Journal of Aerospace Information Systems
Vol. 17, No. 11, November 2020

Long-Range Pose Estimation for Aerial Refueling Approaches Using Deep Neural Networks

Andrew Lee,* Will Dallmann,† Scott Nykl,‡ Clark Taylor,‡ and Brett Borghetti§
Air Force Institute of Technology, Wright–Patterson Air Force Base, Ohio 45433

https://doi.org/10.2514/1.I010842

Automated aerial refueling (AAR) provides unique challenges for computer vision systems. Aerial refueling maneuvers require high-precision low-variance pose estimates. The performance of two stereoscopic (stereo) vision systems is quantified in ground tests specially designed to mimic AAR. In this experiment, three-dimensional (3-D) pose-estimation errors of 6 cm on a target 30 m from the current vision system are achieved. Next, a novel computer vision pipeline to efficiently generate a 3-D point cloud of the target object using stereo vision that leverages a convolutional neural network (CNN) is proposed. Using the proposed approach, a high-fidelity 3-D point cloud with ultra-high-resolution imagery 11.3 times faster than previous approaches can be generated.

I. Introduction

AUTOMATED aerial refueling (AAR) is an important emerging technology for the U.S. Air Force. Currently, remotely piloted aircraft (RPAs) cannot perform in-air refueling maneuvers due to the multisecond latency between the pilot and the remote aircraft. An AAR system bypasses this limitation by allowing the tanker to automatically control the receiver during the refueling process. The ability to refuel RPAs in flight would provide a valuable increase in operational endurance for their mission set. Additionally, an AAR system serves as an important intermediate step to a fully autonomous tanker. An AAR system needs the capability to control the tanker, its refueling boom, and the receiver for the duration of the aerial refueling procedure. Before control logic can be safely implemented, the system must have a high-precision relative pose-estimation process that tracks the receiver in real time. In this paper, we implement a vision system that achieves such requirements.

The greatest challenge for an AAR computer vision system is the long range from the cameras on a tanker to the refueling contact point. The contact point is approximately 30 m from the stereoscopic (stereo) cameras. At contact, the three-dimensional (3-D) pose-estimation error must be less than 10 cm. Due to aerial refueling mission constraints, we cannot modify existing aircraft or use Global Positioning System (GPS) signals to augment a vision-based approach. Parsons et al. have previously proposed a method for solving this problem using a stereo vision system [1]. Here, we seek to validate and improve that system.

One method for improving the accuracy of pose estimates is to increase the stereo image resolution. Unfortunately, an increase in pixel count leads to a proportional increase in processing time. To mitigate this limitation, we draw inspiration from nature. The neural structure dedicated to processing imagery is too small to support high-resolution sensory over a wide field of view. Therefore, the biological eyes in predators have small ultra-high-resolution foveae and a low-resolution periphery. This natural ultra-high-resolution stereo vision system allows precise ranging and pose estimation in predators' brains without requiring more neural processing power. To replicate this nature-based approach, we use a convolutional neural network (CNN) to localize the receiver in each image. Then, the more computationally expensive image processing steps need only be applied to the region containing the receiver.

In this paper, we contribute the following:

1) We demonstrate a novel deep learning approach to speed up stereo vision-based point-cloud generation by 11.3 times in 4 K+ image pairs.

2) We perform pose estimation with an error of 6 cm at a target 30 m away from the camera system.

3) We validate the system with ground experiments that accurately replicate an aerial refueling scenario.

Section II explores the background and related work. Section III outlines the design and execution of a ground experiment specifically designed to evaluate vision systems for AAR. In Sec. IV, we demonstrate how to implement a deep learning augmentation to the stereo vision pipeline. Finally, Sec. V discusses the implications of our results and how they can guide future work. A video of this work is available on YouTube [2].

II. Background and Theory

This section is composed of four main subsections. Section II.A explains current aerial refueling techniques and limitations. Section II.B discusses existing computer vision pipelines for pose estimation. Section II.C calculates the expected benefits of using higher-resolution stereo cameras. Section II.D examines how the field of deep learning has led to improvements in machine vision.

A. Aerial Refueling

Aerial refueling is the process of transferring fuel from one aircraft to another in flight. A *tanker* aircraft transfers fuel to a *receiver* aircraft. There are two primary ways to perform aerial refueling: the boom method and the probe-and-drogue method. The boom method requires stricter tolerances and enables faster fuel-transfer rates than the probe and drogue method. A robust automated solution for the boom method will provide tolerances tight enough to support the probe and drogue method; the converse is not necessarily true. As a result, this research focuses on the boom method. Using this method, the receiver approaches the tanker from behind and below. The approach maneuver begins when the receiver is approximately $1/2$ km from the tanker. The approach ends when the receiver reaches the contact position. For this research, we approximate contact to be 30 m from the tanker's camera system. At the contact position, a boom operator inserts the boom into a receptacle on the receiver and begins refueling. The space the receiver may occupy during refueling is called the refueling envelope. The following constraints guide this research:

1) AAR cannot depend on precision GPS information because the GPS is jammable and spoofable.

2) Receiver aircraft cannot be modified except for computation and control hardware necessary to allow the tanker to automate the refueling approach. While adding sensors and markers to the receiver

Received 12 March 2020; revision received 19 May 2020; accepted for publication 20 May 2020; published online Open Access 24 June 2020. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 2327-3097 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Second Lieutenant U.S. Air Force, AFIT/ENG, 2950 Hobson Way.

†Captain U.S. Air Force, AFIT/ENG, 2950 Hobson Way.

‡Assistant Professor, Computer Science, AFIT/ENG, 2950 Hobson Way.

§Associate Professor, Computer Science, AFIT/ENG, 2950 Hobson Way.

would provide better results, these solutions are impractical and could adversely affect receiver performance.

3) The navigation solution must run in real time. A system that takes too long to calculate its navigation solution cannot safely control the refueling procedure.

4) To safely control the receiver, the tanker must find the exact relative position and orientation of the receiver. Inaccurate pose estimation risks putting the receiver on an incorrect flight path. This could lead to failure to connect and damage to both aircraft.

5) The refueling contact point is 30 m from the stereo vision system.

6) The relative 3-D pose estimate should have an error less than 10 cm at the contact point.

Several efforts have been made to perform AAR. One effort to accomplish pose estimation added markers to the tanker and receiver, using geometric algorithms to calculate pose [3]. Another approach used precision GPS and an inertial measurement unit (IMU) with Kalman Filtering to estimate pose [4]. These solutions do not meet mission constraints outlined previously. Specifically, this work seeks to create a real-time, vision-only pose-estimation process that has a mean 3-D pose-estimation error of less than 10 cm at 30 m. A vision-only approach provides a more robust solution for operational environments; visual markers may be difficult to maintain or impractical for an aircraft's missions, and integrating new sensors is likely cost-prohibitive.

B. Six-Degree-of-Freedom Pose Estimation

One fundamental problem for computer vision systems is to derive a 3-D model of the environment from 2D images. In monocular vision, information comes from a single camera or image. Zhang, Jiang, and Zhang [5] demonstrate a deep learning process that performs object detection and pose estimation from a single camera. While their approach does run in real time, their neural network performs pose estimation on small objects that are very close to the camera. Similarly, Ferrara, Piva, Argenti, Kusuno, Niccolini, Ragaglia, and Uccheddu [6] use monocular and stereo systems to perform pose registration at ranges form 0.5 m–4.0 m. These approaches are not adequate solutions, because our problem requires a high precision solution for a very distant, large object.

Early AAR pose-estimation efforts focused on using monocular vision as a single component of a sensor-fusion relative navigation solution [3,7]. Since both of these added markers to the aircraft, they do not meet the first or second constraints outlined in Sec. II.A.

Stereo vision finds features in images and, after a calibration, undistortion, and rectification process, reprojects these features into 3-D space relative to the cameras using epipolar geometry (for more information, see Ref. [8]). The stereo block matching algorithm locates features in both images and calculates the disparity, or distance in pixel space, between them. Once disparities have been calculated for an image pair, the disparity map can be reprojected into space to create a 3-D point cloud. Stereo block matching requires a series of pixelwise comparisons. Increasing the number of pixels in the image pair leads to a linear increase in computation time. Our real-time constraint imposes a limit on the resolution that can be used in for this process. Once a point cloud has been generated, there are many techniques to perform pose estimation. Since aircraft are rigid bodies, the point-to-point iterative closest point (ICP) [9] was chosen for this work. However, alternate methods such as parallel ICP [10], fast global registration [11], or a deep learning approach [12] may provide different performance and precision tradeoffs.

C. Camera Resolution and Depth Estimation

Intuitively, one expects that a higher-resolution camera pair would improve depth estimation fidelity. The expected error for a stereo camera system at a given range can be calculated using [13,14]

$$\epsilon_z = \frac{z^2}{bf}\epsilon_d \tag{1}$$

where ϵ_z is the depth error, z is the depth, b is the baseline, f is the focal length (in pixels), and ϵ_d is the matching error in pixels (disparity values, which are assumed to be one). In this subsection, we simulate the error in depth reprojection for a single point using extensive open-source computer vision library (OpenCV). With properly calibrated cameras, it is possible to have a mean depth estimation error near zero at long ranges; however, the mean error is often misleading because individual depth estimations may significantly overestimate or underestimate an individual feature's depth. As outlined in Sec. II.B, the entire point cloud contributes to the accuracy of pose estimation. For this reason, we seek to decrease mean absolute error (MAE). To demonstrate the necessity for a using higher-resolution camera, we used the scenario demonstrated in Fig. 1, where the point being triangulated was 30 m away from a stereo camera system employing a $1/2$ m stereo baseline.

