Segmentation and Matching: Towards a Robust Object Detection System

Jing Huang
University of Southern California
huang10@usc.edu

Suya You
University of Southern California
suya@usc.edu

Abstract

This paper focuses on detecting parts in laser-scanned data of a cluttered industrial scene. To achieve the goal, we propose a robust object detection system based on segmentation and matching, as well as an adaptive segmentation algorithm and an efficient pose extraction algorithm based on correspondence filtering. We also propose an overlapping-based criterion that exploits more information of the original point cloud than the number-of-matching criterion that only considers key-points. Experiments show how each component works and the results demonstrate the performance of our system compared to the state of the art.

1. Introduction

As 3D Laser Scanning techniques become popular, directly detecting objects in the scanned point clouds has become an immediate demand for applications such as scene understanding and reconstruction. Particularly, detecting industrial objects in a cluttered scene is an even more challenging problem, since such objects are often connected or occluded by clutters of pipes, planes, etc.

There have been some detection systems e.g. matching-based detection system, in which a target library (Fig. 1) is necessary for performing template matching algorithms. Works such as [8] show the potential of a matching-based system, while in this paper we achieve substantially better results by carefully examining the bottlenecks of [8] in terms of the recall rate.

Since matching between a small part with a large region will waste much time in unnecessary comparisons and also cause confusion, a segmentation step is necessary. The simplest idea is to segment the cloud based on Euclidean distance. However, in complex industrial scenes, if the tolerance of clustering is large, several parts might stay connected in one cluster. On the other hand, if the tolerance is small, certain part can be broken in several pieces. To solve this problem, we propose an adaptive segmentation method that generates clusters with relatively small size for matching while the continuity of a single part is retained.

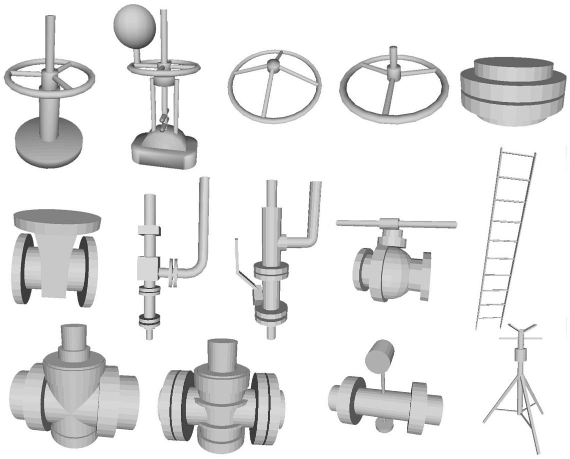


Figure 1. The CAD models of the part templates, which are converted into point clouds by a virtual scanner in pre-processing.

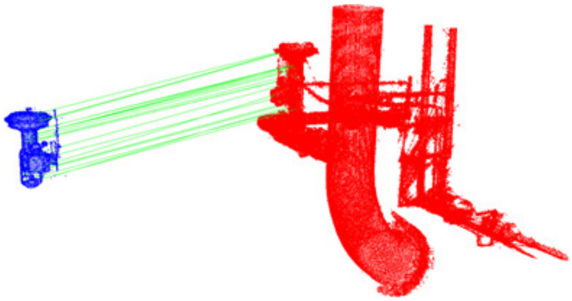


Figure 2. Matching a single part to a big cluster.

Another key part for detection-by-matching approaches is matching. Instead of the feature extraction and descriptor computation, we focus on the correspondence selection/outlier removal algorithm, which could otherwise be a bottleneck in the whole detection system. In fact, the new outlier removal algorithm largely reduces the dependency on the quality of descriptors.

RANSAC [5] is a widely used strategy in outlier removal. However, traditional RANSAC is too general to retain high efficiency. Instead, we found that many outliers could be pruned in the early stage of matching, under various assumptions/constraints. For example, we could impose rigid-body constraint, where a rigid-body coefficient is specified for different applications.

Finally, we explore the nature of a *successful* matching by comparing two different evaluation criteria. Figure 2 shows one example of the matching result between a single part to a big region using the proposed correspondence selection algorithm with the overlapping score criterion, which was impossible for the method in [8] within the same time.

The novelty of this work includes proposing a robust detection system based on segmentation and matching supported by the following new techniques:

1) An adaptive segmentation algorithm that could efficiently segment the full point cloud into a hierarchy of candidates with proper size.

2) An efficient pose estimation algorithm based on correspondence filtering with rigid-body constraint. The process could be generalized to various constraints.

3) An overlapping score method used as the evaluation and alignment criteria.

