Neural Computing and Applications
https://doi.org/10.1007/s00521-024-10543-1

ORIGINAL ARTICLE

Machine visual perception from sim-to-real transfer learning for autonomous docking maneuvers

Derek Worth[1] · Jeffrey Choate[1] · Ryan Raettig[1] · Scott Nykl[1] · Clark Taylor[1]

Received: 18 June 2024 / Accepted: 3 October 2024

Abstract
This paper presents a comprehensive approach to enhancing autonomous docking maneuvers through machine visual perception and sim-to-real transfer learning. By leveraging relative vectoring techniques, we aim to replicate the human ability to execute precise docking operations. Our study focuses on autonomous aerial refueling as a use case, demonstrating significant advancements in relative navigation and object detection. We introduce a novel method for aligning digital twins using fiducial targets and motion capture data, which facilitates accurate pose estimation from real-world imagery. Additionally, we develop cost-efficient annotation automation techniques for generating high-quality You Only Look Once training data. Experimental results indicate that our transfer learning methodologies enable accurate and reliable relative vectoring in real-world conditions, achieving error margins of less than 3 cm at contact (when vehicles are approximately 4 m from the camera) and maintaining performance at over 56 fps. The research findings underscore the potential of augmented reality and scene augmentation in improving model generalization and performance, bridging the gap between simulation and real-world applications. This work lays the groundwork for deploying autonomous docking systems in complex and dynamic environments, minimizing human intervention and enhancing operational efficiency.

Keywords Relative navigation · Annotation automation · YOLO · Perspective-n-point · Autonomous docking · Machine transfer learning

1 Introduction

Docking operations refer to the procedures and activities involved in maneuvering a vehicle, such as a boat, spacecraft, airplane, or ground vehicle, to connect securely with a docking station, specific location, or another vehicle. The goal is to achieve a stable and precise alignment for purposes ranging from maintenance and repair actions to the transfer of fuel, cargo, passengers, or data [31, 50].

Humans possess an innate ability to seamlessly execute docking maneuvers, leveraging their visual perception and hand-eye coordination to adeptly guide multiple objects together [52]. Take, for instance, a skilled driver who relies on vision to safely park a car (Object 1) inside a garage (Object 2). This activity demands the person recognize the two objects within a common field of view, infer from this view the position and orientation (i.e., pose) of each object relative to the observer, and interpret these poses as relative to each other [15]. Remarkably, humans perform these cognitive tasks simultaneously as part of a closed control loop (e.g., steering the car), resulting in docking maneuvers free of collision or mishap.

Unfortunately, having a human-in-the-loop limits docking operations as well. They require special experience and months, or even years, of education and training before attaining proficiency, especially on docking

✉ Derek Worth
derek.worth.2@spaceforce.mil

Jeffrey Choate
jeffrey.choate@au.af.edu

Ryan Raettig
ryan.raettig@au.af.edu

Scott Nykl
scott.nykl@au.af.edu

Clark Taylor
clark.taylor.3@au.af.edu

[1] Department of Electrical and Computer Engineering, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, USA

Published online: 03 December 2024

maneuvers with high complexity—think aerial refueling [14], space rendezvous and proximity operations [18], and maritime ship docking [45]. Moreover, humans need periodic rest and are prone to physical exhaustion, mental fatigue, distractions, emotional distress, task saturation, and other negative influences that often lead to mistakes, property damage, injuries, or worse [53, 68, 71]. Eventually, they also age out and permanently retire from service, sometimes taking with them decades of invaluable skills and experience. But imagine a scenario in which these limitations were eliminated… what if humans were replaced with machines possessing comparable intelligence and visual perception? Is it feasible to develop such a machine—one that can be easily replicated, programmed with accumulated knowledge overnight, and deployed in hazardous environments, thus entirely mitigating risk to human life?

Autonomous docking is not a new concept and has a rich history of innovation and achievement. One of the earliest and most notable instances dates back to October 30, 1967, when the Soviet Union successfully automated the docking between spacecrafts Kosmos-186 and Kosmos-188 [70]. Despite significant advancements since then, state-of-the-art autonomous docking solutions still face substantial limitations and cannot yet fully replace human operators. In some systems, human supervisory control is maintained as a fallback option, allowing operators to intervene if necessary. However, fully removing humans from the loop remains a key challenge, especially in environments like aerial refueling, where GPS-based solutions can be denied [4], inertial systems are prone to noise and drift over time [51], and current vision algorithms do not simultaneously meet the accuracy, reliability, and execution speed requirements needed in such dynamic and unpredictable conditions [84].

Worth et al. proposed an intelligent computer vision pipeline designed to meet these complex requirements in a previous study [84]. It was named *Relative Vectoring using Dual Object Detection* due to its capability to mimic human visual perception by intelligently detecting and simultaneously processing two objects within an image. From this single image, the pipeline estimates the poses of the objects, interprets these poses as relative to each other, and transforms the vector between them into the relative local reference frame of either object—hence the term "relative vectoring." Essentially, this pipeline enables Object 1 to determine the location of Object 2 relative to itself. Simulations demonstrated that this pipeline reliably produces relative vectors with an accuracy within 3 cm at contact, when the vehicles are less than 4 m from the vision sensor. It operates in real time (at least 45 fps) on a laptop equipped with an Nvidia RTX A5000 GPU, is resilient to occlusions, does not require extrinsic camera calibrations, and ultimately facilitates relative navigation between objects using vision alone. These attributes make relative vectoring ideally suited for autonomous docking maneuvers.

One major drawback, however, is that relative vectoring was only demonstrated in simulation [83, 84]. Machine transfer learning, which focuses on transferring knowledge across domains, is a promising methodology for bridging this sim-to-real gap [97]. This paper extends the relative vectoring pipeline into the real world by taking advantage of transfer learning. You Only Look Once (YOLO), a state-of-the-art object detection system renowned for its exceptional speed and accuracy [38], was trained to find features in synthetic imagery generated in a lab.[1] Then, the knowledge learned from this task was transferred to a different but related task, namely, finding those same features in real operational imagery. Specifically, a novel digital twin alignment technique was performed that aligns real objects and their realistic virtual counterparts in 3D space as well as corresponding real and virtual imagery of such objects through the use of fiducial targets and motion capture (mocap) data. Doing so enabled accurate 3D point projections and effective annotation automation of synthetic images containing real objects. Lastly, techniques were proposed for combining synthetic and real operational docking imagery through augmented reality and scene augmentation to produce precisely labeled, blended images with enough variations to improve generalization and enable positive transfer learning.[2]

Although relative vectoring can be adapted to address nearly any autonomous docking problem, the experiments in this work were confined to an autonomous aerial refueling (AAR) use case. To quantitatively assess pipeline performance, relative vectoring was conducted in a laboratory setting using full-scale aerial refueling objects, real camera imagery, and corresponding mocap truth data. Additionally, real flight testing was performed under a variety of operational conditions to confirm positive transfer learning and consistent generalization to previously unseen imagery. Overall, the results demonstrate that the transfer learning techniques proposed in this paper enable real-world relative vectoring with accuracy, reliability, and speed matching previous results observed only in simulation, achieving an error of less than 3 cm at contact 100% of the time and maintaining a frame rate of over 56 fps on a Nvidia RTX 4090 Laptop GPU. The results of relative vectoring between two air vehicles are

[1] For this effort, the unaltered Ultralytics YOLOv5 PyTorch implementation found at [78] was used.

[2] In the context of this work, synthetic imagery is defined as images collected from non-operational sources and can include virtual, hybrid, and real imagery—e.g., real images collected in a laboratory.

analyzed and presented. Furthermore, the findings indicate that augmented reality and scene augmentation significantly enhance YOLO model generalization and performance.

To summarize, the objective of this research was to enable the relative vectoring pipeline on real-world imagery using machine transfer learning, effectively bridging the sim-to-real gap. Contributions set forth by this paper include:

1. A novel method for performing digital twin alignment of machine vision using fiducial targets and mocap data—aligning the virtual world to the real world as observed through corresponding imagery.
2. Novel cost-efficient annotation automation techniques for rapidly generating realistic and precisely labeled YOLO training data.
3. Proven YOLO models that can consistently and accurately find 3D points in real 2D images through machine transfer learning on synthetic data.
4. Novel augmented reality and scene augmentation techniques that increase YOLO model generalization and performance.

The remainder of this paper is divided into five sections. Section 2 discusses related works in the computer vision and machine transfer learning domains. Section 3 details how positive machine transfer learning was enabled through digital twin alignment and annotation automation of synthetic training data. Experiments designed to evaluate pipeline performance are outlined in Sect. 4, followed by experimental results in Sect. 5. Finally, Sect. 6 discusses closing thoughts and future work.

2 Related works

The essence of relative navigation can be summed up as a vehicle's ability to track the pose of another vehicle or object relative to itself [58]. Without relative navigation, autonomous docking is not possible. So, where does this ability come from? In short, all forms of relative navigation between vehicles fall under one or more of the following three categories: signals (e.g., GPS), inertial, or vision-based navigation [84]. Of the three, vision is the only one impervious to interference, jamming, and drift, making it exceptionally useful in autonomous docking operations. Thus, a vision-based approach is the primary focus of this study.

2.1 Traditional vision solutions

Simple vision-based docking methods have been successfully implemented in controlled environments. For example, Wu et al. [87] proposed a docking approach for self-changeable robots using a combination of three primary techniques: YOLOv3-based object detection and tracking for pre-positioning, laser ranging for depth and precise angular alignment, and infrared emitter and receiver control loop for lateral alignment. While this method performed well for guiding a robot to a stationary target in a controlled 2D plane, it remained limited to relatively simple scenarios. These rudimentary methods do not address the complexities of more realistic dynamic environments, such as those encountered in aerial refueling, where docking must be achieved in the face of rapid movements, turbulence, and unpredictable external factors.

More complex vision solutions such as stereo vision [60, 92], light detection and ranging (LiDAR) [7, 16], and structured light [82] compute 3D point clouds directly from sensed data. Despite producing rich spacial information, these approaches are often slow and significantly limited by their sensitivity to environmental conditions, high computational requirements, inability to handle occlusions, and need for complex extrinsic calibrations [84].

Nearly all other vision-based solutions today take advantage of projective geometry from the pinhole camera model, i.e., vectors projected through a common focal point and onto an image plane [76]. In particular, well established algorithms like perspective-n-point (PnP) can convert intrinsic camera parameters, 3D points, and corresponding 2D image projections into accurate 6° of freedom (6DoF) pose estimates [26]. However, accuracy of the algorithm's output is entirely dependent on its input. As long as the physical properties of the camera remain fixed, accurate camera intrinsic parameters can be established through a simple chessboard calibration. Similarly, accurate fixed 3D object points can be established through methods such as total station surveying [30], 3D scanning [17], and motion capture [57]. Unfortunately, the third and final input is not as easy to establish. Current solutions lack the ability to accurately, precisely, and reliably localize corresponding 2D image points in real-time. Hence, image feature detection remains the root problem in most vision-based approaches.

Several researchers have successfully automated feature detection through more traditional solutions like infrared LED blob detection [8], Harris corner detection [6], VisNav [21], Speeded Up Robust Features (SURF), Scale-Invariant Feature Transform (SIFT), fiducial marker detection, HSV color segmentation, and active contouring [22]. To establish a 2D to 3D point correspondence, many of these also require a separate algorithm, like mutual nearest point [6], classical assignment model, perspective transformation based matching [89], maximum clique detection [49], and the Munkres algorithm [23]. Although various combinations of these methods have unique

benefits and can produce predictions with minimal error in controlled environments, they also come with significant drawbacks. For example, LED detection approaches enable operations in low lighting, but also require power to each beacon, are susceptible to over saturation due to light pollution and ambient interference, and most are not resilient to occlusions and missing features. Moreover, traditional solutions are inherently deterministic and do not generalize well in varied conditions. For instance, Noh et al. successfully automated the docking and undocking of an electric vehicle to a charging terminal using ArUco marker detection, but their approach also required optimal lighting for adequate marker visibility [54].

2.2 Machine learning

To overcome these limitations, the computer vision community has increasingly turned to machine learning (ML) techniques. Some have proposed ML solutions that consolidate feature detection, feature matching, and pose estimation into a single network architecture; notable examples include DFPN-6D [9], YOLO-6D+ [41], SSD-6D [42], SMOPE-net [46], BB8 [63], YOLOv8-PoseBoost [79], PoseCNN [88], and a few others [37, 73, 75, 95]. Deploying object detectors for feature detection and matching tasks separate from pose estimation is also common; examples include CenterNet [96], keypoint R-CNN [34], and SuperPoint [20]. The above ML models tend to perform well and generalize across popular datasets like ImageNet [19], PASCAL VOC [25], KITTI [27], LINEMOD [35], T-LESS [36], and COCO [47]. However, they lack deployment in real-world autonomous docking operations, primarily because comparable training datasets unique to the domain do not exist and are cost-prohibitive or impractical to create.

2.3 Current annotation methods

The most important aspect of generating such datasets is establishing accurate annotations, which has a direct impact on model performance [59]. Previous attempts at automating annotations in real imagery have yielded marginal results. For example, Kiyokawa used fiducial markers to fully automate image annotations, but struggled to reduce error from models confusing the markers as part of the object [44]. Similarly, Hammarkvist's attempt at automation through interpolation resulted in decreased model performance [32]. Thus, most reliable datasets today are generated manually. Unfortunately, manual image annotations are costly, cumbersome, error-prone, and time-consuming [65, 90].

2.4 Motivation for transfer learning

Given the challenges in acquiring large annotated datasets in real-world autonomous docking scenarios, transfer learning from synthetic imagery is emerging as a potential solution. By leveraging synthetic data, researchers can rapidly generate precise 3D ground truth and corresponding image projections in simulation, allowing for cost-effective and error-free image labeling automation. This method has already shown promise when applying object detection to pedestrians [33], drones [67], and common refrigerator items [64, 74].

The motivation for using transfer learning in this study arises from the difficulty of creating comparable real-world datasets for autonomous docking, particularly in dynamic environments such as aerial refueling. Synthetic data enables models to be pre-trained on large, varied datasets, which would be prohibitively expensive or impractical to gather manually. These pre-trained models can then be fine-tuned on a smaller set of real-world data, significantly reducing the need for extensive real-world data collection while still achieving high performance. Modern synthetic imagery, which can achieve a level of realism that deceives even the human eye [48], further enhances the effectiveness of transfer learning by improving model generalization from synthetic to real-world conditions.

2.5 Improving Tran's methods

Tran recently attempted real vision-based autonomous docking using transfer learning from synthetic imagery [77]. He used 3D scanning to establish a realistic digital twin of a small satellite, trained YOLO models to find 12 unique features on the satellite using solely virtual imagery generated from the digital twin and attempted PnP pose estimation on resulting YOLO predictions. Though reporting as low as 1.2 cm and 1.22° of position and rotation error respectively at contact in virtual imagery, the same models failed to find enough features for pose estimation in real imagery—an obvious indicator of failed transfer learning.

Several improvements were made to extend Tran's work. First, studies have shown that including real and higher quality synthesized imagery in training datasets consistently improves model performance, positive transfer learning, and overall detection accuracy [55]. To improve image quality for this effort, a novel annotation automation technique was used that labels real images captured in a lab using mocap data. A professional-grade 3D scanner was also used when generating the digital twins, as opposed to Scaniverse on an iPhone. Some other notable experimental design decisions differing from Tran included: increasing

YOLO input resolution from 640×640 to 864×864; increasing training dataset image count from 10K to 17.8K; increasing training epochs from 200 to 2000 and patience from 5 to 30; and decreasing YOLO model size from YOLOv5m to YOLOv5s. Assumptions in Tran's work imply YOLO can only find features that are unique and distinct to humans. This study challenged these assumptions by randomly selecting features regardless of uniqueness. Feature count was also significantly increased from 12 to 90. Lastly, instead of performing relative navigation from image sensor to vehicle (i.e., within camera's local reference frame), relative vectoring [84] was deployed to navigate one vehicle to another.

3 Methodology

The primary objective of this effort was to train YOLO to accurately find 3D object points in real-world imagery, ultimately enabling autonomous docking maneuvers through a technique called relative vectoring. Procuring adequate training data remains the most significant challenge. The data must be accurate and accessible. Real labeled imagery from operational environments has potential to produce the best model performance. However, such data are often difficult or impossible to obtain due to a lack of accurate truth data. Thus, this study proposes using the next best thing: machine transfer learning on synthetic data. In the absence of real operational data, will supervised machine learning on labeled imagery generated in a laboratory or virtual world suffice? Will resulting models generalize well when faced with real-world operational conditions?

To answer these questions, YOLO was trained with (1) labeled virtual imagery produced in simulation, (2) real imagery labeled using 3D re-projections in a mocap space, and (3) hybrid imagery that blends the two. Furthermore, various scene augmentation techniques were deployed to overcome generalization challenges associated with glare, poor lighting, low visibility, and other previously unseen environmental conditions. The remainder of this section describes the proposed machine transfer learning pipeline, as depicted in Fig. 1. This includes how annotation automation of synthetic images was implemented and how models were trained with resulting data to accurately predict 2D image points of 3D features across multiple objects.

