Abstract

We present the first approach for 3D point-cloud to image translation based on conditional Generative Adversarial Networks (cGAN). The model handles multi-modal information sources from different domains, i.e. raw point-sets and images. The generator is capable of processing three conditions, whereas the point-cloud is encoded as raw point-set and camera projection. An image background patch is used as constraint to bias environmental texturing. A global approximation function within the generator is directly applied on the point-cloud (Point-Net). Hence, the representative learning model incorporates global 3D characteristics directly at the latent feature space. Conditions are used to bias the background and the viewpoint of the generated image. This opens up new ways in augmenting or texturing 3D data to aim the generation of fully individual images. We successfully evaluated our method on the KITTI and SunRGBD dataset with an outstanding object detection inception score.

1. Introduction

Domain translation is a well known and widely applied problem. It is typically treated in computer graphics or computer vision. Most research focuses on image-to-image translation (Isola et al., 2016; Zhu et al., 2017; Wang et al., 2018). Examples are Semantic-Labels to Image (e.g. Labels to Street-Scene, Labels to Facades ) or Image conversions (e.g. Day to Night, Black-and-White to Color). Those techniques deal with real domain translation problems, since they convert semantic sensor-independent context into realistic RGB image data or vice versa. However, domain translation is performed on top of images. Both domains encode the information as RGB values in pictures with a spatial dependency. We call that single mode domain translation:

\[ G_{x\rightarrow y} : x \rightarrow y \quad x, y \in \mathbb{R}^{w \times h \times 3} \]

\[ G_{y\rightarrow x} : y \rightarrow x \]  

whereas, \( G_{x\rightarrow y}, G_{y\rightarrow x} \) describe the translation functions between both image domains \( x, y \) with fixed image sizes: \( h \) (height), \( w \) (width).

We propose a novel multi-modal domain translation model using the example of 3D point-cloud to image translation. The treated problem is formally known as:

\[ G_{p\rightarrow y} : p \rightarrow y \quad p \in \mathbb{R}^{n \times 3} \quad y \in \mathbb{R}^{w \times h \times 3} \]  

Here, \( n \) describes the number of points within the point-set. Our work is limited to \( G_{p\rightarrow y} \) (not \( G_{y\rightarrow p} \)). Therefore an extensive new architecture is presented as combination of a typical encoder-decoder for image segmentation (UNet: (Ronneberger et al., 2015)) as proposed by (Isola et al., 2016). More important is the models second input, where the architecture incorporates the real point-set to add 3D characteristics into the global feature space for constraint based individual image generation. We put conditions as viewpoint dependent projection and background image patches for fully individual image generation in compliance with 3D specifications (conditions: background, shape, distance, viewpoint).

2. Related Work

2.1. Image generation

2.1.1. Handcrafted Losses

As image generation could be reduced to per-pixel classification/regression with a wide application area it turns out to have a long tradition (Shelhamer et al., 2017; Xie & Tu, 2015; Yoo et al., 2016; Iizuka et al., 2016). Those applications suppose a conditionally unstructured loss applied on the output space, i.e. a pixel independence in terms of semantic relationship is supposed. The performance of those approaches strongly depends on the loss design, e.g. semantic segmentation (Paszke et al., 2016).
2.1.2. Conditional GANs

Conditional GANs (cGAN) instead learn structured losses that affect the overall output in form of a joint improvement (Isola et al., 2016). In common, the cGAN is applied in a conditional setting. For image generation researchers were setting variable conditions: e.g. discrete labels (Mirza & Osindero, 2014; Denton et al., 2015), text (Reed et al., 2016) and images (Isola et al., 2016; Zhu et al., 2017; Wang et al., 2018).

In general the cGAN performs a mapping function \( G \), called generator, based on a condition \( c \) and a random noise vector \( z \) to generate an image \( y \):

\[
G : \{c, z\} \rightarrow y \quad y \in \mathbb{R}^{w \times h \times 3}
\]

\[
c = \begin{cases} 
\in \mathbb{R} & \rightarrow \text{label-to-image} \\
\in \mathbb{R}^t & \rightarrow \text{text-to-image} \\
\in \mathbb{R}^{w \times h \times 3} & \rightarrow \text{image-to-image}
\end{cases}
\]  

(3)

For image-to-image translation (Isola et al., 2016) proposes a U-Net like structure for \( G \). To create realistic images at higher resolutions (e.g. 1024 x 2048) (Wang et al., 2018) recommend a pyramidal approach for \( G \) similar to a PSPNet (Zhao et al., 2016).

