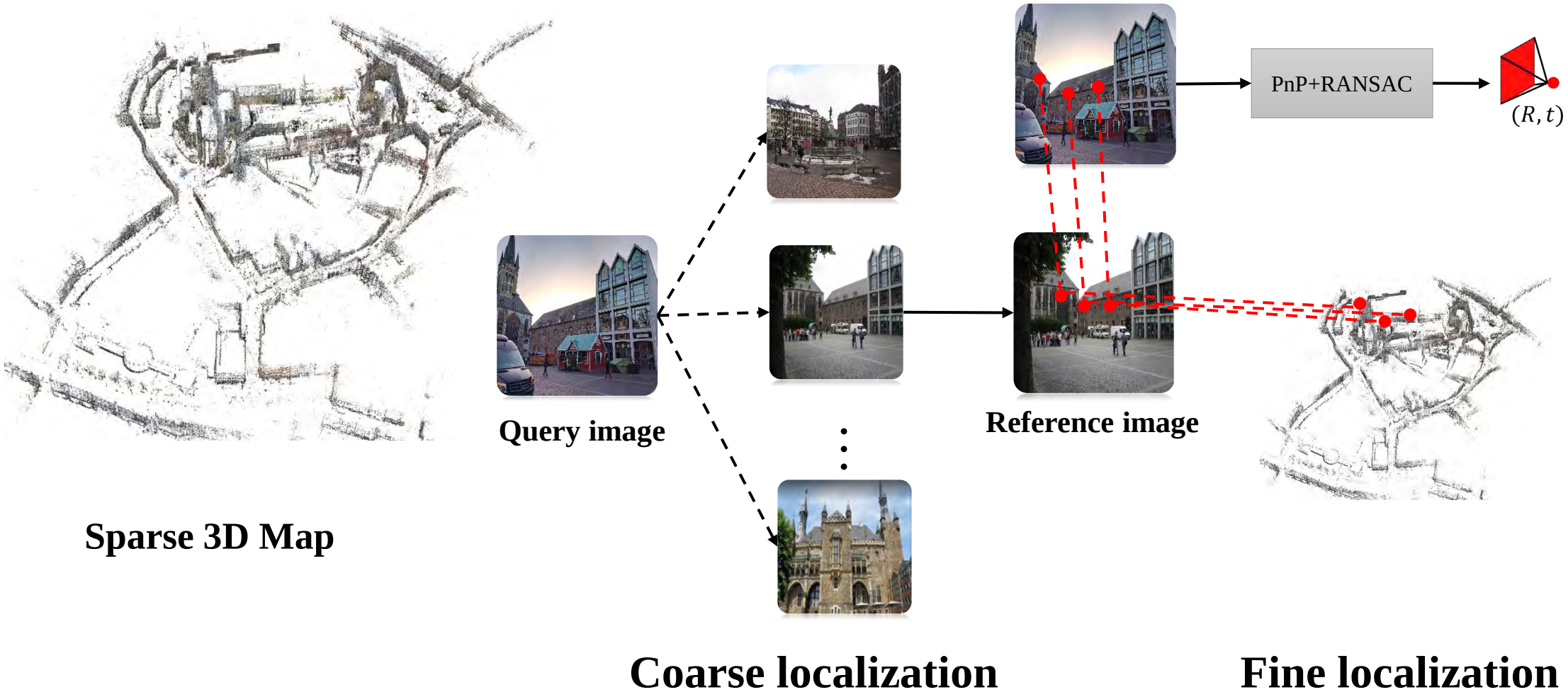
Efficient Large-scale Localization by Global Instance Recognition

