Computer-Aided Design and Applications

http://www.cadanda.com

Extraction of Surface Primitives from Noisy Large-Scale Point-Clouds

H. Masuda[1] and Ichiro Tanaka[2]

[1]The University Tokyo, masuda@nakl.t.u-tokyo.ac.jp
[2]Tokyo Denki University, tanaka@cck.dendai.ac.jp

ABSTRACT

3D shape acquisition of large facilities, such as industrial plants and power plants, has been receiving increasing attention for simulating complicated maintenance and repair tasks. It is widely recognized that model-based planning based on 3D CAD reduces the rework of maintenance tasks to a large extent. The state-of-the-art phase-based laser scanners are very promising to efficiently capture point-clouds of large facilities; because they can produce hundreds of millions point data in several minutes. However, point data captured from the phase-based scanner tend to include large noise components and quite a lot of outliers. In addition, very large memory space is required to generate solid models by processing very large point-clouds. Therefore, it is preferable to develop techniques that can extract surface primitives, such as planes, cylinders, cones, spheres, and tori from very noisy and large-scale point-clouds. This paper introduces a robust streaming smoothing operator for extracting surface primitives. Our method is based on robust estimate and can be applied to hundreds of million point data. We applied our smoothing operator to point-clouds captured from nuclear power plants, and could successfully extract primitive surfaces from large-scale noisy point data.

Keywords: point-based modeling; smoothing; phase-based laser scanner; surface extraction.
DOI: 10.3722/cadaps.2009.387-398

1. INTRODUCTION

The surface reconstruction of real-world artifacts has become popular in a wide variety of applications. In recent years, 3D shape acquisition of large engineering facilities, such as industrial plants and power plants, has been receiving increasing attention for simulating complicated maintenance and repair tasks. It is widely recognized that model-based planning based on 3D CAD reduces the rework of maintenance tasks to a large extent. Several commercial CAD systems actually support such simulations. However, one of the biggest obstacles that prevent utilizing CAD systems is that old facilities lack not only 3D CAD models but also reliable drawings, because many engineering facilities were built one or more decades ago and have been repeatedly renovated in their long lifecycles. In such cases, it is very useful to generate 3D as-built models by measuring real facilities. Most parts in engineering facilities are industry standard components, which consist mostly of simple surfaces, such as planes, cylinders, cones, spheres, and tori. Therefore, the important thing is to precisely extract equations of such primitive surfaces and to determine parameters of standard components rather than to determine general NURBS surfaces.

For constructing as-built 3D models for large facilities, stereo-photogrammetry has been widely used. This technique generates 3D models using two or more photographic images taken from different positions. Common points are identified on each image and 3D coordinates are calculated to determine the sizes and positions of primitive objects. Although stereo-photogrammetry is a practical method, it requires long-term monotonous work for creating complete 3D models, when large facilities, such as power plants, are modeled.

Point-based modeling techniques have possibilities to construct 3D models of large facilities in short time at low cost. Point-based modeling has achieved a great success in reverse engineering of mechanical parts. In most cases, point-clouds are captured using the triangulation-based laser scanner, which can produce dense point-clouds with a high degree of accuracy. However, the triangulation-based laser scanner typically covers only within a range of a few meters and therefore is not suitable for measuring large facilities.

Two other types of laser scanners can be used for measuring large facilities. One is the time-of-flight scanner, which measures the round-trip travel time of the laser pulses. This type of scanner can measure in the range of a few hundred meters. However, it takes a lot of time to measure many points that cover large facilities, because the time-of-flight scanner must wait a round-trip laser pulse whenever it measures the coordinate of each point on the object. The other type is the phase-based laser scanner. This type of scanner radiates continuous modulated laser pulses and calculates distances using the phase difference between the emitted and received signals. The state-of-the-art phase-based laser scanner can measure objects up to 53 or 79 meters [1]. One great advantage of this type of scanner is that it can measure two hundred million points in 6 min 44 sec. These measuring range and speed are sufficient for most engineering facilities. The phase-based laser is considered most promising for as-built modeling of large facilities.

However, there are two major problems to be solved for extracting shape parameters. One is the processing of noise and outliers, and the other is the processing of enormous amount of point data. 3D point data acquired by the phase-based laser scanner include large noise components along the laser pulses (Fig. 1). While the directions of the pulses are precisely aligned mechanically, the measured distances are prone to errors. Since a certain rate of distance errors are greater than 6 mm, the surface of a mesh model often becomes "prickly," as shown in Fig. 1(c). As shown in this figure shows, point-cloud data from the phase-based scanner are much noisier than the data from the triangulation-based scanner. Region-growing [2, 3] is a typical surface extraction method for detecting planes, cylinders, cones, spheres and tori in point-clouds. This method starts from a small seed region and extends the boundary of the region by checking if neighbor points satisfy the surface equation. Since noisy data prevent to grow regions, it is highly required to smooth point data. Other segmentation techniques [4, 5] have also difficulties in the case of noisy data, because they require local differential properties.

The moving-least-squares (MLS) method is a popular tool for generating smooth surfaces [6–8]. The MLS projection calculates a locally smooth surface around each point and projects the point onto the smooth surface. However, MLS projection often fails to preserve geometric features, when outliers and high levels of noise are involved in point data. We applied the MLS method to many point-clouds data captured from the phase-based scanner, and we found that the MLS method failed to generate smooth data while preserving geometric features. This is because the phase-based scanner produces much more outliers and much larger noise components than the triangulation-based scanner.

