Edge-aware simplification of roof and façade point clouds into a uniformly dense point cloud

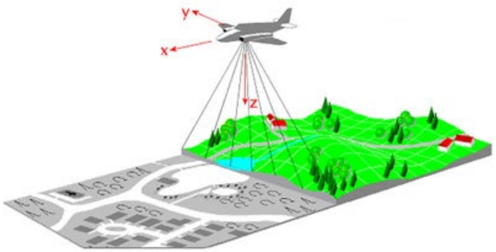
Graduation professor: Sisi Zlatanova
Daily supervisor: Elmar Eisemann & Bert Buchholz
Co-reader: Ben Gorte
MSc student: Jiale (Carl) Chen

Content

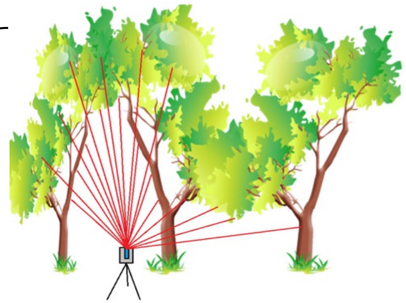
Content

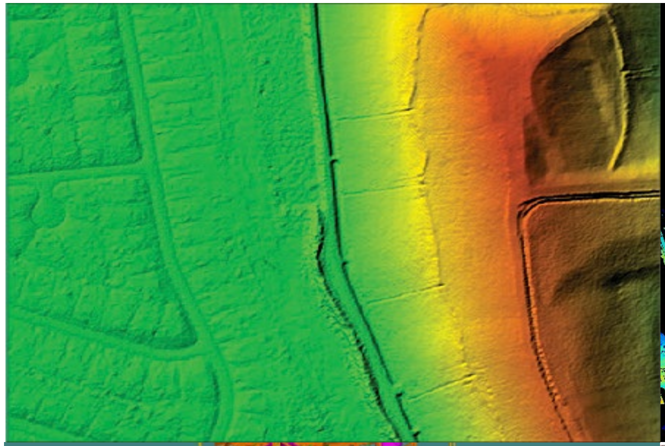
INTRODUCTION

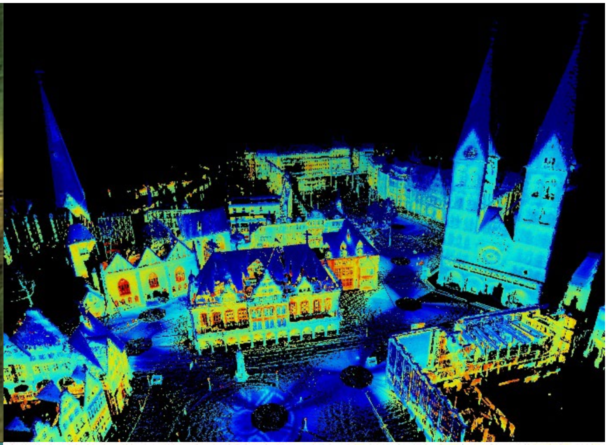
Roof

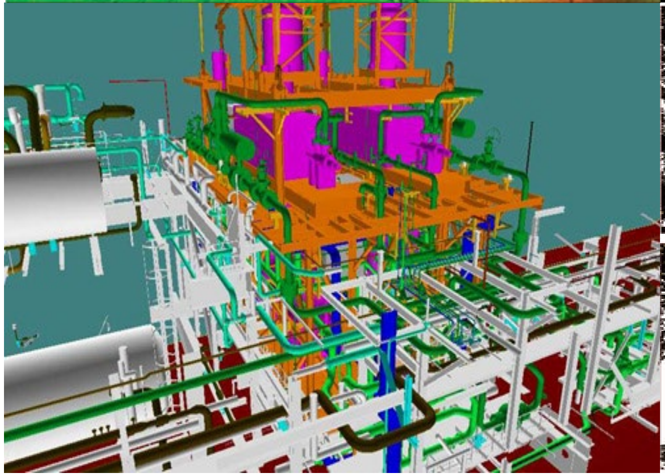


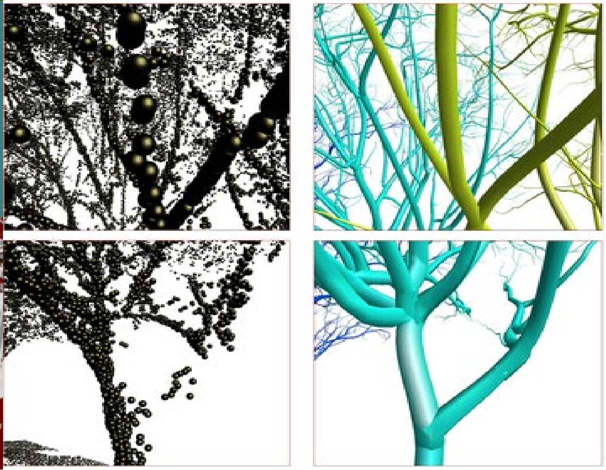
Facade











MOTIVATION

- Combine roof and façade point clouds

- Outlier removal
  - Pre-processing for simplification

- Simplification
  - **Noise smoothing**
  - **Edges preservation**: indicate geometry skeleton; useful for reconstruction, segmentation and other applications.
  - **Uniform density***:* required in multi-level smoothing and texture synthesis

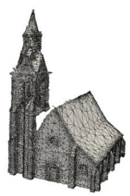
MOTIVATION

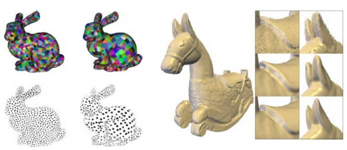
- Problems in existing methods

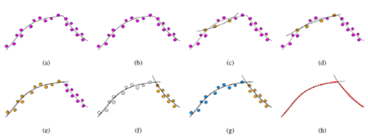
- **Outlier removal**

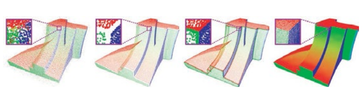
  - Not able to remove both single outliers and small cluster of outliers without over-removing artifacts

- **Simplification**







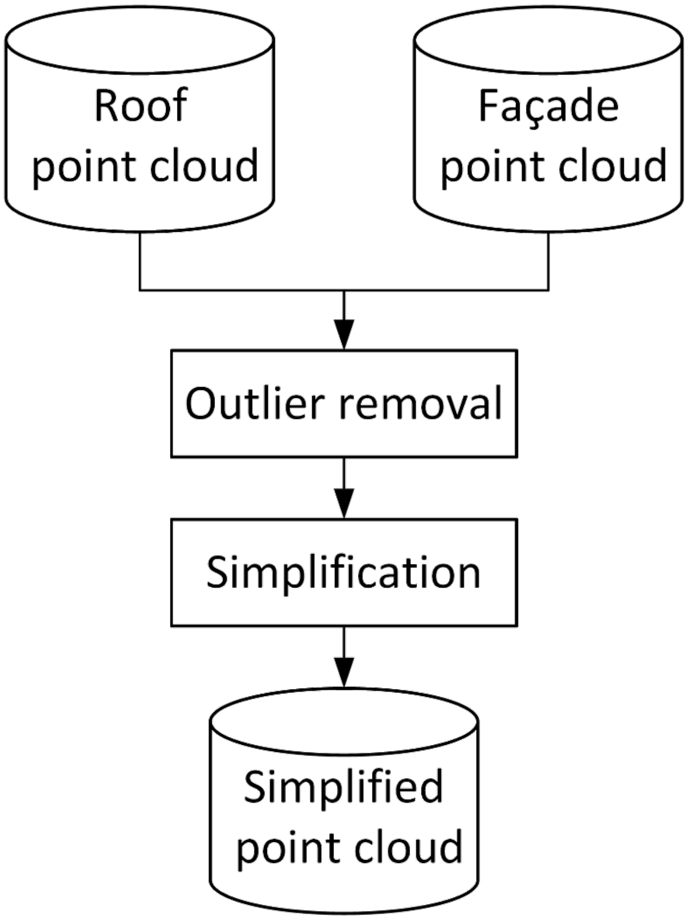


  - Focus only on one or two objectives of Noise smoothing, edges preservation and uniform density
  - Some are slow and inefficient for large-size production
  - No algorithm considers from data source perspective

RESEARCH QUESTION

- *Which algorithms are most suitable to fuse roof and façade point clouds into an edge-aware and uniformly dense color point cloud?*

RESEARCH OVERVIEW

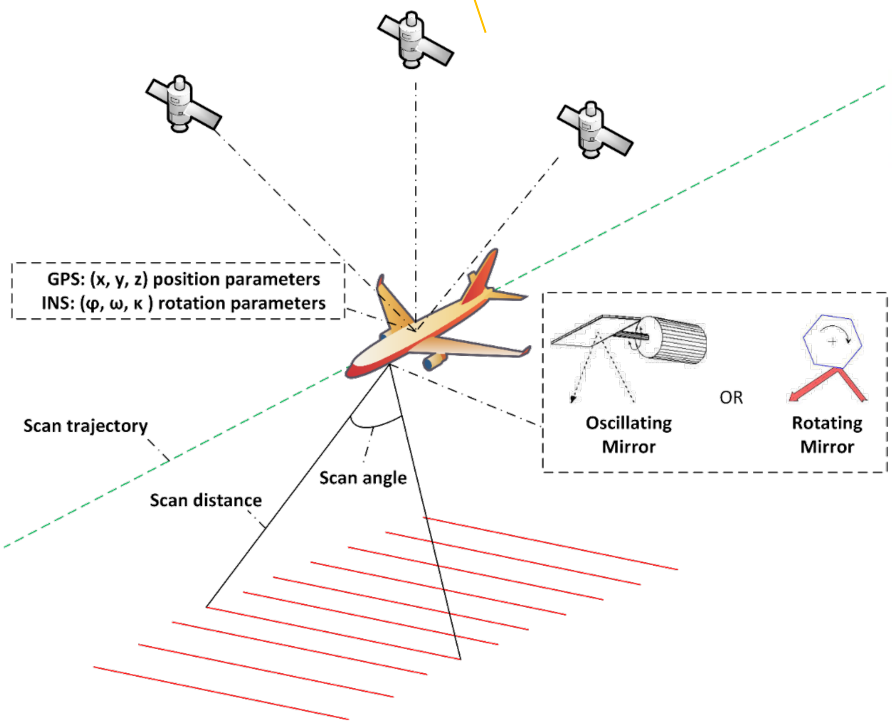


Content

SCANNING METHODS

- Airborne LiDAR, terrestrial LiDAR and panoramic imagery

Line scanner



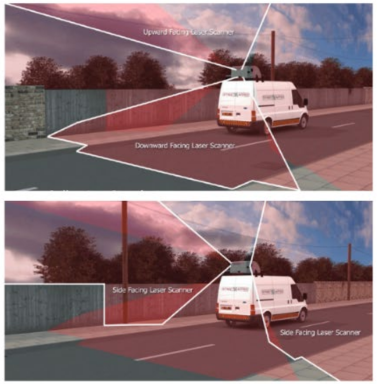
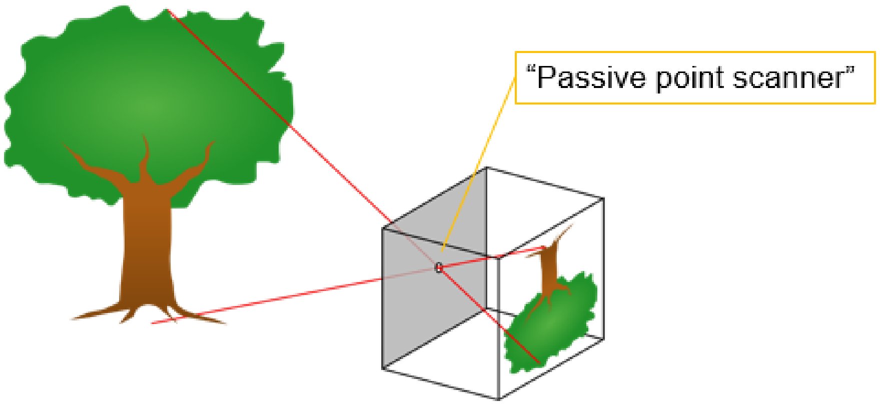


Figure from Haala et al. (2008) and Frohlich et al. (2004)



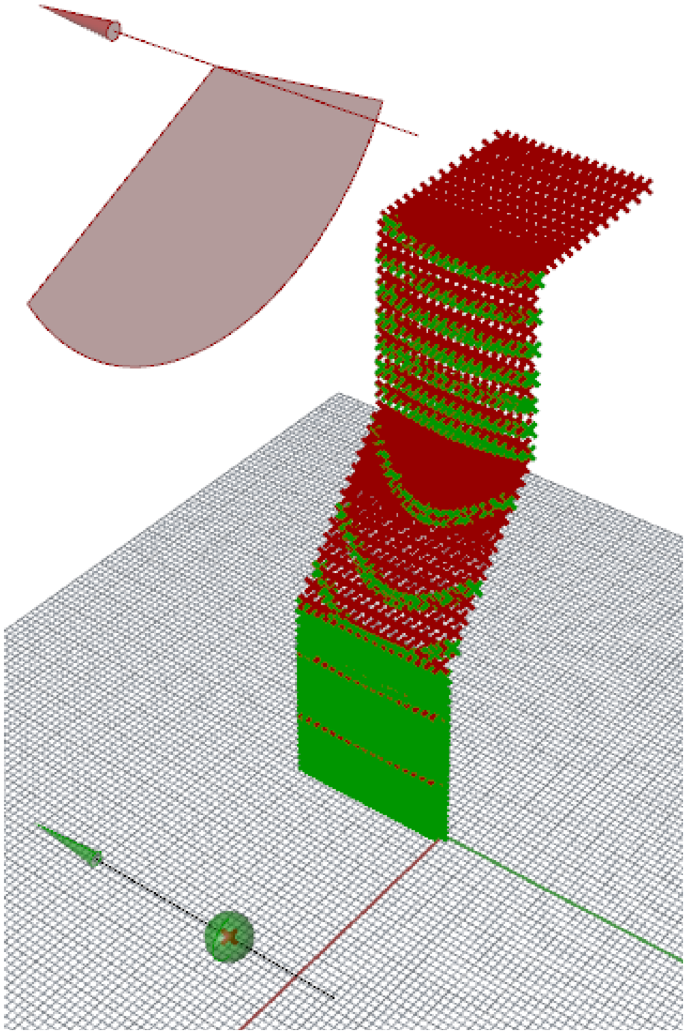
http://en.wikipedia.org/wiki/Camera_lens

ROUGH PRECISION ESTIMATION

- Density weight

- For all point clouds: sparse area
  - Larger distortion
  - Lower resolution

- For all point clouds:
  Density indicates **precision** !



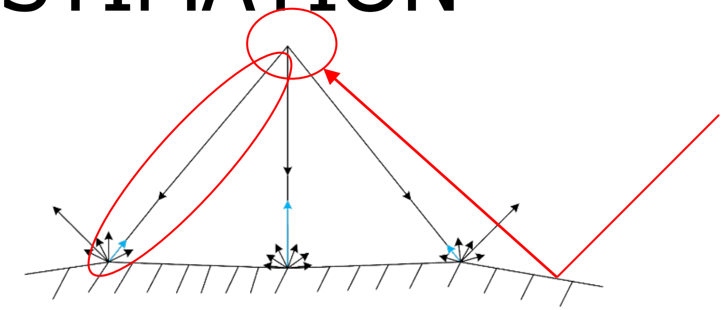
RIGOROUS PRECISION ESTIMATION

- One step further, possible to get mathematical rigorous precision from error propagation theory?

- E.g. $$\sigma_{s_i}^2 = a_1\sigma_1^2 + a_2\sigma_2^2 + ... + a_n\sigma_n^2$$

RIGOROUS PRECISION ESTIMATION

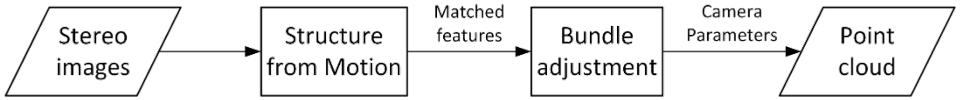
- Error sources : LiDAR



Category	Error source	Quantitative error estimation for each point
Scanner position	GPS position parameters; INS attitude parameters (not for static terrestrial systems)	Difficult to obtain for each scan
Coordinate difference from scanner to point	Incidence angle of the laser ray	Possible to get if recorded for each point in scanning.
Other factors	Laser reflection and refraction errors	In current methods not possible to derive

RIGOROUS PRECISION ESTIMATION

- Error sources : Panoramic imagery



Category	Error source	Quantitative error estimation for each point
Camera position	GPS position parameters; INS rotation parameters	Difficult to obtain for each scan position
Feature points extraction	Object radiometric changes; non-uniform motion blur; non Gaussian-distributed camera structured noise etc.	Currently impossible to estimate precision for all those aspects.
Bundle adjustment	Difference between point positions before and after adjustment.	Possible to derive precision for each point.
Other error sources	Reflection and refraction of material such as glass.	In current methods not possible to derive.

