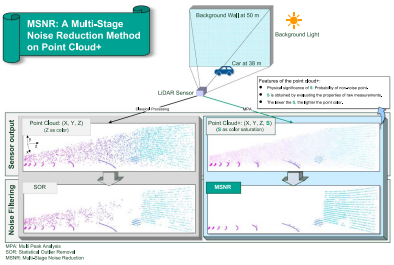
Sensor applications

MSNR: A Multistage Noise Reduction Method on Point Cloud+

Gongbo Chen, Santosh Kumar Kasam, and Christian Wiede
Fraunhofer Institute for Microelectronic Circuits and Systems (IMS), 47057 Duisburg, Germany

Manuscript received 4 April 2024; accepted 25 April 2024. Date of publication 1 May 2024; date of current version 20 May 2024.

Abstract—In autonomous driving, direct time-of-flight light detection and ranging (LiDAR) is one of the most emerging techniques due to its high depth resolution and detection ability in specific environments. Tasks, such as object recognition and scene segmentation, use distance information in LiDAR point clouds as ground truth to improve performance. However, in harsh measurement environments, the amount of noise in a LiDAR point cloud increases significantly. In this case, an algorithm using LiDAR data will be misleading, resulting in performance degradation. Although many noise filtering algorithms at point cloud-level were proposed, their effectiveness is limited, as traditional point clouds contain only distance information. In this letter, a method with multistage noise reduction (MSNR) is proposed. It takes point cloud+ as input. Comparing with the method using conventional point cloud as input, MSNR improves accuracy by 29.50% under high background light.



Index Terms—Sensor applications, data processing, light detection and ranging (LiDAR), noise reduction, point cloud.

I. INTRODUCTION

Light detection and ranging (LiDAR) sensors have been used for nearly four decades in metrology, geology, and space applications [1]. Recently, autonomous driving and the robotics industry started using LiDAR to perceive the surrounding environment [2]. LiDAR provides high-resolution distance information over other sensors, such as cameras, radars, and ultrasonic sensors. Among the LiDAR family, single photon avalanche diode (SPAD)-based direct time-of-flight (d-TOF) flash LiDAR is reliable and durable, since it is a solid-state ranging system without sophisticated moving mechanisms [3]. As a tradeoff, it has a small field-of-view and the optical power received by each pixel is lower than that of a scanning LiDAR [4]. Thus, it is often applied for small and medium range detection.

A SPAD-based d-ToF LiDAR system has a significant amount of noise in the raw data under harsh measurement conditions, e.g., high background light. Neural network-based multi-peak-analysis (NNMPA) [5] is proposed as a software solution to cope with this problem. It extracts important features from LiDAR histograms and determining distance information with a neural network. The outputs are pixel-wised data pairs, including multiple possible distances, as well as the corresponding probablity values. This enables new possibilities to outlier detection and to distinguish between noise and object information. However, the significance have not been discussed and validated at the point cloud-level yet.

In this letter, a point cloud filtering method with multistage noise reduction (MSNR) is proposed. MSNR takes the output generated by NNMPA [5] as input and filters noise points out.

The rest of this letter is organized as follows. In Section II, existing methods for improving the point cloud are introduced. In Section III, the methodology is explained. In Section IV, the used dataset and the evaluation metrics are specified. Afterward, the results are demonstrated and discussed. Finally, Section V concludes this letter.

Corresponding author: Gongbo Chen (e-mail: gongbo.chen@ims.fraunhofer.de).
Associate Editor: Dimitrie Popescu.
Digital Object Identifier 10.1109/LSENS.2024.3395929

The contributions of this work include: 1) introducing a new LiDAR data form point cloud+ and 2) proposing a point cloud+ processing method MSNR with preliminary results on its significance.

II. RELATED WORKS

Noise filtering of point clouds is usually achieved by analyzing the spatial correlation among points. Guided 3-D point cloud filter and iterative guidance normal filter are introduced to improve the quality of a point cloud in [6]. First, a surface normal for each point is estimated and a n-nearest-neighbors-based bilateral filter [7] is applied. Afterward, point positions are updated to match the filtered normals accordingly. This method is suitable for correcting jitter between the measured distance and the ground truth and cannot handle randomly distributed noise points caused by background light.

