

OPACITY-BASED EDGE HIGHLIGHTING FOR TRANSPARENT VISUALIZATION OF 3D SCANNED POINT CLOUDS

K. Kawakami¹, K. Hasegawa¹, L. Li¹, H. Nagata¹, M. Adachi², H. Yamaguchi³, F. I. Thufail⁴, S. Riyanto⁵, Brahmantara⁶, S. Tanaka^{1,*}

¹ College of Information Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan
- stanaka@is.ritsumei.ac.jp, liliang@fc.ritsumei.ac.jp, hasegawa@media.ritsumei.ac.jp

² Shrewd Design Co. Ltd, Fushimi-ku, Kyoto, Kyoto 612-8362, Japan - motoaki.adachi@shrewd-sekkei.com

³ Nara National Research Institute for Cultural Properties, Nijo-cho, Nara City, 630-8577, Japan - yamahiro@nabunken.go.jp

⁴ Research Center for Area Studies, Indonesian Institute of Sciences (P2W-LIPI), Jakarta, Indonesia
- fadjarthufail@gmail.com

⁵ Yogyakarta Archaeology Office (Balai Arkeologi DIY), Yogyakarta, Indonesia - sugeng66riyanto@gmail.com

⁶ Borobudur Conservation Office, Magelang, Jawa Tengah, Indonesia - bramantarayk@gmail.com

KEY WORDS: 3D Scanned Point Clouds, Transparent Visualization, 3D Edges, Opacity-based Edge Highlighting

ABSTRACT:

The recent development of 3D scanning technologies has made it possible to quickly and accurately record various 3D objects in the real world. The 3D scanned data take the form of large-scale point clouds, which describe complex 3D structures of the target objects and the surrounding scenes. The complexity becomes significant in cases that a scanned object has internal 3D structures, and the acquired point cloud is created by merging the scanning results of both the interior and surface shapes. To observe the whole 3D structure of such complex point-based objects, the point-based transparent visualization, which we recently proposed, is useful because we can observe the internal 3D structures as well as the surface shapes based on high-quality see-through 3D images. However, transparent visualization sometimes shows us too much information so that the generated images become confusing. To address this problem, in this paper, we propose to combine “edge highlighting” with transparent visualization. This combination makes the created see-through images quite understandable because we can highlight the 3D edges of visualized shapes as high-curvature areas. In addition, to make the combination more effective, we propose a new edge highlighting method applicable to 3D scanned point clouds. We call the method “opacity-based edge highlighting,” which appropriately utilizes the effect of transparency to make the 3D edge regions look clearer. The proposed method works well for both sharp (high-curvature) and soft (low-curvature) 3D edges. We show several experiments that demonstrate our method’s effectiveness by using real 3D scanned point clouds.

1. INTRODUCTION

The recent development of 3D scanning, that is, photogrammetry and laser scanning, has enabled us to quickly and accurately record complex 3D objects in the real world. The complexity becomes significant in cases in which the scanned objects have internal 3D structures and the acquired point clouds describe both the interior and surface shapes. To observe the whole 3D structure of such complex point-based objects, the point-based transparent visualization, which we recently proposed and named “stochastic point-based rendering (SPBR)” (Tanaka et al., 2016), is useful. By using SPBR, we can observe the internal 3D structures as well as the surface shapes by high-quality see-through 3D images.

However, transparent visualization sometimes shows us too much information so that the generated images become confusing. To address this problem, in this paper, we propose combining edge highlighting with transparent visualization based on SPBR. We highlight 3D edges, which are extracted as high-curvature areas of 3D scanned point clouds, in the transparent visualization. The edge highlighting works well together with the transparent visualization, and we can obtain comprehensible 3D see-through images. In Figure 1, we compare an ordinary transparent visualization (the left image) with an image that incorporates the edge-highlighting effect (the right image).

We can see that highlighting the edges makes the 3D internal structures of the visualized object more understandable.

In addition, we propose a new edge highlighting method applicable to 3D scanned point clouds. We call the method “opacity-based edge highlighting.” Conventional edge-highlighting methods are based on simple feature-value binarization and proper coloring of the high-feature-value areas. However, our new edge highlighting utilizes transparency, which is made available by using SPBR, to make the 3D edges look thinner and sharper. Opacity-based edge highlighting works well for both sharp (high-curvature) and soft (low-curvature) 3D edges.

The organization of this paper is as follows: In Section 2, we review related work. In Section 3, we briefly review the prescription of SPBR, whose ability to realize transparency is used in our proposed method. In Section 4, we describe the method of the proposed opacity-based edge highlighting. In Section 5, we show several case studies to demonstrate the effectiveness of the proposed edge-highlighting method. Section 6 is the concluding section, which summarizes our achievements.

