Evaluation of depth images in the real environment generated by Double-GAN

Javier Maldonado-Romo \(^1\)*, Francisco Suárez-Ruiz \(^1\), Mario Aldape-Pérez \(^1\), and Alejandro Rodríguez-Molina\(^2\)

\(^1\) Postgraduate Department, Instituto Politécnico Nacional, CIDETEC, Mexico City 07700, Mexico; jmaladonaro501@alumno.ipn.mx (J.M.-R.); fsuarezr1300@alumno.ipn.mx (F.S.-R.); maldape@ipn.mx (M.A.-P.)

\(^2\) Tecnológico Nacional de México/IT de Tlanepantla, Research and Postgraduate Division, Estado de México 54070, Mexico; alejandro.rm@tlanepantla.tecm.mx

* Correspondence: jmaladonaro501@alumno.ipn.mx; Tel.: +52-555-729-6000

Abstract: For making explorations in 3D environments, unmanned aircraft systems (UAVs) are used, which requires technologies capable of perceiving the environment to map and estimate the location of objects that could cause accidents and collisions. RGB-D sensors can determine the depth, but adding more accessories to the UAVs requires more considerable energy, causing higher dimensions. Due to measurements, it is not elementary to use UAV in indoor environments. UAVs have a conventional camera, allowing images to be capture for mapping 3D indoor environments. The main advantage of the processing of images is Generative Adversarial Networks (GAN) because these networks can generate realistic images from a source of noise. GAN has demonstrated the capability to create images from a set of samples. Therefore, we propose to use GAN to estimate the depth and segmentation of a real picture from a virtual environment representation to enrich a conventional camera that allows estimating depth. Due to there are three domains of samples, a GAN architecture is not enough. For this reason, we propose Double GAN architecture with noise reduction to represent an RGB-D sensor with a few samples of a real scenario. Besides, the comparison of performance with a physical RGB-D sensor such as sensor Kinect, allowing to create low-cost visual depth perception.

Keywords: Computer vision; Robotics-Perception; 3D Mapping; Machine learning

1. Introduction

Artificial intelligence refers to the science area to help machines with the ability to improve functions [1]. Among the functions is the perception, whose function is to interpret the medium with which it interacts. There are different means to interact with a medium, including visual perception. Its purpose is that from signal processing such as cameras build the environment to interact [2]. It is necessary to determine space transformation based on the perspective of smart agent for building the environment [3].

In the field of mobile robotics, the main challenge is the exploration of a 3D environment [4,5]. Perception is required to estimate the position and possible collisions. There are different approaches to implement indoor navigation systems; one of them is using an RGB-D sensor [6–14]. There are other alternatives such as Lidar sensors [15,16] and ultrasonic sensors [17,18], each one has its characteristics, but an RGB-D type sensor is more accessible for measuring distances. Exploration systems require architectures with specialized equipment. Besides, they need structures and long-term
power through that they are ideal on ground exploration equipment. Due to the characteristics, contributions in exploration in UAVs are limited, the main reasons being the battery and size.

2. Related works

State of the art for indoor exploration, they use ground robotic systems. They are using RGB-D sensors like the Kinect sensor [19–24]. Physical sensors such as the Kinect, due to their dimensions, require batteries and structures that can support it, which leads to having large designs. Furthermore, the Kinect sensor has features that limit it, which is the range of at least 40 cm, so the object must be beyond this distance to be perceived and a coverage range of up to 4 meters. The Kinect sensor has a configuration to up to 6 meters, but they have an 80 cm offset; therefore, it is unsuitable in small scenarios [25].

In recent years of research and development in image processing systems, primarily due to the advances in machine learning solutions, including neural networks, especially deep learning. Deep learning had shown limitations. However, a new approach has been opened in processing data for image processing so that images generated from a noise source give rise to Generative adversarial nets (GAN) [27]. GAN networks is an architecture of two types of neuronal networks. The first architecture, called generative network, is responsible for generating data from a noise source. The weights adjust with the evaluation of the second network called discriminator. It is in charge of extracting a set of known characteristics of samples to validate the model generator [30]. This architecture has proposed the following solutions for the generation of images, obtaining incredible results, to the point that the images generated by these architectures are hard to be perceived as a creation of a neural network [26].

