A SIMULATION ENVIRONMENT FOR DRONE CINEMATOGRAPHY

F. Zhang, D. Hall, T. Xu, S. Boyle and D. Bull

Bristol Vision Institute, University of Bristol, UK

ABSTRACT

Simulations of drone camera platforms based on actual environments have been identified as being useful for shot planning, training and re-hearsal for both single and multiple drone operations. This is particularly relevant for live events, where there is only one opportunity to get it right on the day. In this context, we present a workflow for the simulation of drone operations exploiting realistic background environments constructed within Unreal Engine 4 (UE4). Methods for environmental image capture, 3D reconstruction (photogrammetry) and the creation of foreground assets are presented along with a flexible and user-friendly simulation interface. Given the geographical location of the selected area and the camera parameters employed, the scanning strategy and its associated flight parameters are first determined for image capture. Source imagery can be extracted from virtual globe software or obtained through aerial photography of the scene (e.g. using drones). The latter case is clearly more time consuming but can provide enhanced detail, particularly where coverage of virtual globe software is limited. The captured images are then used to generate 3D background environment models employing photogrammetry software. The reconstructed 3D models are then imported into the simulation interface as background environment assets together with appropriate foreground object models as a basis for shot planning and rehearsal. The tool supports both free-flight and parameterisable standard shot types along with programmable scenarios associated with foreground assets and event dynamics. It also supports the exporting of flight plans. Camera shots can also be designed to pro-vide suitable coverage of any landmarks which need to appear in-shot. This simulation tool will contribute to enhanced productivity, improved safety (awareness and mitigations for crowds and buildings), improved confidence of operators and directors and ultimately enhanced quality of viewer experience.

INTRODUCTION

Recently, unmanned aerial vehicles (UAVs) have been frequently employed as camera platforms in film and broadcast production to capture content offering an enhanced viewing experience. Drones offer flexible camera positioning with multiple angles and uninterrupted coverage, which are important for coverage of live events such as sports. In these cases,

rehearsal opportunities are often limited and there is often only one opportunity for the directors, drone pilots and camera operators to conduct fly and shoot operations. A reliable and realistic simulation tool, supporting the integration of programmable foreground assets into realistic background environments, would therefore be of significant utility for planning, rehearsing and evaluating single and multiple drone operation in preparation for these types of event.

There already exist commercial and royalty-free software packages capable of flight simulation within realistic (1) or virtual environments (2, 3). Examples include DJI Flight Simulator (4), Google Earth Studio (5), AirSim (6) and Microsoft Flight Simulator (7). DJI Flight Simulator focuses on training the flight skills of drone pilots, which offers a selection of virtual background scenarios and can simulate various weather conditions. However all its environmental models are not realistic. Google Earth Studio is a web-based animation tool based on Google Earth's satellite and 3D imagery, which can generate videos with an intuitive UI and features such as keyframe-based animation. Similar to Google Earth (8) and Microsoft Bing Maps (9), it provides sufficient resources for multiple view footage without providing any flight training or planning features. AirSim (Aerial Informatics and Robotics Simulation) is a plug-in package for Unreal Engine 4 (10) and Unity (11). It has often been used as a platform for AI and control system research related to autonomous vehicles (12). However it is not furnished with shot type grammars and does not provide realistic environment assets. Microsoft Flight Simulator 2020 is currently under development for Windows 10 and Xbox One platforms, and is claimed to simulate the entire Earth using textures and topographical data from Bing Maps. It is designed primarily for flight training and simulation rather than drone cinematography, lacking features for shot type grammar and flexible foreground object integration.

In this context, based on the preliminary work reported in (13), a new workflow for developing a fully functional CGI-based simulation tool with realistic background environments is presented here. This also includes recommended parameters for the capture of 2D environmental images as a basis for 3D reconstruction. The prototype simulation interface supports flexible planning and training with example background and foreground 3D assets.

The rest of this paper is organised as follows. The proposed workflow for building a simulation tool with realistic background environments is firstly presented, and its primary steps, environmental image capture, 3D model reconstruction and simulation interface are further described in detail together with demonstration results and images. Finally, the conclusion is outlined alongside future research directions.

THE PROPOSED WORK

In order to simulate realistic environments and activities for drone cinematography, a workflow is proposed based on (i) environmental image capture, (ii) 3D reconstruction and (iii) a user-friendly simulation interface. This is depicted in Figure 1.