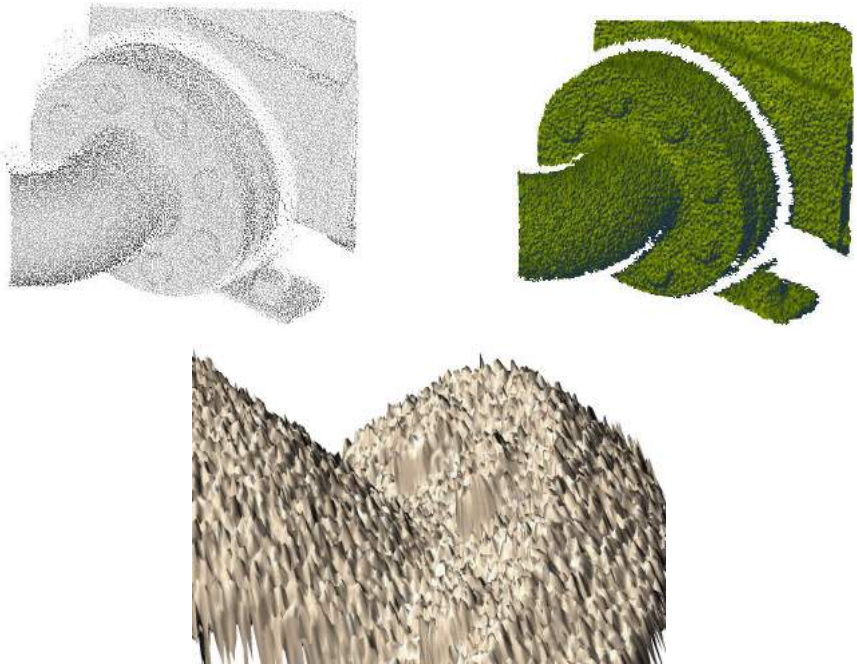


Fig. 1: Noisy mesh model constructed from point data captured by a phase-based scanner: (a) point data (top-left); (b) noisy mesh model (top-right); (c) close-up of the noisy mesh model (bottom).

Several researchers have reported on their efforts to control the parameters of the MLS method for improving the quality of smoothing [9-12]. However, it is not easy to apply these methods to point-clouds in which quite a lot of outliers are included. Fleishman et al. [13] discussed robust statistical methods and introduced the least median of squares metric. Lipman et al. [14] proposed a Locally Optimal Projection operator (LOP) based on the L1 median, which

is the absolute value of a distance function. These methods are robust to outliers, but they are very time-consuming and therefore inadequate to process hundreds of millions of points.

In this paper, we introduce a robust surface estimator based on M-estimate, and we smooth a large number of points in out-of-core manner. Then, we will show that geometric surfaces can be successfully extracted using a region growing method and their equations can be precisely determined. In the following Section, we will explain the phase-based scanner and out-of-core processing. Section 3 presents our smoothing method and shows some experimental results. In Section 4, we will explain surface extraction. Section 5 states conclusions.

2. POINT DATA FROM PHASE-BASED RANGE SCANNER

The phase-based laser scanner can capture hundreds of million points in several minutes. Fig. 2 shows an example of captured points of a power plant. Our current goal is to extract surface primitives from such point-clouds as precisely and stably as possible.

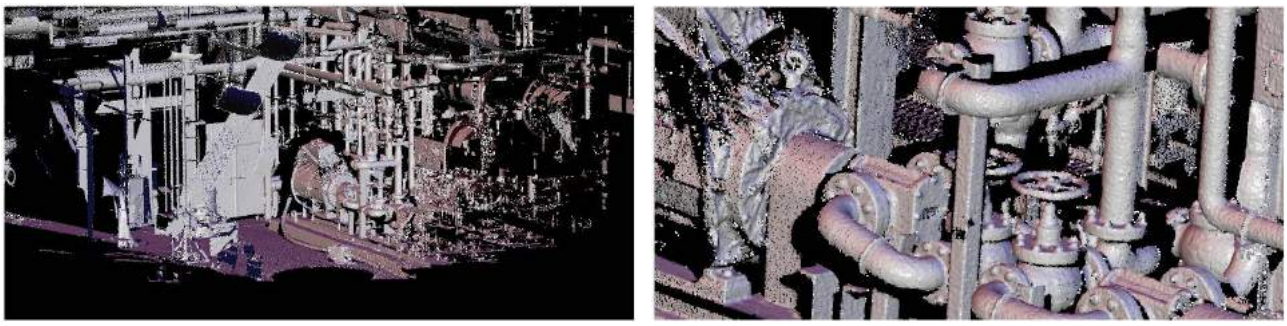
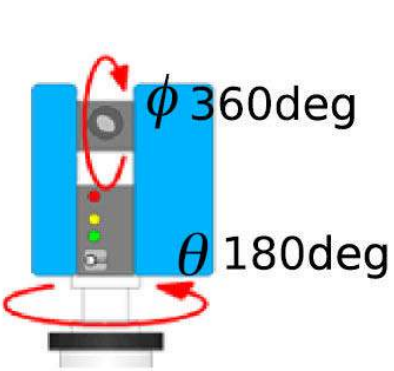


Fig. 2: Dense point-cloud data captured by the phase-based scanner.

Fig. 3(a) shows a phase-based laser scanner and its rotation angles [1]. The phase-based laser scanner emits laser pulses while rotating the direction of laser light along the latitude and longitude. Captured data are ordered along a latitude line, as shown in Fig. 3(b). All point data can be uniquely mapped onto a unit sphere centered at the origin of laser emission, as shown in Fig. 4. The brightness in this figure reflects the intensity of received signals.



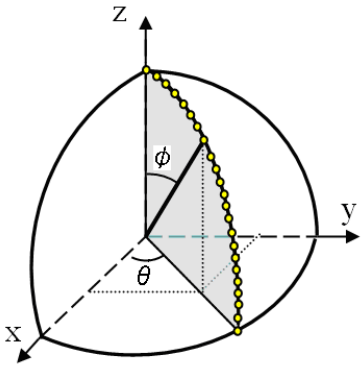


Fig. 3: Rotation angles and captured data of phase-based scanner. Fig. 4: Point data mapped onto a sphere.

