CS230

SpaceGAN: Generative Adversarial Networks for High Fidelity Simulation of Spacecraft Docking

Sumant Sharma
Ph.D. Student, Stanford University
sharmas@stanford.edu

Connor Beierle *
Ph.D. Student, Stanford University
cbeierle@stanford.edu

Simone D'Amico *
Assistant Professor, Stanford University
damicos@stanford.edu

Abstract

This report describes a image rendering method based on a Generative Adversarial Network (GAN). Given an input synthetic image of a target spacecraft with a certain pose configuration, the proposed method outputs a high fidelity (illumination, texture) of the target spacecraft with the same pose. This work will demonstrate this method by generating two datasets of images of a spacecraft, the first set will be produced using a OpenGL rendering pipeline while the second set will be produced by capturing real images of a mock-up spacecraft placed in a high fidelity illumination environment. However, for the milestone, this work demonstrates this method using two different datasets of synthetic imagery containing different level of texture information about the target spacecraft.

1 Introduction

The on-board determination of the pose, i.e. the relative position and attitude, of a noncooperative client spacecraft using a monocular camera is a key-enabling technology for future on-orbiting servicing and debris removal missions such as ESA's e.Deorbit and PROBA-3 [2], ANGELS by US Air Force [13], PRISMA by OHB Sweden [4]. The existing state-of-the-art on monocular pose determination for spacecraft depend on classical image processing algorithms that identify visible target features[3, 1, 10]. However, these methods are disadvantaged due to the lack of robustness in the presence of adverse illumination conditions and the computational complexity resulting from the evaluation of a large number of possible pose hypotheses. To overcome these disadvantages, authors have proposed the use of deep learning techniques [22, 12, 16, 18, 21, 17, 19, 20, 11]. However, the success of these techniques depends on the availability of datasets of the target spacecraft. Unlike datasets of terrestrial objects such as dogs and cats, datasets of spacecraft are expensive to obtain in massive numbers and often lack the illumination and texture fidelity required to guarantee high navigation accuracy. Robotic facilities such as the one present in Space Rendezvous Laboratory (see fig. 1 can be used to reproduce high fidelity illumination and texture conditions, however, they provide limited functionality due to the constraints of size of the target mock-up spacecraft and the motion of the robotic arm.

Hence, this work proposes the use of a Generative Adversarial Network (GAN) to produce high quantity of high fidelity imagery of a target spacecraft. Such imagery would then allow the reliable training of deep learning techniques, which can output the pose of the same target spacecraft during a formation flying mission.

*not enrolled in CS 230.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

The current work utilizes the pix2pix architecture [9] proposed by Isola et al. In order to train the pix2pix architecture, this work entails the generation of two datasets: 1) synthetic imagery generated using a OpenGL rendering pipeline, and 2) real imagery of a mock-up spacecraft placed in a high fidelity illumination environment at the Space Rendezvous Laboratory at Stanford University. However, in order for the milestone report, the current work is substituting real imagery for another set of synthetic imagery of the target spacecraft that contains a higher level of texture information. Since the capture of real imagery using the robotic facility required the time consuming process of assembling a spacecraft mock-up, the subsitution of datasets was done to reduce the development cycle and quickly verify the implementation of the pix2pix architecture. A copy of the code for the dataset generation, processing of images, and training of the pix2pix architecture is hosted on GitHub.

2 Related work

There has been a wide body of work on Generative Adversarial Networks since the marquee paper by Goodfellow et al. in 2014 [8]. This corpus of literature can be roughly divided into two categories based on their loss. For example, a lot of techniques treat the image-to-image translation problems using an "unstructured" loss [14, 23]. In comparison, conditional GANs instead use a structured loss [6, 7, 15]. Conditional GANs are more suitable to our particular application since we need to respect the pose of the spacecraft in the input and generated image.

3 Dataset Generation

Since training of a GAN architecture typically requires a high number of images (at least a few thousand) and there are no benchmark datasets containing space imagery, this work generated synthetic images of the Tango spacecraft used in the PRISMA mission [5]. Each generated image is constrained to be of 256 pixels by 256 pixels so that it could be directly used as an input to the pix2pix architecture. The camera field of view was selected to be 31.5 degrees, modeling after the close range camera flown aboard the Mango spacecraft of the PRISMA mission. For each synthetic image, we defined the position and attitude of the target body frame with respect to the camera frame. A set of points are selected in the three-dimensional space along the surface of a sphere surrounding the Tango spacecraft. To account for scale, spheres of three different radii are selected.

