Graduate School of Energy Science, Kyoto University

Master Thesis in Socio-Environmental Energy Science

VR Earthquake Experience System with Automatic Reconstruction of Indoor Environment

Supervisor: Prof. Hiroshi Shimoda

Author: Daiki Handa

Submission: February 6, 2015

Abstract

Title:
VR Earthquake Experience System with Automatic Reconstruction of Indoor Environment
(室内環境の自動再構築手法を用いた VR 地震体験システム)

Shimoda Laboratory, Daiki Handa

Abstract:

As 2011 Tohoku Earthquake represents, earthquakes could cause enormous human casualties, and it is crucial to reduce their risks through disaster education. One important approach is to increase preparedness towards earthquake by causing fear toward it, under the principle of fear-arousing communications. Various immersive earthquake simulation methods has been developed to serve this purpose. Of these methods, a method based on virtual reality (VR) is attractive, because it only requires a compact hardware unlike simulation vehicles, and it has a potential to be widely adopted. But the contents need to be created manually by artists, and that limits its practicability.

In this research, a new earthquake experience system is proposed in which indoor environments are 3D-scanned, and a virtual environment is automatically created by a novel reconstruction method. Simulation of earthquakes is presented via VR. It can be expected the proposed system removes the cost problem of the previous VR method, while still retaining effectiveness in fear-causing.

To examine whether the system is actually capable of causing fear toward earthquake, an experimental evaluation was conducted. In the experiment, participants experienced earthquakes simulated by the system, and perceived fear, realness, and seismic intensity scales were measured by questionnaires, among other properties. Interviews were also conducted to investigate factors affecting these perceptual properties. The evaluation experiment showed that an earthquake simulation of seismic intensity scale 6.6 could actually cause fear in more than half participants. However, perceived seismic intensity scales were consistently lower than simulated intensity scales. The interview results revealed that the discrepancy could be attributed to omission of vestibular stimulus and lack of details in reconstructed scenes. Of several elements contributing to realness, ground-shaking sounds and overall scene photorealism were scored moderately high. However, collision sounds of objects were felt as unrealistic, because of lacking detail in object recognition.

Although the automatic approach of the proposed system was shown to be viable, increasing reliability and granularity of object recognition would be a hard problem, and would need significant progress in the field of computer vision.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The 2011 earthquake off the Pacific coast of Tohoku had caused enourmous human casualties. Since earthquakes occur frequently, natural disaster education plays an important role in reducing their risk.

A critical aspect of the disaster education is to persuade people to actually prepare for earthquakes through fear-arousing communication[1]. There are three methods to cause fear towards earthquakes: video footages of an earthquake, simulation vehicles, and virtual reality (VR) earthquake simulation[2].

Video footages are widely used since they can provide variety of both indoor and outdoor situations relatively easily. But they are arguably the least effective in causing fear, because watching a video on a screen is not immersive. Simulation vehicles are also commonly used, and they can simulate an earthquake in a small room. A simulation vehicle is a small truck containing a shake table and a small room attached to it. People can experience earthquakes by getting inside it, surrounded by the room and furnitures. Although the experience is very immersive, its realness and flexibility are limited. First, the furnitures must be secured in place to prevent injuries by toppling over people, and simulation of a large room is difficult because of vehicle size. Second, changing the room to different situations is also hard, because physical installation must be changed whenever situation is modified. The VR earthquake simulation[2] is a proposal to display an earthquake in a virtual indoor scene, simulated by computers. In theory, it can simulate any kind of scene and present it immersively by using VR hardware that occupies users' field of view entirely. However, it is hard to simulate many kinds of scenes because of the cost and time required to model furnitures and buildings professionally.

Thus, there are no existing method that can simulate earthquakes in a variety of indoor scenes with high immersiveness, and at lower cost. Of these three methods, the VR earthquake simulation method has arguably the largest potential, because its limiting factor is cost of content generation, unlike other methods which are limited physically.

Meanwhile, fields of computer graphics, computer vision, and virtual reality are rapidly progressing, backed by constantly rising computational power, represented by the Moore's law. Two interesting developments are, gradual availability of high quality VR hardware at consumer prices, and progress of 3D reconstruction techniques in general.

In this research, a new VR earthquake experience system is proposed, which is based on automatic generation of 3D contents from 3D scan data. In the previous method, individual objects in an indoor scene are manually modeled by professional artists, and then motion of the objects are calculated via physics simulation by computers. In the proposed system, such a manual modeling process is replaced by a novel automatic reconstruction method, eliminating the cost problem. The reconstruction method takes 3D scan data of target indoor scene as an input, recognizes objects contained in the data, and then reconstruct appearance and physical models of the objects. Existing methods[3][4] are typically capable of only one of recognition and reconstruction, and are incapable of generating 3D contents for physics simulation. Due to the low cost and automatic nature of the proposed system, it would be able to scan and generate the contents where earthquake education is being conducted, such as schools or public meetings. This is a unique property of the system unlike previous methods, where target scene must be prepared beforehand.

The proposed system was evaluated by an experiment, in which participants experienced the earthquake contents generated by the system, and feedbacks were collected via questionnaires and interviews.

The thesis consists of 5 chapters including Introduction. In chapter 2, the background of research is discussed and existing 3D reconstruction techniques are reviewed. In chapter 3, the proposed system is described as a composition of three subsystems and various methods that consist them, including the new indoor reconstruction method. In chapter 4, an experiment is devised from the fear-arousing communication principle and the system requirements, and the experiment results are analyzed. Finally, chapter 5 summarizes the research and discusses future works to further enhance effectiveness of the system.

Chapter 2

Background and Objective of This Research

In this chapter, we review existing methods of information presentation used in earthquake education, from Disaster Risk Reduction (DRR) and fear-arousing communication perspectives. Then we propose a new VR earthquake experience system based on automatic content generation and realtime computer simulation, while mentioning existing literature on 3D scene understanding.

2.1 Existing Earthquake Education Methods

In literature of DRR, Wisner et al. proposed CARDIAC, which is a set of seven concrete objectives for risk reduction[5]. One of CARDIAC objectives is Communication, which is described as to "Understand and communicate the nature of hazards, vulnerabilities and capacities."[5]. Education of disaster can be considered to be one approach to the communication objective. But the literature also states that effective DRR requires not only informing people about disaster, but also changing actual behavior of people.

One such method of changing behavior is fear-arousing communication[1]. Fear-arousing communication is comprised of 1. fear toward the risk, and 2. knowledge of actions to reduce the risk, and it is known that such a combination increases attitude and action toward the risk. Fig. 2.1 shows this relationship. In case of earthquake, there are several methods of information presentation, which are summarized in Table 2.1 with qualitative evaluation of amounts of informational content and fear caused by the methods.

Text-based method presents information via text and images usually on printed paper, and can convey large amounts of complex information such as mechanism and history of earthquakes. However, it is difficult to arouse fear via use of text-based methods. Video-

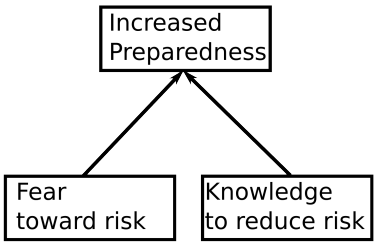

Figure 2.1: Two factors in fear-arousing communications.

Table 2.1: Various methods of earthquake information presentation.

Type of presentation	Information	Fear
Text-based[6]	high	low
Video-based[7]	mid-high	low-mid
Simulation vehicle[8]	low	high
VR[2]	low	high

based method is typically a mixture of recorded disaster footage and descriptive part much like text-based presentation. Thus, video-based method can arouse medium amount of fear with low information content, or low fear with high information content.

On the other hand, simulation vehicles or VR earthquake simulation are mainly designed to cause fear by their immersiveness, rather than to present information. Also, simulation vehicles are limited to indoor scene. Arguably, it is harder to deploy these methods compared to text-based or video-based methods, because of higher cost required to prepare simulation vehicles or professionally created 3D contents for VR.

These two groups of methods correspond to the two complementary factors in fear-arousing communication (Fig. 2.1), and they are also used in a mixed way in practical earthquake education settings. However, the immersive methods are limited in variation of indoor scenes they can present, primarily because of their high cost. Section 2.2 discusses these methods more closely.

2.2 Immersive Earthquake Experiences

A typical earthquake simulation vehicle is equipped with a single small room on a shake table, in which several people can experience earthquake simulataneously[8]. It is common for such a vehicle to be able to simulate earthquakes up to scale 7 (using Japan Meteorological Agency seismic intensity scale[9]). Seismic intensity scale 7 can be thought of as the maximum seismic intensity scale practically required for an indoor earthquake experience system, since it is estimated that most buildings will collapse in an earthquake

stronger than scale 7. It is also technically possible to increase the size of simulation up to a whole building, using a huge shake table such as E-Defense[10].

Although simulation using mechanical shake tables can generate completely accurate experience in theory, they are practically limited by safety and cost constraints. For example, furnitures need to be secured in place to avoid physical harms to the subjects, and simulation of multiple or larger environments is hard due to the cost associated with the physical setup. These constraints hinder realness, because objects such as furnitures can move above seismic intensity scale 4, and this method cannot simulate them. Below scale 4, the value of earthquake simulation is low, because weak earthquakes are pretty harmless to human and also occur relatively frequently.

Another type of immersive earthquake simulation[2] can be created with VR which relies on electronic hardware such as head mounted display (HMD) and headphones to stimulate human senses, and computer to simulate physical process. In this case, environment can be anything within the limit of computation, so the problems of environment size and non-moving furnitures can be resolved.

However, there are two technical problems with the existing VR method. First, sensory modes of stimulus are typically limited to visual and auditory because other modes such as touch and olfaction, are much less understood and hard to synthesize. An excluded sensory mode of particular importance is vestibular sense, which is used to sense acceleration and rotation of head. Vestibular system can be stimulated by using motion platforms, which are essentially same as shake tables. Mixed setup of motion platforms and audio-visual VR hardware is commonly found in aircraft pilot training[11], but there is no known work for earthquake experience utilizing such a setup. Second, 3D content creation by artists can be costly, possibly more than purchasing real furnitures. In a tutorial website for 3D artists[12], it is stated that an artist should charge 500 USD for an empty room, and 50 USD for a single simple object such as a wooden box. Considering a typical room contains hundreds of unique objects (e.g. furnitures, tools, books, clothes) with shapes more complex than a box, it can be argued that modeling a room costs more than 5500 USD.

Both the limitations of simulation vehicles and the lack of vestibular or tactile senses in VR stem from physical constraints, and arguably these disadvantages are hard to overcome. However, cost of 3D content creation in the VR method could be lowered drastically by automatic content generation.

2.3 Objective of This Research

In this research, we propose a system that can generate content automatically and can simulate earthquake in VR using the generated content. Since the system is automatic, it can eliminate the cost problem in existing VR earthquake simulation methods, and also enables modeling of various locations, including a site where earthquake education is conducted. These properties of the system would help spreading VR earthquake simulations, which are conceivably effective in causing fear because of its high immersiveness.

Ability of the system to cause fear and practicability of the system are also evaluated experimentally in this research.

2.3.1 Proposed Earthquake Experience System

The proposed system can generate content automatically and use the content for VR earthquake experience. The automatic generation is enabled by a novel indoor 3D reconstruction method which can generate separated objects with both appearance model and collision model for rigid-body physics simulation, unlike existing 3D reconstruction methods reviewed in section 2.4.

The proposed system is expected to be usable by a non-professional person to scan a real indoor scene, and generate content for earthquake experience of the scene automatically, given measured earthquake data to simulate. Combined with availability of high-quality HMD at consumer prices such as Oculus Rift DK2[13], the system can be expected to be used to scan a real world location, and to present the experiences of the location. The location could be where earthquake education is conducted, or residence of the educatee to enhance the fear.

Fig. 2.2 shows the difference in content creation workflows between the conventional method and the proposed system. Vast reduction of time consuming and costly professional labor is apparent in the proposed system. Due to the difficulty of perfectly recognizing objects and phenomena in the scene, the proposed method will not be able to generate complex phenomena such as building deformation, wall cracking, or object shattering. However, the potential of the proposed method to present on-the-spot experience can be considered to overweigh these disadvantages in realness.

2.4 Existing Literatures on 3D Scene Reconstruction

To reconstruct a real world scene, the scene must be captured by cameras or 3D scanners, and then various properties of things in the scene must be estimated by computers. These properties include 3D object locations, materials, and light locations. Extraction

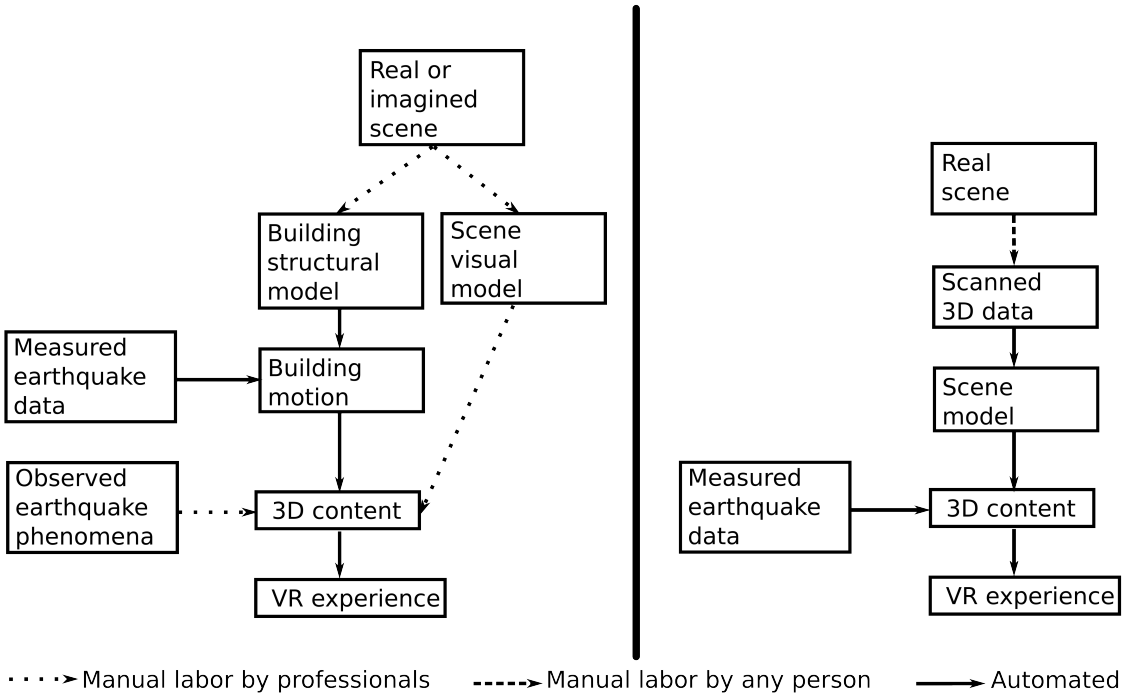

Figure 2.2: Workflows to generate VR contents in a conventional (left) and the proposed (right) method.

of information like this is called *scene understanding*. Then, these properties must be converted to forms suitable for realtime rendering and simulation. Typically, objects are represented by triangle meshes and textures. Finally, the objects and the lights are placed in a virtual scene, and the physical phenomena such as earthquakes become simulatable. Fig. 2.3 shows this conceptual dataflow of scene reconstruction.

An ideal computer vision program would be able to very deeply recognize all objects in an arbitrary image, which means it could estimate all of the following:

- which object a given pixel corresponds to (segmentation)
- linguistic properties (e.g. name) of the objects
- kinetic properties (e.g. mass, rigidity, articulation) of the objects
- visual properties (e.g. color, 3D shape, surface material) of the objects

Fig. 2.4 shows these four concepts visually. However, complete recognition of objects in an arbitrary scene is believed to be the hardest problem in computer vision[14], and existing state-of-the-art methods typically focus on only one or two of these aspects and still yield incomplete results. Existing methods are grouped by their primary aspects, and are reviewed briefly in the following.

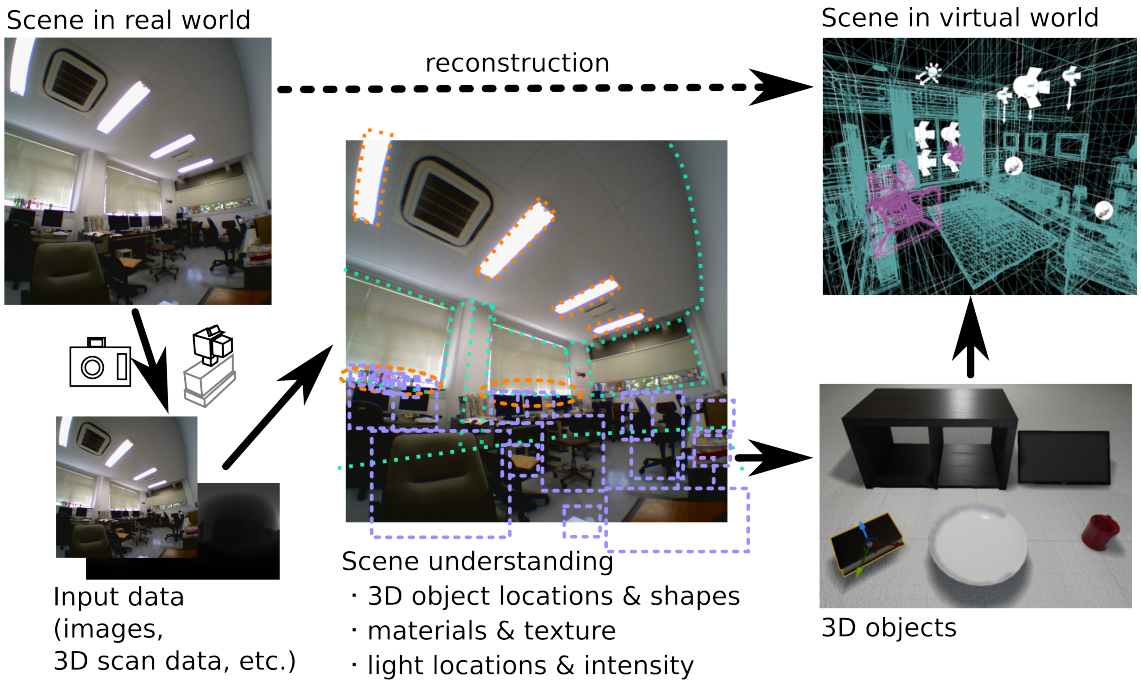

Figure 2.3: Conceptual dataflow in reconstruction.

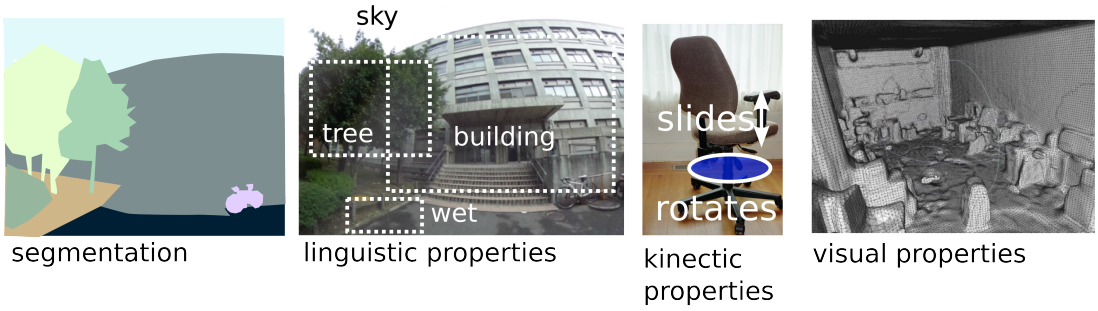

Figure 2.4: Four types of image understanding.

Existing methods[15][16] that can do segmentation require distance information from a depth camera like kinect, or a laser range finder (LRF). These special cameras can obtain distances between the camera and surfaces in each direction. Fig. 2.5 shows one of such result. It is easier to estimate or fit 3D shapes using such distance information compared to RGB images, because RGB images are heavily influenced by lighting conditions and surface materials while distance is not affected by lighting. In these methods, objects are segmented by fitting simple geometric shapes or finding repetitive shapes. However, these methods do not produce appearance models usable for synthesizing images, nor guarantee any physical correctness of the segmentation.

There are other methods that can output names of objects in an RGB image. Some can only recognize a single object occupying a whole image[17], and another can recognize

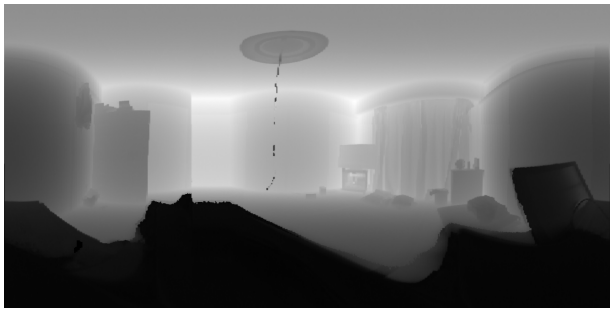

Figure 2.5: An example of distance image of a room. Brightness of each pixel is proportional to the distance between the camera and a point on a 3D surface pointed by the pixel.

approximate 2D locations of multiple objects as bounding rectangles[18]. However, they cannot output any models for rendering or physical simulation, and the detection is far less reliable than the previous group of segmentation methods based on 3D input data.

Another set of methods apply physical reasoning to 3D input data such as distance images, and try to extract physically plausible configuration of objects along with shape of each object[3][19][20]. Physical plausibility of object configuration is twofold. First, no objects should be intersecting with each other. Second, all objects should be stable at the initial state, which means all objects should be at local minima of potential energy. The latter condition is typically satisfied by all objects being supported by the floor either directly or indirectly. In one of the advanced methods[19], potential energy is explicitly minimized when estimating object boundary, and the method even outputs approximate shape of each object as voxels. However, the output shape is rough compared to dedicated segmentation method, and constructing a visual appearance model from such a rough model is relatively hard.

The last kind of methods[21][4] focuses on generating good-looking visual appearance model with little or no segmentation. Which means all objects in the scene are glued together, making it impossible to physically simulate behavior of individual objects. There is a method that can generate relatively clean geometric shape models of multiple objects[22], but the method requires manual interaction to specify approximate boundary of objects in images.

In the proposed system, a new reconstruction method to extract roughly segmented objects with visual appearance models, which also satisfies physical plausibility condition, is required. The reconstruction method approaches this hard problem by limiting the type of scene to a single room and using prior knowledge about such environment, rather than formulating it as a general problem.

Chapter 3

VR Earthquake Experience System

In this chapter, we closely look into the requirements and functions of the system, while considering various trade-offs between cost and realness. And then we propose a novel indoor reconstruction method, and a way to integrate a generated scene model and earthquake data into a virtual earthquake in VR.

3.1 Requirements and Functions of the VR Earthquake Experience System

As described in subsection 2.3.1, the ultimate goal of the proposed system is to cause fear toward earthquake, as part of fear-arousing communication, by providing an immersive earthquake experience via automatic scene reconstruction. The system only considers visual and auditory senses, because most other senses such as touch and olfaction are not strongly affected by a typical earthquake. Although the vestibular sense (i.e. feel of acceleration) is heavily affected by an earthquake, it is omitted because reliably stimulating it requires a large and expensive motion platform. Subsection 3.1.1 and subsection 3.1.2 discuss such trade-offs between realness and cost, from aspects of content generation and sensory stimulation, respectively.

The system goal is represented as two core requirements: automatic earthquake content generation and ability to cause fear. Automatic earthquake content generation can be divided into two sub-requirements, ability to output different scenes, and ability to vary seismic intensity scale. These sub-requirements are realized by data-driven approach. The system takes 3D scan data and earthquake data as inputs, and generates content based on the input. The other core requirement, ability to cause fear toward earthquake, consists of two sub-requirements, high realness of generated virtual scene and high memorability of the experience. The generated scene must represent earthquakes as relistic as possible,

because the system should cause fear toward earthquake, and not toward an arbitrary phenomena. Also, it is desirable the experience has high memorability, otherwise the system would be less effective in fear-arousing communication.

There is one additional requirement, that the system should not cause VR sickness[23]. This is because severe VR sickness would prevent users from using the system and would nullify other capabilities of the system.

Fig. 3.1 shows the three requirements and four sub-requirements and their dependency. These requirements are implemented as three subsystems, because the system needs to 1. capture a real scene, 2. analyze (reconstruct) it, and 3. present it using VR, as shown in Fig. 3.2. The functions of three subsystems are described in the following.

