Title
Contextual Spatial Computing: A Generative Approach

Author
Keshavarzi, Mohammad

Publication Date
2022

Peer reviewed|Thesis/dissertation

Contextual Spatial Computing: A Generative Approach

by

Mohammad Keshavarzi

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Architecture

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Luisa Caldas, Chair
Associate Professor Simon Schleicher
Associate Professor James O'Brien
Dr. Allen Y. Yang

Spring 2022

Contextual Spatial Computing: A Generative Approach

Abstract

Contextual Spatial Computing: A Generative Approach

by

Mohammad Keshavarzi

Doctor of Philosophy in Architecture

University of California, Berkeley

Professor Luisa Caldas, Chair

Spatial Computing interfaces such as augmented reality (AR), virtual reality (VR), and mixed reality (MR) have become promising modalities for next-generation computing platforms. Along with its potential impact on various technological applications, spatial computing comes with spatial limitations itself. Such experiences are physically constrained by the geometry and semantics of the local user's environment where existing physical elements are present. Unlike 2D screens, where a rectangular screen region can host digital content with possible overlay, 3D environments are occupied with diverse physical obstacles and functional constraints. This results in complex and, many times, non-convex activity spaces available for virtual content augmentation. Target environments are not necessarily known to content developers, and hence the ability to deploy large-scale curated experiences that can adapt to a diverse set of user spaces is challenging. This limitation is elevated in remote telepresence scenarios, where identifying a common ground physically accessible for all participants can become difficult, especially if users are unaware of the spatial layout of other participants' physical environments.

Motivated by these spatial challenges, this dissertation works towards developing context-aware generative frameworks which enable large-scale deployment of adaptable spatial computing experiences for everyday users in diverse target environments. By introducing novel workflows to learn from examples as priors and utilizing spatial optimization methods, the systems developed in this dissertation address the spatial challenges in spatial computing in various applications of remote workplaces, multi-user telepresence, and curated experiences such as games and education. The contributions of this dissertation consist of solving two general sets of problems: 1) developing curated context-aware spatial experiences for large-scale deployment, which include a wide variety of diverse target spaces not known to the content developer; and 2) facilitating telepresence experiences, where participants are not aware of each other's local spaces due to the Telepresence Spatial Mapping Problem (TSPM), explained in this dissertation. The frameworks developed in this work can play a role in increasing the adoption of spatial computing interfaces in everyday environments, allowing developers to design and curate content in scale without knowing the target scene of the

user itself. Moreover, the frameworks proposed here can potentially facilitate remote workplace practices and virtual collaborations by decreasing the spatial requirements for telepresence systems.

To my dear Family ...

Contents

List of Figures

List of Tables

List of Acronyms

Acknowledgments

First, I want to thank God. For the privilege of being healthy, happy, and surrounded by the most wonderful of people. During these years far from home, I had the blessing to complete my education in one of the most exciting places in the world while experiencing beautiful moments of peace, self-awareness, and rejoice.

I would like to extend my greatest gratitude to my principal research advisor, Luisa Caldas. Luisa has been a wonderful teacher, mentor, and, most importantly, a role model for me during the last six years. Her tireless personality, passion for design, and expertise in AI research were all elements of inspiration, guidance, and motivation during my time at Berkeley. She believed in me when I did not believe in myself and taught me how to push toward new experiences outside of my comfort zone. Luisa has always provided me with the freedom to explore various directions to find the ones that interest me the most while guiding me in every project to ensure that I make the most out of it. Thank you, Luisa, for the patience and effort you generously devoted to my growth during my graduate studies.

I want to express my deepest appreciation to Allen Yang, who paved my way to explore the fascinating topics of computer vision and graphics while facilitating the integration of spatial computing technologies within my research. Allen's insight allowed me to envision how my research can fit into the larger industry context and helped me identify research problems that can potentially impact spatial computing products in the near future. He patiently provided feedback on every detail of my work, and this dissertation would look very different without his mentoring. With his efforts to push AR/VR initiatives forward on campus, Allen created the opportunity for me to collaborate with great talents and explore interdisciplinary research in a way I could not have imagined.

I am also very grateful to other members of my Ph.D. committee, to Simon Schleicher for his invaluable mentorship, friendly coffee chats, and caring advice on how to make the best out of my Ph.D. experience. I want to thank James O'Brien for his unique feedback and support on my research and for preparing me for my next career decisions. I am also extremely thankful to Avideh Zakhor for introducing me to the wonderful world of 3D recognition and reconstruction and to Nicholas de Monchaux for providing valuable insights into my research for the larger theoretical field of AR/VR and architecture.

I want to thank my senior Ph.D. colleagues in our BSTS cohort Luis Santos, Carlos Duarte, Won Hee Ko, and Haripriya Sathyanarayanan. You guys helped me navigate the Ph.D. program with friendly advice and helpful support. I am grateful to Joe Menke and Oladapo Afolabi in the EECS Department for their collaboration in my research and all the exciting kitchenette chats we had about the future of spatial computing. I would also like to thank all my other co-authors and collaborators: Flaviano Christian Reyes, Ritika Shrivastava, Aakash Parikh, Xiyu Zhai, Woojin Ko, and Melody Mao, all super-talented individuals who helped me develop some of the work that is presented in this dissertation.

During my Ph.D., I was lucky to have the opportunity to engage in industry collaborations with Autodesk Research and Meta Reality Labs. Special thanks to Michael Bergin for being an amazing mentor during his time at Autodesk Research and later when leading his game-changing

startup Higharc. I also thank Mehdi Nourbakhsh, Hans Keller, and Mohammad Asl, who generously guided me during my time at Autodesk by sharing their expertise and experience in the AEC technological landscape. Moreover, I am also grateful to Michael Zollhoefer, Henrik Skovsgaard, and Britta Hummel at Meta Reality Labs for paving the ground for meaningful research internships and guiding me on how to bridge research toward building products in a fast-paced environment.

I want to acknowledge my dearest friends at Berkeley: Reza Abbasi-Asl, Maryam Shadmehr, Ali Yadolalhi, Ameneh Gholipour, Negar Mehr, Payam Delgosha, Vahid Liaghat, and Fereshteh Radai. You guys made our life at Berkeley meaningful, fulfilled with joy, and delightful. You were the best of listeners, the coolest to travel with, and the perfect gang to translate the roughness of student life to pure happiness. My greatest gratitude to Mohammad Soheilypour and Mohasedeh Peyro for being like a caring older brother and sister to us, and to the Meshkat community for making us feel part of a large supportive family. I also want to acknowledge my long-time KSG buddies Mohammad Honarbaksh and Reza Ghoddoosian, who always had my back and wanted the best for me, even as we were thousands of miles apart.

I want to thank my family in Iran. To my parents, Zahra Javid and Parviz Keshavarzi, you taught me the value of learning, not giving up, and to be fearless for dreaming big. You sacrificed a lot and have always put my happiness before yours. I am grateful to my abji Fatemeh, who I know is there when I need her and takes care of everything with an open heart. My deepest appreciation to my mother and father-in-law, Moneer Seyyedhaddadi and Mahmoudreza Yavaribajestani, for being a constant source of unconditional support and inspiration. To Mozhgan and Maryam Yavaribajestani, Mehdi Zangi and Amir Monsefinejad for making me feel part of a new family. I also want to acknowledge my dear Uncle Sirus for being a role model of patience and sacrifice, while devoting this dissertation to my late Grandmother, as she devoted her life to steering our family through crises and challenges to becoming what we are today.

In the end, I want to thank my beloved wife, Yasaman. I have nothing to say. With all the ups and downs, hardships and challenges, you have made life look simple, beautiful, and enjoyable. Thank you for believing in me and for endless love, sacrifice, and support. This milestone would have never happened without you. Let's continue together on this wonderful journey of ours!

Chapter 1

Introduction

The re-emergence of Augmented Reality (AR), Virtual Reality (VR) and Mixed Reality (MR) technology in the past decade as the next-generation human-computer interface has followed a strong industry push for manufacturing affordable consumer-based hardware. Advances in optics, displays, and processing technologies have allowed various disciplines to explore how spatial interactions with virtual objects can benefit their fields. Many definitions have been introduced to describe the broad terms of VR, AR, and MR [47, 156, 196]. Yet, all the above-mentioned terms share a main common attribute: they use space as a medium to interact with virtual technology. To encapsulate this spatial attribute, in the context of this dissertation we use a more recent term that is also used in the academic and industrial community: *Spatial Computing (SC)*. We define SC as a technology that incorporates virtual objects and environments within a spatial context. Therefore, AR, VR, and MR can be all classified as SC technology.

While SC interfaces are forming an expanding market in various applications, they have shown potential success to promote remote workplaces, virtual collaboration [11, 65], immersive healthcare [152, 190, 42] and professional training [248, 126, 20, 46] and design [23, 104, 168, 136]. The deployment of such applications can also reduce carbon footprints by cutting transportation emissions, time, and costs and increasing productivity by taking advantage of the enhanced user interaction modules provided in such interfaces. Telepresence meetings allow participants to remotely join a mutual space [249, 12, 134, 232], while sharing documents or 3D content with one another [135, 157]. Design reviews within immersive environments provide the opportunity to inspect and analyze objects, or experience walk-throughs inside virtual buildings that have not yet been built [52, 25, 23]. In the gaming industry, SC can provide more activity-based leisure incorporated with rich spatial content.

With all the promising future, SC has many hurdles to overcome before being able to serve as a widely accessible computing platform for all. The goal of this dissertation is to shed light on the inherent spatial limitations for next-generation SC platforms and discuss potential solutions by introducing novel generative systems for various SC applications. With the goal of allowing SC platforms to be accessible to a wide variety of users holding diverse cultural and socioeconomic backgrounds, this dissertation starts by identifying the spatial problem in SC and discussing why the large-scale deployment of SC experiences can be challenging with conventional methods.

After highlighting the challenges, we further elaborate on our efforts on how we utilize generative approaches to address the spatial limitation of next-generation SC systems.

1.1 Spatial Computing and Spatial Limitations

SC comes with inherent spatial limitations. Such experiences are physically constrained by the geometry and semantics of the local user environment where existing building elements and furniture may be present [151, 170]. Unlike 2D screens, where a universal rectangular region hosts digital content, 3D environments are often occupied by physical obstacles that serve various functions. The same 3D environment are also used to support various activity functions (such as walkable, sittable, collaborative). This results in diverse and non-convex, usable spaces with various activity functions. In a VR setting, when a user wears an Head-Mounted Display (HMD), the virtual environment is visually detached from its surrounding physical environment and, therefore, can be visually embraced in an infinite space of virtual content. Content can be rendered, augmented, and placed anywhere necessary in the virtual space. However, the user's physical boundaries limit the user to freely walk around as it might physically collide with real-world objects of its local environment. This could prompt possible conflicts between the virtual experience and the physical space. For AR, experiences automatically become less immersive once the user encounters virtual objects placed unrealistically in their environment, conflicting with other existing elements or not holding common physical relationships with the real-world scene. Therefore, one can assess that content placement in SC experiences is highly dependent on the end user's target scene.

Acquiring an accessible activity space is a prerequisite for SC experiences. For many six-degrees-of-freedom SC applications using HMD 's, the user will often be asked to manually initiate a block of free space where user activity for the SC experience can be assumed to be safe. Inferencing the above contextual information can be readily done using several well-established 3D modeling algorithms in computer vision. Current AR devices, such as the HoloLens or MagicLeap, integrate such algorithms to estimate the layout of the space, including floors, walls, and ceilings, and typical furniture objects such as tables and chairs. One can take advantage of the semantic segmentation methods widely investigated in computer vision literature [165, 121, 3] to segment their spatial boundaries and obtain their geometric properties, such as dimensions, position and orientation, object classification, and activity functions. Given such contextual information of individual spaces to be available via either a manual or algorithmic process, accessible spaces can be identified by subtracting the occupied space from the whole room. These spaces, which are different for every user based on their local environment, can serve as areas where virtual content can be augmented in AR and MR, and where users can safely access and interact with other virtual entities within VR environments.

1.2 Large-scale Deployment of Spatial Computing Experiences

A major challenge in the design and development of SC experiences is that the local scene and the inherent spatial limitations of the end-user are not necessarily known to the content developer. This challenge becomes a bottleneck when aiming to develop spatial experiences for a large number of users. End users hold diverse spatial environments, which differ in dimensions, functions (rooms, workplace, garden, etc.), and available activity spaces. For each target space, boundaries, openings, surfaces, and existing furniture and their arrangements are often unknown to the developer, making it challenging to design a virtual experience that would adapt to all end-users' physical environments while avoiding conflicts between the virtual experience and local physical boundaries.

For spatial experiences in AR and MR, which involve virtual object augmentation within the physical space, context plays a key role as the location of the instantiated object becomes critical for the user experience. If the user encounters virtual objects placed unrealistically in their environment, conflicting with other existing elements or not holding common physical relationships with the real-world scene, such experience automatically becomes less immersive and engaging. To avoid this issue, current AR applications address this challenge by asking users themselves to identify the usable spaces in their surrounding environment or manually positioning the augmented object within the scene. In such cases, the holistic spatial placement of the virtual elements would no longer be curated by an expert designer but rather by the end-users themselves. Therefore, virtual object placement in most AR experiences is limited to specific surfaces and locations, e.g. placing objects naively in front of the user with no scene understanding or only using basic surface detection. These simple strategies can work to some extent for small virtual objects, but the methods break down for larger objects, which may not simply sit on top of the nearest surface. In such a context, the spatial design of the virtual elements would no longer be curated by an expert designer but would rely on the placement strategies of the user itself.

A similar challenge is also present in 2D display experiences, where developers tend to develop the same content for displays of various sizes (desktop, tablets, phones). In this regard, responsive design methods [138] provides the ability to design the content in relation to the display margins and digital elements themselves. Content in such methods automatically re-scale and re-arrange to adapt to the end-users display boundaries. However, in 3D environments where usable spaces are not always rectangular, additional complexities are introduced for adapting spatial content to target scenes. Yet, recent research in the field of computer vision has allowed systems to not only reconstruct and map the surrounding environment relative to the user but also semantically segment the reconstructed geometrical data into objects categories. While this would allow end-user interfaces to understand the surrounding environment, the process of how to efficiently position and integrate virtual content with the surrounding world is still an open challenge.

1.3 The Telepresence Spatial Mapping Problem (TSMP)

With the rapidly growing demand for remote communication platforms in workplaces, households, and education institutes, more forms of effective communication technologies have emerged in the past two decades. More recently, advances in consumer-grade SC and standalone headsets and displays have introduced alternative systems of immersive and context-aware communication platforms known as telepresence. Telepresence allows changing the state of one's sense of presence from a physical location to a target remote environment without requiring the physical body to relocate to the target environment [55, 56].

However, as all parties of the telepresence settings hold spatial limitations (room size, furniture settings, etc.), their virtual doubles or avatars may not be able to hold the same spatial relationships they have within their real-world spaces. This challenge is what is referred to in this dissertation as the Telepresence Spatial Mapping Problem (TSMP). The TSMP is a mapping problem between virtual remote avatars and the local physical environment in a spatial computing telepresence experience. TSMP occurs when remote virtual avatars cause geometrical or contextual conflicts with the local physical environment or other participant avatars. TSMP can potentially result in misalignment of head and body gestures, spatial sound errors, and other micro expression inaccuracies due to the incorrect positioning of each member of the virtual call.

This section briefly elaborates on the TSMP and how surrounding spatial limitations in telepresence participants can cause immersion-breaking in multi-user interaction scenarios due to TSMP. We discuss how the immersion-breaking can happen in two aspects a) geometrical conflicts and b) line of sight. To simplify our explanations, we use an example of three remote users wanting to meet each other in a full-body avatar telepresence scenario. As illustrated in Figure 1.1, each of the participants reside in a room with a different size and functions. User A resides in a large living room, User B resides in a small office space, and User C resides in a bedroom. In the illustration, only the participant residing in the room is considered physically present. In contrast, the other two are considered virtual avatars, remotely joining from their own physical spaces.

Geometrical Conflicts

In telepresence settings where full-body realistic avatars are rendered with physically correct occlusions with the environment, users may experience geometrical conflicts between rendered virtual remote participants and their own local physical spaces. Such geometrical conflicts can result from the different room sizes and open space layouts in various participants' local environments. For instance, in our example in Figure 1.1, participants can virtually meet each other in Users A's space holding a socially acceptable distance. However, in contrast, due to the smaller size of room B compared to rooms A and C, participants A and C are not seen in the telepresence experience of User B. While room C comes with a slightly larger open space compared to room B and hence can see other participants within its own bedroom space, the remote participants are rendered in inappropriate locations - User B is rendered in the middle of the bed, and User A is rendered colliding with the door and wall - conflicting with the context of room C's environment. Note that remote users may not be visually aware of the location of their virtual avatars in other spaces, and

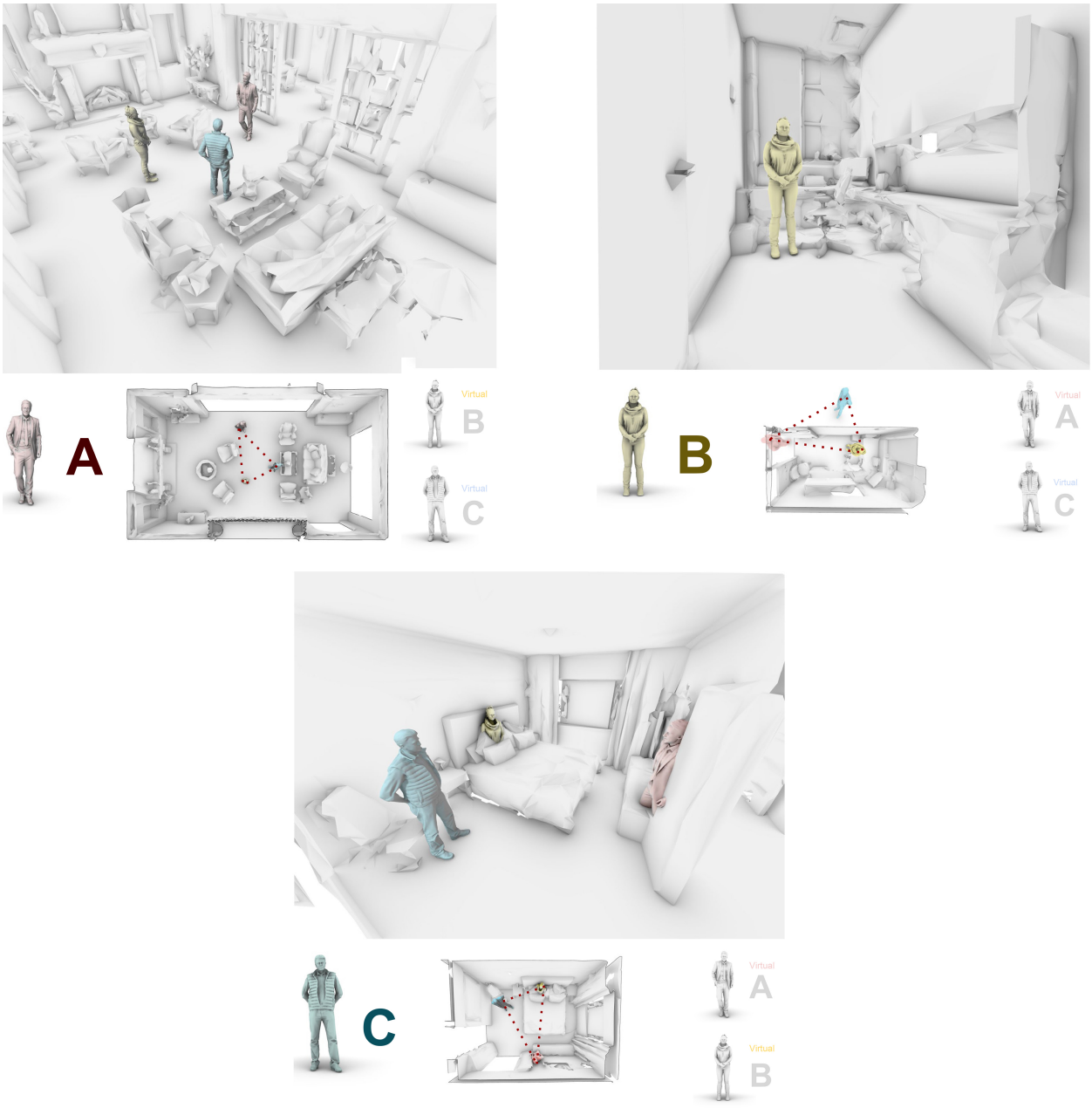
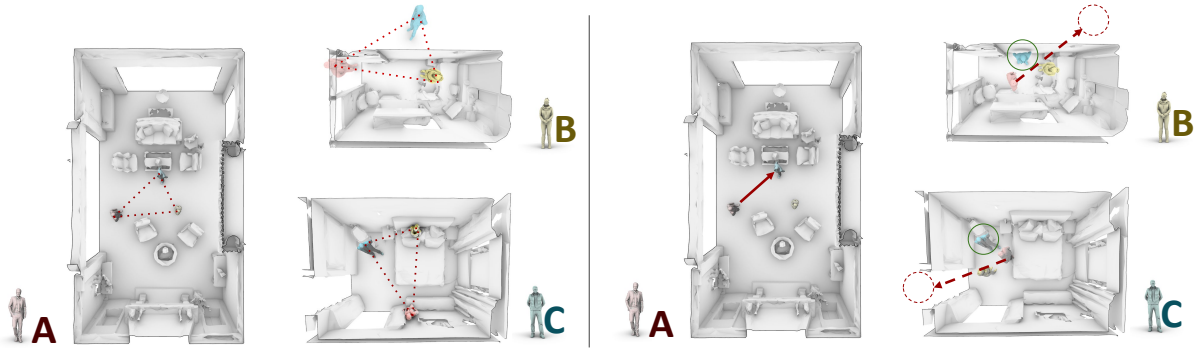


Figure 1.1: Illustration of an example of the TSMP. Three users in different spaces cannot hold the same positional and orientational relationships with each other due to geometrical and contextual conflicts within the room



(a) Constant positioning (b) Force positioning

Figure 1.2: Due to the TSMP, force mapping and manual re-positioning cannot be exercised in AR or MR scenarios since line of sight would be deconstructed.

therefore adjusting to multiple target environments while addressing their local physical boundaries can become challenging.

Line of Sight

To avoid geometrical conflicts, one may argue that allowing the user of each space to force-position other remote participants in a plausible custom location may be a solution to the problem. However, if participants were to have the option to force position other remote participants within their local space, a conflict of the line of sight would potentially occur. As seen in Figure 1.2.a, if participants of each room were to choose a custom arrangement of the remote avatar placements in their own space, the relationship shape of participants would differ in each space, causing visual and social conflicts. For instance, in 1.2.b, if user A looks at User C in its own space, in User B's space, it would seem like User A is looking at the wall behind it instead of User C. In Users C's experience, User A's line of sight is directly towards User B, which should've been directed to itself.

Even if we maintain the shape of participants' relationship with each other and scale the distance in which participants are situated, immersion-breaking can still occur during natural locomotion. For example, if User A starts walking toward User B, it would have a longer length of walking distance to complete within its own experience before colliding with User B, compared to what User B experiences in its own space. Moreover, in this scaling scenario, users would experience different social distances. One user may have a close social distance from the other, being able to observe facial expressions, while the other may be meters away, limited in the social cues gained from the conversation. Hence, one can conclude that to maintain social telepresence and avoid immersion breaking, all participants of a remote telepresence scenario must keep the same arrangement in their local environments.

1.4 Generative Spatial Computing

Motivated by the spatial challenges highlighted in this chapter, this dissertation works towards developing context-aware generative frameworks which would allow adaptable SC experiences to be executed for a large scale of everyday users in diverse target environments. Generative systems have shown promising results in various applications in computer vision, computer graphics, and computational design. A central component of this research discusses novel spatial representation schemas to encapsulate contextual relationships between users, objects, and spatial boundaries. By effectively learning from these representations via large datasets of scene priors and utilizing spatial optimization methods, the systems developed in this dissertation address the spatial challenges in SC in various applications of contextual spatial computing such as multi-user telepresence and scalable virtual content curation via context-aware augmentation for diverse target spaces.

The contributions of this dissertation can be categorized in solving two general sets of problems: 1) developing curated context-aware spatial experiences for a large set of users, which include a wide variety of diverse target spaces not known to the content developer; and 2) facilitating telepresence experiences, where participants are not aware of each other's local spaces due to the TSMP. In both problems mentioned above, the unknown spatial layout of the target user space is considered the mutual challenge, and hence this challenge is considered as the core research problem of this dissertation.

The frameworks developed in this dissertation can play a role in increasing the adoption of SC interfaces in everyday environments. The research would allow developers to design and curate content in scale without knowing the target scene of the user itself. The content itself can adapt to the target scene while addressing context-aware topologies with local physical and virtual objects. Such an approach can pave the way for developing *Responsive Spatial Computing* frameworks. Similar to responsive web design, Responsive Spatial Computing frameworks can be a cost-effective alternative to hard-coded applications due to its ability to house all of the code in a single program.

In addition, the frameworks can facilitate remote workplace practices and virtual collaborations by decreasing the spatial requirements for telepresence systems. Instead of setting up large open spaces required for such workflows, the systems develop in this dissertation would allow users to join from their personal spaces, with minimum modifications to their surrounding environment. Physical and virtual re-arrangements would be optimized based on participants' local environments. In AR experiences, the topological relationship and line of sight between all participants would be maintained without any conflicts between remote users and local physical obstacles, while in VR, the system can generate a fully synthesized scene and recommend spatial modifications to provide the required interaction area between multiple users. Such an approach can promote remote collaborations, effectively decrease required resources for energy for transportation time and costs, and can also increase productivity in a large spectrum of services.

Finally, scene synthesis and spatial manipulation techniques introduced in this work can provide valid synthetic datasets for data-driven systems, especially for learning-based approaches explored in design computing and computer vision applications. In addition, using methods introduced in this thesis, augmentation of currently available 3D scenes and large real-world scanned datasets can be executed, generating a more extensive set of spatial data for researchers in the field.

1.5 Research Objectives

The research objectives of this dissertation are summarized as follows:

- Design and develop a context-aware scene augmentation system for SC applications that can generate and augment virtual content to an already existing scene while considering topological relationships and geometrical constraints of the target environment.
- Introduce a knowledge model to serve as a prior for the scene generation system. This model can be trained by extracting contextual and topological features from previously constructed scenes available through 3D datasets or manually defined by a context creator by providing a limited set of examples.
- Compare the developed scene augmentation system with state-of-the-art constrained scene synthesis techniques. Scene synthesis holds a similar goal in generating plausible scenes for a given target space. Moreover, given that such techniques require a large number of priors for training, develop alternative learning architectures that can achieve similar results with limited scene priors.
- Extend the proposed generative tool for multi-user interaction scenarios. Such an approach would initially require identifying available mutual ground between participants, and if mutual space is insufficient, the system would recommend an alternative arrangement of the surrounding furniture.
- Develop SC prototypes for the proposed frameworks and explore applications of the work in AR, VR, and MR scenarios. Furthermore, perform comparative user studies to evaluate the effectiveness of the proposed system from a user performance standpoint.

1.6 Dissertation Overview

Following this introductory chapter, Chapter 2 provides an overview of related topics covered in this dissertation. Starting with the general subject of SC and its applications, we start by discussing how such systems can be utilized for remote collaboration and telepresence. Moreover, Chapter 2 covers how 3D experiences can be generated via procedural modeling, generative design, and scene manipulation. Furthermore, we discuss more recent approaches of scene graph representations and example-based scene synthesis. This is followed by a review of spatial mapping techniques in spatial computing workflows, allowing virtual experiences to adapt to physical spaces. We further discuss on how the background work relates to our contributions separately in each chapter .

Chapter 3 further elaborates on the TSMP and introduces a Mutual Space Optimization system to identify optimal mutual virtual spaces for multi-user interaction settings. The proposed algorithm can effectively discover optimal shareable space for multi-user virtual interaction and facilitate remote spatial computing communication in various collaborative workflows. In addition, the framework recommends the movement of surrounding furniture objects that expand the size of the

mutual space with minimal physical effort. The work demonstrates the performance of our solution on real-world datasets and also presents an augmented reality prototype developed on HoloLens.

Chapter 4 expands the discussion of contextual scene generation for spatial computing experiences and further introduces SceneGen, a generative contextual scene augmentation framework that predicts virtual object placement within existing scenes. SceneGen takes a scene as input and outputs positional and orientational probability maps for placing virtual content. The research initially formulates a novel spatial Scene Graph representation, which encapsulates explicit topological properties between objects, object groups, and rooms. SceneGen utilizes kernel density estimation to build a multivariate conditional knowledge model trained using prior spatial Scene Graphs extracted from real-world 3D scanned data as priors. To further capture orientational properties, this research also includes developing a fast pose annotation tool to extend current real-world datasets with orientational labels. Furthermore, the chapter reports comparative and user experiments to demonstrate the performance of our system in various indoor scene augmentation scenarios. Finally, to demonstrate SceneGen in action, we present our developed AR application which can contextually augment objects in real-time.

Furthermore, Chapter 5 discusses how 3D datasets can be augmented and manipulated via GenScan, a generative system that populates synthetic 3D scan datasets in a parametric fashion. The system takes an existing captured 3D scan as an input and outputs alternative variations of the building layout, including walls, doors, and furniture with corresponding textures. GenScan is a fully automated system that can also be manually controlled by a user through an assigned user interface. The chapter covers how GenScan utilizes a combination of a hybrid deep neural network and a parametrizer module to extract and transform elements of a given 3D scan, followed by style transfer techniques to generate new textures for the generated scenes.

In an effort to improve SceneGen's constrained scene synthesis method, Chapter 6 addresses the challenge of state-of-the–art deep learning scene synthesis, which requires large datasets for training by introducing GSACNet, a contextual scene augmentation system that can be trained with limited scene priors. GSACNet utilizes a novel parametric data augmentation method combined with a Graph Attention and Siamese network architecture followed by an Autoencoder network to facilitate training with small datasets. The research shows the effectiveness of our proposed system by reporting ablation and comparative studies with alternative systems on the Matterport3D dataset. The results indicate that GSACNet's scene augmentation outperforms prior art in scene synthesis with limited scene priors available.

Finally, Chapter 7 extends our exploration of the TSMP, by providing a solution for VR and MR telepresence experiences. This chapter discusses a novel Mutual Scene Synthesis framework that takes the participants' spaces as input and generates a virtual synthetic scene that corresponds to the functional features of all participants' local spaces. The method combines a mutual function optimization module with a deep-learning conditional scene augmentation process to generate a scene mutually and physically accessible to all participants of a mixed reality telepresence scenario. The synthesized scene can hold mutual walkable, sittable and workable functions, all corresponding to physical objects in the users' real environments. The chapter covers experiments using the MatterPort3D dataset and comparative user studies to evaluate the effectiveness of the proposed MSS system.

In the end, Chapter 8 concludes the work presented in this dissertation by discussing various aspects of the key findings in addition to the challenges and limitations that were discovered during the development of this research.

1.7 Related Publication List

The following publications are partially or entirely included in this dissertation.

- Keshavarzi, Mohammad, Michael Zollhoefer, Allen Y. Yang, Patrik Peluse and Luisa Caldas. "Mutual Scene Synthesis for Mixed Reality Telepresence." arXiv preprint arXiv:2204.00161 (2022).
- Keshavarzi, Mohammad, Flaviano Christian Reyes, Ritika Shrivastava, Oladapo Afolabi, Luisa Caldas, and Allen Y. Yang. "Contextual Scene Augmentation and Synthesis via GSACNet." arXiv preprint arXiv:2103.15369 (2021).
- Keshavarzi, Mohammad, Oladapo Afolabi, Luisa Caldas, Allen Y. Yang, and Avideh Zakhor. "Genscan: A Generative Method for Populating Parametric 3d Scan Datasets." arXiv preprint arXiv:2012.03998 (2020).
- Keshavarzi, Mohammad, Aakash Parikh, Xiyu Zhai, Melody Mao, Luisa Caldas, and Allen Y. Yang. "SceneGen: Generative Contextual Scene Augmentation using Scene Graph Priors." arXiv preprint arXiv:2009.12395 (2020).
- Keshavarzi, Mohammad, Allen Y. Yang, Woojin Ko, and Luisa Caldas. "Optimization and Manipulation of Contextual Mutual Spaces for Multi-user Virtual and Augmented Reality Interaction." In 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), pp. 353-362. IEEE, 2020.

In addition, the list below covers the publications that I co-authored during my Ph.D. studies at UC Berkeley with relevance to the dissertation topic, but are not directly included in this dissertation.

- Keshavarzi, Mohammad, Luisa Caldas, and Luis Santos. "Radvr: A 6DOF Virtual Reality Daylighting Analysis Tool." Automation in Construction 125 (2021): 103623.
- Keshavarzi, Mohammad, Clayton Hotson, Chin-Yi Cheng, Mehdi Nourbakhsh, Michael Bergin, and Mohammad Rahmani Asl. "SketchOpt: Sketch-based Parametric Model Retrieval for Generative Design." In Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1-6. 2021.
- Keshavarzi, M. and Rahmani-Asl, M., 2021. GenFloor: Interactive Generative Space Layout System via Encoded Tree Graphs. Frontiers of Architectural Research, 10(4), pp.771-786.

- Keshavarzi, Mohammad, Ardavan Bidgoli, and Hans Kellner. "V-Dream: Immersive Exploration of Generative Design Solution Space." In International Conference on Human-Computer Interaction, pp. 477-494. Springer, Cham, 2020.
- Keshavarzi, Mohammad, Michael Wu, Michael N. Chin, Robert N. Chin, and Allen Y. Yang. "Affordance Analysis of Virtual and Augmented Reality Mediated Communication." arXiv preprint arXiv:1904.04723 (2019).
- Caldas, Luisa, and Mohammad Keshavarzi. "Design Immersion and Virtual Presence." Technology| Architecture+ Design 3, no. 2 (2019): 249-251.

Finally, the following list represents granted and published patents related to my work co-invented during my Ph.D. studies.

- Keshavarzi, Mohammad, Aakash Parikh, Luisa Caldas, and Allen Y. Yang. "Search and Recommendation Process for Identifying Useful Boundaries in Virtual Interaction Settings." International Patent Application WO2021263018A1, published December 30, 2021.
- Keshavarzi, Mohammad, Luisa Caldas. "Contextual Augmentation Using Scene Graphs." International Patent Application WO2021072301A1, published April 15, 2021.
- Bergin, Michael S., Mehdi Nourbakhsh, Mohammad Keshavarzi, and Chin-Yi Cheng. "Automated parametrization of floor-plan sketches for multi-objective building optimization tasks." U.S. Patent 10,937,211, issued March 2, 2021.

Chapter 2

Background

2.1 Spatial Computing

SC provides immersive visualization through enhanced levels of interaction with virtual objects. With its growing adoption through affordable hardware, new applications and experiences of SC are being launched daily across the categories of healthcare [152, 190, 42], design [23, 104, 88, 136], education [154, 110], and more. Studies suggest that immersive environments enabled through SC interfaces can potentially provide a better spatial understanding for users when compared to 2D or non-immersive 3D representations [186, 160], while enhancing collaboration and team engagement among stakeholders [10, 13, 57]. Such property allows the deployment of SC for exploring new spaces and evaluating and inspecting objects on various scales while providing a platform for training, communication, and collaboration [23].

SC for healthcare, also known as immersive medicine, has been widely explored for the training of the medical community, logistical planning, and rehabilitation of patients. Various studies attempt to evaluate how useful SC can be utilized to improve the learning of anatomy [152, 190, 42] including studies proposing that SC could replace the use of corpses in medical school [221]. Training surgical procedures and the process of medical skill transfer has also drawn major attention in immersive medicine [191, 61, 50, 85]. In this context, many studies attempt to review the effectiveness of VR-based training for surgery including meta-analyses and reviews [86, 248, 126], transfer of training [20, 46], and other specialized applications in the medical field [5, 81, 201].

Education is another area in which SC has shown promising applications. Work of [36] identified positive effects of AR technology on students' development of skills and knowledge, enhancement of learning experiences, and improvement of collaborative learning. The use of SC in education could improve the learning efficiency and provide a fun and engaging experience for students [51]. Like other computational mediums, it can change the abstract into a tangible object [236]. Another advantage of SC in education can be seen in providing a platform that supports physical exploration activity rather than just static observing. For example, if a class needs to learn about the planet's natural wonders, virtual visits can be conducted in VR. Work of [118, 24] are examples of such environments for virtual field trips.

In design, SC environments have been utilized to explore how SC can facilitate designers through assimilating their sense of scale, depth, and spatial awareness. Such platforms integrate the use of SC in various functions of building science research [104, 88] such as construction operations, personnel training, end-user surveys, performance simulations and building information modeling visualization. Clients, architects, and building owners use derived applications from game engines to navigate 1:1 scale BIM models, allowing a virtual walkthrough experience of future buildings [168]. For such use cases, the performance improvement of space navigation between VR HMDs and 2D desktop screens have been investigated in various studies, with some suggesting significant improvement in VR headsets [182, 177] while others indicate no significant difference [176, 178]. Architects and building engineers can also use immersive design tools to model various building elements in VR Computer-Aided Design (CAD) interfaces and apply property modifications to BIM files through such environments.

For building performance engineering and analysis, immersive environments enable the user to focus on performance-based analysis without getting too distracted to operate and navigate the simulation tool [75]. SC applications have been designed for finite element analysis of shell structures. Using stylus and data gloves as input devices, the user can create, modify mesh, and specify boundary conditions. For a simple geometry, real-time color-coded results are obtained by changing loads on the model [124]. Studies have used artificial neural networks (ANN) or approximation methods to achieve real-time interaction for the complex geometry and to simulate its impact via haptic gloves [71]. For energy simulations, [155] developed a VR simulation environment using bi-directional data exchange between Unity and Modelica/Dymola. [179] developed a workflow for managing building information and performance data in VR with equirectangular image labeling methods. For augmenting data on existing buildings, [136] developed a Human Building Interaction system that uses AR to visualize CFD simulations.

In the construction industry, immersive environments have been used to improve site preparations, on-site communication, and collaboration of team members, safety [44] and logistics [144]. For training of construction workers, virtual environments have shown to be highly effective in skill transfer, with studies showing similar performance results to training in real environments [226]. Moreover, virtual platforms are also used in the operation phase of buildings to interact and visualize data with Internet of Things (IoT) devices available in buildings, process improvement and also resource management [233, 43]. Studies show SC platforms can perform future occupant studies in the building design process by allowing pre-construction mock-ups. This would allow the evaluation of alternative design options in the building model in a timely and cost-efficient manner [133]. Studies conducting human experiments have shown users perform similarly in daily office activities (object identification, reading speed, and comprehension) within immersive virtual environments and benchmarked physical environments [73]. In the field of lighting, VR HMDs studies have been used to investigate the influence of façade patterns on the perceptual impressions and satisfaction of a simulated daylit space [26]. Moreover, artificial lighting studies have implemented immersive virtual environments to evaluate end-users lighting preferences of simulated virtual scenes with the controlling of the blinds and artificial lights in the virtual environment [73].

2.2 Collaborative Telepresence Systems

Telepresence allows changing the state of one's sense of presence from a physical location to a target remote environment without requiring the physical body to relocate to the target environment [55, 56]. This dissertation investigates how telepresence experiences can be enhanced and facilitated by integrating spatial mapping and scene synthesis workflows. A large body of previous work has explored how collaborative human-based telepresence can be achieved by capturing a region of each participant's body and space and projecting it to the target environment. Systems developed by [232, 65, 105, 11, 249, 12] are examples of such efforts where participants and a limited range of their surrounding spaces are continuously captured using a cluster of registered depth and color cameras. Recent work of [107] in Project Starline takes this approach one step closer towards a high fidelity co-presence experience. The bi-directional system is able to capture audiovisual cues such as stereopsis, motion parallax, and spatialized audio; while enabling high-resolution communication cues such as eye contact and body language.

However, window-based telepresence systems limit the participant's ability to access each other's spaces. Users are spatially disconnected from each other, and interaction occurs through an audiovisual window acting as a barrier. The importance of free-form user movement and the ability to preserve mobility-based communication features in the context of co-presence has been studied in the work of [9, 93, 128, 23]. Alternatively, research in room-based telepresence systems has gained major momentum in recent years, allowing bilateral telepresence between participants, where participants share a common virtual ground. Work of [157] allows a remote user to be captured and rendered into a local user's space via an AR HMD, providing the feeling that the remote user is present in the local user's space as well. Such an approach is also seen in [135, 56], where the remote and local users do not share the same room layout but are calibrated to provide the required mutual virtual ground between users. [208] enables mutual ground-sharing by capturing the local space of one of the participants and streaming the data to a limited number of remote users. Recent work of Codec Avatars [125, 131] implemented as a decoder network of a Variational AutoEncoder (VAE) demonstrates how high-fidelity animatable human head models can be captured and later rendered in real-time via spatial computing HMDs.

2.3 Generative Design

As discussed in the previous chapter, one of the goals of this dissertation is to explore how large-scale deployment of curated designed SC experiences can be achieved via generative workflows. Generative design is considered a paradigm shift in CAD. Unlike traditional CAD-based processes, where a single design solution was modeled using a set of computational tools, in generative design, designers specify high-level goals and constraints, and the system automatically generates large sets of solutions all corresponding to the defined design criteria. In addition to geometrical attributes, generative design systems can be integrated with performance evaluators [194, 159] and simulation engines [53] to quantitatively assess and optimize the generated solution landscape. With the availability of high-performance computing and cloud services, this process can be parallelized,

allowing faster generation, improving the performance evaluation, and generating larger solution landscapes [6]. Users of such systems are then responsible for choosing between plausible design candidates, which is often considered a complex task [218, 224]. These users should inspect the high-dimensional properties of each solution and assess their aesthetic qualities.

There are two general approaches recognizable among the current generative design workflows: (i) Convergence generative design and (ii) divergence generative design [140]. In convergence generative design, the search mechanism is implemented in a way to converge the solution space into a single solution or a set of limited solutions. However, depending on the level of clarity and accuracy of the goals, constraints, and fitness function, the optimization process may dismiss many potentially acceptable solutions, which could have been otherwise chosen by the designer. Also, automated optimization methods do not leverage human expertise and can only find solutions that are optimal with regard to an invariably defined problem space [188]. On the other hand, in divergent generative design, the whole solution space is generated, then the designers utilize sorting, clustering, and filtering tools to manually navigate and explore the solution space. Rather than looking at a limited set of solutions, designers have the chance to continually redefine their goals and constraints, allowing a more comprehensive control over the generative process. As divergent generative workflows often produce large numbers of solutions, organizing the solution space to effectively explore the data is considered a critical step in design explorations.

There is also a large body of literature focused on how to interpret generative design solutions and provide the user with appropriate workflows to modify and interact with the generated solution space [188, 72, 142]. [32] explored methods and tools for multivariate interactive data visualization of the generated designs and simulation results by enabling designers to not only focus on high-performing results but also examine suboptimal ones. [147] proposed a computational design exploration approach that takes advantage of an interactive evolutionary algorithm to integrate the designers' preferences within the solution search. Work of [91] extends the user interaction with the solution space from 2D input to 3D exploration by developing a virtual reality generative analysis framework for navigating large-scale solution spaces.

2.4 Scene Graphs and Graph Neural Networks

A central component of this research focuses on introducing novel spatial representation schemas via scene graphs to encapsulate contextual relationships between users, objects, and spatial boundaries. Scene graphs have been applied to various computational tasks in the past, including image retrieval [84], visual question answering [215], image caption generation [242], and more. The past research can be divided into two approaches: (1) separate stages of object detection and graph inference and (2) joint inference of object classes and graph relationships. Papers that followed the first approach often leverage existing object detection networks [172, 112, 247, 242, 35]. Similar to other scene understanding tasks, many methods also involved learning prior knowledge of common scene structures in order to apply them to new scenes, such as physical constraints from stability reasoning [241] or frequency priors represented as recurring scene motifs [247]. Most methods were benchmarked based on the Visual Genome dataset [103]. However, recent studies found this

dataset has an uneven distribution of examples across its data space. In response, researchers in [66], and [35] proposed new networks to draw from an external knowledge base and to utilize statistical correlations between objects and relationships, respectively. Our work focuses on the task of construction and utilization of the semantic scene graphs. Work of [225] utilized PointNet [164] and Graph Convolutional Networks to regress a scene graph from the point cloud of a scene. Similar to [225] and [247, 35], the scene graph representations introduced in this dissertation utilize statistical relationships and dataset priors. Yet, unlike these papers, we use an explicit graph representation instead of implicit representations.

Furthermore, semantic scene graphs have been broadly explored to improve the general task of scene understanding. On this topic, a progression of papers attempted to encapsulate human common-sense knowledge in various approaches such as physical constraints and statistical priors [199], physical constraints and stability reasoning [82], physics-based stability modeling [255], language priors [127], and statistical modeling with deep learning [49]. A similar approach was detailed in [96] for 3D reconstruction, taking advantage of the regularity and repetition of furniture arrangements in certain indoor spaces, e.g., office buildings. In [239], the authors proposed a technique that potentially could be well suited to AR applications, as it builds a 3D reconstruction of the scene through consecutive depth acquisitions, which could be taken incrementally as a user moves within their environment. Some recent work has addressed problems such as retrieving 3D layouts from 2D panoramic input [212, 102] or floorplan sketches [95], building scenes from 3D point clouds [162, 197], and 3D plane reconstruction from a single image [246, 122]. One can consult a recent overview of the topic in [123].

