

Automatic 3D Industrial Point Cloud Modeling and Recognition

Guan Pang, Rongqi Qiu, Jing Huang, Suya You, Ulrich Neumann
University of Southern California

{gpang, rqiu, huang10, suya, uneumann}@usc.edu

Abstract

3D modeling of point clouds is an important but time-consuming process, inspiring extensive research in automatic methods. Prior efforts focus on primitive geometry, street structures or indoor objects, but industrial data has rarely been pursued. Our work presents a method for automatic modeling and recognition of 3D industrial site point clouds, dividing the task into 3 separate sub-problems: pipe modeling, plane classification, and object recognition. The results are integrated to obtain the complete model, revealing some issues during the integration, solved by utilizing information gained from each individual process. Experiments show that the presented method automatically models large and complex industrial scenes with a quality that outperforms leading commercial modeling software and is comparable to professional hand-made models.

1 Introduction

3D point cloud modeling (fig. 1) is a crucial precondition for many applications, such as intelligent scene understanding, robotic navigation and augmented reality. It is a very time-consuming process, requiring manual labeling and creation of surfaces and their connections. In this paper, we describe a method for automatic 3D modeling and recognition of an industrial scene from 3D point cloud data. Our 3D model includes polygons or other 3D surface representations and their connectivity. The scene will be recognized by separated processes into pipes, planes, and various types of objects, providing metadata for the models. Creating such models currently requires extensive time by skilled modelers, even using the best software tools available today.

We consider industrial point cloud to be a congregation of pipes, planes and objects, and each part of the data is handled separately. Two of them are processed through primitive extraction that detect cylinder and planar geometry in the scene and estimate models and parameters to fit the evidence. The connecting regions between primitives are evaluated to determine the existence and type of connection joint. Objects are modeled through recognition, with a 3D object recognition process that matches clusters of 3D points to objects stored in a prebuilt object library. The best-matched library object is used to represent the point cluster, with its library mesh model replaced at the recognized locations. Combination of primitive extraction and object recognition processes complete a 3D model for a complex industrial scene. The individual processes can also benefit from each other's results during the integration to align objects and primitives, fill gaps and reduce recognition complexity. With the separately modeled pipes, planes and objects, the merged components are freely displayed in either mesh

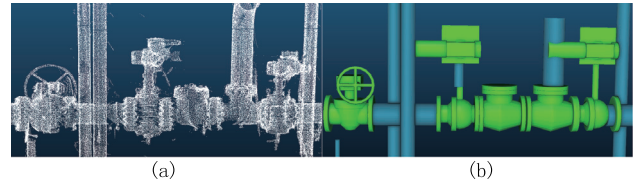


Figure 1: (a) Original industrial point cloud. (b) 3D models automatically built by our method.

model for efficiency or point cloud for accuracy, which won't be achievable in traditional methods. Several industrial point cloud datasets are used in experiments to demonstrate the performance of our method. The completeness and quality of our models is compared to a leading commercial modeling product [8] as well as hand-made models by professional service providers.

Our main contributions include:

- Describe an method combining pipe modeling, plane classification, and object recognition that automatically models 3D industrial point clouds.
- Propose to model complicated industrial object point clouds by 3D recognition and then replacement with library models.
- Utilize results from individual processes to solve some issues that arise in the results integration.
- Present results for several challenging data sets and a multi-mode display capability, uniquely achievable with our method.

2 Related Work

3D point cloud scans are often used to model planar surfaces [1, 2], or recognize specific categories of street objects [3, 4] and indoor objects [5, 6]. In contrast, industrial site scans are a much neglected scenario, including a wide variety of shapes and structures. While pipes and cylinders are prevalent, their junctions are complex, and often connect to valves, pumps, and instrumentation. Existing methods for industrial point cloud modeling [7] are not able to process a point cloud scene containing both primitive shapes such as cylinders, pipes and planar surfaces, as well as objects such as valves, pumps and instrumentations, and also the interconnections between them. Specialized commercial software [8] can automatically extract pipes and allow interactive editing of pipe networks from LiDAR scans of industrial scenes, but performance is not satisfactory, and it completely ignores other scene objects. Hand-made models may achieve arbitrary levels of detail and fidelity, but human mistakes are also common in such models due to fatigue and complexity. Recent advances in primitive extractions [9, 10] and 3D object recognition [12] in industrial data help us develop our 3D industrial point cloud modeling algorithm by dividing and solving several sub-problems.