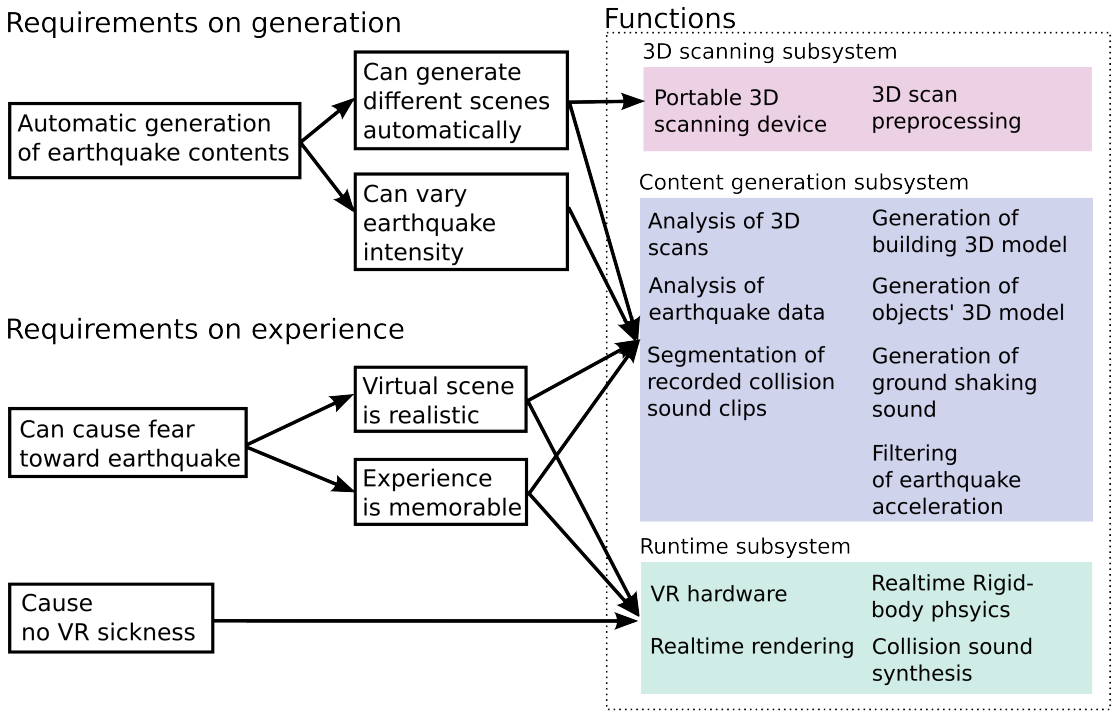

Figure 3.1: System requirements and functions of three subsystems. Arrows depict depends-on relations.

The 3D scanning subsystem is a portable 3D scanning device with software to preprocess the scanned data to make it easier for the content generation subsystem to analyze the 3D indoor environment. The scanner can collect multiple RGB and distance images by rotating a camera and an LRF, resulting in nearly omnidirectional coverage. Preprocessing includes color and exposure correction, and noise removal, which is essential to reduce spurious object detection and increasing final visual quality. The output of the subsystem is a collection of RGB and distance images in equirectangular coordinates, and 3D point clouds corresponding to different vantage points in a target indoor environment. A point cloud is a collection of 3D points, with optional attributes like colors and normals.

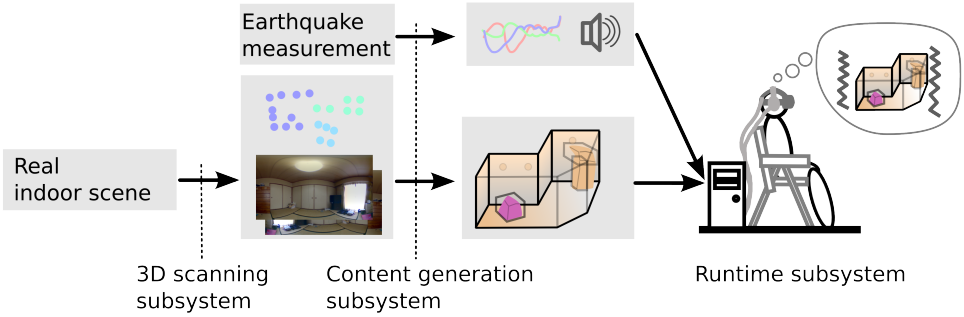

Figure 3.2: Dataflow between subsystems.

Fig. 3.3 shows a point cloud obtained by the subsystem, and the point cloud contains colors and normals, but normals are not shown in the figure. The detail of the subsystem is described in section 3.2.

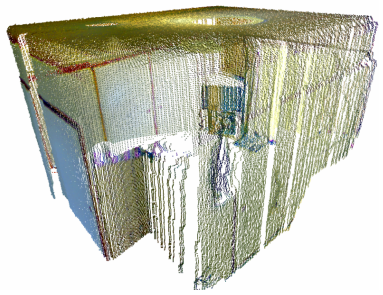

Figure 3.3: A point cloud obtained by the scanning subsystem, at a single scan location.

The content generation subsystem is a software that combines 3D scan data obtained by the scanner and user-specified earthquake data, and generates 3D content for VR by applying various analysis methods. More specifically, the inputs of the subsystem are 3D scan data taken at multiple locations, measured earthquake acceleration sequence, and recorded collision sound clips. The outputs of the subsystem are 3D content, earthquake sound (i.e. ground shaking sound), pre-processed collision sounds, and filtered earthquake acceleration.

The 3D contents are multiple 3D objects roughly corresponding to furnitures (will be called *interior objects*) and a single 3D object to represent ceiling, floor and walls combined (will be called *interior boundary*). These 3D objects not only have appearance model to render them, but also collision models for realtime rigid-body simulation. Since the content generation subsystem is the largest part, its details are described in several sections as follows.

- 3D content generation from 3D scan data: section 3.3
- Collision sound preprocessing: section 3.4

- Earthquake acceleration filtering and ground shaking sound generation: section 3.5

The runtime subsystem consists of VR hardware such as a head mounted display (HMD) and a headphone, and software to feed data into the hardware. The software takes the data generated by the content generation subsystem, and simulates objects' movement using filtered earthquake acceleration, and renders final visual and auditory stimuli to be fed into the hardware. These functions are enabled with a combination of a game engine and custom code. A game engine is a set of programs that provides a variety of functions in computer graphics, physics simulation, and VR hardware control. Since VR requires high framerate and stability to prevent VR sickness, earthquake-specific custom code is implemented on top of an existing game engine, which provides high quality implementation of the basic functions. Section 3.6 describes the details of the subsystem.

Design limitations of the system are discussed in the following subsections.

3.1.1 Aspects of Indoor Earthquake Modeled in the System

Quality and cost of graphics and sound contents would vary depending on whether they are created manually by artists, or automatically by a program. The differences forms a trade-off between high realness and low cost. The proposed system prefers the latter, because its goal is to cause fear toward earthquake in a inexpensive, thus easily adoptable way. However, there is a middle ground between fully manual and fully automatic contents creation, such as placing manually created furnitures automatically by recognizing the objects in the scene. In this subsection, 3D and sound contents related to indoor earthquake situations are discussed, and the design choices of the proposed system are explained.

Before simulating an earthquake, an indoor environment without earthquakes need be simulated. In a typical room without an earthquake, almost all objects are stationary, and there are only quiet ambient sounds caused by things such as fans of electronic devices. A significant exception is people and objects around them. Although behavior of other people under earthquake might have significant effect on the users' emotion including fear, they are omitted because simulation of people would add an insurmountable complexity to the system. Consequently, a real indoor scene is also assumed to be completely static and devoid of people while being scanned. Ambient sounds are also excluded from the simulation, because they would be inaudible because of auditory masking caused by much louder earthquake sounds.

When an earthquake occurs, the wave propagates through the ground and the building, and to the objects inside the room. Acceleration measurements at the ground level are widely available via monitoring networks such as K-NET[24], but the transfer function between the ground and the room is dependent on the building structure and other

factors. For example, it is known that resonance of a building to an earthquake could have significant impact on final motion[25]. In the previous earthquake simulation method[2], the final motions of the room and objects attached to it are calculated by finite element methods (FEM). The calculation is possible because structural model of the building is modeled manually in the previous method, allowing incorporation of material properties which are directly unobservable. In the proposed method, such reliance on external information that cannot be recognized automatically is avoided. Thus, all buildings and objects as treated as rigid bodies, as shown in the right of Fig. 3.4. Under this approximation, physics simulation can be further simplified by considering the room as an inertial frame with the acceleration equal to the acceleration measured on the ground, instead of objects shaken indirectly via moving floor. This simplification would increase stability and efficiency of rigid body simulation.

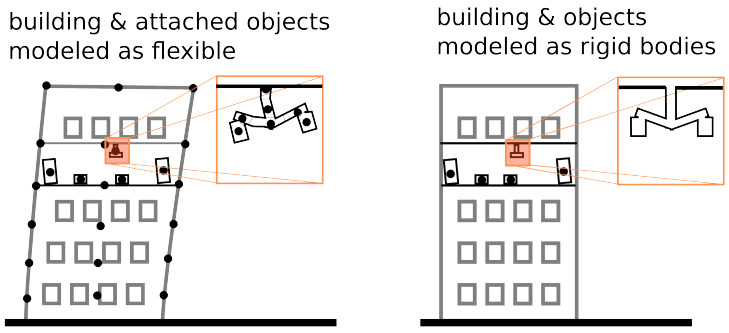

Figure 3.4: Difference of nodes (shown as black dots) between a FEM simulation (left) and a rigid body simulation in a proposed system (right).

When the earthquake waves reach the objects inside the room, several phenomena can occur. First, there would be a low frequency rumbling sound which is supposedly caused by ground shaking. Since there is an existing method[26] to synthesize this sound from accelerations, similar approach is taken in the proposed system. Second, small objects can tremble and make sounds, and furnitures can topple over and make large sounds when they hit the floor. These types of sounds can be generalized as collision sounds, and the motion can be simulated by rigid body simulation. Intuitively, the sounds are sum of short collision sounds caused by individual collisions. There is an existing method[27] that can create variations of recorded collision sounds for games. In the method, types of sounds are associated to types of objects manually, and the associated sound is played with slight random variation when an object collides. We take a similar approach, but since reliable recognition of types of objects is infeasible, a slightly different method that can accommodate the uncertainty of types is proposed.

Finally, there would be more complex interaction of objects that cannot be simulated

solely by rigid body physics. For instance, a cabinet door can spontaneously open and release its contents, a dish can shatter and produce numerous shards, or papers can fall from a desk and flap in midair. As discussed in section 2.4, recognition of objects with this level of detail is completely beyond the current state of the art, thus ignored in the proposed system.

In summary, the proposed system recognizes objects in an indoor scene at a scale of furnitures (e.g. cannot separate individual book in a bookshelf), and apply rigid body simulation and a generic collision sound synthesis method without information of detailed physical properties of each type of objects. This would enable the following aspects to be simulated.

- Rough appearance of objects
- Motion of objects (as rigid bodies)
- Collision sounds
- Ground shaking sound

3.1.2 Sensory Modes Stimulated by the System

There is a similar trade-off between realness and low-cost, when selecting the sensory modes to be simulated by the system. Obviously, stimulating as many senses as possible would maximize realness. However, there is a difference in difficulty of stimulating various senses. For instance, stimulation of vestibular and proprioceptive senses would require large and expensive apparatus such as a motion platform and an exoskeleton, but it is much easier to stimulate visual and auditory senses by using HMD and headphone. In the proposed system, only visual and auditory senses are targeted to lower the cost of the system adoption.

Assuming the user would not interact with objects, tactile sense can be ignored. Fig. 3.5 summarizes the relationships between the other senses and various elements in static (i.e. not shaking) and dynamic (i.e. shaking) scene. Non-simulated elements and modes are shown in gray.

Among the discarded senses, vestibular and proprioceptive senses are relatively important, because human can feel head acceleration and body posture by these senses. In a real earthquake, the whole body will be perceivably accelerated even in weak earthquakes, and the body posture would change in stronger ones. When simulating earthquake using VR, the acceleration and movement of the avatar, a virtual body of the user, should be reflected back to the real body so that the user can perceive a stimulus similar to that of real earthquake. However, known reliable methods to stimulate these senses are limited

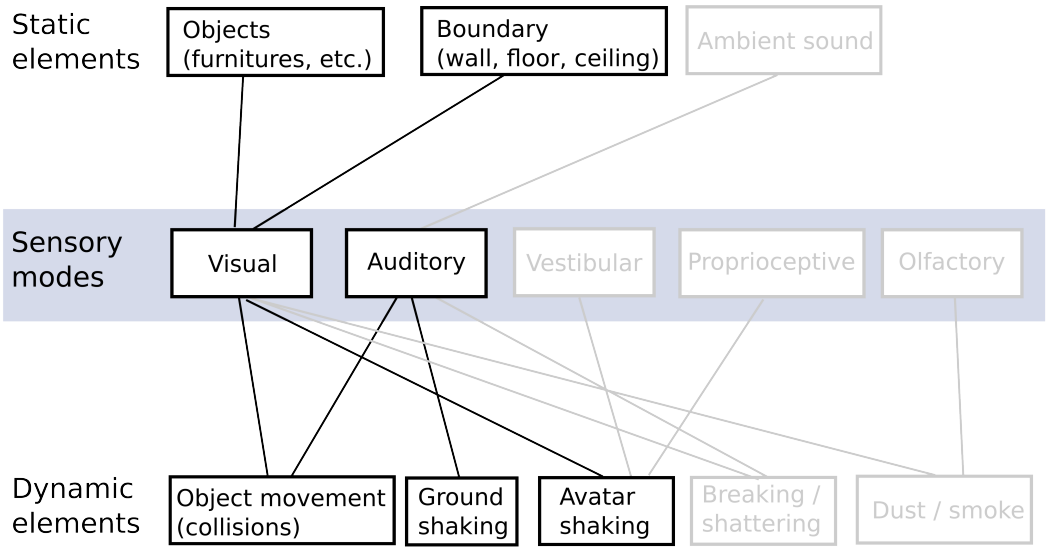

Figure 3.5: Phenomena in an earthquake and corresponding sensory modes.

to motion platforms and force-feedback exoskeleton, and there is a trade-off between realness and portability (or cost). In favor of the latter, we discarded these senses. Olfactory sense is omitted because it is rarely affected by an earthquake.

3.2 3D Scanning of Indoor Environment

The 3D scanner hardware used in the proposed system was developed in our lab as part of an augmented reality research[28]. However, this research requires higher quality data, so additional preprocessing is introduced, which is described in subsection 3.2.2.

3.2.1 3D Scanning Device

The hardware of the 3D scanning subsystem is identical to the previously developed scanner hardware, used for simulation of indoor lighting change using augmented reality [28]. Fig. 3.6 shows the scanner and its main components. The scanner is approximately 30 cm tall.

The scanner contains two sensors, a wide-angle RGB camera, and a laser range finder (LRF). These two sensors are attached on top of a rotating platform. An LRF can be though of as a one-dimensional distance camera, which emits infrared laser to various directions and records intensity of the reflected light and time of flight. From the time of flight, distance between the LRF and a point on a surface can be calculated. The points swept by the LRF forms a curve, called a scanline. Fig. 3.7 shows this operation of an LRF.

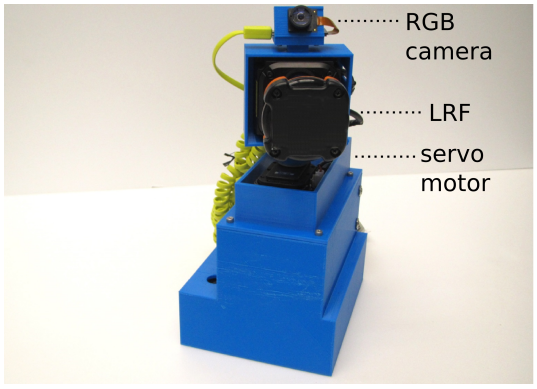

Figure 3.6: Photo of the 3D scanner[28].

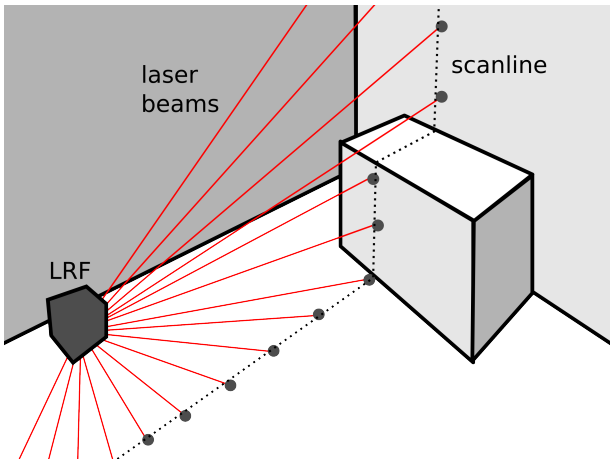

Figure 3.7: Operation principle of LRF.

Since neither the RGB camera nor the LRF covers the whole solid angle, the upper part of the scanner that contains two sensors is rotated by a servo motor to collect multiple images and scanlines at different angles. Fig. 3.8 shows fields of view of both sensors when they are not rotating, and Fig. 3.9 shows rotation of the sensors in top view.

RGB images are collected at interval of 5°, and LRF scanlines are collected at interval of 0.5°. The scanner can cover most of the solid angle, except polar regions. Specifications of the RGB camera and the LRF are shown in Table 3.1 and Table 3.2. The specification of the servo motor used to rotate them is also shown in Table 3.3.

Table 3.1: Specification of the RGB camera in the scanner.

Product name	Asahi Electronics NCM13-K
Imaging device	1/4 inch CMOS
Resolution	1280×1024
Output format	YUV422 / RGB565
Field of view	horizontal:135° vertical:107° diagonal:165°

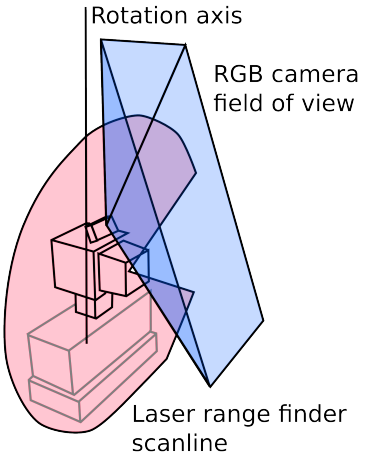

Figure 3.8: Fields of view of the sensors when not rotated.

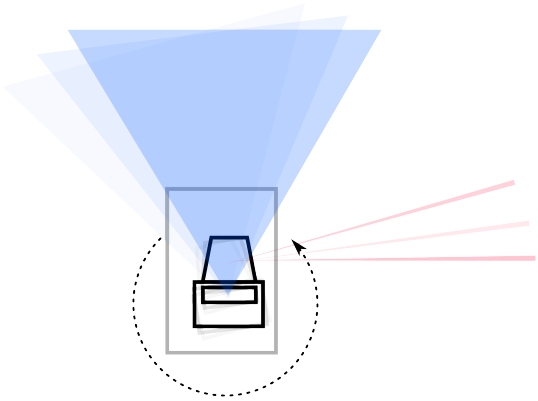

Figure 3.9: Operation of scanner (top view).

Table 3.2: Specification of the LRF in the scanner.

Product name	Hokuyo UTM-30LX
Light source	Laser diode ($\lambda = 905\,\text{nm}$)
Distance range	0.1m-30m
Distance accuracy	0.1m-10m: $\pm 30\,\text{mm}$, 10m-30m: $\pm 50\,\text{mm}$
Distance resolution	1mm
Distance precision	0.1m-10m: $\sigma < 10\,\text{mm}$, 10m-30m: $\sigma < 30\,\text{mm}$
Scanning angle	270°
Angular resolution	0.25°

Table 3.3: Specification of the servo motor in the scanner.

Product name	Dynamixel MX-28R
Stall torque	$2.5\,\text{N} \cdot \text{m}$
Positioning	0°-360°, at 12 bit resolution

3.2.2 Panorama Stitching

Unlike the previous research[28] which uses 3D scan data only for slightly modulating a photo, the proposed method reconstructs 3D meshes for each objects which are directly shown to the users. Thus, visual consistency of RGB images is important. Also, the raw RGB images collected by the scanner contain a large amount of redundancy because of overlaps. This redundancy is removed by merging them into an equirectangular image to increase efficiency of subsequent processes. To merge the images, poses of the camera when the images are taken need to be known. An existing panorama stiching for manually taken photos estimate the relative poses by using feature point matching[29]. In this system, relative poses of the images are calculated from the angles of the servo motor and pre-calibrated dimensions of the scanner, rather than estimated from images themselves.

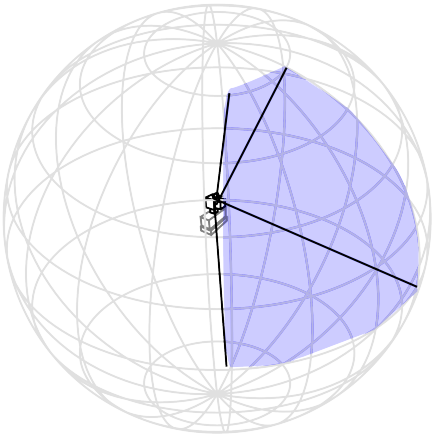

Figure 3.10: A solid angle covered by a single RGB image.

Each RGB image fills a part of the whole solid angle, which can be represented as a region on a sphere as shown in Fig. 3.10. The RGB images will cover a large portion of the sphere with overlaps. Although the sphere is an elegant representation with no singularities, it is inconvenient for image processing, because most image processing algorithms has been developed to process 2D images. Thus, equirectangular projection, which simply maps spherical coordinates (θ, ϕ) to X and Y axis of 2D Euclidean coordinates, is used to convert the image on the sphere to 2D rectangular image. In this thesis, this kind of projected image will be called simply as an equirectangular image. Fig. 3.11 shows the spherical coordinates (θ, ϕ) and its planar projection. Since $0 \leq \theta \leq \pi$ and $0 \leq \phi \leq 2\pi$, the width of an equirectangular image is double of the height.

A single RGB image warped to an equirectangular image is shown in Fig. 3.12. Mapping all RGB images similary and smoothly blending them gives us Fig. 3.13. These images are typical results of operating the scanner on a desk in a room.

Each equirectangular RGB image I_i is indexed by a unique integer i, and the merged

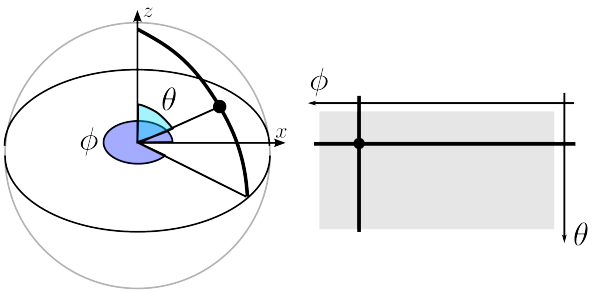

Figure 3.11: Equirectangular projection of spherical coordinates.

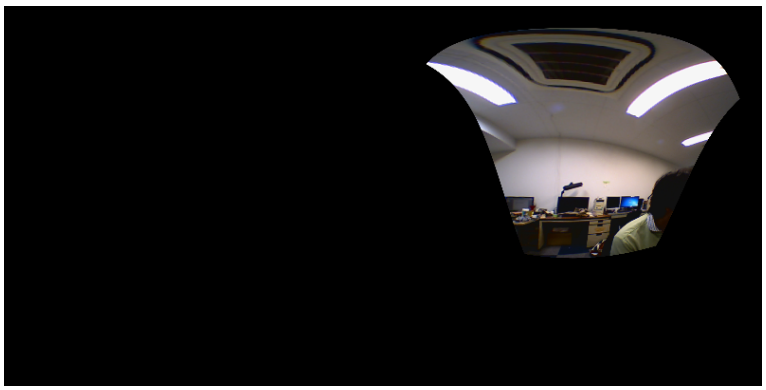

Figure 3.12: A single RGB image warped to equirectangular coordinates.

equirectangular image I can be calculated by smoothly blending the images using Gaussian weight function with the following formula.

$$I(\theta, \phi) = \frac{\sum_i I_i(\theta, \phi) w_i(\phi)}{\sum_i w_i(\phi)} \tag{3.1}$$

$$w_i(\phi) = \exp\left(-\frac{(\phi - \phi_i)^2}{\sigma^2}\right) \tag{3.2}$$

where ϕ_i is the center of scan i. σ is a blending constant and set to approximately 2.5° (half of the capturing interval).

Notice there are several bands of different color and brightness in Fig. 3.13. These artifacts are caused by auto exposure (AE) and auto white balance (AWB) of the RGB camera. We negate AE and AWB by calculating color multiplier α_i for each RGB image, and then applying multiplication before merging images. The multipliers are calculated to equalize average colors of neighboring images in the overlap region. The overlap region is approximated by weighting function $w_{ij}(\phi) = w_i(\phi)w_j(\phi)$. Using this function, multipliers can be calculated as solution of Eq. (3.3).

Figure 3.13: Stitched RGB image.

$$\alpha_i \sum_{\theta,\phi} w_{ij}(\phi) I_i(\theta,\phi) = \alpha_j \sum_{\theta,\phi} w_{ij}(\phi) I_j(\theta,\phi) \tag{3.3}$$

Also, we have a normalization $\Pi_i \alpha_i = 1$. These equations form an over-constrained system, but the solution generally exists in practice.

The merged image after color correction is shown in Fig. 3.14.

Figure 3.14: Stitched RGB image with color correction.

Notice the seams between bands are almost invisible. We also apply vignetting removal using naive $\cos^4$ model[30] to RGB images before merging.

3.2.3 Distance Image Noise Removal and Point Cloud Generation

Equirectangular images are suitable for preserving topology of neighboring pixels, but are unsuitable for operations such as 3D plane fitting which is required for ceiling detection. Another problem of using an equirectangular distance image for 3D processing is

that resampling of the distance is required for virtually any operation, including translation and rotation. Resampling of the LRF distance is carefully avoided in the proposed system, because resampling causes artifacts regardless of interpolation method. Fig. 3.15 shows an original data, and results containing artifacts when resampled at triple resolution with linear interpolation and no interpolation (i.e. nearest neighbor lookup). A pertinent format is a point cloud with colors and normals, which can be represented as a set of (point, color, normal). In this section, however, we extensively use resampled equirectangular images for visualization purpose.

