Synthetic Data Generation for 3D Mesh Prediction and Spatial Reasoning During Multi-Agent Robotic Missions

James E. Ecker*, Benjamin N. Kelley† and B. Danette Allen‡
NASA Langley Research Center, Hampton, VA, 23681, USA

In-space assembly operations require accurate reasoning over the pose, location, and structural organization of both the autonomous agents and assembly materials. In a full six-degree-of-freedom space, an accurate understanding of the full three-dimensional structure of the object of interest greatly enriches information for pose estimation and safe path planning. Current methods of predicting pose estimation require a priori understanding of the shape of the object. Additionally, visual information in the space environment is impacted by variations in contrast and illumination. Using synthetic data allows us to rapidly generate large datasets within varying environments and lighting conditions.

This work details the generation of synthetic data used to explore the application of a region-based convolutional neural networks to detect objects of interest and predict a voxel-based three-dimensional mesh in order to understand their full three-dimensional shape. This mesh provides useful spatial information during in-space assembly operations without requiring either the complexity of maintaining models over the progress of building an object or observations from multiple angles. The generated meshes are then compared to that of ground truth in order to measure its performance.

I. Nomenclature

CNN	=	Convolutional Neural Network
R-CNN	=	Region Convolutional Neural Network
GAN	=	Generative Adversarial Network
VGG-19	=	Visual Geometry Group 19 layer CNN
CAD	=	Computer Assisted Design
RPN	=	Region Proposal Network
RoI	=	Region of Interest

II. Introduction

FULL three-dimensional (3D) structural understanding of objects is a key enabling competency for autonomous in-space assembly agents. Computer vision approaches to various scene-understanding tasks, such as object detection, classification, and pose-estimation are particularly difficult in space environments. Variations in illumination, angle, orientation, and movement of objects under observation introduce complexities far greater than those experienced on Earth. Energy and mass considerations for in-space missions necessitate minimal on-board sensors and conservative movement in the operational environment. In this paper, we explore the use of neural network based object detection, masking, and 3D mesh projection models to provide understanding of an object's full three-dimensional, volumetric configuration with a single, two-dimensional, pixel-only observation from a single camera. Using a single camera and a single observation allows for cost-effective agent mass and navigational complexity, respectively.

Autonomous In-space Assembly (ISA) and, more broadly, On-orbit Servicing, Assembly and Manufacturing (OSAM) are of strategic importance to NASA as we develop space transportation systems for access beyond low-earth orbit and look from the Moon to Mars. NASA Langley Research Center leads the area of In-Space Assembly and is responsible for ensuring "that NASA leverages the burgeoning autonomy technology area" [1] to develop multi-agent

*Research Computer Scientist (AST), Autonomous Integrated Systems Research Branch, NASA Langley Research Center
†Research Computer Scientist (AST), Autonomous Integrated Systems Research Branch, NASA Langley Research Center
‡Senior Technologist (ST) for Intelligent Flight Systems, NASA Langley Research Center, and AIAA Associate Fellow.

robotic systems necessary for large telescopes [2], deep space infrastructure [3], and planetary surface habitats [4] needed for the future of space exploration.

In support of NASA multi-agent in-space assembly missions we generate synthetic data from our simulation environment which will be used to train the various models we explore. This data consists of single observations of in-space assembly parts. We vary relative camera location and angle as well as light sources in order to effect robustness against many of the aforementioned complexities in using vision-based approaches in space environments.

We show the utility of this synthetic data when used to train a combination of object detection, mask prediction, and 3D mesh prediction models of the couplers used to join modular components of a large space system. These models allow for generating a three-dimensional model, accurate with respect to the predicted mask of the object, using a single observation from a single camera. This work was completed as part of the Autonomy Teaming & TRAjectories for Complex Trusted Operational Reliability (ATTRACTOR) project.

III. Related Work

This study explores the use of 3D mesh prediction systems to enrich monocular, camera based data without the use of other sensors. We use methods from both Mask R-CNN [5] and Mesh R-CNN [6] directly and discuss them in more detail in later sections.

In Assistive Relative Pose Estimation for On-orbit Assembly using Convolutional Neural Networks [7], the authors use a parallel VGG-19 [8] architecture to predict the pose of a single object from a single camera view. To train this system they use data generated from a simulation environment, similar to our own dataset.

