CS230

Robust Learning-based Pose Estimation of Noncooperative Spacecraft

Tae Ha (Jeff) Park
Space Rendezvous Laboratory
Department of Aeronautics and Astronautics
Stanford University
tpark94@stanford.edu

1 Introduction

The ability to accurately determine and track the pose of a noncooperative spacecraft is becoming ever more demanding for current and future on-orbit servicing and debris removal missions such as RemoveDEBRIS mission by Surrey Space Centre (1), Phoenix program by DARPA (2), and Restore-L mission by NASA (3). Performing a pose estimation based solely on a monocular camera is especially attractive due to its low power and mass requirements. Current state-of-the-art monocular-based methods (4; 5) resort to classical image processing techniques which tend to lack robustness in spaceborne applications because images taken in space are characterized by low signal-to-noise ratio and adverse illumination conditions. Recent works from Stanford Space Rendezvous Laboratory (SLAB) propose a technique based on Convolutional Neural Network (CNN) that frames the pose estimation as a classification problem by discretizing the pose space (6; 7). Specifically, (7) first discretizes the spacecraft attitude space and uses region-proposal network (8) to regress the region-of-interst and predict the attitude class. While they have presented the potential of using CNN as a more robust mechanism for spacecraft pose estimation, the works are limited to synthetic spacecraft images rendered based on 3D models. In reality, the CNNs must be robust to images of the same target from different sources and environment, which tend to vary in spacecraft textures, surface properties, illumination conditions, and so on. Therefore, this project extends the works done in (6; 7) by framing the pose estimation as a bounding box regression problem while also exploring a training method that can improve the network's robustness to images from different distributions.

The general problem statement is to determine the relative attitude and position of the camera frame, $\mathcal{C}$, with respect to the target's body frame, $\mathcal{B}$. The relative position and attitude are respectively represented by a position vector, $\mathbf{t}_{\text{BC}}$, from the origin of $\mathcal{C}$ to the origin of $\mathcal{B}$, and a quaternion, $\mathbf{q}_{\text{BC}}$, which aligns the reference frame $\mathcal{B}$ with $\mathcal{C}$. Figure 1 graphically illustrates these reference frames and variables.

2 Related Works

The general learning-based pose estimation approaches can be divided into three categories depending on the output of CNN – classification, pose regression, and bounding box regression. Pose classification first discretizes the pose or attitude space into a discrete number of bins and train the network to predict the pose class (9; 6; 7; 10; 11). Given that classification is a well-posed problem in machine learning, this approach is advantageous since it is relatively easy to re-use the state-of-the-art classification networks. However, the network performance depends on how well the pose space is discretized, and the approach generally requires post-refinement process using the crude predicted pose class.

CS230: Deep Learning, Winter 2019, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

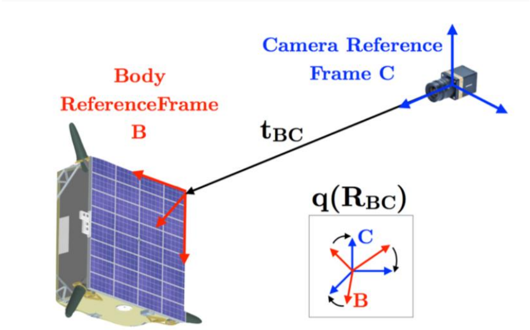


Figure 1: Definition of the reference frames, relative position, and relative orientaiton.

Pose regression, on the other hand, attempts to directly regress the 6D pose information – position vector and relative quaternion or SE(3) transform – using a single or multi-staged CNN architecture. Current state-of-the-art in this approach is PoseCNN (12) which outputs semantic labels and 6D pose as direct results of forward propagation. PoseCNN shows great performance by decoupling 3D translation and 3D rotation estimation and introducing novel loss functions. However, even PoseCNN has limited accuracy by itself and is augmented using a diverse post-refinement procedures, such as Iterative Cloud Point (ICP) with depth information or iterative model matching architecture such as DeepIM (13).