Given the location of the selected geographical area and the proposed camera parameters, the scanning strategy and its associated flight parameters are first determined
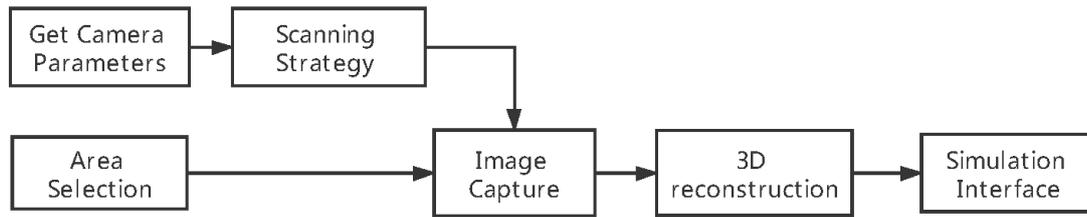
Figure 1 – Diagrammatic illustration of the proposed workflow.

for environmental image capture. The captured images are used to generate 3D background environment models employing photogrammetry techniques. The reconstructed 3D models are then imported into the simulation interface as background environment assets for further shot planning, training and rehearsal.

ENVIRONMENTAL IMAGE CAPTURE

As an important step in photogrammetry, environmental image capture can be either conducted using a drone or helicopter camera platform at the actual geographical location or by photo scanning from specified viewing points within virtual globe software packages such as Google Earth (8) and Microsoft Bing Maps (9). In both approaches, the flight/scanning parameters influence the reconstruction quality of the 3D environment models. These parameters include flight/scanning trajectories, heights, viewing angles and picture overlaps ratios. Due to the limited computational capability, time and resource available, it is important to optimise these parameters to obtain as few images as possible but maintain the required reconstruction quality.

Flight Trajectory

The optimal flight trajectory for a specific background environment is highly dependent on the given landscape and object complexity (14). Notwithstanding this, the most commonly used flight pattern used in practise is grid scanning in two orthogonal horizontal directions. In order to simplify the scanning strategy for both shooting with real drones and capturing within virtual globe software packages, a grid scanning strategy has been employed in this work, as shown in Figure 2.
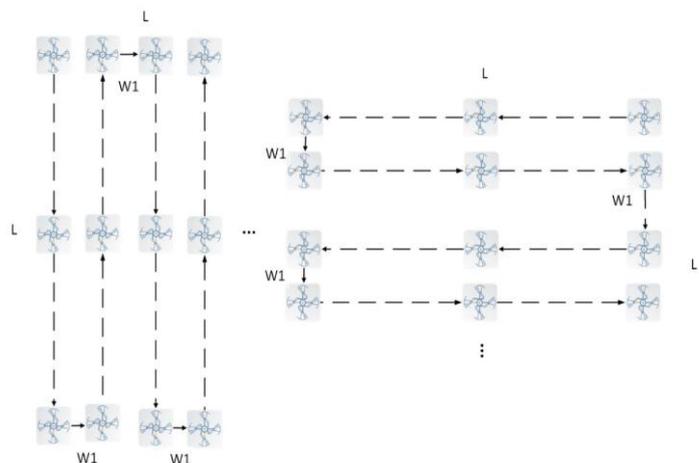

Figure 2 – Grid scanning patterns. To cover of a area with a size of L x L, the drone travels back and forth along straight lines at two orthogonal directions, with a cross-track distance of W1.

Flight Heights

The range of flight heights has been previously recommended in (15), where the results show that a single layer of scanning using a fixed height value can produce reasonably good reconstruction for a woodland

landscape scenario. In this work, to generalise the height configuration for both landscape and urban environments, based on the recommendation in (16), we have employed a three-layer scanning approach.
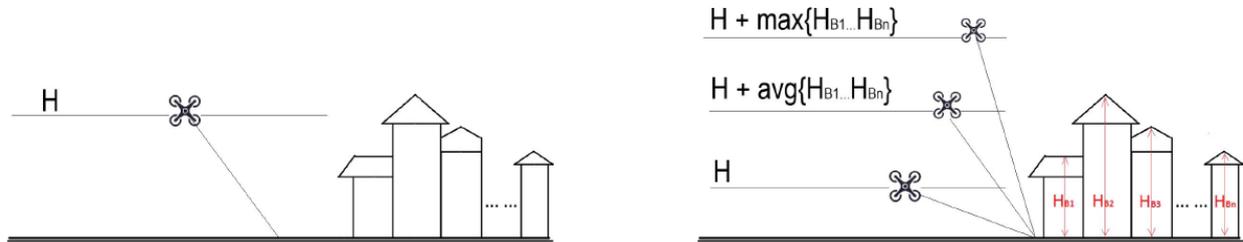

Figure 3 – (Left) Single height scanning in (15). (Right) Multiple height scanning proposed.

