

Large Scale City Modeling

Christoph Strecha
EPFL

Timo Pylvänäinen (Nokia)
Engin Tola (EPFL)
Alexander Bronstein (Israel)
Michael Bronstein (Israel)
Pascal Fua (EPFL)

Large Scale City Modeling

- Aerial / satellite Imagery
 - wide coverage with few images
 - only bird's-eye view
 - precise GPS and IMU
 - excellent image quality
 - Lidar is often captured

www.helimap.ch



Large Scale City Modeling

- Aerial / satellite imagery
- Systematic ground imagery
 - e.g. Google street view, NavTeq (Nokia)
 - good spatial coverage at a single time instance
 - precise GPS and IMU
 - good image quality



NavTeq (Nokia) panorama image of Chicago

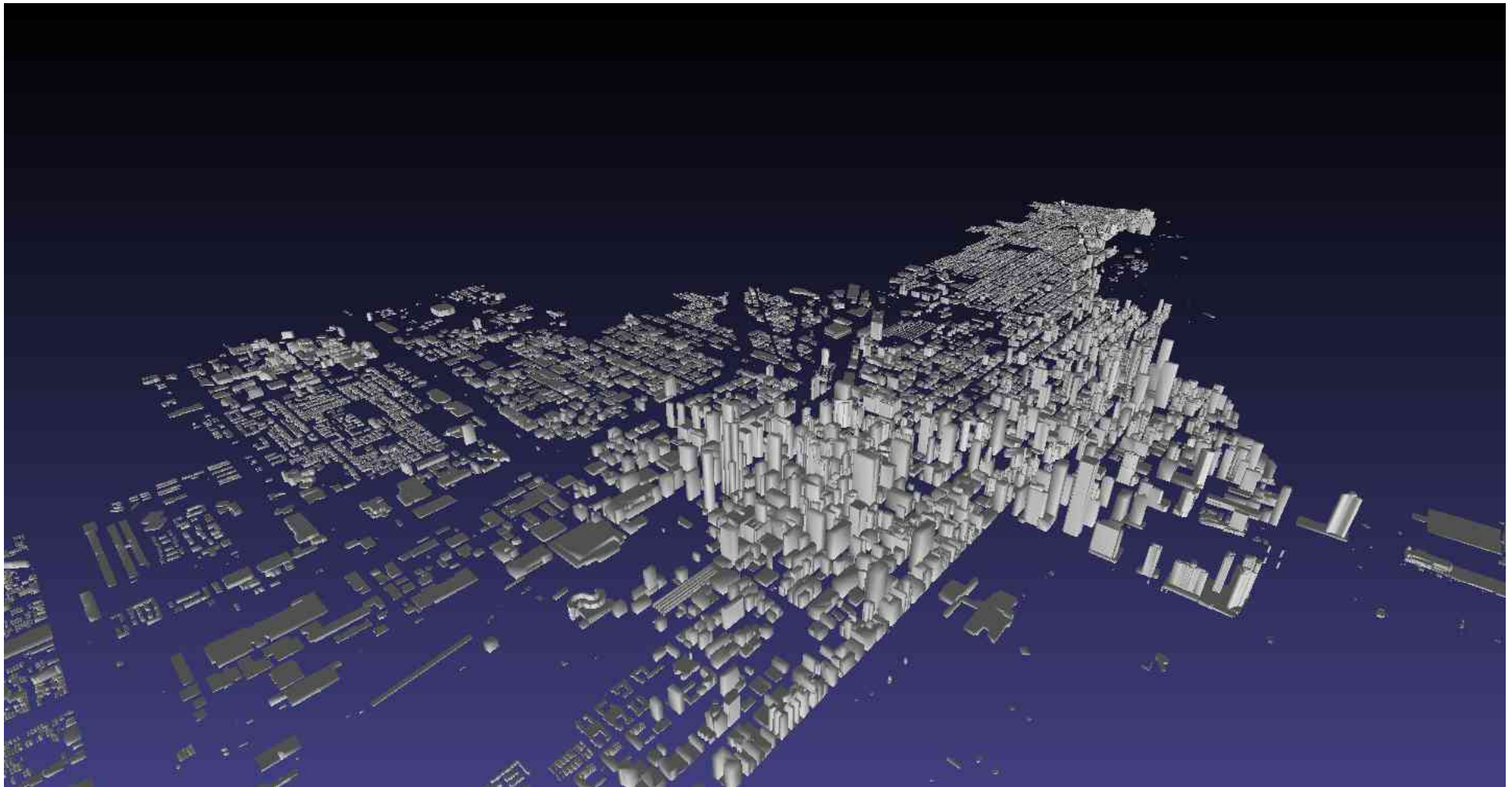


Google

Large Scale City Modeling

- Aerial / satellite imagery
- Systematic ground imagery

Manual building model of Chicago
Goal: automatic refinement and update



Large Scale City Modeling

- Aerial / satellite imagery
- Systematic ground imagery
- Low cost unmanned aircraft systems (UAS)

low image quality

low quality external sensors

good spatial coverage

large temporal coverage possible

low cost system

high demand for computer vision /
photogrammetry

video

www.sensefly.com



Large Scale City Modeling

- Aerial / satellite imagery
- Systematic ground imagery
- Low cost unmanned aircraft systems (UAS)
- Unsystematic Imagery
 - e.g. photo community collections
 - e.g. smart phone cameras (and their sensors)
 - lower quality of images and external sensors
 - low spatial coverage, high temporal coverage

flickr®



Large Scale City Modeling

- Aerial / satellite imagery
- Systematic ground imagery
- Low cost unmanned aircraft systems (UAS)
- Unsystematic Imagery



flickr®



Large Scale City Modeling

- Aerial / satellite imagery
- Systematic ground imagery
- Unmanned aircraft systems (UAS)
- Unsystematic Imagery



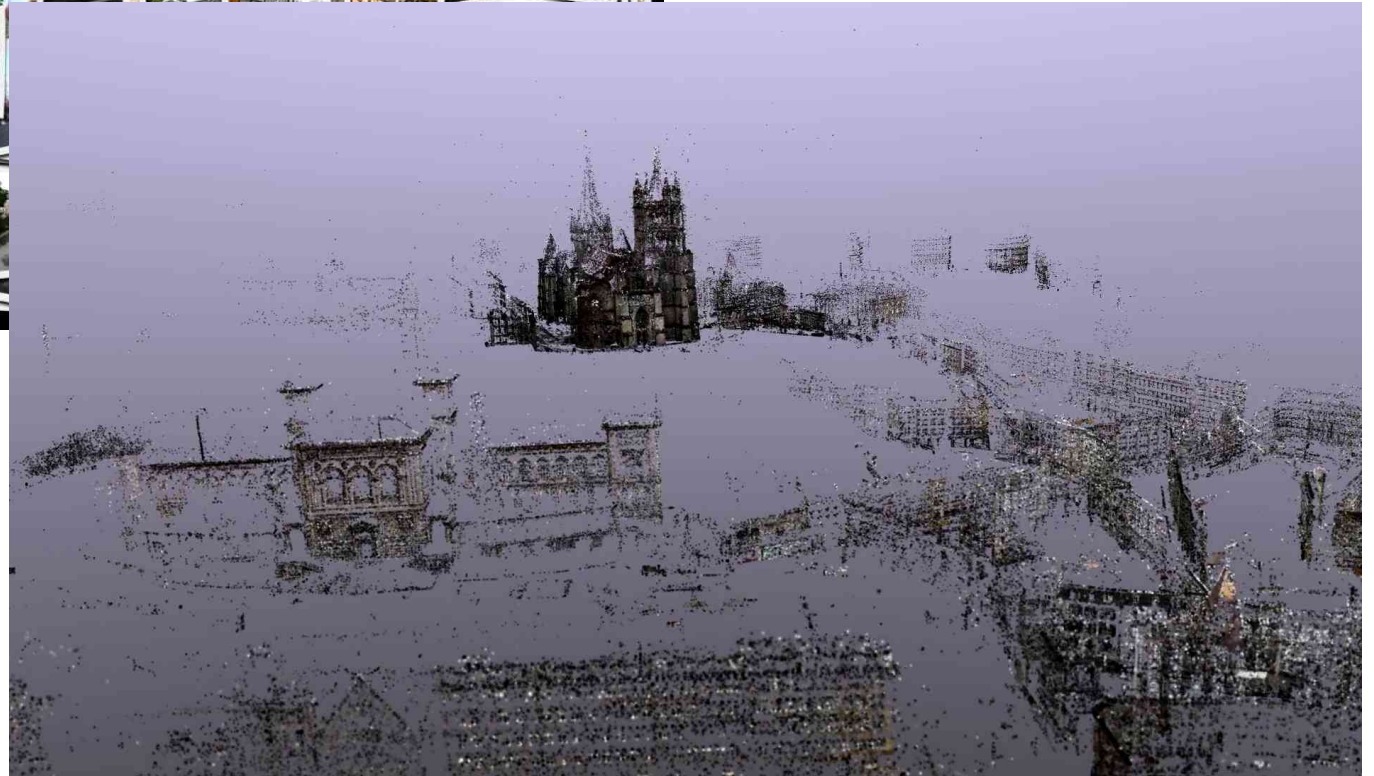
flickr®



Large Scale Terrestrial City Modeling

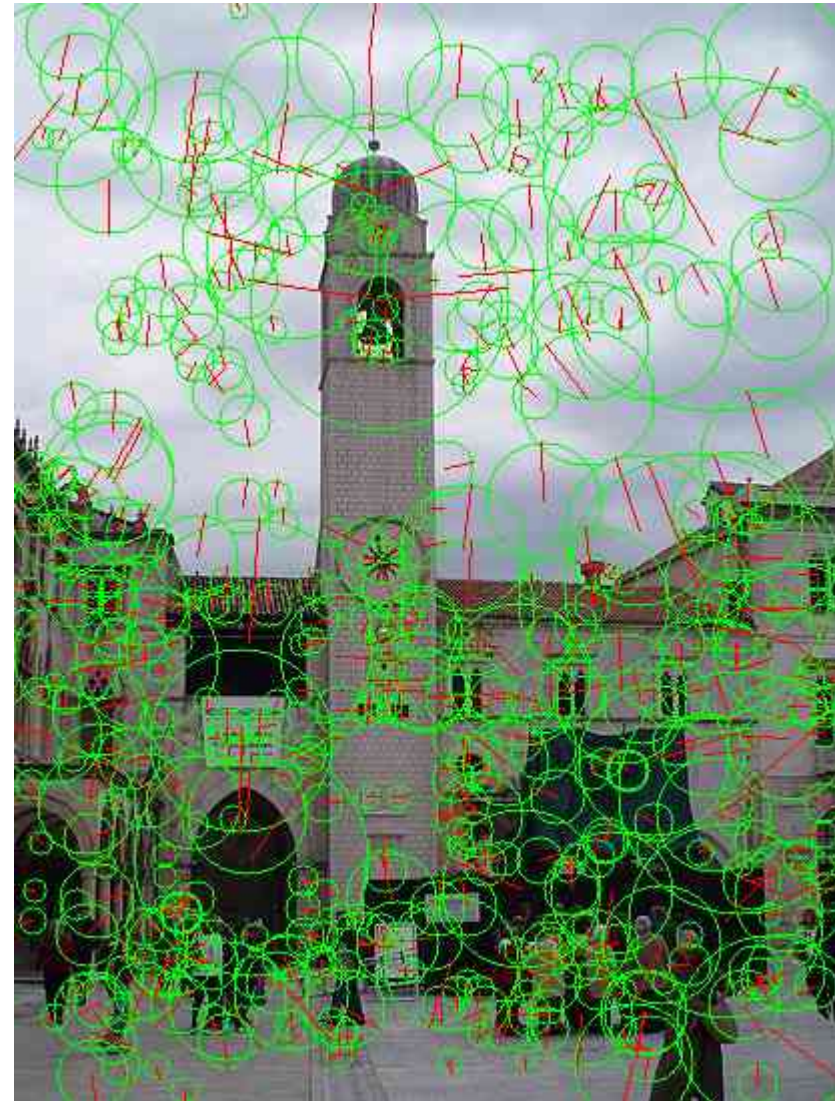
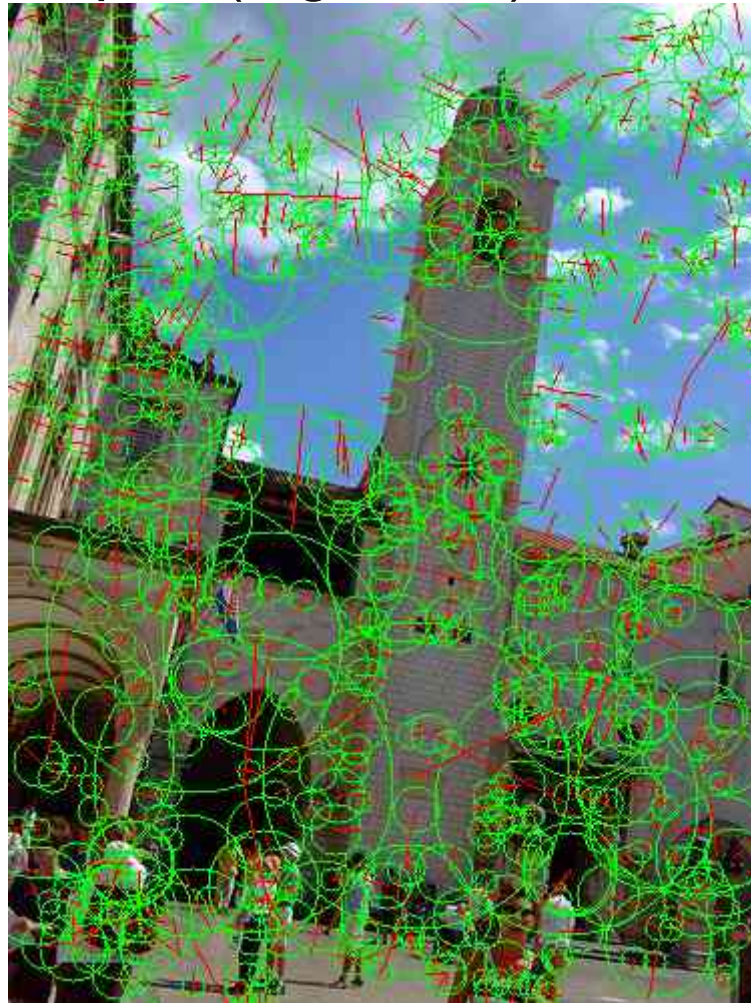


