ISPRS Journal of Photogrammetry and Remote Sensing 164 (2020) 97–105

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Large-scale point cloud contour extraction via 3D guided multi-conditional generative adversarial network

Weini Zhang[a], Linwei Chen[a], Zhangyue Xiong[a], Yu Zang[a,*], Jonathan Li[a,b], Lei Zhao[c]

[a] Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen, FJ 361005, China
[b] Department of Geography and Environmental Management and Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[c] Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, USA

ARTICLE INFO

Keywords:
Contour extraction
Multi-conditional GAN
Large-scale point cloud

ABSTRACT

As one of the most important features for human perception, contours are widely used in many graphics and mapping applications. However, for large outdoor scale point clouds, contour extraction is considerably challenging due to the huge, unstructured and irregular point space, thus leading to massive failure for existing approaches. In this paper, to generate contours consistent with human perception for outdoor scenes, we propose, for the first time, 3D guided multi-conditional GAN (3D-GMcGAN), a deep neural network based contour extraction network for large scale point clouds. Specifically, two ideas are proposed to enable the network to learn the distributions of labeled samples. First, a parametric space based framework is proposed via a novel similarity measurement of two parametric models. Such a framework significantly compresses the huge point data space, thus making it much easier for the network to "remember" target distribution. Second, to prevent network loss in the huge solution space, a guided learning framework is designed to assist finding the target contour distribution via an initial guidance. To evaluate the effectiveness of the pro-posed network, we open-sourced the first, to our knowledge, dataset for large scale point cloud with contour annotation information. Experimental results demonstrate that 3D-GMcGAN efficiently generates contours for the data with more than ten million points (about several minutes), while avoiding ad hoc stages or parameters. Also, the proposed framework produces minimal outliers and pseudo-contours, as suggested by comparisons with the state-of-the-art approaches.

1. Introduction

Recently, rapid development of Light Detection and Ranging (LiDAR) technology makes it possible to acquire 3D geospatial information for large-scale outdoor scenes identified as point clouds. Due to the unstructured, irregular, and non-uniform characteristics of raw point cloud data, various features must be extracted as the basis for further processing.

Unlike point-based features, which have received wide attention in previous efforts (Guo et al., 2014), contour extraction in real practice, especially for large-scale point cloud data greater than 10^8 points, is considerably challenging. A primary reason, as (Hackel et al., 2016) points out, is that the definition of "contour" is difficult to formalize with various rules, because a perceptible contour is actually relevant to complicated factors, such as sudden changes in curvature, sufficient length, uniform local directionality, etc.

In addition, as the scan of point clouds for large outdoor scenes tended to become less difficult and lower in cost, for the large-scale dense point clouds, previous research (Lin et al., 2017; Lin et al., 2015) focused primarily on "line features" rather than "contour" extraction, where line-like structures are detected by explicit and formalized rules (e.g. intersection of two nonparallel surfaces). Although using such a definition enables the detection of fine line primitives, the definition suffers from two problems: (1) When an operator plans to mark a contour on a point cloud, the decision-making process can be very complicated and involve many factors, such as curvature, continuity, directionality, etc. In this sense, "contour" should be described as a line like structure, consistent with human perception. (2) Feature based line extraction techniques often rely on pre-constructed models or surfaces. However, as (Hackel et al., 2016) points out, intuitively, contours must be extracted first. Then the contours are used to guide the further processes, such as surface reconstruction, semantic segmentation,

* Corresponding author.
E-mail addresses: wnzhang95@stu.xmu.edu.cn (W. Zhang), willimchen@foxmail.com (L. Chen), zyxiong@stu.xmu.edu.cn (Z. Xiong), zangyu7@126.com (Y. Zang), junli@xmu.edu.cn (J. Li), leizhao@illinois.edu (L. Zhao).

https://doi.org/10.1016/j.isprsjprs.2020.04.003
Received 4 December 2019; Received in revised form 28 March 2020; Accepted 6 April 2020
Available online 25 April 2020

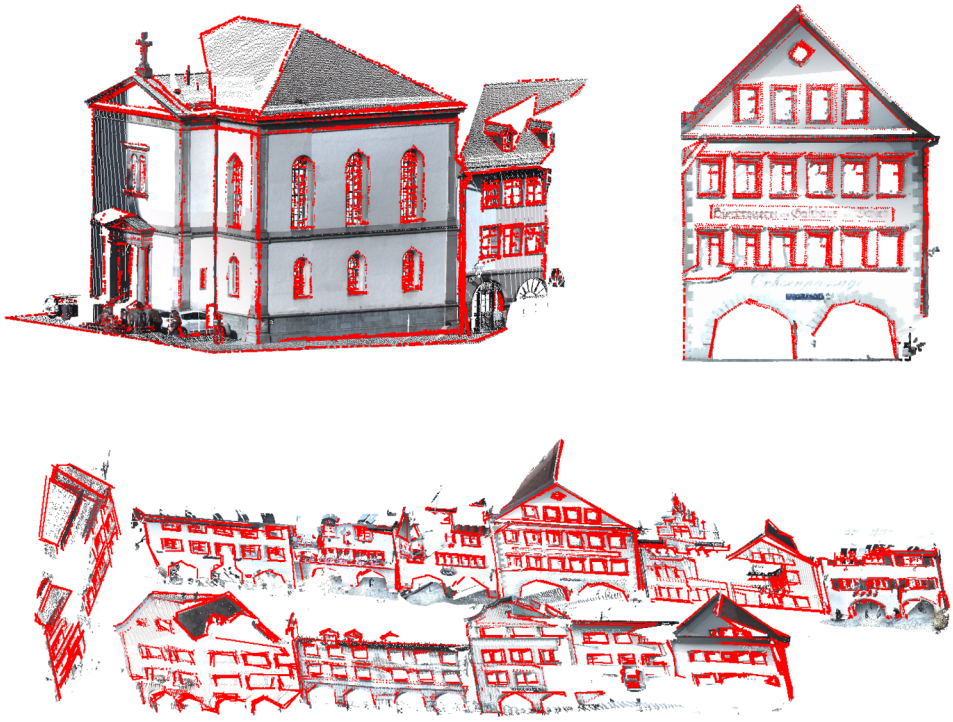


Fig. 1. Contour extraction for large-scale point clouds.

model refinement, etc.

