

VOXEL-BASED VISIBILITY ANALYSIS FOR SAFETY ASSESSMENT OF URBAN ENVIRONMENTS

M. Aleksandrov^{1,*}, S. Zlatanova¹, L. Kimmel¹, J. Barton¹, B. Gorte¹

¹ UNSW Built Environment, Red Centre Building, Kensington NSW 2052, Sydney, Australia
(mitko.aleksandrov, s.zlatanova, laurence.kimmel, jack.barton, b.gorte)@unsw.edu.au

Commission IV, WG IV/10

KEY WORDS: 3D Visibility Analysis, Voxels, Safety, Visualisation, Urban Environment

ABSTRACT:

This paper presents the research and development of a volumetric 3D visibility analysis application to assess public space in regards to safety and security. While most of the academic literature on space visibility concentrates on developing viewsheds from a specific viewpoint, this work integrates dynamic scenarios and real-time calculation of space visibility. Voxel-based spatial representation is utilised as a base for the 3D visibility analysis. Different space configurations are tested illustrating the robustness and usefulness of the developed application. In order to measure and evaluate different safety zones, an innovative approach is tested according to which the number of times each voxel is observed is recorded and further classified into different zones. In this way, an accurate calculation of the observed space can be performed understanding which part of an area of interest is less or more observed. The research has shown great potential for first responders, researchers, architects and safety experts. The developed application can be used as a simulation tool to assess the safety of different urban environments and identify potentially vulnerable locations.

1. INTRODUCTION

3D models provide a better perception and more elaborated options for planning, analysing and decision making. 3D models including terrain and architectural objects can be obtained from different sources such as laser scanning (Haala, Brenner, 1997), 2D GIS data (Shiode, 2000), constructing BIM models (Hardin, McCool, 2015, Diakit , Zlatanova, 2016), etc. By having detailed representation of the 3D indoor and outdoor better informed decision-making could be achieved (Tashakkori et al., 2015), and public safety can be assessed and improved (Stanchev et al., 2009). In this regard, a safety measure assessing the visibility of staff, first responders and CCTVs should be tested to understand if their lines-of-sight are not obscured by trees or other objects (Forman et al., 2009).

Intervisibility has been studied for many years in different contexts. For example, in real estate to understand if a property has an open view (Feitelson, 1992), for navigation of autonomous vehicles (Morales et al., 2014), for situational awareness of employees (Cheng, Teizer, 2012), etc. In terms of the safety, it supports the assumption that additional ‘Eyes on the street’ always reduce crime (Fuller, Moore, 2017), considering the fact that ‘some through movement and strong intervisibility of good number of entrances ... are the safest spaces’ (Hillier, Shu, 2000). One of the possible methods to assess intervisibility is using a voxel-based representation of space. A voxel represents a regular grid in three dimensions, and has been used in many research areas (Kaufman, 1990). For instance, research using voxels was performed examining the visibility of urban environments based on a specific viewpoint (Fisher-Gewirtzman et al., 2013). However, the view of people is not consistent and varies through time as people walk through an area or change their view direction. Many researchers also

used scanning technique generating point clouds to identify which parts of a room are occupied, empty or occluded (Ad n et al., 2015, Alsadik et al., 2014). The other technique considers voxelising the environment, where those voxels are used as targeting locations (Morales et al., 2014). However, in both cases, the visibility of objects (e.g., facade or terrain surface) within person’s view was determined, while the visibility of empty spaces (i.e., voids) is not determined. Therefore, in this paper, we will combine people’s actions and analyse their combined views to form a continuous visibility model.

One of the first steps to establish safety within an area of interest (AoI) is to identify potential targeting locations, which might be of interest during a terrorist attack. In this context, taking potential firearm offenders and crowded place into account is important to assess the visibility for different locations in order to understand where an attacker may have access to the most number of people (ANZCTC, 2017). Thus, open line-of-sight to spaces occupied by crowds should be minimised to maximise safety, in which case visibility plays a key role.

Before discussing the proposed approach and understanding the characteristics of it, it should be noted that modelling people’s actions is out of the scope for this paper. Rather, we provide an approach to analyse how people’s safety can be improved through 3D visibility analysis. The Unity game engine was used to prepare 3D models and implement the proposed approach.

In this paper, we propose voxel-based visibility analysis for the safety assessment of urban environments. The paper is organised as follows: Section 2 explains the approach components and characteristics. The approach is presented in Section 3. Section 4 shows different case scenarios for which the approach is tested. Section 5 was dedicated to conclusions and future work suggestions, where potential aspects that can be improved are presented.