2. RELATED WORK

Highlighting 3D edges or, more generally, feature highlighting of 3D point clouds has been actively studied (see the review in (Rusu, 2013) for details). Varieties of computational techniques have been used for the research. The purposes of the research

*Corresponding author

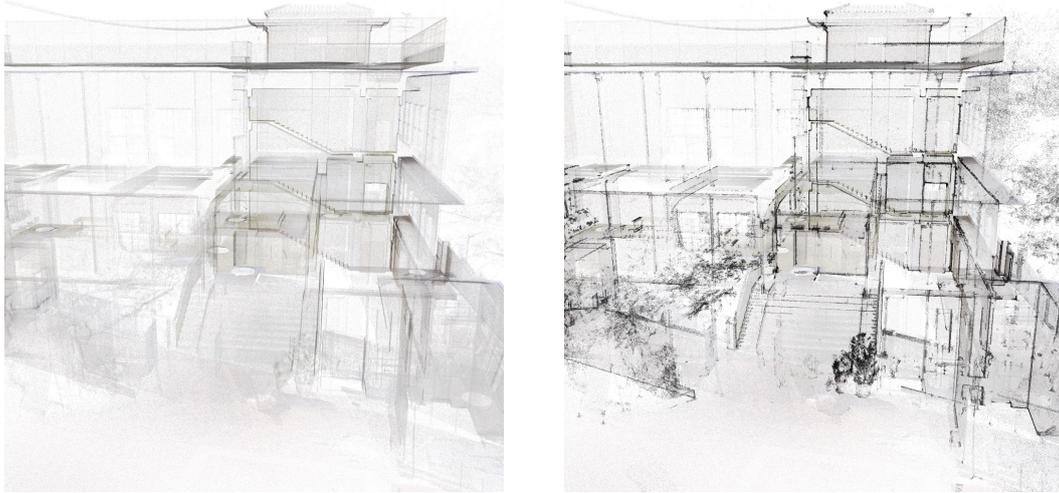


Figure 1. Transparent visualizations of a 3D scanned campus building. The left image shows the ordinary case without the edge-highlighting effect. The right image is created by incorporating our edge-highlighting effect. The campus building is at Kyoto Women's University in Kyoto City, Japan.

are understanding of 3D structures of point-based objects, classification and segmentation of points, and others.

Recently, statistical methods that use eigenvalue-based 3D feature values are becoming popular (West et al., 2004, Rusu, 2010, Toshev et al., 2010, Demantké et al., 2011, Mallet et al., 2011, Weinmann et al., 2013, Weinmann et al., 2014, Dittrich et al., 2017, He et al., 2017). In these statistical methods, the 3D feature values are defined based on the eigenvalues of the local 3D covariance matrix, which is also called the 3D structure tensor (Jutzi, Gross, 2009). For each 3D point, the 3D structure tensor is calculated by numerically investigating local variances and covariances of point distributions within a certain radius of a spherical neighborhood. It is also useful for adaptively tuning the spherical radius based on local distributional properties (Weinmann et al., 2014, He et al., 2017).

In this paper, we also utilize the abovementioned eigenvalue-based 3D feature values to extract high-curvature areas, i.e., 3D edges, from given point clouds. The new contribution of this paper is relating the 3D feature values to the opacity distribution of 3D scanned point-based surfaces, so that we can realize their (transparent) visualization with their various types of 3D edge regions highlighted.

3. BRIEF REVIEW OF SPBR

SPBR (Stochastic Point-based Rendering) realizes fast and precise 3D see-through imaging, that is, transparent visualization of large-scale 3D scanned point clouds. The transparency originates from the stochastic determination of pixel intensities. This stochastic algorithm achieves the correct depth feel without requiring the time-consuming depth sorting of 3D points. Below, we briefly review the method of SPBR (Tanaka et al., 2016) with an extension of the local opacity tuning.

SPBR can control surface opacity by adjusting the point density of the visualized point cloud. We make upsampling or downsampling such that the user-defined opacity is realized according to the opacity formula (see below). If the initial point density is too large, we randomly select an appropriate number of 3D points from the raw data and remove them. If the point density

is too small, we randomly select an appropriate number of 3D points and create the required number of their copies. Then, we randomly divide the density-adjusted point cloud to multiple subgroups, each of which has the same number of 3D points. Below, we call the subgroups “point ensembles.” By creating an image for each point ensemble with the point-occlusion effect incorporated and then averaging the created images, we obtain the final transparent image.

The original paper of SPBR (Tanaka et al., 2016) treats cases that visualized point clouds as having a uniform point distribution. However, the abovementioned point-density adjustment can be executed locally such that we can assign different opacities to selected local areas.

Let us focus on a local area with an area size of S . We execute the point-density adjustment for this local area such that the following user-defined surface opacity α is realized:

$$\alpha = 1 - \left(1 - \frac{s}{S}\right)^{n_{\text{adj}}/L}, \quad (1)$$

where s is the particle cross-section area, which is tuned such that an image of each 3D point overlaps exactly one pixel, n_{adj} is the adjusted number of 3D points in this local area, and L is the number of point ensembles.

The specific steps to execute the transparent visualization are:

- STEP 1. Creation of point ensembles:** Prepare L point ensembles by randomly dividing the given raw point cloud after appropriate point-density adjustment according to formula (1).
- STEP 2. Point projection per point ensemble:** For each point ensemble, independently project its constituent 3D points to the image plane. As a result, L intermediate images are created. In the projection process, we incorporate the point-occlusion effect per pixel.
- STEP 3. Averaging the intermediate images:** Create an average image by averaging intensities of the corresponding pixels in the above L intermediate images. The created average image becomes the final transparent image. (The ensemble number L controls the statistical accuracy and works as an image-quality parameter.)