In general, the cGAN is composed by \( G \) and a competing discriminator \( D \), which distinguishes between real images and created fake ones. A well-established discriminator network is the Patchgan (Li, 2016) proposed by (Isola et al., 2016). Derived from that the competing objective of the cGAN could be described by its loss \( L_{cGAN} \):

\[
L_{cGAN}(G, D) = \mathbb{E}_{c,y} \{ \log(D(c, y)) \} + \mathbb{E}_{c,z} \{ \log(1 - D(c, G(c, z))) \}
\]  

(4)

2.2. Point-cloud processing

High requirements for perception tasks of robotic applications enforced the usage of 3D sensors, e.g. RGBD-cameras (Song & Xiao, 2015), Lidar (Valeo SCALA). Research progress in the field of 3D point-cloud processing received a boost in the recent years. In principle, point-clouds have specific properties that clearly distinguish them from images. Hence, specific processing models are needed. Points usually are not ordered, there is no grid that encodes the 3D position as an image does. The overall category of a point-set is influenced by the interaction of points among others. Only the global sum of the points forms a shape with a meaning. Last, point-sets are invariant to basic transformations like translation or rotation. Therefore the combination of 3D points clouds and machine learning is indispensable. The processing type could be categorized into the following three classes.

2.2.1. Real 3D point-cloud processing

(Qi et al., 2016) proposed the first neural network architecture Point-Net that handles natural points sets for classification and segmentation tasks with outperforming segmentation results on ShapeNet (Chang et al., 2015): mIoU 83.7. The model does not use convolutional layers, but fully connected ones and directly processes the coordinates of the point-set (size \( n \)): \( p = x_1, ..., x_n \) with \( p \in \mathbb{R}^{n \times 3} \). A chain of local transformations \( h \) on the point-set followed by a global max-pooling layer is used to create an overall feature space, i.e. a global approximation function:

\[
f(x_1, ..., x_n) \approx g(h(x_1), ..., h(x_n))
\]

\[
f : \mathbb{R}^{n \times 3} \rightarrow \mathbb{R}^K
\]

\[
h : [\mathbb{R}_1 \times ... \times \mathbb{R}_n] \rightarrow \mathbb{R}^K
\]

I.e. The overall meaning \( f \) (e.g. object class) of a point set \( p \) is approximated by \( g \). The advantage of the architecture is that it is robust against unordered point-clouds and transformations. The independence form the viewpoint variance helps to train with less training samples. Due to disadvantage for learning global features of large point sets the authors developed a second version Point-Net++ (Qi et al., 2017b).

2.2.2. Voxelization

Voxelization approaches make use of the findings performing CNNs on images. Therefore, 3D data is converted to voxels or grid cells. After pre-processing standard machine learning architectures are applied. Unordered point-sets are avoided. Famous applications are 3D object detectors like (Simon et al., 2018; Chen et al., 2016; Luo et al., 2018; Ku et al., 2018)

2.2.3. Combined models

Combined models have often shown most robust results (e.g. 3D object detection) and mostly make use of different sensor types. (Zhou & Tuzel, 2017) investigated a method based on many local Point-Nets followed by a global 3D CNN. (Qi et al., 2017a) architectures works the other way around. With the aid of a camera frustum points are filtered using a camera object detector. The filtered points are processed for 3D object detection with only one Point-Net (Qi et al., 2016) up to the last global max pooling layer ending in a \( 1 \times 1024 \) general feature space. A 8 bit depth projection using the given camera projection matrix \( (c_2) \)

2.2.4. Generative models

Point Cloud GAN by (Li et al., 2018) is a famous approach for point-cloud generation. They do not perform any translation task, but they show that the common discriminator is
Points2Pix: 3D Point-Cloud to Image Translation using conditional GANs

3. Points2Pix

We propose a novel cGAN architecture for generating photorealistic images from pure point-clouds. Additionally, we describe conditions to bias the viewpoint, distance, shape and the background within the latent space. Therefore, we introduce the network architecture consisting of a generator $G$ (converting points to images), a discriminator $D$ and the specific loss.