GAN networks will generate a representation of an RGB-D sensor from samples created by virtual environment representation, representing the measurement of depth and the segmentation of a real scenario. Therefore, it would allow the performance of a low-cost depth sensor without new additions to be used in conventional cameras, which would maintain the structure and energy consumption, allowing it to use in many UAV systems.

To generate the samples that determine the depth, it used the framework Air-Sim [29] to make the samples using the Unreal Engine video game engine. Since a virtual representation of a real-world scenario. The simulator generates a description of different types of examples that include depth estimation and segmentation. However, one of the limitations is the photorealistic detail level generation that they can represent in a framework to create video games.

Most implementations of GAN network solutions are the interaction between the two domains. We propose an approach where there are a limited number of samples to create a virtual environment due to there is no photorealistic representation of the real scene. Hence, it does not seek to carry out a three-dimensional scenario, but rather that there is a representation of the real world. In the virtual stage, it is in charge of generating samples of depth training and segmentation. So we have three domains: the original examples, the virtual representation of the real environment, and the image of depth and segmentation.

In this work, we want to show that a double architecture of GAN networks can provide high-grade behavior to create depth images of being used in robotic UAV-type systems. Using the same camera they have in their design assembly, without adding attachments that can modify their dynamics or add more charge to the battery. Moreover, it proves that the samples generated in a virtual world from a few real-world information can be used in the real world, enriching the scenario, creating the interpretation of depth and segmentation information to map a 3D environment.
3. Procedure

The first step is to develop a 3D environment from 30 samples of the real environment. The more similar the virtual rendering of an environment with the original image is, the better the behavior will be. The samples generated by the simulator will be 246x144, with an aspect ratio of 16:9. Each pixel represents a centimeter at a distance of 1.5 m with a Field of View (FOV) of 82.6. Figure 1.

![Figure 1. Resolution of the virtual environment.](image)

HOG [28] algorithm allows measuring the level of comparison between real and virtual representation, which obtaining a characteristic vector for each of the samples, by calculating the correlation of the characteristic vector of the original sample with its virtual image obtains a coefficient that indicates the level of similarity. Figure 2 describes the real samples and the virtual representation in two different variations of details. The first variation has essential lighting, and the second has a more significant number of directional lighting sources and materials that give more realism to the virtual scenario.

![Figure 2. a) Real sample, b) Virtual representation with an essential light source, c) Virtual image with more details.](image)

Table 1 describes the average and standard deviation of the coefficient generated from the 30 samples of the virtual representation with simple light and in greater detail from the real world. The correlation coefficient of the model with lights and materials increases, compared to the essential light sample, since the lights increase the level of detail, obtaining a higher correlation. However, the coefficient between the virtual samples created from the video game engine and the real examples is not high enough to determine an adequate representation of the real world. Consequently, using a GAN network to represent the real world to the virtual world increases the correlation coefficient obtaining better behavior.

<table>
<thead>
<tr>
<th></th>
<th>Simple environment Virtual-Real</th>
<th>Environment with lights Virtual-Real</th>
<th>GAN Real-Virtual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor Correlation</td>
<td>0.3708266917630846</td>
<td>0.4990953345845597</td>
<td>0.712894188385744</td>
</tr>
<tr>
<td>(mean)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Factor Correlation</td>
<td>0.0824547241496127</td>
<td>0.0755839752235498</td>
<td>0.1021881800163179</td>
</tr>
<tr>
<td>(std)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Correlation between real-world samples and virtual representation.
From training samples, it is possible to generate images with GAN networks. The architecture has two types of neuronal networks, a generative network and a discriminatory network. The generative network creates samples from a noise source, and the next discriminatory network validates the examples of the images created. The main characteristic of Gan networks is that both networks perform a maximize and minimize competition at the same time.