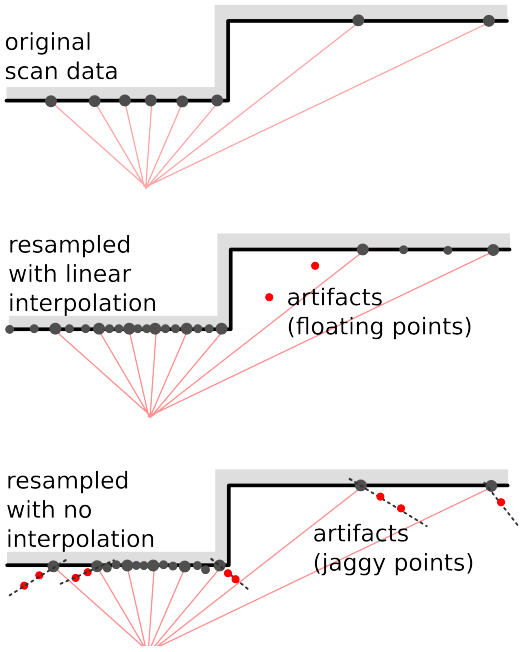

Figure 3.15: Resampling artifacts of distance images.

The raw LRF data contains distance noise of a few centimeter, which is prohibitively large for computing normals from adjacent points. Fig. 3.16 shows an example of point cloud generated without noise reduction. Thus, it is necessary to reduce the noise in distance data.

Normals of each point is calculated as a cross product of tangent vectors. The distances and positions of the points are stored in 2D arrays without any resampling.

$$N(x,y) = \frac{\partial P(x,y)/\partial x \times \partial P(x,y)/\partial y}{\|\partial P(x,y)/\partial x \times \partial P(x,y)/\partial y\|} \tag{3.4}$$

where $N(x,y)$ is normal, and $P(x,y)$ is 3D position at pixel (x,y). P can be calculated from distance image D by multiplying distance and a unit direction vector calculated from spherical coordinates (θ,ϕ). Fig. 3.17 shows an example of D. Although D looks smooth visually, calculating normals from derivatives amplifies the noise, resulting in unusable

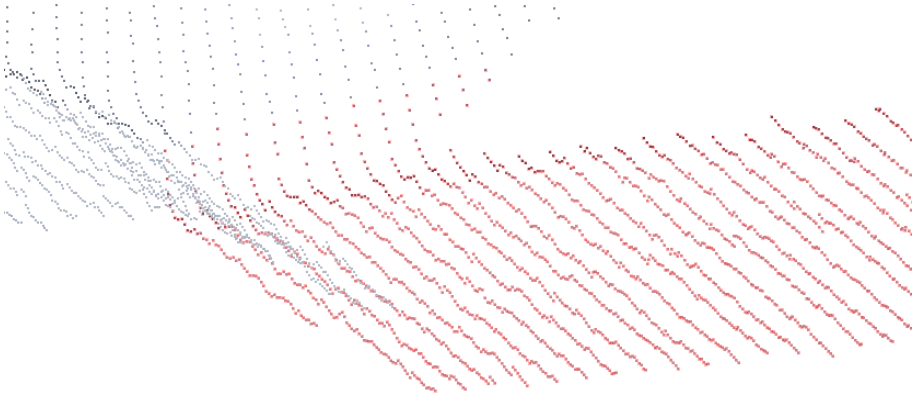

Figure 3.16: A view of a flat region in a point cloud, showing errors of a few centimeters. The flat region is highlighted in orange.

normals shown in Fig. 3.18.

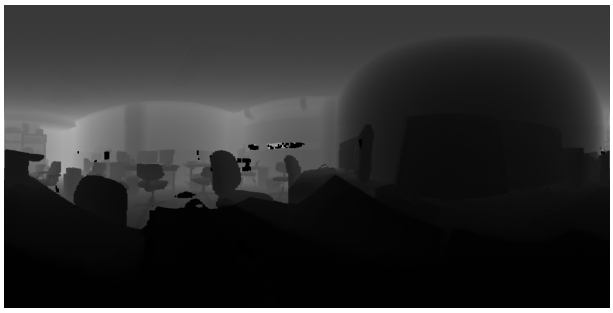

Figure 3.17: Distance image D. Pixel brightness is proportional to distance.

A naive Gaussian blurring of distance images will reduce the noise, but it will also blur the edges. We uses a trilateral filter[31] to blur distance image while keeping edges relatively intact. While a Gaussian blurring filter takes only one image and blurs it, a trilateral filter takes one additional image for edge information extraction purpose.

However, we cannot get edge information from distance image, because D is continuous at edges, and only its derivatives are discontinuous, which is noisy when numerically computed. The RGB image is good in theory, because the amount of light reflected by a surface is heavily influenced by the surface normal. However, the RGB image has offset error of a few pixels relative to LRF images, making it unsuitable. Fortunately, the LRF outputs reflected infrared intensity image R, which is shown in Fig. 3.19.

Since the LRF scans by emitting laser and measuring its reflection, the reflected intensity image R is equivalent to an infrared photo taken at the position of the LRF, with a single infrared point light also at the LRF position illuminating the whole scene. The intensity image contains edge in a clearer way, so R is used at the input of trilateral filtering. A normal image N computed from denoised D is shown in Fig. 3.20.

Note that the image now contains less high frequency noise, and edges are mostly

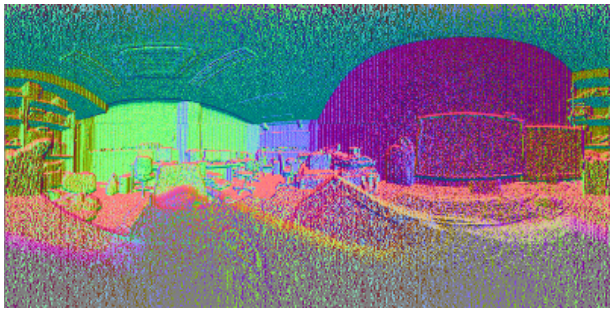

Figure 3.18: Normals calculated from raw distance image. (XYZ components are visualized as RGB color.)

Figure 3.19: Intensity of reflected infrared light.

preserved without getting blurred. The empirically found optimal parameters for the trilateral filtering are $3\,\text{px}, 0.2\,\text{m}, 100$, when the range of the intensity values is $[0, 2^{15}]$ (which is specific to the model of the LRF used in the scanner). Finally, P and N are converted to a point cloud, and the points are colored by looking up the RGB equirectangular image. Fig. 3.21 shows a point cloud obtained by scanning a room from a single location.

3.3 Indoor Environment Reconstruction Method

3.3.1 Overview

In indoor reconstruction step, scene model is generated from 3D scan data. However, as discussed in section 2.4, scene reconstruction is a very hard problem and various compromises are necessary, some of which forms a certain trade-off. The trade-off is between accuracy (i.e. conformity of generated scene model to input data), and realness (i.e. perceived probability of generated scene model occurring in reality). Note that this trade-off is different from the trade-off between cost and realness discussed in subsection 3.1.1. The proposed indoor environment reconstruction method always favors realness to accuracy, because the goal of the system is to trigger fear via realness. The overall dataflow of the reconstruction method is shown in Fig. 3.22.

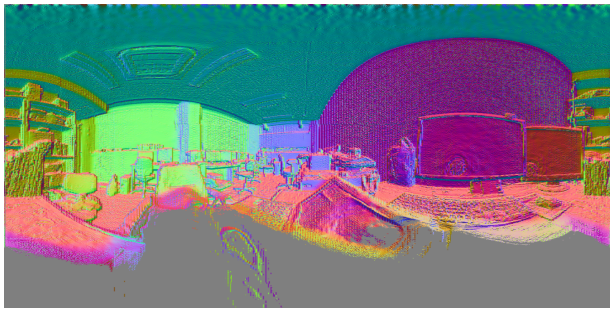

Figure 3.20: Normals calculated from filtered raw distance image.

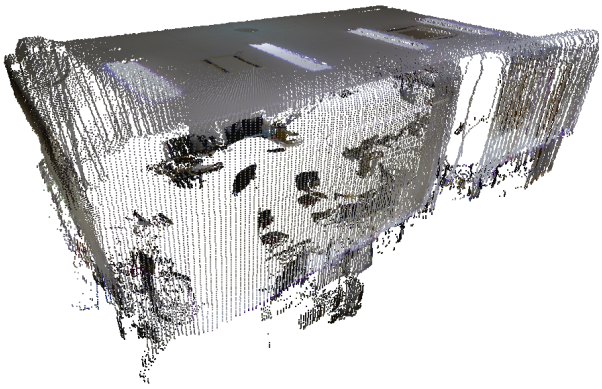

Figure 3.21: A point cloud generated from 3D scan data at a single location.

Initially, the system does not have locations of the scanner at the times of scanning. These locations, or poses, are calculated by aligning 3D scan data into a single point cloud. However, the merged point cloud contains ghosting artifact due to alignment error and scan distortion. So the merged point cloud is only used to extract a *room frame*, which represents overall room shape as an extruded polygon. The extruded polygon is equivalent to a pair of 2D floor map and room height. When calculating the room frame, Manhattan world assumption[32] is used as a hint to make a room frame visually pleasant. Under Manhattan world assumption, all surfaces in a scene are flat and their normals are parallel to either X, Y or Z axis. Although this holds true for many man-made objects, the proposed method does not strictly rely on it because there are also a significant number of exceptions.

Computation after the room frame extraction is based on either individual point clouds and equirectangular images, as they do not suffer from ghosting artifact. The remaining process can be divided into three independent processes of light detection, boundary generation and object extraction.

In the ceiling light detection process, lights in the room are detected from an equirectangular image. Lights are required for generating dynamic shadows when objects move in an earthquake. In the boundary generation process, texture of the room is calculated

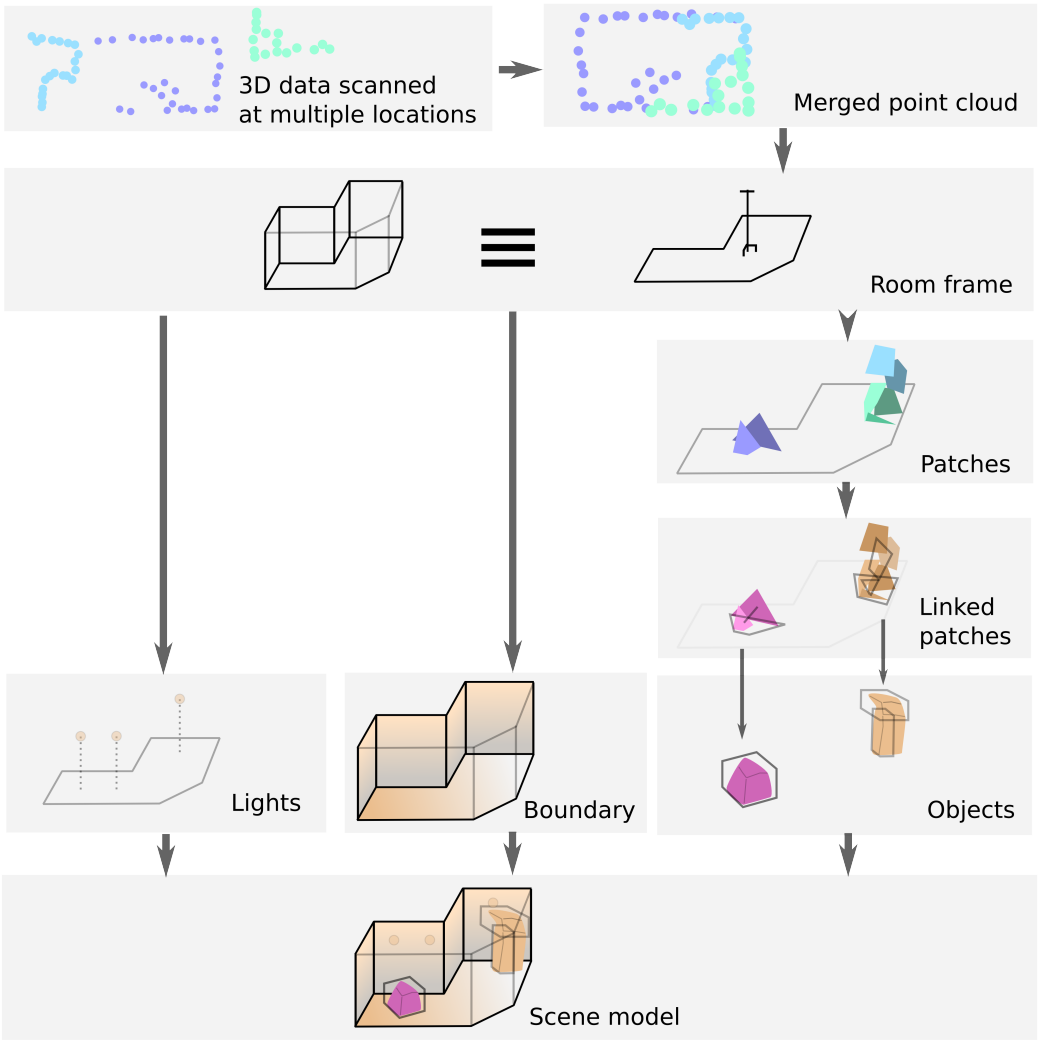

Figure 3.22: Dataflow of indoor reconstruction pipeline.

from an equirectangular image, and then the missing part of the texture occluded by objects are inpainted. In the object extraction process, points corresponding to interior boundary (such as walls) are removed using the room frame, and the remaining point clouds are divided into small patches by geometrical proximity. Since each patch is a subset of the point clouds, a patch itself is also a point cloud. A patch typically represents an incomplete surface within a single object. Then, the patches are linked together to form physically stable *clusters*. A cluster is a set of patches, and it corresponds to a single object detected by the system. Finally, appearance and physical models of objects are created from the clusters.

Finally, the boundary, the objects and the lights are combined to form a complete scene model.

3.3.2 Alignment of 3D Scan Data

The 3D scan data comprises of equirectangular images and point clouds taken from multiple locations. To detect objects in the scene, the system needs to recover the relative locations of the point clouds, which are arbitrary chosen by the operator to maximize a coverage of the scanned room. The process of estimating the locations and forming a single merged point cloud is called alignment.

The alignment is known as a hard problem in a general case. Although several methods[33][34] are known, their accuracy is mostly unpredictable. In the proposed system, a typical property of an indoor room, specifically presence of a flat ceiling, is exploited. Unlike walls and floors which are often occluded by objects, a ceiling can be observed with high probability by the scanner, making it suitable to use as a feature to partially align point clouds.

First, point clouds are partially aligned by using the ceiling as a feature observable in all scan data. After ceiling alignment, the point clouds are constrained to translation and rotation on a 2D plane, leaving only 3 degrees of freedom (DoF). Second, a novel pose-invariant similarity measure is used to associate the point clouds by similarity. This results in a pose tree, from which relative poses of point clouds can be uniquely determined. Finally, a pairwise fine alignment method called iterative closest point (ICP)[35] is applied to the pose tree, and finely aligned relative poses are calculated from the pose tree.

Partial Alignment using Ceiling as a Feature

When the scanner is used, it is placed on a nearly level surface in a location accessible to a human. In conjunction with the fact that the scanner is nearly omnidirectional, there is a high chance that each scan data contains the ceiling. Thus, the ceiling can be used as a feature to align the point clouds partially.

In a scan data, the ceiling forms a nearly level plane because of the ceiling. To detect this plane, RANdom SAmple Consensus (RANSAC)[36] is applied to ceiling candidate points P_C. P_C is calculated as follows from the point cloud P, assuming Z+ corresponds to up direction.

$$P_C = \{p \,|\, p \in P,\, p_z > (\max_{p' \in P} p'_z) - k_{\text{margin}}\} \tag{3.5}$$

where p_z is the z coordinate of point p. k_{margin} is empirically set to 50 cm, so that other large planes such as desks will be rejected, while ensuring to retain the points originating from the true ceiling even in the presence of recessed ceiling decoration.

Point clouds are rotated and translated so that the detected planes become completely coplanar. After this, we only need to consider 2 translational DoFs and 1 rotational DoF

in a plane when trying to align the point clouds completely.

Pose-invariant Similarity Measure

First, a non pose-invariant similarity measure is defined, and then a pose-invariant similarity measure is derived from it assuming the points clouds are partially aligned by the ceiling feature.

The (non pose-invariant) distance d between two point clouds P_1, P_2 is defined using Eq. (3.6). The distance d is a similarity measure, calculated by first binning the points into voxels (cells of 3D lattice), and then comparing average colors and normals in each voxel.

$$d(P_1, P_2) = \frac{1}{M} \sum_{\substack{\mathbf{i} \in \mathbb{Z}^3 \\ \text{if } |V(P_1,\mathbf{i})|>0 \\ \text{and } |V(P_2,\mathbf{i})|>0}} k_n N(V(P_1, \mathbf{i})) \cdot N(V(P_2, \mathbf{i})) + k_c \|C(V(P_1, \mathbf{i})) - C(V(P_2, \mathbf{i}))\| \tag{3.6}$$

$$N(P) = \frac{\sum_{p \in P} (p_x\, p_y\, p_z)^\top}{\| \sum_{p \in P} (p_x\, p_y\, p_z)^\top \|} \tag{3.7}$$

$$C(P) = \frac{1}{|P|} \sum_{p \in P} (p_r\, p_g\, p_b)^\top \tag{3.8}$$

$$V(P, (i, j, k)) = \{p \,|\, p \in P,\ \lfloor p_x/s \rfloor = i,\ \lfloor p_y/s \rfloor = j,\ \lfloor p_z/s \rfloor = k\} \tag{3.9}$$

where M is a number of voxels which contain one or more points from both P_1 and P_2, s is a size of voxel, and k_n and k_c are weights of normal similarity and color similarity. The weights and the voxel size are empirically set to $k_n = 5$, $k_c = 0.01$, and $s = 15\,\text{cm}$ so that d can distinguish different point clouds while not being too sensitive to noise in color and normal.

To create a pose-invariant similarity measure, we need to cancel the effect of translation and rotation on the measure. Basically, multiple variations of a point cloud are created, and each variation is compared to the other point cloud using d. By creating enough variations, one of the variations would correctly align. Thus, using the minimum of d as a measure, a new pose-invartiant measure d_{inv} can be defined.

First, function E that creates variants of a point cloud by rotation, is defined. Also, let A be a function that translates a point cloud so that the center coincides with the origin.

$$E(P) = \{R_z(\frac{2\pi k}{k_{\text{rot}}})P \mid k \in \{1, 2, \ldots, k_{\text{rot}}\}\} \tag{3.10}$$

$$A(P) = \{p - \text{center}(P) \mid p \in P\} \tag{3.11}$$

where center(P) is the center of axis-aligned bounding box of P, $R_z(\theta)$ is rotation along Z axis by angle θ, and k_{rot} is a constant specifying number of generated variants. When k_{rot} is smaller, fewer variants are generated and the computation is faster. However, k_{rot} also needs to be large enough so that the difference of rotation angles between the point clouds (P_1, P_2 in Eq. (3.12)) becomes small enough for the later fine alignment to converge correctly. Empirically, $k_{\text{rot}} = 60$ is found to satisfy both conditions.

In most case, the center of a scan data is the same as the center of the whole room, due to the omnidirectional nature of the scanner. Thus, the center of a point cloud can be used a reliable feature to normalize the translation of point clouds in A. Note that there is a subtle difference in calculating the center in a partially aligned point cloud, and detecting the walls in a unaligned point cloud. The former is significantly more reliable than the latter, because only a single point is enough to calculate the center correctly, but a significant portion of the walls must be observed for direct wall detection.

Unlike translation, rotation cannot be normalized by a simple feature like center of the point cloud. Thus, E samples uniformly from all possible rotations. By combining A and E, the pose-invariant distance d_{inv} of two point clouds can be defined like Eq. (3.12). By keeping track of which variation minimizes d_{inv}, a transform $T_{i \to j}$ that transforms P_i to P_j can be recovered. Accuracy of $T_{i \to j}$, however, can be affected by various factors, such as an amount of noise and how well assumptions (e.g. ceiling feature) hold.

$$d_{\text{inv}}(P_1, P_2) = \min_{P_2' \in E(P_2)} d(A(P_1), A(P_2')) \tag{3.12}$$

Construction of Pose Tree

By using the pose-invariant distance measure d_{inv}, point clouds can be paired against similar point clouds. For all pairs of point clouds, d_{inv} is calculated. The result can be expressed as a symmetric matrix, which is shown in Fig. 3.23. The matrix can be also seen as a graph, whose node is a point cloud and edge is d_{inv}, as shown in the left of Fig. 3.24.

To calculate relative pose $T_{i \to j}$ between P_i and P_j, each transforms on the path need to be multiplied. However, since there are multiple possible paths between nodes in a graph and the transforms are inexact, different paths would result in different relative poses. To uniquely determine relative poses of all point cloud, only one path should exist between any two nodes, as shown in the right of Fig. 3.24. In other words, we need to form a spanning tree of the graph, which we call a pose tree. Also, the sum of d_{inv} in a spanning tree should be minimized, because transforms between similar point clouds (i.e. smaller d_{inv}) have higher chance of being accurate. A pose tree with minimum sum of d_{inv} can be calculated by applying Prim's algorithm[37] to the graph. An example of obtained

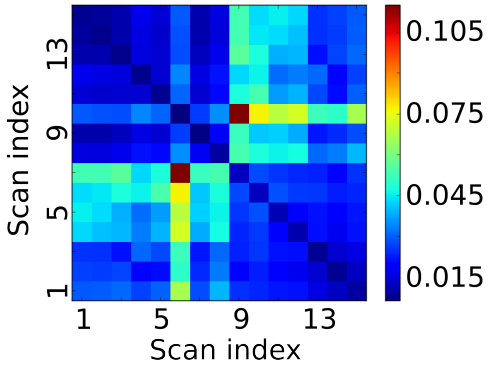

Figure 3.23: d_{inv} for all point cloud pairs.

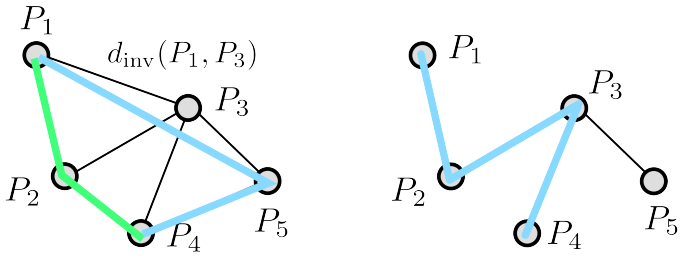

Figure 3.24: Multiple paths from a node to another in a graph (left). There is only single path between nodes in a tree (right).

pose tree is shown in Fig. 3.25.

Recovering Poses from Pose Tree

Although the pose tree provides a way to uniquely determine the relative poses, transforms corresponding to edges of the pose tree have accuracy only up to voxel size s. To obtain the fine relative pose, each edge is finely aligned by a variant of ICP method implemented in point cloud library[38]. An ICP method takes two coarsely aligned point clouds, and aligns them by iteratively minimizing distance between points like gradient descent. The ICP method is applied to all edges of the pose tree, and now finely aligned relative poses can be obtained from the pose tree in the same way coarse relative poses are obtained. We can also merge the all point clouds into a single point cloud by using the fine relative pose. After the alignment is complete, all point clouds are merged into a single point cloud.

Although the alignment is automatic, some scan data could converge to a wrong pose, in which case a manual adjustment is needed.

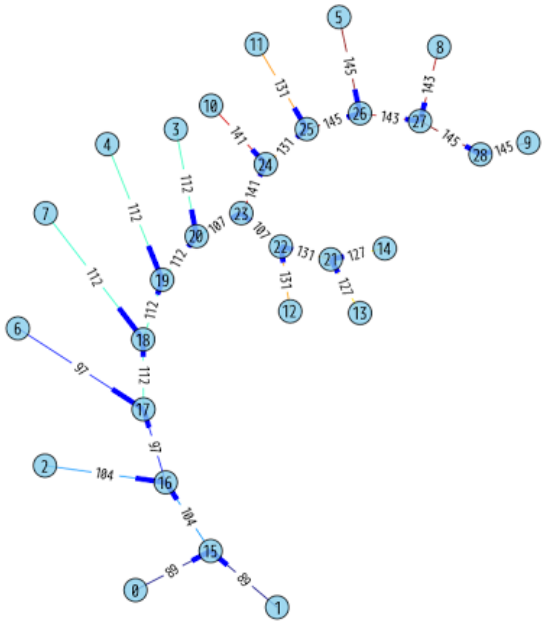

Figure 3.25: A minimum spanning tree of the point clouds. Each node is a point cloud, and each edge is d_{inv} between two point clouds.

3.3.3 Room Frame Extraction

The reconstruction target of the system is an enclosed room, whose walls and ceiling are mostly flat. Although point clouds and equirectangular images contain all information obtained by the scanner, it is useful to have an intermediate representation of the room shape, which is called a *room frame* in this thesis. A room frame is used in subsequent processes such as light detection and object reconstruction. In room frame extraction process, Manhattan world assumption[32] is partially enforced to reduce the noise and to produce more visually pleasant and less computationally demanding results.

Both scan data and a room frame are represented in 3D Cartesian coordinates whose Z axis represents the vertical direction. A room frame represents the shape of a room as an extruded polygon. Thus, a room frame is a composition of 2D polygon in XY coordinates and two values to specify the range of the room on Z axis.

A merged point cloud is projected onto the XY plane, and the resulting 2D points are used to calculate the 2D polygon. An example of projected points is shown in Fig. 3.26. In Fig. 3.26, it can be observed that the region corresponding to inside of the room is very dense, and the outliers are relatively sparse. Upon close inspection of the point cloud, outliers typically originate from exterior scene captured through transparent windows, or from interior scene reflected by windows or other mirror-like objects.

Prior to 2D polygon estimation, the projected points are filtered by a 2D voxel filter of 10 cm. The voxel filter replaces the points in a given voxel with the average of the points.

Figure 3.26: 2D projection of a merged point cloud, consisting of 4,065,020 points. In this 7 m × 4 m office, 15 scan data were taken.

By the filtering, point density is equalized across the space, and most of the outliers are decimated as shown in Fig. 3.27.