4. OPACITY-BASED EDGE HIGHLIGHTING

In this section, we propose our edge-highlighting method, which we call “opacity-based edge highlighting.” In the method, first, we extract high-curvature areas from the given point cloud. We regard these areas as 3D edges of the point-based surface. The extraction is made by referring an appropriate feature value that is calculated and assigned to each point beforehand. Second, we execute the transparent visualization of the point cloud with its high-curvature areas, i.e., 3D edges, highlighted. The key idea of our opacity-based edge highlighting is that we assign higher opacities to the 3D edges. This idea can make the 3D edges look brighter compared with the surrounding nonedge regions. Note that we utilize the degree of opacity, which is made available by using SPBR, for highlighting 3D edges.

4.1 Eigenvalue-based 3D Feature Values

In this paper, we utilize the eigenvalue-based 3D feature values to extract high-curvature areas, i.e., 3D edges, from the given point cloud. Several kinds of eigenvalue-based 3D feature values have been proposed. In our research, we adopt “change of curvature” and “linearity”:

$$\text{Change of curvature: } C_\lambda = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}, \quad (2)$$

$$\text{Linearity: } L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}, \quad (3)$$

where λ_1 , λ_2 , and λ_3 are the eigenvalues of the 3D structure tensor (the local 3D covariance matrix) (Jutzi, Gross, 2009) with $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$. In our implementation, the 3D structure tensor is calculated for each spherical local region centered at each point. Change of curvature C_λ measures the minimal extension of the local point distribution that should vanish in the case of planar distribution. Linearity L_λ measures the difference between the two independent-directional largest extensions of the local point distribution that should also vanish in the case of planar distribution. Thus, a large value of C_λ or L_λ indicates that the examined local set of points forms a 3D edge of the point-based surface.

We also introduce another feature value “aplanarity”:

$$\text{Aplanarity: } \bar{P}_\lambda = 1 - P_\lambda, \quad (4)$$

where $P_\lambda = (\lambda_2 - \lambda_3)/\lambda_1$ is known as feature value “planarity.” Aplanarity \bar{P}_λ is the opposite of planarity and describes the degree of a nonplanar feature of the local point distribution.

In our edge highlighting, we adopt one of C_λ , L_λ , and \bar{P}_λ , depending on a target point cloud. Below, we represent the adopted 3D feature value as f . When using f , we normalize it such that its maximal value in the target point cloud becomes one, and the values of f are distributed between zero and one.

4.2 Method of Opacity-based Edge Highlighting

We define 3D edge regions as areas with large 3D feature values, i.e., values of f . As we mentioned at the beginning of Section 4, the fundamental idea of our opacity-based edge highlighting is to assign higher opacities to the 3D edge regions in the transparent visualization of SPBR. This idea can make the 3D edges look brighter compared with the surrounding nonedge regions. For the color of the 3D edges, we assign an appropriate

highlight color, e.g., red, or keep the original (but brightened) point colors.

To highlight the 3D edges, we introduce three types of functions to relate 3D feature value f to opacity α : type (a), type (b), and type (c). We select one of the three, depending on the shapes and other properties of the 3D edges to be highlighted. Below, we explain the three types.

4.2.1 Type (a) The function of type (a) adopts the simplest binary relation between f and α . We assign a constant high opacity to local areas with f values larger than a user-defined threshold f_{th} . In Figure 2, we show a typical graph of f and α for the function of type (a).

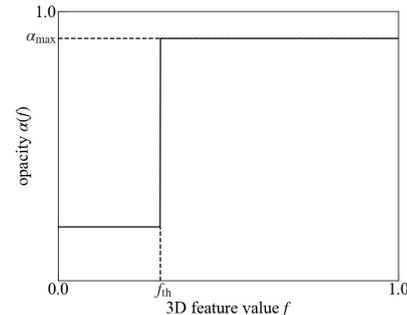


Figure 2. Typical relation between 3D feature value f and opacity α when adopting the function of type (a). The relation has the step-functional feature, and f_{th} is the user-defined threshold value that defines the stepping position.

The simple highlighting strategy of type (a) is robust and works well in many cases. However, the highlighted edge widths cannot be smaller than the diameter of the spherical local regions to calculate the 3D structure tensor, and we are often required more thinning of the edge widths. Type (b) is designed to improve this problem (see Section 4.2.2).

Let us show an experiment to verify the edge highlighting based on the function of type (a). We constructed a point-based 3D model where two squares are joined at a right angle. In the model, a 3D edge should appear between the two squares. We highlighted the 3D edges of this model based on a function of type (a). Change of curvature C_λ is adopted as f . The created 2D image is shown in Figure 3 (right). We can confirm a clear 3D edge as a bright vertical line. At each horizontal position in the edge region of this 2D image, we averaged pixel intensities along the vertical direction. Then, we investigated the change in the average pixel intensity along the horizontal direction across the edge region. We show the result in the graph of Figure 3 (left). The graph forms a square shape approximately corresponding to the step-functional feature of the opacity distribution shown in Figure 2.

4.2.2 Type (b) The function of type (b) is used to visualize 3D edges as thinner lines compared with type (a). This highlighting strategy is suitable for highlighting sharp 3D edges that have high curvatures within narrow local areas.

