# Point2Depth: a GAN-based Contrastive Learning Approach for mmWave Point Clouds to Depth Images Transformation

Walter Brescia, Giuseppe Roberto, Vito Andrea Racanelli, Saverio Mascolo, Luca De Cicco

***Abstract*— The perception of the environment is essential in mobile robotics applications as it enables the proper planning and execution of efficient navigation strategies. Optical sensors offer many advantages, ranging from precision to understandability, but they can be significantly impacted by lighting conditions and the composition of the surroundings. In contrast, millimeter wave (mmWave) radar sensors are not influenced by such adverse condition and are capable of detecting partially or fully obstructed obstacles, resulting in more informative point clouds. However, such point clouds are often sparse and noisy. This work presents Point2Depth, a cross-modal contrastive learning approach based on Conditional Generative Adversarial Networks (cGANs) to transform sparse point clouds from mmWave sensors into depth images, preserving the distance information while producing a more comprehensible representation. An extensive data collection phase was conducted to create a rich multimodal dataset with each information associated with a timestamp and a pose. The experimental results demonstrate that the approach is able to produce accurate depth images, even in challenging environmental conditions.**

***Index Terms*— mmWave sensors, point clouds, cGAN, contrastive learning, mobile robotics, contrastive generative adversarial networks, depth images.**

## I. Introduction

Sensors working in the ultraviolet, visible, and near infrared light are extensively used in mobile robots for the perception of the surroundings. These sensors offer many advantages, including high fidelity, precision, interpretability, and ease-of-use. The most commonly used sensors in this category are RGB and RGB-Depth (RGB-D) cameras and LiDARs, which enable obstacle detection, obstacle avoidance and Simultaneous Mapping and Localization (SLAM). These types of sensors are widely used in the literature and there exist many approaches which leverage their data to effectively tackle various mobile robotics tasks. However, visible light based sensor, can fail to provide reliable data in adverse environmental conditions, such as in the case of the presence of smoke, fog, and occlusions. On the other end, mmWave radar sensors are not affected by such conditions and can provide information of partially or fully obstructed obstacles and of objects made of materials that hardly reflect visible light (f.i., glass), which are often invisible to LiDARs. While mmWave sensors overcome and are more affordable, they are unfortunately more susceptible to noise and produce sparser Point Clouds (PCs) than LiDARs'. In particular, single-chip radars, even at a millimeter wave lengths, have a much lower azimuth resolution compared to LiDARs, resulting in point clouds with lower resolution. This limits their use to basic collision avoidance applications, while more advanced applications might require larger and more expensive mechanical radars that might not be suitable to most mobile robotics applications.

To address the limitations of PCs generated by mmWave radars, several approaches have been proposed in the literature for data processing and interpretation. Some authors have focused on denoising and interpolation methods, such as Kalman filtering and Gaussian Process Implicit Surfaces, to improve the accuracy of the point clouds [1]. Others have proposed feature extraction and classification algorithms to identify objects from the point clouds ([2], [3]).

Recently, there has been growing interest in using deep learning techniques, such as Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs), to improve the interpretation of mmWave data. In particular, cGANs have been applied for various tasks, such as object reconstruction and semantic segmentation [4], [5].

In this work, we propose a cross-modal contrastive learning [6], [3] approach based on cGANs [7] to transform mmWave point clouds into depth images. To this end, an extensive data acquisition has been conducted to create a multi-modal dataset in which each point cloud is strictly related to a depth image. The mmWave PC is rototranslated into the camera reference frame. Furthermore, for each PC, we only retain points in the field of view of the camera, since points outside the field of view cannot be compared to any information in depth image. The goal is to leverage the reliability of the mmWave sensor while mitigating the sparseness of its data.

## II. Related work

### A. Contrastive Learning

Contrastive learning [6] is a method in which a model is trained to identify patterns in the input data , while repelling dissimilar ones. This framework improves performance in both supervised and unsupervised contexts, and has also been used as a "pre-training" technique [8]. Previous studies (such as [9]) have demonstrated that several types of losses can be employed to extract useful information. In [10], contrastive learning was effectively used to extract meaningful representations from high-dimensional data (i.e., images) that were then fed to a Reinforcement Learning (RL) agent. The approach resulted in state-of-art performance in many visual tasks from Deep Mind control suite and Atari games.

