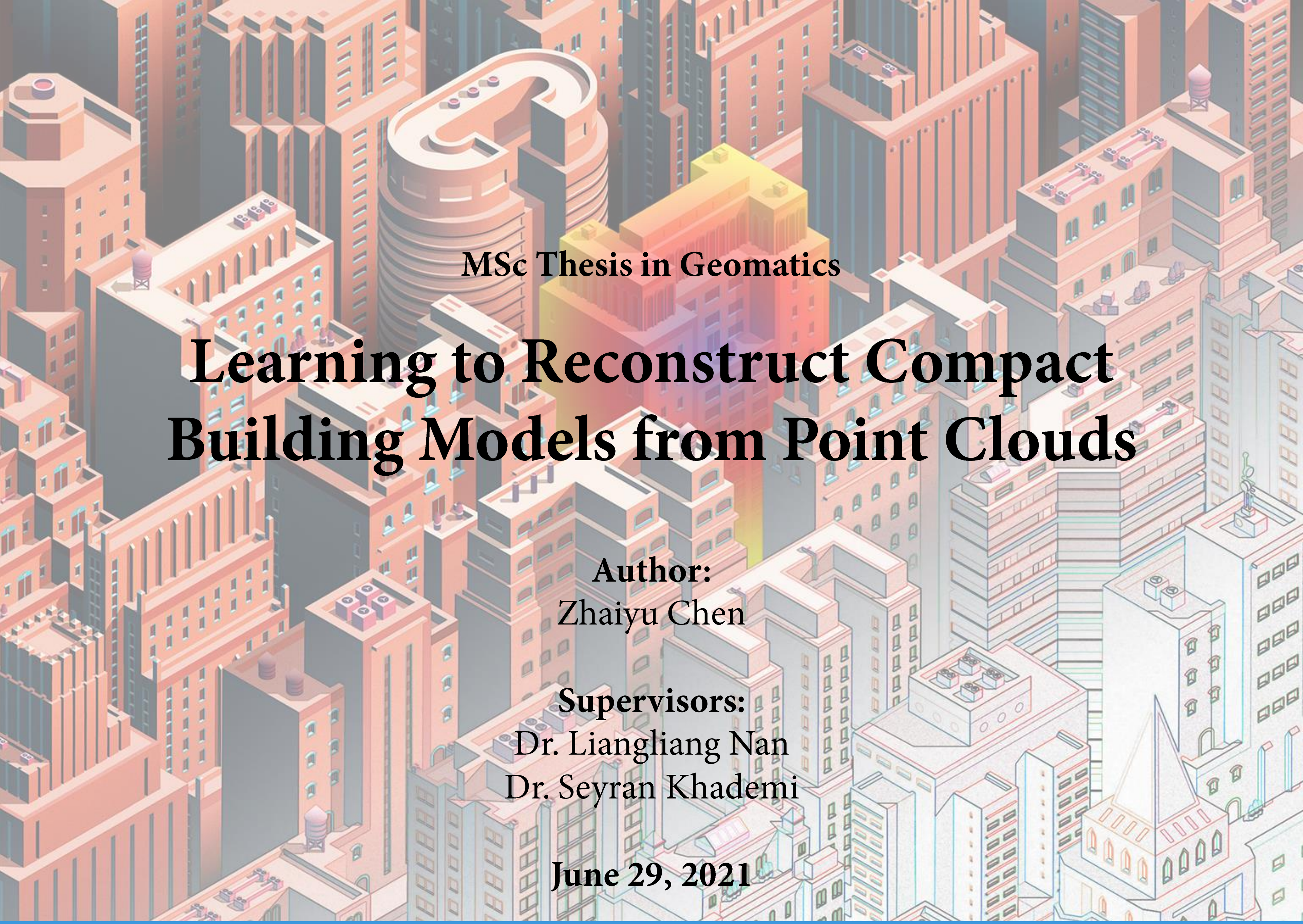


MSc Thesis in Geomatics

Learning to Reconstruct Compact Building Models from Point Clouds

Author:
Zhaiyu Chen

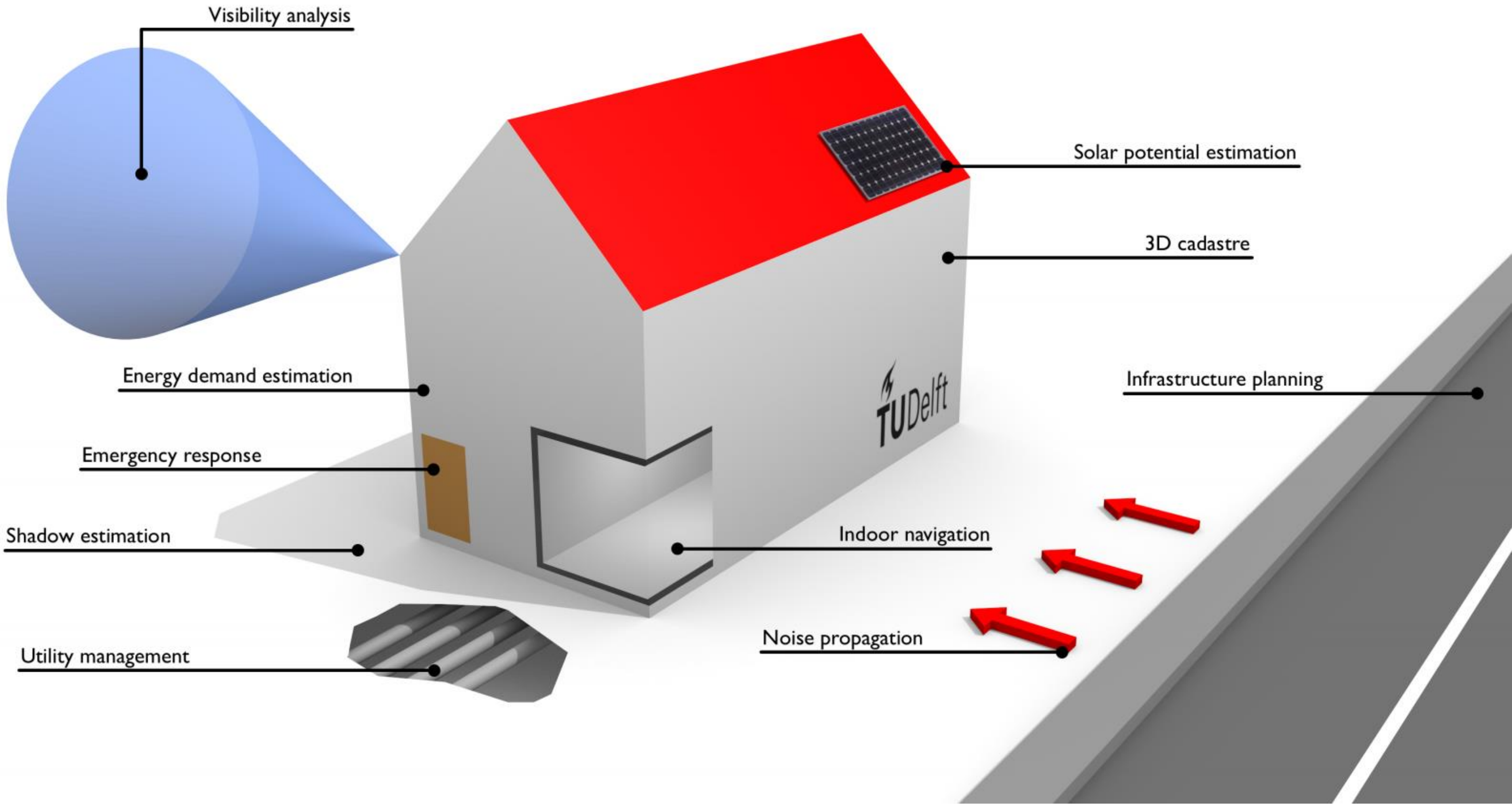
Supervisors:
Dr. Liangliang Nan
Dr. Seyran Khademi

June 29, 2021

- **Introduction**
- **Related work**
- **Methodology**
- **Datasets**
- **Results and discussion**
- **Conclusions**

- **Introduction**
- **Related work**
- **Methodology**
- **Datasets**
- **Results and discussion**
- **Conclusions**

Introduction: 3D building models

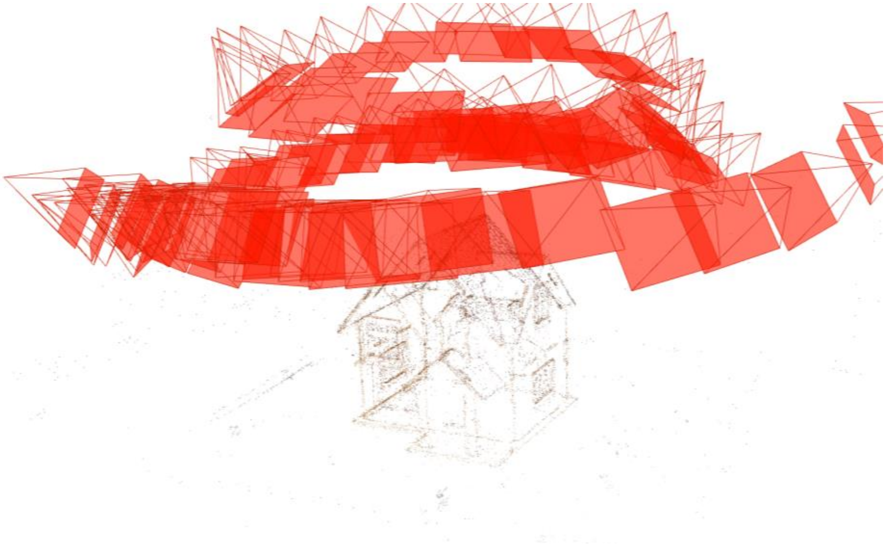


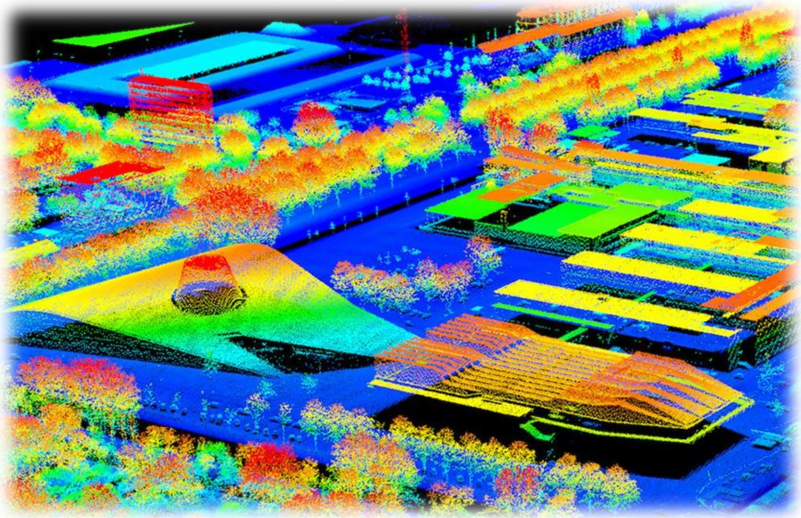
Applications of 3D building models [Biljecki et al., 2015]

Introduction: Point clouds

Acquisition of a point cloud

- Photogrammetry
- LiDAR (Light Detection and Ranging)

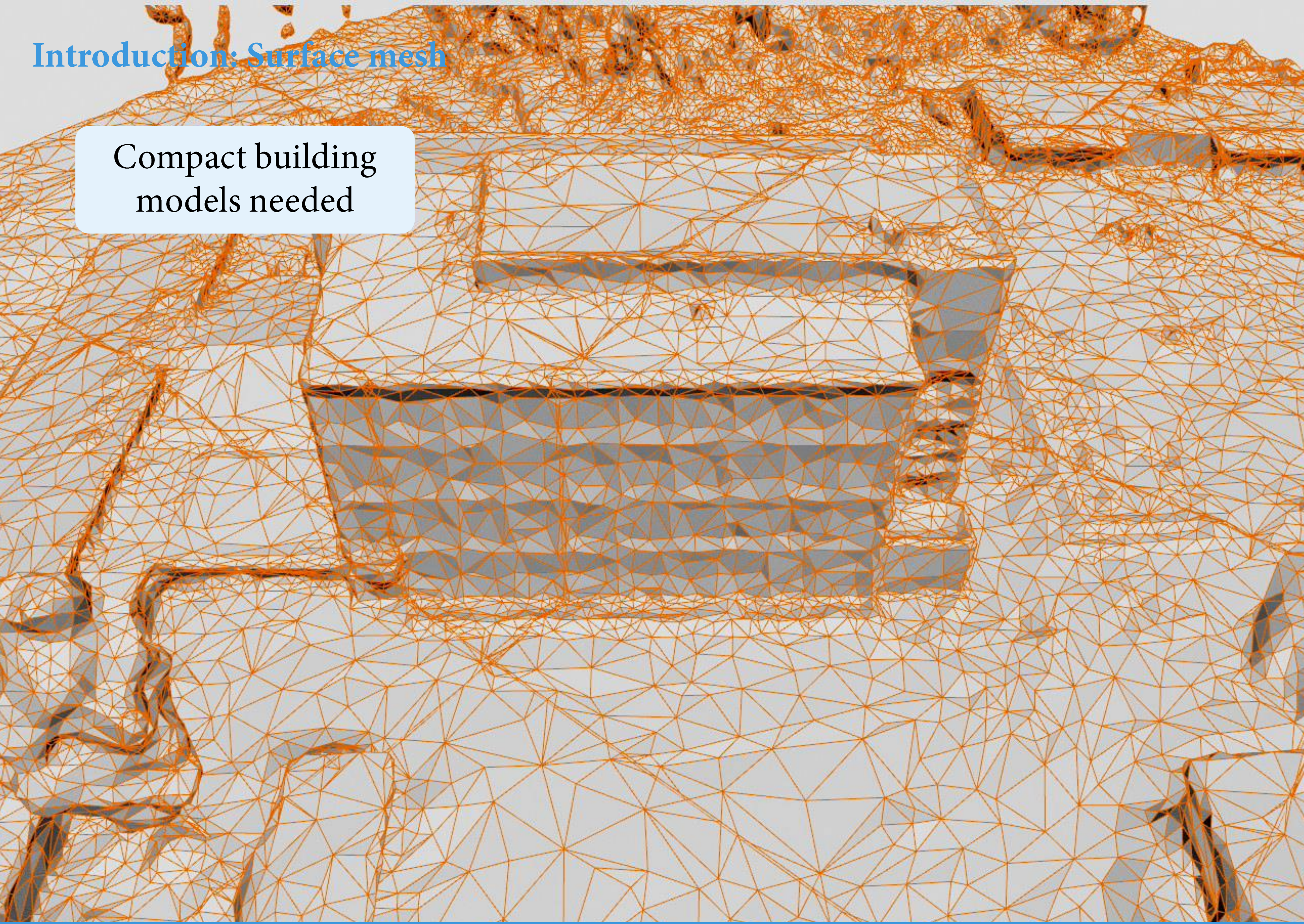




[1]

[1] https://www.tudelft.nl/bk/onderzoek/projecten/geoinformation-technology-governance

Introduction: Surface mesh

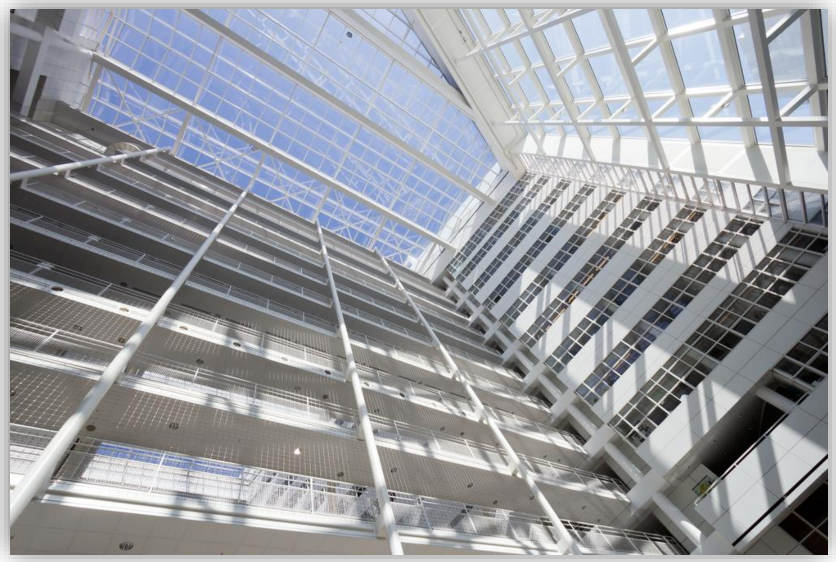


Compact building models needed

Introduction: Piecewise planarity

Piecewise-planar building models

- **Ubiquitous in the built environment**
- Capturing both geometry and topology with non-uniformity
- Compact, efficient with sparse sets of parameters

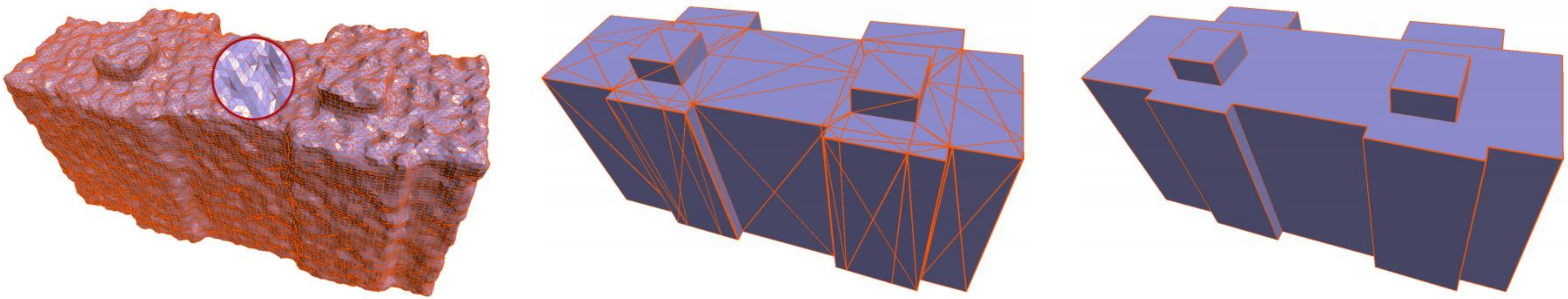

[2]

[2] https://www.tudelft.nl/bk/onderzoek/onderzoek-bij-bouwkunde/management-in-the-built-environment

Introduction: Piecewise planarity

Piecewise-planar building models

- Ubiquitous in the built environment
- **Capturing both geometry and topology with non-uniformity**
- **Compact, efficient with sparse sets of parameters**



Dense triangles (smooth)
326,234 facets

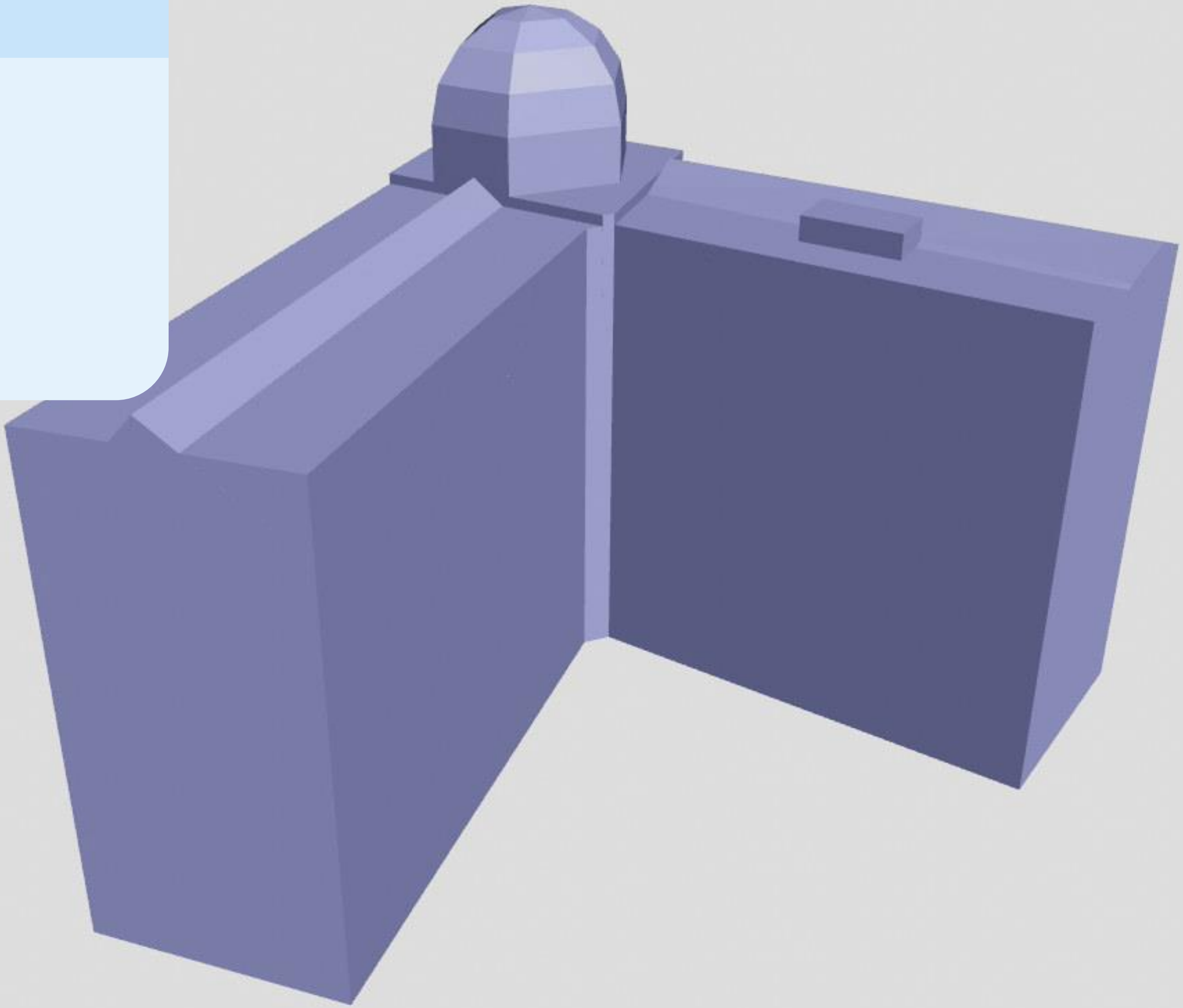
Sparse triangles
198 facets

Sparse polygons
61 facets

Introduction: The *reconstruction* problem

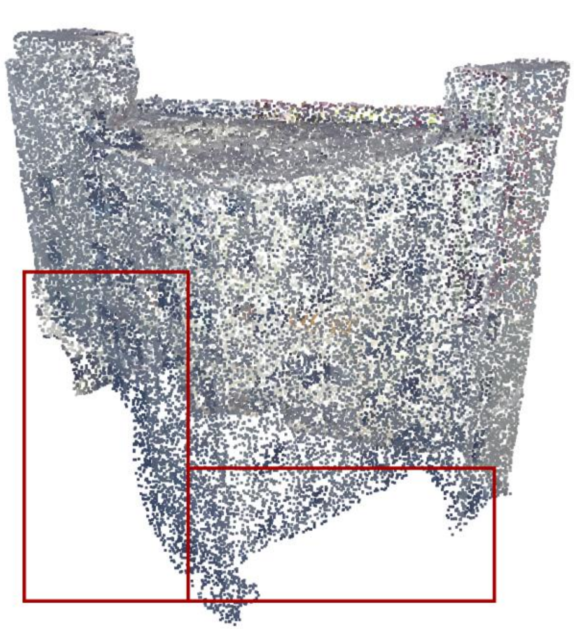
The goal

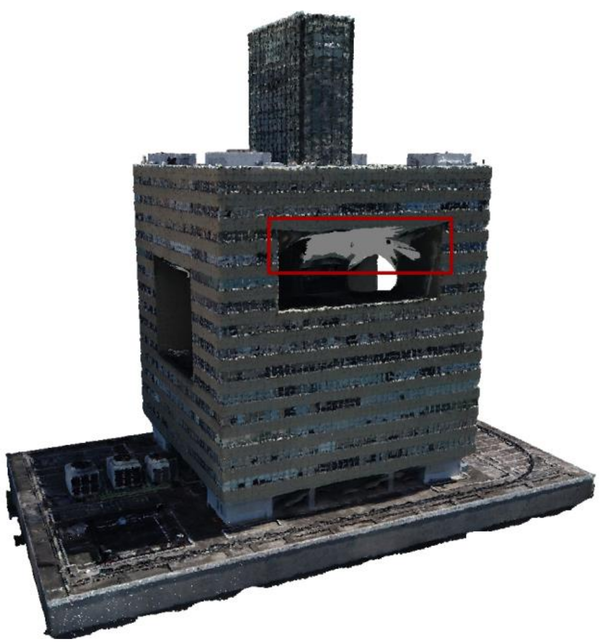
- Compact
- Watertight
- Accurate
- Efficient