In Figure 4, we show a typical graph of f and α for the function of type (b). The 3D edge regions are defined as areas with $f \geq f_{th}$ as in type (a). The graph shows two essential aspects of edge highlighting based on the function of type (b). The first aspect is that the larger f is, the more opaque the corresponding local regions become, which means that the narrow areas

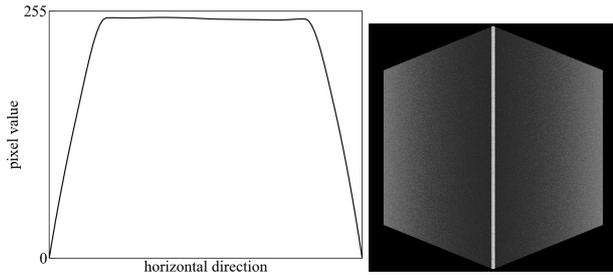


Figure 3. Verification experiment of the function of type (a). The right image shows the edge-highlighting result of a point-based 3D model consisting of two joined squares. The graph on the left shows the change in pixel intensity around the 3D edge in the horizontal direction in the right image. The parameters are set to $L = 100$, $f_{th} = 0.1$, $\alpha_{max} = 0.9$. The opacity of non-edge regions is set to 0.2.

along the centerlines of the 3D edges look brighter compared to the surrounding areas. The second aspect is that the graph has a sharp peak at the maximal value of f , which means that the centerlines are visualized with particularly bright intensities. These aspects make 2D images of the visualized 3D edges thinner compared with the cases of type (a).

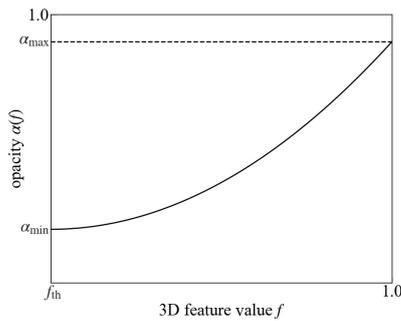


Figure 4. Typical relation between 3D feature value f and opacity α when adopting the function of type (b). α gradually increases for f larger than f_{th} and forms a sharp peak at the maximal value of f .

The relation of f and α as in the graph of Figure 4 is realized by defining the following function $\alpha(f)$:

$$\alpha(f) = \frac{\alpha_{max} - \alpha_{min}}{(1 - f_{th})^d} (f - f_{th})^d + \alpha_{min} \quad (5)$$

On the right-hand side, f_{th} is the threshold value of f , over which the corresponding local regions are regarded as 3D edges. Minimal opacity α_{min} and maximal opacity α_{max} are assigned to areas with $f < f_{th}$ and $f = 1$, respectively. In the intermediate range, $f_{th} \leq f < 1$, the opacity $\alpha(f)$ increases as f becomes larger. The parameter d controls the speed of the opacity increase. Usually, we set d in a range $1.0 \leq d \leq 3.0$.

We executed a verification experiment that is similar to the one in Section 4.2.1 by adopting the function of type (b). We used the same point-based 3D model consisting of the two joined squares. The change of curvature C_λ is adopted as f . The created 2D image is shown in Figure 5 (right). We can confirm a clear 3D edge as a bright vertical line, which is thinner than the one in Figure 3 (right). This experimental result demonstrates the edge-thinning effect of the function of type (b).

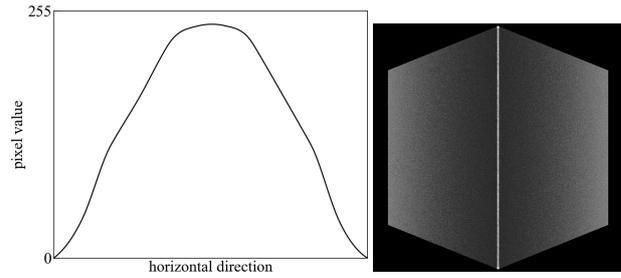


Figure 5. Verification experiment that is similar to Figure 3 for the function of type (b). The parameters are set to $L = 100$, $f_{th} = 0.1$, $\alpha_{min} = 0.2$, $\alpha_{max} = 1.0$, $d = 3.0$. The opacity of nonedge regions is set to 0.2.

4.2.3 Type (c) The function of type (c) enables us to visualize soft 3D edges. We define a soft 3D edge as a region that is not a sharply defined 3D edge but has comparably higher curvatures than the surrounding areas. The soft edges often characterize the 3D structure of 3D scanned objects. Note that the conventional methods of highlighting 3D edges often fail for soft edges.

In Figure 6, we show a typical graph of f and α for the function of type (c). There are two threshold values, f_{th} and F_{th} , with $f_{th} < F_{th}$. The 3D edge regions are defined as areas with $f \geq f_{th}$ as in types (a) and (b). α gradually increases for $f_{th} \leq f < F_{th}$ and becomes constant for $f \geq F_{th}$.

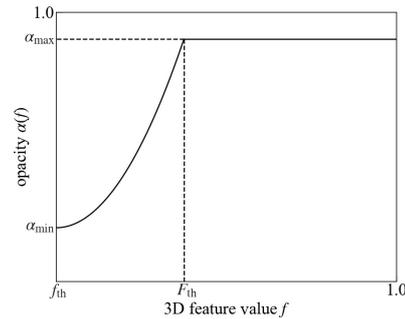


Figure 6. Typical relation between 3D feature value f and opacity α when adopting the function of type (c). α gradually increases in range $f_{th} \leq f < F_{th}$ and becomes constant for f larger than F_{th} .

