

Where was this image taken?

Visual localization with kapture

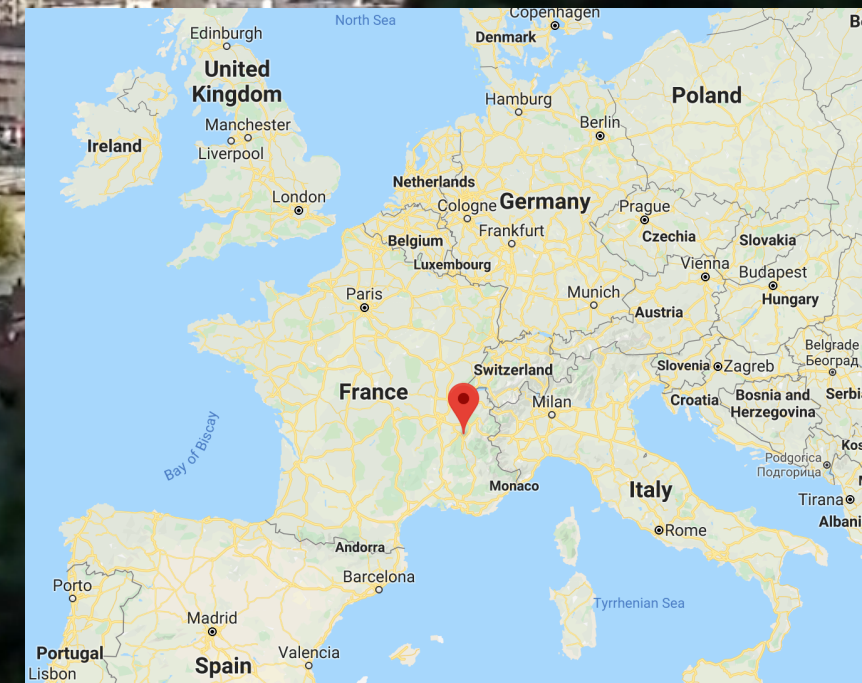
Martin Humenberger
Nicolas Guérin

NAVER LABS Europe

NAVER LABS

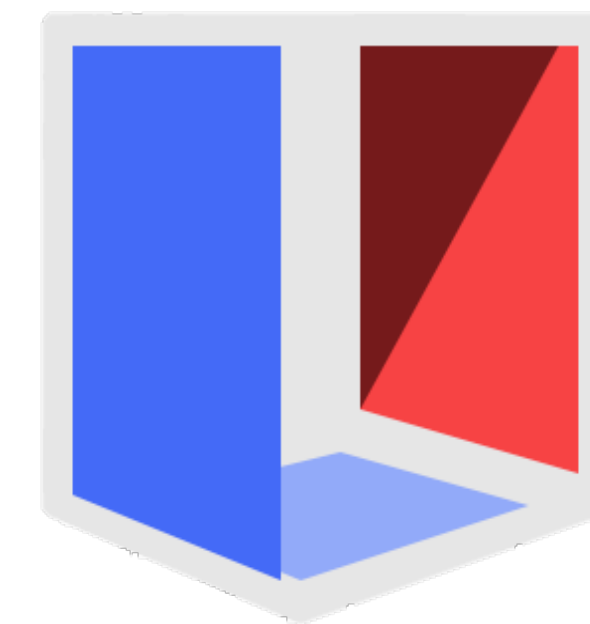
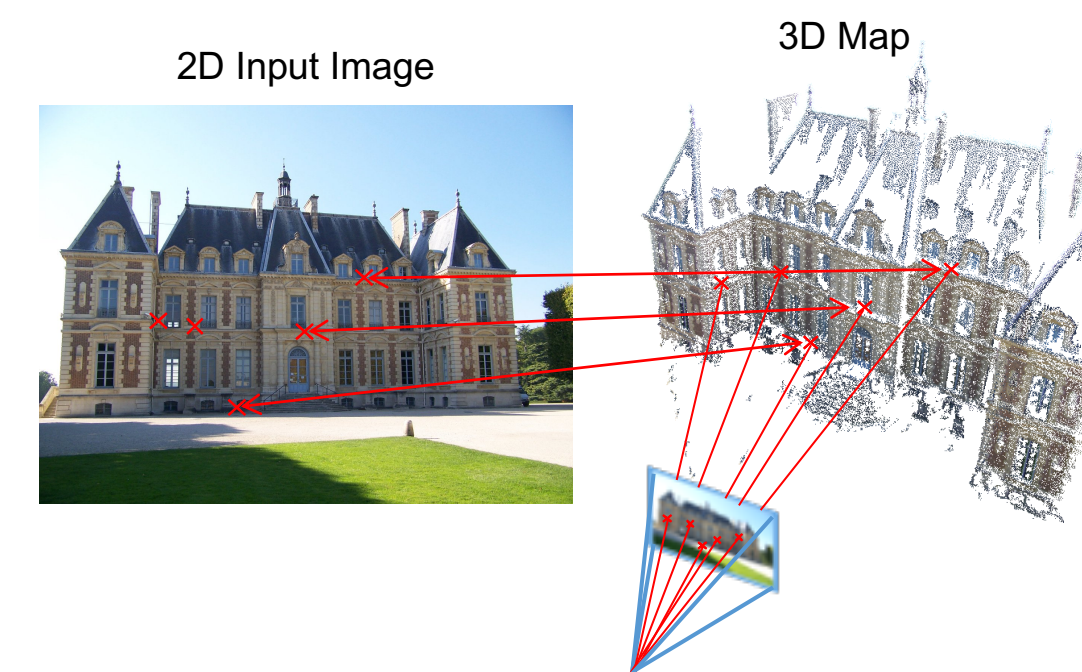
Europe

GRENOBLE, FRANCE

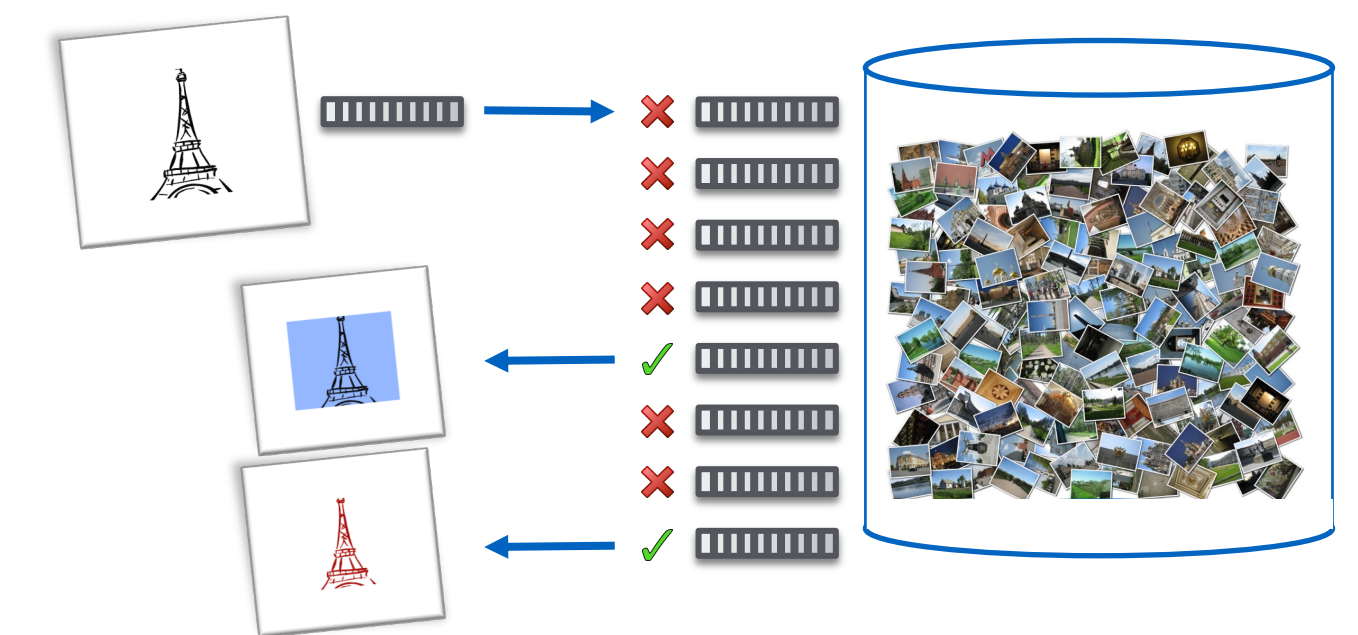


Outline

- Visual localization and its applications
- Methods for visual localization
- Description of a visual localization pipeline based on image retrieval
- Hands-on visual mapping and localization using kapture
 - What is a kapture?
 - Step-by-step: Download a dataset, build a map and localize with kapture



kapture



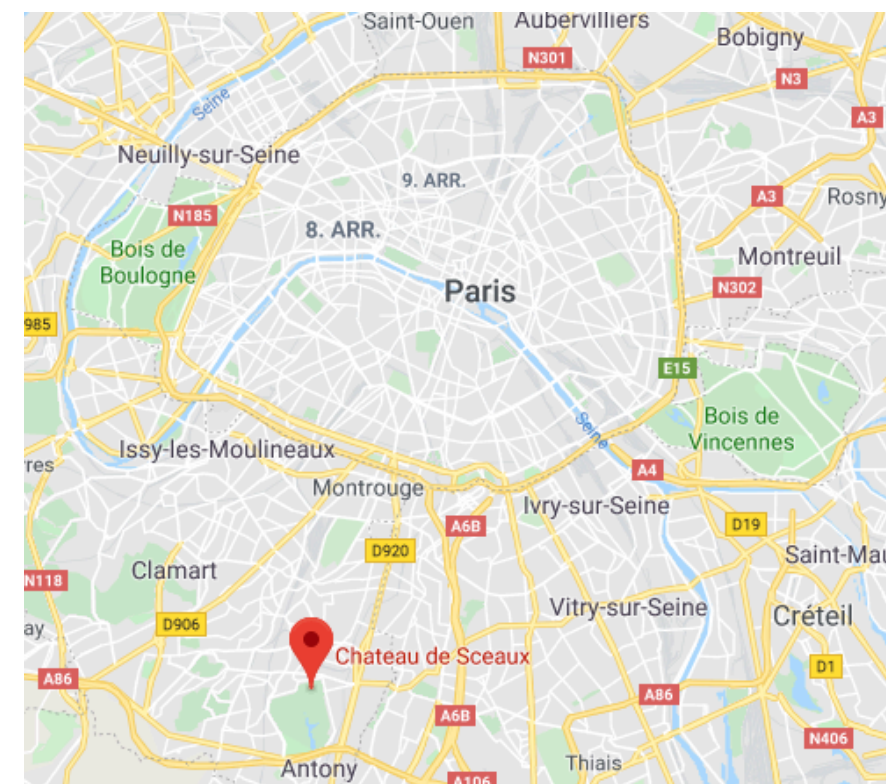
Visual localization and its application

Visual localization and its application



GPS accuracy sometimes not enough.
E.g. for precise robot navigation or
augmented reality.

Position from GPS



Château de Sceaux



Goal: Use an image to estimate the
precise position of the camera within a
given area (map).