When utilizing scene graphs for various computational tasks, graph neural networks have gained immense popularity as a learning methodology for analyzing such graphs. Seminal work of [185] introduced graph neural networks and the idea of message passing or neighborhood aggregation. On a high level, message passing is an iterative update process used to find node representation by using the graph structure to pass information from neighbors of a target node to the target node itself. Originally, graph neural networks were used as a method to classify nodes within a graph. But over the years, graph neural networks have expanded to autoencoder graphs [97], generate graphs [244], solving link prediction problems [250], and segmenting 3D point clouds [137]. Furthermore, different methods of message passing have been developed, as well as their aggregation strategies. Inductive approaches [222], attention mechanisms [223], and gated recurrent units [113] are some of the more popular approaches. For scene synthesis similar to our scene graph approach, the work of [257] utilized a dense scene graph for passing neural messages to augment an input 3D indoor scene with new objects matching their surroundings.

2.5 Data Augmentation and Scene Manipulation

Data Augmentation refers to a class of techniques that aim to enhance the size and quality of a given training dataset. Such techniques can improve the generalization performance of deep neural networks while avoiding the problem of overfitting when trained on limited data. A wide variety of methods such as geometric transformations, color space augmentations, kernel filters, mixing

images, random erasing, feature space augmentation, adversarial training, generative adversarial networks, neural style transfer, and meta-learning have been explored in this field. [198] provides a survey on image data augmentation for deep learning approaches. In Chapter 5, we introduce a scene manipulation, and data augmentation method that takes advantage of a widely used method in computer-aided design in architecture, commonly known as parametric design [22]. In parametric design, various design elements in a procedural model can be transformed using input parameters while maintaining their topological relationships with each other. Work of [95, 92] are examples of systems that utilize parametric workflows for generating various space layout configurations.

In addition, part of our proposed systems intends to determine an optimal arrangement of discrete spatial elements within a room. Such practice is often referred to as floorplanning [54]. Automated floorplanning methodologies have been widely investigated in architectural space layouts, construction [41, 217, 158], electronic design [149, 31, 70], and industrial operation research [1]. Floorplanning aims to achieve a defined functional goal by efficiently generating and evaluating possible spatial combinations while addressing the geometrical and topological constraints of the spatial elements [83]. In electronics, proposed floorplanning methodologies mostly aim at optimizing chip area and wire lengths to reduce interconnections and improve timing [87]. In construction site layout and planning, optimizing the interaction between facilities, such as total inter-facility transportation costs and frequency of inter-facility trips, can also be implemented as objective functions [158].

In floorplanning, various representation methods of spatial arrangements are coupled with optimization engines to efficiently search through all possible combinations of spatial elements. Floorplanning representations are generally divided into two main categories: slicing and non-slicing representations [229]. In slicing methodologies, the floor plan is recursively bisected until each part consists of a single module [235]. Non-slicing representations are utilized for more general use cases where no recursive bisection of a certain area takes place [68, 132, 119]. Multiple studies have integrated these representations with various optimization algorithms such as Simulated Annealing (SA) [98, 99, 235], Genetic Algorithms (GA) [171, 150, 117, 70, 237] and Particle Swarm Optimization (PSO) [213, 33, 89, 205, 146].

Utilizing Generative Advisoral Networks (GAN) have also been widely explored for floorplanning tasks in architectural contexts. Originally introduced by [64], GANs leverage a feedback loop between a Generator and Discriminator model to slowly build an ability to create synthetic data, factoring in phenomena found among observed data. The work of [80] in Pix2Pix extends this generative ability to images by allowing networks to learn a proper mapping from one image to another. [77] take advantage of Pix2PixHD [230] to recognize and generate furnished architectural drawings. They do this by translating floorplan images to programmatic patches of color and, inversely, generating patches of color that are turned into floorplans. Chaillou expands this approach by nesting three models (footprint, program, and furnishing) to generate floorplans, given a set of initial conditions and constraints. Extracting and manipulating existing layouts from images and sketches using model retrieval methods [95, 14] has also been used as a data augmentation method for generating new layouts.

2.6 Scene Synthesis

Indoor scene synthesis aims to generate a feasible furniture layout of various object classes that satisfy both functional and aesthetic criteria [251]. This dissertation utilizes scene synthesis techniques to explore the large-scale deployment of curated context-aware scene augmentation. Early work of synthetic generation focused on hard-coded rules, guidelines, and grammars, resembling a procedural approach for this problem [21, 240, 63]. The work of [143] is a successful example of hard-coded design guidelines as priors for the scene generation process. They extracted these guidelines through consulting manuals on furniture layout [193, 231, 214] and interviewing professional designers who specialize in arranging furniture. A similar approach is also seen in [245], while [243] attempted synthesizing open-world layouts with hard-coded factor graphs.

The work of [59] can be seen as one of the early adopters of example-based scene synthesis. They synthesized scenes by training to build a probabilistic model based on Bayesian networks and Gaussian mixtures. Their problem, however, was one of generating the entire scene, and they utilized a more limited set of input example scenes. In the work of [90], a full 3D scene was synthesized iteratively by adding a single object at a time. This system learned some priors similar to ours, including pairwise and higher-order object relations. The work of [116, 115] and [60] also took room functions into account. While object topologies differ in various room functions, a major challenge in this approach is that not all spaces can be classified with a certain room function. For instance, in a small studio apartment, the living room might serve additional functions such as a dining room and a study space. [184] also proposed a similar approach, involving a Gaussian mixture model and kernel density estimation. However, their system targeted an inverse problem of ours; namely, their problem received a selected object location as input and was asked to predict an object type. We find our problem to be more relevant to the needs of a content creator who knows what object they wish to place in the scene but does not have prior knowledge about a user's surroundings.

Another data-driven approach to scene generation involves modeling human activities and interactions with the scene [58, 129, 60, 167]. Research following this approach generally seeks to model and adjust the entire scene according to human actions or presence. There have also been a number of interesting studies that take advantage of logical structures modeled for natural language processing (NLP) scenarios. The work of [29, 28, 27, 130] are examples of such approach. More specifically, [130] bears a minor resemblance to our approach in training on object relations and the ability to augment an initial input scene. But unlike our work, it augments scenes by merging sub-scenes retrieved from a database. In contrast, we seek to add in individual objects, which is more aligned with the needs of creators of SC experiences. A series of papers (including [192, 34, 7]) proposed generating a 3D scene representation by recreating the scene from RGB-D image input, using retrieved and aligned 3D models. This research, however, involves recreating an existing physical scene, and does not handle adding new objects.

More recent work endeavored to improve learning-based methods, using deep convolutional priors [227], scene-autoencoding [111], and new representations of object semantics [8], to name just a few. [252] addressed a related but distinct problem of synthesizing a scene by arranging and grouping an input set of objects. The work of [174] is another example of using deep generative

models for scene synthesis. Their method sampled each object attribute with a single inference step to allow constrained scene synthesis. This work was extended in PlanIt [228], where the authors proposed a combination of two separate convolutional networks to address constrained scene synthesis problems. They argued that object-level relationships can facilitate high-level planning of how a room should be laid out, while room-level relationships perform well at placing objects in precise spatial configurations. Similar to our scene graph approach is the work of [257], which utilizes a dense scene graph for passing neural messages to augment an input 3D indoor scene with new objects matching their surroundings. However, its scene graph representation does not cover orientational relationships and only covers limited positional relationships between objects.

The method introduced in this dissertation differs from these studies in utilizing an explicit model rather than an implicit structure and taking advantage of alternative discrete relationships with the room itself. Moreover, our model can be trained on datasets that are significantly smaller in size, with faster training and inference time. Furthermore, While work of [228, 257] extend pairwise object to object relations to object to wall relations, we consider the room as a separate entity and simply evaluate whether the object is on edge of the room, the corner of the room, or in the middle of the room. From an architectural perspective, while the walls of indoor spaces are elements that create the encasement of the room, the general location in which the object sits within the room plays a critical role in the collective functionality of the overall space. Therefore, we show modeling the room relationship as a separate explicit entity in the scene graph would benefit the scene augmentation process in both fully automated and user-in-the-loop scenarios.

Moreover, while our work maintains a wide range of overlap with studies in the field of scene synthesis, the main goal is to facilitate SC large-scale content generation in single-blind scenarios where content developers are not aware of the target scenes. Therefore, in this dissertation, we aim to emphasize our explicit structure, which allows SC developers to define new object categories which may not be available in public datasets.

2.7 Spatial Mapping in Spatial Computing

Another feature investigated in this dissertation is to generate a virtual experience that can map to the physical properties of the user's surrounding environment. Such an approach has been widely explored in previous work of [39, 37, 38]. In [40], a group of real people are instructed to dynamically change a physical environment of props to provide haptic feedback for a user in VR. Our work, however aims to generate a virtual experience that corresponds to the natural livable personal environment of the user instead of calibrating props within the physical environment. A similar approach is seen in the work of [120] where a live 3D reconstruction from external depth cameras is utilized to allow modification of the scene, including adding custom virtual objects. In [207], after identifying obstacles and walkable areas of physical space, the authors use a procedural model to generate a planar walkable space within a predefined virtual environment.

For VR environments, techniques in redirected walking [170] also aims to remap the virtual experiences to resolve the possible conflicts between virtual and physical surroundings. While the focus is mainly on providing natural locomotion of a local user, such techniques use subtle

(redirected without the user's knowledge) [19, 15] or overt (detectable by the user) [79, 234, 161] strategies to manipulate the mapping between the user's real and virtual translation and rotation, resulting the user to avoid interference with edges of the usable space or physical obstacles. Architectural manipulation of virtual spaces has also been investigated by re-arranging virtual elements in blind-spots [211] or implementing self-overlapping [210], and flexible virtual spaces [220]. However, redirected walking techniques may introduce simulator sickness [153], interfere with spatial memory [234], and lead to higher cognitive load than real-world locomotion [18]. While such strategies can be applied in VR environments, they cannot generally apply for AR experiences due to the see-through nature of AR.

When addressing the TSMP, mapping virtual avatars within a shared target space while corresponding spatial constraints of each user within their own physical environment is considered an open challenge for next-generation mixed reality telepresence platforms.

Limited work in literature attempt to address this issue. [151] developed a system that generates non-colliding movements for human-like agents interacting with other agents or avatars in a virtual environment. [106] integrate scene semantics with a Markov chain Monte Carlo optimization method to find optimized locations for placing virtual agents close to a single user. Such an approach addresses the spatial limitations of a single user, but not multiple constraints generated by multiple remote users. Alternative methods have been explored in previous work to create mutual grounds and understand user preferences for different types of mutual ground generation. [206] designed three mapping models (scale, kernel, and overlap) for aligning simple rectangular play area spaces. They further conducted extensive user experiments to evaluate participants' sense of co-presence. The work of [109, 45] discussed methods that resembled our module the most. The systems there aimed to optimally map remote environments to maximize user activity space and minimize obstacle discrepancy. In contrast, our mutual space module offers a multi-function optimization workflow, allowing a user in the loop to define weights and constraints for multiple mutual function rooms. For example, instead of finding the maximum walkable space, the user can choose to have a smaller walkable space while maintaining a workable mutual space with other participants. Our method also utilizes an evolutionary optimization algorithm, more suitable for processing multiple input spaces instead of just two spaces.

Chapter 3

Mutual Space Optimization

3.1 Introduction

In the introductory chapter, we elaborated on the TSMP and explained why during telepresence scenarios, finding a common virtual ground physically accessible for all participants is necessary to avoid geometrical and line-of-sight conflicts. In other words, as participants are joining the telepresence experiences from their own spaces, a consensus must be established to identify a mutual space that respects the spatial constraints of all the participants. Yet, locating a mutually accessible virtual ground can be difficult for the users themselves, particularly if they are not aware of the spatial properties of other participants. Moreover, having users manually identify such a mutual space would be imprecise and labor-intensive, especially when considering the contextual properties of the other users' spaces. Without more effective and efficient solutions, the establishment of a contextual mutual space will be a bottleneck for multi-user immersion experiences.

Motivated by this challenge, this chapter presents a novel method to optimize contextual mutual spaces in a multi-user immersion setting. Our method relies on existing semantic scene maps to identify shareable functional spaces and is general enough to optimize contextual mutual spaces even when the users' spaces have very different layouts and sizes (see results in Figure 3.6). For illustration purposes, we will use standable and sittable as the two exemplary contextual functions to develop our method, and the proposed solution is compatible with other contextual functions that can be modeled by the same mathematical framework. The method formulates an optimization problem to seek the maximal mutual spaces. Furthermore, if one can assume the users have the freedom to rearrange furniture objects on the floor, we introduce a more delicate optimization process to further increase the mutual space's size while balancing the users' efforts to physically move the objects as another constraint. To effectively solve the above two problems, we propose to use a generative modeling approach. Clearly, we believe other comparable algorithms that optimize these NP-Hard problems are equally effective. Nevertheless, our results validate a new approach capable of automatically recommending contextual mutual space to multiple participants of virtual immersion experiences in SC applications.

We believe our proposed framework can play a role in facilitating remote workplace practices

and virtual collaborations by decreasing the spatial requirements for telepresence systems. Instead of setting up large open spaces required for such workflows, our system would allow users to join from their personal spaces, with minimum modifications to their surrounding environment. Physical and virtual rearrangements would be optimized based on the number of participants and their local environments. In AR experiences, the topological relationship and line of sight between all participants would be maintained without any conflicts between remote users and local physical obstacles, while in VR, our system can recommend spatial modifications and provide the required interaction area between multiple users.

3.2 Methodology

Our solution consists of the following four steps: (i) semantic segmentation of surrounding environments; (ii) topological scene graph generation; (iii) mutual space identification; and (iv) manipulation of furniture to further maximize the mutual space. In this section, we will elaborate on the details of the four steps. To start, we will define the terminologies and notations used in the chapter.

Given a closed 3D room space in $\mathbb{R}^3$, one can project its enclosure, i.e., floors, ceilings, and walls, via an orthographic projection to form a 2D projection, which is commonly known as the floorplan of the space. If we assign the (x,y) coordinates on the floorplan plane and the z coordinate perpendicular to the floorplan plane, simplifying our optimization problems on to the (x,y) plane significantly reduces the complexity of our algorithms. It also implies an assumption that there is no overlap between two objects on the (x,y) plane but with different z values. Nevertheless, we believe such simplification is reasonable for analyzing the majority of room structures and thus does not compromise the generality of our analysis provided herein.

Hence, we define for each user i their own room space expressed as a 2D floorplan as R_i. Each k-th object (e.g., furniture) in R_i is denoted as $O_{i,k}$.The collection of all n_i objects in R_i is denoted as $\mathcal{O}_i = \{O_{i,1}, O_{i,2}, ...O_{i,n_i}\}$. $\bar{O}_{i,k}$ represents the boundary of the object $O_{i,k}$. Similarly, $\bar{R}_i$ represents the boundary of the room R_i. Finally, we define the area function as $K(O)$.

Semantic Segmentation

Given the measurement of the surrounding physical environments as large sets of point cloud data, one can take advantage of the semantic segmentation methods widely investigated in computer vision literature [165, 121, 3] to segment their spatial boundaries and obtain their geometric properties, such as dimensions, position and orientation, object classification, functional shapes, and their weights. In doing so, we can convert the 3D point cloud data to labeled objects $O_{i,k}$ with a bounding box as $\bar{O}_{i,k}$.

Additionally, in this chapter, we exclude lightweight objects (such as pillows, alarm clocks, laptops, etc.) positioned on larger furniture. This is to simplify our calculations in the next steps as we assume these lightweight objects can be easily moved by the users and do not need to be

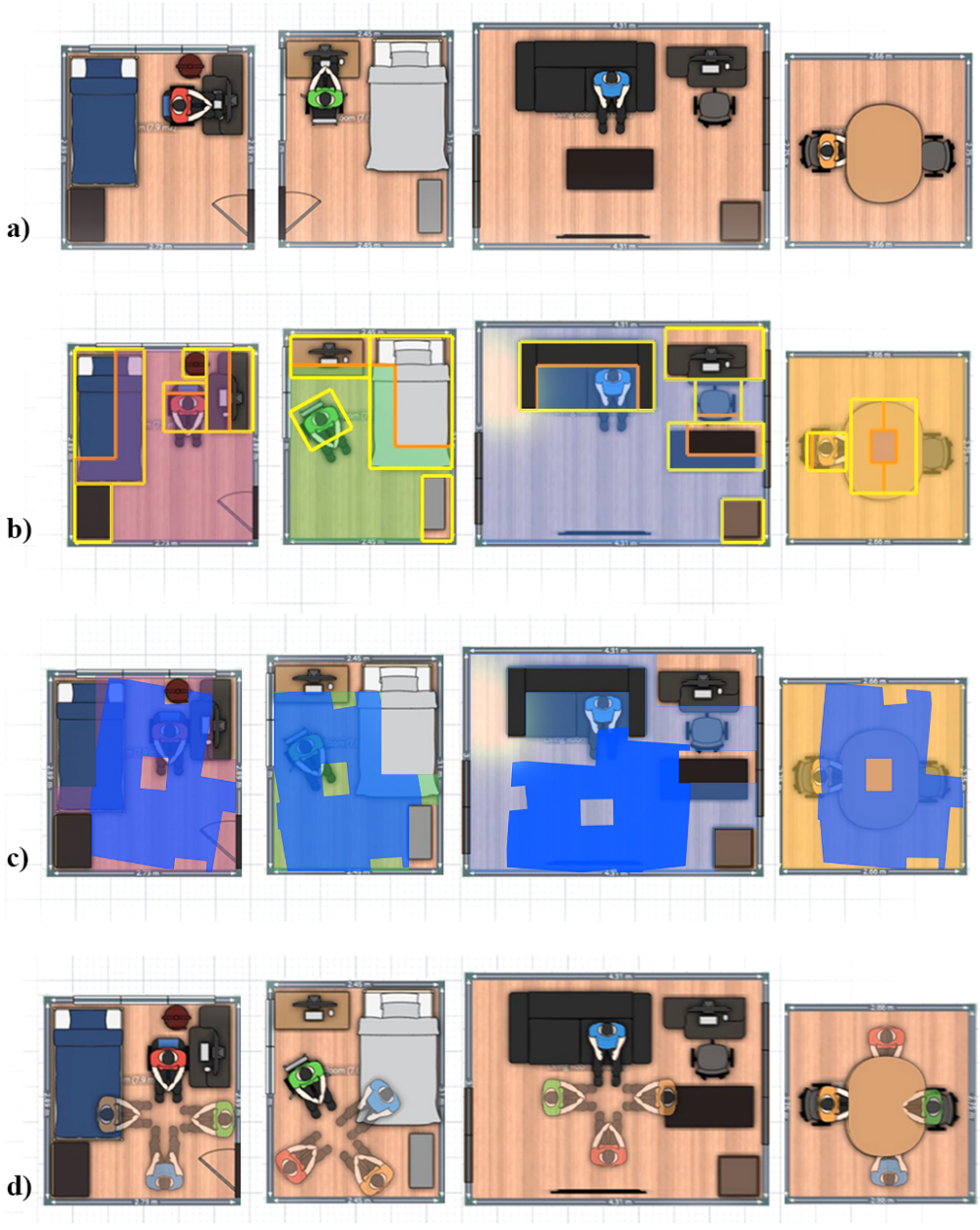


Figure 3.1: Abstract illustration of our proposed framework a) initial settings with different spatial restrictions b) semantic segmentation defining standable (yellow boundaries) and sittable (orange boundaries) areas c) search for mutual sittable space (this step can be before, after or simultaneous with object repositioning) d) virtual arrangement of avatars with deterministic line of sight of all participants.

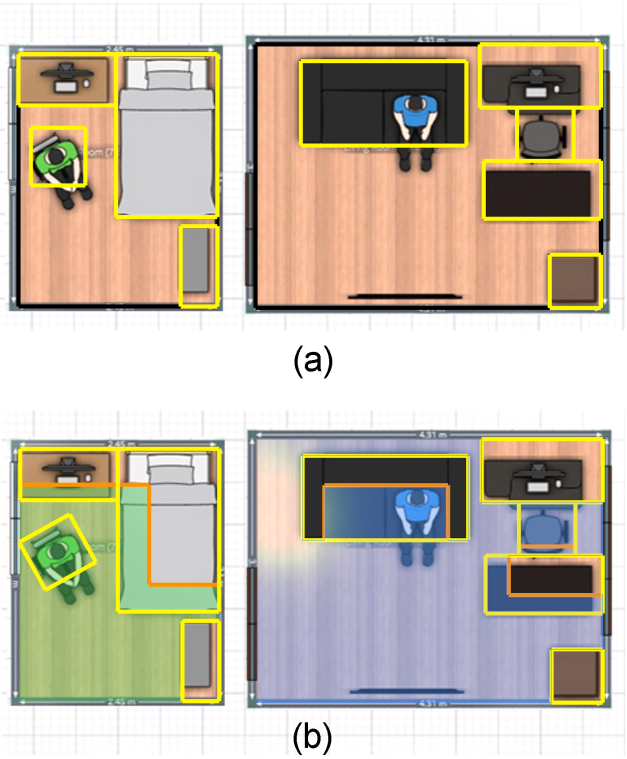


Figure 3.2: Comparison between available (a) standing only and (b) standing and sitting area in rooms.

considered in the optimization criteria. Such classification is dependent on the output labeled object categories above.

In the experiment section below, since the implementation of a computer vision algorithm for semantic segmentation is not the main focus of this chapter, we will directly integrate a modified version of Matterport 3D [30] object classifier in our system. This module can be replaced with any other robust semantic segmentation algorithms, as long as they provide bounding box coordinates for each object category. In a companion Matterport 3D [30] dataset, out of 1,659 unique text labels, we classify 134 of the labels as lightweight objects and filter their corresponding bounding box from our workflow.

Figure 3.2(a) illustrates the result of semantic segmentation of two room spaces projected onto the (x, y) plane.

Topological Scene Graph

After identifying the bounding box, orientation, and category type of each object in the scene R_i, a topological graph is readily generated that describes the relationship and constraints of the objects between one another within R_i. This step will allow us to identify usable spatial functions such as standing in virtual immersion located between the objects. We categorize this type of functions as *standalone spatial functions*, and their spaces are called *standalone spaces*.

A topological scene graph will also allow us to identify other spatial functions of the objects themselves, such as sitting on a chair and working on a table. But note that such functions as sitting or working are also constrained by the distances between the object that performs the function and its adjacent other objects. For example, a side of the table can not be utilized for working purposes if that side is adjacent to other furniture or building elements (such as walls, doors, etc.). We categorize this type of functions as *auxiliary spatial functions*, and their spaces are called *auxiliary spaces*.

In this chapter, we will use two spatial functions *standable* and *sittable* as an example to demonstrate how to integrate both standalone spatial functions and auxiliary spatial functions in the optimization of contextual mutual spaces for multi-user interaction in AR/VR.

Finally, we emphasize that standalone spaces and auxiliary spaces are not mutually exclusive. For example, in this chapter, we will classify that a standable space can be assumed to be sittable as well. However, the vice versa may not be true. For example, a portion of a sittable space involves a part of a bed object, which we will not assume to be standable. Such contextual constraints can be highly customizable based on the content of the AR/VR application. But the framework that we are introducing in this chapter is general enough to accommodate other contextual interpretations of the standalone spatial functions and auxiliary spatial functions.

In our implementation, we use a doubly-linked data structure to construct the graph. For each side face of an object's bounding box, we define the closest adjacent objects to the face and calculate the distance between the object and the specified face. This information would be stored at the object level, where topological distances and constraints are referenced using pointers.

Mathematically, for each object $O_{i,k}$, we define the function $\delta X_{\max}(O_{i,k})$ as the shortest distance between the points in $O_{i,k}$ that have the maximal x value and the other objects including $\bar{R}_i$. Similarly, we define the functions $\delta X_{\min}(\cdot)$, $\delta Y_{\max}(\cdot)$, and $\delta Y_{\min}(\cdot)$.

Mutual Space Identification

In this step, we will identify the geometrical boundaries of available spaces in each room and then align the calculated boundaries of all rooms to achieve maximum consensus on mutual spaces.

First, using the geometrical and topological properties extracted in the previous steps, we are ready to calculate available spaces in each room based on two categories, namely, the standalone spaces and auxiliary spaces. Specifically, we will formulate the calculation of the two most typical spatial functions as examples again, namely, standable and sittable.

Standable Spaces

Standing spaces consist of the volume of the room in which no object located within a human user's height range is present. In such spaces, user movement can be performed freely without any risk of colliding with an object in the surrounding physical environment. Activities such as intense gaming or performative arts can be safely executed within these boundaries. Such spaces are also suitable for virtual reality experiences, where users may not be aware of their physical surroundings.

We calculate the available standing space (S) for room R_i simply as follows:

$$S_i = R_i - \bigcup_{k=1}^{n_i} O_{i,k}. \tag{3.1}$$

Sittable Spaces

The calculation of maximal sittable spaces is more involved than that of the standable spaces above. As we mentioned before, sittable spaces normally extend the standable spaces by adding areas where humans are able to sit. Furniture types such as sofas, chairs, and beds include sitting areas that can extend usable spaces of a room for social functions such as general meetings, design reviews, and conference calls.

To start, we define a sittable threshold $\varepsilon(O_{i,k})$ to calculate the sittable area within the bounding box of the object $O_{i,k}$. In other words, $\varepsilon(O_{i,k})$ is the maximum distance inward from an edge of the object's bounding box that can be comfortably sit on. We use measurements from [169] to define the ε of each furniture type. If object O is classified as non-sittable, then $\varepsilon(O) = 0$.

Therefore, we can first calculate the non-sittable area of an object O as

$$N(O) \doteq \{\forall p \in O : B(p, \varepsilon(O)) \cap O = B(p, \varepsilon(O))\}, \tag{3.2}$$

where $B(p, \varepsilon(O))$ is a sphere in $\mathbb{R}^2$ centered at p and with radius $\varepsilon(O)$.

We note that sittable spaces do not necessarily comprise only sittable objects but rather describe an area where a sittable object can be placed in. For example, while an individual may not be able to comfortably sit on the top of the table, the foot space below the table can be considered a sittable space. Therefore, in such a context, the sittable area of the room is always larger than its standable area.

Moreover, the sittable areas of each object in the room are constrained by the topological positioning of the object. If any of the object's boundaries are adjacent to a non-sittable object (such as a wall, bookshelf, etc.) or does not contain enough standable area between itself and a non-sittable object, the sittable area of the side of the face should be excluded. For instance, if a table is positioned in the center of a room, with no other non-sittable object around it, the sittable area would be calculated by applying the sittable threshold to all four sides of the table's boundaries. However, if the table is positioned in the corner of the room, then there will be no sittable area accumulated for the sides that are adjacent to the wall.

To simplify our calculation, we define a surrounding boundary threshold $\rho(O)$ for object O, which measures the distance from any object's boundary point outward that allows that point to

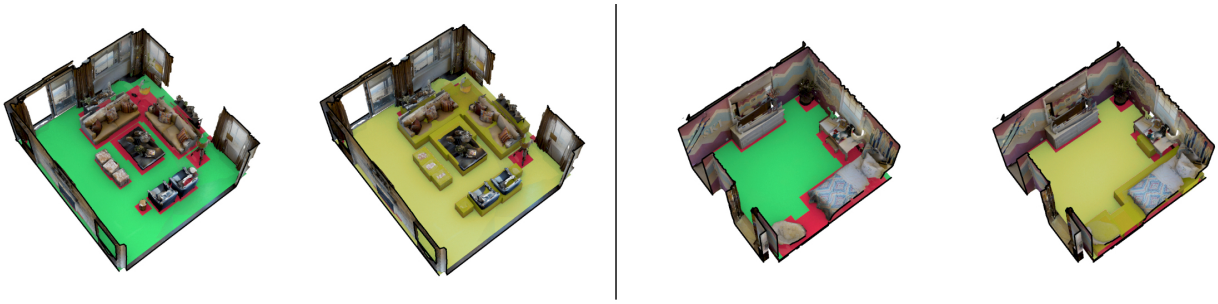


Figure 3.3: Standable (green), non-standable (red) and sittable spaces (yellow) for two example scenes from the Matterport 3D dataset.

remain part of the sittable space of the object. In other words, if the boundary point is close to other objects or the room boundary within distance ρ, then that point is not sittable. $C(O_{i,k})$ defined below collects all such points for exclusion from $O_{i,k}$ in room R_i:

$$C(O_{i,k}) = \begin{array}{l} \{\forall p \in O_{i,k} : B(p, \varepsilon(O_{i,k}) + \rho(O_{i,k})) \cap \bar{R}_i \neq \emptyset \\ \text{or } B(p, \varepsilon(O_{i,k}) + \rho(O_{i,k})) \cap O_{i,h} \neq \emptyset, h \neq k\} \end{array} \tag{3.3}$$

where $\emptyset$ denotes the empty set. Therefore, the sittable space of each object O is simply defined as

$$A(O) = O - N(O) \cup C(O). \tag{3.4}$$

Finally, the total sittable space $A(R_i)$ for the room R_i is

$$A(R_i) = \bigcup_{k=1}^{n_i} A(O_{i,k}) + A(S_i). \tag{3.5}$$

Figure 3.3 illustrates two example rooms and compares their standing and sitting areas.

Maximizing Mutual Spaces

Now we consider an immersive experience where there are m subjects and therefore m room spaces $(R_1, R_2, \cdots, R_m)$, respectively. Then, in the (x, y) coordinates, we define a rigid-body motion in $\mathbb{R}^2$ as $G(F, \theta)$, where θ describes a translation and a rotation.

If we want to maximize a mutual standable space, we can apply one $G(S_i, \theta_i)$ to each individual standable space S_i for the i-th user. The optimal rigid body motion then maximizes the area of the interaction space:

$$(\theta_1^*, \cdots, \theta_m^*) = \arg\max K(\bigcap_{i=1}^{m} G(A(R_i)), \theta_i)). \tag{3.6}$$

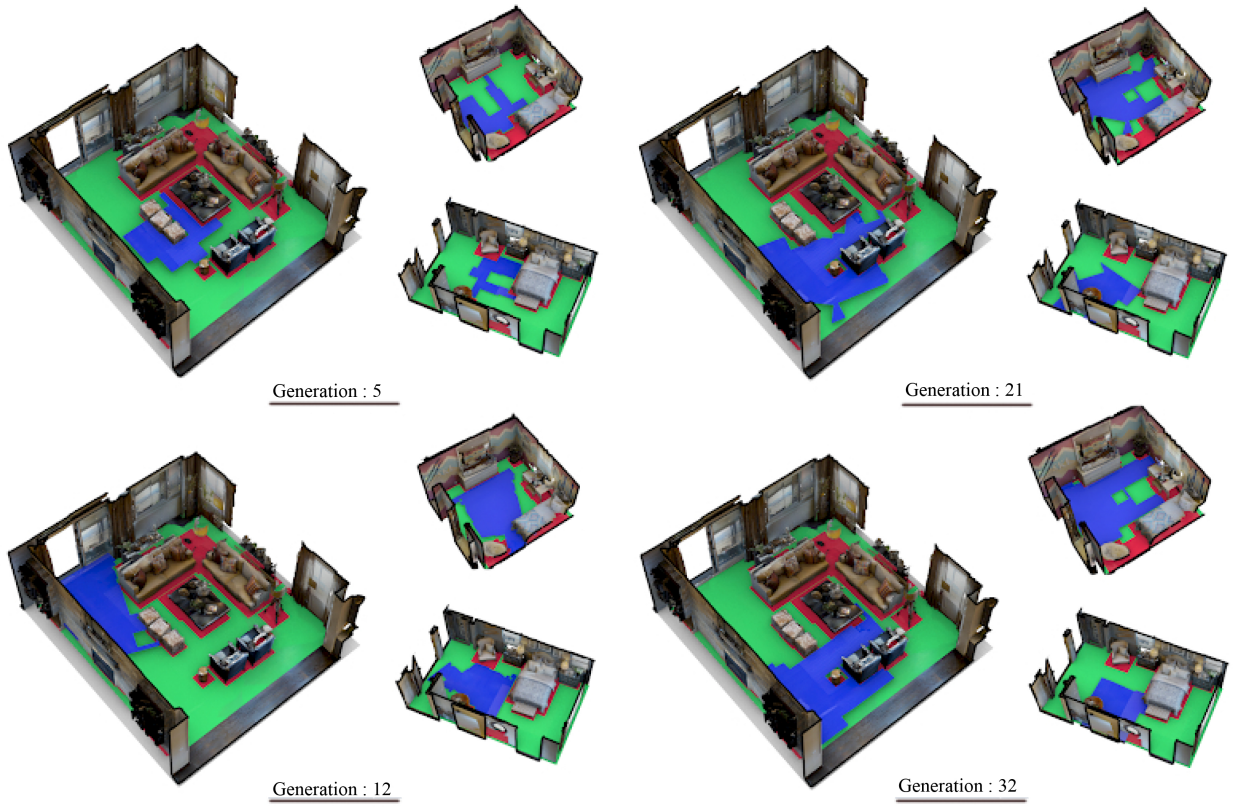


Figure 3.4: Mutual Spatial boundaries (blue) for different generations of the search mechanism. The green area indicates standable spaces and the red area indicates non-standable spaces. The result shows that the optimized mutual standable space increases over generations.

Then the maximal mutual standable space can be calculated as

$$M_A(R_1, \cdots, R_m) = \bigcap_{i=1}^{m} G(A(R_i), \theta_i^*) \tag{3.7}$$

Similarly, one can calculate the maximal mutual sittable space $M_A(R_1, \cdots, R_m)$ by substituting the rigid body motions in (7.3) that maximizes their intersection area function in (7.2).

Furniture Movement Optimization

In the event where individual spaces R_i include movable furniture, additional optimization can be considered to potentially increase the maximal mutual spaces. Diverging from merely considering rigid-body motions to transform just the coordinate representation of the spaces, we consider moving

furniture objects in space, which has an additional cost of human effort. Consequently, we will formulate this effort as part of our optimization objective.

More specifically, given a rigid-body motion G, we definite $\|G\|_t$ as the Euclidean distance of its translation vector. Then we define

$$E = w\|G\|_t, \tag{3.8}$$

where w is a given parameter that approximates the weight of each object. Note that such weight estimate can be looked up using architecture standards such as in [169]. Hence, if a room space R_i has n_i objects, then the total effort to rearrange the space is

$$E(R_i, \Theta_i) = \sum_{k=1}^{n_i} w_k \|G(O_{i,k}, \theta_{i,k})\|_t, \tag{3.9}$$

where $\Theta_i = \{\theta_{i,1}, \cdots, \theta_{i,n_i}\}$ denotes the collection of n_i rigid-body motion parameters.

Since solving for the optimal object transformation is an NP-Hard problem, in this chapter, we will demonstrate a heuristic-based but practical algorithm to optimize it in a step-by-step greedy fashion.

$$\min \sum_{i=1}^{m} E(R_i, \Theta_i^s) \quad \text{subj. to} \quad K^s(\bigcap_{i=1}^{m} G(S_i, \theta_i^s)) \text{ increases } 10\%, \tag{3.10}$$

where K^s indicates the area value at the s-th step with respect to transformation coefficients Θ_i^s and θ_i^s. The iteration would stop if the optimization cannot further increase the area of the mutual space.

3.3 Implementation on a 3D Scanned Dataset

To comprehensively observe how the search and recommendation system performs given various room types with different spatial organizations, we take advantage of available 3D datasets to be able to experiment with large quantities of real-world case studies. We use the Matterport 3D [30] dataset, and randomly sample subsets of varying sizes of 3D scanned scenes, and perform the search and recommendation practice on each subset to observe how the mutual spaces are identified and maximized with our algorithm. Matterport 3D is a large-scale RGB-D dataset containing 90 building-scale scenes. The dataset consists of various building types with diverse architectural styles, each including numerous spatial functionalities and furniture layouts. Annotations of building elements and furniture are provided with surface reconstructions as well as 2D and 3D semantic segmentation. For our experiments, we initially exclude spaces that are not generally used for multi-user interaction (bathroom, small corridors, stairs, closet, etc.). Furthermore, we randomly group the available rooms into groups of 2, 3, and 4. We utilize the object category labels provided in the dataset as the ground truth for our semantic labeling purposes.

We implement our framework using the Rhinoceros 3D (R3D) software and its development libraries. For each room, we convert the labeling data structure provided by the dataset to our proposed topological scene graph. This provides the system with bounding boxes for each object and the topological constraints for their potential rearrangement. Using such a structure, we are

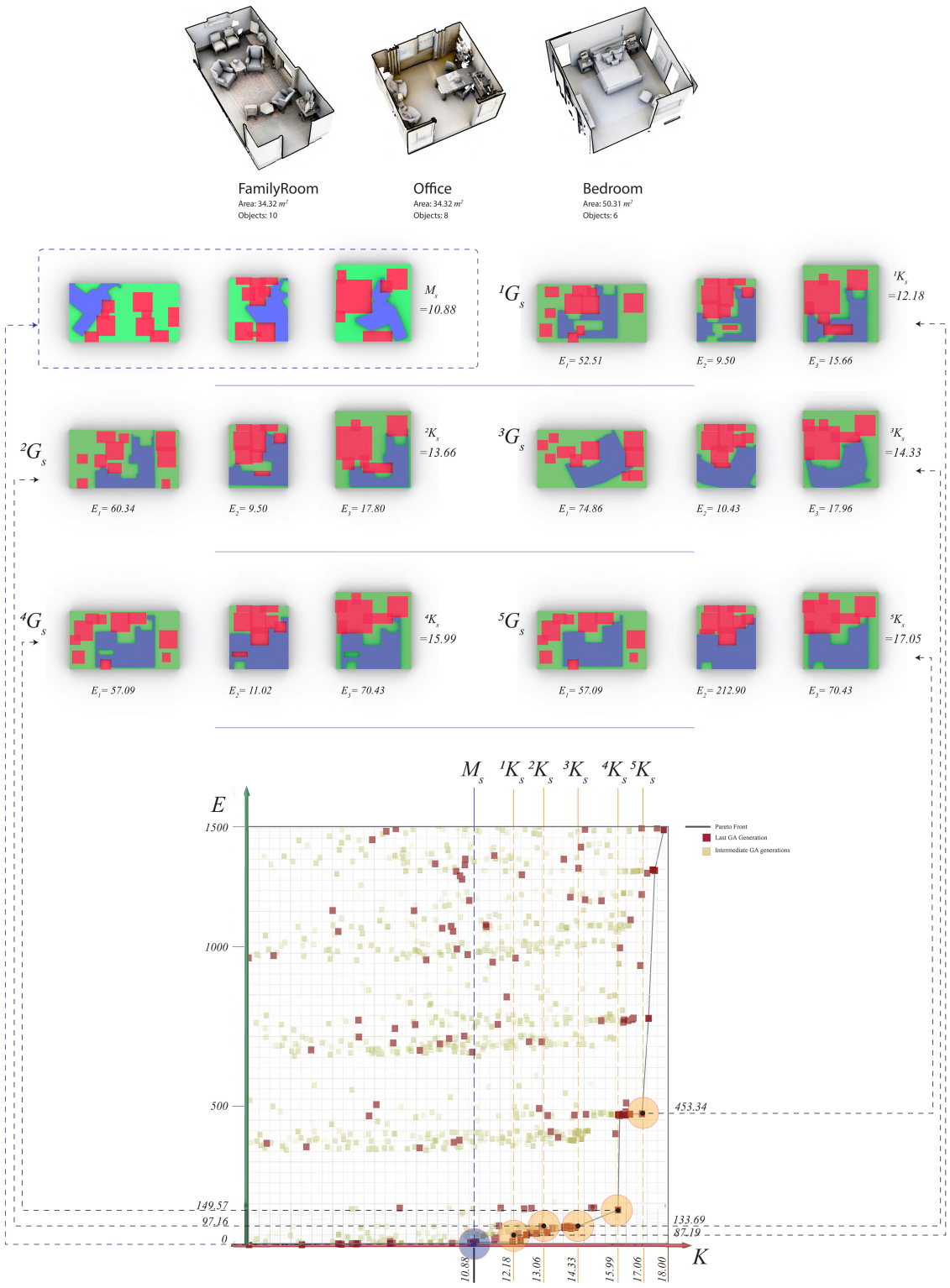


Figure 3.5: Furniture optimization and manipulation. In each step, a 10% increase of mutual space area (K) is determined, while minimizing the overall effort needed (E) for the required transformation (G).

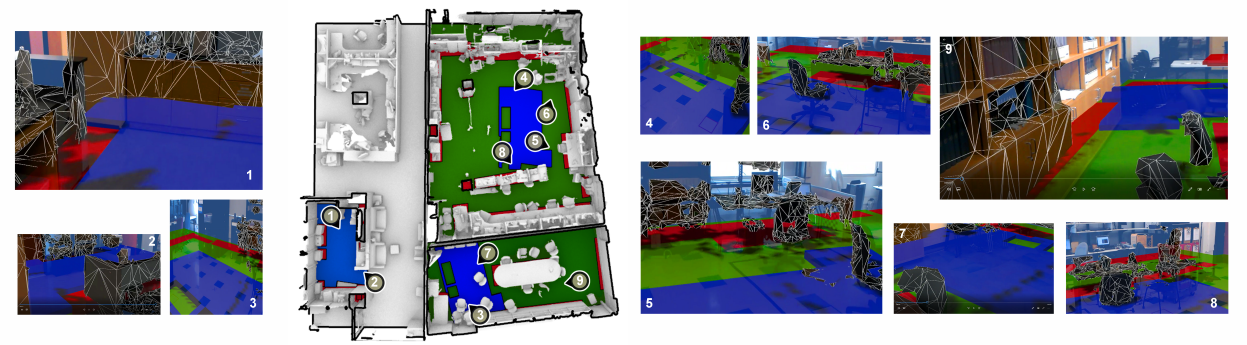


Figure 3.6: Screenshots from HoloLens illustrating the identified mutual boundaries as augmented overlays for three rooms: A) kitchen; B) conference room; C) robotic laboratory. Blue color indicates mutual boundaries, green color indicates standable spaces and red color indicates non-standable spaces.

able to extract the standable and sittable spaces for each room based on our proposed methodology. Figure 3.3 illustrates the available standable and sittable boundaries for two sample rooms processed by our system. We define a constant $\varepsilon_{O_{i,k}} = 70$ cm for all sittable objects.

Next, we integrate our algorithm with a robust Strength Pareto Evolutionary Algorithm 2 (SPEA 2) [258] available through the Octopus multi-objective optimization tool in R3D. The fitness function (7.2) is used to maximize the mutual space for calculated standable spaces. Our genotype is comprised of the transformation parameters $G(F,\theta)$ of each room, allowing free movement and orientation to achieve maximum spatial consensus. Therefore, a total of $3(n-1)$ genes are allocated for the search process. This process would result in the shape, position, and orientation of the maximum mutual boundary of the assigned rooms. We use a population size of 100, mutation probability of 10%, mutation rate of 50%, and crossover rate of 80% for our search. As our solution integrates a genetic search, we expect the result to gradually converge to the global optimum. Figure 3.5 shows how the mutual space boundary is progressively expanded with the increase of the generations in our search.

Expanding further, we extend our search by manipulating the scene with alternative furniture arrangements. As the objective goal is to achieve an increased mutual spatial boundary area with minimum effort, we calculate the E based on the transformation parameters assigned to each object present in the room. However, in our current implementation, the genetic algorithm integrated into our solution is not capable of adapting dynamic genotype values and, therefore, cannot update the topological values of each object (δX_{max}, δX_{min}, δY_{max}, δY_{min}) during the search process. Hence, to avoid transformations that result in physical conflicts of manipulated furniture, we penalize phenotypes that contain intersecting furniture within the scene. This penalty is added to the E value, lowering the probability of such phenotypes being selected or surviving throughout the genetic generations.

The optimization can either be (i) triggered in separate attempts for each step (s), where the mutual area value (K) is constrained based on the resulting step value, or (ii) executed in a single attempt where minimizing E and maximizing K are both set as objective functions. In the latter, M_S is defined as the solution which holds the largest K while $E = 0$. Executing the optimization in a one-time event is also likely to require additional computational costs due to the added complexity to the solution space.

3.4 Results

Figure 3.5 illustrates our results for a furniture manipulation optimization task applied to three example rooms. A total of 34 objects are located in the rooms. To shorten our gene length, we do not apply rotation transformations to objects. We use a population size of 250, mutation probability of 10%, mutation rate of 50%, and crossover rate of 80% for the scene manipulation search. We visualize the standable, sittable, and mutual boundaries for each spatial expansion step. Moreover, we report the corresponding E for each room in the alternative furniture layout. Our results in this example indicate the solution can identify solutions that increase the maximum mutual boundary area up to 65% more than its initial state before furniture movement.

