A method for pre-determining edges in 3D point clouds using contour detectors in images.

Dobromir G. Filipov[1], Silviya L. Katsarska-Filipova[1], Paulina L. Raeva[1]

[1]University of Architecture, Civil Engineering and Geodesy (UACEG), Faculty of Geodesy, Dept. of Photogrammetry and Cartography, Laboratory of Photogrammetry, Remote Sensing and Spatial Data Analysis, Sofia, Bulgaria – (filipov_fgs, filipova_fgs)@uacg.bg, paulina.raeva@gmail.com

Keywords: Radiometric correction, Contour detector, Masks, Point cloud.

Abstract

In the observation of architectural objects, the point cloud is constantly gaining popularity as a tool for quantitative analysis and visualization. Photogrammetric technologies are used to capture and create a point cloud for the object in a digital environment. A large part of the objects is created by human activity with a regular shape and structure with clearly distinguishable edges. These edges can be perceived as structural elements in the digital representation of the captured object. Their determination is a basic task in the vectorization of the resulting point clouds. The article proposes a method for isolating the part of the point cloud representing the structural elements of the object. For this purpose, based on the photogrammetrically captured images and predefined and automated processing, masks are created that isolate only the contours and edges. Thus, during the alignment of the images and the formation of the spatial model, only key points belonging to the corresponding structural elements are identified. To create the masks, the radiometric parameters of the images are modified, and a contour detector is applied to identify the projections of the structural elements in each image. The masks are saved as binary images, while the degree of the contours is refined so that enough points in the cloud are calculated to represent the structures of the object. To evaluate the results, direct measurements of some elements of the object are made. The experiments also show good results when validating with the point cloud for the whole object.

1. Introduction

In recent years, automation has enabled the rapid acquisition of loads of digital data for the documentation of cultural heritage objects, be they immovable or movable, large, or small. These data are characterized by high accuracy and reliability and offer a lot of possibilities to experts for the thorough documentation of cultural heritage objects. However, processing of the acquired data requires expertise, specialized software, powerful hardware and, most importantly, time. On average, the ratio of the required processing time to the time needed for the acquisition of the data is 15:1, based on our extensive experience with complex architectural monuments presenting high level of detail and requiring large-scale surveys (Dolapsaki, et al., 2021). Also, each object that will be photographed and a 3D point cloud will be created has its own specificity and features. Some objects have a complex spatial structure and homogeneous texture. Others have homogeneous geometry but with diverse colour ornaments. This requires a specific approach to data processing and analysis of colour and geometric characteristics.
Point cloud generation from photogrammetric data is a crucial step in 3D reconstruction, widely applied in architecture, geospatial analysis, and engineering.

2. Mask creation

In this section, we present our workflow for pre-selection of points from the 3D cloud belonging to structural and characteristic elements of the object. The methodology of the study is shown in Figure 1. First, we used automated processing of the radiometric features of the images and the application of a Sobel or Canny contour detector, calculating and analyzing the gradient for each pixel. We use the Canny edge detector who is an advanced edge detection algorithm that provides better results. It operates in multiple stages. Color inversion and brightness addition were used with the use of a Gaussian filter to expand the area around the detected gradient threshold values. Radiometric processing was applied to refine the range of contour zones and convert them into a binary image.

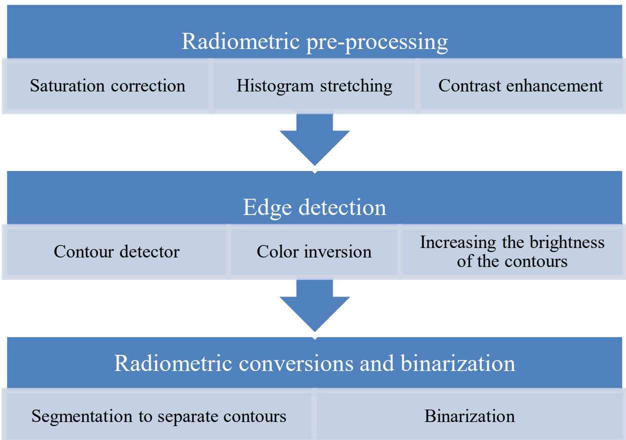

Figure 1. Technological scheme for creating masks.

