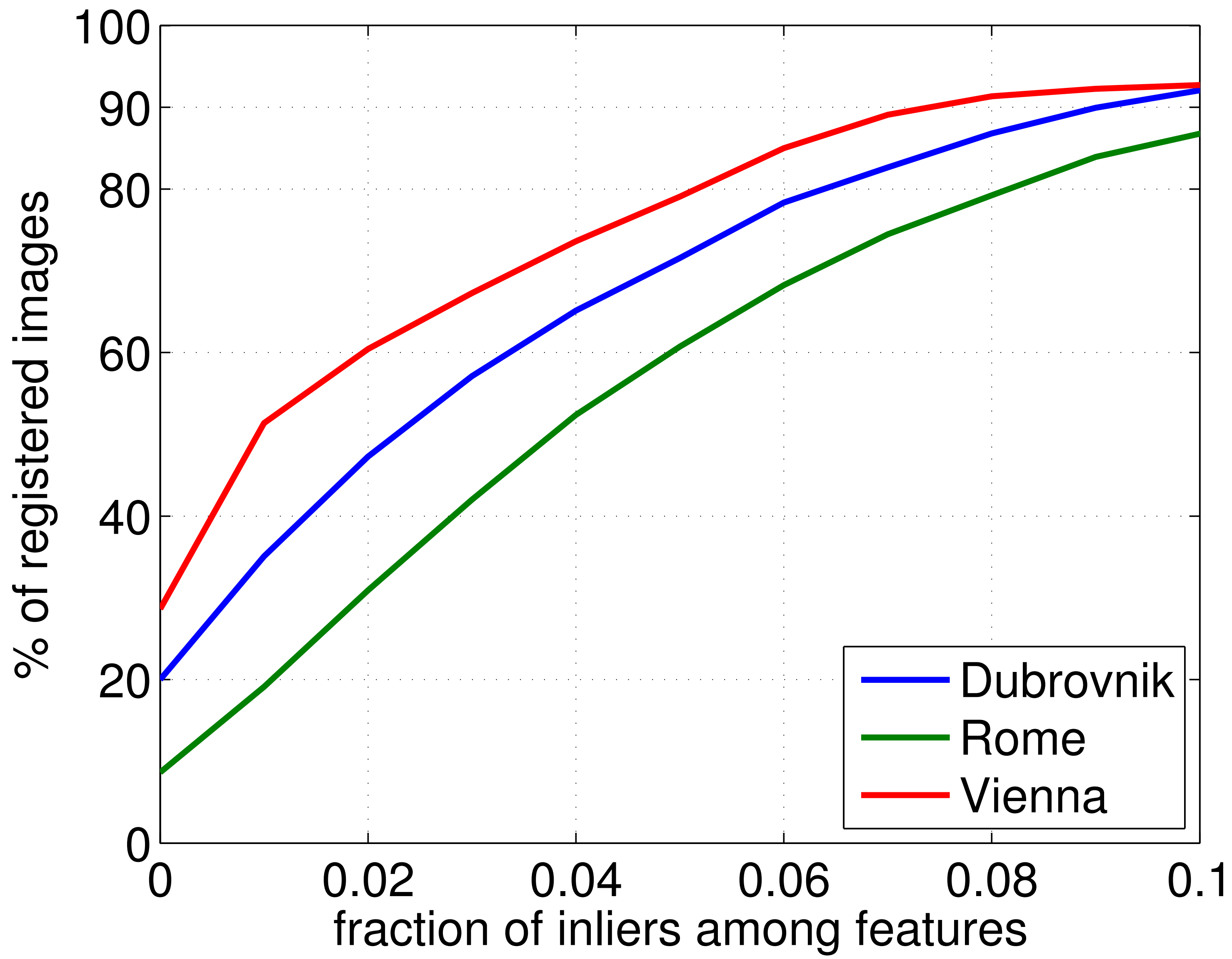
✓ Excellent localization effectiveness…

✗ … but very slow!

Potential for Faster Search

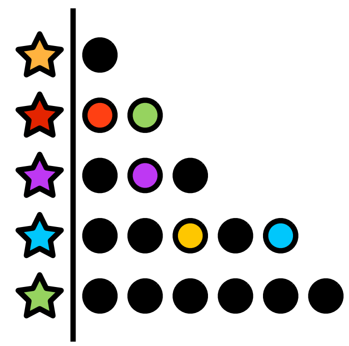
[Sattler et al., ICCV'11]

Potential for Faster Search



[Sattler et al., ICCV'11]

Localization - Overview



	Baseline: kd-tree search [Sattler et al., ICCV'11]	VPS [Sattler et al., ICCV'11]	Active Search [Sattler et al., ECCV'12]	+ Visibility Filtering [Sattler et al., ECCV'12]
effectiveness	✓	✗	✓✓	✓✓
efficiency	✗✗	✓	✗	✓

Probabilistic 2D-to-3D Search

- 10x speed-up ... if we identify matching features **before** matching

Probabilistic 2D-to-3D Search

- 10x speed-up ... if we identify matching features **before** matching
- **Probabilistic approach**:

Probabilistic 2D-to-3D Search

- 10x speed-up ... if we identify matching features **before** matching
- **Probabilistic approach**:
  - p_i: Probability of finding correct match for i^{th} feature

Probabilistic 2D-to-3D Search

- 10x speed-up ... if we identify matching features **before** matching
- **Probabilistic approach**:
  - p_i: Probability of finding correct match for i^{th} feature
  - c_i: Search cost for finding match for i^{th} feature

Probabilistic 2D-to-3D Search

- 10x speed-up ... if we identify matching features **before** matching
- **Probabilistic approach**:
  - p_i: Probability of finding correct match for i^{th} feature
  - c_i: Search cost for finding match for i^{th} feature
  - Select subset of features by solving

$$\min \sum_i X_i c_i \quad \text{s.t.} \quad \sum_i X_i p_i \geq N_t \text{ with } X_i \in \{0, 1\}$$

Probabilistic 2D-to-3D Search

- 10x speed-up ... if we identify matching features **before** matching
- **Probabilistic approach**:
  - p_i: Probability of finding correct match for i^{th} feature
  - c_i: Search cost for finding match for i^{th} feature
  - Select subset of features by solving

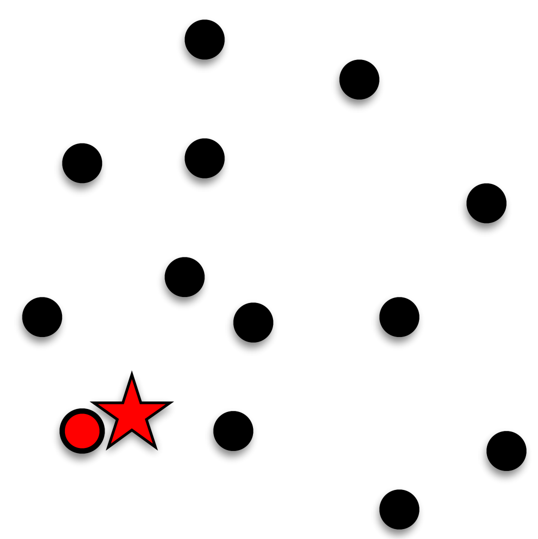
$$\min \underbrace{\sum_i X_i c_i}_{\text{search costs}} \quad \text{s.t.} \quad \sum_i X_i p_i \geq N_t \text{ with } X_i \in \{0, 1\}$$

Probabilistic 2D-to-3D Search

- 10x speed-up ... if we identify matching features **before** matching
- **Probabilistic approach**:
  - p_i: Probability of finding correct match for i^{th} feature
  - c_i: Search cost for finding match for i^{th} feature
  - Select subset of features by solving

$$\underbrace{\min \sum_i X_i c_i}_{\textbf{search costs}} \quad \text{s.t.} \quad \underbrace{\sum_i X_i p_i}_{\textbf{expected \# matches}} \geq N_t \text{ with } X_i \in \{0, 1\}$$

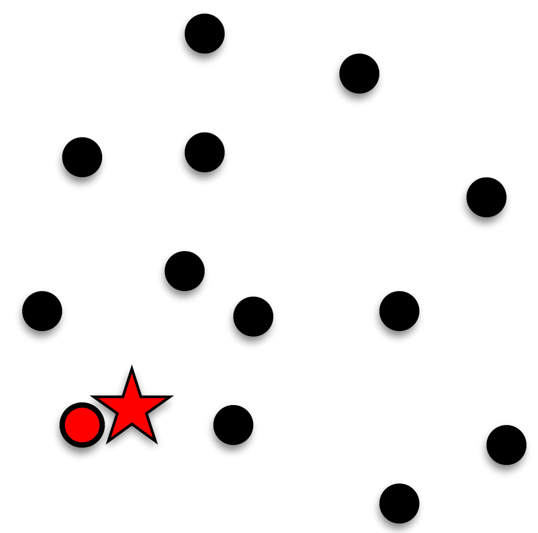
Probabilistic 2D-to-3D Search



Efficient computation of p_i, c_i

Computer Vision and Geometry Group ETH zürich

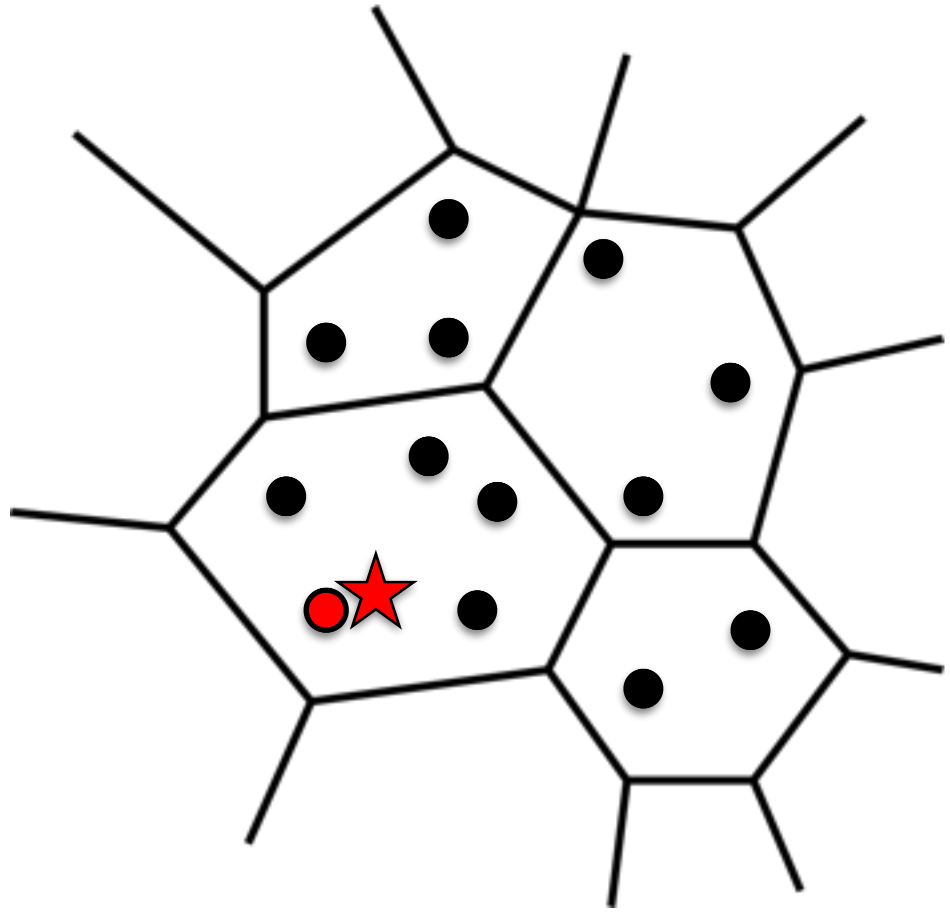
Probabilistic 2D-to-3D Search



Efficient computation of p_i, c_i

- Precompute probabilities for **regions** in descriptor space

Probabilistic 2D-to-3D Search



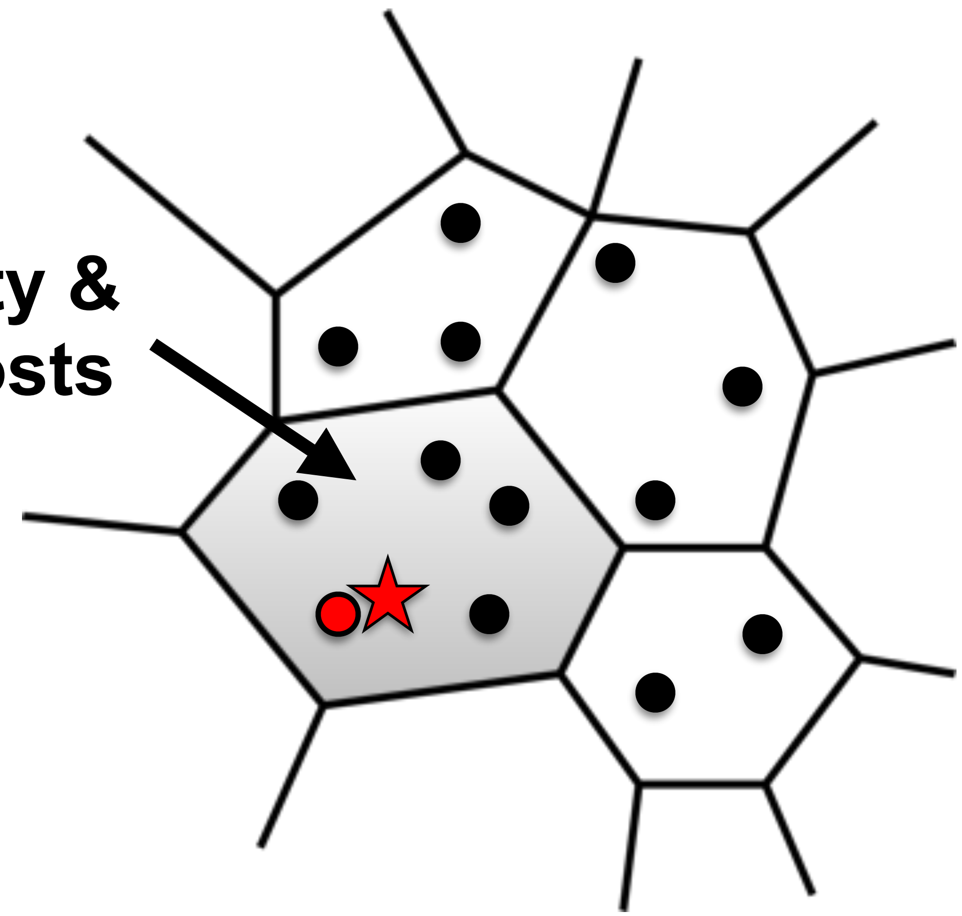
Fixed & Precomputed Visual Vocabulary

Efficient computation of p_i, c_i

- Precompute probabilities for **regions** in descriptor space

Probabilistic 2D-to-3D Search

Constant probability & constant search costs



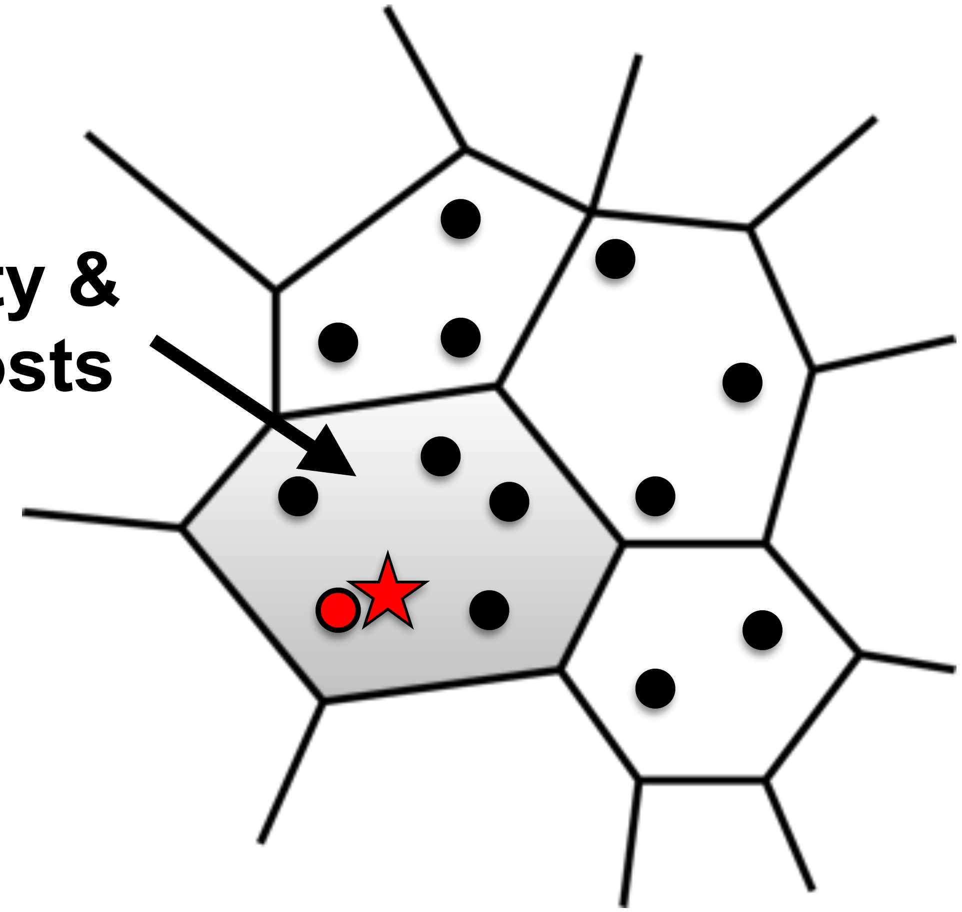
Fixed & Precomputed Visual Vocabulary

Efficient computation of p_i, c_i

- Precompute probabilities for **regions** in descriptor space

Probabilistic 2D-to-3D Search

Constant probability & constant search costs



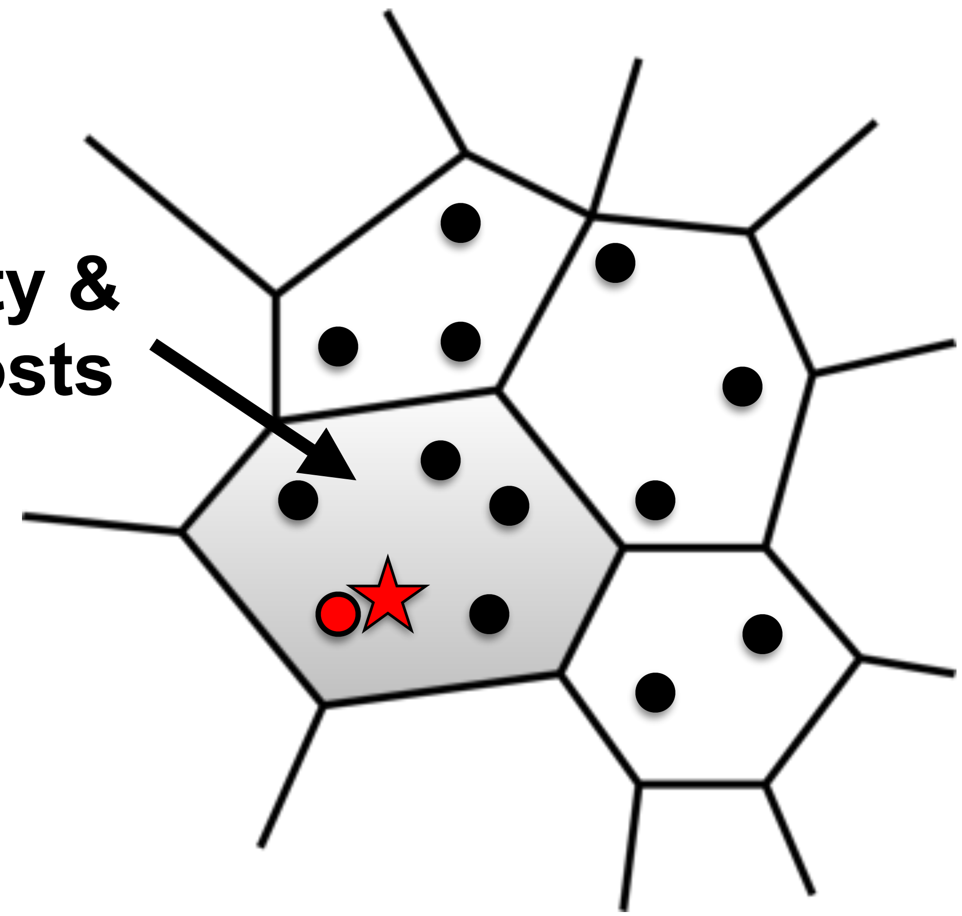
Fixed & Precomputed Visual Vocabulary

Efficient computation of p_i, c_i

- Precompute probabilities for **regions** in descriptor space
- Limit nearest neighbor search to same cell (**Quantized Search**)

Probabilistic 2D-to-3D Search

Constant probability & constant search costs



Fixed & Precomputed Visual Vocabulary

Efficient computation of p_i, c_i

- Precompute probabilities for **regions** in descriptor space
- Limit nearest neighbor search to same cell (**Quantized Search**)
- Computation in constant time for fixed-size vocabulary

A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**

A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:

A Greedy Approach

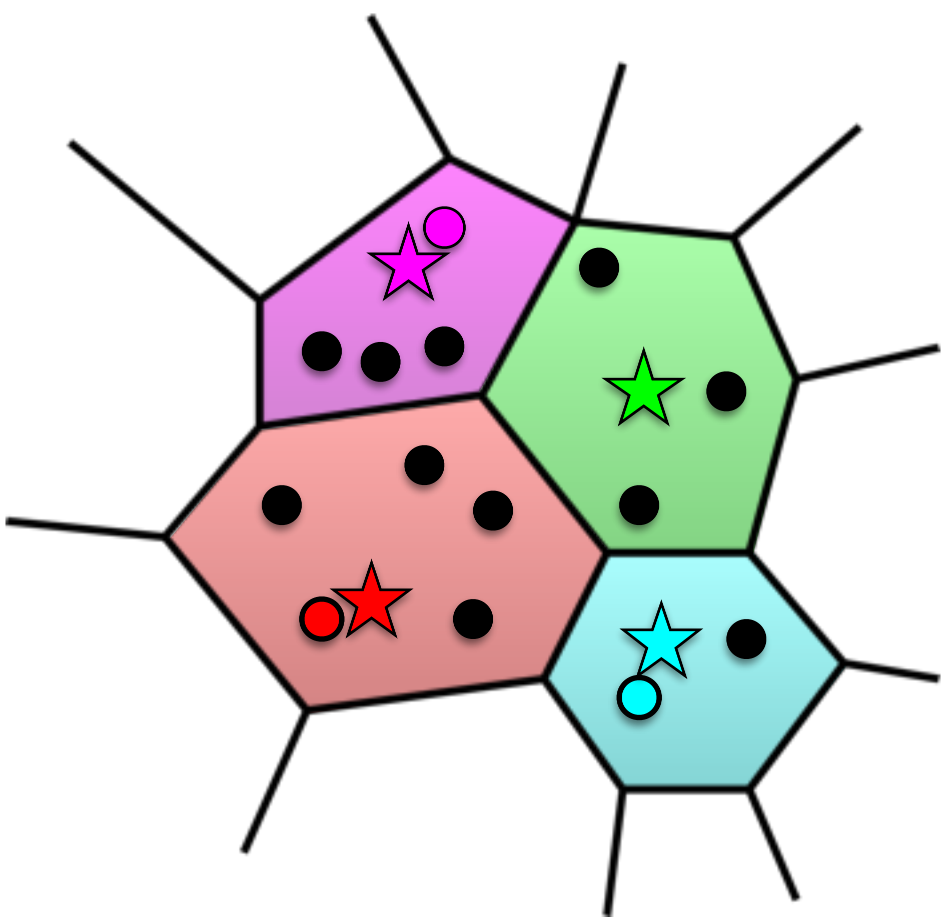
- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with p_i>0 in ascending order of search costs

A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with p_i>0 in ascending order of search costs
  - Select first m features such that $\sum_{i=1}^{m} p_i \geq N_t$

A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with p_i>0 in ascending order of search costs
  - Select first m features such that $\sum_{i=1}^{m} p_i \geq N_t$

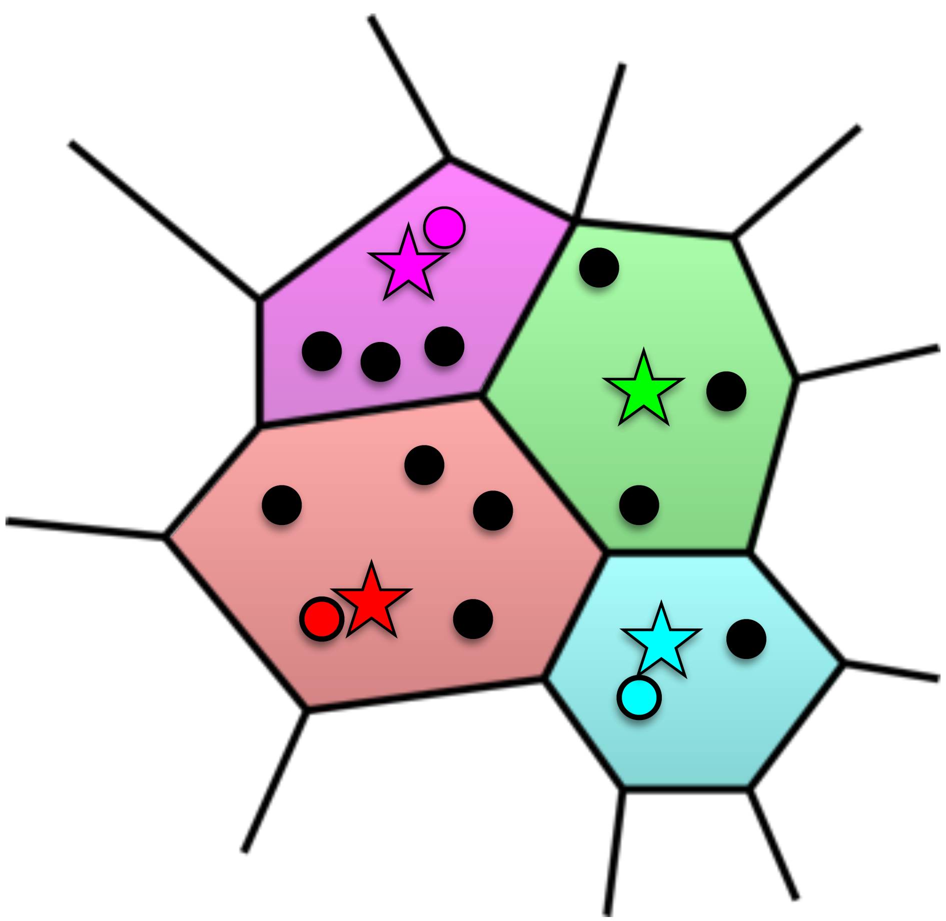


Resulting order:

Computer Vision and Geometry Group ETH zürich

A Greedy Approach

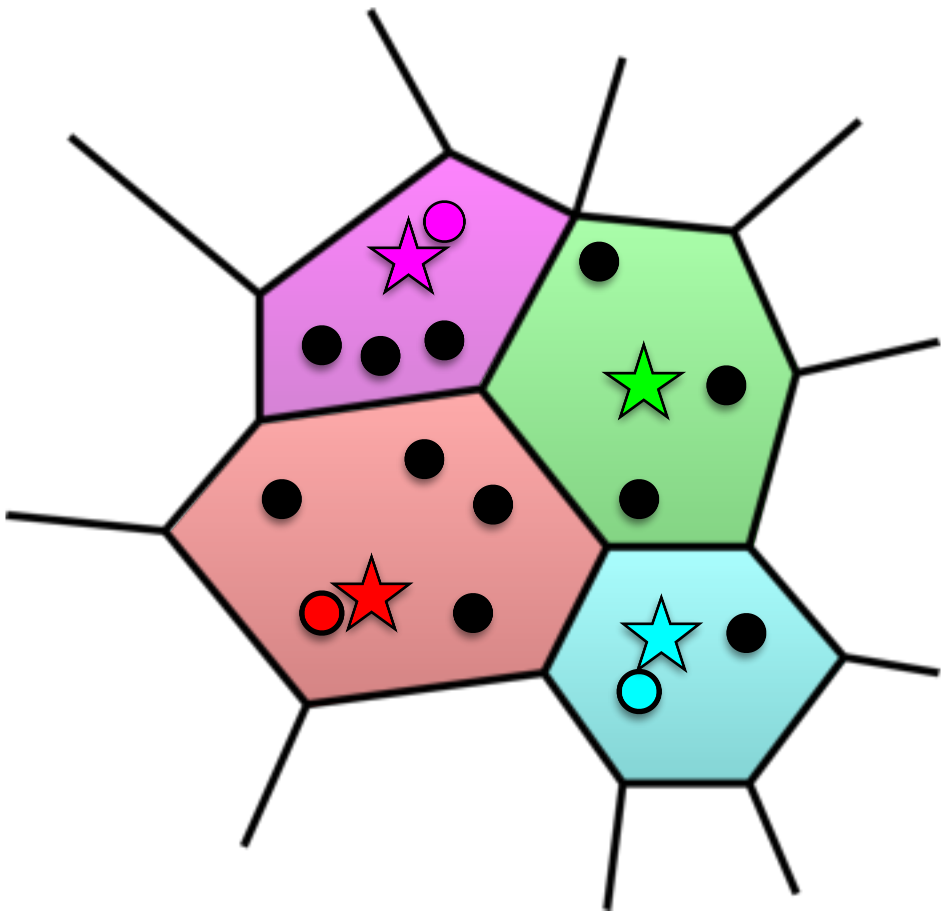
- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with p_i>0 in ascending order of search costs
  - Select first m features such that $\sum_{i=1}^{m} p_i \geq N_t$



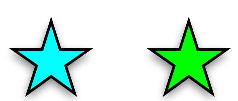
Resulting order:

A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with $p_i>0$ in ascending order of search costs
  - Select first m features such that $\sum_{i=1}^{m} p_i \geq N_t$

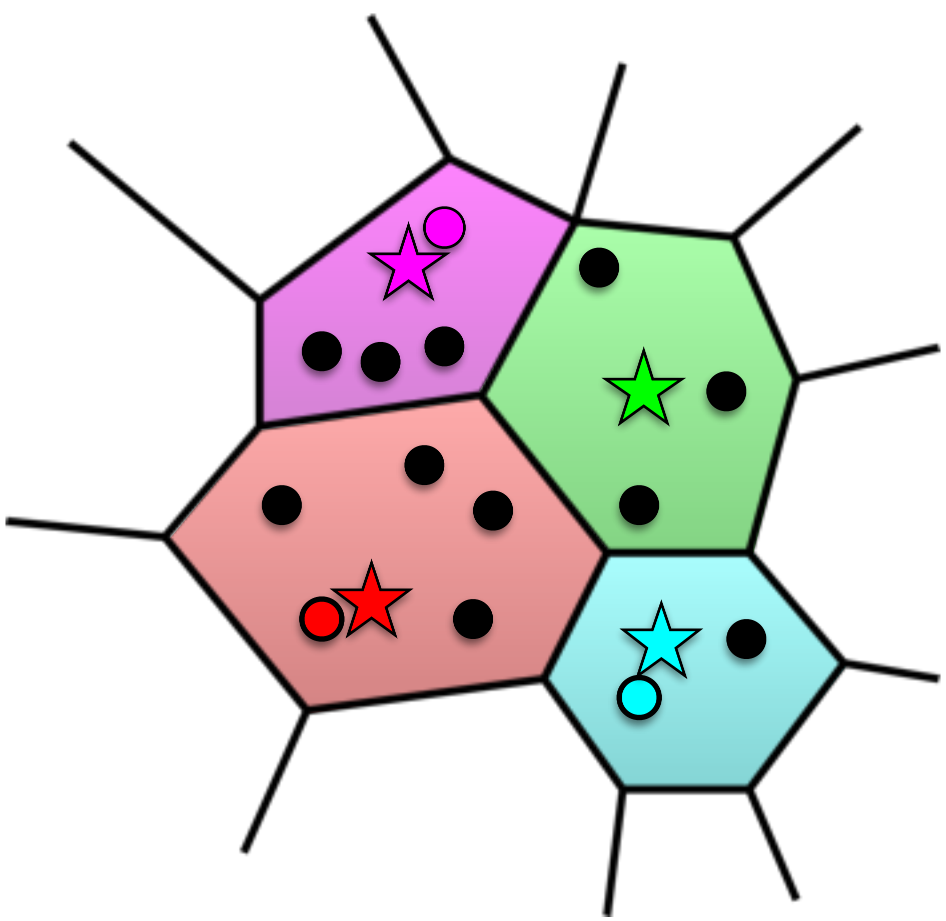


Resulting order:

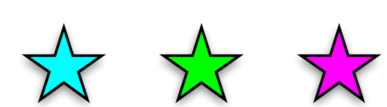


A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with $p_i > 0$ in ascending order of search costs
  - Select first m features such that $\sum_{i=1}^{m} p_i \geq N_t$

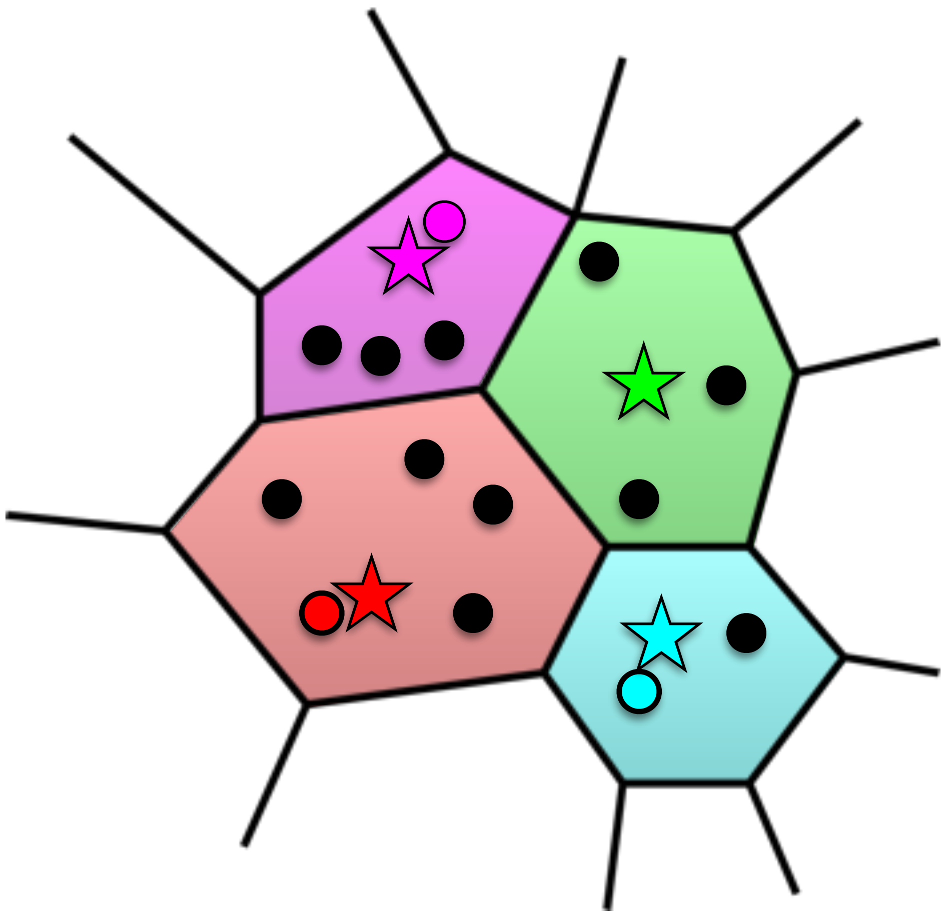


Resulting order:

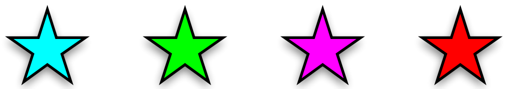


A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with $p_i > 0$ in ascending order of search costs
  - Select first m features such that $\sum_{i=1}^{m} p_i \geq N_t$

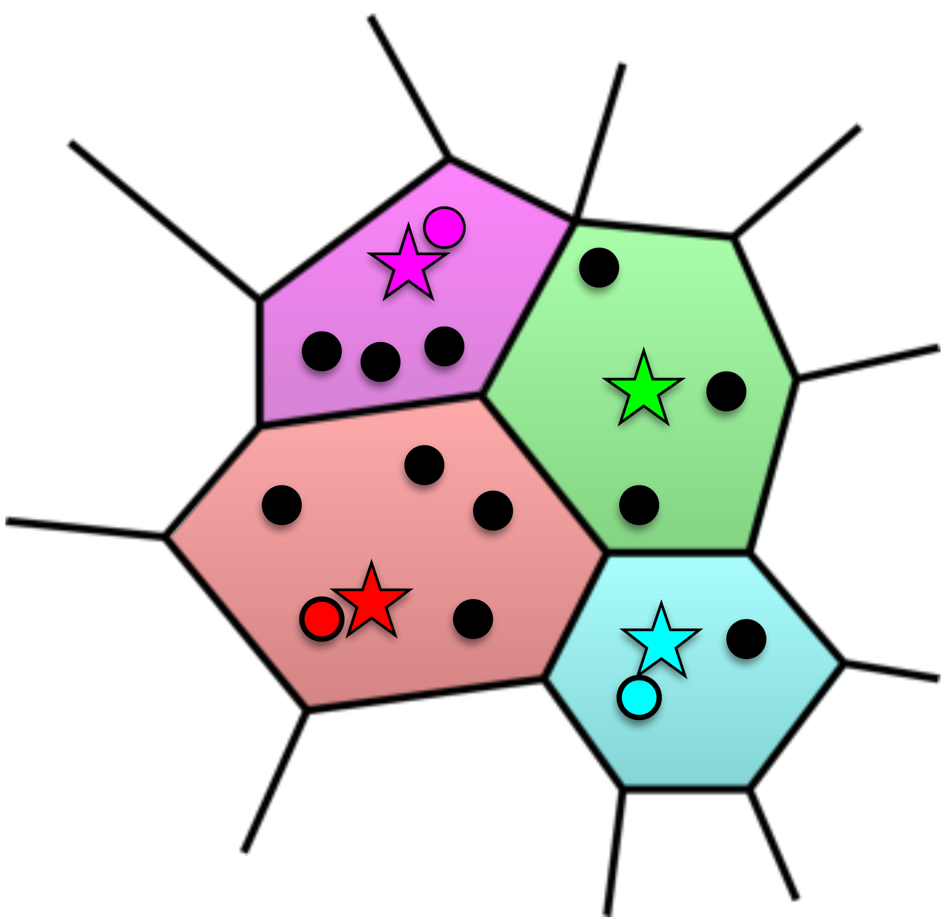


Resulting order:

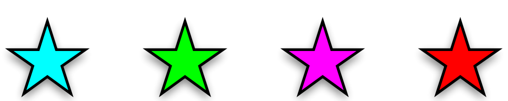


A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with $p_i > 0$ in ascending order of search costs
  - Select first m features such that $\sum_{i=1}^{m} p_i \geq N_t$



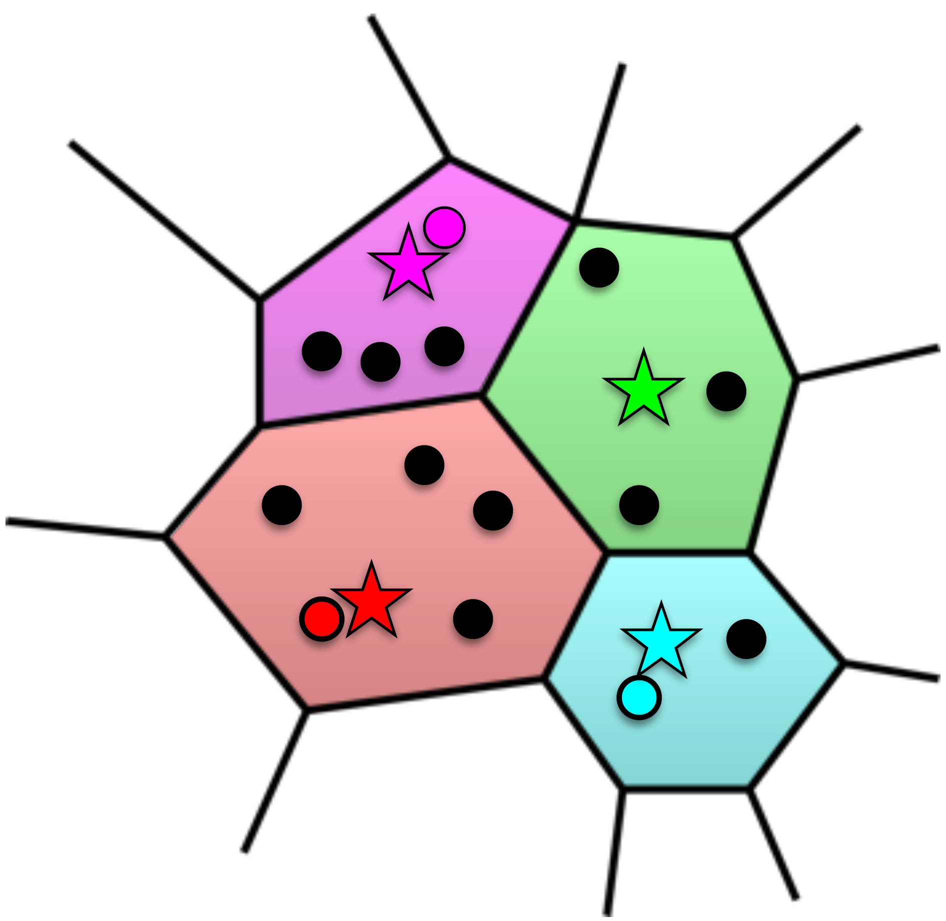
Resulting order:



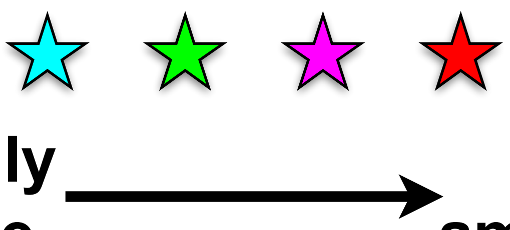
Globally unique

A Greedy Approach

- Solving $\min \sum_i X_i c_i \text{ s.t. } \sum_i X_i p_i \geq N_t$ is **NP-complete**
- Simple Greedy strategy:
  - Sort features with $p_i > 0$ in ascending order of search costs
  - Select first m features such that $\sum_{i=1}^{m} p_i \geq N_t$

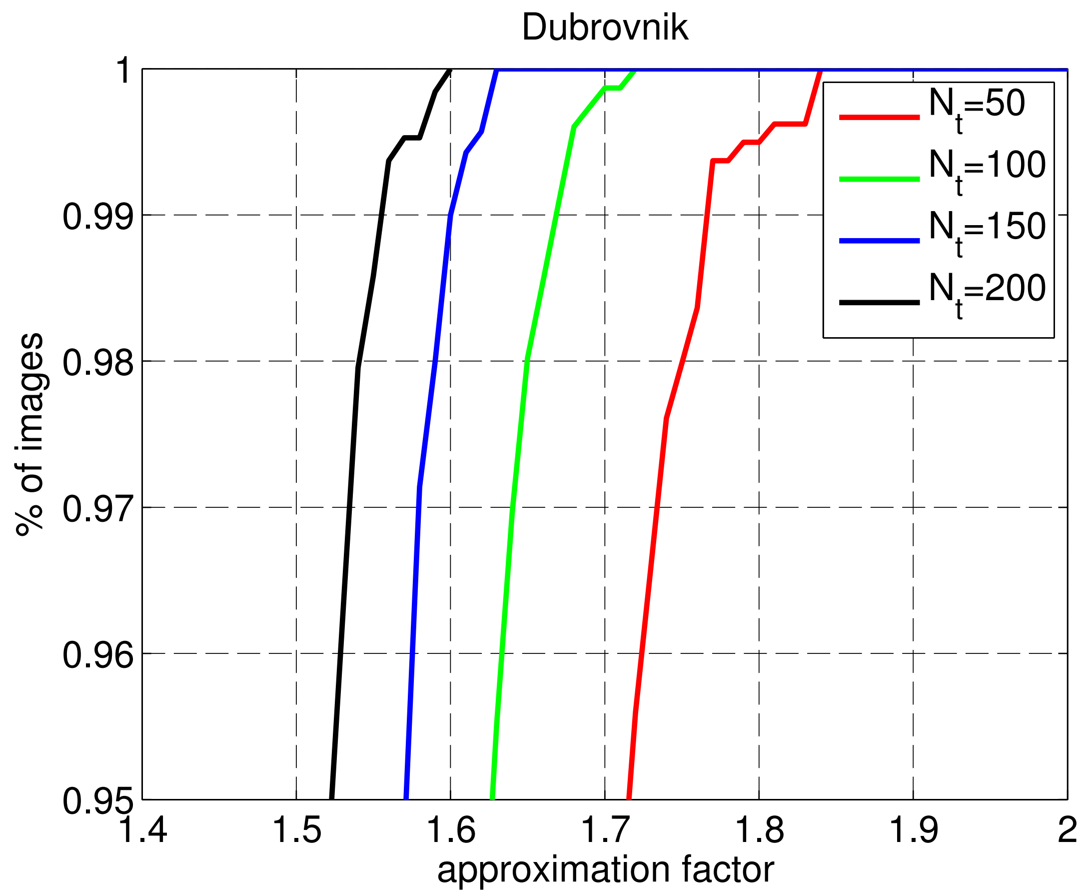


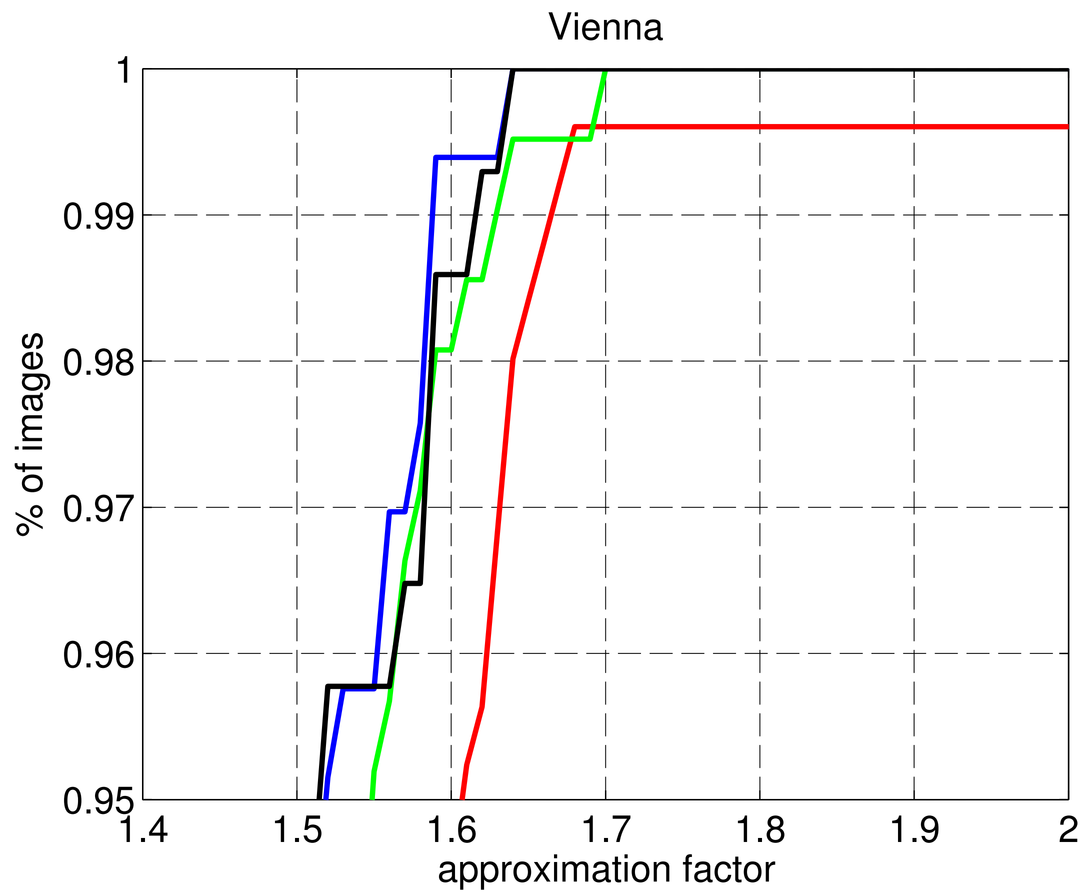
Resulting order:



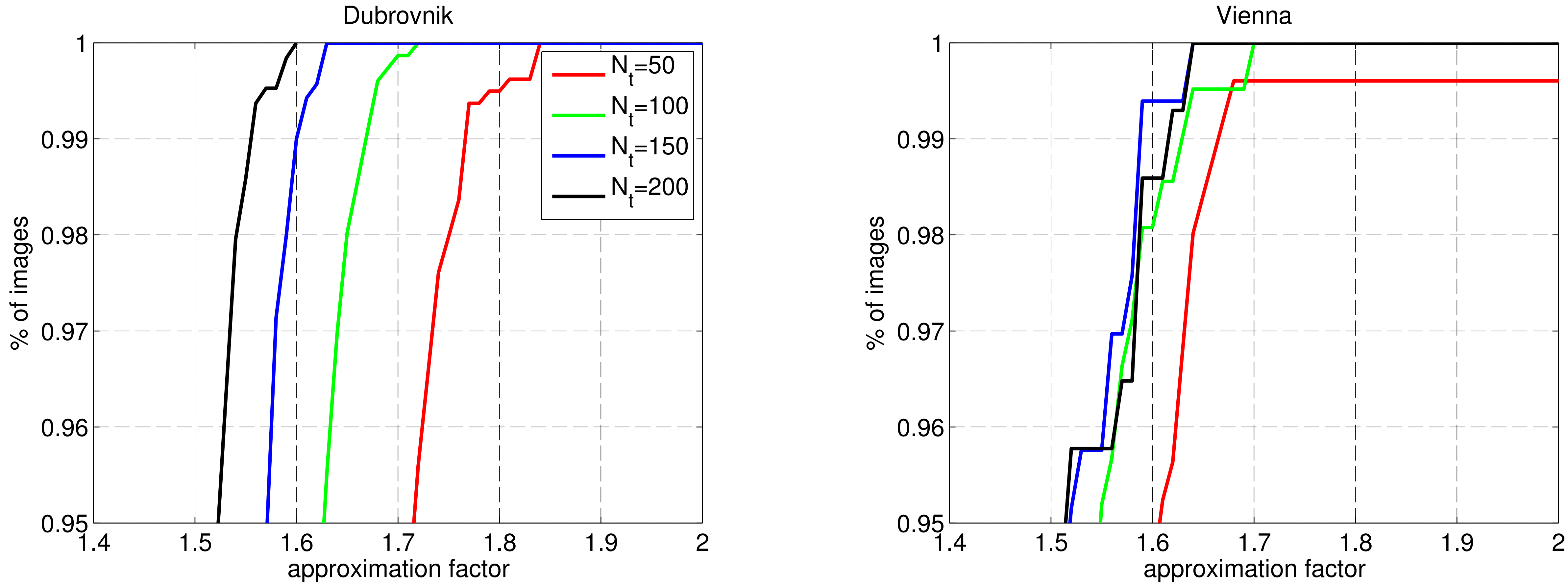
Globally unique → **Highly ambiguous**

Greedy vs. OPT



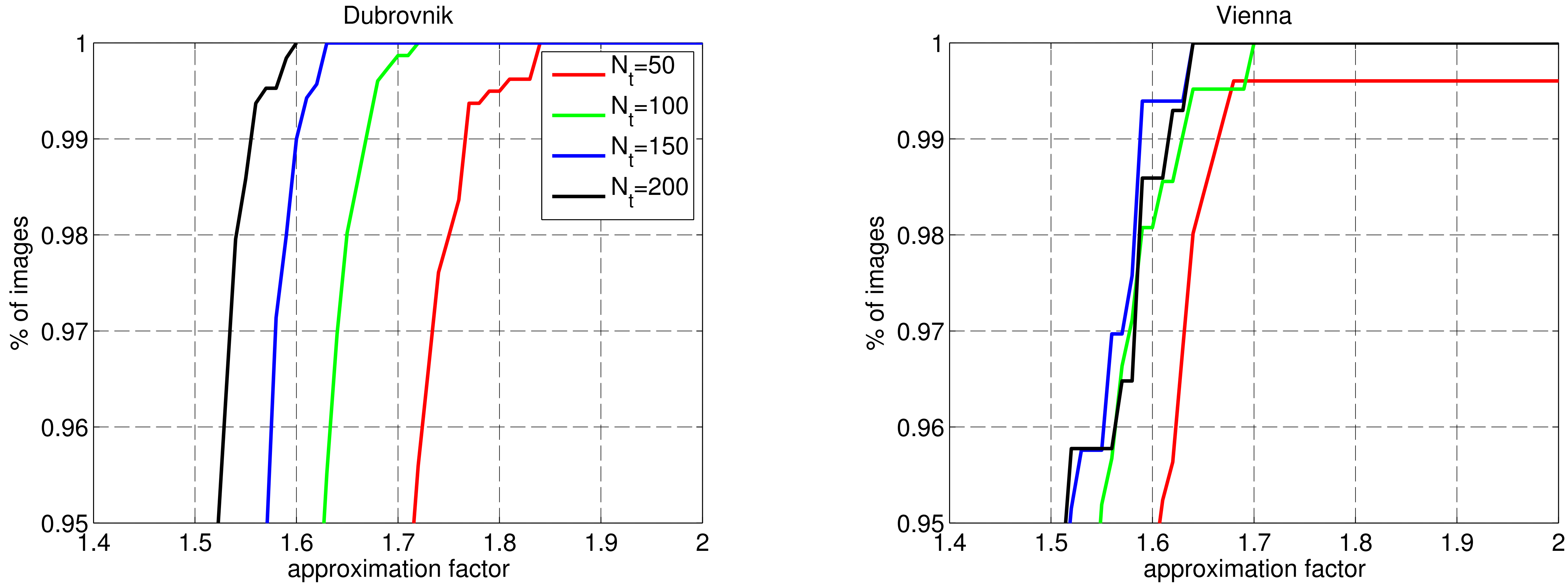


Greedy vs. OPT



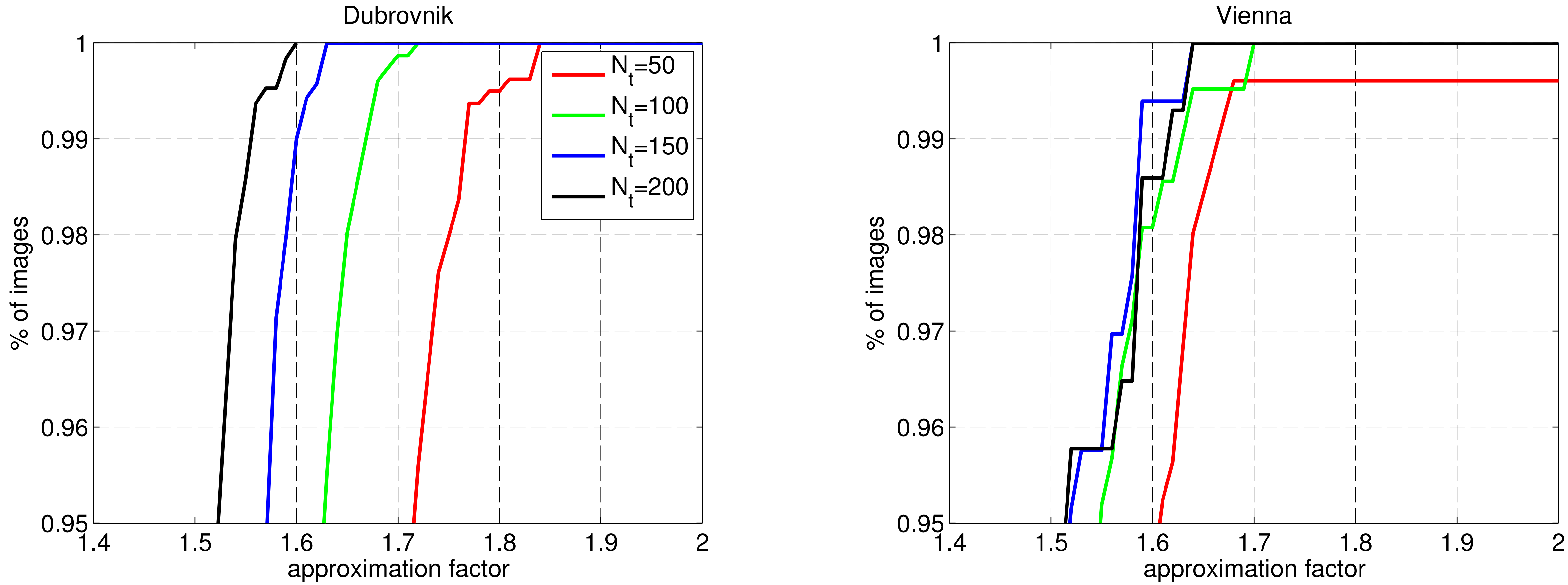
- Greedy performs close to optimal!

Greedy vs. OPT



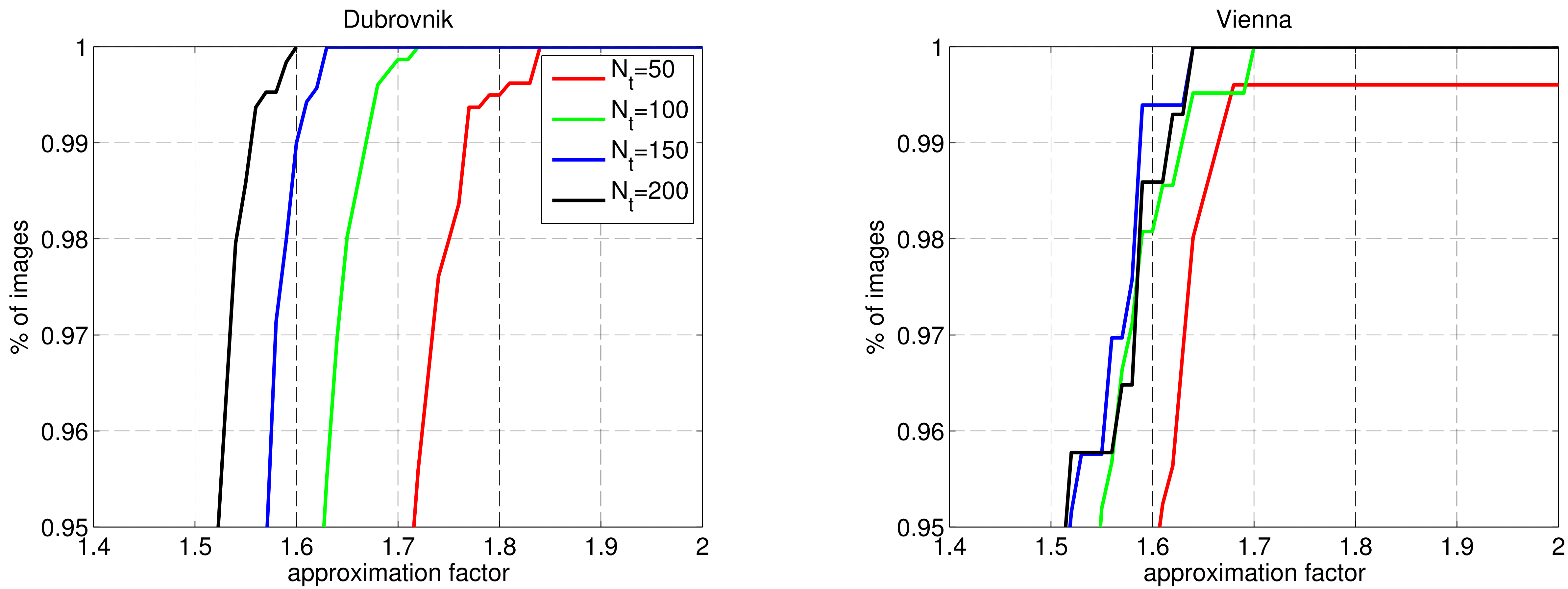
- Greedy performs close to optimal!
- Here: Probabilities learnt from query images

Greedy vs. OPT



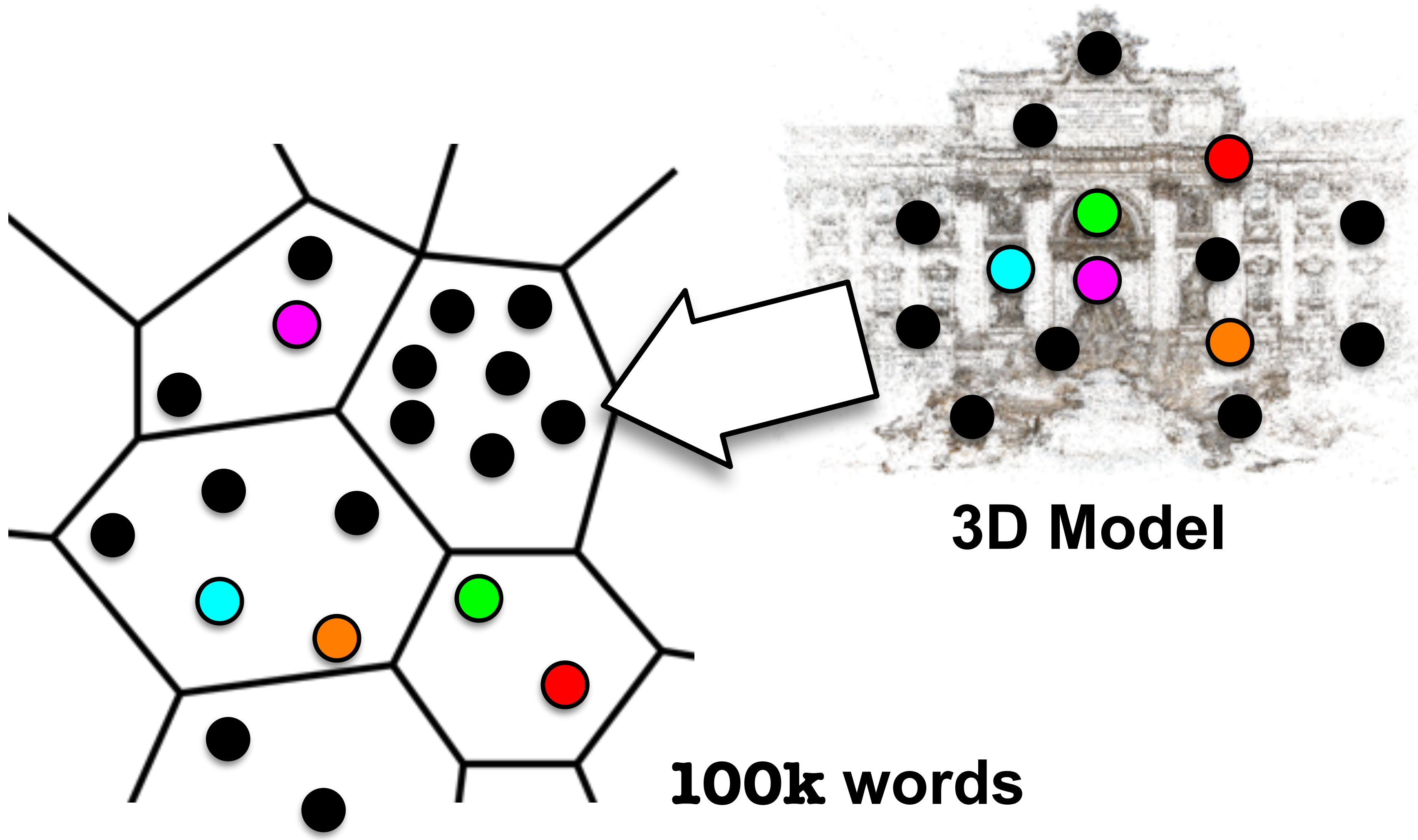
- Greedy performs close to optimal!
- Here: Probabilities learnt from query images
- In practice: Hard to find good training data

Greedy vs. OPT



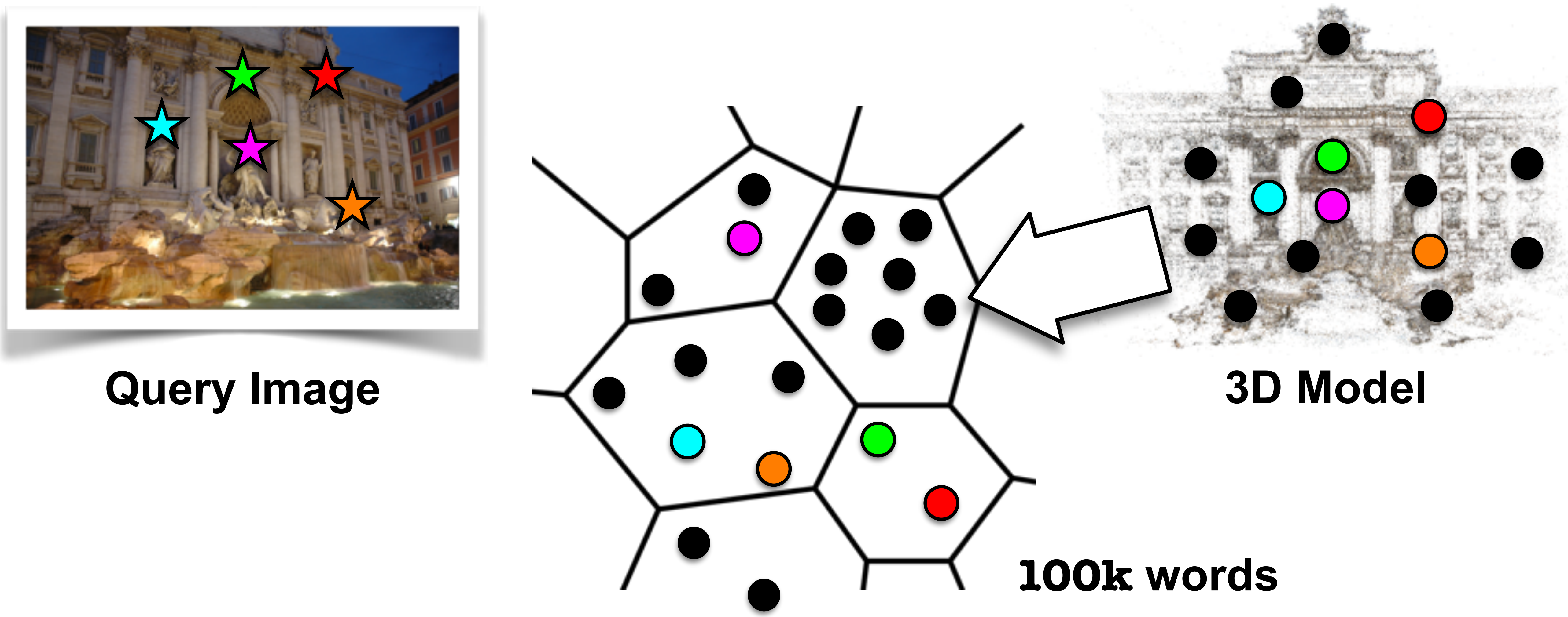
- Greedy performs close to optimal!
- Here: Probabilities learnt from query images
- In practice: Hard to find good training data
  - … but Greedy does not really need probabilities

Vocabulary-Based Prioritized Search (VPS)



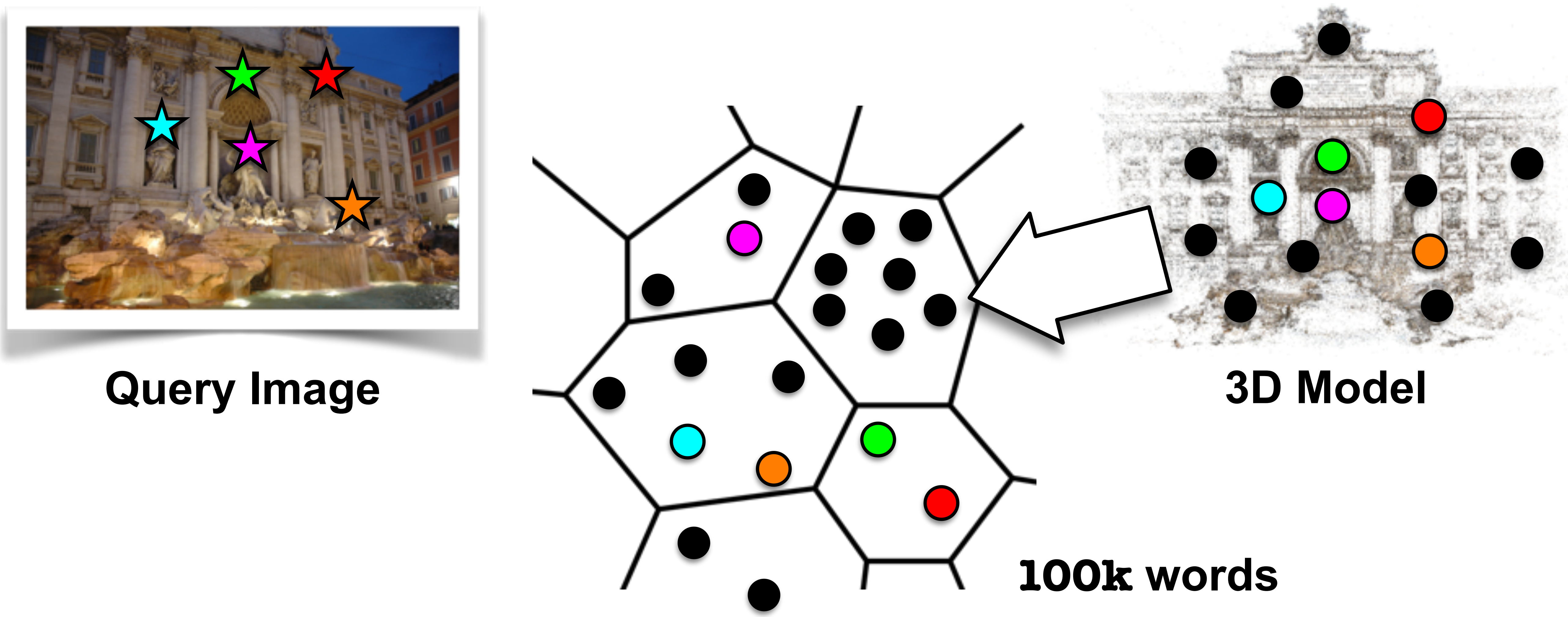
[Sattler et al., ICCV'11] [code]

Vocabulary-Based Prioritized Search (VPS)



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

Vocabulary-Based Prioritized Search (VPS)



