

# Article Unsharp Mask Guided Filtering for Acoustic Point Cloud of Water-Conveyance Tunnel

Jisong Wang <sup>1,2,\*</sup>, Xuewu Zhang <sup>1</sup>, Zhuo Zhang <sup>1</sup> and Xiaolong Xu <sup>1</sup>

- <sup>1</sup> College of Internet of Things Engineering, Hohai University, Changzhou 213022, China; zhangxw@hhu.edu.cn (X.Z.); 20011573@hhuc.edu.cn (Z.Z.); xuxl@hhu.edu.cn (X.X.)
- <sup>2</sup> College of Computer Science and Technology, Huaiyin Normal University, Huaian 223300, China
- \* Correspondence: jssywjs@126.com

Abstract: The inner-surface damage of water conveyance tunnels is the main hidden danger that threatens their safety and leads to serious accidents. The method based on the principle of acoustic reflection is the main means of inspecting damage to water-conveyance tunnels. However, affected by the tunnel environment and equipment noise, the obtained acoustic point cloud model inevitably suffers from noise, which can produce erroneous results. Therefore, we proposed a novel filtering method, called unsharp-mask-guided filtering for 3D point cloud, to reduce the impact of noise on the acoustic point cloud model of water-conveyance tunnels. The proposed method fuses the ideas of guided filtering and the unsharp masking technique and extends them to the 3D point cloud model by considering the position of the point. In addition, edge-aware weighting mean is also used to retain the edge features of the point cloud model while smoothing the noise points. The experimental results show that our method can obtain impressive results and a better performance in both the acoustic point cloud model of the tunnel and the simulated point cloud model than many state-of-the-art methods.

**Keywords:** unsharp masking; edge-aware weighting; guided filtering; 3D acoustic point cloud; water-conveyance tunnel

# 1. Introduction

Water-conveyance tunnels are the main structures of water-diversion projects. When water-diversion projects enter the operational period, special attention must be paid to the health status of water-conveyance tunnels. The regular inspection of water-conveyance tunnels is an important means of ensuring their safe operation and health management. In recent years, in terms of the problem of tunnel-damage detection, researchers have proposed many methods and technologies (e.g., side-scan sonar, multi-beam sonar, and synthetic-aperture sonar) [1-4]. The detection method of tunnel damage based on the principle of pulse reflection forms an acoustic point-cloud model by sampling the reflected echo, and then detects and identifies the damage through acoustic point-cloud data [5]. However, affected by the noise of tunnel environments and inspection systems, as well as the noise of carrier equipment (robot carriers and submersibles, etc.), the raw acoustic point-cloud models obtained from these methods inevitably suffer from noise, which makes it difficult to obtain useful information and increases the fuzziness and randomness of information features. In this case, the detection methods that rely on information features produce erroneous results very easily. Therefore, a filtering operation must be performed on the raw acoustic point-cloud model of the tunnel before further processing (e.g., object recognition [6,7] or 3D reconstruction [8]).

Inspired by the excellent results of the guided image-filtering algorithm introduced by He et al. [9], the classical sharpness enhancement technique of unsharp masking [10–12] and the edge-aware weighted guided image-filtering algorithm [13], this study extends these



Citation: Wang, J.; Zhang, X.; Zhang, Z.; Xu, X. Unsharp Mask Guided Filtering for Acoustic Point Cloud of Water-Conveyance Tunnel. *Appl. Sci.* 2022, *12*, 6516. https://doi.org/ 10.3390/app12136516

Received: 19 April 2022 Accepted: 23 June 2022 Published: 27 June 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). effective methods to the point-cloud model of tunnel and proposes a new filtering method, called unsharp-mask-guided filtering for acoustic point-cloud of water-conveyance tunnel, by considering the position of the point instead of the pixel value. The experimental results show that our method can outperform several competing methods (e.g., bilateral filter [14], moving least-squares filter [15], and guided 3D point-cloud filter [16]), both on the acoustic point-cloud tunnel model and the simulated point-cloud model.

The contributions of our work can be summarized as follows: (1) the idea of guided filtering and the unsharp masking technique are fused to design a novel filtering approach for an acoustic point cloud for water-conveyance tunnels; (2) the idea of edge-aware weighting is used to retain the features while smoothing some of the edges of the point cloud; (3) a comprehensively experimental evaluation between our method and several competing denoising methods is conducted on the tunnel point-cloud model and simulated point-cloud model, respectively.

The remainder of this paper is structured as follows. A brief overview of the background and related work is given in Section 2. Section 3 describes the principle and implementation process of our proposed algorithm in detail. The experimental process and experimental results on different point-cloud models are demonstrated in Section 4. Conclusions and discussions are presented in Section 5.

#### 2. Background and Related Work

## 2.1. Acoustic Point-Cloud Filtering Method

Due to the complexity of underwater environments, the research on the filtering methods of acoustic point clouds, especially the acoustic point-cloud data of water-conveyance tunnels, is limited. Feng et al. [17] proposed a multi-beam point-cloud filtering algorithm that considers the features of underwater terrain. This method is based on the Random Sample Consensus (RANSAC) ideal to fit the local plane, and the coplanar vector feature is used to remove outliers. However, it does not consider that outlier surfaces may be actual obstacles. Xie et al. [18] used the intensity information and elevation information of points to filter the underwater acoustic point cloud. First, a rough classification was carried out according to the intensity information; then, accurate classification and filtering were realized using the intensity information and the elevation information of the seed point and its surrounding points. However, its filtering effect depends entirely on the accuracy of the seed point. Cui et al. [19] studied the combined filtering algorithm of a multi-beam underwater acoustic point cloud. They divided the noise into large-scale and small-scale noise, and then used radius filtering and bilateral filtering, respectively, to achieve point-cloud smoothing. However, this method has the problem of the mistaken deletion of complex terrain. Wang et al. [20] proposed a filtering method that fuses acoustic properties and intensity. They used the filtering method in the process of forming the acoustic point cloud and achieved good results. However, it is not suitable for the filtering of already formed acoustic point-cloud models. Therefore, these studies are not suitable for denoising the acoustic point clouds of water-conveyance tunnels.