The first level, with a height of H, focuses on capturing most of the features of landscape. The second layer scanning (at H + average building height) is used to obtain the details of objects below the average height in the covered area. The highest layer (at H + maximum building height) is used to ensure everything (especially tall objects) can be properly covered during the scanning operation. This configuration is illustrated by Figure 3 and equation (1).

$$\text{Drone Flight Height} = \begin{cases} H, & 1^{\text{st}} \text{ level}; \\ H + \text{average building height}, & 2^{\text{nd}} \text{ level}; \\ H + \text{maximum building height}, & 3^{\text{rd}} \text{ level}. \end{cases} \tag{1}$$

Here a fixed value of 20m is used for H that is within the recommended range in (15) for a default camera sensor size of 23.66 x 13.3mm and a focal length 35mm. This should be adjusted based on the FOV (Field of View) relationship, given by equations (2) and (3), for the actual camera settings used.

$$\text{FL} \times \text{FOV} = \text{SS} \times \text{WD} \tag{2}$$

Here FL and SS stand for focal length and sensor size of the camera respectively, and WD represents working distance (e.g. H in this case). The actual working distance WD_{act} can be calculated by:

$$\text{WD}_{\text{act}} = \frac{\text{SS}_{\text{ref}}}{\text{SS}_{\text{act}}} \times \frac{\text{FL}_{\text{act}}}{\text{FL}_{\text{ref}}} \times \text{WD}_{\text{ref}} \tag{3}$$

in which SS_{ref} and FL_{ref} are default camera parameters as given above, while WD_{ref} is the recommended working distance (e.g. H=20m). SS_{act} and FL_{act} are the actual camera parameters used.

Viewing Angles

Viewing angle (the gimbal rotation angle on the drone) is another important shot parameter in photogrammetry scanning. In order to compare the reconstruction results for various viewing angles, three sets of angle parameters, including 90/67.5/45, 85/60/35, and 70/47.5/25 degrees, were employed, each of which has three different values for three height levels (from highest to lowest respectively) defined above. Other parameters such

Figure 4 – Sample images of the reconstructed models based on images captured using three angle sets. (Left) high angles. (Middle) Low angles. (Right) Intermediate angles.

as flight trajectories, heights and picture overlap ratios are kept identical for three viewing angle sets. Here we used one of the high quality environmental assets Country Side in UE4 market place (17) as the source environment (18). The captured images for each test set have been employed as inputs to a reconstruction software package, 3DF Zephyr (19), to generate a 3D environmental model.

Example images of the reconstructed 3D models for three different viewing angle sets are shown in Figure 4. It can be observed that high angles (90/67.5/45 degrees) lead to significant distortions of the landscape and the road, but provide good reconstruction for certain local details. The low angles (70/47.5/25 degrees) perform well on the shape of landscape but there are localised areas with high distortions. The intermediate angles (85/60/35 degrees) offer the best overall performance, with better reconstruction of landscape and less local detail loss.

Overlapping Ratios

As shown in Figure 5.(left), when environmental images are captured, the overlap between adjacent images (or video frames) captured in the same flight track is defined as in-track overlap, while the cross-track overlap is defined as the overlap between the adjacent images (or video frames) captured at neighbouring flight tracks. It is noted that these two overlap ratios are both related to the total number of images, which determine the reconstruction quality and efficiency.
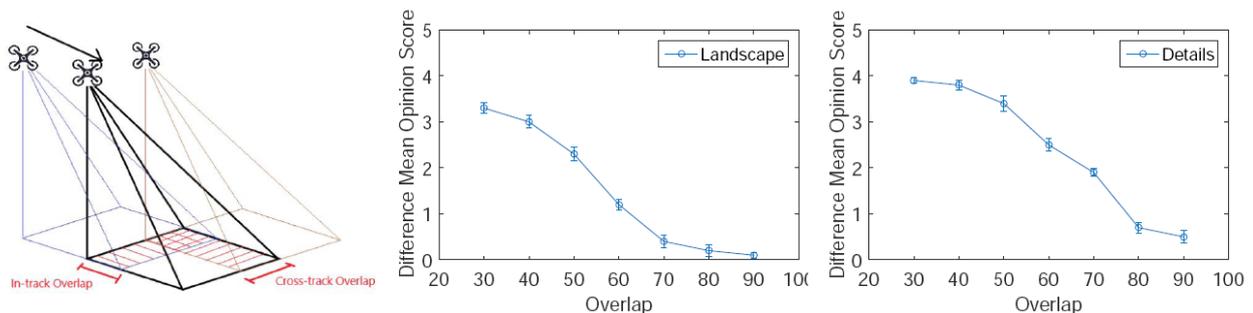

Figure 5 – (Left) Illustration of in-track and cross-track overlaps. (Middle, Right) The DMOS scores collected for 3D models based on different in-track overlap ratios.