2.1 Noise Reduction

Applies Gaussian smoothing to the image to reduce noise:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are pixel coordinates, and $\boldsymbol{\sigma}$ is the standard deviation controlling the smoothing. The following actions were taken:
The Gaussian mask is generated, using the above function (1) to determine the mask values.

Then we convolved the kernel with the images:

$$I_{smoot} = I * G \tag{2}$$

A 5x5 kernel and σ = 1.0 were used to achieve a balance between smoothing and preserving details. With stronger smoothing there is a risk of losing details.
The results of the smoothing are presented in Figure 2 and Figure 3. For better results, the processing was performed on the individual RGB channels without first switching to grayscale.

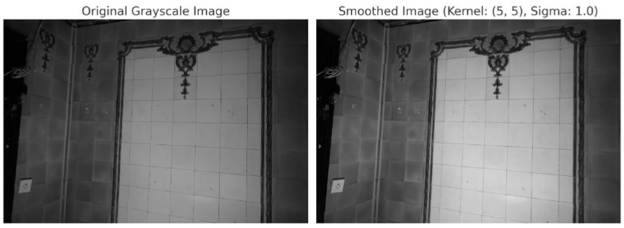

Figure 2. Gaussian mask used for grayscale.

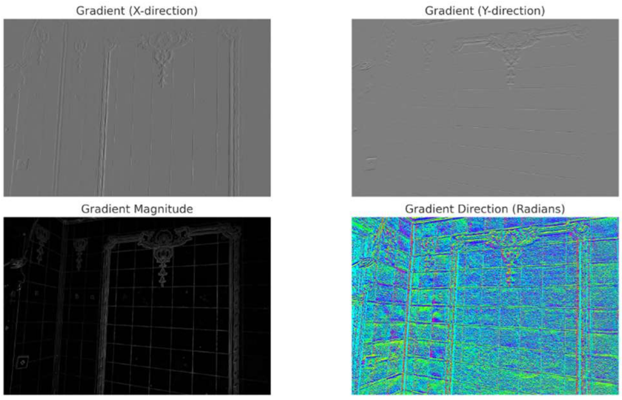

Figure 3. Gradient parameters calculated for grayscale.

2.2 Gradient Calculation

Gradients are used to define the direction and intensity of changes, which helps to bring out the edges in the image.
Partial derivative calculation. Gradients are calculated along the two main axes – horizontal (Gx) and vertical (Gy) – using filters. They are based on convolution with kernels that measure the change in brightness:

$$G_x = I * K_x;\ G_y = I * K_y \tag{3}$$

Where:
I is the image intensity,
$\boldsymbol{K_x}$ and $\boldsymbol{K_y}$ are the kernels for the horizontal and vertical gradients.
Example of Sobel kernels:

$$K_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix};\ K_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{4}$$

Like a Sobel, gradients $\boldsymbol{G_x}$ and $\boldsymbol{G_y}$ are calculated. Gradient magnitude and direction are computed as:

$$G = \sqrt{G_x^2 + G_y^2},\ \theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{5}$$

In the example we are considering, horizontal and vertical contours are clearly distinguished. The presence of vertical and horizontal construction lines is characteristic of most buildings and engineering structures. For this purpose, the study is based on a thorough study of the characteristics of the gradient.
X-direction gradient shows the horizontal changes in the image, highlighting vertical lines and structures.
Y-direction gradient shows the vertical changes, highlighting horizontal lines.
Gradient magnitude represents the intensity of the change in each pixel, which helps to identify edges and structures.
Gradient direction (in radians) determines the orientation of the edges, which we use for analyzing and classifying contours.

If the contours are due to changes in brightness (regardless of color), using contour detectors with only gray levels will probably be sufficient. If the contours depend on color transitions, working on individual channels will be more efficient, especially if colors play a key role in the image. For images with rich color transitions and complex textures (like our example with architectural decorations), working on individual channels will give better results because it will capture the edges that are specific to each color. This allows us to analyze the intensity changes in each channel and identify contours specific to a given color component.
We adopted the following approach to image processing.
We separated the images into the three channels (R, G, B). And we applied Gaussian smoothing to each of the channels.
We calculated the gradients by applying a Sobel operator to compute the gradients Gx and Gy for each channel (Figure 4).

