www.cad-journal.net
Computer-Aided Design and Applications

AI-generated VR: Leveraging AI and VR for Rapid Ideation and Concept Modeling of Interior Design

Chieh-Jen Lin

Tainan University of Technology, t60011@mail.tut.edu.tw

Abstract. This study explores the integration of Generative Artificial Intelligence (GenAI) and Virtual Reality (VR) in interior design, proposing a novel methodology that leverages AI to generate panoramic views consistent with 3D models for VR ideation. Utilizing Stable Diffusion along with panoramatic style LoRA and ControlNet models, this paper demonstrates an efficient approach for aligning AI-generated content with a designer's concepts. By developing Python-based components for Grasshopper in Rhino, the research enables the rapid generation of different types of controlling images from parametric models with low LOD, to control AI-generated panoramas to enhance the design ideation process. This rapid generation method solves the problem of traditional VR production being too cumbersome in specifying materials and lights, which hinders initial ideation application. In addition, sharing panorama via social media reduces the hardware requirements for sharing VR content, provides a more immersive experience than traditional conceptual models, and opens up new avenues for creative imagination.
This study represents a significant stride towards enriching the toolkit available to designers, merging the power of generative AI with the immersive capabilities of VR to bolster creativity and efficiency in interior design.

Keywords: Virtual Reality (VR), Parametric Modeling, AI-generative Panorama, Interior Design, Conceptual Ideation.
DOI: https://doi.org/10.14733/cadaps.2025.629-639

1 INTRODUCTION

With the decline in hardware and software costs and improved computational efficiency, virtual reality (VR) is making a resurgence in design education and practices [6]. VR offers superior communication to 2D drawings and 3D renderings, delivering a spatial experience closest to the actual built result. However, VR is always limited by difficulties in creating available 3D models and is mainly used only for final presentations rather than as an ideation tool or discussion medium in the early design stage. Limitations also arise in the manipulation within VR, allowing only a limited placement and modification of existing objects like furniture and materials within predefined environments. More proper design approaches adapted to VR applications hinder its widespread adoption in the early design stage. Therefore, the previous research proposed a generative modeling approach based on the Level of Development (LOD) concept by using Grasshopper to transform bubble diagrams of a single-story house into orthogonal plans using Voronoi diagrams [14]. This approach allows the quick generation of interior partition concept models that can be previewed in VR, making it more suitable for interior design education and practices than physical models. The quickly generated 3D indoor prototype can only provide the background for placing various objects in VR. However, the rapid generation of more details for design ideation and discussions, such as lighting, material texture maps, and other smaller objects for different style ideas and design concepts, is still a laborious process and full of challenges.

With the emergence of generative artificial intelligence (GenAI) technologies, like Midjourney, DALL-E by OpenAI, and Stable Diffusion by Stability AI, capable of generating images from simple text prompts and image inputs, are beginning to be applied in the creative ideation phase of architectural and interior design. Research indicates that AI-generated contents have the potential to nurture designers' inspiration, creativity, and skill development [4, 21]. However, GenAI can now only generate 2D images or rough 3D object models for now, and the generated results are random. Therefore, controlling the consistency of the different generated stages is complicated. An exterior perspective can only reveal one or two building facades for architectural design. Controlling the viewing angle to generate the building's appearances in different directions skillfully avoids inconsistency. However, for interior design, a single indoor perspective will show two or three interior walls in addition to the ceiling and floor. Hence, it is often difficult to solve the problem of inconsistency by only controlling viewing angles. Therefore, A more feasible way is to generate images of the entire indoor space at once, applying AI to generate a panoramic view of an indoor space. Based on the AI-generated panorama, it is possible to develop further 3D models for applying VR as an ideation and discussion tool. Therefore, this study proposes using Stable Diffusion with panorama LoRA [11], ControlNet models [25], and low LOD 3D models to generate panorama for VR that align with users' proposed design concepts.

2 THE APPROACHES OF AI-GENERATED VIRTUAL REALITY

The advantage of VR is that it can provide correct proportions among the human body with design objects, but also a 1:1 spatial experience. However, it still cannot replace the application of physical mock-up models in education and practice because it needs computers and software and takes a lot of work to create a usable 3D model. On the other hand, VR devices, whether glasses or headsets, are not only expensive but also usually quite bulky. In addition to the content provided by the game platform, self-made content often requires a computer with a high-end GPU to offer high-speed rendering capabilities for high-quality content without appropriate compilation. One of the easiest ways to display and share VR content is through panoramic image sharing on Facebook, which can quickly display the same content between smartphones, tablets, and VR devices. Although the panorama is not an accurate 3D model and lacks the spatial navigation function and object operability, it is usually enough for quick sharing and creative ideations. Especially when viewed through VR devices, such as Meta Quest 2 or 3, users can directly open a Facebook panorama to get a close-to-real experience of entering an indoor or outdoor space. Using Gen AI technology, it can not only quickly generate panoramas but also is possible to obtain additional geometric information through AI analysis, such as canny, depth, normal maps, semantic segmentation, and other features, which can be applied to regenerate 3D prototypes by the generated panorama quickly.

