# Image-Based Localization

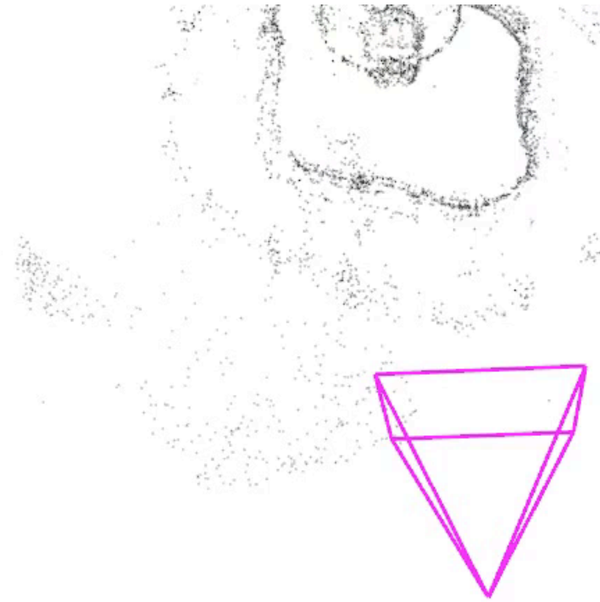
## An Introduction

## GVG 2021 - Lecture 06

CZECH INSTITUTE
OF INFORMATICS
ROBOTICS AND
CYBERNETICS
CTU IN PRAGUE

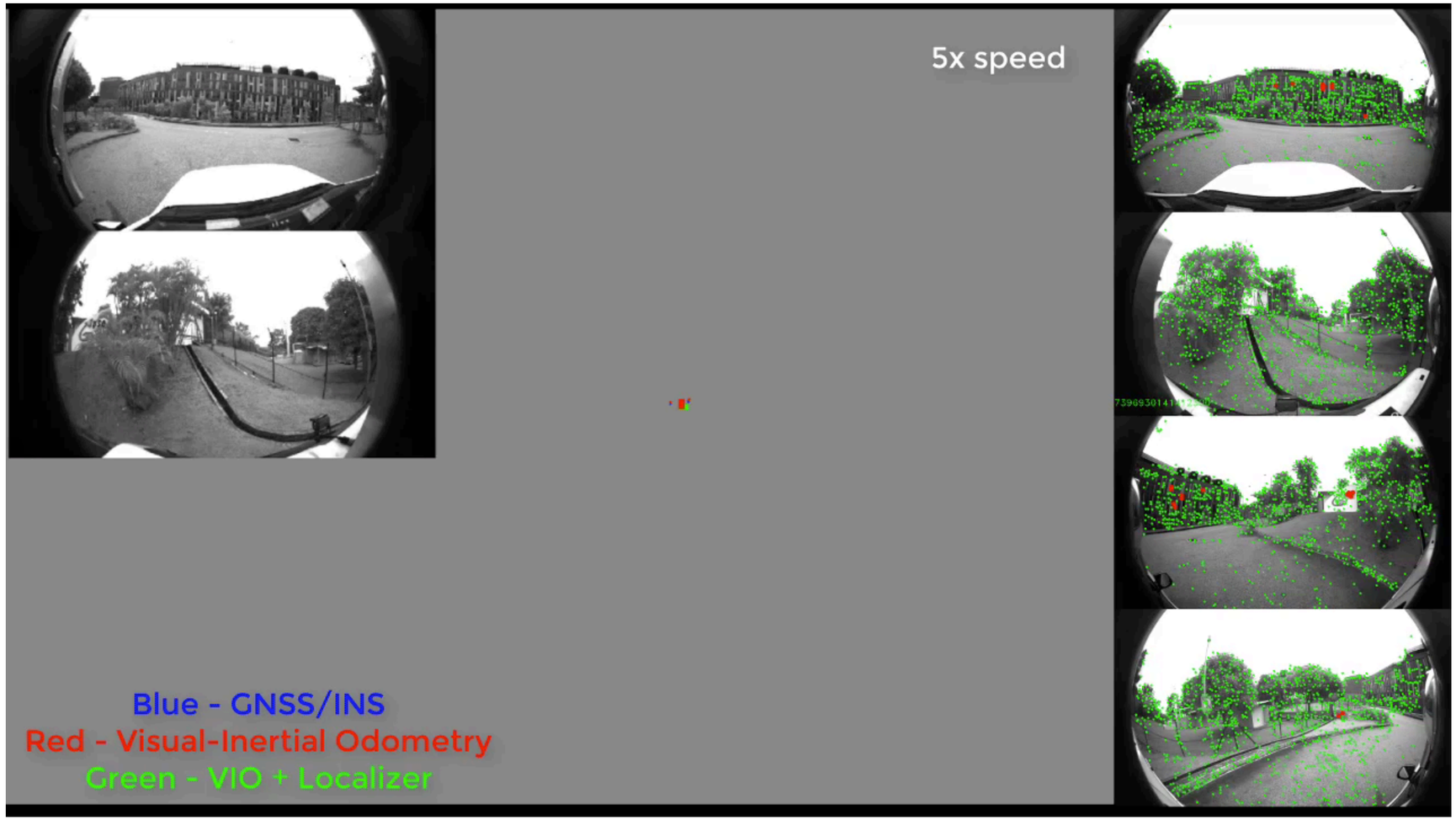
Torsten Sattler

# The Visual Localization Problem

Compute **exact position and orientation** of query image

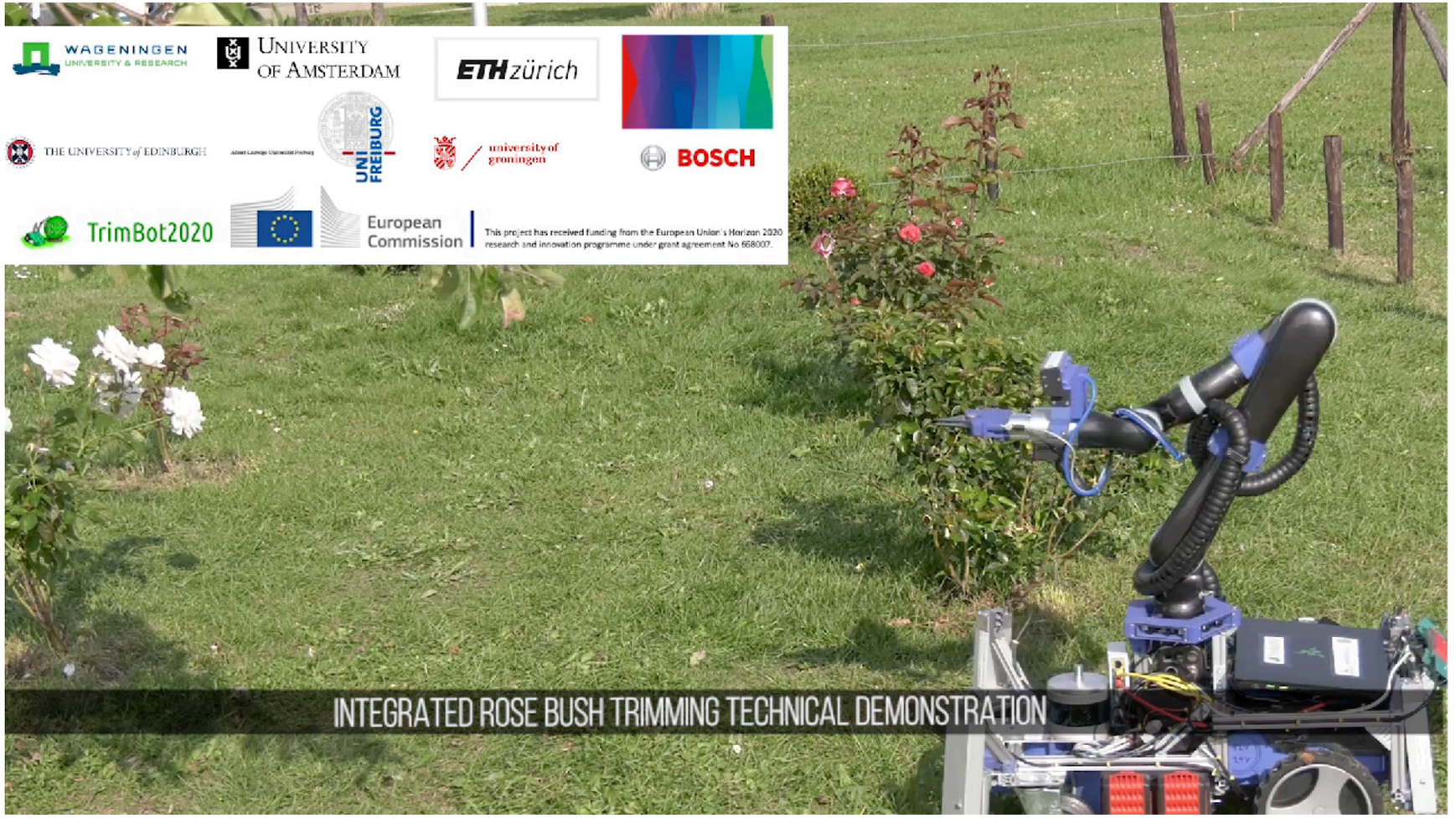
# Applications: Autonomous Driving



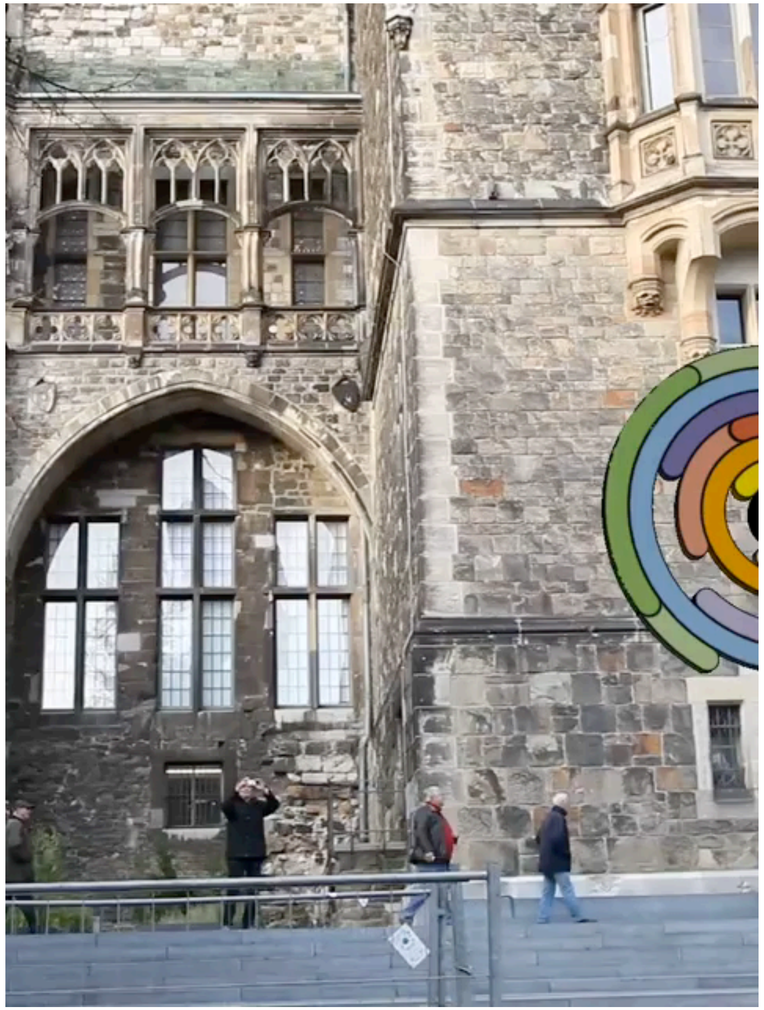

AutoVision
3D Vision for Autonomous Vehicles

[Geppert, Liu, Cui, Pollefeys, Sattler, Efficient 2D-3D Matching for Multi-Camera Visual Localization, ICRA 2019]

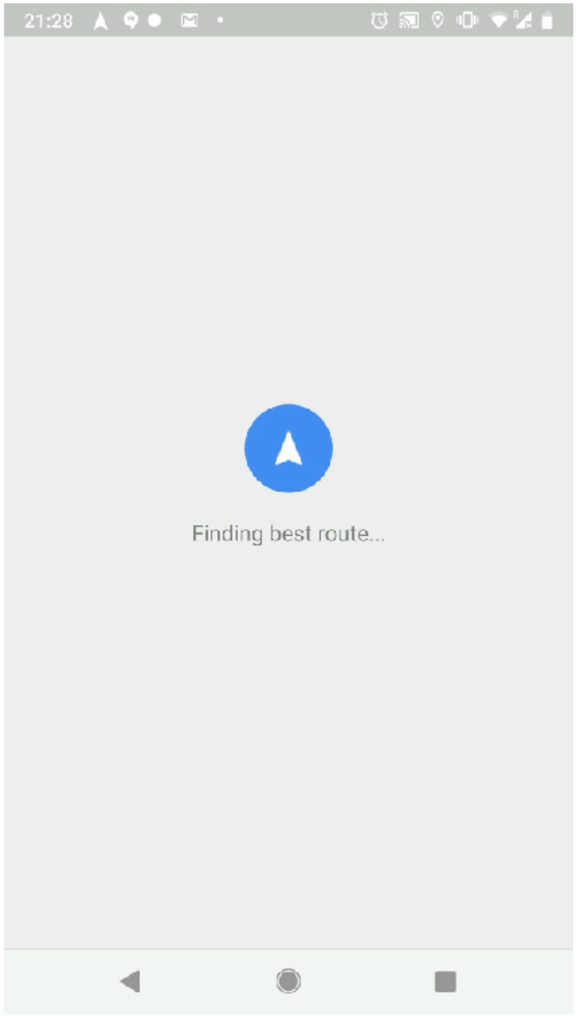
# Applications: Robotics



# Applications: Augmented Reality

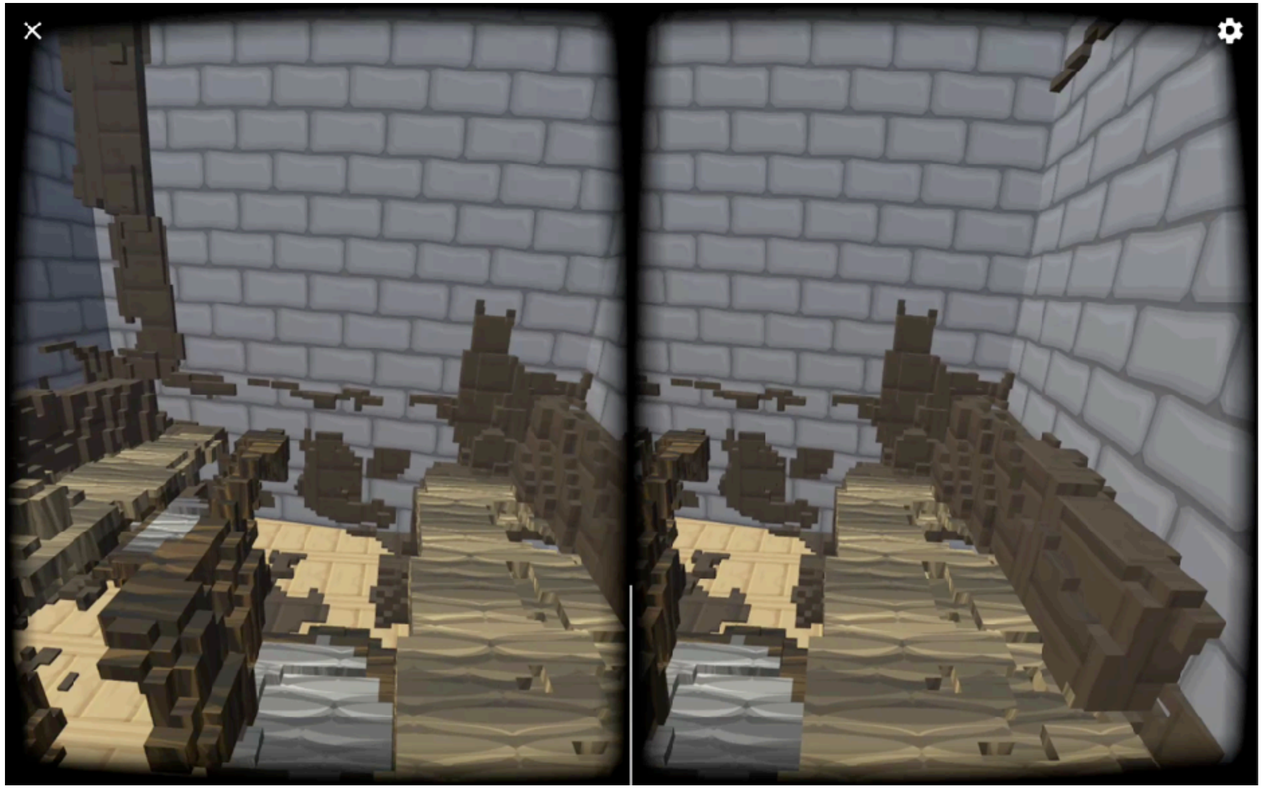


[Middelberg, Sattler, Untzelmann, Kobbelt, Scalable 6-DOF Localization on Mobile Devices, ECCV 2014]




AR navigation in Google Maps

# Applications: Virtual Reality



Master thesis by Benjamin Steger, ETH Zurich
Textures: PureBDcraft ResourcePack by Sphax - BDcraft.net

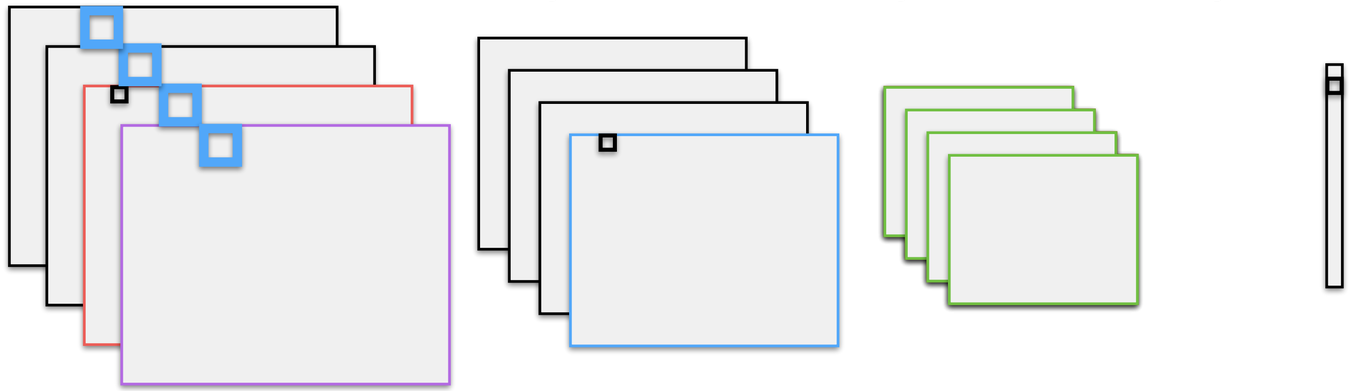
# Overview

- **A (Too) Simple Approach to Visual Localization**
- Structure-Based Localization
- Long-Term Localization
- Privacy-Preserving Localization

# Brief Introduction to Convolutional Neural Networks

all parameters **learned from data**

layer 1 | layer 2 | layer n | fully connected layer



convolution | convolution | convolution

… output

prediction:
“persian cat”

**loss**

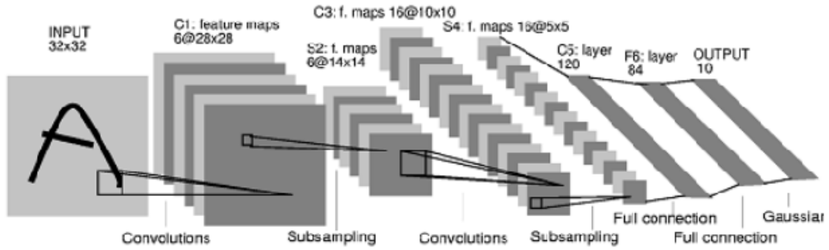
actual:
“tabby cat”

adjust parameters ← adjust parameters ← adjust parameters ← adjust parameters

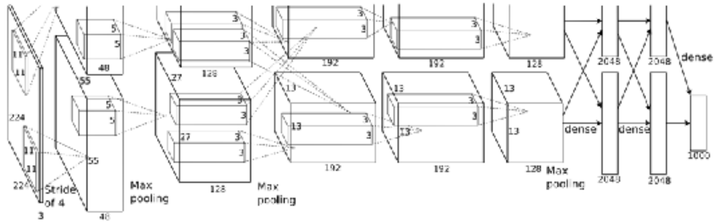
**backpropagation**
(gradient descend based on chain rule)

# Deep Learning

©Buena Vista Pictures

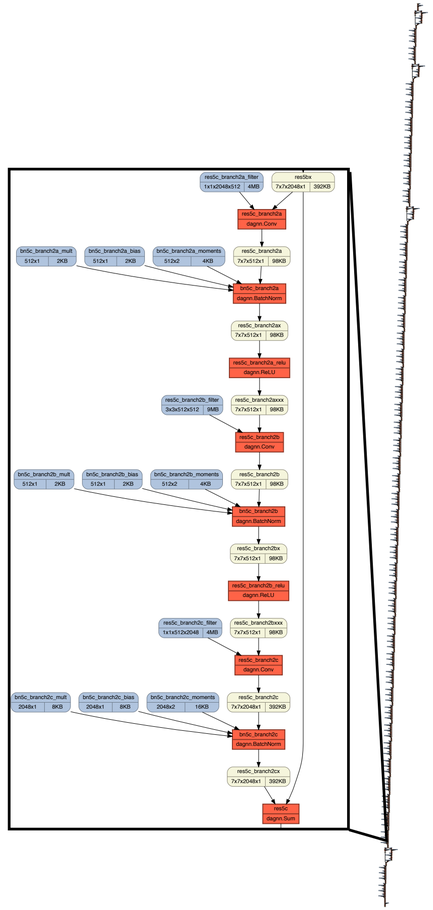



[LeCun et al. 1998]: 7 layers



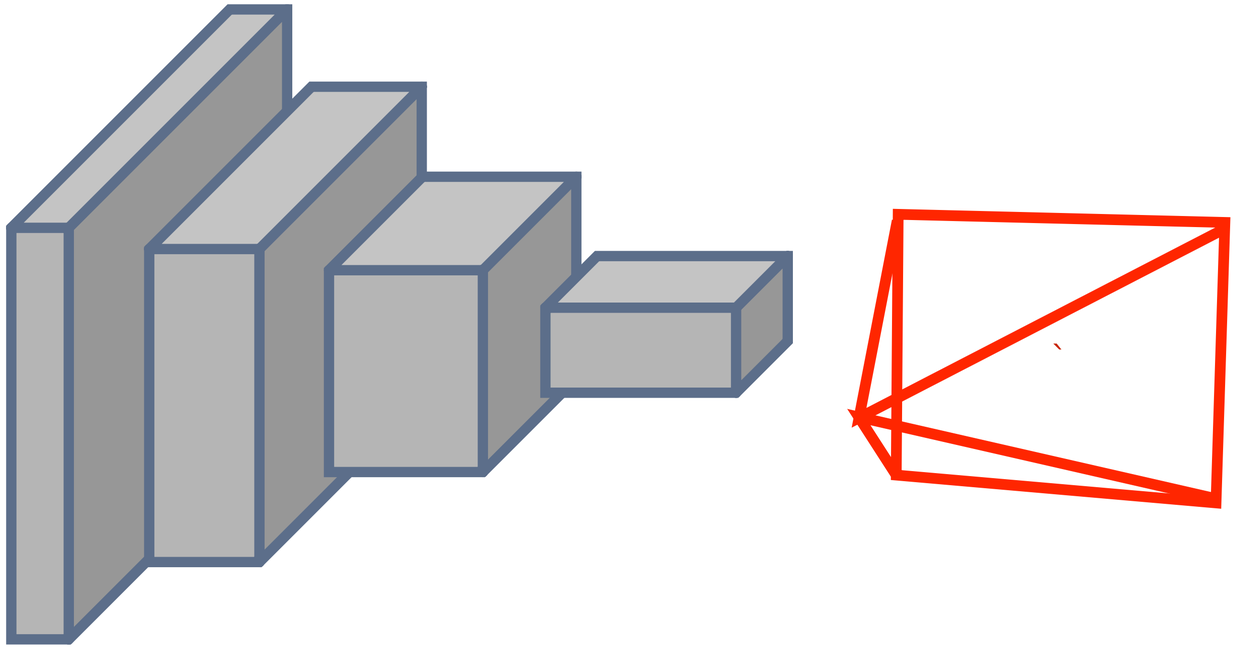

[Krizhevsky et al. 2012]: 8 layers

[Szegedy et al. 2014]: 22 layers



[He et al. 2015]:
152 layers

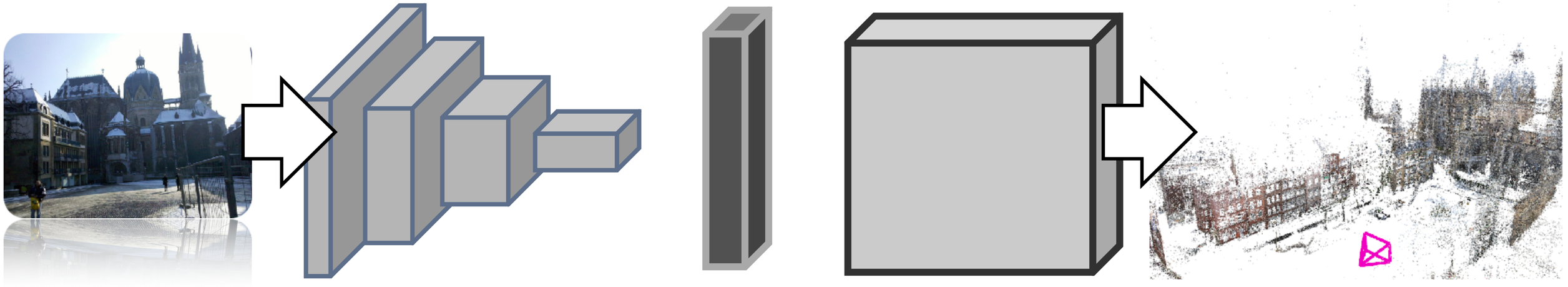
# A Simple Approach to Visual Localization



Convolutional Neural Network

Camera Pose Regression

# Camera Pose Regression



**Feature Extraction** **Embedding** **Projection** **Pose (position, orientation)**

# Training Data

( position **c**, orientation **r** )

( position **c**, orientation **r** )

# Loss Functions

- Hand-tuned [1]: $||\hat{c} - c|| + \beta||\hat{r} - r||$

position error (meters)

orientation error (degrees)

scaling factor trading off position and orientation errors

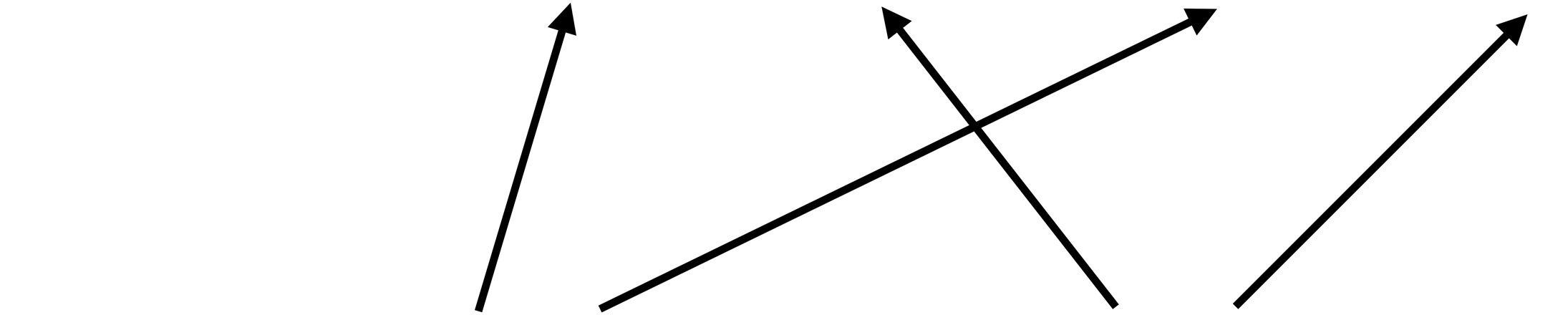
(needs to be set per scene)

[1] [Kendall et al., PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization, ICCV 2015]
[2] [Kendall & Cipolla, Geometric Loss Functions for Camera Pose Regression with Deep Learning, CVPR 2017]

slide credit: Eric Brachmann

# Loss Functions

- Hand-tuned [1]: $||\hat{c} - c|| + \beta||\hat{r} - r||$
- Self-tuned [2]: $||\hat{c} - c||\exp(-\hat{s}_c) + \hat{s}_c + ||\hat{r} - r||\exp(-\hat{s}_r) + \hat{s}_r$



weighting for the position / orientation errors

regularization (prevent loss of zero through large parameters)

[1] [Kendall et al., PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization, ICCV 2015]
[2] [Kendall & Cipolla, Geometric Loss Functions for Camera Pose Regression with Deep Learning, CVPR 2017]

slide credit: Eric Brachmann

# Loss Functions

- Hand-tuned [1]: $||\hat{c} - c|| + \beta||\hat{r} - r||$
- Self-tuned [2]: $||\hat{c} - c||\exp(-\hat{s}_c) + \hat{s}_c + ||\hat{r} - r||\exp(-\hat{s}_r) + \hat{s}_r$
- Reprojection error [2] (needs pre-training to converge):

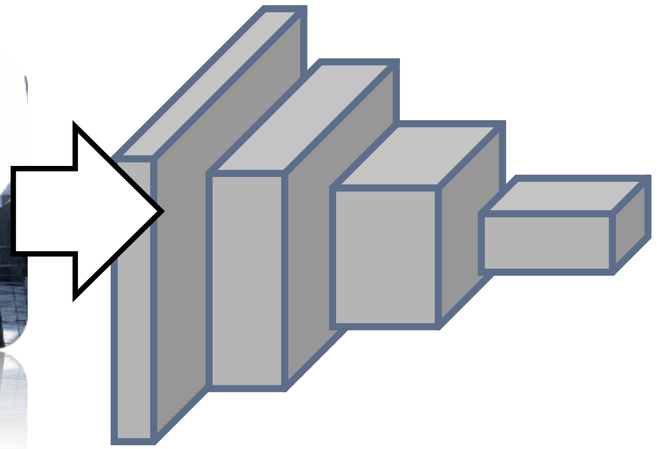
[1] [Kendall et al., PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization, ICCV 2015]
[2] [Kendall & Cipolla, Geometric Loss Functions for Camera Pose Regression with Deep Learning, CVPR 2017]

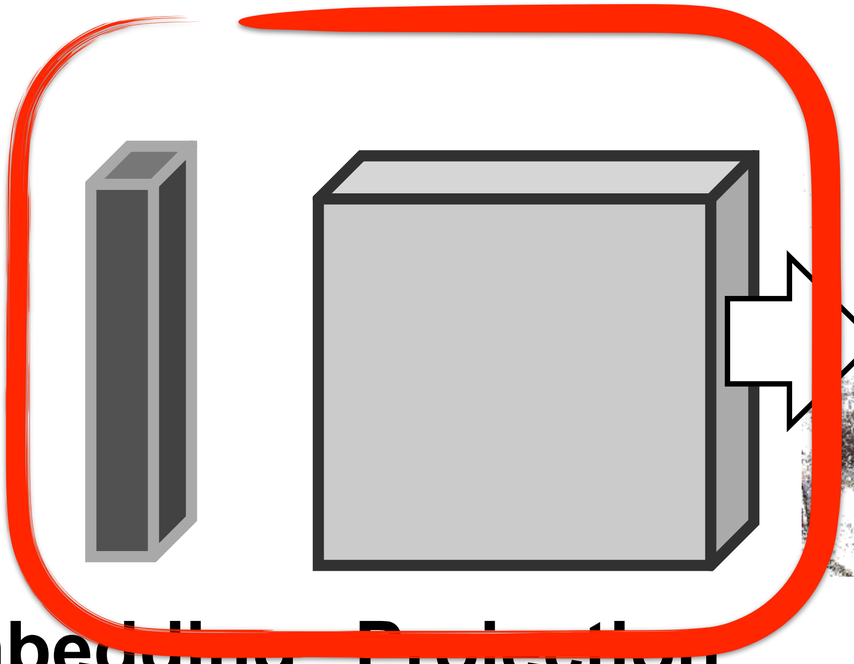
slide credit: Eric Brachmann

# Two Questions

- What do Pose Regression CNNs learn?
- How well do they work?

