Conditional Adversarial Networks for Multimodal Photo-Realistic Point Cloud Rendering

Torben Peters[1] **and Claus Brenner**[1]

1 **Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany**
{peters,brenner}@ikg.uni-hannover.de

Abstract

We investigate whether conditional generative adversarial networks (C-GANs) are suitable for point cloud rendering. For this purpose, we created a dataset containing approximately 150,000 renderings of point cloud image pairs. The dataset was recorded using our mobile mapping system, with capture dates that spread across one year. By parameterizing the recording date, we are showing that it is possible to predict realistically looking views for different seasons, from the same input point cloud.

1998 ACM Subject Classification D.1.3 Concurrent Programming, I.2.6 Learning, I.2.10 Vision and Scene Understanding, I.3.3 Picture/Image Generation, I.4.8 Scene Analysis.

Keywords and phrases Deep learning, MapReduce, GAN, image-to-image translation.

1 Introduction

Laser scanned point clouds are difficult to handle when it comes to photo-realistic rendering. First, a camera calibration is needed in order to colorize each scanned point. However, this does not guarantee that each 3d point is captured by a camera viewpoint. Secondly, since point clouds are sparse, it is difficult to exclude occluded points, e.g. behind walls and buildings. In order to create a continuous surface, splats can be drawn instead of points. A splat is defined as elliptical surface with a size according to the local point density. Lastly, the colorized point cloud does not contain any information about the sky and lighting.
Our approach tries to circumvent the whole process of model-based point cloud rendering, by learning how a possible representation of the point cloud could look like in reality. Our key contributions in this work are:

- Predicting photo-realistic views from point clouds which are containing only (laser) reflectance information.
- Extending a C-GAN to parameterize different seasons and months in order to predict multimodal images.

2 Related work

Conditional GANs attracted a lot of attention in recent years. Most notable are the *pix2pix* network by Isola et al. [3] and the improved version *pix2pixhd* [10], which is able to predict high resolution image-to-image mappings. Like traditional GANs, these networks are using an adversarial loss which is learned by a discriminator network. In contrast to L_1-loss, the adversarial loss leads to less blurry images [3, 10] by learning to distinguish between real and generated images. In addition to image-to-image translation, there exist a wide range

Spatial big data and machine learning in GIScience, Workshop at GIScience 2018, Melbourne.
Editors: Martin Raubal, Shaowen Wang, Mengyu Guo, David Jonietz, Peter Kiefer;

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

of conditional GANs, such as unpaired image-to-image translation [14] and text-to-image [2, 12, 7, 8].

Multimodal image-to-image translation defines the process of mapping one-to-many images, by modeling distributions of possible outcomes with an additional latent space vector or matrix, e.g. as used by *BicycleGAN* [15]. Other notable contributions are *iGAN* [13] and *Scribbler* [9] which directly encode the information.

Depending on the application, **point clouds** are handled in different ways in order to incorporate them into deep neural networks. They can be inferred as unordered 3d point sets [5, 6], using voxels [11, 4], or projected into images in order to use traditional 2d-convolutional networks [1].

3 Method

3.1 Preparing the dataset using MapReduce

To prepare a training dataset, we used mobile mapping data which was captured during 21 measurement campaigns, over the duration of one year. To illustrate the size of the problem, the subset of the data we are using contains 15 billion $(15,017,586,980)$ 3d points and $123,047$ images. Each image capture is given in terms of position (in UTM coordinates) and orientation (roll, pitch, yaw angles). Additionally, the intrinsic parameters of each camera are known. The task is then to project each of the 15 billion 3d points to each of the $123,047$ images. To solve this task, we created a massively parallel point cloud renderer, using the MapReduce framework on an Apache Hadoop cluster.