2. Related Work

Various point cloud processing systems with different goals and/or data types have been proposed in recent years. For example, [13] describes an object map acquisition system for household environments. [1] proposes an object (mainly cars) detection system for urban area with bottom-up and top-down descriptors. [8] presents a matching-based framework for detecting parts in cluttered industrial scenes. Focusing on the similar datasets, we achieve substantially better results compared to the system described in [8].

In order to reduce the size of problem, segmentation techniques are often employed in 3D point cloud preprocessing. Many methods are based on Graph-cut. For example, [11] uses a min-cut algorithm for the outdoor urban scan. [12] presents a set of segmentation methods for different types of point clouds. They also proposed an evaluation metric that could quantify the performance of segmentation algorithms. In this paper, we aim to decompose the point clouds into meaningful regions and separate the parts from each other while avoiding breaking one part into multiple pieces, with an adaptive segmentation scheme.

The extensively applied strategy, feature-based matching, performs the feature extraction and then computes descriptors in each of the extracted key-points. For example, [6] applies the Maxima of Principal Curvature (MoPC) feature detector and the 3D Self-Similarity (3D-SSIM) descriptors. [1] obtains the key-points by placing a 3D grid and computes the Spin Image descriptor [14]. A benchmark for feature detection and descriptor methods could be found in [10]. In this paper, however, we emphasize on another perspective of matching i.e. correspondence selection and pose evaluation.

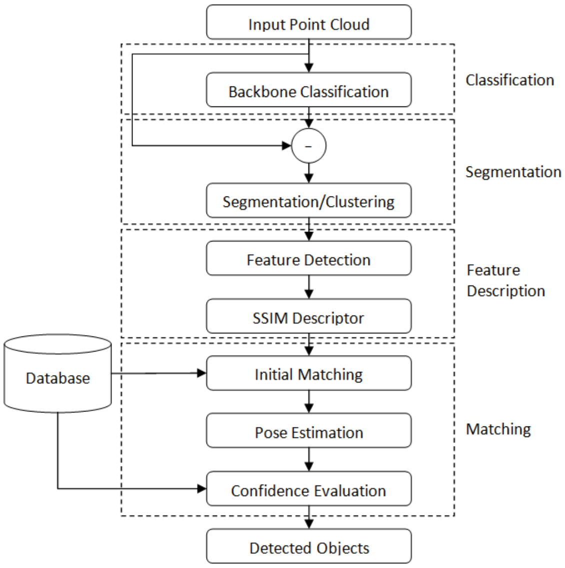


Figure 3. The pipeline of the detection system.

There are several complex correspondence selection schemes, most of which are originated from RANSAC [5]. For example, Maximum Likelihood Estimation by Sample Consensus (MLESAC) scheme proposed in [3] is also applied in [2] for image matching. [4] proposes the Progressive Sample Consensus (PROSAC) that reduces the computation with classification score. [7] presents a randomized model verification strategy for RANSAC, based on Wald's theory of sequential decision making. A comprehensive survey of the RANSAC techniques could be found in [9]. We do not apply a randomized scheme, but instead take advantage of prior knowledge to prune the outliers during the selection procedure.

For the whole system, [18] also propose a segmentation and matching scheme to deal with cluttered scenes, but it's fully based on models instead of point clouds. Since models could be converted to point clouds through a virtual scanner conveniently but not vice versa, and the scanned data are usually in the point cloud format, we provide a viable solution to process 3D data, especially those without edge information.

3. System Overview

The pipeline of the system, basically composed of four stages i.e. classification, segmentation, feature description and matching, is illustrated in Fig. 3.

Given a point cloud of a large scene, we divide all points into two categories i.e. backbone and part. The relationship between the two are like the background and foreground in 2D image segmentation. The goal of classification is thus classifying all points into these two large categories.

When classification is done, large connected components of backbone points are removed from the original point cloud and the residual points are considered as the candidate part point cloud. The candidate point cloud is then adaptively segmented into a set of candidate clusters.

In the matching stage, features are extracted using Maxima of Principal Curvature (MoPC) detector and then, the 3D Self-Similarity descriptors [6] are computed on the detected features of both targets from the database and the candidate clusters generated in segmentation. Given a pair of candidate cluster and database template, the initial feature matches are obtained with the Nearest Neighbor Distance Ratio (NNDR) criterion. Then we try to extract the best alignment between the candidate and database templates. We estimate a series of candidate poses based on different subsets of the initial correspondences induced by the top-rated seed correspondences. Matching scores are evaluated based on the overlapping ratio between the candidate and the transformed database templates, and the template with the best score higher than a confidence threshold is considered as the detection result for the candidate cluster.

