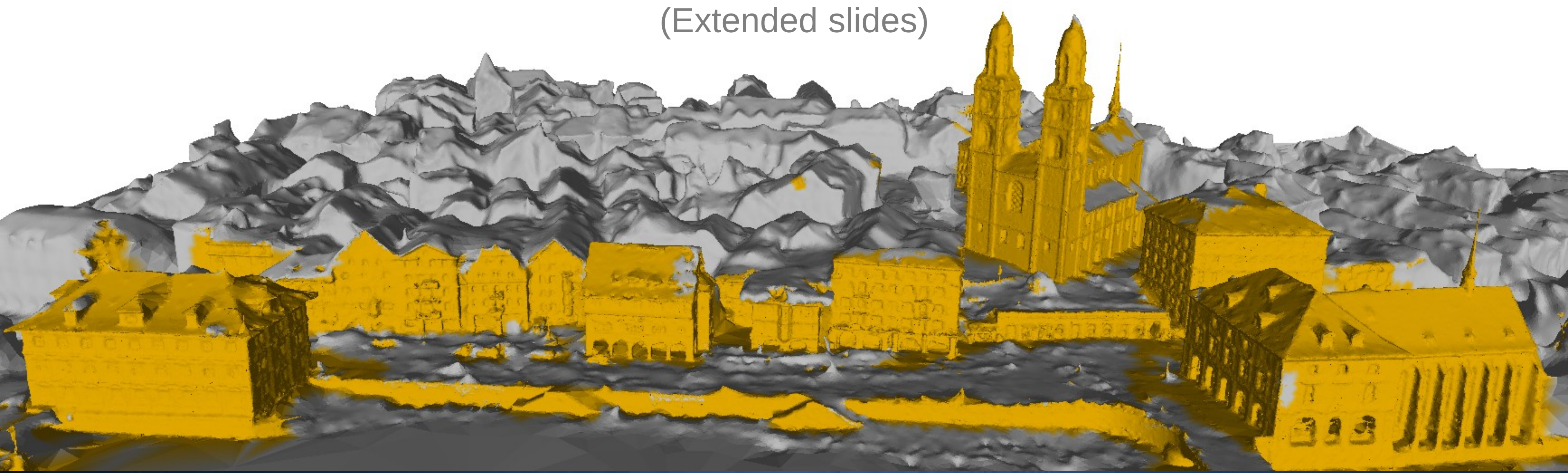
ETH zürich   CVL Computer Vision Lab   erc varcity.eu

ICPR 2016, Cancún, Mexico

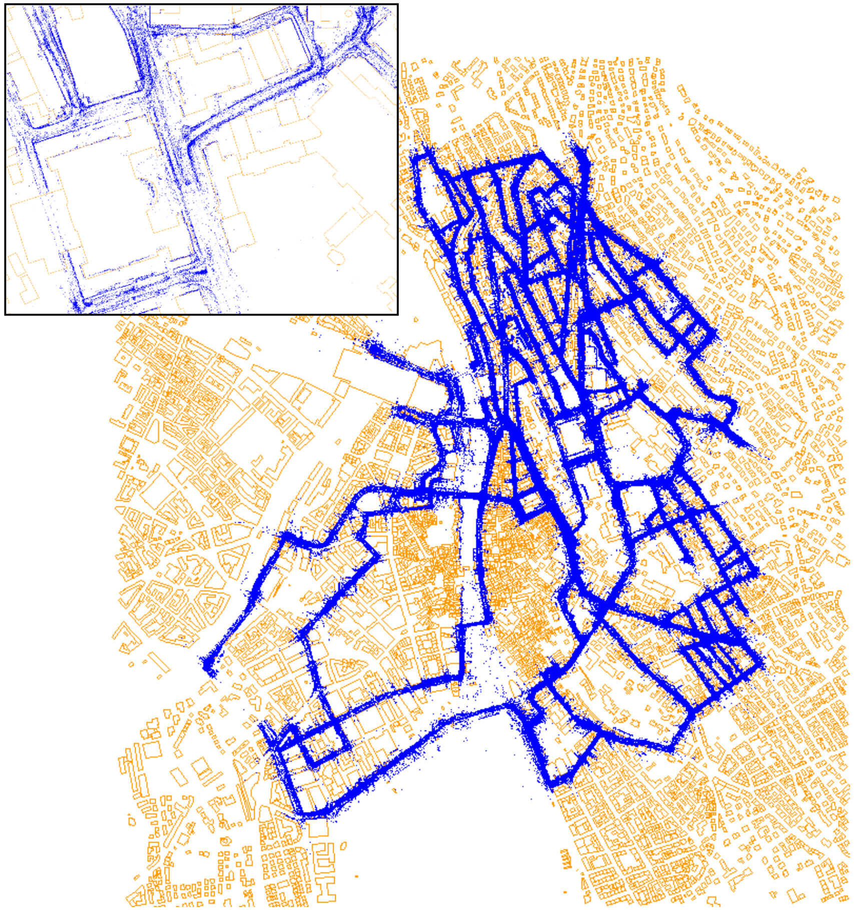
Efficient Volumetric Fusion of Airborne and Street-Side Data for Urban Reconstruction

András Bódis-Szomorú, Hayko Riemenschneider, Luc Van Gool

(Extended slides)

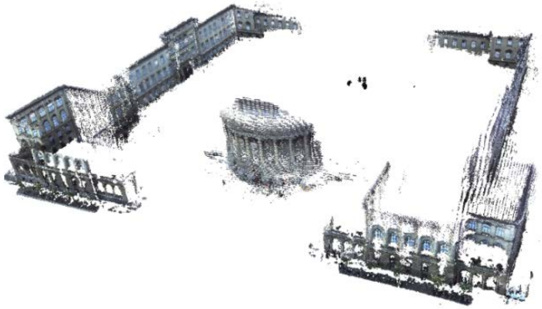


Street-side mobile mapping

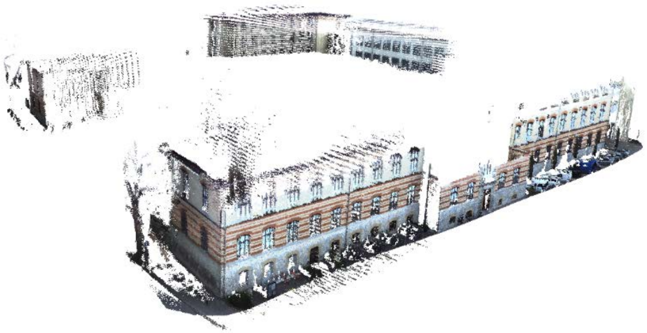


INCOMPLETE MAP COVERAGE

DENSE 3D POINT CLOUD FROM IMAGERY

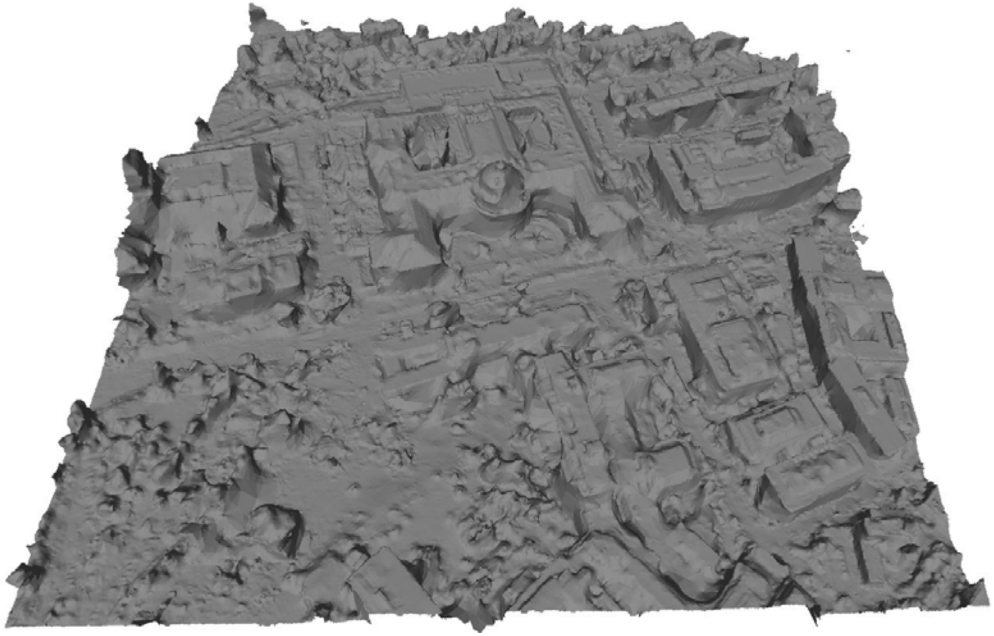


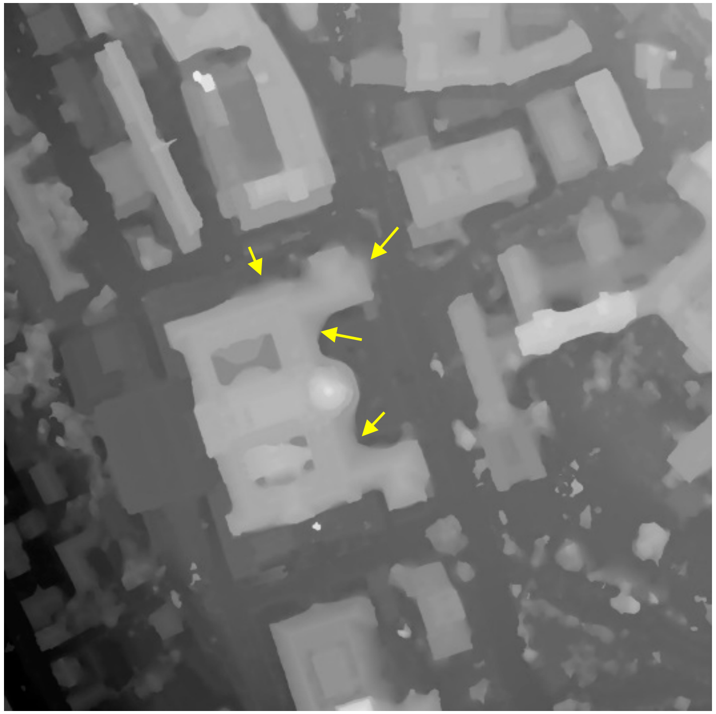
- no upper floors / roofs
- uncovered pedestrian-only areas
- undriven districts
- no courtyards
- occlusions (trees, parking cars, fences)
- often no ground



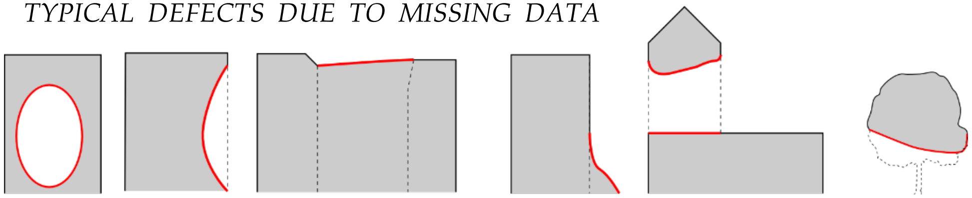
Airborne 3D acquisition

MULTI-VIEW STEREO FROM
15-CENTIMETER NADIR IMAGERY CAPTURED AT 3 KM





TYPICAL DEFECTS DUE TO MISSING DATA



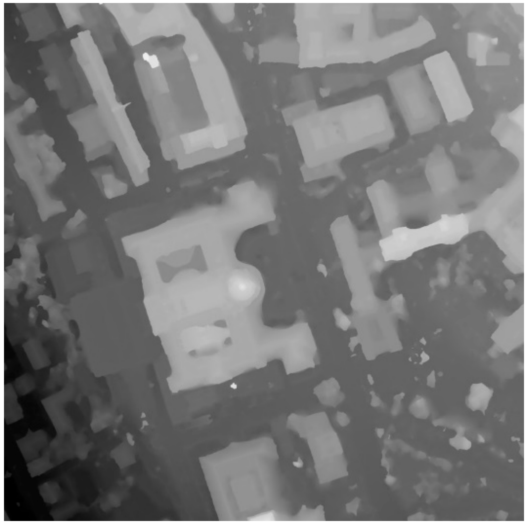
- no detail on street-side
- walls not visible
- shadows
- smoothed (blurry) walls
- defects (holes)

ETH zürich

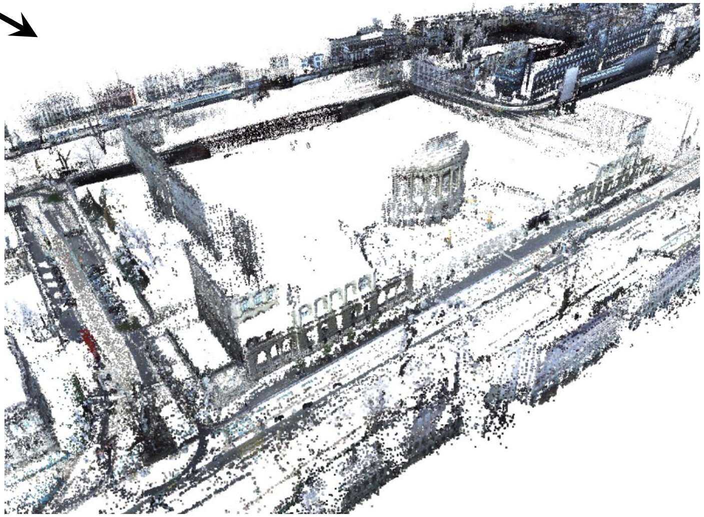
Idea of airborne/street-side fusion

COMPLETENESS

ROOFS
HIGH FLOORS
COURTYARDS
MISSING STREETS



AIRBORNE NADIR 3D

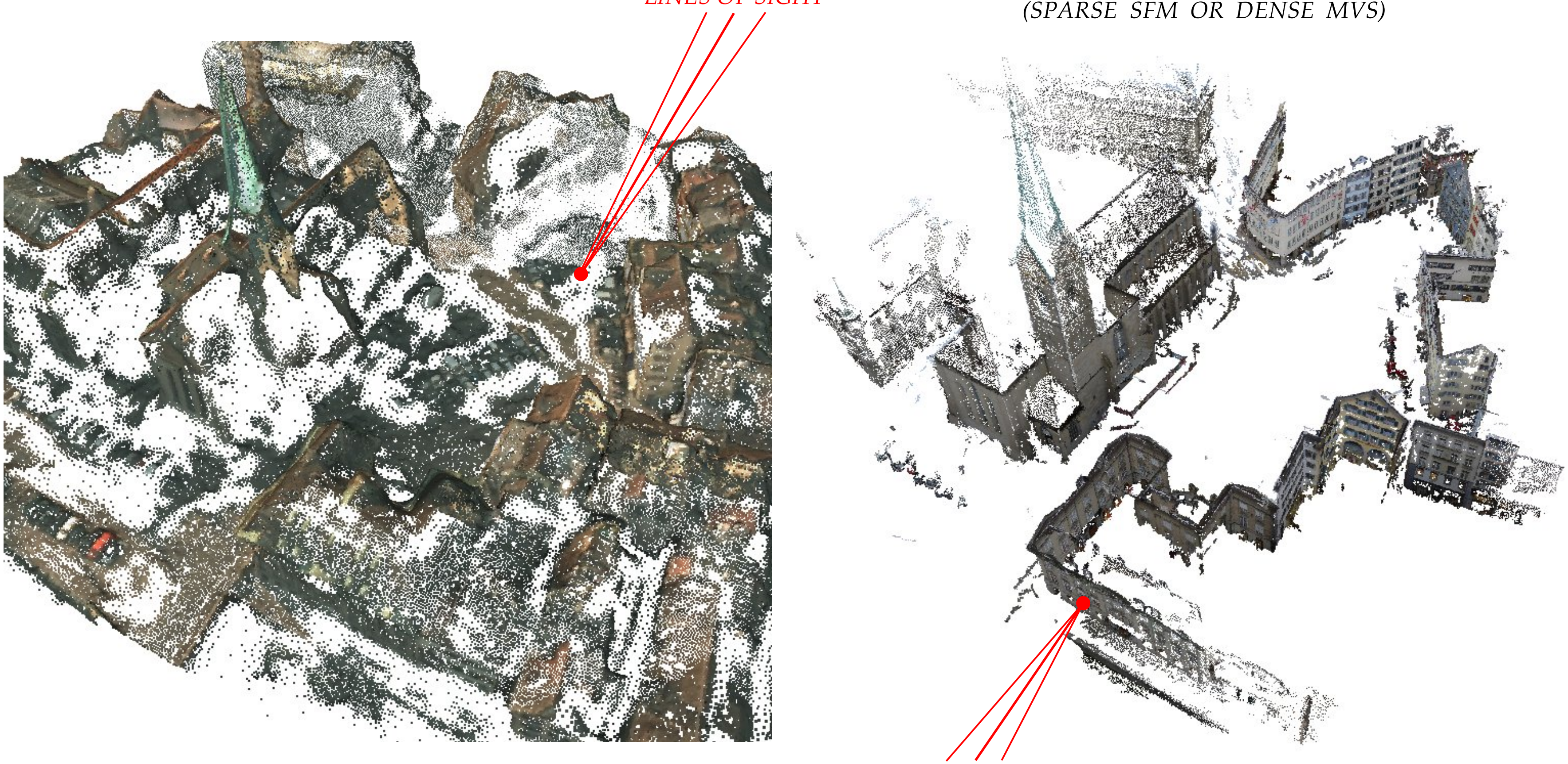


STREET-SIDE 3D
(A MOBILE MAPPING EXAMPLE)

DETAIL

Requirements: efficient + watertight + large-scale

This paper: efficient volumetric airborne/streetside fusion



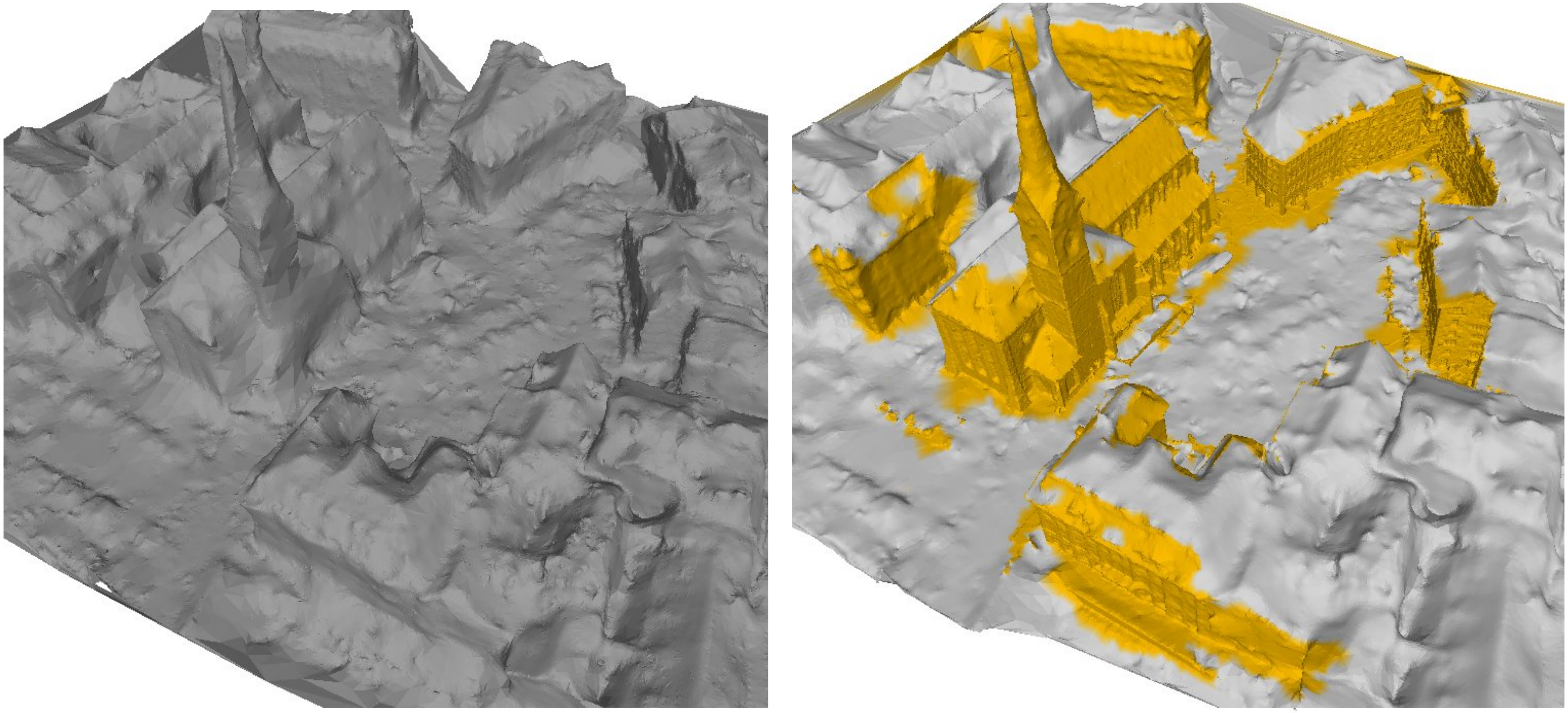
Assumptions:

- visibility data (lines of sight) given
- point clouds are accurately geo-registered (via precise GPS/IMU or GCP's)