3.1 Experimental objects

In this work, objects were defined as anything that has a local reference frame and can be "seen" by a vision sensor within its optical field of view. In doing so, PnP could be

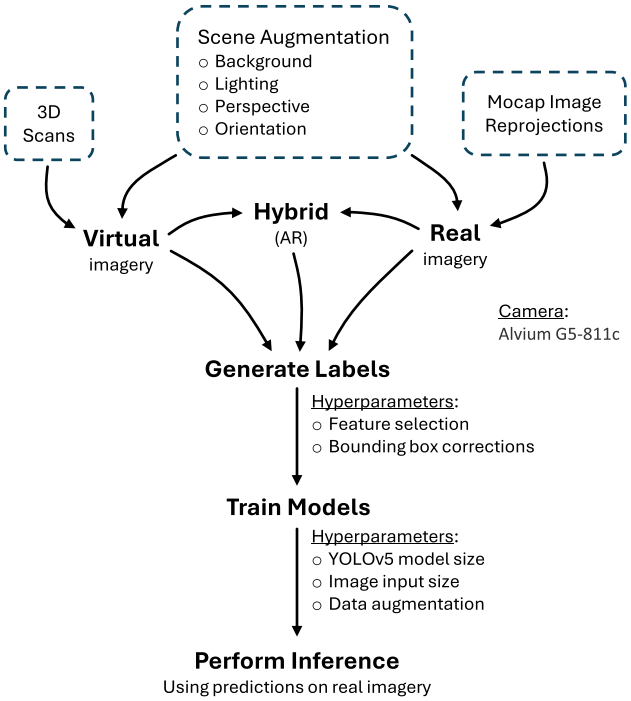

Fig. 1 Sim-to-real transfer learning pipeline

applied to sensed 2D image points and corresponding 3D object points to derive the pose (position and orientation) of each object relative to the camera. Furthermore, since all the estimated poses are defined in a common reference frame, i.e., that of the camera, they can also be represented in the local reference frame of any single object through simple reference frame transformations—this is how image-based relative navigation was enabled in this effort.

The objects of interest here are the two docking vehicles and vision sensor (i.e., camera) as well as the space they operate in. Typically, empty world space is not considered an object. However, it is considered one in this work since it has its own local reference frame and the camera can "see" it, meaning the 3D space can be localized relative to the camera and vice versa. This distinction later helps with digital twin alignment.

3.1.1 AAR use case

Despite this work having relevance in nearly all docking scenarios, autonomous aerial refueling (AAR) was chosen as the primary use case due to its complexity. Aerial refueling remains among the most complex docking maneuvers due to its rapidly changing and volatile nature. Docking a receiver probe into a tanker drogue mid-flight typically requires a highly trained and seasoned human pilot with fast reflexes. Furthermore, probe and drogue refueling, when compared to the boom and receptacle method, is often considered more challenging because the

flexible hose-drogue assembly has uncontrolled and unpredictable movements from turbulent wind effects [39]. Thus, if the proposed algorithm performs well on probe and drogue refueling, it will likely perform equally well on other docking maneuvers with less complexity and larger margins for error, e.g., parking a car inside a garage.

Within a mocap space, a full-scale refueling drogue basket was suspended and flared open within a wooden support structure, see top left of Fig. 2. The support structure was attached to a railed pulley system for vertical and horizontal articulation. All support structures were painted black, so they visually blended in with the background. A full-scale Learjet nose cone with attached refueling probe was also present, see top right of Fig. 2.[3] Wooden boards with hand-drawn instrument panels were attached to imitate what is commonly seen in an aircraft cockpit and a camera was mounted behind it, see Fig. 3. In a way, this mimics real aerial refueling operations while positioning the camera at a realistic vantage point.

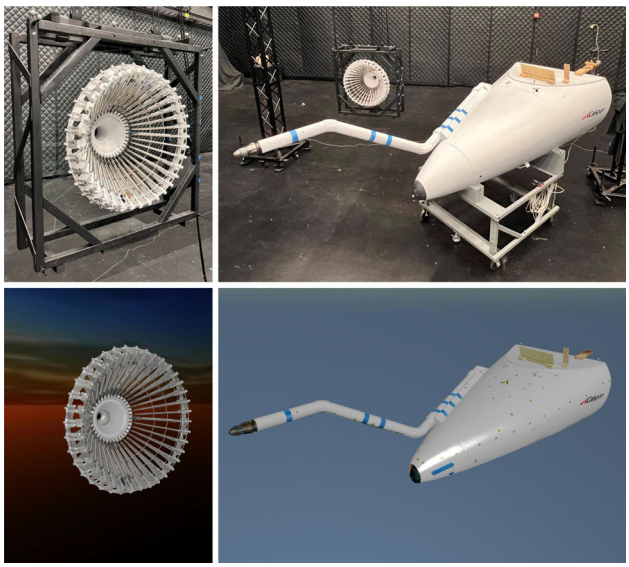

Fig. 2 Digital twins of refueling drogue basket (left) and Learjet nose with attached refueling probe (right)

3.1.2 Vision sensor

An Alvium G5-811c camera [2] served as the vision sensor for all real-world imagery collected in this effort. By default, it has a raw output resolution of 2848×2848 pixels which was resized—using OpenCV and bilinear interpolation—down to 864×864 in preparation for YOLO training and inference. Autoexposure was enabled and focus set to approximately 15 m in front of the camera. All real imagery in this effort came from this camera configuration. Additionally, a common chessboard camera calibration, i.e., Zhang's method [93], was used to correct distortion in images collected in the mocap space; resulting intrinsic camera parameters are listed in Table 1.

3.1.3 Digital twins

Generating virtual imagery necessitated the use of digital twins. In this effort, 3D models of the probe and drogue were created using an Artec Leo 3D scanner [3]. With 0.2 mm 3D resolution and 0.1 mm 3D point accuracy, the scanner generated highly accurate 3D models with rich textures, exportable to OBJ format, see Fig. 2. We subsequently imported these models into a highly customizable virtual world rendered using the AftrBurner graphics engine [56]. In this virtual world, a virtual camera was modeled with a resolution and horizontal field of view similar to that of the Alvium G5-811c. However, instead of modeling the distortion coefficients listed in Table 1, virtual images were produced with no distortion and corresponding real images were undistorted during a digital twin alignment. In addition to the probe, drogue, and camera digital twins, the virtual world coordinate frame was also aligned to that of the mocap space, which occupied a room approximately 20 m long, 16 m wide, and 6 m high. These digital twins accurately modeled the real world and enabled realistic synthetic image generation.

3.1.4 Digital twin alignment

Aligning real-world objects to their digital twins is a trivial task when using a mocap system—just set the pose of each digital twin to its corresponding real-world object pose as reported by a mocap system. Unfortunately, this simple method does not translate directly to minimizing re-projection error in corresponding imagery, especially when dealing with distortion effects. One method found in the literature is to perform an extrinsic camera calibration to establish an optimal transformation between the camera reference frame tracked by a mocap system and the true optical camera reference frame [5, 11, 66]. Regrettably, solving for a fixed transformation will not consistently preserve alignment with noisy mocap estimates. Despite state-of-the-art technology, the most expensive, highest-resolution, and well-calibrated mocap systems still generate noisy data—in which even the slightest positional and rotational deviations from truth can result in significant re-projection errors. For example, the Prime^{X} 41 OptiTrack cameras used in this work have a rotational accuracy of ±0.5° [57]; this has potential to result in over 13 cm in

[3] For simplicity, the entire Learjet is considered an extension of the probe, e.g., features found on the Learjet nose cone are simply referred to as probe features.

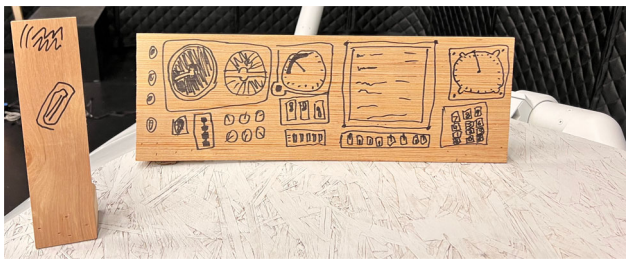

Fig. 3 Feature-rich hand-drawn cockpit dashboard

positional error across the length of the mocap space, see Fig. 4.

In this effort, a slightly different approach was used. Instead of applying a single optimized transformation across all images, the variance in mocap precision between frames was overcome by computing a unique transformation for each image using PnP. To enable this capability, a field of known 2D to 3D correspondences were established by positioning fiducial targets throughout the mocap space, see Fig. 5. The 36h11 AprilTag family was used and the mocap system was configured to report the 3D position of each tag's 2D center. Automating 2D tag center localization in imagery came from computing the intersection of diagonals between AprilTag corners as depicted in Fig. 6.

With 2D tag center points and corresponding 3D mocap space coordinates, PnP enabled automation of camera pose localization relative to the space itself. In this effort, OpenCV's solvePnPRansac method was used. In addition to 2D to 3D point matches, the intrinsic camera parameters listed in Table 1 were passed in, extrinsic guess was disabled, and the iterations count, re-projection error threshold, and confidence threshold were set to 500, 4.0, and 0.9999, respectively. The subsequent output of this method was a 6DoF pose estimate in the form of a Rodrigues rotation vector, *rvec*, and a *z*-forward translation vector, *tvec*. These were used to set the virtual camera pose, effectively aligning image re-projections of digital twins in both real and virtual imagery. The diagram in Fig. 7 depicts a generalized overview of this technique.

3.1.5 Reference frames

How can a machine move in a specific direction to avoid an obstacle or dock with another object without a sense of direction? Without well-defined reference frames, coordinating complex maneuvers among multiple objects is impossible. Thus, the last important aspect of establishing the experimental objects was defining their local reference frames. They were defined here as 3D Euclidean spaces with *x*-axis facing forward, *y*-axis to the left, and *z*-axis up. Furthermore, the origins of the receiver aircraft and refueling drogue were defined as the probe tip and refueling coupler centers, respectively, see Fig. 8. Probe-to-drogue (PtD) contact is made when these two origins coincide, i.e., the PtD vector magnitude is zero. For the camera, OpenCV's standard convention was used: *z*-axis forward (aligned with optical axis), *x*-axis to the right, and *y*-axis down. In pixel space, the image origin was defined as the top leftmost pixel with *x*-axis right and *y*-axis down.

Table 1 Intrinsic camera parameters inside mocap space

Resolution	864 × 864 pixels
Horizontal FOV	∼49.375°
Radial dist. (k_1, k_2, k_3)	−0.192, 0.153, −0.042
Tangential dist. (p_1, p_2)	1.46e-4, −5.55e-4
Optical center (c_x, c_y)	431.80, 422.61 pixels
Focal lengths (f_x, f_y)	939.34, 939.67 pixels

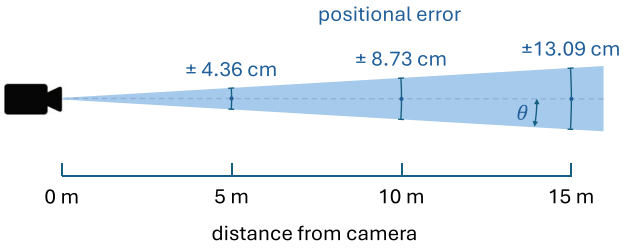

Fig. 4 Rotational error of $\theta = \pm 0.5°$ potentially equates to over 13 cm in positional error when projected out to 15 m from the camera

3.2 Labeling techniques

Now that the experimental objects and corresponding coordinate frames are defined, a naïve approach could be to consider collecting imagery next. However, accompanied labels are a prerequisite to supervised machine learning. Moreover, automation is crucial since manual post-capture labeling is laborious, expensive, slow, and error-proned [91]. Furthermore, high performing models require large training datasets. In this work, three different labeling techniques are proposed for generating such datasets: (1) reusable manual labeling of real probe images, (2) virtual projection labeling of simulated drogue images, and (3) mocap re-projection labeling of real and hybrid drogue images.[4]

3.2.1 Reusable manual labeling

As previously mentioned, manually labeling objects in large image datasets is largely untenable, except when the

[4] In the context of this work, hybrid refers to imagery of a real drogue placed in a virtual scene through the use of augmented reality.

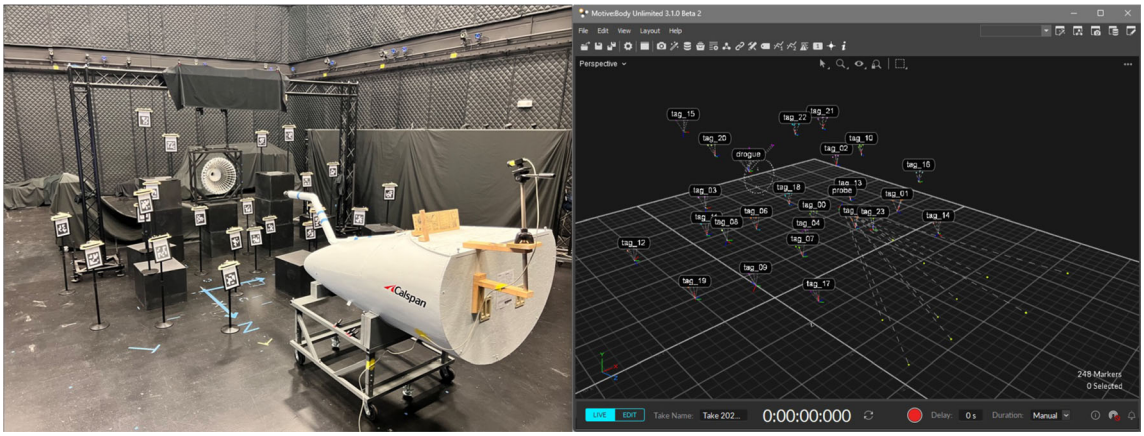

Fig. 5 AprilTags used for digital twin alignment (left) with corresponding mocap data generated by OptiTrack (right)

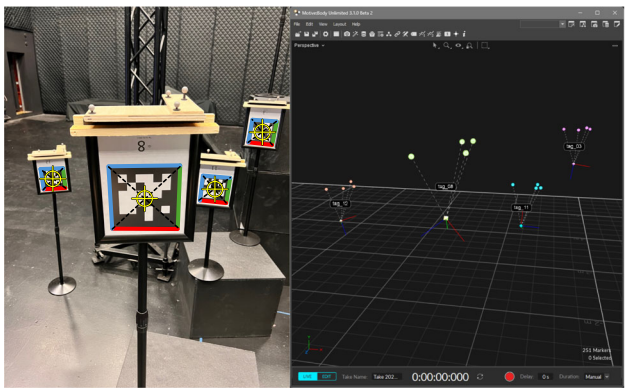

Fig. 6 AprilTag 2D centers found in imagery (left) with corresponding 3D centers reported by OptiTrack (right)

objects in the images are stationary. A stationary object, such as the vehicle to which the camera is attached, requires only one set of labels. These labels can then be reused across all other images containing the same stationary object. This labeling method is particularly advantageous because it can be applied to real operational imagery, thereby reducing the need for generating synthetic data. However, this raises a crucial question: How does one manually label 2D features in imagery that correspond to 3D object points?

To answer this question, it is helpful to work backward starting with the 3D points. First, the 3D positions of probe features visible in the image were surveyed across the probe's digital twin, (see left side of Fig. 9). Next, these 3D local space points (probe reference frame) were transformed into world space points (mocap reference frame) using the probe pose reported by the mocap system. Leveraging the camera pose, i.e., *rvec* and *tvec*, computed during the digital twin alignment, and intrinsic parameters from Table 1, OpenCV's projectPoints method subsequently transformed these world space points into image space as 2D point projections (see right side of Fig. 9). Finally, to establish YOLO labels for these 2D points, each were encapsulated within a small bounding box, centered precisely on the point itself, as illustrated in Fig. 10.[5] Although manually labeling a single image in this manner was slow and cumbersome, the upfront cost ultimately translated to accurate Learjet training label automation for hundreds of thousands of images, rendering it highly efficient in the long run. Unfortunately, labels could only be reused if corresponding object features remained stationary within the image. Labeling a moving drogue required a different approach.

3.2.2 Virtual projection labeling

Images contain perspective distortion, resulting in different bounding box sizes and aspect ratios for the same object depending on its position on the image plane [12]. To account for this distortion during drogue label automation, drogue features were randomly selected, and 3D points were surveyed around each feature. Each feature was thus defined by a 3D point cloud, with the geometric centers of these point clouds serving as the 3D points that YOLO was trained to detect. In simulation, these 3D points, along with extrinsic and intrinsic camera parameters (such as position, orientation, and the K-matrix), are precisely known, making the projection of points onto the camera's image plane a straightforward task using OpenCV's projectPoints method.

