applied sciences

MDPI

Article

A Deep Learning Architecture For 3D Mapping Urban Landscapes

Armando Levid Rodríguez-Santiago [1], José Aníbal Arias-Aguilar [1,†], Hiroshi Takemura [2,†] and Alberto Elías Petrilli-Barceló [2,*,†]

[1] Graduate Studies Division, Universidad Tecnológica de la Mixteca, Km. 2.5 Carretera a Acatlima, Huajuapan de Léon 69000, Oaxaca, Mexico; levid.rodriguez@gmail.com (A.L.R.-S.); anibal@mixteco.utm.mx (J.A.A.-A.)

[2] Department of Mechanical Engineering, Faculty of Science and Technology, Tokyo University of Science, 2641 Yamazaki, Noda 278-8510, Chiba, Japan; takemura@rs.tus.ac.jp

* Correspondence: petrilli@rs.tus.ac.jp

† These authors contributed equally to this work.

check for updates

Citation: Rodríguez-Santiago, A.L.; Arias-Aguilar, J.A.; Takemura, H.; Petrilli-Barceló, A.E. A Deep Learning Architecture For 3D Mapping Urban Landscapes. *Appl. Sci.* **2021**, *11*, 11551. https://doi.org/10.3390/app112311551

Academic Editor: Wonjoon Kim, Sekyoung Youm and Sungbum Jun

Received: 29 September 2021
Accepted: 30 November 2021
Published: 6 December 2021

Abstract: In this paper, an approach through a Deep Learning architecture for the three-dimensional reconstruction of outdoor environments in challenging terrain conditions is presented. The architecture proposed is configured as an Autoencoder. However, instead of the typical convolutional layers, some differences are proposed. The Encoder stage is set as a residual net with four residual blocks, which have been provided with the necessary knowledge to extract the feature maps from aerial images of outdoor environments. On the other hand, the Decoder stage is set as a Generative Adversarial Network (GAN) and called a GAN-Decoder. The proposed network architecture uses a sequence of the 2D aerial image as input. The Encoder stage works for the extraction of the vector of features that describe the input image, while the GAN-Decoder generates a point cloud based on the information obtained in the previous stage. By supplying a sequence of frames that a percentage of overlap between them, it is possible to determine the spatial location of each generated point. The experiments show that with this proposal it is possible to perform a 3D representation of an area flown over by a drone using the point cloud generated with a deep architecture that has a sequence of aerial 2D images as input. In comparison with other works, our proposed system is capable of performing three-dimensional reconstructions in challenging urban landscapes. Compared with the results obtained using commercial software, our proposal was able to generate reconstructions in less processing time, with less overlapping percentage between 2D images and is invariant to the type of flight path.

Keywords: deep learning; CNN; autoencoder; GAN; point cloud; 3D reconstruction; aerial images

1. Introduction

Three-dimensional reconstruction and visual representation is a broadly studied problem and can be used in many applications such as object recognition and scene understanding. State-of-the-art 3D reconstruction algorithms show important results and propose solutions to the Structure from Motion (SfM) and Simultaneous Localization And Mapping (SLAM) problems with important results and proposals that give a solution to these problems [1–11]. Techniques perform localization and mapping, and 3D reconstruction using active sensors (e.g., LiDAR scanners) and passive sensing (e.g., stereo cameras).

However, in 3D reconstruction, none of these methods perform well on practical scenarios, and given the ambiguous correspondences between pixels and 3D spatial points, projection from 2D to 3D remains remarkably difficult and intuitive, these models are typically incapable of producing reliable matches in regions with repetitive patterns, homogeneous appearance, or large illumination change, a typical problem in photogramatry [12–15]. The problem more challenging when working with aerial images of external environments.

Nevertheless, with the current advancements in Deep Learning, it is possible to apply different architectures to obtain similar o better results using and combining different configurations of deep neuronal networks (e.g., Autoencoders). For example, using large existing datasets and considering sensor fusion of data, obtained from a stereo camera and an active 3D LiDAR [16–18], we are able to perform long-range depth estimation and 3D reconstruction.

In this work, our main interest is that from input 2D aerial images, perform three-dimensional reconstructions. Due to the orographic conditions of the state of Oaxaca in Mexico, it is very difficult to obtain data from different areas. Therefore, a three-dimensional model would allow us to have information about the areas of interest. In particular, the conditions of the terrain of the Technological University of the Mixteca, in the highlands of Oaxaca state, Mexico, present variations in height and several areas large amount of homogeneous vegetation. Therefore, a three-dimensional model with point cloud generated with a neural network architecture from only 2D images as input would allow us to obtain important information about the university areas.

Precise 3D reconstruction of the campus is important for several projects, ranging from infrastructure expansion to water recuperation in the roofs and green energy plants location passing for virtual tours proposed to new students. With a surface of 1.0×10^6 m^2 and many high buildings in the campus, we need the support of aerial images to cover the dimension of the university and with the information provided from different perspectives, carry out a digital reconstruction.

The motivation that drives us in this direction is an observation in the sense that the proposals in the current state-of-the-art do not focus in the use 2D aerial images. Furthermore, these proposals use information from other sensors like LiDARs [16–18]. On the other hand, current advances indicate that it is possible to combine different Deep Learning architectures to perform digital 3D reconstructions such as [19,20]. Therefore, in this work, we propose to use an Autoencoder [21–25] architecture and Generative Adversarial Networks [26–30] with a 2D aerial images sequence as input to the proposed neural network architecture to obtaining as output a three-dimensional reconstruction with point clouds.

Our contribution is summarized in two aspects:

1. Based on an Autoencoder architecture, we propose a deep residual neural network for the Encoder stage and a GAN network for the Decoder stage. This configuration generates a point cloud using a sequence of 2D aerial images as inputs. The proposed methodology does not need information from other sensors such as LiDARs to deliver reliable results similar to those of commercial software.
2. This proposal works at different altitudes (100–400 m), at low overlapping percentages between each image (30–80%), and is independent of the flight path used to captured the image sequences of the target area.

2. Related Works

We reviewed the current state-of-the-art and found different methods to perform 3D structure inference of an object using a single image. These works also attempt to solve the SfM and SLAM problems [31–34].

Recent works combine Deep Learning techniques to perform three-dimensional reconstructions, using data from the stereo cameras, mono-LiDAR, and stereo-LiDAR cameras, merging merging the data from these sensors to obtain better results to obtain better results [35–39]. However, these proposals are focused on solutions for individual objects and many of them only focus on reconstructing structured environments. On the other hand, only a few works [40–44] focus on the reconstruction of environments with challenging conditions such as high altitudes, textures, etc. In addition, these works do not contemplate the limitations of working only with monocular and aerial 2D images and only a few of them use deep learning techniques.