3.1. Generator

The objective of our generator $G(c_1, c_2, c_3)$ is to translate point-clouds into realistic-looking images, while using three conditions $c_1, c_2, c_3$. The whole architecture is shown in Figure 1. The design is inspired by (Isola et al., 2016), which serves as base.

3.1.1. Condition one

First, $c_1$ as raw point-cloud $c_1 = \{x_1, ..., x_n\}$ is processed by PointNet (Qi et al., 2016). The model samples $n = 1024$ points as input, applying an input transformation and aggregates global point features using fully connected layers and a generic max pooling (see equation 5):

$$f(c_1) \approx g(h(x_1), ..., h(x_n))$$

$$n = \{1...1024\}$$

(Achlioptas et al., 2017) performs label to point-cloud translation by using representative learning and introduce several 3D GAN derivatives. A similar study is published by (Wu et al., 2017) with focus on latent space analysis. However, learning 3D representations to generate viewpoint based images is missing within the research community. Therefore, we propose our novel technique Points2Pix.

Figure 1. Points2Pix Generator Architecture. The figure outlines the overall pipeline of the Points2Pix generator. In general we split the design into three areas: Top: the PointNet for a raw point-cloud processing; Bottom: Unet with skip connections for image generation; Middle: Global feature space concatenation from point-set (top) and image processing (bottom) pipeline. The model needs only a raw point-set as input, which acts as condition $c_1$. The point-sets coordinates will be directly processed by PointNet. A projection into the image plane is used for UNet as input, whereas as the camera projection matrix $C_c$ works as condition $c_2$. Additionally an arbitrary background patch $c_3$ is used for background generation.
However, in contrast to the basic PointNet pipeline, the proposed model incorporates the the global 3D features space \((n \times 1024)\) using concatenation at the innermost part within the Image encoder-decoder (UNet). Hence, \(h(x_1), ..., h(x_n)\) are applied by the PointNet part, \(g\) is implicitly performed with the aid of UNets decoder (see Fig. 1).

### 3.1.2. Condition Two

The second condition denoted as \(c_2 = C_c\) is an image projection of the point-cloud using a perspective projection matrix \(P\):

\[
\begin{pmatrix}
  0 & 0 & 0 & 0 \\
  0 & 0 & f_c & 0 \\
  0 & -f_c & f_c \cdot n_c & 0 \\
  0 & n_c & 0 & 0
\end{pmatrix}
\]

\[
s = \frac{1}{\tan \left(\frac{\text{fov}}{2} \cdot \frac{\pi}{180}\right)}
\]

\[
x_{\text{pixel}} = P \cdot T_{\text{cam}} \cdot x_i
\]

\[
y_i(y_{\text{pixel}}) = \frac{|x_{\text{cam}} - x_i|}{d_{\text{max}}}
\]

\[
y_{\text{b}}(y_{\text{pixel}}) = I(x_i)
\]

with a scaling \(s\) according to the horizontal field of view denoted as \(\text{fov}\) in degrees, near clipping plane denoted as \(n_c\) and far clipping plane \(f_c\). We encode radial depth \((y_g(x_{\text{pixel}}) \rightarrow \text{green channel})\) with a normalized depth \(d_{\text{max}}\) and intensities \(I(x_i)\) of the measured reflectance for each point falling into the projection image \((y_b(x_{\text{pixel}}) \rightarrow \text{blue channel})\). Before applying \(P\), all points are transformed into the camera coordinate system using the extrinsic calibration \(T_{\text{cam}}\). In this way we ensure the consistent viewpoint during training compared to the raw ground truth rgb image.

### 3.1.3. Condition Three

Finally, the third condition \(c_3\) is an arbitrary image background patch constraining environmental texturing. A surrounding image patch of the object cropped from the data set centered at the object origin up to a size of \(256 \times 256\) is extracted. During training the image background patch is compliant to the ground truth. In test-mode, background patches can be randomly mixed with point-clouds.

Both, \(c_2\) and \(c_3\) are combined to an \(256 \times 256\) input image, which is fed into a UNet with skip connections. At the innermost part, down sampled input features are concatenated with the global 3D feature space from \(c_1\). After up-sampling the output is a generated image with \(256 \times 256\) pixels. Since, we use a cGAN for training, there is no need for an unstructured Loss. The assessment of the output is performed by the discriminator. As a note, we do not use a random noise vector \(z\) (3). Noise is only incorporated as dropout similar to (Isola et al., 2016).

