A POINT-BASED RENDERING APPLICATION WITH A HIGH-SPEED SPATIAL INTERPOLATION

Masafumi Nakagawa, Satoshi Kuramochi, Anna Nakanishi and Yuta Sugaya
Department of Civil Engineering, College of Engineering, Shibaura Institute of Technology,
3-7-5, Toyosu, Koto-ku, Tokyo 135-8548, Japan; Tel: +81-3-5859-8355
Email:mnaka@shibaura-it.ac.jp

KEY WORDS: LiDAR, Terrestrial 3D mapping, Point cloud, Point-based rendering, Panorama imagery, VR

ABSTRACT: Accurate 3-D scanning techniques are used for various applications. However, a viewpoint translation in point-cloud rendering reduces the visualization quality because of noticeable occlusion exposure and a noticeably uneven point distribution. In addition, recent increases in scanning speed and scanning data volume require a large video memory and high-frequency CPU. Therefore, we have developed a point-based rendering application with a high-speed spatial interpolation using a point cloud to generate virtual reality (VR) data called LiDAR VR. We have focused on the following features. First, panorama imagery can be generated automatically from arbitrary viewpoints from one observation data set. Second, it is possible that a spatial interpolation with rendering results can reduce the processing task. These features have been verified using case studies.

1. INTRODUCTION

An advantage of 3D point-cloud data is that it allows accurate display from an arbitrary viewpoint. By contrast, panorama imagery such as QuickTime VR (Shenchang, 1995) by Apple Inc. has the advantage of providing higher attractiveness with less data. In addition, panoramic image georeference (Jason, 2000) and distance-value-added panoramic image processing (Edward, 2005) show that both advantages can be combined for 3D-GIS visualization. Thus, we focus on the possibility that these advantages can be combined by a point-cloud projection into panorama space. Splat-based ray tracing (Lars, 2007) is a methodology to generate a photo-realistic curved surface on a panoramic view using normal vector from point-cloud data. A problem is the large amount of time required for surface generation in 3-D work space. Furthermore, the curved-surface description is inefficient at representing urban and natural objects as GIS data. In particular, we consider that a simpler filtering algorithm is important to achieve high volumes of point-cloud processing with high speeds. Thus, our objective is to develop a point-based rendering application with a simpler filtering algorithm to generate VR data, which we call LiDAR VR data.

We have focused on the following features. The first feature is LiDAR VR imagery generation with a translated viewpoint. In general, an image from a viewpoint should be created from an observation. Moreover, when panorama images are taken from several viewpoints, position data from RTK-GPS and rotation data from a gyrosensor are required to calculate relative positions and rotations between cameras. LiDAR VR images from many viewpoints can be simulated from observation data through the approach proposed in this research. Moreover, the proposed methodology can simulate camera translation and generate new simulated imagery without the position and rotation sensors. The second feature is real-time spatial interpolation. In general point-cloud interpolation processing, missing points can be interpolated by a traditional 3-D filter from points in a space. This type of 3-D interpolation requires huge amounts of processing capability. Therefore, we focus on the possibility that spatial interpolation with rendered results can reduce the processing task. We have also developed a pixel-selectable averaging filter. This function can replace spatial filtering in 3-D space by spatial filtering in 2-D space with real-time processing. Thus, any quality reduction caused by viewpoint translation can be solved in real time.

2. METHODLOGY

The processing flow of the proposed methodology in this research is described below. First, sensors acquire a point cloud with additional color data such as RGB data or intensity data. The sensor position is defined as an origin point in a 3-D work space. If color data cannot be acquired, distance values are attached to a color index. Thus, for example, we can use a laser scanner, stereo camera, or time-of-flight camera. Second, a LiDAR VR image from the simulated viewpoint is generated using the point cloud. Finally, the generated LiDAR VR image is filtered to generate missing points in the rendered result using distance values between the viewpoint and objects.