Equation 1 describes the cost function to both discriminator $D_{1,2}$, since the features of GAN this cost function is maximization, where $G_{1,2}$ is the generator network and $Z_{1,2}$ is the source of the noise.

$$M_{cf_{D_{1,2}}} = \frac{1}{m} \cdot \sum_{i=0}^{m} \log(D_{1,2}(x^i)) + \log(1 - D_{1,2}(G_{1,2}(Z_{1,2}^i)))$$  

(1)

The equation 2 is minimization cost functions that describes the Generator.

$$m_{cf_{G_{1,2}}} = \frac{1}{m} \cdot \sum_{i=0}^{m} -\log(D_{1,2}(G_{1,2}(Z_{1,2}^i)))$$  

(2)

The full cost function 3 is the sum of the 1 and 2.

$$GAN_{cf_{1,2}} = M_{cf_{D_{1,2}}} + m_{cf_{G_{1,2}}}$$  

(3)

Finally, equation 3 is also minimization cost function for noise reduction the last outcome.

$$m_{cf_{NR}} = \frac{1}{m} \cdot \sum_{i=0}^{m} |y_i' - y_i|$$  

(4)

In most of the contributions that implement navigation solutions, they use real-world samples in-state of the art. It is necessary to have access to the place to interact with the real environment to make the data for training. We propose that the first GAN creates an interpretation of the real environment into a virtual world, allowing training and development to be carried out in a virtual environment, using this architecture as an intermediary between both realities.

### 4. Our proposal

In this proposal, there are mainly three domains: the samples of the real world, the second domain is the samples of the virtual scene, and the third domain is the generated images that represent the segmentation and depth information. The segmentation and depth images created from the framework AirSim; both models have the configuration for 2 and 5 meters. Figure 3.

From the implementation, we propose the following architecture 4. It consists mainly of three stages. The first stage is the intermediary, primarily responsible for converting a real image to a virtual representation generated by a GAN architecture network. The second stage is the generator of the depth and segmentation samples [33–35]. Finally, the third stage is noise reduction [31,32]. There are different solutions to reduce the randomness of the outputs. To stabilize the terminal data, since the output of the generated stage is instability in the values; therefore, when it is implementing a noise reducer, the values stabilize, obtaining a uniform distribution of the output.
Figure 3. a) First domain: real samples. b) Second domain: virtual samples. c) Third domain: samples to obtain.

Figure 4. Architecture to create depth and segmentation samples from the virtual environment since real image.

The suggested architecture seeks to obtain data stability from interoperability between three domains, which is required to create a representation of depth and segmentation of a sample taken from the real world.

5. Experimental Phase

The architecture is implemented in TensorFlow 2.0 because of three stages for training; each stage has its memory resources, a g4dn.xlarge instance used in AWS has the following specifications: 4 VCPU XEON 8259CL 2.5GHz, 16GB RAM, 125GB SDD storage, with NVIDIA Tesla T4 GPU with 320 Tensor Core with 16GB RAM, the training time was 3 hours and 10 minutes. The architecture with decoupled design can be separated to carry out each execution independently to reduce resources.

Because errors obtained, the dimensions of the samples are cropped each sample samples as of 256x128; therefore each image 8 pixels were removed from the top and eight more pixels from the bottom of each sample, because when making Pooling operations having multiple dimensions of eight have better behavior. The experiment was carried out with 30 samples for stage 1, which has the first GAN network architecture. It should be remembered that a reconstruction of the virtual stage is planned from a limited number of samples. For the second stage, there are 2000 samples generated from the virtual stage. Finally, for the third stage, the same 2000 virtual samples are used, but they are processed through a serial network of convolutions to reduce the randomness of the data to have
better uniformity in the readings.