*Corresponding author

2. APPROACH REQUIREMENTS

The main intention of developing this approach is to predict potential areas vulnerable to terrorist attacks and minimise risk, as well as to identifying space visibility coverage needed for staff, first responders or CCTVs. Locations such as sports grounds, transport hubs and entertainment venues potentially expose crowds to malicious intentions. These should be assessed ensuring effective plans and measures are put in place to mitigate any impacts (Benton et al., 2015).

To develop the approach, there are several components to consider allowing us to model the 3D visibility analysis. Three main components are identified including agent, area of interest and data structure. By modelling these components various safety-related scenarios can be considered for assessing different viewpoints of attackers or professional personnel.

2.1 Agent

Since there are several types of agents that can use the system and be assessed in terms of visibility, the approach should take into account different parameters for each agent. Thus, agents' characteristics and movement should be modelled and integrated within the approach (Aleksandrov et al., 2018). Aspects to consider are agent's field of view and actions, which are covered in the following subsections.

2.1.1 Field of View

Field of view (FoV) represents the observable area that an agent can perceive through his/her eyes. FoV allows covering an area rather than observing a single focus point. At the same time, the area observed is in 3 dimensions since agents have horizontal and vertical FoV. Representing the space through volumetric models such as voxels provides the ability to accurately measure the visible space of an agent. When it comes to horizontal FoV of a person, a total angle of 180° can be observed by two eyes, where binocular overlap of 120° is achieved (Duh et al., 2001). Regarding vertical FoV, a total angle of 120° can be achieved with eye rotation, where the constant angle in focus is around 60° . It should mention that neck rotation is also possible, but it does not impact the observable FoV. Another parameter that needs evaluation is related to the maximum distance that person or sensor can observe. In this aspect, eyes and cameras almost do not have a limitation until are blocked by an object. In terms of safety control, first responders and staff try to establish a perimeter which they can cover easily and be in charge of during an event. Figure 1 highlights the parameters considered for the visible space coverage analysis. Visibility conditions can vary or change due to weather (Hassan, Abdel-Aty, 2011), a period of the day or some works, impacting FoV and particularly the maximum distance that an agent can observe.

2.1.2 Actions

Surveillance Cameras are usually in fixed positions. They can, however, have different actions such as zooming and rotation. Other agents have higher levels of freedom and may move continuously or just observe the surrounding area. Therefore, agent's actions are equally important to find out and consider during the safety assessment (Lovreglio et al., 2016). In this regard, some of the agents' actions like rotation and movement are modelled and presented in the experiments section.

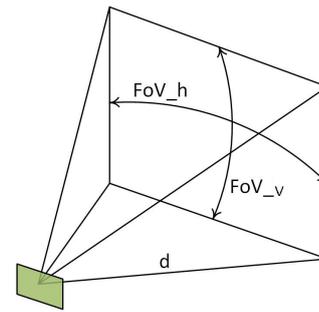


Figure 1. FoV characteristics.

2.2 Area of Interest

Another aspect that should be explained is related to the AoI definition. AoI should be able to cover not only 2D space considering the ground but also 3D space which can be occupied by people or vehicles at some point. Vertical architectural components are required since shooters use those places to have exposure to more people or a clear target. To consider these requirements AoI is represented as a rectangular box, fitting regular size voxels within it.

In order to determine which voxels are visible, the raycast method is used (Figure 2). Raycasting is a commonly used method to determine the line-of-sight of an agent in computer games (Möller, Trumbore, 2005). The term was first used to present an approach for rendering constructive solid geometry models (Roth, 1982). For each voxel, a number of views is recorded capturing the time component.

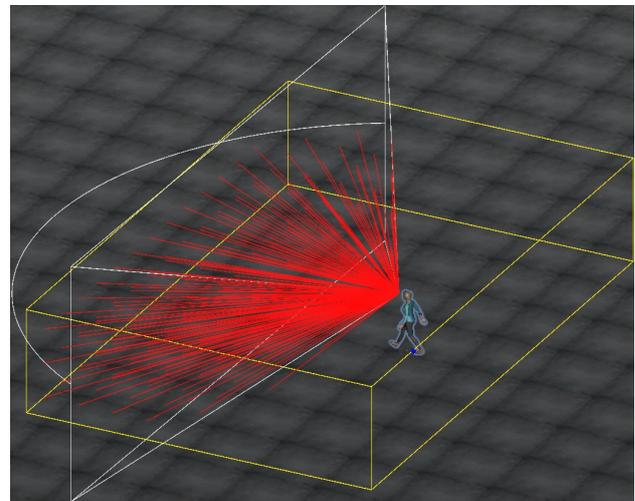


Figure 2. Visible voxels within AoI. Rays targeting visible voxels presented in red, while the area of interest is drawn in yellow.