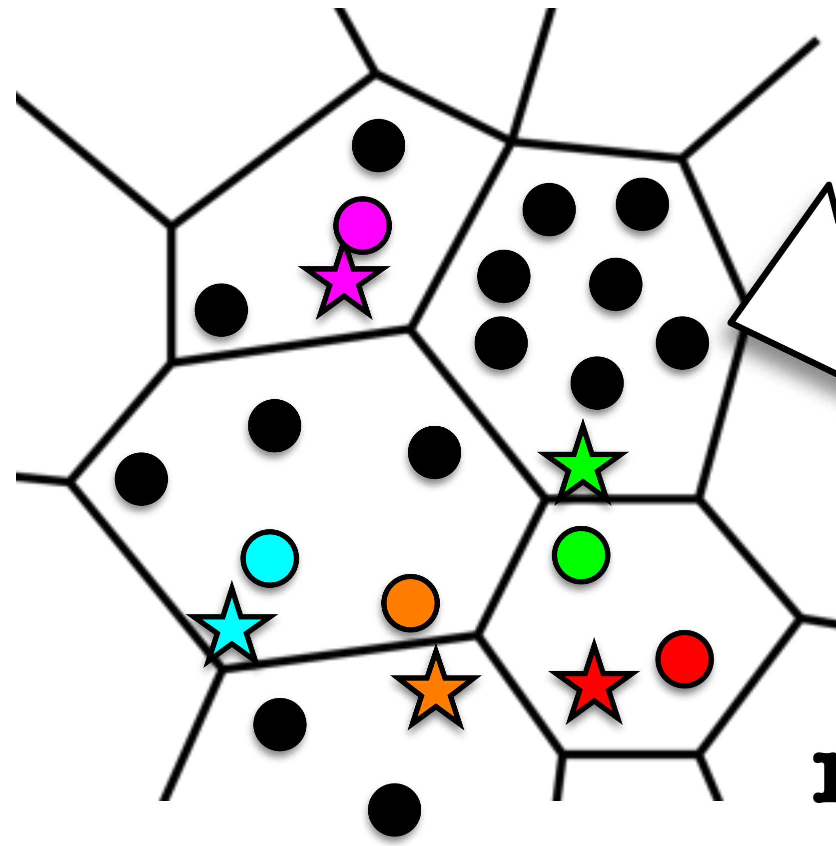
Assign features to words

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

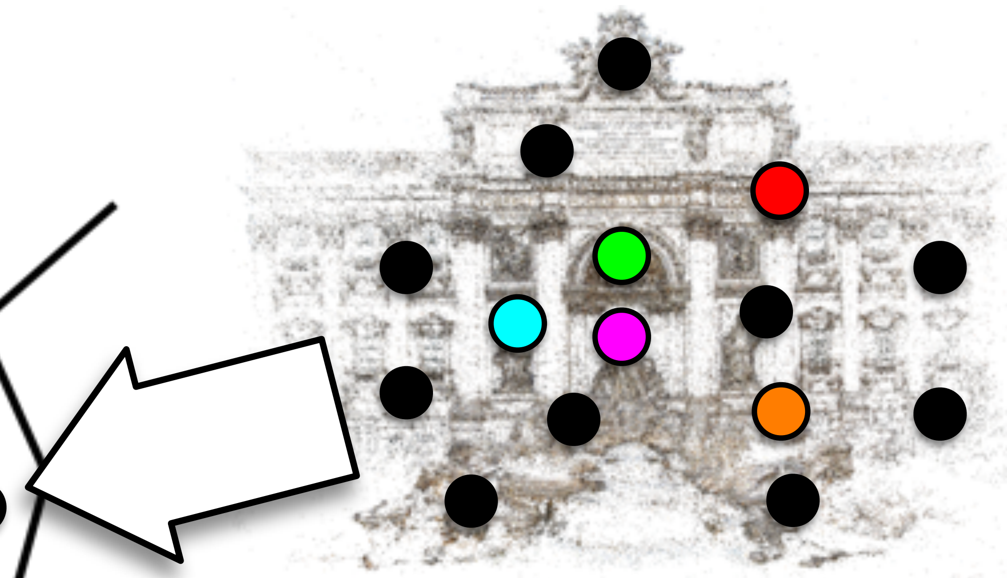
Computer Vision and Geometry Group ETH zürich

Vocabulary-Based Prioritized Search (VPS)

Query Image



100k words

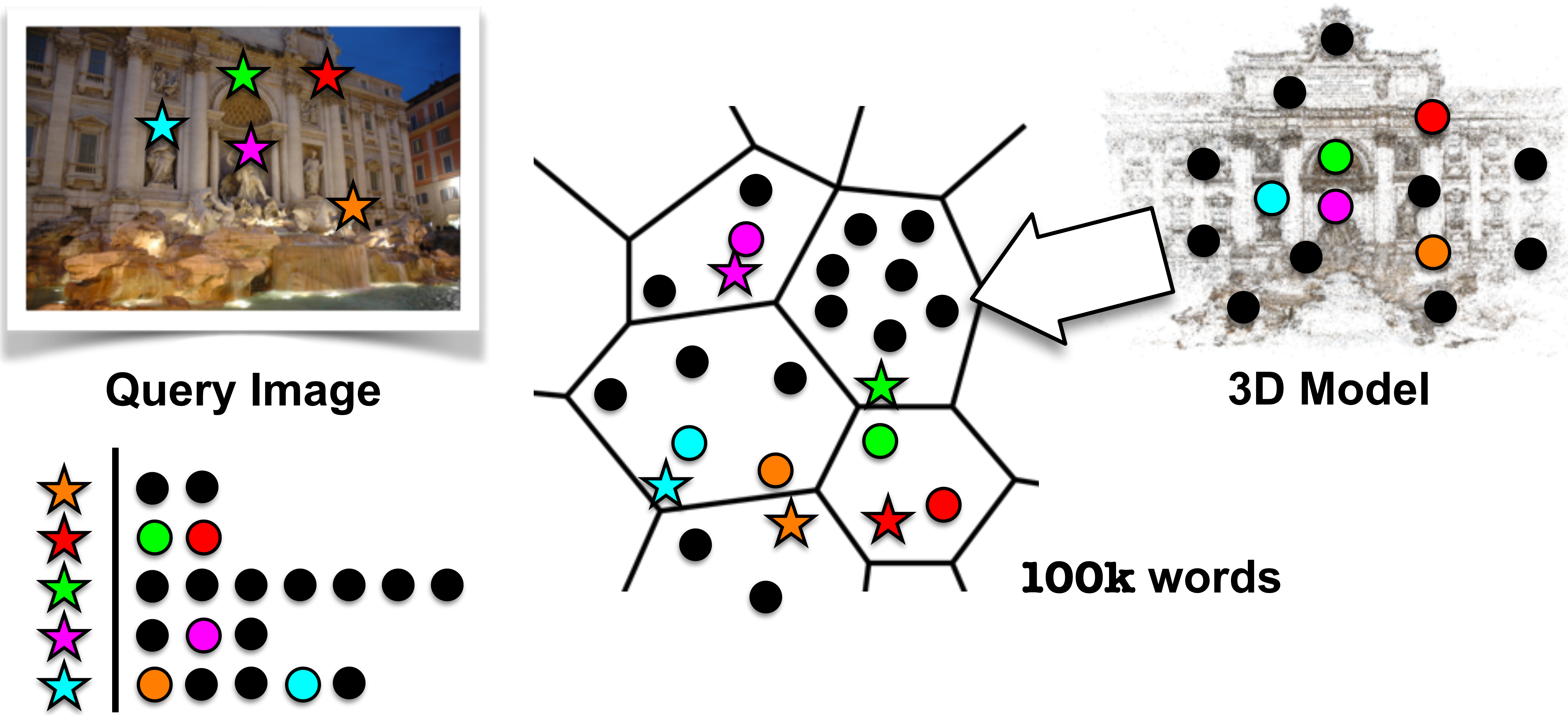


3D Model

Assign features to words

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

Vocabulary-Based Prioritized Search (VPS)

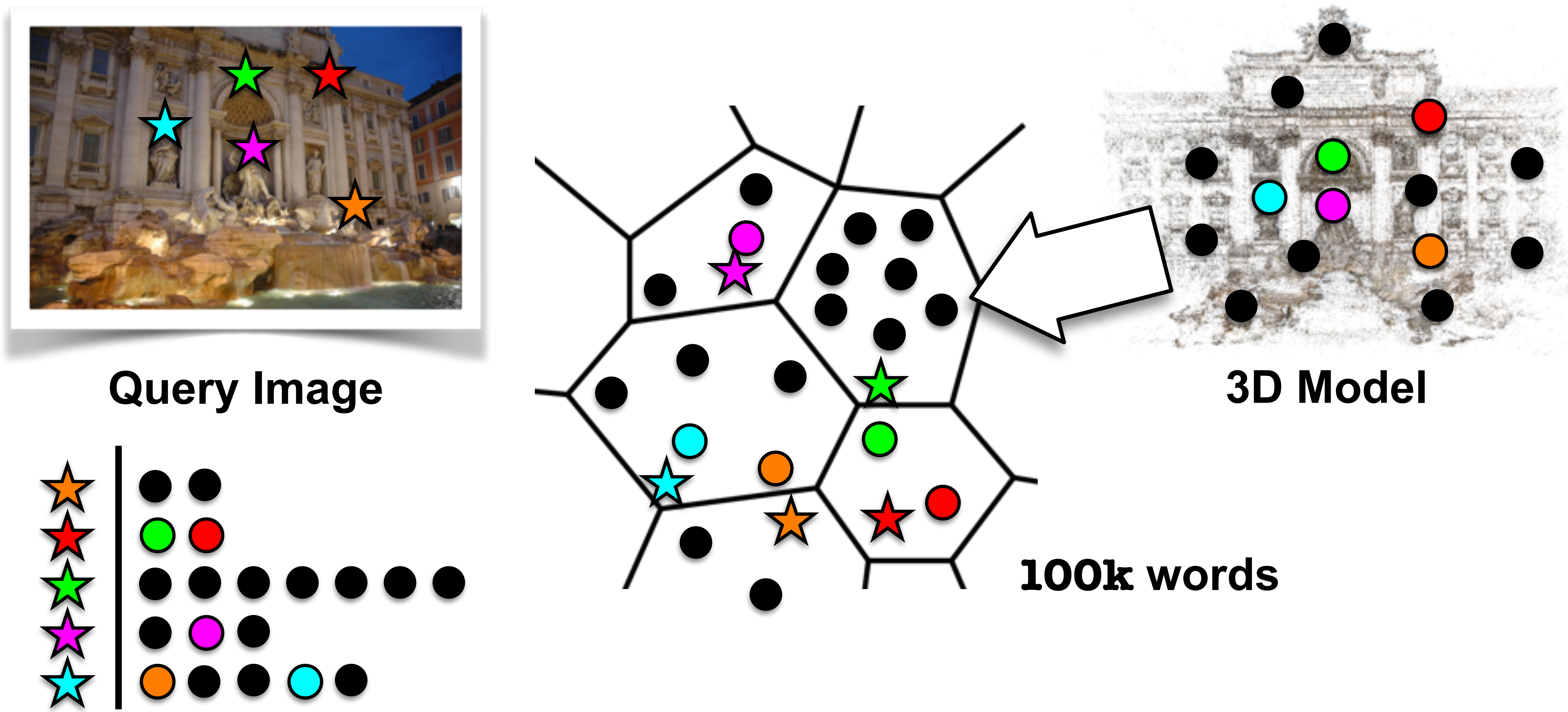


Assign features to words

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

Computer Vision and Geometry Group ETH zürich

Vocabulary-Based Prioritized Search (VPS)



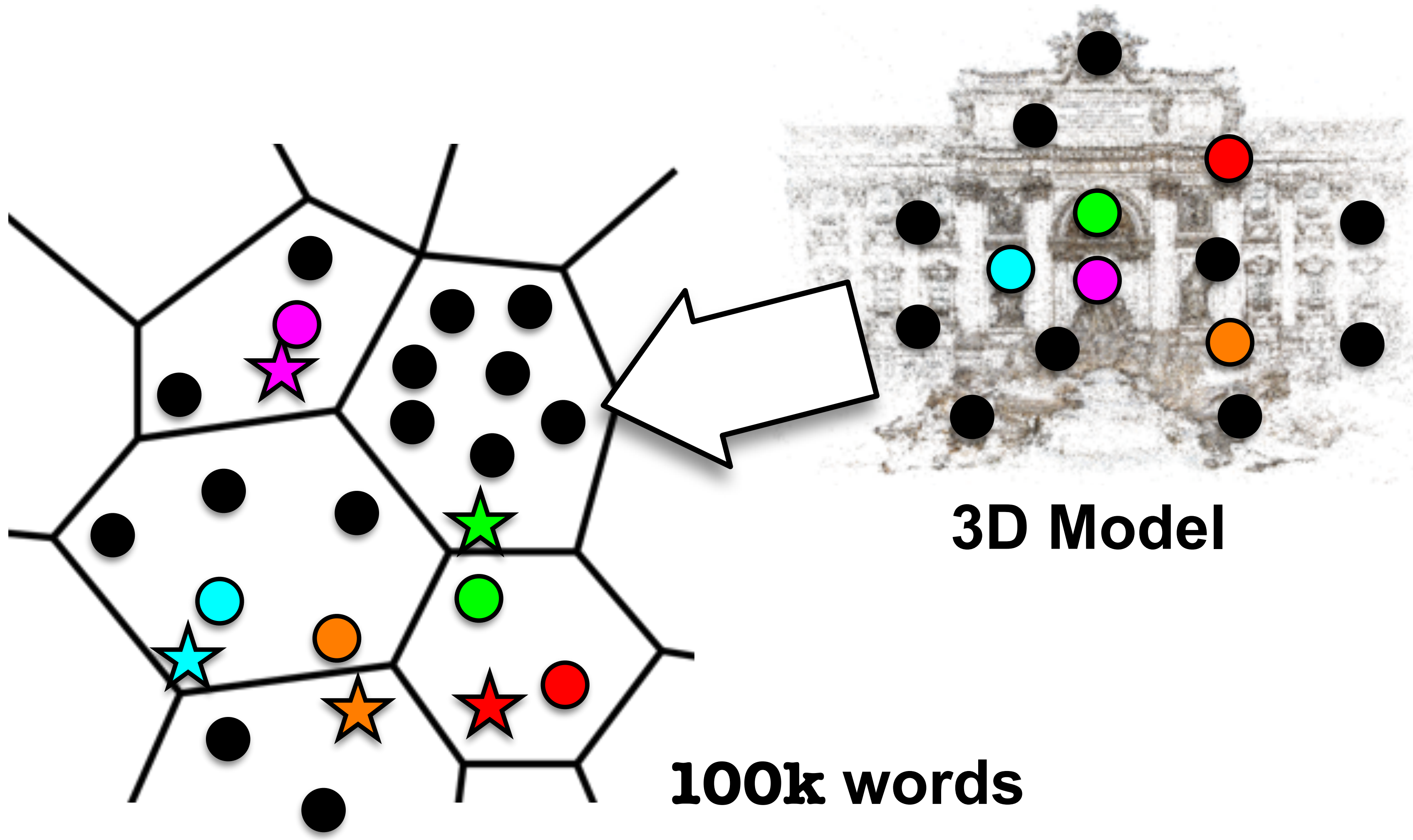
Assign features to words

Sort based on costs

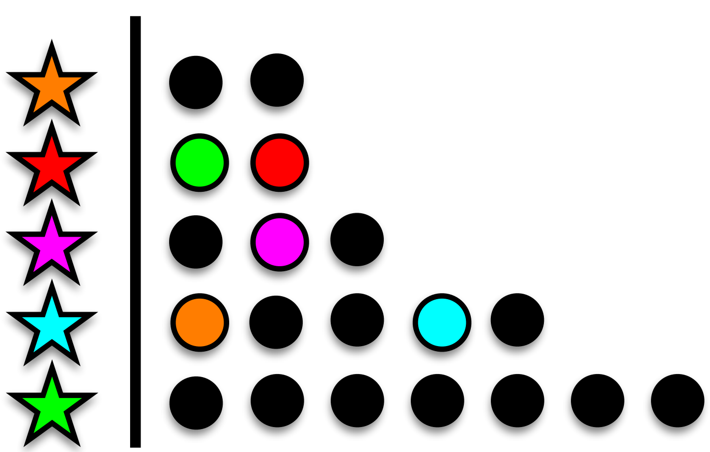
[Sattler et al., ICCV'11] [code]

Vocabulary-Based Prioritized Search (VPS)

Query Image



3D Model



Assign features to words

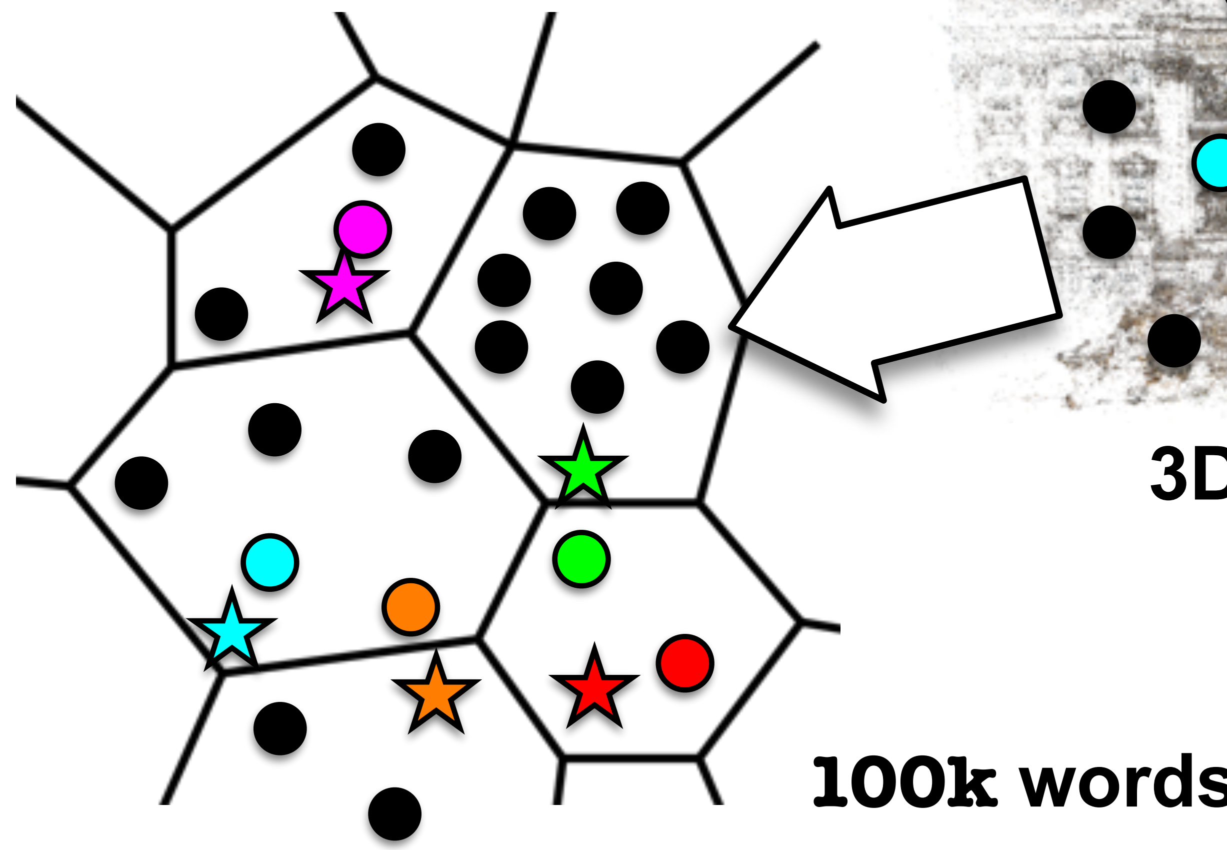
Sort based on costs

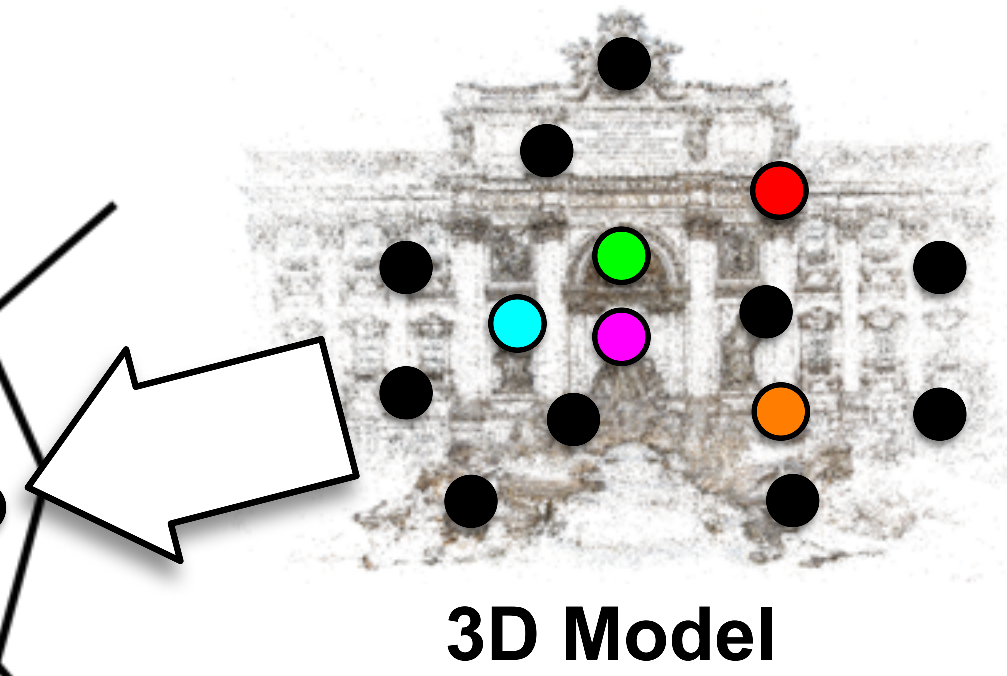
[Sattler et al., ICCV'11] [code]

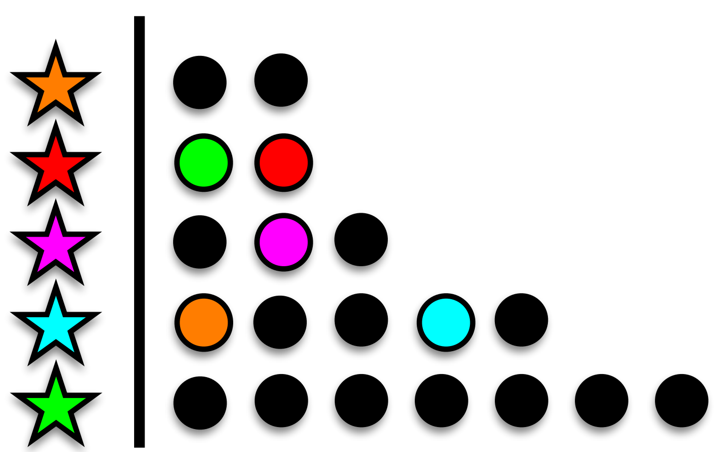
Computer Vision and Geometry Group

ETH zürich

Vocabulary-Based Prioritized Search (VPS)







Assign features to words

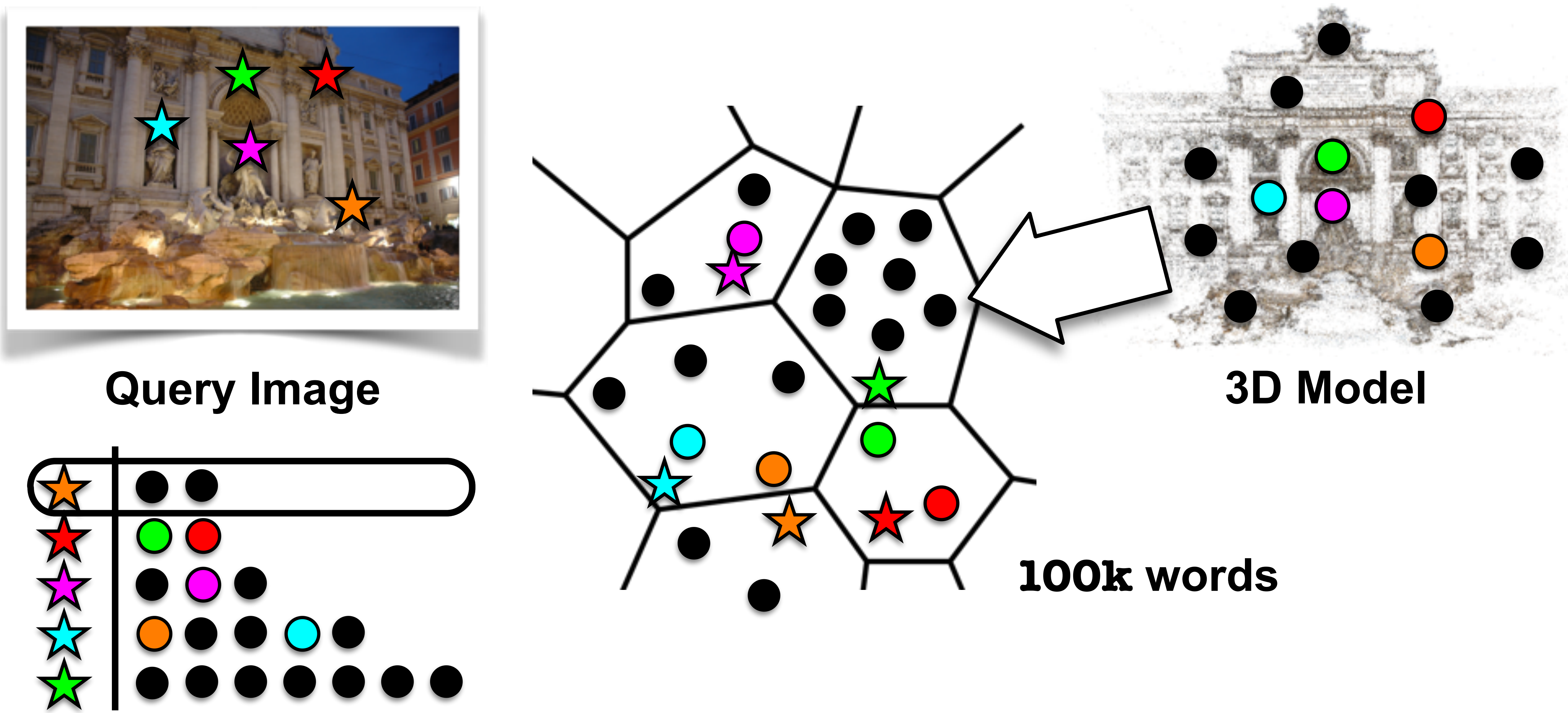
Sort based on costs

Linear search through words

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

Computer Vision and Geometry Group ETH zürich

Vocabulary-Based Prioritized Search (VPS)



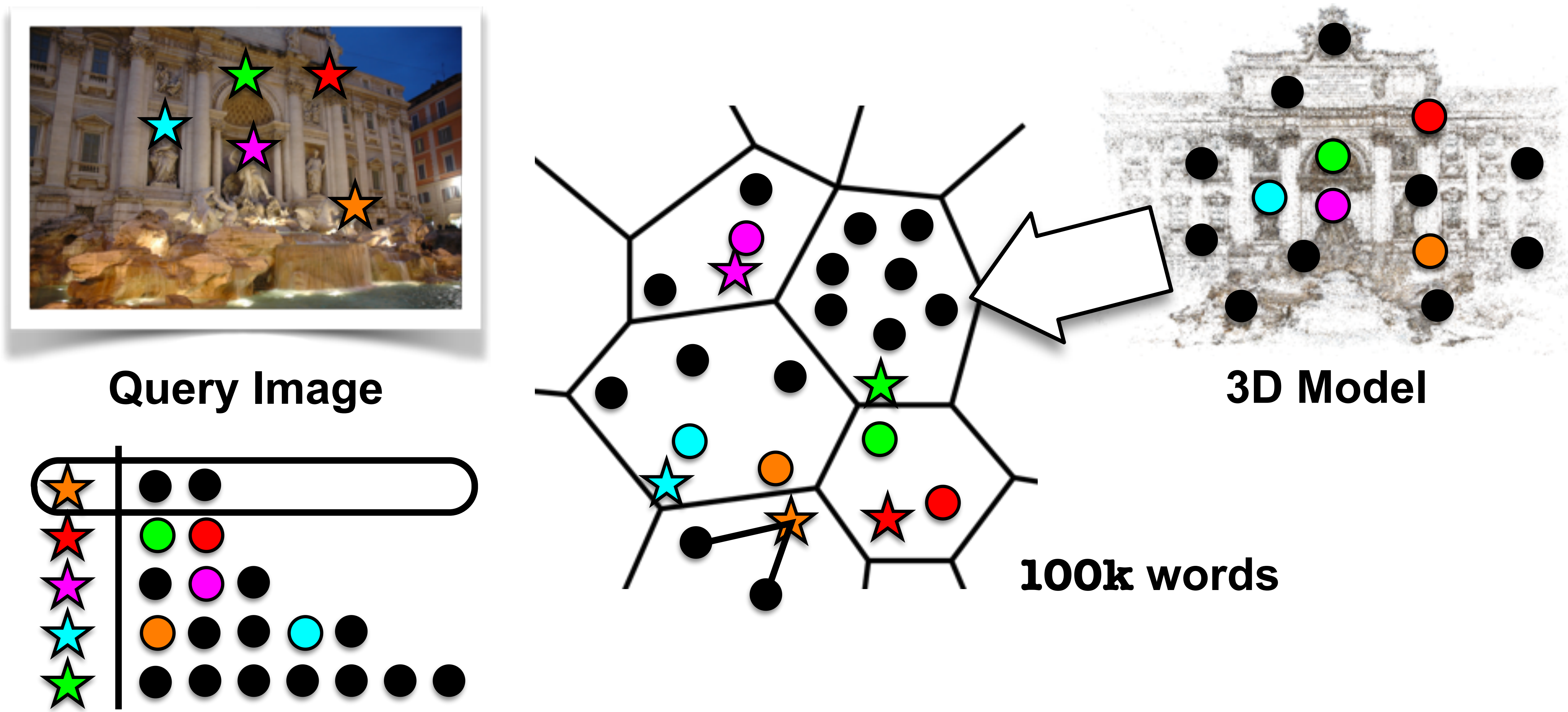
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



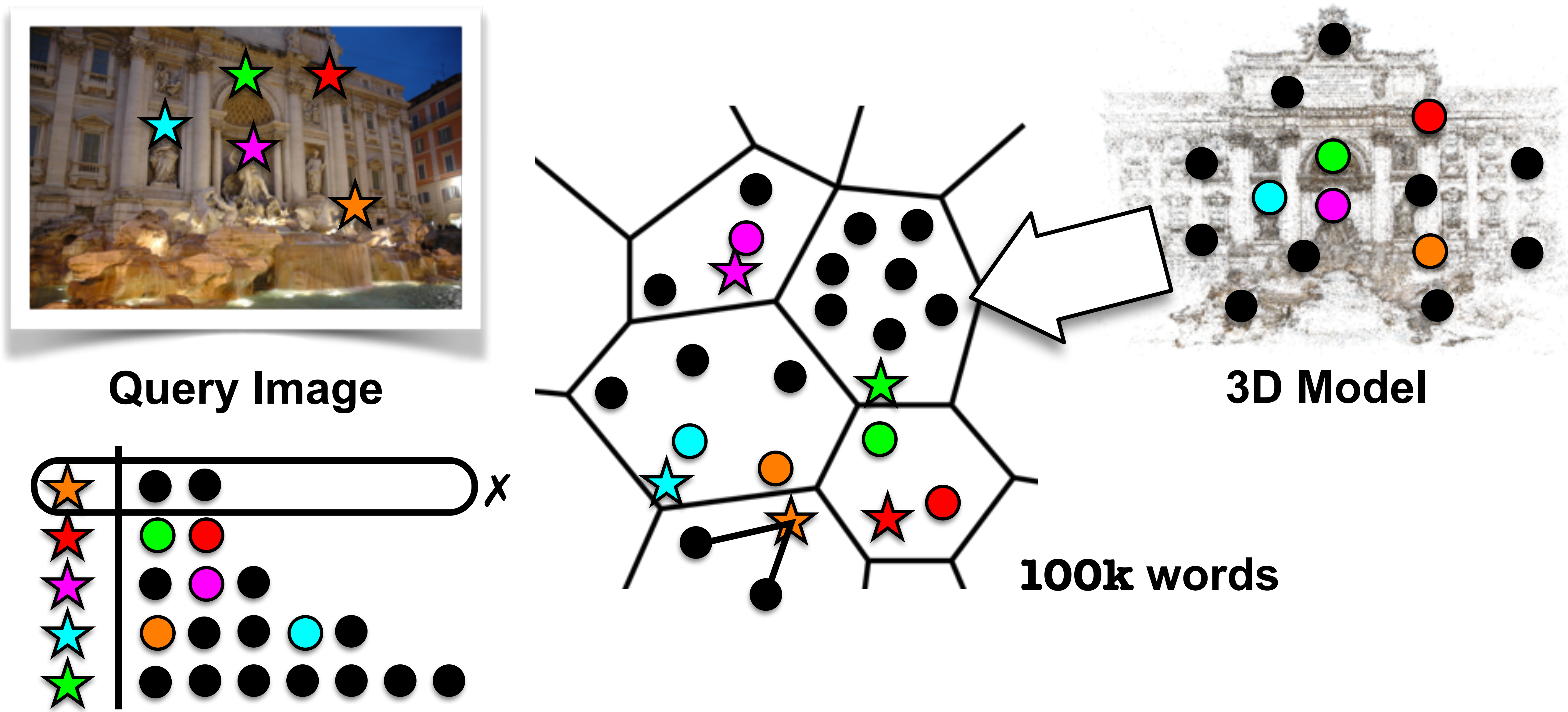
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



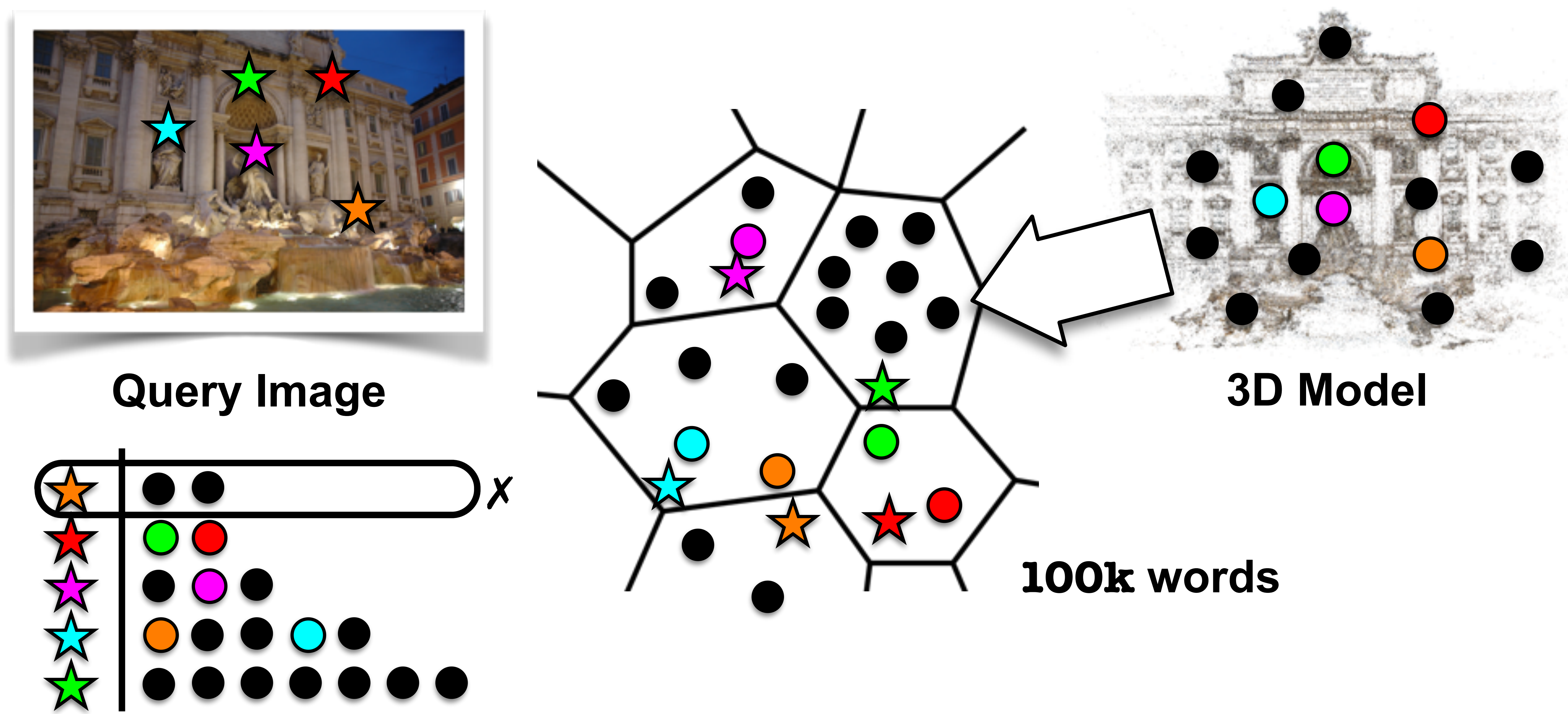
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



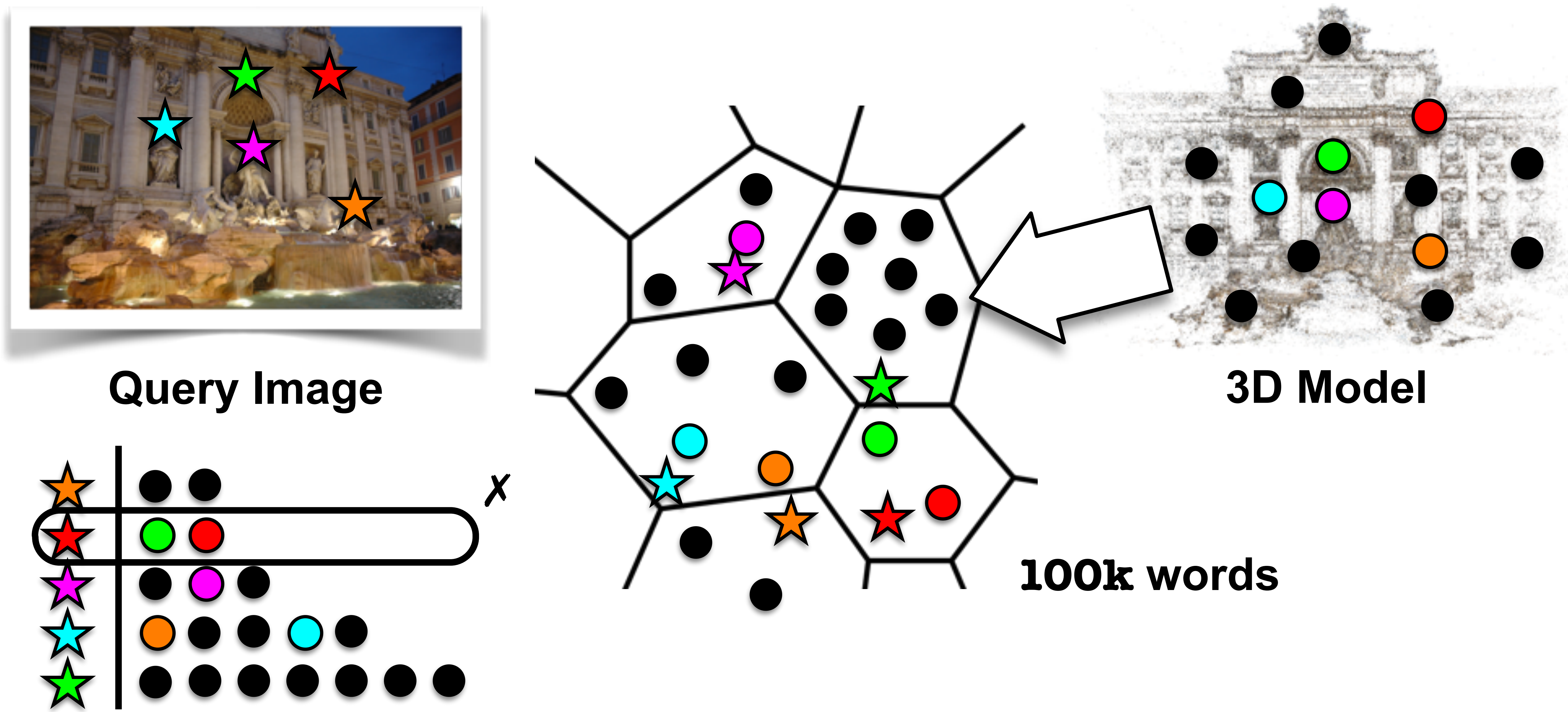
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



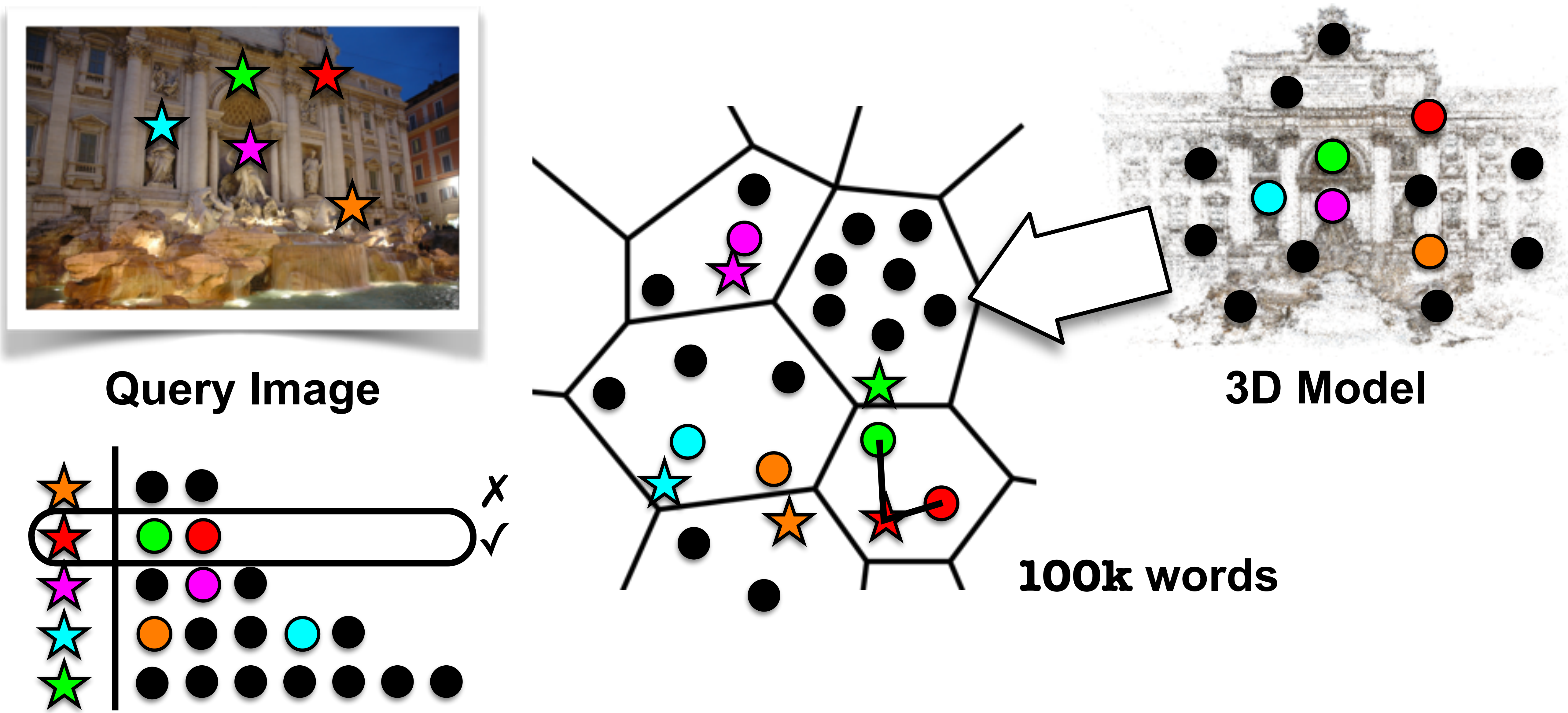
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



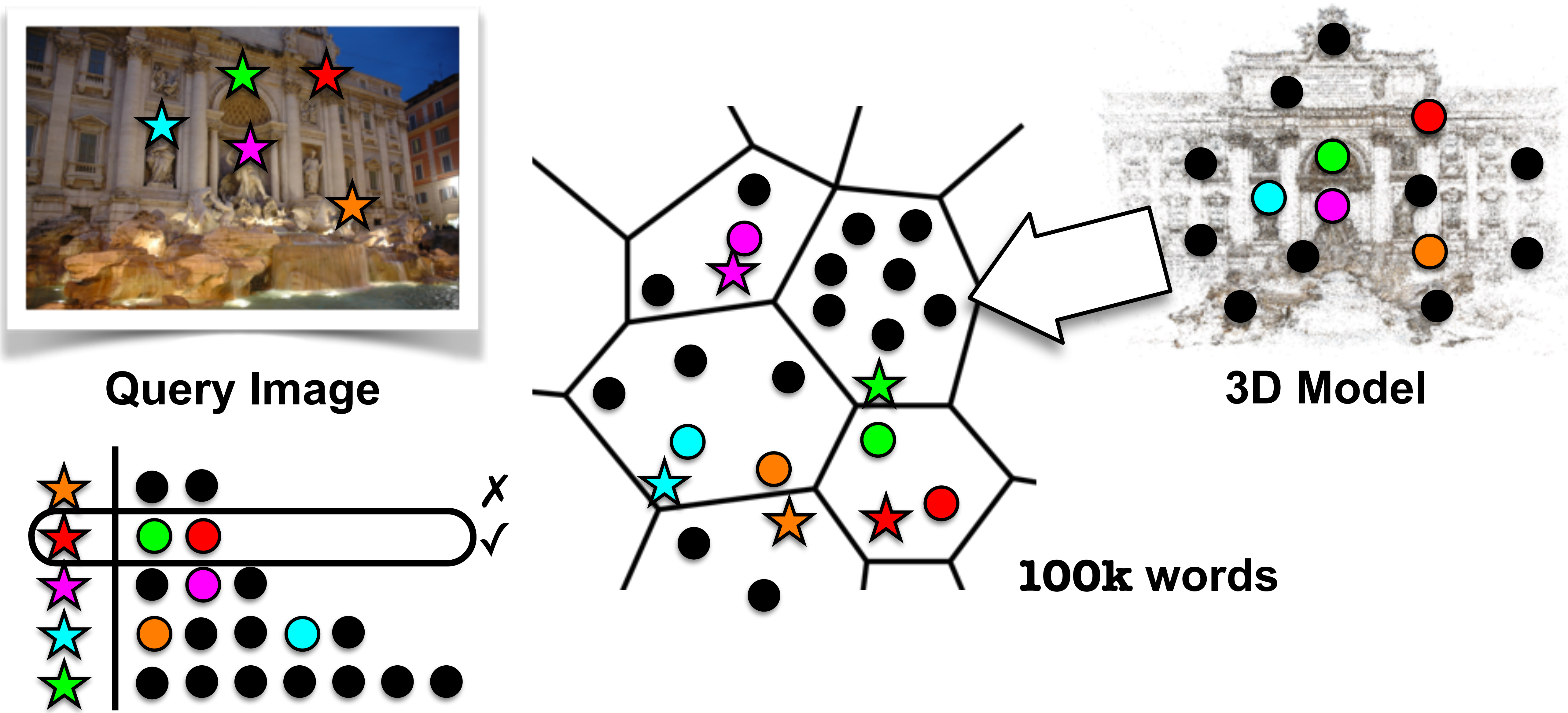
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



Assign features to words

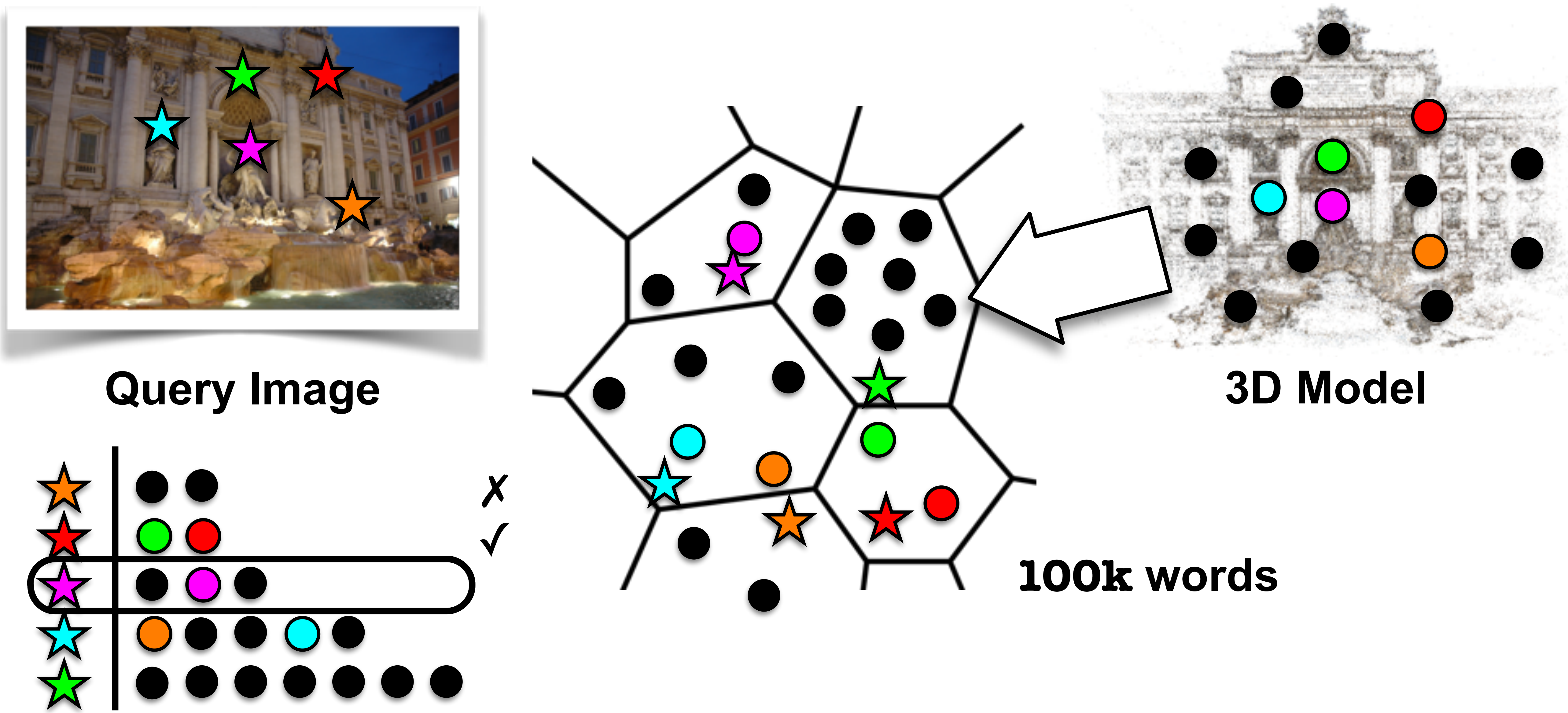
Sort based on costs

Linear search through words

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

Computer Vision and Geometry Group ETH zürich

Vocabulary-Based Prioritized Search (VPS)



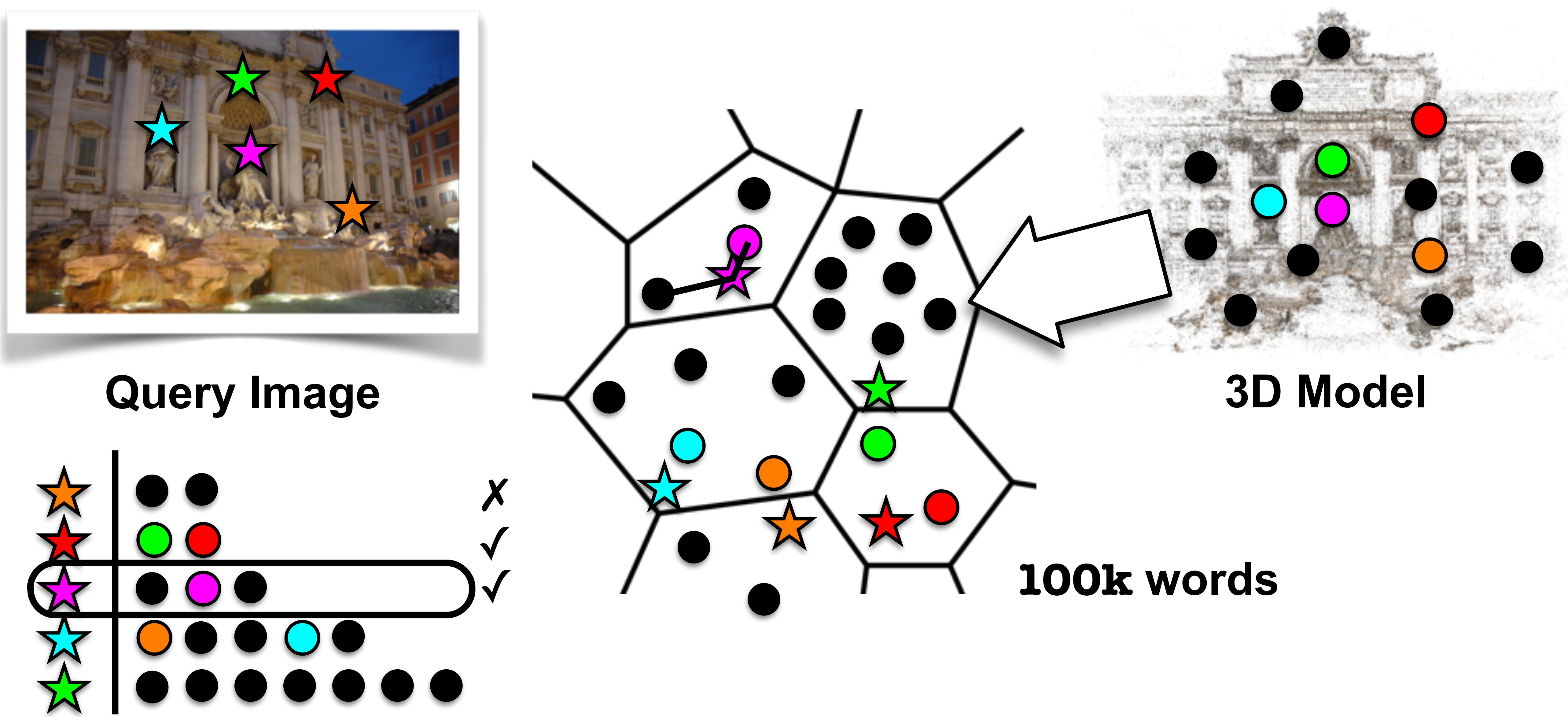
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



Assign features to words

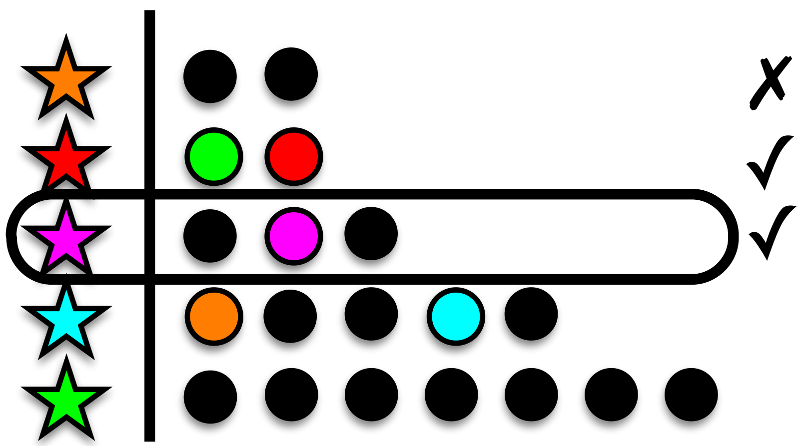
Sort based on costs

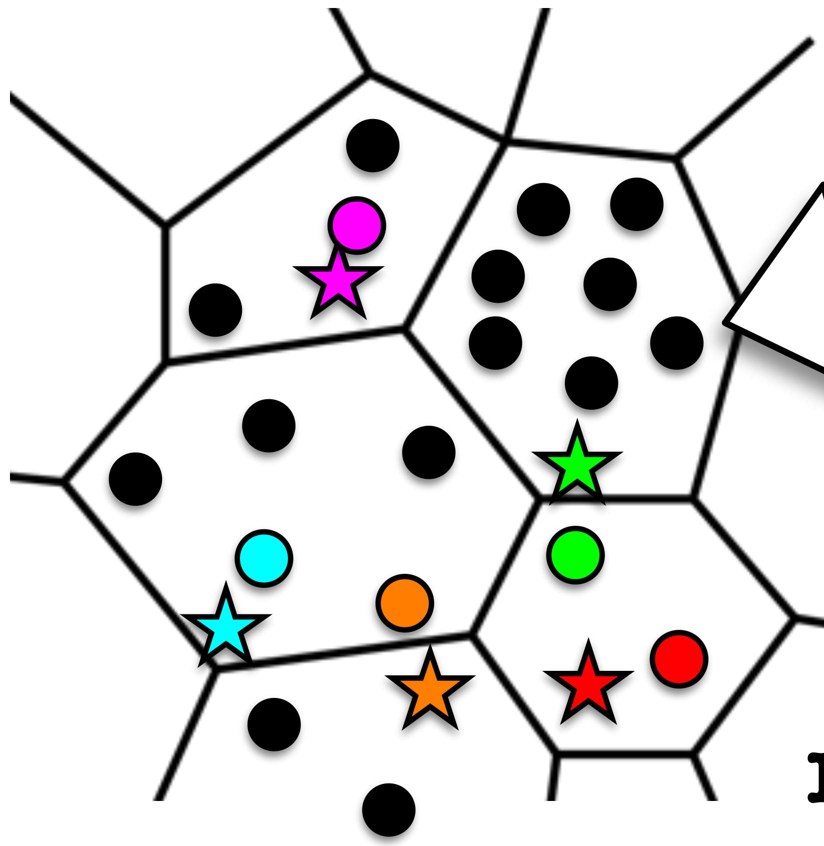
Linear search through words

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

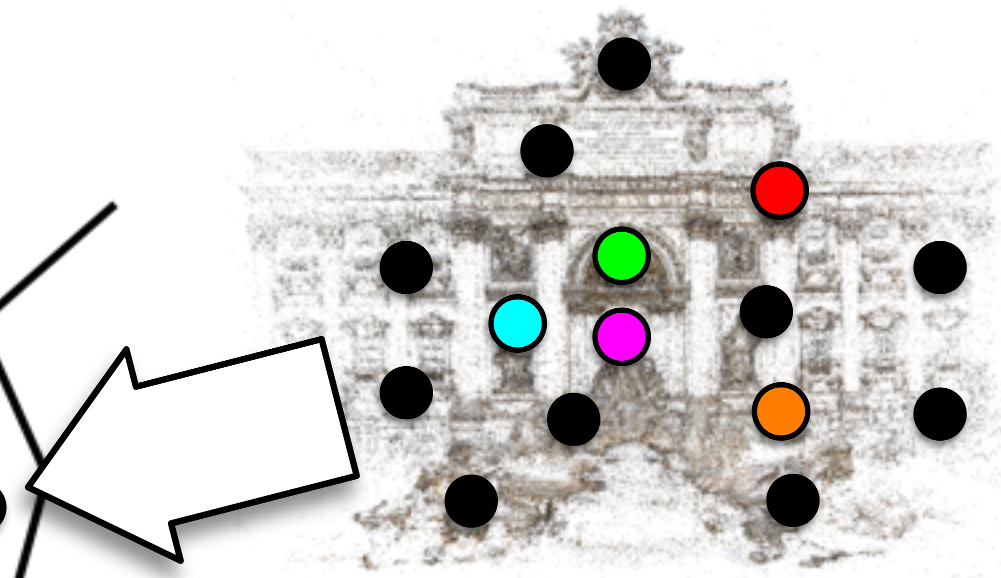
Vocabulary-Based Prioritized Search (VPS)

Query Image





100k words



3D Model

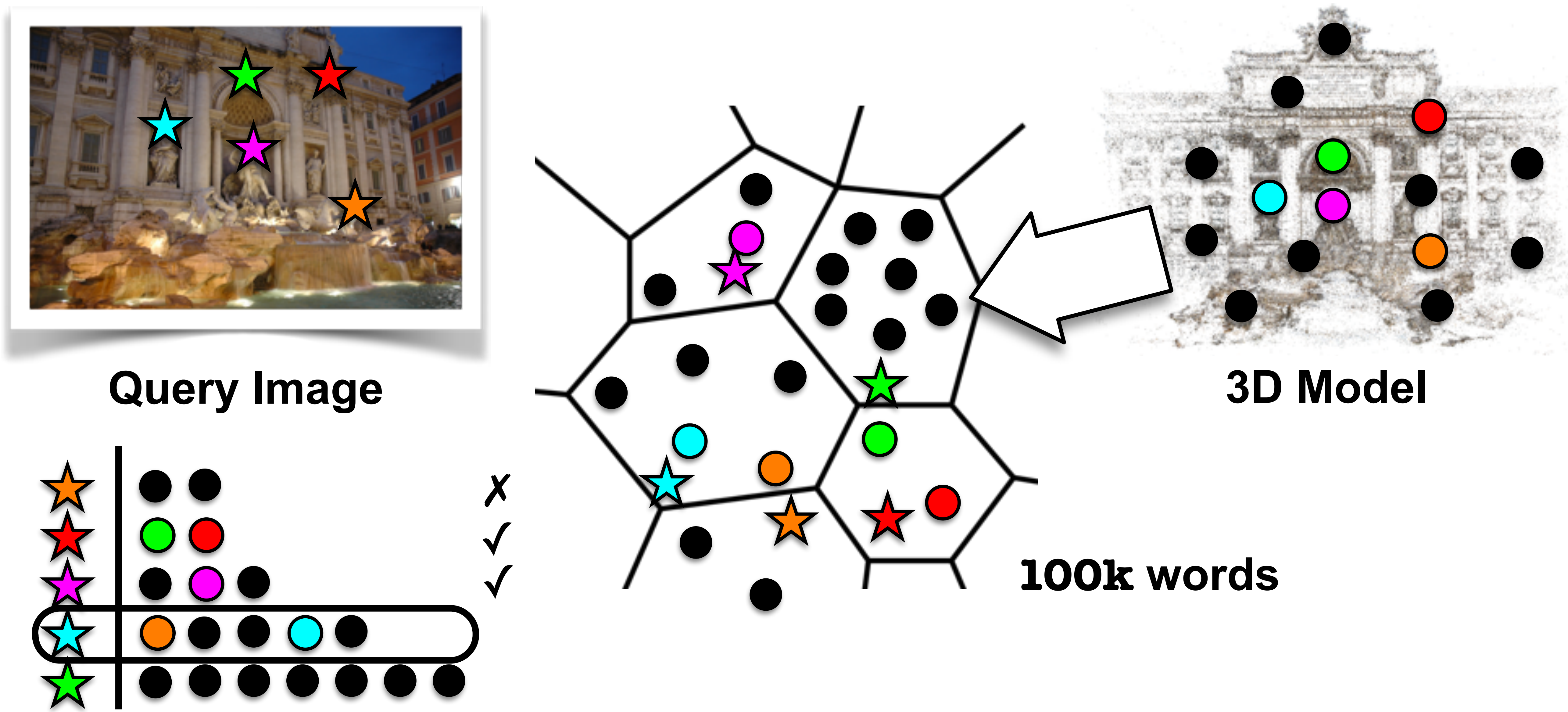
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)

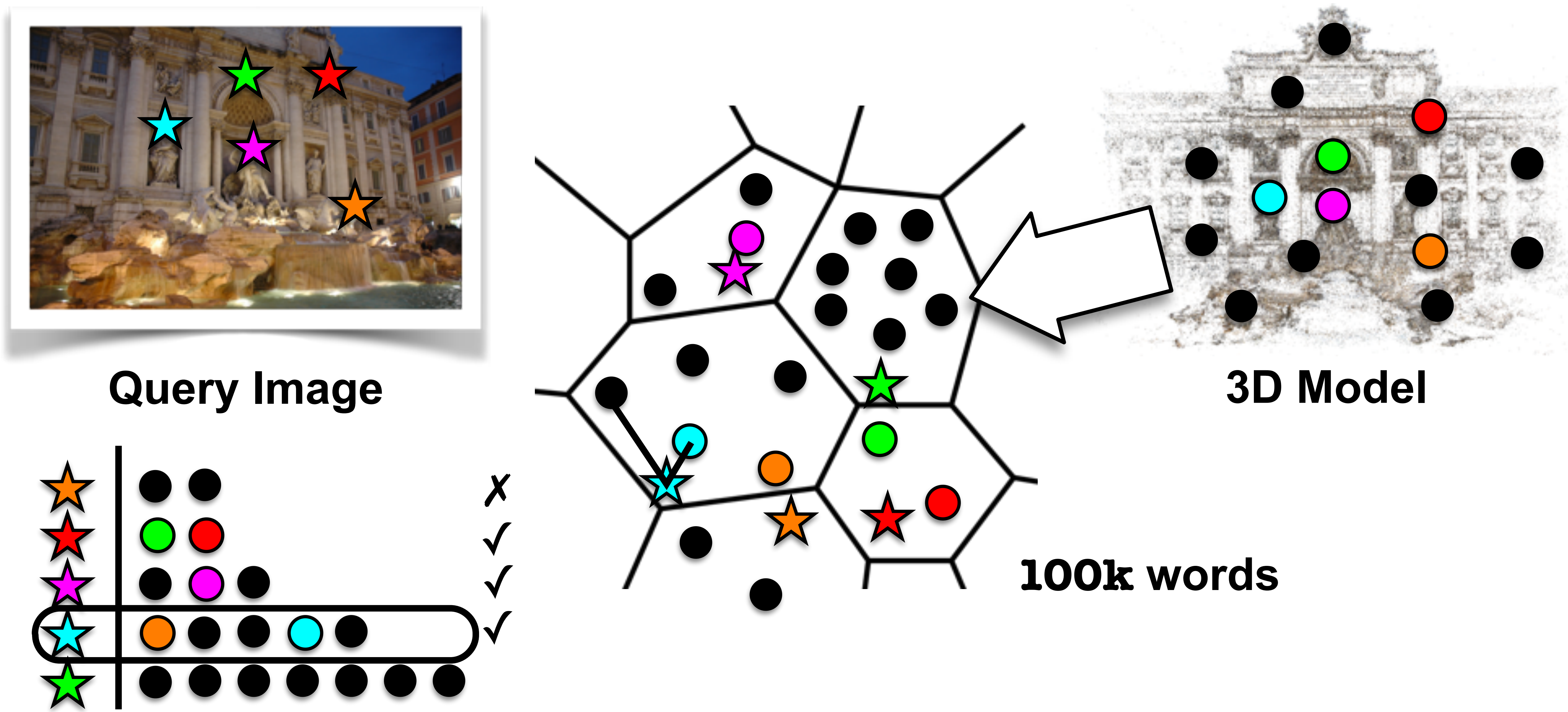


Assign features to words → **Sort based on costs** → **Linear search** through words

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

Computer Vision and Geometry Group ETH zürich

Vocabulary-Based Prioritized Search (VPS)



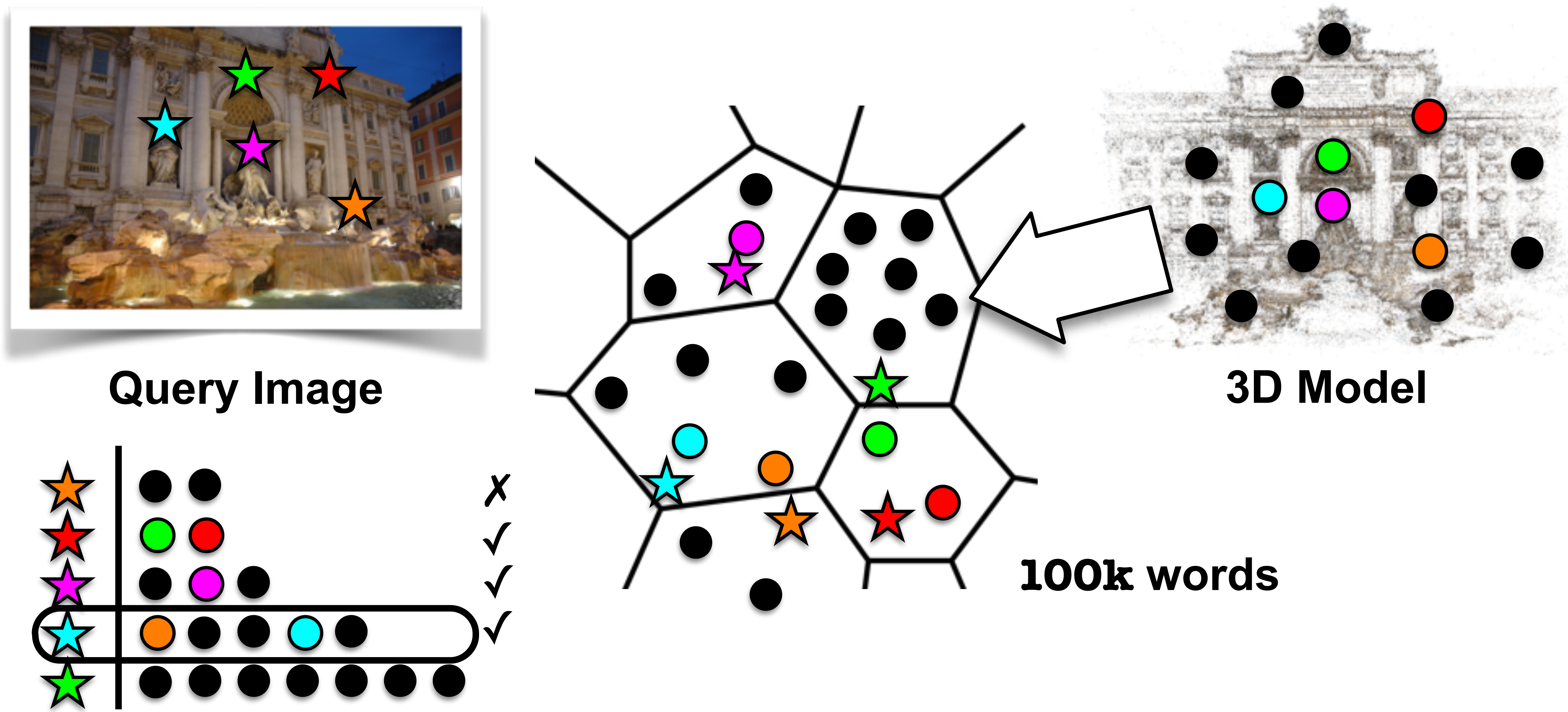
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



