

Noise-robust transparent visualization of large-scale point clouds acquired by laser scanning



Tomomasa Uchida^a, Kyoko Hasegawa^a, Liang Li^a, Motoaki Adachi^b, Hiroshi Yamaguchi^c, Fadjar I. Thufail^d, Sugeng Riyanto^e, Atsushi Okamoto^f, Satoshi Tanaka^{a,*}

^a College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan

^b Shrewd Design Co. Ltd, Fushimi-ku, Kyoto, Kyoto 612-8362, Japan

^c Nara National Research Institute for Cultural Properties, Nijo-cho, Nara City 630-8577, Japan

^d Research Center for Area Studies, Indonesian Institute of Sciences (P2SDR-LIPI), Jakarta, Indonesia

^e Yogyakarta Archaeology Office, Yogyakarta, Indonesia

^f Institute of History, Otemae University, Nishinomiya, Hyogo 662-0965, Japan

ARTICLE INFO

Keywords:

Laser-scanned point cloud
Large-scale data
Visualization
High quality 3D see-through imaging
Stochastic noise transparentization

ABSTRACT

We propose a high-quality transparent visualization method suitable for large-scale laser-scanned point clouds. We call the method “stochastic point-based rendering (SPBR),” which is based on a novel stochastic algorithm. SPBR enables us to clearly observe the deep interior of laser-scanned 3D objects with the correct feeling of depth. The high quality of SPBR originates from the effect of “stochastic noise transparentization,” which is an effect to make the measurement noise transparent and invisible in the created images. We mathematically prove that this effect also makes the created transparent images coincide with the results of the conventional methods based on the alpha blending, which is time-consuming and impractical for large-scale laser-scanned point clouds. We also demonstrate the effectiveness of SPBR by applying it to modern buildings, cultural heritage objects, forests, and a factory. For all of the cases, the method works quite well, realizing clear and correct 3D see-through imaging of the laser-scanned objects.

1. Introduction

The recent rapid development of laser scanning has enabled us to precisely record various real 3D objects as large-scale point clouds digitally. The appearance of high-functionality and lightweight UAVs (unmanned aerial vehicles) has also made the scanning activity easier. In this situation, studies have been conducted to make good use of the acquired point clouds. For example, studies on data acquisition and registration, classification, segmentation and identification, and vegetation applications have been actively investigated (Li et al., 2017a). However, fewer efforts have been undertaken to develop comprehensible visualization that assists researchers and engineers to analyze and understand the acquired point clouds. Such visualization is vital because the large-scale digitally preserved point clouds usually describe very complex 3D shapes of real objects, e.g., cultural assets, landscapes, and equipment layouts in factories.

On the other hand, comprehensible visualization of large-scale complex 3D objects has been actively investigated in studies of visualizing big scientific data, primarily in the field of volumetric

visualization (Brodie and Wood, 2001). In the volumetric visualization task, an essential key to understanding complexity is “transparency,” which enables us to look deep inside of 3D objects, including internal human organs, 3D structures of complex machines, distributions of ocean temperature, and many others. We believe that transparency is also useful for analyzing and understanding laser-scanned data. The reason is that the data often record the inside of the scanned object as well as its external surfaces in detail. Such complex data are generated by measuring the inside and the outside of the target object separately and then merging both results to create a larger dataset that describes the whole object. Note that the complexity also tends to increase the data size, such that it can exceed tens or hundreds of millions of 3D points. Therefore, we contend that we should establish a precise and quick transparent visualization method for large-scale and complex point clouds acquired by laser scanning.

Recently, we proposed a novel method of high-quality and quick transparent visualization that is suitable for large-scale laser-scanned point clouds (Tanaka et al., 2016). In the reference, we issued a preliminary report that describes its basic algorithm and application to

* Corresponding author.

E-mail address: stanaka@is.ritsumei.ac.jp (S. Tanaka).



Fig. 1. Transparent visualization of a laser-scanned point cloud based on our proposed method (SPBR). We have used the laser-scanned point cloud of the campus building of Kyoto Women's University in Kyoto city, Japan. The number of points in the point cloud is 4.0×10^8 .

traditional festival floats of high cultural value. Below, we call the visualization method “stochastic point-based rendering (SPBR)” because it adopts the original stochastic algorithm to realize the transparency and uses the laser-scanned points directly as rendering primitives. The high quality of SPBR is realized by statistical effects that become more effective in proportion to the data size; therefore, SPBR is suitable for the recent large-scale data. Fig. 1 is an example of applying SPBR to laser-scanned data (4.0×10^8 points) of a university campus building. We can observe the 3D internal structure of the laser-scanned building as well as its outer shape.

The current paper is a full-report paper of SPBR. The new contributions added to the preliminary report (Tanaka et al., 2016) are as follows: (1) The idea of “stochastic noise transparentization,” which is an excellent noise-reduction effect, is proposed, and its theoretical analysis and experimental evaluation are presented; (2) A “simplified SPBR” is proposed that does not require upsampling of points and is therefore advantageous when treating very large-scale data; and (3) The usefulness of SPBR is demonstrated for varieties of laser-scanned data taken from both historical and modern scenes, with a particular emphasis on the noise reduction of raw laser-scanned data. In addition to the three above, we also present an updated description of the SPBR algorithm that is more sophisticated and easier for implementation.

For the visualizations demonstrated in this paper, the computations were executed on a Linux PC with an Intel Xeon CPU E5-2687 W (3.00 GHz, 512 GB of memory) CPU and an NVIDIA Quadro P5000 (16 GB of memory) GPU. We confirmed that this PC could handle $(several) \times 10^9$ 3D points (we also confirmed that 10^8 3D points could be rendered using a laptop PC with a 3.07 GHz Intel Core i7 processor, 8 GB of memory and an NVIDIA GeForce GT 480 M GPU).

The organization of this paper is as follows. In Section 3, we explain the refined algorithm of SPBR. We also propose the new “simplified SPBR,” suitable for large-scale data. In Section 4, we propose the idea of the “stochastic noise transparentization.” In Section 5, we demonstrate the usefulness of SPBR for varieties of laser-scanned data taken from

both historical and modern scenes. Section 6 is the concluding section that summarizes our achievements.

2. Related work

The most straightforward strategy for visualizing a remotely sensed (laser-scanned or photogrammetrically scanned) point cloud is point-based rendering (Gross and Pfister, 2007; Sainz and Pajarola, 2004; Kobbett and Botsch, 2004), in which the measured 3D points are directly used as the rendering primitives. For opaque visualization, the point-based rendering works quite well, and it has been used, e.g., for beautiful 3D imaging of cultural heritage objects (El-Hakim et al., 2005; Guidi et al., 2005; Ikeuchi et al., 2007; Yastikli, 2007; Laycock et al., 2008; Remondino et al., 2009; Koller et al., 2009; Dylla et al., 2009; Kersten et al., 2012; Aicardia et al., 2018). Recently, opaque point-based rendering based on remotely sensed massive point clouds has also been applied to virtual reality and web-based systems (Discher et al., 2018a; Thiel et al., 2018; Discher et al., 2018b).

The pioneering technique of point-based transparent visualization is EWA (elliptical weighted average) splatting (Zwicker et al., 2002). However, this technique requires sorting of a large number of 3D points along the line of sight, namely, the depth sort, to realize the correct feeling of depth. Therefore, this approach suffers from long computation times, so it is not practical for dealing with large-scale remotely sensed point clouds. Moreover, the depth-sorted result may become mathematically indefinite for a scanned object with a complex shape, and the indefiniteness may cause rendering artifacts, for example, wrong stripes. More direct ways to achieve the transparency, that is, using transparent points in a conventional point-based rendering technique such as the point sprites (Kessenich et al., 1996), is also possible (Seemann et al., 2018). However, this approach encounters the same problems that arise from the depth sort of a large number of points.