Stable Diffusion, developed by Stability AI [20] based on the Latent Diffusion Model [19], is a popular open-source text-to-image (t2i) and image-to-image (i2i) generating model. Unlike other Latent-Diffusion-based but private models such as Midjourney or DALL-E, Stable Diffusion not only allows local and remote installation but also derives many personal training fine-tuning models (called checkpoints) and extra networks models such as Embedding, LoRA, and ControlNet models. These model files are available from AI community websites such as Hugging Face [12] or CivitAI [3] and can be manipulated through the Stable Diffusion WebUI developed by AUTOMATIC1111 [2] (hereinafter referred to as SD-WebUI). For generating a panorama by Stable Diffusion, an extra LoRA (Low-Rank Adaptation) model must be used to modify the generating style to generate a panoramic-style image. LoRA is a novel approach proposed by Microsoft's researchers, who suggested modifying specific layers' weights in deep learning models in a low-rank manner, achieving quick and effective adaptation to new tasks while avoiding extensive retraining of the original model [11]. For generating image tasks, a LoRA model can affect the overall style or the theme of the generated image through a specific token and weighting. By applying a panoramic style LoRA model, generating a panorama by inputting text prompts is easy. However, it still cannot avoid the problem of wrong proportions among the space and objects inside.

To solve the problem of wrong proportions among the space and objects inside, the ControlNet models can be applied by the features of a given reference image, including contours, depth, and normal maps, semantic segmentation, and so on [25]. Due to the powerful capabilities of ControlNet models, some plugins or apps for 3D modeling and BIM software, such as SketchUp Diffusion [24], Ambrosinus Toolkit [1] for Grasshopper, AI Visualizer for ArchiCAD [9], and Veras for Revit [5], has been introduced as an alternative tool for real-time photorealistic 3D rendering. However, just like the problem with VR applications, the creative ideation stage is unnecessary and usually takes no time to build a high LOD model for rendering by a rendering engine or AI. In the creative ideation stage, it is more feasible to build a low-LOD conceptual model quickly, but too few geometric features make it difficult to apply AI rendering. On the other hand, it is still necessary to provide a panoramic style reference image through this low LOD model to control the panoramic LoRA results.

Most 3D rendering software, such as Vray, Enscape, Lumion, and Twinmotion, can easily import 3D models to generate panoramas. However, the software aims to produce photorealistic rendering effects. Therefore, they need high-LOD models to provide enough geometric features, which usually render too many unnecessary features for applying ControlNet. A line drawing, such as an image generated by Pen or the Technical display mode of Rhino, is more suitable for ControlNet to generate panoramas without considering detailed styles, materials, and lighting beforehand. However, most 3D or BIM software has no function to convert a line drawing into a panorama directly. This study, therefore, develops Python-based Grasshopper components for generating a panorama based on the display mode of the given view. This approach can control the necessary features of the panoramic image for applying different ControlNet models and, therefore, is not limited to the contour features of most AI-rendering apps.

Based on the above technical requirements and explanations, this study proposes using Rhino and Grasshopper as the platforms to apply GenAI to generate panorama from low LOD 3D models for generating VR quickly. The approach of this study to develop Grasshopper components shows as Fig. 1.

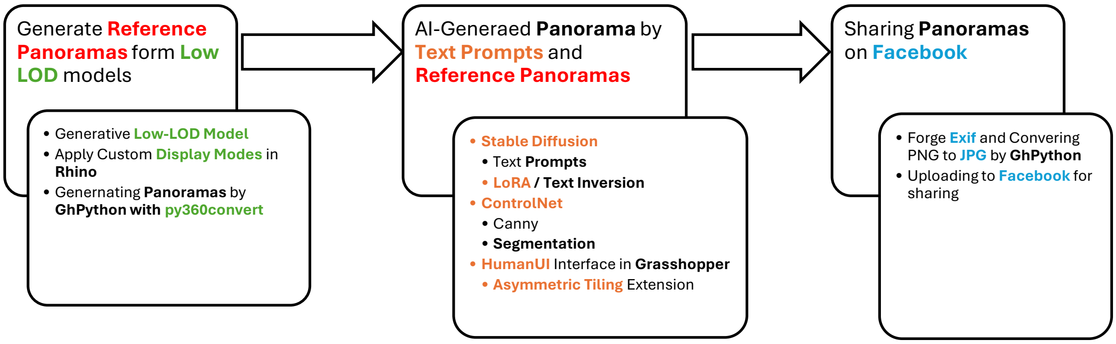

Figure 1: The flowchart of the AI-generated VR: (a) To generate different types of reference panoramas from low-LOD models; (b) To apply text-to-image Stable Diffusion with LoRA and ControlNet for generating panoramas; (c) To share and display the generated panoramas on Facebook.