Commercial software such as Pix4DMapper, Agisoft Photoscan, DroneDeploy, and others are able to perform 3D reconstruction of outdoor environments, but often require specific configurations to guarantee results. For example, a minimum flight height and overlapping percentages between the image sequence.

3. Network Architecture

The model proposed to infer a complete 3D shape of a landscape and the objects present in the terrain from a 2D aerial images sequence is shown in Figure 1. It consists of an autoencoder configuration, with the main parts being the Encoder, Bottleneck, and Decoder, which are described in detail below.

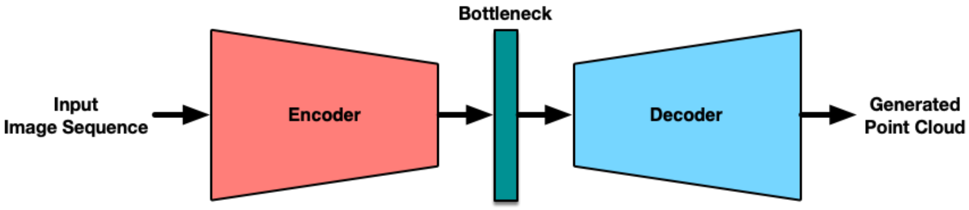


Figure 1. General configuration of the proposed model. The configuration is composed with an Encoder and Decoder and a bottleneck that connects the two.

The Encoder has been configured as a Residual Network (ResNet) and is composed of four Residual Convolutional Blocks. This configuration allows obtaining dense feature maps from the input image sequence. At the output of the Encoder, we obtain data correspondences to be able to generate point clouds with the next stage.

On the other hand, and unlike a classic autoencoder architecture, in this proposal the decoder is based on a Generative Adversarial Network (GAN) architecture. Which is composed of a Generator network and a Discriminator network, capable of generating a point cloud from the input image sequence and the correspondences generated in the previous stage. This stage is called GAN-Decoder.

3.1. Encoder

The encoder stage is set up with four residual blocks. Each one is designed with two convolutional layers followed by batch-normalization layers and Parametric ReLU [45] as the activation function. The network's layers are shown in Figure 2. These layers are used to extract dense features maps.

To correctly generate the geometry, a fully connected layer and an Image Retrieval layer are appended. To obtain key points and point correspondences between each input image sequence. Furthermore, a max-pooling layer and a fully connected layer are added to apply geometric correction.

Compared to traditional convolutional layers, the residual convolutional layers allow for a more efficient extraction of dense features in aerial images, useful to better describe the objects present in the target scene. While the vector of features extracted by the encoder, Dense Feature Vector DFV, has a size of $1 \times 1 \times 1024$ that will be reshaped and concatenated with the point cloud for the training stage.

3.2. GAN-Decoder

The generator proposed is based on a GAN network. It consists of a generator network and a discriminator network. During training, a feature vector from the encoder stage is concantenated with each point of an initial uniformly spaced point cloud, thus making a new feature vector of size 1024×1. Figure 2 shows the improvement and concatenation process. The new vector is used by the GAN-Generator. After three FC layers followed by ReLU, the generator ends with a fully connected layer and a max-pooling layer that predicts the final point cloud with a 1024×3 shape. The difference between our proposal and other proposed networks is how the models are reconstructed. By using an initial point cloud with real feature maps from aerial images, the GAN-Decoder performs better and point cloud inferences. With that, the GAN-Decoder performs better end point cloud inferences.

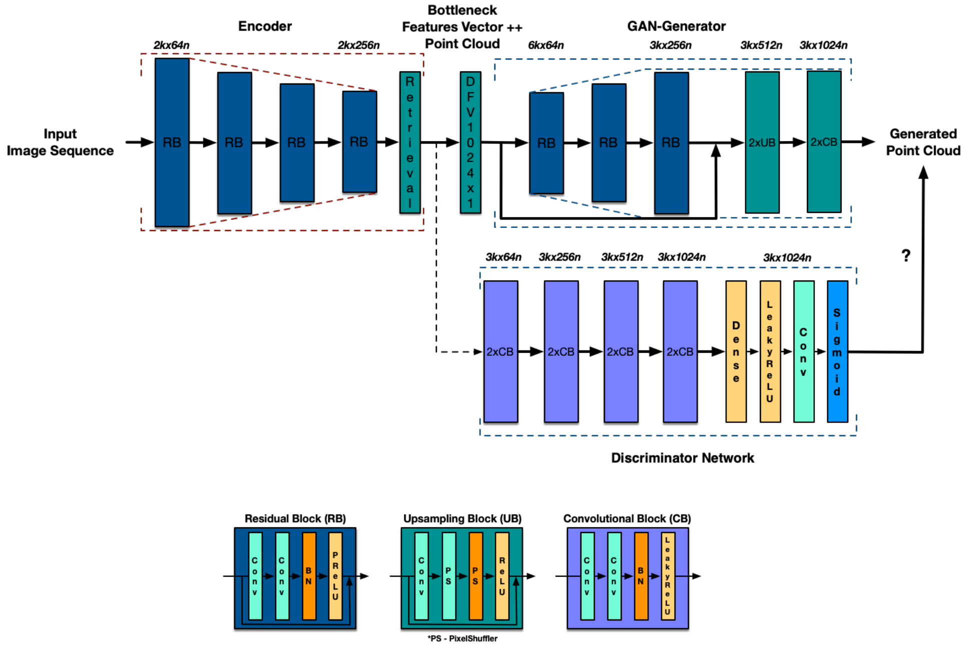


Figure 2. The general configuration of the proposed model is shown in detail. The configuration of the internal layers of each stage of the architecture, both the Encoder and the Decoder, are now called GAN-Decoder. The special blocks used, such as the Residual Block (RB) and the Upsampling Block (UB), and Convolutional Block (CB) are shown in detail at the bottom.

3.2.1. Generator Network

The core of the proposed generator network, which is illustrated in Figure 2, tend identical residual blocks are used in the Encoder stage. To improve the precision of generated point cloud vector, two fully connected and two max-pooling layers are added in the sub-pixel convolution block and trained according to the model proposed by Shi et al. [46]. Although, in that model, a point cloud database is used to increase the knowledge of the proposed model.

3.2.2. Discriminator Network

To discriminate real point clouds from generated, the Discriminator Network follows the architectural guidelines summarized by Ledig et al. [47] and Goodfellow et al. [48] by using a LeakyReLU activation function ($\alpha = 0.3$) and avoiding max-pooling throughout the network, finally we use the sigmoid function to normalize the output of the module. With this configuration, we reduced the complexity of the model and improved processing time.