In 3D Point Cloud Generation from 2D Depth Camera Images using Successive Triangulation [9], the authors are able to generate a 3D point cloud from several images of an object from various angles.

Learning Localized Representations of Point Clouds with Graph-Convolutional Generative Adversarial Network [10] studies the use of Graph Convolutional layers within the generator of a Generative Adversarial Network [11]. Similarly, the authors of Spectral-GANS for High Resolution 3D Point Cloud Generation [12] propose a generative adversarial network that operates entirely in the spectral domain to generate scalable high-dimensional point clouds.

IV. Synthesized Dataset

The classes of neural network models used in this study have a high sample complexity requirement for training, requiring a training set containing on the order of thousands of images [13] [5] [6]. Collecting the amount of real world images required for such training proves to be difficult given an operational environment in space. As such, we rely on simulated observations which can be generated on-demand using accurate assembly computer aided design (CAD) models and varying backgrounds. We can place three dimensional instances of assembly objects "in space" and collect these observations automatically, as shown in Figure 1.

Fig. 1 Example of image generated from simulation environment.

Models trained using simulated data introduce significant error when validated against real-world data [14]. In order to mitigate this issue, we employ Domain Randomization [15] by generating instances of assembly objects in various environments on the ground in addition to space environments. This allows the model to focus on aspects of the image which contribute the most to variances in the data and minimize the influence of those which are randomized (in our

case, the environmental background).

Each generated image is paired with a java script object notation (JSON) file containing metadata for its respective image. This metadata serves as target labels for the various models we train. Tables 1 and 2 describe the metadata schema for each instance of data when predicting the mask and 3D mesh, respectively.

Key	Description
camera	camera attributes
camera:position	camera's ground truth position w.r.t the origin in the simulated environment
regions	each region corresponds to an object in the image
region:orientation	orientation of the object w.r.t. the origin of the simulated environment
region:position	object's ground truth position w.r.t the origin in the simulated environment
region:bounding box	ground truth for a box localizing each object in the image
region:bounding box:x	the x coordinate for the bounding box's top left corner
region:bounding box:y	the y coordinate for the bounding box's top left corner
region:bounding box:width	length of the bounding box along the x axis
region:bounding box:height	length of the bounding box along the y axis
region:mask	ground truth for a mask containing all of the pixels segmenting the object
region:mask:vertices	list of x/y coordinates defining the object's mask
region:class	the objects classification target
region:shape attributes	ground truth for a mask containing all of the pixels segmenting the object
region:shape attributes:name	prescribes the class of the mask's shape
region:shape attributes all points x	list of x coordinates defining the object's mask
region:shape attributes all points y	list of y coordinates defining the object's mask

Table 1 Mask R-CNN JSON Metadata

Key	Description
camera:focal length	estimated focal length in mm
camera:in-plane rotation	estimated in-plane camera rotation assuming object at origin
region:2d keypoints	positions of 2D keypoints on the object
3D keypoints	path to file describing the keypoints of the object's 3D model
model	path to the object's 3D model file
voxel	path to the object's voxelized 3D model

Table 2 Mesh R-CNN JSON Metadata

We initially generate a parent dataset consisting of 20,000 image/metadata pairs from which we uniformly sample a training dataset for each round of training. This allows us to vary the size of each training set, bounded by the size of the parent, as a hyperparameter for each model. The process of sampling and building the training datasets includes merging and configuring the data in accordance to the requirements of each model. We further split the data into training and validation sets for evaluating the model after training. Each instance will contain at least one and at most two annotated objects. The objects will not necessarily be completely in frame and may be occluded.

V. Mask Prediction

Predicting the segmentation masks of each object in the scene is achieved using the Mask R-CNN [5] architecture trained on our custom dataset.

A. Mask R-CNN Advantages

Mask R-CNN is a state-of-the-art object detection system built on the ResNet-50-FPN [16] convolutional network. It introduces a region proposal network (RPN) and the region of interest align (RoIAlign) operation [5] in order to predict a category classification, bounding box, and segmentation mask for each object and its corresponding region in an input image. There are two main advantages to using Mask R-CNN which contribute to learning object-level 3D meshes: Transfer Learning and the RoIAlign operation.