2.3 Data Structure

Modelling 3D environments accurately is a very important aspect of safety. For hostile vehicle mitigation terrain should be identified precisely, capturing all terrain features (Forman et al., 2009). Regarding 3D visibility analysis, all environmental features should be present within the scene to realistically convey the visible space.

As mentioned previously, Unity software is used for the front-end user interface. It mainly works with meshed objects,

allowing the direct import of obj or fbx files, and the ability to draw objects manually or through code. The software also provides the option to model or import terrain and to include different types of trees and vegetation in the scene.

On the other hand, to represent empty spaces geometrically voxels represent the best option (Gorte et al., 2019). This requires that objects are voxelised either with unified or uneven voxel size. The main reason for this is to generalise the space optimally so the main characteristics of the environment are retained. With such representation, some of the computational operations can be reduced and the result is still accurate enough for our purpose (Gorte et al., 2018). This approach is also very useful for modelling objects with irregular shapes such as trees. Tree voxelisation can ensure realistic looking trees that are computationally efficient. This helps in assessing visible spaces as representations of the real-world.

For visibility analysis in Unity, all objects within the scene should be classified as 'obstacles' in order to be considered as a natural blockage of a person's view. At the same time, objects should have mesh 'colliders' matching precisely their structure to let the rays to detect collision.

3. IMPLEMENTATION

As mentioned previously, the Unity game engine was used to implement the approach where all whole process is divided into 3 steps covering agent and AoI modelling as well as voxel visualisation.

3.1 Agent Modelling

Since Unity software supports object-oriented programming, each object can have its own properties. For the agents based on the characteristics presented in Section 2.1.1, vertical and horizontal FoV along with the maximum distance could be defined. At the same time, some of the prebuilt functions in Unity are used to model more realistic scenarios. Thus, the implemented navigation system was used in identifying the navigable area for moving agents. This considered the agent's radius, height, maximum climbing slope and step height (Kyaw, 2013). This shows the accessibility of an agent as another safety measure, especially for vehicles (Figure 3). It should be highlighted that this method also uses voxels in the background to determine the navigable area. The shortest path between an agent's position and a destination could also be determined and used for the navigation of the agent.

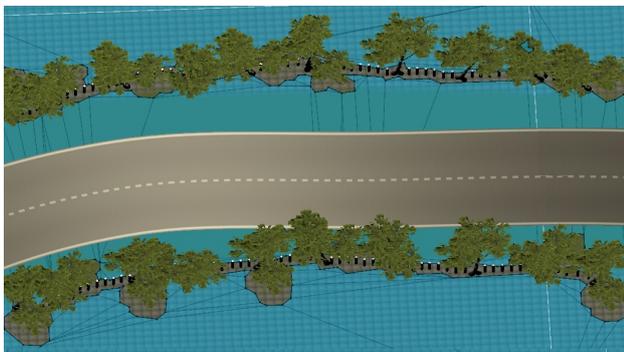


Figure 3. Accessible area for a car presented in blue without considering the road.

3.2 AoI Modelling

Regarding the AoI, the user was enabled to define different rectangles and place as many as required arbitrarily in the scene. The maximum number of unified size voxels fit to cover these defined areas. The whole process of identifying which voxels are visible within an AoI can be followed in the provided pseudocode in Algorithm 1:

Algorithm 1: The process of identifying and updating visible voxels.

Input: Agent's properties such as transformation (i.e., position and orientation), angles for horizontal and vertical FoV and perimeter; 3 dimensions representing the AoI

Output: Updated visible voxels

```

mbbAgent = MBB(agent)
mbbAoI = MBB(AoI)
mbb = Intersection(mbbAgent, mbbAoI)
for each x in [min(mbb[0]), max(mbb[0])] do
    for each y in [min(mbb[1]), max(mbb[1])] do
        for each z in [min(mbb[2]), max(mbb[2])] do
            voxel = AoI.voxels[x, y, z]
            vector = voxel.pos - agent.pos
            dir = Normalise(vector)
            aglFwd = Angle(agent.forward, dir)
            if angle <= FoVh then
                aglUp = Abs(Angle(agent.up, dir) - 90)
                if aglUp <= FoVv then
                    dst = Distance(agent.pos, voxel.pos)
                    if dst <= perimeter then
                        if !Raycast(voxel.pos) then
                            AoI.voxels[x, y, z].value ++;
    
```

The functions used in the algorithm are the following:

MBB(agent): find the Minimum Bounding Box (*mbb*) considering *agent*'s position, orientation and FoV. At the same time, vertices of the object are normalised to keep the orientation consistent.

