**European Journal of Applied Sciences – Vol. 13, No. 04**
**Publication Date:** August 18, 2025
**DOI:**10.14738/aivp.1304.19230.

Lin, J.-W., & Chang, C.-C. (2025). An Augmented Reality System for Presenting Architecture Models in Outdoor Scenes. *European Journal of Applied Sciences, Vol - 13(04).* 271-280.

# An Augmented Reality System for Presenting Architecture Models in Outdoor Scenes

**Jhan-Wei Lin**
Department of Computer Science and Information Engineering,
National United University, Taiwan

**Chin-Chen Chang**
Department of Computer Science and Information Engineering,
National United University, Taiwan

## ABSTRACT

**Due to the rapid development of mobile devices, augmented reality (AR) has been utilized increasingly, and thus applications of AR have increased for indoor and outdoor scenes. However, most of them focus on dealing with indoor scenes and the research for outdoor scenes is rare. In this paper, we propose a system to explore an application of AR for presenting 3D architecture models in outdoor scenes. We aim to enhance the realism of a 3D architectural model for an outdoor scene and provide users with a realistic experience. We first capture a color image through a camera and detect a plane on the image. When obtaining the detected plane, we can generate an anchor and attach virtual objects (3D architecture models) to this anchor. After that, we use the environmental high dynamic range mode of ARCore to obtain lighting information. Thus, our system can present a more realistic light and dark contrast and shadows on the virtual objects. Moreover, we propose an occlusion handling technique to obtain the correct relative positions of real objects and virtual objects. We use a depth prediction model to obtain depth information enabling the real objects to occlude the virtual objects. Finally, the processed augmented color image is rendered. The experiments demonstrate that our system can produce favorable results within a reasonable distance.**

**Keywords:** Augmented reality, Occlusion handling, Depth prediction, Real-time rendering.

## INTRODUCTION

Augmented reality (AR) [1-2] is a technology that uses devices such as smartphones, tablets, or wearable displays, equipped with built-in cameras, sensors, or other peripheral devices to detect and collect necessary information in real-world scenes. It integrates digital information, such as images, videos, and 3D models, into the real-world environment, allowing users to experience and interact with virtual elements in real time.

During the early stages of augmented reality technology development, constrained by the limitations of software and hardware at the time, research in this field struggled to achieve significant progress in the integration of the virtual and real environments. In recent years, with the improvement in the performance of electronic devices, more research has focused on enhancing visual consistency between the virtual and real scenes to achieve a better user experience. Additionally, the applications of augmented reality have expanded from initial use in education and learning to various fields such as entertainment, advertising, and architecture.

Research on the application of augmented reality in outdoor scenes has been relatively limited, especially in the general presentation of architectural models [3-7]. This paper aims to develop an augmented reality system for real-time presentation of user-designed architectural models in outdoor scenes. Through methods such as ray tracing and depth prediction, environmental information is collected to enable the architectural model to reflect realistic lighting, draw shadows, and accurately handle occlusion. The goal is to enhance the realism of the architectural model in the scene, providing users with an experience where the building appears to exist realistically in the environment.

The proposed augmented reality system utilizes ARCore's high dynamic range (HDR) mode [8] for light source detection, allowing for the representation of light and shadow variations on virtual objects. Additionally, the system employs the FastDepth depth prediction model [9] to obtain depth information, enabling real-world objects to occlude virtual objects. Through the experiments, the proposed system has demonstrated the ability to enhance the realism of architectural models on the screen at an appropriate distance, thereby improving the user experience.

## RELATED WORKS

In research exploring the application of augmented reality in the field of architecture, most studies focus on integrating building information modeling (BIM) with augmented reality for use during or after the construction process. BIM involves the digitalization of information about architectural components, including geometric, geographic, and spatial information, during the design and construction phases. The use of BIM in the construction process can effectively reduce overall construction time and error rates, contributing to cost savings in various aspects of construction projects. Isnaeni [10] proposed an application that integrates building information modeling with augmented reality technology. Compared to using computers, the system leverages the convenience of mobile devices, allowing users to easily access various information within the BIM during the construction process. This facilitates more effective supervision and management of the entire construction process.