2.1 Generation of Controlling Images for AI-generated Panoramas from the Given 3D Prototype

AI-rendering apps mostly only apply the contour (canny) features of the given 3D model. Its advantage is not only that the generated image can be highly close to the contour features of the given model, but it also can generate other rich image features by simple text prompts, such as materials and lighting. However, its shortcomings are also evident. First, a higher LOD model is required to provide sufficient contour features, which are usually not available in the ideation stage. Still, the images generated by a low LOD model may differ significantly from the designer's ideas because the contour feature is actually a design style. Secondly, oversimplified contours cannot guarantee that the correct objects will be generated. For example, based on the canny map from a low LOD model (Fig. 2a), ControlNet correctly generated the chairs at the table on the left of the image because of enough features. Still, the sofa on the right of the image cannot be generated because its contour is too simplified (Fig. 2b). The problem of randomly generating objects based on text prompts is more severe for interior design than architectural design.

Figure 2: The AI-generated images from a low LOD model: (a) A low-LOD model applying the "Pen" display mode in Rhino for generating a canny controlling map, (b) two generated images by the canny ControlNet model.

2.1.1 Semantic Segmentation Maps for ControlNet

The solution to the above two shortcomings is the semantic segmentation model of ControlNet, which was initially used to segment objects and tag semantics in an image. For example, the ADE20K developed by MIT can recognize 150 classifications of objects. [26], and the COCO developed by Microsoft can recognize only 80 classifications [15]. The segmentation preprocessing of ControlNet can apply ADE20K or COCO to generate a segmentation map from a given image. This segmentation map uses color to distinguish the object types and their regions in the image. Based on this map, ControlNet can generate the correct types of objects in the image. In addition, to generate segmentation maps from a given image, users can directly provide the map by hand-drawing or image-processing software. Therefore, if a segmentation map can be generated through a low LOD model, even without detailed geometric features, objects with correct proportions can be generated at the users' expected position while keeping the flexibility of other characteristics, such as contours, colors, and materials. Therefore, this research created a Rhino template based on ADE20K's 150 categories. There are 150 layers created in the template with both English and Chinese names of ADE20K classifications, and each layer was assigned the correct classification color. For example, the hexadecimal code of the kitchen classification color is #33FF00; the sofa classification is #0B66FF. Then, by modifying the technical display mode in Rhino, this research develops a custom display mode that can correctly display the classification colors of ADE20K (Fig.3a) for users to generate a segmentation map from a low LOD model. According to this segmentation map, the proper objects can be generated at the correct regions in the images, such as the sofa on the right of the image that cannot be generated in the previous canny example and the kitchen island countertop on the left side of the images (Fig.3b). In addition, more specified objects are generated due to the segmentation map, such as the floor-to-ceiling windows with views and curtains at the rear wall in the images (Fig.3b).

Figure 3: The Rhino template based on the ADE20K classifications: (a) A segmentation map generated by a low LOD model applying the custom display mode of Rhino, (b) two generated images by the segmentation ControlNet model.

2.1.2 Equirectangular Projection of Different Types of Controlling Maps

As mentioned above, the panoramic results of 3D rendering software are too realistic, and the 3D modeling software cannot convert line drawings into panoramic images. Therefore, this study develops GhPython components to generate panoramic images from the camera position of a given Rhino's scene. Those GhPython components use a Python library named py360convert [22] to convert the front, rear, left, right, and upper projection images (Fig. 4b) into a 1:2 ratio equirectangular projected panorama (Fig. 4c). Since Rhino 8 began to support Python 3 with the standard Python Package Index (pip) function, that can automatically download the py360convert and its depending on packages, such as OpenCV and Numpy. This panoramic function can be implemented in some GhPython components of Rhino 8 without additional rendering apps, and it is difficult to implement in the old version GhPython of Rhino 7, which only supported the particular version of Iron Python 2 without pip. For generating different types of controlling images, however, the rendering style of this panorama still needs to be set in the current Rhino view in advance, such as Shaded, Technical, Monochrome, Pen, or Arctic display mode for generating necessary image features. For example, the Arctic display mode can also generate a panoramic image for apply the canny ControlNet model. Therefore, by applying the previous ADE20K template with the custom display mode, a panoramic segmentation map can be quickly generated by the GhPython components (Fig. 5c).