# Camera Pose Regression





**Feature Extraction**

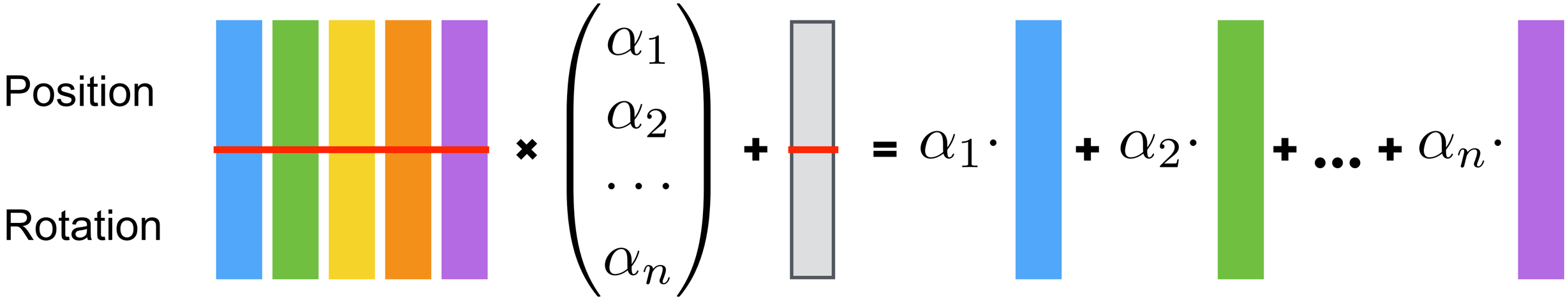
**Embedding**

**Projection**

**Pose (position, orientation)**

# Looking Inside The Black Box

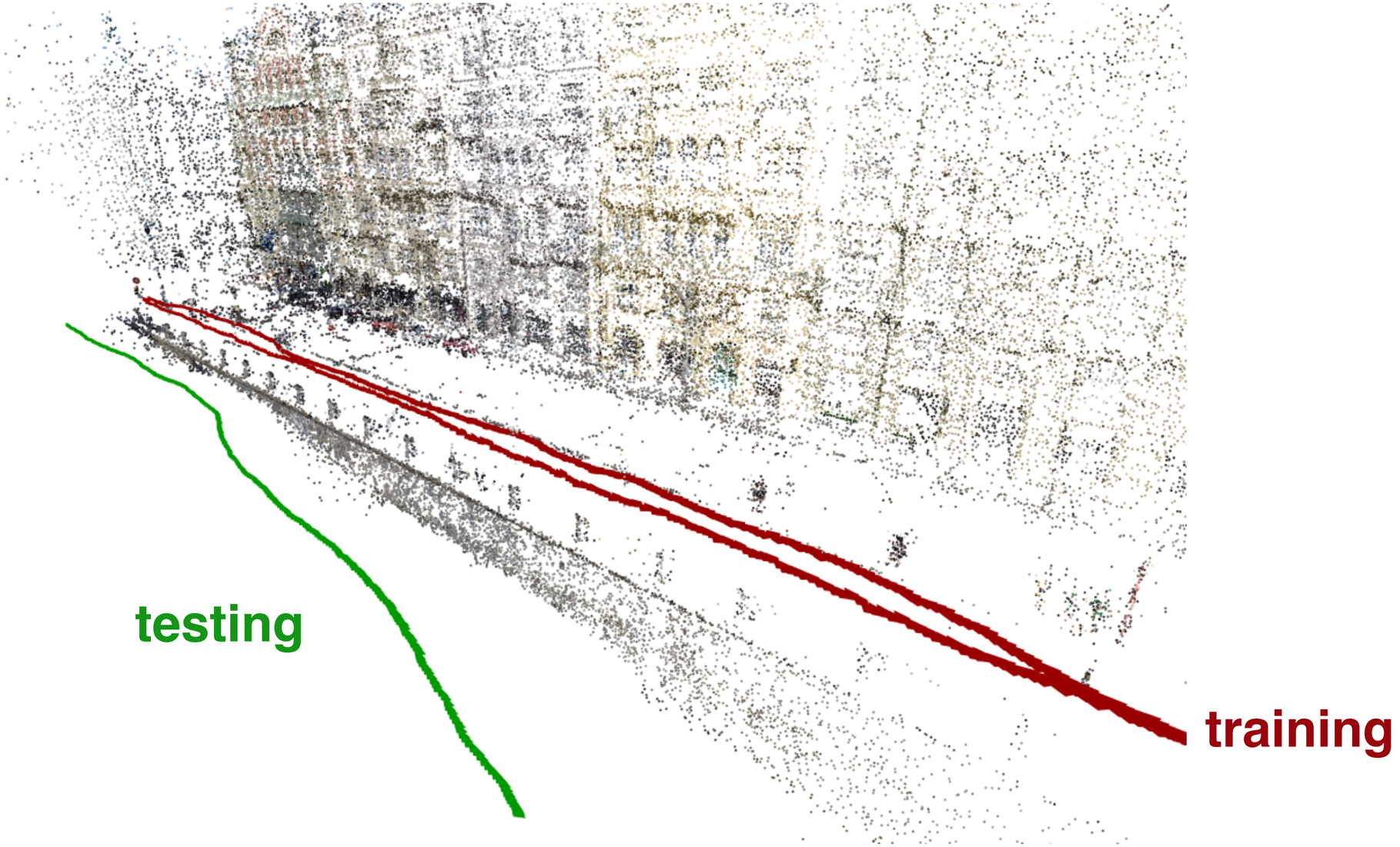
- Pose regression in **last FC layer** as **linear combination of base poses**:




**Projection** **Embedding** **Bias** **Predicted Camera Pose**

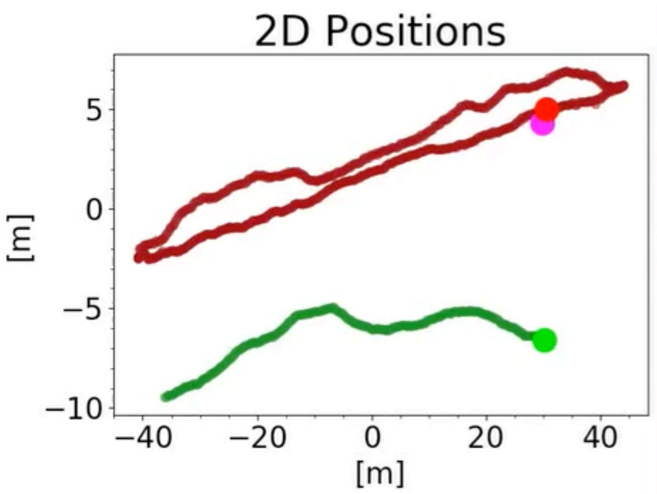
[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

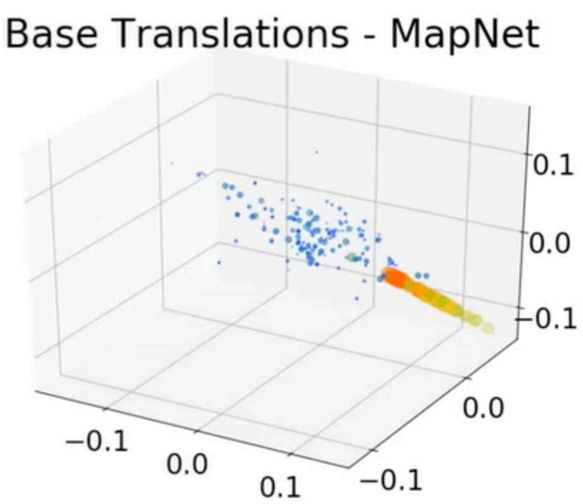
# Camera Pose Regression Example




[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

# Camera Pose Regression Example




GT training poses

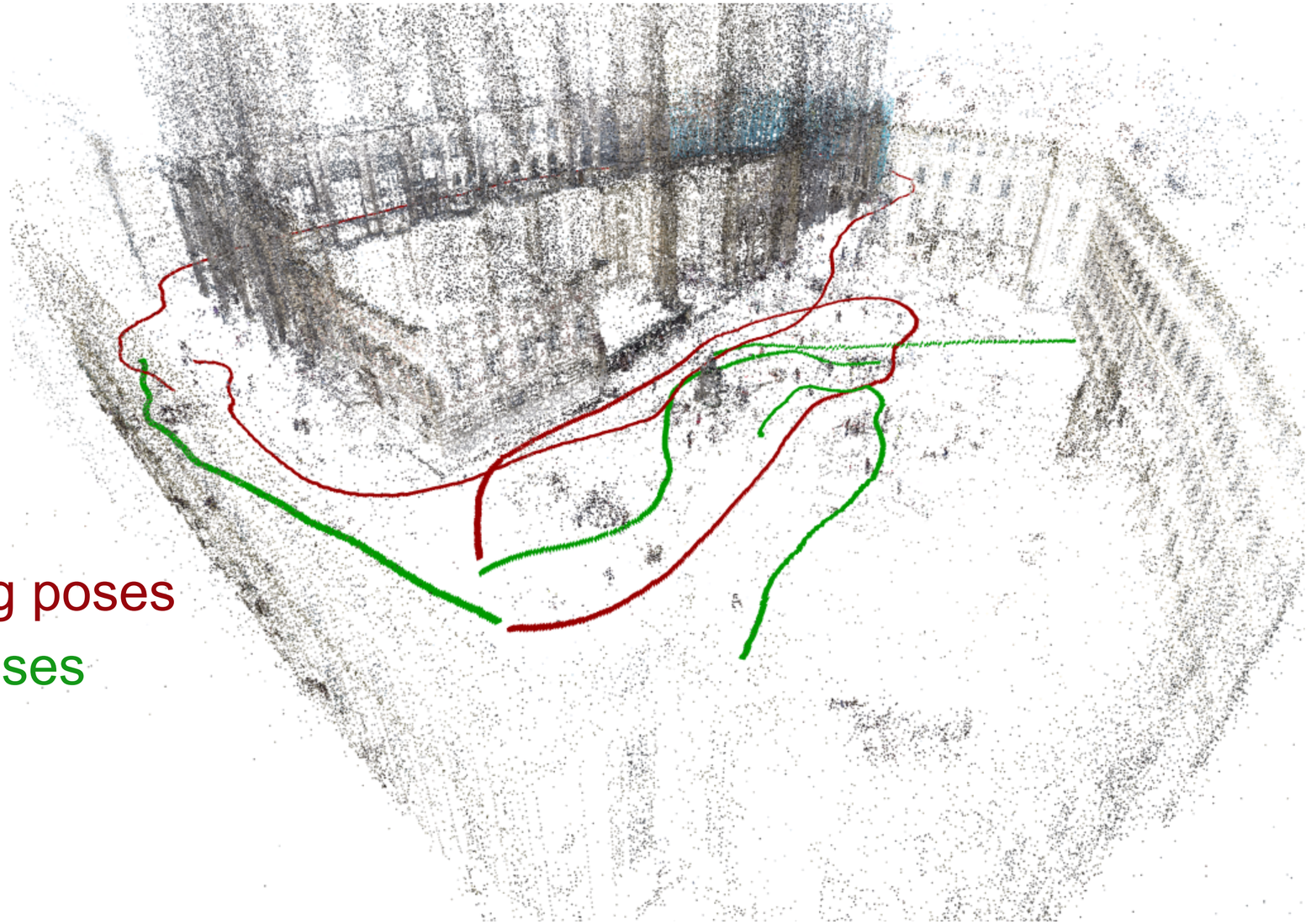
GT test poses

Predicted test pose

Pose of most similar training image

[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

# Camera Pose Regression Example

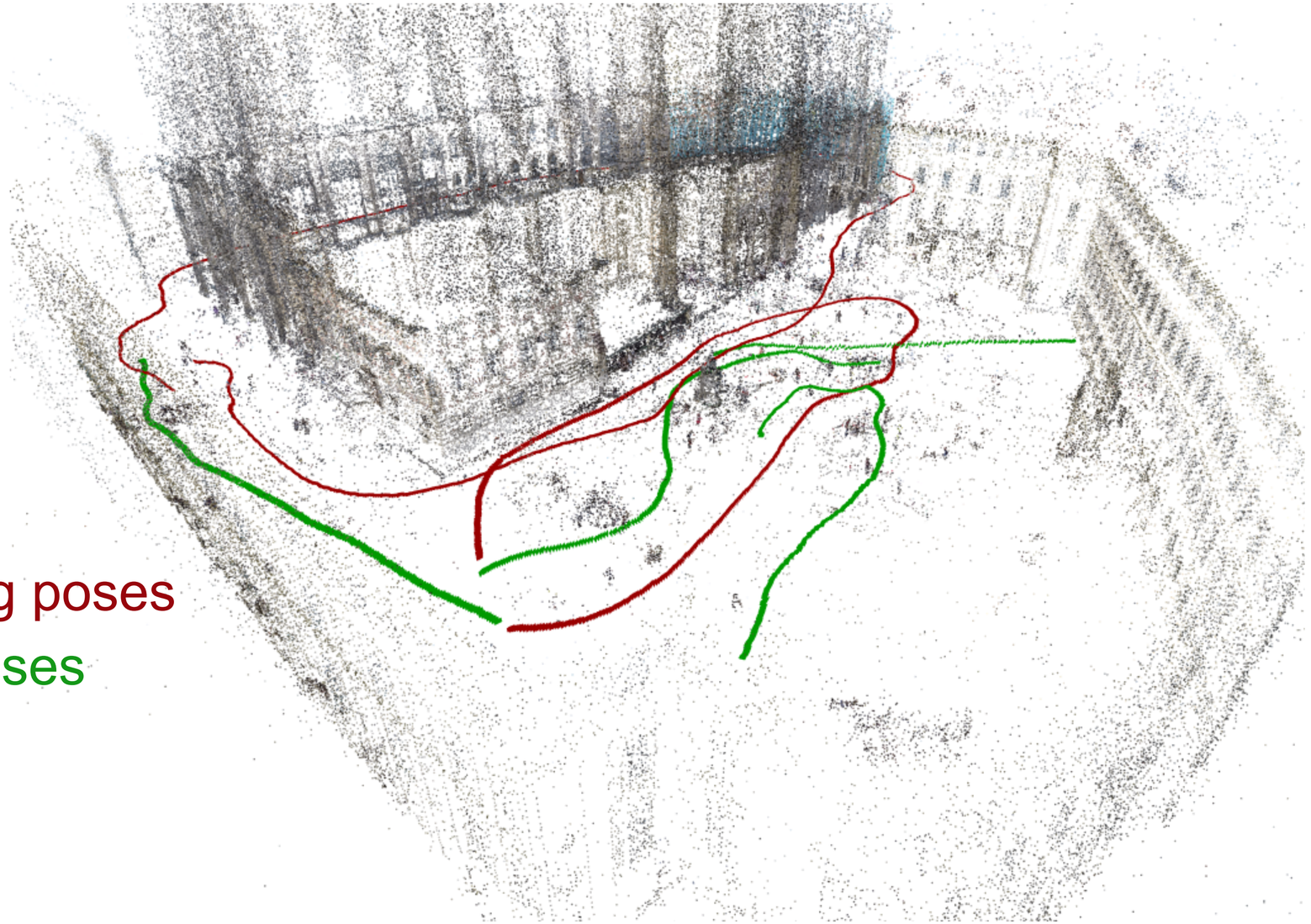


GT training poses

GT test poses

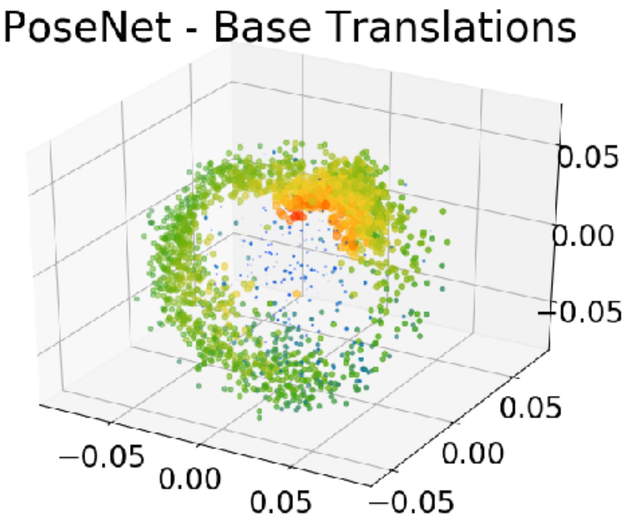
[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

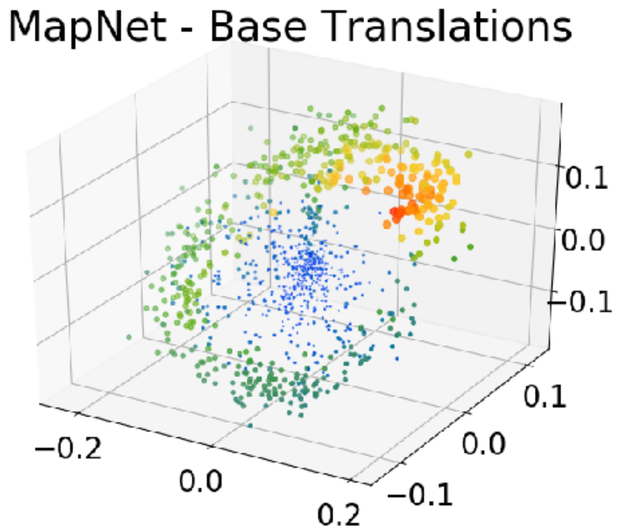
# Camera Pose Regression Example



GT training poses

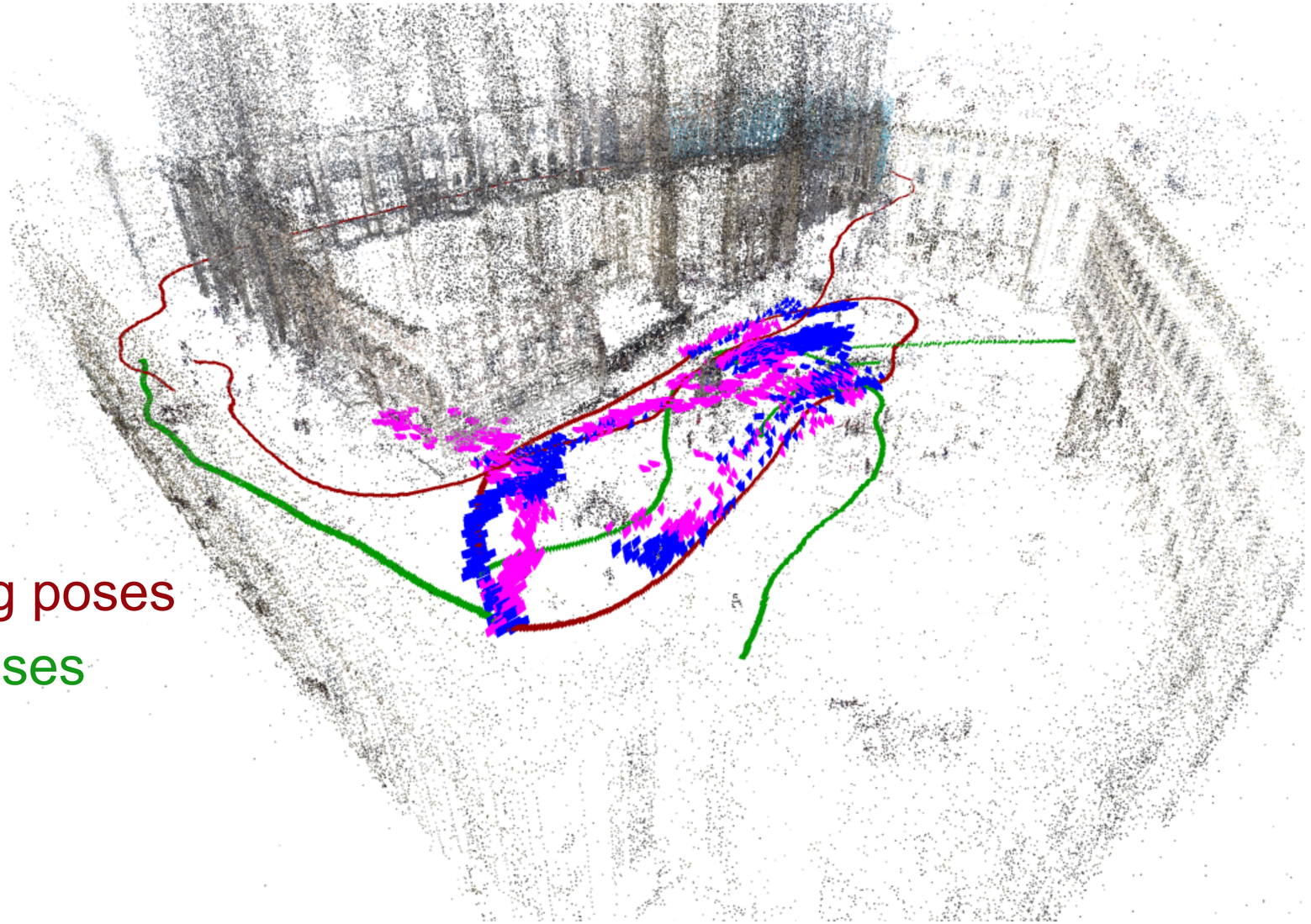
GT test poses




[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

# Camera Pose Regression Example

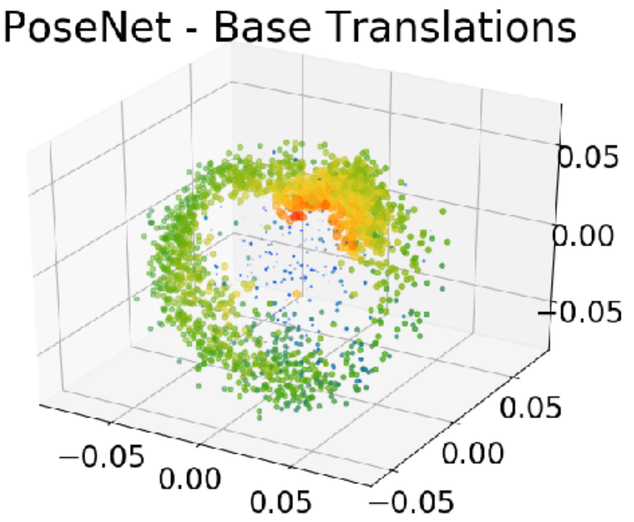


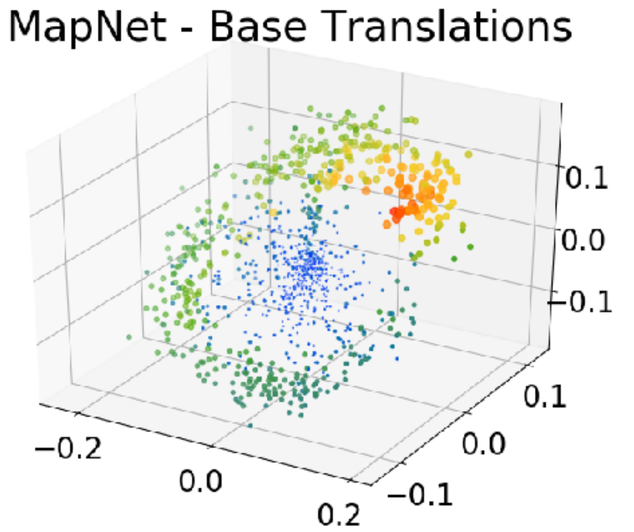
GT training poses

GT test poses

MapNet

PoseNet



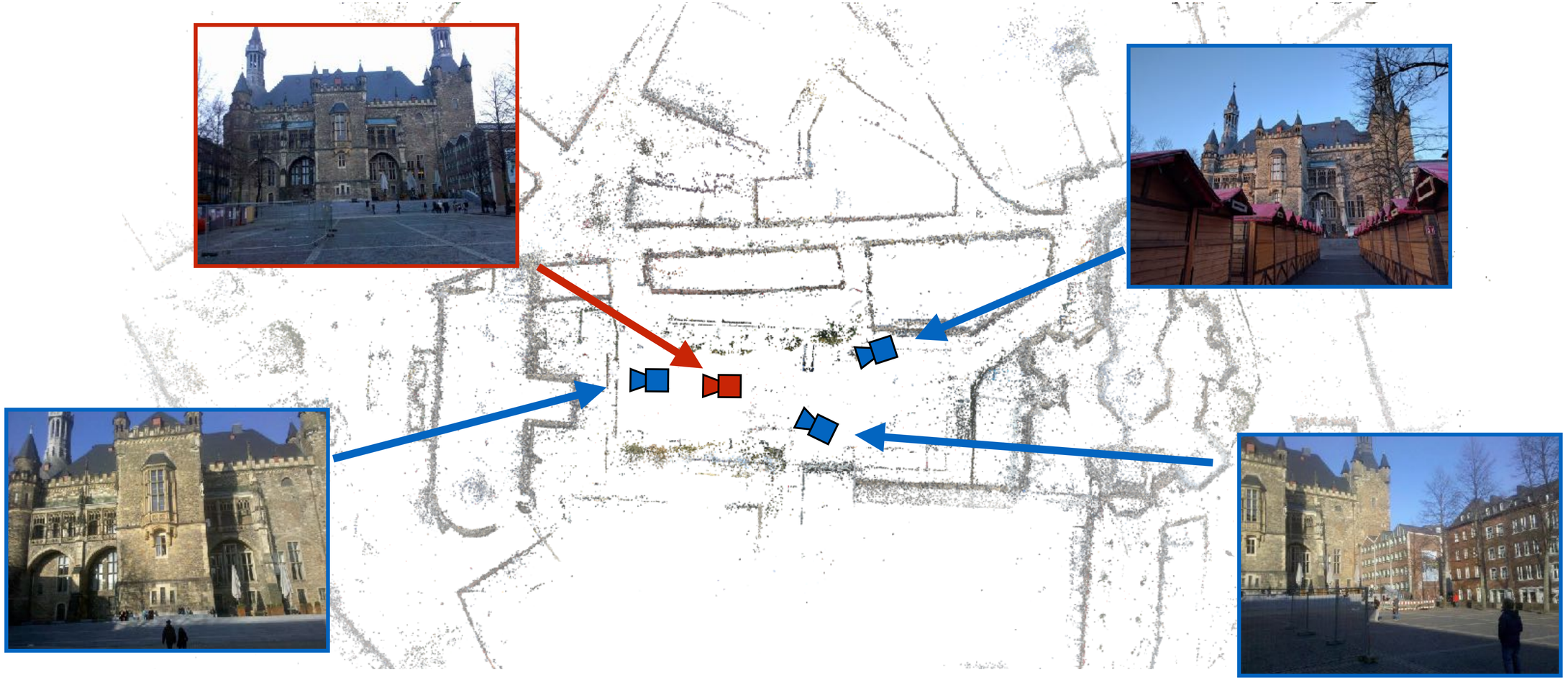

[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

# Two Questions

- What do Pose Regression CNNs learn?
  - A set of base poses and how to combine them based on visual features into camera poses.
- How well do they work?

# Baseline 1: Image Retrieval

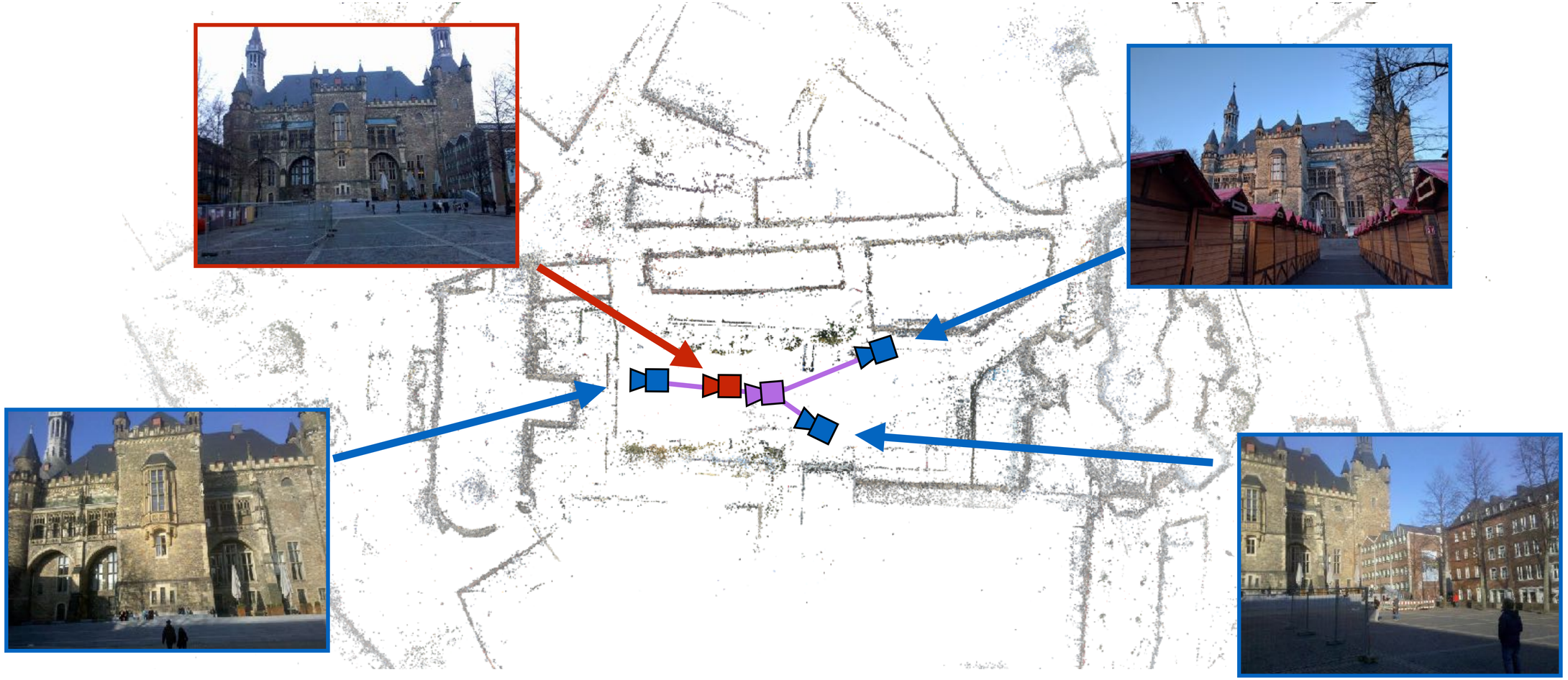
- **Approximate** pose of **test** image via **training** image poses



[Torii, Sivic, Pajdla, Visual localization by linear combination of image descriptors, ICCV Workshops 2011]

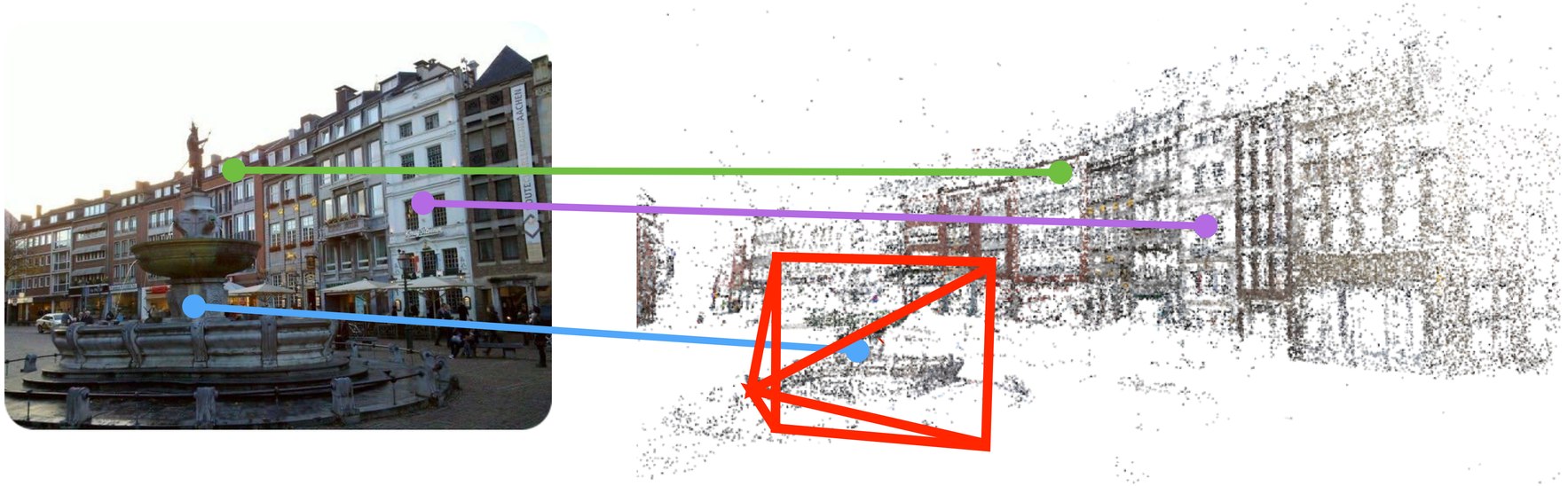
# Baseline 1: Image Retrieval

- **Approximate** pose of **test** image via **training** image poses



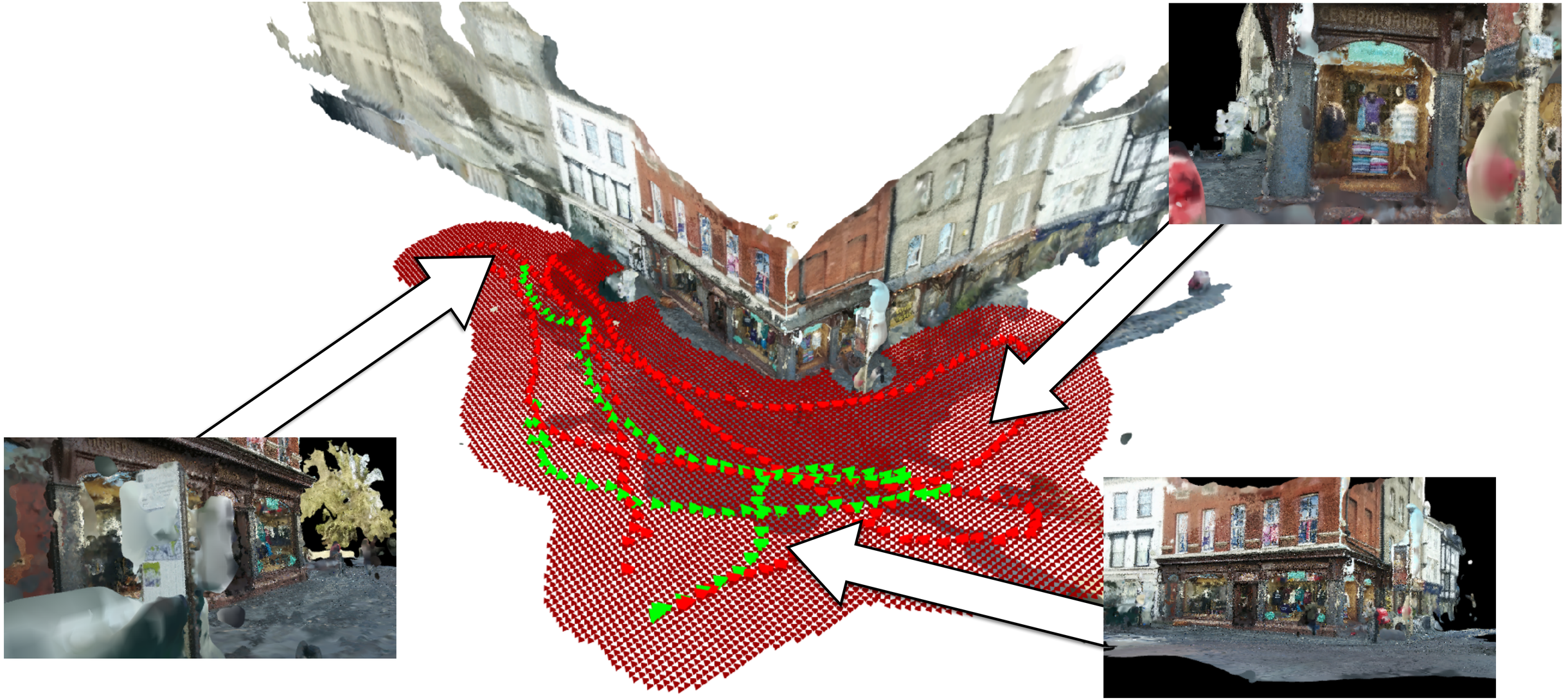
[Torii, Sivic, Pajdla, Visual localization by linear combination of image descriptors, ICCV Workshops 2011]

# Baseline 2: Structure-Based Localization



Structure-based Localization

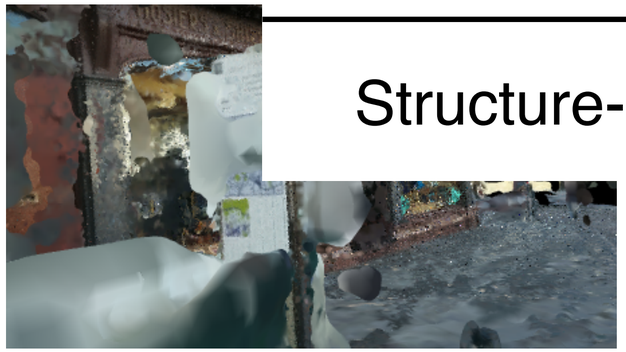
# Synthetic Data

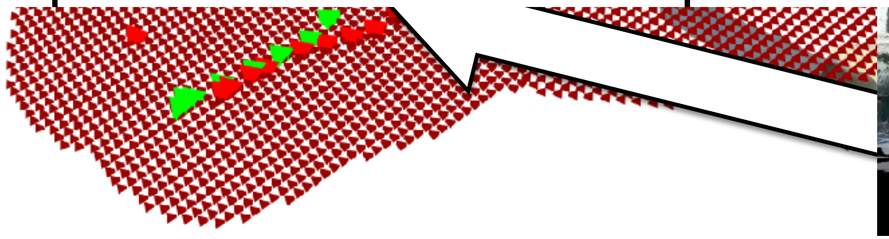


[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

# Synthetic Data

| | 203 Training images | 9,412 Training images |
|---|---|---|
| MapNet | 1.07m / 4.70deg | 0.33m / 1.46deg |
| Image Retrieval | 0.89m / 5.71deg | 0.38m / 6.41deg |
| Structure-Based | 0.01m / 0.04deg | |





[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

# On Real Data

| | Cambridge Landmarks | | | | | 7 Scenes | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Kings | Old | Shop | St. Mary's | Street | Chess | Fire | Heads | Office | Pumpkin | Kitchen | Stairs |
| PoseNet (PN) [30] | 1.92/5.40 | 2.31/5.38 | 1.46/8.08 | 2.65/8.48 | | 0.32/8.12 | 0.47/14.4 | 0.29/12.0 | 0.48/7.68 | 0.47/8.42 | 0.59/8.64 | 0.47/13.8 |
| PN learned weights [29] | 0.99/1.06 | 2.17/2.94 | 1.05/3.97 | 1.49/3.43 | 20.7/25.7 | 0.14/4.50 | 0.27/11.8 | 0.18/12.1 | 0.20/5.77 | 0.25/4.82 | 0.24/5.52 | 0.37/10.6 |
| Bay. PN [28] | 1.74/4.06 | 2.57/5.14 | 1.25/7.54 | 2.11/8.38 | | 0.37/7.24 | 0.43/13.7 | 0.31/12.0 | 0.48/8.04 | 0.61/7.08 | 0.58/7.54 | 0.48/13.1 |
| geo. PN [29] | 0.88/1.04 | 3.20/3.29 | 0.88/3.78 | 1.57/3.32 | 20.3/25.5 | 0.13/4.48 | 0.27/11.3 | 0.17/13.0 | 0.19/5.55 | 0.26/4.75 | 0.23/5.35 | 0.35/12.4 |
| LSTM PN [76] | 0.99/3.65 | 1.51/4.29 | 1.18/7.44 | 1.52/6.68 | | 0.24/5.77 | 0.34/11.9 | 0.21/13.7 | 0.30/8.08 | 0.33/7.00 | 0.37/8.83 | 0.40/13.7 |
| GPoseNet [12] | 1.61/2.29 | 2.62/3.89 | 1.14/5.73 | 2.93/6.46 | | 0.20/7.11 | 0.38/12.3 | 0.21/13.8 | 0.28/ 8.83 | 0.37/6.94 | 0.35/8.15 | 0.37/12.5 |
| SVS-Pose [50] | 1.06/2.81 | 1.50/4.03 | 0.63/5.73 | 2.11/8.11 | | | | | | | | |
| Hourglass PN [44] | | | | | | 0.15/6.17 | 0.27/10.8 | 0.19/11.6 | 0.21/8.48 | 0.25/7.01 | 0.27/10.2 | 0.29/12.5 |
| BranchNet [78] | | | | | | 0.18/5.17 | 0.34/8.99 | 0.20/14.2 | 0.30/7.05 | 0.27/5.10 | 0.33/7.40 | 0.38/10.3 |
| MapNet [11] | 1.07/1.89 | 1.94/3.91 | 1.49/4.22 | 2.00/4.53 | | 0.08/3.25 | 0.27/11.7 | 0.18/13.3 | 0.17/5.15 | 0.22/4.02 | 0.23/4.93 | 0.30/12.1 |
| MapNet+ [11] | | | | | | 0.10/3.17 | 0.20/9.04 | 0.13/11.1 | 0.18/5.38 | 0.19/3.92 | 0.20/5.01 | 0.30/13.4 |
| MapNet+PGO [11] | | | | | | 0.09/3.24 | 0.20/9.29 | 0.12/8.45 | 0.19/5.42 | 0.19/3.96 | 0.20/4.94 | 0.27/10.6 |
| DenseVLAD [71] | 2.80/5.72 | 4.01/7.13 | 1.11/7.61 | 2.31/8.00 | 5.16/23.5 | 0.21/12.5 | 0.33/13.8 | 0.15/14.9 | 0.28/11.2 | 0.31/11.3 | 0.30/12.3 | 0.25/15.8 |
| DenseVLAD + Inter. | 1.48/4.45 | 2.68/4.63 | 0.90/4.32 | 1.62/6.06 | 15.4/25.7 | 0.18/10.0 | 0.33/12.4 | 0.14/14.3 | 0.25/10.1 | 0.26/9.42 | 0.27/11.1 | 0.24/14.7 |

Not consistently better than image retrieval!