The selection of both overlap ratios has been previously reported (16, 20, 21), with recommendations between 60% and 75% for practical use. To further investigate their influence on 3D reconstruction quality, seven different in-track overlap values (30%, 40%, 50%, 60%, 70%, 80% and 90%) were employed to capture 2D images for a selected small
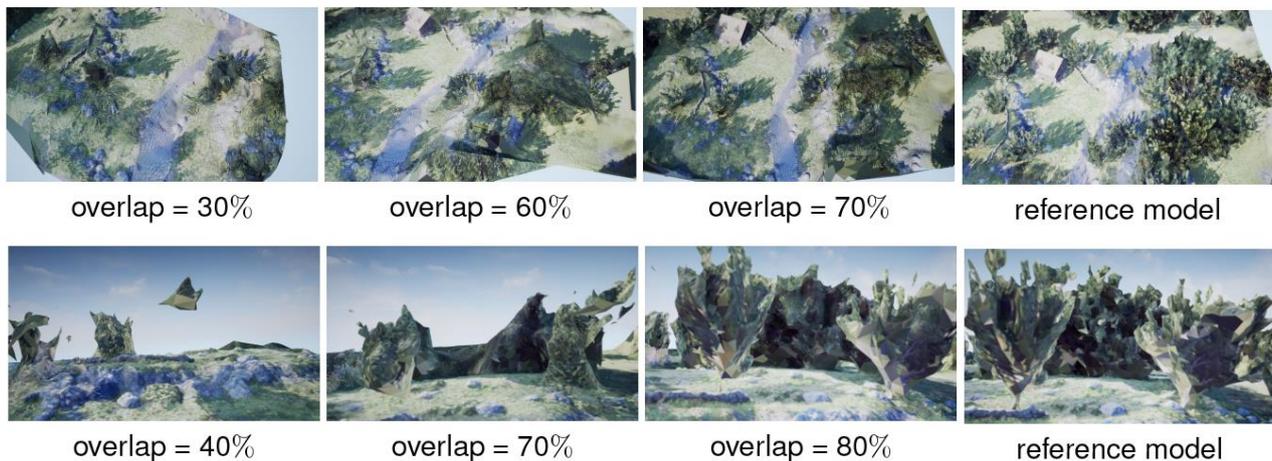
Figure 6 – Example images showing 3D models based on different in-track overlap ratios. (Top) for landscape. (Bottom) for details.

area (50mx50m) within the Country Side environmental asset from the UE4 marketplace (17). We used the same flight height and trajectories, viewing angles and cross-track overlap (70%). Cross-track overlap ratio is fixed here in order to solely test in-track overlap. Actually, the experimental results for in-track overlap will be still valid for cross-track overlap if image size and aspect ratio are taken into account.

Based on each reconstructed model (and within the original 3D asset), a ten second free flying shot was generated using UE4, with identical flight path. All eight video clips (seven test videos plus an original) were viewed by 15 participants using a double stimulus continuous quality scale (DSCQS) methodology (22). All the participants were requested to provide subjective scores for both landscape shapes and local details.

The collected DMOS (Differential Mean Opinion Score) results, comparing the reference video (ground truth) and each test version for both landscape and details, are plotted in Figure 5.(middle, right). It can be observed that, as the in-track overlap ratio increases, the DMOS values becomes lower for both landscape and details. To achieve relatively high reconstruction quality with DMOS being lower than 1, the overlap ratio needs to be greater than 70% for landscape and 80% for details. These characteristics can also be observed in Figure 6, where example figures of the reconstruction models using different in-track overlap ratios are illustrated.

3D MODEL RECONSTRUCTION

Pre-processing

The input image dataset obtained may contain artefacts due to photogrammetry errors or object motion, and these can result in significant distortions during reconstruction. Such defects can be removed or corrected through texture in-painting (23, 24). It is noted that most in-painting algorithms are relatively complex and time consuming when processing a large number of images. During the generation of our demonstration results, in order to achieve efficient reconstruction, simple manual outlier rejection was applied instead on the input image dataset.

Photogrammetry Reconstruction

Numerous open source and commercial photogrammetry software packages are available for producing 3D models from multiple view 2D images. Notable examples include Autodesk ReCap (25), 3DF Zephyr (19), and Pix4D Mapper (26). Autodesk Recap produces relatively poor results from Google Earth captured images, and requires 'cloud credits' to perform online analysis on AutoDesk servers. 3DF Zephyr Aerial generates 3D models with improved quality, especially for object-based scenarios (13). However it requires excellent local graphical calculation capability and large GPU memory for processing. Pix4D Mapper has been specifically designed for professional drone mapping. It creates reconstructions, for the same input images, with equivalent or better quality than 3DF Zephyr and with lower computation complexity. In this work, Pix4D mapper has therefore been adopted as the reconstruction software of choice.

Post Processing

Although photogrammetry software can provide reasonably good reconstruction results, a large number of visible artefacts (e.g. bumps and holes) remain and these can impair viewing experience. 3D model editing (e.g. based on Blender (27)) can be employed to further correct these distortions. When an initial reconstructed model is imported into Blender, it can be used to enhance the 3D texture mesh, using features such as Surface Smoothing, Flatten Mesh Modification and Texture Painting. Example results are shown in Figure 7-10.
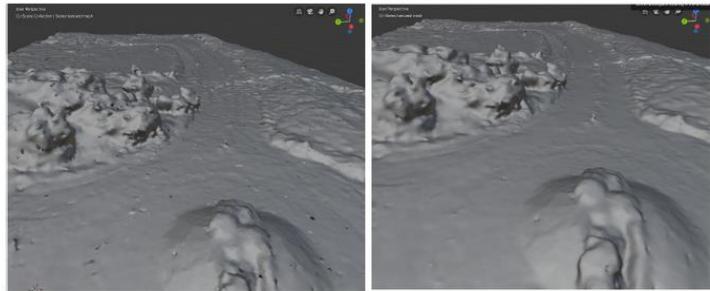

Figure 7 – An example of using the Smooth tool in Blender for post-processing. (Left) The initial 3D model structure with visible bumpy artefacts on the flat surface. (Right) The processed 3D model structure after applying the Smooth tool.