In the research on augmented reality applications in the architectural field, several studies focus on heritage-related aspects. Augmented reality guiding addresses these challenges by pre-recording guiding information as audio explanations. Visitors can use their mobile devices to interact with points of interest, obtaining corresponding guiding information. This not only saves manpower and resources in the exhibition but also provides visitors with more freedom and interactivity during the guiding process. Marker-based augmented reality is known for its stable marker recognition functionality. However, it requires the placement of black-and-white barcodes, causing minimal environmental impact. One drawback is that virtual objects are confined to appearing only on the barcodes.

Verykokou et al. [11] aimed to recreate the central colonnade of the ancient Agora in Greece. They proposed a markerless augmented reality system, relying on terrain features to identify specific scenes. This allowed them to present the heritage model of the central colonnade at designated locations on the screen. Relevant architectural information and historical culture were collected, and the simultaneous localization and mapping (SLAM) [12] technology was employed to scan the Saikyoji Temple in Japan. This data was then used to estimate and establish a model of the Saikyoji Temple in Taipei. Augmented reality was used to guide visitors to explore its former site and gain insights into its historical appearance and cultural background. Markerless augmented reality eliminates the need for additional barcodes but can be influenced by lighting conditions, and user prompts may be required for a larger recognition range to ensure a smooth user experience.

However, whether in heritage guiding or heritage reconstruction, both augmented reality systems share a common prerequisite: the need to predefine a specific location. This implies that with a change in location, preprocessing is required for the new site; otherwise, the system becomes ineffective. Furthermore, heritage guiding focuses more on providing users with relevant information about the heritage site, while heritage reconstruction aims to collect past information to faithfully recreate the original appearance of the heritage. Both applications generally do not prioritize the realism of virtual objects.

## THE PROPOSED SYSTEM

We develop an augmented reality system designed to present virtual objects in various outdoor scenes. We adopt an interactive augmented reality system design, allowing users to independently determine the location for rendering virtual objects.

### Pretrained Depth Prediction Model

To obtain depth information [9, 13, 14] for occlusion processing in the system, we utilize the FastDepth depth prediction model [9] to generate depth maps. However, the original dataset was NYU Depth v2 dataset [15], consisting exclusively of indoor scene images. To adapt the model for outdoor scenes, we conduct training using the KITTI dataset [16]. The KITTI dataset offers nearly 7500 images collected by various onboard sensors in diverse outdoor scenes, providing high-resolution RGB images along with corresponding depth images. After training, we can obtain the depth prediction model tailored for our augmented reality system. Figure 1 demonstrates the results of generating depth maps through the model.

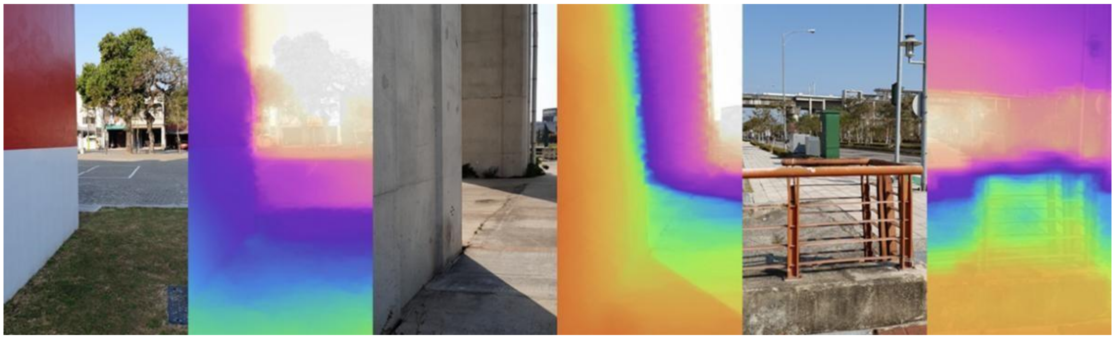


**Figure 1: Three depth maps generated through the depth prediction model.**

### User Interface

The user interface of our augmented reality system is shown in Figure 2. The red dot at the center of the screen serves as an indicator for plane detection. It turns green when a plane is detected. Among the buttons at the bottom, "OCCLUSION" toggles the occlusion processing feature, "LIGHT" activates/deactivates the light detection function, and "SHADOW" enables/disables the shadow rendering feature. These three functionalities are initially turned off. The "PLACE" button is used to position virtual objects, while "PHOTO" is utilized to capture the current image.