Intersection(mbbAgent, mbbAoI): make a set of ordered 3D points representing the intersection of agent's mbb and mbb of considered AoI.

Normalise(vector): normalise a vector to a length of 1.

Angle(vector1, vector2): find an angle between two vectors.

Abs(angle): find an absolute value of an angle.

Distance(agent, voxel): identify a distance between the agent and a voxel.

Raycast(voxel): find if a ray hits an object with a collider on its way, and understands consequently whether the targeting voxel is visible by the agent.

The presented algorithm first enables finding roughly for each agent the intersection with which AoIs based on its mbb. After that, the algorithm determines more precisely which voxels should be within the observing FoV and distance of agent. As a final step, a ray is sent towards the middle of the voxel determining its access and visibility, based on which the voxels are updated as part of the AoI structure.

3.3 Voxel Visualisation

After running a scenario for some time and measuring the visibility of each voxel, the voxels are drawn within the AoI. One of the options is to draw each voxel as an individual cube, however, this approach is computationally expensive. The other approach is to group them into one object. To unify each observed voxel into one object all the voxels' faces that are inside the object are omitted to speed up the drawing process and reduce the number of faces to be rendered. In Unity the faces of a cube are commonly known as quads. Before constructing the voxel-based object, overlapping quads should be eliminated and the normals should point towards the outside (Figure 4).

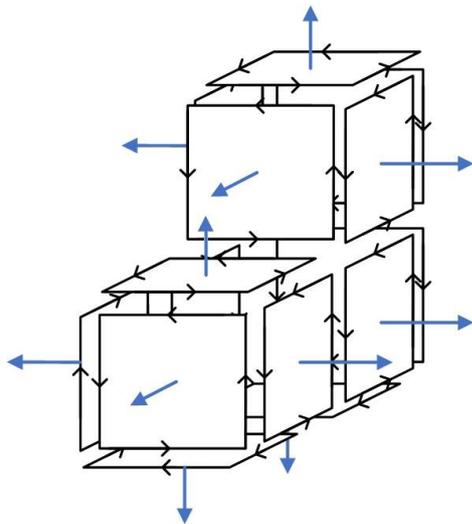


Figure 4. Voxel model construction.

Drawing the voxel-based object as one mesh can result in problems if the area of interest is too large. For example, if the voxel model contains more than 65,000 vertices one mesh cannot be used. Therefore, the whole voxel model was divided into several smaller chunks to tackle this problem. In this way, each chunk was drawn as a separate mesh. This approach is also useful when the voxel model is updated, in which case only those parts that require updating are redrawn as part of the model.

When it comes to the mesh object voxelisation, the code from the unity-voxel project available on github platform was utilised (Mattatz, 2018). In this way, objects that are with high details could be voxelised, such as trees in some of the scenarios (Figure 5).

4. EXPERIMENTS AND DEMONSTRATIONS

To demonstrate the whole process, several examples were tested examining the usefulness of each approach along with its capabilities.

Considering the information in the paper, default values have been assigned for the FoV and maximum distance that people can observe. For the horizontal and vertical FoV values of 120° and 60° have been assigned, respectively. For the distance, a value of 100 meters was considered, establishing an observing perimeter for an individual. In terms of the AoI, the size of the voxels was 1 meter, where the updating interval to observe the

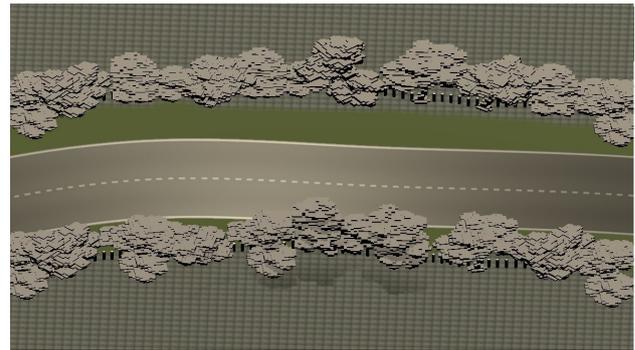


Figure 5. Voxelised trees.

voxels was set up to 1 second. The voxel size considered for objects voxelisation was 10 cm.