By contrast, several studies have been conducted on filtering methods for ground-laser point clouds. Bilateral filtering, originally introduced by Tomasi et al. [21], is a robust edge-preserving filtering method, which has been applied to 3D mesh filtering [22–24]. However, these methods need to form mesh models, whose generation process inevitably suffers from noise [25]. Moreover, these methods are always imperfect at retaining the detailed features of point-based data models [26]. In order to overcome this problem, Han et al. [16] proposed an effective point-cloud denoising method by considering the position information of the point. However, in order to reduce the calculation cost, they ignore the fact that the point may be included in different neighborhoods when calculating the conversion coefficient. Therefore, the filtering effect in the acoustic point-cloud model is not ideal and the edge features of the point-cloud model are not well maintained in the smoothing process. Alexa et al. [15] proposed an approach that relies on the idea of implicitly defining a surface for a given set of points. The main idea is to define a

projection procedure that projects any point near the point set onto a surface. Next, the MLS (moving least-squares) surface is defined as the set of points projecting onto itself. This method achieves a good smoothing effect, but it smooths the edge features of the point set because of the multiple projection process. Therefore, it is not suitable for surface-feature reconstruction based on the tunnel point cloud.

Currently, most point-cloud filtering methods are based on laser-point cloud data. Owing to the particularity and complexity of the water conveyance tunnel project, the research on acoustic point-cloud filtering methods is limited. Therefore, the acoustic pointcloud filtering methods of water-conveyance tunnels have important practical research value.

#### 2.2. Classical Guided Filtering

Guided image filtering is a well-known time-efficient feature-preserving smoothing operator [9]. The filtered results are obtained by considering the content of the guidance image, and the guidance image can be the input image itself or a different image. This assumes that the filtered output image is a linear transformation of the guidance image in a window. It can transfer the structure of the guidance image to the filtered output, enabling new filtering applications, such as dehazing and feathering. Therefore, guided filtering is essential to the rapid production of edge-feature-preserving denoising applications.

The guided image filtering is given by

$$\hat{I}_i = a_k G_i + b_k, \ \forall i \in \omega_k.$$
<sup>(1)</sup>

where  $\hat{l}$  and G are the filtered output image and the guidance image, respectively. i is the index number of the pixel within window  $\omega_k$  centered at pixel k.  $a_k$  and  $b_k$  are two constants confined to the window  $\omega_k$ . Their values can be calculated by solving

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left( (a_k G_i + b_k - I_i)^2 + \varepsilon a_k^2 \right).$$
<sup>(2)</sup>

where  $\varepsilon$  is a regularization parameter that penalizes large values of  $a_k$  and I is the original input image. The optimal values of  $a_k$  and  $b_k$  are calculated by

$$a_{k} = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_{k}} I_{i}G_{i} - \overline{I}_{k}\overline{G}_{k}}{\sigma_{k}^{2} + \varepsilon}, \ b_{k} = \overline{I}_{k} - a_{k}\overline{G}_{k}.$$
(3)

Here,  $\overline{G}_k$  and  $\sigma_k^2$  are the mean and variance of G in  $\omega_k$ , respectively.  $|\omega|$  is the number of pixels in  $\omega_k$  and  $\overline{I}_k$  is the mean of I in  $\omega_k$ . To simplify the computation, we assume the guidance image to be the same as the input image [9]. Therefore, we can obtain

$$a_k = \sigma_k^2 / (\sigma_k^2 + \varepsilon), \ b_k = (1 - a_k)\overline{G}_k.$$
 (4)

According to Equations (1) and (4), the regions with variance  $(\sigma_k^2)$  much larger than  $\varepsilon$  are preserved, and the regions with variance  $(\sigma_k^2)$  much smaller than  $\varepsilon$  are smoothed. However, because  $\varepsilon$  is a constant in all the neighborhood windows, halos are unavoidable when the filter is forced to smooth some edge features.

#### 2.3. Edge-Aware Weighted Guided Filtering

When the value of a certain pixel at a sharp edge is large, the pixel value in the neighborhood centered on this pixel is affected. The original pixel with a small value becomes larger, and the phenomenon of expansion occurs at the edge. Therefore, the filtered image appears as halo artifacts. If the halo artifacts are not processed, problems such as color distortion and blurred details in the filtered image occur. To overcome the halo artifacts, edge-aware weighted guided image filtering is proposed [13]. It inherits the advantages of both global and local filtering methods in terms that: (1) its complexity

is same as that of guided image filtering, and (2) it can avoid halo artifacts such as the currently used global smoothing filters.

Let  $\sigma_G^2(p')$  be the variance of *G* in the 3 × 3 window,  $\Omega(p')$ . An edge-aware weighting  $\Gamma_G(p')$  is defined by utilizing the local variances of all pixels within a 3 × 3 windows. It is given by

$$\Gamma_G(p') = \frac{\sigma_G^2(p') + \varsigma}{\frac{1}{M} \sum\limits_{p=1}^M \sigma_G^2(p) + \varsigma}.$$
(5)

Here, *M* is the number of the pixels and  $\varsigma$  is a small constant value to prevent the denominator from being 0. Consequently, they use  $\varepsilon = \lambda/\Gamma_G(p')$  to perform edge enhancement. The weighting  $\Gamma_G(p')$  measures the importance of pixel p' in the overall guidance image. If the pixel p' is at an edge, the value of  $\Gamma_G(p')$  is usually larger than 1. Therefore,  $\varepsilon$  becomes smaller and  $a_k$  in Equation (4) becomes larger. This implies that the sharp edges of *G* in  $\omega_k$  are enhanced.