To address these problems, recent point cloud contour detection methods provide some beneficial inspirations. Hackel et al. (2016) proposed the first learning based framework, which avoids the endless rule definition for complicated contour cases. Nevertheless, the training process is based on some pre-defined local features; thus, a high order Markov field is solved subsequently to obtain the final contours. Inspired by recent point cloud learning networks (Qi et al., 2017a; Qi et al., 2017b), in the following study, Yu et al. (2018) proposed a more direct way, "EC-Net", to extract contours. The aim of their work is to extract object contours by remembering the distance distribution from each point to the edges. This approach works in point space, resulting in making it possible for this approach to reach its limit for large-scale point cloud processing.

As it turns out, contour extraction for large scale point clouds is considerably challenging, due to huge, unstructured, and irregular point data. For such tasks, we propose a deep neural network based learning framework, 3D guided multi-conditional GAN (3D-GMcGAN), as shown in Fig. 1. The technique contributions of this paper rely on two types of efforts, aiming to make the network capable of remembering and generating the desired contour distribution consistent with human labeled ground truth.

First, because contours can be easily represented in parametric space, both the training and testing processes are designed to directly modify the parameters of the line like structures via a novel similarity measurement of two parametric models. This parametric space based learning framework significantly compresses the huge point data space, making it much easier for the network to "remember" the target distribution. Second, to prevent network loss in huge solution space and convergence to some bad local extrema, a guided learning framework is designed to assist finding the target contour distribution via an extra guidance branch in the network.

In additional, we have also open-sourced a public dataset, with considerable scale, for large scale point clouds contour extraction. Specifically, it is built based on a widely employed public dataset for large scale point cloud segmentation *Semantic3D* (Hackel et al., 2017), while with complete *contour annotation system*. Such a dataset, to our knowledge, is the first public dataset for large scale point cloud contour extraction, and can effectively support the dataset *Semantic3D*.

2. Related work

Contour extraction is a long-standing, yet still active topic, for 2D images (Von Gioi et al., 2010; Arbelaez et al., 2011) or 3D models reconstructed by multiview images (Heuel and Forstner, 2001; Jain et al., 2010; Attene et al., 2005). Based on the methodology for 2D images, early 3D line extraction methods basically followed a two-part definition-and-detection strategy (Demarsin et al., 2007; Schnabel et al., 2010): (i) Define what a line is according to some criteria; (ii) Then, detect the line structures by some strategies. Specifically, these works can be further divided into two types: multiview image based and surface fitting based methods.

Multiview image based methods. Due to the expensive cost of 3D laser scan sensors, early 3D models were often built from multiview stereo images (Schmid and Zisserman, 1997; Ok et al., 2012; Ceylan et al., 2012; Hofer et al., 2015). The first batch of 3D line extraction methods tended to consider the 3D model as a collection of 2D images from different views. Thus, the major problem is the matching of various lines over different views. By regarding the total squared distance between the observed 2D and projected 3D lines, Taylor and Kriegman (1995) reconstructed 3D lines according to a well-designed objective function. To match lines over multiple views, Heuel and Forstner (2001) used statistical hypothesis testing to explore relations between geometric entities points, lines, and corresponding half planes in 2D and 3D spaces. Inspired by the idea to represent line correspondences via a pre-defined matrix Matinec and Pajdla (2003), Jain et al. (2010), reconstructed 3D line segments from different stereo images, and then merged them by depth evaluation and connectivity constraints. By using shaded images, Lin et al. (2015) extracted line features by combining both the shaded images and point clouds.

Surface fitting based methods. As point cloud acquisition became much easier, the following works focused more on 3D scattered point cloud data. Based on implicit surface fitting, by defining the ridge-valley lines as curves along a sharply changed implicit surface of mesh

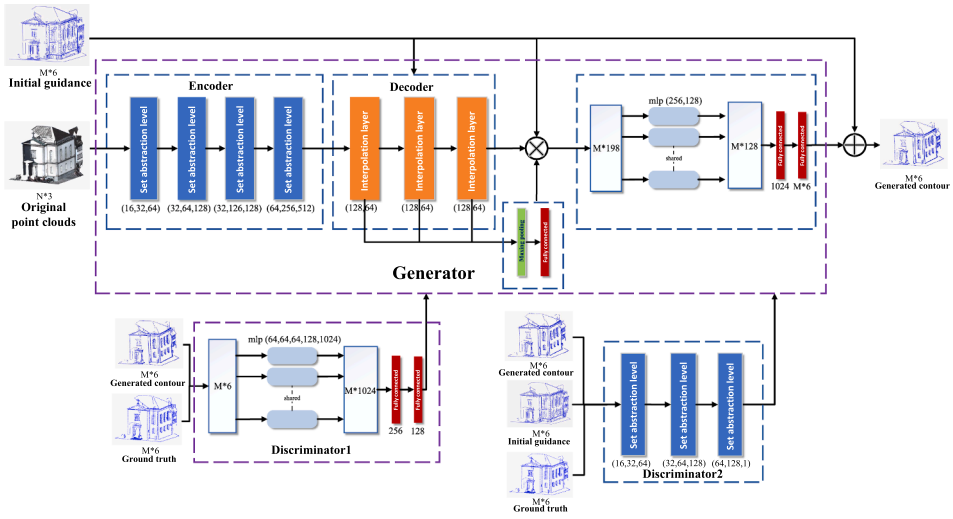


Fig. 2. Network architecture.

vertices, Ohtake et al. (2004) detected ridge vertices by estimating curvature tensors and derivatives on the fitted surface and then connected them to form the ridge-valley lines. Following such an idea, Kim (2013) fit the points via a modified moving least-squares (MLSs) and connected the ridge vertices along the principal curvature direction to extract ridge-valley lines. Daniels et al. (2007) proposed an approach to project locally fitted points onto the intersection of various surfaces and then derived 3D polylines from the projected points. Noting that most man-made scenes are piecewise-planar, plane fitting based methods extract line like structures by defining adjacent or discontinuous planes (Moghadam et al., 2013; Borges et al., 2010; Lin et al., 2017). As an alternative case to surface fitting based works, plane fitting based methods produce more robust results under strong noise and outliers.

In fact, most of these approaches essentially tend to formally define line like structures by combining features such as curvature, surface, plane etc. However, as (Hackel et al., 2016) points out, such a rule-based manner may lead to an adverse situation. A rule-based manner must struggle exhaustively to cover various expected types of lines, thus making it increasingly difficult to tune such a system manually for complicated large scale point clouds.

To address this problem, Timo Hackel et al. (2016) first proposed a learning based 3D contour extraction approach for outdoor point clouds, where each point is first predicted with a classifier to denote the contour-like likehood. Then, the optimal set of contours is selected by solving high-order MRFs. Inspired by recent deep neural networks for point clouds, such as PointNet (Qi et al., 2017a) and PointNet++ (Qi et al., 2017b), Yu et al. attempted an end-to-end framework for contour extraction of small point cloud objects. However, due to the highly complex data, such a distance bias based approach is not suitable for close-range large scale point clouds.