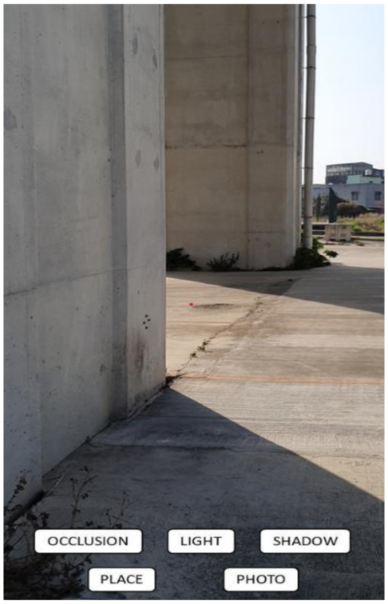


**Figure 2: User interface of the proposed system.**

## System Flow

The system flow of the proposed augmented reality system is shown in Figure 3, divided into six stages as follows:

- **Stage 1. Image Information Acquisition:** We capture the original color image through the camera and detect planes. ARCore continually processes the image, detecting feature points, tracking their threedimensional spatial information, and forming planes based on adjacent positions or patterns of similar feature points.
- **Stage 2. Placing Virtual Objects:** In the presence of detected planes, users can click the corresponding button. Through ray casting for collision detection, when hitting the detected plane, we generate an anchor and attach the virtual object to this anchor. This ensures that ARCore can track the object's position, preventing it from shifting due to ARCore updating environmental information. Subsequently, the virtual object is rendered on the GPU, allowing access to virtual color and depth images in its buffer.
- **Stage 3. Light Source Detection:** When the corresponding function is enabled, we can use the original color image to obtain lighting information. ARCore's environmental HDR mode employs machine learning to analyze the current scene, predicting the main direction and intensity of real light sources. It also calculates the environmental spherical harmonics to supplement overall lighting generated from all directions, thereby rendering the corresponding brightness and darkness on virtual objects.
- **Stage 4. Shadow Rendering:** If the function is enabled and the light source detection function is active, using pre-acquired lighting information, shadows corresponding to the main light source direction are drawn on the detected plane.

- **Stage 5. Occlusion Processing:** With occlusion processing enabled, we utilize a pre-trained depth prediction model to generate the original depth image. By comparing the virtual depth image with the original depth image, occlusion processing is performed.
- **Stage 6. Image Output:** The processed augmented color image is rendered on the device screen.

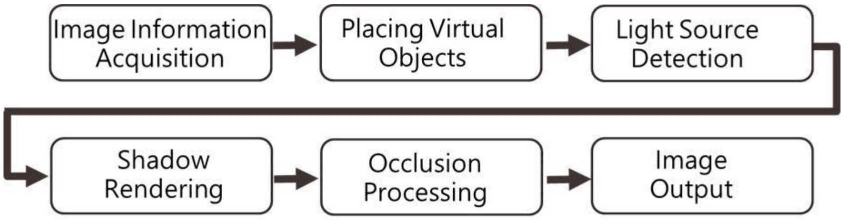


**Figure 3: System flow.**

## Lighting and Shadows

As augmented reality applications become increasingly widespread, more research is being conducted to enhance the fusion of virtual and real elements. The consistency between virtual and real elements on the screen is crucial for the user experience. The interaction of virtual objects with real-world lighting is a significant aspect of achieving this consistency. By analyzing lighting information, such as intensity and direction, virtual objects can exhibit phenomena like refraction and reflection, and corresponding shadows can be rendered, enhancing the realism of virtual objects.

The environmental high dynamic range (HDR) mode is a lighting assessment mode provided by ARCore [19]. It utilizes machine learning to analyze the current camera image, enabling real-time acquisition of lighting intensity, primary light source direction, environmental spherical harmonics, and cube maps. Lighting intensity and the direction of the light source affect the shadows on virtual objects and the variations in brightness on their surfaces. The cube map allows the surfaces of objects to reflect information from the surrounding real environment, making the appearance of virtual objects more closely resemble the real environment.

## Occlusion Handling

Occlusion handling involves rendering scenes based on the depth information from the real environment to achieve the correct occlusion of virtual objects by real ones. The most effective and easily achievable method for obtaining depth information is using a depth camera. However, depth cameras, due to their lower resolution and optimal processing range of less than one meter, are not suitable for medium to long-distance outdoor scenes. On the other hand, with the improvement in hardware capabilities, the time required for training models in deep learning has significantly decreased. Related studies, such as optical flow prediction and depth prediction, have been employed to address occlusion handling issues. Although the accuracy of depth information obtained through deep learning models may not match that of depth cameras, their practical application is preferred due to fewer limitations.