Let us consider the meaning of the opacity graph in Figure 6 in more detail. f in the soft-edge regions takes more varieties of values than sharp-edge regions. Therefore, we need the plateau area at $f > F_{th}$ to cover all soft-edge regions. In addition, we make the opacity gradually decrease as the position moves away from the centerline of the edge regions (see the area $f_{th} \leq f < F_{th}$). This opacity distribution gives the narrow areas around the centerlines comparably higher opacity, and the areas are highlighted with brighter colors.

The relation of f and α as in the graph of Figure 6 is realized by defining the following function $\alpha(f)$:

$$\alpha(f) = \begin{cases} \frac{\alpha_{max} - \alpha_{min}}{(F_{th} - f_{th})^d} (f - f_{th})^d + \alpha_{min} & (f_{th} \leq f < F_{th}) \\ \alpha_{max} & (F_{th} \leq f \leq 1) \end{cases} \quad (6)$$

Definitions of the parameters other than F_{th} on the right-hand side are the same as formula (5).

In Figure 7, we compare the results of using the functions of types (a) (left) and type (c) (right) for a cubic surface with rounded (soft) edges. Only 3D edge regions are visualized, eliminating the other areas. The change of curvature C_λ is adopted as f . In the right figure, we can see that the use of the type (c) function makes the centerlines of the 3D edges brighter than the surrounding areas. The brightness gradually decreases as the position moves away from the centerlines. This gradation effect makes the 3D edges look thinner than in the left image with type (a). For the left image, the parameters are set to $L = 100$, $f_{th} = 0.3$, $\alpha_{max} = 0.9$. For the right image, the parameters are set to $L = 100$, $f_{th} = 0.3$, $F_{th} = 0.8$, $\alpha_{min} = 0.2$, $\alpha_{max} = 1.0$, $d = 2.0$.

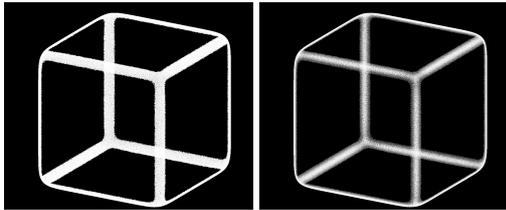


Figure 7. Comparison of the visualization results of using the functions of type (a) (left) and type (c) (right).

5. CASE STUDIES

In this section, we present three case studies that demonstrate the effectiveness of our opacity-based edge highlighting for various 3D scanned data. More specifically, we present examples where the functions of types (a), (b), and (c) work well.

Here, we summarize some parameter values used for creating figures in this section: We set the radius of the spherical local region for calculating the 3D structure tensors (see Section 4.1) such that each local region includes a few hundred points. It is convenient to define the radius in units of the bounding-box diagonal length of the target point cloud. For Figures 1, 10, and 11, the spherical radii are adaptively tuned by referring to the local measure of eigenentropy (Weinmann et al., 2014). For Figure 1, the maximal and minimal radii are $1/150$ and $1/500$, respectively. For Figures 10 and 11, the maximal and minimal radii are $1/300$ and $1/600$, respectively. For Figures 8, 9, 12, and 14, the spherical radius is fixed to $1/300$, and $1/500$ for Figure 15. The opacity of nonedge regions is set to 0.5 for Figures 14 and 15, and 0.2 for the other figures. The ensemble number L is set to 100 for all the figures. The image resolution is set to 1024^2 for all the figures except for Figures 14 and 15. For the two figures, an image is first created with resolution 2000^2 for the whole of one relief panel, and its part is clipped.

5.1 Application of the type (a) function

In this subsection, we show edge-highlighting examples when adopting the type (a) function, which defines the simplest binary step-functional opacity distribution of Figure 2.

In Figure 1, for a campus building, we compare the ordinary transparent visualization by SPBR (the left image) and the edge-highlighted transparent visualization realized by adopting the type (a) function (the right image). In the right image, we

observe the complex internal 3D structures of the building more understandably by highlighting the 3D edges as black lines.

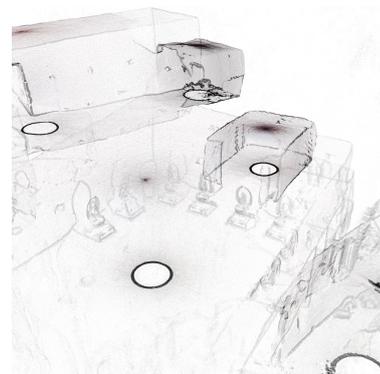
To create Figure 1 (right), change of curvature C_λ is adopted as the 3D feature value f . The black color is assigned to the 3D edge regions as the highlight color. The parameters that appear in Figure 2 are set as follows: $f_{th} = 0.25$, $\alpha_{max} = 0.9$.

In Figures 8 and 9, we show similar comparisons for the 3D scanned data of the prayer chapel of the Zuiganji Buddhist Temple, which was made by excavating rock walls. The chapel consists of many cuboid-shaped rooms, in each of which many stone Buddhist statues are enshrined. Figure 8 compares the see-through bird's-eye views with and without highlighting the 3D edges. In Figure 8 (b) with the edge highlighting, we can observe the whole 3D structure of the chapel better, guided by the highlighted 3D edges. Human-made structural objects often have clearly defined 3D edges, which are available in our edge-highlighting method to improve the comprehensibility. Images of Figure 9 are created by setting the viewpoint inside a chapel room. In Figure 9 (b) with the edge highlighting, we can observe outlines of the stone Buddhist statues more clearly. This example demonstrates the usability of our method to highlight object outlines that define the silhouettes of 3D scanned objects.