This paper: efficient volumetric airborne/streetside fusion

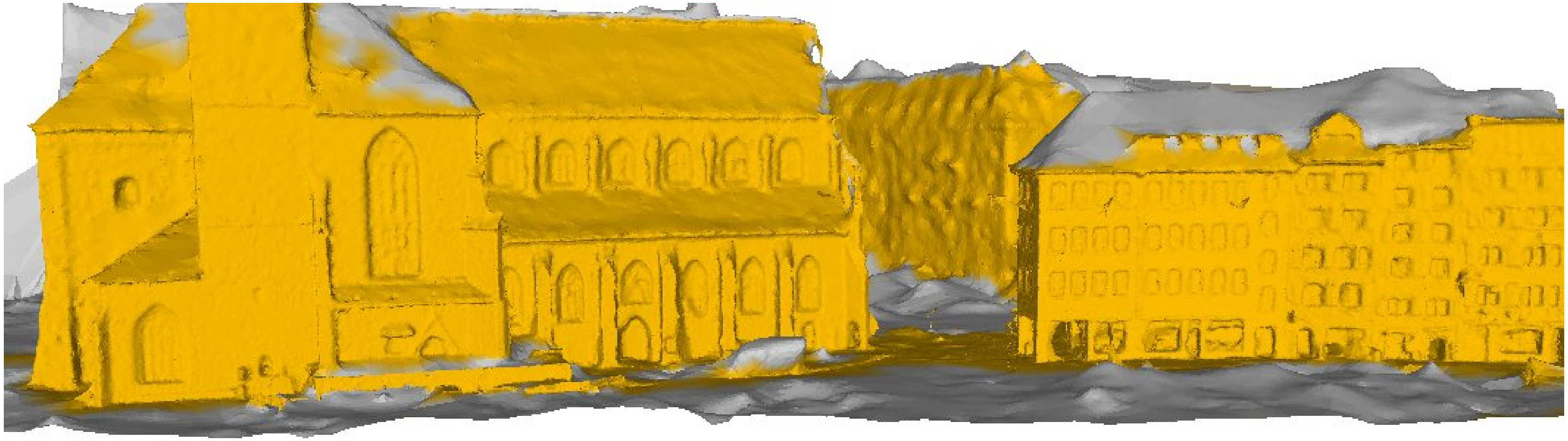
AIRBORNE ONLY (FOR COMPARISON)

OUR FUSION RESULT

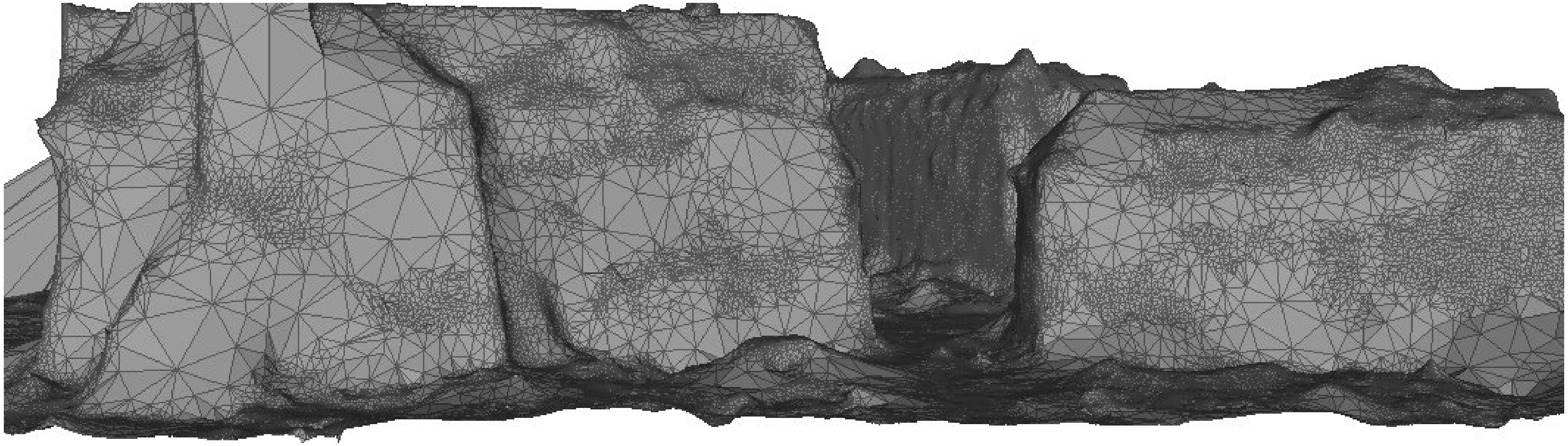


This paper: efficient volumetric airborne/streetside fusion

OUR FUSION RESULT



AIRBORNE ONLY (FOR COMPARISON)



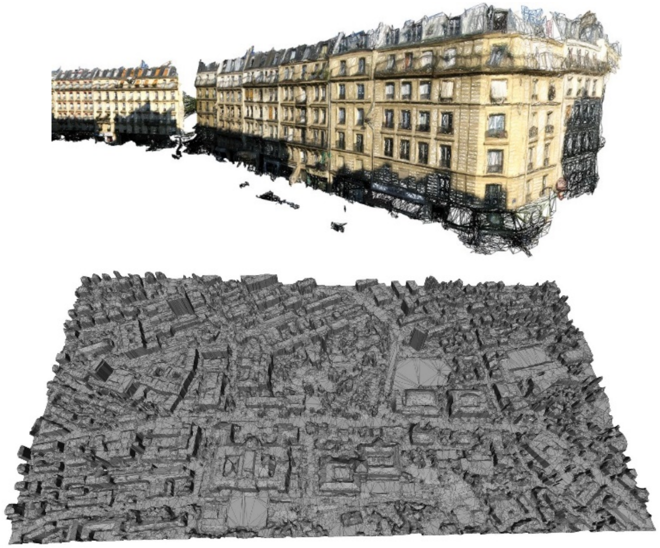
Related work

General surface reconstruction

- explicit methods, e.g. zippering
- depth map integration via TSDF, e.g. VRIP, KinectFusion (Curless & Levoy SG'96, Izadi et al. '06)
- s/t cut over voxel grid (unsigned, UDF) (Hornung & Kobbelt EG'06, Lempitsky & Boykov CVPR'07)
- Poisson Surface Reconstruction (octree) (Kazhdan EG'06)
- convex variational (voxels or height map), e.g. TV-L1, TGV-fusion
- cell complex (Chauve CVPR'10)
- 3DT approaches (Labatut et al. ICCV'07, Lafarge & Alliez EG'13)

Street-side & airborne data

- superpixel meshes (Bodis et al. CVIU'16)
- most address geo-localization / registration
- DSM + street-side LiDAR (Fruh & Zakhor CGA'03)
- Octree & Dual Contouring (Fiocco et al. 3DIM'05)
- Poisson Surface Reconstruction (Shan et al. 3DPVT'13)

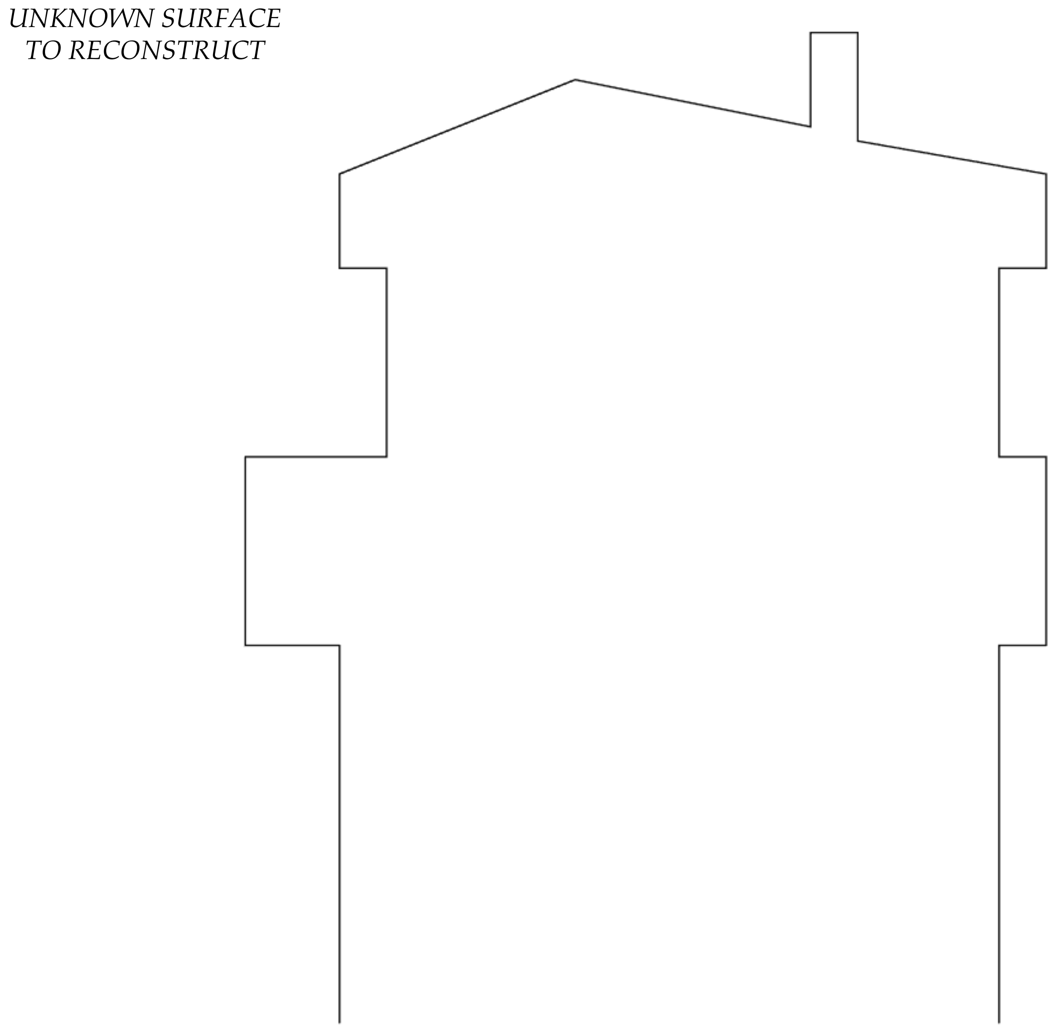


Superpixel meshes (Bodis et al. CVIU'16)

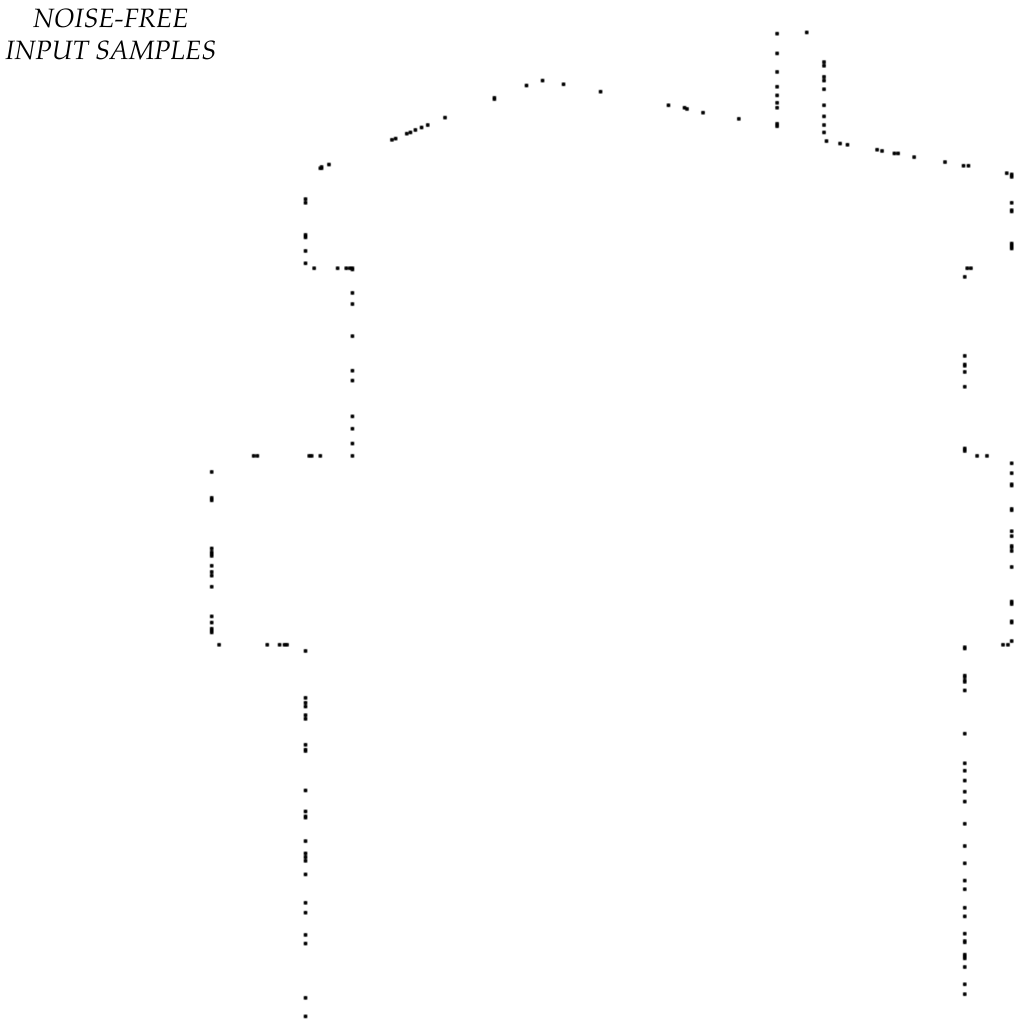
Contributions

- *1st to propose 3DT-fusion for airborne/street-side*
- *Point cloud blending against gross ray conflicts*
- *Techniques to reduce workload (large urban scenes)*
- *Many experiments on detail vs. workload (sparse & dense input)*

Volumetric surface reconstruction via 3DT

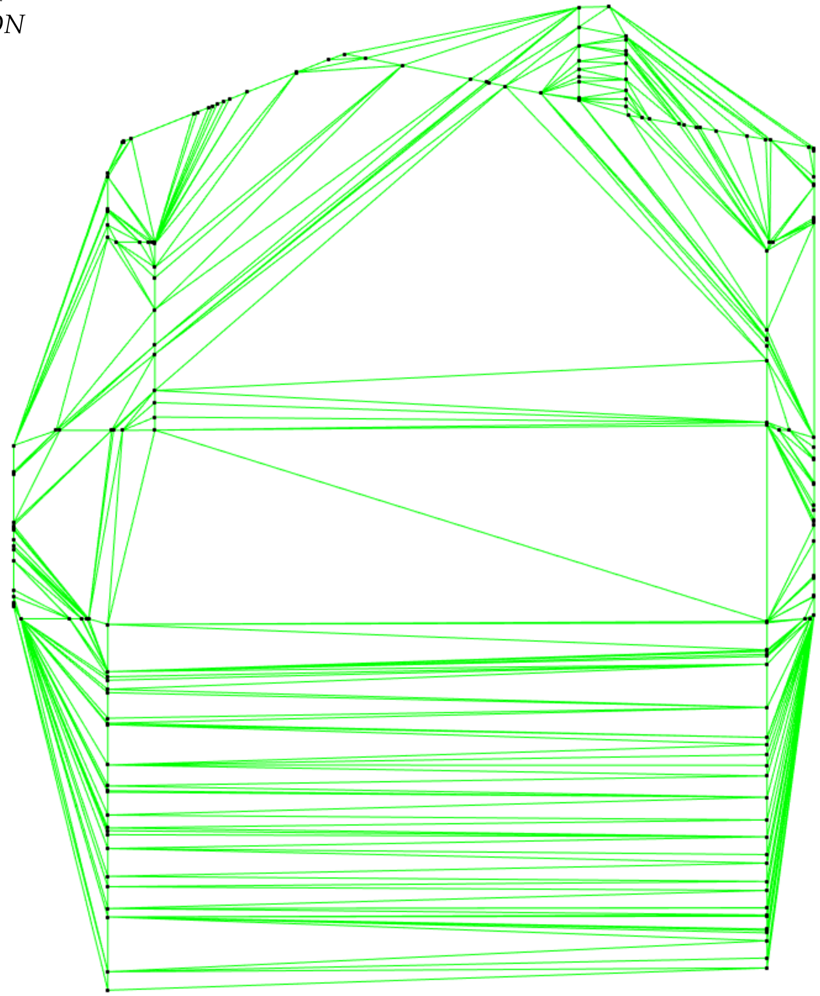


Volumetric surface reconstruction via 3DT



Volumetric surface reconstruction via 3DT

3D DELAUNAY TRIANGULATION



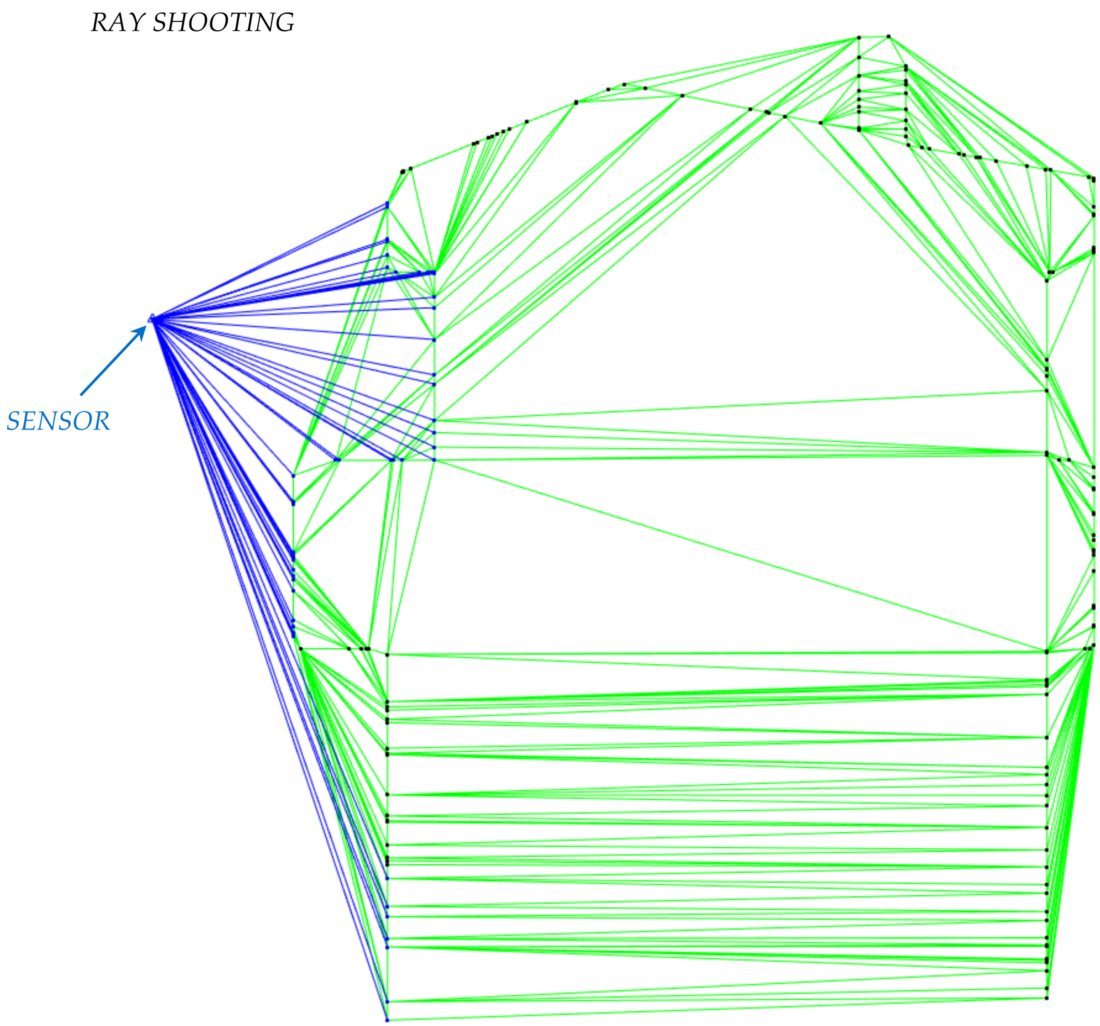
Alternatives

- voxels (poor scalability)
- stixels (2.5D)
- octrees (more complicated)
- cell complex
  (poor scalability, needs planes)

Why 3DT?

