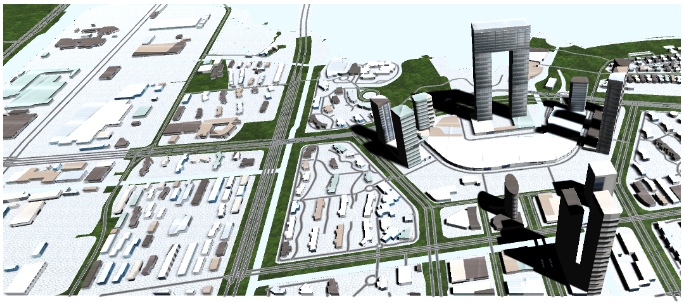
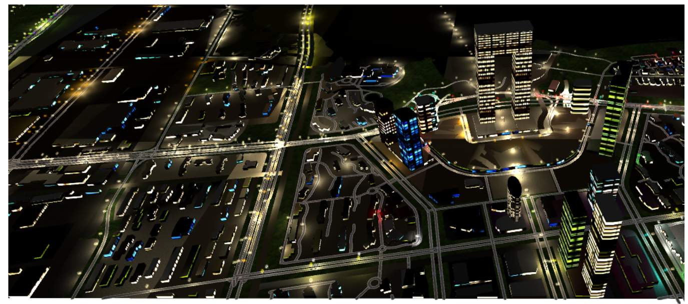
Synthesizing Virtual Night Scene on OpenStreetMap

Wanwan Li
University of South Florida
Tampa, Florida, USA
wanwan@usf.edu



(a) Synthesized Urban Scene on OpenStreetMap (Daytime)



(b) Synthesized Urban Scene on OpenStreetMap (Night)

Fig. 1. This figure shows the experiment results from our proposed procedural modeling approach that automatically synthesizes realistic virtual night scenes (b) from OpenStreetMap data (a). The location presented in this OpenStreetMap is Suzhou, China.

***Abstract*—As modern game development technologies are emerging rapidly, virtual content synthesis for immersive gaming experiences became a popular research topic recently. Especially, as the game genres driven by modern life experiences within realistic virtual urban environments are gaining popularity, efficient and effective modeling approaches for synthesizing realistic urban environments are necessary. As a procedural modeling approach, synthesizing realistic urban scenes using OpenStreetMap (OSM) can result in satisfying results without demanding heavy manual efforts from game content designers. However, existing works that synthesize realistic virtual urban environments using OSM have not proposed the technical approach and experiment results that present realistic night scenes of virtual urban environments. Therefore, in this paper, we propose a novel procedural modeling approach to automatically synthesize realistic virtual night scenes on OpenStreetMap. The experiment results of synthesized urban night scenes validate the efficacy of our proposed technical approach.**

***Keywords*—Procedural Modeling, OpenStreetMap (OSM)**

I. Introduction

With the increasing popularity of virtual content synthesis for immersive gaming experiences, OpenStreetMap (OSM) [1] has become a go-to platform for virtual content generations which are applied to the game genres driven by modern life experiences within realistic virtual urban environments. OSM is a collaborative project that aims to create a free, editable map of the world [2]. It is built by a community of mappers who contribute and maintain data about roads, buildings, and other features [3]. With the advent of the latest technology in computer procedural modeling, OSM has evolved to include the synthesis of virtual urban environments.

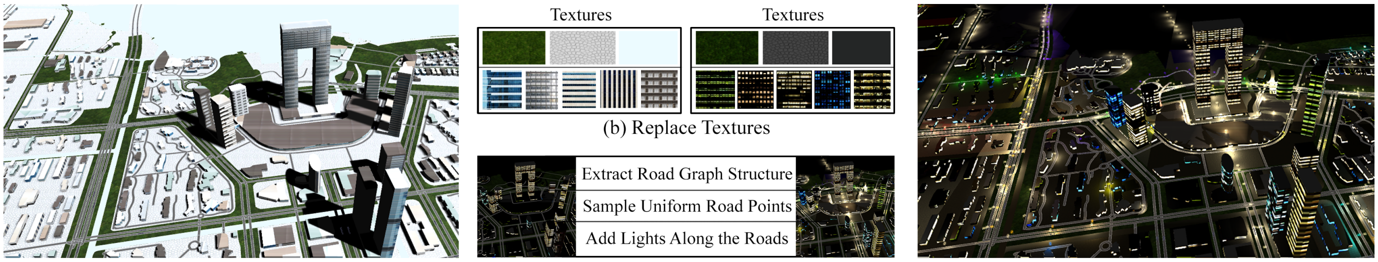
OSM's open-source nature and its accurate reconstruction from real-world data [4] make it a reliable source of information for navigation, tracking, and visualization. Meanwhile, as one effective procedural modeling approach, synthesizing realistic urban scenes using OSM can result in satisfying results without demanding heavy manual efforts from game content designers. However, one aspect that has been lacking in existing works that synthesize realistic virtual urban environments using OSM is the technical approach to synthesizing virtual urban night scenes with OSM. There are several benefits to synthesizing virtual night scenes using OSM data.

