[DRAFT]

Evaluation of Augmented and Virtual Reality Applications for the Study of Social Impacts of Drone Delivery Networks

Marsh, C. Z. A., Crowther, W. J., Parslew, B.

Abstract

There is significant commercial interest in developing networks of unmanned aerial vehicles for transport tasks in urban environments. The aim of the present work is to develop a plausible drone transport model for an urban area and to investigate the visual impact of this through evaluation in Virtual and Augmented Reality (VR and AR). An agent-based drone transport network is developed, using the Unity real time development environment, for the City of Manchester based on Census data and estimated requirements for small package delivery and security tasks. The physical city environment is provided via a high resolution 3d model of the city centre. Both AR and VR outputs are based on use of existing tools within the Unity environment. AR running on a consumer tablet provides excellent visualisation of the drones against a realistic background with realistic dynamic action of cars and pedestrians, however camera motion tracking is only effective when there is significant texture in the scene. Furthermore, modelling of occlusion by buildings is not straightforward to implement. VR implemented using a dedicated headset and desktop computer allows real time simulation and visualisation of a transport network with many thousands of drones with occlusion and all aspect viewing, but has reduced value for social impact studies due to lack of realistic visual context compared to AR. The AR studies show that, in a busy city environment, the visual impact to pedestrians of small delivery drones away from distribution hubs is relatively minor for drone operating altitudes above 100m. Visual impact is high directly around distribution hubs however these can be placed away from areas of high population density. Visual impact at night depends on standards set for anti-collision lights on drones, and may be higher than for daytime operations.

1 Introduction

The adoption of new technologies is often hindered by speculation about the social impacts of the technology. This work investigates the utility of immersive visualisations of drone operations in a city centre setting for social impact studies. We look to answer the question: how easily can AR and VR be used to render results of simulations?

VR is based on insertion of a user in a fully virtual world whereas AR is based on overlay of visuals onto a position-referenced live camera feed from the real world. The work used Unity, primarily a game development engine, to simulate and render the drones in both VR and AR settings. An agent-based model was developed with high level tasks, such as deliveries, described by sub-tasks which could be executed by the drone agents. Population distribution data for the city centre was obtained from the 2011 UK Census results. This was combined with data on parcel and takeaway deliveries completed in the UK to approximate the demand figures used in task generation.

2 Drone Delivery Model

This section discusses the implementation of the agent-based simulation model for drone delivery traffic in the city centre. The simulation model was developed in the Unity engine using C# code and used a 3D model of Manchester City centre (provided by Arup) for geometry and obstacles in navigation.

2.1 Demand Model

Agent-based drones were assigned tasks involving collection and delivery of parcels, and navigating between points within the city. Idle drones waited in hangars distributed around the delivery sources in the model. For simplicity, the delivery tasks of a day were spread uniform-randomly across a 1-hour window. This allowed for faster simulation of the deliveries, but also would affect the results by increasing the apparent traffic density.

The UK 2011 census gives a total population of 63.182 million and Manchester City centre a total population of 34,947.

Takeaway demand figures were taken from (https://assets.kpmg/content/dam/kpmg/uk/pdf/2021/07/kpmg-food-for-thought-2021.pdf). We assume takeaway ordering is evenly distributed among the population and the days of the year. I.e. there are an equivalent number of orders placed on a Tuesday and a Friday for any given week. Takeaway demand was found to be 1 order per week for 76% of the population. The number of daily orders is found as:

$$\frac{0.76 \times Manchester\ population}{7\ days}$$

$$\frac{0.76 \times 34{,}947}{7\ days} = 3{,}794\ daily\ takeaways$$

Doole et al. (https://www.sciencedirect.com/science/article/pii/S0969699719304004?pes=vor) estimate the number of drone eligible parcels delivered in the UK throughout the year of 2019 to be 1.56 billion. We assume that parcel deliveries are spread evenly across the population. Thus, parcel deliveries in Manchester City centre are found as:

$$\frac{Manchester\ population}{UK\ population} \times total\ deliveries$$

$$\frac{34{,}947}{63{,}182{,}000} \times 1{,}560{,}000{,}000 = 862{,}862$$

This means that Manchester City centre may be estimated to account for 862,862 deliveries in the year 2019. Assuming deliveries are not weighted by the time of year (which they clearly will be, Christmas being a strong example) the daily deliveries can be approximated as:

$$\frac{862{,}862}{365\ days} = 2{,}364\ \text{daily parcel deliveries}$$

2.2 Spatial Model

A 3D model of Manchester City centre (provided by Arup) was used as the setting for the simulation. Takeaway deliveries were sent from locations with high densities of restaurants. Parcel deliveries were treated as coming from a warehouse on the outskirts of the City centre.

Fig. 1 Delivery sources used in the simulation model based on locations in Manchester City centre. Food delivery sources are highlighted orange and a domestic source is blue.

The postcodes M1, M2, M3, and M4 within the city centre were approximated with rectangular grids and divided into equal cells based on the number of households within the postcode. Each cell was used as a potential destination for parcel or food deliveries.

