Augmenting Flight Imagery from Aerial Refueling

James D. Anderson[1], Scott Nykl[2(✉)], and Thomas Wischgoll[1]

[1] Wright State University, Dayton, OH 45435, USA
{anderson.10,thomas.wischgoll}@wright.edu
[2] Air Force Institute of Technology, Dayton, OH 45435, USA
scott.nykl@afit.edu
http://www.avida.cs.wright.edu

Abstract. When collecting real-world imagery, objects in the scene may be occluded by other objects from the perspective of the camera. However, in some circumstances an occluding object is absent from the scene either for practical reasons or the situation renders it infeasible. Utilizing augmented reality techniques, those images can be altered to examine the affect of the object's occlusion. This project details a novel method for augmenting real images with virtual objects in a virtual environment. Specifically, images from automated aerial refueling (AAR) test flights are augmented with a virtual refueling boom arm, which occludes the receiving aircraft. The occlusion effects of the boom are quantified in order to determine which pixels are not viable for stereo image processing to reduce noise and increase efficiency of estimating aircraft pose from stereo images.

Keywords: Augmented reality · Virtual reality simulation · Vision occlusion

1 Introduction

Virtual 3D graphic environments offer the opportunity to simulate the real-world in a deterministic, controlled manner. These environments are also capable of playing back data collected from real-world experiments and introduce conditions and constraints not possible during data collection. For instance, images captured from real cameras can be augmented with virtual objects overlaid. This augmentation visualizes how the scene would appear had the virtual object been present during initial data collection. When analyzing the augmented images, the effects of the virtual object can be taken into account when comparing the augmented image to the original image.

Work on Automated Aerial Refueling (AAR) has focused on utilizing a stereo vision camera system to accurately inform the flight crew with relative position

Supported by Air Force Research Labs - Aerospace Systems (AFRL/RQ).

G. Bebis et al. (Eds.): ISVC 2019, LNCS 11845, pp. 154–165, 2019.
https://doi.org/10.1007/978-3-030-33723-0_13

data of the aircraft involved. The receiver is the airplane requesting fuel, and the tanker provides the fuel to the receiver via a boom arm. The tanker aircraft is equipped with such a boom and a stereo camera system located on the bottom of the fuselage, behind the boom and facing toward the rear and around 25° down from horizontal. Figure 1 depicts a virtual rendering of a refueling approach as well as the frusta of the stereo cameras.

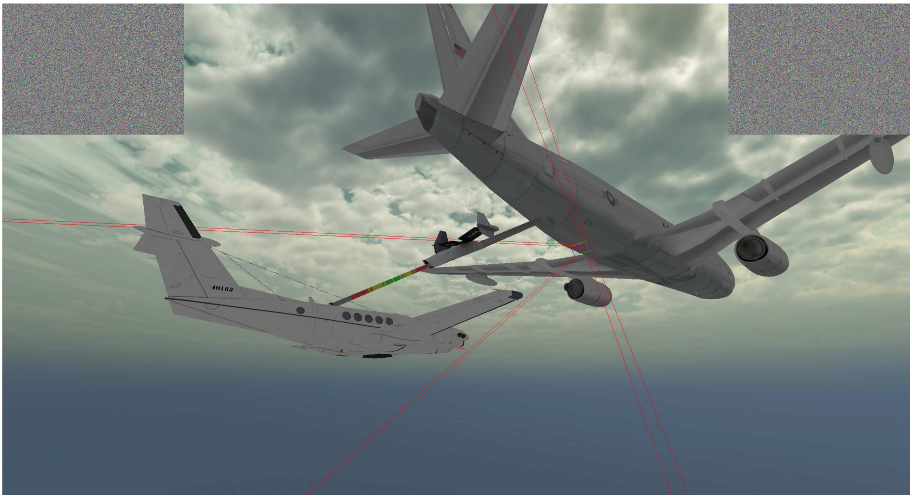


Fig. 1. Virtual rendering of tanker based on a 767 airframe and a Beechcraft as receiver. The stereo camera frusta are visualized pointed toward the boom and receiver.