First of all, synthesizing virtual night scenes on OSM is capable of simulating and testing lighting scenarios for the game and movie design process. By creating a virtual environment of the urban night scene, designers can experiment with different lighting schemes in gameplay or movie design, identify potential problems or imperfections, and develop solutions before implementing those design plans. This procedural modeling approach can save time and resources while also ensuring that the lighting is optimal and meets the needs of the designers.

Secondly, procedural synthesis of virtual night scenes on OSM can enhance the users' gameplay experience. Virtual urban night scenes can provide users with more realistic, immersive, and stimulus visual effects that make it a more attractive feature for a game to navigate and explore the virtual world at night so as to improve the level of enjoyment.

Thirdly, OSM night scenes match the real-world environments. This makes it possible to train a user to adapt to a new environment at night. For example, after navigating in the synthesized virtual world at night, a user can easily identify landmarks such as buildings and monuments, which are highlighted with streetlights, and use them as reference points for navigation. This can be particularly useful for individuals who are unfamiliar with a city or area.

Lastly, synthesizing virtual night scenes on OSM can also



(a) Virtual Urban Scene on OSM (b) Replace Textures (c) Add Road Lights (d) Virtual Night Scene on OSM

Fig. 2. Overview of our approach.

be beneficial for businesses that rely on location-based services in virtual reality. For example, a restaurant can use a virtual urban night scene to showcase its location using a VR display, making it easier for potential customers to find it. Similarly, a hotel can use a virtual urban night scene to highlight its surroundings and provide guests with a better understanding of the area through a VR headset such as Oculus Quest.

However, existing works that synthesize realistic virtual urban environments using OSM have not proposed the technical approach and experiment results that present realistic night scenes of virtual urban environments. Therefore, in this paper, we propose a novel procedural modeling approach to automatically synthesize realistic virtual night scenes on OpenStreetMap. Contributions of our work include:

- We propose a novel problem statement for synthesizing virtual urban environments using OpenStreetMap data.
- We propose a novel approach to automatically synthesize realistic virtual night scenes on OpenStreetMap.
- We validate the efficacy of our proposed technical approach by demonstrating the experiment results of virtual urban night scenes synthesized on OpenStreetMap.

II. Related Work

Recent studies demonstrate an increasing interest in OpenStreetMap studies and applying OSM data in procedural content generation. In 2011, Luxen et al. [5] proposed a valuable solution for real-time routing with OpenStreetMap data which can identify the most efficient path between two points, taking into account factors such as traffic, road closures, and other facts. In 2014, Neis et al. [6] investigated the recent developments and future trends in volunteered geographic information research through the case of OpenStreetMap. In 2020, Vargas et al. [7] analyzes the challenges and opportunities in machine learning and remote sensing via OpenStreetMap. In 2021, Li et al. [8] proposed a novel technical approach for the automatic optimization of Uber schedules using OpenStreetMap data. By using OSM data, the Uber app's schedule can be optimized in a realistic virtual urban environment to balance the driver's average workload while minimizing the customer's average waiting time. In 2022, Li et al. [9] proposed a technical approach for simulating virtual construction scenes on OpenStreetMap for visualizing realistic virtual construction environments automatically. By using OSM data to procedurally create a geometry model of construction sites, this approach has numerous potential applications in the construction industry for generating realistic 3D models of the construction site, identifying potential safety issues, and improving communication and collaboration among construction professionals. During the same year, Li et al. [10] proposed PM4VR, a scriptable parametric modeling interface for conceptual architecture design in VR on OpenStreetMap. PM4VR is a programming interface for interactive parametric modeling on an OSM data-generated virtual environment to realize an effective and immersive conceptual architecture design experience. By using OSM data and VR technology proposed by this work, architecture designers can create parametric models, automate repetitive tasks, and interact with the conceptual designs in a more immersive way. This approach has numerous potential benefits for the architecture industry such as increasing design efficiency, improving collaboration and communication among architects, and achieving better design outcomes. In 2023, Li et al. [11] proposed a novel optimization approach for location-aware adaptation of AR narratives using OpenStreetMap data that creates engaging AR experiences in outdoor environments which match the game story. By using OSM data, optimized AR deployment can specify location according to the user's narratives, increasing engagement and realism of the storytelling.