Next, the tightest fitting bounding box surrounding each 2D projected feature point cloud was computed. The sides of each bounding box were then expanded until their centers aligned with corresponding 3D geometric feature centers projected onto the image plane. These resulting bounding boxes were saved as YOLO training labels for

[5] Arbitrarily sized to approximately 1–5% of the image width and height.

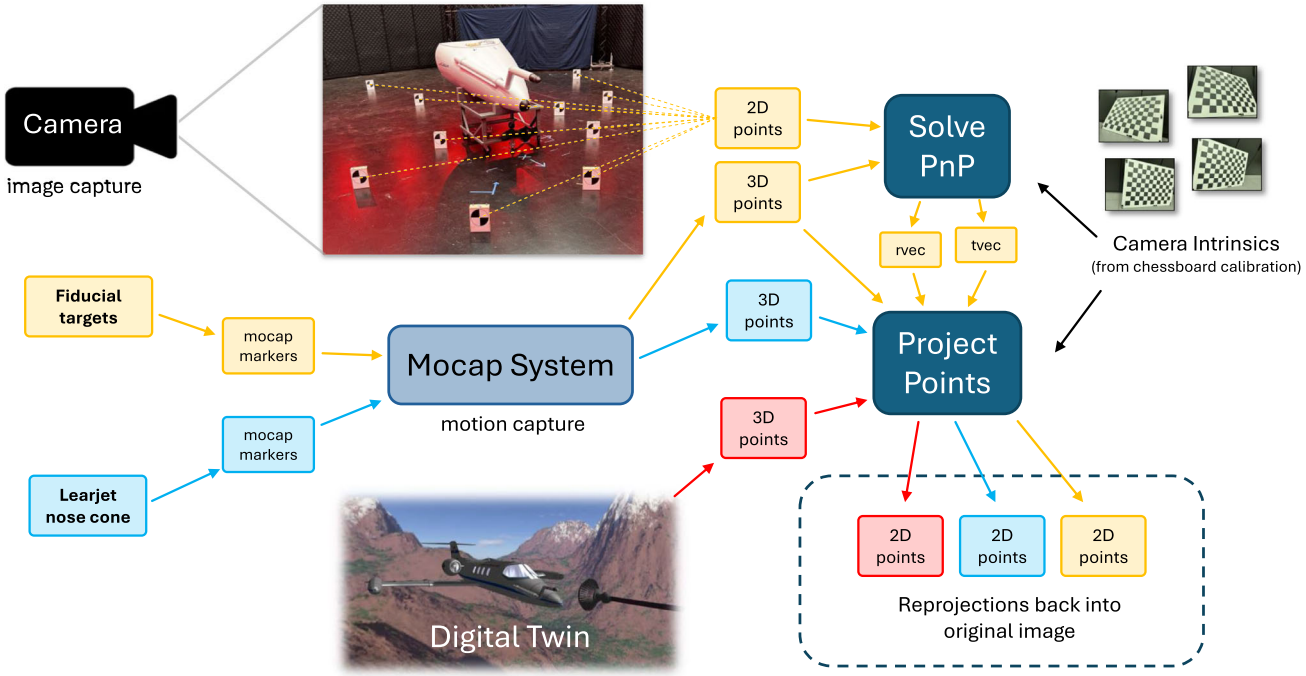

Fig. 7 During digital twin alignment, fiducial target detection software reports the 2D image locations of specific target points (top middle) while a mocap system reports their corresponding 3D locations within the mocap space (yellow center path). PnP subsequently converts the 2D to 3D point matches to an optimized transformation between mocap space and the camera's true optical local reference frame (*rvec* and *tvec*, top right). Finally, the transformation is used to project both real and virtual points (blue and red bottom paths) back into the original image. Note that reprojection error can be measured directly by computing the difference between the 2D points passed into Solve PnP and the corresponding output of Project Points

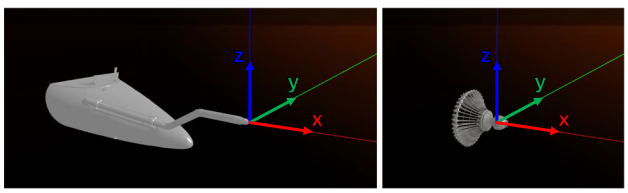

Fig. 8 Local reference frames

each virtual image. Repeating this process in Monte Carlo simulation allows for the generation of unlimited training data that is precisely labeled, error-free, inexpensive, and quick to produce. For more details on this process, refer to [84].

3.2.3 Mocap re-projection labeling

High-resolution digital twins offer significant potential for successful machine transfer learning. However, real imagery introduces a level of realism that even advanced rendering techniques like ray tracing and Phong illumination [61] cannot fully achieve. Therefore, this work also explored training YOLO with real labeled imagery. Using a method similar to virtual projection labeling, the labeling of real lab-generated images was automated by utilizing the 3D truth of predefined drogue features. However, instead of

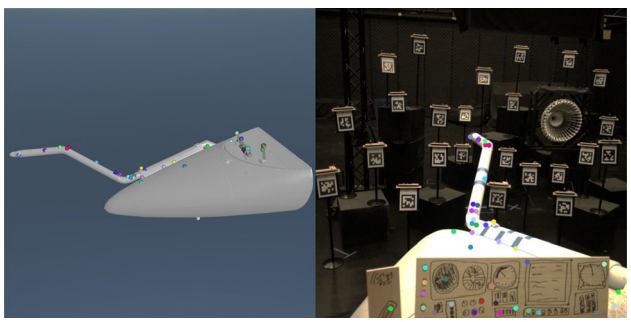

Fig. 9 Manually surveyed 3D local frame points from digital twin (left) and their 2D projections in real imagery (right)

projecting simulation-provided 3D truth into virtual imagery, digital twin alignment was used to re-project mocap data into real imagery (see Fig. 11). This approach ensured valid labels for both the real images and their aligned virtual counterparts, a benefit that was also leveraged to label hybrid images.

3.3 Hybrid drogue imagery

Although YOLO, with sufficient training data, can learn the underlying structure of a real drogue from real images captured in a controlled environment, evidence throughout

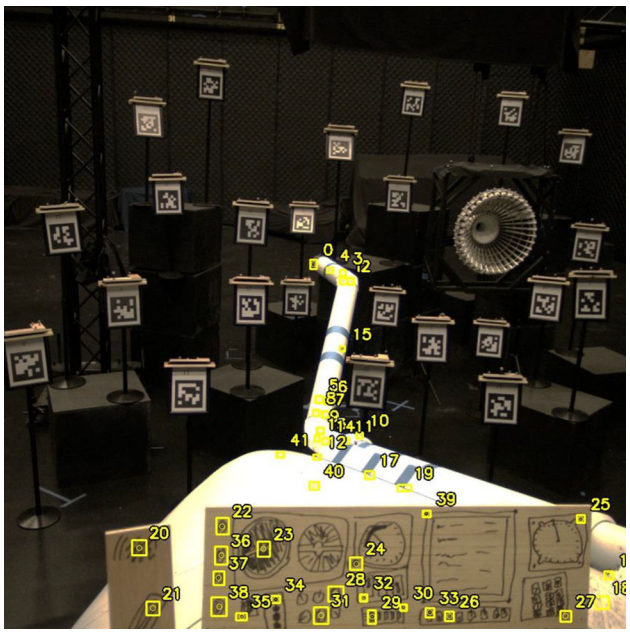

Fig. 10 Manually generated reusable YOLO labels of stationary probe features

the literature suggests contextual information can improve object detection generalization and performance on unseen data, such as images of the same real drogue flying in a brightly lit sky [1, 13, 24, 28, 43, 47, 80, 94]. To explore this, a form of augmented reality was adopted that replaces the laboratory scene surrounding the real drogue (e.g., lab walls and flooring) with virtual scenes containing more realistic context (e.g., clouds in the sky). Specifically, chroma keying was used by following Smirnov's OpenCV implementation of the Vlahos algorithm [72]. However, merely adding a green screen backdrop proved insufficient for hiding the AprilTags and drogue support structure without residual artifacts or significant manual post-processing. Instead, a reverse green screen approach was employed, setting the drogue's digital twin to a uniform green. This enabled automated replacement of the virtual drogue with a real one using chroma keying, as depicted in Fig. 12.

3.4 Scene augmentation

As previously mentioned, object detectors can benefit from contextual information. In addition to removing the unrealistic laboratory scene in hybrid imagery, it is important to incorporate common contextual elements present in real operational environments, such as variations in background, lighting, and perspective. To account for these factors and improve YOLO's generalization and performance, various scene augmentation techniques were explored.

Backgrounds In virtual reality simulation, skyboxes are commonly used to create the illusion of distant 3D surroundings [29, 40]. This approach was employed to vary the backgrounds of virtual imagery. By using 32 different skybox patterns, a wide range of environments were simulated, including clouds, overcast, mountainous terrain, oceans, vibrant sunsets, starry nights, and clear skies. Each virtual image was assigned a randomly selected skybox, which was also randomly rotated to enhance YOLO's ability to generalize across different banking maneuvers.

Lighting Real aerial refueling operations can occur at any time of day *or night*, and factors such as aircraft heading, attitude, altitude, and weather conditions can create unique lighting effects. To anticipate glare, shadows, reflections, and reduced visibility during real-world operations, both virtual and real light augmentation techniques were implemented. In simulation, the position and orientation of the virtual light source were randomized between image captures. In the lab, overhead lights were toggled on and off, and portable work lights were randomly directed at the probe and drogue from different angles, as shown in the left image of Fig. 12.

Perspective To effectively generalize detection of moving 3D drogue features in 2D imagery, YOLO must learn what such features look like from different perspectives [12]. For instance, these features grow and shrink in imagery as distance between the drogue and camera varies, and they undergo perspective distortion as they move across the image plane. To train YOLO on these different perspectives in the lab, a camera was attached to a tripod, simulating its mounting in the Learjet cockpit. During image collection, the drogue was articulated in three dimensions within the camera's field of view, maintaining a realistic orientation by pointing in the direction of estimated airflow with zero roll, pitch, and yaw.

Matching these images to corresponding mocap data requires high-precision time alignment between the camera and mocap system—a capability that was not available in this work. To mitigate this challenge, data was collected only while the drogue was stationary. Imagery with corresponding mocap data was captured for 278 random stationary drogue positions evenly distributed throughout the camera's viewing frustum, at a range of approximately 4.5 to 17 m. At each position, approximately 130 images were collected at about 5 fps while performing lighting augmentation. This process generated approximately 36,000 unique images of a real drogue—each containing 65 precisely labeled drogue features captured from various perspectives in less than four hours.

In simulation, a similar process was followed. Prior to each image capture, the drogue was randomly placed within the same range in the camera's viewing frustum, with background and lighting augmentation. Without the

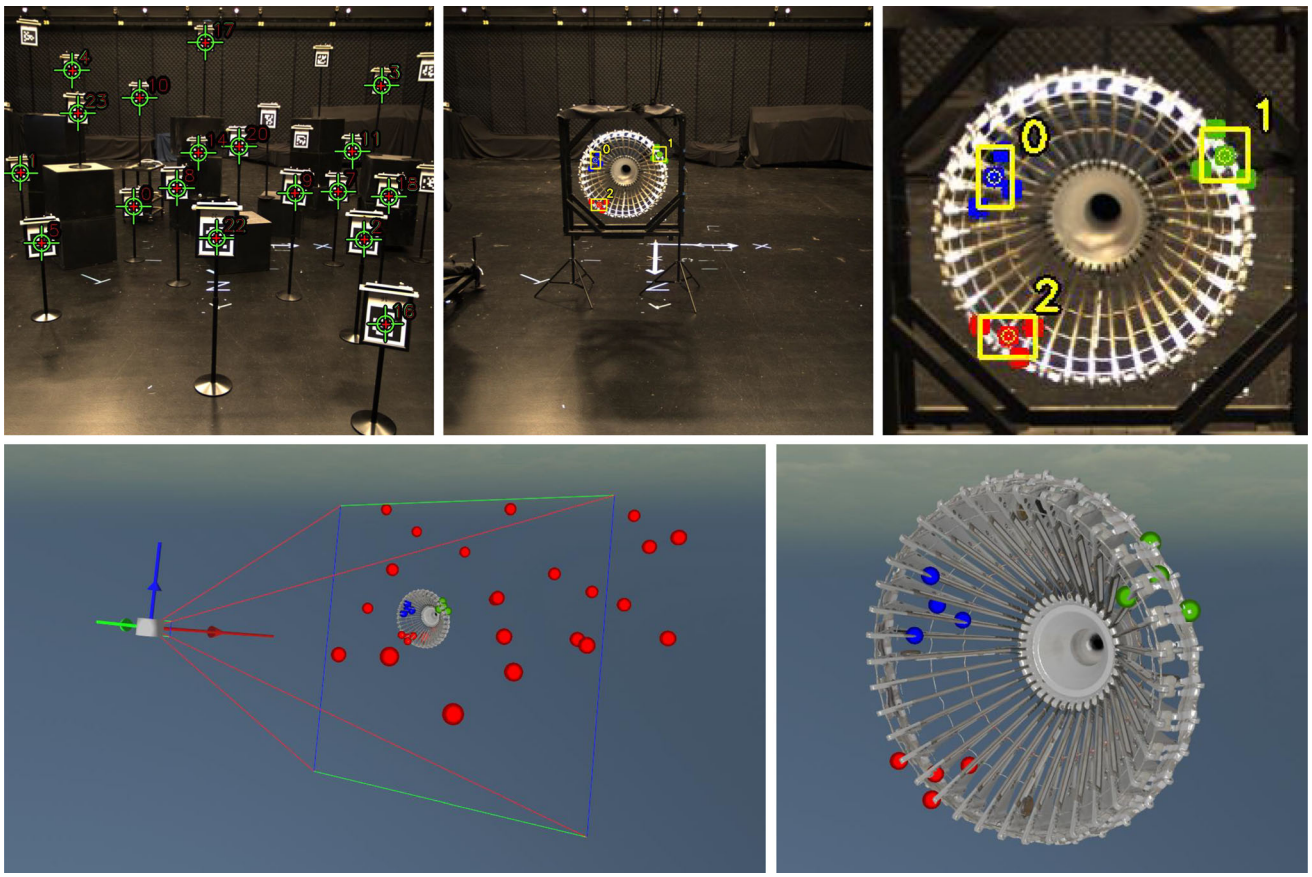

Fig. 11 Mocap re-projection labeling begins with a digital twin alignment, in which an image is captured of targets distributed throughout the mocap space (top left). PnP subsequently converts their 2D and 3D positions—as reported by AprilTag and OptiTrack software respectively—into a camera pose relative to the targets; this aligns the 3D mocap space to the camera's optical local reference frame (bottom left). Note that a digital twin alignment remains valid as long as the camera remains stationary, thus allowing for reuse of the alignment on the drogue after removing the targets (top middle). Finally, the positions of predefined feature point clouds, also reported by OptiTrack (bottom right), were reprojected into the camera's image plane, enabling accurate annotation automation (top right)

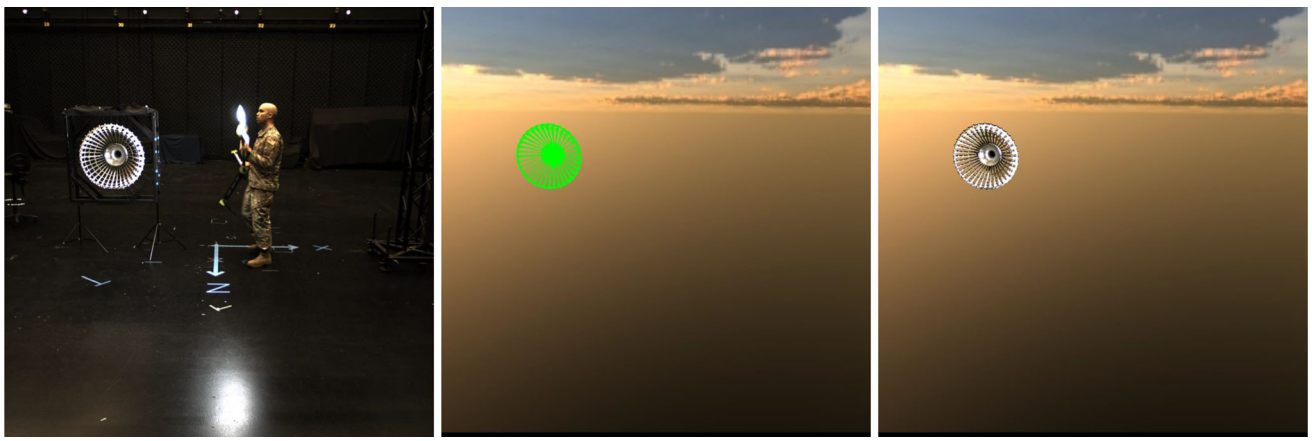

Fig. 12 Chroma keying a real image (left) behind a reverse green screened digital twin (middle) resulted in a hybrid image (right) depicting a real drogue suspended within a virtual scene

Table 2 Platform configuration

Environment setting	Version/specification
Computer (laptop)	Dell Precision 7680
Operating system	Windows 10 (64-bit)
Processor	Intel Core i7-13850HX
Memory (RAM)	64GB
Disk storage	1TB
GPU	Nvidia GeForce RTX 4090
Nvidia CUDA	v12.1
OpenCV (with cuDNN)	v4.8.0
OpenGL	v4.3
YOLOv5 model format	.onnx, exported from .pt

limitations of time alignment, the simulation allowed for imagery generation from significantly more perspectives than possible in the laboratory. This method resulted in an unlimited source of unique and precisely labeled virtual images, generated at approximately 21 fps on the platform listed in Table 2.[6]

3.5 Multi-object image merging

The labeling and scene augmentation techniques proposed thus far are object-dependent, focusing on either the probe or the drogue individually. However, real operational imagery will contain both objects. To accurately represent this realistic context, a method was needed to merge the two image sources. Fortunately, since the stationary probe represents the foreground in each image, a simple gradient alpha mask was applied to achieve the merger.[7] Similar to manual reusable labeling, the benefit of applying a single mask to all probe images outweighed the initial cost of manually generating it. Figure 13 illustrates this masking process, where randomly selected real lab-generated images of the probe were merged with virtual, hybrid, and real images of the drogue to populate the final training datasets.