All Authors are with Politecnico di Bari. E-mails: walter.brescia@poliba.it, g.roberto1@studenti.poliba.it, vitoandrea.racanelli@poliba.it, mascolo@poliba.it, luca.decicco@poliba.it

In this work, we aim to leverage contrastive learning as a training technique to derive a cross-modal representations from mmWave PCs to depth images. This process enforces a comparable latent space between the two types of data.

### B. Conditional Generative Adversarial Networks

GAN ([11]) were first introduced as a training framework for generative models. The GAN framework consists of two models: a *Generator* $\mathscr{G}$ and a *Discriminator* $\mathscr{D}$, in competition with each other. The generator's goal is to produce a faithful duplicate of a target signal from a noise signal, while discriminator's goal is to differentiate between signals produced by the generator and true signals from the given dataset. This creates a dynamic in which the generator tries to deceive the discriminator into believing its signal is a true signal, while the discriminator corrects the generator by accurately distinguishing between true and falsified signals. However, the designer has no control over the generation of fake signals in this setting.

In [7], the authors expanded on this work to introduce control over the way the generator outputs signals, resulting in the development of cGANs. In [12] a cGAN with an autoencoder-like generator is used to learn a mapping from an input image to an output image.

In this work, we utilize a cGAN model incorporating an autoencoder-like ([13]) generator to establish a mapping between an input color image to an output depth one. This results in the synthesis of an informative latent representation.

### C. Point Cloud elaboration

Point Cloud (PC) elaboration is an open issue in literature. Unlike other types of data (f.i. images), PCs are unordered data, meaning that a particular value has the same informative value independently of its position in the data structure. This property makes it difficult to apply traditional approaches that rely on position information, such as f.i., convolutional and pooling layers. In [2], authors propose a methodology to extract both local and global features through the use of a *transformation network* (T-Net). The proposed technique is demonstrated effective for the classification and segmentation of large PCs. [3] proposes a cross-modal contrastive learning approach to learn a LiDAR-like latent representation which is used for semantic labeling in an occupancy grid and as input for an RL agent for autonomous navigation.

In this work, we draw inspiration from [12] and apply a deep learning approach to derive a latent representation of *sparse* PC to effectively convert the input PC into a depth image.

### D. RGB to Depth

In [14], the authors utilize a sparse depth map and the associated RGB image to solve the depth scale ambiguity. The results show high accuracy compared to the original depth image. [15] presents a model architecture that advances the state-of-art performance, making use of an encoder that extracts multi-scale features, which are given in input to the decoder, making use of skip-connections.

In this work, we employ a deep neural network to generate a latent representation of the input image, which encodes all the necessary information about the distance. It is important to note that, as only the encoder part of the autoencoder will be employed in the final model, skip-connections cannot be utilized.

## III. mmWave Point Clouds to Depth Images

In this section we describe the methodology adopted to transform sparse mmWave PCs into depth images.

First, we observe that PCs and depth images share, to some extents, the same kind of information, which is the distance between an object and the sensor. However, if on one hand the depth image is usually produced by leveraging two RGB cameras and the related physical information, e.g. displacement between sensors, focal distance, field of view and so on, on the other hand the mmWave relies on its physical principles to produce PCs.

We also note that, when both sensors share the same point of view, mmWave PCs can be considered as a sparser representation of the depth image produced by an RGB-D camera. However, mmWave are not impacted by adverse light and environmental conditions, making them a more reliable source of depth information. Further, due to the different principles on which the two sensors are based, the resulting data cannot be considered as one the down-sampling of the other: two different scenes will produce two distinct results for both sensors.

We present *Point2Depth*, a model designed to tackle the sparsity of mmWave PCs, consisting of two components as shown in Figure 1a: (1) a PC Encoder Neural Network (NN), *Point2Latent*, that takes mmWave PCs in input and generates a *latent* representation; (2) *Latent2Depth*, a decoder NN which converts the *latent* representation into a depth image.