Three different types of objects are in the real environment to implement the experiment. Two of the objects used are cardboard boxes differentiated in color and size and a ball. The Sensor Kinect V1 has a near field configuration, an offset of 40 cm, its range of 4 meters for validating the error. The surfaces of objects have a flat and spherical surface to measure consistent data. The samples of the experiment to estimate a depth have distances of 2 m and 5 m. For the 2 m samples, each pixel is equal to 0.7843 cm, while for the configuration of 5 meters is 1.9607 cm, so there may be a more significant error on the 5 m scale.

The experiment had 1000 epochs. Figure 5 describes the result of the complete cost function on a scale of 100 to 1. The graph shows that the $GAN_1$ network is more unstable than the $GAN_2$ network due to the few real-world samples used. Instead, the $GAN_2$ network improves the output, allowing it to obtain uniformity in the data.

![Figure 5. Behavior of the full cost function of the $GAN_{1,2}$ networks.](image)

Figure 6 presents the outcomes of the training of the discriminator, and the generator, in addition to noise reduction. Due to the feedback between two networks, the results of $D_2$ and $G_2$ networks have a stable behavior than $D_1$ and $G_1$ networks. The result of noise reduction has excellent behavior.

Figure 7 presents the final results in the representation of the samples representing depth and segmentation. The results manifest in different stages and environments of the experiment. In turn, a 3D construction of the points with the cloud points of each image, showing four views: front, right side, left side, and top view.

For the validity of Double-GAN, the experiment performs using the virtual samples for training, evaluated with original images. When evaluating the GAN network with its real representation that is the input to estimate a segmentation and depth has the result shown in Figure 7 a), it displays that there is terrific randomness in the output on flat surfaces, for this reason, an intermediary network between both domains.

The following samples correspond to the Double-GAN architecture. Figure 7 b) describes the results that are more consistent with the expected sample; however, when performing the 3D representation, irregularities are observed in the flat parts of the objects. The figure 7 c) displays a
noise reduction implementation to obtain consistent data. Finally, Figure 7 d) shows the results of the test with a depth range of 5m. With this range, more excellent range, obtaining more extra information for exploration.

6. Results and Discussion

The table 2 shows the average and standard deviation of 50 samples of flat surfaces for each distance of three different objects. There are six measurement points in a coverage range of 1 meter; this range is defined because it is a coverage range with Kinect sensor reading. The DG-2 and DG5 models correspond to the architecture output without noise reduction at 2 and 5 meters. The noise elimination tests are labeled DGN-2 and DGN-5 for both distances.

<table>
<thead>
<tr>
<th>Dis. (cm)</th>
<th>DG-2 mean</th>
<th>DGN-2 mean</th>
<th>DG-2 std</th>
<th>DGN-2 std</th>
<th>DG-5 mean</th>
<th>DGN-5 mean</th>
<th>DG-5 std</th>
<th>DGN-5 std</th>
<th>Kinect mean</th>
<th>DGN-5 mean</th>
<th>Kinect std</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>82.41792</td>
<td>69.69397</td>
<td>6.43045</td>
<td>2.45368</td>
<td>106.03873</td>
<td>83.92959</td>
<td>8.74690</td>
<td>3.74782</td>
<td>106.03873</td>
<td>83.92959</td>
<td>8.74690</td>
</tr>
<tr>
<td>90</td>
<td>108.30978</td>
<td>88.26262</td>
<td>6.93490</td>
<td>2.36616</td>
<td>125.59681</td>
<td>110.82624</td>
<td>8.07790</td>
<td>3.03058</td>
<td>125.59681</td>
<td>110.82624</td>
<td>8.07790</td>
</tr>
<tr>
<td>130</td>
<td>144.35014</td>
<td>125.14221</td>
<td>4.65563</td>
<td>4.00988</td>
<td>148.22550</td>
<td>142.67215</td>
<td>8.17382</td>
<td>3.66867</td>
<td>142.67215</td>
<td>142.67215</td>
<td>8.17382</td>
</tr>
<tr>
<td>150</td>
<td>162.39459</td>
<td>142.62788</td>
<td>4.99336</td>
<td>3.55857</td>
<td>164.65840</td>
<td>159.19699</td>
<td>8.38470</td>
<td>3.73801</td>
<td>159.19699</td>
<td>159.19699</td>
<td>8.38470</td>
</tr>
</tbody>
</table>

**Table 2.** The experiment results describe the average and standard deviation of different evaluations, including depth images of 2 m and 5 m with Double GAN, Double GAN with noise reduction, and the results of Kinect.