3.6 Handling occlusions

In previous studies [12, 84], it was demonstrated that training on occluded images can enhance YOLO model performance and enable a capability similar to object permanence or "x-ray vision." Consequently, in this research, all feature labels were retained regardless of their occlusion status in the image. However, training YOLO to

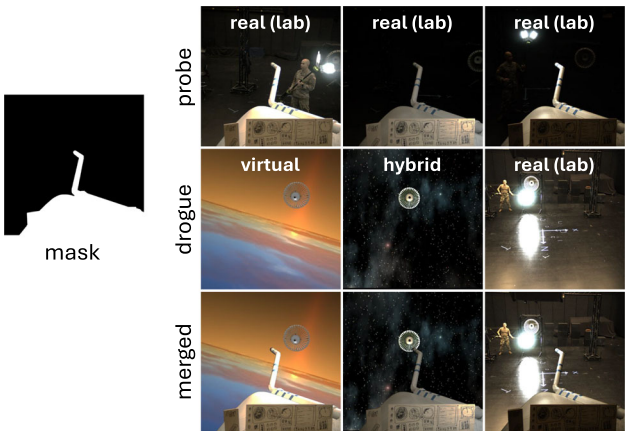

Fig. 13 Probe and drogue images were merged using a single-channel gradient alpha mask (left). In this mask, white represents pixels retained from probe images (top row), while black represents pixels retained from drogue images (middle row). The bottom row displays the resulting merged imagery used to train YOLO models

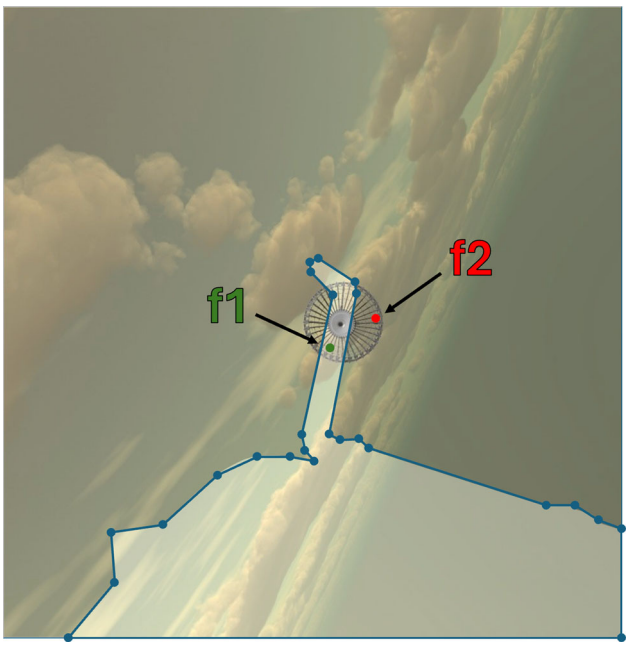

Fig. 14 Automated occlusion detection was achieved by defining a polygon with vertices along the outer perimeter of the probe's silhouette (blue) and determining whether features reside within the polygon. In this image, drogue feature f1 is deemed occluded by the probe, as it falls within the polygon, whereas f2 is not

detect a fully or mostly occluded drogue presents challenges, including potential confusion for the model or an increase in false positives. To address this challenge, an occlusion threshold was established, leading to the discarding of any labeled images in which more than half of the drogue features were occluded. Figure 14 illustrates the

[6] This platform was also used for all benchmarks and experiments reported throughout the remainder of this paper.

[7] The merging process also included a merger of corresponding labels.

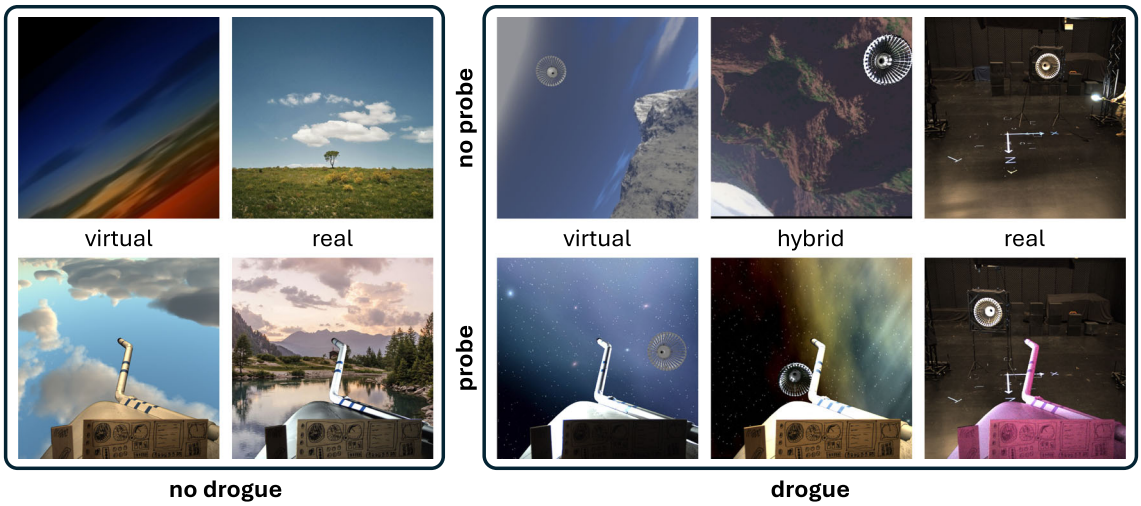

Fig. 15 Examples of ten training data image types used in this work. Recall that hybrid images contain a real drogue chroma keyed on to a virtual background; thus, no distinction is made for hybrid images without a drogue

use of a point-in-polygon algorithm to automate this thresholding process.

3.7 Supervised machine learning

Utilizing the methods described above, training datasets were generated, comprising various combinations of ten image types: virtual, hybrid, and real images with and without the probe, as well as with and without the drogue, as depicted in Fig. 15. For real imagery containing the drogue, raw images captured in the laboratory were utilized. In contrast, real images without the drogue were randomly selected from unsplash.com search results using the keywords sky, landscape, and land. Each training dataset consisted of 5% images containing only a probe, 5% only a drogue, 5% neither probe nor drogue, and 85% both probe and drogue. Additionally, each training dataset was populated with approximately 17,800 images and underwent an 80/20 train-validation split.

All training tasks commenced from the YOLOv5s pretrained checkpoint, running for 2000 epochs with early stopping and a patience of 30, employing a batch size of 16, and utilizing an image input size of 864. During inference, YOLO predictions were filtered by applying objectness, class probability, and non-maximum suppression thresholds of 0.200, 0.250, and 0.200, respectively. Unique features without duplicates were defined, and YOLO outputs were further filtered to retain only the single prediction with the highest probability per class.

3.8 Applying relative vectoring

Ultimately, the final step of the machine transfer learning pipeline, as depicted at the bottom of Fig. 1, involves

Table 3 Background data subsets (non-probe images)

src[1]	drg[2]	15,130	7565	5044	890	594	445	297
v	–	–	–	–	*j*	*o*	*p*	–
r	–	–	–	–	*k*	–	*q*	*u*
v	√	*a*	*d*	*g*	l	–	*r*	*v*
h	√	*b*	*e*	*h*	*m*	–	*s*	*w*
r	√	*c*	*f*	*i*	*n*	–	*t*	*x*

Numbered headers correspond to image counts per subset

[1]Data sources included virtual (v), hybrid (h), and real (r) images; see top row of Fig. 15 for examples

[2]All background images either contained or did not contain a drogue, but none contained a probe

Table 4 Training dataset compositions

Composition				5%	5%	5%	85%
ID	v	h	r	probe[1]	drogue	neither	both[1]
–r	0	0	1	*k*	*n*	*k*	*c*
-h-	0	1	0	*j*	*m*	*j*	*b*
-hr	0	1/2	1/2	*p*, *q*	*s*, *t*	*p*, *q*	*e*, *f*
v–	1	0	0	*j*	*l*	*j*	*a*
v-r	1/2	0	1/2	*p*, *q*	*r*, *t*	*p*, *q*	*d*, *f*
vh-	1/2	1/2	0	*j*	*r*, *s*	*j*	*d*, *e*
vhr	1/3	1/3	1/3	*o*, *u*	*v*, *w*, *x*	*o*, *u*	*g*, *h*, *i*

Specified data subsets *a* through *x* (from Table 3) served as the background images for each training dataset

[1]Random probe foregrounds were merged with drogue backgrounds for the probe and both data subsets

Fig. 16 Learjet 25 receiver (left) and Gulfstream III tanker (right)

Fig. 17 The top left image depicts the Learjet 25 receiver aircraft retrofitted with a refueling probe and cockpit-mounted camera. The portion of its nose seen by the camera was 3D scanned (bottom left) and a digital twin alignment was performed with a mobile mocap system for accurate feature point re-projections and reusable manual labeling (right)

Table 5 Intrinsic camera parameters behind cockpit windshield

Resolution	864×864 pixels
Horizontal FOV	∼58.096°
Radial dist. (k_1, k_2, k_3)	−0.088, −0.098 0.168
Tangential dist. (p_1, p_2)	0.032, 0.011
Optical center (c_x, c_y)	535.38, 489.68 pixels
Focal lengths (f_x, f_y)	963.95, 963.80 pixels

YOLO performing inference on real imagery. Transfer learning is considered successful if YOLO, when trained on synthetic imagery, accurately predicts bounding boxes centered on corresponding trained features in real imagery.

These accurate 2D center predictions, along with corresponding 3D model points, can subsequently feed PnP to produce accurate 6DoF pose estimates of the probe and drogue relative to the camera. Relative vectoring, a technique described in detail in [84], then transforms these pose estimates into an estimated position of the drogue relative to the probe.[8] Finally, these accurate probe-to-drogue (PtD) vectors serve as visual perception for autonomous agents, enabling them to automate synchronized docking maneuvers.

[8] Relative vectoring omits the orientation component of the target vehicle when transforming both camera frame vehicle poses into a relative position. However, the orientation component can be easily included in the transformation if needed, resulting in a relative 6DoF pose (instead of merely a 3D position).

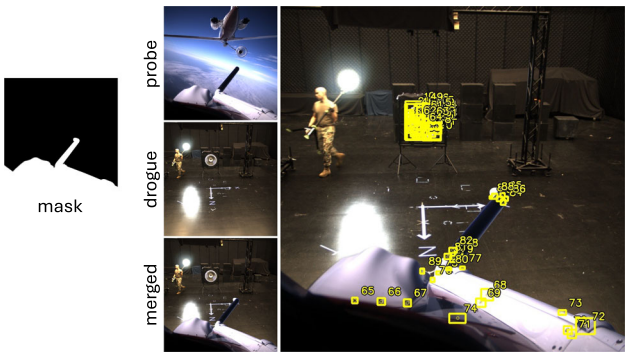

Fig. 18 Image merging using alpha gradient masking on the real Learjet is depicted on the left while resulting merged labels are depicted on the right. Note that the real drogue in the top probe image must be removed from training data due to the absence of corresponding truth labels

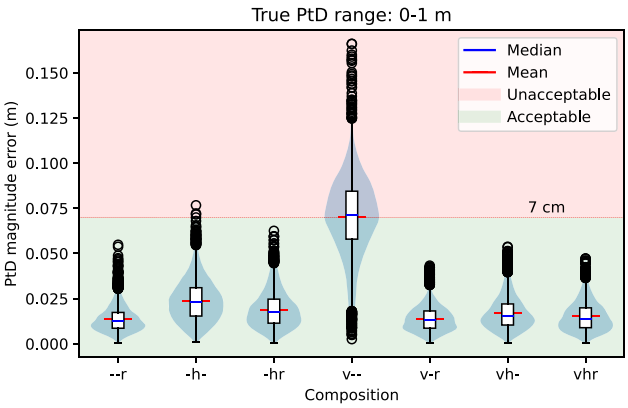

Fig. 19 Machine transfer learning combination training results—error at contact (0–1 m)

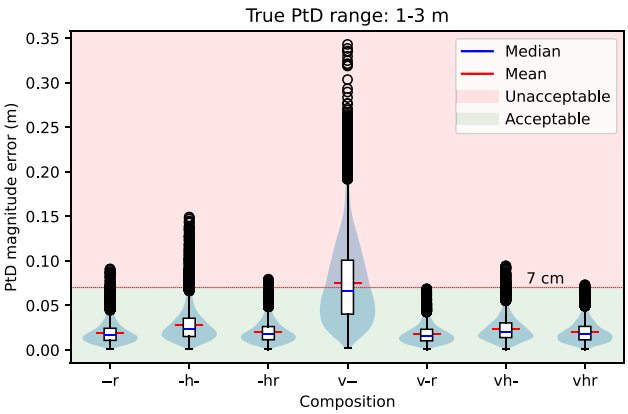

Fig. 20 Machine transfer learning combination training results—error at inner close range (1–3 m)

4 Experiments

The first two experiments were aimed at quantitatively validating successful machine transfer learning by comparing relative vector predictions from lab-collected

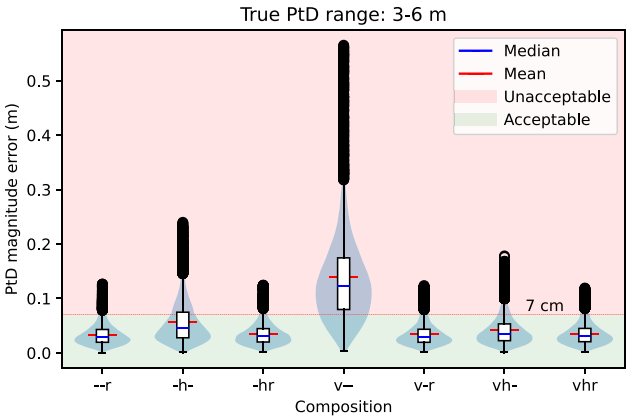

Fig. 21 Machine transfer learning combination training results—error outer close range (3–6 m)

imagery with corresponding mocap system truth. The third experiment involves collecting imagery onboard real flight tests to qualitatively assess pipeline performance and generalization in real-world conditions. This section provides a description of these experiments.

4.1 Combination training

The primary focus of this study was to enable accurate relative vectoring in real imagery following machine transfer learning from synthetic imagery. Thus, the objective of the first experiment was to determine the optimal composition of synthetic training data for generating the highest performing YOLO models. To assess the feasibility of positive transfer learning across the three data sources (v = virtual, h = hybrid, and r = real), separate YOLO models were trained for each of seven different data source combinations: –r, -h-, -hr, v–, v-r, vh-, and vhr. Model performance was evaluated as part of the relative vectoring pipeline. Each model was trained to detect 107 common features across the probe and drogue. Specifically, 65 point clouds were randomly selected on the drogue, with each point cloud containing 3–4 points. Feature point clouds were labeled using virtual projections and mocap re-projections, as described in Sect. 3.2. Additionally, the 42 manually labeled probe features from Fig. 10 were utilized.

Once generated, forming manageable subsets from each available data source permitted granular control over dataset compositions. Table 3 lists data subsets *a* through *x*, each containing a specified number of randomly selected images from one of the three data sources.[9] These subsets were utilized, in conjunction with randomly selected probe images, to populate the backgrounds and foregrounds of

[9] A uniform distribution was used when randomly selecting images to mitigate clumping and maximize drogue perspective diversity in training datasets.