3.3. Implementation Details

The development of the proposed model, the fine-tuning and the transfer learning were performed in Keras and Tensorflow [49]. The Encoder was trained with the images from a previously generated dataset that includes 2000 aerial images distributed as shown in Table 1. It is important to note that the aerial images from this database have been captured in a circular (Circular Mission) path around the target area. With multiple viewpoints of the area flown over by the drone, is expected to have a 360° perspective of the area of interest.

Table 1. Number of images for each configuration. Images have been captured in a circular path and taken at two different heights and two different overlapping percentages.

Height\Overlapping	**30% × 30%**	**50% × 50%**
100 m.	300	700
150 m.	300	700
Total	600	1400

In this way, it is possible to perform the most detailed reconstruction of the area of interest with multiple objects and complex backgrounds. The model is trained until that the validation precision stopped increasing. To perform fine-tuning and transfer learning, 1500 images were used for training and 500 for validation. Furthermore, we trained for 50 epochs and used a batch size of 20. The training was carried out in a machine with two NVIDIA RTX 2080Ti graphic cards, Ubuntu 19.04 operating system, and 32 GB of RAM memory.

First, we trained the Encoder for approximately 24 h and obtained a training and validation loss of 0.623 and 0.219 (see Figure 3a), respectively, and an training and validation accuracies of 80.25% and 93.75%, respectively (see Figure 3b).

In the second step, the training of the GAN-Decoder is carried out using Adam's optimizer [50], alternatively updating the Generator and Discriminator network. Furthermore, as the Generator uses convolutional blocks with skip connections, similar to those of the RestNet model and identical to those used in the Encoder stage, we decided to use the blocks and their corresponding weights obtained after training the Encoder.

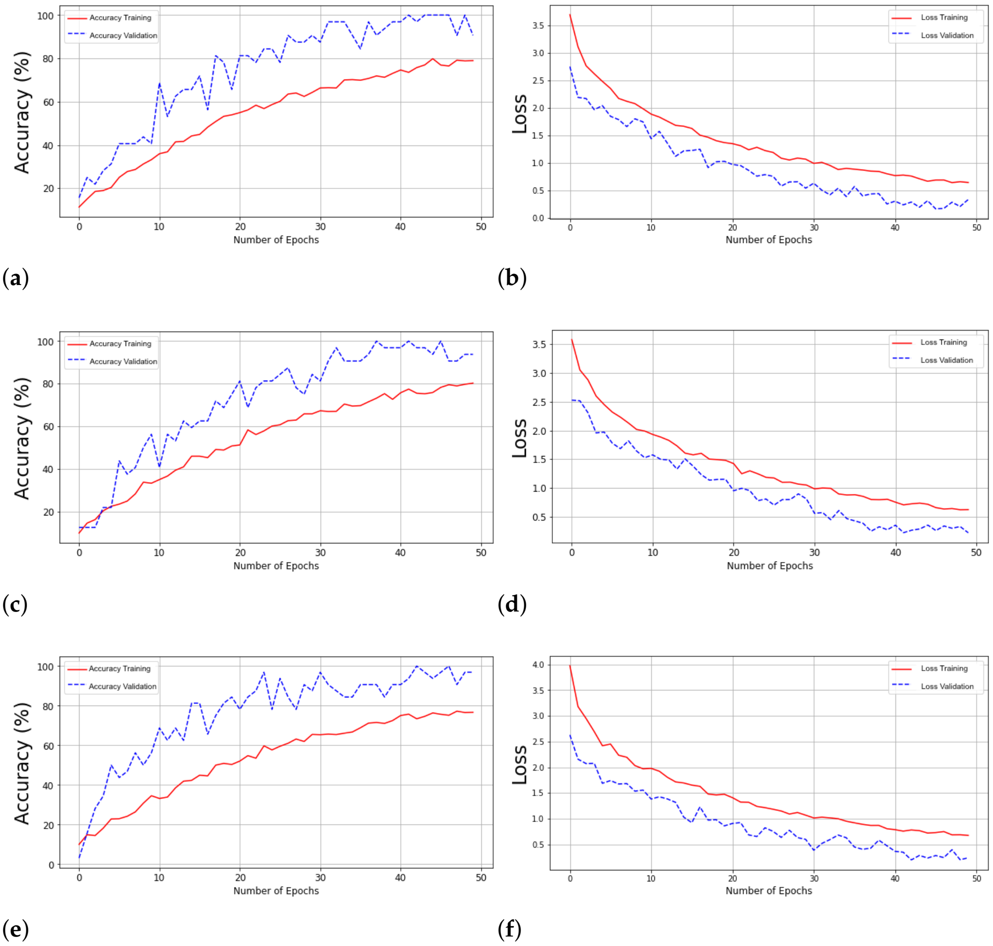
Additionally, following the work in [29], the Discriminating Networks is trained following using the key points obtained from aerial images generated by the Encoder stage and using the maximization function shown in Equation (1).

$$min_G max_D E(D,G) = E_{p^{out} \sim p_{train}(p^{out})}[log D_{\theta_D}(P^{out})] + E_{p^{input} \sim p_G(p^{input})}[log(1 - D_{\theta_D})(G_{\theta_G}(P^{input}))] \quad (1)$$

where p_G is the generator distribution over the input data P, and G_{θ_G} generator with with its specific weights and biases denoted by θ_G. D represents the discriminator, with a D_{θ_D} distribution representing the probability that the data or a point came from the data p_G. The discriminator D is trained to maximize the probability of assigning the correct point to the samples taken from G and, in consequence, minimizes the fail probability of samples generated by G. Moreover, to constrain the range of the discriminator point output, we propose to use a sigmoid activation at the end, we found it useful to stabilize the training in our experiments between the residual input points and the point cloud generated [51–54].

With the above configurations we are able to reduce the complexity of the model and improve processing time. After training the Discriminator in the GAN-Decoder, we obtained a final loss of 0.647 in training and 0.338 in validation (see Figure 3c) and an accuracy of 78.04% in training and 90.63% in validation (see Figure 3d).

Finally, in the third step, we train the complete architecture and obtain a training and validation loss of 0.673 and 0.237, respectively (see Figure 3e), and a training and validation accuracy of 76.68% and 96.88%, respectively (see Figure 3f).



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 3. Accuracy and Loss graphs during training and validation for every stage of the proposed architecture, the Encoder stage (**a**,**b**), GAN-Decoder (**c**,**d**), and full proposed model (**e**,**f**).