RIGOROUS PRECISION ESTIMATION

- Conclusions

- Not yet possible to derive a mathematical rigorous model …

- Existing method: from Post-processing NOT data origin
  - e.g. point distance to plane by RANSAC fitting

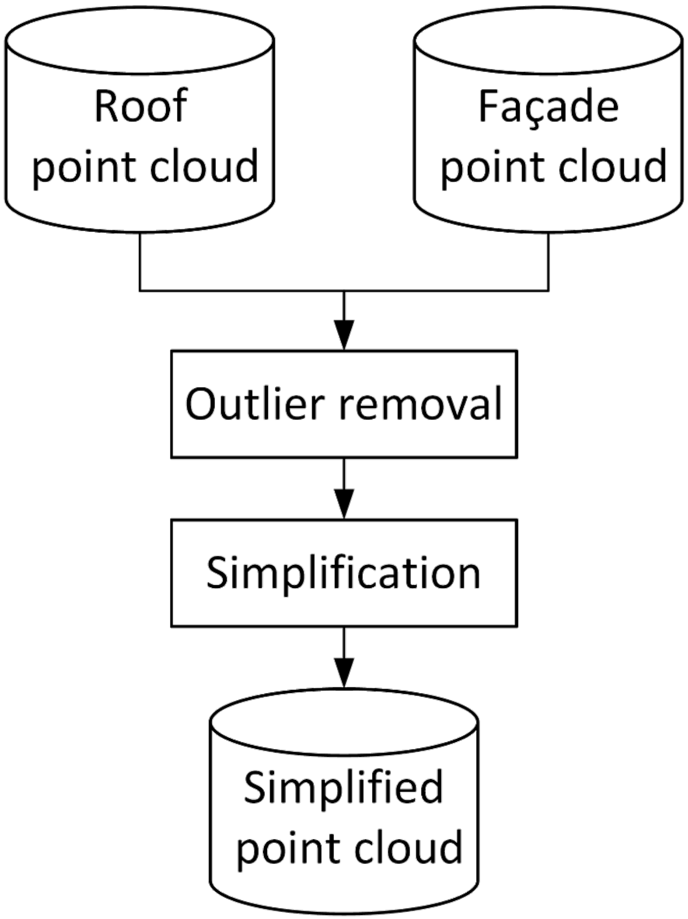
C. v. d. Sande et al. (2010). Assessment of relative accuracy of AHN-2 laser scanning data using planar features

- Leave as an open research topic…

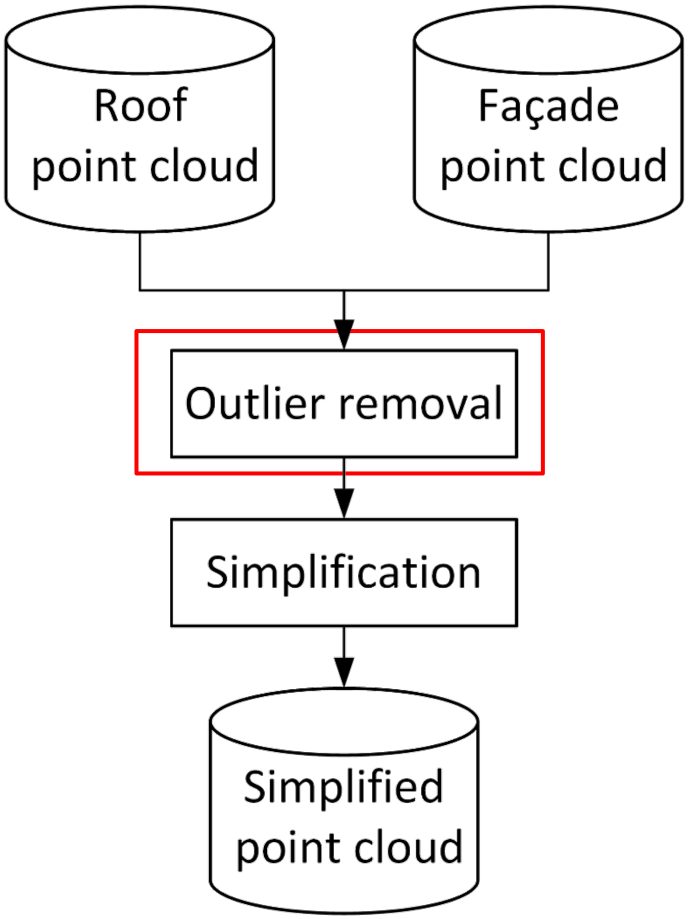
Content

ALGORITHM PIPELINE

- Outlier removal algorithm
- Simplification algorithm
- Integration into one pipeline

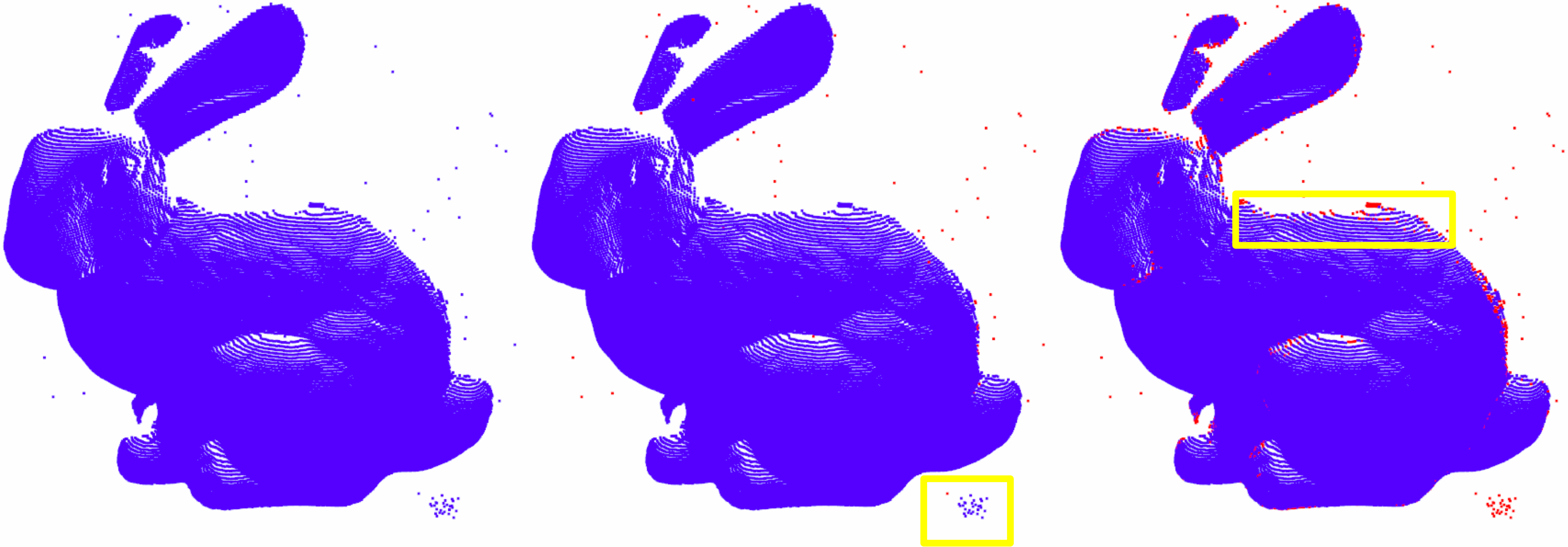


ALGORITHM PIPELINE



OUTLIER REMOVAL

- Artifacts of existing methods



Original LDOF NNR

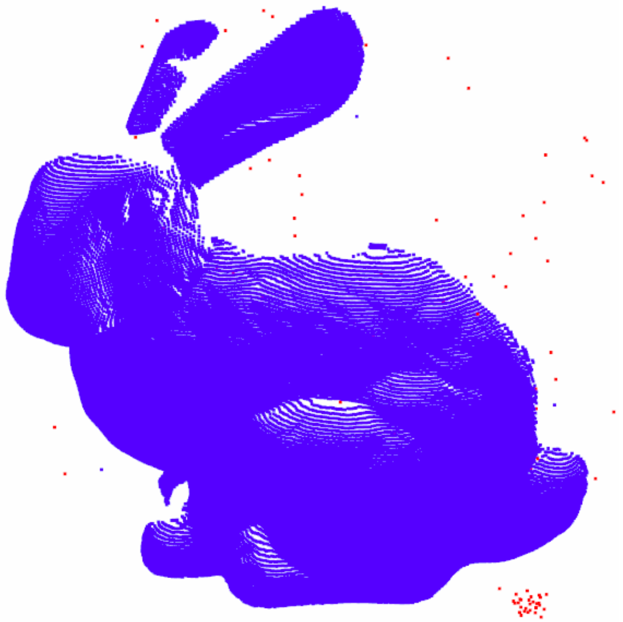
Zhang et al (2009). A new local distance-based outlier detection approach for scattered real-world data
Weyrich et al. (2004). Post-processing of scanned 3d surface data

OUTLIER REMOVAL

- Our method: Improved NNR

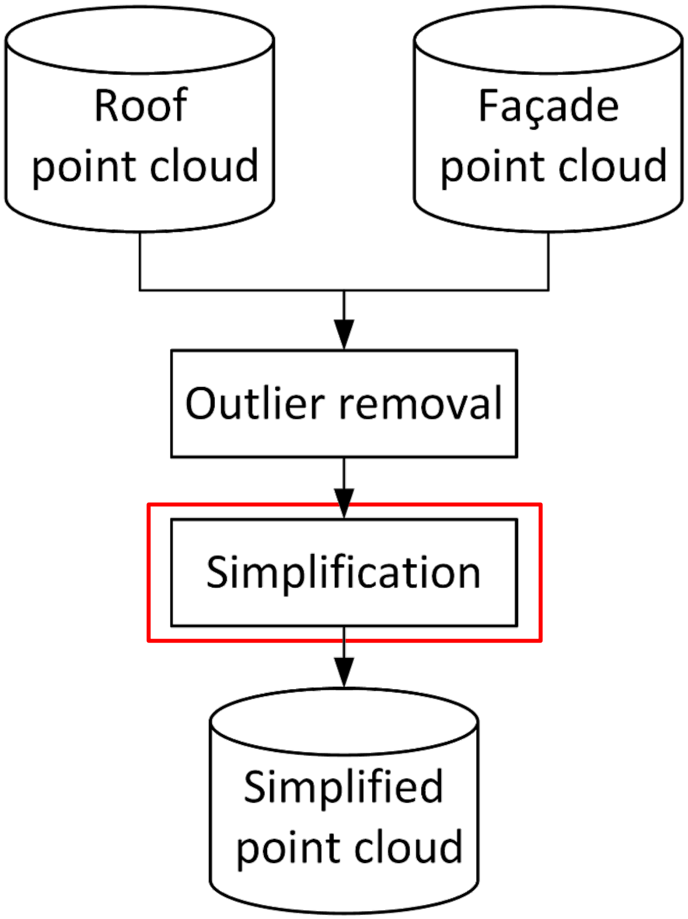
Algorithm 1 Outlier reduction algorithm

procedure OUTLIER REDUCTION(op, c) ▷ outlier percent op; max cluster size c
    calculate cluster-adapted LDOF value for each point
    sort points using LDOF value in descending order
    $outliersize \leftarrow inputsize * op$
    $LDOFoutliers \leftarrow$ the first $outliersize$ items in sorted result
    for $p_i \in P$ **do**
        $percent \leftarrow$ NNR unidirectional neighbor ratio of each point
        if $percent \leq op$ **then**
            if $p_i \notin LDOFoutliers$ **then**
                p_i is inlier
            else
                p_i is outlier