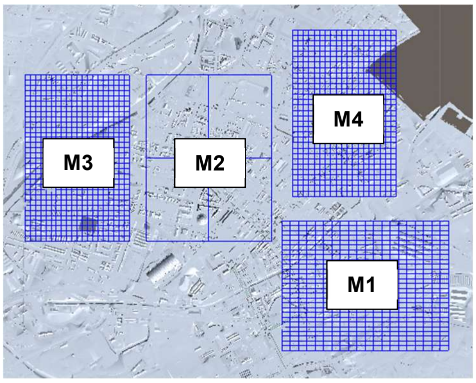

Fig. 2.1 *Postcodes of the city centre divided into cells based on the number of households they contain*

Table 2.1 *Number of households in Manchester City centre postcodes reported in the 2011 UK Census.* (https://www.ons.gov.uk/census/2011census/2011censusdata/2011censusdatacatalogue)

Post Code	Number of households
M1	5233
M2	54
M3	6667
M4	6072

The drone agents navigated the 3D environment using Unity's navigation system, based on the A* algorithm. This is limited to motion along a plane with changes in elevation defined by discrete steps or ramps. A single plane was used for navigation along a fixed height above the ground. It was deemed suitable to consider only the motion of the drones along a fixed height as the initial and final stages of the delivery might considered as being purely vertical motion.

Future implementations of this model could include third party implementations of A*, or other pathfinding algorithms, which are available to purchase on the Unity asset store and provide navigation in 3D spaces.

3 Data Collection

Drone positions were recorded throughout the simulation to be replayed in the AR application. No recording was necessary for VR as the simulation was run during rendering. Three locations were selected within the city for the AR results to be rendered. The locations were:

- Domestic – within the M3 postcode, near large apartment buildings
- Delivery Source – next to one of the takeaway delivery sources
- Central – in the centre of the city, where drone paths would likely intersect

4 Virtual and Augmented Reality

This section presents the method of development for the VR and AR apps which were used to render the simulation results.

4.1 Augmented Reality Development

Augmented Reality uses accelerometer, gyroscope and camera data to determine the motion of the rendering device, which is usually a smartphone or tablet. From this data, the AR application is able to overlay images on the camera feed, moving and rotating them to compensate for the device's motion. This gives the effect of the overlaid images appearing to "exist" in the real world. An AR app only requires the developer to provide models for the overlaid imagery, the rest of the environment is the real world and is captured by the device's camera.

The AR app was built for a Samsung Galaxy Tab S5e running an Android operating system. iOS development was not considered due to the requirement of membership with the Apple Development Program, which costs $99 (USD) per year (https://developer.apple.com/support/enrollment/#:~:text=The%20Apple%20Developer%20Program%20annual,currency%20during%20the%20enrollment%20process). However, the process of development would be similar to the method outlined here, further information may be found in the Unity documentation.

The AR development used the following hardware and software:

- Desktop computer
- Free Unity License with these additional modules/packages installed:

    - AR Foundation
    - Android SDK and NDK
    - OpenJDK
    - ARCore
- Android device with Android version 7.0 or greater and ARCore support (see: https://developers.google.com/ar/devices)

4.2 Virtual Reality Development

The VR app was developed for PC by adding a camera to the simulation model scene in Unity and enabling the virtual reality SDK. This meant that, when running the simulation, Unity would look for a VR headset connected to the PC and render the simulation to the headset.

Unity provides an OpenXR package which is compatible with various mixed reality devices. The Oculus VR headsets were dismissed for use in this study as they require the user to be logged into a Facebook account to use. Instead, the Valve Index was selected. This headset also boasts the highest refresh rate (120 Hz) available at the time of study (2021) as well as a high quality, 1600 x 1440 px, resolution per eye. The Valve Index is a tethered headset, meaning that it must be connected to a computer which runs the VR application. The headset also uses lighthouses as part of its tracking. The lighthouses emit infrared rays which may be interfered with by sunlight. These factors strongly limit the headset to indoor use only. There are commercial headsets which use “inside out tracking” which do not require lighthouses. Such headsets are also usually wireless or self-contained, meaning that they do not require a tether to a computer. This is much more suited to outreach applications where multiple headsets could be provided at a stall or exhibit.

For the VR app, the following hardware and software were used:

- Desktop computer with minimum specs (https://help.steampowered.com/en/faqs/view/105E-66E3-962A-1577):
    - Processor: Dual core with hyper-threading
    - Memory: 8 GB RAM
    - Graphics: Nvidia GeForce GTX 970 / AMD RX 480
    - DisplayPort 1.2 and USB (2.0+) *a HDMI to DisplayPort adapter will not work*
- Valve Index headset, lighthouses, and controllers
- 4 mains power sockets (including 1 for the computer, not including computer monitor power supplies)
- Free Unity License with an additional package installed:
    - OpenXR

The SteamVR package was used for the additional tools and features it provides, such as controller tooltips, teleporting and other VR interactions. SteamVR was not necessary for the development of the application.