1. Transfer Learning

When solving a problem, it is beneficial to consider the solution to a similar problem. When considering the problem of detecting a particular object, one can leverage the weights of an object detection network trained to detect another object by using them to initialize the new network they are training. This process is called Transfer Learning [17].

The Residual Network (ResNet) [16] was first developed in order to mitigate the vanishing gradient problem inherent in the ever growing deep neural network architectures which preceded it. At the time, deep learning architectures were evaluated on their object detection performance on the ImageNet [18] dataset. Researchers found that designing deeper architectures with increasingly smaller convolutional filters generally resulted in better performance on ImageNet.

Eventually, however, neural networks can reach a depth in which they stop converging due to limitations with the back propagation algorithm [19]. The gradient signal decreases as it passes back through the learning network during the learning phase. As networks become deeper, the gradient signal begins to become small enough that earlier layers in the network don't receive enough information to change their weights and learning in the network suffers.

The authors in [16] found that, instead of the traditional sequential path between layers in a neural network, they could introduce what they call a "skip connection" which allows the input to bypass some n number of layers and add them before the activation from the $n + 1$ layer. They call this construct a "residual block."

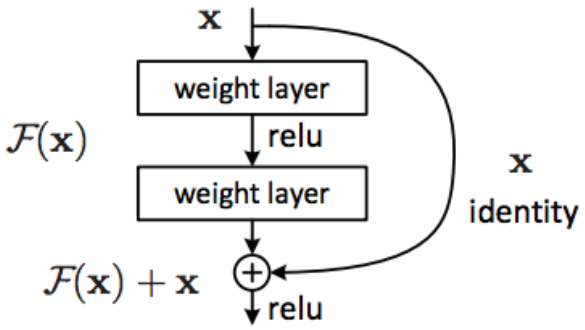

Fig. 2 A residual block [16]

The skip connection is key to mitigating vanishing gradients. Figure 2 shows a typical configuration of a residual block along with the skip connection between its input, x, and the residual block's final layer output. The skip connection introduces a secondary path for the learning gradient which allows information to get to earlier layers. Stacking residual blocks into deep neural network architectures allowed for deeper networks to be built and achieve state-of-the-art results on ImageNet [16]. Mask R-CNN uses ResNet-50 as its backbone network. Bootstrapping off of ResNet-50 allows the network to take advantage of transfer learning [17] by using it as a starting point without having to train a network from random initial weights.

2. RoIAlign

One of the key contributions of [5] is the introduction of the Region of Interest Align (RoIAlign) operation to solve the problem of quantized stride in previous region-based object detection systems such as its immediate predecessor, Faster R-CNN [20].

The authors of Faster R-CNN applied a process known as Region of Interest Pooling (RoIPooling) [21], shown in Figure 3. RoI Pooling is a feature map downsampling scheme which takes feature maps output by a convolutional network and downsamples them according to any proposed regions of interest to which they correspond. Detecting multiple objects in one image requires detecting regions in which an object may occupy. Each of these objects may be

different sizes, so region proposals should be able to vary in dimension. This causes a problem when attempting to send the features localized within each RoI to a fixed-size fully connected layer for further processing for classification.

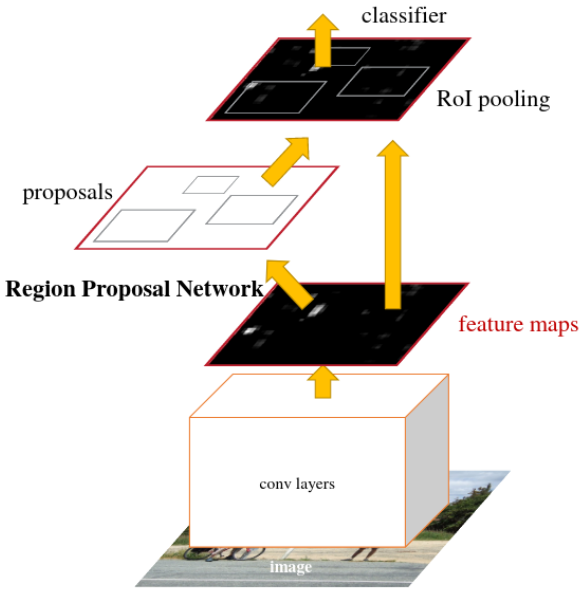

Fig. 3 Faster R-CNN Feature Mapping, RPN, and ROI Pooling architecture [20]