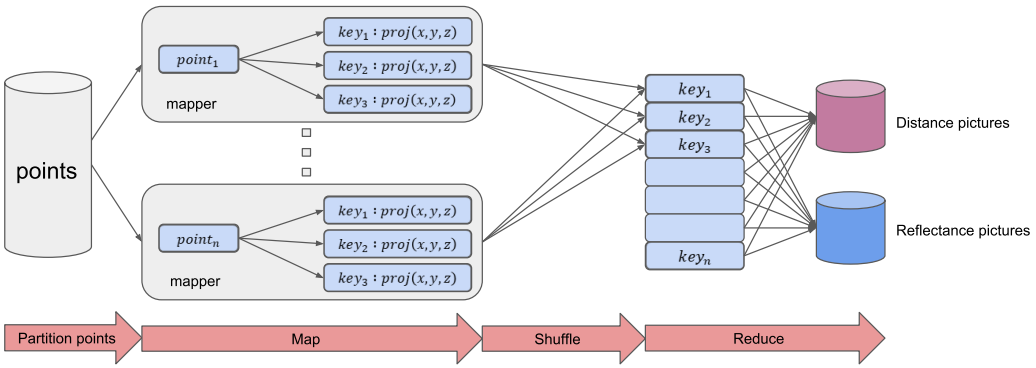

Figure 1 Our MapReduce approach for rendering large point clouds.

In order to apply MapReduce, each mapper has a list of all image orientations. According to the MapReduce principle, it receives a subset (split) of 3d point coordinates and their reflectance values (reflectance is an entity measured by the laser scanner). In order to reduce the amount of points emitted by the mapper, we exclude points that are behind the camera or are further away than 500 m. The mapper possibly emits multiple key-value pairs per incoming 3d point, depending on the number of images the point appears in. The key is defined by the image name, identifying a single image take, whereas the value contains the distance, reflectance and the image coordinates of the point. Each reducer receives all necessary information, grouped by image (key), and computes two 16-bit gray-value images per key, one containing the distance and the other one containing the reflectance values per

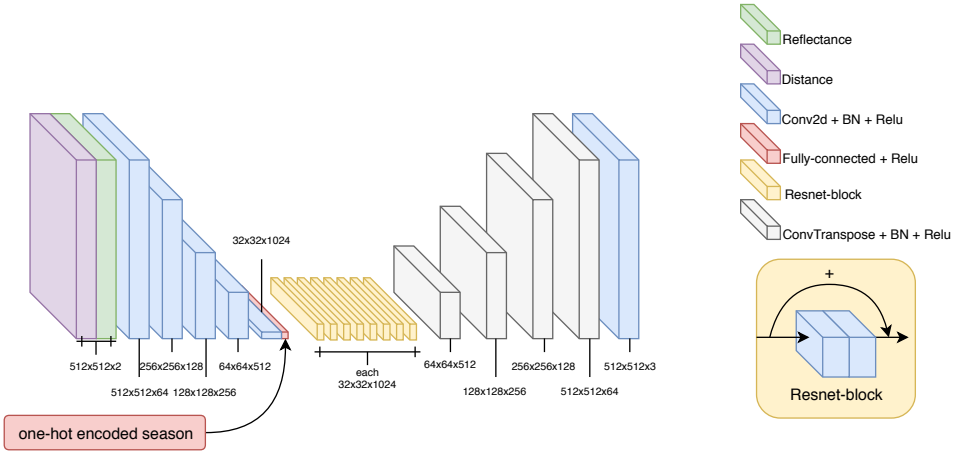

Figure 2 Our adapted generator network. Note how the capture date is injected by concatenating a fully connected layer.

point. Depending on the scanning situation, the points appear more or less sparse on the image plane.

3.2 C-GAN

Our approach is heavily inspired by the *pix2pixhd* network by Wang et al. [10]. The *pix2pixhd* network incorporates instance segmentation information and label-maps in order to enable object manipulation. By encoding the features of one instance, it is able to generate diverse images from the same input.

We modified this generator network as follows. We removed the instance- and labels-maps from the network architecture because we don't have any information about the class or instance of each point. We also reduced the number of generator networks to one $\{G\}$ and the number of multiscale discriminators to two $\{D_1, D_2\}$. However, the discriminator networks have exactly the same architecture as defined by Wang et al. [10]. Each discriminator operates on a different image scale, D_1 at the original scale 512×512, and D_2 at 256×256. We adopted the $\mathcal{L}_{GAN}$ part of the loss function as follows:

$$\min_{G} \max_{D_1,D_2} \sum_{k=1,2} \mathcal{L}_{GAN}(G, D_k) = \sum_{k=1,2} E_{(x,y)}[logD_k(x,y)] + E[log(1 - D_k(x, G(x,s)))]. \quad (1)$$