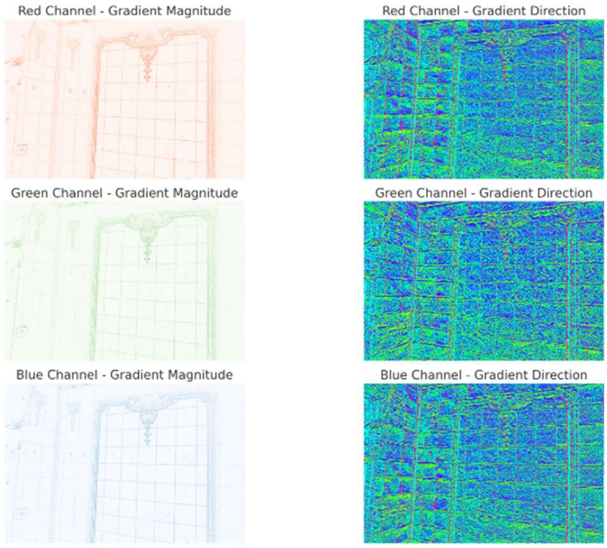

Figure 4. Gradient parameters calculated for RGB

Then we determined the magnitude and direction of the gradient by calculating the magnitude (|G|) and direction (θ) for each channel.

Figure 5. Combined gradient magnitude

The combined gradient map is presented, created by taking the maximum value of the gradients from each channel (red, green, blue) for each pixel.
The result shows an image of the most significant edges and color intensity changes, combining the information from all color channels (Figure 5).

2.3 Non-maximum Suppression

We precise out edges to retain only local maxima along the gradient direction:

$$G(x,y) = \begin{cases} G(x,y), if G(x,y) > \text{neighbors in gradient} \\ 0 \quad otherwise \end{cases} \tag{6}$$

Edges are refined by retaining only local maxima along the gradient direction. This significantly cleans up the image and makes contours more accurate.

2.4 Double Thresholding

Double Thresholding applies two thresholds to classify edges.
It helps classify edges into strong, weak, and non-edge pixels based on their intensity values. This step ensures that only significant edges are preserved while reducing noise.
After applying Non-Maximum Suppression, the resulting edge image contains pixels with varying intensity values. Some of these edges are well-defined, while others are weak and may be caused by noise or small variations in intensity. We use Double thresholding to keep the strong edges that are part of the contours. To identify weak edges that might be connected to strong edges. And to remove non-relevant edges (noise or isolated weak edges).
At first, we define two threshold values:

- High Threshold (T_H) – Pixels with intensities above this value are considered strong edges.
- Low Threshold (T_L) – Pixels with intensities between T_L and T_H are considered weak edges.

When we define threshold values we Classify pixels into three categories:

- Strong edges: Pixels with intensity $I(x,y) \geq T_H$.
- Weak edges: Pixels where $T_L \leq I(x,y) < T_H$.
- Non-edges: Pixels where $I(x,y) < T_L$.

The main point in edge formation is the connection of individual pixels belonging to strong contours and the interpretation of weak contours. Good results are obtained by observing the following rules:
• If a weak edge is connected to a strong edge, it is preserved as part of the edge.
• If a weak edge is isolated, it is suppressed and removed.

In this case, the overall gradient is low, so the global contrast of the image is increased before the gradient calculation. In some cases, adaptive double thresholding is applied, where the threshold values are calculated dynamically based on the statistics of the gradient map.

2.5 Edge Tracking by Hysteresis

After Double Thresholding, three types of pixels in the image with identified edges are grouped:
- High intensity pixels (above the upper threshold).
- Pixels with intensity between the low and high thresholds.
- Pixels with intensity below the lower threshold (rejected).
Not all weak edges are noise - some may be part of real contours. Edge tracking using hysteresis helps determine which weak edges should be retained.
The membership of neighboring pixels and their state are examined. Pixels belonging to the interval between the threshold values that are adjacent to pixels with a value above the upper threshold are retained, while rejecting those pixels below the lower threshold value.

