

Cylinder Detection in Large-Scale Point Cloud of Pipeline Plant

Yong-Jin Liu, *Member, IEEE*, Jun-Bin Zhang, Ji-Chun Hou, Ji-Cheng Ren, and Wei-Qing Tang

Abstract—The huge number of points scanned from pipeline plants make the plant reconstruction very difficult. Traditional cylinder detection methods cannot be applied directly due to the high computational complexity. In this paper, we explore the structural characteristics of point cloud in pipeline plants and define a structure feature. Based on the structure feature, we propose a hierarchical structure detection and decomposition method that reduces the difficult pipeline-plant reconstruction problem in \mathbb{R}^3 into a set of simple circle detection problems in \mathbb{R}^2 . Experiments with industrial applications are presented, which demonstrate the efficiency of the proposed structure detection method.

Index Terms—3D reconstruction, point cloud, structure analysis

1 INTRODUCTION

VIRTUAL 3D scene reconstruction with the aid of laser scanning has attracted considerable attention in computer vision and computer graphics. State-of-the-art applications have been exploited, such as building and city modelings using large-scale point sets [16]. In our study, we refer to *large scale* if the number of points reaches the magnitude of 10^6 - 10^9 . We present a structure detection method in this paper for digital plant reconstruction from large-scale point clouds. Below we first present the background and motivation of this research project.

Plants in petrochemical industry widely exist. Three-dimensional digital plant models are important in training, fire drill and disaster simulation using virtual reality techniques. Although modern plants can be designed and constructed directly using 3D modeling software such as AutoCAD Plant 3D, many existed plants have to be digitally reconstructed from physical environment. In a joint project of Tsinghua University and Zhongke Fulong Computer Technology Co., Ltd, we scan the plant using a Trimble GX200 laser scanner (see Fig. 1). Since petrochemical plants usually consist of a large number of pipelines, fast and accurate detection of these pipelines in large-scale point cloud is the core of this project.

Pipelines are modeled by cylinders. Cylinder detection in point cloud has been extensively studied; some representative works include [2], [4], [12], [13], [18], [20]. Broadly these methods can be classified into three classes: 1) region growing, 2) RANSAC, and 3) Hough transform. These

methods cannot be directly applied in our application, mainly due to two characteristics of pipeline plant scanning:

- C1. There are hundreds of pipelines of different radii and lengths that need to be detected in millions of points.
- C2. The scanned points are not accurate and contain noise.

The detailed explanation of these two characteristics and why existing methods fail in this case is presented in the next section.

In this work, we exploit the structural characteristics in pipeline plant scanning and propose a structure detection and decomposition method. The resulting decomposition efficiently partitions the large-scale point cloud into small subsets, each of which contains a few pipelines of the same orientation. By projecting points in each subset to the plane perpendicular to its related orientation, the difficult problem of detecting a large number of pipelines in large-scale point cloud is reduced to a set of easy subproblems, each of which needs only detect circles in the projected plane with a small subset of point data.

2 PROBLEM IDENTIFICATION AND RELATED WORK

In the described project, we use Trimble GX200 to scan the plant. As a typical example, we scanned a PTA plant in SINOPEC Beijing Yanshan Company by manually partitioning the plant into five divisions with 257 sites. At each site, a $360^\circ \times 120^\circ$ field of view was scanned. After registering and merging all scanning data and cleaning up using RealWork Survey software, we obtained totally 104,011,586 points, in which there were 1,100 pipelines. Fig. 2 shows a subset of this data. This subset has 5,234,852 points, containing one air chimney, two oil tanks, 310 oil pipelines of different sizes. We model all these primitives by cylinders and Table 1 summarizes their statistic data. Observed from Table 1, the pipelines have different numbers of points ranged from 10^3 to 10^5 (characteristic C1).

The Trimble GX200 scanner has a high scanning accuracy of tolerance 1.4 mm at 30 m. However, the pipelines are

• Y.-J. Liu, J.-B. Zhang, and J.-C. Hou are with the TNList, Department of Computer Science and Technology, Tsinghua University, Room 3-530, FIT building, Haidian District, Beijing 100084, P.R. China.
E-mail: liuyongjin@tsinghua.edu.cn.

• J.-C. Ren and W.-Q. Tang are with Beijing Zhongke Fulong Computer Technology Co., Ltd, Floor A-9, Information Mansion, No. 28, Information Road, Shangdi, Haidian District, Beijing 100085, P.R. China.

Manuscript received 20 Oct. 2012; revised 18 Feb. 2013; accepted 15 Apr. 2013; published online 18 Apr. 2013.

Recommended for acceptance by P. Cignoni.

For information on obtaining reprints of this article, please send e-mail to: tcvg@computer.org, and reference IEEECS Log Number TVCG-2012-10-0234. Digital Object Identifier no. 10.1109/TVCG.2013.74.

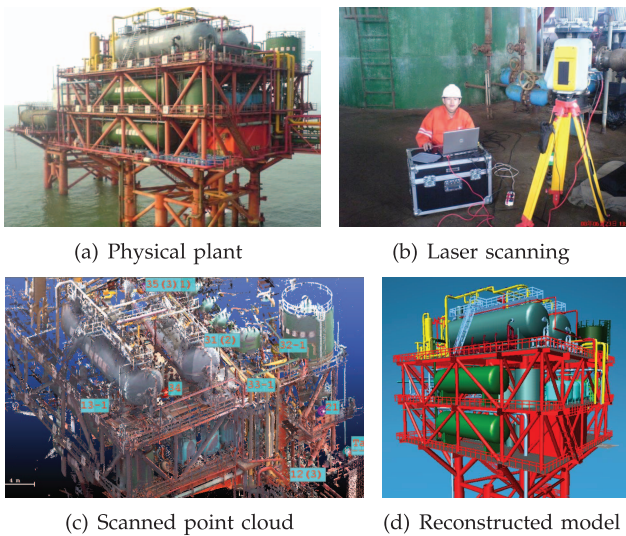


Fig. 1. Three-dimensional digital pipeline plant reconstruction.

seldom naked and usually wrapped up by electronic or thermal insulating layers. Thus, the scanned data is not accurate (characteristic C2). If we use neighborhood information to estimate point normal or high-order derivatives, these data also contain noise. Given the characteristics C1 and C2, the traditional region growing methods (e.g., [13], [20]) that start from some random seeds cannot be applied, since the pipelines may be adjacent to each other very closely.