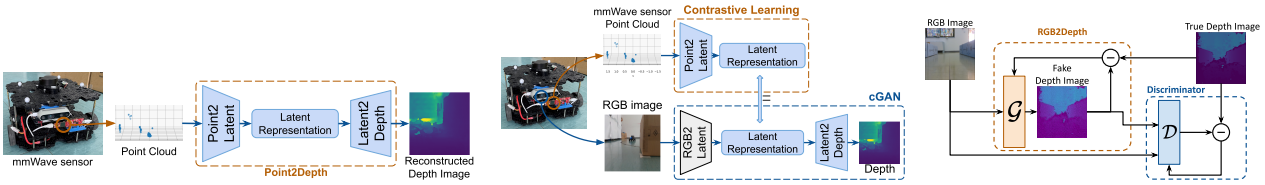
As shown in Figure 1c, this decoder is part of a cGAN, namely *RGB2Depth*, that is trained to generate faithful depth images from RGB images while encoding the initial information into a latent space. To ensure the produced latent representation is useful, the Point2Latent Encoder is trained using the contrastive learning paradigm. The overall training scheme is summarized in Figure 1b.

Point2Latent is trained to produce a latent space that is as close as possible to the one produced by RGB2Depth, when observing the mmWave PC. This allows Point2Latent to translate the input PC into a depth image-like latent representation. The Latent2Depth then decodes this representation into a depth image, in a decoupled process.

In the following, an in-depth description of each involved component is provided: Section III-A provides a detailed description of the approach used to train the depth image decoder; Section III-B explains the training of the PC encoder to generate a useful latent representation.

### A. RGB to Depth Image cGAN

We now describe the training scheme employed for the component *RGB2Depth* used to convert RGB images to a




(a) Point2Depth (b) Point2Depth (c) RGB2Depth training scheme

depth image. Figure 1c shows that this component is trained as a cGAN with a generator structure that can be viewed as an auto-encoder. RGB2Depth takes as input a color image rather than the corresponding depth image, which could lead to an encoded latent space optimized for image reconstruction instead of conveying effective distance information. This approach, while seemingly counter-intuitive, has the merit of encouraging the neural network to extract meaningful features from the input data, resulting in a more informative latent representation. Additionally, this approach reduces the risk of overfitting to the training dataset.

Let $\mathscr{G}$ and $\mathscr{D}$ identify the *generator* and the *discriminator*, respectively. The training dataset is defined by the couples $X^i = \langle X^i_{RGB}, X^i_D \rangle, i = \{0, ..., |X|\}$ with $|X|$ being the dataset's size. Note that $X^i_{RGB}$ identifies the i-th color image, $X^i_D$ represents the depth image associated to the i-th color one.

***Discriminator*** $\mathscr{D}$*:* In our approach, $\mathscr{D}$ is trained to observe both an RGB image and its associated depth one, and to determine whether the input depth image is a genuine or fake one. During the training process, the following loss function is optimized over a batch $b$ of data:

$$\mathscr{L}_{\mathscr{D}} = \frac{1}{b}\sum_{i=1}^{b}\Big[\log \mathscr{D}\left(X^i_{RGB}, X^i_D\right) + \log\left(1 - \mathscr{D}\left(X^i_{RGB}, \mathscr{G}\left(X^i_{RGB}\right)\right)\right)\Big] \tag{1}$$

The loss function (1) encourages the discriminator to correctly distinguish between true depth images from the dataset and fake depth images produced by the generator.

***Generator****:* $\mathscr{G}$ will observe a colour image and will produce a related depth image. During training, it will minimize the following loss function:

$$\mathscr{L}_{\mathscr{G}} = \frac{1}{b}\sum_{i=1}^{b}\Big[\log\left(1 - \mathscr{D}\left(X^i_{RGB}, \mathscr{G}\left(X^i_{RGB}\right)\right)\right) + \lambda_{L1} \cdot L1(\mathscr{G}\left(X^i_{RGB}\right), X^i_D)\Big] \tag{2}$$

with $\lambda_{L1}$ being a custom weight and $L1(X_{D_fake}, X^i_D)$ being the L1 norm between the real depth image and the one generated. Note that the term $\log\left(1 - \mathscr{D}\left(X^i_{RGB}, \mathscr{G}\left(X^i_{RGB}\right)\right)\right)$ rewards the generator for fooling the discriminator, while the second term $\lambda_{L1} \cdot L1(y_{fake}, y)$ encourages the generator to produce depth images as faithful as possible to the original ones. Also note that the generator will never observe the actual depth images and, therefore, the latter component will push the generator towards the recognition of patterns and identification of distance values directly from the colour image, ideally improving the latent features.