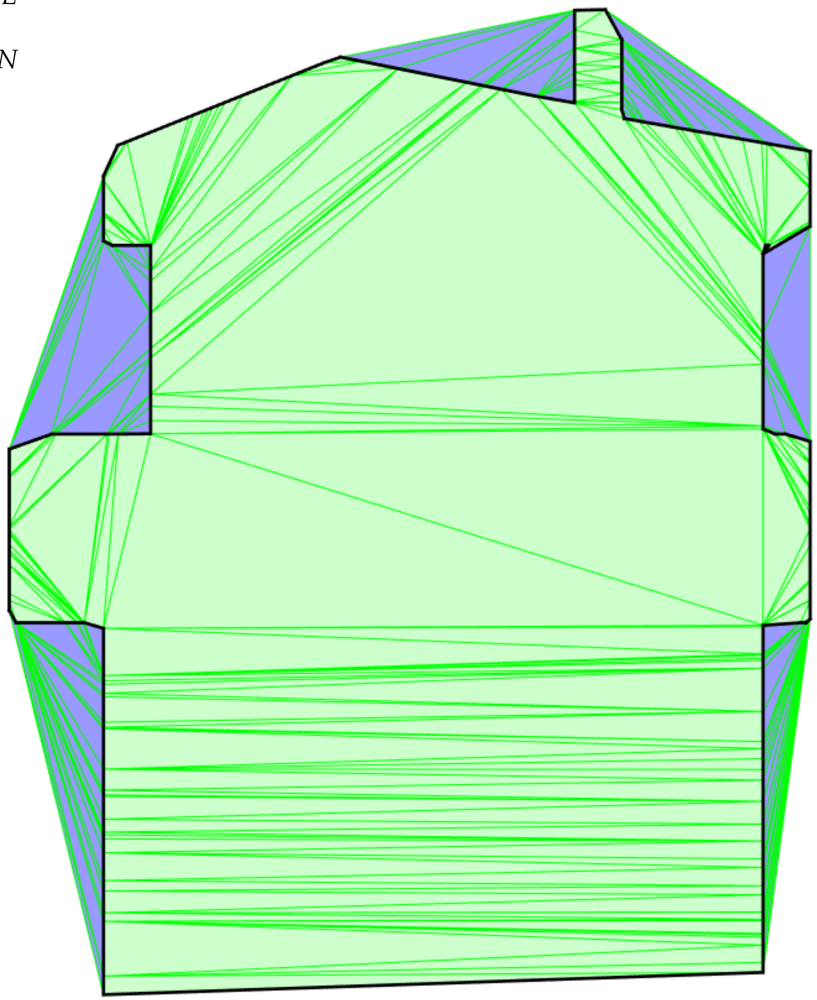
- simple
- adaptive / scalable
- efficient
- detail-preserving

Volumetric surface reconstruction via 3DT



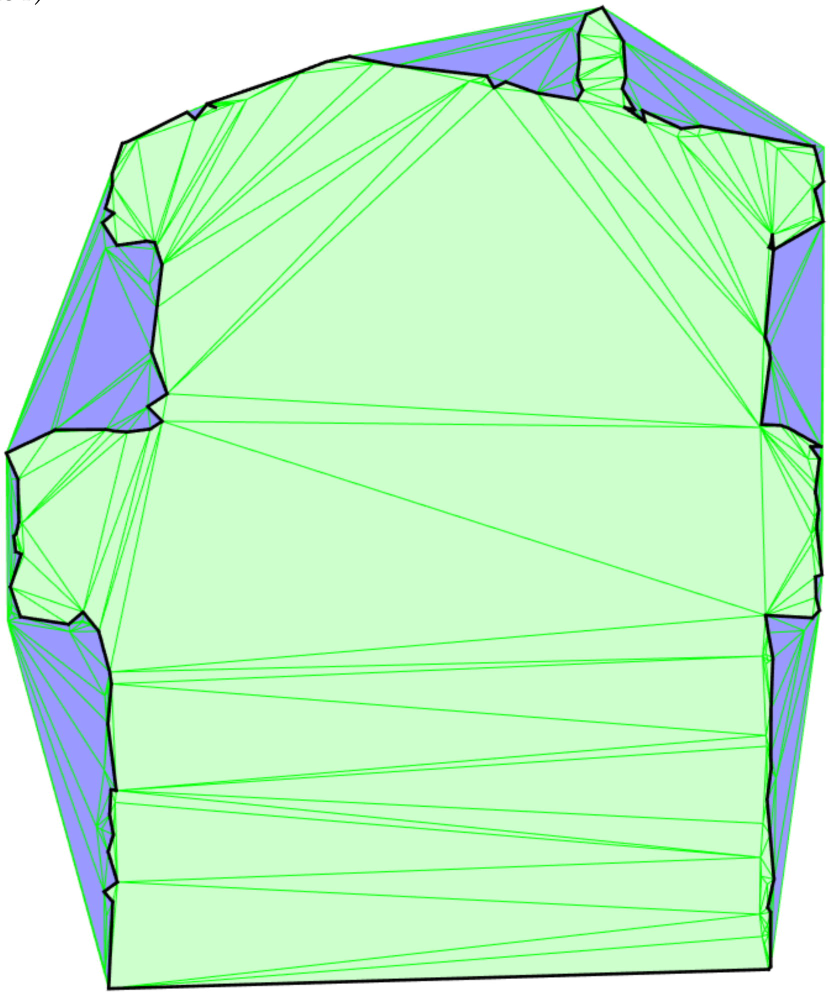
Volumetric surface reconstruction via 3DT

INSIDE/OUTSIDE VOLUMETRIC CLASSIFICATION



Volumetric surface reconstruction via 3DT

REAL DATA (NOISY)



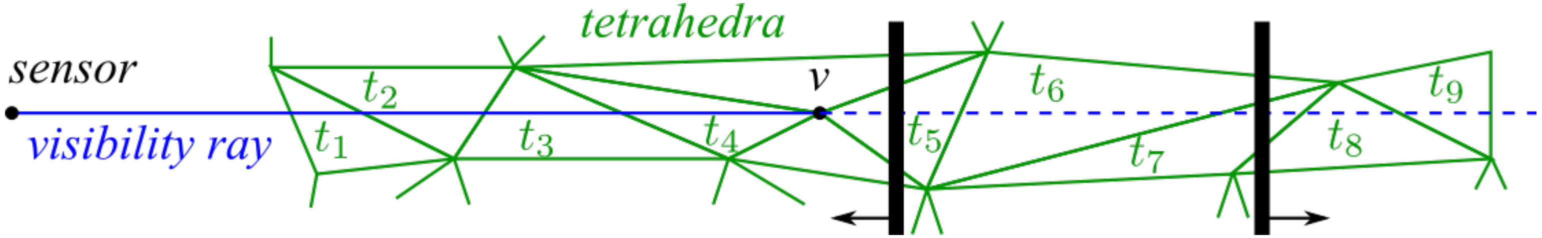
Post processing

- Denoising: simple smoothing
- Remove floating components

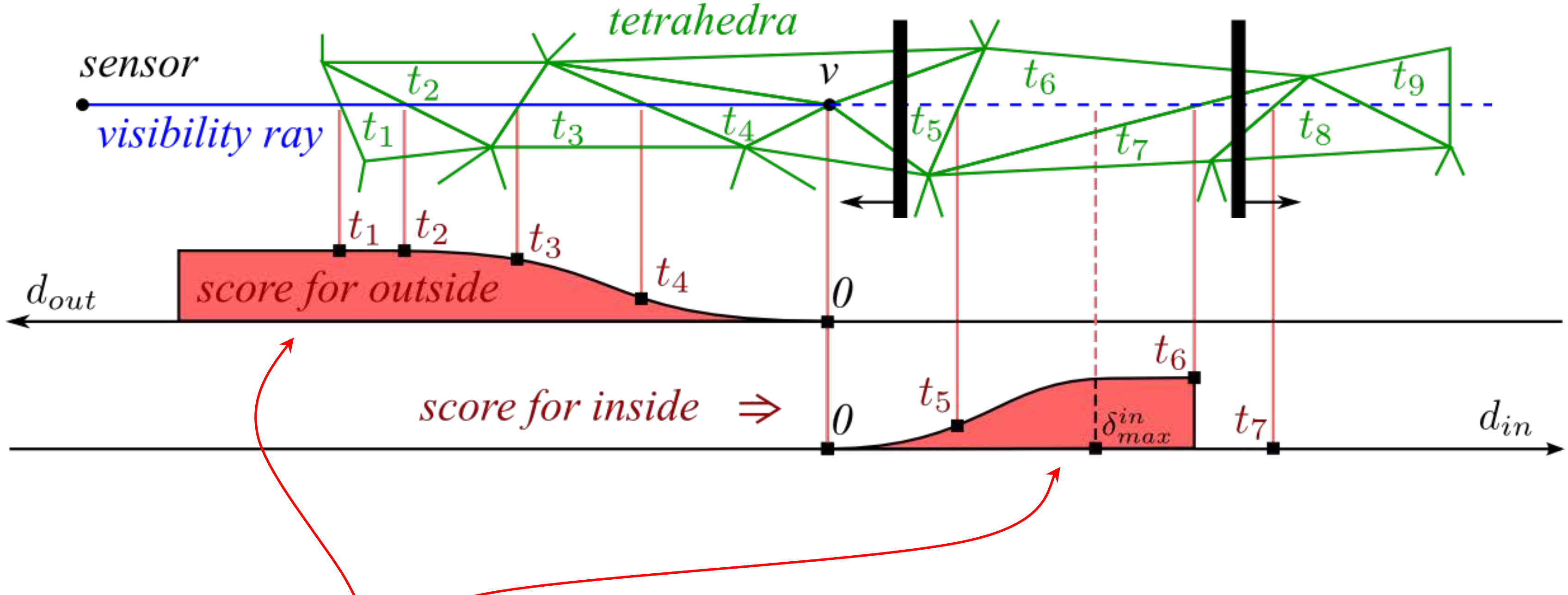
Expensive alternative

- Mesh tuning for photoconsistency (see Vu et al. PAMI 2012)

Raycasting and voting scheme



Raycasting and voting scheme



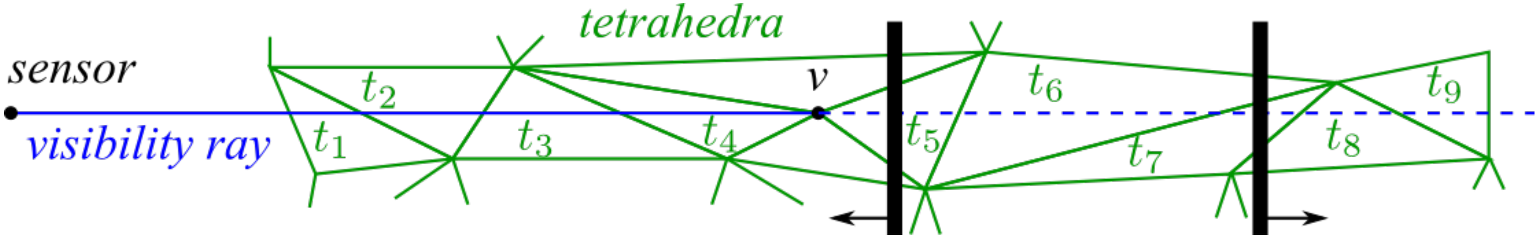
$$S_l(r,t) = 1 - e^{-d_l^2(r,t)/(2\sigma_l^2)} \qquad l \in \{in, out\}$$

inside and outside score
per ray per tetrahedron

ETH zürich

CVL Computer Vision Lab

Volumetric optimization



$$E(\mathcal{L}) = \underbrace{\sum_{i:t_i \in \mathcal{T}} E_i(l_i)} + \sum_i \underbrace{\sum_{j:j<i} E_{ij} \cdot \mathbb{I}[l_i \neq l_j]}$$

unary preference for in/out per tetrahedron

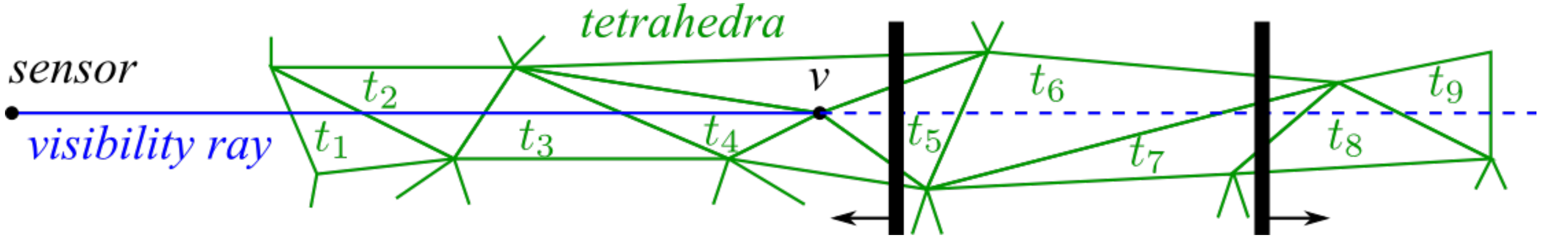
penalty for a face to be part of the surface

$l_i \in \{in, out\}$ binary labels

$\mathcal{T} = \{t_i\}$ tetrahedra

Optimization: globally optimal GCO

Volumetric optimization



$$E(\mathcal{L}) = \sum_{i:t_i \in \mathcal{T}} E_i(l_i) + \sum_i \sum_{j:j<i} E_{ij} \cdot \mathbb{I}[l_i \neq l_j]$$

unary preference for in/out per tetrahedron

penalty for a face to be part of the surface

$l_i \in \{in, out\}$ binary labels

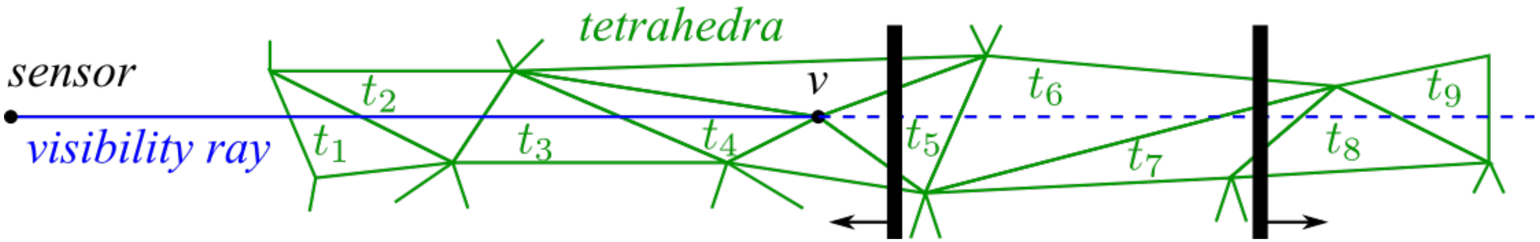
$\mathcal{T} = \{t_i\}$ tetrahedra

Optimization: globally optimal GCO

$U_i(l) = \sum_r S_{\bar{l}}(r, t_i)$ sum over all rays

$E_i(l) = 1 - e^{-U_i(l)/\gamma_l}$ normalization

Volumetric optimization



$$E(\mathcal{L}) = \underbrace{\sum_{i:t_i \in \mathcal{T}} E_i(l_i)} + \sum_i \sum_{j:j<i} \underbrace{E_{ij} \cdot \mathbb{I}[l_i \neq l_j]}$$

unary preference for in/out per tetrahedron

penalty for a face to be part of the surface

$l_i \in \{in, out\}$ binary labels

$\mathcal{T} = \{t_i\}$ tetrahedra

Optimization: globally optimal GCO

$$U_i(l) = \sum_r S_{\bar{l}}(r, t_i)$$ sum over all rays

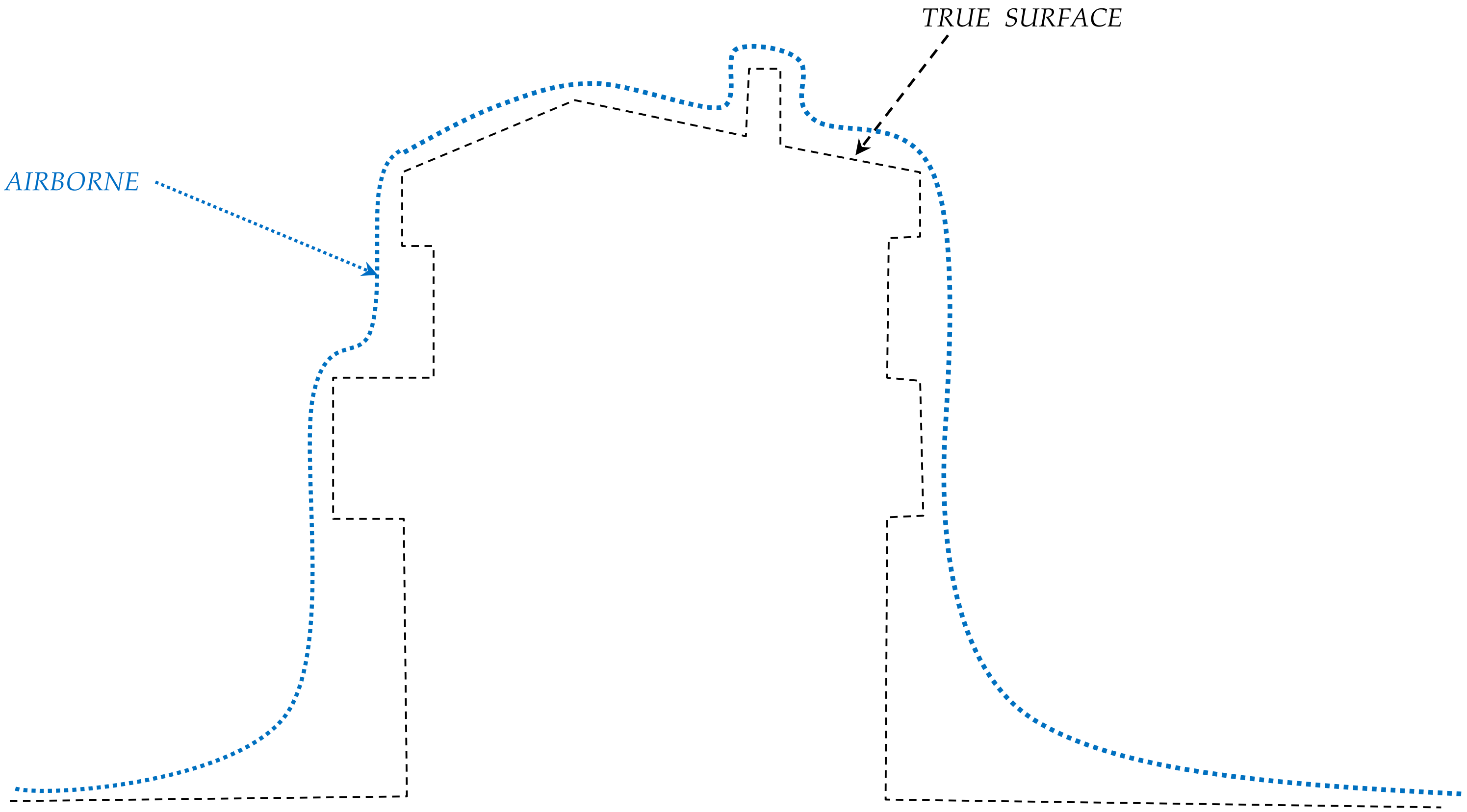
$$E_i(l) = 1 - e^{-U_i(l)/\gamma_l}$$ normalization

$$E_{ij} = \lambda A_{ij}$$ area penalty (worked best with our unary)

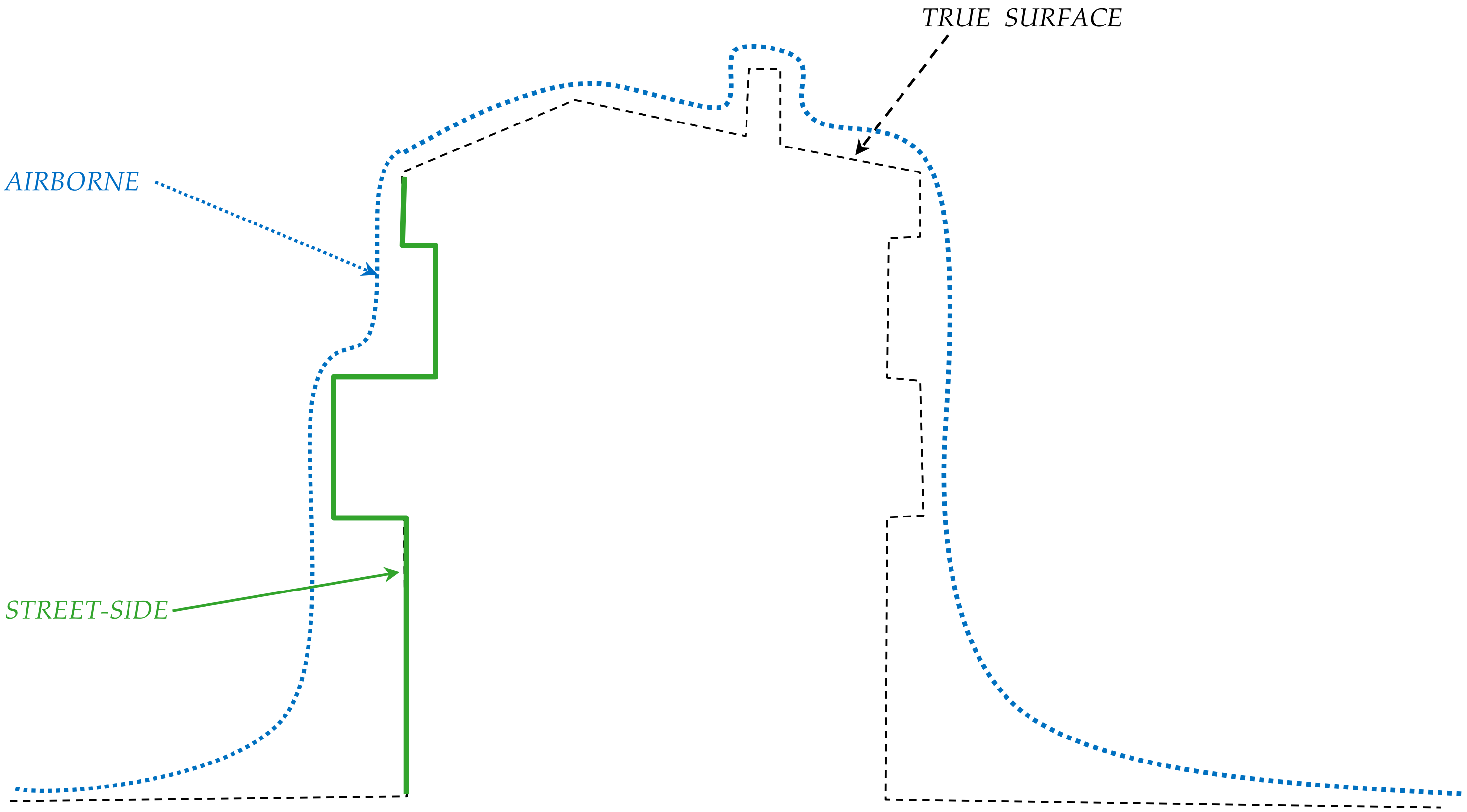
Other examples tried

- triangle elongatedness
- long edge-triangles
- beta-skeleton (Labatut et al., CGF 2009)