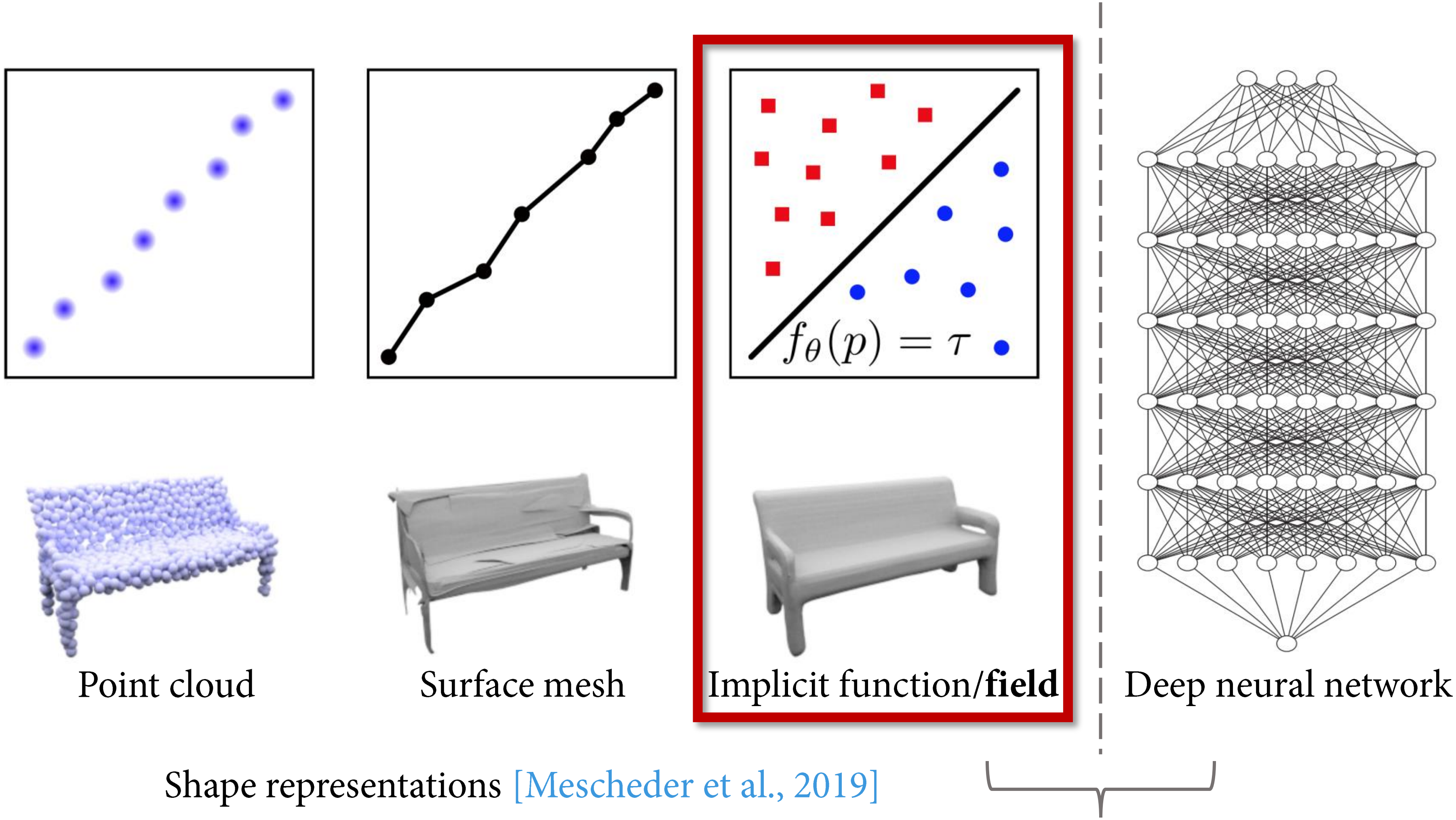
Introduction: Challenges

- Compactness, watertightness, efficiency
- **Limited input data quality**





Introduction: Inspiration and research question



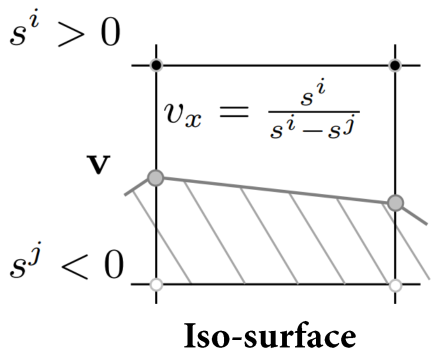
Shape representations [Mescheder et al., 2019]

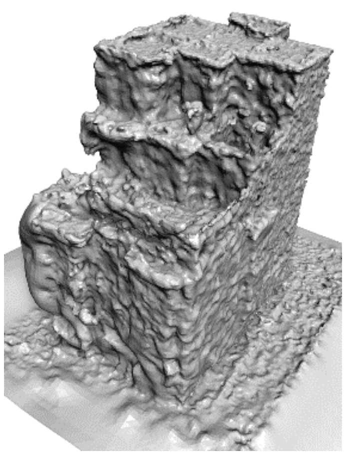
How can deep implicit fields be used for compact building model reconstruction?

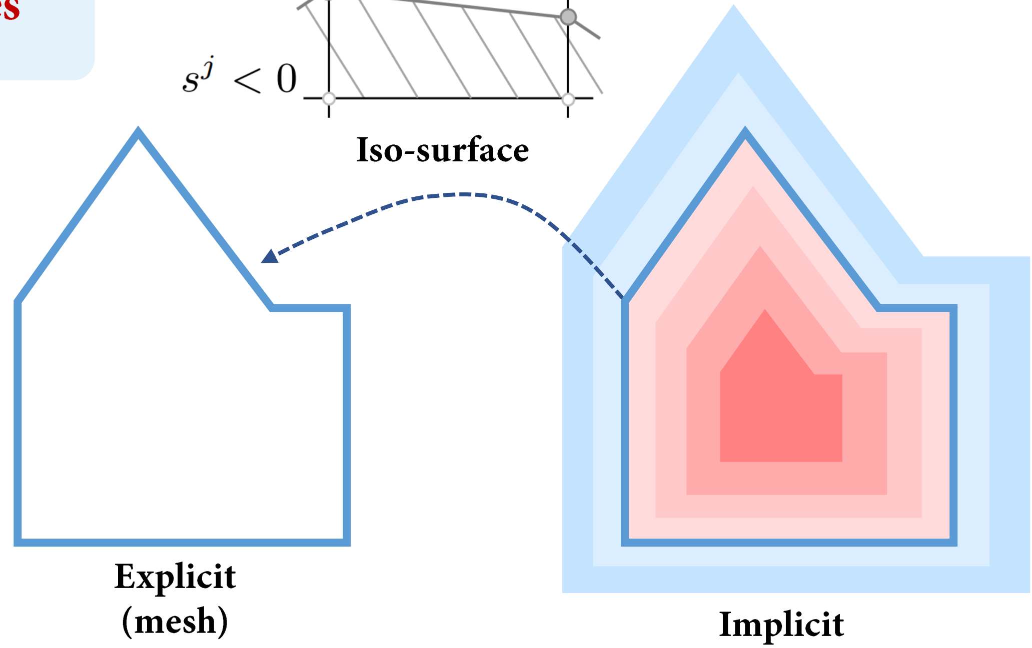
Related work: Shape reconstruction (smooth)

- Poisson reconstruction [Kazhdan et al., 2006]
- Points2Surf [Erler et al., 2020]

Massive triangles

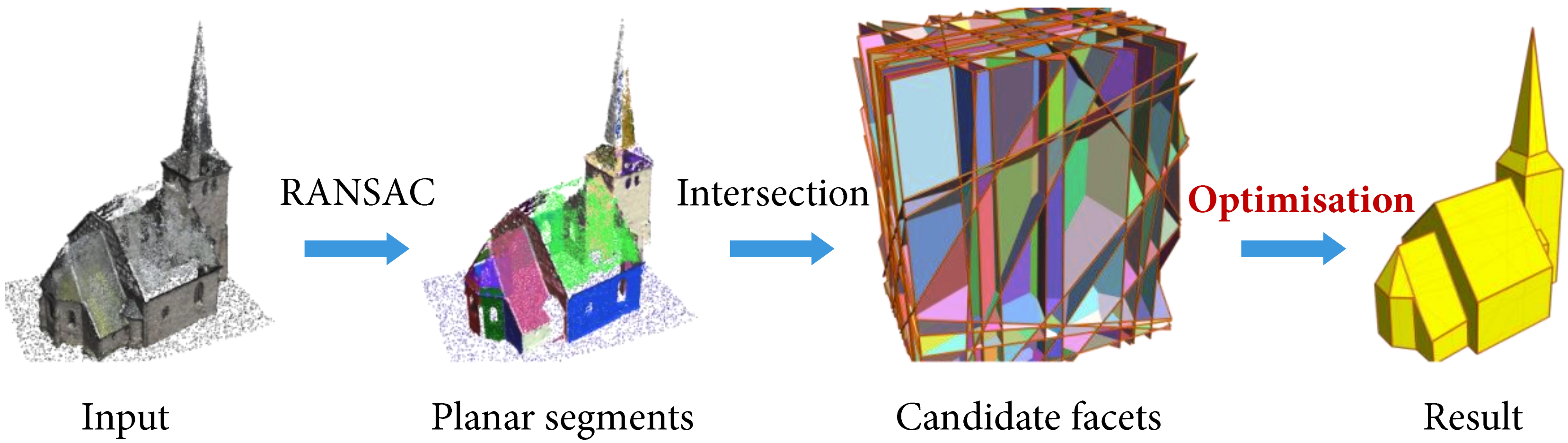


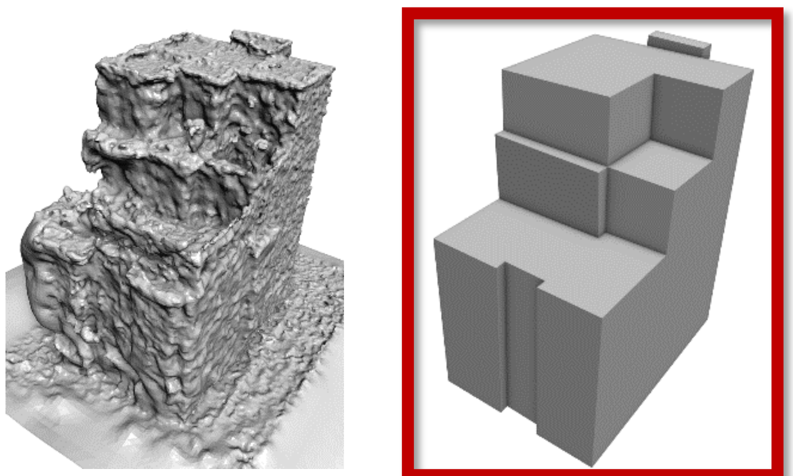




Related work: Shape reconstruction (piecewise-planar)

- PolyFit [Nan and Wonka, 2017]





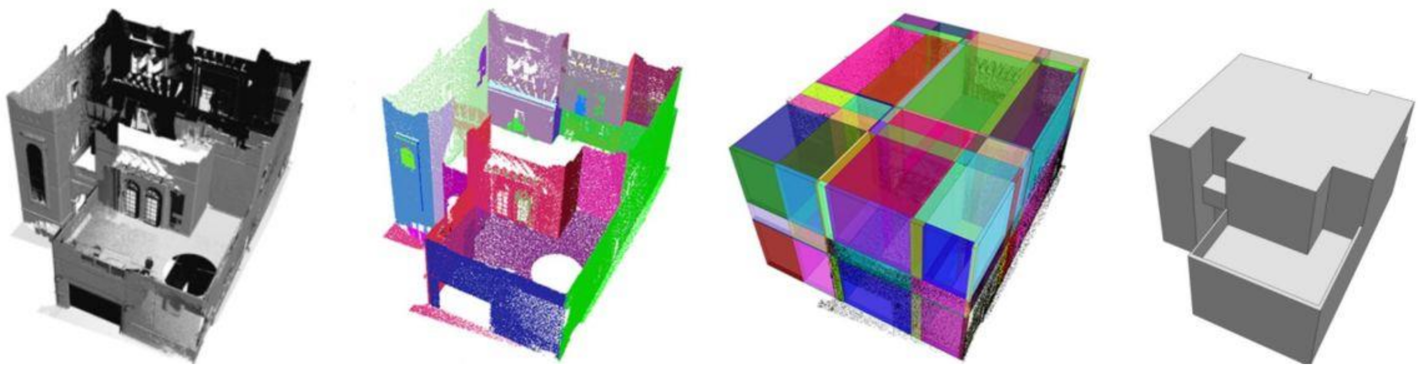
Scalability issue

with its

integer programming solver

Related work: Geometry simplification

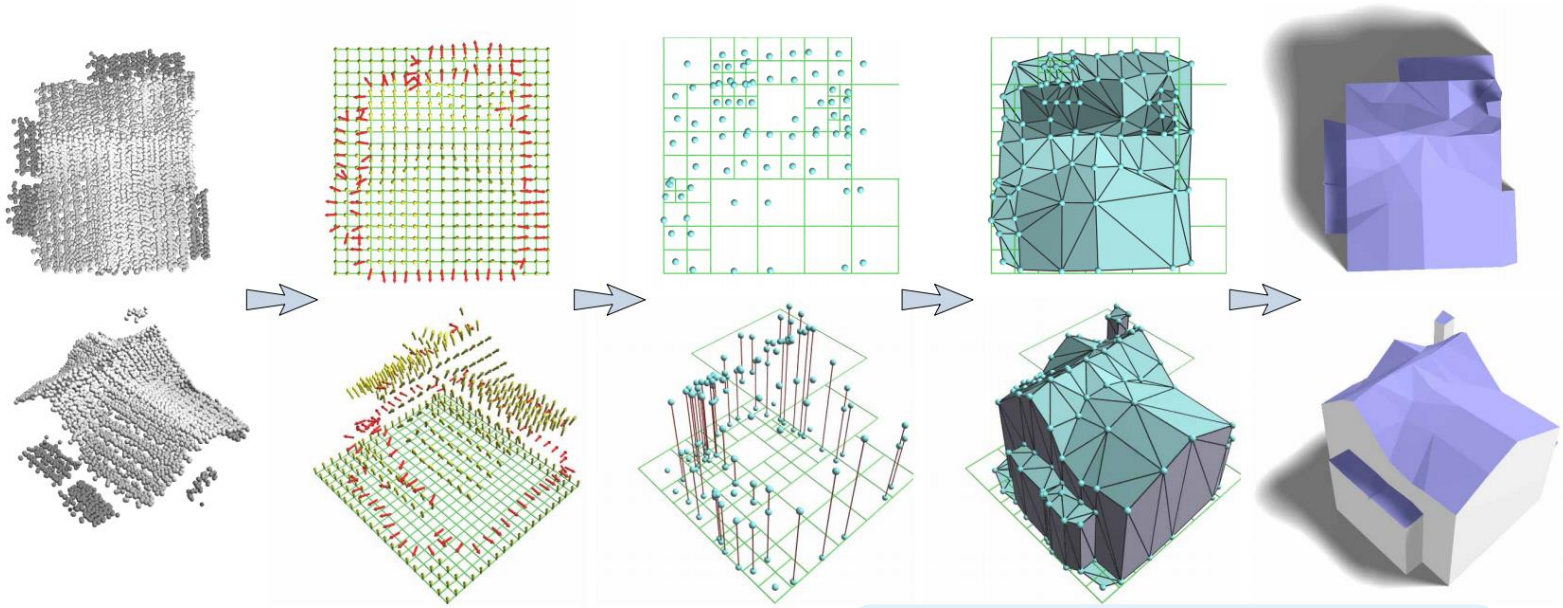
- **Manhattan-world reconstruction** [Li et al., 2016b]
- 2.5D Dual Contouring [Zhou and Neumann, 2010]



Not generic with only boxes

Related work: Geometry simplification

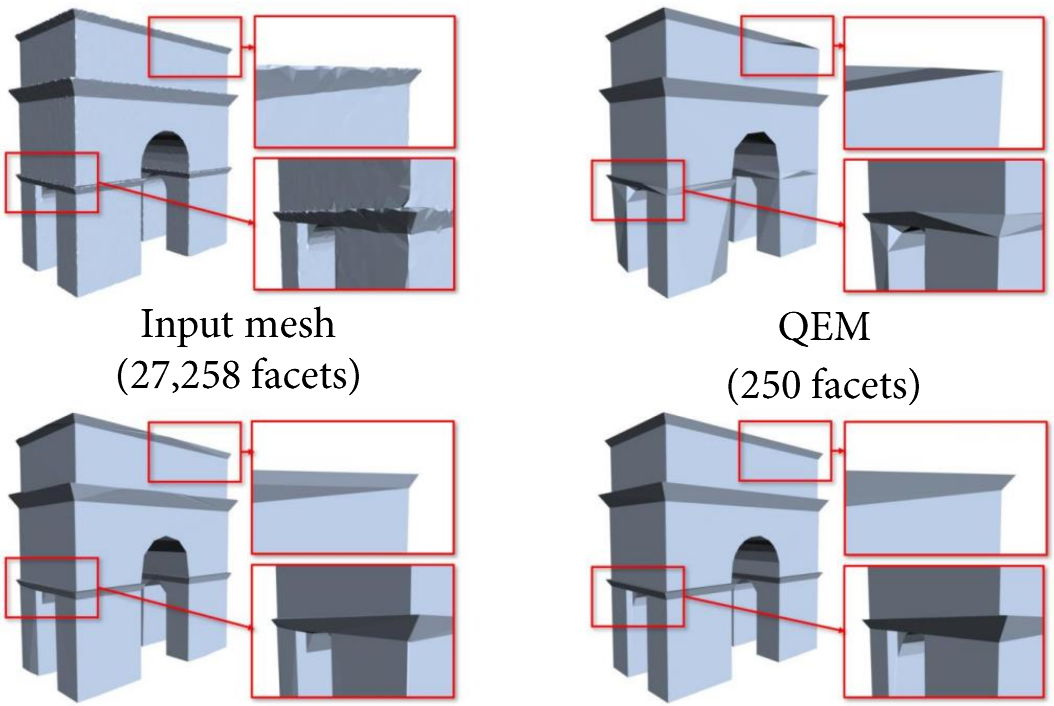
- Manhattan-world reconstruction [Li et al., 2016b]
- **2.5D Dual Contouring** [Zhou and Neumann, 2010]



Not generic with only 2.5D

Related work: Surface approximation

- Quadric error metrics (QEM) [Garland and Heckbert, 1997]
- Variational shape approximation (VSA) [Cohen-Steiner et al., 2004]
- Structure-aware mesh decimation (SAMD) [Salinas et al., 2015]



Input mesh (27,258 facets)

QEM (250 facets)

VSA (250 facets)

SAMD (250 facets)

Dependent on input mesh

[Bouzas et al., 2020]

Related work: Summary

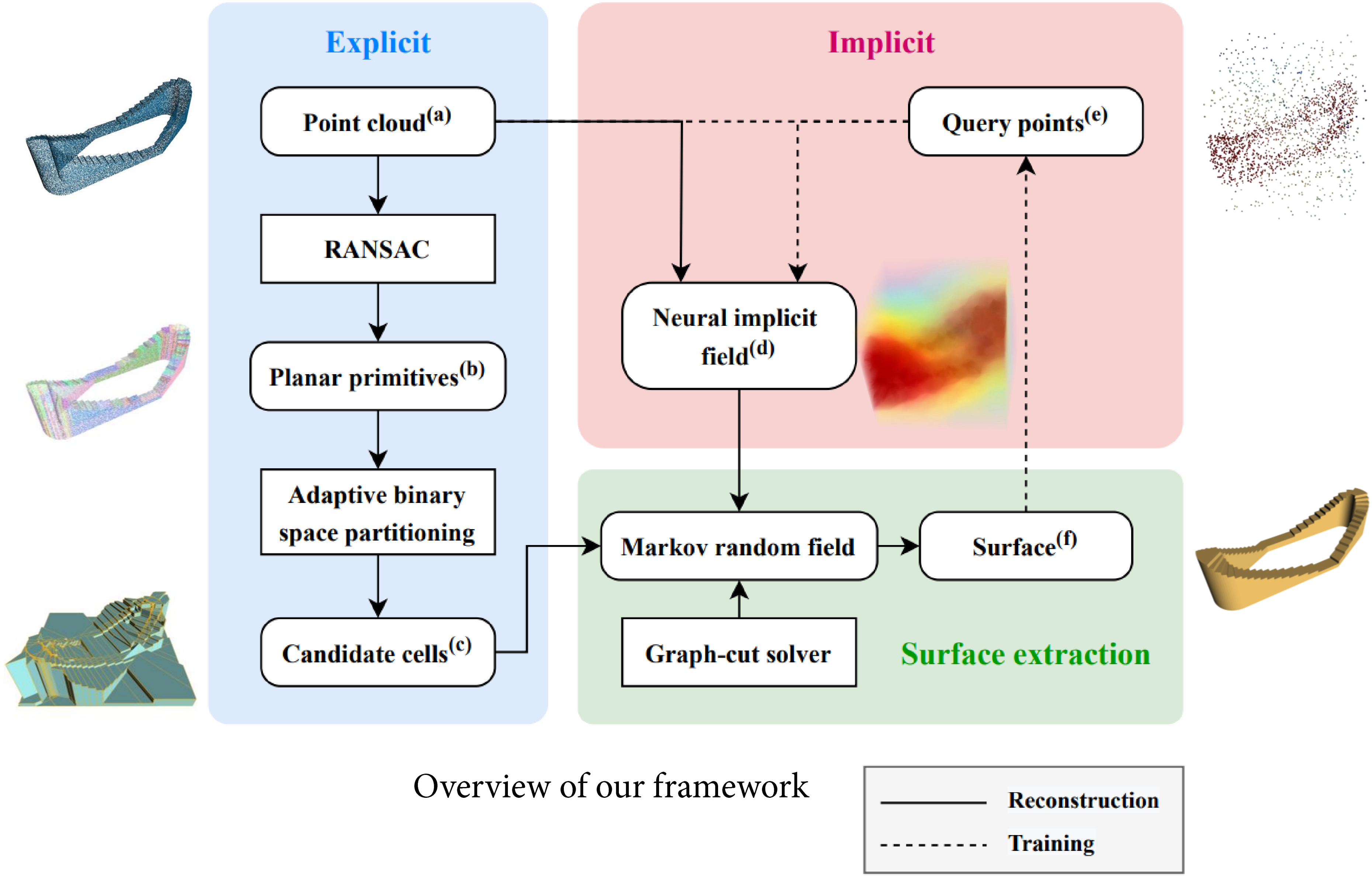
Related work	Characteristics				
Name	**Category**	**Compact**	**Watertight**	**Generic**	**Efficient**
Poisson [Kazhdan et al., 2006]	RC	✗	✗	✓	✓
Points2Surf [Erler et al., 2020]	RC	✗	✗	✓	✗
PolyFit [Nan and Wonka, 2017]	RC	✓	✓	✓	✗
QEM [Garland and Heckbert, 1997]	AP	✓	✗	✓	✗
SAMD [Salinas et al., 2015]	AP	✓	✗	✓	✗
VSA [Cohen-Steiner et al., 2004]	AP	✓	✗	✓	✗
Manhattan-world [Li et al., 2016b]	SP	✓	✓	✗	✓
2.5D DC [Zhou and Neumann, 2010]	SP	✗	✓	✗	✓
Ours	RC	✓	✓	✓	✓

Characteristics overview of related work[3]

[3] Only methods in comparison through experiments (with official open-source code); See in the thesis a complete literature review

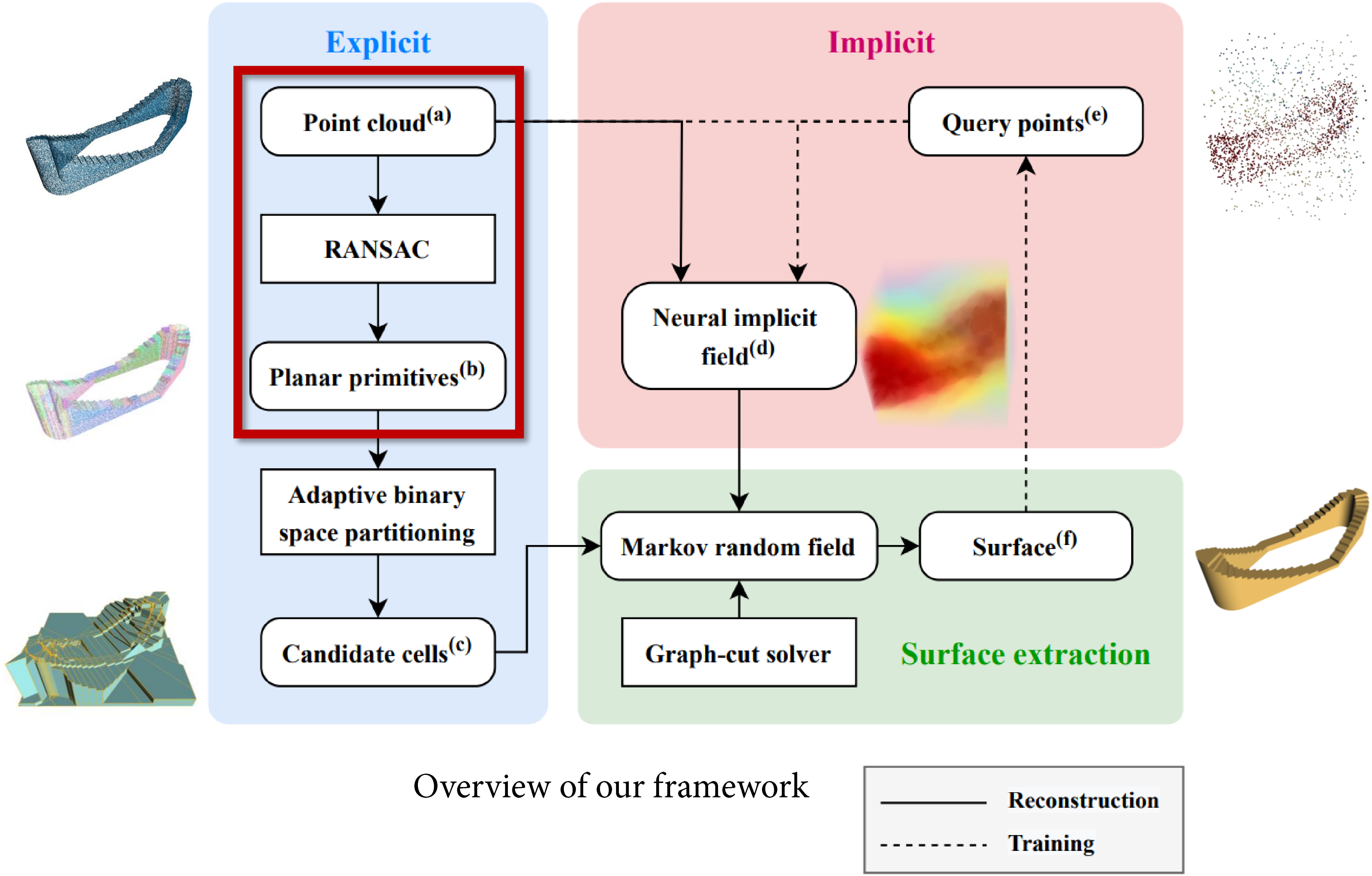
- **Introduction**
- **Related work**
- **Methodology**
- **Datasets**
- **Results and discussion**
- **Conclusions**