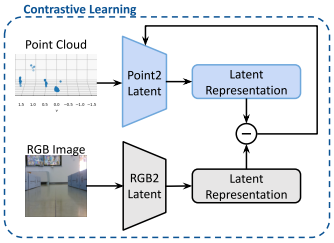



Fig. 1: Point2Latent training process

We denote with $\mathscr{N}$ the encoder and with $\mathscr{L}$ its output, i.e., the latent representation. We design the generator as follows:

*RGB2Latent*: an encoder composed of "down-sampling" blocks, convolutional layers which reduce the input data to a lower size. Each convolutional layer is followed by normalization, max-pooling and dropout layers and it will produce a latent representation, whose size is another hyperparameter.

*Latent2Depth*: a decoder composed of "up-sampling" blocks. Each block can be composed of a nearest neighbor up-sampling layer, followed by dropout, convolution and normalization layers. We have also experimented with transpose convolution to upscale from the latent space, but we found it to have worse performance in each tested configuration. Therefore, for brevity, we have neglected it in the following.

Note that such a configuration takes inspiration from [12]. The main differences from state-of-art algorithms are in the objective for which such network is trained for (henceforth the type of conversion), which leads to the impossibility of leveraging skip-connections: these type of connections require, at least to some extent, consistency between the input and the output data, which, in this case, is not met.

### B. Point2Latent

The goal of this network is to learn a representation of the input PC in the same latent space as the RGB2Latent encoder. The Point2Latent training process is depicted in Figure 1. The Point2Latent encoder $\mathscr{O}$ structure design is inspired by the work of [2]. It comprises two T-Nets: the first transforms the input PC into an order-independent representation, while the second extracts useful local features. The T-Nets are interleaved with several convolutional and max-pooling layers. The model's output is produced by a fully connected layer that shares the same output shape of $\mathscr{N}$. An example of this encoder structure is depicted in Figure 2.

$\mathscr{O}$ is trained following the contrastive learning paradigm. Let $X^i_{pc}$ be the PC associated to the i-th colour (and depth) image. The network will optimize the following loss function:

$$L1(\mathscr{N}\left(X^i_{RGB}\right), \mathscr{O}\left(X^i_{pc}\right)) + ||I - AA^T||^2 \tag{3}$$

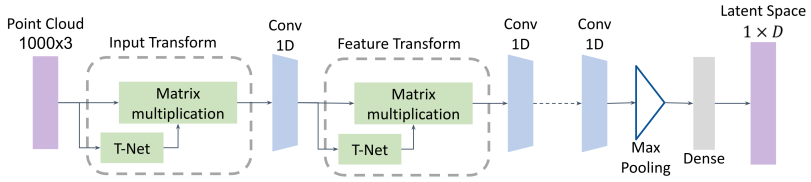



Fig. 2: Point2Latent Structure

with $I$ being the identity matrix and $A$ the T-Net transformation matrix. The term $L1(\mathscr{N}\left(X^i_{RGB}\right), \mathscr{O}\left(X^i_{pc}\right))$ computes the distance between the two representations, while $||I - AA^T||^2$ is a regularization term that leads the T-Net towards an orthogonal transformation. Note that, in order to train this encoder, we first train the RGB2Depth model and then fix its RGB2Latent weights.

## IV. Results

This section presents the results of our proposed approaches, as detailed in Section III.

### A. Dataset

In order to conduct our experiments and train the networks, we utilized a *Turtlebot3 Waffle*, a differential drive robot that comes equipped with multiple exteroceptive sensors. Specifically, the robot has a RP-LiDAR , a depth camera producing $640 \times 480$ frames for both color and depth images, and a mmWave sensor by Texas Instruments. Due to sensor's nature, depth images can present flaws or missing information. The applied pre-elaboration aims at removing missing regions and regularizing data into a $[-1, +1]$ range. Also for point clouds a pre-elaboration is required: we crop points into th e camera's field of view; the number of points is fixed to 1000 per frame and the points are normalized, also in this case.