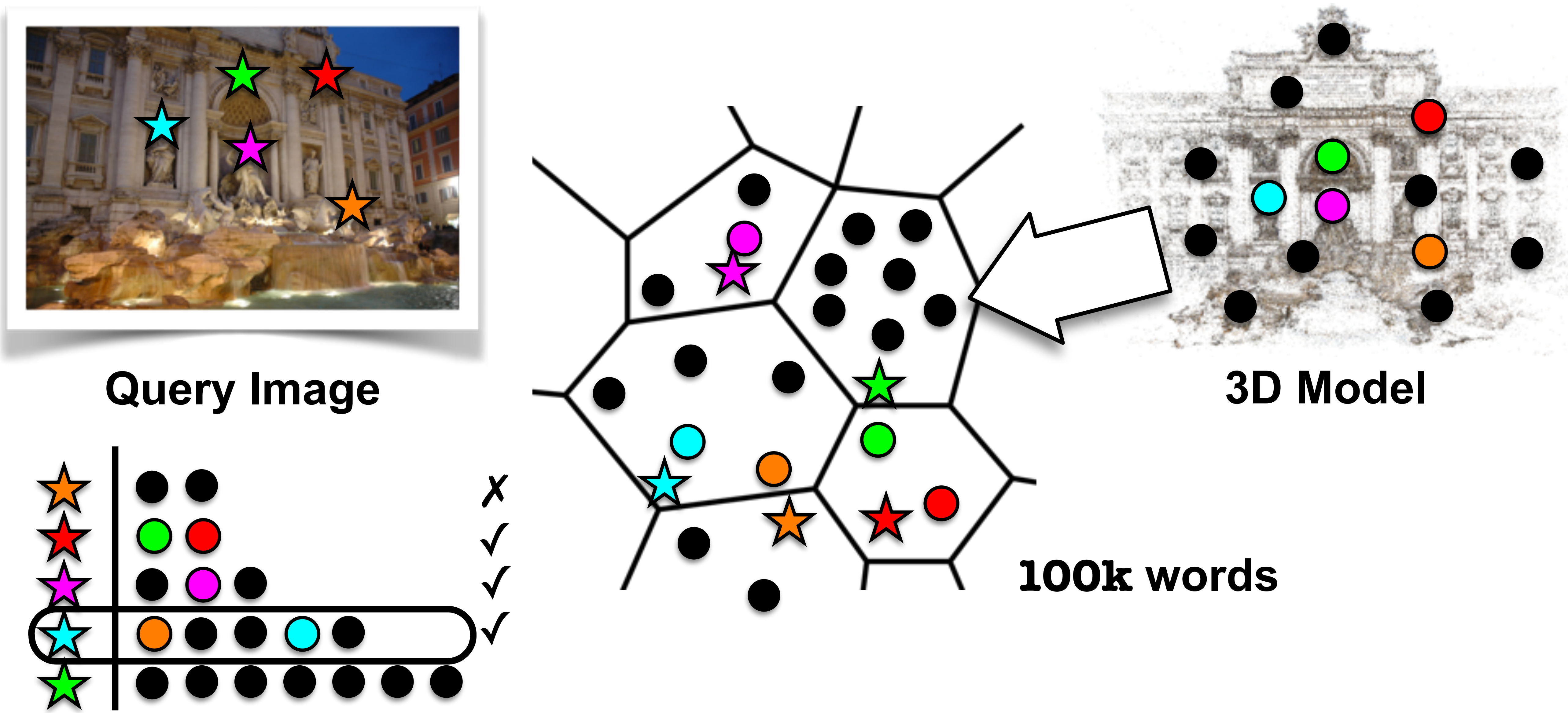
Assign features to words

Sort based on costs

Linear search through words

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

Vocabulary-Based Prioritized Search (VPS)



Assign features to words

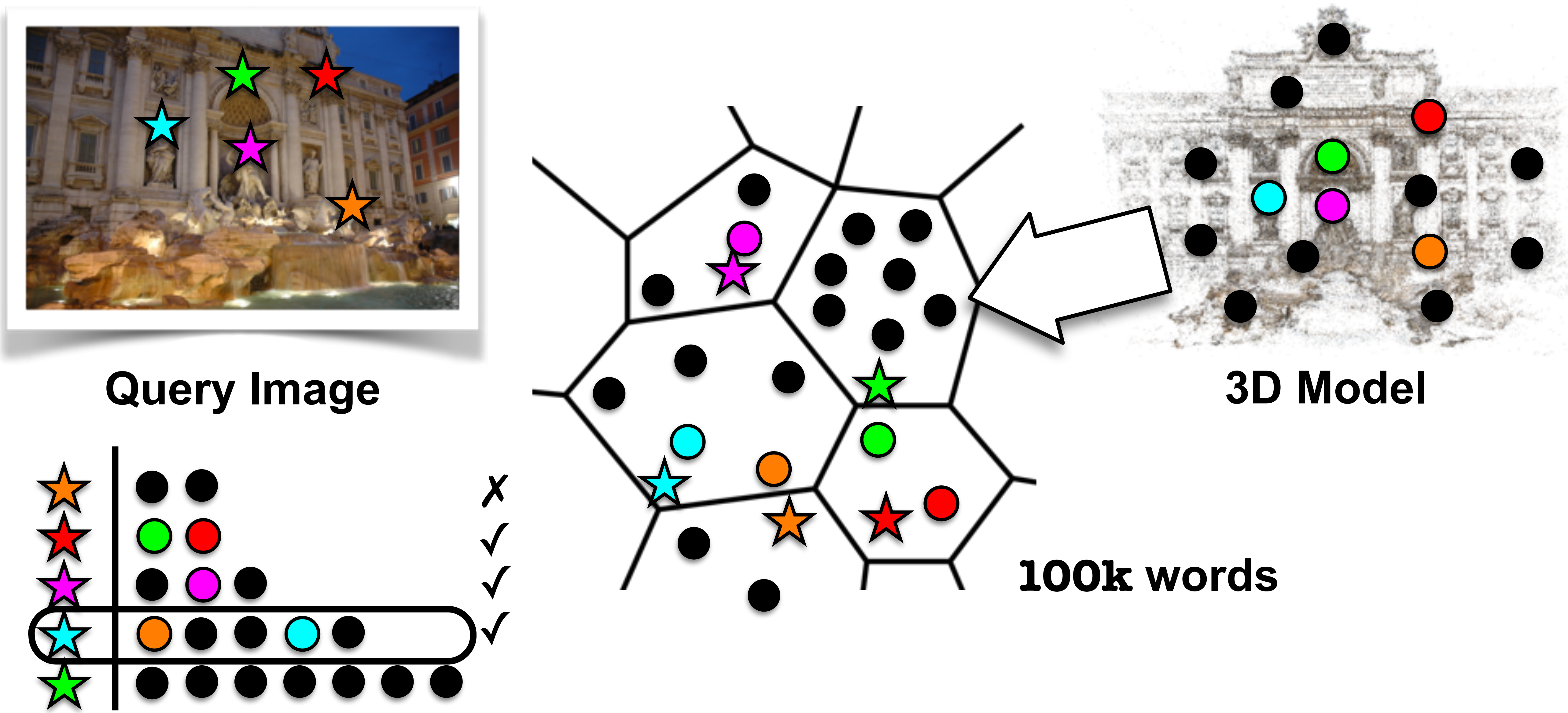
Sort based on costs

Linear search through words

Stop after 100 **matches**

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

Vocabulary-Based Prioritized Search (VPS)

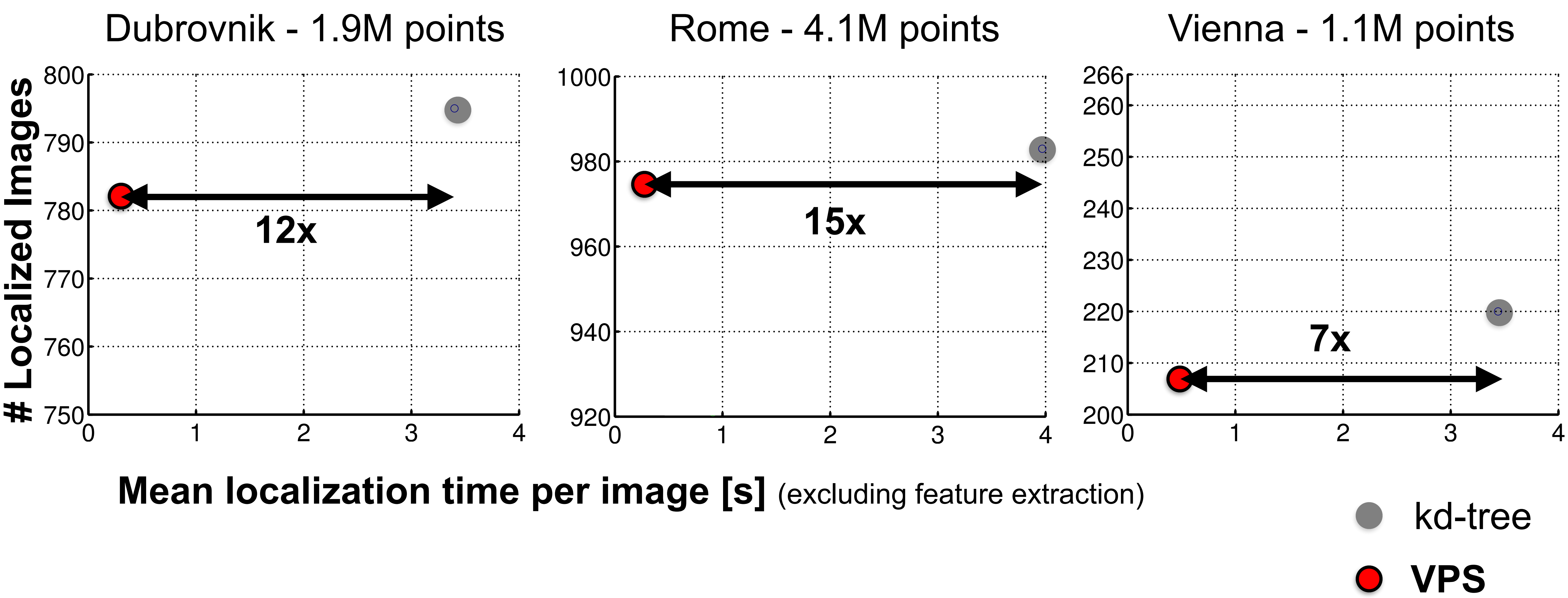


Assign features to words	Sort based on costs	Linear search through words	Stop after 100 matches	Pose estimation: RANSAC + p6p

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

Computer Vision and Geometry Group ETH zürich

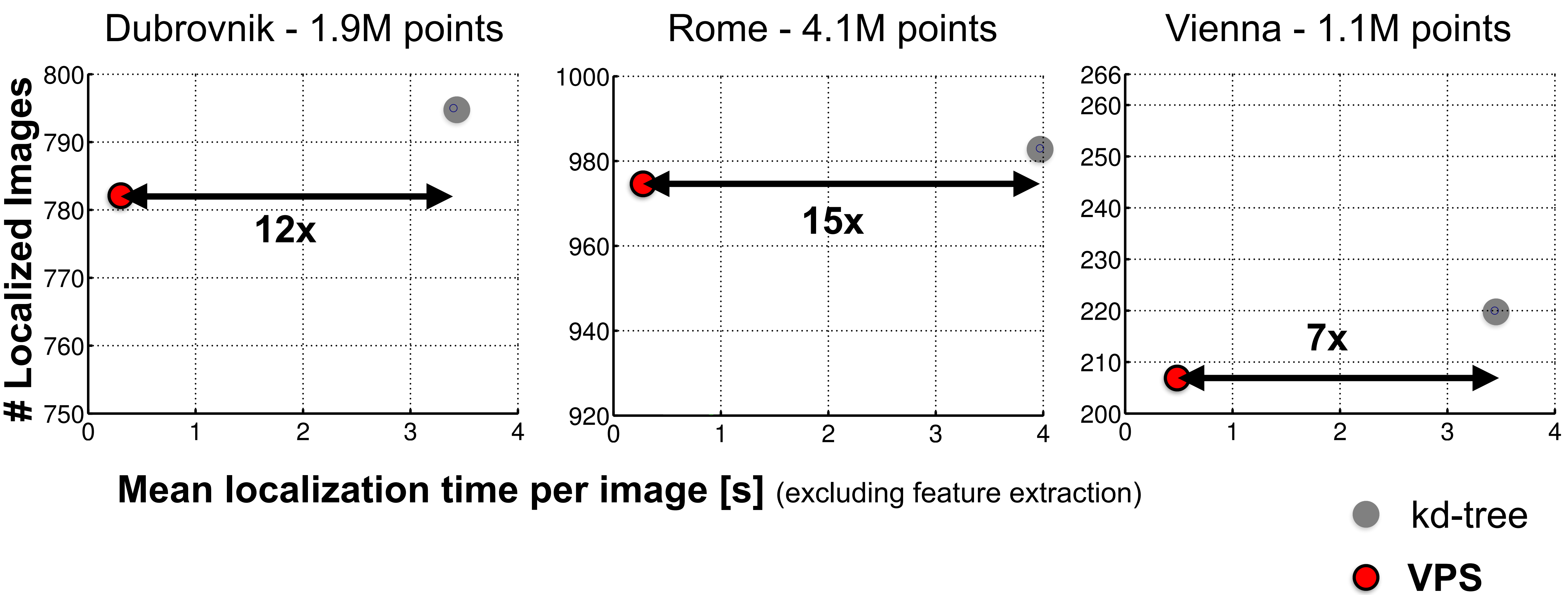
Results



Computer Vision and Geometry Group

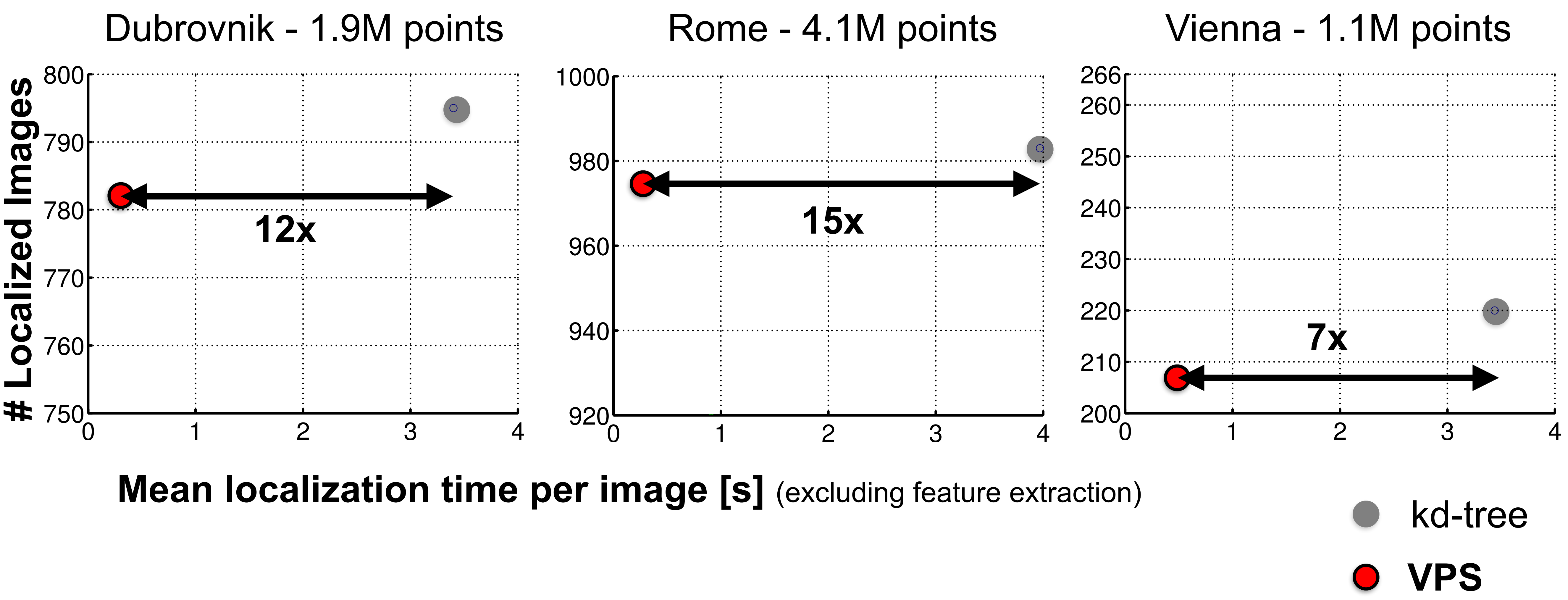
ETH zürich

Results



✓ 10x faster than kd-tree

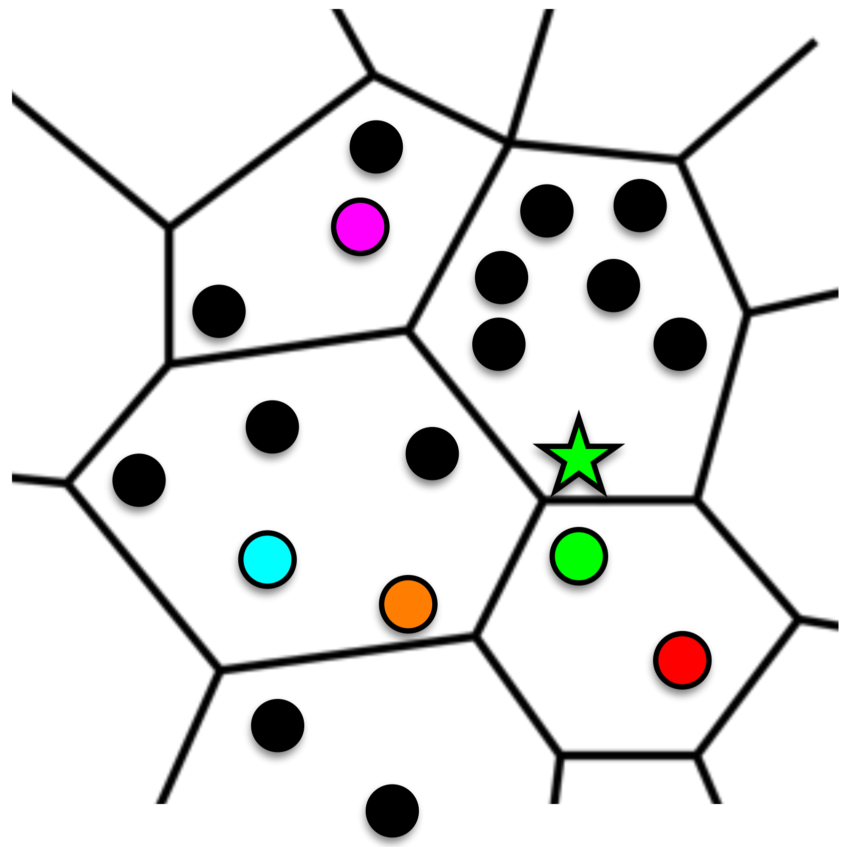
Results



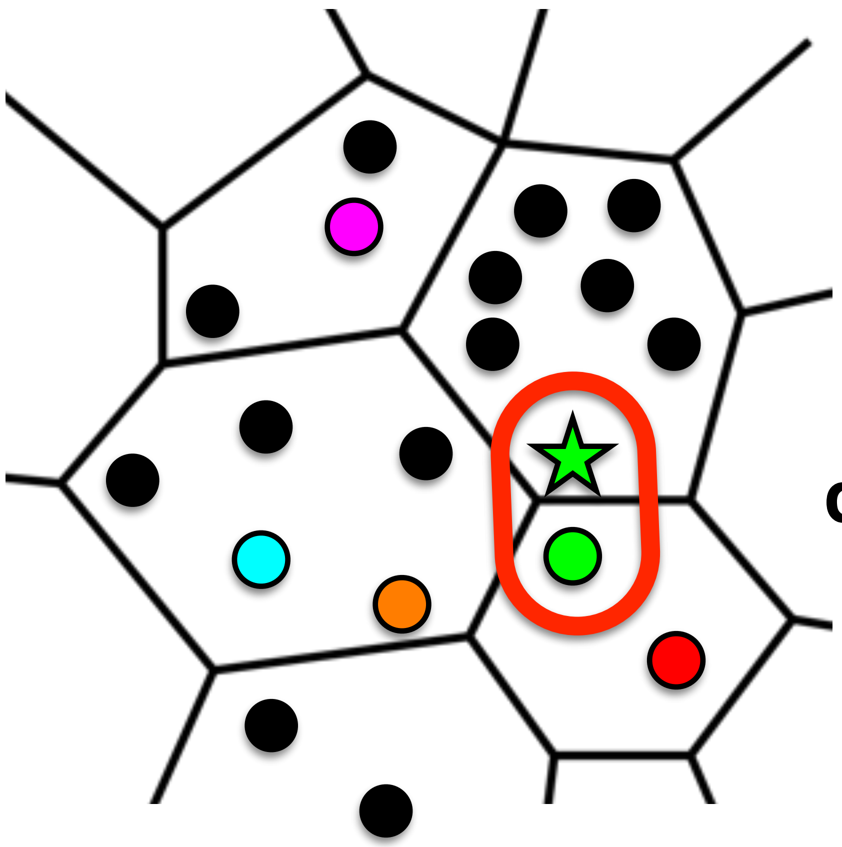
✓ 10x faster than kd-tree

✗ … but less effective than kd-tree

Quantization Artifacts

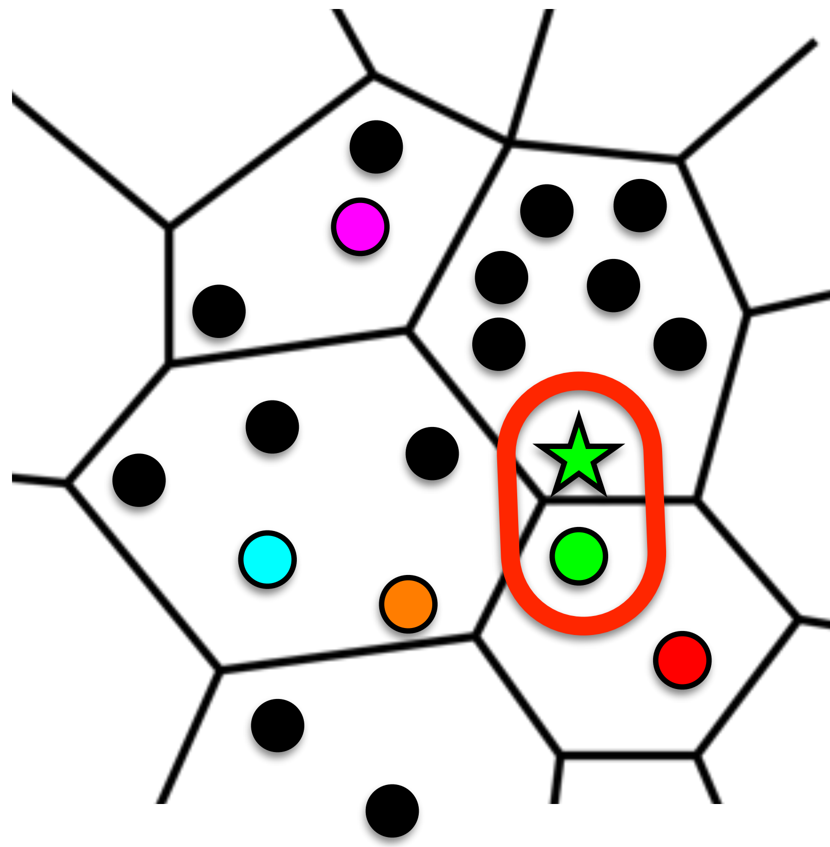


Quantization Artifacts



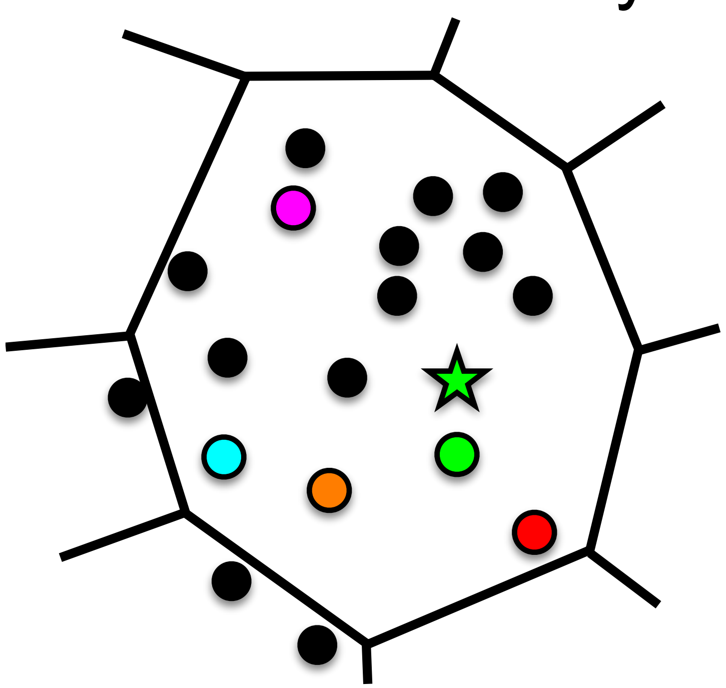
Match missed due to quantization!

Quantization Artifacts

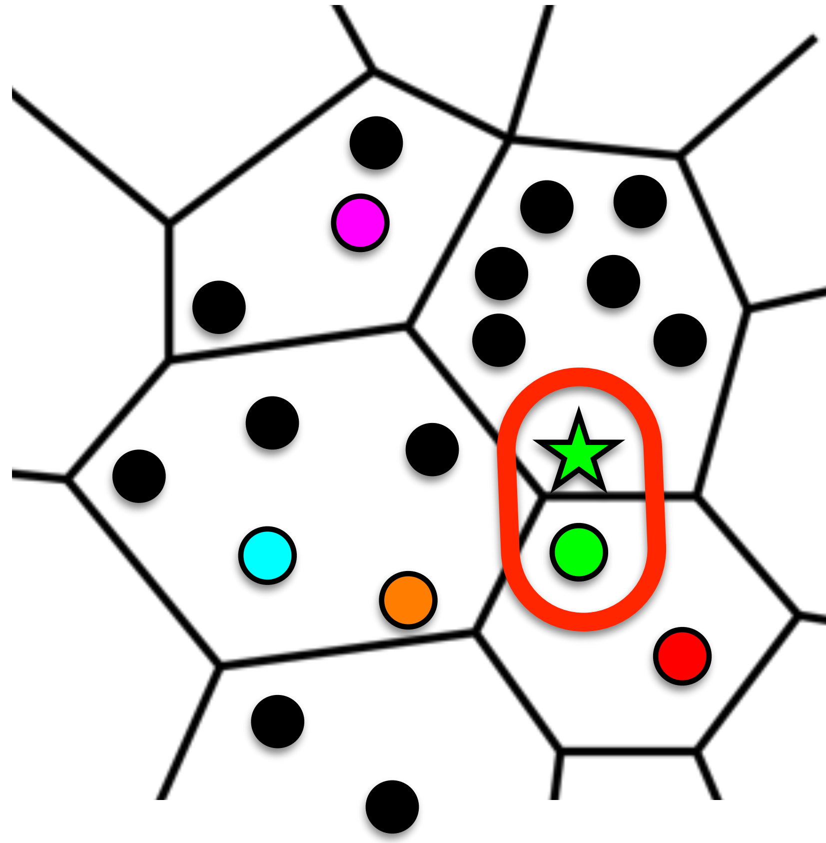


Match missed due to quantization!

Smaller Vocabulary

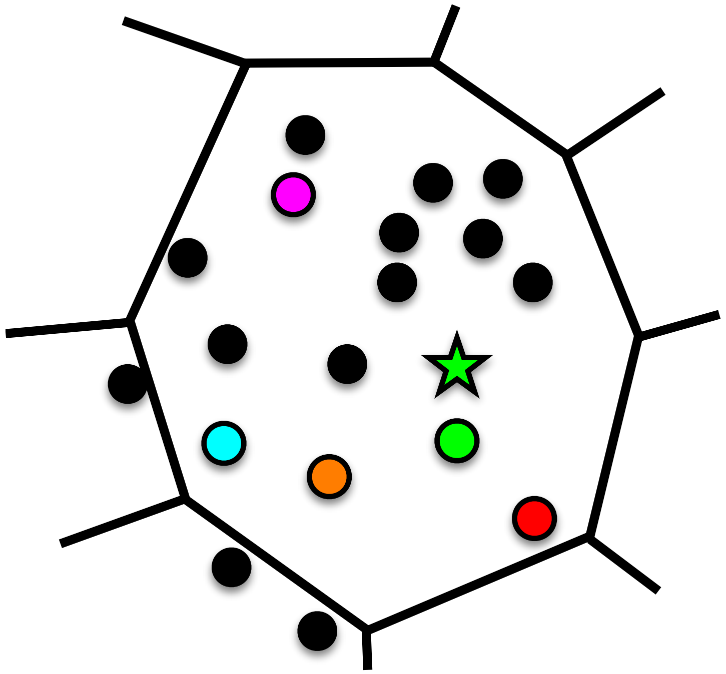


Quantization Artifacts

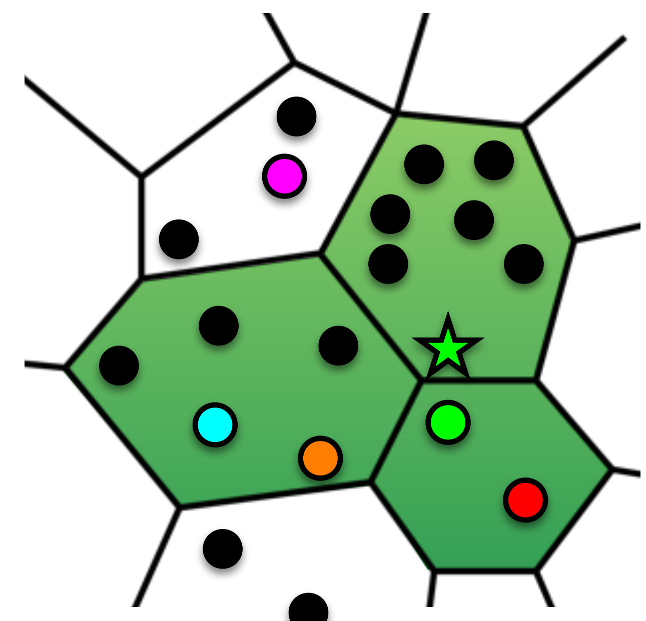


Match missed due to quantization!

Smaller Vocabulary

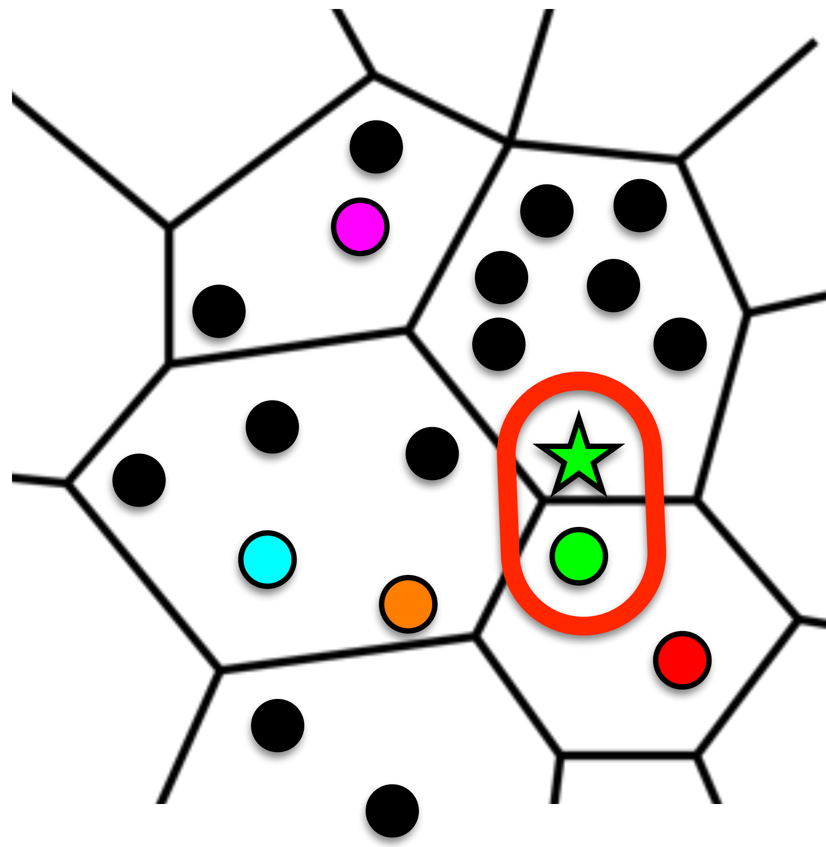


Soft Assignments



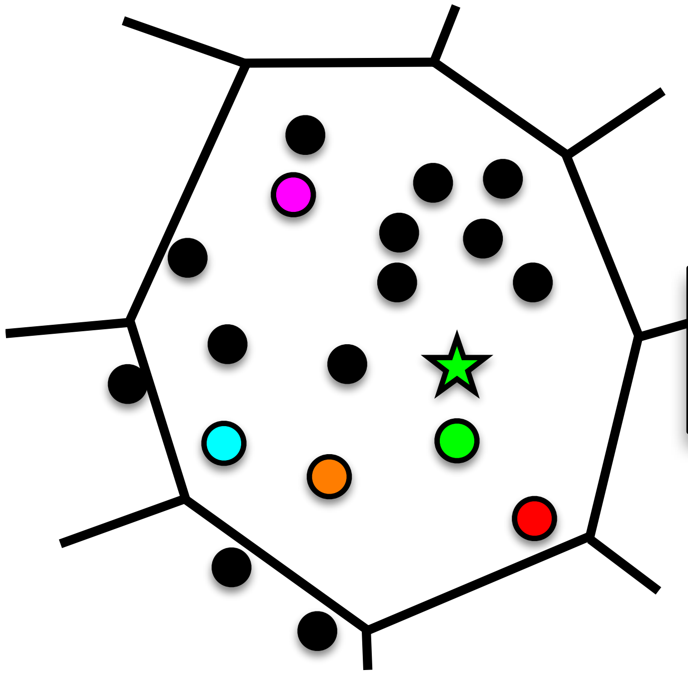
[Philbin et al., CPVR'08]

Quantization Artifacts



Match missed due to quantization!

Smaller Vocabulary



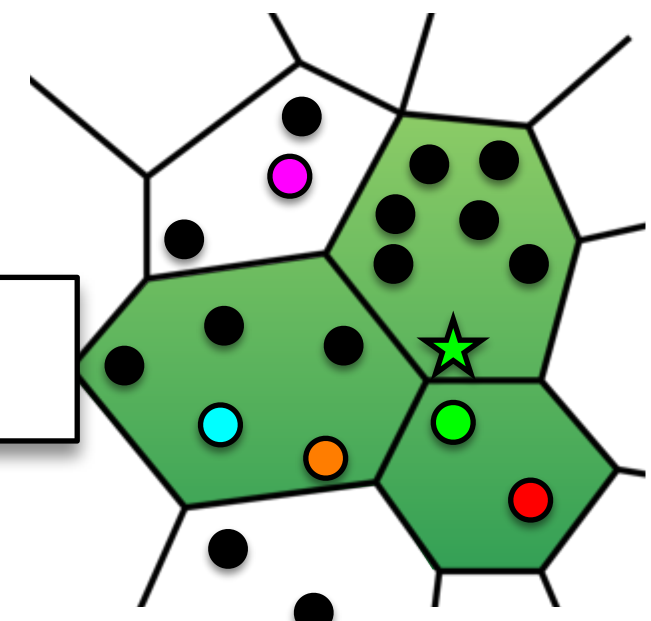
Additional costs

for each feature

$$\mathcal{O}(P/W)$$

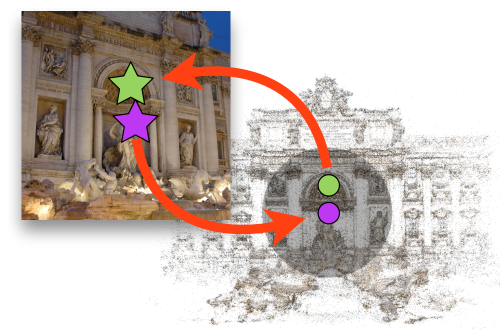
W words, P points

Soft Assignments



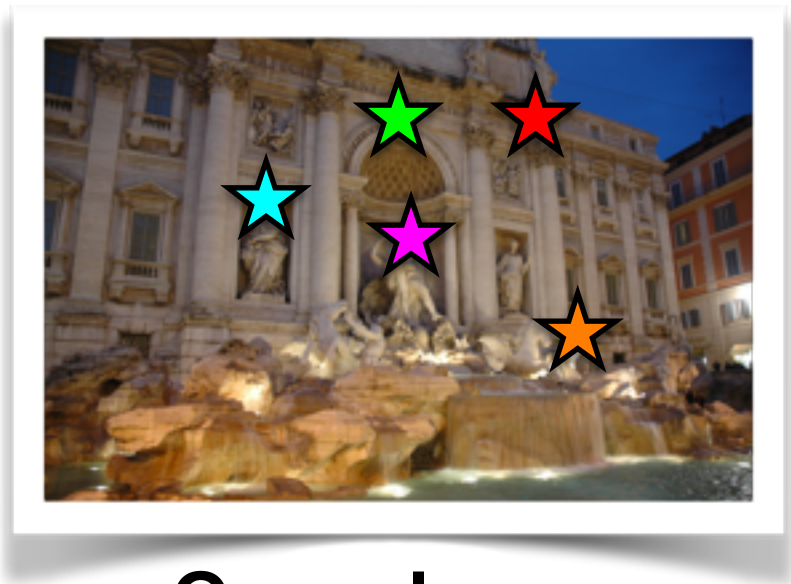
[Philbin et al., CPVR'08]

Localization - Overview

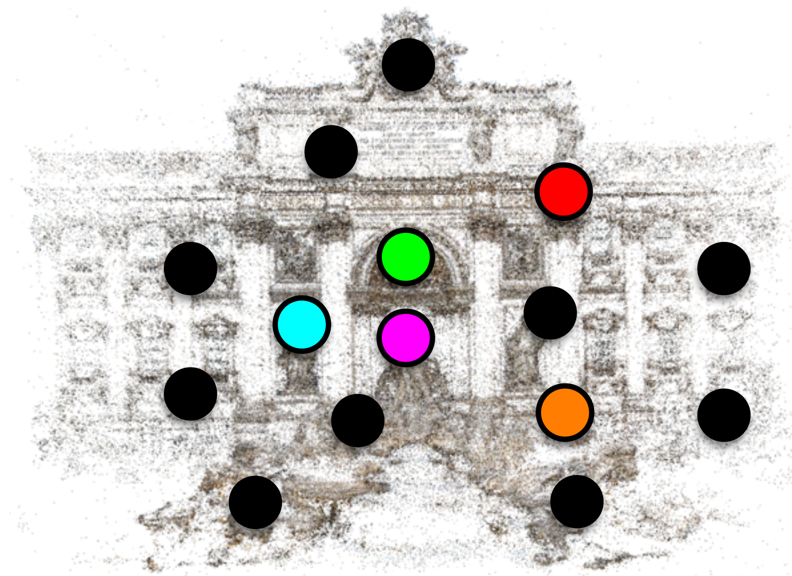
				
	Baseline: kd-tree search	**VPS**	**Active Search**	**+ Visibility Filtering**
	[Sattler et al., ICCV'11]	[Sattler et al., ICCV'11]	[Sattler et al., ECCV'12]	
effectiveness	✓	✗	✓✓	✓✓
efficiency	✗ ✗	✓	✗	✓

Active Search

Idea: Exploit **co-occurrence of matches** to recover matches



Query Image



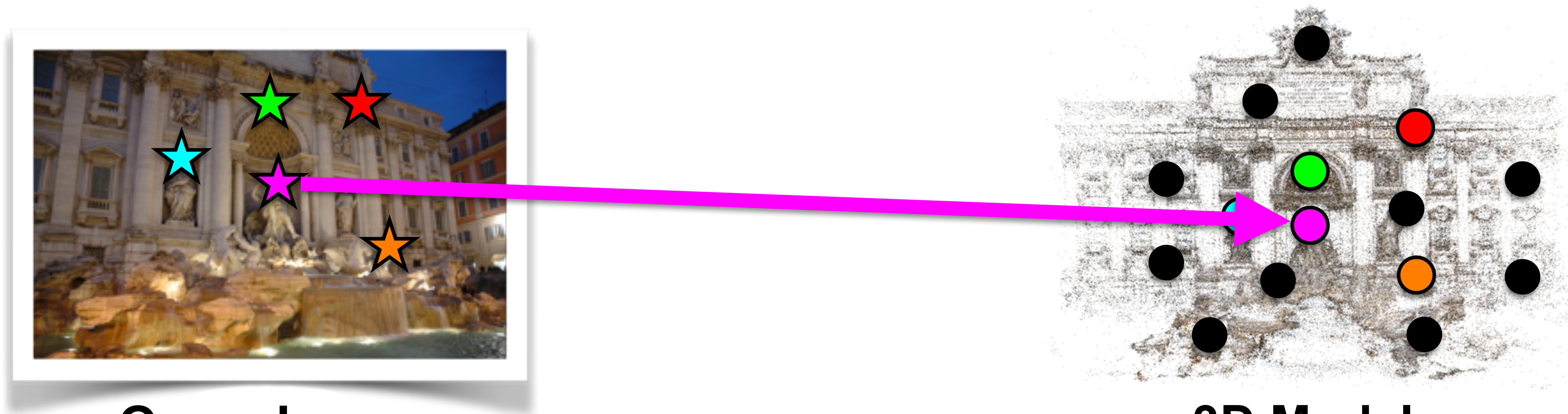
3D Model

[Sattler et al., ECCV’12] [code]

Computer Vision and Geometry Group ETH zürich

Active Search

Idea: Exploit **co-occurrence of matches** to recover matches



Query Image **3D Model**

[Sattler et al., ECCV’12] [code]

Active Search

Idea: Exploit **co-occurrence of matches** to recover matches

- Points surrounding 2D-to-3D match should also be visible:

[Sattler et al., ECCV'12] [code]

Active Search

Idea: Exploit **co-occurrence of matches** to recover matches

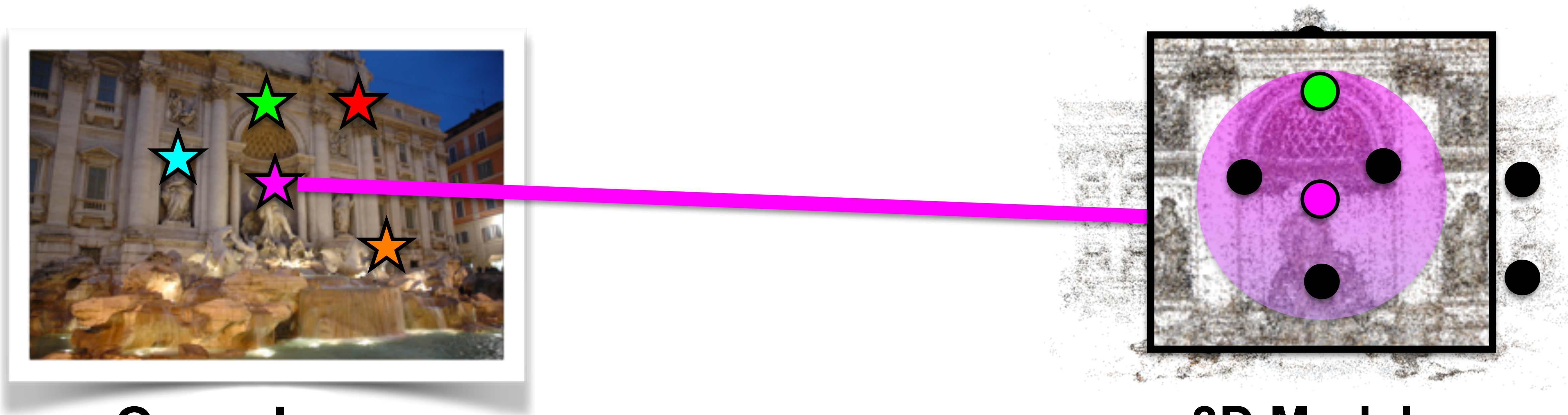
- Points surrounding 2D-to-3D match should also be visible:

[Sattler et al., ECCV'12] [code]

Computer Vision and Geometry Group ETH zürich

Active Search

Idea: Exploit **co-occurrence of matches** to recover matches



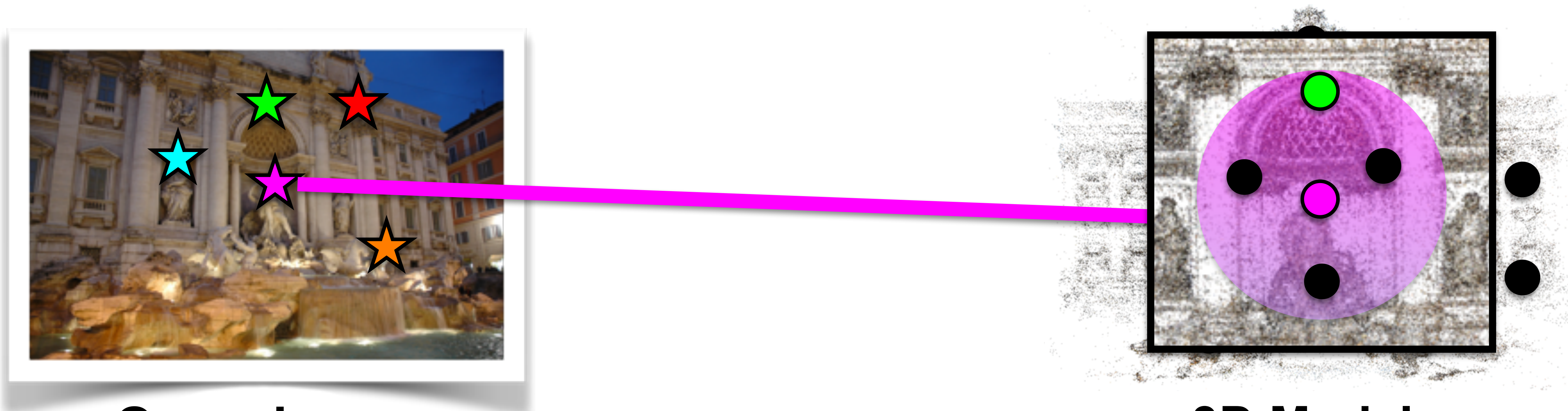
Query Image **3D Model**

- Points surrounding 2D-to-3D match should also be visible:
  - Find nearest neighbors in 3D around matching point

[Sattler et al., ECCV'12] [code]

Active Search

Idea: Exploit **co-occurrence of matches** to recover matches



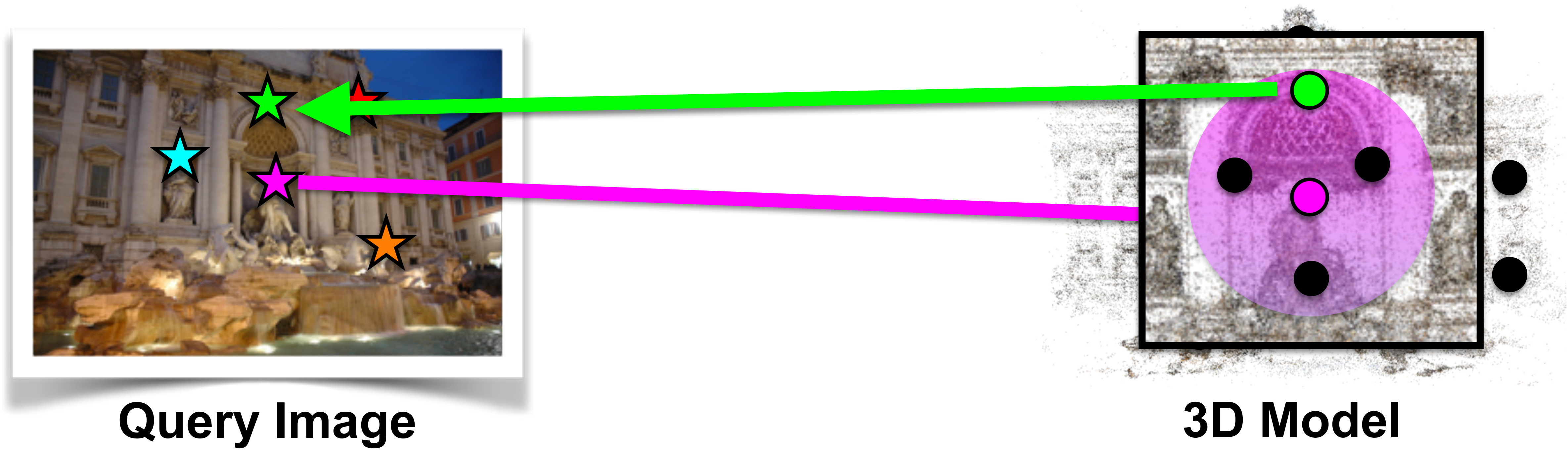
Query Image **3D Model**

- Points surrounding 2D-to-3D match should also be visible:
  - Find nearest neighbors in 3D around matching point
  - Perform 3D-to-2D search for neighbors

[Sattler et al., ECCV'12] [code]

Active Search

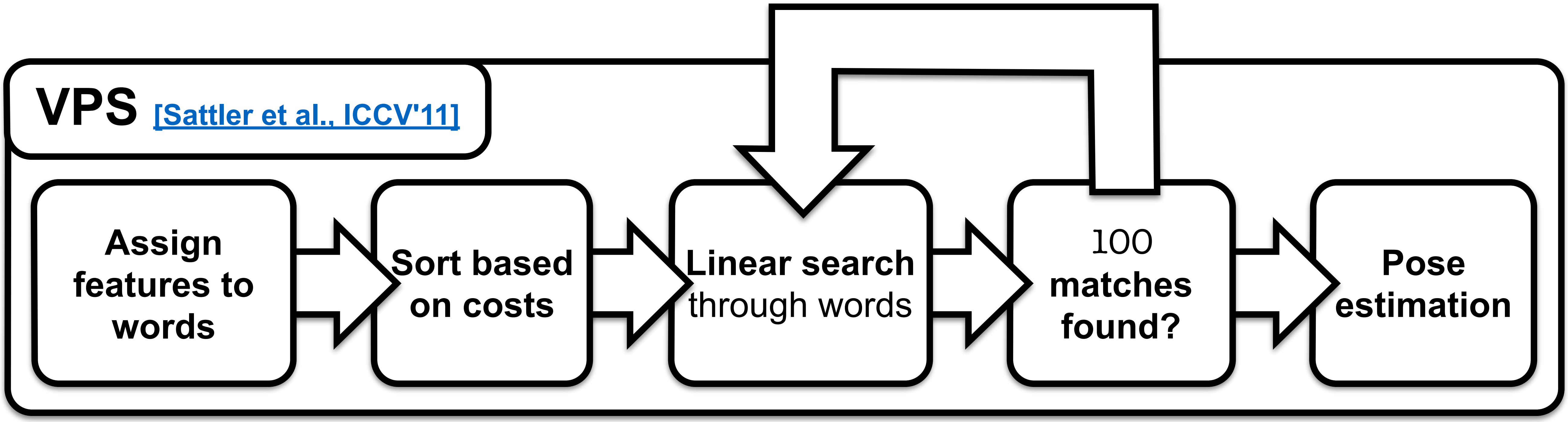
Idea: Exploit **co-occurrence of matches** to recover matches



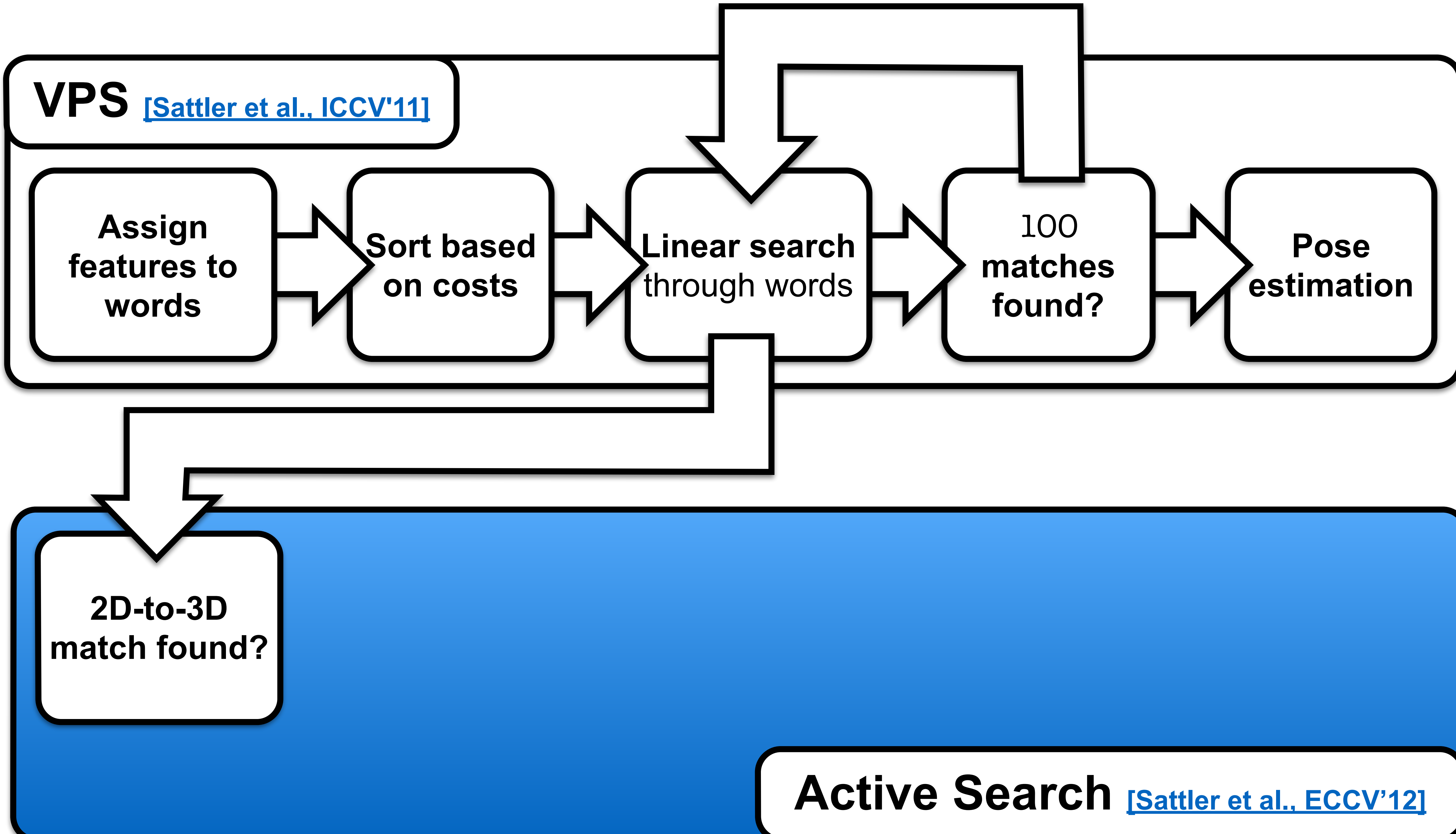
- Points surrounding 2D-to-3D match should also be visible:
  - Find nearest neighbors in 3D around matching point
  - Perform 3D-to-2D search for neighbors

[Sattler et al., ECCV'12] [code]