### 3.2. Discriminator

We use the Markovian discriminator PatchGAN (Isola et al., 2016) that tries to distinguish between fake \(D[G(c_1, c_2, c_3)]\) and real images \(D[y]\) at the scale of \(N \times N\) patches as well as possible. In contrast to (Isola et al., 2016) we do not take the condition \(c_1, c_2, c_3\) into account. The output depends only on the generated image. Therefore, it consists of an \(L1\) term to force low-frequency correctness (Zhu et al., 2017) and is applied convolutionally across the image, averaging all responses. We only use 5 convolutional layers with batch and instance normalization. In this way, it effectively solves the problem to be able to model high- and low frequency structures at once.

### 3.3. Loss

The objective of a basic GAN can be explained by an additive combination of the generative network \(L_G\) and the discriminative network \(L_D\) loss. In order to iteratively improve results during training \(L_G\) should be reduced while \(L_D\) grows ideally. Consequently the basic cGAN loss can be described as follows assuming the three input conditions \((c_1, c_2, c_3)\):

\[
L_{cGAN}(G, D) = E_y [\log(D(c_2, c_3, y))] + E_{c_1, c_2, c_3} [\log(1-D(c_2, c_3, G(c_1, c_2, c_3))]
\]

Random noise \(z\) (3) is only realized using dropout. Compared to the typical cGAN loss \(L_{cGAN}\) (4), the model does not involve all conditions into the discriminator. However, we implicitly force conditions to be compliant within the output by using a weighted \(L1\) term (Wang et al., 2018) in the overall loss. This part describes the \(L1\) difference between the output and the ground truth. The final loss can
be written as:

\[
L_{\text{Points2Pix}} = L_{c\text{GAN}}(G, D) + \lambda_{L1} \cdot \mathbb{E}_{c_1, c_2, c_3, y} \| y - G(c_1, c_2, c_3) \|_1
\]  

(10)

4. Experiments

We conduct experiments on KITTI (Geiger, 2012) for outdoor and SunRGBD (Song et al., 2015) for indoor scenarios to explore the general validity of the method. Additionally, we show that the approach works for both, Lidar generated point-clouds and point-clouds coming from by RGB-D sensors. Following the recommendations of (Isola et al., 2016), the quality of the synthesized images is evaluated using an object based inception score. Furthermore, classification and diversity scores are added as additional assessment. Finally, we present some insights into our architecture decisions with additional ablation experiments.

4.1. Metrics

To assess the realism of the produced images, YOLOv3 (Redmon & Farhadi, 2018) is used for validation. It is an off-the-shelf state of the art 2D object detector pre-trained on ImageNet and fine-tuned on the MS-Coco (Lin et al., 2014) data-set. This model includes overlapping classes in comparison to our experiments, e.g. cars (for KITTI) and chair (for SunRGBD). For the quantitative metrics we follow the instructions recommended by (Wang & Gupta, 2016).

4.1.1. Classification Score

With the aid of YOLOv3 the number of correct detected classes is measured. This could be achieved due to object centered image patches in our experiments. The classification score \( S_c \) ratio is then given by the detection ratio of fake images and ground truth \( TP \rightarrow \text{true positives} \). The score could be directly affected by adjusting the confidence rate of the 2D object detector: \( S_c = \frac{TP_{\text{fake}}}{TP_{\text{real}}} \).

4.1.2. Object Based Inception Score\(^1\)

For positive results in terms of classification we measure the intersection over union \( IoU_{\text{Points2Pix}} \) of the predicted bounding box \( BB \) coming out of YOLOv3 for the ground truth and the accompanied fake image.

\[
IoU_{\text{Points2Pix}} = \frac{BB_{\text{fake}} \cap BB_{\text{real}}}{BB_{\text{fake}} \cup BB_{\text{real}}} \quad | \quad S_c = 1
\]  

(11)

\(^1\)We call it inception score, because its similar to the proposal of (Salimans et al., 2016). We do not use an inception model.

4.1.3. Diversity Score

We measure the diversity ability of our cGAN to produce a wide spread of different output features using a diversity score. Our objective is to bias the shape, distance and 3D characteristics of the object. We collect different background image patches, while keeping the point-cloud constant \( (c_1 = \text{const} \; \text{and} \; c_2 = \text{const}, \; c_3 \rightarrow \{1...10\}) \). This leads to different output images that all should have the same 3D object inside. Therefore we compare the ground truth YOLOv3 results and all the fake images with the aid of calculating the mean \( S_c \) and the mean \( IoU \).