Using cameras with a fixed, 56 deg field of view, the camera resolution was varied and compared the average error in depth estimation as a function of distance from the camera baseline. With Gaussian noise and a 1 pixel standard deviation in both images, a 1280×960 image resulted in a 0.4598 m MAE in distance from the cameras. By using a higher-resolution camera of 4896×3264, a 0.38 m MAE is achieved. This demonstrates the potential for significantly improving the relative pose computation of a stereo vision system by increasing the resolution of the cameras. These results correspond well to the calculated error as shown in Table 1.

D. Deep Learning in Computer Vision

Deep learning provides different tradeoffs and benefits for real-time computer vision than conventional methods. CNNs require training time and labeled training datasets. However, online they execute in a short, constant time for each input. As Ref. [15] discusses, using deep learning with conventional techniques may provide robust solutions.

Deep learning techniques have made substantial progress in recent years toward recognizing various objects in images. Since 2010, error rates in detecting objects in an image of a large dataset have decreased from over 20% down to 1%. Architectures vary from a sequential series of convolution filters all the way to dense connections where each layer takes in the output of every previous layer as input [16,17]. Additionally, many newer CNNs have demonstrated an ability to localize the objects they identify in an image [18–20]. There have been experiments that show deep learning can outperform traditional methods [21]; however, conventional computer vision algorithms' intermediate representations, such as point clouds, have been shown to improve performance of deep learning solutions compared to image-only networks [22,23].

In this work, a deep CNN is trained and tested using simulated imagery. Recent literature suggests that a network capable of performing well on the simulated imagery could be trained to perform as well on imagery from physical cameras. The work in Refs. [24,25] suggests that the main issue is that the virtual camera is a different sensor from a physical camera, and domain adaptation is required

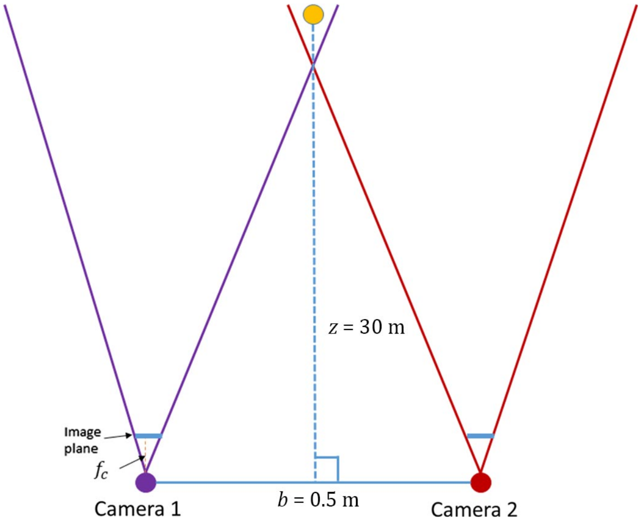


Fig. 1 scenario for single-point depth estimation.

when changing sensors. Goyal et al. [26] also show that augmenting semantic segmentation image datasets with synthetic imagery can improve results. Ros et al. [27] demonstrate the benefits of this approach. Based on these findings, we assume the CNN we trained and evaluated on virtual imagery will perform just as well when trained and evaluated on physical imagery.

III. Ground Experiment Design and Results

This section explains the experiments that were conducted to validate our previously proposed computer vision pipeline. The base pipeline comes from Parsons et al.'s previous work [1] and functions as follows:

1) Capture stereo imagery.
2) Generate a disparity map.
3) Convert the disparity map into a 3-D point cloud.
4) Use ICP to register the receiver's pose.

In this section, we examine the affects of camera resolution on long-range pose estimation using a ground experiment designed to mimic an aerial refueling approach. Ongoing work suggests that this experiment's residual errors at a given range closely reflect a real test flight's residual errors at the same range [28]. Section III.A describes the ground experiment we designed to mimic an aerial refueling approach. Next, Sec. III.B analyzes the experiment's results and their importance to future AAR work.

A. Experiment Design

The ground test compared the high-resolution cameras with lower-resolution cameras mimicking an aerial refueling approach as much as possible. The experiment was run in two parts: the first with lower-resolution electrooptical (EO) and infrared (IR) cameras, and the second with high-resolution EO cameras.

1. *Stereo Camera System*

Two separate stereo vision systems comprising two pairs of stereo EO cameras and one pair of IR cameras were employed. Using both EO and IR cameras increases the diversity of the experiments and gives additional data collection sources. The use of IR cameras provides the opportunity to validate stereo IR cameras as a viable option for stereo vision in the AAR domain.

Allied Vision Proscilica GT1290C EO cameras were chosen for the low-resolution EO stereo vision system. The GT1290Cs capture 24 bit red, green, blue (RGB) images at a resolution of 1280 × 960 and have adjustable focal points and apertures. The adjustable focal point has the advantage of setting the focus to infinity to maximize image clarity for objects at long distances, since a receiver in the experiments is at a contact distance of about 30 m. Additionally, the cameras do not autofocus, which interferes with the camera calibration. For the high-resolution cameras, Allied Vision Prosilica GT4905C EO cameras were chosen. These have a compatible application programming interface (API) with the GT1290C, allowing for the same configuration, except for the resolution. The cameras are capable of a full resolution of 4896 × 3264; however, to achieve 10 Hz frame rates, the high-resolution cameras were configured to capture images at a resolution of 2448 × 1632 in a smaller field of view; they maintain 4 K + pixel density if extended to the full field of view. The IR cameras chosen for the project had an image resolution of 1024 × 768, and the images produced are 16 bit grayscale images. Like the EO cameras, the IR cameras were also focused to infinity. All three systems have similar full fields of view and aspect ratios.

Figure 2 shows the stereo camera configuration for the low-resolution EO cameras and the IR cameras. The cameras were configured to trigger on a hardware signal controlled by the collection program. This ensures that each stereo image pair is captured at exactly the same time, and the pairs are time-stamped for alignment with truth data. Images were collected at 10 Hz.

2. *Calibration*

To perform image rectification and feature extraction, cameras must be calibrated properly. A metallic checkerboard with 30 mm^2 tiles was used to capture calibration images for the low-resolution EO cameras and the IR cameras. The high-resolution EO cameras were calibrated using a larger, matte checkerboard. OpenCV's stereo calibration function was used to compute the calibration parameters as described in Ref. [14]. The checkers on the metallic board were painted using white heat-insulating paint. This creates a temperature differential that mimics the color differential, and the same board could be used to calibrate the EO and IR cameras; however, it is easier to calibrate accurately with the larger board.

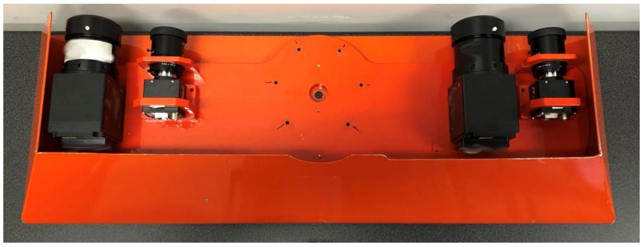


Fig. 2 Low-resolution EO cameras and IR cameras mounted for the first experiment.

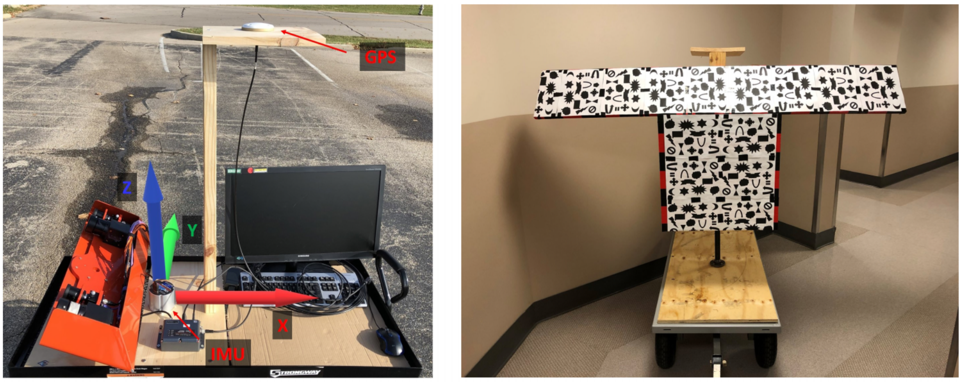
3. *Truth Data*

To produce accurate relative position between the pseudoreceiver and tanker used during experiments, a differential GPS (DGPS) system made by Swift Navigation [29] was used. Prior characterization of these systems shows the errors to be less than 1 cm at least 90% of the time with a standard deviation of 3 mm [30]. Because of this high accuracy, we consider the relative position estimates from this system to be "truth." While the DGPS only produces relative position, in true aerial refueling scenarios, small changes in orientation would quickly force the receiver out of the refueling envelope, creating an invalid approach; therefore, while ICP returns a six-degree-of-freedom rigid-body registration, the main concern is the 3-D offset vector. The DGPS collects at 5 Hz. To ensure that the pose is correlated to the correct stereo image pair, the computer's clock is synced with GPS time using a Time Machine TM2000A time server, and each image is time-stamped. The two closest DGPS solutions are linearly interpolated to the time that the image pair was captured: this solution is used as the truth data for a given image pair. The DGPS has centimeter-level error. An error this small would be sufficient for an aerial refueling connection using the tanker method. A system verified by this methodology would be directly deployable.

4. *Pseudotanker and Pseudoreceiver*

The pseudotanker was designed using a wagon that could securely support the vision system, the data collection computer, the primary DGPS, and a power supply system. Additionally, the GPS antenna was placed above all of the equipment to mitigate multipath or occlusion of GPS signals. Figure 3a shows the pseudotanker with the relative coordinate frame. We use a right-handed coordinate system, defining x pointing out of the front of the cart and the cameras on the back of the cart, y as pointing to the left of the cart, and z as pointing up.