Methodology: Overview



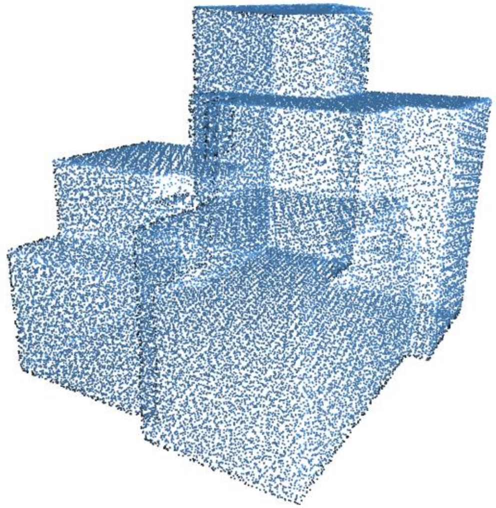
Overview of our framework

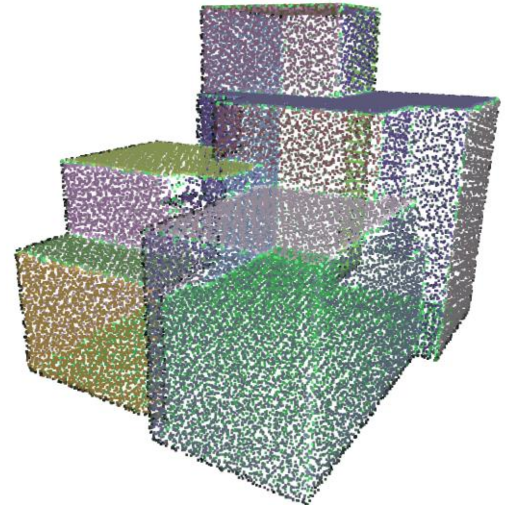
Methodology: Overview



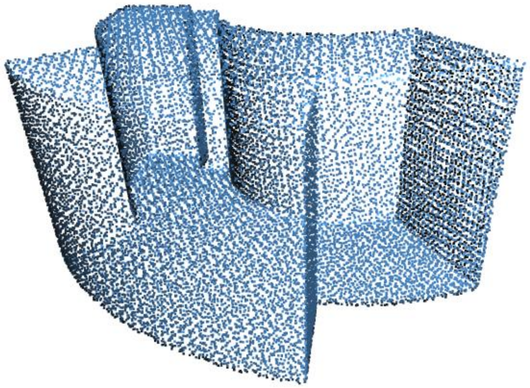
Overview of our framework

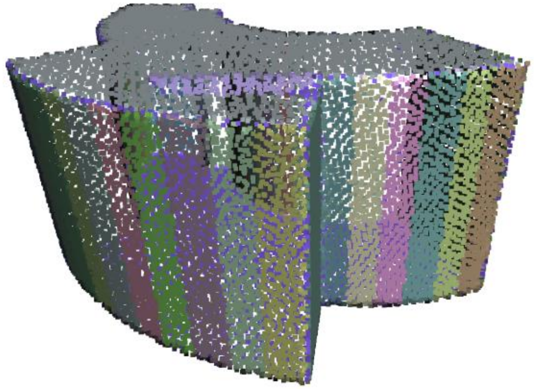
Methodology: Adaptive binary space partitioning





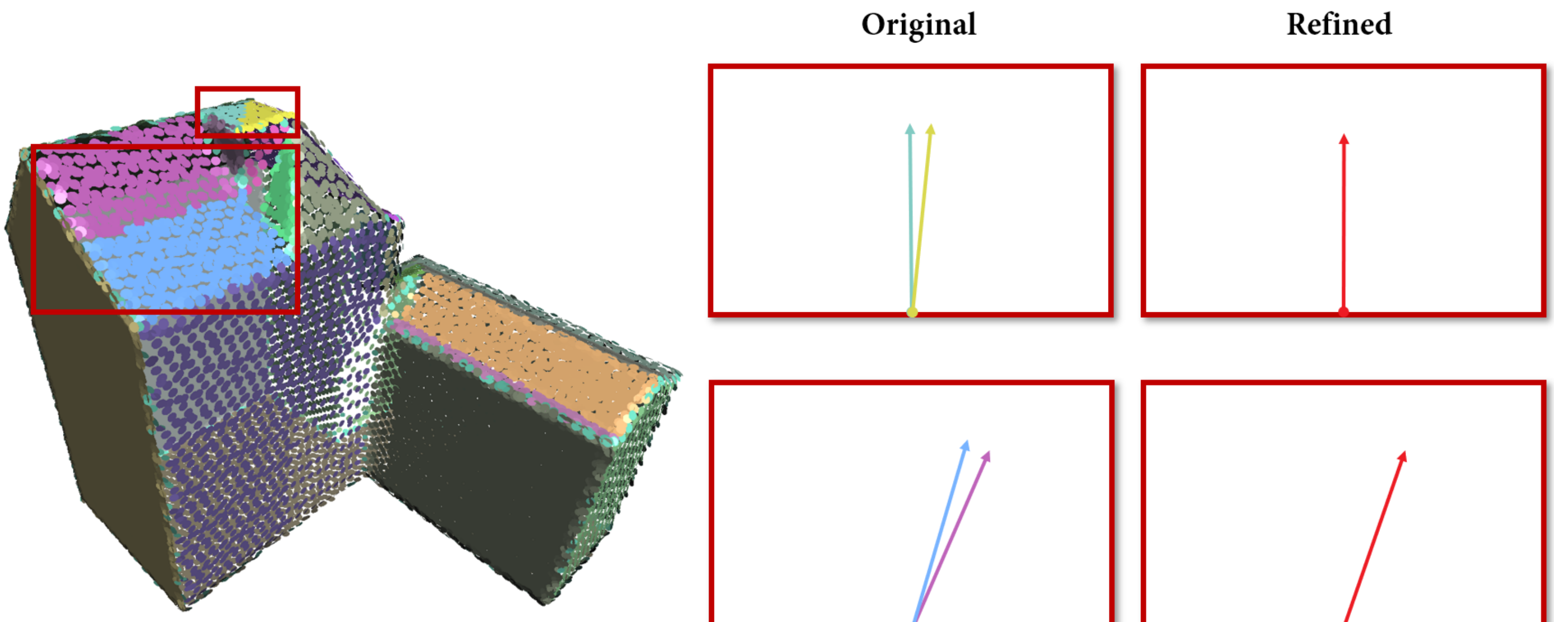
RANSAC





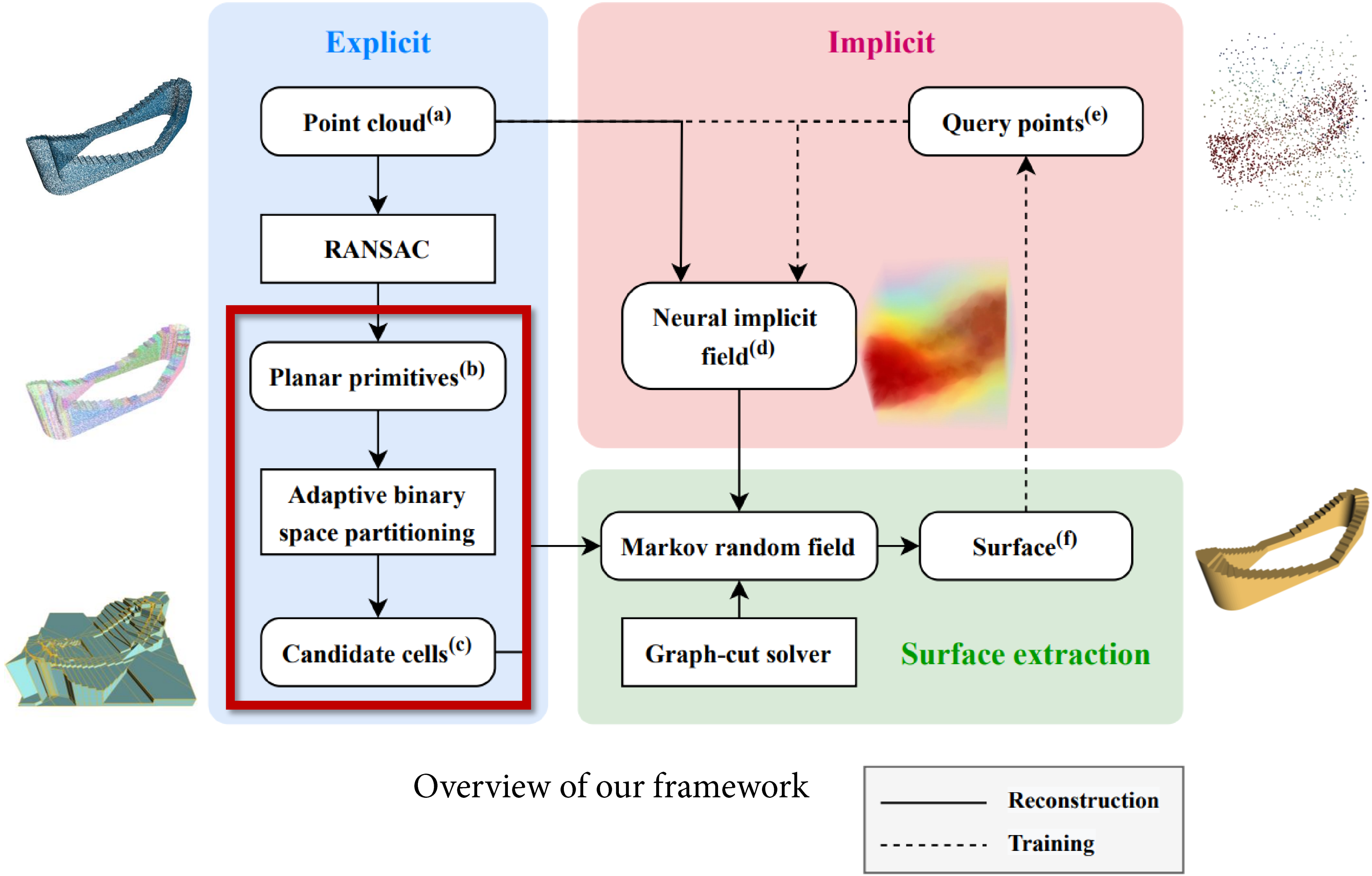
Planar primitive detection

Methodology: Adaptive binary space partitioning



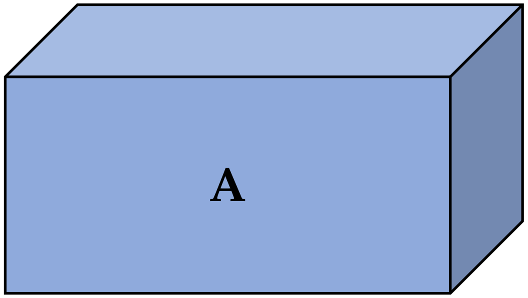
Planar primitive refinement

Methodology: Overview

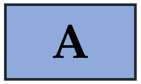


Overview of our framework

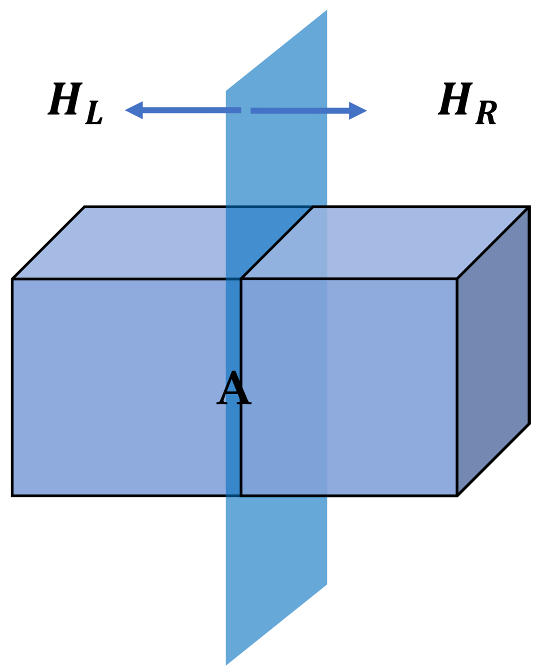
Methodology: Adaptive binary space partitioning

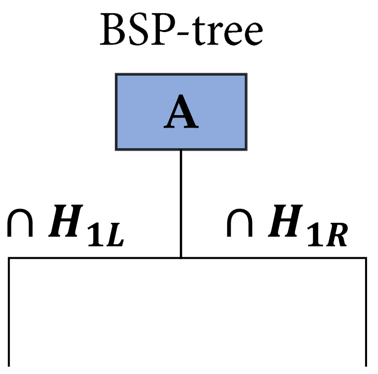


BSP-tree

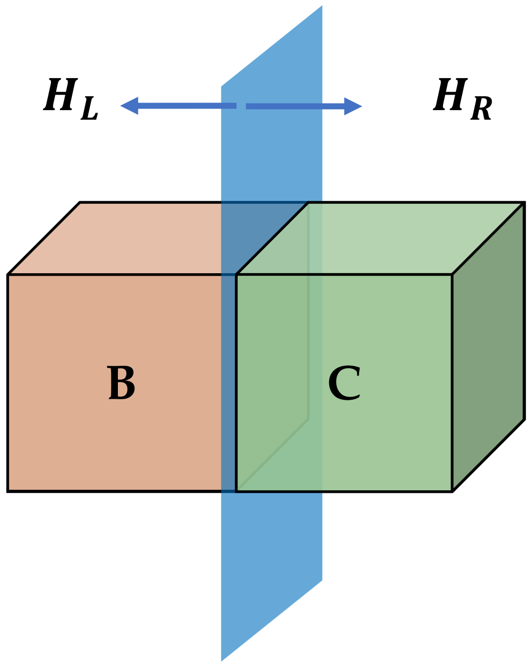


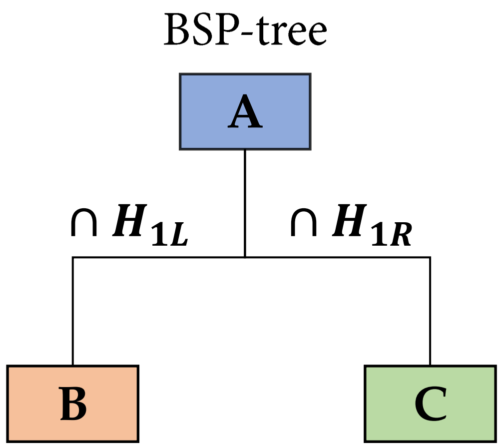
Methodology: Adaptive binary space partitioning



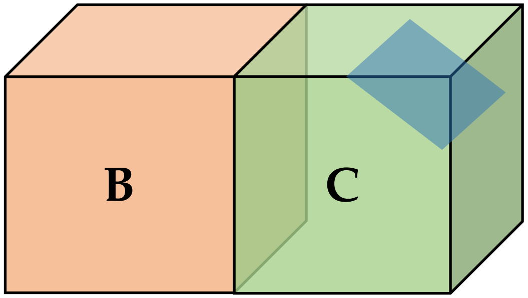


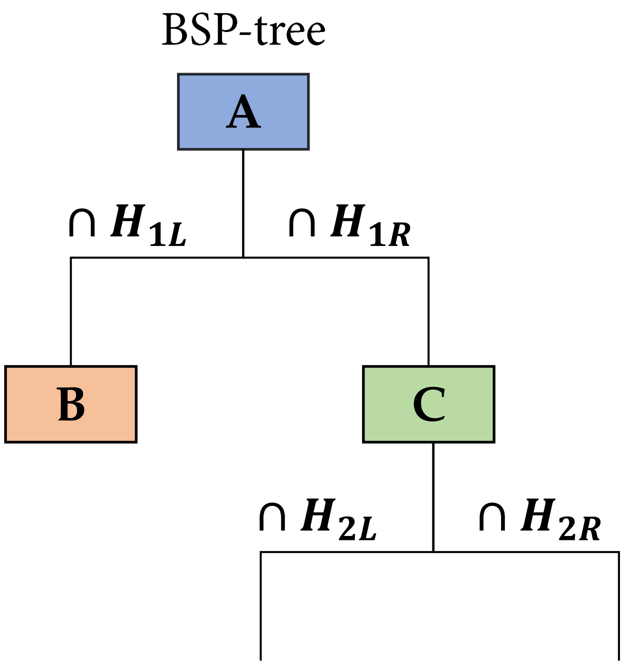
Methodology: Adaptive binary space partitioning



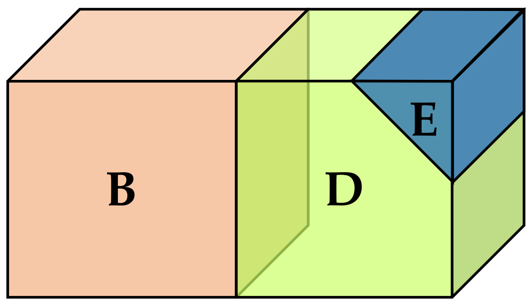


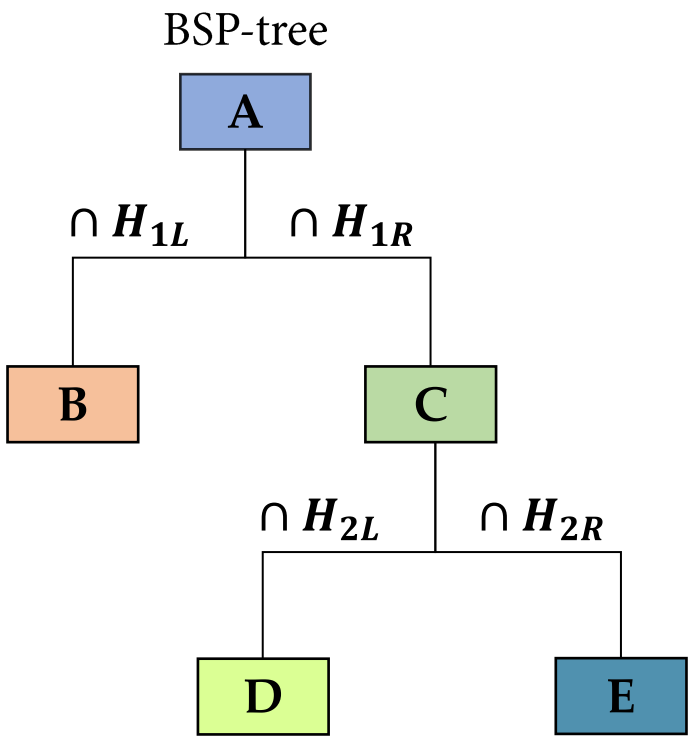
Methodology: Adaptive binary space partitioning



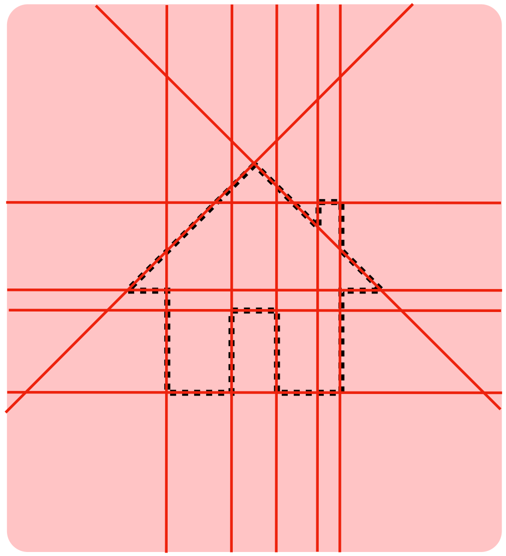


Methodology: Adaptive binary space partitioning

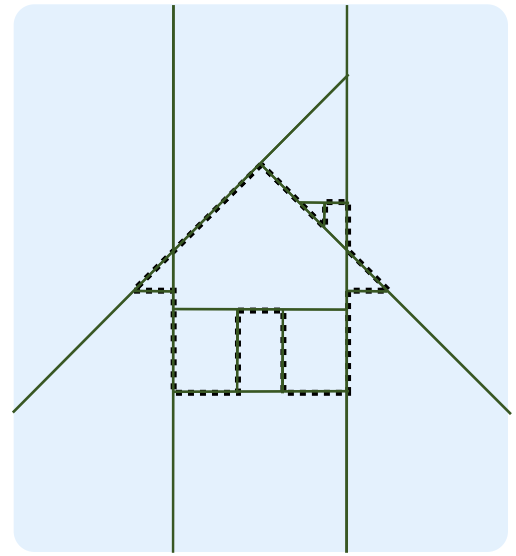




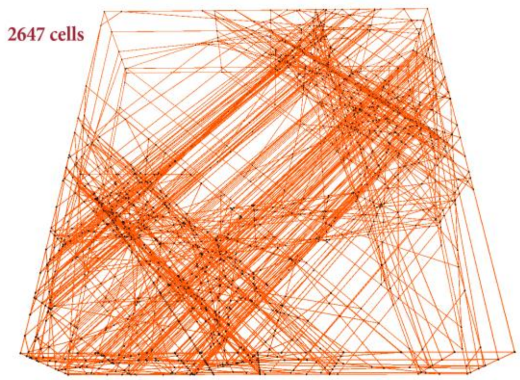
Methodology: Adaptive binary space partitioning