In the depth map, the values define the position of a two-dimensional pixel along the $Z$-axis in threedimensional space. Therefore, knowing these values allows us to determine the relative distance, whether far or near, of each point to the camera position. Let the original color image be denoted as $R$, the corresponding original depth image generated by the depth prediction model as $R_D$, the virtual color image obtained by rendering virtual objects as $V$, and its corresponding virtual depth image as $V_D$. Through the simple comparison by equation (1), we can obtain the augmented color image $A$.

$$A(x, y) = \{\ R(x, y), if\ R^{D(x,y) < VD(x,y),}\ (1)\ V(x, y), otherwise.$$

## RESULTS

The device used for actual testing was the Samsung Galaxy Tab S8. The experimental scenes included various outdoor locations, aiming to investigate the performance in different contexts.

### Results of Occlusion Handling

The experimental results of occlusion handling were shown in Figure 4. We conducted tests in two different scenes, and the 3D model of the purple Android logo represents the virtual object. The images from left to right demonstrated the occlusion handling effects in scenes where the virtual object was at varying distances. From the first three images in each scene, we observed that within a distance of 15 to 20 meters, occlusion handling using the pre-trained depth prediction model yielded satisfactory results. However, in the fourth image, when the distance was extended beyond 20 meters, noticeable errors in occlusion handling appeared.

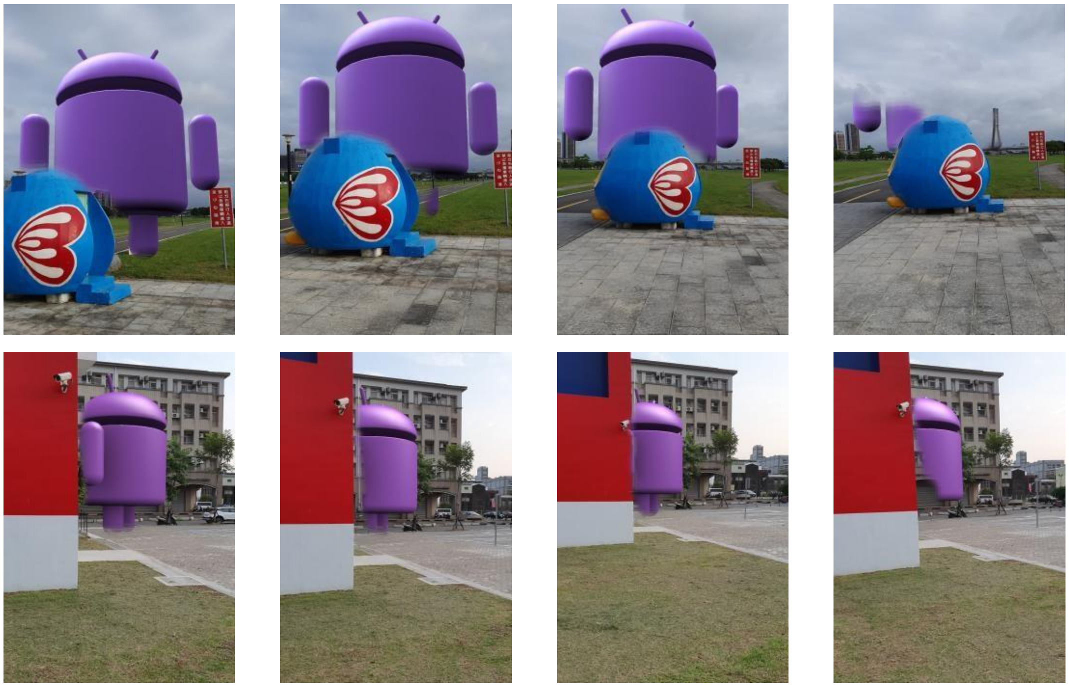


**Figure 4: Results of augmented reality occlusion handling system.**

Moving on to Figure 5, even without extending the distance, we can observe severe errors in the occlusion handling results in the red-boxed region. We attributed this issue to the depth prediction model. To enhance the efficiency of the prediction process, the model initially classified pixels in the image through semantic segmentation. As illustrated in Figure 6, comparing the occlusion handling results with their corresponding depth maps, the ideal scene would have lower depth values in the gaps between the railings, indicating a greater distance from the camera and partially revealing the virtual objects. However, due to semantic segmentation, this region was classified as the same object category, resulting in similar depth values and consequently inaccurate occlusion handling results.