Our method

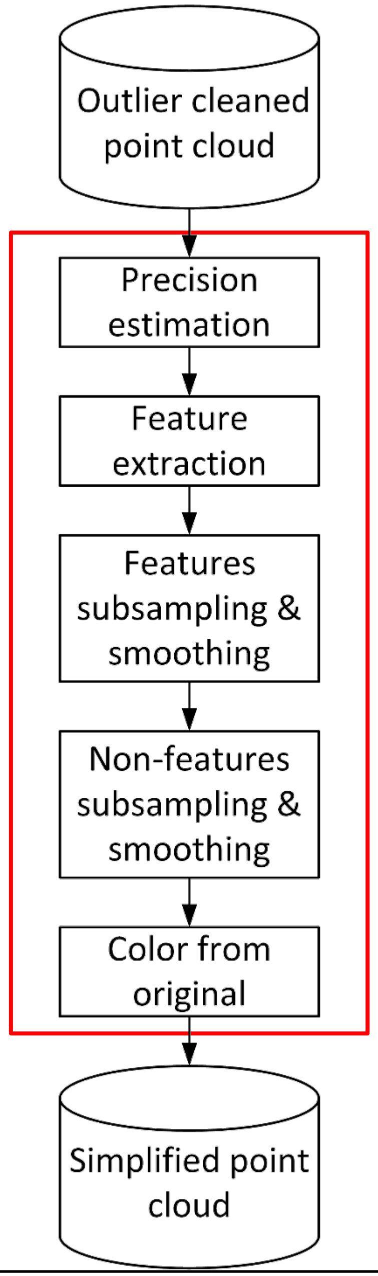
ALGORITHM PIPELINE



SIMPLIFICATION

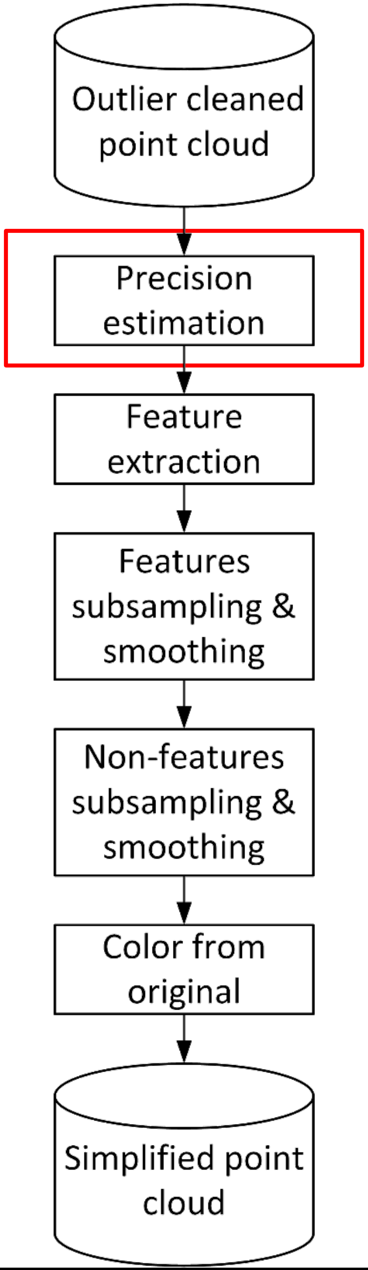
- Pipeline

- Sub-steps



SIMPLIFICATION

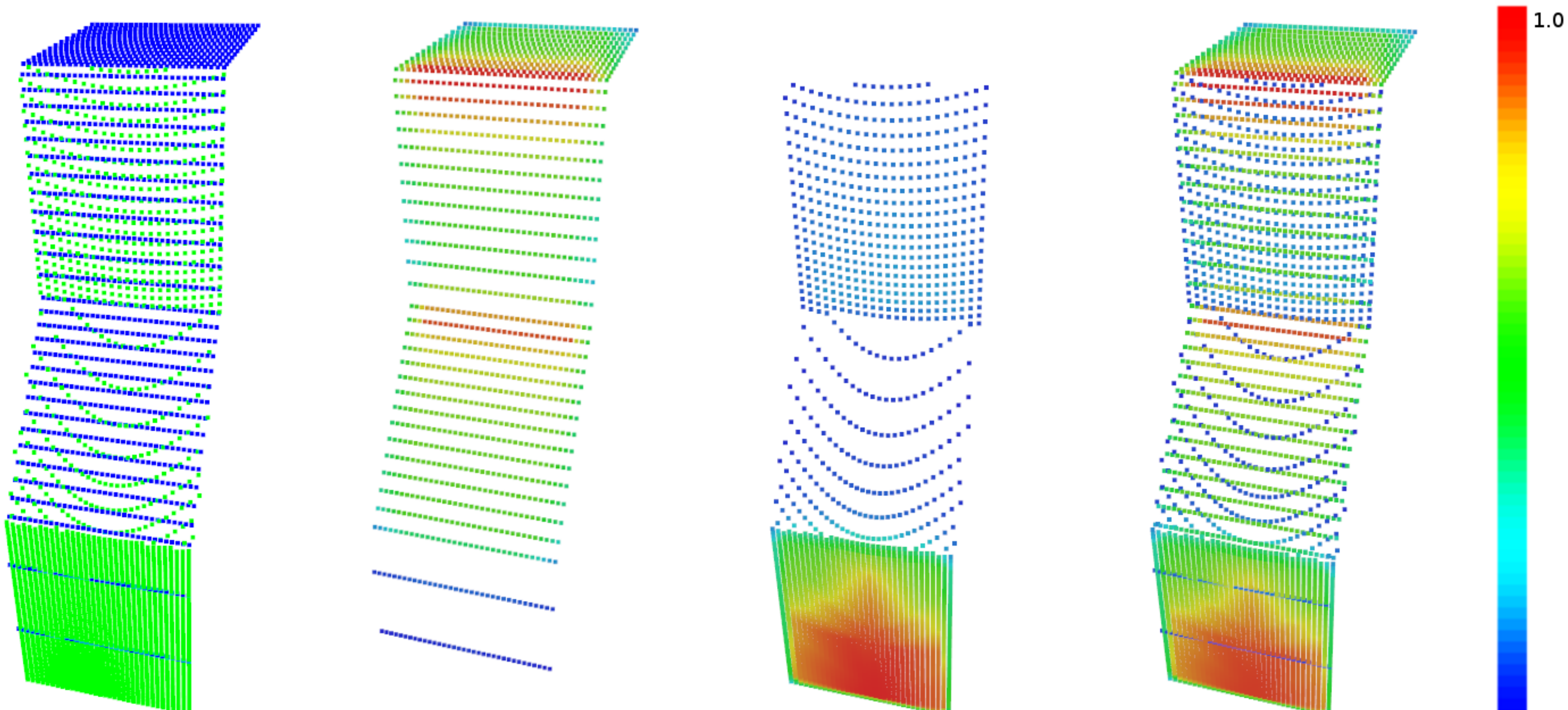
- Pipeline



SIMPLIFICATION

- Precision (Importance) estimation by density

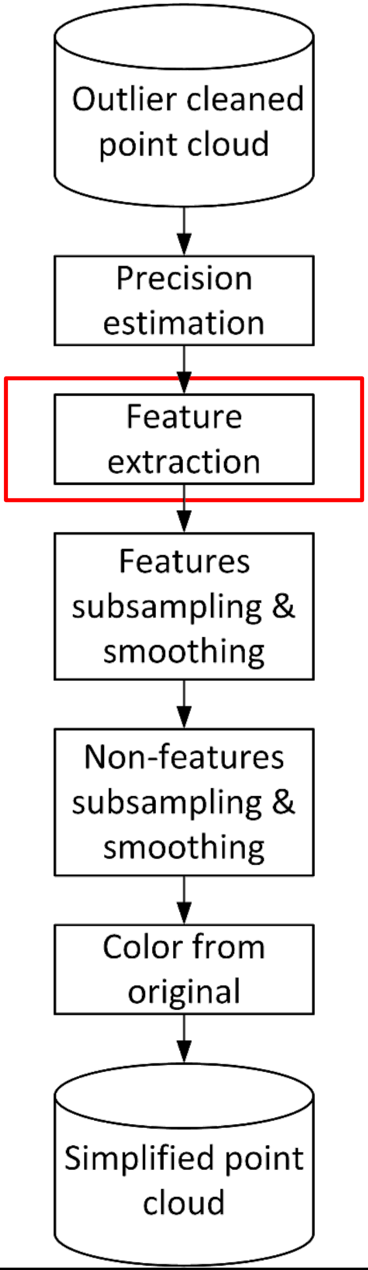
$$\sigma_i = \frac{K}{\pi R_i^2} \qquad Iv_i = e^{-\frac{\overline{\sigma}}{\sigma_i}}$$



(a) (b) (c) (d)

SIMPLIFICATION

- Pipeline



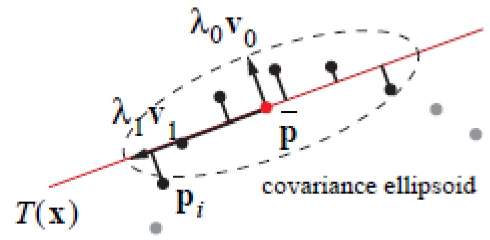
SIMPLIFICATION

- Feature extraction: surface variation OR curvature

- Surface variation (Pauly et al, 2002)

$$\sigma_n(\mathbf{p}) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}$$

where eigen values $0 \leq \lambda_0 \leq \lambda_1 \leq \lambda_2$

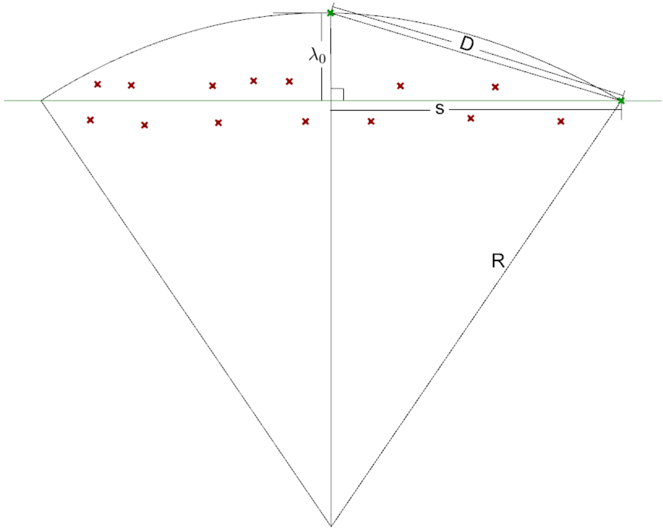


- Curvature (Gumhold et al, 2001)

$$\kappa = \frac{2\lambda_0}{D^2}$$

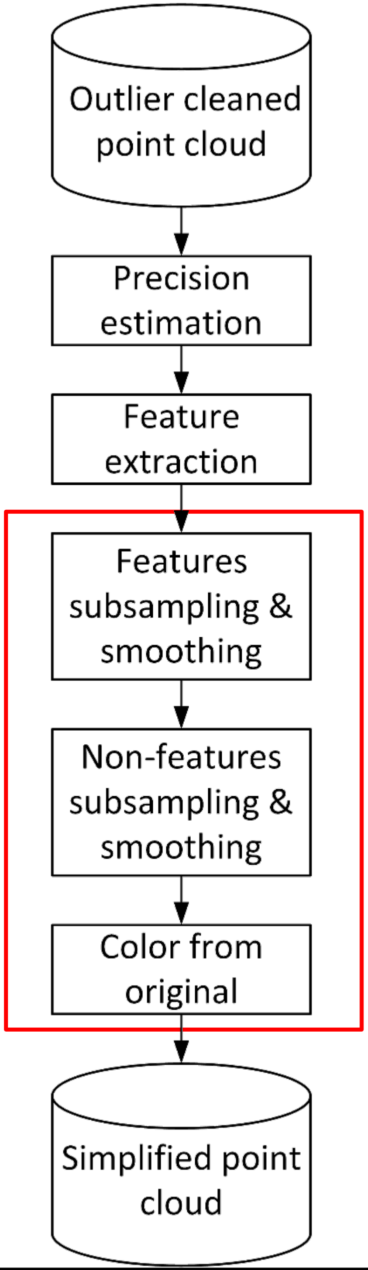
λ_0: smallest eigen value

D: search radius



SIMPLIFICATION

- Pipeline



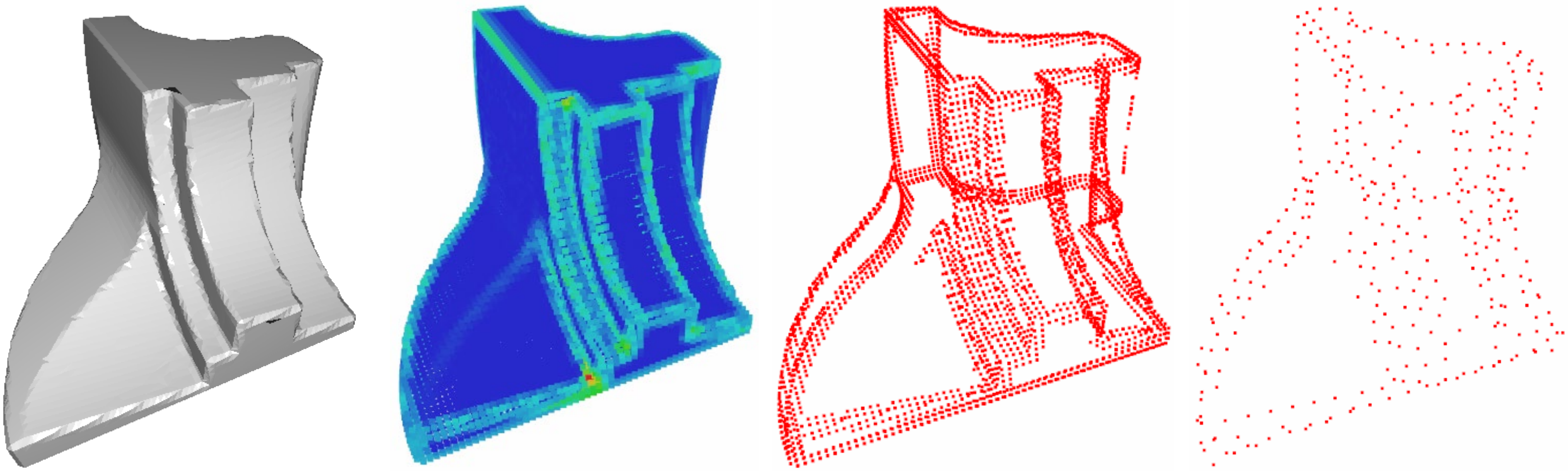
SIMPLIFICATION

- Sub-sampling & smoothing strategy

- Same strategy for features and non-features
- Only difference
  - Feature points first
  - Non-features move with respect to features

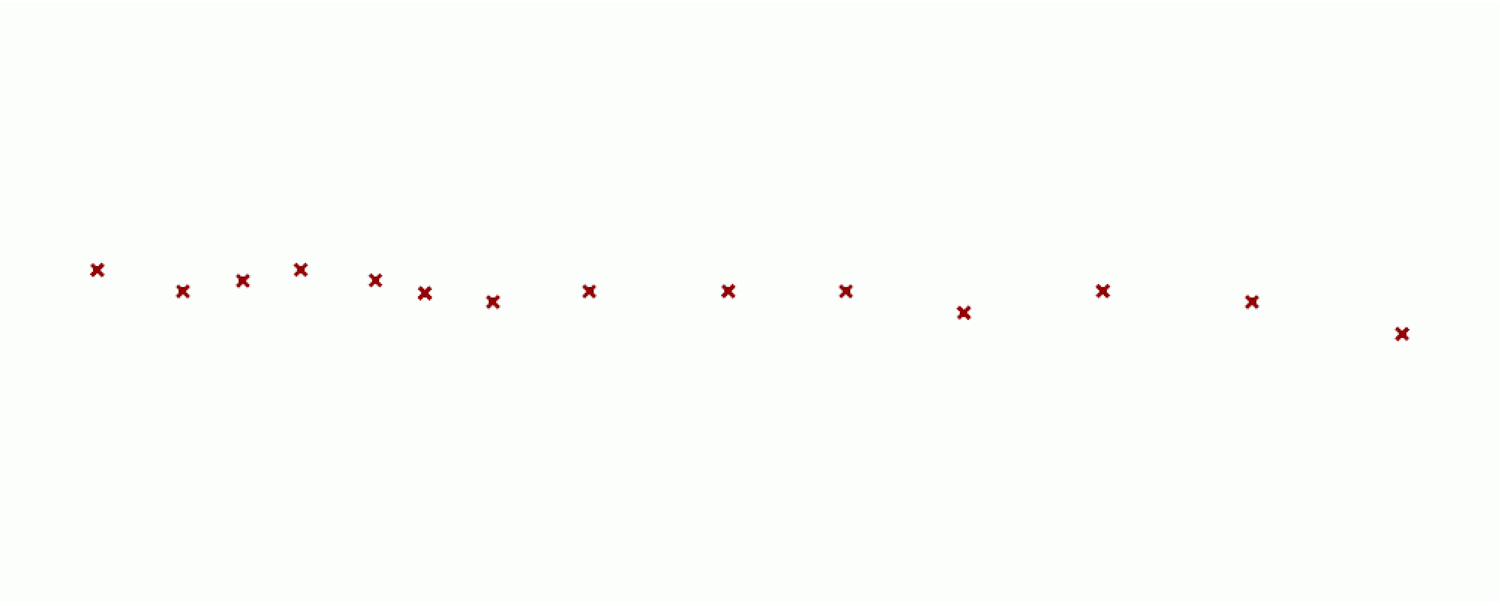
SIMPLIFICATION

- Feature extraction, subsampling and smoothing



SIMPLIFICATION

- Subsampling



SIMPLIFICATION

- Smoothing: WLOP - Noise smoothing & Keep uniform

- Weighted Locally Optimal Projector (WLOP):
  - Original point cloud P_J ; Projected (subsampled) point cloud X_I
  - Apply several iterations. The next iteration of projection Q is to minimize

$$Q = G(Q)$$

$$G(C) = argmin_{X=\{x_i\}i\in I}\{E_1(X, P, C) + E_2(X, C)\}$$

Lipman et al. (2007) Parameterization-free Projection for Geometry Reconstruction
Huang et al. (2009). Consolidation of Unorganized Point Clouds for Surface Reconstruction

SIMPLIFICATION

$$Q = G(Q)$$

$$G(C) = argmin_{X=\{x_i\}i\in I}\{E_1(X,P,C) + E_2(X,C)\}$$

- Smoothing: WLOP - Noise smoothing & Keep uniform

$$E_1(X,P,C) = \sum_{i\in I}\sum_{j\in J} ||x_i - p_j||\theta(||c_i - p_j||)$$

Multivariate median (l_1 median) – noise smoothing
- Leads to projection points moving toward the local distribution center

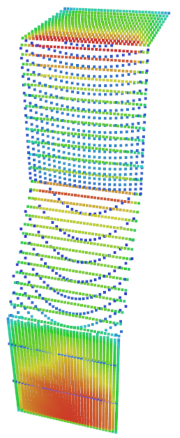
$$E_2(X,C) = \sum_{i'\in I}\lambda_{i'}\sum_{i\in I\setminus\{i'\}} \eta(||x_{i'} - c_i||)\theta(||c_{i'} - c_i||)$$

Repulsion term – Keep uniform
- Penalizing projection points that get too close to each other

SIMPLIFICATION

- Smoothing: WLOP - Noise smoothing & Keep uniform

- Combine original point Importance value Iv_j into the term



$$G(C) = argmin_{X=\{x_i\}i\in I}\{E_1(X,P,C) + E_2(X,C)\}$$

$$E_1(X,P,C) = \sum_{i\in I}\sum_{j\in J} ||x_i - p_j||\theta(||c_i - p_j||)\boxed{Iv_{p_j}}$$

$$E_2(X,C) = \sum_{i'\in I}\lambda_{i'}\sum_{i\in I\backslash\{i'\}}\eta(||x_{i'} - c_i||)\theta(||c_{i'} - c_i||)$$