From another perspective, the classification and feature description could be viewed as the preparation stage for segmentation and matching, respectively. Among all the modules, the adaptive segmentation and the best pose extraction in matching are the emphasis of this paper.

4. Backbone Classification

Typically, the backbone structures contain large planes (e.g. ground and walls), pipes and edges (the intersection of two planes), which intrinsically differ from each other. Therefore, in backbone classification, similar to [8], five basic categories are defined, including four backbone categories i.e. plane, pipe, edge and thin-pipe, as well as the remaining category, known as the part category since parts are supposed to be in this category.

Four two-class linear SVM classifiers [16] are trained between the backbone categories and the part category. To train each of the classifiers, we manually label $250k$ points with one of the five categories ($\approx 50k$ each), and compute the 33-dimensional FPFH descriptor [15] on each of the points. The positive examples for classifier of category A are the $50k$ descriptors of points labeled as category A, and the negative examples are $50k$ descriptors selected from the remaining $200k$ points labeled as other categories. In our experiments, the training set comes from a different scene point cloud to the test point cloud.

During testing, an FPFH descriptor is computed on each point from the input cloud and is tested with the trained SVM classifiers such that each point is assigned with four labels (i.e. plane/non-plane, pipe/non-pipe, edge/non-edge, thin-pipe/non-thin-pipe). Figure 4 shows an example of backbone classification. Note that it's possible that some point has multiple positive labels (e.g. a point could be both pipe and thin-pipe).

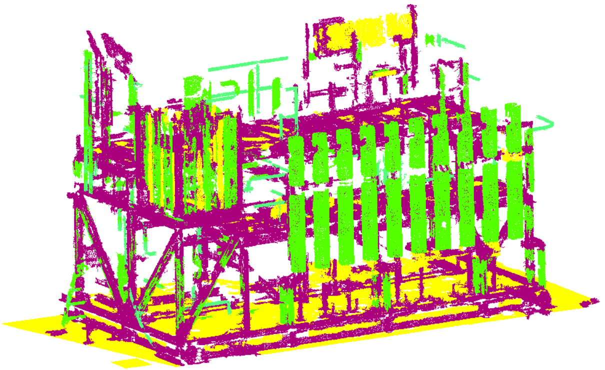


Figure 4. The backbone cloud. Points classified as plane / pipe / edge / thin-pipe are displayed in yellow / green / purple / light green, respectively.

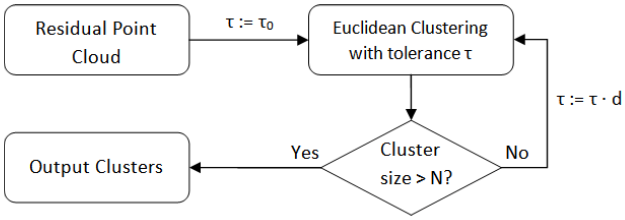


Figure 5. The process of adaptive segmentation.

5. Adaptive Segmentation

According to the labels, we are able to filter out the backbone points through the following subtraction process: we perform an Euclidean clustering for points within each of the four backbone categories, and remove them from the original cloud only if they belong to a sufficiently large cluster ($\geq$ 1000 points). In this way, we obtain the residual point cloud. In [8], the residual point cloud is segmented by a one-time Euclidean clustering. However, there are some problems in the segmentation results:

1) Big clusters can still exist after segmentation.

2) There are many parts in such big clusters, however, it's difficult for the matching stage to handle clusters that are too big.

As the solution, we propose the following adaptive segmentation procedure (Fig. 5). Instead of fixing the tolerance in Euclidean clustering, we iteratively decrease the tolerance for large clusters so that each cluster contains no more than a fixed number of points. In other words, only large clusters will be re-segmented. This process generates a tree of clusters. The tolerance is smaller for deeper layers, and the leaf nodes are the final result of segmentation.

Although parts that are close to each other could be connected due to noise or incomplete removal of backbone points, which can lead to wrong clustering by Euclidean distance, we observe that in most cases these connections would be cut off earlier than the connections among the same objects, which ensures the feasibility of the algorithm.

6. Feature Description

After segmentation, the point clouds are divided into a series of clusters with comparable size as the template point clouds scanned from the models in the database.

We apply the Maxima of Principal Curvature feature detection algorithm [6] to extract the keypoints from each point cloud, in order to reduce the number of descriptors need to be examined in matching while keeping the salient features.

Then, 3D Self-Similarity descriptors [6] are computed on each of the keypoints. We use $5 \times 5 \times 5 = 125$ dimensions for the descriptor, and the similarity value is evaluated based on normal and curvature differences. The features and descriptors are precomputed and stored for database templates in the offline processing, so we only need to compute those for the segmented clusters in the online processing.