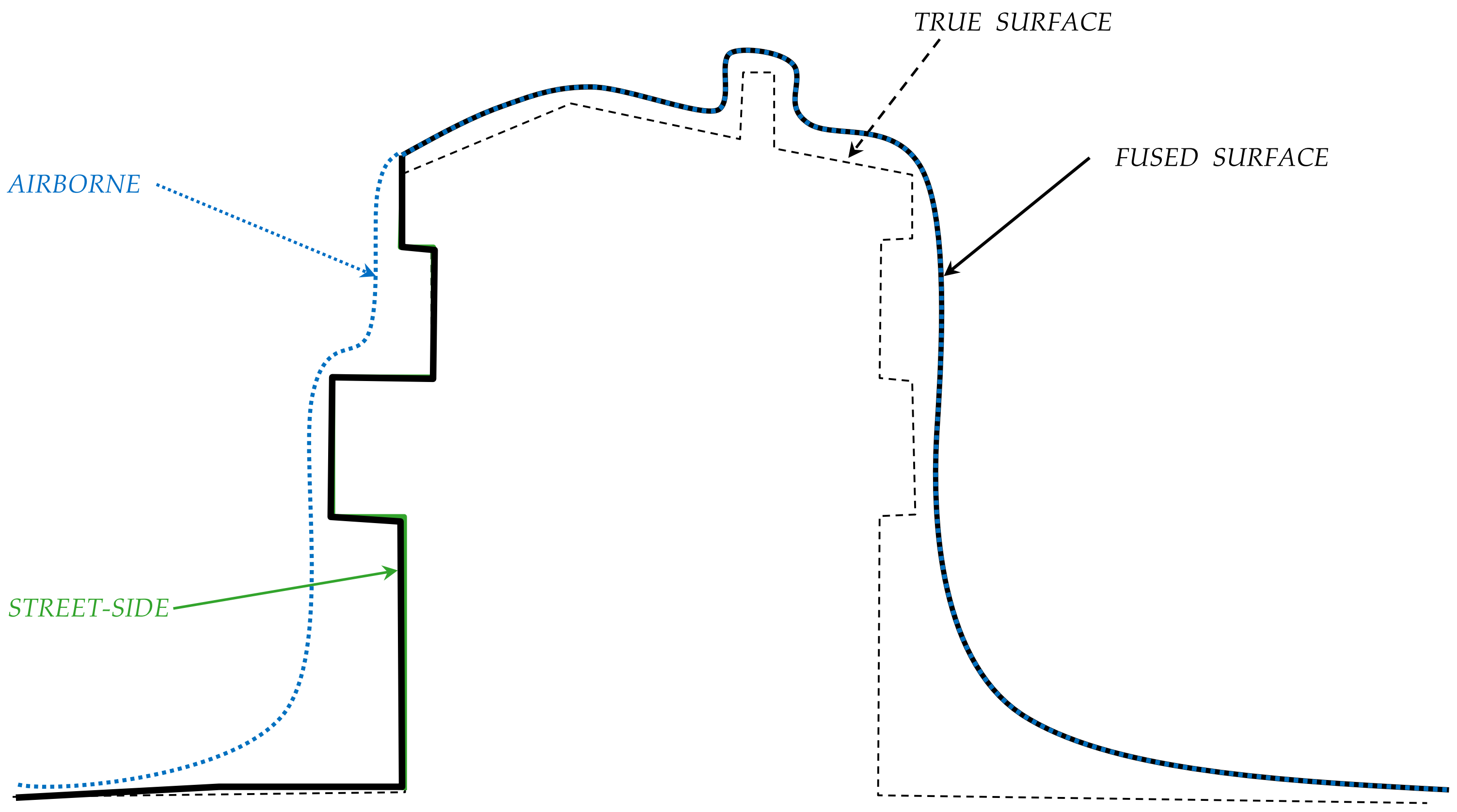
Airborne / street-side surface fusion



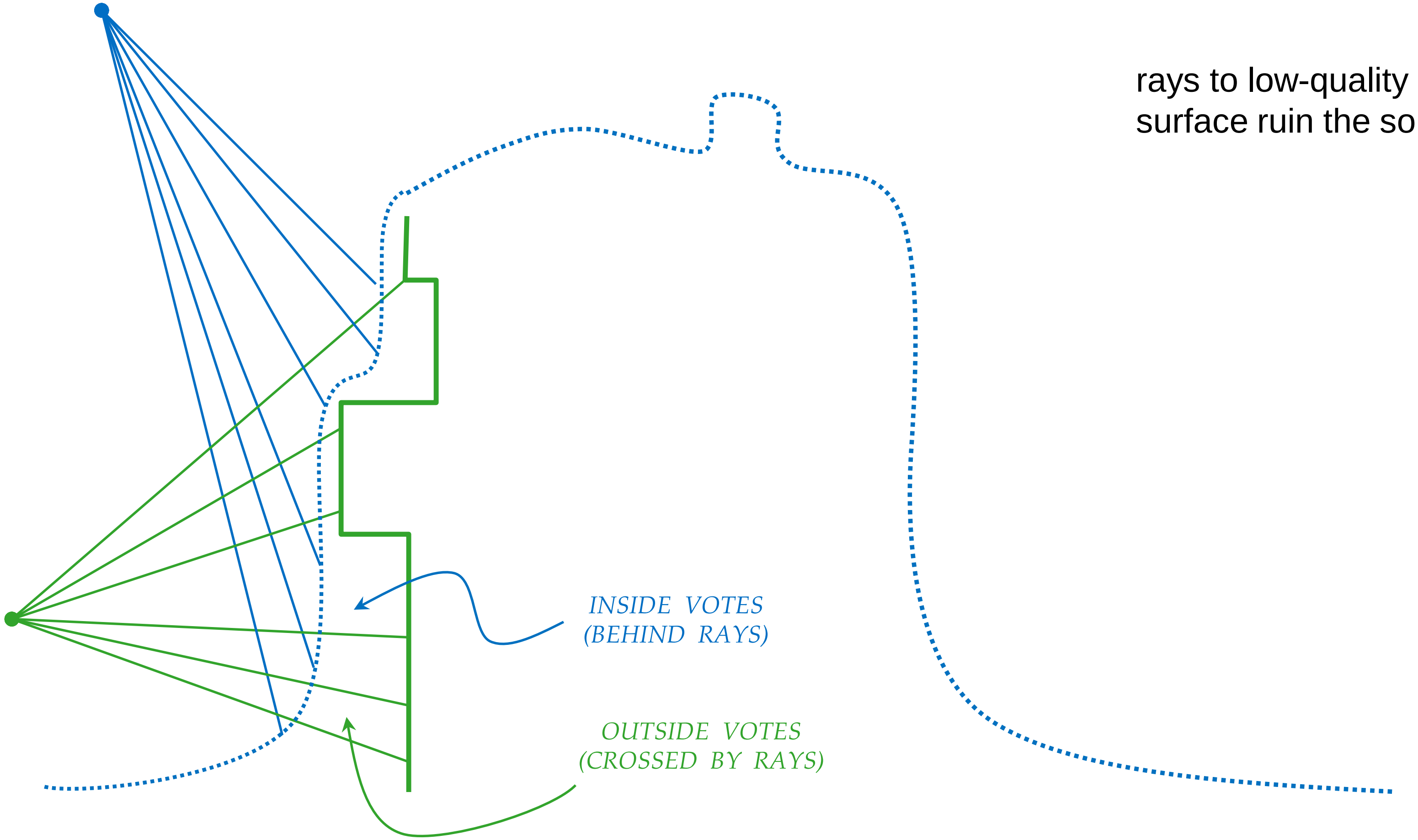
Airborne / street-side surface fusion



Airborne / street-side surface fusion

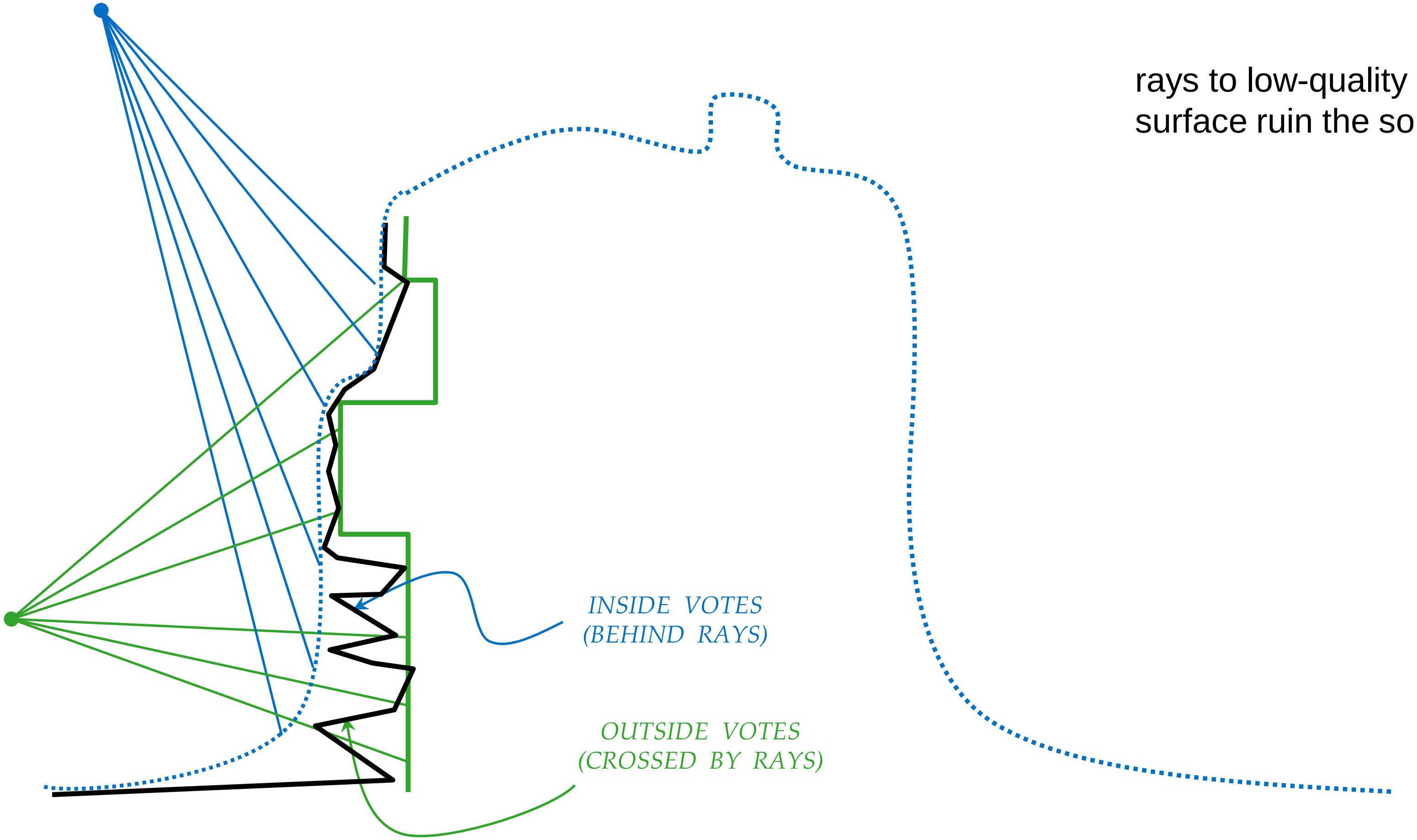


Airborne / street-side ray conflicts



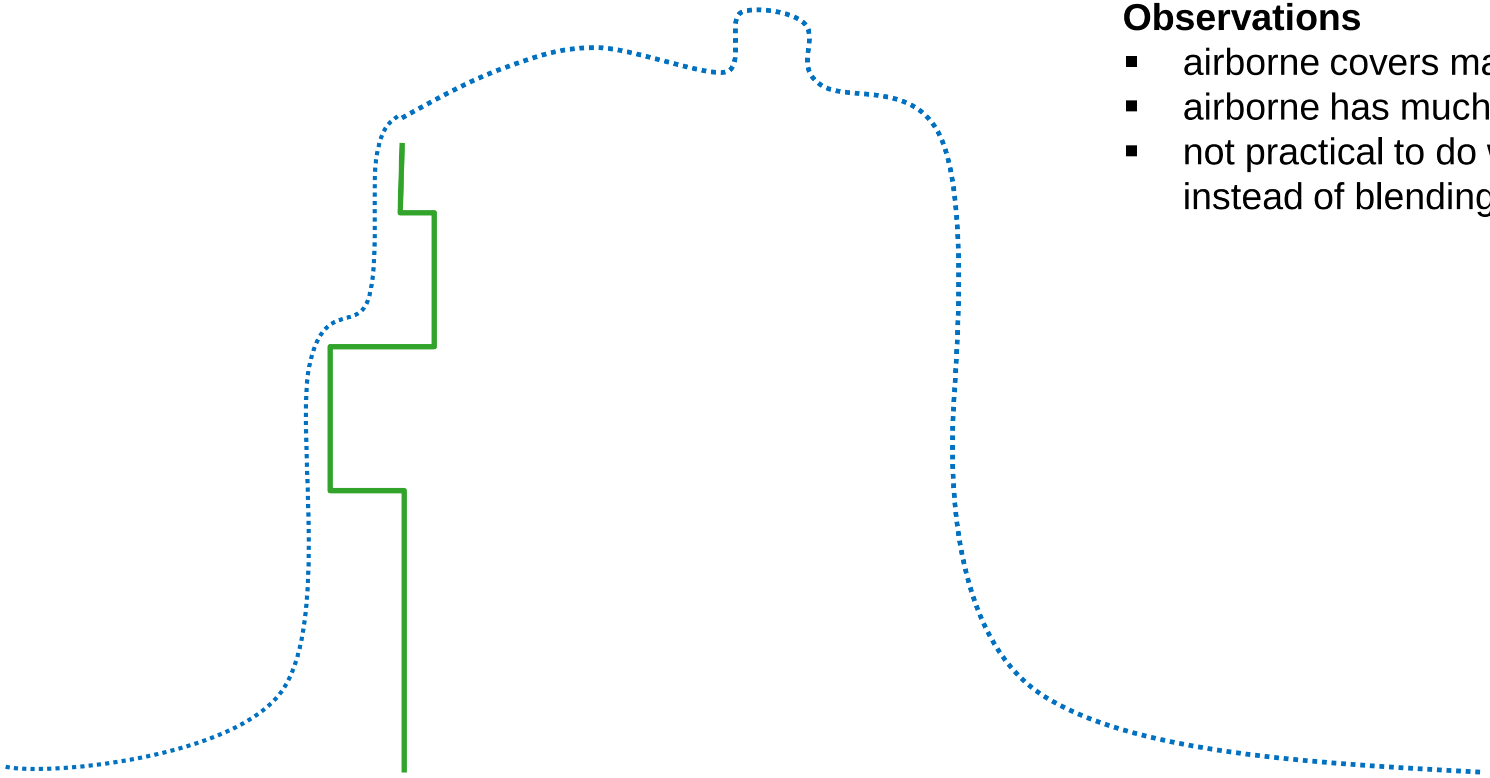
rays to low-quality airborne surface ruin the solution

Airborne / street-side ray conflicts



rays to low-quality airborne surface ruin the solution

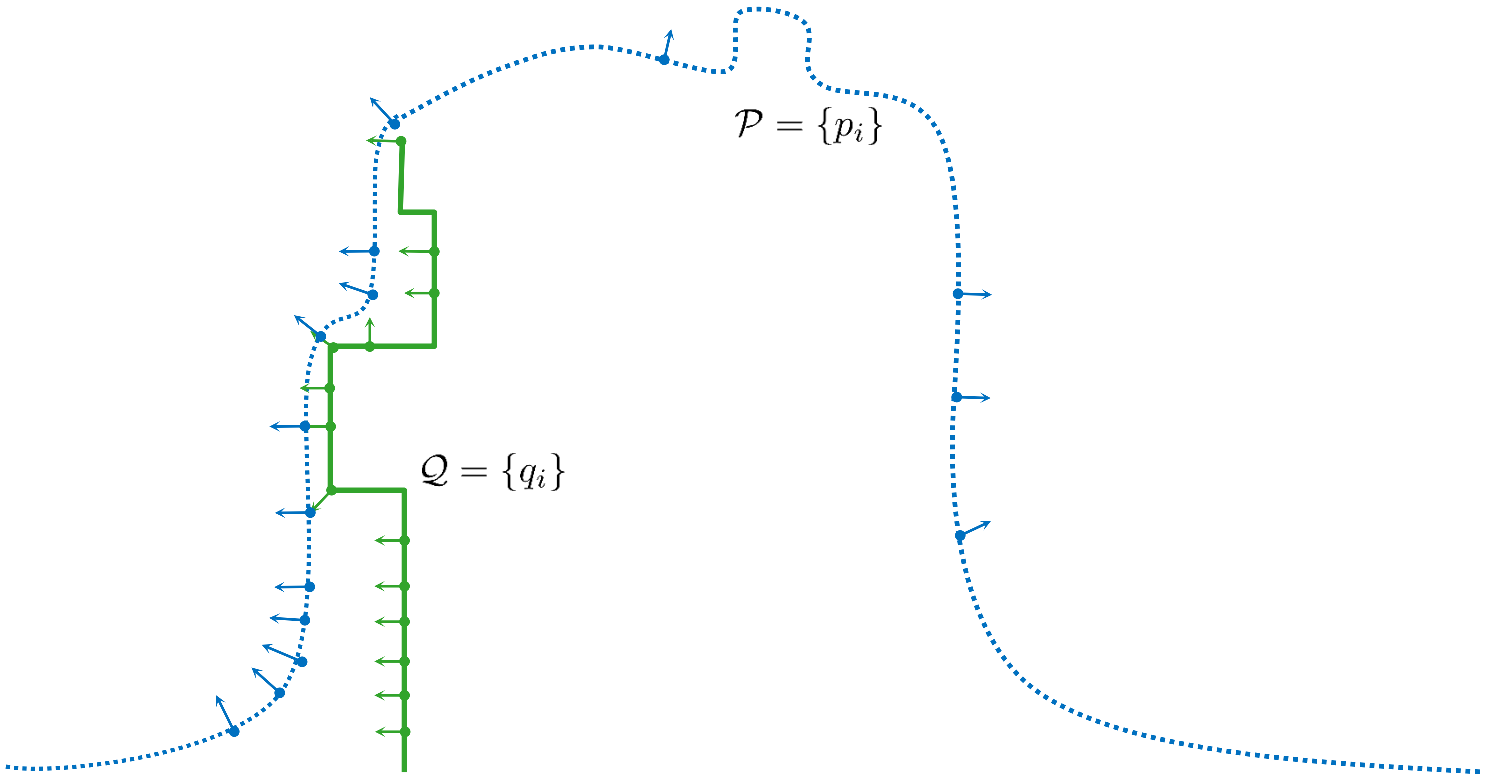
Airborne / street-side point cloud blending



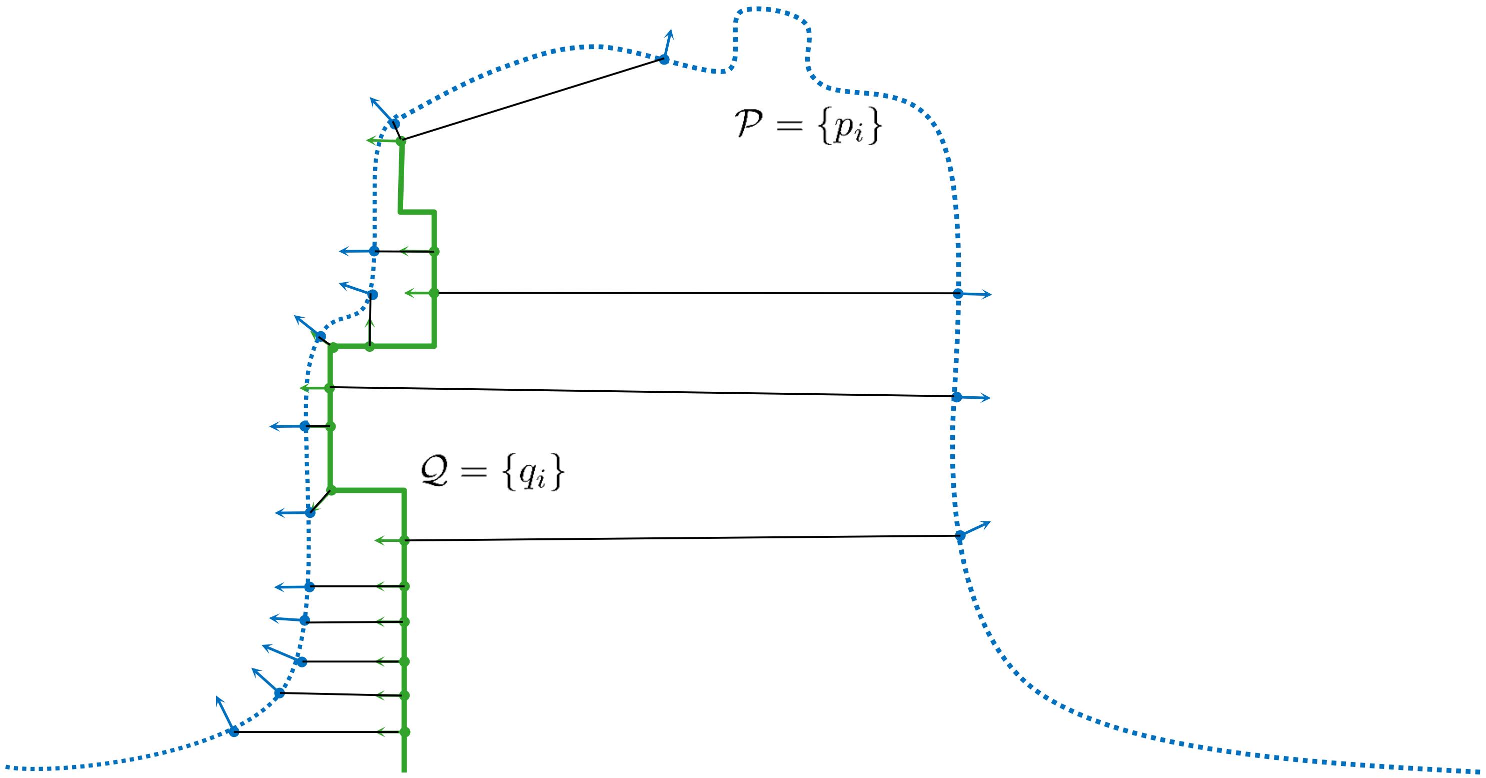
Observations

- airborne covers map
- airborne has much less detail
- not practical to do weighting instead of blending