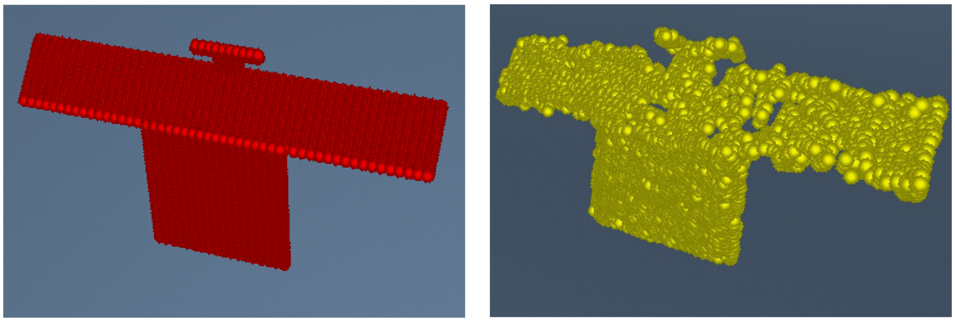
The pseudoreceiver was designed to mimic the scaled-down behavior of a generic receiver in a refueling approach. The main structure is a wing and body with patterns printed on it. Figure 3b shows a front view. Patterns are placed on the surface to mimic the paint variations, rivets, and other surface features that stereo block matching can locate on the surface of an approaching aircraft. The patterns were placed on the pseudoreceiver randomly to eliminate the existence of parallel lines, which would prevent stereo block matching from determining where matched pixels exist along the lines. Likewise, an aircraft has many unique corners and nonparallel features. For example, the corners between the fuselage and wings, the end of the wings, and the engines all provide unique corners for stereo block matching. Therefore, the randomized patterns were required to be placed throughout the pseudoreceiver to achieve a similar feature pattern to an aircraft.



a) Rear view of pseudotanker cart with primary axes shown

b) Front view of pseudoreceiver cart

Fig. 3 Pseudotanker and pseudoreceiver.



a) Red reference point cloud

b) Yellow sensed point cloud

Fig. 4 Reference model and sensed model as point clouds.

For pose registration, a reference point cloud (red) is matched onto a sensed point cloud (yellow) using the ICP [9]. Figure 4 shows the reference point cloud for the pseudoreceiver and an example of a sensed point cloud. The reference point cloud is assumed to be a known, uniformly sampled, geometrically accurate model of the approaching receiver. A database of these must be available to the pose registration of this algorithm. Previous work at the Air Force Institute of Technology has successfully used a different CNN to identify the approaching receiver type with over 91% accuracy [31].

5. *Running the Experiment*

The experiment was conducted in a parking lot to allow a large, relatively flat, open space. The pseudotanker remained stationary, and the pseudoreceiver was pushed toward it. Several approaches were conducted.

After the tests were conducted and truth data were obtained from postprocessing the DGPS data, we applied the computer vision pipeline to estimate the pseudoreceiver's pose. Figure 5 shows an example of registration being visualized in the virtual environment. This enabled recreation of the experiment and visualization of the pose estimation in postprocessing.

B. Experiment Results

This subsection discusses the results from the ground test. In Sec. II, we examined the expected effects of camera resolution on depth estimation. Here, we see the effects of 3-D point clouds generated by different camera systems on pose estimation.

Figure 6 shows the results for one approach using the IR low-resolution EO and high-resolution EO cameras, respectively. In this approach, the pseudoreceiver was pushed toward the pseudotanker as directly as possible. Note that the IR and low-resolution EO cameras struggle to find a meaningful registration at a range of 20 m, with residual errors near 0.5 m. In contrast, the high-resolution cameras have errors smaller than 0.1 m at ranges near 35 m.

Figure 7 shows the results for the second approach. In this approach, the pseudoreceiver was pushed a short distance and then halted for a few seconds. Since a real AAR approach might

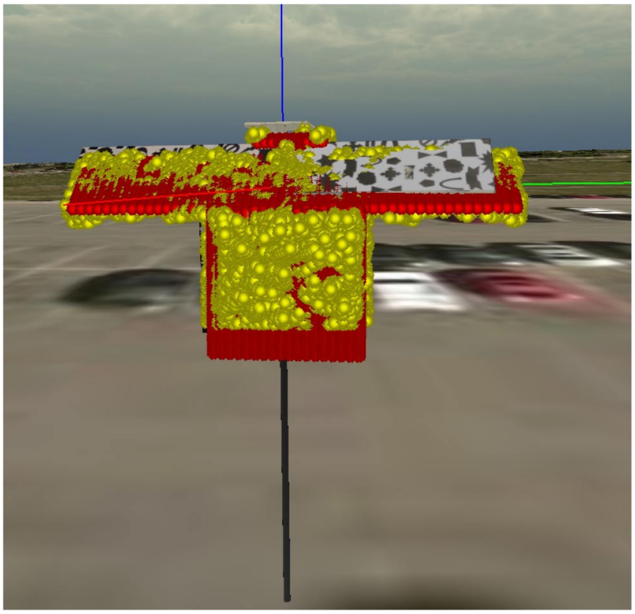


Fig. 5 Registration of the reference point cloud with the sensed point cloud.

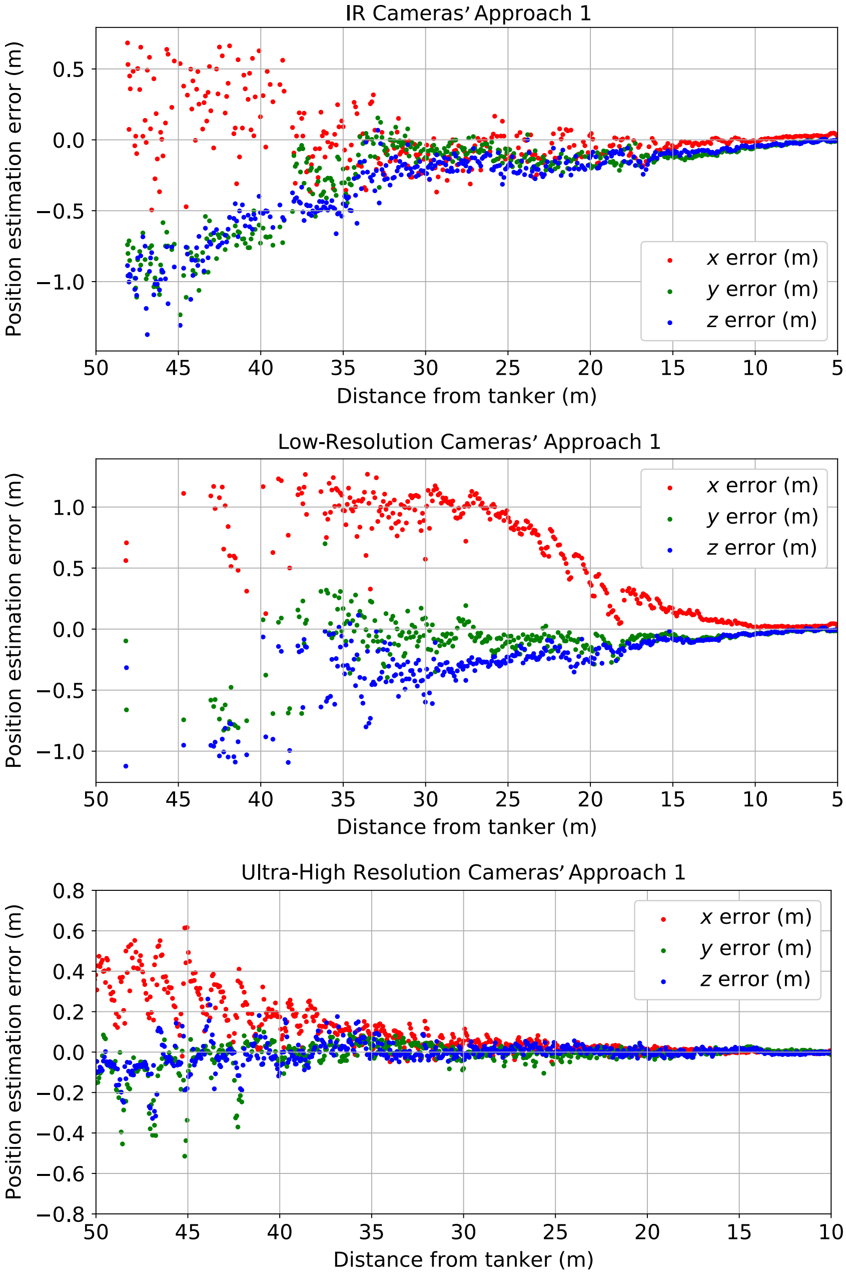


Fig. 6 Residual errors for the first approach.

not be fully continuous, it is important to show that the technique can accurately track changes in motion as well. The results are nearly indistinguishable from the first approach, which further validates that the high-resolution cameras provide a solution accurate enough for this application at distances up to about 40 m, unlike the lower-resolution cameras.

In the approaches shown by Fig. 8,[¶] the pseudoreceiver was moved side to side as it approached. This was designed to imitate an approach with suboptimal conditions that required frequent correction. Each camera system performed slightly worse; however, the high-resolution system still maintained errors smaller than 0.1 m at the target contact point of 30 m.