Visual localization and its application



Visual localization and its application

강남역 AR Navigation
복층 경로 1
B2 승강장 - 11번 출구 (B1)

NAVER LABS

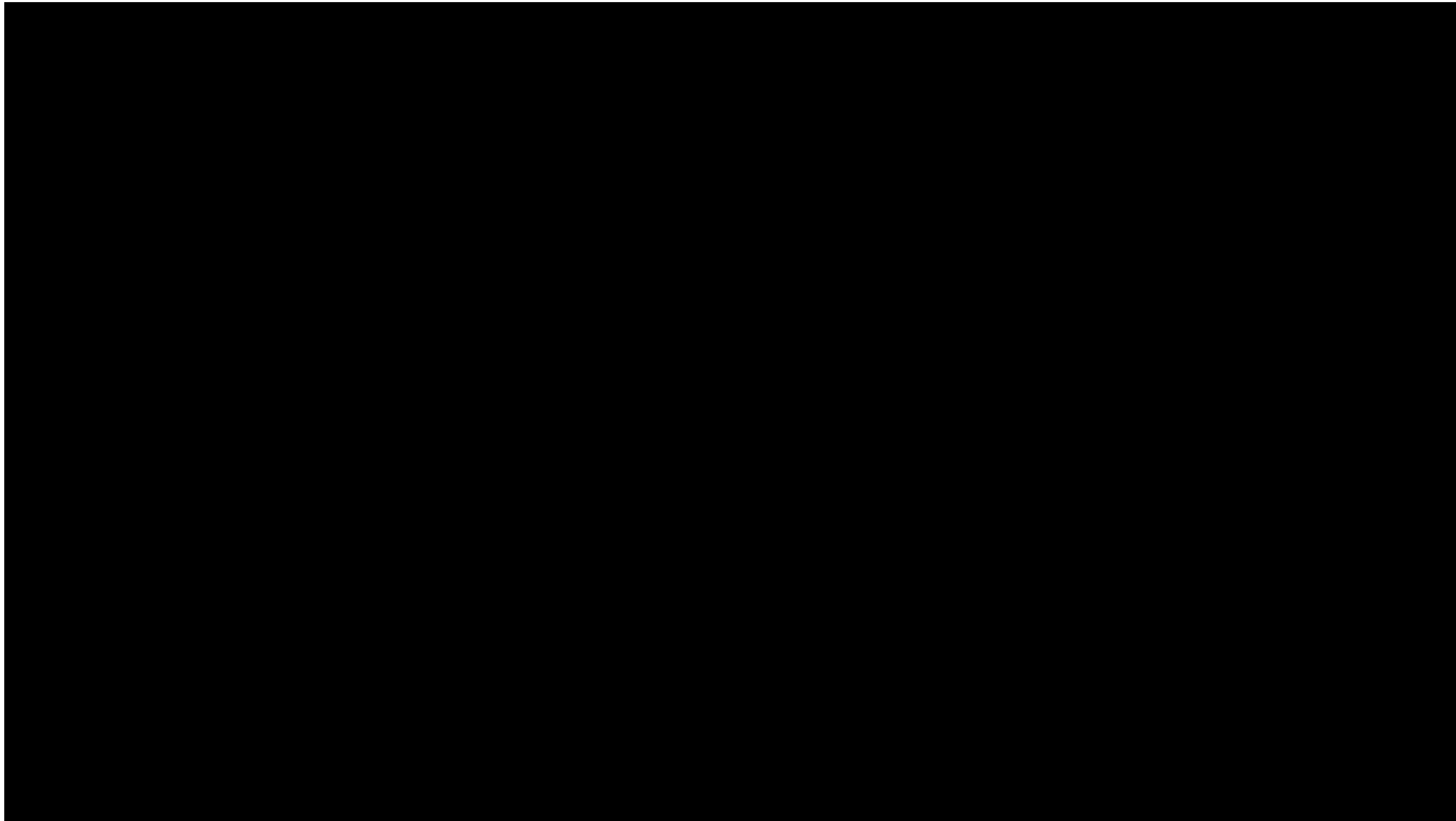
NAVER LABS

Visual localization and its application



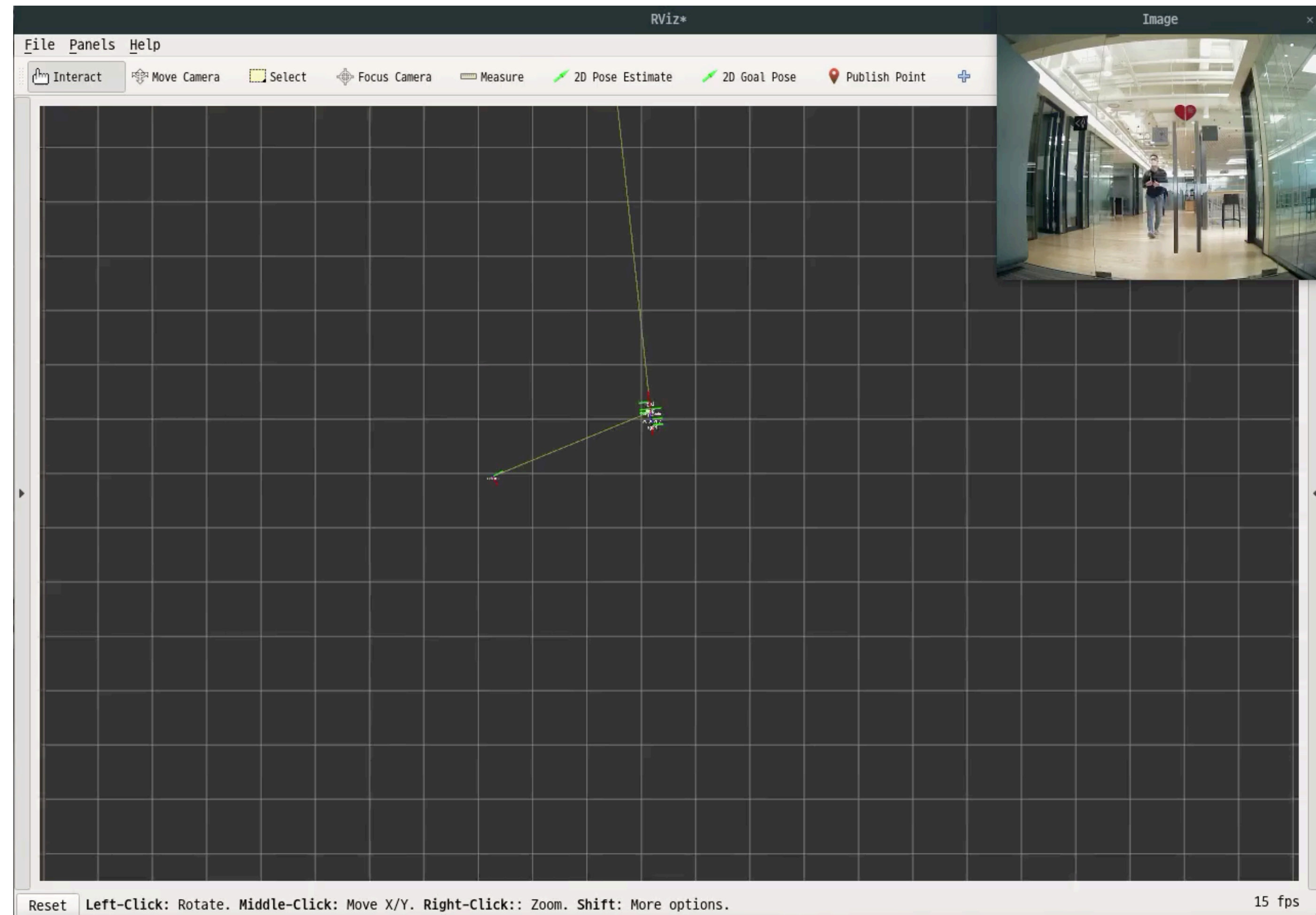
iRobot®

Visual localization and its application



NAVER LABS

Visual localization and its application



NAVER LABS

Methods of Visual Localization



Methods of Visual Localization - Principle

Mapping

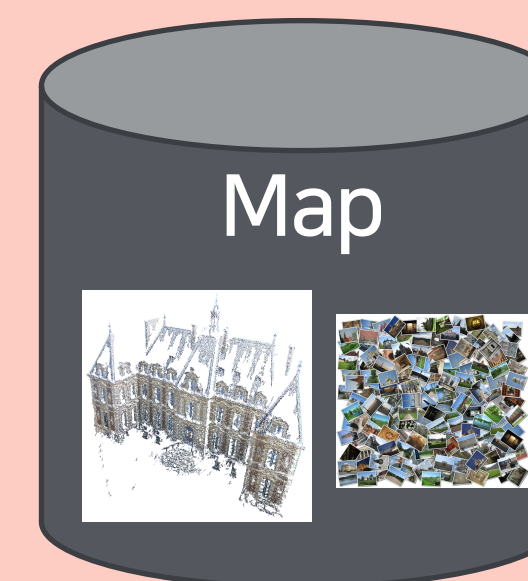
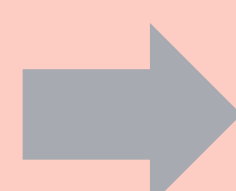
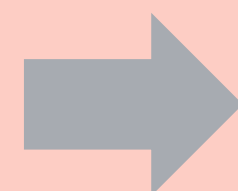


Image acquisition

- handheld camera
- car
- robot

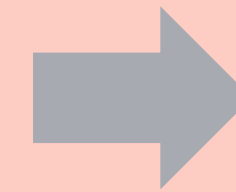
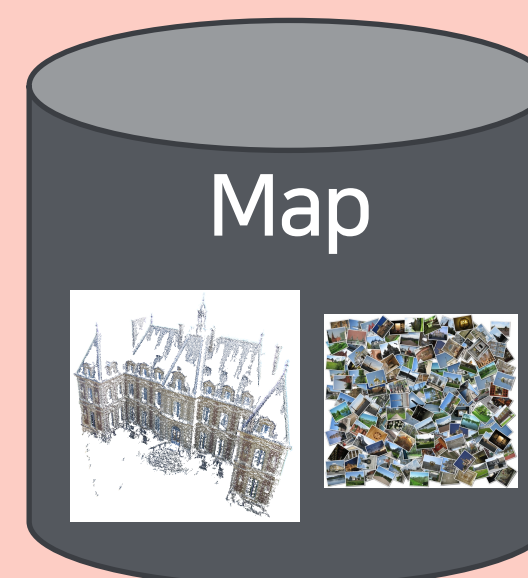
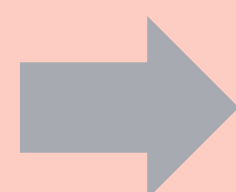
Pose labeling