The point-based transparent rendering proposed by reference Zhang and Pajarola (2007) is also an excellent work, which aims to avoid the rendering artifacts by dividing the visualized point cloud into many layers. However, for a very dense point cloud, it is difficult to divide the points into proper layers. Moreover, for large-scale data, which are too voluminous to be cached in GPU memory, the frame rate tends to degrade.

3. Stochastic Point-Based Rendering (SPBR)

In this section, first, we present the algorithm of SPBR in a form more refined manner than its original proposal (Tanaka et al., 2016). For simplicity and without loss of generality, we fix the image resolution in the description. Second, in the final subsection, we propose a new simplified SPBR that is convenient for treating large-scale data.

3.1. Opacity formula

For transparently visualizing a laser-scanned surface that is recorded in the form of a point cloud by using SPBR, we control the surface opacity by adjusting point density. In this subsection, we derive a formula that relates the point density to the surface opacity.

We assume that the input point cloud has a uniform point density. Under this assumption, we aim to visualize the point-based surface transparently with uniform opacity. (We can assume the distributional uniformity for many of raw and artificially processed laser-scanned point clouds. In cases where the uniformity does not hold, however, the arguments below are still valid if we regard the user-defined opacity as the average surface opacity.) We use the input points directly as rendering primitives and assign the following properties to each point: (1) Each point occupies a cross-sectional area with size s on the visualized surface, and its projected image takes the size of exactly one pixel. (2) Each point is fully opaque such that we should incorporate the effect of

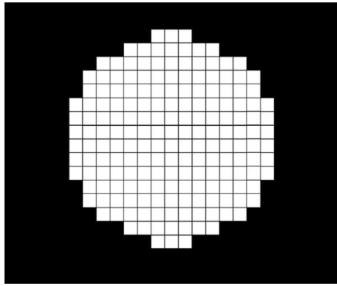


Fig. 2. Schematic picture of the 3D local surface segment (white region) with area S . There are M_p small squares that represent the pixels in the image of the segment. Each input 3D point has a cross-sectional area s , and its image overlaps one of the pixels in the image plane exactly.

the hidden-point removal during the point projection to the image plane. (3) Each point has the RGB color acquired in the laser scanning, usually from an interlocked camera. (4) Each point has a surface normal such that we can incorporate the shading effect in its rendering (this property is omitted).

Now, we consider a small 3D local surface segment with area S on the visualized point-based surface. We assume that this surface segment is approximately a flat plane and parallel to the 2D image plane. Let n be the number of 3D points that distribute uniformly in the segment, and let M_p be the number of pixels contained in the 2D image of the segment (see the small squares in Fig. 2). Every 3D point in the segment should be projected into one of the M_p pixels. On the other hand, each point occupies area s within the segment area S . Thus, the relation $M_p = S/s$ holds.

In SPBR, we realize the transparent feature of a visualized surface based on a probabilistic theory. That is, we define the surface opacity α by the probability that each pixel becomes the point color. Let us focus on an arbitrary pixel in the image of the abovementioned local surface segment; that is, we focus on one of the small squares in Fig. 2. For simplicity, we presume that every 3D point in the surface segment has the same color. If at least one point, which is colored and fully opaque as we have defined, is projected onto the focused pixel, the pixel becomes the point color. Otherwise, it remains the background color. Therefore, we can express α as the probability that at least one point is projected onto the focused pixel from the local surface segment.

Based on the above consideration, we can derive an opacity formula as follows. For an arbitrary pixel, we consider the number of projected points, x , which is an integer-valued random variable in the range between 0 and n . Because of the assumed distributional uniformity, the probability that each point in the local surface segment is projected onto this pixel is $1/M_p$ (see Fig. 2). Then, x obeys the probability of the binomial distribution $B(n, 1/M_p)$: $p(x) = (n!/x!(n-x)!)(1/M_p)^x(1-1/M_p)^{n-x} = (n!x!(n-x)!)(1/M_p)^x(1-s/S)^{n-x}$. Thus, we obtain the opacity formula that expresses the probability that at least one point is projected onto the focused pixel:

$$\alpha = 1 - p(0) = 1 - \left(1 - \frac{s}{S}\right)^n, \quad (1)$$

in which the parameters n , S , and s are all under our control. Below, we call the α determined by the right-hand side of formula (1) the “standard opacity,” which is the opacity for the surface portions parallel to the image plane. In the right-hand side of formula (1), the point cross-sectional area s is indirectly determined when we decide the camera setting and the image resolution. It is because s should become one-pixel size in the image plane. The local surface segment area S is determined when we set the segment area at an arbitrary position on the visualized surface for counting the number n there.

Let us consider a more general case such that the local surface segment is nonparallel to the image plane. Let the angle formed by the local surface segment with the image plane be θ ($0 \leq \theta < \pi/2$). Then,

the opacity formula (1) is revised as follows:

$$\alpha(\theta) = 1 - \left(1 - \frac{s}{Scos\theta}\right)^n, \quad (2)$$

given that the apparent segment area, which contributes to the image size of the segment, is reduced to $Scos\theta$ by multiplying the cosine factor. Formula (2) implies that the more the local surface segment becomes nonparallel to the image plane, the more the segment appears opaque. This automatic opacity change, which depends on the segment orientation, acts as a kind of shading effect. This effect improves the comprehensibility of the created images drastically. Usually, a raw laser-scanned point cloud does not have information on normal vectors of the constituent points, so we cannot incorporate the shading effect in the conventional opaque point-based rendering. However, SPBR can naturally realize the quasi-shading effect based on formula (2).

3.2. Steps of SPBR

Based on the opacity formula (1), the transparent visualization of a laser-scanned point-based surface is realized by following the three steps below:

STEP 1. Creation of point ensembles: Prepare multiple point sets, which we call “point ensembles,” by randomly dividing the given raw point cloud after appropriate point-density adjustment. Each point ensemble should be statistically independent and have the same point density corresponding to the user-defined standard opacity α according to formula (1). Below, we call the number of point ensembles the “ensemble number” and denote it by L . We will explain more details of STEP 1 in Section 3.3.

STEP 2. Point projection per ensemble: For each point ensemble, project its constituent 3D points to the image plane independently. As a result, L intermediate images are created. In the projection process, we incorporate the point occlusion effect per pixel so that we can reflect the opaque feature (see the second paragraph of Section 3.1) of the 3D points. (The point occlusion realizes the correct feeling of depth in the final transparent image.)

STEP 3. Averaging the intermediate images: Create an average image by averaging values of the corresponding pixels in the L intermediate images. The created average image becomes the final transparent image. The ensemble number L controls the statistical accuracy and therefore works as an image-quality parameter.

The above three steps realize transparent visualization of laser-scanned point clouds with high quality and correct feeling of depth. The free user parameters are the standard opacity α , the ensemble number L , and the point cross-sectional area s . Since we fix the image resolution for simplicity of our explanation, we can regard s as a constant number. Our experiments also show that the images created by following the above steps realize theoretically estimated opacities correctly with sufficiently small statistical errors (see Tanaka et al., 2016 for the experiments).

Fig. 1 is an example of applying SPBR to a laser-scanned point cloud of a campus building. Owing to the transparency, we can observe the detailed internal structure of the building as well as the surficial shape. Fig. 3 is an example of applying SPBR to a laser-scanned point cloud of a cityscape. We can see each building along the street through transparent electric poles and wires. Fig. 4 is an example of “fused” transparent visualization of two point clouds with different opacities. A traditional festival float of high cultural value is laser-scanned separately and merged with the point cloud of Fig. 3. The standard opacity of the festival float is made larger such that it looks clearer than the background cityscape.