RoI Pooling addresses this problem by quantizing the stride respective to the features mapped to the RoI such that the features being pooled match the fixed size of the fully connected layer. However, this can lead to precision loss since quantization discretizes a continuous space. In order to fix this problem, RoIAlign no longer quantizes according to a desired output size. Instead, it interpolates the features, via bilinear interpolation, in such a way that the downsampling outputs to the desired size without throwing away any features in the RoI (information loss) nor adding any features which didn't belong in the RoI (misalignment). The resulting alignment of the features in the RoI allows the network to predict a pixel-level segmentation mask in addition to a classification and bounding box.

B. Mask R-CNN Performance

The Mask R-CNN architecture expands on Faster R-CNN by adding a third parallel branch for predicting the segmentation mask, enabled by the RoIAlign operation. The Mask R-CNN architecture is shown in Figure 4. By replacing the RoI Pooling operation in the RPN with RoIAlign, a new branch for predicting the mask is added while maintaining the classification and bounding box prediction branch from Faster R-CNN.

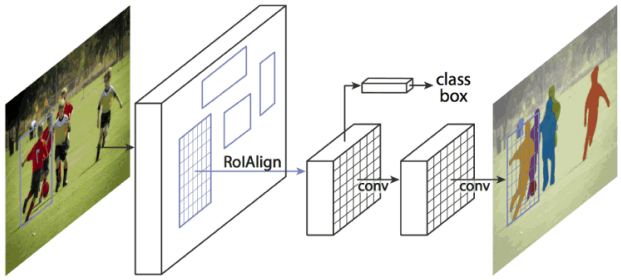

Fig. 4 The Mask R-CNN architecture [5]

A fully trained Mask R-CNN model allows us to efficiently categorize and localize objects in each observation. Figure 5 shows the results of using the trained Mask R-CNN on dynamically generated images which the Mask R-CNN never saw in training. We are able to leverage transfer learning from a previously trained Mask R-CNN model in order to train a new Mask R-CNN for predicting specific in-space assembly couplers in a sample efficient manner (1500 instances). In addition to using the predictive data directly, we input the predicted masks when predicting the 3D mesh of an object in the observation. Training Mask R-CNN against our custom dataset using 1500 instances of training data and 500 instances used for validation yields a model with an average 98.9% classification and segmentation accuracy.