Unlike the original guided filtering, the value of  $\varepsilon$  in edge-aware weighted guided filtering is no longer fixed, but adapts according to the different edge characteristics. Therefore, the edge features can be maintained well while smoothing the image.

#### 2.4. Unsharp Masking

Unsharp masking is a classical sharpness-enhancement technique [12]. The enhancement framework can be summarized as follows. The original image is decomposed into two layers by applying a linear shift-invariant low-pass filter (e.g., the Gaussian filter) [27]. The resulting image is called the base layer, containing the main structure of the original image. The difference produced between the base layer and the original image is called the detail layer, revealing the fine details of the original image. It can be described by

$$\hat{I} = \lambda (I - \Phi_L(I)) + I, \tag{6}$$

where I represents the enhanced image and I is the original input image.  $(I - \Phi_L(I))$  denotes an unsharp mask, where  $\Phi_L$  represents a low-pass filter.  $\lambda$  controls the effect of enhancement achieved at the output and is the only coefficient to be estimated. Essentially, guided filtering has the same edge enhancement function as unsharp masking by transferring structure from an additional guidance image.

Based on the information above, we derived a novel guided point-cloud denoising formulation from the original guided filter, with the estimation of only one coefficient, akin to the formulation of unsharp masking. To overcome the halo artifacts caused by  $\varepsilon$  being a constant value, as described in Section 2.3, an edge-aware weighted method is also proposed to perform edge enhancement while smoothing the edge of the point cloud.

### 3. Method

Our approach is motivated by the classical sharpness-enhancement techniques of unsharp masking, guided image filtering, and edge-aware weighted guided image filtering. However, these methods cannot be directly applied to 3D point clouds because some point cloud models have only spatial information, and no intensity-attribute information. Therefore, we construct a new widely applicated filtering method by using the position information of points to replace the pixel value in three classical image-filtering methods above. The block diagram of proposed method is shown in Figure 1.



Figure 1. The block diagram of proposed method.

Given target point-cloud set  $P = \{p_i \in R^3\}$  and guidance point-cloud set  $G = \{g_i \in R^3\}$ , the neighborhood  $N(p_i) = \{p_{ij} \in P\}$  and  $N(g_i) = \{g_{ij} \in G\}$  are searched by using the k-nearest neighbor (KNN) method, where  $p_{ij}$  and  $g_{ij}$  represent the *j*th neighboring point of  $p_i$  and  $g_i$ , respectively. As mentioned before, we also assume that the filtered output point cloud has a linear model with the guidance point cloud in a particular neighborhood, that is,

$$\boldsymbol{p}_{ii}' = a_i \boldsymbol{g}_{ii} + \boldsymbol{b}_i. \tag{7}$$

Here,  $p'_{ij}$  represents the filtered output point of  $p_{ij}$ .  $a_i$  and  $b_i$  are the conversion coefficients in the neighborhood  $N(g_i)$ . As with Equation (2) and its optimization solution, we can obtain the values of  $a_i$  and  $b_i$  by

$$a_{i} = \frac{\frac{1}{|N(g_{i})|} \sum_{g_{ij} \in N(g_{i})} p_{ij} \cdot g_{ij} - \overline{p}_{i} \cdot \overline{g}_{i}}{\sigma_{g_{i}}^{2} + \varepsilon},$$
(8)

$$\boldsymbol{b}_i = \overline{\boldsymbol{p}}_i - a_i \overline{\boldsymbol{g}}_i. \tag{9}$$

where  $|N(g_i)|$  represents the number of points in  $N(g_i)$ .  $\varepsilon$  is a regularization parameter penalizing large  $a_i$ .  $\overline{g}_i$  and  $\sigma_{g_i}^2$  are the mean and variance of guidance point  $g_i$  in  $N(g_i)$ .  $\overline{p}_i$  is the mean of  $p_i$  in  $N(p_i)$ . Taking the calculation of  $\overline{p}_i$  in  $N(p_i)$  as an example, it is given by

$$\overline{\boldsymbol{p}}_{i} = \frac{1}{|N(\boldsymbol{p}_{i})|} \sum_{\boldsymbol{p}_{ij} \in N(\boldsymbol{p}_{i})} \boldsymbol{p}_{ij}.$$
(10)

The filtered output point can be obtained by Equation (7). However, the output point  $p'_{ij}$  has different values because  $g_{ij}$  may be contained in different neighborhoods. Therefore, we calculate the filtered point  $\hat{p}'_{ij}$  by averaging all possible values of  $p'_{ij}$  with

$$\hat{\boldsymbol{b}}_{ij}' = \frac{1}{N} \sum \left( a_i \boldsymbol{g}_{ij} + \boldsymbol{b}_i \right). \tag{11}$$

where N represents the number of neighborhoods which containing point  $g_i$ .

Motivated by unsharp masking, summarized in Equation (6), we insert Equation (9) into Equation (11) to eliminate  $b_i$ , and obtain

$$\hat{p}'_{ij} = \bar{a}_i (g_{ij} - \tilde{g}_i) + \tilde{p}_{i'}$$
(12)

where  $\overline{a}_i = \frac{1}{N} \sum a_i$ ,  $\widetilde{g}_i = \frac{1}{N} \sum \overline{g}_i$ , and  $\widetilde{p}_i = \frac{1}{N} \sum \overline{p}_i$ .