SIMPLIFICATION

- Smoothing: WLOP - Noise smoothing & Keep uniform

- Final weight density adapted WLOP:

$$x_i^{k+1} = \sum_{j \in J} p_j \frac{\alpha_{ij}^k / v_j}{\sum_{j \in J} (\alpha_{ij}^k / v_j)} + \mu \sum_{i' \in I \; \{i\}} \delta_{ii'}^k \frac{w_{i'}^k \beta_{ii'}^k}{\sum_{i' \in I \; \{i\}} (w_{i'}^k \beta_{ii'}^k)}$$

where:

$$\theta(r) = e^{-\frac{r^2}{R^2}}$$

$$\delta_{ii'}^k = x_i^k - x_{i'}^k$$

$$v_j = 1 + \sum_{j' \in J \; \{j\}} \theta\left(\left\| p_j - p_{j'} \right\|\right) \boxed{I v_{j'}}$$

$$w_i^k = 1 + \sum_{i' \in I \; \{i\}} \theta(\| x_i^k - x_{i'}^k \|)$$

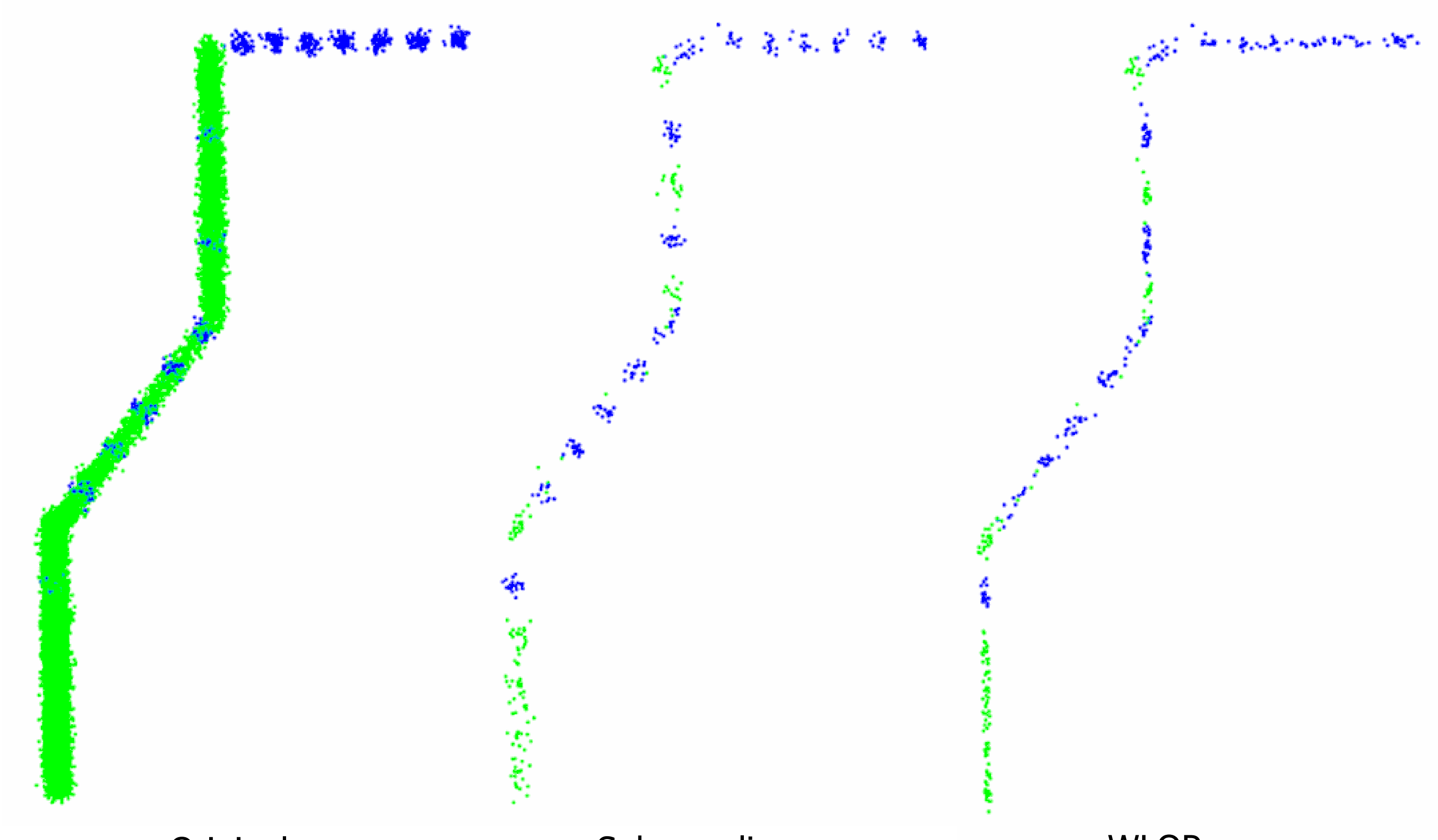
$$\alpha_{ij}^k = \frac{\theta\left(\left\| x_i^k - p_j \right\|\right) \boxed{I v_j}}{\left\| x_i^k - p_j \right\|}$$

$$\beta_{ii'}^k = \frac{\theta\left(\left\| x_i^k - x_{i'}^k \right\|\right)}{\| x_i^k - x_{i'}^k \|}$$

$$\mu = 0.45$$

SIMPLIFICATION

- Smoothing: WLOP - Noise smoothing & Keep uniform



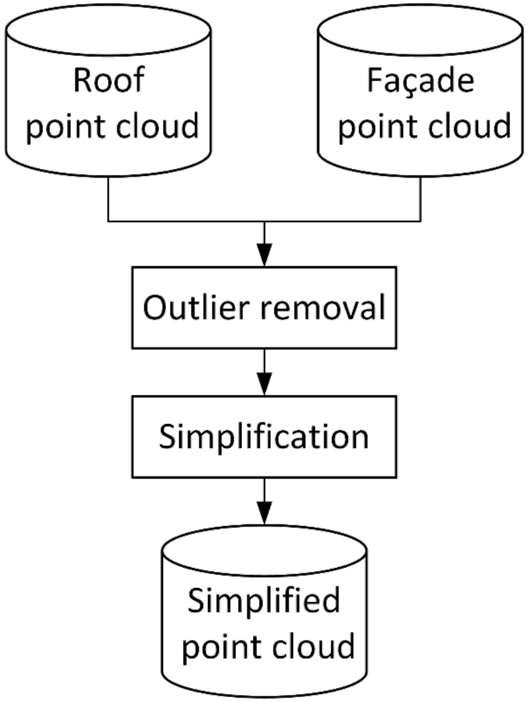
Original
std=0.0092

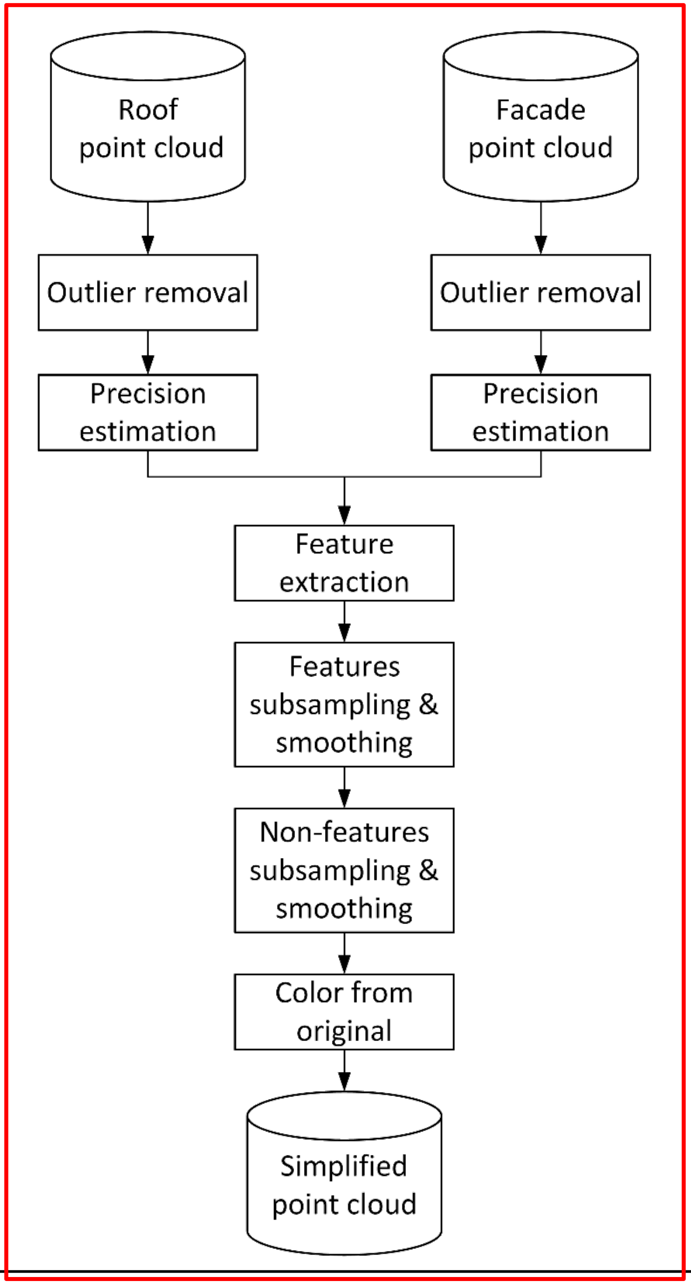
Subsampling
std=0.0006

WLOP
std=0.0006

INTEGRATION

- Outlier – Precision - Combine – Simplify





Content

Content

IMPLEMENTATION

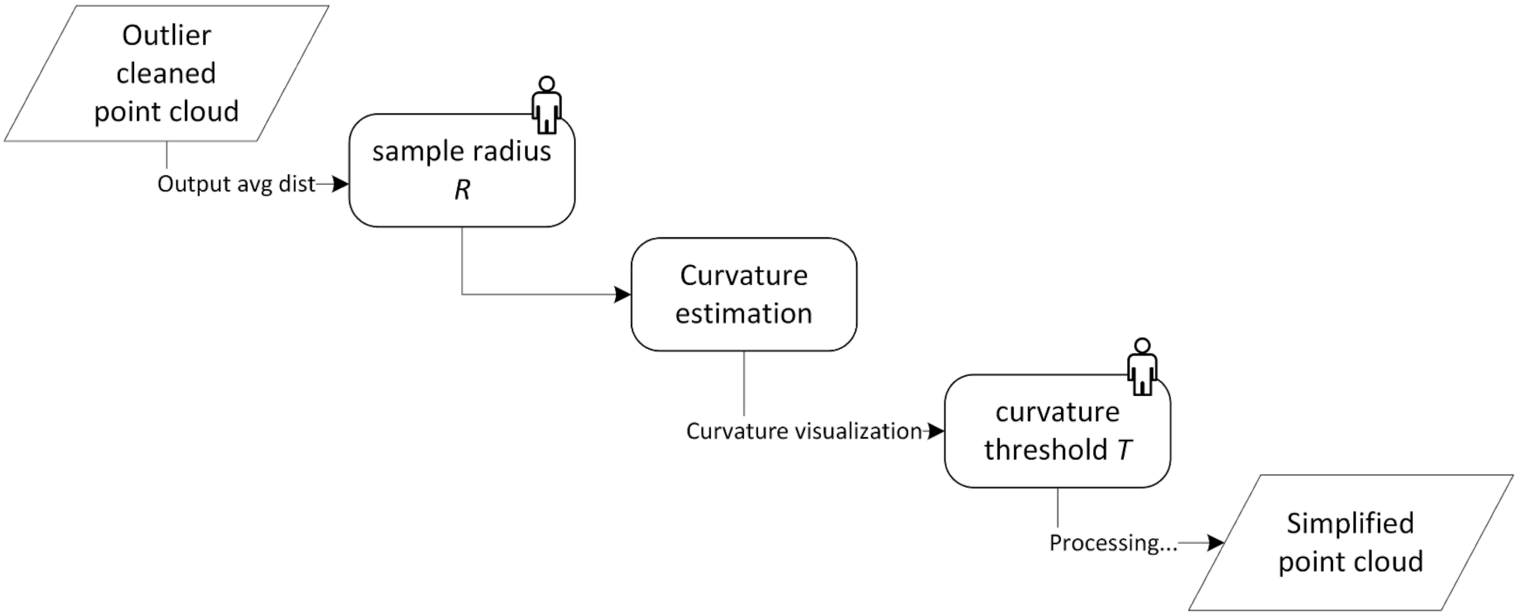
- Data & Tools

- Data
  - Simulated point cloud
  - Roof point cloud: AHN2 *AND* Aerial images (luchtfoto)
  - Façade point cloud: Cyclomedia imagery *OR* Fugro terrestrial LiDAR
- Tools
  - *Simulation* : Rhino / Grasshopper; Blender
  - *Development* : C++; Point Cloud Library (PCL); Qt

IMPLEMENTATION

- Parameters tweaking in simplification

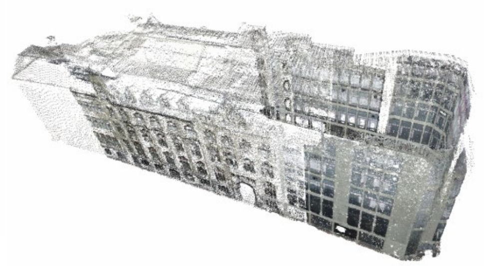
Para	Explanation
R	Sample radius. Reasonably larger than the input average distance
T	Curvature threshold. Points larger than this value are edge points