It is convenient to represent each point in spherical coordinates (r,θ,ϕ), where r is the distance from the origin, θ is the longitude, and ϕ is the latitude. Fig. 5 shows a 2D image generated by developing the unit sphere to the $\theta-\phi$ plane. In the case of point-clouds from the phase-based scanner, point data can be easily converted to a mesh model by using the adjacency relationships on the $\theta-\phi$ plane. This is an advantage of this type of scanners. Fig. 1(b) shows such a mesh model.

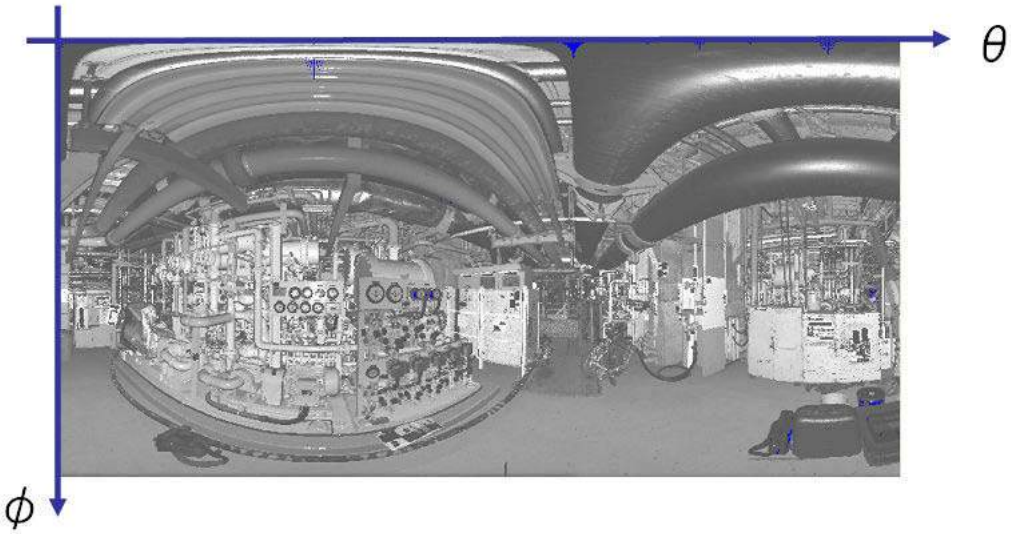


Fig. 5: Projection of point-cloud onto the $\theta - \phi$ plane.

In addition, the phase-based laser scanner outputs from tens of millions to hundred millions points in each scan. This means that the data size easily exceeds the capacity of ordinary 32-bit PCs. Despite this, it is not a good choice to reduce the number of captured points, because statistical processing is required to estimate the real positions of measured points, as discussed later. For precisely calculating geometric properties, out-of-core methods are preferable to process whole point-clouds captured by the phase-based scanner.

Fortunately, out-of-core streaming processes are easily realized in this case, because point data captured by the phase-based scanner are coherently ordered in files, as shown in Fig. 6. Since our algorithms require only a certain range of neighbor points, all we have to do is to load points in the sufficient number of lines in Fig. 6, and only maintain them in RAM. This simple method enables efficient streaming processing for hundreds of million points.

Fig. 6: The data coherency that can be used for streaming processing.

3. SMOOTHING NOISY POINT-CLOUDS

3.1 MLS Projection

MLS projection [6,7] is a popular tool for calculating smooth surfaces from noisy point sets. This method approximates the neighborhood of a point by a local polynomial surface and projects the point onto the local surfaces. However, in our experiments, conventional MLS projection is not necessarily suitable for point-clouds captured by the phase-based scanner. Here, we will briefly explain MLS projection and show point data to which MLS projections are applied.

In the first step of MLS projection, a local reference plane $\mathbf{n}_i^t(\mathbf{x} - \mathbf{c}_i) = 0$ is fitted to the neighborhood of point $\mathbf{p}_i$ by minimizing:

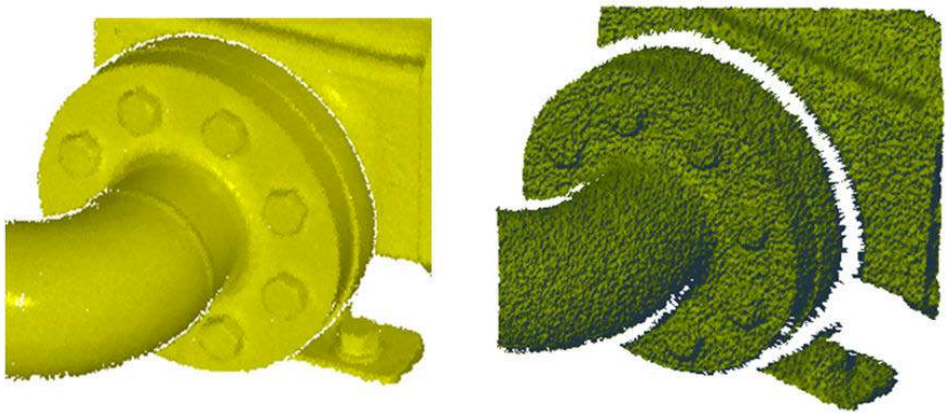
$$\sum_{j\in N_i}\left(\mathbf{n}_i{}^t\mathbf{p}_j-\mathbf{n}_i{}^t\mathbf{c}_i\right)^2\chi(\|\mathbf{p}_j-\mathbf{c}_i\|), \tag{3.1}$$

where N_i is an index set of neighboring points, $\mathbf{n}_i$ is the normal vector of the reference plane, and χ is a monotonically decreasing function. In the next step, the neighborhood points $\{\mathbf{p}_j\}$ are projected onto the reference plane and they are parameterized as (u_j,v_j) on this plane. Then quadratic function $g(u,v)$ is calculated by minimizing:

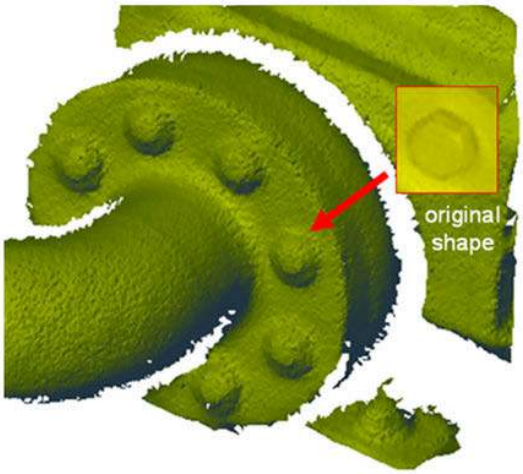
$$\sum_{j\in N_i}(g(u_j,v_j)-f_j)^2\chi(\|\mathbf{p}_j-\mathbf{c}_i\|), \tag{3.2}$$

where f_j is the distance between the reference plane and point $\mathbf{p}_j$. Finally, the projection of $\mathbf{p}_i$ is calculated as $\mathbf{c}_i+g(0,0)\mathbf{n}_i$.

We applied this method to point data captured by the phase-based scanner. Fig. 7 shows a result of MLS projection. Fig. 7(a) is a brightness image of a target part. Fig. 7(b) is the input noisy mesh model, which was generated using adjacency relationships on the $\theta-\phi$ plane. Fig. 7(c) is a mesh model to which MLS operation is applied. Apparently, planar regions are bumpy and sharp edges of bolts are seriously damaged. We tested several point-clouds of large facilities, but MLS operators failed to generate smooth and feature-preserving results in all cases.



(a) Brightness image. (b) Noisy mesh model.



(c) Mesh model smoothed by MLS projection.

Fig. 7: An experimental result of MLS operation.

3.2 Surface Estimation Based on M-Estimate

For robustly smoothing noisy point-clouds, we first assume probability density functions for the distribution of noise components and then calculate the maximum likelihood surfaces. This approach can be formalized as M-estimate [15-

17]. While the conventional MLS projection implicitly assumes the normal distribution, which is sensitive to large noises and outliers, we design a new smoothing operator on the basis of more robust distribution functions.

Let points $\mathbf{x}_i \in \mathbb{R}^3$ $(1 \le i \le N)$ be sampled from a smooth surface. N is the number of neighborhood points. $S(\mathbf{x} \mid \mathbf{a}) = 0$ is an equation of a smooth surface, where $\mathbf{a} = \{a_k\}$ $(1 \le k \le M)$ denotes parameters of the smooth surface and M is the number of parameters.

Our goal is to estimate the most likely parameters of the surface. We represent the residual of point $\mathbf{x}_i$ as:

$$r_i = \frac{S(\mathbf{x}_i \mid \mathbf{a})}{\sigma} \qquad (1 \le i \le N), \tag{3.3}$$

where σ is the standard deviation of the values of $S(\mathbf{x}_i \mid \mathbf{a})$. Given the probability density function $f(r)$, the maximum likelihood estimate $\hat{\mathbf{a}}$ can be calculated as:

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} \prod_{i=1}^{N} \left[f(r_i) \right], \tag{3.4}$$

or equivalently,

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \sum_{i=1}^{N} \rho(r_i), \tag{3.5}$$

where $\rho(r_i) = -\log\left(f(r_i)\right)$. When $\rho(r)$ is differentiable, $\hat{\mathbf{a}}$ can be solved by:

$$\frac{\partial}{\partial a_k}\left(\sum_{i=1}^{N} \rho(r_i) \right) = \frac{1}{\sigma} \sum_{i=1}^{N} w_i(r_i) \left(\frac{\partial S(\mathbf{x}_i \mid \mathbf{a})}{\partial a_k} \right) = 0, \tag{3.6}$$

where

$$w_i(r) \equiv \frac{\partial \rho}{\partial r}(r) \tag{3.7}$$

is a weighting function of point $\mathbf{x}_i$.

Various functions can be assumed as the probability density function $f(r)$. Fig. 8 shows three well-known distributions. The tail of the Lorentzian distribution is the longest among the three distributions. Roughly speaking, a longer tail means higher robustness against outliers.

When $f(r)$ is the normal distribution, ρ is represented as $\rho(r) = r^2$. In this case, the maximum likelihood is equivalent to the result of least-squares estimation. The weighting function is represented as $w_i(r) = 2r$, which implies that a large residual results in a large weight. This is why the least-squares estimator is easily biased by outliers with large residuals. When $f(r)$ is the double exponential distribution, function ρ is $\rho(r) = |r|$ and the weight is $w_i(r) = \pm 1$, which implies that all points have the same weights regardless of the size of the residuals. This estimator is more robust than the least-squares estimator.

The Lorentzian distribution provides an even more robust function:

$$\rho(r) = \log(1 + r^2), \tag{3.8}$$

and the weight is:

$$w_i(r) = \frac{2r}{1 + r^2}. \tag{3.9}$$

This weight implies that larger residuals have smaller weights, and therefore outliers will have no practical impact on the estimation of parameters. Therefore, it seems that the Lorentzian distribution is the most suitable for our purpose in the three distribution functions. Then we derive projection operators based on the Lorentzian distribution, and evaluate its quality using real data sets.

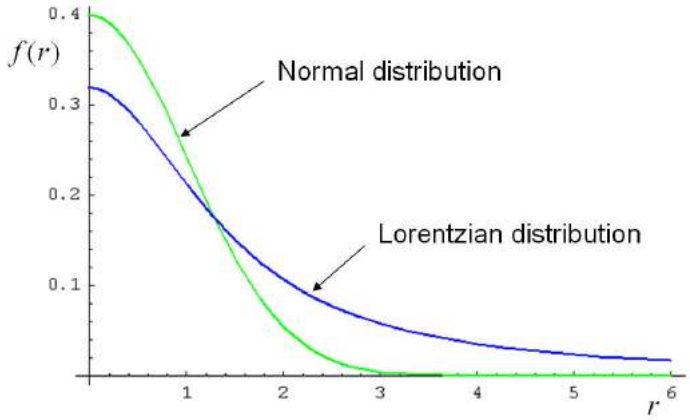


Fig. 8: Examples of probability density functions.