In all, it is clear that increasing resolution does improve pose-estimation accuracy. Moreover, while the low-resolution camera system struggles to obtain meaningful registrations at 20 m, the high-resolution system can perform well at ranges near 50 m. Note that, in Figs. 6–8, the y axes for the different cameras are different, with the high-resolution cameras consistently having the tightest bounds. Not only is the high-resolution camera providing good results at longer distances, but those results are better than the low-resolution cameras even at close distances. This effect is further demonstrated in Fig. 9, where we show the aggregate path estimation errors for each approach. The x, y, and z components display the mean absolute error at a given range across all three approaches. The 3-D error is obtained by taking the Euclidean distance between the sensed position and the truth position at each range. The error bars show a one-standard-deviation certainty associated with each mean. Once again, note that the y-axis limits are smaller for the high-resolution camera than the other two cameras. Furthermore, note that the 3-D error for the high-resolution cameras plus the error bound is less than the 10 cm benchmark required for AAR from 30 m in.

Finally, in Fig. 10, we examine the standard deviation of the errors directly. Note that the general trends of the low-resolution EO and IR cameras are generally the same, whereas the high-resolution

[¶] The receiver briefly left the IR camera field of view twice in this experimental approach; this is why there are no data from 35 to 30 m and an uptick in error at 12 m. The receiver did not leave the other cameras' fields of view because the IR cameras were mounted to the left of the EO cameras in that approach, and the receiver went too far right.

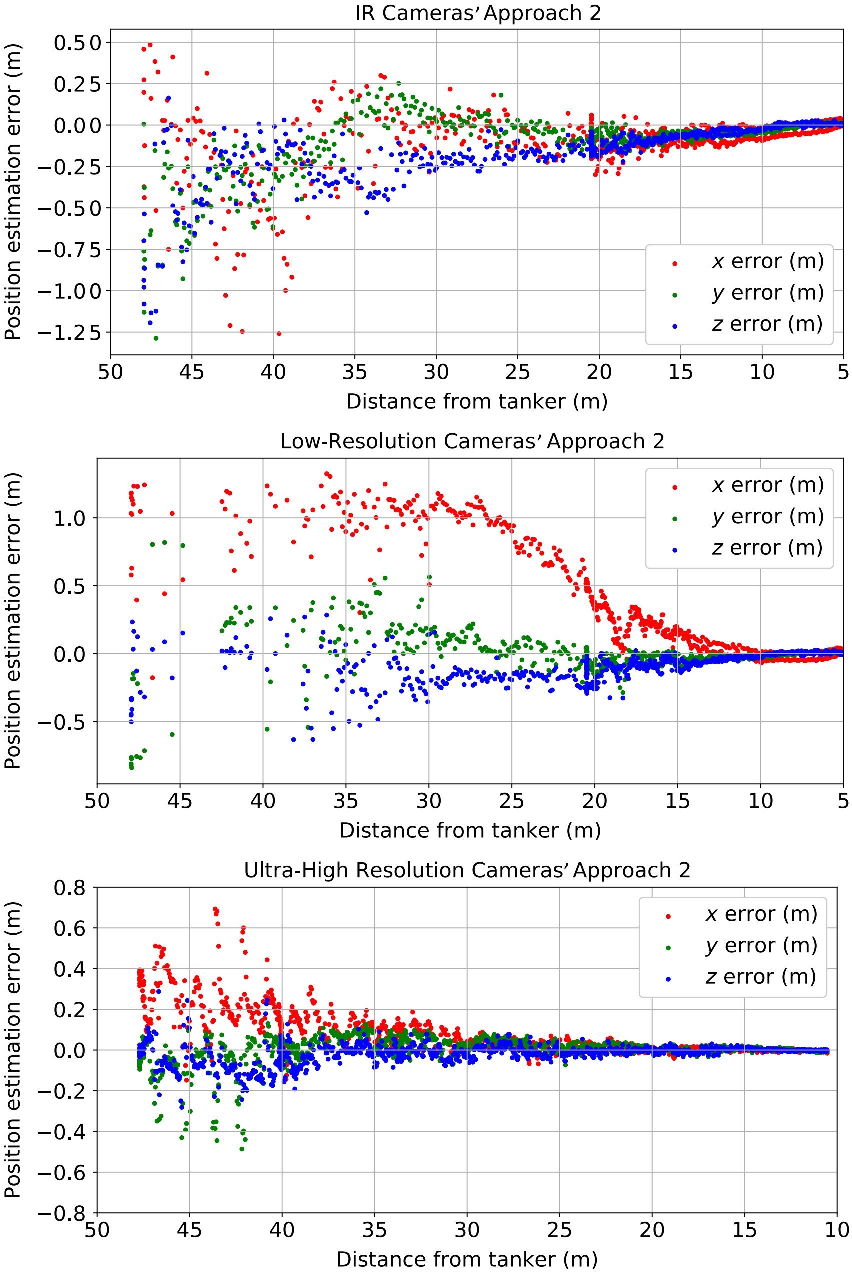


Fig. 7 Residual errors for the second approach.

EO cameras consistently and reliably have less standard deviation. This supports the premise that it is the increase in resolution that is important and not the specific modality being used.

These results indicate that our high-resolution camera system is capable of safely providing a sensed point cloud as a basis for pose estimation in AAR. However, as discussed in Sec. I, the increase in pixel count drastically increases the computation time required to generate a point cloud. The next section contributes a novel method to decrease point-cloud generation timing with minimal impact on pose-estimation fidelity.

IV. CNN Augmentation and Performance

To generate a 3-D point cloud faster while using high-resolution imagery, we modify the base vision pipeline (shown in Sec. III) by cropping the high-resolution images before generating a disparity map. This leads to an improved pipeline that we now seek to evaluate:

1) Capture high-resolution stereo imagery.
2) Use CNN to dynamically crop the stereo images.
3) Generate a disparity map only for the region of interest.
4) Convert the disparity map into a 3-D point cloud.
5) Use an ICP to register the receiver's pose.

To perform the dynamic cropping, we augment our vision pipeline with a deep learning model that is trained to segment computer-simulated imagery of a receiver aircraft. This model is deployed in a 3-D virtual world refueling simulation. The model crops the stereo images to only include a tightly bound rectangular portion of the original image containing the receiver; thus, significantly fewer pixels need to be processed using the stereo block matching algorithm. This results in a significant speedup without sacrificing precision. Section IV.A describes the 3-D virtual world. Section IV.B details how our deep learning model was designed and tested. Section IV.C explains how it was integrated into the stereo vision pipeline. Section IV.D examines the computation time required to generate a 3-D point cloud, demonstrates the speedup from using a CNN to perform image segmentation, and shows that image segmentation does not adversely affect pose registration accuracy.

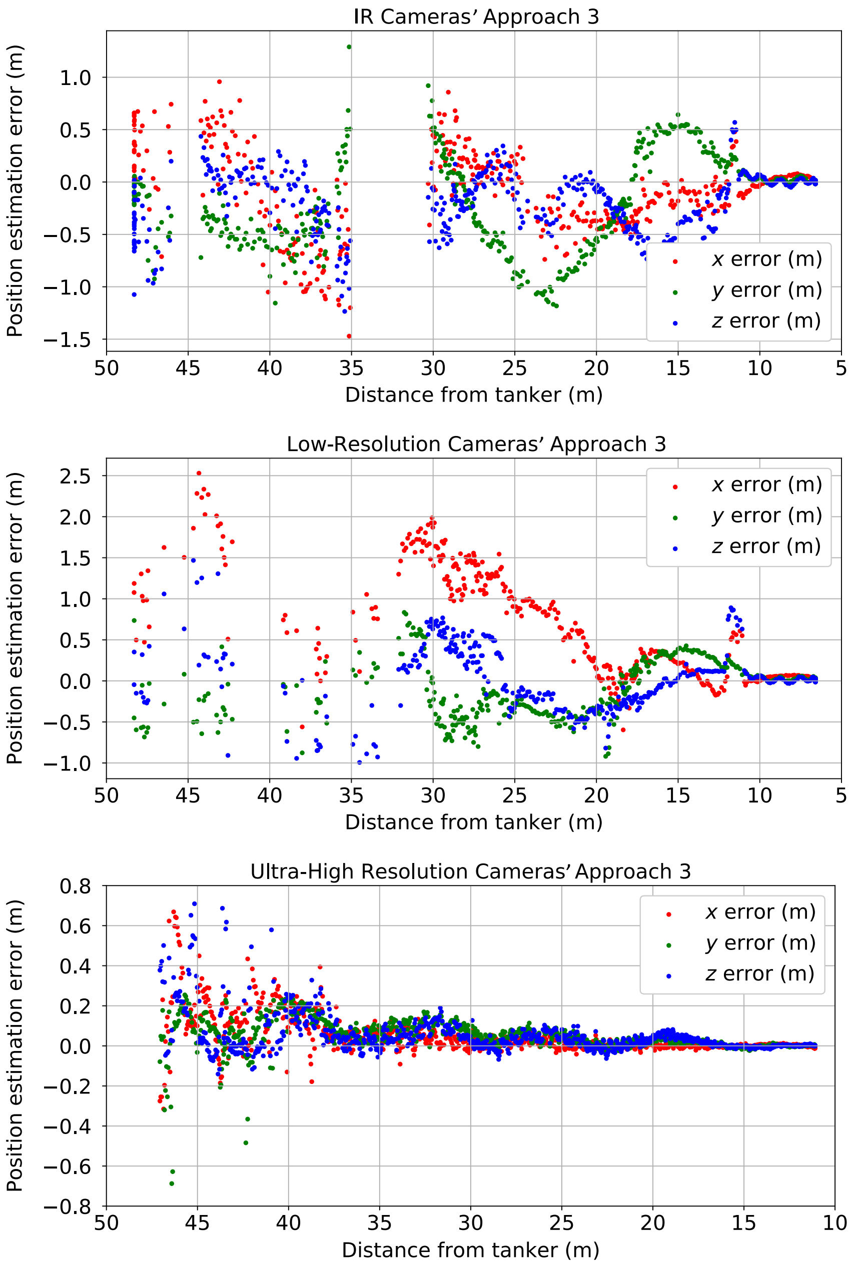


Fig. 8 Residual errors for the third approach.