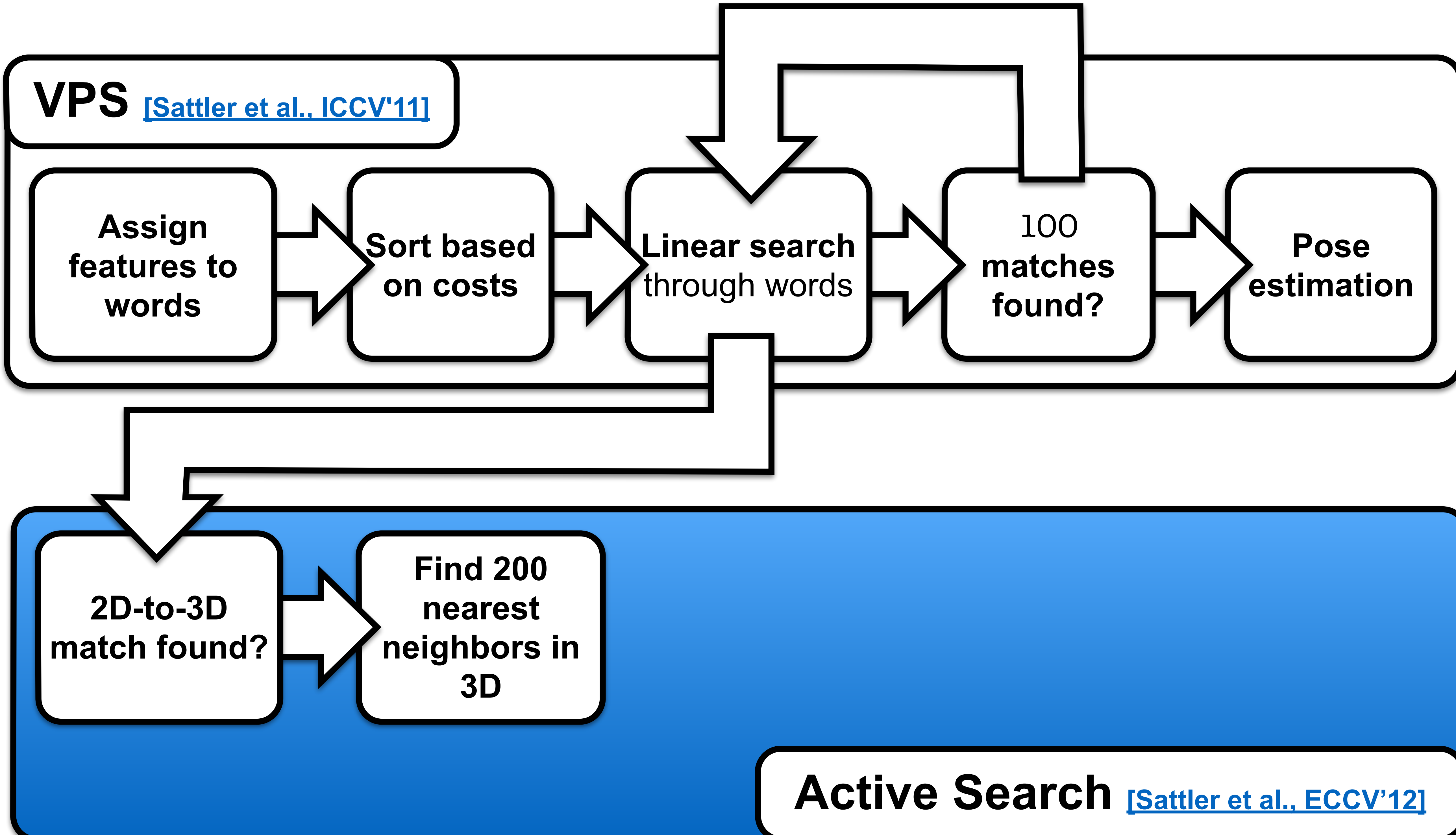
Integration Into VPS



Integration Into VPS

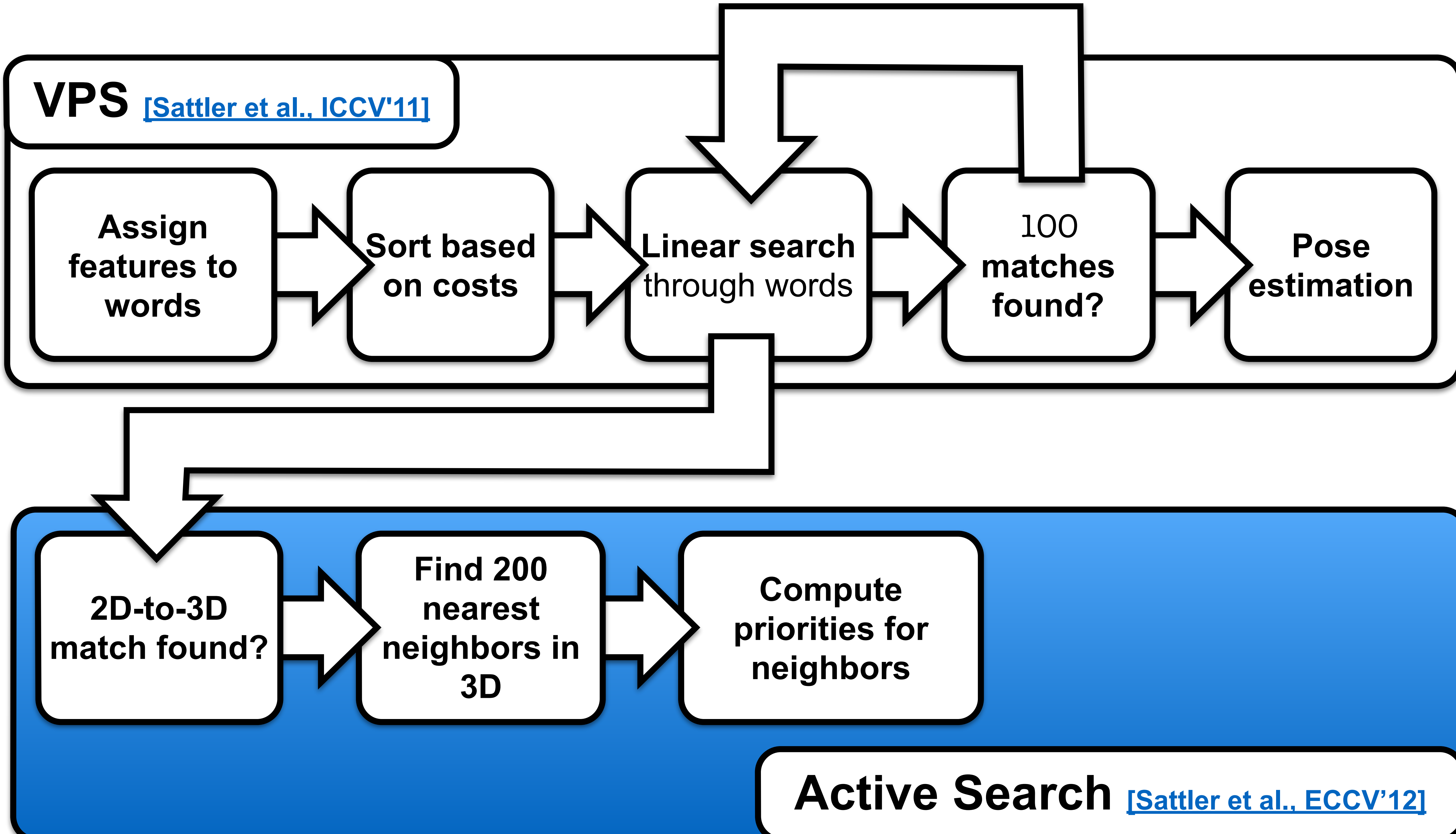


Integration Into VPS

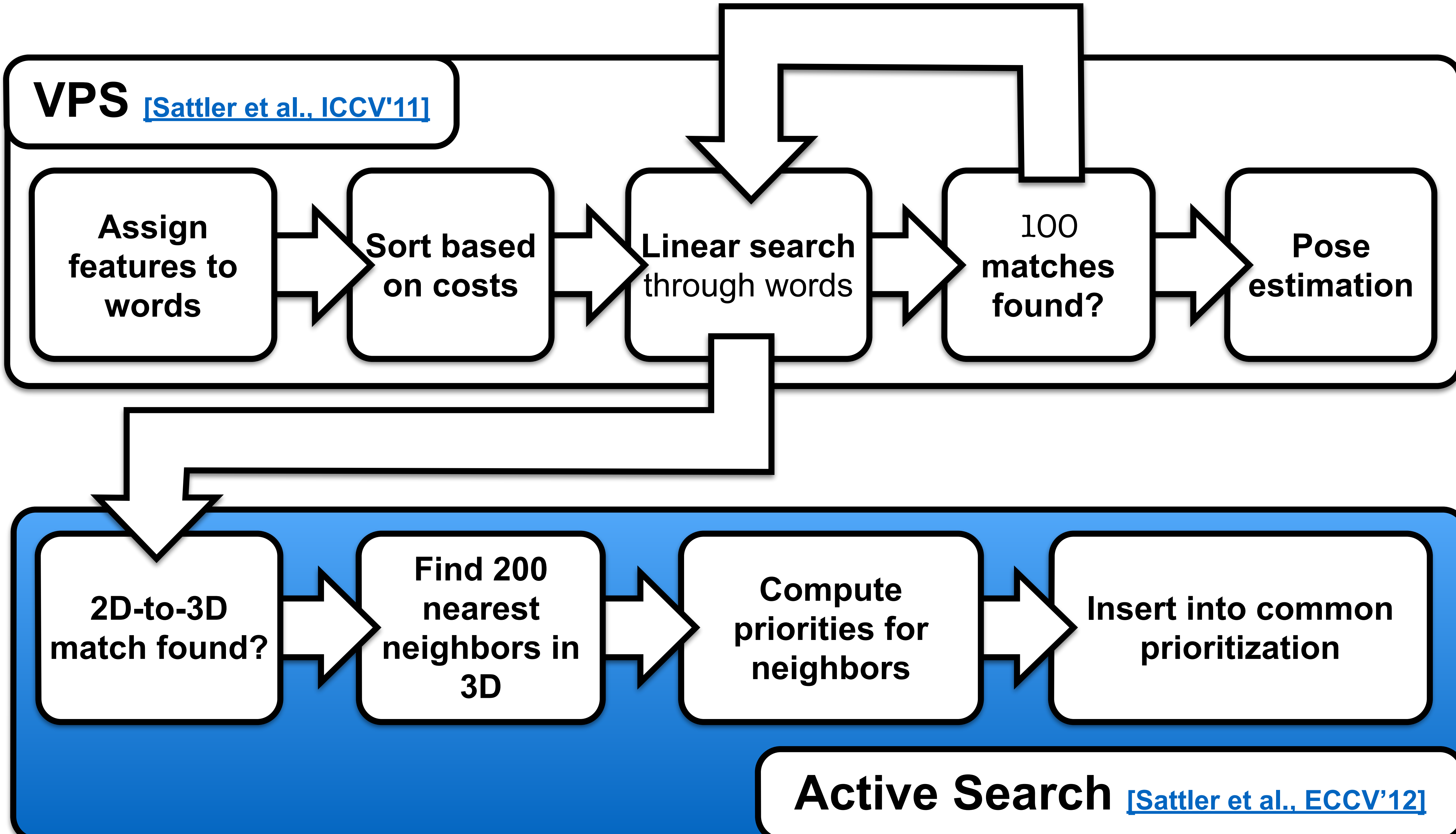


Computer Vision and Geometry Group ETH zürich

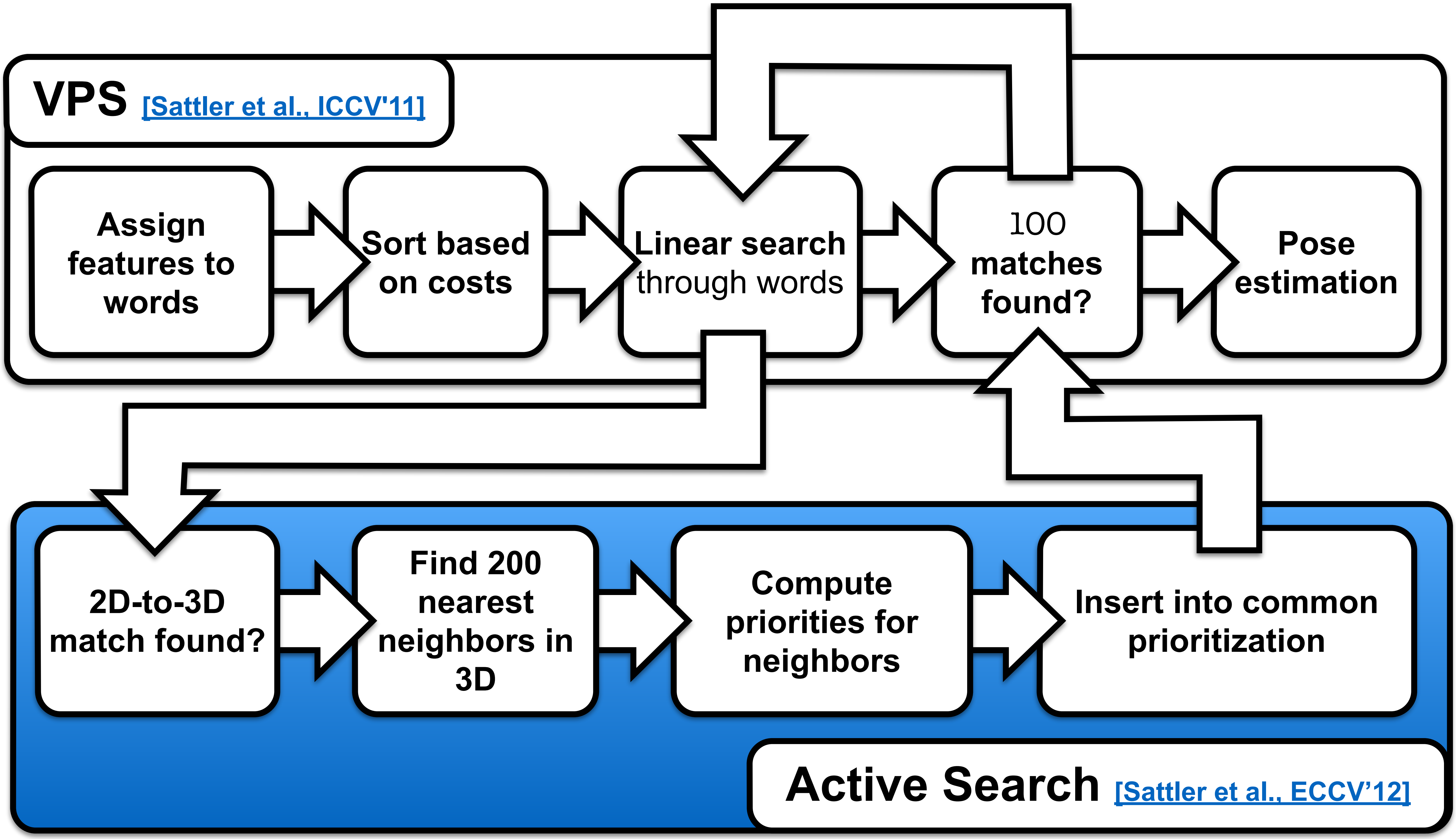
Integration Into VPS



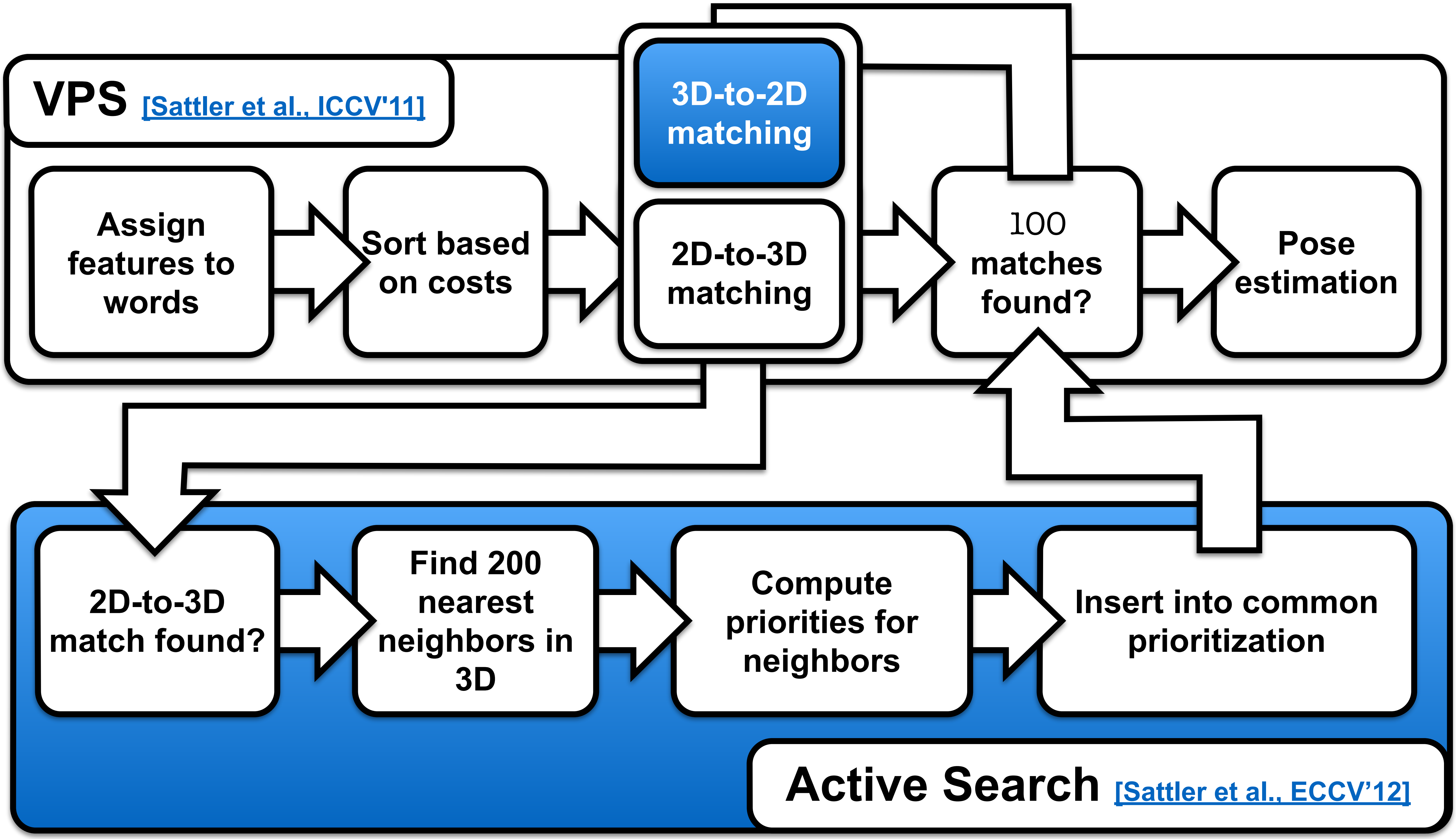
Integration Into VPS



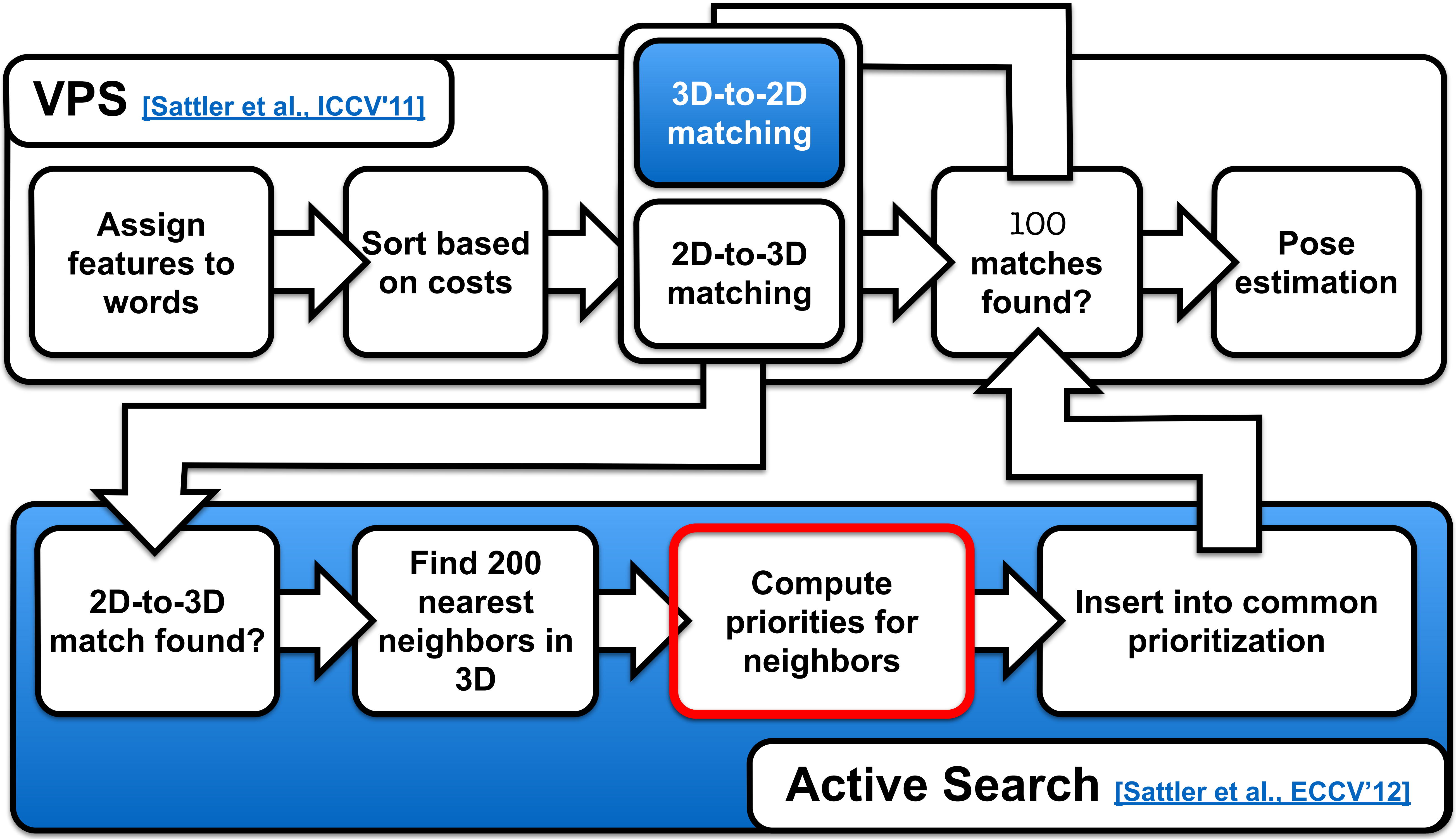
Integration Into VPS



Integration Into VPS

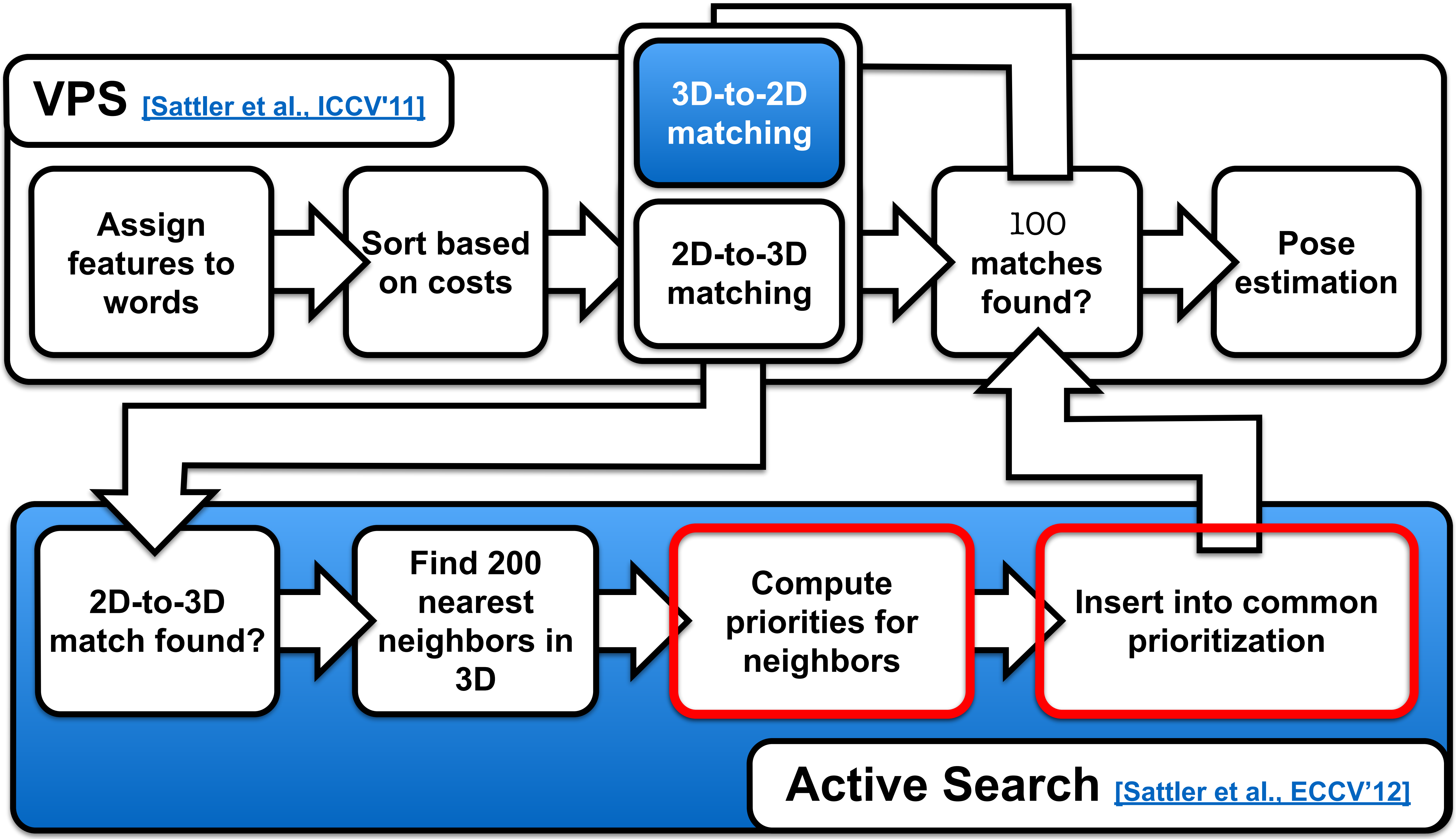


Integration Into VPS

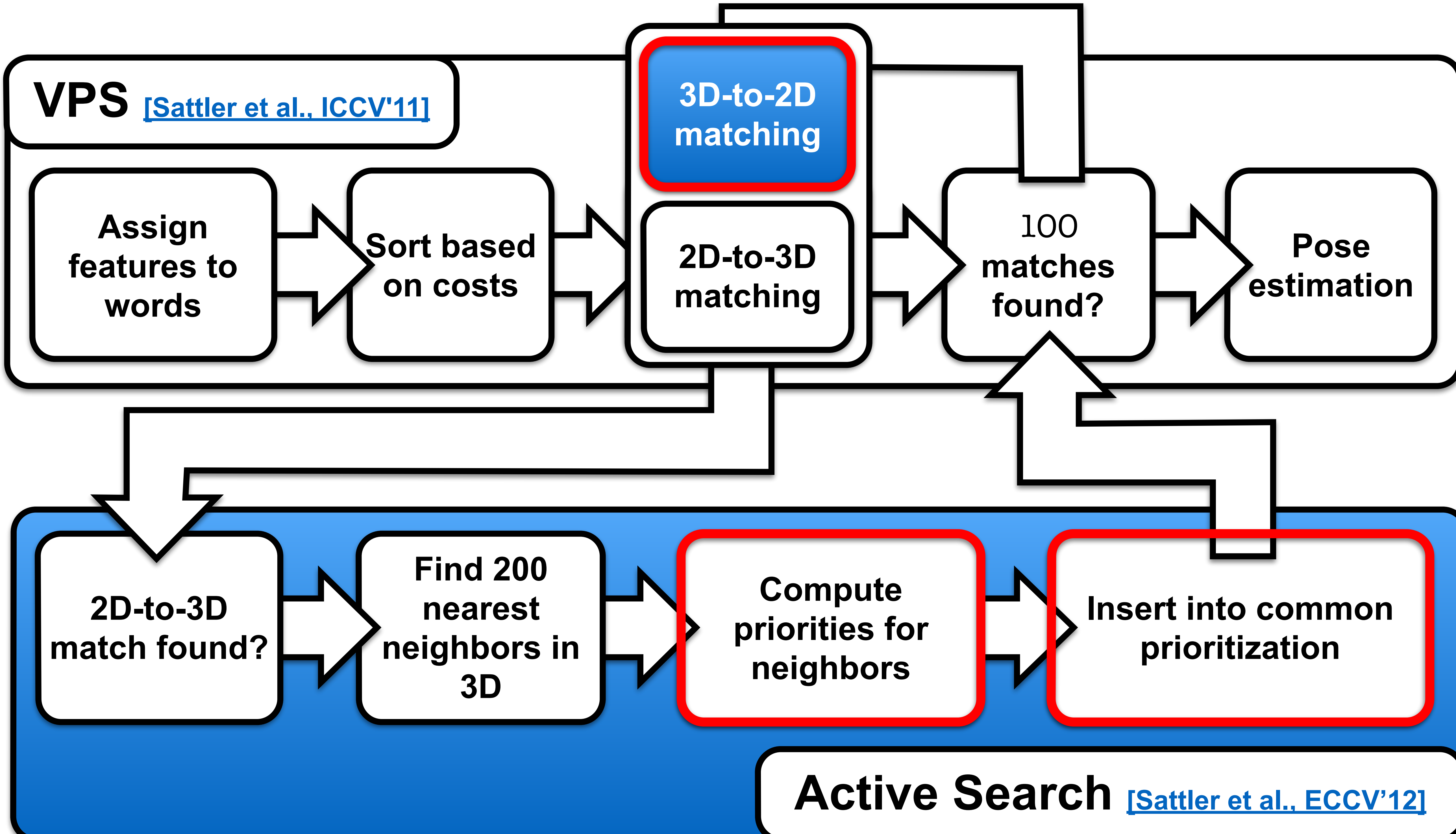


Computer Vision and Geometry Group ETH zürich

Integration Into VPS



Integration Into VPS



Integration 3D-to-2D Matching

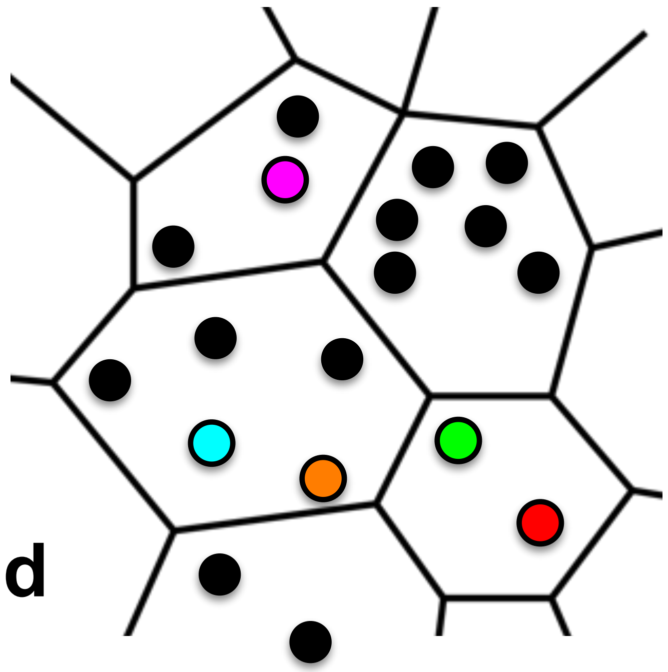
- Reduce quantization artifacts

Integration 3D-to-2D Matching

- Reduce quantization artifacts
- Reuse existing data structures

Integration 3D-to-2D Matching

- Reduce quantization artifacts
- Reuse existing data structures

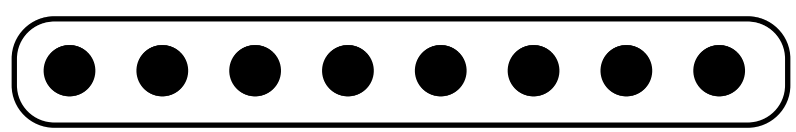


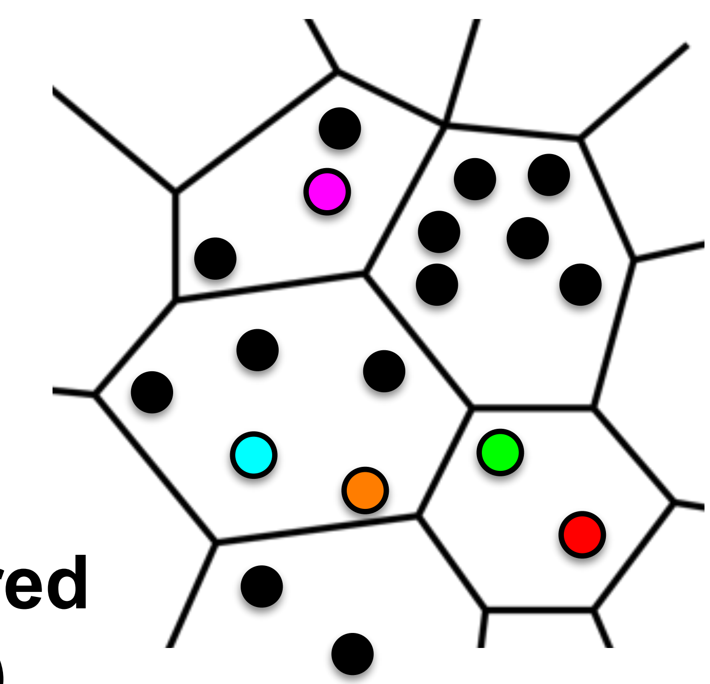
Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

- Reduce quantization artifacts
- Reuse existing data structures

Vocabulary Tree



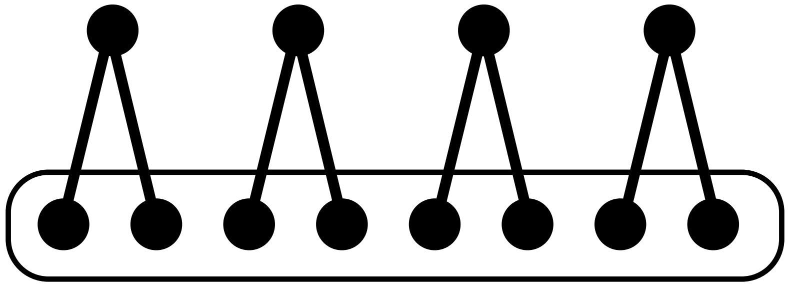


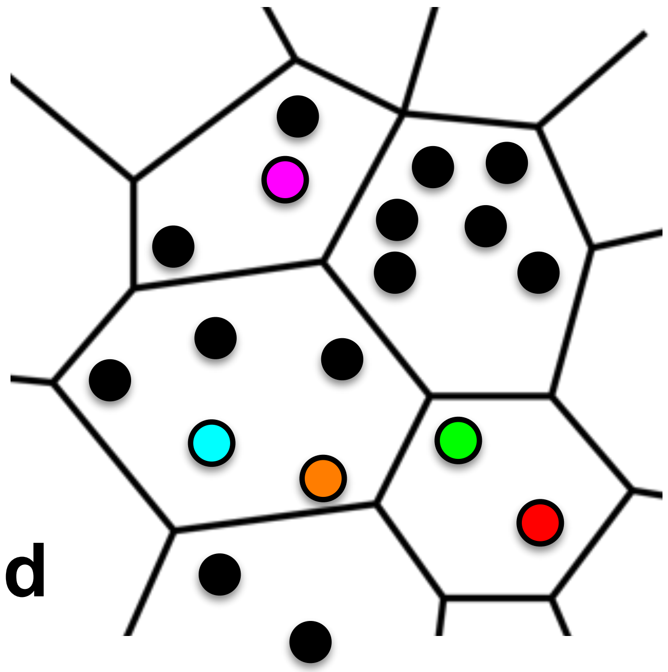
Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

- Reduce quantization artifacts
- Reuse existing data structures

Vocabulary Tree



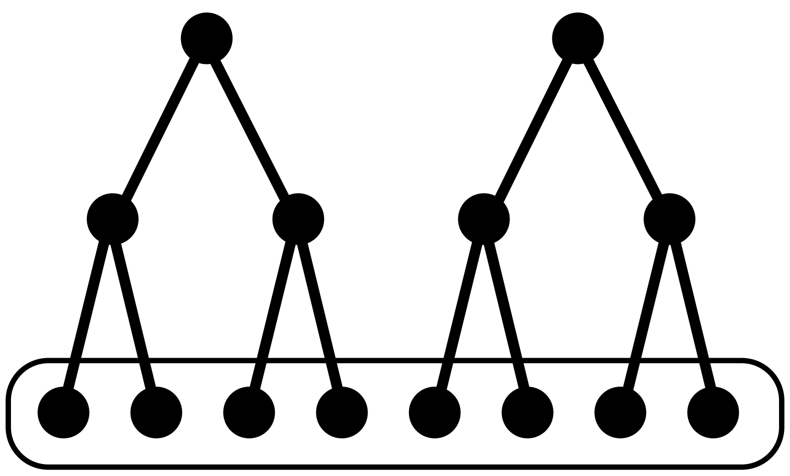


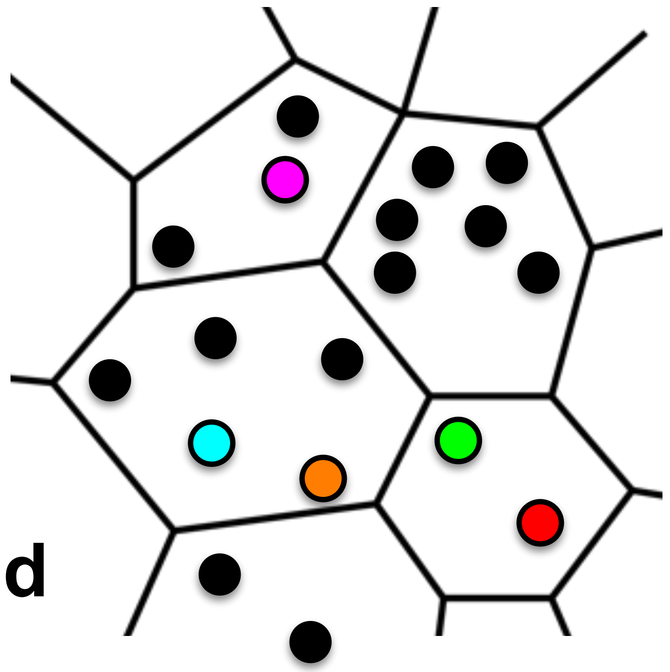
Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

- Reduce quantization artifacts
- Reuse existing data structures

Vocabulary Tree



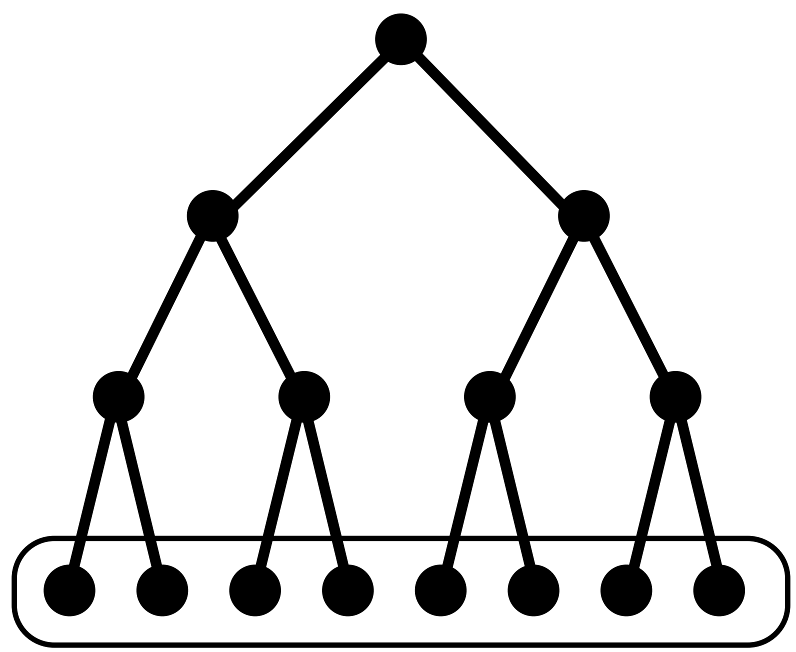


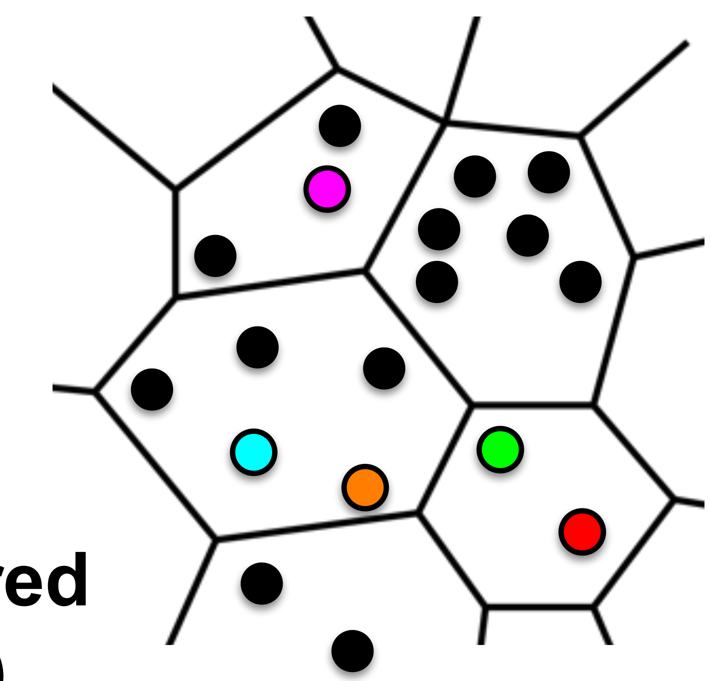
Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

- Reduce quantization artifacts
- Reuse existing data structures

Vocabulary Tree



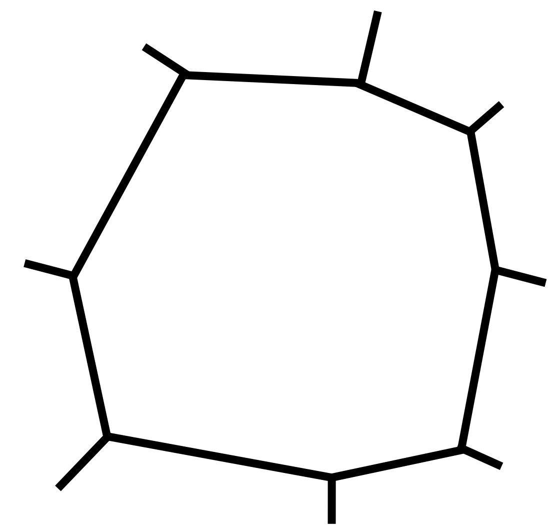


Large Vocabulary required for VPS (100k words)

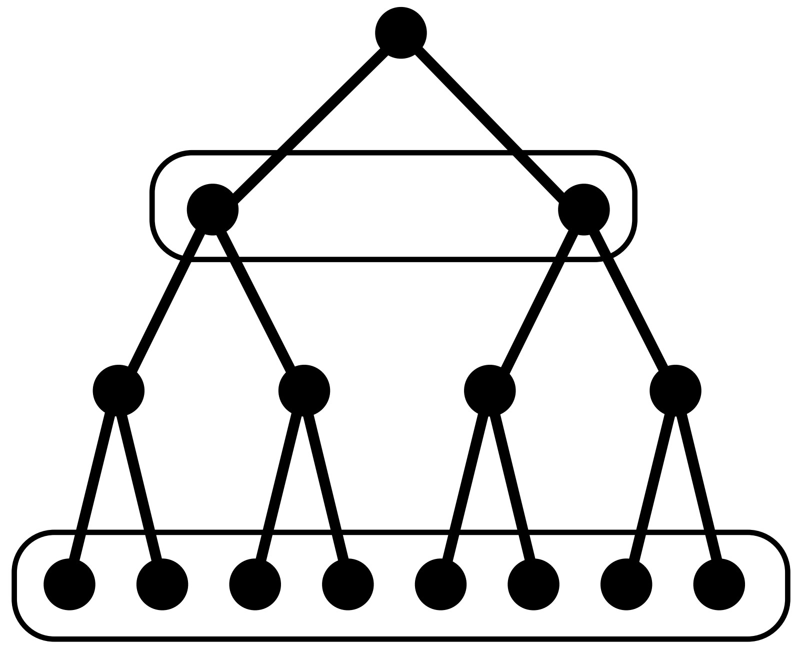
Integration 3D-to-2D Matching

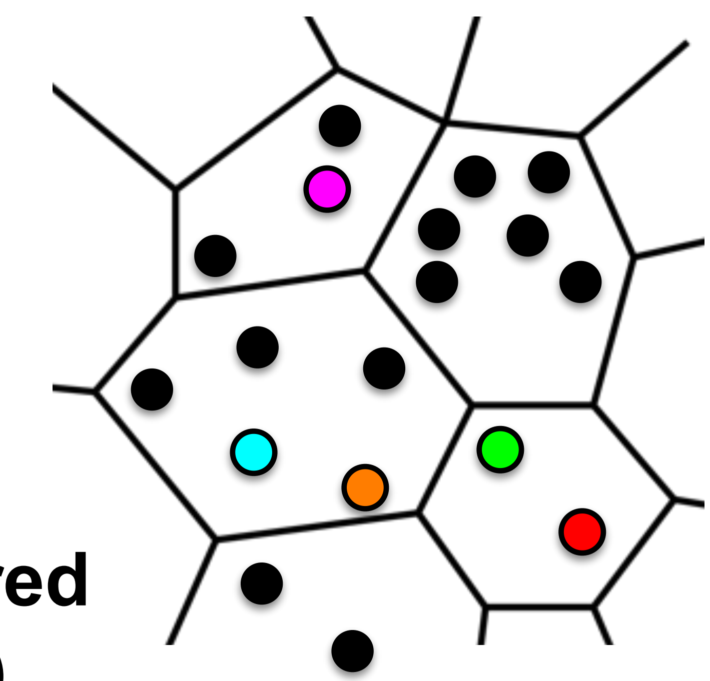
- Reduce quantization artifacts
- Reuse existing data structures

Small Vocabulary for 3D-to-2D search (100-1k words)



Vocabulary Tree



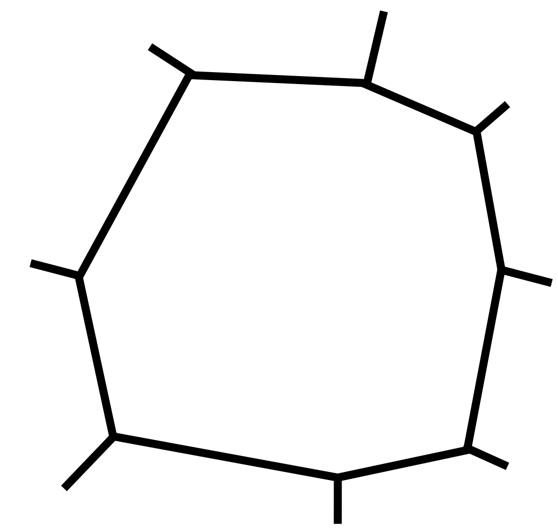


Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

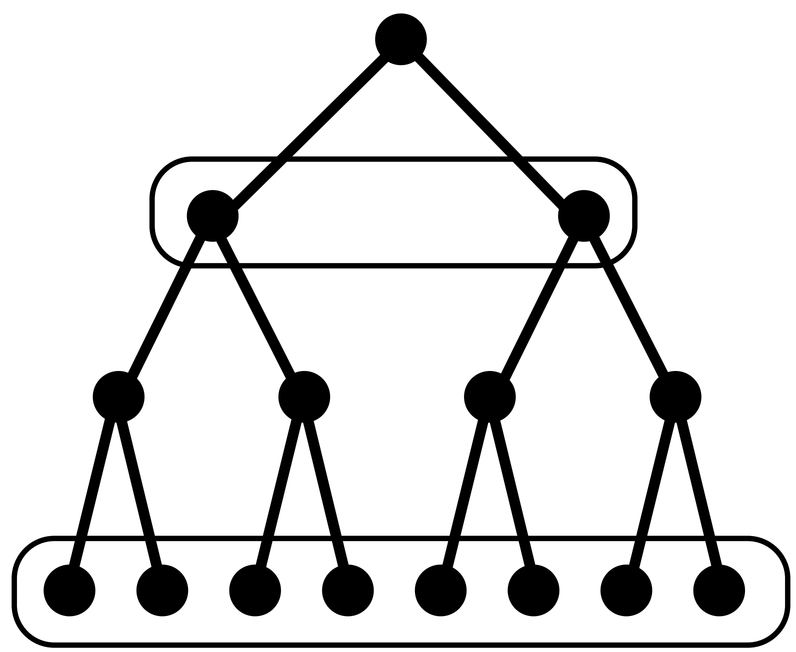
- Reduce quantization artifacts
- Reuse existing data structures

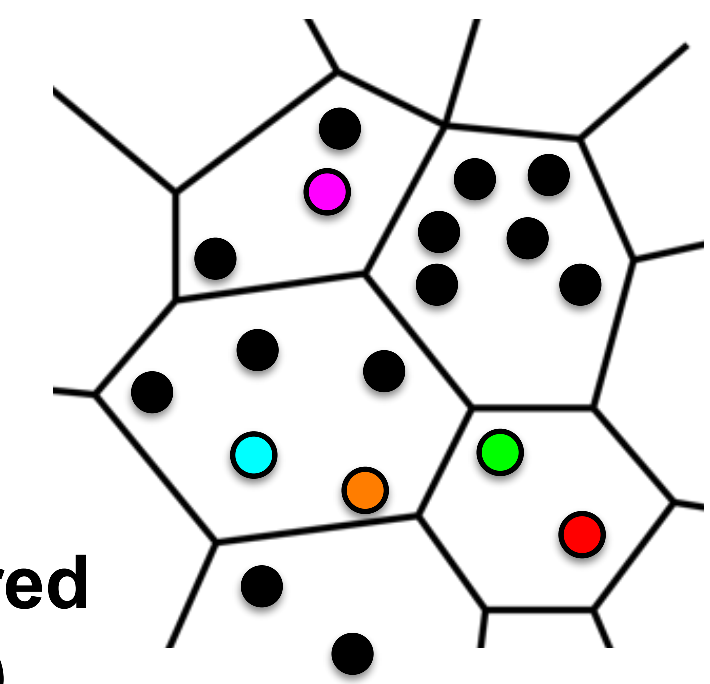
Small Vocabulary for 3D-to-2D search (100-1k words)



Query Image

Vocabulary Tree





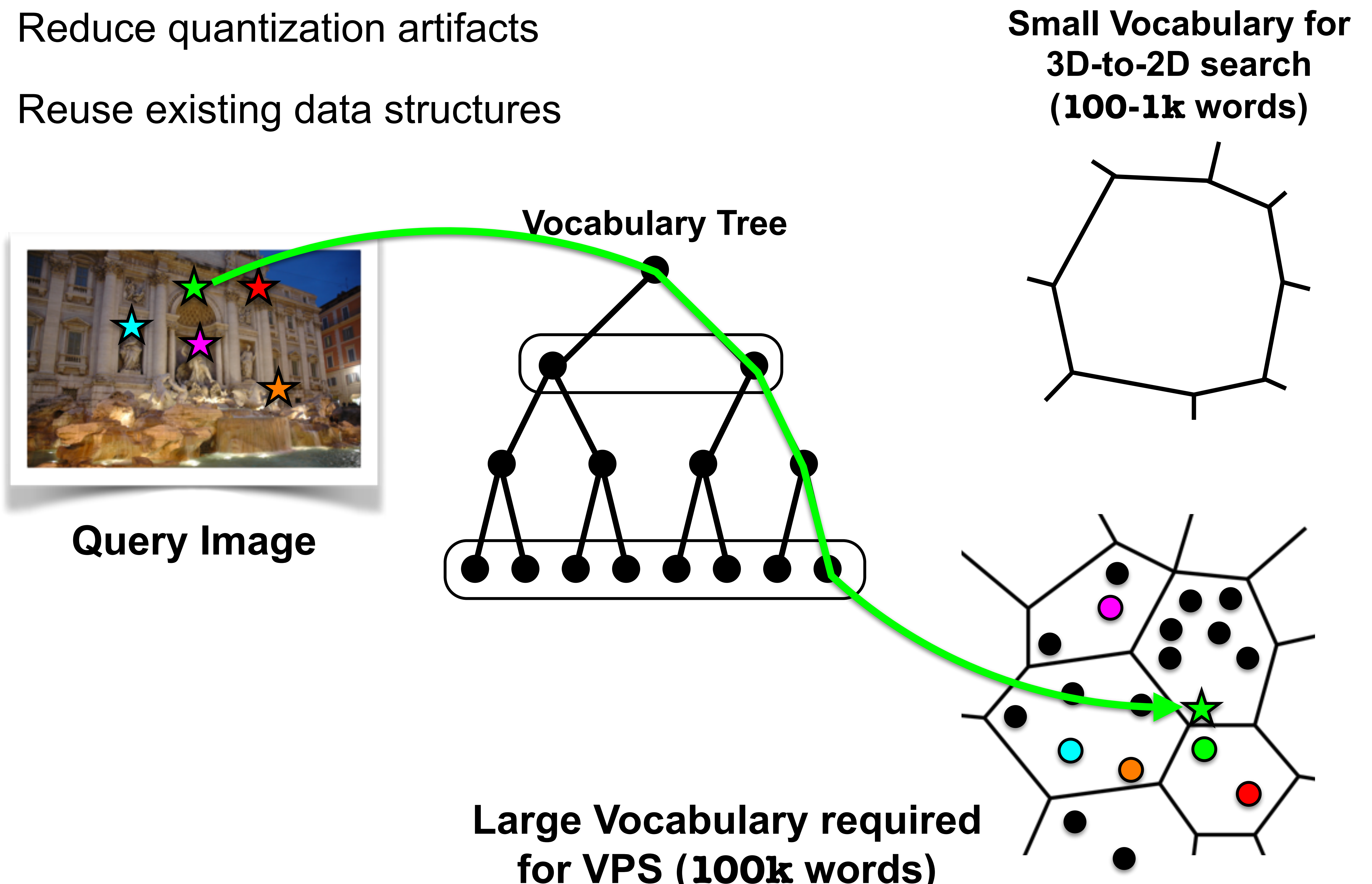
Large Vocabulary required for VPS (100k words)

Computer Vision and Geometry Group

ETH zürich

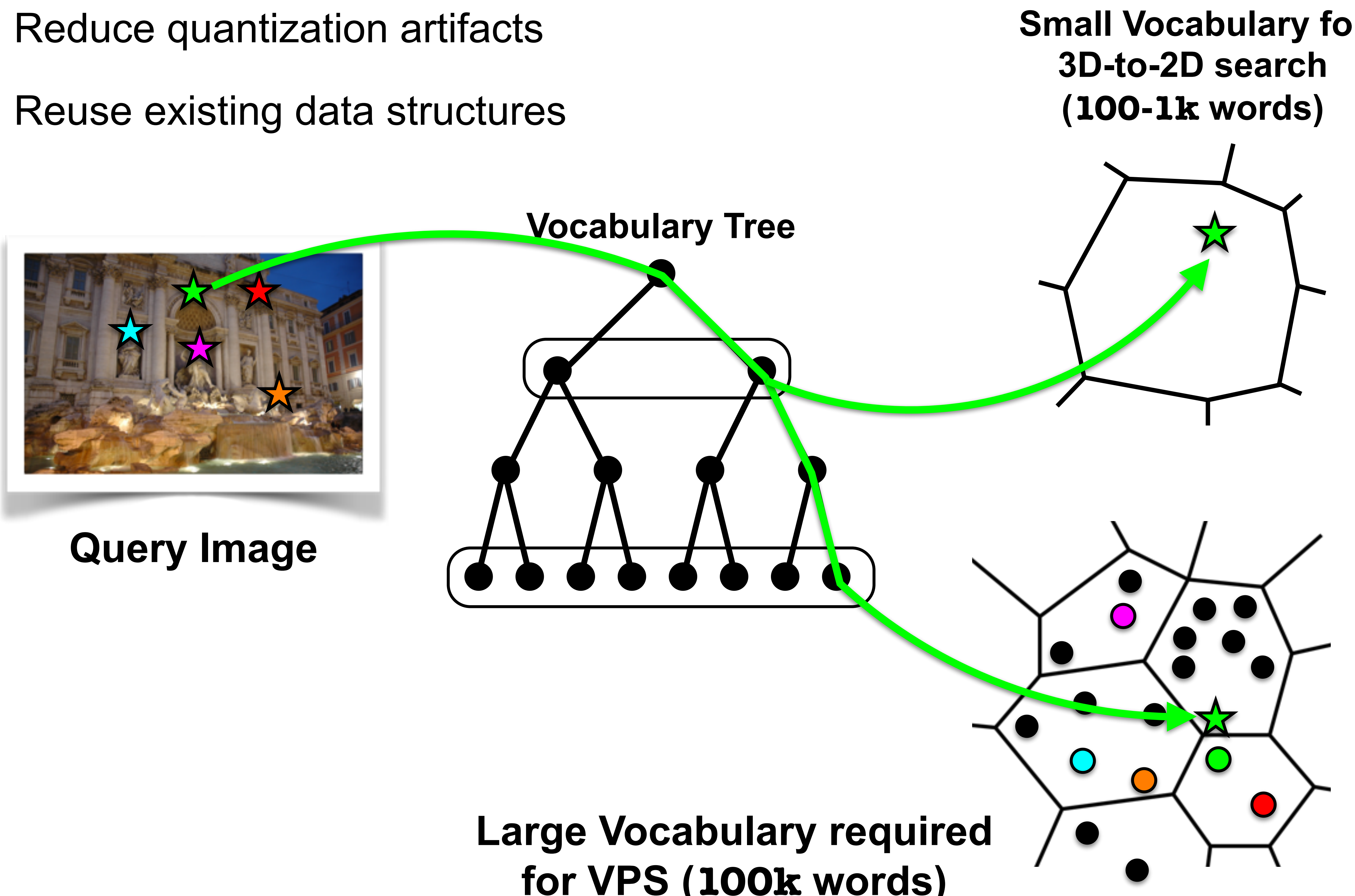
Integration 3D-to-2D Matching

- Reduce quantization artifacts
- Reuse existing data structures



Integration 3D-to-2D Matching

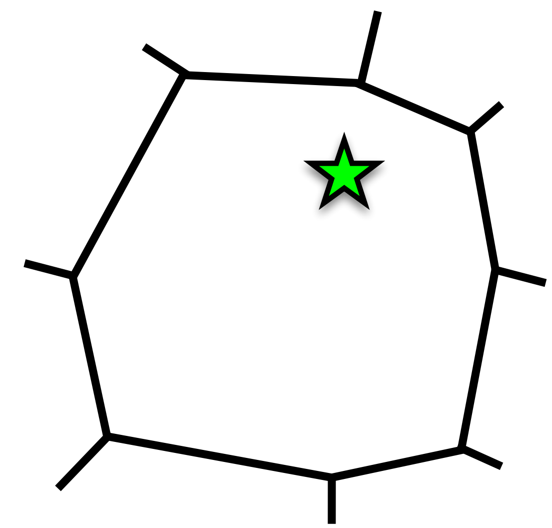
- Reduce quantization artifacts
- Reuse existing data structures



Integration 3D-to-2D Matching

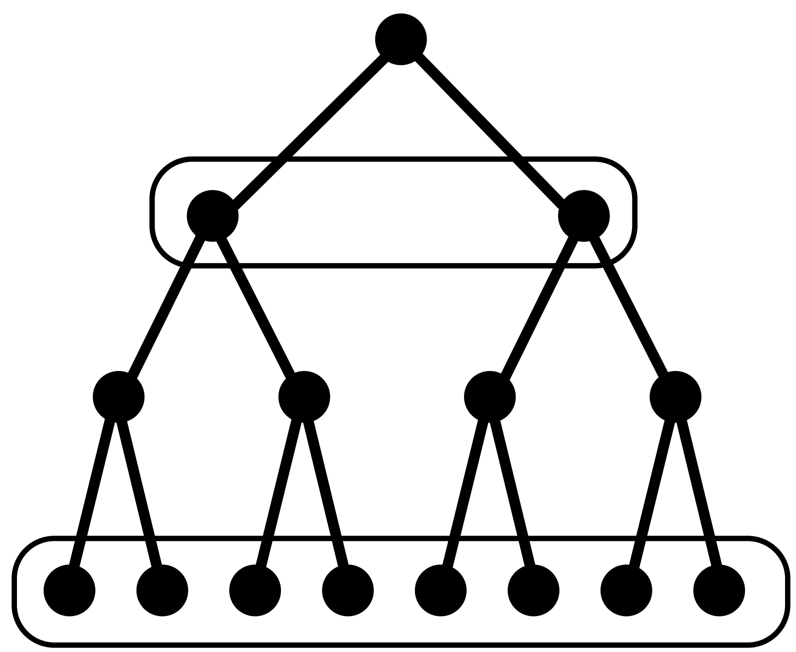
- Reduce quantization artifacts
- Reuse existing data structures

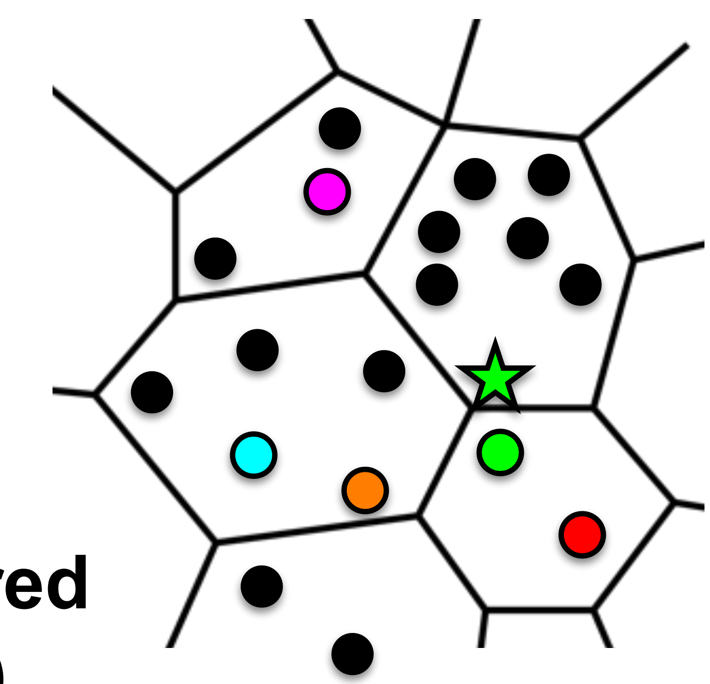
Small Vocabulary for 3D-to-2D search (100-1k words)



Query Image

Vocabulary Tree





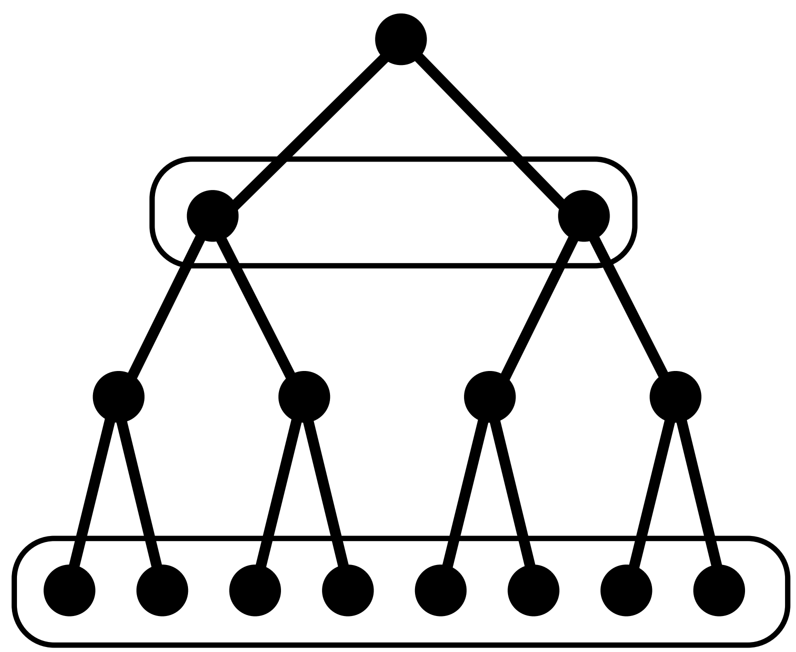
Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

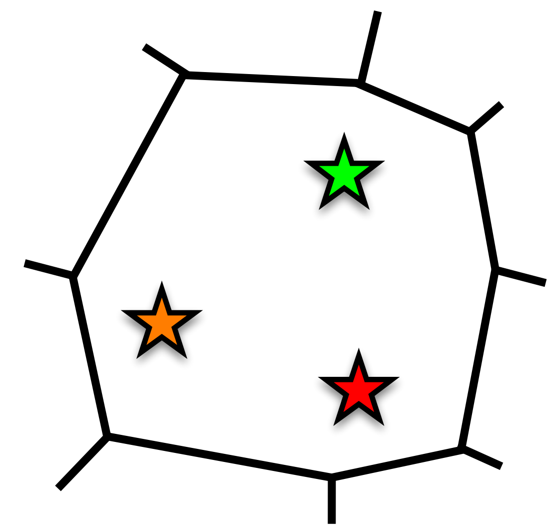
- Reduce quantization artifacts
- Reuse existing data structures

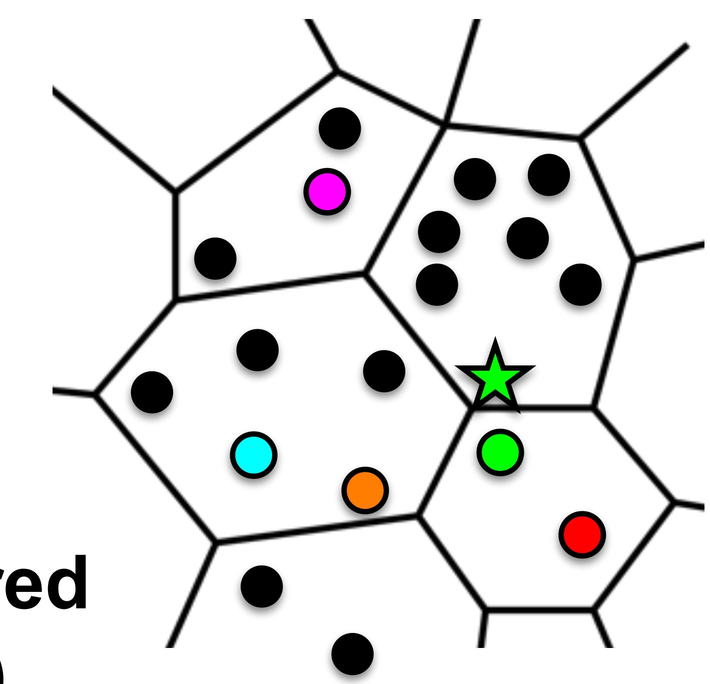
Query Image

Vocabulary Tree



Small Vocabulary for 3D-to-2D search (100-1k words)





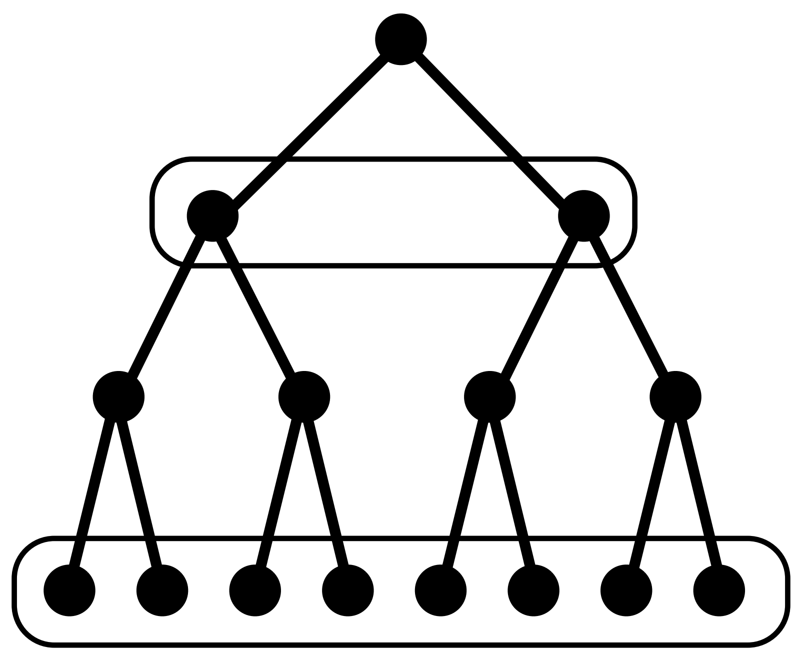
Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

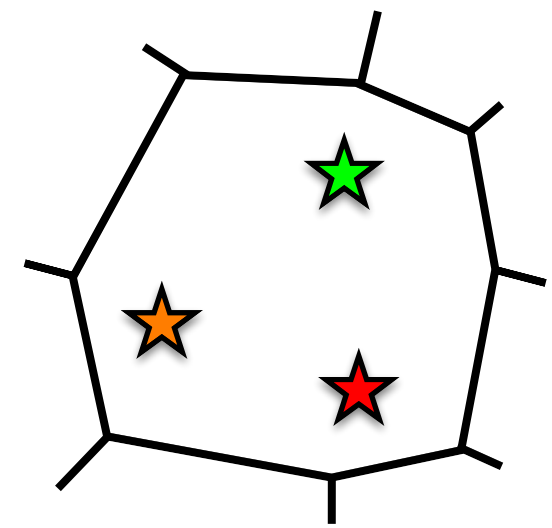
- Reduce quantization artifacts
- Reuse existing data structures

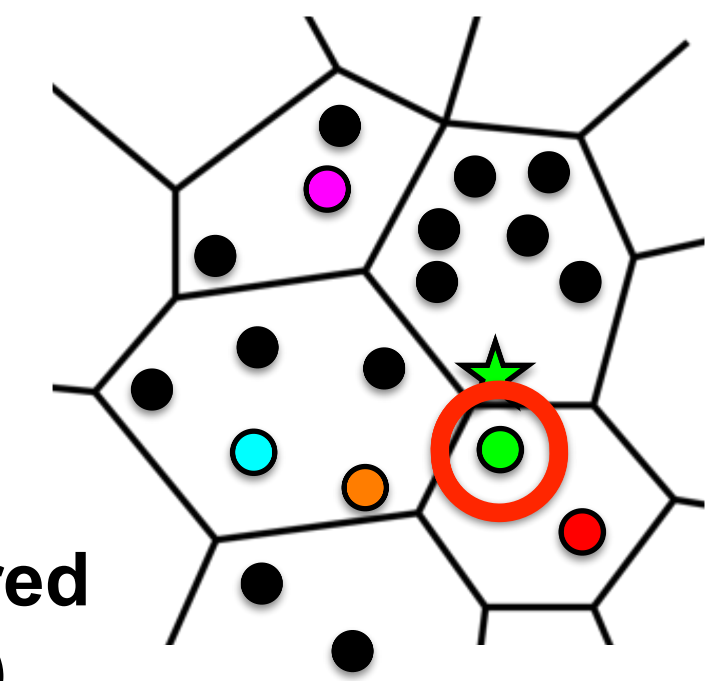
Query Image

Vocabulary Tree



Small Vocabulary for 3D-to-2D search (100-1k words)



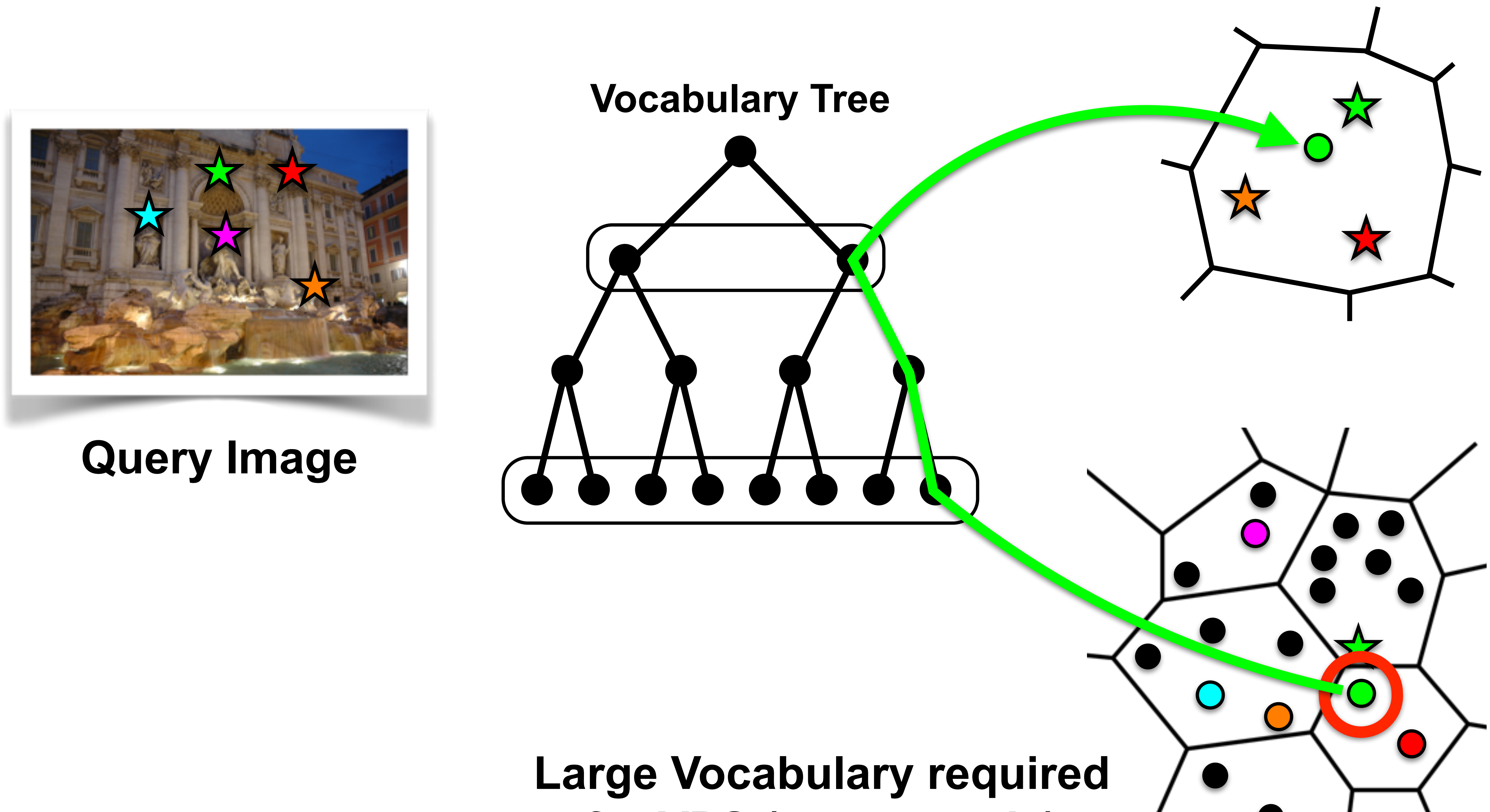


Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

- Reduce quantization artifacts
- Reuse existing data structures

Small Vocabulary for 3D-to-2D search (100-1k words)



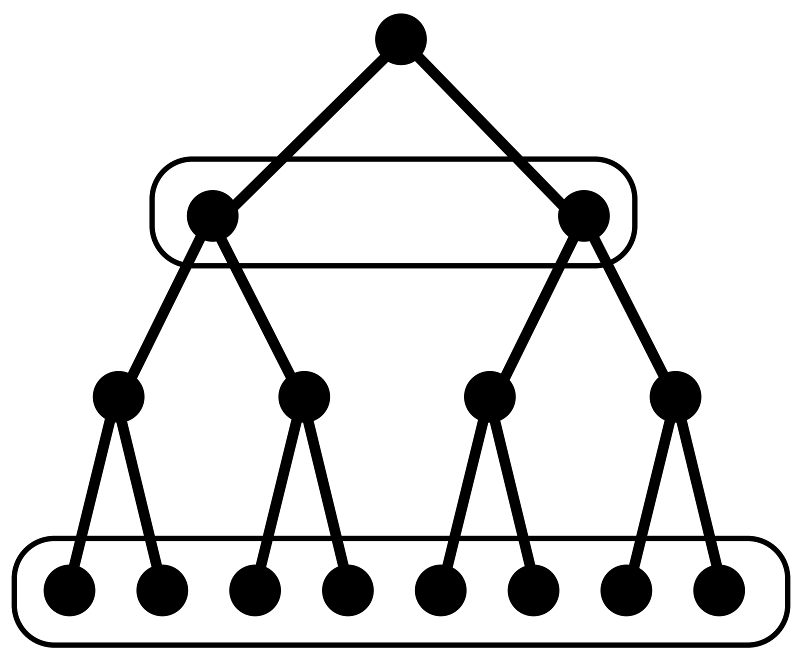
Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

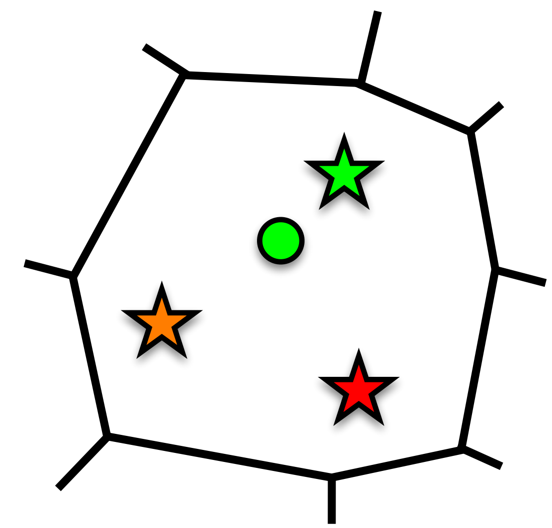
- Reduce quantization artifacts
- Reuse existing data structures

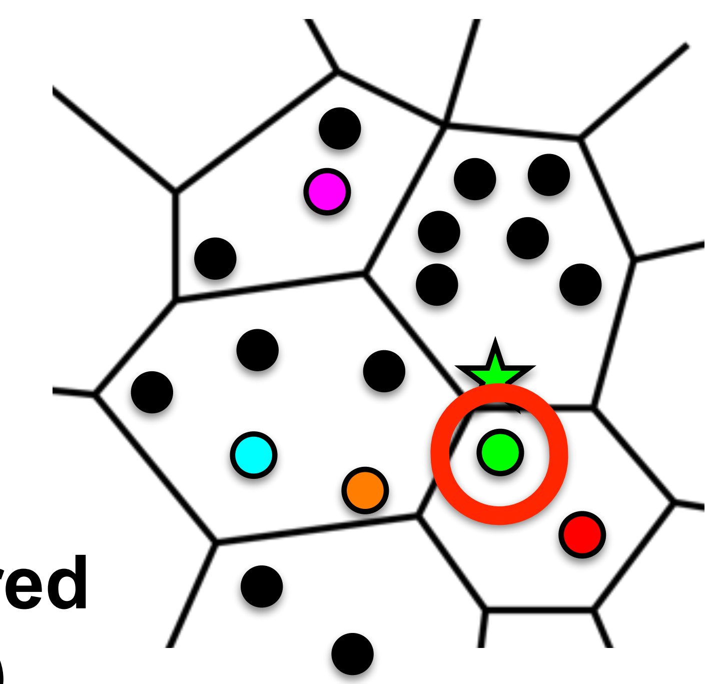
Query Image

Vocabulary Tree



Small Vocabulary for 3D-to-2D search (100-1k words)





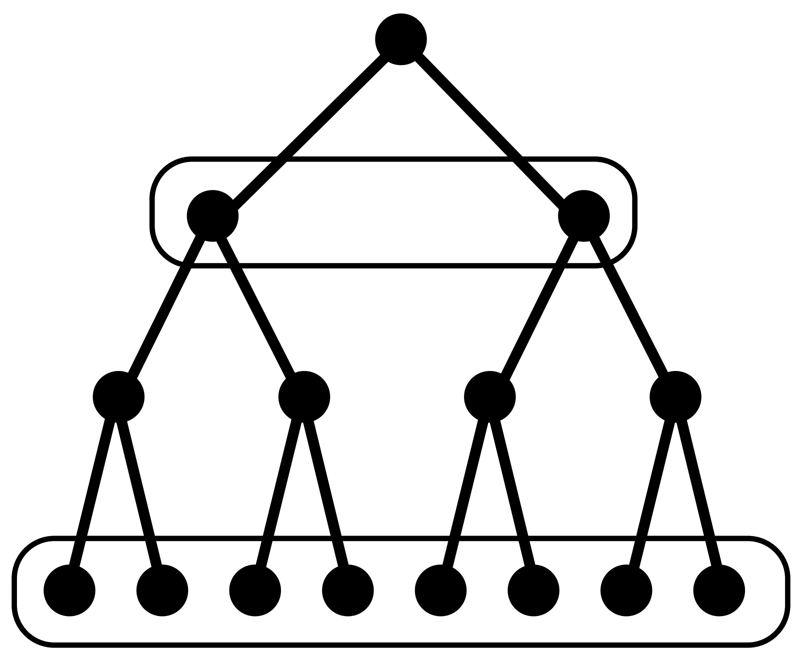
Large Vocabulary required for VPS (100k words)

Integration 3D-to-2D Matching

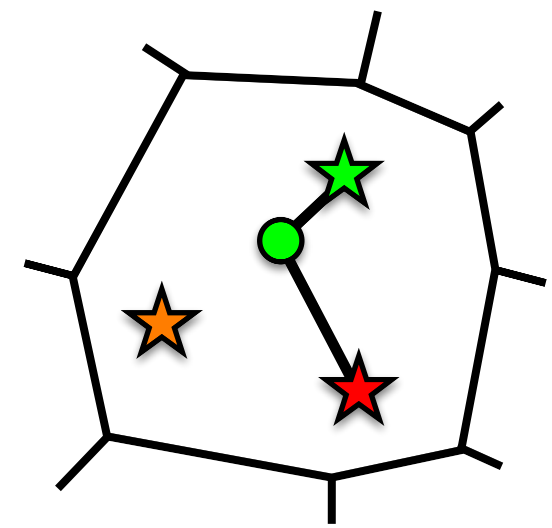
- Reduce quantization artifacts
- Reuse existing data structures

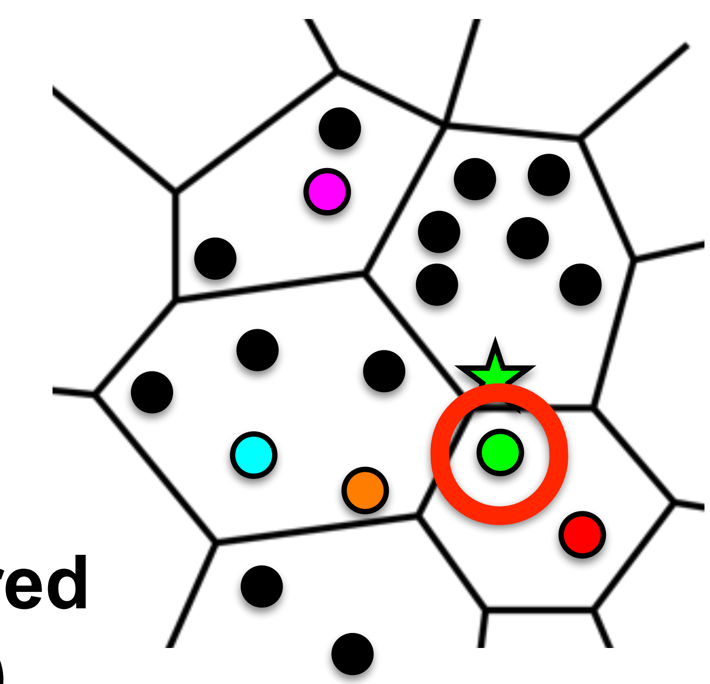
Query Image

Vocabulary Tree



Small Vocabulary for 3D-to-2D search (100-1k words)

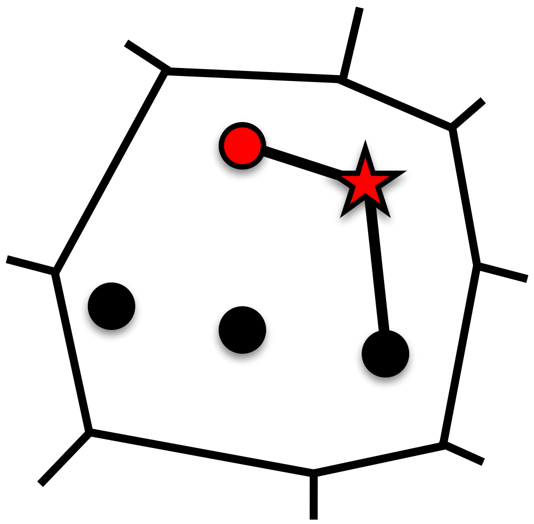




Large Vocabulary required for VPS (100k words)