7. Matching and Pose Extraction

After the features with descriptors are extracted and computed, we are able to estimate the pose transformation between any two point clouds. The general idea is, instead of extracting only the largest group of matches among different hypotheses that obey the rigid-body constraint, we first get all possible rigid transformations among different hypotheses of matches and then, evaluate the quality of transformations (Eq. 6) and extract the best one (Eq. 7).

7.1. Correspondence Selection

The goal of correspondence selection is to select feasible sets of correspondences from which possible transformation hypotheses could be computed.

Formally, we define a correspondence between keypoint x in point cloud X and keypoint y in point cloud Y as a triple $c = (x, y, f)$, where f is the confidence value.

Given a keypoint x, we are only interested in the keypoint y_1 that minimizes the difference of descriptors $||d(x) - d(y)||$. Also, we need to ensure that it's outstanding enough to beat other candidates, thus the negative of Nearest Neighbor Distance Ratio (NNDR) is used as the confidence value, such that the higher the confidence value is, the more outstanding the top correspondence is (Eq. 1).

$$f(c) = -\frac{||d(x) - d(y_1)||}{||d(x) - d(y_2)||} \tag{1}$$

where y_1 and y_2 are the top two corresponding points in Y that minimize $||d(x) - d(y)||$.

Now, we can assign the set of initial correspondences generated from the keypoints in X with sorted confidence values, i.e. $C_0 = \{c_i; f(c_1) \geq f(c_2) \geq ... \geq f(c_n)\}$.

For any correspondence set C, we define the seed correspondence of it as $\alpha(C)$, the correspondence with the highest confidence value (Eq. 2).

$$\alpha(C) = t \in C(\forall c \in C(f(t) \geq f(c))) \tag{2}$$

In the rare case that there are multiple correspondences with the same confidence value as the highest, any one of them could be the seed.

In the k-th hypothesis, we assume that the k-th correspondence is correct, while the first to $(k-1)$-th correspondences are wrong. Formally, we only consider the subset $C_k^{(0)} = \{c_i \in C_0; i \geq k\}$, and the remaining correspondence with the highest confidence $c_k = \alpha(C_k^{(0)})$ is the seed correspondence.

Starting from c_k, we can gradually add feasible correspondences to the final correspondence set, denoted as S_k. Initially we have $S_k^{(0)} = \emptyset$. Then, each round we add one more seed to the final correspondence set:

$$S_k^{(j+1)} = S_k^{(j)} \cup \{\alpha(C_k^{(j)})\} = \bigcup_{l=0}^{j} \{\alpha(C_k^{(l)})\}. \tag{3}$$

Meanwhile, we only preserve correspondences that obey the rigid-body constraint, i.e. the rigid-body transformation is distance-preserving (Eq. 4).

$$\begin{aligned} C_k^{(j+1)} &= \{c \in C_k^{(j)} \backslash \{\alpha(C_k^{(j)})\}; \\ \forall s \in S_k^{(j)} &(\frac{1}{\gamma} < \frac{||x(c) - x(s)||}{||y(c) - y(s)||} < \gamma)\} \end{aligned} \tag{4}$$

where γ is the rigid-ratio threshold, and for ideal rigid-body $\gamma = 1$. Using the fact that the elements in $C_k^{(j)}$ must satisfy the constraint with elements in $S_k^{(j-1)}$, we can further simplify the computation from $O(n^2)$ to $O(n)$:

$$\begin{aligned} C_k^{(j+1)} &= \{c \in C_k^{(j)} \backslash \{\alpha(C_k^{(j)})\}; \\ \frac{1}{\gamma} &< \frac{||x(c) - x(\alpha(C_k^{(j)}))||}{||y(c) - y(\alpha(C_k^{(j)}))||} < \gamma\} \end{aligned} \tag{5}$$

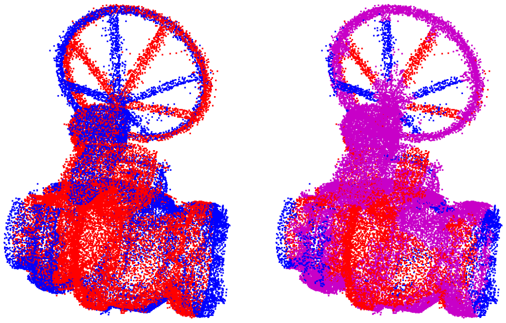
This process could be carried on until $C_k^{(j)} = \emptyset$.

Based on each self-consistent correspondence set S_k, we can estimate a transformation matrix T_k that minimize the squared difference between the correspondences and normalize T_k with the Gram-Schmidt process.