3. Method

The huge data space of large scale point clouds (i.e. more than 10^8 points) makes it considerably challenging to create a deep neural network based contour generation framework. On the one hand, when human operators label the contours of point clouds, they often employ local lines to approximate various curve contours, thus forming a natural parametric representation of training samples. On the other hand, by combining with previous works (Lin et al., 2017; Yu et al., 2018; Hackel et al., 2016), it is easy to acquire the initial line feature description for a point cloud.

Such observations inspire us to use a parametric space based guided learning framework, i.e., the goal of the learning network is to learn the parameter distribution of the human labeled contours and generate the parametric representation of the contours, rather than directly produce the contour points in the raw point space. Such a framework significantly compresses the huge data space, thus making it much easier for the network to "remember" the distributions of the human labeled samples.

Thus, we propose a 3D-GMcGAN, a guided learning network adapted for the processing of large scale point cloud data. With extra initial guidance, the network can effectively avoid convergence at some bad local extrema. Then, to evaluate whether the parameter distribution of the ground truth is appropriately simulated, a simple yet efficient measurement is designed to measure the similarity of the two parametric models.

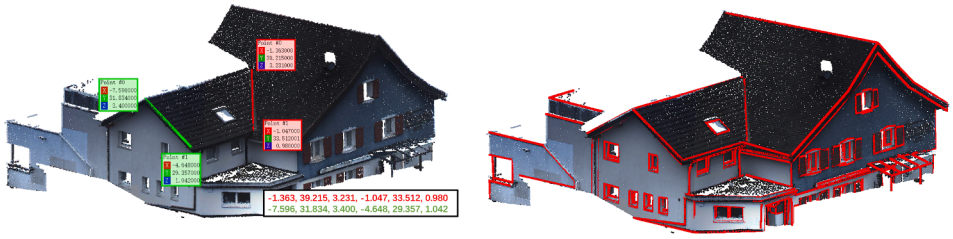
3.1. Network architecture

In the proposed 3D-GMcGAN, besides the original point cloud and human labeled ground truth, an initial line feature description (generated from previous work (Lin et al., 2017)) is employed to jointly guide the training of the network. Such guidance provides several benefits: first, the initial line feature description can be easily represented as parameters (6 parameters for each 3D line), thus making it possible to apply most of the operations in parametric space (except the feature extraction of the original point cloud). Second, the introduced guidance provides a natural initialization for the input data, thus enabling the network to generate the desired contours by calculating a set of "bia" to adjust the initial line distribution. Such a manner effectively prevents the network converging to some bad local extrema.

Denote the input point cloud, initial guidance, human labeled ground truth and the output of the generator as P, L, R and g respectively. Then, the number of points in P and the number of lines in L, R, g can be denoted as N and M. It should be noted that, except for the original point cloud data, P, the other data L, R and g are a set of lines, which can be easily manipulated in parametric space. The architecture of the network is shown in Fig. 2.

The whole network includes mainly two branches. The first is similar to traditional GAN, which aims to make the output of the generator g as similar as possible to ground truth R. The other branch, which is inspired by the "triplet loss" (Schroff et al., 2015), takes guidance into account and aims to maximize the distance between the output, g, and the guidance, L, while minimizing the distance between g and the ground truth, R. Then, the generator is jointly trained by the two branches to generate a set of bias to modify the guidance, L, to form the desired parameterized contour.

Specifically, in our proposed network, one generator (denoted as G) and two discriminators (denoted as D_1 and D_2) are involved. Due to the



(a) Label description (b) Labeled data

Fig. 3. Label description and visualization of labels.

Fig. 4. Some samples of our labeled data.

irregular and non-uniform characteristics of the scanned point cloud, it is difficult to process the point cloud by traditional Convolutional Neural Networks (CNN). Encouraged by the point learning network, PointNet++ (Qi et al., 2017b), the generator of our approach contains mainly three parts.

Encoder. The first part is an encoder, in which four set abstraction levels are employed to extract the features of the input point cloud, P. Each level of the encoder contains three sublayers: a sampling layer, a grouping layer and a PointNet layer. The aim of this part is to build a hierarchical grouping of the point features.

Decoder. The second part is the decoder, which consists of three interpolation layers, one max pooling layer, and a fully connected layer. The interpolation layer, which is guided by the parameterized initial line distribution, L, aims to extract features according to the constraint of the guidance and the skipped links from the encoding layers. Then, based on the attention mechanism, the four hierarchical features are combined. Such a mechanism is essentially the weighted sum of the four decoding layers, where the weights are derived from the output of the max pooling layer and the fully connected layer.

1D-convolutional layers and connected layers. The third part of the generator contains two 1D-convolutional layers and two fully connected layers. Such a design aims to promote the feature fusion between various local areas, thus making the network more capable to learn the desired distribution. In our experiments, it was found that the fully connected layer is very important for a satisfied generation result. We assume the reason is that such a layer effectively builds the connection between the feature of various local areas, thus significantly expanding the ability of the learning network. It should also be noted that, in point space, such a fully connected layer can be quite resource consuming, thus making it extremely difficult for application to large-scale point cloud data.

The first discriminator, based on the PointNet, is composed of five 1D-convolutional layers, one max pooling layer, and two fully connected layers. The major task of D_1 is to identify whether the generated contour is satisfied. The other discriminator, D_2 is designed to take into account the relationship of the original point cloud P, the guidance L and the generated parameterized contour g. Encouraged by the idea of "triplet loss" (Schroff et al., 2015), the output contour, g, is considered as anchor samples; the ground truth, R, and the guidance, L, are considered as positive and negative samples, respectively. Then, the aim of G is to force the generated contour, g, to be similar to the ground truth R, yet different from the guidance L.Mathematically, the distance $d(g, R)$ is minimized, while the distance $d(g, L)$ is maximized. The structure of D_2 is similar to the encoder, where three set abstraction levels are involved, which enables it to extract the hierarchical spatial distribution of g and R.

3.2. Network loss functions

To form a learning framework in parametric space, the most important thing is to evaluate the similarity between the generated contour and human labeled reference. To achieve this, first two primary generative losses are designed: the model similarity metric and the parameter similarity metric.