4. Experimental Results

The proposal was evaluated using quantitative and qualitative measures, and these show how effective the model is for 3D reconstruction of urban landscapes. As the Pix4DMapper is used in professional applications, its results were used as ground truth and they were compared with the results obtained using our proposed model.

Taking into account the fact that most commercial software, such as Pix4DMapper, DroneDeploy, and DJIGO4, have similar requirements to generate valid reconstructions and, therefore, to compare our results, we decided to use two experiments with different overlapping and height configurations. The first configuration uses a Circular Mission path with 80% overlapping and the second one uses a Grid Mission path with 50% overlapping. For both experiments, the images were taken at $4K$ resolution and at a height of 150 m.

The results for the first experiment are shown in Figure 4, where the first column shows the results obtained with Pix4DMapper and the second one shows the ones obtained with the proposed methodology. Each row contains the reconstructions of different target areas. The point clouds obtained show similar results, but the Pix4DMapper software generates more accurate point clouds using this configuration.

Analyzing the results, it is possible to observe that the proposed model generates a point cloud that represents both the compound shapes and textures and objects as trees, hallways, vehicles, and buildings. The generated point cloud is uniformly distributed and incorporates a considerable part of the selected area and the objects present.

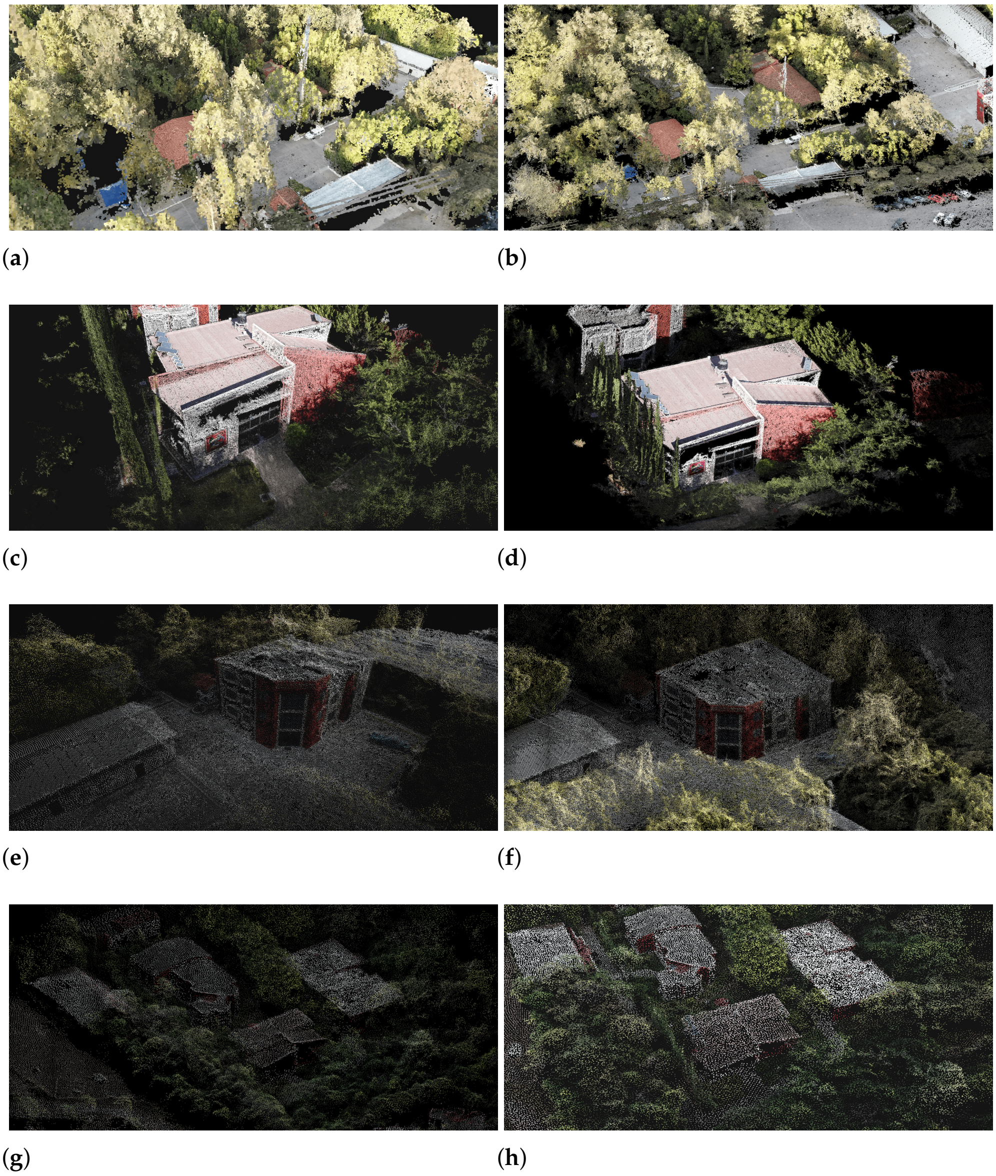

(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 4. (**a**–**h**) 3D reconstructions. Comparison of the qualitative results obtained with Pix4DMapper and the proposed architecture on the different areas of the UTM campus. From left to right: Pix4DMapper or ground-truth and the point cloud generated from the proposed model.

The results have shown that it is possible to recognize the features of each object in the scene. However, there are clear spaces separating the areas, which could be relatively important for the application, however, the results are visually acceptable, and compared to those obtained by commercial software they are valid enough.

Furthermore, a quantitative analysis of the point cloud generated is performed using the Chamfer [55,56] distance (2) and Earth Mover's Distance EMD [57,58] Equation (3) as similarity metrics. The analysis of this evaluation allows recognizing the similarity of the reconstructions (the generated one and that of the commercial software) through the distance between points. In Equations (2) and (3), $\hat{P}$ represents the point cloud generated with our proposal and P represents the one generated using the commercial software. To obtain the Chamfer distance, we find the sum of the squared distances obtained from each point and its closest neighbor. The chamfer distance is smooth and continuous in parts, and the search process is independent for each point. The lower value, the better and more accurate the similarity will be between the two point clouds. In the case of EDM, the bijection $\phi : \hat{P} \rightarrow P$ is employed. In EMD, each point from $\hat{P}$ corresponds to one unique point in P. In this way, it enforces a point-to-point assignment between the two point clouds. Table 2 shows the results of the distance between the point cloud generated by the proposed methodology and the one generated by the commercial software and the low $d_{Chamfer}$ distances indicate their close similarity.

$$d_{Chamfer}(\hat{P}, P) = \sum_{x \in \hat{P}} min_{y \in P} \|x - y\|_2^2 + \sum_{y \in \hat{P}} min_{x \in P} \|x - y\|_2^2 \tag{2}$$

$$d_{EMD}(\hat{P}, P) = min_{\phi: \hat{P} \rightarrow P} \sum_{x \in \hat{P}} \|x - \phi(x)\|_2 \tag{3}$$

Table 2. Comparison between the reconstruction results obtained using Pix4DMapper and the results obtained using our proposal. The Metrics are computed on 1024 points. Additionally, the results are computed using 1400, 700, and 300 aerial images with 80%, 50%, and 30% overlapping percentage, respectively. The distances with no value indicate an a overflow of the data.