A. Computer Simulation

To quantify performance benefits and simulation accuracy, simulations are performed in the Aftr Burner Engine. The Aftr Burner Engine is a general purpose 3-D graphics engine based on OpenGL. In this work, we use geometrically accurate models, high-quality textures, and realistic lighting to replicate real refueling approaches and generate synthetic imagery. This is the same simulation environment that several researchers have used [1,32,33] for their AAR experiments. The cameras in the simulation have the same resolution and field of view as their physical counterparts used in the ground experiment. To remove jagged edges caused by OpenGL's rendering pipeline, we perform multisample antialiasing with eight samples. Calibration values are calculated from known simulation parameters (resolution, field of view, and relative offset between the cameras) as described by Kaehler and Bradski [14]. To verify that increased resolution improves pose estimation, a simulated refueling approach was conducted in the virtual world using cameras at different resolutions, and the fidelity of the pose estimation is compared.

B. CNN Design

For this research, a basic deep CNN architecture was created with the goal of placing a rectangular bounding box around a receiver's location in an image. There are many existing architectures, such as You Only Look Once (YOLO) [18], that could perform a similar function; however, most image segmentation networks are designed for more general purposes. This one is designed for high-fidelity real-time segmentation on a specific target. We then demonstrate how this deep learning augmentation improves our computer vision pipeline, yielding large performance benefits. The remainder of this subsection explains how the network was trained and evaluated. Its image segmentation performance is then discussed.

1. Data

The dataset for this project was created using the Aftr Burner engine described in Sec. IV.A. In the simulation, a virtual receiver was placed at random, uniformly distributed locations within the camera's field of view at distances between 20 and 100 m, and its

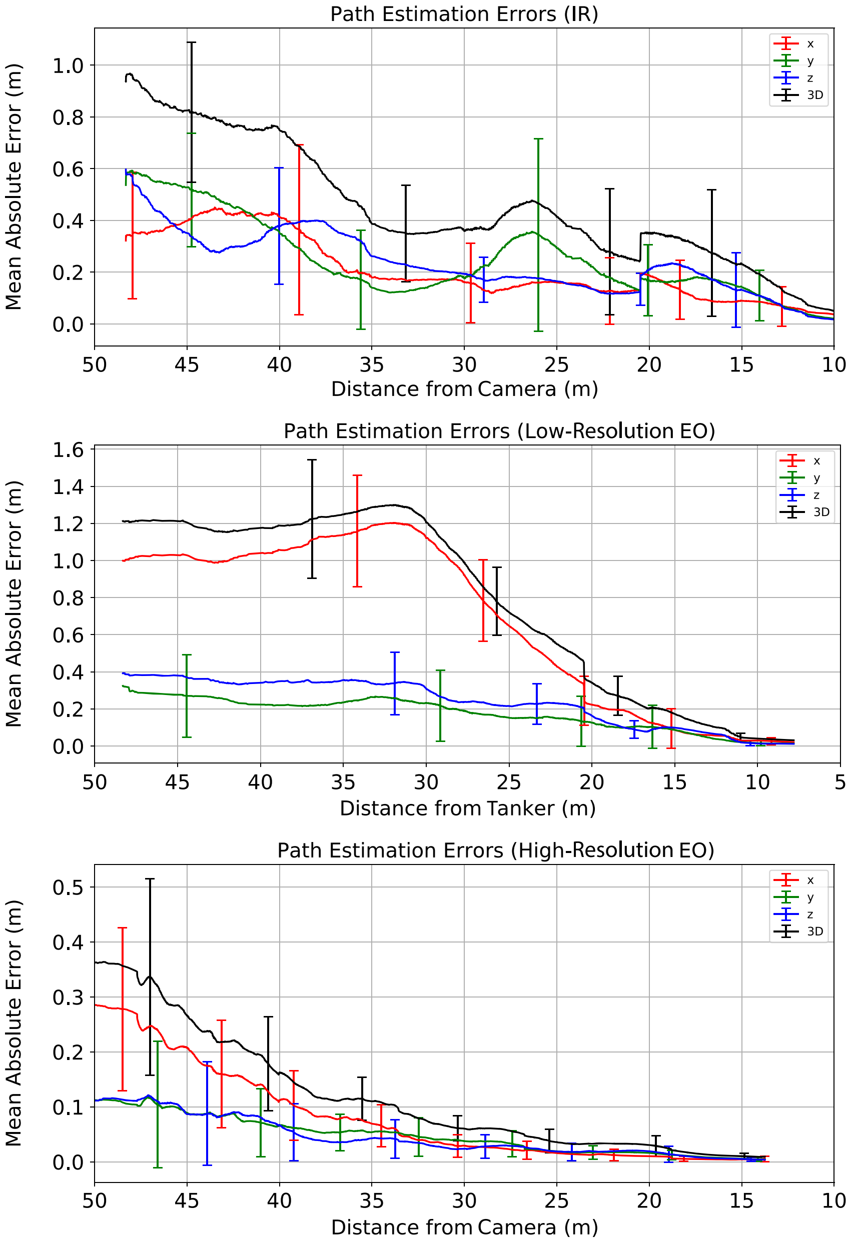


Fig. 9 Aggregate errors for each camera system across all three approaches.

orientation was randomly adjusted by small amounts to ensure diversity. One of several background images of real landscapes from aerial views was placed in the background. The engine used a virtual camera to capture a 1280×960 resolution image of the simulated scene. Next, the simulation was modified to reskin the receiver in a flat, distinct color. This color did not naturally occur in the sample images, allowing a color mask to easily locate the receiver. After a pair of images was captured, the background was changed and the

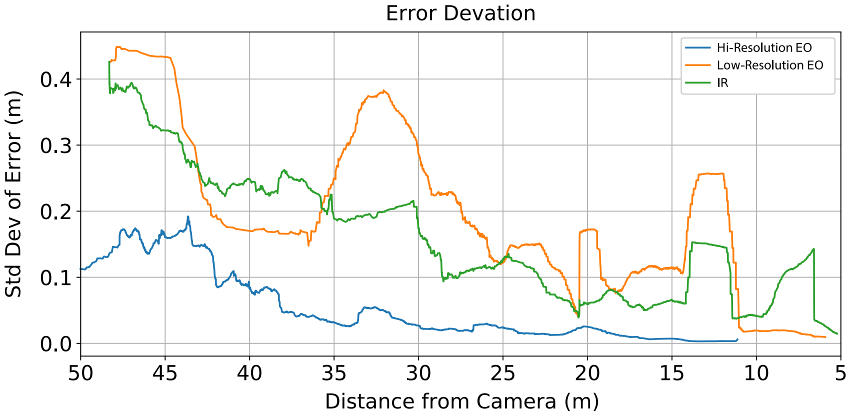


Fig. 10 Standard deviation (Std Dev) of the error in each camera system as a function of distance.

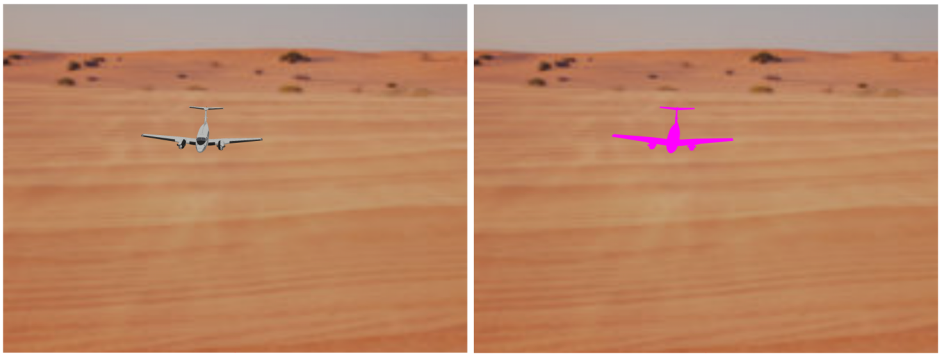


Fig. 11 Example training image pair captured from Aftr Burner.

receiver was moved. Figure 11 shows an example of an image pair. For this research, 5500 input/truth image pairs were generated. The project used 5000 pairs for training and validation and 500 pairs for testing. The test set was generated using different background images that were unseen in training.

To create the truth data, a mask was applied to the image to locate the pink skin (shown in the right picture of Fig. 11) on the receiver in the truth images. The minimum and maximum pixel coordinates were used to calculate the center x, center y, width, and height of the box in pixel space. These were saved in a comma separated values with the associated image number. Each of the training images was blurred using a 3×3 low-pass blurring filter to help prevent the model from overfitting potential sharp edges in the simulated imagery.

The model was further designed to perform image segmentation specifically for AAR. High-resolution cameras can often capture at higher frame rates in grayscale than in color; however, full resolution is not necessary to localize the receiver. Downsampling images allow for accurate localization with smaller networks. For these reasons, training images were downsampled to 512×386 and converted to grayscale. Brightness was varied in each image between 5 and 300% to help feature selection become less dependent on specific lighting. Pixel values were then rescaled to floats between zero and one. For testing, the pixel values are rescaled but not otherwise augmented.