Statistical outlier removal (SOR) is a general purpose outlier removal method [8]. It determines whether a point is an outlier based on the sparsity of its neighborhood. Based on the mean absolute distance (MAD), a threshold T_{MAD}, is calculated as

$$T_{\mathrm{MAD}} = \mu_{\mathrm{MAD}} + \theta_{\mathrm{MAD}} f_{\mathrm{SOR}} \ . \tag{1}$$

Here, θ_{MAD} is the standard deviation, μ_{MAD} is the mean of the MADs, and f_{SOR} is a tunable parameter. The point with MAD greater than T_{MAD} is identified as an outlier. However, SOR does not account for the fact that distant points are sparser than near points in a LiDAR point cloud. To cope with this problem, dynamic SOR (DSOR) is proposed in [9]. It introduces a dynamic threshold T_{D} depending on the distance. DSOR is adopted in this work at Stage 2.

Another way to compensate noise points in a single point cloud is to consider the time dimension, i.e., processing multiple temporally continuous point clouds. However, it is rather for the purpose of recognition for motion patterns [10] and semantic segmentation [11].

In addition to noise filtering, data reduction is another issue that needs to be addressed. For example, Velodyne-64 has up to 440 000 points per point cloud [12]. This brings enormous challenges to data transmission and processing. To cope with this problem, Voxel grid

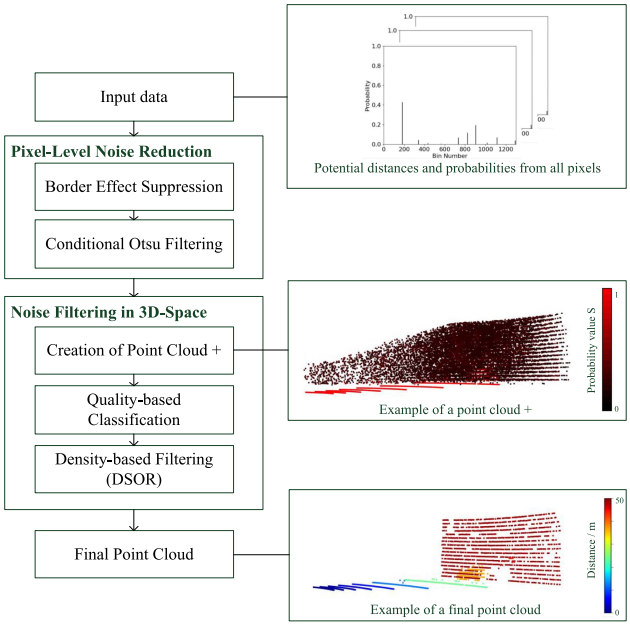


Fig. 1. Processing framework of MSNR.

filtering [13] was proposed to remove redundant points. It divides the point cloud into 3-D voxels and retains only one point per voxel. RANSAC Segmentation [14] is another classical method that removes the ground plane in a point cloud. These methods assume all data to be correct and are suitable to be applied on high-quality point cloud with barely false points.

Nevertheless, the impact of noise points is often overlooked. Besides, conventional point clouds contain only distance information, which makes it challenging for point cloud-based algorithms to identify noisy points. In harsh measurement situations, the amount of noise points increases substantially and reduces the reliability of these methods. In this work, the problem is addressed by incorporating point cloud+ (pixel-wise data pairs with multiple distances and quality values) and into MSNR.

III. METHODOLOGY

In this section, the methodology is described, including the specification of the input data and two filtering stages. The overall processing framework of MSNR is shown in Fig. 1, including a initial pixel-level noise reduction and noise filtering in 3D-Space using spatial relations and quality information in the point cloud+.

A. Input Specification

MSNR uses the distance prediction results of NNMPA [5] as input. Specifically, the data pairs $\{(d_1,\ s_1),\ (d_2,\ s_2),\ \ldots\}$ of each pixel are used, where d_n refers to the nth potential distance and s_n refers to the probability value to the nth distance (higher s_n indicate that the corresponding distance is more likely to be the true distance).