4.2. Training Details

We train the network on both data sets separately for 100 epochs from scratch each, using the ADAM optimizer (Kingma & Ba, 2014), with a learning rate of 0.0002 and momentum parameters \( \beta_1 = 0.5, \beta_2 = 0.999 \) such as \( \lambda_{L1} = 100 \). For our background condition \( c_3 \), we use image patches with a border width of 15 pixels. We found using objects containing at least 700 points in their point-cloud as a good trade-off for minimum point density as well as object size.

4.2.1. KITTI

In a pre-processing step, we split the 7481 training examples of the 3D object detection benchmark and use 3712 samples for training and 3769 for evaluation. Therefore, we generate more than 20k training images for the class car only using \( d_{\text{max}} = 60m \). Thus, each camera image is cropped centered at one labeled object with \( 256 \times 256 \) pixels. At the same time, strongly occluded or truncated objects are skipped.

4.2.2. SunRGBD

We extract 3267 images from the SunRGBD data-set containing the following classes: chair, table, desk, pillow, sofa and garbage bin. The split for training and validation is a 90/10 ratio. Image patches are extracted at the object center from the cameras point of view with a size of \( 256 \times 256 \) pixels with \( d_{\text{max}} = 4m \). The depth information comes from either MS kinect v1 or v2 and the Intel real-sense. Since, those sensors do not measure a reflectance, we only encode the radial depth inside the projection of \( c_2 \). Hence, the projection image contains one channel only.

4.3. Results

In Fig. 3 we show qualitative results for both data-sets and four different classes. Widely distributed output images are produced by alternating the background while keeping the point-cloud constant. An interesting point is, that our model learns 3D characteristics. This could be proven with different outputs (backgrounds) where the objects geometry
Points2Pix: 3D Point-Cloud to Image Translation using conditional GANs

Figure 3. Qualitative results of Points2Pix: The figure shows four different classes (cars (top) → 3 samples of KITTI; table, chair, pillow (bottom) → in each case one sample of SunRGBD). The results are taken from the test-set and never seen during training. The left column shows the ground truth image and the corresponding point-cloud in the second column. Fake images are generated based on a constant point-cloud $c_1, c_2 = \text{const}$ and ten alternating background patches $c_3$ (column 3-10). The model retains 3D characteristics of the objects. Stake constant. Note, even the objects color stays the same apart from slight differences in reflections and illuminations. This means, the model associates a color with a specific 3D shape represented within the 3D latent feature space. Hence, alternating backgrounds do not affect the objects representation (geometry, color).

Tables 1 and 2, as well as Fig. 4 show quantitative results based on our metrics described in 4.1. We achieve extreme positive results for KITTI ($S_c, \text{IoU}$) and sufficient values for SunRGBD. SunRGBD includes a higher number of occlusions which drastically affects the scores. Additionally, there are far less samples on each class compared to cars in KITTI. Qualitative results of the inception score are shown in Fig. 5.

4.3.1. ABLATION STUDY

Architectural Review For completeness, we test two derivative architectures of our full pipeline (Fig. 7). In this way, we successfully show a point-cloud to image translation only based on the point cloud itself (PointNet only). Doing this, the whole training procedure runs much faster.

Figure 4. Points2Pix classification score. The plot shows classification scores $S_c$ for KITTI and SUN-RGBD of our full Points2Pix architecture as well as two derivative architectures (see Fig. 7) over confidence thresholds used for object detection with YOLOv3. The full architecture outperforms for KITTI as well as for SunRGBD.
Figure 5. Qualitative object based inception results. The figure shows several generated cars and chairs (left) together with their accompanied real images (right). Green bounding boxes indicate detections on the real rgb image patches and red boxes visualize the corresponding ones on the fake images. The blue value obtains the IoU of both.

