

# Machines' Perception of Space

by

**Wenzhe Peng**

Master of Architecture  
University of California, Berkeley, 2015

Bachelor of Architecture  
Southeast University, 2013

SUBMITTED TO THE DEPARTMENT OF ARCHITECTURE IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE IN ARCHITECTURE STUDIES  
AT THE  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY  
JUNE 2018

©2018 Wenzhe Peng. All rights reserved.

The author hereby grants to MIT permission to reproduce and to distribute  
publicly paper and electronic copies of this thesis document in whole or in  
part in any medium now known or hereafter created.

Signature of Author: \_\_\_\_\_

**Signature redacted**

Department of Architecture  
May 24, 2018

Certified by: \_\_\_\_\_

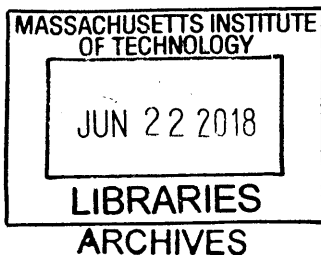
**Signature redacted**

Takehiko Nagakura  
Associate Professor of Design and Computation  
Thesis Supervisor

Accepted by: \_\_\_\_\_

**Signature redacted**

Sheila Kennedy  
Professor of Architecture  
Chair of the Department Committee on Graduate Students





## **Thesis Committee**

### **Takehiko Nagakura**

Associate Professor of Design and Computation,  
Department of Architecture

### **George Stiny**

Professor of Design and Computation,  
Department of Architecture

# **Machines' Perception of Space**

by

**Wenzhe Peng**

Submitted to the Department of Architecture  
on May 24, 2018 in Partial Fulfillment of the  
Requirements for the Degree of Master of Science in Architecture Studies

## **ABSTRACT:**

Architectural design is highly dependent on the architect's understanding of space. However, in the era of digital revolution, when efficiency and economy are the major concerns in most industrial fields, whether a computer can gain human-like understanding to read and operate space and assist with its design and analysis remains a question.

This thesis focuses on the geometrical aspects of spatial awareness. Machine systems that have similar behaviors to humans' perceptions of space in geometric aspects will be developed employing techniques such as isovist and machine learning, and trained with open-sourced datasets, self-generated datasets or crowdsourced datasets. The proposed systems simulate behaviors including space composition classification, space scene classification, 3D reconstruction of space, space rating and algebraic operations of space. These aspects cover topics ranging from pure geometrical understandings to semantic reasoning and emotional feelings of space.

The proposed systems are examined in two ways. Firstly, they are applied to a real-time space evaluation modeling interface, which gives a user instant insights about the scene being constructed; Secondly, they are also undertaken in the spatial analysis of existing architectural designs, namely small designs by Mies van der Rohe and Aldo van Eyck. The case studies conducted validate that this methodology works well in understanding local spatial conditions, and that it can be helpful either as a design aid tool or in spatial analysis.

### **Thesis Supervisor:**

Takehiko Nagakura

Associate Professor of Design and Computation

## **Acknowledgments:**

First of All, I would like to thank my thesis advisor, Professor Takehiko Nagakura, for all his inspiration and trust along the way. This adventure would not have been possible without his support and foresight.

I would like to thank Professor George Stiny for joining my committee as a reader. His insights give me an excellent opportunity for retrospection.

Thanks to Fan for his friendship, and our conversation helps to shape a big part of this thesis.

Thanks to Carlos and NJ for their friendship and confidence in me, and I'm always encouraged talking with them.

I would like to thank the computation group, including Professor Terry Knight and Professor Larry Sass, for the incredible two years at the MIT.

More than anybody else, I wish to thank my parents and my fiancée Yu for their silent support. They are always by my side for all my decisions. I appreciate that so much.

## **LIST OF CONTENT**

<b>1 Introduction</b>	<b>8</b>
<b>2 Backgrounds</b>	<b>13</b>
2.0 Space and Architectural Design	
2.1 Stages of human vision	
2.2 Isovist	
2.3 Machine Learning	
<b>3 Methodology</b>	<b>33</b>
3.0 Machines' perception of space through machine learning	
3.1 Space Sampling	
3.2 Training Data Acquisition	
3.3 Space Composition Classification	
3.4 Scene Classification	
3.5 3D Reconstruction of Space through Element Segmentation	
3.6 Space Calculation Based on Auto-encoder	
3.7 Space Rating System	
<b>4 Applications</b>	<b>72</b>
4.1 Interactive Modeling Based on Space Awareness	
4.2 Analysis of Existing Buildings	
<b>5 Discussion</b>	<b>98</b>
<b>6 Appendix</b>	<b>102</b>
<b>Bibliography</b>	<b>106</b>

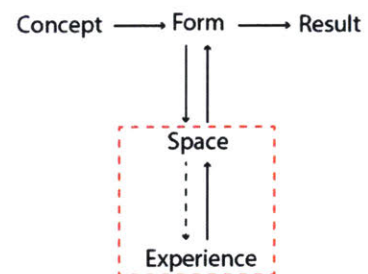


# 1 INTRODUCTION

Architectural design is human-centric. A key component is the understanding and experiencing of space, which is usually brought into play by humans. For example, walking on Broadway in New York City, one can feel stressed due to the height of the high-rise buildings and narrow streets. In designing an architectural masterpiece, each wall and column is placed deliberately to form a harmonious spatial experience. When a nice piece of furniture is placed carefully in the corner of a room, it creates a unique spatial influence that would not exist otherwise.

Unlike a machine that is designed only by following specifications, architectural space is harmonious, easy to use and provides creative opportunities thanks to the engagement of a human's subjective perception. Architecture is not only a design-by-program process by arranging individual elements (such as walls and columns), it is also a design-by-experience craft. An architect will have to "orient" himself/herself in this architectural space to acquire experience, either in their imagination or in a model, and redesign the whole space and thus the experience according to their subjective judgment of it.

However, since architectural design is human-centric, and almost every design step involving subjective perception is human-based, its efficiency and economy can hardly be improved, even in the era of digital revolution. Compared to the design tools dating back to medieval times<sup>1</sup>, the current computer-aided design (CAD) tools in the industry are still within the concept of "pens and paper," but with a digital interface. Although computers have a good grasp of the geometrical properties of architectural elements, they tend to have difficulty defining space, which is ambiguous. With the use of computers, it is possible to obtain statistical reports and cost estimates for a construction project, including the details of where and how many nails are used in a building. Even with a generative design system, a building fulfilling certain numerical objectives can be "computed." Nevertheless, it is hard to numerically describe a building's spatial experience, such as private vs. public, dynamic vs. static, conforming vs. inconsistent, interesting vs. dull. This limitation results in two dead-ends: human-friendly architecture takes a "traditional" low-efficiency process to design, yet "computer generated" buildings are hardly human-friendly.



**Figure 1-1:** Form and experience design of an architectural design

<sup>1</sup> "The History of Blueprints - PlanGrid Construction Productivity Blog." 12 Apr. 2016, <https://blog.plangrid.com/2016/04/the-history-of-blueprints/>. Accessed 20 May. 2018.



Is it possible to find a mid-ground, a solution where designs with a satisfactory spatial experience can be produced more efficiently with the aid of computers? I believe the key lies in developing machines' perception of architectural space. Rooted in a human's perception of space, a computer system can be designed and attempt to achieve some human-like perception tasks. If a computer is capable of reading, computing, and operating space to some extent, CAD tools can help designers with an extra dimension: the perception of space. With a computer that has space awareness, a designer can easily and more efficiently evaluate the spatial quality of a design, can iteratively get feedback in every adjustment of the design process, can reconstruct the semantic model of a scene, and can even uncover unprecedented design opportunities, as the nuance of experience can be evaluated more efficiently once it can be computed.