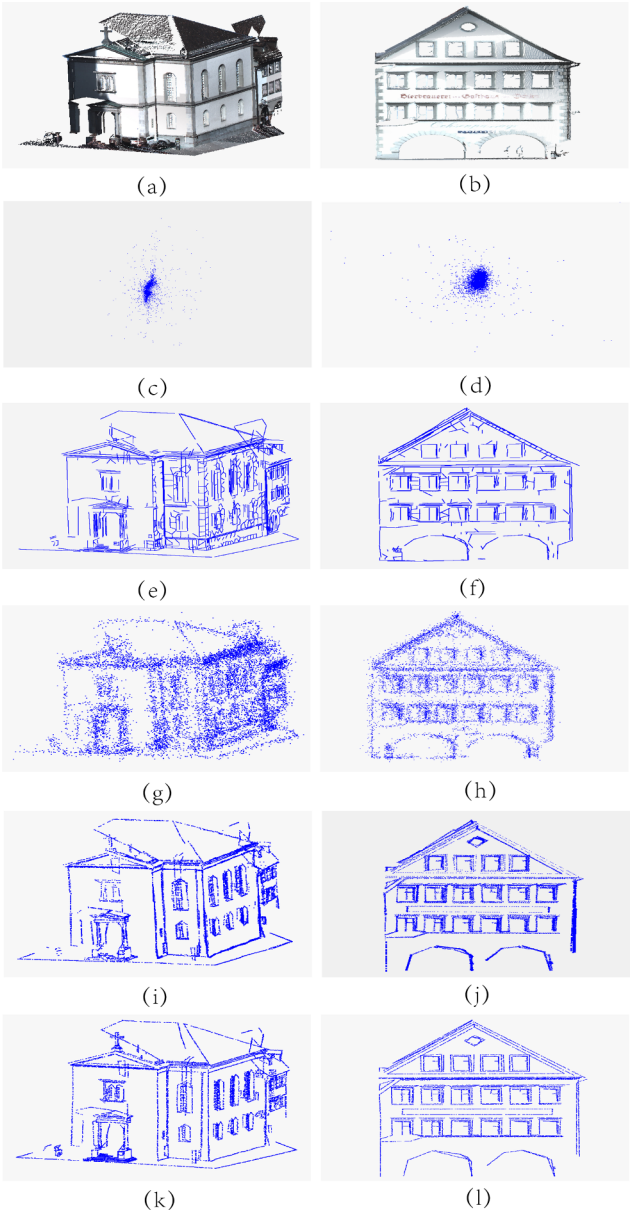


Fig. 5. Results of network with different structures. (a)(b) are the input point cloud slices. (c)(d) are the results of network working in parametric space with no guidance. (e)(f) are the results of previous work (Lin et al., 2017). (g)(h) are the results of network working in point space with guidance. (i)(j) are the results of our approach working in parametric space with guidance. (k)(l) are th.e ground truths.

Model similarity metric. The model similarity metric aims to directly measure the similarity of two line models derived from the parameters. Thus, we propose a Mean Measure (MM) to evaluate the similarity between the generator outputs, g, and the human labeled ground truth, R. Here, our goal is to approximate the model, R, by adjusting the data, g. The proposed model similarity metric, $L_l(\cdot,\cdot)$, is defined as the ratio of the error from the data to the model (EDM) and the measure of the data $|g|$:

$$L_l(G) = \frac{\sum_{x \in g} d_m(x, R)}{n_g |l_g| + \epsilon} \tag{1}$$

where $d_m(\cdot,\cdot)$ represents the EDM, n_g is the total number of sampled points in g;l_g denotes the total length of the data g. To prevent the zero divisor, ϵ is applied. It should be noted that both g and R are represented as parameters; therefore, in our implementation, to calculate EDM, we first uniformly sample the lines in g and R.

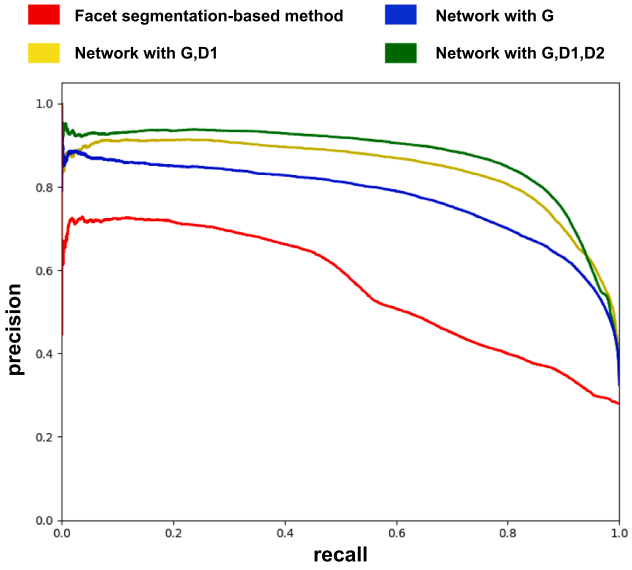


Fig. 6. Precision-recall curves of previous work (Lin et al., 2017) and networks with differen.t architectures.

According to Nyquist's law of sampling, the sampling rate of g should be twice that of R. Then EDM is calculated as the average of the minimum distance for each sample point from g to R.

Our proposed MM, which is insensitive to noise, is quite suitable for estimating the similarity of the two models.

Theoretically, it should be sufficient to build the loss by applying only the model similarity metric; however, in practice, we found that because the contour distribution of the large-scale point could be very complicated, employing only the model similarity metric, MM, may not prevent outliers or pseudo contours. Thus, we propose the parameter similarity metric to jointly measure the similarity between g and R.

Parameter similarity metric. The aim of such a metric is to measure the similarity of the parameter distribution between g and R.

A 3D line, defined by two points in 3D space, can be represented as a six dimension vector. Then, the parametric representations of g and R can be considered as two point sets in 6D space, and the parameter similarity metric essentially measures the distance between two 6D point sets. In our approach, the Chamfer Distance (CD) (Fan et al., 2017), which is a commonly applied metric for two point sets, is employed to build this loss. Here, to adapt the three dimensional CD metric to our task, we modify it to six dimensions. Because the scale of the parametric space is quite small, such a loss can be calculated rapidly. Specifically, loss is defined as follows:

$$\begin{aligned} L_p(G) = &\frac{1}{n_g} \sum_{x \in g} \min_{y \in R} \|x - y\| \\ &+ \frac{1}{n_R} \sum_{y \in R} \min_{x \in g} \|y - x\| \end{aligned} \tag{2}$$

where n_g and n_R represent the number of points in g and R, respectively. $\|\cdot\|$ represents l_2 distance.