In the following text, we explore individuals and vehicles as agents, different object shapes for the AoI and grouping voxels with the similar characteristics by colour and height. All the 3D geometrical models are either imported from Unity Asset Store or generated by Unity default tools.

4.1 Terrain

The impact of terrain was presented in the first scenario to show how the system works. This scenario considers one AoI placed on the terrain blocking the line-of-sight of the agent. The visible voxels were calculated and built after the agent travels to the destination (Figure 6). From the scene, it can be inferred that the terrain can reduce a person's view. This can be of interest for the prevention of hostile vehicle encroachment since drivers will not have a clear view into the occluded scene.

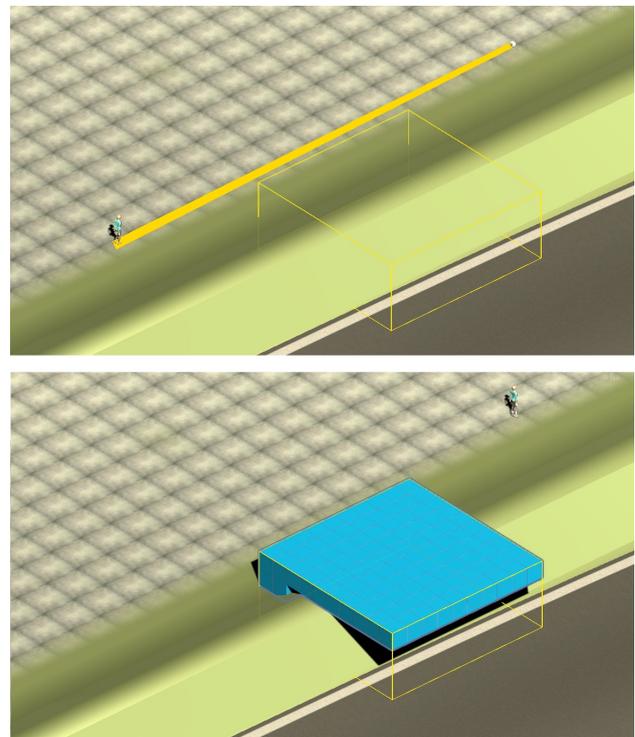


Figure 6. Pathfinding (top image) and impact of terrain on person's visibility (bottom image).

4.2 Buildings and Multiple AoI

As previously mentioned, the user can define multiple AoIs and examine particular aspects within these. In this regard, two AoIs were defined covering the exterior of two buildings. Also, a destination for the agent was defined and covered space was presented (Figure 7). Since the agent was closer to the first building on the left at the beginning of the scenario, fewer voxels were observed compared to the second building presented on the right. This scenario is of particular interest for first responders and staff to cover the space vertically (in the Z direction) in order to protect, for instance, a crowded place or a specific person from shooters that can be located in one of the buildings.

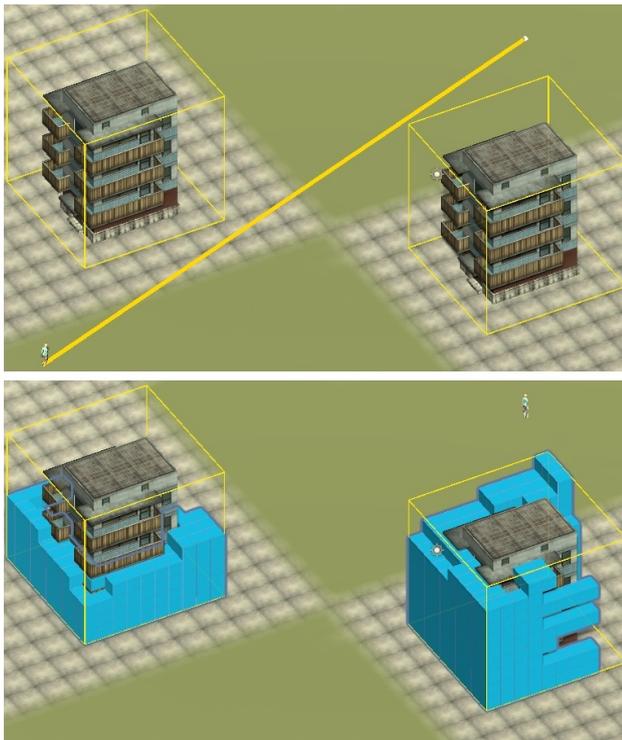


Figure 7. Understanding intervisibility for exterior of buildings.