B. Stage 1: Pixel-Level Noise Reduction

In this stage, data pairs are processed individually within each pixel as follows.

1) Border-Effect Suppression: As discussed in [5], NNMPA suffers from the border effect, when the object locates at the junction of the two subregions. In this case, the distance information of an object is captured twice, resulting in two neighboring distance points. The interval between these two points is less than a laser pulsewidth. Thus, the border effect is considered present, if $\exists I_i = (d_i, s_i)$ and $I_j = (d_j, s_j)$, where

$$B_n - W_{\mathrm{L}} < d_i < B_n \quad \text{and} \quad B_n < d_j < B_n + W_{\mathrm{L}}\ . \tag{2}$$

Here, B_n refers to the position of the nth border and W_{L} refers to the width of the used laser pulse. Afterward, the corresponding s are modified by

$$s_i' = \begin{cases} s_i + s_j, & s_i > s_j \\ 0, & \text{else} \end{cases} \quad \text{and} \quad s_j' = \begin{cases} s_i + s_j, & s_j > s_i \\ 0, & \text{else.} \end{cases} \tag{3}$$

2) Conditional Otsu Filtering: According to [5], s_n refers to the probability of the distance being the object distance. Thus, the conditional Otsu filtering, modified from the automatical thresholding method Otsu [15], is applied to remove the noise points with small s_n in the following three steps.

1) The range of s_n (between 0 and 1) is divided into M subranges D of

$$D = \frac{1}{M}\ . \tag{4}$$

2) The between-class variance θ_B^2 is determined by

$$\theta_B^2(k_1,\ k_2) = \theta_B^2(k_1) + \theta_B^2(k_2) \tag{5}$$

where $\theta_B^2(k_n)$ is given by

$$\theta_B^2(k) = \frac{(\mu_T\ \omega(k) - \mu(k))^2}{\omega(k)\,(1 - \omega(k))} \tag{6}$$

where $k \in \{1, \ldots, L\}$ refers the k^{th} subrange and

$$\omega(k) = \sum_{i=1}^{k} \frac{n_i}{N}\ ,\quad \mu(k) = \sum_{i=1}^{k} i\,\frac{n_i}{N}\ ,\ \text{and}\ \ \mu_T = \sum_{i=1}^{M} i\,\frac{n_i}{N}\ . \tag{7}$$

Therein, n_i refers to the number of s_n values within subrange i and N refers to the total number of features in a histogram.

3) The optimal threshold k_{Opt} is determined by

$$k_{1,\mathrm{Opt}},\ k_{2,\mathrm{Opt}} = \mathrm{argmax}_{1 \le k_1 < k_2 < L}\ \theta_B^2(k_1,\ k_2)\ . \tag{8}$$

When the objective function is bimodal, the term associated with k_2 in the equation is neglected.

Finally, the points with the probability value lower than k_{Opt} are considered as noise points and are removed.

C. Stage 2: Noise Flitering in 3D-Space

The previously mentioned processing preliminarily filters the noise points individually on each pixel. In the following steps, the association between different pixel points is studied by projecting remaining points into the 3D-space, resulting in a point representation transformation of data pair from $(d,\ s)$ to $(x,\ y,\ z,\ s)$. Compared with conventional point clouds, these points contain probability value s and the total number is larger than the number of pixels (each pixel could correspond to more than one data pair). In the following context, the set of these points is called point cloud+.

1) Gaussian Filter for Probability Diffusion: Due to the consistency of objects in space, the higher the average probability value of a local point-cluster, the more likely the center point of that cluster represent the object distance. This characteristic is addressed using a Gaussian

filter $G(p)$ and the probability value s_p is updated by

$$s_p = G(p) = \frac{\sum_{c \in C_p(r)} s_c \, \omega_c}{\sum_{c \in C_p(r)} \omega_c} \tag{9}$$

where p represents a point in the point cloud. $C_p(r)$ is the set of neighbor points of p with radius r. The Gaussian weights is calculated using

$$\omega_c = \mathrm{e}^{-\frac{d_{c,p}^2}{2\sigma_p^2}} \tag{10}$$

with a tunable parameter σ_p. Therein, $d_{c,p}$ refers to the distance between point c and p.