Then, for the discriminator D_1, traditional conditional adversarial (Mirza and Osindero, 2014) loss is employed. It is defined as follows:

$$\begin{aligned} L_{d1}(D_1) = &E_{y,x,c \in P_d(R,P,L)}[(1 - D_1(y|x, c))^2] \\ &+ E_{x,c \in P_d(P,L)}[D_1(G(x, c)|x, c)^2] \end{aligned} \tag{3}$$

$$L_{d1}(G) = E_{x,c \in P_d(P,L)}[(1 - D_1(G(x, c)|x, c))^2] \tag{4}$$

where x, y and c represent the sampled points corresponding to P, R and L, respectively. P_d represents the joint distribution of the input data.

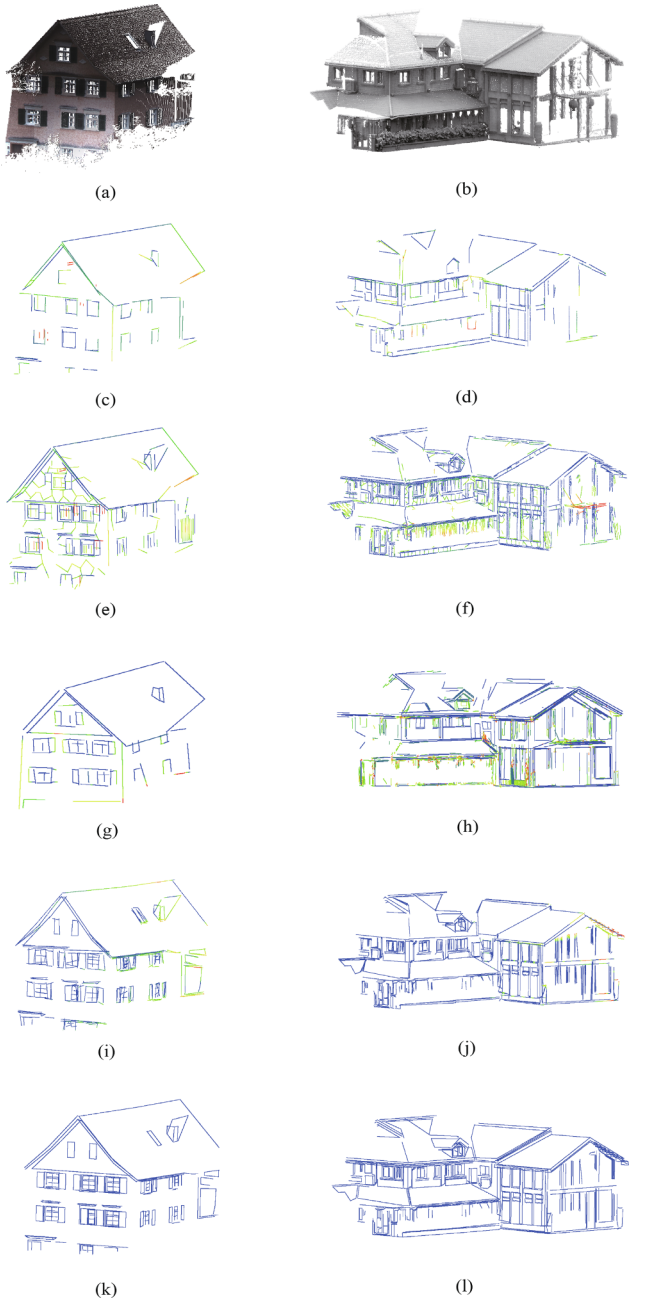


Fig. 7. Comparisons to previous works. (a)(b) are the input point cloud slices. (c)(d), (e)(f), (g)(h), (i)(j) are the results of Lin et al. (2017), Borges et al. (2010), Lu et al. (2019) and our approach, respectively. (k)(l) are the .ground truths..

For the loss of discriminator D_2, encouraged by the "triplet loss", three inputs g, L, R are fed into the discriminator, and the corresponding output feature vectors are considered as anchors, positive samples and negative samples, respectively. The aim of loss $L_{d2}(D_2)$ is to minimize the difference between the distance $d(g, L)$ and $d(g, R)$, which can be formalized as follows:

$$L_{d2}(D_2) = \max_{y,x,c\in P_{tl}(R,P,L)} (0, \alpha_1 - d(D_2(G(x, c)|x, c), D_2(y)) + d(D_2(G(x, c)|x, c), D_2(c))) \quad (5)$$

On the one hand, for generator G, the aim of loss $L_{d2}(G)$, which is opposite to $L_{d2}(D_2)$, is to maximize the difference between the distance $d(g, L)$ and $d(g, R)$. Specifically, the loss is formalized as follows:

$$L_{d2}(G) = \max_{y,x,c\in P_{tl}(R,P,L)} (0, \alpha_2 + d(D_2(G(x, c)|x, c), D_2(y)) - d(D_2(G(x, c)|x, c), D_2(c))) \quad (6)$$

where $d(\cdot,\cdot)$ represents the l_2 distance. Both α_1 and α_2 are set at 1.

With the above designed losses, the overall objective of our network can be written as follows:

$$G^* = \arg\min_G [k_1 L_p(G) + k_2 L_l(G) + L_{d1}(G) + L_{d2}(G)] \quad (7)$$

$$D_1^* = \arg\min_{D_1} L_{d1}(D_1) \quad (8)$$

$$D_2^* = \arg\min_{D_2} L_{d2}(D_2) \quad (9)$$

where, generator G is jointly trained by terms $L_{d1}(G)$, $L_{d2}(G)$, $L_p(G)$ and $L_l(G)$ to generate the contours as close as possible to ground truth, R. The discriminators D_1 and D_2 are trained by the terms $L_{d1}(D_1)$ and $L_{d2}(D_2)$ respectively, to identify the output of the generator, g, thus forming the adversarial process.

4. Results and analysis

4.1. Dataset and Implementation details

In our experiments, two sets of data were selected to comprehensively evaluate our proposed approach. To our knowledge, there are several outdoor large-scale 3D point cloud datasets for classification and localization. However, no such dataset with contour annotation information is publicly available. So we first create a dataset, with considerate scale, for the large scale point cloud contour extraction. Specifically, such a data set is consist of two parts.

The first part is acquired by our RIEGL VMX-450 MLS system (with two full-view RIEGL VQ-450 laser scanners, and can produce 1.1 million range measurements per second, capable to acquire nearly 100 GB point clouds data in 1 h.) and RIEGL VZ-1000 TLS system (with the accuracy of 8 mm, precision of 5 mm, scan range varies from 2.5 to 1000 m). Most of these data are collected in various regions in China, and cover different scenes such as urban, town, village, etc. The other part, is based on a public dataset *semantic3D.net* (Hackel et al., 2017), which covers various urban scenes with a total of over four billion points. For both of the dataset, we remove some objects which are extremely incomplete or difficult to recognize the contours even for human, such as the vegetation, pedestrians and hard scape like garden walls, fountains, etc.