2.1 LiDAR VR image generation using the point cloud

The colored point cloud is projected from 3-D space to panorama space. This transformation simplifies viewpoint translation, filtering, and point-cloud browsing. The LiDAR VR data consist of a panorama model and range data. The panorama space can be a cylindrical model, a hemispherical model, or a cubic model. Here, the spherical model is described. The measured point data are projected onto a spherical surface, and can be represented as range data as shown in Figure 1. The range data can preserve measured point data such as X, Y, Z, R, G, B, and intensity data in the panorama space in a multilayer style. Azimuth and elevation angles from the viewpoint to the measured points can be calculated using 3-D vectors generated from the view position and the measured points. When azimuth angles and elevation angles are converted to column counts and row counts in the range data with adequate spatial angle resolution, a spherical panorama image can be generated from the point cloud.

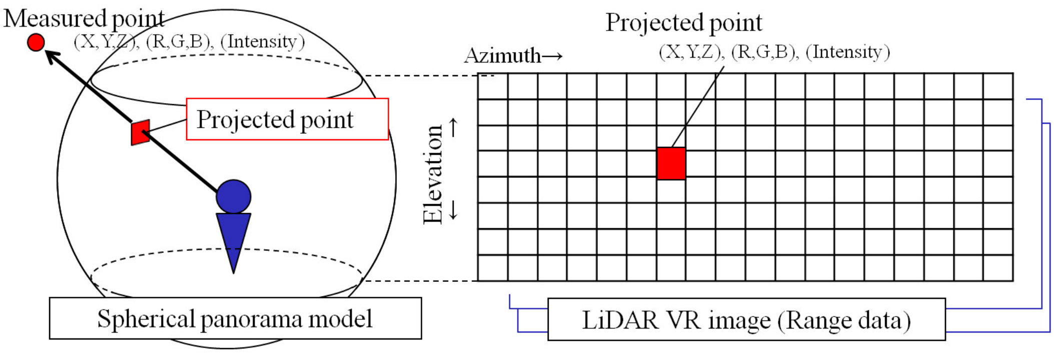

Figure.1 LiDAR VR consisting of a spherical panorama (left side in the figure) and range data (right side in the figure)

Based on this spherical panorama, the range data are generated using the point cloud with a translated viewpoint, as shown in Figure 2. When points from P_1 to P_{10} are projected into a panorama space generated from a viewpoint X_o, these points are arranged continuously from P_1 to P_{10} in the range data. An azimuth or elevation angle from a viewpoint X_o to a measured point P_1 is denoted as R_o. When the same scene is captured from a different viewpoint X_t, the angle from the viewpoint X_t to the measured point P_1 is denoted as R_t. The position of the projected point in the range data moves according to the change in angle from R_o to R_t.