Content

RESULTS

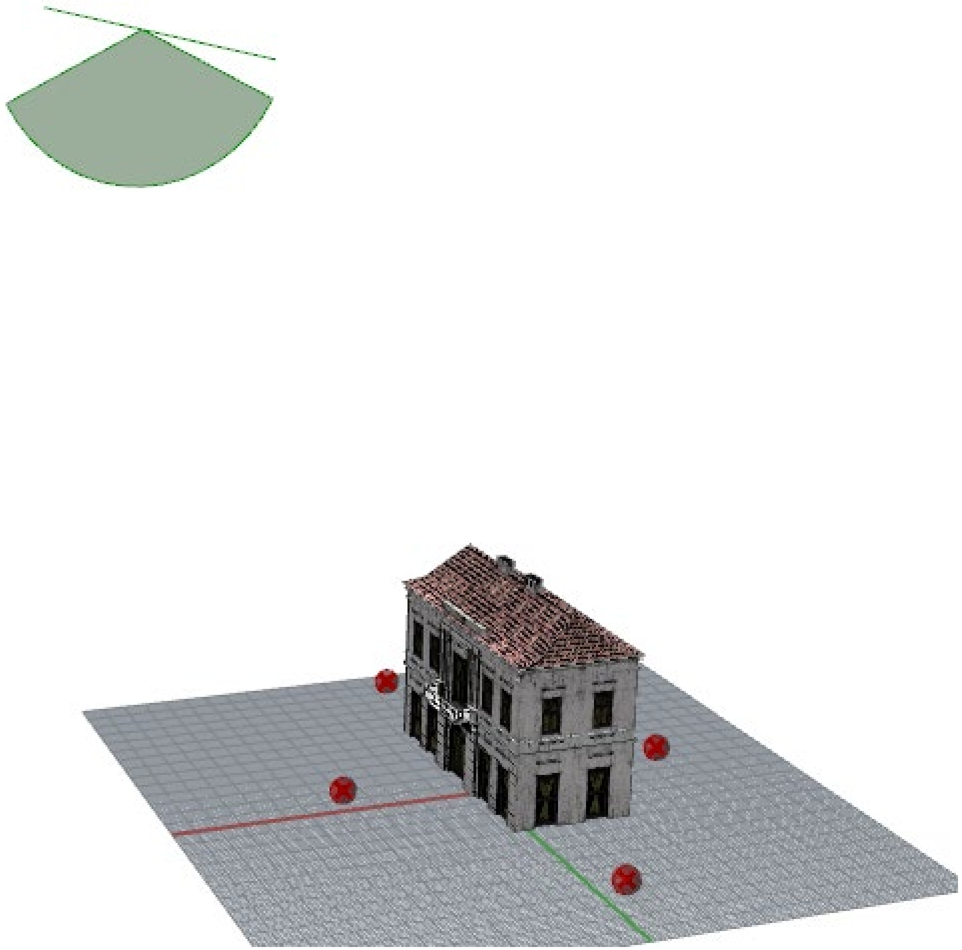
- 4 demos



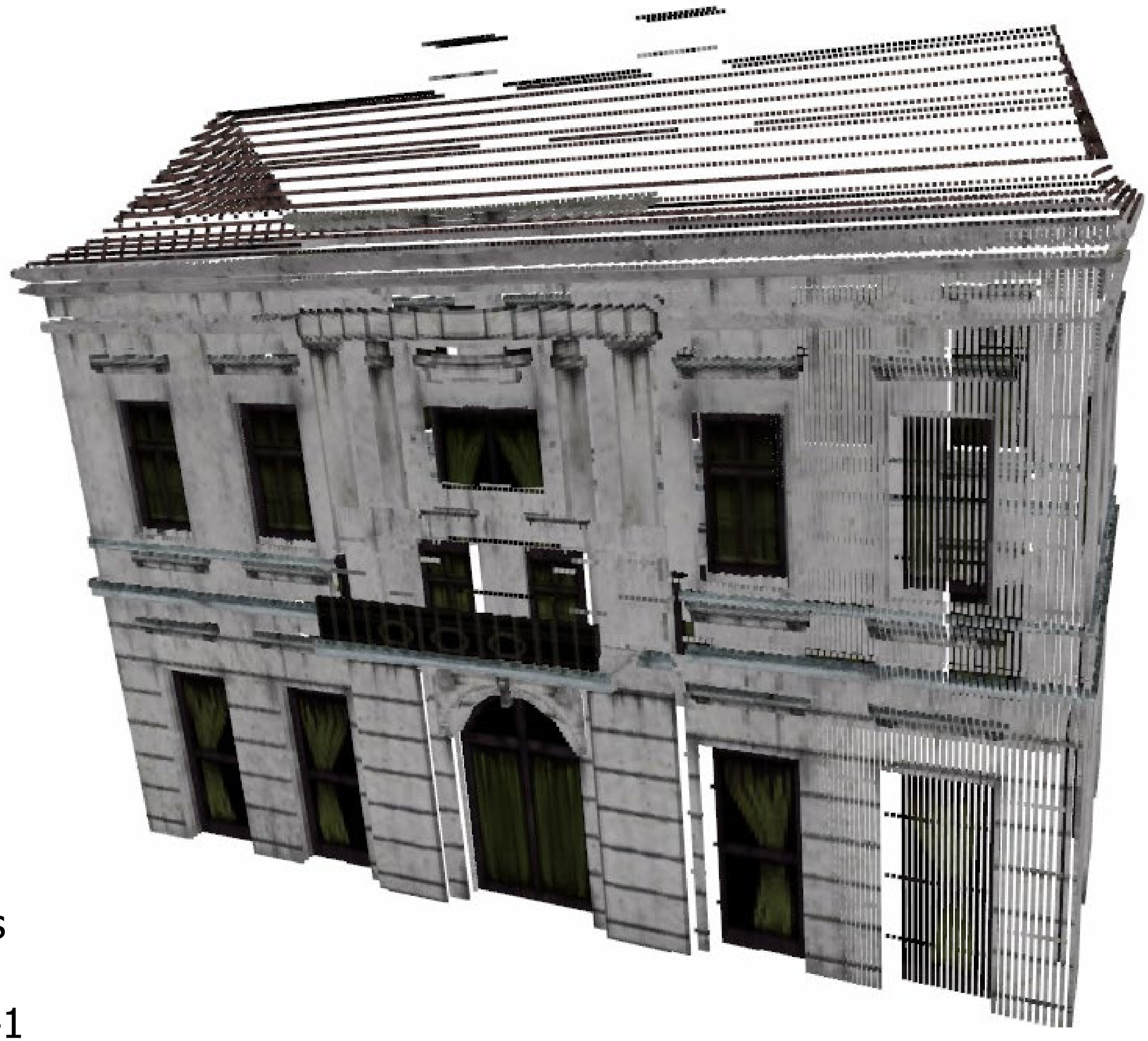
- House. Simple virtual scene
- Villa. Complicate virtual scene
- OTB building. Terrestrial LiDAR + AHN2
- Amsterdam building. Panoramic imagery point cloud + AHN2

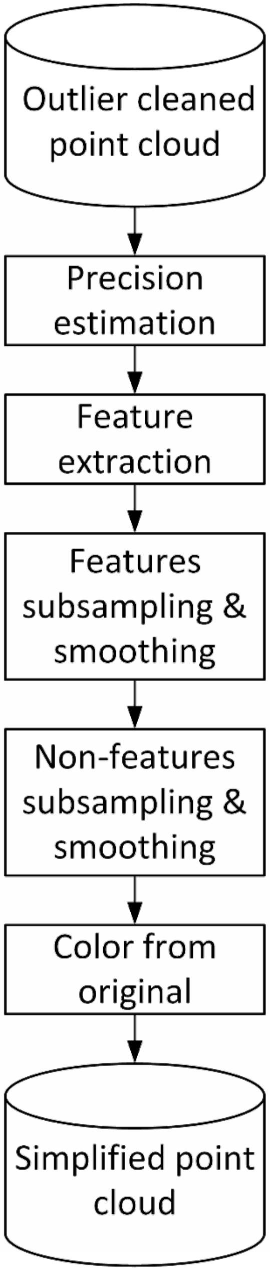
DEMO 1: HOUSE (VIRTUAL)

- Scanning simulation



DEMO 1: HOUSE (VIRTUAL)

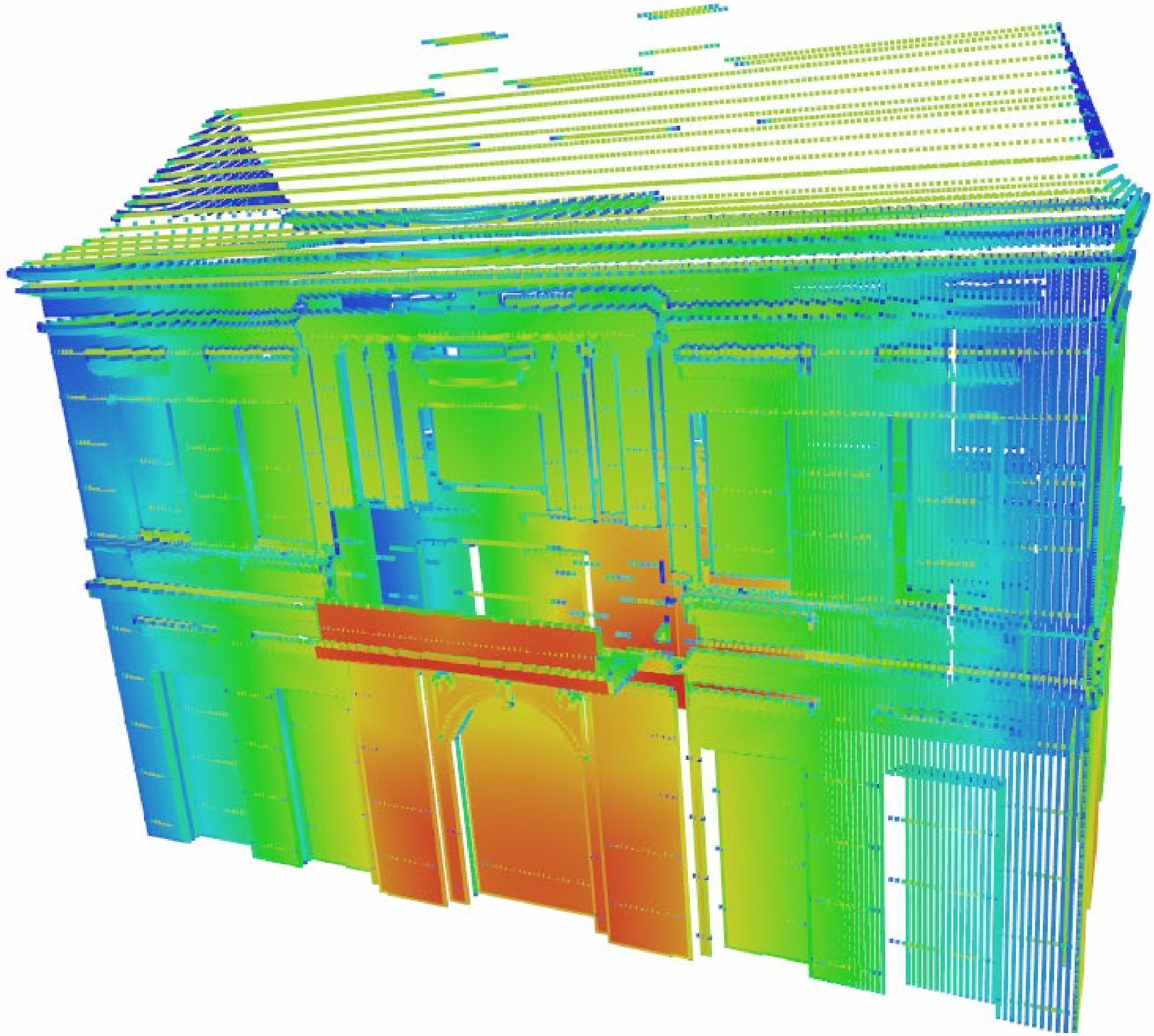


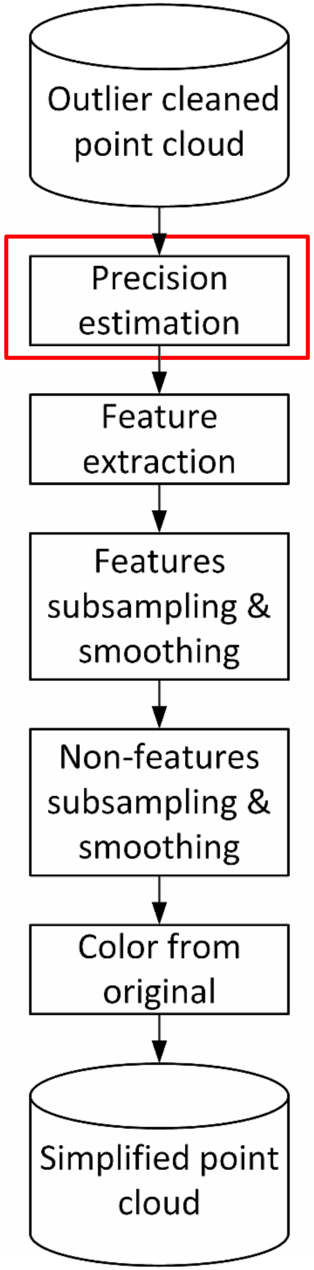


252890 pts

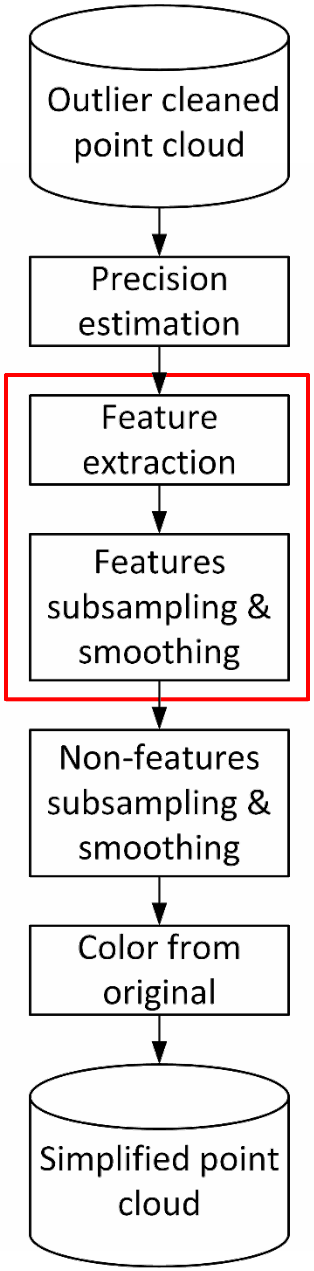
Std = 0.0341

DEMO 1: HOUSE (VIRTUAL)

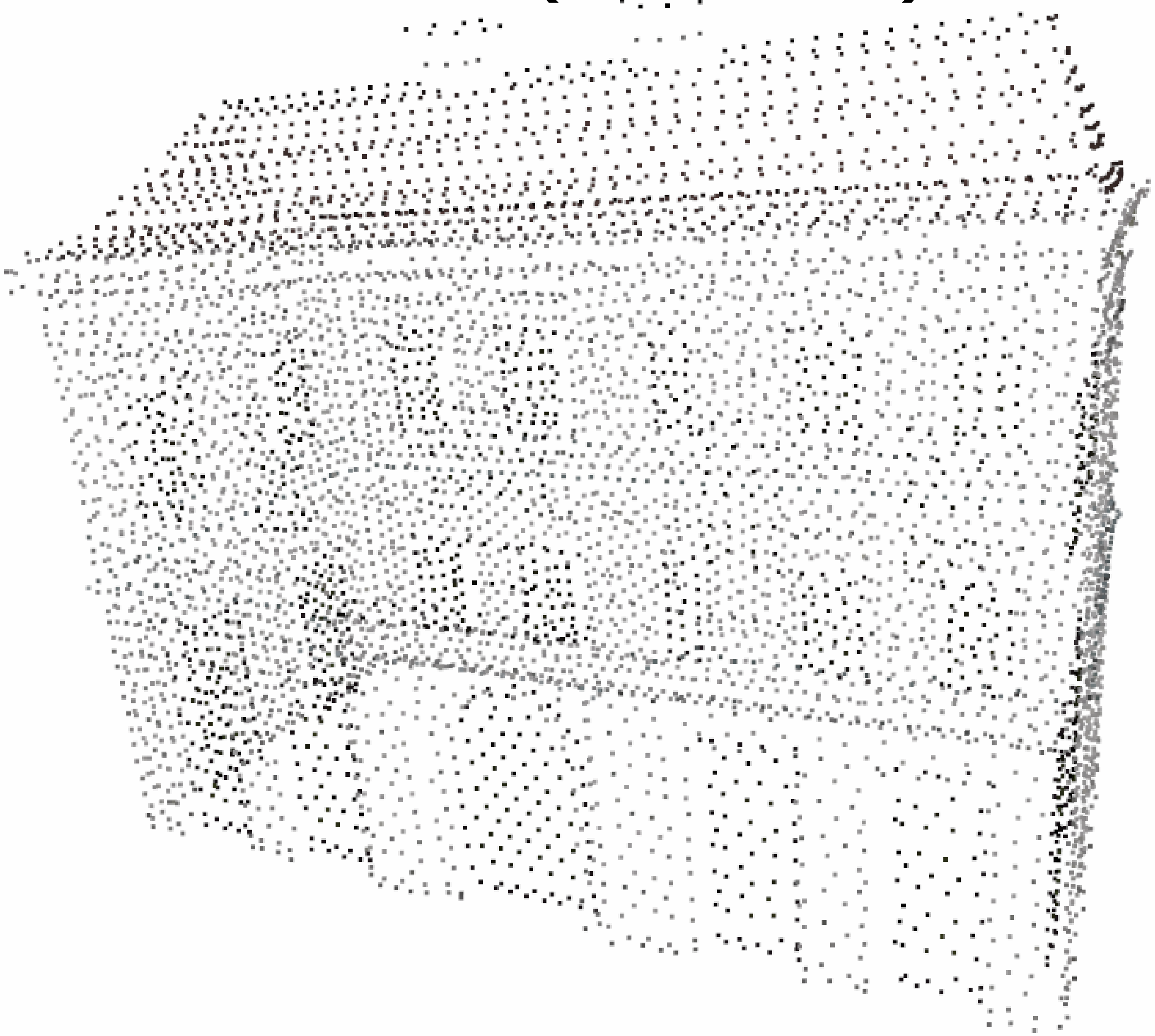




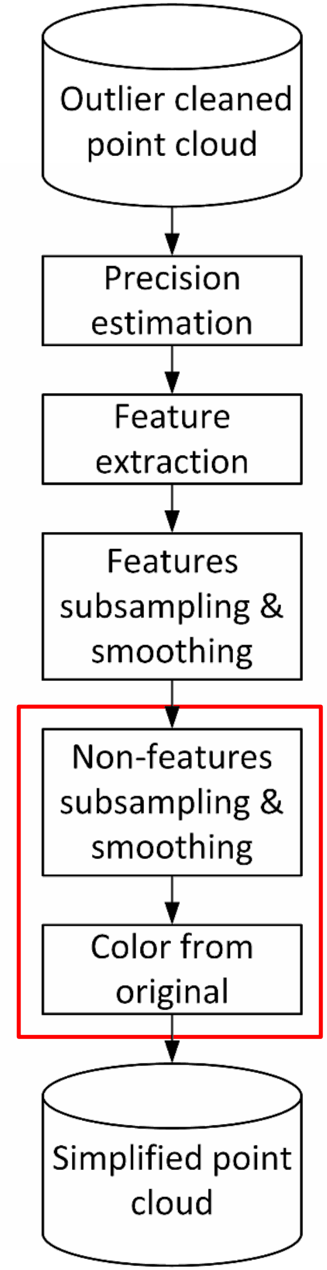
DEMO 1: HOUSE (VIRTUAL)