The results of each reference point display that the DG-2 and DG-5 models have a higher average and standard deviation compared to the remaining models due to the depth resolution range of 256, which generates an error major. On the other hand, noise elimination models have a normal behavior; however, the DGN-2 model has results that are closer to the actual measurement than the DGN-5 model. The difference is due to the value that corresponds to each pixel. Because the error will increase if the range distance increases since it has a resolution of 256, the distribution of the Kinect data has behavior higher. However, it is a dedicated sensor for the distance measure. Even this sensor has an error that increases by the coverage distance.

In summary, all the models have randomness, being the models with noise elimination, and the Kinect has a uniform distribution. The absolute difference generates the DGN-2 model with an offset of 7 cm, the DGN-5 model has an offset of 20 cm, and the Kinect has an offset of 9 cm. The
DGN-2 model has the smallest offset for a coverage range of one meter.

To enrich a conventional camera to estimate the segmentation and depth in an environment, it requires a constant estimation. For this reason, the following experiment builds a 3D environment from a sequence of samples, with a step of 20 cm, Figure 8.

The representation of the 6-frame point cloud is displayed. Figure 9 shows the construction of the stage in 3D with configurations of 2 and 5 meters.

The image manifests the construction of a 3D environment created by the proposed architecture by rendering six frames creating the corresponding point cloud. The consistency persists in the flat areas of the objects. The DGN-2 model data has a fewer range, but reaching just 2 meters produces an unstable building; however, it has a better approximation to representation in the real-world distribution. On the other hand, the DGN-5 model has more coverage, better forming the scenario. Due to the offset error being more significant than the DGN-2 model, there are advantages and disadvantages for each model.

This experiment concludes that the representation of depth and segmentation sensor-generated by representing the virtual samples of a real scene. It is possible to configure the depth range based on

---

Figure 7. Results of the experiment. a) simple-GAN 2, b) DG-2, c) DGN-2, d) DGN-5
needs. However, the measurements obtained in the proposed experiment have obtained acceptable behaviors compared to a Kinect. Even the Kinect sensor also has an error due to the distance being more significant with virtual samples; in the end, an additional offset to the data obtained as parameters to avoid possible collisions. For improving the behavior, the configuration at 5 meters for indoors has a better coverage field; however, instead of using a resolution of 256, it is better to use matrices with a resolution of 16 or 32 bits would have a proper resolution and a good range.

7. Conclusion

The main contribution is to create a low-cost depth sensor through conventional from training samples of limited real images created in a virtual environment. The representation and behavior of real depth and segmentation sensors can be simulated and used in robotic systems with limited design to have dedicated sensors, which allows for lower-cost solutions. A 3D model with the details and lightings necessary to create a virtual representation, creating more detailed models requiring more time and more resources. On the other hand, the disadvantage is the little to no efficiency in scenarios with low lighting, since in physical sensors, it is possible to obtain information in low illumination. It is useful and functional in environments with stable and controlled lighting and the camera focus. Low-cost equipment has cameras with basic functionalities that could alter this proposal of the solution.

As work in the future, the next stage is to implement this method to use a conventional camera to detect moving obstacles. It seeks to use virtual training with N possibilities of combinations of the movements to be recreated in the real world at low cost and good performance.

Funding: This research received no external funding.

Acknowledgments: This work was supported in part by the Secretaría de Investigación y Posgrado (SIP), and in part by the Comisión de Operación y Fomento de Actividades Académicas (COFÁA) of the Instituto Politécnico Nacional (IPN).

Conflicts of Interest: The authors declare no conflict of interest.

References