Different from previous work (Hackel et al., 2016), rather than directly labeling the points on contours, we label the contours with 3D lines represented as parameters (as mentioned in Section 3.1). Such annotations can not only simply generate contour points used in Hackel et al. (2016) by sampling the points near the labeled contour lines, but also provide parametric contour information. For both of the dataset, we tend to label the objects with clear contour structure, such as buildings, roads, cars, etc., and use several short lines to fit curved contours. For the objects with slightly incomplete parts caused by occlusion, we will complete the contour when labeling, otherwise, the incomplete parts will be ignored. Fig. 3 shows the labeling process and some of the examples. During contour labeling, we label the two endpoints of a fitted line segment, whose coordinates are concatenated as the parametric label for this line segment(shown in the bottom right of Fig. 3(a)). Fig. 3(b) shows the complete labeled contours with red lines on the original point cloud. More samples can be viewed in Fig. 4.

The human labeled reference contours were created by eight operators working for two months. The overall scale of the effective training data is more than 1 TB. Here, we have already open-sourced the results for the large scale point cloud semantic labeling data set [1],

[1] http://www.semantic3d.net/.

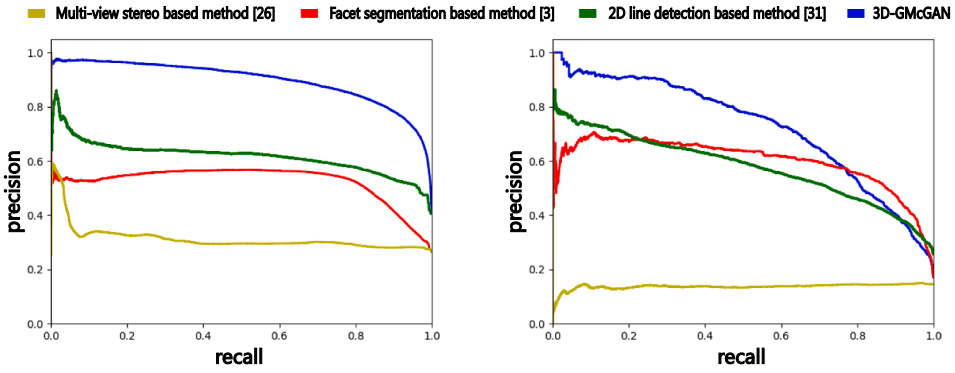


Fig. 8. Precision-recall curves of different methods for the point clouds in Fig. 7(a)(b).

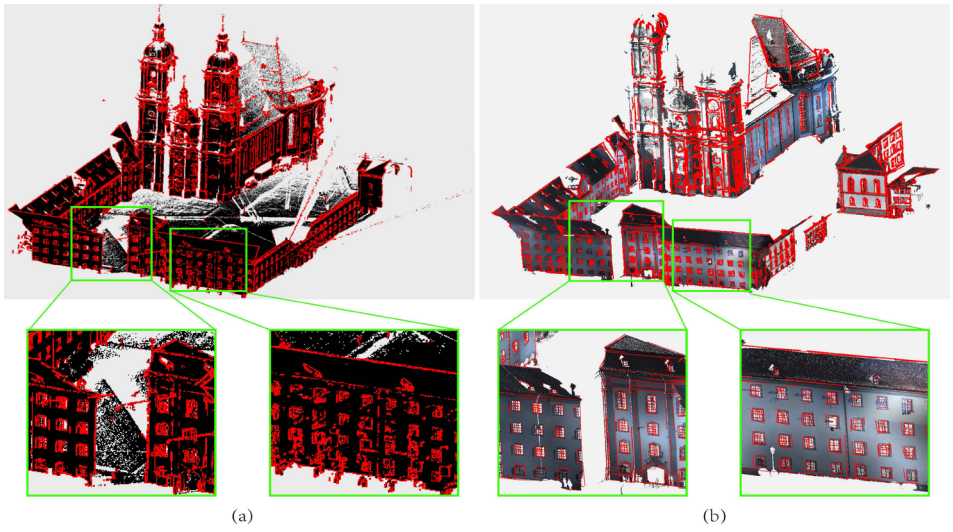


Fig. 9. Comparison to previous work (Hackel et al., 2016). (a) is the result of Hackel et al. (2016). (b) is the result .of our approach.

Table 1
Computing time of Lin et al. (2017), Lu et al. (2019) and our approach on point clouds of different scales.

Point cloud	Number of points	Running time(s)		
		Lin et al. (2017)	Lu et al. (2019)	3D-GMcGAN
Fig. 7(a)	40874	0.513	0.176	15.932
Fig. 7(b)	782197	8.756	3.425	273.3
Fig. 9	31179770	527.105	142.194	11055.103

and can be downloaded from [2]. For the rest of the data, we plan to open it after approved by the national government.

In this work, TensorFlow framework is used to build the entire network on a PC with one Titan X GPU. The training process is based on Adam solver (Kingma and Ba, 2014); the learning rate is 0.001. The weights of the networks are initialized from a Gaussian distribution with mean, $\mu = 0$ and standard deviation, $\sigma = 0.02$. The number of training epochs is set at 300; the learning rate linearly decreases to zero. The weights k_1 and k_2 are set at 1 and 0.2, respectively. By considering a huge number of original points and the limited capability of the GPU, the density of the input point cloud data is reduced first; the ratio depends on the memory of the GPU and the number of points.

Using the two datasets, two groups of experiments were performed to comprehensively evaluate the proposed approach. The purpose of the first group of experiments is to evaluate how different parts of the losses affect the extraction results. Then, using both of the datasets, we compared our approach with several state-of-the-art approaches. We present typical results, along with quantitative evaluations, to show the performance of our proposed learning framework.

4.2. Analysis of the network structure

Evaluation. For the quantitative measurement, precision-recall curves (Arbelaez et al., 2011) are employed to evaluate the performance of various methods. Here, precision is defined as the probability that a point on an extracted contour belongs to the ground truth; recall is defined as the probability that a ground truth point lies on an extracted contour. Actually, the original input point cloud is binary labeled as contour or non-contour points. For each contour point, defined by our generated result, we calculate the minimum distance to the corresponding point in the ground truth, $d_p(g, R)$. Then, a distance threshold, t, is designated to determine whether a point lies on the extracted contours; we draw the precision-recall curve as moving from the minimum to maximum value of $d_p(g, R)$. It is readily seen that high precision means low false positives; whereas, high recall means low false negatives. The ideal curve has both high precision and high recall, as indicated by the larger area under the curve.