Another approach is to regress the eight corners of a 3D bounding box around the target (14; 15). Then, given the predicted 2D corners of a bounding box and a known 3D model of the target, Perspective-n-Point (PnP) problem can be solved (16) to extract the rotation matrix and position vector. In this approach, camera intrinsic is decoupled from CNN, since 2D-3D transformation is performed *after* CNN prediction. Therefore, if the network is trained to extract the corners based on the shape of the object, the network theoretically need not be re-trained if a servicing spacecraft is equipped with different camera.

3 Dataset

One of the main difficulties of training a neural network for pose estimation is lack of target images with annotated pose labels. Especially in the context of *spacecraft* pose estimation, it is extremely challenging to even obtain an image of a spacecraft on-orbit using the camera on another spacecraft, let alone annotate the pose. Fortunately, SLAB has recently developed the Spacecraft PosE Estimation Dataset (SPEED) (7) which is capable of generating synthetic images of spacecraft using MATLAB and OpenGL with desired orientation and position from a camera with known instrinsics. Moreover, by specifying the position of Earth and Sun, it can also simulate a realistic illumination condition due to Earth albedo and sunlight. These synthetic images generated from SPEED can then be used to train and evaluate any pose estimation network. In this project, SPEED is used to generate images of Tango spacecraft from PRISMA mission (17) with desired relative position and attitude with respect to the camera (Fig. 3(a)). The camera model is identical to the one used in Mango spacecraft, which captured the images of Tango during its proximity operation phase.

This paper also uses PRISMA-21 dataset, which includes 21 images of Tango spacecraft during PRISMA mission (17) taken by Mango spacecraft (Fig. 3(b,c)). These 21 images have hand-labeled pose information, giving us the ability to test the performance of a network on real on-orbit images.

4 Methods

In this project, the CNN is trained to regress the bounding box corners which can then be used to extract pose information by solving the PnP. This method is preferred over other approaches due to the fact that the camera property is decoupled from the CNN, potentially preventing the need of re-training the entire network if different cameras are to be used. Moreover, the predicted bounding box corners can be regarded as the *measured features* that can be tracked using the standard state estimation filters on-board the spacecraft.

4.1 Pose Estimation Network Architecture

The architecture used in this project is developed by Tekin et al. (15), who takes YOLOv2 (18) as its backbone. Figure 2 visualizes the network structure. By making the modification such that each final grid detects normalized (x, y) coordinates of eight 3D bounding box corners and a centroid, this network is able to efficiently regress bounding box corners in a single-shot manner. Tekin et al. also adds a pass-through layer to leverage features detected from an earlier layer for improved performance. The network has an overall stride factor of 32. It takes $(416 \times 416 \times 3)$ RGB input and reduces it to $(13 \times 13 \times (19 + C))$, where C is the number of detectable classes, and the dimension size of 19 includes eight 2D coordinates of bounding box corners, a centroid, and objectness score. Since this project assumes a single target in image frame, the final dimension is $(13 \times 13 \times 20)$, and the class index is left to be zero. Just as in YOLOv2, the bounding box corners from the grid with the highest objectness score are taken as the predicted final bounding box.

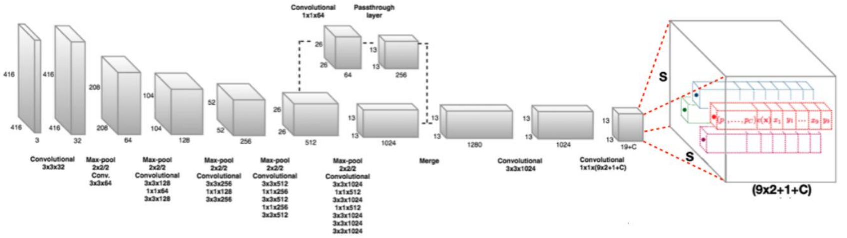

Convolutional 1x1x64
Passthrough layer
416
416
3
416
416
32
208
208
64
104
104
128
52
52
256
26
26
512
26
26
64
13
13
256
13
13
1024
13
13
1280
13
13
1024
13
13
19+C
S
S
(9x2+1+C)
Convolutional 3x3x32
Max-pool 2x2/2 Conv. 3x3x64
Max-pool 2x2/2 Convolutional 3x3x128 1x1x64 3x3x128
Max-pool 2x2/2 Convolutional 3x3x256 1x1x128 3x3x256
Max-pool 2x2/2 Convolutional 3x3x512 1x1x256 3x3x512 1x1x256 3x3x512
Max-pool 2x2/2 Convolutional 3x3x1024 1x1x512 3x3x1024 1x1x512 3x3x1024 3x3x1024 3x3x1024
Merge
Convolutional 3x3x1024
Convolutional 1x1x(9x2+1+C)