To create Figures 8 (b) and 9 (b), aplanarity \bar{P}_λ is adopted as f . For curved outlines, aplanarity often works better than linearity and change of curvature (see also Section 5.4). The parameters are set as follows: $f_{th} = 0.4$, $\alpha_{max} = 0.9$.



(a) Ordinary transparent visualization without edge highlighting



(b) Opacity-based edge highlighting (type (a))

Figure 8. Comparative visualizations of the prayer chapel in the Zuiganji Buddhist Temple (1): (a) The ordinary transparent visualization by SPBR without edge highlighting. (b) Opacity-based edge highlighting (type (a)).

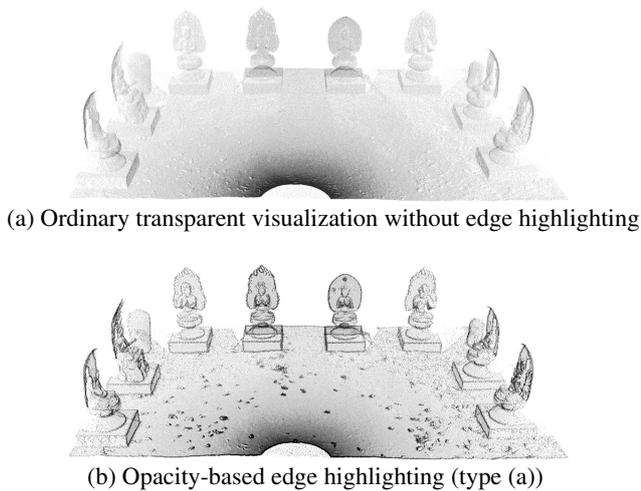


Figure 9. Comparative visualizations of the prayer chapel in the Zuiganji Buddhist Temple (2): viewpoint inside the chapel.

5.2 Application of the type (b) function

In this subsection, we show edge-highlighting examples when adopting the type (b) function, which defines the opacity distribution with a gradually increasing region and a sharp peak (see Figure 4). This function works well for highlighting sharply defined 3D edges.

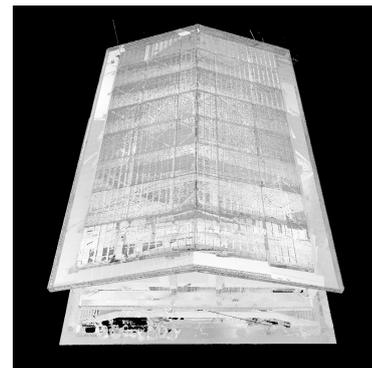
Figure 10 shows the visualization results obtained by using our 3D scanned point cloud of a gymnasium in an elementary school. Figure 10 (a) shows the original 3D scanned point cloud. Figure 10 (b) shows the visualization created by the conventional opaque point-based rendering, where only 3D edge regions are visualized. Figure 10 (c) shows the transparent visualization with our opacity-based edge highlighting with the type (b) function. We can see that the visualized 3D edges look much thinner than the conventional visualization by gradually increasing the transparency, depending on the distance of the position from the 3D-edge centerlines. To create Figure 10 (c), change of curvature C_λ is adopted as f . The parameters that appear in formula (5) are set as follows: $f_{th} = 0.3$, $\alpha_{min} = 0.2$, $\alpha_{max} = 1.0$, $d = 3.0$.

Figure 11 demonstrates the effect of increasing parameter d in formula (5). We can see that the 3D edges for $d = 3.0$ (Figure 11 (b)) are thinner than those for $d = 1.0$ (Figure 11 (a)), which proves that d can control the speed of edge thinning.

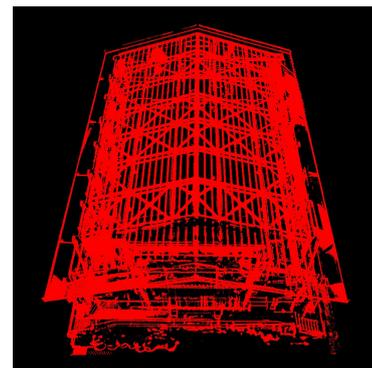
Figure 12 shows the edge-highlighting transparent visualization of a festival float with high cultural value. Linearity L_λ is adopted as the 3D feature value f . The 3D edges are highlighted by adopting the type (b) function with $f_{th} = 0.25$, $\alpha_{min} = 0.2$, $\alpha_{max} = 1.0$, and $d = 2.0$. This float is composed of many narrow square timbers. For such an object, sharply highlighted 3D edges of the timbers work effectively for us to understand how the timbers are built up.

5.3 Application of the type (c) function

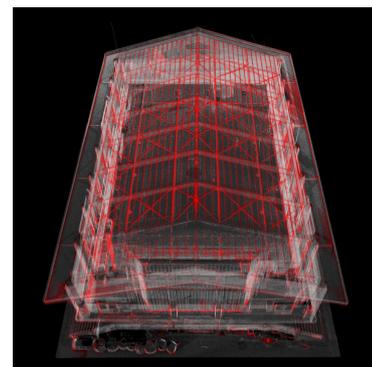
In this subsection, we show edge-highlighting examples when adopting the type (c) function, which defines the opacity distribution with a plateau area for high opacities (see Figure 6). This function works well for highlighting soft (round) 3D edges.