Selected Area	$d_{Chamfer}$	d_{EMD}	$t_{proposal}$	$t_{Pix4DMapper}$
Overlapping = 80%				
Main courtyard	11.99	18.51	50 min	90 min
Laboratory buildings	18.56	20.15	98 min	180 min
Institute buildings	21.56	32.15	120 min	180 min
Classroom buildings	18.79	20.23	80 min	120 min
Overlapping = 50%				
Main courtyard	41.99	48.51	30 min	50 min
Laboratory buildings	58.56	50.15	60 min	80 min
Institute buildings	41.56	62.15	80 min	70 min
Classroom buildings	58.79	40.23	50 min	70 min
Overlapping = 30%				
Main courtyard	51.99	45.51	30 min	60 min
Laboratory buildings	–	–	30 min	60 min
Institute buildings	61.56	62.15	40 min	60 min
Classroom buildings	–	–	30 min	60 min

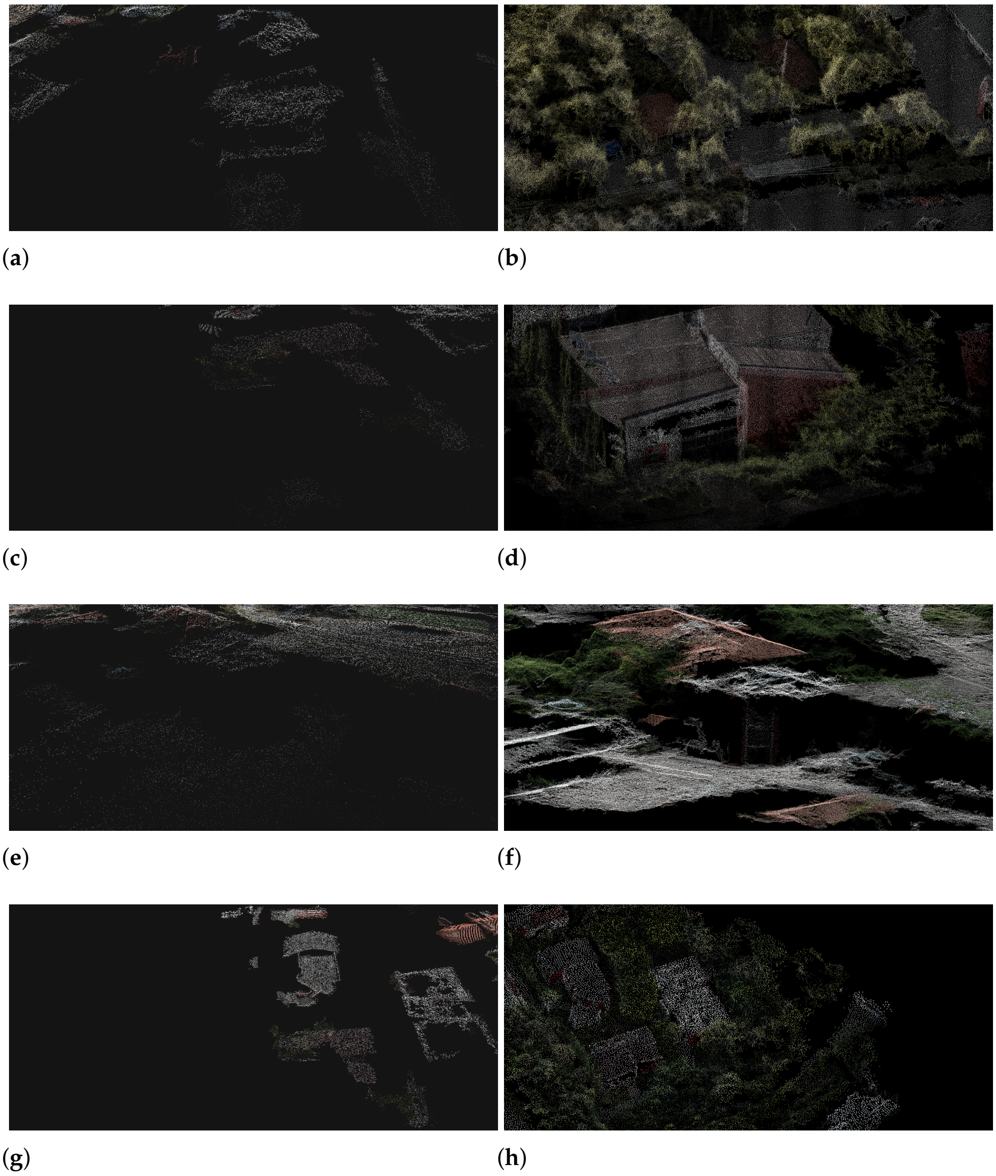
The results of the three-dimensional reconstruction obtained with Pix4DMapper and the results obtained with our proposal are shown in Table 2. To determine the similarity distance $d_{Chamfer}$ and d_{EMD}, between the results with the commercial software and our proposal, are used 1024 points samples, with 1400, 700, and 300 aerial images an overlapping percentage of 80%, 50%, and 30%, respectively, and 150 m height.

The results show that the similarity is very close in percentages of 80% (the minimum percentage that commercial software requires to guarantee results). However, for lower overlapping percentages, commercial software cannot perform a three-dimensional reconstruction, while with the proposed methodology it may do. This causes the similarity metrics to begin to increase (see Figure 5).

Moreover, the processing time of the proposed methodology ($t_{proposal}$) is considerably reduced compared to commercial software ($t_{Pix4DMapper}$). With this proposed, the point cloud generated from aerial images will provide high-quality reconstruction in textures, meshes, and volumetric structures present in the several objects in the urban landscapes. Furthermore, similar enough to those obtained by Pix4DMapper but with less processing time.

In the second configuration, we use the same target areas used with the previous configuration. In this configuration we used the Grid Mission path and the results obtained are shown in Figure 5. The first column shows the results obtained with the Pix4DMapper and the second column shows the results obtained using our proposed method.