3.3 Robust Surface Smoothing Operation

Let $S_i(\mathbf{x} \mid \mathbf{a}) = 0$ be the equation of a quadric polynomial surface, which is used to locally approximate the neighborhood of point $\mathbf{p}_i$. We introduce a new MLS-like surface estimator based on robust estimate. The smooth surface is then calculated as:

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \left(\sum_{j \in N_i} \rho\left(\frac{S_i(\mathbf{p}_j \mid \mathbf{a})}{\sigma_i} \right) \chi(\| \mathbf{p}_j - \mathbf{c}_i \|) \right), \tag{3.10}$$

where σ_i is the standard deviation of $S_i(\mathbf{x}_i \mid \mathbf{a})$; N_i is the neighborhood of $\mathbf{p}_i$; $\mathbf{c}_i$ is a reference point; and χ is a monotonically decreasing function typically defined by the Gaussian:

$$\chi(d) = e^{-\frac{d^2}{h^2}}, \tag{3.11}$$

where h is a scaling parameter reflecting the spacing between neighboring points. Parameter h can be used to control the level of smoothness. Reference point $\mathbf{c}_i$ is a likely estimate of the position of point $\mathbf{p}_i$. This point is calculated by fitting a plane to the neighborhood of $\mathbf{p}_i$ and projecting $\mathbf{p}_i$ onto the plane.

When the normal distribution is assumed, the maximum likelihood surface can be calculated using $\rho(r) = r^2$ as:

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \left(\sum_{j \in N_i} \frac{S_i(\mathbf{p}_j \mid \mathbf{a})^2}{\sigma_i^2} \chi(\| \mathbf{p}_j - \mathbf{c}_i \|) \right). \tag{3.12}$$

For the Lorentzian distribution, the maximum likelihood surface can be calculated using $\rho(r) = \log(1 + r^2)$ as:

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \left(\sum_{j \in N_i} \log\left\{ 1 + \frac{S_i(\mathbf{p}_j \mid \mathbf{a})^2}{\sigma_i^2} \right\} \chi(\| \mathbf{p}_j - \mathbf{c}_i \|) \right). \tag{3.13}$$

3.4 Experimental Results

Given error-free data, these two types of optimizations, each of which is based on $\rho(r) = r^2$ and $\rho(r) = \log(1 + r^2)$, obviously produce the same results. However, they produce quite different smoothed results for noisy point data.

Eqns. (3.12) and (3.13) can be solved by nonlinear optimization techniques, such as the Levenberg-Marquardt and Gauss-Newton methods [18]. After quadric polynomial surfaces are obtained, point $\mathbf{p}_i$ is replaced to the projected point on the surface. Fig. 9 shows a comparative evaluation of an MLS projection and the Lorentzian surface estimator (LSE). We evaluated two operators using various parameters for h, which is defined in Eqn. (3.11). The value of δ in Fig. 9 is the average spacing between neighboring points. Clearly, the method developed here produced better results from the standpoint of eliminating noise and preserving features. MLS operators caused artifacts on the surfaces and incorrectly smoothed out the corners of the object. The close-ups of the smoothed models are shown in Fig. 10.

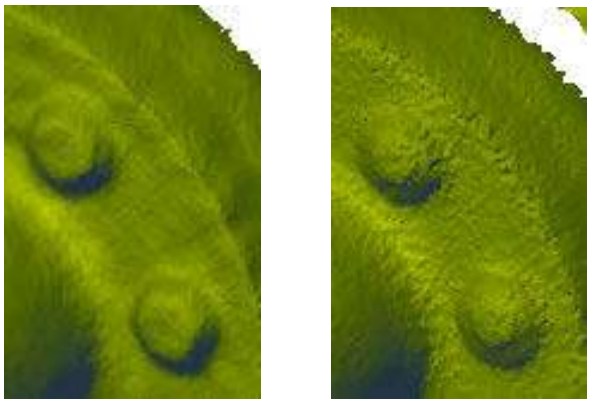
$h = \delta$ $h = 2\delta$ $h = 4\delta$

(a) Lorentzian Surface Estimator

$h = \delta$ $h = 2\delta$ $h = 4\delta$

(b) MLS Projection Operator

Fig. 9: Comparison of the Lorentzian surface estimator and the MLS projection operator.



(a) LSE (b) MLS

Fig. 10: Close-ups of smoothed models ($h = 4\delta$).

3.5 Streaming Processing

Fig. 11 illustrates how to process large-scale point data. Point data are almost coherently ordered, but there are some missing points and un-arrayed points, as shown in Fig. 11. Therefore, we segment point data on the $\theta - \phi$ plane into long rectangular strips, as shown in Fig. 6, and we convert point data in each strip into a mesh model. Fig. 11 shows three strips. Each strip is sequentially triangulated using the constrained Delaunay triangulation. In Fig. 11, Strip-1 is first triangulated without constraints, and then Strip-2 is triangulated using the boundary edges of Strip 1 in the constrained Delaunay triangulation. The boundary edges of Strip-2 are also used as the constraints for Strip-3. When the process of Strip 2 is finished, point data in Strip 1 are eliminated from RAM.

The mesh data of each strip are used as the input to the Lorentzian surface estimator, which detects neighbor points and smoothes noisy points. Fig. 12 illustrates a pipeline for generating a smooth mesh model from large-scale point cloud. This pipeline allows processing large-scale point data in a streaming manner. In this pipeline, the most time-consuming task is to solve the non-linear optimization for Lorentzian surface estimator. Since the smoothing of each point can be calculated independently, the smoothing process can be parallelized on multi-core CPUs.