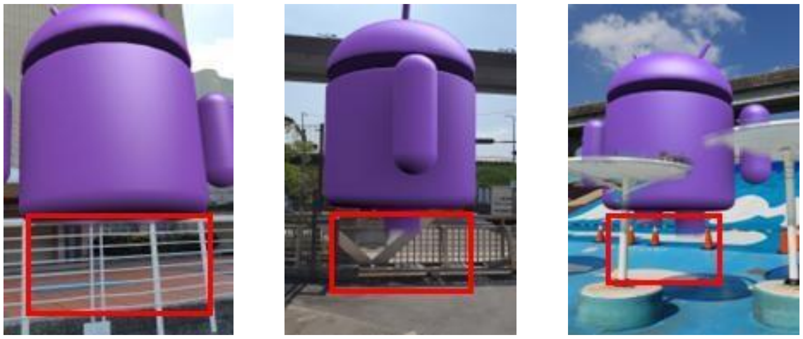


**Figure 5: Impact of semantic segmentation on occlusion handling.**

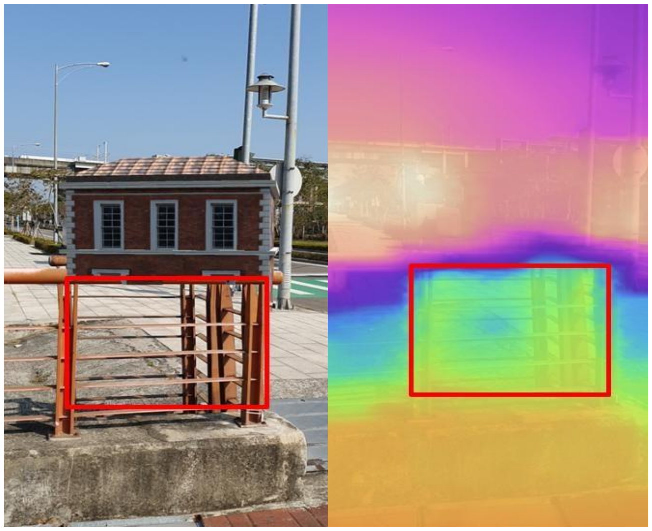
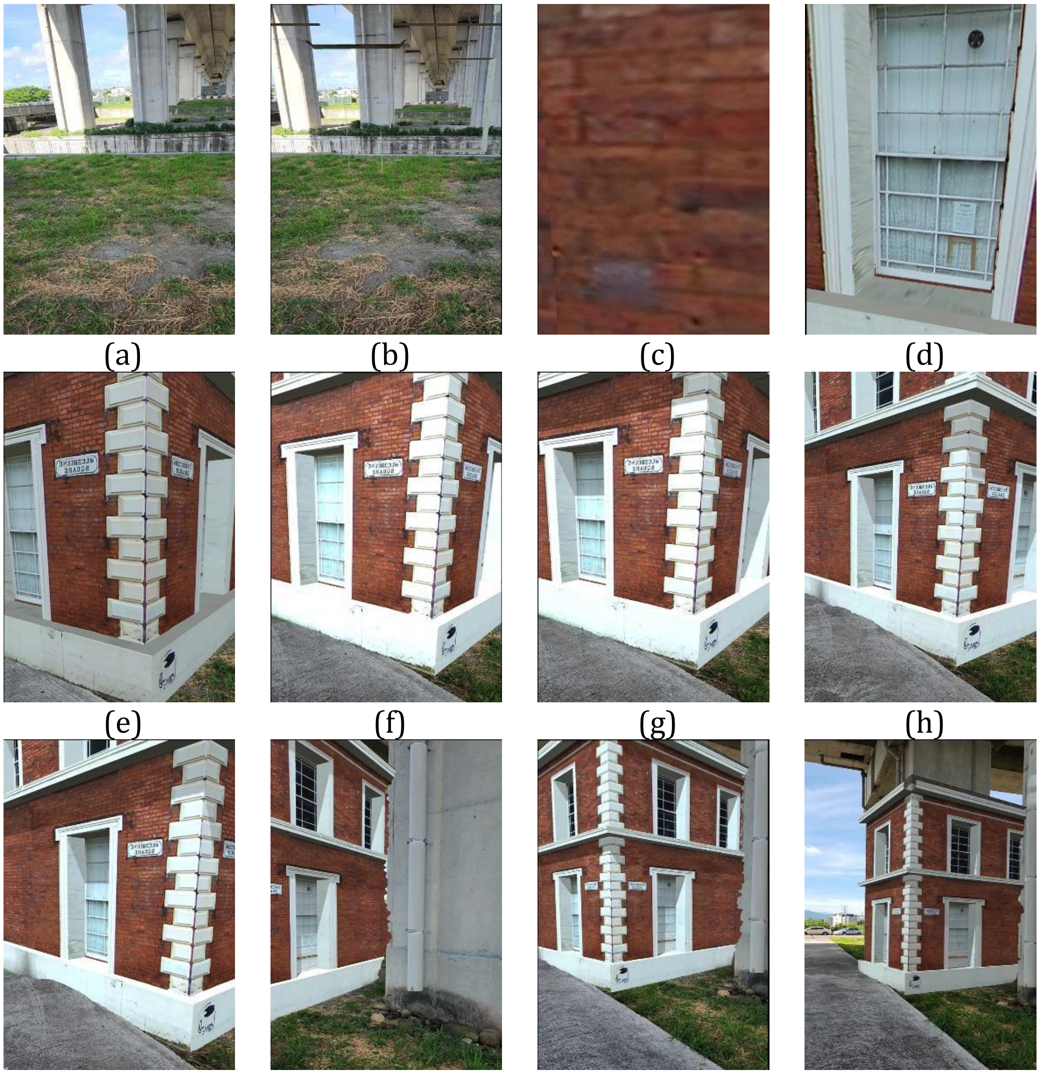