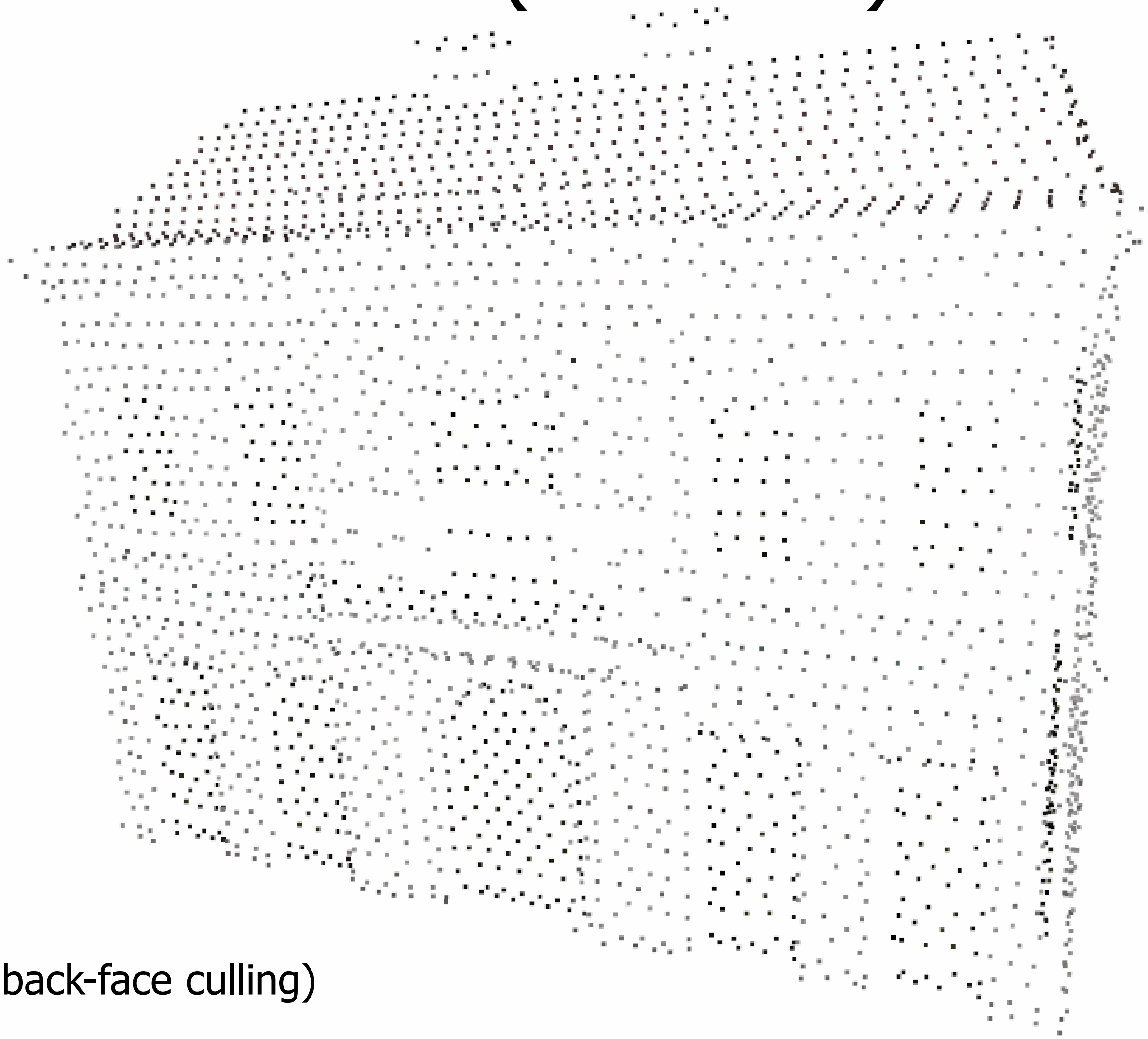
DEMO 1: HOUSE (VIRTUAL)



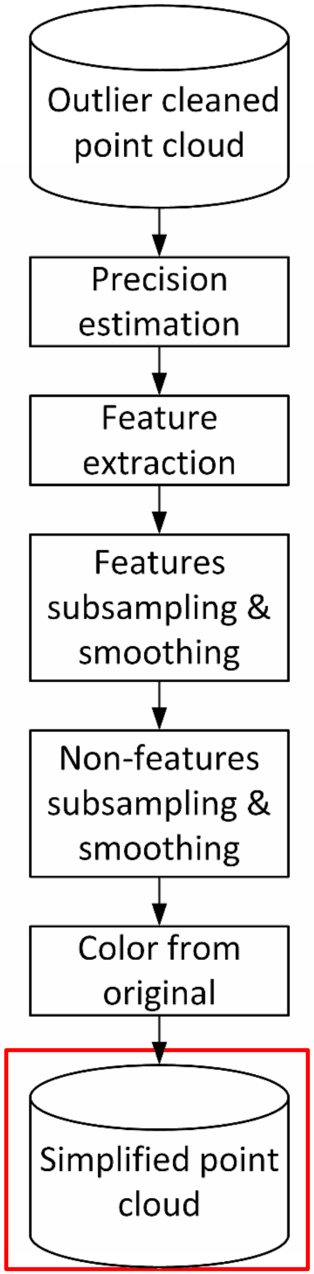
Result



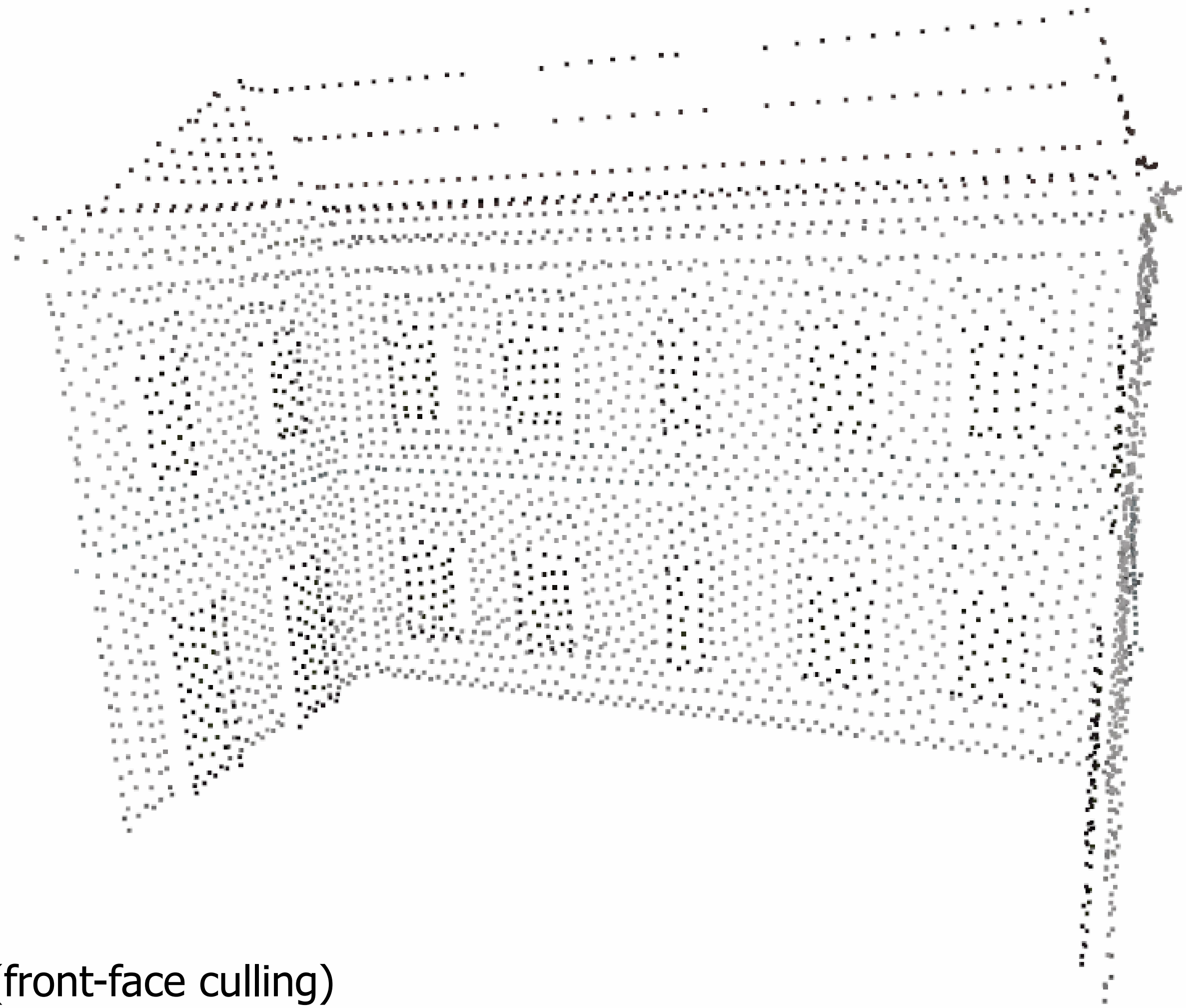
DEMO 1: HOUSE (VIRTUAL)



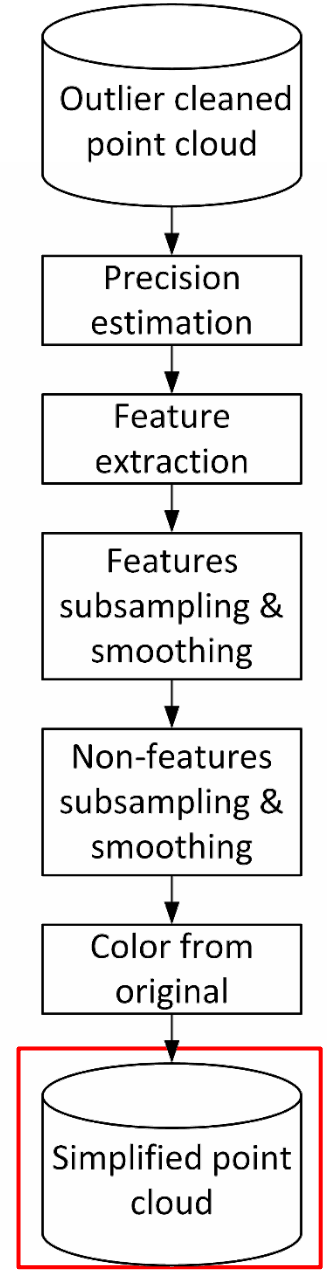
Result (back-face culling)



DEMO 1: HOUSE (VIRTUAL)

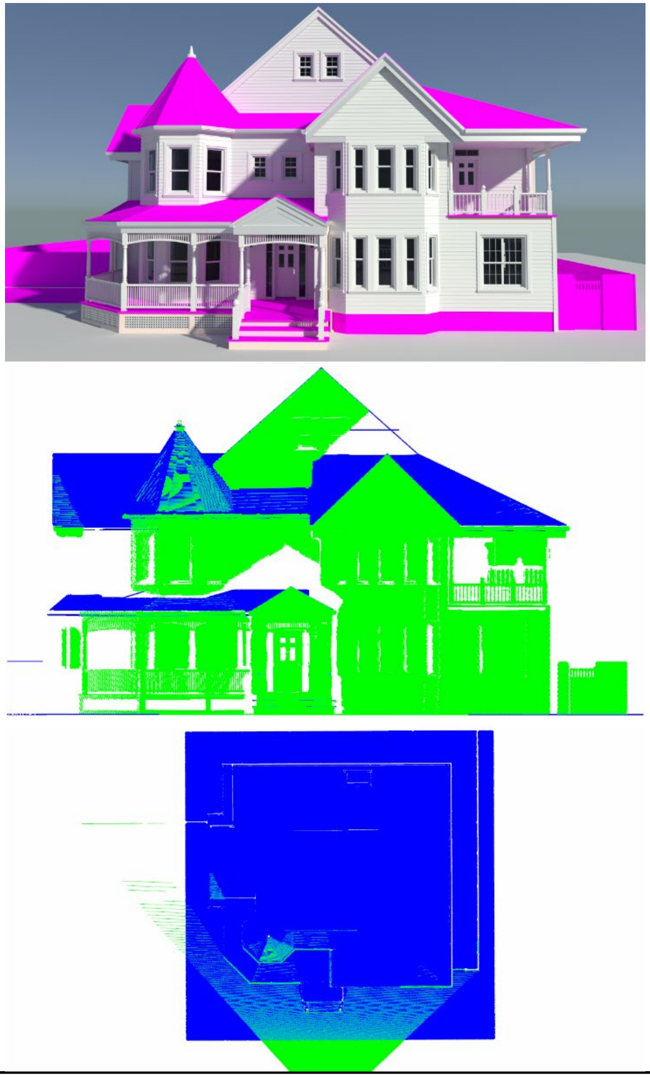


Result (front-face culling)



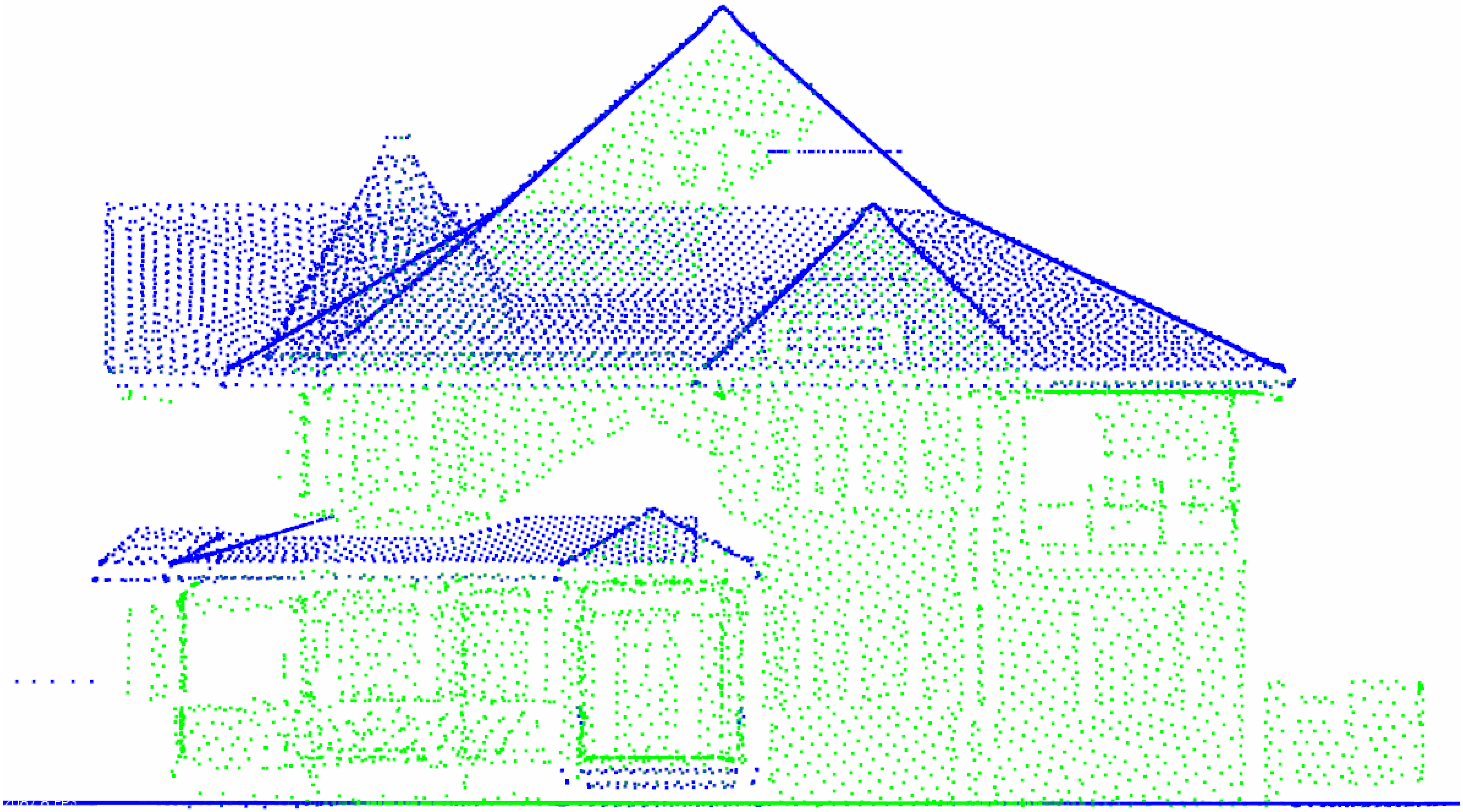
DEMO 2: VILLA (VIRTUAL)