This paper details work on augmenting images taken from test flights of refueling approaches with a virtual rendering of the boom arm. During the test flights, two Beechcraft airplanes acted as both receiver and tanker. Since it is infeasible to attach a boom to a Beechcraft, the real world test flight images do not contain the boom. However, as depicted in Fig. 1, the boom arm occludes the receiver from the perspective of the cameras. This occlusion introduces difficulties when attempting to determine the position and orientation of the receiver relative to the tanker's stereo vision system. Augmenting the images with a virtual boom lets researchers deterministically test various occlusion solutions using real imagery.

When augmenting the images with a virtual boom, the number of pixels in the left and right stereo images which occlude the vision of the receiver can be quantified. Additionally, because of the nature of stereo vision, certain features visible by one camera are obscured in the other camera. These features are not viable for utilization with stereo vision but can be exploited by monocular vision. Rendering the virtual boom allows the capability to quantify not only pixels obscured in both cameras but also to measure the pixels seen in one camera but not the other.

The work presented in this paper makes the following contributions: (1) development of a real-time approach to augmenting real-world images with 3D virtual objects, (2) utilizing augmented images to test occlusion effects of objects not present in the original images, and (3) quantifying the intersection and disjunction of pixels being occluded with a stereo vision system.

2 Related Work

Typically, when discussed in augmented reality literature, occlusion refers to virtual objects occluding the user's vision of objects in the real world [12,20]. For example, a user may be wearing augmented reality glasses, and a virtual menu might appear, blocking the user's view of the real world located behind the menu. Work done in this field has focused on reducing the occlusion by either making the virtual objects transparent, or by utilizing smart computing methods to place the objects such that they do not occlude important features of the real world. Additionally, real objects may appear to occlude virtual objects, confusing the users' perception. In contrast, the approach toward occlusion with respect to AAR relates to a physical object blocking the view of the desired target object. Specifically, the tanker's boom arm occludes the cameras' view of the approaching receiver.

Virtual 3D environments are ubiquitous for simulating real-world scenarios. AAR is a natural candidate for such work as the flight approaches can be deterministically replicated. In the real world, deterministically repeating identical approaches is infeasible; furthermore, real test flights with two aircraft flying refueling approaches is prohibitively expensive. Simulation environments detailed in [3,6,19] for refueling unmanned aerial vehicles (UAV) provide the capability to generate virtual images that vision techniques can utilize.

With respect to utilizing computer vision algorithms to aid in AAR, [2] produces pose estimations from Gaussian least squares differential correlation (GLSDC) [11], while [10] utilizes an extended Kalman filter. Instead of using vision algorithms to determine the receiver pose with respect to the tanker, [13] compares the performance of point matching algorithms to determine the tanker's pose from the point of view of the receiver. In [7,8], virtual 3D environments simulate AAR with the capability of capturing virtual images via a Digital Frame Grabber which in turn are sent to various feature and pose estimation algorithms.

Previous approaches at the Air Force Institute of Technology (AFIT) have focused on utilizing computer vision techniques such as stereo block matching to generate disparity maps that produce a sensed point cloud [15,16]. A truth point cloud generated from the known geometry of the receiver (shown in Fig. 2) can then be registered with the sensed point cloud via iterative closest point (ICP). The six degrees of freedom (DoF) vector returned is the relative $x, y, z, roll, pitch, yaw$ between the tanker and receiver – this is the most important product from the vision system. Work has been done with electro-optical (EO) and infrared (IR) sensors [5]; however, this project focuses solely on augmenting the EO imagery.