A widely accepted primitive (including cylinders) detection method is the random sample consensus (RANSAC) [2], [8]. Many RANSAC-based cylinder detection methods (e.g., [2], [4], [19]) directly use point normal to determine cylinder parameters. Since point normals are noised in our application, we cannot directly use point normals to estimate the cylinder parameters. Unlike the planar circles that can be uniquely determined by three planar points in general positions, it is well known in computational geometry that it is difficult to determine how many points are sufficient to determine an arbitrary cylinder in \mathbb{R}^3 . It was proved in [7] that there are at most six cylinders through five points in \mathbb{R}^3 . If we arbitrary select $k = 5$ points from a given point set S to determine a cylinder's parameters and set the probability of finding a correct cylinder larger than 90 percent, the number of random selections for each cylinder in Fig. 2 is summarized in Table 1.

The huge number of times of random selections in Table 1 demonstrate that RANSAC cannot be directly used in our application due to the large-scale point set and the large number of cylinder primitives of diverse sizes (characteristic C1). Several fast versions of RANSAC variants have been proposed recently. The $T_{d,d}$ test [14] and the optimal randomized RANSAC [15] speed up the hypothesis evaluation procedure by only using subsets of the whole point set. The bail-out test [3] uses the assumption that if the probability of "the currently evaluated hypothesis contains as many inliers as the so-far-the-best hypothesis" is less than a predefined threshold, the hypothesis can be discarded earlier before the whole set is evaluated. All these methods are suitable for the scenario

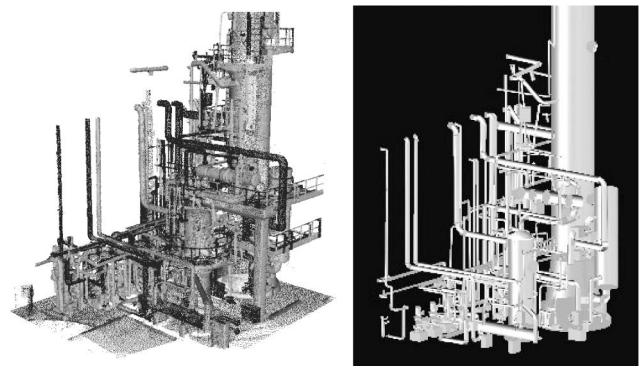


Fig. 2. A subset of scanned SINOPEC Beijing Yanshan PTA plant. Left: point cloud of 5,234,852 points. Right: reconstructed model of 313 pipelines.

that there is only one potential correct hypothesis contained in a noised sample set, for example, there is only one correct fundamental matrix in the narrow baseline stereo matching. However, in our case there are different subsets corresponding to different cylinders and thus these fast RANSAC methods cannot be used.

Another popular method for geometric primitive detection is the Hough transform [1], [18]. Given a parametric representation of a geometric primitive, Hough transform uses a duality principle in the parametric space. Recently some improved Hough transforms have been proposed, either better locating the local maxima in the parametric space [6] or speeding up the procedure by splitting and pruning the parametric space [17]. However, since a cylinder has five parameters [20], given the large-scale point set with a large number of pipeline primitives, search in 5D parameter space may result in a prohibitively high complexity for Hough transform methods.

As a short summary, the problem we solved in this work is to automatically detect pipelines (straight cylinders) as many as possible in a large-scale noised point set. Traditional methods such as region growing, RANSAC and Hough transform cannot be directly used in our case, due to the large numbers of noised scanned points (up to 10^6 - 10^9) and pipelines (up to 10^2) of different sizes. We propose a structure detection and decomposition method that converts the problem of finding cylinders in \mathbb{R}^3 into an easy problem of finding circles in \mathbb{R}^2 . After we find pipelines in the scanned points (see Fig. 8 top and middle rows), adjacent pipelines are detected and connected by elbow pipes. Finally, the reconstructed plant model is made up by manually assembling necessary accessories (see Fig. 8 bottom row) such as valves, meters, caps of oil

TABLE 1
Data Summary of Pipelines in Fig. 2

| Primitive | Point No. | Percentage in total pts. | Selection No. in RANSAC |
|---------------------------------|-----------|--------------------------|------------------------------|
| Air chimney | 852,943 | 16% | 20,050 |
| Oil tank 1 | 101,835 | 1.9% | 826,518,135 |
| Oil tank 2 | 99,297 | 1.8% | 937,685,265 |
| Oil pipeline 1 | 119,014 | 2.3% | 379,094,485 |
| Oil pipeline 2 | 31,720 | 0.6% | $\approx 2.8 \times 10^{11}$ |
| Each of all other oil pipelines | < 25,000 | < 0.45% | $> 9.2 \times 10^{11}$ |

tanks, staircase, centrifuges, and so on. In the remainder of this paper, we use terms of pipeline and cylinder interchangeably.

3 STRUCTURE DETECTION IN PLANT DATA

We use the following structural information from pipeline assembly in plant construction. The pipeline directions can be classified into two types:

- T1. The pipelines are perpendicular to the ground.
- T2. The pipelines are parallel to the ground.

Accordingly, we proposed the structure detection algorithm for pipeline plant reconstruction as below.

In Algorithm 1 (Steps 5.1 and 8.3), we use a structure feature in pipeline point cloud: *if we project the points of pipelines that have the same orientation n into the plane $\phi_n(x) = n \cdot x + d = 0$, d is an arbitrary constant, then finding pipelines in \mathbb{R}^3 reduces to finding circles in the plane ϕ_n .* Based on this structure feature, we first detect the pipeline direction of type T1, then detect pipeline directions of type T2. The method we use to detect pipeline directions is based on the Gauss map [9], which maps the normals of points in S to the unit sphere \mathbb{S}^2 . We propose a Gaussian spherical histogram on \mathbb{S}^2 in Section 4.3 for detecting pipeline directions of type T2. Note that in our method, we do not directly use point normal information to determine the pipeline parameters; instead, we only use point normals of high confidence to filter points for later projection. The details are presented below.