- huge collection of images
- Geo-tags / GPS might be available → inaccurate



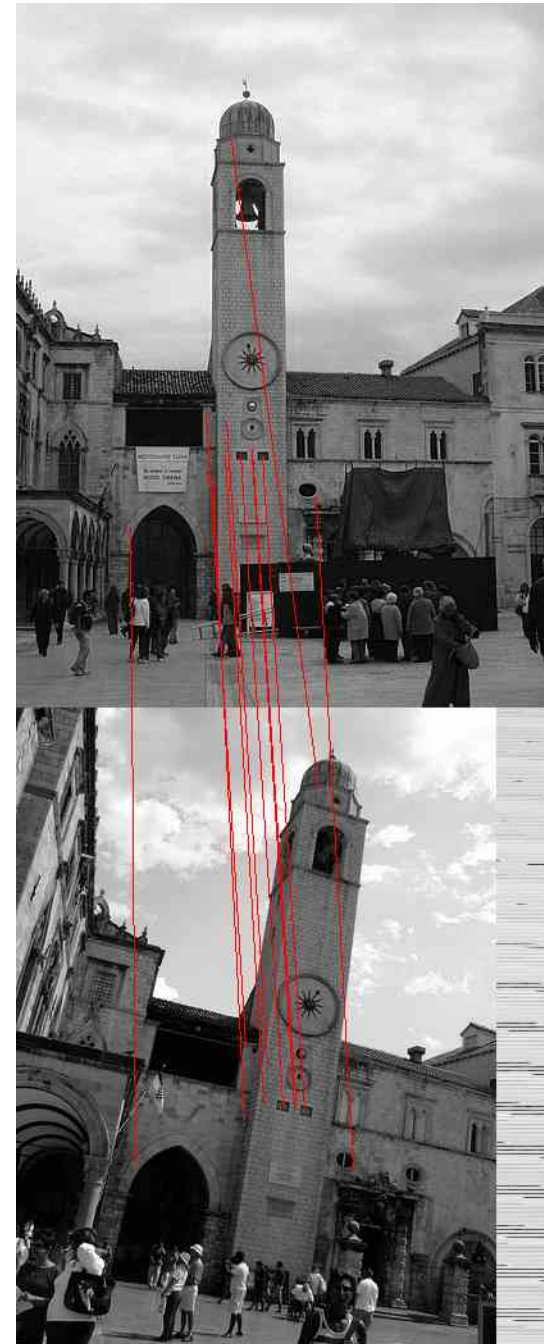
Pipeline

- invariant keypoint detection
- keypoint descriptor (e.g. SIFT)



Pipeline

- invariant keypoint detection
- keypoint descriptor (e.g. SIFT)
- keypoint matching



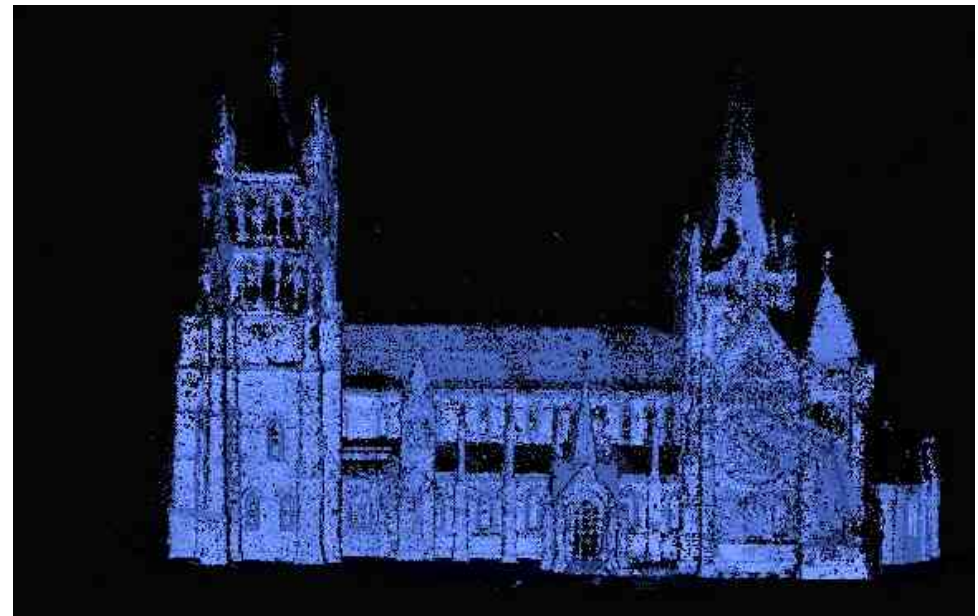
Pipeline

- invariant keypoint detection
- keypoint descriptor (e.g. SIFT)
- keypoint matching
- self-calibration
- bundle adjustment



Pipeline

- invariant keypoint detection
- keypoint descriptor (e.g. SIFT)
- keypoint matching
- self-calibration
- bundle adjustment
- dense matching
- dense model building



Scalability issues

- SIFT feature → 128 values → 512 bytes each
5..20 Mb for one image!!!
- matching all pairs of images → slow
- bundle adjustment → memory and speed issues
for city wide settings
- dense matching on many images → slow

Current approaches

- “Building Rome in one Day” Snavely et.al. ICCV 2009
massive parallelization on cloud computer
→ bandwidth is major problem
- “Building Rome on a Cloudless Day”, Frahm et al., ECCV 2010
single computer
→ massive parallelization by using multiple graphics cards

Scalability issues

- SIFT feature → 128 values → 512 bytes each
5..20 Mb for one image!!!
- matching all pairs of images → slow
- bundle adjustment → memory and speed issues
for city wide settings
- dense matching on many images → slow

Dynamic and Scalable Large Scale Image Reconstruction

C. Strecha (EPFL)
T. Pylvänäinen (NOKIA)
P. Fua (EPFL)

CVPR 2010

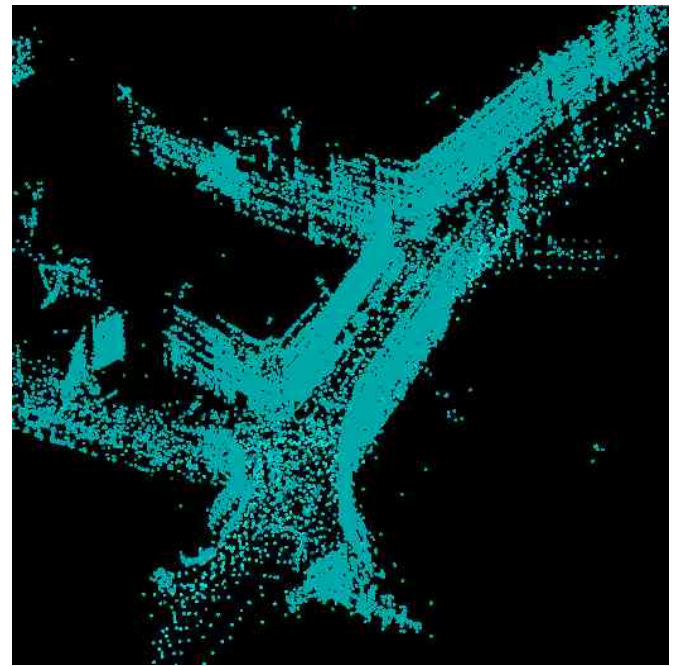
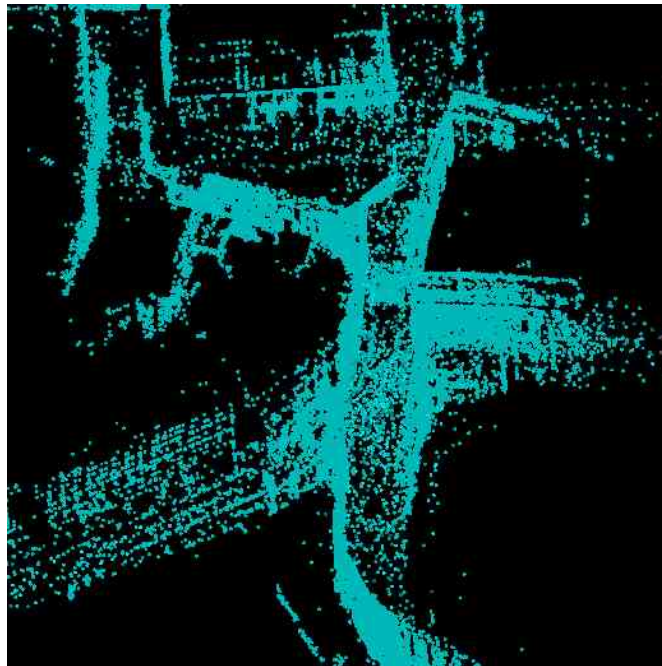
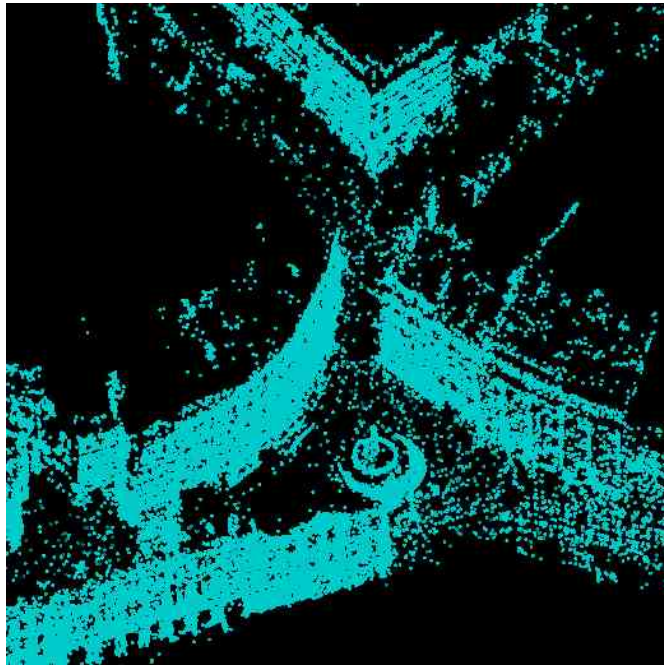
<http://cvlab.epfl.ch/research/calibration>

Current Issues for Large scale reconstructions

- “Building Rome ...” actually builds disconnected landmarks
- City wide reconstructions need images to connect landmarks
- Bundle Adjustment does not scale to entire city
largest reported Bundle adjustment by Snavely et.al.
San Marco Square with 14,079 images and 4,515,157 points
- Loosely connected images introduce drift in reconstruction
- Reconstructions are not geo-referenced
- Large calibration clusters are not efficient when images are added and bundled every time

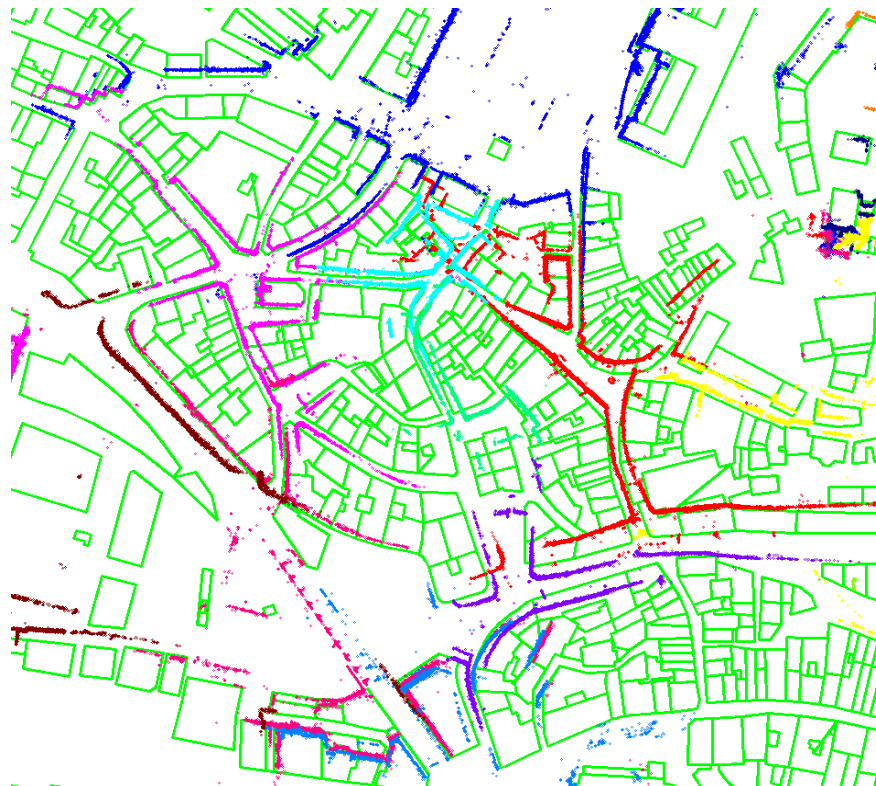
Our approach

- Keep clusters small such that bundle adjustment is still feasible
- New available images can be added efficiently to clusters