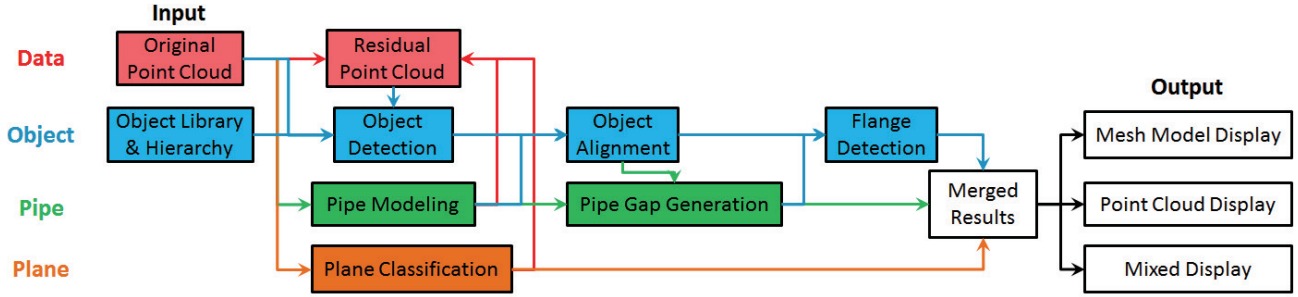


Figure 2: Algorithm flow - relationship of each process and results merging issues.

3 Algorithm

The core idea of our 3D modeling algorithm is to treat an industrial point cloud as a congregation of pipes, planes and objects. Pipes and planes are individually extracted by separate processes for pipe modeling and plane classification. Objects are modeled through 3D object recognition then replaced with library object models. The complete model is produced by merging results from the three independent steps, which can benefit from each other during integration. This results in an overall better complete model, which is another benefit of modeling pipes, planes and objects separately. The relationship of each process and merging issues are illustrated in the diagram in fig. 2.

3.1 Primitive Extraction

Since pipe-runs are critical and prevailing shapes in typical industrial sites, we explicitly detect pipe-runs in the first place. The pipe-run modeling is implemented based on the work in [9], which takes 3D raw point clouds as input and extracts pipes and reconstruct joints in the scene into a complete pipe network.

Many points in an industrial scene are part of planes (e.g. ground, platforms and boxes), thus identifying clusters of such points will greatly ease and accelerate the overall modeling process and help us focus on objects with finer details. The planar surface points classification scheme presented in [10] is applied, combining the SVM-based classification with the Fast Point Feature Histogram (FPFH) [11] as descriptor to characterize the neighborhood of each point.

3.2 Modeling Objects by Recognition

After primitive shapes such as pipes and planes are extracted, more complicated objects will be modeled following an idea of modeling by recognition, using the 3D object recognition algorithm in [12]. Each class of object has a detector trained using Adaboost, which detects 3D objects in an exhaustive window-scanning search, achieving a recall rate of 85.5% under cluttered and noisy industrial scenes. The use of 3D summed area table helps ensure the algorithm efficiency (e.g. detecting a 10k points object in a middle-size industrial scene with 10 million points takes about half an hour). After the objects are recognized, their library models will be replaced at the recognized location to complete the modeling for objects. Multiple objects in the library can be detected independently in parallel for

efficiency, with detection results merged according to confidence.

3.3 Object Alignment

Each type of objects is recognized independently by exhaustive window search with a discrete step size, and each object instance is detected independently as a local confidence maximum. As a result, the recognized location of objects may appear with a small displacement from original location, and thus not aligned with each other like in fig. 3(b)¹. After merged with the pipe models, the visual artifacts caused by these small displacements are worsen, because most objects are supposed to be aligned with pipes.