Fei Xue Ignas Budvytis Daniel Olmeda Reino Roberto Cipolla

Structure-based localization



Challenges

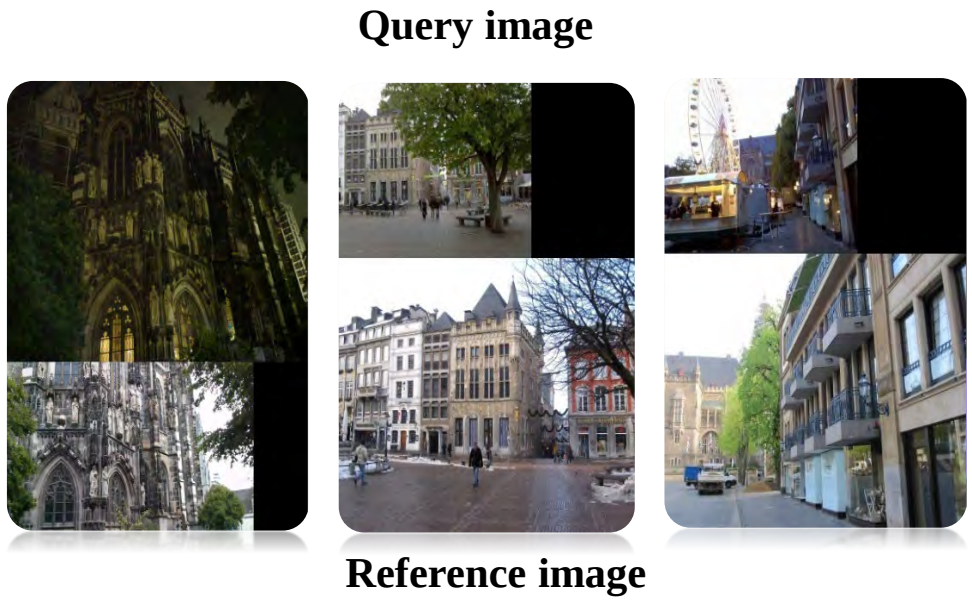
1. Large-scale

- Slow search for reference from the whole database

Aachen city (1.5km x 1.5km, 6697 reference images)[1]

2. Appearance changes

- Wrong matches between keypoints



Illumination changes Seasonal variations Changing environments

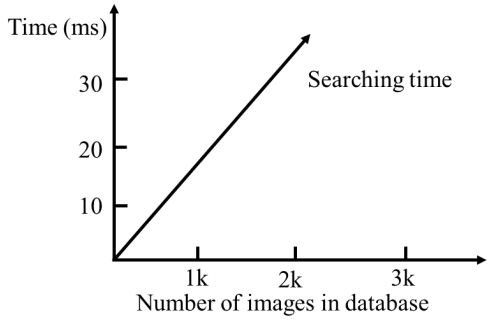
Images from Aachen dataset[1]

[1] Sattler et al., Image Retrieval for Image-Based Localization Revisited. BMVC 2012

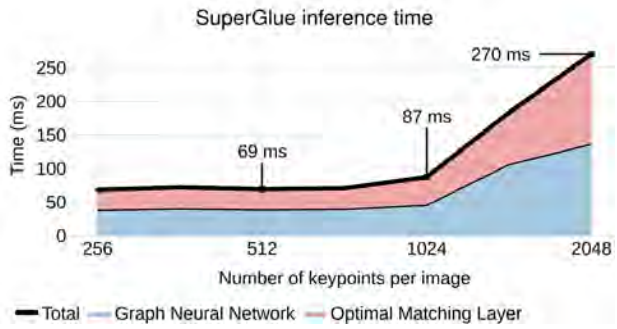
Prior solutions

1. Hierarchical localization methods

- NetVLAD[2] + Superpoint[3]
- NetVLAD[2]+Superpoint[2]+SuperGlue[6]
- Slow global reference search
- Slow advanced matchers



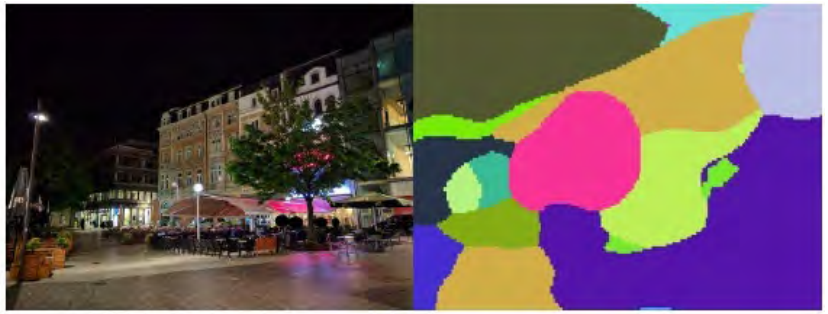
Linear complexity of reference search



Quadratic complexity of SuperGlue[6]

2. Semantics-based localization

- Globally unique instance[7], SSM[4], SMC[5]
- Direct filtering
- Fragile to segmentation errors (e.g., night images)