Human perception gains an understanding of a scene through vision and awareness. According to David Marr's hypothesis about human vision (Section 2.2), from a computation point of view, "when a human interprets a visual scene, the brain first creates a '2.5-D sketch' of the objects it contained — a representation of just those surfaces of the objects facing the viewer. Then, from the 2.5-D sketch — not the raw visual information about the scene — the brain infers the full, three-dimensional shapes of the objects." The subjective feeling and personal experience of space, which eventually helps to shape design, is then inferred from this geometrical awareness.

In that sense, there are two key processes the human brain undergoes in understanding the spatiality of space: (1) Automatically filter out visual noises and construct the "2.5-D sketch" of space that represents the surfaces of the objects facing the viewer. (2) From the "2.5-D sketch," infer the subjective experience and feeling of space or, in other words, high-level features of space. From my point of view, they are also the critical processes in building machines' perception of space systems – systems that have similar behaviors to a human's perception of space.

Isovist (Section 2.3), or an observation point along with the region can be seen from the observation point in space, is a computation method mainly used in sight line analysis. Benedikt concluded that some computation methods with the low-level features of Isovist, such as area, perimeter, and complexity, can be applied. However, isovists can also be applied in acquiring the panoramic depth of space, which is the geometric measurement of a "2.5-D Sketch", without involving the human eye and brain. By measuring the distances from the viewpoint to every point on the boundary of its isovist, a panoramic depth sample

can be acquired. Some isovist backgrounds are introduced in Section 2.3, and the panoramic depth sampling method is illustrated in Section 3.1.

The recent breakthroughs in machine learning and deep learning (or artificial deep neural network) in the field of computer science and artificial intelligence brought new possibilities to other fields (Section 2.4). Deep learning has proven useful in areas involving language processing<sup>2</sup> and computer vision<sup>3</sup> as it has excellent performance in high-level feature extraction. Artificial deep convolutional neural networks (DCNN) have been proven to have a performance similar to human vision in certain contexts<sup>4</sup>. High-level features can be extracted from input vision data by running through a deep multi-layer network and eventually forming a latent vector or feature vector that embeds abstract concepts related to the system's training targets. This ability allows for exploration of high-level or awareness-related feature extraction of a visual input, such as the panoramic depth image of a space.

It is very likely that simulating actual human perception of space will not be possible in the near future, but it is possible to simulate its behavior by considering it as a prediction problem in machine learning. By sampling the depth of space carefully using the specially developed methodology based on isovists and applying machine learning algorithms on sampled datasets, systems can be trained to make preliminary recognition, prediction and rating of a space from its geometric aspects.

In Section 3, based on the experiments conducted, systems that gain human-like perceptions of space so that they can be applied to help with spatial design and analysis are proposed. The proposed systems simulate behaviors including space composition classification, space scene classification, space 3d reconstruction, space rating and algebraic operations of space. These aspects cover topics ranging from pure geometrical understandings to semantic reasoning and emotional feelings of space.

In Section 4, using case studies, the proposed systems are examined in two ways. Firstly, they are applied to a real-time space evaluation modeling interface, which gives a user instant insights and modeling suggestions about the scene being constructed. Secondly,

---

<sup>2</sup> Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013.

<sup>3</sup> Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

<sup>4</sup> Kheradpisheh, Saeed Reza, et al. "Deep networks can resemble human feed-forward vision in invariant object recognition." *Scientific reports* 6 (2016): 32672.

they are utilized in the spatial analysis of existing architectural designs, namely small designs by Mies van der Rohe and Aldo van Eyck. The case studies validate that this methodology works well in understanding local spatial conditions, and it can be helpful either as a design aid tool or in spatial analysis.



## 2 BACKGROUND

### 2.1 Space and Architectural Design

"Space" started to become a major concern in the field of architecture since the forming of modern architecture when the development of structural engineering gives architects more freedom in creating spatial experiences in comparison to older times when space had to follow the forms of its physical structure more strictly. "Space," following other terms such as "Design" and "Form," became one of the fundamental concepts of modern architecture.

In 1893, August Schmarsow<sup>5</sup> first suggested the word "raum" ("space" in German) as the key of architecture design and proposed the concept of "spatial construct," and it is a major landmark in the study of architectural space. However, the emergence of "space" is more than a coincidence, but a consequence of multiple factors. Different concepts and theories related to "Space" and "Design" had been proposed and discussed over the last two centuries.

From Adrian Forty's discussion in the book "Words and Building: A Vocabulary of Modern Architecture,"<sup>6</sup> there were primarily three types of theories about space at the time:

- (1) Consider space as a form of "enclosure." It was first proposed by Gottfried Semper in 1852. This theory also emphasized the element of enclosure — walls. This concept was widely accepted by practice architects and was rooted more in the "physicality" of space.
- (2) Consider space as a form of "continuum," and see interior and exterior space connected and extendable. Instead of the "physicality" shown by "enclosure," the "continuum" space show more of "spatiality."
- (3) Consider space as an "extension of the body." It is originated from Schmarsow's theory and suggests that one feels its space by the imaginary extension of the body in a volume. This concept was later developed by Siegfried Ebeling that see space as a "membrane" between a human and the exterior world, or a "field" that affects the experience following the physiological feeling.

For a typical architecture space research, the concern lies in the compromise of the two concepts about space: One is the "physicality" of space rooted on practice and physical

---

<sup>5</sup> Schmarsow, August. "The essence of architectural creation." *Empathy, Form and Space—Problems in German Aesthetics* 1893 (1873): 125-148.

<sup>6</sup> Forty, Adrian, and Adrian Forty. *Words and buildings: A vocabulary of modern architecture*. Vol. 268. London: Thames & Hudson, 2000.

elements, meaning space is a specific object in real world which has its scale and dimension, so that it can be operated by architects; The other is the abstract experience about space or the "spatiality" of space which is evolved from the philosophical and conceptual awareness, suggesting that space is a feature of human awareness, a mean of human perception. In that sense, space is a coin that has two sides. However, from another point of view, "physicality" and "spatiality" are two closely integrated parts that cannot be separated, they are both concrete features of space. The "spatiality" of space is derived from "physicality" through the participant of human perception.

The two meanings of space are very similar to the two strategies of architecture design. One is a form oriented, design by program approach that design is preceded by configuring the patterns, functions, proportions, layouts, and rules of physical elements; The other is an experience-oriented design approach that design is undergone by crafting the human-centric space and spatial experience, which has much in common with the "spatiality" space theory. The latter considers that architecture is not standing by itself but interacts with its users, while the former forms design by only following rules about physical elements.

A good example of the design by program strategy can be seen in "The four books of architecture" by Palladio<sup>7</sup>:

*Of the loggia's, entries, halls, rooms, and of their form. The loggia's, for the most part, are made in the fore and back front of the house, and are placed in the middle, when only one is made, and on each side when there are two... The rooms ought to be distributed on each side of the entry and hall; and it is to be observed, that those on right correspond with those on the left ...*

In Palladio's description, an architecture is designed through the definition of specific element rules, how everything fits together and functions together. That makes this process more concrete and shares significant similarities with the "physicality" concept of space, where space can be manipulated like an object.