3.3. Details of STEP 1 (Creation of Point Ensembles)

In this subsection, we describe the details of STEP 1 outlined in the previous section. For user-defined values of standard opacity α and



Fig. 3. Transparent visualization of a laser-scanned cityscape by using SPBR. The point cloud consists of 9.3×10^8 points. The ensemble number is $L = 500$.



Fig. 4. An example of transparent fused visualization of two different point clouds. The festival float “Hachiman-Yama” in the Gion Festival (Kyoto city, Japan), is laser-scanned separately and merged with the point cloud of Fig. 3. The standard opacities of the cityscape and the festival float are tuned to 0.3 and 0.5, respectively. The total number of the points after the point-density adjustment is 1.49×10^8 . The ensemble number is $L = 500$.

ensemble number L , we create the required L point ensembles from a given raw point cloud as follows:

1. Consider a small hypothetical “counting sphere” that is centered at one of the points in the raw point cloud. We define the surface portion included in the counting sphere as the local surface segment (see the white region of Fig. 2, for example). We approximate its

area S by the great circle area of the counting sphere.

- Selection of the centered point is arbitrary because of the assumed uniformity of the point distribution.
- In our implementation, we set the sphere radius to 1/50 of the diagonal length of the bounding box of the given raw point cloud.
- 2. Calculate the required number of points, n , in area S set in the above, corresponding to the user-defined value of α according to formula (1).
- 3. Count the actual number of points, N_{raw} , in the counting sphere.
 - For statistical accuracy, it is better to execute the counting for multiple counting spheres and adopt the average point number as N_{raw} .
 - In our implementation, we prepared 1000 spheres centered at points randomly selected from the raw point cloud.
- 4. Execute the point-density adjustment uniformly for the whole point cloud such that the counting sphere evolves to have nL points in it: perform downsampling if $N_{\text{raw}} > nL$ and upsampling if $N_{\text{raw}} < nL$.
 - The downsampling is executable by randomly removing points.
 - We can execute the upsampling by “copying” each of the original points with an appropriate probability. For example, if nL/N_{raw} is 2.5, we duplicate all the original points first; then, additionally, we duplicate each original point with probability 0.5. It is true that the added points are copies of the original points. However, they are treated as independent points after the point-cloud division below, if L is sufficiently large.
- 5. Randomly divide the density-adjusted point cloud such that we obtain the resultant L point ensembles, each of which consists of n points.

By executing STEP 2 and STEP 3 in the previous section following the above five detailed steps of STEP 1, we obtain a 3D transparent image of the laser-scanned point cloud with any user-defined standard opacity α .

3.4. Simplified SPBR without necessity of upsampling

By following the three steps described in Sections 3.2, supplemented by the prescription of Section 3.3, we can visualize a given laser-scanned point cloud transparently with any user-defined standard opacity α . The image quality is determined by the ensemble number L , which is the other user parameter. However, if we hope to achieve better image quality by increasing L while keeping the standard opacity α unchanged, we must execute upsampling to increase the number of points according to the prescription of substep 4 in Section 3.3. Increasing the number of points leads to longer rendering time and more massive memory consumption. We hope to avoid this inconvenience as much as possible when treating large-scale data.

Therefore, we propose “simplified SPBR,” which only divides the given point cloud into L point ensembles without executing upsampling/downsampling in STEP 1. In this simplified method, the only user parameter is L . As we increase L , i.e., the division number of the given point cloud, the point number in each ensemble, n , decreases in the right-hand side of the opacity formula (1), which leads to a decrease in α . This means that we can use L as the user parameter to control surface opacity indirectly. It is true that this strategy of parameter tuning is inconvenient for strict control of the α value. However, it is usually more essential for a user to create a comprehensible image quickly rather than realizing a particular α value exactly with effort.

Although the abovementioned opacity control via L is convenient for practical use, it has a drawback. As we increase L aiming at decreasing opacity, the image brightness is reduced. This brightness reduction often impairs the visibility of the created image. This drawback is, however, avoidable by increasing image brightness artificially as the postprocessing. We can determine the brightness amplification factor, which we denote p below, such that we recover the brightness of the image of $L = 1$, that is, the brightness of the opaque point rendering.

Now, we describe the algorithm of the abovementioned simplified SPBR concretely. We determine the value of p such that the ratio of the top 1% of bright pixels of the brightened image becomes the same as the opaque image created with $L = 1$. We assume a black background color, which means that the background color does not affect the brightness of the created image. The algorithm is as follows:

1. Execute SPBR using a user-defined L without the upsampling/downsampling in STEP 1 to create an original color image. Then, convert the created color image to a grayscale image, which we call the “original grayscale image” below.
 - The user-defined L determines surface opacity α according to formula (1) with $n = N_{\text{raw}}/L$.
2. Execute SPBR with $L = 1$ or, equivalently, with the opaque point rendering using the same camera parameters as 1. We convert the created color image into a grayscale image, which we call the “reference grayscale image” below.
3. In the reference grayscale image, obtain the maximal pixel value, B_{\max} , and the threshold pixel value, B_{th} , over which the number of pixels becomes more than 1% of all pixels.
4. Determine the value of brightness amplification factor p as follows:
 - (a) Update p repetitively based on the recurrence formula, $p = p + 0.01$, starting from the initial value $p = 1$ and incrementing the value by 0.01 every time.
 - (b) Every time p is updated, execute a trial of multiplying the current p to each RGB pixel value of the original color image, and convert the resultant color image to a grayscale image.
 - (c) Terminate updating p when over 1% of the pixels in the above brightened grayscale image have pixel values between B_{th} and B_{\max} .
5. For the original color image created in 1, multiply the red, green, and blue components of each pixel color by the p determined above. The resultant image becomes the finalized color image made brighter.

The above five steps of the simplified SPBR enable us to create bright and clear transparent images quite easily without the necessity of the upsampling of the original point cloud. The multiplication of p is made for every pixel uniformly, which means that we obtain a brighter image with user-defined opacity kept in effect. Typical values of p are less than 3. It usually takes less than a few seconds to find an appropriate value of p .

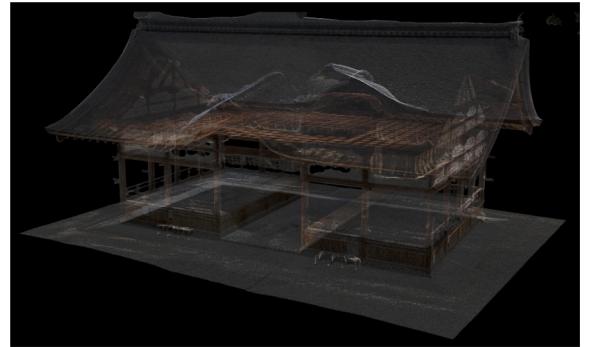
Fig. 5 shows an example of the brightness amplification. Fig. 5(a) and (b) show images before and after the brightness amplification, respectively. We can confirm that clearer visualization is realized in (b), keeping the same opacity as (a).

3.5. Rendering speed and other computational information

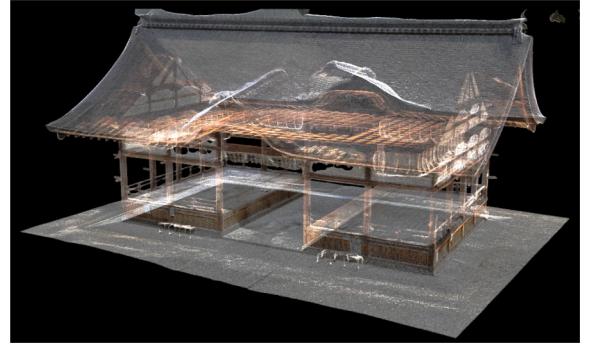
In Table 1, we summarize rendering speeds (fps) and other computational information for the creation of Figs. 1, 3, and 4. The computations were executed in the environment described in Section 1. In the table, the row of fps shows computational speeds in the rendering stage (STEP 2 and STEP 3). We can see that the rendering is executable with interactive frame rates for real laser-scanned data, after the pre-processing stage (STEP 1) that takes several tens of seconds. Note that the pre-processing stage should be executed only once, and the resultant data can be reused in the rendering stage. Besides, we can skip the pre-processing stage in the simplified SPBR proposed in Section 3.4.