In contrast to the previous configuration, the results obtained from this experiment show a clear improvement in 3D reconstruction, when using our proposed methodology. Pix4DMapper and other commercial software require special configurations, such as the flight path and camera configurations. If the configurations are slightly changed, as shown in our tests, the results become unfavorable. In comparison, the methodology presented in this paper is robust to these types of factors. This proposal does not depend on any especial configurations to be able to generate clear and legible point clouds.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

(**g**) (**h**)

Figure 5. (**a**–**h**) Comparison of the results of three-dimensional reconstructions of different areas of the university. The results are obtained after processing images with 50% overlap, taken in a Grid Mission route. The first column shows the results obtained with the commercial software Pix4DMapper and the second column shows the results obtained with our proposed architecture.

It is worth noting that our proposal is able to reconstruct areas that were not in the training dataset (areas outside the university campus) and Figure 6 shows an example. The performance of the proposal is shown by obtaining three-dimensional reconstructions using point clouds. The first column shows the results from Pix4DMapper and the second column shows the results from the methodology proposal.

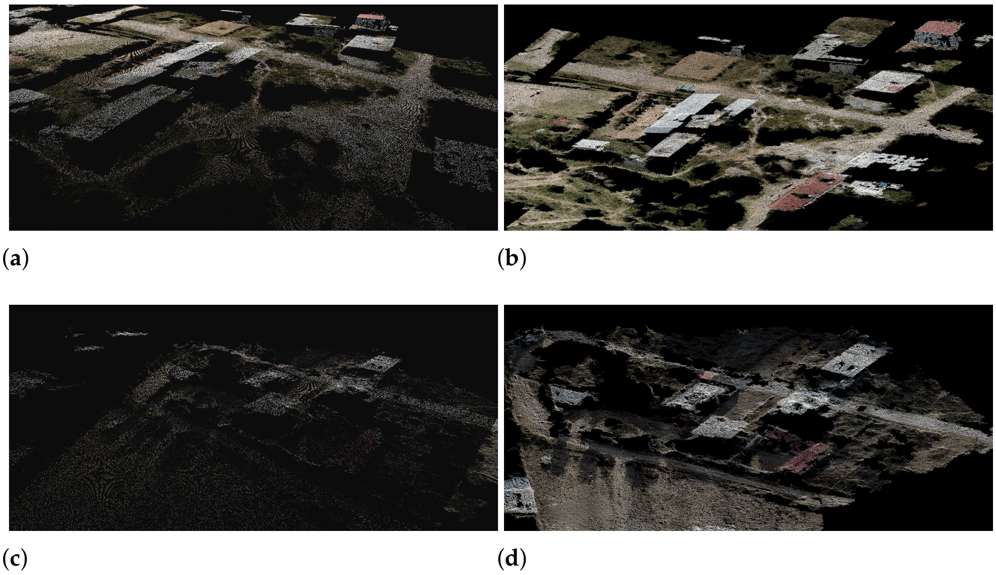

(**a**) (**b**) (**c**) (**d**)

Figure 6. Three-dimensional reconstructions of areas not in the UTM campus. The results obtained using our methodoly (**b**,**d**) are compared with commercial software (**a**,**c**).

On the other hand, we carried out tests at heights varying from 300 m to 500 m. The results show that it is necessary to improve the architecture to be able to perform three-dimensional reconstructions at heights greater than 300 m. Figure 7 shows the reconstruction of an area at 150 m (Figure 7a) and 400 m (Figure 7d). From these results, we can see that, using the images taken at 400 m, the proposed method will not generate enough points to perform 3D reconstruction of the target area.

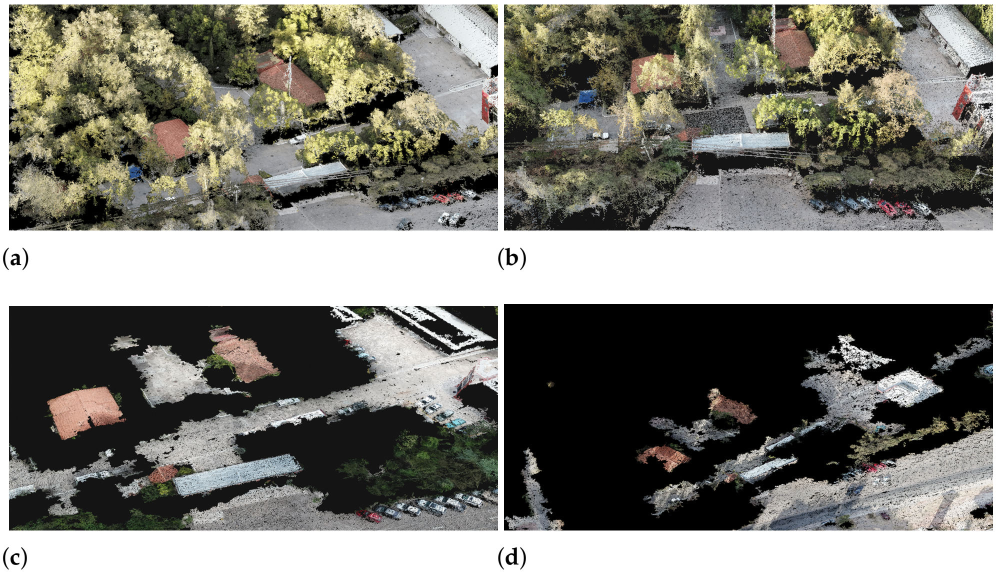

(**a**) (**b**) (**c**) (**d**)

Figure 7. Three-dimensional reconstructions using aerial images taken at a height of 150 m (**a**), 200 m (**b**), 300 m (**c**), and with 400 m (**d**). These results show that our methodology performs 3D reconstructions at different high altitudes. However, at altitudes greater than or equal to 300 m, it presents some difficulties to perform three-dimensional reconstructions.

5. Discussion and Conclusions

In this work, a novel deep neural network architecture was presented for the generation of point clouds from aerial images of urban and natural landscapes. A notable contribution are the Autoencoder settings. The classical configuration was adapted with a residual network in the Encoder stage and a GAN network for the Decoder stage, which has been called GAN-Decoder. Using this architecture, it was possible to obtain results similar to those obtained using commercial software and, in some aspects, even superior.

The proposed methodology is robust to variations of flight configurations for image acquisition. Fundamentally, this methodology does not depend on and does not need a special flight over the interest zone for the acquisition of information. Moreover, it is possible to obtain enough valid results with an inferior overlapping percentage in the images acquired and in less processing time.