- structure from motion
- other sensors, e.g., on a robot

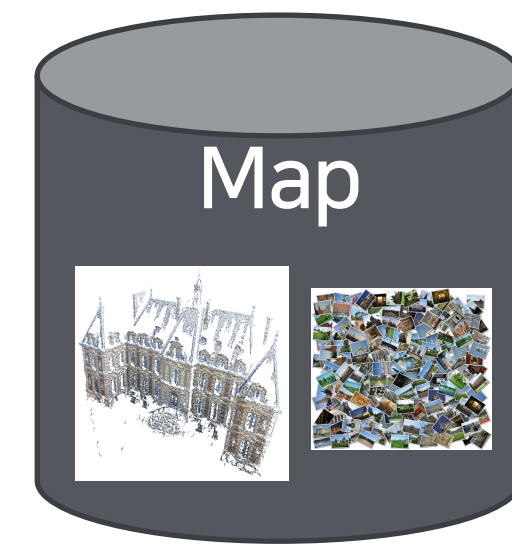
Map generation

- images + poses
- structure from motion
- CNN

Localization



Methods of Visual Localization - Challenges



reference image

Images to localize:



easy



illumination



occlusion



viewpoint and scale



viewpoint, occlusion, weather

Structure-based Visual Localization

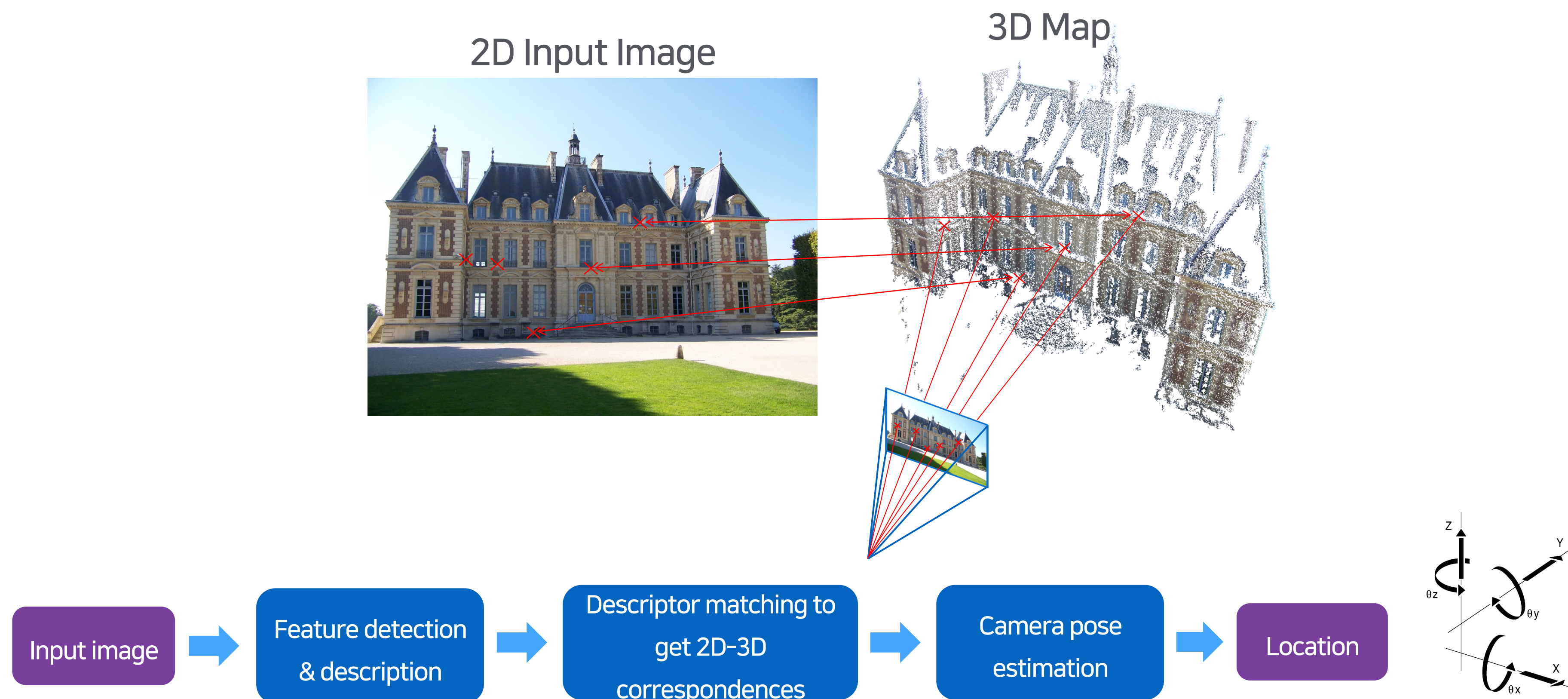


Image Retrieval

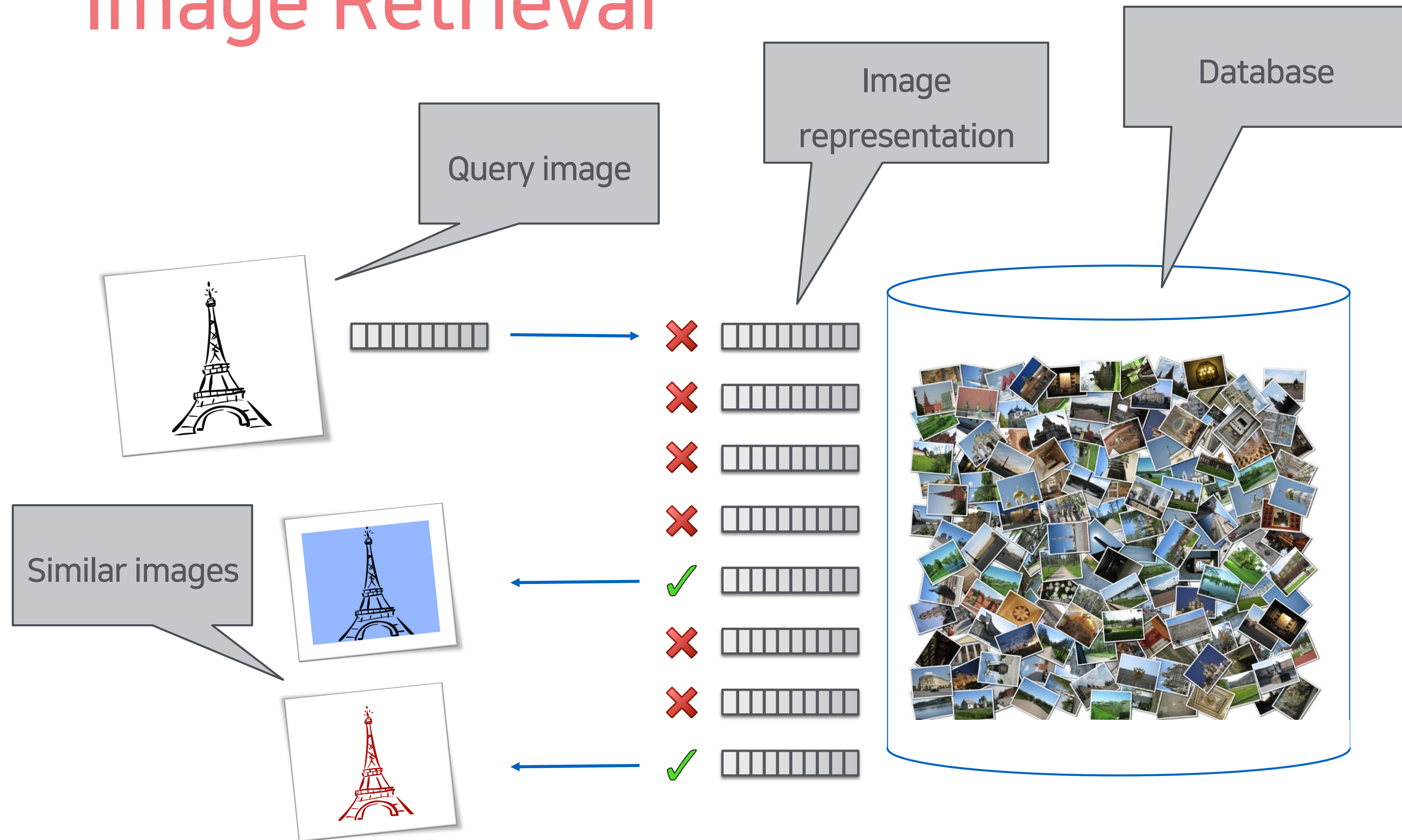
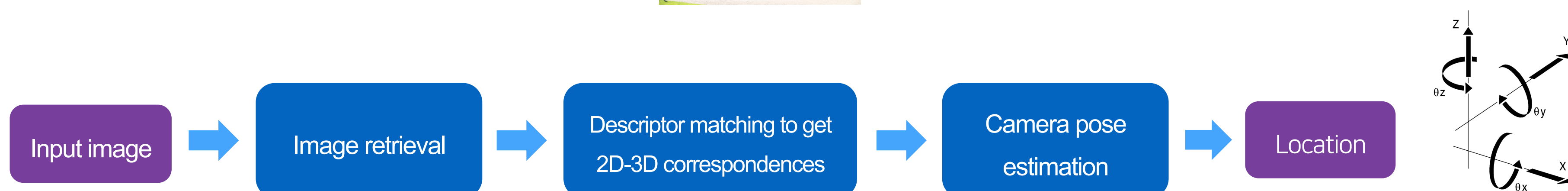
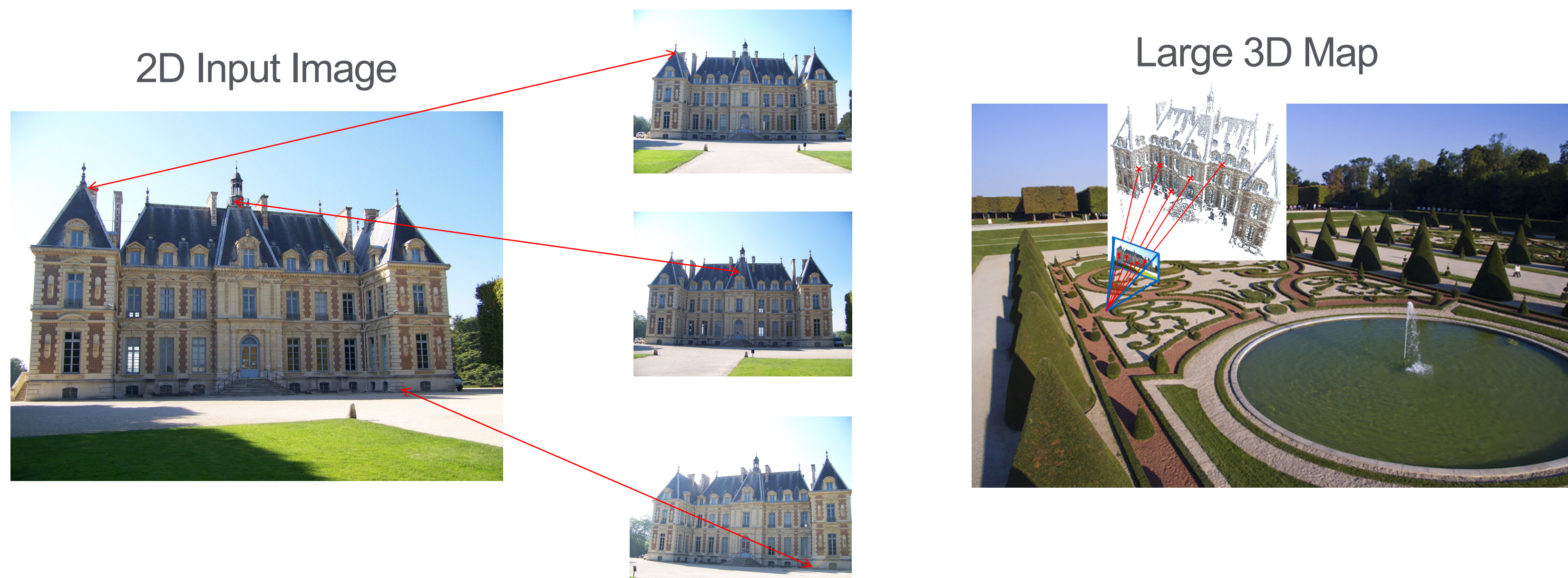