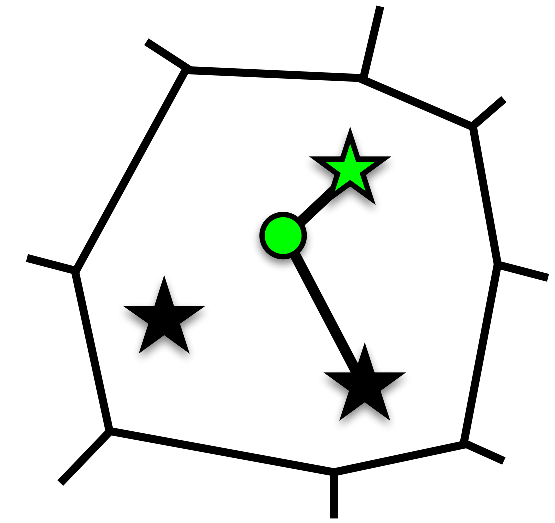
Common Prioritization Scheme

2D-to-3D Matching



- Find neighbors inside visual word
- Same definition of search costs

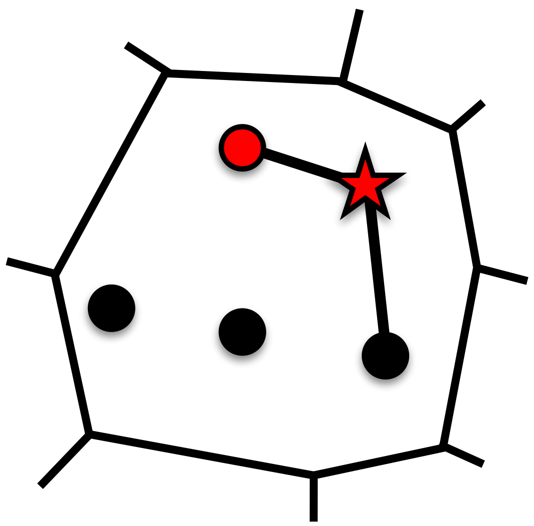
3D-to-2D Matching



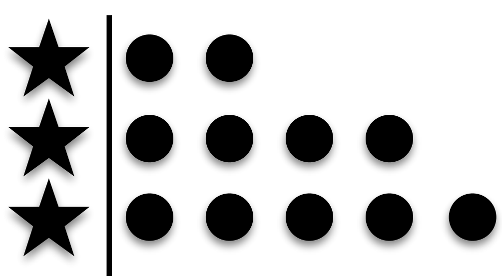
[Sattler et al., ECCV'12]

Common Prioritization Scheme

2D-to-3D Matching

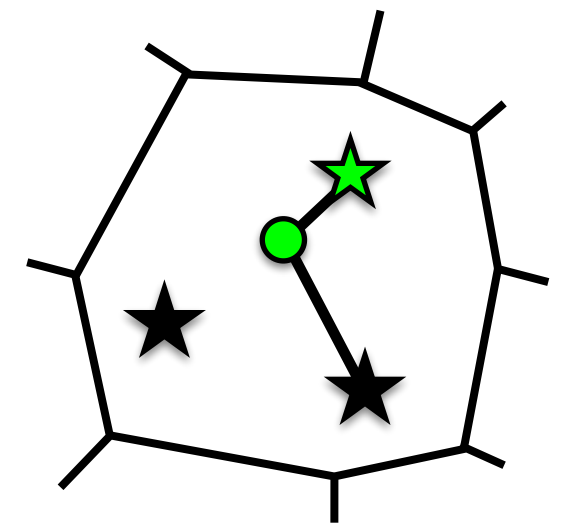


Priorities



- Find neighbors inside visual word
- Same definition of search costs

3D-to-2D Matching

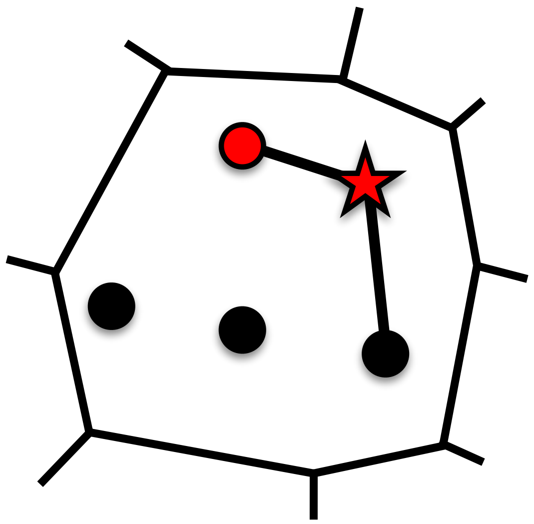


Priorities

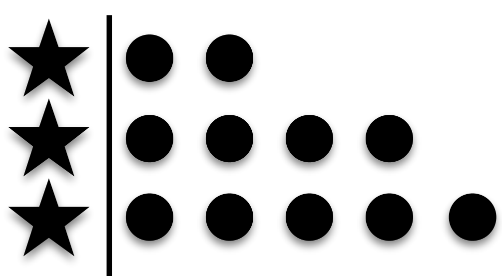
[Sattler et al., ECCV'12]

Common Prioritization Scheme

2D-to-3D Matching

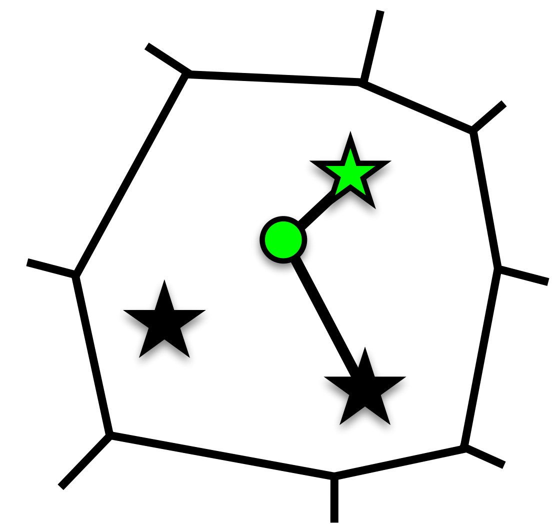


Priorities



- Find neighbors inside visual word
- Same definition of search costs

3D-to-2D Matching

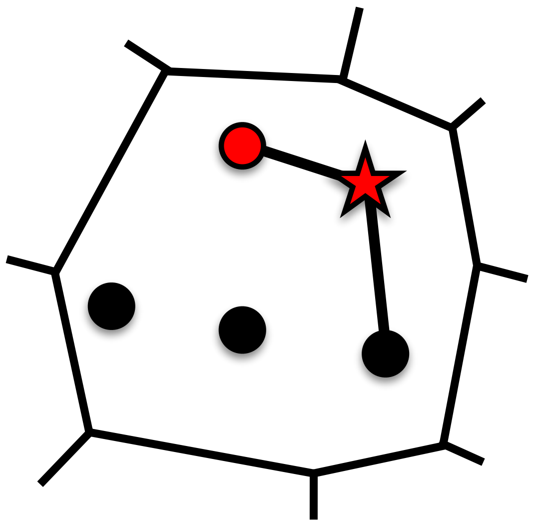


Priorities

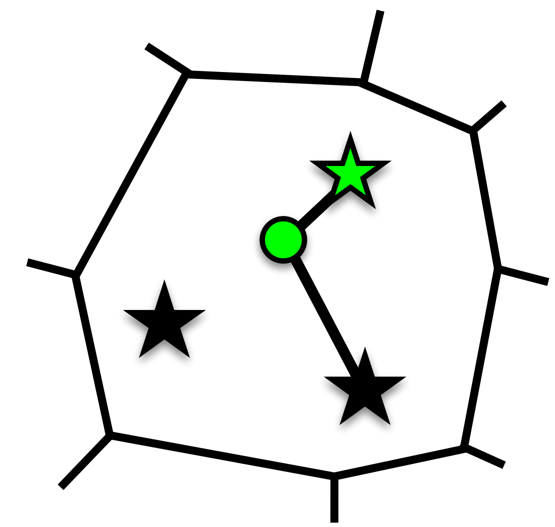
[Sattler et al., ECCV'12]

Common Prioritization Scheme

2D-to-3D Matching

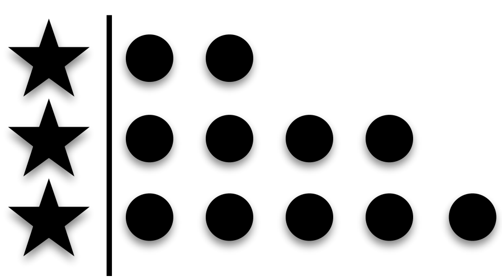


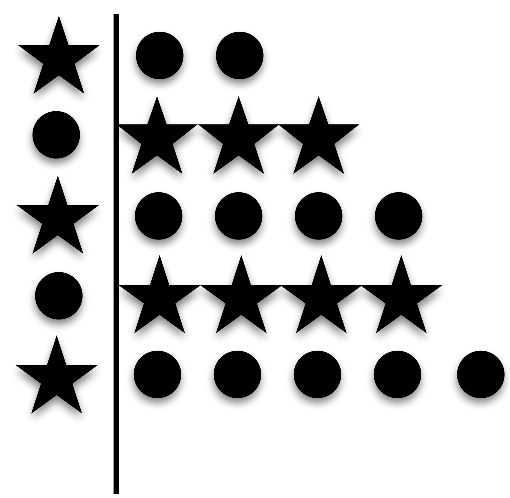
3D-to-2D Matching



- Find neighbors inside visual word
- Same definition of search costs

Priorities





Priorities

[Sattler et al., ECCV'12]

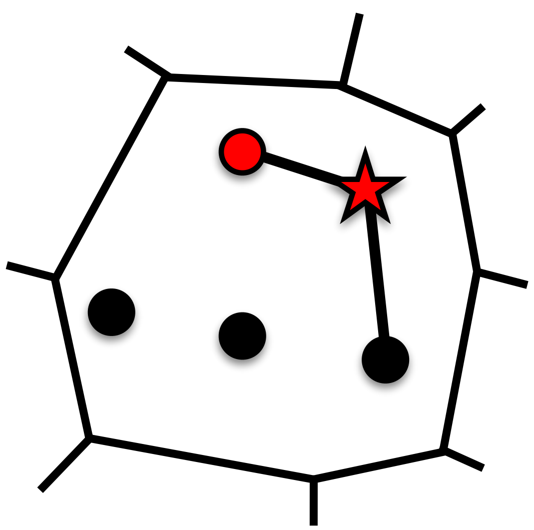
Computer Vision and Geometry Group

ETH zürich

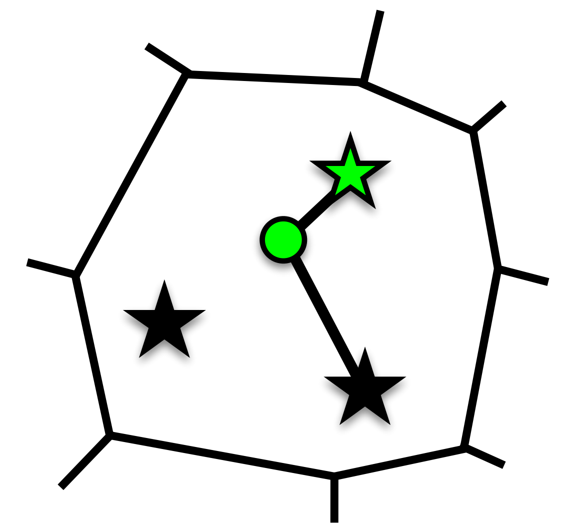
Common Prioritization Scheme

2D-to-3D Matching

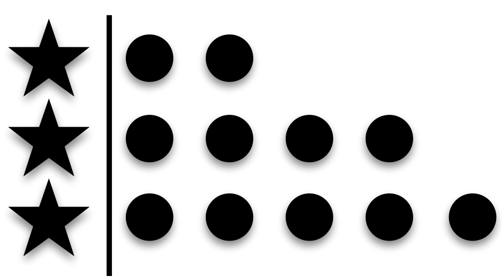
3D-to-2D Matching



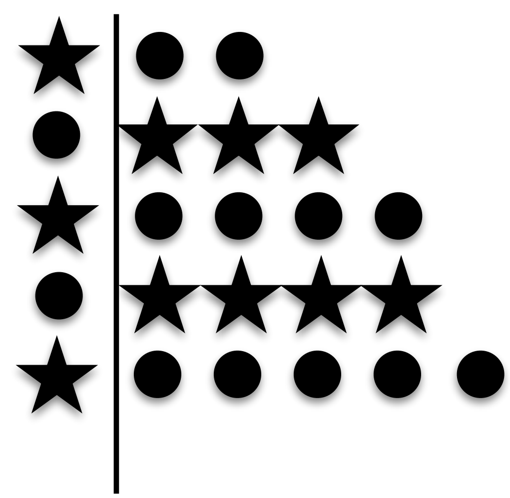
- Find neighbors inside visual word
- Same definition of search costs



Priorities



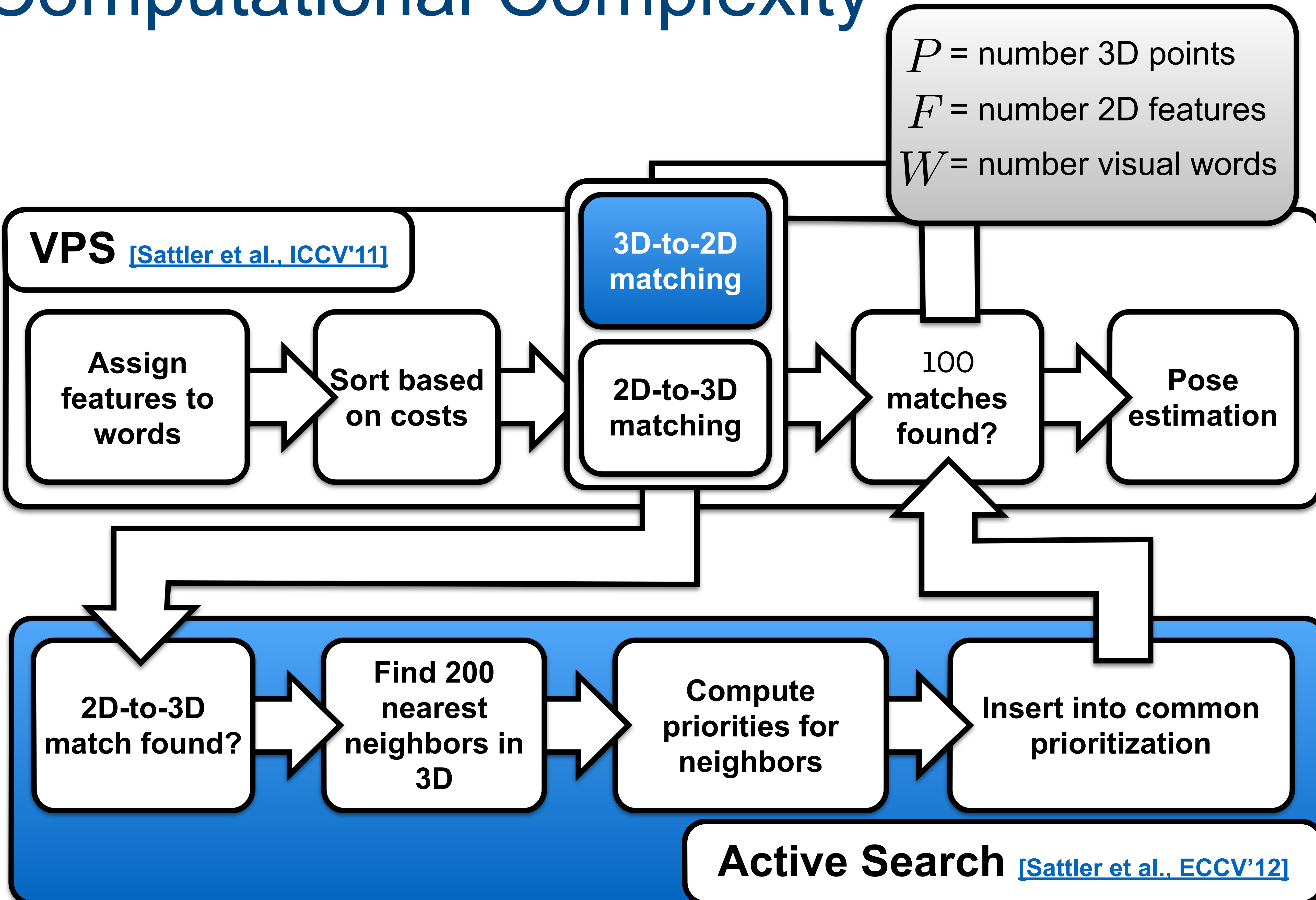
Priorities



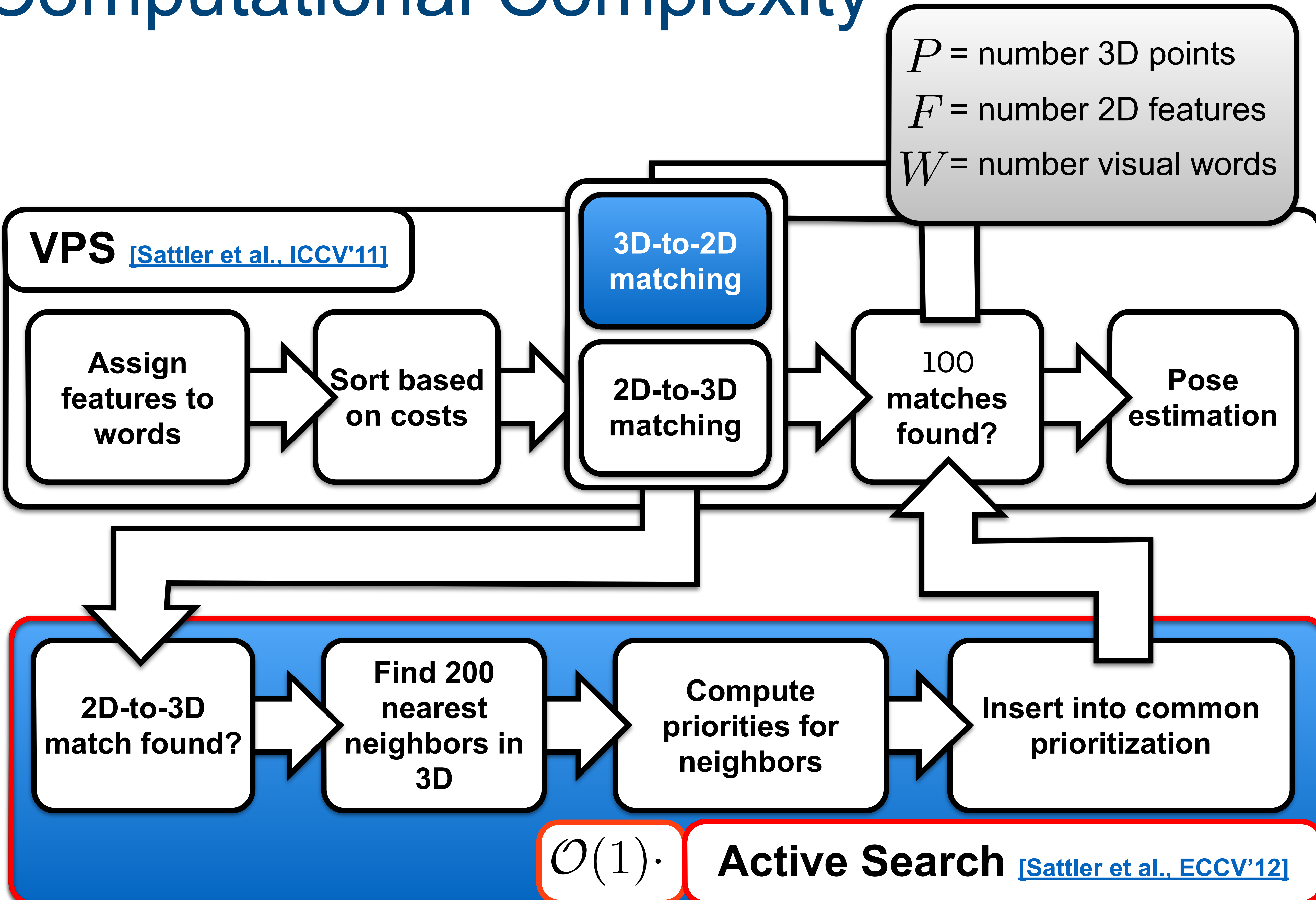
Common prioritization: Prefer cheaper search direction

[Sattler et al., ECCV'12]

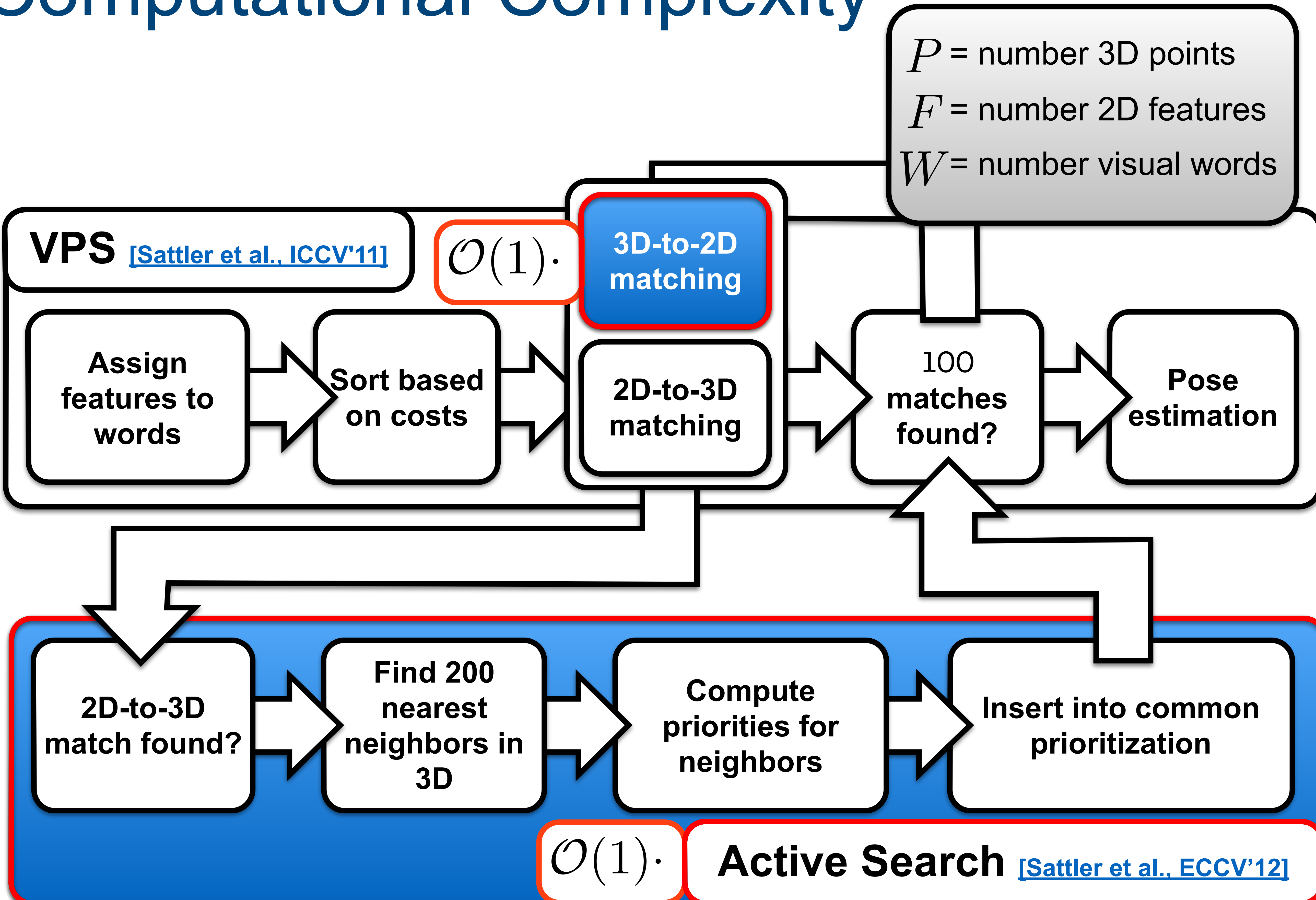
Computational Complexity



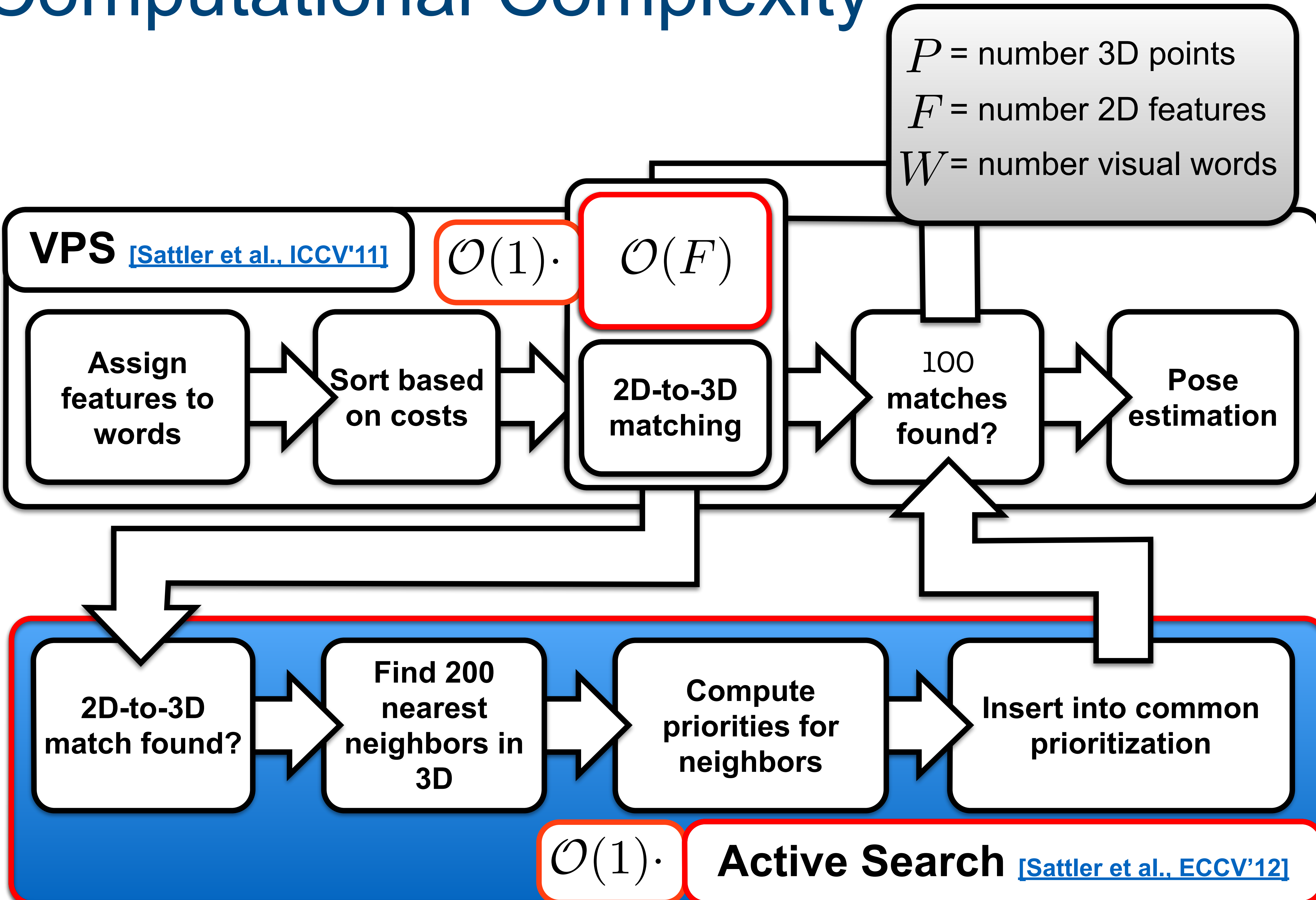
Computational Complexity



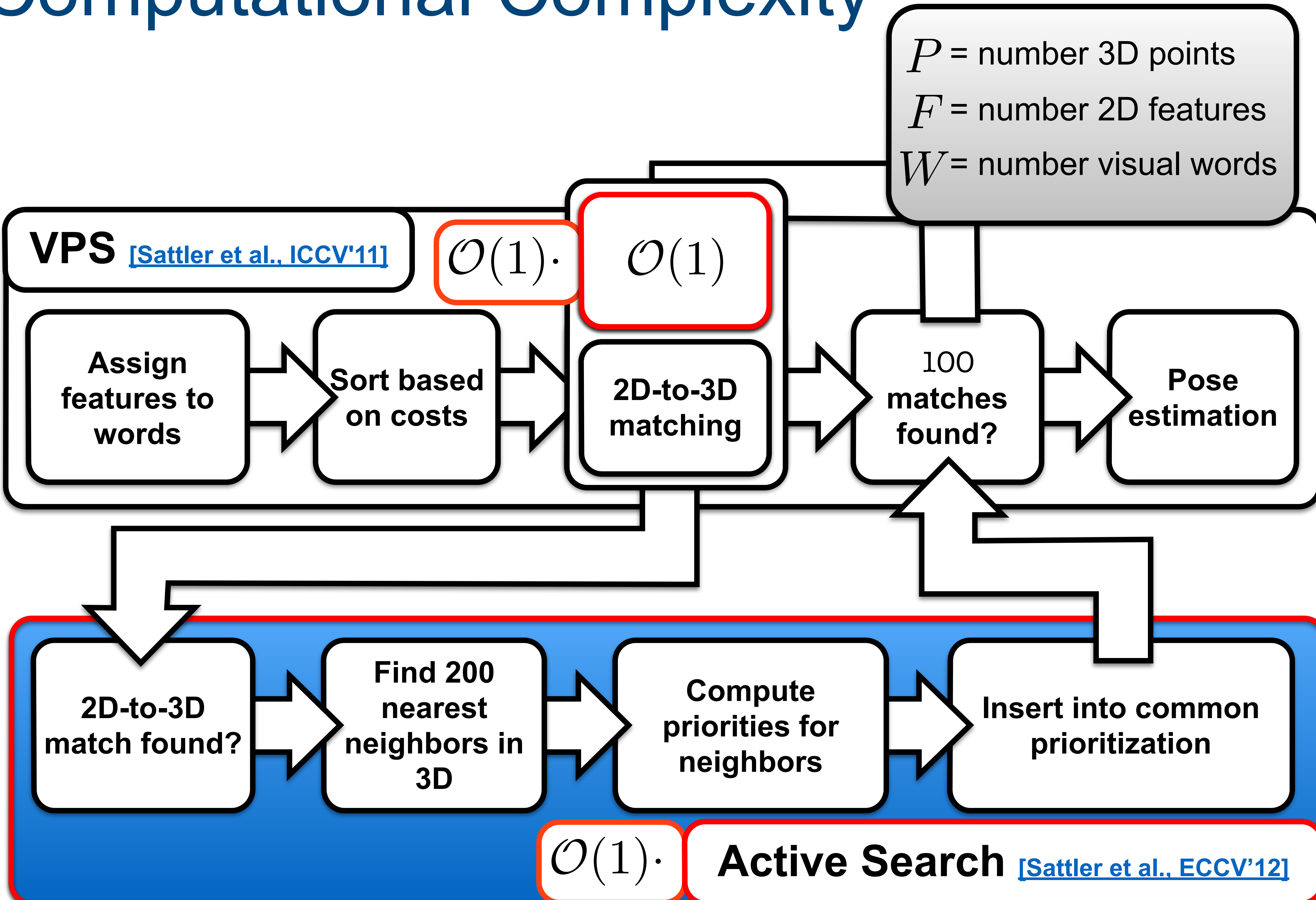
Computational Complexity



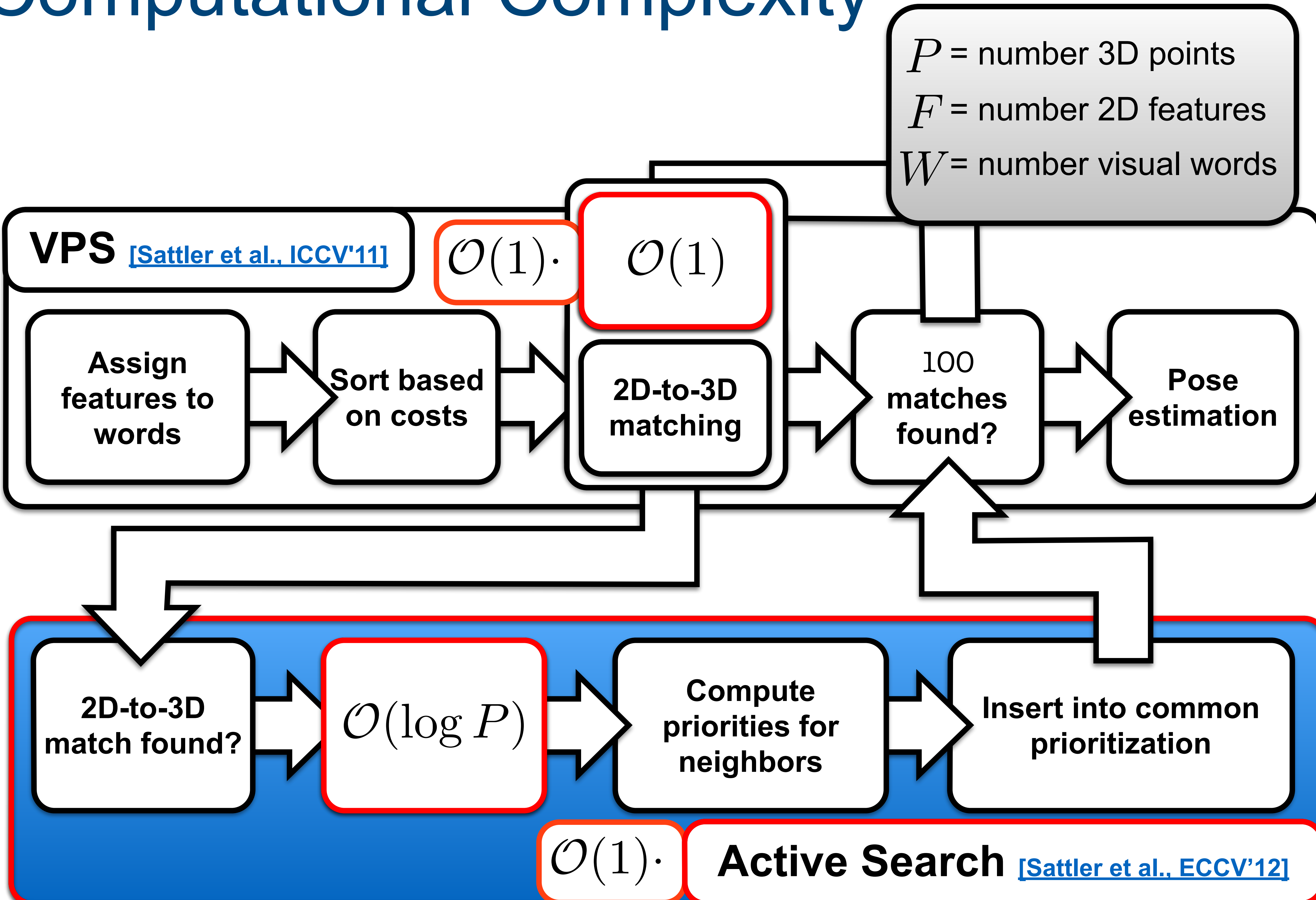
Computational Complexity



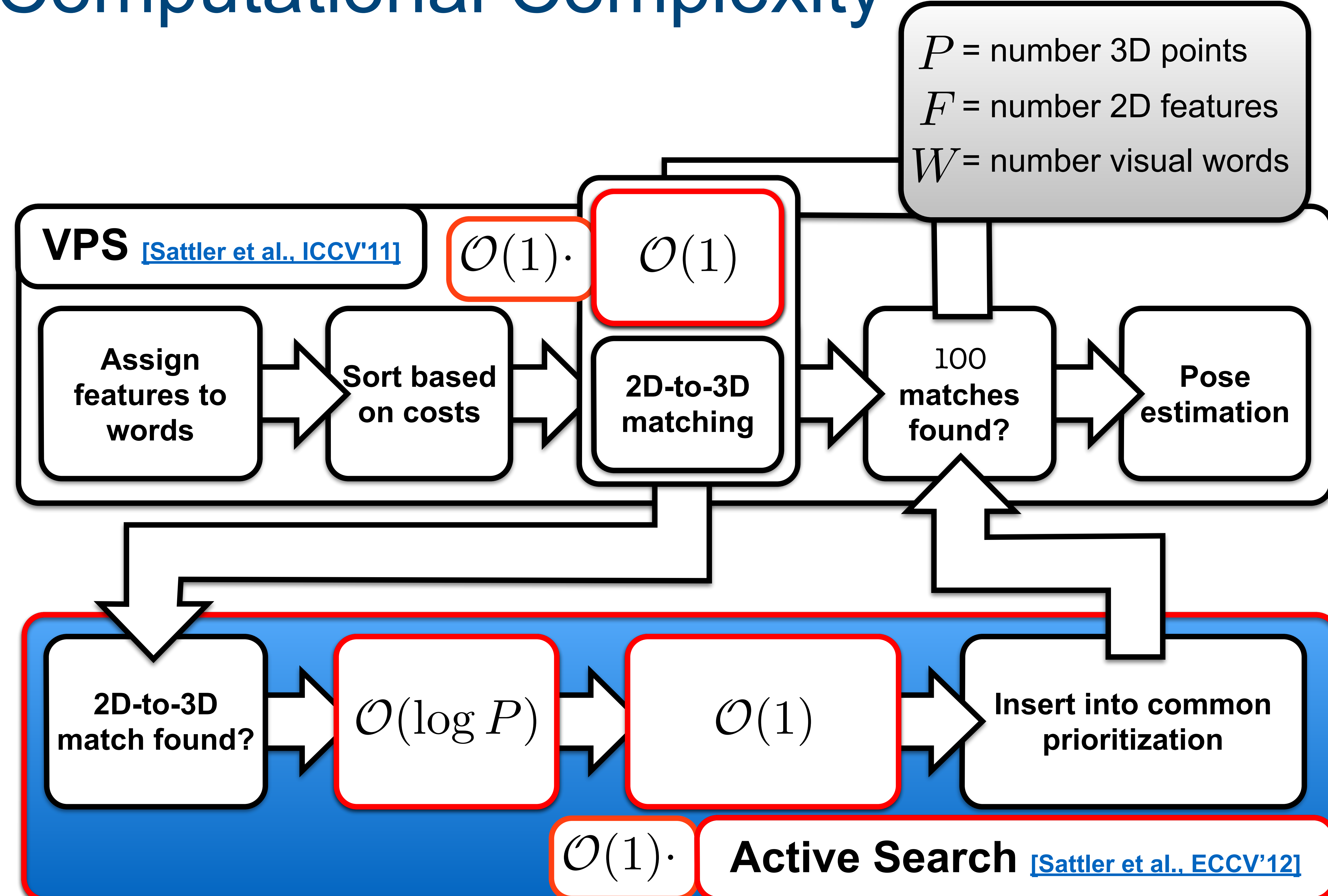
Computational Complexity



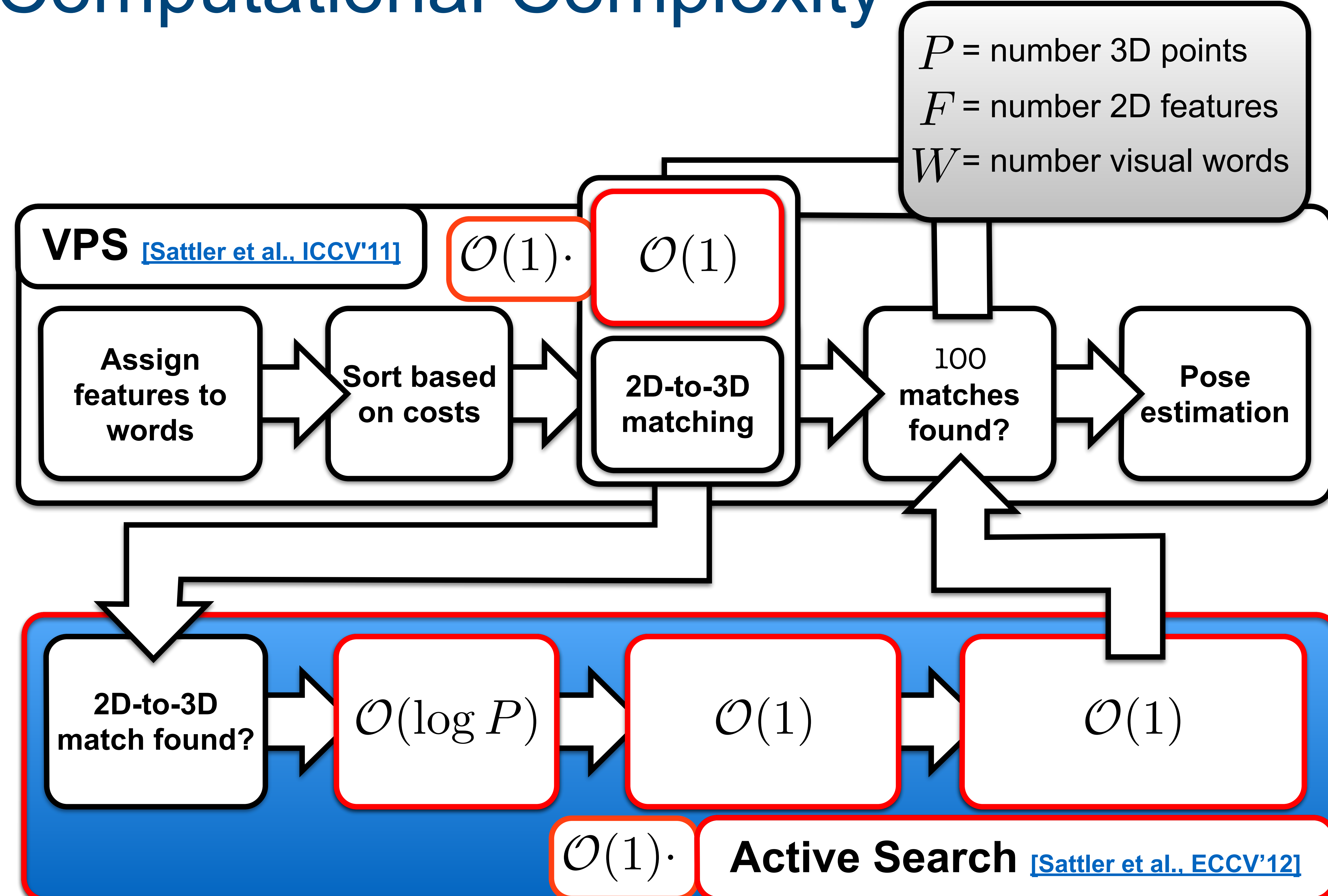
Computational Complexity



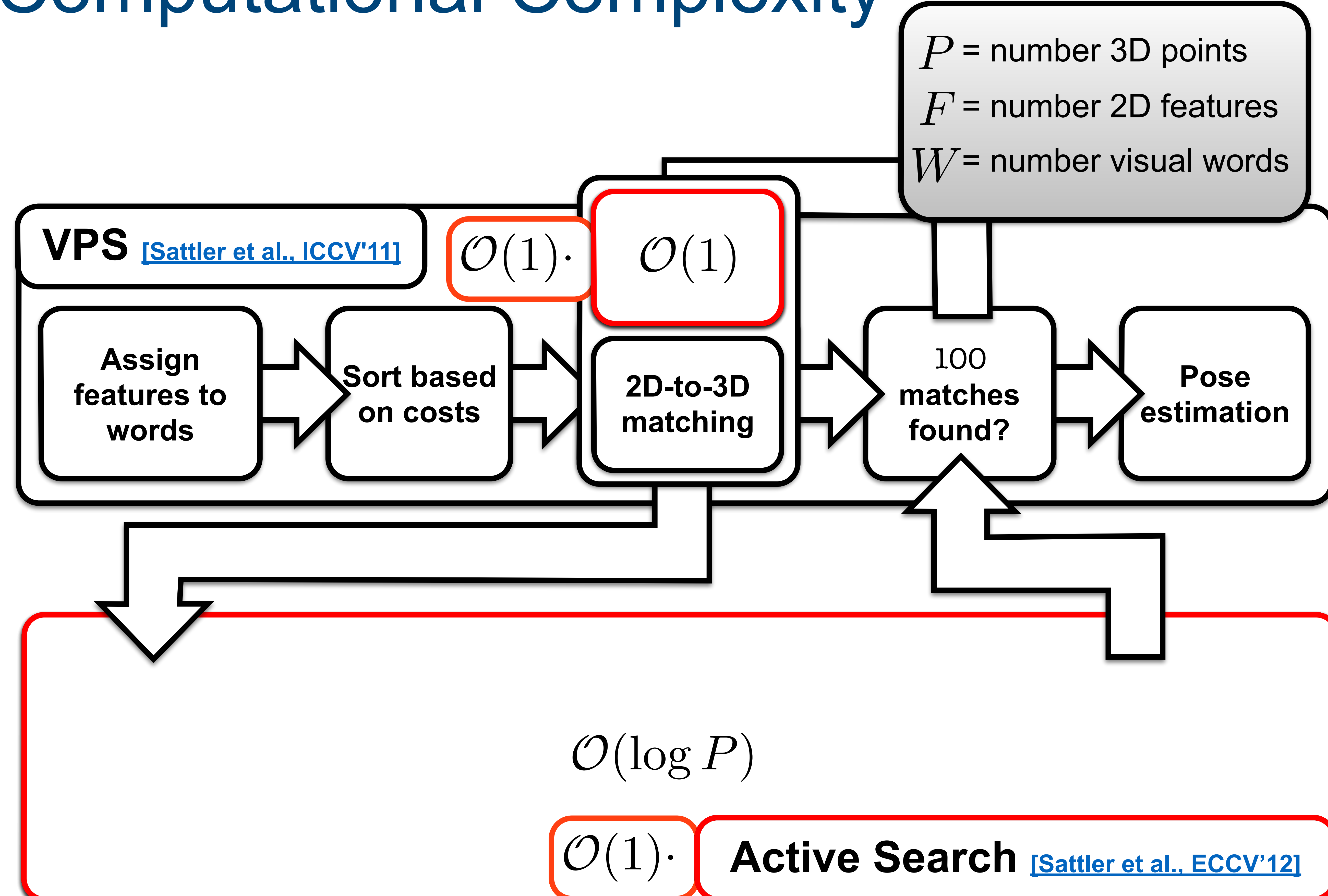
Computational Complexity



Computational Complexity



Computational Complexity



Computational Complexity

P = number 3D points

F = number 2D features

W = number visual words

VPS [Sattler et al., ICCV'11]

$\mathcal{O}(1)\cdot$ $\mathcal{O}(1)$

fe

on

Total effort added by Active Search: $\mathcal{O}(\log P)$

$\mathcal{O}(\log P)$

$\mathcal{O}(1)\cdot$ **Active Search** [Sattler et al., ECCV'12]

Computational Complexity

P = number 3D points
F = number 2D features
W = number visual words

VPS [Sattler et al., ICCV'11]

$\mathcal{O}(1)\cdot$ $\mathcal{O}(1)$

fe

on

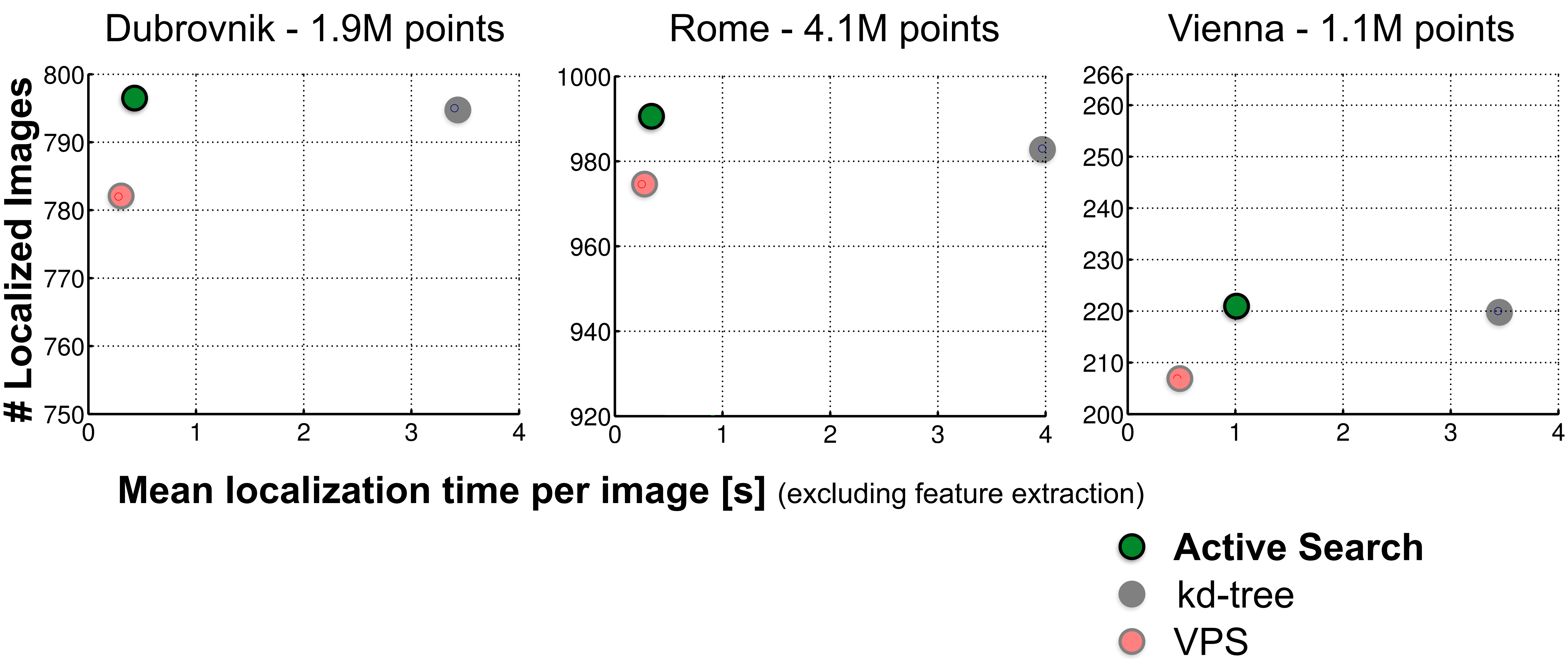
Total effort added by Active Search: $\mathcal{O}(\log P)$

Effort added by Soft Assignments: $\mathcal{O}(P/W)$ **per feature**

$\mathcal{O}(\log P)$

$\mathcal{O}(1)\cdot$ **Active Search** [Sattler et al., ECCV'12]

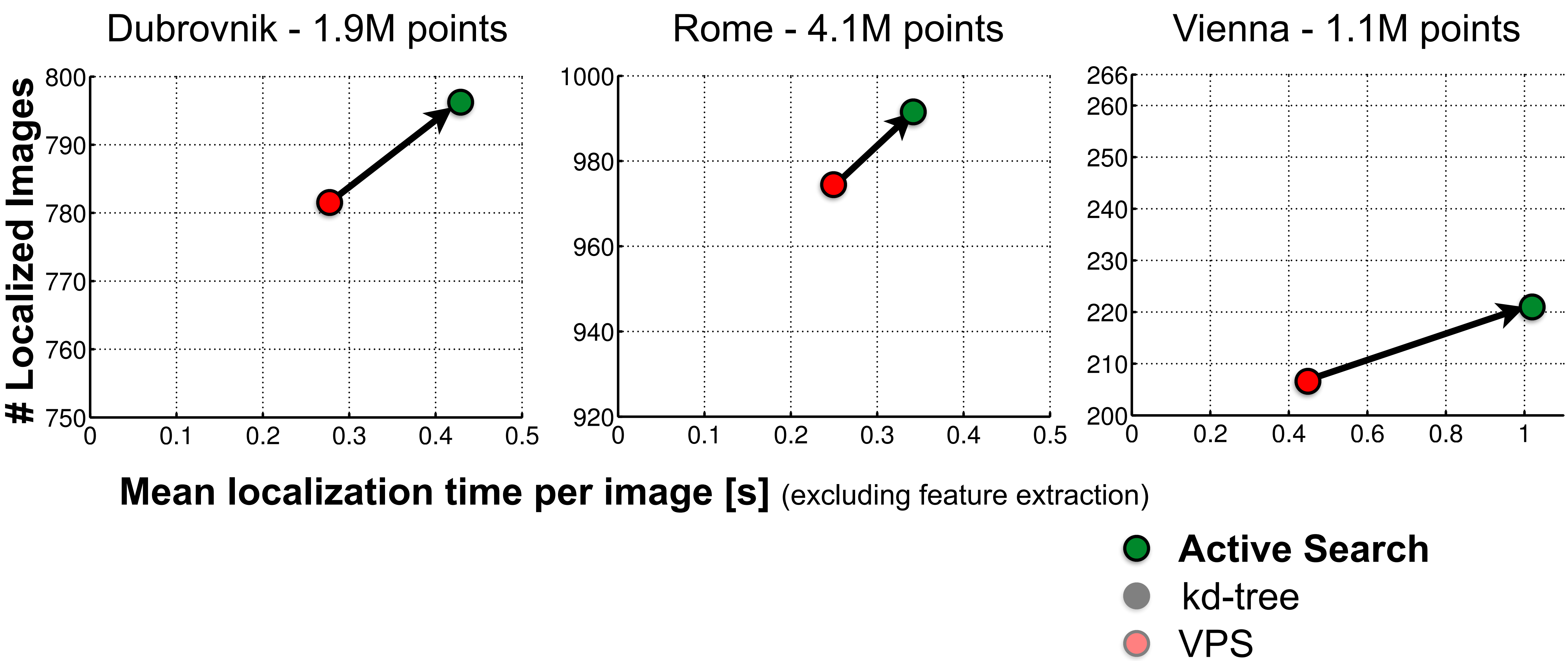
Results



✓ As effective as kd-tree or better

✗ Less effective than VPS due to additional computations

Results



✓ As effective as kd-tree or better

✗ Less effective than VPS due to additional computations

Localization - Overview

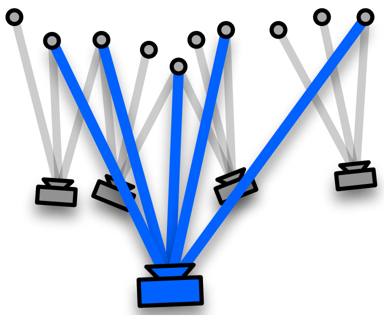
Baseline: kd-tree search

[Sattler et al., ICCV'11]

VPS

[Sattler et al., ICCV'11]

Active Search

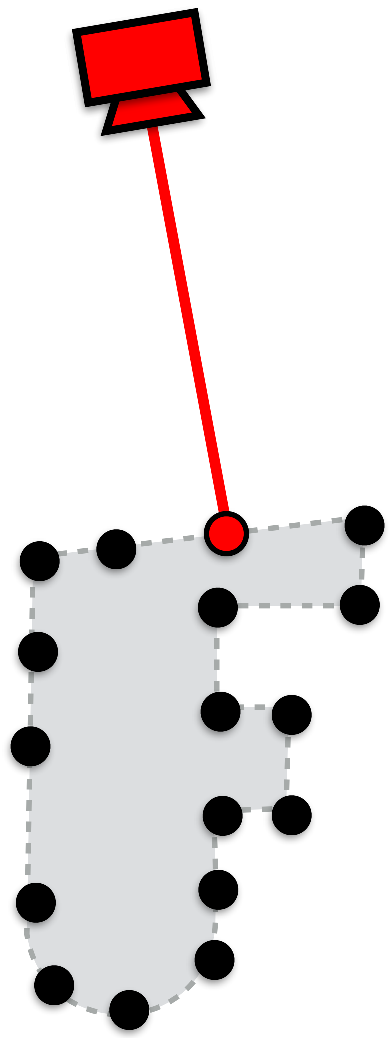


+ Visibility Filtering

[Sattler et al., ECCV'12]

	Baseline: kd-tree search	VPS	Active Search	+ Visibility Filtering
effectiveness	✓	✗	✓✓	✓✓
efficiency	✗ ✗	✓	✗	✓

Visibility Filtering

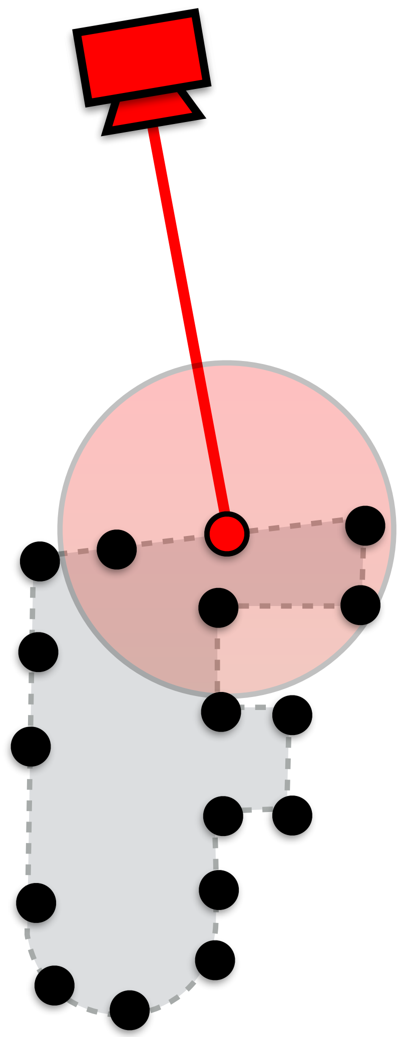


Filter out 3D-to-2D matching candidates

[Sattler et al., ECCV'12]

Computer Vision and Geometry Group ETH zürich

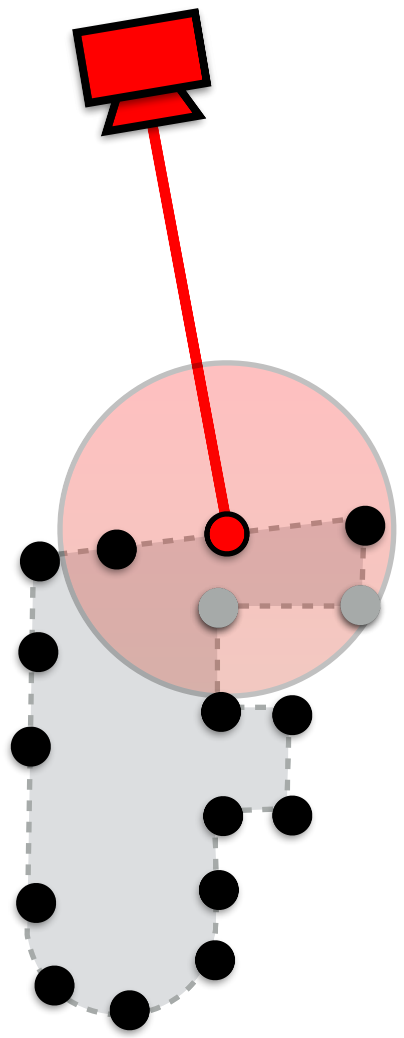
Visibility Filtering



Filter out 3D-to-2D matching candidates

[Sattler et al., ECCV'12]

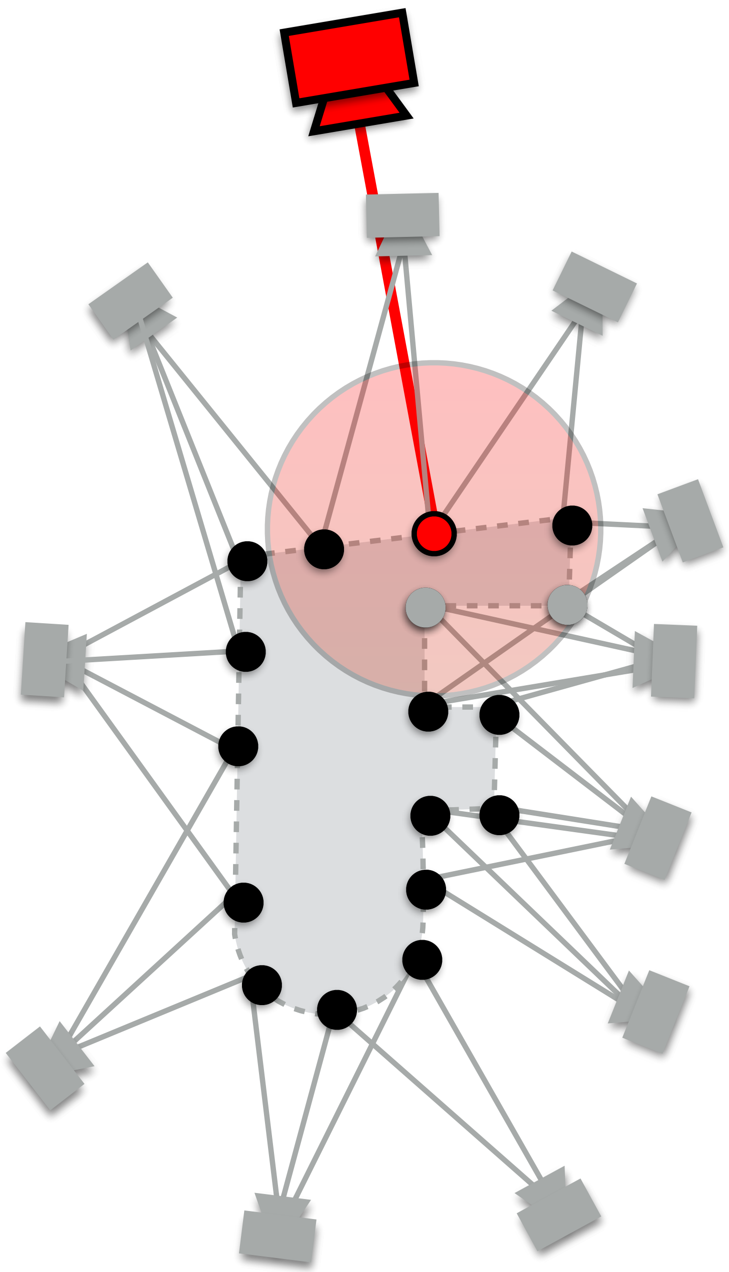
Visibility Filtering



Filter out 3D-to-2D matching candidates

[Sattler et al., ECCV'12]

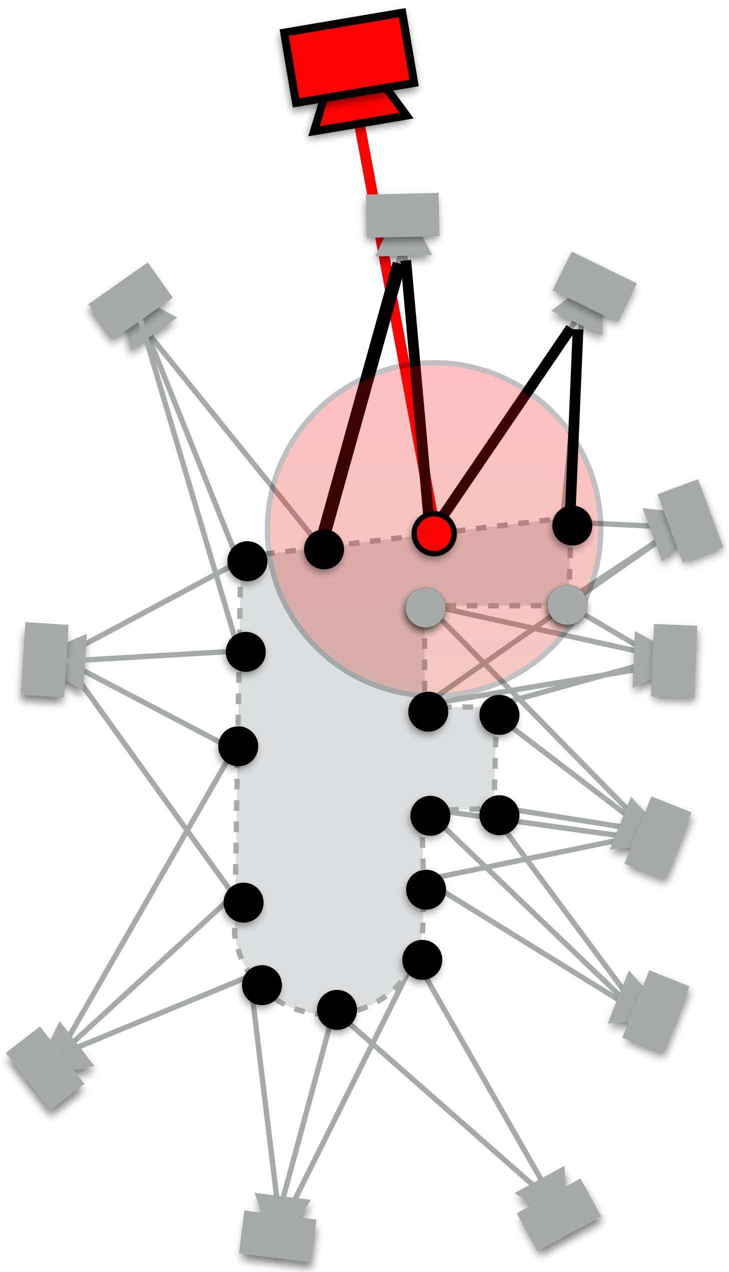
Visibility Filtering



Filter out 3D-to-2D matching candidates

[Sattler et al., ECCV'12]

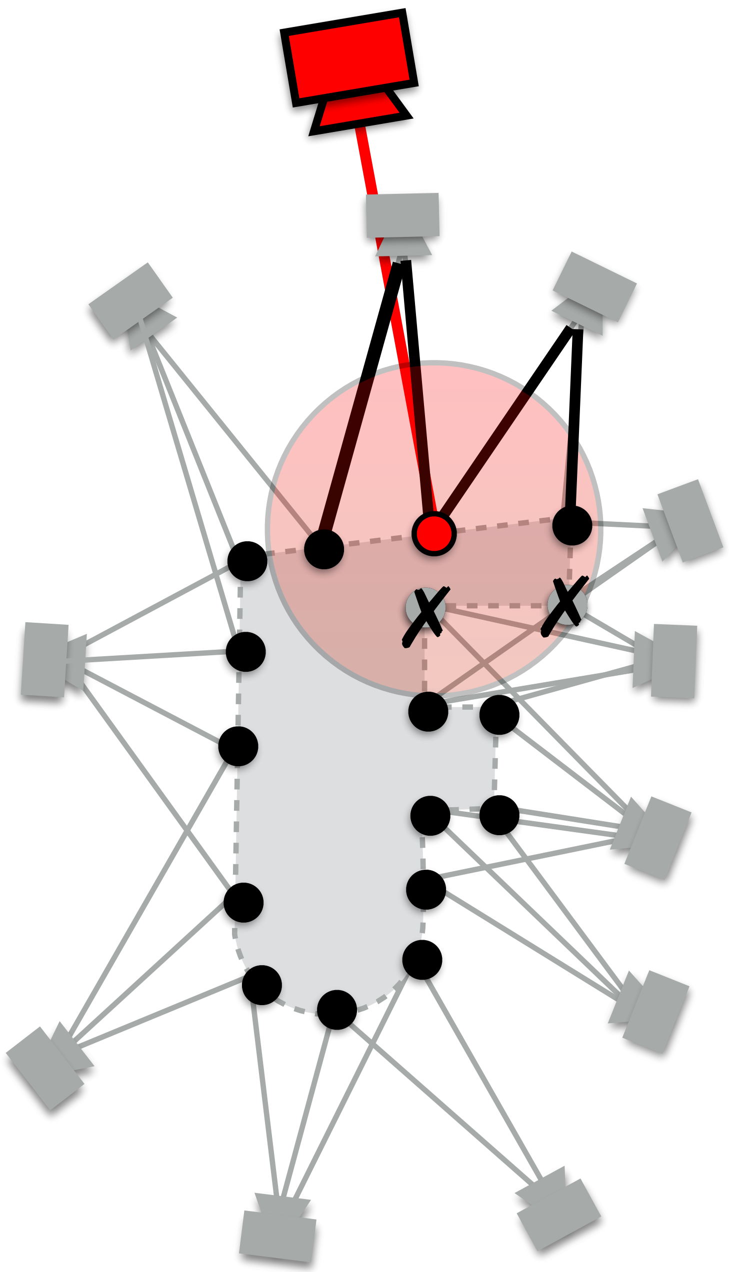
Visibility Filtering



Filter out 3D-to-2D matching candidates

[Sattler et al., ECCV'12]