Design by experience is another primary design strategy in practice. In this process, the interaction between space and humans is a significant part of the design. Space is not only integrated and connected as a whole, but it is also like an extension of the human body. How a human feels a space -- by vision, by smell, by touch -- and every possible sensory of

---

<sup>7</sup> Palladio, Andrea. *The four books of architecture*. Vol. 1. Courier Corporation, 1965.

him/her becomes a contributing part of the design. That is where architecture becomes harmonious, easy to use and full of creativity. Tadao Ando described his design like this<sup>8</sup>:

*When I design buildings, I think of the overall composition, much as the parts of a body would fit together. On top of that, I think about how people will approach the building and experience that space.*

Architecture design is not only a process of forming an overall composition, but it is also a human-centric craft shaped by experience. An architect will have to “orient” himself/herself in space to acquire subjective experience, either in imagination, with a sketch, or in a model, and redesign the whole space and thus the experience according to the subjective perception of the environment.

In reality, an architectural design process is more like the combination of the two strategies, is both about form and space. In his book *Architecture: “Form, Space, and Order”*<sup>9</sup>, Francis D.K. Ching stated the relationship between form and space:

*Space constantly encompasses our being. Through the volume of space, we move, see forms, hear sounds, feel breezes, smell the fragrances of a flower garden in bloom. It is a material substance like wood or stone. Yet it is an inherently formless vapor. Its visual form, its dimensions and scale, the quality of its light—all of these qualities depend on our perception of the spatial boundaries defined by elements of form. As space begins to be captured, enclosed, molded, and organized by the elements of mass, architecture comes into being.”*

According to Stiny’s shape grammar theory<sup>10</sup>, forms are identified through human vision, and different forms can be “embedded” to a single pattern. In that sense, considering its “physicality” feature, space can be considered as a special type of form. This form represents the volume enclosed by the other forms, and can be “embedded” or “fused” to other forms through human perception (vision).

The process of “embedding” and “fusing” space from other enclosing elements by itself is objective and deterministic, as it is purely a rule-determined geometrical operation —

---

<sup>8</sup> "Spotlight: Tadao Ando | ArchDaily." 13 Sep. 2017, <https://www.archdaily.com/427695/happy-birthday-tadao-ando>. Accessed 20 May. 2018.

<sup>9</sup> Ching, Francis DK. *Architecture: Form, space, and order*. John Wiley & Sons, 2014.

<sup>10</sup> Stiny, George. "Introduction to shape and shape grammars." *Environment and planning B: planning and design* 7.3 (1980): 343-351.

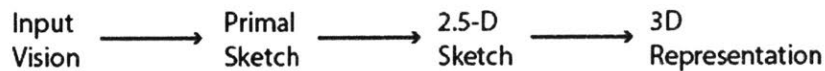
capturing the spatial boundaries used to enclose the volume. However, the experience and perception of space is human-centric and is non-deterministic. It is very likely that the same space volume will create a different spatial experience to different observers, even to the same observer but in a different time. However, there might be some statistics or rules that can suggest how experience is related to the volume of space.

As illustrated above, architectural design is the combination of form operation and experience design. Based on an architect's subjective perception, he or she can "embed" and "fuse" space from and to form, switching between the form-oriented design by program approach, and the space-oriented design by experience approach, crafting a human-centric space by manipulating physical elements. This process is undoubtedly closely connected to one's subjective understanding of the spatial design.



## 2.2 Stages of Human Vision

The book by David Marr "Vision: A computational investigation into the human representation and processing of visual information"<sup>11</sup> had a key role in the beginning and rapid growth of computational neuroscience field dates back to 1979. Marr described vision as a proceeding from a two-dimensional visual array (on the retina) as input to a three-dimensional description of the scene as output.



**Figure 2.2-1:** Stages of Human Vision, David Marr, "Vision," 1982.

According to Marr's stages of vision theory, when a human interprets a visual scene, he first sees a primal sketch of it. The primal sketch is a mental representation corresponding to local features of the stimulus. Light enters the retina and generates a patchwork of oriented edges, bars, ends, and blobs. Computationally, it can be useful to think of the primal sketch as a pixel array. That pixel array has not yet been unified with each other to generate a coherent representation of the entire scene so that it will not encode the stereoscopic depth information.

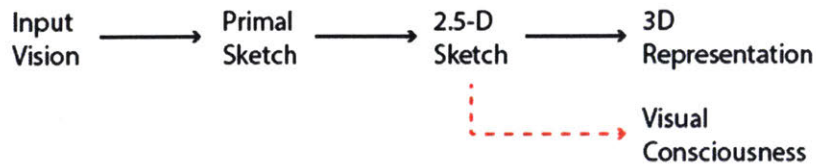
At the next stage of the process, the brain creates a "2.5-D Sketch" of the objects it contained — a representation of just those surfaces of the objects facing the viewer. The 2.5D sketch represents that in reality we do not see all of our surroundings but construct the viewer-centered three-dimensional view of our environment. This representation unifies the pixels into a coherent representation of the scene's boundaries. It represents the textures of the surfaces, separates figures from the ground, and uses shading and stereoscopic information to infer the information about the depth of spatial boundaries. Unlike the primal sketch, it does represent the bounded contours of objects, but it represents those objects from a specific vantage point. So in this sense, every time a human encounters the same object from a different vantage point, he or she ends up with a different 2.5-D sketch.

In the last stage of the process, the vision system needs to determine that the representations from different viewpoints are images of the same object. For that, Marr supposes that the vision system generates structural descriptions. The brain uses information in the 2.5-D sketch to infer what three-dimensional forms comprise the object

---

<sup>11</sup> Marr, David. "Vision: A computational investigation into the human representation and processing of visual information. MIT Press." *Cambridge, Massachusetts* (1982).

that is currently being perceived. The resulting representation is a 3D model, and this representation captures the entire three-dimensional structure of the scene, rather than merely capturing depth information from a single viewpoint. The brain then stores the 3D models in memory and matches these stored models against the models generated by what we see at any future moment.



**Figure 2.2-2:** Arise of visual consciousness

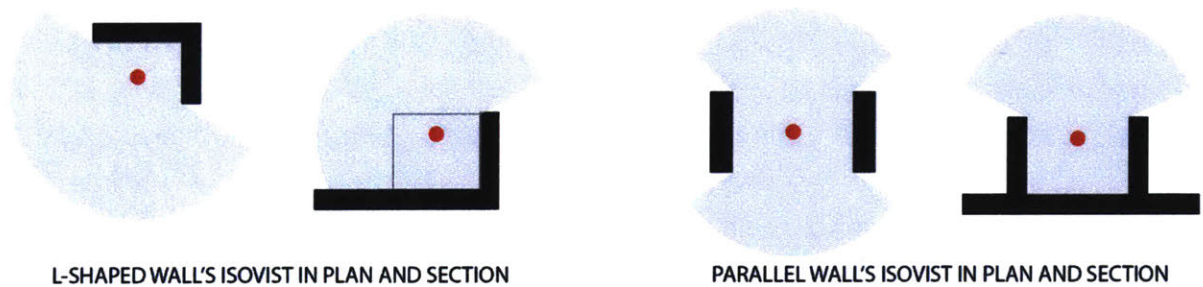
With Marr's theory in mind, Ray Jackendoff also discussed possible outcome of consciousness in visual processing<sup>12</sup>. According to his theory, we do not have a visual experience corresponding to the primal sketch, but only edges, blobs and other flat disunified jumble. Nor do we have a visual experience corresponding to the 3D model stage. Since 3D models are invariant across viewpoints, they abstract away local features that are only presented from the specific points of view. Of Marr's three levels, only the 2.5-D sketch corresponds to consciousness. From the 2.5-D sketch, we consciously experience a world of surfaces and shapes oriented in different ways at various distances from us. From Jackendoff's point of view, visual consciousness arises at a level of processing that is neither too specific, nor too abstract. It arises at an intermediate level, a level that is between discrete pixels and abstract models.