Image Retrieval-based Visual Localization

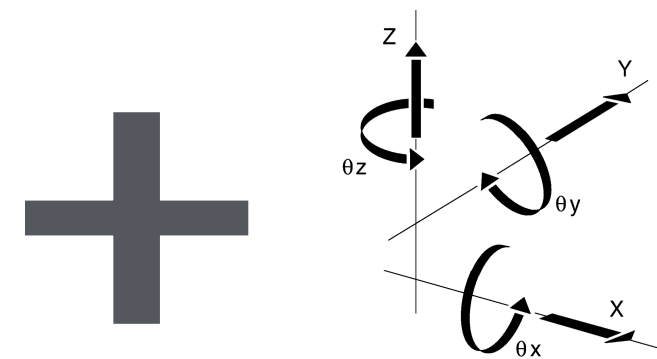


“Benchmarking Image Retrieval for Visual Localization” Pion, Humenberger, Csurka, Cabon, Sattler, 3DV 2020

Pose Regression-based Visual Localization



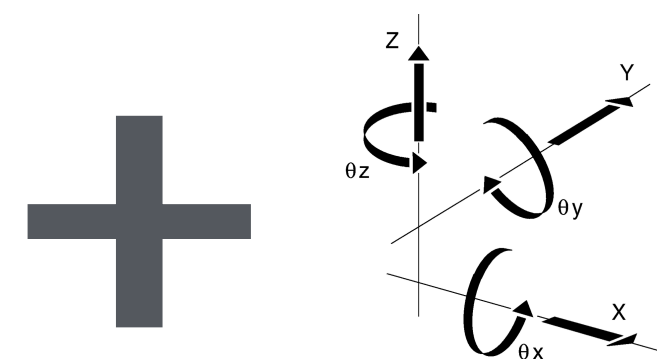
image



camera pose



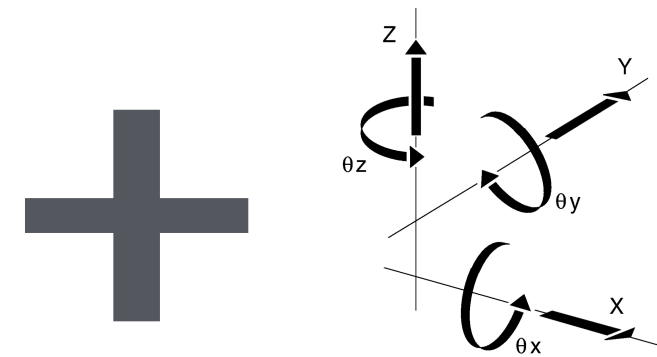
image



camera pose

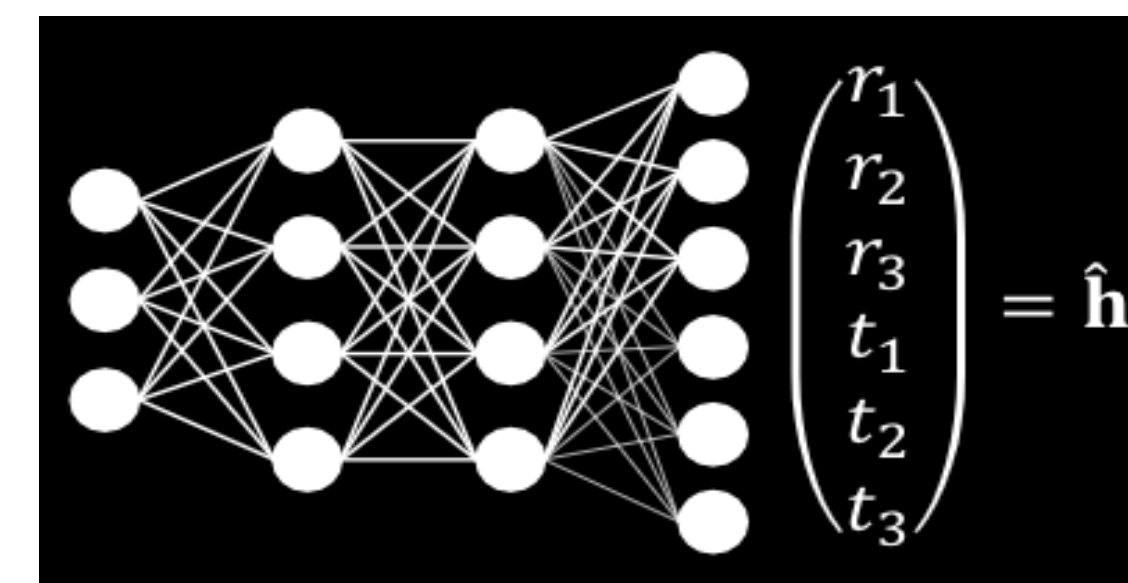


image



camera pose

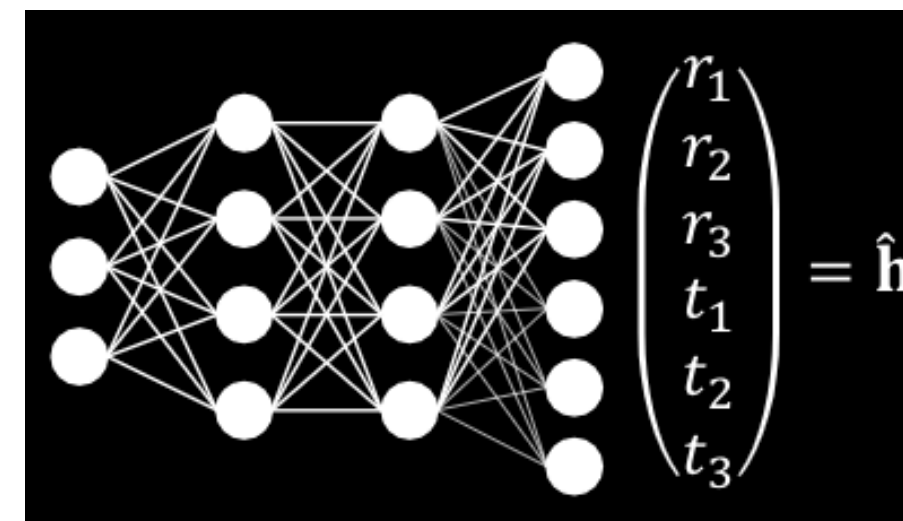
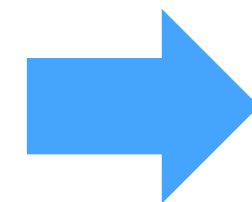
No 3D Map
but...



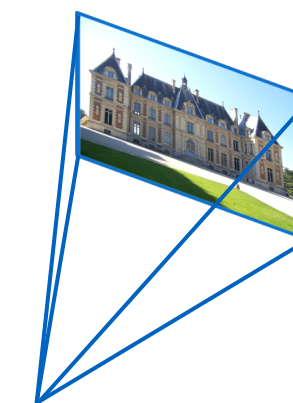
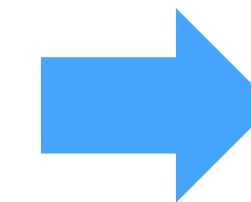
CNN

Pose Regression-based Visual Localization

2D Input Image



CNN



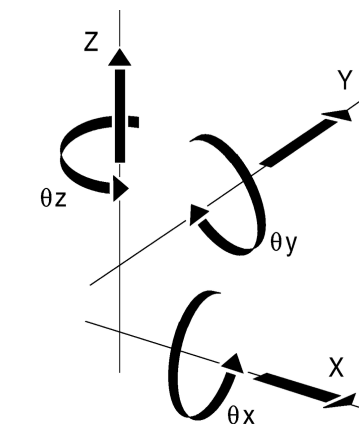
Input image



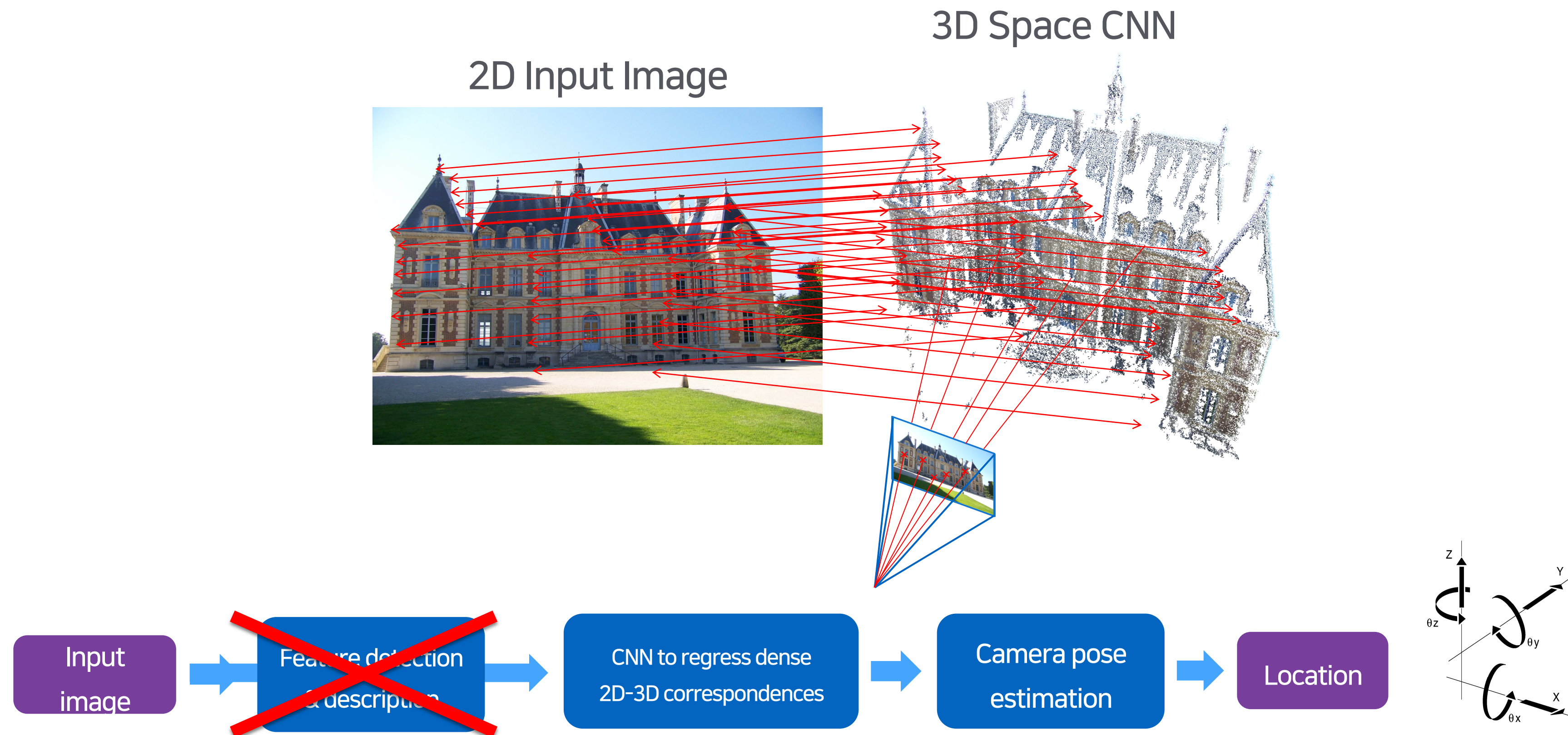
CNN to directly estimate
the camera pose