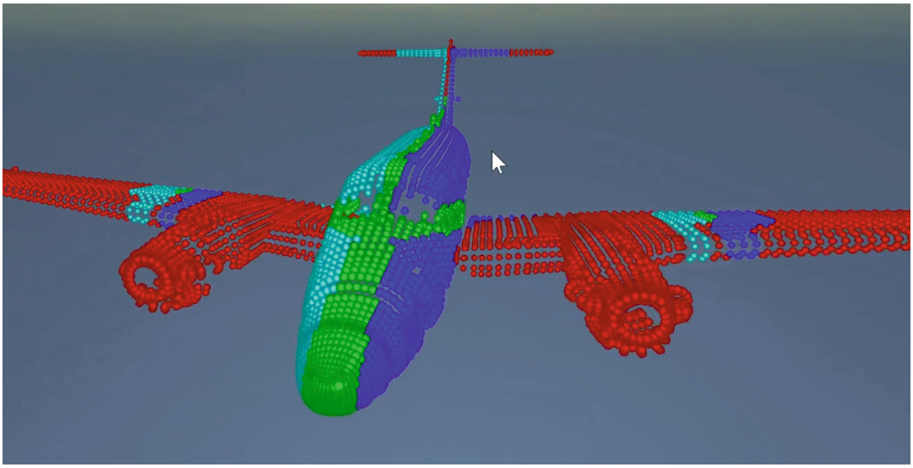


Fig. 2. Visualization of shadow volume method for determining occluded areas. (Color figure online)

Other work undertaken at AFIT, specifically in the AAR domain to reduce the boom's effect on stereo image processing, has involved calculating the volume of space occluded by the boom. In [18], shadow volumes were generated by casting rays from the point of view of the camera. Since these volumes represent the space occluded in either camera [4], any features detected within the volume can be discarded. An example of this technique is shown in Fig. 2 where red points can be seen by both cameras, green points by neither, and cyan and blue points can only be seen from one camera's point of view. In contrast to this method, augmenting the real images transforms a 3-dimensional problem into just 2-dimensions, whereas the shadow volume approach is done entirely in 3-dimensions. Additionally, the features detected within the shadow volume must go through stereo block matching first before they are discarded as viable points. With the augmented images, occluded pixels can be detected before the vision algorithm. Reducing the number of pixels input to the stereo vision algorithm decreases the computational time for stereo feature matching. Finally, pixels corresponding to the boom will not be matched to features on the plane since those pixels are not utilized in the image processing, increasing the accuracy of feature matching.

With respect to object occlusion, literature typically discusses methods to reconstruct the occluded object as seen in [9] where structure-from motion techniques are utilized for aiding robotic systems. This example demonstrates the capability to extend work done augmenting real imagery with virtual objects into other fields such as robotics. Another approach detailed in [21] utilizes single view approaches to generate 3D point clouds and model occlusion. A technique is presented in [1] to reduce the weight of occluded pixels during feature mapping with stereo matching. The work on augmented boom images can similarly improve matching algorithms by rejecting certain pixels that are known to be subject to occlusion.

3 Methodology

Previous work on AAR at AFIT has resulted in a simulation environment capable of playback of truth data collected from Global Positioning System (GPS) and inertial measurement unit (IMU) devices as well as visualization of point-clouds generated from stereo vision image processing. The application development has been realized with AfterBurner [14], a graphics engine built in C++ utilizing OpenGL for rendering. In addition, the stereo vision pipeline utilizes OpenCV. Figures 1, 2, 4 and 6 were generated with the AfterBurner engine.

Simulating the AAR approaches in the virtual world, the aircraft position and orientation can be replayed from data logs to accurately visualize the approach. Virtual cameras are placed on the refueling tanker, and with their orientation and view frusta conforming to their real-world counterparts, the images captured by these virtual cameras will be identical to the real images.

3.1 Test Flights

In March of 2019, researchers conducted aerial refueling test flights at Edwards Air Force Base. The aircraft involved in these flights were two Beechcraft, one as the receiver and one as the tanker with stereo EO and IR cameras attached. Figure 3 depicts the stereo camera setup for the test flights with the cameras mounted to the underbelly of the aircraft below the co-pilot's seat. Imagery was captured from these cameras and linked to truth data collected from GPS and IMU devices, resulting in accurate pose truth data for each aircraft.

Since a Beechcraft substituted an actual tanker in the test flights, it was impractical to attach a boom. Thus, the test images do not contain a boom occluding the receiver. In order to examine the effects of occlusion from the real test flights, an augmented reality solution detailed below was conceived with the goal of quantifying the occlusion of the receiver caused by the boom.