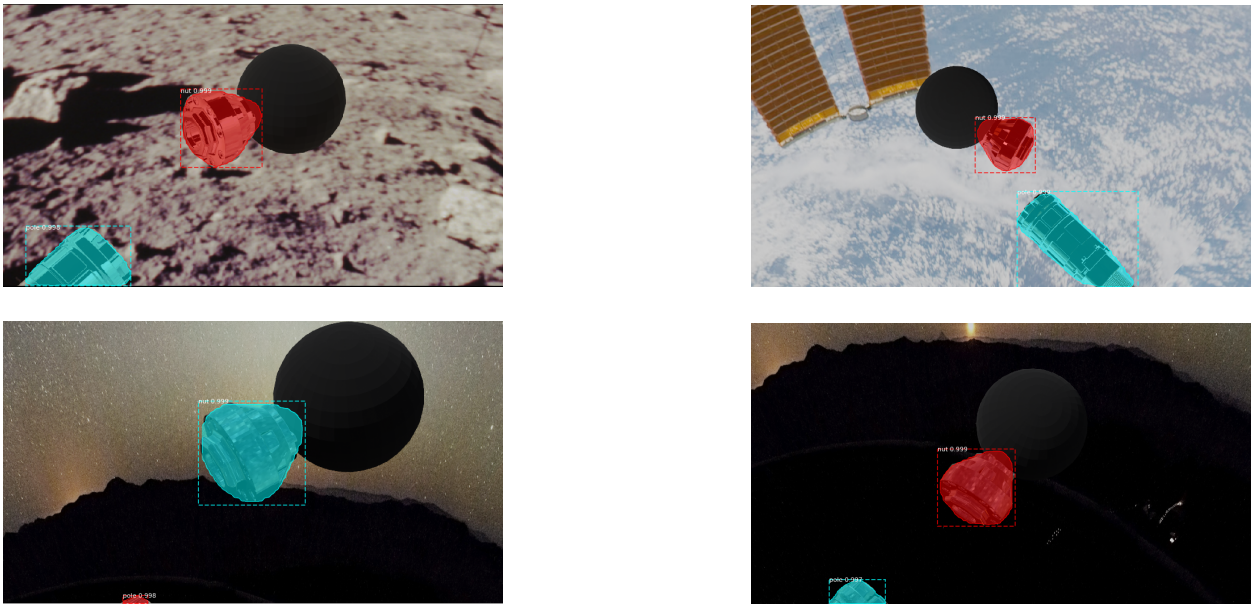

Fig. 5 Examples of predicted masks for ISA couplers.

VI. Mesh Prediction

Mesh R-CNN [6] expands on Mask R-CNN [5] by adding a mesh predictor in addition to the classification, bounding box, and segmentation mask predictor from Mask R-CNN as shown in Figure 6. The mesh predictor is composed of two new branches in the architecture, the voxel and the mesh refinement branches.

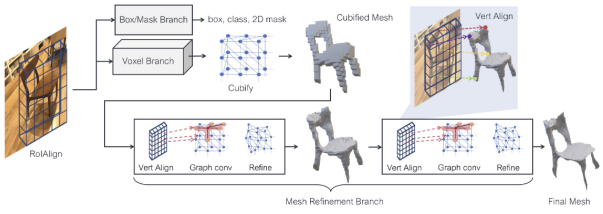

Fig. 6 The Mesh R-CNN architecture [6]

A. Voxel Branch

The voxel branch can be seen as the 3D analogue of Mask R-CNN's segmentation mask branch [6]. Instead of predicting a 2D mask containing all of the pixels associated with a detected object it predicts a voxel occupancy probability grid. This occupancy probably grid describes the object's three dimensional shape as a coarse low-resolution projection of cubes, with distances interpolated using the source camera's intrinsic matrix [6]. This process is illustrated in Figure 7.

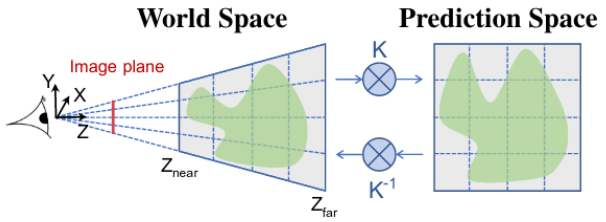

Fig. 7 Interpolation of pixel distances using source camera's intrinsic matrix K [6]

The predicted voxel occupancy grid is then transformed into a triangle mesh via the Cubify operation [6]. The Cubify operation binarizes the voxel occupancy probabilities and then generates a cuboid triangular mesh for each voxel which is occupied.

B. Mesh Refinement Branch

The triangular mesh produced by the Cubify operation is then refined into a higher resolution by the Mesh Refinement Branch. The Mesh Refinement Branch is implemented based on the Pixel2Mesh architecture [22]. It process the initial 3D mesh from the Cubify operation through three main operations: vertex alignment, graph convolution, and vertex refinement. Figure 8 shows how vertex alignment with respect to the original 2D image is achieved by computing a bilinear interpolation for each projected vertex position.

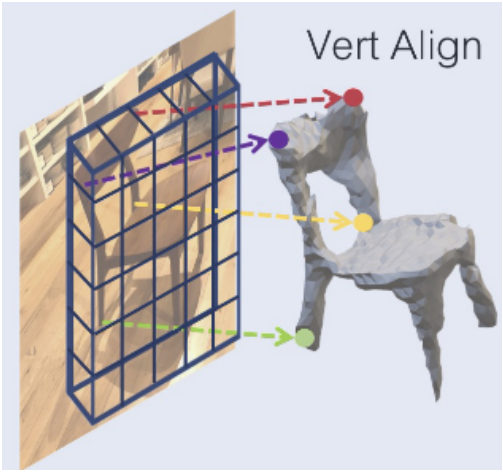

Fig. 8 Visualization of the Vertex Alignment Operation [6]

The triangular mesh can be treated as a computational graph and each vertex's features can be updated via learned weight matrices in a series of graph convolutional layers [23]. The updated weights are then sent as input to the vertex refinement operation, which adds new vertices creating higher resolution 3D mesh on each sequence in the branch.