(a) 3D scanned point cloud



(b) Conventional opaque point-based rendering of 3D edges



(c) Opacity-based edge highlighting (type (b))

Figure 10. Comparison of visualizations of a gymnasium in an elementary school.



(a) $d = 1.0$



(b) $d = 3.0$

Figure 11. Effect of the parameter d in formula (5).

For the demonstration, we use 3D scanned data of relief pan-

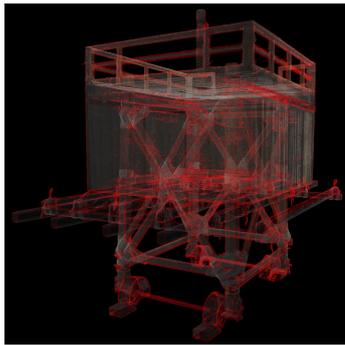


Figure 12. Opacity-based edge highlighting (type (b)) of a 3D scanned point cloud of the Hachiman-Yama float used in the Gion Festival in Kyoto City, Japan.

els of the Borobudur Temple (UNESCO World Heritage Site) in Yogyakarta, Indonesia (see Figure 13).

Figure 14 shows the edge-highlighting visualization of a part of Karmavibhangga relief. Many human characters, each of which has soft 3D edges as its outline, are drawn on the panel. Figure 14 (a) is created by the conventional opaque point-based rendering, where regions with $f \geq f_{th}$ are assigned the red color. Since the soft edges have larger widths compared with the sharply defined edges, the visualized red regions form wide bands rather than thin lines, so that the outlines of human characters become unclear. In contrast, Figure 14 (b) shows the visualization with our opacity-based edge highlighting with the type (c) function. The gradation of the red color appears, and the brightness becomes higher around the centerline of the red regions, which makes the outlines more recognizable.

Figure 15 shows the similar edge-highlighting visualization of the famous ancient ship drawn in a relief panel of the Borobudur Temple. In the right image, the detailed design of the ship is more understandable than in the left image due to the color-gradated 3D edges and outlines.

For creating Figures 14(b) and 15 (right), change of curvature C_λ is adopted as f . The parameters in formula (6) are set as follows: $f_{th} = 0.03$ (Figure 14 (b)), $f_{th} = 0.02$ (Figure 15 (right)), $F_{th} = 0.2$, $\alpha_{min} = 0.2$, $\alpha_{max} = 1.0$, and $d = 2.0$.

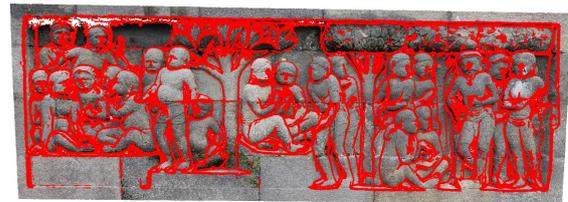


Figure 13. 3D scanned data of reliefs of the Borobudur Temple.

We make one comment here. It is true that Figures 14 (b) and 15 (right) are transparent visualizations. However, the effect of transparency only reduces the brightness here. As long as there are no hidden objects behind, the transparent visualizations cannot be distinguished from opaque visualizations. In this sense, the visualizations of Figures 14 (b) and 15 (right) can also be regarded as improved opaque edge-highlighting visualizations.

5.4 Comparative visualizations of 3D feature values

Selection of the most effective type of 3D feature value depends on the target point cloud. Generally speaking, change



(a) Conventional opaque point-based rendering



(b) Opacity-based edge highlighting (type (c))

Figure 14. Comparative visualizations of the Borobudur relief (1): Highlighting outlines of human characters in the relief.

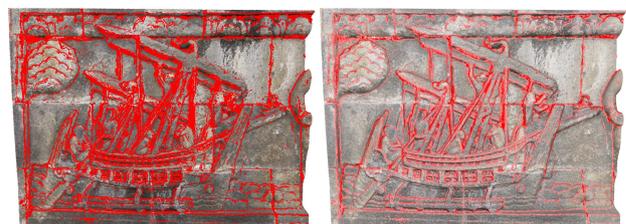


Figure 15. Comparative visualizations of the Borobudur relief (2): The left image is created by the conventional opaque point-based rendering and the right image by SPBR with the opacity-based edge highlighting (type (c)).

of curvature C_λ is mostly robust (Dittrich et al., 2017). Linearity L_λ works well for detecting outlines of surfaces to which no neighboring surfaces are connected. This situation often occurs when the neighboring surfaces are invisible from the scanning viewpoint. Aplanarity P_λ works similarly to linearity, but it works better for detecting curved outlines.

We executed the edge-highlighted visualization for the data in Figure 9 by adopting change of curvature, linearity, and aplanarity, respectively. Figure 16 shows the comparison result of the partial enlargement of the same portion in Figure 9. We can see that the edges extracted using aplanarity are the sharpest, followed by linearity and change of curvature.

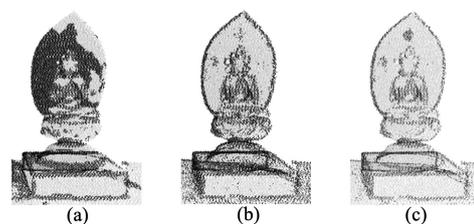


Figure 16. Comparative visualizations with different 3D feature values: (a) change of curvature, (b) linearity, (c) aplanarity.