VR technology allows for fully immersive experiences in environments which may generally be unsafe or out of reach for the user to physically travel to. For such an experience, the developer must provide or produce models for anything which is to be seen by the viewer, as the headset obscures their view from the outside world. For this work, models for Manchester City centre and the drones were externally sourced. Everything was rendered to scale and the user is placed in an arbitrary location within the city while the simulation runs in real-time.

4.3 App Development Process

The delivery network simulation model was developed on PC using the Unity engine. The computer was used to run the model and collect results while in the Unity Editor application.

To produce the AR application, probes were added to the simulation which recorded the position, time, and ID number for any drone agents entering or exiting the area around a point of interest. This data was used to recreate the simulation without much of the computational overhead. This was a necessary step for the augmented reality application as the computational power of a tablet is insufficient to run such a simulation.

For each point of interest in the city centre, a new Unity scene was created with a “session origin” object and camera placed in the same position that a person using the AR application would stand. The session origin defines the position of the user within the AR world. When the AR app is started, the user will be positioned at the session origin in the AR world. The orientation of the AR world may be aligned with the real world using the compass in the phone/tablet e.g. north from the compass may equate to the global Z axis in Unity. Alternatively, the scene may be set such that the global Z axis points in a notable direction, and the user is instructed to face in the same direction when starting the AR app on the device. The VR app is simple to align as the virtual environment is all the user is able to see, thus, their orientation in the real world is only a consideration for preventing the user from crashing into walls or furniture.

5 Results

5.1 VR Renders

The virtual reality application placed the user next to the Central location. As the VR headset used requires a computer to run, the agent-based simulation was also run in real time. This allowed for easy changes, such as the navigation height of the drones, to be made during testing.

The VR application was not portable and could not easily be brought to the actual viewing location due to the large computer and various cabled devices required to run the application.

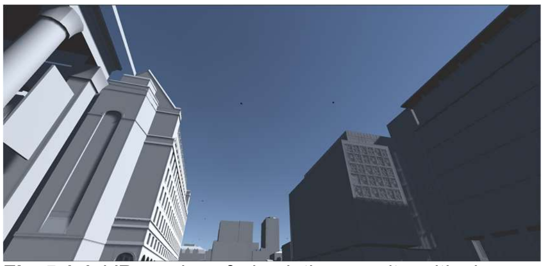

Fig 5.1.1 *VR render of simulation results with drones flying over the Central location.*

While a good quality model was sourced for this work, the VR application was dependent on having such a model. This would prove to be a limiting factor for development of VR renders for other locations as a 3D model of the location would either need to be sourced or produced.

Speaking anecdotally: without hyper-realistic graphics and textures, the VR app felt more like a video game than a real world example and failed to give a sense of what the simulation would look like in the city centre. Overall, this meant that the application was only useful in providing a sense of scale for the drones, i.e. how large they would appear in the sky for a given flight height. Future implementations may look to use 3D cameras and photogrammetry to produce realistic environments for VR applications (https://www.e-drone.org/).

5.2 AR Results

Most AR devices combine the data from an inertial measurement unit with optical flow data from the camera. Optical flow requires a feature rich environment, which is generally the case when observing an environment such as a city. However, this is not the case when pointing the camera at the sky during the night. In this situation, the optical flow data becomes noisy or unusable and the AR application fails to update the positions of the drones.

The AR software development kit in Unity provides feature detection, including the identification of planar surfaces seen by the camera. This is especially useful for placing objects on the ground and may be useful for ensuring proper occlusion of objects as they move behind buildings. Occlusion was a significant issue in this test case as drones which should be hidden behind buildings were rendered over the buildings, breaking the immersion. Occlusion could be handled in the simulation by adding a step in the process where the recorded simulation data is replayed and any drones which are occluded from the desired position of the user are hidden.

Fig. 5.2.1 *AR render in the Central location. 6 drones are visible in the image.*

Fig. 5.2.2 *Another AR render in the Central location. 3 drones are visible in the image.*

Fig. 5.2.3 *AR render in the Delivery Source location. 5 drones, and a seagull, are visible in the image.*

Fig. 5.2.4 *AR render in the Domestic location. 1 drone is visible in the image.*

6 Conclusions

Augmented reality provides a useful fusion of real and simulated information. It saves time and effort in the development of realistic models as the real world provides an ideal backdrop for visualisations. The rendered videos and images from the AR application provide a convincing depiction of the simulation results and may be useful in further studies of the societal impacts that future drone use may have. Issues with tracking while the AR device points at the sky and the immersion-breaking lack of proper occlusion are significant hindrances to the technology's application.

Virtual reality provides a much more immersive experience than augmented reality devices currently can but requires the development, or procurement, of high quality 3D models and textures of the environment. Some VR hardware is constrained to use indoors with access to power supplies which may be limiting in survey studies. This work did not investigate the portable hardware due to their lower specifications and Facebook account requirements imposed by the hardware developers.

Both virtual and augmented reality proved easy to develop applications using Unity's software development kits.