3.2 Virtual Cameras

In synthesizing real images with virtual objects, one objective is to render the virtual 3D models geometrically perspective-correct onto the real images. To achieve this result, a virtual construct is generated consisting of a quad or virtual "green screen" (see Fig. 4). The quad is simply a 2D textured planar rectangle serving as a surface to render the real images. This quad is placed near the far plane of a virtual frustum which shares the same aspect ratio and field of view as a virtual camera. The quad is oriented and locked with respect to the frustum such that it remains static when viewed from the frustum origin. The scene is then rendered with an OpenGL frame-buffer object (FBO) with the camera position and direction set to the frustum pose. Thus, as long as the field of view of the frustum is identical to the real world cameras capturing imagery, any virtual objects within the view frustum can be rendered perspective-correct into the real images. A summary of the image pipeline is shown in Fig. 5.

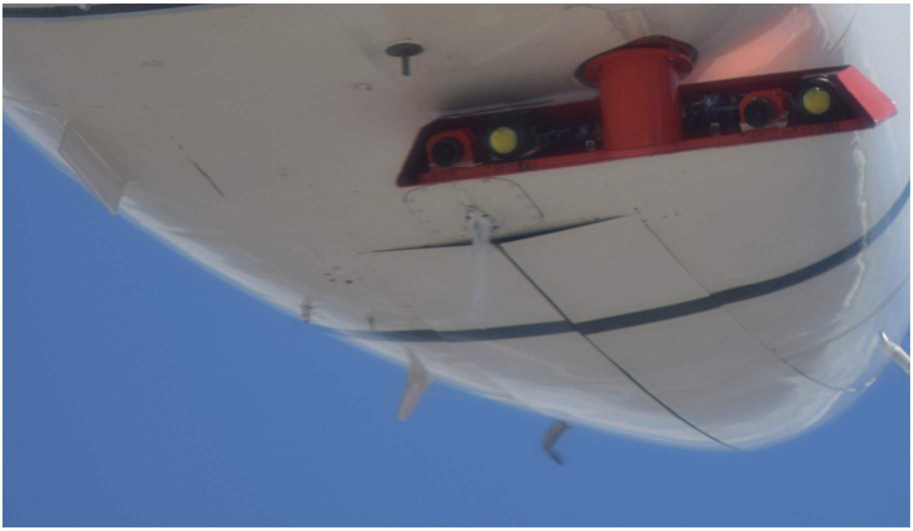


Fig. 3. Reference image of stereo EO and IR cameras mounted underneath the co-pilot.

Figures 4 and 6 depict the virtual camera arrangement for creating augmented images of aerial refueling test flight images. The images appearing in the upper corners are the raw image data captured during the test flight. Below the aircraft are visualizations of the stereo camera frusta, and positioned just in front of the far plane, quads utilize the raw stereo images as textures. The rationale for placing the quads just in front of the far plane (the distance from the frustum origin to the quad is 99% the distance to the far plane) is to minimize any errors caused by floating point precision loss which may cause the quad to not correctly render from the perspective of the virtual camera.

Near the bottom of Fig. 6 and appearing with black borders are the resulting render from the point of view of the virtual cameras located on the bottom of the plane. For this figure, the boom has been selectively rendered into the FBOs, whereas the Beechcraft model was excluded from the render. If the Beechcraft model were rendered, it would perfectly align with its real-world counterpart when visualized inside the augmented images. Note that since Fig. 6 is rendered with the world camera's pose different than the virtual cameras', the virtual Beechcraft is not aligned with the real aircraft. If the virtual model did not line up with the real aircraft image, it could be indicative of an error in the camera's parameters such as the position and rotation. Thus, the augmented image system also allows researchers to visually validate the correctness of the truth data collected during the test flights.

Some camera parameters were chosen based on the real camera measurements, while some were arbitrary. The near-plane for the virtual camera was set to 1 m as no objects, real nor virtual, came within 1 m of the camera. The far-plane distance and virtual quad position were set to 100 m from the camera.