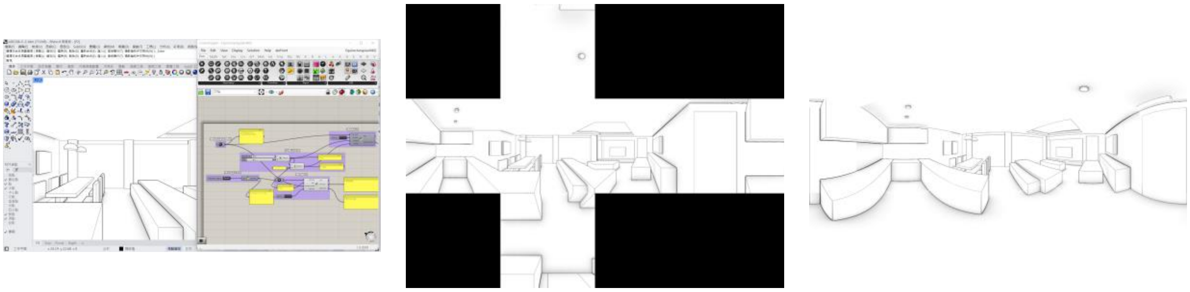

Figure 4: Test generation of an equirectangular projection image from a scene in Rhino: (a) Setting the Arctic display mode of the current view, (b) the generated cubic projection images, and (c) the converted equirectangular projection image.

2.2 Generation of Panoramas by Text Prompts with Stable Diffusion, ControlNet, and LoRA

SD-WebUI can be installed on cloud computing platforms such as Google Colab. However, since Google Colab has banned the free GPU usage times to generate images by SD-WebUI, installing SD-WebUI on a local PC has become a relatively cheap option but is itself a technical challenge.

Figure 5: The generation of a panoramic segmentation map from a scene in Rhino: (a) Setting the custom display mode for ADE20K semantic segmentation, (b) the generated cubic projection images, and (c) the converted equirectangular projection image.

Fortunately, there is a package management software named StabilityMatrix [17] that can install SD-WebUI and other similar WebUI packages such as Fooocus [16] with one click. It also allows the downloading and managing of model files, including checkpoints, embedding, ControlNet, LoRA, and other types of model files from CivitAI and Hugging Face, reducing the technical barriers to applying Stable Diffusion.

In addition, since SD-WebUI provides an Application Programming Interface (API) that allows remote operation of t2i and i2i generation, so it allows 3D modeling software to generate images remotely. There is already a Grasshopper toolset named Ambrosinus Toolkit (ATK) [1] that provides the basic operation of t2i, i2i, and ControlN functions by the API of SD-WebUI. However, ATK doesn't support manipulations of other extensions, such as Asymmetric Tiling [23] for seamless panoramas. On another hand, ATK cannot automatically retrieve important features of current SD-WebUI installation such as the names of checkpoints, samplers, and ControlNet models. Therefore, inspired by ATK, this study applies the HumanUI with GhPython in Grasshopper to develop our interface that can remotely operate the SD-WebUI to generate panorama images by Stable Diffusion and ControlNet in Grasshopper.

2.2.1 HumanUI Interface for Generating Panorama Images by Stable Diffusion and ControlNet

This study used the GhPython component of Rhino 8 to develop Python 3 scripts to operate the t2i generation with ControlNet functions remotely (Fig.6a). However, the original interface of Grasshopper needs to be more friendly for the operations of Stable Diffusion and ControlNet that require inputting reference images and other parameters. Therefore, this study applies the HumanUI components to develop a simplified version of the SD-WebUI interface to keep the necessary parameters of Stable Diffusion but to allow enough operations of ControlNet within the Grasshopper.

For designers who only use AI-generated images to assist interior or architectural design, the original interface of SD-WebUI is too complicated. Therefore, an extremely simplified SD-WebUI named Fooocus appeared [16], and it covers most parametric options other than prompt input and simplifies the basic parameter settings such as resolution and style. Fooocus is suitable for beginners who want to focus on writing prompts to generate images, but it lacks necessary operations for different types of ControlNet models. Therefore, the interface developed in this study also blocks most of the parameters of the SD-WebUI. In addition to the prompt field input, only necessary parameters such as checkpoint and sampler selection, resolutions, configuration scale (CFG), and seed are retained, then the URL of the remote server and the local path for image storage is added (Fig. 6b). The necessary input parameters of ControlNet are also simplified but still retain the ability to apply up to three ControlNet models simultaneously (Fig. 6c).

2.2.2 Asymmetric Tiling Extension for Generating Seamless Panorama Images

In addition to ControlNet and LoRA, the SD-WebUI needs to install additional extensions to generate an seamless panorama for displaying on a VR device. AI-generated panoramas, such as those generated by Midjouney, often have a seam when the pixels on the left and right ends of the panorama cannot be connected smoothly.