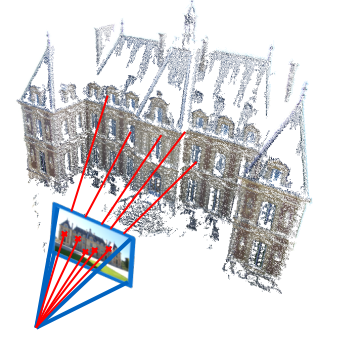
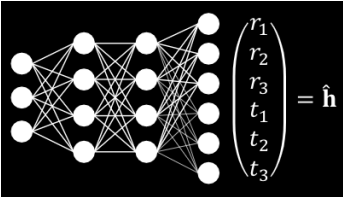
Location



Scene Coordinate Regression-based Visual Localization



Methods of Visual Localization - Summary

Structure-based methods		Active Search [1] OpenMVG [2]	+ Perform very well on most datasets -> high accuracy - More difficult for very large environments (memory and processing time)
Image retrieval-based methods		Kapture [3]	+ Improve speed and robustness for large scale settings - Quality heavily relies on image retrieval
Camera pose regression methods		PoseNet [4]	+ Pose can be directly estimated - Low accuracy
Scene coordinate regression methods		DSAC++ [5]	+ Very accurate in small scale settings - Does not yet work in large scale environments (but can be clustered)

[1] T. Sattler et al., Improving Image-Based Localization by Active Correspondence Search, ECCV 2012

[2] P. Moulon, OpenMVG: <http://github.com/openMVG/openMVG>

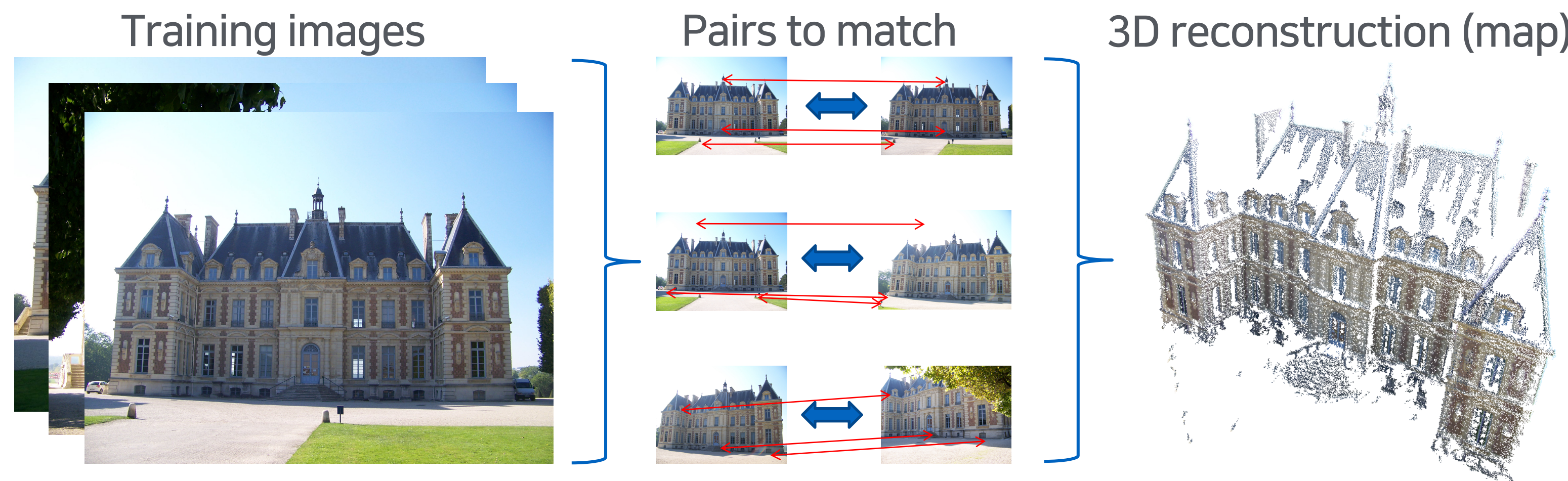
[3] Humenberger et al., "Robust Image Retrieval-based Visual Localization using Kapture." arXiv 2020

[4] A. Kendall et al., PoseNet: <http://mi.eng.cam.ac.uk/projects/relocalisation/>, ICCV 2015

[5] E. Brachmann et al., Learning Less is More – 6D Camera Localization via 3D Surface Regression, CVPR 2018

Visual Localization at NAVER LABS Europe

Visual Localization at NAVER LABS Europe

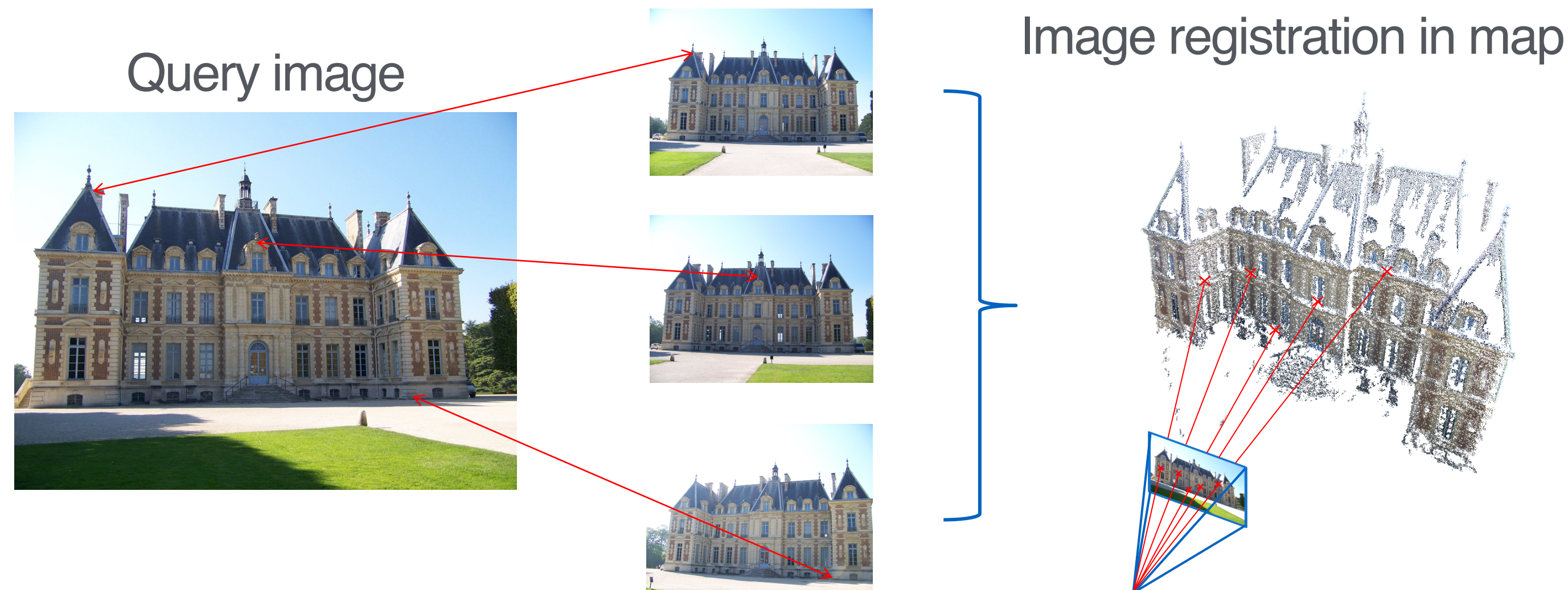


APGeM: <https://github.com/naver/deep-image-retrieval>

R2D2: <https://github.com/naver/r2d2>

COLMAP: <https://colmap.github.io>

Visual Localization at NAVER LABS Europe



APGeM: <https://github.com/naver/deep-image-retrieval>

R2D2: <https://github.com/naver/r2d2>

COLMAP: <https://colmap.github.io>

Visual Localization at NAVER LABS Europe



Results on public datasets from <https://www.visuallocalization.net>



Aachen Day-Night v1.1

<https://data.ciirc.cvut.cz/public/projects/2020VisualLocalization/Aachen-Day-Night/>

Rank 1



RobotCar Seasons v2

<https://data.ciirc.cvut.cz/public/projects/2020VisualLocalization/RobotCar-Seasons/>

Rank 1



SILDa Weather and Time of Day dataset

Balntas et al., SILDa: Scape Imperial Localisation Dataset
<https://www.visuallocalization.net/>, 2019

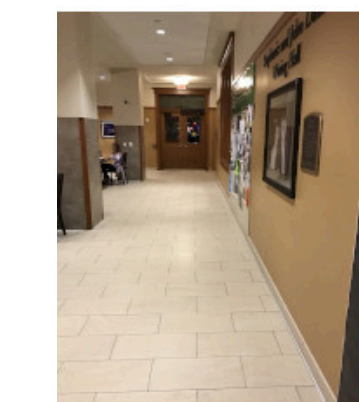
Rank 2



Extended CMU

<https://data.ciirc.cvut.cz/public/projects/2020VisualLocalization/Extended-CMU-Seasons/>

Rank 3

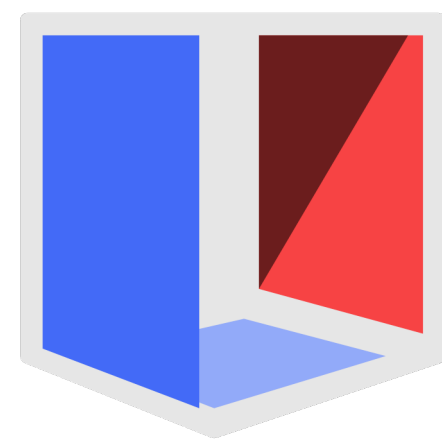


Inloc

<http://www.ok.sc.e.titech.ac.jp/INLOC/>

Rank 14

Hands-on visual mapping and localization using kapture



What is kapture?

Visual Localization datasets may provide different kinds of data:

- Different sensor data
- Reconstruction data

Additionally, Most datasets use a custom data format

It makes it difficult to benchmark different algorithms on many datasets.

This is why we created kapture

- a flexible data format that can be used to describe any localization dataset, including sensor data and processed data.
- a toolbox for localization tasks

<https://github.com/naver/kapture>

kapture



kapture GitHub.com
repository – BSD license

COLMAP pipelines	Tests
Command line tools Converters/evaluation/ visualization	Documentation
Python library	Installer pip/Docker
core I/O algo/utils	Dataset downloader
kapture specifications	

kapture data
format specification

Sensor data	Images	GNSS	WiFi	...
	Sensor/camera parameters			
	Trajectories			
	Camera rig configuration			
Reconstruction data	Global features			
	Local features			
	3D points	Matches		
	Observations			

kapture datasets

Aachen Day/Night 	RobotCar Seasons 	Extended CMU Seasons 
SILDa 	InLoc 	...