Our approach

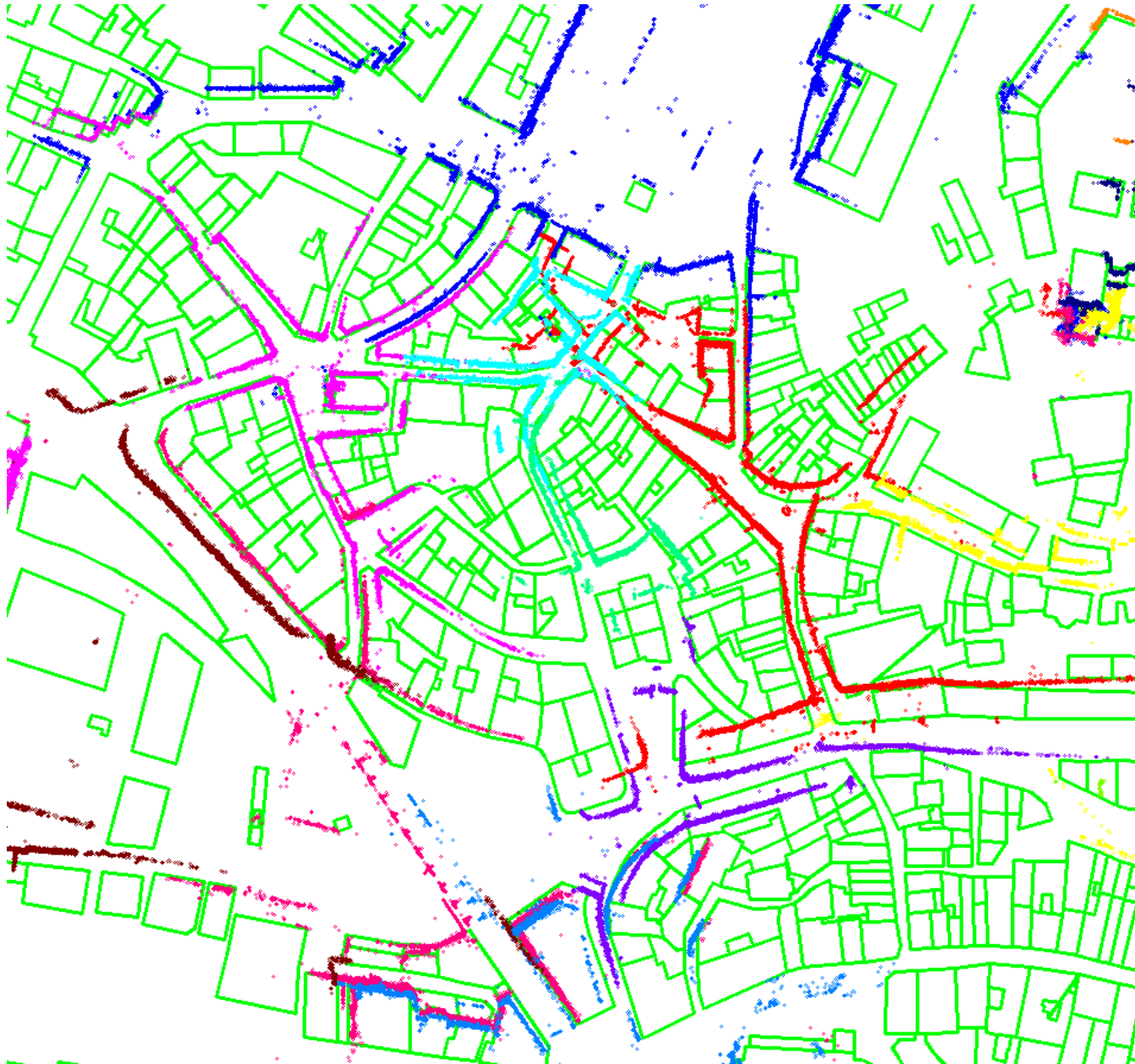
- global optimization brings all clusters into alignment by using
 - matches between clusters
 - gps / geo-tags that might be available for some images
 - building footprint model (publicly available – openstreetmap)
 - rough Digital Elevation model (DEM) publicly available



Our approach

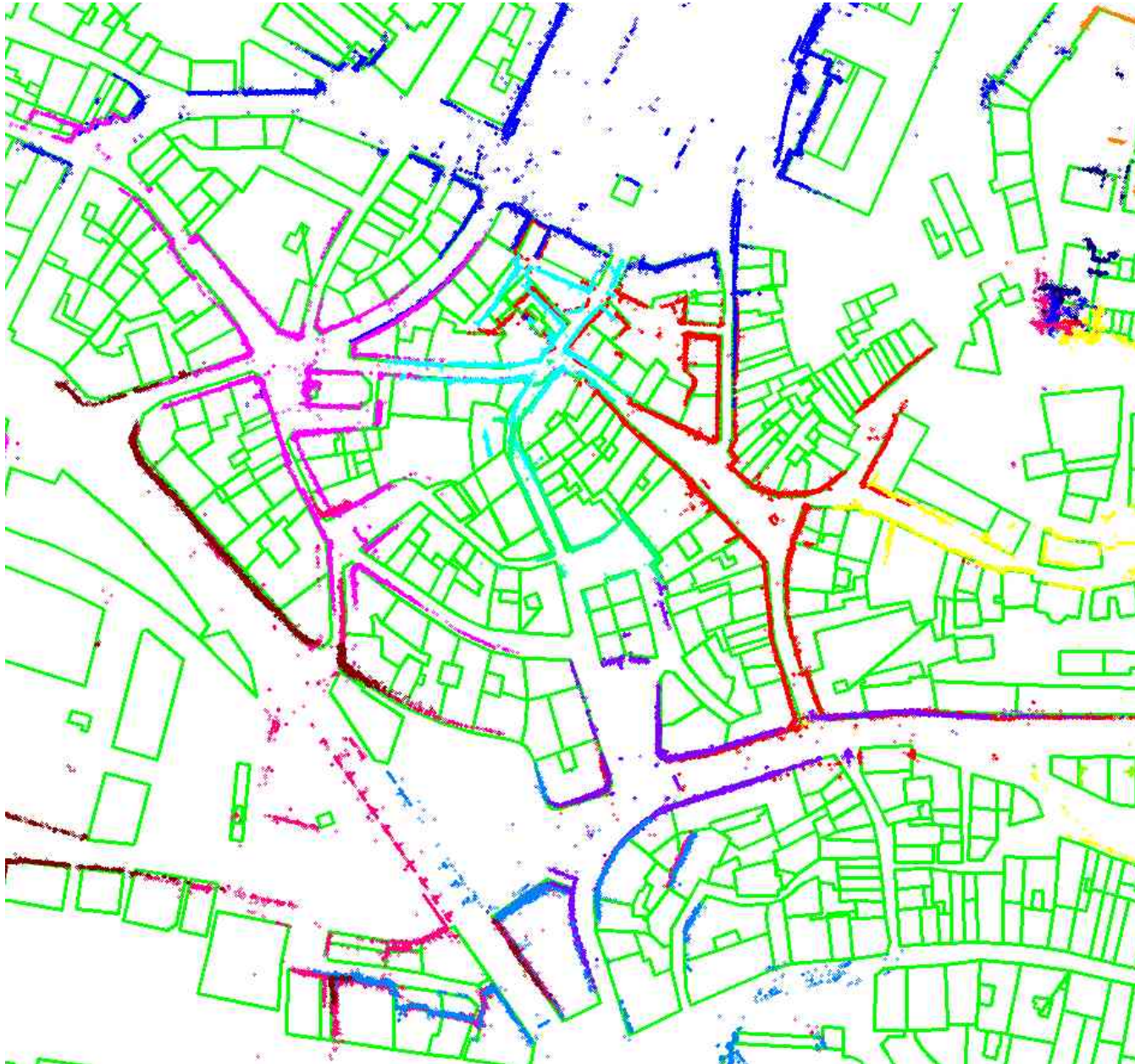
- Cluster alignment is fast – optimize only one rigid transformation for each cluster
- Does scale to whole city models
- Cluster alignment is possible for totally unconnected clusters
- Yields geo-referenced reconstructions
- New images can be added easily

Rough alignment



- Based on GPS or geo-tags using RANSAC
- Each color represents 3D-points of one cluster
- Building footprint model in green

Final alignment



- Based on robust integration of
- GPS or geo-tags
- Building model
- Cluster overlap
- Cluster matches
- DEM

Probabilistic Modeling

Each constraint:

contains outliers
has varying unknown accuracy

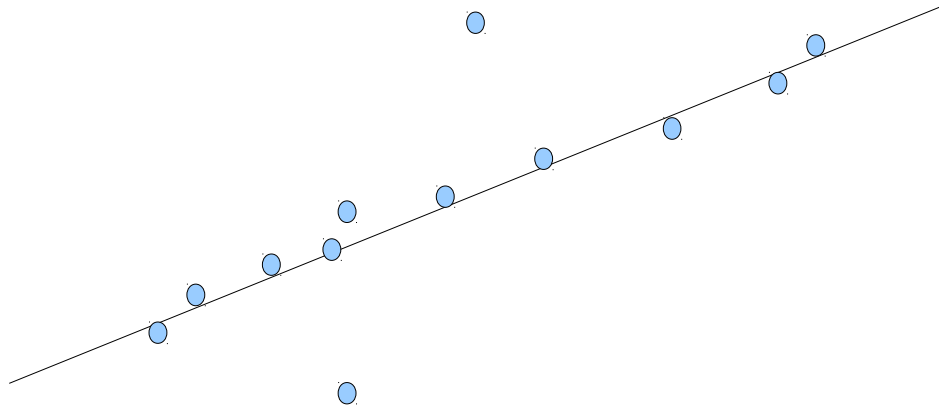
Each measurement has a hidden variable
assigned to it

$$p(y_i|\boldsymbol{\theta}) = \begin{cases} \mathcal{N}(y_i(\boldsymbol{\theta}); 0, \Sigma) & \text{if } x_i = 1 \\ g & \text{if } x_i = 0 \end{cases}$$

y measurement

x hidden variable labels outliers

Probabilistic Modeling



Example line fitting

y_i 2D points

$$\theta = \{a, b, \Sigma\}$$



Example GPS fitting

y_i 3D GPS coordinates

$$\theta = \{ \textit{rigid transformation}, \Sigma \}$$

$$p(y_i | \theta) = \begin{cases} \mathcal{N}(y_i(\theta); 0, \Sigma) & \text{if } x_i = 1 \\ g & \text{if } x_i = 0 \end{cases}$$

EM solution

Estimate: expected values of hidden variables given the current value of the parameters