4. Stochastic noise transparentization

In this section, we explain the idea of “stochastic noise transparentization,” which is an automatic noise reduction effect inherent to SPBR. We also perform a theoretical analysis and experimental evaluation of the effect. Although the effect well explains the high quality



(a) the original color image before the brightness amplification
($L = 100, \alpha = 0.1$)



(b) the obtained color image after the brightness amplification
($p = 2.52$)

Fig. 5. Comparison of the images before and after the brightness amplification. The laser-scanned point cloud (16 million points) of the worship hall of Fujinomori Shinto Shrine (Kyoto city, Japan) is used for the visualizations. The ensemble number L is set to 100. The automatically determined value of p is 2.52. It took 1.48 seconds to find the value of p .

Table 1
Detailed computational data for the creation of Figs. 1, 3, and Fig. 4.

image	Fig. 1	Fig. 3	Fig. 4
α	0.3	0.3	cityscape: 0.3 festival float: 0.5
L	150	500	500
image resolution	1920 × 1080	1024 ²	1024 ²
original number of points	2.61×10^8	9.44×10^8	7.89×10^6
number of points	7.33×10^8	9.44×10^8	1.49×10^8
after resampling	(upsampling)	(no resampling)	(upsampling)
time for resampling	85.77 [sec]	—	15.44 [sec]
fps	3.61	1.01	0.92

of the images created by SPBR, it has not been investigated explicitly in our previous work (Tanaka et al., 2016; Li et al., 2017b). We also consider how to utilize this effect to the maximum in visualizing laser-scanned data.

4.1. Noise included in laser-scanned data

Here, we describe a more precise definition of the term “noise” used in this paper. We consider two types of noise that a laser-scanned point cloud commonly contains. We call the first type “random noise” and the second type “outlier noise”. Random noise is the local fluctuation of points around their correct positions. In a visualized image, random noise makes the object look unsharp. Outlier noise is a point or a point cluster that exists apart from the scanned surface at a disconnected position. In a visualized image, outlier noise appears as dots apart from

the scanned object.

Usually, we perform the processing of a raw laser-scanned point cloud for denoising before using it in practice. To eliminate the random noise, we often perform the processing called “voxelization” (Blas et al., 2009; Papon et al., 2013; Rusu, 2013). That is, we divide the whole space into small cubes, replace all the points in each cube with an artificially added point at the cubic center and assign the average color of the neighboring points to the added point. This voxelization can create beautiful images by smoothing the random noise. The major problem of the voxelization technique is, however, deterioration of the measurement precision to the cubic width at worst. Additionally, it is often unpreferable that the original points are removed and replaced with the artificially added points. For example, we should preserve the raw measurement data as much as possible in the digital archiving of cultural heritage objects.

Regarding outlier noise, denoising usually means the elimination of outlier 3D points, i.e., points existing at positions disconnected from the scanned surface, from point cloud data. Although there are methods to execute this task automatically (Slob et al., 2005; Lichten, 2007; Rusu, 2013), they are not suitable for all situations. For example, the methods often fail if the outlier 3D points exist near the scanned surface and disconnectivity of the points from the surface becomes vague. In such a case, we need time-consuming manual work.

The idea of stochastic noise transparentization is to make the noise “invisible” by making the noise transparent rather than eliminating it. In other words, we utilize the transparency-realizing capability of SPBR for making the noise invisible in the created images. By this strategy, we can keep all the original points of the measurement data and maintain the initial measurement accuracy. Note that the transparency-realizing capability of SPBR works better for large-scale data because it is based on the stochastic theory: we can make better use of the redundancy of large-scale data for improving the statistical accuracy. Below, we explain the transparentization of the above two types of noise one-by-one. We conduct both the theoretical arguments and experimental analyses. The image resolution is set to 1024^2 pixels in the experiments.

4.2. Transparentization of random noise

4.2.1. Theoretical arguments

Regarding random noise, stochastic noise transparentization is realized when making the average image of the L intermediate images in STEP 3 of SPBR. Note that the intermediate images are averaged per pixel, so the image resolution is not deteriorated at all. Since SPBR does not require 3D voxelization as well, we can make the best use of both the image resolution and the measurement accuracy.

Let us focus on an arbitrary pixel in the image plane. For simplicity, we assume that points are projected from a single surface to this pixel. The random noise makes points that should be projected to the neighboring pixels project to the focused pixel incorrectly. This error in the projection makes pixel values of the intermediate images fluctuate at the focused pixel.

For the focused pixel, the pixel value B of the final transparent image is calculated as the average of the pixel values of the corresponding pixels, B_1, B_2, \dots, B_L , in the L intermediate images:

$$B = \frac{1}{L}(B_1 + B_2 + \dots + B_L), \quad (3)$$

for each of the red, green, and blue color components. Let M be the number of intermediate images to which point colors are assigned to the focused pixel. Then, the number of the remaining intermediate images in which the background color is assigned to the focused pixel becomes $L - M$. Therefore, after proper renumbering of index i of B_i such that B_1, B_2, \dots, B_M become the point color, formula (3) is rewritten as

$$B = \frac{M}{L} \left\{ \frac{1}{M}(B_1 + B_2 + \dots + B_M) \right\} + \frac{L - M}{L} C_{bg}, \quad (4)$$

where C_{bg} is the user-defined background color that is a constant. On the right-hand side of (4), randomness is included only in the first term, whereas the second term is constant.

We further rewrite the right-hand side of expression (4) by using the central limit theorem of statistics. Let the statistical variance of each B_i ($i = 1, 2, \dots, M$) be σ_{raw}^2 . For a large M , which is usually realized if we choose sufficiently large L , the partial average $(B_1 + B_2 + \dots + B_M)/M$ obeys the Gaussian distribution $N(C_{pt}, \sigma_{raw}^2/M)$, where C_{pt} is the mean value of the distribution. It is reasonable to regard this C_{pt} as a pixel value that reflects the denoised point color. For a sufficiently large M , the variance, σ_{raw}^2/M , is very small. Therefore, we can approximately replace the above partial average with its mean value C_{pt} . Then, the expression (4) becomes:

$$B \simeq \frac{M}{L} C_{pt} + \frac{L - M}{L} C_{bg}. \quad (5)$$

Here, we recall that the standard opacity α is defined as the probability that a pixel becomes a point color rather than the background color. Since M is the number of the intermediate images where the focused pixel becomes the point color and L is the total number of the intermediate images, the relationship between L and α can be expressed as follows:

$$\frac{M}{L} \simeq \alpha. \quad (6)$$

Substituting (6) into (5), we finally obtain

$$B \simeq \alpha C_{pt} + (1 - \alpha) C_{bg}. \quad (7)$$

Formula (7) is nothing other than the ordinary alpha blending formula (Porter and Duff, 1984), where C_{pt} is the surface color and C_{bg} is the background color. Namely, B becomes the correct pixel value of a transparent image of a surface with color C_{pt} and opacity α when adopting background color C_{bg} . Thus, we have proved that the transparent image created by SPBR converges to the right transparent image for sufficiently large L and M .