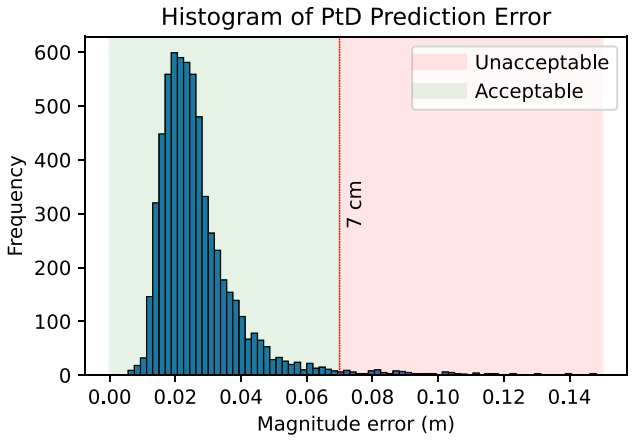

Fig. 22 PtD error during moving camera collect (relative vectoring with model v-r)

Table 6 Average pipeline execution time

Pipeline operation	Duration
Image loaded from file	13.602057 ms
YOLO inference	17.845157 ms
Dual solve PnP	0.000401 ms
PtD vector calculation	0.000210 ms
Total	31.447825 ms

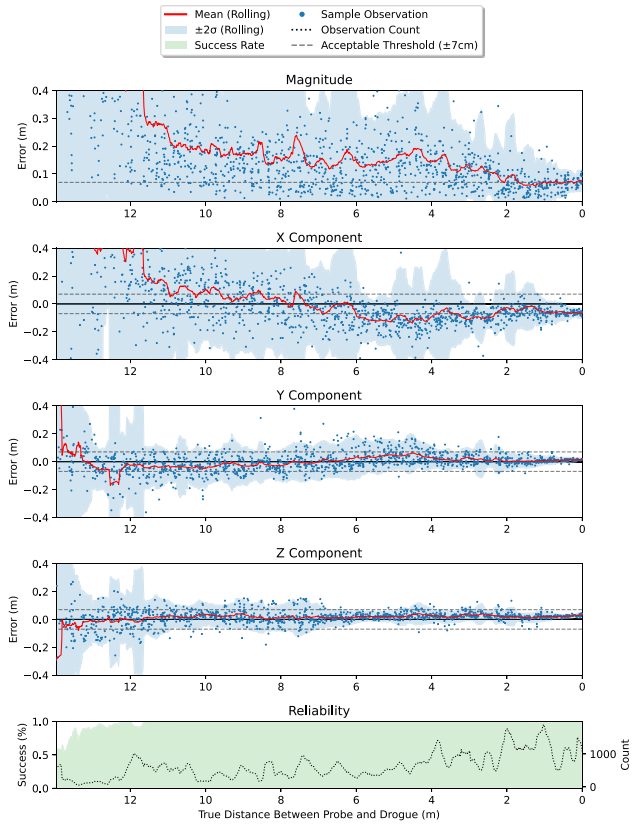

Fig. 23 Machine transfer learning training results for lowest performing model, composition v– (virtual only)

training datasets –r through vhr as detailed in Table 4. These datasets encompassed all possible combinations of the three data sources such that each source was evenly represented. Subsequently, a distinct YOLO model was trained and deployed for each training dataset, following the parameters and procedures outlined in Sect. 3.7.

To create a standardized test set for these models, a dataset containing 35,918 images with corresponding mocap truth was collected, depicting a stationary probe and a moving drogue. During continuous image collection, the drogue's position within the camera's field of view was slowly varied to capture a diverse range of perspectives and distances between the probe and drogue.[10] Subsequently, relative vectoring was performed with each model across the entire test set. Accuracy was defined as the magnitude difference between the predicted PtD vector and its corresponding mocap truth, with magnitude errors less than 7 cm at contact considered acceptable for autonomous aerial refueling operations. Execution speed was measured as average time in milliseconds per prediction and decomposed into individual pipeline operations. Furthermore, pipeline reliability was defined as the percentage of successful predictions returned over the total number of predictions attempted.[11] For instance, PnP requires at least three matches to predict a pose. Therefore, prediction attempts are considered unsuccessful when YOLO detects two or fewer features of an object in an image, since PnP cannot make a prediction in such cases.

4.2 Moving camera

In addition to evaluating relative vectoring performance using a stationary camera, the resiliency of the pipeline to extrinsic camera perturbations was also assessed. Various external factors, such as inadvertent contact by a maintenance technician or vibrations during mid-flight turbulence, can induce movement in the camera. Continuously generating new training data and corresponding YOLO models every time the camera shifts is impractical and unsustainable for relative vectoring. Therefore, the second experiment involved applying the highest-performing model from Experiment 4.1 to relative vectoring on imagery captured from a moving camera. Specifically, the probe

[10] The drogue was moved slowly enough that time alignment between the camera and mocap system were negligible.

[11] Note that all test set images contained both a probe and drogue, thus reliability was measured across all images.

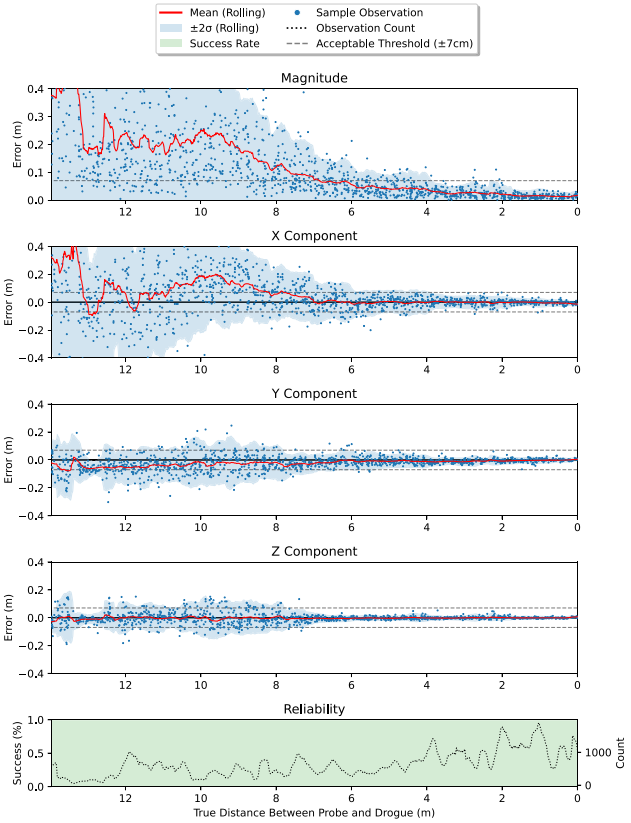

Fig. 24 Machine transfer learning training results for highest performing model, composition v-r (virtual and real)

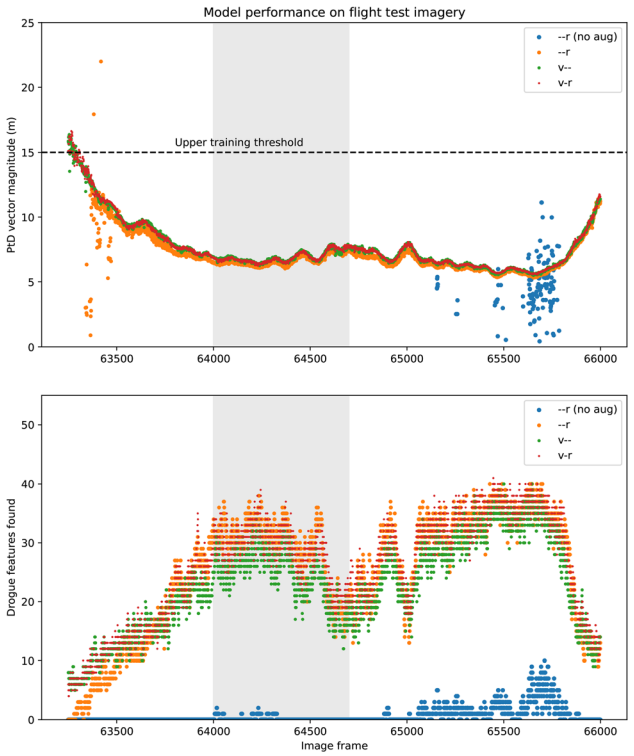

Fig. 25 Highlighted image frames depicted in Fig. 26

and drogue were positioned at fixed positions approximately 4 ms apart. Subsequently, the camera was systematically and randomly panned and tilted while capturing over 6,000 images along with corresponding mocap truth data. Finally, the resulting error in relative vector predictions was assessed.

4.3 Flight testing

In the final experiment, two objectives were pursued: (1) confirming positive machine transfer learning during real operations, and (2) assessing the benefits of scene augmentation. To accomplish these goals, relative vectoring was performed on imagery from real aerial refueling sorties using models trained both with and without scene augmentation. Despite the absence of flight truth data (e.g., differential GPS), qualitative analysis provided compelling evidence of pipeline performance and its ability to generalize when confronted with real environmental conditions. For instance, visual comparison between the digital twins of resulting predictions and corresponding imagery facilitated qualitative answers to questions such as: Did YOLO identify the correct features as lighting conditions changed? Did resulting relative vectors consistently align with what was depicted in the imagery for most observations?

A Learjet 25 receiver aircraft retrofitted with a refueling probe and a Gulfstream III tanker equipped with a refueling pod served as the docking vehicles in this experiment, see Fig. 16. To train YOLO to recognize the real Learjet, the same procedures previously applied to the mock probe in the lab were performed on the Learjet. This included mounting the camera in the cockpit, 3D scanning the front of the Learjet to establish a digital twin and performing digital twin alignment as shown in Fig. 17. The curved glass of the cockpit windshield introduced a significant secondary source of image distortion. Fortunately, a new chessboard calibration resulted in minimal re-projection error. Table 5 summarizes the updated intrinsic parameters applied to all flight test imagery. Over 890,000 images were collected at 20 fps across 8 sorties, encompassing a wide variety of flight profiles and weather conditions, including wings-level and banking approaches with daytime glare, overcast skies, shadows, clear skies, and transitions from dusk to night flying. From this dataset, a small subset of randomly selected images (16,020 total) from both day and nighttime sorties was set aside as probe foregrounds for training and validation data. All other collected images composed the test set.

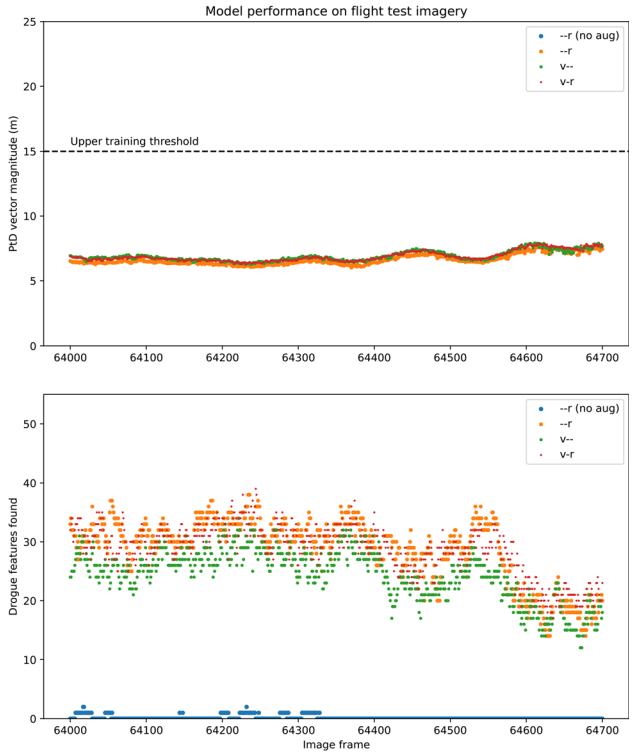

Fig. 26 Observations included ∼20° banked turn; background sun glare; windshield glare and shadow effects; bright horizon

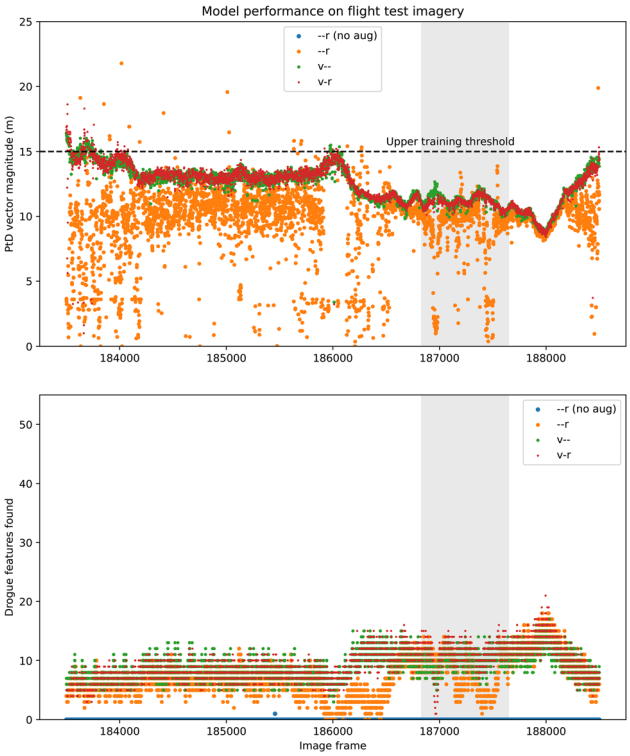

Fig. 27 Highlighted image frames depicted in Fig. 28

As a baseline, a YOLO model was trained on real-only drogue imagery without scene augmentation, –r (no aug). For comparison, scene augmentation was enabled for three other models: real-only (–r), virtual-only (v–), and the data source composition matching the highest-performing model from Experiment 4.1 (excluding –r and v–). Although the drogue background images were derived from previous synthetic sources, all mock probe foregrounds were replaced with the real samples set aside from the day and nighttime flight tests. These were merged using a new alpha mask, as shown in Fig. 18. Finally, the resulting predictions from all models were compared.

5 Results and discussion

5.1 Quantitative analysis

Violin plots in Figs. 19, 20 and 21 summarize model performance from Experiment 4.1 within close range, i.e., PtD distance of 0 to 6 m. Additionally, Figs. 23 and 24 depict model performance across the entire range of observations for the lowest and highest performing models, v– and v-r,

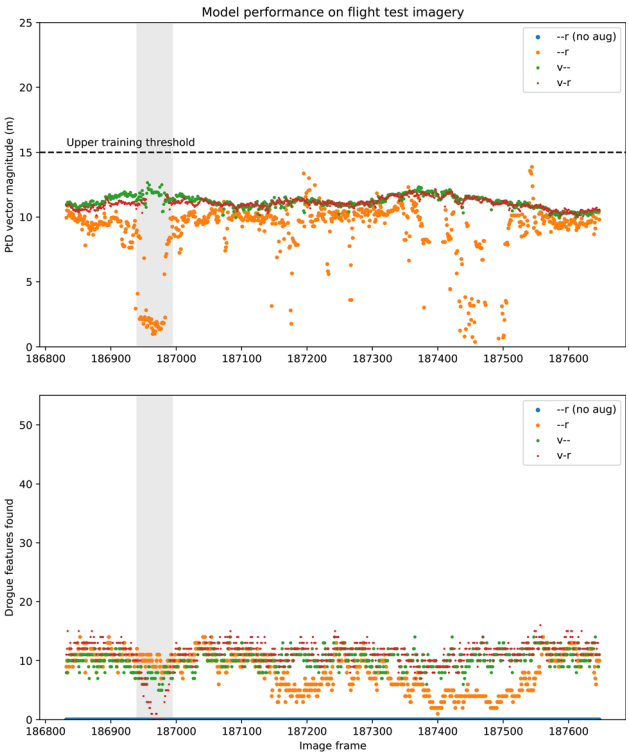

Fig. 28 Observations included ∼25° banked turn; bright horizon; steady approach; wings level; minimal clouds; background sun glare; drogue disappeared within windshield glare from frame 186940 to 186994 (highlighted)

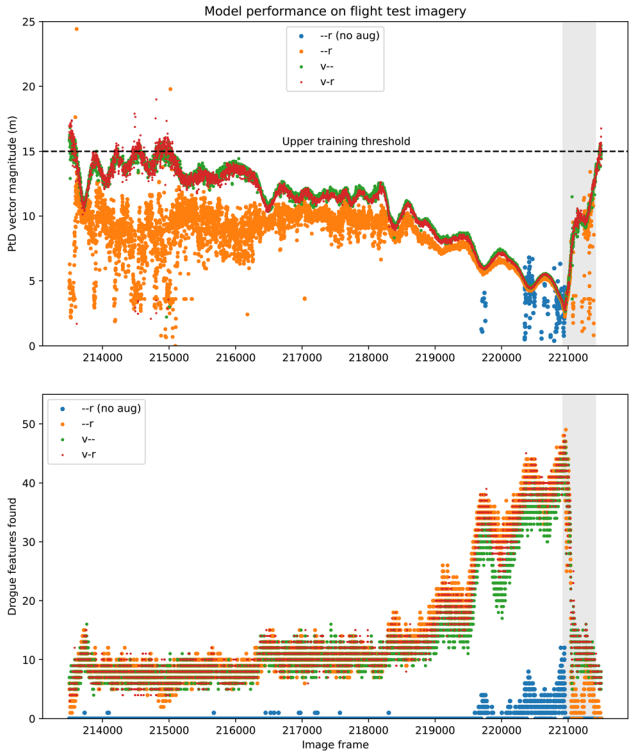

Fig. 29 Highlighted image frames depicted in Fig. 30

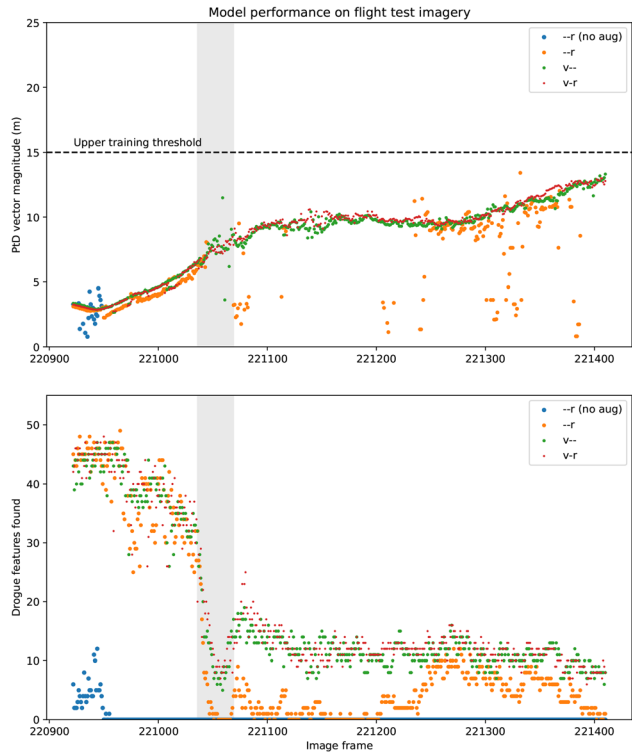

Fig. 30 Observations included wings level; bright horizon; drogue rapidly sweeps behind tanker; drogue partially departs camera FOV from frame 221036 to 221069 (highlighted)

respectively.[12] Most error for all models was observed in the x-dimension (depth). In contrast, there was minimal error in both y and z dimensions, and magnitude error was lowest at contact. Fortunately, these are also the most critical dimensions and range during docking maneuvers. As expected, relative vector prediction accuracy diminished as the distance between the probe and drogue increased. However, the top-performing model achieved nearly 100% reliability across all test images—only failing to predict PtD vectors for 3 of the 35,918 test set images. Additionally, PtD magnitude error at contact was well below the 7 cm acceptable threshold for all but the lowest performing model, v–. These results demonstrate that while positive machine transfer learning from virtual-only training to real-world inference was achieved, model performance increased significantly when also trained on real lab imagery.

