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; jmaldonador0501@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: jmaldonador0501@alumno.ipn.mx; Tel.: +52-555-729-6000

Abstract: In order to make explorations in 3D environments, unmanned aircraft systems (UAVs) are used, which require technologies capable of perceiving the environment to map and estimate the location of objects that could cause accidents and collisions. RGB-D sensors help 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 the 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 image from a virtual environment representation to enrich a conventional camera that allows estimating depth. However, the problem is caused by three domains of samples. Thus, basic 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

Robotics research has a long tradition, in which mobile robotics has an important issue, as it avoids obstacles and collisions. One of the major topics to be investigated in this field is the perception of external environments. A common technique is to use the perception [1] through sensors since a sensor is a device that responds to stimulation from input to convert the output into processable data [2], such as camera and Inertial Measurement Unit (IMU), that is used as a complement for Global Navigation Satellite Systems (GNSS) in outsides environments [3]. The camera device converts the light signal into digital data, and the IMU is composed of two principal sensors, such as accelerometers and gyroscope, to change the movement in a digital signal. However, the architectures to use sensors include significant investment and consider a big battery for a long time, increasing the dimensions, consequently using artificial intelligence helps develop alternative techniques to reduce the cost of implementation.

Artificial intelligence refers to the science area to help machines with the ability to improve
functions [4]. The perception is the principal function to add intelligence to autonomous systems, whose function is to interpret the environment to develop a path planning solution. There are different ways to interact with external scenarios, including visual perception as processing signals from devices that allows knowing the features of the environment[5] because the design of some robotic systems is fragile[6].

In mobile robotics, the main challenge is the exploration of a 3D environment because it is essential to move between two points in the shortest possible time[7,8]. This phenomenon has been widely observed [9], practically the path planning problem has two approaches to describe it, either taking the robotic system as a point of reference or focus on the environment. In the first, the analysis is about the movement of the robotic system itself, and we called it the first-person perspective; second, the analysis is considered the robotic system as a particle, called the third-person perspective.

The objective is to develop virtual sensors into conventional camera-based in the third-person perception of the environment on micro-UAV [10], considering that the features of micro-UAVs have not regulated this category since these vehicles pose minimal threat to human life or property, and are adequate to indoor navigation because of dimensions [11]. Besides, the perception allows estimating the position and possible collisions.

There are different approaches to implement indoor navigation systems. One of them is using an RGB-D sensor, which is used in a wide variety of solutions such as complement with RGB data [12], tracking of objects [13]. Improve the accuracy of the objects classification [14], is a complement with other sensors as IMU to improve the location [15,16], allows to create a system to avoid collisions [17,18], and map the environments with the movement of the vehicle [19,20]. Moreover, there are other alternatives such as Lidar sensors [21,22] and ultrasonic sensors [23,24], each one has its characteristics, but an RGB-D type sensor is more accessible for measuring distances.

Exploration systems require architectures with specialized equipment. Therefore, 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, size, and time of flying.

We propose a method to simulate an RGB-D sensor from virtual environments using GAN[25], principally to minimize the resources of the conventional micro-UAV. GAN is attracted considerable interest due to the generation of an image from the noise source. However, we have problems when implementing the GAN on two domains, the real and depth images. Accordingly, we use the virtual domain to create an intermediary between both domains. Thus, the problem of the three-domain implementing a solution with Double-GAN.

In this work, we evaluated the GAN networks to create a different virtual environment and prove the capability to be executed over the real environment. We describe the architecture implemented, performance, and issues using visual perception, comparing with rel RGB-D sensor such as Kinect.

2. Related works

It is well known that machine learning [26] allows to improve the analysis of the perception of the robotics field. Despite the advances of machine learning, there has been less previous evidence for developing sensors based on data. Thus, the ultimate goal is to produce a sensor with conventional sensors using machine learning, minimize cost, and improve the current micro-UAVs with advanced skills that were not included during the process design.
Until now, visual processing has been considered a tool to analyze and find features on the input samples and create the constraints of path planning. However, the machine algorithms have been improved until a grade of generating information from noise data. This topic is essential because we can take streams data from the conventional camera and map them to obtain either a depth image, segmentation image, or both, but we require real sensors to compare the performance and measure if it is possible to simulate depth sensors from the conventional camera.

During the last decade has seen a renewed importance in perception of the environment through RGB-D sensors as Kinect sensor[27]. Physical sensors as Kinect has allowed developing multiple solutions on the field of perception, allows to develop solutions to provide autonomy to ground-vehicles [28], offers solutions in order to navigate around space [29], is fused with other hardware to improve the navigation [30], finally have implemented solutions to avoid [31] and detect obstacles [32,33]. Due to batteries and structures to support them, the dimensions which tend to have high designs. Furthermore, the Kinect sensor has features that limit it, which is the range of at least 40 cm. In consequence, 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; consequently, it is unsuitable in small scenarios [34].