Maximize: parameters using the expected values of x

Map Constraint



D

$$\theta = T_i, \Sigma_i$$

Parameters \rightarrow rigid transformation of each cluster

$$y_i = P T_j P_i$$

One measurement for each 3D point

P_i

3D point

T_j

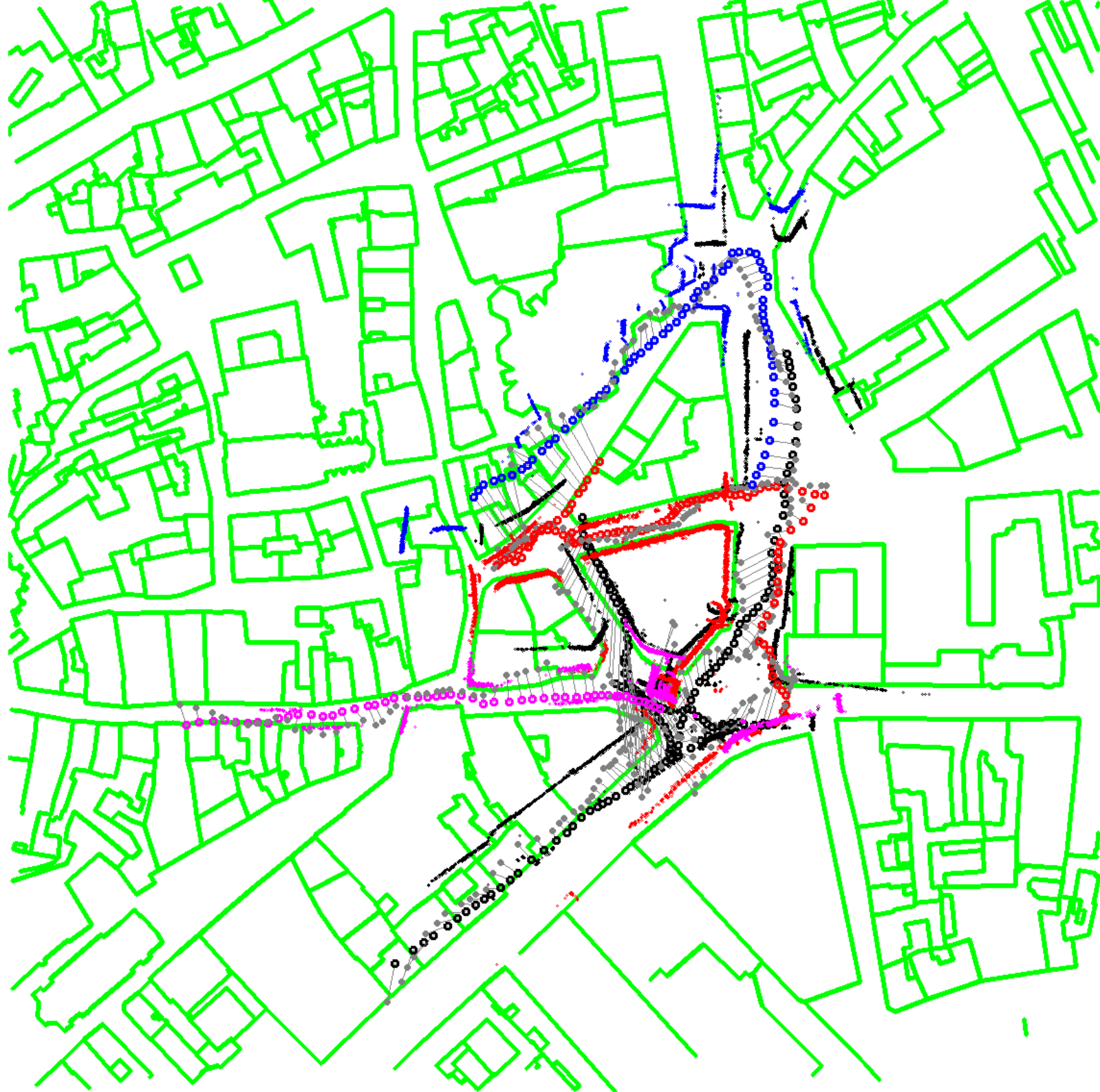
rigid transformation

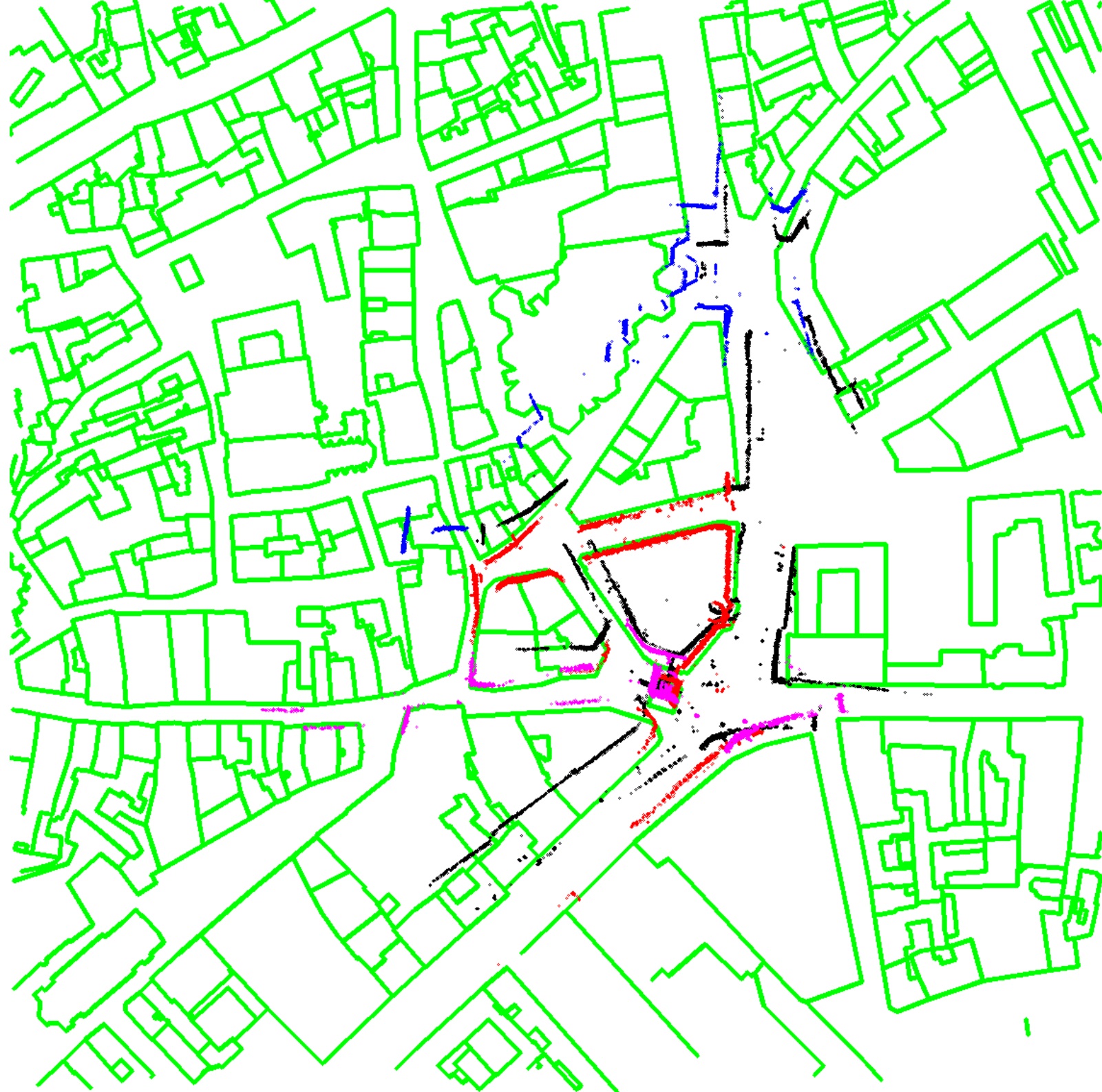
P

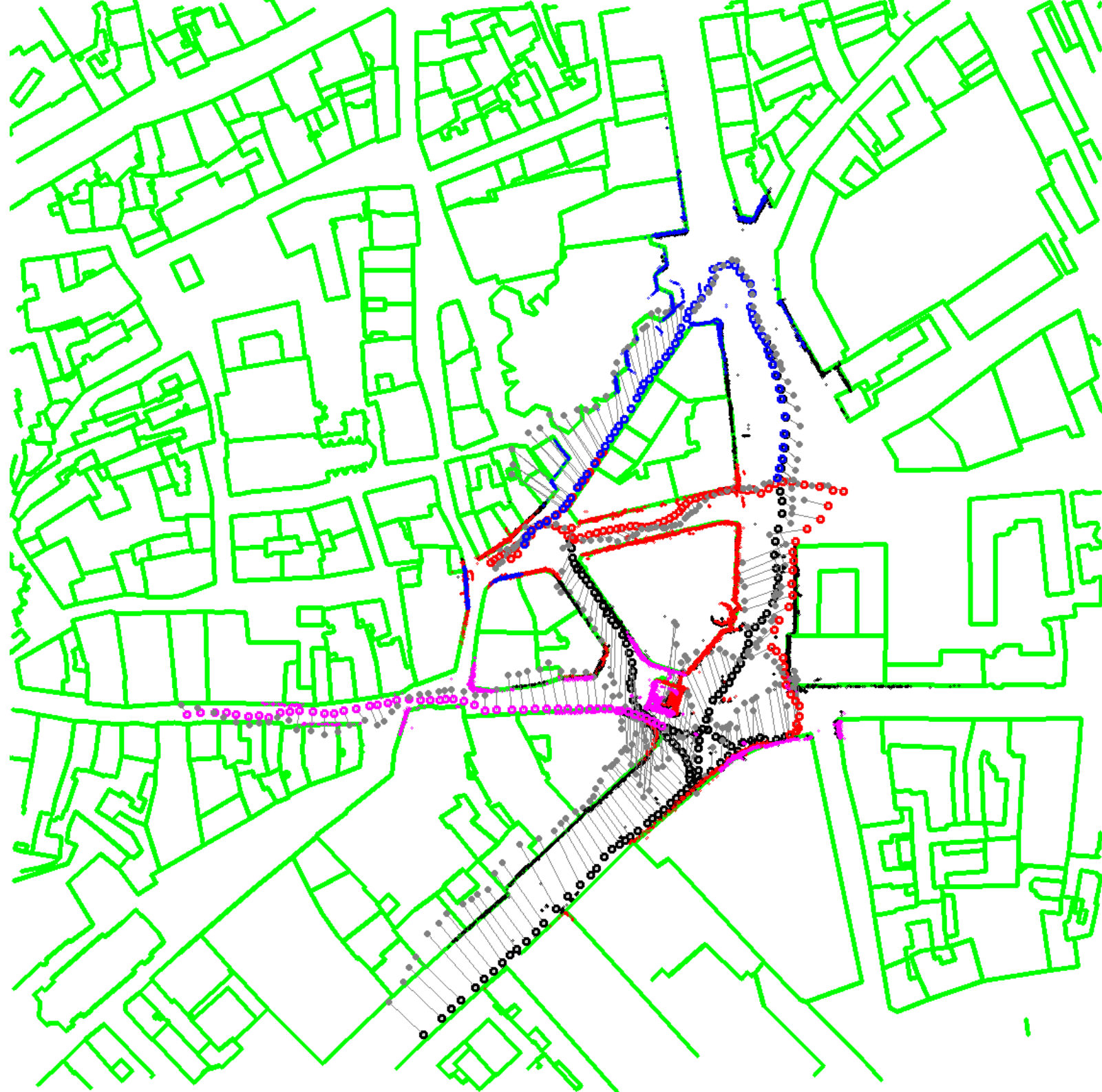
projection onto map: $\mathbb{R}^3 \rightarrow \mathbb{R}^2$

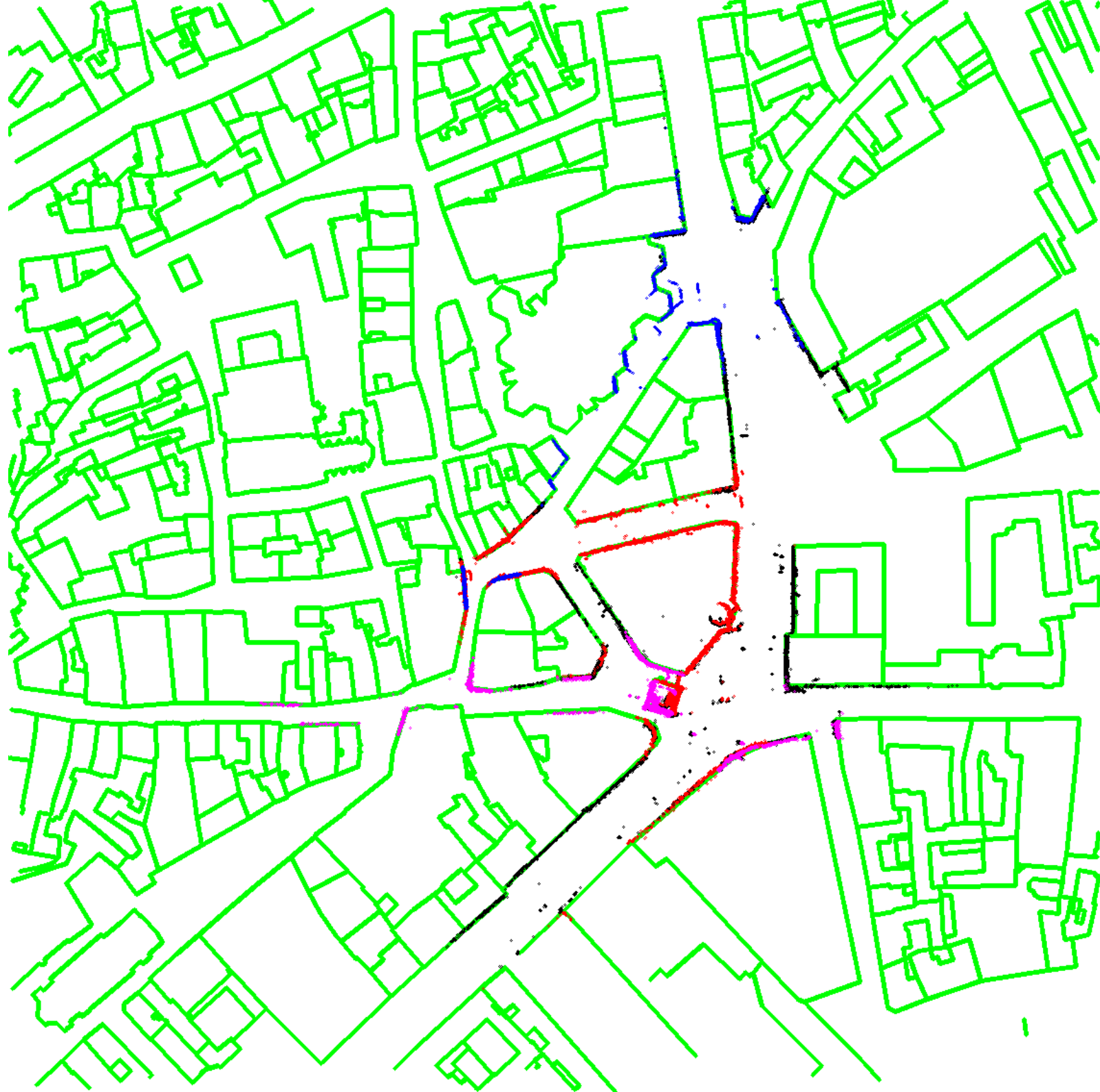
$$p(y_i | \theta) \sim \exp(-D(y_i, \theta))$$