Figure 2: The network structure of Tekin et al. (15)

4.2 Texture Randomization

Inspired by (19), the neural style transfer algorithm by Jackson et al. (20) is used to efficiently generate texture-randomized dataset of synthetic spacecraft images offline. The idea is to randomize the target texture such that the network is forced to leverage the features that are invariant to the local texture, i.e. the global shape of the target.

The novelty of (20) is the embedding of the style into a vector. During training, the authors keeps track of the mean (μ) and covariance (Σ) of the style embeddings for the entire dataset of style images. Then, at testing, they propose to simply sample the random style from the multivariate normal distribution parametrized by (μ, Σ). The randomly sampled style is then interpolated with the content style via Eq. (1),

$$\boldsymbol{z} = \alpha \mathcal{N}(\mu, \Sigma) + (1 - \alpha) P(\boldsymbol{c}) \tag{1}$$

where α is the parameter determining the degree of interpolation. Naturally, higher α leads to more intense variation of the texture.

4.3 Training

Using SPEED, 9,600 images of Tango spacecraft from PRISMA mission are initially generated. The spacecraft in each image has random attitude, but they have a fixed relative position such that the centroid of the spacecraft is aligned at the image center with constant distance of 10 meters. For each image, the black-and-white masks are created as well to enable accurate cropping of the image. Then, a separate style-transfered dataset is created by applying (20) with $\alpha = 0.25$ then cropping with the provided mask in order to prevent potential distortion of the shape.

Then, at the training stage, either style-transfered or original images are selected based on a hand-tuned probability. For this instance of training, style-transfered images are selected with the probability of p = 0.75 to allow for more variation in spacecraft texture. Then, the chosen image is randomly translated and scaled. Afterwards, randomly cropped Earth image from Himawari-8 Earth imagery[1] is inserted in the background with probability of p = 0.5. Lastly, if the training image is original, synthetic

[1] https://himawari8.nict.go.jp/

image, random Gaussian noise is applied with p = 0.5. If the training image is style-transfered one, then a random occlusion (dark rectangle) is applied with p = 0.5 to mimic the shadowing effect due to sunlight.

Out of 9,600 images, 80% are chosen as training dataset. The network is trained with the batch size of 48 using Adam optimizer. The momentum for the optimizer is chosen as $\beta_1 = 0.5$, and a constant learning rate of 1×10^{-4} is employed. For testing, only random translation and scaling are applied to the original synthetic images. The codebase for this project is written using PyTorch v1.0.0.

4.4 Metrics

The following three metrics are used to evaluate the accuracy of predicted bounding box corners.

- **Mean Pixel Error (MPE)** is computed by summing the magnitude of predicted and ground-truth bounding box corners and centroids scaled by the original image size (w, h), i.e.

$$\text{MPE} = \sum_{i=0}^{8} w||x_i - \tilde{x}_i||_2 + h||y_i - \tilde{y}_i||_2$$

- **Translation Error ($E_{\mathbf{T}}$)** is computed by taking the magnitude of the difference between the ground-truth position vector ($\mathbf{t}_{\text{BC}}$) and the predicted position vector ($\tilde{\mathbf{t}}_{\text{BC}}$) via solving PnP with 3D model and predicted bounding box corners, i.e.

$$E_{\text{T}} = |\mathbf{t}_{\text{BC}} - \tilde{\mathbf{t}}_{\text{BC}}|$$

- **Rotation Error ($E_{\mathbf{R}}$)** Is computed via the following:

$$E_{\text{R}} = \arccos \frac{\text{tr}(\mathbf{R}_{\text{BC}}\tilde{\mathbf{R}}_{\text{BC}}^{\top}) - 1}{2}$$

where $\boldsymbol{R}_{\text{BC}}$ is the ground-truth direction cosine matrix from $\mathcal{C}$ to $\mathcal{B}$, and $\tilde{\mathbf{R}}_{\text{BC}}$ is the predicted rotation matrix obtained by solving the same PnP problem.