4.3 Visibility Analysis for Multiple Agents

Another possible scenario is to have several agents covering the same space (Chen et al., 2013). To provide an example an AoI was specified with the size of $100 \times 100 \times 4$ meters along with three agents traveling towards the middle area of the AoI. In order to show which voxels are more observed, the number of views for each voxel was normalised and divided into five groups considering a range of 0.2. As expected, voxels in the middle are observed the most, presented in red, while other voxels distributed towards the edges of the AoI were checked fewer times (Figure 8 - top image).

Since the voxel visualisation method unifies all voxels with the same characteristics, they need to be presented as separate objects (Figure 8 - bottom image). In this way, the user can better realise which space is better covered. The volume of each group can be measured to understand quantitatively the ratios between groups and to reveal equally distributed coverage by the agents.

Table 1 shows the number of voxels for each object group considering the same scenario. Based on the results, there are

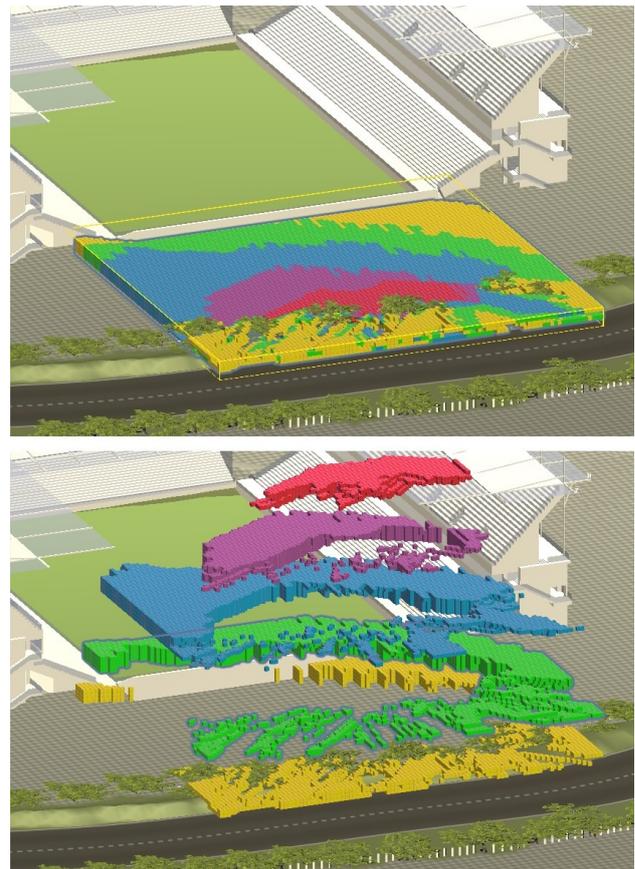


Figure 8. Visibility analysis considering several agents. Top image - observed voxels represented through a heat map; bottom image - extended heat map in Z direction.

more voxels covered fewer times (i.e., yellow + green voxels) than voxels that are observed many times (i.e., purple + red voxels), making the observing process by the agents unevenly distributed within the AoI.

Red	Purple	Blue	Green	Yellow
4014	5332	11288	10344	6014

Table 1. Number of voxels in each object group

4.4 Visibility Analysis for Vehicles

One of the safety measures that can be conducted for vehicles is assessing driver or passenger visibility (Dewar et al., 2007). For this scenario, the vehicle itself should be considered as a moving obstacle, since it can block the view to people who are either inside or outside the car. The same should be considered for crowded places, since people can obscure easily the line-of-sight of staff and first responders. Another aspect that needs evaluation is to examine whether the vehicle is parked or it is moving. Also, speed and vehicle trajectory can have huge impact on results.

Figure 9 presents the 3D visibility analysis of a driver. The defined size of the AoI was $120 \times 100 \times 4$ meters, where the speed and acceleration of the vehicle were set up at 30 km/h (4 m/s^2), respectively. Along the trajectory, more voxels were observed as opposed to the area behind the trees. The considered speed and updating interval of 1 second created a

regular pattern as a gap between the voxel groups, which could be even larger if the speed of vehicle is higher and vice versa. The voxelised trees did not obscure the driver's view since they are above the eye-level of the driver, otherwise the treetops could make a more significant impact. In examining these aspects, the drivers visibility could be identified along with some vulnerable spots.