Finally, we investigate appropriate values of L theoretically. According to the argument in the above, the variance of random noise is estimated as σ_{raw}^2/M . Let us require that the standard deviation of random noise is smaller than β in units of pixels. Since $L \geq M$, this requirement can be expressed as

$$\beta^2 > \frac{\sigma_{raw}^2}{M} \geq \frac{\sigma_{raw}^2}{L}. \quad (8)$$

Rewriting this inequality as a requirement for L , we obtain

$$L > \frac{\sigma_{raw}^2}{\beta^2}. \quad (9)$$

For example, in case that $\sigma_{raw}^2 = 10$, requiring the standard deviation smaller than the pixel width ($\beta = 1$) means that L should be larger than 10. Similarly, $\beta = 1/3$ means that L should be larger than 90.

4.2.2. Validation experiments

Here, we show our experiments for validating the above theoretical arguments on the random noise.

Our first experiment uses an artificial square-shaped point cloud (side = 1 in the absolute dimension) that contains 1.0×10^7 uniformly distributed random points on the square rectangle. The color of the points was set to white (gray level is 255). Gaussian noise was added by randomly selecting 10% of the points and shifting their positions according to a three-dimensional Gaussian distribution that has a mean of zero and a variance of 1.0×10^{-5} . This variance corresponds to $\sigma_{raw}^2 = 10$ in units of pixel 2 . The color of the modified points was set to red (gray level is 76). Therefore, the modified point cloud contains 9.0×10^6 white non-noisy points and 1.0×10^6 red noisy points.

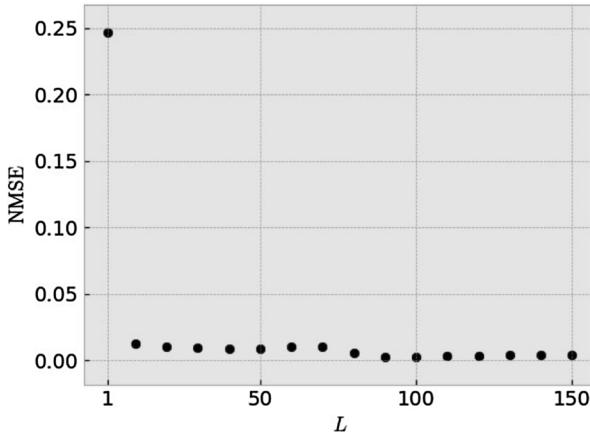


Fig. 6. NMSE between the original and noisy images with ensemble number $L = 1$ to 150. The original image is created from a point cloud of 10 million uniformly distributed random points on a square rectangle. The noisy image is created by adding Gaussian random noise to the original point cloud.

Hereafter, we use the term “original image” for the image generated from the artificial point cloud without noise, whereas “noisy image” is used for the image generated from the modified point cloud with Gaussian noise. To evaluate the reduction of the random noise in the created images quantitatively, we adopt the normalized mean square error (NMSE). The NMSE shows the similarity between the standard image and the target image. Given a $l \times m$ target image K and its standard image I , the NMSE is defined as

$$\text{NMSE} = \frac{\sum_{i=0}^{l-1} \sum_{j=0}^{m-1} \{I(i, j) - K(i, j)\}^2}{\sum_{i=0}^{l-1} \sum_{j=0}^{m-1} I(i, j)^2}. \quad (10)$$

In general, a smaller NMSE value indicates that the target image is more similar to the standard image. In our experiment, the standard image I in (10) is the original image, whereas the target image K is the noisy image.

Fig. 6 shows the change in NMSE of the two images with ensemble number $L = 1$ to 150. As shown in the figure, the NMSE is at a relatively high value of 0.247 when $L = 1$. However, the NMSE drastically decreases when L is approaching 10. After that, the NMSE tends to decrease gradually. This tendency is consistent with the theoretical argument for the case of $\sigma_{\text{raw}}^2 = 10$ described at the end of Section 4.2.1. The decreasing NMSE indicates that the noisy image is increasingly more similar to the original image as the ensemble number L increases. In other words, the noise reduction effect works as expected. The evaluation result also shows that a satisfactory result of noise reduction can be obtained even at a low ensemble number.

Let us go on to the second experiment. This time, we use the laser-scanned point cloud of a festival float with high cultural value (see Fig. 7). The image of Fig. 7 is created by using SPBR with $L = 1$, which is equivalent to the conventional opaque point-based rendering. As in the first experiment, a noisy point cloud is created by adding Gaussian noise to the point cloud of the festival float in Fig. 7: we randomly select 10% of the points in the original point cloud and shift their positions according to a three-dimensional Gaussian distribution with a mean of zero and a variance of 0.1. The variance is determined such that the standard deviation becomes approximately 1% of the bounding box length of the festival float.

Fig. 8 shows transparent visualization of the created noisy point cloud by using simplified SPBR for various values of L . We have used simplified SPBR because we can use the same point cloud for all the L 's without the necessity of upsampling or downsampling, which is appropriate for comparing the images. Fig. 8(a) for $L = 1$ shows the conventional opaque point-based rendering. We can observe the random noise that makes the visualized object fuzzy in its entirety.



Fig. 7. A laser-scanned point cloud of the festival float “Fune-Hoko” of the Gion Festival in Kyoto city, Japan. The number of points is 25,427,464. The bounding box size of the point cloud is $7.5 \times 3.4 \times 6.7$ in units of meters.

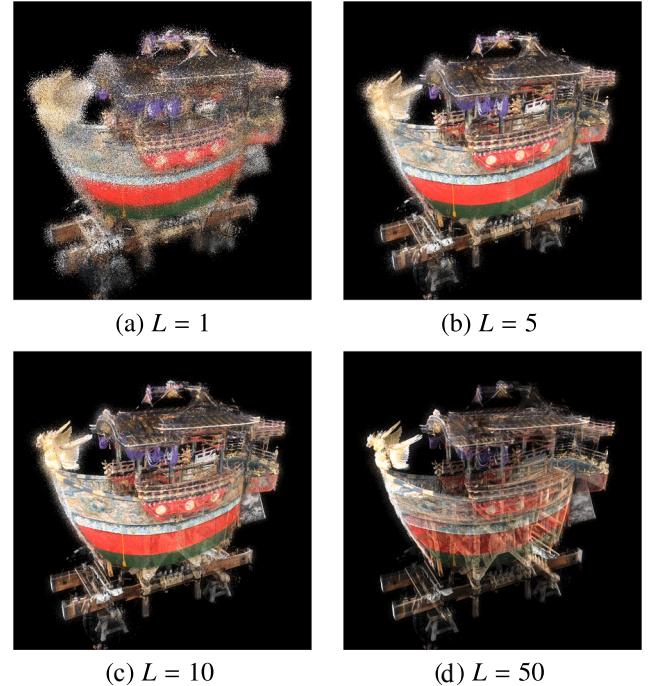


Fig. 8. Stochastic noise transparentization of the random noise for various values of L . We created a noisy point cloud by selecting 10% of the points in the point cloud of Fig. 7 and shifting their positions according to a three-dimensional Gaussian distribution with a mean of zero and a variance of 0.1.

However, as L increases, the fuzziness disappears rapidly (see Figs. 8(b), (c), and (d)). In general, the speed with which the noise becomes invisible is data dependent. However, generally, we can tell that the random noise becomes almost invisible at approximately $L \simeq 10$ and completely invisible at approximately $L \simeq 50$ or 100.

We also carried out a similar experiment by using the point cloud of Fig. 7. This time, random noise is added to modify point colors rather than point positions. Gaussian noise with zero mean and variance 100 is added in the 256-tone color space for 10 percent of points. We obtained an experimental result similar to Fig. 8. We can see that the random noise almost disappears at approximately $L \simeq 10$ and completely disappears at approximately $L \simeq 50$ or 100.