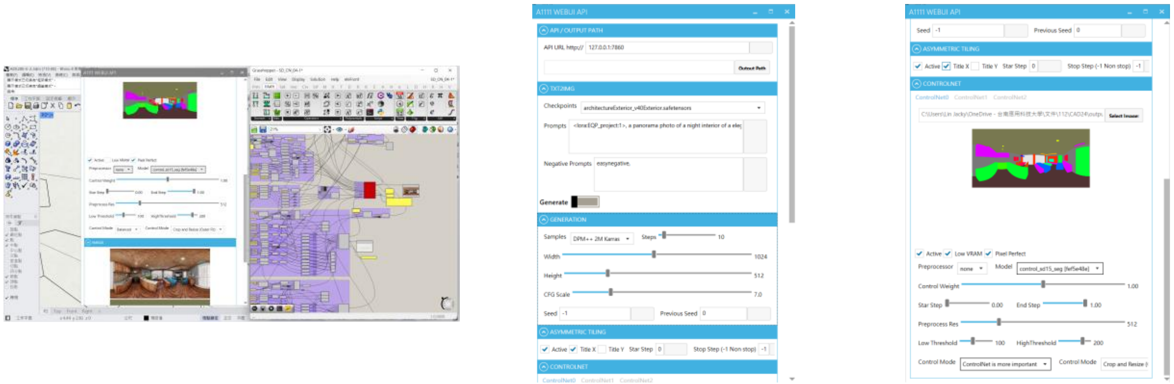

Figure 6: The interface for generating panorama by Stable Diffusion and ControlNet: (a) the GhPython and HumanUI components in Grasshopper, (b) the simplified interface of Stable Diffusion with Asymmetric Tiling extension, and (c) the simplified interface of ControlNet.

This phenomenon can only be discovered when viewing within a VR device, causing viewers to lose a sense of reality, which is important for the viewers' spatial experience. The Asymmetric Tiling extension [23] of SD-WebUI allows the generated image to avoid apparent seams in the X, Y, or both axis directions. Therefore, the interface developed in this study also integrates the operation of Asymmetric Tiling, and its interface is located between Stable Diffusion and ControlNet (Fig. 6b).

Although the Stability Matrix app can facilitate the management of checkpoint, ControlNet, and LoRA model files, currently installing extended functions such as ControlNet and Asymmetric tiling still requires the SD-WebUI to shut down or stop the API access function. To simplify the operation, the results of this research don't allow users to install additional extensions and models through the interface in this study by themselves. Users with sufficient knowledge and authority can still locally or remotely perform these operations through the standard SD-WebUI.

2.2.3 Default Text Prompts for the LoRA and Text Inversion Embedding

In addition to the complicated operation of ControlNet, additional models, such as LoRA, and Embedding installed beforehand, can be operated directly through text prompts. The original web interface of SD-WebUI provides the ability to display installed additional models as cards, allowing users to click on the cards' icon of the models to enter text prompts for triggering the model's functions. Because the simplified interface of this study has obscured these auxiliary functions, this study instead provides preset prompts to trigger the pre-installed LoRA and Text Inversion models. For example, for generating a panorama, this study pre-installed a LoRA model named LatentLabs360 [13] on the server. Its trigger prompt is "<lora:EQR_project:1>", which will be pre-entered in the positive prompt field of the interface developed in this research (Fig.7a).

Embedding model (as known as Text Inversion) is a method of fine-tuning the Stable Diffusion model [7]. Through training, the features represented by multiple tokens of images can be embedded into the original model by a new token that has not yet been learned in the original model. In addition to fine-tuning the original model to learn new objects or features, text inversion is often used for negative prompts to avoid long prompt inputs and improve the quality of the generated images. Therefore, this study pre-installed a commonly used model named EasyNegative [8] and pre-entered its trigger prompt "easynegative" into the negative prompt field (Fig.7a). With the help of LoRA, Text Inversion, and ControlNet, designers can use Grasshopper in Rhino to generate panoramas that conform to low LOD models and their design concepts described in text prompts (Fig. 7).

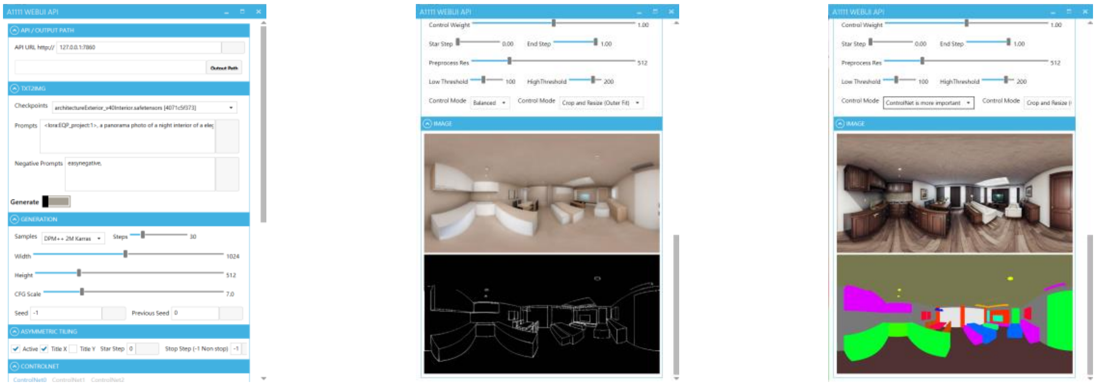

Figure 7: The AI-generated panorama by Stable Diffusion, ControlNet, and LoRA: (a) Basic parameters with default prompts of Stable Diffusion (Left), (b) A generated result by the canny model of ControlNet (Middle), and (c) A generated result by the semantic segmentation model of ControlNet (Right).