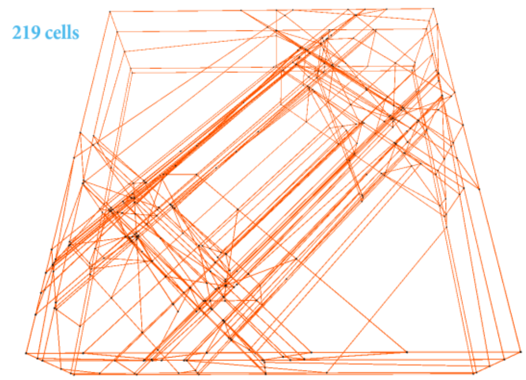


Exhaustive

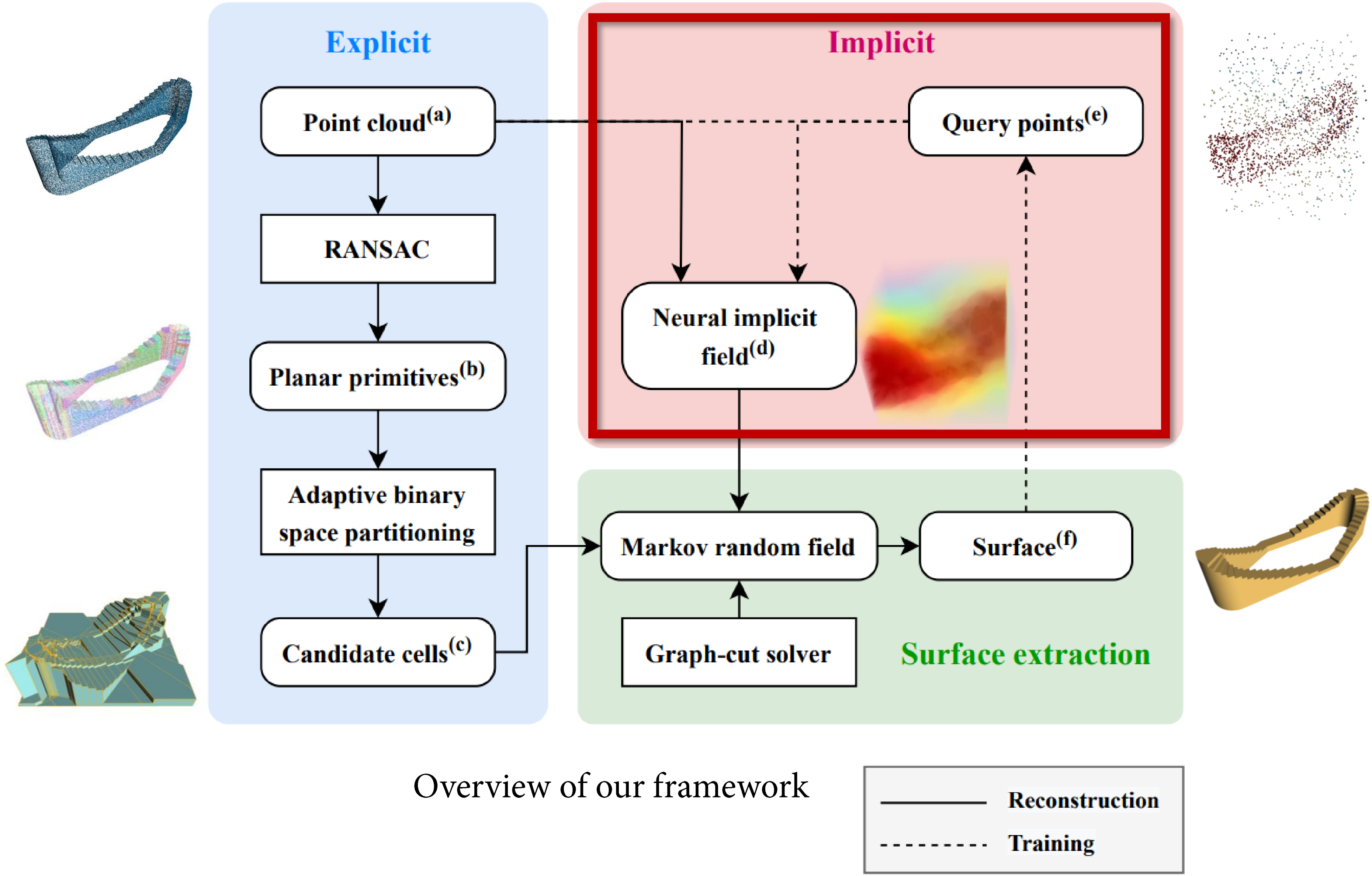


Adaptive





Methodology: Overview



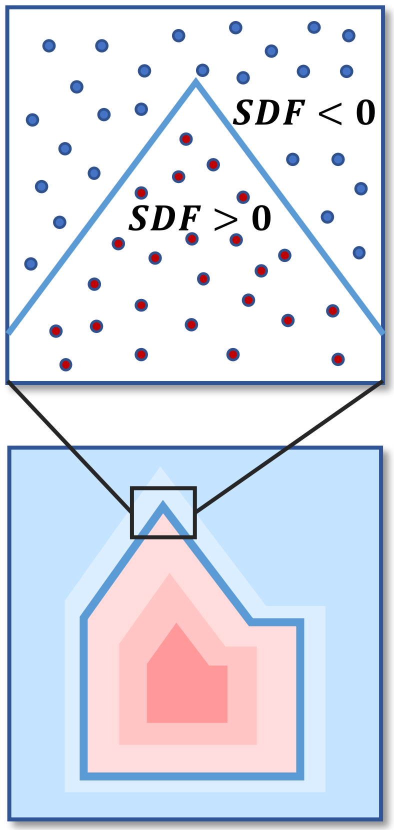
Overview of our framework

Methodology: Occupancy learning in function space

Signed distance function

$$SDF(\mathbf{x}) = s : \mathbf{x} \in \mathbb{R}^3, s \in \mathbb{R}.$$

Surface at $SDF(\cdot) = 0$



Methodology: Occupancy learning in function space

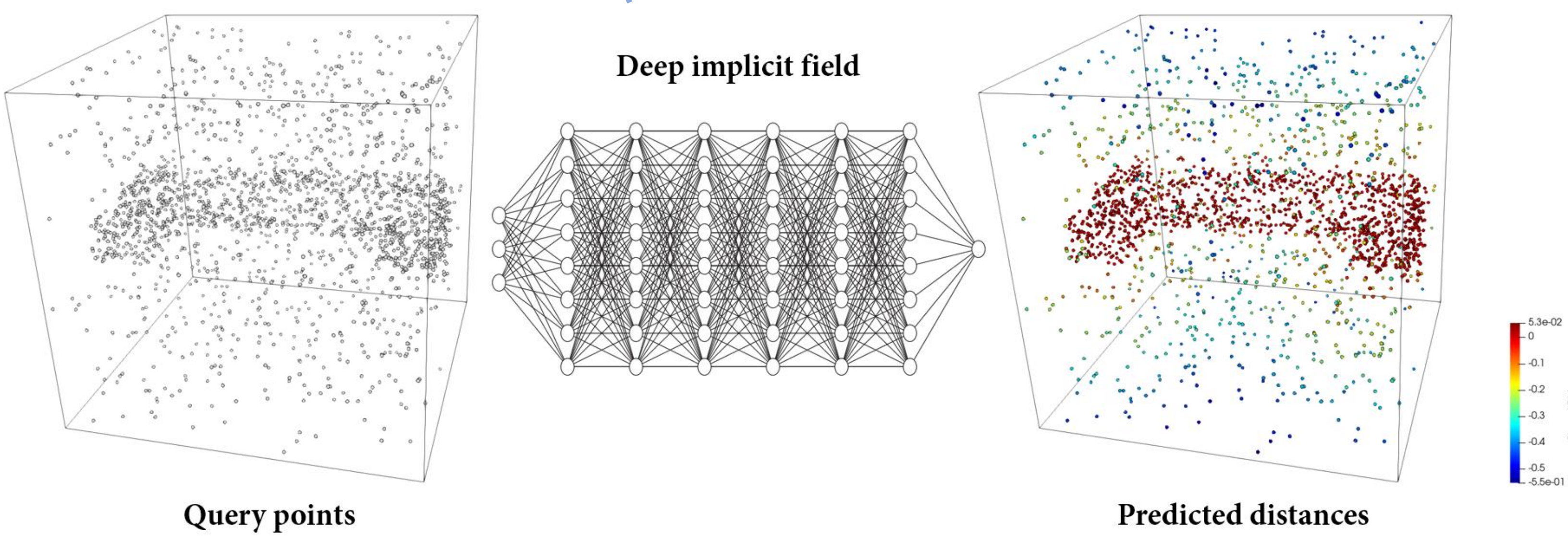
Signed distance function

$$SDF(\mathbf{x}) = s : \mathbf{x} \in \mathbb{R}^3, s \in \mathbb{R}.$$

Surface at $SDF(\cdot) = 0$

$$SDF(\mathbf{x}) \approx \tilde{f}_P(\mathbf{x}) = s_\theta(\mathbf{x} \mid \mathbf{z}), \text{ with } \mathbf{z} = e_\varphi(P)$$

P: point cloud
θ: NN parameters
e : encoder

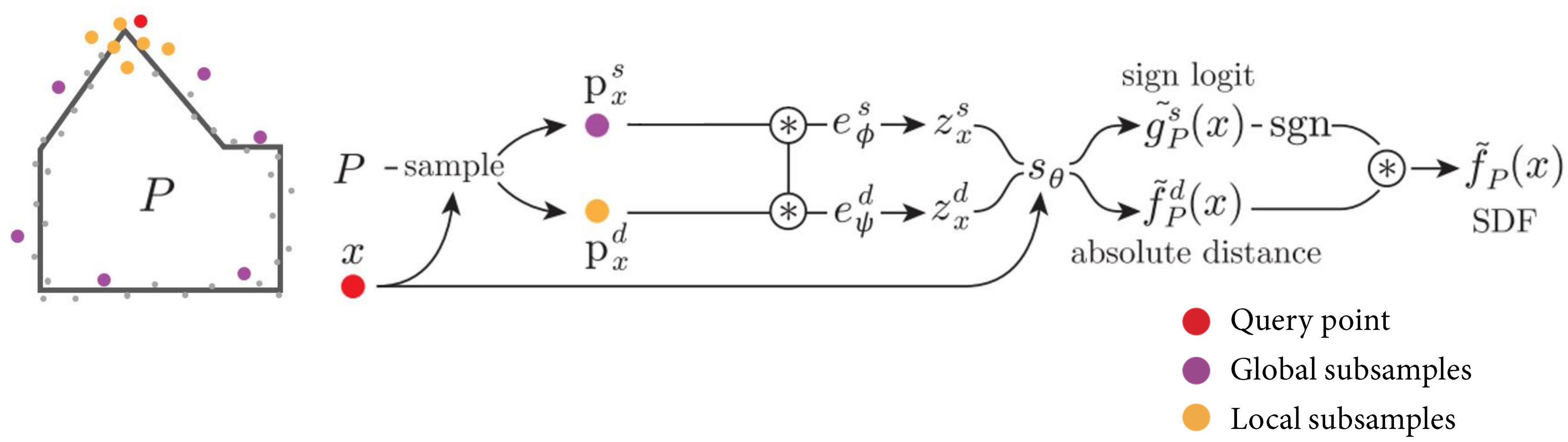


Methodology: Occupancy learning in function space

Points2Surf neural network architecture [Erler et al., 2020]

$$SDF(\mathbf{x}) \approx \tilde{f}_P(\mathbf{x}) = s_\theta(\mathbf{x} \mid \mathbf{z}), \text{ with } \mathbf{z} = e_\varphi(P)$$

- $\tilde{f}^d_P(\mathbf{x}) = s^d_\theta\left(x \mid \mathbf{z}^d_x\right)$, with $\mathbf{z}^d_\mathbf{x} = e^d_\varphi\left(\mathbf{p}^d_\mathbf{x}\right)$ *Absolute distance*
- $\tilde{f}^s_P(\mathbf{x}) = \text{sgn}\left(\tilde{g}^s_P(\mathbf{x})\right) = \text{sgn}\left(s^s_\theta\left(\mathbf{x} \mid \mathbf{z}^s_\mathbf{x}\right)\right)$, with $\mathbf{z}^s_\mathbf{x} = e^s_\psi\left(\mathbf{p}^s_\mathbf{x}\right)$ *Sign*

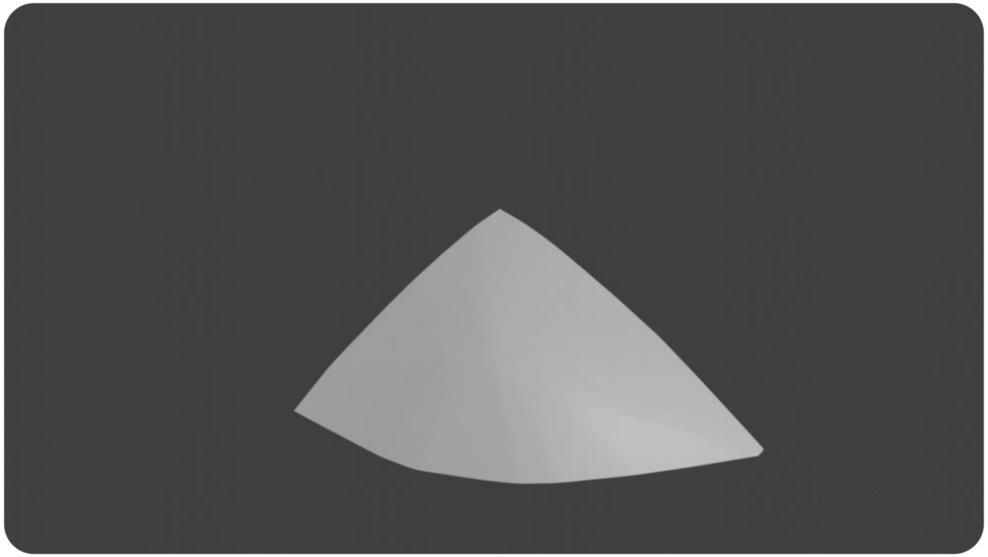


Methodology: Occupancy learning in function space

Training with loss function

$$\sum_{(P,S)\in\mathcal{S}} \sum_{\mathbf{x}\in\mathcal{X}_S} \mathcal{L}^d(\mathbf{x}, P, S) + \mathcal{L}^s(\mathbf{x}, P, S)$$

- $\mathcal{L}^d(\mathbf{x}, P, S) = \left\| \tanh\left(\left|\tilde{f}_P^d(\mathbf{x})\right|\right) - \tanh(|d(\mathbf{x}, S)|) \right\|_2^2$ *Error of distance prediction*
- $\mathcal{L}^s(\mathbf{x}, P, S) = H\left(\sigma\left(\tilde{g}_P^s(\mathbf{x})\right), [f_S(\mathbf{x}) > 0]\right)$ *Error of sign prediction*



Sanity check: overfitting one shape

P: point cloud
S: surface
H: binary cross entropy
σ: sigmoid

Methodology: Occupancy learning in function space

Signed distance voting

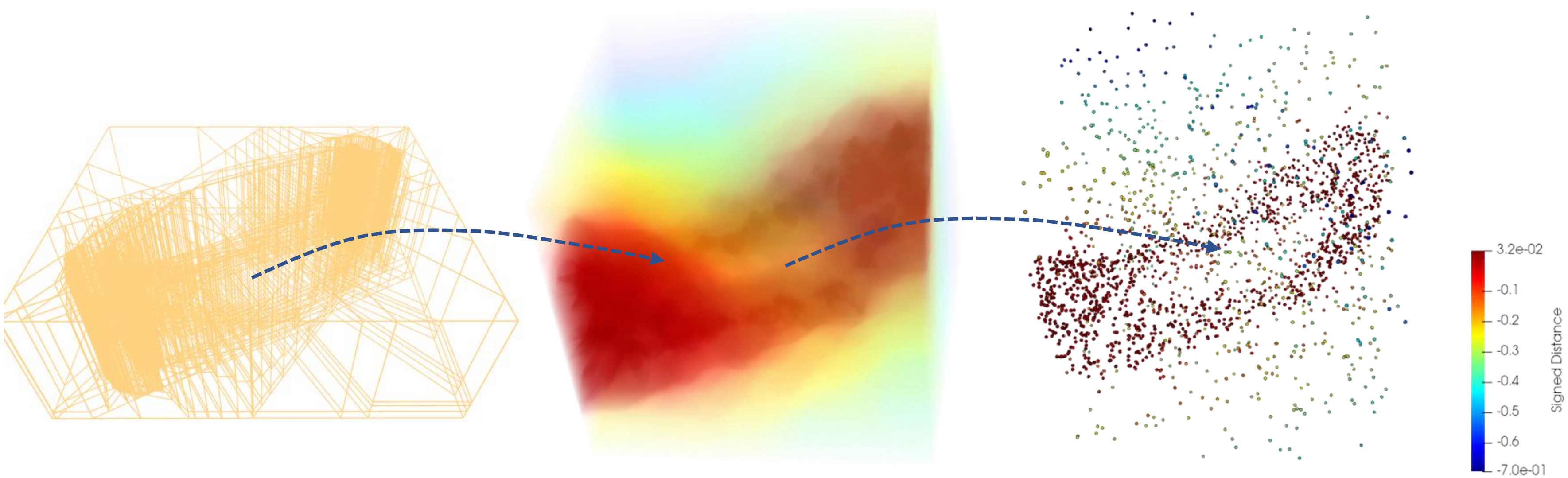
$$\bar{SD}^P = \frac{1}{p} \sum_{i \in P} SD_i^P$$

Point cloud

Methodology: Occupancy learning in function space

Signed distance voting

$$\bar{SD}^{P} = \frac{1}{p}\sum_{i\in P} SD_i^{P}$$

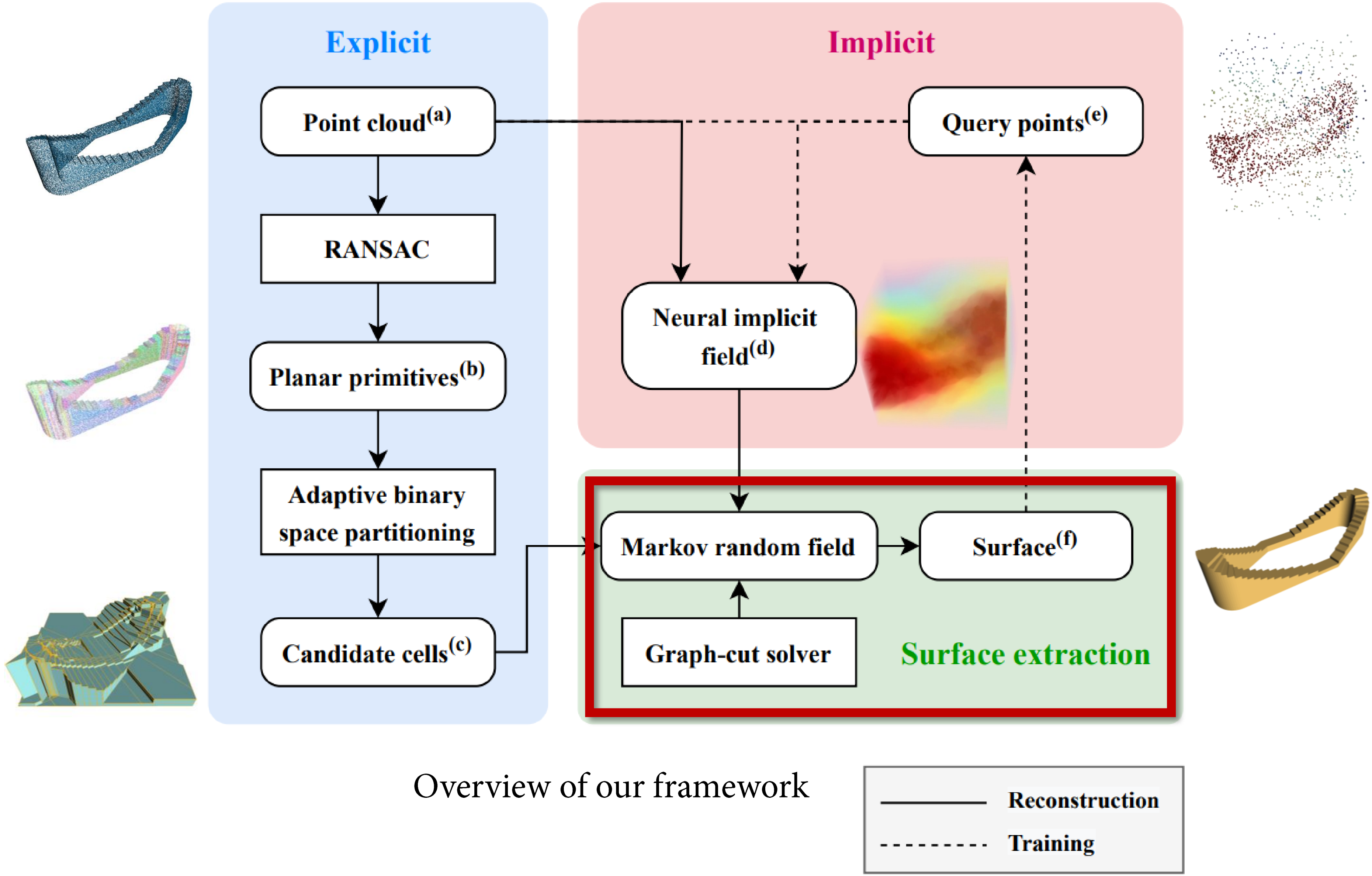


Candidate cells

SDF

Signed distance for each candidate

Methodology: Overview



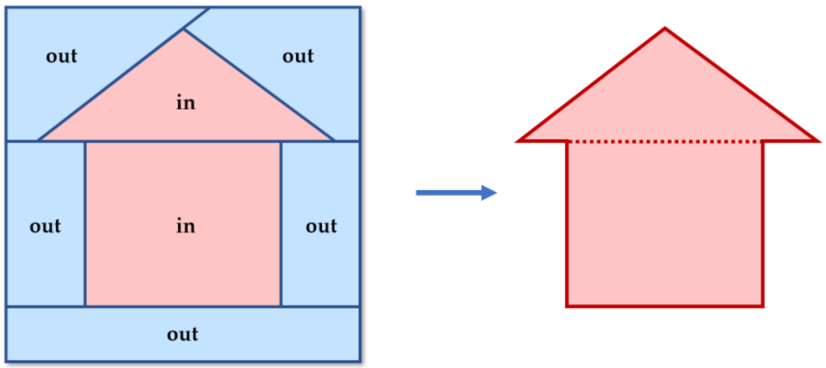
Overview of our framework

Methodology: Surface extraction

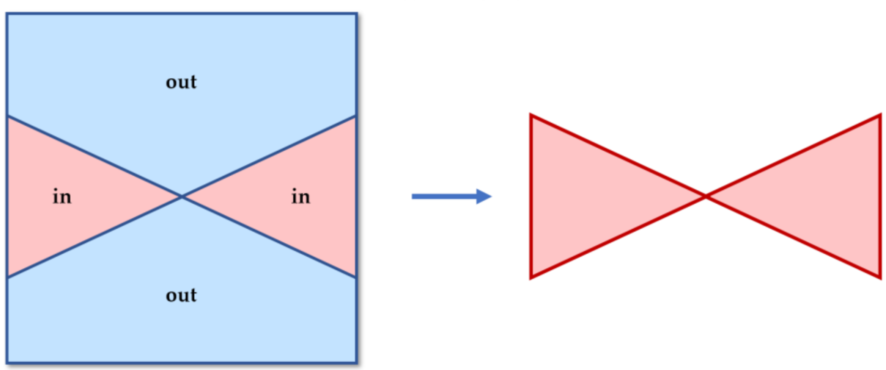
Energy formulation (Markov random field)

$$E(x) = D(x) + \lambda V(x)$$

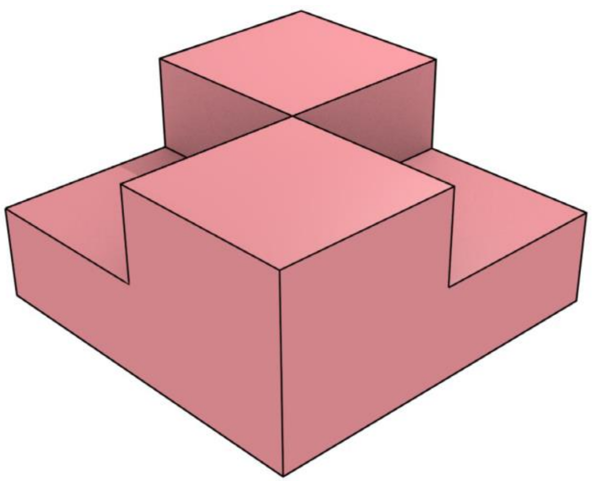
$$x_i = \{in, out\}$$



Manifold



Non-manifold



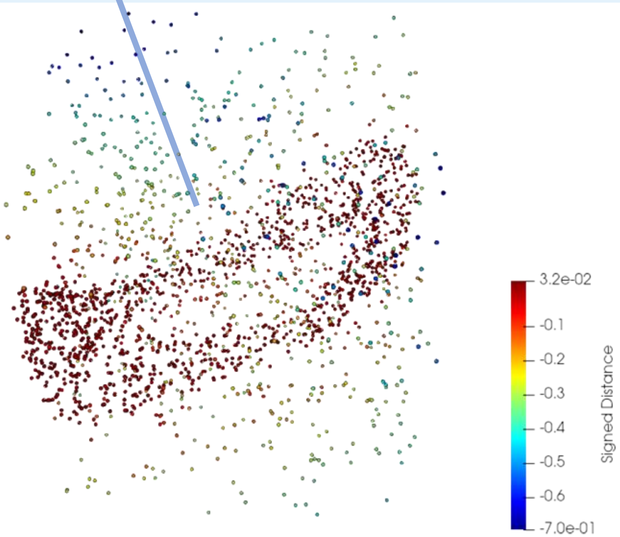
[Ohori, 2016]

Methodology: Surface extraction

Fidelity term (unary potential)

$$D(X) = \frac{1}{|C|} \sum_{i \in C} d_i(C_i, x_i)$$

- $d_i(C_i, x_i) = |probability(C_i) - x_i|$
- $probability(C_i) = sigmoid(SD_i \cdot volume_i)$



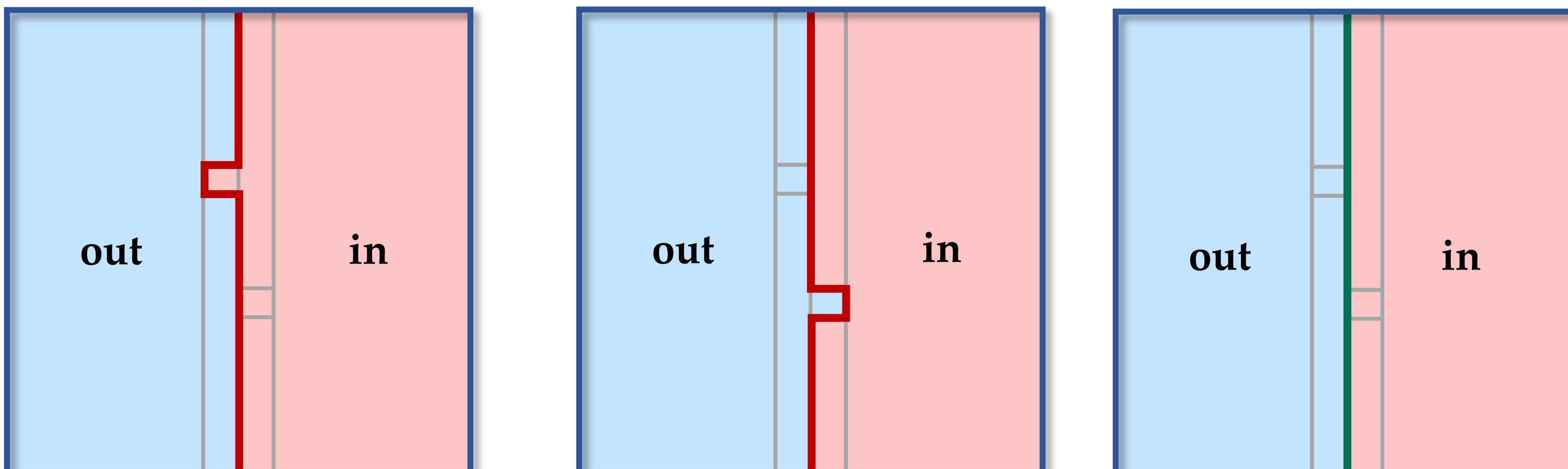
Signed distance for each candidate

Methodology: Surface extraction

Complexity term (pairwise potential)

$$V(X) = \frac{1}{A} \sum_{\{i,j\} \in C} a_{ij} \cdot 1_{x_i \neq x_i}$$

- $\{i, j\} \in C$ represents pairs of adjacent polyhedra
- a_{ij} denotes the shared area