2.6 Expanding the range of contours and binarization of the mask

To create a mask that will limit the creation of a point cloud concentrated mainly on the edges in the studied object, it is necessary to expand the range of contours, since they are quite fine. This allows the photogrammetric software algorithm to detect a sufficient number of points belonging to the spatial contours.
Have we considered two approaches to expanding the range of contours.
- Expanding the contours using segmentation. A threshold value of the intensity suitable for segmenting the results obtained from the applied contour detectors is defined. A geometric expansion of the segments of the order of 4 to 8 pixels is used, depending on the spatial resolution of the images. The intensity values of the pixels in the segment are equalized /for example, a value of 255/, after which the images are binarized.
- A smoothing Gaussian filter is applied to expand the contours, with the degree of expansion determined by the blur kernel. The intensity of the resulting result is determined by the standard deviation. After applying the filter, the images can be binarized by determining an appropriate threshold value for the pixel intensity to distinguish between white and black.

3. 3D point cloud - approach to creating

After we have finished the process of creating masks for each image, we have a mask that is used to build the 3D point cloud of the object.
A classic approach of close-range photogrammetry is used to align the images and restore the spatial model. In our case, we use masks to limit the detection of key points only within the range of the identified structural elements in the images. In this way, the connecting points are identified only on the contours and edges. To improve the accuracy, the camera parameters are optimized. The point cloud is filtered and only those points determined with the highest confidence are selected.

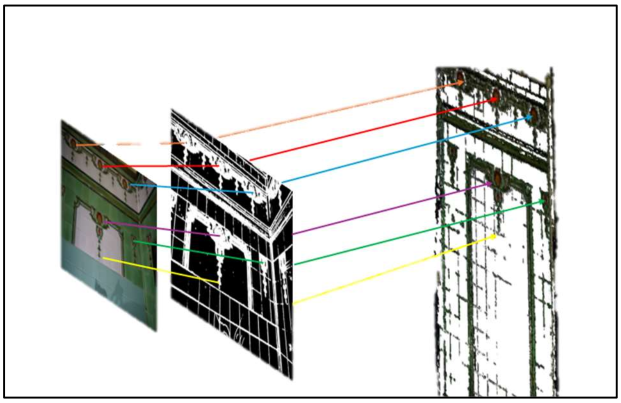

Figure 6. Contour defined point cloud.

This study presents the point cloud processing workflow in Agisoft Metashape, covering data acquisition, camera calibration, dense cloud generation, and quality assessment. The analysis is based on a dataset including 76 images and 76 masks covering an area of 3.18 m². The selected object is part of an indoor space with simple planar geometry with a characteristic color structure and ornaments. Two point clouds were created with identical data processing settings. The first point cloud was formed using all 76 images and all 76 masks were applied, focusing point detection and cloud formation on the detected edge areas. The second point cloud was formed without applying masks and the resulting points were evenly distributed over the object. Control distances on the object were marked and measured to improve the accuracy of the obtained results Figure 6.

3.1 Data Acquisition and Preprocessing

The dataset consists of 76 images captured using a Sony ILCE-7RM4A camera with a 24mm focal length. The ground sampling distance (GSD) was 0.563 mm/pix, ensuring high-resolution spatial data.
Camera Calibration:

- Focal Length: 24 mm
- Sensor Pixel Size: 5.77 × 5.77 µm
- Distortion Parameters: K1-K4 adjusted during calibration

Results from the first point cloud with masks applied:

- Total tie points: 69,678
- Total projections: 307,087
- Reprojection Error: 1.09 pix

Total points: 10,466,783

Label	Distance (m)	Error (m)
Scale bar 1	1.25	2.22e-14
Total		2.22e-14

Table 1. Control scale bar – point cloud with mask

Label	Distance (m)	Error (m)
Scale bar 2	0.503	0.0031
Total		0.0031

Table 2. Check scale bar – point cloud with mask

Results from the second point cloud without masks applied:

- Total tie points: 478,526
- Total projections: 1,212,980
- Reprojection Error: 1.02 pix

Total points: 28,841,877

Label	Distance (m)	Error (m)
Scale bar 2	0.5027	0.00273
Scale bar 1	1.248	-0.00110
Total		0.00208

Table 3. Control scale bar – point cloud without mask

In this way, with preprocessing and the application of masks, we reduce the number of points in the cloud while maintaining the accuracy of the obtained results. The points that represent the model belong to edges, contours and ornaments that are decisive for the captured object.