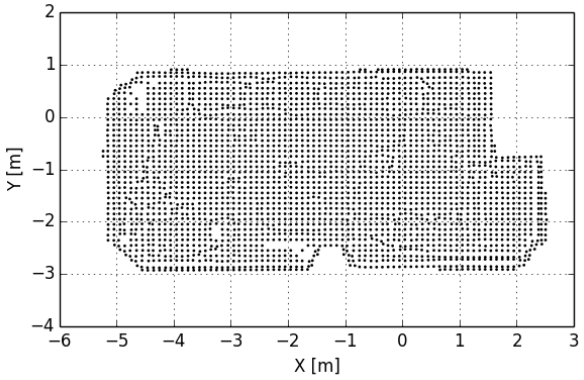

Figure 3.27: 2D Points after filtering.

Then, a concave hull is calculated from the filtered points. Unlike the convex hull, the concave hull is ill-defined and the results depends on a selection of a method and its parameters. For room frame extraction, concave hull extraction method by Moreira et al.[39] is used. The method takes an extreme point in a direction as the input, and a concave hull containing larger density region is carved out from the starting point. Since the outliers are highly decimated by the filtering, the method would obtain a convex hull corresponding the room shape, if given a correct starting point.

This is achieved with a rejection sampling method with the following steps. First, the extreme point in a randomly chosen direction is chosen as the starting point. Then, the resulting concave hull is accepted as correct if it is large enough in area, and not too narrow. More concretely, it will be rejected if it spans less than 1 m in any direction. Otherwise, it is rejected and a new starting point is chosen. These steps are repeated until an acceptable solution is found. A result of the process is shown in Fig. 3.28.

The concave hull expresses the shape of the room at a moderate quality, but there

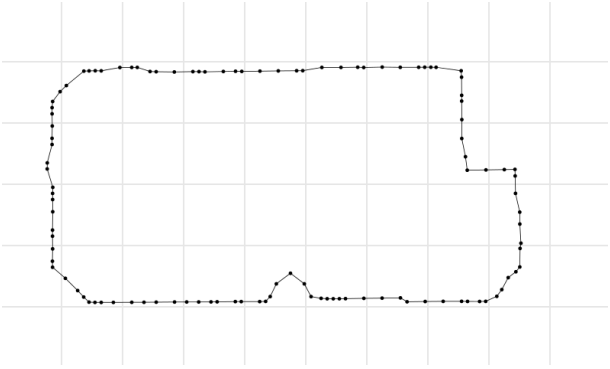

Figure 3.28: A concave hull extracted from a filtered points.

are two issues that prevent using the shape as the 2D polygon of the room frame. First, the shape is too noisy. In particular, false unevenness in flat regions and slightly off right angles would result in visually unpleasant and unnatural look, as shown in Fig. 3.29. Second, there are too many vertices in the shape to allow efficient querying such as inside/outside test. However, it is problematic to enforce the Manhattan world assumption completely, because it will obscure the chamfered edges at the corners of the room. The presence of chamfered or rounded corners is not limited to the example, but they are also common in a typical indoor environment.

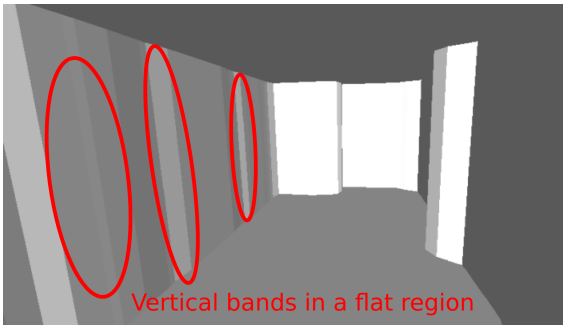

Figure 3.29: Vertical bands on a flat region, caused by noisy normals.

Manhattan world assumption is partially enforced by forming a non-uniform grid, and then *snapping* vertices to the grid. First, a histogram of the edge normals is calculated, and the principle axes X' and Y' are estimated from its mode. Then, a non-uniform structured grid is formed by clustering the X' and Y' elements of the vertices independently.

To this end, the k-means method is used. The k-means method is a clustering method, that can detect k groups of concentrated samples. Typically, k must be specified manually, but in the proposed method, k is automatically determined by minimizing Akaike Information Criterion (AIC)[40]. The values can be though of as being sampled from unknown true values with a certain error distribution. Intuitively, AIC can find an appropriate balance between likelihood of samples and information required to represent the

clusters. Relationship between the samples and the clusters are shown in Fig. 3.30.

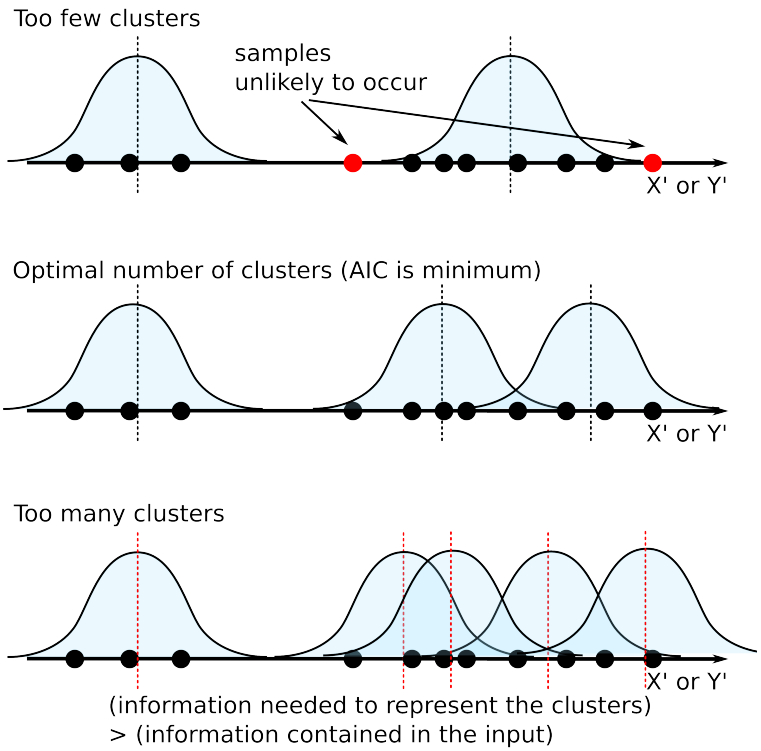

Figure 3.30: Samples (input values) and clusters, seen from the AIC perspective.

The statistical model used for AIC calculation is shown in Eq. (3.13), and the obtained clusters are shown as faint red and green lines in Fig. 3.31.

$$\mathrm{AIC}(n) = 2k - 2\ln \mathcal{L}(n) \tag{3.13}$$

$$\mathcal{L} = \prod_{0 \le i < m} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - x_i')^2}{2\sigma^2}} \tag{3.14}$$

where σ is the standard deviation in a Gaussian error model (set to 5 cm), x_i is the coordinate the i-th projected vertex, and x_i' is the nearest cluster centroid.

The vertices are snapped to the closest the non-uniform grid, if they are close enough to it. Specifically, each component (X', Y') of a vertex is truncated to the nearest grid line if the nearest grid line is within distance of σ. Otherwise, a component of a vertex is left as-is. Consecutive collinear edges, and other such degeneracy are removed from the snapped vertices. The final result is shown in Fig. 3.31. Notice that straight edges and chamfer are preserved with considerably less vertices compared to Fig. 3.28.

Points outside the room is rejected using the 2D polygon, and the Z range of the room frame is calculated simply as 1st and 99th percentiles of the Z coordinates of the inside points.

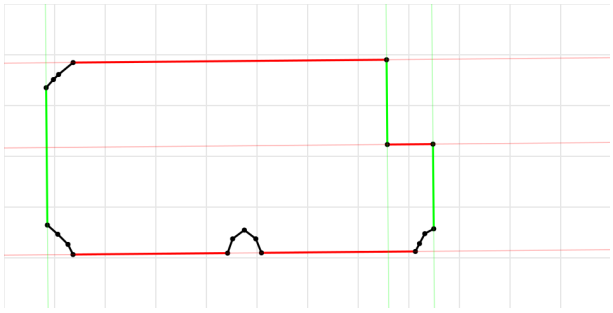

Figure 3.31: Detected Manhattan frame.

3.3.4 Ceiling Light Detection

A real world scene typically contains one or more light sources that illuminate the scene, and a reconstructed scene also requires light sources to realistically simulate shades and shadows of the objects inside the room. In the reconstructed scene, all lights are approximated by point lights near the ceiling. They are detected through image processing of an equirectangular image projected to the ceiling.

Since one RGB equirectangular image is contained in scan data from a single location, the number of the available RGB equirectangular images is the same as the number of locations. To select the optimal equirectangular image among them, average resolution of the projection is used as the quality criterion. The resolution of projected image varies spatially, and is proportional to $\partial L/\partial\theta$ as shown in Fig. 3.32.

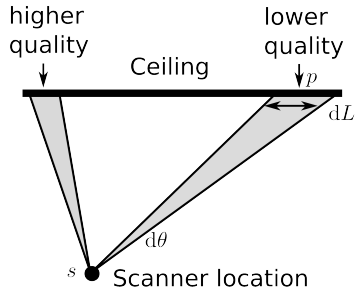

Figure 3.32: Quality of image at various points on the ceiling.

The resolution can be calculated by Eq. (3.15).

$$\frac{\partial L}{\partial \theta} = \frac{\|s-p\|}{\frac{s-p}{\|s-p\|}\cdot n} \tag{3.15}$$

where p is coordinates of a 3D point on the ceiling, s is the scanner location, and $n = (0\ 0\ -1)^\top$ is the normal of the ceiling.

The projection of the optimal equirectangular image to the ceiling is shown in the left of Fig. 3.33. Lights are first detected in the 2D image, and then converted to 3D

positions. Lights in the image are detected by a method similar to an existing work[28]. The projected image is converted to grayscale, and binarized by thresholding at 60% of the maximum brightness. The binarized image is shown in the right of Fig. 3.33. Then the blobs in the binary image are detected, whose centroids are converted to 3D world coordinates. Lights in 3D scene are placed a few centimeters below the ceiling, so that the area around the lights will not be infinitely bright. Luminous intensity distribution of all the lights are set to a uniform distribution of white color.

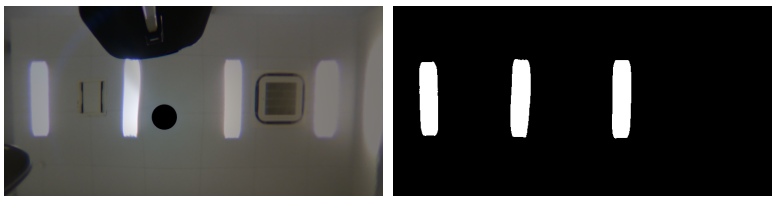

Figure 3.33: Images projected to the ceiling, containing 4 fluorescent lights. Left: Color, Right: binary after threshold.

3.3.5 Patch Extraction from Individual 3D Scan Data

Extracting objects directly from the merged point cloud is difficult because of ghosting error, which duplicates parts of the scene in an irregular way. Fig. 3.34 shows three examples of ghosting. Ghosting can be caused by distortion of the point clouds making it impossible to align them perfectly with rigid transforms. Another cause of the ghosting is alignment error. Unmerged point clouds can have distortion error, but not ghosting. Thus, patch extraction from individual scan data acts as a kind of noise removal method.

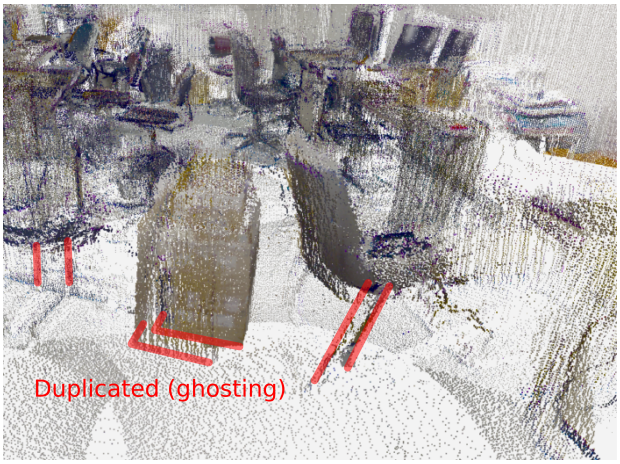

Figure 3.34: Ghosting errors in a point cloud. Legs of chairs, and surface of a box are duplicated.

A patch is a collocated subset of a point cloud, covering a fraction of the surface of a single object inside the room. Patches are extracted from a single scan data in three steps. First, a triangle mesh created from the room frame is deformed non-rigidly to align well with the point cloud. Second, points corresponding to the room boundary or the exterior are removed. Finally, the remaining points are clustered into patches. These three steps are shown in Fig. 3.35.

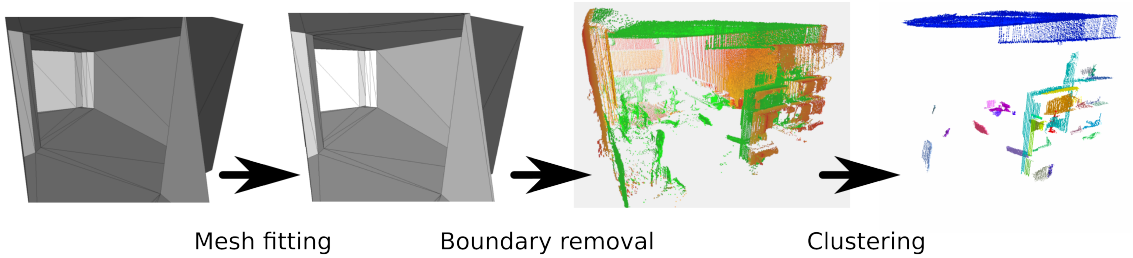

Figure 3.35: Three steps in patch extraction.

The vertices V of the triangle mesh generated from the room frame is deformed by moving vertices, so that J, the squared sum of distance between the points and the mesh is minimized.

$$J(V) = \sum_{p \in P} \min_{q \in mesh(V)} \|p - q\|^2 \tag{3.16}$$

where P is the point cloud, and $mesh(V)$ is the collection of points on the surface of the triangle mesh defined by V. To minimize J efficiently, the derivative $\partial J / \partial V$ needs to be known. Fortunately, the derivative can be approximated by projecting each p to the nearest triangle, and distributing to the three vertices of the triangle using barycentric coordinates. The optimization is accomplished by the gradient descent. Calculation of the nearest triangle for each point in the point cloud is accelerated by using the Axis-Aligned Bounding Box (AABB) tree, implemented in Computational Geometry Algorithms Library (CGAL)[41].

After the triangle mesh is deformed, non-inside points are rejected by simple thresholding on the point-mesh distance. The threshold is conservatively set to 15 cm, which favors false negatives (i.e. removing too much inside points) over false positives (i.e. keeping boundary points).

In the final step, the points are clustered by Euclidean clustering[42], with linking distance of 2 cm. The specific linking distance is the typical distance between neighbor points in the point cloud, which is a product of scanner-object distance and scanner LRF resolution. By rejecting too small clusters with less than a hundred points, outliers far from other clusters can be removed. The remaining clusters are the patches, which are

shown in Fig. 3.36.

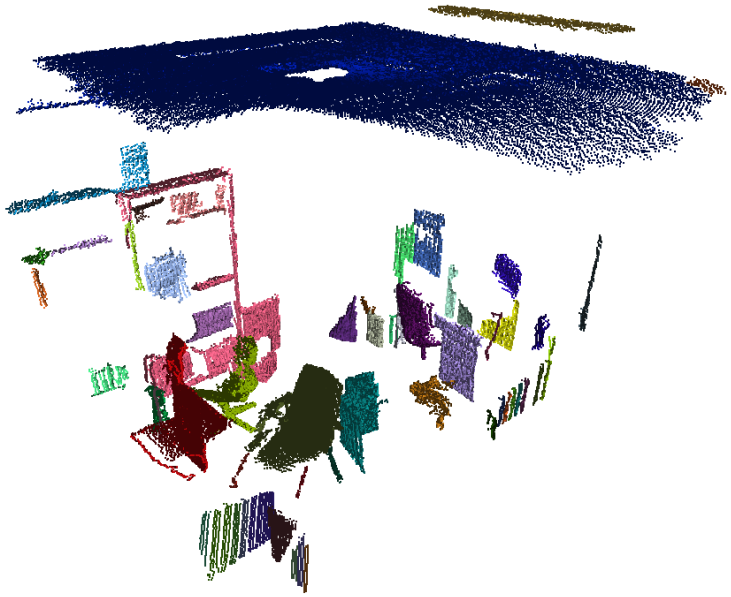

Figure 3.36: Patches from a single scan location, colored by cluster.

Fig. 3.36 is the result of processing a point cloud from a single scan data, which covers approximately half of a room. Two large patches near the ceiling can be observed despite the boundary and exterior rejection in the second step. This is due to mis-estimation of room height due to complex ceiling condition such as beams and recessed ceiling. This kind of error is removed in the physical reasoning described in subsection 3.3.6.

3.3.6 Patch Linking with Physical Stability Reasoning

When earthquake force is not applied to the scene, it is extremely unnatural if objects start moving or toppling abruptly. Such unnaturalness can be prevented by ensuring that recognized objects remain stationary when no disturbance force is acting on them. In a real scene without earthquakes, objects remain stationary because they are supported by the room itself either directly or via other objects. This type of balance must be achieved in the reconstructed scene. Basic approach is to link all the patches form clusters such that each cluster is physically stable when simulated. The clusters obtained by this process become interior objects in subsequent process. An example of patches in shown in Fig. 3.37.

Naive clustering of patches based solely on geometric properties such as inter-cluster distance, would result in unstable or even floating objects. The kind of physical reasoning based on stability can be seen in existing works [3][20][19]. However, one of such works[3] is

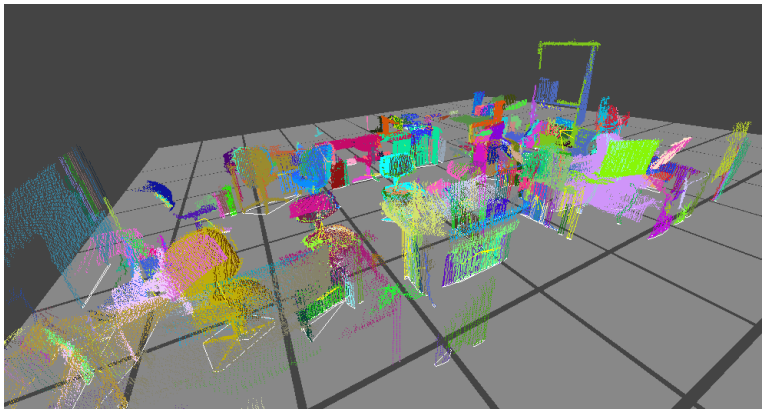

Figure 3.37: All patches from a single room. This particular example has 405 patches in total from 15 scan locations.

limited to cuboid-shaped objects, and the others[20][19] are based on voxel representation and incapable of handling partially overlapping point clouds. In the proposed method, stability of objects are reasoned using support polygons. The idea of support polygon is commonly found in robotics, since it is a static special case of zero-moment point method[43], which is used to control a robot so that it can move on the floor without falling down. A support polygon is defined as a planar convex hull of part of an object touching a supporting surface. When the center of gravity of the object falls inside the support polygon, the object remain stable, as shown in Fig. 3.38.

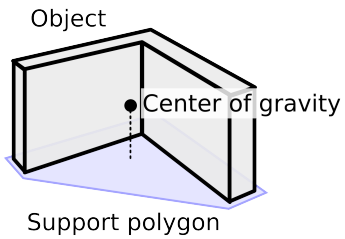

Figure 3.38: An object is stable when its center of gravity falls inside the support polygon.

Using the support polygon, the goal of the linking process can be stated as follows: to aggregate patches into clusters, so that all clusters are stable, while keeping distant patches as separated as possible. To do this, a greedy iterative method is proposed. The pseudocode of the method is shown in **Algorithm 1**.

Basically, the algorithm starts by treating each patch as a cluster containing only one patch, and then links clusters one-by-one to turn unstable clusters into stable clusters. To remove erroneous near-ceiling patches from subsection 3.3.5, a simple heuristic is used in addition to the algorithm. When listing up possible linking actions, an action with large MinDistance and near-ceiling patch can be rejected. This kind of amenability is one of

Algorithm 1 Iterative patch linking algorithm.

```
1: procedure LinkPatches(patches)
2:     clusters ← [[patch] for patch in patches]
3:     loop
4:         stable, unstable ← partition(clusters, IsStable)
5:         actions ← []
6:         for all (c, u) ∈ clusters × unstable do
7:             if IsStable(c ∪ u) and c ≠ u then
8:                 actions ←+ Link(c, u, cost = MinDistance(c, u))
9:             end if
10:        end for
11:        if actions is empty then
12:            return stable
13:        else
14:            bestAction ← min(actions, by = cost)
15:            apply bestAction to clusters
16:        end if
17:    end loop
18: end procedure

19: procedure IsStable(cluster)
20:    points ← all points in cluster
21:    center ← Average(points)
22:    polygon ← ConvexHull(points near floor)
23:    return center falls inside polygon
24: end procedure

25: procedure MinDistance(cluster0, cluster1)
26:    return minimum distance between a point in cluster0 and a point in cluster1
27: end procedure
```

the advantages of the algorithm. A result of the linking is shown in Fig. 3.39.

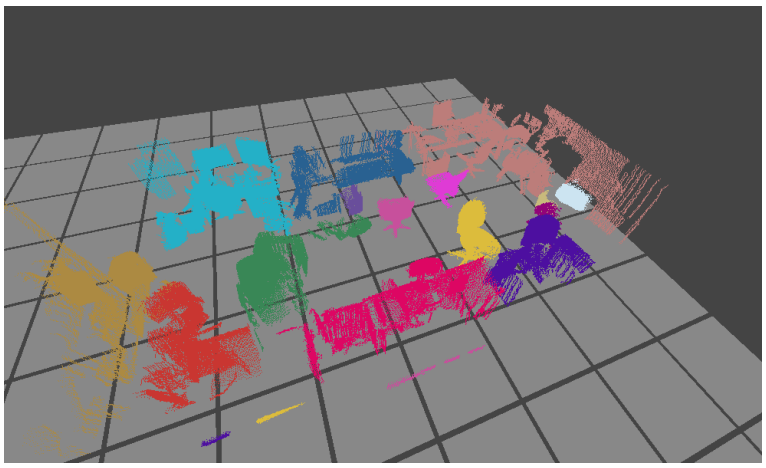

Figure 3.39: Final stable patch clusters.

3.3.7 Textured Mesh and Collision Shape Generation for Boundary

The room itself takes up the largest part in fields of view of the users. Thus, its visual quality is more important than that of individual objects. The appearance of interior boundary is represented by a textured mesh. The texture is calculated by projecting an equirectangular image to the boundary, and then synthesizing occluded part of the floor. Also, to prevent interior objects from escaping from the room in physics simulation, collision shape is calculated in addition to the textured mesh for appearance.

Textured Mesh Generation for Boundary

Appearance model of an interior boundary consists of shape and material. The shape is represented by a triangle mesh, and the material is represented by Bidirectional Reflectance Distribution Function (BRDF). Typically, BRDF is specified by a reflection model (such as Lambertian model), whose parameter distribution over the surface is given by a texture. A texture is a 2D image containing parameters for different points on the 3D surface. Any point on the 3D surface has exactly one corresponding point on the texture, and the parameter at the point is determined by the pixel value in the texture. In this way, a texture can specify a parameter distribution on any 3D surface. By convention, coordinates of the texture is denoted UV, to make it easier to distinguish them from normal XY coordinates. The 3D to 2D mapping is called a UV mapping, and the mapping is expressed by attaching UV coordinates to each vertices in the triangle mesh. Fig. 3.40 shows an example of UV mapping between a triangle mesh of a sphere

and triangles in UV coordinates. UV values of vertices are directly determined by the attached UV coordinates, and UV coordinates of all other points is linearly interpolated.

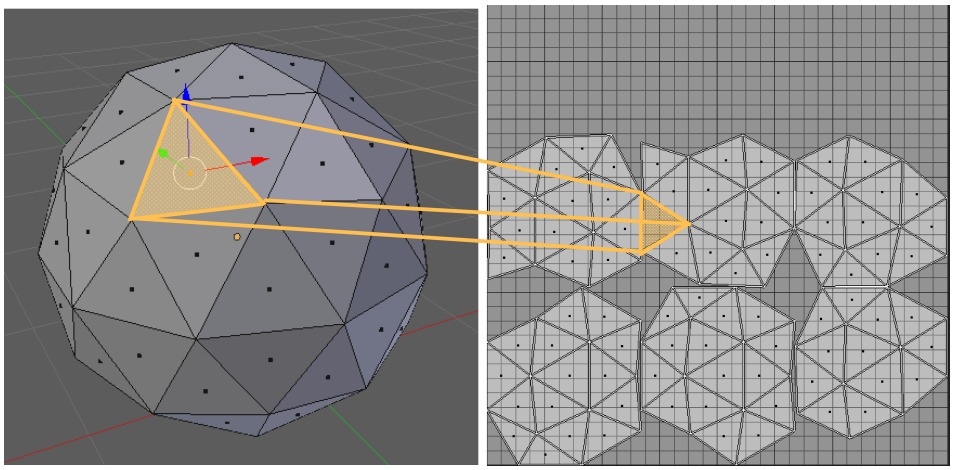

Figure 3.40: A 3d triangle mesh (left), and triangles unwrapped in UV coordinates (right). Orange lines denote UV mapping.

However, since recognition of object materials is a challenging problem, the default reflection model provided by the game engine, which is a certain mixture of diffuse and specular, is used to represent the interior boundary. This reflection model requires only a single texture to specify its color distribution. To generate the texture of a interior boundary, its UV mapping must be calculated first.