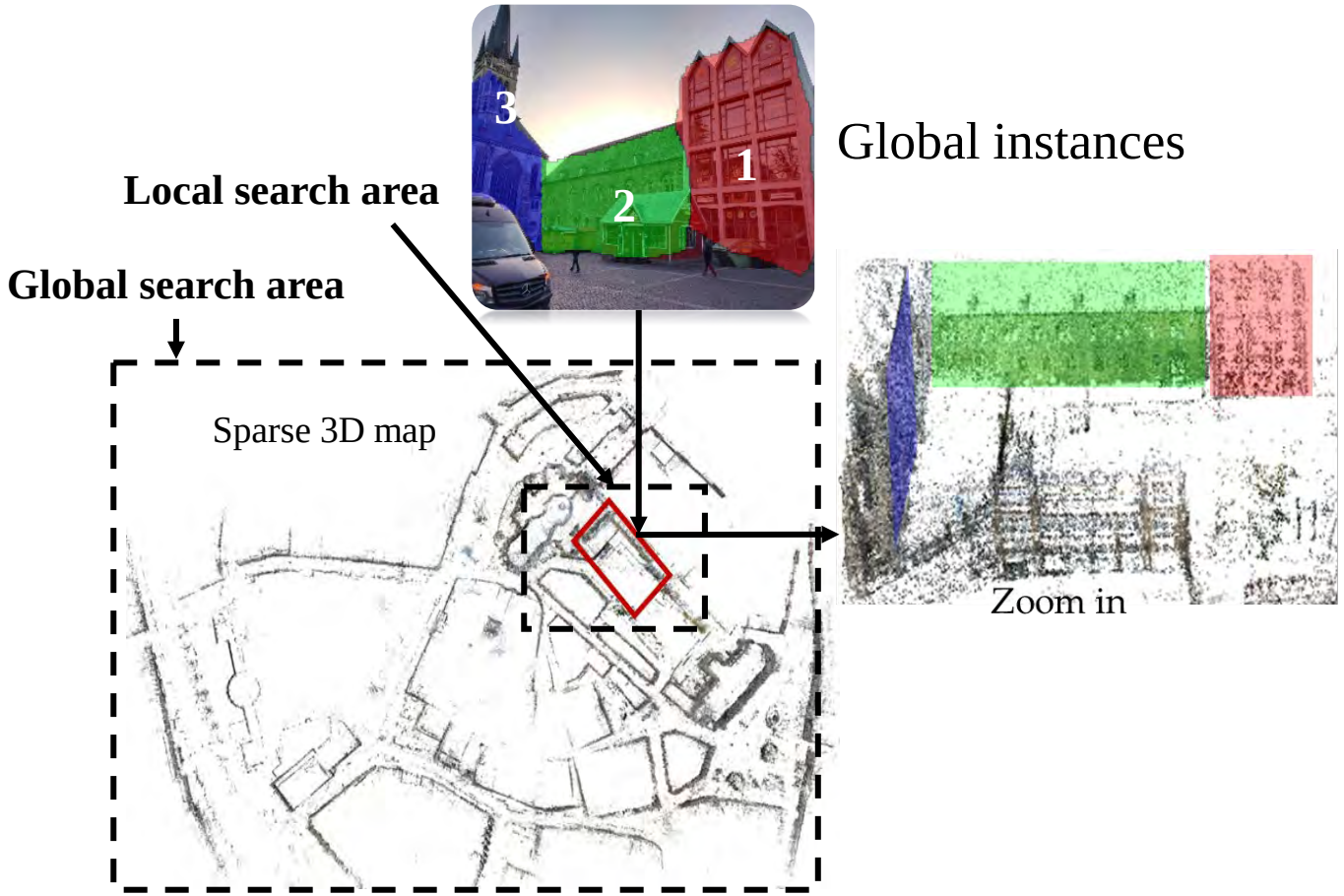
Segmentations on night images of Aachen dataset[1]

[1] Sattler et al., Image Retrieval for Image-Based Localization Revisited. BMVC 2012
[2] Arandjelovic, et al., NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. CVPR 2016
[3] DeTone et al., Superpoint: Self-supervised interest point detection and description. CVPRW 2018
[4] Shi et al., Visual localization using sparse semantic 3D map. ICIP 2019
[5] Toft et al., Semantic match consistency for long-term visual localization. ECCV 2018
[6] Sarlin et al., SuperGlue: Learning Feature Matching with Graph Neural Networks. CVPR 2020
[7] Budvytis et al., Large Scale Joint Semantic Re-Localisation and Scene Understanding via Globally Unique Instance Coordinate Regression. BMVC 2019