THE SIMULATION INTERFACE

The simulation interface was built using Unreal Engine 4 (10) and has been packaged as a standalone application. This software provides three primary modes: editing, simulation and free play.
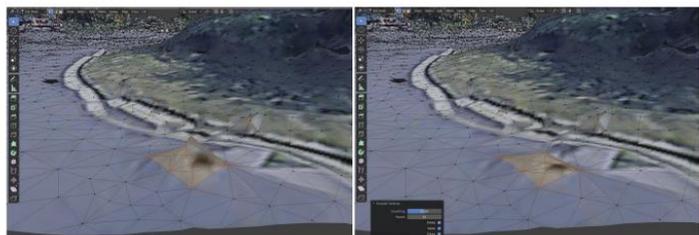

Figure 8 – An example of using the Flatten tool in Blender for post-processing. (Left) 3D Model before Flatten vertices operation. (Right) 3D Model after Flatten vertices operation.

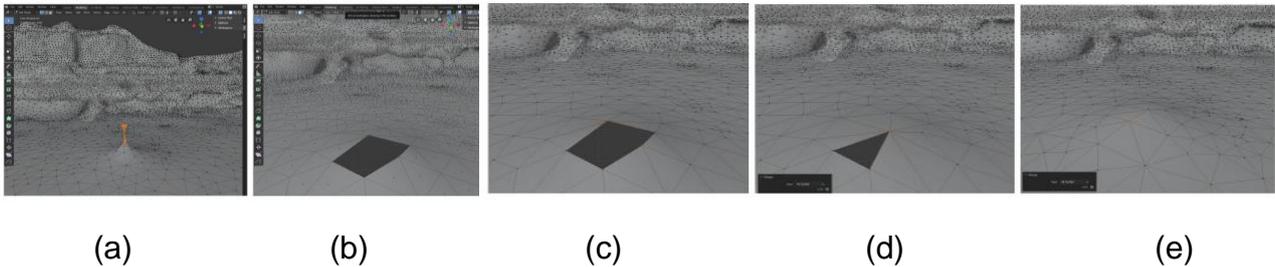|				
(a)	(b)	(c)	(d)	(e)

Figure 9 – An example of mesh modification using Blender. (a) A water surface that has been smoothed but still contains spikes where the mesh is highly distorted. (b) The artefact is deleted. (c-e) It is then filled with the similar textures as in the neighbouring area by applying Merge vertices operation.

Pre-generated 3D Environments and Objects

Two example environments, Clifton Downs and Harbourside (both are within the city of Bristol, UK), have been pre-generated for this simulation tool using the workflow described above. Both of these are reconstructed based on the source data from Google Earth. Example images are shown in Figure 11.

As well as the drone object, a further three foreground objects, car, cyclist and boat, have also been integrated into the software tool. Figure 12 shows example figures for these four objects.

Figure 10 – (Left) A screen shot of the 3D Model before applying Texture Paint with distorted road markings. (Right) A screen shot of the 3D Model after applying Texture Paint

Editing Mode

In editing mode, a customised interface has been configured which allows users to drag and drop objects (e.g. drones, cars, cyclists and boats) into the map, as shown in Figure 13.(a). Once objects are placed in the map, it is possible to move, rotate and scale the entire object and/or edit its associated waypoints. There are also a range of options for each dynamic object edited via the actor options tab. Figure 13.(b) illustrates the options for the drone which include speed, shot type and if the drone will follow the target. Figure 13.(c) shows the options for other objects (e.g. car, cyclist and boat), such as speed and path configuration. There are also additional options for the map visuals such as the time of day, lighting, cloud thickness and cloud speed, which are shown in Figure 13.(d).