Research in machine learning had shown limitations because just was used over classification [35,36] and regression [37,38] solutions. However, a fundamental approach has been opened in processing data for image processing to generated images from a noise source give rise to Generative adversarial nets are based on a competition approach between two players [39]. 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 [40]. This architecture has proposed the following solutions for the generation of images, obtaining incredible results, such as the transformation of an image to other representation of the data in a different domain[41], generate data to create an image with different machine learning approaches[ 42], generation of sequences without pre-training data[ 43], and the analysis of convergence policies during training. [44].

We have investigated the effect of GAN networks; therefore, we pretend to generate a representation of an RGB-D sensor from samples created by the virtual environment, representing the measurement of depth and the segmentation of a real scenario. Hence, 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 micro-UAV systems.

The design of the samples that determine the depth and segmentation, it is used the framework Air-Sim [45] 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 photo-realistic [46] detail level generation that they can represent in a framework to create video games.

A different approach to the traditional problem is given in GAN architecture because it usually implements two domains that contain a set of samples with similar features[47]. We propose an approach where there are a limited number of samples to create a virtual environment due to there is no photo-realistic [48] 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 environment, it is in charge of generating samples of depth training and segmentation. Thus, we have three domains: the original examples, the virtual representation of the real environment, and the
image of depth and segmentation.

Besides, it is known that the GAN network architecture requires two networks running in parallel means that the memory is upper than typical design. In this proposal, there are at least four simultaneously. Consequently, we recommended using matrix decomposition techniques to reduce training requirements\cite{49,50}.

Furthermore, a technique that is widely used to complement RGB-D sensors is segmentation through the use of different image processing algorithms \cite{51} such as using convolutional networks to separate objects through the use of masks \cite{52}, as well as advantages in improving the detection of objects and disadvantages such as increasing tasks in real-time systems \cite{53}. Therefore, generating both images in the same execution time enriches the virtual sensor. Due to the noise generated by the proposed design, it is necessary to reduce it. Some alternatives are using convolutional networks to reduce noise\cite{54} and to improve the quality of the images\cite{55}.

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

3. Procedure

First of all, it is to build a 3D environment based on a limited real sample of the environment. We are limiting our experiment considering that we have difficult access to the real environment; taking 30 photos renders the 3D space. The more similar the virtual rendering of an environment with the original image will be better the behavior. The samples generated by the simulator have a resolution 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.](https://github.com/javiermr/DoubleGan/dataset)

Previous work has only been limited about data set composed by two domains principally, some data set to evaluate the problem in semantic segmentation are: KITTI (2012) \cite{56}, CityScapes (2016) \cite{57}, and BDD100K (2020) \cite{58}. The GAN architectures are proved and offer adequate behavior. Consequently, we defined the next hypothesis, if we replace the original data set, we could obtain a depth sensor with a semantic image with real sensors data set. Consequently, we expected an outcome with acceptable performance using our data set \footnote{https://github.com/javiermr/DoubleGan/dataset} in order to be used in a particular solution, but we observed that the results were inadequate because there is noise.
The implementation is described in Figure 2, which is composed of an auto-encoder as the generator network and a deep convolutional network as the discriminator. Nevertheless, the output has a behavior unacceptable because of the domains [59], since the domains must share some features[60]. For this reason, the set of the real domain has a different performance from the depth and virtual domain. As a consequence, we need an intermediary to minimize the noise and share some features among the domains. The ideal result is to render the virtual scenery as the real world on a video game platform, but requires time and extra resources. Thus, we must measure the domains in order to compare them and improving performance.

![Figure 2. Simple GAN implementation.](image)

The features of each network are shown in Table 1. The kernel size is four; the strides is equal to two in the Generator network and stride equal to one in the discriminator network. The Relu function is implemented as the activation function in both networks.