2. *Model*

This research used a new deep CNN model. The model has 16 convolution layers and two fully connected layers before output. It takes an image as described in Sec. IV.B.1 and outputs regression values for the bounding box as center-x, center-y, width, and height values, normalized to be in range [0, 1] as a proportion of the original image. Figure 12 shows a high-level view of the model. The first layer performs a 12×9 convolution with a 4×3 stride and 128 filters to create a square $256 \times 256 \times 128$ tensor. The remaining 15 convolutional layers alternate between 3×3 and 1×1 filters with several pooling layers to perform feature extraction. Finally, there are two fully connected layers with 1024 and 512 nodes, followed by the four-node output layer that regresses the x, y, w, and h of the bounding box.

Batch normalization is performed at each layer. The leaky rectified linear unit function** serves as the activation function for each layer. There is a 40% dropout before each fully connected layer and a 20% dropout before the output layer. These regularization techniques help prevent overfitting the training data. The training loss function is the MSE for each predicted value (x, y, width, height, scaled [0, 1]).

The workstation training and evaluating the model had an Intel i7-7820X processor, 96 GB of main memory, and an Nvidia 1080Ti graphics processing unit (GPU). The model trained in less than 3 h. By functioning on a consumer-level computer, the model demonstrates that it can be used in practical settings.

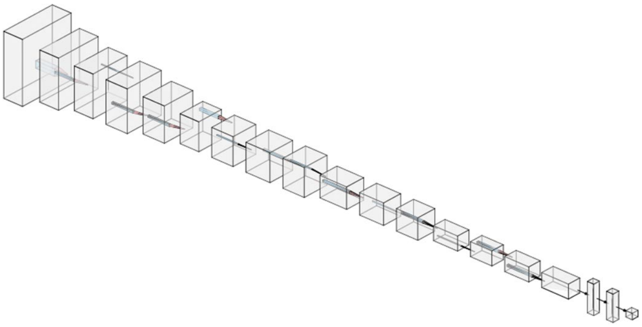


Fig. 12 CNN model used for this research.

3. *Testing*

Test results are calculated using a set of 500 images that were generated and labeled as described in Sec. IV.B.1. The backgrounds used in these images are totally different from the backgrounds used in training, and they are not seen by the model before the testing. The model's prediction can be directly compared to the truth data. To quantify the model's performance, we measure the distance between the predicted bounding box's center and the true bounding box's center in pixel space. We measure the CNN's error in pixel space because stereo block matching generates disparity values in pixel space as well. One of the best ways to assess accuracy for this application, though, is by viewing the images directly. Section IV.B.4 shows examples of bounding boxes generated by the model and compares these to the truth data of the original images.

4. *Network Performance*

Table 2 shows the quantitative measures of the network's performance. We evaluate the root-mean-square error (RMSE) and MAE of the predicted bounding box compared to the truth bounding box in pixels. A few outliers slightly skew the errors; however, most are very near zero. Figure 13 shows the distribution of errors in x, y, width, and height. On average, the network performs very well. Figure 14 shows the amount of the true bounding box that the predicted bounding box covers as a function of distance between the camera and the receiver. In the entire test set of 500 images, there is only one image where the predicted bounding box does not overlap the truth bounding box. Moreover, on average, the model's prediction overlaps 90% of the true bounding box. These results demonstrate stable behavior throughout the refueling approach.

Figure 15 shows four examples of the CNN's typical performance. These are consistent with the model's performance across the test dataset. The model can also be used without transfer learning or other modification to evaluate images from a real flight test where a C-12 is the receiver.†† Figure 16 shows an example of the model directly being used on imagery from a physical camera. Unfortunately, we do

**$$f(x) = \begin{cases} 0.1x & x \le 0 \\ x & x > 0 \end{cases}.$$

††Modification to the network would likely be needed if the receiver were a different aircraft.

Table 1 Estimated depth for a feature located 30 m away from stereo cameras

Resolution	Mean depth estimation, m	MAE	Calculated error, m	Percent difference, %
1280 × 960	30.10 m	1.427 m	1.496 m	4.6
1920 × 1440	30.02 m	0.9435 m	0.997 m	5.3
3840 × 2880	30.00 m	0.4650 m	0.498 m	6.6
4896 × 3264	30.00 m	0.3844 m	0.3910 m	1.6

Table 2 Errors for the deep learning model (in pixels, images at 1280 × 960) on the test set

	x	*y*	Width	Height
RMSE	14.72	9.99	19.41	10.43
RMSE%	1.15	1.04	1.52	1.06
MAE	10.46	6.21	13.18	6.31
RMSE%	0.82	0.65	1.03	0.66

not have a large enough labeled dataset using physical imagery with varied views of a receiver to compute full evaluation metrics.

C. CNN Application Procedure

To provide a speedup to AAR pose estimation, the CNN's bounding box is used to reduce the computational cost of stereo block matching. Once the stereo images are captured, the left image is downsampled from the original resolution to 512 × 386 and passed as input to the CNN. While it would be possible to perform bounding on both images, the disparities at 30 m are only a few pixels. This means that the error from assuming both bounding boxes are the same is small enough that it does not appreciably decrease the CNN's performance. Additionally, this means the network only runs once, saving valuable computation time. The bounding box is then used to mask a precomputed rectification map (Kaehler and Bradski give an in-depth explanation for the rectification process [14]). The captured images are remapped using this now-cropped rectification map into a final pair of rectified, undistorted, and cropped images. These images are then passed into OpenCV's stereo block matcher to generate a disparity map. Finally, the disparity map is reprojected into 3-D space for use as a point cloud for pose registration. To compare the previous pipeline with the new one, we collect data on the precision of the pose-estimation process for a simulated approach and time the point-cloud generation process for stereo camera pairs at a variety of resolutions.

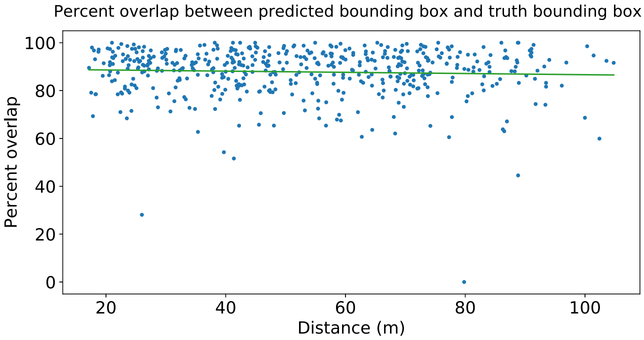


Fig. 14 Percent of the true bounding box that the predicted bounding box covers and the best-fit line.

D. Point-Cloud Generation Timing and Pose-Estimation Precision

This subsection evaluates the CNN's performance at improving the pose-estimation process for AAR. First, we demonstrate that the CNN-augmented pipeline requires much less time to generate a 3-D point cloud. Next, we examine the effects of using the CNN-augmented pipeline on the precision and deviation in pose-estimation accuracy. Table 3 shows the time required to generate a 3-D point cloud at four sample resolutions with and without the CNN augmentation. The CNN itself takes between 5 and 6 ms to execute. Because the image is downsampled before the CNN runs, its execution time is independent of input resolution. It is clear that the CNN provides a substantial speedup at all resolutions. Creating a 3-D point cloud for the 1280 × 960 image pairs is 3.6 times faster with the CNN augmentation. When the resolution increases to 4896 × 3264, the CNN gives an 11.3-time speedup.

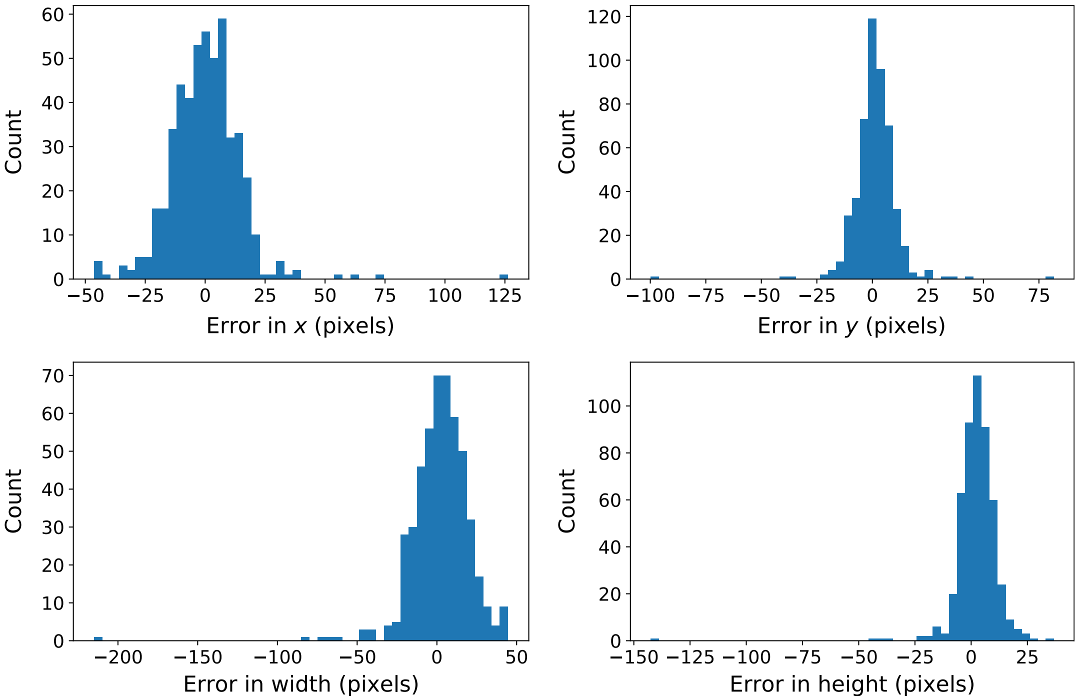


Fig. 13 CNN predicted bounding box errors in *x*, *y*, *w*, and *h* (in pixels).