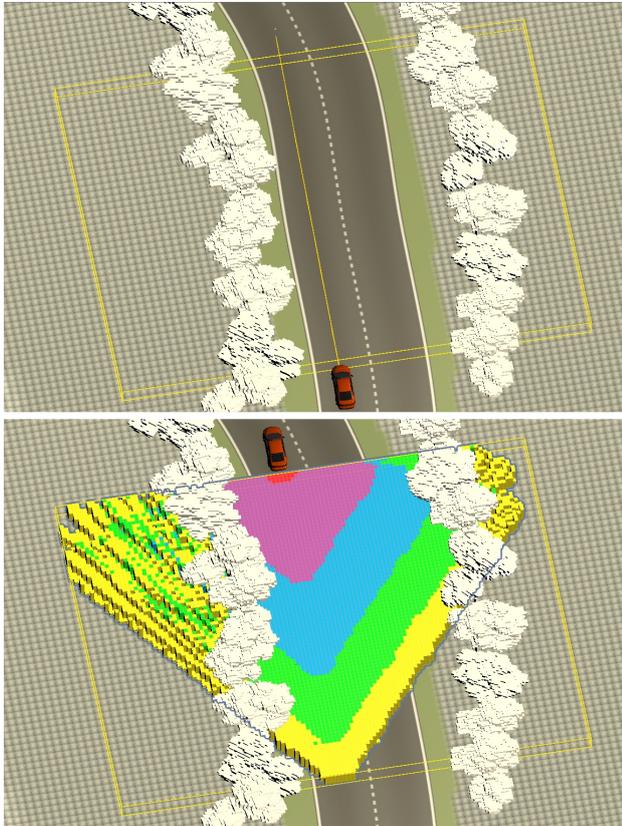


Figure 9. Visibility analysis of a driver. Top image - the vehicle trajectory and AoI; bottom image - the 3D visibility analysis.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented an approach for 3D visibility analysis and illustrated the benefits of using it for safety assessment within urban environments. Unity software was used to develop the approach and test different scenarios. The approach integrates several components including agent-based modelling, determination of visible voxels and voxel visualisation. The considered scenarios suggest that 3D visibility analysis can be used by staff, first responders and other agents in a variety of situations to improve the safety of people. Voxel classification and visualisation based on similar characteristics have been done understanding in which way agents observed an AoI.

The performance of the implementation was not tested since all the scenarios were executed in real-time, without any delays. However, to model more complex and demanding scenarios, a detailed evaluation is required and optimisation of the number of raycast calls allowing calculation of the visible voxels for many people. The weather impact was not considered in the paper either. The effect of the weather can be significant as it can limit an agent's visibility to a good extent.

In future work, there are several aspects that will be investigated. For example, we had visualised via different colours how people observe a space through time. Important considerations would include real-time visualisation of the observed area. Since Unity game engine supports parallel processing using all cores, it would be of interest to understand the improvement level which can be achieved with it. Also, asynchronous calculation of visible space within the updating interval can enable integrating more agents in the scene and should be investigated.

More research is needed in the area of agent-based modelling, investigating and integrating more realistic actions for the agents and predicting some of the safety-related aspects more accurately. The optimisation of agents distribution considering agents actions and the 3D visibility analysis would definitely improve the safety of people and should be investigated in the future.

ACKNOWLEDGEMENTS

Acknowledgements of supporting this project goes to Landcom and State Government Contract. The project is about identifying strategies for planning safe and secure public domains (InfoEd Ref: RG182869).

REFERENCES

- Adán, Antonio, Quintana, Blanca, Vázquez, Andres, Olivares, Alberto, Parra, Eduardo, Prieto, Samuel, 2015. Towards the automatic scanning of indoors with robots. *Sensors*, 15, 11551–11574.
- Aleksandrov, Mitko, Rajabifard, Abbas, Kalantari, Mohsen, Lovreglio, Ruggiero, González, Vicente A, 2018. People choice modelling for evacuation of tall buildings. *Fire technology*, 54, 1171–1193.
- Alsadik, B, Gerke, M, Vosselman, G, 2014. Visibility analysis of point cloud in close range photogrammetry. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2, 9.
- ANZCTC, 2017. Active armed offender guidelines for crowded places. <https://www.nationalsecurity.gov.au/Media-and-publications/Publications/Documents/active-armed-offender-guidelines-crowded-places.pdf>.
- Benton, Nigel et al., 2015. Not if, but when. *Australasian Leisure Management*, 50.
- Chen, Huan-Ting, Wu, Si-Wei, Hsieh, Shang-Hsien, 2013. Visualization of CCTV coverage in public building space using BIM technology. *Visualization in Engineering*, 1, 5.
- Cheng, Tao, Teizer, Jochen, 2012. Modeling tower crane operator visibility to minimize the risk of limited situational awareness. *Journal of Computing in Civil Engineering*, 28, 04014004.
- Dewar, Robert E, Olson, Paul L, Gerson, JA, 2007. Human factors in traffic safety.