Algorithm 1: Pipeline_Reconstruction.

Input. A large-scale point set S , thresholds τ_1, τ_2

Output. The parameters of detected pipelines

1. For each $s \in S$, estimate the normal n_s at s with a confidence $f(s)$ (Section 4.1)
2. Delete all points with $f(s) < \tau_1$ from S
3. Detect the ground plane $\phi(x) = n_\phi \cdot x + d = 0$ (Section 4.2)
4. From S collect all points with $|n_s \cdot n_\phi| < \tau_2$ in S_ϕ
5. Project points in S_ϕ into plane ϕ
- 5.1. Detect circles in plane ϕ
6. $S = S \setminus S_\phi$
7. Find pipeline directions (of type T2) in S using Gaussian spherical histogram (Section 4.3)
8. For each found pipeline direction n_i
 - 8.1. Collect all points with $|n_s \cdot n_i| < \tau_2$ in S_i
 - 8.2. Project points in S_i into the plane ϕ_i perpendicular to n_i
 - 8.3. Detect circles in plane ϕ_i
 - 8.4. $S = S \setminus S_i$

4 ALGORITHMIC DETAILS

4.1 Preprocess

For each point s in point set S , we collect k nearest neighbors¹ in S , denoted by $Nbr(s)$, using a variant of octree-based external memory structure [5]. The normal n_s

of s is estimated by fitting a tangent plane with $Nbr(s)$, using the covariance matrix $C = \sum_{v \in Nbr(s)} (v - E(v))(v - E(v))^T$, where $E(v) = \frac{1}{k} \sum_{v \in Nbr(s)} v$ and v is column vector. Let $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ be three eigenvalues of C . We choose the eigenvector corresponding to λ_3 as n_s . The estimated normal n_s is still subject to a \pm sign for a consistent orientation. However, our method for pipeline direction detection and plane projection does not need such as a global information.

We define a confidence level as $f(s) = \frac{\lambda_1}{\lambda_3 + \varepsilon}$, where $\varepsilon = 0.05$ is a small constant that prevents denominator is close to zero. The larger $f(s)$ is, the higher confidence of n_s being the true normal. We use $f(s)$ to filter out unreliable normals by setting a threshold τ_1 . We set τ_1 in Algorithm 1 as a user input parameter. τ_1 can also be determined automatically. In our practice, setting τ_1 empirically as the median of all $f(s)$ in S achieves a good performance.

Connectivity in subsets of S . We record $Nbr(s)$ for each $s \in S$ in the external memory. Given any subset S' of the original set S , we connect points in S' into a graph $G(S')$ based on the Nbr information. The connected components of $G(S')$ can be determined in linear time (with respect to the number of points in S'), using the graph scan method.

4.2 Ground Plane Detection

Applying the principal component analysis method to whole plant data globally cannot reliably detect the ground plane. In our method, we use the Gauss map (which maps each $s \in S$ to a point in unit sphere $n_s \in \mathbb{S}^2$) and the following properties:

- The normals of points in a cylinder are mapped to a great circle in \mathbb{S}^2 .
- The ground is one of the largest plane in the scene (there may be some other planes such as walls in the plant).
- For the pipelines of type T1, their point normals are mapped into the great circle in \mathbb{S}^2 that is perpendicular to the ground normal.

We partition \mathbb{S}^2 into m equal-area cells² using the EQ algorithm [10]. Only a hemisphere \mathbb{H}^2 is used, since our estimated normals are subject to a \pm sign. Then either n_s or $-n_s$ hit a cell in \mathbb{H}^2 . For each cell c_i in \mathbb{H}^2 , we define a measure as

$$V(c_i) = (1 - \lambda) \frac{MC(c_i)}{\max_j \{MC(c_j)\}} + \lambda \frac{GCircle(c_i)}{\max_k \{GCircle(c_k)\}},$$

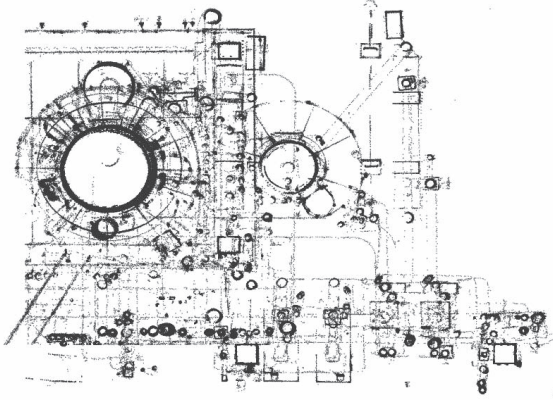
where λ is a weight, $MC(c_i)$ is the number of the maximal connected component (MCC) of points whose normals fall into c_i , and $GCircle(c_i)$ is the number of points whose normals are closely perpendicular to the representative normal $n(c_i)$ in c_i by satisfying³ $|n_s \cdot n(c_i)| < \tau_2$. We least-squares fit a plane to the points in the MCC of c_i and $n(c_i)$ is set to be the normal of that plane.

The value of measure $V(c_i)$ is balanced by two factors. The first is the normalized number of MCC in c_i , whose

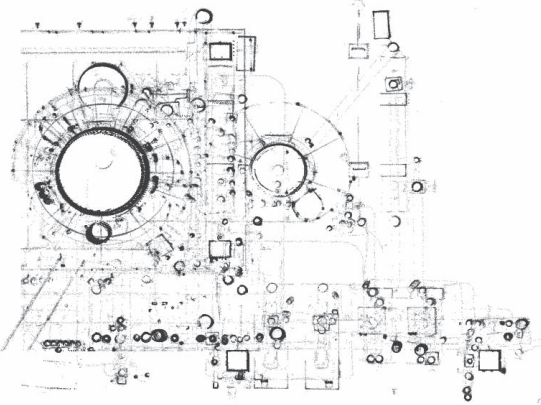
2. In our experiment, $m = 2^8$ works well.

3. τ_2 is a user-input parameter in Algorithm 1. In our experiment $\tau_2 = 0.1$ works well.