---

<sup>12</sup> Jackendoff, Ray. *Consciousness and the computational mind*. The MIT Press, 1987.

## 2.3 Isovist

An isovist<sup>13</sup> is a computation method mainly used in studying space compositions and sightlines. Benedikt first formally described the concept of isovist in 1979. An isovist is a set of all points visible from a given vantage point in space and with respect to an environment. The shape and size of an isovist are liable to change with position. A 2D isovist captures the essential boundary of a 2D plan. Similarly, a 3D isovist, acquired by projecting rays spherically, captures the spatial boundary of a three-dimensional space composition from an observer's perspective. The figure below shows the isovists of two different spatial compositions, in plan and section. The isovist, in gray shades, represents the differences of spatial boundaries defined by different elements.

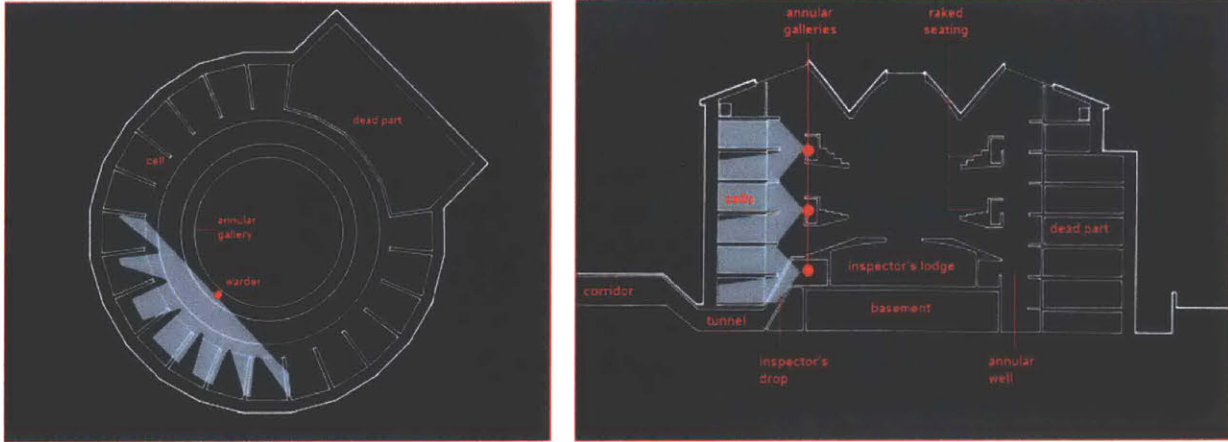


**Figure 2.3-1:** Isovists of two different spatial compositions, in plan and section

The concept of isovist was widely utilized long before Benedikt's documentation about it in 1979. Dates back to as early as 1791, when Jeremy Bentham proposed the Panopticon project, he used the concept of isovist to conduct sightline analysis on the prisoners in cells and preachers outside<sup>14</sup>. Moreover, in most practices long after that, isovist was used in sightline analysis.

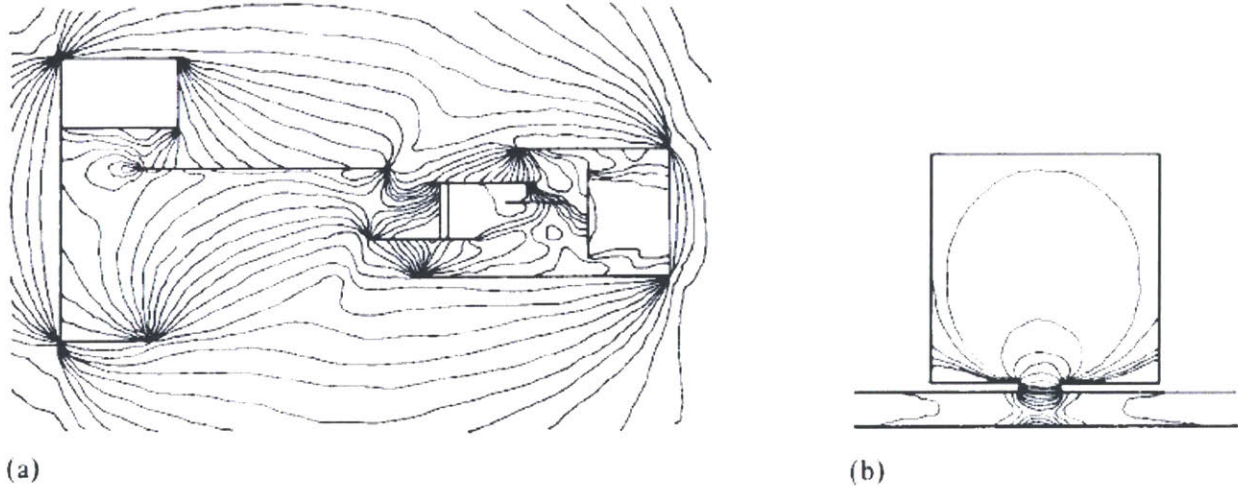
<sup>13</sup> Benedikt, Michael L. "To take hold of space: isovists and isovist fields." *Environment and Planning B: Planning and design* 6.1 (1979): 47-65.

<sup>14</sup> Steadman, Philip. "The contradictions of Jeremy Bentham's Panopticon penitentiary." *Journal of Bentham Studies* 9 (2007): 1-31.



**Figure 2.3-2:** Isovist applied in Jeremy Bentham's Panopticon penitentiary. (Source: Steadman, P., 2007.)

Aside from the definition of "Isovist", Benedikt also proposed some applications of isovist in spatial analysis. The major application is the concept that he calls isovist field. Different visual representations of a given space can be generated using different measurements of the isovist samples. The measurements he proposed include: (a) The area of the isovist. (b) The real-surface perimeter of the isovist. (c) The occlusivity of the isovist. (d) The variance of the radials. (e) The skewness of the radials. (f) The circularity of the isovist.



**Figure 2.3-3:** The drawing of the isovist field measured by "Area" for Barcelona Pavilion designed by Mies, and for a room off a hallway. (Source: Benedikt, 1979).

These isovist fields are easy to get, as it is required to merely sample through every observation point in a model, compute the measurement values, and plot them back to the model. They can be helpful as they allow people to see additional layers of the geometric

space, they represent space in a better way compared to the raw isovist samples or the model being sampled, and bring in regional visual representation about a specific measurement of the space.

However, these traditional isovist fields can be of limited help to design, which is concerned with experience over statistics. The isovist fields are plots of measurements based on low-level handcrafted features of isovist samples. These measurements may be related to experience to some extent, but it is certain that the relationship cannot be linear. For example, the publicness of a space might have some relationship to the openness of it, which is computed with the area of the isovist. But human's feeling of publicness hardly only relies on how large the volume is, but the affordance of it. A corner space and a corridor space may have the same openness measurement, but a corridor space is usually considered more public.

Isovist is a deterministic representation of spatial boundaries since Isovist is purely geometry oriented. One thing to notice is that an Isovist only captures the depth information of the surrounding space compositions from the viewpoint. Other features of the context such as material, texture, and lighting condition are ignored by an Isovist. It is the form of representation that only keeps the spatial boundaries facing the viewpoint.