5.5 Performance

In Table 1, we summarize the rendering speeds in frames per second (fps) together with the number of points in the point

clouds after the point density adjustment (see Section 3). We can see that the rendering speeds are a few tens of fps for 10^7 points and several fps for 10^8 points. These speeds are fast enough for interactive visual analysis. Roughly speaking, the number of points is doubled by the point-density adjustment in the 3D edge regions.

Figure	1(b)	8(b)	10(c)	12	13(b)
fps	49.4	63.6	31.6	47.8	8.53
Points ($\times 10^6$)	35.8	15.3	63.7	44.4	87.1

Table 1. Rendering speeds (fps) and numbers of points for point clouds with 3D edges highlighted.

The rendering speeds are measured for STEPs 2 and 3 of SPBR. The computations were executed on a Linux PC with an Intel Xeon E5-2687W (3.00 GHz, 512 GB of memory) CPU and an NVIDIA Quadro P5000 (16 GB of memory) GPU.

6. CONCLUSIONS

In this paper, we proposed a novel method to highlight 3D edges of a 3D scanned point cloud in its transparent visualization. We call the method opacity-based edge highlighting. The key idea is that we assign higher opacities to the 3D edges to make the edges brighter. This idea is realized by utilizing SPBR, which was recently proposed for high-quality transparent visualization of large-scale 3D scanned point clouds. The proposed method makes the 3D edges look brighter compared with the surrounding nonedge regions. This effect improves visibility and compensates for the disadvantage of transparent visualization, that is, showing too much information, which is confusing.

We investigated three types of functions to relate the 3D feature value to the surface opacity. The type (a) function defines the basic binary step-functional relation, which is useful in many cases. The type (b) function with a sharp peak works to make the 3D edges look thinner and sharper, and this type is useful to highlight sharply defined 3D edges. The type (c) function, which has a wide plateau area, enables us to visualize soft (round) 3D edges, realizing gradation of the highlight color that becomes brightest at the centerlines of the 3D edges. We can select one of the three types, depending on the edge shape. We applied our opacity-based edge highlighting with each function type to real 3D scanned point clouds and demonstrated the effectiveness of the method successfully.

The limitation of the proposed method is that we need to set several parameters manually. Our future target is to realize semi-automatic determination of the parameter values by investigating local point distributions. Adaptive selection of the opacity-function type is also promising, which can make the method applicable to shapes with varieties of curvatures. Incorporation of the idea of the contour detection in (Hackel et al., 2017) into our method is also interesting.

ACKNOWLEDGMENTS

The images of the campus building, the prayer chapel, the gymnasium, the Hachiman-Yama float, and the reliefs are presented with the permission of Kyoto Women's University, Zuiganji Temple and Matsushima-cho, the elementary school, the Hachiman-Yama Preservation Society, and the Borobudur Conservation Office, respectively. We thank Prof. Y. Kitao

(Kyoto Women's University) for his kind support. This research is supported by JSPS KAKENHI Grant Number 19KK0256 and budgets of the Asia-Japan Research Institute (Ritsumeikan University).

REFERENCES

- Demantké, J., Mallet, C., David, N., Vallet, B., 2011. Dimensionality based scale selection in 3D lidar point clouds. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. XXXVIII, Part 5/W12, 97-102.
- Dittrich, A., Weinmann, M., Hinz, S., 2017. Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3D point cloud data. *ISPRS J. Photogramm. Remote Sens.*, 126, 195-208.
- Hackel, T., Wegner, J. D., Schindler, K., 2017. Joint classification and contour extraction of large 3D point clouds. *ISPRS J. Photogramm. Remote Sens.*, 130, 231-245.
- He, E., Chen, Q., Wang, H., Liua, X., 2017. A Curvature Based Adaptive Neighborhood for Individual Point Cloud Classification. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. XLII-2/W7, 219-225.
- Jutzi, B., Gross, H., 2009. Nearest neighbour classification on laser point clouds to gain object structures from buildings. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. XXXVIII, Part 1-4-7/W5.
- Mallet, C., Bretar, F., Roux, M., Soergel, U., Heipke, C., 2011. Relevance assessment of full-waveform lidar data for urban area classification. *ISPRS J. Photogramm. Remote Sens.*, 66(6), S71-S84.
- Rusu, R. B., 2010. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. *KI - Künstliche Intelligenz*, 24(4), 345-348.
- Rusu, R. B., 2013. *Semantic 3D Object Maps for Everyday Robot Manipulation*. Springer Tracts in Advanced Robotics 85, Springer-Verlag, ISSN 1610-7438.
- 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.*, III(5), 73-80.
- Toshev, A., Mordohai, P., Taskar, B., 2010. Detecting and parsing architecture at city scale from range data. *Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 398-405.
- 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.*, II(5/W2), 313-318.
- Weinmann, M., Jutzi, B., Mallet, C., 2014. Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, II(3), 181-188.
- West, K. F., Webb, B. N., Lersch, J. R., Pothier, S., Triscari, J. M., Iverson, A. E., 2004. Context-driven automated target detection in 3d data. *Proceedings of SPIE*, Vol. 5426, 133-143.