In order to support simulation of typical shot types in drone cinematography, a range of typical shot trajectories have been configured. Five examples, including ESTABLISH, CHASE, FLYBY, ELEVATOR and ORBIT, are integrated in the prototype system for the purpose of capability demonstration. All the default parameters recommended are based subjective study results reported in (28) where a camera with 23.66mm x 13.3mm sensor
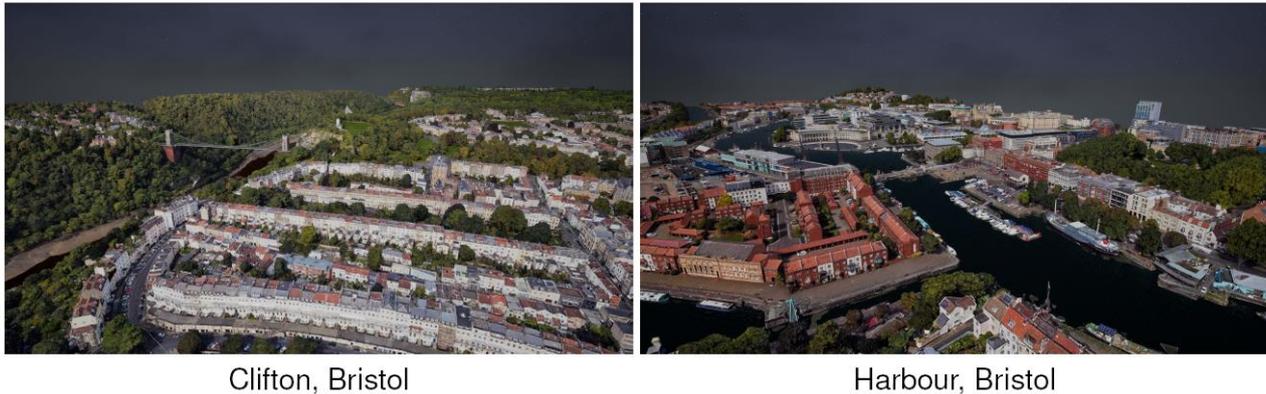
Clifton, Bristol   Harbour, Bristol

Figure 11 – Pre-defined environments.

(a)   (b)   (c)   (d)

Figure 12 – Pre-defined foreground objects: (a) drone (b) car (c) cyclist (d) boat.

size and a focal length of 35mm was employed. When different camera setting is selected, these shot parameters will be re-calculated based on the Field of View (FOV) formula show in equation (3).

Simulation Mode

Once the simulation begins, each object within the map will move along the path configured by the user. There is an option to view the simulation either from the floating camera or from a drone perspective. An option is also available to view from the perspective of more than one drone by using the camera window in the top left of the screen. The option interface for the simulation mode is illustrated in Figure 14.(a)-(c).

Free Play Mode

Free play is similar to the simulation mode except that the user has the option to manually control one of the drones (keyboards and game controllers are currently supported; in future a full drone controller interface will be developed). All the other dynamic objects will behave the same as they would in the simulation mode. There are additional options for the free play mode which include wind parameters and the ability to record the flight path of the manually controlled drone (for off-line evaluation). When the free play mode ends, it is possible to recreate and edit the flight path within editing mode. The option interface for the free play mode is shown in Figure 14.(d).
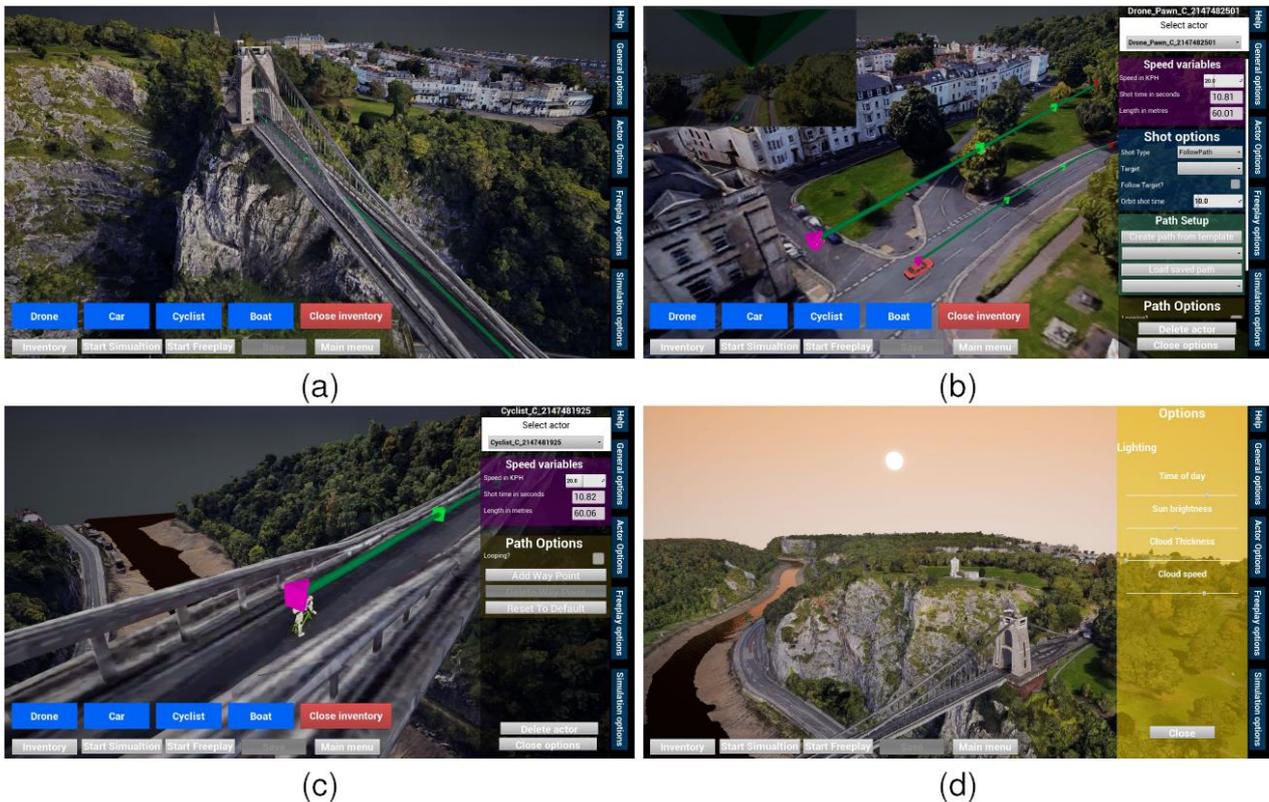
Figure 13 – (a) The custom interface at the editing mode. (b) The option interface for a drone. (c) The option interface for a cyclist object. (d) The additional options for the map visuals.