III. Overview

In our proposed work, synthesizing virtual night scenes on OpenStreetMap involves a combination of Unity3D programming, graph algorithm, and geometric computations. First, the data from OpenStreetMap is processed in Unity3D through OSM Google Map API to extract urban environment features such as buildings, roads, land, and water. Next, a 3D model of the scene is automatically generated through an open-source Unity 3D asset, GO Map [12], which is then used to synthesize a realistic night scene. This automatic scene construction process involves the use of geometric modeling techniques such as extruding, which can represent the realistic geometry of the urban environment. The key contribution of our work is considering some other features specially applied for urban night scene synthesis. For example, those features such as streetlights, shadows, ground reflections, building reflections, etc. are added so as to create a realistic representation of an urban scene at night. Especially, the use of building textures at night in our approach further enhances the realism of the synthesized virtual urban night environment.

Figure 2 shows the overview of our approach. Give the OSM Google Map API supported on Unity3D Editor. the virtual urban scene is synthesized on OSM as shown in (a). In order to simulate the behavior of light and shadows to create a photorealistic image of the night scene, we first replace the textures for grass, land, water, and buildings from daytime to night as shown in (b). Then, another three steps are included for adding road lights as shown in (c). These three main steps include extracting the road graph structure, sampling points uniformly on the edges of the road graph, and adding streetlights uniformly along each edge of the road graph according to those sampled points. After applying all these steps, the final realistic and immersive virtual urban night scene is automatically synthesized on OSM as shown in (d).

IV. Technical Approach

In order to generate immersive night scenes on OSM, first, we attach realistic textures to the scene geometry generated on the OSM. Those textures include ground, land, roads, water, and buildings. In this work, we manually create more than ten different types of realistic building textures and put them into a texture pool. Then, during the scene procedural generation process, each build is attached with one texture randomly picked from the texture pool. Such texture pools are designed for two different versions: one for the daytime and another for the night scene. Therefore, the daytime urban scene is generated with the daytime texture pool while the night urban scene is generated with the night texture pool.

Road Graph. In order to add the streetlights procedurally on the synthesized OSM urban night scene, a data structure called road graph is employed in our technical approach. As shown in Figure 3, a road graph is denoted as $G = (V, E)$ where vertices $V = \{v_i | i = 1, 2, ..., |V|\}$ represent the road crossings and edges $E = \{(v_i, v_j) \in V^2 | i, j \in [1, |V|] \wedge i \neq j\}$ represent the roads. Road graph $G = (V, E)$ is the fundamental data structure upon which streetlights are procedurally added. Two steps are applied to construct road graphs automatically. These two steps are road merge and road split respectively.