[Sattler, Zhou, Pollefeys, Leal-Taixé, Understanding the Limitations of CNN-based Absolute Camera Pose Regression, CVPR 2019]

# Two Questions

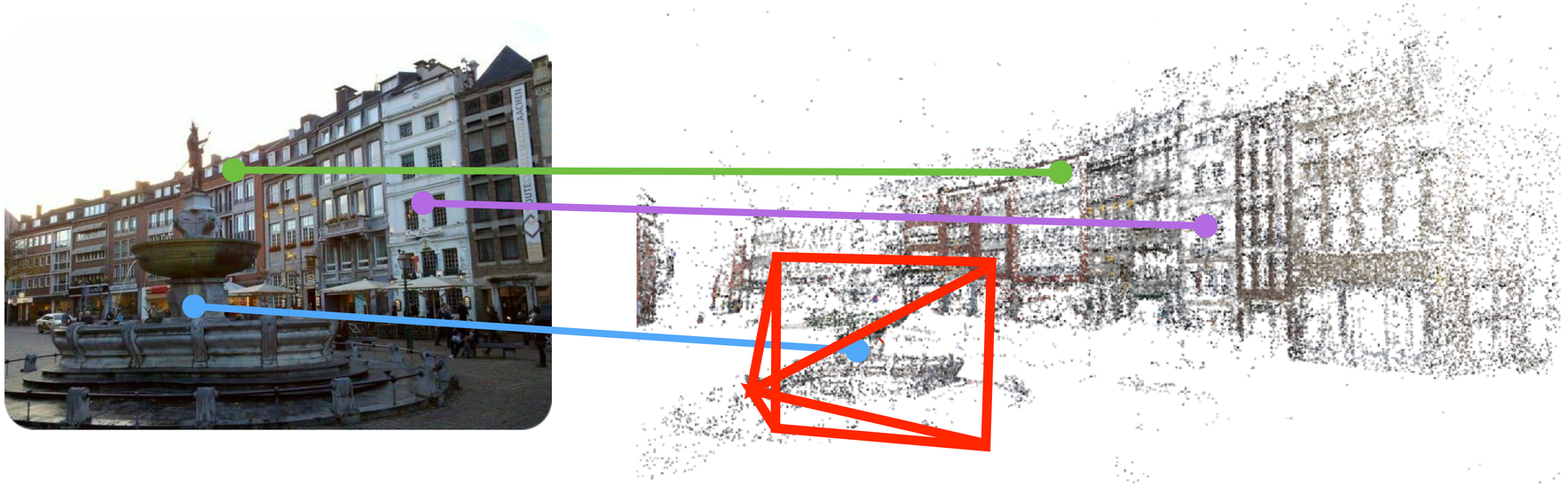
- What do Pose Regression CNNs learn?
  - A set of base poses and how to combine them based on visual features into camera poses.
- How well do they work?
  - Not much better than simple pose approximation via image retrieval.

# Quiz Time

# Overview

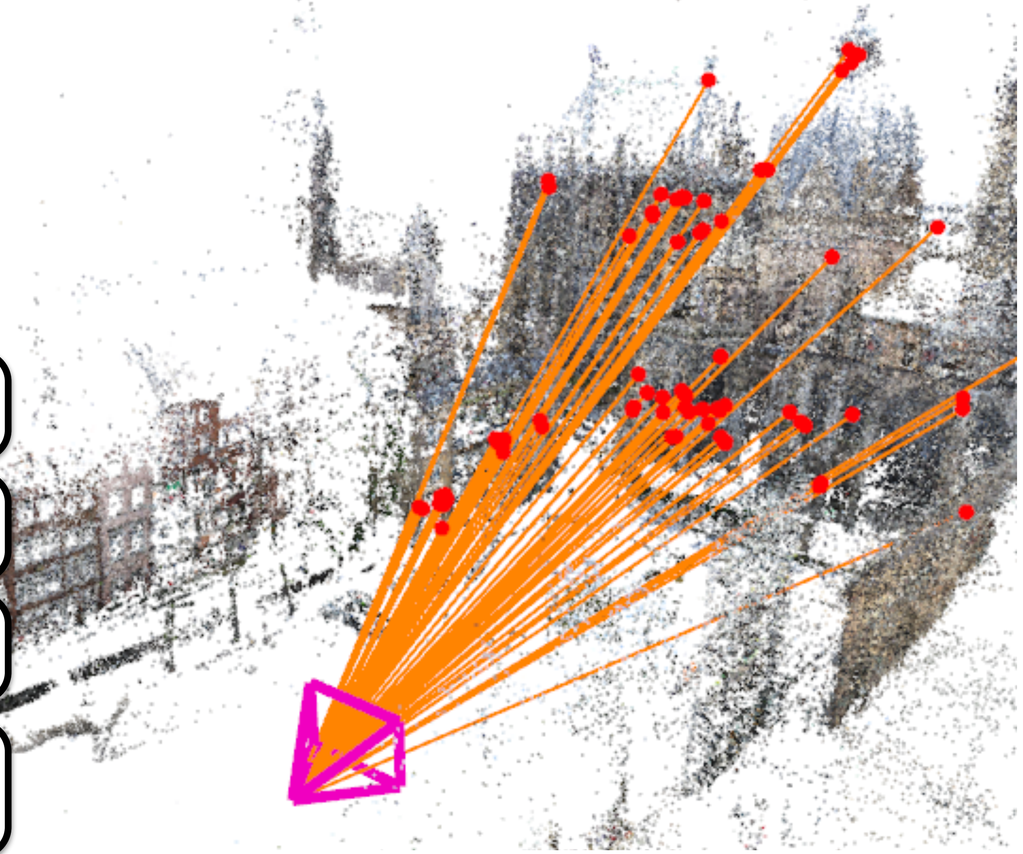
- A (Too) Simple Approach to Visual Localization
- **Structure-Based Localization**
- Long-Term Localization
- Privacy-Preserving Localization

# Structure-Based Localization



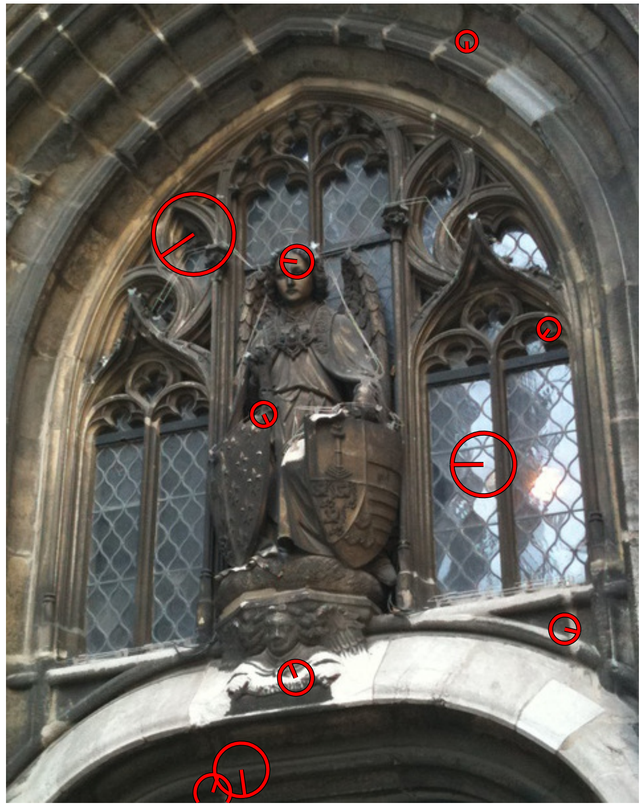
Structure-based Localization

# Local Feature-based Localization



- Feature Detection
- Feature Description
- Descriptor Matching for 2D-3D Matching
- Estimate Camera Pose (RANSAC + n-point-pose solver)

# Feature Detection



Keypoint Detection

- Scale-Invariant Feature Transform (**SIFT**):
  - Detect keypoints at multiple scales → zooming in or out will produce the same keypoints
  - Assign each keypoint an orientation → rotating the image will not change the keypoints

[Lowe, Distinctive Image Features from Scale-Invariant Keypoints, IJCV 2004]

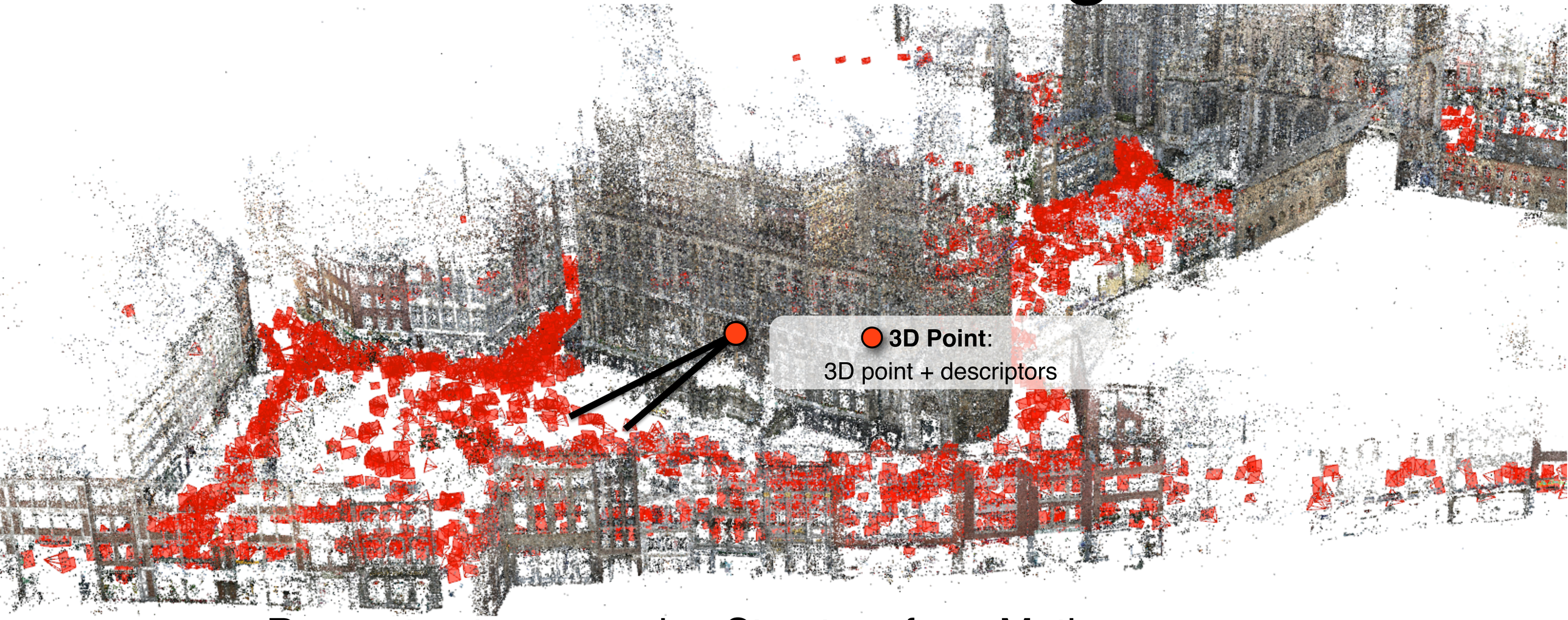
# Feature Description

Keypoint Detection

- Scale-Invariant Feature Transform (**SIFT**):
  - Consider region around keypoint
  - Size of region depends on scale
  - Orientation of region depends on keypoint orientation
  - Compute a **descriptor** (high-dimensional vector) from the patch, e.g., 128-dimensional for SIFT

[Lowe, Distinctive Image Features from Scale-Invariant Keypoints, IJCV 2004]

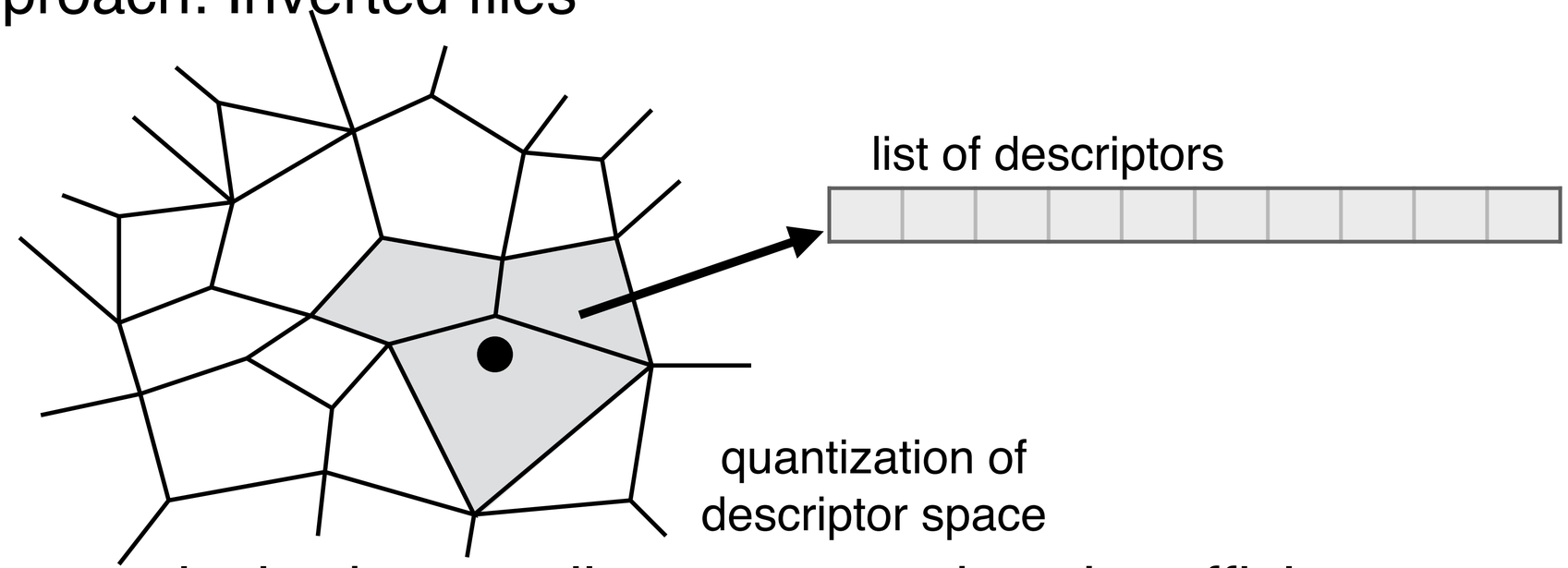
# 2D-3D Matching




- Reconstruct scene using Structure-from-Motion
- Associate each 3D point with local image descriptors (SIFT)

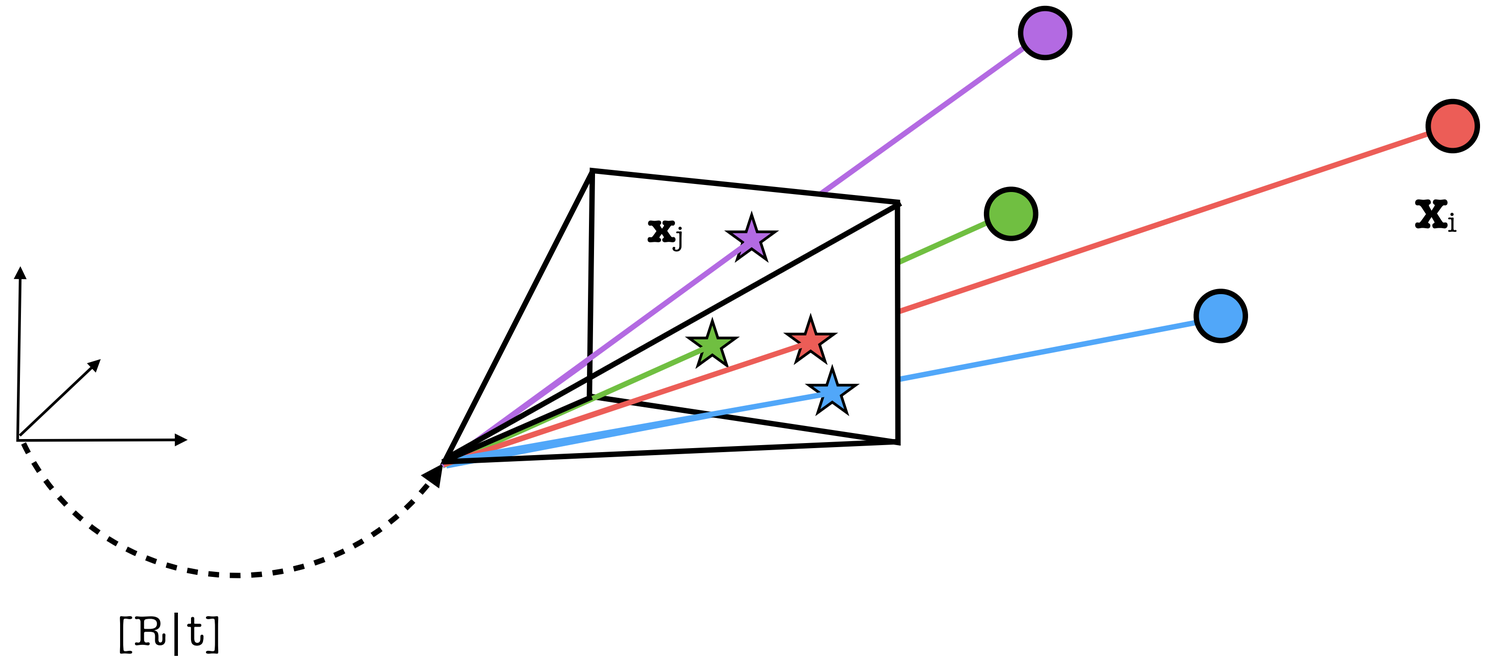
# Nearest Neighbor Search

- 2D-3D matching via nearest neighbor search in descriptor space
- Approximate search due to curse of dimensionality
- Popular approach: Inverted files




- Linear time complexity, but small constant and cache efficiency
- Very good software libraries:
  - FLANN [Muja, Lowe, Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, VISAPP 2009] [code]
  - FAISS [Johnson, Douze, Jégou, Billion-scale similarity search with GPUs. arXiv:1702.08734] [code]

# The n-Point Pose (PnP) Problem




- Given: *n* 2D-3D correspondences $(\mathbf{x}_i, \mathbf{X}_i)$
- Compute pose $[\mathtt{R}|\mathtt{t}]$ s.t. $\mathtt{K}[\mathtt{R}|\mathtt{t}]\mathbf{X}_i = \alpha_i\mathbf{x}_i$, $\alpha_i>0$
- Optionally: Also estimate internal calibration matrix $\mathtt{K}$, e.g., [Larsson, Kukelova, Zheng, Making minimal solvers for absolute pose estimation compact and robust, ICCV 2017][Bujnak, Kukelova, Pajdla, A general solution to the P4P problem for camera with unknown focal length, CVPR 2008]

# Robust Estimation via RANSAC

**While** probability of missing correct model >η

Estimate model from *n* random data points

Estimate support (**#inliers** / **robust cost func.**) of model

If new best model

**Perform Local Optimization (LO)**

update best model, η

**Return**: Model with most inliers / lowest cost

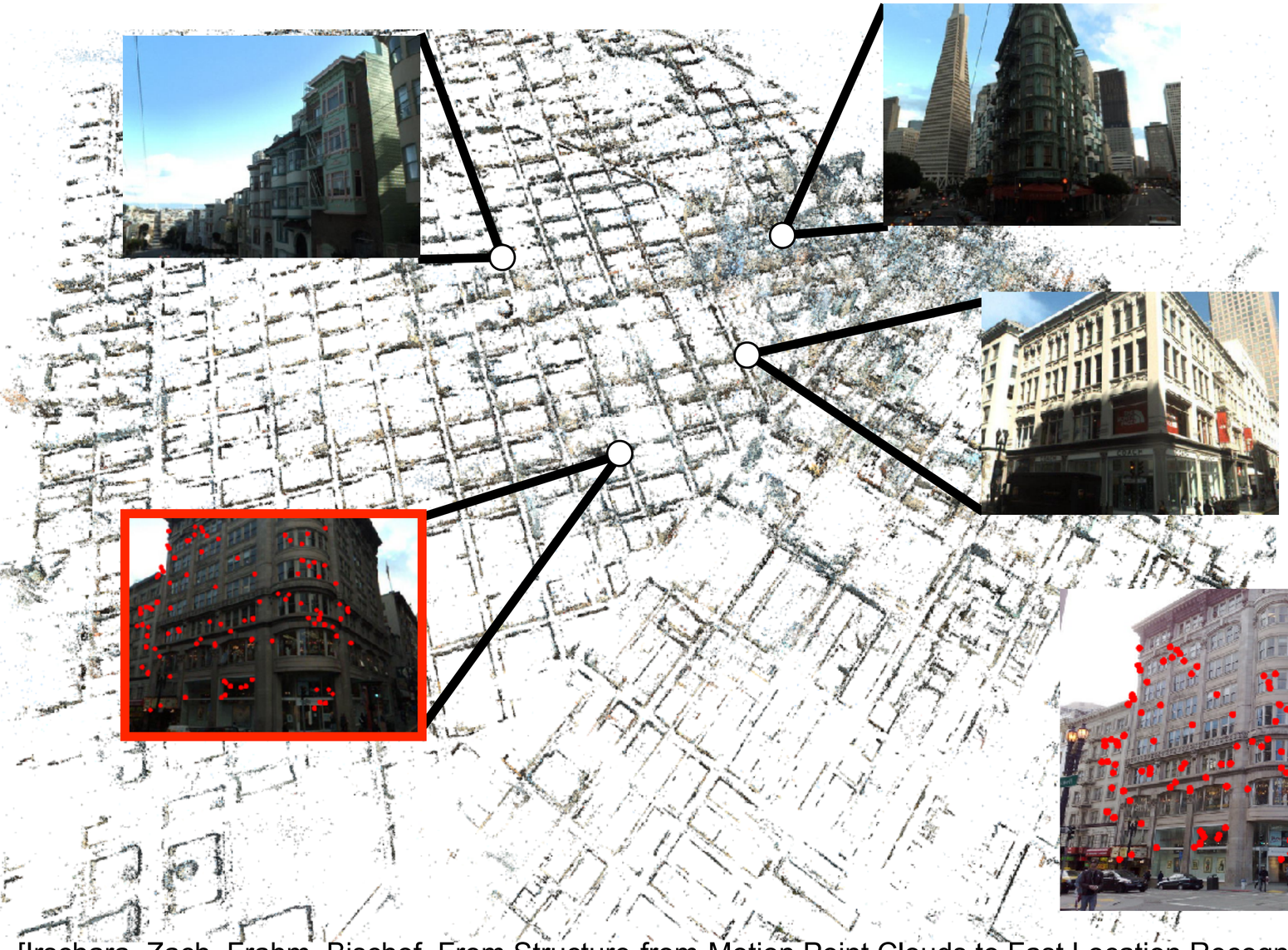
[Chum, Matas, Optimal Randomized RANSAC. PAMI 2008]

[Lebeda, Matas, Chum, Fixing the Locally Optimized RANSAC. BMVC 2012] [code]

- See also USAC [Raguram et al., PAMI'13] [code] (good overview, nice implementation)
- Never use standard RANSAC!

[Fischler & Bolles, Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. CACM 1981]

# Large-Scale Localization via Image Retrieval





Perform **image retrieval**

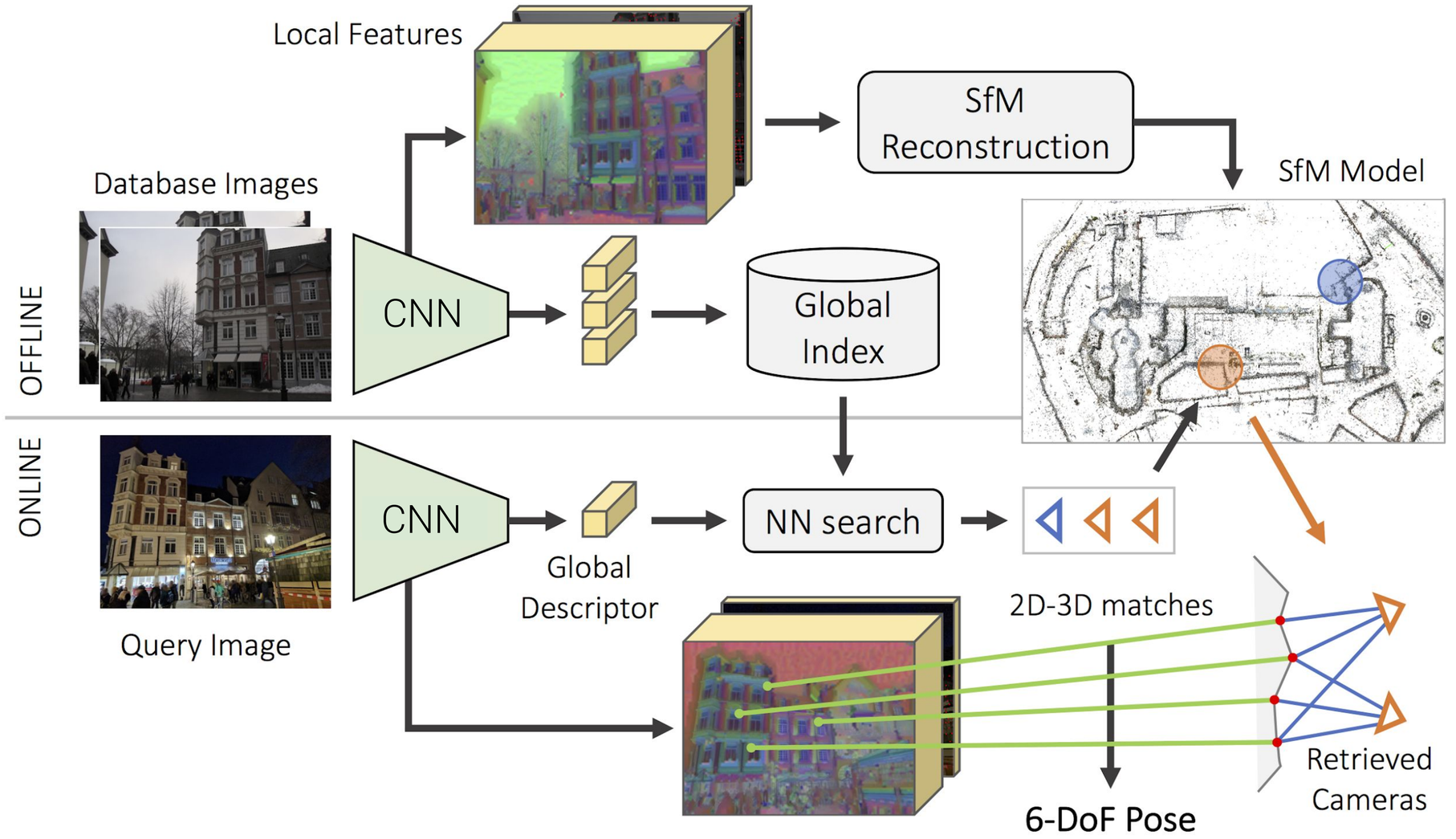
Establish **2D-2D matches**

Establish **2D-3D matches**

Robust **pose estimation**

[Irschara, Zach, Frahm, Bischof, From Structure-from-Motion Point Clouds to Fast Location Recognition, CVPR 2009]

# Large-Scale Localization via Image Retrieval




[Sarlin, Cadena, Siegwart, Dymczyk, From Coarse to Fine: Robust Hierarchical Localization at Large Scale, CVPR 2019] slide credit: Paul-Edouard Sarlin

# Are Large-Scale 3D Models Necessary?

**Divide scene** into 150m x 150m tiles

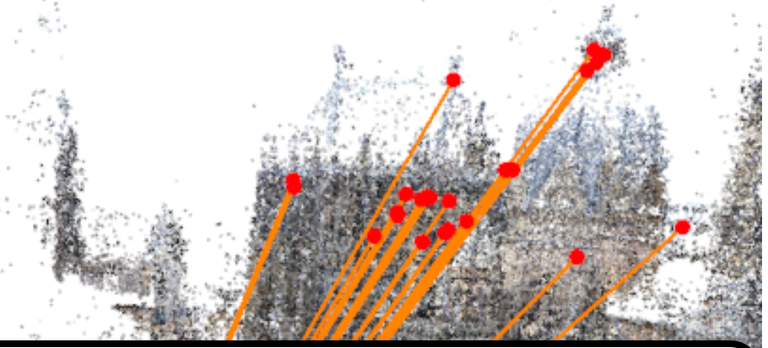
**Compress structure & appearance** per tile

**Pose (GPS) prior** to determine relevant tiles

Matching & pose estimation per tile

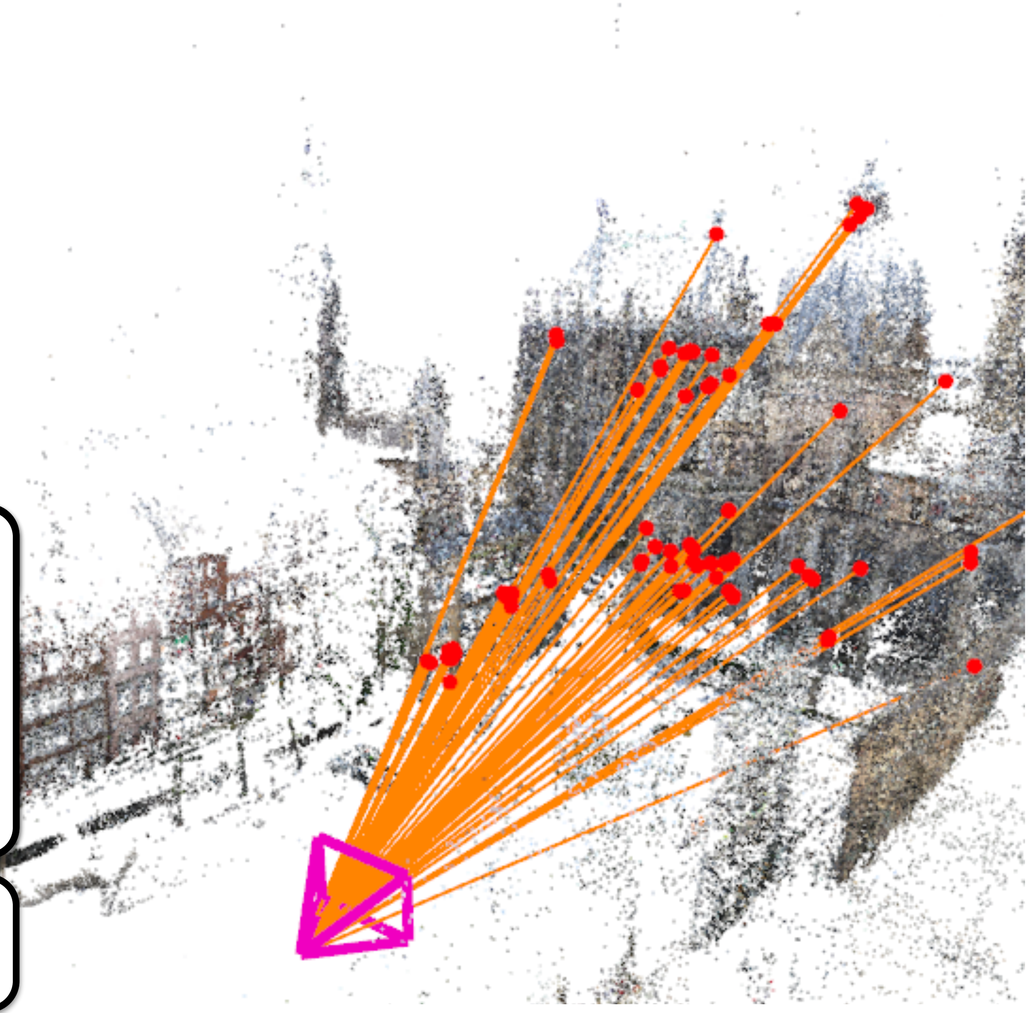
[Lynen, Zeisl, Aiger, Bosse, Hesch, Pollefeys, Siegwart, Sattler, Large-scale, real-time visual-inertial localization revisited, IJRR 2020]

# “New School” Localization



| | |
|---|---|
| Feature Detection | 1999 (SIFT) |
| Feature Description | |
| Descriptor Matching for 2D-3D Matching | 1975 (kd-trees) |
| Estimate Camera Pose (RANSAC + n-point-pose solver) | ≤1773 (Lagrange), 1841 (P3P, Grunert), 1981 (RANSAC) |

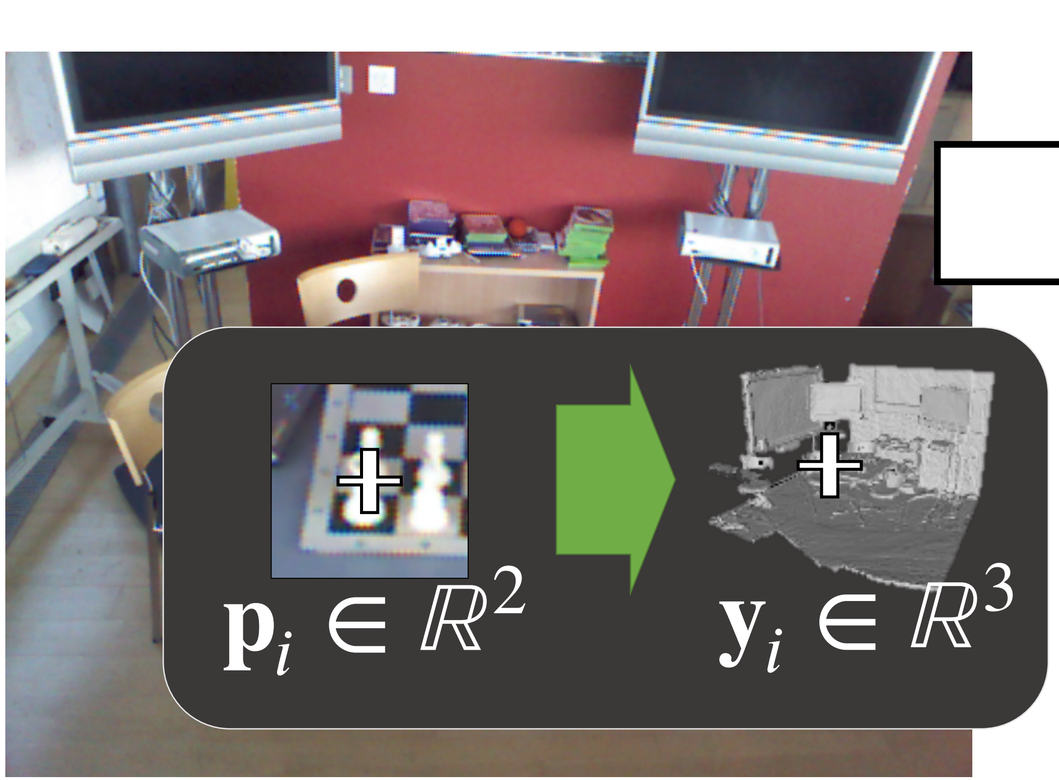
# “New School” Localization



Predict 2D-3D Matches from Patches
(CNN, Random Forest, …)

Estimate Camera Pose (RANSAC + n-point-pose solver)

# Scene Coordinate Regression

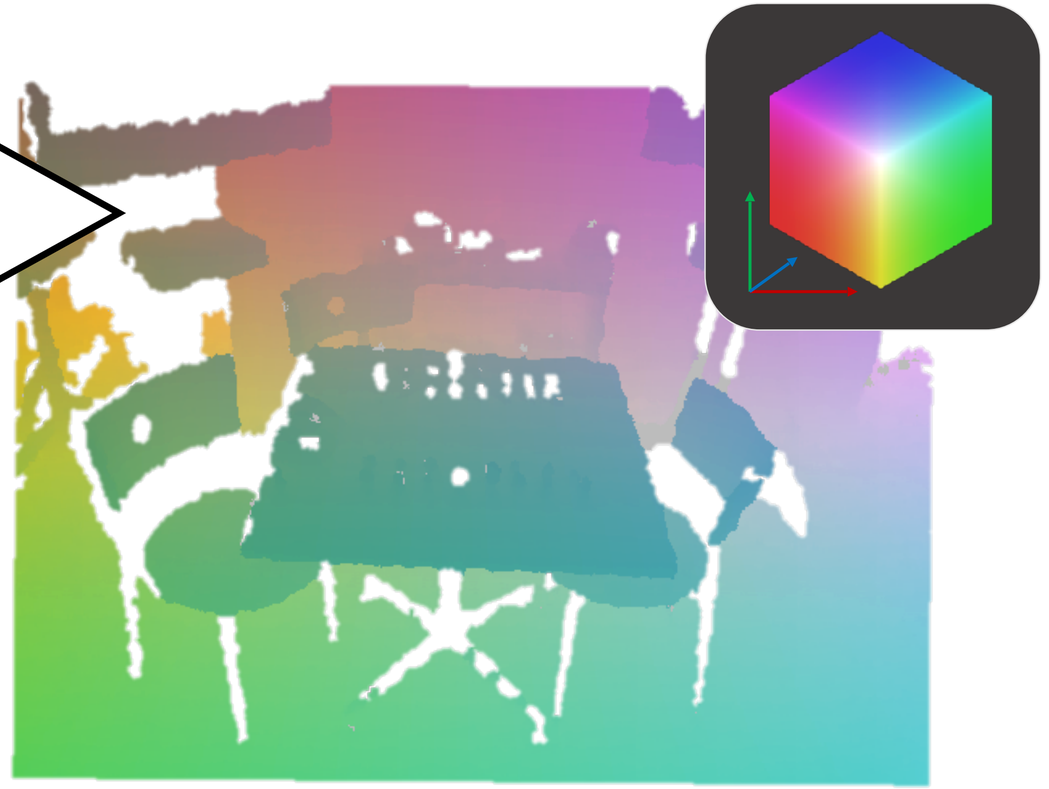
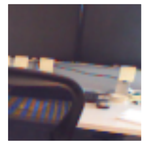