According to Equation (12), we can more intuitively understand how the unsharp mask guiled filtering achieves edge-preservation and structure-transferring effects. Specifically, the point  $p_i$  is smoothed to remove noise and the result is denoted by  $\tilde{p}_i$ . Next, to retain the fine details, an unsharp mask  $(g_{ij} - \tilde{g}_i)$  with fine features generated from the guide point cloud is added to  $\tilde{p}_i$  under the control of the coefficient  $\bar{a}_i$ , enabling the transfer of edge details from the guide point cloud to the filtered-output point cloud.

However, it is clearly visible from Equation (12) that the denoising effect is greatly affected by the number of k-neighborhood points, which determines the values of  $\bar{a}_i$ ,  $\tilde{g}_i$ , and  $\tilde{p}_i$ . If the value of *k* is too large, the resulting point cloud usually loses sharp edges, resulting in over-smoothing problem. Furthermore, if the value of *k* is too small, it cannot achieve the desired denoising effect.

Inspired by the edge-aware weighted guided image-filtering method [13], we introduce the edge-aware mechanism to reduce the impact of edge loss. We use  $\xi = \lambda / \Gamma_g$  to replace  $\varepsilon$  in Equation (8), and  $\Gamma_g$  is given by

$$\Gamma_g = \frac{\sigma_g^2(p') + \varsigma}{\frac{1}{n} \sum\limits_{p=1}^n \sigma_g^2(p) + \varsigma},$$
(13)

where  $\lambda$  is a regularization parameter that penalizes large  $a_i$ .  $\zeta$  is a small constant and its value is empirically set as  $(0.001 \times d)^2$ , while d is the farthest Euclidean distance between two points in the point-cloud model. n is the number of points in the point-cloud model.  $\sigma_g^2(p')$  represents the variance of the point in its k-nearest neighbors and the value of k' here is selected as 10. When the point is located at a sharp edge, the value of  $\Gamma_g$  is usually lager than 1 and  $\xi$  becomes small. That is, the punishment for  $a_i$  in Equation (8) is small and  $\overline{a}_i$  in Equation (12), it is large. Therefore, the fine edge features generated from the guidance point cloud can be maintained well while smoothing noise points.

The results of unsharp mask guided filtering are directly related to the selection of the guidance point-cloud model. Ideally, the ground-truth model without noise should be chosen as the guidance point-cloud model, but this may not be obtained in practical applications. Therefore, we use the raw noisy point cloud as the guidance point cloud. Compared with classical guided filtering, which needs to calculate two parameters (a, b), our novel filtering method needs to estimate only one coefficient *a*. At the same time, it can maintain the edge features while smoothing the point-cloud model.

#### 4. Results

Our approach was implemented using MATLAB 2019 on a PC with AMD R5-3500U CPU and 8 GB of memory. We show the results of our method tested on the pointcloud model of a water-conveyance tunnel, and conduct a comparative evaluation with other advanced algorithms. To verify the practicability of the proposed method, we also compared and evaluated the performance of our approach tested on various simulated 3D point-cloud models corrupted with Gaussian noise ranges from 1% to 3% of the diagonal length of the model bounding box. The datasets of the water-conveyance tunnel were derived from the simulated tunnel in the laboratory and the real tunnel on the site. The simulated 3D point-cloud datasets were derived from Large Geometric Models datasets, ModelNet datasets, and ShapeNet datasets [28], respectively.

#### 4.1. Parameters

As introduced in Section 3, the proposed algorithm has two key parameters in addition to the constant parameter  $\varsigma$  and the neighborhood parameter k' for edge perception. One is the parameter k of KNN, which is used to calculate the geometrical neighborhood. For point-cloud models with different numbers of points, the choice of k has a great influence on the filtering effect. Another key parameter is  $\lambda$ , which is a regularization parameter penalizing large  $a_i$ . It can simply control the filtering quality. Therefore, we selected a point-cloud model with a different number of points to analyze the impact on the results under different k and  $\lambda$ . The models and parameters used in this experiment are shown in Table 1. Figures 2–4 illustrate the results of our method using different values of kand  $\lambda$ , respectively. Tables 2 and 3 illustrate the quantitative evaluation of the results using the error metric  $D_{mean}$  in [29]. When verifying the influence of one parameter on the experimental results, we kept the other parameter constant.

Models	Points	Parameters				
Horse	48,485	4	10	k	50	100
Cactus	3280	4	10	k	50	100
Bunny	35,947	0.01		$\lambda \\ 0.1$		0.9

Table 1. The models and parameters used in experiment.



**Figure 2.** Filtering results with different *k* values on horse model ( $\lambda = 0.1$ ). (a) Horse model with noise; (b) filtering result with k = 4; (c) filtering result with k = 10; (d) filtering result with k = 50; (e) filtering result with k = 100; (f) the ground-truth model of horse.



**Figure 3.** Filtering results with different *k* values on cactus model ( $\lambda = 0.1$ ). (a) Cactus model with noise; (b) filtering result with k = 4; (c) filtering result with k = 10; (d) filtering result with k = 50; (e) filtering result with k = 100; (f) the ground-truth model of cactus.



**Figure 4.** Filtering results with different  $\lambda$  values on bunny model (k = 50). (**a**) Bunny model with noise; (**b**) filtering result with  $\lambda = 0.01$ ; (**c**) filtering result with  $\lambda = 0.1$ ; (**d**) filtering result with  $\lambda = 0.9$ ; (**e**) the ground-truth model of bunny.

**Table 2.** Analysis of the results with different *k* values on horse and cactus model. The best results are displayed in bold.