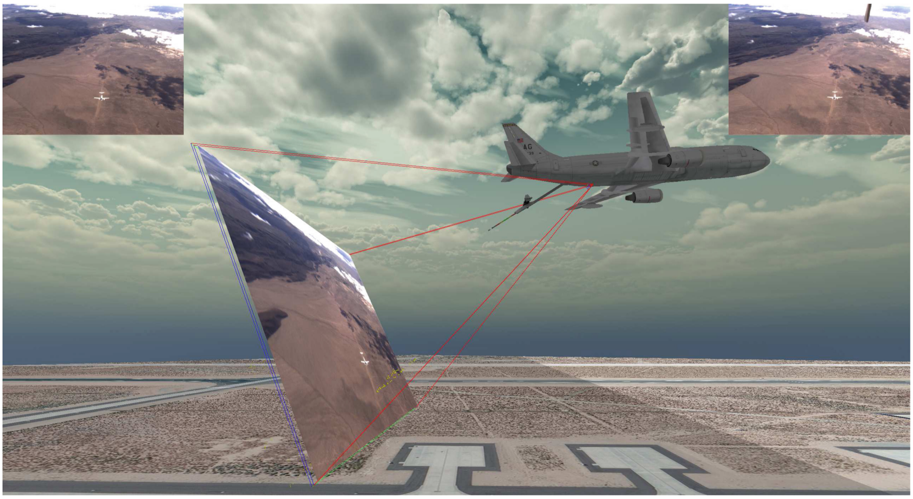


Fig. 4. Depiction of virtual quad with real imagery rendered into it. The top two quads contain the raw image, and the quads behind the image use the raw image as their texture.

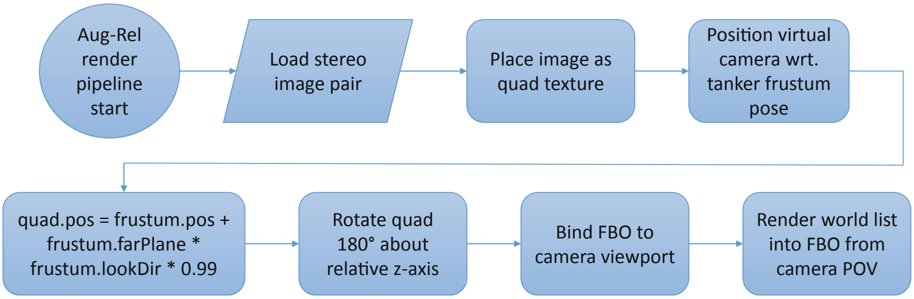

Aug-Rel render pipeline start
Load stereo image pair
Place image as quad texture
Position virtual camera wrt. tanker frustum pose
quad.pos = frustum.pos + frustum.farPlane * frustum.lookDir * 0.99
Rotate quad 180° about relative z-axis
Bind FBO to camera viewport
Render world list into FBO from camera POV

Fig. 5. Augmented reality render pipeline

This value was influenced by two factors. First and most importantly, the refueling envelope begins at 100 m from the tanker, thus the test flights focused on approaches with the receiver closer than 100 m. Second, a larger far-plane value would significantly increase the size of the quad and texture and likely decrease performance of the visualization. The horizontal field of view of the cameras utilized during the test flights was 56°. The aspect ratio of the real cameras was 4:3, and the resolution was 1280 by 960. The virtual camera was given this same aspect ratio, and the FBO texture the augmented images were rendered into were set to the same resolution.

Because the real image augmented with the virtual boom goes through the OpenGL rendering pipeline, the resulting composite image can go through multi-sample anti-aliasing (MSAA). As shown in [17], when stereo vision processes are