2) Probability-Based Classification: Afterward, the remaining points are further classified into object and noise using a segmented linear function. First, the points are subdivided as: 1) division of points regarding background light: LiDAR systems can often acquire background light photon rate r_B (using a counting mode or determining the constant component in a histogram) while measuring distance. Thus, the points are divided into N_B subsets according to the trend of the probability value decreasing with background light and 2) division of points regarding distance: The reflected laser intensity is inversely proportional to the square of the distance and ultimately influences the probability value s. Thus, the points are further divided into N_D subsets according to the trend of the probability value decreasing with distance. Afterward, N subsets of points are formed. Each subset is filtered by adopting corresponding thresholding function f_n, which is determined by the exhaustive line fitting method as

$$f_n = \frac{(d - d_\mathrm{L})\,(s_{\mathrm{R,Opt}} - s_{\mathrm{L,Opt}})}{d_\mathrm{R} - d_\mathrm{L}} + s_{\mathrm{L,Opt}} \tag{11}$$

by optimizing the F1-score F_1 as

$$s_{\mathrm{L,Opt}},\ s_{\mathrm{R,Opt}} = \mathrm{argmax}_{s_\mathrm{L},\, s_\mathrm{R} \in [0,1]}(F_1(s_\mathrm{L}, s_\mathrm{R})) \tag{12}$$

where d_L and d_R refer to the left and the right boundary positions of the subset. s_L and s_R refer to the probability values corresponding to d_L and d_R.

3) Dynamic Statistical Outlier Removal: Point-clusters from objects typically have a relatively high point density, while noise points are randomly distributed. Thus, DSOR is applied. Similar to the implementation in [9], the dynamic threshold T_D is computed using

$$T_\mathrm{D} = f_\mathrm{DSOR}\, T_\mathrm{MAD}\, x \tag{13}$$

where f_DSOR is a factor for point spacing, T_MAD is calculated by (1), and x is the x-coordinate value of the corresponding point.

IV. CONCLUSION

A. Dataset Specification

The dataset used in this work includes 16 point clouds and refers to the prediction result of NNMPA on the raw histograms, which is 12 data pairs $\{(d_1, s_1), \ldots (d_n, s_n)\}$ per pixel. The raw histograms are generated by using the data simulator CARLA and a histogram simulation tool based on the Monte Carlo principle. The detailed histogram generation process refers to [5] and the used parameters in this work can be found in Table 1.

B. Evaluation Metrics

The goal of the filtering method is to remove as much noise point as possible and meanwhile to retain as many object points as possible.

Table 1. Parameter Specification of Dataset

Parameter Name	Value
Detection Range / m	(0.5, 60)
Distance of Car d_Car / m	{25, 38}
Distance of Wall d_Wall / m	50
Background Photon Rate r_B / MHz	{1, 2, 3, 4, 5, 6 ,7, 8}
Laser Photon Rate r_L / MHz	$28000/d^2$
Number of Measurements N_M	400
Depth Resolution R_D / ps	312.5
Laser Width W_L / m	0.75
Number of Pixels	6208

Thus, the following metrics are considered:

1) Error Tolerance ΔE: A point (d, s) is annotated as a true point, if

$$(1 - \Delta E)\, d_\mathrm{GT} < d < (1 + \Delta E)\, d_\mathrm{GT} \tag{14}$$

where d_GT is the ground truth distance. Otherwise, it is annotated as a false point.

2) Quality of Point Cloud Q: It describes the percentage of true points in the current point cloud and is given by

$$Q = \frac{N_{\mathrm{T},n}}{N_{\mathrm{Total},n}} \tag{15}$$

where $N_{\mathrm{T},n}$ refers to the number of true points, and $N_{\mathrm{Total},n}$ refers to the total number of points in the current point cloud after stage n.

3) Benchmarking Method DSOR: In order to compare the effect on noise filtering before and after the introduction of point cloud+, DSOR (applied in the last stage of MSNR as well) is used as the benchmark algorithm. The same NNMPA outputs as in MSNR are used to generate the conventional point cloud by extracting distance corresponding to the largest s in each pixel. Afterward, the point cloud is fed to DSOR to get the benchmarking result.