Our method: global instance recognition

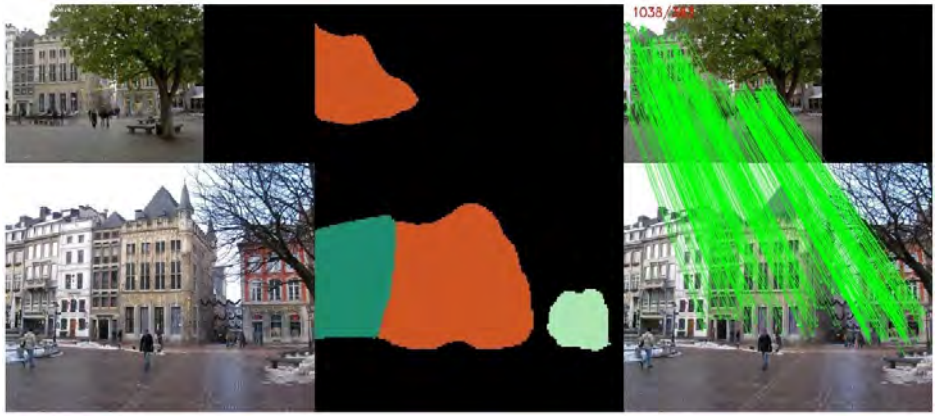
1. Discriminative for locations

- From **global search** to **local search**

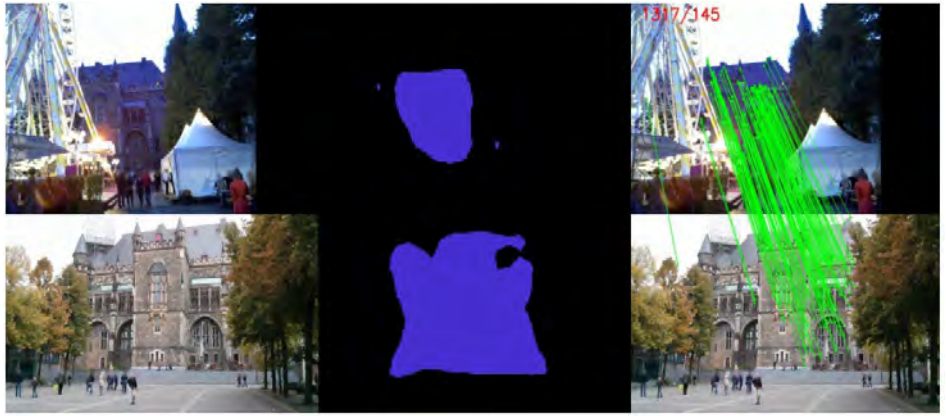


2. Robust to appearance changes

- Instance-wise detection & matching



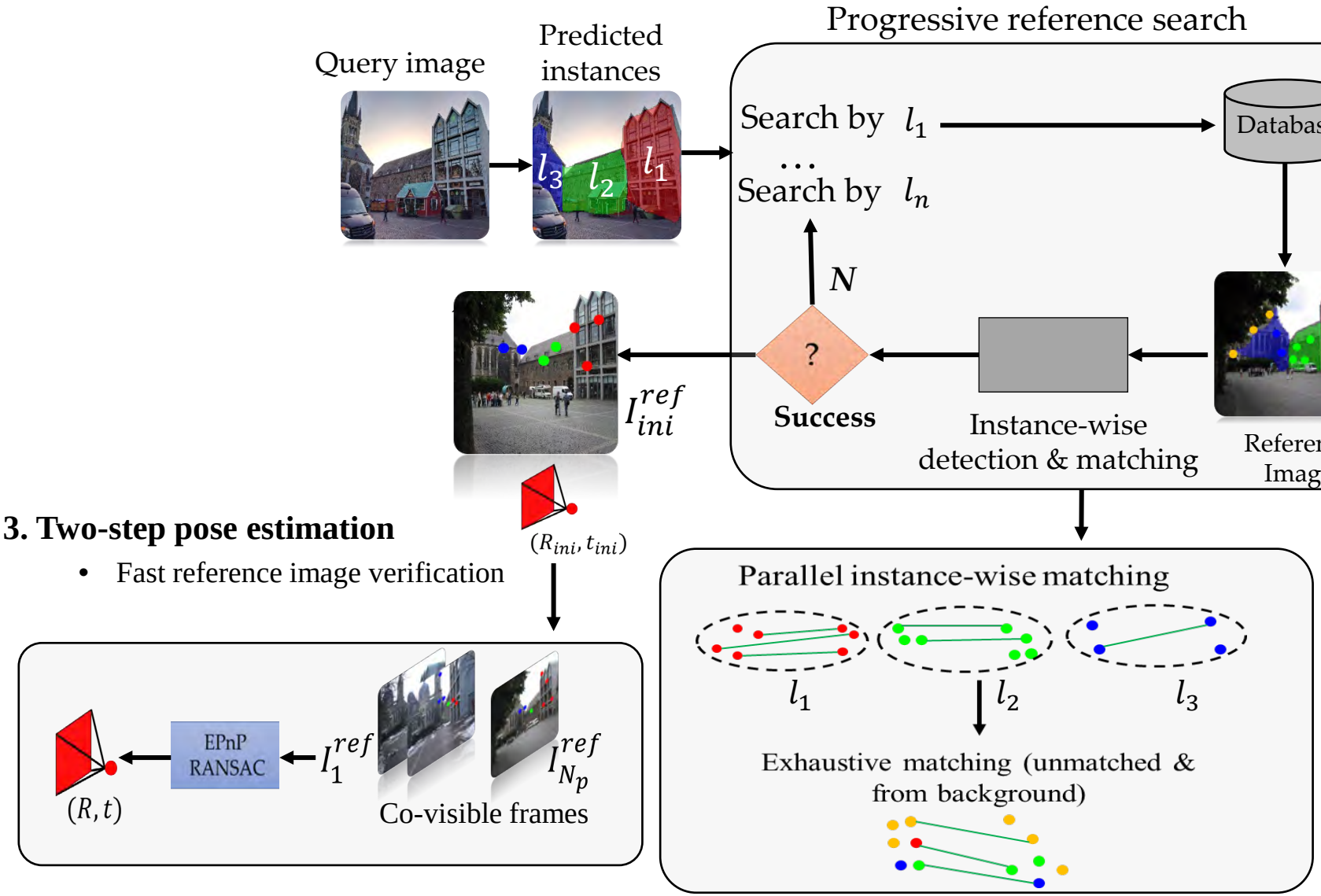
Matches between images[1] in changing scenes