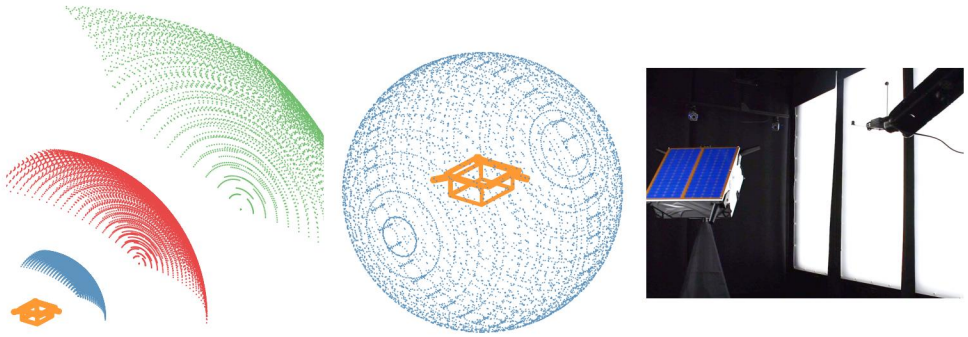

Figure 1: Visualization of the viewpoints used in the synthetic dataset creation. Left: Partial visualization. Middle: Complete visualization. A wireframe model of Tango spacecraft (in orange) is shown here but a complete texture model was used during image rendering. Right: Robotic facility of Space Rendezvous Lab at Stanford with camera mounted on end-effector of a robotic arm pointed at a mock spacecraft.

To show how the points of the three spheres in Figure 1 are selected, let us first define the target object reference frame, $\mathcal{T}$, and the camera reference frame, $\mathcal{C}$, as shown in Figure 2. The relative attitude between the two frames can be described by a set of Euler angles, α, β, γ. Each of these Euler angles are defined between values of 0 and 360 degrees. We discretized this range in 22 linearly equally spaced points. This produced a set of 10648 points in the three dimensional Euler angle space. For a given radius of the sphere shown in Figure 1, these points produce 10648 camera orientations. We

Table 1: Details of the four synthetic datasets generated as part of this work.

Dataset	Texture Information	Resolution	# images
Syn1	High	256 x 256	1250
Syn2	Low	256 x 256	1250
Syn3	Medium	512 x 512	1250
Syn4	High	256 x 256	1250
Syn5	Medium	512 x 512	100
Real1	-	512 x 512	100

then chose the magnitude of the relative position of the target spacecraft w.r.t. the camera, $\left\|{}^{\mathcal{C}}\mathbf{t}\right\|$, (or the radii of the spheres in Figure 1) as 9 meters.

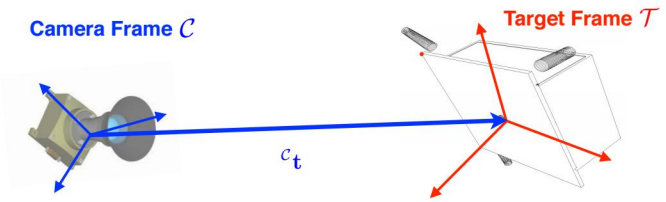

Figure 2: The two reference frames used in synthetic image generation.

Finally, for each of these pose configurations, we produced four synthetic images with varying levels of texture information and pixel resolution. In total, we generated 42592 images. A subset of these were then partitioned into Syn-1, Syn-2, Syn-3, and Syn-4 respectively. Table 1 presebts the details for these datasets. Some sample images from these four datasets are shown in Figure 3.