1. In our experiment, $k = 8-12$ works well.



(a) A binary image: each projected point is rendered as a black pixel.



(b) A grey image: the grey value ($\frac{n_i}{n_{max}}$) at each pixel i is determined by the number n_i of projected points.

Fig. 3. Projection of points satisfying $|n_s \cdot n_\phi| < \tau_2$ for the point model in Fig. 2 to the ground plane ϕ .

value ranges in $[0, 1]$: the larger of this normalized number, the higher confidence $n(c_i)$ is the ground normal. However, the normalized number of MCC cannot determine the ground normal solely: there may exist large walls in the plant which may occasionally have larger normalized number of MCC. So we balance it using the second factor, which is the normalized number in $[0, 1]$ of points lying in pipelines whose orientations coincide⁴ with $n(c_i)$, i.e., distributed in a great circle in \mathbb{H}^2 . Only the cell containing the ground normal can have both large normalized $MC(c_i)$ and large normalized $GCircle(c_i)$. We use $\lambda = 0.5$ and set the ground normal n_ϕ as $n(c_j)$, where $V(c_j) = \max_i \{V(c_i)\}$.

4.3 Decomposition and Detection of Pipelines

After detection of ground normal n_ϕ , we project all points whose normals satisfy $|n_s \cdot n_\phi| < \tau_2$ into the ground plane $\phi(x) = n_\phi x + d = 0$, where d does not affect projection and we use $d = 0$. One projection result is illustrated in Fig. 3. Based on this 2D projection, detection of pipelines perpendicular to the ground in \mathbb{R}^3 (type T1) is reduced to detection of circles in the projection plane ϕ . The details are presented in Section 4.4.

4. Due to noises and floating-point computation, we use a tolerance τ_2 as in footnote 3.

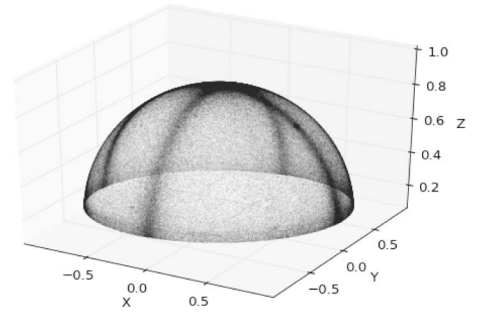


Fig. 4. Mapping of S into \mathbb{H}^2 for the model in Fig. 2.

To detect pipelines of type T2, we remove from S all points satisfying $|n_s \cdot n_\phi| < \tau_2$ (related to type T1 pipelines) or $|n_s \cdot n_\phi| > 1 - \tau_2$ (related to the ground). We then map S into \mathbb{H}^2 (see Fig. 4) and define a Gaussian spherical histogram $GSH(\mathbb{H}^2)$ as follows: Refer to Fig. 5. We orient \mathbb{H}^2 in such a way that the only complete great circle in \mathbb{H}^2 is perpendicular to the normal n_ϕ of detected ground plane. On the complete great circle, we choose an arbitrary point as the zero longitude. $GSH(\mathbb{H}^2)$ is defined with the abscissa being longitude ranged in $[0, \pi)$. For any $L \in [0, \pi)$, we use the number of points on \mathbb{H}^2 lying in the same absolute longitude L (i.e., $-L$ or L) to define the ordinate of $GSH(\mathbb{H}^2)$. Note that $GSH(\mathbb{H}^2)$ is π -periodic, i.e., $GSH(\mathbb{H}^2)(L + \pi) = GSH(\mathbb{H}^2)(L)$. $GSH(\mathbb{H}^2)$ is designed to take the characteristic that any pipeline parallel to the ground will have its all points lying in the same absolute longitude L .

In addition to pipeline points, there are also noise points in S and so is \mathbb{H}^2 (see Fig. 4 for an example). We design the following score function and use it as the ordinate of $GSH(\mathbb{H}^2)$. For each absolute longitude $L \in [0, \pi)$, let $P(L)$ be the set of all points in \mathbb{H}^2 that have the absolute longitudes in $(L - \Delta L, L + \Delta L)$. We partition $P(L)$ into connected components using the connectivity defined in Section 4.1. We first remove from $P(L)$ those components of which the point number is smaller than a predefined threshold τ_h . For the remaining components $P'(L) = \{c_1, c_2, \dots, c_m\}$, we project them into the plane parallel to the great circle indicated by the longitude L and least-squares fit a circle of radius r_i for each component c_i . Ideally, for a pipeline of a large radius, there should be a large number of points on it. Meanwhile, the larger the number m of components in $P'(L)$ is, the more significant $P'(L)$ is. Thus we define the following score function for the absolute longitude L in $GSH(\mathbb{H}^2)$:

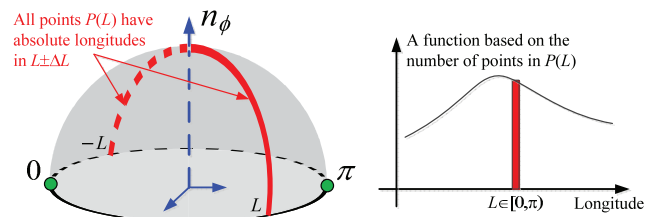


Fig. 5. Gaussian spherical histogram. Left: \mathbb{H}^2 is oriented such that the only complete great circle in \mathbb{H}^2 is perpendicular to the normal n_ϕ . Right: The corresponding Gaussian spherical histogram $GSH(\mathbb{H}^2)$.