The optimization process was able to generate a well-defined Pareto front, as seen on the bottom of Figure 3.5, locating both the two extreme points and numerous intermediate trade-off points representing non-dominated solutions. The bottom region of the curve is flat, indicating that for a similar amount of effort, a significant increase in mutual standable area can be achieved. The trade-off frontier thus starts at point M_S, becoming very densely populated in its initial soft slope. This shows that for each modest increase in physical effort (that is, in moving furniture), there can be extensive gains in the mutual shareable area, which is an interesting result. After $s = 4$, the Pareto front becomes increasingly steep, signaling that the user would now have to significantly increase physical effort levels for modest gains in the shareable area. Point ${}^4G_s$ thus seems to indicate a breaking point of diminishing returns.

Similar to the M_S search, in smaller furniture optimization steps, the algorithm seeks solutions that are highly dependent on the transformation parameters $G(F,\theta)$ of the room itself, whereas, in larger steps, we observe the algorithm correctly moving the objects to the more populated side of the room in order to increase the empty spaces in available. In rooms where objects are facing the center, and empty areas are initially located in the middle portion of the space, we see the objects being pushed towards the corners or outer perimeter of the room in order to increase the initial unoccupied areas.

Due to the smaller gene size, calculating the optimal M_S (maximum mutual space without furniture manipulation) executes much faster compared to $E(R_i,\Theta_i^s)$ optimization, where the complexity of the search mechanism radically increases due to the additional object transformation parameters. The speed of the $E(R_i,\Theta_i^s)$ optimization is also highly dependent on the transformation range of each object, meaning that objects in larger rooms have more movement options to choose from than those in small, constrained rooms. We observe an example of this effect in the later AR experiment (Section 3.5), where the smaller space (kitchen) dominates the search process, causing

the final mutual outcome between the rooms to maintain a very similar shape to the open boundaries of the smaller space. While such an effect would still provide a well-constrained problem for medium-sized rooms with multiple objects (such as the conference room), there are many possible ways of fitting the smaller space in larger rooms with open spaces (such as the robotics laboratory), resulting in an under-constrained optimization problem.

3.5 Augmented Reality Visualization

To explore the usability aspect of our solution in real-world scenarios, we deploy the resulting spatial segmentation in AR using the Microsoft HoloLens, a mixed reality HMD. In this experiment, three types of rooms were defined as potential telecommunication spaces: (i) a conventional meeting room, where a large conference table is placed in the middle of the room, and unused spaces are located around the table (ii) a robotics laboratory, where working desks and equipment are mainly located around the perimeter of the room, while some larger equipment and a few tables are disorderly positioned around the central section of the lab (iii) a kitchen space, where surrounding appliances and cabinets are present in the scene.

After the initial scan of the surrounding environment by the user of each room, the geometrical mesh data is sent to a central server for processing. This process happens in an offline manner, as the current HoloLens hardware is incapable of processing the computations that our solution would require. In addition, we scan the space using a Matterport camera and perform the semantic segmentation step using Matterport classifications to locate the bounding boxes of all the furniture located in the room. We then feed the bounding box data to our algorithm for mutual boundary search. The implementation outputs spatial coordinates for standable and sittable areas, which are automatically updated in the Unity Game Engine to be rendered in the HoloLens.

Figure 3.6 shows how the spatial boundary properties are visualized within the HoloLens AR experience. The red spaces indicate non-standable objects, the green spaces indicate standable boundaries, and the blue spaces indicate mutual boundaries that are accessible between all users. The visualized boundaries are positioned slightly above the floor level, allowing users to identify the mutual accessible ground between their local surroundings and the remote participant's spatial constraints.

Visualizing the mutual ground within the space itself using HoloLens allows us to understand how complex the problem can be when executed in a manual fashion. Some corner spaces that are not typically used as default social areas of a certain room may become the only required common ground for interaction with other rooms. Overcoming this spatial bias is easily executed within the algorithm; meanwhile, this may not happen so easily and instantly when individuals are left to deal with it on their own. However, due to the limited field of view of the HoloLens, detecting non-physical boundaries placed at a lower visual height becomes difficult to follow. This issue proved more challenging when walking closer to the non-orthogonal edges of the mutual bounding area, where an individual could easily step outside the designated area. The shareable area also included a number of voids, which resulted in an inconsistent walking path inside the standable spaces. Moreover, the accuracy of the real-time mesh reconstruction in HoloLens played a critical

role in calculating the required rendering occlusions for the visualized boundaries. This was mainly because the position of the visualization was reflected close to the floor with many objects placed over it, therefore failing to detect occluding objects, a fact that often misled the user in identifying whether the space was mutually accessible or not.

3.6 Conclusions

In this chapter, we introduce a novel optimization and manipulation framework to generate an optimal common virtual space for interactions that mostly involve standing and sitting. Our framework further recommends the movement of surrounding furniture objects that can expand the size of the mutual space with minimal physical effort. We integrated our framework with a Strength Pareto Evolutionary Algorithm for an efficient search and optimization process. The multicriteria optimization process was able to generate a well-defined Pareto front of trade-offs between maximizing mutual space and minimizing physical effort. The Pareto front is more densely populated in some sections of the frontier than others, clearly identifying the best trade-offs region and the on-start of diminishing returns.

Furthermore, we demonstrate how output solutions can be visualized using a HoloLens application. Results show that the proposed framework can effectively discover optimal shareable space for multi-user virtual interaction and thus provides a better user experience compared to manually labeling shareable space, which would be a labor-intensive and imprecise workflow. In such a context, if all participants stand within the calculated mutual spatial boundaries, the line of sight between all participants will be deterministic. In addition, no remote participant will be positioned in a conflicting location for any local user and would comply with the spatial constraints for all other participants.

There are, of course, limitations to our work. First, furniture with fixed positions is not automatically detected in our current implementation. We believe such a feature can be integrated with further improvements in semantic segmentation methodologies or can be optionally specified by the user whether an object is fixed or not. In addition, the furniture weight is calculated based on standard assumptions. We envision that with the growth of spatial computing procedures, such metadata of the surrounding environment will be customizable by the user itself and can be loaded upon each mutual spatial search execution. Future work can consist of integrating robust floorplanning representations with the current search mechanism to minimize computation cost and complexity.

Chapter 4

Contextual Scene Generation

4.1 Introduction

As previously discussed in Chapter 1.1, SC experiences are physically constrained by the geometry and semantics of the 3D user environment where existing furniture and building elements are present [151, 170]. Contrary to traditional 2D graphical user interfaces, where a flat rectangular region hosts digital content, 3D SC environments are usually occupied by physical obstacles that are diverse in their shape and function. Therefore, how one can assess content placement in SC experiences is highly dependent on the user's target scene.

However, since different users may reside in different spatial environments, which differ in dimensions, functions (rooms, workplace, garden, etc.), and open usable spaces, existing furniture and their arrangements are often unknown to the developers, making it very challenging to design a virtual experience that would adapt to all users' environments. Currently, contextual placement is addressed by asking users themselves to identify the usable spaces in their surrounding environments or manually positioning the augmented object(s) within the scene. Then, virtual object placement in most AR experiences is limited to specific surfaces and locations, e.g., placing objects naively in front of the user with no scene understanding or only using basic horizontal or vertical surface detection. These simplistic strategies may work to some extent for small virtual objects, but the methods break down for larger objects or complex scenes with multiple object augmentation requirements.

The task of adding objects to existing constructed scenes falls under the problem of *constrained scene synthesis*. The work of [90, 129, 111, 165, 174, 228] are examples of such an approach. However, there are two major challenges in the general literature that create bottlenecks for virtual content augmentation in SC experiences. First, current scanned 3D datasets publicly available are limited in size and diversity and may not offer all the data required to capture the topological properties of the rooms. For instance, *pose*, the direction in which the object is facing, is a critical feature for understanding the orientational property of an object, and yet, such a property is not clearly annotated for any objects in many large-scale real-world datasets such as SUN-RGBD and Matterport3D. Therefore, more recent research has adapted synthetic datasets, which do not

necessarily need to be manually annotated to pose prior information.

However, a critical drawback of using synthetic datasets is that they cannot capture the natural transformation and topological properties of objects in real-world settings. Indeed, topological relationships between objects in real-world scenes typically exceed the theoretical design assumptions of an architect and instead capture contextual relationships from a living environment. Moreover, the limitations of the modeling software for synthetic datasets can also introduce unwanted biases to the generated scenes. The SUNCG [204] dataset, for instance, was built with the Planner5D platform, an online tool that any user around the world can use. However, it comes with modeling limitations for generating rooms and furniture. Orientations are also snapped to right angles by default, which makes most scenes in the dataset Manhattan-like. More importantly, there is no indication whether the design is complete or not; namely, a user may just start playing with the software and then leave at a random time, while the resulting arrangement is still captured as a legitimate human-modeled arrangement in the dataset.

Second, recent models take advantage of implicit deep learning models and have shown promising results in synthesizing indoor scenes. Yet, these approaches fall short for content developers to parameterize customized placement in relation to standard objects in the scene and to generate custom spatial functionalities. One major limitation of these studies is that they do not have direct control over objects in the generated scene. For example, authors of [111] reported they could not specify object counts or constrain the scene to contain a subset of objects. Such limitations come from the implicit nature of such neural networks. Implicit models produce a black-box tool, which is difficult to comprehend should an end-user wish to tweak its functions. In cases where a new object type (which has not been previously seen in prior datasets) needs to be placed, implicit structures may not provide abilities to take into account manually defined topological properties by a user. Moreover, training deep neural networks requires large datasets, a bottleneck that we have discussed above.

Motivated by these challenges, in this Chapter, we introduce SceneGen, a generative contextual augmentation framework that leverages explicit scene graph models to predict the functional placements of new virtual objects in an indoor target scene. Contrary to the implicit models, SceneGen is based on clear, logical object attributes and their architectural relationships with other objects and the room. In light of the existing body of literature on semantic scene graphs, we leverage this approach to encapsulate the relevant object relationships for scene augmentation. Scene graphs have already been in use for general scene generation tasks; they can also inform the intelligent placement of virtual objects in physical scenes. We use kernel density estimation (KDE) to build a multivariate conditional model to encapsulate explicit positioning and clustering information for objects in various room types. This information will allow our algorithm to calculate a probability distribution to place and orient the new object in a scene while satisfying its physical and functional requirements. From the calculated probabilities, we generate a score for each potential placement of the new object, visualized in a heat map over the room. Our system is designed for both fully automated scene augmentation and also user-in-the-loop scenarios, allowing the user to understand the influence of the relationship features and their impact on the results.

Our contributions can be summarized as follows:

1. We introduce a spatial Scene Graph representation that encapsulates the positional and orientational relationships of a scene. Our proposed Scene Graph captures pairwise topology between objects, object groups, and the room.

2. We develop a prediction model for contextual object augmentation in existing scenes. We construct an explicit Knowledge Model which is trained from Scene Graph representations captured from real-world 3D scanned data.

3. To learn orientational relationships from real-world 3D scanned data, we have manually labeled the Matterport3D dataset with pose directions using an open-source labeling tool for fast pose labeling.

4. We develop an AR application that scans a user's room and generates a Scene Graph based on the existing objects. Using our model, we sample poses across the room to determine a probabilistic heat map of where the object can be placed. By placing objects in poses where the spatial relationships are likely, we are able to augment scenes that are realistic.

We believe our proposed system can facilitate a wide variety of SC applications. Augmenting virtual objects to scenes has been explored in online-shopping settings, and collaborative environments require placing one user's objects into another user's surroundings. In addition, content creation for SC experiences requires long hours of cross-platform development on current applications, so our system will allow faster scene generation and content generation in AR/VR experiences.

4.2 SceneGen Overview

SceneGen is a framework to augment scenes with virtual objects using a generative model to maximize the likelihood of the relationships captured in a spatial Scene Graph. Specifically, if given a partially filled room, SceneGen augments it with one or more new virtual objects in a realistic manner using an explicit model trained on relationships between objects in the real world. The SceneGen workflow is shown in Figure 7.4.

In this Chapter, we first introduce a novel Scene Graph that connects the objects and the room (both represented as nodes) using spatial relationships (represented as edges) in Section 4.3. For each object, these relationships are determined by positional and orientational features between itself and other objects, object groups, and the room.

In Section 4.4 we show how from a dataset of rooms, we can extract these Scene Graphs to construct a Knowledge Model that is used to train explicit models that approximate the probability density functions of position and orientation relationships for a given object using kernel density estimation. In order to augment a scene with a virtual object, SceneGen samples possible positions and orientations in a scene, building updated Scene Graphs for each sample. We estimate the probability of each sample and place an object at the most likely pose. SceneGen also shares a heat map of the likelihood of each sample to suggest alternate high probability placements. This can be repeated to augment multiple virtual objects.

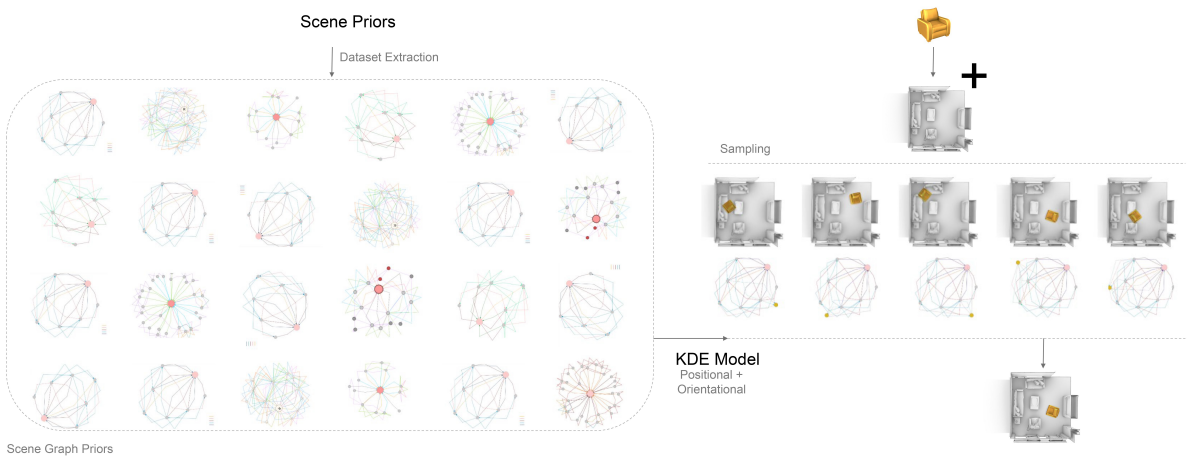


Figure 4.1: End-to-end workflow of SceneGen shows the main modules of our framework to augment rooms with virtual objects. Left: the training procedure including scene prior processing for the Knowledge Model creation. Right: the test time procedure of sampling and prediction.

Our implementation of SceneGen is built using data extracted from the Matterport3D dataset as priors and is detailed in Section4.5. As using object scans results in unoriented bounding boxes in Matterport3D, we develop an application to facilitate the labeling of the facing direction of each object.

We assess the effectiveness of SceneGen in Sections 4.6 and 4.7 for eight categories of objects across several types of rooms including bedroom, living room, hallway, and kitchen. In order to understand the effectiveness of each relationship in predicting where and how a new object should be placed, we run a series of ablation tests on each feature. We use K-fold cross-validation to partition the Matterport3D dataset, building the Knowledge Model on a training set and assessing how well the model can replace removed objects from a validation set. Additionally, we carry out a user study to analyze how SceneGen compares with random placements and the reference scene in placing new objects into virtual rooms and to evaluate the value of a heat map showing the probability of all samples.

Finally, Section 4.8 details an AR mobile application that we have developed to demonstrate the user experience when employing SceneGen to add new virtual objects. This application locally computes the semantic segmentation and generates a Scene Graph before estimating sample probabilities on an external server, and then parses and visualizes the prediction results.

4.3 Scene Representation

Graph Representation based on Extracted Features

In this section, we introduce a novel spatial Scene Graph that represents a room and objects in it as a graph using extracted spatial features. A Scene Graph $\mathscr{G}$ is defined by nodes representing objects, object groups, and the room, and by its edges representing the spatial relationships between the nodes. While various objects hold different individual functions (e.g., a chair to sit, a table to dine, etc.), their combinations and topological relationships tend to generate the main functional purpose of the space. In other words, spatial functions are created by the pairwise topologies of objects and their relationships with the room. In our proposed Scene Graph representation, we intend to explicitly extract a wide variety of positional and orientational relationships that can be present between objects. We model descriptive topologies that are commonly utilized by architects and interior designers to generate spatial functionalities in a given space. Therefore, our Scene Graph representation can also be described as a function map, where objects (nodes) and their relationships (edges) correspond to a single or multiple spatial functionalities in a scene. Figure 4.2 illustrates two examples of our Scene Graph representation, where a subset of topological features are visualized in the graph.

Definitions for Room and Objects

We consider a room or a scene in 3D space where its floor is on the flat (x,y) plane and the z-axis is orthogonal to the (x,y) plane. In this orientation, we denote the room space in a floorplan representation as R, namely, an orthographic projection of its 3D geometry plus a possible adjacency relationship that objects in R may overlap on the (x,y) plane but on top of one another along the z-axis. Specifically, the "support" relationship is defined in Section 6.2. This can also be viewed as a 2.5D representation of the space.

Further denote the k-th object (e.g., a bed or a table) in R as O_k. The collection of all n objects in R is denoted as $\mathscr{O} = \{O_1, O_2, ...O_n\}$. $B(O_k)$ represents the bounding box of the object O_k. $\dot{O}_k$ represents the center of the object O_k. Every object O_k has a label to classify its type. Related to the same R, we also have a set of groups $G = \{g_1, ..., g_m\}$, where each group g_i contains all objects of the same type within R.

Furthermore, each O_k has a primary axis a_k and a secondary axis b_k. For Frontal Facing objects, a_k represents the orientation of the object. a_k and b_k are both unit vectors such that b_k is a $\frac{\pi}{2}$ radian counter clockwise rotation of a_k. We define θ_{a_k} and θ_{b_k} to be the angle in radians represented by a_k and b_k respectively.

For each room R, we define $\mathscr{W} = \{W_1, W_2, ..., W_l\}$ where each W_k is a wall of the l-sided room. In the floorplan representation, W_k is represented by a 1D line segment. We also introduce a distance function $\delta(a,b)$ as the shortest distance between a and b objects. For example, $\delta(B(O_k), \dot{R})$ is the shortest distance between the bounding box of O_k and the center of the room R.

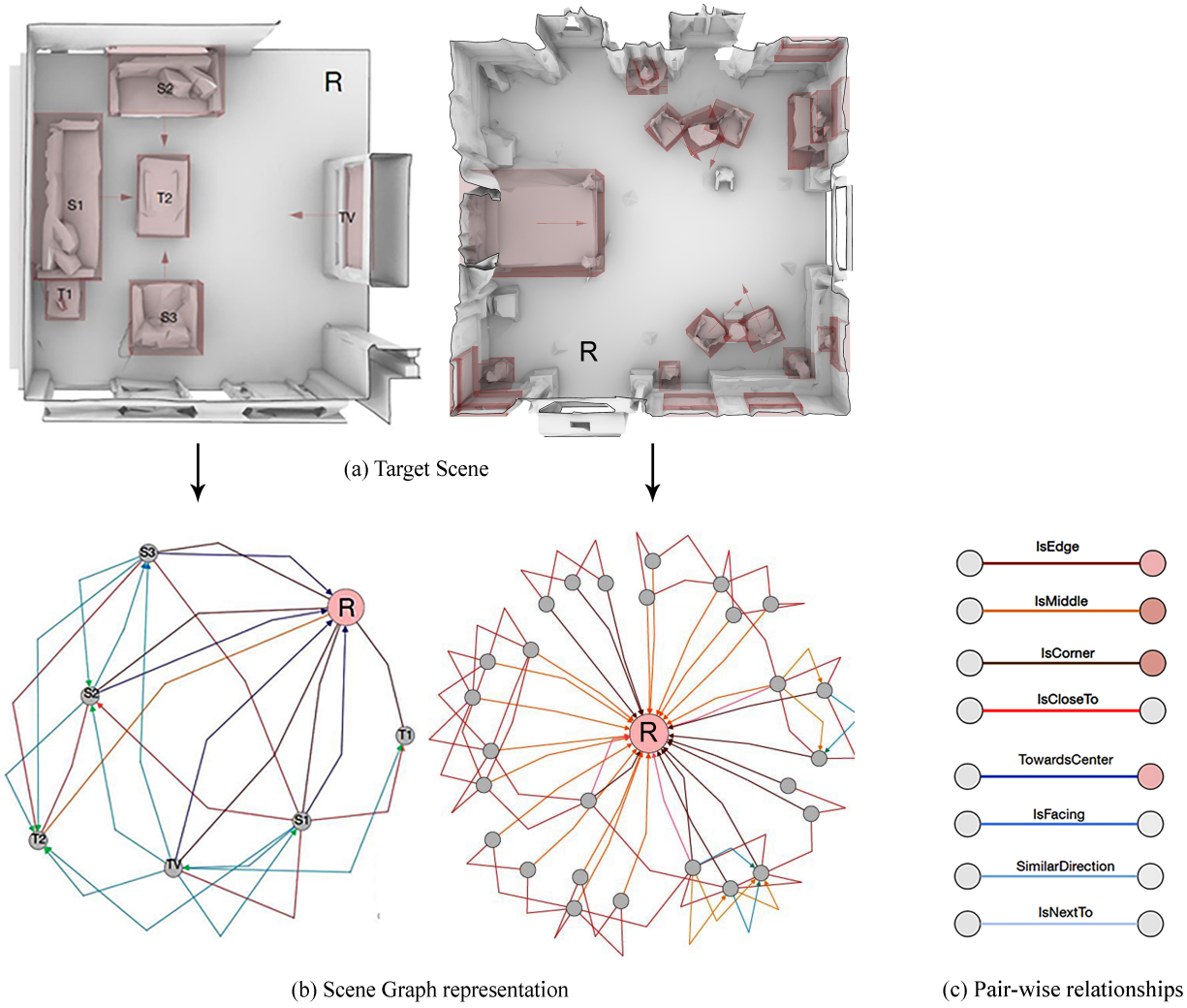


Figure 4.2: Our proposed Scene Graph representation is extracted from each scene capturing orientation and position based relationships between objects in a scene (pairwise) and between objects and the room itself. Visualization shows only a subset of features for clarity.

Positional Relationships

We first introduce features for objects based on their spatial positions in a scene. We include both pairwise relationships between objects (e.g., between a chair and a desk), object groups (e.g., between a dining table and dining chairs), and relationships between an object and the room.

Object to Room Relationships

RoomPosition: The room position feature of an object denotes whether an object is at the middle, edge, or corner of a room. This is based on how many walls an object is less than ρ distance from:

$$\text{RoomPosition}\,(O_k,R) = \sum_{W_i \in (W)} \mathbb{1}(\delta(B(O_k,W_i) < \rho). \tag{4.1}$$

In other words, if RoomPosition$(O_k,R) \geq 2$, the object is near at least two walls of a room and hence is near a *corner* of the room; if RoomPosition$(O_k,R) = 1$, the object is near only one wall of the room and is at the *edge* of the room; otherwise, the object is not near any wall and is in the *middle* of the room.

Object to Object Group Relationships

AverageDistance: For each object, and each group of objects we calculate the average distance between that object and all objects within that group. For cases where the object is a member of the group, we do not count the distance between the object in question and itself in the average.

$$\text{AverageDistance}(O_k,g_i) = \sum_{\substack{O_j \in g_i \\ j \neq k}} \delta(B(O_k),B(O_j))/\sum_{\substack{O_j \in g_i \\ j \neq k}} 1. \tag{4.2}$$

SurroundedBy: For each object, and each group of objects, we compute how many objects in the group are within a distance ε of the object. For cases where the object is a member of the group, we do not count the object in question.

$$\text{SurroundedBy}\,(O_k,g_i) = \sum_{\substack{O_j \in g_i \\ j \neq k}} \mathbb{1}(\delta\left(B(O_j),B(O_k)\right) < \varepsilon\right). \tag{4.3}$$

Object Support Relationships

Support: An object is considered to be supported by a group if it is directly on top of an object from the group or supports a group if it is directly underneath an object from the group.

$$\text{Support}\,(O_k, g_i) = \begin{cases} 1 & \exists O_j \in g_i \text{ where } O_k \text{ is on top of } O_j; \\ -1 & \exists O_j \in g_i \text{ where } O_k \text{ is under } O_j; \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

Orientation Relationships

We categorize the objects in our scenes into three main groups:

1. G_{omn}: Omnidirectional objects such as coffee tables and house plants that have no clear front-facing direction;
2. G_{frf}: Frontal Facing objects such as beds and chairs that can be oriented to face in a specific direction;
3. G_{in}: Inside Facing objects such as paintings and storage that are always facing opposite to the wall of the room where they are situated.

In this section, we discuss features applicable to objects with defined facing directions and not to omnidirectional objects.

Object to Room Relationships

We first define an indicator equation that is 1 if a ray extending from the center in the direction d_k of an object intersects a wall W_i:

$$f(\dot{O}_k, d_k, W_i) = \mathbb{1}(\exists \gamma \geq 0 | \dot{O}_k + \gamma d_k \in W_i). \tag{4.5}$$

TowardsCenter: An object is considered to be facing towards the center of the room, if an ray extending from the center of the object intersects one of the furthest $\frac{l}{2}$ walls from the object:

$$\begin{aligned} c_1 &= \underset{W_i \in (W)}{\text{argmax}}\, \delta(\dot{O}_k, W_i); \\ c_2 &= \underset{W_i \in (W \setminus c_1)}{\text{argmax}}\, \delta(\dot{O}_k, W_i); \\ &\ldots \\ c_{\frac{l}{2}} &= \underset{W_i \in (W \setminus c_1 \ldots c_{\frac{l}{2}-1})}{\text{argmax}}\, \delta(\dot{O}_k, W_i). \end{aligned} \tag{4.6}$$

$$\text{TowardsCenter}(O_k) = f(\dot{O}_k, a_k, c_1) \vee \ldots \vee \text{f}(\dot{O}_k, a_k, c_{\frac{l}{2}-1}). \tag{4.7}$$

AwayFromWall: An object is considered facing away from a wall if it is oriented away from and is normal to the closest wall to the object:

$$\begin{aligned} c_1 &= \underset{W_i \in (W)}{\text{argmin}}\, \delta\left(B(O_k), W_i\right); \\ \text{AwayFromWall}(O_k) &= \text{f}(\dot{O}_k, -a_k, c_1) \wedge (a_k \perp c_i). \end{aligned} \tag{4.8}$$

DirectionSimilarity: An object has a similar direction as one or more objects within a constant ε distance from the object if the other objects are facing in the same direction or in the opposite direction (π radians apart) from the first object subject to some small angular error φ:

$$\begin{aligned}
\text{Same}\,(O_k) &= \sum_{\substack{O_j \in \mathscr{O}, j \neq k \\ \delta\left(B(O_k), B(O_j)\right) \leq \varepsilon}} \mathbb{1}(|\theta_{a_k} - \theta_{a_j}| \leq \varphi), \\
\text{Opp}\,(O_k) &= \sum_{\substack{O_j \in \mathscr{O}, j \neq k \\ \delta\left(B(O_k), B(O_j)\right) \leq \varepsilon}} \mathbb{1}(|\pi - |\theta_{a_k} - \theta_{a_j}|| \leq \varphi), \\
\text{DirectionSimilarity}(O_k) &= [\text{Same}(O_k), \text{Opp}(O_k)] \in \mathbb{R}^2.
\end{aligned} \tag{4.9}$$

Object to Object Group Relationships

We first define an indicator function that is 1 if a ray extending from the center of the object in direction d_k intersects the bounding box of a second object:

$$h(\dot{O}_k, d_k, B(O_j)) = \mathbb{1}(\exists \gamma \geq 0 | \dot{O}_k + \gamma d_k \in B(O_j)). \tag{4.10}$$

Facing: Between an object and a group of objects we count how many objects of the group are within a distance ε of the object and are in the direction of the primary axis of the first object:

$$\text{Facing}(O_k, g_i) = \sum_{\substack{O_j \in g_i, j \neq k \\ \delta(B(O_k), B(O_j)) \leq \varepsilon}} h(\dot{O}_k, a_k, B(O_j)). \tag{4.11}$$

NextTo: Between an object and a group of object we count how many objects of the group are within a distance ε of the object and are in the direction of the positive or negative secondary axis of the first object:

$$\text{NextTo}\,(O_k, g_i) = \sum_{\substack{O_j \in g_i, j \neq k \\ \delta\left(B(O_k), B(O_j)\right) \leq \varepsilon}} h\left(\dot{O}_k, \pm b_k, B(O_j)\right). \tag{4.12}$$

4.4 Knowledge Model

Feature Vectors for Position and Orientation

To evaluate the plausibility of a new arrangement, we compare its corresponding Scene Graph with a population of viable Scene Graphs priors. By extracting Scene Graphs from a corpus of rooms, we construct a Knowledge Model, which serves as our spatial priors for the position and orientation relationships of each object group. For each object instance, we assemble a data vector for positional features from $\mathscr{G}$. For Frontal Facing objects, we similarly create a data vector for

orientational features. First, we define the following that represents an object's relationships with all groups G = $\{g_1, ..., g_m\}$:

$$\begin{aligned} \mathrm{AD}(O_k) &= [\mathrm{AverageDistance}(O_k, g_i) | i = 1, \cdots, m] \in \mathbb{R}^m, \\ \mathrm{S}(O_k) &= [\mathrm{SurroundedBy}(O_k, g_i) | i = 1, \cdots, m] \in \mathbb{R}^m, \\ \mathrm{F}(O_k) &= [\mathrm{Facing}(O_k, g_i) | i = 1, \cdots, m] \in \mathbb{R}^m, \\ \mathrm{NT}(O_k) &= [\mathrm{NextTo}(O_k, g_i) | i = 1, \cdots, m] \in \mathbb{R}^m, \\ \mathrm{SP}(O_k) &= [\mathrm{Support}(O_k, g_i) | i = 1, \cdots, m] \in \mathbb{R}^m. \end{aligned} \tag{4.13}$$

This allows us to construct data arrays, $d_p(O_k)$ and $d_o(O_k)$, containing features that relate to the position and orientation of an objects respectively. RoomPosition is also included in the data array for orientational features, d_o, since the other features of d_o are strongly correlated with an object's position in the room. This is abbreviated as RP. We also abbreviate TowardsCenter as TC and DirectionSimilarity as DS. For succinctness, when using these abbreviations for our features, the parameter O_k is dropped:

$$\begin{aligned} d_p(O_k) &= [\mathrm{RP} \in \mathbb{R}, \mathrm{AD} \in \mathbb{R}^m, \mathrm{SP} \in \mathbb{R}^m, \mathrm{S} \in \mathbb{R}^m] \in \mathbb{R}^{3m+1}, \\ d_o(O_k) &= [\mathrm{RP} \in \mathbb{R}, \mathrm{TC} \in \mathbb{R}, \mathrm{DS} \in \mathbb{R}^2, \mathrm{F} \in \mathbb{R}^m, \mathrm{NT} \in \mathbb{R}^m] \in \mathbb{R}^{2m+4}. \end{aligned} \tag{4.14}$$

Finally, given one feature vector per object for position and orientation, respectively, we can collect more samples from a database, which we will discuss in Section 4.5, to form our Knowledge Model. The model collects feature vectors separately with respect to different object types in multiple room spaces. To do so, we introduce $g_{i,j}$ to collect all of the i-th type objects in room $R_j, j = 1, \cdots, r$. Without loss of generality, we assume that the i-th object type is the same across all rooms. Therefore, we can collect all the objects of the same i-th type from a database as

$$g_{i,*} = \bigcup_{j=1}^{r} g_{i,j}.$$

Then $D_p(g_{i,*})$ and $D_o(g_{i,*})$ represent the collections of all feature vectors in (4.14) from objects in $g_{i,*}$:

$$\begin{aligned} \mathscr{D}_p(g_{i,*}) &= \{d_p(O_k) \,|\forall O_k \in g_{i,*}\}, \\ \mathscr{D}_o(g_{i,*}) &= \{d_o(O_k) \,|\forall O_k \in g_{i,*}\}. \end{aligned} \tag{4.15}$$

Scene Augmentation

Given the feature samples for the same type of object in (4.15), now we can estimate their likelihood distribution. In particular, given an object placement O of the i-th type, we seek to estimate the likelihood function for its position features:

$$P(d_p(O) | \mathscr{D}_p(g_{i,*})). \tag{4.16}$$

If O is Frontal Facing, we also seek to estimate the likelihood function for its orientation features:

$$P(d_o(O) | \mathscr{D}_o(g_{i,*})). \tag{4.17}$$

However, if O is an Inside Facing object, then with certainty, its orientation will be determined by that of its adjacent wall. Similarly, an Omnidirectional object O has no clear orientation. Therefore, for these categories of objects, estimation of their orientation likelihood is not needed. In this section, we discuss how to estimate (4.16) and (4.17)

We can approximate the shape of these distributions using multivariate kernel density estimation (KDE). Kernel density estimation is a non-parametric way to create a smooth function approximating the true distribution by summing kernel functions, K, placed at each observation $X_i...X_n$ [195]:

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right). \tag{4.18}$$

This allows us to estimate the probability distribution function (PDF) of the position and orientation relationships from the spatial priors in our Knowledge Model, $\mathscr{D}_p(g_{i,*}), \mathscr{D}_o(g_{i,*})$ for each group g_i.

SceneGen Algorithm

Algorithm 1 describes the SceneGen algorithm. Given a room model R and a set of existing objects $\mathscr{O} = \{O_1, O_2, ...O_n\}$, the algorithm evaluates the position and orientation likelihood of augmenting a new object O' and recommends its most likely poses.

Algorithm 1: SceneGen Algorithm

Given a training database, calculate $\mathscr{D}_p(g_{i,*})$ and $\mathscr{D}_o(g_{i,*})$ as prior.
For a given room R, construct the Scene Graph $\mathscr{G}$ of its objects $\mathscr{O}$.
while *Sample the position of O' of type i in R* **do**
  Calculate $P(d_p(O')|\mathscr{D}_p(g_{i,*}))$.
  while *Sample the orientation of $O' \in [0, 2\pi)$* **do**
    Calculate $P(d_o(O')|\mathscr{D}_o(g_{i,*}))$
  end
end
Generate a heat map displaying the likelihood distributions.
Make recommendation to place O' at the highest probability pose.

Figure 7.4 shows how potential scene graphs are created for sampled placements. For scenes where multiple objects need to be added, we repeat Algorithm 1 for each additional object.

4.5 Implementation

In this section, we discuss the implementation detail of SceneGen based on the relationship data learned from the Matterport3D dataset.

Dataset

Matterport3D [30] is a large-scale RGB-D dataset containing 90 building-scale scenes. The dataset consists of various building types with diverse architectural styles, including numerous spatial functionalities and furniture layouts. Annotations of building elements and furniture have been provided with surface reconstruction as well as 2D and 3D semantic segmentation.

Pose Standardization

In order to use the Matterport3D dataset as prior for SceneGen, we must make a few modifications to standardize object orientation using an annotation tool that we have also developed. In particular, different from Section 4.3, our annotation tool interacting with the dataset is fully in 3D environment (i.e., through Unity 3D). After the annotation, the relationship data are then consolidated back to the 2.5D representation, conforming to the computation of the SceneGen models.

For each object O_k, the Matterport3D dataset supplies labeled oriented 3D bounding boxes $B(O)$ aligned to the (x,y) plane. This is defined by a center position $\dot{O}$, primary axis a, secondary axis b, an implicit tertiary axis c, and $r \in \mathbb{R}^3$ denotes the radius vector of O. However, the Matterport3D dataset does not provide information about which labeled direction the object is facing or aligns with the z-axis. Hence, it will rely on our labeling tool to resolve the ambiguities.

To provide a consistent definition, we describe a scheme to label these axes such that the primary axis a points in the direction the object is facing as a^*. Since we know that only one of these three axes has a z component, we shall store this in the third axis c^* and define b^* to be orthogonal to a^* on the x,y plane. The box size r will also be updated to correspond to the correct axes. By constraining these aligned axes to be right-handed, for a given a^* we have:

$$c^* \doteq [0,0,1], \quad b^* \doteq c^* \times a^*. \tag{4.19}$$

In order to correctly relabel each object, we have developed an application to facilitate the identification of the correct primary axis for all the Frontal Facing objects and supplemented this to the updated data set. For each object, we view the house model mesh at different camera positions around the bounding box in order to determine the primary axis of the object as displayed in Figure 4.3. Our annotation tool shown in Figure 4.4 allows a labeler to select from two possible directions at each camera position or can move the camera clockwise or counterclockwise to get a better view. Once a selection is made, the orienting axis a^* can be determined. We then use (4.19) to standardize the axes. Using the annotation tool, the average time for each object to be labeled is 2.6 seconds. For example, if a MatterPort3D house has 80 Frontal Facing objects that need to be labeled, it would take an estimate of only 3.5 minutes for the annotation task of the house. a

Category Reduction

For this study, we have reduced the categories of object types considered for building our model and placing new objects. Though the Matterport3D dataset includes many different types of furniture,

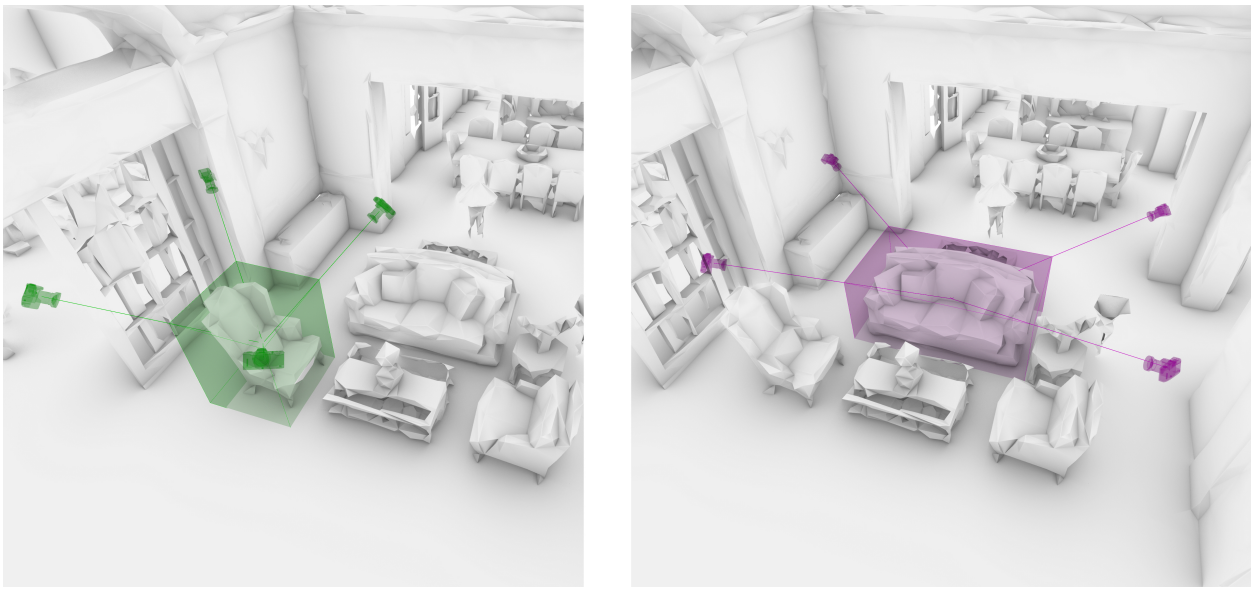


Figure 4.3: In our annotation tool, a camera is orbited around each object to facilitate labeling of object orientations.

organized with room labels to describe furniture function (e.g., "dining chair" versus "office chair"), we found that the dataset has a limited amount of instances for many object categories. Because we build statistical models for each object category, we require an adequate representation of each category. Thus, we reduce the categories to a sufficiently represented subset and filter objects that do not fall in these categories for the purposes of this study.

We group the objects into 8 coarse categories: $G = \{$Bed, Chair, Decor, Picture, Sofa, Storage, Table, TV$\}$. Each of these categories has a specific type of orientation, as described in Section 4.3. Of these categories, Frontal Facing objects are $G_{\text{frf}} = \{$Bed, Chair, Sofa, TV$\}$, Omnidirectional objects are $G_{\text{omn}} = \{$Decor, Table$\}$, and Inside Facing objects are $G_{\text{in}} = \{$Picture, Storage$\}$.

For room types, we consider the set $\{$ library, living room, meeting room, TV room, bedroom, rec room, office, dining room, family room, kitchen, lounge$\}$ to avoid overly specialized rooms such as balconies, garages, and stairs. We also filter rooms that hold more than 95% unoccupied areas to avoid unusual empty rooms that come without any spatial arrangements. After the data reduction, a total of 1,326 rooms and 7,017 objects are in our training and validation sets. The object and room categories used can be expanded if sufficient data are available.

Knowledge Model

We use the processed dataset as prior to train the SceneGen Knowledge Model. The procedure first estimates each object O_k according to (4.14), and subsequently constructs $\mathscr{D}_p(g_{i,*})$ and $\mathscr{D}_o(g_{i,*})$

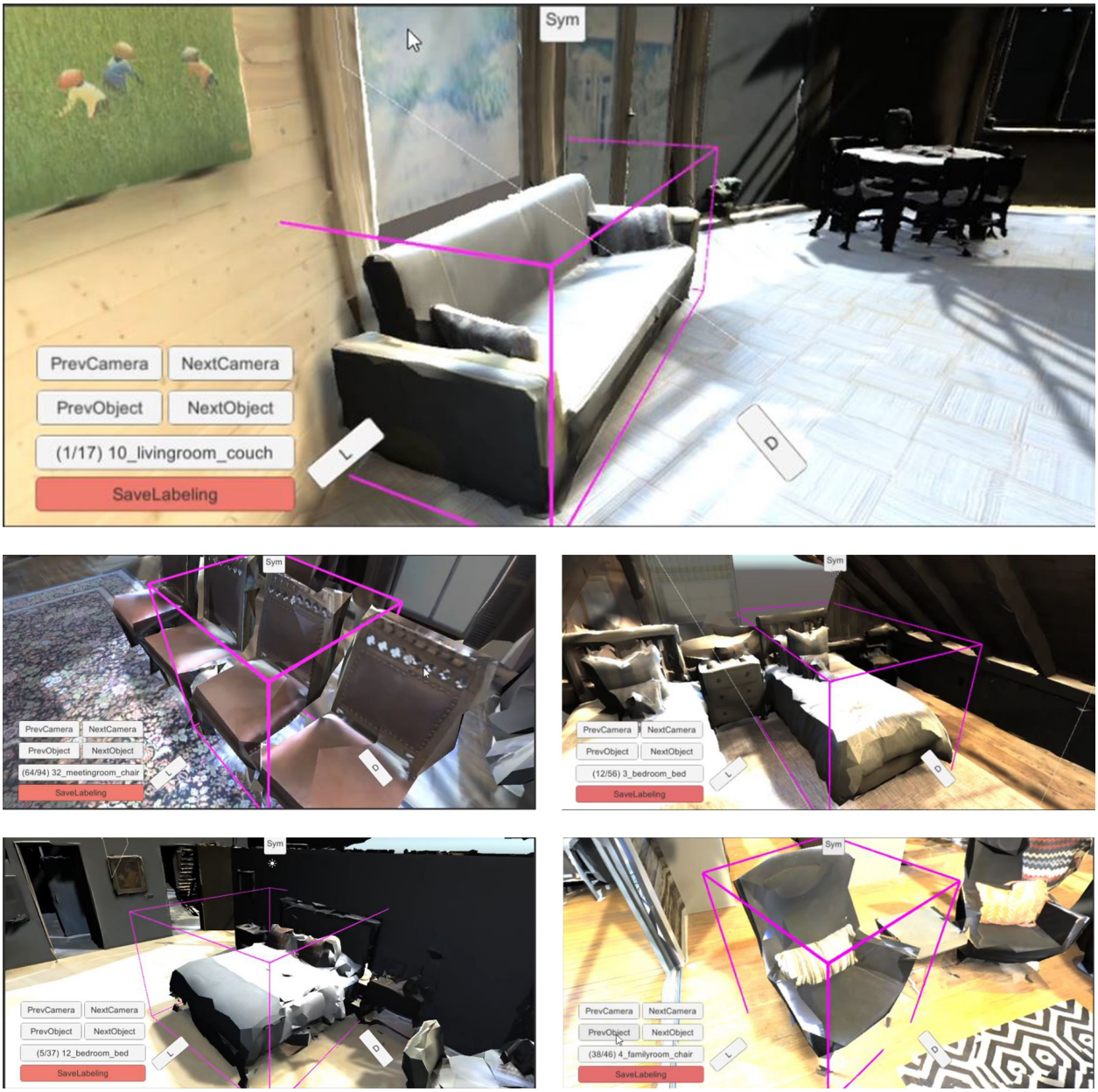


Figure 4.4: A labeler using our annotation tool can select which direction the object is facing or move to the next camera to get a better view. The selection is used to automatically standardize the axes of each object's bounding box.