**Figure 6: Occlusion handling result and their corresponding depth map.**

### Full Feature Testing

We presented the results of testing with all three features enabled. First, we examined Figure 7, where the results were arranged from left to right and top to bottom based on the increasing distance. In this context, (a) represented the scene after detecting the plane but before placing the building model. (b) showed the result after placing the building model. Since the utilized building model was in a 1:1 real scale, the entire model was not directly observable immediately after rendering the virtual object. As the distance increased, we can gradually get a complete view. Because all features are initially turned off, (e) to (h) showed the progressive contrast. (f) represented the result after enabling light detection, (g) with shadow rendering enabled, and (h) with occlusion processing activated. Subsequently, (i) to (l) displayed the results with all features enabled as the distance gradually increased.



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

(e) (f) (g) (h)

(i) (j) (k) (l)

**Figure 7: Full feature testing results.**

### Execution Time for Each Feature

Finally, to assess the real-time interactivity of our augmented reality system for users, this experiment statistically analyzed the time consumption for four key features, averaging their execution times as presented in Table 1. The most time-consuming task was plane detection, taking an average of 3.6 seconds. In practice, this longer time is primarily required at the system's initiation, where users needed to gradually move the device screen to detect and collect a sufficient number of feature points to construct virtual spatial coordinates. As the augmented reality system was continuously used and environmental information was gathered, the execution time for plane detection decreased accordingly. The second-highest time-consuming task was occlusion processing. However, due to the use of a lightweight depth prediction model, the average time spent was only 113 milliseconds, and users typically do not experience noticeable lag. Following this was the light detection of the highdynamic-range mode, which took 35 milliseconds, less than one-third of the time required for occlusion processing. Finally, shadow drawing, relying on the lighting information obtained directly from light detection, had a minimal execution time of only 18 milliseconds.

**Table 1: Execution time.**

| Feature | Average Execution Time |
|---|---|
| Plane Detection | 3.6 seconds |
| Lighting | 35 milliseconds |
| Shadow | 18 milliseconds |
| Occlusion | 113 milliseconds |

## CONCLUSION

In this paper, we have presented an interactive AR system designed for presenting architectural models in outdoor environments. To achieve visual consistency between the architectural models and the real environment, we utilized ARCore's HDR mode for light source detection. This allowed us to effectively calculate the direction and intensity of the light source, enabling the architectural models to exhibit appropriate brightness contrast and cast shadows on the ground in the corresponding direction. Additionally, to ensure that real objects in the environment correctly occlude the architectural models on the screen, we employed the lightweight depth prediction model. In practical tests, the occlusion processing function demonstrated good results within a reasonable distance.