2.3 Display and Sharing the Generated Panoramas via Facebook

As mentioned above, the panorama-sharing feature of Facebook is one of the easiest and fastest ways to present and share visual effects of VR on smartphones, tablets, and other VR devices (Fig. 8). Not limited to expensive and bulky VR devices, sharing on Facebook should be more suitable for discussions and remote communication in design studios, and you can still use VR devices to watch for a better spatial experience. However, if the AI-generated panorama in PNG format is uploaded directly to the Facebook website, it will not produce a panoramic visual effect. Since the Facebook panorama sharing function was originally used for sharing panoramas taken by a panoramic camera, the AI-generated panorama needs to forge its camera source, that is, the 'Make' and 'Model' information of the EXIF metadata, such as "Ricoh" and "Ricoh THETA S," before uploading to Facebook. Secondly, due to the functional limitations of the Facebook, PNG files cannot be recognized as panoramas through a personal computer (PC), they must be uploaded through the Facebook app on a smartphone to be recognized as a panorama.

However, all AI-generated panorama in this study were completed in Rhino with Grasshopper on a PC. Uploading to Facebook through a smartphone will affect the efficiency of sharing results. Since the PNG format has many restrictions on the processing of EXIF information, and a JPG file will be correctly recognized as a panorama whether it is uploaded from a smartphone or a PC. This study therefore uses GhPython to convert PNG panorama into JPG format and add the required EXIF information at the same time. This JPG file can be uploaded to Facebook through a PC or a smartphone, and keep the panoramic VR visual effect on Facebook. However, converting PNG into JPG format will inevitably lose the AI-generating parameters stored in the meta-information of the original PNG image, such as checkpoints, prompts, samplers, and other parameters used during AI generating.

Compared with VR scenes created by higher LOD models, Facebook's panoramic photo album has lower hardware requirements. It can quickly switch among different panoramas (Fig. 8b,c), making it easier for designers or teachers to communicate and discuss with clients or students. For traditional VR production methods, creating multiple VR scenes is time-consuming and labor-intensive, and switching between VR scenes often requires considerable time for files to be loaded. In addition, by applying a low LOD model with the semantic segmentation model of the ControlNet, multiple realistic panoramas with different ideas can be quickly generated (Fig. 8a), which is difficult for traditional VR technology to achieve, and cannot be imagined before GenAI emerging.

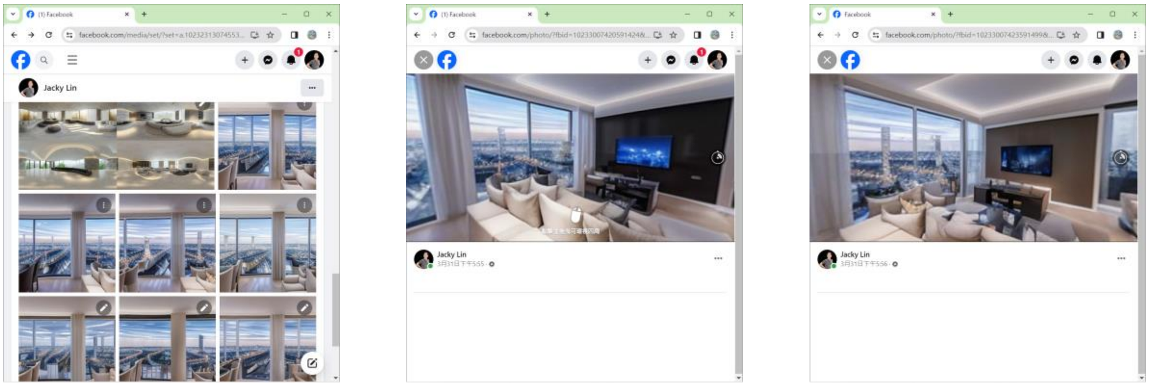

Figure 8: Sharing panoramas on Facebook: (a) Serial panoramas album (Left), (b) operating a panorama on the PC by mouse (Middle), and (c) quickly switching to another panorama (Right).