NG 1.1.	D <sub>mean</sub>					
Models	(k = 4)	(k = 10)	(k = 50)	(k = 100)		
Horse	1.399	1.172	0.638	0.790		
Cactus	0.574	0.512	0.953	1.633		

**Table 3.** Analysis of the results with different  $\lambda$  values on bunny model. The best results are displayed in bold.

M. 1.1.	D <sub>mean</sub>				
Models	$(\lambda = 0.01)$	$(\lambda = 0.1)$	$(\lambda = 0.9)$		
Bunny	0.587	0.476	1.111		

It can be clearly seen from Figure 2b,c that some noise points may not be removed when k is too small. Furthermore, if k is too large, as shown in Figure 2e, the point-cloud model shrinks obviously. It is clearly demonstrated in Table 2 that k = 50 provides the best error-metrics values for the horse model. However, it reveals the different views in Figure 3. The cactus model shrinks obviously when k is 50, and the desirable result appears when k is 10. It is also demonstrated in Table 2 that k = 10 provides the best error-metrics values for the cactus model. The reason for the phenomenon of the different k values is that the number of points contained in the model and the shape of the model are all different. Furthermore, the choice of k is also different under different noise levels.

The second key parameter is  $\lambda$ , which is a regularization parameter penalizing large  $a_i$ . It simply controls the filtering quality. As shown in Figure 4d, if the value of  $\lambda$  is too large, the filtered results of the bunny model are overly smoothing. Furthermore, if the value of  $\lambda$  is too small, the excessively large filtering coefficient  $a_i$  creates negative effects. It is demonstrated in Figure 4 and Table 3 that  $\lambda = 0.1$  provides a desirable result for the point-cloud model of bunny.

Therefore, the parameters of the point-cloud model (e.g., the number of points, the size of the model and shape features, etc.) should be fully considered in order to obtain the desired filtering results. Both of these two parameters need to be adjusted by users according to the experimental requirements and the structural characteristics of the point-cloud model to obtain the best filtering effect.

#### 4.2. Results and Comparison

After a comprehensive analysis of the effect of the parameters in our method, we proceeded to carry out experiments on the point-cloud model of the water-conveyance tunnel and simulated 3D models to compare our algorithm with some state-of-the-art

methods, with the aim of exhibiting its performance. In our experiments, we chose three different filtering algorithms, namely, bilateral filter [14], moving least-squares filter [15], and guided 3D point-cloud Filter [16]. Moreover, the results of these algorithms were reproduced on the same computer according to the optimal parameters provided by their authors.

#### 4.2.1. Scene Experiment

The purpose of developing the proposed algorithm is to filter the acoustic point-cloud model of water-conveyance tunnels. Therefore, we conducted an experiment on a real test site to verify the performance of our algorithm. The test site is a small water-conveyance tunnel with a diameter of 80 cm. The ground-truth shape of the tunnel is known. An underwater robot was used as the carrier to collect data [20]. The point-cloud model of the tunnel was corrected by the carrier-attitude information. The raw point-cloud model and the ground-truth model are shown in Figure 5; there are many noise points around the inner surface of the tunnel. In this experiment, we compared the results obtained by the proposed method with the results of the three different filtering algorithms mentioned above.



**Figure 5.** Raw point-cloud model of water-conveyance tunnel. (**a**) Raw point-cloud model; (**b**) front view of (**a**); (**c**) the ground-truth model of real water-conveyance tunnel; (**d**) front view of (**c**).

Compared with Figure 5a, it is demonstrated in Figure 6a,c,e,g that the four denoising algorithms all achieved the purpose of filtering. This can be more clearly observed in Figure 6b,d,f,h. For the small-scale noise points near the inner surface of the tunnel, it is clearly visible in Figure 6b that the guided 3D point-cloud filter achieved poor effects in smoothing the small-scale noise points. As shown in the blue circular dotted-line area in Figure 6d,f, the bilateral filter and the moving least-squares filter had the problem of over-smoothing while filtering the noise points. Figure 6h demonstrates that the proposed method can prevent over-smoothing while filtering out small-scale noise points. For large-scale noise far away from the inner surface of the tunnel, none of the three methods achieved a good result. By contrast, the results of the proposed method are better than other those of the other three algorithms.



**Figure 6.** The results of different algorithms tested on the point-cloud model of water-conveyance tunnel. (a) The result of guided 3D point-cloud filter; (b) front view of (a); (c) the moving least-squares filter results; (d) front view of (c); (e) the bilateral filter results; (f) front view of (e); (g) the results of proposed algorithm (k = 300 and  $\lambda = 0.5$ ); (h) front view of (c).

To quantitatively evaluate the denoising results, four error-metrics,  $D_{mean}$ ,  $\sigma$ ,  $r_{ave}$ , and  $r_{error}$ , were used in our experiment.  $D_{mean}$  represents the average distance between the points in the resulted point-cloud and the corresponding points in the ground-truth model [29].  $\sigma$  represents the variance of the distance.  $r_{ave}$  is the average radius of the filtered point-cloud model of the tunnel.  $r_{error}$  represents the reconstruction accuracy of the point-cloud model of the water-conveyance tunnel. The calculation results of the error metrics are shown in Table 4.

Method	D <sub>mean</sub> (cm)	$\sigma$ (cm)	r <sub>ave</sub> (cm)	r <sub>error</sub> (cm)
Guided 3D Point-Cloud Filter	4.73	1.77	41.48	1.48
Moving Least-Squares Filter	4.64	1.75	41.58	1.58
Bilateral Filter	4.79	1.78	40.67	0.67
Proposed algorithm	4.49	1.67	40.45	0.45

**Table 4.** Analysis of the results of different algorithms tested on the point-cloud model of waterconveyance tunnel (the best results are displayed in bold).