Calculating UV mapping, in other words, parameterizing 3D surface, is non-trivial problem for general surfaces. Fortunately, the room boundary model is limited in the sense it have sharp edges and relatively small number of triangles. In the proposed system, a fast ad hoc parametrization method described in Appendix A, is used to generate UV mappings for interior boundary. Alternatively, more sophisticated methods such as least squares conformal map[44] can be used.

After the UV mapping is determined, the texture is generated by projecting a single RGB equirectangular image to a triangle mesh generated from the room frame. A texture generated by projecting an equirectangular image is shown in the left of Fig. 3.41. However, projected texture contains visually unpleasant distorted furnitures. This is especially noticeable for the floor because of its lack of features unlike windows on the wall or light fixtures on the ceiling. Also, in a preliminary evaluation experiment of the system, it was found that users tend to look straight forward, or slightly downward. Hence, a method that can 1. identify the region containing distorted objects on the floor, and 2. synthesize a plausible content to fill the gap, is required. For brevity, the region on the floor containing distorted furnitures is called an invalid region in the remaining of this

section.

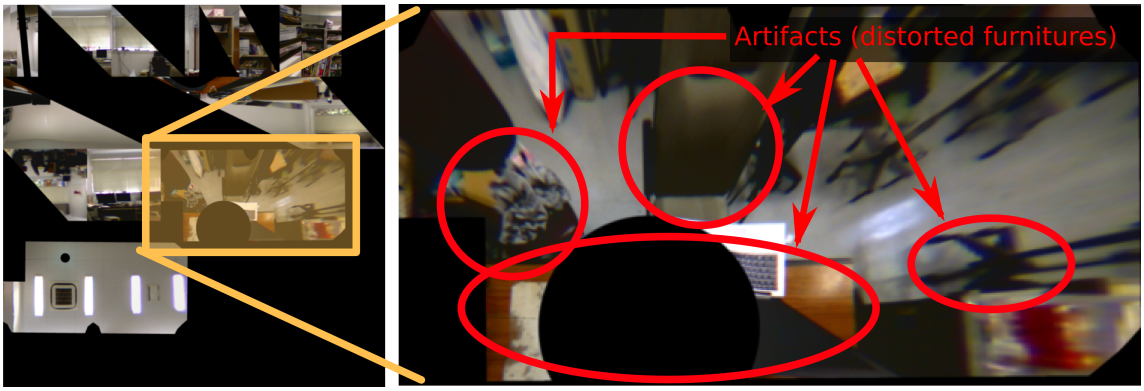

Figure 3.41: The texture of the interior boundary (left) and a zoom of the floor (right). Most of the floor is occupied by distorted objects.

Fig. 3.42 shows the mechanism of how distorted furnitures can appear on the floor texture. When a line of sight from the scanner is blocked by a furniture, surface color of the furniture is projected onto the floor, making the corresponding floor region invalid. This can be detected by comparing distance α between the scanner and the point a on the nearest surface, and distance β between the scanner and the point b on the triangle mesh. When $|\alpha - \beta|$ is large, the corresponding texture region is invalid. Note that $|\alpha - \beta| = \|a - b\|$.

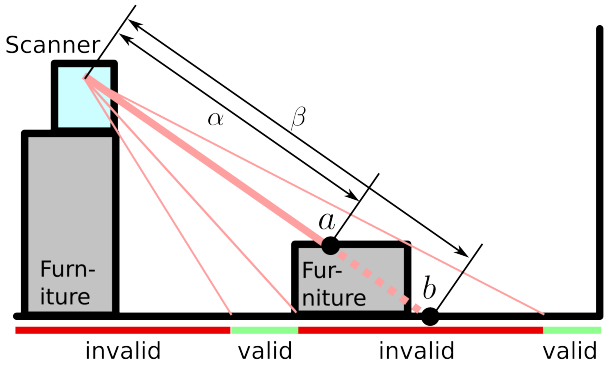

Figure 3.42: Relationship between distance between the scanner and surfaces, and invalid regions.

Fig. 3.43 shows a, b, $\|a-b\|$ in UV space. As expected, invalid regions in Fig. 3.41 have large $\|a - b\|$ on the right in Fig. 3.43. However, there are three factors that could cause error in the detected invalid region. First, RGB image (from RGB camera) and XYZ distance (from LRF) do not align completely due to the offset between the two sensors. Second, thin objects like legs of chairs and cables cannot be detected reliably. Third, shadows cast on the floor by nearby objects cannot be detected by distance comparison.

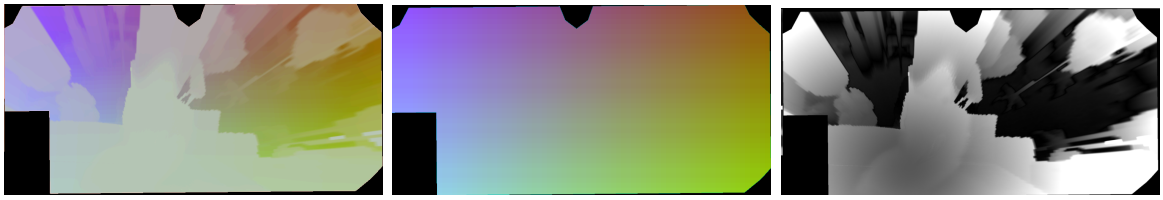

Figure 3.43: From left to right: $a, b, \|a - b\|$. X, Y, Z components of 3D position is represented by R,G,B colors, and distance is represented by brightness.

To overcome errors due to these factors, grabcut[45] is used to refine the mask based on image content. Fig. 3.44 shows the invalid regions before and after refinement.

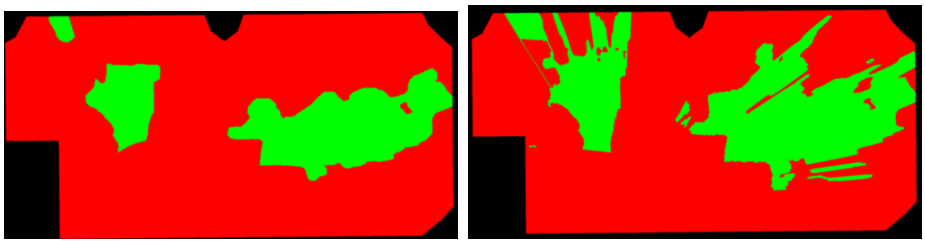

Figure 3.44: Masks used for invalid region identification (red: invalid floor, green: valid floor), before adjustment (left) and after adjustment (right).

The refined invalid region is then removed from the texture, and a plausible content is synthesized by an existing inpainting method[46]. Fig. 3.45 shows the difference between the original texture and the inpainted texture.

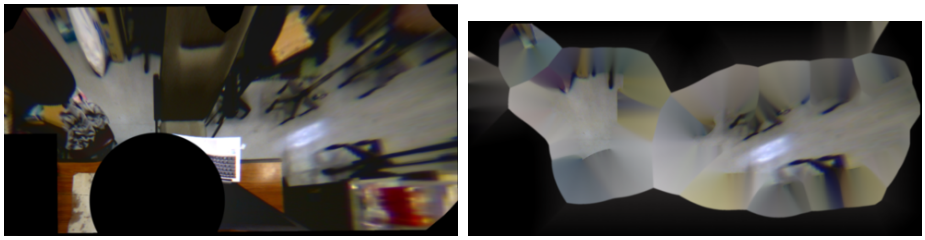

Figure 3.45: Floor parts of textures before invalid region detection (left) and texture after inpainting (right).

Collision Shape Generation for Boundary

Collision shape of the boundary is required to ensure that an avatar (i.e. virtual body of the user) or an interior object remain inside while the simulation is running. Unlike an appearance model which is represented by an arbitrary triangle mesh, collision shape is represented as union of several Oriented Bounding Boxes (OBB), because OBBs are much efficient when computing collisions.

The OBBs can be generated by replacing wall sections in the room frame with cuboids, and adding two OBBs for the floor and the ceiling respectively. The combination of textured triangle mesh and collision shape is shown in Fig. 3.46.

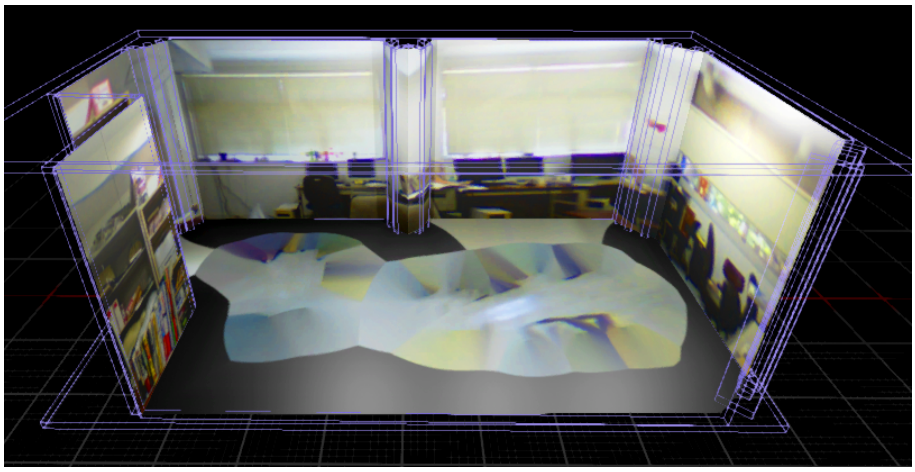

Figure 3.46: Generated textured mesh and collision shape for an interior boundary. Collision OBBs are shown as purple boxes.

3.3.8 Textured Mesh and Collision Shape Generation for Objects

One object is generated per cluster of patches acquired in the patch linking process, described in subsection 3.3.6. Since both a patch and a cluster of patches are a collection of points with normals, clusters are also point clouds. A common method for creating 3D mesh from a point cloud is Poisson reconstruction[47]. However, Poisson reconstruction does not work well with missing or dislocated points. And a robust method such as eigencrust[48] would have difficulty processing the patches. This is because these methods heavily rely on normals, and do not assume the situation where a significant portion of the normals are effectively random because of heavy dislocations.

Our approach is to ignore normals completely, and extract iso-surface created by sum d of kernels placed at the points, like metaballs[49]. However, unlike metaballs, whose aim is to create organic-looking objects, objects in a room are mixtures of round and sharp shapes. So the Gaussian kernel with faster falloff is used in Eq. (3.17) and Eq. (3.18).

$$d(p) = \sum_{0 \leq i < |P|} w_i(p) \tag{3.17}$$

$$w_i(p) = \exp\left(-\frac{\|p_i - p\|^2}{\sigma^2}\right) \tag{3.18}$$

where P is the points in a patch cluster, and p_i is the i-th point in it. σ is the standard deviation of the kernel, set to 2 cm after error model of the LRF. To increase the efficiency of the computation, w_i is cut off at 3σ. For given p, only the points within 3σ radius need to be considered, and such neighbor points can be efficiently queried by using a kd-tree[50].

A triangle mesh is extracted from the implicit iso-surface of d, by using an existing method[51]. The normals of the vertices in the triangle mesh is calculated by $-\frac{\nabla d}{\|\nabla d\|}$. Colors of the mesh can be calculated by the weighted average using functions w_i and the point colors. However, describing colors using solely vertices would require too many vertices, which will decrease framerate of the VR content. Thus, a textured mesh is used in a similar way as the interior boundary. First, UV mapping of the mesh is calculated. By using the inverse of UV mapping, 3D positions of each pixel in the texture can be determined. From these positions, pixels are colored by weighted average color of points similar to Eq. (3.17). Fig. 3.47 shows the triangle mesh without and with texture.

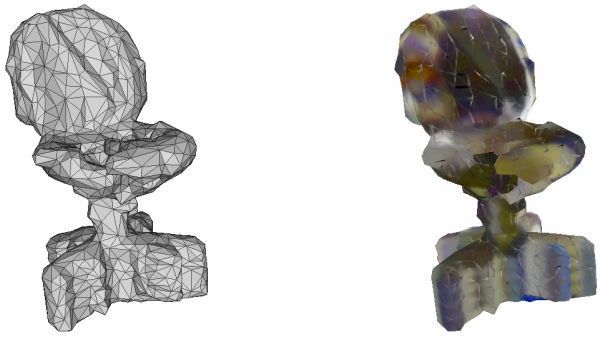

Figure 3.47: An example of generated textured object for an object. (left: before texture generation, right: after texture generation)

Similar to an interior boundary, interior objects also requires collision shapes. The collision shape of an object can be calculated as a union of AABBs, one AABB from each patch of the cluster. The collision shape is shown in purple boxes in Fig. 3.48.

The calculated collision shape is larger than the actual shape, and it also has redundant AABBs such as one completely contained in another AABB. The former could cause a stability issue by making objects overlap, and the latter could cause performance issue by having unnecessarily many AABBs. However, at the granularity of objects in the proposed system, these issues do not arise in practice.

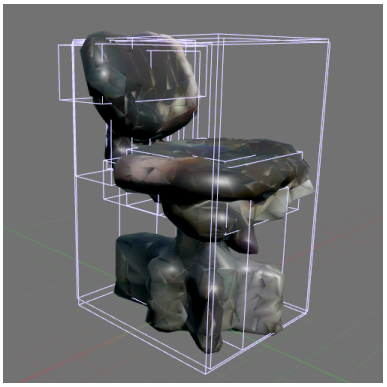

Figure 3.48: An example of textured mesh and collision shape generated for an interior object.

3.3.9 Indoor Environment Reconstruction Results and Analysis

In this section, reconstructions of a few indoor scenes are shown, and several failure modes and possible future improvements are discussed.

To examine the capability of the reconstruction method, four indoor environments were scanned with the system. Table 3.4 shows the descriptions of the rooms. Note that the numbers in the column "ceiling lights" only counts ceiling lights that are turned on. Scan locations within each room is chosen so that most objects larger than 50 cm can be covered from multiple directions when possible.

Table 3.4: Scanned indoor scenes and their descriptions.

ID	Ceiling lights	Approx. size	Description
un1	long ×4	7m × 4m	University office room used by 6 students.
re1	circular ×1	3.5m × 2.5m	Residential storage room.
re2	0 (sunlit)	3.5m × 2.5m	Residential bedroom.
re3	circular ×2	6m × 3m	Residential dining and kitchen, connected in L-shape.

The scan data were processed by the proposed indoor reconstruction method, on a PC with a Intel Core i5-3570K CPU and 32 GB RAM. Table 3.5 shows number of scan locations, runtime of reconstruction and number of detected objects. Table 3.6 shows processing times divided in to several processes. Manual scanning took 4 min per location on average, when operated by a trained person. Fig. 3.49 shows the comparisons of real photos of the scenes and images of the reconstructed scenes.

About half of the run times are taken up by alignment. The other remaining run times mainly consists of patch extraction time and object extraction time, which are roughly proportional to the number of scan locations and objects, respectively. Number of scan locations tend to be larger in a room with many objects away from walls, because such object layout causes a large amount of occlusions and occluded regions need to covered

Table 3.5: Reconstruction results and time to process the scenes.

Scene ID	#location	Reconstruction time	#lights	#objects
un1	15	37.2 min	3	14
re1	7	14.2 min	1	3
re2	5	6.3 min	0	3
re3	7	15.6 min	12	6

Table 3.6: Processing time of reconstruction broken up in several parts.

Scene ID	Alignment	Patch extraction	Object reconstruction	Others
un1	18.3 min	6.3 min	6.0 min	6.6 min
re1	5.4 min	1.8 min	3.0 min	4.0 min
re2	3.2 min	0.7 min	0.3 min	2.1 min
re3	5.5 min	2.4 min	3.0 min	4.7 min

by scanning from other locations.

Since the longest run time of the reconstruction process is under 40 min and a more performance-conscious implementation would be several times faster, it can be expected that such an implementation of the system could reconstruct a scene in a few minutes. This makes it viable to use the proposed system to generate contents on the spot of scanning and/or education.

One light is missed in reconstruction of un1. The missed light does not have significant impact since the room contains 4 lights, but it would be problematic in a smaller room with fewer lights. Conversely, 12 lights are detected in re3 where there are only 2 real lights. This is caused by bright regions on the walls, as shown in the left of Fig. 3.50. In re2 whose only light source is the sunlight through the curtains, no lights were detected because the system is designed only to handle near-ceiling lights.

A missed light in un1 and false lights in re3 are caused by the same problem. Fig. 3.50 shows bright regions that the proposed method might mistake for lights. Most of the bright regions are reflections by nearby walls. In RGB images, some of them are as bright as real lights, making it difficult to distinguish false and true lights via simple thresholding. To resolve this ambiguity, information such as typical shape of light fixtures, as well as the scene geometry including detailed shape near the ceiling, are required. Although ignored in the proposed system, an ability to estimate the intensity distribution of a light source (e.g. point light or directional light) would be useful in creating realistic shadows. In summary, to detect more types of lights more reliably, some type of reasoning on light transport would be required. Although the principles of light transport is very well understood[52], reasoning based on it is an open problem.

Detected number of objects were vastly less than the true number of objects. As a

Figure 3.49: Top row: photos of each scene. Bottom row: reconstruction, rendered by the runtime subsystem. From left to right: un1, re1, re2, and re3.

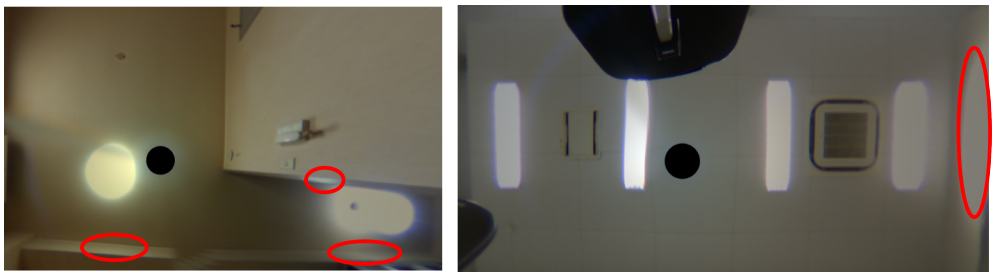

Figure 3.50: Ceiling images of re3 (left) and un1 (right). Red circles denote a bright region with no light fixture.

reference, Fig. 3.51 shows 20 manually counted objects in a small visible portion of un1. In addition to Fig. 3.51, stacks of books, dishes or clothes are common occurrences in an indoor environment. Considering these two facts, a typical room could easily contain more than one hundred visible objects. To analyze missing objects more closely, individual detected objects in un1 is shown in Fig. 3.52.

The smaller objects (1-6, and 10) seemingly correspond to true object one-to-one, although parts of some of them are missing due to occlusions. However, larger objects (7-9 and 11-14) consist of multiple true objects. Especially the bookshelf (14), would contain several dozens of books inside it in the real scene, which are not reflected in the reconstruction. Also, fusion of nearby pairs of chairs and desks can be observed in 8, 12 and 13. To resolve these fusions, prior information about appearance of typical indoor objects would be needed.

Figure 3.51: 20 objects manually counted in a part of an RGB image.

3.4 Collision Sound Extraction from Recorded Sound Clips

As discussed in subsection 3.1.1, collisions and ground-shaking sounds are simulated by the proposed system. In this section, existing methods of collision sound synthesis are reviewed, and a slightly modified version of the method[27] to accomodate lack of material information, is described.

Existing physically-based sound synthesis methods can produce modal sounds[53][54], cloth sounds[55], and more generic impact sounds[27]. However, the method that purely relies on modal sounds without any supply of recorded sounds [53] produces less realistic sounds, compared to a method utilizing such external information [54]. For more complex objects, such as cloth or multi-part objects, which often deviate from simple physical models, manually annotated recorded sounds are required to synthesize high-fidelity sounds[55][27]. Since there are many collision events in an earthquake, a notion of using a long recorded sound of a real earthquake, instead of adding individual collision sounds, is tempting. Such a method would be computationally efficient and would allow a simple implementation. However, it is suggested[56] that human can perceive temporal difference of 75 ms between visual and auditory stimuli in presence of two close visual stimuli, and multiple temporally close stimuli might increase the sensitivity. This invalidates the notion of using a long recorded sound.

In this research, a method similar to the existing method[27] is proposed. However, there are two major differences in the two methods. First, materials of objects are unavailable to the proposed system unlike in the existing method. Thus, in the proposed method, sounds are chosen randomly from a set of several types of collision sounds, and modal analysis of each sounds are omitted. Second, the proposed method automates some part of recorded collision sound collection process by providing a heuristics to split collision in a short recorded clips of several seconds. The method consists of sound splitting and play-

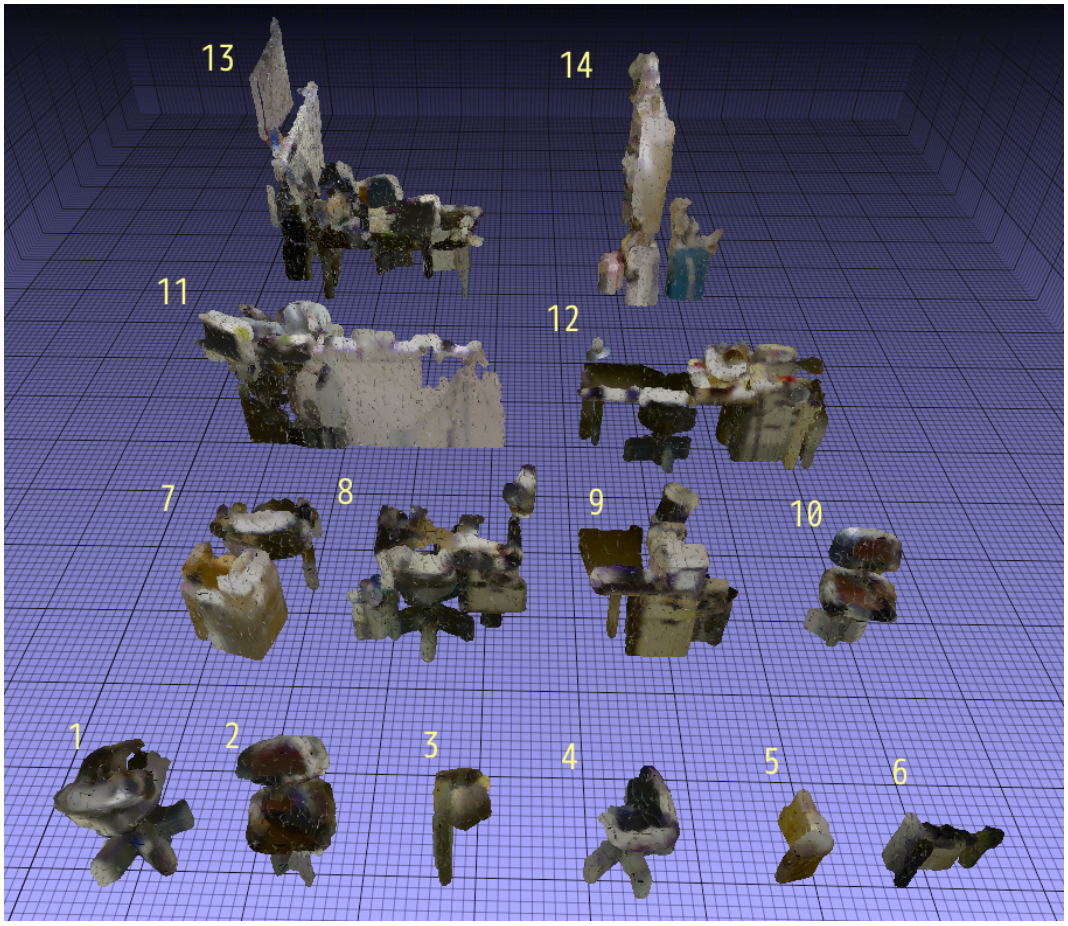

Figure 3.52: 14 objects in the reconstruction of `un1`.

back, and this section describes the former as a part of the content generation subsystem, and the latter is described in subsection 3.6.2 as a part of the runtime subsystem.

First, short recorded sound clips containing collisions are collected. Sounds of this type are publicly available in abundance from websites like freesound[57], where various short sounds uploaded by anonymous users are hosted for use as special effects in films, games, or music. The sounds clips are easily found by searching with combinations of a keyword for collision such as “hit”, “drop”, “collision” and a keyword for material or object such as “wooden”, “box”, “plate”. It might be possible to automate this process in the future, which would enable retrieving sounds by object names, for instance. But in this research, several sounds clips of this type are manually collected.

This type of sounds contain multiple consecutive collisions to include random variations, and silent gaps before and after the actual collisions. As suggested by Roseboom et al.[56], these individual collisions should be extracted within error of 75 ms. Since this splitting process is time-consuming, simple heuristics is proposed to automatically detect peaks of collisions and split the sound clips into individual collision sounds.

Let $s(t)$ be an input sound amplitude which takes a value in $[-1, 1]$. Smoothed sound

level L is calculated as follows.

$$L = |s| * b \tag{3.19}$$

$$b(t) = \begin{cases} 1/\alpha & \text{if } 0 \leq t \leq \alpha \\ 0 & \text{otherwise} \end{cases} \tag{3.20}$$

where $\alpha = 10\,\text{ms}$ is a constant specifying the width of box window b. It is specified to be shorter than the width of peaks in the sounds by a small amount.

Regions where $L > 0.5$ can be considered to contain peaks of collisions. From the peaks, L is searched forward and backward until silent regions ($L < 0.01$, which is inaudible). By this procedure, individual collision sounds can be obtained. Examples are shown in Fig. 3.53.