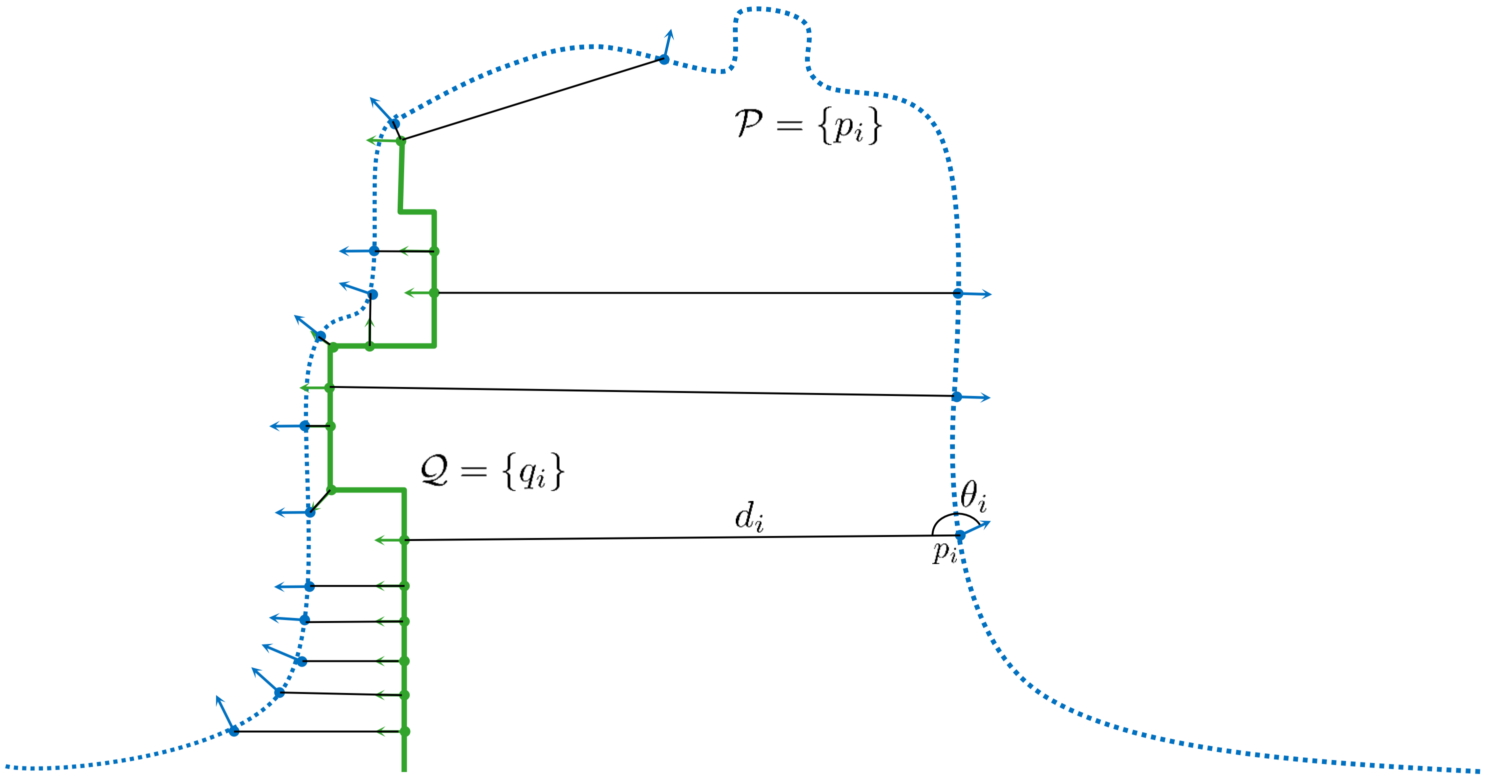
Airborne / street-side point cloud blending



Airborne / street-side point cloud blending



Airborne / street-side point cloud blending

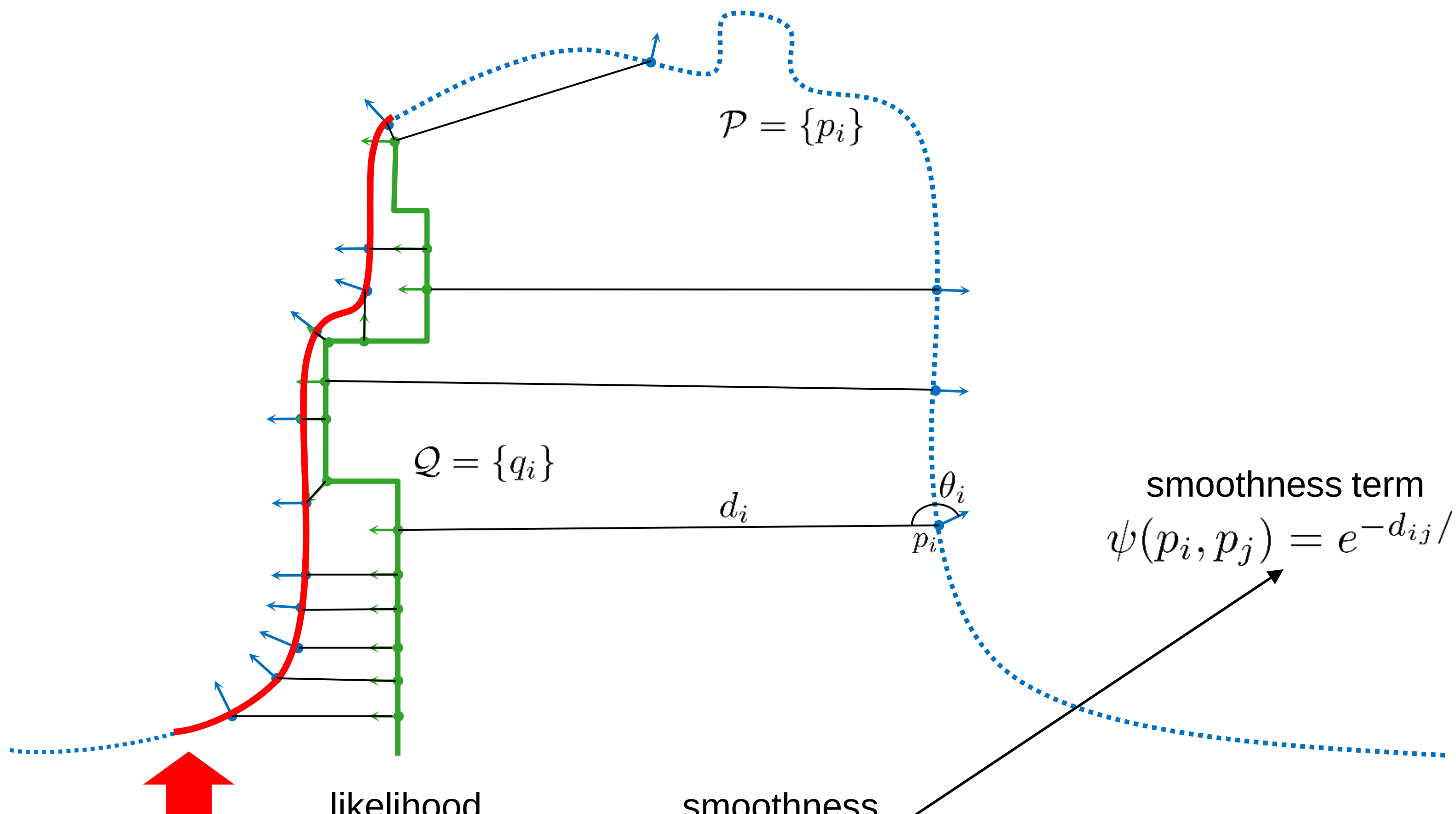


distance term | normal term

$$\phi_i = \phi(d_i, \theta_i) = e^{-d_i^2/(2\sigma_b^2)} \cdot \max\{0, \cos\theta_i\}$$

substitute likelihood

Airborne / street-side point cloud blending



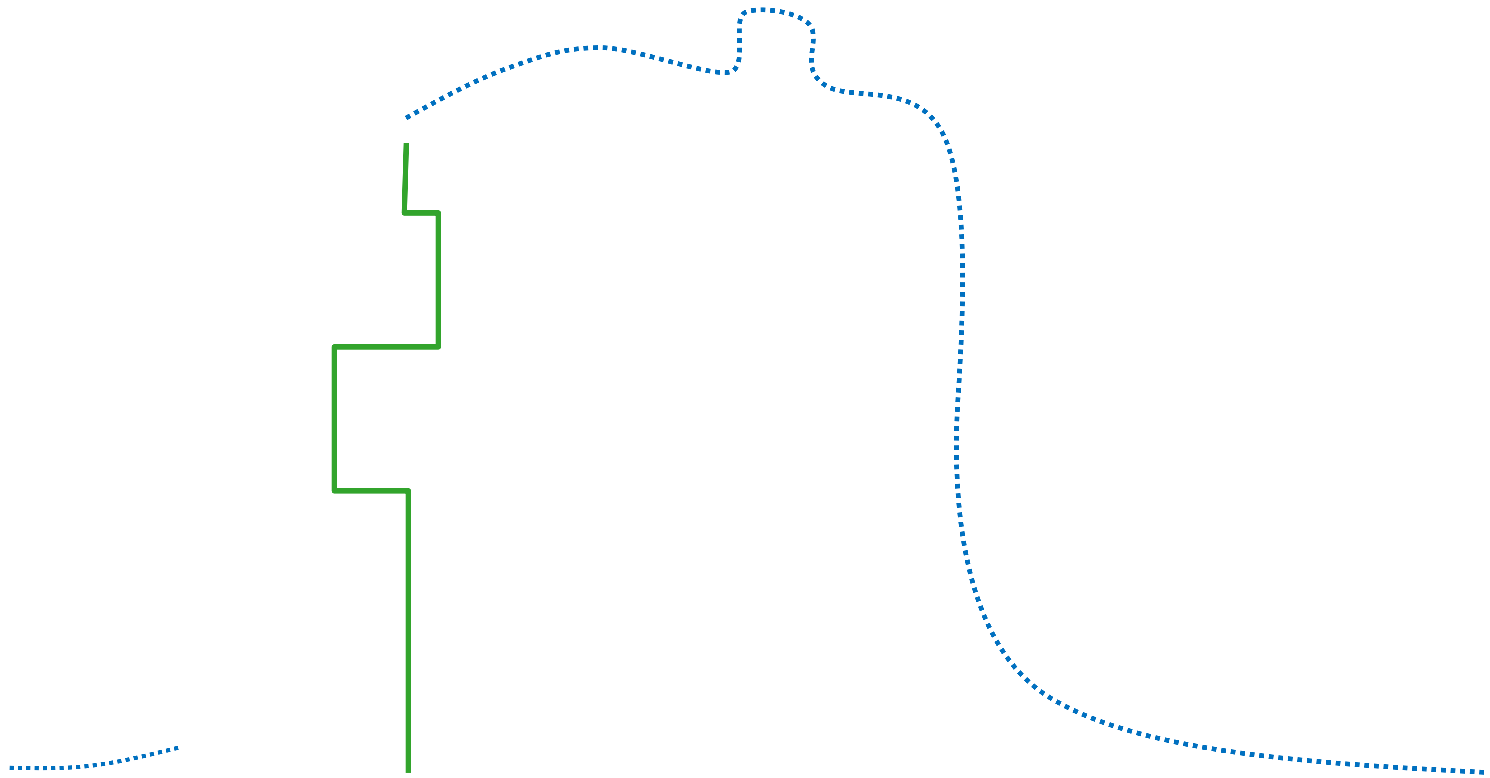
smoothness term

$$\psi(p_i, p_j) = e^{-d_{ij}/\mathrm{med}\, d_{ij}}$$

$$E^b(\mathcal{L}) = \sum_{i:p_i\in\mathcal{P}} \overbrace{E_i^b(l_i)}^{\text{likelihood}} + \lambda_b \sum_{ij} \overbrace{\psi(p_i, p_j)}^{\text{smoothness}} \cdot \mathbb{I}[l_i \neq l_j]$$

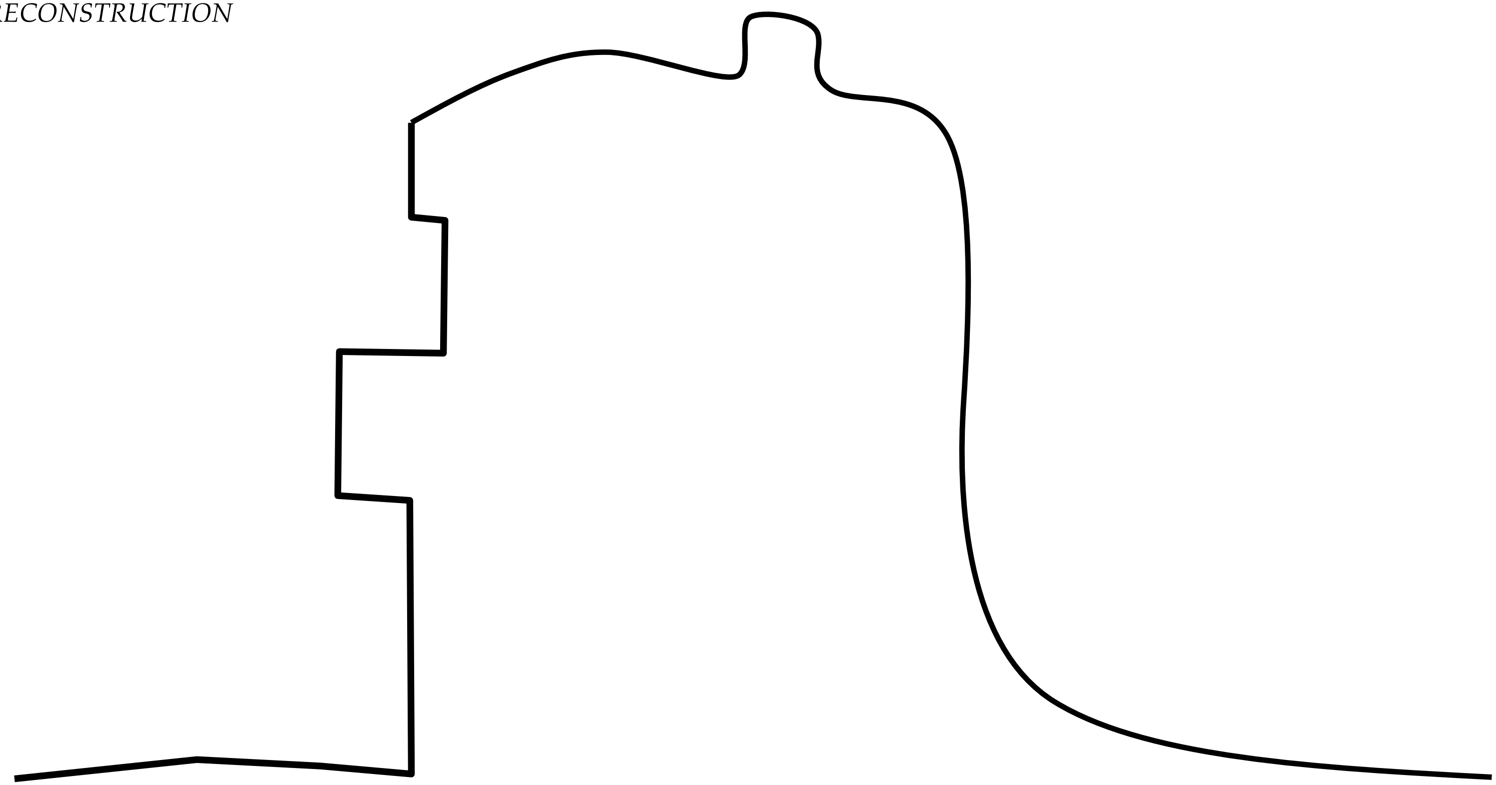
blending energy

Airborne / street-side point cloud blending



Airborne / street-side point cloud blending

3DT-BASED SURFACE RECONSTRUCTION



Data reduction / speed boost for large datasets

(illustration: mesh from airborne-only input)

O(10^7) points (street / aerial)
O(10^8) rays
O(km^2) area

Data reduction / speed boost for large datasets

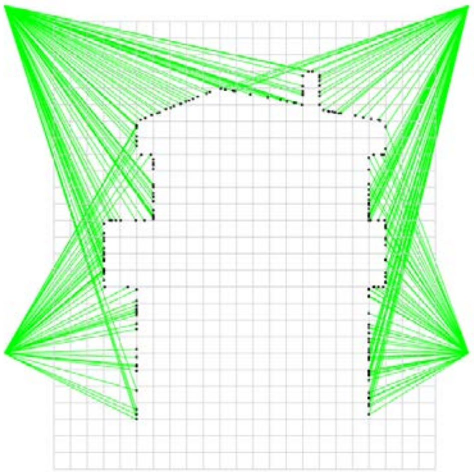
(1) Point decimation

(2) Ray decimation

(3) Ray truncation

Data reduction / speed boost for large datasets

(1) Point decimation



- voxel grid (with hashing)
- point clustering
- centroid prototype
- merge rays
- parameter: voxel size

(2) Ray decimation

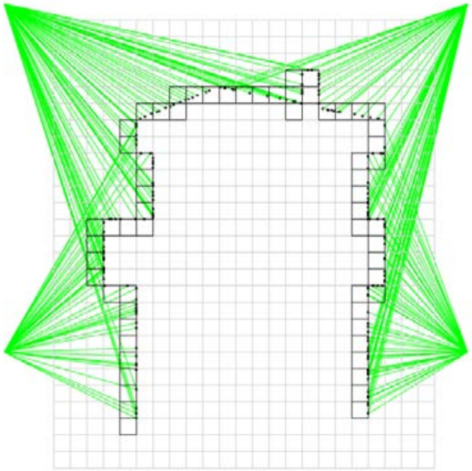
(3) Ray truncation

ETH zürich

CVL Computer Vision Lab

Data reduction / speed boost for large datasets

(1) Point decimation



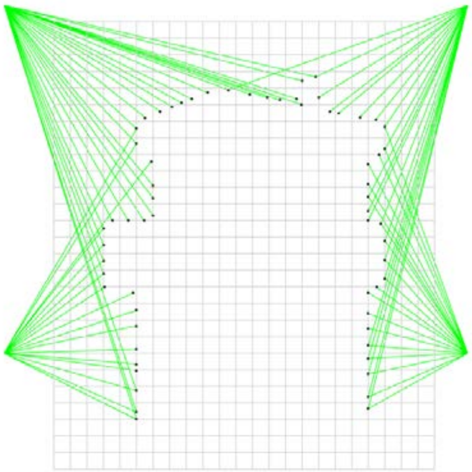
- voxel grid (with hashing)
- point clustering
- centroid prototype
- merge rays
- parameter: voxel size

(2) Ray decimation

(3) Ray truncation

Data reduction / speed boost for large datasets

(1) Point decimation



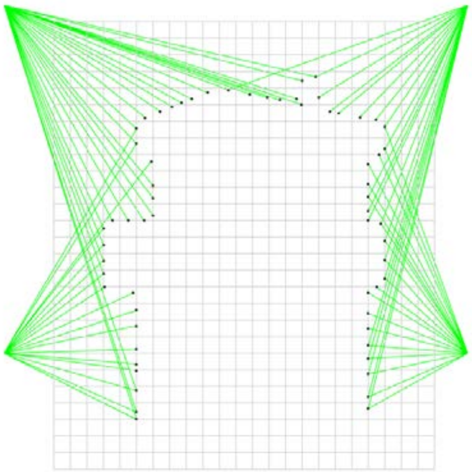
- voxel grid (with hashing)
- point clustering
- centroid prototype
- merge rays
- parameter: voxel size

(2) Ray decimation

(3) Ray truncation

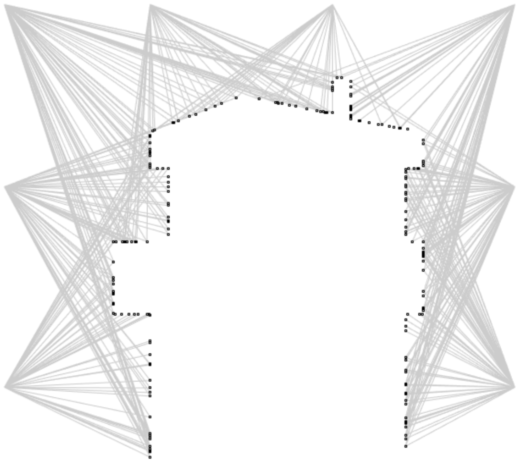
Data reduction / speed boost for large datasets

(1) Point decimation



- voxel grid (with hashing)
- point clustering
- centroid prototype
- merge rays
- parameter: voxel size

(2) Ray decimation

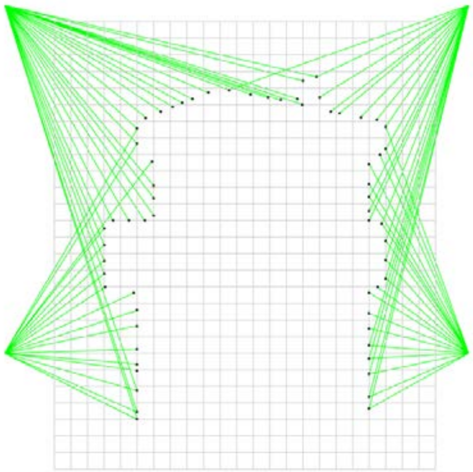


- rays are dense
- keep one per point
- most perpendicular to reduce mistakes

(3) Ray truncation

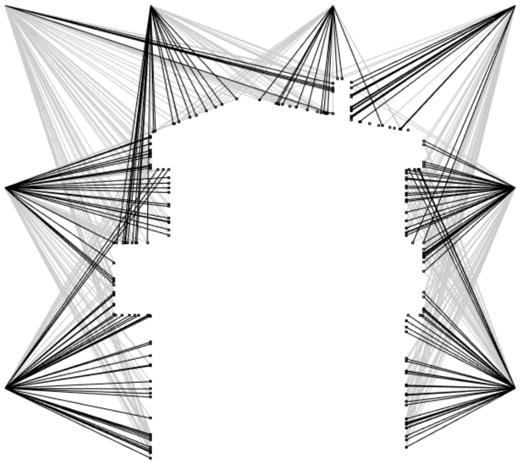
Data reduction / speed boost for large datasets

(1) Point decimation



- voxel grid (with hashing)
- point clustering
- centroid prototype
- merge rays
- parameter: voxel size

(2) Ray decimation

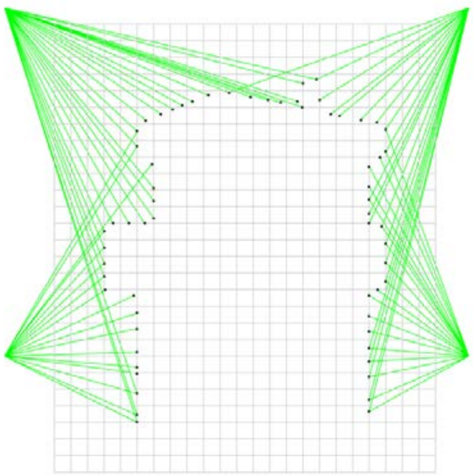


- rays are dense
- keep one per point
- most perpendicular to reduce mistakes

(3) Ray truncation

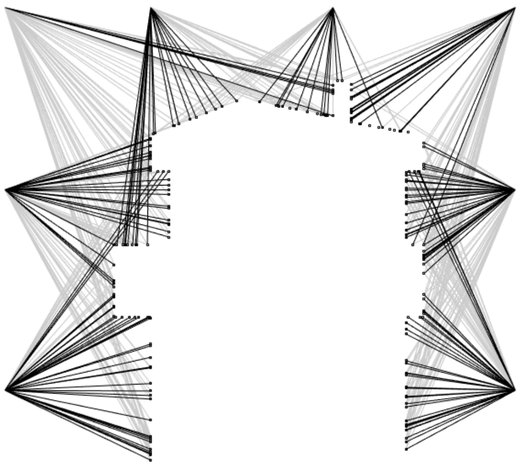
Data reduction / speed boost for large datasets