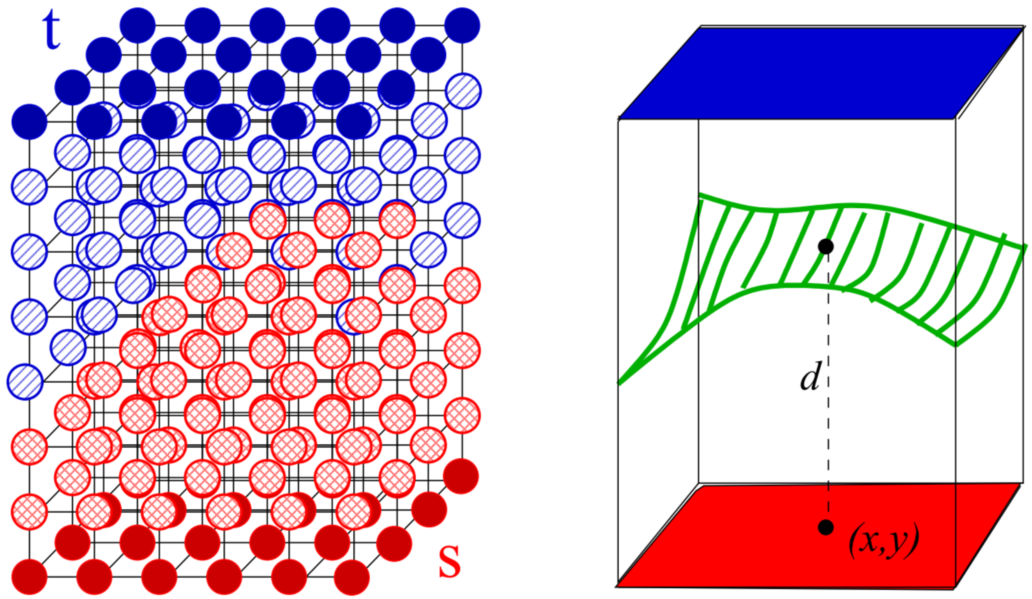


Less zigzagging

Methodology: Surface extraction

Graph-cut solver for the Markov random field

$$E(x) = D(x) + \lambda V(x)$$
$$x_i = \{in, out\}$$

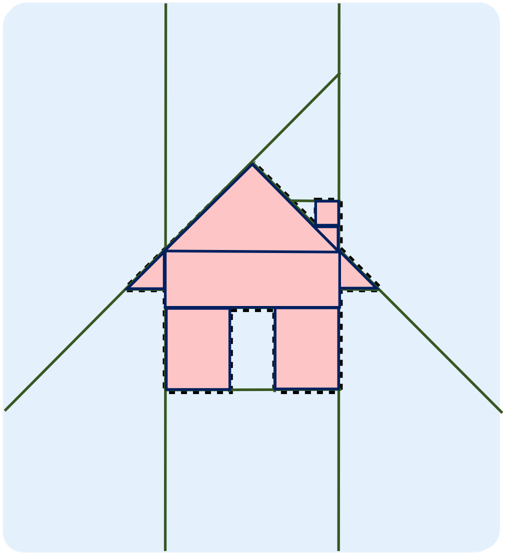


Graph cuts [Boykov and Funka-Lea, 2006]

Methodology: Surface extraction

Graph-cut solver for the Markov random field

$$E(x) = D(x) + \lambda V(x)$$
$$x_i = \{in, out\}$$



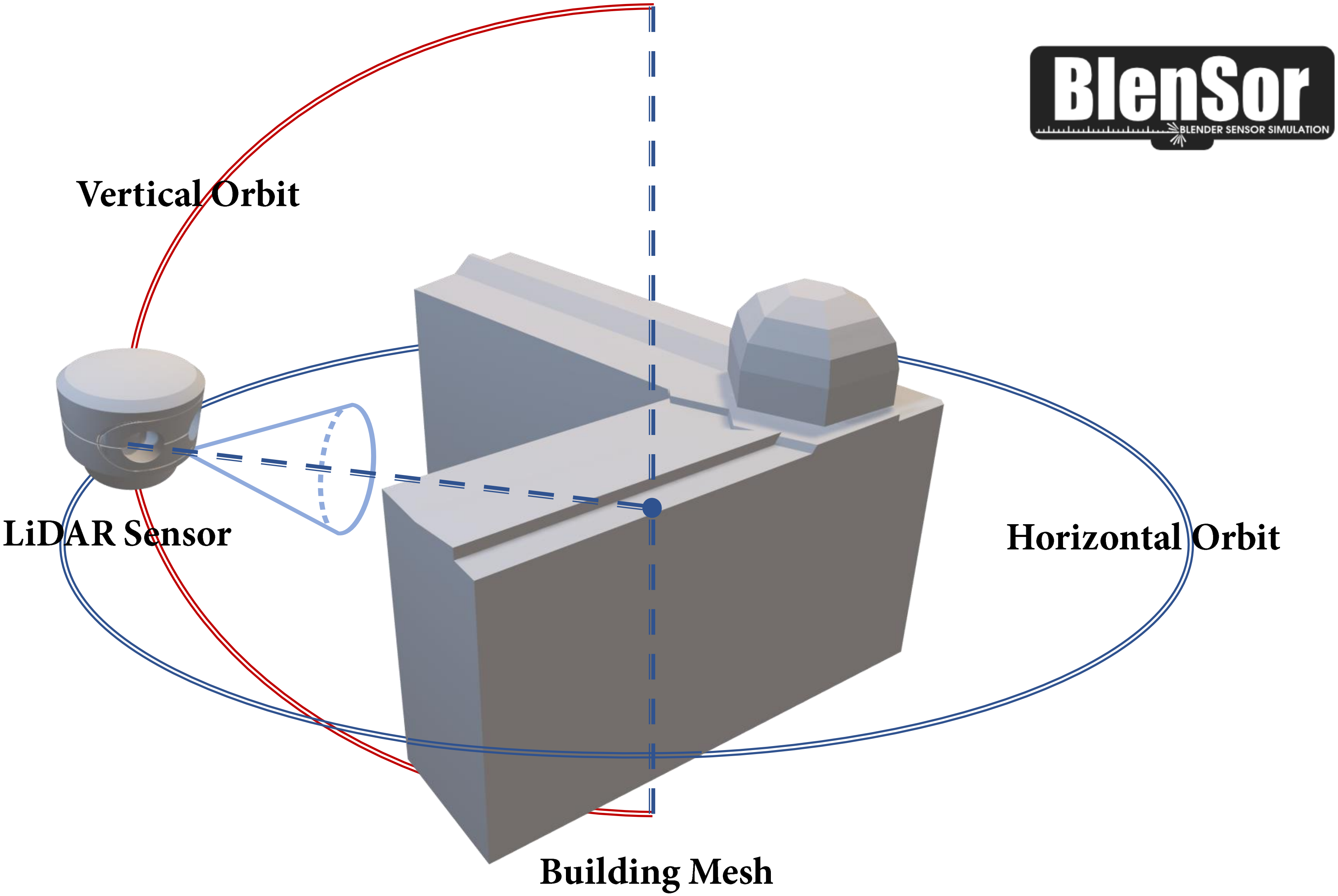
- Introduction
- Related work
- Methodology
- **Datasets**
- Results and discussion
- Conclusions

Datasets: *Helsinki*

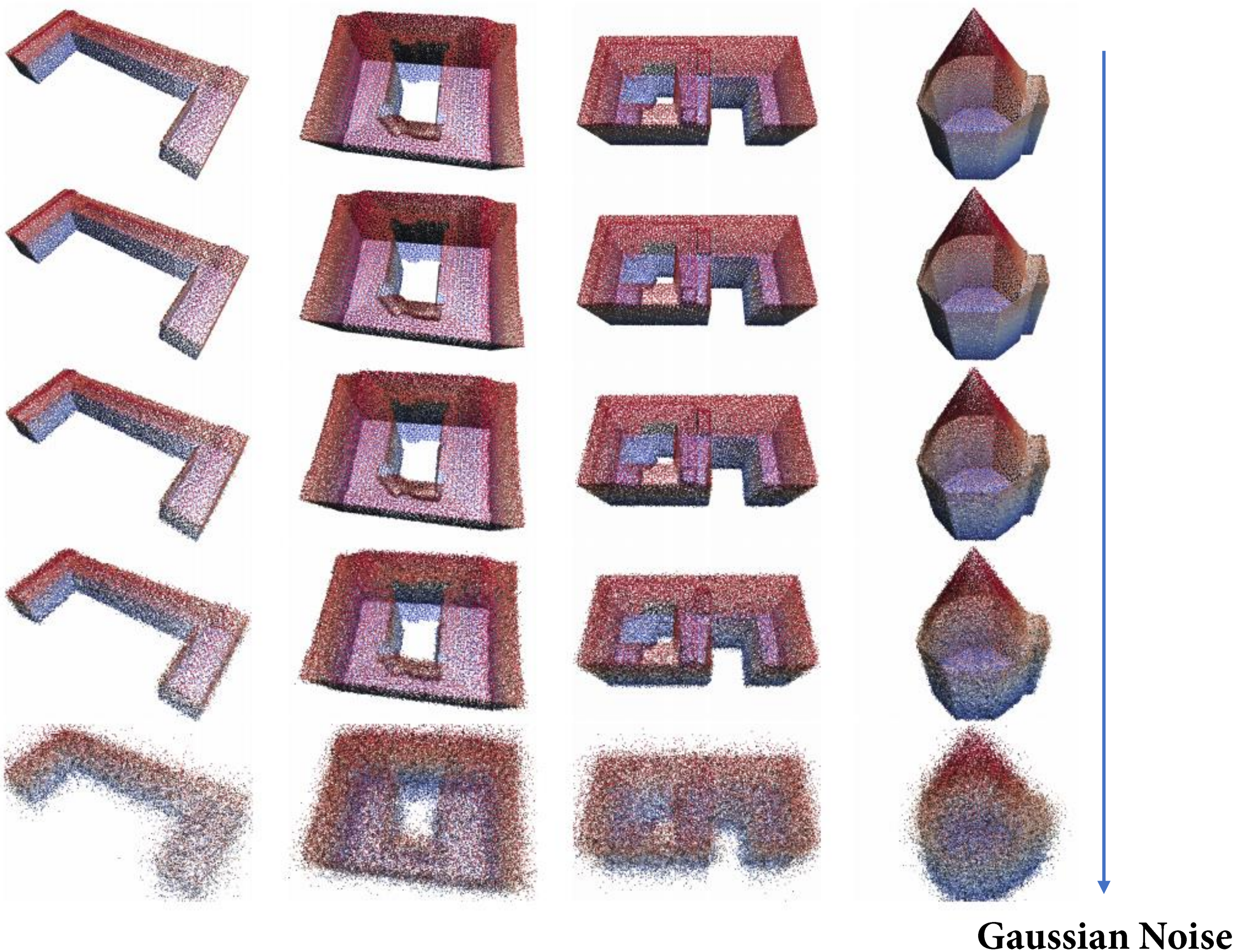
Simulated LiDAR scanning from CityGML models

- Point clouds
- Surface -> Sampled query points with signed distance values

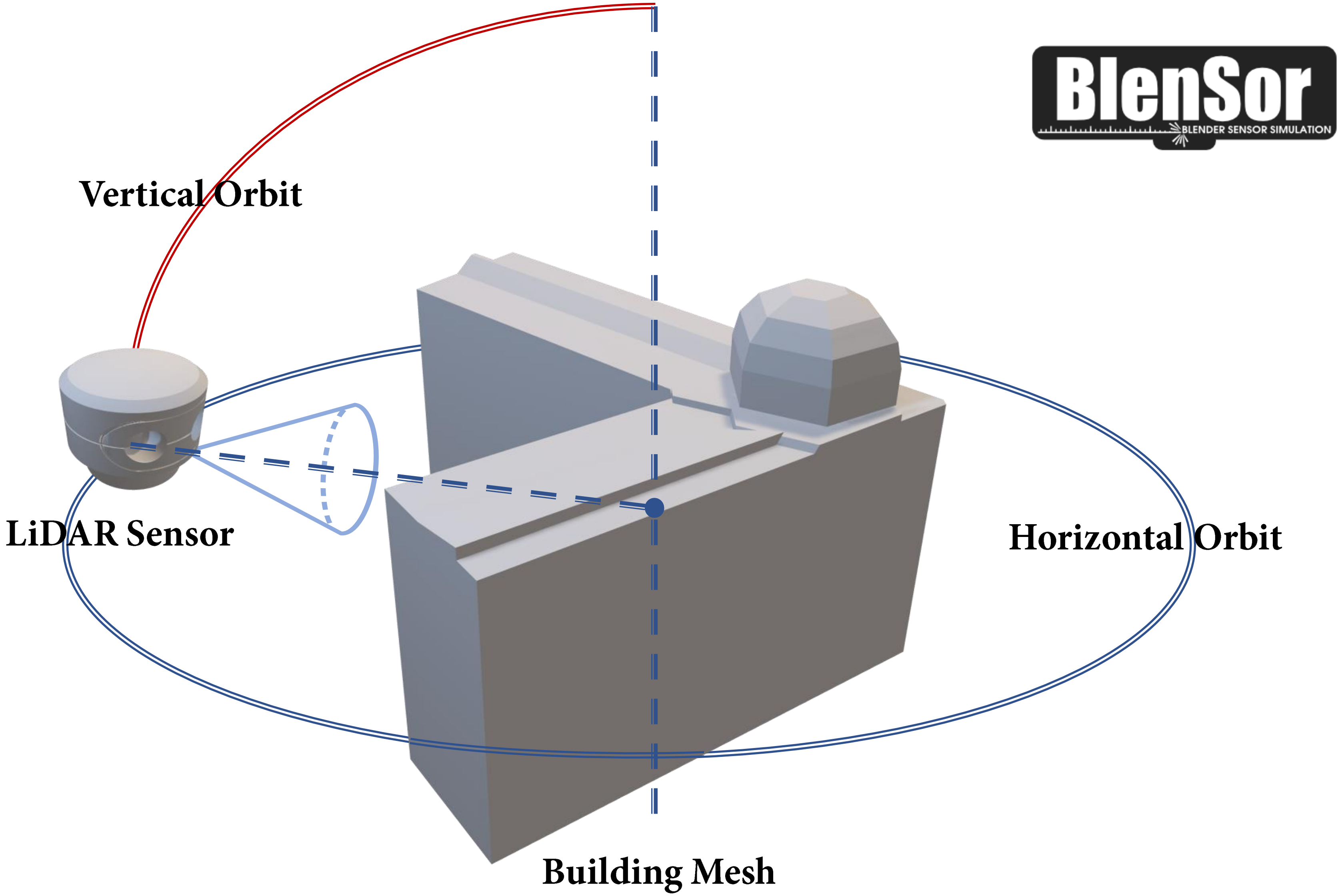
Datasets: *Helsinki full-view*



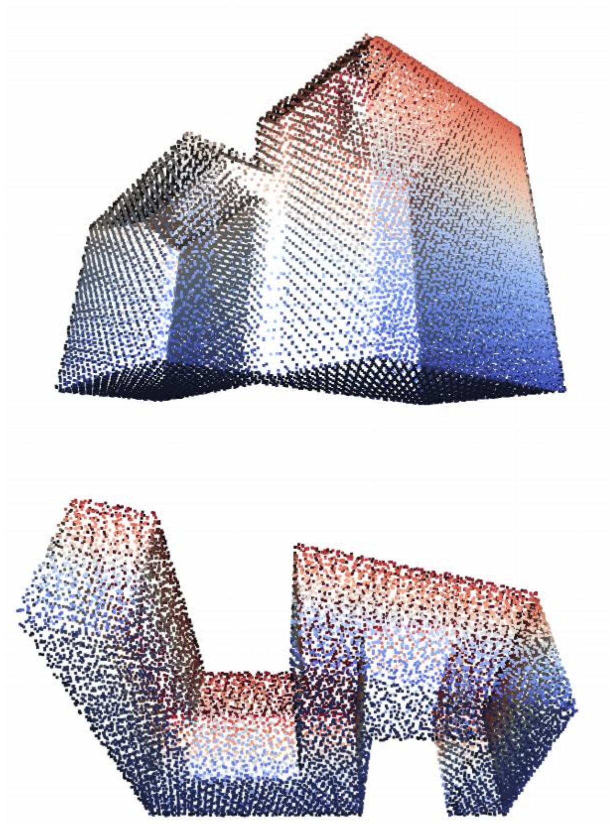
Datasets: *Helsinki full-view*



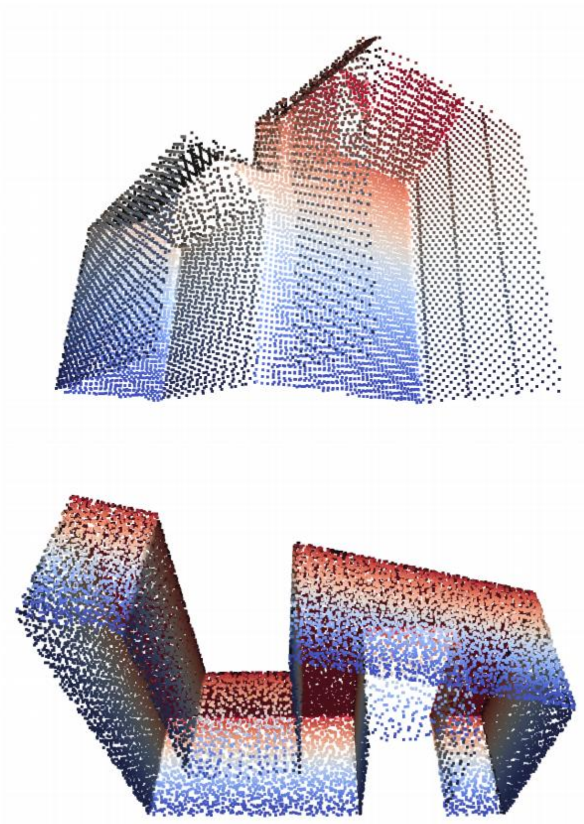
Datasets: *Helsinki no-bottom*



Datasets: *Helsinki*



Helsinki full-view



Helsinki no-bottom

Datasets: *Shenzhen*

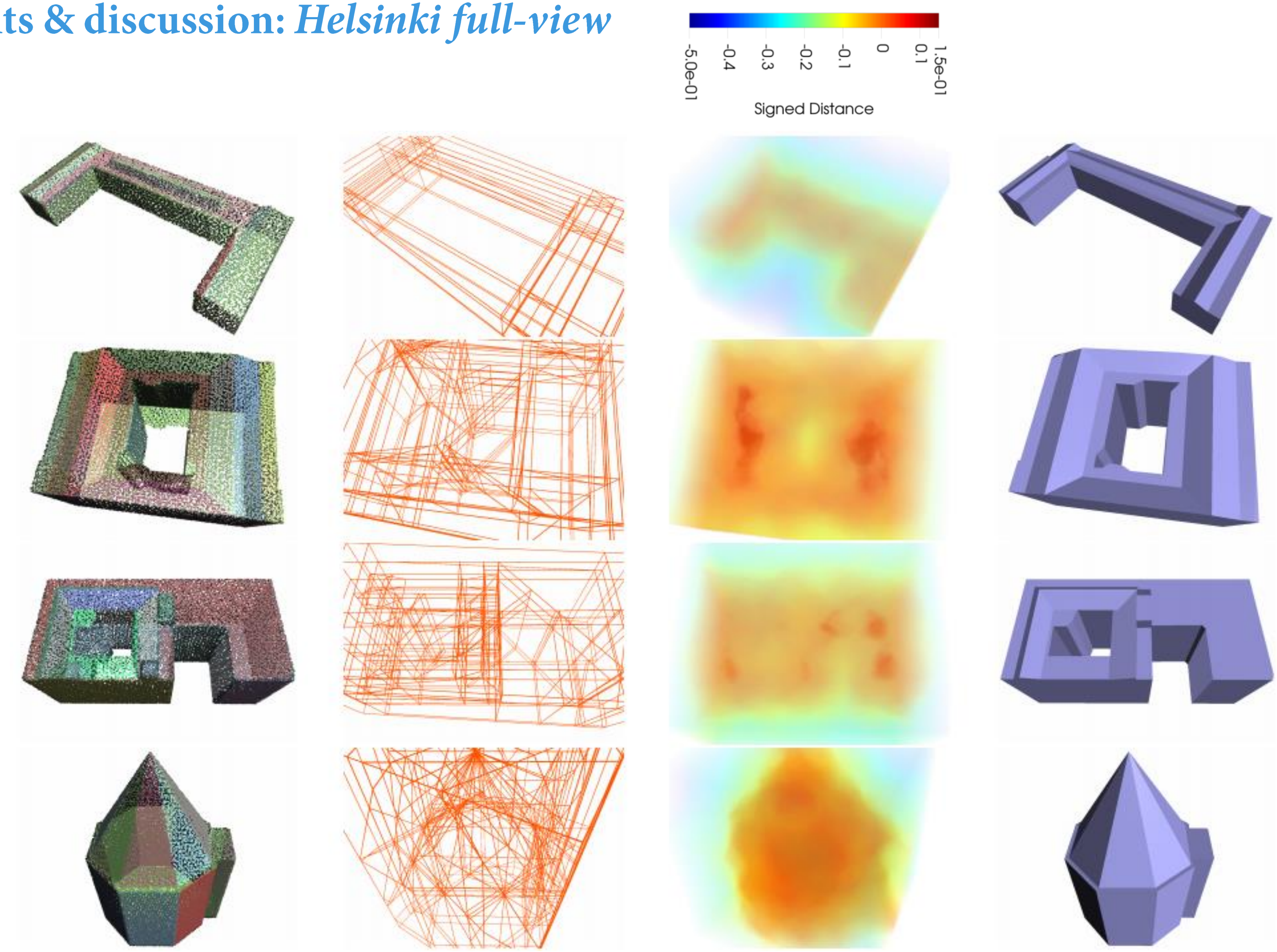
Data courtesy of Linfu Xie [Xie et al., 2021]

Datasets

Name	Type	Perspective			Quantity	Usage
		Top	Bottom	Lateral		
Helsinki full-view	Simulated LiDAR	✓	✓	✓	768	Training + evaluation
Helsinki no-bottom	Simulated LiDAR	✓	✗	✓	768	Training + evaluation
Shenzhen	Real-world MVS	✓	✗	✓	6	Evaluation

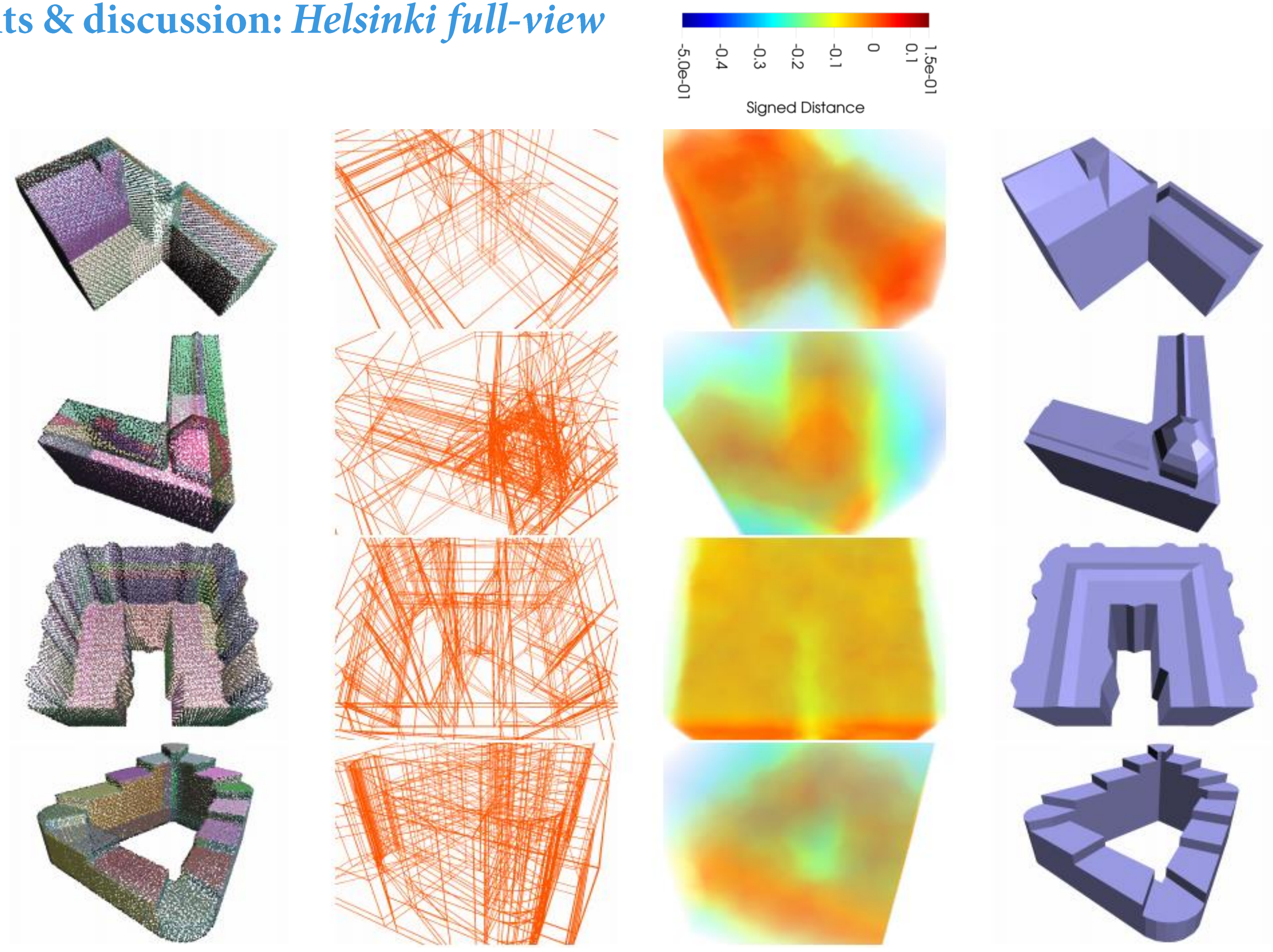
- **Introduction**
- **Related work**
- **Methodology**
- **Datasets**
- **Results and discussion**
- **Conclusions**