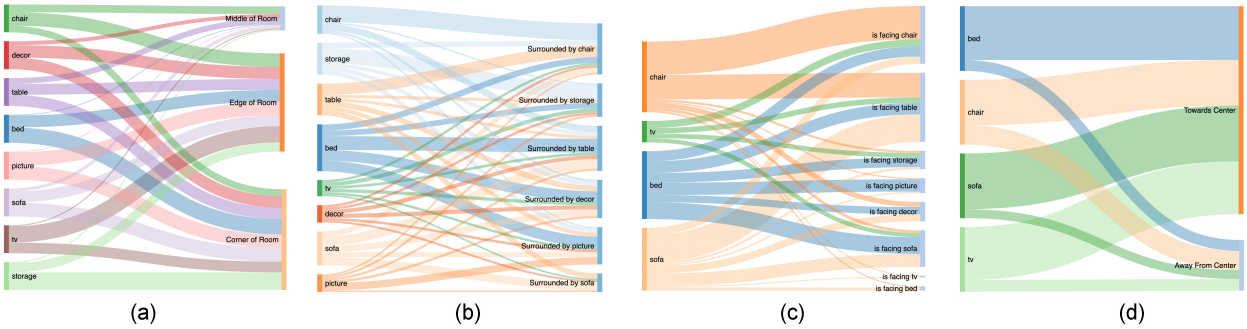


Figure 4.5: Visualization of the Knowledge Model built from Scene Graphs extracted from the Matterport3D Dataset shows for each group of objects: (a) frequency of each RoomPosition, (b) frequency the object is surrounded by multiple objects from another group, (c) frequency the object is facing an object from another group, (d) frequency the object is facing towards the center of the room or not.

in (4.15) for categories in G and G_{frf} respectively. Given our priors, we estimate the likelihood functions $P(d_p(O)|\mathscr{D}_p(g_{i,*}))$ and $P(d_o(O)|\mathscr{D}_p(g_{i,*}))$ from (4.16) and (4.17) using Kernel Density Estimation.

We utilize a KDE library developed by [189] with a normal reference rule of thumb bandwidth with ordered, discrete variable types. We make an exception for AverageDistance, which is continuous. When there are no objects of a certain group, g_i in a room, the value of AverageDistance(O_k, g_i) is set to a large constant (1000), and we use a manually tuned bandwidth (0.1) to reduce the impact of this on the rest of the distribution.

Furthermore, we found for this particular dataset, a subset of features, Facing, TowardsCenter, and RoomPosition, are most impactful in predicting orientation, as detailed in Section 4.7. Therefore, while we model all of the orientational features, we only use the Facing, TowardsCenter, and RoomPosition features for our implementation of SceneGen and in the User Studies. Finally, due to overlapping bounding boxes in the dataset, calculating object support relationships (SP) precisely is not possible. Thus in our implementation, we allow the certain natural overlaps defined heuristically instead of using these features. A visualization of our priors from the Matterport3D dataset can be seen in Figure 4.5.

We use Algorithm 1 to augment a room R with an object of type i and generate a probability heat map. This can be repeated in order to add multiple objects. To speed up computation in this implementation, we first sample positions and then sample orientations at the most probable position instead of sampling orientations at every possible position.

Figure 4.6 shows how our implementation of SceneGen adds a new object to a scene, and examples of scenes are augmented with multiple objects iteratively are shown in Figure 4.7. The illustrated heatmaps are color-coded based on their normalized probability rank. In positional visualizations, the top k of valid samples are rendered from zero opacity to a dark solid color (high

probability), while in orientational visualizations, angular samples are color-coded from red (low probability) to red (high probability).

Computation Time We train and evaluate our model using a machine with a 4-core Intel i7-4770HQ CPU and 16GB of RAM. In training, creating our Knowledge Model and estimating distributions for eight categories of objects takes approximately 12 seconds. In testing, it takes $\approx$2 seconds to extract a scene graph and generate a heat map indicating the probabilities of 250 sampled poses.

4.6 Experiments

Ablation Studies

To evaluate our prediction system, we run ablation studies, examining how the presence or absence of particular features affects our object position and orientation prediction results. We use a $K = 4$-fold cross-validation method in our ablation studies, with 100 rooms in each validation set and the remaining rooms in our training set.

Position Features Evaluation

The full position prediction model, SceneGen, trains three features: AverageDistance (AD), SurroundedBy (S), and RoomPosition (RP). The combination is denoted as AD+S+RP. We further consider three reduced versions of our system: AD+RP, using only AverageDistance and RoomPosition features; S+RP, using only Surrounding and RoomPosition features; and RP, solely using the RoomPosition feature.

We evaluate each system using the K-fold method described above. In this study, we remove each object in the validation set, one at a time, and use our model to predict where the removed object should be positioned. The orientation of the replaced object will be the same as the original. We compute the distance between the original object location and our system's prediction.

However, as inhabitants of actual rooms, we are aware that there is often more than one plausible placement of an object, though some may be more optimal than others. Thus, we raise the question of whether there is more than one ground truth or the correct answer for our object placement problem. Hence, in addition to validating our model's features, our first ablation study validates them in relation to the simple approach of taking the single highest-scored location from our system. Meanwhile, our second ablation study uses the top 5 highest-scored locations, opening up the examination to multiple potential "right answers. "

Orientation Features Evaluation

We run a similar experiment to evaluate our orientation prediction models for Frontal Facing objects. Our Scene Graphs capture five relationships based on the orientation of the objects: Facing

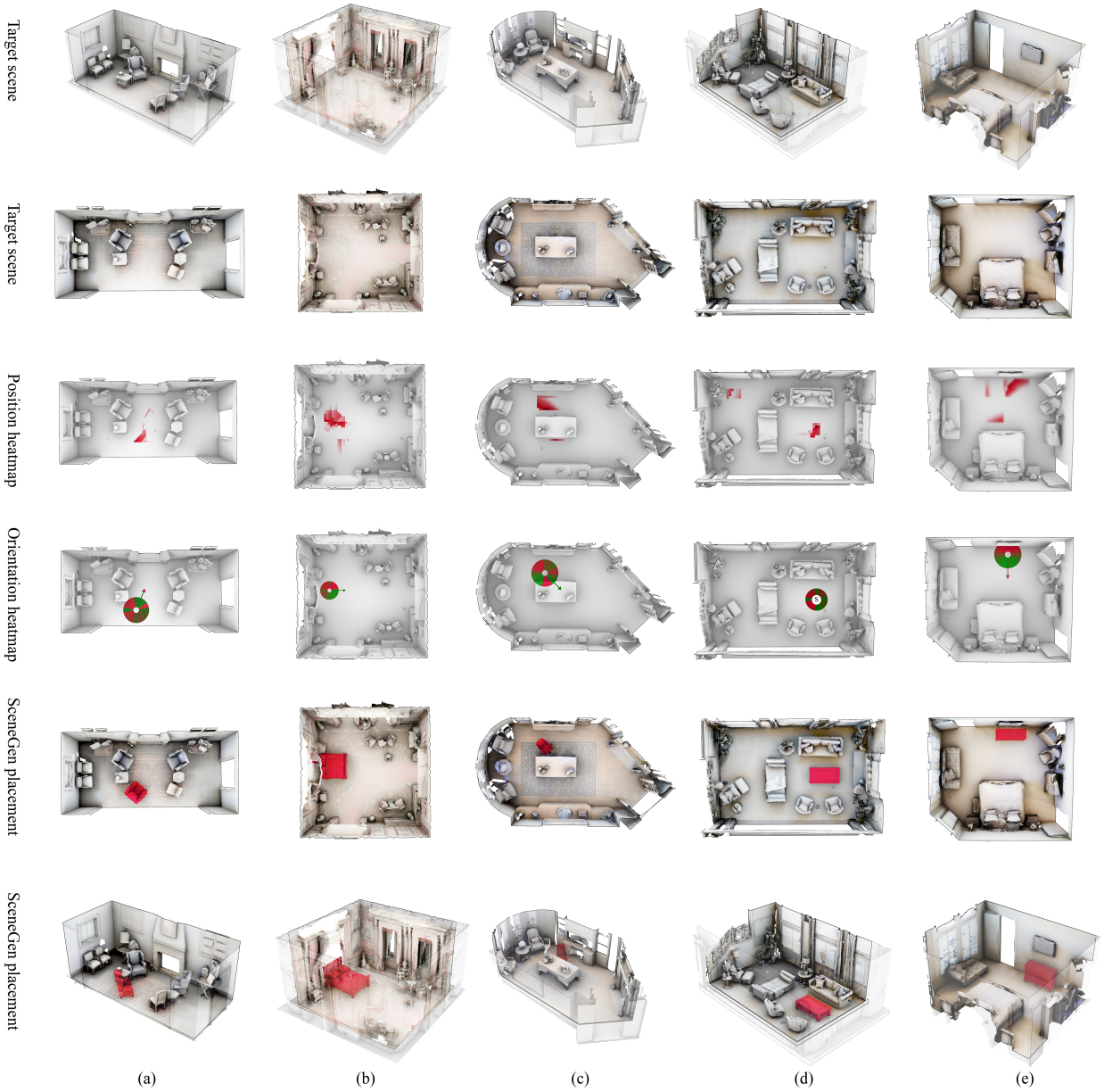


Figure 4.6: Scene Gen places objects into scenes by extracting a Scene Graph from each room, sampling positions and orientations to create probability maps, and then placing an object in the most probable pose. (a) A sofa placed in a living room, (b) a bed placed in a bedroom, (c) a chair placed in an office, (d) A table placed in a family room, and (e) storage placed in a bedroom.

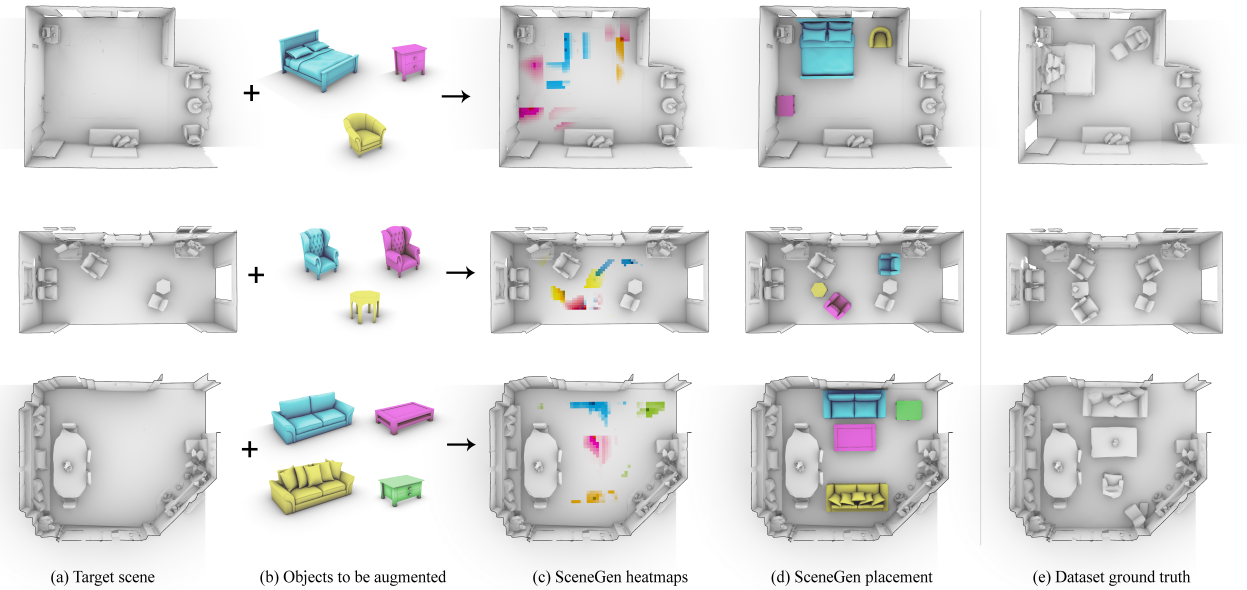


Figure 4.7: Examples of adding multiple virtual objects to a scene using SceneGen. Each object is placed in the most likely position and orientation iteratively into a partially decorated room. Top: A bed, storage, and sofa are first extracted from the room model, then reorganized in a viable alternative to the dataset ground truth; Middle: Two sofas and a table are reorganized by SceneGen to a living room in an arrangement similar to ground truth; Bottom: A sofa, a table are reorganized, and another sofa and a table are added to a family room, showing an augmented scene with new virtual objects compared to the ground truth.

(F), TowardsCenter (C), NextTo (NT), DirectionSimilarity (DS), and RoomPosition (RP). We assess models based on several combinations of these relationships.

We evaluate each of these models using the same K-fold approach, removing the orientation information of each object in the validation set and then using our system to predict the best orientation, keeping the object's position constant. We compare the angular difference between the predicted and the original orientations.

Comparative Studies

We compare the performance system of our system with SceneGraphNet [257] in both quantitative and qualitative experiments. The experiments are similar to the K-fold ablation study trained on the MatterPort3D dataset mentioned in the previous subsection, where we remove each object in the validation set dataset and compare the model's ability to predict where the removed object should be positioned. However, due to the fact that SceneGraphNet does not predict orientations of the object

placement, we only limit the experiment to positional calculations only. In our qualitative results, we used the original orientation of the ground truth as a basis for SceneGraphNet's augmentation.

SceneGraphNet utilizes a neural message-passing approach to the scene synthesis problem. However, the system is primarily designed to predict the probability distribution over the type of objects given a query position in a scene. To get the probability distribution for the placement of a specific object across a scene, their code was augmented with a sampling mechanism similar to SceneGen to evaluate all possible coordinates which fit in the room. Next, each coordinate was fed as an input to the SceneGraphNet procedure to predict the probability distribution for all categories at that position. Using this exhaustive search approach, we are able to calculate the most probable location to place a specific object category in the scene.

User Evaluation

We conduct user studies with a designed 3D application based on our prediction system to evaluate the plausibility of our predicted positions and the usefulness of our heat map system. We recruited 40 participants, of which 8 were trained architects. To ensure unbiased results, the participants were randomly divided into four groups. Each group of users was shown five scenes from each of the five levels for a total of 25 scenes. The order in which these scenes were presented was randomized for each user, and they were not told which level a scene was at.

We reconstructed 34 3D scenes from our dataset test split, where each scene had one object randomly removed. In this reconstruction process, we performed some simplification and regularized the furniture designs using prefabricated libraries so that users would evaluate the layout of the room rather than the design of the object itself while matching the placement and size of each object. An example of this scene reconstruction and simplification can be seen in Figure 4.12(a-b).

The five defined levels test different object placement methods as shown in Figure 4.12(c-g) to replace the removed object. Levels I and II are both random placements generated at run time for each user. The Level I system initially places the object in a random position and orientation in the scene. The Level II system places the object in an open random position and orientation, where the placement does not overlap with the room walls or other objects. Levels III and IV use SceneGen predictions. The Level III system places the object in the position and orientation predicted by SceneGen. Level IV also places the object in the predicted position and orientation but also overlays a probability map. The Level V system places the object at the position it appears in the Matterport3D dataset, i.e., the ground truth.

We recorded the users' Likert rating of the plausibility of the initial object placement on a scale of 1 to 5 (1 = implausible/random,3 = somewhat plausible, 5 = very plausible). We also recorded whether the user chose to adjust the initial placement, the Euclidean distance between the initial placement and the final user-chosen placement, and the orientation change between the initial orientation and the final user-chosen orientation. No comparison was made between the scenes by the participants, and the scores were to be given to each scene independent of another. We expect higher initial Likert ratings and smaller adjustments to position and orientation for levels initialized by our system than for levels initialized to random positions.

Each participant used an executable application on a desktop computer. The goal of the study was explained to the user, and they were shown a demonstration of how to use the interface. For each scene, the user was shown a 3D room and an object that was removed. After inspecting the initial scene and clicking "place object," the object was placed in the scene using the method corresponding to the level of the scene. In Level IV Scenes, the probability heat map was also visualized. Note that participants were not aware whether Level IV was generating results from our system, random, or ground truth. The user was shown multiple camera angles and was able to pan, zoom and orbit around the 3D room to evaluate the placement.

For each scene, the user was first asked to rate the plausibility of placement on a Likert Scale from 1-to 5. Following this, the user was asked if they wanted to move the object to a new location. If they answered "no," the user would progress to the next scene. If they answered "yes," the UI displayed transformation control handles (position axis arrows, rotation axis circles) to object position and orientation. After moving the object to the desired location, the user could save the placement and progress to the next scene. An IRB approval was maintained ahead of the experiment.

4.7 Results

Ablation & Comparative Studies

Position Features

In Figure 4.8, we plot the cumulative distance between the ground truth position and the top position prediction, and in Figure 4.9, we plot the cumulative distance between the ground truth position and the nearest out of the top 5 position predictions, using our full system and three ablated versions.

We find that the full SceneGen system predicts a placement most similar to the ground truth than any of the ablated versions, followed by the models using AverageDist and RoomPosition features (AD+RP) and SurroundedBy and RoomPosition (S+RP). The predictions furthest from the ground truth are generated by only using the RoomPosition (RP) feature. These curves are consistent between the best and the closest of the top 5 predicted positions, and the fact indicates that each of our features for position prediction contributes to the accuracy of the final result.

In addition, when the top 5 predictions are considered, we see that each system we assess is able to identify high probability zones closer to the ground truth compared to only using the best prediction. This is supported by the slope of the curves in Figure 4.9, which rise much more sharply than in Figure 4.8. This difference provides support for the importance of predicting multiple locations instead of simply returning to the highest-scored locations. A room can contain multiple plausible locations for a new object, so the system's predicted location with the highest score may not necessarily be the same as the ground truth. For this reason, our system returns probabilities across sampled positions using a heat map to show multiple viable predictions for any placement query.

Table 4.1 shows the mean distance of the position prediction to ground truth position separated by object categories in all SceneGen ablation and also SceneGraphNet. We find that the object

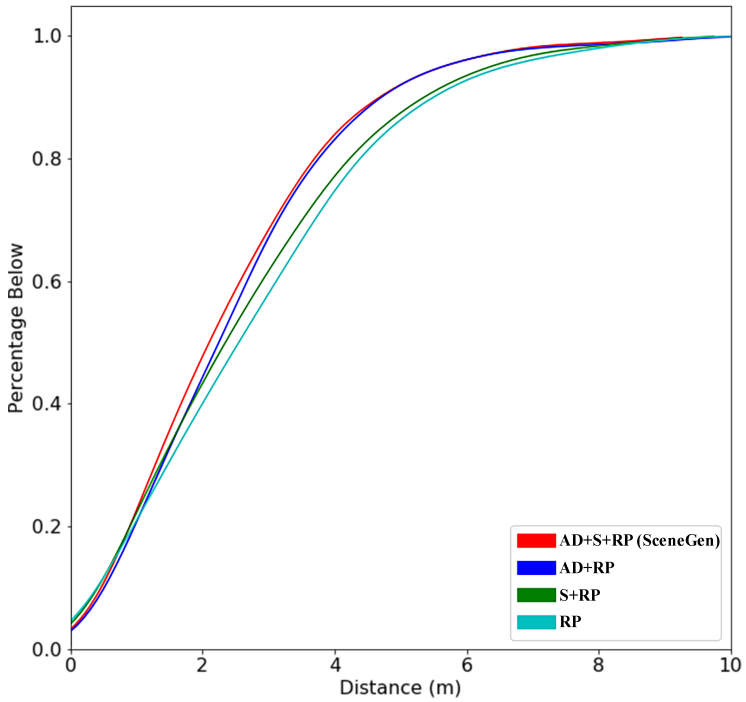


Figure 4.8: Distance between a ground truth object's position and where SceneGen and other ablated versions of our system predict the object should be re-positioned is shown in a cumulative density plot.

categories where the full SceneGen system outperforms its ablations are chairs, storage, and decor. For beds and TVs, SceneGen only produces the closest placements out of the system versions when considering the top five predictions. For pictures and tables, SceneGen's top prediction is closest to ground truth and is only slightly further when comparing the nearest of the top 5 predictions.

Furthermore, our results indicate SceneGen outperforms SceneGraphNet in all categories in the positional placement experiment. Figure 4.10 illustrates a qualitative comparison of the object augmentation tasks between the two systems on MatterPort3D scenes. While both systems show their ability to predict plausible placement in relation to other objects in the target scene, we observe a slightly better performance in SceneGen when taking into account the object's position in relation to the room.

Orientation Features Results

In our orientational ablation studies, we assess the ability of various versions of our model to reorient Frontal Facing objects from test scenes. In Figure 4.11, we plot the angular difference

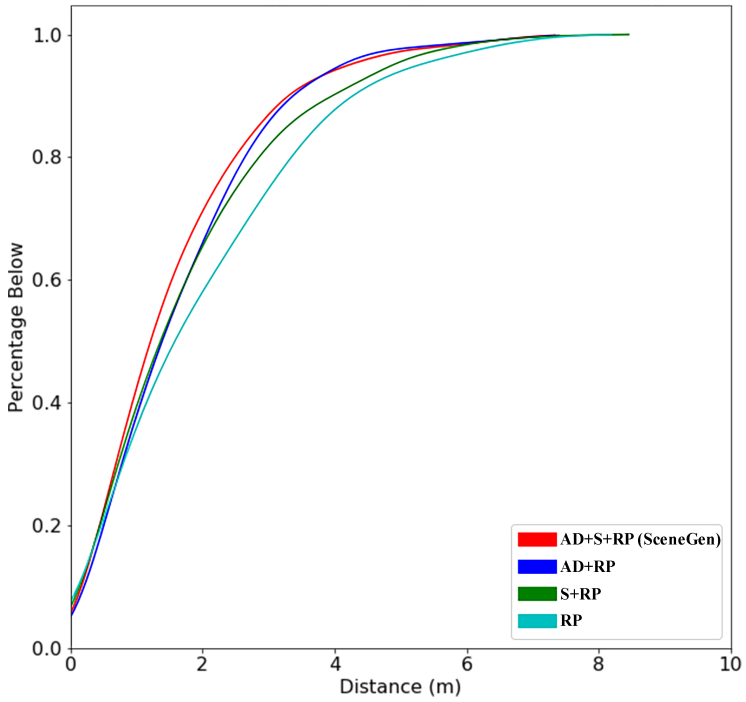


Figure 4.9: Distance between the ground truth object's position and the nearest of the 5 highest probability positions predicted by SceneGen and other ablated versions of our system is shown in a cumulative density plot.

Table 4.1: Distance between ground truth and predicted position for different models, with smallest distances for each object type in bold (ablation study). Topology features are abbreviated as follows: AverageDistance as AD, SurroundedBy as S, and RoomPosition as RP.

System	*Bed*		*Chair*		*Storage*		*Decor*		*Picture*		*Table*		*Sofa*		*TV*		*Overall*	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5
AD+S+RP (SceneGen)	1.58	**0.87**	**2.26**	**1.35**	**2.27**	**1.45**	**2.71**	**1.71**	**2.80**	1.99	**2.15**	1.47	2.56	1.58	2.49	**1.52**	**2.40**	**1.54**
AD + RP	**1.40**	0.95	2.40	1.47	2.55	1.67	2.79	1.96	2.95	2.03	2.26	**1.46**	2.58	**1.58**	**2.39**	1.731	2.49	1.65
S + RP	1.85	1.32	2.46	1.56	2.46	1.64	3.38	2.14	2.82	**1.92**	2.67	1.72	**2.53**	1.64	2.51	1.55	2.67	1.73
RP	1.99	1.31	2.95	2.31	2.75	1.53	3.12	2.56	2.95	2.21	2.70	1.57	2.55	1.72	2.95	2.32	2.80	1.96
SceneGraphNet [257]	1.91	1.56	3.01	2.49	2.37	1.95	3.14	2.70	3.36	2.94	3.80	3.31	3.57	3.12	3.97	3.40	3.25	2.80

Table 4.2: Angular difference in radians between ground truth and predicted orientation for different model architectures (ablation study). Topology features are abbreviated as follows: Facing as F, TowardsCenter as C, RoomPosition as (RP), NextTo as NT, DirectionSimilarity as DS.

System	*Bed*	*Chair*	*Sofa*	*TV*	*Overall*
F+C+RP (SceneGen)	**0.65**	**0.98**	**0.67**	0.66	**0.85**
F only	1.13	1.66	1.51	0.91	1.54
F+C	1.13	1.55	1.18	0.49	1.35
F+C+NT	1.18	1.53	1.23	**0.46**	1.35
F+C+DS	1.54	1.55	1.21	0.59	1.39
F+C+DS+NT	1.22	1.50	1.23	0.63	1.35

between the ground truth orientation and the top orientation prediction from various versions of our system. The base model includes only Facing (F) and is the lowest-performing. We find that the system that also includes TowardsCenter and RoomPosition features performs best overall. We use this system (F+C+RP) in our implementation of SceneGen. The other four versions of our system perform similarly to each other overall.

Table 4.2 shows the results of the orientation ablation study separated by object category. In this case, the system with Facing, TowardsCenter, and RoomPosition features (F+C+RP) outperform all other versions across all categories except for TVs, where the system that includes Facing, TowardsCenter, and NextTo (F+C+NT) produces the least deviation. In fact, all three of the systems that included either DirectionSimilarity or NextTo, predict the orientation of TVs more closely than the overall best performing system but perform more poorly on other objects such as beds when compared with systems without those features. This suggests that for other datasets, these features could be more effective in prediction orientations.

User Study Results

Plausibility of Placement Results

We show the distributions of Likert ratings by levels in Figure 4.13. We also run a one-way ANOVA test on the Likert ratings of initial placements, finding significant differences between all pairs of levels except for Levels IV and V. In other words, the ratings for Level IV's representation of our prediction system are not significantly different from ground truth placements. Across multiple tests, we see that Level IV result ratings are significantly different from levels based on randomization, while those from Level III are not as significant. The difference between Levels III and IV could support our conjecture that accounting for multiple "right answer" placements improves the predictions.

(a) SceneGraphNet b) SceneGen (Ours)

Figure 4.10: Comparison between SceneGraphNet [257] (left/yellow) and our proposed system (right/red) for the scene augmentation task on example MatterPort3D scenes. Objects are removed and augmented back into the scene via the constrained scene augmentation models. Illustration includes augmentation comparison of a bed (top), sofa+ table (middle) in an office, and a storage (bottom).

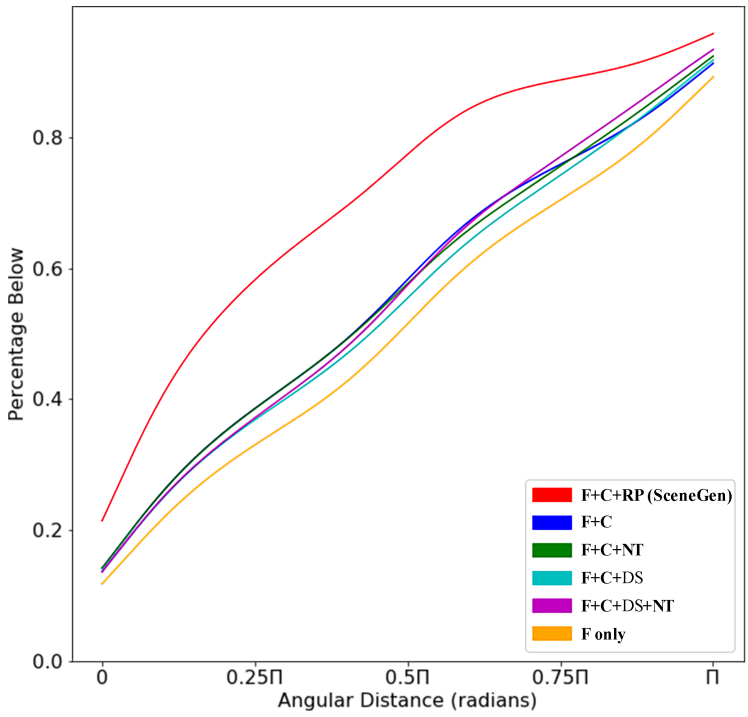


Figure 4.11: Cumulative density plot indicates angular difference between ground truth orientation and our system's predicted orientation for SceneGen and other subsets of orientation features. The range is $[0, \pi)$.

Position Prediction Results

We analyze how participants' choices to adjust placement vary across different scene levels. Results of this can be seen in Figure 4.14. A one-way ANOVA test of the distance users moving objects from their placements finds a significant difference ($p = 1.8622e^{38}$) between two groupings of levels: 1) Levels I and II (with higher means), and 2) Levels III, IV, and V (with lower means). The differentiation in groupings supports the plausibility of our system's position prediction over random placements.

Orientation Prediction Results

A one-way ANOVA test is also performed on the change in object orientation from the participants' manual adjustment and finds a significant difference ($p = 1.8112e^{16}$) between a different pair of level groupings: 1) Levels I, II, and III, and 2) Levels IV and V. In Figure 4.15, we show the distributions of the angular difference between the initial object orientation and the final user-chosen

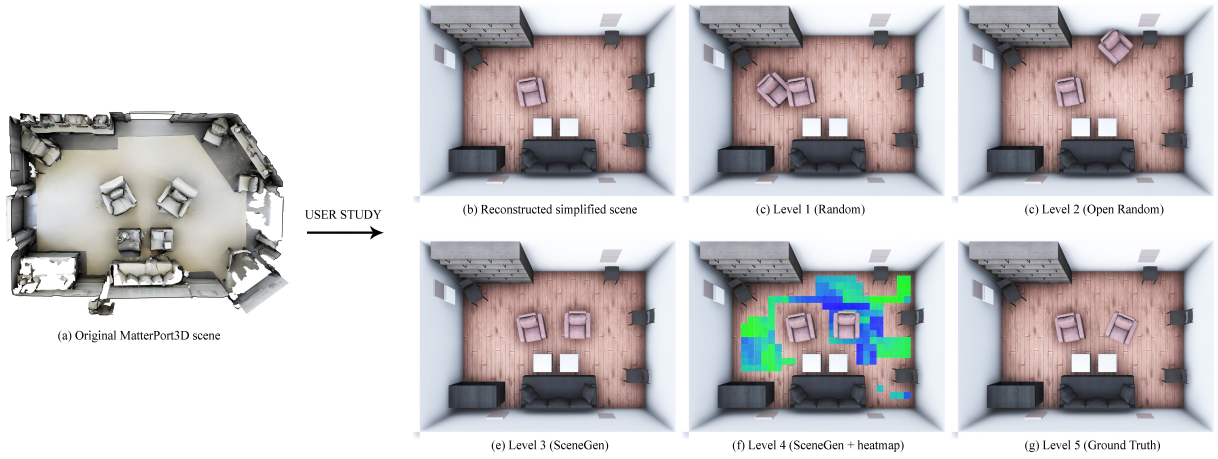


Figure 4.12: Users are shown scene models that are simplified based on original Matterport3D rooms. An object is reorganized using each of the five levels of the systems. Level I places the object randomly in the room. Level II replaces the object randomly in an open space. Levels III and IV use SceneGen to predict the most likely placement and orientation, and Level IV further shows a heat map visualizing the underlying probability score at each sampled position. In Level V, the user sees the original placement in the ground truth. When providing scores during the experiment, the user has multiple camera angles available and is able to pan, zoom, and orbit around the room to evaluate the placement.

orientation, for each level. Levels IV and V have distributions that are most concentrated at no rotation by the user. In Levels I and II, the users rotate objects more than half of the time, with an average rotation greater than $\frac{\pi}{6}$ radians. A vast majority of objects placed by Levels III, IV, and V systems are not rotated by the user, lending support to the validity of our prediction system.

4.8 Augmented Reality Application

To demonstrate a way to integrate our prediction system in action, we have implemented an AR application that augments a scene using SceneGen. Users can overlay bounding boxes over the existing furniture to see the object bounds used in our predictions. On inserting a new object into the scene, the user can visualize a portability map to observe potential positions. Our AR application consists of five main modules: (i) local semantic segmentation of the room; (ii) local Scene Graph generation, (iii) heat map generation, which is developed on an external server, (iv) local data parsing and visualization, and finally (v) the user interface. A brief demonstration of the AR application's interface and workflow is shown in Figure 4.16.

Semantic segmentation of the room can be done either manually or automatically, using inte-

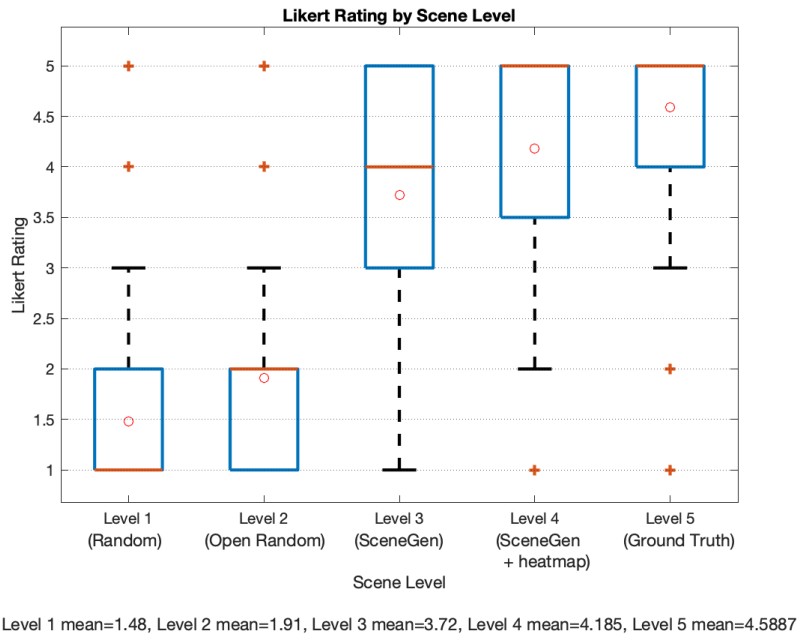


Figure 4.13: Users rank the plausibility of object placement averaged on the Likert Scale from 1 to 5. (1= Implausible/ Random, 3= Somewhat Plausible, 5 = Very Plausible). Scores are displayed in a box plot separated by the user study level.

grated tools available on AR devices. However, as not all current AR devices are equipped with depth-sensing capturing hardware, we use techniques previously introduced by [183], allowing the user themselves to manually generate and annotate semantic bounding boxes of objects of the target scene. The data acquired are then converted to our proposed spatial Scene Graph, resulting in an abstract representation of the scene. Both semantic segmentation and graph generation modules are performed locally on the AR devices, ensuring the privacy of the raw spatial data of the user.

Once the Scene Graph is generated, it is sent to a remote server where the SceneGen engine can calculate positional and orientation augmentation probability maps for all object categories for the target scene. Such an approach would allow faster computation time since current AR devices come with limited computational and memory resources. The results are sent back to the local device, which can be parsed and visualized using the AR GUI. For simplicity, we limit the positional sampling to only on the floor. However, based on the object category, if an object is already present in that location, the probability map will render on top of the object to maintain a clearer visualization of the probability map.

The instantiation system can toggle between two modes: *Manual* and *SceneGen*. In Manual mode, the object is placed in front of the user at the intersection of the camera's front-facing vector

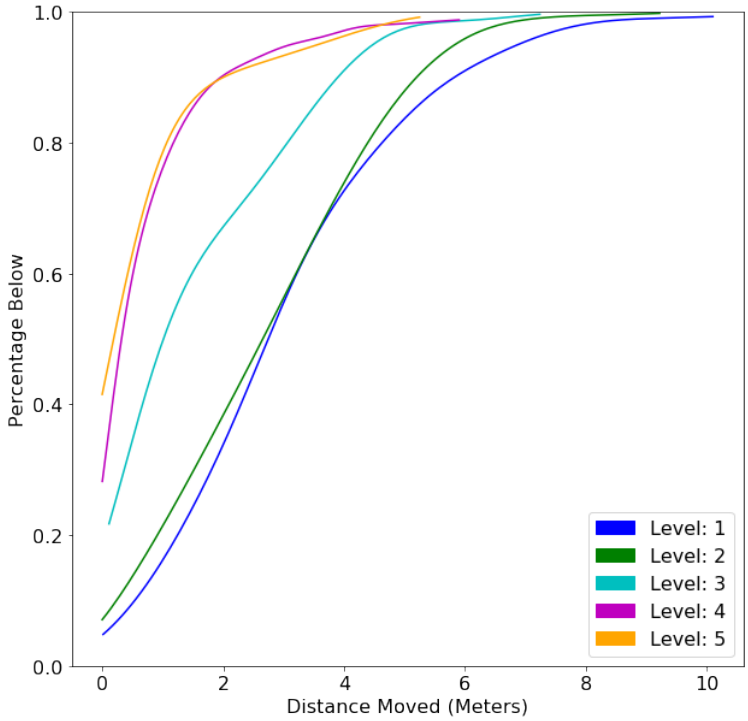


Figure 4.14: Cumulative density plot indicates the distance an object is moved from its predicted placement in each level by users.

direction with the floor. This would normally result in augmenting the object in the middle of the screen. While such a conventional approach allows the user to control the initial placement by determining the pose of the AR camera, in many cases, additional movements are necessary to place the object in a plausible final location. In such cases, the user can then further move and rotate the objects to their desired location. In SceneGen mode, the virtual object is augmented using the prediction of our system, resulting in faster and contextual placements.

4.9 Discussion

Features and Predictions

The scene Graph we introduce in this Chapter is designed to capture spatial relationships between objects, object categories, and the room. Overall, we have found that each of the relationships we have presented improves the SceneGen algorithm's ability to augment virtual objects in realistic placements in a scene. These relationships are important to understanding the functional purposes of

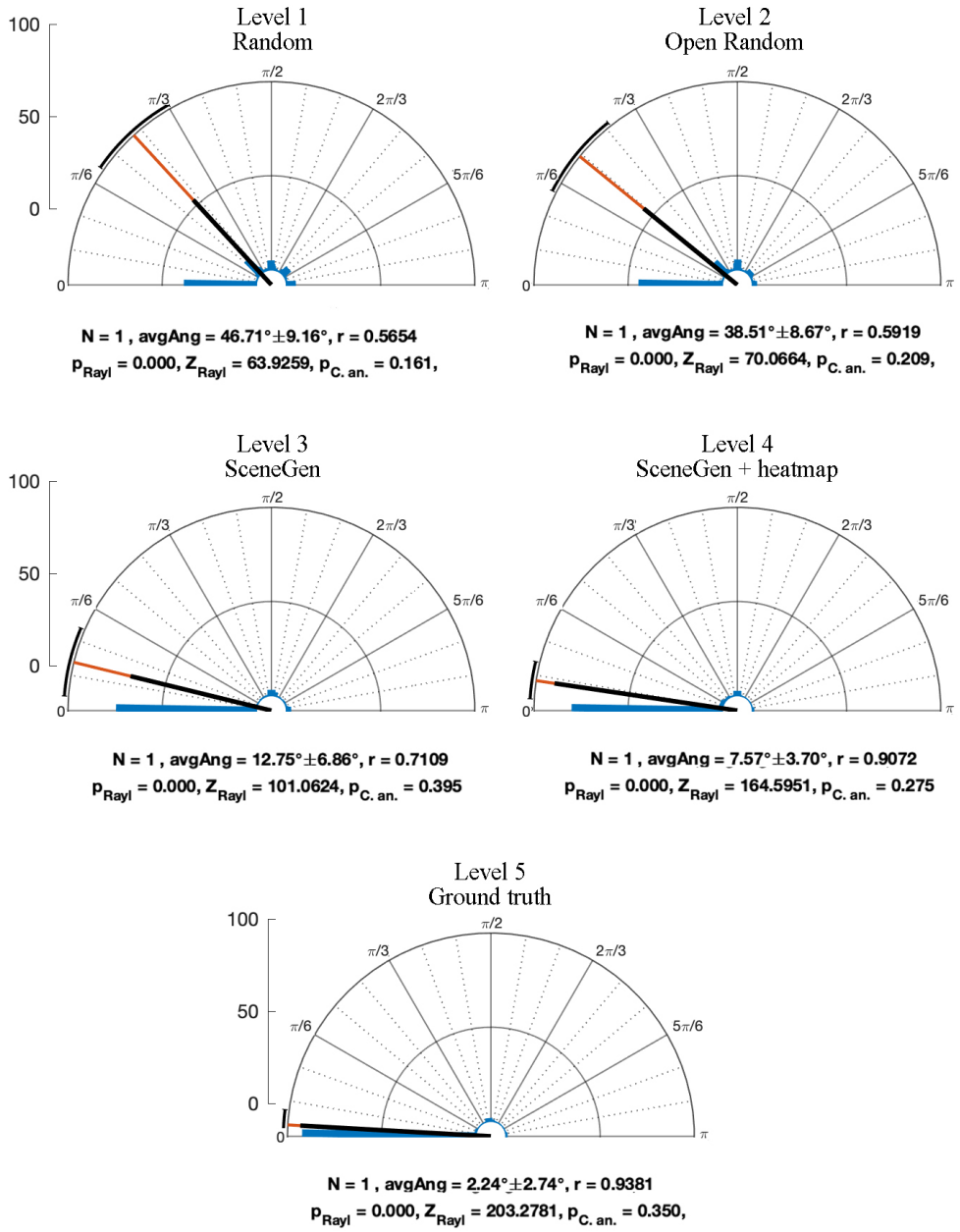


Figure 4.15: Radial histograms display distribution of how much a user rotates an object from its orientation in each level of the user study.

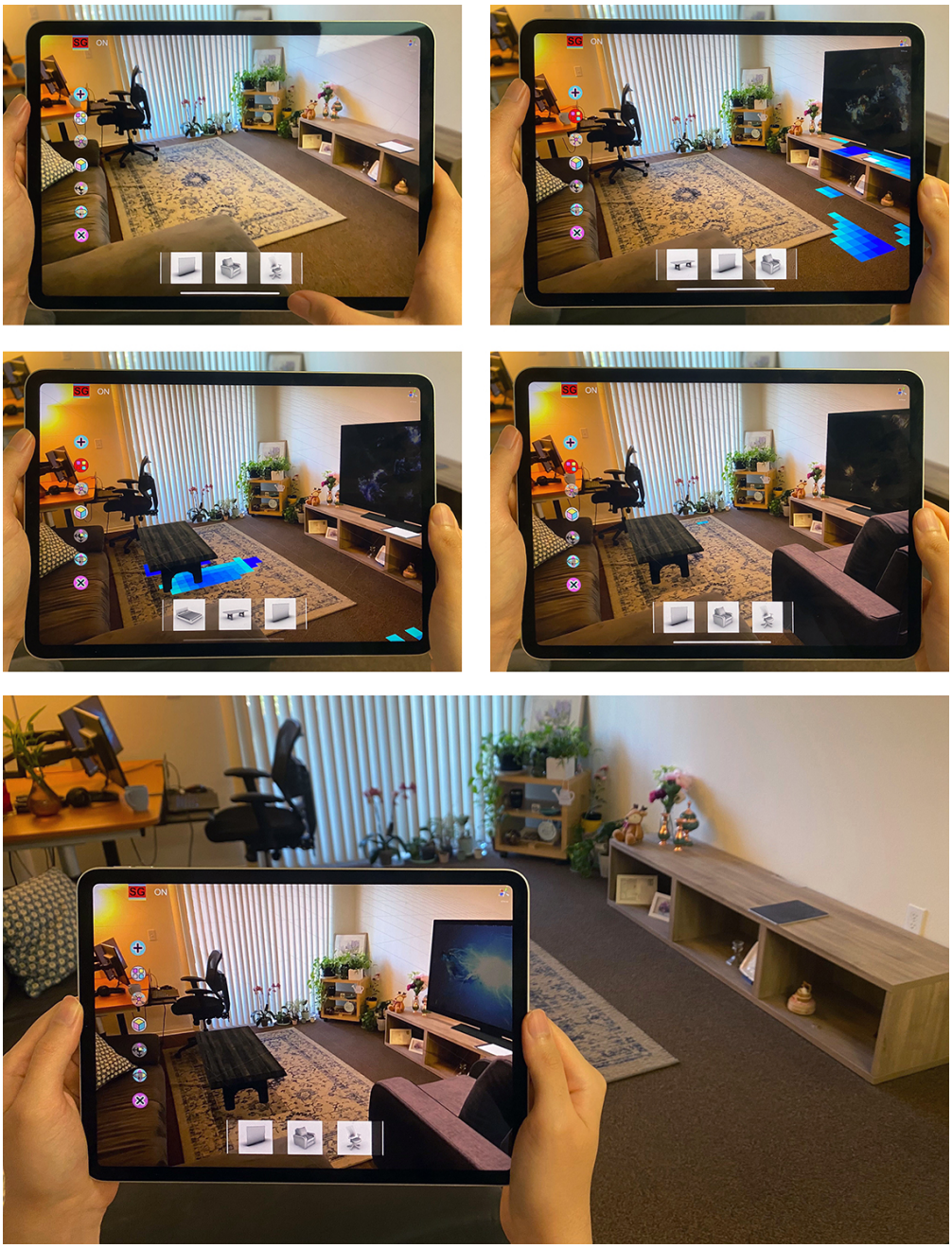


Figure 4.16: Augmented Reality application demonstrates how SceneGen can be used to add virtual objects to a scene. Top Left: the target scene, Top Right: adding a TV, Middle Left: adding a table, Middle Right: adding a sofa. A probability map displays how likely each position is. Bottom: the AR application with virtual objects is compared to the original scene.

the space in addition to the individual objects. Our approach can also be extended to predicting the best category of a new object type to augment by running an exhaustive search on all the categories for a given input room coordinate.