C. Parameter Determination

In this section, the parameters on each stage are determined based on the analysis on the given dataset.

According to Chen et al. [5], the border effect is present at each region border d_Border. Since the laser pulsewidth is 5 ns (i.e., 0.75 m), the border effect is examined and removed within the range of $(d_\mathrm{Border} \pm W_\mathrm{L})$.

σ_p in the Gaussian filter determines the impact of neighbor points to s of a certain point. In this work, σ_p is set to 1.5. $s_{\mathrm{L,Opt}}$ and $s_{\mathrm{R,Opt}}$ are determined according to (12) as

$$s_{\mathrm{L,Opt}} = \begin{cases} 0.77 - 0.05\,(r_\mathrm{B} - 3)\,, & 30 < d \le 40 \\ 0.26 - 0.015\,(r_\mathrm{B} - 3)\,, & 40 < d \le 60 \end{cases} \tag{16}$$

and

$$s_{\mathrm{R,Opt}} = \begin{cases} 0.769 - 0.0523\,(r_\mathrm{B} - 3)\,, & 30 < d \le 40 \\ 0.2595 - 0.0145\,(r_\mathrm{B} - 3)\,, & 40 < d \le 60. \end{cases} \tag{17}$$

The parameter f_SOR influences the basic threshold for removing noise points, while f_DSOR influences the overall threshold according to the distance of a point. In this work, the best f_SOR and f_DSOR in the range of $(0,\ 3)$ are obtained by maximizing Q under different range of r_B as

$$f_\mathrm{SOR} = \begin{cases} 2.5\,, & 0 < r_\mathrm{B} \le 3 \\ 0.5\,, & 3 < r_\mathrm{B} \le 6 \\ 0.1\,, & 6 < r_\mathrm{B} \le 8. \end{cases} \quad \text{and} \quad f_\mathrm{DSOR} = 0.02 \tag{18}$$

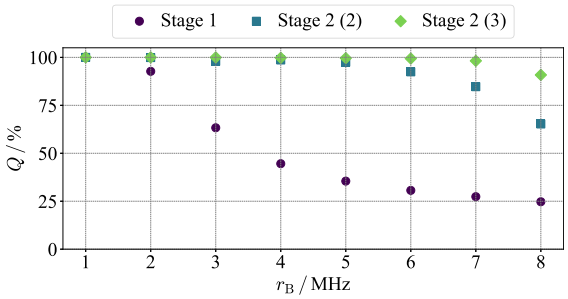


Fig. 2. Quality of the point cloud after each processing stages under different background lights.

Table 2. Comparison of Point Cloud Quality (Q)

r_B / MHz	1	2	3	4	5	6	7	8
MSNR / %	100	100	100	99.6	99.4	99.2	98.3	89.1
DSOR / %	100	100	100	99.5	95.7	82.2	67.5	57.5

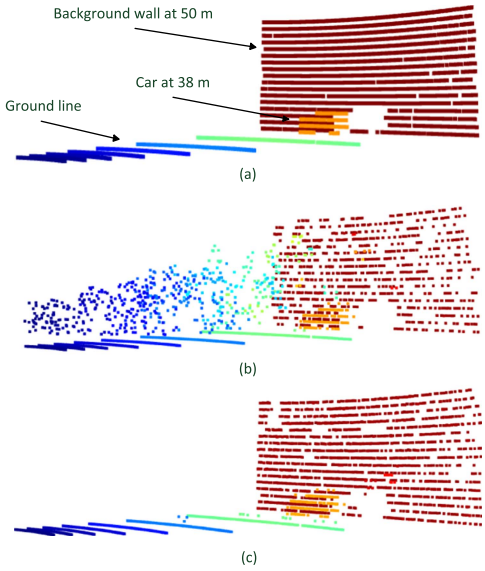


Fig. 3. Point cloud examples with different processing methods. (a) Ground truth. (b) DSOR. (c) MSNR.