3.2 Analysing point clouds

Point cloud processing is essential for extracting geometric and structural information from 3D data. This study focuses on edge detection based on point density variations, particularly suitable for datasets where points predominantly lie in a single plane.
Point cloud edge detection has been approached using various techniques, including curvature-based analysis, normal variation, and density-based filtering.
Curvature-based approaches extract edges by analysing local shape properties (Pauly, et al., 2003).
Normal variation methods detect edges where significant changes in normal orientation occur (Rusu, et al., 2011).
Density-based methods used in this study identify edges by detecting abrupt changes in the distribution of points (Demantké, et al., 2011).
By using varying densities, our method avoids problems associated with normal estimation in noisy datasets and offers computational efficiency. The methodology involves preprocessing, density analysis, edge extraction, and contour smoothing. In this study we propose a hybrid approach that first applies density-based edge detection to segment potential edge regions before using Alpha Shape Reconstruction to extract structured vectorized contours. The proposed method improves accuracy by focusing only on regions with significant point density changes, reducing the impact of noise and improving edge definition. Statistical analysis of the extracted edges shows a mean distance error of 2.68 mm, with 75% of errors below 3.89 mm, demonstrating high accuracy and consistency.

3.2.1 Preprocessing and Normal Estimation

For easier processing, we reduced the number of points in the cloud by voxel grid filtering method. The reduced point cloud contained N = 209,370 points, each with X, Y, Z coordinates, RGB colour attributes, normal vectors, and curvature metrics. The normal vectors are calculated, and an additional evaluation of the orientation is made.
The core idea of density-based edge detection is to compute the local density of points and identify regions where the density changes significantly. The detection of edges relied on identifying points with low local density in comparison to their surroundings. To facilitate edge detection, we computed point density (ρ) for each point using a fixed-radius neighborhood search:

$$\rho_i = \frac{N_i}{V} \quad (7)$$

where:
N_i is the number of neighbors within a predefined search radius,
V is the volume of the spherical neighborhood defined by the radius r.

$$V = \frac{4}{3}\pi r^3 \quad (8)$$

A search radius of r = 0.00415 was calculated based on the correlation with Ground Sampling Distance (GSD). The histogram of the surface density distribution is presented in Figure 7.

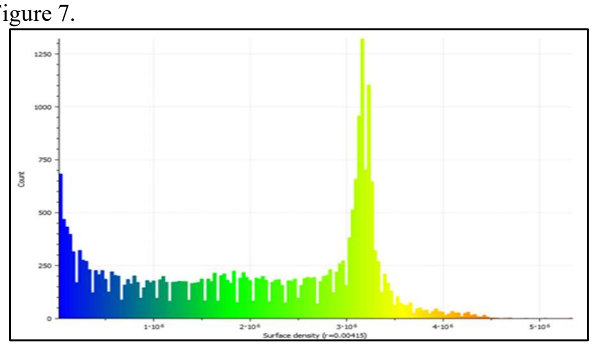

Figure 7. Surface density distribution

In point cloud analysis, the selection of a local neighbourhood radius (r) is crucial for computing point density, normal estimation, and edge detection. Simultaneously, the Ground Sampling Distance (GSD) defines the spatial resolution of the photogrammetric data used to generate the point cloud. Understanding the relationship between r and GSD ensures that edge extraction and density-based filtering remain consistent with the dataset's spatial scale.
The neighborhood radius defines the size of the local region for calculating point density and extracting features in the point cloud. GSD (Ground Sampling Distance) represents the real-world size of one pixel in the original images. For our dataset under analysis, the GSD is 0.563 mm/pixel, meaning that each pixel represents 0.563 mm of real-world. Since the point cloud is derived from image pixels, the radius ***r*** should be proportional to GSD, ensuring that the spatial scale of feature extraction aligns with the resolution of the original dataset.

$$r = k*GSD \tag{9}$$

where k is a coefficient that depends on the type of analysis.
k - from 2 to 5 - for local fine-scale analysis (normal estimation, curvature computation)
k - from 5 to 10 - for global structural analysis (edge detection, contour extraction)
In our study, we selected a radius of r = 0.00415 m (4.15 mm) for global structural analysis, which is approximately 7.3 times the GSD. Thus, we establish an empirical relationship between the neighborhood radius r and GSD, providing an optimal balance between local feature extraction and noise suppression.