5 Results

5.1 Training without style transfer

Table 5.1 shows the results of the network's performance when the style transfer is *not* applied. We see the network trained on synthetic spacecraft images perform extremely well on synthetic testing images as well with mean angular error on the order of 2° and millimeter-level position error along the image plane. Note that the biggest source of position is along Z-axis, i.e. the direction of camera boresight, which is expected as any change in box dimension due to pixel error results in bigger depth change when the spacecraft is farther away from the camera. See Fig. 3(a). The same model, however, performs poorly on real images. Interestingly, the network outputs a box-shaped prediction; however, the box is either in completely off attitude and often away from the region of interest (figures are included in the poster).

Metrics	SPEED	PRISMA-21
MPE [pix]	3.615	113.5
Mean E_{T} [m]	[0.009, 0.008, 0.188]	[0.366, 0.138, 1.715]
Median E_{T} [m]	[0.006, 0.005, 0.101]	[0.153, 0.074, 1.665]
Mean E_{R} [deg]	2.202	61.396
Median E_{R} [deg]	1.818	22.395

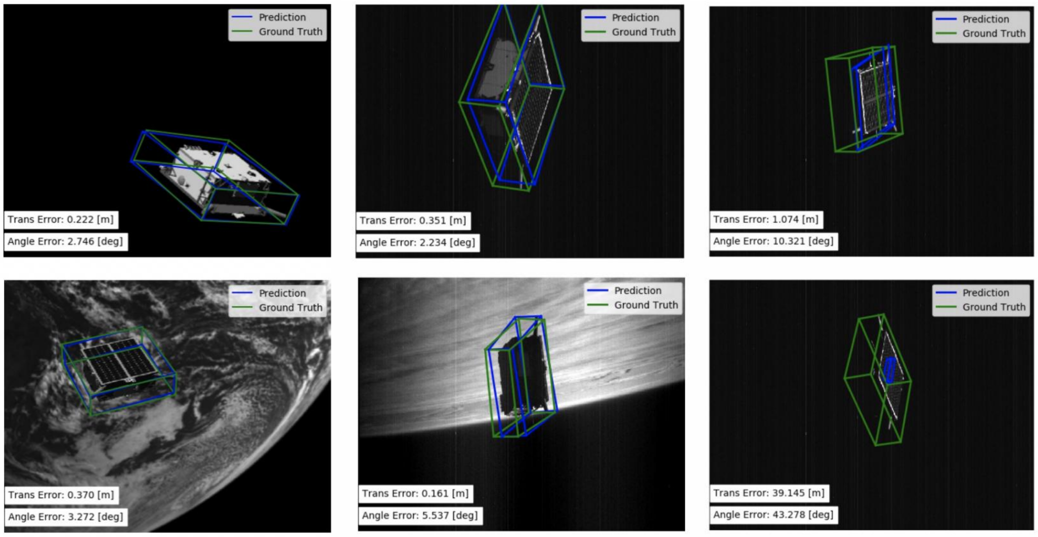
Table 1: Testing results on SPEED test set and PRISMA-21 images *without* style transfer.

5.2 Training with style-transfer

Table 5.2 lists the results on SPEED and PRISMA-21 images when style transfer is performed to the dataset with $\alpha = 0.25$. First note the overall improvement on PRISMA-21 dataset in all metrics except mean position error. However, significant improvements in overall mean pixel, rotation error, and median position error suggest the network is able to perform bounding box regression without having been trained on them (Fig. 3(b)). There are, of course, a number of failure cases. It turns out the network fails at bounding box regression when the shadowing due to the illumination is so severe such that the image only contains partial shape of the target, as shown in Fig. 3(c). Since the network is supposedly focusing on the global shape rather than local texture, it is reasonable that the network fails at regression when the shape is partially missing. This also suggests that random occlusion at testing is not a solution to this problem.