D. Stage-Wise Performance

Fig. 2 shows the performance of each filtering stage under different r_B. With $r_B \leq 2$ MHz, target distance and noise are easily distinguishable. In this case, the majority of noise can be removed on Stage 1. With $3 \leq r_B \leq 6$ MHz, noise reduction only with Stage 1 is no longer sufficient, since only the information within each individual pixel is accounted in. By introducing the point-to-point spatial relationship with s on Stage 2 (2), the remaining noise is further removed, resulting in a Q-value of at least 97.52%. When $6 < r_B \leq 8$ MHz, noise cannot be completely distinguished from target points by s. By calculating the density of the point-clusters, Stage 2 (3) further removes the sparsely distributed noise and Q ultimately reaches over 90.84%.

E. Comparison

Table 2 lists the quanlity value Q of MSNR and DSOR under different background lights. It can be seen that MSNR based on point cloud+ has a better robustness than DSOR on conventional point cloud. Fig. 3 shows a comparison between MSNR and DSOR in an example scene with $r_B = 6$ MHz. It can be seen that without considering the quality value s, there is still a significant amount of noise points in short range after appling DSOR [shown in Fig. 3(b)] and there is a noticeable absence of background wall points. This could pose a challenge for subsequent object recognition. In contrast, the result of MSNR contains only a small number of outliers and is visually clearer, shown in Fig. 3(c).

V. CONCLUSION

In this letter, a new data form point cloud+ is introduced. The point cloud+ includes multiple potential distances in each pixel and probability values s. To validate its significance, MSNR is proposed. It processes the point cloud+ in two stages and applies DSOR to generate the final data. Its performance under different background light intensities is demonstrated by comparing it with DSOR using conventional point cloud. It can be concluded that by using the point cloud+ as input, MSNR shows reliable performance in scenes with low background light and superior robustness in scenes with high background light in the dataset used in this letter. For example, the accuracy of MSNR point cloud is 29.50% higher than DSOR with 8 MHz background photon rate. Future works could be: 1) investigation on the effective range of the algorithm and generalization and 2) development of algorithms using point cloud+ for other purposes, such as object recognition and tracking.

REFERENCES

[1] F. Amzajerdian, V. Roback, A. Bulyshev, P. Brewster, and G. Hines, "Imaging flash LiDAR for autonomous safe landing and spacecraft proximity operation," in *AIAA SPACE*, p. 5591, 2016.

[2] Y. Li and J. Ibanez-Guzman, "LiDAR for autonomous driving: The principles, challenges, and trends for automotive LiDAR and perception systems," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, Jul. 2020.

[3] A. K. Aijazi, L. Malaterre, L. Trassoudaine, and P. Checchin, "Systematic evaluation and characterization of 3D solid state LiDAR sensors for autonomous ground vehicles," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 199–203, 2020.

[4] J. Hu, B. Liu, R. Ma, M. Liu, and Z. Zhu, "A 32 x 32-pixel flash LiDAR sensor with noise filtering for high-background noise applications," *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 69, no. 2, pp. 645–656, Feb. 2022.

[5] G. Chen, F. Landmeyer, C. Wiede, and R. Kokozinski, "Feature extraction and neural network-based multi-peak analysis on time-correlated LiDAR histograms," *J. Opt.*, vol. 24, no. 3, Feb. 2022, Art. no. 034008.

[6] X.-F. Han, X.-Y. Yan, and S.-J. Sun, "Novel methods for noisy 3D point cloud based object recognition," *Multimedia Tools Appl.*, vol. 80, pp. 26121–26143, 2021.

[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. Sixth Int. Conf. Comput. Vis.*, 1998, pp. 839–846.

[8] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 1–4.

[9] A. Kurup and J. Bos, "DSOR: A scalable statistical filter for removing falling snow from LiDAR point clouds in severe winter weather," 2021, *arXiv:2109.07078*.

[10] D. Salami, S. Palipana, M. Kodali, and S. Sigg, "Motion pattern recognition in 4D point clouds," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process.*, 2020, pp. 1–6.

[11] H. Shi, G. Lin, H. Wang, T.-Y. Hung, and Z. Wang, "SpSequenceNet: Semantic segmentation network on 4D point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4573–4582.

[12] "HDL-64E S3 datasheet: High definition real-time 3D LiDAR," 2023. Accessed: Mar. 29, 2023.

[13] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[15] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.