7.2. Evaluation Criteria

We define the overlapping score between the two point clouds A and B, with overlapping threshold θ, as the proportion of the points in A having at least one point in B in its θ-neighborhood (Eq. 6) (Fig. 6).

$$\Omega(A, B) = \frac{|\{x \in A; \exists y \in B(||x - y|| < \theta\}|}{|A|} \tag{6}$$



(a) (b)

Figure 6. Illustration of overlapping score. (a) shows the alignment of two point clouds and (b) highlights the overlapping area within certain threshold. The overlapping score reflects the ratio between the size of overlapping area and one of the original point clouds.

Note that in most cases $\Omega(A,B) \neq \Omega(B,A)$. In implementation, a kd-tree is built for both point clouds A and B, such that the running time for each find-nearest-neighbor operation is $O(log(|A|))$ or $O(log(|B|))$, thus the time complexity for the routine is $O(|A| \cdot log(|B|) + |B| \cdot log(|A|))$.

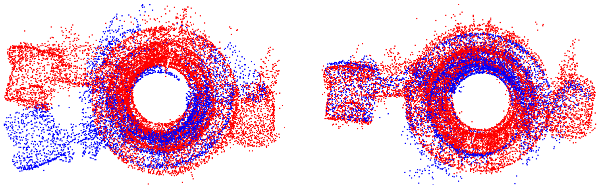
Based on the overlapping score, we define the following criteria (Eq. 7) for the best transformation extraction:

$$T_{best} = \arg\max_{T_k}(\Omega(T_k \times P_t, P_c)), \tag{7}$$

where P_t is the template point cloud and P_c is the candidate cluster. That is, we transform the template point cloud with the estimated transformations, compare them to the candidate cluster, and extract the transformation that generates the largest overlapping score as the best transformation between the pair P_t and P_c. The range of the Ω function is $[0, 1]$, and its value would reach 1 even if P_t is only a subset of P_c, meaning that the partial matching is enabled. In such cases, we would remove the overlapping part from P_c and iteratively perform the matching on the remaining point cloud until it's small enough (e.g. less than 500 points).

Here are some insights why our evaluation criterion is better than the traditional maximum-number-of-matching criterion: the criterion ensures the nearly optimum solution under the final evaluation criterion of to what extent the two point clouds are overlapping with each other, while not consuming much more computation time. Also, it is a kind of the quality control of matching at the final stage, meaning that once a good transformation has been generated, it would not be contaminated by other steps.

Moreover, this framework of 1) getting all possible transformations, 2) evaluating overlapping scores and 3) extracting best transformations based on overlapping scores could be generalized to non-rigid pose estimation as well.



(a) (b)

Figure 7. Comparison of alignment result using (a) maximum number of matching criterion and (b) overlapping score criterion. The point clouds in (a) are misaligned.

8. Experiments

8.1. Rigid Ratio Threshold

The rigid ratio threshold γ, or the maximum distance ratio between two selected correspondences, is an important threshold that imposes the distance consistency during the correspondence selection.

Generally speaking, the size of the final correspondence set will decrease as the threshold decreases, while the accuracy of the correspondences will increase. If the threshold is too tight, there will be too few correspondences to estimate the transformation, while if the threshold is too loose, the accuracy will decrease. We empirically optimize this threshold and set $\gamma = 1.08$ for all the tests.

8.2. Number of Matching Attempts

Though best correspondences typically have the best confidence values, we find that making more attempts actually does improve the results, since our evaluation mechanism, in most cases, ensures the monotonicity of performance against the number of attempts, while only adding a negligible increase in the computational complexity of the method. In the following experiments, we try 50 different hypotheses during best pose extraction.

8.3. Comparison of Evaluation Criteria

In this section, we perform the experiment based on the parameters suggested in Section 8.1 and 8.2, i.e. rigid ratio = 1.08, number of attempts = 50. The only variable is the scoring method, i.e. matching score (number of inlying matches after correspondence selection) proposed in [8] vs. overlapping score proposed in this paper.

Here is an example to illustrate the difference (Fig. 7). If we use the matching number as score, the result on the left will be selected since it has more matches (143) than the result on the right (108). However, if we apply the transformation computed from the two sets of matches, we'll find that the alignment on the left is worse than the one on the right. In a word, the overlapping score uses a more direct and final evaluation of the quality of alignment and thus gives a better result.

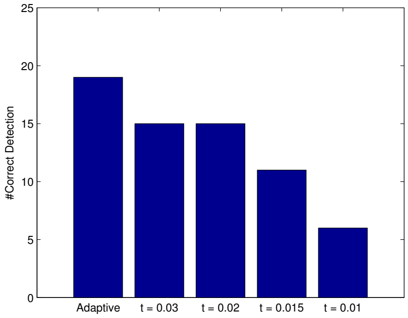


Figure 9. Comparison of segmentation / clustering methods. The vertical axis represents the number of correct detections using different methods. The first column is adaptive segmentation, while the remaining columns are segmentation with a fixed tolerance t.