Composition –r was expected to produce the highest performing model since it closely resembled the test set, comprising only real lab imagery. However, training on both virtual and real images (v-r) resulted in a slight overall performance improvement, possibly due to the increased variety of drogue perspectives in the training dataset. Although models trained on hybrid data (-h-, -hr, vh-, and vhr) outperformed those trained on virtual-only data, they did not show significant improvements beyond the virtual/real composition, v-r. This suggests that including hybrid, i.e., reverse green-screened, training data may offer little to no additional benefit in this particular use case.

For Experiment 4.2, the highest performing model, v-r, was deployed in the relative vectoring pipeline while the camera was in motion. As shown in Fig. 22 and video [85], the pipeline maintained high performance regardless of camera orientation. Although a few outliers were observed when the camera was jostled suddenly, causing blurred images, or when too many probe and drogue features left the image frame, the pipeline remained consistently stable and accurate. These results clearly demonstrate that the relative vectoring pipeline does not rely on extrinsic camera calibrations and is resilient to dynamic changes in camera orientation.

Table 6 presents the execution time for each pipeline operation. While the total time from loading the image to producing a PtD vector was nearly 31.5 ms, parallelizing the pipeline—loading the current image while processing the previous one—reduced the execution time between

[12] See Appendix A for violin plots beyond 6 m and full error distributions for the other five models.

Table 7 Flight test video [86] summary

Image frames	Sortie	Duration	Observations	Results
63,250–66,000	dt3	2:18	Banked turn, bright glare from sun sweeping across field of view	Fig. 25
183,500–188,500	dt4	4:10	Mostly level flight with a single banked turn, clear skies, and various shadow/glare effects	Fig. 27
213,500–221,500	dt4	6:40	Level flight, gradual approach, near contact ($\sim$3 m out), drogue rapidly sweeps across FOV and behind tanker drogue rapidly sweeps across FOV and behind tanker	Fig. 29
326,500–330,000	dt5	2:55	Elevated cloud cover, dark shadows on receiver, sun glare completely occludes drogue	Fig. 31
445,800–449,300	dt6	2:55	Flying straight and level just above cloud ceiling, near contact ($\sim$3 to 7 m out), probe tip occludes drogue, drogue momentarily departs field of view	Fig. 33
549,350–603,384	nt0	45:02	Night flying: transition from sunset to near total dark- ness, four drogue LEDs brightly lit	Fig. 35

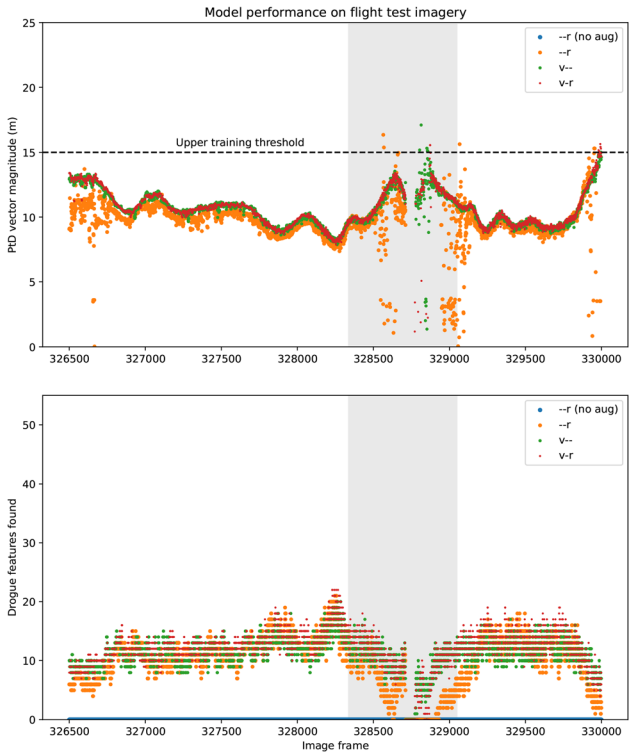

Fig. 31 Highlighted image frames depicted in Fig. 32

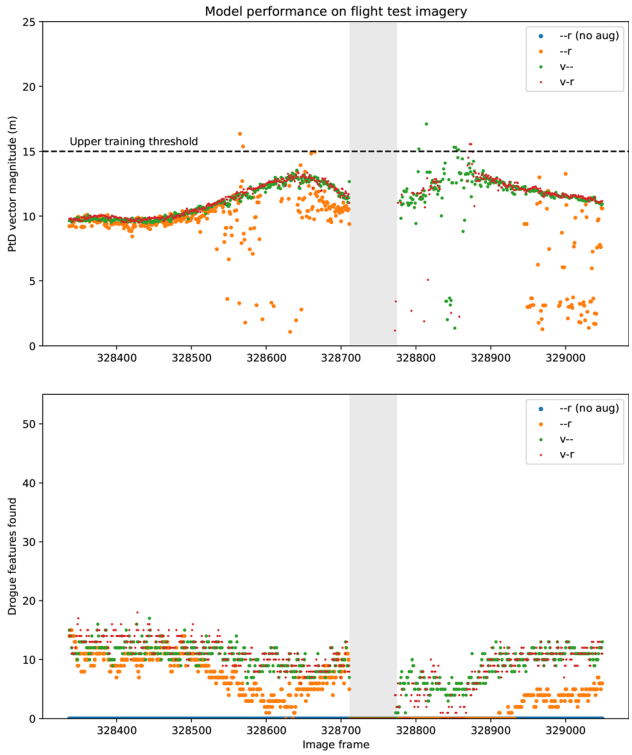

Fig. 32 Observations included $\sim$25° banked turn; elevated presence of clouds; sun glare; drogue completely disappears in windshield glare from frame 328712 to 328774 (highlighted)

predictions to approximately 17.846 ms, achieving a speed of 56 fps.[13]

5.2 Qualitative analysis

As previously mentioned, the highest performing dataset composition from Experiment 4.1 was v-r, which resulted in less than 3 cm of error at contact. Thus, a corresponding model for it was trained and tested as part of Experiment 4.3. The first 24 s of video [86] showcase the flight test training datasets used to train YOLO models for Experiment 4.3, which included: –r (no aug), –r, v–, and v-r. The remainder of the video [86] visually presents corresponding YOLO model performance and resulting relative vector predictions under various operational conditions; a summary of these conditions is provided in Table 7. Additionally, the plots in Figs. 25 through 36 summarize model performance for all images depicted in the video [86]. Overall, positive machine transfer learning to real flight test imagery was successfully achieved. However, as with any supervised machine learning task,

[13] Note that this frame rate does not account for camera speed. The camera used in this work was limited to 20 fps, but also collected imagery at an unnecessarily high resolution. Future work could test this pipeline using cameras with higher frame rates and an output resolution matching that of the YOLO input size.

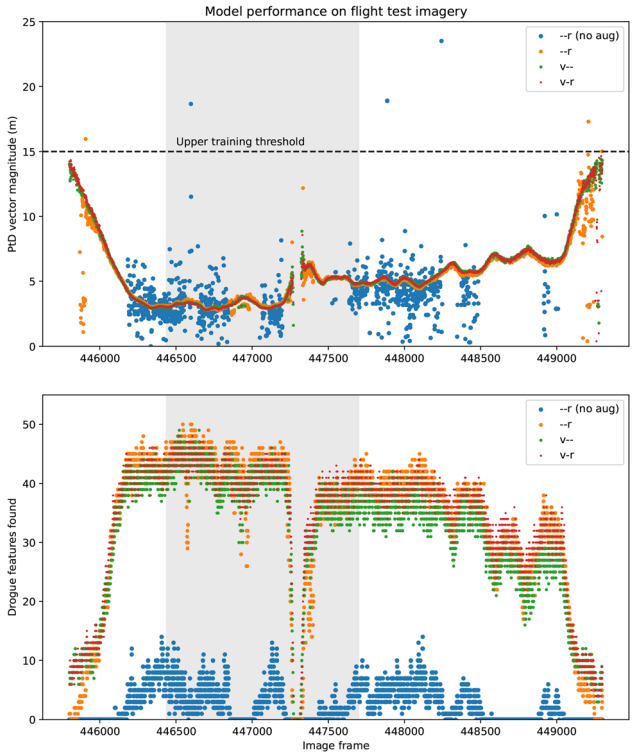

Fig. 33 Highlighted image frames depicted in Fig. 34

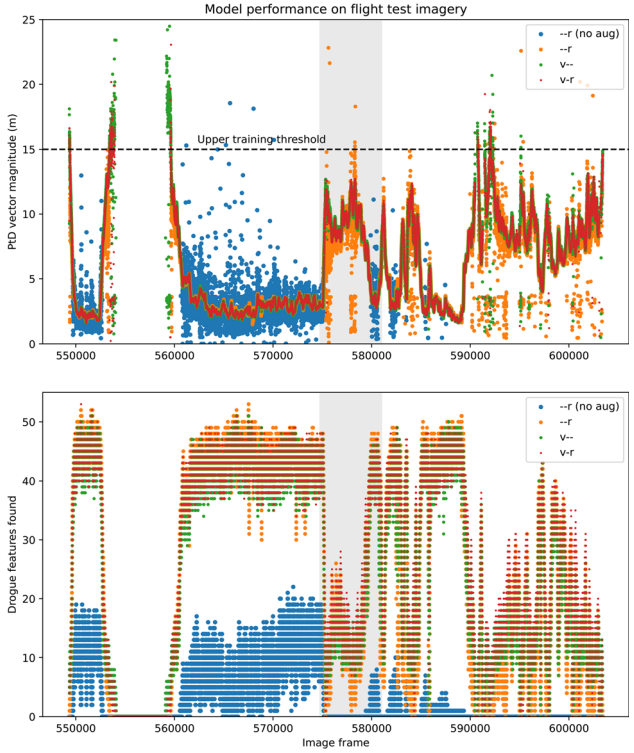

Fig. 35 Highlighted image frames depicted in Fig. 36

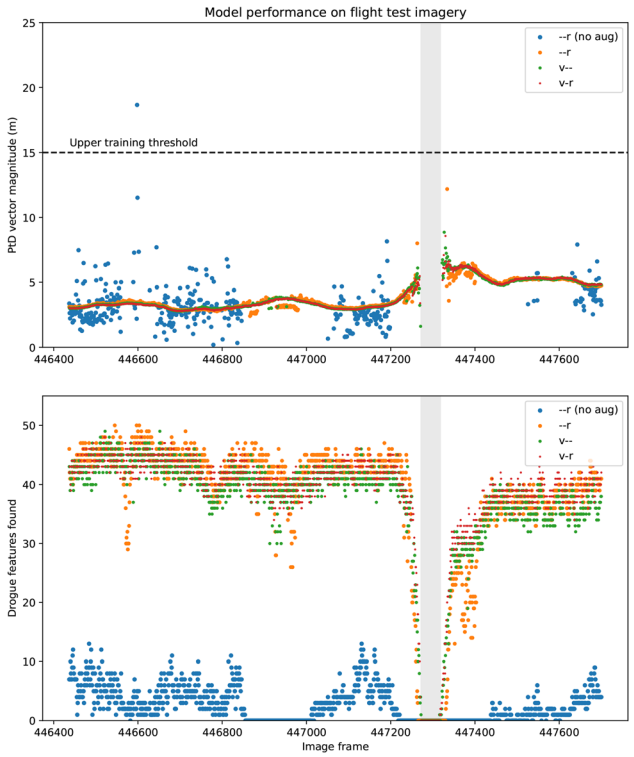

Fig. 34 Observations included wings level; elevated presence of clouds; probe occludes drogue; drogue sweeps behind tanker; drogue completely departs camera FOV from frame 447272 to 447318 (highlighted)

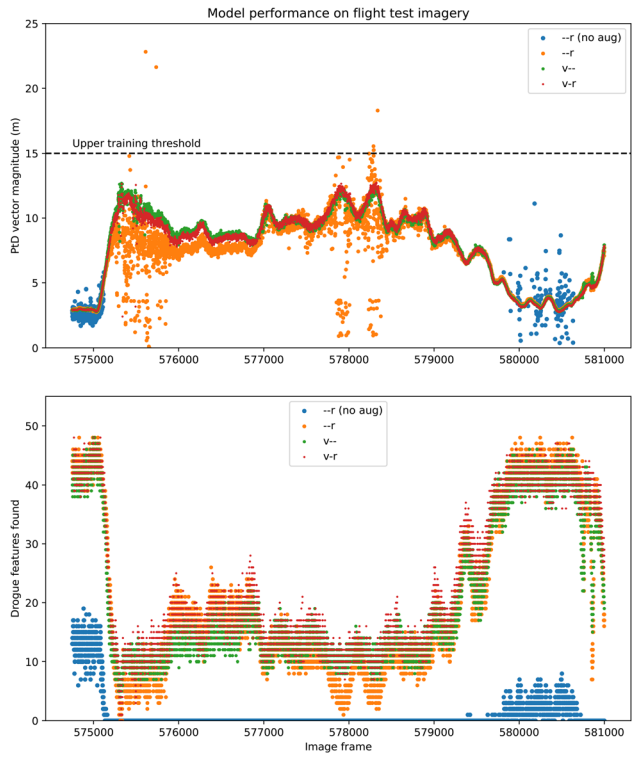

Fig. 36 Observations: $\sim 0^{\circ}$ to 25° banked turn; heavy cloud cover; dark shadows; transition from day with bright horizon to night flying; LED beacons illuminated

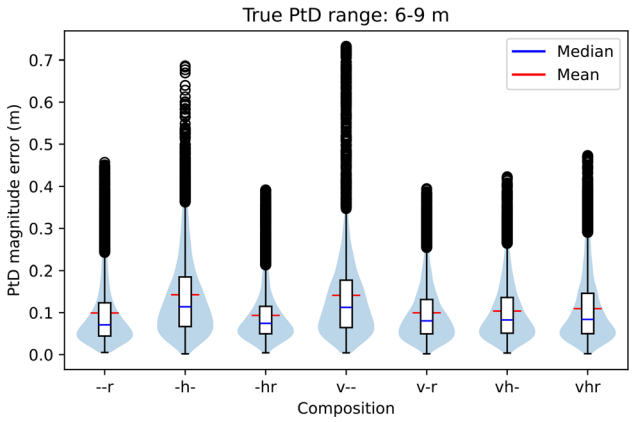

Fig. 37 Machine transfer learning combination training results—error at inner mid-range (6–9 m)