Results & discussion: *Helsinki full-view*



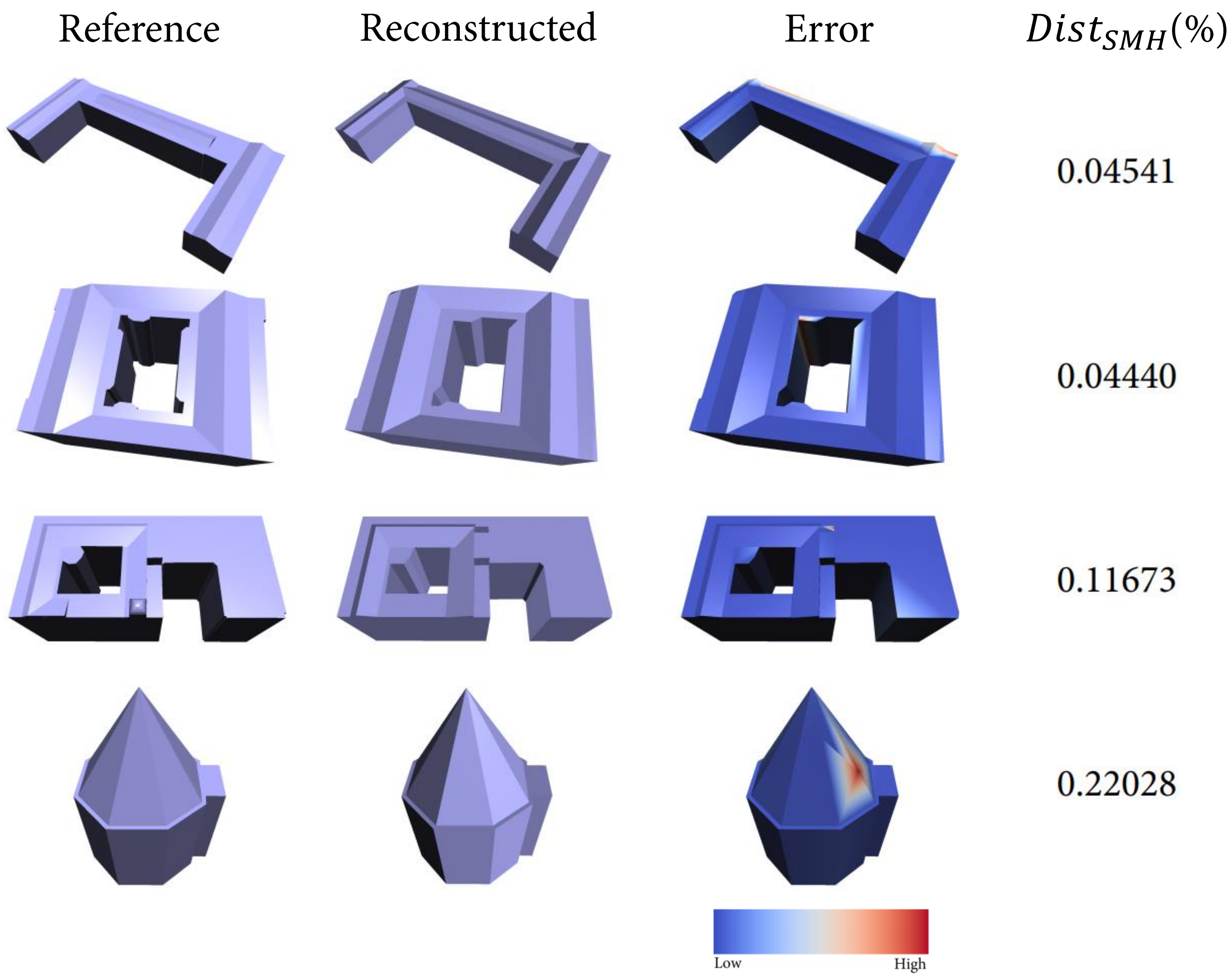
Point cloud | Candidate polyhedra | SDF | Reconstructed

Results & discussion: *Helsinki full-view*

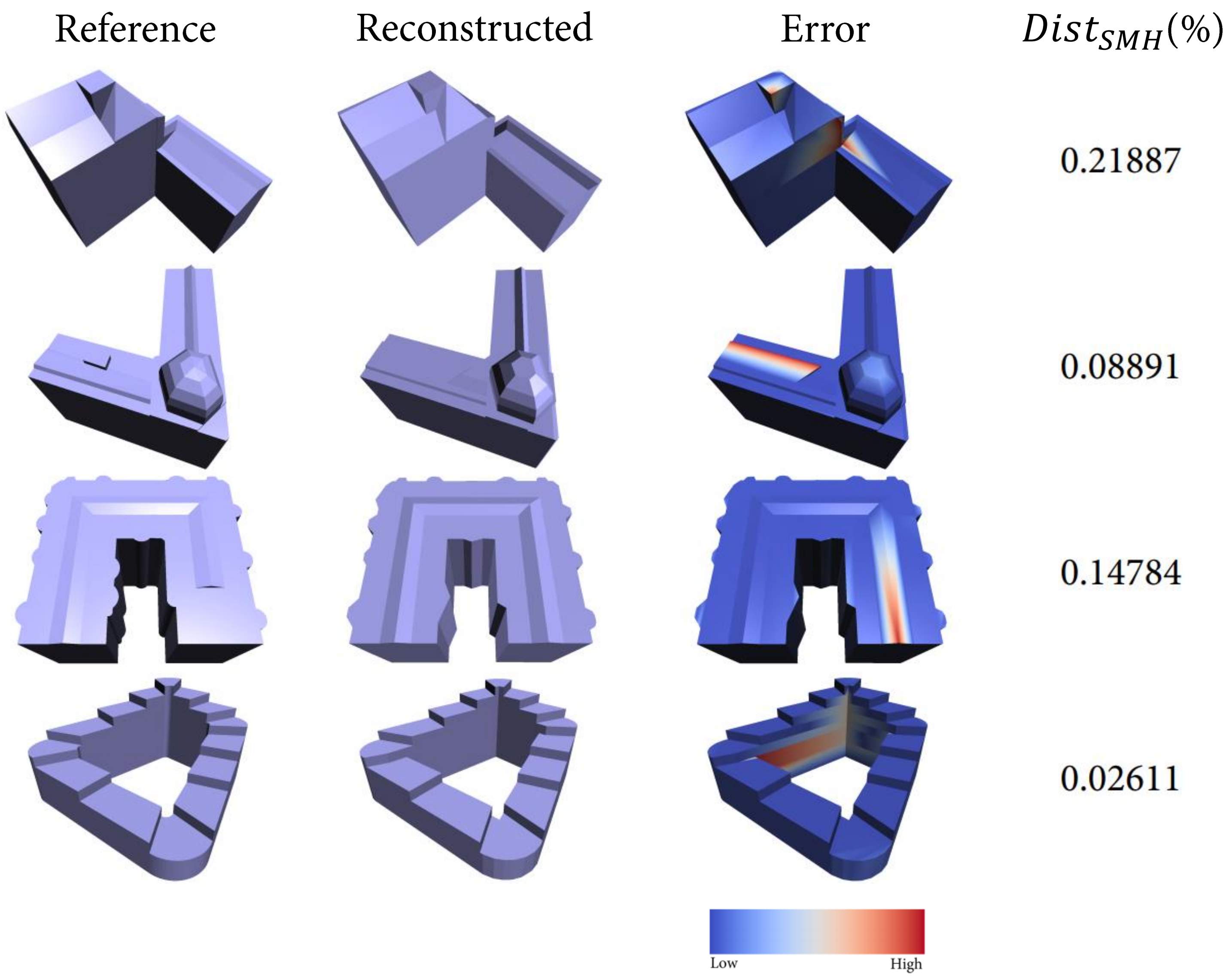


Point cloud Candidate polyhedra SDF Reconstructed

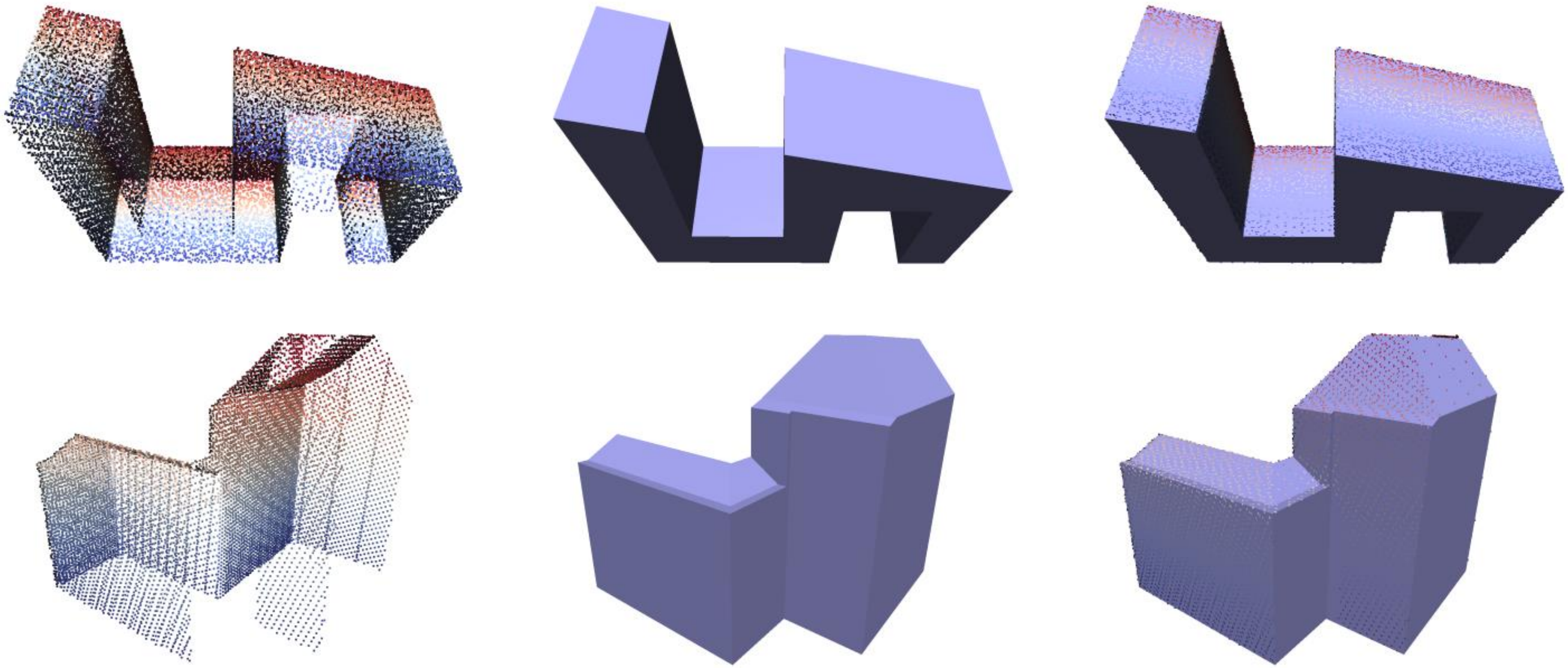
Results & discussion: *Helsinki full-view*



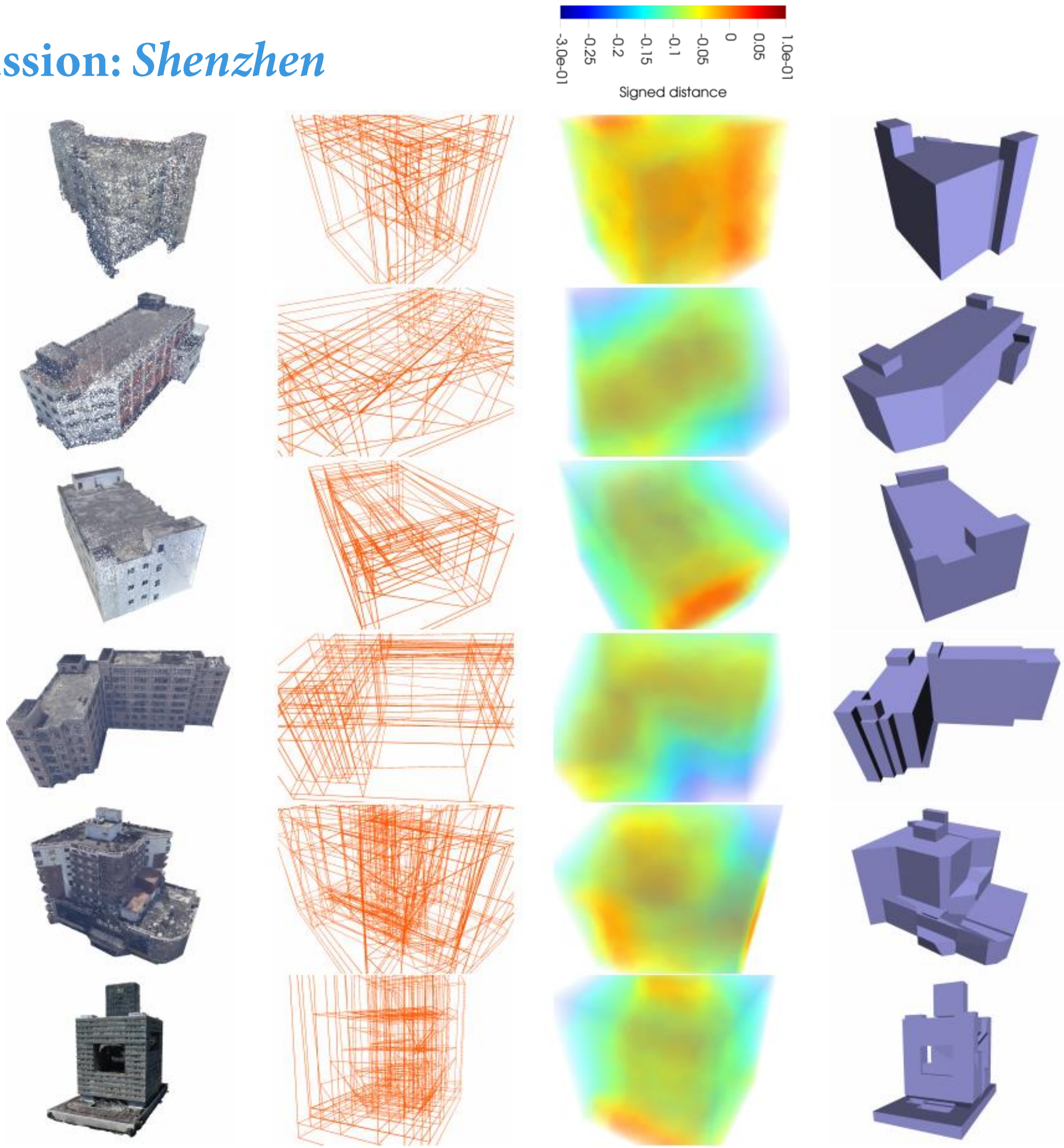
Results & discussion: *Helsinki full-view*



Results & discussion: *Helsinki no-bottom*

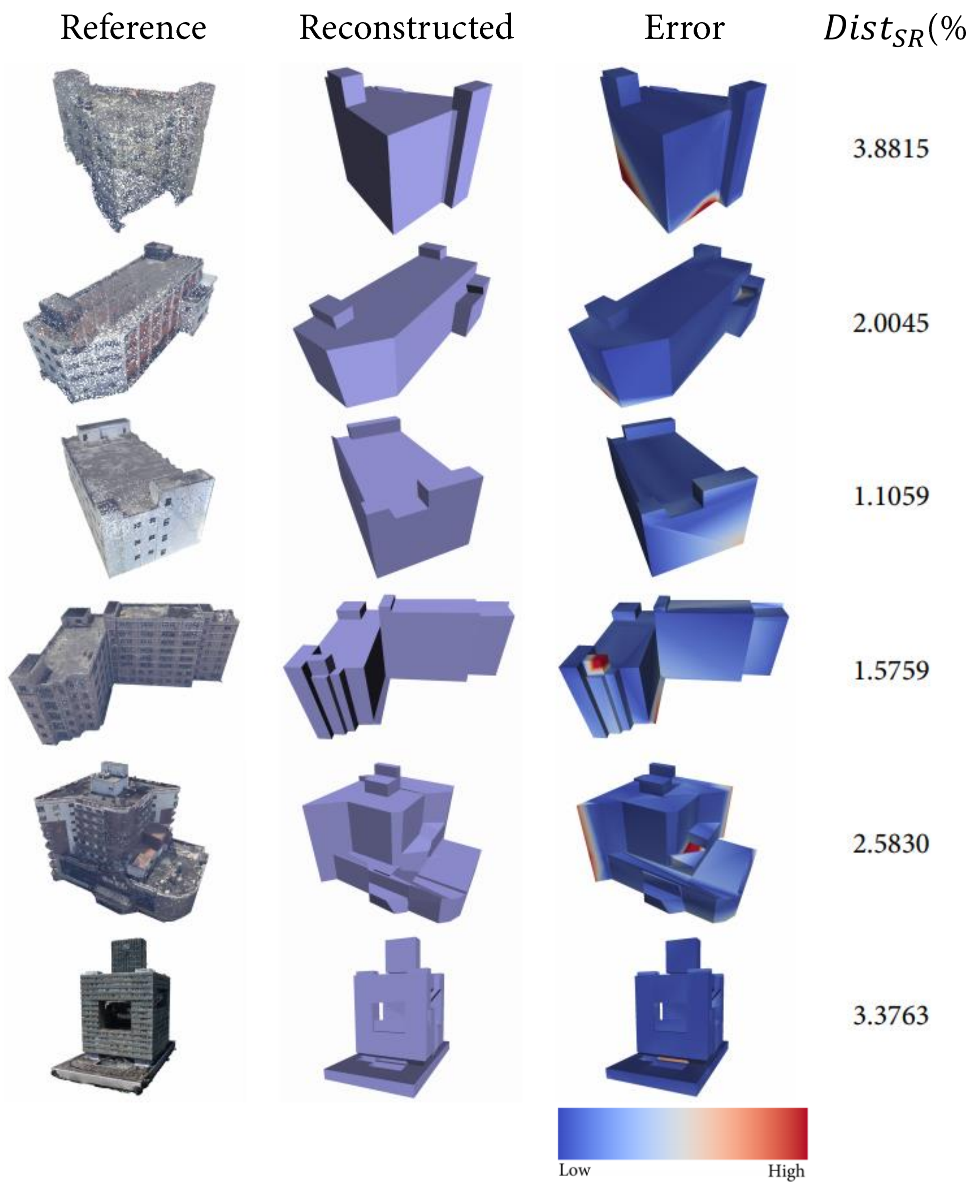


Results & discussion: *Shenzhen*

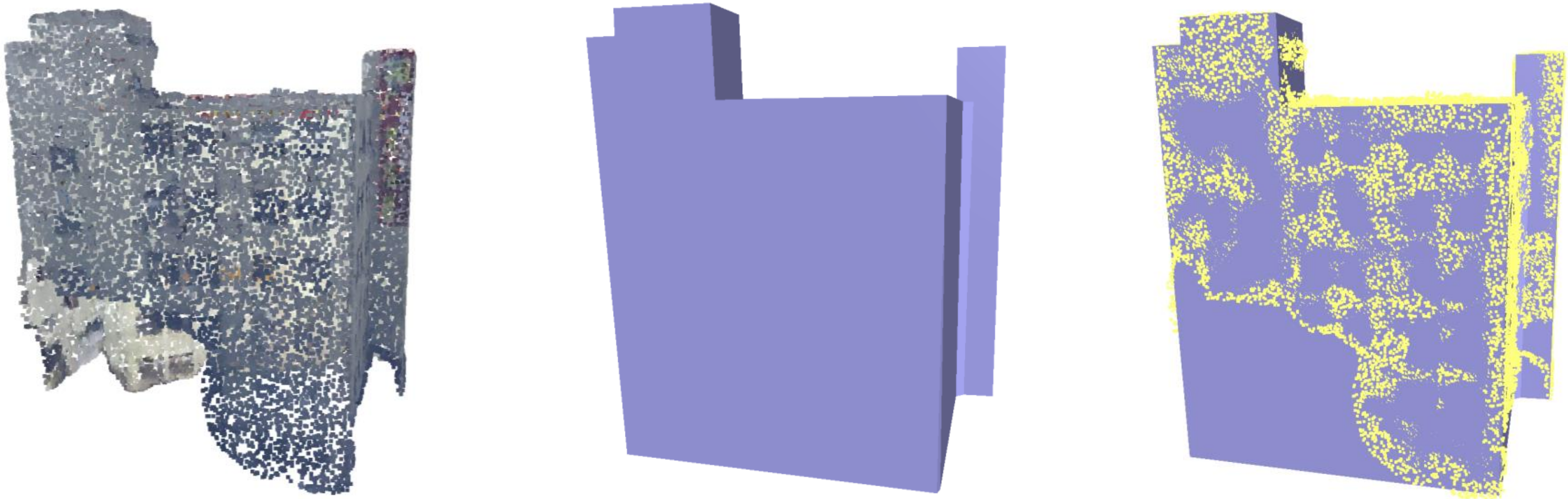


Point cloud | Candidate polyhedra | SDF | Reconstructed

Results & discussion: *Shenzhen*

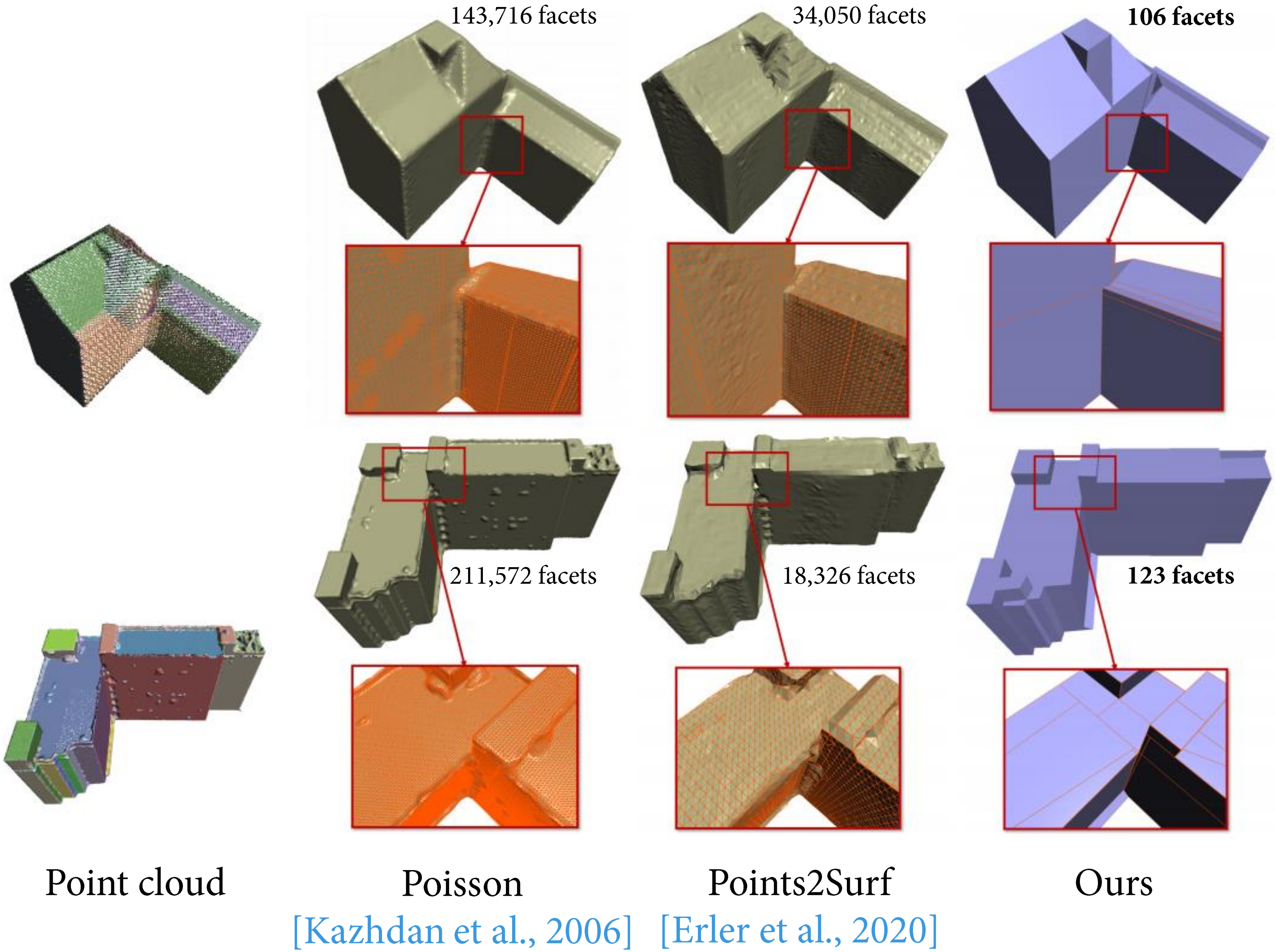


Results & discussion: *Shenzhen*



Reconstruction from insufficient scans

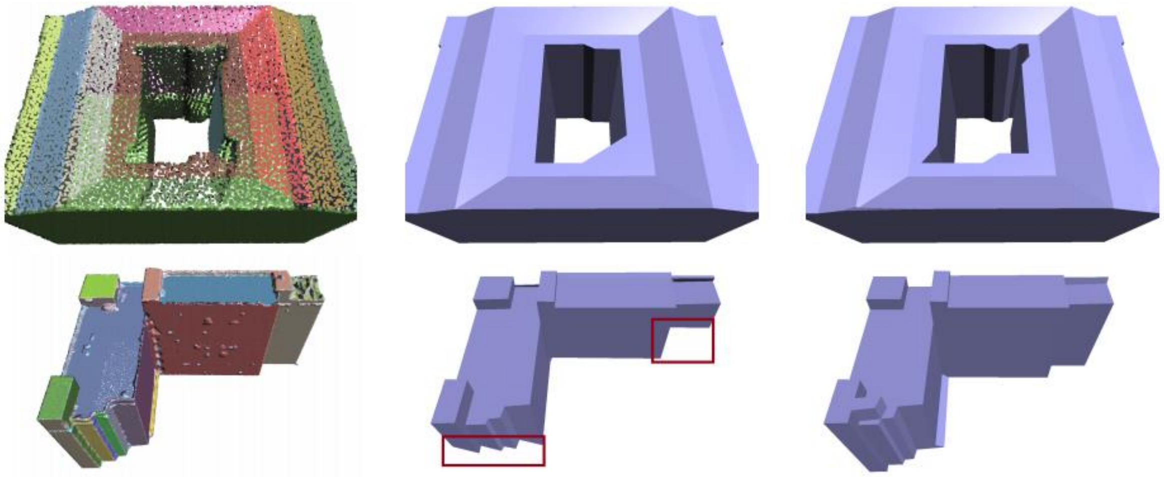
Results & discussion: Comparison with smooth reconstruction