C. Metrics

The Chamfer and F1 metrics are used to evaluate the difference between ground truth and predicted meshes. We compute both the Chamfer and F1 scores using the same methods in [6]. Let $\Lambda_{P,Q} = \{(\mathrm{p}, \mathrm{argmin}_q ||p - q||) : p \in P\}$ be the set of pairs (p, q) such that each point q is the nearest neighbor of p in Q and let u_p be the unit normal to point p.

The chamfer distance between two pointclouds P, Q is given by:

$$\mathcal{L}_{\text{chamfer}} = |P|^{-1} \sum_{(p,q)\in\Lambda_{P,Q}} ||p - q||^2 + |Q|^{-1} \sum_{(q,p)\in\Lambda_{Q,P}} ||q - p||^2 \tag{1}$$

and the absolute normal distance is given by

$$\mathcal{L}_{\text{normal}} = -|P|^{-1} \sum_{(p,q)\in\Lambda_{P,Q}} |u_p \cdot u_q| - |Q|^{-1} \sum_{(q,p)\in\Lambda_{Q,P}} ||u_q \cdot u_p||^2 \tag{2}$$

and the edge loss, as a shape regularizer, is given by

$$\mathcal{L}_{\text{edge}}(V, E) = \frac{1}{|E|} \sum_{(v,v')\in E} ||v - v'||^2 \tag{3}$$

where $E \subseteq V \times V$ are the edges between each vertex $v \in V$ in the predicted mesh.

Our trained model yielded a Chamfer score of 0.621 and F1 score of 47.51, where a lower score on Chamfer and higher score on F1 are better for each. For comparison, the best scores reported by [6] for a Mesh R-CNN model trained against the Pix3D dataset were 0.306 and 74.84, respectively.

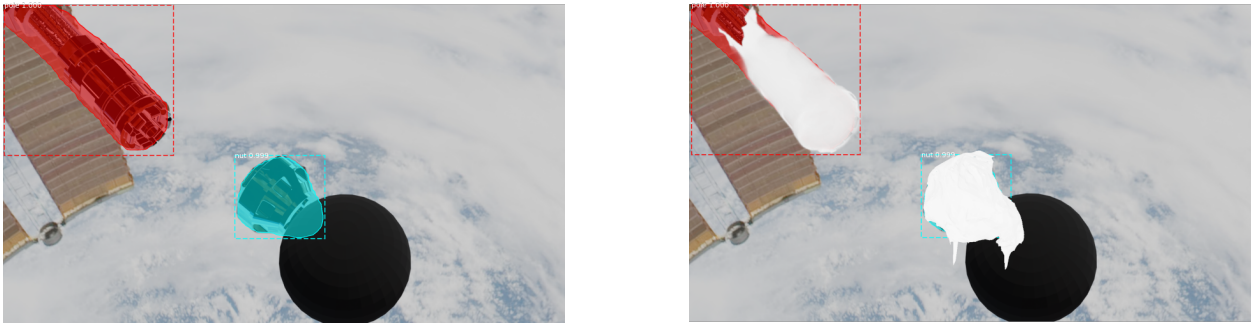

Fig. 9 Example of predicted masks with associated meshes.

Figure 9 shows the typical meshes predicted by the network as currently trained. While mask prediction accuracy is high (98%), the mesh prediction branches yield relatively inaccurate meshes when compared to the model reported in [6]. One reason for this is that the model reported in [6] was trained using eight Tesla V100 GPUs while ours was trained using two Quadro 6000 RTX GPUs. The difference in GPU resources during training requires tuning of various Mesh R-CNN hyperparameters. While we attempted to optimize hyperparameter tuning there may be more optimal configurations, especially considering training on larger training sets.

VII. Conclusion

Enriching data using predictive, generative data without need for extra sensors or multiple views of the same object is highly beneficial in operational environments where mass and energy are at a premium. This study explores the use of neural network based 3D mesh prediction in order to provide data needed for full three-dimensional spatial reasoning. Future work includes further refinement of hyperparameter tuning as well as exploring the use of pixel alignment methods to estimate the pose and orientation of detected objects using an intersection between their segmentation mask and predicted mesh. Further domain randomization may be needed in the case of each object's rendered skin in order to mitigate problems with lighting and reflectivity. Additionally, extending Mesh R-CNN to use generative adversarial network models to generate predictive point clouds in place of the voxel branch would result in 3D mesh predictions with much higher resolution.