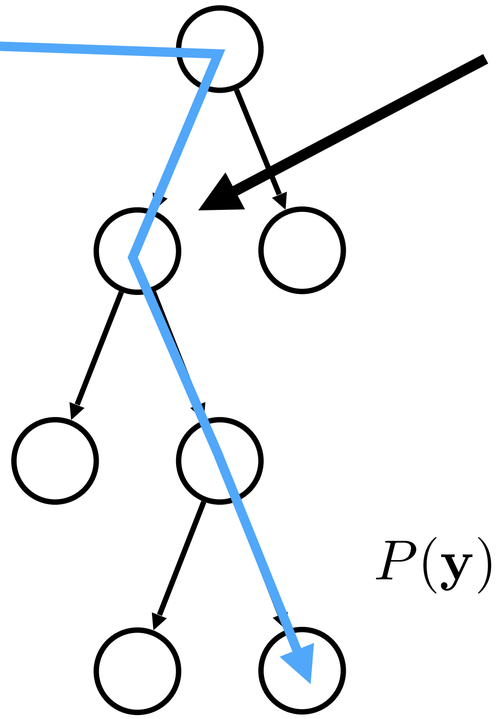
**Image** **3D points in global coordinates**

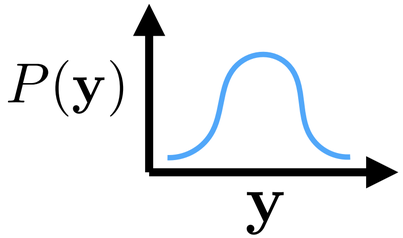
[Shotton et al., Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images, CVPR 2013] slide credit: Eric Brachmann

# Scene Coordinate Regression

Patch
(RGB-D)






Random Forest

Optimize information gain at each split node:

$$\max_{\theta} E(\mathcal{S}_n) - \sum_{i \in \{L,R\}} \frac{|\mathcal{S}_n^i(\theta)|}{|\mathcal{S}_n|} E(\mathcal{S}_n^i(\theta))$$

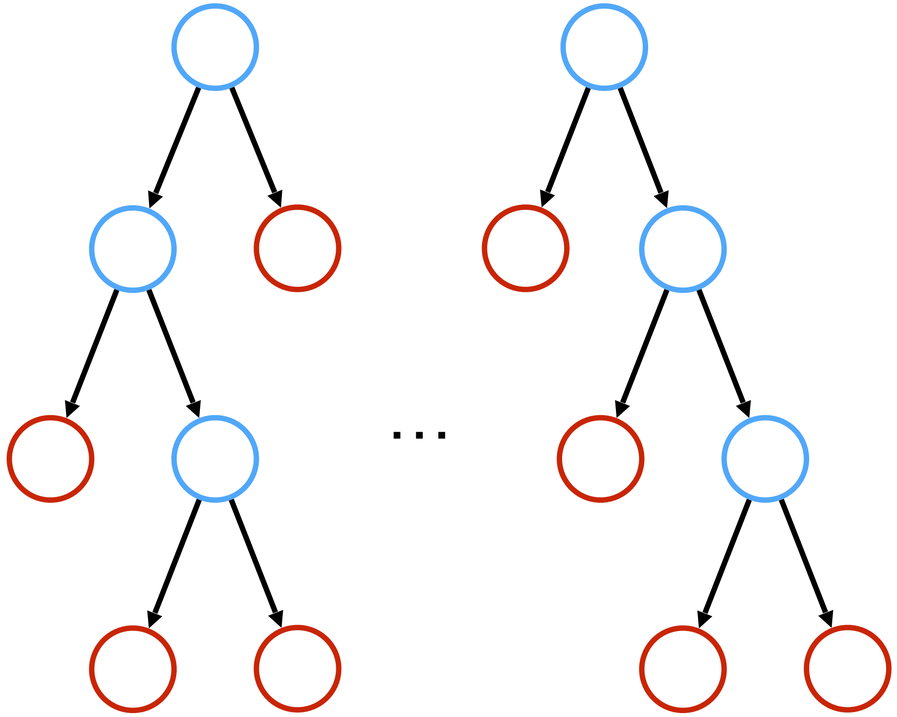
Uncertainty of predicted coordinates, e.g.,

$$E(\mathcal{S}_n) = \frac{1}{|\mathcal{S}_n|} \sum_{\mathbf{y} \in \mathcal{S}_n} ||\mathbf{y} - \bar{\mathbf{y}}||$$

[Shotton et al., Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images, CVPR 2013]
[Valentin et al., Exploiting Uncertainty in Regression Forests for Accurate Camera Relocalization, CVPR 2015]

slide credit: Eric Brachmann
Jaime Shotton

# Online Adaption to New Scenes

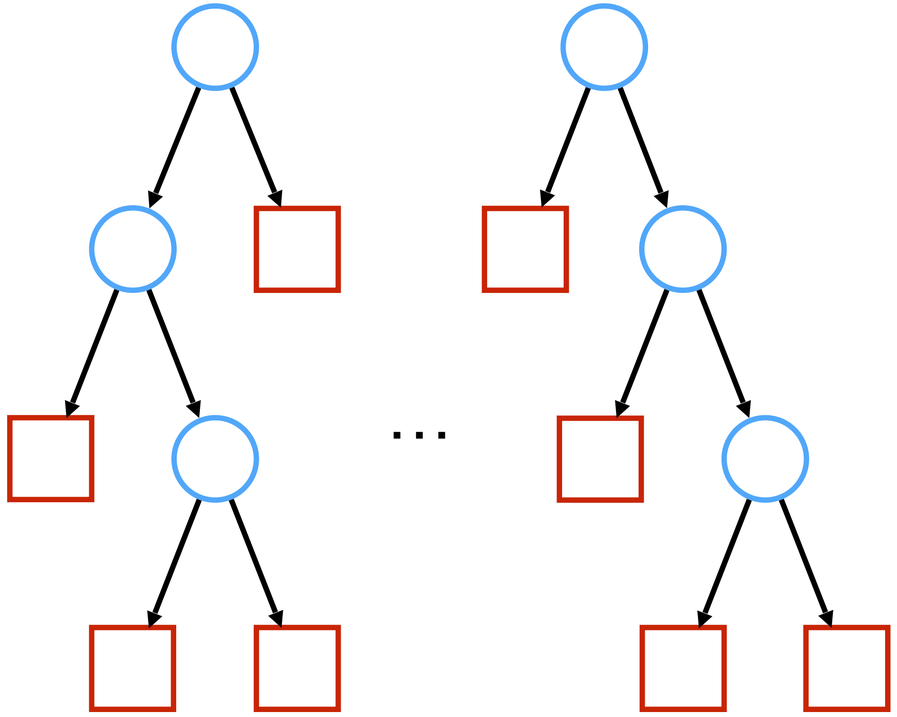




Random Forest

- Random forest trained on other scene
- **Key observation**: Split nodes generalize rather well
- Only retrain **predictions in leave nodes**

[Cavallari, Golodetz, Lord, Valentin, Di Stefano, Torr, On-The-Fly Adaptation of Regression Forests for Online Camera Relocalisation, CVPR 2017]

# Online Adaption to New Scenes

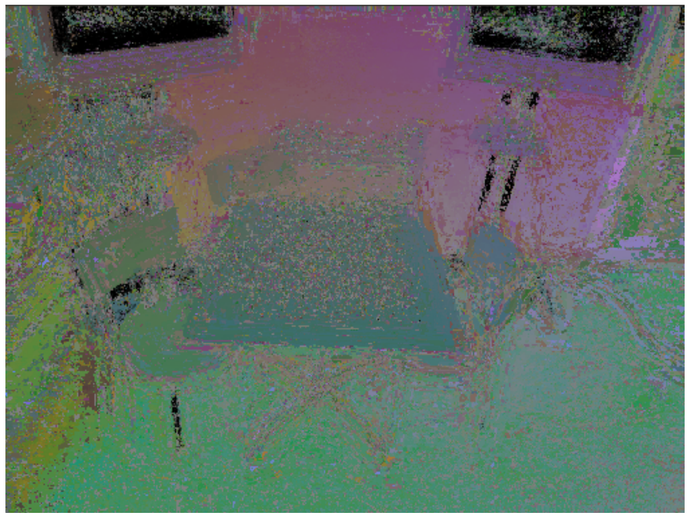

Random Forest

- Random forest trained on other scene
- **Key observation**: Split nodes generalize rather well
- Only retrain **predictions in leave nodes**
- Can be done **efficiently** (even **online**)
- Analogy for local features: Re-use search structure [Sattler, Leibe, Kobbelt, Fast Image-Based Localization using Direct 2D-to-3D Matching. ICCV 2011]

[Cavallari, Golodetz, Lord, Valentin, Di Stefano, Torr, On-The-Fly Adaptation of Regression Forests for Online Camera Relocalisation, CVPR 2017]

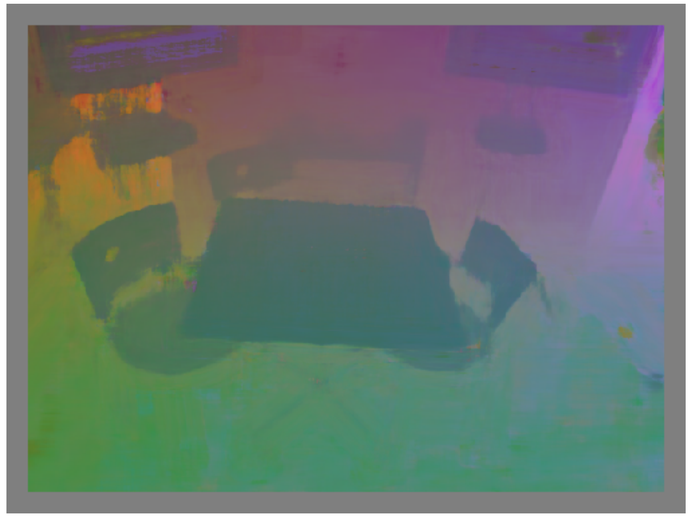
# CNNs vs. Regression Forests


Forest Prediction:



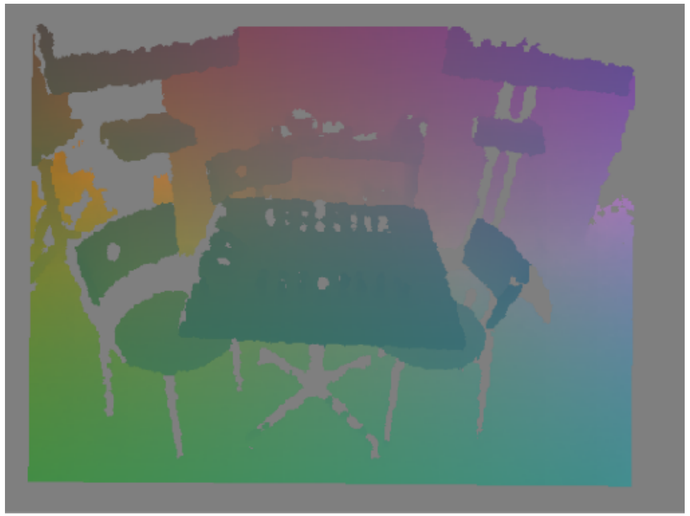
Pose Estimation Succeeds
(< 5cm, 5°)

CNN Prediction:



Pose Estimation Fails
(> 5cm, 5°)

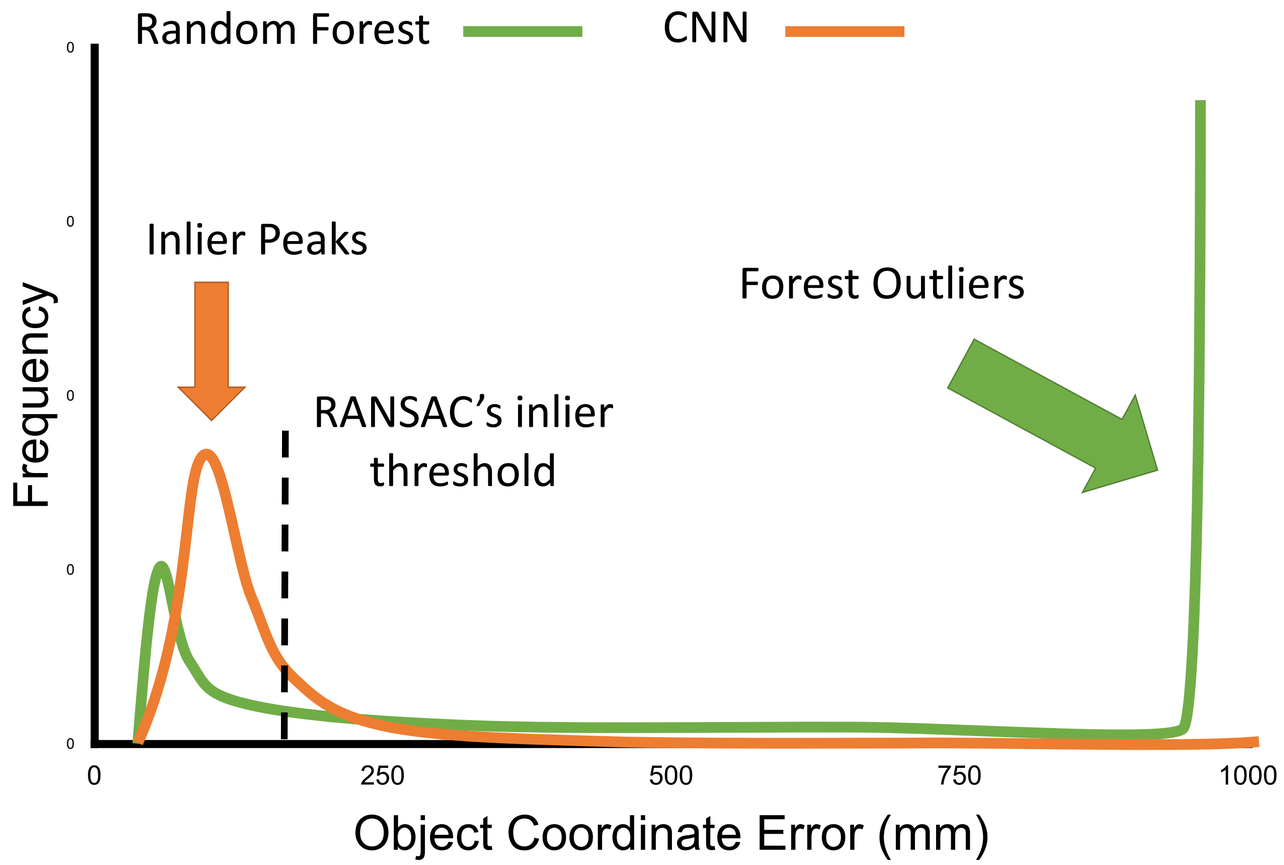
Ground Truth:



slide credit: Eric Brachmann

[Massiceti, Krull, Brachmann, Rother, Torr, Random Forests versus Neural Networks − What's Best for Camera Localization?, ICRA 2017]

# CNNs vs. Regression Forests

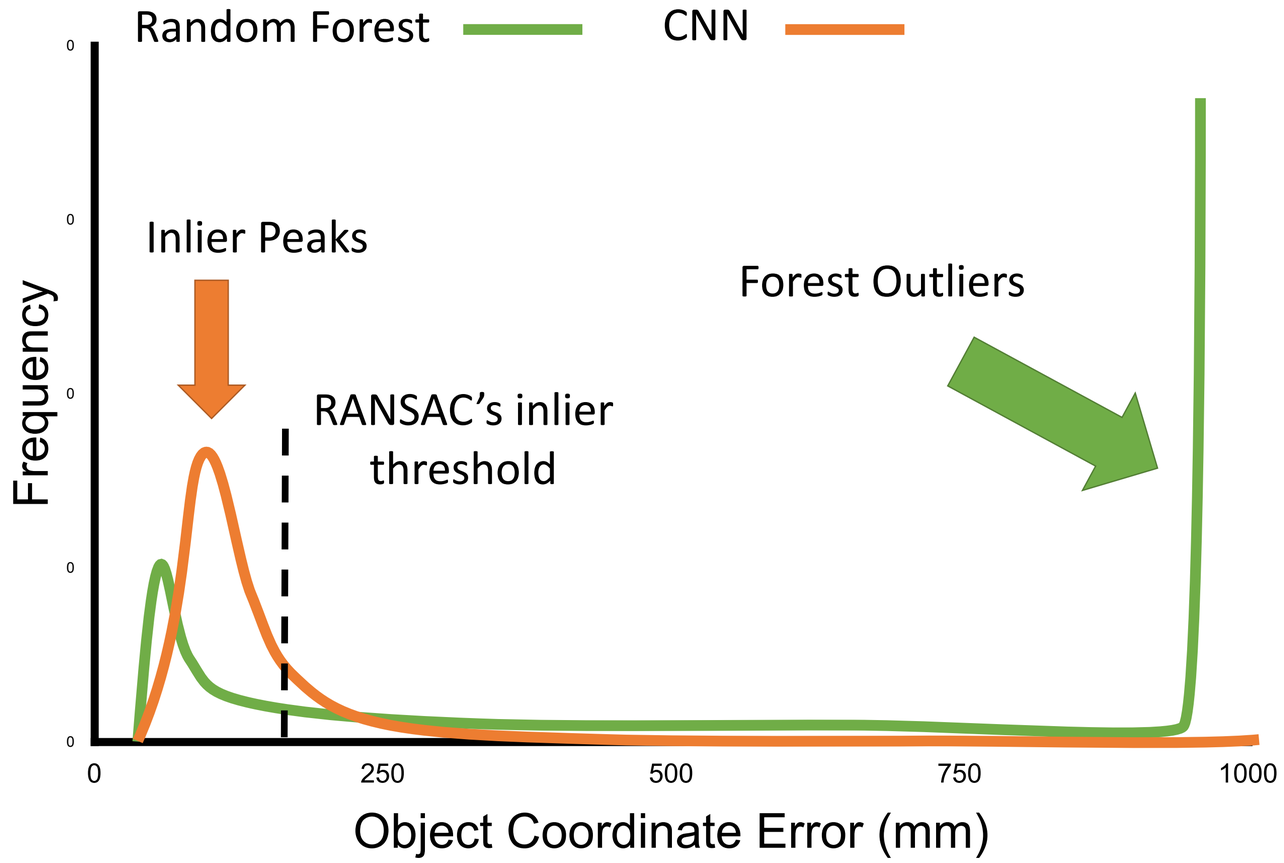




slide credit: Eric Brachmann

# CNNs vs. Regression Forests

Image → **Scene Coordinate Regression (CNN)** → RANSAC → 6DOF Pose

What we optimize: $||\mathbf{y} - \hat{\mathbf{y}}||_1$

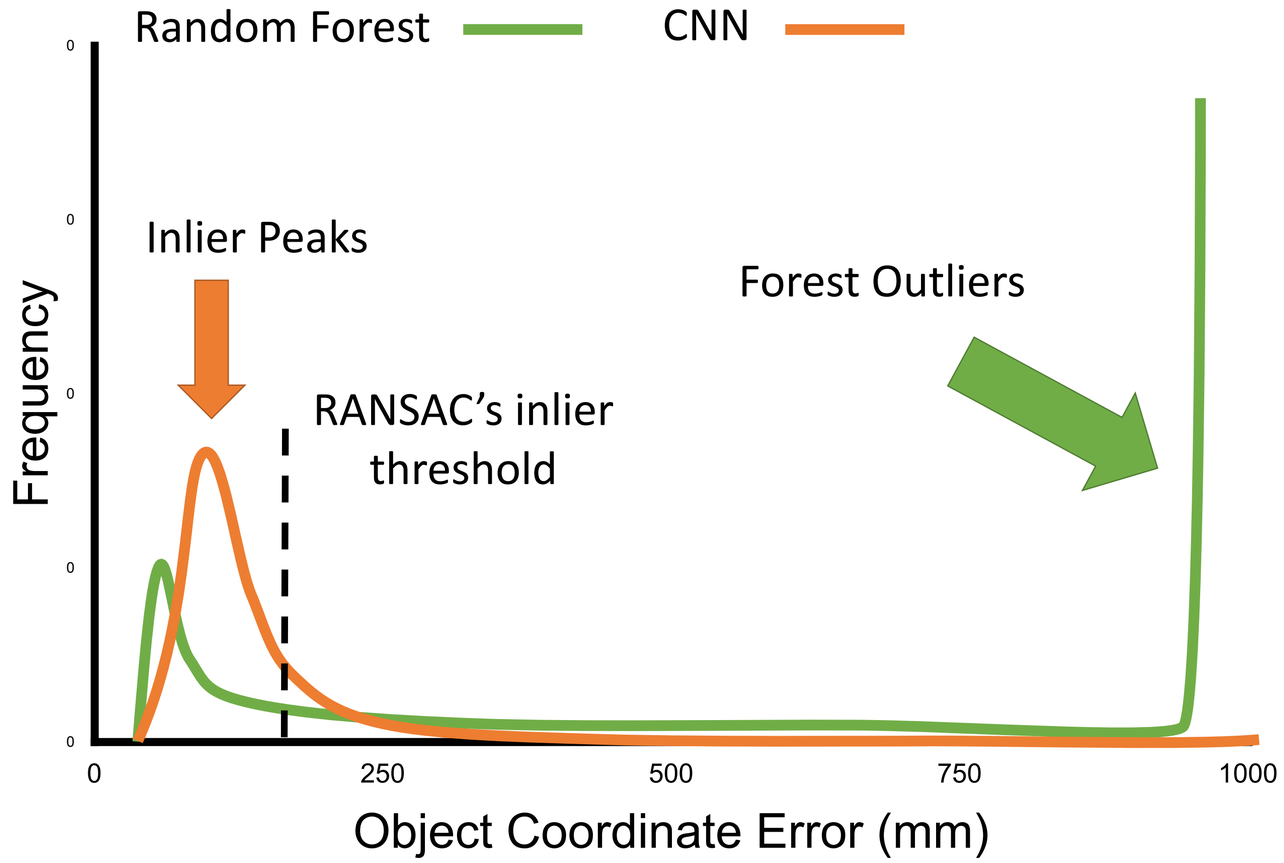



slide credit: Eric Brachmann

# CNNs vs. Regression Forests

Image → **Scene Coordinate Regression (CNN)** → RANSAC → 6DOF Pose

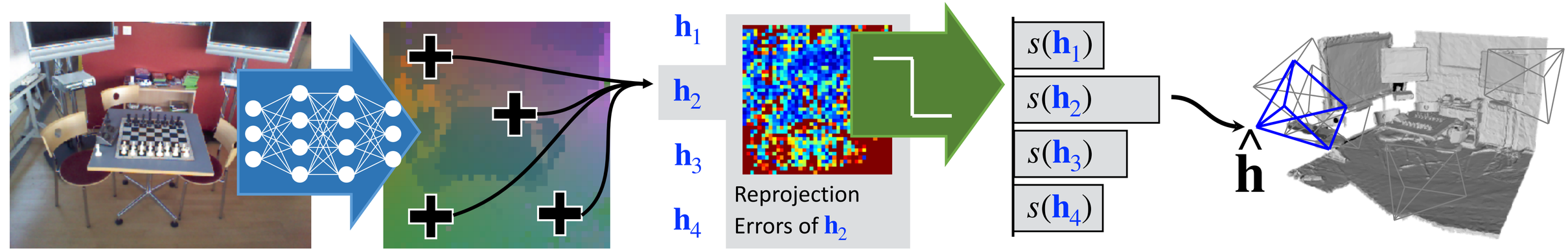
What we optimize: $||\mathbf{y} - \hat{\mathbf{y}}||_1$




What we should optimize!

slide credit: Eric Brachmann

# Differentiable RANSAC (DSAC)

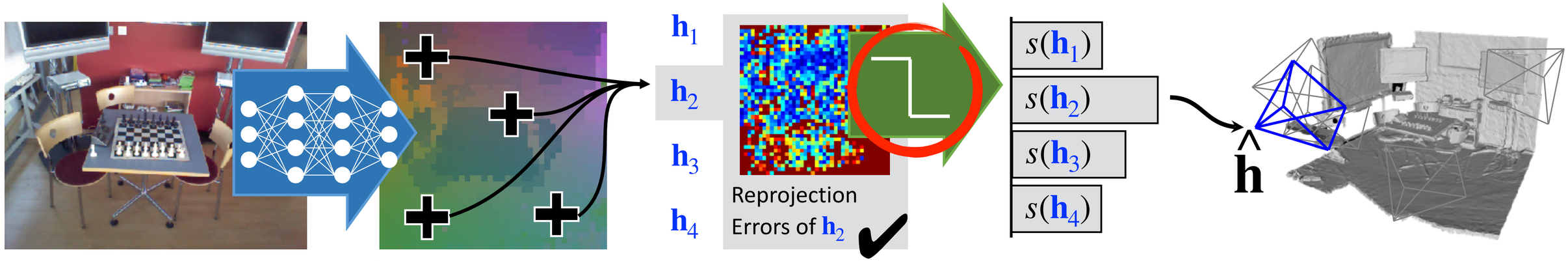



Input RGB | Scene Coordinate Regression ($\mathbf{w}$) | Hypothesis Sampling | Scoring | Hypothesis Selection | Result

slide credit: Eric Brachmann

[Brachmann, Krull Nowozin, Shotton, Michel, Gumhold, Rother, DSAC - Differentiable RANSAC for Camera Localization, CVPR 2017]
[Brachmann, Rother, Learning Less is More - 6D Camera Localization via 3D Surface Regression, CVPR 2018]

# Differentiable RANSAC (DSAC)



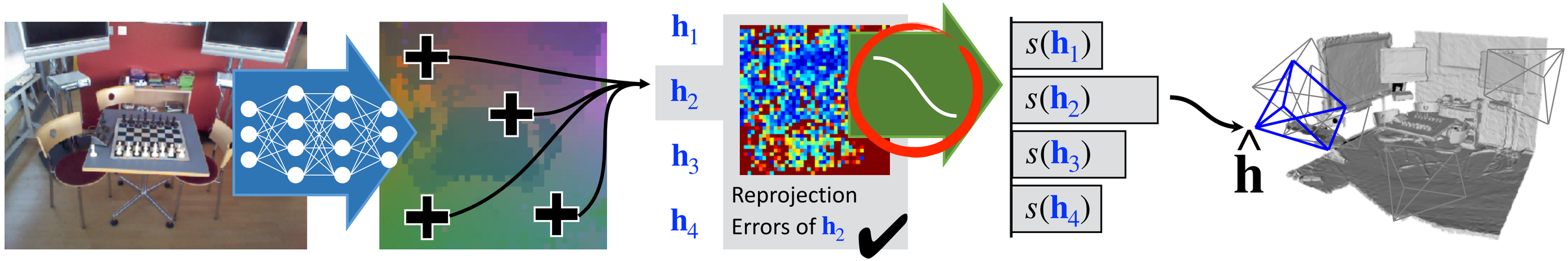

Input RGB | Scene Coordinate Regression ($\mathbf{w}$) ✔ | Hypothesis Sampling ✔ | Scoring ✔ | Hypothesis Selection | Result

Compute gradient of loss: $\frac{\partial}{\partial \mathbf{w}} \ell(\hat{\mathbf{h}}, \mathbf{h}^*)$

slide credit: Eric Brachmann


[Brachmann, Krull Nowozin, Shotton, Michel, Gumhold, Rother, DSAC - Differentiable RANSAC for Camera Localization, CVPR 2017]
[Brachmann, Rother, Learning Less is More - 6D Camera Localization via 3D Surface Regression, CVPR 2018]

# Differentiable RANSAC (DSAC)



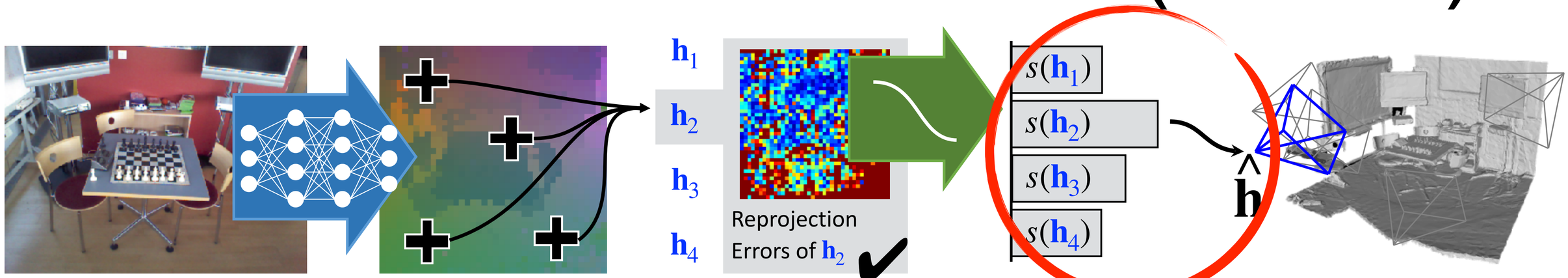

Input RGB | Scene Coordinate Regression ($\mathbf{w}$) ✔ | Hypothesis Sampling ✔ | Soft Scoring | Hypothesis Selection | Result

Compute gradient of loss: $\frac{\partial}{\partial \mathbf{w}} \ell(\hat{\mathbf{h}}, \mathbf{h}^*)$

slide credit: Eric Brachmann

[Brachmann, Krull Nowozin, Shotton, Michel, Gumhold, Rother, DSAC - Differentiable RANSAC for Camera Localization, CVPR 2017]
[Brachmann, Rother, Learning Less is More - 6D Camera Localization via 3D Surface Regression, CVPR 2018]

# Differentiable RANSAC (DSAC)



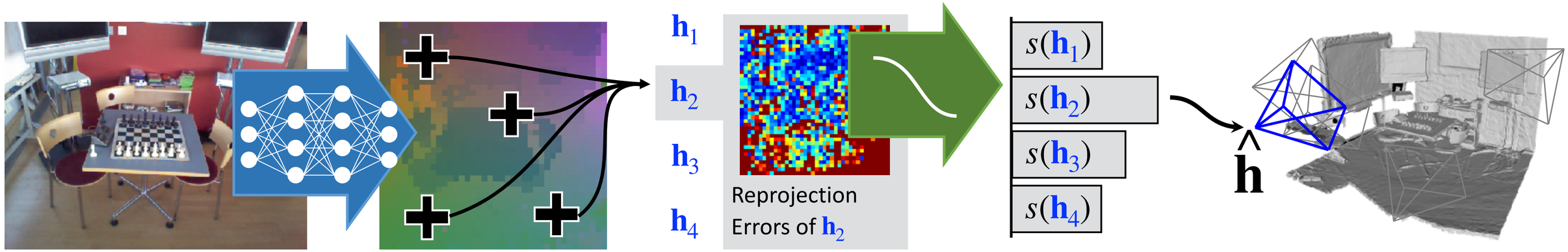

Input RGB | Scene Coordinate Regression ($\mathbf{w}$) ✔ | Hypothesis Sampling ✔ | Soft Scoring ✔ | Hypothesis Selection | Result

Compute gradient of loss: $\frac{\partial}{\partial \mathbf{w}} \ell(\hat{\mathbf{h}}, \mathbf{h}^*)$

slide credit: Eric Brachmann


[Brachmann, Krull Nowozin, Shotton, Michel, Gumhold, Rother, DSAC - Differentiable RANSAC for Camera Localization, CVPR 2017]
[Brachmann, Rother, Learning Less is More - 6D Camera Localization via 3D Surface Regression, CVPR 2018]

# Differentiable RANSAC (DSAC)




Input RGB | Scene Coordinate Regression (**w**) | Hypothesis Sampling | Soft Scoring | Hypothesis Selection | Result

argmax Selection

$$\mathbf{h}_{\text{AM}} = \underset{\mathbf{h}_j}{\text{argmax}} s(\mathbf{h}_j)$$

hard decision

non-differentiable

Soft argmax Selection

$$\mathbf{h}_{\text{SoftAM}} = \sum_j \frac{\exp(s(\mathbf{h}_j))\mathbf{h}_j}{\sum_k \exp(s(\mathbf{h}_k))}$$

soft decision

differentiable

**Probabilistic Selection**

$$\mathbf{h}_{\text{DSAC}} = \mathbf{h}_j, \text{ where } j \sim \frac{\exp(s(\mathbf{h}_j))}{\sum_k \exp(s(\mathbf{h}_k))}$$

hard decision

differentiable

slide credit: Eric Brachmann

[Brachmann, Krull Nowozin, Shotton, Michel, Gumhold, Rother, DSAC - Differentiable RANSAC for Camera Localization, CVPR 2017]
[Brachmann, Rother, Learning Less is More - 6D Camera Localization via 3D Surface Regression, CVPR 2018]

# DSAC - Loss

- Probabilistic selection criterion (hard decision):

$$\mathbf{h}_{\text{DSAC}} = \mathbf{h}_j, \text{ where } j \sim \frac{\exp(s(\mathbf{h}_j))}{\sum_k \exp(s(\mathbf{h}_k))} = P(j \mid \mathbf{w})$$

- Minimize **expected** loss:

$$\mathbb{E}_{j \sim P(j|\mathbf{w})} \left[ \ell(\mathbf{h}_j, \mathbf{h}^*) \right]$$

Minimize P(j|w) if pose error large

Minimize pose error if P(j|w) large

slide credit: Eric Brachmann

[Brachmann, Krull Nowozin, Shotton, Michel, Gumhold, Rother, DSAC - Differentiable RANSAC for Camera Localization, CVPR 2017]
[Brachmann, Rother, Learning Less is More - 6D Camera Localization via 3D Surface Regression, CVPR 2018]

# Quiz Time

# What Should I Use?

- **Local feature-based methods**
  - Work reliably as long as features work
  - Accuracy strongly depends on number of matches
- **Scene coordinate regression**
  - Shown to be highly accurate
  - Scaling to larger scenes an issue
  - Generalization to new viewing conditions?

# Overview

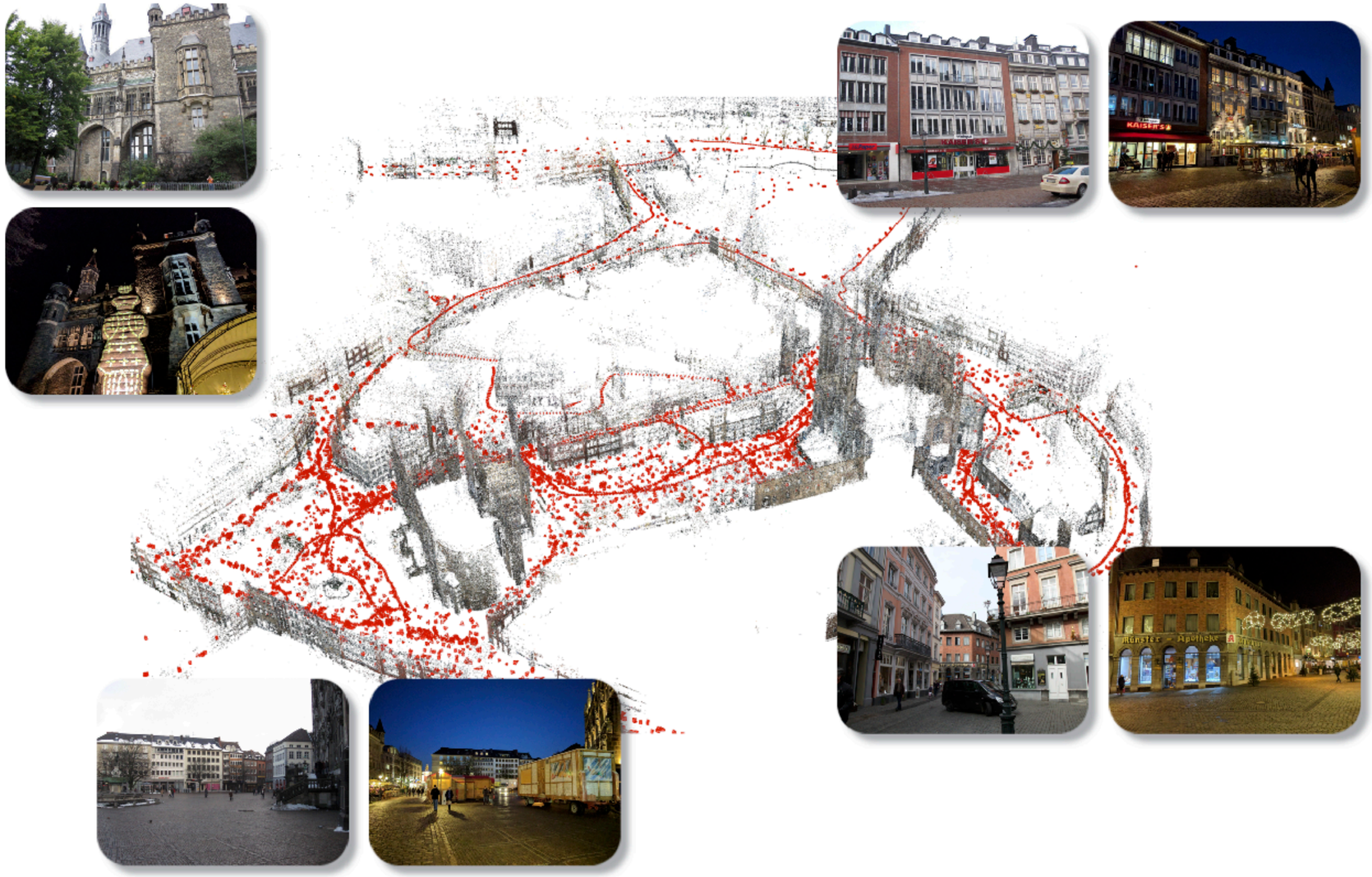
- A (Too) Simple Approach to Visual Localization
- Structure-Based Localization
- **Long-Term Localization**
- Privacy-Preserving Localization

# Long-Term Visual Localization



World is **not static**, **appearance and geometry change**!
How **robust** are visual localization algorithms?