3 DISCUSSIONS

The results of AI-generated images from text are affected by many factors. The main factors are (1) the basic model or the checkpoints, (2) text prompts, (3) generating parameters, and (4) additional control models. The first is that the basic model or the checkpoint is the key to the content and quality of the generated image. For example, many professional users recognize the aesthetic quality and photo-realism of Midjourney. The generation quality and functions of different versions of Midjourney are also different. For architecture and interior design, checkpoints, which are fine-tuning models based on Stable Diffusion by developers on the CivitAI, often generate better-quality images than the original Stable Diffusion model. For different purposes, there are many checkpoints on CivitAI that are fine-tuning to create specific styles or characters, as well as specific architectural styles or building types. How to choose appropriate checkpoints still needs more investigation. The examples in this article mainly use the v4.0 Interior model of "architecture_Exterior_SDlife_Chiasedamme [10]" on CivitAI. The same author's Exterior model usually generates the indoor ceiling as an outdoor sky. This study does not discuss the selection of checkpoints more because, for interior design, the impact on quality is less significant than other factors such as materials, colors, styles, and lighting.

Secondly, the language model used by the basic model to process text will affect how the model processes prompts for generation and, therefore, determines the contents of the generated images. DALL-E, based on the ChatGPT [18], is considered to have a better understanding of complex prompts than Stable Diffusion based on the CLIP [19]. In fact, most rapid development of GhPython components in this study benefited from the Python code generation capabilities of ChatGPT4. For different basic models, the text prompts can also embed the operating instructions. For example, the LoRA and Text Inversion syntax demonstrated previously only apply to Stable Dissusion and cannot be directly applied to DALL-E or Midjourney. Writing text prompts also requires many skills, collectively called "prompt engineering," for controlling the generating results. This research has yet to be particularly in-depth in the part of prompt engineering. For creative ideas in the early stages of design, in addition to finding the appropriate and correct prompts, the professional terminology of interior design may not be embedded in the checkpoints. How to solve this problem, such as training Embedding or LoRA for professional purposes, requires more investigation.

Then, the basic generation parameters of Stable Diffusion, such as the sampler, sampling steps, resolution, and CFG Scale, are critical factors affecting the final results. This paper is not discussed in depth due to page limitations. For example, when viewing panoramas on VR devices, insufficient resolution is a common complaint during testing. Even without considering the insufficient problem of GPU's VRAM, increasing the generation resolution may not produce usable results. Limited by the original training resolution of Stable Diffusion 1.5, only 512x512 pixels, excessively increasing the generating resolution may sometimes lead to unpredictable results, such as split screen images. Because the ADE20k segmentation model can now only apply to the checkpoints of Stable Diffusion 1.5, it cannot be used for the XL version with a higher training resolution of 1024x1024 pixels. Applying the built-in high-resolution-fix or Upscale functions of SD-WebUI can improve the final generated resolution, but the simplified interface we developed requires modification for adding more parameters.

Finally, additional network models, such as LoRA and ControlNet, are attached to fine-tuning the basic model, which can control the generated results. Since Midjourney and DALL-E are closed private models, the relevant functions either need to be added or directly embedded in the system. They must be operated through text prompts as instructions, and users cannot inject extra control models during the image generation as they like. Due to the open source policy, the Stable Diffusion has produced many available resources, which is why it was chosen for this study. For interior design, applying appropriate control to generate images that align with designers' concepts while maintaining their creativity and flexibility is more important than generating realistic or unrestrained contents.

4 CONCLUSIONS

The confluence of VR and GenAI technologies has begun to reshape the landscape of design education and practice, heralding a new era where the rapid ideation and conceptualization of interior design are not just possible but increasingly accessible. As hardware and software costs continue to decline, and computational efficiency sees significant improvements, VR emerges from the periphery to become a central figure in the design process. Unlike traditional methods reliant on 2D drawings and static 3D renderings, VR offers an immersive spatial experience that closely mirrors the eventual built environment. This advancement holds particular promise for interior design, where the spatial experiences play a critical role in the overall design outcome.

With the launch of cheaper all-in-one VR devices such as Meta Quest 2 and 3, sharing VR content will become easier. The next difficulty will be how to produce VR content quickly. Compared with traditional VR production, which has cumbersome processes and the manipulation is limited by high-end 3D computing equipment, this research uses quickly AI-generated panoramas, which can sufficiently display spatial effects on relatively cheap VR devices and smartphones to reduce hardware requirements for sharing VR contents. In addition, compared with traditional VR production, which can only operate a scene at a time and can make partial modifications, sharing multiple panoramas at once through Facebook and quickly switching among them is more suitable for creative ideations and discussions further to inspire new ideas, or then to confirm design concepts. By combining Stable Diffusion, ControlNet, and the panoramic LoRA with Grasshopper's parametric modeling capabilities, this research demonstrates a method to rapidly generate VR scenes that align with existing conceptual interior spaces and design ideas by text prompts. However, AI-generated panoramas have the potential to reversely generate further 3D models, such as back-projecting the panorama to a low-LOD model to improve their quality or using ControlNet to obtain more image features such as depth and normal maps and then to generate higher LOD models based on those features. These applications are still under development due to time constraints. It is scheduled to continue to explore more applications of AI-generated images combined with VR and generative models in interior design in future research.