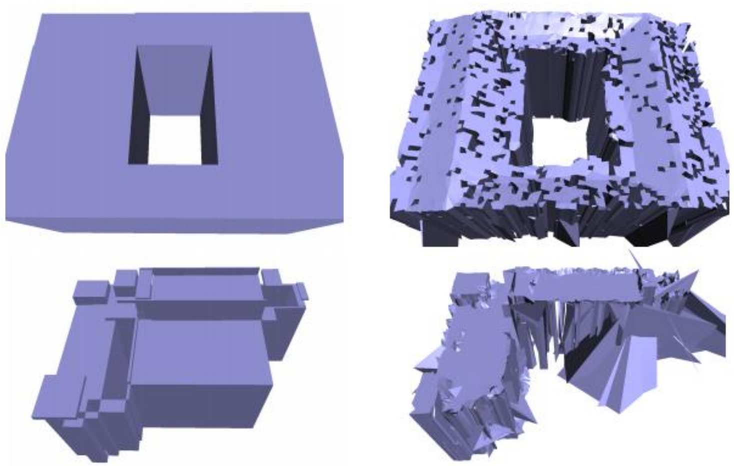
Results & discussion: Comparison with piecewise-planar reconstruction

Helsinki full-view

Shenzhen

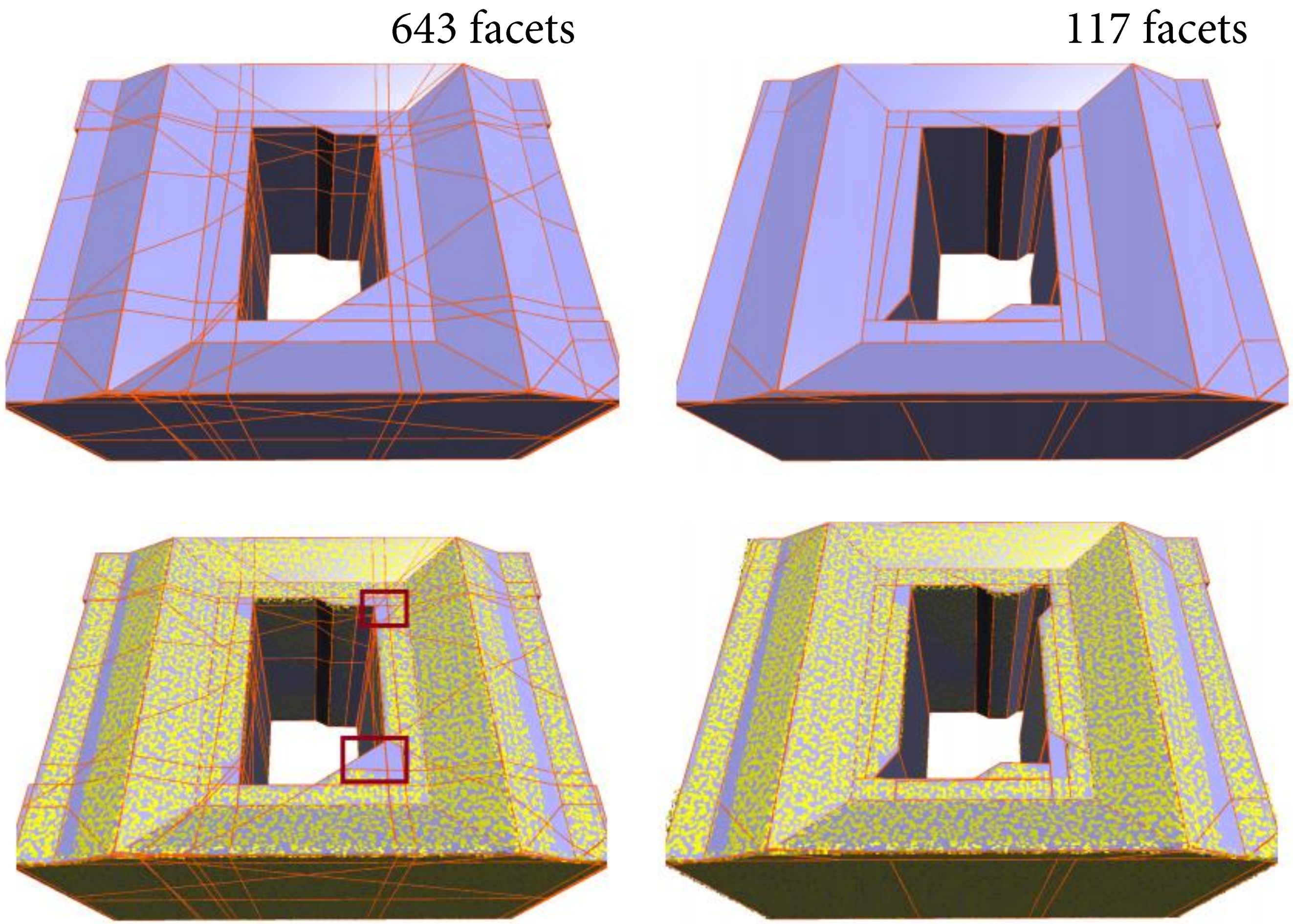


Point cloud　　PolyFit [Nan and Wonka, 2017]　　Ours



Manhattan-world [Li et al., 2016b]　　2.5D DC [Zhou and Neumann, 2010]

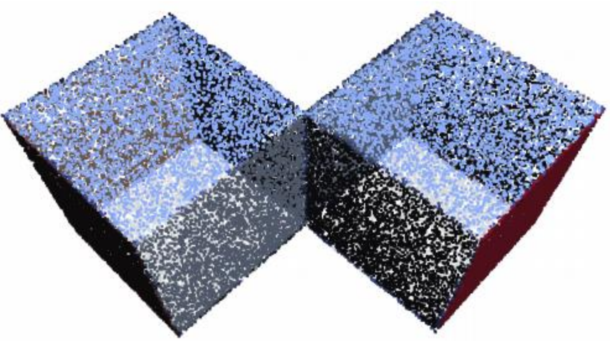
Results & discussion: Comparison



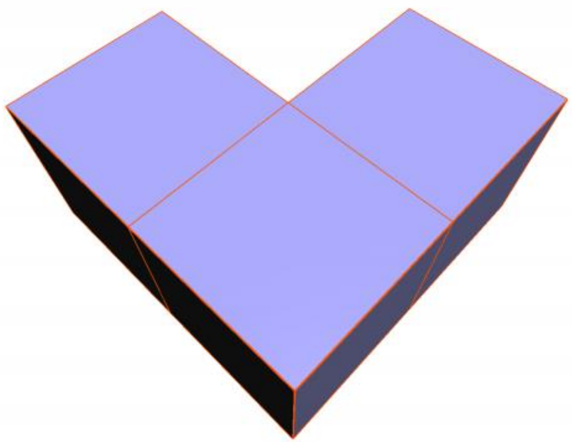
PolyFit
[Nan and Wonka, 2017]

Ours

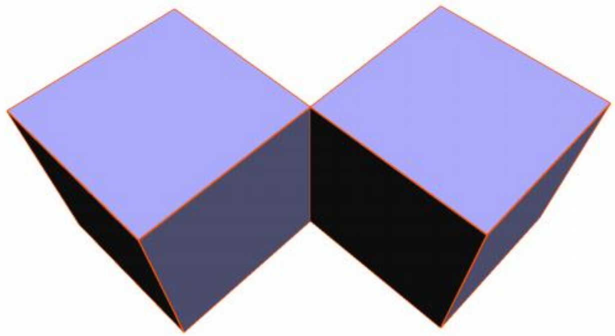
Results & discussion: Comparison



Point cloud

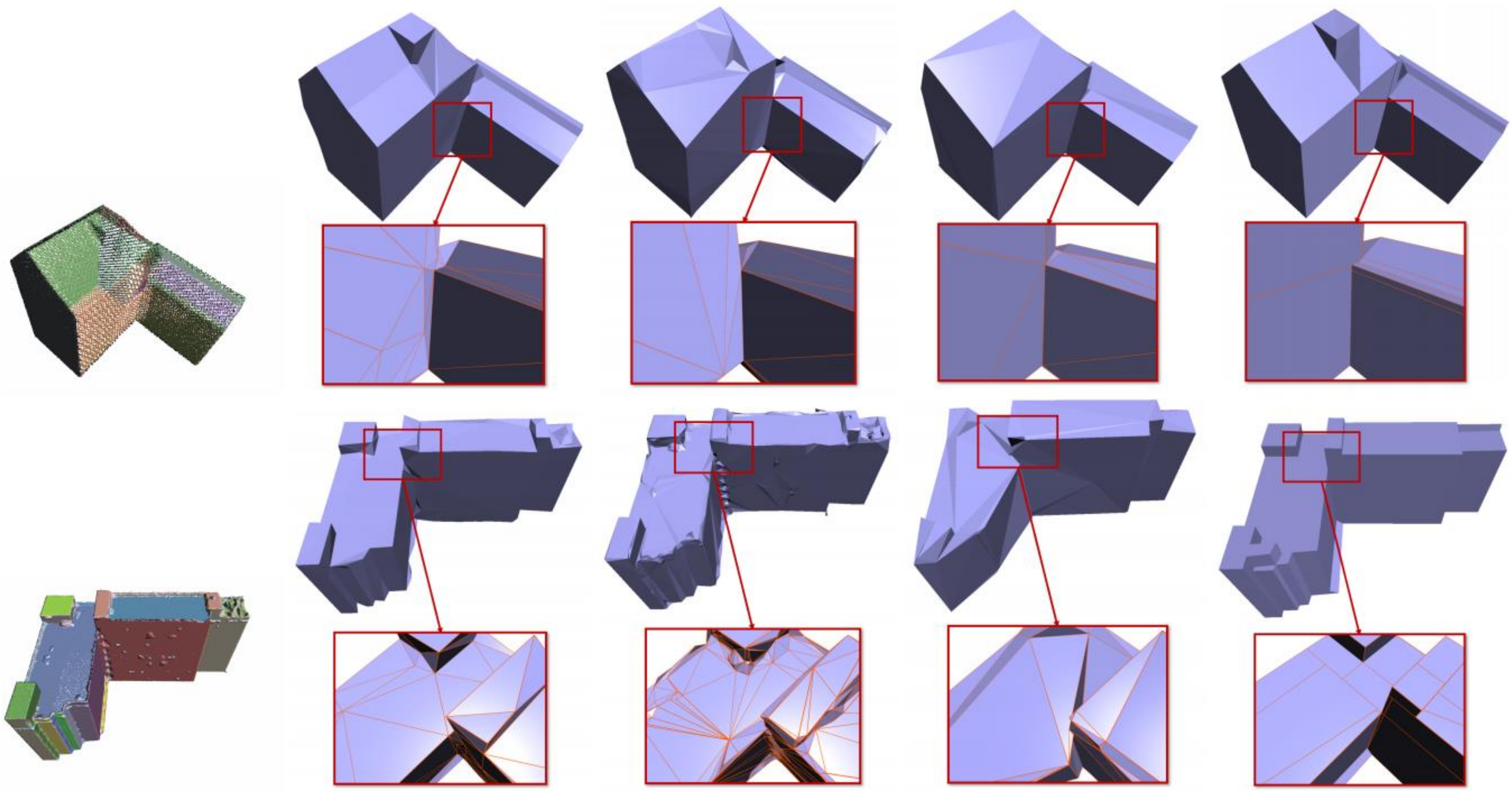


PolyFit
[Nan and Wonka, 2017]



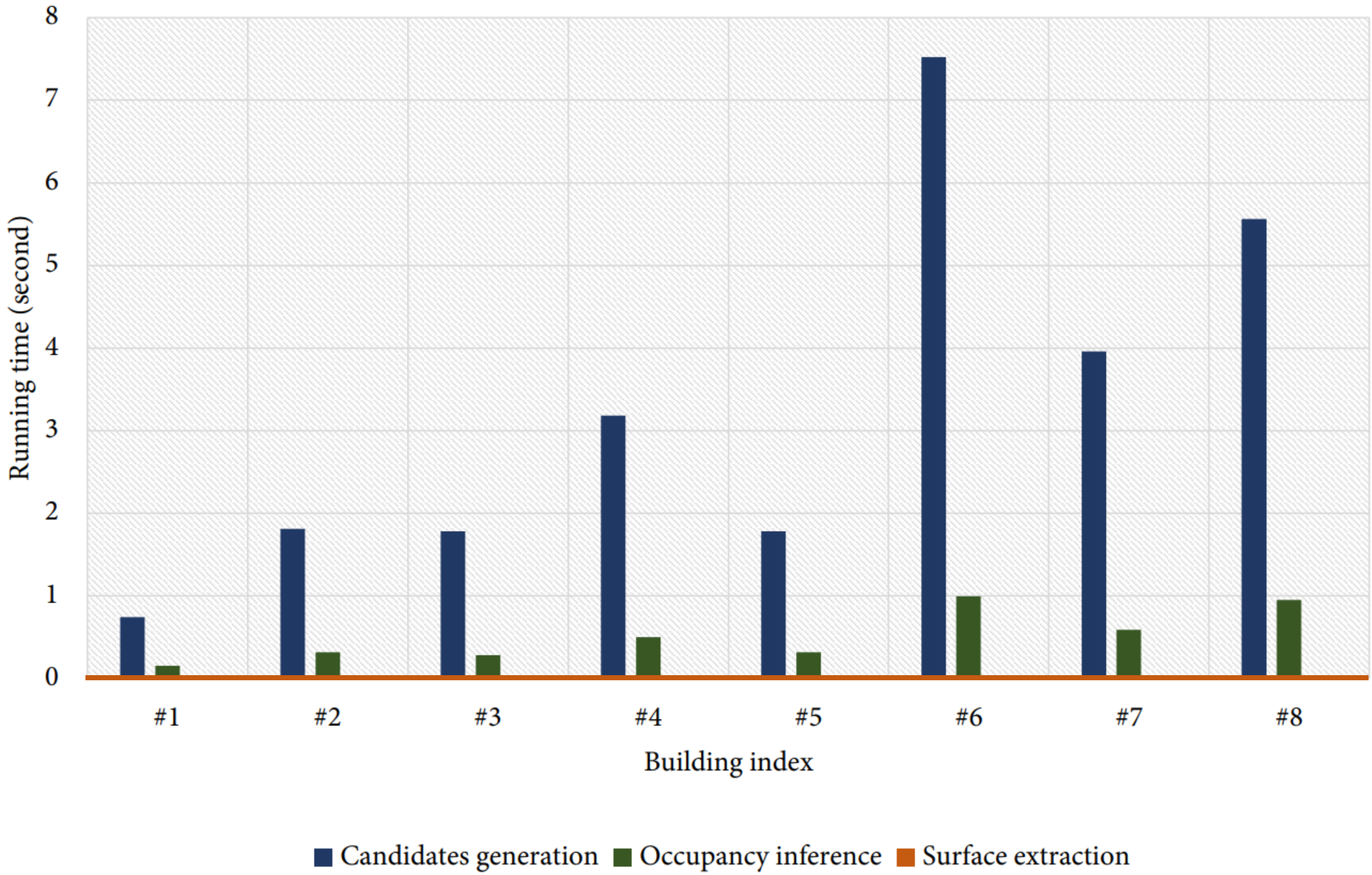
Ours

Results & discussion: Comparison with surface approximation methods

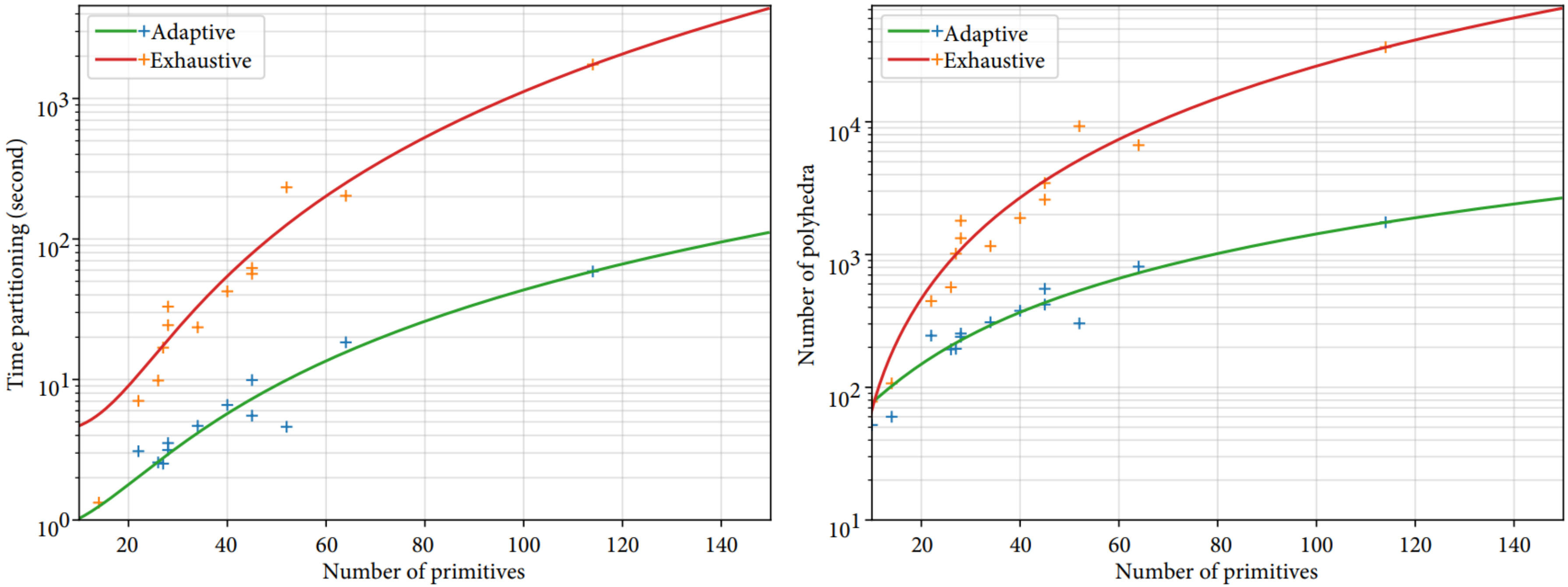


Point cloud | QEM [Garland and Heckbert, 1997] | SAMD [Salinas et al., 2015] | VSA [Cohen-Steiner et al., 2004] | Ours

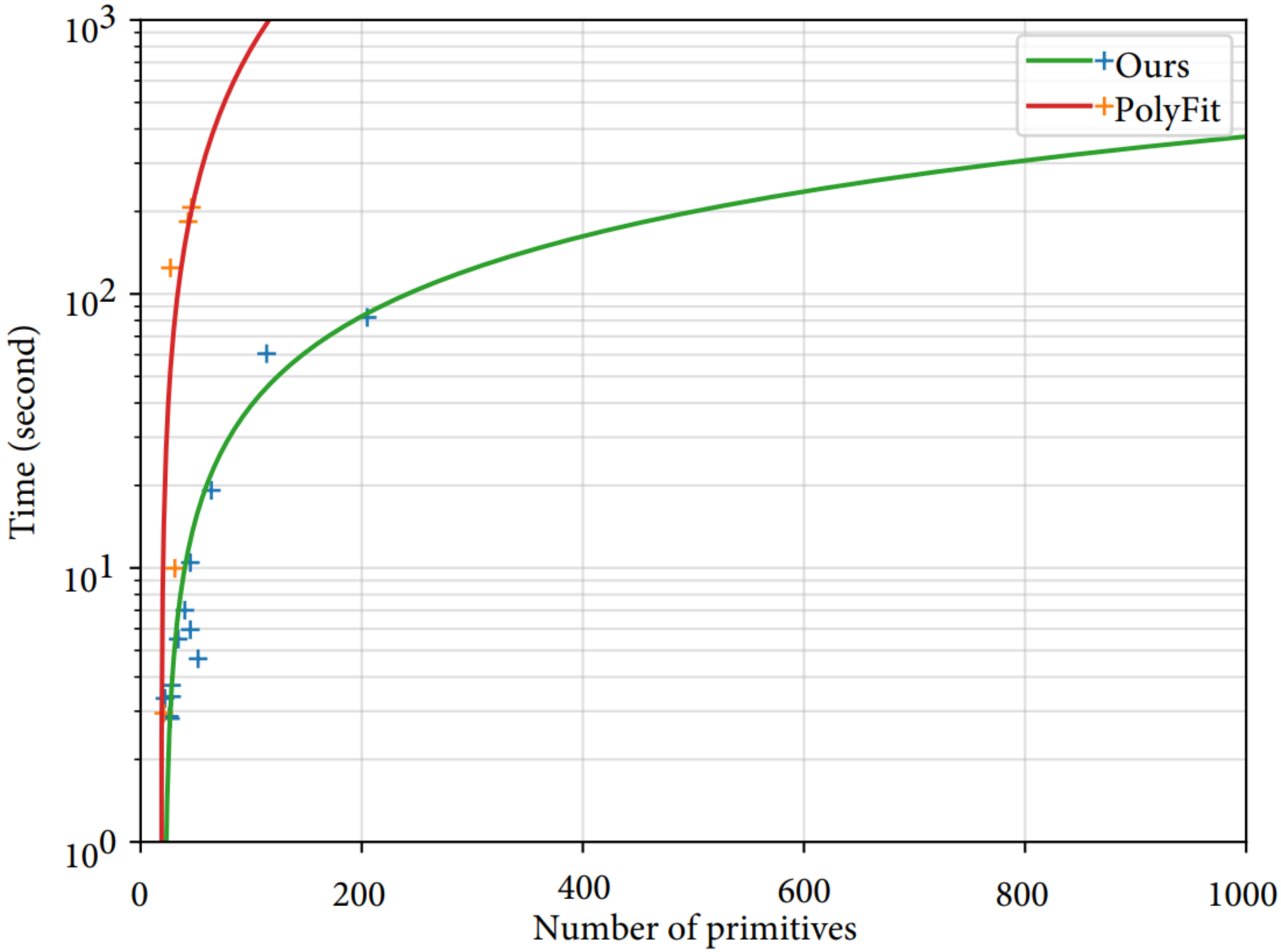
Results & discussion: Efficiency



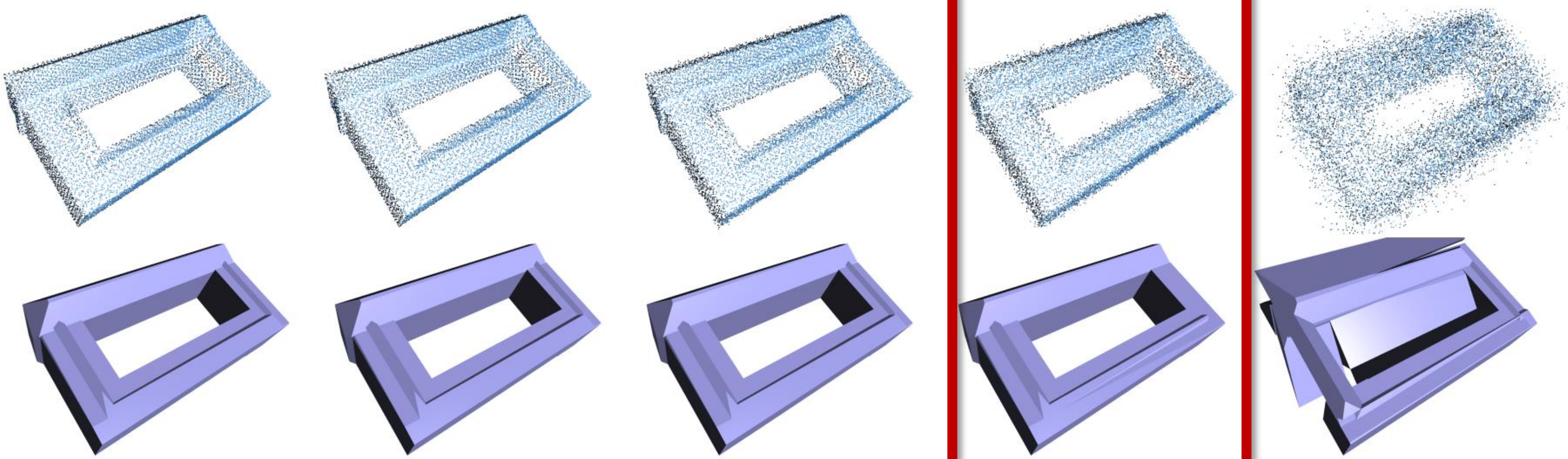
Results & discussion: Efficiency



Results & discussion: Scalability



Results & discussion: Robustness to noise



0R

0.001R

0.005R

0.010R

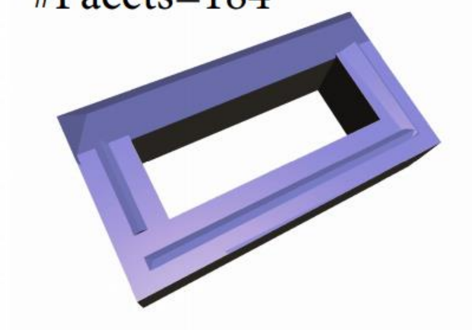
0.050R

1 meter against 100 meter-sided building

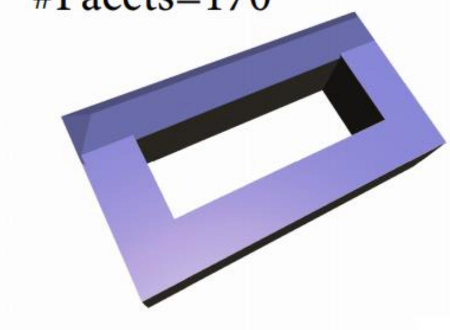
Results & discussion: Impact of parameter λ