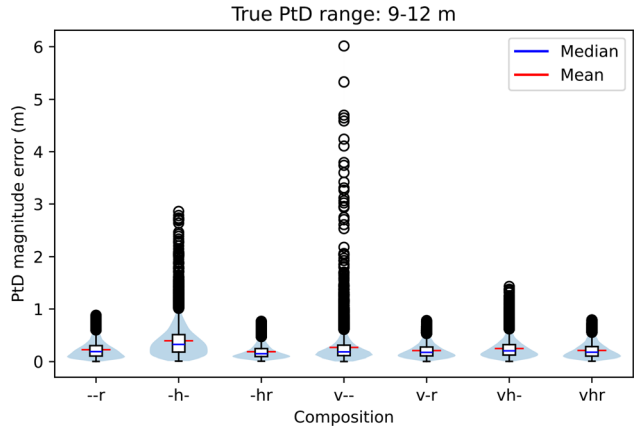

Fig. 38 Machine transfer learning combination training results—error at outer mid-range (9–12 m)

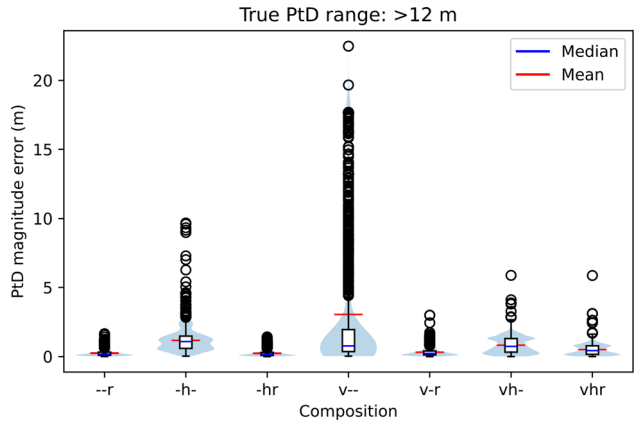

Fig. 39 Machine transfer learning combination training results—error beyond mid-range (greater than 12 m)

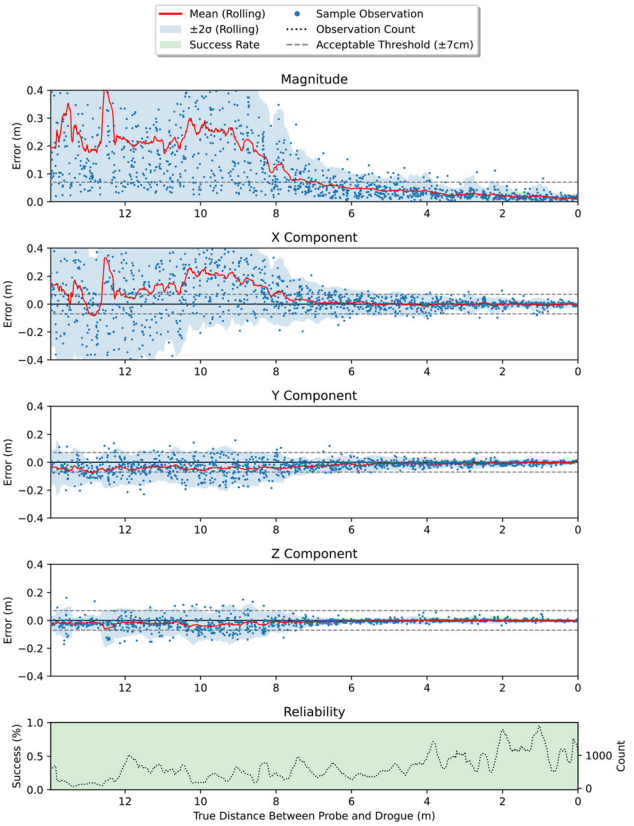

Fig. 40 Machine transfer learning training results for composition –r (real only)

YOLO's performance varied significantly depending on its training dataset composition.

Probe feature detection All models demonstrated nearly equal performance, achieving close to 100% reliability in identifying probe features, even during ascents and descents through clouds (not depicted in video [86])—only struggling during the darkest scenes of the nighttime sortie, nt0. This consistent high performance across all models was anticipated since they were all trained on the same probe foregrounds from real flight test imagery. Additionally, all models successfully identified the correct features despite variations in camera rotation between sorties, further confirming that relative vectoring does not depend on extrinsic camera calibrations.

Drogue feature detection When identifying drogue features, the higher feature counts detected at closer ranges correlated to more stable predictions with less variation. Moreover, model performance roughly correlated to the amount of scene augmentation applied. Models trained with augmentation significantly outperformed the one without it, regardless of time of day, flight maneuver, or weather conditions. Similarly, models trained on a combination of lighting and background augmentation (v– and v-r) performed better than the one with only lighting

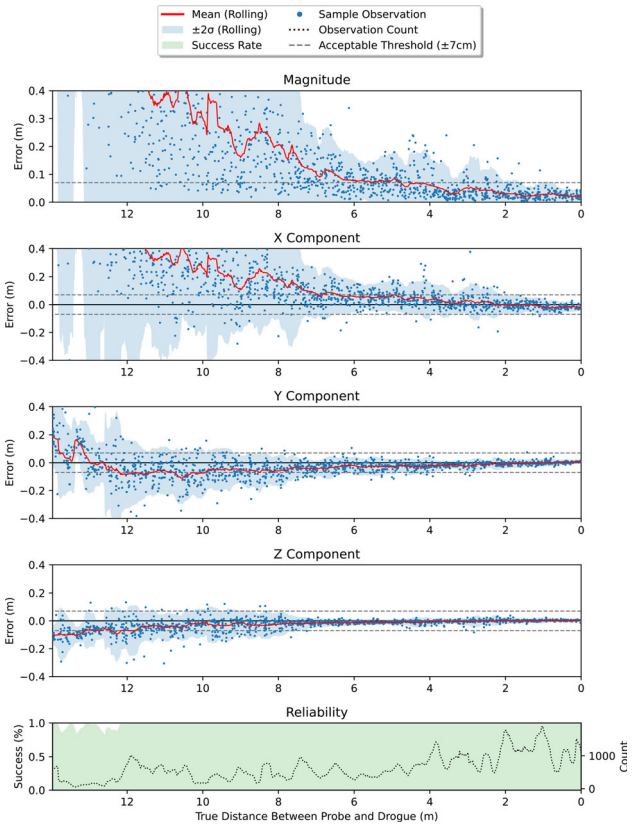

Fig. 41 Machine transfer learning training results for composition -h- (hybrid only)

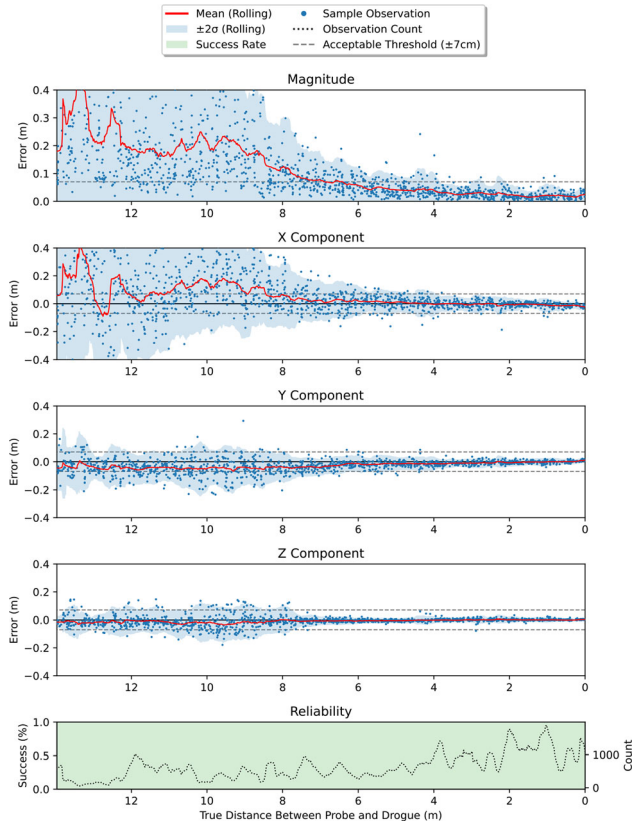

Fig. 42 Machine transfer learning training results for composition -hr (hybrid and real)

augmentation (–r). More specifically, the –r model without scene augmentation rarely detected enough features to make a PtD vector prediction. In contrast, the –r model with scene augmentation performed much better, though it often made drogue feature predictions that were obviously incorrect or not on the drogue. Depending on lighting, this model produced PtD vectors with high stability within 4 m but high instability beyond 8 m. Both models v– and v-r produced PtD vectors with high stability out to 15 m regardless of lighting, with the v-r model being slightly more stable, especially after dark.

Remarkably, models v– and v-r accurately and consistently found drogue features after sunset, even when the human eye could no longer differentiate features in the darkness. These models generalized well at night despite never training on the four green LEDs brightly lit on the drogue after dark (video [86] from 22:43–26:52). During the day, these two models also performed well, even when encountering a wide variety of shadows, reflections, and glare from the sun passing directly behind the drogue. No significant performance impacts were observed for these models when the probe occluded the drogue or the drogue entered the tanker's silhouette. Overall, analysis of flight test data revealed that scene augmentation is critical for generalization and high model performance. Furthermore, the training dataset composition v-r produced the most successful YOLO model, demonstrating that positive machine transfer learning is possible from synthetic-only data.

5.3 Performance evaluation

The performance metrics achieved—error margins of less than 3 cm at contact and over 56 fps—are well-suited for dynamic environments like aerial refueling. These results compare favorably with other autonomous docking systems, which typically target error margins of 1–5 cm depending on task complexity and conditions [10, 62, 81]. Achieving less than 3 cm error while both vehicles are in motion, despite factors like turbulence, demonstrates the robustness of our approach. Additionally, human operators, for comparison, are capable of manual docking and typically respond to visual cues with a reaction time of greater than 200 ms (5 fps) [69], which is several times slower than our system's processing speed. This suggests that a frame rate of 56 fps offers more than enough time for the system to sufficiently react to changes in the environment.

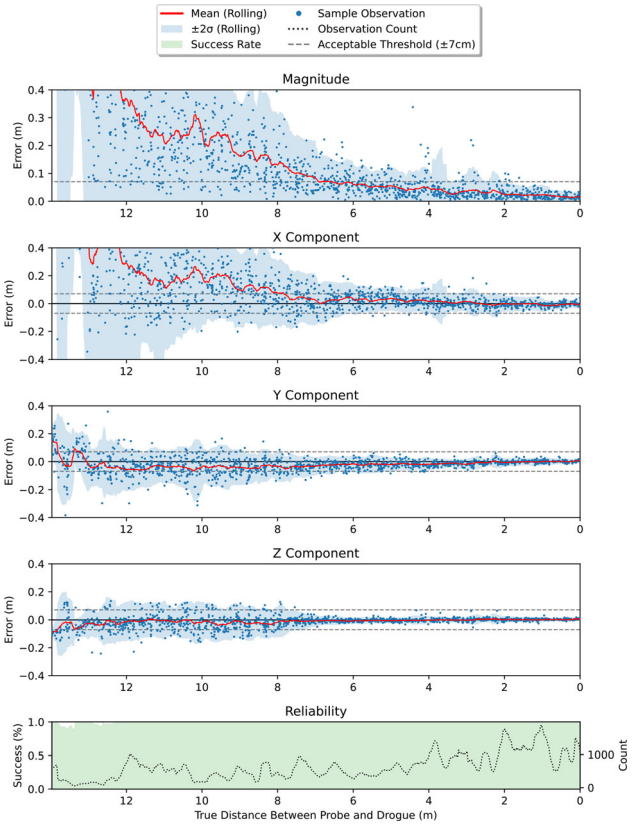

Fig. 43 Machine transfer learning training results for composition vh- (virtual and hybrid)

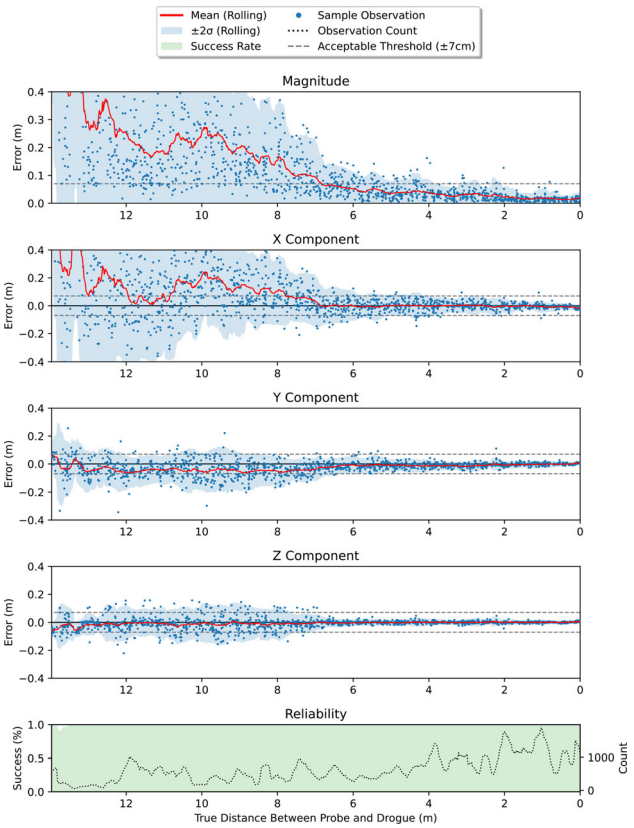

Fig. 44 Machine transfer learning training results for composition vhr (virtual, hybrid, and real)

Key challenges that emerged during testing include handling sensor noise, real-time processing of high-resolution imagery, and overcoming environmental factors like fluctuating lighting conditions and the unpredictable behavior of docking components. These challenges, while notable, did not prevent the system from maintaining high accuracy and reliability in docking maneuvers, underscoring the system's effectiveness in dynamic and complex conditions.

6 Conclusions

This study demonstrates the effectiveness of machine transfer learning in enhancing relative vectoring for autonomous docking maneuvers, particularly in autonomous aerial refueling (AAR). Utilizing digital twin alignment from fiducial targets and motion capture data, a reliable framework was developed for producing accurate and realistic synthetic training data. The innovative approach to annotation automation significantly improved the performance and generalization of YOLO object detection models, ultimately enabling real-world relative vectoring with high accuracy and reliability.

A key achievement of this research was overcoming the sim-to-real gap. This was tackled by integrating augmented reality and scene augmentation techniques, enriching synthetic training data with realistic variations. The combined approach of using synthetic and real data enabled the models to generalize better to real-world conditions. Additionally, the use of fiducial targets and motion capture data ensured precise alignment and calibration, further bridging the gap between simulation and reality.

Despite these advancements, several limitations persist. The reliance on synthetic data, while beneficial, still falls short of capturing the full complexity of real-world conditions. The current scope of testing has been limited to laboratory environments and controlled flight tests, necessitating further validation in diverse operational settings. Additionally, while annotation automation techniques have reduced manual effort, there remains potential for further increasing their accuracy and efficiency.

Several limitations persist. The reliance on synthetic data, while beneficial, still falls short of capturing the full

complexity of real-world conditions. The current scope of testing has been limited to laboratory environments and controlled flight tests, necessitating further validation in diverse operational settings. Additionally, while annotation automation techniques have reduced manual effort, there remains potential for further increasing their accuracy and efficiency.

Future research should focus on enhancing the realism of synthetic data and exploring training datasets that combine real and synthetic images.[14] Extensive real-world testing under varying conditions is crucial to ensure reliability and scalability. Additionally, more sophisticated annotation automation methods and adaptive learning techniques should be investigated to improve labeling precision and model adaptability. In this context, "adaptive" is defined as the system's ability to adjust in real time to variations in environmental conditions, such as lighting, turbulence, or sensor noise. Further exploration of the system's ability to recover from failures or unexpected disruptions—key to enhancing robustness—would also be beneficial. Integrating multi-sensor fusion (e.g., vision and laser distance sensors) could further improve adaptability and resilience in complex, dynamic environments.

By addressing these limitations and pursuing these avenues, future work can build on the findings in this paper to develop more robust, adaptable, and efficient autonomous docking systems, ultimately achieving the vision of fully automated docking without human intervention, thereby enhancing the capabilities of AAR and related applications.

Disclaimer

The views expressed are those of the author and do not reflect the official policy or position of the US Air Force, Department of Defense, or the US Government.

Supplementary information

This paper has accompanied videos which can be found at [85, 86].

Appendix A Additional combination training test results

Section 5 presents the top-performing results from Experiment 4.1. This appendix complements those findings by providing additional related results, allowing for a comprehensive comparison and deeper understanding of the experimental outcomes. Specifically, the violin plots in Figs. 37 through 39 illustrate the machine transfer learning combination training results beyond 6 m. Meanwhile, Figs. 41 through 44 display the full error distributions for the lower-performing models (Figs. 37, 38, 39, 40, 41, 42, 43 and 44).

[14] Additional related research topics that extend this work are listed in Appendix B.