The training dataset is given as a set of tuples of corresponding images and dates $\{(x_i, s_i, y_i)\}$, where x_i is the input-/reflectance-image, y_i is the real image, taken by a camera of our mapping van, and s_i is the date the image was taken. In order to encode the capture date s_i, we added a fully connected layer as latent vector of the generator network. Similar to the findings of Zhu et al. [15], we also observed that noise fed additionally and directly into the generator was completely ignored by the network. As shown in figure 2, the fully-connected layer was instead concatenated to the bottleneck after convolving the input. The following layers of the generator network are identical to the *pix2pixhd* network. We used a one-hot encoding for each capture date s_i, as follows:

$$f(s_i) = \begin{cases} 1 + \mathcal{N}(\mu, \sigma^2), & \text{if } s_i = date \\ 0 + \mathcal{N}(\mu, \sigma^2), & \text{otherwise.} \end{cases} \quad (2)$$

In order to induce stochasticity, we added noise to each input, using $\mu = 0$ and $\sigma^2 = 0.1$.

4 Results

We trained the Networks for 20 epochs with a batch size of 1. Figure 3 shows an example of a predicted image. Remember that this is computed using only the reflectance and distance information from the point cloud. Note that the predicted building is colored in a typical color (white walls and red roof), while in reality, the building has quite different colors (red walls and dark roof). We believe that the color information is thus mostly derived by the spatial information and not by the (laser scanner) reflectance of the points themselves.

Figure 3 Input image (reflectance, left), synthesized image (middle) and real image (right).

Figure 4 shows that by shifting the value in the one-hot encoded season vector, we are able to predict different seasons for the same laser scanner input. In this case, we used a point cloud that was recorded in Germany in March and predicted an image for June and December. However, we think that some features will stay encoded in the picture itself. For example, the amount of leaves which are captured by the laser scanner directly. It is also worth to mention that there are a large number of occluded points in the left pane of figure 4. From the middle and right pane of figure 4, it can be seen that the generator has learned to hide occluded points.

Figure 4 Summer (middle) and winter (right) representation of the same input point cloud (left).

In Figure 5 we are showing additional examples of synthesized images from the same input but different season. We created also a video `https://youtu.be/mQINboXOvRM` which shows the difference between summer and winter.

Figure 5 Examples for different seasons with the same input per row.

5 Conclusion and outlook

In this work we have shown that it is possible to predict realistically looking images, using only point cloud data. By parameterizing the different capture dates of the images and point measurements, we were able to map the same point cloud to different seasons. We have shown that the GAN was able to encode seasonal information like snow in winter or green trees in summer. Furthermore, the generator was able to hide occluded points.

For future work we would like to test if this procedure is able to colorize point clouds without using cameras. Provided that our point cloud is labelled, it is imaginable that this framework allows us to project high precision labels to the generated images in order to create or enrich data sets for semantic segmentation. Since our approach allows to define arbitrary view positions and angles, this would enable us to generate an infinite amount of training examples.

Acknowledgement

This work was funded by the German Research Foundation (DFG) as a part of the Research Training Group GRK2159, ‘Integrity and collaboration in dynamic sensor networks’ (i.c.sens).

References

1 Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers & Graphics*, 2017.

2 Ayushman Dash, John Cristian Borges Gamboa, Sheraz Ahmed, Marcus Liwicki, and Muhammad Zeshan Afzal. Tac-gan-text conditioned auxiliary classifier generative adversarial network. *arXiv preprint arXiv:1703.06412*, 2017.

3 Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

4 Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015.

5 Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017.

6 Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.

7 Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

8 Scott E Reed, Zeynep Akata, Santosh Mohan, Samuel Tenka, Bernt Schiele, and Honglak Lee. Learning what and where to draw. In *Advances in Neural Information Processing Systems*, pages 217–225, 2016.

9 Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

10 Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *arXiv preprint arXiv:1711.11585*, 2017.

11 Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

12 Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *IEEE Int. Conf. Comput. Vision (ICCV)*, pages 5907–5915, 2017.

13 Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative visual manipulation on the natural image manifold. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

14 Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.

15 Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. *arXiv preprint arXiv:1711.11586*, 2017.