<table>
<thead>
<tr>
<th>Generator</th>
<th>Discriminator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoder</td>
<td>Encoder</td>
</tr>
<tr>
<td>1 256,128,3</td>
<td>1 256,128,3</td>
</tr>
<tr>
<td>2 128,64,64</td>
<td>2 128,64,64</td>
</tr>
<tr>
<td>3 64,32,128</td>
<td>3 64,32,128</td>
</tr>
<tr>
<td>4 32,16,256</td>
<td>4 32,16,256</td>
</tr>
<tr>
<td>5 16,8,512</td>
<td>5 1024,1</td>
</tr>
<tr>
<td>Decoder</td>
<td>Decoder</td>
</tr>
<tr>
<td>7 4,2,1024</td>
<td>7 4,2,1024</td>
</tr>
<tr>
<td>6 8,4,1024</td>
<td>6 8,4,1024</td>
</tr>
<tr>
<td>5 16,8,512</td>
<td>5 16,8,512</td>
</tr>
<tr>
<td>4 32,16,256</td>
<td>4 32,16,256</td>
</tr>
<tr>
<td>3 64,32,128</td>
<td>3 64,32,128</td>
</tr>
<tr>
<td>2 128,64,64</td>
<td>2 128,64,64</td>
</tr>
<tr>
<td>1 256,128,3</td>
<td>1 256,128,3</td>
</tr>
</tbody>
</table>

Table 1. Simple GAN parameters

One of the most well-known tools for assessing the correlation between two images is the Histogram
of Oriented Gradients (HOG)[61]. This algorithm allows measuring the comparison between real and virtual representation, which obtains a characteristic vector for each of the samples and obtains a coefficient that indicates the similarity level, whose hyper-parameters are: orientation equal to 8, pixels per cell equal to 32x32, and the cells per block equal to 4x4. Figure 3 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.

Comparing the two results of each domain, it can be seen in Table 2 that the average and standard deviation of the coefficient generated with 30 samples of the virtual representation with simple light and 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 in the video game engine and the real examples is not high enough to determine an adequate representation of the real world. Consequently, using a standard GAN to represent the real world to the virtual world increases the correlation coefficient obtaining better behavior. Thus, we chose GAN as an intermediary between the real and virtual domain.

<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><strong>Factor Correlation</strong></td>
<td><strong>(mean)</strong></td>
<td><strong>(mean)</strong></td>
<td><strong>(mean)</strong></td>
</tr>
<tr>
<td></td>
<td>0.3708266917630846</td>
<td>0.4990953345845597</td>
<td>0.7128941888385744</td>
</tr>
<tr>
<td><strong>Factor Correlation</strong></td>
<td><strong>(std)</strong></td>
<td><strong>(std)</strong></td>
<td><strong>(std)</strong></td>
</tr>
<tr>
<td></td>
<td>0.0824547241496127</td>
<td>0.075839752235498</td>
<td>0.102188180163179</td>
</tr>
</tbody>
</table>

**Table 2.** Correlation between real-world samples and virtual representation.

The benefit of this approach is to reproduce a sensor with virtual information with the GAN network, whose architecture has two types of neuronal networks, a generative and a discriminatory network. The generative network creates samples from a noise source, and subsequently, the 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 simultaneously.

The ultimate goal is to produce an adequate implementation of GAN network based on the 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.

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

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

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

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

$$GAN_{cf_{1,2}} = M_{cf_{D_{1,2}}} + m_{cf_{G_{1,2}}} \tag{3}$$

Finally, equation 3 is likewise a minimization cost function for noise reduction in the last outcome.

$$m_{cf_{NR}} = \frac{1}{m} \cdot \sum_{i=0}^{m} |y'_i - y_i| \tag{4}$$

A number of techniques have been developed over navigation solutions. However, these techniques use real-world samples. Therefore, it is necessary to have access to the place to interact with the real environment to implement mobile robotics 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.

In this proposal, there are mainly three domains: the real-world samples, the second domain is the virtual scene samples, and the third domain is the generated images that represent the segmentation and depth information. The segmentation and depth images were created in the framework AirSim$^2$ though was modified to create distance 2m and 5m respectively; both models have the configuration for 2 and 5 meters. Figure 4.

![Figure 4](https://github.com/javiermr/DoubleGan/airsim)

**Figure 4.** a) First domain: real samples. b) Second domain: virtual samples. c) Third domain: samples to obtain.

We propose the following architecture 5. It consists mainly of three modules. The first is the intermediary, primarily responsible for converting a real image to a virtual representation generated by a GAN architecture network. The second module is the generator of the depth and segmentation samples. Finally, the third module is noise reduction. 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 implementing a noise reducer, obtaining a uniform
distribution of the output.

Figure 5. Architecture to create depth and segmentation samples generated in a the virtual environment since real image.

The complete parameters are described in the Table 3 with respect the same parameter just is added a layer in generator because the final image has 256x256 pixels.

The suggested architecture tries to obtain data stability between three domains, which is required to create a representation of depth and segmentation of a sample taken from the real world. We expect to create different virtual environments in order to generate many samples without to have the real scenery.