Road Merge. Downloaded raw road data by default are segmented irregularly and can not form a graph structure where the vertices represent the road crosses and edges represent the roads. The reason for this is the raw road data contains lots of mismatches on the road crosses. Therefore, we first merge the roads wherever there are any two road segments sharing the same ending points close to each other and are contingent on each other with similar tangent directions. This operation is called road merge. Mathematically, downloaded raw road data $R^0 = \{\mathbf{r}_i^0(t) | i = 1, 2, ..., M_0 \wedge t \in [0, 1]\}$ where $\mathbf{r}_i^0(t)$ are the 3D parametric curve representing the i^{th} road. Then, let the merged roads at iteration k is: $R^k = \{\mathbf{r}_i^k(t) | i = 1, 2, ..., M_k\}$, then, the merged roads at iteration k+1 is: $R^{k+1} = \{\mathbf{r}_i^{k+1}(t) | i = 1, 2, ..., M_{k+1}\}$ where $\mathbf{r}_i^{k+1}(t) = \mathbf{r}_i^k(t) \circ \mathbf{r}_j^k(t)\ \forall j \in [i+1, M_k]$ if $\mathbf{r}_i^k(1) \approx \mathbf{r}_j^k(0) \wedge \dot{\mathbf{r}}_i^k(1) \approx \dot{\mathbf{r}}_j^k(0)$; Or, $\mathbf{r}_i^{k+1}(t) = \mathbf{r}_j^k(t) \circ \mathbf{r}_i^k(t)\ \forall j \in [i+1, M_k]$ if $\mathbf{r}_j^k(1) \approx \mathbf{r}_i^k(0) \wedge \dot{\mathbf{r}}_j^k(1) \approx \dot{\mathbf{r}}_i^k(0)$; Otherwise, $\mathbf{r}_i^{k+1}(t) = \mathbf{r}_i^k(t)$.

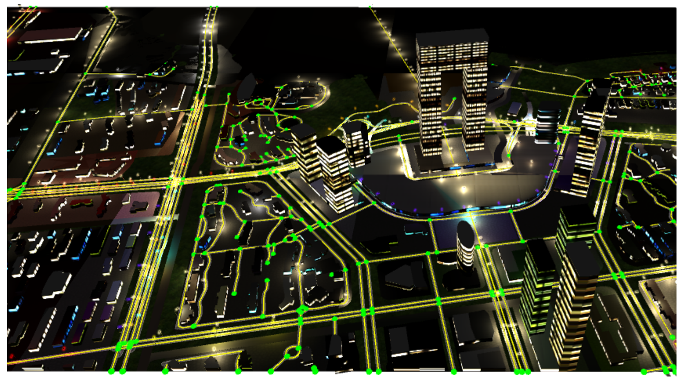


Fig. 3. Road Graph Generation. This figure shows the result of the road graph generated with our approach. In this road graph, vertices (marked as green dots) represent the road crossings and edges (marked as yellow curves) represent the roads.

Road Split. After the road merge step, we split the merged road segments $R = \{\mathbf{r}_i(t) | i = 1, 2, ..., M\}$ according to intersections and T-junctures detection operator $\otimes$. The condition to meet with a T-juncture is if there is one ending point of a road segment is lying on another road segment. After this splitting process, the road graph $G = (V, E)$ is constructed. Mathematically, road graph $G = (V, E)$ is calculated as:

$$V = \bigcup_{i=1}^{M-1} \bigcup_{j=i+1}^{M} \mathbf{r}_i(t) \otimes \mathbf{r}_j(t) \quad E = \bigcup_{i=1}^{|V|} \bigcup_{j=1}^{|V|} \{\mathbf{e}_{i,j}(t) \mid \delta_{i,j} = 1\}$$

where $\delta(i, j) = 1$ if $\exists \mathbf{e}_{i,j}(t) \in R \Rightarrow \mathbf{v}_i, \mathbf{v}_j \in \mathbf{e}_{i,j}(t) \wedge i \neq j$.