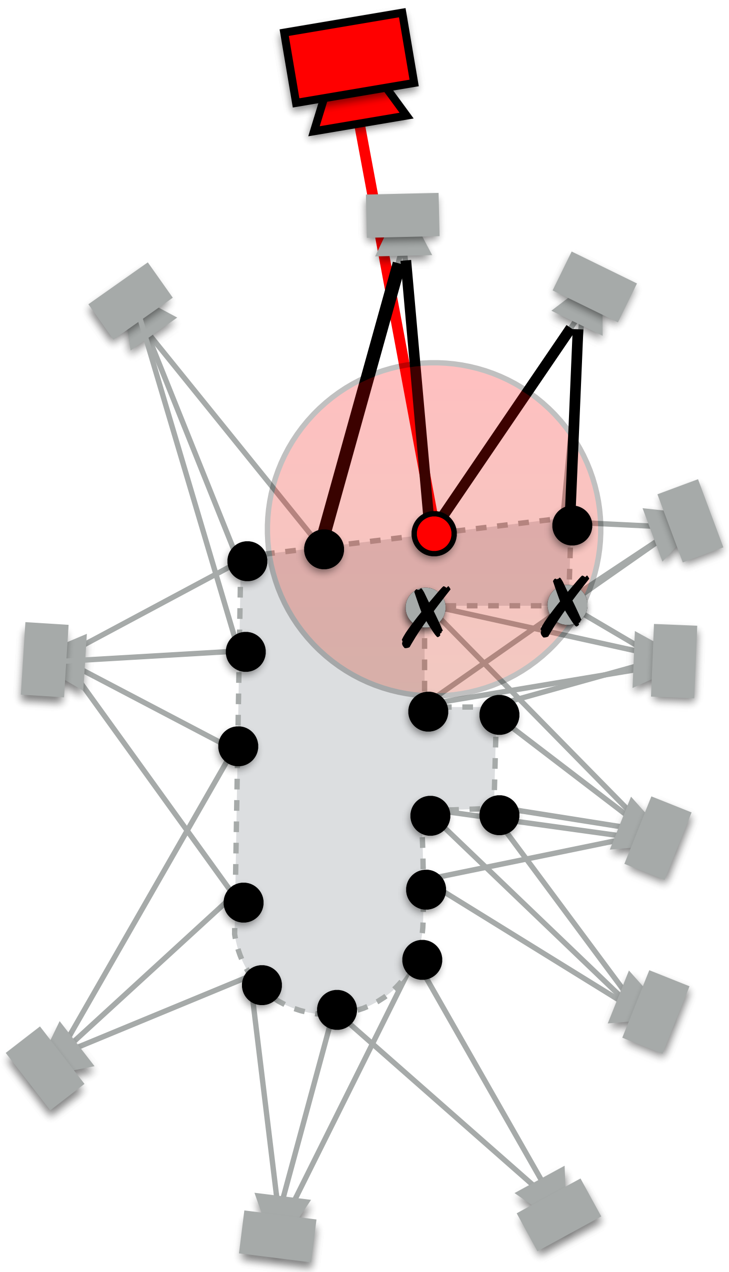
Visibility Filtering



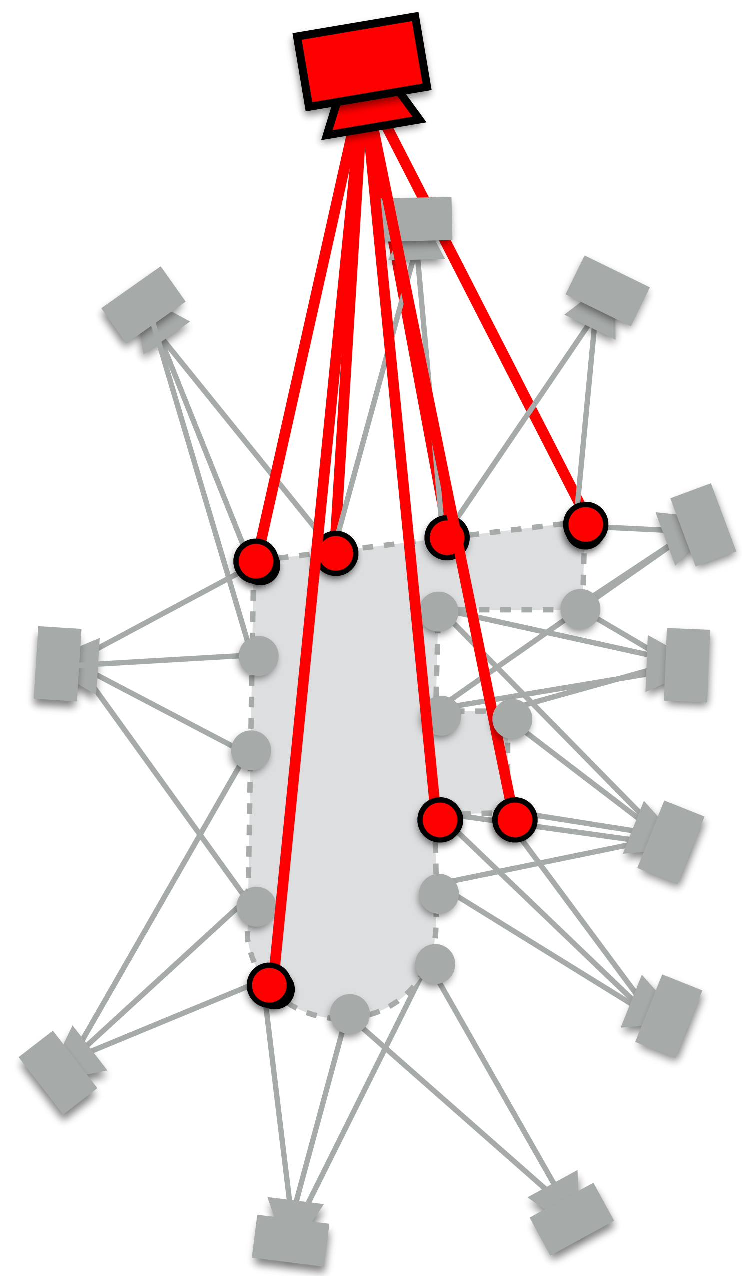
Filter out 3D-to-2D matching candidates

[Sattler et al., ECCV'12]

Visibility Filtering



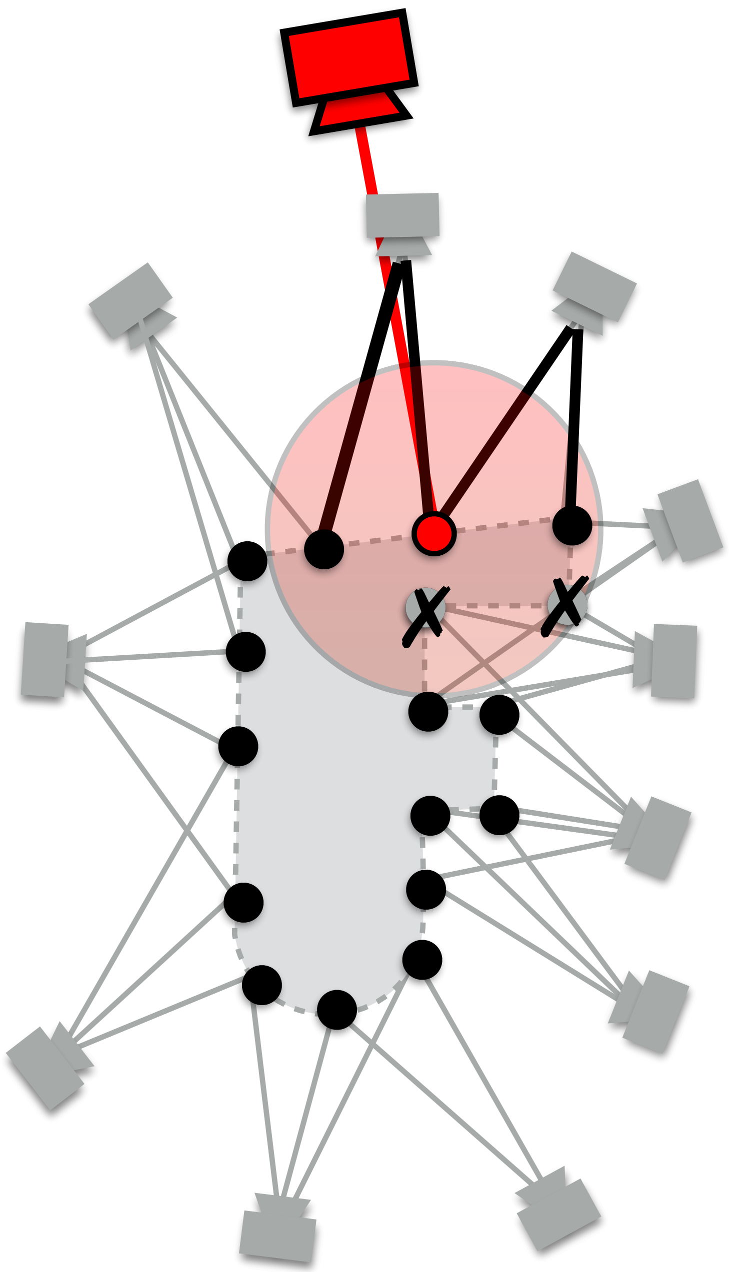
Filter out 3D-to-2D matching candidates



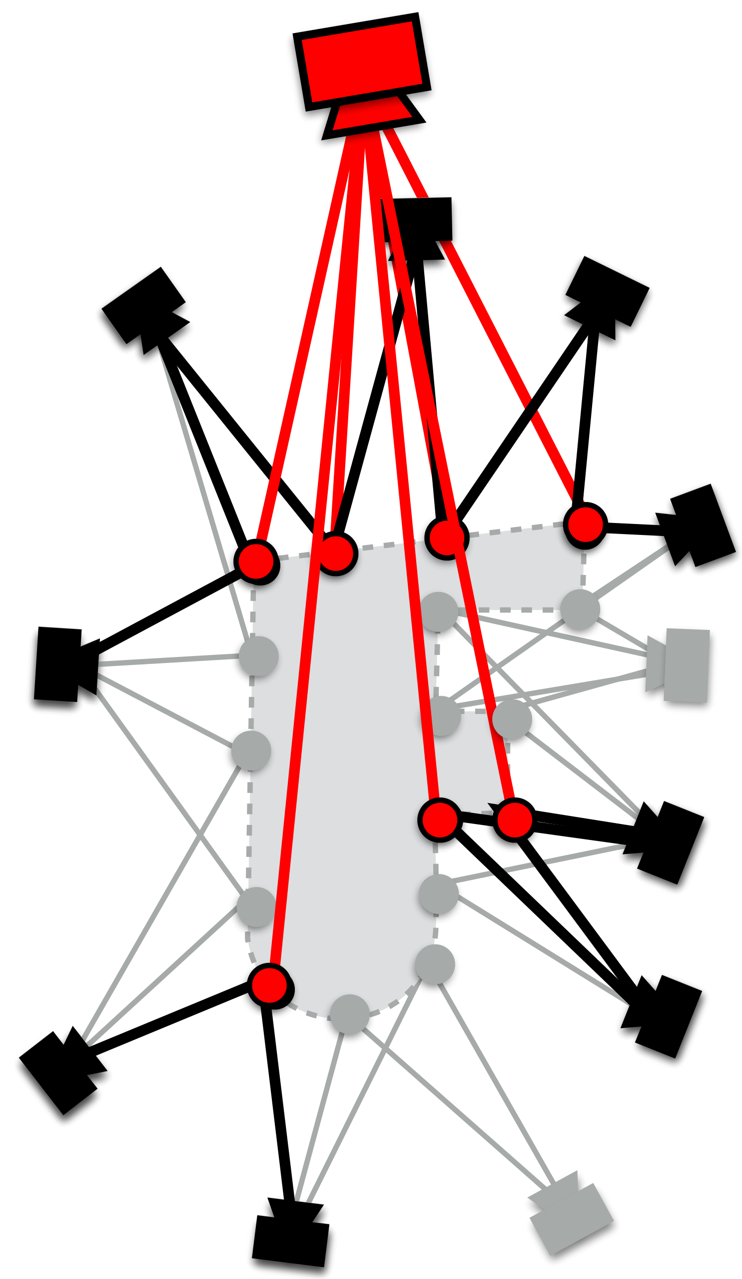
Remove wrong matches before RANSAC

[Sattler et al., ECCV'12]

Visibility Filtering



Filter out 3D-to-2D matching candidates



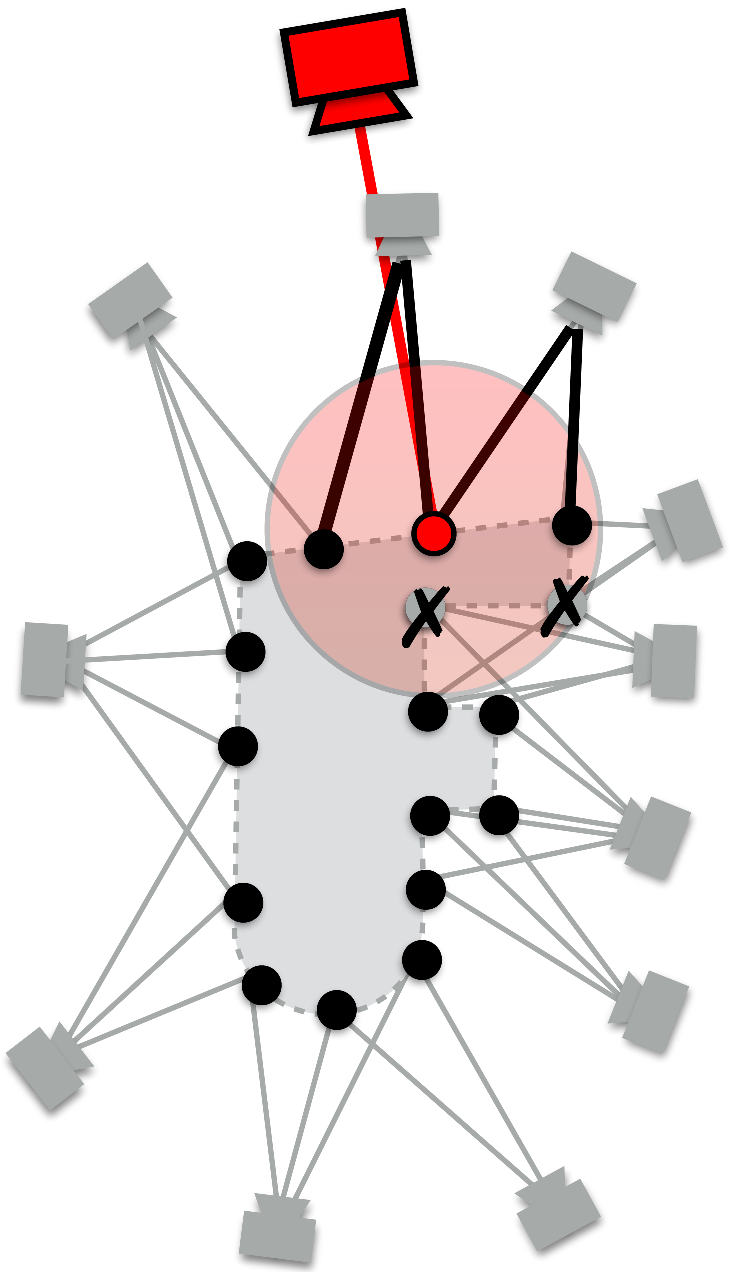
Remove wrong matches before RANSAC

[Sattler et al., ECCV'12]

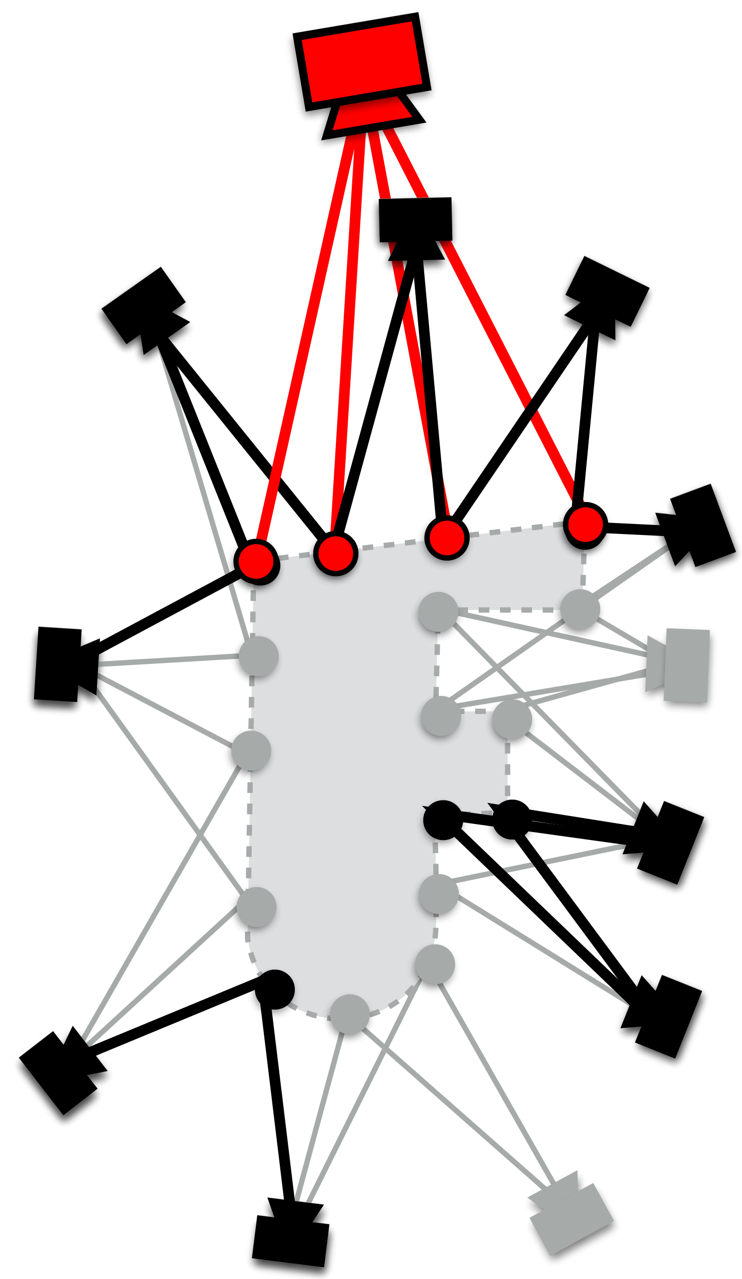
Computer Vision and Geometry Group

ETH zürich

Visibility Filtering



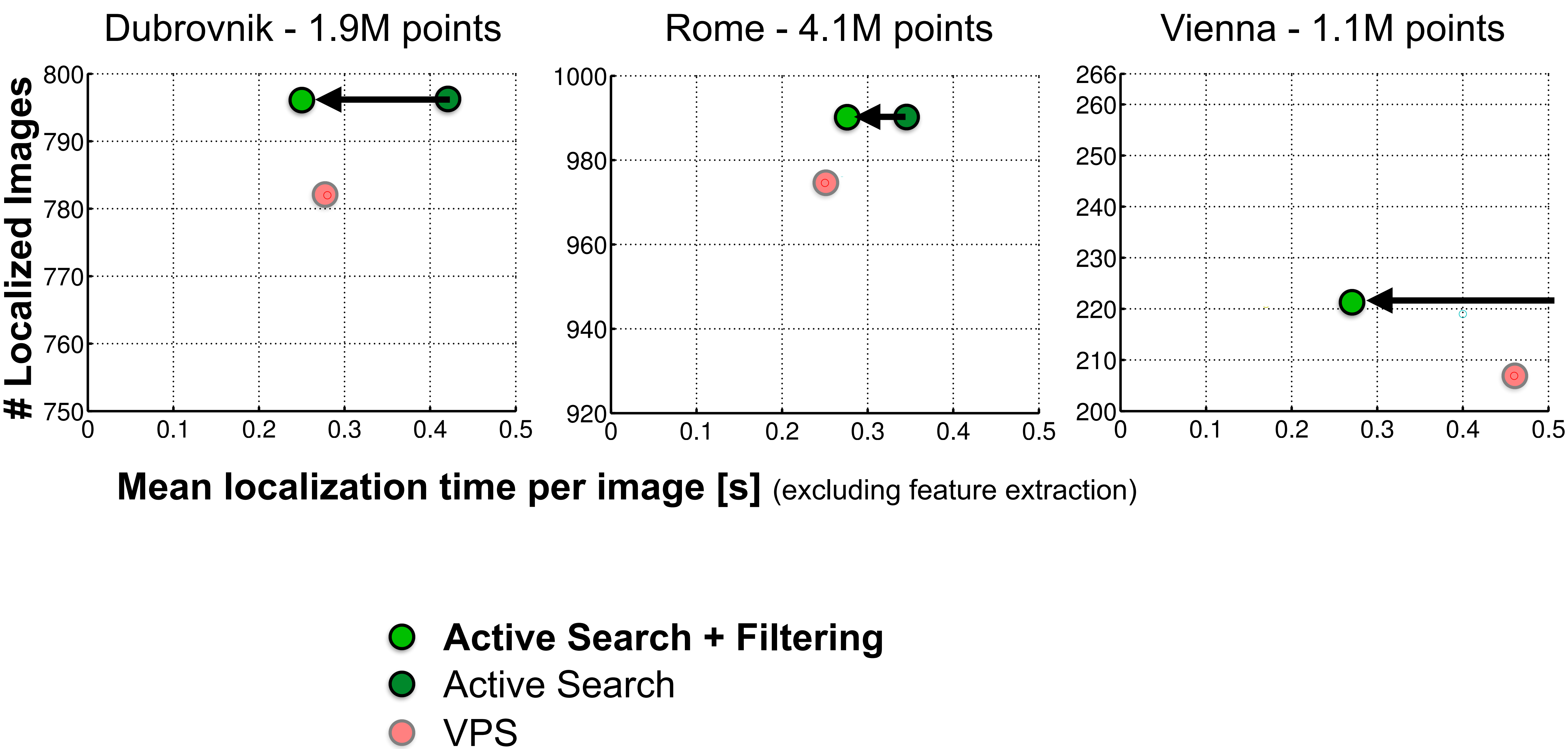
Filter out 3D-to-2D matching candidates



Remove wrong matches before RANSAC

[Sattler et al., ECCV'12]

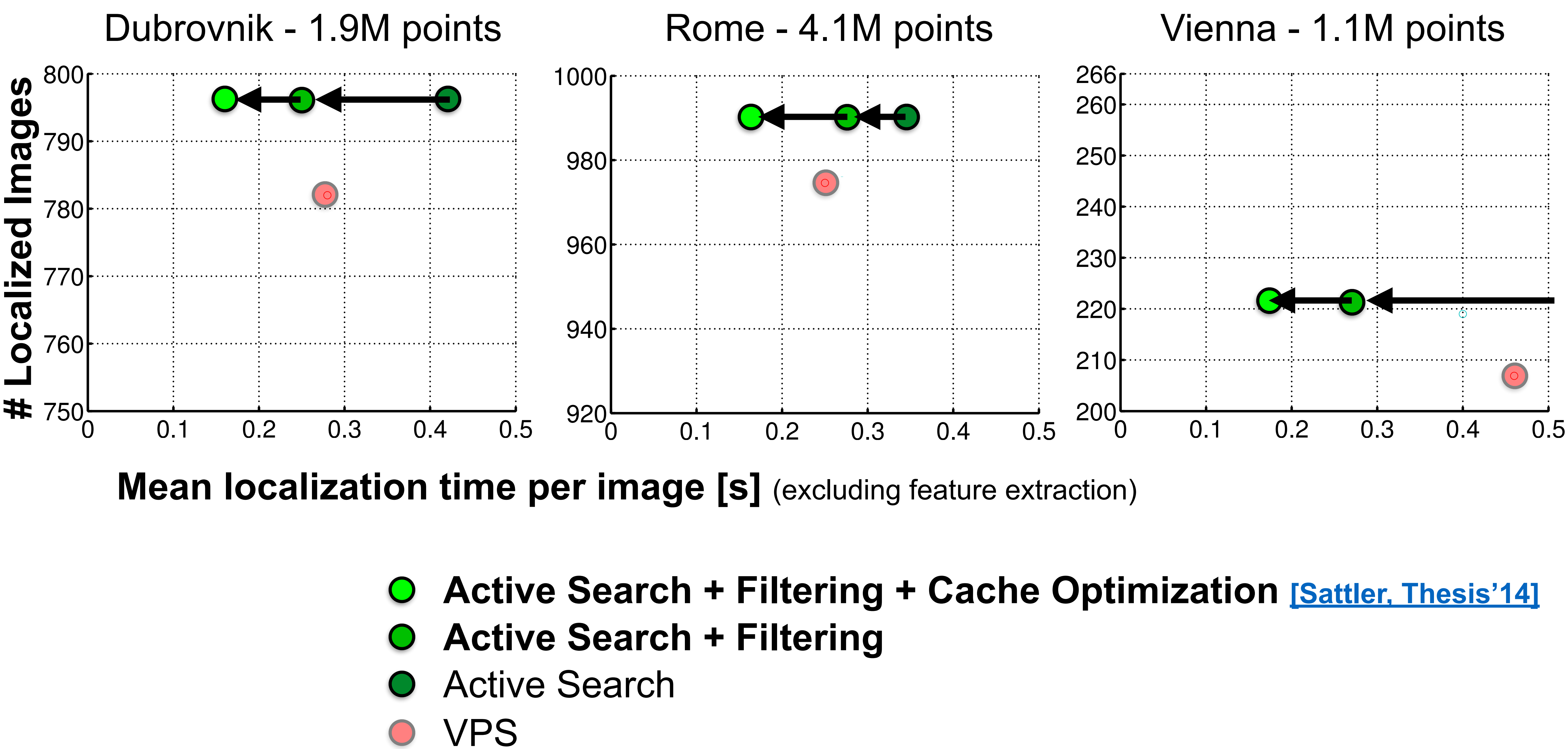
Results



code will be available “soon”

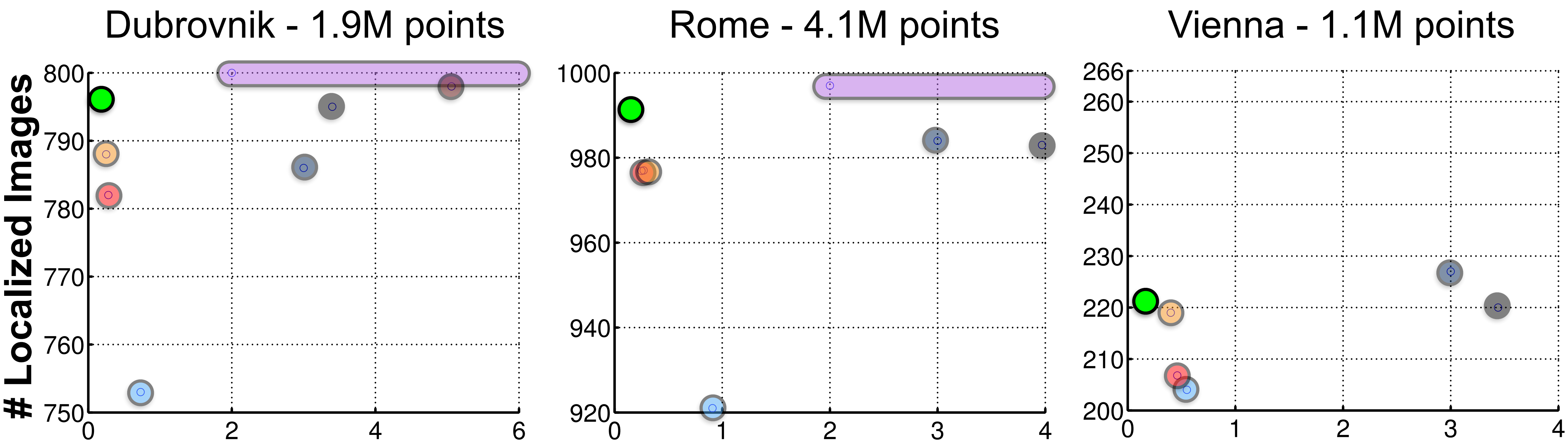
Computer Vision and Geometry Group ETH zürich

Results



code will be available "soon"

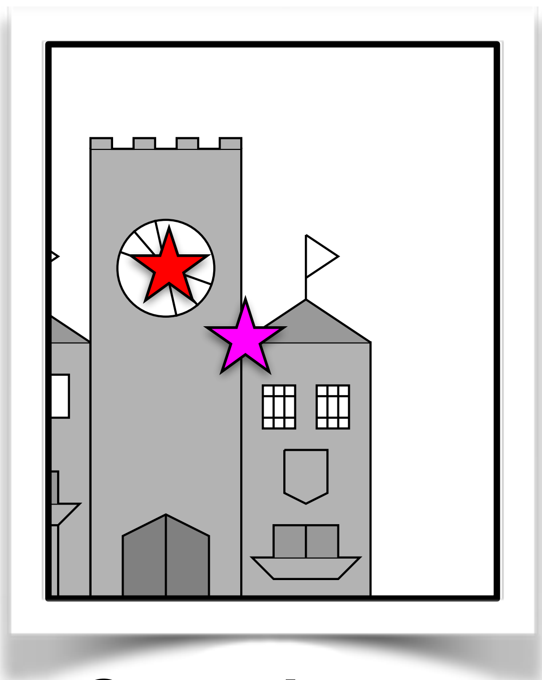
Results



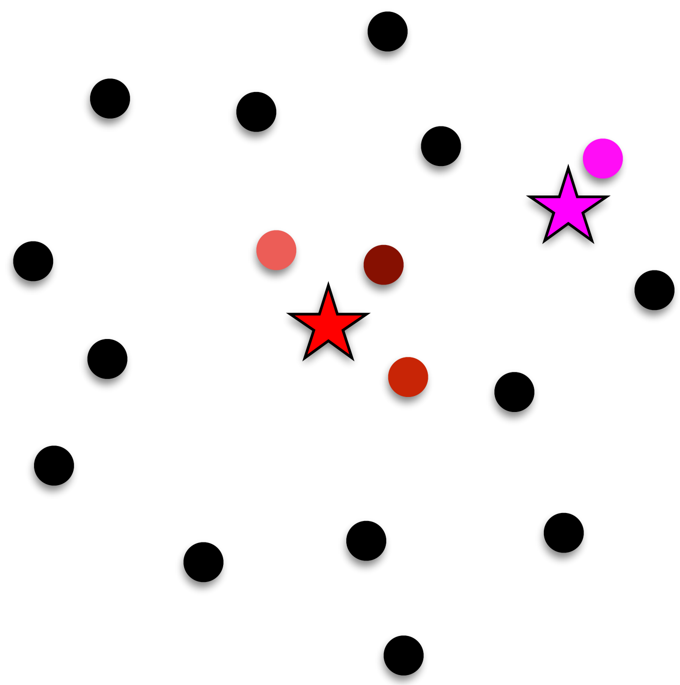
Mean localization time per image [s] (excluding feature extraction)

- **Active Search + Filtering + Cache Optimization** [Sattler, Thesis'14]
- kd-tree
- P2F [Li et al.,ECCV'10]
- VPS
- PGPM [Choudhary,ECCV'12]
- WPE [Li et al.,ECCV'12]
- Hamming Voting [Sattler et al.,BMVC'12]
- [Svarm et al.,CVPR'14]

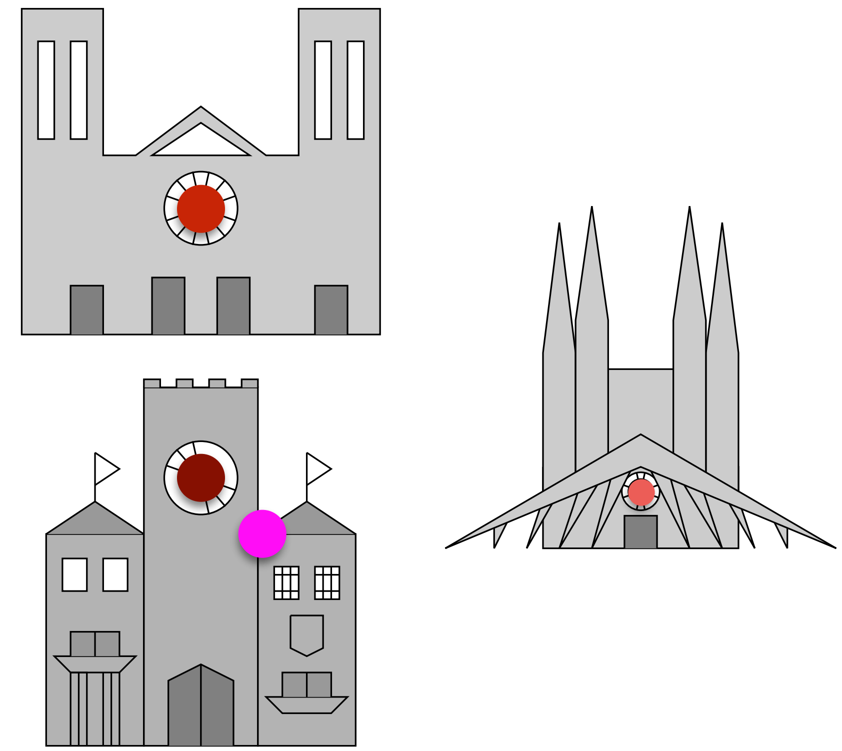
Active Search vs. Pure 2D-to-3D Matching



Query Image

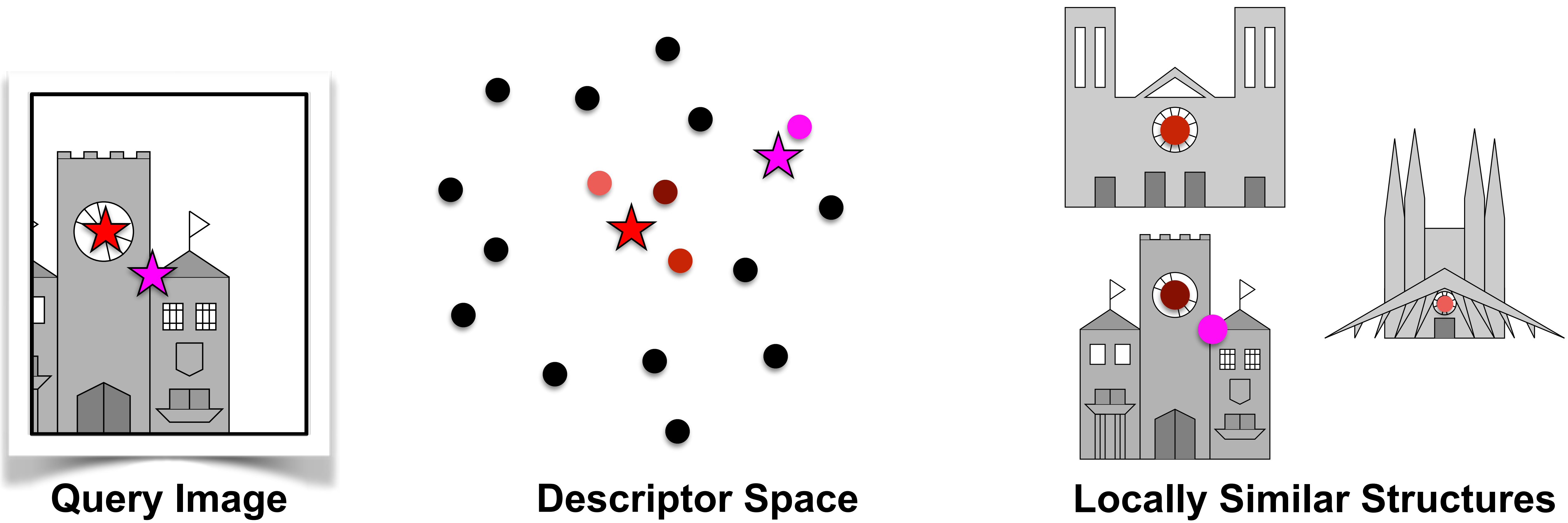


Descriptor Space



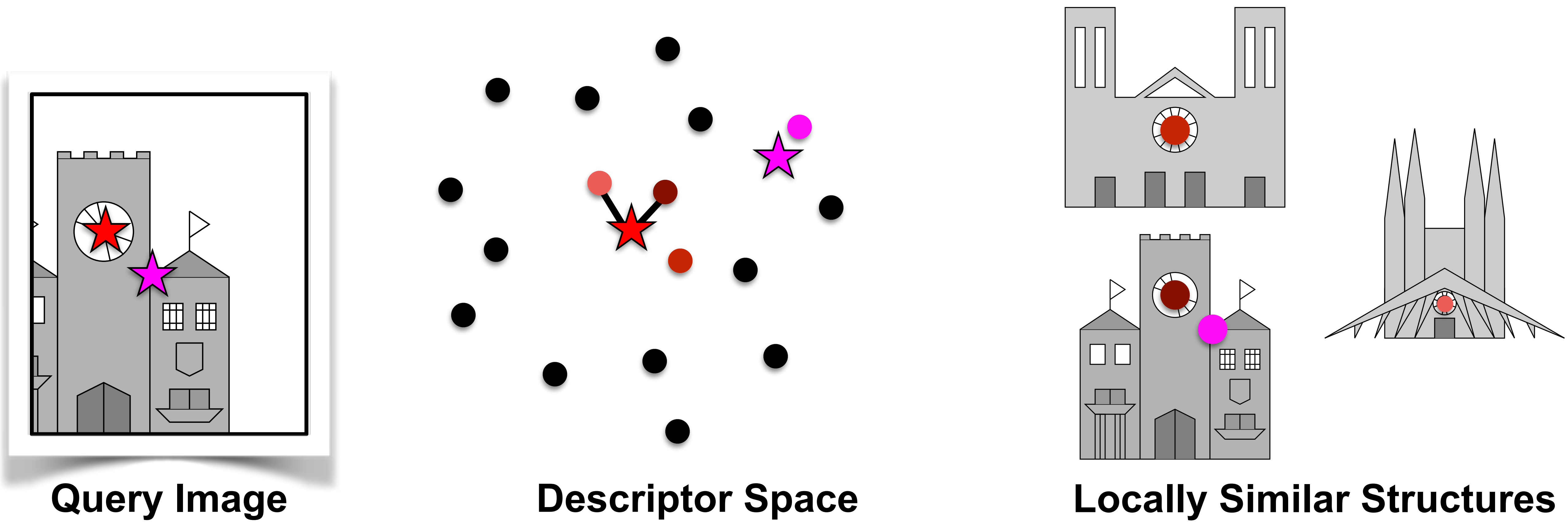
Locally Similar Structures

Active Search vs. Pure 2D-to-3D Matching



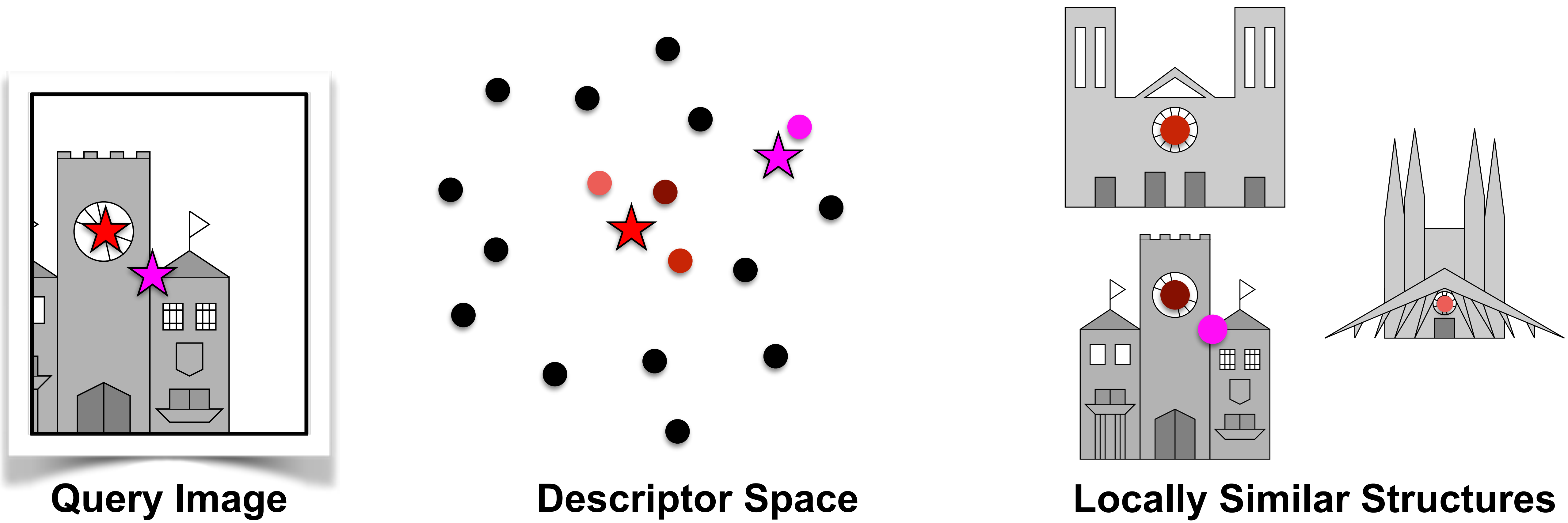
- Ratio test for 2D-to-3D matching rejects globally ambiguous matches

Active Search vs. Pure 2D-to-3D Matching



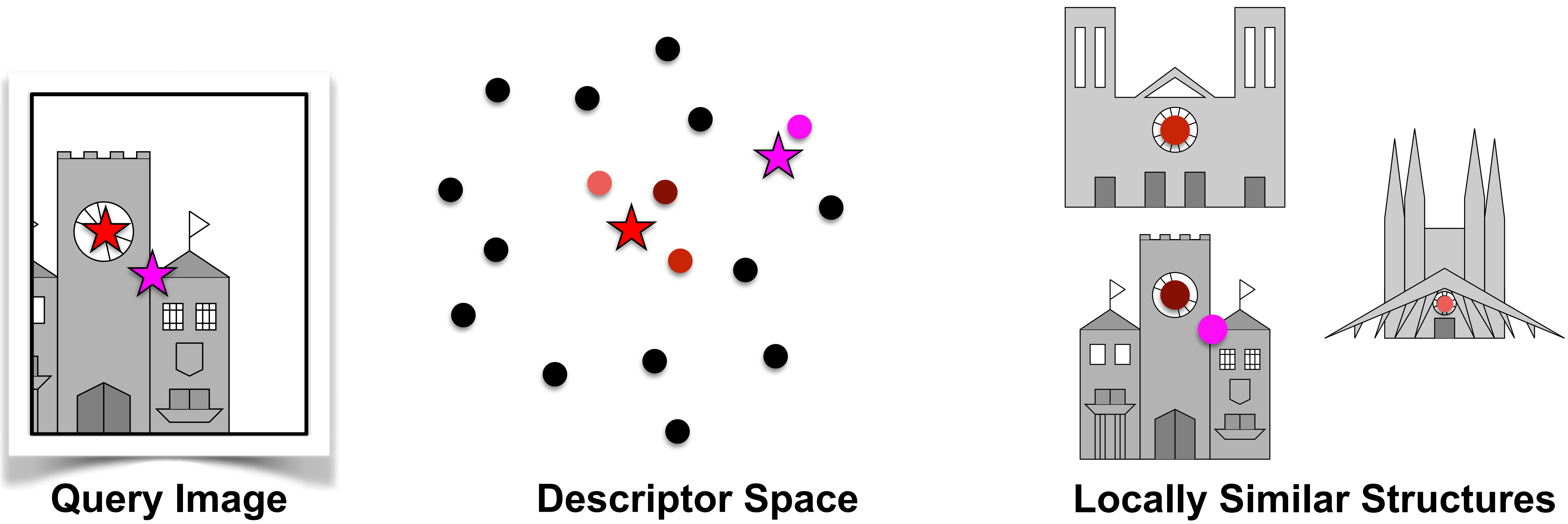
- Ratio test for 2D-to-3D matching rejects globally ambiguous matches

Active Search vs. Pure 2D-to-3D Matching



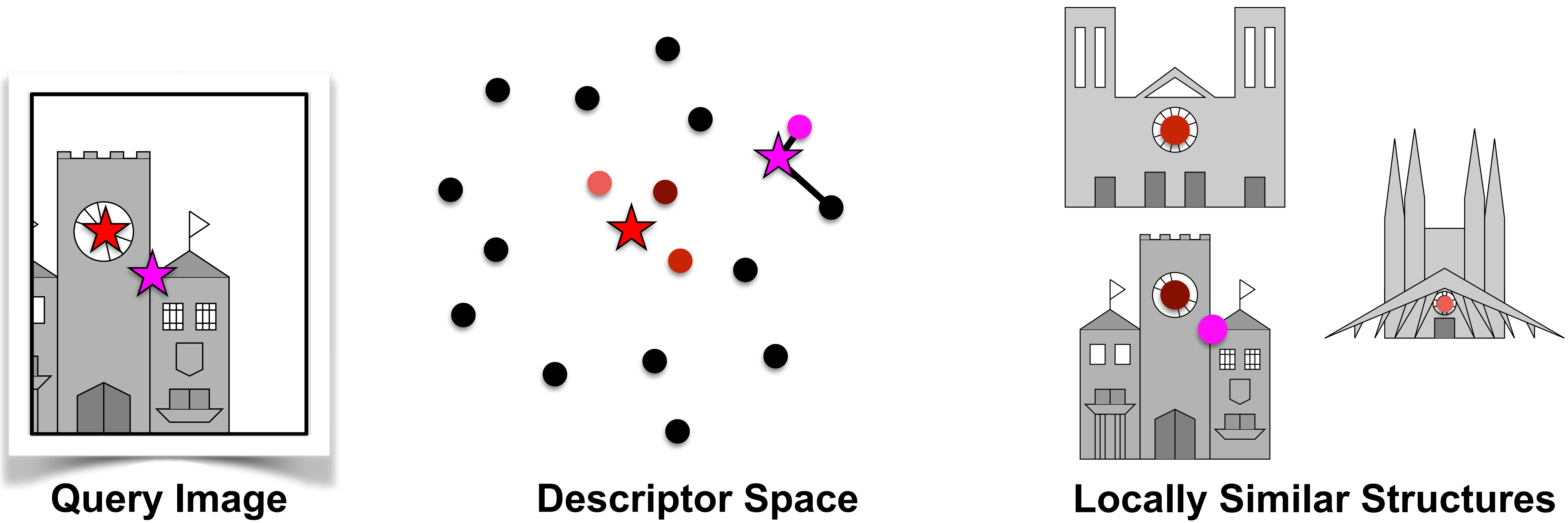
- Ratio test for 2D-to-3D matching rejects globally ambiguous matches

Active Search vs. Pure 2D-to-3D Matching



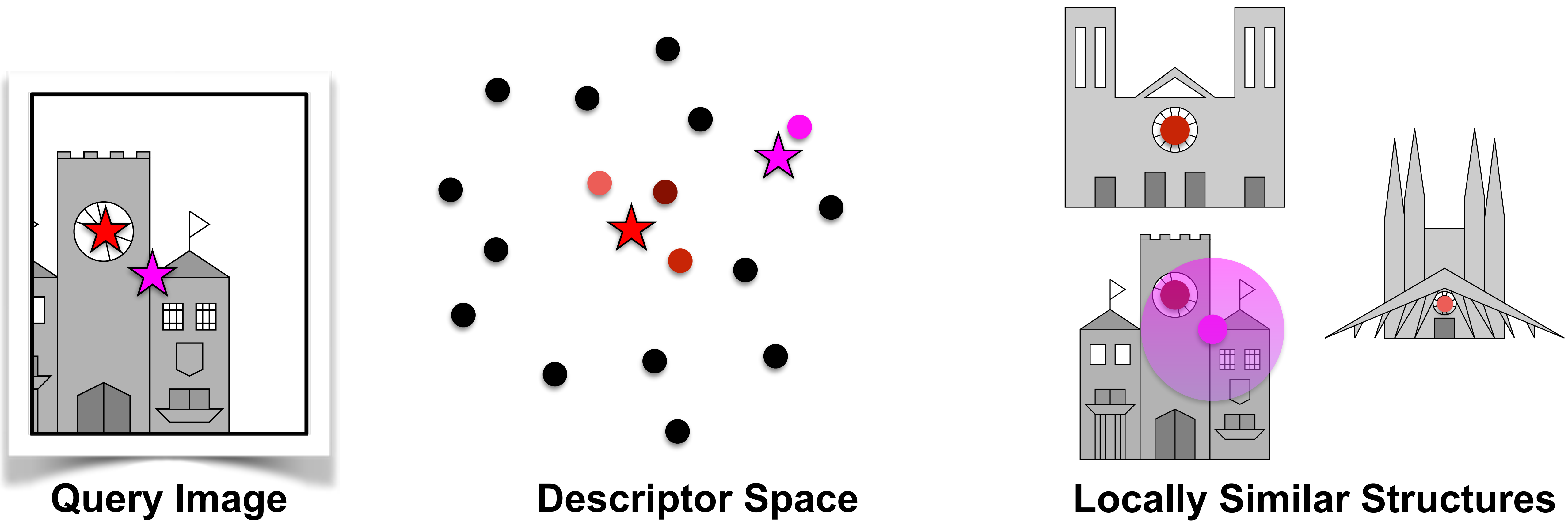
- Ratio test for 2D-to-3D matching rejects globally ambiguous matches
- Active Search can recover rejected matches using 3D-to-2D matching

Active Search vs. Pure 2D-to-3D Matching



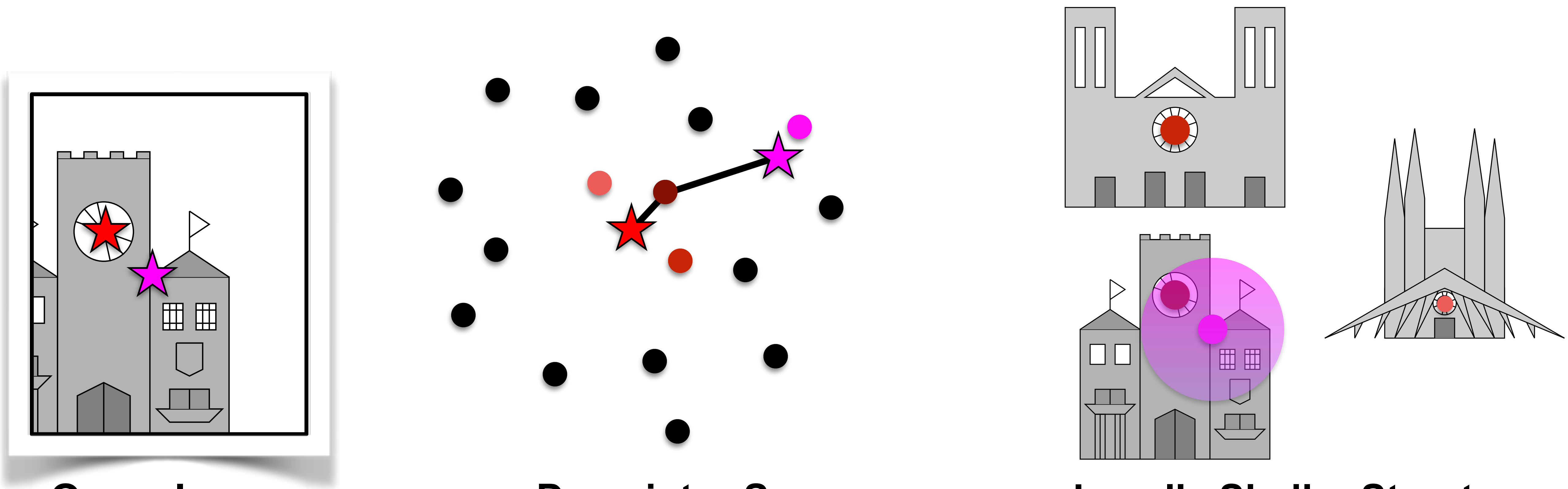
- Ratio test for 2D-to-3D matching rejects globally ambiguous matches
- Active Search can recover rejected matches using 3D-to-2D matching

Active Search vs. Pure 2D-to-3D Matching



- Ratio test for 2D-to-3D matching rejects globally ambiguous matches
- Active Search can recover rejected matches using 3D-to-2D matching

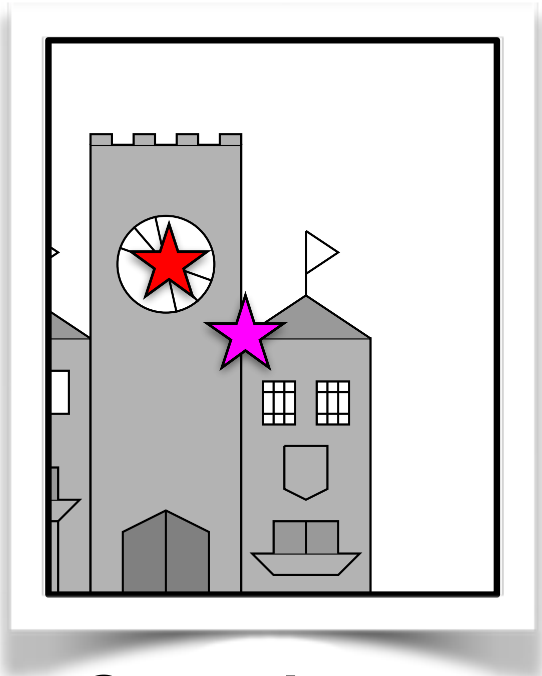
Active Search vs. Pure 2D-to-3D Matching



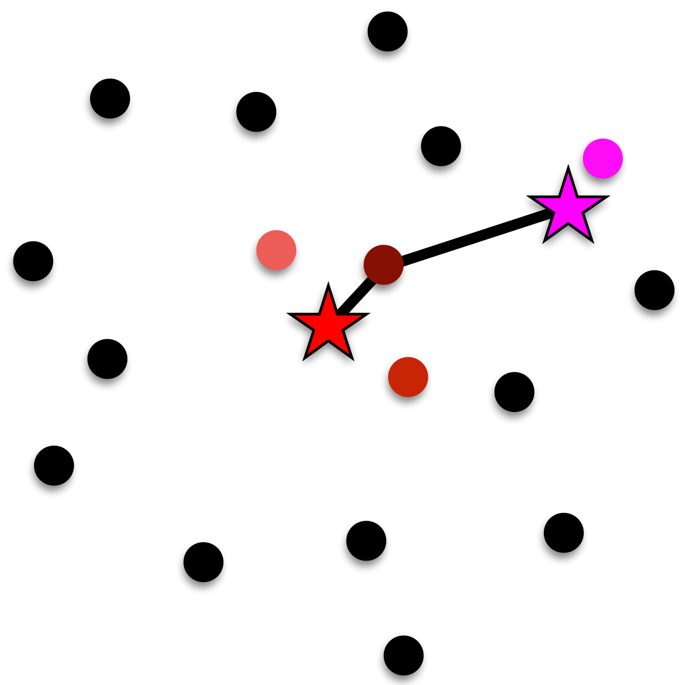
Query Image Descriptor Space Locally Similar Structures

- Ratio test for 2D-to-3D matching rejects globally ambiguous matches
- Active Search can recover rejected matches using 3D-to-2D matching

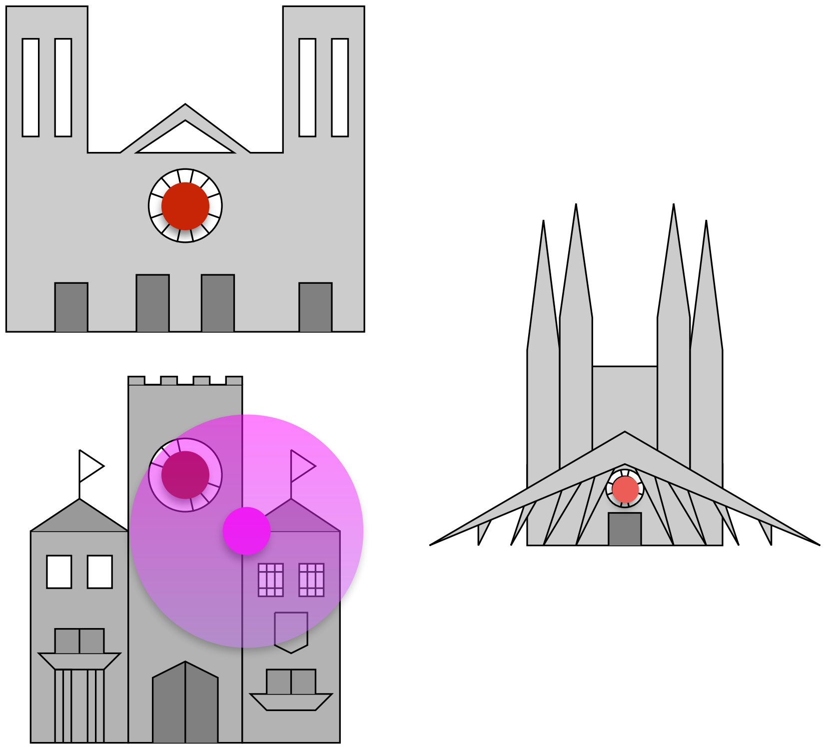
Active Search vs. Pure 2D-to-3D Matching



Query Image



Descriptor Space



Locally Similar Structures

- Ratio test for 2D-to-3D matching rejects globally ambiguous matches
- Active Search can recover rejected matches using 3D-to-2D matching
  - Scalability: Globally ambiguous structures more likely for larger models

Active Search vs. Pure 2D-to-3D Matching

Method	% Localized Images	Mean Localization Time [s]
kd-tree [Li et al., ECCV'12]	~87	"few seconds"
VPS	85.47	0.89
Active Search	**95.34**	**0.48**

Landmarks 1k dataset [Li et al., ECCV'12]

- Most popular 1k landmarks from Flickr
- 38M points reconstructed from 204k images
- 10k query images

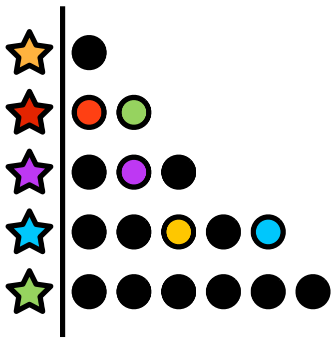
Active Search vs. Pure 2D-to-3D Matching

Method	% Localized Images	Mean Localization Time [s]
kd-tree [Li et al., ECCV'12]	~87	"few seconds"
VPS	85.47	0.89
Active Search	**95.34**	**0.48**
WPE [Li et al., ECCV'12]	**98.95**	"few seconds"

Landmarks 1k dataset [Li et al., ECCV'12]

- Most popular 1k landmarks from Flickr
- 38M points reconstructed from 204k images
- 10k query images

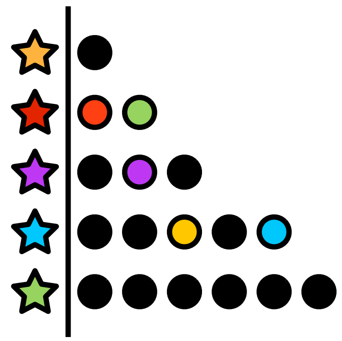
Key Insights



Prioritized Search

- 2D-to-3D matching more reliable than 3D-to-2D search
- Efficient search through **prioritization**
- Effectiveness reduced by **quantization**

Key Insights



Prioritized Search

- 2D-to-3D matching more reliable than 3D-to-2D search
- Efficient search through **prioritization**
- Effectiveness reduced by **quantization**

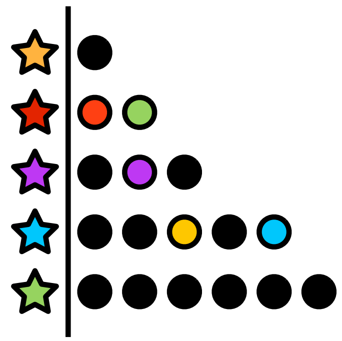
Active Search

- **Recover** missing matches via 3D-to-2D search

➡**State-of-the-art localization effectiveness**

Computer Vision and Geometry Group ETH zürich

Key Insights



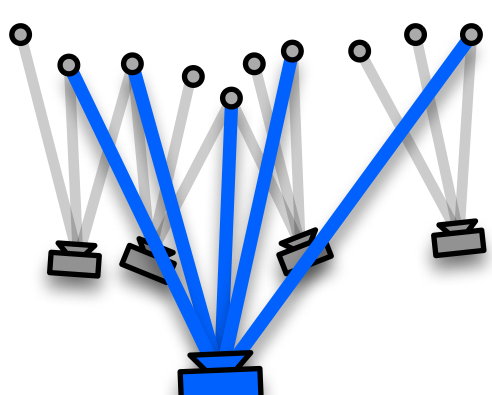
Prioritized Search

- 2D-to-3D matching more reliable than 3D-to-2D search
- Efficient search through **prioritization**
- Effectiveness reduced by **quantization**

Active Search

- **Recover** missing matches via 3D-to-2D search

➡**State-of-the-art localization effectiveness**



Visibility Filtering

- Accelerate both 3D-to-2D matching & pose estimation

➡**State-of-the-art localization efficiency & effectiveness**

Overview

- Efficient & Effective Large-Scale Localization
- Real-Time Mobile Localization
- Open Challenges

Localization On Mobile Devices

Goals:

- Real-time localization on mobile device
- Scalable, independent from scene size

Localization On Mobile Devices

Goals:

- Real-time localization on mobile device
- Scalable, independent from scene size

Challenges:

Localization On Mobile Devices

Goals:

- Real-time localization on mobile device
- Scalable, independent from scene size

Challenges:

- Limited memory

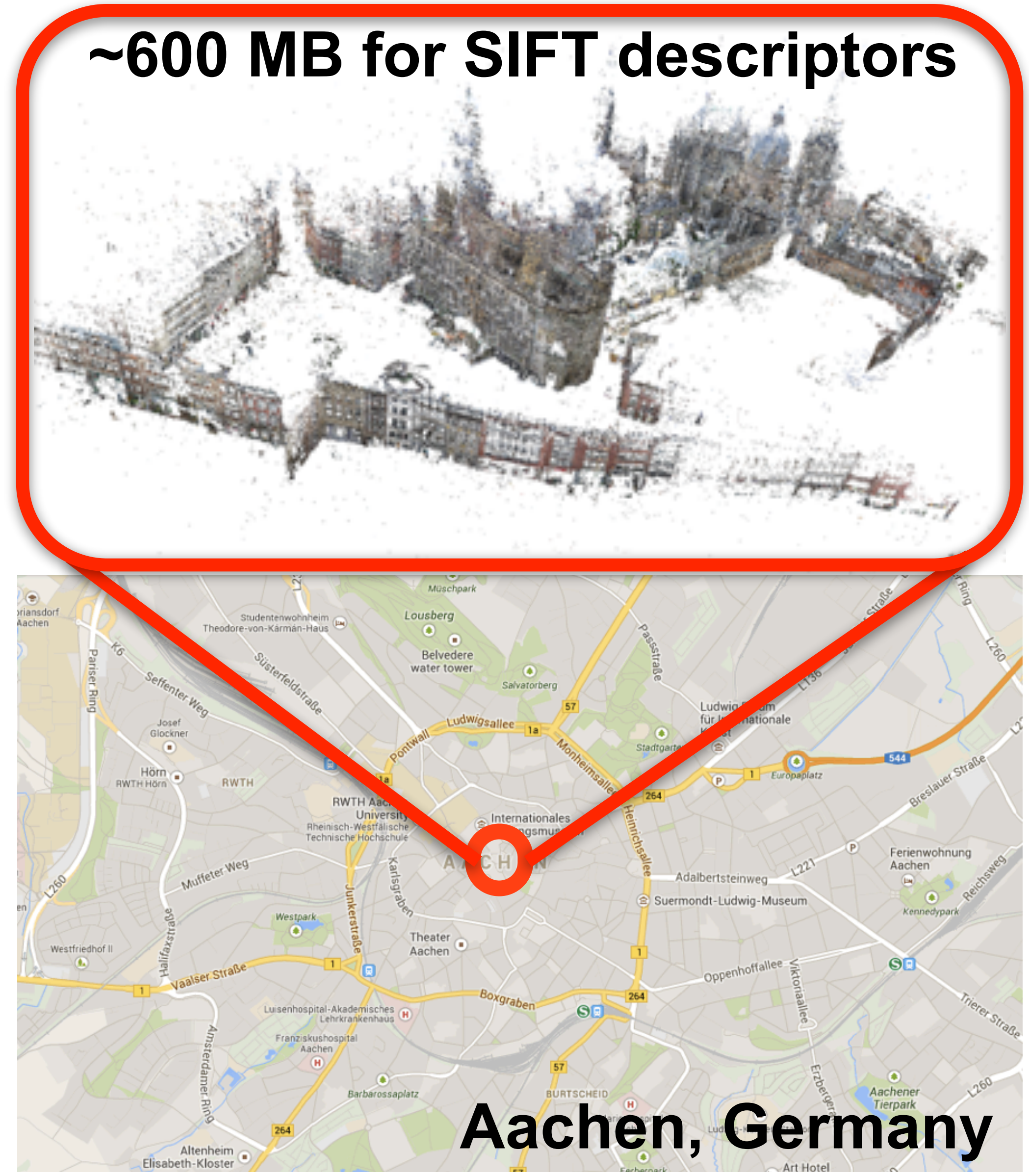
Localization On Mobile Devices

Goals:

- Real-time localization on mobile device
- Scalable, independent from scene size

Challenges:

- Limited memory



[Aachen dataset]

Computer Vision and Geometry Group ETH zürich

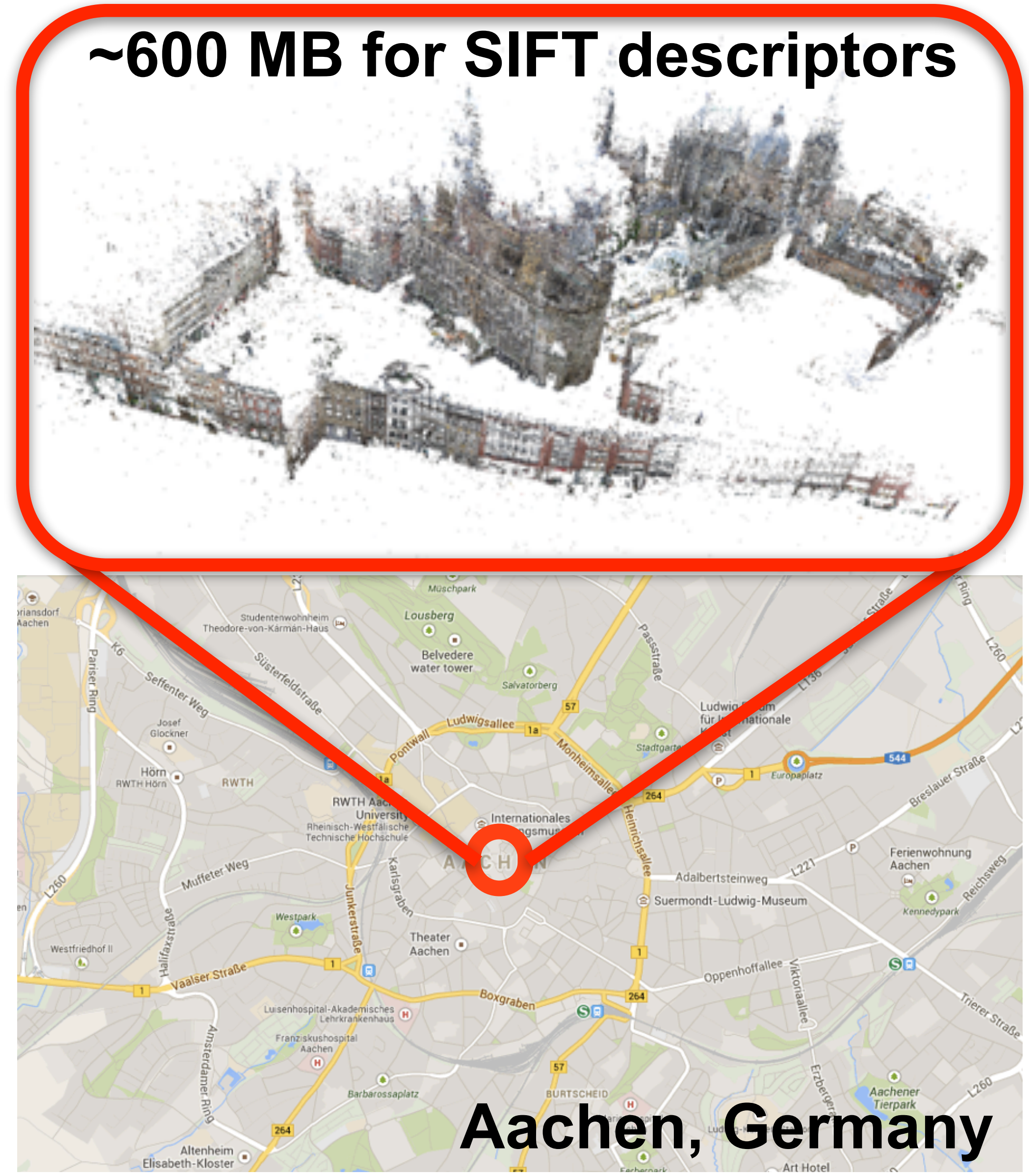
Localization On Mobile Devices

Goals:

- Real-time localization on mobile device
- Scalable, independent from scene size

Challenges:

- Limited memory
- Limited computational capabilities



[Aachen dataset]

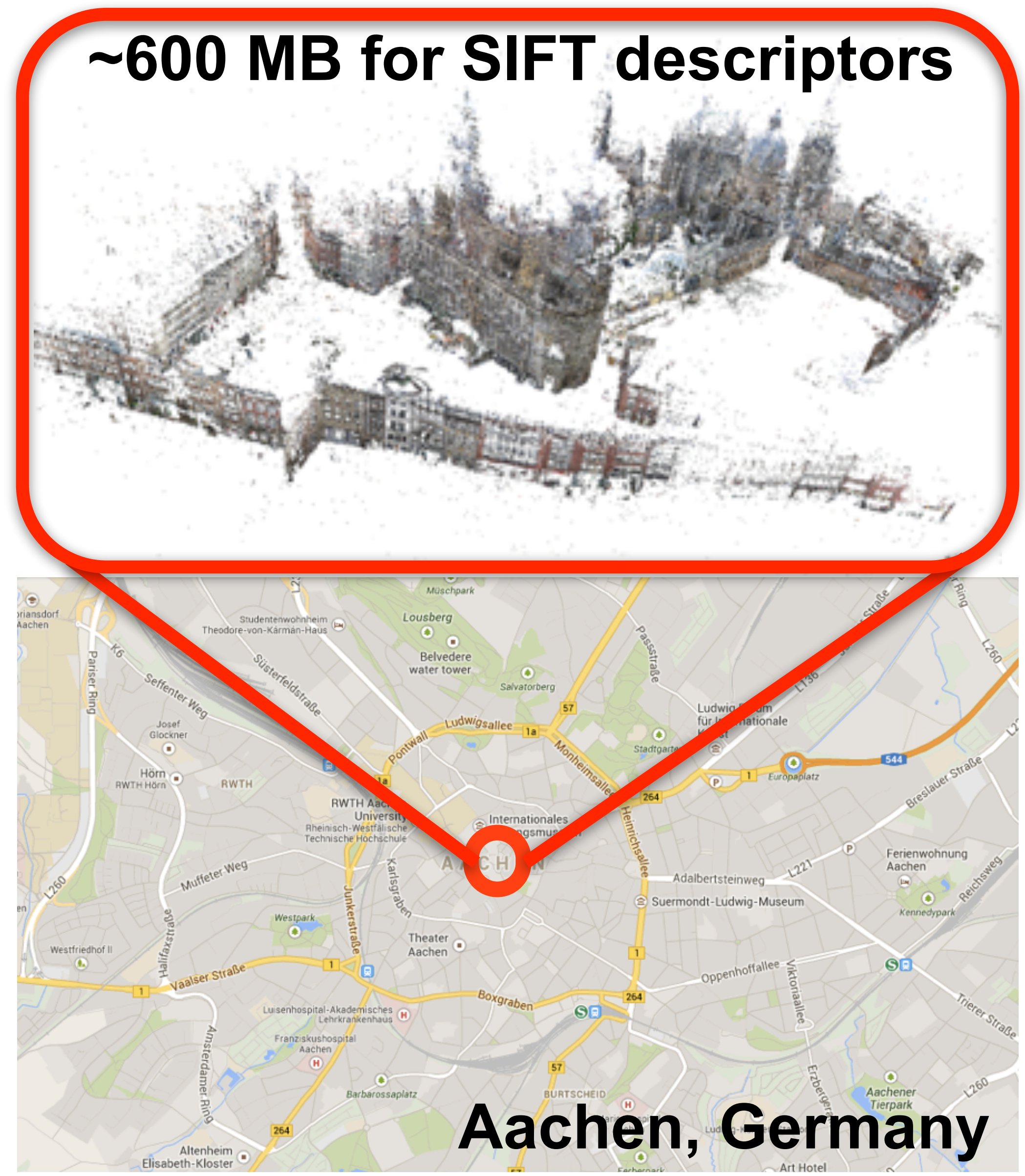
Localization On Mobile Devices

Goals:

- Real-time localization on mobile device
- Scalable, independent from scene size

Challenges:

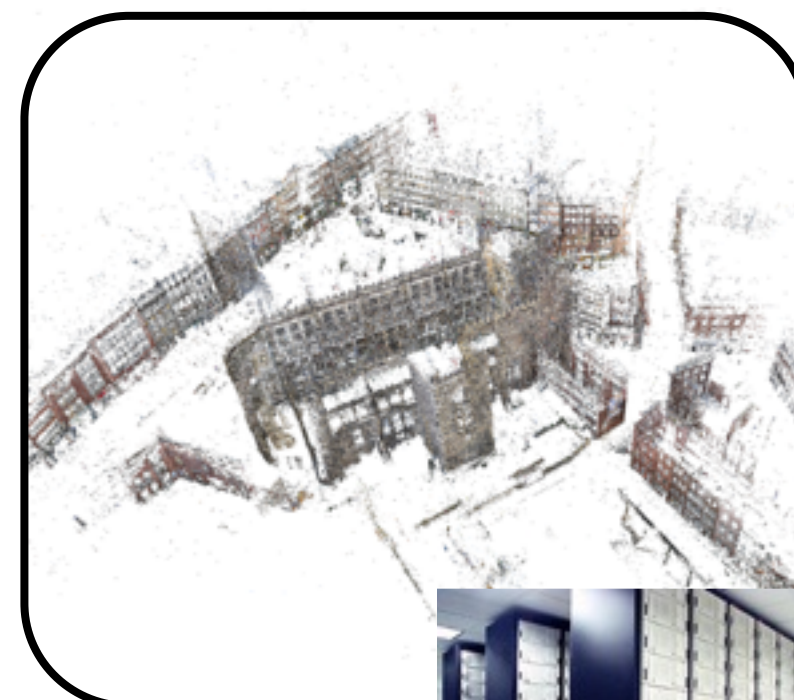
- Limited memory
- Limited computational capabilities
- Localization accuracy



[Aachen dataset]

A Scalable Mobile Localization Pipeline

Mobile Device



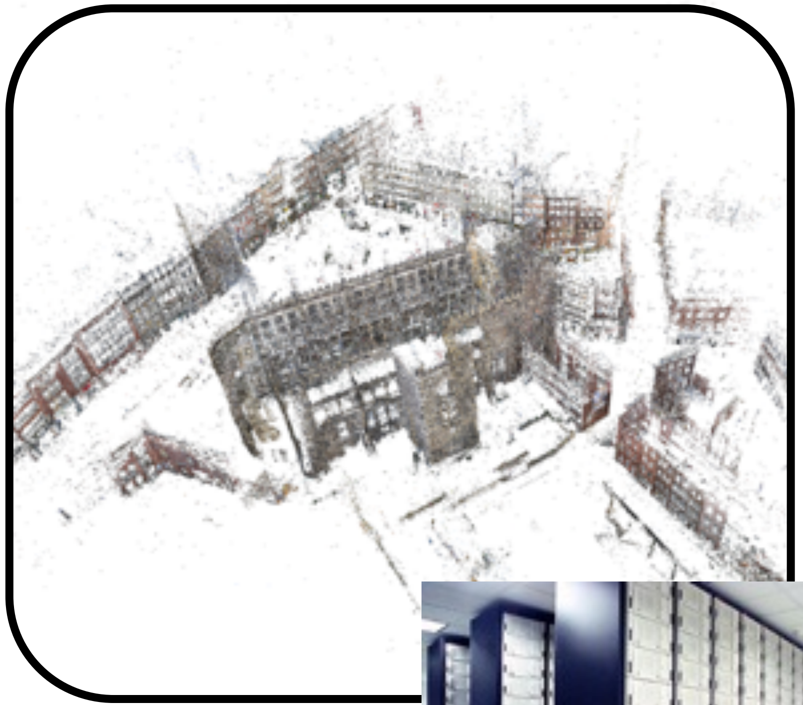
Localization Server

[Middelberg et al., ECCV'14]

A Scalable Mobile Localization Pipeline

Mobile Device

Send Image



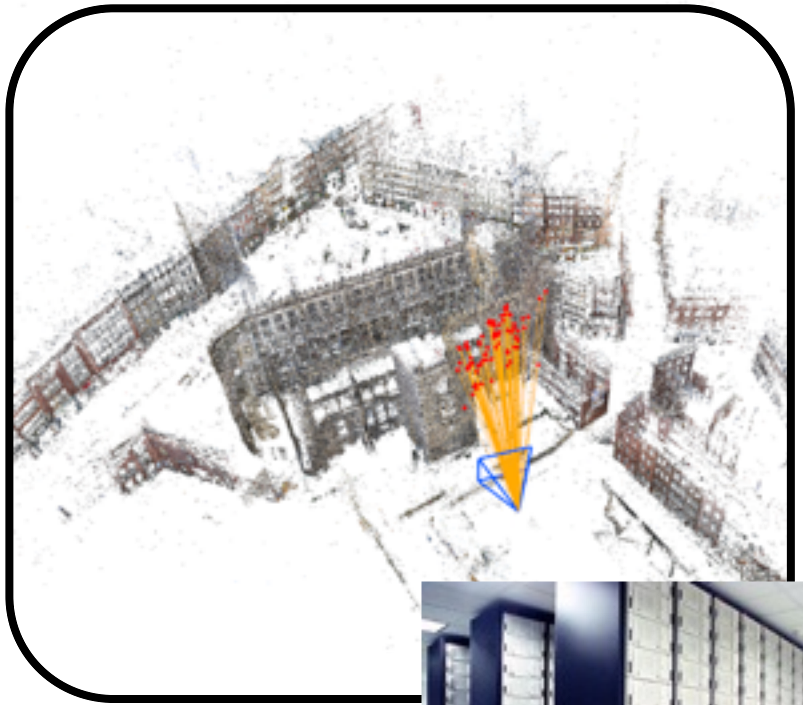
Localization Server

[Middelberg et al., ECCV'14]

A Scalable Mobile Localization Pipeline

Mobile Device

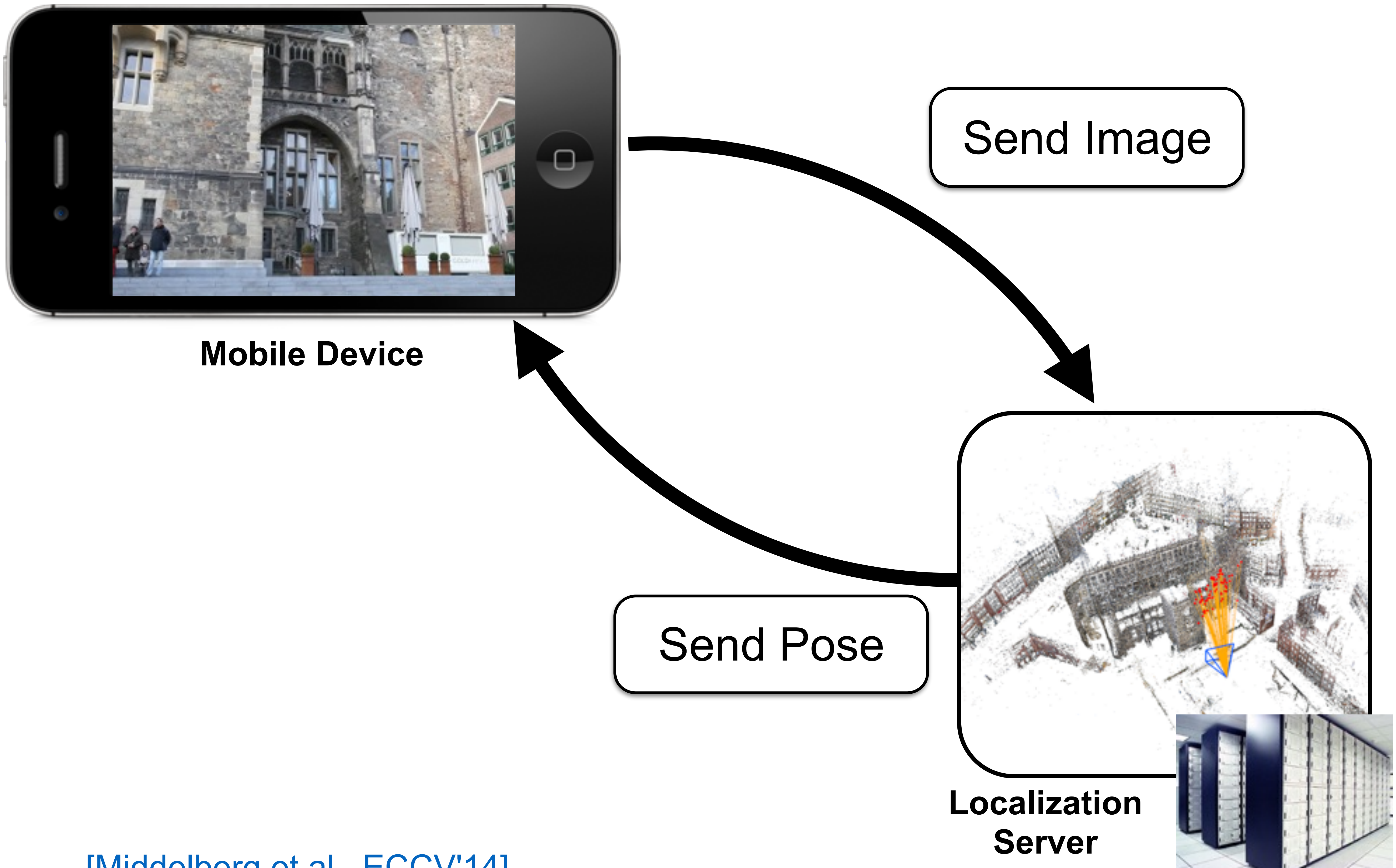
Send Image



Localization Server

[Middelberg et al., ECCV'14]

A Scalable Mobile Localization Pipeline



[Middelberg et al., ECCV'14]

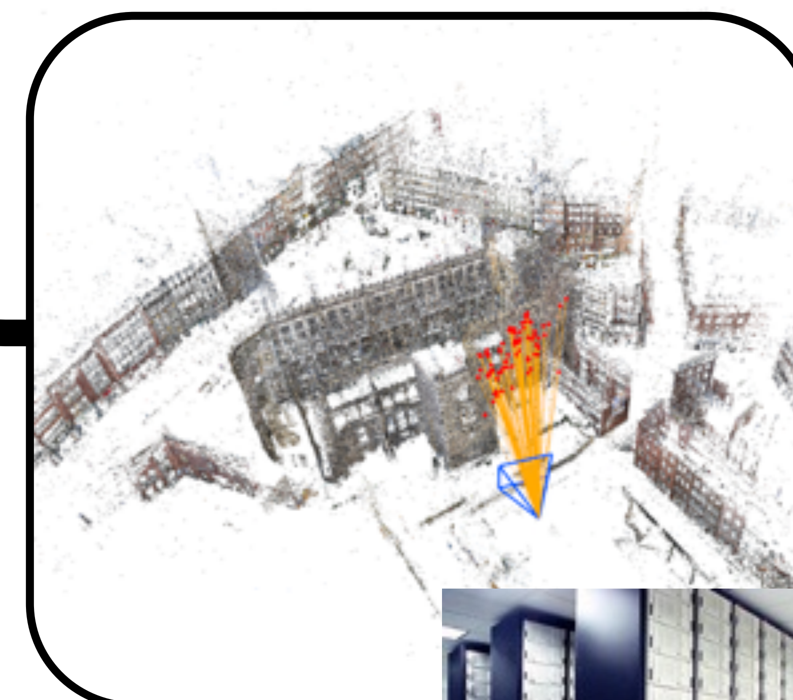
A Scalable Mobile Localization Pipeline

Mobile Device

Send Image

≥1s delay!

Send Pose



Localization Server

[Middelberg et al., ECCV'14]

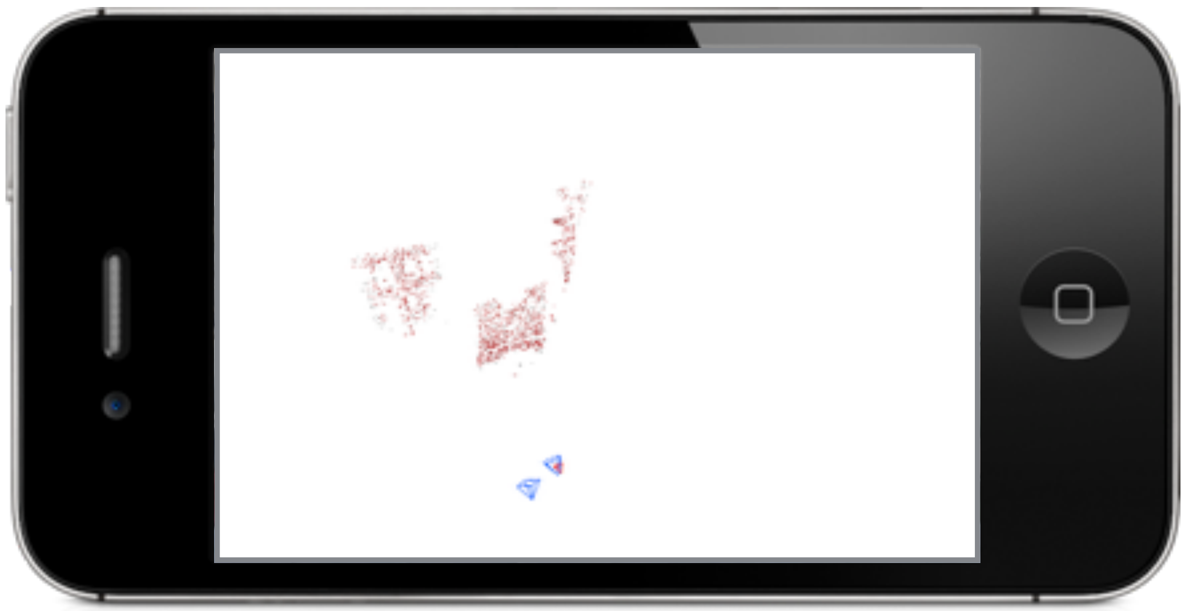
A Scalable Mobile Localization Pipeline

Mobile Device

Localization Server

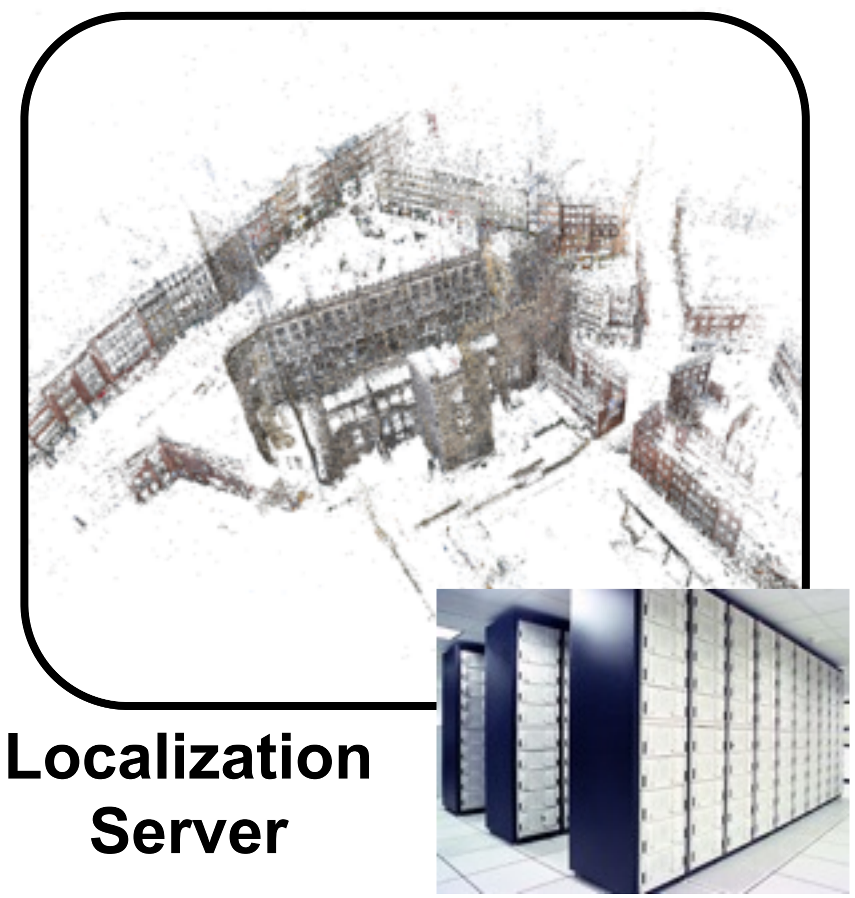
[Middelberg et al., ECCV'14]

A Scalable Mobile Localization Pipeline



Mobile Device

- Run SLAM / PTAM for **real-time camera tracking**



Localization Server

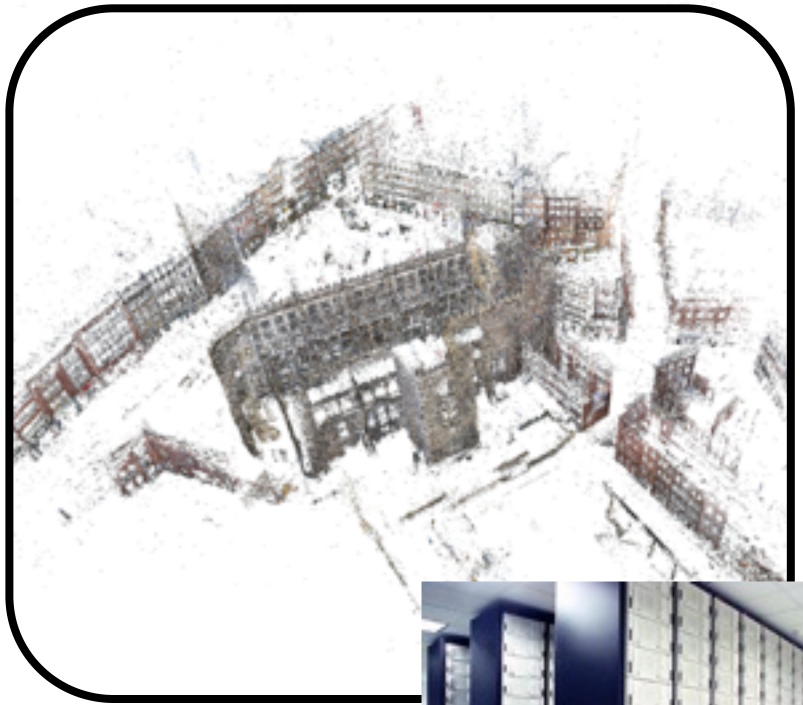
[Middelberg et al., ECCV'14]

Computer Vision and Geometry Group ETH zürich

A Scalable Mobile Localization Pipeline

Mobile Device

- Run SLAM / PTAM for **real-time camera tracking**



Localization Server

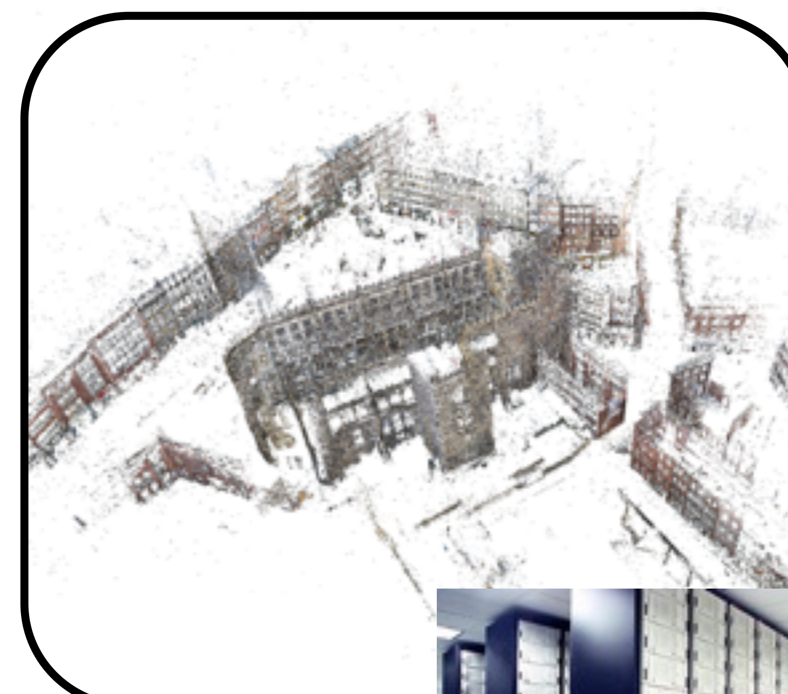
[Middelberg et al., ECCV'14]

A Scalable Mobile Localization Pipeline

Mobile Device

Send Image

- Run SLAM / PTAM for **real-time camera tracking**



Localization Server

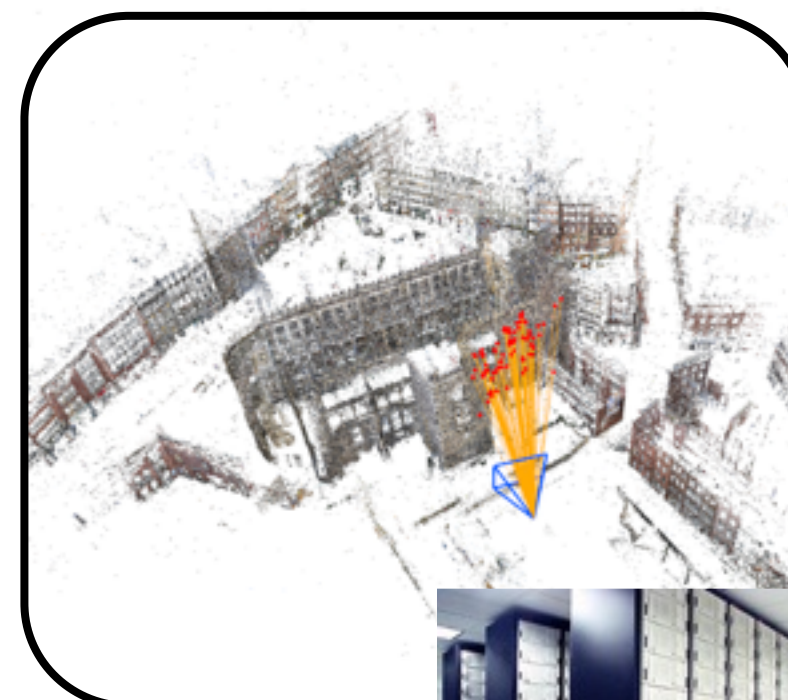
[Middelberg et al., ECCV'14]

A Scalable Mobile Localization Pipeline

Mobile Device

- Run SLAM / PTAM for **real-time camera tracking**

Send Image



Localization Server

[Middelberg et al., ECCV'14]

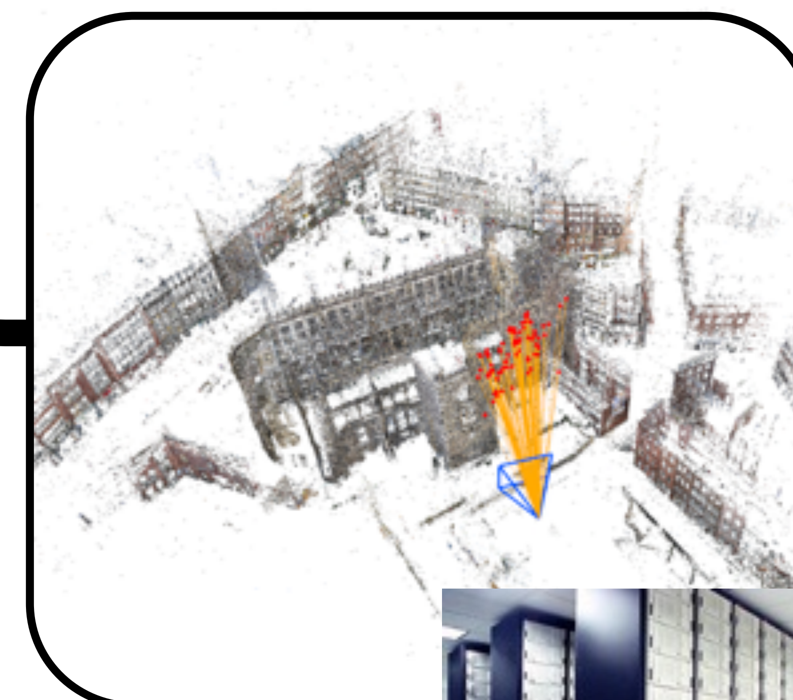
A Scalable Mobile Localization Pipeline

Mobile Device

- Run SLAM / PTAM for **real-time camera tracking**

Send Image

Send Pose



Localization Server

[Middelberg et al., ECCV'14]

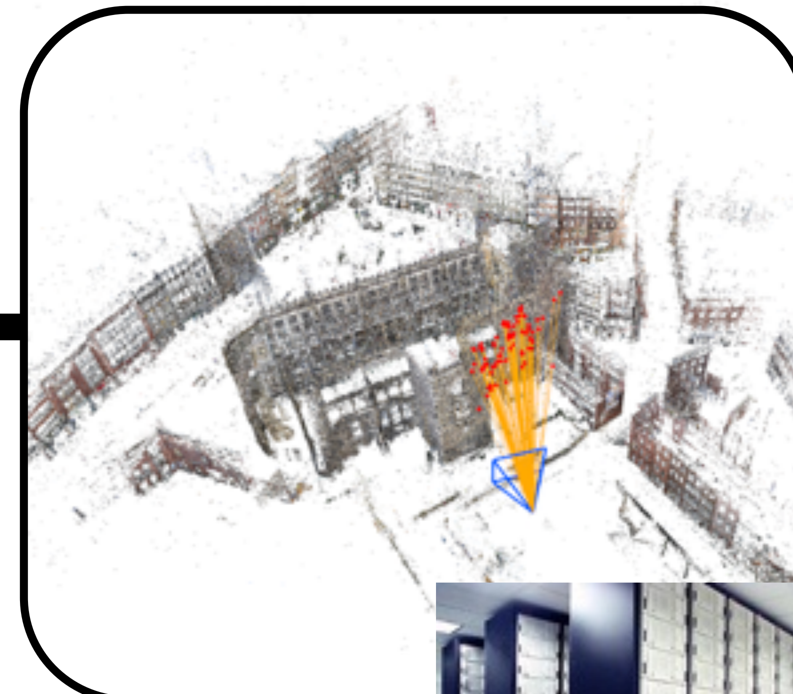
A Scalable Mobile Localization Pipeline

Mobile Device

Send Image

- Run SLAM / PTAM for **real-time camera tracking**
- Use global poses to **prevent drift**

Send Pose



Localization Server

[Middelberg et al., ECCV'14]

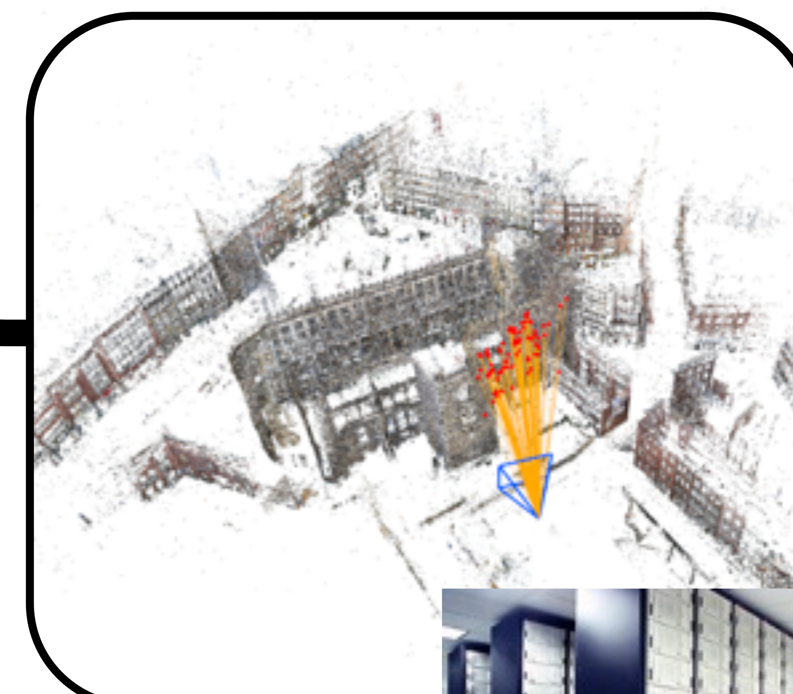
A Scalable Mobile Localization Pipeline

Mobile Device

Send Image

- Run SLAM / PTAM for **real-time camera tracking**
- Use global poses to **prevent drift**
  - Can be done in 2nd thread

Send Pose



Localization Server

[Middelberg et al., ECCV'14]

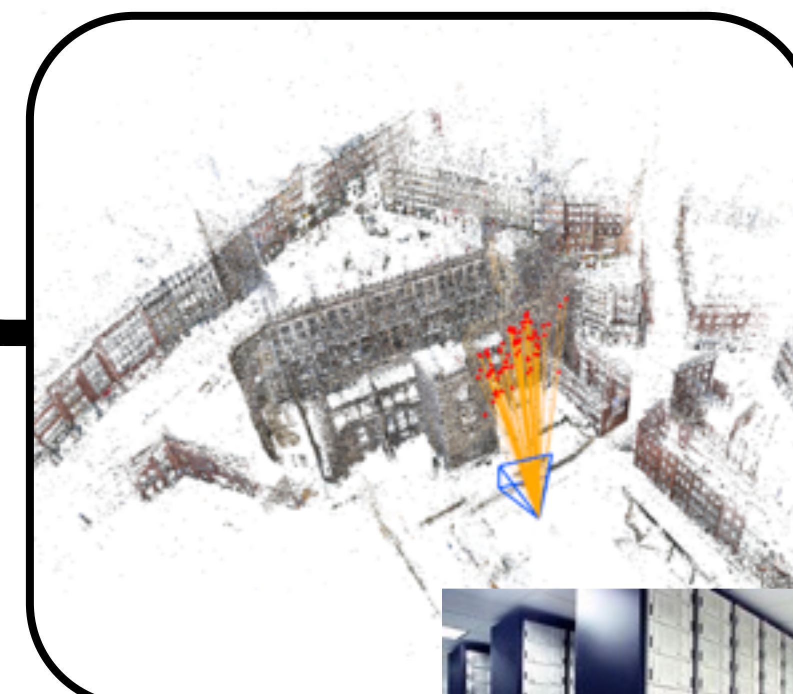
A Scalable Mobile Localization Pipeline

Mobile Device

Send Image

- Run SLAM / PTAM for **real-time camera tracking**
- Use global poses to **prevent drift**
  - Can be done in 2nd thread
- Fixed number keyframes = **fixed memory consumption**

Send Pose



Localization Server

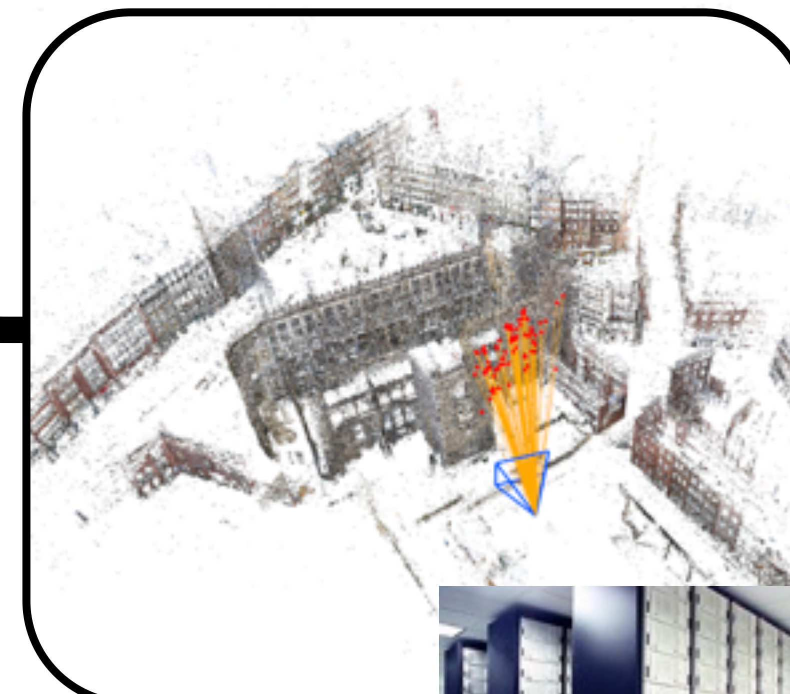
[Middelberg et al., ECCV'14]

A Scalable Mobile Localization Pipeline

Mobile Device

Send Image

Send Pose

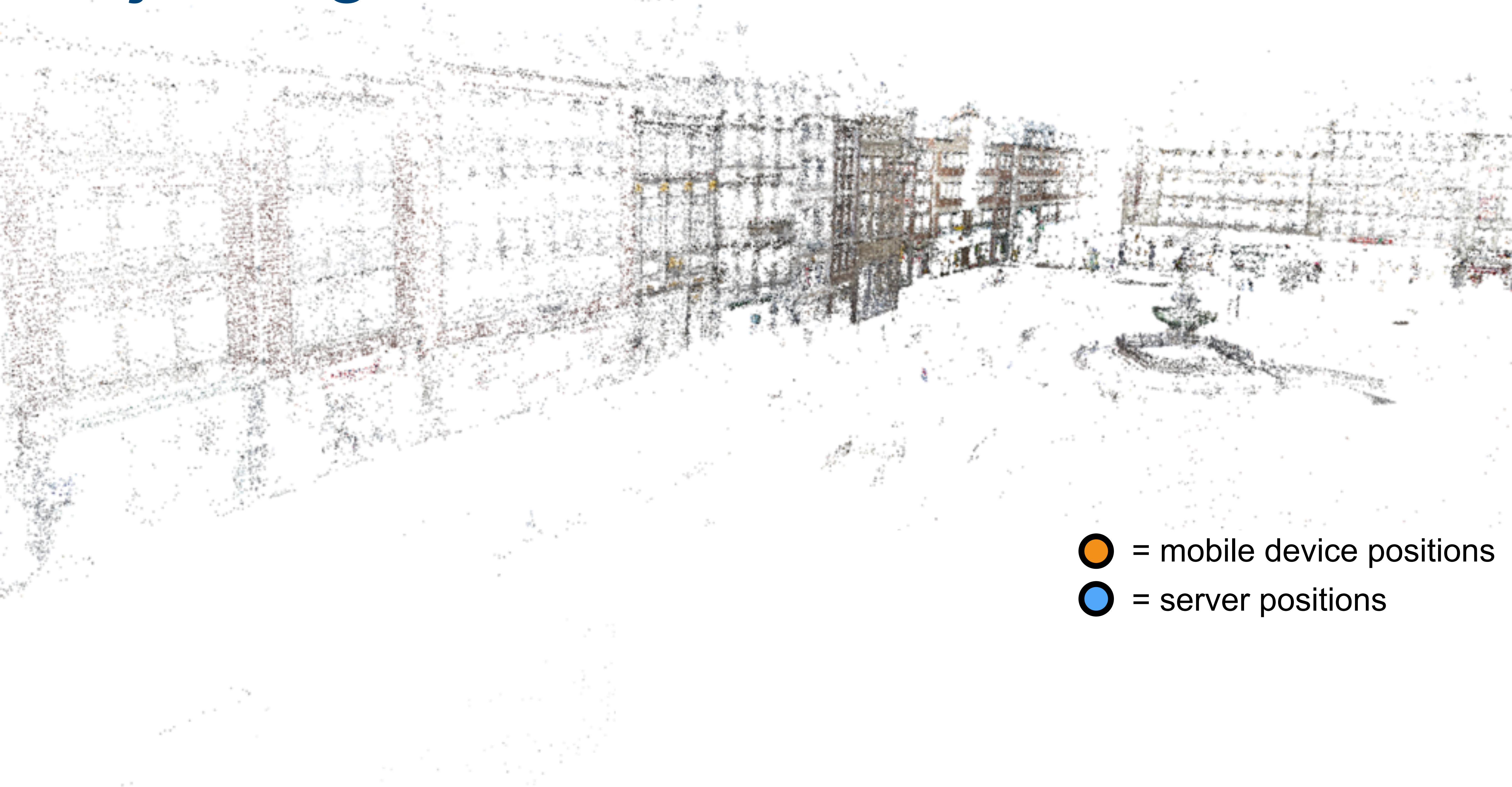


Localization Server

- Run SLAM / PTAM for **real-time camera tracking**
- Use global poses to **prevent drift**
  - Can be done in 2nd thread
- Fixed number keyframes = **fixed memory consumption**
- [SLAM code available]

[Middelberg et al., ECCV'14]

Adjusting the Model



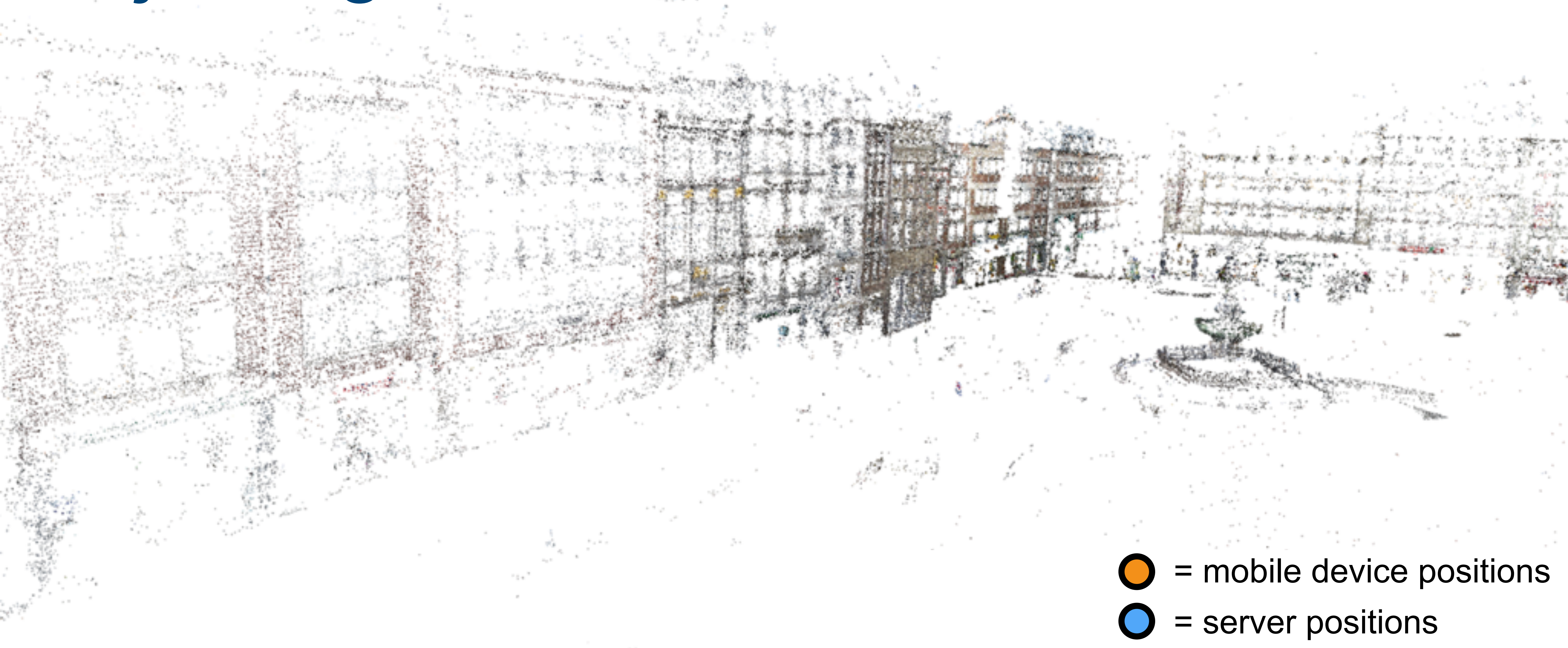
= mobile device positions

= server positions

Computer Vision and Geometry Group

ETH zürich

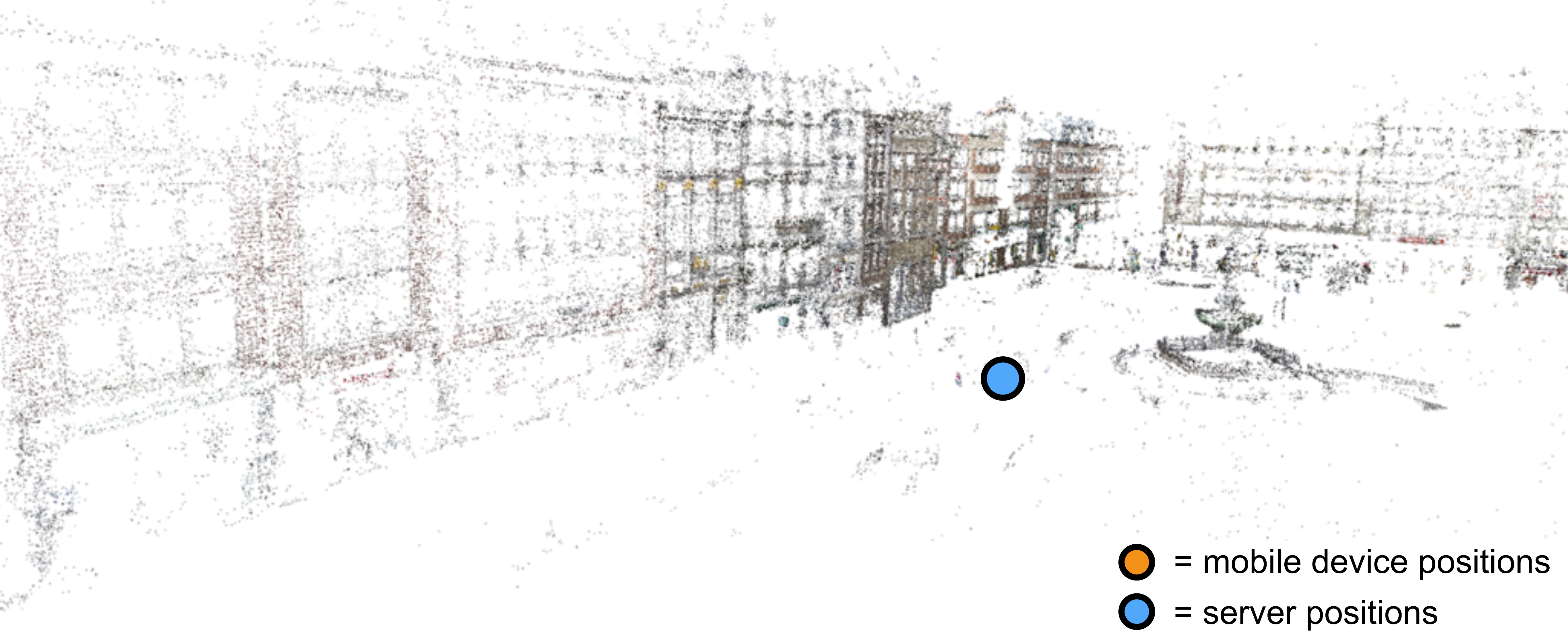
Adjusting the Model



- Initialization from first two keyframes + gravity direction

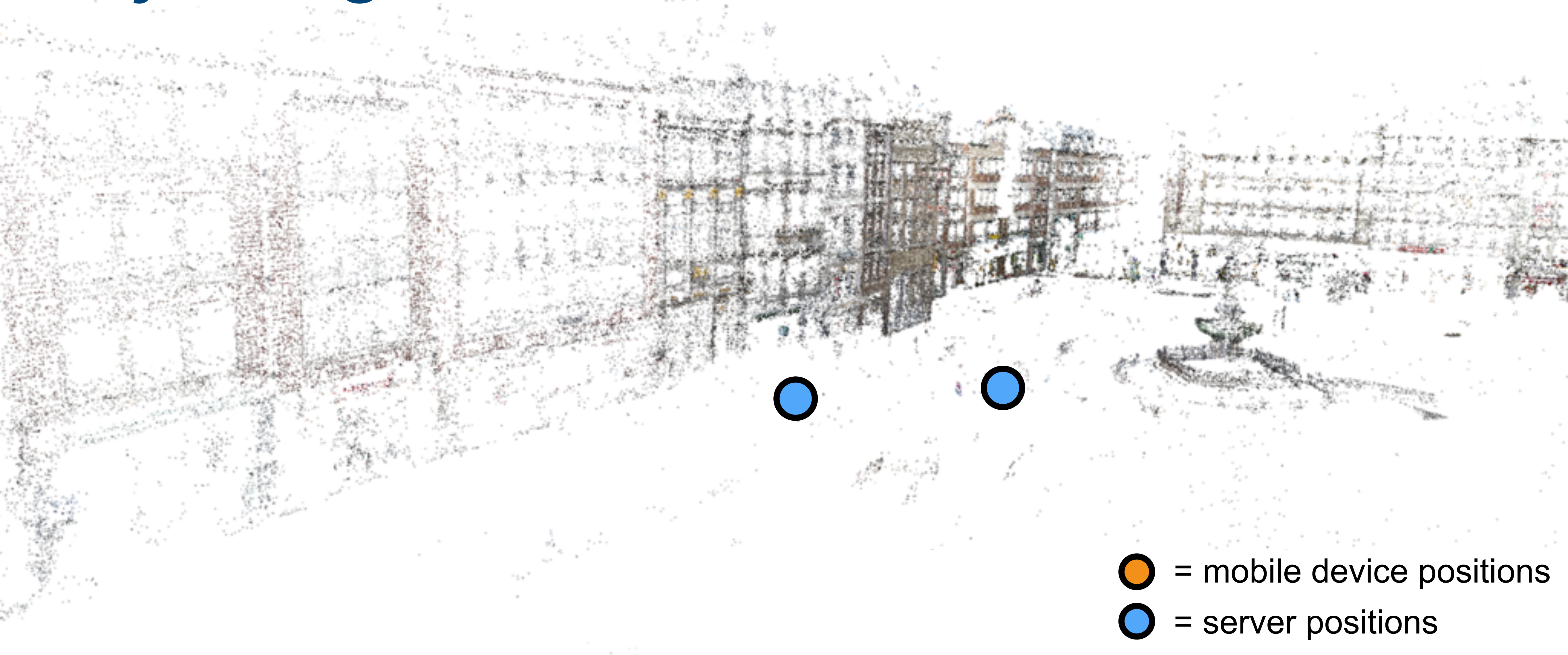
Computer Vision and Geometry Group ETH zürich

Adjusting the Model



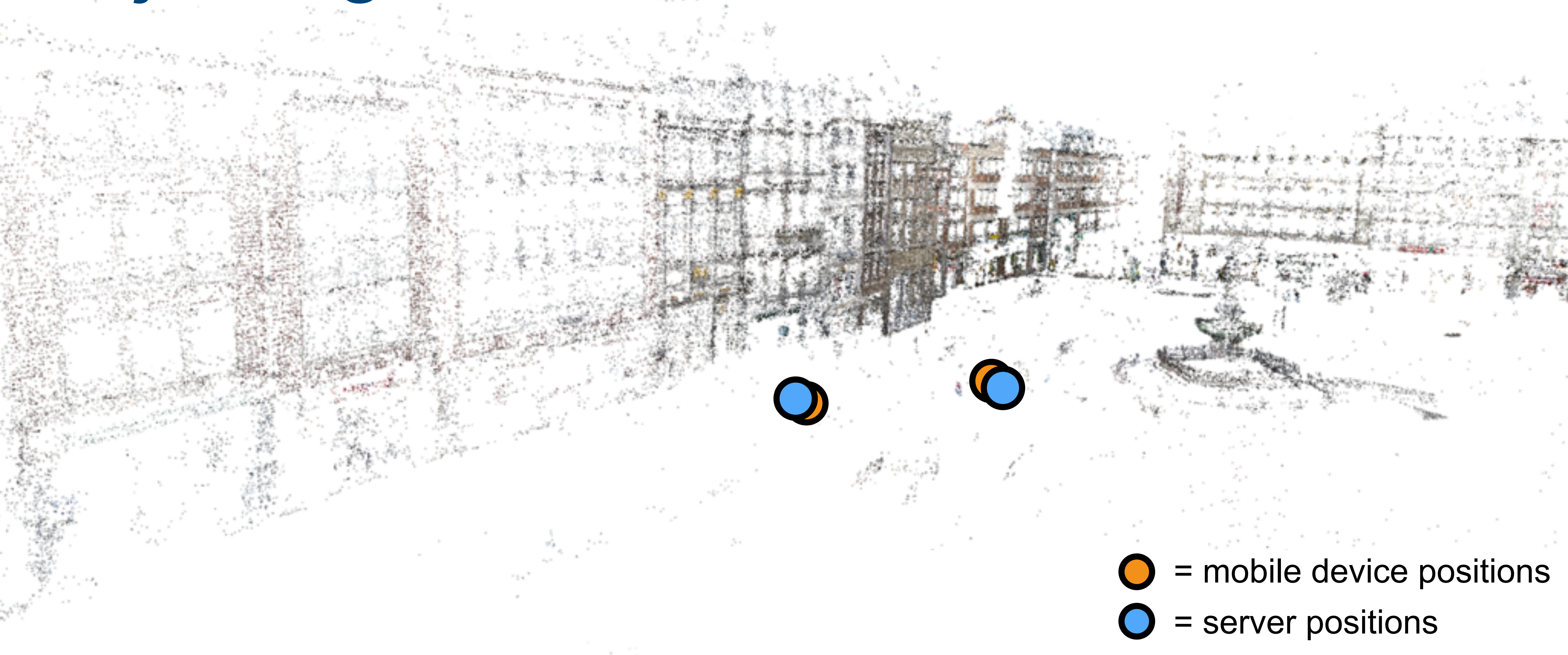
- Initialization from first two keyframes + gravity direction

Adjusting the Model



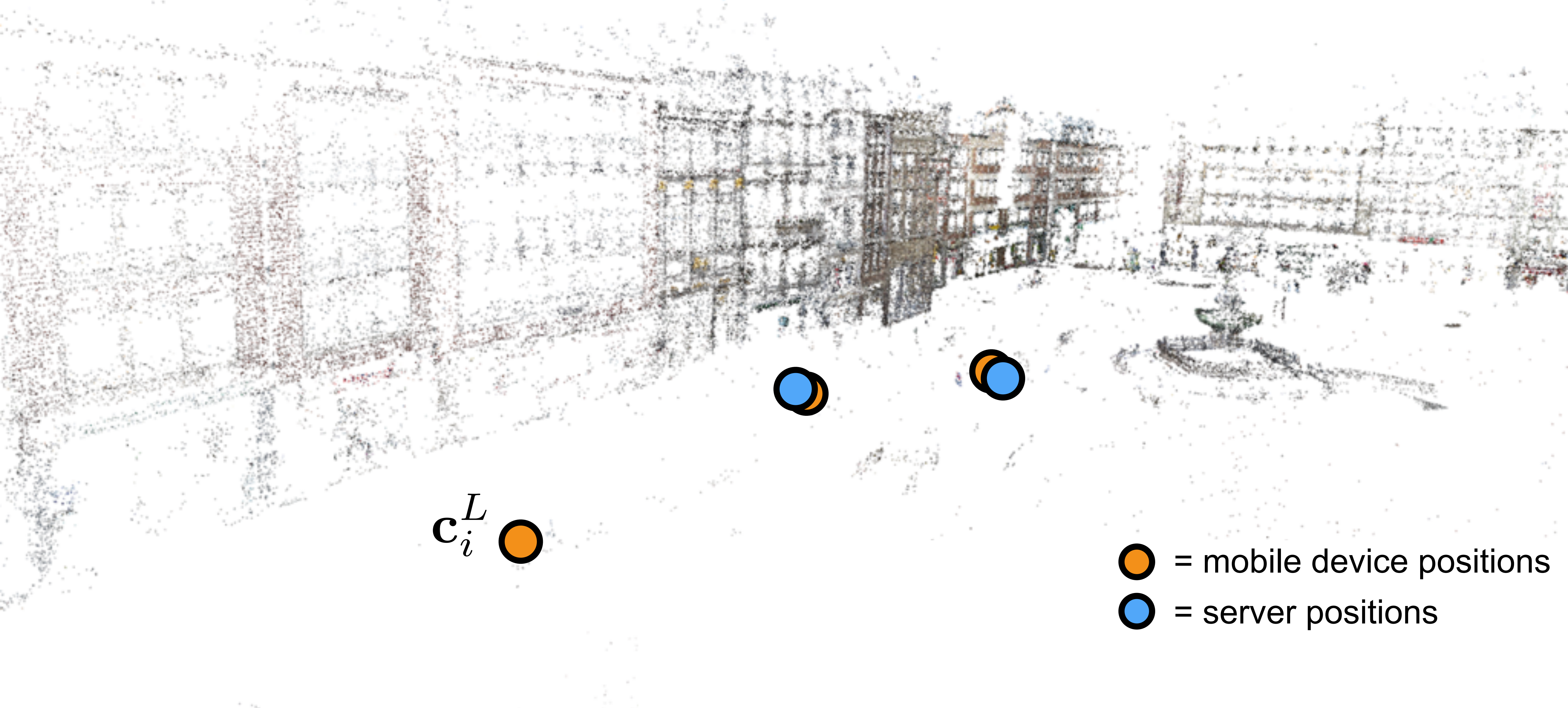
- Initialization from first two keyframes + gravity direction

Adjusting the Model



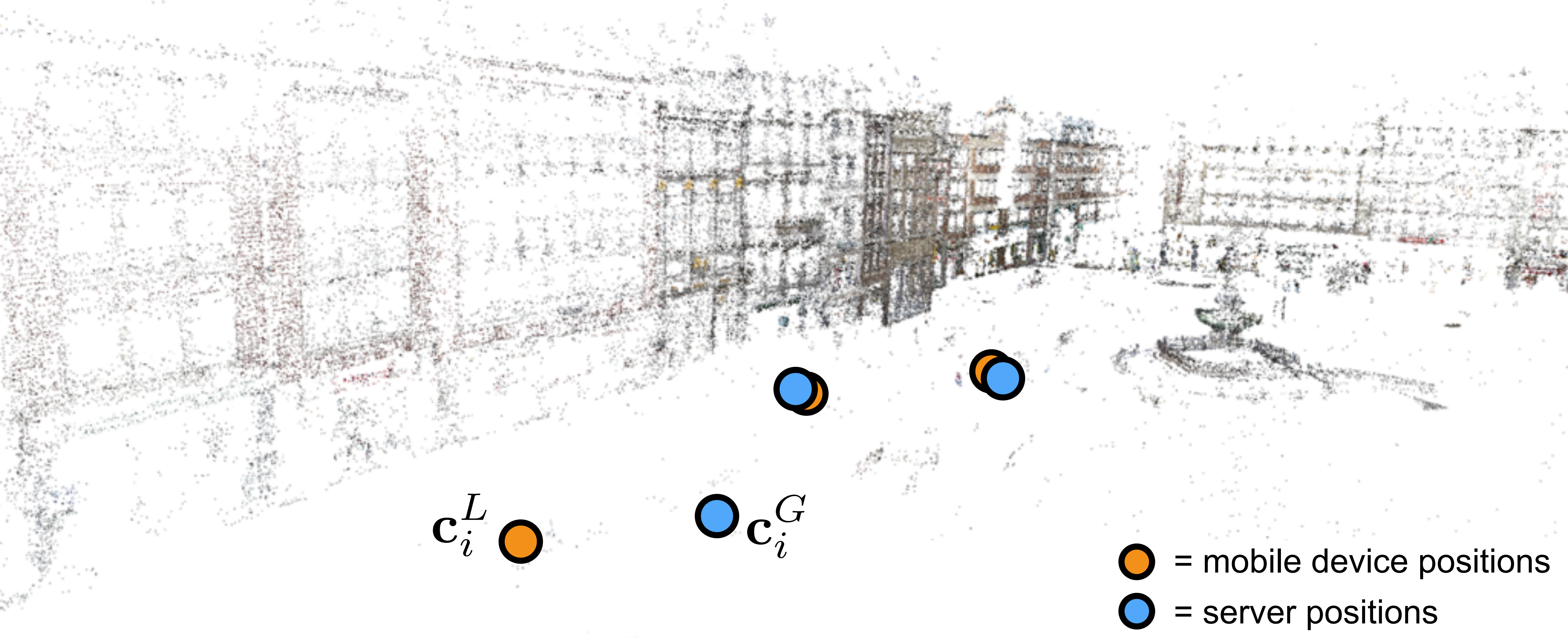
- Initialization from first two keyframes + gravity direction

Adjusting the Model



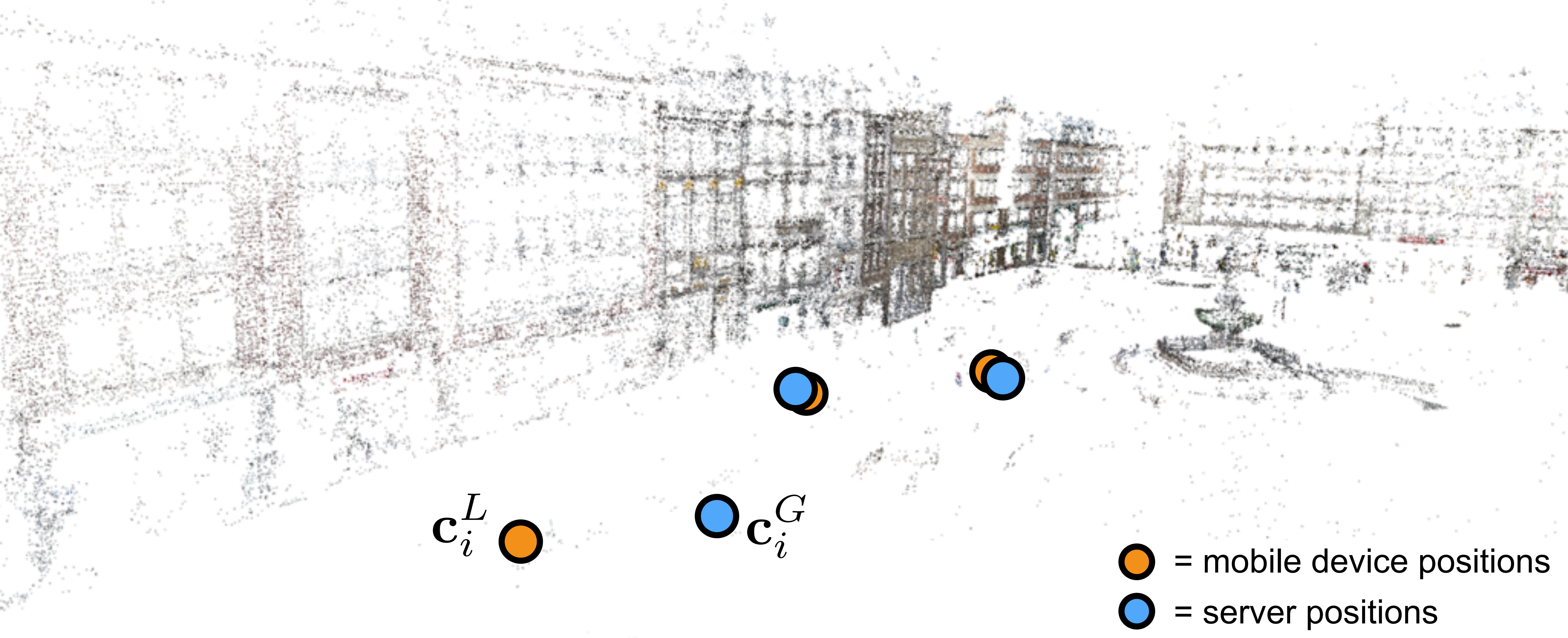
- Initialization from first two keyframes + gravity direction

Adjusting the Model



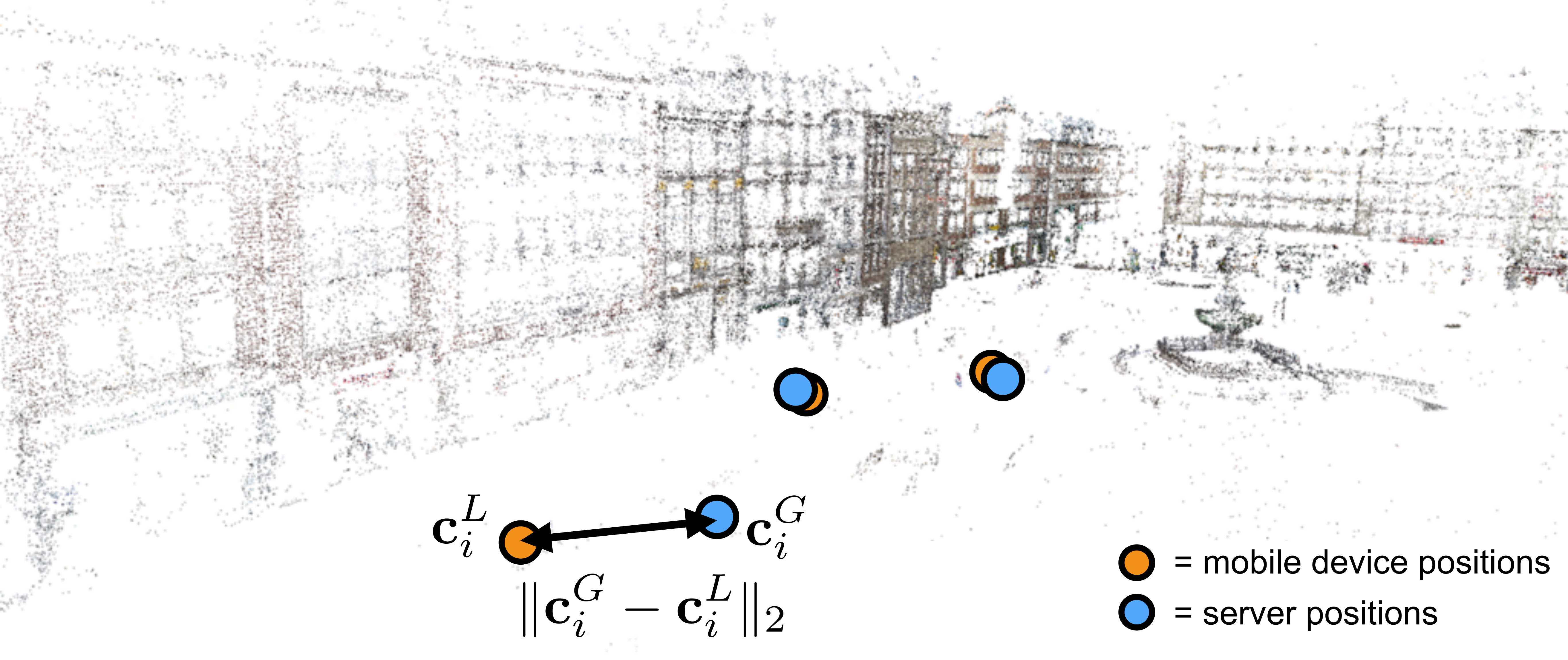
- Initialization from first two keyframes + gravity direction

Adjusting the Model



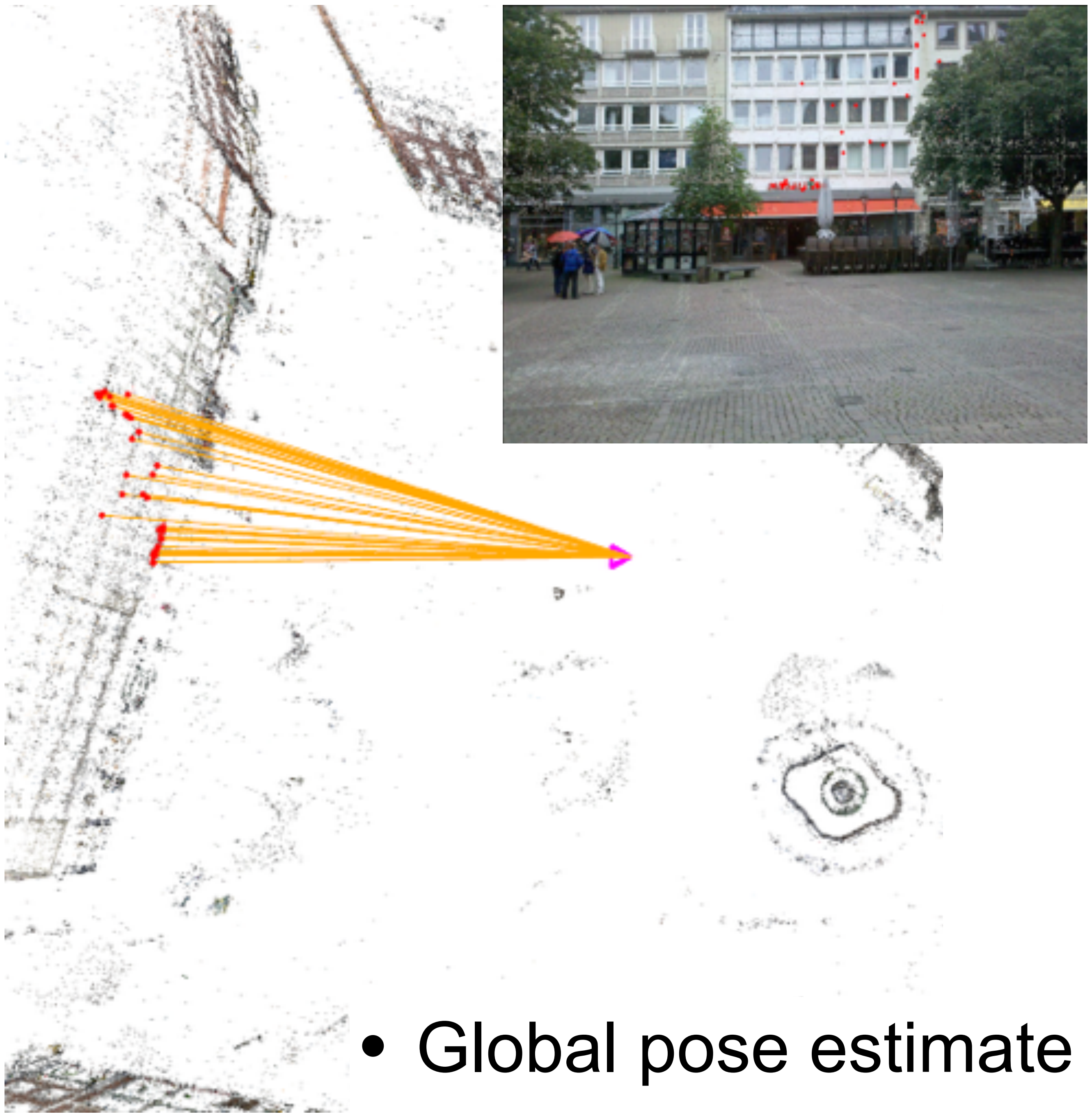
- Initialization from first two keyframes + gravity direction
- Try to minimize distance to camera positions reported by server

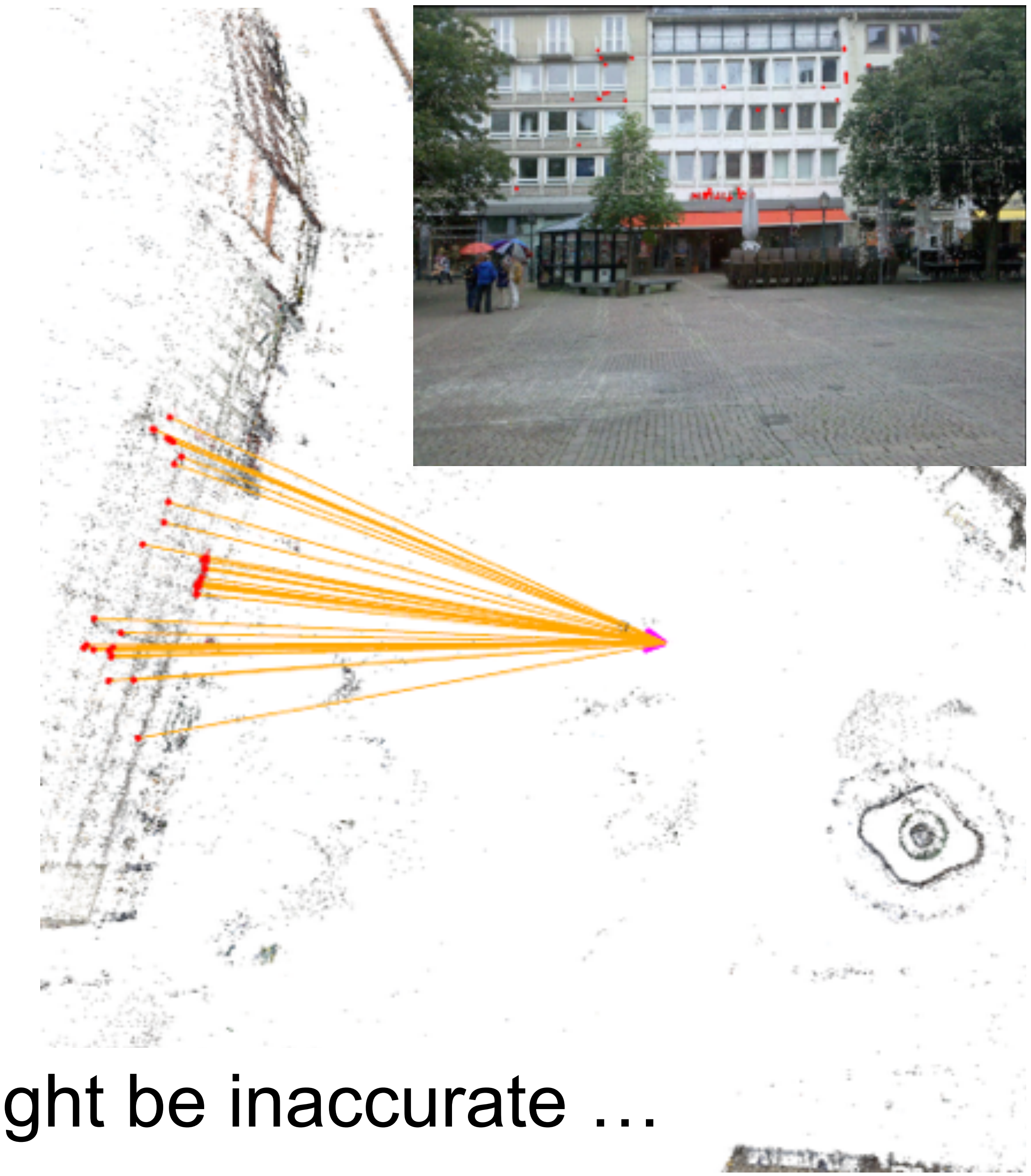
Adjusting the Model



- Initialization from first two keyframes + gravity direction
- Try to minimize distance to camera positions reported by server