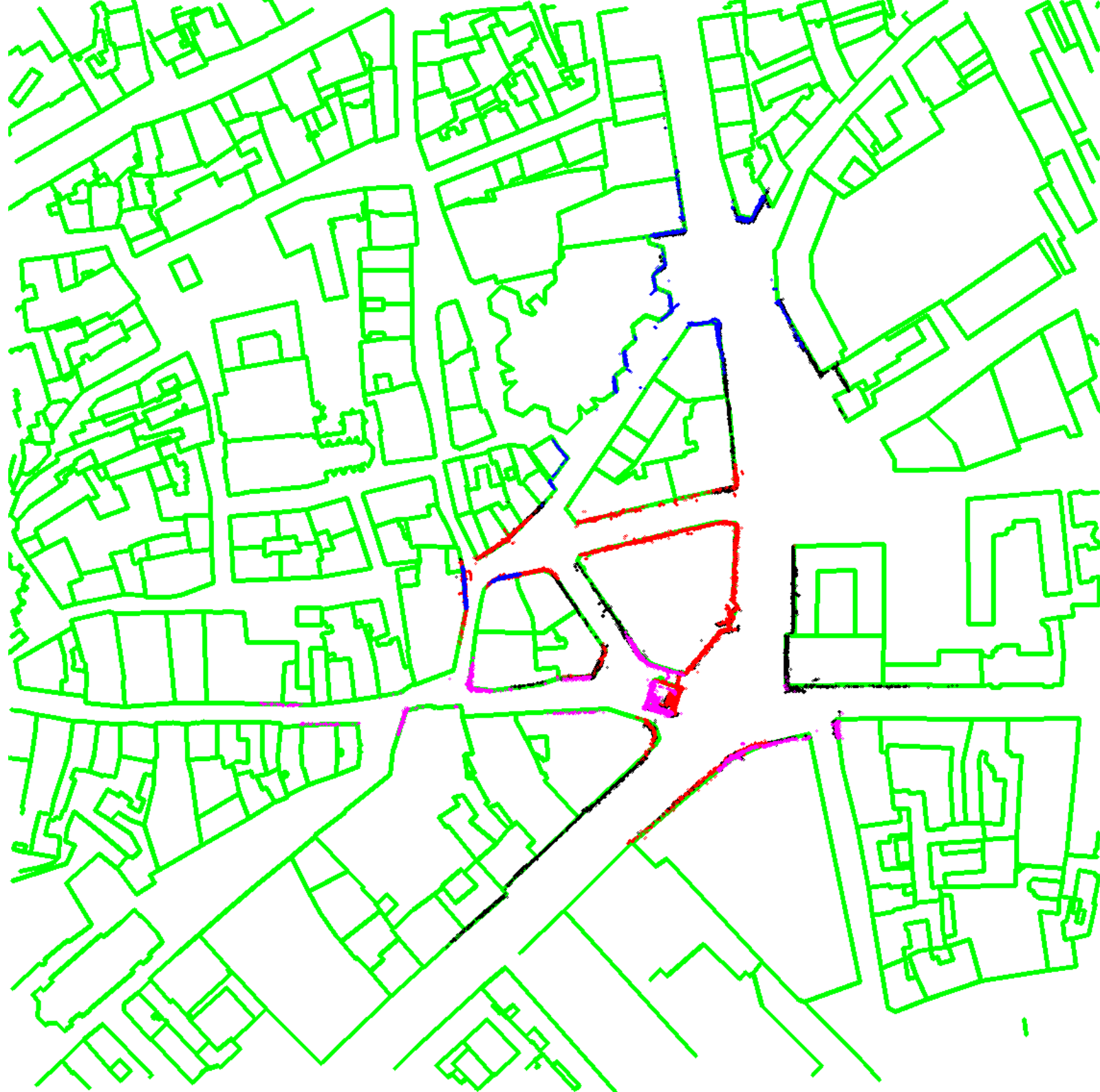
$$E_{map} \sim -\sum_i D(y_i)$$



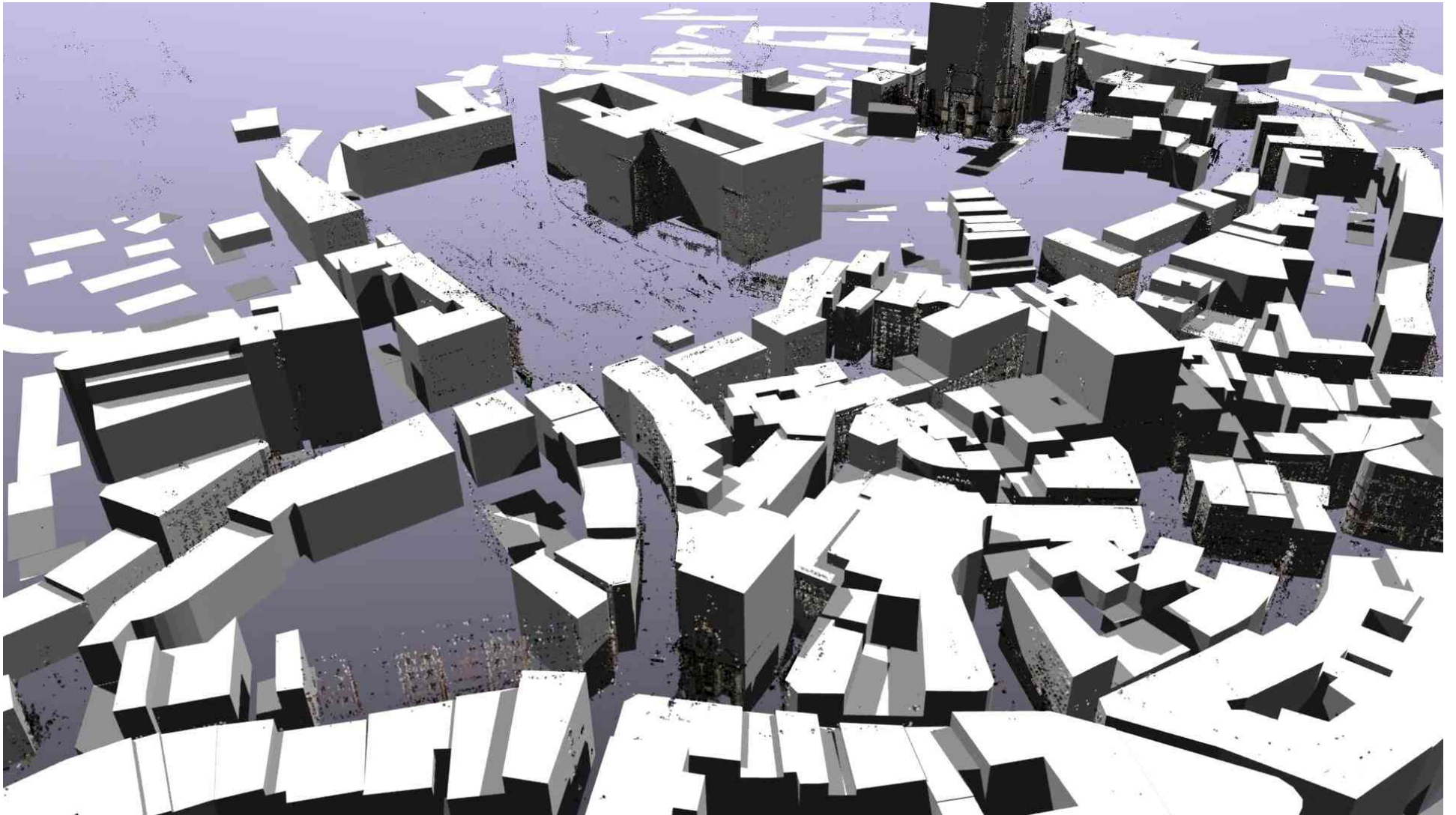






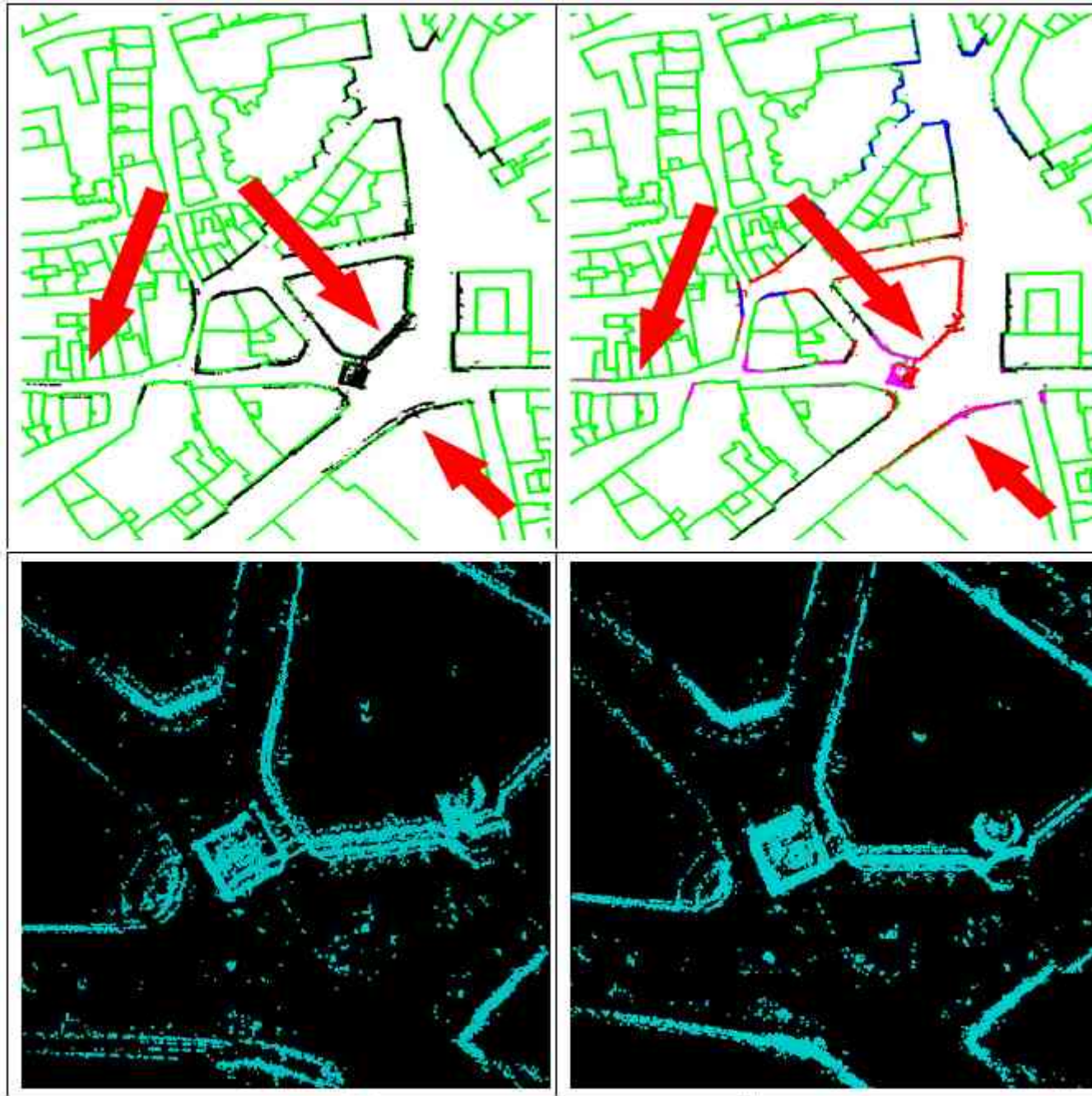


Results



[video](#)

Bundle adjustment vs cluster alignment



Experiment:

Four clusters with overlap
are merged to a single
reconstruction and bundled
→ some facades appear twice

Four clusters are aligned
using our approach
→ better reconstruction by
using building model

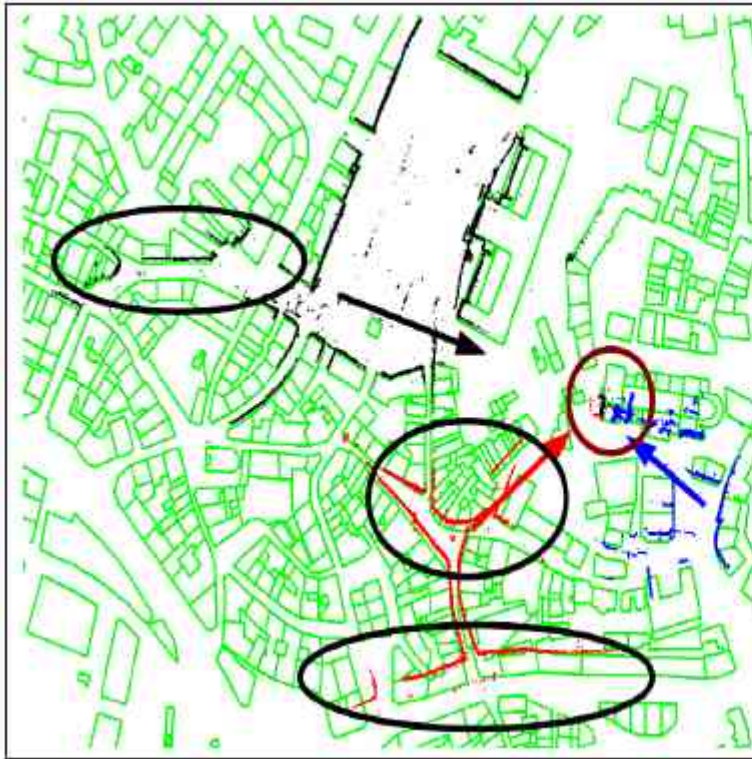
Alignment of non-overlapping clusters



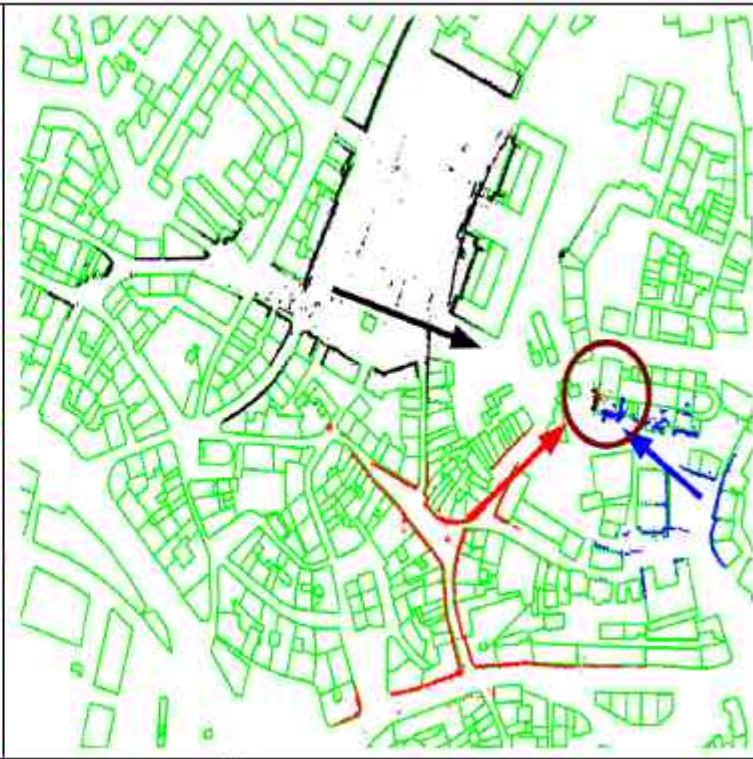
Example Lausanne:
cathedral is seen from west, south-west and east
matching is very difficult