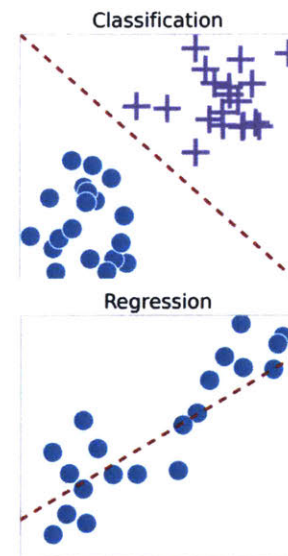
## 2.4 Machine learning

Machine learning is a data-driven approach capable of solving prediction problems through learning from past observations or historical measurement datasets. A typical prediction problem can be the prediction of the weather tomorrow, what people might purchase, would someone like a specific movie, and so on.

Machine learning can be utilized in solving multiple types of prediction problems. Generally, there are two main categories of them: supervised and unsupervised. Supervised machine learning problems are problems where predictions are made based on a set of existing categorized examples. Unsupervised machine learning problems are problems where there is not a defined set of categories, but instead, the solver is expected to re-organize the data, such as clustering.

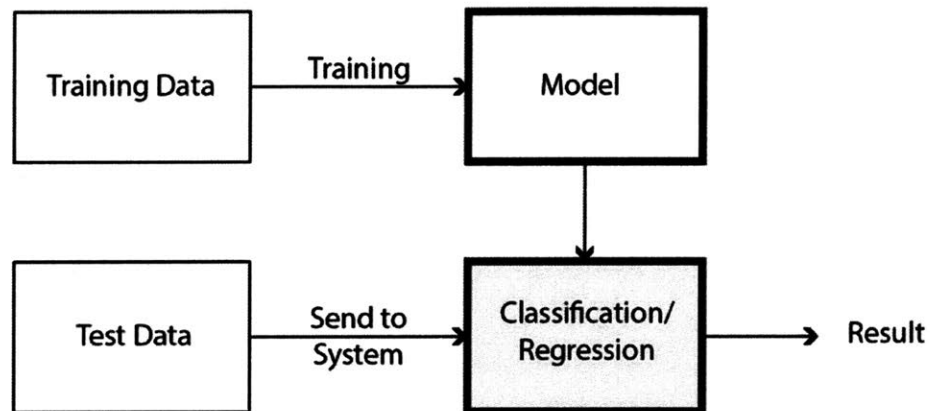
Within supervised machine learning, the problems can be further categorized into classification problem, and regression problem, as listed below:

- (1) Classification problem: A classification problem is a problem where a model is trained to predict which category something falls into, the data can include biological samples, images, texts, and so on. Usually, for a classification problem, there are already a set of predefined labels, and the system outputs a predicted probability distribution for all of the labels.
- (2) Regression problem: Regression problems, on the other hand, are problems where numerical predictions are made on a continuous scale. Examples could be predicting the stock price of a company or predicting the temperature tomorrow based on historical data.



**Figure 2.4-1:**  
Concept of  
classification and  
regression in  
machine learning

## 2.4.1 Training of a supervised machine learning system



**Figure 2.4.1-1:** Training and testing of a machine learning algorithm

Most of the data today are in an unstructured format, such as news articles, books, and albums. It is not even possible to organize all the data by hand. Automated tools are required to filter and organize the data according to our needs or preferences. It can be a tough task to do it by writing down “if-then” rules, since it is time-consuming, cumbersome, and would not perform well. Instead, it can be easier if just a small number of documents are annotated first, and a machine learning algorithm can be trained to learn the “rules” from the labels automatically, instead of writing the rules by hand. The set of an annotated dataset which is then used for training the algorithm is known as the training set. With the training set in hand, the machine learning algorithm tries to find an optimized classifier from a set of predefined classifiers that can be used to do similar tasks most precisely on new data to give. This process of optimizing classifier is the training of a machine learning model.

Unlike a typical programming task, a supervised machine learning system is developed through training the model with a large training dataset. This dataset is a collection of correctly annotated labels and is one of the most critical aspects of the success of the machine learning system. Other than a proper training dataset, the quality of the classifier set (such as the network architecture) and the algorithm used to optimize the classifier are also crucial factors for a successful machine learning system.

The input to a machine learning system to be classified can be a sample, an image, or a document. If it is shaped as an array, it can be denoted as vector  $x = [x_1, \dots, x_d]^T \in R^d$ . Possible labels of the system can be maybe dogs, cats, fish, and many other species. If there are in total  $k$  labels, one label can be denoted as a vector of  $k$  dimensions, and the

value of each position of the vector represents the status of a label. The values are binary, representing the status are either positive or negative. So dog can be denoted as  $t = [1, 0, \dots, 0] \in R^K$ , similarly cat can be  $t = [0, 1, \dots, 0] \in R^K$ .

The system that tries to classify an input is also called a classifier, denoted  $h$ . So the task of the machine learning system is given an input  $x$ , it computes an output  $h(x)$ , and that output is also the result of the prediction. Any classifier, therefore, divides the space  $R^d$  into regions that each represents a possible predicted label.

If the set of hypotheses about the rules that govern how labels are related to inputs, or a set of classifiers can be denoted as  $H$ , then the goal of a learning algorithm is to select one  $\hat{h} \in H$  based on the training set  $S_n$ , so that it would have the best chance of correctly predicting for new inputs that were not a part of the training set.

The selection of  $\hat{h}$  from  $H$  is done through the minimization of *Loss*. A *Loss* is a function to compute how close the prediction and ground truth is on training sets. A *Loss* output is larger than 0, and the smaller the value is, the closer the prediction is to its ground truth labels. For label targets in the format similar to  $t = [0, 1, \dots, 0] \in R^k$ , a function called Cross entropy loss<sup>15</sup> is utilized to computing *Loss*. Other label formats may end up using other *Loss* computing functions. The formula for computing a Cross entropy loss can be written as:

$$V(f(x) - t) = -t \ln(f(x)) - (1-t) \ln(1-f(x))$$

## 2.4.2 Feature Vectors

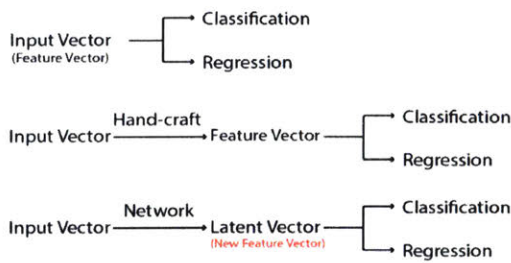
A feature vector is meant to represent the input data to be predicted in a way that the information pertaining to the prediction is more easily accessible.

A feature vector can either be the input vector itself, a hand-crafted vector dedicated for prediction, or a latent vector computed through a neural network. Directly utilize the input vector as the feature vector may not result in a good performance, as the input itself are usually formed in human-friendly formats. Because of that, usually, a new feature vector is computed for each input that has much better performance than using the input vector directly.

---

<sup>15</sup> De Boer, Pieter-Tjerk, et al. "A tutorial on the cross-entropy method." *Annals of operations research* 134.1 (2005): 19-67.





**Figure 2.4.2-1:** Different methods of computing a feature vector

A poor feature vector may lose or hide the relevant information that is important for the classifiers to learn. The more specific the critical pieces of information are in these vectors, the simpler the task is for the learning algorithm to solve. Besides that, both the data in the training set as well as the new samples to be classified should be mapped to feature vectors by the same procedure. This procedure makes sure that all cases share the same feature extraction functions.

For an image input, a typical input format representing that image is to concatenate all the pixel values (RGB value or grayscale intensity) into a lengthy feature vector or persist the image as a 2D matrix. However, this form of representation may not work well if directly utilized as a feature vector. The reason is the input by itself encodes almost nothing about the critical “features” of an image. For instance, if the task is to predict the gender of a person, hair, skin color, eyes, the shape of the face and many other features are critical for such a prediction, but none of those are already present in the raw input vector.