Appendix B Additional future work

The following list comprises additional subjects related to the domains of computer vision and machine learning that warrant exploration to further enrich the research presented in this paper:

- Generating labeled training data using Neural Radiance Fields (NeRFs) and Gaussian Splatting.
- Conducting relative vectoring without electro-optical data (e.g., long-wave infrared).
- Validating pipeline performance with differential GPS truth.
- Integrating a rear-facing camera for dual prediction comparison.
- Implementing Kalman/low-pass filtering techniques.
- Employing stateful noise reduction methodologies.
- Investigating cross-platform generalization, e.g., training on docking vehicles with different but similar profiles, surface textures, and configurations.
- Employing genetic algorithms for feature selection optimization.
- Conducting feature selection studies to ascertain the trade-offs between performance and feature type/quantity.
- Training distinct models to operate in parallel, focusing on learning diverse objects or features to enhance modularity, flexibility, and certifiability.
- Utilizing fisheye lens/360 imagery and exploring "relative vector chaining."
- Automating scene augmentation through mechanical actuators to expedite the generation of training sets.
- Exploring the potential of event-based vision sensors (EVS).
- Examining the efficacy of machine transfer learning across different camera configurations, including variations in horizontal field of view, resolution, distortion, and aspect ratio.
- Applying relative vectoring (or relative 6DoF pose estimation) to other autonomous docking scenarios.
- Incorporating multiple sensing modalities, such as combining vision with laser distance sensors, to enhance the reliability and accuracy of autonomous docking systems.

Acknowledgements We would like to thank all our sponsors, including the Naval Air Systems Command (USN/NAVAIR) and Aerospace Systems Directorate (USAF/AFRL/RQ). Your partnership and financial support made this research possible.

Funding This study was funded by the US Naval Air Systems Command and the Air Force Research Laboratory Aerospace Systems Directorate.

Data availability The datasets generated during and/or analyzed in this work are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Alamri F, Pugeault N (2020) Improving object detection performance using scene contextual constraints. IEEE Trans Cogn Dev Syst 14(4):1320–1330
2. Allied Vision (2024) Alvium g5-811: 5 gige vision. https://www.alliedvision.com/en/products/alvium-configurator/alvium-g5/811/
3. Artec 3D (2024) Technical specifications. https://www.artec3d.com/portable-3d-scanners/artec-leo#tech-specs
4. Balamurugan G, Valarmathi J, Naidu V (2016) Survey on UAV navigation in gps denied environments. In: 2016 International conference on signal processing, communication, power and embedded system (SCOPES). IEEE, pp 198–204
5. Bownes VJ (2021) Using motion capture and augmented reality to test aar with boom occlusion. Master's thesis, Air Force Institute of Technology
6. Campa G, Napolitano MR, Fravolini ML (2009) Simulation environment for machine vision based aerial refueling for UAVS. IEEE Trans Aerosp Electron Syst 45(1):138–151
7. Chen CI, Koseluk R, Buchanan C et al (2015) Autonomous aerial refueling ground test demonstration-a sensor-in-the-loop, non-tracking method. Sensors 15(5):10948–10972
8. Chen S, Duan H, Deng Y et al (2017) Drogue pose estimation for unmanned aerial vehicle autonomous aerial refueling system based on infrared vision sensor. Opt Eng 56(12):124105
9. Cheng J, Liu P, Zhang Q et al (2021) Real-time and efficient 6-d pose estimation from a single RGB image. IEEE Trans Instrum Meas 70:1–14
10. Chinchilla S, Saito T, Oikawa R et al (2024) Real-time marker-based monocular autonomous docking in semi-unstructured indoor environments. In: 2024 IEEE/SICE International Symposium on System Integration (SII). IEEE, pp 1561–1568
11. Chiodini S, Pertile M, Giubilato R et al (2018) Camera rig extrinsic calibration using a motion capture system. In: 2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace). IEEE, pp 590–595
12. Choate J, Worth D, Nykl S et al (2024) An analysis of precision: occlusion and perspective geometry's role in 6d pose estimation. Neural Comput Appl 36(3):1261–1281
13. Chu W, Cai D (2018) Deep feature based contextual model for object detection. Neurocomputing 275:1035–1042
14. Clark N (2015) What it takes to be a boom operator. https://www.af.mil/News/Article-Display/Article/585353/what-it-takes-to-be-a-boom-operator
15. Crawford JD, Medendorp WP, Marotta JJ (2004) Spatial transformations for eye–hand coordination. Journal of neurophysiology
16. Curro J, Raquet J, Pestak T et al (2012) Automated aerial refueling position estimation using a scanning lidar. In: Proceedings of the 25th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2012), pp 774–782
17. Daneshmand M, Helmi A, Avots E et al (2018) 3d scanning: A comprehensive survey. arXiv preprint arXiv:1801.08863
18. Dempsey DL, Barshi I (2020) Applying research-based training principles: Toward crew-centered, mission-oriented space flight training. In: Psychology and human performance in space programs. CRC Press, p 63–80
19. Deng J, Dong W, Socher R et al (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255
20. DeTone D, Malisiewicz T, Rabinovich A (2018) Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 224–236
21. Du JY (2004) Vision based navigation system for autonomous proximity operations: an experimental and analytical study. Texas A &M University
22. Duan H, Zhang Q (2015) Visual measurement in simulation environment for vision-based UAV autonomous aerial refueling. IEEE Trans Instrum Meas 64(9):2468–2480
23. Duan H, Xin L, Chen S (2019) Robust cooperative target detection for a vision-based UAVS autonomous aerial refueling platform via the contrast sensitivity mechanism of eagle's eye. IEEE Aerosp Electron Syst Mag 34(3):18–30
24. Dvornik N, Mairal J, Schmid C (2019) On the importance of visual context for data augmentation in scene understanding. IEEE Trans Pattern Anal Mach Intell 43(6):2014–2028
25. Everingham M, Van Gool L, Williams CK et al (2010) The pascal visual object classes (voc) challenge. Int J Comput Vision 88:303–338
26. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
27. Geiger A, Lenz P, Urtasun R (2013) Vision meets robotics: the kitti dataset. Int J Robot Res 32(11):1231–1237
28. Georgakis G, Mousavian A, Berg AC et al (2017) Synthesizing training data for object detection in indoor scenes. arXiv preprint arXiv:1702.07836
29. Gil A, Khurshid A, Postal J et al (2019) Visual assessment of equirectangular images for virtual reality applications in unity. Anais Estendidos do XXXII Conference on Graphics. Patterns and Images, SBC, pp 237–242
30. Gill S, Aryan A (2016) To experimental study for comparison theodolite and total station. Int J Eng Res Sci 3:153–160
31. Grlj CG, Krznar N, Pranjić M (2022) A decade of UAV docking stations: a brief overview of mobile and fixed landing platforms. Drones 6(1):17
32. Hammarkvist T (2021) Automatic annotation of models for object classification in real time object detection

33. Hattori H, Naresh Boddeti V, Kitani KM et al (2015) Learning scene-specific pedestrian detectors without real data. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3819–3827
34. He K, Gkioxari G, Dollár P et al (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
35. Hinterstoisser S, Lepetit V, Ilic S et al (2013) Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Computer Vision–ACCV 2012: 11th Asian Conference on Computer Vision, Daejeon, Korea, November 5–9, 2012, Revised Selected Papers, Part I 11, Springer, pp 548–562
36. Hodan T, Haluza P, Obdržálek Š et al (2017) T-less: An rgb-d dataset for 6d pose estimation of texture-less objects. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp 880–888
37. Huang WL, Hung CY, Lin IC (2021) Confidence-based 6d object pose estimation. IEEE Transactions on Multimedia
38. Hussain M (2023) Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection. Machines 11(7):677
39. Jinrui R, Quan Q (2023) Progress in modeling and control of probe-and-drogue autonomous aerial refueling. Chinese Journal of Aeronautics
40. Joshi AG, Dabhade AS, Borse AS (2015) Virtual reality in android gaming. Int Res J Eng Technol (IRJET) 2:2322–2327
41. Kang J, Liu W, Tu W et al (2020) Yolo-6d+: single shot 6d pose estimation using privileged silhouette information. In: 2020 International Conference on Image Processing and Robotics (ICIP). IEEE, pp 1–6
42. Kehl W, Manhardt F, Tombari F et al (2017) Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In: Proceedings of the IEEE international conference on computer vision, pp 1521–1529
43. Keshavarzi M, Parikh A, Zhai X et al (2020) Scenegen: Generative contextual scene augmentation using scene graph priors. arXiv preprint arXiv:2009.12395
44. Kiyokawa T, Tomochika K, Takamatsu J et al (2019) Fully automated annotation with noise-masked visual markers for deep-learning-based object detection. IEEE Robotics Automat Lett 4(2):1972–1977
45. Li C, Yan X, Li S et al (2020) Survey on ship autonomous docking methods: Current status and future aspects. In: ISOPE International Ocean and Polar Engineering Conference, ISOPE, pp ISOPE–I
46. Li C, Sun S, Song X et al (2022) Simultaneous multiple object detection and pose estimation using 3d model infusion with monocular vision. arXiv preprint arXiv:2211.11188
47. Lin TY, Maire M, Belongie S et al (2014) Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13, Springer, pp 740–755
48. Lu Z, Huang D, Bai L et al (2023) Seeing is not always believing: A quantitative study on human perception of ai-generated images. arXiv preprint arXiv:2304.13023
49. Mammarella M, Campa G, Napolitano MR et al (2010) Comparison of point matching algorithms for the UAV aerial refueling problem. Mach Vis Appl 21(3):241–251
50. McFatter J, Keiser K, Rupp TW (2018) Nasa docking system block 1: Nasa's new direct electric docking system supporting iss and future human space exploration. In: Aerospace Mechanisms Symposium, JSC-E-DAA-TN51081
51. Narasimhappa M, Mahindrakar AD, Guizilini VC et al (2019) Mems-based IMU drift minimization: sage Husa adaptive robust Kalman filtering. IEEE Sens J 20(1):250–260
52. Navarro J, Hernout E, Osiurak F et al (2020) On the nature of eye-hand coordination in natural steering behavior. PLoS ONE 15(11):e0242818
53. Nechaev A (2001) Work and rest planning as a way of crew member error management. Acta Astronaut 49(3–10):271–278
54. Noh D, Kim S, Kim S et al (2023) Docking method for electric vehicle charging terminal using monocular camera. https://assets-eu.researchsquare.com/files/rs-3180077/v1/cab07e06-9bb1-4eec-9dae-783dbc8b9179.pdf
55. Nowruzi FE, Kapoor P, Kolhatkar D et al (2019) How much real data do we actually need: Analyzing object detection performance using synthetic and real data. arXiv preprint arXiv:1907.07061
56. Nykl S (2022) Aftrburner 3d visualization engine. http://www.nykl.net/aburn
57. OptiTrack (2024) Primex 41. https://optitrack.com/cameras/primex-41/
58. Ostroumov I, Kuzmenko N, Bezkorovainyi Y et al (2022) Relative navigation for vehicle formation movement. In: 2022 IEEE 3rd KhPI Week on Advanced Technology (KhPIWeek). IEEE, pp 1–4
59. Pande B, Padamwar K, Bhattacharya S et al (2022) A review of image annotation tools for object detection. In: 2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC). IEEE, pp 976–982
60. Parsons C, Paulson Z, Nykl S et al (2019) Analysis of simulated imagery for real-time vision-based automated aerial refueling. J Aerosp Inf Syst 16(3):77–93
61. Phong BT (1975) Illumination for computer generated pictures. CACM
62. Pirat CS, Mäusli PA, Walker R et al (2018) Guidance, navigation and control for autonomous cooperative docking of cubesats. In: The 4S Symposium 2018
63. Rad M, Lepetit V (2017) Bb8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth. In: Proceedings of the IEEE international conference on computer vision, pp 3828–3836
64. Rajpura PS, Bojinov H, Hegde RS (2017) Object detection using deep cnns trained on synthetic images. arXiv preprint arXiv:1706.06782
65. Real E, Shlens J, Mazzocchi S et al (2017) Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 5296–5305
66. Rodenburgh E, Taylor C (2020) A system for evaluating vision-aided navigation uncertainty. In: Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), pp 2272–2280
67. Rozantsev A, Lepetit V, Fua P (2015) On rendering synthetic images for training an object detector. Comput Vis Image Underst 137:24–37
68. Shappell S, Stringfellow P, Baron J et al (2008) The effects of shiftwork on human performance and its implications for regulating crew rest and duty restrictions during commercial space flight. Tech. rep., Clemson University
69. Shelton J, Kumar GP (2010) Comparison between auditory and visual simple reaction times. Neurosci Med 1(01):30–32
70. Siddiqi AA (2000) Challenge to Apollo: the Soviet Union and the space race, 1945–1974, vol 4408. National Aeronautics and Space Administration, NASA History Division, Office
71. Signal TL, Gander PH, van den Berg MJ et al (2013) In-flight sleep of flight crew during a 7-hour rest break: implications for research and flight safety. Sleep 36(1):109–115
72. Smirnov A (2020) Chroma keying with opencv/c++. https://smirnov-am.github.io/chromakeying

73. Sundermeyer M, Marton ZC, Durner M et al (2018) Implicit 3d orientation learning for 6d object detection from rgb images. In: Proceedings of the european conference on computer vision (ECCV), pp 699–715
74. Talukdar J, Gupta S, Rajpura P et al (2018) Transfer learning for object detection using state-of-the-art deep neural networks. In: 2018 5th international conference on signal processing and integrated networks (SPIN). IEEE, pp 78–83
75. Tekin B, Sinha SN, Fua P (2018) Real-time seamless single shot 6d object pose prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 292–301
76. Tomasi C (2015) A simple camera model. In: Notes from computer science 527, https://courses.cs.duke.edu//fall16/compsci527/notes/camera-model.pdf
77. Tran Q, Choate J, Taylor CN et al (2023) Monocular vision and machine learning for pose estimation. 2023 IEEE/ION Position. Location and Navigation Symposium (PLANS). IEEE, pp 128–136
78. Ultralytics (2024) Yolov5 in pytorch. https://github.com/ultralytics/yolov5
79. Wang F, Wang G, Lu B (2024) YOLOv8-PoseBoost: advancements in multimodal robot pose keypoint detection. Electronics 13(6):1046
80. Wang T, Anwer RM, Khan MH et al (2019) Deep contextual attention for human-object interaction detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 5694–5702
81. Wirtensohn S, Hamburger O, Homburger H et al (2021) Comparison of advanced control strategies for automated docking. IFAC-PapersOnLine 54(16):295–300
82. Wong XI, Majji M (2016) A structured light system for relative navigation applications. IEEE Sens J 16(17):6662–6679
83. Worth D, Choate J, Lynch J et al (2023) Relative vectoring using dual object detection for autonomous aerial refueling. https://youtu.be/RXbrBl8Re7M
84. Worth D, Choate J, Lynch J et al (2024a) Relative vectoring using dual object detection for autonomous aerial refueling. Neural Comput Appl 38(2):1123–1138
85. Worth D, Choate J, Nykl S et al (2024b) Moving camera: relative vectoring lab test results. https://youtu.be/A6c6xcV1OeM
86. Worth D, Choate J, Nykl S et al (2024c) Sim-to-real transfer learning flight test results for relative vectoring. https://youtu.be/REle9bJ5mLY
87. Wu J, Yuan C, Yin R et al (2020) A novel self-docking and undocking approach for self-changeable robots. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, pp 689–693
88. Xiang Y, Schmidt T, Narayanan V et al (2017) Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. arXiv preprint arXiv:1711.00199
89. Xin L, Luo D, Li H (2018) A monocular visual measurement system for UAV probe-and-drogue autonomous aerial refueling. Int J Intell Comput Cybern 11(2):166–180
90. Yabuki N, Nishimura N, Fukuda T (2018) Automatic object detection from digital images by deep learning with transfer learning. In: Advanced Computing Strategies for Engineering: 25th EG-ICE International Workshop 2018, Lausanne, Switzerland, June 10–13, 2018, Proceedings, Part I 25, Springer, pp 3–15
91. Yang Y, Liang KJ, Carin L (2020) Object detection as a positive-unlabeled problem. arXiv preprint arXiv:2002.04672
92. Zhang J, Liu Z, Gao Y et al (2020) Robust method for measuring the position and orientation of drogue based on stereo vision. IEEE Trans Industr Electron 68(5):4298–4308
93. Zhang Z (2000) A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell 22(11):1330–1334
94. Zheng WS, Gong S, Xiang T (2011) Quantifying and transferring contextual information in object detection. IEEE Trans Pattern Anal Mach Intell 34(4):762–777
95. Zhou R, She J, Qi N et al (2022) Pose estimation algorithm for helicopter landing based on yolo and pnp. In: Advances in Guidance, Navigation and Control: Proceedings of 2020 International Conference on Guidance, Navigation and Control, ICGNC 2020, Tianjin, China, October 23–25, 2020, Springer, pp 3019–3028
96. Zhou X, Wang D, Krähenbühl P (2019) Objects as points. arXiv preprint arXiv:1904.07850
97. Zhuang F, Qi Z, Duan K et al (2020) A comprehensive survey on transfer learning. Proc IEEE 109(1):43–76