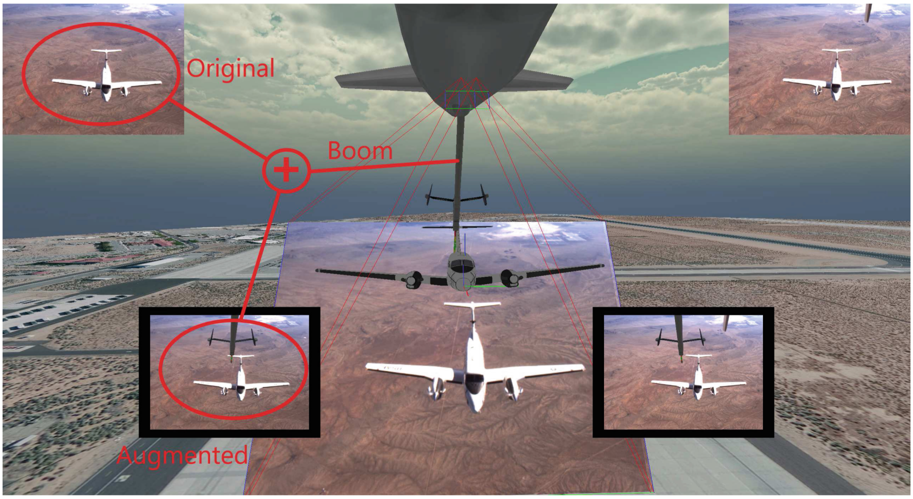


Fig. 6. A virtual model of a Beechcraft is placed in the truth location with respect to the tanker. Although this object falls within the frusta, it can be omitted from rendering into the augmented textures, which are rendered at the bottom with black borders.

performed on virtual objects, performing MSAA on the images before imparting them to the vision pipeline significantly reduces the error in the resulting point cloud.

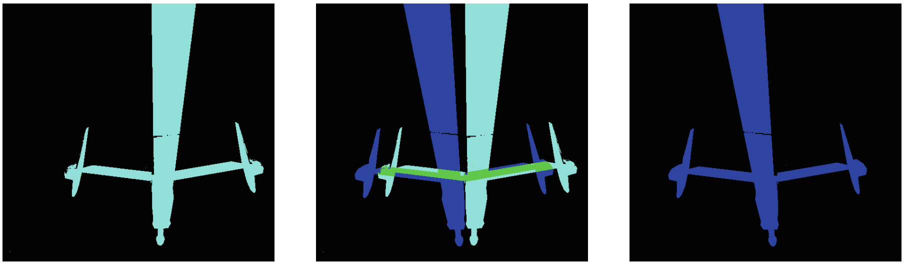
4 Results

To evaluate the augmented images, simple image processing was performed to create masks for the boom and receiver. The boom mask was obtained by subtracting the original image from the augmented image. Since the pixels in the augmented image correspond to those in the original image that have the same value, they will become zero. Elsewhere, the pixels will likely be non-zero and thus can be utilized to create a mask. The original images were captured with a resolution of 1280 by 960, and Figs. 7a, b, c and 8a and b depict cropped segments of the original, so the resolutions will be less than 1280 by 960.

Figure 7a, b and c depict the masks created by subtracting the augmented images from the original images in RGB color-space. The figures have been cropped, focusing on the boom to enhance visibility. The left and right masks are colored cyan and blue, respectively, and when these two images are again subtracted, the overlapping areas appear green as shown in Fig. 7b. From this method, it is easy to visualize which pixels in each image are unavailable for stereo image processing. Black pixels are not occluded in either camera. Green pixels represent features occluded by the boom in both images, thus they are unsuitable for stereo image processing. Cyan and blue pixels correspond to features the boom occludes in either just the left or right image. These features are also unsuited for stereo processing since the boom blocks the feature in exactly one of the cameras; however, this feature could be utilized with monocular vision processing such as object or motion tracking if it were matched to the previous frame.

In addition to measuring the number of pixels occluded by the boom, the augmented imagery may be utilized to visualize and quantify areas of a specific object occluded by the boom. In Fig. 8a and b the masks from Fig. 7a and c have been added to the mask of the plane in RGB color-space. In Fig. 8a, the overlapping pixels of the plane and boom appear white, while in Fig. 8b, they appear magenta.

Table 1 presents data from one image pair regarding the number and percentage of pixels quantified by the masks. The first four entries show the number and percent of pixels which the plane and boom cover in the left and right images, respectively. The fifth entry shows the number of pixels overlapping from the left and right boom images. The percentage is how many pixels overlap compared to how many are contained within the boom masks on both images. The final two entries are how many pixels the boom occludes the plane in the left and right images, and the percentage shown is the percentage of how many pixels of the plane are occluded.