In a traditional computer vision approach, systems developed for image predictions often map images to feature vectors with the help of detectors of edges, color patches, different texture detectors and so on. The result of these detectors can then be concatenated to the original feature vector and use the resulting feature vector for prediction.

Modern computer vision algorithms learn the features from the images along with solving the classification task with artificial neural networks, other than the traditional way of computing handcrafted features. A neural network is trained end to end with the classifier so that the feature detectors can be acquired directly through training. Recent works proved that using a deep network, or multi-layer network, has been an effective approach in feature extraction<sup>16 17</sup> for objects and scenes. Compared to traditional computer vision

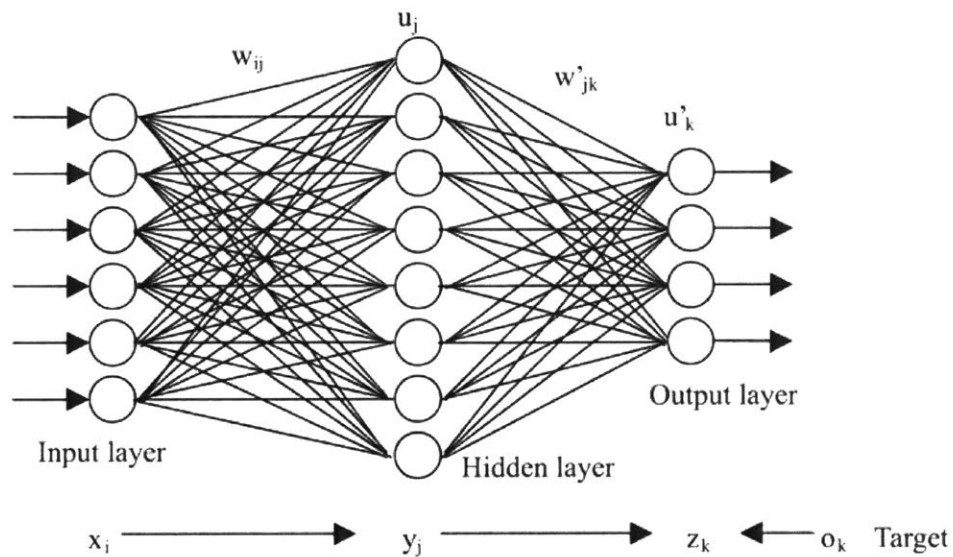
<sup>16</sup> Oquab, Maxime, et al. "Learning and transferring mid-level image representations using convolutional neural networks." *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014.

<sup>17</sup> Donahue, Jeff, et al. "Decaf: A deep convolutional activation feature for generic visual recognition." *International conference on machine learning*. 2014.

techniques, a neural network approach has a significant advantage at learning features, which has proven more feasible and efficient than hand-crafted features.

### 2.4.3 Artificial Neural Network

An artificial neural network, or neural network, is a computation system motivated by how human visual system processes the signal coming to the eyes in massively parallel stages. A neural network consists of a large number of simple computational units, such as linear classifiers. These units are also called neurons. They together form a network that computes how the input vector is processed towards the final prediction decision. In a regular Neural Network, the units are arranged in layers, where each layer defines how the input signal is transformed in stages.



**Figure 2.4.3-1:** Structure of a common artificial neural network

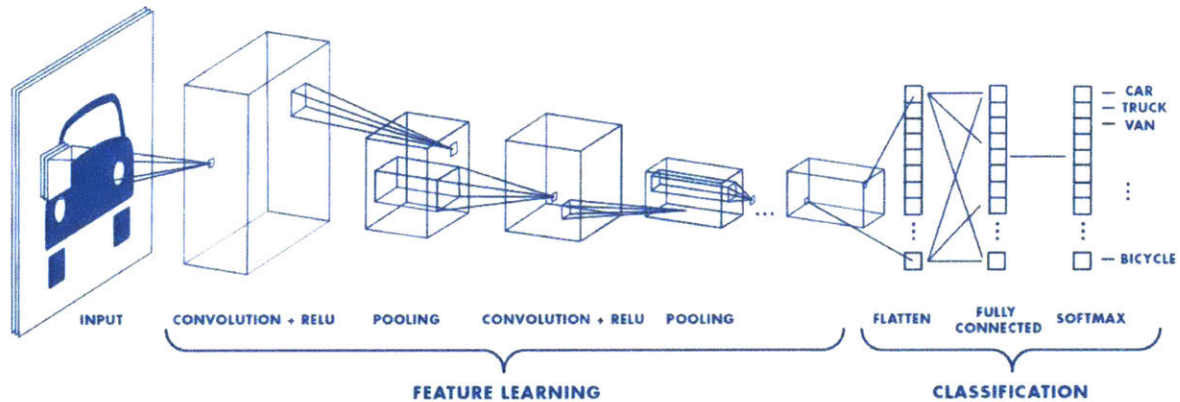
These layers can be categorized by position:

- (1) An input layer is where the units store the coordinates of the input vector. Usually, one unit is assigned to each coordinate. The input layer is special as it does not involve any computation yet, and the activation of each unit corresponds to the input coordinates.
- (2) A hidden layer represents the computation networks of the input signal from one layer to the next towards the final classifier. These units determine their activation by aggregating the input from the preceding layer output, and pass the output as the inputs of the next layer.

(3) An Output layer can be a single unit or multiple units depending on the use case of the network. It takes the output from the preceding hidden layer and computes the activation of the unit as the result of the whole network.

By function and specialized structure of a layer, it can also be named as Convolutional Layer, Pooling Layer, normalization layer and so forth.

## 2.4.4 Convolutional Neural Network



**Figure 2.4.4-1:** Convolutional neural network. (Source: <https://www.mathworks.com/discovery/convolutional-neural-network.html>)

For vision data or image data, a type of particularly designed networks, convolutional neural networks (CNN), have been proved to have excellent performance in feature extraction. CNNs are first proposed by Lecun in 1989<sup>18</sup>. A CNN consists of interspersed layers of convolution and pooling operations.

A convolution layer applies simple local image “filters” across an input image, producing new images — known as feature maps — where the “pixels” in the feature map represent how much the simple features were present in the corresponding location. This process in concept breaks the original image into overlapping little image patches, and apply a classifier to each little patch. The size of the patches, and how much they overlap, can vary. For example, a convolution operation on a 1-dimensional input  $g$ , with a filter applied to the input  $h$ , linear convolution, denoted  $\circ$ , of  $h$  and  $g$  is:

<sup>18</sup> LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." *Neural computation* 1.4 (1989): 541-551.

$$f[n] = h \circ g = \sum_{k=0}^{N-1} h[n-k]g[k]$$

In two dimensions, the process is similar. The filter can also be in 2D in this case, which is a sliding window. The filter window is flipped vertically and horizontally, then slid over the image to record the inner product with image window over the input image. Formally, this process can be written as:

$$f[m,n] = h \circ g = \sum_{k,l} h[m-k,n-l]g[k,l]$$

A pooling layer, on the other hand, abstracts away from the location where the features are, only storing the maximum or average activation within each local area. Through pooling, the network captures "what" is there rather than "where" it is. There are many different types of pooling operations, but the simplest one is called "max-pooling," which stores the activation by the maximum value in a patch. A max-pooling also computes by applying a filter across the image. So the result of the patch is just replaced by the maximum value. For a max-pooling filter sized  $k$  with stride 1, it can be written as:

$$f_i = \max_{j \in \{1, \dots, k\}} x_{i+j-\frac{k}{2}-1}$$

For  $i = 1, \dots, n$ .