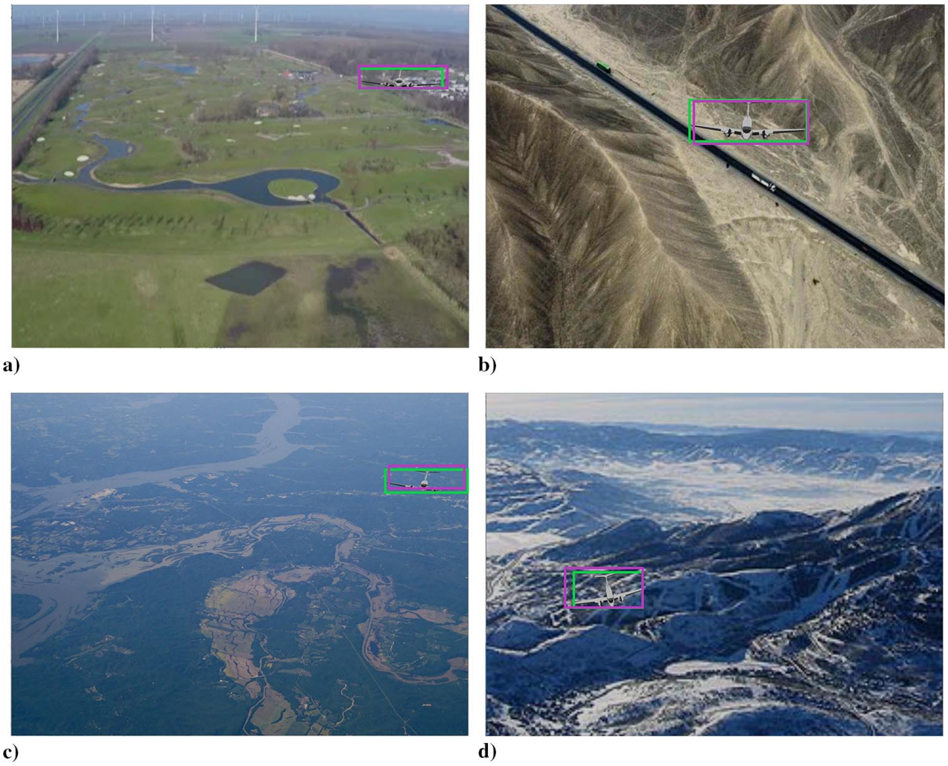


Fig. 15 Examples of CNN model performance. The green box represents the ground-truth bounding box. The purple box represents the network's predicted bounding box.

Finally, it is important to verify that the CNN augmentation does not adversely affect pose estimation. As with the ground experiment, Fig. 17 shows the 3-D path estimation error over the approach. Figure 18 shows the standard deviation of the 3-D error for the 1280 × 960 camera system and the 4896 × 3264 camera system with and without CNN augmentation to the vision pipeline. The approach path taken by the receiver in this section is visualized virtually on YouTube [34]. Note that the low-resolution system's solution, even in the simulation, is consistently worse than the high-resolution system. Also note there is not an appreciable difference between the high-resolution systems error with or without the CNN augmentation. This shows the system can reliably perform as well as the original pipeline while also gaining a large speedup. The data appear noisy because the receiver's path in the simulation was generated by replaying navigation data from a real aerial refueling approach. The velocity was not constant, and certain distances have many more data points. However, there is no appreciable difference between the high-resolution system with and without the CNN augmentation, further validating its viability. The apparent decrease in error at about 48 m is from the receiver, holding its position for a period of time before continuing its advance toward the tanker. The increase in error around 35 m, as shown in the video [34], is caused by the tail of the receiver briefly leaving the camera's field of view.

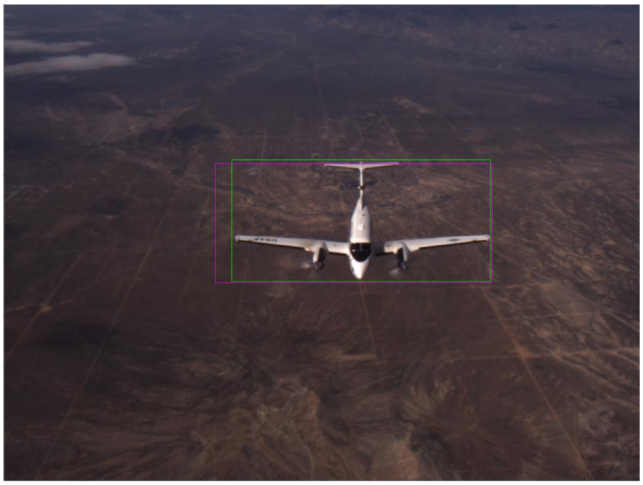


Fig. 16 Performance example for the CNN model with real flight-test imagery.

V. Conclusions

In this paper, the current stereo vision pipeline is validated to perform pose estimation for AAR with a novel ground experiment. The system consistently achieves 3-D pose-estimation errors of less than 6 cm. Based on these results, a stereo camera system with adequate resolution can safely control a receiver in the refueling envelope to make and maintain contact. However, high-resolution imagery comes with a computation-time cost.

Next, a computer vision pipeline is outlined that combines conventional stereo vision with deep learning to greatly accelerate the process of generating a 3-D point cloud of the receiver. It is further verified that the speedup does not decrease the precision gained from using high-resolution stereo imagery. While this system was developed specifically for AAR, any real-time computer vision application could benefit from its use. For example, a CNN can identify and label several objects of interest in a stereo image pair and then perform the pose-estimation process quickly on each of them. Since the point clouds are generated from finely cropped images, the resulting 3-D point clouds are already semantically segmented. This technique

Table 3 Point-cloud generation time with and without using the CNN

Resolution	CNN off, ms	CNN on, ms	Speedup
1280 × 960	36.13	10.57	3.42
1920 × 1440	76.77	14.67	5.23
3840 × 2880	357.49	35.03	10.19
4896 × 3264	524.28	46.52	11.27

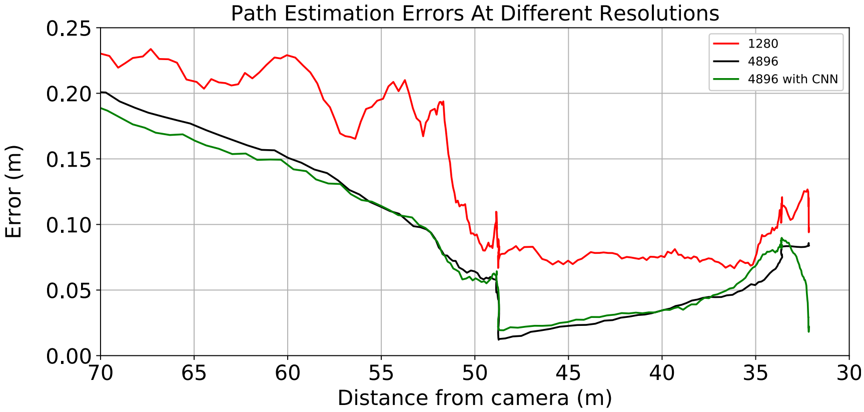


Fig. 17 Path estimation error (Euclidean distance from truth position to sensed position, in meters).

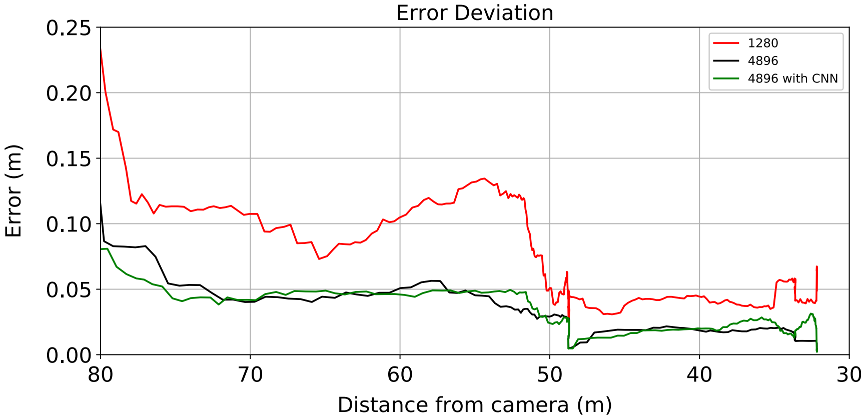


Fig. 18 Standard deviation of the error of each camera system's pose estimation as a function of distance.

could be used to provide benefits for many computer vision applications, including autonomous vehicles, robot navigation, or structure from motion.

The greatest remaining limitation for pose-estimation speed is point-cloud registration. Using a parallel ICP or development of a faster point-cloud registration algorithm will be important for any future efforts to increase pose-estimation rates. Further study could be done to determine if pixelwise image segmentation could yield even greater speedups for the AAR domain. A pixelwise mask could be used to crop the images similarly to the rectangle the current method uses. Additionally, errors in pose estimation and image segmentation could likely be greatly reduced by leveraging the time-series nature of most visual relative navigation tasks. For segmentation, leveraging a recurrent neural network to perform image segmentation may allow better tracking of objects. Future work to improve pose estimation may employ a Kalman filter.

Acknowledgments

This work was funded by the U.S. Air Force Research Laboratory Aerospace Systems (AFRL/RQ). Special thanks to Michael Crowl for his help in setup and data collection for the ground experiment. The views expressed in this paper are those of the authors and do not reflect the official policy or position of the U.S. Air Force, the U.S. Department of Defense, or the U.S. Government.