Matches between images[1] across seasons

[1] Sattler et al., Image Retrieval for Image-Based Localization Revisited. BMVC 2012

Our method: robust localization by recognition



1. Progressive reference search

- Fast search in local area
- Robust to recognition errors

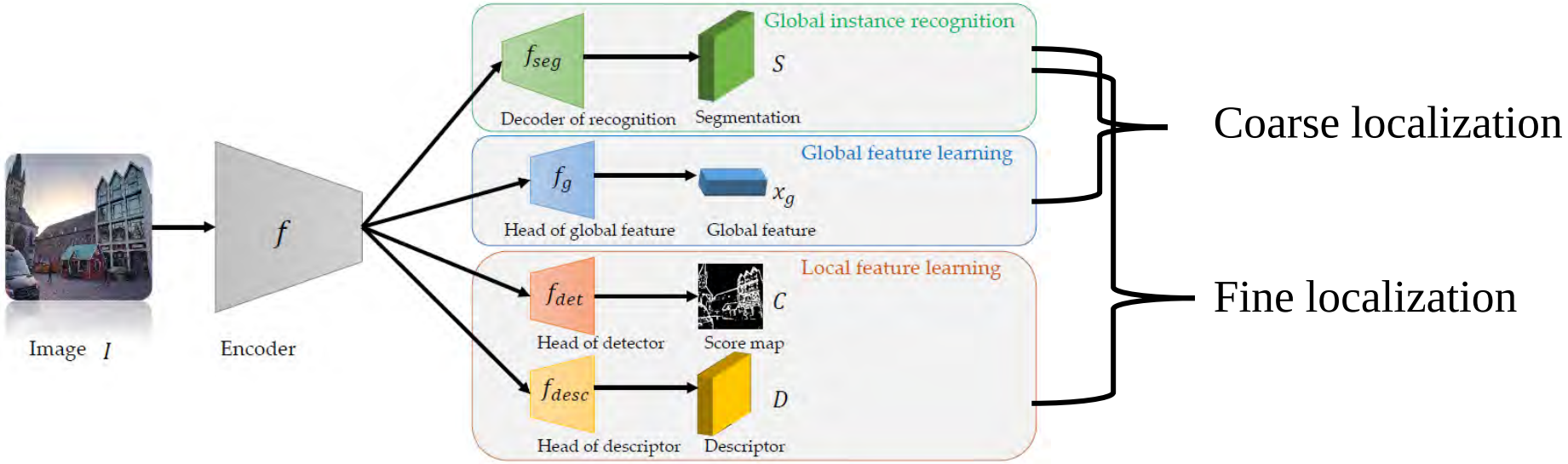
2. Robust instance-wise matching

- Robust to segmentation errors