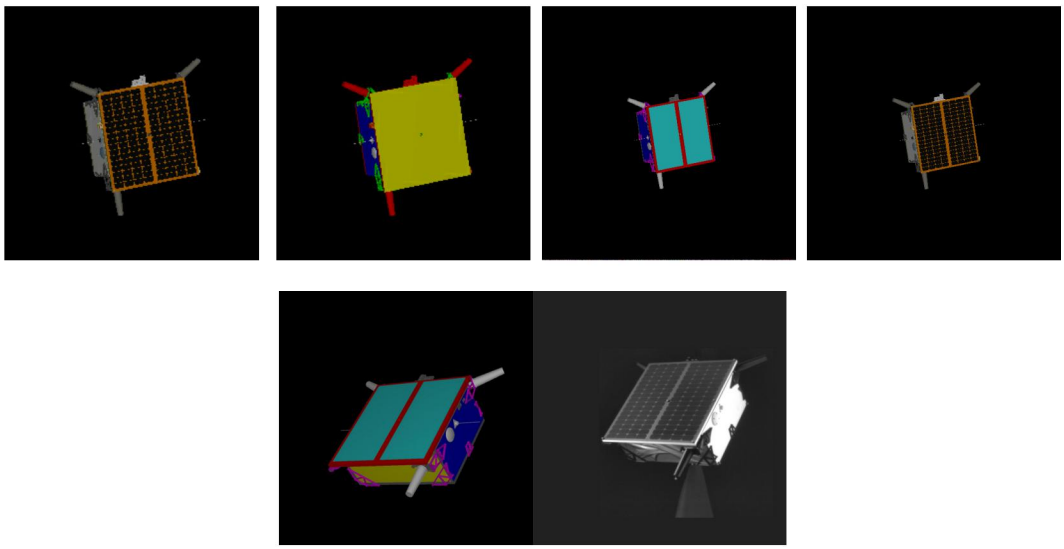

Figure 3: Montage of the front view of the Tango spacecraft as present in the five datasets.

4 Methods

GANs have been explored at depth in the last two-three years ever since the marquee paper by Goodfellow et al. [8]. However, the pix2pix architecture proposed by Isola et al. [9] focused

on "conditional" GANs such that a conditional generative model can be learned from pairs of images to enable image-to-image translation tasks such as the one described in this work. The pix2pix architecture was chosen as the baseline architecture for two reasons. Firstly, their use of a convolutional "PatchGAN" classifer in the descriminator would penalize at the scale of image patches. This would make it relatively straightforward for our application to achieve the desired level of texture fidelity in the output imagery. Secondly, unlike other works such as CyclGAN [24], their architecture uses pair-wise inputs leading to higher fidelity in the output imagery, a luxury that we can afford in our particular application. Both the generator and descriminator architectures of pix2pix use modules of of the form convolution-BatchNorm-Relu. The details of their architecture is provided in the original paper, and hence will not be repeated here for the sake of brevity. Do note that the final objective that is being minimized is:

$$G^* = argmin_G max_D L_{c,GAN}(G, D) + \lambda L_{L1}(G) \tag{1}$$

where $L_{c,GAN}(G, D)$ is the objective function of the conditional GAN and $L_{L1}(G)$ is the L1 distance between the generator output and the ground truth image.

The pix2pix source code was implemented in TensorFlow and largely inspired by the implementation of Christopher Hesse.

5 Experiments/Results/Discussion

Using the baseline pix2pix architecture, we conducted three experiments using the images of the following datasets: experiment 1: Syn 1 $\leftrightarrow$ Syn 2, experiment 2: Syn 3 $\leftrightarrow$ Syn 4, and experiment 3: Syn 5 $\leftrightarrow$ Real1. In particular, the objective of the first and second experiments was to train the generator to output imagery resembling the Syn 2 and Syn 4 datasets, respectively. The objective of the third experiment was to emulate illumination conditions present in the real dataset.

For the first two experiments, we produced 1000 labeled pairs as part of the training set. The other 250 pairs were used as the validation set. We ran the training for 200 epochs with a batch size of 1 (pair). We used the Adam optimizer to minimize the loss presented in equation 1, where the weight λ was chosen to be 100.0. The initial learning rate for Adam was chosen as 0.0002 with the momentum term, β_1 as 0.5. Note that images were 256 x 256 in size for the first experiment while they were 512 x 512 in size for the second experiment. The $L_{L1}(G)$ loss for the two experiments, as reported by tensorboard, is shown in Figure 4. Qualitatively, the results look promising, however, the generator had trouble emulating the solar panel texture correctly due to the presence of repeating features (solar cells).

For the third experiment, we captured a series of 100 images from the robotic facility at the Space Rendezvous Laboratory at Stanford University. For each image, we measured the relative position and orientation of the mock-up spacecraft with respect to the camera mounted on the end-effector of the robotic arm. Since the number of images available for this experiment was small, we used transfer learning. In particular, we resumed the training of the generator and the adversarial networks from the last training checkpoing of experiment 2 using the same hyperparameters as before. To examine the results from the three experiments, we generated images using the trained generator network. Some select results are shown in Figure 5.

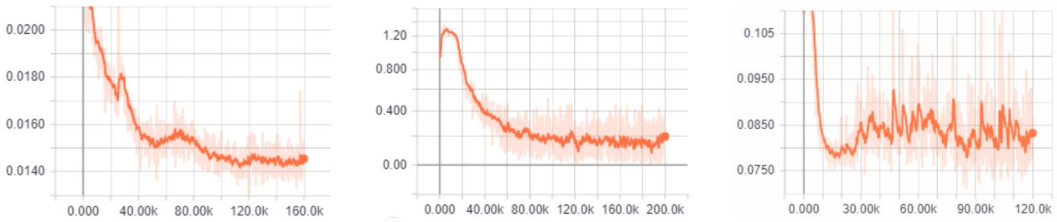

Figure 4: The $L_{L1}(G)$ loss for the three experiments, as reported by tensorboard.