In order to collect relevant and realistic data, the robot was tasked with navigating a $6 \times 6$m area filled with obstacles, capturing poses of both the robot and the obstacles via the Vicon Tracker[1]. A variety of obstacles were utilized, including *standard* ($0.49 \times 0.36 \times 0.25$m), *medium* ($0.36 \times 1.47 \times 0.25$m), *long* ($1.96 \times 0.36 \times 0.25$m), and *high* ($0.23 \times 0.95 \times 0.45$m) obstacles.

### B. Performance Analysis

In order to study and validate the performance of the proposed approach, we considered several network configurations. Note that the performance of the GAN is not influenced by the Point2Latent encoder, but the latent representation identified by the auto-encoder strongly impacts the Point2Latent training results. In fact, the RGB2Depth auto-encoder may learn a representation that is not suitable for the Point2Latent encoder. Therefore, when a good RGB2Depth configuration is found, several tests are conducted on the Point2Latent parameters. Many preliminary tests are conducted to reduce the number of hyperparameters left to identify. In the remaining, we will use "Adam" optimizer with 0.0001 as learning rate, a $\lambda_{L1}$ equal to 100 and each model is trained for 200 epochs. In the following section, we introduce the results of the tests conducted on the cGAN model and evaluate the Point2Latent training performance.

***RGB2Depth:*** The aim of this model is to identify a significant latent representation which will be later shared with the Point2Latent model. For the sake of brevity, only the most significant tests are reported, although many other hyperparameters have been tuned and altered, and several configurations resulted in ineffective models.[2]

The best-performing models are achieved with a latent space with 512 features, using both 8 and 4 down-sampling blocks and 8 up-sampling blocks , as shown in Figure 3a. Although their learning curves almost overlap, we observed that the two networks produce distinct high-level performance. A lower number of down-sampling blocks results in an ineffective extraction of useful information, leading to a decoded depth image with several artifacts and inaccuracies. For these reasons, we consider the model with 8 down-sampling blocks as the backbone model for the RGB2Depth in the following tests.

***Point2Latent:*** As previously mentioned, this network is inspired by [2], particularly its classification branch, but with a modified last layer to suit the purpose of this work. Based on the results of training the RGB2Depth model, we fix the latent representation to 512 features. In order to deterimine the optimal configuration, we repeat the training while changing the activation function (linear: l, ReLU: R, Tanh: T, Sigmoid: S) and applying (A) or not (N) normalization to the latent layer. These two letters are postponed to the notation used for experiments. The results are presented in Figure 3c.

The best performing configurations employ all four types of activation functions on the latent layer: linear, sigmoid, tanh, and ReLU, with the first two models using normalization, namely lA-A, SA-A, TN-N, and RN-N. In order to evaluate the generalization capabilities, we test these models on a separate test set and compute the mean and standard deviation (Std.D.) of two error metrics: L1 norm and Mean Squared Error (MSE). As shown in Table I, all models perform similarly, including those that performed poorly in the training phases. For this reason, it is necessary to conduct a practical study of the model's performance.

[1] https://www.vicon.com/software/tracker

[2] To simplify the notation, down and up -sampling blocks are named with the initials of each type of layer of which they are composed. A prefix number will specify the number of such blocks, while an afterword number will specify the size of the latent space (e.g. an auto-encoder with 8 down-blocks and 8 up-blocks will be referred to as "8CND-512-8UCND", where "512" identifies the latent size.

| Model | L1 | L1 Std.D. | MSE | MSE Std.D. |
|---|---|---|---|---|
| lN-N | 639 | 374 | 1114061 | 968965 |
| lA-A | 635 | 370 | 1108748 | 938931 |
| RN-N | 660 | 405 | 1191768 | 1039095 |
| TN-N | 658 | 380 | 1140771 | 935083 |
| SA-A | 650 | 377 | 1161711 | 955506 |

TABLE I: Point2Depth test errors

During preliminary tests, we observed that several models, even those with good performance, produced depth images with inaccuracies and artifacts that were not captured by the considered metrics. Further investigation revealed that the best performing model, which produced accurate depth images without artifacts, is the RN-N model. An analysis of results shown in Figure 4 reveals two main findings: (1) model RN-N is capable of representing the environment with a good accuracy, with the obstacle on the left and an open area on the right, while the remaining models fail to detect the obstacle altogether; (2) even in only slightly adverse environmental conditions, the real depth image can be very misleading, emphasizing the importance of using a sensor that is robust to such conditions for optimal perception of the environment and safe navigation. In order to further study the generalization capabilities and effectiveness of the identified best performing models, additional tests were conducted and are discussed in the following.