# Aachen Day-Night Dataset



[Sattler, Weyand, Leibe, Kobbelt, Image Retrieval for Image-Based Localization Revisited, BMVC 2012]

[Sattler, Maddern, Toft, Torii, Hammarstrand, Stenborg, Safari, Okutomi, Pollefeys, Sivic, Kahl, Pajdla, Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions, CVPR 2018]

# RobotCar Seasons Dataset

[Maddern, Pascoe, Linegar, Newman, 1 Year, 1000km: The Oxford RobotCar Dataset, IJRR 2016]

[Sattler, Maddern, Toft, Torii, Hammarstrand, Stenborg, Safari, Okutomi, Pollefeys, Sivic, Kahl, Pajdla, Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions, CVPR 2018]

# (Extended) CMU Seasons Dataset

[Badino, Huber, Kanade. Visual topometric localization, IV 2011]

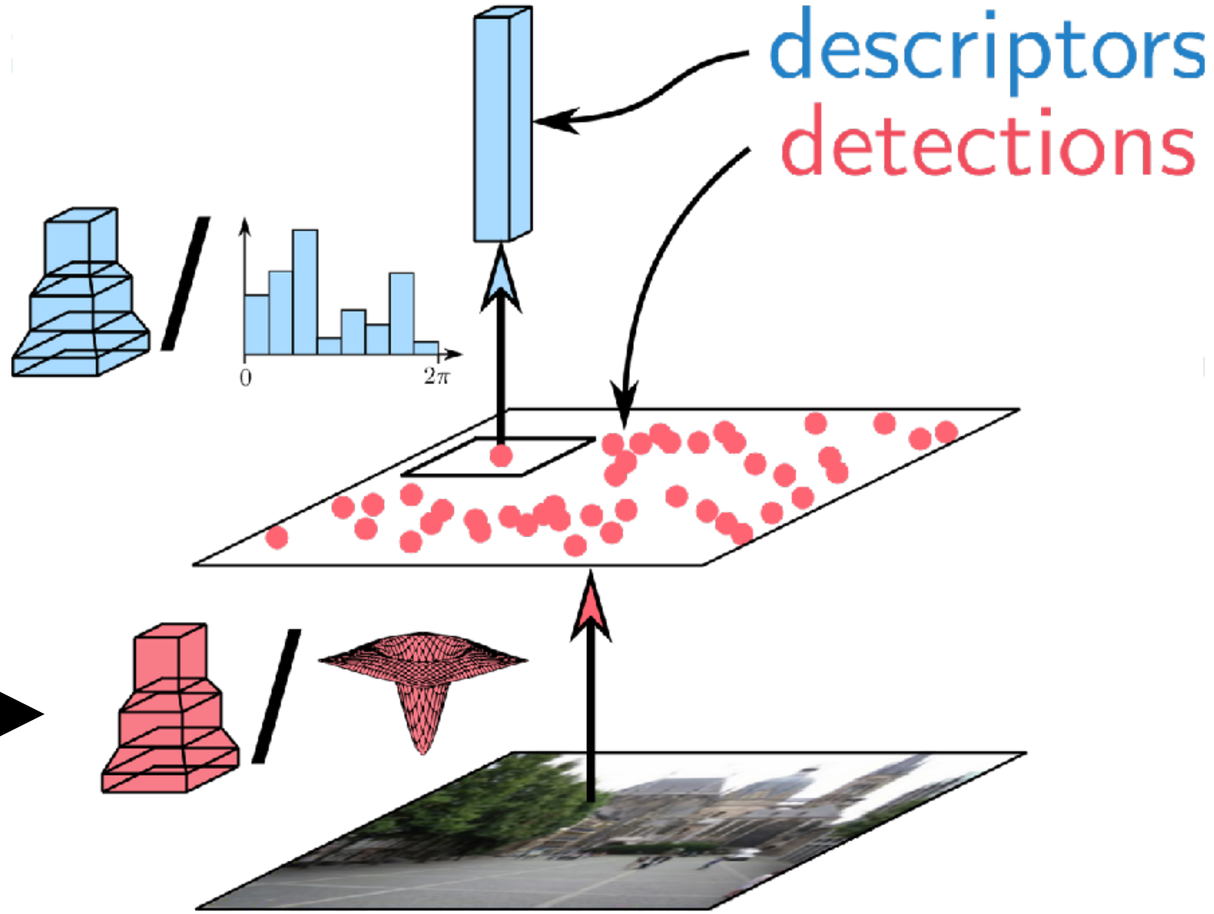
[Sattler, Maddern, Toft, Torii, Hammarstrand, Stenborg, Safari, Okutomi, Pollefeys, Sivic, Kahl, Pajdla, Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions, CVPR 2018]

# Classical Local Features

slide credit: Hugo Germanin

# Classical Local Features

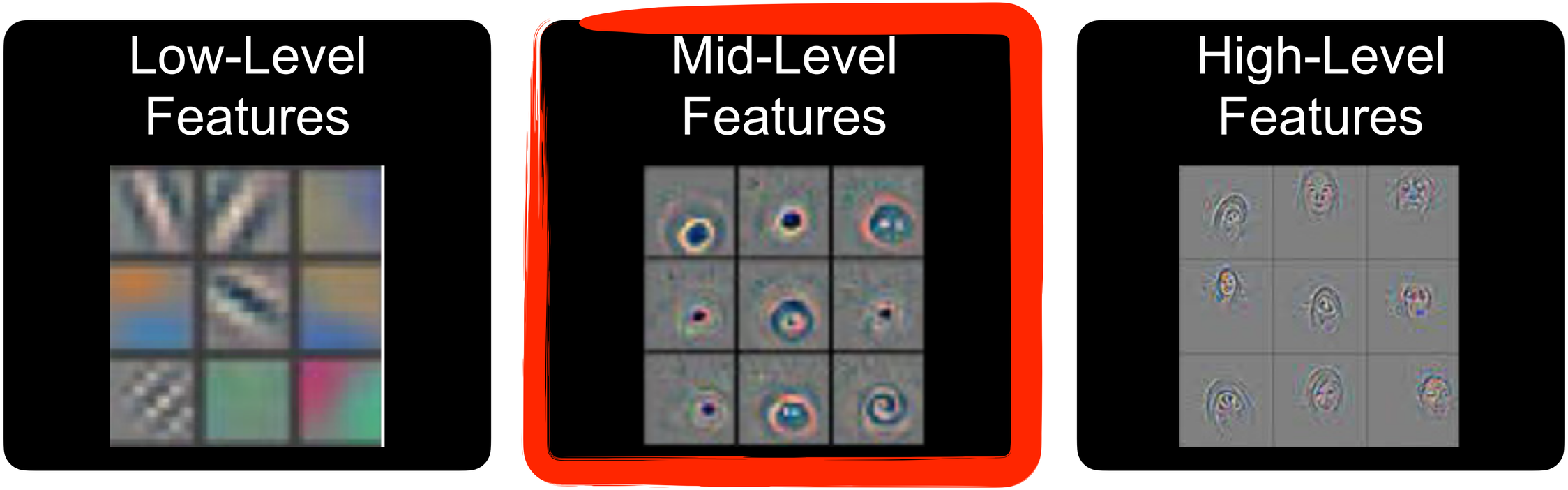
**Detect-then-Describe:**




Efficient, looking at low-level structures / statistics

[Dusmanu, Rocco, Pajdla, Pollefeys, Sivic, Torii, Sattler, D2-Net: A Trainable CNN for Joint Detection and Description of Local Features, CVPR 2019]

# CNNs as Object Detectors

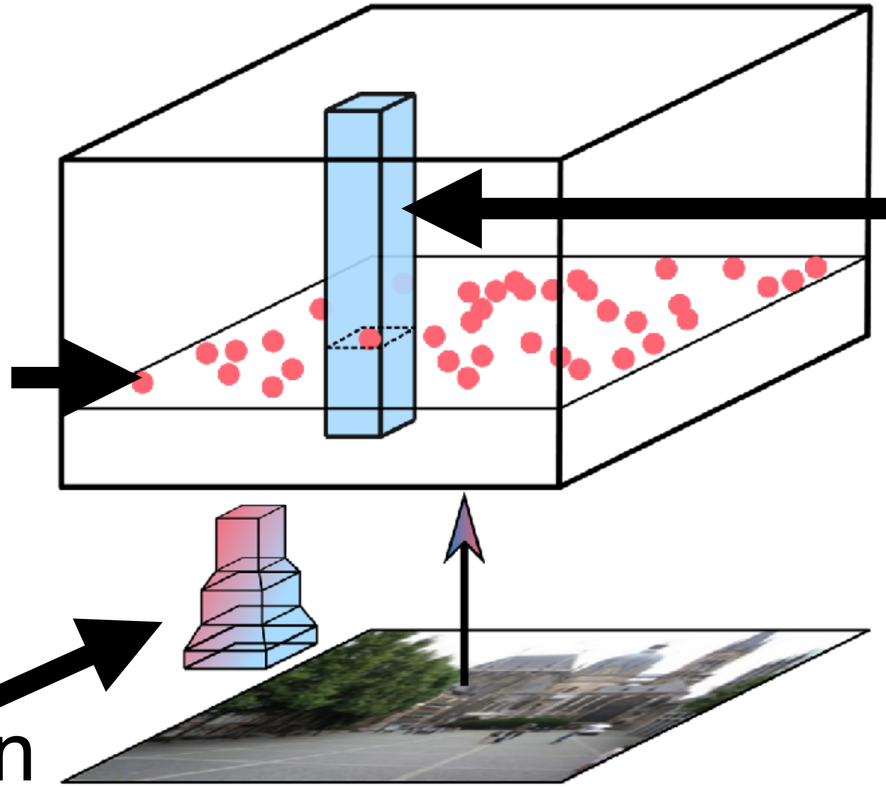



[Zeiler & Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014]

# Detect-And-Describe Approach

Local maxima ≈
Object detections

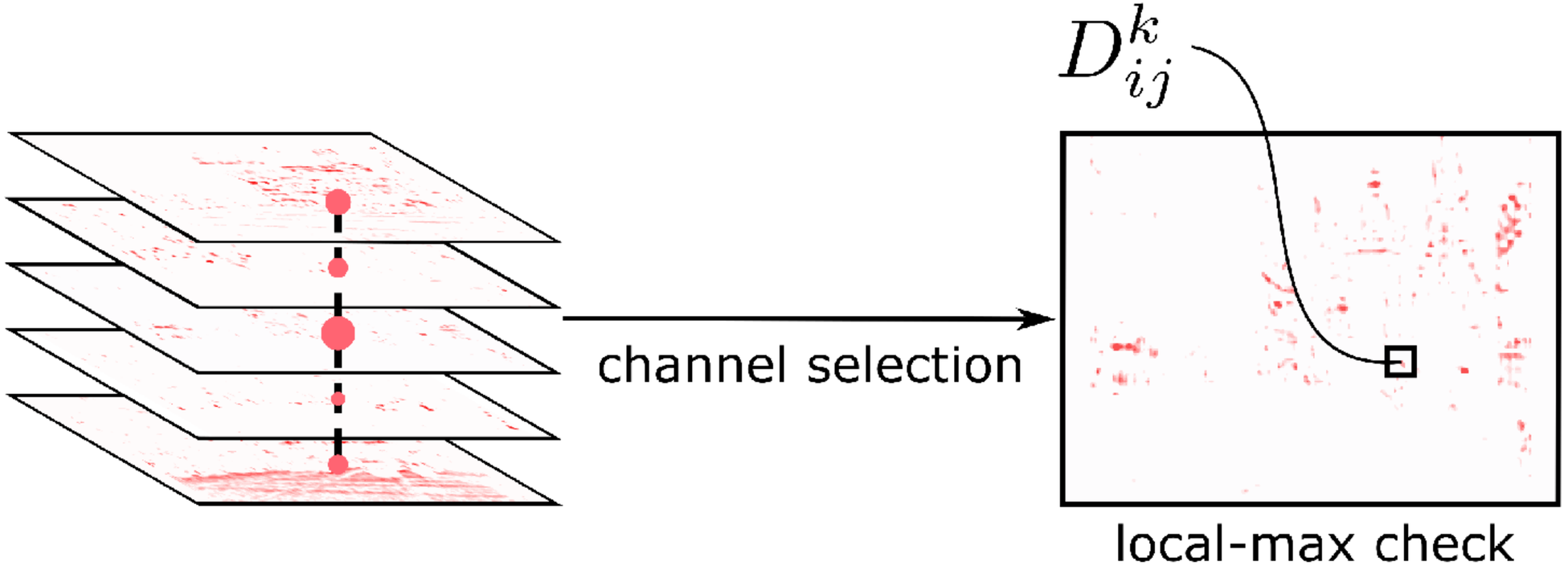
Same CNN for
detection & description



Descriptor ≈
“Objectness” scores

[Dusmanu, Rocco, Pajdla, Pollefeys, Sivic, Torii, Sattler, D2-Net: A Trainable CNN for Joint Detection and Description of Local Features, CVPR 2019]

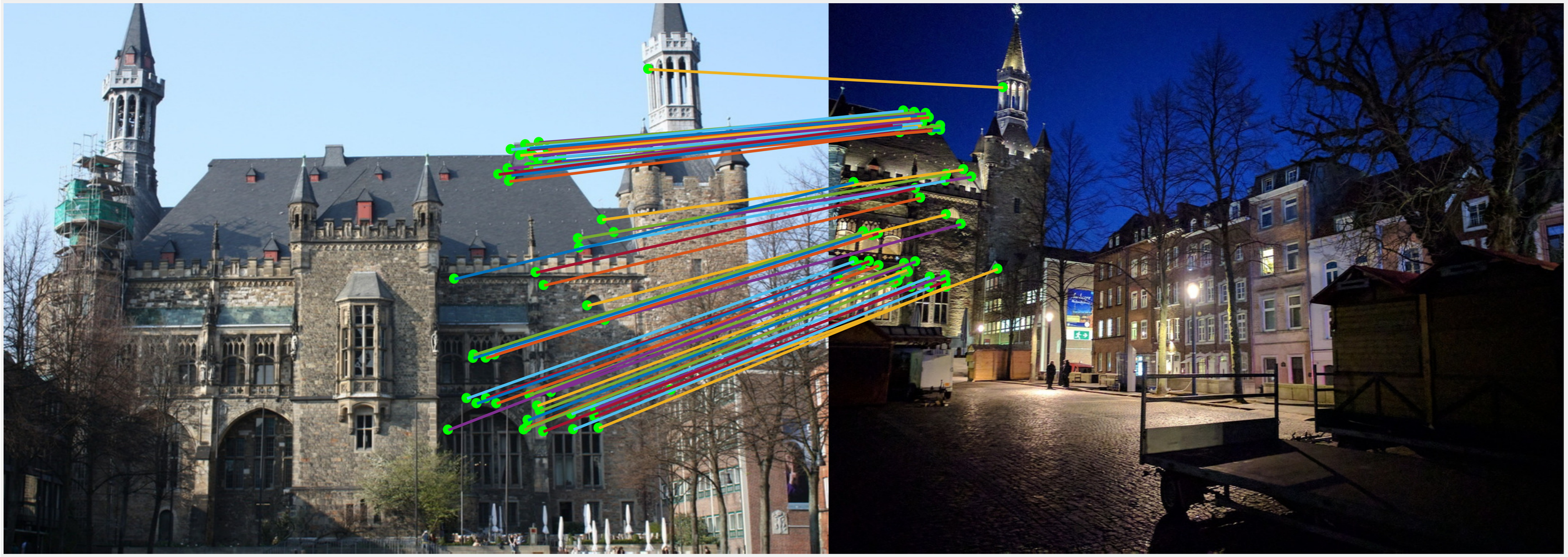
# Hard Keypoint Detection




$$(i,j) \text{ is a detection} \iff D^k_{ij} \text{ is a local max. in } D^k \quad ,$$
$$\text{with } k = \arg\max_t D^t_{ij} \quad .$$

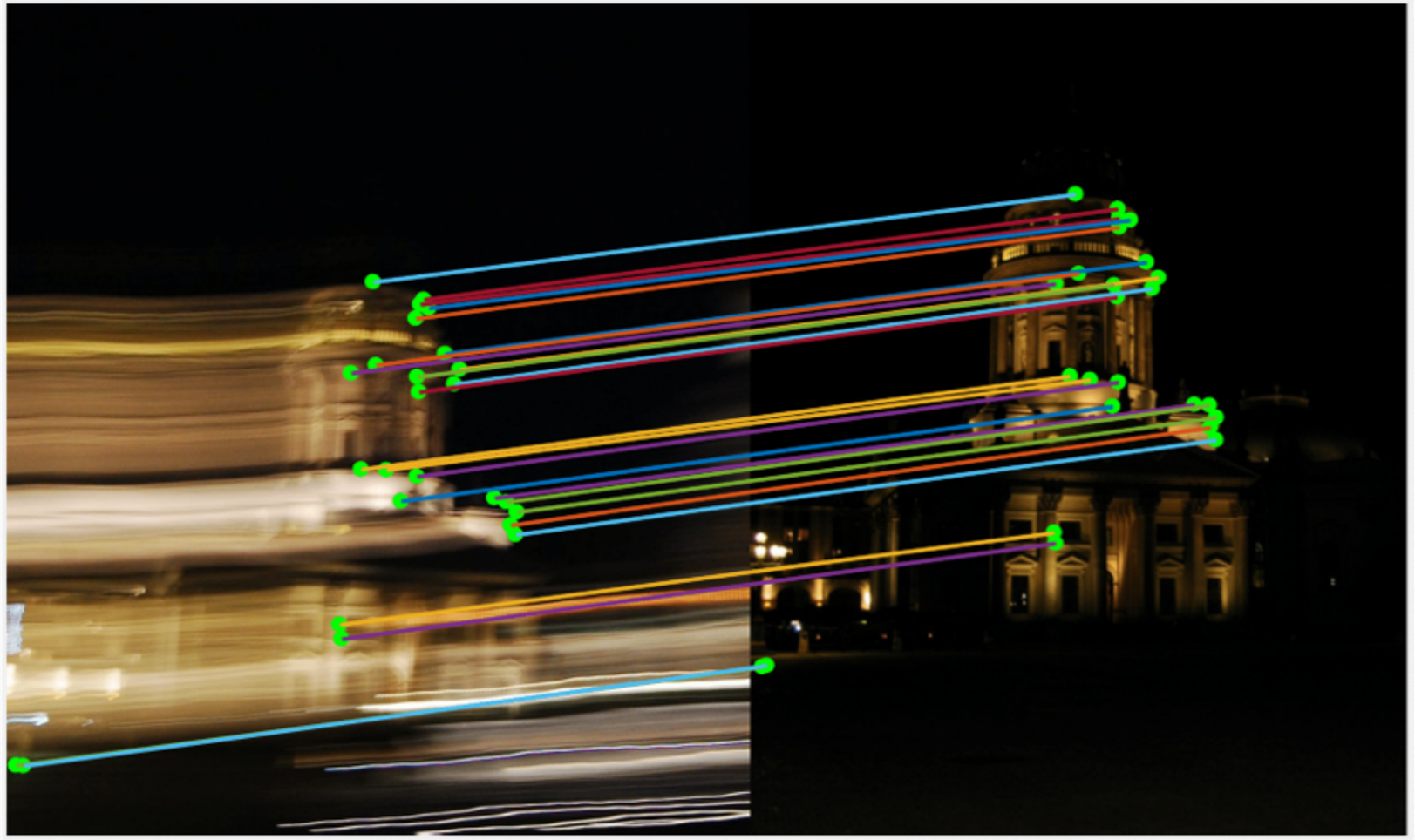
[Dusmanu, Rocco, Pajdla, Pollefeys, Sivic, Torii, Sattler, D2-Net: A Trainable CNN for Joint Detection and Description of Local Features, CVPR 2019]

# The Good



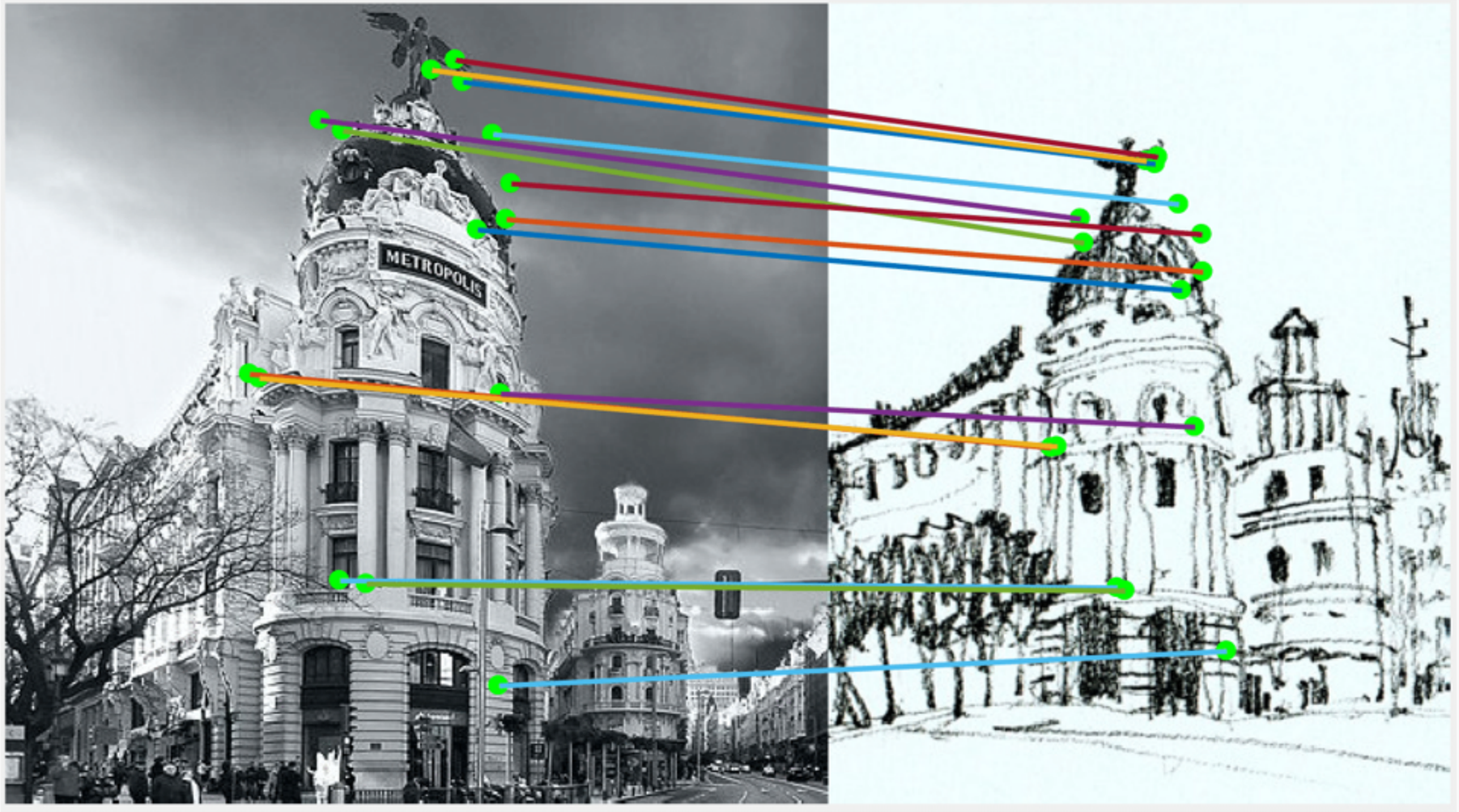
[Dusmanu, Rocco, Pajdla, Pollefeys, Sivic, Torii, Sattler, D2-Net: A Trainable CNN for Joint Detection and Description of Local Features, CVPR 2019]

# The Good



[Dusmanu, Rocco, Pajdla, Pollefeys, Sivic, Torii, Sattler, D2-Net: A Trainable CNN for Joint Detection and Description of Local Features, CVPR 2019]

# The Good

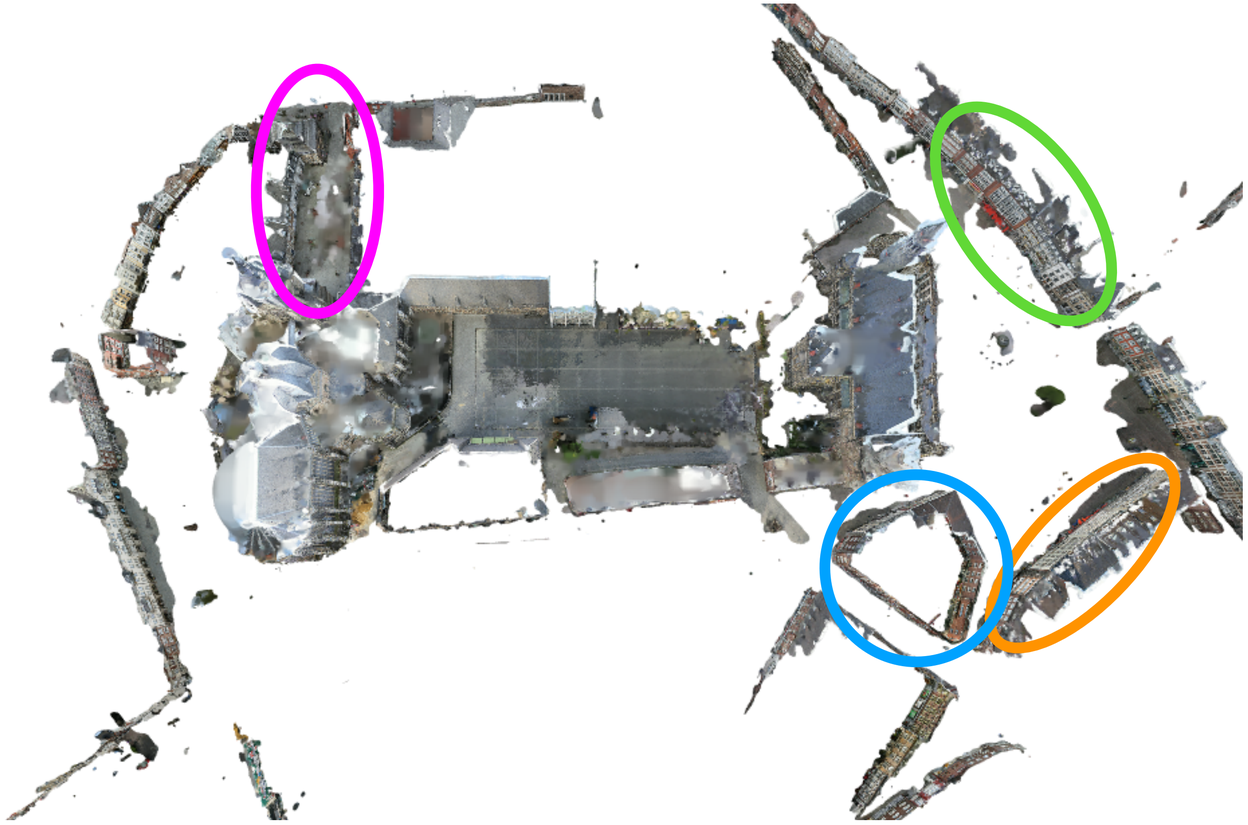



[Dusmanu, Rocco, Pajdla, Pollefeys, Sivic, Torii, Sattler, D2-Net: A Trainable CNN for Joint Detection and Description of Local Features, CVPR 2019]

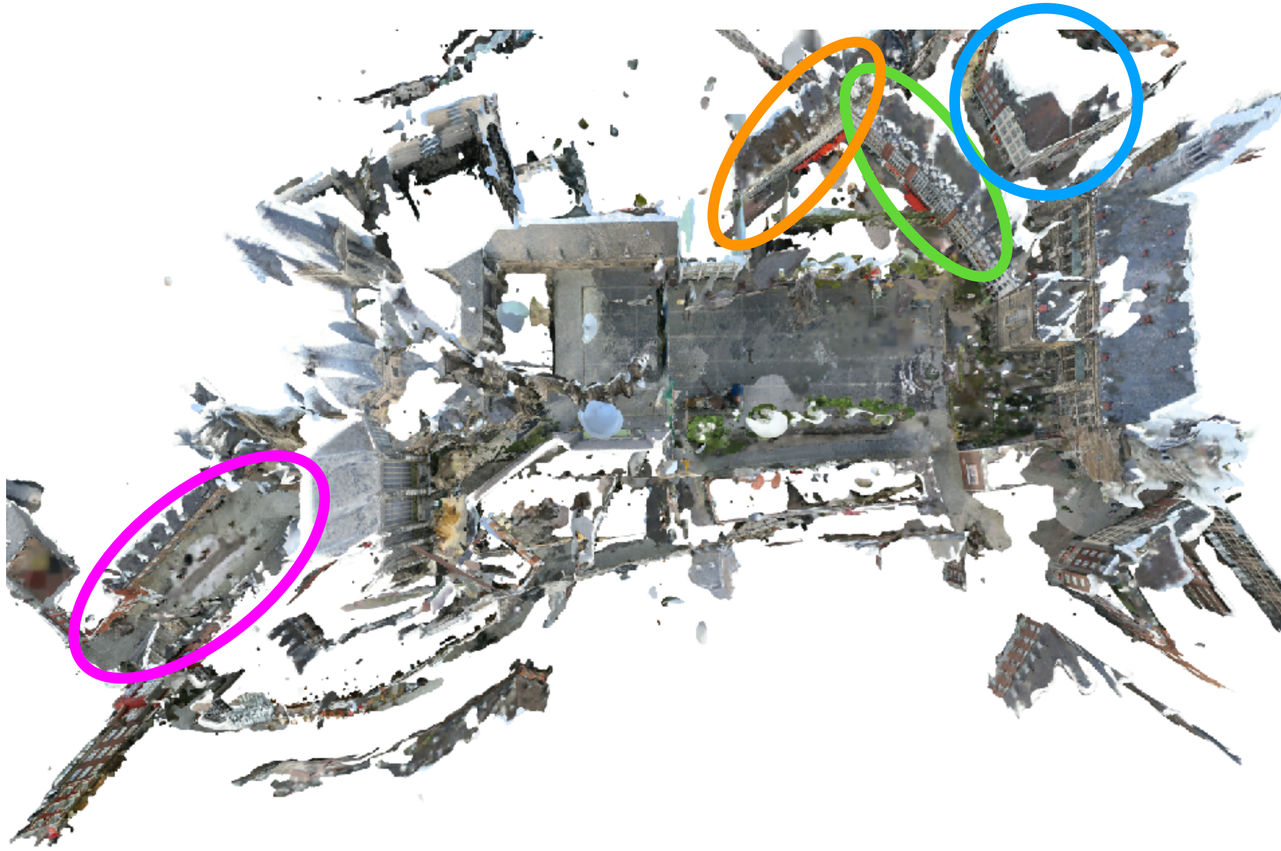
# The Ambiguous

[Dusmanu, Rocco, Pajdla, Pollefeys, Sivic, Torii, Sattler, D2-Net: A Trainable CNN for Joint Detection and Description of Local Features, CVPR 2019]

# The Ambiguous



SIFT daytime



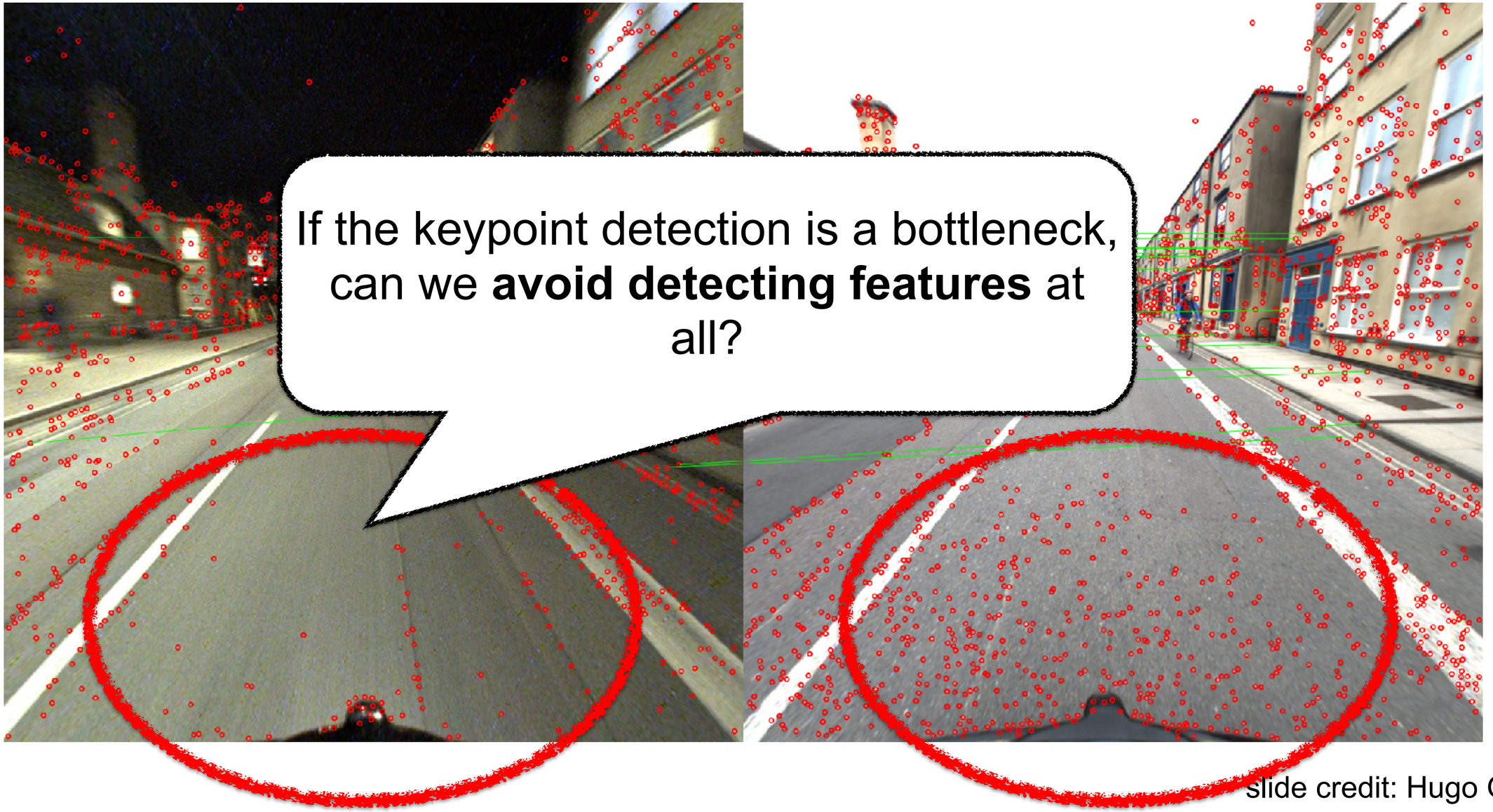
D2-Net daytime

[Zhang, Sattler, Scaramuzza, Reference Pose Generation for Long-term Visual Localization via Learned Features and View Synthesis, IJCV 2020]

# Learning Local Features for Long-Term Localization

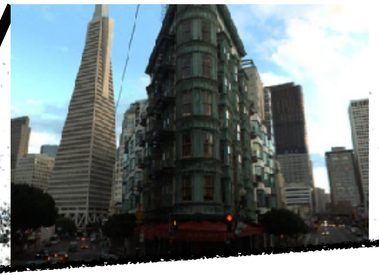
- D2-Net just one example for learned features (SuperPoint and R2D2 are notable others)
- Learned local features **much** more robust to changes in viewing conditions
- Robustness comes at a prize: Often many matches with irrelevant parts of the scene
- Learning robust and descriptive features still open problem
- Long-term localization far from being solved

# An Alternative Perspective

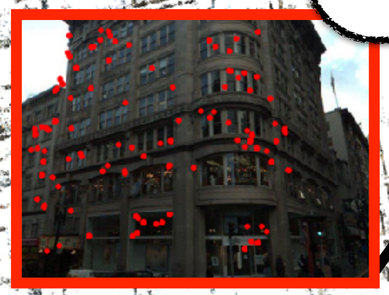


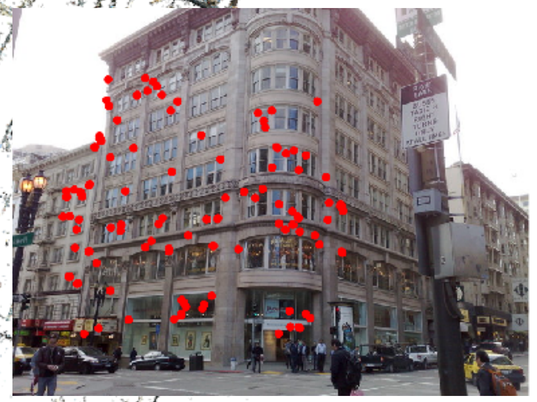

slide credit: Hugo Germanin

# Recap: Image Retrieval-based Localization



Gives us an initial estimate for the pose (pose of retrieved image)!