- Simulation



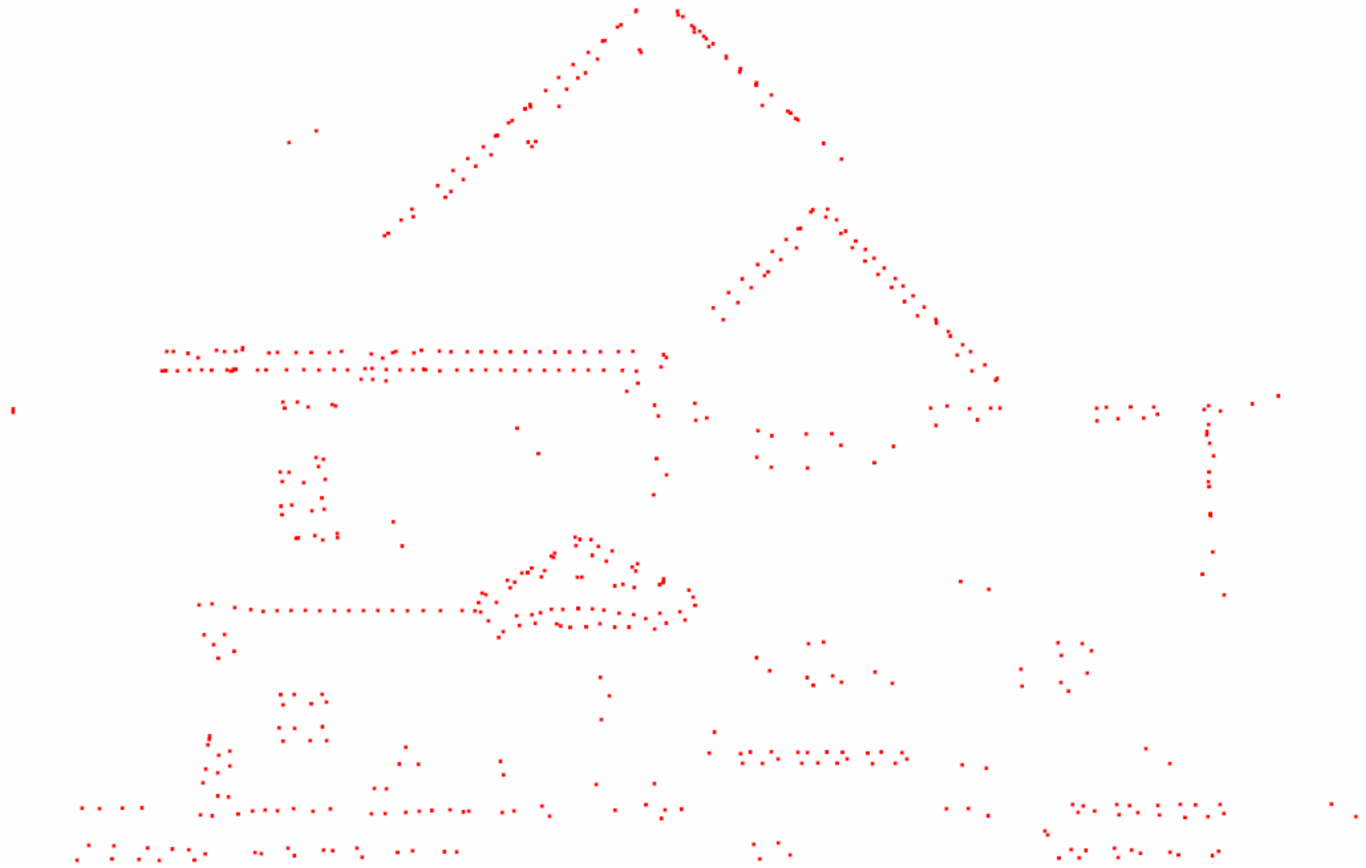
DEMO 2: VILLA (VIRTUAL)

- Result



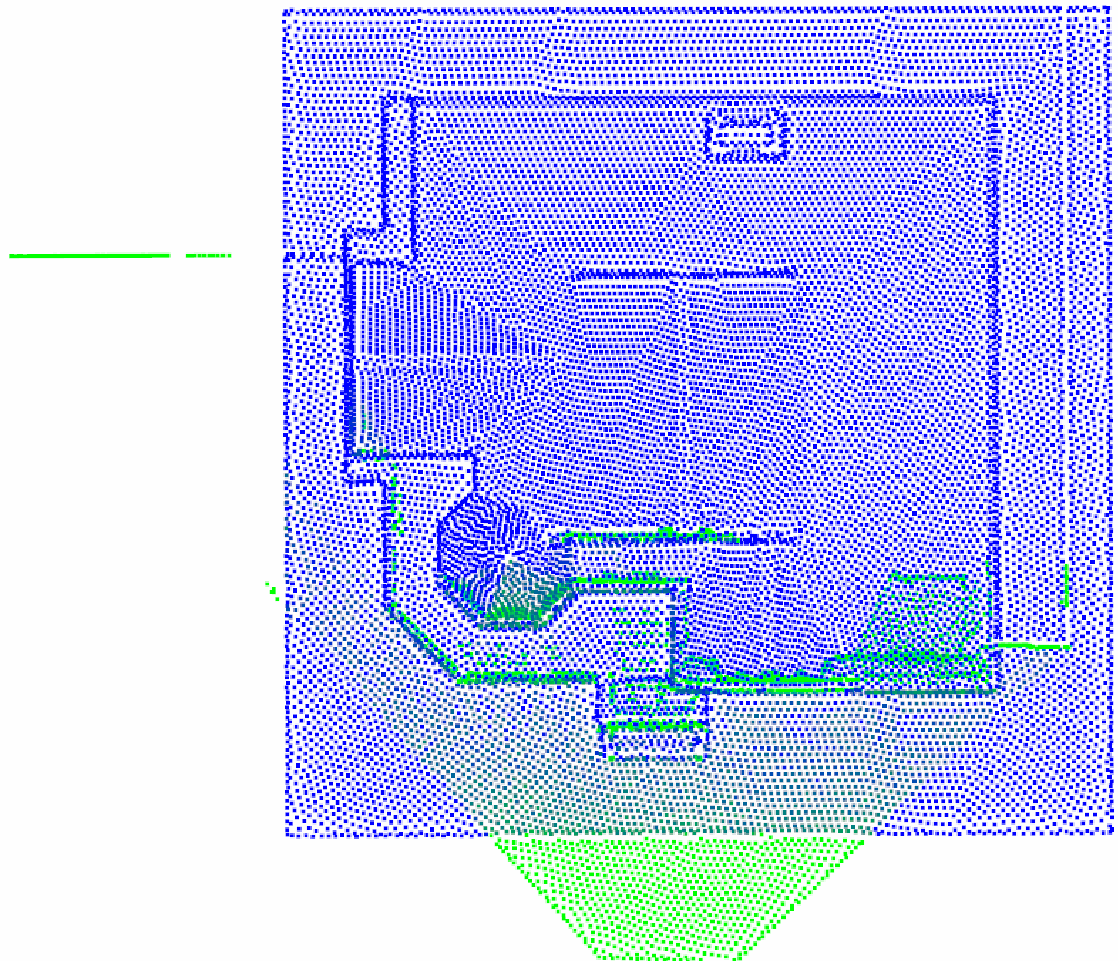
DEMO 2: VILLA (VIRTUAL)

- Result



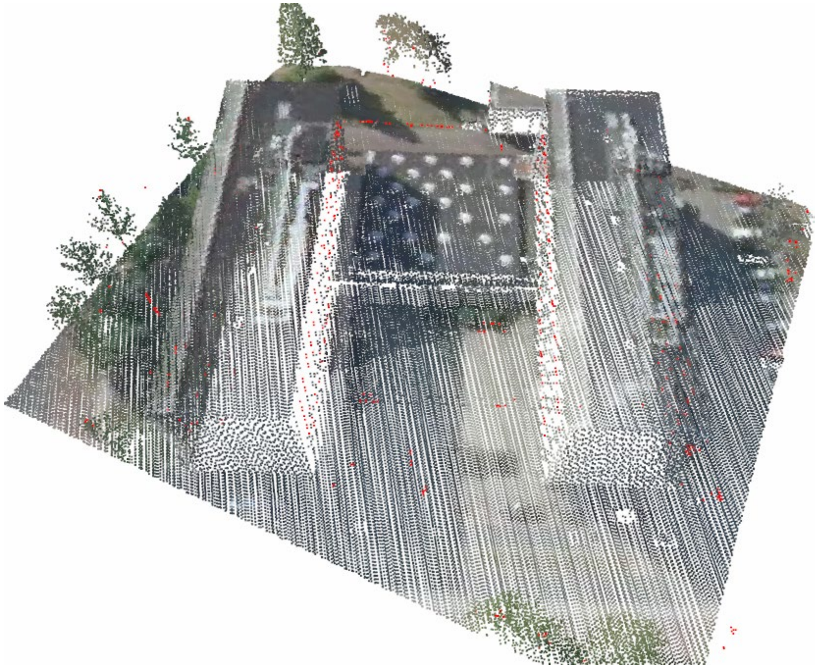
DEMO 2: VILLA (VIRTUAL)

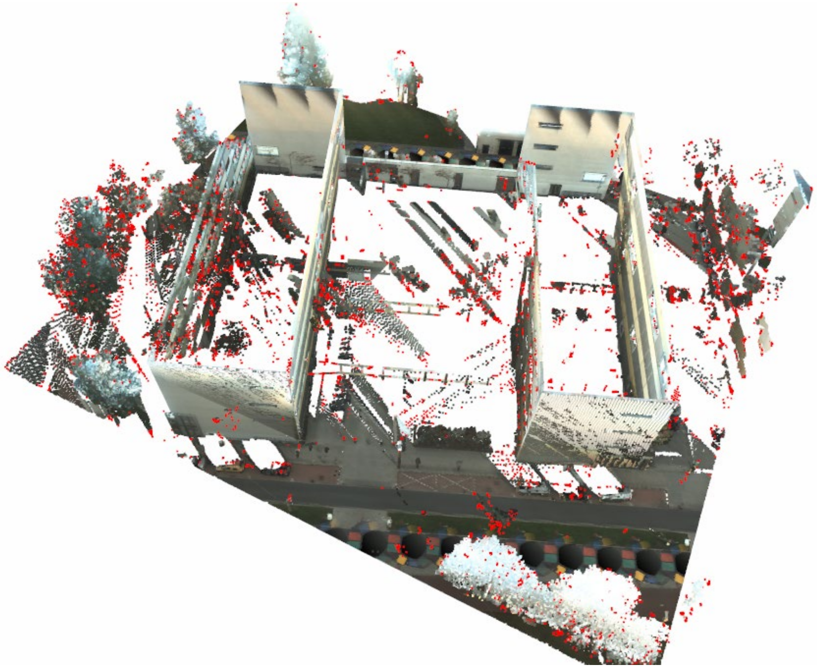
- Result



DEMO 3: OTB BUILDING

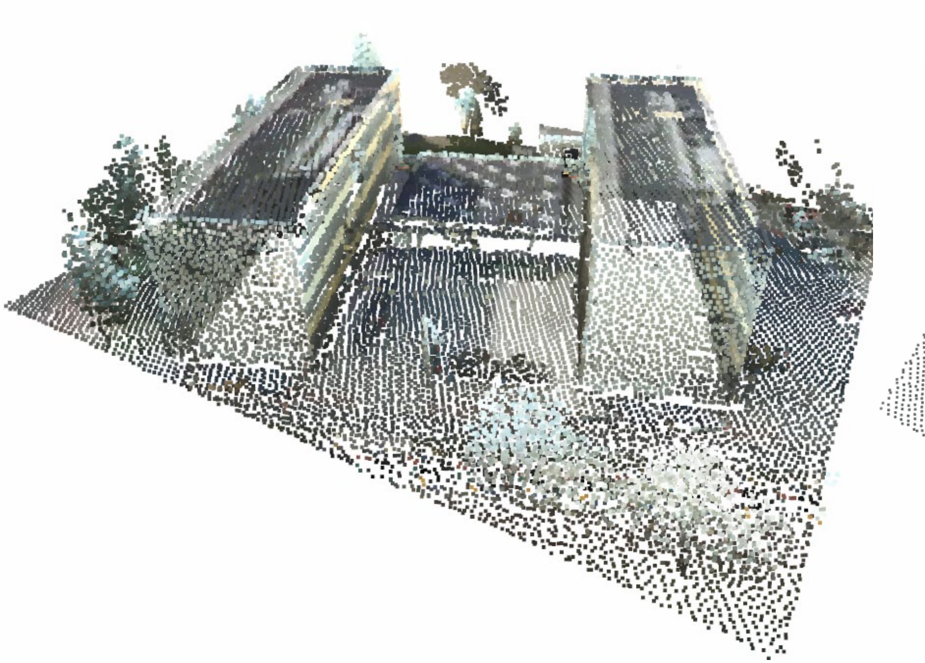
- AHN2 + Fugro terrestrial LiDAR (outliers)

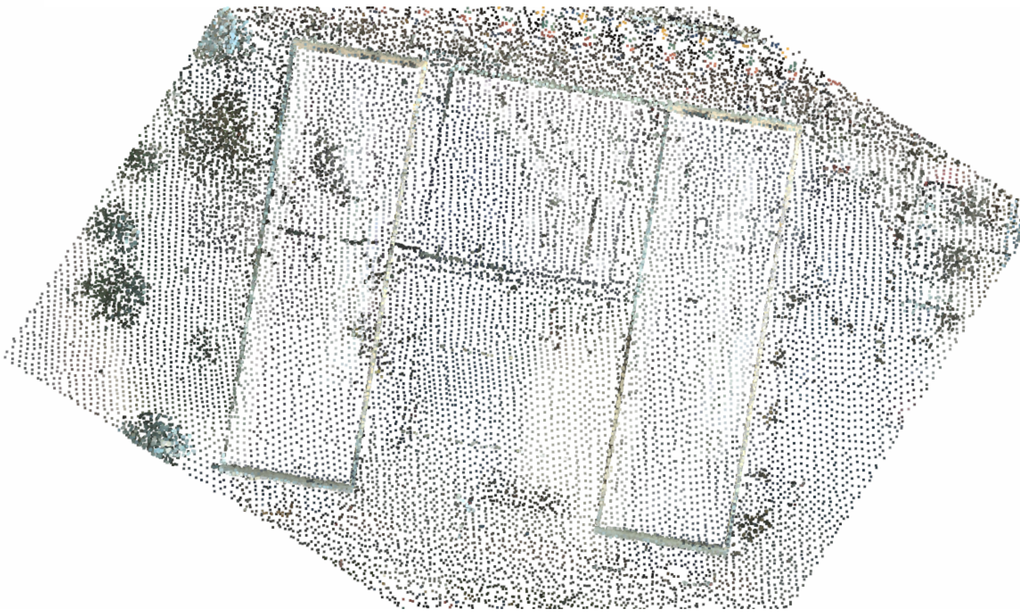




DEMO 3: OTB BUILDING

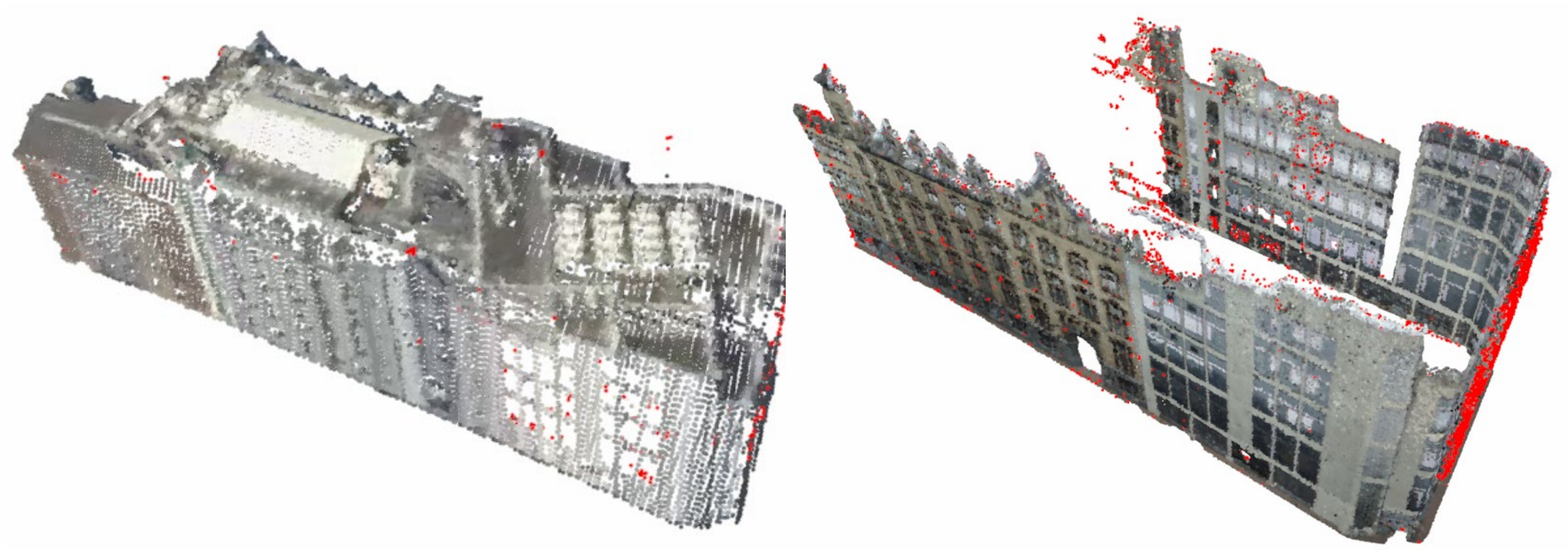
- AHN2 + Fugro terrestrial LiDAR (Simplified)