### C. Experiments

In this section, we investigate the performance and limitations of the best-performing model by considering two insightful scenarios. These tests aim to provide practical insights into the model's performance. For each test, we provide three images: the RGB image captured by the camera, the related depth image, and the point cloud captured by the mmWave sensor. Additionally, two figures report the predictions of both the RGB2Depth and Point2Depth networks of the RN-N model for each scenario.

***Test 1:*** in this test, the robot is facing three different obstacles. The first obstacle (right of second red line) is a known obstacle observed during training. The second obstacle (left of the first red lines) is a row of desks covered with white thin paper, which is particularly challenging since it is not easily detectable by the mmWave sensor. The third obstacle (between the two red lines) is an unseen object, another robot, much more complex than objects seen during training (see Figure 5). Figure 6 shows the RGB2Depth and Point2Depth predictions. The RGB2Depth prediction appears similar to the original depth image shown in Figure 5b, yet a key piece is missing: the unseen object cannot be associated with any of the regions between the two red lines. On the other hand, the Point2Depth prediction introduces distance values associated with the distance from the object, even if it resembles a box (an object seen during training). This is an interesting finding that we ascribe to two key reasons: (1) the RGB2Depth is actually trained to find the most effective latent representation, rather than a faithful depth image; (2) the mmWave PC contains information of such object and the Point2Latent model is able of conveying such information effectively into the latent space, allowing the Latent2Depth model to represent such information into the decoded image. However, the Latent2Depth decoder is not able of properly representing such object, as it has never seen it during training, replacing it with a familiar obstacle. Even if not optimal, we consider this result still useful, especially for navigation purposes.

***Test 2:*** the robot is facing directly the row of desks covered with paper. Figure 8 shows both the RGB2Depth prediction and the Point2Depth one. The RGB2Depth prediction is faithful to the original depth image shown in Figure 7b. On the other hand, the Point2Depth prediction is far from the original image, appearing as an image with an obstacle on the left and free space on the right. This result is expected as the model is observing an uninformative PC as shown in Figure 7c). In fact, the white paper is not detected and the points in the cloud are coming from the desk behind the paper. Therefore, while the prediction is far from the original depth image, it is actually producing a faithful image with respect to the input PC. Of course, the result is still not acceptable, but the goal of such a model is not to make up information that the mmWave sensor is not producing.

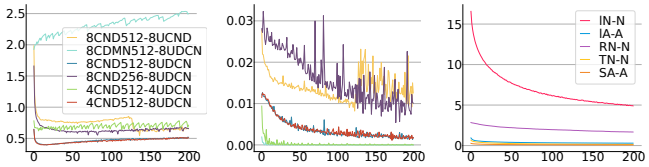
## V. Conclusions

Perception of the surroundings is a crucial task for mobile robotics, enabling robots to navigate their environment safely while fulfilling high-level tasks and avoiding obstacles. LiDARs, RGB and RGB-D cameras are popular sensors for this purpose due to their precision and ease-of-use. However, working in the visible light, they can be impaired to the point of being non-usable when utilized in harsh environments. In contrast, mmWave sensors are less affected by environmental factors due to their use of physical principles. As a result, they are gaining increased attention from both industry and academia. However, mmWave sensors produce sparse Point Clouds (PCs).

In this work, we presented a cross-modal contrastive learning approach based on cGANs to translate mmWave PCs into depth images. We built a multimodal dataset containing strongly correlated color images, depth images, and mmWave PCs. We conducted several tests to determine the best topology and hyperparameter configuration for each neural network involved in the approach. Further tests were conducted to evaluate the effectiveness of the best performing model. Our results demonstrate that the final proposed model can effectively produce information similar to depth images and can generalize to unseen objects, even when actual depth images cannot provide the necessary information. We have also shown that this approach can overcome impairments typically associated with sensors operating in the visible spectrum. As a further investigation, an in-depth analysis of the performance of the proposed model in dynamic and more complex environments will be evaluated.