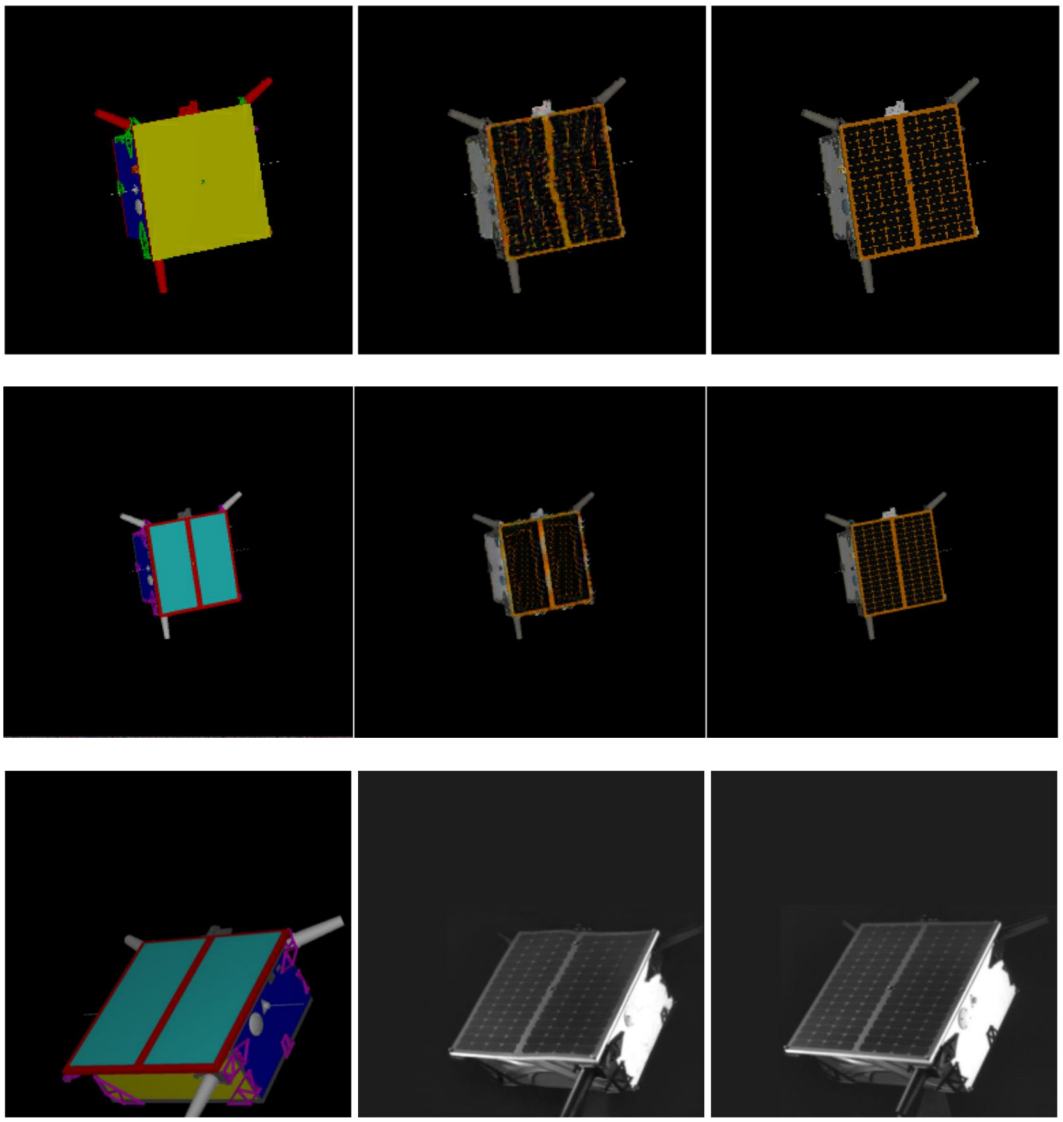

Figure 5: Some results from experiments 1, 2, and 3. Left column shows, the input image to the generator, middle column shows the output, and the right column shows the ground truth.