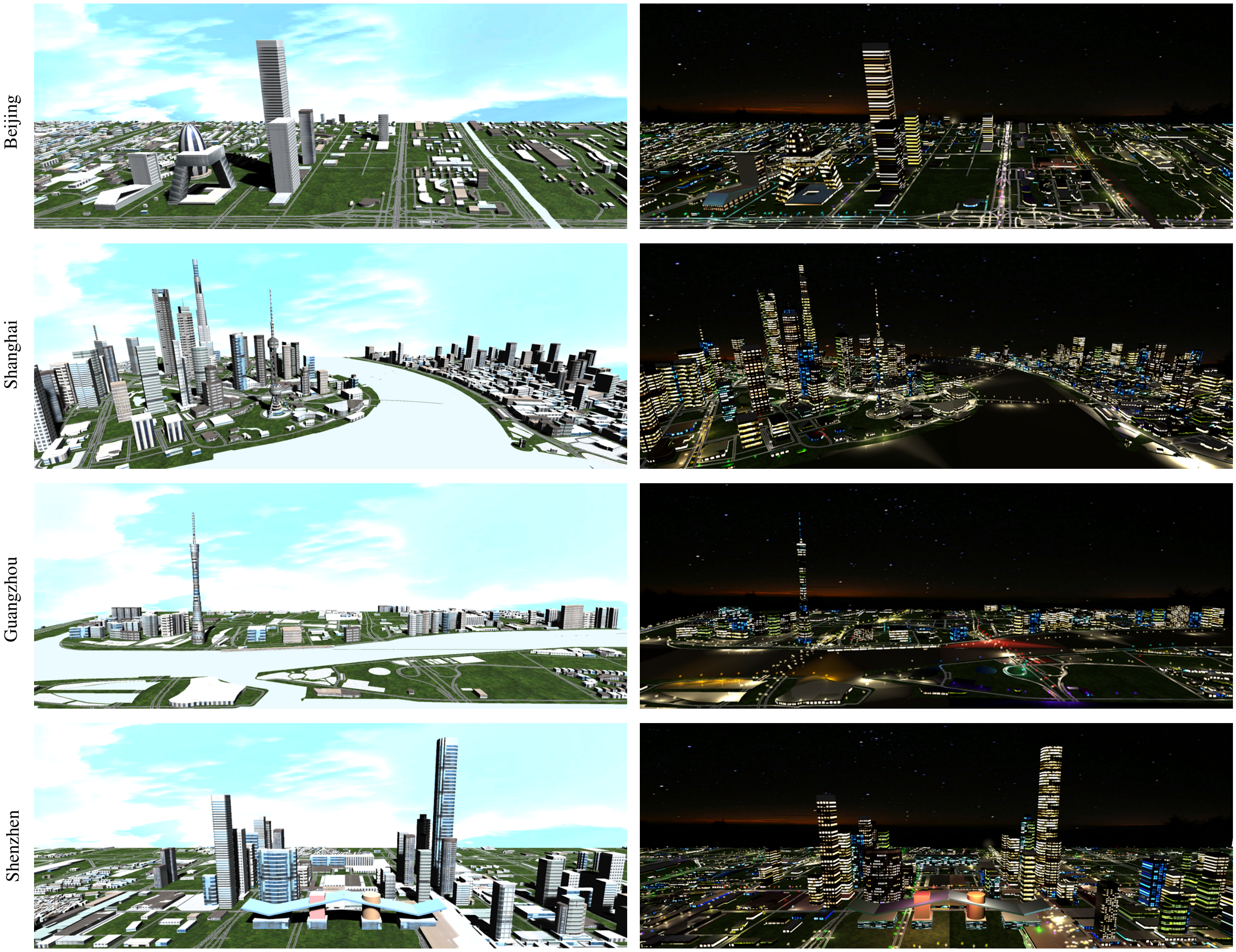
Road Lights. After the road split step, the road graph $G = (V, E)$ is constructed. Then, road lights are procedurally added onto each edge in the road graph. Then road lights $L = \{(\mathbf{p}_i, \mathbf{f}_i) | i = 1, 2, ...\}$ is a list of tuples containing the position $\mathbf{p}_i$ and the orientation $\mathbf{f}_i$ for each road light. Then, mathematically, road lights L is calculated as:

$$L = \bigcup_{i=1}^{|V|} \bigcup_{j=1}^{|V|} \bigcup_{k=2}^{N-1} \left\{ \left(\mathbf{e}_{i,j} \left(\frac{k}{N} \right), -\dot{\mathbf{e}}_{i,j} \left(\frac{k}{N} \right) \times \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \right) \right\}$$

where $\mathbf{e}_{i,j}(t) \in E$, $N = \left\lceil \int_{\mathbf{e}_{i,j}(t)} \mathrm{d}t / \zeta \right\rceil$, and light interval ζ.

V. Experiment Result

In order to validate the efficacy of our proposed technical approach, a group of experiments is conducted on synthesizing urban night scenes on OpenStreetMap. Figure 4 shows the experiment results of synthesizing virtual urban night scenes at four different locations in China, namely Beijing, Shanghai, Guangzhou, and Shenzhen. The figure is divided into four rows, with each row corresponding to a specific location setting. The first row shows the area near the CCTV headquarters in Beijing, China. The second row shows the area near the Oriental Pearl Tower in Shanghai, China. The third row shows the area near the Canton Tower in Guangzhou, China. The last row shows the area near the Ping An International Finance Centre in Shenzhen, China. Each row of the figure



(a) Synthesized Virtual Urban Scenes.