4.3. Transparentization of outlier noise

4.3.1. Theoretical arguments

Regarding the outlier noise, stochastic noise transparentization is realized based on the noise feature, that is, the density of the outlier 3D points is much lower than the point density on the actual laser-scanned surface.

Let us estimate the opacity of the outlier noise based on opacity formula (1). We consider a hypothetical local surface segment with area S , around which n outlier 3D points exist. We assume the segment is parallel to the 2D image plane. We hypothetically project these n points onto the local surface segment and regard the resultant surface opacity as the opacity of the outlier noise. Since s/S should be small because s is the 3D area corresponding to one pixel on the image plane, formula (1) is approximated as follows:

$$\alpha = \frac{sn}{S}. \quad (11)$$

Formula (11) means that the opacity of the outlier noise is proportional to the outlier 3D point density, which is usually much smaller than the point density on the actual laser-scanned surface. Therefore, the outlier noise becomes transparent and invisible.

We can also explicitly show that the pixel value B of the final transparent image is consistent with the alpha blending formula (7). The abovementioned low density of outlier 3D points indicates that M is sufficiently smaller than L , and therefore formula (5) can be approximated as

$$B \approx C_{\text{bg}}, \quad (12)$$

which is the special case of the alpha blending formula (7) when $\alpha \approx 0$.

4.3.2. Validation experiments

Here, we show our experiments for validating the above theoretical arguments on the outlier noise. We use the point cloud shown in Fig. 7 again, this time adding the outlier noise artificially.

For the validation experiment, a noisy point cloud is created by adding uniform random noise to the point cloud of the festival float in Fig. 7: we select 10% of the points in the original point cloud and move them to random positions within the bounding box of the festival float.

Fig. 9 shows transparent visualization of the created noisy point cloud by using simplified SPBR for various values of L . Fig. 9(a) for $L = 1$ shows the standard opaque point rendering. We can observe many dots throughout the scene. However, as L increases, the dots disappear rapidly (see Figs. 9(b), (c), and (d)). It is because the opacity of the noise regions converges to the theoretical values of formula (11) as L increases, and then the lowness of the opacity becomes evident in comparison with the regions of the actual laser-scanned surface. According to our experience, for most of the real laser-scanned point clouds, the outlier noise becomes invisible for L larger than 10 such as in the case of the random noise.

Fig. 10 shows the result of another validation experiment of transparentizing outlier noise. We selected pixels, where correctly laser-scanned points are not projected and only outlier 3D points are projected, in the noisy image of Fig. 9(a). Then, we calculated the average of M/L for the selected pixels. Fig. 10 shows change in the average M/L with ensemble number $L = 1$ to 150. We can see that M/L drastically decreases when L is approaching 10, and then, decreases gradually. As theoretically explained in Section 4.3.1, the decreasing M/L indicates that the pixel colors approach the background color and that outlier 3D points become invisible.

5. Case studies

In this section, we present three case studies that demonstrate the effectiveness of SPBR in various scenes of laser scanning. More concretely, we take three cases, in which the laser scanning often

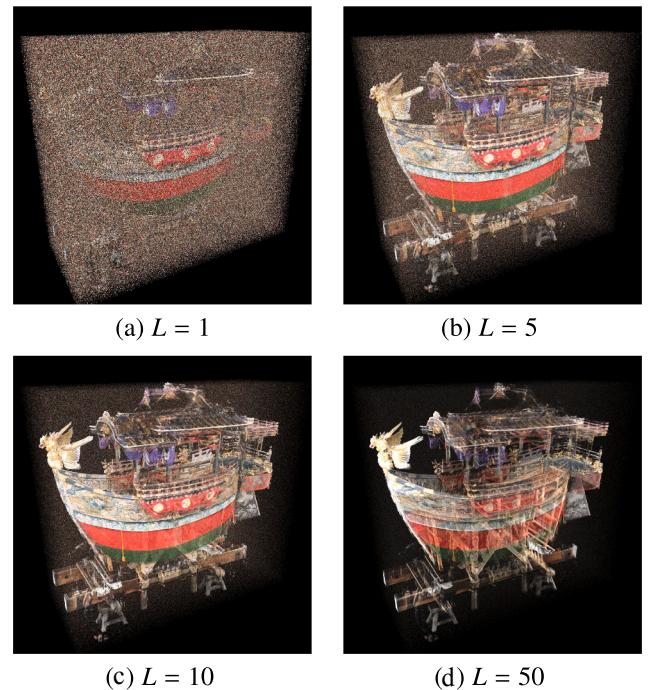


Fig. 9. Stochastic noise transparentization of the outlier noise for various values of L . We created a noisy point cloud by selecting 10% of the points in the point cloud of Fig. 7 and moving them to random positions within the bounding box of the festival float.

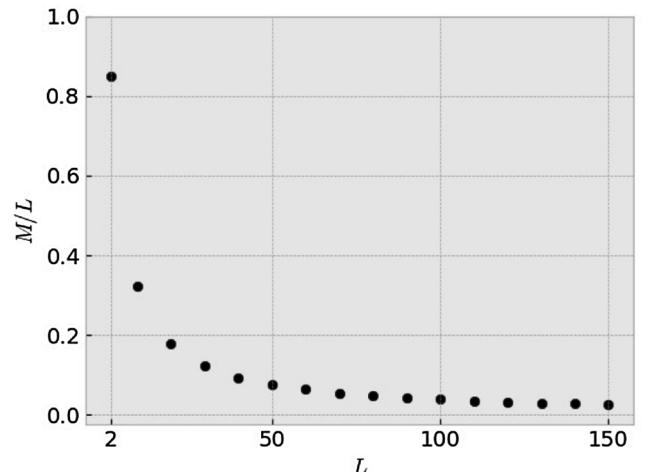


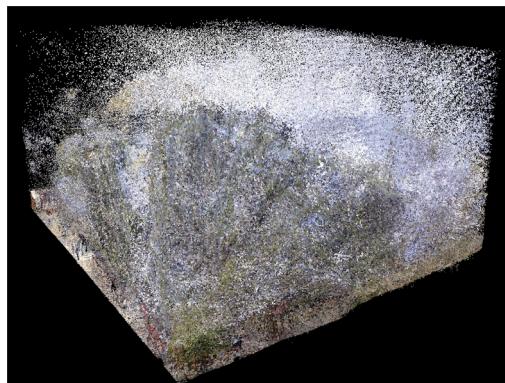
Fig. 10. Change in the average M/L with ensemble number $L = 1$ to 150. The average is calculated for pixels, to which only outlier points are projected, in the noisy image of Fig. 9(a).

encounters measurement noise. We show that SPBR can produce high-quality 3D transparent images with minimal adverse effects of the noise. The presented images were created by simplified SPBR (see Section 3.4) without upsampling or downsampling. The image resolution is set to 1024² pixels in the experiments.

5.1. Scene with trees

It is beneficial to achieve high-quality visualization of scenes with many trees based on laser-scanned data. Such visualization should become a useful tool to study cultural properties and other 3D objects that exist inside forests. High-quality visualization of scenes with trees is also in high demand in the study of forestry.