6 Conclusion/Future Work

This report presents a method to simulate images of a target spacecraft with special emphasis on high fidelity reproduction of the pose and illumination conditions. Such a method can be used to input synthetic images generated using a simple image renderer to produce images with illumination, pose, and texture properties of a real spacecraft. Hence, this method can be used to generate thousands of images of such spacecraft, which can then be used to train algorithms required for spacecraft docking. The baseline architecture adopted in this work, pix2pix, qualitatively had very promising results due to its use of a structured loss during training. In the future, we would like to experiment with a larger dataset of real imagery obtained from the robotic testbed, perform quantiative evaluation of the pose difference between generator output and input images, and experiment with other architectures.

7 Contributions

Sumant Sharma wrote the code to produce the pose configurations for the approximately 40K images generated in this work. He implemented the pix2pix architecture in TensorFlow (inspired by Christopher Hesse). He pre-processed the images before the training, ran the training jobs, and synthesized the results. Connor Beierle wrote the C++/OpenGL code to render the images and programmed the robotic facility.

References

[1] Mathias Benn. *Vision Based Navigation Sensors for Spacecraft Rendezvous and Docking*. PhD thesis, Danish Technical University, 2010.

[2] Lorenzo Tarabini Castellani, Jesús Salvador Llorente, José María Fernández Ibarz, and Mercedes Ruiz. PROBA-3 mission. *International Journal of Space Science and Engineering*, 1(4), 2013.

[3] A Cropp and P Palmer. Pose Estimation and Relative Orbit Determination of a Nearby Target Microsatellite using Passive Imagery. *5th Cranfield Conference on Dynamics and Control of Systems and Structures in Space 2002*, pages 389–395, 2002.

[4] Simone D'Amico, J.-S. Ardaens, and Robin Larsson. Spaceborne Autonomous Formation-Flying Experiment on the PRISMA Mission. *Journal of Guidance, Control, and Dynamics*, 35(3):834–850, 2012.

[5] Simone D'Amico, Per Bodin, Michel Delpech, and Ron Noteborn. PRISMA. In Marco D'Errico, editor, *Distributed Space Missions for Earth System Monitoring*, pages 599–637. 2013.

[6] Emily Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *NIPS*, pages 1–9, 2015.

[7] Jon Gauthier. Conditional generative adversarial nets for convolutional face generation. *Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester 2014*, 5:2, 2014.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. *Advances in Neural Information Processing Systems 27*, pages 2672–2680, 2014.

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. 2016.

[10] Keyvan Kanani, Antoine Petit, Eric Marchand, Thomas Chabot, and Bernard Gerber. Vision Based Navigation for Debris Removal Missions. *63rd International Astronautical Congress*, pages 1–8, 2012.

[11] Alex Kendall, Matthew Grimes, and Roberto Cipolla. PoseNet: A convolutional network for real-time 6-dof camera relocalization. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:2938–2946, 2015.

[12] Alireza Khotanzad and James J H Liou. Recognition and pose estimation of unoccluded three-dimensional objects from a two-dimensional perspective view by banks of neural networks. *IEEE Transactions on Neural Networks*, 7(4):897–906, 1996.

[13] Air Force Research Laboratory. Fact sheet: Automated navigation and guidance experiment for local space (ANGELS), 2013.

[14] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:3431–3440, 2015.

[15] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. 2014.

[16] Antonio Rubio Romano. *3D Pose Estimation Using Convolutional Neural Networks*. PhD thesis, Escola Tècnica Superior d'Enginyeria Industrial de Barcelona, 2015.

[17] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. 2013.

[18] Hao Su, Charles R. Qi, Yangyan Li, and Leonidas J. Guibas. Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. *Proceedings of the IEEE International Conference on Computer Vision*, 11-18-Dece:2686–2694, 2016.

[19] Jonathan Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. pages 1–9, 2014.

[20] Alexander Toshev and Christian Szegedy. DeepPose: Human pose estimation via deep neural networks. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1653—-1660, 2014.

[21] M W Wright and F Fallside. Object Pose Estimation by Neural Network. In *International Conference on Image Processing and its Applications*, pages 602–603, Maastricht, Netherlands, 1992. IET.

[22] P. Wunsch, S. Winkler, and G. Hirzinger. Real-Time Pose Estimation of 3-D Objects from Camera Images Using Neural Networks Institute for Robotics and System Dynamics. In *International Conference on Robotics and Automation*, pages 3232–3237, Albuquerque, New Mexico, 1997.

[23] Saining Xie and Zhuowen Tu. Holistically-Nested Edge Detection. *International Journal of Computer Vision*, 125(1-3):3–18, 2017.

[24] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. 2017.