- Diakit , Abdoulaye Abou, Zlatanova, Sisi, 2016. Valid space description in BIM for 3D indoor navigation. *International Journal of 3-D Information Modeling (IJ3DIM)*, 5, 1–17.
- Duh, HB-L, Lin, JW, Kenyon, Robert V, Parker, Donald E, Furness, Thomas A, 2001. Effects of field of view on balance in an immersive environment. *Proceedings IEEE Virtual Reality 2001*, IEEE, 235–240.
- Feitelson, Eran, 1992. Consumer preferences and willingness-to-pay for water-related residences in non-urban settings: A vignette analysis. *Regional studies*, 26, 49–68.
- Fisher-Gewirtzman, Dafna, Shashkov, Alona, Doytsher, Yerach, 2013. Voxel based volumetric visibility analysis of urban environments. *Survey Review*, 45, 451–461.
- Forman, Paul, Evans, Dorian, Heward, Gary, 2009. Vehicle-borne threats and the principles of hostile vehicle mitigation. *Blast effects on buildings*, Thomas Telford Publishing, 250–273.
- Fuller, Martin, Moore, Ryan, 2017. *The death and life of great American cities*. Macat Library.
- Gorte, B., Zlatanova, S., Fadli, F., 2019. Navigation in indoor voxel models. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5, 279–283.
- Gorte, BGH, Zhou, K, Van Der Sande, CJ, Valk, C, 2018. A computationally cheap trick to determine shadow in a voxel model. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4.
- Haala, Norbert, Brenner, Claus, 1997. Generation of 3d city models from airborne laser scanning data. *Proceedings EARSEL Workshop on LIDAR remote sensing on land and sea, Tallinn/Estonia*.
- Hardin, Brad, McCool, Dave, 2015. *BIM and construction management: proven tools, methods, and workflows*. John Wiley & Sons.
- Hassan, Hany M, Abdel-Aty, Mohamed A, 2011. Analysis of drivers' behavior under reduced visibility conditions using a structural equation modeling approach. *Transportation research part F: traffic psychology and behaviour*, 14, 614–625.
- Hillier, Bill, Shu, SC, 2000. Crime and urban layout: the need for evidence. *Secure foundations: Key issues in crime prevention, crime reduction and community safety*, 224.
- Kaufman, Arie, 1990. Volume visualization. *The visual computer*, 6, 1–1.
- Kyaw, Aung Sithu, 2013. *Unity 4. x Game AI programming*. Packt Publishing Ltd.
- Lovreglio, Ruggiero, Ronchi, Enrico, Nilsson, Daniel, 2016. An Evacuation Decision Model based on perceived risk, social influence and behavioural uncertainty. *Simulation Modelling Practice and Theory*, 66, 226–242.
- Mattatz, 2018. Voxelize mesh algorithm in unity. <https://github.com/mattatz/unity-voxel>.
- M ller, Tomas, Trumbore, Ben, 2005. Fast, minimum storage ray/triangle intersection. *ACM SIGGRAPH 2005 Courses*, ACM, 7.
- Morales, Yoichi, Even, Jani, Kallakuri, Nagasrikanth, Ikeda, Tetsushi, Shinozawa, Kazuhiko, Kondo, Tadahisa, Hagita, Norihiro, 2014. Visibility analysis for autonomous vehicle comfortable navigation. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2197–2202.
- Roth, Scott D, 1982. Ray casting for modeling solids. *Computer graphics and image processing*, 18, 109–144.
- Shiode, Narushige, 2000. 3D urban models: Recent developments in the digital modelling of urban environments in three-dimensions. *GeoJournal*, 52, 263–269.
- Stanchev, Hristo, Palazov, Atanas, Stancheva, Margarita, 2009. 3D GIS Model for flood risk assessment of Varna Bay due to extreme sea level rise. *Journal of Coastal Research*, 1597–1601.
- Tashakkori, Hosna, Rajabifard, Abbas, Kalantari, Mohsen, Aleksandrov, Mitko, 2015. Indoor incident situation awareness using a 3d indoor/outdoor spatial city model. *2015 2nd International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, IEEE, 240–245.