ACKNOWLEDGEMENTS

The National Science and Technology Council of Taiwan supported this paper under the grant number NSTC 112-2221-E-165-001-MY3.

REFERENCES

[1] Ambrosini, L.: Ambrosinus Toolkit, https://github.com/lucianoambrosini/Ambrosinus-Toolkit, Luciano Ambrosini.
[2] AUTOMATIC1111: Stable Diffusion Webui, https://github.com/AUTOMATIC1111/stable-diffusion-webui, github.com.
[3] CivitAI: The Home of Open-Source Generative AI, https://civitai.com/, CivitAI.
[4] Dortheimer, J.; Schubert, G.; Dalach, A.; Brenner, L.; Martelaro, N.: Think AI-side the Box! Exploring the Usability of Text-to-Image Generators for Architecture Students, in: eCAADe 2023, 2023, 567-576. https://doi.org/10.52842/conf.ecaade.2023.2.567
[5] EvolveLab: Veras, https://www.evolvelab.io/veras, EvolveLab.
[6] Faraj Al-Suwaidi, M.; Agkathidis, A.; Haidar, A.; Lombardi, D.: Immersive Technologies in Architectural Education: A pedagogical framework for integrating virtual reality as the main design tool in a fully virtualised architectural design studio environment, in: eCAADe 2023, 2023. https://doi.org/10.52842/conf.ecaade.2023.1.089
[7] Gal, R.; Alaluf, Y.; Atzmon, Y.; Patashnik, O.; Bermano, A.H.; Chechik, G.; Cohen-Or, D.: An Image is Worth One Word: Personalizing Text-to-Image Generation using Textual Inversion, in, 2022, arXiv:2208.01618. 10.48550/arXiv.2208.01618
[8] gdgsfsfs: EasyNegative, https://huggingface.co/datasets/gsdf/EasyNegative, Hugging Face.
[9] Graphisoft: Archicad AI Visualizer, https://graphisoft.com/solutions/innovation/archicad-ai-visualizer, Graphisoft.
[10] Hoang_StablediffusionLife: architecture_Exterior_SDlife_Chiasedamme, https://civitai.com/models/114612?modelVersionId=124044, CivitAI.
[11] Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.: LoRA: Low-Rank Adaptation of Large Language Models, in, 2021, arXiv:2106.09685. https://doi.org/10.48550/arXiv.2106.09685
[12] HuggineFace: The AI community building the future, https://huggingface.co/, Huggine Face.
[13] LatentLabs: LatentLabs360, https://civitai.com/models/10753, CivitAI.
[14] Lin, C.-J.: Plan1-2-3: Developing LOD-Based Parametric Floor Plan of Interior Design for Virtual Design Studio, in: CAD'23, 2023, 248-253. https://doi.org/10.14733/cadconfP.2023.248-253
[15] Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C.L.; Dollár, P.: Microsoft COCO: Common Objects in Context, in, 2014, arXiv:1405.0312. 10.48550/arXiv.1405.0312
[16] lllyasviel: Fooocus, https://github.com/lllyasviel/Fooocus, github.com.
[17] LoykosAI: StabilityMatrix, https://github.com/LykosAI/StabilityMatrix, Loykos AI.
[18] OpenAI: ChatGPT — Release Notes, https://help.openai.com/en/articles/6825453-chatgpt-release-notes, OpenAI.
[19] Rombach, R.; Blattmann, n.; Lorenz, D.; Ommer, P.E.B.: High-Resolution Image Synthesis with Latent Diffusion Models, in: the IEEE/CVF conference on computer vision and pattern recognition, 2022, 10684-10695.
[20] Stability.AI: Image Models, https://stability.ai/stable-image, Stability.AI.
[21] Stigsen, M.B.; Moisi, A.; Rasoulzadeh, S.; Schinegger, K.; Rutzinger, S.: AI Diffusion as Design Vocabulary, in: eCAADe 2023, 2023, 587-596.
[22] sunset1995: py360convert, https://github.com/sunset1995/py360convert, github.com.
[23] tjm35: Asymmetric Tiling for stable-diffusion-webui, https://github.com/tjm35/asymmetric-tiling-sd-webui, github.com.
[24] Trimble: SketchUp Diffusion, https://help.sketchup.com/en/sketchup-diffusion, Trimble Inc.
[25] Zhang, L.; Rao, A.; Agrawala, M.: Adding Conditional Control to Text-to-Image Diffusion Models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023, 3836-3847. https://doi.org/10.1109/ICCV51070.2023.00355
[26] Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; Torralba, A.: Semantic Understanding of Scenes through ADE20K Dataset, International Journal on Computer Vision, 127, 2019. https://doi.org/10.1007/s11263-018-1140-0