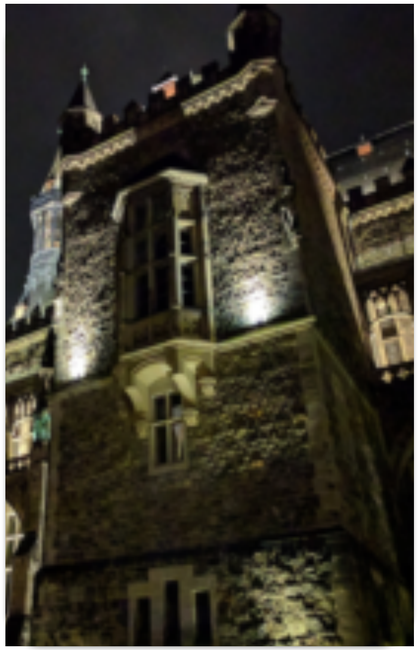
Perform **image retrieval**

Establish **2D-2D matches**

Establish **2D-3D matches**

Robust **pose estimation**

# Direct Pose Refinement



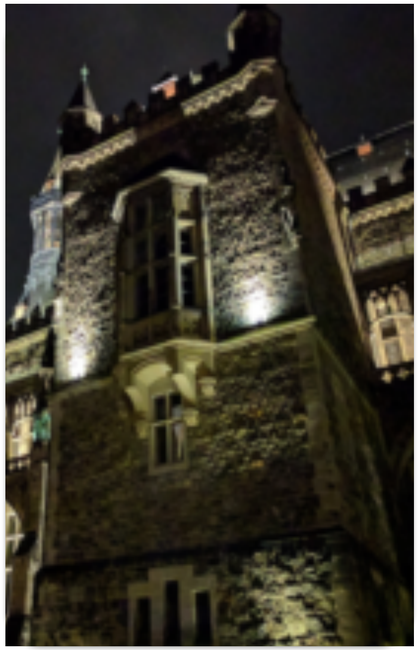
query image

retrieved image

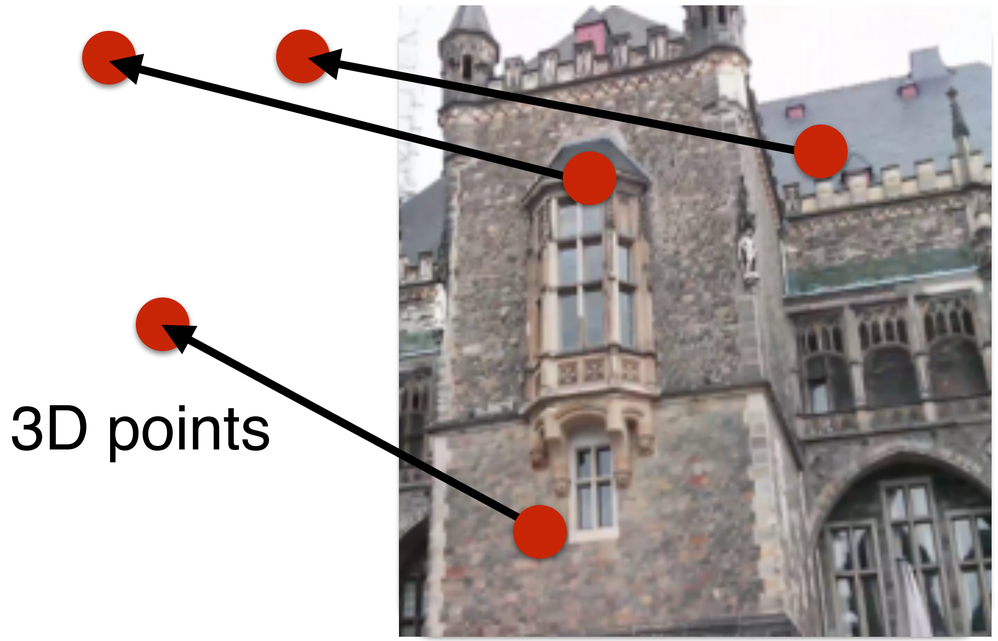
slide credit: Paul-Edouard Sarlin

[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

# Direct Pose Refinement



query image

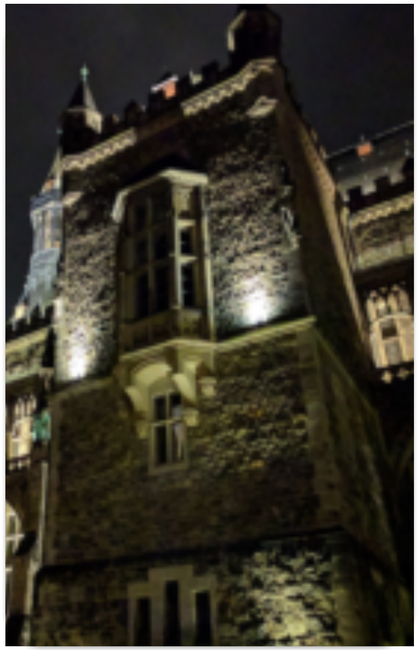



retrieved image

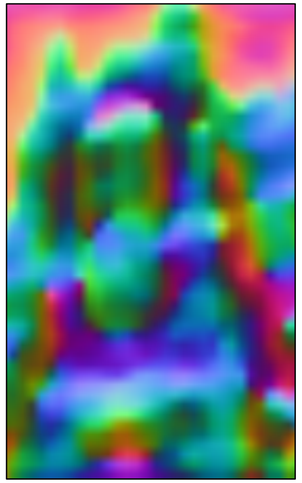
slide credit: Paul-Edouard Sarlin

[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

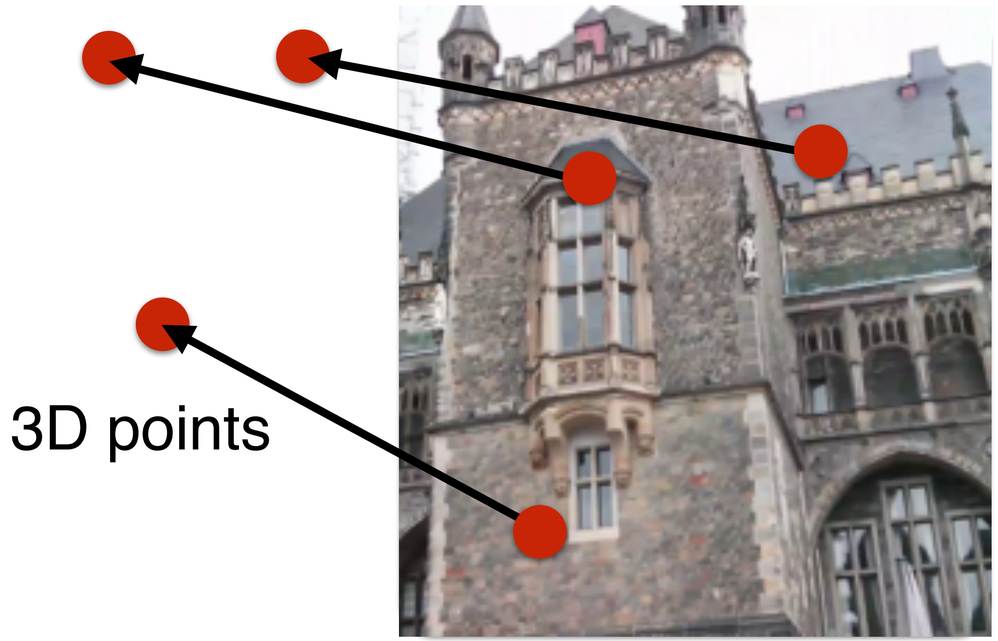
# Direct Pose Refinement



CNN



query image

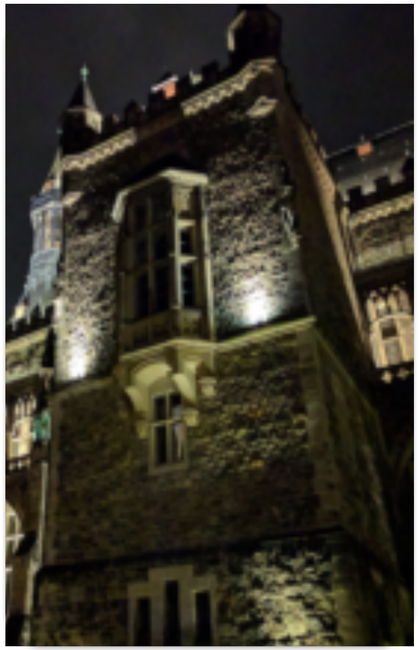



retrieved image

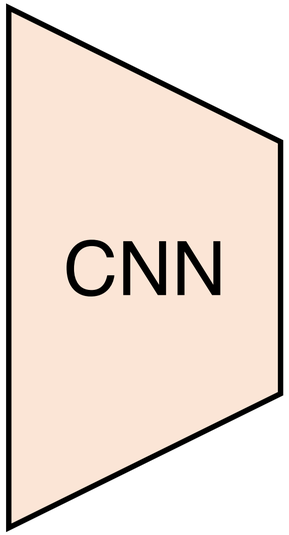
slide credit: Paul-Edouard Sarlin

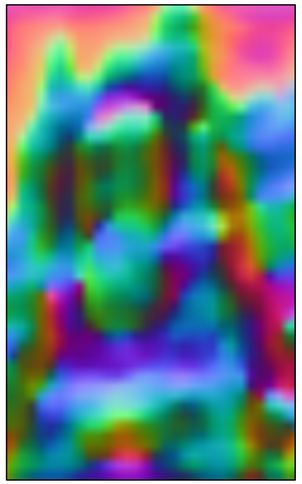
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

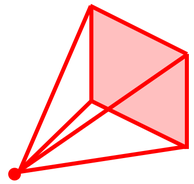
# Direct Pose Refinement



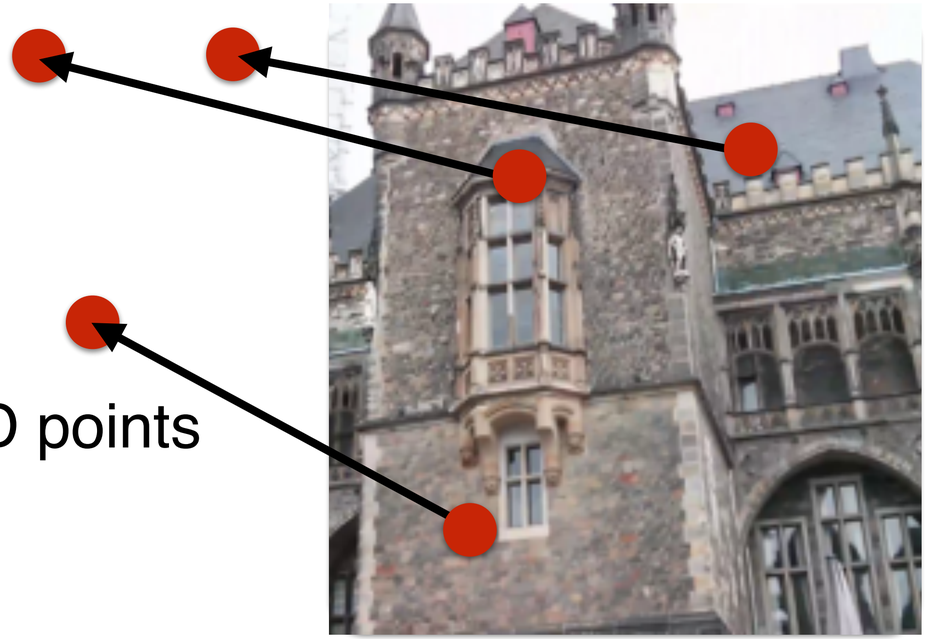
query image





pose estimate

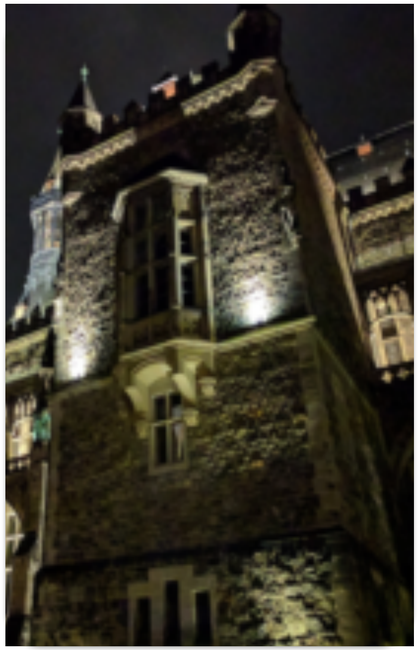



retrieved image

slide credit: Paul-Edouard Sarlin

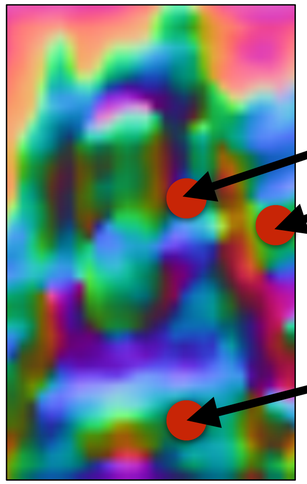
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

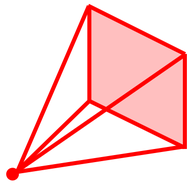
# Direct Pose Refinement



query image

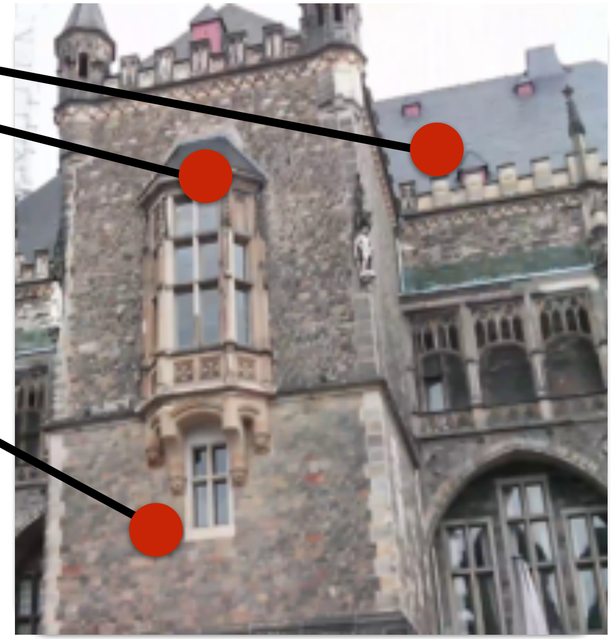
CNN





pose estimate

3D points

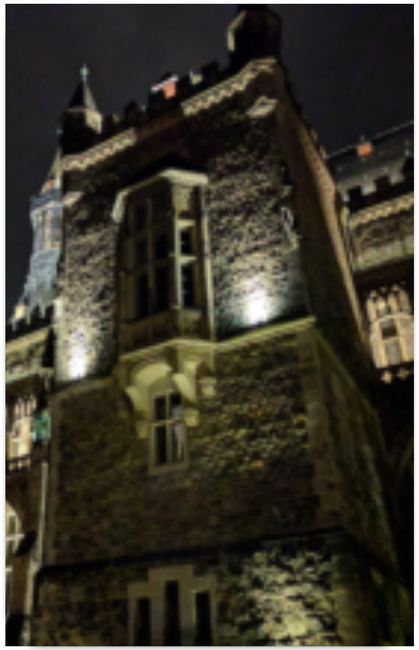


retrieved image

slide credit: Paul-Edouard Sarlin

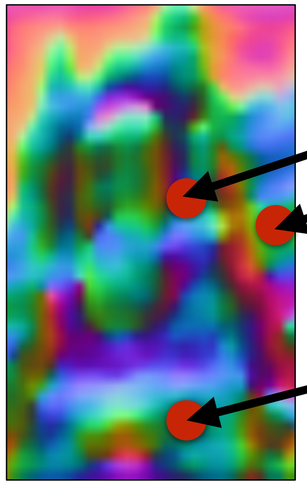
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

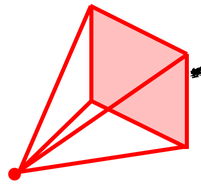
# Direct Pose Refinement



query image

CNN

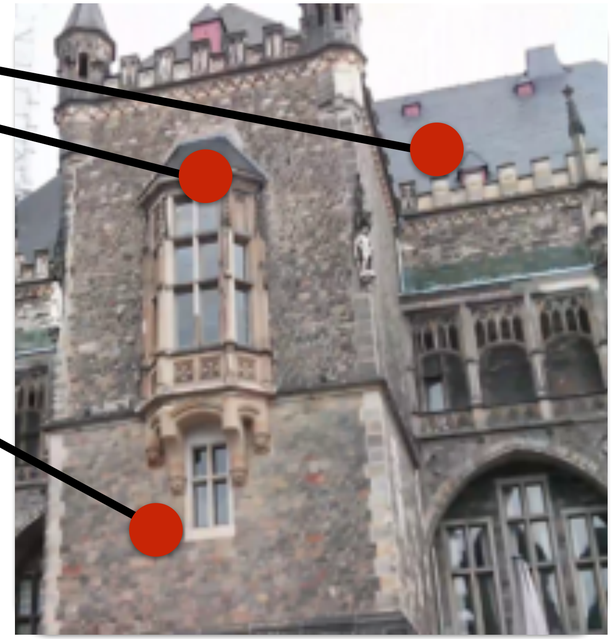




pose estimate

find displacements

3D points

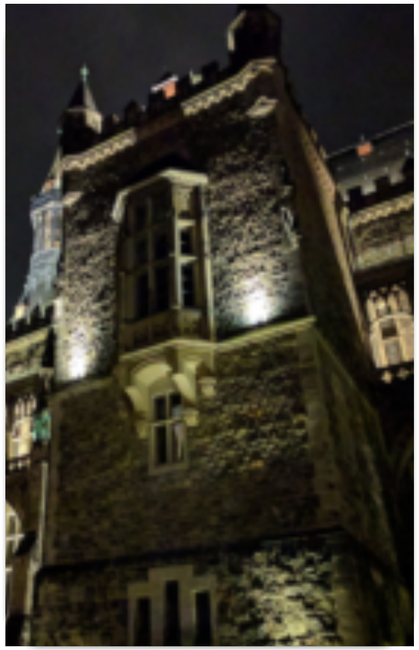


retrieved image

slide credit: Paul-Edouard Sarlin

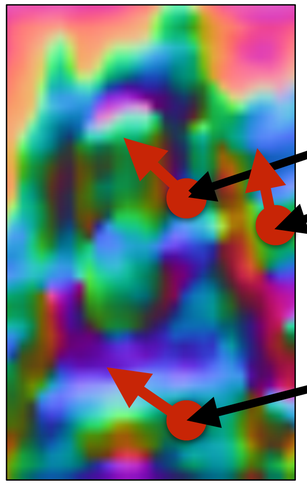
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

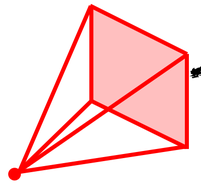
# Direct Pose Refinement



query image

CNN

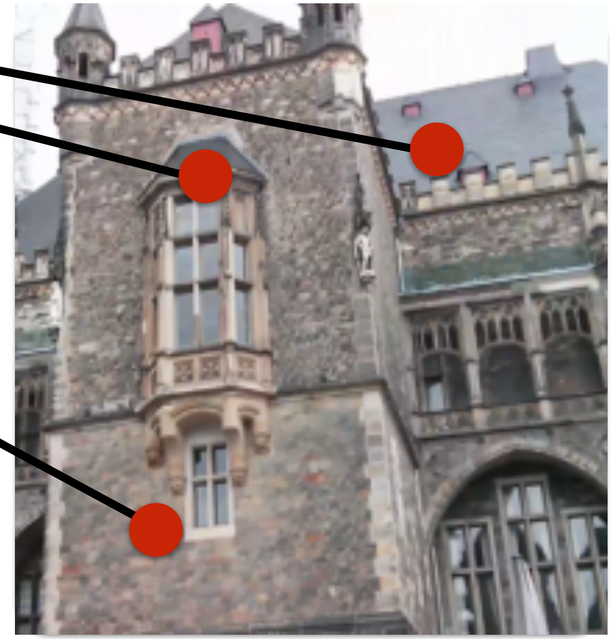




pose estimate

find displacements

3D points

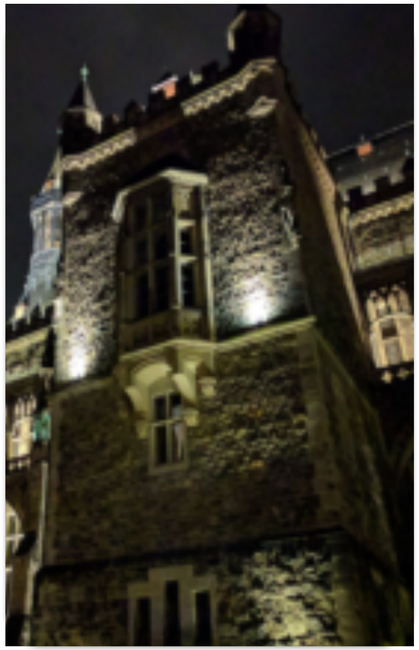


retrieved image

slide credit: Paul-Edouard Sarlin

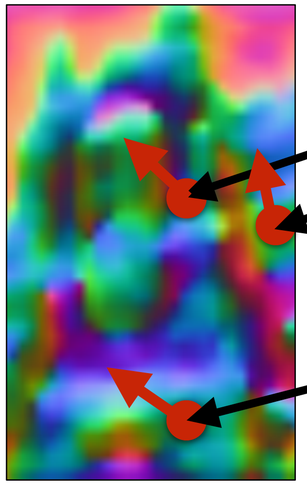
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

# Direct Pose Refinement

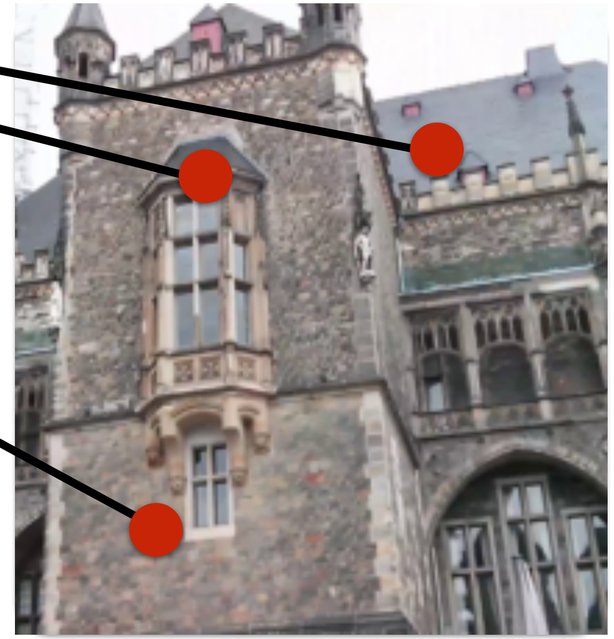


query image

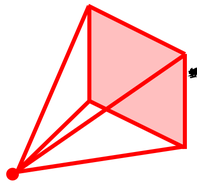
CNN



3D points



retrieved image



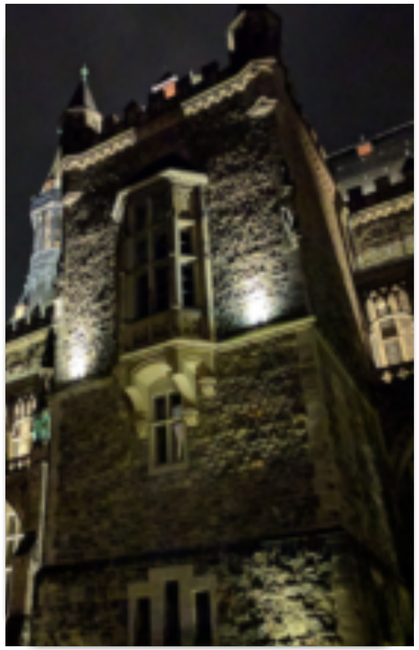
pose estimate

find displacements

slide credit: Paul-Edouard Sarlin

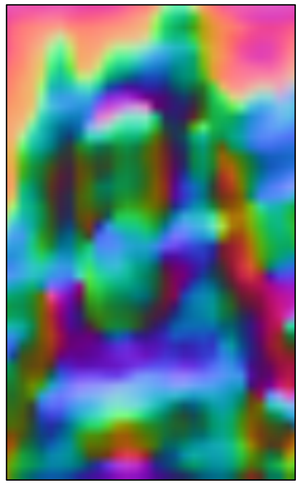
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

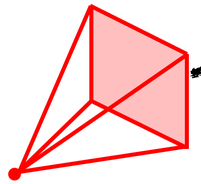
# Direct Pose Refinement



query image

CNN

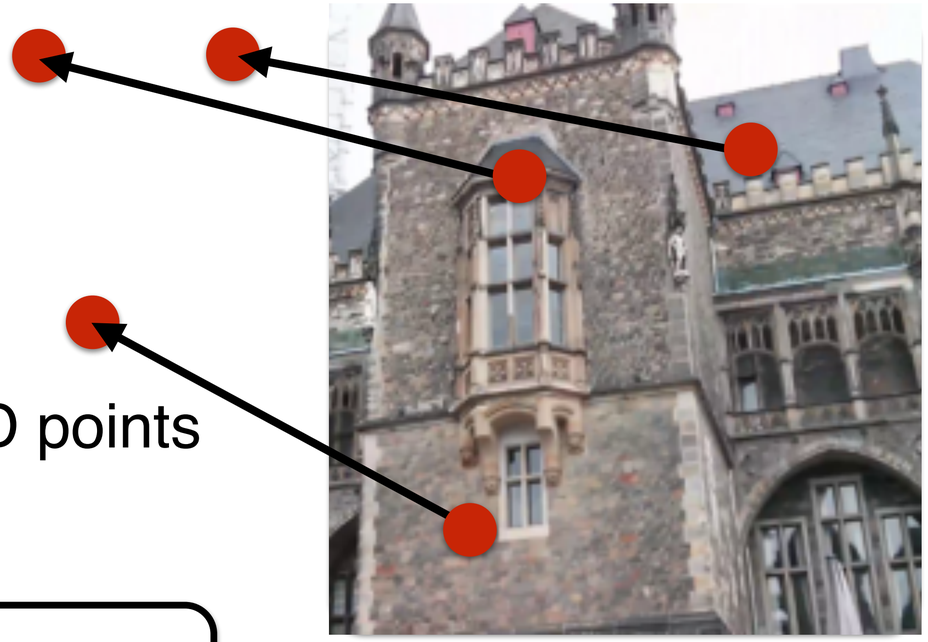




pose estimate

find displacements

3D points

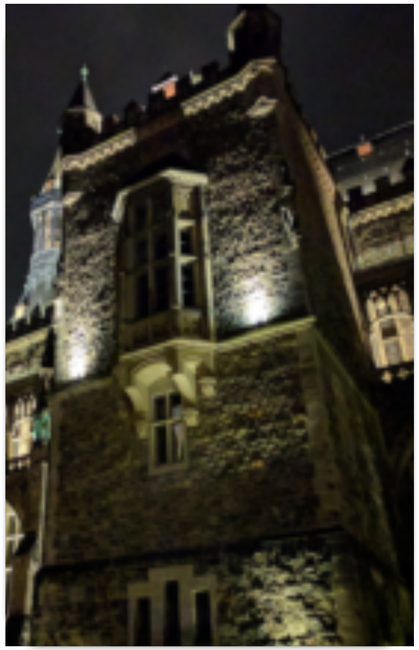


retrieved image

slide credit: Paul-Edouard Sarlin

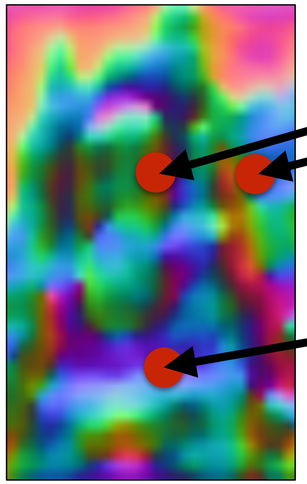
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

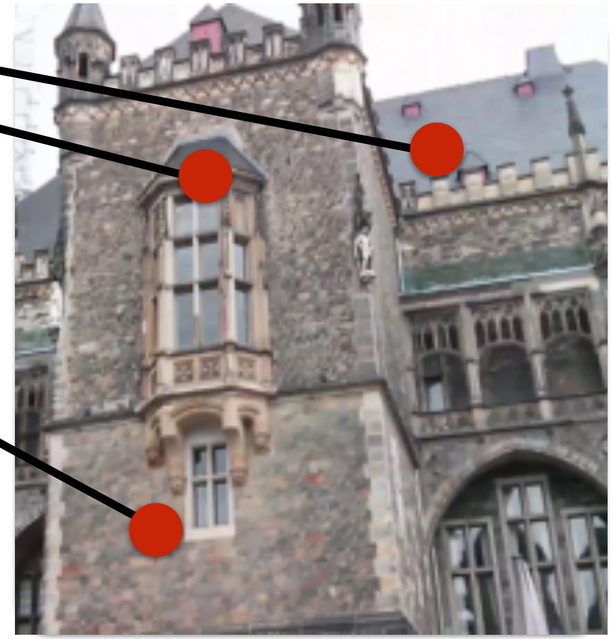
# Direct Pose Refinement



query image

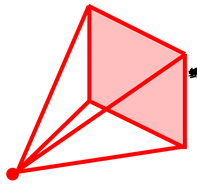
CNN





3D points

retrieved image



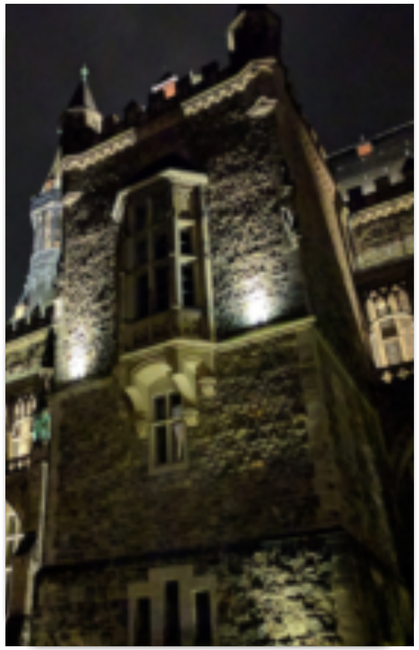
pose estimate

find displacements

slide credit: Paul-Edouard Sarlin

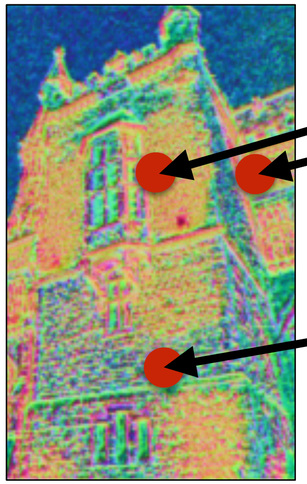
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

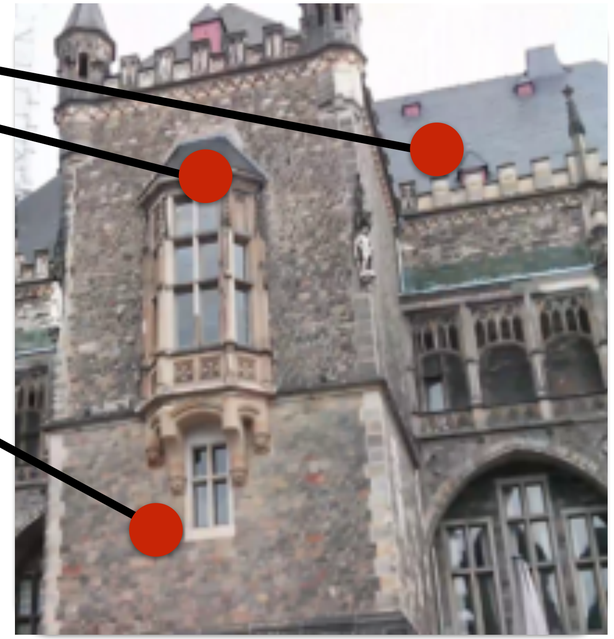
# Direct Pose Refinement



query image

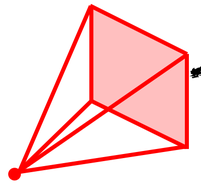
CNN





3D points

retrieved image



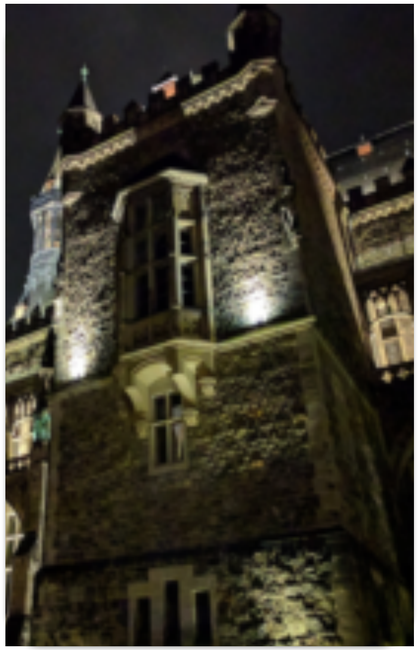
pose estimate

find displacements

slide credit: Paul-Edouard Sarlin

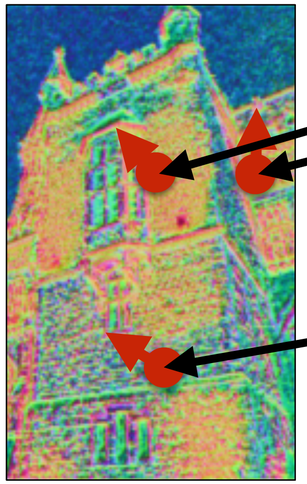
[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

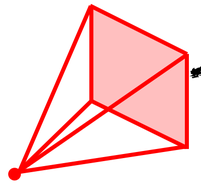
# Direct Pose Refinement



query image

CNN

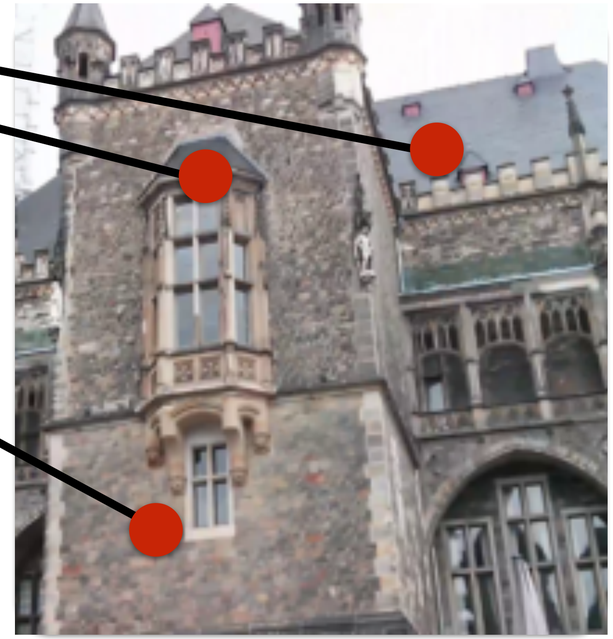




pose estimate

find displacements

3D points

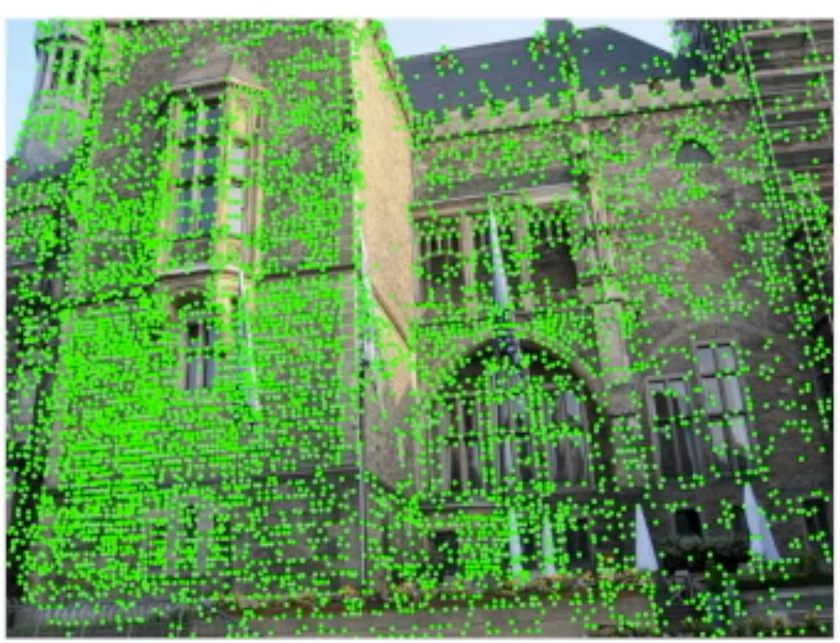


retrieved image

slide credit: Paul-Edouard Sarlin

[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

# Direct Pose Refinement



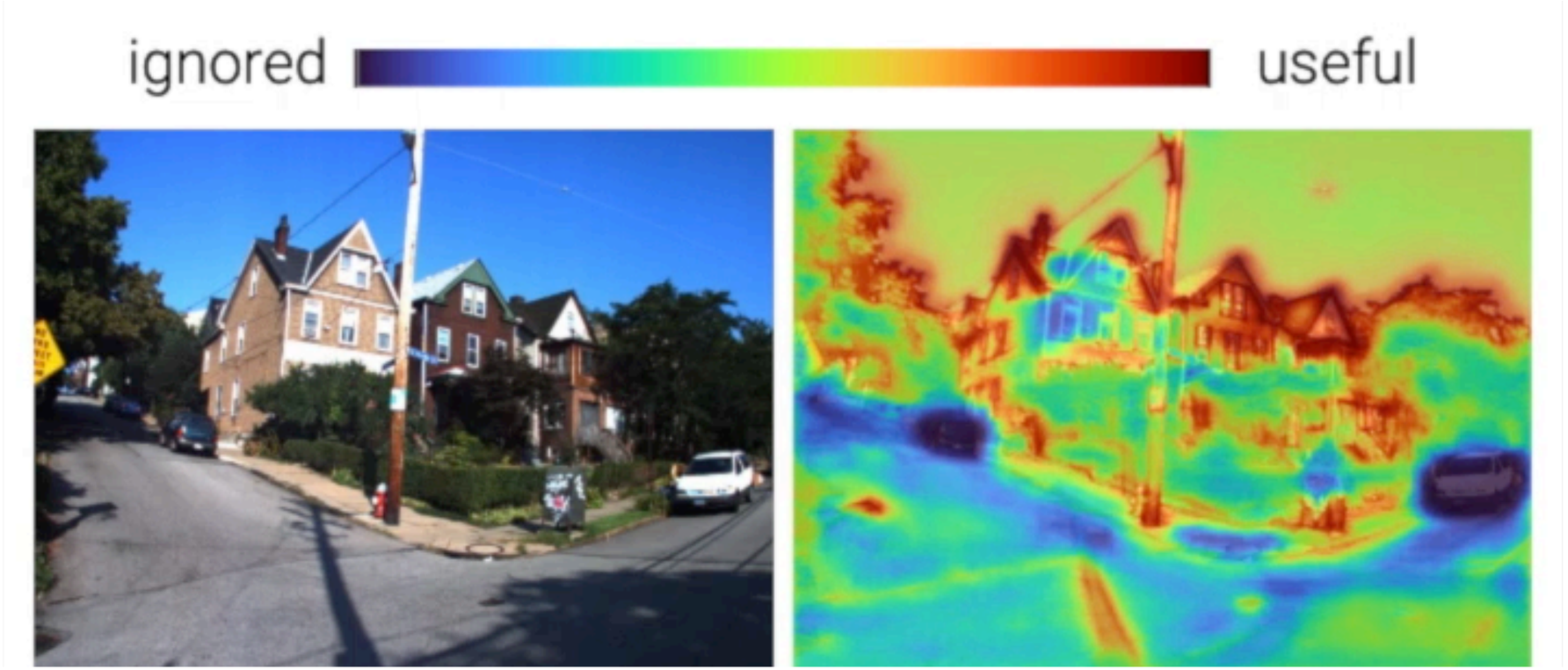
reference image

query image

slide credit: Paul-Edouard Sarlin

[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

# Learning Feature Confidence




reference image

query image

slide credit: Paul-Edouard Sarlin

[P.-E. Sarlin, A. Unagar, M. Larsson, H. Germain, C. Toft, V. Larsson, M. Pollefeys, V. Lepetit, L. Hammarstrand, F. Kahl, T. Sattler, Back to the Feature: Learning Robust Camera Localization from Pixels to Pose, CVPR 2021]