Starting with kapture

```

/bin/bash

~$ pip install kapture kapture-localization
~$ kapture_
kapture_colmap_build_map.py      kapture_import_bundler.py
kapture_colmap_build_sift_map.py kapture_import_colmap.py
kapture_colmap_localize.py
kapture_import_Extended_CMU_Seasons.py
kapture_colmap_localize_sift.py  kapture_import_IDL_dataset_cvpr17.py
kapture_compute_image_pairs.py  kapture_import_image_folder.py
kapture_compute_matches.py      kapture_import_image_list.py
kapture_download_dataset.py     kapture_import_nvm.py
kapture_evaluate.py             kapture_import_openmvg.py
kapture_export_colmap.py        kapture_import_opensfm.py
kapture_export_image_list.py    kapture_import_RobotCar_Seasons.py
kapture_export_LTVL2020.py      kapture_import_rosbag.py
kapture_export_openmvg.py       kapture_import_silda.py
kapture_export_opensfm.py       kapture_import_utbm.py
kapture_export_ply.py           kapture_import_virtual_gallery.py
kapture_extract_exif.py         kapture_merge.py
kapture_import_7scenes.py       kapture_print.py

```

Datasets

```

/bin/bash

~$ kapture_download_dataset.py update
INFO  ::downloader: updating dataset list from https://github.com/naver/kapture/raw/master/dataset ...
INFO  ::downloader: dataset index retrieved successfully: 128 datasets

~$ kapture_download_dataset.py list
not installed | Aachen-Day-Night-v1.1_mapping          |
             https://data.ciirc.cvut.cz/public/projects/2020VisualLocalization/kapture//Aachen-Day-Night-v1.1_mapping.tar.gz
not installed | Aachen-Day-Night-v1.1_query_day                    |
             https://data.ciirc.cvut.cz/public/projects/2020VisualLocalization/kapture//Aachen-Day-Night-v1.1_query_day.tar.gz
not installed | Aachen-Day-Night-v1.1_query_night                  |
             https://data.ciirc.cvut.cz/public/projects/2020VisualLocalization/kapture//Aachen-Day-Night-v1.1_query_night.tar.gz
not installed | Extended-CMU-Seasons_slice10_mapping              |
             https://data.ciirc.cvut.cz/public/projects/2020VisualLocalization/kapture//Extended-CMU-Seasons_slice10_mapping.tar.gz
not installed | Extended-CMU-Seasons_slice10_query                 |
             https://data.ciirc.cvut.cz/public/projects/2020VisualLocalization/kapture//Extended-CMU-Seasons_slice10_query.tar.gz
...

~$ kapture_download_dataset.py install "Aachen*"
INFO  ::downloader: 3 dataset will be installed.
INFO  ::downloader: Aachen-Day-Night-v1.1_mapping: starting installation ...
/home/guerin/Aachen-Day-Night-v1.1_mapping.tar.gz:  2%|###                               | 143M/5.54G

```


Aachen Day-Night Dataset

Outdoor dataset captured in the city of Aachen in Germany.
Provides mapping data and query images in day and night conditions
<https://www.visuallocalization.net/>

Mapping Images



Query Images



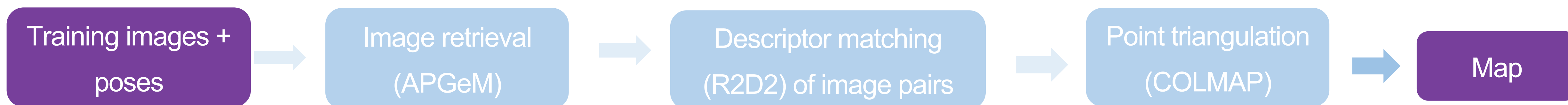
Aachen Day-Night Dataset in kapture format

Aachen/mapping/	Dataset root path
├─ sensors/	Sensor data root path
│ └─ sensors.txt	list of cameras with their parameters (intrinsics)
│ └─ trajectories.txt	extrinsics (timestamp, sensor, pose)
│ └─ records_camera.txt	images: timestamp, sensor and path
│ └─ records_data/	image folder
│ └─ db/1.jpg	
│ └─ db/2.jpg	
│ └─ ...	
└─ reconstruction/	(empty)

```
Aachen/query_day/
├─ sensors/
│   └─ sensors.txt
│   └─ records_camera.txt
│   └─ records_data/
│       └─ query/day/nexus4/IMG_20130210_163122.jpg
│       └─ ...
```

```
Aachen/query_night/
├─ sensors/...
```


Mapping Pipeline with Kapture



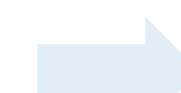
Training images +
poses



Image retrieval
(APGeM)



Descriptor matching
(R2D2) of image pairs



Point triangulation
(COLMAP)



Map

 /bin/bash

```
~$ kapture_extract_global_features.py -i mapping dirtorch
```

```
~$
```

```
Aachen/mapping/  
├─ sensors/  
└─ reconstruction/  
    └─ global_features/  
        ├── global_features.txt  
        ├── db/1.jpg.gfeat  
        ├── db/2.jpg.gfeat  
        └─ ...
```

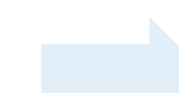

Training images +
poses



Image retrieval
(APGeM)



Descriptor matching
(R2D2) of image pairs



Point triangulation
(COLMAP)



Map

/bin/bash

```
~$ kapture_extract_global_features.py -i mapping dirtorch
~$ kapture_compute_image_pairs.py -i mapping -o mapping/pairfile.txt
image_retrieval --topk 5 --frustum-check
~$
```

Aachen/mapping/
├ sensors/
├ reconstruction/
└ pairfile.txt

/bin/bash

```
~$ cat training/pairfile.txt
db/1.jpg, db/2.jpg, 0.8620584011077881
db/1.jpg, db/1524.jpg, 0.8210515379905701
db/1.jpg, db/539.jpg, 0.8130918741226196
db/1.jpg, db/3.jpg, 0.8045650124549866
db/1.jpg, db/2405.jpg, 0.7997030019760132
db/2.jpg, db/3.jpg, 0.9318576455116272
db/2.jpg, db/1.jpg, 0.8620584011077881
...
```



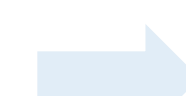

Training images +
poses



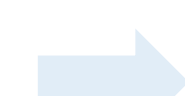
Image retrieval
(APGeM)



Descriptor matching
(R2D2) of image pairs



Point triangulation
(COLMAP)



Map



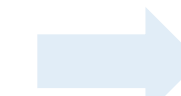
Training images +
poses



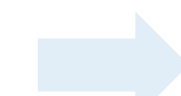
Image retrieval
(APGeM)



Descriptor matching
(R2D2) of image pairs



Point triangulation
(COLMAP)



Map

```
/bin/bash

~$ kapture_run_matching.py -i mapping pairsfile --pairsfile-path mapping/pairfile.txt
~$
```

```
Aachen/mapping/
├─ sensors/
├─ reconstruction/
│   └─ matches/
│       └─ 1.jpg.overlapping/
│           ├── 2.jpg.matches
│           ├── 1524.jpg.matches
│           ...
```

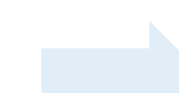

Training images +
poses



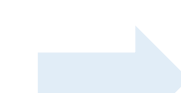
Image retrieval
(APGeM)



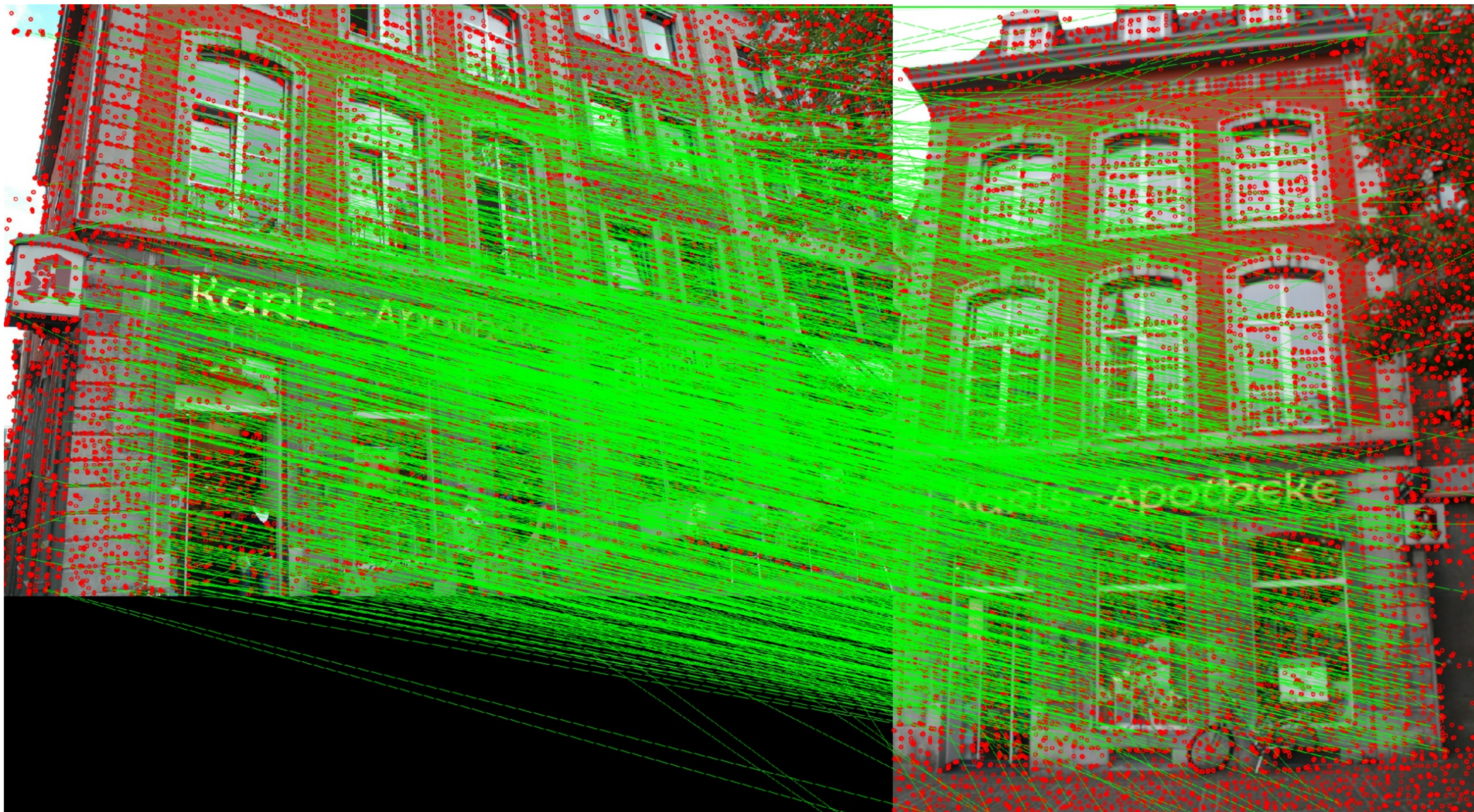
Descriptor matching
(R2D2) of image pairs



Point triangulation
(COLMAP)



Map



Visualization of matches between two similar images

Training images +
poses



Image retrieval
(APGeM)



Descriptor matching
(R2D2) of image pairs



Point triangulation
(COLMAP)



Map

/bin/bash

```
~$ kapture_colmap_build_map.py -i mapping -o colmap --pairs-file-path mapping/pairfile.txt
~$ kapture_import_colmap.py -db colmap/colmap.db -txt colmap/reconstruction -o model --skip_reconstruction
```

```
Aachen/model/
├─ sensors/
└─ reconstruction/
    ├─ observations.txt
    └─ points3d.txt
```

/bin/bash

```
~$ cat points3d.txt
# X, Y, Z, R, G, B
9.671530e+02, -5.930670e+00, -1.504399e+00, 8.200000e+01, 7.500000e+01, 6.000000e+01
7.210230e+02, 3.875140e+00, -2.108259e+02, 2.600000e+01, 3.900000e+01, 3.400000e+01
...
~$ cat observations.txt
# point3d_id, [image_path, feature_id]*
0, db/3048.jpg, 1895, db/3050.jpg, 3500, db/3046.jpg, 2166, db/3046.jpg, 9994
1, db/2403.jpg, 13693, db/81.jpg, 8966, db/2404.jpg, 10155
2, db/3376.jpg, 11590, db/3372.jpg, 13332, db/3380.jpg, 6762
```