Table 1. The object based inception score $IoU_{Points2Pix}$ is calculated on the test set for both data-sets. We show results for varying confidence thresholds, i.e. 0.3, 0.5, 0.7 for KITTI and 0.1, 0.2, 0.3 for SunRGBD. Minus indicates no detections.

| dataset  | class | 0.3 | 0.5 | 0.7 |
|----------|-------|-----|-----|-----|
| KITTI    | car   | 0.76| 0.77| 0.77|
|          |       | 0.1 | 0.2 | 0.3 |
| SunRGBD  | sofa  | 0.52| 0.77| 0.77|
|          | table | 0.70| -   | -   |
|          | chair | 0.60| 0.58| 0.58|

Table 2. Diversity score is calculated on the test-set for both data-sets. Each sample is recomputed ten times with a random image background patches. A minus indicates no detections for the associated class.

| dataset  | class | 0.3 | 0.5 | 0.7 |
|----------|-------|-----|-----|-----|
| KITTI    | car   | 0.71| 0.70| 0.68|
|          |       | 0.1 | 0.2 | 0.3 |
| SunRGBD  | sofa  | 0.16| -   | -   |
|          | table | 0.24| 0.22| -   |
|          | chair | 0.45| 0.37| 0.33|
Learning 3D representations: The full Points2Pix architecture learns 3D representations. The model offers a high flexibility in generation of different viewpoints by adjusting condition $c_2$. The upper part shows two examples of KITTI when rotating the point-cloud slightly by 20 degrees around the y-axis. The bottom (SunRGBD) shows the results when flipping the projection by 180 degrees around the x-axis.

Figure 7. Architectural review: Two derivatives from the basic Points2Pix $G(c_1, c_2, c_3)$ (left) generator are tested regarding their classification score (see Fig. 4). One on hand a $Unet$ only version $G(c_2, c_3)$ (middle), on the other hand a $PointNet$ only $G(c_1)$ (right) version is tested. The full model outperforms the others regarding the classification score.

Due to far less parameters to optimize. Nevertheless, sometimes a repeating noise with a high contrast similar to Moire effects appears, which indicates instabilities and uncertainties. Generated objects are in compliance with their 3D specifications, but in order to enlarge variance of the outputs and to control background conditions $c_2$ and $c_3$ are required. We found that the first part of the $Unet$ and the viewpoint dependent projection $c_2$ especially help to reduce the mentioned noise effects. They provide additional information in 2D space and stabilize the network. As a fallback we additionally test a $Unet$ only version (Fig. 7). However, our full pipeline significantly outperforms the derivative architectures in terms of classification (Fig. 4).

Rotations To further emphasize the influence of $c_2$ and to show our models ability to constrain object viewpoints, we rotate all input points $x_i$ for $c_2$. We test that for KITTI with a rotation of 20 degrees around the y-axis and for SunRGBD with a rotation of 180 degrees around the x-axis (see Fig. 6). Note, that our point-cloud condition $c_1$ stays unmodified, because $PointNet$ approximates a symmetric function to be invariant of rotations. The test shows that rotations can be implicitly learned. This offers many opportunities in generating 3D data.

5. Conclusion

In this work, we propose a novel approach for 3D point-cloud to image translation based on conditional GANs. Our network handles multi-modal sources from different domains and is capable of the translating unordered point-clouds to regular image grids. We use three conditions to generate a high diversity, while being flexible and keeping 3D characteristics. We prove that the model learns 3D characteristics, what even makes it possible to sample images from different viewpoints. Those networks are applicable in a wide variety of applications, especially 3D texturing.
References

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. J. Representation learning and adversarial generation of 3d point clouds. CoRR, abs/1707.02392, 2017. URL http://arxiv.org/abs/1707.02392.

Chang, A. X., Funkhouser, T. A., Guibas, L. J., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., and Yu, F. ShapeNet: An information-rich 3d model repository. CoRR, abs/1512.03012, 2015. URL http://arxiv.org/abs/1512.03012.

Chen, X., Ma, H., Wan, J., Li, B., and Xia, T. Multi-view 3d object detection network for autonomous driving. CoRR, abs/1611.07759, 2016. URL http://arxiv.org/abs/1611.07759.

Denton, E. L., Chintala, S., Szlam, A., and Fergus, R. Deep generative image models using a laplacian pyramid of adversarial networks. CoRR, abs/1506.05751, 2015. URL http://arxiv.org/abs/1506.05751.

Geiger, A. Are we ready for autonomous driving? the kiti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR ’12, pp. 3354–3361, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-1-4673-1226-4. URL http://dl.acm.org/citation.cfm?id=2354409.2354978.