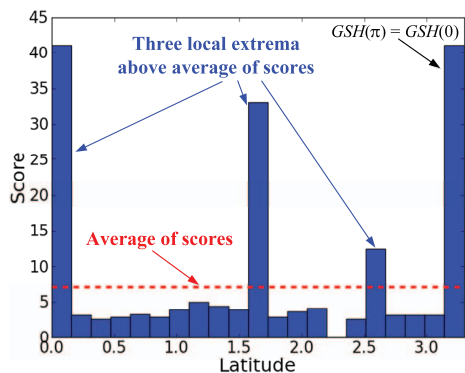


Fig. 6. The Gaussian spherical histogram for the model shown in Figs. 2 and 4.

$$Score(L) = m + \frac{\lambda}{m} \left(\sum_{i=1}^m \frac{n(c_i)}{\tau_i^\alpha} \right),$$

where $n(c_i)$ is the number of points in the component c_i , m is the number of components in $P(L)$, λ and α are the constants of weight and exponential. There are several parameters in this structure detection method that should be determined by the characteristics of scanners and scanned pipeline plants. In the example shown in Figs. 2 and 4, $\Delta L = 0.087$ (5 degree), $\tau_h = 200$, $\lambda = 10^{-3}$, and $\alpha = 1.5$. To quantize $GSH(\mathbb{H}^2)$, we use a bin number 20 in our experiments. We assume that noises are randomly distributed and an average of scores gives a good indication above which a local extremum identifies a pipeline direction. In the example shown in Fig. 6, there are three directions detected, for example, at longitudes 0 , $\pi/2$ and $3\pi/4$.

Both the method [4] and ours decompose the points into subsets, each of which has the same cylinder orientation. The key difference is that for each cylinder orientation, a great circle is fitted in [4] on Gaussian image; while we simultaneously estimate all cylinder orientations parallel to the ground using $GSH(\mathbb{H}^2)$ which takes the pipeline structure into account and results in a faster and more robust computation.

4.4 Circle Detection in Projected Plane

Our structure detection method partitions a large-scale plant point data S into subsets, corresponding to T1 or T2 types of pipelines. We project each subset into a plane whose normal is indicated by either n_ϕ or $L_i \in (L_1, L_2, \dots)$ detected by $GSH(\mathbb{H}^2)$. To simultaneously detect the multiple circles in the projected plane, we adapt the lazy evaluation scheme in [19] to develop the following RANSAC algorithm.

Algorithm 2: RANSAC($P(d)$).

1. $\Psi \leftarrow \emptyset$ // Initialize the set of extracted circles
2. $\mathcal{C} \leftarrow \emptyset$ // Initialize the set of candidate circles
3. **Do**
- 3.1. $nc = \text{NewCandidates}()$ // Randomly select three points and generate a candidate circle
- 3.2. $Cost(nc) = \frac{n(nc)}{r(nc)^\alpha}$ // Evaluate the cost of nc using the radius $r(nc)$ of nc and the number $n(nc)$ of points which fall into nc within a given tolerance
- 3.3. $\mathcal{C} \leftarrow \mathcal{C} \cup nc$

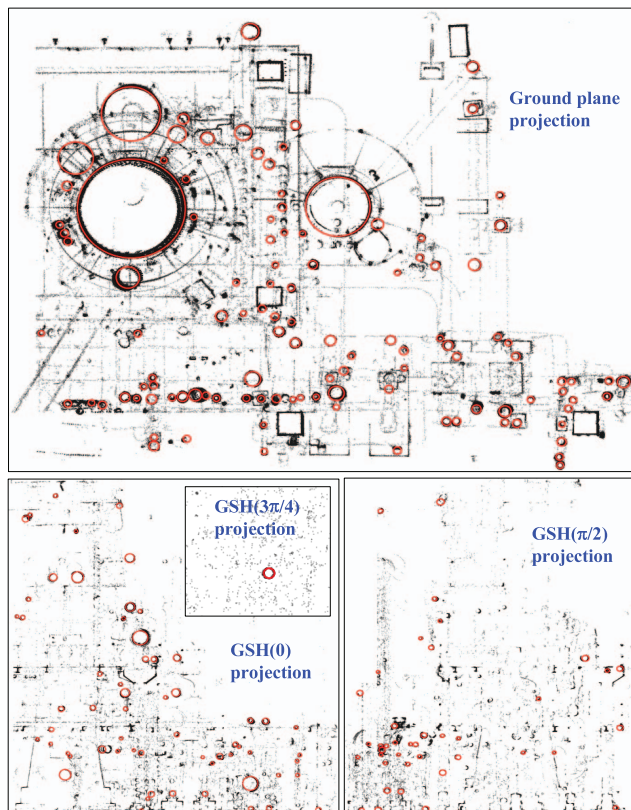


Fig. 7. Circle (shown in red) detection in four projected planes (ground and three indicated by 0 , $\pi/2$ and $3\pi/4$ of GSH in Fig. 6) for the model shown in Fig. 2.

- 3.4. $bc \leftarrow \text{bestCandidate}(\mathcal{C})$ // Assign the best candidate (i.e., of the highest cost) in \mathcal{C} to bc
- 3.5. **If** $(P(n(bc), n(\mathcal{C})) > 90\%)$ // If the possibility of bc being correct is higher than 90 percent
 - 3.5.1. $P(d) \leftarrow P(d) \setminus P(bc)$ // Remove all points devoted to the cost of bc from $P(d)$
 - 3.5.2. $\Psi \leftarrow \Psi \cup bc$
 - 3.5.3. $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{C}'$, $\mathcal{C}' = \{c \in \mathcal{C}, c \cap bc \neq \emptyset\}$ // Remove the candidate circles which intersect bc from \mathcal{C}
4. **While** $(P(\tau, n(\mathcal{C})) > 90\%)$ // $\tau = 200$ is a predefined minimal number of points for determining a pipeline

In above algorithm, $n(A)$ is the cardinality of set A , $P(m, s) = 1 - (1 - P(m))^s$, $P(m) = \left(\frac{m}{n(P(d))}\right)^3$. The main difference between Algorithm 2 and the algorithm in [19] lies in Step 3.5.3: since dozens of candidate circles may very close to each other, our algorithm removes all of them if one of them is detected and later reboot the detection of nearby circles, while the algorithm in [19] only removes the detected one from \mathcal{C} and then may lead to false detection of nearby circles.