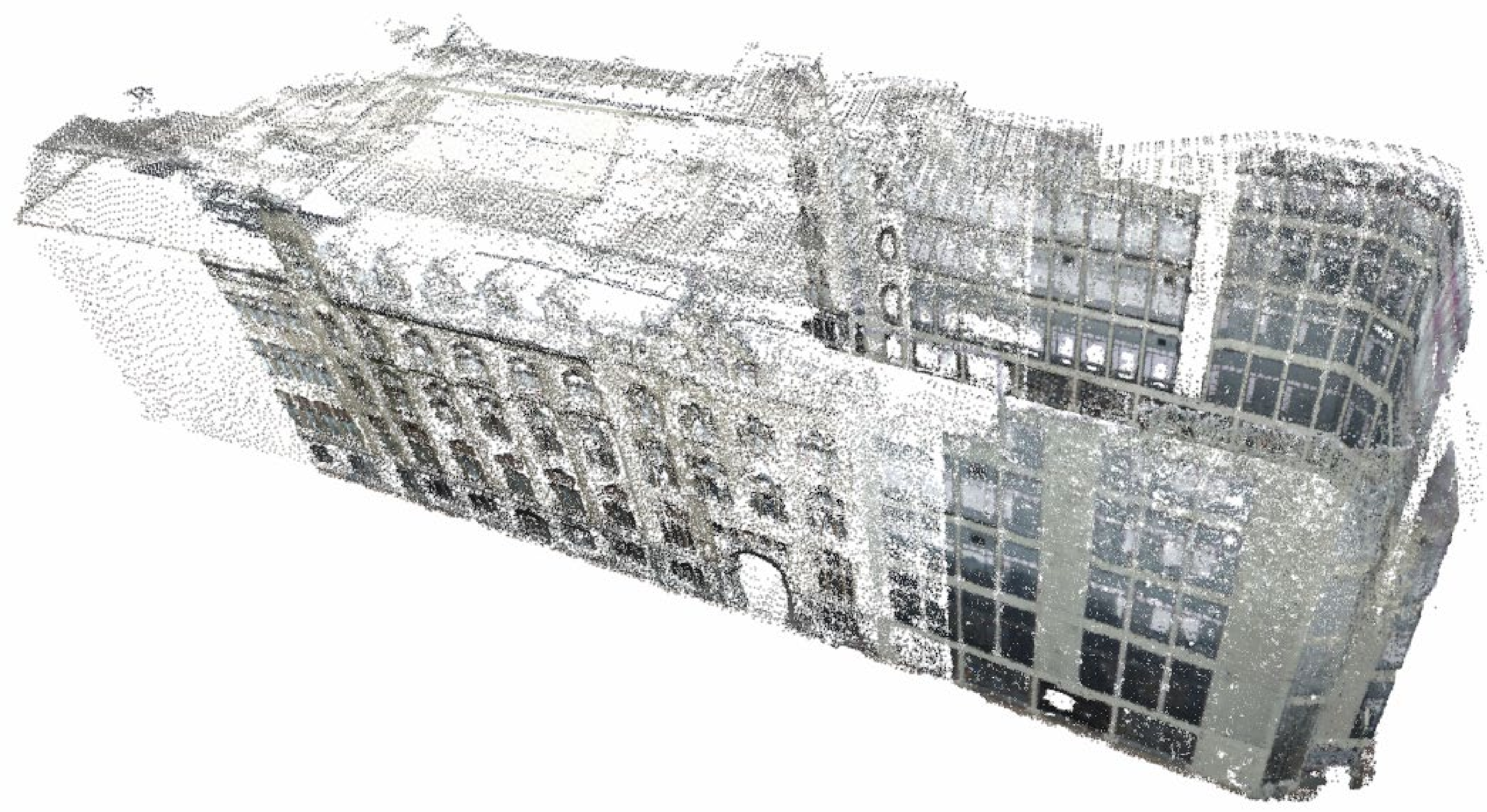
DEMO 4: AMSTERDAM BUILDING

- AHN2 + Cyclomedia panoramic imagery point cloud (Outliers)



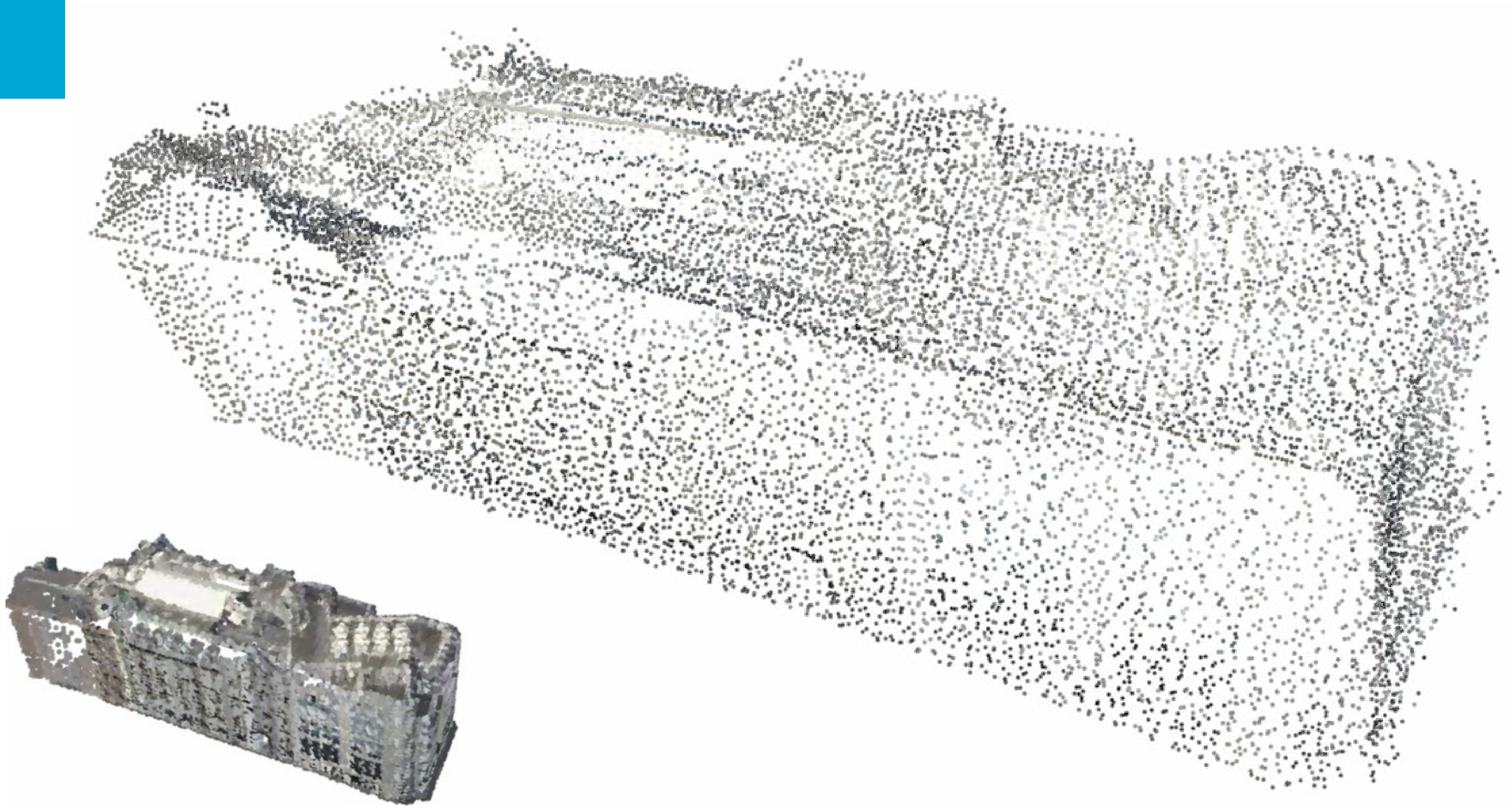
DEMO 4: AMSTERDAM BUILDING

- AHN2 + Cyclomedia panoramic imagery point cloud (Outlier cleaned)



DEMO 4: AMSTERDAM BUILDING

- AHN2 + Cyclomedia panoramic imagery point cloud (Simplified)



RESULTS

- Efficiency analysis: CPU run time (second)

	House(252915)	Villa(1668625)	OTB(4204626)	Amst(833584)
Outlier removal	N/A	24.19	57.91	14.09
Weight calculation	0.61	4.11	12.24	2.15
curvature estimation	0.17	0.91	2.44	0.47
Subsample	5.02	6.82	1.01	0.30
WLOP initialization	12.25	155.71	525.57	27.81
WLOP iteration x 3	0.60	11.94	28.19	4.95
Color	0.01	0.08	0.19	0.06
Total	18.66	203.76	627.55	35.74

- Most expensive: radius neighbor search in CPU
- Suggestion: Adapt to Multi-threading or GPU!

Content

VALIDATION

- Check list

1. Edge points preservation

2. Uniform density

3. Noise smoothing

VALIDATION

- 1. Edge points preservation

- Rely on user input sample radius and curvature threshold
- Check only qualitatively by visualization

VALIDATION

- 2. Uniform density

- Qualitatively by visualization
- OR Quantitatively by density standard deviation

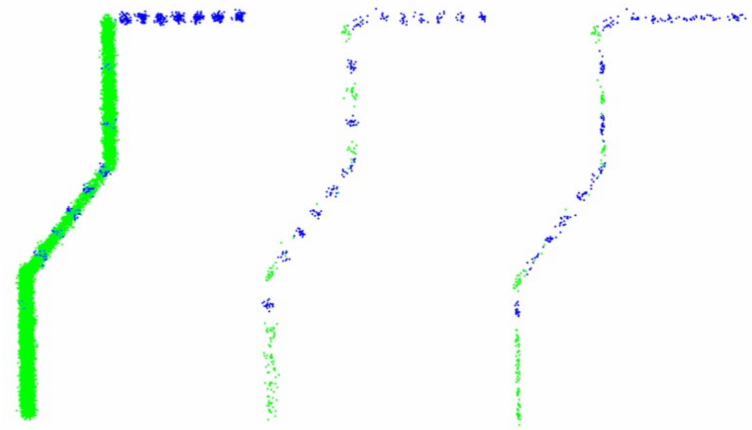
	House	Villa	OTB	Amst
Original point cloud	0.0506	7.4009	2.7294	0.6384
After Sub-sampling	0.0016	0.0217	0.0044	0.0053
After WLOP smoothing	0.0016	0.0212	0.0047	0.0052

Table 5-2: Density standard deviation changes for different phases.

VALIDATION

- 3. Noise smoothing

- Qualitatively by visualization



- Quantitatively: distance to intrinsic model (only in virtual)

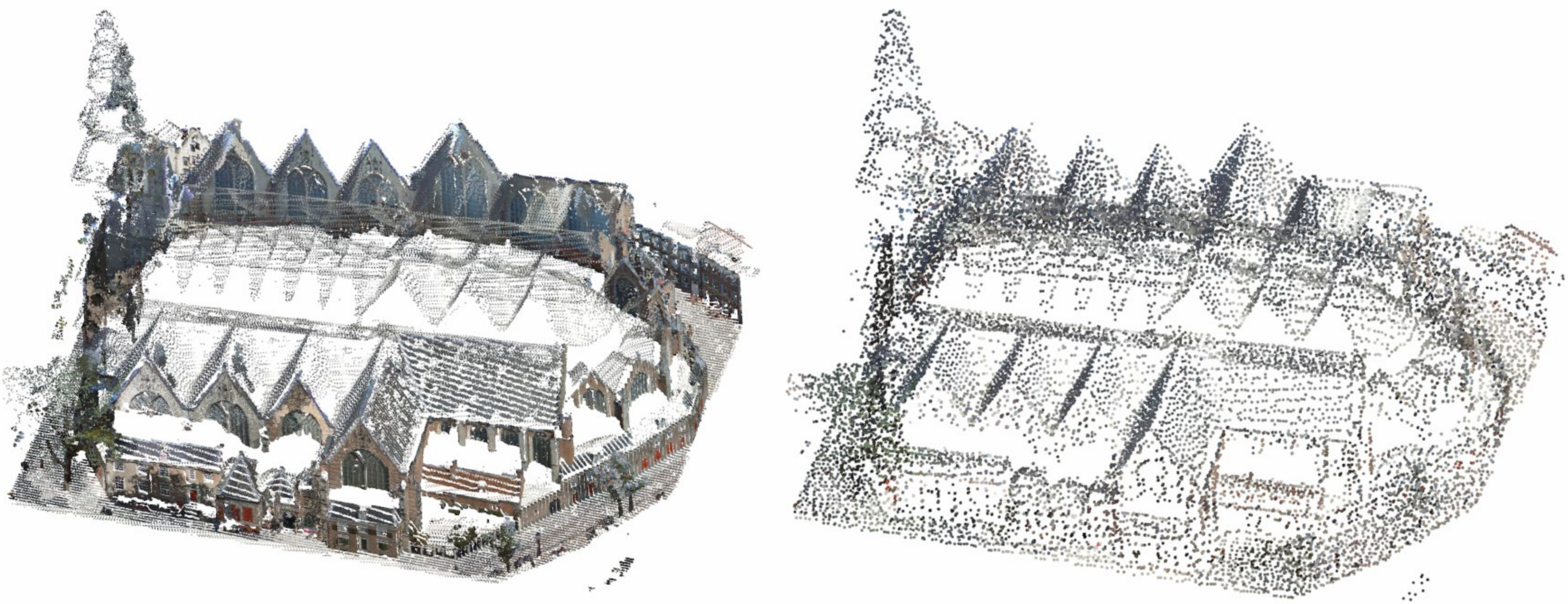
	Max	Avg	Std
Noisy input	0.478	0.077	0.059
Noisy input simplified result	0.434	0.055	0.054

Table 5-3: Noise smoothing effect analysis for House demo added with Gaussian noise.

Content

LIMITATION

- Holes (only reduce points); Edge extraction sensitive to severe noise



Content

CONCLUSION

- Answer to research question

Q: *Which algorithms are most suitable to fuse roof and façade point clouds into an edge-aware and uniformly dense color point cloud?*

A: We designed the most appropriate algorithms for fusing roof and façade point clouds either by developing algorithms ourselves or using and adapting existing algorithms *and integrated into a unified algorithm pipeline*, where outlier removal and simplification are performed and integrated that can produce an *outlier-cleaned, noise-reduced, edge-aware and uniform* point cloud.

CONCLUSION

- Contributions and difference from other methods

- **Outlier removal**
  - Ours can handle both singly scattered and small cluster of outliers without over-removing artifacts and can be applied in processing any point cloud

- **Simplification**
  - Different from others focusing on one or two objectives of noise reduction, uniform density and edge-awareness, we can achieve them all in an integrated and unified pipeline

  - Suitable for large-size processing

  - Consider precision distribution according to data source

RECOMMENDATIONS

- Future work

- **Data**

- Rigorous quantitative precision estimation

- **Implementation**

- Radius neighbor search in multi-threading or GPU

- **Algorithm Components**

- Gaps and holes: filled in point clouds matching and registration
- Holes: distinguish between occlusion and real ones
- Edge points extraction: adapt globally to anti severe noise

QUESTIONS / SUGGESTIONS?