In Table 4, the best results are displayed in bold. It is clearly demonstrated that the results of our method were the best in  $D_{mean}$  and  $\sigma$ . In other words, the filtered points were close to the inner surface of the tunnel and had good aggregation properties. The results of the proposed method also have high accuracy in  $r_{ave}$  and  $r_{error}$ . Moreover, it is clearly seen from  $r_{error}$  that the point-cloud model of the real tunnel obtained by the proposed method has a high reconstruction accuracy, which provides important support for defects detection on the tunnel surface.

#### 4.2.2. Simulation Experiment

To verify the applicability of the proposed method, we also executed our algorithm on a simulated model of the water-conveyance tunnel and nine simulated 3D point-cloud models with different shapes and sizes to highlight the advantages. Affected by the noise of the electronic components of the acquisition equipment, the acquired point-cloud model is easily polluted by noise that obeys the Gaussian distribution. Therefore, the Gaussian noise generated using a zero-mean Gaussian function with a standard deviation was added to the simulated 3D point cloud models. The Gaussian noise ranges were from 1% to 3% of the diagonal length of the model bounding box.

As shown in Figure 7, the results of the guided 3D point-cloud filter, moving leastsquares filter, and bilateral filter are poor, and there are many noise points around the tunnel. The bilateral filtering results were relatively smooth, but the denoising effect was not perfect, and the tunnel outline experienced distortion. Comparing the inner part of the red rectangle in Figure 7b–e, the denoising effects of the model obtained by the proposed algorithm were much better than those obtained by the other three methods, and the denoising effect of the outliers was obvious. The filtered point-cloud model was closer to the real cylindrical container. The method proposed in this study has an obvious suppressive effect on the outliers while preserving the contour.

We also used the error metrics  $D_{mean}$ ,  $\sigma$ ,  $r_{ave}$ , and  $r_{error}$  to quantitatively evaluate the denoising results. As shown in Table 5, the metrics of the moving least-squares filter and bilateral filter were similar. Furthermore, the metrics of the guided 3D point-cloud filter were all poor. The four error metrics of the other three methods were all larger than those of the proposed algorithm. This means that the reconstruction accuracy of the point-cloud model obtained using these methods is poor. By contrast, our method is better than the other three methods in terms of the error metrics, which reflects the superiority of our algorithm.

Figures 8–10 are the filtering results of the unsharp-mask-guided filtering and several competing methods. Here, we only present pictures of the block, chair, and elephant models. Moreover, the results of these comparison algorithms were obtained according to the optimal parameters provided by their authors. It is clearly visible in Figure 8e that our algorithm can preserve the edge features well without shrinkage or deformation in the block model, while the results of the moving least-squares filter were deformed. For the 3D point-cloud model of the chair, it is shown in Figure 9b,d that the results of the guided 3D point-cloud filter and moving least-squares filter were poor. Moreover, the results of the bilateral filter were deformed. As shown in Figure 9e, the proposed method achieves more satisfactory results with the chair model. For the 3D point-cloud model of the elephant, it is shown in Figure 10 that the comparison methods all achieved satisfactorily smooth results, except for the blurring problem at the mouth of the elephant. It is clearly shown in

the enlarged map in Figure 10e that our method can preserve the local feature information of the elephant's mouth while smoothing the noise points. Furthermore, our method also achieved satisfactory results in six other, different, 3D point-cloud models, according to the values of the error metrics shown in Table 6.



**Figure 7.** The results of different algorithms tested on the point-cloud model of simulated waterconveyance tunnel. (**a**) The original point-cloud model of the simulated tunnel; (**b**) the guided 3D point-cloud filter results; (**c**) the bilateral filter results; (**d**) the moving least-squares filter results; (**e**) the results of proposed algorithm (k = 300 and  $\lambda = 0.5$ ).

**Table 5.** Analysis of the results of different algorithms tested on the point-cloud model of simulated water-conveyance tunnel (the best results are displayed in bold).

Method	D <sub>mean</sub> (mm)	$\sigma$ (mm)	r <sub>ave</sub> (mm)	r <sub>error</sub> (mm)
Guided 3D Point-Cloud Filter	16.53	10.74	520.71	40.71
Moving Least-Squares Filter	15.36	9.27	510.46	30.46
Bilateral Filter	15.35	9.17	510.43	30.43
Proposed algorithm	14.20	7.89	500.27	20.27

We also used the error metrics  $D_{mean}$  and  $\sigma$  to quantitatively evaluate the filtering results of the different methods. The error metrics for nine different models are shown in Table 6, and the best error-metrics values are highlighted in bold. According to these values, as expected, it is evident that the proposed method obtain better denoising results on most simulated 3D point-cloud models, with the exception of the elephant model. Although the best error-metric values were not obtained on the point cloud-model of the elephant, it can be seen in Figure 10e that our method achieved satisfactory results in maintaining the feature details of the elephant's mouth. The denoising effect of the bilateral filter in the point-based model was mainly reflected in the smooth effect of the point-cloud model. Furthermore, the error metrics  $D_{mean}$  and  $\sigma$  reflect the overall performance of the resulting model rather than the local detailed properties. Therefore, in our study, there was better error measurement and poor detail-feature retention, such as the elephant's mouth shape. This also reflects the compromise strategy adopted by the proposed method, which preserves the details as much as possible while maintaining the denoising effect. Overall,



the experimental results show that our method is effective on simulated 3D point models, and can ensure the features are more accurate while obtaining smoother filtering results.