Iizuka, S., Simo-Serra, E., and Ishikawa, H. Let there be Color!: Joint End-to-end Learning of Global and Local Image Priors for Automatic Image Colorization. ACM Transactions on Graphics (Proc. of SIGGRAPH 2016), 35(4):110:1–110:11, 2016.

Isola, P., Zhu, J., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. CoRR, abs/1611.07004, 2016. URL http://arxiv.org/abs/1611.07004.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

Ku, J., Moziian, M., Lee, J., Harakeh, A., and Waslander, S. Joint 3d proposal generation and object detection from view aggregation. IROS, 2018.

Li, B. 3d fully convolutional network for vehicle detection in point cloud. CoRR, abs/1611.08069, 2016. URL http://arxiv.org/abs/1611.08069.

Li, C., Zaheer, M., Zhang, Y., Póczos, B., and Salakhutdinov, R. Point cloud GAN. CoRR, abs/1810.05795, 2018. URL http://arxiv.org/abs/1810.05795.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. CoRR, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Luo, W., Yang, B., and Urtasun, R. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

Mirza, M. and Osindero, S. Conditional generative adversarial nets. CoRR, abs/1411.1784, 2014. URL http://arxiv.org/abs/1411.1784.

Paszke, A., Chaurasia, A., Kim, S., and Culurciello, E. ENet: A deep neural network architecture for real-time semantic segmentation. CoRR, abs/1606.02147, 2016. URL http://arxiv.org/abs/1606.02147.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. arXiv preprint arXiv:1612.00593, 2016.

Qi, C. R., Liu, W., Wu, C., Su, H., and Guibas, L. J. Frustum pointnets for 3d object detection from RGB-D data. CoRR, abs/1711.08488, 2017a. URL http://arxiv.org/abs/1711.08488.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. CoRR, abs/1706.02413, 2017b. URL http://arxiv.org/abs/1706.02413.

Redmon, J. and Farhadi, A. Yolov3: An incremental improvement. CoRR, abs/1804.02767, 2018. URL http://arxiv.org/abs/1804.02767.

Reed, S. E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative adversarial text to image synthesis. CoRR, abs/1605.05396, 2016. URL http://arxiv.org/abs/1605.05396.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. CoRR, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. CoRR, abs/1606.03498, 2016. URL http://arxiv.org/abs/1606.03498.

Shelhamer, E., Long, J., and Darrell, T. Fully convolutional networks for semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell., 39(4):640–651, 2017. doi: 10.1109/TPAMI.2016.2572683. URL https://doi.org/10.1109/TPAMI.2016.2572683.
Simon, M., Milz, S., Amende, K., and Gross, H. Complex-yolo: Real-time 3d object detection on point clouds. CoRR, abs/1803.06199, 2018. URL http://arxiv.org/abs/1803.06199.

Song, S. and Xiao, J. Deep sliding shapes for amodal 3d object detection in RGB-D images. CoRR, abs/1511.02300, 2015. URL http://arxiv.org/abs/1511.02300.

Song, S., Lichtenberg, S. P., and Xiao, J. SUN RGB-D: A RGB-D scene understanding benchmark suite. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 07-12-June-2015:567–576, 2015. ISSN 10636919. doi: 10.1109/CVPR.2015.7298655.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional gans. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

Wang, X. and Gupta, A. Generative image modeling using style and structure adversarial networks. CoRR, abs/1603.05631, 2016. URL http://arxiv.org/abs/1603.05631.

Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, W. T., and Tenenbaum, J. B. MarrNet: 3D Shape Reconstruction via 2.5D Sketches. In Advances In Neural Information Processing Systems, 2017.

Xie, S. and Tu, Z. Holistically-nested edge detection. CoRR, abs/1504.06375, 2015. URL http://arxiv.org/abs/1504.06375.

Yoo, D., Kim, N., Park, S., Paek, A. S., and Kweon, I. Pixel-level domain transfer. CoRR, abs/1603.07442, 2016. URL http://arxiv.org/abs/1603.07442.

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. CoRR, abs/1612.01105, 2016. URL http://arxiv.org/abs/1612.01105.

Zhou, Y. and Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. CoRR, abs/1711.06396, 2017. URL http://arxiv.org/abs/1711.06396.

Zhu, J., Park, T., Isola, P., and Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. CoRR, abs/1703.10593, 2017. URL http://arxiv.org/abs/1703.10593.