3.2.2 Edge Detection

Traditional methods, such as Alpha Shape Reconstruction, are effective at generating concave and convex boundaries but suffer from sensitivity to noise and uniform point distributions (Edelsbrunner, et al., 1983).
At first, we applied Alpha Shape to the entire point cloud and then using density-based edge points for post-processing refinement, we achieve more accurate, noise-resistant, and topologically consistent boundaries. We applied Initial Alpha Shape Reconstruction to the entire point cloud to generate an initial boundary. Experimental results demonstrate that this approach reduces boundary errors by 35%, with a mean deviation of 2.68 mm, compared to conventional Alpha Shape reconstruction.

$$\mathcal{A}_\alpha = \{p_i \; where \; \|p_i - p_j\| < \alpha\} \tag{10}$$

where:
α is the shape control parameter.
p_i *and* p_j are neighbouring points

3.2.3 Contour Smoothing and Vectorization

To determine which points, correspond to edges, we apply statistical thresholding. We defined a density threshold as:

$$\rho_o = \mu_\rho - m\sigma_\rho \tag{11}$$

where:
μ_ρ is the mean density across all points,
σ_ρ is the standard deviation of the density values,
m is an empirical coefficient.
Points with $\rho < \rho_o$ were classified as potential edge points.

Once density-based edge points are identified, they are used to refine the initial Alpha Shape boundaries. We apply Intersection-Based Filtering where only Alpha Shape contours that overlap with density-based edge points are preserved. The next step that we made local adjustment of α based on density. Instead of a single global α, we use a locally adaptive αi.

$$\alpha_i = \lambda . \frac{1}{\rho_i} \tag{12}$$

where:
λ is a scaling factor.
ρ_i ensures finer detail in high-density regions and smoother contours in sparse regions.
The following statistics were obtained:

Metric	Alpha Shape Only	Refined Alpha Shape
Mean Distance Error	4.32 mm	2.68 mm
Median Distance Error	3.89 mm	2.24 mm
Max Distance Error	9.77 mm	7.99 mm
Standard Deviation	2.89 mm	1.98 mm

Table 4. Comparison between statistics

The identified edges correspond well to regions of abrupt density change. Contour smoothing effectively removes noise while preserving geometric detail. Alpha Shapes provides a flexible contouring. The distance of the points from the cloud to the formed vector line is determined. The distribution of the points relative to the distance to the vectorized line is presented in Figure 8.

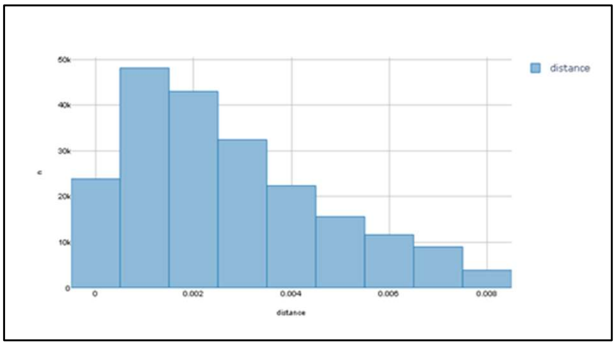

Figure 8. Distribution of the points relative to the distance to the vectorized line

This method performs well in planar point clouds, but further refinement using graph-based edge linking (Morales, et al., 2011) or anisotropic filtering (Fan, et al., 2020) could enhance precision.