A typical CNN system consists of convolution layers and pooling layers of variant sizes and strides, which ensures that the system is trained to activate for patterns of different sizes and orientations.

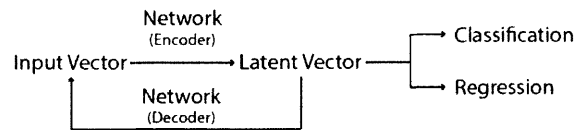
There is a new research trend on using deep convolutional neural network (DCNN) models to recognize places<sup>19</sup>. Compared with traditional computer vision techniques, a deep-learning approach has a significant advantage at learning features, which has proven more feasible and efficient than hand-crafted features. The importance of DCNNs has been well recognized due to their impressive performance on visual recognition tasks.

## 2.4.5 Auto-encoder

---

<sup>19</sup> Zhou, Bolei, et al. "Learning deep features for scene recognition using places database." *Advances in neural information processing systems*. 2014.

Within the concept of Auto-encoder<sup>20</sup>, through the training process of a Neural Network, a latent vector  $v_l$  can be computed for each input, and that is known as the encoded representation of the input, mostly used as the feature vector for prediction. This process of computing latent vector is called encoding, and this network can also be called an encoder. One of the essential features of the latent vector  $v_l$  is that the distance between different  $v_l$  represents the similarity of their original inputs in the concept of the prediction task that the system is trained for. This allows for the computation of abstract contents through the computation of latent vectors<sup>21</sup>.



**Figure 2.4.5-1:** Concept of an auto-encoder.

In the training process of an encoder, a backward network can also be trained in parallel which generates back the original input from the latent vector. This backward network is called a decoder. If the encoder and decoder are trained together with the prediction task, the network can compute not only the designated prediction task but the conversion from an input data to and from its corresponding latent vector. This "translation" function can be handy as it allows the computation of inputs with their latent vectors.

The Auto-encoder method can also be used to correlate different abstract contents. The network to be structured like a dual-end auto-encoder, so that each end of the network takes a specific format of content, such as image or text description, assuming the text is the caption of the image. In its training process, the network matches both inputs to a same latent vector. Once the training is done, in concept the network can generate captions (text) from images and can generate an image according to a text description.

Similar projects include projects working on correlations like text with image<sup>22</sup>, image with image<sup>23</sup>, image with 3D model<sup>24</sup>, and audio with image<sup>25</sup>.

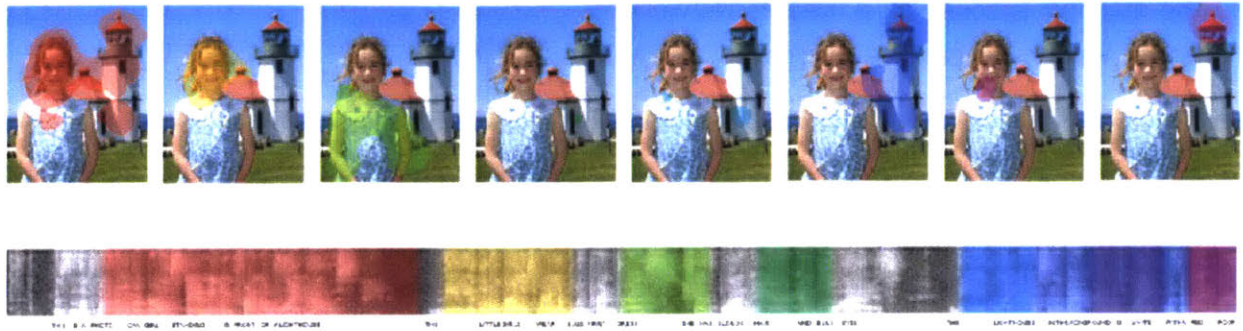
<sup>20</sup> Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *science* 313.5786 (2006): 504-507.

<sup>21</sup> Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.

<sup>22</sup> Antol, Stanislaw, et al. "Vqa: Visual question answering." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

<sup>23</sup> Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2018): 834-848.

<sup>24</sup> Chang, Angel X., et al. "Shapenet: An information-rich 3d model repository." *arXiv preprint arXiv:1512.03012* (2015).



**Figure 2.4.5-2:** Audio-Image content correlation (Source: Harwath, David, et al., 2016)

### 2.4.6 High-Dimensional Dataset Visualization

A critical problem in machine learning is that most datasets have a high number of dimensions. Visually exploring the data can then become challenging and most of the time even practically impossible to do manually. However, such visual exploration is incredibly helpful for any data-related problem. Therefore, it is vital to visualize high-dimensional datasets. To visualize a high-dimensional vector, techniques known as dimensionality reduction are widely used.

Dimensionality reduction techniques reduce the number of dimensions drastically while trying to retain as much of the “variation” in the information as possible. t-Distributed Stochastic Neighbor Embedding (t-SNE) is a widely used technique for dimensionality reduction and is particularly well suited for the visualization of high-dimensional datasets. According to its original paper<sup>26</sup>:

*t-Distributed stochastic neighbor embedding (t-SNE) minimizes the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corresponding low-dimensional points in the embedding.*

Once high dimension vectors can be “squeezed” to 2 dimensions, they then can be plotted in a 2-D space. In the 2-D plot, the closer two points are, the more similar their original high

<sup>25</sup> Harwath, David, Antonio Torralba, and James Glass. “Unsupervised learning of spoken language with visual context.” *Advances in Neural Information Processing Systems*. 2016.

<sup>26</sup> Maaten, Laurens van der, and Geoffrey Hinton. “Visualizing data using t-SNE.” *Journal of machine learning research* 9.Nov (2008): 2579-2605.

dimensional data points are. Therefore, the 2-D plot can represent the original data points and their relationships.





## **3 METHODOLOGY**

### **3.1 Machines' perception of space through machine learning**

Although the architectural design is human-centric, its efficiency and economy can be hardly improved even in the era of digital revolution, and almost every design step involving subjective perception is human based. Computational or generative design, the design approaches that are considered more efficient, ignores human-friendly qualities of space in favor of performance metrics; Yet in a regular design process, although CAD tools and other simulation software are employed, the design of space can still be time-consuming, as the design of space is purely hand-crafted. This limitation results in two dead-ends: human-friendly architecture takes a "traditional" low-efficiency process to design, yet "computer generated" buildings are hardly human-friendly.

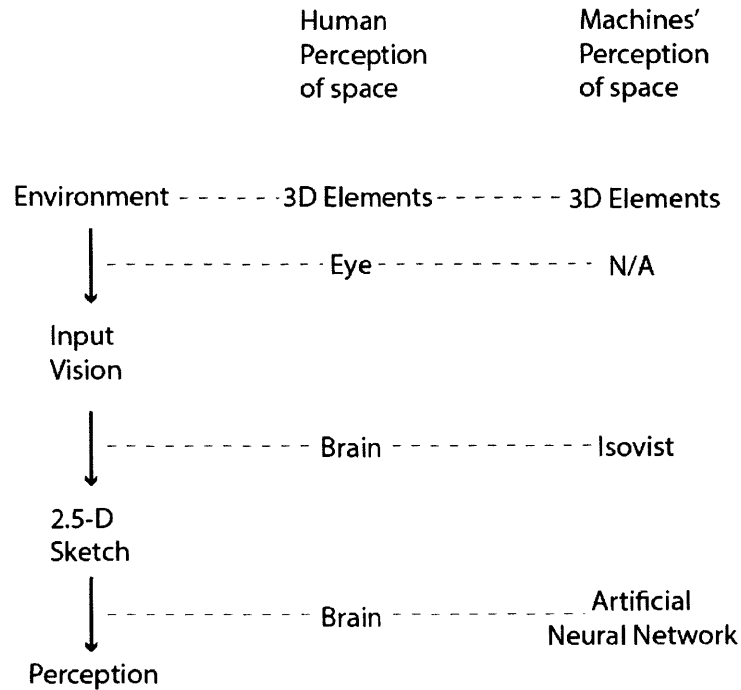
Is it possible to find a mid-ground, a solution where designs with satisfactory spatial experience can be designed more efficiently with the aid of computers? It is believed that the key lies in machines' perception of architectural space, machine systems that can simulate human's perception of space.