However, trees often cause measurement noise. In laser scanning,



(a) conventional point-based rendering

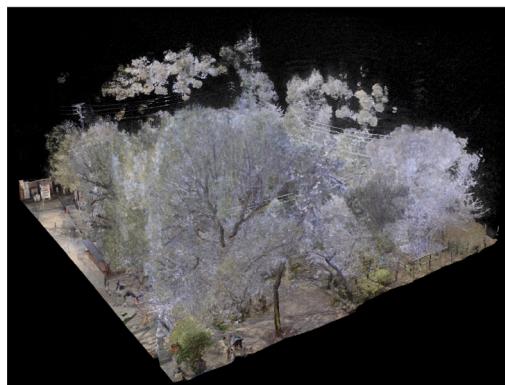
(b) SPBR with $L = 100$

Fig. 11. Visualizations of a laser-scanned forest using (a) the conventional point-based rendering (SPBR with $L = 1$) and (b) SPBR with $L = 100$. The visualized point cloud has 1,165,757,917 points. The simplified SPBR is used for the visualization with $p = 1.0$.

trees with leaves quivering in the wind make the acquired positional information inaccurate, which causes random noise. Moreover, the glitter of tree leaves makes the laser light scatter, which generates many outlier 3D points at positions disconnected from the scanned object.

Fig. 11 shows visualizations of the forest of Fujinomori Shinto Shrine (Kyoto city, Japan). Fig. 11(a) is created by the conventional point-based rendering (SPBR with $L = 1$), whereas Fig. 11(b) is created by SPBR with $L = 100$. In Fig. 11(a), we can see that the trees of the forest partially hide the gate. However, in Fig. 11(b) created by SPBR with $L = 100$, the gate is more visible seen through the transparent trees. Laser-scanned cultural heritage objects are often in a forest. In such cases, the high-quality transparent visualization, in which the measurement noise is significantly reduced, should be quite useful.

Fig. 12 shows the similar images of visualizing a Shinto shrine gate “torii” in the same forest. In Fig. 12(a), which is created by the conventional point-based rendering (SPBR with $L = 1$), we can see that the trees of the forest partially hide the gate. However, in Fig. 12(b) created by SPBR with $L = 100$, the gate is more visible seen through the transparent trees. Laser-scanned cultural heritage objects are often in a forest. In such cases, the high-quality transparent visualization, in which the measurement noise is significantly reduced, should be quite useful.

5.2. Factory with powder dust

Factories are frequent targets of laser scanning because the scanning can accurately record their current situation, even if it is not depicted in the design drawing. The recorded data are usable for plant simulations, for example, in virtual trial-and-error simulations to find the best



(a) conventional point-based rendering

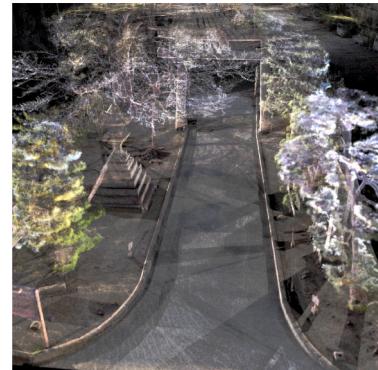
(b) SPBR with $L = 100$

Fig. 12. Visualizations of a laser-scanned Shinto shrine gate “torii” (the structural object in the upper half of the image), which is partially hidden by the trees of the forest. Each image is created by using (a) the conventional point-based rendering (SPBR with $L = 1$) and (b) SPBR with $L = 100$. The visualized point cloud has 588,290,603 points. The simplified SPBR is used for the visualization with $p = 2.06$.

arrangement of factory equipment. High-quality visualization is crucial when setting up the simulation and when analyzing the simulation results.

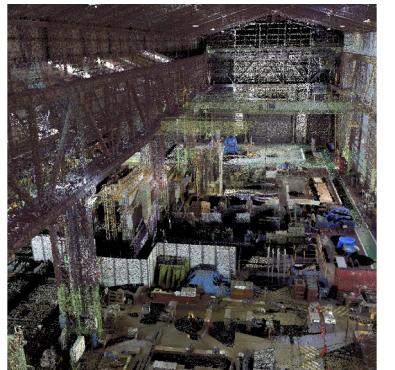
However, there is a problem in the laser scanning of factories. It is the existence of powder dust in the air, which scatters laser light. This scattering causes both random noise and outlier noise, making the visualization unclear. Stochastic noise transparentization works quite well to erase the unclear aspects.

Fig. 13 shows the visualization of a laser-scanned factory. The image of Fig. 13(a), which is created by the conventional point-based rendering (SPBR with $L = 1$), looks blurred because of the noise emanating from the powder dust. On the other hand, in the image of Fig. 13(b), which is created by SPBR with $L = 100$, the noise disappears satisfactorily. Fig. 14 shows similar comparative visualizations of a part of the same factory. In the denoised image of Fig. 14(b), we can read the clear-cut Japanese word meaning “iron powder” on the walls more clearly.

5.3. Scene with light reflection on glass windows

In the laser scanning of modern buildings, the reflection of laser light on glass windows often causes measurement noise, especially outlier noise. Even in such a case, stochastic noise transparentization works well.

Fig. 15 shows the visualization of laser-scanned campus buildings of the same university shown in Fig. 1. The scanned buildings have many glass windows, and the glass reflects the laser light. This reflection



(a) conventional point-based rendering

(b) SPBR with $L = 100$

Fig. 13. Visualizations of a laser-scanned scene of a factory using (a) the conventional point-based rendering (SPBR with $L = 1$) and (b) SPBR with $L = 100$. The visualized point cloud has 691,706,402 points. The simplified SPBR is used for the visualization with $p = 1.78$.

causes outlier noise. In Fig. 15(a), which is created by the conventional point-based rendering (SPBR with $L = 1$), we can see many outlier 3D points apart from the buildings (see the area indicated by the ovals). On the other hand, in Fig. 15(b), which is created by SPBR with $L = 100$, the outlier 3D points disappear satisfactorily.

6. Conclusions

In this paper, we have proposed stochastic point-based rendering (SPBR), which works well for large-scale laser-scanned point clouds and realizes high-quality transparent visualization of the recorded 3D object. SPBR also achieves interactive rendering speeds that are sufficient for practical use.

The high quality of SPBR derives from the effect of stochastic noise transparentization, which is included in the probabilistic algorithm of SPBR, in particular, the pixel-oriented ensemble averaging. We have conducted a detailed theoretical investigation regarding this noise reduction effect, and we mathematically proved that the created images converge to those of ideal alpha blending if the ensemble number is sufficiently large.

We have also demonstrated the effectiveness of our proposed visualization by applying it to modern buildings, cultural heritage objects, forests, and a factory. For all the cases, our visualization works quite well, realizing precise and intuitive transparent (see-through) 3D imaging of the laser-scanned objects with the measurement noise made invisible.

The limitation of the proposed visualization method is the rendering speed to execute STEP 2 and STEP 3. Our GPU-based implementation has achieved interactive frame rates for real laser-scanned data. However, we still hope to accelerate the speed such that real-time frame



(a) conventional point-based rendering

(b) SPBR with $L = 100$

Fig. 14. Comparative visualizations similar to Fig. 13 for a part of the same factory. A Japanese word *kiriko* meaning “iron powder” is written on the red walls with white characters. The simplified SPBR is used for the visualization with $p = 1.28$.

rates can be realized. This goal will be achieved by introducing more explicit parallel computing because STEP2 is executable in parallel for each ensemble (STEP 3 takes a much shorter time than STEP 2). Use of a multi-thread rendering environment is promising. We also plan to use a supercomputer for the parallel processing of very large-scale laser-scanned data.

It should be mentioned that the improvement of image quality by the effect of transparency does not apply to all situations. For example, very fine 3D structures or detailed decorations made on the surface of a cultural heritage object may look unclear when transparency is introduced. The transparent visualization is not suitable for analysis of small physical/chemical damages, such as walls containing fissures, moisture stains due to leaks, and physical changes due to wear, either. However, these drawbacks of the transparent visualization can be partially solved by properly highlighting high-curvature parts and other feature regions using principal component analysis (Weinmann et al., 2013).