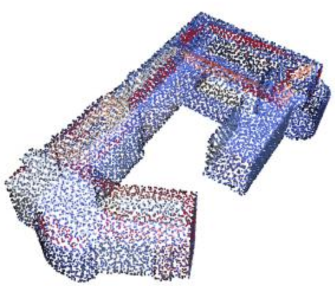
$$E(x) = D(x) + \lambda V(x)$$

$Dist_{SMH}$=0.04268
#Facets=184



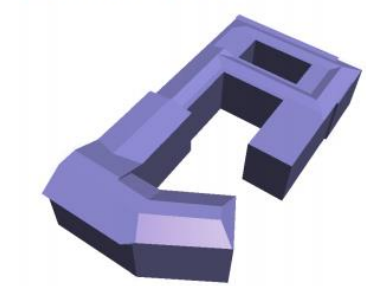
$Dist_{SMH}$=0.04324
#Facets=170





$Dist_{SMH}$=0.07495
#Facets=236

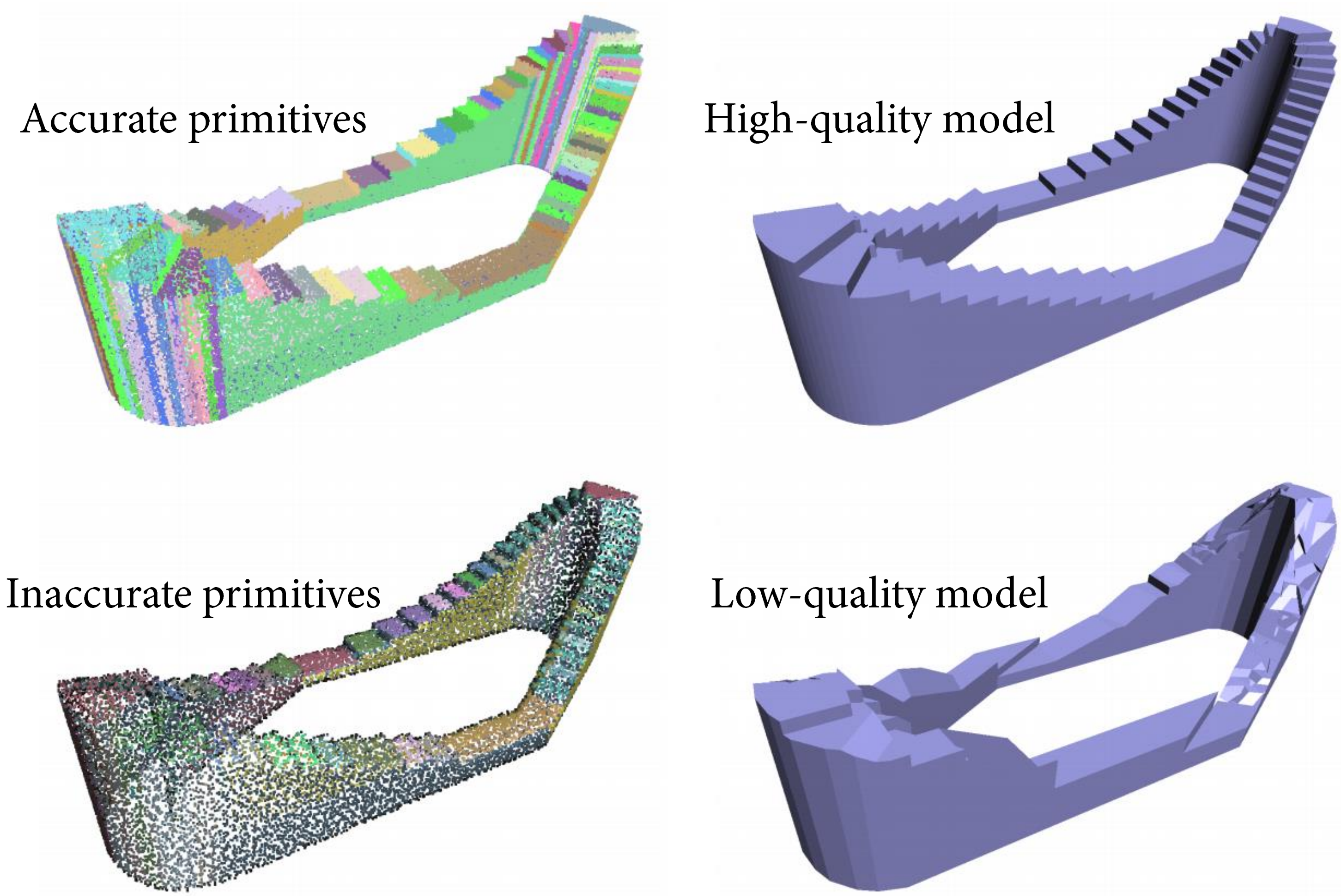
$Dist_{SMH}$=0.07522
#Facets=220



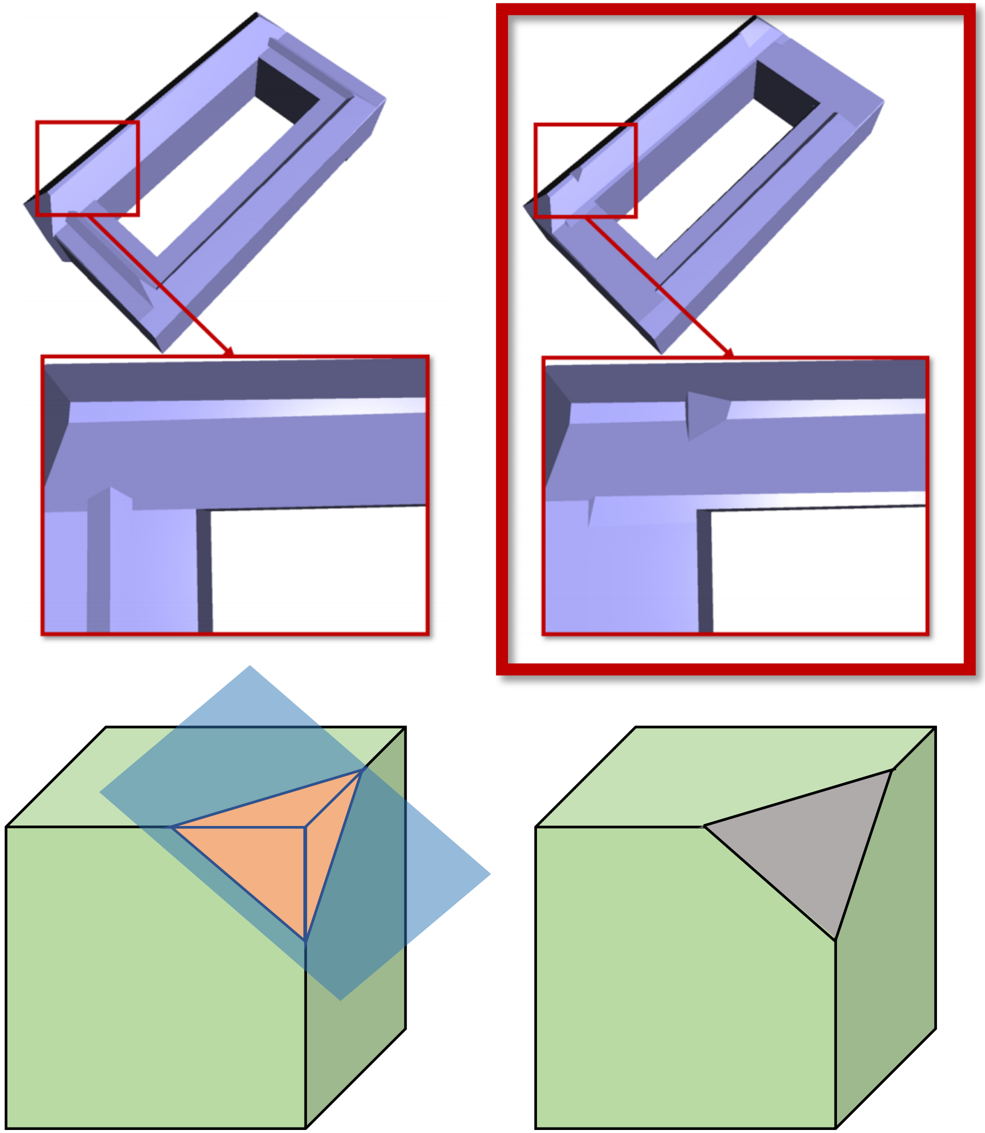
$\lambda = 0.002$

$\lambda = 0.01$

Results & discussion: limitations

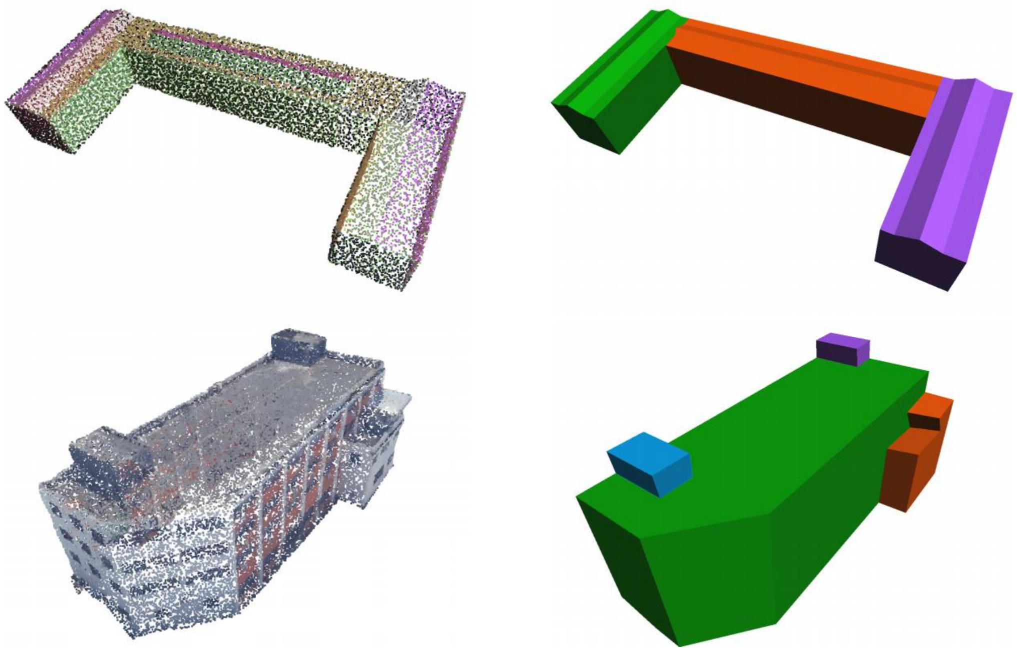


Results & discussion: limitations



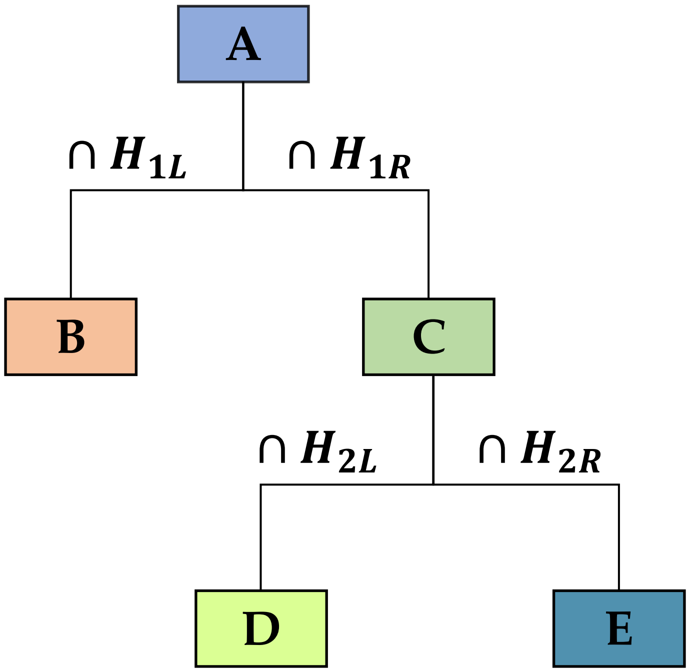
'Caved' artefact

Results & discussion: Applications



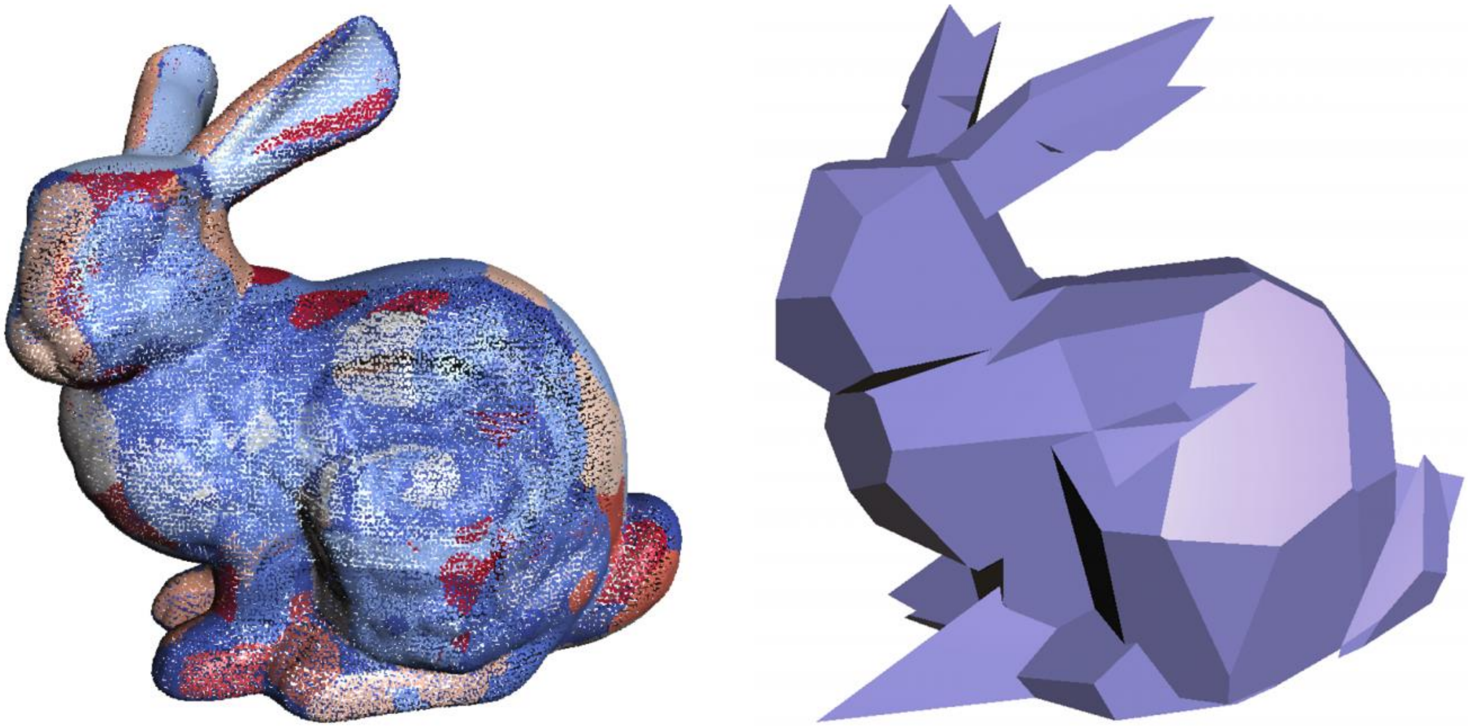
Building component analysis

BSP-tree



Results & discussion: Applications

- Compression
- Physical Simulation



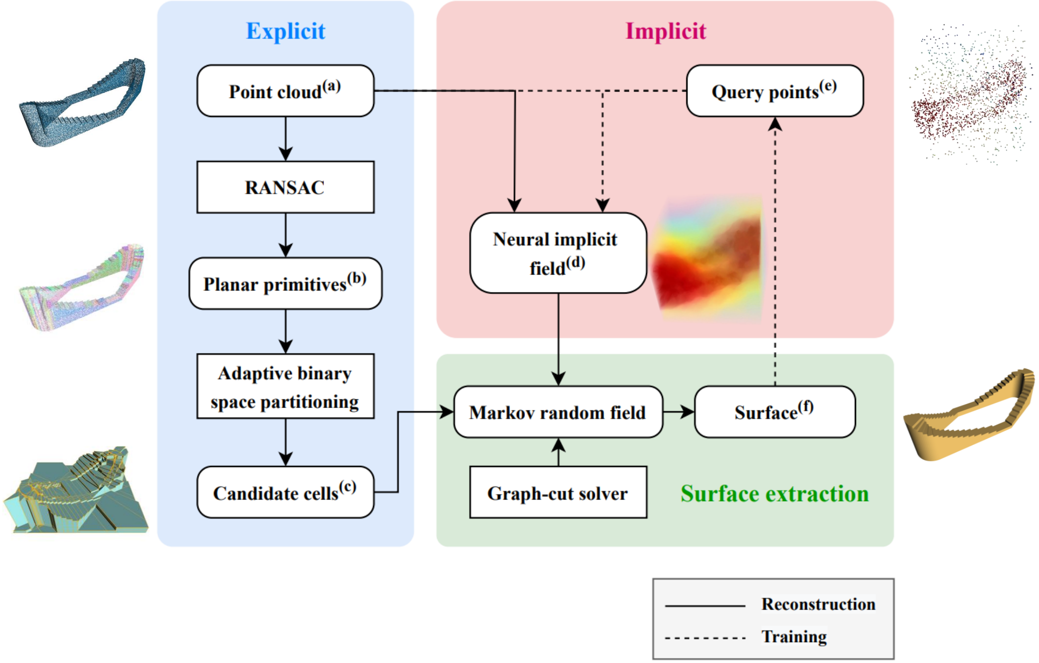
Generic shape reconstruction

- **Introduction**
- **Related work**
- **Methodology**
- **Results and discussion**
- **Conclusions**

Conclusion: Research question revisited

How can deep implicit fields be used for compact building model reconstruction?

- Compactness and watertightness ✓
- Generalisation ✓
- Robustness ✓
- Advantages & disadvantages ✓



Conclusion

Contributions

- A learning-based framework to incorporate deep implicit fields into piecewise-planar urban building reconstruction
- An adaptive space partitioning strategy for cell complex construction
- An MRF formulation for efficient surface extraction
- Open synthetic building point cloud dataset

Conclusion

Future work

- End-to-end neural network architecture
- Extension to more general primitives

……

Conclusion

Source code

- https://github.com/chenzhaiyu/absp
- https://github.com/chenzhaiyu/points2poly

Dissemination

- Thesis & Slides available at TU Delft Repositories
- ISPRS Journal manuscript in progress

Learning to Reconstruct Compact Building Models from Point Clouds with Deep Implicit Fields

Zhaiyu Chen[a], Seyran Khademi[b], Hugo Ledoux[a] and Liangliang Nan[a,*]

[a] 3D Geoinformation Research Group, Faculty of Architecture and the Built Environment, Delft University of Technology, 2628 BL, Delft, The Netherlands
[b] Theory of Architecture and Digital Culture Group, Faculty of Architecture and the Built Environment, Delft University of Technology, 2628 BL, Delft, The Netherlands

ARTICLE INFO

Keywords:
3D reconstruction
Point cloud
Compact building model
Deep neural network
Implicit field

ABSTRACT

We present a novel framework for reconstructing compact, watertight, polygonal building models from point clouds. Our method comprises three components: *(a)* a cell complex is generated via adaptive space partitioning that provides a polyhedral embedding as the candidate set; *(b)* an implicit field is learnt by a deep neural network that facilitates building occupancy estimation; *(c)* a Markov random field is formulated for surface extraction via combinatorial optimisation. We extensively evaluate the proposed method in comparison with state-of-the-art methods in shape reconstruction, surface approximation and geometry simplification. Experimental results reveal that, with our neural-guided strategy, high-quality building models can be obtained with significant advantages over fidelity, compactness and computational efficiency. Our method shows robustness to noise and insufficient measurements, and generalise well directly from synthetic scans to real-world measurements.

1. Introduction

Three-dimensional (3D) building models play a pivotal role in shaping the digital twin of our world, and are facilitating various intelligent applications in urban planning (Herbert and Chen, 2015), solar potential analysis (Machete et al., 2018), environmental simulation (Stoter et al., 2020), etc. Recently, with the development of augmented and virtual reality applications, the demand for high-quality building modelling is growing rapidly (Blut and Blankenbach, 2021).

Most reconstruction methods are dedicated to smooth surfaces represented as dense triangles, irrespective of piecewise planarity that exhibits in the built environment (Kazhdan et al., 2006; Erler et al., 2020). Simplification is therefore required as a follow-up procedure to convert the smooth surface into a compact one (Garland and Heckbert, 1997; Cohen-Steiner et al., 2004; Salinas et al., 2015; Bouzas et al., 2020). Although some works claim the possibility of reconstructing piecewise-planar shapes directly from point clouds, they suffer from serious scalability issues (Boulch et al., 2014; Mura et al., 2016; Nan and Wonka, 2017). In this work, we aim at efficiently reconstructing compact building surfaces directly from point clouds.

3D shapes are not confined to as explicit representations (e.g., point cloud, surface mesh, voxels), but can be encoded implicitly in a function space. A signed distance function (SDF), for instance, can describe an implicit field, where the surface of a shape is implicitly interpreted as zero-set of the SDF. A learnable indicator function of the SDF takes as input a query point and yields an indication on whether the point belongs to the shape. The explicit geometry is then often extracted from the field via computational-expensive isosurfacing (Mescheder et al., 2019). Compared with explicit expressions that are heterogeneously distributed, this homogeneous functional representation is particularly favourable for geometric machine learning. Especially recently, the scheme for learning in the function space has shown its competence in 3D geometric modelling (Park et al., 2019).

In this paper, we propose a novel framework for reconstructing compact, watertight, polygonal building meshes from point clouds by incorporating implicitly encoded function space with explicitly constructed geometry. The explicit geometry provides a polyhedral embedding as the candidate set, from which extraction of the building's surface is neural-guided by a learnt implicit field. We formulate a Markov random field (MRF) to introduce configurable surface complexity, and solve this optimisation problem using an efficient graph-cut solver. With our neural-guided strategy, we demonstrate that high-quality building models can be obtained with significant advantages over fidelity, compactness and computational efficiency against state-of-the-art methods in shape reconstruction, surface approximation and geometry simplification.

The main contributions of this paper are as follows:

(i) A learning-based framework for compact building model reconstruction. To the best of our knowledge, this is the first work where a deep implicit field is explored for building reconstruction. Our method shows significant performance and quality advantage over state-of-the-art methods for urban building reconstruction, especially for complex building models.

(ii) An adaptive space partitioning solution for generating a cell complex of candidate polyhedra. Compared with the exhaustive baseline, our adaptive strategy can efficiently partition the space, minimising redundant

*Corresponding author
zhaiyu.chen@outlook.com (Z. Chen); s.khademi@tudelft.nl (S. Khademi); h.ledoux@tudelft.nl (H. Ledoux); liangliang.nan@tudelft.nl (L. Nan)

Thanks! Questions?