Alignment of non-overlapping clusters

without
building
model



with
building
model



black



red



blue

Scalability issues

- SIFT feature → 128 values → 512 bytes each
5..20 Mb for one image!!!
- matching all pairs of images → slow
- bundle adjustment → memory and speed issues
for city wide settings
- dense matching on many images → slow

Efficient Large Scale MVS for Ultra High Resolution Images

E. Tola, C. Strecha and P. Fua

<http://cvlab.epfl.ch/research/emvs>

Dense Stereo Matching

History:

- NCC on window
- Smoothness priors
- Efficient optimization
- Large scale



Scalable Dense Stereo Matching

$$E(d) = \lambda \sum_i \rho(I_i(x) - I_2(x + d)) + \sum_{ij \in N} (\nabla d)^2$$

matching term
data term

smoothness term
prior term

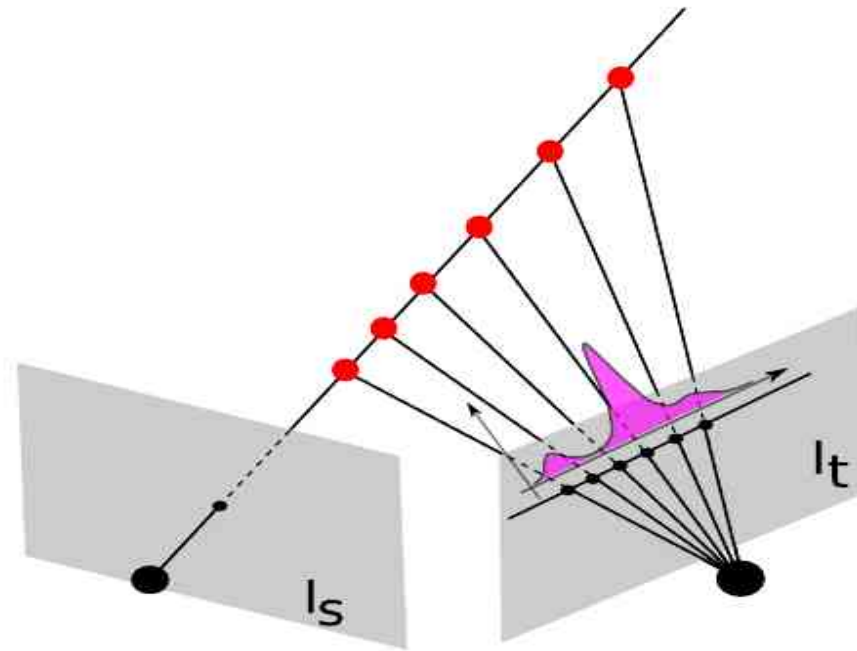
Needs:

- graph cuts
- belief propagation

For a **small** 640x480 image and with 100 depth states
30.720.000 nodes in the graph
infeasible for large (6-40 Mpixel) images

Scalable Dense Stereo Matching

$$E(d) = \lambda \sum_i \rho(I_i(x) - I_2(x+d)) + \sum_{ij \in N} (\nabla d)^2$$



Dense Stereo Matching

$$E(d) = \lambda \sum_i \rho(D_i(x) - D_2(x + d)) + \sum_{ij \in N} (\nabla d)^2$$

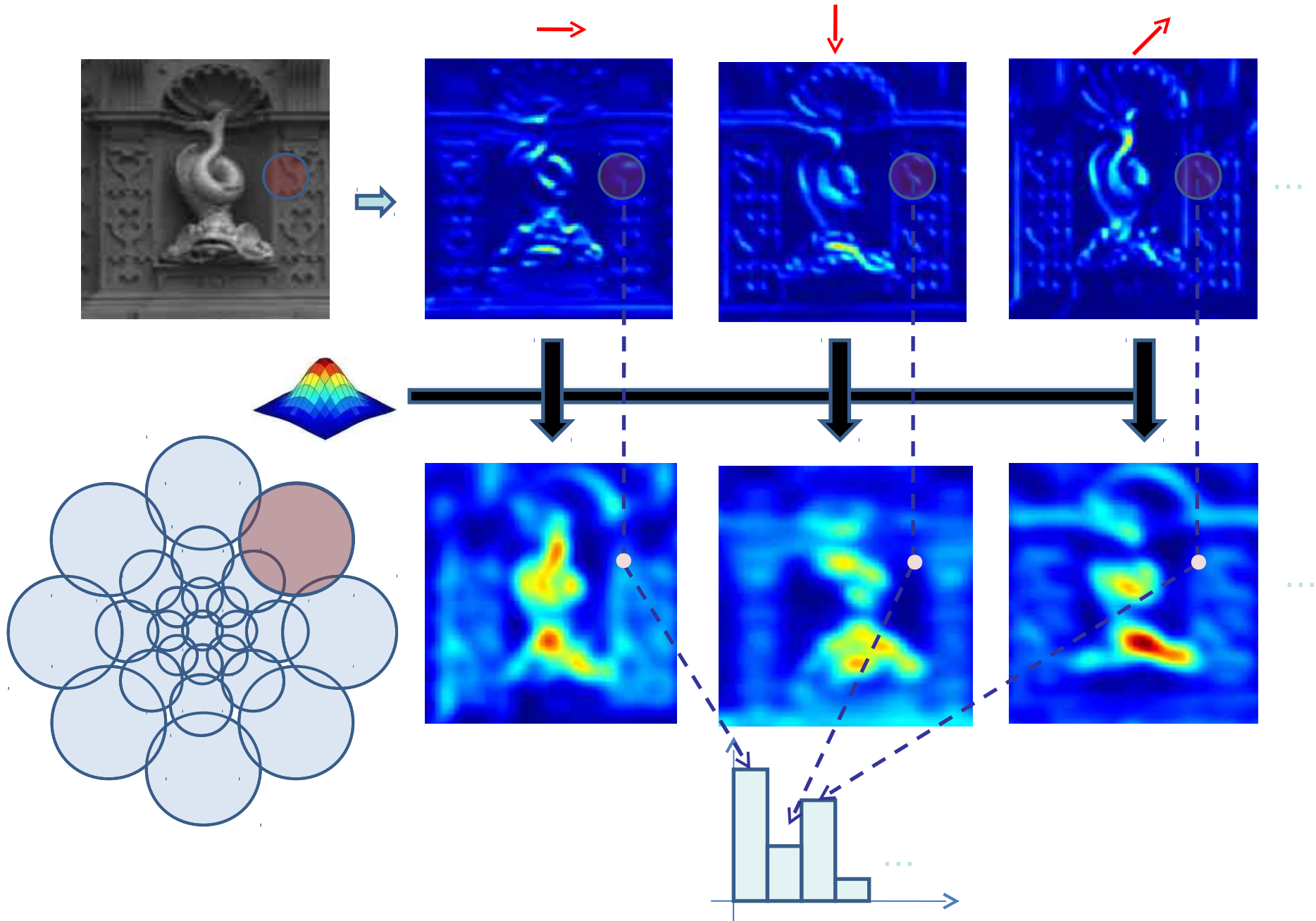
matching term
data term

Look for data term, such that ambiguities are minimal

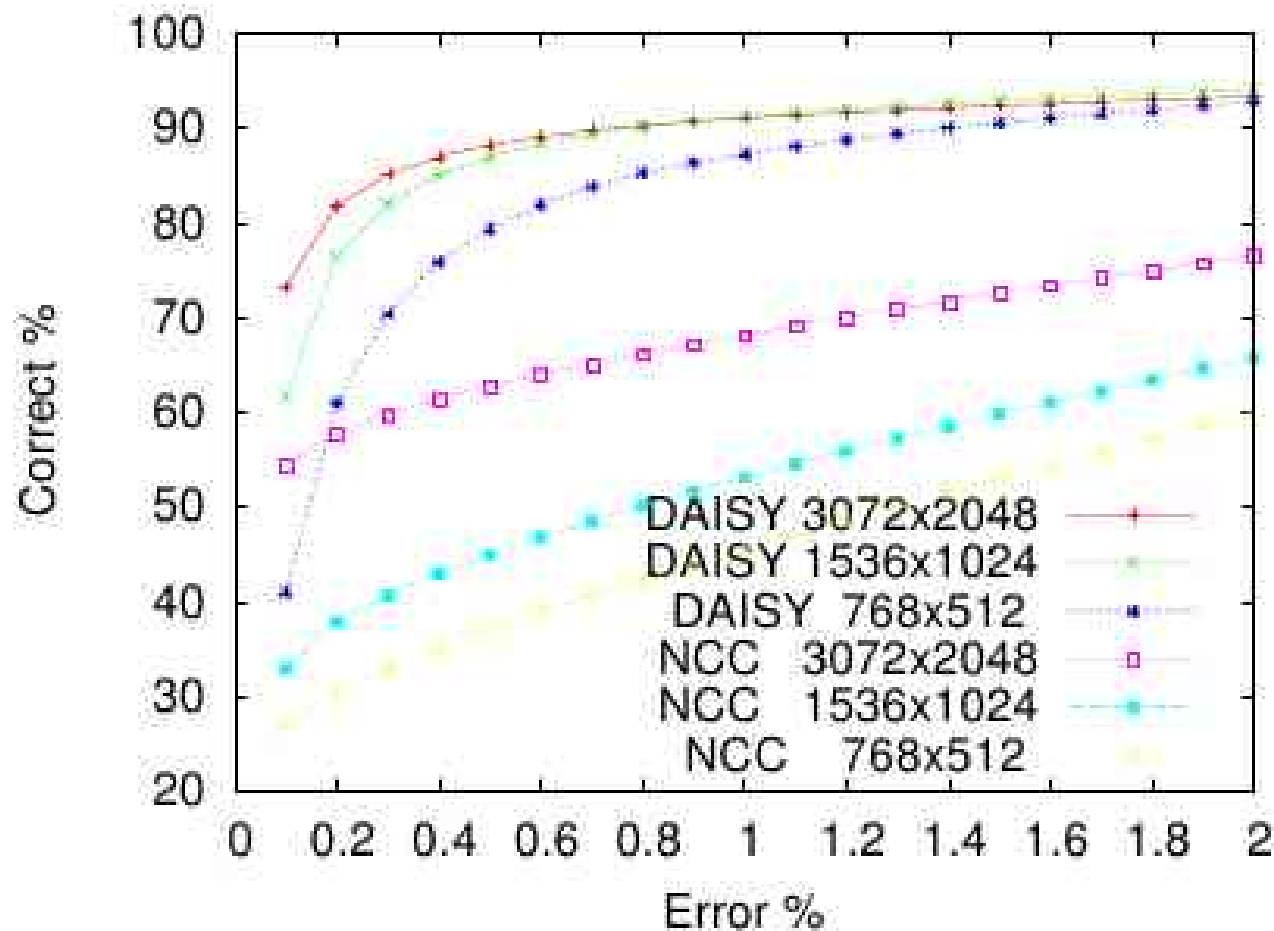
Transform each pixel into a descriptor image

DAISY

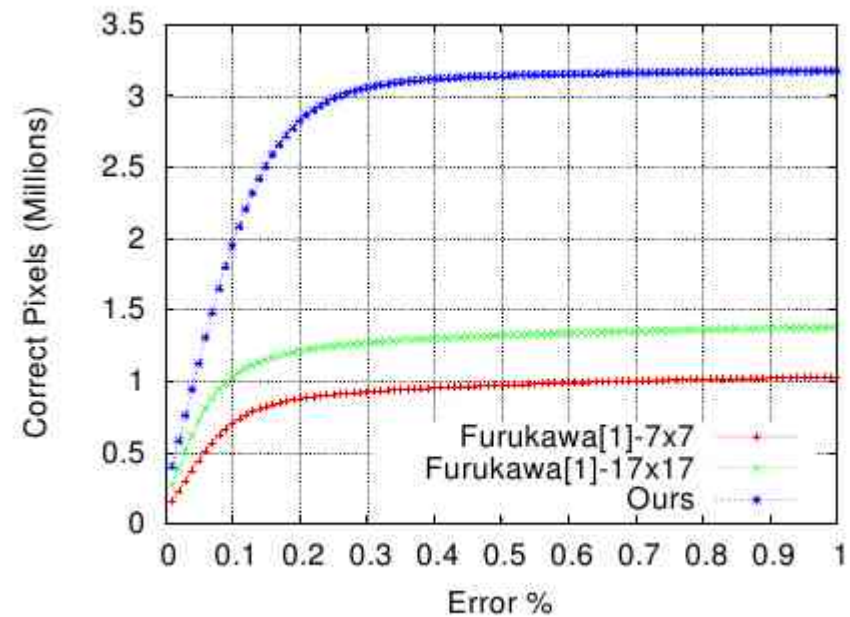
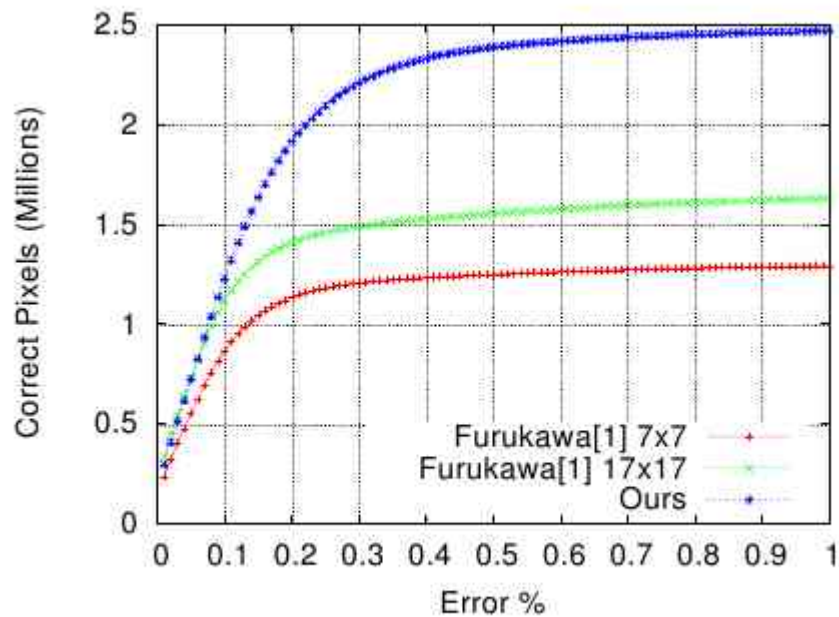
DAISY Computation