# Direct Pose Refinement

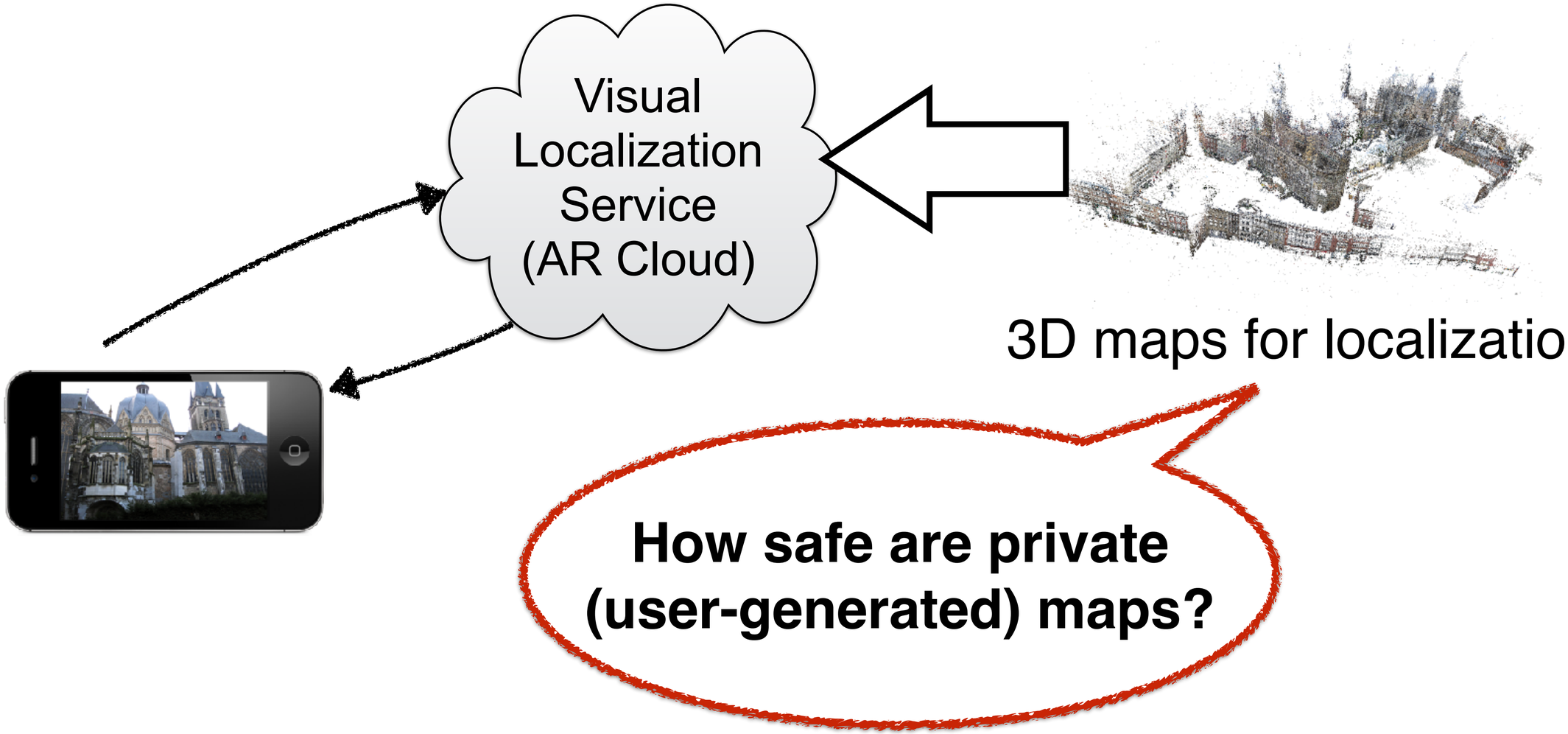
- Has been used in the context of Simultaneous Localization and Mapping / Structure-from-Motion for quite some time
- Becoming popular for long-term localization: learn robust feature maps (see also work by Daniel Cremers' group)
- Training involves solving classical reprojection error via least-squares fitting (e.g., Levenberg-Marquardt)

# Quiz Time

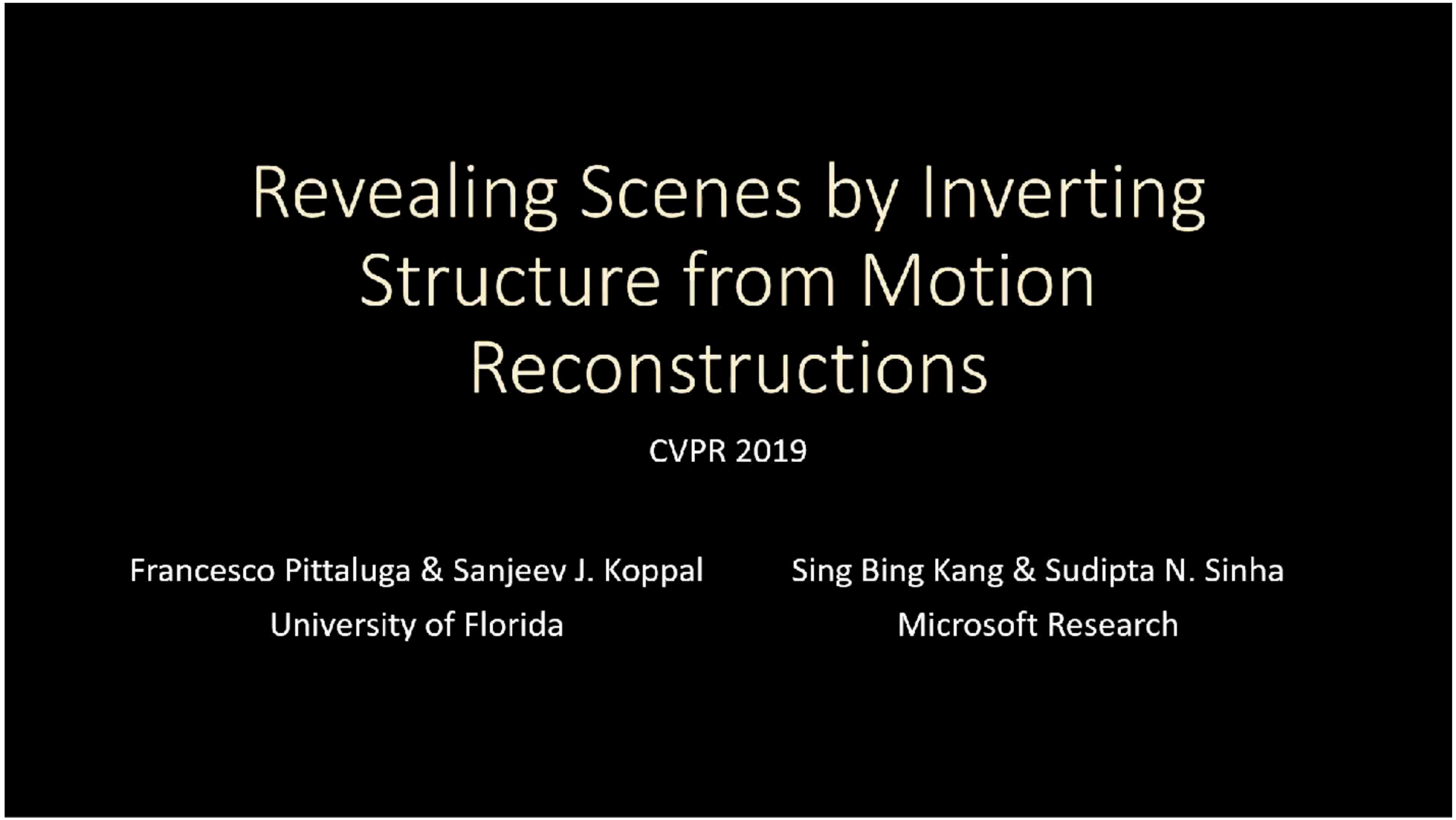
# Overview

- A (Too) Simple Approach to Visual Localization
- Structure-Based Localization
- Long-Term Localization
- **Privacy-Preserving Localization**

# The AR Cloud

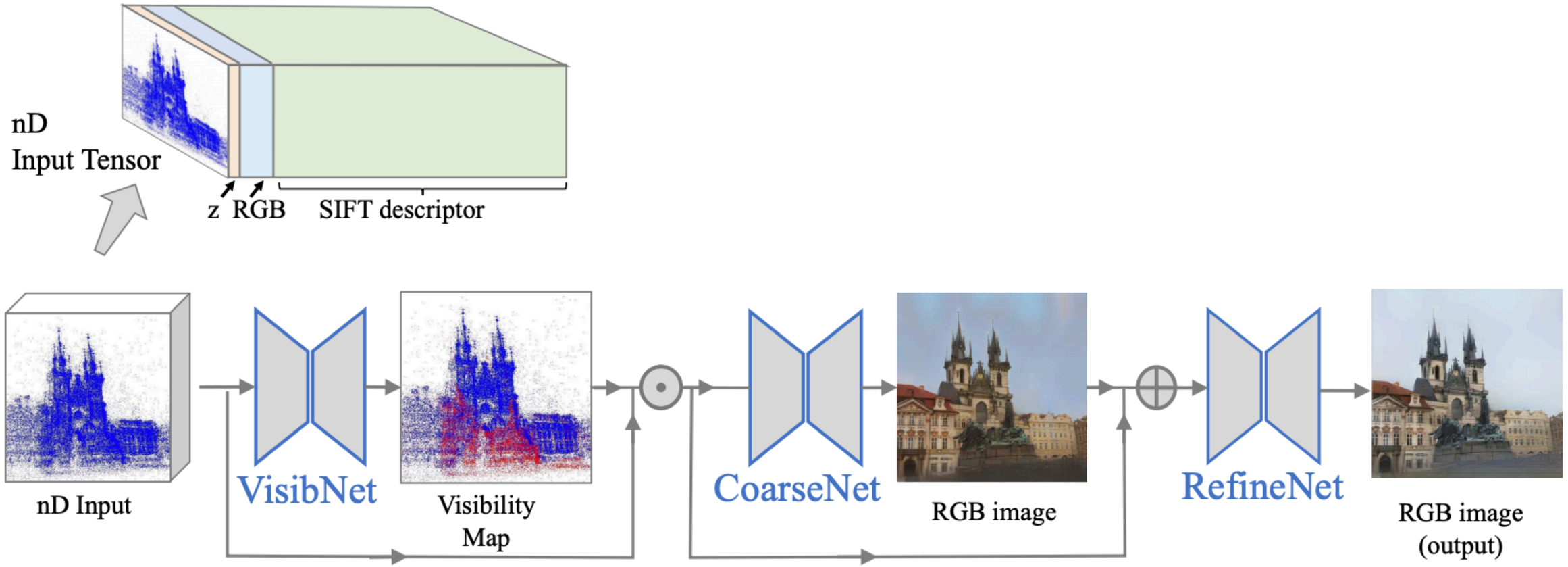

Visual
Localization
Service
(AR Cloud)
3D maps for localization
How safe are private
(user-generated) maps?

# Privacy Issues in Visual Localization



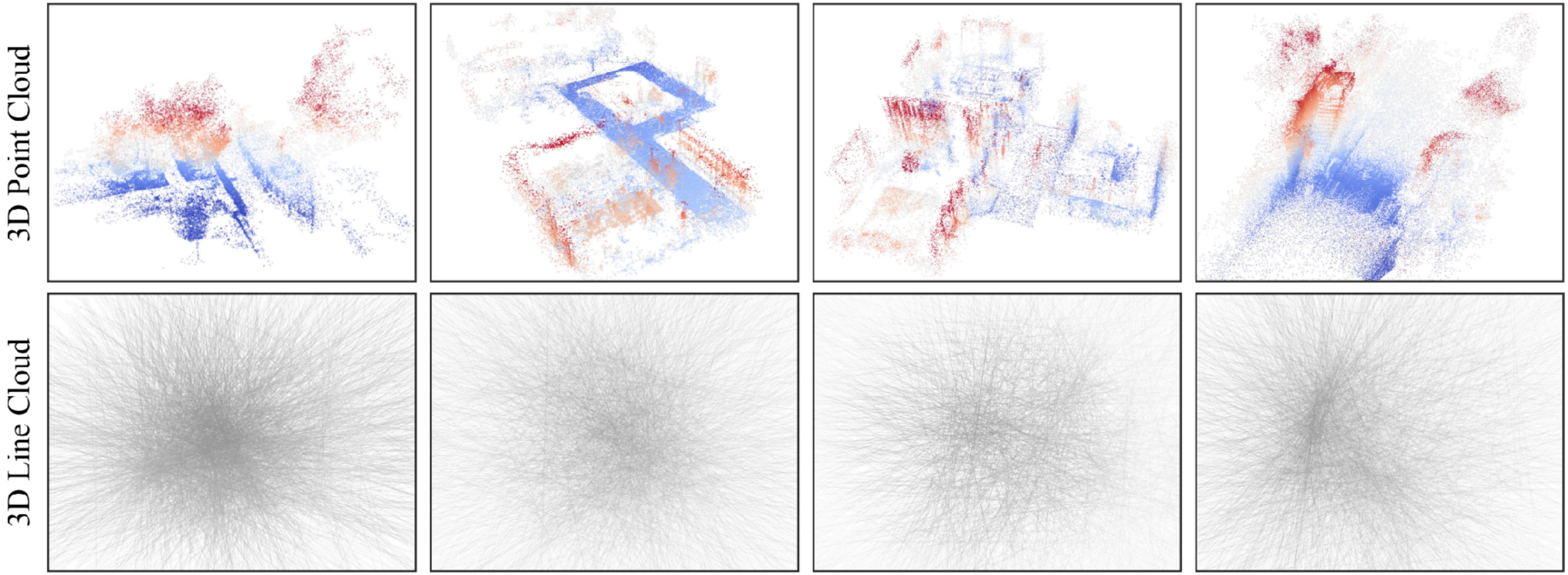

[Pittaluga, Koppal, Kang, Sinha, Revealing Scenes by Inverting Structure From Motion Reconstructions, CVPR 2019]

# Privacy Issues in Visual Localization




[Pittaluga, Koppal, Kang, Sinha, Revealing Scenes by Inverting Structure From Motion Reconstructions, CVPR 2019]

# Privacy Issues in Visual Localization




[Speciale, Schönberger, Kang, Sinha, Pollefeys, Privacy Preserving Image-Based Localization, CVPR 2019]

# Are Line Clouds Necessarily Privacy Preserving?

Lifting points to lines

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Are Line Clouds Necessarily Privacy Preserving?

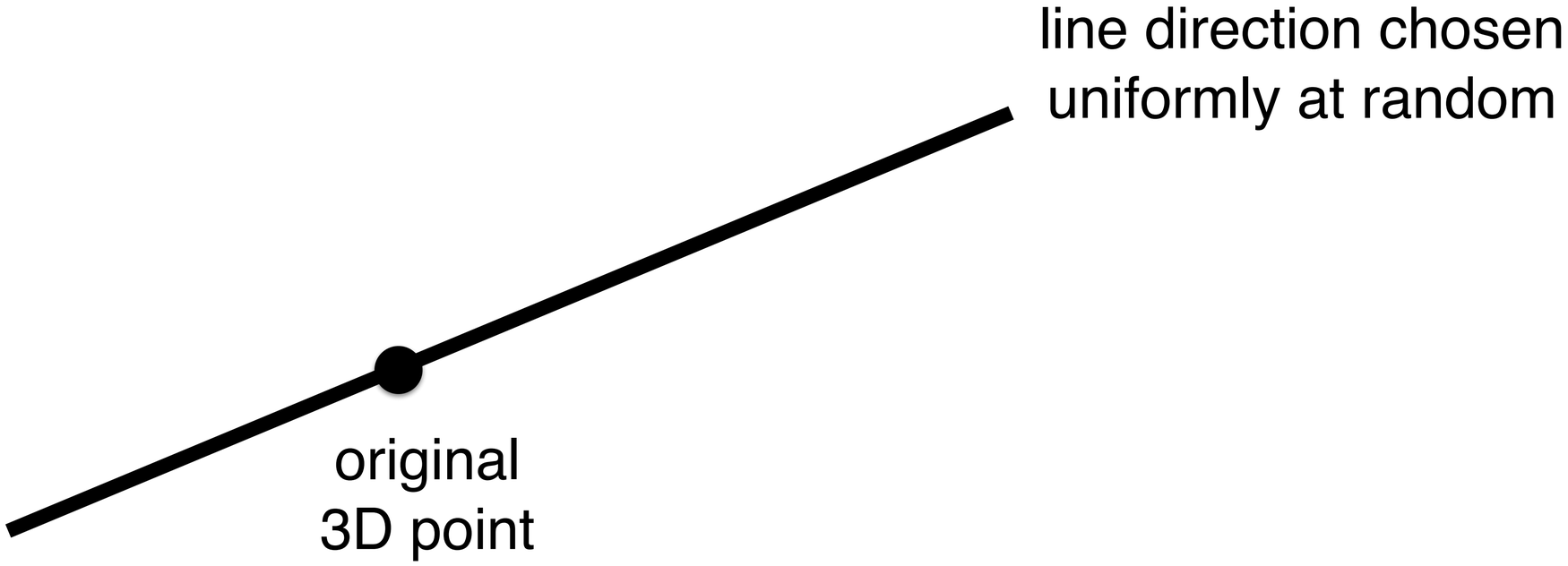
## Lifting points to lines

original
3D point

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Are Line Clouds Necessarily Privacy Preserving?

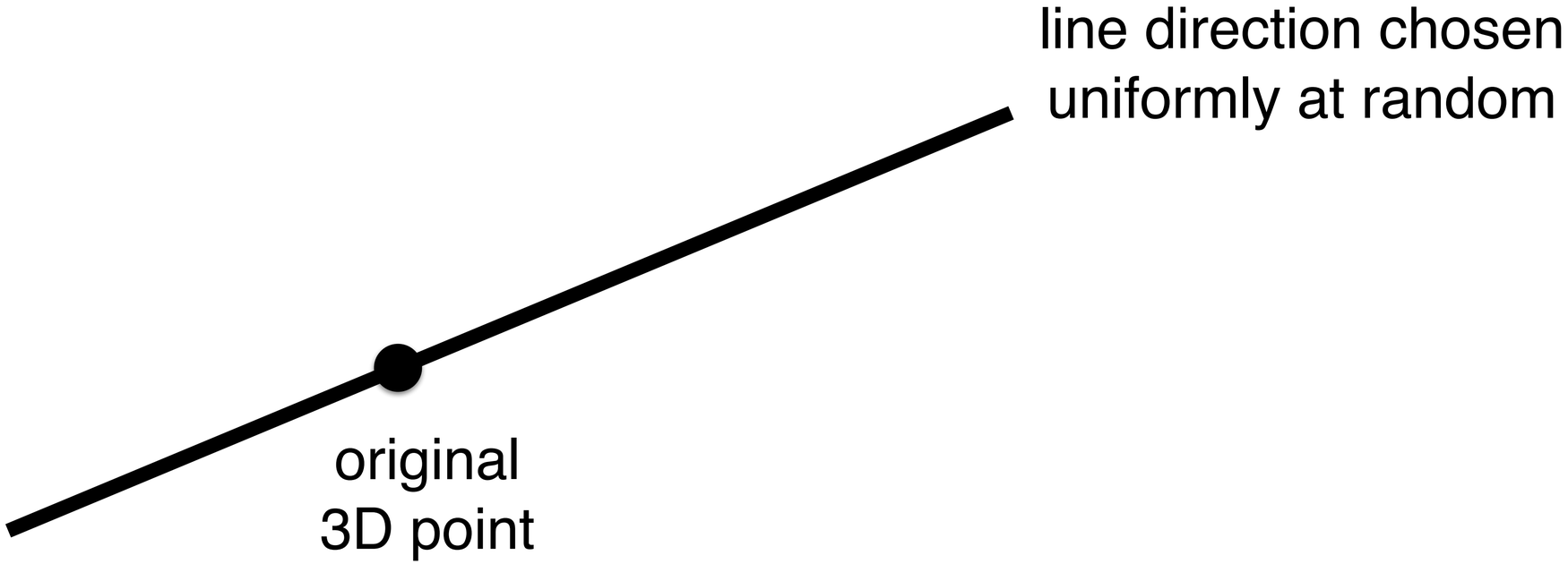
## Lifting points to lines




[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Are Line Clouds Necessarily Privacy Preserving?

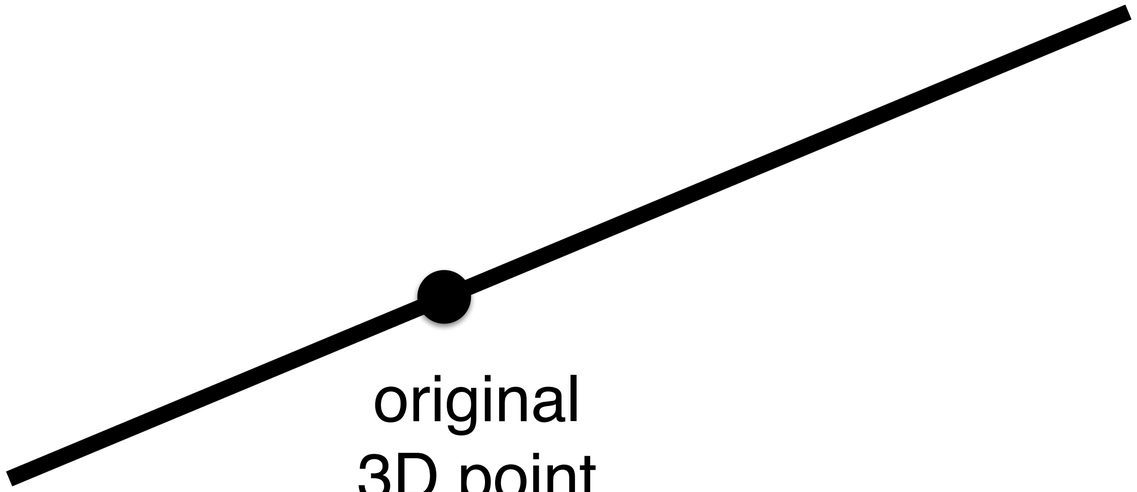
## Lifting points to lines




[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Are Line Clouds Necessarily Privacy Preserving?

## Lifting points to lines

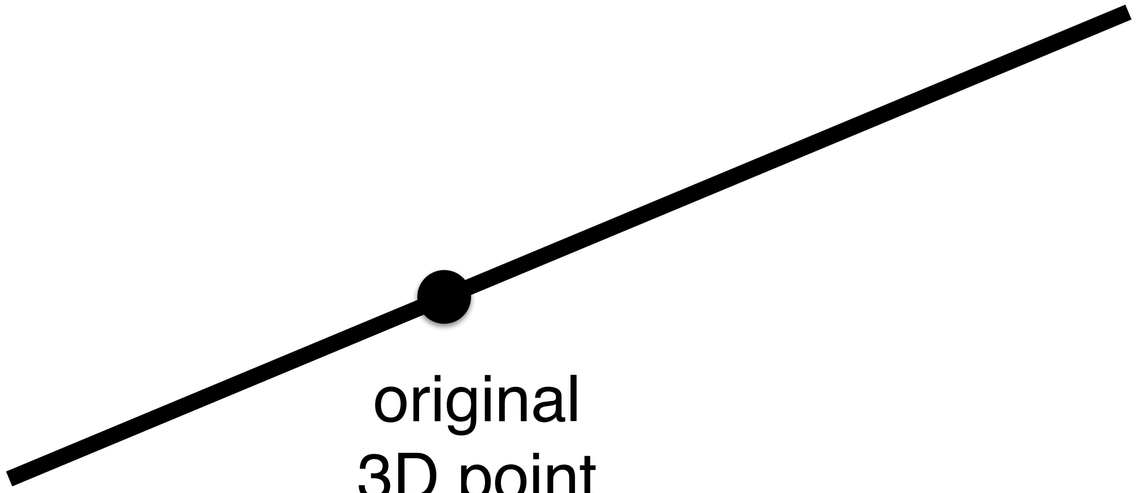



A single line is perfectly privacy-preserving!

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Are Line Clouds Necessarily Privacy Preserving?

## Lifting points to lines

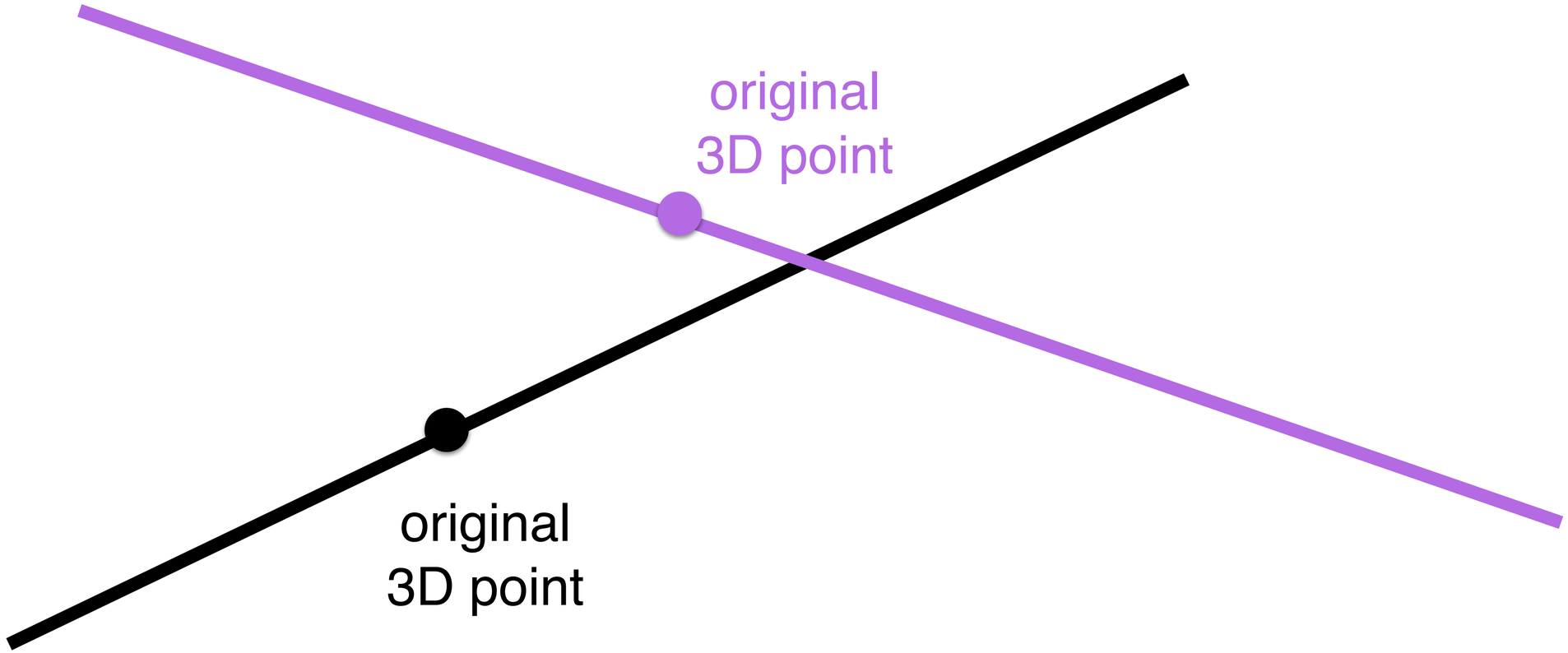



A single line is perfectly privacy-preserving!

But we have many lines!

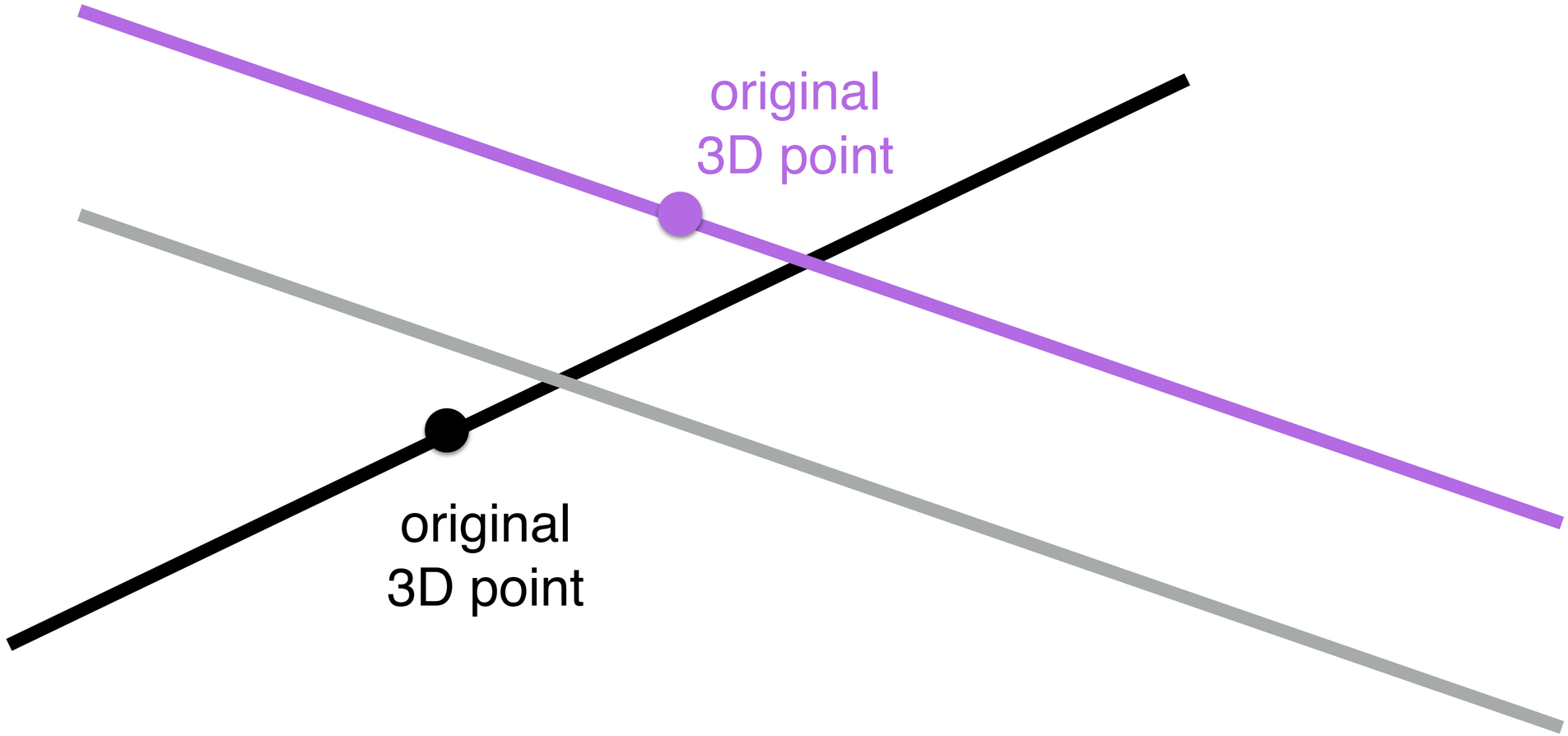
[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Case of Two Lines