For the point model shown in Fig. 2, there are four projected planes (one for ground plane and three from $GSH(\mathbb{H}^2)$). Circle detections on these four projected planes using Algorithm 2 are shown in Fig. 7 in which the detailed statistic data including three performance measures are summarized in the next section. Note that in the projection $P(GSH(\frac{3\pi}{4}))$, there is only one detected circle $m = 1$ with a strong intensity measured by $n(c)/r^\alpha$. Since the detected

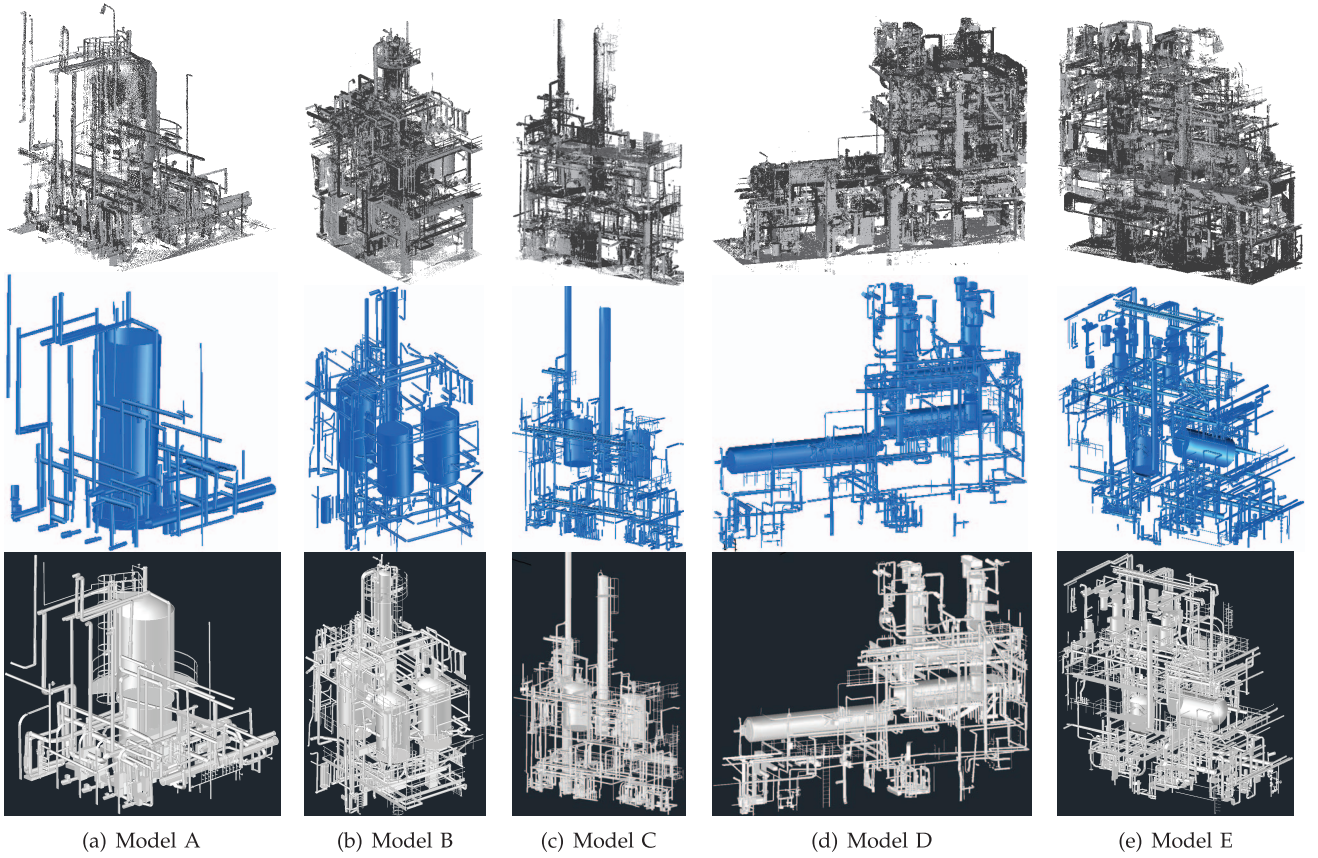


Fig. 8. More experiments on pipeline plant reconstruction from large-scale point clouds. Top row: point cloud. Middle row: detected pipelines by the proposed method. Bottom row: reconstructed pipeline plants.

TABLE 2
The Statistical Data of Reconstructing Pipelines from Large-Scale Point Cloud Models Shown in Figs. 2 and 8

| Model | Point No. | N_T | Our method | | | | | The method in [19] | | | | |
|-----------|-----------|-------|------------|-------|-------|------------|------------|--------------------|-------|-------|-----------|------------|
| | | | r_R | r_E | r_P | Time (sec) | C_{cplx} | r_R | r_E | r_P | Time(sec) | C_{cplx} |
| Fig. 2 | 5,234,852 | 313 | 0.595 | 0.453 | 0.568 | 57.560 | 1.100 | 0.676 | 1.743 | 0.279 | 85.474 | 1.633 |
| Fig. 8(a) | 1,091,166 | 246 | 0.748 | 0.496 | 0.601 | 17.260 | 1.582 | 0.488 | 0.585 | 0.455 | 17.712 | 1.623 |
| Fig. 8(b) | 4,342,436 | 308 | 0.558 | 0.494 | 0.531 | 92.012 | 2.119 | 0.545 | 2.247 | 0.195 | 257.108 | 5.921 |
| Fig. 8(c) | 4,390,326 | 350 | 0.543 | 0.400 | 0.576 | 79.044 | 1.800 | 0.343 | 2.326 | 0.128 | 150.510 | 3.428 |
| Fig. 8(d) | 4,087,532 | 274 | 0.667 | 0.690 | 0.492 | 115.439 | 2.824 | 0.690 | 1.172 | 0.370 | 143.302 | 3.506 |
| Fig. 8(e) | 6,779,602 | 394 | 0.579 | 0.579 | 0.500 | 124.112 | 1.831 | 0.305 | 1.690 | 0.153 | 181.138 | 2.672 |

N_T the number of total true pipelines in the scene, r_R is the recall ratio, r_E is the error ratio, r_P is the precision ratio.

normals of projected planes have a tolerance τ_2 , based on all detected points around a separated pipeline within a given tolerance, the parameter of the pipeline is finally reestimated in \mathbb{R}^3 [13] using the point positions only.