8.4. Evaluation of Segmentation

We use a difficult scenario (Fig. 8) to compare the adaptive segmentation with fixed-tolerance segmentation.

For adaptive segmentation we use the following parameters: the initial tolerance $\tau_0 = 0.03$, decay coefficient $d = 0.9$, and upper bound of cluster size $N = 50000$. Except the segmentation method, the matching techniques applied are exactly the same in this experiment.

Figure 8 shows the comparison results of a cluttered scene between adaptive segmentation and fixed-tolerance segmentation. The clusters obtained from adaptive segmentation are more evenly distributed and properly segmented without disconnecting single parts.

Figure 9 shows the number of correct detections by different methods. When we simply reduce the fixed tolerance to break down the clusters, the detection rate doesn't increase as wished, but instead has a dramatically drop when the tolerance is comparable to the precision of point cloud, resulting in over-segmentation of parts. However, adaptive segmentation avoids such situation by keeping parts from being broken down when they're already small enough.

8.5. Comparison with Existing System

In this section, we run our detection system on a large industrial scene dataset containing over 15 million points. Figure 10 presents the detection result, in which the point clouds of the detected parts are replaced with the corresponding templates. The pipe models are generated with the method described in [17]. This result also demonstrates the potential application of our detection system in the point cloud modeling system. Table 1 summarizes the statistics for the full industrial scene using our method versus the method in [8]. Our method is significantly better in terms of the recall rate. More results are shown in Fig. 11, containing several pairs of original point cloud and the classification and detection results.

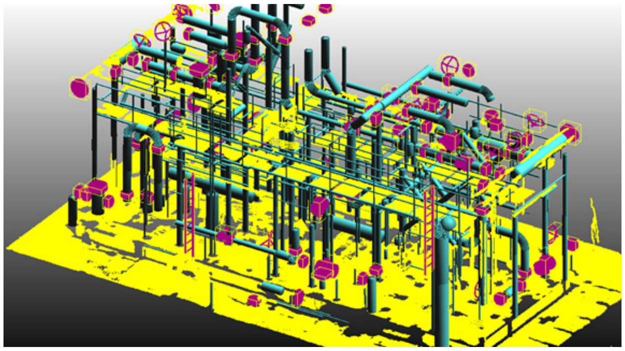


Figure 10. Detection result in an industrial scene. The detected parts are shown in purple and inside the bounding boxes.

9. Conclusion

We propose an improved framework for 3D object detection based on segmentation and matching. The adaptive segmentation algorithm greatly reduces the burdens for matching by limiting the size of clusters. Also, by applying the corresponding selection algorithm with the overlapping score criterion, the point cloud matching module has become robust enough that most pairs of well-segmented clouds of the same category of objects can be matched properly. This means at least with some human interactions, the parts could be correctly matched. Experimental results show the robustness of our system on large-scale industrial scenes and the potential application in scene reconstruction.

Acknowledgement

This research was supported by CiSoft project sponsored by Chevron. We appreciate the management of Chevron for the permission to present this work.

References

[1] A. Patterson, P. Mordohai, and K. Daniilidis. Object detection from large-scale 3d datasets using bottom-up and top-down descriptors. In Proc. ECCV, 2008. 2

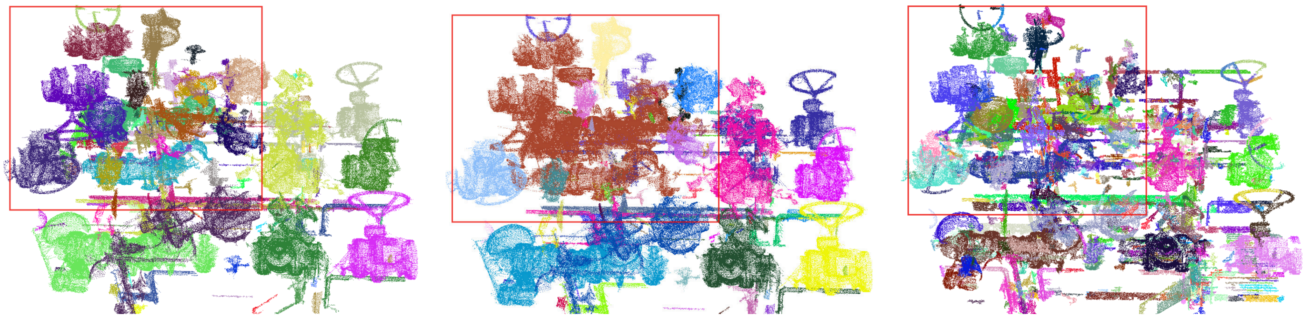
[2] B. J. Tordoff and D. Murray. Guided-MLESAC: Faster Image Transform Estimation by using Matching Priors. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 27, Issue 10 pp. 1523-1536, 2005. 2