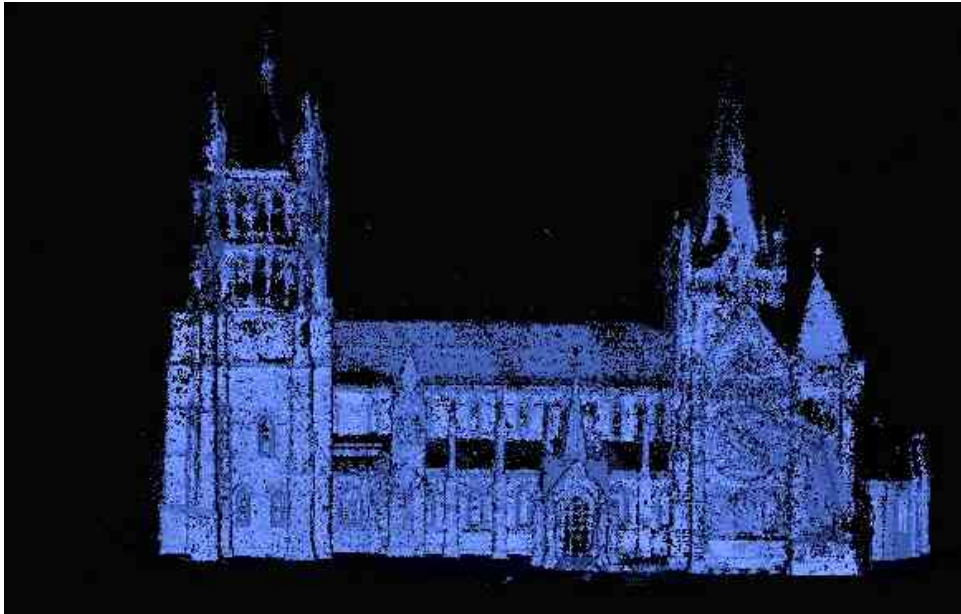
Ambiguities in Matching Daisy / NCC



Ambiguities in Matching Daisy / Furukawa



Results



Lausanne cathedral (ground)



Lausanne aerial

Results

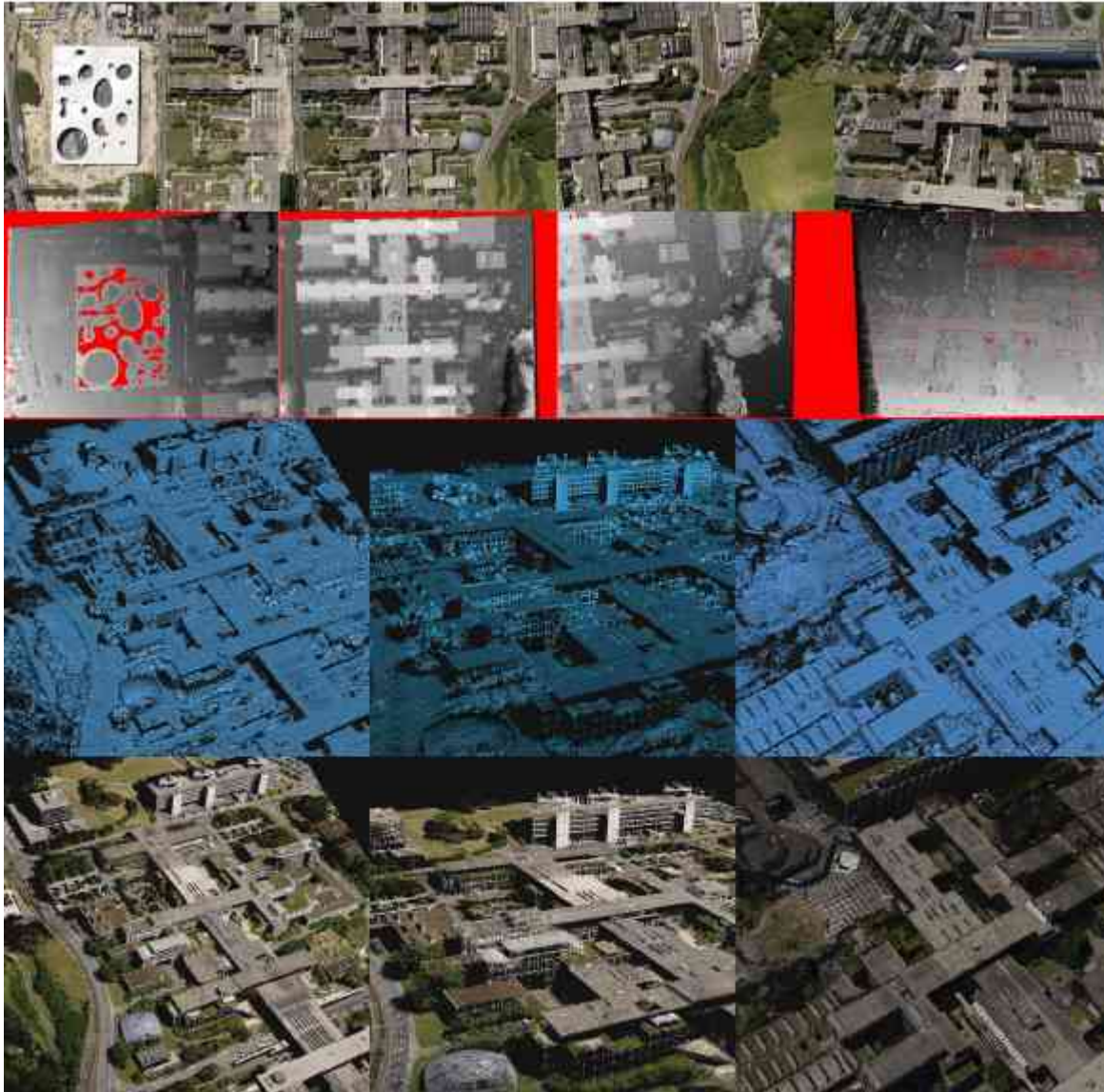


Lausanne cathedral (pillar)



Lausanne cathedral (statue)

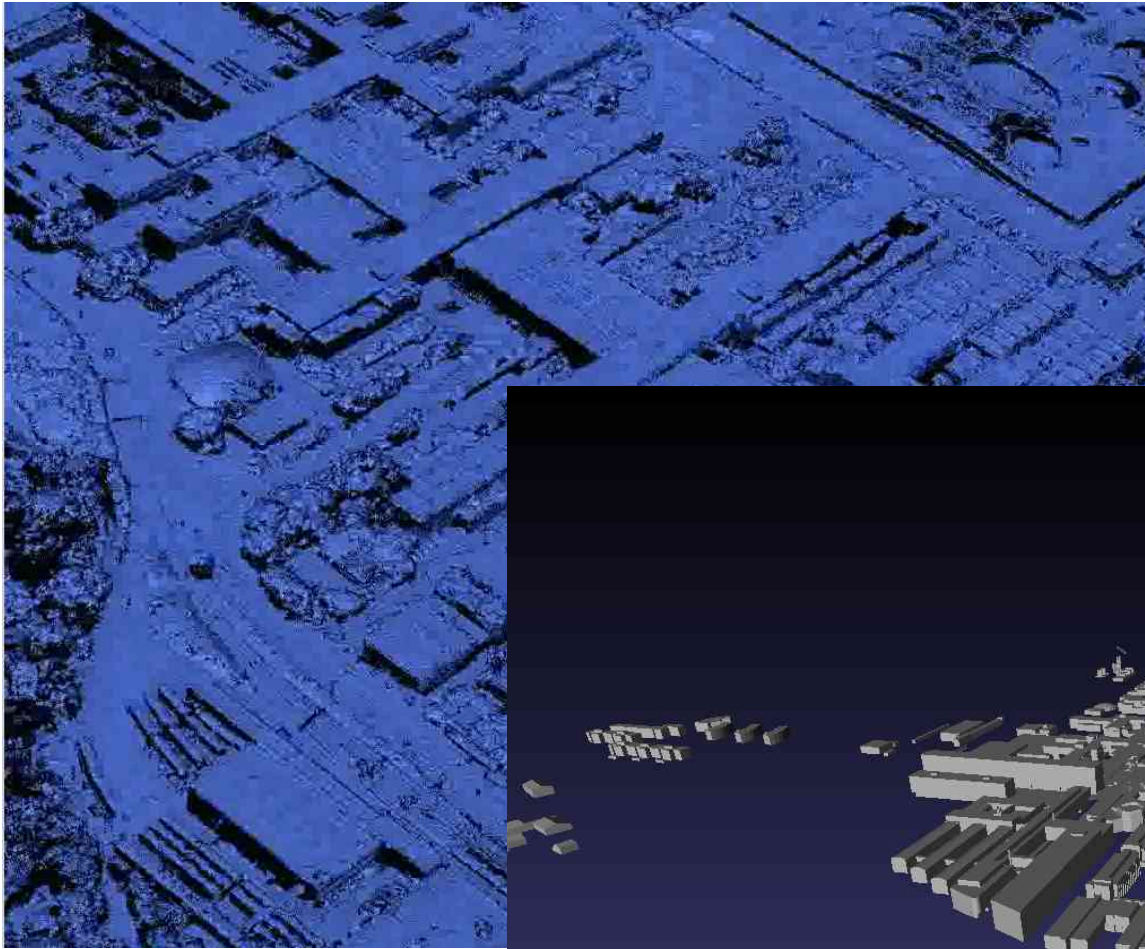
Results



video

Conclusions

- Scalability in calibration can be achieved
 - with additional information
 - building footprint model is important
- Efficient Stereo reconstruction on large images
 - without smoothness prior
 - turning image into descriptor image
such that matching ambiguities are reduced



Thank you!



Binary SIFT source code online:
<http://cvlab.epfl.ch/research/detect/ldahash>

Multi-view stereo evaluation:
<http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html>

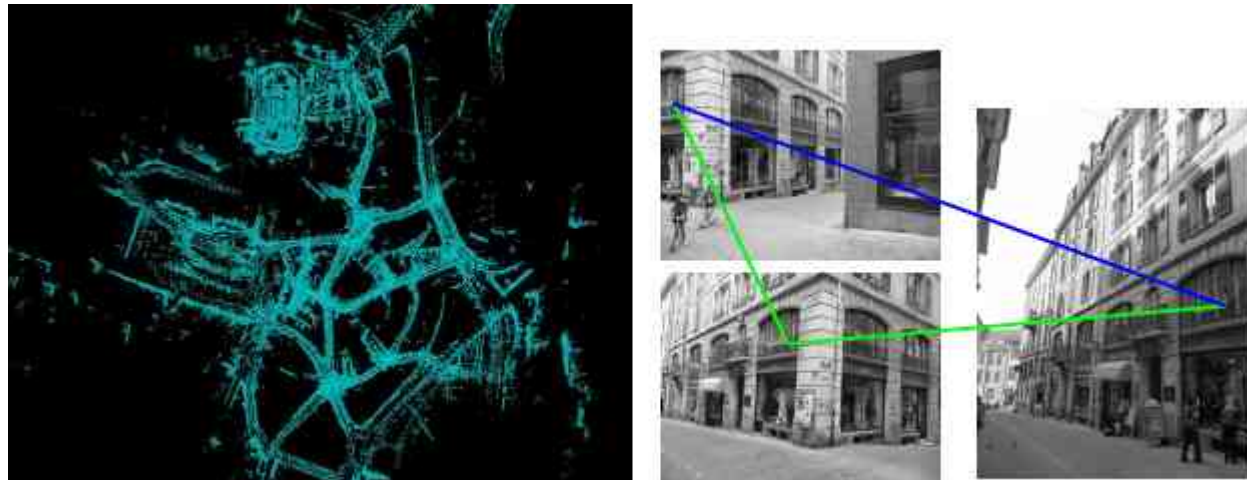
Outline

- SIFT feature → 128 values → 512 bytes each
5..20 Mb for one image!!!
- matching all pairs of images → quadratic → slow
- bundle adjustment → memory and speed issues
for city wide settings
- dense matching → quadratic → slow

SIFT-Hash better matching with smaller descriptors

Using training data:

- Transform SIFT descriptor into a compact bit-string
- Hamming distance yields better matching
- Fast matching using XOR and pop-count → single instruction