In SceneGen, RoomPosition is used as a feature in predicting both orientation and position of an object. While this feature is based solely on the position of the object, it also has a strong impact on how it should be oriented. For example, a chair in the corner of the room is very likely to face toward the center of the room, while a chair in the middle of the room is more likely to face toward a table or a sofa. When analyzing our placement predictions probability maps and our user study results, we have observed that the best orientation is not only affected by the nearby objects but also by the sampled position within the room.

Explicit Knowledge Model

In our evaluation of SceneGen, we have found a number of benefits in using an explicit model to predict object placements. One benefit is that if we want to define a non-standard object to be placed in relation with standard objects by specifying your own relationship distributions, it is feasible with our system but would not be possible for implicit models. For example, in a collaborative virtual environment, where special markers are desired to be placed near each user, one could directly specify distributions for relationships such as NextTo chair and Facing table without retraining an implicit model such as neural networks.

Another benefit is that explicit models can be easily examined directly to understand why objects are being placed where they are. For example, the Bed orientation feature distribution, based on the Matterport3D priors in Figure 4.5, marginalized with respect to all other variables except TowardsCenter, shows that beds are nearly 5 times as likely to face the center of the room while marginalizing features except the position of the Storage show that storage is found in the corner of a room 63% of the time, along an edge 33% of the time, and only in the middle of the room in 4% of occurrences.

Subjectivity of Placements

Where and how an object is placed in a scene is often very subjective, and preferences can differ between users. This is demonstrated by the Likert scale plausibility ratings in Level V reference scenes in the user studies. Figures 4.13 and 4.18 show that some users would only give scores of *somewhat plausible* to scenes that are modelled from real-world ground truth Matterport3D rooms. This supports providing a heat map of probabilities for each sampled placement, as alternate high probability positions may be preferable to different users. Our results also indicate that most users prefer level IV scenes, with the heat map, compared to level III scenes, even though the placements use the same SceneGen models. This suggests that the inclusion of the heat map guides the users towards the system's placement and may help in convincing them of the viability and reasoning for such a choice.

We also see that some users move objects to other high probability alternatives, as seen in Figure 4.17. This is a similar result to the position prediction experiment, which compares the ground truth

position to the closest of SceneGen's top 5 predictions and shows that while the reference position may not always be the top prediction, it was often one of the top predictions. Moreover, results in Figure 4.18 show the subjectivity of an object placement is highly dependent on the size and type of the object itself. In any room, there are very few natural places to put a bed. Hence the results for placing beds cluster in one or two high probability locations. Other objects such as decor are more likely to be subject to user preferences.

4.10 Conclusion

In this Chapter, we introduce a framework to augment scenes with one or more virtual objects using an explicit generative model trained on spatial relationship priors. Scene Graphs from a dataset of scenes are aggregated into a Knowledge Model and used to train a probabilistic model. This explicit model allows for direct analysis of the learned priors and allows for users to input custom relationships to place non-standard objects alongside traditional objects. SceneGen places the object in the highest probability pose and also offers alternate highly likely placements.

We implement SceneGen using the Matterport3D, a dataset composed of 3D scans of lived-in rooms, in order to understand object relationships in a real-world setting. The features that SceneGen extracts to build our Scene Graph are assessed through an ablation study, identifying how each feature contributes to our model's ability to predict realistic object placements. User Studies also demonstrate that SceneGen is able to augment scenes in a much more plausible way than a system that places objects randomly or in open spaces. We also found that different users have their own preferences for where an object should be placed. Suggesting multiple high probability possibilities through a heat map allows users an intuitive visualization of the augmentation process.

There are, of course, limitations to our work. While SceneGen is able to iteratively add objects to a scene, the resulting layout is dependent on the order in which objects are placed. Such an approach does not consider all possible permutations of the possible arrangements. In addition, it can narrow down the possible open spaces for later objects, forcing placements that are far from optimal. Moreover, in scenarios where a large number of objects are to be augmented, the current approach may not have the ability to *fit* all the objects within the usable space, as initial placements are not aware of upcoming objects. Future work can comprise incorporating floorplanning methodologies with the current sampling mechanism allowing a robust search in the solution space while addressing combinatorial arrangement.

Moreover, SceneGen is a framework that naturally fits into SC applications. We demonstrate this in a AR application that augments a scene with a virtual object using SceneGen. Contextual scene augmentation can be useful in augmenting collaborative mixed reality environments or in other design applications, and using this framework allows for fast and realistic scene and content generation. We plan on improving our framework by providing the option to contextually augment non-standard objects by parameterizing topological relationships, a feature that would facilitate content creation for future SC workflows.

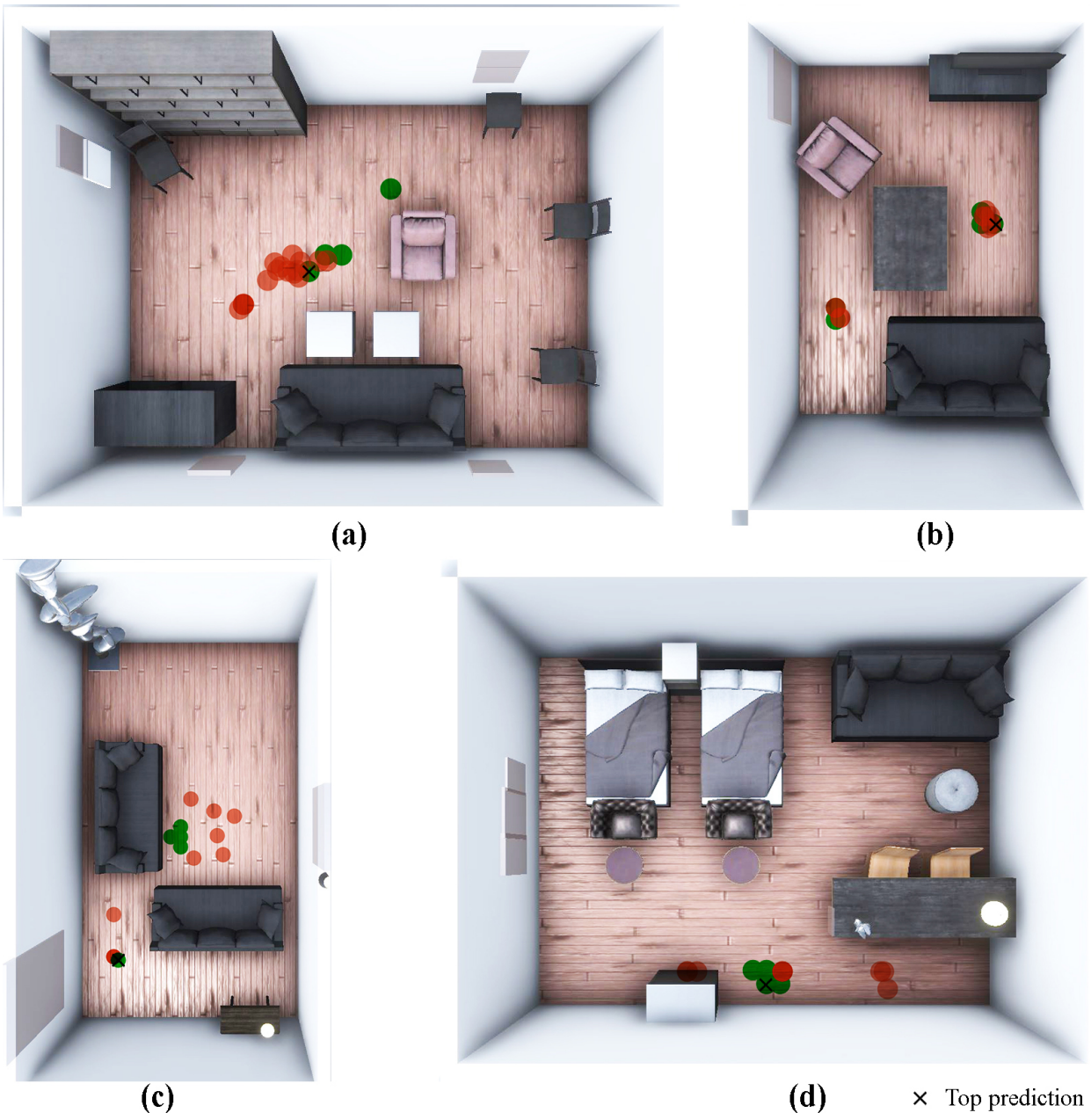


Figure 4.17: Top 5 highest probability positions for placing sofa (a,b), table (c) and TV (d) predicted by SceneGen (green) are compared to the user placements (red) showing that different users's preferences do vary and SceneGen find the clusters as the users' best consensus.

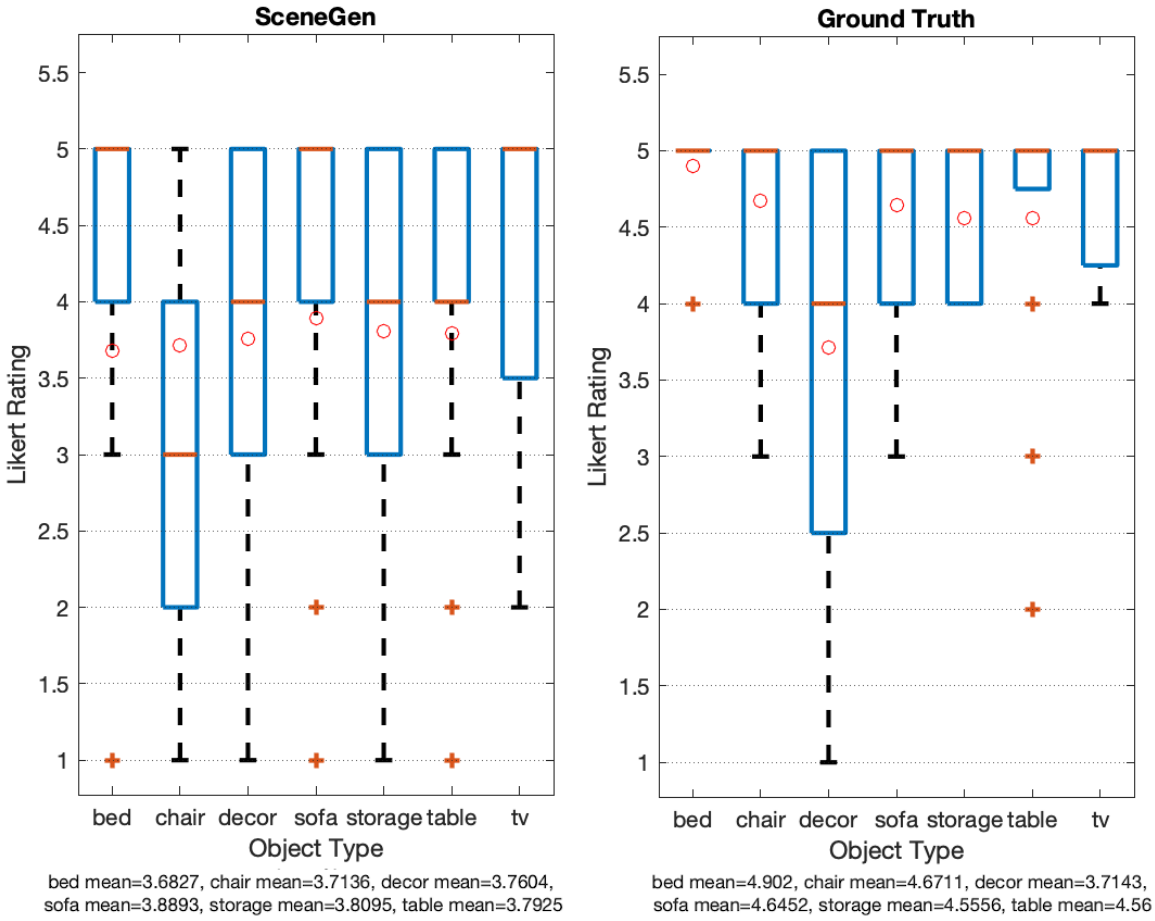


Figure 4.18: The plausibility score for each object category on the Likert Scale given by users is compared between the average scores from SceneGen Levels III and IV (left) and the ground truth Level V (right).

Chapter 5

Generation and Manipulation of Spaces

5.1 Introduction

In the previous chapter, we discussed how example-based scene augmentation could be utilized for allowing large-scale deployment of curated SC experiences. In this chapter, we aim to explore how scene manipulation can be utilized to augment datasets used as scene priors for scene synthesis and augmentation. In addition to synthesizing new environments, the utilization of deep learning techniques via training on large datasets has been widely explored in cross-disciplinary fields of architecture, computer graphics, and computer vision. For tasks such as semantic segmentation [3, 141], object recognition [166], and 3D reconstruction [69, 204], integrating 3D deep learning methodologies have brought a promising direction in the state-of-the-art research. However, the success of many learning-based models is highly dependent on the availability of the appropriate datasets. In contrast to 2D image recognition tasks, where training labeled datasets are available in large quantities, 3D indoor datasets are limited to only a small number of open-source datasets. Capturing 3D geometry is seen to be much more expensive than capturing 2D data in terms of both hardware and human resources.

3D data for training resources for computer vision tasks can be found in two general categories (a) real-world captured data and (b) synthetic data. The first approach involves scanning RGB-D data using high-end capturing systems or commodity-based sensors. To this extent, a number of open-source datasets are available with various scales and capture qualities. The ETH3D dataset contains a limited number of indoor scans [187], and its purpose is for multi-view stereo rather than 3D point-cloud processing. The ScanNet dataset [48] and the SUN RGB-D [203] dataset capture a variety of indoor scenes with added semantic layers. However, most of their scans contain only one or two rooms, which is not suitable for larger-scale layout reconstruction problem. Matterport3D [30] provides high-quality panorama RGB-D image sets for 90 luxurious houses captured by the Matterport camera. The 2D-3D-S dataset [4] provides large-scale indoor scans of office spaces by using the same Matterport camera.

The second approach is to utilize synthetic 3D data of building layouts and indoor scenes, which has also been recently produced in mass numbers to fill the void of rich semantic 3D data. SUNCG

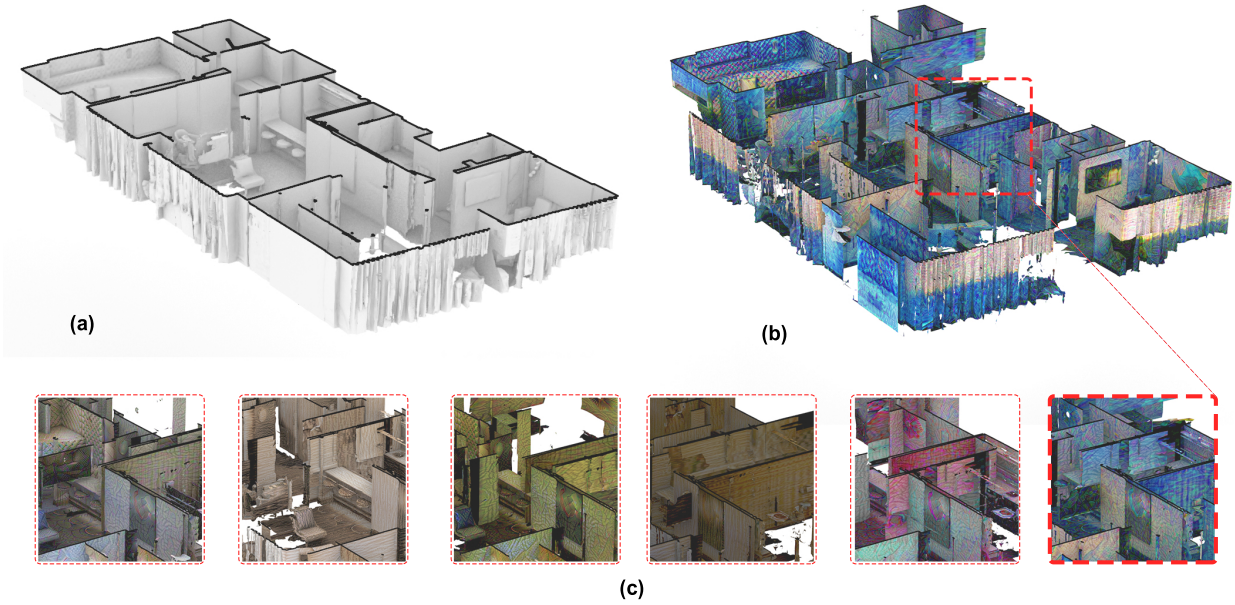


Figure 5.1: GenScan takes an existing captured 3D scan (a) as an input and outputs alternative parametric variations of the building layout (b) including walls, doors, and furniture with (c) new generated textures.

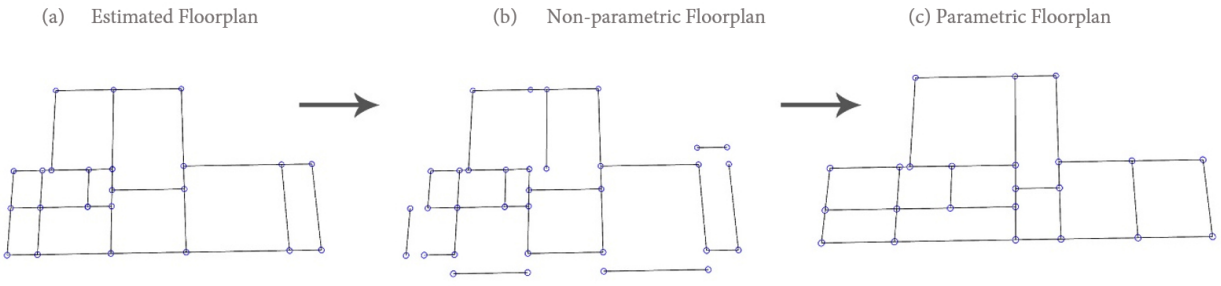


Figure 5.2: Applying individual transformations to wall segments results in the inconsistency of the output layout (b). Using the Parametrizer module we avoid unwanted voids and opening in the building's walls

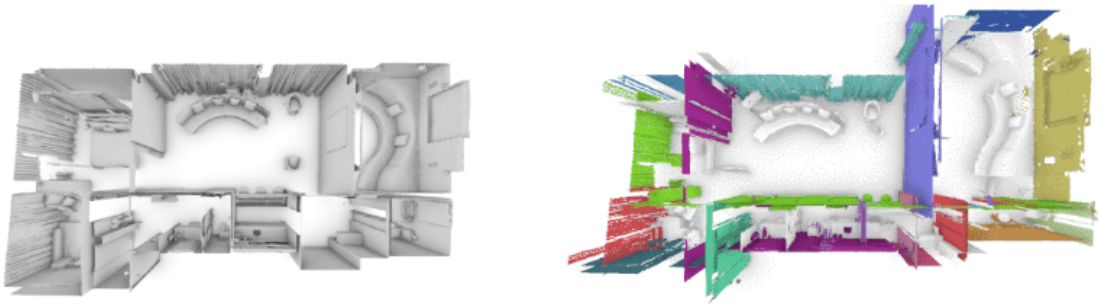


Figure 5.3: Results of the parametric modification (right) of an input scan (left)

[204] offers a variety of indoor scenes with CAD-quality geometry and annotations. However, the level of detail and complexity of the different building elements in such crowd-sourced synthetic approaches is extremely limited when compared to 3D scanned alternatives. Synthetic datasets lack the natural transformation and topological properties of objects in real-world settings.

Furthermore, there is a broad body of literature focused on synthesizing indoor scenes by learning from prior data [251]. While such approaches are mainly focused on predicting furniture placements and arrangement in an empty [111, 59] or partially populated scene [90], they are also dependent on the quality and diversity of the input data in their training stage. Procedural models have also been widely used in generating full buildings [148, 181], furniture layout [143, 63] and manipulating indoor scenes [245, 94]. Yet again, the outputs of such methods lack the complexity of real-world captured data, falling short of being effectively utilized in common computer vision tasks.

Therefore, augmenting large-scale datasets of 3D geometry which correspond to the complexity of the built environments is still an open challenge. Motivated by this challenge, we introduce GenScan, a generative system that populates synthetic 3D scan datasets. GenScan generates new semantic scanning datasets by transforming and re-texturing the existing 3D scanning data in a parametric fashion. The system takes an existing captured 3D scan as an input and outputs alternative variations of the building and furniture layout with manipulated texture maps. The process is fully automated and can also be manually controlled with a user in the loop. Such an approach results in the production of multiple data points from a single scan for 3D deep learning applications.

5.2 Methodology

The general workflow of the system consists of four main components. First, we predict the floorplan of the input 3D scan using a hybrid deep neural network (DNN). We classify what type of building the input model is and estimate what common finishing wall-to-wall distance the input model holds. Second, to avoid inconsistencies in the manipulated walls, we parameterize all

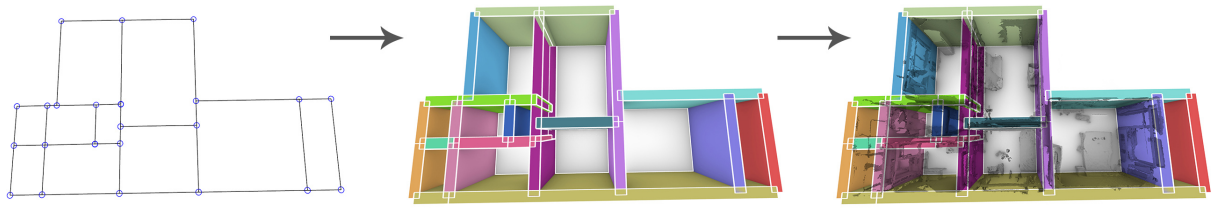


Figure 5.4: Wall extraction module. We use the estimated floorplan layout and door sizes to construct threshold bounding boxes centered on each parametric line. With this method we classify wall elements (colored) and non-wall elements (white) in the scene.

generated vectors to prepare for element transformation. Third, we classify wall elements of the 3D scan using the predicted floorplan and automated thresholds and apply parametric transformations to all wall and non-wall elements separately. Finally, we apply a style transfer algorithm using a combination of a pre-trained VGG network and gradient descent module to current texture maps to generate new textures for the generated scenes.

Parameterization

As shown in Figure 5.2.b, moving an individual wall or a group wall with a certain transformation matrix produces inconsistency in the generated output layout, with unwanted gaps and voids emerging between corner points of the floorplan. We instead assign transformations to the corresponding nodes of the corner coordinates of the target wall elements. We utilize a modified implementation of [95, 14] to parametrize the extracted floorplan. This would manipulate all lines connected to the transformed node. However, to avoid distortion of the orthogonal nature of the building floorplans, we merge co-linear paths that connect to each other with a mutual node and share the same direction vector. Next, we identify the array of nodes that are located on the co-linear lines. After applying transformations to the connected line node array, we construct new polylines from each node array. This would result in a fully automated parametric model that takes transformation vectors and connected line indices as an input and outputs a new floorplan layout without producing undesired gaps and floorplan voids.

Wall Extraction

To classify the walls and movable edges of the input 3D scan, we use the original parametric model to extrude threshold bounding boxes centered on each of the co-linear parametric lines generated in the previous step. We then construct a bounding box for each available mesh in the 3D scan input and test if they inscribe within any of the connected line bounding boxes. With this method, we estimate whether a mesh is part of the building wall system or not, and if so, we can find out which connected wall it is subscribed to. To define the width and threshold of the connected line

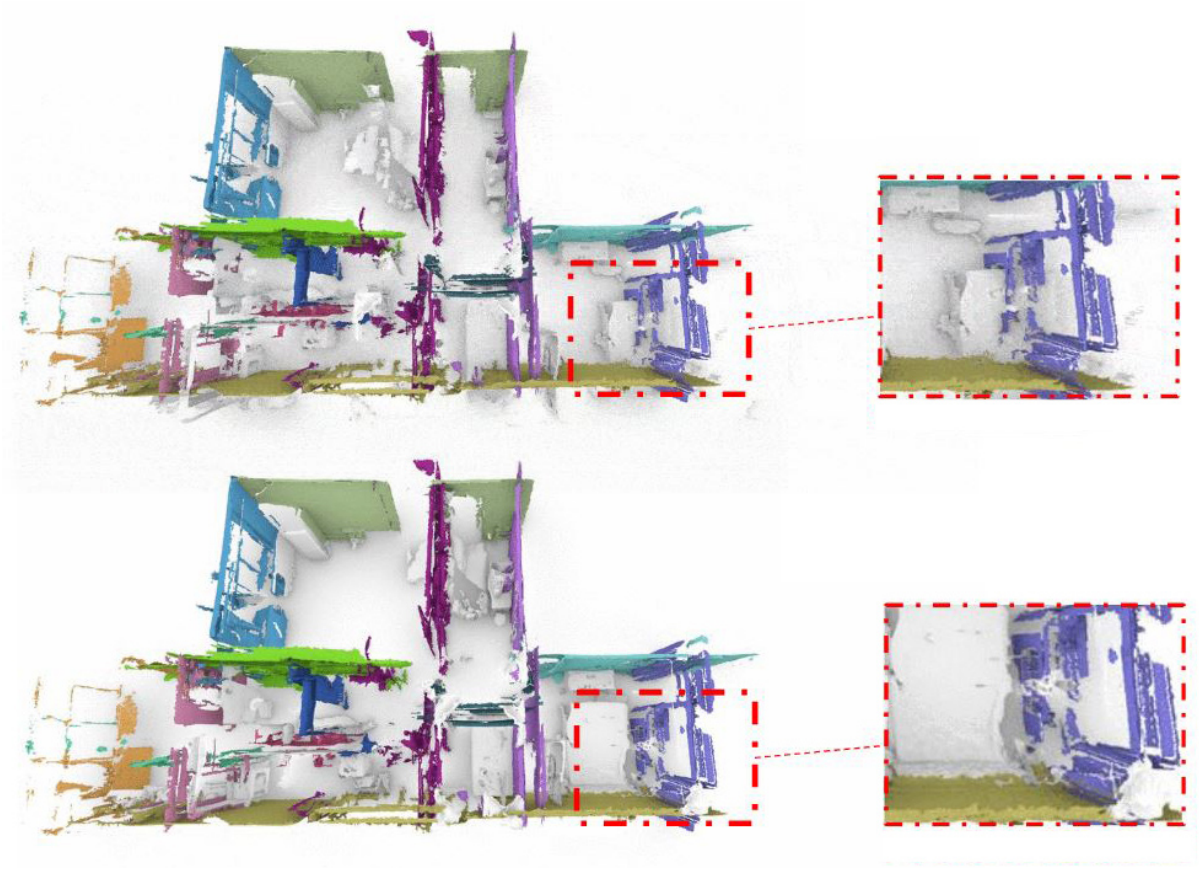


Figure 5.5: Transformation on wall elements only (top). Transformations on wall elements and closest furniture correspondingly

bounding box, we take advantage of the extracted door sizes provided by the hybrid DNN module introduced in [121]. Based on the door sizes, we can heuristically classify the building type of the input model and estimate what common finishing wall-to-wall distance the input model holds. This distance can later be verified by measuring whether a significant peak in the average height takes place within the calculated range. However, both heuristics are not always precise, as elements such as tall bookshelves and cabinets may interfere with the thickness estimation of the walls.

Model Transformation

Given a connected line index and an offset value, all nodes corresponding to the target line would be transformed in the direction perpendicular to the connected line. In many cases, this would not only affect the target line itself but also change the size of neighboring connected lines (Figure 5.2.c). After all the transformations are applied to the nodes of the graph, we calculate the difference between the transformation matrix of the initial geometry and the final geometry.

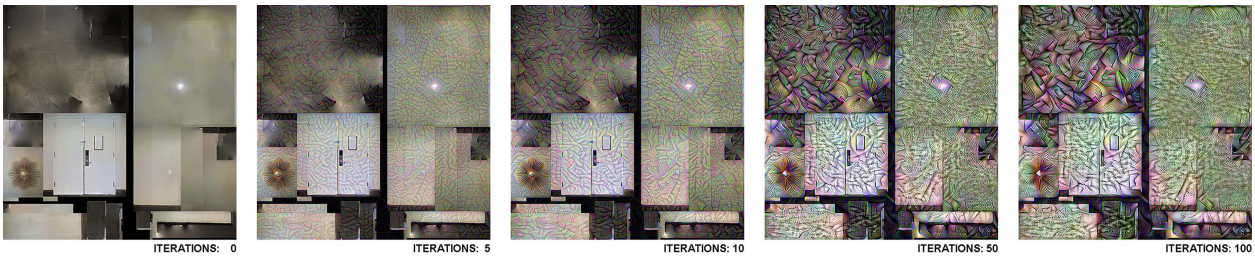


Figure 5.6: Iterations of the style transfer gradient descent algorithm.

This includes a two-dimensional translation vector defining the variance in the position and also a scale factor computed from the center of each line. Next, we apply the transformation vector of each connected line to all input meshes included in the corresponding bounding box. This would result in the parametric movement of the estimated walls while maintaining the overall node graph constructed between all wall elements. By applying the scale transformation specifically to the x and y directions, we stretch and shrink the walls to avoid unwanted architectural inconsistencies and prevent the transformed output from containing irrelevant voids and structural gaps.

However, as shown in Figure 5.4, in many cases, the modification made to the walls can overlap with non-wall elements or the building furniture. This can result in conflicting mesh artifacts in certain clusters. To address this problem, we calculate the center coordinates of each bounding box assigned to non-wall meshes and perform the closest point search with the parametric line system to find the closest wall. We then transform each mesh with the two-dimensional position translation vector of the corresponding closest wall, with a non-linear factor of its distance to the wall. Therefore, a non-wall mesh element closer to the wall would have a much similar transformation function to the wall itself than a non-wall mesh element located in the middle of the room. This would allow furniture to move close and far in relation to each other instead of moving in a similar direction altogether.

Model Generation

The parametric model can be modified to alternate layouts using two main approaches. First, by manually inputting the system a list of parametric line indices and a corresponding offset value, which requires a user in the loop. The second approach is by providing a random range of offset values to be assigned to random parametric lines of the model. Such a method allows mass generations of synthetic 3D scans, which can be later filtered and sorted by implementing evaluation functions. Figure 5.5 illustrates a random floorplan generation of a 3D scan using this method.

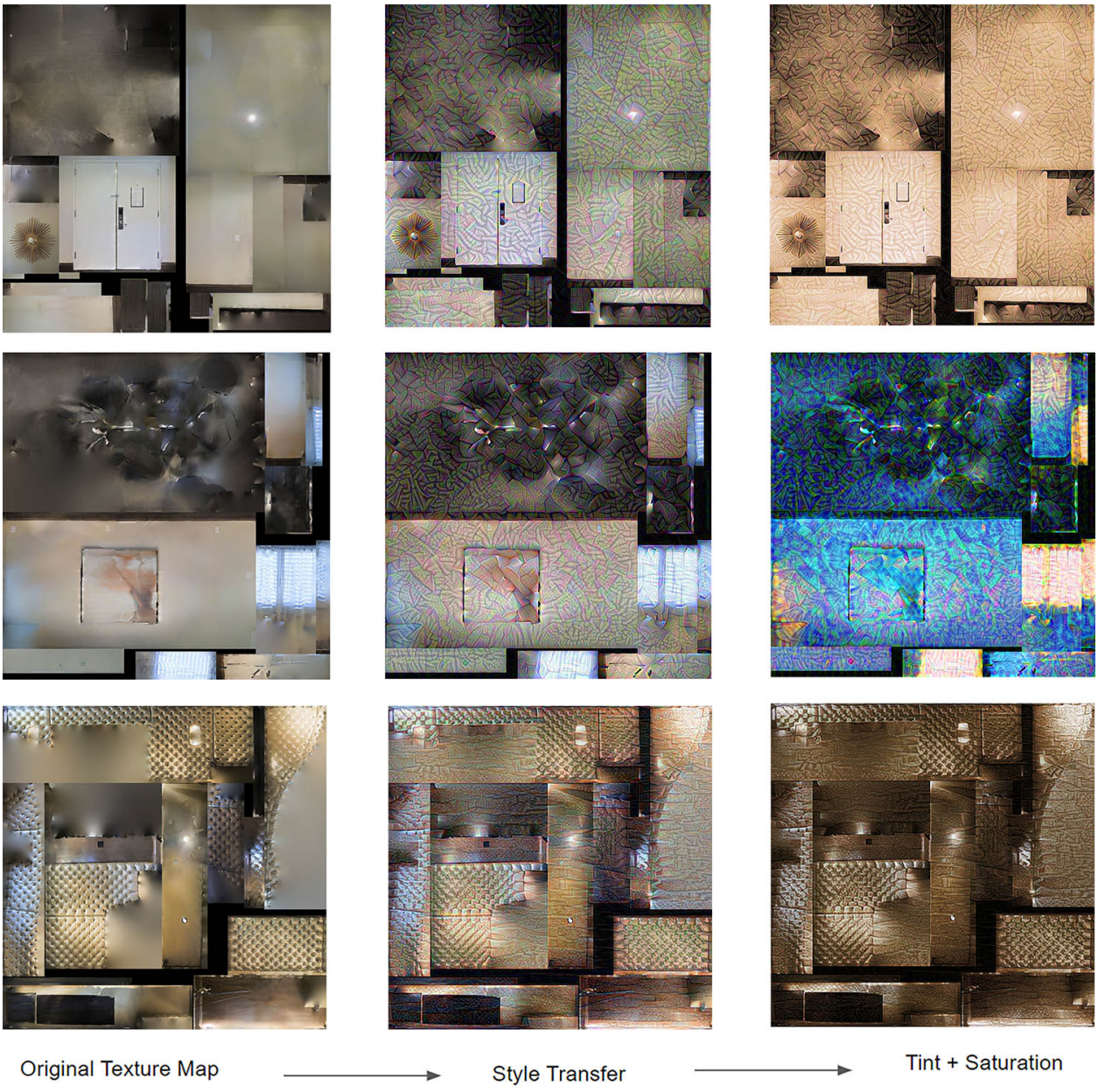


Figure 5.7: Different texture maps modified through style transfer and color modification. Permutations of matching style transfer with modified tints, hues, and saturation can be applied to generate diverse texture maps.

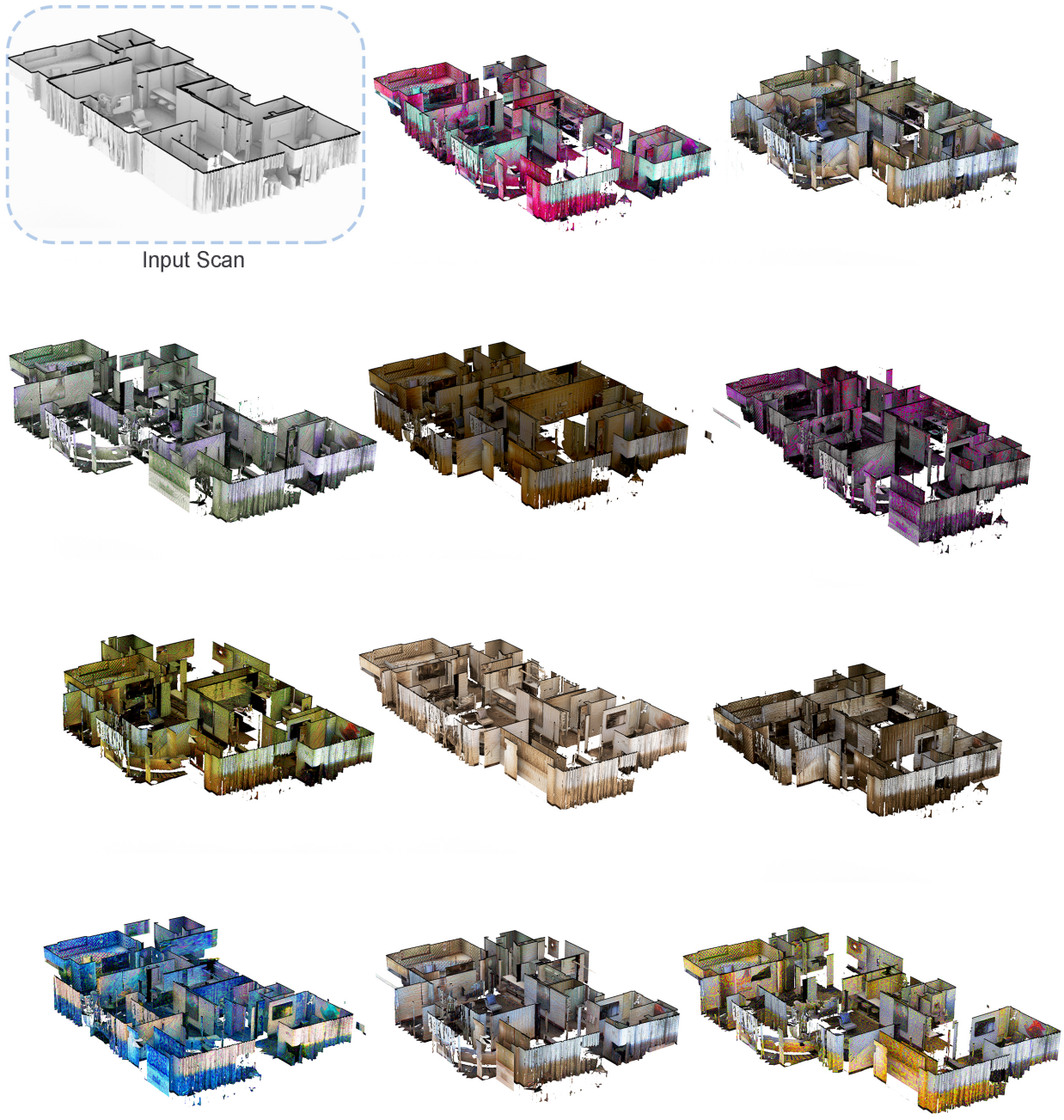


Figure 5.8: Examples of 3D mesh population from an input scan (top left) with modified floor geometries, texture elements, and colors.

5.3 Texture Generation

After applying parameterized geometry transformations to the scanned data, we aim to change the overall visual appearance of the newly generated mesh by editing the associated texture maps. Within our texture modification pipeline, we follow two steps to modify the texture maps of the original mesh provided by the input scan data. First, we take all the texture maps associated with one scan and apply a simple style transfer to each of the textures. Next, we take the generated texture map and apply corrections to its image characteristics such as hue, saturation, tint, etc. Finally, by updating the texture coordinates of the vertices in the newly generated geometry, we are able to match the style-transferred texture maps accordingly.

We implement the style transfer method introduced by [62]. We incorporate a pre-trained VGG network to output a style-transferred texture map. We calculate the content loss and style loss of our generated image at each iteration of the algorithm and run a gradient descent module until we reach an iteration that looks visually convincing. In Figure 5.6, we illustrate how the output image converges to the style image while the content loss and image loss are being minimized. The higher the number of iterations, the more distinct the style is on the texture map; therefore, to result in a more subtle effect, we choose a lower number of iterations for its realism. We apply the transfer technique to modify the texture maps included with the Matterport scans. Style transfer would allow diverse modifications of the input textures, an easy and efficient way to blend a generate variations within a single content texture. The style transfer implemented can be the same for each texture or unique. For example, each room or part of the mesh can have its own different texture modification. Our application of style transfer is to change our existing texture maps to look like new textures using this established technique. By using different style images, we create rooms that look like they are made from brick, wood, or even wallpaper laid on them. This versatility of style transfer allows the subset of data regeneration to be limitless and provides a unique enough new mesh that can be used for our original motivation.

Finally, to allow for more texture variation and realism, we apply a post-processing module of hue, saturation, and tint adjustment to the texture maps. In Figure 5.7, we illustrate a variety of textures we can generate with control over these parameters. At the end of our pipeline, we use the original texture to adjust these parameters of the texture map image. We achieve this by converting the image into an RGBA array that we can shift and scale dictated by the desired effect. Overall, through just the texture modification process, we have control and access to infinite choices in style images and parameterization of image characteristics mentioned above. Figure 5.8 displays just a few of the possible final floor layouts created with GenScan.

5.4 Discussions and Conclusion

GenScan applies automated parameterization, and texture modification of 3D scanned geometrical data to produce bootstrapped samples of 3D scanned data. Given data for just a single scan, GenScan actively produces valid synthetic geometric and textured data of multiple potential layouts resulting in floor plans with modified floor geometries, texture elements, and colors. We believe our

system would allow for mass parametric augmentation to expand the currently limited 3D geometry datasets commonly used in 3D computer vision and deep learning tasks. Such an approach results in the production of multiple data points from a single scan for 3D deep learning methodologies. This methodology can facilitate applications across multiple disciplines, including design optimization, computer vision, virtual and augmented reality, and construction applications.

While the current GenScan system has the ability to parameterize walls and major building elements extracted from the floorplan layout, it does not cover parameterizing smaller room elements such as chairs, beds, tables, desks, etc. Such objects not only need to be identified using semantic segmentation methods, but a parametric relationship would also need to be established to allow relevant layout modifications. Furthermore, generating non-orthogonal layouts and extending parameterization to distorted and curved layouts can also be considered as the next steps in this study. Another limitation of our system lies in the inability to modify the textures of specific walls and non-wall objects of our choosing. Identifying specific areas of the texture maps to regenerate and filling in gaps produced by expanding the layout would result in a cleaner 3D model. Moreover, applying unique changes to specific parts of the texture maps instead of the whole map would allow for even greater customization, variability, and realism of the data. Finally, streamlining our implementation of the texture modification process in our pipeline will achieve higher texture resolution quality in an efficient time period.

Chapter 6

One Shot Learning for Scene Generation

6.1 Introduction

We previously discussed that a major challenge for large-scale deployment of curated SC experiences is that the target scene is not necessarily known to the content developer. Furthermore, In Chapter 4, we explored how scene augmentation can be performed by learning from scene graph priors. However, to formulate a good solution for contextual augmentation, it is tempting to adopt the recent trend of using deep neural networks (DNN). Nevertheless, it is also well known that any modern DNN solution would require large amounts of training data to estimate a set of optimal parameters. This requirement can be met by using either elaborately scanned building datasets [30, 48, 203] or synthetic 3D building datasets [204, 101, 114, 175]. In this work, we further propose a novel method to perform critical training data augmentation step in DNN training via contextual synthesis based on real scanned datasets, a good balance between the above two distinct approaches.

Together, our proposed algorithm is called GSACNet, which is an acronym for *Graph attention Siamese AutoenCoder Network*. Its main contributions are as follows:

1. GSACNet combines parametric data augmentation techniques with a novel network architecture to achieve plausible indoor scene layouts with small training data.
2. By sampling the user's target room space, we generate topological scene graphs to represent the high-level relationship between objects in the room. This serves as an input to the Graph Attention Network, followed by a Siamese Network.
3. Finally, autoencoder networks cast the plausibility prediction as an anomaly detection problem. Using such workflow, we can generate probability maps for an object augmentation in a target scene.

Siamese networks were first introduced in [17] to solve signature verification as an image matching problem. A Siamese neural network consists of twin networks, each accepting distinct inputs but joined by an energy function at the top. This function computes some metric between the highest-level feature representation on each side [100]. Siamese networks have been used in various

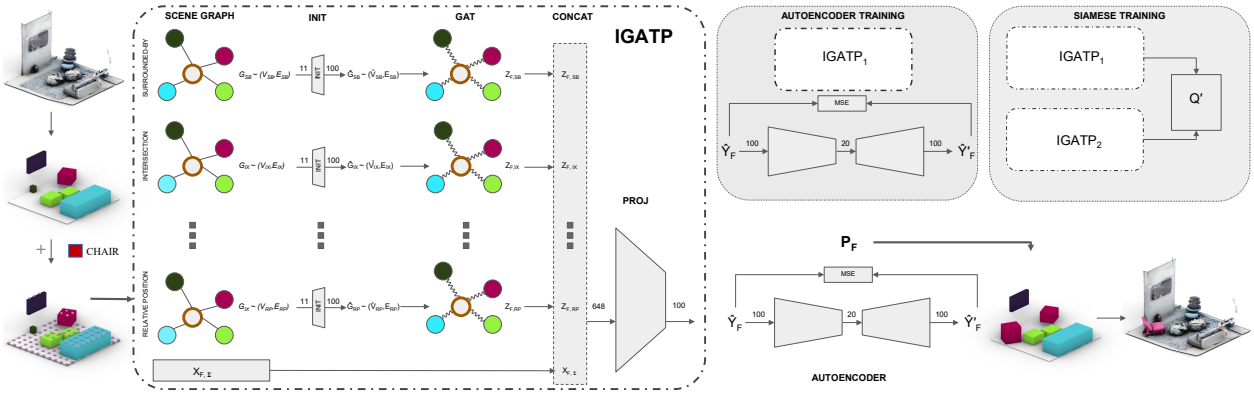


Figure 6.1: To contextually place an object within a scene, GSACNet takes a semantically labeled indoor scene as input and outputs a plausible placement of the object. The system consists of a graph attention, Siamese and auto-encoder network that can be trained with limited scene priors.

applications of indoor design and floorplanning due to their ability to learn from limited data. For example, [67] used Siamese networks for scene change detection.