Despite achieving considerable consistency between the virtual and real elements under specific conditions, there were still several aspects to improve in the fusion of augmented reality in outdoor scenes. The depth information inaccuracies caused by semantic segmentation in specific scenes could potentially be addressed by employing more accurate depth prediction models, such as the optical flow prediction model [17]. Moreover, future improvements may be possible with advancements in hardware capabilities or updates in neural network models, providing better solutions to address these challenges.

## ACKNOWLEDGMENTS

The authors would like to thank the National Science and Technology Council of Taiwan for financially supporting this research under contract no. NSTC 114-2221-E-239-024-.

## References

[1]. Azuma, R., A Survey of Augmented Reality. Presence: Teleoperators and Virtual Environments, 1997. 6(4): p. 355–385.

[2]. Rosenberg, L.B., Virtual Fixtures: Perceptual Tools for Telerobotic Manipulation. Proceedings of IEEE Virtual Reality Annual International Symposium, 1993: p. 76–82.

[3]. Abidin, S.Z., Omar, M.Z., and Tahir, N.M., Augmented Reality in Outdoor Settings: Evaluation of a Hybrid Image Recognition Technique. International Journal of Architectural Computing, 2021. 19(2): p. 180–196.

[4]. Bekele, M.K., Pierdicca, R., Frontoni, E., Malinverni, E.S., and Gain, J., Architecture MAR: A Mobile Augmented Reality System for Historical Building Reconstruction. ISPRS International Journal of Geo-Information, 2022. 7(12): p. 463.

[5]. Chiu, H.K., Zhang, Y., Mildenhall, B., Barron, J.T., and Srinivasan, P.P., Neural Light Field Estimation for Street Scenes with Differentiable Virtual Object Insertion. arXiv preprint, arXiv:2208.09480, 2022.

[6]. Fernández-Palacios, B.J., Morabito, D., and Remondino, F., LagunAR: A Mobile AR Application for Outdoor Visualization of Historical Architecture. Sensors, 2023. 23(21): p. 8905.

[7]. Fuchs, C., and Broll, W., GHAR: GeoPose-based Handheld Augmented Reality for Architectural Positioning, Manipulation and Visual Exploration. arXiv preprint, arXiv:2506.14414, 2025.

[8]. Google Developers, Updates to ARCore Help You Build More Interactive & Realistic AR Experiences. [Online] Available: https://developers.googleblog.com/2019/05/ARCoreIO19.html, May 2019.

[9]. Wofk, D., Ma, F., Yang, T., Karaman, S., and Sze, V., FastDepth: Fast Monocular Depth Estimation on Embedded Systems. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2019: p. 6101–6108.

[10]. Isnaeni, N.N., The Applicability of Augmented Reality (AR) and BIM Integration on Under Construction Project. Master Thesis, Department of Civil and Construction Engineering, National Yunlin University of Science & Technology, 2021.

[11]. Verykokou, S., Ioannidis, C., and Kontogianni, G., 3D Visualization via Augmented Reality: The Case of the Middle Stoa in the Ancient Agora of Athens. Proceedings of the 5th International Conference on Cultural Heritage, EuroMed, 2014: Limassol, Cyprus.

[12]. Durrant-Whyte, H., and Bailey, T., Simultaneous Localization and Mapping: Part I. IEEE Robotics and Automation Magazine, 2006. 13(2): p. 99–110.

[13]. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint, arXiv:1704.04861, 2017.

[14]. Yang, T.J., Howard, A., Chen, B., Zhang, X., Go, A., and Sadler, M., et al., NetAdapt: PlatformAware Neural Network Adaptation for Mobile Applications. Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[15]. Silberman, N., Hoiem, D., Kohli, P., and Fergus, R., Indoor Segmentation and Support Inference from RGBD Images. Proceedings of the European Conference on Computer Vision (ECCV), 2012: p. 746–760.

[16]. Geiger, A., Lenz, P., Stiller, C., and Urtasun, R., Vision Meets Robotics: The KITTI Dataset. International Journal of Robotics Research, 2013.

[17]. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., and Brox, T., FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: p. 2462–2470.