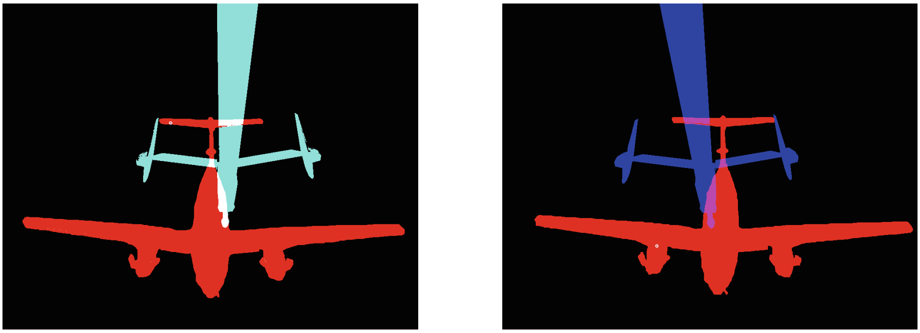


(a) Boom rendered cyan from POV of left camera.

(b) Combined render of boom from left and right cameras; green pixels are where boom overlaps in each camera.

(c) Boom rendered blue from POV of right camera.

Fig. 7. Masks generated of boom from POV of left and right cameras (Color figure online)



(a) From left camera POV, mask of boom (cyan) and plane (red), added creating overlapping pixels (white).

(b) From right camera POV, mask of boom (blue) and plane (red), added creating overlapping pixels (magenta).

Fig. 8. Visualization of pixels occluded in each camera frame (Color figure online)

Table 1. Pixel counts and percentages of image masks

Image	Pixels	Percent	Percent reference
Plane left (red pixels in Fig. 8a)	1164420	31.59%	Whole image
Plane right (red pixels in Fig. 8b)	1164983	31.60%	
Boom left (cyan pixels in Fig. 7a)	40970	1.11%	
Boom right (blue pixels in Fig. 7c)	42710	1.16%	
Boom overlap (green pixels in Fig. 7b)	5609	7.18%	Boom pixels
Boom occlusion left (white pixels in Fig. 8a)	2866	0.25%	Plane pixels
Boom occlusion right (magenta pixels in Fig. 8b)	3270	0.28%	

5 Conclusion

Combining real world imagery with virtual 3D models provides simulations with the capability to examine the obscuring effect caused by those objects if they had been present in the original scene. In order to accurately place the virtual objects into the real scene, the models must be rendered perspective-correct with respect to how the original images were captured. By placing the real imagery as a texture behind the virtual objects and locking the texture with respect to the camera's view point, the scene can be rendered with the blend of real and virtual objects. In the case of AAR, images captured from actual test flights can be enhanced to visualize and test the effect a boom arm occludes. This process can also be utilized to mitigate the effect the boom has on stereo vision processing by eliminating specific pixels from further feature matching, since it is known those pixels are being occluded by the boom. While this process was developed specifically for augmenting refueling images, it could be extended to other topics such as robotics and augmented reality.