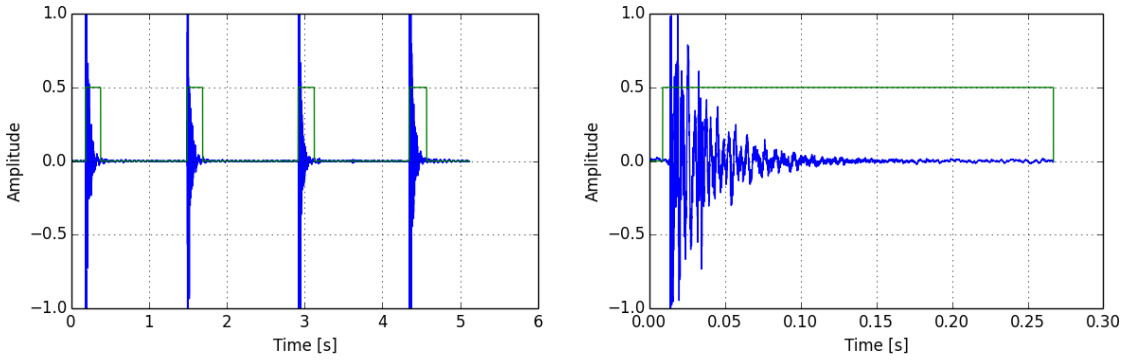

Figure 3.53: Example results of collision sound extraction. Green boxes denote extracted parts of the sounds.

3.5 Earthquake Acceleration Filtering and Ground Shaking Sound Generation

Motion of objects and ground-shaking sound are heavily affected by input earthquake acceleration. From a sequence of 3D acceleration measured at a single point, a filtered acceleration suitable for realtime rigid-body physics simulation, and a ground shaking sound are calculated.

3.5.1 Temporally Changing Inertial Force Acting on Objects

As discussed in subsection 3.1.1, all objects including buildings are approximated by rigid bodies, and motion of interior objects are calculated by using inertial force instead

of moving the room floor. Using the inertial force has advantages over the moving floor, as it allows static lighting computation and increases simulation stability.

The acceleration used by the physics simulator is almost identical to the raw acceleration measurement data. The east-west, north-south, up-down (EW, NS, UD) components of the measurement data are mapped to X, Y, Z axes respectively, since rotation around Z (or UD) axis would not affect experience of the users. Then, the measurement data is resampled to the framerate of the runtime system, with optional low-pass filtering to prevent aliasing when the framerate is lower than the input sampling rate.

3.5.2 Ground Shaking Sound Generation

The exact mechanism of so-called *ground-shaking* sounds is not well understood, but it is safe to assume the sound is caused presumably by vibration of building or ground, which is heavily correlated to the acceleration. In an existing earthquake sound synthesis work[26], it is speculated that seismograph data do not contain any audible sound, and the authors proceed to add an pitch-shifted sound artificially. Although not justified in the work, it is highly probable that some kind of non-linear phenomena is emitting such higher frequency harmonic components in real earthquakes.

However, there are two reasons to believe that such an artificial pitch shifting is not strictly necessary. First, sampling frequency of modern seismometers can reach 100 Hz[24] whose Nyquist frequency is 50 Hz, which is above the 20 Hz, the lower threshold of the audible sound. Second, there is a possibility that infrasound (sound below the threshold) is actually audible to some extent contrary to the popular belief[58].

Thus, we take a more physically based approach of not adding artificial harmonic components. By assuming that sound amplitude is proportional to the displacement of the building or the ground, and all directional components equally contributes to the final sound, the sound earthquake $s(t)$ can be expressed like Eq. (3.21).

$$s(t) \propto \sum_{1 \leq i \leq 3} \int_0^t \int_0^{t'} a_i(t'') \mathrm{d}t'' \mathrm{d}t' \tag{3.21}$$

where $a_i(t)$ is a acceleration component in the i-th axis.

In the frequency domain, the double integration is equivalent to multiplying with gain $G(\omega) = -1/\omega^2$ where ω is an angular frequency. This process effectively boosts the lower frequency components. After this, a high-pass filter of 5 Hz is applied to maximize dynamic range by removing large inaudible component.

3.6 VR Runtime System for Earthquake Experience Presentation

The data generated by the proposed methods need to be aggregated, simulated, and presented to the users. The runtime system achieves these with a combination of hardware and software.

The hardware consists of an HMD and a headphone, with a camera to track the head position of users. An example of the devices are shown in Fig. 3.54. However, it is infeasible to treat the software as a single monolithic program due to its complexity. Thus, a game engine, in conjunction with the hardware, is treated as a conceptual VR platform that takes a 3D content of a scene as input, and provides users with experience of the scene in VR, with appropriate physics simulations.

Figure 3.54: A headphone, an HMD and a tracking camera.

Collision sound synthesis is one of the methods to be implemented on the platform, and is specific to the proposed system. The method plays back individual collision sounds in synchrony to collisions in physics simulation.

3.6.1 VR and Physics Simulation Platform

Of the requirements of the runtime system shown in section 3.1, "cause no VR sickness" requires a constantly high framerate of the simulation and rendering. In practice, this requires a tremendous amount of high quality code, optimized to a specific VR device and graphics processors. To satisfy these constraints, the proposed runtime system uses a game engine to provide these basic (but large amount of) functionalities.

A typical game engine provides an editor, in which artists can create contents. It also have an ability to create a package, which is a unmodifiable set of files users execute to

the experience, typically a game. Both the editor and the packages runs on a collection of software libraries that enable all kinds of computation necessary for games, such as physics simulation, animation, renderers, and user interfaces.

Fig. 3.55 shows an image of an artist-created room, rendered in realtime by a game engine with focus on high realness of graphics. It demonstrates that a modern game engine can achieve almost perfect photorealism in realtime, when the proper contents are available. Thus, the approach of the proposed system, which is to rely on the realtime rendering and simulation, can be said to be a viable approach.

Figure 3.55: An artist-created example of indoor scene running in a state-of-the-art game engine.

3.6.2 Collision Sound Playback

In section 3.4, individual collision sounds are created by combination of manual collection and automatic splitting. The runtime subsystem play them back in synchrony to collisions events in rigid-body physics simulation, to make collision sounds synchronized to collisions visually observable to the users.

However, there are two problems preventing the seemingly simple solution of playing back collision sounds. First, types or materials of objects are unavailable to the system, as discussed in section 3.4. Second, the collision detection of a realtime rigid-body physics simulator could be erratic with many false positives, especially when there is a tiny force acting on the objects. A plausible mechanism of these false positives are as follows. Typically, objects in a rigid-body simulator has special flags to store whether an object is completely stationary relative to other objects and to increase simulation stability without

decreasing simulation timestep. When a tiny force (like a weak earthquake) is applied to an object, such an optimization is invalidated, and objects could be slightly unstable.

Within these constraints, sounds are played with the simple rules shown in **Algorithm 2**, where *sounds* is the collection of pre-processed collision sounds from section 3.4. In [27], strengths of modes (or frequencies) in the sounds are randomized. **Algorithm 2** randomizes the pitch of whole sound without modal analysis, which results in faster computation.

Algorithm 2 Collision sound play back rule, which is called on every collision event.

```
1: procedure OnCollision(vel0, vel1)
2:     rvel ← ‖vel0 − vel1‖
3:     if rvel < v_θ then
4:         do nothing
5:     end if
6:     if with probability 1 − p_θ then
7:         do nothing
8:     end if
9:     sound ← ChooseRandom(sounds)
10:    pitch ← RandomBetween(0.5, 2)
11:    volume ← k_v rvel
12:    play sound at pitch, volume
13: end procedure
```

Parameters v_θ, p_θ is empirically set to 10 cm/s and 0.03 respectively. The eventual relation between the volume and the relative velocity is shown in Fig. 3.56. The upper limit of the volume is determined by the sound interface hardware and safety constraint. k_v is an implementation-dependent parameter, which is manually tuned to achieve a natural balance with the earthquake shaking sound.

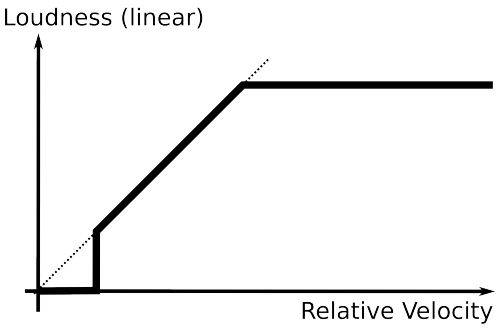

Figure 3.56: Loudness of collision sounds vs. relative velocity between the two colliding objects.

Chapter 4

Evaluation of the VR Earthquake Experience System

In this chapter, the goal of the proposed system is reviewed, and concrete evaluation objectives are derived from it. To achieve these objectives, an experimental evaluation protocol is devised and the results of the experiment are analyzed.

4.1 Purpose of the Experimental Evaluation

The ultimate purpose of the system is to cause fear towards earthquake, and section 3.1 discusses its requirements. Fig. 4.1 shows the requirements, along with whether each requirement is already shown to be satisfied or needs subjective evaluation. Chapter 3 shows that automatic content generation is possible, but whether users can perceive difference in simulated seismic intensity scales needs to be evaluated. All other requirements depend heavily on perception by users, and need to be evaluated experimentally.

Thus, the purpose of the experimental evaluation is to measure perceived qualities affecting these requirements, namely: realness, memorability, seismic intensity scales, fear, and VR sickness.

4.2 Experimental Evaluation Method

The evaluation experiment was conducted with 18 participants. Each participant underwent three 2-minute earthquake experience sessions, and answered questionnaires and were interviewed about the experience.

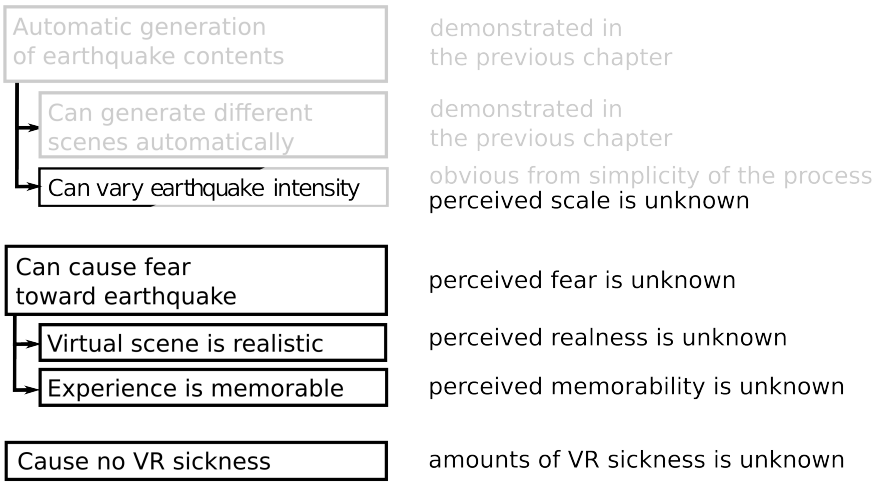

Figure 4.1: Requirements of the system.

4.2.1 Experiment Conditions

VR Hardware Setup

The HMD used in the experiment is Oculus Rift DK2, which is a state-of-the-art HMD at a consumer price. Its specification is shown in Table 4.1, and a photograph of it is shown in Fig. 4.2. As shown in the specification, the HMD also provides an external tracking camera that can track the head position and rotation. Combined with stereoscopic rendering, the HMD was deemed immersive enough, which is a prerequisite for the proposed system. The tracking information is used by the game engine to produce different graphics based on the head pose.

Table 4.1: Specification of the HMD and the tracking system.

Product Name	Oculus VR, Oculus Rift Development Kit 2
Resolution	960 × 1080 per eye
Refresh Rate	75 Hz, 72 Hz, 60 Hz
Persistence	2 ms, 3 ms, full
Viewing Optics	100° Field of View (nominal)
Internal Tracking Sensors	Gyroscope, Accelerometer, Magnetometer
Internal Tracking Update Rate	1000 Hz
Positional Tracking Sensors	Near Infrared CMOS Sensor
Positional Tracking Update Rate	60 Hz
Weight	434 g (without cable)

However, the HMD does not contain headphones, so a separate headphone was used. The experiment was conducted in a room without sound shielding, so external sound can be heard by the participants. Fans of a controller PC also emit sound noise. To prevent these sounds from masking sounds of the system, a noise-canceling headphone was used

Figure 4.2: Photo of the HMD.

in the experiment. The specification of the headphone is shown in Table 4.2.

Table 4.2: Specification of the noise-canceling headphone.

Product Name	audio-technica, ATH-ANC9
Type	Active noise-canceling
Driver Diameter	40 mm
Frequency Response	10 - 25,000 Hz
Active Noise Reduction	Up to 30 dB
Sensitivity	100 dB
Impedance	100 ohms
Battery	AAA
Weight	220 g (without cable and battery)

Finally, a PC to run the game engine and to drive the HMD and the headphone is required. The specification of the PC used is shown in Table 4.3.

Table 4.3: Specification of the controller PC.

CPU	Intel Core i7-3770
RAM	16 GB
GPU	NVIDIA GTX780Ti

The PC is capable of displaying the scene in the HMD with 75 fps (frames per second), which is the minimum required framerate to reduce VR sickness as recommended by the HMD vendor.

Input Data of the Reconstruction System

A small room in Kyoto university (`un1` in subsection 3.3.9) was used as the input scene. The room contains office-like furnitures, but it contains a significant amount of clutter

because the room is regularly used by students of a lab. The room was chosen because of the following two reasons. First, it is small and cuboid shaped, so the participants would be able to see everything in the field of view relatively clearly, despite the limited resolution of the HMD. Second, the room contains more than 10 objects, and perceived realness of collisions motion and sound would reflect the property of the proposed method and not an arbitrary object.

To evaluate whether participants can perceive different simulated seismic scales as different, a wide variety of measurements with different seismic intensity scales are necessary as input data. An earthquake with a large magnitude would enable such a varied measurements, because the maximum seismic intensity scale would be large and many measurements of lower scales would be available at different locations. Thus, measurements of the Tohoku earthquake (2011) at three different stations of K-NET[24], a Japan-wide sensor network for seismic activity monitoring, were used as the input.

The number of seismic intensity scales needs to be minimized to make counterbalancing feasible, because the perceived fear and seismic intensity scales might be affected by the order they are presented. Also, duration of each earthquake simulation, especially that of barely perceptible tremors, should be short. Because otherwise, the simulation would needlessly make participants VR sick and bored, thus introducing a noise to the experiment.

With these constraints in mind, three measurements were chosen, and three two-minute earthquake sessions (S1, S2, S3) are created from the measurements by selecting measurements in the interval (2011/03/11 14:46:20 - 14:48:20), which contains clear peaks in acceleration. Table 4.4 shows the measurement station names and the maximum seismic intensity scales during the session.

Table 4.4: Used seismographs and their maximum seismic intensity scales.

Session ID	Measurement station ID	Maximum seismic intensity scale
S1	MYG004	6.6
S2	GNM011	5.0
S3	AKT006	4.0

The three seismic intensity scales $4.0, 5.0, 6.6$ were chosen because they would uniformly cover the whole perceptible range of seismic intensity scales for most users. The upper limit of 6.6 is simply the maximum available seismic intensity scale of the chosen earthquake. The lower limit is set to 4.0 because an earthquake with a scale of 3.0 was imperceptible to most users in a preliminary experiment.

4.2.2 Experimental Protocol

Participants were recruited from students of Kyoto University, and 18 undergraduate students (13 males) participated the experiment. The participants have no prior experience with the proposed earthquake experience system. Also, participants were not informed that the earthquake experience was automatically generated by 3D scanning.

Although no interaction among participants were intended, the experiment was conducted in groups of 3 or 4 participants to make the process efficient. The protocol is shown in Fig. 4.3. To prevent a participant from knowing about the system before experience, two separate rooms, a waiting room and an experiment room, were used. The participants were asked to be silent in the waiting room, and the system is only used in the experiment room, for only one participant at a time.

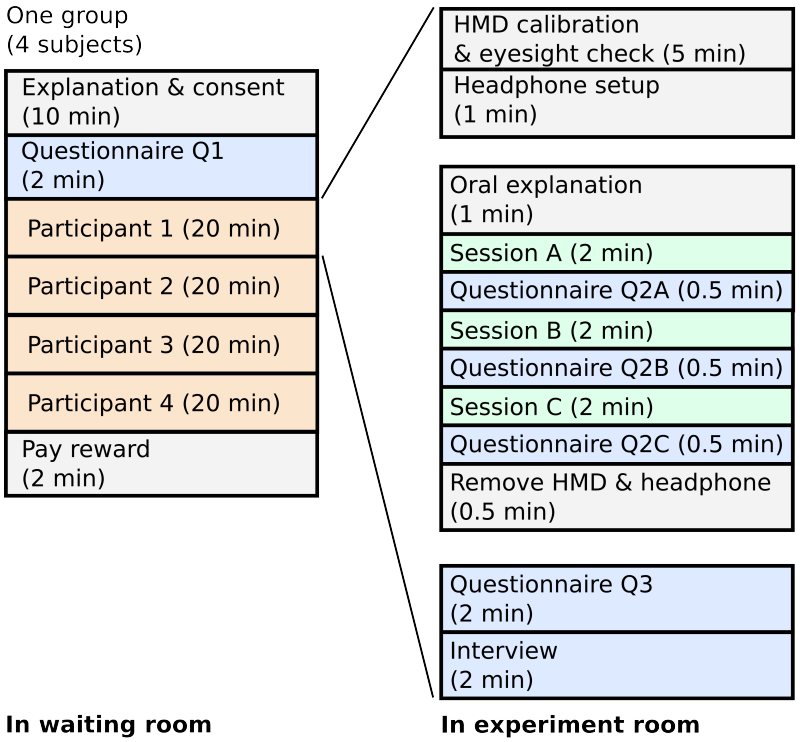

Figure 4.3: Experimental protocol of a single group (1.5 hr).

A group of participants was gathered in a waiting room, where an experimenter explains the purpose and the simplified schedule without any mention on the right part of Fig. 4.3. They were told to answer questionnaire Q1 containing only one question, which asks the maximum seismic intensity scale of their past earthquake experiences. At the same time, the participants were told to consult a table describing seismic intensity scales created by JMA[59], which is shown in Table 4.5. They were also told to casually remember the table so that they will be able to remember it only by looking at illustrations when

wearing a HMD. Since questionnaire Q1 requires answering a seismic intensity scale, it encourages the participants to actually study the table.

Table 4.5: Description of seismic intensity scales used in the experiment[59].

Scale	Human perception and reaction	Indoor situation
0	Imperceptible to people, but recorded by seismometers.	-
1	Felt slightly by some people keeping quiet in buildings.	-
2	Felt by many people keeping quiet in buildings. Some people may be awoken.	Hanging objects such as lamps swing slightly.
3	Felt by most people in buildings. Felt by some people walking. Many people are awoken.	Dishes in cupboards may rattle.
4	Most people are startled. Felt by most people walking. Most people are awoken.	Hanging objects such as lamps swing significantly, and dishes in cupboards rattle. Unstable ornaments may fall.
5 lower	Many people are frightened and feel the need to hold onto something stable.	Hanging objects such as lamps swing violently. Dishes in cupboards and items on bookshelves may fall. Many unstable ornaments fall. Unsecured furniture may move, and unstable furniture may topple over.
5 upper	Many people find it hard to move; walking is difficult without holding onto something stable.	Dishes in cupboards and items on bookshelves are more likely to fall. TVs may fall from their stands, and unsecured furniture may topple over.
6 lower	It is difficult to remain standing.	Many unsecured furniture moves and may topple over. Doors may become wedged shut.
6 upper	It is impossible to remain standing or move without crawling. People may be thrown through the air.	Most unsecured furniture moves, and is more likely to topple over.
7		Most unsecured furniture moves and topples over, or may even be thrown through the air.

After the explanation and questionnaire Q1, the participants were processed sequentially in an separate experiment room, one at a time. The participants underwent the VR hardware setup, the earthquake experience sessions. After each earthquake session, they were presented with questionnaires Q2 with regards to the experiences. Finally, they were presented with questionnaire Q3 and interviewed regarding the overall experience. The hardware setup and a participant in the experiment room are shown in Fig. 4.4 and Fig. 4.5.

The HMD was worn with corrective eyewear, when physically possible. When it was

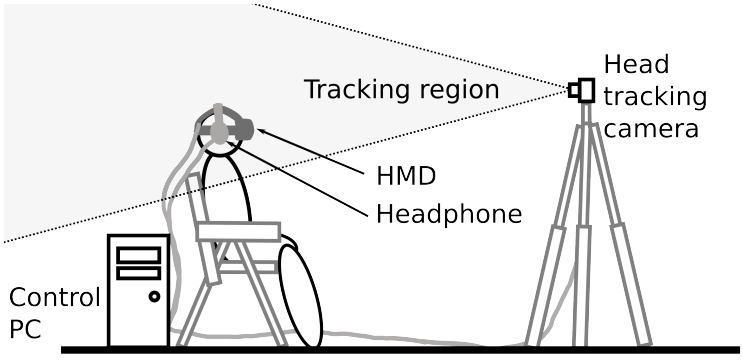

Figure 4.4: Side-view of the participant and apparatus.

Figure 4.5: A photo of the experiment. The person in the middle is a participant, and the other person is an experimenter.

not possible (e.g. too large glasses), the uncorrected vision of the participant was assessed based on their prescriptive value and the experiment proceeded without vision correction if deemed sufficient. In a rare occasion of a participant with too low uncorrected vision wearing too large glasses, the lenses of the HMD were changed to corrective ones (which are included as accessories of the HMD). In order to confirm that the eye correction worked, the participants were asked whether letters in a calibration instruction (English text) can be clearly seen. The text is shown in a virtual screen, and the height of "m" letter is 5 mm and the virtual screen is 60 cm away from the eyes. All 18 participants passed this test after slight adjustment of the HMD belts. The HMD was then calibrated to account for different interpupillary distance by using an utility program provided with the HMD.

After the HMD was corrected, the headphone was put on the participant, and an

experimenter informed the participant that there would be three 2-minute sessions, and perceived seismic intensity scale and fear would be asked after each session. The end of each session is marked with an alarm sound. The existence of the alarms was also informed in the explanation.

The three earthquake experience sessions (S1, S2, S3) were sequenced in a permutation of 3 session shown in Table 4.4. In order to compensate for the order effect, all 6 permutations are equally distributed over 18 participants, making it completely counterbalanced. Since the participants were wearing HMD throughout the sessions, questionnaire Q2 was orally presented, accompanied by visual description of seismic intensity scales and listing of 5-scale fear on the virtual screen. A scene with a virtual floating screen is shown in Fig. 4.6.

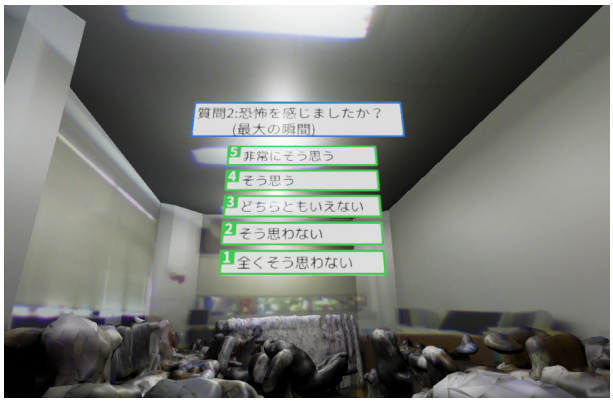

Figure 4.6: A virtual floating screen asking the level of fear.

After the earthquake experience sessions, the participants were asked to fill in questionnaire Q3 which contains the four realness questions, the memorability question, the VR sickness question and an optional free comment question. The interview was conducted immediately after completion of Q3, where the participant was asked reasons for their answers in the realness questions.

4.2.3 Questionnaire and Interview Items

Since all participants were residents of Japan, all questionnaires and interviews were conducted in Japanese. Only English translations are shown in the following, and the original Japanese version is shown in Appendix B.

There are six element-sensory mode pairs in Fig. 3.5, and in principle, measurement of realness can be decomposed to six questions, one for each element. However, some of such questions would be harder to understand correctly for participants. Thus, similar questions are merged together, resulting in 4 questions in total. The pairs and the questions are shown in Table 4.6.

Table 4.6: Questions on elements of realness in questionnaire Q3.

Element	Sensory mode	Corresponding question
Visual	Objects	When the scene is not shaking, do you think the scene looked photorealistic?
Visual	Boundary	
Visual	Object movement	Do you think the movement of the objects inside the room were similar to that of real earthquakes?
Visual	Avatar shaking	Did you think the movement of the room itself were similar to that of real earthquakes?
Auditory	Object collision	Do you think the sounds were similar to that of real earthquakes?
Auditory	Ground shaking	

Visual-objects and visual-boundary were merged because they would be hard to distinguish for participants, especially when the scene is static. For instance, it is uncertain whether a participant would categorize a distorted object on the wall texture as an artifact of the interior boundary, or an object. Similarly, auditory-collision and auditory-shaking were merged because sounds are harder to distinguish when participants have no idea on its constituents.

Since object collision (both visual and auditory) or earthquake sounds are not frequently encountered in real life, these three questions were phrased to ask similarity to earthquakes, instead of realness.

The participants were asked to answer each question within 7 levels of agreements: “strongly agree”, “agree”, “somewhat agree”, “neutral”, “somewhat disagree”, “disagree”, and “strongly disagree”.

To acquire more detailed feedback for improvement, a free comment section was included in the questionnaire, and interviews following the questionnaire were also conducted. In the interviews, participants were asked reasons of their scoring for each of the four realness questions.

The other items measured are VR sickness, memorability of the experience, perceived fear, and perceived seismic intensity scales, which correspond to the requirements other than high realness. These questions are shown in Table 4.7. Since perceived fear and seismic intensity scales will be dependent on strength of simulated earthquakes, they were asked during each earthquake session.