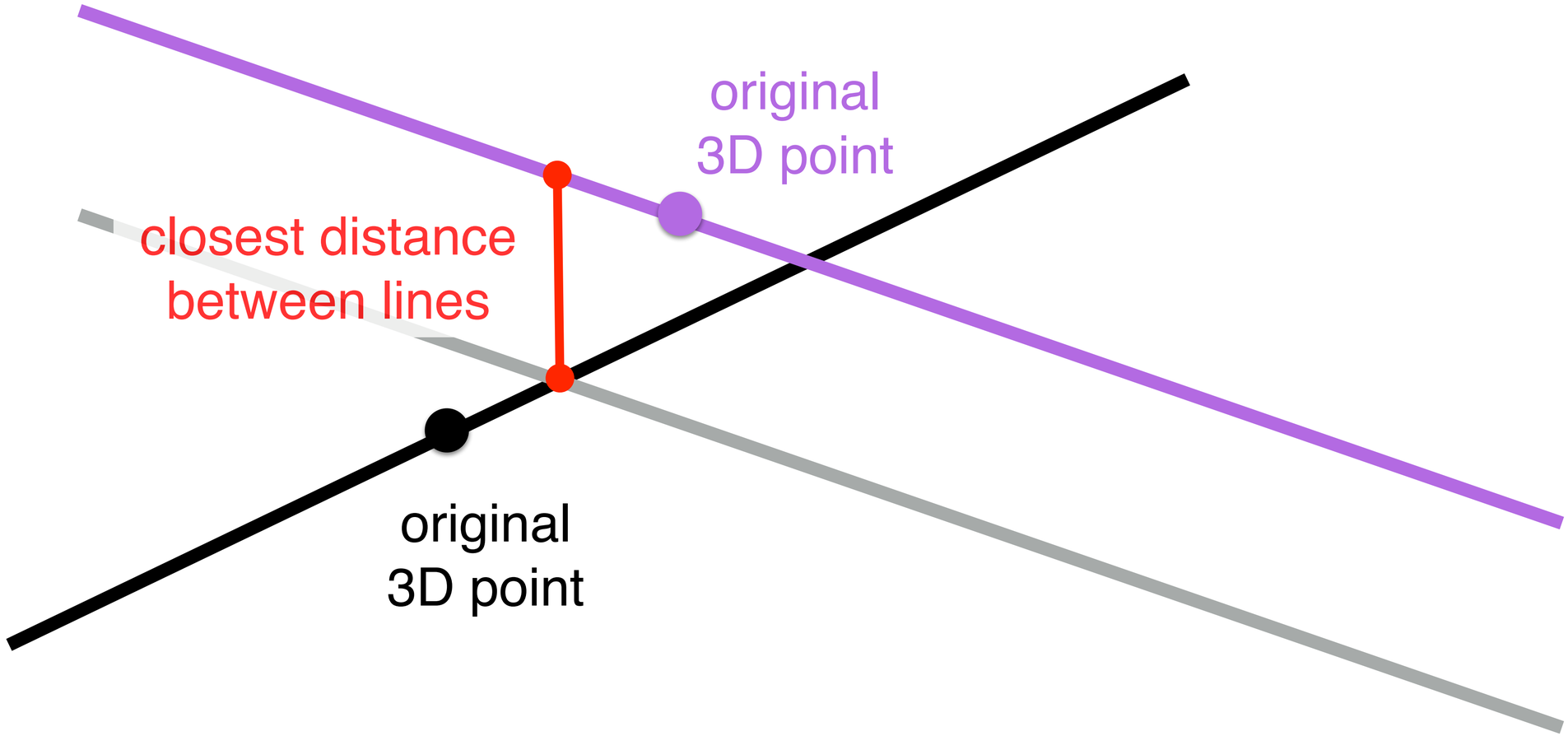

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Case of Two Lines



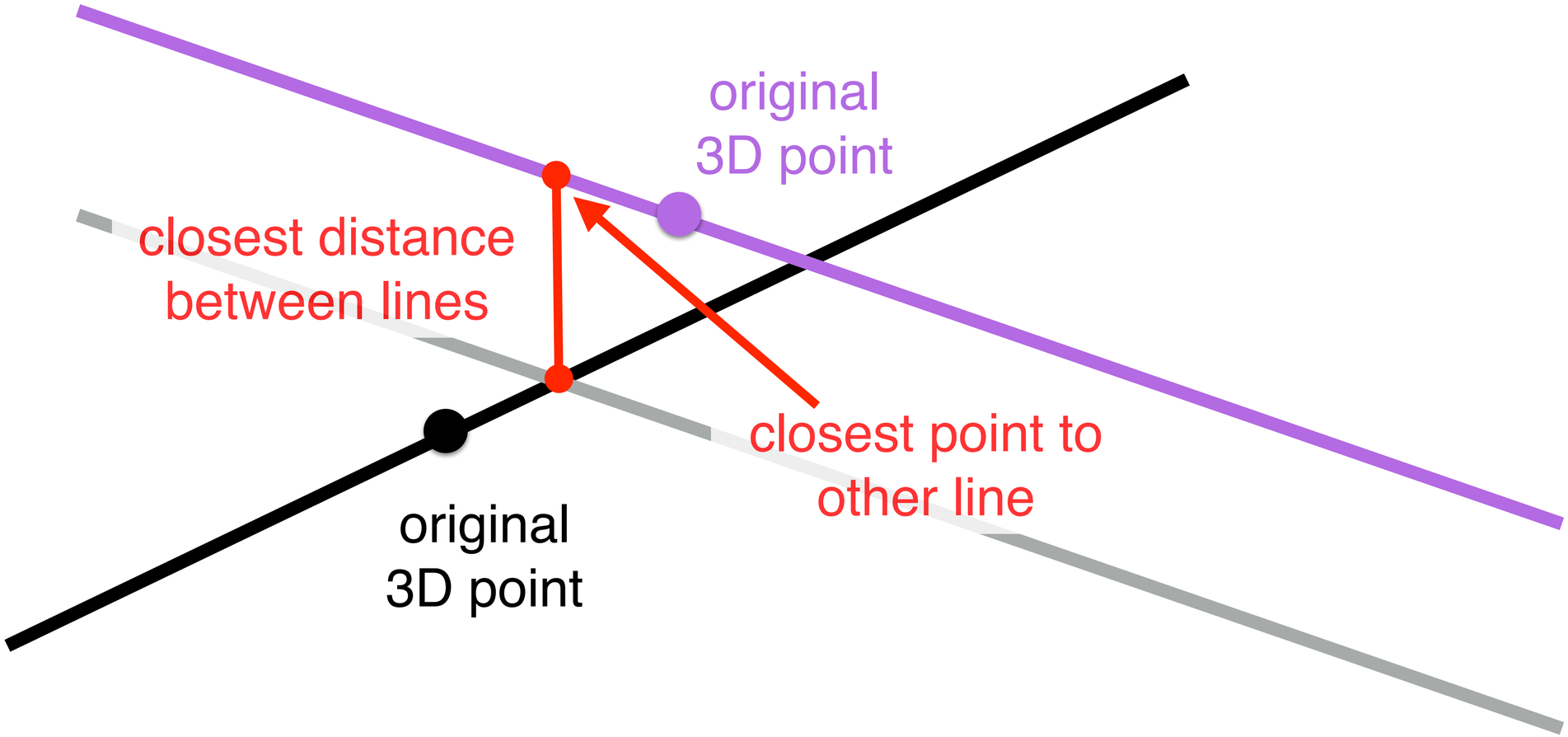

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Case of Two Lines



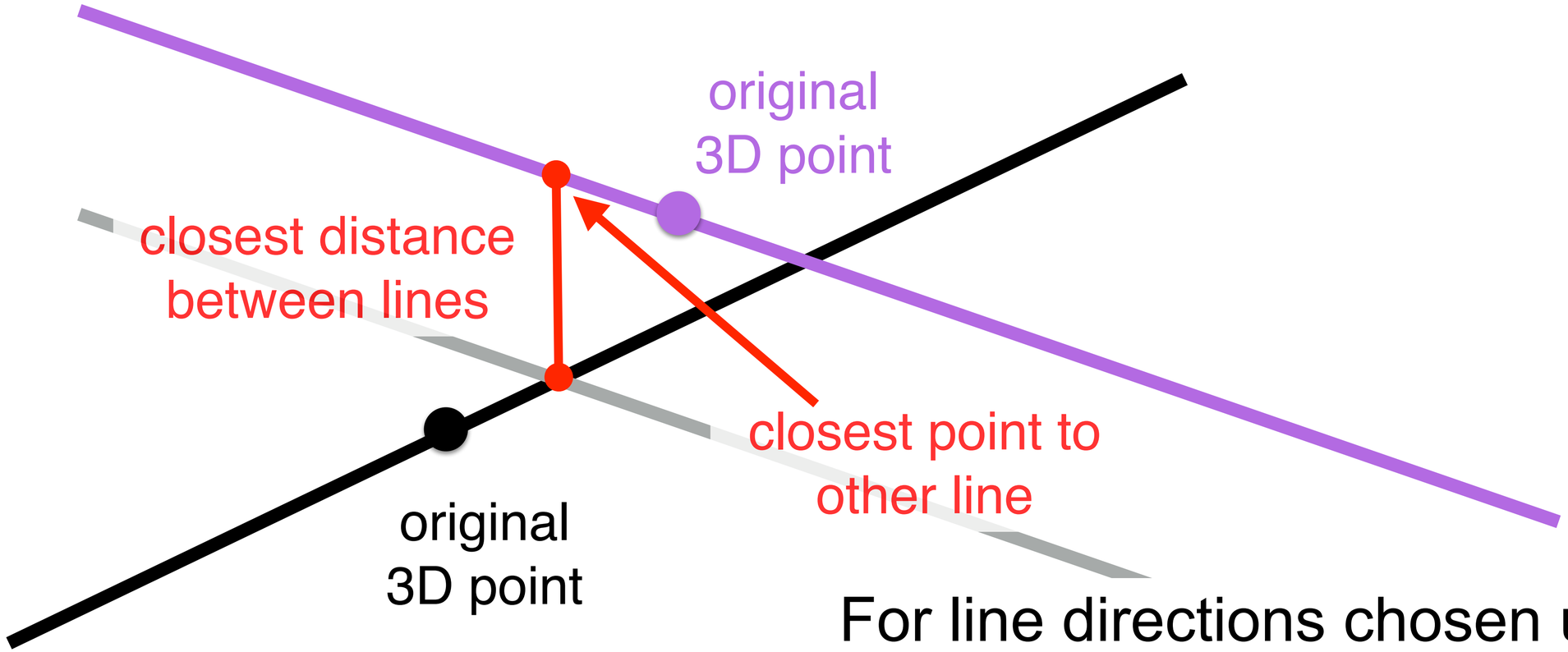

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Case of Two Lines




[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

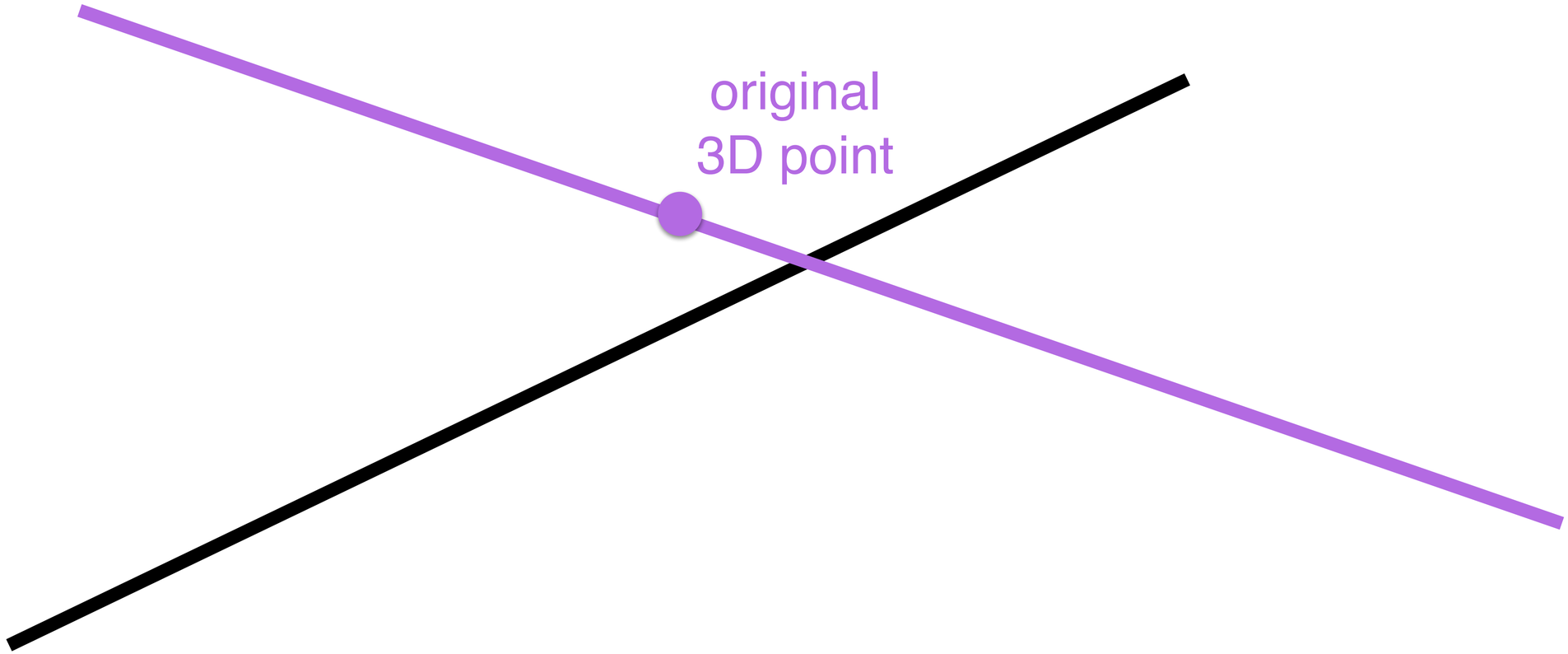
# Case of Two Lines




For line directions chosen uniformly at random, closest point often a good approximation to original point!

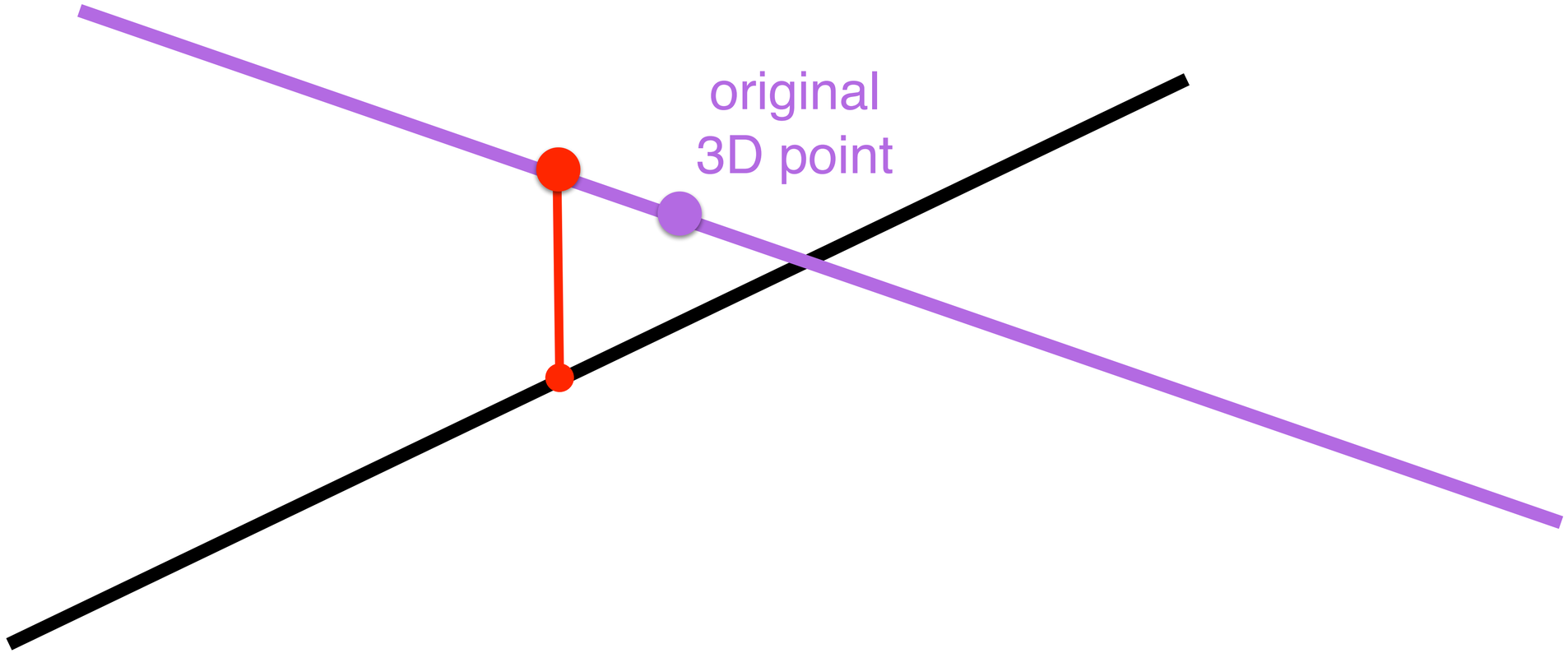
[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# N Line Case



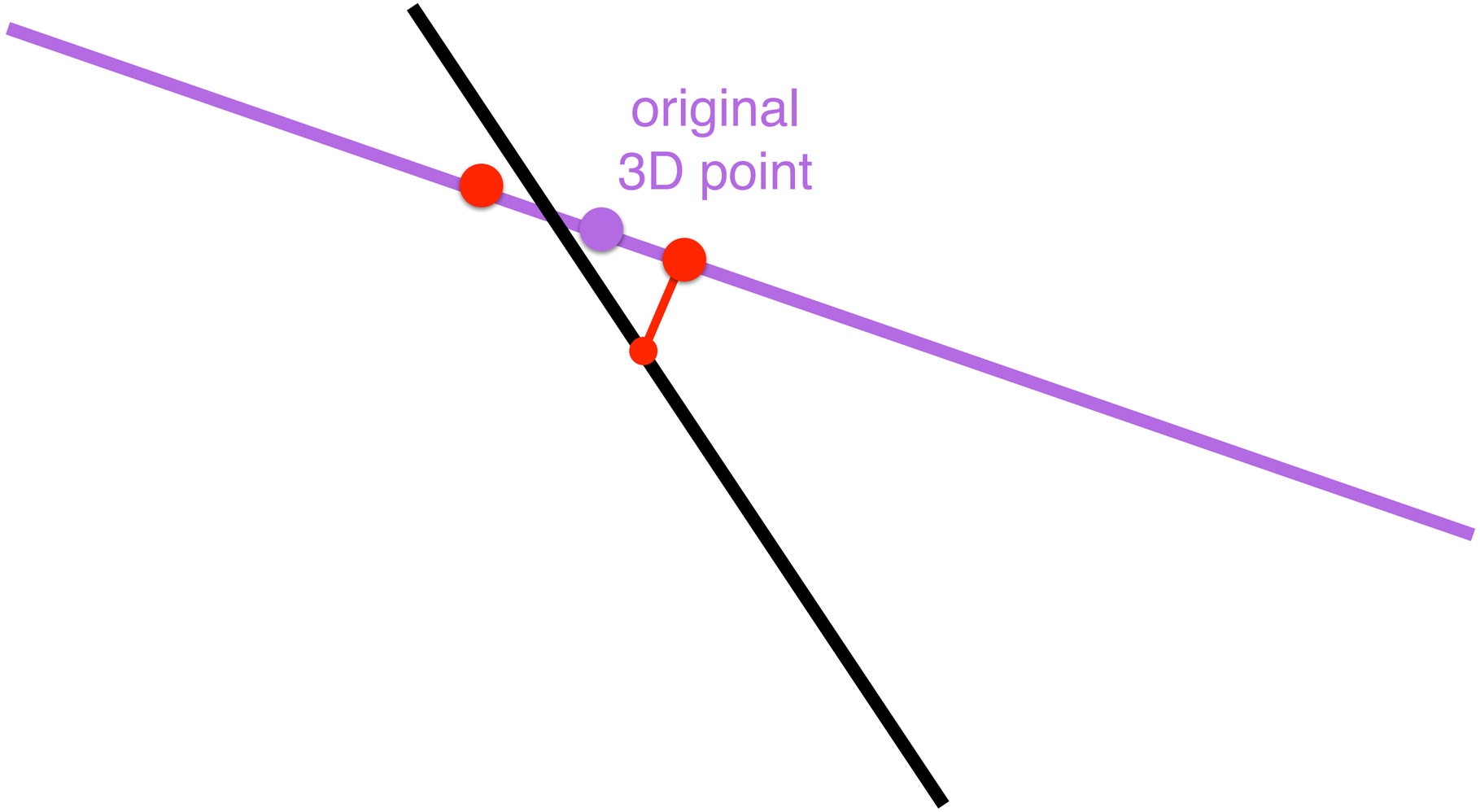

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# N Line Case



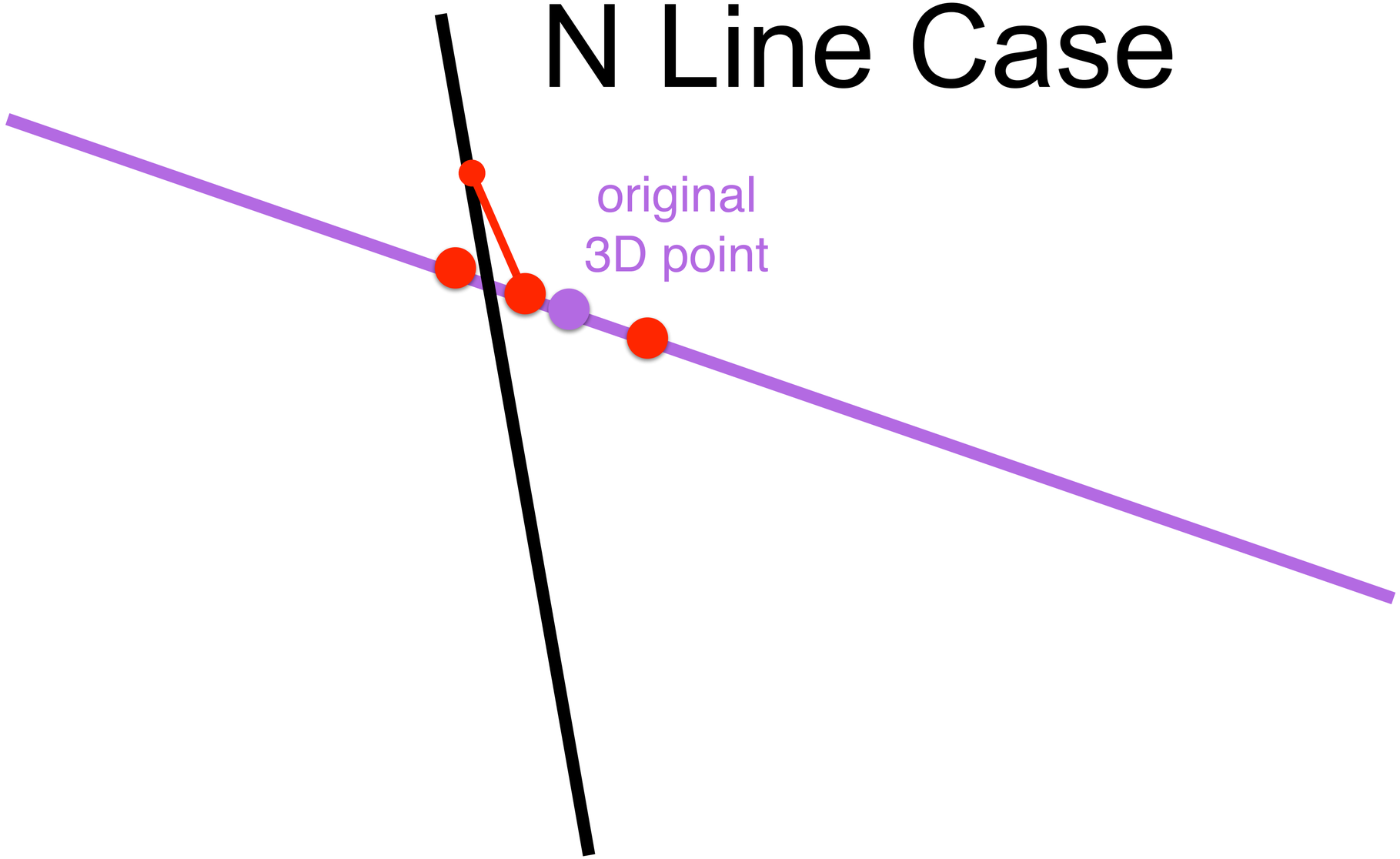

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# N Line Case



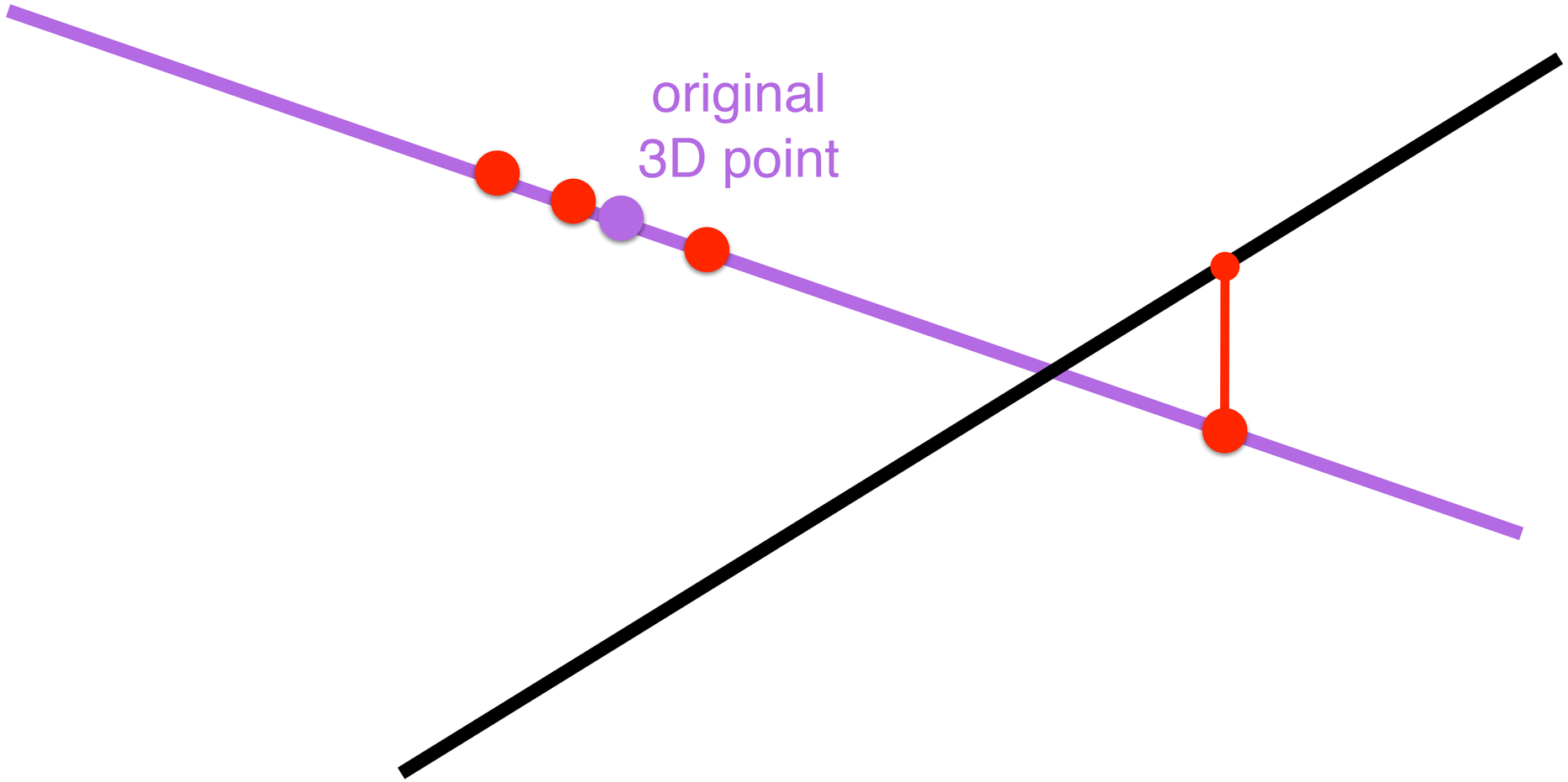

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# N Line Case



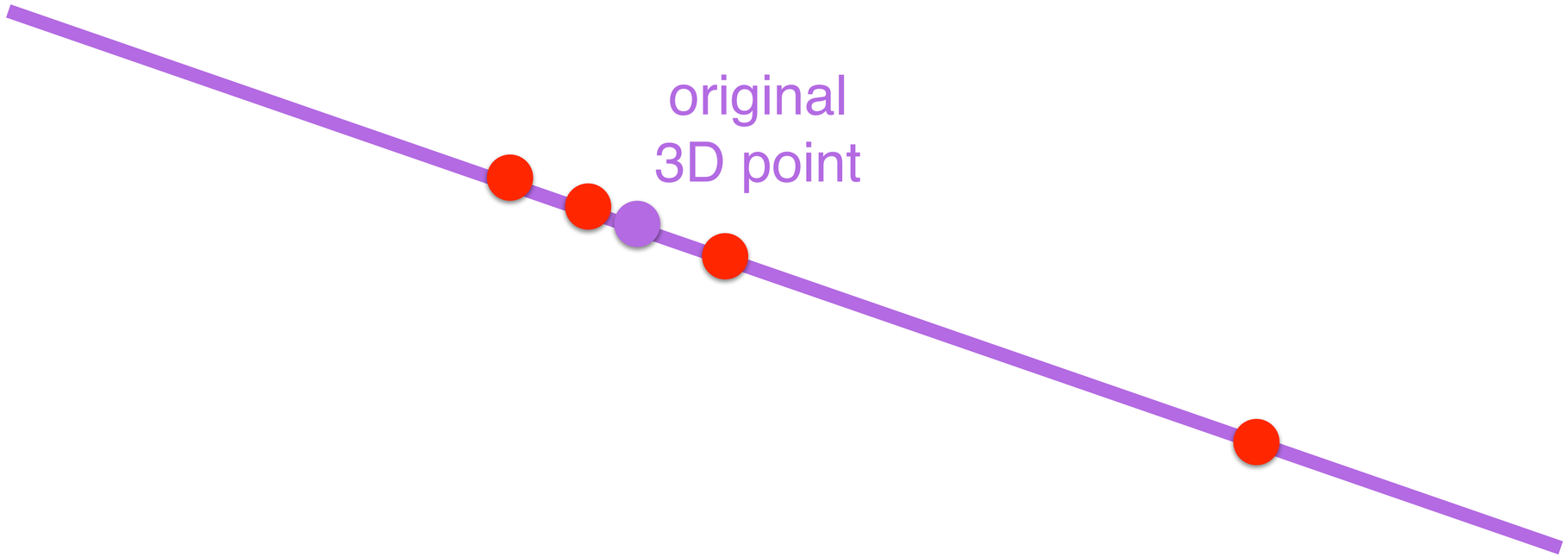

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# N Line Case



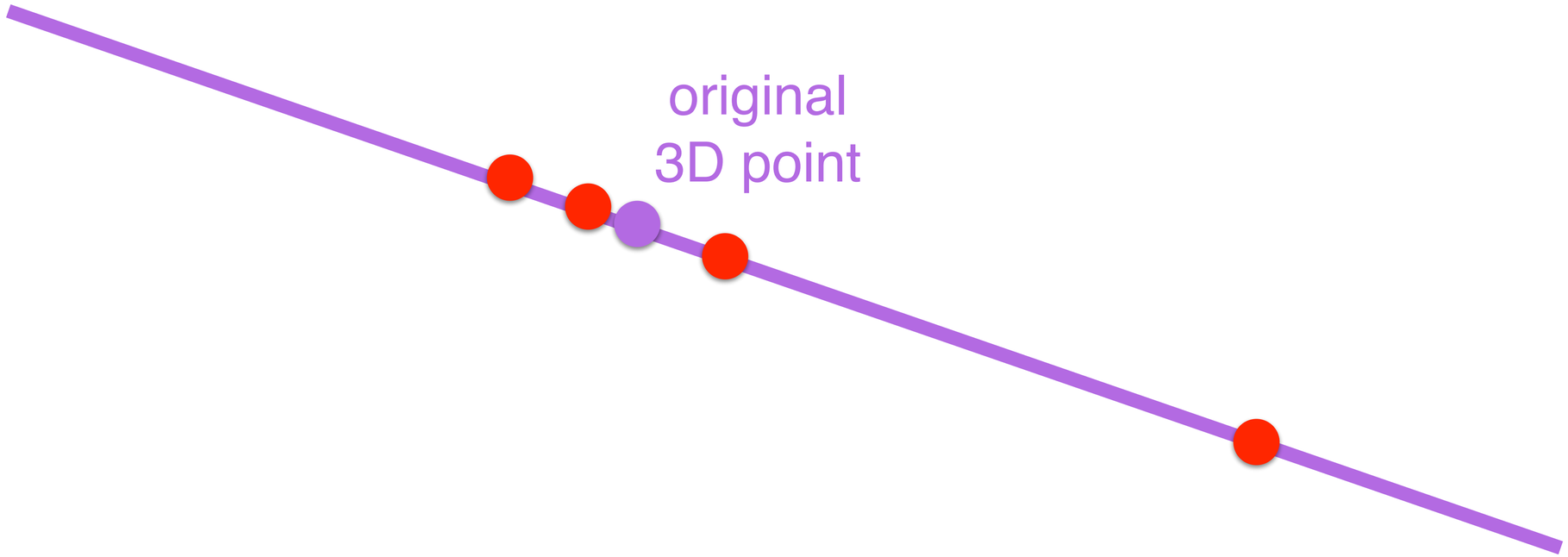

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# N Line Case




[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# N Line Case

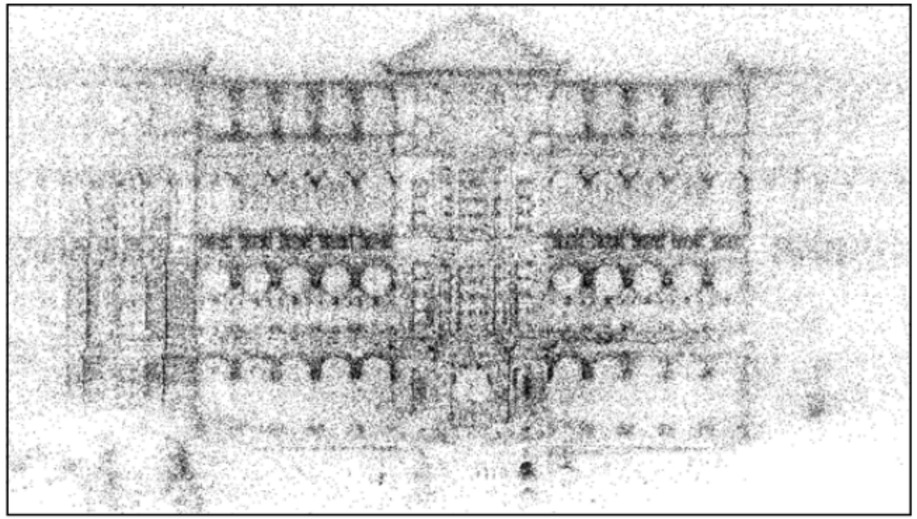



Find cluster(s) of closest points to approximate original point

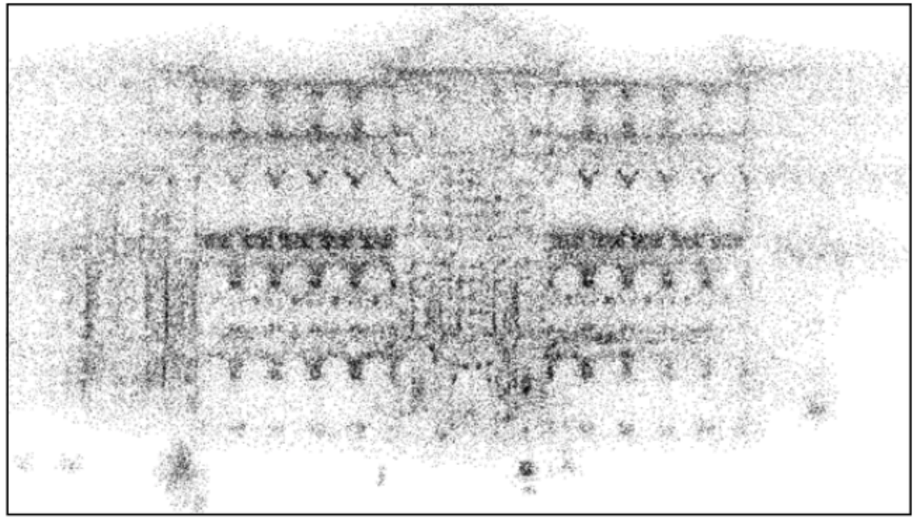
[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Results

Original Points



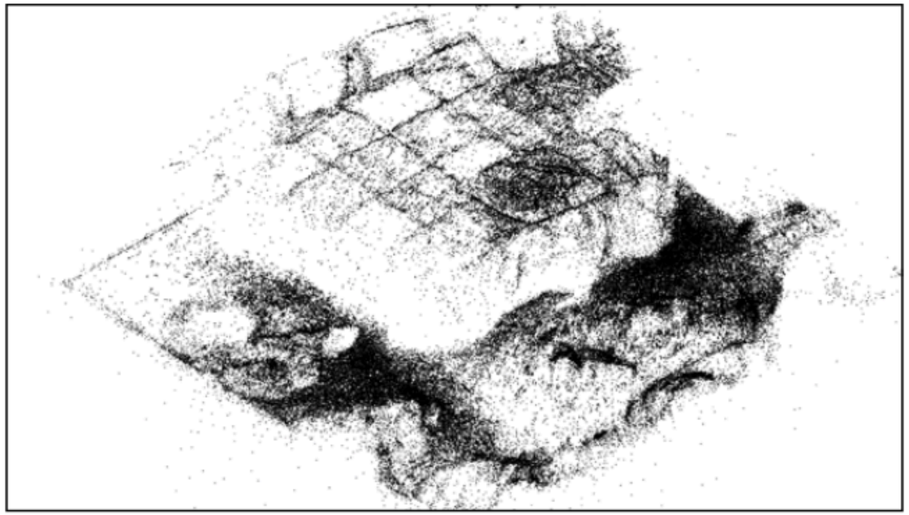
Recovered



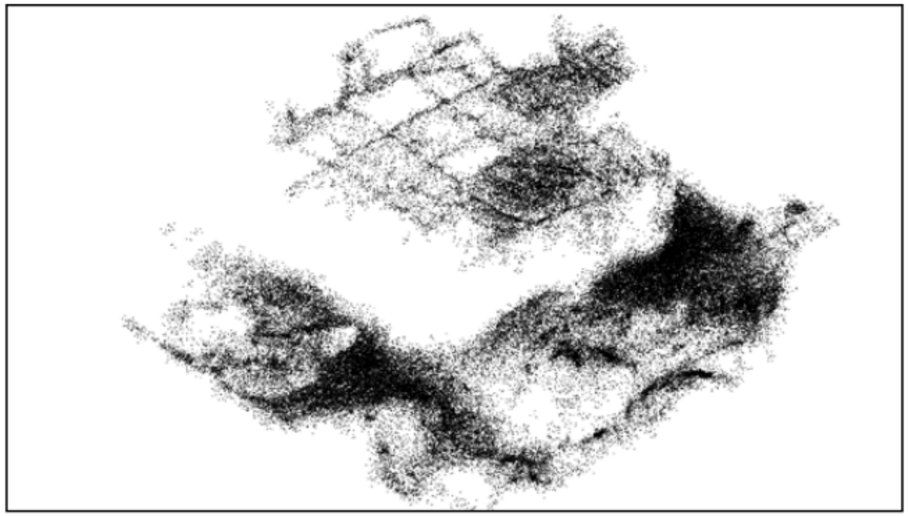
[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Results

Original Points

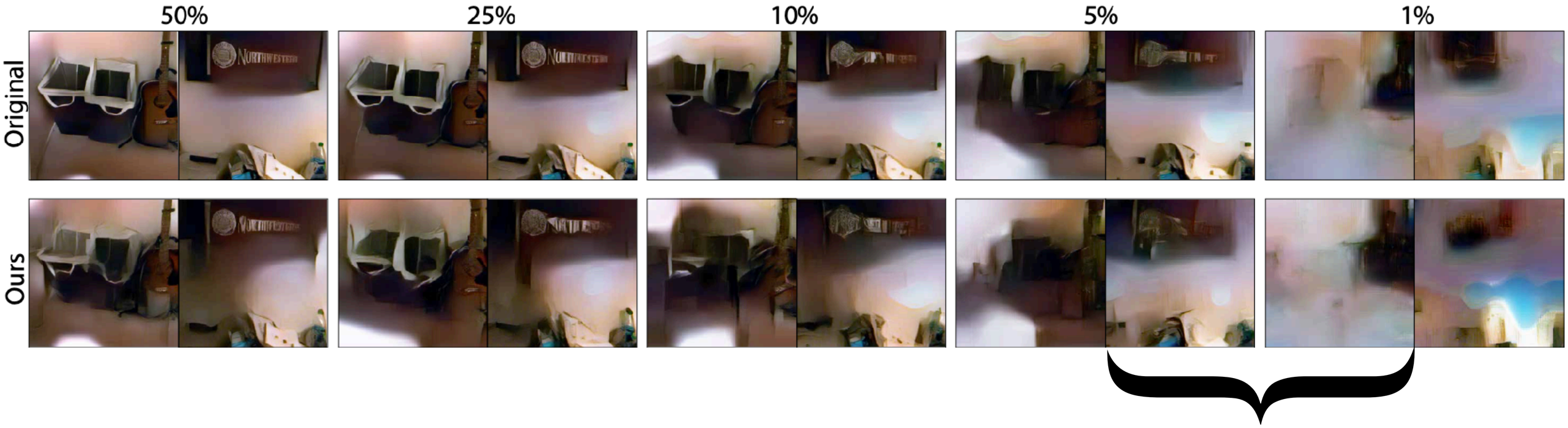


Recovered



[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# The Importance of Sparsity




localization still possible!

Recovering points is harder when there are few lines
(but sparse point clouds are already quite safe)

[Chelani, Kahl, Sattler, How Privacy-Preserving are Line Clouds? Recovering Scene Details from 3D Lines, CVPR 2021]

# Privacy-Preserving Localization

- Very young sub-field
- A lot of interesting open questions:
  - How to define privacy-preserving 3D models?
    - Are there line distributions that preserve privacy?
  - How to ensure that the AR cloud obtains as little information as possible from the client?
  - How to measure privacy?

# Main Takeaways

- Visual Localization is an interesting and important problem
- Dominant approach (use CNN to solve problem) does not work out of box
- **Geometric reasoning taught in this course still relevant!**
- Overview over different approaches to localization
- Long-term localization problem still far from being solved
- Privacy is only starting to be explored