4. 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 of problems 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, they have better behavior. The experiment is carried out with 30 samples for module 1, the first GAN network architecture. It should be remembered that a reconstruction of the virtual environment
has a limited number of samples. For the second module, there are 2000 samples generated from the virtual environment. Finally, for the third module, 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 samples.

The scenery is composed of three different types of objects in the real environment to implement the experiment. Two of the objects are cardboard boxes differentiated in color and size and a ball. The Sensor Kinect $V_1$ has a near field configuration, an offset of 40cm, its range of 4m 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 2m and 5m. For the 2m samples, each pixel is equal to 0.7843cm, while for the configuration of 5m is 1.9607cm, there may be a more significant error on the 5m scale.

The experiment has 1000 epochs. Figure 6 describes the result of the complete cost function. 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 7 shows the discriminator training outcomes, the generator, and 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.

On the other hand, Figure 8 presents the final results in representing the depth and segmentation samples. The outcomes 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.

In order to validate the 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 8 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 8 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 8 c) displays a noise reduction implementation to obtain consistent data. Finally, Figure 8 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.

5. Results and Discussion

The table 4 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 m; 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 m. 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>DG-2 std</th>
<th>DGN-2 mean</th>
<th>DGN-2 std</th>
<th>DG-5 mean</th>
<th>DG-5 std</th>
<th>DGN-5 mean</th>
<th>DGN-5 std</th>
<th>Kinect mean</th>
<th>Kinect std</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>62.75409</td>
<td>11.77072</td>
<td>50.89602</td>
<td>2.11002</td>
<td>96.27473</td>
<td>11.51952</td>
<td>62.46814</td>
<td>2.28018</td>
<td>52.04003</td>
<td>1.45995</td>
</tr>
<tr>
<td>70</td>
<td>82.41792</td>
<td>6.43345</td>
<td>69.09397</td>
<td>2.43568</td>
<td>106.03883</td>
<td>8.74690</td>
<td>83.92597</td>
<td>3.74182</td>
<td>66.38124</td>
<td>1.82744</td>
</tr>
<tr>
<td>90</td>
<td>108.50978</td>
<td>6.93490</td>
<td>88.26242</td>
<td>2.36616</td>
<td>125.59681</td>
<td>8.07790</td>
<td>110.82624</td>
<td>3.03588</td>
<td>85.09930</td>
<td>1.96748</td>
</tr>
<tr>
<td>110</td>
<td>125.81072</td>
<td>4.10603</td>
<td>106.47944</td>
<td>1.39989</td>
<td>138.31267</td>
<td>7.26897</td>
<td>126.03647</td>
<td>3.48149</td>
<td>103.89276</td>
<td>1.37248</td>
</tr>
<tr>
<td>130</td>
<td>144.35014</td>
<td>4.60560</td>
<td>125.14221</td>
<td>4.00988</td>
<td>148.22550</td>
<td>8.17382</td>
<td>142.67215</td>
<td>3.66067</td>
<td>122.21436</td>
<td>1.36986</td>
</tr>
<tr>
<td>150</td>
<td>162.39459</td>
<td>4.90386</td>
<td>142.82388</td>
<td>3.55859</td>
<td>164.85840</td>
<td>8.38470</td>
<td>159.19099</td>
<td>3.73801</td>
<td>141.501138</td>
<td>1.35848</td>
</tr>
</tbody>
</table>

Table 4. 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 Kinect measurements.

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 the most useful behavior. 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.

This solution allows to be applied on a conventional camera to estimate the segmentation and depth in an environment 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, using the Kalman Filters to keep the state of movement[62]. It is shown in the Figure 9.

The representation of the 6-frame point cloud is displayed. Figure 10 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, which is 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 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.
6. Conclusion and future work

The main contribution is to create a low-cost depth sensor through virtual samples of a real environment. The representation and behavior of real depth and segmentation sensors can be simulated and implemented in limited robotic systems such as micro-UAV instead of dedicated to using sensors that require much energy, which allows for lower-cost solutions.

A 3D model with the details and lighting needs to create a virtual representation since creating more detailed models requires more time and more resources, and it is not necessary to have access to the real environment. This solution can be used in imaging scenarios with difficult access to being human. Applying virtual reality to create the scenarios allows extending the application on limited robotic systems.

However, it is to change the date in at least two domains compared to other solutions; therefore, it requires an extra time of execution. Thus, implementing this type of machine learning techniques in solutions whose target is to minimize the execution time can reduce it.

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 take advantage of this proposal to
obtain solutions more accessible.

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 great 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 (COFAA) of the Instituto Politécnico Nacional (IPN).

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