[3] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. CVIU,78:138C156, 2000. 2

[4] O. Chum and J. Matas. Matching with PROSAC - Progressive Sample Consensus. In Proc. CVPR, 2005. 2

[5] 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. 1, 2

[6] J. Huang and S. You. Point Cloud Matching based on 3D Self-Similarity. International Workshop on Point Cloud Processing (Affiliated with CVPR 2012), Providence, June, 2012. 2, 3, 4



(a) (b) (c)

Figure 8. Comparison of segmentation result using adaptive segmentation (a), as well as fixed-tolerance segmentation with tolerance 0.03 (b) and 0.01 (c). The congested area highlighted by the bounding box (b) is successfully separated into several smaller part-like clusters by adaptive segmentation (a), while over-segmentation (c) is avoided.

Category	**#Truth**	**Proposed method**					**Huang [8]**				
		#TP	#FP	#FN	Recall	Precision	#TP	#FP	#FN	Recall	Precision
Cap	4	2	0	2	0.50	1.00	1	0	3	0.25	1.00
Cross Junction-1	5	2	1	3	0.40	0.67	0	0	5	0.00	-
Cross Junction-2	2	2	0	0	1.00	1.00	1	0	1	0.50	1.00
Flange	66	45	12	21	0.68	0.79	18	4	48	0.27	0.82
Flange-Cross	4	4	0	0	1.00	1.00	1	0	3	0.25	1.00
Flange-L	19	13	2	6	0.68	0.87	8	1	11	0.42	0.89
Ladder	7	3	0	4	0.43	1.00	0	1	7	0.00	0.00
Handwheel-L-4	13	11	0	2	0.85	1.00	8	0	5	0.62	1.00
Handwheel-M-4	1	0	0	1	0.00	-	0	0	1	0.00	-
Junction-Pneumatic	1	1	0	0	1.00	1.00	1	0	0	1.00	1.00
Tripod	5	4	1	1	0.80	0.80	3	0	2	0.60	1.00
Valve-Handwheel-3	1	1	0	0	1.00	1.00	0	2	1	0.00	0.00
Valve-L-4	1	1	0	0	1.00	1.00	1	0	0	1.00	1.00
Summary	129	89	16	40	0.69	0.85	42	8	87	0.33	0.84

Table 1. Statistics of detection. There are 13 categories, and 129 instances (Ground-truth) of targets in the scene. Our method correctly identifies 89 instances among them, while 16 detections are wrong, and 40 instances are missed. The missed instances are mainly due to large ($> 50\%$) occlusion. These results, especially the recall, are substantially better than the results of the system proposed in [8].

[7] O. Chum and J. Matas. Optimal randomized RANSAC. Pattern Analysis and Machine Intelligence, 2008. 2

[8] J. Huang and S. You. Detecting Objects in Scene Point Cloud: A Combinational Approach. 2013 International Conference on 3D Vision (3DV), Seattle, WA, June 2013. 1, 2, 3, 5, 6, 7

[9] R. Raguram, J.-M. Frahm, and M. Pollefeys. A Comparative Analysis of RANSAC Techniques Leading to Adaptive Real-Time Random Sample Consensus. In Proc. ECCV, 2008. 2

[10] E. Boyer *et al*. SHREC 2011: robust feature detection and description benchmark. ArXiv e-prints, February 2011. 2

[11] A. Golovinskiy and T. Funkhouser. Min-cut based segmentation of point clouds. Computer Vision Workshops (ICCV Workshops), 2009. 2

[12] B. Douillard, J. Underwood, K. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel. (2011, May). On the segmentation of 3-D lidar point clouds. In IEEE International Conference on Robotics and Automation, Shanghai, China. 2

[13] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3D Point Cloud Based Object Maps for Household environments. Robotics and Autonomous Systems Journal (Special Issue on Semantic Knowledge), 2008. 2

[14] A. Johnson and M. Hebert. Object recognition by Matching Oriented Points. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on (pp. 684-689). IEEE. 2

[15] R. B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, May 12-17 2009. 3

[16] C.-C. Chang and C.-J. Lin. LIBSVM: A Library for Support Vector Machines. (2011) [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm 3