The solution space for a 3D case in a learning task is far more complicated than for a 2D case. Also, the scale of points for an outdoor scene could be huge (often more than 10^8 for a middle-sized building acquired by RIEGL VMX-450 or VZ-1000 scanning systems), thereby making it surprisingly difficult for the network to find a satisfactory

[2] ftp://182.61.174.17/contour_semantic3D/.

approximation of the desired ground truth.

To address such a problem, we propose two ideas in this work to help the network. (1) Manipulated spaces are simplified to parametric spaces, thereby significantly compressing the potential solution spaces. (2) A guided learning framework is proposed to introduce an initial distribution to assist the network to find the satisfactory solution.

To evaluate the effectiveness of the two proposed schemes, in this part, two groups of experiments are applied: (1) manipulate data in point space. (2) use network without the guidance branch. Both of the input point clouds, shown in Fig. 5(a)(b), are sliced data cut from the large scale point cloud of the outdoor scene. For each slice, 696,196 and 275,435 points are involved. The results of directly working in parametric space, with no guidance, are shown in Fig. 5(c)(d). Here, the network does not remember the distribution of training samples and obtains bad results. Shown in Fig. 5(e)(f) are the results of previous work (Lin et al., 2017), where chaotic line like features are detected and some pseudo contours are apparent. With such results as a guide, the point space manipulated results, where CD distance is employed to measure the similarity of two point cloud models, are shown in Fig. 5(g)(h). Because of the huge solution space, it is difficult to recognize the contours from the large number of outliers. The results of our approach, with the proposed guided learning framework in parametric space, are shown in Fig. 5 (i)(j). This network denotes the contours consistent with the human labeled ground truths, which are shown in Fig. 5(k)(l).

To study the contributions of the generator and discriminators in our network, we conduct ablation experiments with different network architecture: (1) network with only the generator G. (2) network with the generator G and the discriminator D_1. (3) network with the generator G and the discriminator D_1 and D_2. For all experiments, we manipulate data in point space and use the network with guidance branch. Fig. 6 shows the precision-recall curves of previous work (Lin et al., 2017) and these three groups of experiments on test set. Compared to previous work (Lin et al., 2017) (the red curve in Fig. 6), our network with only the generator G (the blue curve in Fig. 6) has higher precision and recall. The yellow curve and green curve show the result of our network with G, D_1 and G, D_1, D_2 respectively. It is seen that adding discriminator D_2 can improve the performance of our network.

4.3. Comparisons with state-of-the-art approaches

To comprehensively evaluate our proposed approach, we compare it to the latest methods (Lin et al., 2017; Borges et al., 2010; Hackel et al., 2016; Lu et al., 2019). In the first group of results shown in Fig. 7, previous feature based works (Lin et al., 2017; Borges et al., 2010; Lu et al., 2019) are used for comparison. To better compare each method, we incorporate a blue-to-red image (Lin et al., 2017) sequence to visualize contour extraction results. Various colors indicate the distance from each point in an extracted contour to its nearest ground truth point. Red represents the highest distance; blue represents the lowest. The input point cloud slices are shown in Fig. 7(a)(b). The left cloud slice is from the dataset of Hackel et al. (2016); the right one was acquired by our VMX-450 systems. The results of Borges et al. (2010), Lin et al. (2017), Lu et al. (2019) and our approach are shown in Fig. 7(c)(d), (e)(f), (g)(h) and (i)(j), respectively. Ground truth is shown in Fig. 7(k)(l). From Fig. 7(c)(d), it is seen that some salient contours are missing from the results for Borges et al. (2010), thus leading to incomplete extraction results. From Fig. 7(e)(f), it is seen that, due to the line features defined as local plane intersections, there are visible pseudo contours in the results for Lin et al. (2017). From Fig. 7(g)(h), it is viewed that the results of previous work (Lu et al., 2019) are largely relied on the 2D contour detection process, due to the 3D-2D projection. So some of the contours may miss when the point cloud is sparse, as shown in Fig. 7(g); or the results will be sensitive to the surface textures when the point cloud is dense, due to the CannyLines detector. In contrast, our approach (Fig. 7(i)(j)) provides contours consistent with human labeled ground truth, while barely producing pseudo lines.

Then, corresponding precision-recall curves are shown in Fig. 8, where the precision-recall curves of previous works (Lin et al., 2017; Borges et al., 2010; Lu et al., 2019) and our approach for the two examples in Fig. 7(a)(b) are presented. It is seen that, for the previous works (Borges et al., 2010; Lin et al., 2017; Lu et al., 2019), because of incomplete extractionor or highly false extracted results, precision is relatively low; whereas, most often, the results of our approach are visibly better than those of the two competitors.

In the second group of results, the proposed 3D-GMcGAN is compared with the latest large-scale point cloud contour detection approach based on learning framework (Hackel et al., 2016). Because we did not have labeled training samples, our network, showing an intuitive comparison, was run on a typical large-scale point cloud in dataset (Hackel et al., 2016). Shown in Fig. 9(a) are the results of Hackel et al. (2016) with the marked contour points on the original data (For better viewing, some patches are zoomed in.). The corresponding results of our approach are shown in Fig. 9(b). Intuitively, we see that our results, which are less sensitive to fine local structures, provide more explicit contours. Actually, although both of the works are based on a learning framework, the mechanisms are quite different: for the work of Hackel et al. (2016), a simple classifier is first trained by some pre-defined local features and applied to provide the likelihood of each point belonging to the contour. Then optimal contours are generated by solving high order MRFs. Our approach, which tends to conduct a deep neural network based guided learning framework, provides a convenient solution for contour detection of large-scale point clouds.

In terms of computational complexity, we show the computing time of previous work (Lin et al., 2017; Lu et al., 2019) and our approach on point clouds of different scales. Lin et al. (2017) and Lu et al. (2019) are the feature based approaches without learning, while our approach is based on a designed deep neural network that requires training on GPUs. We use the source code for Lin et al. (2017), Lu et al. (2019) provided by authors and conduct the experiment on a PC with Intel Core i5-9400 2.9 GHz CPU and 16 GB RAM. Table 1 demonstrates the statistic results of examples in Fig. 7(a)(b) and Fig. 9. From Table 1 it is seen that feature based approaches(Lin et al., 2017; Lu et al., 2019) are faster than learning based approach. However, when the computing resources are sufficient, learning based method can provide results more consistant with human perception, while without complicated hyperparameter optimization.