Aachen Day-Night reconstruction based on 6700 images

Localization Pipeline with Kapture

Map + Query
Images

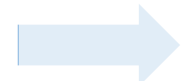
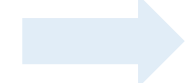


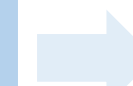
Image retrieval
(APGeM)



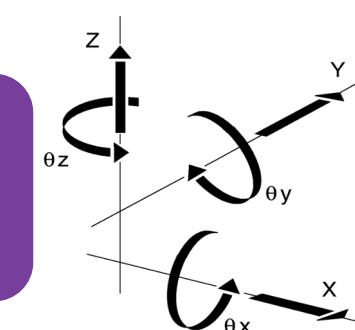
Descriptor matching
(R2D2) of image pairs

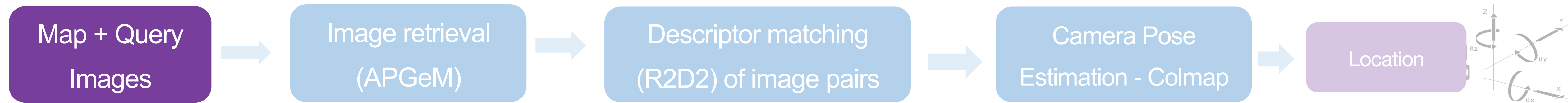


Camera Pose
Estimation - Colmap



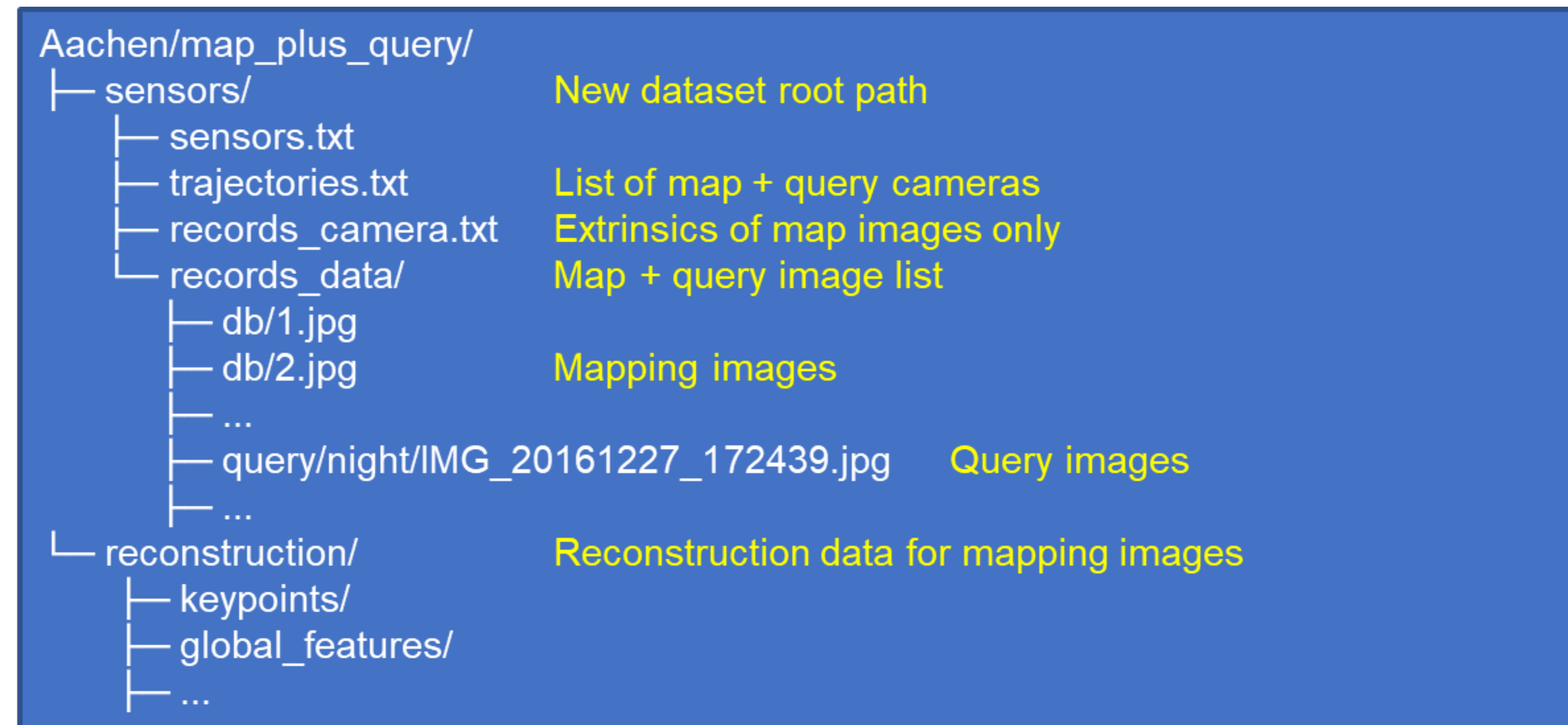
Location

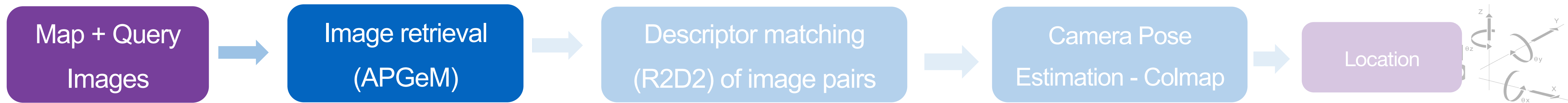




```
/bin/bash

~$ kapture_merge.py -y -i mapping query_night -o map_plus_query
~$
```





```
/bin/bash

~$ extract_global_features.py -i map_plus_query dirtorch
~$ compute_image_pairs.py -i map_plus_query -o map_plus_query/pairfile.txt
image_retrieval --topk 5 --frustum-check
~$ cat training/pairfile.txt
query/night/nexus5x/IMG_20161227_172439.jpg, db/1853.jpg, 0.5139860510826111
query/night/nexus5x/IMG_20161227_172439.jpg, db/1286.jpg, 0.5060933828353882
query/night/nexus5x/IMG_20161227_172439.jpg, db/1860.jpg, 0.5056981444358826
query/night/nexus5x/IMG_20161227_172439.jpg, db/1861.jpg, 0.5053839683532715
query/night/nexus5x/IMG_20161227_172439.jpg, db/1851.jpg, 0.5009866952896118
...
```



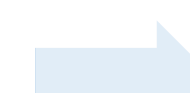

Map + Query
Images



Image retrieval
(APGeM)



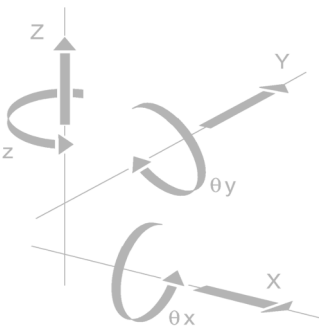
Descriptor matching
(R2D2) of image pairs

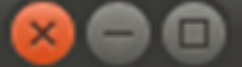


Camera Pose
Estimation - Colmap



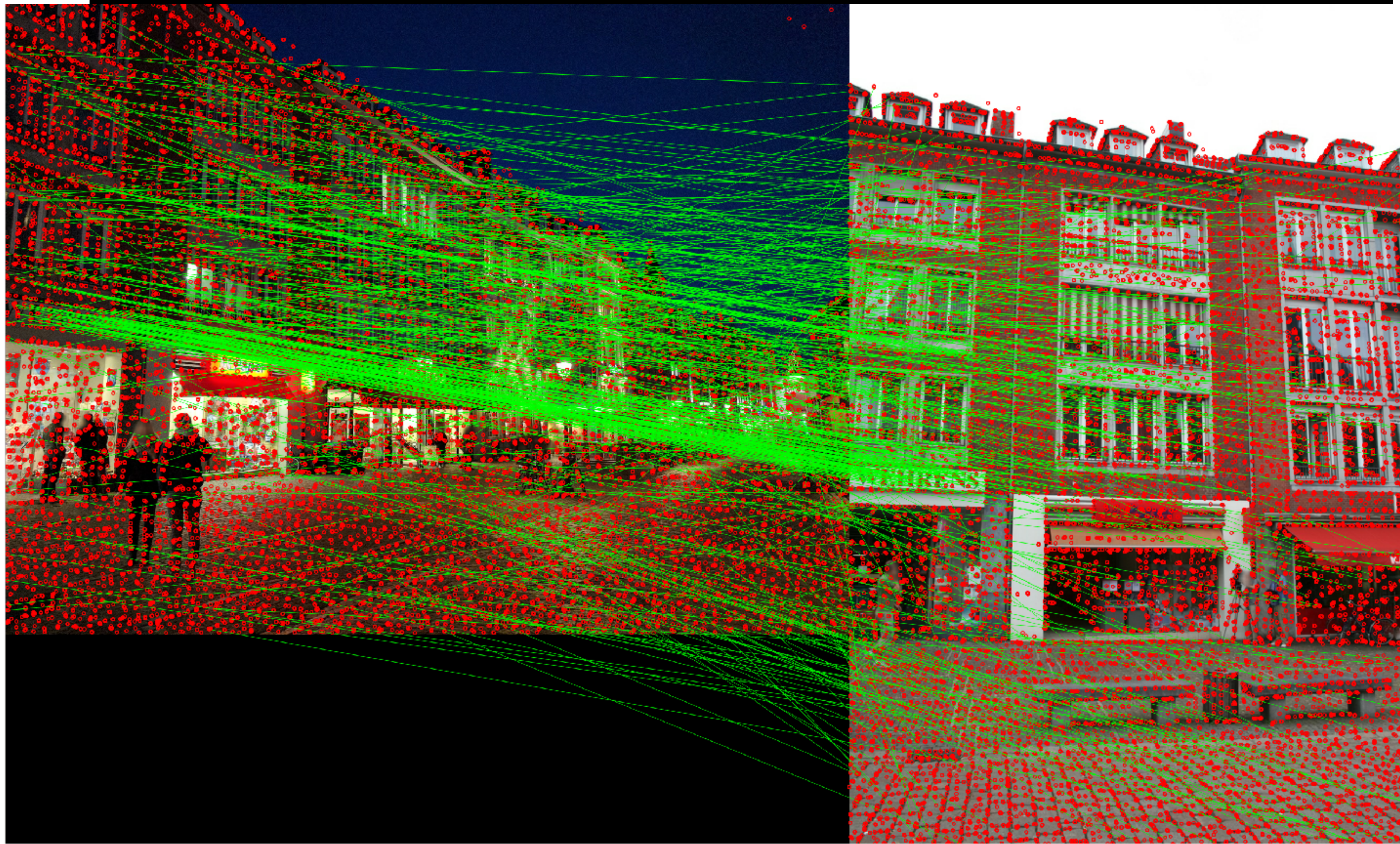
Location

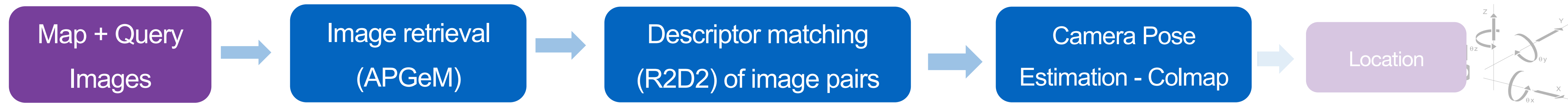


 /bin/bash

```
~$ extract_local_features.py -v 20 -i mapping r2d2
```

```
~$ run_matching.py -i mapping pairsfile --pairsfile-path mapping/pairsfile.txt
```