Although it was expected that almost all participants have prior earthquake experience and can reason about realness of the simulation, a question asking the maximum seismic intensity scale they experienced was included in the questionnaire Q1 to validate this assumption.

Table 4.7: Questions about perceived seismic intensity scale, fear, memorability, and VR sickness, in questionnaire Q2 and Q3.

Per-session	In	Question	Scale
yes	Q2	What is the maximum seismic in the experience?	JMA seismic intensity scale
yes	Q2	Did you feel fear in the experience?	5-level
no	Q3	Did you find the experience memorable?	7-level
no	Q3	Did you feel sick during the experience?	4-level

4.3 Experimental Results and Analysis

4.3.1 Questionnaire Results and Analysis

The result of questionnaire Q1 is shown in Fig. 4.7. It shows 15 out of 18 participants had prior experience of scale 3 or 4 earthquake. Additionally, no participant had experience of scale 6 (equivalent to session S1), and almost all (17) participants lack experience of even session S2 level earthquakes.

The lack of experience in scale 6 earthquake is not a grave problem as the participants do have some degree of regarding the matter via public education and mass media.

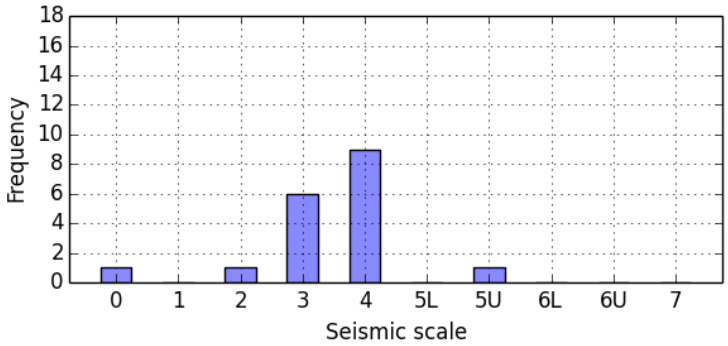

Figure 4.7: Histogram of maximum earthquake seismic intensity scale experienced by participants in the past. 0 denotes no experience.

Results of Q2 are shown in Fig. 4.8 and Fig. 4.9. Categorical seismic intensity scales are converted to continuous scales using centroids of ranges. For example, "5 lower" is converted to 4.75 because it corresponds to range $[4.5, 5)$. It can be observed that perceived seismic intensity scales increased monotonically with the simulated scale. However, perceived seismic intensity scale by all participants were smaller than simulated scale. Also, larger variance of perceived scales can be observed for weaker earthquakes (S2, S3), compared to S1. Three participants even reported the absence of earthquake for the weakest earthquake session S3. Possible reasons for this condition is discussed in subsection 4.3.2.

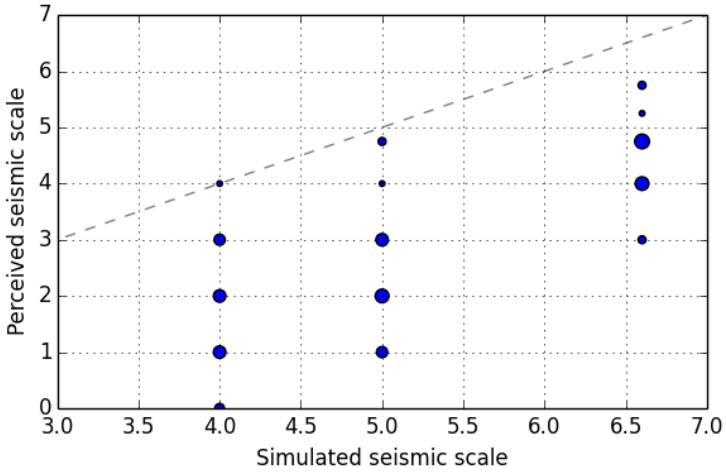

Figure 4.8: Perceived seismic intensity scale vs. simulated seismic intensity scale.

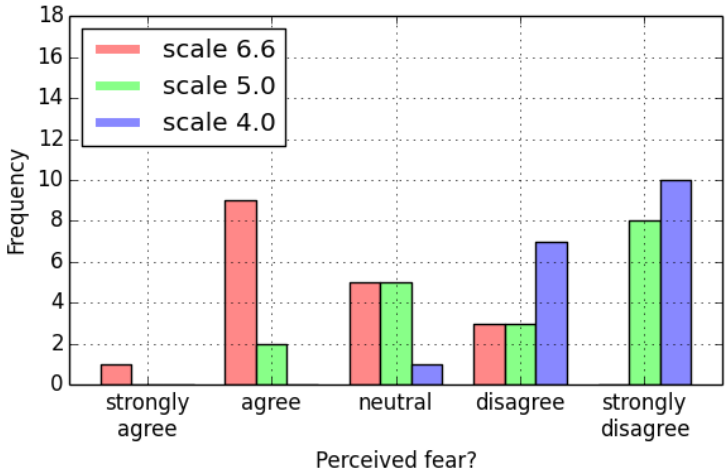

Figure 4.9: Perceived fear for each simulated scale.

Fig. 4.9 shows that more than half of the participants felt that the experience of scale 6.6 earthquake (S1) was fearful. One participant even described it "as fearful as real earthquake", in the free comment. Arguably, most people (with past experience) would not feel fear in an earthquake of around seismic intensity scale 3.0. Thus, the fact average perceived scales of S2 and S3 are under 3.0 agrees with the low fear level for S2 and S3.

So it can be stated that the system performs its primary goal quite well, which proves the approach is at least feasible. However, the reasons that perceived scales were lower than simulated need to be discussed more closely with the interview results.

The average of memorability of the experience was 5.3 ($\sigma = 1.2$), using converted scores of Table 4.8. The score corresponds to a little bit above "agree". Since no participant mentioned past VR experience of any kind, it is considered that the score reflects both

novelty and fear. The experiment cannot separate the two effects, but it can be argued that the experience of the system will be retained in a prolonged duration, possibly contributing to the preparation for earthquakes.

The averages of the four realness question answers are shown in Fig. 4.10. The first three questions measures visual realness of static overall, dynamic behavior interior objects, and dynamic behavior of the room itself, respectively. The final question corresponds to sounds as a whole, which is actually a composition of ground shaking sound and collision sounds. The categorical answers were converted to numeric scores using Table 4.8.

Table 4.8: 7-Level Likert scale and corresponding numeric scores.

7-level scale	Score
Strongly agree	7
Agree	6
Somewhat agree	5
Neutral	4
Somewhat disagree	3
Disagree	2
Strongly disagree	1

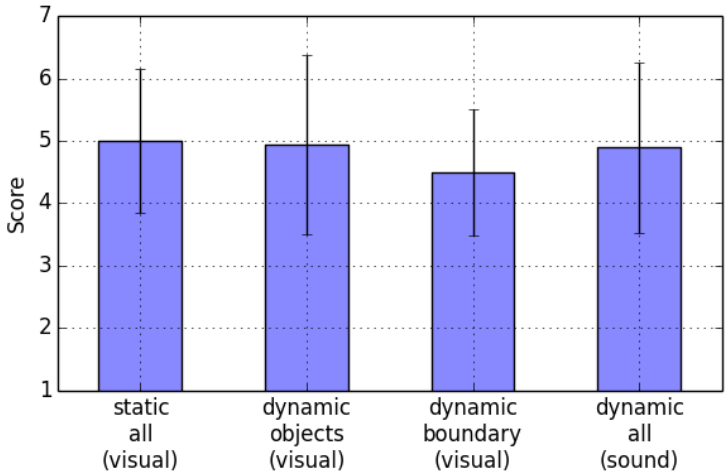

Figure 4.10: Means and standard deviations of scores for each element of the realness. Error bars denote standard deviations.

The averages of the entries except dynamic boundary visual reality are nearly 5, which corresponds to "somewhat agree". The entry of dynamic boundary visual reality scored closer to 4 ("neutral") and have smaller variance compared to others. It was found in the interview that significant number of participants were not looking at the interior boundary when earthquake is ongoing, and therefore they answered "neutral".

Since the participants were not informed that the VR content was automatically generated and existence of 3D reconstruction techniques are not widely known outside the field, the expectation of the quality would have been that of artists-created contents. So the average scores of "somewhat agree" mean that the reconstruction method can produce content with acceptable level of realness at the current state, despite it being obviously worse than an artist-made content running in a state of the art game engine (e.g. Fig. 3.55).

The VR sickness results are shown in Fig. 4.11. It shows that most (15) people felt no VR sickness. The average HMD-wearing time of the participants was roughly 10 minutes, including earthquake sessions, calibration and questionnaire using virtual screens. Thus, it can be argued that use of VR for earthquake experience does not induce problem of VR sickness, and the immersiveness will not be lessened by VR sickness in a practical shorter setting. However, combining earthquake with other VR content such as educational one, might pose a problem because of prolonged use of HMD.

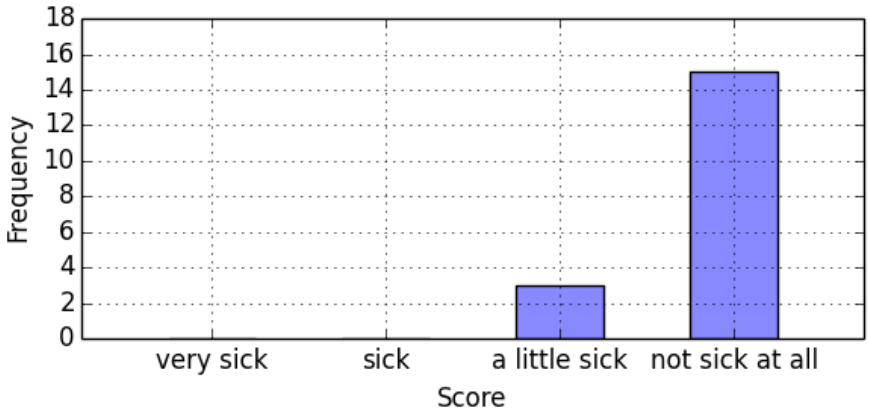

Figure 4.11: Histogram of sickness during the experience.

Since most of the free comments overlap with the interview results, they are discussed in subsection 4.3.2 along with the interview results, while the the full list is shown in Appendix C.

4.3.2 Interview Results and Analysis

Interview results are qualitatively analyzed for each realness question. Full interview result is shown in Appendix C. During interview for a question a participant answered as one of "somewhat agree", "neutral" or "somewhat disagree", the participant were explicitly asked about both realistic and unrealistic points to extract meaningful information.

Static Visual Reality of Whole Scene

There were roughly three kinds of answers: comments on VR hardware, explicit comments on the graphics, and overall feeling.

The first kind contains 10 positive comments and 1 negative comment. The former is mainly about head tracking, stereoscopy, and feeling of presence, stated by phrases such as “3D-ness”, “tangible feeling”, “really being in a room”. One of 2 participants that felt the scene seemed tangible, actually proceeded to wave the hands at virtual objects. The negative comment was about the low resolution of the screen.

The second kind contains 4 positive comments and 10 negative comments. The positive comments include reality of the walls, light fixtures, chairs, and desks. The subjects of the negative comments can be further divided into specific objects and aspects shared by all objects. In the former case, an artifact caused by failure to correctly remove wall parts from interior objects (probably Fig. 4.12), and an strangely-shaped object from missing points were mentioned. In this particular case, the LRF failed to capture LCD surface points because of low surface reflectance. The latter includes rough object silhouette, blurriness of textures, and lack of color in some textures.

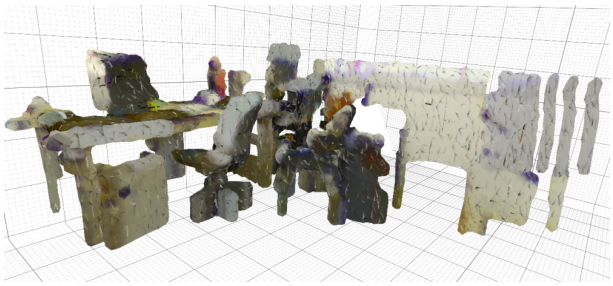

Figure 4.12: A part of wall (a flat vertical plate on the right) mistakenly included in an interior object.

The third kind of comments focused mainly on more semantic aspects of the scene. All four comments of this kind were positive, and they can be summarized as “the room can be recognized as an office-like room”.

Dynamic Visual Reality of Interior Objects

There were 13 positive comments, and their topics are decomposed into realness of earthquake pattern and physics simulation. Instances of the former are “the way things gradually began to shake was realistic”, and “the shaking have some randomness (thus it felt realistic)”. The latter includes “the objects seemed to follow a physics law similar to that of the reality”. They show advantages of using physically based simulation with real measurement data without artificial modification. However, there were 5 negative

comments in the following 3 categories.

First, one participant noted that an earthquake should contain more vertical motion. This might be attributed to a bias in the past earthquake experience of the participant, or to a lack of modeling building deformation. Second, some participants found the physics simulation unstable. Namely, several objects were vibrating seemingly on their own without external force. Since this phenomena were not always reproducible, the instability might be attributed to subtle numeric error in the implementation of the physics simulator. But there is a possibility that generating tighter collision shape for interior objects can help alleviate the issue. Finally, some participants noted that the behavior of objects are unrealistic because of too simple physics model (i.e. rigid body only) or limitation of object recognition. An example raised by one of them is behavior of chairs with casters, which should slide sideways before toppling.

In related to the last category, two participants noted in the free comments that smaller objects such as individual books in bookshelves would help in increasing realness.

Dynamic Visual Reality of Interior Boundary

This question was meant to measure perceived avatar movement which is visually equivalent to the room movement. But more than half of the participants answered that they were not paying attention to the room itself.

One participant felt a slight acceleration in S1, probably from the vection (optical flow in field of view). This is encouraging because it shows possibility of simulating acceleration without huge motion platforms. However, three other participants explicitly stated that they felt discrepancy between apparent vibration of the room and no perceived acceleration. There is a high chance that the lack of vestibular stimuli is lowering the perceived seismic intensity scales for weaker earthquakes, because acceleration is one of the few cues in weaker earthquakes. The same could be said for stronger earthquake, but stronger earthquakes also involve breaking and shattering objects, thus the extent of the effects caused by the lack of real acceleration is unknown.

However, since the participants comments were not detailed in acceleration directions or their patterns, it might be possible that the accuracy requirement of vestibular stimuli is low. If this is the case, galvanic vestibular stimulation (GVS)[60] might be used with a chance of success.

Dynamic Auditory Reality of Whole Scene

The sound generated by the system is a mixture of earthquake sound and collision sounds. The question was merged together to avoid any preconception that would be caused by asking them separately. Nonetheless, some participants noted both kinds of

sounds. But two participants did not remember the ground shaking sound, even when an experimenter asked explicitly it. However, all participants who noticed the earthquake sound commented that it was very realistic. Since the sound level was significantly lower than real earthquakes to avoid possibility of ear injury, one possible cause is the difference in sensitivity of low frequency sounds among the participants.

The collision sounds received mixed answers with 6 positive comments and 4 negative comments. No specific reason can be obtained from the positive comments, but two primary reasons were mentioned in the negative comments. The first is physics simulator problem, similar to the previous issue of unstable objects. The comment states that there were instances of collision sounds without any visible movement of objects. The second is recognition and sound synthesis problem. In this case, participants noted that there were too few types of sound patterns, compared to real earthquakes. There was also a comment related to the both, and it stated that the collision sounds did not correspond to the objects in the room. The collision sounds, and the pitch randomization parameters used in the experiment targeted lower frequency sounds (i.e. collision between larger objects such as furnitures). The parameters were chosen as so because a preliminary experiment had shown that higher frequency sounds become very apparent when not synchronized with the behavior of objects. However, two participants mentioned that the simulation should contain more higher frequency collision sounds of smaller objects.

Thus, resolving the collision sound issues would require much fine-grained understanding of the scene, including materials of objects.

Chapter 5

Conclusion

In this research, a new earthquake experience system is proposed. The proposed system simulates indoor environments in a earthquake using VR. Unlike previous VR earthquake experience systems, the proposed system generates earthquake contents automatically. The primary objective of the proposed system is to cause fear towards earthquakes at lower cost, within the existing context of fear-arousing communication in disaster education. The automatic contents generation is based on a newly proposed indoor reconstruction method.

The proposed system consists of 3D scanning subsystem to scan a room, contents generation subsystem to reconstruct it, and the runtime subsystem to simulate and present earthquakes to the users. The reconstruction method models a room as a combination of an interior boundary and interior objects and generates corresponding contents. The runtime subsystem simulates motion of the generated objects using rigid-body physics simulation, and presents the simulation using a headphone and an HMD. It was shown that a non-optimized implementation of the reconstruction method can actually handle a few types of indoor scenes, within reasonable computation times of less than 40 minutes.

An experimental evaluation with 18 participants was conducted in order to examine whether the system can achieve its primary objective of causing fear. In the experiment, participants experienced the earthquake simulated by the system, and perceived fear, realness, memorability, seismic intensity scales, and VR sickness were measured by questionnaires. Interviews were also conducted to investigate factors affecting perceived realness. The evaluation experiment showed that the strongest earthquake session of seismic intensity scale 6.6 could actually cause fear in more than half participants, despite the lack of acceleration stimuli through vestibular senses. Thus, it can be said that the system is capable in reaching its primary goal. Also, the realness of each scene element scored moderately good results, especially considering that the participants were unaware of the fact that the contents are automatically generated.

However, it was found that perceived seismic intensity scales were consistently lower than the seismic intensity scales of simulated earthquakes. Qualitative analysis of the interview results shows that the discrepancy could be attributed to lack of vestibular senses and lack of scene details. Earthquake sounds generated directly from earthquake acceleration were evaluated as very realistic by all participants who noticed them, although its exact physical mechanism remains unknown. However, several participants were unable to hear it, probably due to low sound pressure level compared to the real earthquake. There is a need to investigate sound pressure levels of real phenomena including both earthquake itself and collisions, in order to ensure they are audible and physically plausible.

The reconstructed interior boundary was received as realistic, despite the simplistic modeling as an extruded polygon. However, there was an instance where a piece of wall was mistakenly recognized as an object, because of errors due to recessed windows. Considering the current inability to detect sunlight from windows, the room frame model could be more sophisticated. For example, it might be viable to model windows or beams explicitly as 3D shapes attached to the room frame. Motion of objects and collision sounds suffered from lack of recognition of articulated parts, smaller objects, and materials. Since these are instances of the hardest computer vision problems, a satisfactory solution would require significant development in the field.

In addition to lack of complex behavior and proper collision sounds, the usability of the system has a room for improvement in the following regards. First, the scanning device should have a user interface to direct operators to optimal scan locations. Second, details of the simulation such as initial location of avatars, should be determined automatically. The lack of acceleration stimuli could be addressed by galvanic vestibular stimulation or motion platforms.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to Prof. Hiroshi Shimoda and Asst. Prof. Hirotake Ishii, who have guided me with numerous advices and vast knowledge. Their guidance greatly helped me not only in writing this thesis, but nurturing attitude toward research in general.

Also, I would like to thank Yongxin Wang and Razana Hsuni, who supported me by proof-reading the thesis and gave me insightful comments.

Furthermore, I would like to greatly thank Ikumi Fusho who helped me with a large ammount of paperworks associated with experiments as a secretary. Without her help, the experiment would be impossible. My gratitude also goes to Shota Shimonaka and Takuya Fujii, who helped me as experiment assistants. I greatly thank Taro Kimura and Masanari Furuta, for bearing my terrible execution of experimenter role, and yet giving me many useful comments in beta-testing phase.

Finally, I would like to thank all labmates for insightful discussion and having fun time with me during all three years.

Bibliography

[1] Natascha de Hoog, Wlfgang Stroebe, John B. F. de Wit: The Impact of Fear Appeals on Processing and Acceptance of Action Recommendations. Personality and Social Psychology Bulletin, **31(1)**, pp. 24–33 (2005).

[2] Ravi Sinha, Ashish Sapre, Atul Patil, Ankur Singhvi, Mukund Sathe, Vivek Rathi: Earthquake Disaster Simulation in Immersive 3D Environment. In Proceedings of 15th World Conference on Earthquake Engineering (2012).

[3] Zhaoyin Jia, Andrew Gallagher, Ashutosh Saxena, Tsuhan Chen: 3D-Based Reasoning with Blocks, Support, and Stability. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2013).

[4] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, Andrew Fitzgibbon: KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, pp. 559–568 (2011).

[5] Ben Wisner, Piuers Balikie, Terry Cannon, Ian Davis: At Risk: Natural hazards, people's vulnerability and disasters, pp. 286–287. Psychology Press (2003).

[6] Japan Meteorological Agency: パンフレット「地震と津波」[Pamphlet "Earthquake and Tsunami"]. `http://www.jma.go.jp/jma/kishou/books/jishintsunami/index.html`. Accessed 2015-01-09.

[7] Cabinet Office: 自然災害の記録 (地震編) [Record of Natural Disaste (Earthquake)]. `http://nettv.gov-online.go.jp/prg/prg555.html`. Accessed 2015-01-09.

[8] Ryoso Motor Industries Ltd.: 起震車諸元 [Specification of Earthquake Simulation Vehicle]. `http://www.ryoso.com/syasyu/kishin/index2.html`. Accessed: 2015-01-01.

[9] Japan Meteorological Agency: 震度について [About Seismic Scale]. `http://www.jma.go.jp/jma/kishou/know/shindo/`. Accessed 2015-01-04.

[10] Keiichi Ohtani, Nobuyuki Ogawa, Tsuneo Katayama, Heki Shibata: Construction of E-Defense (3-D full-scale earthquake testing facility). In Proceedings of 2nd International Symposium on New Technologies for Urban Safety of Mega Cities in Asia, pp. 69–76 (2003).

[11] Alfred T. Lee, Steven R. Bussolari: Flight Simulator Platform Motion and Air Transport Pilot Training. Aviation Space and Environmental Medicine, **60(2)**, pp. 136–140 (1989).

[12] How Much Should I Charge For Freelance 3D Modelling Work? `http://www.katsbits.com/articles/how-much-should-i-charge-for-freelance-3d-modeling-work.php`. Accessed 2015-01-29.

[13] Oculus VR LLC: Oculus Rift - Virtual Reality Headset for 3D Gaming. `https://www.oculus.com/dk2/`.

[14] Richard Szeliski: Computer Vision: Algorithms and Applications, pp. 656–657. Springer (2010).

[15] Oliver Mattausch, Daniele Panozzo, Claudio Mura, Olga Sorkine-Hornung, Renato Pajarola: Object Detection and Classification from Large-Scale Cluttered Indoor Scans. Computer Graphics Forum, **33(2)**, pp. 11–21 (2014).

[16] Andreas Richtsfeld, Thomas Morwald, Johann Prankl, Michael Zillich, Markus Vincze: Segmentation of Unknown Objects in Indoor Environments. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4791–4796 (2012).

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich: Going Deeper with Convolutions. Computing Research Repository, **arXiv:1409.4842** (2014).

[18] Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov: Scalable Object Detection using Deep Neural Networks. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, volume abs/1312.2249, pp. 2155–2162 (2014).

[19] Bo Zheng, Yibiao Zhao, J.C. Yu, K. Ikeuchi, Song-Chun Zhu: Detecting potential falling objects by inferring human action and natural disturbance. In Proceedings of IEEE International Conference on Robotics and Automation, pp. 3417–3424 (2014).

[20] Bo Zheng, Yibiao Zhao, Joey C. Yu, Katsushi Ikeuchi, Song-Chun Zhu: Beyond Point Clouds: Scene Understanding by Reasoning Geometry and Physics. In Proceedings

of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3127–3134 (2013).

[21] Alexander G. Schwing, Tamir Hazan, Marc Pollefeys, Raquel Urtasun: Efficient Structured Prediction for 3D Indoor Scene Understanding. In Conference on Computer Vision and Pattern Recognition, pp. 2815–2822 (2012).

[22] Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, Daniel Cohen-Or: 3Sweepp: Extracting Editable Objects from a Single Photo. ACM Transactions on Graphics, **32(6)**, pp. 195:1–195:10 (2013).

[23] Eugenia M. Kolasinski: Simulator sickness in virtual environments. Technical Report 1027, U.S. Army Research Institute for the Behavioral and Social Sciences (1995).

[24] National Research Institute for Earth Science, Disaster Prevention: Strong-motion seismograph networks (K-NET, KiK-net). `http://www.kyoshin.bosai.go.jp/`. Accessed 2015-01-02.

[25] Toshinori Ariga, Yoshihiro Kanno, Izuru Takewaki: Resonant Behaviour of Base-Isolated High-Rise Buildings under Long-Period Ground Motions. The Structural Design of Tall and Special Buildings, **15(3)**, pp. 325–338 (2006).

[26] Takashi Hirai, Kazumi Kurata, Nobuo Fukuwa, Masafumi Mori: Synthesis of Earthquake Sound Using Seismic Motion Record and its Application to Audiovisual Earthquake Experience System. In Proceedings of 15th World Conference on Earthquake Engineering (2012).

[27] Brandon Lloyd, Nikunj Raghuvanshi, Naga K. Govindaraju: Sound Synthesis for Impact Sounds in Video Games. In Proceedings of Symposium on Interactive 3D Graphics and Games, pp. 55–62 (2011).

[28] Yuta Yamamoto: 拡張現実感技術を用いた照明シミュレーションシステムの開発 [Development of Lighting Simulation System Using Augmented Reality]. B.S. thesis, Kyoto University (2014).

[29] Matthew Brown, David G. Lowe: Automatic Panoramic Image Stitching Using Invariant Features. International Journal of Computer Vision, **74(1)**, pp. 59–73 (2007).

[30] Dan B. Goldman: Vignette and Exposure Calibration and Compensation. IEEE Transactions on Pattern Analysis and Machine Intelligence, **32(12)**, pp. 2276–2288 (2010).