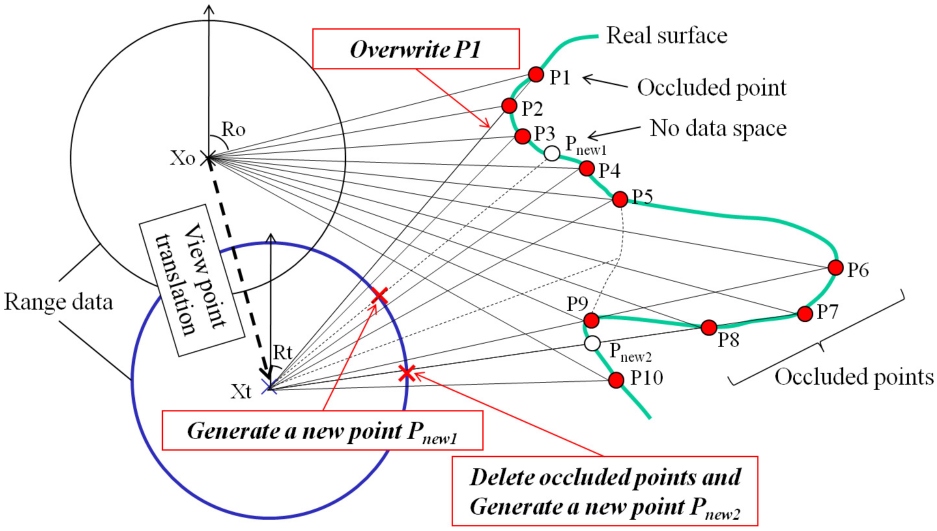

Figure.2 Point distribution calculated by viewpoint translation in range data, occlusion detection using the point cloud, and point-cloud interpolation with distance information

2.2 Filtering using distance values between a viewpoint and measured points

Three types of filtering are executed after the viewpoint translation, as shown in Figure 2. The first filtering is an overwriting of the occluded point. When the viewpoint is translated to X_t, the projected point P_1 becomes an occluded point behind P_2. Therefore, P_1 is overwritten by P_2. The second filtering is a generation of new points in no data space. This occurs when the viewpoint is translated to X_t and no data space exists between the projected points P_3 and the projected points P_4. Thus, Figure 2 shows that P_{new1} is generated. The third filtering is detection of occluded points and generation of new points instead of detected occluded points. When the viewpoint is translated to X_t, the point P_8 exists between P_9 and P_{10} after the first filtering. However, the actual point P_8 should be occluded because the point P_8 exists behind the real surface. Therefore, the occluded point P_8 should be given a new distance value as P_{new2}, calculated with interpolation processing using the distance values of points P_9 and P_{10}. In addition, new points are generated with a pixel-selectable averaging filter developed in this research as follows.

2.3 Pixel-selectable averaging filter

In general, when an image is transformed, each pixel in the image has its color data resampled by using pixel values around it. Points projected into the panorama space are also processed using a similar technique to improve the quality of the range data. However, general resampling techniques such as nearest interpolation reduce the quality of the range data because valid, erroneous, and missing data are blended in the resampling. Therefore, a focused pixel-selectable averaging filter is applied to this problem. The filtering processing uses only valid pixels around a pixel for the resampling. This processing is equivalent to missing-point regeneration without reducing geometrical accuracy by a uniform smoothing effect. For example, a block of three by three pixels in the range data set is prepared. The center point in the block is the focus point in the range data set. When the focus point is a missing point and the other points are valid points in the point cloud, a new pixel value is given to the focus point using the other points. This processing is applied to each channel in the RGB image and to an intensity image.
Thus, this processing uses only valid points in the range data. The detailed flow of the pixel-selectable averaging filter is described as follows. First, the data are checked to see whether or not valid points exist. When more than one pixel in a three by three block exists, including the focus point and the points around it, processing proceeds to the next step. If no point exists in the block, the next processing step is omitted.
Second, the number of valid pixels in the block is counted. When there are more than two valid pixels, the processing proceeds to the next step. If there is only one valid pixel and the extracted point is the focus point, it is deleted as spike noise. If there is only one valid pixel and the extracted point is not the focus point, the focus point remains without the filtering processing.
Third, after these point-extraction steps, a range of search in distances is given to extract valid points. The start value of the search range is the distance from the viewpoint to the nearest point found among the extracted points, while the end value is the start value plus a defined distance parameter. Then, all valid points in the block within the search range are extracted. The defined distance parameter depends on the continuity of the points in the point cloud. For example, when the measurement scale is 100 m to 500 m, the defined parameter would be 1 m to 10 m from empirical experience.
Finally, an average distance value from the viewpoint to the valid points is calculated. Then, the focus point value is overwritten by the average value. However, when the focus point has a distance value within the search range, the point is defined as the nearest surface approximately, and the overwriting processing is not performed in this case. This processing sequence is applied to all points.