User Feedback and Evaluation

The current version of this simulation software was demonstrated at the Bristol Drone Cinematography Workshop held in Bristol, UK, in December 2019. Positive (informal) feedback was obtained from all delegates who viewed the demonstration including five international drone cinematography experts. The main challenges remaining are to streamline the environment creation process for users to easily create their own environments, to integrate this software with VR devices to achieve more immersive visual experiences, and to enable its compatibility to primary auto pilot software for autonomous drone operation. Demo videos are available at https://vilab.blogs.bristol.ac.uk/?p=2456 for access.

Conclusion

In this paper, a flexible and functional drone training, pre-visualisation and planning simulator is described that is capable of simulation based on actual environments. The tool allows creation of accurate real world environments, and the incorporation of programmable foreground assets as filming targets. It also supports paramaterisable pre-programmed shot types within a user friendly interface which is built upon UE4. Future work will focus on the development of a more flexible environment creation workflow to enable users to create 3D maps for specific areas, and the interface to an HMD to enable a VR platform to achieve more immersive visual experiences.
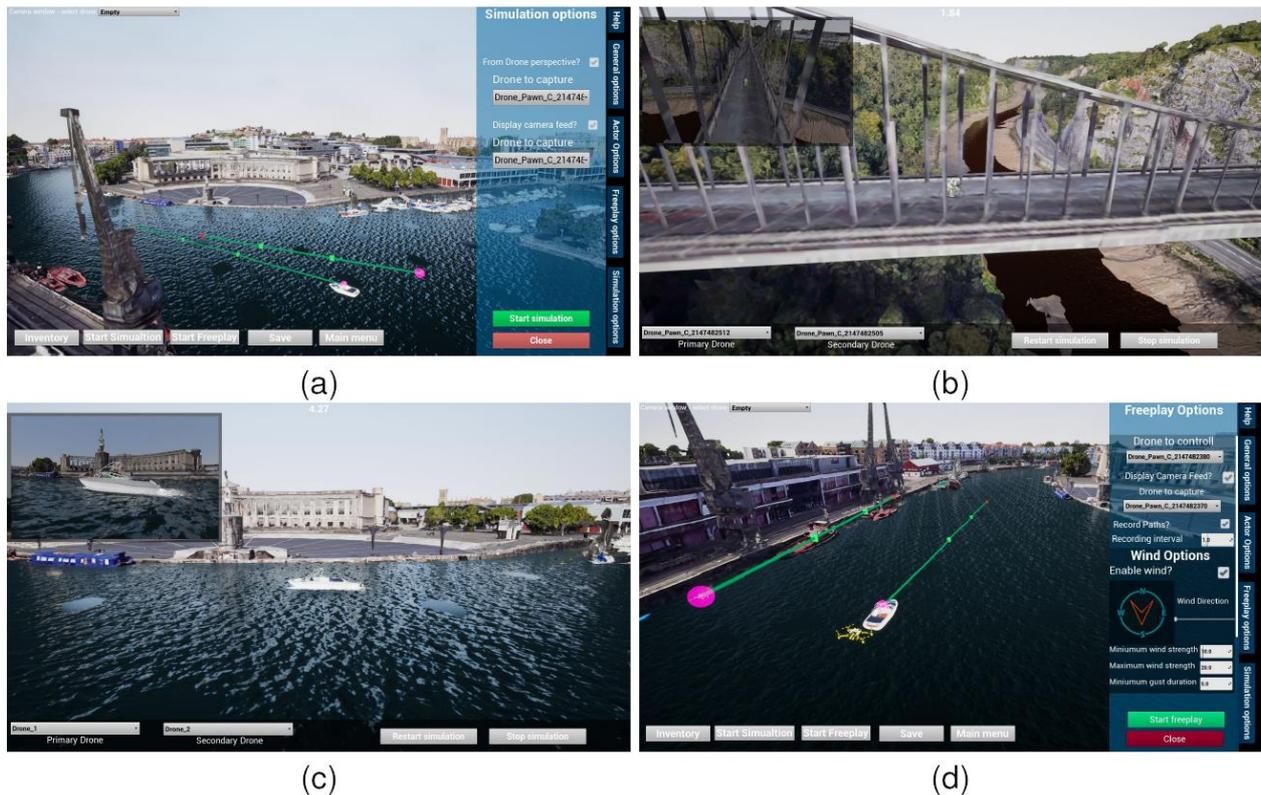
Figure 13 – (a) The option interface in the simulation mode. (b) A screen shot of simulation mode interface when a cyclist object is moving on a bridge. (c) A screen shot of simulation mode interface when a boat is sailing in a river. (d) The free play option interface.