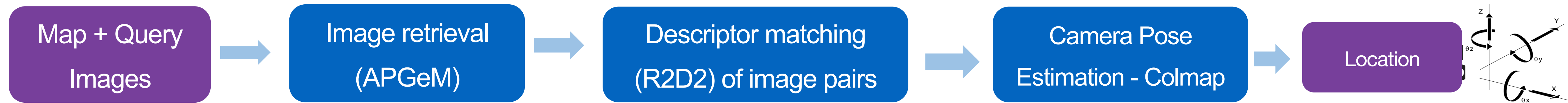




```
/bin/bash

~$ kapture_colmap_localize.py -i map_plus_query -db colmap/colmap.db -txt
colmap/reconstruction -o colmap_localized --pairs-file-path map_plus_query/pairfile.txt
~$ kapture_import_colmap.py -db colmap_localized/colmap.db -txt
colmap_localized/reconstruction -o model_localized --skip_reconstruction
~$ recover_timestamps.py -i model_localized -ref mapping -o localized
```

Aachen/localized/	Localized dataset root path
├─ sensors/	
│ └─ sensors.txt	
│ └─ trajectories.txt	Extrinsics of all localized images
│ └─ records_camera.txt	
│ └─ records_data/	
└─ reconstruction/	(empty)

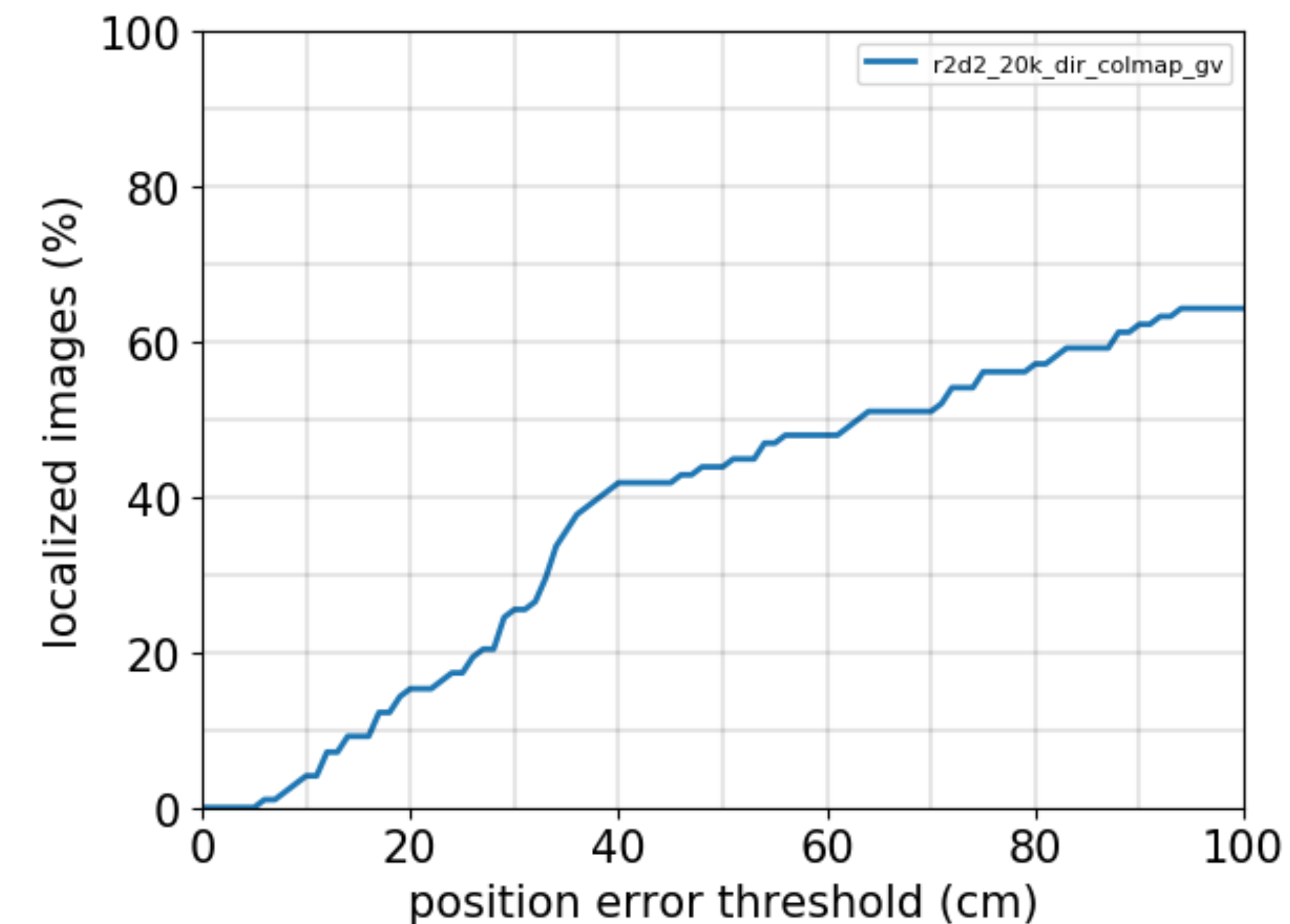


```
/bin/bash

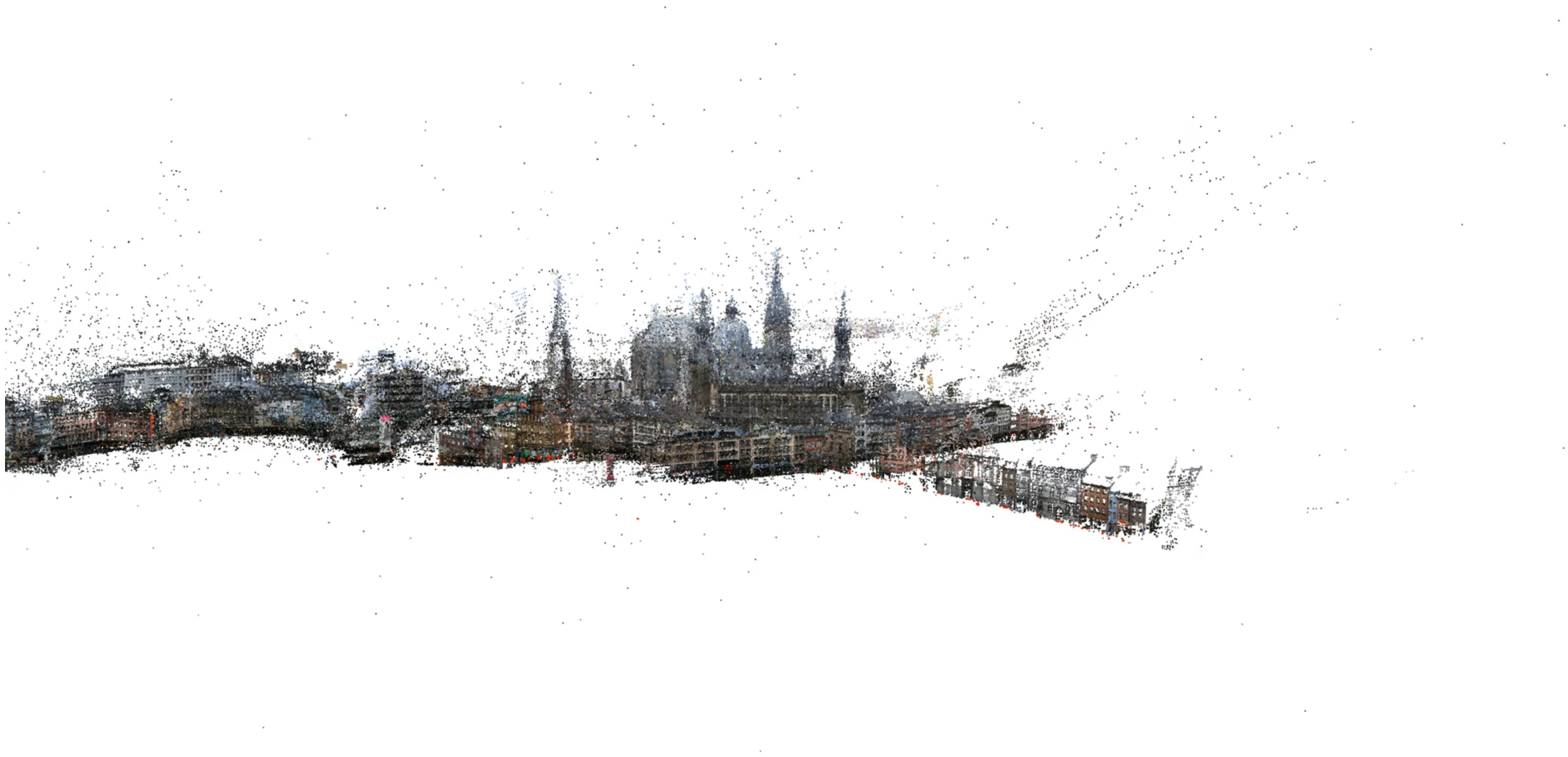
~$ kapture_evaluate.py -i localized --labels colmap -gt ground_truth -o localized/eval
~$ cat localized/eval/stats.txt
Model: r2d2_20k_dir_colmap_gv

Found 98 / 98 image positions (100.00 %).
Found 98 / 98 image rotations (100.00 %).
Localized images: mean=(4.4421m, 6.3671 deg) / median=(0.6270m, 1.3195 deg)
All: median=(0.6270m, 1.3195 deg)
Min: 0.0571m; 0.1931 deg
Max: 215.1474m; 166.2683 deg

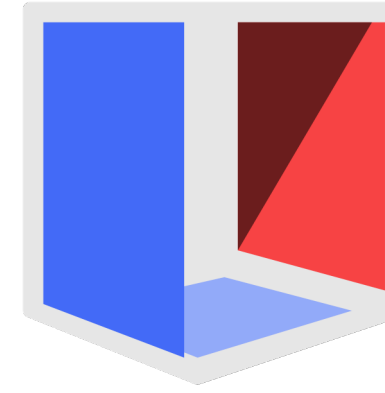
(0.5m, 2.0 deg): 42.86%
(1.0m, 5.0 deg): 61.22%
(5.0m, 10.0 deg): 84.69%
```



Aachen Map built with kapture pipeline



kapture next steps



We already have a few ideas:

- Extend kapture to other localization methods
- Provide simpler tools to run a Visual Localization pipeline
- Add more options: local features, global features, matching algorithms, etc.
- Increase number of downloadable datasets
- Encourage external usage and contributions.

Do you want to contribute?

- Use kapture and give us feedback!
- Provide new datasets
- Integrate kapture support in your pipeline or tools
- Improve kapture

If you are interested, we are very happy to help, so don't hesitate to contact us.



Thank You



If you want to know more about kapture:

Code and specifications <https://github.com/naver/kapture>

More code and tools <https://github.com/naver/kapture-localization>

R2D2 local features <https://github.com/naver/r2d2>

ApGem global features <https://github.com/naver/deep-image-retrieval>

Latest News <https://europe.naverlabs.com/blog/kapture>