3. EXPERIMENT

We conducted experiments using a point cloud taken from a terrestrial laser scanner VZ-400 produced by RIEGL Laser Measurement Systems GmbH. This scanner can acquire panoramic distance data and corresponding color data over 360° in the horizontal direction. The test field was a park that has good visibility, including flat ground planes, trees, and puddles. The input data set consisted of 2.9 million points after 3 cm^3 spatial filtering.

3.1 Result

Part of a panorama image set as 0.2° spatial angle resolution after a viewpoint translation from the sensor point to a point 2.5 m distant is shown in Figure 3, and part of the panorama image after the viewpoint translation and iterative filterings (three times) using the panorama imagery is shown in Figure 4. The average processing time for the panorama image conversion and iterative filterings (three times) by parallel programming in C was 0.14 s (Intel Core i7 (2.80 GHz, eight threads)) for one-shot generation in multiple viewpoints without file I/O.

Figure.3 Part of the panorama image generation result after a viewpoint translation

Figure.4 Part of the panorama image generation result after a viewpoint translation and filtering

3.2 Discussion

We can verify the features of our LiDAR VR generation as follows. The first feature is panorama-image generation with an arbitrary viewpoint. We have verified that the proposed methodology can successfully generate LiDAR VR images from many viewpoints automatically with one observation data set from one viewpoint. Moreover, we have verified that camera movement and rotation from a viewpoint can be simulated successfully to generate a new image without a position sensor such as RTK-GPS or a direction sensor such as a digital compass. In addition, we have confirmed that continuous camera trajectory simulation can achieve LiDAR VR imagery generation from accurate viewpoints. The second feature is real-time spatial interpolation using the point cloud. We have confirmed that spatial interpolation with rendering can be processed in real time (approximately 7 fps with parallel processing) including panorama-image generation. First, a viewpoint translation simulates a new LiDAR VR image with some missing point data as an interleaved pixel array. Next, the interpolation processing for missing points in the panorama space generates new point data from the new viewpoint. Thus, almost all of the missing data are recovered after filtering. As a result, a higher-quality image has been generated, as shown in Figure 4. Finally, although the panorama projection requires a long processing time in proportion to the number of points, we have confirmed that the proposed overall processing can be achieved in real time. Moreover, even if missing points remain, they can be removed via two or more iterative filterings in most cases. However, larger translations of the viewpoint expose more occlusions in the point cloud. Therefore, even if iterative processing is applied to a generated image from a more distant viewpoint, the interleaved pixels may not be correct. However, this is not a critical problem because a common interpolation is undesirable in a case with data.

6. CONCLUSION

We have developed a point-based rendering application called LiDAR VR with high-speed spatial interpolation using a point cloud to generate VR data. This application has achieved not only a transformation from point cloud to VR data but also automatic regeneration of missing points in the rendered result from a chosen viewpoint. Moreover, we have focused on two features of the LiDAR VR generation. The first is that panorama imagery can be generated automatically from arbitrary viewpoints with one observation data set. The second is the possibility that spatial interpolation with rendering results can reduce the processing tasks. Finally, we have presented results that confirm these features in case studies.

References:

[1] Shenchang, E.C., 1995, QuickTime VR - An image-based approach to virtual environment navigation, Proceedings of ACM SIGGRAPH'95, pp. 29–38.

[2] Jason, D., 2000, An approach to virtual environments for visualization using linked geo-referenced panoramic imagery, Computers, Environment and Urban Systems 24, pp. 127-152.

[3] Edward, V., Sisi, Z. & Sander, D., 2005, Distance-value-added panoramic images as the base data model for 3D-GIS, Panoramic Photogrammetry Workshop.

[4] Lars, L., Karsten, M. & Paul R., 2007, Splat-based Ray Tracing of Point Clouds. Journal of WSCG, Vol.15, Issue: 1-3, pp. 51-58.

Acknowledgement: This work is supported by Strategic Information and Communications R&D Promotion Programme (SCOPE) of the ministry of internal affairs and communications, Japan.