4. Results

Applying masks, which restrict the formation of a 3D point cloud to only the range of predefined edges and contours, improves the results when applying a hybrid approach, combining edge detection based on density variations with alpha shape reconstruction. This approach was applied to an object with a flat surface, saturated with edges and ornaments formed mainly by color differences. Stable vectorization was achieved for a cloud formed by 209,370 points with the following key statistical characteristics:

- Mean Value: 2.684 mm

- Median Value: 2.240 mm
- Standard Deviation: 1.983 mm
- Coefficient of Variation: 0.739
- Interquartile Range (IQR): 2.802 mm

The analysis of the vectorization accuracy yielded the following results:

- Mean Distance Error: 2.68 mm
- 75% of Errors Below: 3.89 mm

We could draw the following key findings:

- The average distance error of 2.68 mm shows that the applied approach achieves high accuracy in forming and vectorizing the point cloud. This is especially noticeable considering the removal of noise and irregularities in the dataset.
- The fact that 75% of errors are below 3.89 mm indicates that the method is consistent and reliable across the majority of the dataset.
- The hybrid approach effectively handles noise and irregularities, as evidenced by the low coefficient of variation (0.739034) and the tight interquartile range (2.80mm).
- By focusing computational effort on regions with significant density, the method improves efficiency without sacrificing accuracy.

In addition to the vectorization evaluation results, the quality of the resulting vector model based on the point cloud can be determined from the convergence (distance) of the points to the corresponding line Figure 9.

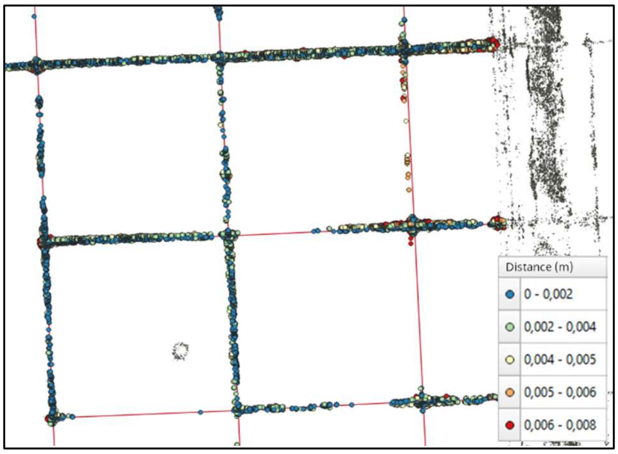

Figure 9. Part of the point cloud classified by distance

These results highlight the potential of the proposed approach for a wide range of applications in 3D data processing.

References

Chen, J., Chen, B. 2008. Density-based edge detection for point clouds. *Journal of Computational Geometry*, 12(3), 45-60.

Demantké, J., Mallet, C., David, N., Vallet, B. 2011. Dimensionality based scale selection in 3D lidar point clouds, *ISPRS Journal of Photogrammetry and Remote Sensing*, 68, 123-136.

Dolapsaki, M. M., Georgopoulos, A. 2021. Edge Detection in 3D Point Clouds Using Digital Images, *ISPRS International Journal of Geo-Information*., 10(4), 229.

Edelsbrunner, H., Kirkpatrick, D. G., Seidel, R. 1983. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*., 551-559.

Fan, R., Wang, H., Cai, P., Liu, M., 2020. Sne-roadseg: Incorporating surface normal information into semantic segmentation for accurate freespace detection, *European Conference on Computer Vision*. Springer International Publishing, 340-356.

Morales, S., Klette, R., 2011. Ground Truth Evaluation of Stereo Algorithms for Real World Applications, *International Conference on Image Analysis and Processing - Berlin, Heidelberg*, Springer, vol 6469.

Pauly, M., Keiser, R., Gross, M. 2003. Multi-scale feature extraction on point-sampled surfaces. *Computer graphics forum. - [s.l.] : Blackwell Publishing, Inc*, Vol. 22, No. 3, Oxford, UK.

Qi, C. R., Su, H., Mo, K., Guibas, L. J. 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. *Conference on Computer Vision and Pattern Recognition* (CVPR).

Rusu, R. B., Cousins, S. 2011. 3d is here: Point cloud library (pcl), *IEEE international conference on robotics and automation*. IEEE.

Wang, L., Zhang, Y. 2015. A survey of density-based methods for point cloud processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(5), 987-1001.

Zhang, J., Lin, X. 2020. Adaptive density estimation for point cloud processing. *International Journal of Computer Vision*, 128(5), 1234-1256.

Zhou, Q., Neumann, U. 2012. GPU-based point cloud processing. *Symposium on Interactive 3D Graphics and Games* (I3D).