Acknowledgments

The authors would like to thank the Convergent Aeronautics Solutions (CAS) Project of the NASA ARMD's Transformative Aeronautics Concepts Program (TACP) for supporting this work via ATTRACTOR.

References

[1] NASA, "NASA Strategic Plan 2018," 2018.

[2] Mukherjee, R., Siegler, N., and Thronson, H., "The Future of Space Astronomy will be Built: Results from the In-Space Astronomical Telescope (iSAT) Assembly Design Study," 2019.

[3] Mars, K., "Gateway," , Aug 2016. URL https://www.nasa.gov/gateway.

[4] Warner, C., "NASA Outlines Lunar Surface Sustainability Concept," , Mar 2020. URL https://www.nasa.gov/feature/nasa-outlines-lunar-surface-sustainability-concept.

[5] He, K., Gkioxari, G., Dollár, P., and Girshick, R., "Mask R-CNN," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988. https://doi.org/10.1109/ICCV.2017.322.

[6] Gkioxari, G., Johnson, J., and Malik, J., "Mesh R-CNN," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9784–9794. https://doi.org/10.1109/ICCV.2019.00988.

[7] Alimo, R., Jeong, D., and Man, K., "Explainable Non-Cooperative Spacecraft Pose Estimation using Convolutional Neural Networks," *AIAA Scitech 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-2096, URL http://dx.doi.org/10.2514/6.2020-2096.

[8] Simonyan, K., and Zisserman, A., "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, Vol. abs/1409.1556, 2015.

[9] Pal, B., Khaiyum, S., and Kumaraswamy, Y. S., "3D point cloud generation from 2D depth camera images using successive triangulation," *2017 International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, 2017, pp. 129–133. https://doi.org/10.1109/ICIMIA.2017.7975586.

[10] Valsesia, D., Fracastoro, G., and Magli, E., "Learning Localized Representations of Point Clouds with Graph-Convolutional Generative Adversarial Networks," *IEEE Transactions on Multimedia*, 2019.

[11] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y., "Generative Adversarial Nets," *Advances in Neural Information Processing Systems*, Vol. 27, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, Curran Associates, Inc., 2014, pp. 2672–2680. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[12] Ramasinghe, S., Khan, S. H., Barnes, N., and Gould, S., "Spectral-GANs for High-Resolution 3D Point-cloud Generation," *CoRR*, Vol. abs/1912.01800, 2019. URL http://arxiv.org/abs/1912.01800.

[13] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, Vol. 25, edited by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Curran Associates, Inc., 2012, pp. 1097–1105. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[14] Jacobi, N., Husbands, P., and Harvey, I., "Noise and the Reality Gap: The Use of Simulation in Evolutionary Robotics," *Proceedings of the Third European Conference on Advances in Artificial Life*, Springer-Verlag, Berlin, Heidelberg, 1995, p. 704–720.

[15] Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., and Abbeel, P., "Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World," *CoRR*, Vol. abs/1703.06907, 2017. URL http://arxiv.org/abs/1703.06907.

[16] He, K., Zhang, X., Ren, S., and Sun, J., "Deep Residual Learning for Image Recognition," *CoRR*, Vol. abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[17] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H., "How transferable are features in deep neural networks?" *CoRR*, Vol. abs/1411.1792, 2014. URL http://arxiv.org/abs/1411.1792.

[18] Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. https://doi.org/10.1109/CVPR.2009.5206848.

[19] Hochreiter, S., "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions," *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, Vol. 6, 1998, pp. 107–116.

[20] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems*, Vol. 28, edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Curran Associates, Inc., 2015, pp. 91–99. URL https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf.

[21] Girshick, R., "Fast R-CNN," *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE Computer Society, USA, 2015, p. 1440–1448. https://doi.org/10.1109/ICCV.2015.169, URL https://doi.org/10.1109/ICCV.2015.169.

[22] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., and Jiang, Y.-G., "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images," *ECCV*, 2018.

[23] Kipf, T., and Welling, M., "Semi-Supervised Classification with Graph Convolutional Networks," *ArXiv*, Vol. abs/1609.02907, 2017.