References

1. Bobick, A.F., Intille, S.S.: Large occlusion stereo. Int. J. Comput. Vision **33**, 181–200 (1999). https://doi.org/10.1023/A:1008150329890
2. Campa, G., Mammarella, M., Napolitano, M.R., Fravolini, M.L., Pollini, L., Stolarik, B.: A comparison of pose estimation algorithms for machine vision based aerial refueling for UAVs. In: 14th Mediterranean Conference on Control and Automation, MED 2006 (2006). https://doi.org/10.1109/MED.2006.328769
3. Campa, G., Napolitano, M.R., Fravolini, M.L.: Simulation environment for machine vision based aerial refueling for UAVs. IEEE Trans. Aerosp. Electron. Syst. **45**, 138–151 (2009). https://doi.org/10.1109/TAES.2009.4805269
4. Crow, F.C.: Shadow algorithms for computer graphics. ACM SIGGRAPH Comput. Graphics **11**, 242–248 (2005). https://doi.org/10.1145/965141.563901
5. Dallmann, W.E.: Infrared and electro-optical stereo vision for automated aerial refueling. Master's thesis, Air Force Institute of Technology (2019)
6. Duan, H., Zhang, Q.: Visual measurement in simulation environment for vision-based UAV autonomous aerial refueling. IEEE Trans. Instrum. Meas. **64**, 2468–2480 (2015). https://doi.org/10.1109/TIM.2014.2343392
7. Fravolini, M.L., Brunori, V., Ficola, A., La Cava, M., Campa, G.: Feature matching algorithms for machine vision based autonomous aerial refueling. In: 14th Mediterranean Conference on Control and Automation, MED 2006 (2006). https://doi.org/10.1109/MED.2006.328792
8. Fravolini, M.L., Campa, G., Napolitano, M.R.: Evaluation of machine vision algorithms for autonomous aerial refueling for unmanned aerial vehicles. J. Aerosp. Comput. Inf. Commun. **4**, 968–985 (2008). https://doi.org/10.2514/1.17269
9. Huang, X., Walker, I., Birchfield, S.: Occlusion-aware reconstruction and manipulation of 3D articulated objects. In: Proceedings - IEEE International Conference on Robotics and Automation (2012). https://doi.org/10.1109/ICRA.2012.6224911
10. Johnson, D.T., Nykl, S.L., Raquet, J.F.: Combining stereo vision and inertial navigation for automated aerial refueling. J. Guidance Control Dyn. **40**, 2250–2259 (2017). https://doi.org/10.2514/1.g002648
11. Kimmett, J., Valasek, J., Junkins, J.: Autonomous aerial refueling utilizing a vision based navigation system. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. American Institute of Aeronautics and Astronautics (2002). https://doi.org/10.2514/6.2002-4469
12. Lepetit, V.: On computer vision for augmented reality. In: Proceedings - International Symposium on Ubiquitous Virtual Reality, ISUVR 2008 (2008). https://doi.org/10.1109/ISUVR.2008.10
13. Mammarella, M., Campa, G., Napolitano, M.R., Fravolini, M.L.: Comparison of point matching algorithms for the UAV aerial refueling problem. Mach. Vis. Appl. **21**, 241–251 (2010). https://doi.org/10.1007/s00138-008-0149-8
14. Nykl, S., Mourning, C., Leitch, M., Chelberg, D., Franklin, T., Liu, C.: An overview of the STEAMiE educational game engine. In: Proceedings - Frontiers in Education Conference, FIE (2008). https://doi.org/10.1109/FIE.2008.4720454
15. Parsons, C.A.: Improving automated aerial refueling stereo vision pose estimation using a shelled reference model. Master's thesis, Air Force Institute of Technology (2017)
16. Parsons, C., Nykl, S.: Real-time automated aerial refueling using stereo vision. In: Bebis, G., et al. (eds.) ISVC 2016. LNCS, vol. 10073, pp. 605–615. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-50832-0_59

17. Parsons, C., Paulson, Z., Nykl, S., Dallman, W., Woolley, B.G., Pecarina, J.: Analysis of simulated imagery for real-time vision-based automated aerial refueling. J. Aerosp. Inf. Syst. **16**(3), 77–93 (2019). https://doi.org/10.2514/1.I010658
18. Paulson, Z., Nykl, S., Pecarina, J., Woolley, B.: Mitigating the effects of boom occlusion on automated aerial refueling through shadow volumes. J. Defense Model. Simul. **16**, 175–189 (2019). https://doi.org/10.1177/1548512918808408
19. Pollini, L., Campa, G., Giulietti, F., Innocenti, M.: Virtual simulation set-up for UAVs aerial refuelling. In: AIAA Modeling and Simulation Technologies Conference and Exhibit. American Institute of Aeronautics and Astronautics (2012). https://doi.org/10.2514/6.2003-5682
20. Shah, M.M., Arshad, H., Sulaiman, R.: Occlusion in augmented reality. In: 8th International Conference on Information Science and Digital Content Technology (ICIDT 2012) (2012)
21. Zanfir, A., Sminchisescu, C.: Large displacement 3D scene flow with occlusion reasoning. In: Proceedings of the IEEE International Conference on Computer Vision (2015). https://doi.org/10.1109/ICCV.2015.502