The results of this proposal are compared with the results obtained with commercial software. Having a Pix4D Mapper license allowed us to validate and compare our results. Our proposed methodology is robust to variations in flight configurations for images acquisition. Fundamentally, this methodology does not depend on and does not need a special flight over the interest zone during the acquisition of information. Moreover, we are able to obtain results in less processing time and using images with less overlapping percentage.

Additionally, most works presented in the literature focus on three-dimensional reconstruction of controlled environments or solid and individual objects and uses fuse stereo images and point cloud from LiDAR sensors. In comparison, our proposal has focused on the three-dimensional reconstruction of urban landscapes just using from an aerial images sequence. In addition, we take advantage of the potential of GANs to distinguish true and false data without training data with many annotations. Therefore, we improved a discriminating network will make the generator network improve its data generation process. We have thoroughly trained a discriminate, based on the Adam's optimizer [50] and a maximization function (1), until the GAN can no longer distinguish true and false data. With which we have been able to solve the problem with the GAN models is its ability to trust the exit data.

However, working in extreme conditions such as heights above 300 m and exploring areas with homogeneous textures remains a challenge.

Author Contributions: Conceptualization, A.L.R.-S., J.A.A.-A., H.T. and A.E.P.-B.; methodology, A.L.R.-S., J.A.A.-A. and A.E.P.-B.; software, A.L.R.-S.; validation, A.L.R.-S., J.A.A.-A., H.T. and A.E.P.-B.; formal analysis, A.L.R.-S. and A.E.P.-B.; investigation, A.L.R.-S.; resources, A.L.R.-S., J.A.A.-A., H.T. and A.E.P.-B.; data curation, A.L.R.-S. and A.E.P.-B.; writing—original draft preparation, A.L.R.-S., J.A.A.-A., H.T. and A.E.P.-B.; writing—review and editing, A.L.R.-S. and A.E.P.-B.; visualization, A.L.R.-S., J.A.A.-A. and A.E.P.-B.; supervision, J.A.A.-A. and A.E.P.-B.; project administration, J.A.A.-A.; funding acquisition, J.A.A.-A. and A.E.P.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Acknowledgments: The authors would like to thank University Technological of the Mixteca for providing its facilities and technical equipment to carry out this research project. The engineer Roberto Bretado for his support in all the field tests carried out. We would also thanks to CONACyT and the Tokyo University of Science for their financial support.

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

References

1. Beardsley, P.A.; Zisserman, A.; Murray, D.W. Sequential updating of projective and affine structure from motion. *Int. J. Comput. Vis.* **1997**, *23*, 235–259. [CrossRef]
2. Molton, N.; Davison, A.J.; Reid, I. Locally Planar Patch Features for Real-Time Structure from Motion. In Proceedings of the British Machine Vision Conference, London, UK, 7–9 September 2004; pp. 1–10.

3. Pollefeys, M.; Van Gool, L.; Vergauwen, M.; Verbiest, F.; Cornelis, K.; Tops, J.; Koch, R. Visual modeling with a hand-held camera. *Int. J. Comput. Vis.* **2004**, *59*, 207–232. [CrossRef]
4. Akhter, I.; Sheikh, Y.; Khan, S.; Kanade, T. Nonrigid structure from motion in trajectory space. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, USA, 8–11 December 2008; pp. 41–48.
5. Mouragnon, E.; Lhuillier, M.; Dhome, M.; Dekeyser, F.; Sayd, P. Generic and real-time structure from motion using local bundle adjustment. *Image Vis. Comput.* **2009**, *27*, 1178–1193. [CrossRef]
6. Häming, K.; Peters, G. The structure-from-motion reconstruction pipeline—A survey with focus on short image sequences. *Kybernetika* **2010**, *46*, 926–937.
7. Schonberger, J.L.; Frahm, J.M. Structure-from-motion revisited. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4104–4113.
8. Fuentes-Pacheco, J.; Ruiz-Ascencio, J.; Rendón-Mancha, J.M. Visual simultaneous localization and mapping: A survey. *Artif. Intell. Rev.* **2015**, *43*, 55–81. [CrossRef]
9. Carlone, L.; Tron, R.; Daniilidis, K.; Dellaert, F. Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, USA, 26–30 May 2015; pp. 4597–4604.
10. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
11. Ren, R.; Fu, H.; Wu, M. Large-scale outdoor slam based on 2d lidar. *Electronics* **2019**, *8*, 613. [CrossRef]
12. Golparvar-Fard, M.; Pena-Mora, F.; Savarese, S. Monitoring changes of 3D building elements from unordered photo collections. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 6–13 November 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 249–256.
13. Rothermel, M.; Wenzel, K.; Fritsch, D.; Haala, N. SURE: Photogrammetric surface reconstruction from imagery. In Proceedings of the LC3D Workshop, Berlin, Germany, 4–5 December 2012; Volume 8.
14. Nikolakopoulos, K.G.; Soura, K.; Koukouvelas, I.K.; Argyropoulos, N.G. UAV vs. classical aerial photogrammetry for archaeological studies. *J. Archaeol. Sci. Rep.* **2017**, *14*, 758–773. [CrossRef]
15. Zhang, Y.; Wu, H.; Yang, W. Forests growth monitoring based on tree canopy 3D reconstruction using UAV aerial photogrammetry. *Forests* **2019**, *10*, 1052. [CrossRef]
16. Wang, T.H.; Hu, H.N.; Lin, C.H.; Tsai, Y.H.; Chiu, W.C.; Sun, M. 3D lidar and stereo fusion using stereo matching network with conditional cost volume normalization. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 5895–5902.
17. Cheng, X.; Wang, P.; Guan, C.; Yang, R. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 10615–10622.
18. Choe, J.; Joo, K.; Imtiaz, T.; Kweon, I.S. Volumetric propagation network: Stereo-lidar fusion for long-range depth estimation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4672–4679. [CrossRef]
19. Afifi, A.J.; Magnusson, J.; Soomro, T.A.; Hellwich, O. Pixel2Point: 3D object reconstruction from a single image using CNN and initial sphere. *IEEE Access* **2020**, *9*, 110–121. [CrossRef]
20. Zhang, Y.; Liu, Z.; Liu, T.; Peng, B.; Li, X. RealPoint3D: An efficient generation network for 3D object reconstruction from a single image. *IEEE Access* **2019**, *7*, 57539–57549. [CrossRef]
21. Makhzani, A.; Shlens, J.; Jaitly, N.; Goodfellow, I.; Frey, B. Adversarial Autoencoders. *arXiv* **2015**, arXiv:1511.05644.
22. Kingma, D.P.; Welling, M. An Introduction to Variational Autoencoders. *arXiv* **2019**, arXiv:1906.02691.
23. Pinaya, W.H.L.; Vieira, S.; Garcia-Dias, R.; Mechelli, A. Autoencoders. In *Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 193–208.
24. Tolooshams, B.; Dey, S.; Ba, D. Deep residual autoencoders for expectation maximization-inspired dictionary learning. *IEEE Trans. Neural Networks Learn. Syst.* **2020**, *32*, 2415–2429. [CrossRef]
25. Zamorski, M.; Zięba, M.; Klukowski, P.; Nowak, R.; Kurach, K.; Stokowiec, W.; Trzciński, T. Adversarial autoencoders for compact representations of 3D point clouds. *Comput. Vis. Image Underst.* **2020**, *193*, 102921. [CrossRef]
26. Metz, L.; Poole, B.; Pfau, D.; Sohl-Dickstein, J. Unrolled Generative Adversarial Networks. *arXiv* **2016**, arXiv:1611.02163.
27. Makhzani, A.; Frey, B. PixelGAN Autoencoders. *arXiv* **2017**, arXiv:1706.00531.
28. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]
29. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
30. Mo, S.; Zabaras, N.; Shi, X.; Wu, J. Integration of adversarial autoencoders with residual dense convolutional networks for estimation of non-Gaussian hydraulic conductivities. *Water Resour. Res.* **2020**, *56*, e2019WR026082. [CrossRef]
31. Bresson, G.; Alsayed, Z.; Yu, L.; Glaser, S. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Trans. Intell. Veh.* **2017**, *2*, 194–220. [CrossRef]
32. Nüchter, A.; Lingemann, K.; Hertzberg, J.; Surmann, H. 6D SLAM—3D mapping outdoor environments. *J. Field Robot.* **2007**, *24*, 699–722. [CrossRef]