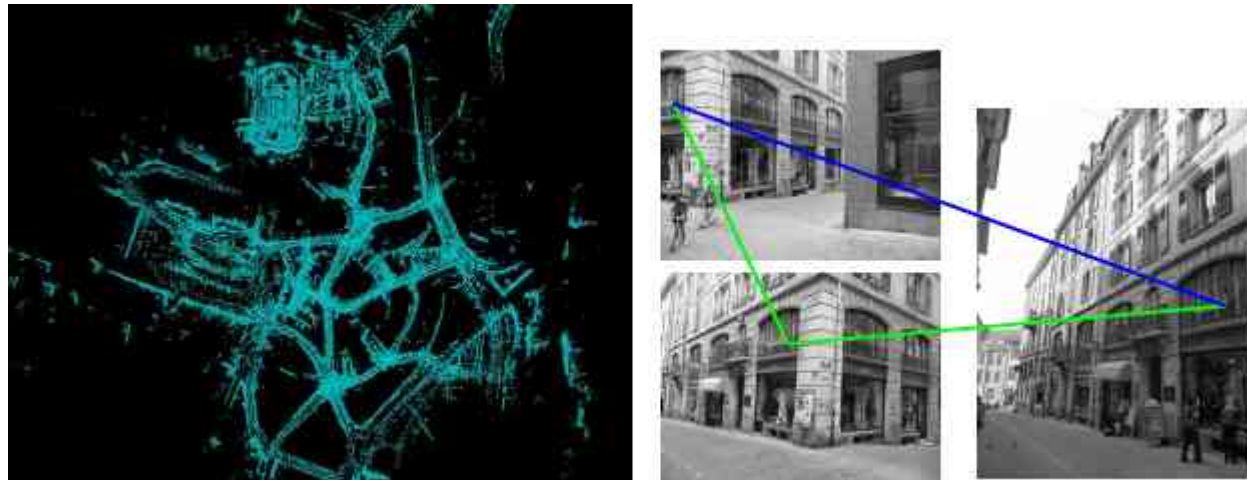
SIFT-Hash better matching with smaller descriptors

$$B = \text{sign}(P f + t)$$

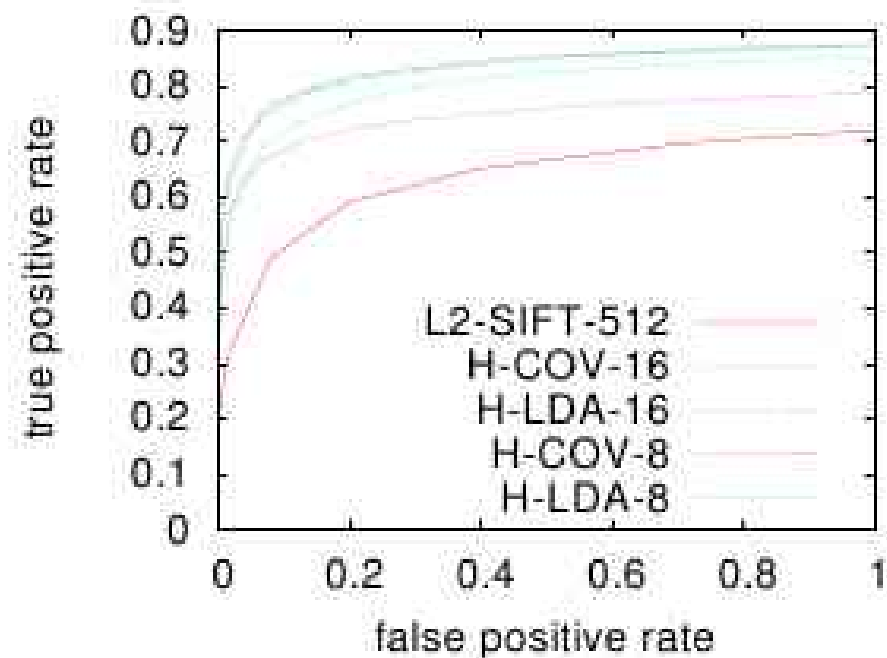
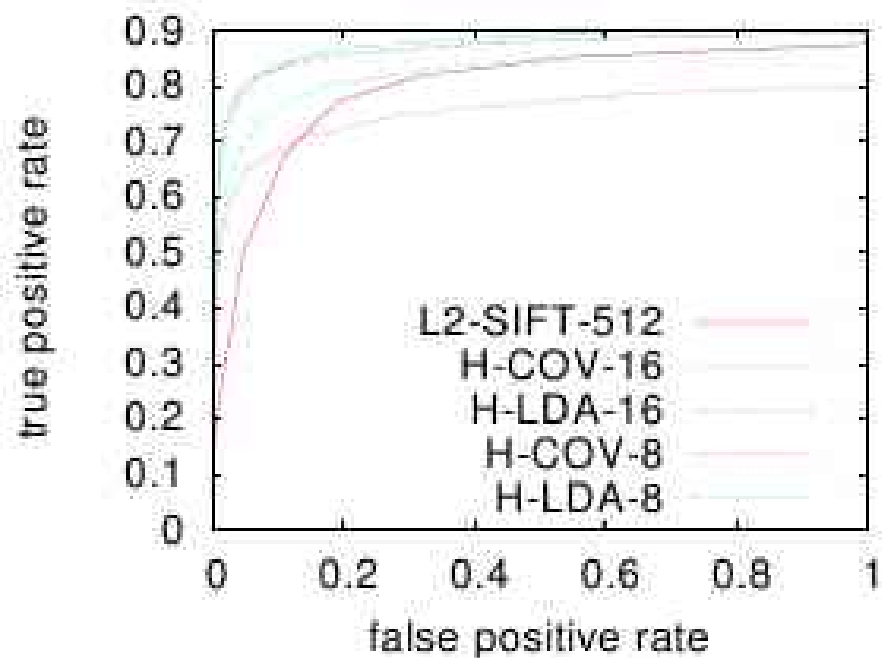
f SIFT feature

Find P and t such that energy is minimized:

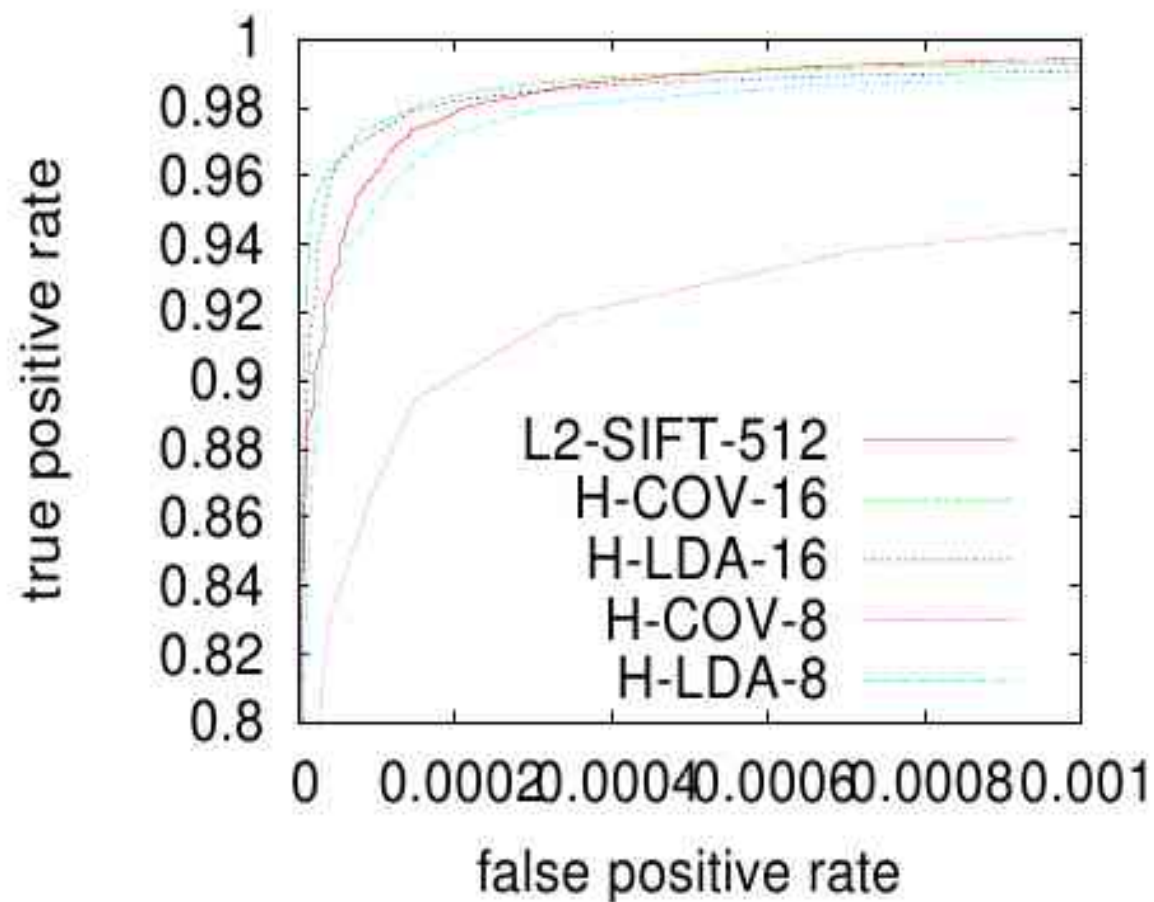
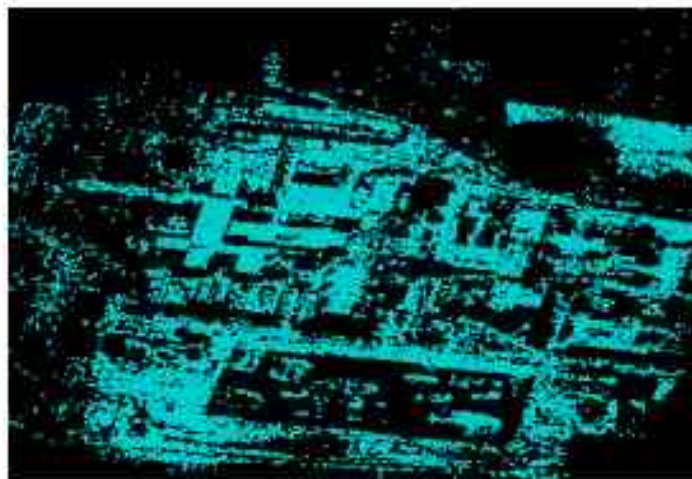
$$E = \lambda \sum_{ij \in p} (B_i - B_j)^2 - \sum_{ij \in n} (B_i - B_j)^2$$



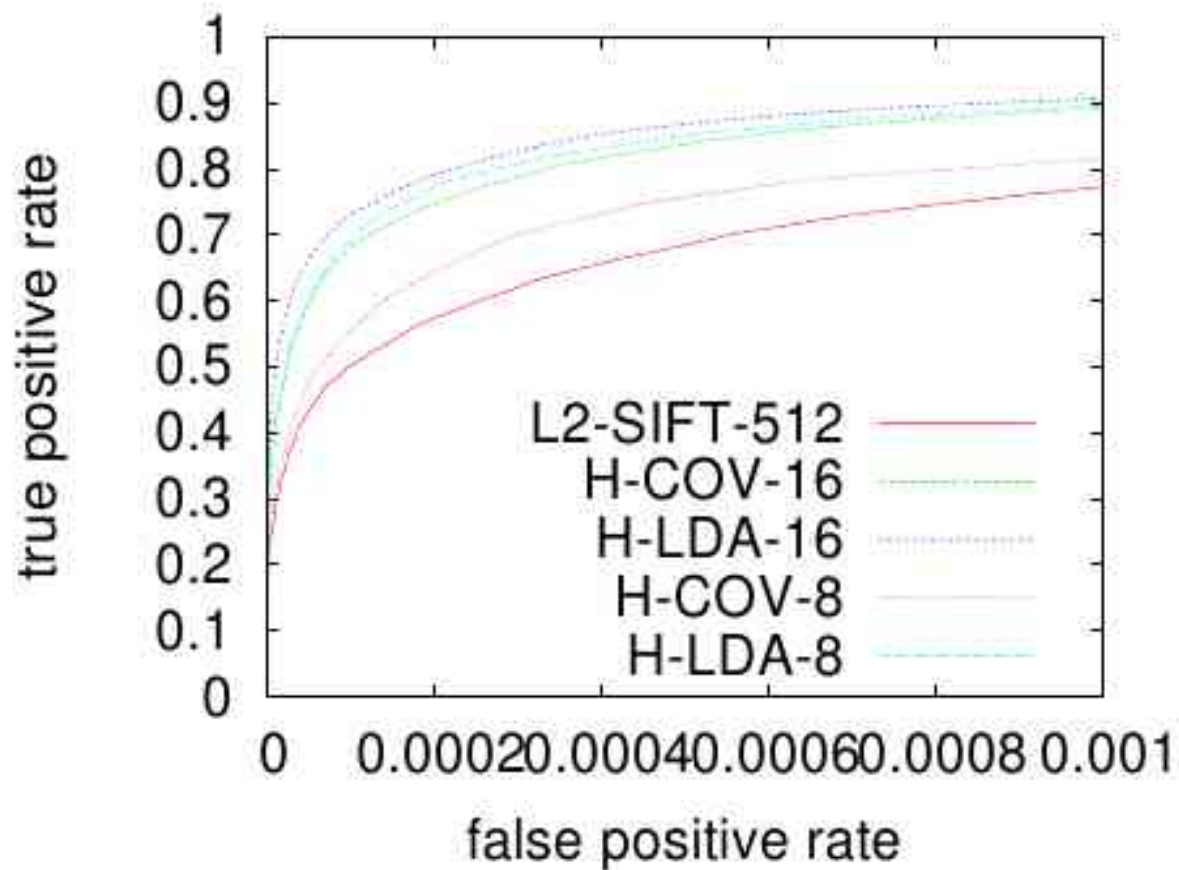
Results on Ground Truth Data



Test on Aerial Images



Results on Urban Scene



Why does this work



Conjunctive closure matches

Largest tracks from Venice
photo community collection



Thank you!