- Robust to images without global instances

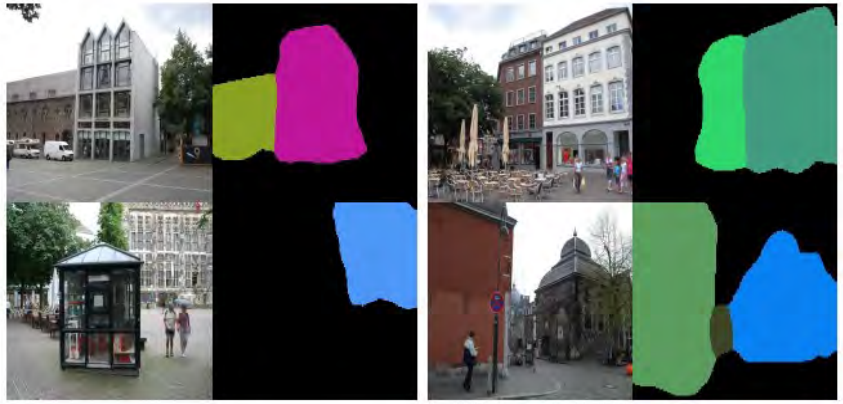
3. Two-step pose estimation

- Fast reference image verification

Experimental setup



Architecture of our network



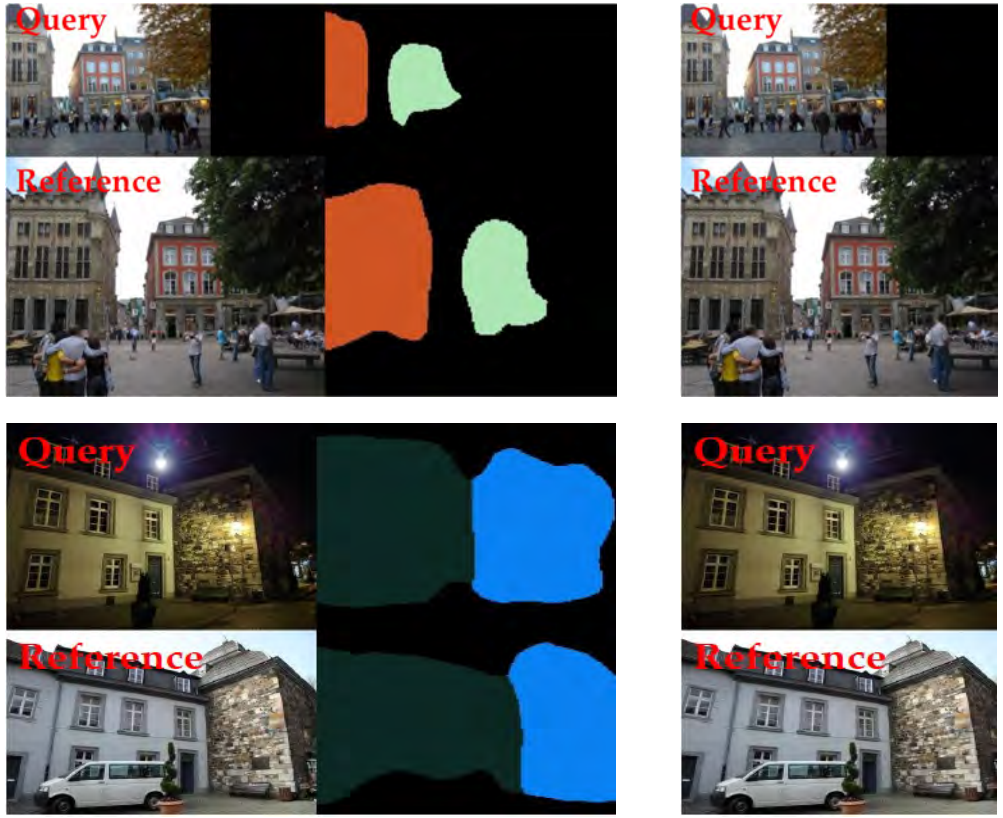
Aachen dataset[1]
(452 global instances, 6697 reference images)

RobotCar-Seasons dataset[2]
(692 global instances, 8707 reference images)