(1) Point decimation



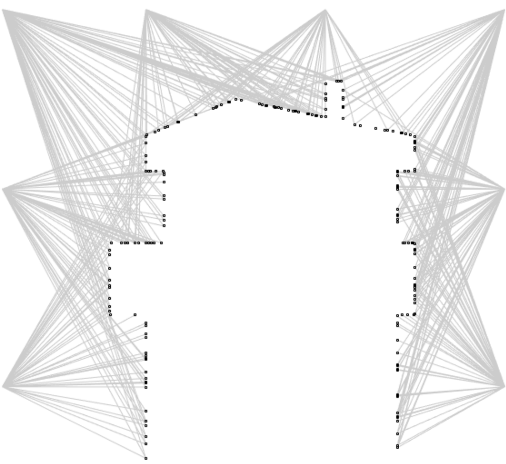
- voxel grid (with hashing)
- point clustering
- centroid prototype
- merge rays
- parameter: voxel size

(2) Ray decimation



- rays are dense
- keep one per point
- most perpendicular to reduce mistakes

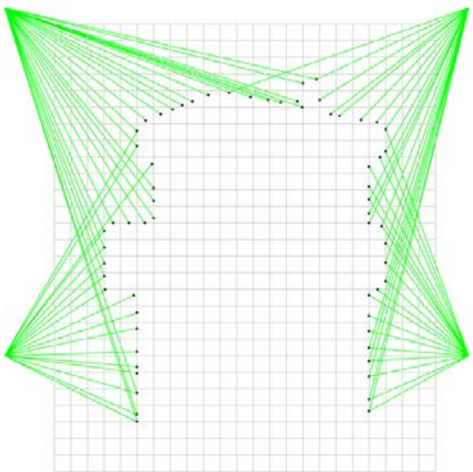
(3) Ray truncation



- similar to limiting penetration depth
- truncate rays at points
- less tetrahedra crossed

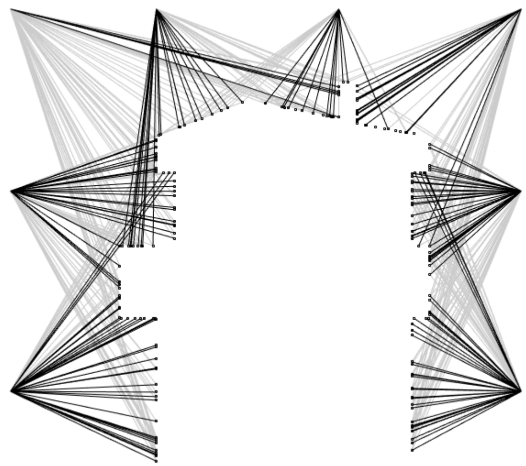
Data reduction / speed boost for large datasets

(1) Point decimation



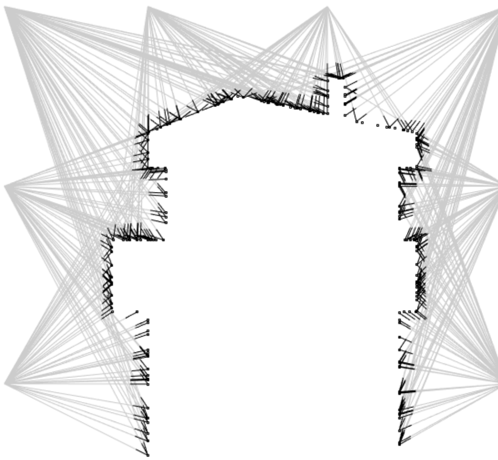
- voxel grid (with hashing)
- point clustering
- centroid prototype
- merge rays
- parameter: voxel size

(2) Ray decimation



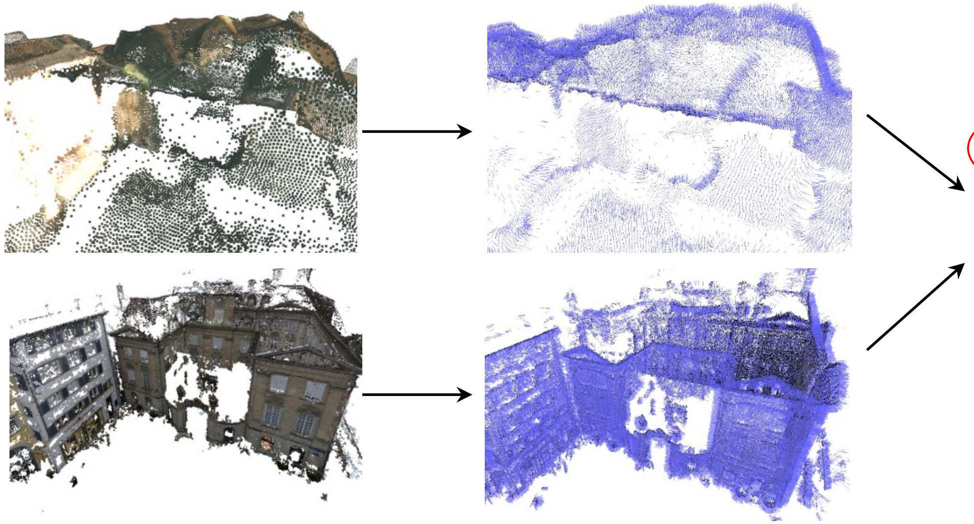
- rays are dense
- keep one per point
- most perpendicular to reduce mistakes

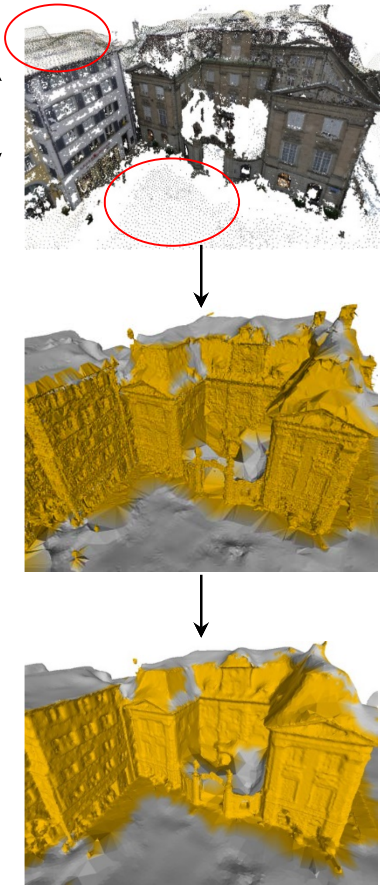
(3) Ray truncation



- similar to limiting penetration depth
- truncate rays at points
- less tetrahedra crossed

Pipeline summary





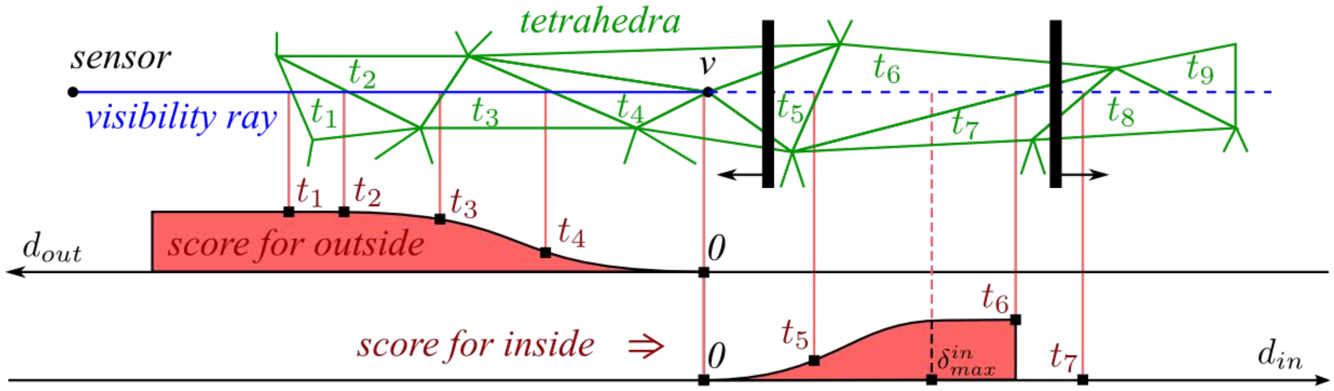
Implementation

- 3DT partitioning (Matlab ~ CGAL)
- Graph cuts (GCoptimization library)
- 3DT adjacencies & ray shooting (C++)

Parameters

Parameter	Symbol	Value
Point neighborhood for normals	k	10
Blending distance control	σ_b	2 m
Blending smoothness	λ_b	1
Inside scoring distance softness	σ_{in}	0.1 m
Outside scoring distance softness	σ_{out}	0.5 m
Ray penetration limit	δ_{in}	$3\sigma_{in}$
Ray truncation distance (optional)	δ_{out}	$3\sigma_{out}$
Inside ray count softness	γ_{in}	2
Outside ray count softness	γ_{out}	2
Smoothness weight (area term)	λ	1–3

fixed in our experiments (applies to rows k through γ_{out})



Datasets

Airborne

- Nadir
- 15 cm GSD
- CapturingReality
- 50 GCPs

Street-side

- Hand held camera
- VisualSfM (sparse) / PMVS (dense)
- registered to the airborne data
  (via IMU/GPS in industrial mobile mapping)

Munsterhof (140x160 m^2)

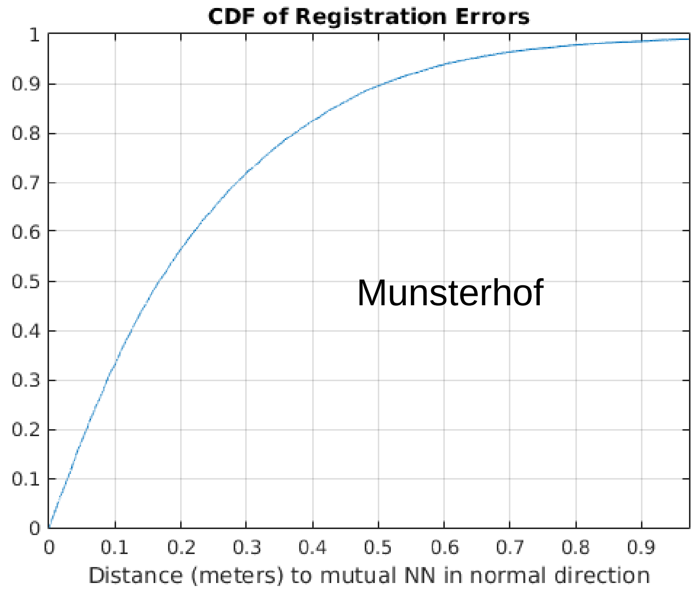
629 street-side images

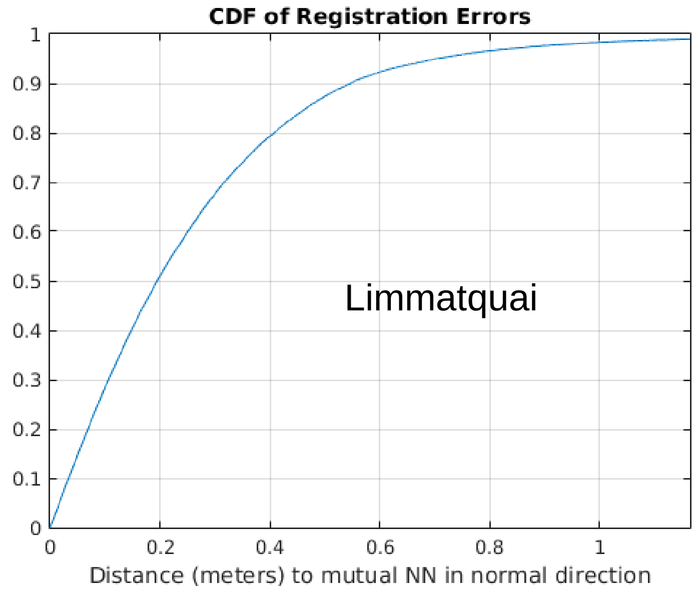
Limmatquai (400x400 m^2)

847 street-side images

Registration accuracy in our experiments

registration errors between airborne and street-side point clouds at overlaps
(distances in normal direction between mutual NNs)

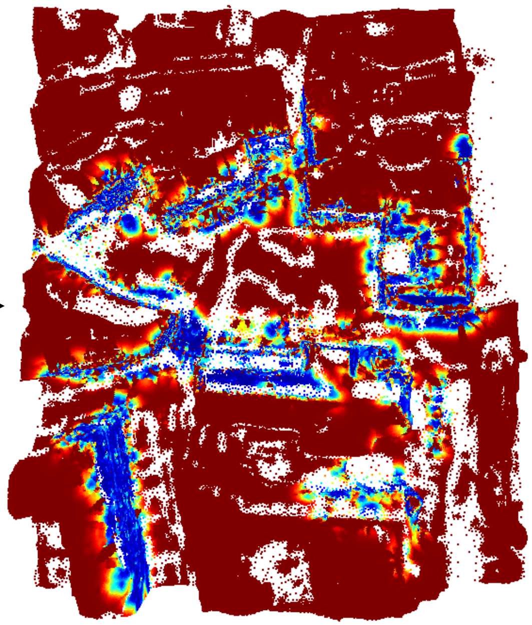




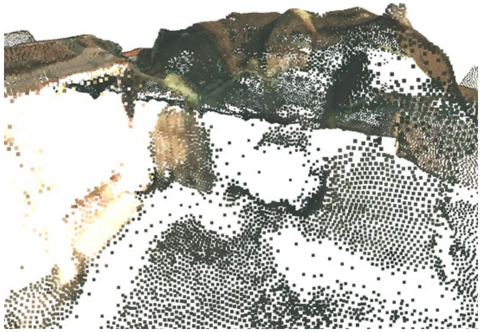
Results: Process on Munsterhof

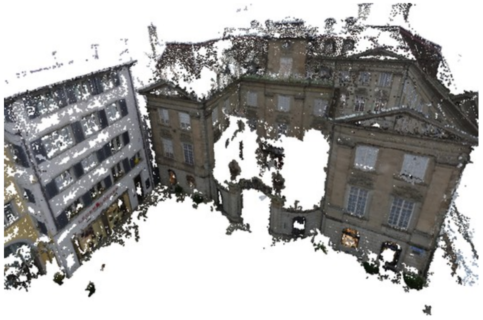
Airborne input (272k points)

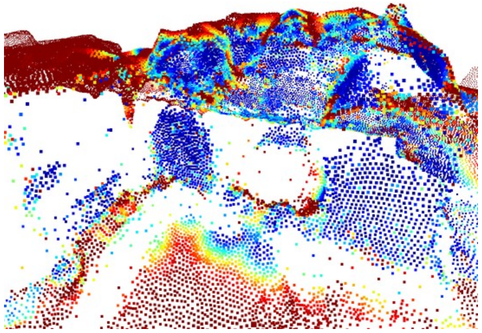
Street-side input (1.5M points, PMVS)