[17] R. Qiu *et al.* Pipe-Run Extraction from Laser Point Clouds. Submitted to CVPR 2014. 6

[18] A. Mian, M. Bennamoun and R. Owens. 3D Model-based Object Recognition and Segmentation in Cluttered Scenes. IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI), vol. 28(10), pp. 1584–1601, 2006. 2

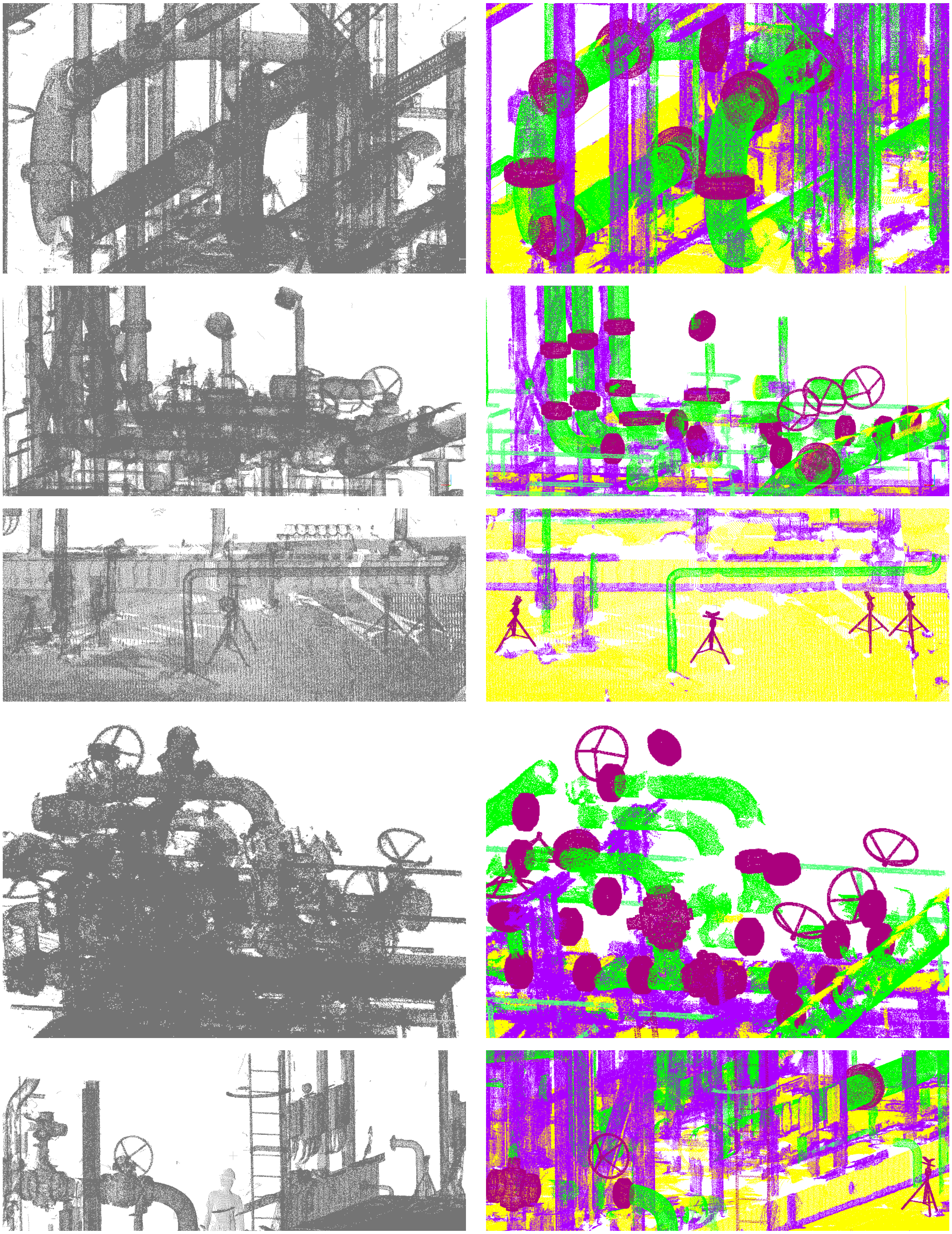


Figure 11. Detection results by the proposed method. The left column are the original point clouds, while the right column are the classification and detection results. The planes are shown in yellow, pipes shown in green, edges shown in light purple, and the detected parts are shown in dark purple. Despite the presence of large number of pipes (1st/2nd row), planes (3rd row) and clutters (2nd/4th/5th row), our system is capable of detecting all kinds targets ranging from flanges, hand wheels (2nd/4th row), tripods (3rd row) to large ladders (5th row).