Furthermore, Autoencoders [16, 74] are a type of neural networks designed to map high-dimensional input data to a low-dimensional latent representation that captures most of the important information needed to reconstruct the data back. This is achieved by a sequence of nonlinear mapping (encoder network) from the input space to a latent space, followed by another sequence of nonlinear mapping (decoder network) from the latent space back to the input space. The parameters in these mapping networks are chosen to minimize the difference between the input data and its reconstructed image. A byproduct of this design is that autoencoders may be used to detect data that strays from the input distribution during training. The idea is that the low-dimensional latent space is forced to capture only information about the subset of the input space that data is drawn from and accurately reproduce data living in this space. All other data points will suffer a significant loss in fidelity when mapped through the autoencoder. Consequently, autoencoders have been used for anomaly detection in a variety of applications [259, 256, 180]. Variations of autoencoders and other similar networks such as Generative Adversarial Networks have been used as tools for generating plausible indoor scenes by sampling from their associated latent spaces [163, 111]. In this work, we make use of autoencoders to discriminate between natural-looking and random scene arrangements and separate a scene proposal and scene generation as two independent tasks.

6.2 Methodology

Figure 6.1 shows the general workflow of our system. Given a semantically segmented target room, our system aims to contextually place objects within the scene while maintaining a plausible relationship with the room and its objects. To do so, the room space will be sampled uniformly

where the sample points are considered the center of possible placement, and the plausibility probability of each sample is then calculated. The GSACNet architecture involves five modules: (1) scene graph extraction; (2) initialization; (3) graph attention; (4) projection into learned space; and (5) plausibility assessment via an autoencoder network. The integral copies of the first four modules together are called IGATP (Initialization Graph ATtention Projection), and the modules are then used for Siamese training; the autoencoder is trained separately. In the following subsections, we first define the topological relationships in which the scene graphs utilize, followed by the formulations of the various components of our network architecture.

Definitions

We consider a room or a scene in 3D space where its floor is on the flat (x,y)-plane and the z-axis is orthogonal to the (x,y)-plane. We denote the room space in a floorplan representation as R, namely, an orthographic projection of its 3D geometry plus a possible adjacency relationship that objects in R may overlap on the (x,y)-plane but on top of one another along the z-axis. This can also be viewed as a 2.5-D representation of the space.

Further denote the k-th object (e.g., a bed or a table) in R as O_k. The collection of all n objects in R is denoted as $\mathscr{O} = \{O_1, O_2, ...O_n\}$. $B(O_k)$ represents the bounding box of the object O_k. $\dot{O}_k$ represents the center of the object O_k. For convenience, we will also define $\dot{O}_{kxy}$ as $\dot{O}_k$ but projected onto either existing furniture below or the floor plane of R. Every object O_k has a furniture group to classify its type. The set of all furniture groups is called $G = \{g_1, ..., g_m\}$, where each group g_i contains all objects of the same furniture group. Furthermore, $\dot{O}_k$ is the centroid of $B(O_k)$.

For each room R, we define $\mathscr{W} = \{W_1, W_2, ..., W_l\}$ where each W_k is a wall of the l-sided room. In the floor plan representation, W_k is represented by a 1D line segment. We also introduce a distance function $\delta(a,b)$ as the shortest distance between a and b objects. For example, $\delta(B(O_k), \dot{R})$ is the shortest distance between the bounding box of O_k and the center of the room R. Intersection of bounding boxes is regarded as $\delta(B(O_k), B(O_j)) = 0$.

Spatial Relationships

Object to Room Relationships

RoomPosition: The room position feature of an object denotes whether an object is at the middle, edge, or corner of a room. This is based on how many walls are less than a distance ρ away from an object. For convenience, we define ϕ as follows:

$$\phi\left(O_k, W_i\right) = \mathbb{1}(\delta(B(O_k), W_i) < \rho). \tag{6.1}$$

Using ϕ, we define RoomPos as follows:

$$\text{RoomPos}\left(O_k, R\right) = \sum_{W_i \in (W)} \phi(O_k, W_i). \tag{6.2}$$

In other words, if $\text{RoomPos}(O_k, R) \geq 2$, the object is near at least two walls of a room and hence is near a *corner* of the room. If $\text{RoomPos}(O_k, R) = 1$, the object is near only one wall of the room and

is at the *edge* of the room. Otherwise, the object is not near any wall and is in the *middle* of the room.

Object to Object Group Relationships

AverageDistance: For each object and each group of objects, we calculate the average distance between that object and all objects within that group.

$$\text{AvgDist}(O_k, g_i) = \frac{\sum_{O_j \in g_i} \delta(B(O_k), B(O_j))}{|\{g_i\}|}. \tag{6.3}$$

SurroundedBy: For each object and each group of objects, we compute how many objects in the group are within a close proximity of an object. Suppose O_k and O_j are within room R. O_j is within the proximity of O_k if $\delta(B(O_k), B(O_j)) < \varepsilon_k = \|[L_k, W_k]\|_2$, where L_k, W_k refer to the length and width of $B(O_k)$, respectively. For convenience, we define a function σ as follows:

$$\sigma\left(O_k, O_j\right) = \mathbb{1}(\delta\left(B(O_k), B(O_j)\right) < \varepsilon_k). \tag{6.4}$$

Using σ, we define the surrounded-by function SurrBy as follows:

$$\text{SurrBy}\left(O_k, g_i\right) = \sum_{O_j \in g_i} \sigma(O_k, O_j). \tag{6.5}$$

IntersectionXY: For each object and each group of objects, we compute how many objects in the group are intersecting an object in the (x, y) plane. Suppose O_k and O_j are within room R. O_j intersects O_k in the (x, y) plane if $\delta(B_{xy}(O_k), B_{xy}(O_j)) = 0$. $B_{xy}(O_k)$ refers to the bounding box of O_k projected onto the ground floor plane of R. For convenience, we define a function ι as follows:

$$\iota\left(O_k, O_j\right) = \mathbb{1}(\delta\left(B_{xy}(O_k), B_{xy}(O_j)\right) = 0). \tag{6.6}$$

Using ι, we define the intersection-XY function InterXY as follows:

$$\text{InterXY}\left(O_k, g_i\right) = \sum_{O_j \in g_i} \iota(O_k, O_j). \tag{6.7}$$

Co-Occurence: Given a room R, an object O_k and another object $O_j, k \neq j$ are said to co-occur if they exist within R.

$$\text{Cooc}\left(R, O_k, O_j\right) = \mathbb{1}(O_k, O_j \in R \wedge k \neq j). \tag{6.8}$$

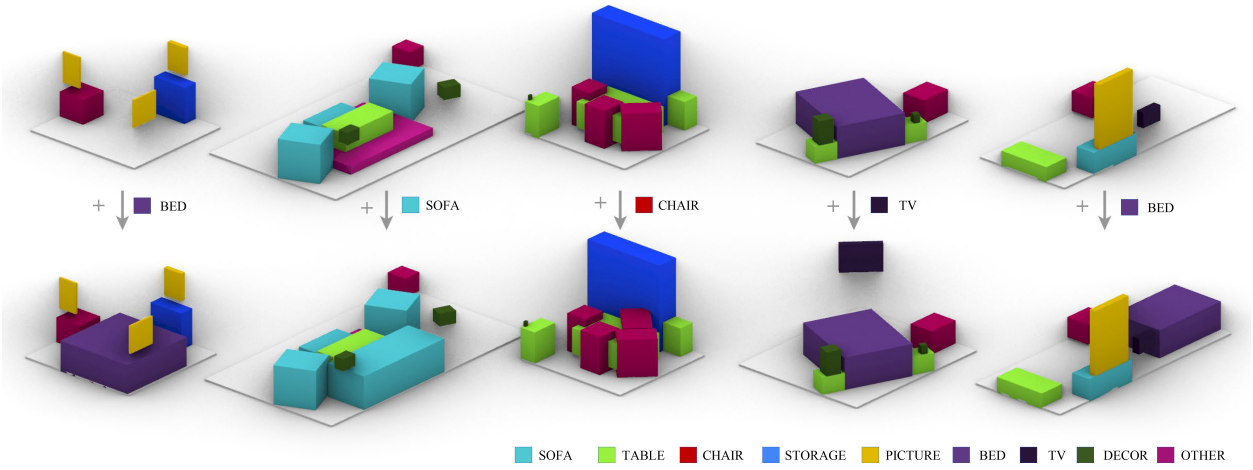


Figure 6.2: Example of contextual scene augmentation results. Top row illustrates the target scene, and bottom row illustrates the augmented scene.

Object Support Relationships

Support: An object is considered to be supported by a group if it is on top of an object from the group or supports a group if it is underneath an object from the group. Due to erroneous bounding box intersections within our dataset, we relax the definition of support by enforcing a threshold τ on the separation distance between the bottom bounding box plane of the top object and the top bounding-box plane of the bottom object. For convenience, we define a function ψ as follows:

$$\psi\left(O_k, O_j\right) = \begin{cases} 1 & 0 < B(O_k)_{bottom} \text{ - } B(O_j)_{top} < \tau; \\ -1 & 0 < B(O_j)_{bottom} \text{ - } B(O_k)_{top} < \tau; \\ 0 & \text{otherwise.} \end{cases} \tag{6.9}$$

Using ψ, we define more specific support relationships. Specifically, the function SuppBy describes the number of objects that support O_k:

$$\text{SuppBy}\left(O_k, g_i\right) = \sum_{O_j \in g_i} \mathbb{1}\left(\psi(O_k, O_j) = 1\right). \tag{6.10}$$

Similarly, the function SuppTo describes the number of objects that O_k is supporting:

$$\text{SuppTo}\left(O_k, g_i\right) = \sum_{O_j \in g_i} \mathbb{1}\left(\psi(O_k, O_j) = -1\right). \tag{6.11}$$

Network Architecture

Scene Graph Extraction

We sample points uniformly in the (x,y)-plane. Regarding each point as the center of possible placement for a target object O_k on the ground floor plane (the sample point will be temporarily considered as $\dot{O}_k(x,y)$), we form a summary vector X_{O_k} and scene graphs $\mathscr{G}_r \sim (V_r, E_r)$ for each relationship $r \in \mathscr{R}$, where $\mathscr{R}$ is the set of all spatial relationships under consideration. As an important aside, in our scene synthesis system, there is a model per furniture group. For the target object O_k, we will use the model associated with its furniture group g_{O_k}. We denote such model usage by sub-scripting the main modules by g_{O_k}.

We use homogeneous scene graphs to represent spatial relationships. Objects are defined as nodes, and relationships are defined as edges. The target object node refers to the node in a scene graph associated with the object we want to place in the scene (the target object O_k). We refer to this node as v_{O_k}. Secondly, given O_a and O_b both in a room R, the edge from node v_{O_a} to node v_{O_b} is referred to as $e_{(a,b)}$. In the following paragraphs, we will describe the connection criteria per scene graph that we use in our system. A scene graph's connection criteria refers to the rules that determine whether or not there exists an edge between two nodes. We utilize homogeneous scene graphs such as in [257], and we utilize spatial relationship criterion from chapter 4 to construct those scene graphs. However, we introduce the intersection scene graph as a new spatial relationship consideration.

- Intersecting Objects Scene Graph.

$$E_{IX} = \{e_{(j,k)} | \forall O_j \in R, j \neq k, \iota(O_k, O_j) = 1\} \tag{6.12}$$

- Surrounded-By Scene Graph.

$$E_{SB} = \{e_{(j,k)} | \forall O_j \in R, j \neq k, \sigma(O_k, O_j) = 1\} \tag{6.13}$$

- Support-By Objects Scene Graph.

$$E_{SBY} = \{e_{(j,k)} | \forall O_j \in R, j \neq k, \psi(O_k, O_j) = 1\} \tag{6.14}$$

- Support-To Objects Scene Graph.

$$E_{STO} = \{e_{(j,k)} | \forall O_j \in R, j \neq k, \psi(O_k, O_j) = -1\} \tag{6.15}$$

- Relative Position Scene Graph. Suppose W_R refers to the set of walls of room R and F_R refers to the floor of R. Furthermore, $W_i \in W_R$ is defined similarly as $O_j \in R$, but W_i is signified to represent a wall object. The same is true of F_R. Lastly, $e_{(w_j,k)}$ refers to the edge from v_{W_j} and v_{O_k}, and $e_{(f,k)}$ refers to the edge from v_{F_R} to v_{O_k}.

$$E_{RP,W} = \{e_{(w_j,k)} | \forall W_j \in W_R, \phi(O_k, W_j) = 1\} \tag{6.16}$$

$$E_{RP,F} = \{e_{(f,k)} | E_{RP,W} = \emptyset\} \tag{6.17}$$

$$E_{RP} = E_{RP,W} \cup E_{RP,F} \tag{6.18}$$

Another way to describe E_{RP} is if a wall is within the proximity of an object, then an edge is drawn from the node associated with the wall to the target object. If no walls meet this criteria, an edge is drawn from the floor node to the target object node.

- Co-occurring Scene Graph.

$$E_{CO} = \{e_{(j,k)} | \forall O_j \in R, \text{Cooc}(R, O_k, O_j) = 1\} \tag{6.19}$$

- Graph Feature Vectors and Default Nodes. Nodes have a feature vector associated with them. In particular, the node feature vector is in 11-D space, where the first 10-D represents the one-hot encoding of the object furniture group (first 8-D for the furniture groups and the last 2-D are for the walls and floors, respectively), and the last dimension represents the distance ordering from the target node O_k. Distance ordering refers to an object's rank of how close they are to the target object. For instance, suppose there is a table, a chair, and a bed in the room. The table is considered to be the target object, and the chair is closer to the table than the bed. Then, the table receives a distance order of 0, the chair receives a distance order of 1, and the bed receives a distance order of 2.

  In each scene graph, there also exists a default node such that its feature vector is a zero vector, except for the component associated with relative ordering. For default nodes, the relative ordering is set to -1. The edge exists from the default node to the target object, and if only the default node exists within a scene graph, then no objects meet the connection criteria for the specific scene graph.

Summary Feature Vector

For a proposed floor plane centering $\dot{O}_{k_{xy}}$ of object O_k in room R, the summary vector X_{O_k} can be described as follows:

$$X_{O_k} = \begin{bmatrix} 3C, & EB, & CB, & AD, & SB, \\ IX, & SBY, & STO & & \end{bmatrix} \in R^{48}$$

3_{closest} ($3C$): [257] utilizes an ordered aggregation scheme for message passing. Specifically, messages are passed through a GRU in the order of farthest object-node to closest object-node. Inspired by this idea, our summary vector takes into account the three closest furniture groups in the $3_{\text{closest}} \in R^3$ vector. Closeness is measured by the δ function, and furniture groups are stored such that the closest group is the first component of 3_{closest} and the farthest group is the third component.

$$[EB, CB] = \begin{cases} [1,0] & \text{RoomPos}(O_k, R) = 1; \\ [0,1] & \text{RoomPos}(O_k, R) \geq 2; \\ [0,0] & \text{otherwise.} \end{cases} \tag{6.20}$$

$$AD = [\text{AvgDist}(O_k, g_1), ..., \text{AvgDist}(O_k, g_m)] \tag{6.21}$$

$$SB = [\text{SurrBy}(O_k, g_1), ..., \text{SurrBy}(O_k, g_m)] \tag{6.22}$$

$$IX = [\text{InterXY}(O_k, g_1), ..., \text{InterXY}(O_k, W_R)] \tag{6.23}$$

$$SBY = [\text{SuppBy}(O_k, g_1), ..., \text{SuppBy}(O_k, W_R)] \tag{6.24}$$

$$STO = [\text{SuppTo}(O_k, g_1), ..., \text{SuppTo}(O_k, W_R)] \tag{6.25}$$

Initialization

Features vectors associated with nodes in the scene graphs are passed through a 4-layer initialization neural network $\text{INIT}_{g_{O_k}}$, which transforms the dimensionality of the feature vector from 48 dimensions to 100. The resulting node set then becomes $\hat{V}_r$ to represent nodes associated with the transformed feature vectors $\hat{V}_{r,\text{feats}}$, and the resulting graph becomes $\hat{\mathscr{G}}_r \sim (\hat{V}_r, E_r)$.

$$\forall r \in \mathscr{R}, \hat{V}_{r,\text{feats}} = \text{INIT}_{g_{O_k}}(V_{r,\text{feats}}) \tag{6.26}$$

Graph Attention

Each scene graph $\mathscr{G}_r$ for a spatial relationship r is fed into its respective attention graph layer $\text{GAT}_{g_{O_k},r}$. Multi-head attention is suggested to stabilize the learning process, and applying dropout to the attentional coefficients is found to be a highly beneficial regularizer [223]. Therefore, for each $\text{GAT}_{g_{O_k},r}$, we use 10 heads, each with output dimension of 10, and a dropout of 0.8 for each $\text{GAT}_{g_{O_k},r}$. Concatenating the outputs of each head results in the final output vector of dimension 100, and there is a 100-dimensional output vector given to each node in a scene graph. In the message-passing context, we consider this vector as the finalized message passed to a node.

$$\forall r \in R, Z_r = \text{GAT}_{g_{O_k},r}(\hat{V}_r, E_r) \tag{6.27}$$

Projection

After each scene graph is passed through the scene graph attention module, we extract messages passed to the node associated with the furniture O_k. We concatenate messages Z_r per scene graphs (n total) with the summary vector X_{O_k}. We pass the concatenated vector into a 4-layer network $\text{PROJ}_{g_{O_k}}$, which acts as a method to project the concatenated vector into a space such that data points representing plausible placements are clustered together while data points representing unplausible placements are separated from the cluster. Our resulting projected matrix is labelled as $\hat{Y}$.

$$\hat{Y} = \text{PROJ}_{g_{O_k}}([Z_{r_1}, ..., Z_{r_n}, X_{O_k}]) \tag{6.28}$$

Table 6.1: Data augmentation method with the smallest average distance error between ground truth and top-1 (T1) and top-5 (T5) predicted positions for scene augmentation task.

Furniture	T1	T5	*Furniture*	T1	T5
Bed	M3DPIA	M3DPIA	*Chair*	M3DP	M3DP
Decor	M3DPIA	M3DPIA	*Picture*	M3D	M3DR4PIA
Sofa	M3DPIA	M3DPIA	*Storage*	M3DR4PIA	M3DR4PIA
Table	M3D	M3DPIA	*TV*	M3DP	M3DR4PIA

Plausibility Assessment

Finally, we output a probability of plausible placement P using the reconstruction error produced by an autoencoder $\text{AE}_{g_{O_k}}$. Specifically, the 4-layer encoder of $\text{AE}_{g_{O_k}}$ is given $\hat{Y}$, which converts the input to a coded vector. Then, with the decoder, $\text{AE}_{g_{O_k}}$ will attempt to reconstruct $\hat{Y}$ based off the coded vector. Autoencoders are shown to carry a built-in anomaly detector because decoders will be able to better reconstruct an input to the encoder if the input has been seen before [259]. By training on $\hat{Y}$'s corresponding to real placements of furniture group l_F, we allow $\text{AE}_{g_{O_k}}$ to learn real placements as non-anomalies. With this anomaly detection ability of $\text{AE}_{g_{O_k}}$ in mind, suppose we call the output of the decoder as $\hat{Y}'$. We measure the reconstruction error via the mean squared error MSE between $\hat{Y}$ and $\hat{Y}'$, and we use the reconstruction error as the negative log probability of plausibility. Finally, to convert to a valid probability, we use the mean squared error as the power to an exponential function.

$$P = e^{-MSE(\hat{Y},\hat{Y}')} \tag{6.29}$$

Training

For our system, we have a model M_{g_i} for each furniture group $g_i \in G$, and each model follows the network architecture described in Section 6.2. By using a model per g_i, each model is trained to specialize in the plausible placement of g_i.

We train each model M_{g_i} using two separate training processes. In the first process, we train together IGATP and siamese network projection modules. In the second process, we use the outputs of the first training process as input and train the autoencoder module alone. The following paragraphs detail both training processes.

Siamese Learning

In this training process, we consider the initialization, scene graph extraction, and project modules as one large siamese network IGATP. In our case, labels are binary, where 1 means a plausible placement for g_i and 0 means otherwise.

$\forall i \in \{1,2\}$, unprocessed input D_i contains a room R_i, a furniture to be placed O_i of furniture group g_{O_i}, and placement center $\dot{O}_{ixy}$, and $L_i \in \{0,1\}$ describes whether or not O_i centered at $\dot{O}_{ixy}$ in R_i is plausible. We train this siamese network by giving pairs of unprocessed input. Suppose we

Table 6.2: Average distance error in meters between ground truth and top-1 (T1) and top-5 (T5) predicted positions for scene augmentation task via different models.

System	*Bed*		*Chair*		*Decor*		*Picture*		*Sofa*		*Storage*		*Table*		*TV*		*Overall*	
	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5	T1	T5
Siamese	1.77	1.77	3.03	2.90	2.92	2.86	3.44	3.24	3.49	3.43	2.89	2.86	2.47	2.47	3.23	3.07	2.90	2.82
GAT+ResNet	3.64	3.64	4.89	4.89	3.75	3.75	3.53	3.53	4.43	4.43	2.48	2.48	3.86	3.86	3.53	3.53	3.72	3.72
GAT+Siamese	2.99	2.5	2.87	2.31	**2.85**	2.45	**3.17**	2.08	3.05	2.55	3.12	2.63	2.91	2.35	2.62	2.37	2.98	2.35
Siamese+KDE	2.90	2.90	3.13	3.13	3.65	3.56	3.38	3.23	4.13	4.21	**2.27**	2.52	3.70	3.42	3.28	3.03	3.37	3.03
GAT+Siamese+KDE	1.75	**1.14**	3.25	2.31	3.14	1.75	2.99	2.24	**2.18**	1.07	2.77	1.85	2.95	2.12	2.70	1.99	2.80	1.88
GSACNet (Ours)	**1.47**	1.29	**2.45**	**1.72**	3.10	**2.06**	3.34	**1.83**	2.25	**1.03**	2.48	**1.50**	**2.35**	**1.51**	**2.56**	**1.06**	**2.66**	**1.63**
SceneGraphNet [257]	2.77	2.42	3.56	3.11	3.51	3.02	3.64	3.21	4.59	3.83	2.88	2.51	4.04	3.58	4.46	3.98	3.61	3.15

consider the first unprocessed data point as D_1 and the second as D_2, along with their labels L_1 and L_2. Scene graphs and summary vectors are extracted from R_i from the perspective of O_i, and as described in Section 6.2, IGATP takes these scene representations as input and outputs a vector $\hat{Y}_i$. Therefore, the output associated with D_1 and D_2 would be $\hat{Y}_1$ and $\hat{Y}_2$, respectively.

Now that we have $\hat{Y}_1$ and $\hat{Y}_2$, we calculate the max margin contrastive loss $\mathscr{L}$ between these two outputs.

$$\mathscr{L}(\hat{Y}_1, \hat{Y}_2) = \begin{cases} \|\hat{Y}_1 - \hat{Y}_2\|_2^2 & L_1 = L_2 \\ \max(0, m - \|\hat{Y}_1 - \hat{Y}_{F_2}\|_2^2) & L_1 \neq L_2 \end{cases}$$

$m > 0$ is the margin parameter for the contrastive loss function, and it acts as a lower bound on the distance between a pair of data points with different labels (i.e. $L_1 \neq L_2$).After calculating the contrastive loss, we backpropagate the loss to update weights across the siamese network.

Autoencoder Training

In this training pipeline, we use $\hat{Y}_i$ from the trained siamese network as input to train the autoencoder $AE_{g_{O_i}}$. All $\hat{Y}_i$ correspond to $L_i = 1$ because we want the autoencoder to familiarize itself with plausible placements of furniture group g_{O_i}. As a result, the reconstruction error of plausible ($L_i = 1$) $\hat{Y}_i$ will be low, while the reconstruction error of implausible ($L_i = 0$) $\hat{Y}_i$ will be high. From the perspective of anomaly detection, input vectors associated with implausible layouts will be regarded as anomalies.

As described in Section 6.2, the encoder of $AE_{g_{O_i}}$ takes in $\hat{Y}_i$ as input and transforms the input vector into another vector of smaller dimensionality to create a bottleneck effect. On the other end, the decoder is forced to use the smaller vector to reconstruct the input. We use the mean squared error MSE between the input and the output as the reconstruction error RE.

$$RE = MSE(\hat{Y}_i, \hat{Y}_i')$$

Finally, we backpropagate RE to all the layers of $AE_{g_{O_i}}$.

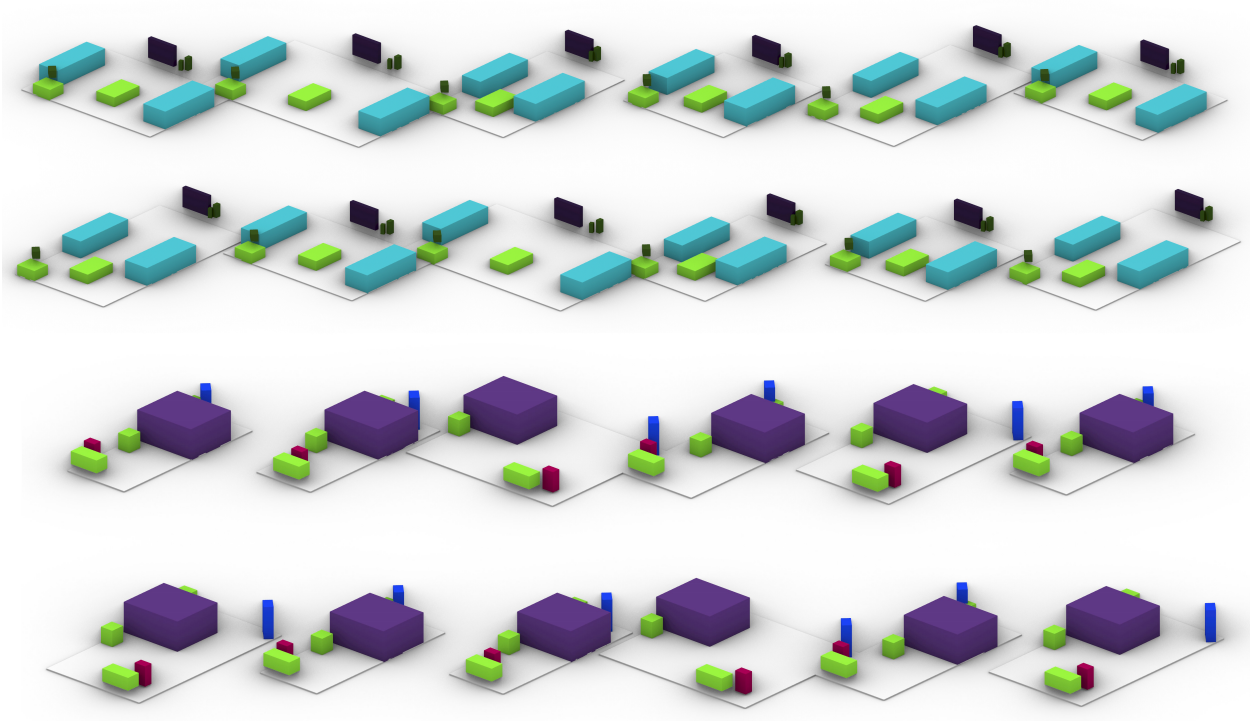


Figure 6.3: Example of parametric data augmentation.

Data Preparation

Dataset

We use Matterport3D (M3D) [30] which consists of various building types with diverse architectural styles, including numerous spatial functionalities and furniture layouts. Annotations of building elements and furniture are provided with surface reconstruction as well as 2D and 3D semantic segmentation. For this study, we reduce the categories of object types considered for building our model and placing new objects. We group the objects into eight coarse categories: $G = \{$Bed, Chair, Decor, Picture, Sofa, Storage, Table, TV$\}$. For room types, we consider the set $\{$Library, Living Room, Meeting Room, TV Room, Bedroom, Recreation Room, Office, Dining Room, Family Room, Kitchen, Lounge$\}$ to avoid overly specialized rooms such as balconies, garages, and stairs. We also filter rooms that hold more than 95% unoccupied areas to avoid unusual empty rooms that come without any spatial arrangements.

Parametric Data Augmentation

We use a modified version of the scene augmentation method introduced in Chapter 5. In this method, after extracting parametric floorplans of the rooms, the model constantly permutates boundary geometry and their adjacent furniture while maintaining a set of functional constraints within the room. This is performed by calculating the center coordinates of each bounding box assigned to objects and performing a closest point search with a parametric wall system to find the closest wall. We then transform each object with the two-dimensional position translation vector of the corresponding closest wall, with a nonlinear factor of its distance to the wall so that an object closer to the wall would have a much similar transformation function to the wall itself than an object located in the middle of the room. This would allow furniture to move close and far in relation to each other instead of moving in a similar direction altogether. During the transformation, the proportional distance of the object with the adjacent walls to the corresponding wall is maintained.

Furthermore, we run two sets of area checks: (a) to check whether the area of the open space of the room is not larger than a certain percentage of the overall area of the room. This would disqualify the augmentations, which result in overly large rooms with extensive open space. Next, (b) we check whether the intersection of two non-colliding objects is not larger than a percentage of the smaller object. Finally, we run another round of data augmentation by removing n smallest objects for the generated scenes. Figure 6.3 illustrates an example of the parametric data augmentation for two example furniture arrangements.

6.3 Experiments

To evaluate our prediction system, we run ablation studies, examining how the presence or absence of particular features affects our prediction results. We use a subset of our dataset, which include 200 room with a 4-fold cross-validation method and an 80/ 20 split between the training and validation set. In these studies, we remove each object in the validation set, one at a time, and use our model to predict where the removed object should be positioned. We compute the distance between the original object location and our system's top prediction. We also compute the smallest distance to the top 5 predictions to address the multi-model property of objects which can be placed in several valid locations.

Data Augmentation

In the data augmentation experiments, we prepare four datasets. The original Matterport3D dataset (M3D), the M3D dataset with Parametric Data Augmentation (M3DP), the M3DP dataset with the area and intersection checks (M3DPIA), and the M3DPIA dataset + 4 smallest items iteratively removed from a room to create four new rooms (M3DR4PIA). By conducting the object removal experiment, we aim to find which of the mentioned datasets achieve lower distance errors for each object category via the GSACNet model. As shown in Table 6.1, we find that the data augmentation proposed in this study effectively improves the scene augmentation workflow.

Comparative Studies

We compare the performance of our system with alternative learning models and also SceneGraphNet [257]. The results can be seen in Table 6.2 in which GSACNet outperforms alternative learning models, and [257] in nearly all categories. Details of the implementation of various models can be found in the supplementary material.

6.4 Conclusion

The network that we presented in this paper takes a novel approach to contextual scene augmentation through a Graph Attention and Siamese network architecture, followed by an autoencoder network and its implementation of parametric data augmentation of a 3D space with objects. We find that utilizing such a model improves the ability to augment virtual objects in plausible placements in a scene despite a small set of training data. By training on a parametrically augmented version of the Matterport3D dataset, we show our network architecture outperforms state-of-the-art scene synthesis networks such as [257]. Our work comes with a number of limitations. First, our current system does not conduct pose estimation for augmented objects. Moreover, in multi-object placement scenarios, the resulting predictions are highly dependent on the order in which objects are to be placed. Such an approach does not consider all possible combinations of the possible arrangements. Future work can comprise incorporating floorplanning methodologies with the current sampling mechanism allowing a robust search in the solution space while addressing combinatorial arrangement.

Chapter 7

Mutual Scene Synthesis

As previously explored in Chapter 3, due to the TSMP, a major challenge in developing telepresence systems is how to align and map virtual avatars within a target space, while addressing the spatial constraints of each user within their own local environments. Prototypes of high-fidelity telepresence systems [125, 131, 173] could avoid this challenge by placing remote users in an empty virtual space exclusively defined for the task. Considering natural locomotion as a key aspect of maintaining high-fidelity experiences, users can therefore only perform interaction and navigation tasks within their own boundaries. However, the method lacks rendering a mutual environment and does not hold spatial correspondence with its local surroundings for each participating user, limiting free body movement the use of mixed reality features such as pass-through objects and preventing the ability to interact with existing physical entities within the telepresence experience.

Since shared mutual environments can potentially play an important role in increasing productivity and social engagement. There has been a large body of literature focusing on capturing surrounding environments that can be utilized in spatial computing applications [145, 216, 209, 139]. Such captures can be used as a spatial background for telepresence avatars while matching their environmental lighting for additional photo-realism. However, attempting to capture detailed information from personal spaces can potentially cause privacy concerns and may unwillingly expose socioeconomic information of individuals during a telepresence call. Capturing public spaces (such as office spaces, conference rooms, or cafes) can also be integrated within the telepresence environment. Yet, this approach also lacks spatial customization and interaction with the physical environment itself, isolating the experience to predefined spaces or calibrated functions.

Studies have shown that users of immersive experiences report a higher sense of presence when a match between proprioception and sensory data is achieved [202]. It is due to this match that natural locomotion has been shown to be superior to other navigation methods such as teleportation, flying or utilizing game controllers [219]. In addition, from a safety perspective, as users of immersive environments are visually detached from their surrounding physical spaces, various techniques are utilized to inform the user of their physical surrounding, or deter them to prevent physical collisions. Alternative approaches aim to to generate a virtual experience that map to physical elements of the user's surrounding. For instance, a wall in the physical space would render as a barrier in the virtual environment, or a chair in the virtual environment would be sittable in the physical world. With

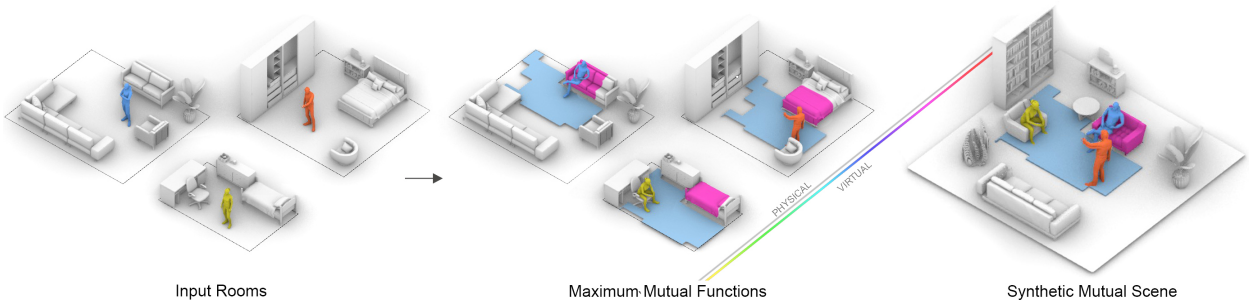


Figure 7.1: Mutual Scene Synthesis from three input rooms in a telepresence scenario. The system calculates optimal alignments to maximize mutual functional spaces, and furthermore generates a synthetic scene which incorporates the mutual functions with contextual placement of augmented objects.

all its potentials, such techniques do not extend to multi-user scenarios and cannot generate virtual environment that are adaptable to all participant's physical spaces.

Motivated by challenges mentioned above, we propose a contextual Mutual Scene Synthesis (MSS) system for spatial computing telepresence scenarios. Given a set of captured rooms, our proposed system generates a synthetic virtual scene that holds maximum functionality between the captured rooms and corresponds to their individual layouts. Users can safely navigate within the synthetic scene with natural locomotion and interact with mutual furniture that will have a physical correspondence in their surrounding local environment. Our work builds on an emerging body of literature on scene synthesis, while taking advantage of the works done in mutual spatial alignment for telepresence scenarios. As illustrated in Figure 7.2, when compared to alternative environment generation for telepresence scenarios, our mutual scene synthesis system enables a shared environment while maintaining privacy and spatial correspondence for each of the participants. We believe that utilizing this method can potentially facilitate spatial adaptations of next-generation computer-mediated communication platforms in spatial computing.

7.1 Introduction

7.2 System Overview

Figure 7.3 shows the workflow of our proposed system. The system takes the collection of rooms of the remote participants as input and generates a synthetic virtual scene with maximum mutual functions corresponding to the input rooms. Our proposed system consists of three main components: (i) Semantic Extraction: where a simplified semantic scene graph representation of the room is extracted; (ii) Mutual Scene Optimization: where the maximum mutual functions are calculated between the input rooms; and (iii) Mutual Scene Augmentation: where conditional scene

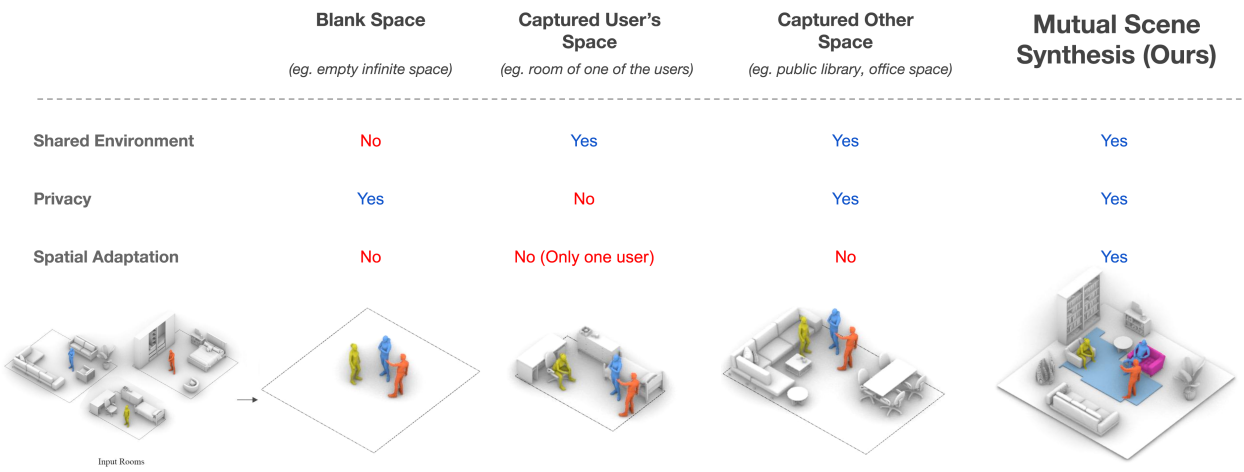


Figure 7.2: Comparison between various telepresence scenarios. Our proposed mutual scene synthesis system can allow remote users to share a mutual environment while maintaining privacy and spatial adaption of their local physical environment.

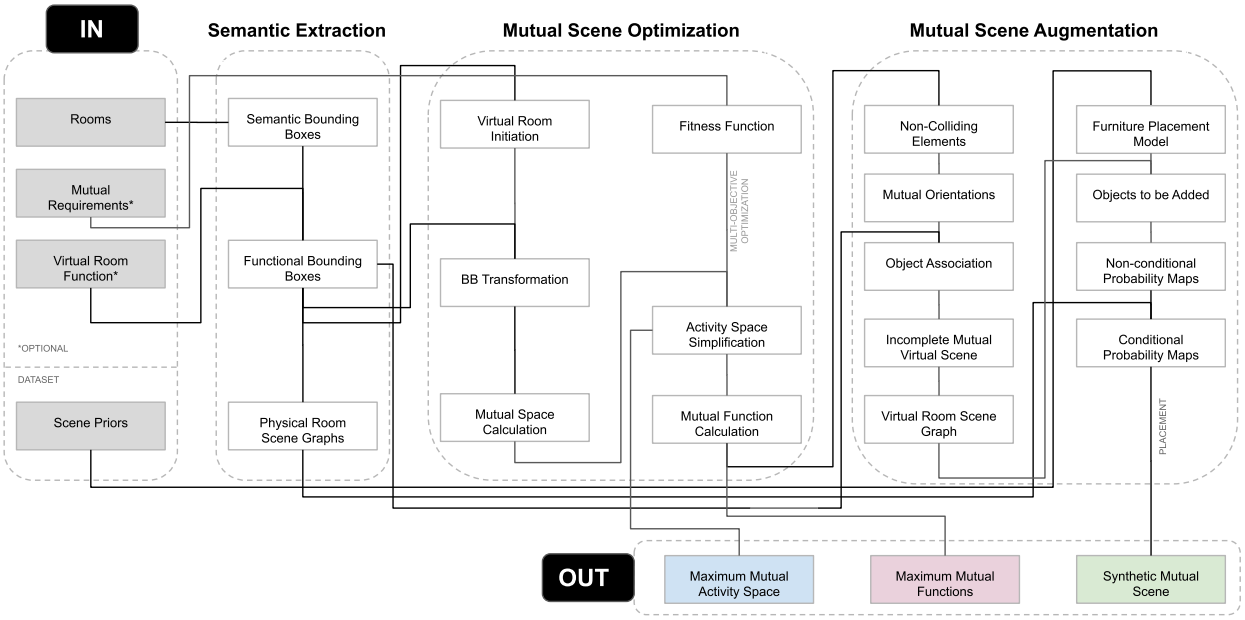


Figure 7.3: General workflow of the MSS system.

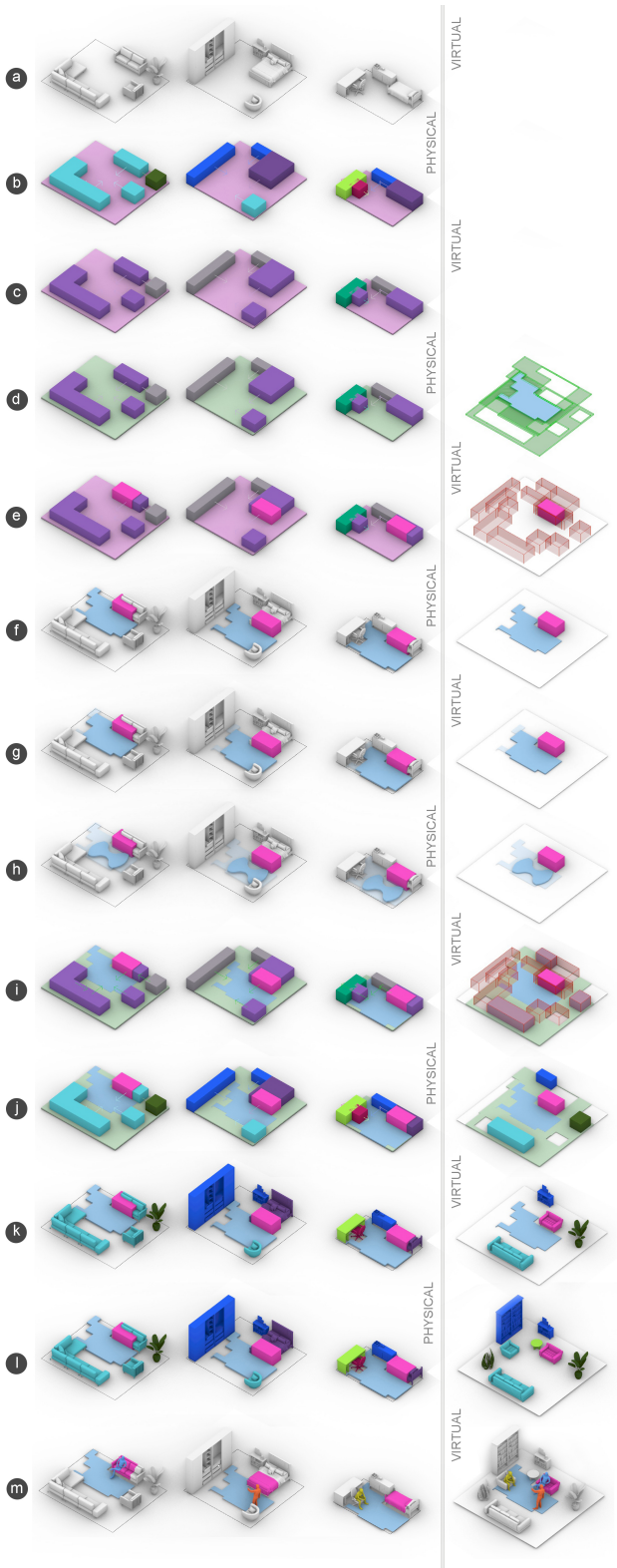


Figure 7.4: A step by step example of each of the modules in our MSS framework. General components include: (b) input rooms (b-c) semantic extraction (d-h) mutual scene optimization (i-l) mutual scene augmentation (m) synthetic mutual scene output.

augmentation is conducted using deep neural network models trained via scene priors. Figure 7.4 illustrates various steps of the mutual scene synthesis system using an example set of rooms. We discuss the details of each component in the following sections.

7.3 Scene Representation

Rooms and Objects

In this chapter, we define the room space R as an orthographic projection of its 3D geometry on the (x,y)-plane. We denote the k-th object (e.g., a chair or a bed) in R as O_k. The collection of n objects in R is denoted as $\mathscr{O} = \{O_1, O_2, ...O_n\}$. $B(O_k)$ represents the bounding box of the object O_k. Every object O_k has a label to classify its type. As our work requires multiple user spaces, we define for each user i their own room space expressed as R_i and the k-th object in R_i is denoted as $O_{i,k}$. Hence, the collection of all n_i objects in R_i is denoted as $\mathscr{O}_i = \{O_{i,1}, O_{i,2}, ...O_{i,n_i}\}$. Finally, we define the area function as $K(O)$.

We also differentiate between physical rooms and the virtual room in our notation. A virtual room is considered a room fully or partially rendered in mixed reality. We denote the virtual room as R' and the virtual objects as O'. In addition, we introduce a distance function $\delta(a,b)$ as the shortest distance between a and b objects. For example, $\delta(B(O_k),\dot{R})$ is the shortest distance between the bounding box of O_k and the center of the room R.