To solve this issue, we utilize the information gained from pipe models, by trying to align the detected objects with one of the pipe segments. Each modeled pipe segment has an axis vector, two endpoints and a radius. Each type of target object supposed to be connected to a pipe (usually with a pipe-shaped base equivalent to a pipe segment) will also be defined with an axis and two endpoints. After results are merged, each detected object will be attempted to align with a nearby pipe segment in similar orientation, so the endpoints lie on the same axis. As shown in fig. 3(c), all objects in the scene are aligned with a pipe segment.²

3.4 Pipe Generation in Gaps

In fig. 3(c), there is a single object on the left separated from others, but still aligned. It seems a pipe segment is supposed to connect the object to others, but pipe modeling failed to do it because it's not long enough to establish a confident pipe segment there. Nonetheless, it's still long enough to become a visual artifact. It's not easy to solve this by the pipe modeling process alone, but the detected and aligned objects provide extra information. All objects and pipe segments with similar axis will be analyzed. If two objects on the same axis has a small gap in between, we can predict there might be a pipe segment should connect them. But coaxial pipe segments may not always be connected, so the predicted pipe segment

¹Please note that some object models are inconsistent with the point cloud (e.g. short horizontal pipe on pump head, smaller valve handle) because of the data provider, not the error of our algorithm (we are only replacing their object CAD models at object locations recognized by our algorithm in 3D point clouds).

²Notice that the objects may appear slightly overlapped because library models from source data is not accurate - bigger than actual point cloud.

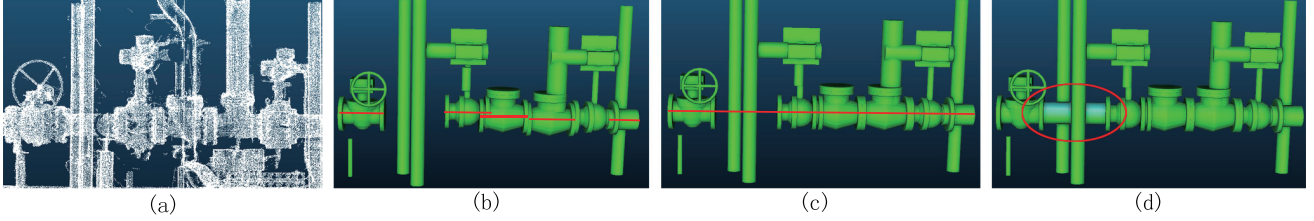


Figure 3: Comparison of gradually improved modeling results by our algorithm. (a) Original industrial point cloud. (b) Unaligned results after merging pipe models and detected 3D objects. Red lines mark object axes. Notice the small relative displacement between objects. (c) Objects are aligned to pipes (compared to (b)). Red line marks the aligned axis. (d) Pipe segments are predicted in small gaps between objects on the same axis.

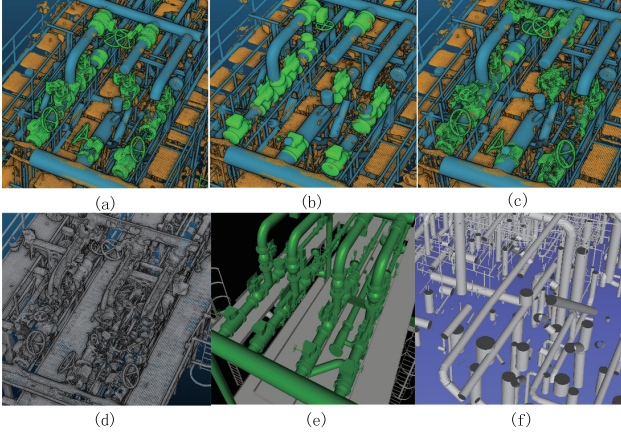


Figure 4: (a) Results by our automatic 3D modeling, with detected objects rendered in point cloud. (b) Results by our method, with detected objects rendered in mesh models. (c) Results by our method, with detected objects replaced by data at corresponding location from the original point cloud. (d) Original point cloud. (e) Manual CAD models by professional modelers with one extra (mistaken) pipe line and other small errors. (f) Models by commercial software [8] with lots of errors but none of the objects.

will need to be verified with the original point cloud. Fig. 3(d) shows the gap is filled successfully.

3.5 Other Integration Features

In industrial scenes, flange is a special type of object that is too complex to be captured by pipe modeling, but yet too simple to be recognized by object recognition with both good precision and false alarm rate. However, the extra pipe information can help locate the flanges successfully through 3D point cloud object recognition. Since flanges should always align with a pipe, searching only along (but not limited to) the established pipe axis can ensure good precision while effectively reducing false alarms.