33. Suzuki, T.; Amano, Y.; Hashizume, T.; Suzuki, S. 3D terrain reconstruction by small unmanned aerial vehicle using SIFT-based monocular SLAM. *J. Robot. Mechatronics* **2011**, *23*, 292–301. [CrossRef]
34. Shang, Z.; Shen, Z. Real-time 3D Reconstruction on Construction Site using Visual SLAM and UAV. *arXiv* **2017**, arXiv:1712.07122.
35. Kurenkov, A.; Ji, J.; Garg, A.; Mehta, V.; Gwak, J.; Choy, C.; Savarese, S. Deformnet: Free-form deformation network for 3d shape reconstruction from a single image. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 858–866.
36. Mandikal, P.; Navaneet, K.L.; Agarwal, M.; Venkatesh Babu, R. 3D-LMNet: Latent Embedding Matching for Accurate and Diverse 3D Point Cloud Reconstruction from a Single Image. *arXiv* **2018**, arXiv:1807.07796.
37. Wang, N.; Zhang, Y.; Li, Z.; Fu, Y.; Liu, W.; Jiang, Y.G. Pixel2mesh: Generating 3d mesh models from single rgb images. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 52–67.
38. Lu, Q.; Lu, Y.; Xiao, M.; Yuan, X.; Jia, W. 3D-FHNet: Three-dimensional fusion hierarchical reconstruction method for any number of views. *IEEE Access* **2019**, *7*, 172902–172912. [CrossRef]
39. Lu, Q.; Xiao, M.; Lu, Y.; Yuan, X.; Yu, Y. Attention-based dense point cloud reconstruction from a single image. *IEEE Access* **2019**, *7*, 137420–137431. [CrossRef]
40. Xiang, Y.; Choi, W.; Lin, Y.; Savarese, S. Data-driven 3d voxel patterns for object category recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1903–1911.
41. Tang, J.; Folkesson, J.; Jensfelt, P. Geometric Correspondence Network for Camera Motion Estimation. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1010–1017. [CrossRef]
42. Chen, Y.; Wang, Y.; Lu, P.; Chen, Y.; Wang, G. Large-scale structure from motion with semantic constraints of aerial images. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Guangzhou, China, 23–26 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 347–359.
43. Dissegna, M.A.; Yin, T.; Wei, S.; Richards, D.; Grêt-Regamey, A. 3-D reconstruction of an urban landscape to assess the influence of vegetation in the radiative budget. *Forests* **2019**, *10*, 700. [CrossRef]
44. Özdemir, E.; Toschi, I.; Remondino, F. A multi-purpose benchmark for photogrammetric urban 3D reconstruction in a controlled environment. In Proceedings of the Evaluation and Benchmarking Sensors, Systems and Geospatial Data in Photogrammetry and Remote Sensing, Warsaw, Poland, 16–17 September 2019; Volume 42, pp. 53–60.
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
46. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
47. Ledig, C.; Theis, L.; Huszár, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4681–4690.
48. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Adv. Neural Inf. Process. Syst. (NIPS)* **2014**, *27*, 2672–2680.
49. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.
50. Kingma, D.; Ba, J. ADAM: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015.
51. Wu, B.; Zhou, X.; Zhao, S.; Yue, X.; Keutzer, K. SqueezeSegV2: Improved Model Structure and Unsupervised Domain Adaptation for Road-Object Segmentation from a LiDAR Point Cloud. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4376–4382. [CrossRef]
52. Ni, D.; Nee, A.; Ong, S.; Li, H.; Zhu, C.; Song, A. Point cloud augmented virtual reality environment with haptic constraints for teleoperation. *Trans. Inst. Meas. Control* **2018**, *40*, 4091–4104. [CrossRef]
53. Hui, Z.; Jin, S.; Cheng, P.; Ziggah, Y.Y.; Wang, L.; Wang, Y.; Hu, H.; Hu, Y. An Active Learning Method for DEM Extraction from Airborne LiDAR Point Clouds. *IEEE Access* **2019**, *7*, 89366–89378. [CrossRef]
54. Wang, H.; Jiang, Z.; Yi, L.; Mo, K.; Su, H.; Guibas, L.J. Rethinking Sampling in 3D Point Cloud Generative Adversarial Networks. *arXiv* **2020**, arXiv:2006.07029.
55. Navaneet, K.; Mandikal, P.; Agarwal, M.; Babu, R.V. Capnet: Continuous approximation projection for 3d point cloud reconstruction using 2d supervision. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8819–8826.
56. Nacken, P.F. Chamfer metrics in mathematical morphology. *J. Math. Imaging Vis.* **1994**, *4*, 233–253. [CrossRef]
57. Zhang, Z.; Zhang, Y.; Zhao, X.; Gao, Y. EMD Metric Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
58. Mandikal, P.; Radhakrishnan, V.B. Dense 3d point cloud reconstruction using a deep pyramid network. In Proceedings of the 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, 7–11 January 2019; pp. 1052–1060.