Semantic Scene Extraction

We consider the input to our system to also include semantically labeled bounding boxes. Semantic bounding boxes can either be defined manually by the user in MR [183], or be an output of automated semantic segmentation systems such as [165, 121, 3]. Therefore, every object O_k has a label to classify its functional type (see Figure 7.4. b). Furthermore, we define functional categories to describe objects and spaces with similar functional types. In our current implementation, each R_i can hold various functional categories of walkable (W_i), sittable (S_i) and workable (T_i) spaces. Walkable spaces consist of the area of the room in which no object located within a human user's height range is present. In walkable spaces, user movement can be performed freely without any risk of colliding with an object in the room. We calculate the available (W_i) for room R_i simply as follows:

$$W_i = R_i - \bigcup_{k=1}^{n_i} O_{i,k}. \tag{7.1}$$

Sittable and workable spaces correspond directly to a group of objects within a room. For example, chairs, sofas, beds, stools, etc. are all considered to have a sittable functionality. Objects such as desks, tables, etc. are considered workable functions. For R_i we have $S_i = \{O^s{}_{i,1}, O^s{}_{i,2}, ...O^s{}_{i,n_i}\}$, where $O^s{}_{i,k}$ is considered an object in R_i which holds a sittable function. A similar notation is

true for workable function groups defined as T_i. Figure 7.4. c) illustrates how functional semantic segmentation takes place in the input rooms.

Semantic Scene Graphs

To capture contextual topologies between objects of a scene, we represent rooms via semantic scene graphs. We utilize homogeneous scene graphs via the spatial relationship introduced in Chapter 4 to construct those scene graphs. Nodes in the scene graph represent objects, object groups, and the room; and edges represent the spatial relationships between the nodes allowing to describe the pair-wise topologies of objects and their relationship with the room. In the proposed scene graph representation, an explicit extraction of (a) positional and (b) orientational relationships take place by modeling descriptive topologies that are commonly utilized by architects and interior designers to generate spatial functionalities in a given space. Figure 7.5 illustrates an example of semantic scene graph collections for two input scenes. Note that each edge color corresponding to a spatial relationship represents a separate scene graph. Such a representation allows contextual scene augmentation to be utilized for an incomplete scene by training with previous scene graph priors. Further details of the scene augmentation process is discussed in Section 7.5.

7.4 Mutual Space Optimization

The goal of this module is to calculate optimal functional mutual spaces between participants by aligning the participants local spaces within the virtual environment. The mutual functional spaces generated in the virtual environment correspond to real-world functions in remote participants within their local environments. Such spaces are calculated by finding the optimal transformation function for each space to maximize the intersection of all spaces. We consider an immersive experience where there are m subjects and therefore m room spaces $(R_1, R_2, \cdots, R_m)$, respectively. Then, in the (x, y)-coordinates, we define a rigid-body motion in $\mathbb{R}^2$ as $G(F, \theta)$, where θ describes a translation and a rotation.

To maximize the mutual walkable space, we apply one $G(W_i, \theta_i)$to each individual walkable space W_i for the i-th user. The optimal rigid body motion then maximizes the area of the interaction space:

$$(\theta_1^*, \cdots, \theta_m^*) = \arg\max K(\bigcap_{i=1}^{m} G(W_i, \theta_i)). \tag{7.2}$$

Hence the maximal mutual walkable space can be calculated as

$$M_W(R_1, \cdots, R_m) = \bigcap_{i=1}^{m} G(W_i, \theta_i^*) \tag{7.3}$$

Mutual Functions

Similar to walkable spaces, our system calculates mutual areas of remaining functional categories namely mutual sittable (M_S) and mutual workable (M_T) spaces. The main difference between mutual walkable spaces and mutual function areas is that mutual functions require pose estimation. We use the following heuristic to define the pose of the calculated mutual functions: If the objects constructing the mutual function share the same pose direction, the mutual function area would also hold that pose direction. Else, the mutual function would be facing the center of the room. In Figure 7.4. e) mutual function optimization takes place, classifying a section of the sittable area of the sofa in R_1, and the bedspace in R_2 and R_3. Due to the fact that the pose of the bed in R_3 differs from the other corresponding sittable functions in R_1 and R_2, the resulting sittable space pose is calculated towards the center of the room.

Geometry Simplification

In certain scenarios, participants of a telepresence experience may require the mutual activity space to comply to a minimum area or hold a certain shape. Games for instance may require users to hold a safe play area, often being a quadrilateral or circular space to avoid physical conflicts. Another possible example is when users are surrounding and inspecting an object, and the activity space is preferred to be a circular shape with the object placed in the center. To this extent, mutual spaces solely calculated based on maximizing functional areas may hold non-convex peninsula-like geometry, which can become inaccessible for various activities. For instance, the mutual walkable space calculated in Figure 7.4. f) holds areas which a regular human body cannot perform free body movement without colliding with the boundaries of the space.

To address such scenarios, we add two optional post-processing modules to our workflow to generate safe activity spaces which allow: (a) simplification of the resulting mutual geometry to exclude peninsula-like areas (Figure 7.4. g); and (b) calculation of the largest custom activity shape inscribed in the mutual geometry (Figure 7.4. h). For simplification, a double-stage offsetting procedure takes place. In the first stage an inward offset with a distance of ε is conducted from the bounding polygon of the mutual space. Edges with more than two intersections are removed, resulting in a simplified inward offset of the shape. An outward offset with distance ε is followed as the second stage, generating a shape similar to the initial shape with excluded peninsula-like areas. The size of ε can be defined based on the activity. For instance, intense gaming applications that involve a high level of free-body movement would require a larger ε than a normal natural locomotion activity.

For calculating the largest inscribed custom activity space L_S, we define a rigid body function in $\mathbb{R}^2$ as $J(L,\theta,s_x,y)$, where L is the custom activity shape, θ describes a translation and a rotation, and s_x, s_y are scale factors applied to L is the x,y direction respectively. We run the following optimization:

$$\begin{aligned}(\theta^*, s_x^*, s_y^*) = \arg\max\,(&K(J(L,\theta,s_x,s_y)\bigcap M_W) \\ &- K(J(L,\theta,s_x,s_y)\bigcap M_W{}'))\end{aligned} \tag{7.4}$$

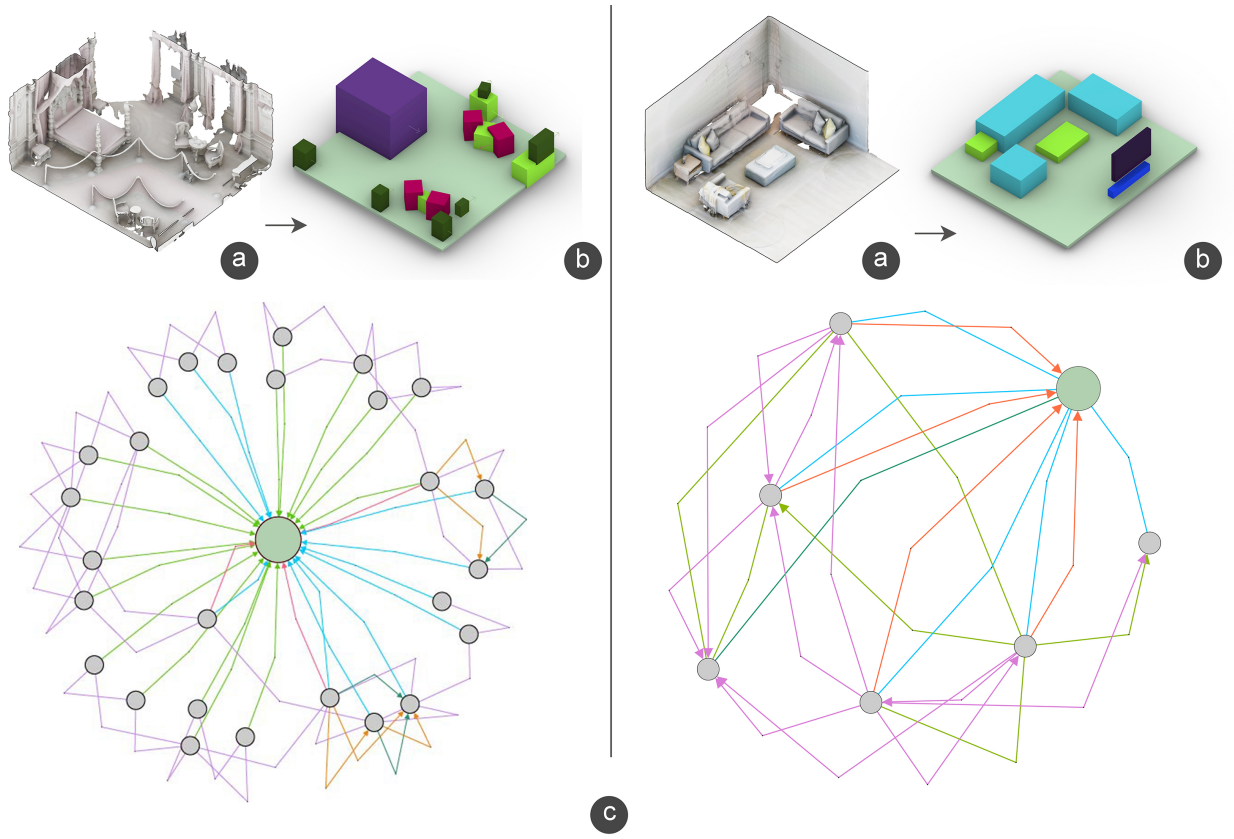


Figure 7.5: Examples of semantic scene graph extraction. (a) input room; (b) semantic segmentation (c) collection of semantic scene graphs. Semantic scene graphs represent pair-wise relationships between objects and the room.

Where $M_W{}'$ is the inverse of the mutual space $(R_i - M_W)$. Hence, the largest custom activity space is calculated as:

$$L_S = J(L, \theta^*, s_x^*, s_y^*) \tag{7.5}$$

Figure 7.4. h) shows an example of a optimization achieved to find the custom polygon inscribed in the mutual boundaries

Optimization

Considering various user-in-the-loop scenarios, optimizations can be defined as single objective or multi-objective problems. In telepresence settings that require only one mutual function type to be maximized, a single objective optimization is utilized to find the required transformation parameters for the room alignment. Alternatively, multiple functions can also have various weights and

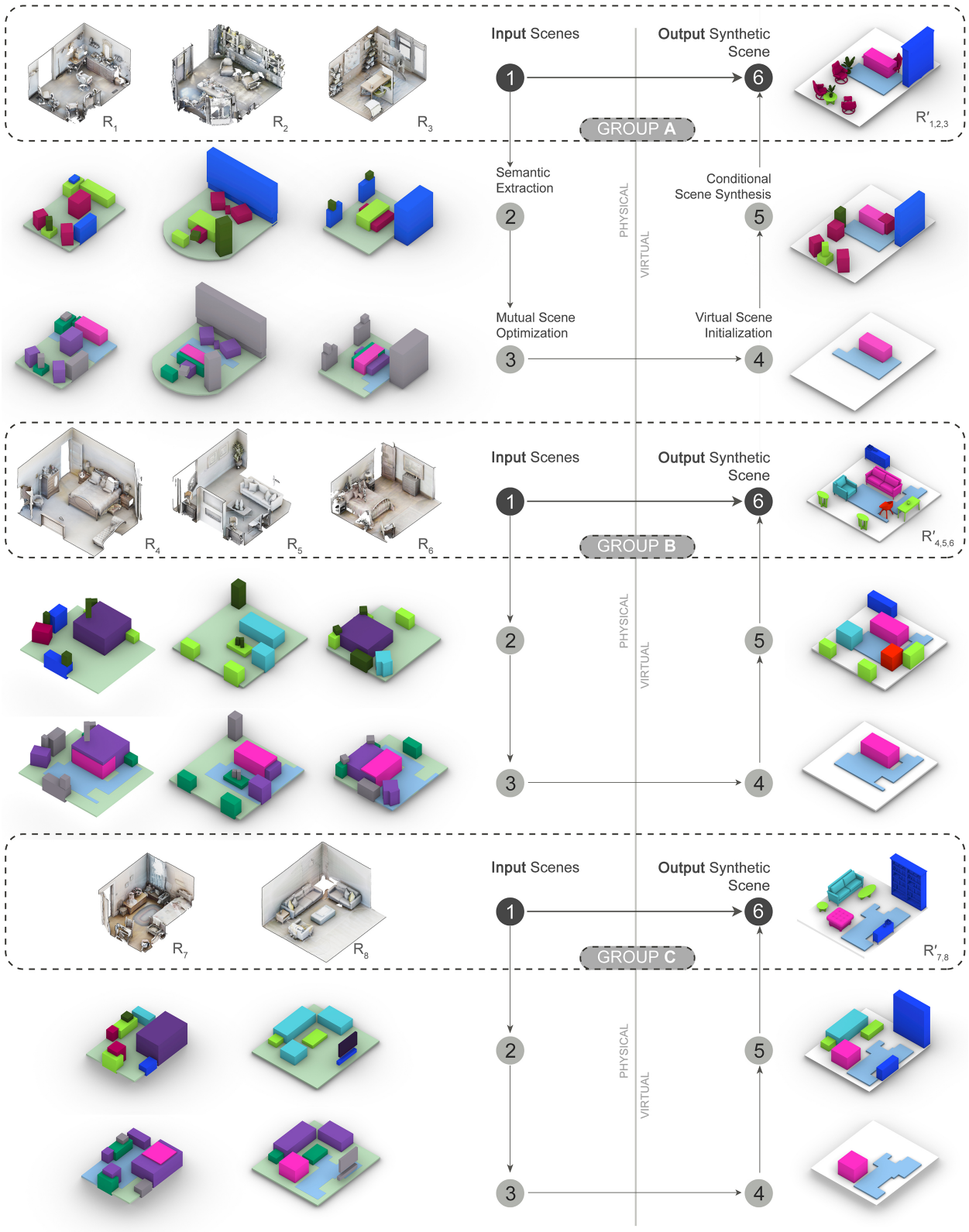


Figure 7.6: Results of the MSS system on MatterPort3D dataset examples.

constraints (such as minimum sittable or workable area), reducing the dimension of the optimization to a single-objective function. This approach was utilized in the mutual function optimization in Figure 7.4. d) and 7.4. e) where a minimum projected mutual sittable space of $1m^2$ was defined as a constraint, while maximization of the walkable space took place. Users can also be involved within the workflow for multi-function scenarios, where a set of solutions representing the Pareto frontier of the multi-objective optimization would be presented to the user. After considering trade-offs, the user can choose which spatial configuration would be more suitable for their activity.

7.5 Mutual Scene Augmentation

After calculating the optimal alignment of target rooms, we aim to synthesize a new virtual scene which would incorporate the mutual spaces and provide a plausible virtual environment spatially corresponding to all target users. The Mutual Scene Augmentation process consists of two modules: the first module utilizes a procedural grammar for initializing the scene, followed by the second module that uses scene priors for conditional scene synthesis.

Procedural Initialization

As a first step of the virtual scene initialization, we define the base floor of the synthetic room as the smallest circumscribed rectangle of the union of all the aligned rooms. This would guarantee users to access all their available physical space within the virtual environment. Furthermore, we populate the synthetic room with non-colliding elements of each local space (Figure 7.4. i) An object is considered non-colliding if (a) its transformed projection on the (x, y) plane does not collide with another room's walkable space and (b) its transformed position does not collide with another transformed object in the mutual alignment. Adding non-colliding objects to the virtual scene would add an additional visual barrier to prevent a physical collision for the user holding the object in its local space. Once the bounding boxes of mutual functions and non-colliding objects are calculated, the system takes on the task of associating each calculated bounding box to a function type and furthermore to a designated mesh. During the object association step (Figure 7.4. j), the function of the mutual room determines what objects should be placed in the scene synthesis step. The synthetic room function is an optional input given by the user of the system. If no input is given, the system uses the most repeated room function in the target set. If no majority is present, one of the room functions would be assigned randomly.

Conditional Scene Synthesis via Priors

As a final step, we use a deep-learning model to complete the room with additional furniture. The furniture is augmented in a conditional manner, taking into account relative furniture arrangements of input rooms. We utilize a modified version of GSACNet, discussed in Chapter 6 for the conditional scene augmentation process. GSACNet combines graph attention, siamese, and autoenoder networks to perform iterative scene synthesis for new or constrained scenes. For the training process, in

order to achieve robust results with limited scene priors, we propose utilizing the parametric data augmentation method introduced in Chapter 5. In this method, after building parametric floorplans of the rooms, boundary geometry and their adjacent furniture are constantly permutated while maintaining a set of functional constraints within the room.

For the scene augmentation process, the system initially samples points uniformly in the (x,y)-plane. Each point is considered as the center of possible placement for a target object $\dot{O}_k(x,y)$ on the ground floor plane, and forms its corresponding scene graphs discussed in Section 7.3. Next, the system passes feature vectors associated with nodes in the scene through an initialization neural network followed by a respective graph attention layer. Messages passed to the node associated with the object's furniture type are extracted and concatenate the messages per each scene graph with the summary vector. Furthermore, the concatenated vector is projected via a 4-layer network into a space such that data points representing plausible placements are clustered together while data points representing implausible placements are separated from the cluster. Finally, we output a probability of plausible placement P using the reconstruction error produced by the an autoencoder. Studies have shown autoencoders to perform well as anomaly detectors [259]. In our scene synthesis system, there is a model per furniture group. The system trains each model using two separate training phases. In the first phase, the initialization, scene graph extraction, and project modules are trained as one large siamese network, with a siamese network projection module. In the second phase, the outputs of the first training process are used as input and train the autoencoder module alone.

In a conventional scene synthesis scenario, $\dot{O}_k(x,y)$ is placed in the location with the highest P. Instead, in our approach, we add an additional conditional module to allow contextual placements to take into account the arrangement of the real-world user target scenes in addition to the generated synthesized scene. The conditional module takes the n top samples of P and sorts them based on their distance to the closest object in the same functional type from all the input physical spaces. In simple terms, from the placements that the scene synthesis module considers plausible, the system chooses the final placement based on its proximity of real world objects in one (or more) of the real-world user spaces. Such an approach would assist the scene augmentation process to place objects closer to where they are in the real-world, potentially corresponding to one of the target room furniture arrangements. Hence, slightly contrary to conventional scene synthesis systems, our proposed approach populates the virtual scene by placing objects close to their real-world setting while being contextually relevant to the mutual virtual scene (Figure 7.4.l).

7.6 Experiments

Synthetic Generation via Real-world Datasets

To evaluate how our proposed mutual scene synthesis system performs with various room types and different spatial organizations, we utilize available 3D datasets from captured real-world scenes as case studies. We use the Matterport 3D [30] dataset and sample subsets of varying size and functions of rooms, to observe how mutual spaces are optimized and the corresponding synthetic

scene is generated. Matterport 3D is a large-scale RGB-D dataset containing 90 building-scale captured models. The dataset consists of various building types with diverse architecture styles, each including numerous spatial functionalities and furniture layouts. Human-defined annotations of building elements and furniture are provided with surface reconstructions as well as 2D and 3D semantic segmentation. We utilize this data for the semantic segmentation process. In addition, we exclude spaces that are not typically used for telepresence spaces (bathroom, small corridors, stairs, closet, etc.).

For the mutual function optimization procedure, we utilize a Strength Pareto Evolutionary Algorithm 2 (SPEA 2) [258] algorithm to calculate the maximum mutual functions between the rooms. We use a population size of 100, mutation probability of 10%, mutation rate of 50% and crossover rate of 80% for our search. As our solution integrates a evolutionary search, we expect the result to gradually converge to the global optimum. We stop the optimization after 80 generation runs. Room translations are executed in 10cm steps in the (x, y) plane and 15° orientation gains for the optimization process. For our conditional scene synthesis module, we train our model on the same dataset excluding the input rooms used in our experiment. As the MatterPort3D dataset does not offer pose annotation, we use the rapid-annotation tool in Chapter 4 to label pose data within the scenes.

Figure 7.6 shows the results of three sets of real-world captured rooms, each including rooms with different room sizes and functions. After extracting semantic labels of the objects (steps 2,3), the system performs mutual function optimization with functional semantics (step 3,4). Our proposed system is able to locate mutual walkable, sittable and workable functions in target rooms and align the physical environments to maximize the mutual functions. Furthermore, the system aims to complete the initialized synthetic rooms with the conditional scene synthesis process (steps 5,6).

User Studies

In a comparative user study, we aim to measure the participant's ability to find the maximum mutual functions between the rooms and compare it with the outcomes of our proposed mutual function system. We recruited 25 participants (m=10, f=15), which were skilled in 3D annotations to find mutual walkable and sittable functions of groups of rooms. We utilized 17 rooms of the MatterPort3D dataset which were organized in groups of three, and one group of two. We developed a 3D annotation application, which allowed participants to view all the rooms of the group in 3D, and annotate what they believed is the mutual areas between them. The tool also allowed the modification of annotated geometry after initial annotation. Participants were not aware that they were going to be compared to an automated system and were just asked to provide their best annotation skills for the task. Before data collection, the experiment operator demonstrated an example of how to use the annotator tool and answered questions on what is considered a mutual space. The data collection process from each participant took approximately 30 minutes, allocating 5 minutes to each room group for indicating mutual walkable and sittable spaces.

However, as anticipated, participants where not able to annotate 3 exactly similar areas in all three rooms. Therefore, in our analysis, we performed an extra step of aligning the user annotated

Figure 7.7: Results of the comparative user study showing top 3 human classification of walkable spaces (M_U) compared with the MSS system (M_W).

spaces in a brute-force search process. The polygons are centred in a mutual point, and an exhaustive search is conducted between all possible orientations of the polygons to calculate the maximum intersection between them. We denote the maximum intersection as M_U and further compare to M_W and M_S predicted by our system. The optimization implementation of the system were similar to what was described in Section 7.6.

Figure 7.7 shows the top 3 highest mutual area classification task (out of 25) for walkable spaces

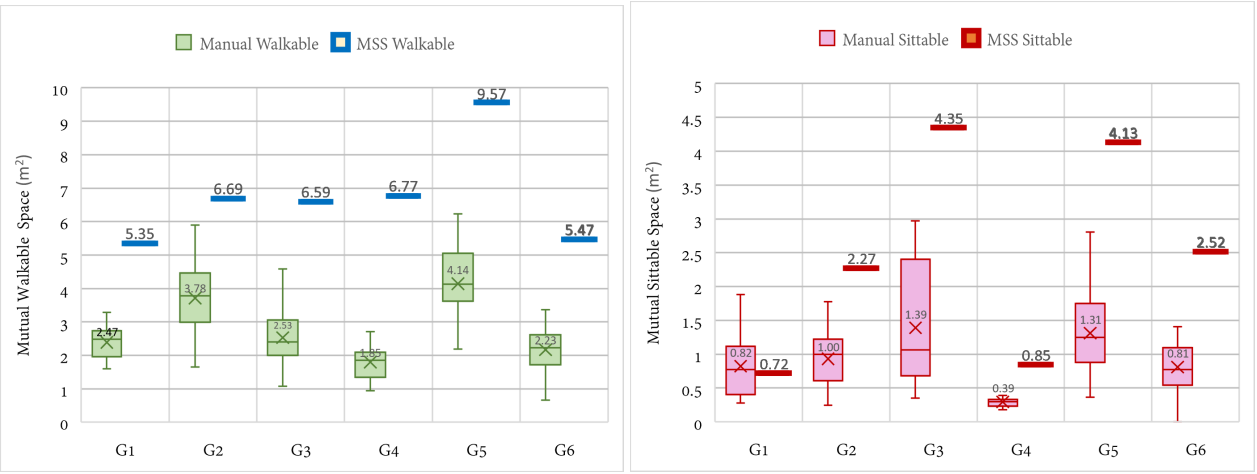


Figure 7.8: Comparison between manual annotations (M_U) and our MSS system for indicating mutual walkable (M_W) and sittable spaces (M_S) in 6 Matterport3D room groups

performed by the users in green, compared to the system's calculation M_S illustrated in blue for all six groups. As seen in the figure, the automated system clearly outperforms user performance in this classification task. A common technique that was observed is that most human annotators aimed to start with the smaller room and try to find corresponding spaces in the other rooms. This of course requires numerous editing attempts for the mutual space geometry to be modified accordingly.

Figure 7.8 shows a numerical comparison of the mutual area indication task between human-annotators (M_U) and our MSS system for walkable and sittable spaces. For each room group, we plot a whisker-plot to visualize the distribution of M_U for all participants, while a thick line represents MSS calculation. As seen in the figure, our system significantly finds larger areas of mutual spaces than human annotators with an average increase of 58.68% for walkable spaces and 56.00% average increase for sittable spaces.

7.7 Discussions

As presented in our results from real-world captured room examples, furniture topology in the resulting synthetic scene often corresponds to objects present in physical environments. For instance in Figure 7.6, in Group A, chairs and tables correspond to the location of office space R_1, while the storage space can also be attributed to R_3. All rooms hold part of their desk space as a mutual workable space. In Group B, a mutual sittable space is extracted from the area attributed to the bed in the bedrooms and a the larger sofa in the living room. In Group C, we see how small spaces such as the bedroom R_7 can also contribute to generating plausible spaces using our system. While the mutual spaces are considered limited, yet the resulting synthetic scene has

utilized non-colliding functions from R_8 in its procedural generation before completing the space with additional contextual furniture.

In the user study, there are a number of exceptional instances that participants classify larger walkable spaces than our automated system. This is because (a) we do not cross-validate the participants annotation as we consider any walkable or sittable area defined by the user to be correctly annotated (b) the system uses annotated labels from the dataset which are also annotated by humans and prone to error. However, since manually defining constant shapes between all rooms using our tool was seen as challenging, the actual mutual space post-processed by an exhaustive search module resulted in a significantly smaller area than each annotated room. Part of the classification inconsistencies could be attributed to the limitations in the annotation tool (eg. duplicating the annotation from one room to the other was not possible), hence, enhancing workflows to improve user classification could have changed the outcome of the experiment.

For the conditional scene synthesis module, a major challenge when relying on learning-based methods is that they are heavily biased towards the data. While the initial phase of our proposed scene augmentation module integrates a procedural approach, the final steps include populating the scene with additional furniture learned for scene priors. Real-world spaces are not generally designed for hosting virtual users. Hence, defining which scenes from the dataset are suitable for a meeting setting can be a challenging process. Models can be trained to filter room functions such as meeting-room spaces and offices-space, however, many spaces cannot be specifically classified to hold a single room functionality. For instance, a captured space from a studio or a dining room can serve as multiple functions. Another limitation of real-world datasets is their low label accuracy due to the labour intensive manual annotation process.

7.8 Conclusion

In this chapter we have presented a method for synthesizing a virtual environment for telepresence settings which corresponds to the spatial arrangement of the participants' physical local environments. Our method aims to calculate the maximum mutual walkable, sittable and workable spaces between users, allowing the synthesized virtual scene to hold areas of mutual ground for efficient virtual interaction. We utilize state-of-the-art scene synthesis methods to populate the virtual room with objects that hold topological and functional relationships with elements of the scene. We extend the scene augmentation process by introducing a conditional mechanism, allowing virtual objects to position themselves close to objects with same functionalities in the physical environment.

Our proposed system comes with a number limitations and failure cases. In scenarios with a large number of participants, the mutual space optimization module may fail to locate mutual function spaces that are present in all participants' spaces. In such cases, the current system relies on the procedural module to initialize a virtual scene using non-colliding functions. If this step is also implausible due to the furniture arrangement of target rooms, the system can fail in generating a mutual space. An alternative mechanism is to locate mutual spaces for subgroups, and initialize the scene augmentation process from output of the subgroup mutual space. This, however, would

significantly increase the complexity of the optimization, as the system would need to initially search for the best subset of rooms.

Chapter 8

Conclusion

8.1 Mutual Space Finding and Optimization

In this dissertation, we discuss how calculating the maximum mutual space between multiple remote participants in SC telepresence scenarios can be a solution to the TSMP. When utilizing the proposed mutual space optimization techniques, users avoid geometrical and line of sight conflicts and can freely move around the space via natural locomotion and interact with each other's remote avatars. In Chapter 3 we define a mathematical formulation for calculating walkable and sittable spaces, while Chapter 7 introduces a more generalized formulation for mutual functions. By using various optimization methods, we show how our proposed workflow maximizes mutual spaces to enhance user interaction between one another. We evaluate the proposed systems by performing experiments on 3D captures of real-world rooms and demonstrate plausible results that can be utilized by users in various room functions and spatial organizations.

Furthermore, custom activity spaces can be located and maximized using the optimization framework introduced in Chapter 7. Such an approach would allow a more targeted calculation of the remote mapping procedure to enhance the user experience for custom activities and increase safety for games that involve a lot of body movement. Moreover, we formulate mutual area simplification procedures for mutual space geometry to allow the optimization procedure to avoid calculating peninsula-like areas which cannot be accessible during natural locomotion. In addition to calculating the mutual space, the recommendation system developed in this dissertation can serve as a spatial guide for the user to re-arrange the furniture in their rooms to gain larger mutual spaces with minimum effort for telepresence workflows. Our results show significant increases in mutual spaces can be found when utilizing the proposed recommendation system. We also evaluate whether the proposed mutual space finding system offers any benefits compared to manual mapping by the users themselves. The results of the user study in Chapter 7 show that our system significantly finds larger areas of mutual spaces than human annotators, with an average increase of roughly 58% for walkable spaces and 56% increase for sittable spaces. Regardless of annotation accuracy and time, we observe manually aligning mutual spaces as a challenging task for individuals. Identifying an acceptable boundary and checking whether all rooms comply with the defined geometry can

take multiple iterations of modifications. Such a process is time-consuming and can be potentially challenging to execute for novice users in spatial computing platforms. The task may become more challenging if we consider privacy concerns, preventing users from viewing other participant spaces during the telepresence setup. In the absence of mutual space generation systems, users would need to communicate with each other to find suitable conditions that would address all spatial needs.

8.2 Context-aware Virtual Scene Augmentation

Another aspect explored in this dissertation is the ability to augment scenes with virtual objects that can adapt to the context. In what we refer to as *contextual scene augmentation*, given a real-world scene, the goal is to add one or multiple virtual objects while maintaining topological and functional relationships between the objects, users, and the room. This dissertation introduces three methodologies for the scene augmentation task. All three hold a similar property of representing the scene as an explicit semantic scene graph, which we formulate as part of the contributions of this dissertation. By learning from example scene graphs, either extracted from 3D indoor datasets or inputted by the user itself, the developed contextual scene augmentation systems can calculate a plausibility map of where to place and orient a given object in a scene. SceneGen, introduced in Chapter 4, utilizes a kernel density estimation to build a multivariate conditional model for the scene augmentation process. This approach is improved with the formulation of GSACNet, introduced in Chapter 6, where we combine graph attention, siamese, and autoencoder networks to perform iterative scene synthesis for new or constrained scenes. Finally, in Chapter 7, we introduce a conditional scene synthesis method that adds an additional bias toward the scene augmentation process to correspond to remote user's furniture arrangements in a telepresence setting.

While some aspects of the scene augmentation task overlap with the task of scene synthesis studied in computer graphics literature, there are many identifiable differences between the two. Scene synthesis comes with the overall goal of generating a plausible scene, while our approach initially aims to add defined objects to an already constructed scene in a curated fashion. Our approach also covers virtual object placement of abstract entities that are not necessarily seen in everyday scenes. For instance, if an augmented reality content developer intends to augment a custom dragon in a set of target spaces unknown to the developer, how can this be done? Using methods introduced in recent state-of-the-art scene synthesis literature, where implicit models are trained through large datasets, addressing the task above is challenging, if not impossible. There is no large dataset that contains a dragon placement in a room, and even if there were, there is no guarantee that the placements are aligned with the custom definition of the content developer. However, when using explicit models utilized in this dissertation, which can be trained with limited data points, the problem above can be potentially addressed. A content developer can provide limited examples while knowing what attributes the system would be measuring for the scene augmentation. Nevertheless, scene augmentation can be combined with other scene generation components to perform scene synthesis, something we introduced in Chapter 7 as mutual scene synthesis.

8.3 Generating Mutual Experiences

Next-generation SC platforms can potentially create new modalities of workplace and collaboration environments. Using SC telepresence platforms, virtual meetings, gatherings, and meetups can take place without occupying a designated physical space, such as an office, designed and built only to accompany a meeting function. One of the main goals of this dissertation is to facilitate the process of creating mutual experiences in SC platforms. Followed by the formulation of mutual scene optimization procedures and the ability to complete scenes with contextual scene augmentation, in Chapter 7 we combine the two techniques to present an MSS method for synthesizing a virtual environment for telepresence settings which corresponds to the spatial arrangement of the participants' physical local environments. MSS aims to calculate the maximum mutual walkable, sittable, and workable spaces between users, allowing the synthesized virtual scene to hold areas of mutual ground for efficient virtual interaction. We utilize a scene augmentation method to populate the virtual room with objects that hold topological and functional relationships with various scene elements. We extend the scene augmentation process by introducing a conditional mechanism, allowing virtual objects to position themselves close to objects with the same functionalities in the physical environment.

Our experiments demonstrate our proposed mutual scene synthesis method in action using real-world captured rooms as inputs to our system. By using our proposed method, meaningful spaces suitable for meeting spaces are synthetically generated while holding mutual functional areas for users to utilize. Furthermore, by performing a series of user studies to compare task performance between manual and automated mutual space classification, we show our proposed system can locate significantly larger mutual spaces in a fraction of the time. Therefore, an automated system to generate synthetic spaces can potentially facilitate the adaption of mixed reality telepresence platforms.

8.4 Scaling Spatial Computing Experiences

As discussed in Chapter 1.2, one of the main hurdles in SC development is the inability to scale curated experiences that can adapt to a large number of user spaces. This dissertation introduces a number of methods to facilitate this challenge by developing systems that can serve as an automated middle-man between the content developer and millions of users. The context-aware scene augmentation and scene synthesis process can allow curated SC experiences to be generated by developers while preserving the spatial privacy of end-users. This would allow a content developer to develop a single SC program or application; using the methods introduced in this dissertation, the program itself can potentially scale to be adopted by millions of users, all with different spatial organization and room layouts unknown to the content developer. Furthermore, this dissertation constitutes an attempt to explore how adaptable SC experiences can be generated and curated through a system learning from examples. This would potentially allow content developers to provide examples of their desired scene arrangements and typologies instead of hard-coding rules for a large set of target users. Such an approach can itself play a role in increasing the adaption

of SC interfaces for a wide range of applications as developing adaptable SC experiences would require less programming skills and instead be developed by providing desirable examples for the system.

As a significant part of our everyday activities is conducted in social contexts, the framework introduced in this dissertation can facilitate virtual collaborations and remote workplace practices by decreasing spatial requirements for telepresence systems. Telepresence applications can utilize the proposed mutual space and function optimization methods introduced in this dissertation to allow a scalable number of participants to hold telepresence experiences between each other without being affected by the TSMP. Instead of setting up large physical spaces required for meetings and collaborations, the system would allow users to join from their personal spaces, with minimum modifications to their surrounding environment. Telepresence participants are not aware of remote users' space, and similar to the content developer discussed above, hold a core challenge of *unknown spatial layout of the target user space*. Hence, this dissertation aims to facilitate the efforts for scaling SC experiences for the masses by addressing this core challenge with the proposed algorithms and paving the way towards *Responsive Spatial Computing*.

8.5 Outlook and Future Work

This final section highlights what the author believes as possible future research explorations based on the results and observations gained from this dissertation.

Multi-object Inferencing

In the context of generating scalable curated experiences via scene augmentation, one may require to augment multiple objects within the scene as part of the content design process. The scene augmentation methods introduced in this dissertation inference objects one at a time, resulting in an iterative scene augmentation process. This itself is considered a limitation of our workflow since the layout is dependent on the order of the object placement and does not calculate all possible permutations of the possible arrangements. Such an approach can narrow down the possible open spaces for later objects, forcing placements that are far from optimal. Moreover, in scenarios where a large number of objects are to be augmented, or the scene is changed due to external modifications, the current approach may not have the ability to *fit* all the objects within the usable space as initial placements are not aware of upcoming objects. Hence, a more comprehensive search method needs to be utilized to efficiently search all the possible permutations of the order in which objects are placed. Due to the combinatorial enumeration of spatial relationships and the subjective nature of their inter-dependencies, such a problem is considered an NP-Hard problem, as the size of solution spaces grows exponentially as the size of the objects to be augmented increases.

Future work can comprise incorporating space layout methodologies with the current sampling mechanism allowing a robust search within the solution space while addressing combinatorial arrangement with virtual a physical objects. Space layout planning is the process of placing a set of discrete but independent spatial elements while attempting to satisfy geometrical, topological, and

performance goals in their layout. Applications of space layout planning are not limited to indoor scene augmentation and can be seen in a wide variety of fields, such as integrated circuits [200], architecture, urbanism, and operational research [2]. The inherent level of complexity in space layout planning has encouraged researchers to also explore this problem by developing generative systems that take advantage of various learning-based and meta-heuristic search approaches. Similar to the workflows explored in this dissertation, the general process of space layout methodologies can be described in three modules (i) generating various layouts via a generative function, (ii) analyzing the layouts using certain design objectives via a fitness function, or comparing them to a set of priors; (iii) iterating this search until the optimal solutions are found [92]. Generating layouts often involves encoding a layout as a solution to a layout representation function. This representation not only defines the complexity of the solution but also impacts the efficiency of the search process to find a desired floorplan in the solution landscape. Hence, exploring scene graph representations introduced in this dissertation as an encoding mechanism for the generative models, and integrating them with multi-objective optimization procedures, can be seen as a possible future research direction for addressing the multi-object inferencing problem.

Overcome Learning Bias

The methods introduced in this dissertation take advantage of both procedural modeling and learning-based model training techniques for the contextual scene augmentation tasks. As procedural modeling consists of hard-coded rules, the logic behind the scene augmentation procedure can be understood as the system follows a certain interpretable procedure to generate the results. In contrast, learning-based methods, especially techniques that utilize deep neural networks, are challenging to interpret. It is hard to logically narrow down the exact reasoning of the system during the inferencing phase from a previously trained model. While this dissertation highlights the necessity for using explicit scene graph representations as opposed to implicit representations for contextual scene synthesis, there is still a large degree of bias towards the input data that is fed to the system as scene priors.

One important consideration in our choice of utilizing the MatterPort3D dataset in our work is that we aim to learn spatial relationships for real-world scenes. One can imagine idiosyncrasies of lived-in rooms, such as an office chair that is not always tucked into a desk but often left rotated away from it or a dining table pushed into a wall to create more space in a family room. Using personal living spaces, from the Matterport3D dataset, as our priors, we can capture these relationships that exist only in the real world, lived-in scenes. Yet, one drawback of using real-world datasets such as the Matterport3D dataset is that it is not as large as some synthetic datasets. For instance, the SUNCG [204] synthetic dataset, which was unavailable during the course of our study, holds more than 45,000 environments, while our models were trained on only 1,326 rooms from the Matterport3D dataset. In our implementation, we were forced to group objects into broader groups to ensure adequate representation to ensure that all object categories are represented well enough to approximate the distribution of large feature space. A larger dataset would have allowed us to model more diverse object categories in a data-driven approach.

Current real-world datasets also come with a number of cultural biases. First, they are often geographically limited to a certain building type, area, or country. MatterPort3D, for instance, consists of large houses located in the United States only. Moreover, an average of 22.8 rooms are scanned in each house, which usually includes multiple bedrooms, bathrooms, hallway spaces, etc. This rate is far higher than the average of nine rooms per house in the United States [78], which can indicate the houses are owned by wealthy occupants. Therefore, furniture arrangements captured in this dataset may not accurately correspond to average-sized rooms and their layouts. Residential spaces with limited areas, such as apartments and dorms, are not seen in the dataset. Such limitation may cause biased results when the target scenes are spaces with limited open areas. Another downside of using a real-world dataset is its accuracy in labeling, where many human errors occur in this labor-intensive process. Such mismatches are unlikely to happen in synthetic datasets as the geometry is already assigned in a digital format. To mitigate some of these concerns, we have developed a labeling application that allows us to determine the correct orientation of each object and also filter out rooms with corrupted scans and inaccurate labeling.

On the other hand, a critical drawback of synthetics datasets is that they cannot capture the natural transformation and topological properties of objects in real-world settings. Furniture in real-world settings is a product of the gradual adoption of space, contributing to the functionality of the room and surrounding items. Topological relationships between objects in real-world scenes exceed the design assumption of a designer and capture contextual relationships from a living environment. Moreover, the limitations of the modeling software for synthetic datasets can also introduce unwanted bias to the generated scenes. The SUNCG dataset, for instance, was built with the Planner5D platform, an online tool that any user around the world can start using. The software workflow is similar to building a house in The Sims, where simple functions allow users to create rooms and add objects from an internal library. However, like any other design software, it comes with modeling limitations for generating rooms and furniture. Orientations are also snapped to right angles as default, which has made most scenes in the dataset Manhattan-based. More importantly, there is no indication if the design is complete or not. Users may have started to play with the software and leave the platform, while such arrangement was captured as a legit human modeled arrangement for the dataset.

In addition to synthetic and scanned 3d datasets, future work can consist of utilizing novel computer vision techniques to predict 3D bounding boxes from 2D images. There has been a large body of research aiming to extract semantic 3D bounding boxes from monocular images [260, 108, 238, 253, 254, 76]. Such an approach would allow priors to be trained by more diverse 2D indoor scene datasets available through online repositories or image search engines. As the knowledge models proposed in this dissertation is independent of the input dataset, the integration of such systems can be considered as a possible future alternative to current RGB-D datasets. An example of using 2D images for 3D scene synthesis can be seen in the work of [129] where they extract the relationship between human-based activities and various furniture to build a human-actively model for the synthesis generation.

Usability Studies

Finally, conducting thorough user experiments by developing additional features in the mixed reality prototypes introduced in this dissertation can help identify the challenges of such frameworks from a user standpoint. For instance, exploring effective techniques for users to interact with a synthetic scene generator while allowing them to modify and adjust the output of such systems can be possible future directions. Moreover, usability studies can be performed to identify the best strategies for user-in-the-loop input when the system offers more than one option for spatial arrangement or manipulation. In our developed mutual function optimization framework in Chapter 7, we propose utilizing multi-objective optimization processes, which in theory generate a selection of Pareto-optimal solutions instead of a single optimal instance. Hence, a user should guide the system in the trade-off process by choosing what objective functions to prioritize in the multi-objective criteria. Another alternative is to visualize all the Pareto-optimal options and allow the user to choose between them. This itself would require additional research and user experiments to understand the efficient visualization practices for user performance in such tasks. Furthermore, for telepresence scenarios, improving the framework to address scenarios where multiple users are present in each space can be explored, as currently, only one user is considered to be present in each space. When multiple users are present, challenges such as spatial audio and virtual geometrical conflicts would need to be resolved to ensure a realistic experience for telepresence participants. Lastly, future usability studies can be conducted to further improve the visualization strategies for the mutual scene optimization workflows so participants can feel present in the telepresence experiences without getting distracted from maintaining their spatial position in the allocated mutual ground.

Future studies mentioned above, along with similar research avenues that focus on resolving spatial limitations of SC can pave the way to facilitate large-scale contextualized deployment and the enhancement of user experiences within SC systems. Creators can develop and curate virtual experiences for millions of users worldwide, regardless of their geolocation and local spatial constraints, opening the doors for next-generation SC interfaces to become the dominant computing platform in various everyday applications.

Bibliography

[1] Sue Abdinnour-Helm and Scott W Hadley. "Tabu search based heuristics for multi-floor facility layout". In: *International Journal of Production Research* 38.2 (2000), pp. 365–383. ISSN: 0020-7543.

[2] Miguel F. Anjos and Manuel V.C. Vieira. *Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions*. Aug. 2017. DOI: 10.1016/j.ejor.2017.01.049.

[3] Iro Armeni et al. "3d semantic parsing of large-scale indoor spaces". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1534–1543.

[4] Iro Armeni et al. "Joint 2d-3d-semantic data for indoor scene understanding". In: *arXiv preprint arXiv:1702.01105* (2017).

[5] Asit Arora et al. "Virtual reality simulation training in Otolaryngology". In: *International Journal of Surgery* 12.2 (2014), pp. 87–94.

[6] Autodesk. *Project Dreamcatcher | Autodesk Research*. URL: https://autodeskresearch.com/projects/dreamcatcher (visited on 11/19/2019).

[7] Armen Avetisyan et al. "Scan2cad: Learning cad model alignment in rgb-d scans". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 2614–2623.

[8] J Timothy Balint and Rafael Bidarra. "A generalized semantic representation for procedural generation of rooms". In: *Proceedings of the 14th International Conference on the Foundations of Digital Games*. ACM. 2019, p. 85.

[9] Jakob E Bardram. "Activity-based computing: support for mobility and collaboration in ubiquitous computing". In: *Personal and Ubiquitous Computing* 9.5 (2005), pp. 312–322.

[10] May Bassanino et al. "The impact of immersive virtual reality on visualisation for a design review in construction". In: *Proceedings of the International Conference on Information Visualisation* (2010), pp. 585–589. ISSN: 10939547. DOI: 10.1109/IV.2010.85.