References

[1] Parsons, C., Paulson, Z., Nykl, S., Dallman, W., Woolley, B. G., and Pecarina, J., "Analysis of Simulated Imagery for Real-Time Vision-Based Automated Aerial Refueling," *Journal of Aerospace Information Systems*, Vol. 16, No. 3, 2019, pp. 77–93. https://doi.org/10.2514/1.I010658

[2] Lee, A., and Nykl, S., "Long Range Pose Estimation for Aerial Refueling Approaches Using Deep Neural Networks," May 2020, https://youtu.be/7YFlrhCXzME [retrieved 19 May 2020].

[3] Fravolini, M. L., Campa, G., and Napolitano, M. R., "Modelling and Performance Analysis of a Machine Vision-Based Semi-Autonomous Aerial Refuelling," *International Journal of Modelling, Identification and Control*, Vol. 3, No. 4, 2008, Paper 357. https://doi.org/10.1504/IJMIC.2008.020544

[4] Liu, K., Moore, C., Buchler, R., Bruner, P., Fax, A., Hinchman, J., Nguyen, B., Nelson, D., Ventrone, F., and Thorward, B., "Precision Relative Navigation Solution for Autonomous Operations in Close Proximity," *2008 IEEE/ION Position, Location, and Navigation Symposium*, 2008, pp. 1246–1251. https://doi.org/10.1109/PLANS.2008.4570045

[5] Zhang, X., Jiang, Z., and Zhang, H., "Real-Time 6D Pose Estimation from a Single RGB Image," *Image and Vision Computing*, Vol. 89, Sept. 2019, pp. 1–11. https://doi.org/10.1016/j.imavis.2019.06.013

[6] Ferrara, P., Piva, A., Argenti, F., Kusuno, J., Niccolini, M., Ragaglia, M., and Uccheddu, F., "Wide-Angle and Long-Range Real Time Pose Estimation: A Comparison Between Monocular and Stereo Vision Systems," *Journal of Visual Communication and Image Representation*, Vol. 48, Oct. 2017, pp. 159–168. https://doi.org/10.1016/j.jvcir.2017.06.008

[7] Fravolini, M. L., Mammarella, M., Campa, G., Napolitano, M. R., and Perhinschi, M., "Machine Vision Algorithms for Autonomous Aerial Refueling for UAVs Using the USAF Refueling Boom Method," *Innovations in Defence Support Systems–1*, Studies in Computational Intelligence, Vol. 304, Springer, New York, 2010, pp. 95–138. https://doi.org/10.1007/978-3-642-14084-6_5

[8] Scharstein, D., "A Taxonomy and Evaluation of Dense Two-Frame Stereo," *International Journal of Computer Vision*, Vol. 47, No. 1, 2002, pp. 7–42. https://doi.org/10.1023/A:1014573219977

[9] Besl, P., and McKay, N., "Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, 1992, pp. 239–256. https://doi.org/10.1109/34.121791

[10] Mourning, C., Nykl, S., Xu, H., Chelberg, D., and Liu, J., "GPU Acceleration of Robust Point Matching," *International Symposium on Visual Computing: Advances in Visual Computing*, Vol. 6455, *Lecture Notes in*

Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, New York, 2010, pp. 417–426.
https://doi.org/10.1007/978-3-642-17277-9_43

[11] Zhou, Q. Y., Park, J., and Koltun, V., "Fast Global Registration," *European Conference on Computer Vision*, Vol. 9906, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, pp. 766–782.
https://doi.org/10.1007/978-3-319-46475-6_47

[12] Elbaz, G., Avraham, T., and Fischer, A., "3-D Point Cloud Registration for Localization Using a Deep Neural Network Auto-Encoder," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Publ., Piscataway, NJ, 2017, pp. 4631–4640, http://openaccess.thecvf.com/content_cvpr_2017/papers/Elbaz_3-D_Point_Cloud_CVPR_2017_paper.pdf [retrieved 15 July 2019].

[13] Gallup, D., Frahm, J.-M., Mordohai, P., and Pollefeys, M., "Variable Baseline/Resolution Stereo Project AutoVision: Localization and 3-D Scene Perception for an Autonomous Vehicle with a Multi-Camera System View Project CNN based Autonomous Driving View project Variable Baseline/Resolution Stereo," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Publ., Piscataway, NJ, 2008, pp. 1–12.
https://doi.org/10.1109/CVPR.2008.4587671

[14] Kaehler, A., and Bradski, G., *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*, 1st ed., O'Reilly Media, Sebastopol, CA, 2016, pp. 703–783.

[15] O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., and Walsh, J., "Deep Learning vs. Traditional Computer Vision," *Advances in Intelligent Systems and Computing*, Vol. 943, 2020, pp. 128–144.
https://doi.org/10.1007/978-3-030-17795-9_10

[16] Krizhevsky, A., "Convolutional Deep Belief Networks on CIFAR-10," *Advances in Neural Information Processing Systems*, AlexNet, 2010, pp. 1–12, https://www.cs.toronto.edu/kriz/conv-cifar10-aug2010.pdf.

[17] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q., "Densely Connected Convolutional Networks," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Publ., Piscataway, NJ, 2017, pp. 4700–4708, https://github.com/liuzhuang13/DenseNet [retrieved 15 July 2019].

[18] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Publ., Piscataway, NJ, 2016, pp. 779–788.

[19] Redmon, J., and Farhadi, A., "YOLO9000: Better, Faster, Stronger," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Publ., Piscataway, NJ, 2017, pp. 7263–7271, http://pjreddie.com/yolo9000/.

[20] Zhao, Z.-Q., Zheng, P., Xu, S.-T., and Wu, X., "Object Detection with Deep Learning: A Review," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 30, No. 11, 2019, pp. 3212–3232.
https://doi.org/10.1109/TNNLS.2018.2876865

[21] Lee, A., "Comparing Deep Neural Networks and Traditional Vision Algorithms in Mobile Robotics," Swarthmore Univ. TR CS81F2015, Swarthmore, PA, 2015, https://www.cs.swarthmore.edu/~meeden/cs81/f15/papers/Andy.pdf [retrieved 12 Aug. 2019].

[22] Zhou, B., Krähenbühl, P., and Koltun, V., "Does Computer Vision Matter for Action?" *Science Robotics*, Vol. 4, No. 30, 2019, Paper eaaw6661.
https://doi.org/10.1126/scirobotics.aaw6661

[23] Garcia-Garcia, A., Gomez-Donoso, F., Garcia-Rodriguez, J., Orts-Escolano, S., Cazorla, M., and Azorin-Lopez, J., "PointNet: A 3-D Convolutional Neural Network for Real-Time Object Class Recognition," *Proceedings of the International Joint Conference on Neural Networks*, IEEE, New York, 2016, pp. 1578–1584.
https://doi.org/10.1109/IJCNN.2016.7727386

[24] Torralba, A., and Efros, A. A., "Unbiased Look at Dataset Bias," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Soc., Piscataway, NJ, 2011, pp. 1521–1528.
https://doi.org/10.1109/CVPR.2011.5995347

[25] Vazquez, D., Lopez, A. M., Marin, J., Ponsa, D., and Geronimo, D., "Virtual and Real World Adaptation for Pedestrian Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 4, 2014, pp. 797–809.
https://doi.org/10.1109/TPAMI.2013.163

[26] Goyal, M., Rajpura, P., Bojinov, H., and Hegde, R., "Dataset Augmentation with Synthetic Images Improves Semantic Segmentation," *Computer Vision, Pattern Recognition, Image Processing, and Graphics*, edited by R. Rameshan, C. Arora, and S. Dutta Roy, NCVPRIPG 2017, Communications in Computer and Information Science, Vol. 841, Springer, Singapore, 2018, pp. 348–359.
https://doi.org/10.1007/978-981-13-0020-2_31

[27] Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M., "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes," *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Publ., Piscataway, NJ, 2016, pp. 3234–3243.

[28] French, B., and Nykl, S., "Determining Virtually Simulated Aerial Refueling Fidelity using Physically Collected Stereo Vision Images and DGPS-Based Truth Data," *Joint Navigation Conference*, Inst. of Navigation, 2019, pp. 1–8.

[29] "Swift Navigation," 2020, https://www.swiftnav.com/ [retrieved 9 June 2020].

[30] Schmalzried, R., "*The Role of RTK in the Autonomous System Sensor Suite: An Examination of Moving Baseline RTK, RTK-Based Heading Technology, and How RTK-Based Solutions Support Autonomous Vehicle Sensor Edge Cases*," Swift Navigation TR, San Francisco, CA, 2017, https://www.swiftnav.com/whitepaper/rtk-autonomous-system-sensor-suite [retrieved 9 June 2020].

[31] Mash, R., Borghetti, B., and Pecarina, J., "Improved Aircraft Recognition for Aerial Refueling Through Data Augmentation in Convolutional Neural Networks," *International Symposium on Visual Computing*, Vol. 10072, Springer, New York, 2016, pp. 113–122.
https://doi.org/10.1007/978-3-319-50835-1_11

[32] Seydel, N., Dallmann, W., and Nykl, S., "Visualizing Behaviors when Using Real vs Synthetic Imagery for Computer Vision," *16th International Conference on Scientific Computing*, Las Vegas, NV, 2018, pp. 1–8.

[33] Anderson, J. D., Nykl, S., and Wischgoll, T., "Augmenting Flight Imagery from Aerial Refueling," *International Symposium on Visual Computing*, 2019, pp. 154–165.
https://doi.org/10.1007/978-3-030-33723-0_13

[34] Parsons, C., and Nykl, S., "Automated Aerial Refueling Using Stereo Vision and Iterative Closest Point," 2017, https://youtu.be/JbSWxybMG0?t=129 [retrieved 19 May 2020].

M. J. Kochenderfer
Associate Editor