Blending unary energies



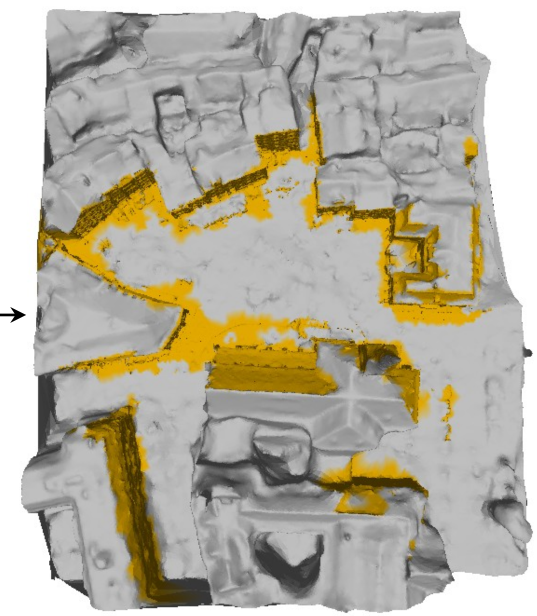




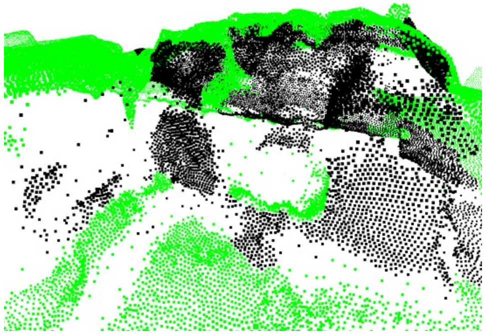
Results: Process on Munsterhof

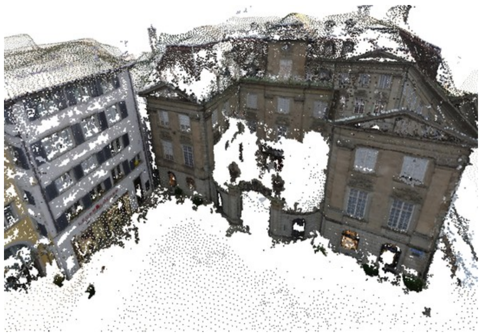
Point cloud segmentation

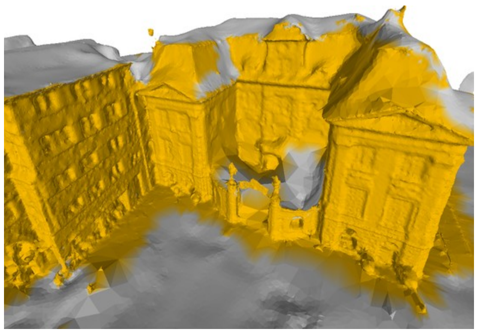
Blended point clouds



Surface reconstruction



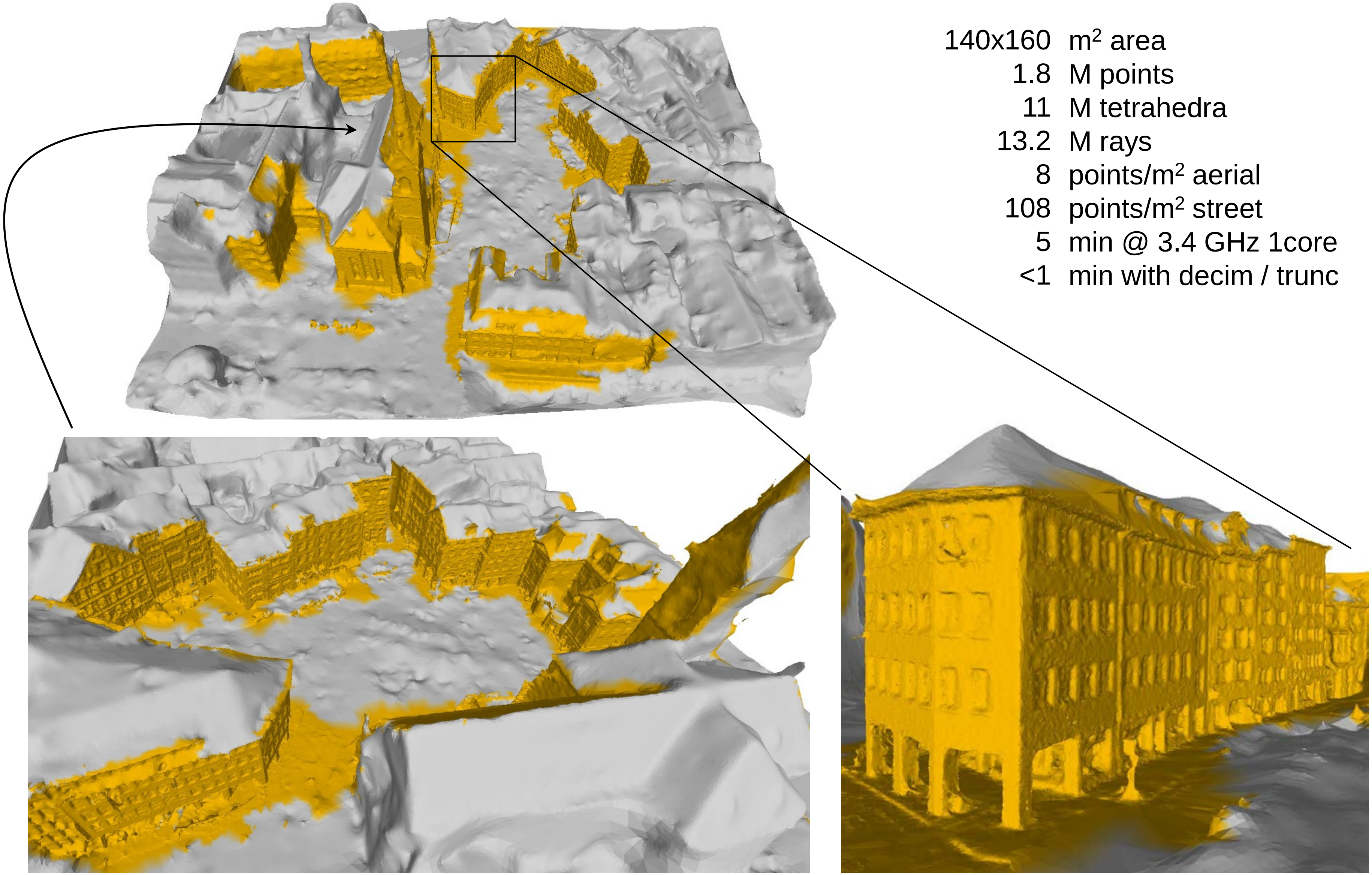




ETHzürich

CVL Computer Vision Lab

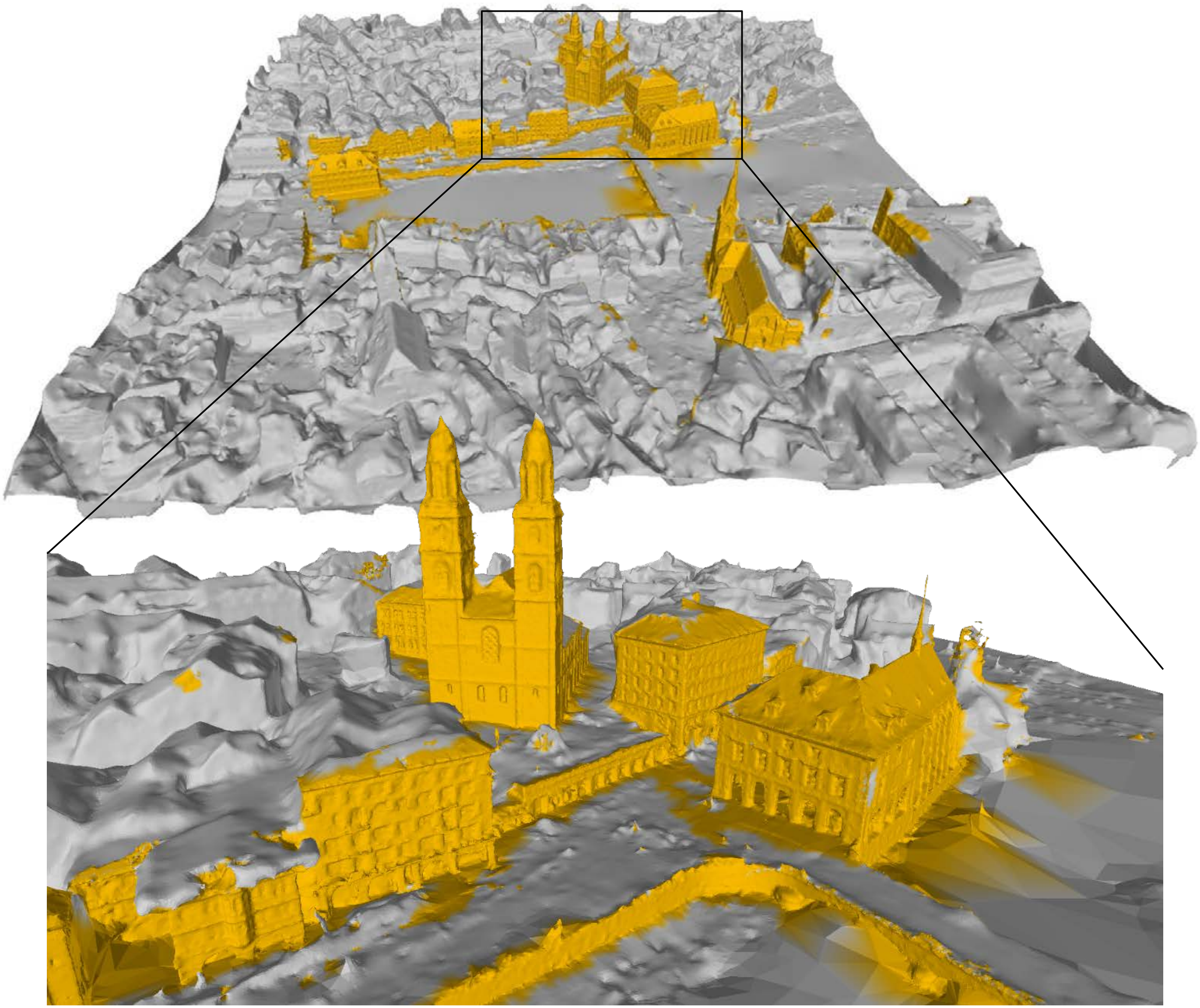
Munsterhof Results



140x160	m^2 area
1.8	M points
11	M tetrahedra
13.2	M rays
8	points/m^2 aerial
108	points/m^2 street
5	min @ 3.4 GHz 1core
<1	min with decim / trunc

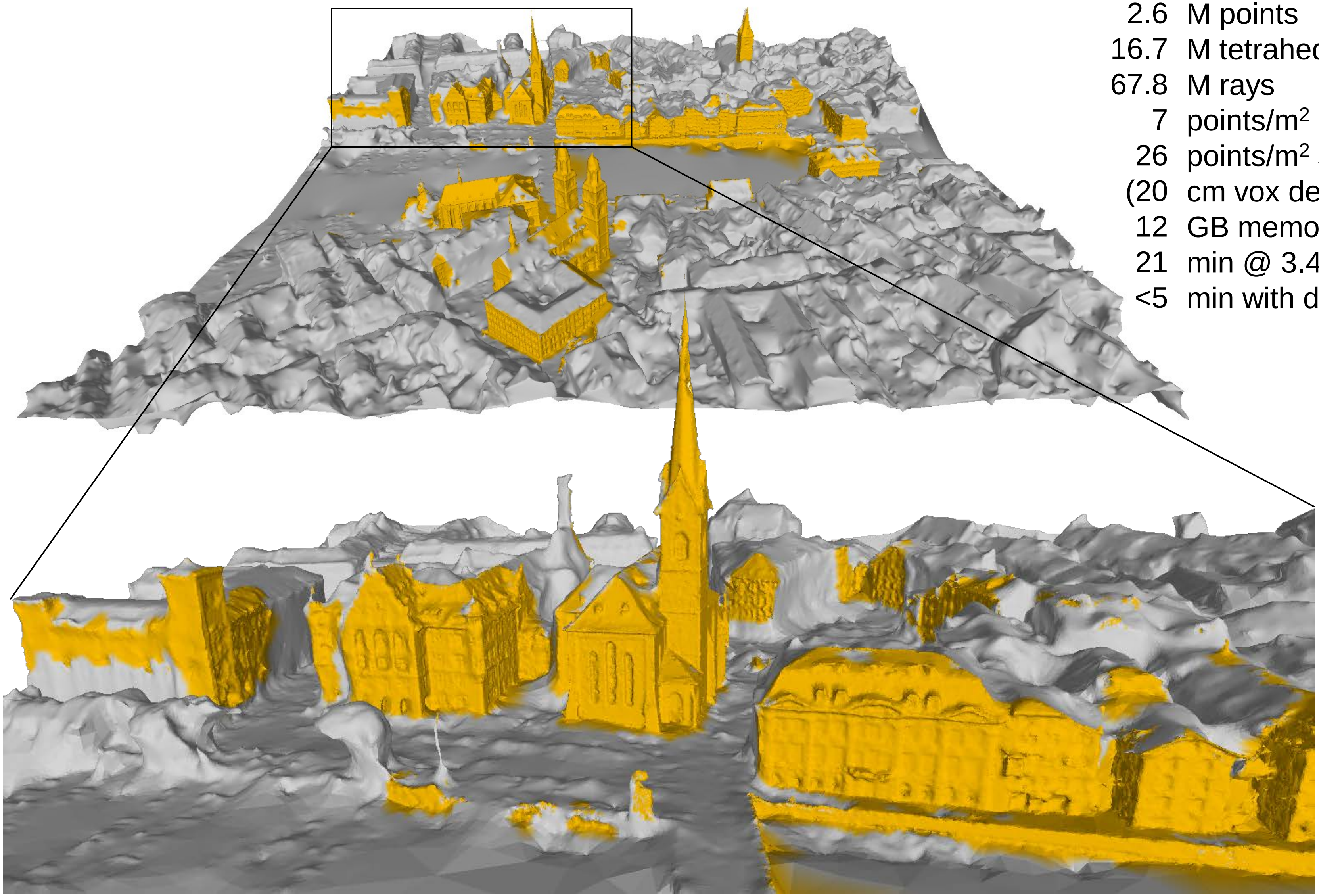
Limmatquai Results

400x400	m^2 area
2.6	M points
16.7	M tetrahedra
67.8	M rays
7	points/m^2 aerial
26	points/m^2 street
(20	cm vox decimation)
12	GB memory
21	min @ 3.4 GHz 1core
<5	min with decim / trunc



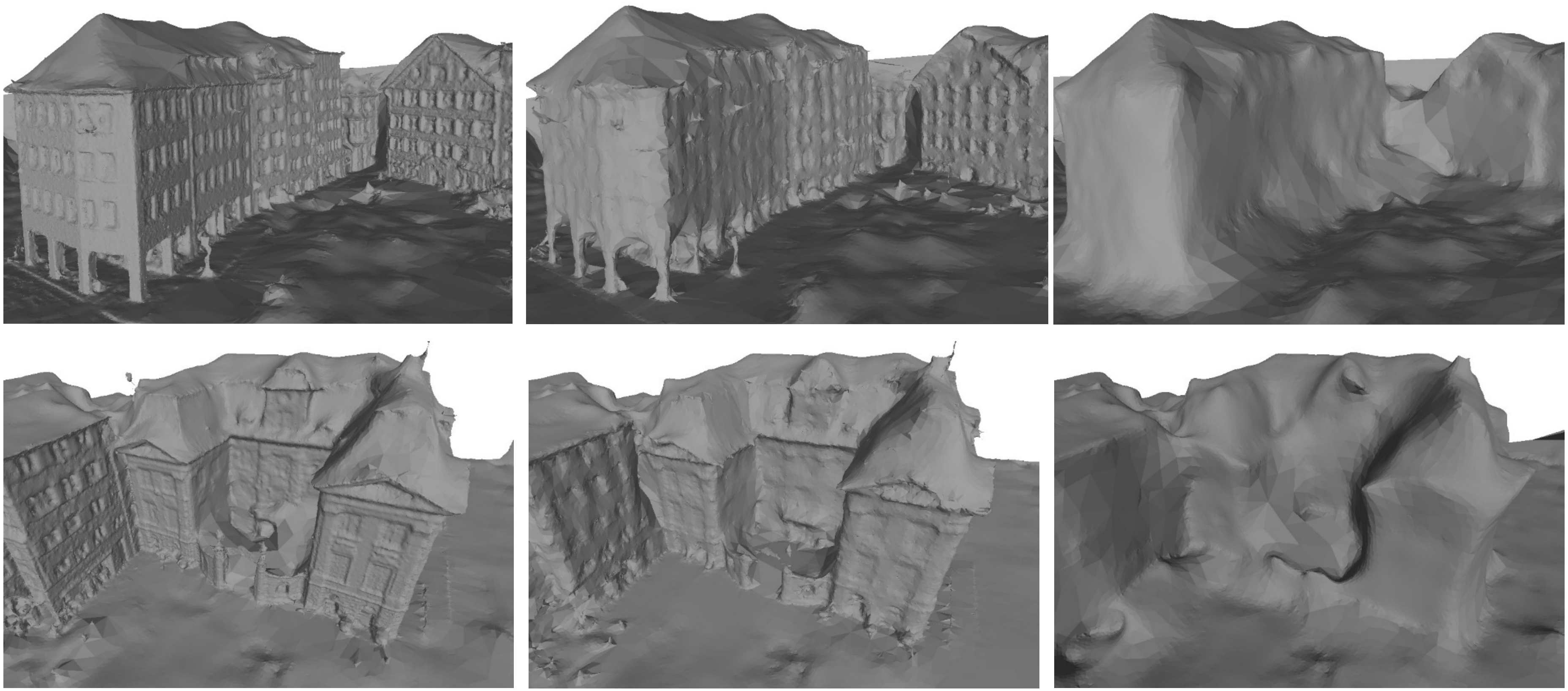
Limmatquai Results

400x400	m^2 area
2.6	M points
16.7	M tetrahedra
67.8	M rays
7	points/m^2 aerial
26	points/m^2 street
(20	cm vox decimation)
12	GB memory
21	min @ 3.4 GHz 1core
<5	min with decim / trunc



Results on different input types

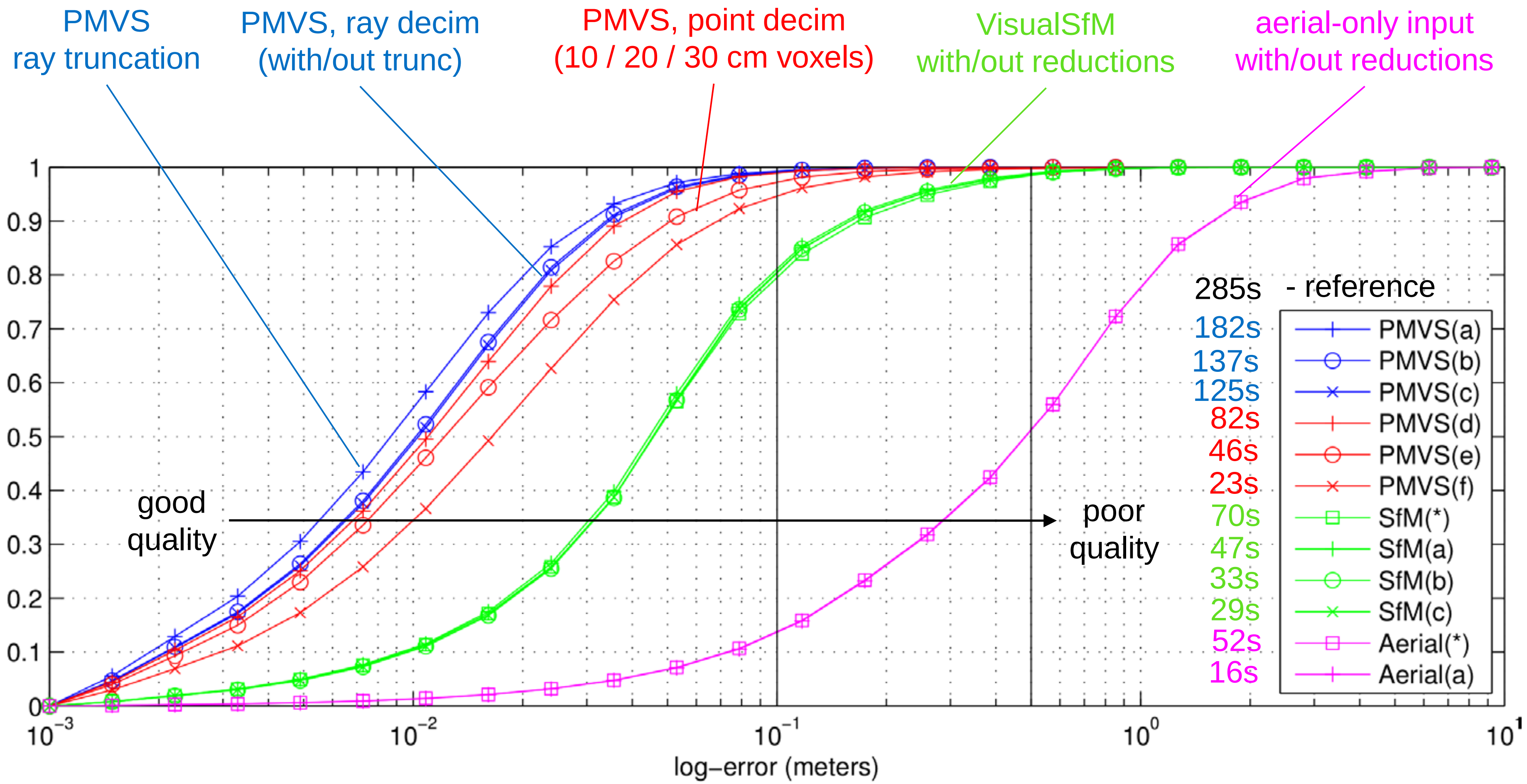
Fusion aerial + PMVS | Fusion aerial + VisualSfM | Airborne only



~108 points/m^2 | ~19 points/m^2 | ~8 points/m^2

Cumulated histogram of mesh errors (vs runtimes)

reference: our highest-density fused mesh (Munsterhof)
timings: on a single 3.4 GHz CPU core



Numerical details (see paper)

Parameter variations (others fixed)

Datasets

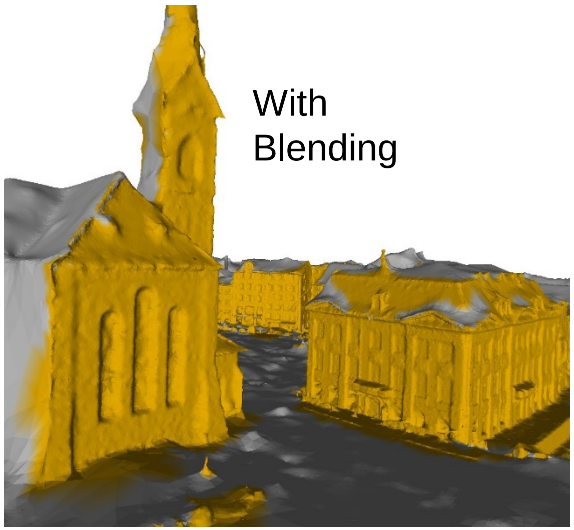
Detailed statistics

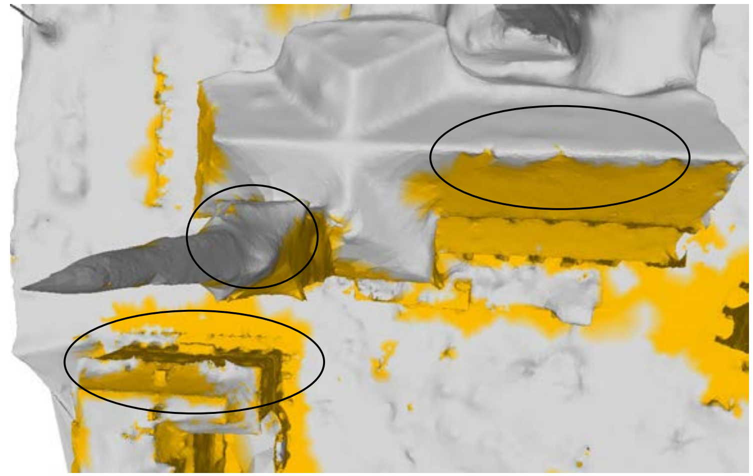
Accuracy measures

Timings

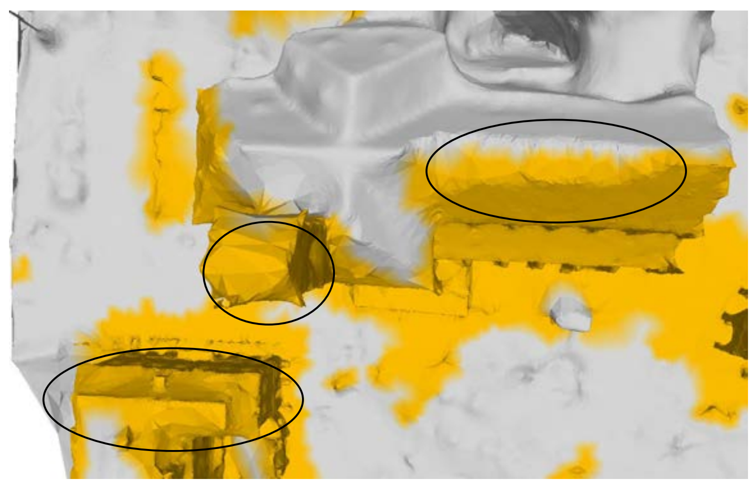
		Params			Airborne data			Street-side data			3DT & Ray shooting			Mesh	Mesh distance from ref (cm)				Timings			
	Experiment	vox	tr	λ	#pts	#/m^2	#vis	#pts	#/m^2	#vis	#verts	#tets	#rays	#verts	mean^a	mean^s	10cm^s	50cm^s	tet	ray	gco	total
Münsterhof	PMVS (*)	0		3	272k	8.0	11.9	1.54M	108.0	6.8	1.75M	11.0M	13.2M	1.44M	(ref)	(ref)	(ref)	(ref)	32s	182s	27s	285s
	PMVS (a)	0	✓	3	272k	7.9	11.9	1.54M	106.8	6.8	1.75M	11.0M	13.2M	1.43M	2.9cm	1.3cm	0.7%	0.0%	32s	**80s**	28s	**182s**
	PMVS (b)	0		3	272k	8.4	**1.0**	1.54M	108.9	**1.0**	1.75M	11.0M	**1.75M**	1.24M	6.7cm	1.5cm	0.8%	0.0%	32s	**28s**	**34s**	**137s**
	PMVS (c)	0	✓	3	272k	8.5	**1.0**	1.54M	108.5	**1.0**	1.75M	11.0M	**1.75M**	1.23M	8.6cm	1.6cm	0.9%	0.0%	32s	**15s**	**34s**	**125s**
	PMVS (d)	0.10		3	272k	8.4	**1.0**	**780k**	**56.3**	**1.0**	**996k**	**6.27M**	**996k**	**756k**	6.8cm	1.7cm	1.0%	0.0%	**18s**	**16s**	**24s**	**82s**
	PMVS (e)	0.20		3	266k	8.2	**1.0**	**310k**	**23.5**	**1.0**	**522k**	**3.33M**	**522k**	**416k**	7.1cm	2.2cm	2.5%	0.1%	**9.4s**	**7.9s**	**16s**	**46s**
	PMVS (f)	0.35		1	**213k**	**6.3**	**1.0**	**127k**	**6.3**	**1.0**	**293k**	**1.89M**	**293k**	**262k**	3.0cm	3.1cm	5.1%	0.3%	**5.2s**	**4.2s**	**6.0s**	**23s**
	SfM (*)	0		1	272k	7.8	11.9	233k	18.6	4.6	452k	2.84M	3.91M	410k	3.6cm	7.7cm	19.8%	1.4%	8.0s	45s	7.6s	70s
	SfM (a)	0	✓	1	272k	7.6	11.9	233k	18.7	4.6	452k	2.84M	3.91M	412k	3.6cm	7.1cm	18.1%	1.1%	9.8s	**19s**	6.5s	**47s**
	SfM (b)	0		1	272k	7.8	**1.0**	233k	18.7	**1.0**	452k	2.84M	**452k**	**382k**	3.6cm	7.5cm	18.9%	1.3%	8.1s	**6.4s**	8.4s	**33s**
	SfM (c)	0	✓	1	272k	7.7	**1.0**	233k	19.0	**1.0**	452k	2.84M	**452k**	**383k**	3.5cm	7.5cm	18.7%	1.3%	8.5s	**3.2s**	8.4s	**29s**
	aerial (*)	0		1	272k	6.2	11.9	not used			272k	1.77M	3.24M	269k	3.1cm	69cm	86.4%	49.1%	5.0s	41s	3.3s	52s
	aerial (a)	0	✓	1	272k	6.3	**1.0**	not used			272k	1.77M	**272k**	**253k**	3.1cm	70cm	86.5%	49.2%	**7.3s**	**2.0s**	**4.0s**	**16s**
Limmatquai	PMVS (*)	0.20		3	1.65M	6.8	11.5	1.14M	26.0	43.6	2.60M	16.7M	67.8M	2.41M	(ref)	(ref)	(ref)	(ref)	49s	1059s	53s	1261s
	PMVS (a)	0.20	✓	3	1.65M	6.7	11.5	1.14M	26.3	43.6	2.60M	16.7M	67.8M	2.45M	0.2cm	0.3cm	0.9%	0.1%	51s	**314s**	53s	**517s**
	PMVS (b)	0.20		3	1.65M	6.9	**1.0**	1.14M	26.5	**1.0**	2.60M	16.7M	2.60M	**2.11M**	3.7cm	2.6cm	8.1%	0.3%	49s	**50s**	**79s**	**277s**
	PMVS (c)	0.20	✓	3	1.65M	6.8	**1.0**	1.14M	26.9	**1.0**	2.60M	16.7M	2.60M	**2.13M**	3.8cm	3.0cm	8.9%	0.4%	50s	**19s**	**78s**	**248s**
	SfM (*)	0		1	1.68M	6.5	11.5	259k	9.1	4.6	1.79M	11.5M	19.6M	1.72M	0.3cm	20cm	44.4%	8.9%	35s	262s	35s	380s
	SfM (a)	0	✓	1	1.68M	6.5	11.5	259k	9.2	4.6	1.79M	11.5M	19.6M	1.73M	0.3cm	19cm	42.4%	7.7%	35s	**95s**	35s	**213s**
	SfM (b)	0		1	1.68M	6.6	**1.0**	259k	9.0	**1.0**	1.79M	11.5M	**1.79M**	1.62M	0.6cm	19cm	44.0%	8.4%	34s	**30s**	39s	**153s**
	SfM (c)	0	✓	1	1.68M	6.5	**1.0**	259k	9.5	**1.0**	1.79M	11.5M	**1.79M**	1.62M	0.6cm	21cm	44.0%	8.8%	34s	**12s**	39s	**134s**
	aerial (*)	0		1	1.68M	6.0	11.5	not used			1.68M	10.9M	19.3M	1.66M	0.0cm	81cm	88%	47.3%	33s	269s	33s	340s
	aerial (a)	0	✓	1	1.68M	6.0	**1.0**	not used			1.68M	10.9M	**1.68M**	1.57M	0.3cm	81cm	88%	47.4%	33s	**12s**	**26s**	**103s**