**Figure 8.** The results of different algorithms tested on block model. (a) Block model with noise; (b) the guided 3D point-cloud filter results; (c) the bilateral filter results; (d) the moving least-squares filter results; (e) the results of proposed algorithm (k = 20 and  $\lambda = 0.5$ ); (f) the ground-truth model of block.



**Figure 9.** The results of different algorithms tested on chair model. (a) Chair model with noise; (b) the guided 3D point-cloud filter results; (c) the bilateral filter results; (d) the moving least-squares filter results; (e) the results of the proposed algorithm (k = 100 and  $\lambda = 0.1$ ); (f) the ground-truth model of chair.



**Figure 10.** The results of different algorithms tested on elephant model. (a) Elephant model with noise; (b) the guided 3D point-cloud filter results; (c) the bilateral filter results; (d) the moving least-squares filter results; (e) the results of proposed algorithm (k = 100 and  $\lambda = 0.3$ ).

Model	Error Metrics	Guided 3D Point-Cloud Filter	Bilateral Filter	Moving Least-Squares Filter	Our Method
Block	D <sub>mean</sub>	0.57	0.58	0.56	0.45
	$\sigma$	0.21	0.22	0.23	0.18
Carlas	$D_{mean}$	0.66	0.67	0.76	0.71
Cactus	$\sigma$	0.28	0.30	0.34	0.26
Chain	$D_{mean}$	2.55	2.10	2.69	2.23
Chair	$\sigma$	1.50	1.24	1.55	1.21
Electron	$D_{mean}$	0.57	0.49	0.63	0.72
Liephan	$\sigma$	0.31	0.28	0.41	0.36
Llinna	$D_{mean}$	0.78	0.34	0.37	0.34
прро	$\sigma$	0.34	0.22	0.29	0.22
	$D_{mean}$	0.81	0.94	0.74	0.58
Horse	$\sigma$	0.48	0.56	0.55	0.28
Monster	$D_{mean}$	0.78	0.80	0.92	0.64
	$\sigma$	0.34	0.34	0.49	0.26
Skull	$D_{mean}$	0.14	0.14	0.12	0.13
	$\sigma$	0.08	0.09	0.07	0.05
Bunny	$D_{mean}$	0.60	0.59	0.62	0.52
	$\sigma$	0.24	0.23	0.45	0.20

Table 6. Error metrics for different methods and the best results are displayed in bold.

The proposed method can achieve more satisfactory results on real site models and simulated models than many other competing methods. And the model obtained by our method is more suitable for the subsequent object recognition or 3D reconstruction tasks of water-conveyance tunnel.

#### 5. Conclusions and Discussion

To address the safety-monitoring problem of a water-conveyance tunnel during the operation stage, we proposed a novel filtering method, called unsharp-mask-guided filtering for 3D point cloud, to reduce the impact of noise on the point-cloud model of the water-conveyance tunnel. The proposed method fuses the ideas of classical guided filtering and the unsharp masking technique and extends them to the 3D point-cloud model by considering the position of the point. In addition, an edge-aware weighting idea was also used to retain the edge features of the point-cloud model while smoothing the noise points.

In the experiment, we examined the influence of two key parameters on the experimental results. We also carried out experiments on real-world 3D models of the tunnel and nine simulated 3D models to compare our algorithm with some state-of-the-art methods. The experimental results show that the proposed method can outperform several competing methods, both on the real-tunnel point-cloud model and simulated point-cloud models. The model provides greater accuracy for the further processing of point clouds. However, we did not consider the efficiency of our algorithm in the experiment. Because our method needs to search the neighborhood twice for each point and to judge the inclusion relationship between the neighborhoods of different points, it does not have an obvious advantage in efficiency. The running time of point-cloud models with different number of points is different. Taking the Stanford rabbit model as an example, the time cost of our algorithm on the hardware device based on this study is 7.2 s, while the time costs of bilateral filtering, MLS filtering, and guidance filtering are 4.5 s, 4.8 s, and 3.7 s, respectively. These time costs are all higher than those in the references because of the weak processing power of the computer equipment used in this study. Therefore, we need to optimize the calculation method to improve the efficiency of the proposed algorithm in future work.

**Author Contributions:** Conceptualization, J.W. and X.Z.; methodology, J.W.; software, J.W.; validation, J.W., X.Z. and Z.Z.; formal analysis, J.W.; investigation, J.W.; resources, X.Z.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, J.W.; visualization, J.W.; supervision, X.Z.; project administration, X.X.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by National Key R&D Program of China, grant number 2018YFC0407101, and National Natural Science Foundation of China, grant number 61671202.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

**Data Availability Statement:** The data presented in this study are available from the corresponding author upon request.

**Acknowledgments:** The authors would like to thank all the reviewers for their valuable comments. We would like to thank Georgia Tech for their Large Geometric Models. We are grateful to Princeton ModelNet project for sharing the ModelNet datasets. We also thank the researchers at Princeton, Stanford, and TTIC for sharing the ShapeNet datasets.

Conflicts of Interest: The authors declare no conflict of interest.