5 EXPERIMENTS

In addition to the scanned PTA plant in SINOPEC Beijing Yanshan Company (see Fig. 2), we also scan two large-scale pipeline plants in China: one is the alkene plant in SINOPEC Yangzi Petrochemical Company and the other is Chibi power plant in China Resources Enterprise. Five scanned large-scale point data and reconstructed pipeline plants are illustrated in Fig. 8. The statistical data of these point models are summarized in Table 2.

Let N_T be the number of total true pipelines in the plant, N_{rec} the number of true pipelines recognized by the algorithm, and N_{error} the number of false pipelines that

output from the algorithm. To compare the algorithmic performance, we define the following three measures:

- Recall ratio $r_R = N_{rec}/N_T$. An algorithm that has a good capacity of recognizing more true pipelines should have a higher value of recall ratio.
- Error ratio $r_E = N_{error}/N_T$. An algorithm may output a huge number of pipelines that include all true ones. However, a good algorithm should have a low error ratio at the same time.
- Precision ratio $r_P = N_{rec}/(N_{rec} + N_{error})$. $N_{rec} + N_{error}$ is the number of pipelines that are detected by the algorithm. Since $r_P = \frac{r_R}{r_R + r_E}$ and a good algorithm should have a high r_R and a low r_E at the same time, we use r_P as an auxiliary measure to evaluate the algorithmic performance.

We compare our method with two state-of-the-art algorithms [11], [19], using the running time and the three measures defined above. All comparisons are performed on

a desktop PC with Intel(R) Core(TM) I7 CPU, running at 2.80 GHz with 8-GB RAM.

The arterial snake [11] is a novel optimization algorithm that can simultaneously capture the topology and geometry of arbitrary curved pipelines from point data. After initialization using snakelets, the arterial snake algorithm alternates between topology and geometry optimization. To recover a general curved network of arterial snakes, in addition to the point data S , the algorithm in [11] stores intermediate information including a skeletal curve for each snake as a polyline of n nodes, a cross section for each node in skeletal curves as a polyline of m nodes, a tensor at each point in S , and an affine matrix at each vertex in the skin mesh. Furthermore a globally coupled optimization is used in [11] and thus the arterial snake algorithm cannot handle large-scale point data. In our application, for the model shown in Fig. 2, the arterial snake algorithm can only handle points up to 7.3×10^4 and its total running time is 1.91×10^6 seconds by outputting the measures $r_R = 0.128$ and $r_E = 0.296$. As a comparison, our method takes pipeline structure in plant into account and is faster and more accurate (see Table 2).

Since the arterial snake cannot handle large point data, below we only compare our method with the method in [19]. As demonstrated by the data summarized in Table 2, the method [19] generally recognizes comparable true pipelines with our methods; however, the method [19] at the same time generates a large number of false pipelines that did not exist in the point data and thus has a high error ratio r_E . As a comparison, our method improves the recall ratio r_R on average by 32 percent, reduces the error ratio r_E on average by 60 percent, and accordingly improves the precision ratio r_P on average by 153 percent.

Both the method [19] and our method use a similar RANSAC scheme.⁵ To check the empirical time complexity of the algorithm in terms of point number, we define a complexity constant $C_{cplx} = 10^5 \cdot \text{Time}/\text{Point_No}$. The data C_{cplx} summarized in Table 2 show that if the point density is close to each other in all data, both the method [19] and our method have an empirically linear time complexity with respect to the point number. Recall that the probability of one random selection of k points in RANSAC that obtains a correct solution is $(\frac{m}{N})^k$ [8], where m and N are the number of points in the detected primitive and in the whole point set. Benefitting from the structure detection and decomposition, our method detects circles in \mathbb{R}^2 (with $k = 3$ and a small N), while the method [19] directly works in \mathbb{R}^3 (with⁶ $k = 5$ and a large N). On average our method reduces 33 percent running time when compared to the method [19].

The limitation of our method is its only support to detect pipelines either perpendicular or parallel to the ground. Although sufficient for pipeline plant reconstruction, it is interesting to extend our method for including those cylinders of arbitrary orientations. After detection of pipelines perpendicular or parallel to the ground, our method can be extended to detect arbitrary cylinder orientation in the following way. Any two points in \mathbb{H}^2

5. This RANSAC scheme is linear in the form of $O(n(\mathcal{P}) \cdot n(\mathcal{C}) \cdot \text{Cost}(c))$, where $n(\mathcal{P})$, $n(\mathcal{C})$ are the numbers of detected primitives and drawn candidate primitives, $\text{Cost}(c)$ is the evaluating cost.

6. We do not use the noised normal information and $k = 5$ points determine a cylinder in \mathbb{R}^3 [7].

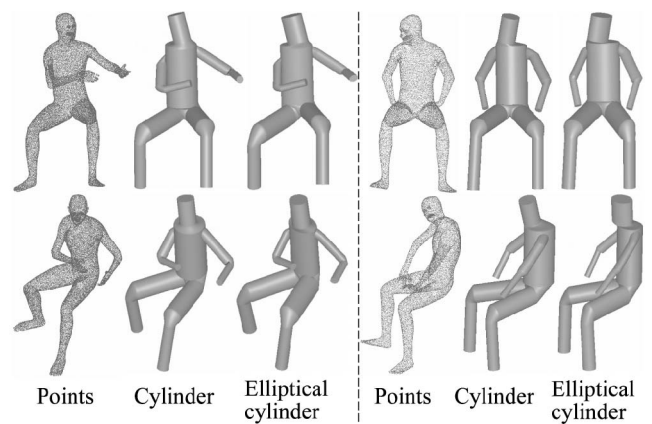


Fig. 9. Extension of our method for detecting arbitrary cylinder orientation. Neighboring cylinders are connected with elbow pipes.

uniquely determine a plane through the spherical center. We find such planes of highest costs one by one using the RANSAC algorithm that evaluates planes using the number of points in \mathbb{H}^2 that fall into the plane within given tolerance. After projecting related points in S into the detected plane, our method can also be extended easily to detect general ellipses using RANSAC in the plane. Two examples on general human point data are illustrated in Fig. 9, in which the detection of elliptical cylinders of arbitrary orientations can serve as a preprocess step for pose and motion detection.