Qualitative effect of prior blending with normals





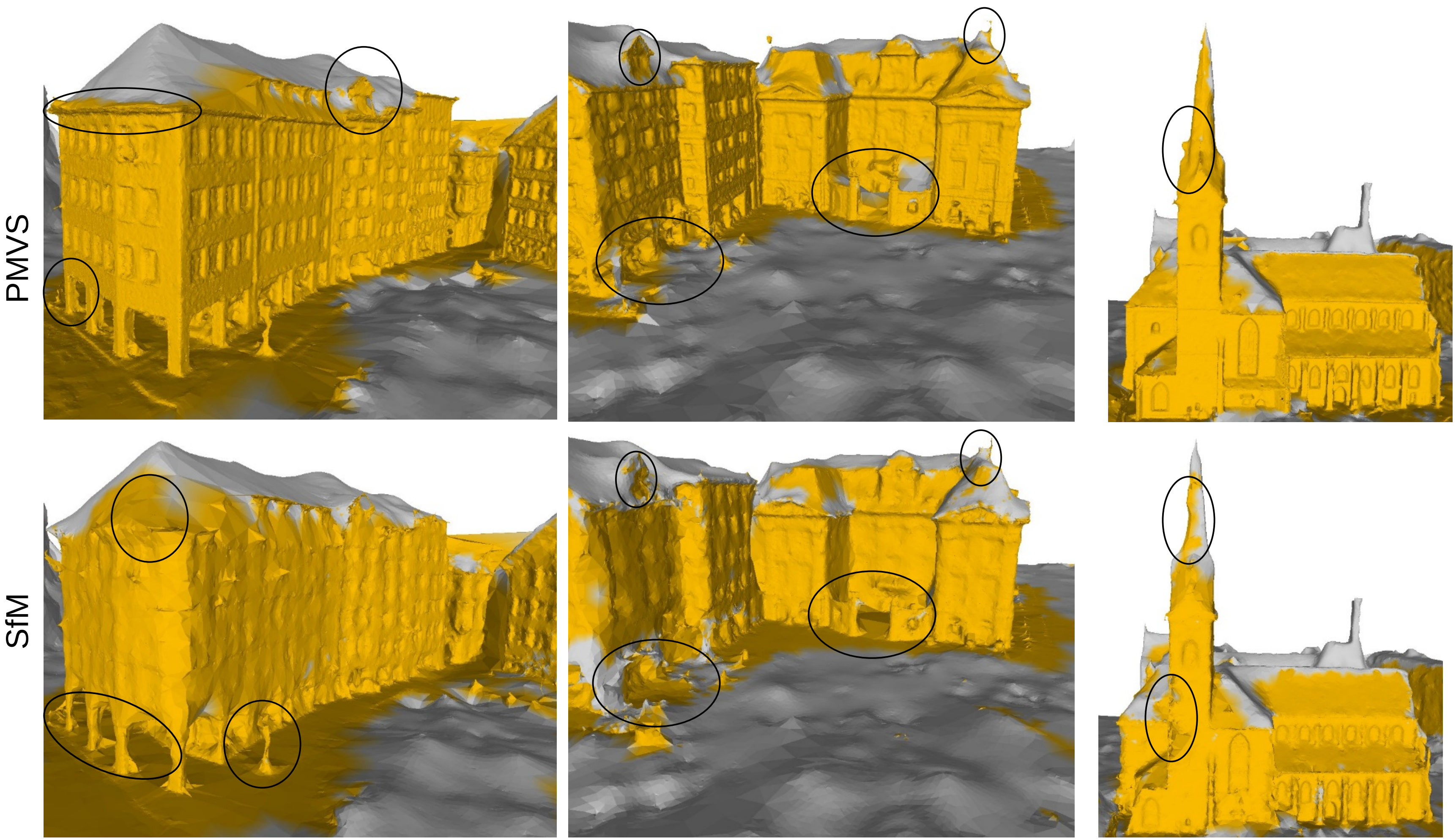
With Normals



Without Normals
(Distance Only)

Qualitative effect of ray decimation and truncation

No reductions

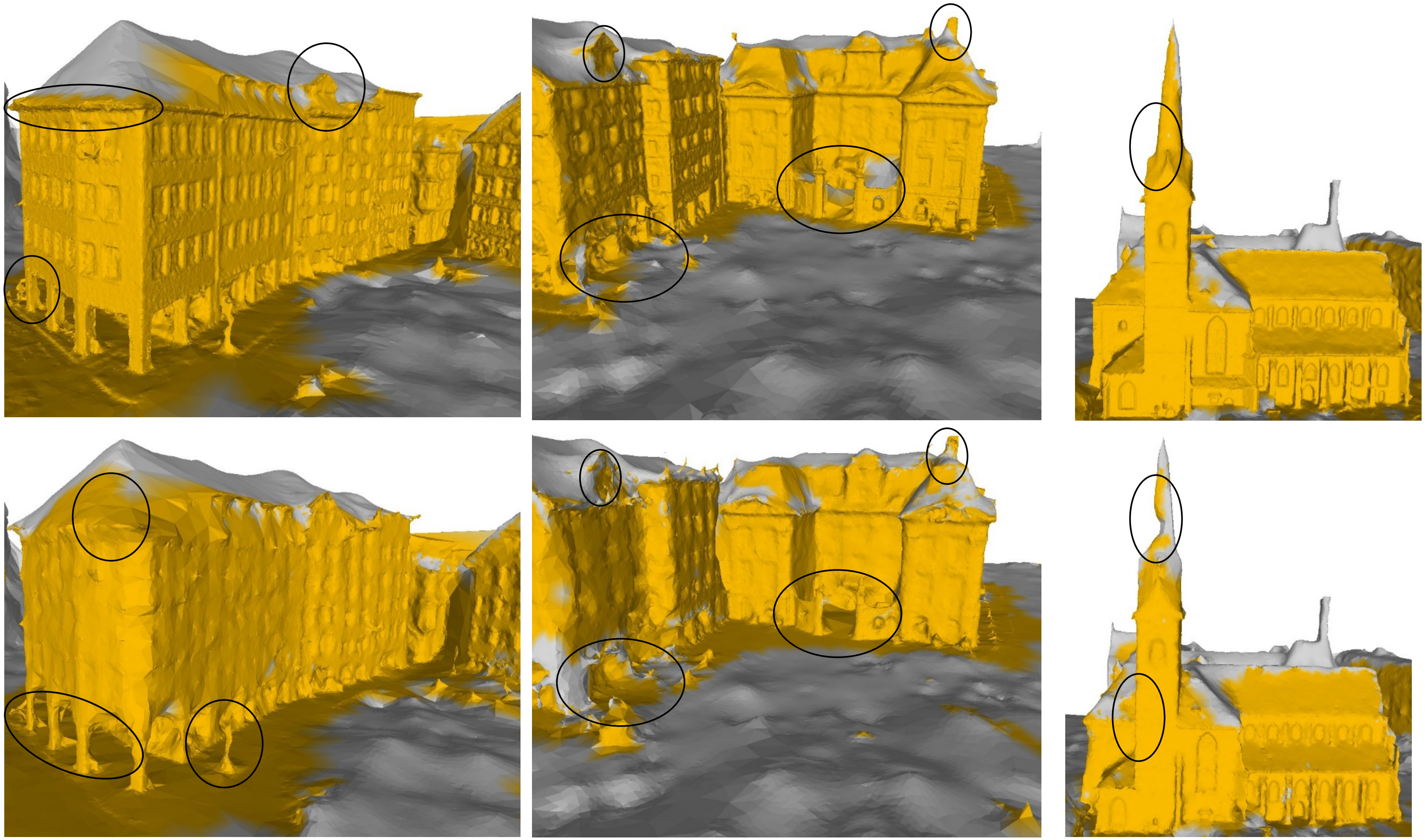


Qualitative effect of ray decimation and truncation

Ray decimation

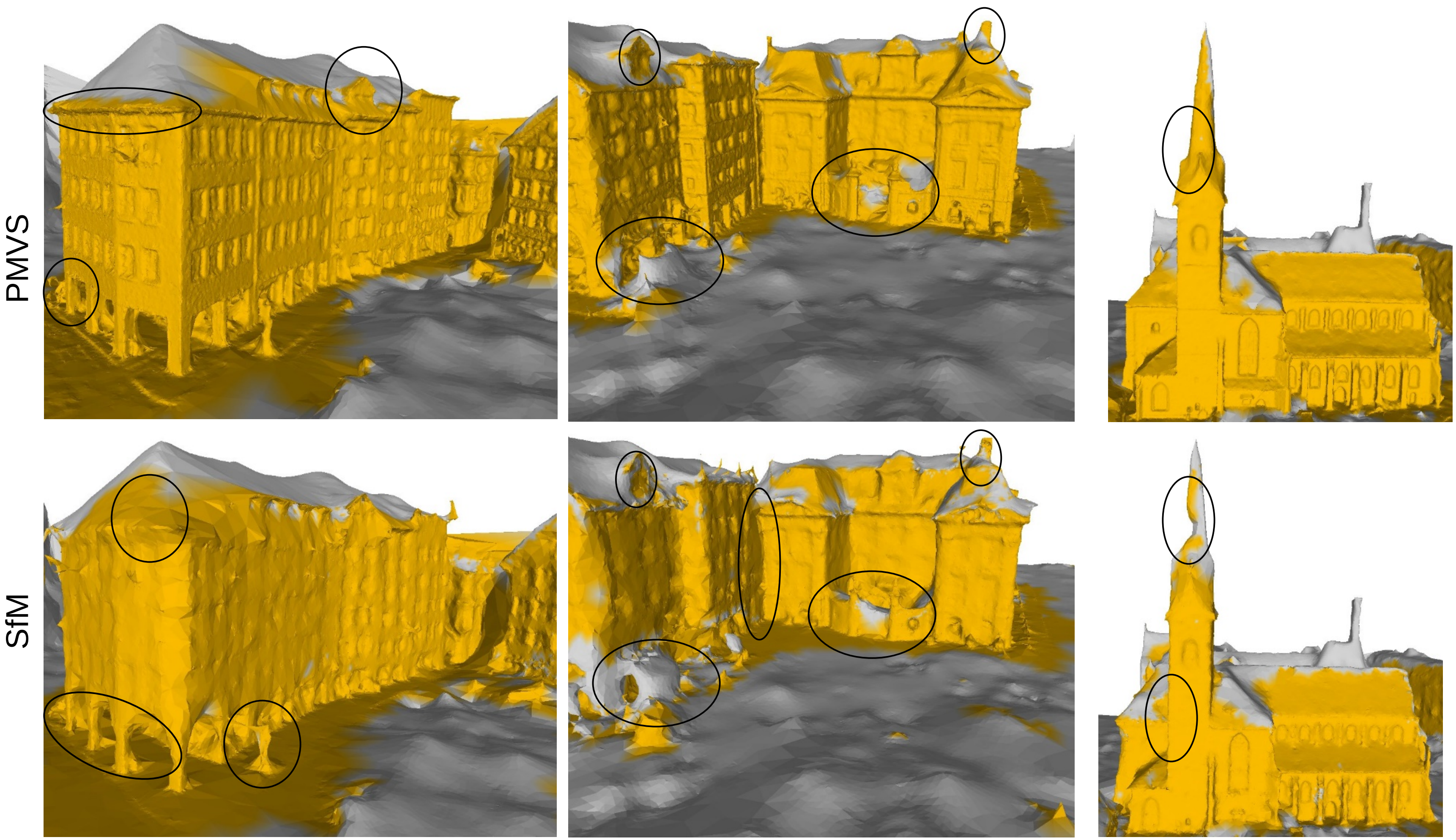
PMVS

SfM

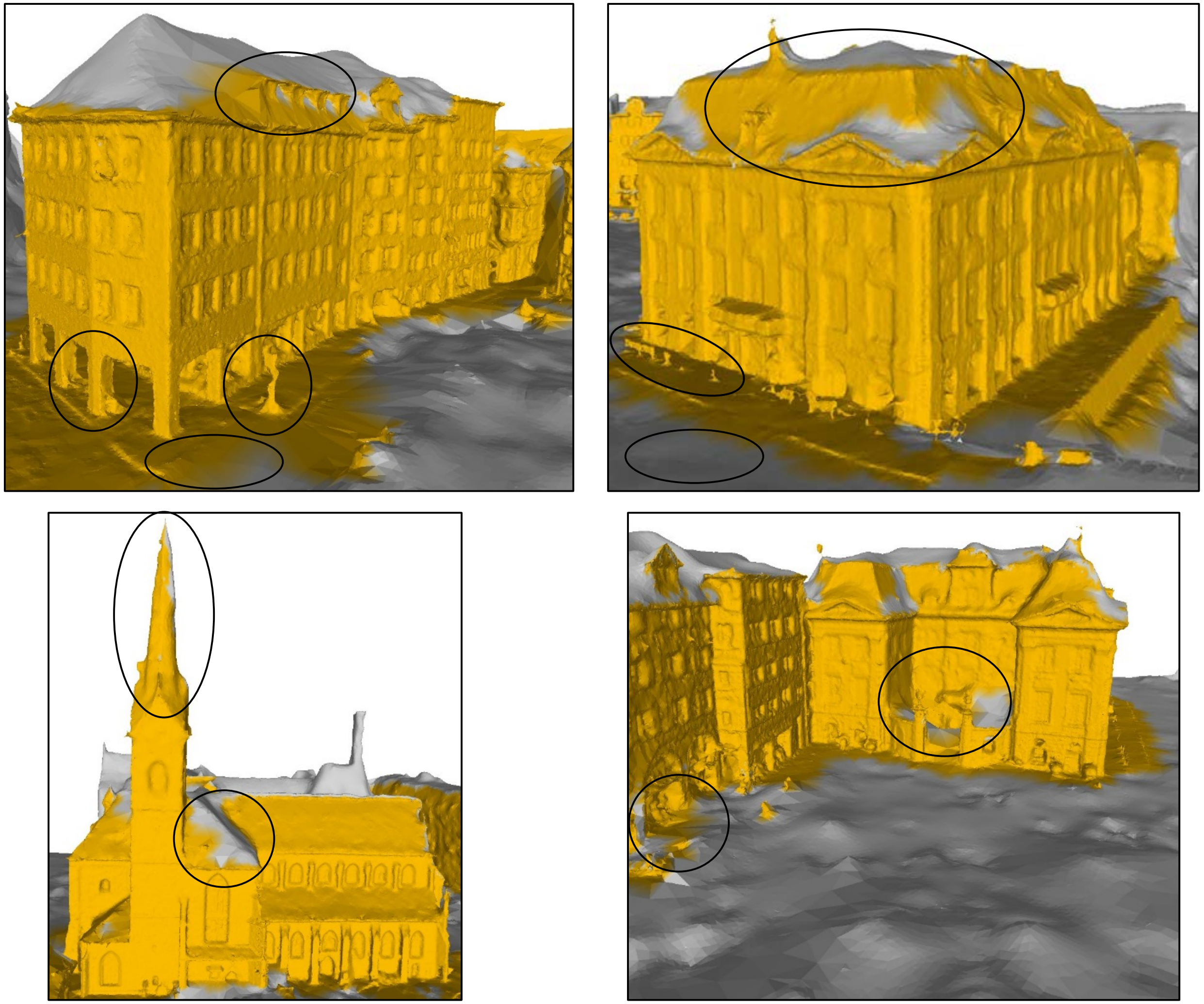


Qualitative effect of ray decimation and truncation

Ray decimation and truncation



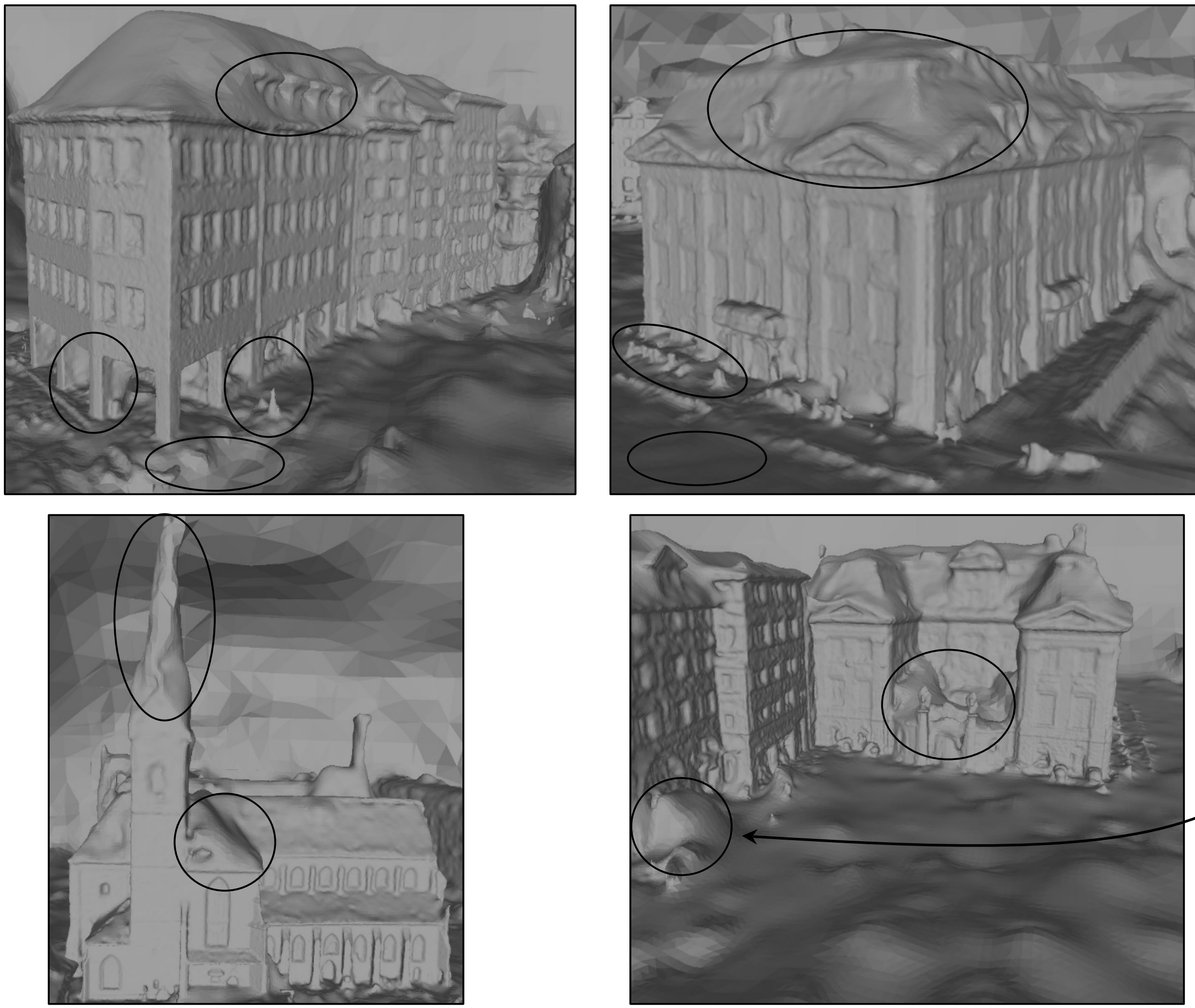
Quick Comparison to Poisson Surface Reconstruction



Ours

- PMVS+aerial
- after blending
- full rays

Quick Comparison to Poisson Surface Reconstruction



Poisson

- PMVS+aerial
- after blending
- Depth 12, Div 8
- no rays used
- Inflated / smoother
- ground collapses
- narrow struct lost
- aerial artifacts remain

Conclusions

- *3DT-fusion of airborne & street-side data*
- *Point cloud blending against gross ray conflicts*
- *Reduction techniques for large urban scenes*
- *Detailed runtime vs. quality experimentation*
- *Complete & detailed (LoD-3) models in minutes / km^2*

erc *varcity.eu*

Efficient Volumetric Fusion of Airborne and Street-Side Data for Urban Reconstruction

András Bódis-Szomorú, Hayko Riemenschneider, Luc Van Gool

Thank you!