#### References

- Zhu, X.; Wang, T.; Liu, Y.; Huang, T. A new defect detection technology for long-distance water conference tunnel. *Water Resour. Hydropower Eng.* 2010, *41*, 78–81. [CrossRef]
- Liao, J.; Yue, Y.; Zhang, D.; Tu, W.; Cao, R.; Zou, Q.; Li, Q. Automatic Tunnel Crack Inspection Using an Efficient Mobile Imaging Module and a Lightweight CNN. *IEEE Trans. Intell. Transp. Syst.* 2022, 1–14. [CrossRef]

- 3. Jitao, L.; Gande, L. Study on long-term operation safety detection technical system of large and long diversion tunnel. *Water Resour. Hydropower Eng.* **2021**, *52*, 162–170. [CrossRef]
- 4. Jin, H.; Yuan, D.; Zhou, S.; Zhao, D. Short-Term and Long-Term Displacement of Surface and Shield Tunnel in Soft Soil: Field Observations and Numerical Modeling. *Appl. Sci.* 2022, *12*, 3564. [CrossRef]
- Wang, J.; Wang, C.; Han, Z.; Jiao, Y.; Zou, J. Study of hidden structure detection for tunnel surrounding rock with pulse reflection method. *Measurement* 2020, 159, 107791. [CrossRef]
- Park, J.; Kim, H.; Tai, Y.-W.; Brown, M.S.; Kweon, I. High quality depth map upsampling for 3D-TOF cameras. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 1623–1630.
- Zheng, W.; Xie, H.; Chen, Y.; Roh, J.; Shin, H. PIFNet: 3D Object Detection Using Joint Image and Point Cloud Features for Autonomous Driving. *Appl. Sci.* 2022, 12, 3686. [CrossRef]
- 8. Rangel, J.C.; Morell, V.; Cazorla, M.; Orts-Escolano, S.; García-Rodríguez, J. Object recognition in noisy RGB-D data using GNG. *Pattern Anal. Applic.* **2017**, *20*, 1061–1076. [CrossRef]
- 9. He, K.; Sun, J.; Tang, X. Guided Image Filtering. IEEE Trans. Pattern Anal. Mach. Intell. 2013, 35, 1397–1409. [CrossRef]
- 10. Ye, W.; Ma, K.-K. Blurriness-Guided Unsharp Masking. IEEE Trans. Image Processing 2018, 27, 4465–4477. [CrossRef]
- 11. Deng, G. A Generalized Unsharp Masking Algorithm. *IEEE Trans. Image Processing* 2011, 20, 1249–1261. [CrossRef]
- 12. Polesel, A.; Ramponi, G.; Mathews, V.J. Image enhancement via adaptive unsharp masking. *IEEE Trans. Image Process.* 2000, *9*, 505–510. [CrossRef] [PubMed]
- 13. Li, Z.; Zheng, J.; Zhu, Z.; Yao, W.; Wu, S. Weighted Guided Image Filtering. In *IEEE Transactions on Image Processing*; IEEE: Toulouse, France, 2015; Volume 24, pp. 120–129. [CrossRef]
- 14. Digne, J.; de Franchis, C. The Bilateral Filter for Point Clouds. Image Processing Line 2017, 7, 278–287. [CrossRef]
- 15. Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; Silva, C.T. Computing and rendering point set surfaces. *IEEE Trans. Vis. Comput. Graph.* **2003**, *9*, 3–15. [CrossRef]
- 16. Han, X.-F.; Jin, J.S.; Wang, M.-J.; Jiang, W. Guided 3D point cloud filtering. Multimed Tools Appl. 2018, 77, 17397–17411. [CrossRef]
- Feng, D.; Shi, B.; Xiushan, L.U.; Guoyu, L.I. A Multi-Beam Point Cloud Denoising Algorithm Considering Underwater Topographic Features. J. Geomat. Sci. Technol. 2017, 34, 364–369.
- 18. Xie, Q.; Tian, M.; Feng, C.; Yu, J.; Pang, Y. Study on Filtering Algorithm Combining Multibeam Point Cloud Intensity and Elevation Information. *Hydrogr. Surv. Charting* **2021**, *1*, 65–69. [CrossRef]
- 19. Cui, X.; Shen, W.; Shuai, C.; Hui, X. Preliminary research and application analysis of multi-beam point cloud filtering algorithm. *Hydrogr. Surv. Charting* **2021**, *5*, 12–16. [CrossRef]
- 20. Wang, J.; Zhang, X.; Xu, X.; Zhang, Z.; Song, K. Remote inspection method for water conveyance tunnel based on single-beam scanning sonar. *JARS* **2022**, *16*, 024522. [CrossRef]
- Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), Bombay, India, 7 January 1998; pp. 839–846.
- 22. Fleishman, S.; Drori, I.; Cohen-Or, D. Bilateral mesh denoising. ACM Trans. Graph. 2003, 22, 950–953. [CrossRef]
- Lee, K.-W.; Wang, W.-P. Feature-preserving mesh denoising via bilateral normal filtering. In Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05), Hong Kong, China, 7–10 December 2005.
- Jones, T.; Durand, F.; Desbrun, M. Non-Iterative, Feature-Preserving Mesh Smoothing. ACM Trans. Graph. 2003, 22, 943–949. [CrossRef]
- Manzo, M.; Rozza, A. DOPSIE: Deep-Order Proximity and Structural Information Embedding. Mach. Learn. Knowl. Extr. 2019, 1, 684–697. [CrossRef]
- Zaman, F.; Wong, Y.P.; Ng, B.Y. Density-Based Denoising of Point Cloud. In Lecture Notes in Electrical Engineering, 9th International Conference on Robotic, Vision, Signal Processing and Power Applications; Springer: Singapore, 2017; pp. 287–295. [CrossRef]
- Shi, Z.; Chen, Y.; Gavves, E.; Mettes, P.; Snoek, C.G.M. Unsharp Mask Guided Filtering. *IEEE Trans. Image Process.* 2021, 30, 7472–7485. [CrossRef] [PubMed]
- The Stanford 3D Scanning Repository. Available online: http://www.graphics.stanford.edu/data/3Dscanrep/ (accessed on 17 June 2022).
- 29. Han, X.-F. Research on Denoising Processing and Feature Description for 3D Point Cloud. Ph.D. Thesis, Tianjin University, Tianjin, China, 2019.