Fig. 13 shows an example of large-scale point-clouds. This data was captured by a single scan of the phase-based scanner. The number of points is about fifty million. The smoothed result was calculated using a streaming process. CPU time for generating this large smooth mesh model was about 13 min on the Intel 2.66GHz quad-core CPU. Our current implementation is parallelized for multi-core CPUs and the performance is nearly proportional to the number of processor cores.

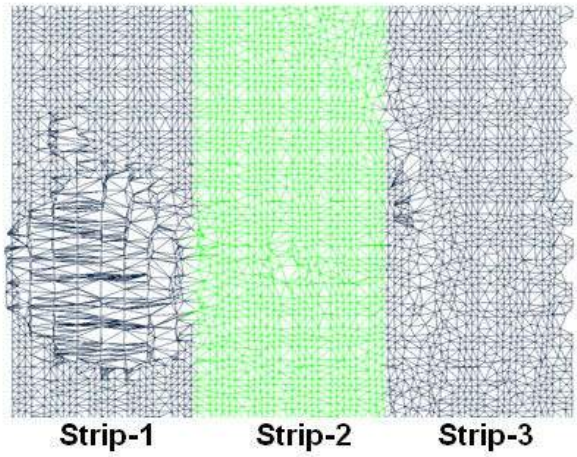


Fig. 11: Strip-based mesh generation.

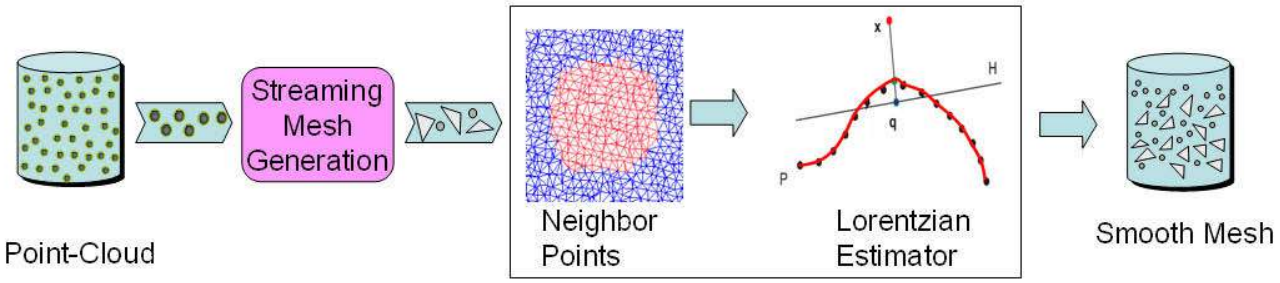


Fig. 12: Pipeline to generate a smooth large-scale mesh model.

4. SURFACE EXTRACTION

In the next step of processing, primitive surfaces are extracted from smoothed point-cloud data. We extract regions of surface primitives by the following steps:

(1) A mesh model is generated from the smoothed point-cloud data using streaming processing.
(2) The $\theta - \phi$ plane is subdivided into small rectangle regions.
(3) A primitive surface is fitted to points in a rectangle region. If fitted, the region is used as a seed of region growing.
(4) The seed region repeatedly grows if adjacent points fit to the primitive surface.