5. Conclusion

In this paper, we proposed 3D guided multi-conditional GAN (3D-GMcGAN), the first deep neural network based learning network for large scale point cloud contour extraction. The contributions of this paper rely on two aspects: (1) Both the training and testing processes are designed to directly modify the parameters of the line-like structures via a novel similarity measurement of two parametric models. Such a parametric space based learning framework significantly compresses the huge point data space, making it much easier for the network to “remember” the target distribution. (2) To prevent network loss in the huge solution space and convergence to some bad local extrema, a guided learning framework was designed to assist finding the target contour distribution via an extra guidance branch in the network.

Our approach is the first contour extraction framework for large scale point clouds based on a deep neural network. Huge labeled training samples are the major limitation of such a task. We have labeled our outdoor large scale point cloud data for various kinds of cities, towns or villages acquired by RIEGL VMX-450 MLS and VZ-1000 TLS systems and a widely employed public dataset *Semantic3D*. After two months of labeling by eight professional operators, we obtained about 1 TB of labeled data, and we have open-sourced the first dataset based on *Semantic3D* with contour annotation information for large scale point cloud. In the future, more labeled data is required, and we still seek to have our data released.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We would like to thank the anonymous reviewers for their valuable comments. This work is partly supported by the National Natural Science Foundation of China, with number 61971363.

References

Arbelaez, P., Maire, M., Fowlkes, C., Malik, J., 2011. Contour detection and hierarchical image segmentation. IEEE Trans. Pattern Anal. Machine Intell. 33 (5), 898–916.

Attene, M., Falcidieno, B., Rossignac, J., Spagnuolo, M., 2005. Sharpen&bend: recovering curved sharp edges in triangle meshes produced by feature-insensitive sampling. IEEE Trans. Visualizat. Comput. Graph. 11 (2), 181–192.

Borges, P., Zlot, R., Bosse, M., Nuske, S., Tews, A., 2010. Vision-based localization using an edge map extracted from 3d laser range data. In: 2010 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 4902–4909.

Ceylan, D., Mitra, N.J., Li, H., Weise, T., Pauly, M., 2012. Factored facade acquisition using symmetric line arrangements. Comput. Graphics Forum 31 (2pt3), 671–680.

Daniels, J.I., Ha, L.K., Ochotta, T., Silva, C.T., 2007. Robust smooth feature extraction from point clouds. In: IEEE International Conference on Shape Modeling and Applications, 2007. SMI'07. IEEE, pp. 123–136.

Demarsin, K., Vanderstraeten, D., Volodine, T., Roose, D., 2007. Detection of closed sharp edges in point clouds using normal estimation and graph theory. Comput. Aided Des. 39 (4), 276–283.

Fan, H., Su, H., Guibas, L.J., 2017. A point set generation network for 3d object reconstruction from a single image. In: CVPR, vol. 2, p. 6.

Guo, Y., Bennamoun, M., Sohel, F., Lu, M., Wan, J., 2014. 3d object recognition in cluttered scenes with local surface features: a survey. IEEE Trans. Pattern Anal. Mach. Intell. 36 (11), 2270–2287.

Hackel, T., Wegner, J.D., Schindler, K., 2016. Contour detection in unstructured 3d point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1610–1618.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M., 2017. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-1-W1, pp. 91–98.

Heuel, S., Forstner, W., 2001. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001, vol. 2, IEEE, pp. II–II.

Hofer, M., Maurer, M., Bischof, H., 2015. Line3d: Efficient 3d scene abstraction for the built environment. In: German Conference on Pattern Recognition, pp. 237–248.

Jain, A., Kurz, C., Thormählen, T., Seidel, H.-P., 2010. Exploiting global connectivity constraints for reconstruction of 3d line segments from images. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 1586–1593.

Kim, S.-K., 2013. Extraction of ridge and valley lines from unorganized points. Multimedia Tools Appl. 63 (1), 265–279.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

Lin, Y., Wang, C., Cheng, J., Chen, B., Jia, F., Chen, Z., Li, J., 2015. Line segment extraction for large scale unorganized point clouds. ISPRS J. Photogramm. Remote Sens. 102, 172–183.

Lin, Y., Wang, C., Chen, B., Zai, D., Li, J., 2017. Facet segmentation-based line segment extraction for large-scale point clouds. IEEE Trans. Geosci. Remote Sens. 55 (9), 4839–4854.

Lu, X., Liu, Y., Li, K., 2019. Fast 3d line segment detection from unorganized point cloud, arXiv preprint arXiv:1901.02532.

Matinec, D., Pajdla, T., 2003. Line reconstruction from many perspective images by factorization. In: Proceedings. 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, vol. 1, IEEE, pp. I–I.

Mirza, M., Osindero, S., 2014. Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.

Moghadam, P., Bosse, M., Zlot, R., 2013. Line-based extrinsic calibration of range and image sensors. In: 2013 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 3685–3691.

Ohtake, Y., Belyaev, A., Seidel, H.-P., 2004. Ridge-valley lines on meshes via implicit surface fitting. In: ACM transactions on graphics (TOG), vol. 23, ACM, pp. 609–612.

Ok, A.O., Wegner, J.D., Heipke, C., Rottensteiner, F., Soergel, U., Toprak, V., 2012. Matching of straight line segments from aerial stereo images of urban areas. Isprs J. Photogramm. Remote Sens. 74 (6), 133–152.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proc. Computer Vision and Pattern Recognition (CVPR). IEEE 1(2), 4.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in Neural Information Processing Systems, pp. 5099–5108.

Schmid, C., Zisserman, A., 1997. Automatic line matching across views. In: Conference on Computer Vision and Pattern Recognition, pp. 666.

Schnabel, R., Wahl, R., Klein, R., 2010. Efficient ransac for point-cloud shape detection. Comput. Graphics Forum 26 (2), 214–226.

Schroff, F., Kalenichenko, D., Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Taylor, C.J., Kriegman, D.J., 1995. Structure and motion from line segments in multiple images. IEEE Trans. Pattern Anal. Mach. Intell. 17 (11), 1021–1032.

Von Gioi, R.G., Jakubowicz, J., Morel, J.-M., Randall, G., 2010. Lsd: A fast line segment detector with a false detection control. IEEE Trans. Pattern Anal. Machine Intell. 32 (4), 722–732.

Yu, L., Li, X., Fu, C.-W., Cohen-Or, D., Heng, P.-A., 2018. Ec-net: an edge-aware point set consolidation network. In: ECCV.