In this paper, we have focused on laser-scanned data. However, the proposed visualization method is also applicable to point clouds acquired by the photogrammetric scanning without any modification. Note that the photogrammetric scanning is also burdened with similar measurement noises. For example, the reflection of the sunlight on glass windows and the quivering of tree leaves in the wind causes random noise or outlier noise. Therefore, our method is useful to both the laser-scanned data and the photogrammetrically scanned data.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.



(a) conventional point-based rendering

(b) SPBR with $L = 100$

Fig. 15. Visualizations of laser-scanned campus buildings with many glass windows by using (a) the conventional point-based rendering (SPBR with $L = 1$) and (b) SPBR with $L = 100$. The laser-scanned data include outlier noise in the area indicated by the ovals, but the noise is invisible in the image of (b). The visualized point cloud has 1,942,127,651 points. The simplified SPBR is used for the visualization with $p = 1.42$.

Acknowledgements

Images of the university campus buildings, the Hachiman-Yama float, the Shinto-shrine buildings and the forest, the Fune-Hoko float, and the factory are presented with the permission of Kyoto Women's University, the Hachiman-Yama Preservation Society, Fujinomori Shrine, the Fune-Hoko Preservation Society, and the factory manager, respectively. We thank these associations and related persons for their cooperation in our laser-scanning activities. We thank Prof. Y. Kitao and Prof. K. Koyamada for their valuable advice. This research is supported by JSPS KAKENHI Grant No. 16H02826.

References

- Aicardia, I., Chiabrandob, F., Lingua, A., Noardo, F., 2018. Recent trends in cultural heritage 3d survey: the photogrammetric computer vision approach. *J. Cult. Heritage* 32, 257–266.
- Brodie, K., Wood, J., 2001. Recent advances in volume visualization. *Comput. Graph. Forum* 20, 125–148.
- Discher, S., Masopust, L., Schulz, S., Richter, R., Döllner, J., 2018a. A point-based and image-based multi-pass rendering technique for visualizing massive 3d point clouds in VR environments. *J. WSCG* 26, 76–84.
- Discher, S., Richter, R., Döllner, J., 2018b. A scalable webGL-based approach for visualizing massive 3D point clouds using semantics-dependent rendering techniques. In: Proceedings of the 23rd International ACM Conference on 3D Web Technology, Web3D 2018, Poznań, Poland, June 20–22, 2018, pp. 19:1–19:9.
- Dylla, K., Frischer, B., Mueller, P., Ulmer, A., Haegler, S., 2009. Rome reborn 2.0: a case study of virtual city reconstruction using procedural modeling techniques. In: Proc. CAA, pp. 62–66.
- El-Hakim, S.F., Beraldin, J.A., Gonzo, L., Whiting, E., Jemtrud, M., Valzano, V., 2005. A hierarchical 3d reconstruction approach for documenting complex heritage sites. In: Proc. CIPA XX International Symposium, pp. 790–795.
- Gross, M.H., Pfister, H. (Eds.), 2007. Point-Based Graphics. Series in Computer Graphics. Morgan Kaufmann Publishers.
- Guidi, G., Frischer, B., Simone, M.D., Cioci, A., Spinetti, A., Carosso, L., Micoli, L.L., Russo, M., Grasso, T., 2005. Virtualizing ancient rome: 3d acquisition and modeling of a large plaster-of-paris model of imperial rome. *Proc. SPIE* 5665, 119–133.
- Ikeuchi, K., Oishi, T., Takamatsu, J., 2007. Digital Bayon temple — e-monumentalization of large-scale cultural-heritage objects —. In: Proc. ASIAGRAPH 2007, pp. 99–106.
- Kersten, T.P., Keller, F., Saenger, J., Schiewe, J., 2012. Automated generation of an historic 4d city model of hamburg and its visualisation with the ge engine. In: Progress in Cultural Heritage Preservation (Lecture Notes in Computer Science 7616), pp. 55–65.
- Kessenich, J., Sellers, G., Shreiner, D., 1996. OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.5 with SPIR-V. Addison-Wesley.
- Kobbelt, L., Botsch, M., 2004. A survey of point-based techniques in computer graphics. *Comput. Graph.* 28, 801–814.
- Koller, D., Frischer, B., Humphreys, G., 2009. Research challenges for digital archives of 3d cultural heritage models. *ACM J. Comput. Cult. Heritage* 2, Article No.7.
- Laycock, R.G., Drinkwater, D., Day, A.M., 2008. Exploring cultural heritage sites through space and time. *ACM J. Comput. Cult. Heritage* 1 Article No.11.
- Li, D., Gong, J., B. Yang, et al. (Eds.), 2017a. Proceedings of ISPRS Geospatial Week 2017. ISPRS (online).
- Li, W., Shigeta, K., Hasegawa, K., Li, L., Yano, K., Tanaka, S., 2017b. Collision visualization of a laser-scanned point cloud of streets and a festival float model used for the revival of a traditional procession route. *ISPRS Archives (Proc. ISPRS Geospatial Week 2017) XLII-2/W7*, pp. 255–261.
- Lichti, D.D., 2007. Error modelling, calibration and analysis of an am-cw terrestrial laser scanner system. *ISPRS J. Photogramm. Remote Sens.* 61, 307–324.
- Blas, M.R., Blanke, M., Rusu, R.B., Beetz, M., 2009. Fault-tolerant 3d mapping with application to an orchard robot. In: Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, pp. 893–898.
- Papon, J., Abramov, A., Schoeler, M., Worgotter, F., 2013. Voxel cloud connectivity segmentation - supervoxels for point clouds. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Porter, T., Duff, T., 1984. Compositing digital images. *Comput. Graph. (Proc. ACM SIGGRAPH '84)* 18, 253–259.
- Remondino, F., Girardi, S., Rizzi, A., Gonzo, L., 2009. 3d modeling of complex and detailed cultural heritage using multi-resolution data. *ACM J. Comput. Cult. Heritage* 2, Article No.2.
- Rusu, R.B., 2013. Semantic 3D Object Maps for Everyday Robot Manipulation. Springer.
- Sainz, M., Pajarola, R., 2004. Point-based rendering techniques. *Comput. Graphics* 28, 869–879.
- Seemann, P., Palma, G., Dellepiane, M., Cignoni, P., Goesele, M., 2018. Soft transparency for point cloud rendering. In: Eurographics Symposium on Rendering - Experimental Ideas & Implementations. The Eurographics Association.
- Slob, S., van Knapen, B., Hack, R., Turner, K., Kemeny, J., 2005. Method for automated discontinuity analysis of rock slopes with three-dimensional laser scanning. *Transp. Res. Rec. J. Transp. Res. Board* 1913, 187–194.
- Tanaka, S., Hasegawa, K., Okamoto, N., Umegaki, R., Wang, S., Uemura, M., Okamoto, A., Koyamada, K., 2016. See-through imaging of laser-scanned 3d cultural heritage objects based on stochastic rendering of large-scale point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci (Proc. XXIII ISPRS Congress, Prague, 2016) III*, 73–80.
- Thiel, F., Discher, S., Richter, R., Döllner, J., 2018. Interaction and locomotion techniques for the exploration of massive 3D point clouds in VR environments. *ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. (Proc. 3D Spatial Information Science - The Engine of Change) XLII*, 623–630.
- Weinmann, M., Jutzi, B., Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3d point cloud data. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci (Proc. ISPRS Workshop Laser Scanning 2013, Antalya, 2013) II*, 313–318.
- Yastikli, N., 2007. Documentation of cultural heritage using digital photogrammetry and laser scanning. *J. Cult. Heritage* 8, 423–427.
- Zhang, Y., Pajarola, R., 2007. Image composition for single-pass point rendering. *Comput. Graphics* 31, 175–189.
- Zwicker, M., Pfister, H., van Baar, J., Gross, M., 2002. EWA splatting. *IEEE TVCG* 8, 223–238.