[11] Stephan Beck et al. "Immersive group-to-group telepresence". In: *IEEE Transactions on Visualization and Computer Graphics* 19.4 (2013), pp. 616–625.

[12] Hrvoje Benko, Ricardo Jota, and Andrew Wilson. “MirageTable: freehand interaction on a projected augmented reality tabletop”. In: *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM. 2012, pp. 199–208.

[13] Leif P. Berg and Judy M. Vance. “An Industry Case Study: Investigating Early Design Decision Making in Virtual Reality”. In: *Journal of Computing and Information Science in Engineering* 17.1 (2016), p. 011001. ISSN: 1530-9827. DOI: 10.1115/1.4034267. URL: http://computingengineering.asmedigitalcollection.asme.org/article.aspx?doi=10.1115/1.4034267.

[14] Michael S Bergin et al. *Automated parametrization of floor-plan sketches for multi-objective building optimization tasks*. US Patent App. 16/681,591. May 2020.

[15] Benjamin Bolte and Markus Lappe. “Subliminal reorientation and repositioning in immersive virtual environments using saccadic suppression”. In: *IEEE transactions on visualization and computer graphics* 21.4 (2015), pp. 545–552.

[16] Hervé Bourlard and Yves Kamp. “Auto-association by multilayer perceptrons and singular value decomposition”. In: *Biological cybernetics* 59.4 (1988), pp. 291–294.

[17] Jane Bromley et al. “Signature verification using a" siamese" time delay neural network”. In: *Advances in neural information processing systems* (1994), pp. 737–737.

[18] Gerd Bruder, Paul Lubos, and Frank Steinicke. “Cognitive resource demands of redirected walking”. In: *IEEE transactions on visualization and computer graphics* 21.4 (2015), pp. 539–544.

[19] Gerd Bruder et al. “Tuning self-motion perception in virtual reality with visual illusions”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.7 (2011), pp. 1068–1078.

[20] Christina E Buckley et al. “Is the skillset obtained in surgical simulation transferable to the operating theatre?” In: *The American Journal of Surgery* 207.1 (2014), pp. 146–157.

[21] Richard W Bukowski and Carlo H Séquin. “Object associations: a simple and practical approach to virtual 3D manipulation”. In: *Proceedings of the 1995 symposium on Interactive 3D graphics*. 1995, 131–ff.

[22] Inês Caetano, Luıs Santos, and António Leitão. “Computational design in architecture: Defining parametric, generative, and algorithmic design”. In: *Frontiers of Architectural Research* (2020).

[23] Luisa Caldas and Mohammad Keshavarzi. “Design Immersion and Virtual Presence”. In: *Technology| Architecture+ Design* 3.2 (2019), pp. 249–251.

[24] Onur Çaliskan. “Virtual field trips in education of earth and environmental sciences”. In: *Procedia-Social and Behavioral Sciences* 15 (2011), pp. 3239–3243.

[25] Fadi Castronovo et al. “An evaluation of immersive virtual reality systems for design reviews”. In: *Proceedings of the 13th international conference on construction applications of virtual reality*. Vol. 47. 2013.

[26] Kynthia Chamilothori, Jan Wienold, and Marilyne Andersen. "Daylight patterns as a means to influence the spatial ambiance: a preliminary study". In: *Proceedings of the 3rd International Congress on Ambiances*. CONF. 2016.

[27] Angel X Chang et al. "SceneSeer: 3D scene design with natural language". In: *arXiv preprint arXiv:1703.00050* (2017).

[28] Angel Chang, Manolis Savva, and Christopher D Manning. "Interactive learning of spatial knowledge for text to 3D scene generation". In: *Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces*. 2014, pp. 14–21.

[29] Angel Chang, Manolis Savva, and Christopher D Manning. "Learning spatial knowledge for text to 3D scene generation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2014, pp. 2028–2038.

[30] Angel Chang et al. "Matterport3D: Learning from RGB-D data in indoor environments". In: *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017*. 2018, pp. 667–676. ISBN: 9781538626108. DOI: 10.1109/3DV.2017.00081. arXiv: 1709.06158.

[31] Yun-Chih Chang et al. "B*-trees: a new representation for non-slicing floorplans". In: *Proceedings 37th Design Automation Conference*. 2000, pp. 458–463. ISBN: VO -. DOI: 10.1109/DAC.2000.855354.

[32] Andre Chaszar, Peter Von Buelow, and Michela Turrin. "Multivariate Interactive Visualization of Data in Generative Design". In: *Symposium on Simulation for Architecture and Urban Design, SimAUD*. London, 2016.

[33] Guolong Chen et al. "VLSI floorplanning based on particle swarm optimization". In: *2008 3rd International Conference on Intelligent System and Knowledge Engineering*. Vol. 1. IEEE. 2008, pp. 1020–1025.

[34] Kang Chen et al. "Automatic semantic modeling of indoor scenes from low-quality RGB-D data using contextual information". In: *ACM Transactions on Graphics* 33.6 (2014).

[35] Tianshui Chen et al. "Knowledge-Embedded Routing Network for Scene Graph Generation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6163–6171.

[36] Kun-Hung Cheng and Chin-Chung Tsai. "Affordances of augmented reality in science learning: Suggestions for future research". In: *Journal of science education and technology* 22.4 (2013), pp. 449–462.

[37] Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. "Mutual human actuation". In: *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. 2017, pp. 797–805.

[38] Lung-Pan Cheng et al. "iturk: Turning passive haptics into active haptics by making users reconfigure props in virtual reality". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–10.

[39] Lung-Pan Cheng et al. “Sparse haptic proxy: Touch feedback in virtual environments using a general passive prop”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2017, pp. 3718–3728.

[40] Lung-Pan Cheng et al. “Turkdeck: Physical virtual reality based on people”. In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 2015, pp. 417–426.

[41] M Y Cheng. “Automated Site Layout of Temporary Construction Facilities Using-Enhanced Geographic Information Systems (GIS)”. In: *Ph. D. Disst., Depart. of Civil Engineering, University of Texas at Austin, Texas, USA* (1992).

[42] Anthony M Codd and Bipasha Choudhury. “Virtual reality anatomy: Is it comparable with traditional methods in the teaching of human forearm musculoskeletal anatomy?” In: *Anatomical sciences education* 4.3 (2011), pp. 119–125.

[43] Simone Colombo, Salman Nazir, Davide Manca, et al. “Immersive virtual reality for training and decision making: preliminary results of experiments performed with a plant simulator”. In: *SPE Economics & Management* 6.04 (2014), pp. 165–172.

[44] Simone Colombo et al. “Towards the automatic measurement of human performance in virtual environments for industrial safety”. In: *ASME 2011 World Conference on Innovative Virtual Reality*. American Society of Mechanical Engineers Digital Collection. 2011, pp. 67–76.

[45] Ben J Congdon, Tuanfeng Wang, and Anthony Steed. “Merging environments for shared spaces in mixed reality”. In: *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*. 2018, pp. 1–8.

[46] Michael Connolly et al. “Validation of a virtual reality-based robotic surgical skills curriculum”. In: *Surgical endoscopy* 28.5 (2014), pp. 1691–1694.

[47] Alan B Craig. *Understanding augmented reality: Concepts and applications*. Newnes, 2013.

[48] Angela Dai et al. “Scannet: Richly-annotated 3d reconstructions of indoor scenes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5828–5839.

[49] Bo Dai, Yuqi Zhang, and Dahua Lin. “Detecting visual relationships with deep relational networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3076–3086.

[50] David L Dawson. “Training in carotid artery stenting: do carotid simulation systems really help?” In: *Vascular* 14.5 (2006), pp. 256–263.

[51] Chris Dede. “Immersive interfaces for engagement and learning”. In: *science* 323.5910 (2009), pp. 66–69.

[52] Juan Manuel Davila Delgado et al. “A research agenda for augmented and virtual reality in architecture, engineering and construction”. In: *Advanced Engineering Informatics* 45 (2020), p. 101122.

[53] Tomás Méndez Echenagucia et al. "The early design stage of a building envelope: Multi-objective search through heating, cooling and lighting energy performance analysis". In: *Applied Energy* 154 (2015), pp. 577–591.

[54] Hans Eisenmann, Frank M Johannes, and Frank M Johannes. "Generic global placement and floorplanning". In: *Proceedings of the 35th annual Design Automation Conference*. ACM. 1998, pp. 269–274.

[55] Fazliaty Edora Fadzli et al. "A Review of Mixed Reality Telepresence". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 864. 1. IOP Publishing. 2020, p. 012081.

[56] Allen J Fairchild et al. "A mixed reality telepresence system for collaborative space operation". In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.4 (2016), pp. 814–827.

[57] T P Fernando, Kuo-Cheng Wu, and M N Bassanino. "Designing a novel virtual collaborative environment to support collaboration in design review meetings". In: *Journal of Information Technology in Construction* 18.August (2013), pp. 372–396. ISSN: 14036835.

[58] Matthew Fisher et al. "Activity-centric scene synthesis for functional 3D scene modeling". In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), pp. 1–13.

[59] Matthew Fisher et al. "Example-based synthesis of 3D object arrangements". In: *ACM Transactions on Graphics* 31.6 (2012), p. 1. ISSN: 07300301. DOI: 10.1145/2366145.2366154.

[60] Qiang Fu et al. "Adaptive synthesis of indoor scenes via activity-associated object relation graphs". In: *ACM Transactions on Graphics (TOG)* 36.6 (2017), pp. 1–13.

[61] Anthony G Gallagher and Christopher U Cates. "Approval of virtual reality training for carotid stenting: what this means for procedural-based medicine". In: *Jama* 292.24 (2004), pp. 3024–3026.

[62] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. "A Neural Algorithm of Artistic Style". In: (Aug. 2015). arXiv: 1508.06576. URL: http://arxiv.org/abs/1508.06576.

[63] Tobias Germer and Martin Schwarz. "Procedural Arrangement of Furniture for Real-Time Walkthroughs". In: *Computer Graphics Forum*. Vol. 28. 8. Wiley Online Library. 2009, pp. 2068–2078.

[64] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[65] Markus Gross et al. "blue-c: a spatially immersive display and 3D video portal for telepresence". In: *ACM Transactions on Graphics (TOG)*. Vol. 22. 3. ACM. 2003, pp. 819–827.

[66] Jiuxiang Gu et al. "Scene graph generation with external knowledge and image reconstruction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1969–1978.

[67] Enqiang Guo et al. “Learning to measure change: Fully convolutional siamese metric networks for scene change detection”. In: *arXiv preprint arXiv:1810.09111* (2018).

[68] Pei-Ning Guo, Chung-Kuan Cheng, and Takeshi Yoshimura. “An O-tree representation of non-slicing floorplan and its applications”. In: *Proceedings 1999 Design Automation Conference (Cat. No. 99CH36361)*. IEEE, 2003, pp. 268–273. ISBN: 1-58113-092-9. DOI: 10.1145/309847.309928. URL: http://ieeexplore.ieee.org/document/781324/.

[69] Ruiqi Guo, Chuhang Zou, and Derek Hoiem. “Predicting Complete 3D Models of Indoor Scenes”. In: (2015). URL: http://arxiv.org/abs/1504.02437.

[70] Bah-Hwee Gwee and Meng-Hiot Lim. “A GA with heuristic-based decoder for IC floorplanning”. In: *INTEGRATION, the VLSI journal* 28.2 (1999), pp. 157–172.

[71] Ridha Hambli, Abdessalam Chamekh, and Hédi Bel Hadj Salah. “Real-time deformation of structure using finite element and neural networks in virtual reality applications”. In: *Finite elements in analysis and design* 42.11 (2006), pp. 985–991.

[72] Michael Herdy. “Evolution strategies with subjective selection”. In: *Parallel Problem Solving from Nature — PPSN IV*. Ed. by Hans-Michael Voigt et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 22–31. ISBN: 978-3-540-70668-7.

[73] Arsalan Heydarian et al. “Immersive virtual environments versus physical built environments: A benchmarking study for building design and user-built environment explorations”. In: *Automation in Construction* 54 (2015), pp. 116–126.

[74] Geoffrey E Hinton and Richard S Zemel. “Autoencoders, minimum description length, and Helmholtz free energy”. In: *Advances in neural information processing systems* 6 (1994), pp. 3–10.

[75] JM Huang, Soh-Khim Ong, and Andrew YC Nee. “Visualization and interaction of finite element analysis in augmented reality”. In: *Computer-Aided Design* 84 (2017), pp. 1–14.

[76] Siyuan Huang et al. “Holistic 3D scene parsing and reconstruction from a single RGB image”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 187–203.

[77] Weixin Huang and Hao Zheng. “Architectural drawings recognition and generation through machine learning”. In: *Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, Mexico City, Mexico*. 2018, pp. 18–20.

[78] US HUD. “American Housing Survey for the United States: 2011”. In: *Census Bureau, US* (2011).

[79] Victoria Interrante, Brian Ries, and Lee Anderson. “Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments”. In: *2007 IEEE Symposium on 3D User interfaces*. IEEE. 2007.

[80] Phillip Isola et al. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[81] Katrine Jensen et al. “Simulation-based training for thoracoscopic lobectomy: a randomized controlled trial”. In: *Surgical endoscopy* 28.6 (2014), pp. 1821–1829.

[82] Zhaoyin Jia et al. “3d-based reasoning with blocks, support, and stability”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1–8.

[83] Jun H. Jo and John S. Gero. “Space layout planning using an evolutionary approach”. In: *Artificial Intelligence in Engineering* 12.3 (July 1998), pp. 149–162. ISSN: 0954-1810. DOI: 10.1016/S0954-1810(97)00037-X. URL: https://www.sciencedirect.com/science/article/pii/S095418109700037X.

[84] Justin Johnson et al. “Image retrieval using scene graphs”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3668–3678.

[85] Jost B Jonas, Stefan Rabethge, and Hans-Joachim Bender. “Computer-assisted training system for pars plana vitrectomy”. In: *Acta ophthalmologica Scandinavica* 81.6 (2003), pp. 600–604.

[86] Azzam S Al-Kadi et al. “The effect of simulation in improving students’ performance in laparoscopic surgery: a meta-analysis”. In: *Surgical endoscopy* 26.11 (2012), pp. 3215–3224.

[87] Andrew B Kahng et al. *VLSI physical design: from graph partitioning to timing closure*. Springer Science & Business Media, 2011.

[88] Vineet R Kamat and Julio C Martinez. “Visualizing simulated construction operations in 3D”. In: *Journal of computing in civil engineering* 15.4 (2001), pp. 329–337.

[89] Prabhjit Kaur. “An enhanced algorithm for floorplan design using hybrid ant colony and particle swarm optimization”. In: *Int. J. Res. Appl. Sci. Eng. Technol* 2 (2014), pp. 473–477.

[90] Z Sadeghipour Kermani et al. “Learning 3D Scene Synthesis from Annotated RGB-D Images”. In: *Computer Graphics Forum*. Vol. 35. 5. Wiley Online Library. 2016, pp. 197–206.

[91] Mohammad Keshavarzi, Ardavan Bidgoli, and Hans Kellner. “V-Dream: Immersive Exploration of Generative Design Solution Space”. In: *International Conference on Human-Computer Interaction*. Springer. 2020, pp. 477–494.

[92] Mohammad Keshavarzi and Mohammad Rahmani-Asl. “GenFloor: Interactive generative space layout system via encoded tree graphs”. In: *Frontiers of Architectural Research* 10.4 (2021), pp. 771–786.

[93] Mohammad Keshavarzi et al. “Affordance Analysis of Virtual and Augmented Reality Mediated Communication”. In: *arXiv preprint arXiv:1904.04723* (2019).

[94] Mohammad Keshavarzi et al. “Optimization and Manipulation of Contextual Mutual Spaces for Multi-User Virtual and Augmented Reality Interaction”. In: *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2020, pp. 353–362.

[95] Mohammad Keshavarzi et al. “SketchOpt: Sketch-based Parametric Model Retrieval for Generative Design”. In: *arXiv preprint arXiv:2009.00261* (2020).

[96] Young Min Kim et al. “Acquiring 3D indoor environments with variability and repetition”. In: *ACM Transactions on Graphics (TOG)* 31.6 (2012), p. 138.

[97] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[98] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. “Optimization by simulated annealing”. In: *science* 220.4598 (1983), pp. 671–680.

[99] Koji Kiyota and Kunihiro Fujiyoshi. “Simulated annealing search through general structure floor plans using sequence-pair”. In: *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)* 88.6 (2005), pp. 28–38.

[100] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. “Siamese neural networks for one-shot image recognition”. In: *ICML deep learning workshop*. Vol. 2. Lille. 2015.

[101] Eric Kolve et al. “Ai2-thor: An interactive 3d environment for visual ai”. In: *arXiv preprint arXiv:1712.05474* (2017).

[102] Yash Kotadia et al. “IndoorNet: Generating Indoor Layouts from a Single Panorama Image”. In: *Advanced Computing Technologies and Applications: Proceedings of 2nd International Conference on Advanced Computing Technologies and Applications—ICACTA 2020*. Springer. 2020, pp. 57–66.

[103] Ranjay Krishna et al. “Visual genome: Connecting language and vision using crowdsourced dense image annotations”. In: *International Journal of Computer Vision* 123.1 (2017), pp. 32–73.

[104] Saskia F Kuliga et al. “Virtual reality as an empirical research tool—Exploring user experience in a real building and a corresponding virtual model”. In: *Computers, environment and urban systems* 54 (2015), pp. 363–375.

[105] Claudia Kuster et al. “Towards next generation 3D teleconferencing systems”. In: *2012 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE. 2012, pp. 1–4.

[106] Vining Lang, Wei Liang, and Lap-Fai Yu. “Virtual agent positioning driven by scene semantics in mixed reality”. In: *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2019, pp. 767–775.

[107] Jason Lawrence et al. “Project Starline: A high-fidelity telepresence system”. In: *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)* 40(6) (2021).

[108] Chen-Yu Lee et al. “Roomnet: End-to-end room layout estimation”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 4865–4874.

[109] Nicolas H. Lehment, Daniel Merget, and Gerhard Rigoll. "Creating automatically aligned consensus realities for AR videoconferencing". In: *ISMAR 2014 - IEEE International Symposium on Mixed and Augmented Reality - Science and Technology 2014, Proceedings* September (2014), pp. 201–206.

[110] Philipp Leinen et al. "Virtual reality visual feedback for hand-controlled scanning probe microscopy manipulation of single molecules". In: *Beilstein journal of nanotechnology* 6.1 (2015), pp. 2148–2153.

[111] Manyi Li et al. "GRAINS: Generative recursive autoencoders for indoor scenes". In: *ACM Transactions on Graphics (TOG)* 38.2 (2019), p. 12.

[112] Yikang Li et al. "Scene graph generation from objects, phrases and region captions". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1261–1270.

[113] Yujia Li et al. "Gated Graph Sequence Neural Networks". In: (2017). arXiv: 1511.05493 [cs.LG].

[114] Zhengqin Li et al. "OpenRooms: An End-to-End Open Framework for Photorealistic Indoor Scene Datasets". In: *arXiv preprint arXiv:2007.12868* (2020).

[115] Yuan Liang, Song-Hai Zhang, and Ralph Robert Martin. "Automatic data-driven room design generation". In: *International Workshop on Next Generation Computer Animation Techniques*. Springer, 2017, pp. 133–148.

[116] Yuan Liang et al. "Knowledge graph construction with structure and parameter learning for indoor scene design". In: *Computational Visual Media* 4.2 (2018), pp. 123–137. ISSN: 20960662. DOI: 10.1007/s41095-018-0110-3.

[117] Chang-Tzu Lin, De-Sheng Chen, and Yi-Wen Wang. "An efficient genetic algorithm for slicing floorplan area optimization". In: *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No. 02CH37353)*. Vol. 2. IEEE. 2002, pp. II–II.

[118] Hui Lin et al. "Virtual geographic environments (VGEs): a new generation of geographic analysis tool". In: *Earth-Science Reviews* 126 (2013), pp. 74–84.

[119] Jai-Ming Lin and Yao-Wen Chang. "TCG: A transitive closure graph-based representation for general floorplans". In: *IEEE transactions on very large scale integration (VLSI) systems* 13.2 (2005), pp. 288–292.

[120] David Lindlbauer and Andy D Wilson. "Remixed reality: manipulating space and time in augmented reality". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–13.

[121] Chen Liu, Jiaye Wu, and Yasutaka Furukawa. "FloorNet: A Unified Framework for Floorplan Reconstruction from 3D Scans". In: (2018), pp. 1–18. arXiv: 1804.00090. URL: http://arxiv.org/abs/1804.00090.

[122] Chen Liu et al. “Planercnn: 3d plane detection and reconstruction from a single image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4450–4459.

[123] Daqi Liu, Miroslaw Bober, and Josef Kittler. “Visual Semantic Information Pursuit: A Survey”. In: *arXiv preprint arXiv:1903.05434* (2019).

[124] Alfredo Liverani, Falko Kuester, and Bernd Hamann. “Towards interactive finite element analysis of shell structures in virtual reality”. In: *1999 IEEE International Conference on Information Visualization (Cat. No. PR00210)*. IEEE. 1999, pp. 340–346.

[125] Stephen Lombardi et al. “Deep appearance models for face rendering”. In: *ACM Transactions on Graphics (ToG)* 37.4 (2018), pp. 1–13.

[126] GR Lorello et al. “Simulation-based training in anaesthesiology: a systematic review and meta-analysis”. In: *British journal of anaesthesia* 112.2 (2014), pp. 231–245.

[127] Cewu Lu et al. “Visual relationship detection with language priors”. In: *European Conference on Computer Vision*. Springer. 2016, pp. 852–869.

[128] Paul Luff and Christian Heath. “Mobility in Collaboration.” In: *CSCW*. Vol. 98. 1998, pp. 305–314.

[129] Rui Ma et al. “Action-driven 3D indoor scene evolution.” In: *ACM Trans. Graph.* 35.6 (2016), pp. 173–1.

[130] Rui Ma et al. “Language-driven synthesis of 3D scenes from scene databases”. In: *SIGGRAPH Asia 2018 Technical Papers*. ACM. 2018, p. 212.

[131] Shugao Ma et al. “Pixel codec avatars”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 64–73.

[132] Yuchun Ma et al. “VLSI floorplanning with boundary constraints based on corner block list”. In: *Proceedings of the 2001 Asia and South Pacific Design Automation Conference*. ACM. 2001, pp. 509–514.

[133] Morad Mahdjoub et al. “A collaborative design for usability approach supported by virtual reality and a multi-agent system embedded in a PLM environment”. In: *Computer-Aided Design* 42.5 (2010), pp. 402–413.

[134] Andrew Maimone and Henry Fuchs. “Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras”. In: *2011 10th IEEE International Symposium on Mixed and Augmented Reality*. IEEE. 2011, pp. 137–146.

[135] Andrew Maimone et al. “General-purpose telepresence with head-worn optical see-through displays and projector-based lighting”. In: *2013 IEEE Virtual Reality (VR)*. IEEE. 2013, pp. 23–26.

[136] Ali M Malkawi and Ravi S Srinivasan. “A new paradigm for Human-Building Interaction: the use of CFD and Augmented Reality”. In: *Automation in Construction* 14.1 (2005), pp. 71–84.

[137] Jiageng Mao, Xiaogang Wang, and Hongsheng Li. "Interpolated convolutional networks for 3d point cloud understanding". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 1578–1587.

[138] Ethan Marcotte. *Responsive web design: A book apart n 4*. Editions Eyrolles, 2017.

[139] Ricardo Martin-Brualla et al. "Nerf in the wild: Neural radiance fields for unconstrained photo collections". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7210–7219.

[140] Justin Matejka et al. "Dream lens: Exploration and visualization of large-scale generative design datasets". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM. 2018, p. 369.

[141] John McCormac et al. "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 4628–4635. ISBN: 978-1-5090-4633-1. DOI: 10.1109/ICRA.2017.7989538. URL: http://ieeexplore.ieee.org/document/7989538/.

[142] David Meignan et al. "A review and taxonomy of interactive optimization methods in operations research". In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5.3 (2015), p. 17.

[143] Paul Merrell et al. "Interactive furniture layout using interior design guidelines". In: *ACM transactions on graphics (TOG)* 30.4 (2011), pp. 1–10.

[144] John I Messner. "Evaluating the use of immersive display media for construction planning". In: *Workshop of the European Group for Intelligent Computing in Engineering*. Springer. 2006, pp. 484–491.

[145] Ben Mildenhall et al. "Nerf: Representing scenes as neural radiance fields for view synthesis". In: *European conference on computer vision*. Springer. 2020, pp. 405–421.

[146] D Jackuline Moni and S Arumugam. "VLSI Floorplanning based on Hybrid Particle Swarm Optimization". In: *Karunya Journal of Research* 1.1 (2009), pp. 111–121.

[147] Caitlin T Mueller and John A Ochsendorf. "Combining structural performance and designer preferences in evolutionary design space exploration". In: *Automation in Construction* 52 (2015), pp. 70–82.

[148] Pascal Müller et al. "Procedural modeling of buildings". In: *ACM SIGGRAPH 2006 Papers*. 2006, pp. 614–623.

[149] Shigetoshi Nakatake et al. "Module placement on BSG-structure and IC layout applications". In: *Proceedings of the 1996 IEEE/ACM international conference on Computer-aided design*. IEEE Computer Society. 1997, pp. 484–491.

[150] Shingo Nakaya, Tetsushi Koide, and Si Wakabayashi. "An adaptive genetic algorithm for VLSI floorplanning based on sequence-pair". In: *2000 IEEE International Symposium on Circuits and Systems. Emerging Technologies for the 21st Century. Proceedings (IEEE Cat No. 00CH36353)*. Vol. 3. IEEE. 2000, pp. 65–68.

[151] Sahil Narang, Andrew Best, and Dinesh Manocha. "Simulating movement interactions between avatars & agents in virtual worlds using human motion constraints". In: *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2018, pp. 9–16.

[152] Daren T Nicholson et al. "Can virtual reality improve anatomy education? A randomised controlled study of a computer-generated three-dimensional anatomical ear model". In: *Medical education* 40.11 (2006), pp. 1081–1087.

[153] Niels Christian Nilsson et al. "15 Years of Research on Redirected Walking in Immersive Virtual Environments". In: *IEEE Computer Graphics and Applications* 38.2 (2018), pp. 44–56. ISSN: 02721716. DOI: 10.1109/MCG.2018.111125628.

[154] Magnus Norrby et al. "Molecular rift: virtual reality for drug designers". In: *Journal of chemical information and modeling* 55.11 (2015), pp. 2475–2484.

[155] Christoph Nytsch-Geusen et al. "BuildingSystems_VR–A New Approach or Immersive and Interactive Building Energy Simulation". In: *Building Simulation 2017, Proceedings of the Fifteenth IBPSA Conference*. 2017, pp. 628–634.

[156] Yuichi Ohta and Hideyuki Tamura. *Mixed reality: merging real and virtual worlds*. Springer Publishing Company, Incorporated, 2014.

[157] Sergio Orts-Escolano et al. "Holoportation". In: 2017, pp. 741–754. DOI: 10.1145/2984511.2984517.

[158] Hesham M. Osman, Maged E. Georgy, and Moheeb E. Ibrahim. "A hybrid CAD-based construction site layout planning system using genetic algorithms". In: *Automation in Construction* 12.6 (2003), pp. 749–764. ISSN: 09265805. DOI: 10.1016/S0926-5805(03)00058-X.

[159] Rivka Oxman. "Performance-based design: current practices and research issues". In: *International journal of architectural computing* 6.1 (2008), pp. 1–17.

[160] Daniel Paes, Eduardo Arantes, and Javier Irizarry. "Immersive environment for improving the understanding of architectural 3D models: Comparing user spatial perception between immersive and traditional virtual reality systems". In: *Automation in Construction* 84.August 2016 (2017), pp. 292–303. ISSN: 09265805. DOI: 10.1016/j.autcon.2017.09.016. URL: http://dx.doi.org/10.1016/j.autcon.2017.09.016.

[161] Tabitha C Peck, Henry Fuchs, and Mary C Whitton. "An evaluation of navigational ability comparing Redirected Free Exploration with Distractors to Walking-in-Place and joystick locomotio interfaces". In: *2011 IEEE Virtual Reality Conference*. IEEE. 2011, pp. 55–62.

[162] Francesco Pittaluga et al. "Revealing scenes by inverting structure from motion reconstructions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 145–154.

[163] Pulak Purkait, Christopher Zach, and Ian Reid. "SG-VAE: Scene Grammar Variational Autoencoder to generate new indoor scenes". In: *European Conference on Computer Vision*. Springer. 2020, pp. 155–171.

[164] Charles R Qi et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 652–660.

[165] Charles R. Qi et al. "Volumetric and Multi-View CNNs for Object Classification on 3D Data". In: (2016). ISSN: 10636919. DOI: 10.1109/CVPR.2016.609. arXiv: 1604.03265. URL: http://arxiv.org/abs/1604.03265.

[166] Charles R. Qi et al. "Volumetric and Multi-View CNNs for Object Classification on 3D Data". In: (2016). ISSN: 10636919. DOI: 10.1109/CVPR.2016.609. URL: http://arxiv.org/abs/1604.03265.

[167] Siyuan Qi et al. "Human-centric indoor scene synthesis using stochastic grammar". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5899–5908.

[168] Anett Racz and Gergo Zilizi. "VR aided architecture and interior design". In: *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*. IEEE. 2018, pp. 11–16.

[169] Charles George Ramsey. *Architectural graphic standards*. John Wiley & Sons, 2007.

[170] Sharif Razzaque, Zachariah Kohn, and Mary C Whitton. "Redirected Walking". In: *Proceedings of EUROGRAPHICS* (2001), pp. 289–294. ISSN: 00044172.

[171] Maurizio Rebaudengo and Matteo Sonza Reorda. "GALLO: A genetic algorithm for floorplan area optimization". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 15.8 (1996), pp. 943–951.

[172] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[173] Alexander Richard et al. "Audio-and gaze-driven facial animation of codec avatars". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, pp. 41–50.

[174] Daniel Ritchie, Kai Wang, and Yu-an Lin. "Fast and flexible indoor scene synthesis via deep convolutional generative models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 6182–6190.

[175] Mike Roberts and Nathan Paczan. "Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding". In: *arXiv preprint arXiv:2011.02523* (2020).

[176] George Robertson, Mary Czerwinski, and Maarten Van Dantzich. "Immersion in desktop virtual reality". In: *Proceedings of the 10th annual ACM symposium on User interface software and technology*. 1997, pp. 11–19.

[177] Roy A Ruddle, Stephen J Payne, and Dylan M Jones. "Navigating large-scale virtual environments: what differences occur between helmet-mounted and desk-top displays?" In: *Presence: Teleoperators & Virtual Environments* 8.2 (1999), pp. 157–168.

[178] Roy A Ruddle and Patrick Péruch. "Effects of proprioceptive feedback and environmental characteristics on spatial learning in virtual environments". In: *International Journal of Human-Computer Studies* 60.3 (2004), pp. 299–326.

[179] Adam Rysanek, Clayton Miller, and Arno Schlueter. "A workflow for managing building information and performance data using virtual reality: an alternative to BIM for existing buildings?" In: August (2017).

[180] Mayu Sakurada and Takehisa Yairi. "Anomaly detection using autoencoders with nonlinear dimensionality reduction". In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. 2014, pp. 4–11.

[181] M Saldana and C Johanson. "Procedural modeling for rapid-prototyping of multiple building phases". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 5 (2013), W1.

[182] Beatriz Sousa Santos et al. "Head-mounted display versus desktop for 3D navigation in virtual reality: a user study". In: *Multimedia tools and applications* 41.1 (2009), p. 161.

[183] Vedant Saran, James Lin, and Avideh Zakhor. "Augmented Annotations: Indoor Dataset Generation with Augmented Reality". In: *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences* (2019).

[184] Manolis Savva, Angel X Chang, and Maneesh Agrawala. "Scenesuggest: Context-driven 3D scene design". In: *arXiv preprint arXiv:1703.00061* (2017).

[185] F. Scarselli et al. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80. DOI: 10.1109/TNN.2008.2005605.

[186] Marc Schnabel and Thomas Kvan. "Spatial Understanding in Immersive Virtual Environments". In: *International Journal of Architectural Computing* 1.4 (2003), pp. 435–448. ISSN: 1478-0771. DOI: 10.1260/147807703773633455. URL: http://dx.doi.org/10.1260/147807703773633455%5C%5Cnhttp://multi-science.metapress.com/content/wv67120120230387/?genre=article%5C&id=doi:10.1260/147807703773633455%5C%5Cnhttp://www.metapress.com/content/wv67120120230387/fulltext.pdf.

[187] Thomas Schops et al. "A multi-view stereo benchmark with high-resolution images and multi-camera videos". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3260–3269.

[188] Stacey D Scott, Neal Lesh, and Gunnar W Klau. "Investigating human-computer optimization". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM. 2002, pp. 155–162.

[189] Skipper Seabold and Josef Perktold. "statsmodels: Econometric and statistical modeling with python". In: *9th Python in Science Conference*. 2010.

[190] Stéfanie A Seixas-Mikelus et al. "Can image-based virtual reality help teach anatomy?" In: *Journal of endourology* 24.4 (2010), pp. 629–634.

[191] Neal E Seymour et al. "Virtual reality training improves operating room performance: results of a randomized, double-blinded study". In: *Annals of surgery* 236.4 (2002), p. 458.

[192] Tianjia Shao et al. "An interactive approach to semantic modeling of indoor scenes with an rgbd camera". In: *ACM Transactions on Graphics (TOG)* 31.6 (2012), p. 136.

[193] V. Sharp. *The Art of Redesign*. Sharp Publishing, 2008. ISBN: 9780980883503. URL: https://books.google.com/books?id=2kxqIfFC1EcC.

[194] Kristina Shea, Robert Aish, and Marina Gourtovaia. "Towards integrated performance-driven generative design tools". In: *Automation in Construction* 14.2 (2005), pp. 253–264.

[195] Simon J. Sheather. "Density Estimation". In: *Statistical Science* 19.4 (2004), pp. 588–597. ISSN: 08834237. URL: http://www.jstor.org/stable/4144429.

[196] William R Sherman and Alan B Craig. *Understanding virtual reality: Interface, application, and design*. Elsevier, 2002.

[197] Yifei Shi et al. "Hierarchy Denoising Recursive Autoencoders for 3D Scene Layout Prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1771–1780.

[198] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of Big Data* 6.1 (2019), pp. 1–48.

[199] Nathan Silberman et al. "Indoor segmentation and support inference from rgbd images". In: *European conference on computer vision*. Springer. 2012, pp. 746–760.

[200] Rajendra Bahadur Singh, Anurag Singh Baghel, and Ayush Agarwal. "A review on VLSI floorplanning optimization using metaheuristic algorithms". In: *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*. Institute of Electrical and Electronics Engineers Inc., Nov. 2016, pp. 4198–4202. ISBN: 9781467399395. DOI: 10.1109/ICEEOT.2016.7755508.

[201] Siddharth Singh, Robert E Sedlack, and David A Cook. "Effects of simulation-based training in gastrointestinal endoscopy: a systematic review and meta-analysis". In: *Clinical Gastroenterology and Hepatology* 12.10 (2014), pp. 1611–1623.

[202] Mel Slater and Martin Usoh. "Representations systems, perceptual position, and presence in immersive virtual environments". In: *Presence: Teleoperators & Virtual Environments* 2.3 (1993), pp. 221–233.

[203] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. "Sun rgb-d: A rgb-d scene understanding benchmark suite". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 567–576.

[204] Shuran Song et al. "Semantic scene completion from a single depth image". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1746–1754.

[205] B Sowmya and MP Sunil. “Minimization of floorplanning area and wire length interconnection using particle swarm optimization”. In: *International Journal of Emerging Technology and Advanced Engineering* 3.8 (2013).

[206] Misha Sra, Aske Mottelson, and Pattie Maes. “Your place and mine: Designing a shared VR experience for remotely located users”. In: *Proceedings of the 2018 Designing Interactive Systems Conference*. 2018, pp. 85–97.

[207] Misha Sra et al. “Procedurally generated virtual reality from 3D reconstructed physical space”. In: *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*. 2016, pp. 191–200.

[208] Patrick Stotko et al. “SLAMCast: Large-scale, real-time 3D reconstruction and streaming for immersive multi-client live telepresence”. In: *IEEE transactions on visualization and computer graphics* 25.5 (2019), pp. 2102–2112.

[209] Shih-Yang Su et al. “A-nerf: Surface-free human 3d pose refinement via neural rendering”. In: *arXiv preprint arXiv:2102.06199* (2021).

[210] Evan A. Suma et al. “Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture”. In: *IEEE Transactions on Visualization and Computer Graphics* 18.4 (2012), pp. 555–564. ISSN: 10772626. DOI: 10.1109/TVCG.2012.47.

[211] Evan A. Suma et al. “Leveraging change blindness for redirection in virtual environments”. In: *Proceedings - IEEE Virtual Reality*. 2011, pp. 159–166. ISBN: 9781457700361. DOI: 10.1109/VR.2011.5759455.

[212] Cheng Sun et al. “Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1047–1056.

[213] Tsung-Ying Sun et al. “Floorplanning based on particle swarm optimization”. In: *IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI’06)*. IEEE. 2006, 5–pp.

[214] Carole Talbott and Maggie Matthews. *Decorating for Good: A Step-by-step Guide to Rearranging What You Already Own*. Clarkson Potter, 1999.

[215] Damien Teney, Lingqiao Liu, and Anton van den Hengel. “Graph-structured representations for visual question answering”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1–9.

[216] Ayush Tewari et al. “Advances in neural rendering”. In: *arXiv preprint arXiv:2111.05849* (2021).

[217] I. D. Tommelein et al. “SightPlan experiments: alternate strategies for site layout design”. In: *Computing in Civil Engineering* 5.1 (1991), pp. 42–63. ISSN: 00320935. DOI: 10.1007/BF01927759.

[218] Michela Turrin, Peter Von Buelow, and Rudi Stouffs. "Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms". In: *Advanced Engineering Informatics* 25.4 (2011), pp. 656–675.

[219] Martin Usoh et al. "Walking> walking-in-place> flying, in virtual environments". In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 1999, pp. 359–364.

[220] Khrystyna Vasylevska et al. "Flexible spaces: Dynamic layout generation for infinite walking in virtual environments". In: *IEEE Symposium on 3D User Interface 2013, 3DUI 2013 - Proceedings*. 2013, pp. 39–42. ISBN: 9781467360975. DOI: 10.1109/3DUI.2013.6550194.

[221] Lizeth Vega-Medina, Gerardo Tibamoso, and Byron Perez-Gutierrez. "VR tool for interaction with the abdomen anatomy". In: *International Conference on Human-Computer Interaction*. Springer. 2013, pp. 235–239.

[222] Petar Veličković et al. "Deep graph infomax". In: *arXiv preprint arXiv:1809.10341* (2018).

[223] Petar Veličković et al. "Graph attention networks". In: *arXiv preprint arXiv:1710.10903* (2017).

[224] Peter Von Buelow. "ParaGen: Performative Exploration of generative systems". In: *Journal of the International Association for Shell and Spatial Structures* 53.4 (2012), pp. 271–284.

[225] Johanna Wald et al. "Learning 3D Semantic Scene Graphs from 3D Indoor Reconstructions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 3961–3970.

[226] David Waller, Earl Hunt, and David Knapp. "The transfer of spatial knowledge in virtual environment training". In: *Presence* 7.2 (1998), pp. 129–143.

[227] Kai Wang et al. "Deep convolutional priors for indoor scene synthesis". In: *ACM Transactions on Graphics (TOG)* 37.4 (2018), p. 70.

[228] Kai Wang et al. "Planit: Planning and instantiating indoor scenes with relation graph and spatial prior networks". In: *ACM Transactions on Graphics (TOG)* 38.4 (2019), p. 132.

[229] Laung-Terng Wang, Yao-Wen Chang, and Kwang-Ting (Tim) Cheng, eds. *Electronic Design Automation: Synthesis, Verification, and Test*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009. ISBN: 9780080922003.

[230] Ting-Chun Wang et al. "High-resolution image synthesis and semantic manipulation with conditional gans". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8798–8807.

[231] Lauri Ward. *Use what You Have Decorating: Transform Your Home in One Hour with Ten Simple Design Principles Using...* Penguin, 1999.

[232] Wei-Chao Wen et al. "Toward a Compelling Sensation of Telepresence: Demonstrating a portal to a distant (static) office". In: *Proceedings Visualization 2000. VIS 2000 (Cat. No. 00CH37145)*. IEEE. 2000, pp. 327–333.

[233] VE Whisker et al. “Using immersive virtual environments to develop and visualize construction schedules for advanced nuclear power plants”. In: *Proceedings of ICAPP*. Vol. 3. 2003, pp. 4–7.

[234] Betsy Williams et al. “Exploring large virtual environments with an HMD when physical space is limited”. In: *Proceedings of the 4th symposium on Applied perception in graphics and visualization*. 2007, pp. 41–48.

[235] D F Wong and C L Liu. “A New Algorithm for Floorplan Design”. In: *Proceedings of the 23rd ACM/IEEE Design Automation Conference*. DAC ’86. Piscataway, NJ, USA: IEEE Press, 1986, pp. 101–107. ISBN: 0-8186-0702-5. URL: http://dl.acm.org/citation.cfm?id=318013.318030.

[236] Hsin-Kai Wu et al. “Current status, opportunities and challenges of augmented reality in education”. In: *Computers & education* 62 (2013), pp. 41–49.

[237] Wang Xiaogang et al. “VLSI Floorplanning Method Based on Genetic Algorithms [J]”. In: *Microprocessors* 1 (2002), p. 1.

[238] Bin Xu and Zhenzhong Chen. “Multi-level fusion based 3d object detection from monocular images”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2345–2353.

[239] Kai Xu et al. “3d attention-driven depth acquisition for object identification”. In: *ACM Transactions on Graphics (TOG)* 35.6 (2016), p. 238.

[240] Ken Xu, James Stewart, and Eugene Fiume. “Constraint-based automatic placement for scene composition”. In: *Graphics Interface*. Vol. 2. 2002, pp. 25–34.

[241] Michael Ying Yang et al. “On support relations and semantic scene graphs”. In: *ISPRS journal of photogrammetry and remote sensing* 131 (2017), pp. 15–25.

[242] Ting Yao et al. “Exploring visual relationship for image captioning”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 684–699.

[243] Yi-Ting Yeh et al. “Synthesizing open worlds with constraints using locally annealed reversible jump mcmc”. In: *ACM Transactions on Graphics (TOG)* 31.4 (2012), pp. 1–11.

[244] Jiaxuan You et al. “GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models”. In: (2018). arXiv: 1802.08773 [cs.LG].

[245] Lap-Fai Yu et al. “Make it Home: Automatic Optimization of Furniture Arrangement Lap-Fai”. In: *ACM Transactions on Graphics* 30.4 (July 2011), p. 1. ISSN: 07300301. DOI: 10.1145/2010324.1964981. URL: http://portal.acm.org/citation.cfm?doid=2010324.1964981.

[246] Zehao Yu et al. “Single-image piece-wise planar 3d reconstruction via associative embedding”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 1029–1037.

[247] Rowan Zellers et al. “Neural motifs: Scene graph parsing with global context”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5831–5840.

[248] Benjamin Zendejas et al. “State of the evidence on simulation-based training for laparoscopic surgery: a systematic review”. In: *Annals of surgery* 257.4 (2013), pp. 586–593.

[249] Cha Zhang et al. “Viewport: A distributed, immersive teleconferencing system with infrared dot pattern”. In: *IEEE MultiMedia* 20.1 (2013), pp. 17–27.

[250] Muhan Zhang and Yixin Chen. “Link prediction based on graph neural networks”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 5165–5175.

[251] Song-Hai Zhang et al. “A Survey of 3D Indoor Scene Synthesis”. In: *Journal of Computer Science and Technology* 34.3 (2019), pp. 594–608.

[252] Song-Hai Zhang et al. “Fast 3D Indoor Scene Synthesis with Discrete and Exact Layout Pattern Extraction”. In: *arXiv preprint arXiv:2002.00328* (2020).

[253] Yibiao Zhao and Song-Chun Zhu. “Image parsing with stochastic scene grammar”. In: *Advances in Neural Information Processing Systems*. 2011, pp. 73–81.

[254] Yibiao Zhao and Song-Chun Zhu. “Scene parsing by integrating function, geometry and appearance models”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 3119–3126.

[255] Bo Zheng et al. “Scene understanding by reasoning stability and safety”. In: *International Journal of Computer Vision* 112.2 (2015), pp. 221–238.

[256] Chong Zhou and Randy C Paffenroth. “Anomaly detection with robust deep autoencoders”. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 665–674.

[257] Yang Zhou, Zachary While, and Evangelos Kalogerakis. “SceneGraphNet: Neural Message Passing for 3D Indoor Scene Augmentation”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7384–7392.

[258] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. “SPEA2: Improving the strength Pareto evolutionary algorithm”. In: *TIK-report* 103 (2001).

[259] Bo Zong et al. “Deep autoencoding gaussian mixture model for unsupervised anomaly detection”. In: *International Conference on Learning Representations*. 2018.

[260] Chuhang Zou et al. “Layoutnet: Reconstructing the 3d room layout from a single rgb image”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 2051–2059.