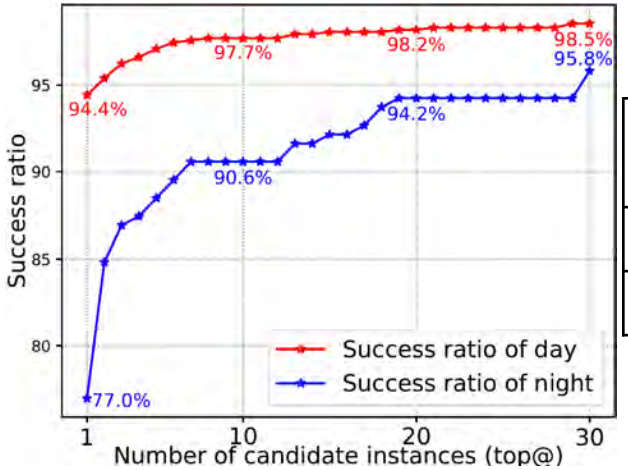
[1] Sattler et al., Image Retrieval for Image-Based Localization Revisited. BMVC 2012
[2] Maddern, et al., 1 Year, 1000km: The Oxford RobotCar Dataset. IJRR 2017

Experiment 1: progressive reference search

Ours (search frames: 80) **NetVLAD[2] (search frames: 6697)**



As good as NetVLAD, but 33x, 10x faster on day/night images



	Avg. search frames (day/night)
NetVLAD	6697/6697
Ours	**202/650**

Candidate instances and search frames on Aachen[1]

Success ratio (night/day)	@10	@20	@50
NetVLAD	97.9/98.8/**6697**	99.0/99.2/**6697**	99.3/99.5/**6697**
Ours	90.6/97.7/**800**	94.2/98.2/**1600**	99.3/99.5/**6697**

Success ratio and number of reference images

[1] Sattler et al., Image Retrieval for Image-Based Localization Revisited. BMVC 2012
[2] Arandjelovic, et al., NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. CVPR 2016

Experiment 2: fine localization accuracy

We achieve best (bold) or second best fine localization accuracy on Aachen[1]

		Day	Night
Classic	**CPF[5]**	76.7 / 88.6 / 95.8	33.7 / 48.0 / 62.2
Semantic-aware	**SMC[7]**	71.8 / 91.5 / 96.8	58.2 / 76.5 / 90.8
Learned feature	**D2Net[6]**	84.8 / 92.6 / 97.5	84.7 / 90.8 / 96.9
Advanced matcher	**SPP+SuperGlue[3,4]**	**89.6** / **95.4** / **98.8**	**86.7 / 93.9 / 100.0**
Instance, no advanced matcher	**Ours**	**88.3** / **95.6** / **98.8**	**84.7** / **93.9 / 100.0**

Our method is much faster than prior SOTA (NetVLAD+SPP+SuperGlue)

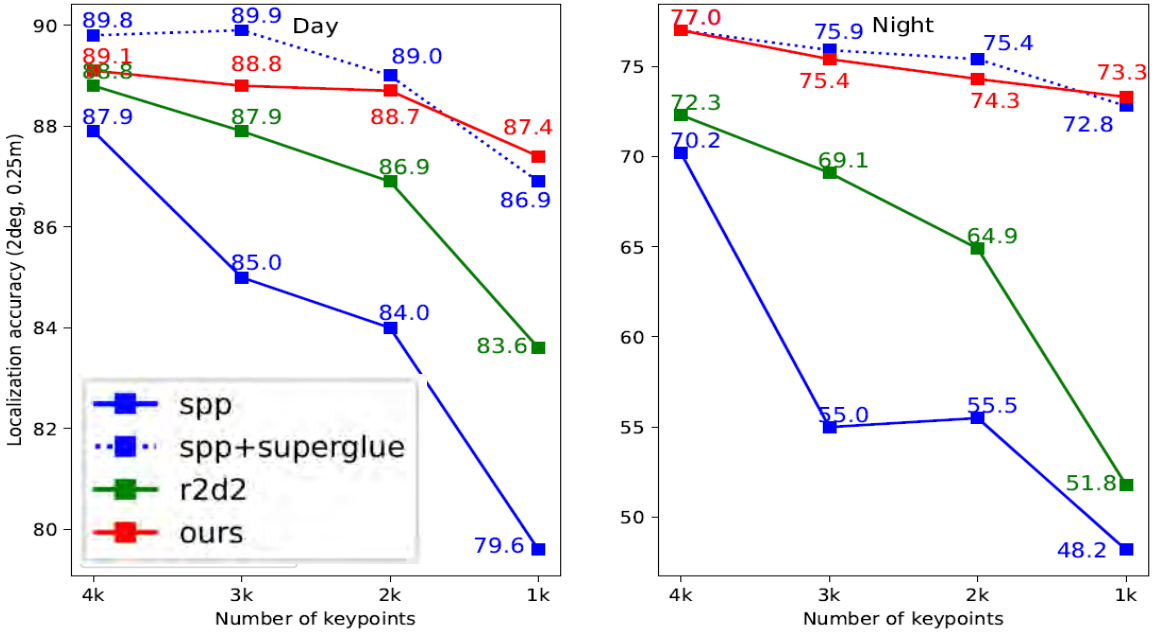
NetVLAD+SPP+SuperGlue	**NetVLAD[2] (1024x1024)**	**SPP (1024x1024)**	**SuperGlue (4k kpts)**	**Total**
	31.9ms	12.0ms	146.8ms	190.7ms
Ours	**Recognition (256x256)**	**Local feature (1024x1024)**	**Instance-wise match (4k kpts)**	
	9.2ms	30.1ms	3ms	42.3ms