Exhaustive searching in 3D space for 3D object recognition is very time-consuming. However, we have primitive shape points of pipes or planes, which are usually not considered as objects. Therefore, to reduce search space, we may remove all the points belonging to pipes or planar surfaces from the original point cloud, resulting in a residual point cloud. The object detector can then limit its search space according to the residual points. However, many industrial objects

usually contain part of pipe or plane, such as T-junctions or pumps, which might be incomplete after points removal. The solution is to use both point clouds: the residual for locating search windows, and the original for computing and evaluating the detector.

4 Results

4.1 Display Modes

Fig. 4(a, b, c) shows the models created by our method for a large industrial scene. Pipes, planes and objects are rendered in different colors, showing they are independently recognized and modeled. The result can be freely displayed in either mesh model for efficiency or point cloud for accuracy, not achievable in traditional methods. Individually detected and labeled components also open the possibility for extra metadata and interaction in virtual environment. Fig. 4(a, b, c) demonstrates three display modes, with objects displayed in point clouds, mesh model or original point cloud data cropped at detected location, respectively. While mesh models are more efficient, highly precise mesh model is hard to come by due to time and effort required, thus it's usually less accurate than point cloud display. With separately recognized components, we can choose to display the more accurately modeled primitives in mesh model, while leaving complex objects in point cloud. Rendering part of the results in point cloud mode is not just a simple clone, but more precise reconstruction of the original point cloud, while still preserving the meta-information of the recognized primitives and objects.

4.2 Comparison

Fig. 4(a, b, c) present the modeling results by our method in different display modes. Fig. 4(d) provides the original point cloud for reference. Fig. 4(e) shows the CAD models bundled with the source data, manually created by professional service providers. The hand-made model has 4 pipe lines on the top platform, which is wrong compared to the original point cloud who only has 3. On the other hand, our method correctly modeled the 3 pipe lines, showing that our automatic method sometimes even outperform human eyes in complex scenes. In addition, some objects in the CAD model are very inaccurate, such as the small valves. The problem will not present if we render the objects in point cloud as in fig. 4(a, c). The hand-made model is also worse on rails and platforms. Fig. 4(f) shows the result by a leading commercial automatic