REFERENCE

1. Nägeli, T., Meier, L., Domahidi, A., Alonso-Mora, J. and Hilliges, O.,2017. Real-time planning for automated multi-view drone cinematography. ACM Transactions on Graphics, vol. 36, no. 4, p. 132.

2. The Real Team, 2019. Real drone simulator. https://www.realdronesimulator.com/.

3. Hobby, H., 2019. Real flight. https://www.realflight.com/.

4. SZ DJI Technology Co, 2019. Dji flight simulator. https://www.dji.com/uk/simulator

5. Google, 2018. Google earth studio. https://earth.google.com/ studio/docs/

6. Microsoft, 2019. AirSim. https://microsoft.github.io/AirSim/

7. Microsoft, 2019. Microsoft flight simulator. https://www.flightsimulator.com/.

8. Google,2018. Google earth. https://www.google.com/earth/.

9. Microsoft, 2019. Bing maps. https://www.bing.com/maps.

10. Epic Games, 2019. Unreal engine. https://www.unrealengine.com.

11. Unity Technologies, 2019. Unity. https://unity3d.com/.

12. Shah, S., Dey, D., Lovett, C., and Kapoor A., 2017. AirSim: High-fidelity visual and physical simulation for autonomous vehicles.

13. Boyle, S., Newton, M., Zhang, F. and Bull, D. R., 2019. Environment capture and simulation for UAV cinematography planning and training. EUSIPCO.

14. Smith, N., Moehrle, N., Goesele, M. and Heidrich, W.,2018. Aerial path planning for urban scene reconstruction: A continuous optimization method and benchmark. SIGGRAPH Asia, pp. 183.

15. Seifert, E., Seifert, S., Vogt, S., Drew, D., van Aardt, J., Kunneke, A. and Seifert, T., 2019. Influence of drone altitude, image overlap, and optical sensor resolution on multi-view reconstruction of forest images. Remote Sensing, vol. 11, no. 10, pp. 1252.

16. Hawkins, S., 2016. Using a drone and photogrammetry software to create orthomosaic images and 3d models of aircraft accident sites. ISASI Seminar, pp. 1–26.

17. PolyPixel. Countryside. http://www.polypixel3d.com/.

18. Lewis, J., Brown, D., Cranton, W. and Mason, R., 2012. Simulating visual impairments using the unreal engine 3 game engine," IEEE International Conference on Serious Games & Applications for Health.

19. 3dflow, 2019. 3df zephyr. https://www.3dflow.net/

20. Ventura, D., Bruno, M., Lasinio, G. J., Belluscio, A. and Ardizzone, G., 2016. A low-cost drone based application for identifying and mapping of coastal fish nursery grounds. Coastal and Shelf Science, vol. 171, pp. 85–98.

21. James M. R. and Robson, S., 2012. Straightforward reconstruction of 3d surfaces and topography with a camera: Accuracy and geoscience application. Journal of Geophysical Research: Earth Surface, vol. 117, no. F3.

22. Recommendation ITU-R BT.500-12, 2012. Methodology for the subjective assessment of the quality of television pictures, ITU-R Std.

23. Tschumperle, D. and Deriche, R., 2005. Vector-valued image regularization with pdes: A common framework for different applications. IEEE transactions on pattern analysis and machine intelligence, vol. 27, no. 4, pp. 506–517.

24. Wong A. and Orchard, J., 2008. A nonlocal-means approach to exemplar-based inpainting. IEEE International Conference on Image Processing, pp. 2600–2603.

25. AutoDesk, 2018. ReCap: Reality Capture and 3D Scanning Software.

26. Pix4D, 2019. Pix4D mapper. https://www.pix4d.com/

27. Blender, 2019. Blender documentation. https://docs.blender.org/.

28. Boyle, S., Zhang, F. and Bull, D. R., 2019. A subjective study of viewing experience for drone videos. IEEE Int Conf. on Image Processing.

ACKNOWLEDGEMENTS