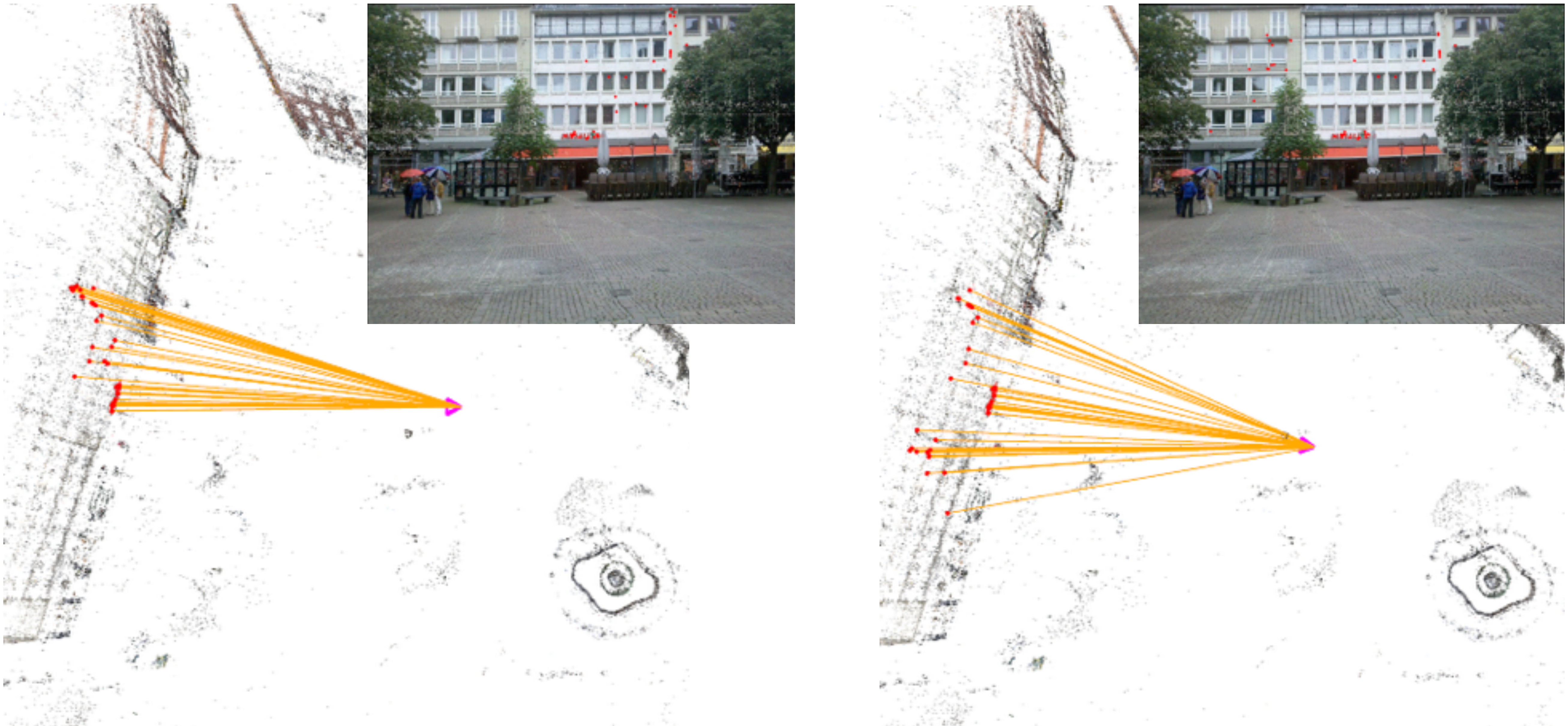
Adjusting the Model





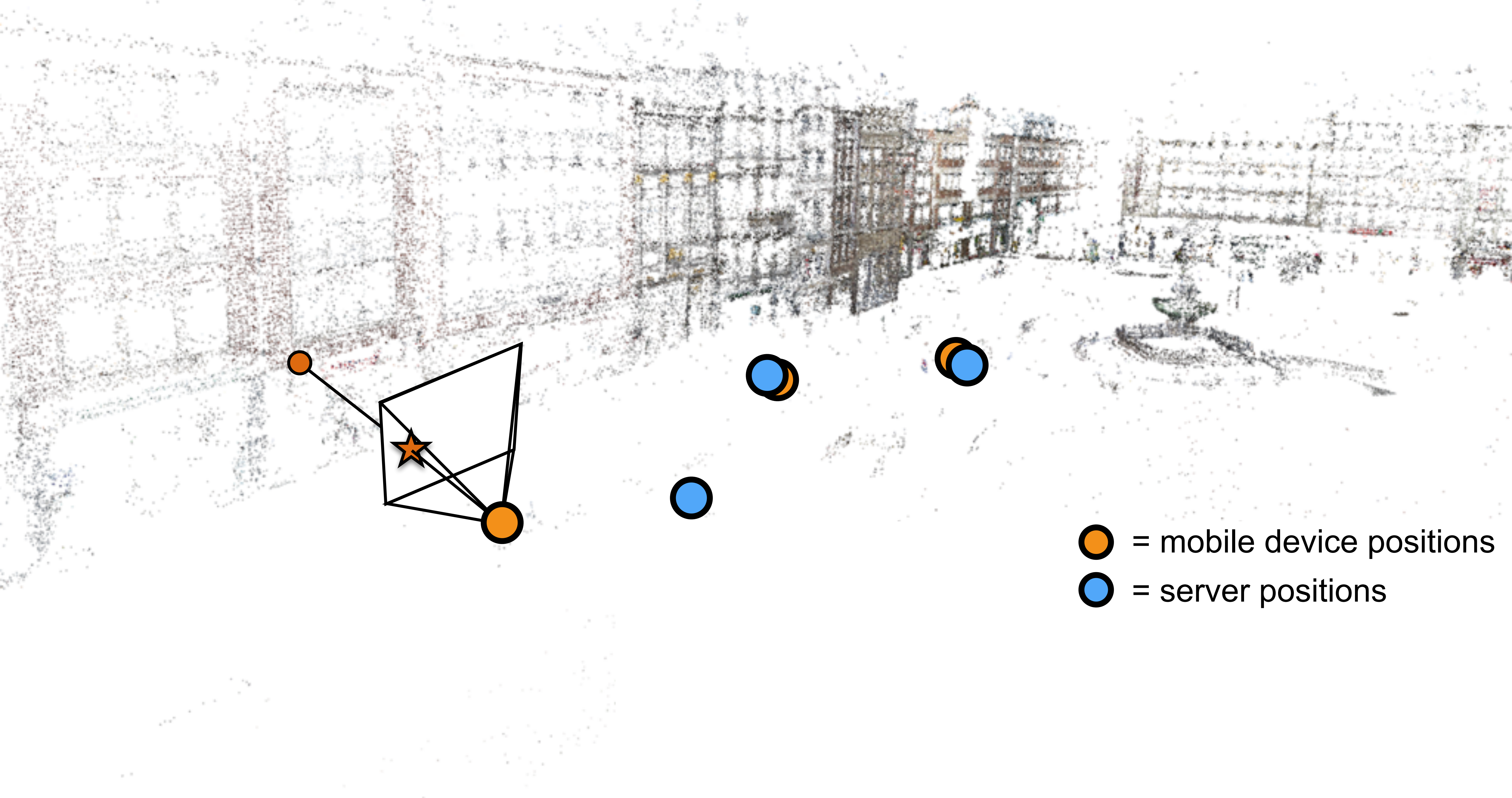
- Global pose estimate might be inaccurate …

Adjusting the Model

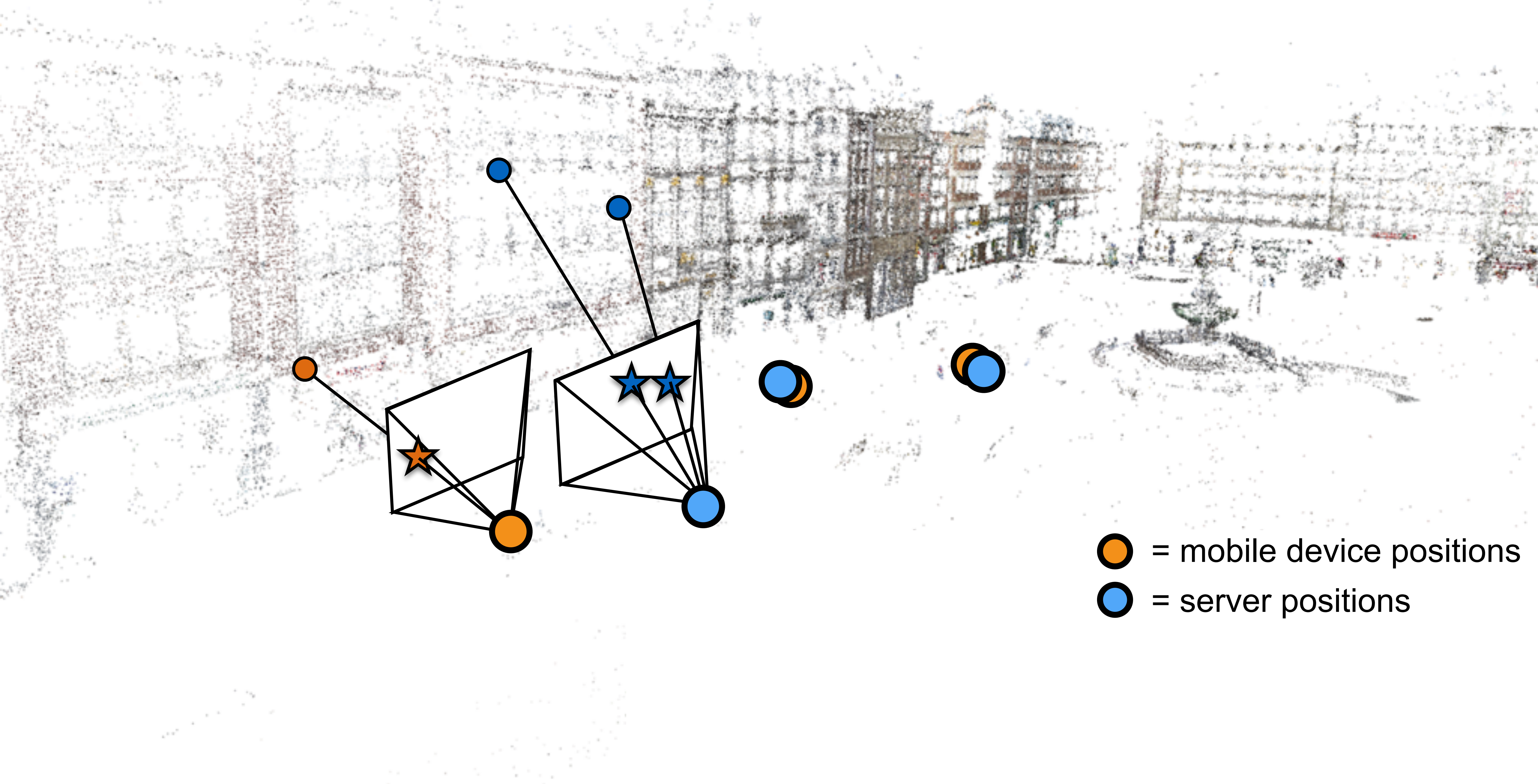


- Global pose estimate might be inaccurate …
- … but 2D-3D matches are still correct!

Adjusting the Model

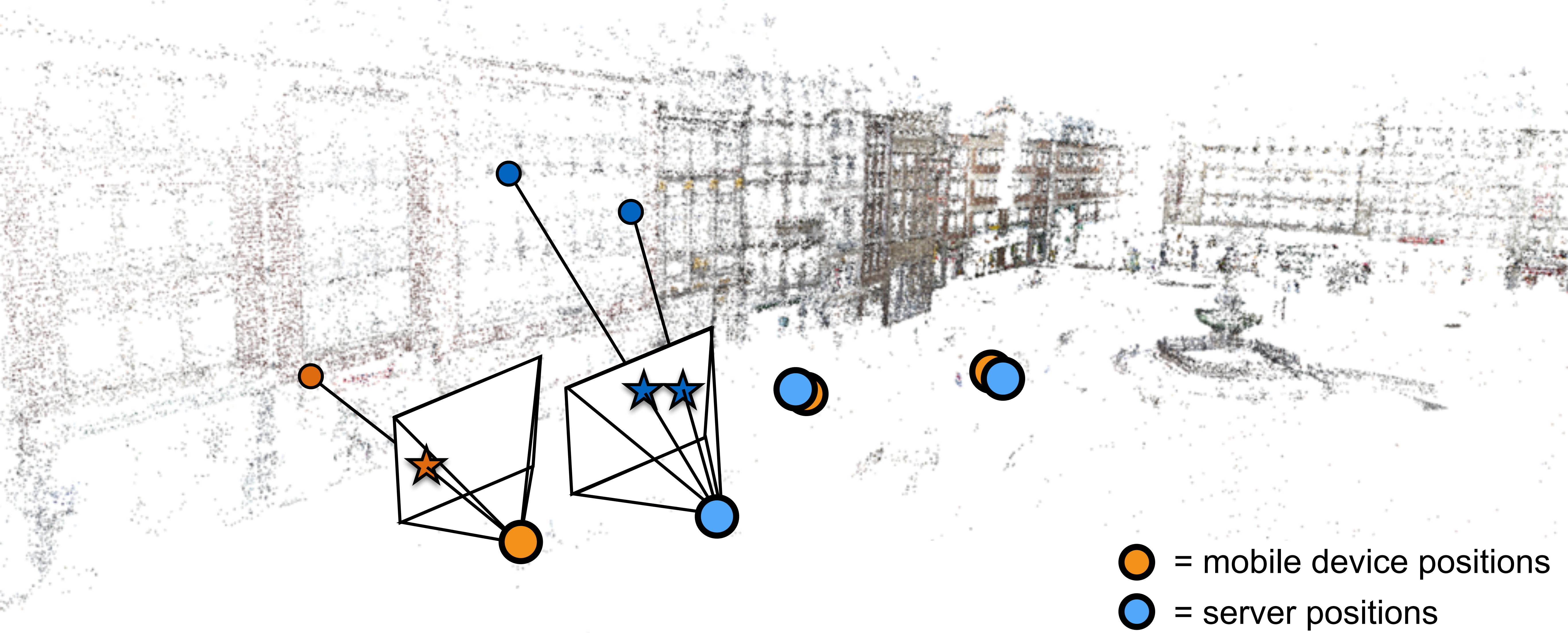


Adjusting the Model



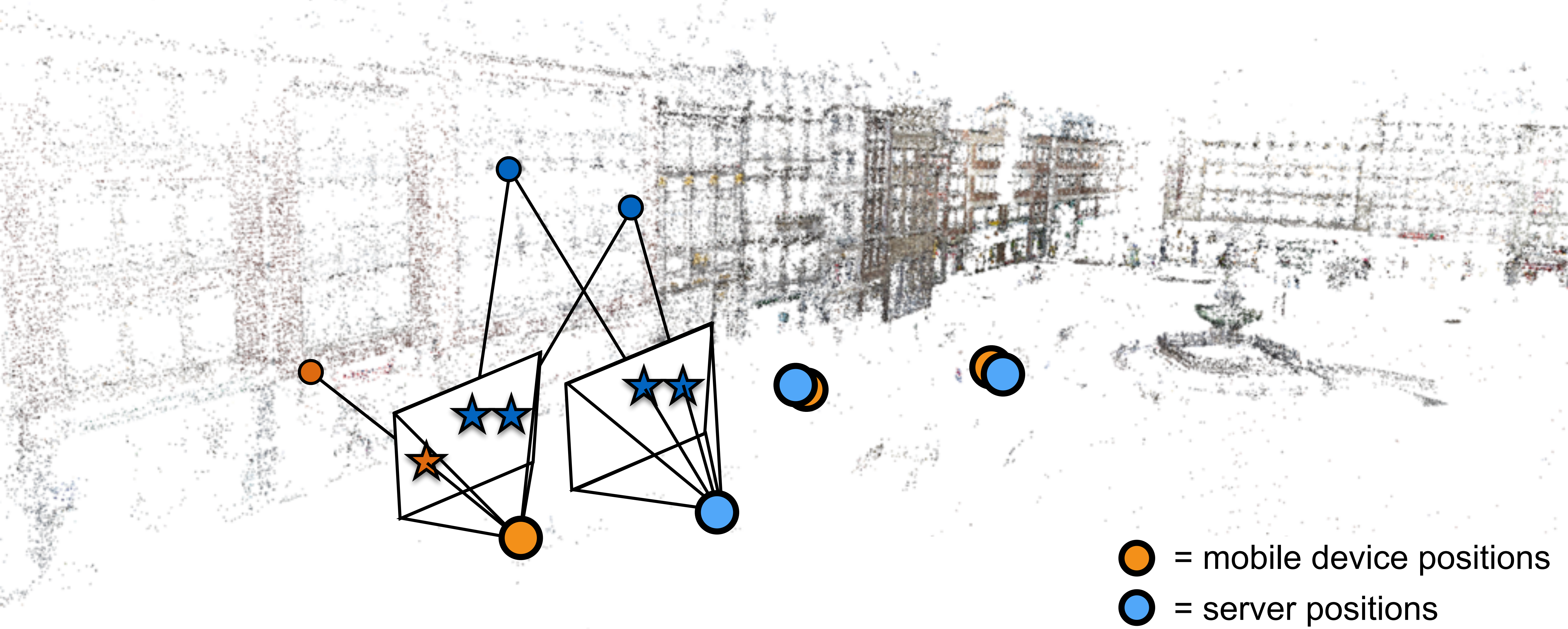
Computer Vision and Geometry Group ETH zürich

Adjusting the Model



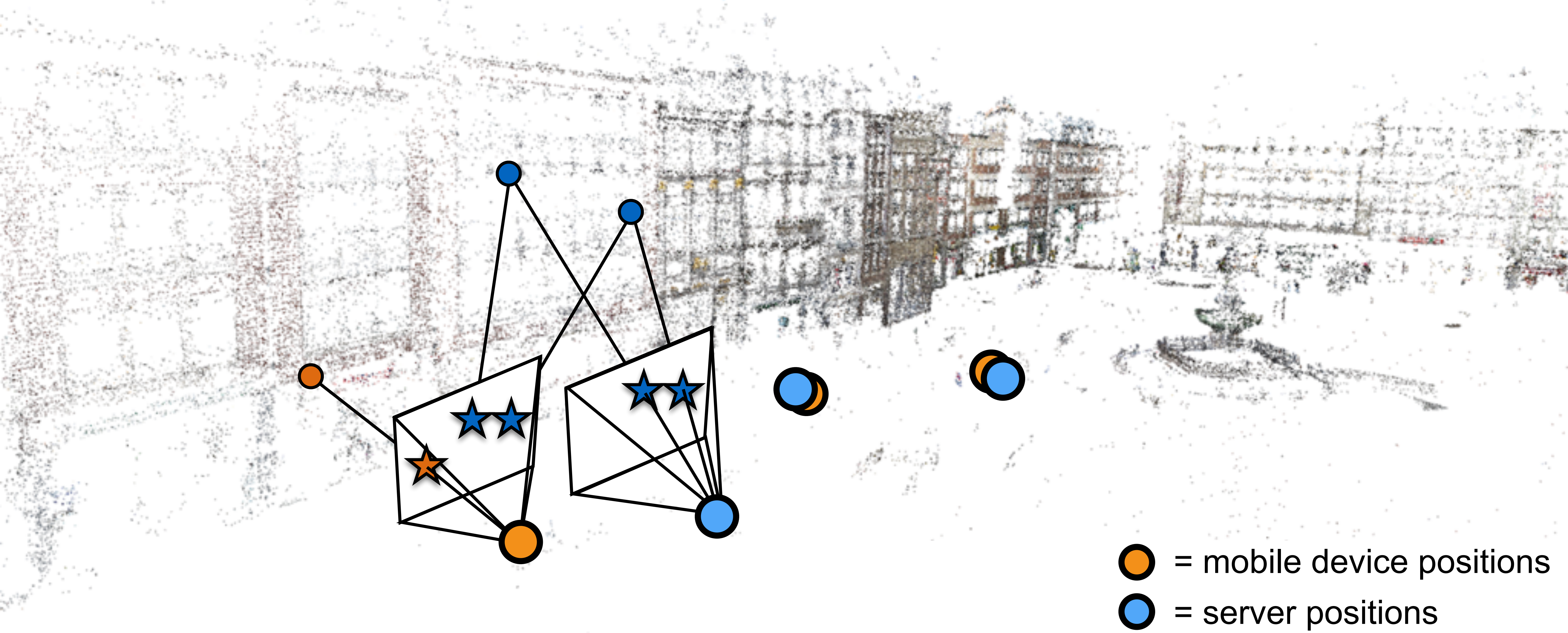
- Use 2D-3D matches from server as control points

Adjusting the Model



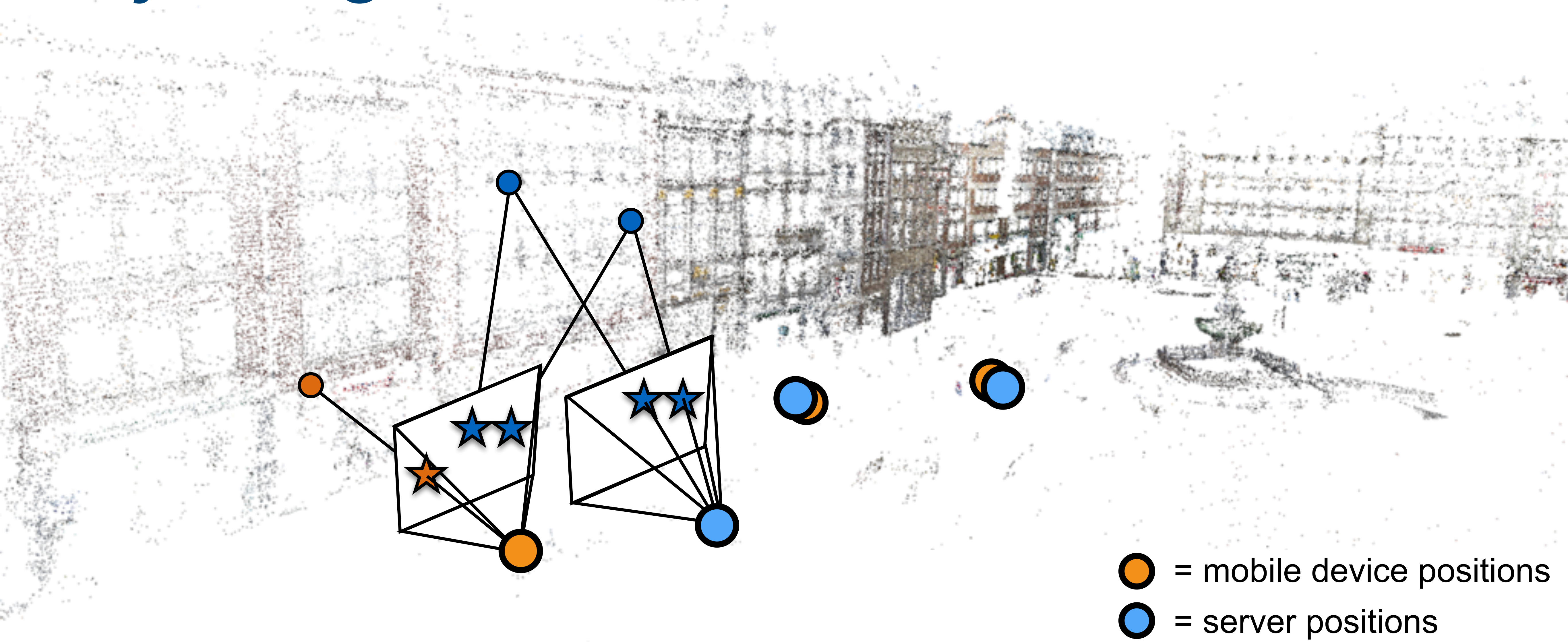
- Use 2D-3D matches from server as control points

Adjusting the Model



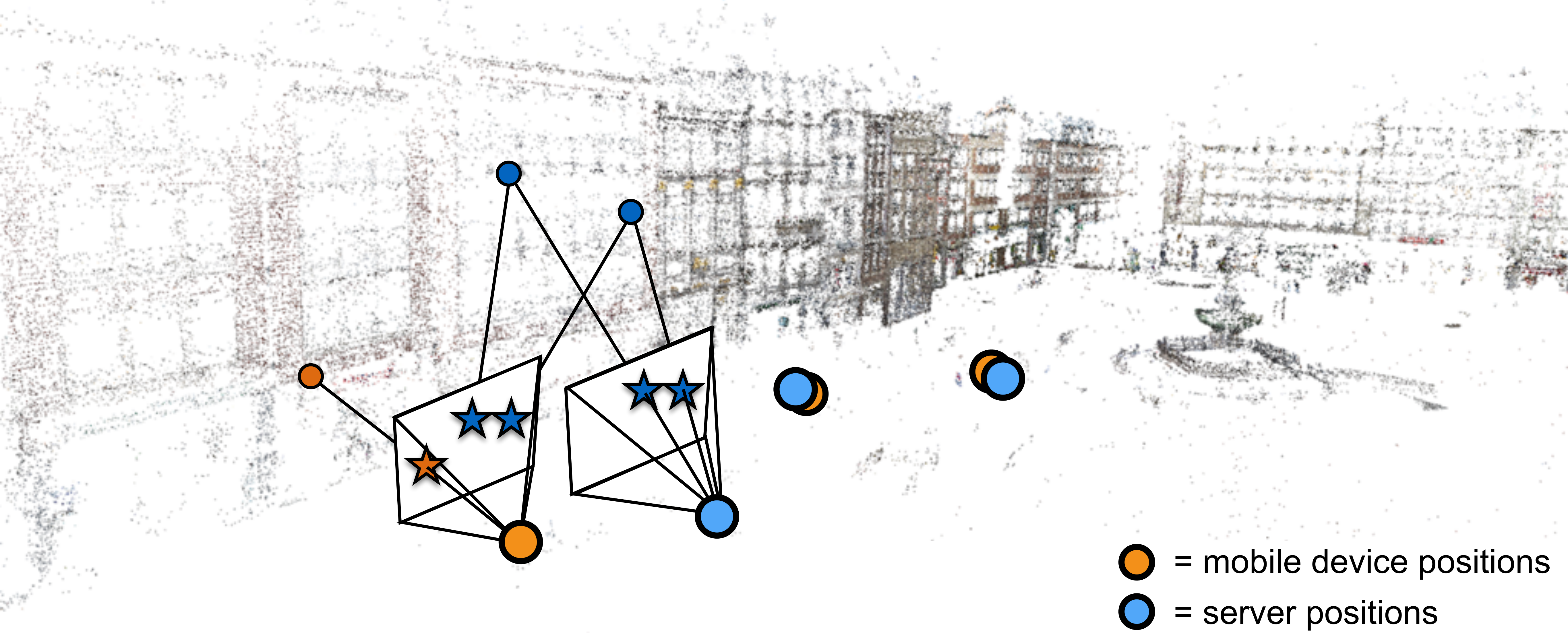
- Use 2D-3D matches from server as control points
  - Anchor local model, prevent drift

Adjusting the Model



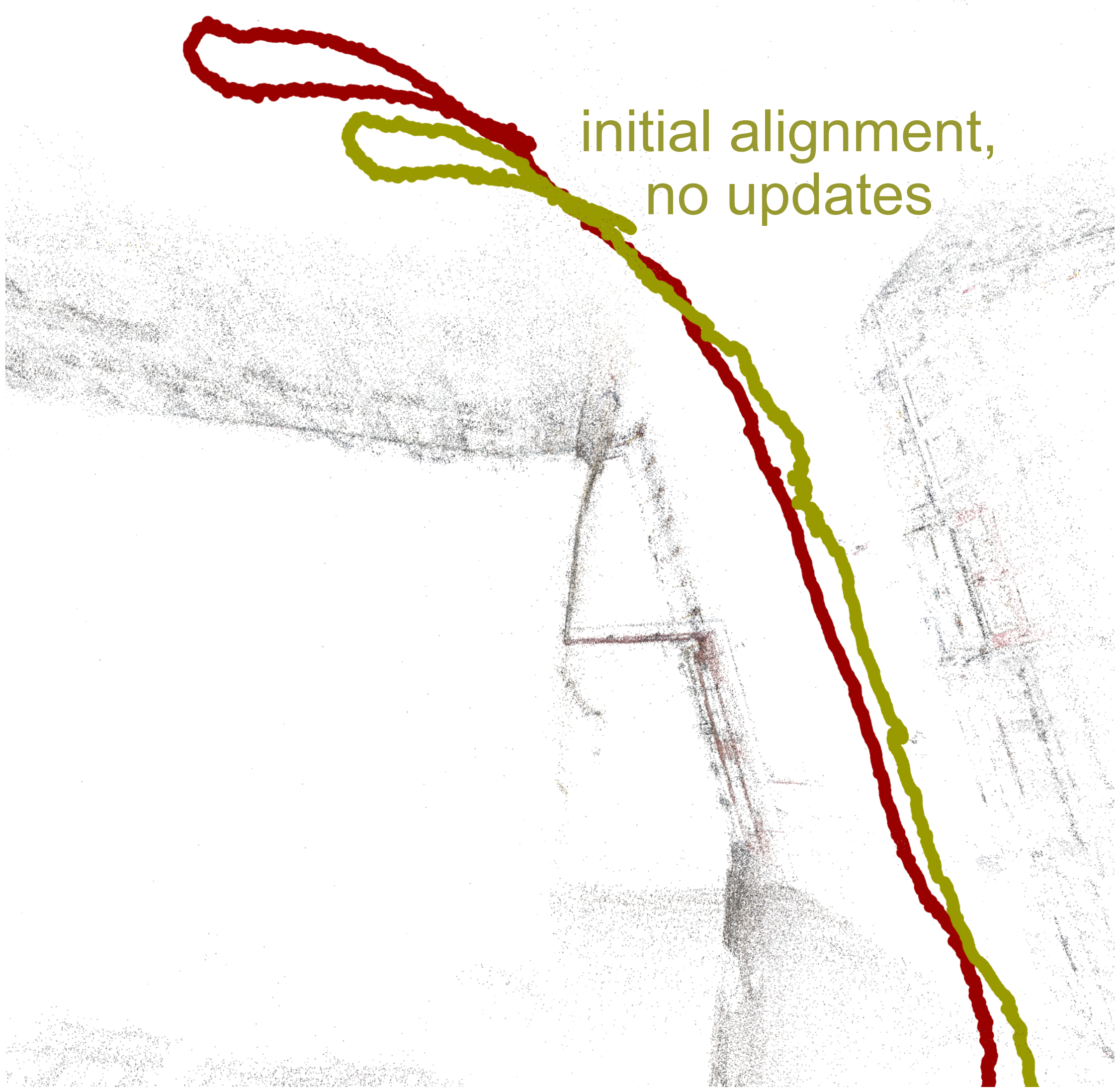
- Use 2D-3D matches from server as control points
  - Anchor local model, prevent drift
  - Little additional costs during Bundle Adjustment

Adjusting the Model

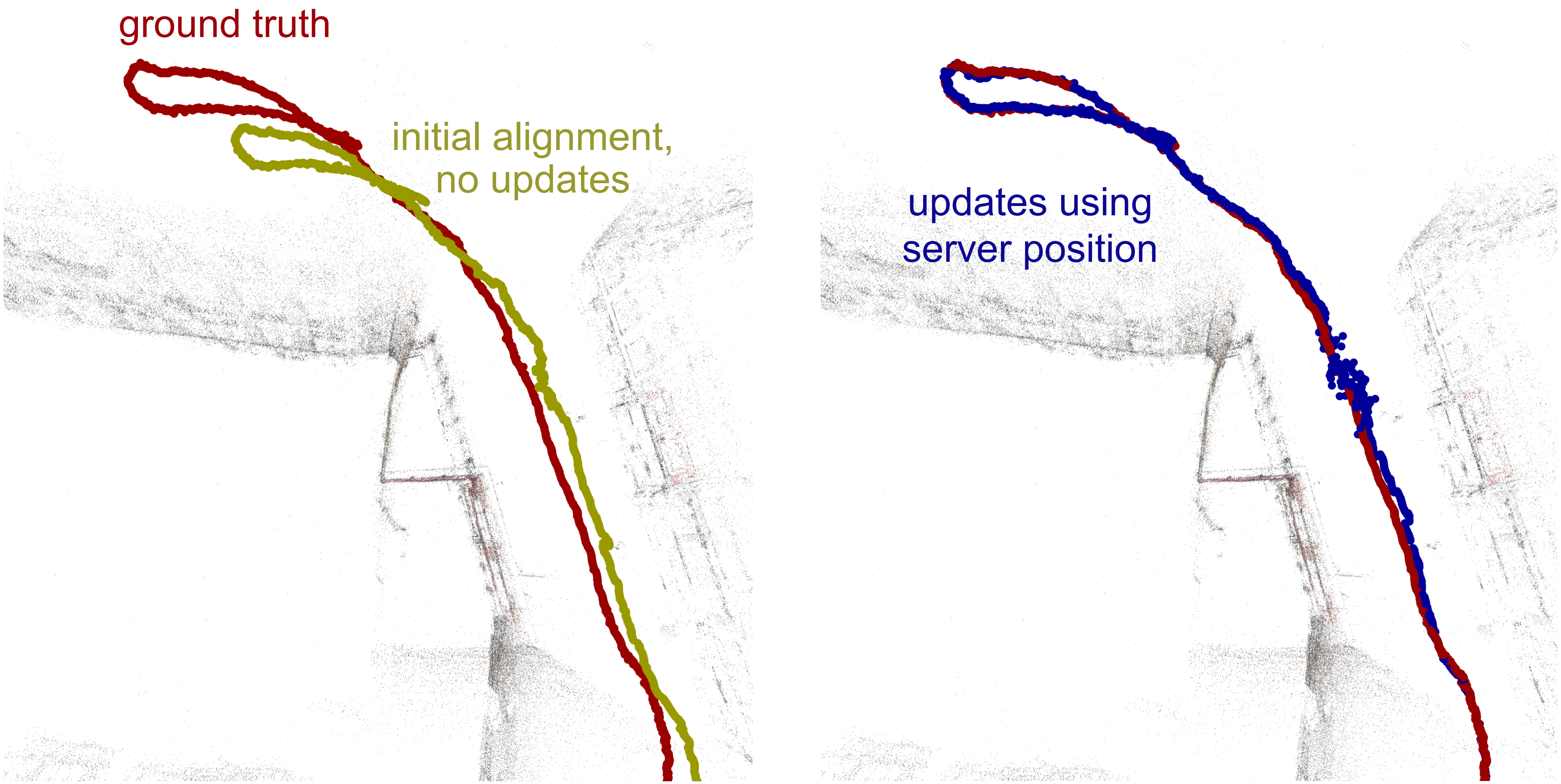


- Use 2D-3D matches from server as control points
  - Anchor local model, prevent drift
  - Little additional costs during Bundle Adjustment
- Still need weighting since fewer matches from server

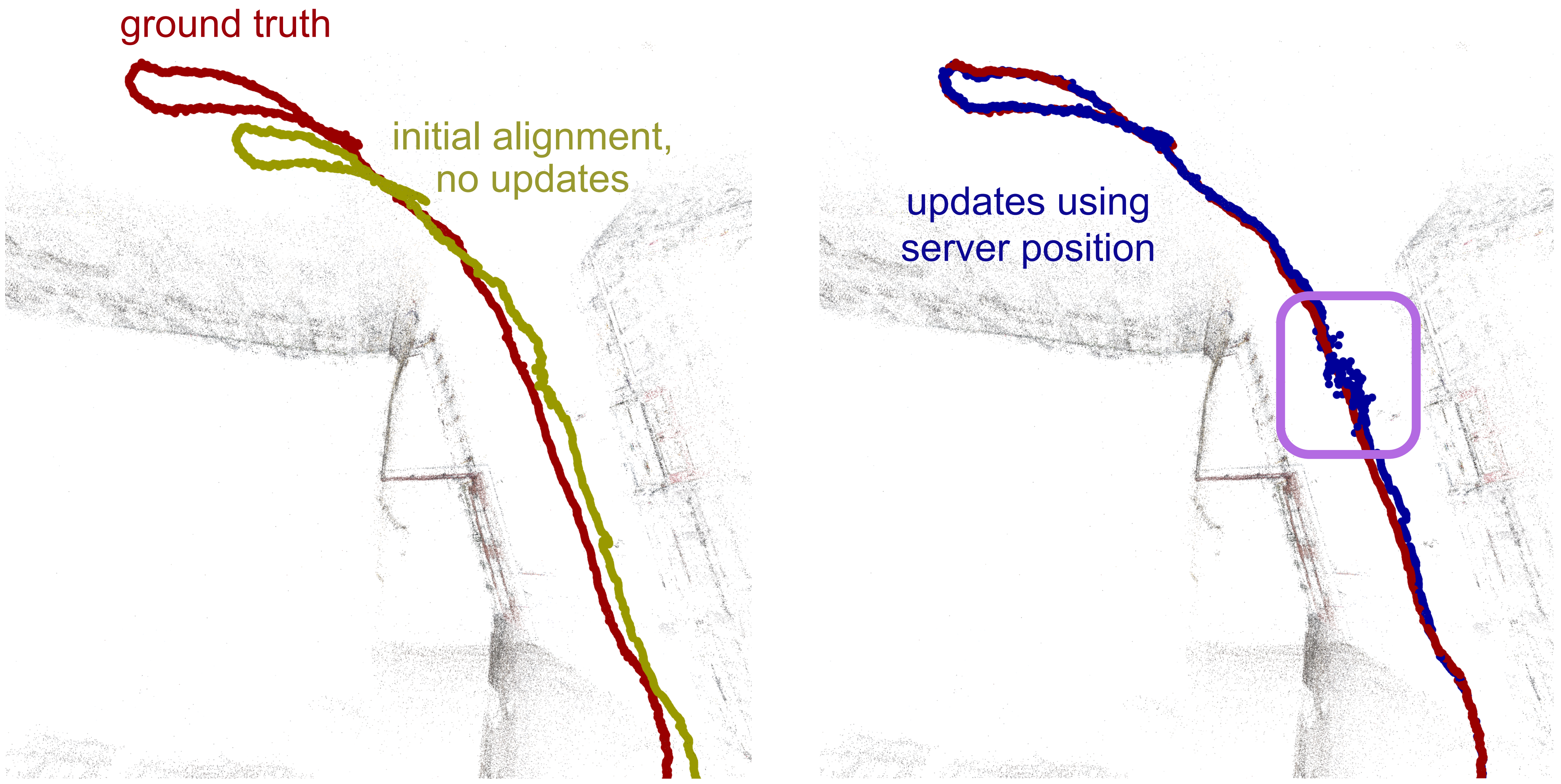
Results



Results

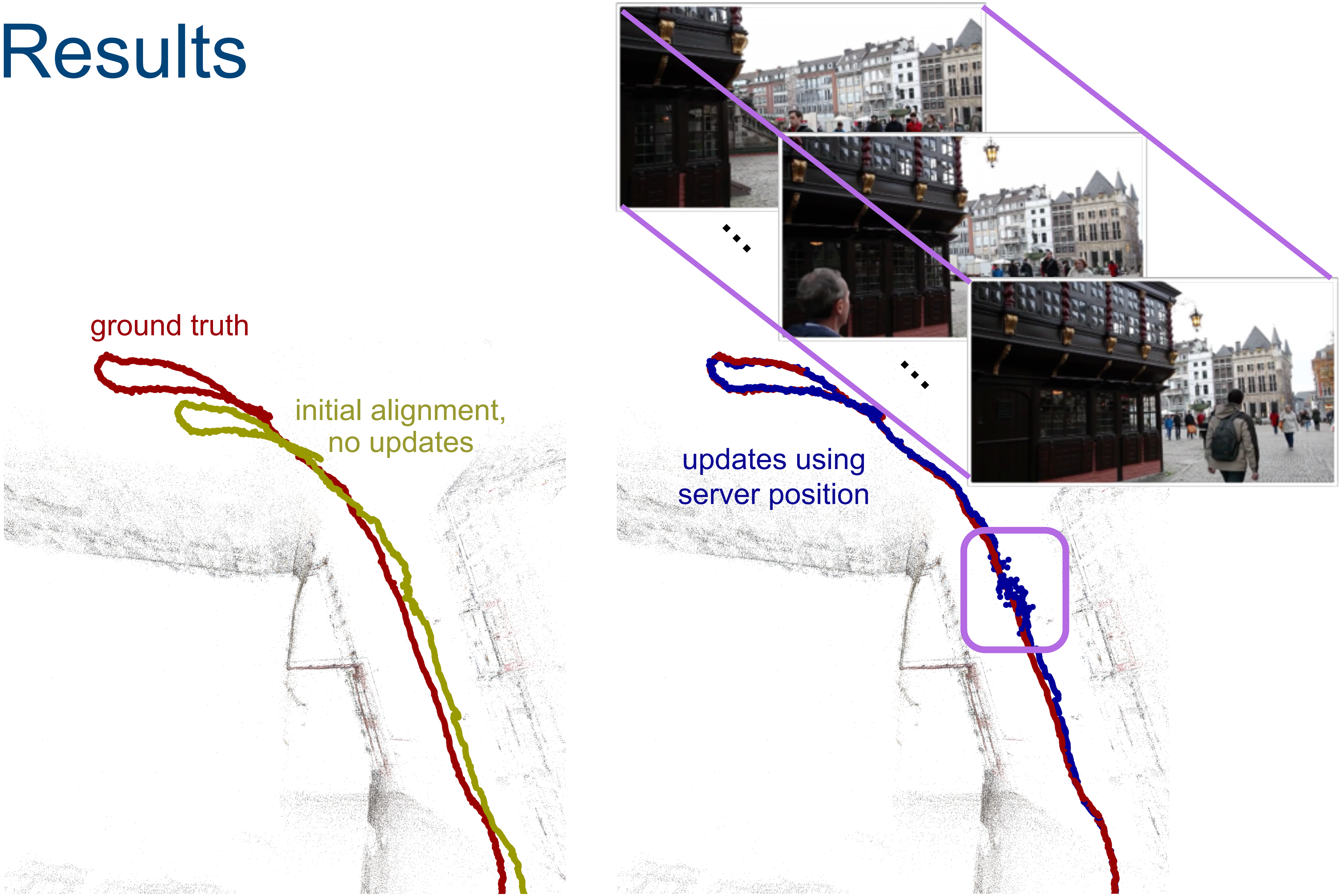

ground truth
initial alignment,
no updates
updates using
server position

Results



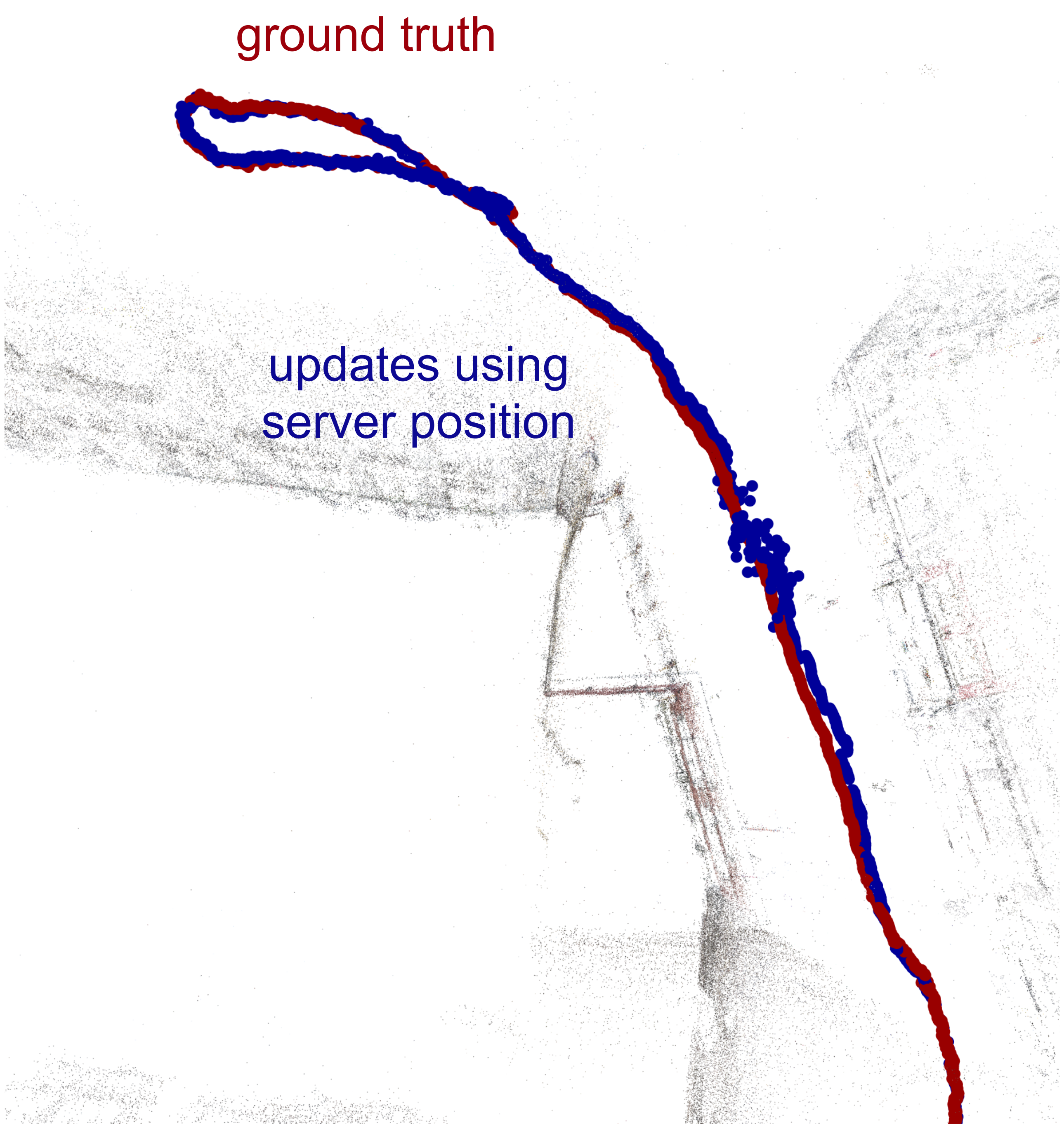
Computer Vision
and Geometry Group
ETH zürich

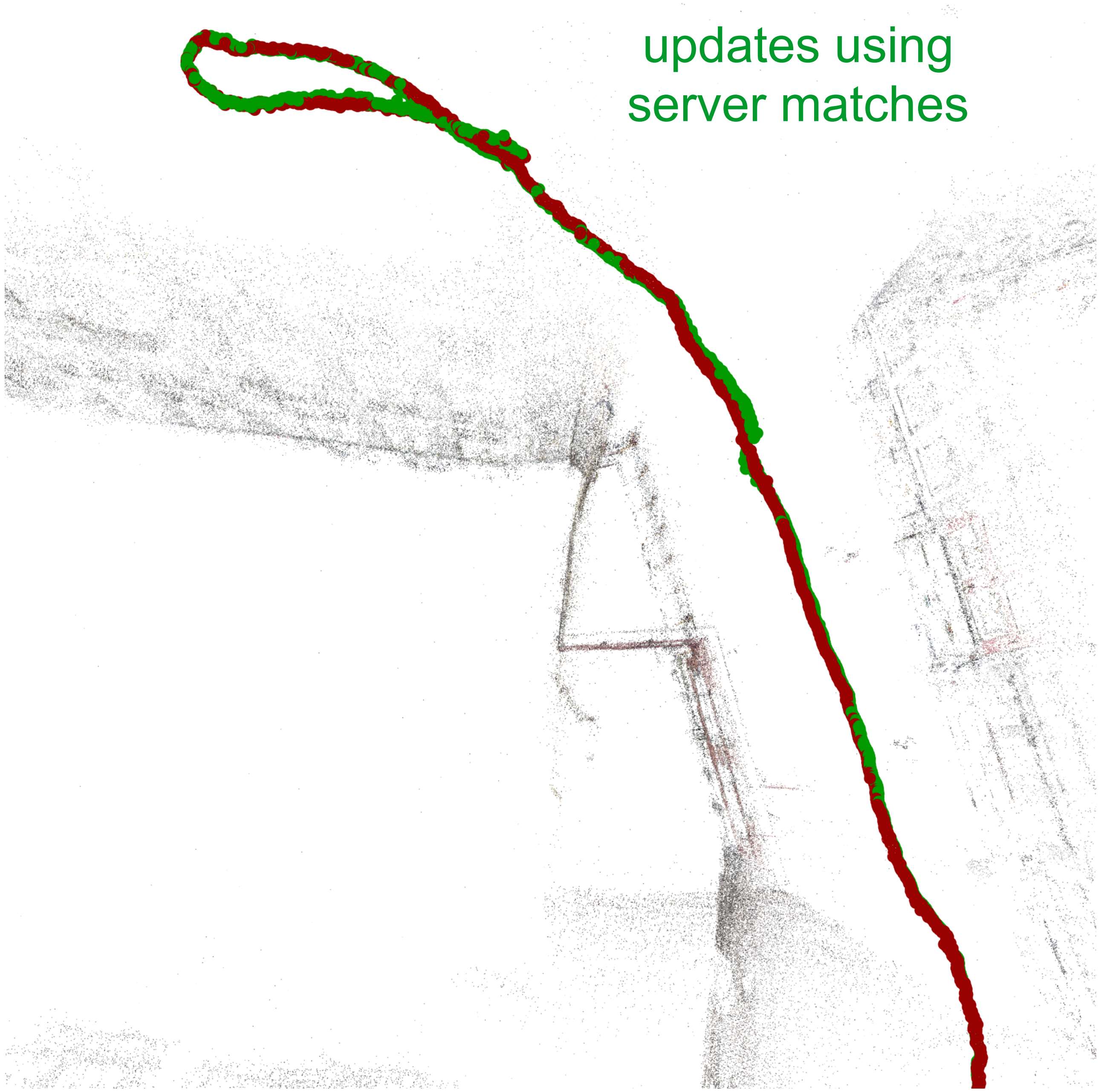
Results



ground truth
initial alignment,
no updates
updates using
server position

Results


ground truth
updates using
server position


updates using
server matches

Summary

Summary

Summary

- Scalability from Server-Client architecture

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm
  - Average FPS: 18 (~55-60ms per frame)

Computer Vision and Geometry Group ETH zürich

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm
  - Average FPS: 18 (~55-60ms per frame)
  - Memory consumption for local map: 5MB

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm
  - Average FPS: 18 (~55-60ms per frame)
  - Memory consumption for local map: 5MB
  - [Source code for local SLAM system available]

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm
  - Average FPS: 18 (~55-60ms per frame)
  - Memory consumption for local map: 5MB
  - [Source code for local SLAM system available]

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm
  - Average FPS: 18 (~55-60ms per frame)
  - Memory consumption for local map: 5MB
  - [Source code for local SLAM system available]

- Significant room for improvement:

Computer Vision and Geometry Group ETH zürich

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm
  - Average FPS: 18 (~55-60ms per frame)
  - Memory consumption for local map: 5MB
  - [Source code for local SLAM system available]
- Significant room for improvement:
  - Priors for server-side localization

Computer Vision and Geometry Group ETH zürich

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm
  - Average FPS: 18 (~55-60ms per frame)
  - Memory consumption for local map: 5MB
  - [Source code for local SLAM system available]

- Significant room for improvement:
  - Priors for server-side localization
  - Visual Inertial Odometry for mobile camera tracking

Summary

- Scalability from Server-Client architecture
- Use 2D-3D matches from server to stabilize local SLAM system
- Current results (iPad Mini 2nd Generation):
  - Localization error < 50 cm
  - Average FPS: 18 (~55-60ms per frame)
  - Memory consumption for local map: 5MB
  - [Source code for local SLAM system available]
- Significant room for improvement:
  - Priors for server-side localization
  - Visual Inertial Odometry for mobile camera tracking
  - Semi-Dense SLAM

Overview

- Efficient & Effective Large-Scale Localization
- Real-Time Mobile Localization
- Open Challenges

The Ratio Test Problem

© Google

The Ratio Test Problem

© Google

- Larger models contain more locally similar structures

The Ratio Test Problem

© Google

- Larger models contain more locally similar structures

  ➡Ratio test for 2D-to-3D search rejects more and more matches

The Ratio Test Problem

© Google

- Larger models contain more locally similar structures
  - ➡Ratio test for 2D-to-3D search rejects more and more matches
- Happens already for Landmarks 1k dataset

The Ratio Test Problem

- Two possible solutions:

The Ratio Test Problem

- Two possible solutions:
  - Image Retrieval: No ratio test required during voting

The Ratio Test Problem

- Two possible solutions:
  - Image Retrieval: No ratio test required during voting
    - Need to consider too many top-ranked images for large models

The Ratio Test Problem

- Two possible solutions:
  - Image Retrieval: No ratio test required during voting
    - Need to consider too many top-ranked images for large models
  - Use full descriptors & relax matching criterion

The Ratio Test Problem

- Two possible solutions:
  - Image Retrieval: No ratio test required during voting
    - Need to consider too many top-ranked images for large models
  - Use full descriptors & relax matching criterion
    - Need to handle higher outlier ratios (>99%)

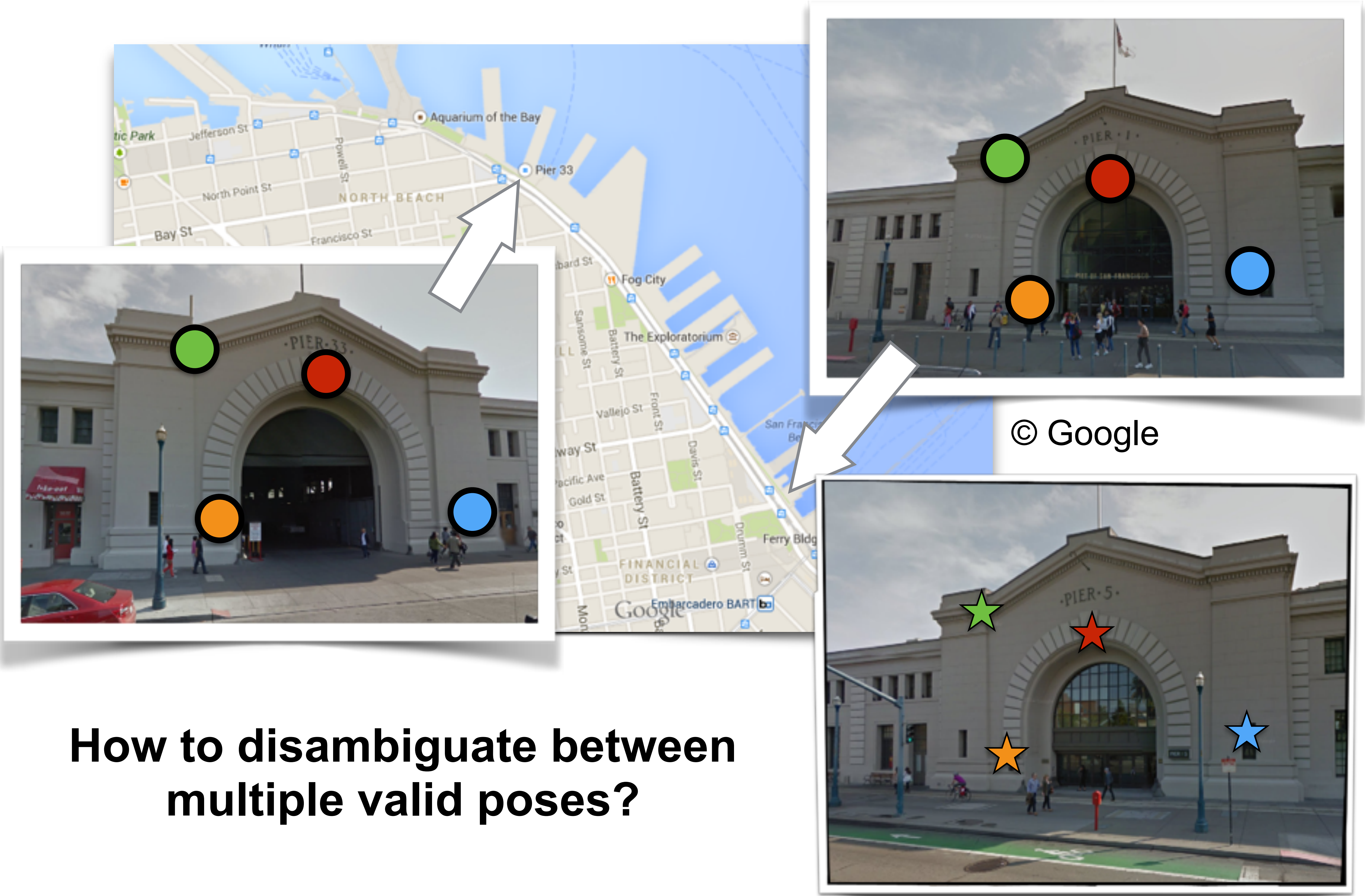
The Ratio Test Problem

- Two possible solutions:
  - Image Retrieval: No ratio test required during voting
    - Need to consider too many top-ranked images for large models
  - Use full descriptors & relax matching criterion
    - Need to handle higher outlier ratios (>99%)
    - Promising results: [Li et al., ECCV'12] [Svärm et al., CVPR'14]

The Ratio Test Problem

- Two possible solutions:
  - Image Retrieval: No ratio test required during voting
    - Need to consider too many top-ranked images for large models
  - Use full descriptors & relax matching criterion
    - Need to handle higher outlier ratios (>99%)
    - Promising results: [Li et al., ECCV'12] [Svärm et al., CVPR'14]
    - … pose estimation times grow too fast

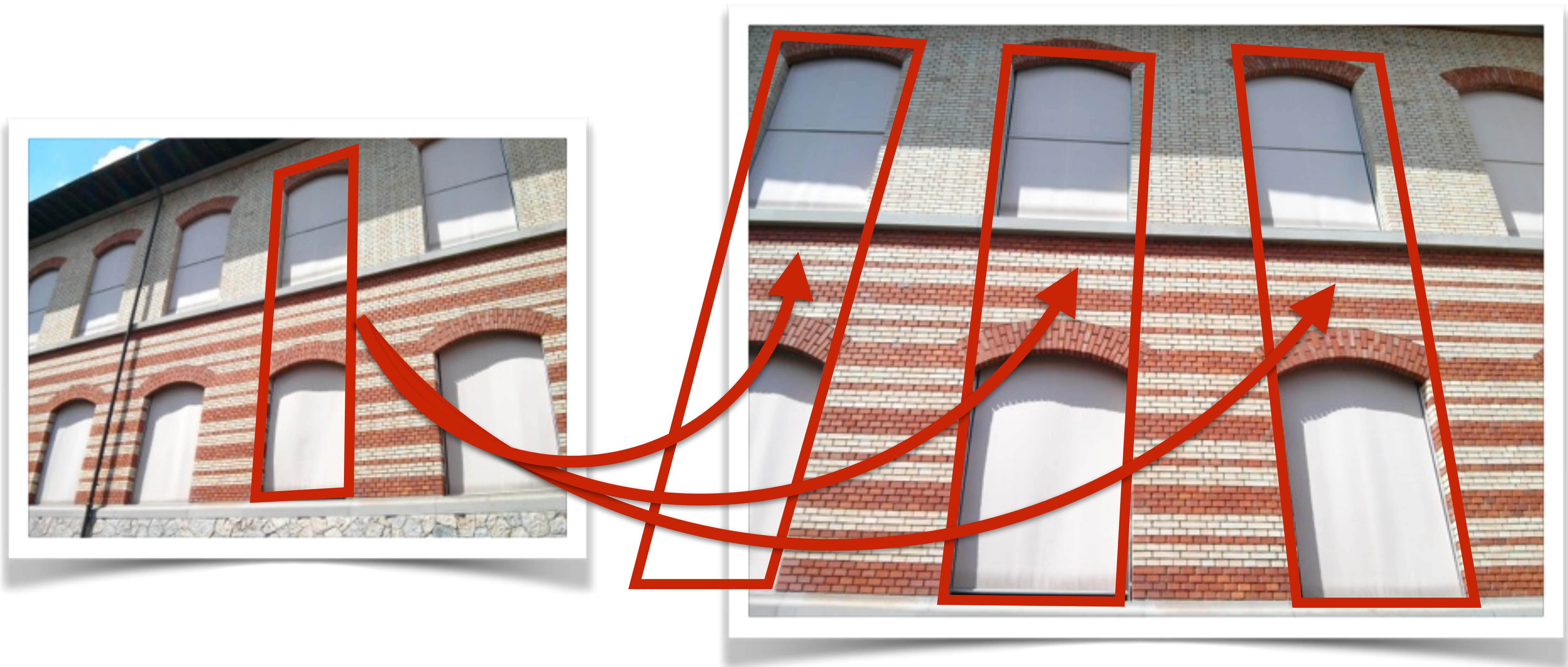
(Quasi-)Identical Structures



How to disambiguate between multiple valid poses?

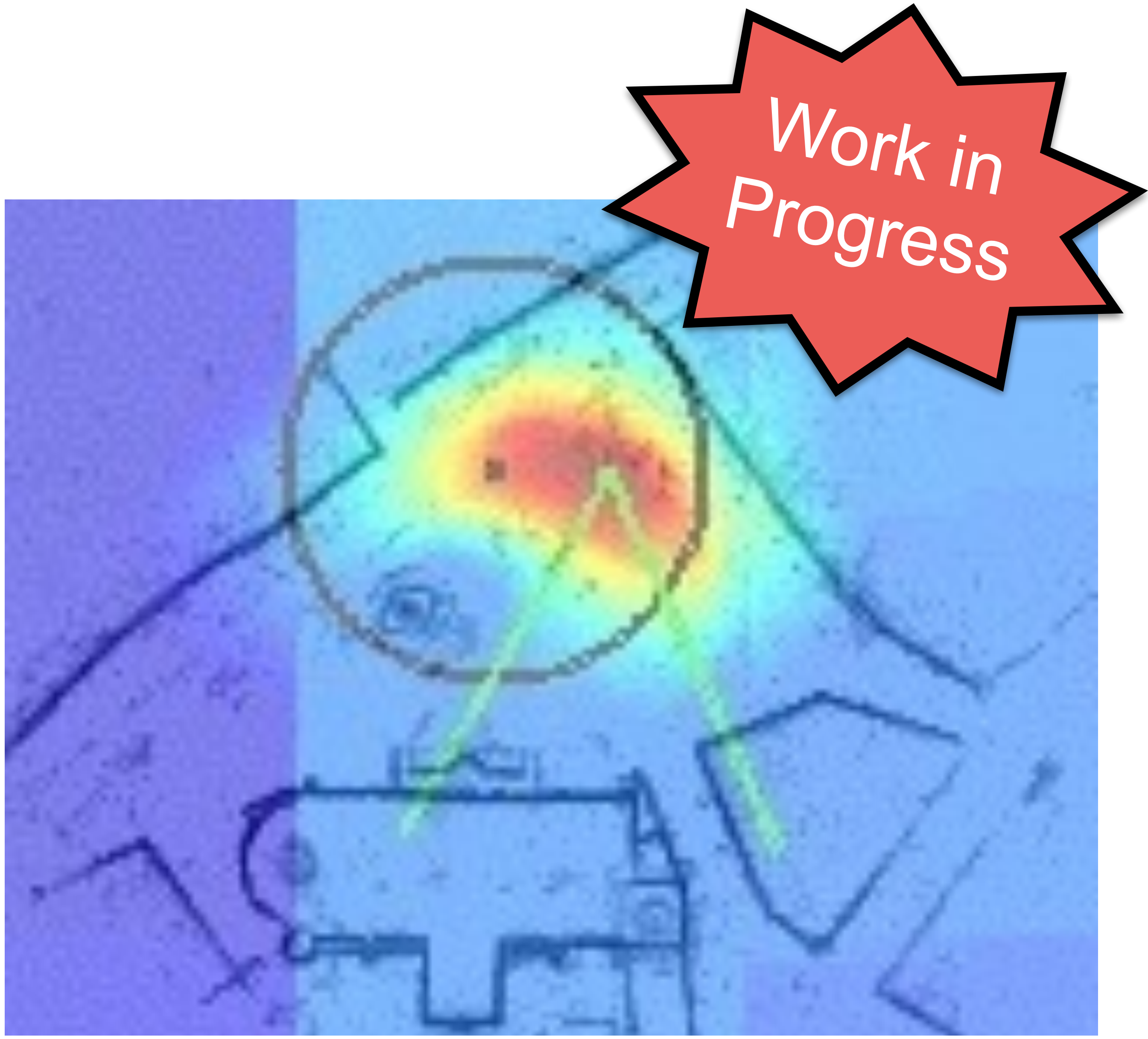
Computer Vision and Geometry Group ETH zürich

(Quasi-)Identical Structures



- What to do if we can't disambiguate?
- Can we get at least all plausible poses?

Camera Pose Voting



[Aachen dataset]

- Assume known gravity direction, ground plane
- Iterate over camera height, orientation, vote for position
- Linear in number of matches

Illumination Changes

© Google

- Feature detector fires at completely different positions
- Can we learn co-occurrence between day and night features?

General Changes

General Changes

- Can we learn co-occurrence / changes over time?

General Changes

- Can we learn co-occurrence / changes over time?
- What can we use to distinguish between places?

What Have We Solved?

Easy

Hard

What Have We Solved?

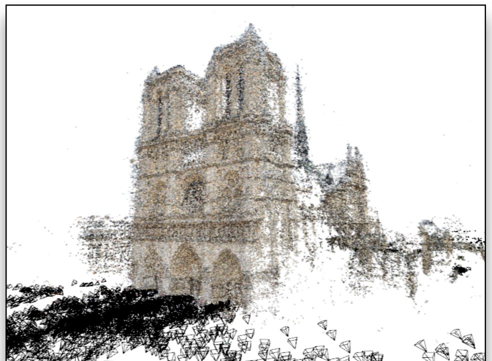
Easy

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

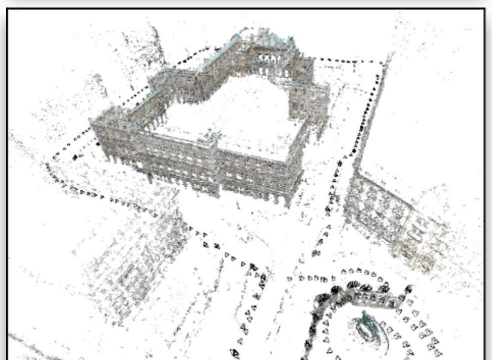
Hard

What Have We Solved?

Easy



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



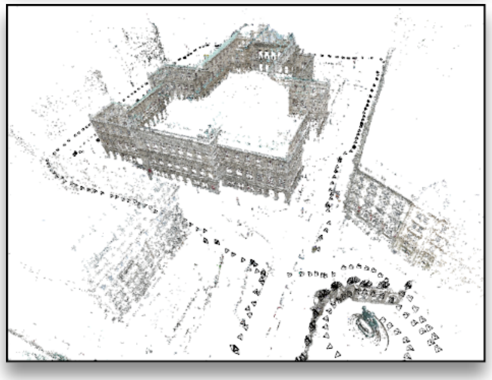
- Database & query images from different spatial distributions
- 70% - 90% localization rates
- *Challenges*: Deal with larger variety in viewpoints

Hard

What Have We Solved?

Easy

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



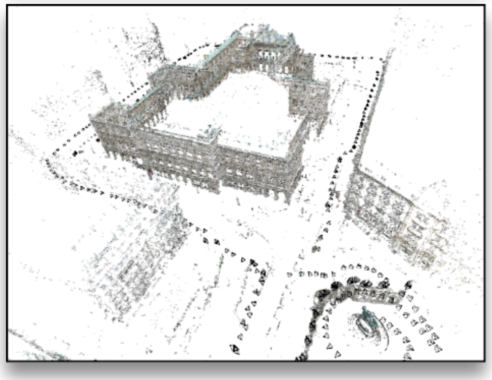
- Database & query images from different spatial distributions
- 70% - 90% localization rates
- *Challenges*: Deal with larger variety in viewpoints

Hard

What Have We Solved?

Easy

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



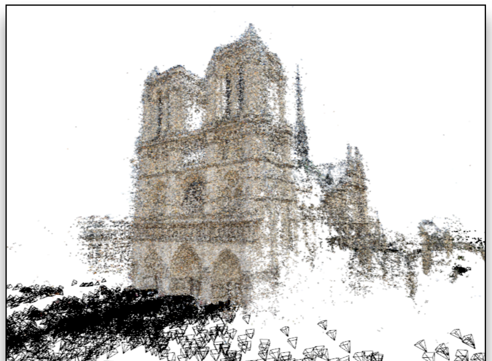
- Database & query images from different spatial distributions
- 70% - 90% localization rates
- *Challenges*: Deal with larger variety in viewpoints

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

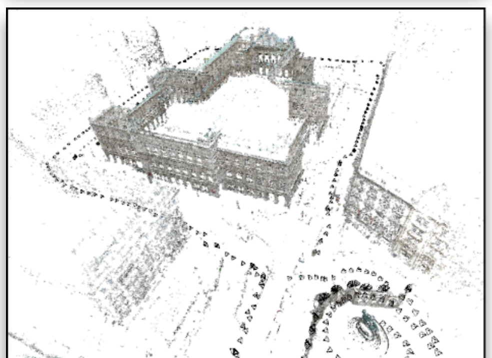
Hard

What Have We Solved?

Easy

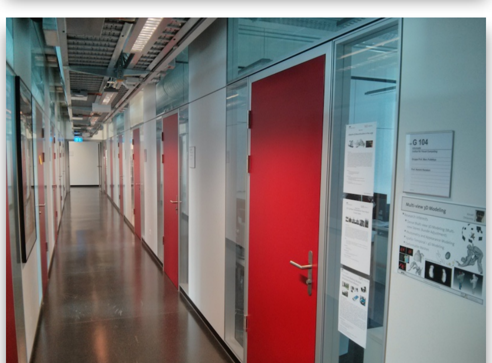


- Database & query images from same source, e.g., Flickr
- 97% - 100% localization rates
- *Challenges*: Run-time & memory consumption for large scale



- 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

Hard

Has the (Large-Scale) Image-based Localization Problem been solved?