(b) Synthesized Virtual Night Scenes.

Fig. 4. Experiment Result (Part 1). This figure shows the experiment results of synthesizing virtual urban night scenes at different places. For each row, the left subplot shows the daytime urban scene reconstructed from the OSM data given a specific location setting. The right subplot shows its corresponding urban night scenes synthesized with our approach.

consists of two subfigures, one showing the daytime urban scene reconstructed from OSM data, and the other showing the corresponding urban night scene synthesized with our approach. As we can see from the experiment results, the synthesis of each urban scene is visually striking, with the virtual urban night scenes appearing realistic and vivid. The lighting effects are immersive, with the illumination of buildings and streets appearing natural and accurate. This experiment result demonstrates the effectiveness of our approach in synthesizing virtual urban night scenes across different locations and highlights its potential applications in various fields, such as urban planning, virtual tourism, architecture design, etc. This figure presents the experiment results of synthesizing virtual urban night scenes at different locations which turns out to prove the potential that our automatic urban night scene synthesis approach can be generalized into other location settings.

Figure 5 presents the details of a synthesized virtual night urban scene from five different views rendered by five different cameras placed within the scene. The location setting for this scene is the area near the Gate of the Orient in Suzhou, China. The virtual urban night scene has been synthesized using our approach, and the results of the synthesis are visually impressive. The five different views provide a comprehensive and detailed view of the urban night scene from different angles. In the top-left subfigures, number labels ranging from 1 to 5 are specifying the position of each camera in this scene. Viewpoint 1 is in front of the Gate of the Orient. Viewpoint 2 is far away from the Gate of the Orient. Viewpoint 3 is above the water and looking towards the Gate of the Orient. Viewpoints 4 and 5 are close to the Gate of the Orient and some other landmarks. Each view provides a unique perspective of the urban landscape, highlighting different aspects of the scene,