6 CONCLUSION

In this paper, we present a structure detection method, which can efficiently process the large-scale point data of pipeline plants with the following merits: 1) based on the structural information inherent in the data, the original large-scale point data is decomposed using a Gaussian spherical histogram into small subsets that are further processed separately, and 2) different from previous work that detect pipelines in \mathbb{R}^3 , for each subset the 3D pipeline detection problem is reduced to a much simpler 2D circle detection problem.

The proposed structure detection method is simple and its time complexity is linear with respect to the point number. Experimental results demonstrate that when comparing to two state-of-the-art algorithms [11], [19], our method for pipeline plant reconstruction is faster and more accurate by taking structural information in plant into account.

ACKNOWLEDGMENTS

The authors thank the editor and all reviewers for their constructive comments that help improve this paper. They also appreciate Dr. Guo Li and Prof. Ligang Liu at Zhejiang University for providing their code in [11]. This work was supported by the NSFC (61272228), the 973 Program of China (2011CB302202), and the 863 program of China (2012AA011801). The work of Y.J. Liu was supported in part by NCET-11-0273 and TNList Cross-discipline Foundation.

REFERENCES

- [1] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981.

- [2] R.C. Bolles and M.A. Fischler, "A RANSAC-Based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data," *Proc. Int'l Joint. Conf. Artificial Intelligence (IJCAI '81)*, pp. 637-643, 1981.
- [3] D. Capel, "An Effective Bail-Out Test for RANSAC Consensus Scoring," *Proc. British Machine Vision Conf.*, pp. 629-638, 2005.
- [4] T. Chaperon and F. Goulette, "Extracting Cylinders in Full 3D Data Using a Random Sampling Method and the Gaussian Image," *Proc. Vision Modeling and Visualization Conf. (VMV '01)*, pp. 35-42, 2001.
- [5] P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno, "External Memory Management and Simplification of Huge Meshes," *IEEE Trans. Visualization and Computer Graphics*, vol. 9, no. 4, pp. 525-537, Oct.-Dec. 2003.
- [6] R. Dahyot, "Statistical Hough Transform," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1502-1509, Aug. 2009.
- [7] O. Devillers, B. Mourrain, F.P. Preparata, and P. Trebuchet, "Circular Cylinders through Four or Five Points in Space," *Discrete and Computational Geometry*, vol. 29, no. 1, pp. 83-104, 2002.
- [8] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, no. 6, pp. 381-395, 1981.
- [9] M.P. do Carmo, *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Inc., 1976.
- [10] Y.J. Liu, Z.Q. Chen, and K. Tang, "Construction of Iso-contours, Bisectors and Voronoi Diagrams on Triangulated Surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1502-1517, Aug. 2011.
- [11] G. Li, L. Liu, H. Zheng, and N. Mitra, "Analysis, Reconstruction and Manipulation Using Arterial Snakes," *ACM Trans. Graphics*, vol. 29, no. 6, Article 152, 2010.
- [12] T. Lozano-Perez, W.E.L.G. Grimson, and S.J. White, "Finding Cylinders in Range Data," *Proc. Int'l Conf. Robotics and Automation*, pp. 202-207, 1987.
- [13] G. Lukacs, R. Martin, and A.D. Marshall, "Faithful Least-Squares Fitting of Spheres, Cylinders, Cones and Tori for Reliable Segmentation," *Proc. Fifth European Conf. Computer Vision (ECCV '98)*, pp. 671-686, 1998.
- [14] J. Matas and O. Chum, "Randomized RANSAC with $T_{d,d}Test$," *Image and Vision Computing*, vol. 22, no. 10, pp. 837-842, 2004.
- [15] J. Matas and O. Chum, "Optimal Randomized RANSAC," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1472-1481, Aug. 2008.
- [16] L. Nan, A. Sharf, H. Zhang, D. Cohen-Or, and B. Chen, "SmartBoxes for Interactive Urban Reconstruction," *ACM Trans. Graphics*, vol. 29, no. 4, Article 93, 2010.
- [17] C.F. Olson, "Locating Geometric Primitives by Pruning the Parameter Space," *Pattern Recognition*, vol. 34, no. 6, pp. 1247-1256, 2001.
- [18] T. Rabbani and F. van den Heuvel, "Efficient Hough Transform for Automatic Detection of Cylinders in Point Clouds," *Proc. ISPRS Workshop Laser Scanning*, pp. 60-65, 2005.
- [19] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214-226, 2007.
- [20] G. Taubin, "Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 11, pp. 1115-1138, Nov. 1991.



Yong-Jin Liu received the BEng degree from Tianjin University, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, China, in 2003. He is now an associate professor with the Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, China. His research interests include pattern recognition, computer graphics, computational geometry, and computer-aided design. He is a member of the IEEE, IEEE Computer Society, and IEEE Communications Society. More research topics about him can be found at <http://cg.cs.tsinghua.edu.cn/people/~Yongjin/Yongjin.htm>.



Jun-Bin Zhang received the BEng degree from the Department of Computer Science and Technology, Zhejiang University, in 2009. He is currently working toward the master's degree in the Department of Computer Science and Technology, Tsinghua University, China, under the supervision of Dr Yong-Jin Liu. His research interests include computer graphics and computer-aided design.



Ji-Chun Hou received the BEng degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2011. He is currently working toward the master's degree in the Department of Computer Science and Technology, Tsinghua University, China, under the supervision of Dr Yong-Jin Liu. His research interests include computer graphics, computer animation, and computer-aided design.



Ji-Cheng Ren received the bachelor of computer sciences degree from Shandong University, China, in 1989, and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 1999. He is now a senior researcher and the vice president of Beijing Zhongke Fulong Computer Technology Co., Ltd. His research interests include computer graphics, computational geometry, computer-aided design, and enterprise informatization.



Wei-Qing Tang received the BEng and MEng degrees from the Nanjing University of Science and Technology, China, and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, China, in 1993. He is now a professor at the Institute of Computing Technology, Chinese Academy of Sciences, China, and the president of Beijing Zhongke Fulong Computer Technology Co., Ltd. His research interests include computer graphics, computational geometry, computer-aided design, and enterprise informatization.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.