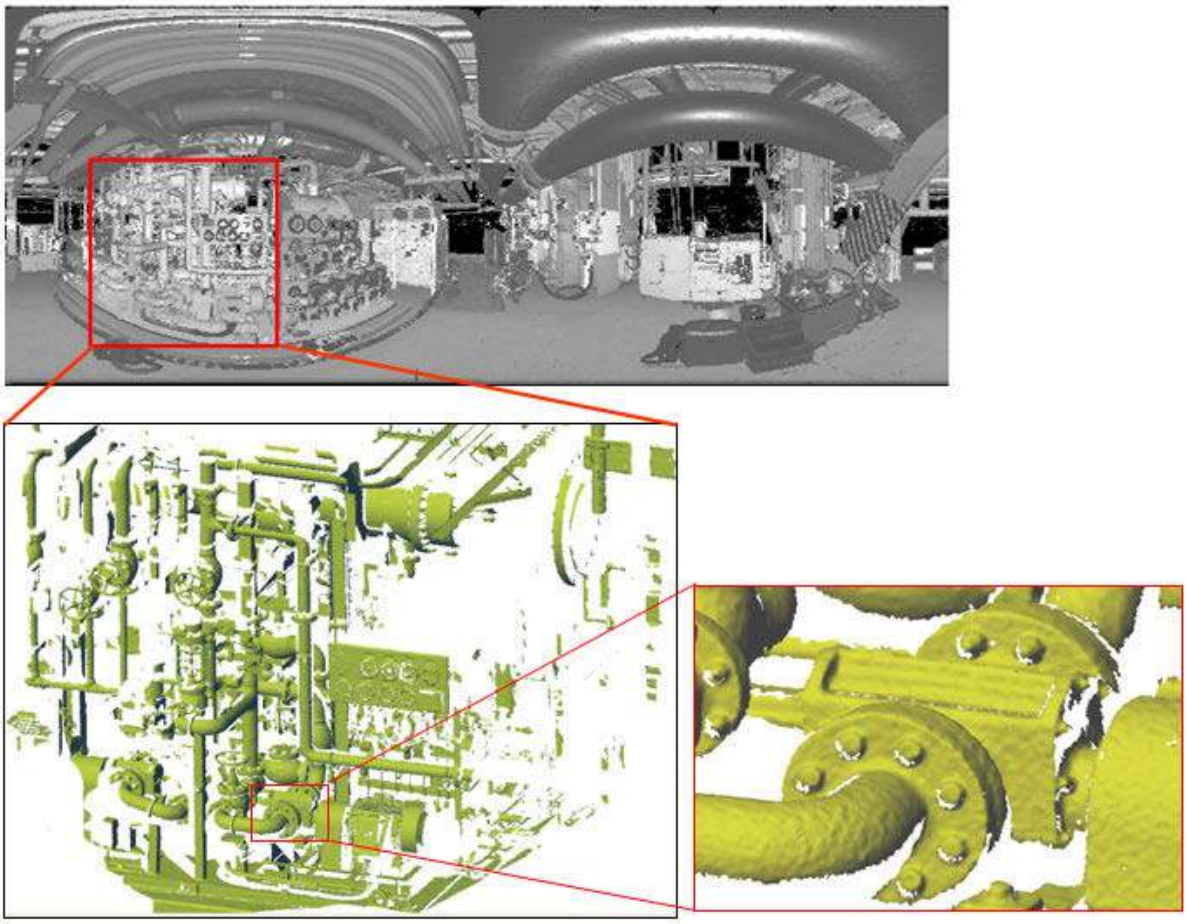


Fig. 13: A smoothed mesh model and its close-ups.

It is important to calculate surface equations as faithfully and precisely as possible. We used the fitting method proposed by Lukacs, et al [9]. This method represents the distance function between primitive surface $S(\mathbf{x} \mid \mathbf{a}) = 0$ and point $\mathbf{p}_i$ as the following form:

$$d(\mathbf{a}, \mathbf{p}_i) = \sqrt{g(\mathbf{a}, \mathbf{p}_i)} - h(\mathbf{a}, \mathbf{p}_i), \tag{4.1}$$

where g and h are functions of parameters $\mathbf{a}$ and coordinate $\mathbf{p}_i$. For example, the distance from a sphere can be described as $g(\mathbf{a}, \mathbf{p}_i) = (x_i - a)^2 + (y_i - a)^2 + (z_i - a)^2$ and $h(\mathbf{a}, \mathbf{p}_i) = r$, where (x_i, y_i, z_i) is a coordinate of $\mathbf{p}_i$; (a, b, c) is the center of the sphere; r is the radius of the sphere. Lukacs, et. al. described all primitive surfaces in this form and solved surface parameters using the following minimization.

$$\sum \tilde{d}(\mathbf{s}, \mathbf{p}_i)^2 = \sum \{(g - h^2)/2h\}^2 \rightarrow \min. \tag{4.2}$$

$\tilde{d}(\mathbf{s}, \mathbf{p}_i)$ has the same derivative values as $d(\mathbf{s}, \mathbf{p}_i)$ when $d(\mathbf{s}, \mathbf{p}) = \tilde{d}(\mathbf{s}, \mathbf{p}) = 0$. According to [9], the minimization of this function produces more stable and faithful fitting results than the ones of $\sum (g - h^2)^2$ and $\sum \left(\sqrt{g} - h\right)^2$.

Since Eqn. (4.2) is a non-linear optimization, good initial values are required. In our method, normal vectors can be calculated at each vertex of mesh models. We estimate initial values of primitive surfaces using positions and normals of smoothed points. For example, Fig. 14 shows how to estimate the initial parameters of a cylinder. Since the axis $\mathbf{v}$ of a cylinder is perpendicular to normal vectors, $\mathbf{v}$ can be calculated as the minimization of

$\sum(\mathbf{v},\mathbf{n}_i)^2$, where $\mathbf{n}_i$ is the normal vector at point $\mathbf{p}_i$. Then each normal vector is projected on a plane that is perpendicular to the axis, and the center is calculated as the intersection point of the lines in the right figure of Fig. 14. The initial parameters of other primitive surfaces can be similarly calculated using the positions and normal vectors of smoothed points.

Fig. 15 shows an example of surface extraction. In this figure, automatically detected surfaces are rendered in different colors, and cylinders are shown in wireframes. The right figure of Fig. 15 shows that a cylinder was reasonably fitted to the original point set, even when the cylinder was only partially sampled. This example indicates that our method can stably extract primitive surfaces from very noisy point-cloud data captured by the phase-based scanner.

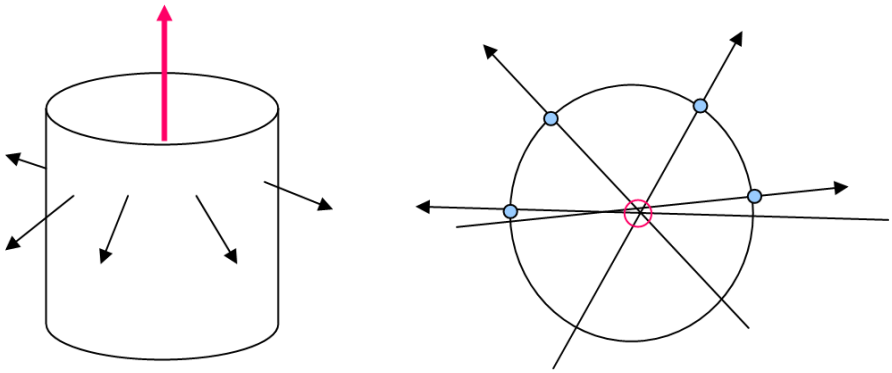


Fig. 14: Estimation of initial parameters of a cylinder.