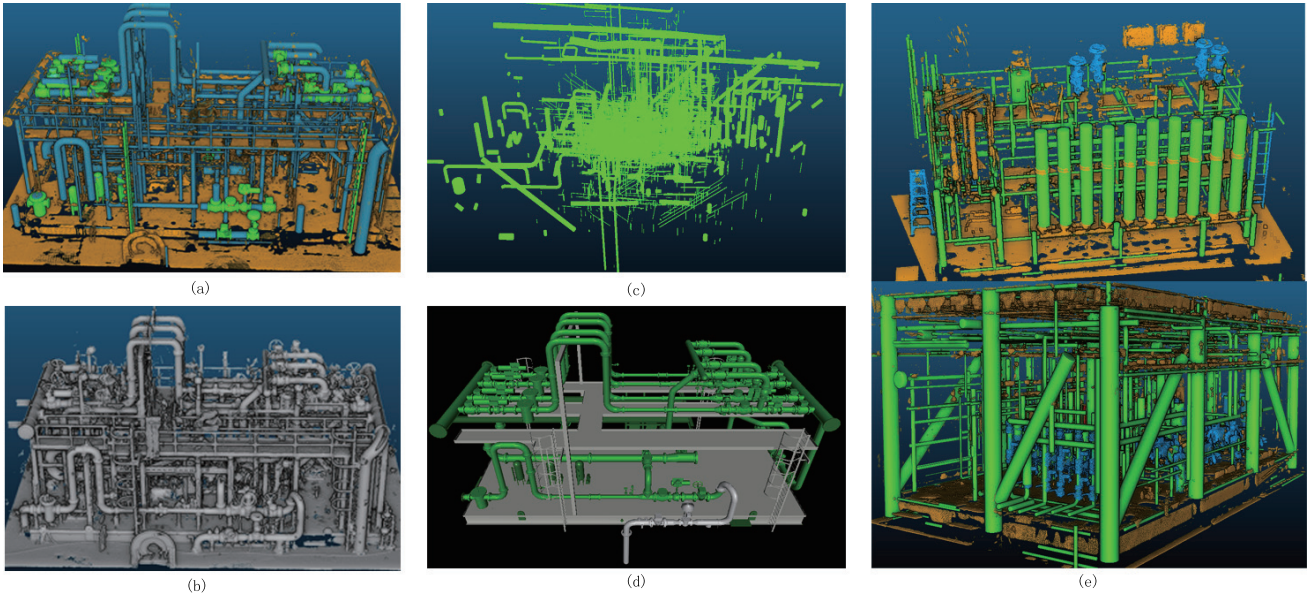


Figure 5: (a) Complete industrial scene reconstruction by our automatic 3D modeling method. (b) Original point cloud. (c) Models by commercial software [8]. (d) CAD models hand-made by professionals, cleaner but lose too much details. (e) Results by our method on more industrial scenes and datasets.

modeling product, ClearEdge 3D [8], which is only able to correctly model some pipes but none of the objects, together with lots errors.

4.3 Complete Reconstructed Scene

Fig. 5(a) shows a full industrial scene reconstruction result by our automatic 3D modeling method, demonstrating its capability in handling large-scale complex data. Fig. 5(c) shows the result by ClearEdge 3D [8], almost unrecognizable. Fig. 5(d) shows the hand-made model by professionals, which does look cleaner but loses too much details compared to our result, and contains some inaccuracies. Fig. 5(b) is the original point cloud. Fig. 5(e) presents more results of our method on other industrial datasets and scenes.

5 Conclusion

In this paper, we present a 3D modeling algorithm to automatically create models from 3D industrial point clouds. We recognize and model pipes, planes and objects in individual steps, and utilize their results to further improve each other and solve some issues that arise when integrating separate results into complete models. Pipes, planes and different types of objects are recognized separately, allowing freely switchable display in mesh model and/or point cloud for balanced efficiency and accuracy. Experiments show that our method can successfully model large complex industrial point clouds, outperforming leading commercial automatic modeling software, and comparable to professional hand-made models.

Acknowledgement

This work is supported by Chevron U.S.A. Inc. under the joint project Center for Interactive Smart Oilfield Technologies (CiSoft), at the University of Southern California.

References

- [1] R. Triebel, F. Dellaert, and W. Burgard. Using hierarchical EM to extract planes from 3d range scans. ICRA, 2005.
- [2] S. Obwald, J.S. Gutmann, A. Hornung, M. Bennewitz. From 3D point clouds to climbing stairs: A comparison of plane segmentation approaches for humanoids. Humanoids, 2011.
- [3] A Patterson, P Mordohai, K Daniilidis. Object Recognition from Large-Scale 3D Datasets using Bottom-up and Top-down Descriptors. ECCV, pp.553-566, 2008.
- [4] A. Golovinskiy, V.G. Kim, T. Funkhouser. Shape-based Recognition of 3D Point Clouds in Urban Environments. ICCV, 2009.
- [5] B. Steder, G. Grisetti, M. Van Loock, W. Burgard. Robust On-line Model-based Object Recognition from Range Images. IROS, pp.4739-4744, 2009.
- [6] M. Ruhnke, B. Steder, G. Grisetti, W. Burgard. Unsupervised Learning of Compact 3D Models Based on the Recognition of Recurrent Structures. IROS, pp.2137-2142, 2010.
- [7] A. Johnson and M. Hebert. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. PAMI, 21(5): 433-449, 1999.
- [8] ClearEdge3D. Edgewise plant, Dec. 2012.
- [9] R. Qiu and U. Neumann. Pipe-run Extraction and Reconstruction from Point Clouds. ECCV, 2014.
- [10] J. Huang and S. You. Detecting Objects in Scene Point Cloud: A Combinational Approach. Int. Conf. on 3D Vision (3DV), 2013.
- [11] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz. Persistent Point Feature Histograms for 3D Point Clouds. Intelligent Autonomous Systems, 2008.
- [12] G. Pang and U. Neumann. Training-based Object Recognition in Cluttered 3D Point Clouds. Int. Conf. on 3D Vision (3DV), 2013.