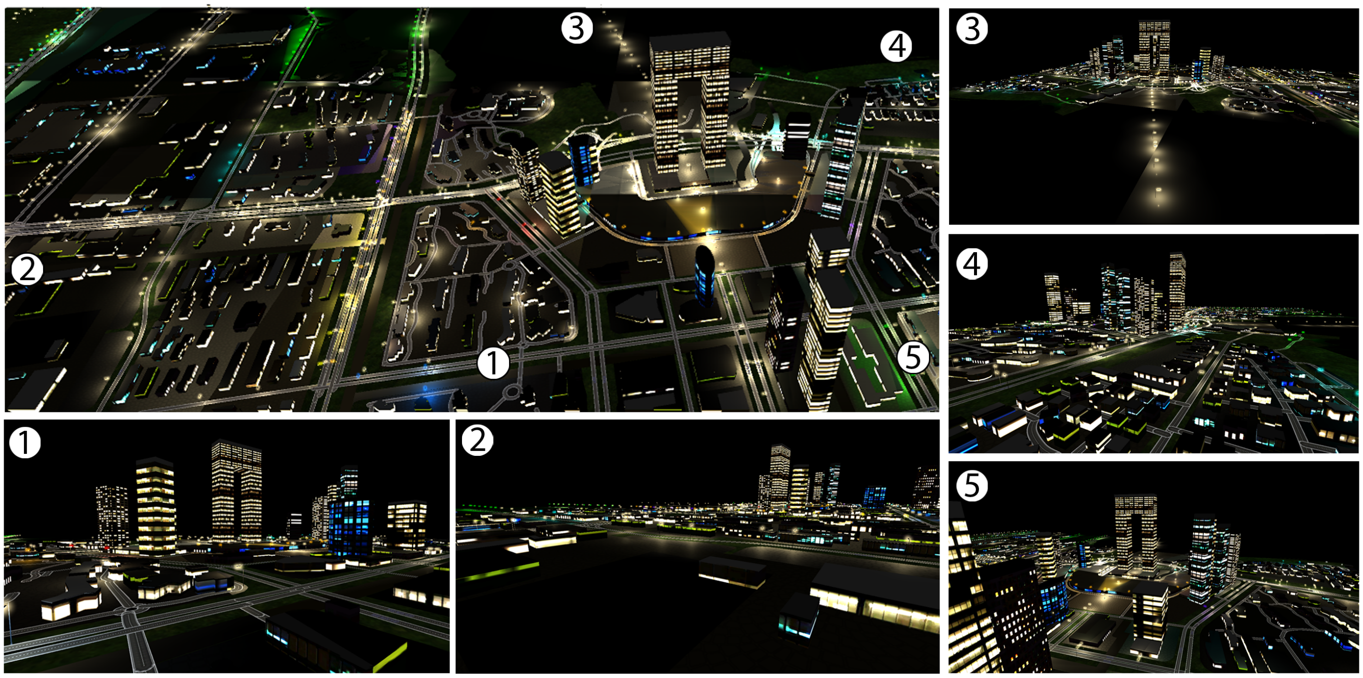


Fig. 5. Experiment Result (Part 2). This figure shows the details of synthesized virtual night urban scenes. Five different views are rendered from five different cameras placed in the virtual night urban scene that is synthesized with our approach.

such as the glow of streetlights, the reflections on buildings, and the shadows cast by the scene objects. The lighting effects in the scene are immersive, with the illumination of buildings and streets appearing natural and realistic. The figure demonstrates the effectiveness of our approach in synthesizing detailed virtual urban night scenes that can be viewed from multiple perspectives, highlighting its potential applications in various fields, such as game authoring, movie design, urban scene design, architectural visualization, etc.

VI. Conclusion

In this paper, we propose a novel procedural modeling approach to automatically synthesize realistic virtual night scenes on OpenStreetMap (OSM). Given the virtual urban scene is synthesized on OSM through Google Map API, after replacing the textures from daytime to night, another three steps are included for adding road lights such as extracting the road graph, sampling edge points, and adding road lights. In the end, we validate the efficacy of our proposed approach through the experiment results of urban night scenes synthesized on OSM from different locations and different viewpoints.

In future work, we expect to see further improvements in the quality and accessibility of synthesizing the virtual night scenes on OSM. Our approach can be applied to safety and security training in VR. By providing a realistic representation of the scene at night, users can identify potential hazards and take appropriate measures to mitigate them. Our work also has the potential to enhance tourism. Tourists can use the virtual night scene to plan their itinerary, identify places of interest, and navigate unfamiliar environments.

References

[1] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.

[2] P. Mooney, M. Minghini *et al.*, "A review of openstreetmap data," *Mapping and the citizen sensor*, pp. 37–59, 2017.

[3] J. J. Arsanjani, C. Barron, M. Bakillah, and M. Helbich, "Assessing the quality of openstreetmap contributors together with their contributions," in *Proceedings of the AGILE*, 2013, pp. 14–17.

[4] A. Pluta and O. Lünsdorf, "esy-osmfilter–a python library to efficiently extract openstreetmap data," *Journal of Open Research Software*, vol. 8, no. 1, 2020.

[5] D. Luxen and C. Vetter, "Real-time routing with openstreetmap data," in *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2011, pp. 513–516.

[6] P. Neis and D. Zielstra, "Recent developments and future trends in volunteered geographic information research: The case of openstreetmap," *Future internet*, vol. 6, no. 1, pp. 76–106, 2014.

[7] J. E. Vargas-Munoz, S. Srivastava, D. Tuia, and A. X. Falcao, "Openstreetmap: Challenges and opportunities in machine learning and remote sensing," *IEEE Geoscience and Remote Sensing Magazine*, vol. 9, no. 1, pp. 184–199, 2020.

[8] W. Li, "Make uber faster: Automatic optimization of uber schedule using openstreetmap data," in *Proceedings of the 2021 EURASIAGRAPHICS*, 2021, pp. 19–26.

[9] ——, "Simulating virtual construction scenes on openstreetmap," in *Proceedings of the 6th International Conference on Virtual and Augmented Reality Simulations*, 2022, pp. 14–20.

[10] ——, "Pm4vr: A scriptable parametric modeling interface for conceptual architecture design in vr," in *The 18th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*, 2022, pp. 1–8.

[11] W. Li, C. Li, M. Kim, H. Huang, and L.-F. Yu, "Location-aware adaptation of augmented reality narratives," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 2023, pp. 1–12.

[12] A. Grant, "Go map - unity 3d asset, 3d map for ar gaming," https://gomap-asset.com/, 2017.