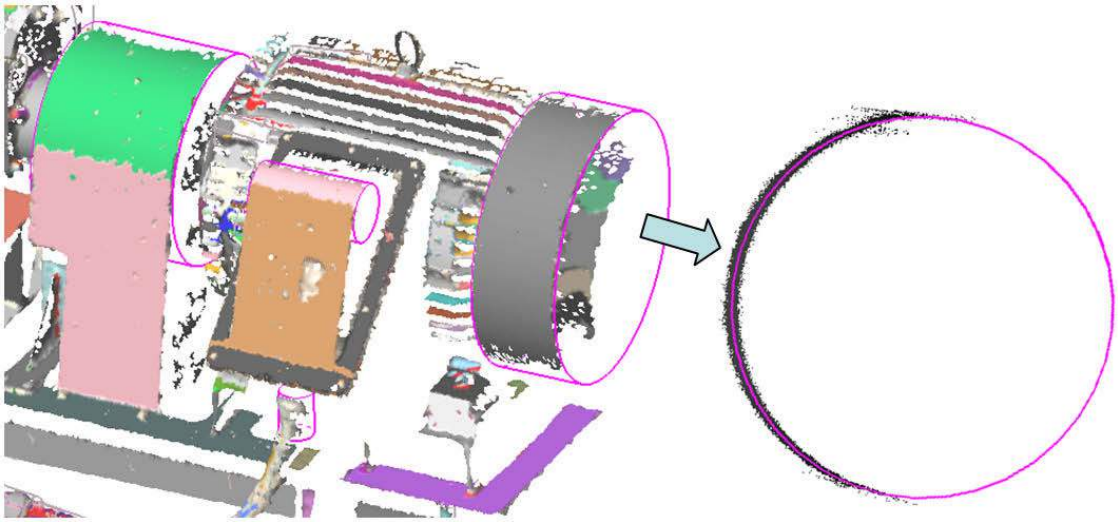


Fig. 15: Detected primitive surfaces. (Right: original noisy points and detected cylinder.)

6. CONCLUSIONS

The state-of-the-art phase-based scanner produces hundreds of millions point data by a single scan. Such point data tend to include large noise components and quite a lot of outliers. In this paper, we extracted primitive surfaces from such point data. For managing noise and outliers, we introduced a new robust smoothing operator based on the Lorentzian estimate. We applied the smoothing operator to point data captured from the phase-based scanner and showed that the Lorentzian surface estimator worked better than the conventional MLS projection. Since our smoothing operator allows streaming and parallelized calculations, large point-clouds, which sometimes exceed the limit of 32-bit PCs, could be efficiently processed on ordinary PCs. In addition, we showed

our method could be used as a good pre-process for extracting primitive surfaces from large noisy point-cloud data.

In future work, we would like to identify distribution functions of errors in captured data. In addition, it would be desirable to improve the performance of the optimization processes, because it is still time-consuming to process hundreds of millions of points. Finally, to construct a whole 3D model of a large facility, it is necessary to merge point data captured from several positions. There is therefore a need to investigate registration methods for very large and noisy point data.

7. ACKNOWLEDGEMENTS

The authors would like to thank Kenji Murakami and Satoshi Fujii for helping the development of the system. This research was partially supported by the Ministry of Education, Science, Sports, and Culture, Grant-in-Aid for Scientific Research (B), 17360414, 2007.

8. REFERENCES

[1] Technical Data IMAGER, http://www.zf-laser.com, Zoller-Fröhligh.
[2] Vieira, M.; Shimada, K.: Surface mesh segmentation and smooth surface extraction through region growing, Computer Aided Geometric Design, 22, 2005, 771–792.
[3] Yang, M.; Lee, E.: Segmentation of measured point data using a parametric quadric surface approximation, Computer-Aided Design, 31(7), 1999, 449–457.
[4] Mangan, A. P.; Whitaker, R. T.: Partitioning 3D surface meshes using watershed segmentation, IEEE Transactions on Visualization and Computer Graphics. 5(4), 1999, 308–321.
[5] Shami, A.: A survey on mesh segmentation techniques, Computer Graphics Forum, 27(6), 2008, 1539-1556.
[6] Alexa, M.; Behr, J.; Cohen-Or, D.; Fleishman, S.; Levin, D.; Silva, C.T.: Point set surfaces, IEEE Visualization, 2001, 21–28.
[7] Levin, D.: Mesh-independent surface interpolation, Geometric Modeling for Scientific Visualization, 2003, 37–49.
[8] Amenta, N.; Kil, Y.J.: Defining point-set surfaces, ACM Transaction on Graphics 23(3), 2004, 264–270.
[9] Pauly, M.; Mitra, N.J.; Guibas, L.J.: Uncertainty and variability in point cloud surface data, Proceedings, Eurographics Symposium on Point-Based Graphics, 2004, 77-84.
[10] Schall, O.; Belyaev, A.; Seidel, H.-P.: Robust filtering of noisy scattered point data, Proceedings, Eurographics Symposium on Point-Based Graphics, 2005, 71–77.
[11] Reuter, P.; Joyot, P.; Trunzler, J.; Boubekeur, T.; Schlick, C.: Surface reconstruction with enriched reproducing kernel particle approximation, Proceedings, Symposium on Point-Based Graphics, 2005, 79-87.
[12] Lipman, L.; Cohen-Or, D.; Levin, D.: Data-dependent MLS for faithful surface approximation, Proceedings, Eurographics Symposium on Point-Based Graphics, 2007, 59-68.
[13] Fleishman, S.; Cohen-Or, D.; Silva, C. T.: Robust moving least-squares fitting with sharp features, ACM Transactions on Graphics, 24(3), 2005, 544–552.
[14] Lipman, Y.; Cohen-Or, D.; Levin D., and Tal-Ezer, H. T.: Parameterization-free projection for geometry reconstruction, ACM Transactions on Graphics, 26(3), 2007, Article No. 22.
[15] Huber, P. J.: Robust Statistics. Wiley Series in Probability and Statistics, Wiley-Interscience, 2003.
[16] Stewart, C. V.: Robust parameter estimation in computer vision, SIAM Review, 41(3), 1999, 513–537.
[17] Wheeler, M. D.; Ikeuchi, K.: Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(3), 1995, 252–265.
[18] Nocedal, J.; Wright, S.J.: Numerical Optimization, Second Edition, Springer, 2006.
[19] Lukacs, G.; Marshall, A. D.; Martin, R. R.: Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. Proceedings, 5th European Conference on Computer Vision, 1998, 671–686.