Running time of different components (RTX 3090)

[1] Sattler et al., Image Retrieval for Image-Based Localization Revisited. BMVC 2012
[2] Arandjelovic, et al., NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. CVPR 2016
[3] DeTone et al., Superpoint: Self-supervised interest point detection and description. CVPRW 2018
[4] Sarlin et al., SuperGlue: Learning Feature Matching with Graph Neural Networks. CVPR 2020
[5] Cheng et al., Cascaded parallel filtering for memory-efficient image-based localization. ICCV 2019
[6] Dusmanu et al., D2-Net: A trainable CNN for joint description and detection of local features. CVPR 2019
[7] Shi et al., Visual localization using sparse semantic 3D map.ICIP 2019

Experiment 3: robust instance-wise detection and matching

1. Robust to reduced number of keypoints

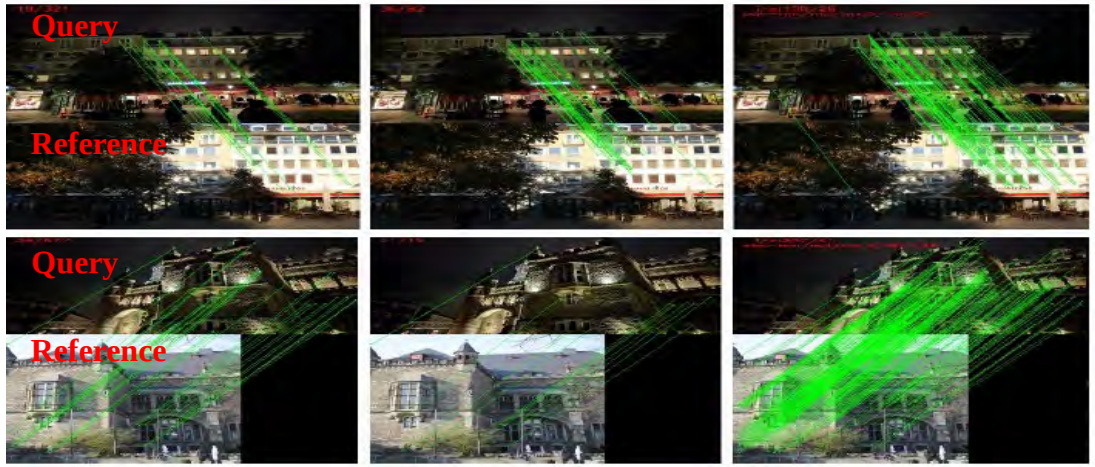


Reduced number of keypoints (Aachen day/night)[1]

2. More robust features and inlier matches

SPP[2] SPP+SuperGlue[23] Ours

Tracking length: green (short) to red (long)



SPP[2] SPP+SuperGlue[23] Ours

Inlier matches

[1] Sattler et al., Image Retrieval for Image-Based Localization Revisited. BMVC 2012
[2] DeTone et al., Superpoint: Self-supervised interest point detection and description. CVPRW 2018
[3] Sarlin et al., SuperGlue: Learning Feature Matching with Graph Neural Networks. CVPR 2020
[4] Revaud et al., R2D2: repeatable and reliable detector and descriptor. NeurIPS 2019.

Conclusion and limitations

- **Localization by recognizing global instances**
  - Progressive reference search (fast and robust to recognition errors)
  - Robust instance-wise matching (fast and robust to segmentation errors)

- **Limitations and future work**
  - Current instances are defined on buildings -> extension to other objects
  - It works in large-scale scenes -> application to larger-scale scenes (city-scale)

Source code: https://github.com/feixue94/lbr