## Acknowledgements

This work has been partially supported through the AGREED project (ARS01_0025) funded by the Italian Ministry of University and Research.

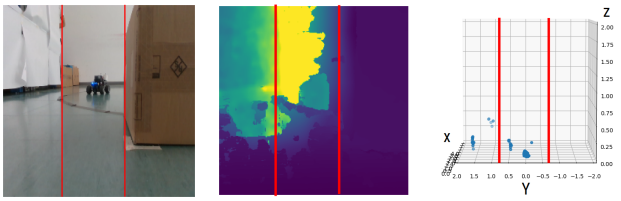



(a) (b) (c)

Fig. 3: Training Losses for Generator (a), Discriminator (b) and Point2Latent (c)

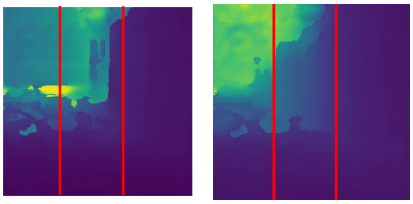
(a) (b) (c) (d)

Fig. 4: Real Colour Image (a), Real Depth Image (b), SA-A prediction (c) and RN-N prediction (d)



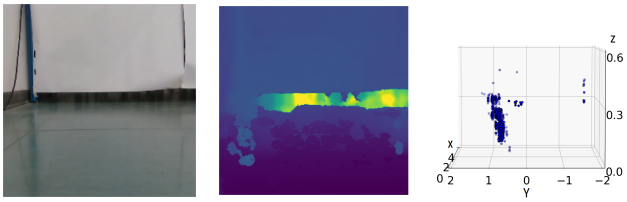
(a) Colour Image (b) Depth Image (c) Point Cloud

Fig. 5: Test 1



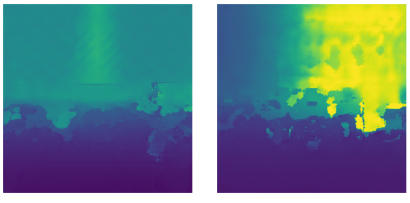
(a) RGB2Depth (b) Point2Depth

Fig. 6: Test 1 - Model predictions



(a) Colour Image (b) Depth Image (c) Point Cloud

Fig. 7: Test 2



(a) RGB2Depth (b) Point2Depth

Fig. 8: Test 2 - Model predictions

## References


[1] P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham, "mid: Tracking and identifying people with millimeter wave radar," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2019, pp. 33–40.

[2] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. of IEEE conference on CVPR*, 2017, pp. 652–660.

[3] J.-T. Huang, C.-L. Lu, P.-K. Chang, C.-I. Huang, C.-C. Hsu, P.-J. Huang, H.-C. Wang *et al.*, "Cross-modal contrastive learning of representations for navigation using lightweight, low-cost millimeter wave radar for adverse environmental conditions," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3333–3340, 2021.

[4] S. M. S. S. Basha, S. Sridhar, S. Kaushik, and H. Kumar, "Millinet: Applied deep learning technique for millimeter-wave based object detection and classification," *IETE Journal of Research*, vol. 0, no. 0, pp. 1–7, 2022.

[5] Y. Sun, Z. Huang, H. Zhang, Z. Cao, and D. Xu, "3DRIMR: 3D reconstruction and imaging via mmWave radar based on deep learning," in *Proc of. IEEE International Performance, Computing, and Communications Conference (IPCCC)*, 2021, pp. 1–8.

[6] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[7] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[8] O. Henaff, "Data-efficient image recognition with contrastive predictive coding," in *ICML*. PMLR, 2020, pp. 4182–4192.

[9] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification." *Journal of machine learning research*, vol. 10, no. 2, 2009.

[10] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *ICML*. PMLR, 2020, pp. 5639–5650.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," pp. 1125–1134, 2017.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[14] Z. Chen, V. Badrinarayanan, G. Drozdov, and A. Rabinovich, "Estimating depth from RGB and sparse sensing," in *Proc. of European Conference on Computer Vision (ECCV)*, 2018, pp. 167–182.

[15] J. Hu, M. Ozay, Y. Zhang, and T. Okatani, "Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1043–1051.