[31] Shuji Oishi, Ryo Kurazume, Yumi Iwashita, Tsutomu Hasegawa: Denoising of Range Images using a Trilateral Filter and Belief Propagation. In Proceedings of International Conference on Intelligent Robots and Systems, pp. 2020–2027 (2011).

[32] James M. Coughlan, Alan L. Yuille: The Manhattan World Assumption: Regularities in scene statistics which enable Bayesian inference. In Proceedings of Neural Information Processing Systems Conference (2000).

[33] Radu B. Rusu, Nico Blodow, Zoltan C. Marton, Michael Beetz: Aligning Point Cloud Views using Persistent Feature Histograms. In Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3384–3391 (2008).

[34] Nicolas Mellado, Dror Aiger, Niloy J. Mitra: Super 4PCS Fast Global Pointcloud Registration via Smart Indexing. Computer Graphics Forum, **33(5)**, pp. 205–215 (2014).

[35] Szymon Rusinkiewicz, Marc Levoy: Efficient variants of the ICP algorithm. In Proceedings of 3rd International Conference on 3-D Digital Imaging and Modeling, pp. 145–152 (2001).

[36] Martin A. Fischler, Robert C. Bolles: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, **24(6)**, pp. 381–395 (1981).

[37] Robert C. Prim: Shortest Connection Networks and Some Generalizations. Bell System Technical Journal, **36(6)**, pp. 1389–1401 (1957).

[38] Radu Bogdan Rusu, Steve Cousins: 3D is here: Point Cloud Library (PCL). In IEEE International Conference on Robotics and Automation (2011).

[39] Adriano J. C. Moreira, Maribel Yasmina Santos: Concave Hull: A k-nearest Neighbours Approach for the Computation of the Region Occupied by a Set of Points. In Proceedings of International Conference on Computer Graphics Theory and Applications, pp. 61–68 (2007).

[40] Hirotsugu Akaike: A New Look at the Statistical Model Identification. IEEE Transactions on Automatic Control, **19(6)**, pp. 716–723 (1974).

[41] CGAL, Computational Geometry Algorithms Library. `http://www.cgal.org`. Accessed 2015-01-13.

[42] Radu B. Rusu: Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. Ph.D. thesis, Computer Science department, Technische Universitaet Muenchen, Germany (2009).

[43] Miomir Vukobratović, Branislav Borovac: Zero-Moment Point - Thirty Five Years of its Life. International Journal of Humanoid Robotics, **1(1)**, pp. 157–173 (2005).

[44] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, Jérôme Maillot: Least Squares Conformal Maps for Automatic Texture Atlas Generation. In Proceedings of ACM SIGGRAPH, pp. 362–371 (2002).

[45] Carsten Rother, Vladimir Kolmogorov, Andrew Blake: "GrabCut": Interactive Foreground Extraction Using Iterated Graph Cuts. In Proceedings of ACM SIGGRAPH, pp. 309–314 (2004).

[46] Alexandru Telea: An Image Inpainting Technique Based on the Fast Marching Method. Journal of Graphics, GPU, and Game Tools, **9(1)**, pp. 23–34 (2004).

[47] Michael Kazhdan, Matthew Bolitho, Hugues Hoppe: Poisson Surface Reconstruction. In Proceedings of the Fourth Eurographics Symposium on Geometry Processing, pp. 61–70 (2006).

[48] Ravikrishna Kolluri, Jonathan R. Shewchuk, James F. O'Brien: Spectral Surface Reconstruction from Noisy Point Clouds. In Proceedings of Symposium on Geometry Processing, pp. 11–21. ACM Press (2004).

[49] James F. Blinn: A Generalization of Algebraic Surface Drawing. ACM Trans. Graph., **1(3)**, pp. 235–256 (1982).

[50] Jon Louis Bentley: Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM, **18(9)**, pp. 509–517 (1975).

[51] Jean-Daniel Boissonnat, Steve Oudot: Provably Good Sampling and Meshing of Surfaces. Graph. Models, **67(5)**, pp. 405–451 (2005).

[52] Matt Pharr, Greg Humphreys: Physically Based Rendering, Second Edition: From Theory To Implementation (2010).

[53] Changxi Zheng, Doug L. James: Toward High-quality Modal Contact Sound. ACM Transactions on Graphics, **30(4)**, pp. 38:1–38:12 (2011).

[54] Zhimin Ren, Hengchin Yeh, Ming C. Lin: Example-guided Physically Based Modal Sound Synthesis. ACM Trans. Graph., **32(1)**, pp. 1:1–1:16 (2013).

[55] Steven S. An, Doug L. James, Steve Marschner: Motion-driven Concatenative Synthesis of Cloth Sounds. ACM Trans. Graph., **31(4)**, pp. 102:1–102:10 (2012).

[56] Warrick Roseboom, Shin'ya Nishida, Derek H. Arnold: The Sliding Window of Audio-Visual Simultaneity. Journal of Vision, **9(12)**, pp. 1–8 (2009).

[57] freesound. `http://freesound.org/`. Accessed: 2014-12-31.

[58] Henrik Møller, Christian S. Pedersen: Hearing at low and infrasonic frequencies. Noise & Health, **6(23)**, pp. 37–57 (2004).

[59] Japan Meteorological Agency: 気象庁震度階級関連解説表 [Tables explaining the JMA Seismic Intensity Scale]. `http://www.jma.go.jp/jma/kishou/know/shindo/kaisetsu.html`. Accessed 2015-01-21.

[60] Richard C. Fitzpatrick, Jane E. Butler, Brian L. Day: Resolving Head Rotation for Human Bipedalism. Current Biology, **16(15)**, pp. 1509–1514 (2006).

[61] Japan Meteorological Agency: 震 度 解 説 概 要 [Summary of Seismic Intensity Scales]. `http://www.jma.go.jp/jma/kishou/know/shindo/jma-shindo-kaisetsu-gaiyo.pdf`. Accessed 2015-01-27.

Appendix A

UV Parametrization of Triangle Mesh

One way to represent a 3D surface is a triangle mesh, which is used throughout the system. A triangle mesh is a composition of vertices and triangles. Together, they represent a 2D manifold. The goal of UV parametrization is to embed the manifold in a 2D plane. In general, splitting the manifold into several parts is necessary to achieve it. For instance, a sphere cannot be continuously transformed to any 2D planar region without creating seams in it. These parts are called charts.

In the ad-hoc parametrization method used in the system, completely planar regions formed by connected triangles are treated as charts. This works well for mostly planar shapes such as interior boundary. In the worst case of a round shape that all edges have some angle, each triangle become a chart. An example of the results that would be generated by a triangle mesh of a pointy cube is shown in Fig. A.1.

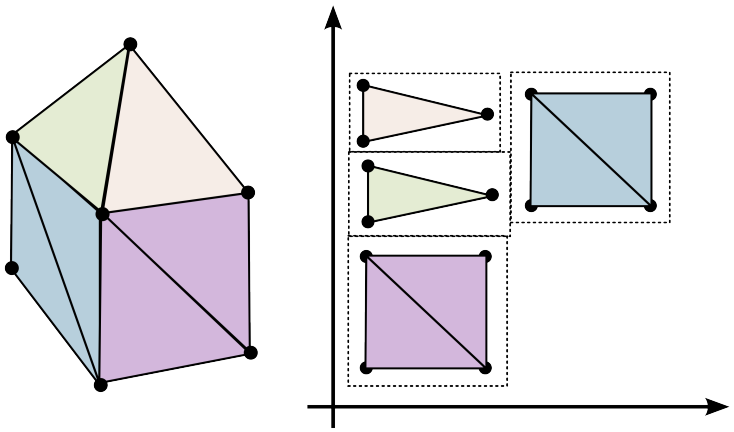

Figure A.1: A triangle mesh (left) and the UV coordinates of the vertices (right).

The method works as follows. First, a dual graph of the input triangle mesh is cre-

ated. Then, edges with non-zero angles are removed from the graph. Since each connected component of the graph corresponds to a planar region in 3D space, the connected components become charts. Since each chart consists of completely coplanar triangles, it can be projected to the coplanar plane without distortion. Then, an AABB is calculated for the projected triangles, and margins of a few pixels are inserted around the AABB to inhibit unintended blending with other charts.

At this stage, all charts have AABBs that wraps them. These AABBs are packed into a square with a greedy algorithm. The initial size of the square is determined by the square root of the surface area of the mesh. Because of gaps between AABBs, they cannot be packed to this minimum square in general. So, the size of the squares is gradually increased, until the AABBs can be fitted in the square.

Finally, the square is scaled to $[0, 1]^2$, typically used ranges of the UV coordinates.

Appendix B

Questionnaires

Floating screens were used to clearly convey meanings of the questionnaire while participants are wearing an HMD.

The questionnaire before the experience is shown in Fig. B.2 (Q1), and the questionnaire after the experience is shown in Fig. B.3 (Q2), Fig. B.4 (Q3).

Fig. B.1 shows the table of seismic intensity scale descriptions, shown in the beginning of the experiment.

An illustrated seisimic scale table, accompanied with the question written in the top, is shown in Fig. B.5. Although participants were asked to remeber seismic intensity scale table in text form before the earthquake experience, this screen was used to aid participants futhermore.

A question to ask fear level using Likert scale is shown in Fig. B.6. The screen is used because the answer list is hard to convey orally.

参考 震度と体感・屋内の状況

震度階級	人の体感・行動	屋内の状況
1	屋内で静かにしている人の中には、揺れをわずかに感じる人がいる。	－
2	屋内で静かにしている人の大半が、揺れを感じる。眠っている人の中には、目を覚ます人もいる。	電灯などのつり下げ物が、わずかに揺れる。
3	屋内にいる人のほとんどが、揺れを感じる。歩いている人の中には、揺れを感じる人もいる。眠っている人の大半が、目を覚ます。	棚にある食器類が音を立てることがある。
4	ほとんどの人が驚く。歩いている人のほとんどが、揺れを感じる。眠っている人のほとんどが、目を覚ます。	電灯などのつり下げ物は大きく揺れ、棚にある食器類は音を立てる。座りの悪い置物が、倒れることがある。
5弱	大半の人が、恐怖を覚え、物につかまりたいと感じる。	電灯などのつり下げ物は激しく揺れ、棚にある食器類、書棚の本が落ちることがある。座りの悪い置物の大半が倒れる。固定していない家具が移動することがあり、不安定なものは倒れることがある。
5強	大半の人が、物につかまらないと歩くことが難しいなど、行動に支障を感じる。	棚にある食器類や書棚の本で、落ちるものが多くなる。テレビが台から落ちることがある。固定していない家具が倒れることがある。
6弱	立っていることが困難になる。	固定していない家具の大半が移動し、倒れるものもある。ドアが開かなくなることがある。
6強	立っていることができず、はわないと動くことができない。揺れにほんろうされ、動くこともできず、飛ばされることもある。	固定していない家具のほとんどが移動し、倒れるものが多くなる。
7		固定していない家具のほとんどが移動したり倒れたりし、飛ぶこともある。

（気象庁震度階級関連解説表より）

3

Figure B.1: Seismic intensity scales and descriptions.[59]

参加者 ID	

質問 1 実際に経験したことのある最大の震度はいくつですか？（数値を覚えていない場合は、実験説明資料の最後のページの表を参考にしてください）

☐ 震度＿＿＿＿

☐ 地震を実際に経験したことがない

Figure B.2: Pre-VR questionnaire page 1.

参加者 ID	

質問 1 揺れていない時 3D 環境は写実的に見えましたか？

非常にあてはまる	あてはまる	どちらかといえばあてはまる	どちらともいえない	どちらかといえばあてはまらない	あてはまらない	全くあてはまらない
□	□	□	□	□	□	□

質問 2 部屋の中の物体の揺れを実際の地震と似ていると感じましたか？

非常にあてはまる	あてはまる	どちらかといえばあてはまる	どちらともいえない	どちらかといえばあてはまらない	あてはまらない	全くあてはまらない
□	□	□	□	□	□	□

質問 3 部屋自身の揺れを実際の地震と似ていると感じましたか？

非常にあてはまる	あてはまる	どちらかといえばあてはまる	どちらともいえない	どちらかといえばあてはまらない	あてはまらない	全くあてはまらない
□	□	□	□	□	□	□

質問 4 音を実際の地震と似ていると感じましたか？

非常にあてはまる	あてはまる	どちらかといえばあてはまる	どちらともいえない	どちらかといえばあてはまらない	あてはまらない	全くあてはまらない
□	□	□	□	□	□	□

質問 5 この地震シミュレーションは印象に残りましたか？

非常にあてはまる	あてはまる	どちらかといえばあてはまる	どちらともいえない	どちらかといえばあてはまらない	あてはまらない	全くあてはまらない
□	□	□	□	□	□	□

質問 6 シミュレーションでどの程度酔いを感じましたか？

非常に気持ち悪い	気持ち悪い	少し気持ち悪い	全く気持ち悪くない
□	□	□	□

Figure B.3: Post-VR questionnaire page 1.

質問 7 実験全体について何か意見や感想があればお書きください。

2

Figure B.4: Post-VR questionnaire page 2.

Figure B.5: An illustrated table to help participants remember seismic intensity scales[61], rearranged to increase visiblity in VR.

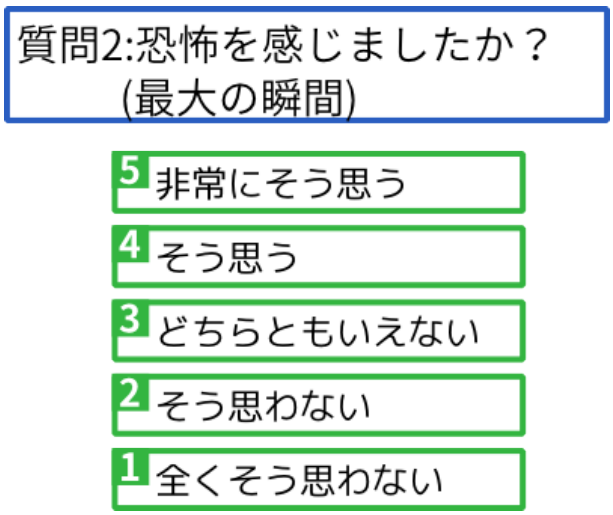

Figure B.6: Five-level Likert scale to ask level of perceived fear.

Appendix C

Full Questionnaire and Interview Results

All answers in the interview is shown in Table C.1. Content in bracket is estimated by an experimenter from context or gestures of participants.

Table C.1: All interview answers.

Participant	Question	Answer (Japanese)
1	1	触りたくなるような感じだった。CG の限界はあるが、まぁ触りたい程度には [リアル]。
1	2	実際の体験がないのでよくわからない。TV などのメディアで目にする地震かは [今回の実験に比べて] 大きすぎるのでやっぱりわからない。机とか椅子に注目していた。
1	3	先ほどと同様で、実際の体験がないのでよくわからない。机とか椅子に注目していたので、他の部分はよく見ていなかった。
1	4	同様に、実際の体験がないのでよくわからない。椅子とか机のガシャガシャ言うのはそれっぽい。
2	1	壁の様子、照明がすごくありそうで現実的。ダメな点は、部屋の奥の方のオブジェクト [薄い板] が不自然。
2	2	最大の時に一気に揺れて、そこまで徐々に振動が強くなるのがリアル。音はするのに、音と揺れが対応していない。揺れがランダムに感じる時があって不自然
2	3	壁にかかってもいない、照明がぶら下がり式でわからない。自分自身の揺れはわからない。
2	4	徐々に大きくなるのがいい。低い振動音が実際と似ている。
3	1	思ってたよりも 3D 感があった。

Participant	Question	Answer (Japanese)
3	2	映像がすごいリアルなので、本物っぽい。しかし実際に揺れを感じない点が良くない。
3	3	映像がすごいリアルなので、地震っぽい。色味などが実際の部屋と違うのでちょっとだめ。
3	4	実際はもっと食器の音などもする、揺れてる時の音っぽくない。
4	1	目の前にある椅子を触りたくなる。[HMD とかの] 経験がないので、すごく写実的だと思った。
4	2	「非常にそう思う」ではない理由は、物が滑らないから。揺れはかなり再現されていると思う。
4	3	部屋の揺れはそれほど見ていなかったが、トータルではいいと思ったので。
4	4	音をあまり意識したことがなかった、ゴーっていう音については [感じなかったので] わからない。衝突音についてはすごいリアル。
5	1	椅子とか机は本物に近い。全体として輪郭などがぼやけていた。
5	2	椅子がこう [横に] 揺れてるのが実際っぽい
5	3	雰囲気がそういう感じ。揺れたらこんな感じかなーと感じた。
5	4	ゴーっていう音が、自宅はマンションだが、実際の地震が来た時に、最初は子供とかが遊んでるのかと思ったが、その時の感覚と似ていた。
6	1	こういうの [VR] はじめてで、[トラッキングが] 画期的。ゲーム画面っぽいので写実的でない。
6	2	一箇所、椅子がすごい揺れて単のがぽいなーと、経験はそんなにないが、それっぽい。
6	3	震度が大きそうなときに、リアルに感じられたのが似ている。
6	4	[大きい地震] の経験がないので、こんなものかなーと思った。[メディアで見る地震からの想像]
7	1	立体的に見えたのがよい。全体的に部屋に居るようには感じたが、中にあるもの [オブジェクト] が現実的ではないように感じた。
7	2	ぱっと見た印象で実際の世界にあるようなものと似た物理法則に従っているようには見える。椅子とか机とかが実際より軽く見える。

Participant	Question	Answer (Japanese)
7	3	部屋を見て、実際地震があればこういう揺れがあるんだろうなーという印象。自分が実際に揺れてないので、それで違和感。
7	4	地震を体験した時に、地鳴りのような音が実際と似ていると感じた。あまり地震の音の経験がないので。
8	1	距離感とかはいい感じ、物体の質感などが合成映像のように見える。
8	2	ガタガタと小刻みに揺れたり、とんと揺れたり、実際の揺れの時に感じたのと同じように感じた。
8	3	大きな揺れの時は揺れを感じたが、小さい揺れの時は画面のブレは感じるが、地震の感覚がないのが不自然。
8	4	実際に体験したのと同じような音。[衝突音れがリアル、地鳴りは聞こえなかった]
9	1	写実的の定義がよく分からなかった。全体的に物体がなにか分かる。
9	2	イメージと合ってる気がする
9	3	イメージと合ってる気がする
9	4	一番弱い地震では音が聞こえなかった。
10	1	椅子の感じとかがうまく出来ていた。説明はできないがリアルに感じた。
10	2	あまり地震の体験がないので、わからない。
10	3	あまり地震の体験がないので、わからない。
10	4	がちゃがちゃする音はリアルだと思った。
11	1	画像が荒いので、作り物感があった。スクリーンの粗さはちょっとあるが、物体の形状が荒いのが 8 割ぐらい。
11	2	メディアなどで見る映像に似ている。
11	3	壁とかは物に比べると揺れてないと感じた。揺れの体感がない。
11	4	音の種類が実際より少ない。ご一っというのと、物の当たる音だが、衝突音の種類が少ない、パターンが不自然。
12	1	あんまり鮮明に見えてなかった、シルエットはわかるんけど、表面とかがよくわからない
12	2	よく、講義などでも災害の映像などを見るが、徐々に揺れ始めて揺れが強くなっていく様子がそれに似ていた。

Participant	Question	Answer (Japanese)
12	3	物が揺れているのはわかったが、部屋の揺れはそれほど感じなかったし、それほど注視していなかった。
12	4	地響きみたいなのが似てる、一番最後の映像で金属類が揺れるような音はなんの音かわからなかった。
13	1	普通にそういうふう [現実のシーンのよう] に見えた
13	2	一番強い地震で、机が不自然に感じた。オブジェクトが揺れの強さに対して、倒れないのが不自然。椅子も倒れないのが不自然。
13	3	実体験ではベッドの上とかにいたので、揺れを感じたことがなかった。今回もあまり見てなかったのでよくわからない。自分自身は最大の地震では少し揺れを感じた。
13	4	ゴ―っていう音が揺れてる時に聞くのに似ているなーと思った。
14	1	目の前の椅子は結構立体的でいい感じ、右側のスクリーンが台形みたいに歪んでいてそれが不自然に感じた。奥行きとかは立体的に見える。
14	2	最初は椅子がカタカタ揺れていたが、それが徐々に足が浮くように強く揺れる様子がリアルだと思った。
14	3	地震体験車などと違って自分が揺れていても、実際に揺れていないので揺れてる実感が少なかった。
14	4	自分の経験のゴ―って音と今回の音がかなり似ていた。あまり揺れてない時でも音がするのはちょっと不自然。
15	1	立体的に机とか椅子とか部屋とかが見えたので、3Dっぽいから写実的かな、写真とかと比べると画像感はある。
15	2	がたがたっていうときの動きが、すごくリアル。
15	3	そんなに部屋全体を注視してなかったので当てはまるとは言えないが、違和感はなかった。地震の様子が伝わってきた。
15	4	地震の体験はあるが、物体のがたがたする音は似ていたと思う。ゴォ―っていう音が、自信はないけど、似てた気がする。
16	1	見た目が現実的で、よくあるオフィスの風景に見えた。
16	2	実際はもうちょっと縦方向とか、複雑な揺れ方をする気がする。[揺れが] 単調に感じる。
16	3	地震の経験がそれほどないのでなんとも言えない。[揺れを vection で感じてはいない]

Participant	Question	Answer (Japanese)
16	4	結構、からんからんという音があったが、実際の地震の時にそういうのを聞いたことはない。
17	1	周りを見回した時に、自分より後ろに物があったのは写実的。しかし、映像がなんとなく、物の形はそれっぽいがたとえば色が無かったりして、写実性がない。
17	2	揺れてるなーという感じはした、研究室っぽい感じで、普段の空間と違ったのでよくわからなかったが、揺れ自体はそれっぽく感じた。
17	3	先ほどと同じような理由
17	4	物が落ちる音とかは良かった、地響きみたいな音があまり感じなかった、音量は十分だと思うがなんとなく迫力がない。
18	1	実際に [頭を動かして] 見回しても、見てる方向に対応して部屋があって、本当に居るように思った。
18	2	椅子などがキャスター付きのように思ったが、固定されているように見えて、動いていかなかったので作り物っぽかった
18	3	さっきと少し似てる、現実では自分が止まってて部屋が揺れる感じだが、今回は部屋も何かに固定さてて揺れてるように見えたので、揺れ方がなにか違う。
18	4	椅子などががたがた言ってるのは映像と合ってたが、後ろで物で落ちたりするものがないので、その音が無くて、不自然。音だけ聞えるケースがある。

Nine participants answered the free comment section. The transcriptions are shown in Table C.2.

Table C.3 shows results of questionnaire Q1 and Q2, along with session orders of participants. Table C.4 shows non-text results of questionnaire Q3.

Table C.2: Free comments.

Participant	Comment
1	揺れが小さくて分かりにくかった
2	手前のオブジェクトだけがずっと小刻みに揺れていたのが気になった
6	選択肢の図がボケていてよく見えませんでした / 座りの悪い小物がもっとあれば震度が判断しやすかったのかなと感じた
8	大きな揺れの時には実際に地震を経験しているような恐怖感があった。
11	解像度があまり高くないので現実と比べるとあまりリアリティを感じなかった。反対に、物の揺れ方は現実味を感じることができた。ただ、地震の揺れ方のパターンが同じような感じでもう少し起伏のある方が自分にとっては実際の地震に用に感じると思う。
12	ヘッドホンからの音の情報がもう少しあればいいと思った。
14	本などが入った棚があればわかりやすかった。
18	VR による空間は良く出来ているように思った。本当に映しだされた空間にいるみたいだった。

Table C.3: Answers to Q1 and Q2.

			S1		S2		S3	
Participant	Max Scale	Order	Fear	Scale	Fear	Scale	Fear	Scale
1	2	S1,S2,S3	2	3	1	1	1	0
2	4	S1,S3,S2	2	3	1	1	1	1
3	4	S2,S1,S3	4	4	4	2	3	1
4	5 強	S2,S3,S1	3	5 弱	1	2	1	1
5	4	S3,S1,S2	4	5 弱	1	1	2	2
6	4	S3,S2,S1	4	4	2	2	2	3
7	3	S1,S2,S3	3	5 弱	3	3	2	2
8	3	S1,S3,S2	4	5 弱	2	3	1	1
9	0	S2,S1,S3	3	4	1	2	1	0
10	4	S2,S3,S1	4	5 弱	1	3	1	2
11	4	S3,S1,S2	3	4	1	2	1	2
12	3	S3,S2,S1	5	6 弱	4	5 弱	2	3
13	3	S1,S2,S3	3	4	1	1	1	1
14	4	S1,S3,S2	4	5 弱	3	3	2	3
15	4	S2,S1,S3	2	4	3	3	1	2
16	3	S2,S3,S1	4	5 弱	2	2	1	0
17	4	S3,S1,S2	4	5 強	3	4	2	3
18	3	S3,S2,S1	4	6 弱	3	5 弱	2	4

Table C.4: Answers to Q3. (V: visual aspect, A: auditory aspect)

Participant	Static scene (V)	Objects (V)	Boundary (V)	All (A)	Memorable	Sick
1	5	4	4	4	2	1
2	5	5	4	6	6	1
3	6	5	5	3	5	1
4	7	6	6	6	6	1
5	5	6	6	6	5	1
6	4	6	6	6	6	1
7	5	5	5	5	7	1
8	3	6	5	6	6	1
9	5	6	6	5	5	1
10	6	4	4	7	4	1
11	3	6	3	3	4	1
12	3	6	4	5	6	1
13	6	2	4	6	6	2
14	5	7	3	6	7	1
15	5	6	5	5	4	1
16	5	3	4	2	6	1
17	5	4	4	3	5	2
18	7	2	3	4	6	2