Metrics	SPEED	PRISMA-21
MPE [pix]	6.291	25.678
Mean E_{T} [m]	[0.024, 0.023, 0.655]	[0.106, 0.160, 3.972]
Median E_{T} [m]	[0.011, 0.009, 0.216]	[0.028, 0.030, 0.704]
Mean E_{R} [deg]	4.907	11.334
Median E_{R} [deg]	3.330	6.273

Table 2: Testing results on SPEED test set and PRISMA-21 images *with* style transfer.



(a) Synthetically generated images via SPEED

(b) Real images from PRISMA mission

(c) Failure cases on real images

Figure 3: Comparison of network performance. (a) is using the network trained with vanilla SPEED images. (b) and (c) are using the network trained with style-transfered images.

6 Conclusion

This project employed the technique of bounding box regression to perform pose estimation on spacecraft images. It also incorporated the idea of texture randomization via neural style transfer such that the network is forced to ignore the texture of the target, which happens to be one of the major differences between synthetic and real images from on-orbit missions. In the future, I would like to explore different techniques that can be added to improve robustness against occlusion. I would also like to test its performance against the images taken with different cameras.

7 Notes

This project is part of my on-going research in Space Rendezvous Laboratory. The codes are available at `https://github.com/tpark94/SLAB_ModelFreePoseEst.git` under `develop` branch.

References

[1] J. L. Forshaw, G. S. Aglietti, N. Navarathinam, H. Kadhem, T. Salmon, A. Pisseloup, E. Joffre, T. Chabot, I. Retat, R. Axthelm, and et al., "Removedebris: An in-orbit active debris removal demonstration mission," *Acta Astronautica*, vol. 127, p. 448–463, 2016.

[2] B. Sullivan, D. Barnhart, L. Hill, P. Oppenheimer, B. L. Benedict, G. V. Ommering, L. Chappell, J. Ratti, and P. Will, "Darpa phoenix payload orbital delivery system (pods): "fedex to geo"," *AIAA SPACE 2013 Conference and Exposition*, 2013.

[3] B. B. Reed, R. C. Smith, B. J. Naasz, J. F. Pellegrino, and C. E. Bacon, "The restore-l servicing mission," *Aiaa Space 2016*, 2016.

[4] S. Sharma, J. Ventura, and S. D'Amico, "Robust Model-Based Monocular Pose Initialization for Noncooperative Spacecraft Rendezvous," *Journal of Spacecraft and Rockets*, p. 1–16, 2018.

[5] V. Capuano, K. Kim, J. Hu, A. Harvard, and S.-J. Chung, "Monocular-Based Pose Determination of Uncooperative Known and Unknown Space Objects," *69th International Astronautical Congress (IAC)*, 2018.

[6] S. Sharma, C. Beierle, and S. D'Amico, "Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks," in *2018 IEEE Aerospace Conference*, pp. 1–12, March 2018.

[7] S. Sharma and S. D'Amico, "Pose estimation for non-cooperative rendezvous using neural networks," in *2019 AAS/AIAA Astrodynamics Specialist Conference, Ka'anapali, Maui, HI*, January 13-17 2019.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 91–99, 2015.

[9] S. Tulsiani and J. Malik, "Viewpoints and keypoints," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[10] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "SSD-6D: Making RGB-Based 3D Detection and 6D Pose Estimation Great Again," *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[11] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, "Implicit 3d orientation learning for 6d object detection from rgb images," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[12] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes," *Robotics: Science and Systems XIV*, 2018.

[13] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: Deep Iterative Matching for 6D Pose Estimation," *Computer Vision – ECCV 2018 Lecture Notes in Computer Science*, p. 695–711, 2018.

[14] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects," *CoRR abs/1809.10790*, 2018.

[15] B. Tekin, S. N. Sinha, and P. Fua, "Real-Time Seamless Single Shot 6D Object Pose Prediction," *CVPR*, 2018.

[16] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An Accurate O(n) Solution to the PnP Problem," *International Journal of Computer Vision*, vol. 81, no. 2, p. 155–166, 2008.

[17] S. D'Amico, P. Bodin, M. Delpech, and R. Noteborn, *PRISMA*, p. 599–637. Springer, 2013.

[18] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.," in *International Conference on Learning Representations*, 2019.

[20] P. T. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, "Style augmentation: Data augmentation via style randomization," 2018.