According to Marr's stages of vision theory, the process of human vision can be considered as a process with three stages where before interpreting a 3D structure from the "primal sketch," the brain infers a "2.5-D sketch" of the scene. Ray Jackendoff suggests that visual consciousness arises at this intermediate level where it is neither too specific nor too abstract. For an architect, what matters is not only the 3D structure, or the physicality of a scene, but also the consciousness or perception, the spatiality of the scene.

It is very likely that simulating actual human perception of space will not be possible in the near future, but it is possible to simulate the behavior of human perception. If a part of perception is considered as recognition, prediction, or rating problems, they can be solved using machine learning algorithms. Unlike a typical programming task, a machine learning system is developed through training the model with an extensive historical dataset and used to solve classification or regression problems for any new input data.

Unlike a human where a "2.5-D sketch" is acquired through eye and brain, a machine will have to use sensors such a camera or depth detector to acquire the "2.5-D sketch" of a real scene. However, it can be more straightforward if only to acquire a "2.5-D sketch" for a digital model. In this thesis, only the perception of geometric features of space is studied,

and by sampling space using Isovist (will be explained in section 3.2), a panoramic depth image of a scene can be easily acquired. The panoramic depth image is a kind of “2.5-D sketch” for the scene that captures no more than the geometric features of a space. The panoramic depth image has each of its pixel representing depth in the scene targeting one direction. This panoramic depth image can be considered as the input of a machine learning system and used to generate prediction results.

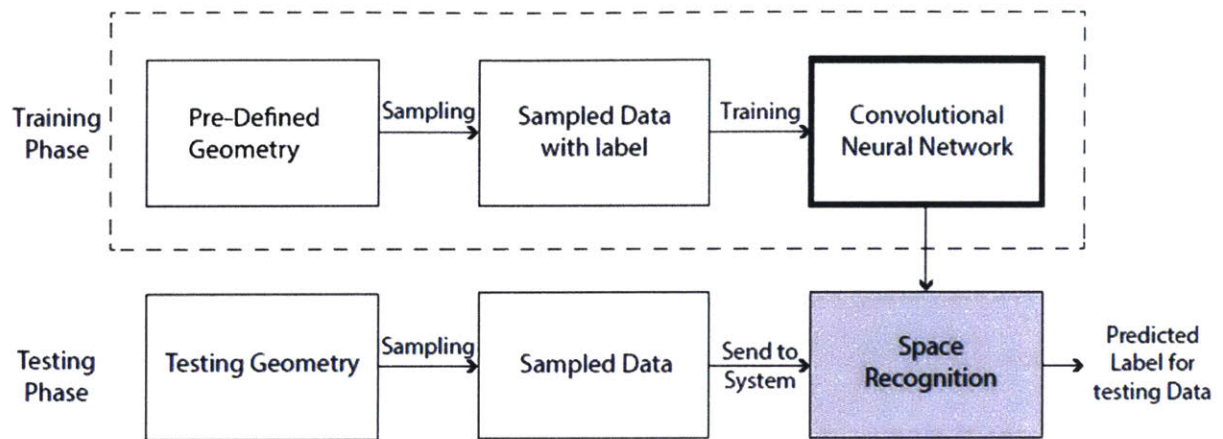


**Figure 3.1-1:** Framework of machines’ perception of space in comparison with human perception

How to utilize the captured panoramic depth image in machine learning algorithms can be a challenge. Since it is a panoramic capture spherically conducted from the viewpoint, an equirectangular projection can be applied to the spherical sample to format it as a 2-D matrix. Since the 2-D matrix dataset is derived from vision, it can be treated as a regular 1-channel image dataset, which has the grayscale pixel values representing the original depths.

Human has certain understandings of each scene, which can be formatted as labels or values of specific properties. The labels and values are chosen in certain ways that represent the behaviors of space perception in different aspects. If these labels or values are treated as ground truth target  $\square$ , and the panoramic depth image of each scene is then the input vector  $\square$ , and some of the perceptions of space problems can be formulated as

discriminative classification tasks, or regression tasks in machine learning, for acquiring the labels and values respectively (as Section 2.4).



**Figure 3.1-2:** Workflow of Training a Machine learning Model for space data prediction

Compared to developing a regular machine learning system seen in Section 2.4.1, the process of developing a space perception system is similar: both processes require the training of a system using existing data and testing new data with the pre-trained system. The difference is that before training, each space will first be sampled to get a panoramic depth image and use the sampled data for training or testing. The labels and values of the sampled data are identical to the original space. This process is illustrated in Figure 3.1-2.

Training data acquisition is an important part to consider in constructing a machine learning system. In the experiments introduced in this thesis, three dataset sources are mainly utilized: customized dataset generated by computers (Section 3.3.1), existing depth dataset found on the internet (Section 3.3.2), and crowdsourced dataset (Section 3.3.3) which is acquired through crowdsourcing.

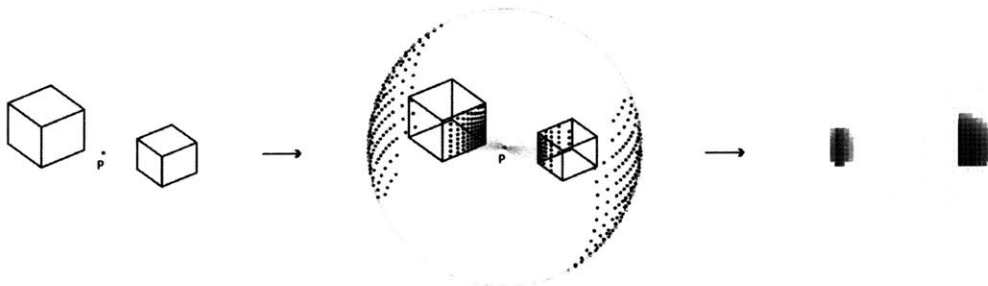
Further, several experiments are conducted that solve machines' perception of space problem as different tasks. Firstly, a machine learning system is trained with customized training dataset that achieves the recognition of local spatial compositions through classification (Section 3.4). Secondly, utilizing existing scene depth datasets, a system is developed that classifies space by scene categories (Section 3.5). Thirdly, a system is developed that achieved 3-D reconstruction of space through element segmentation, or multi-target classification (Section 3.6). Fourthly, through the supervised training of an auto-encoder, a space calculation methodology is proposed that computes space in latent vector space (Section 3.7). Lastly, trained with a crowdsourced dataset with space rating

values acquired from human intelligence, a space rating system is developed through regression (Section 3.8).

### 3.2 Space Sampling

To analyze space from a specific viewpoint, space should be captured with both essentiality and conciseness. Essentiality means the representation should contain the basic configuration of the space from the selected observation point, capturing the corresponding spatial boundaries. Conciseness makes sure that the data captured should also be simplified and can extract certain characters of the original space composition that makes the later analysis feasible. The features of essentiality and conciseness meet perfectly with the concept of 2.5-D sketch proposed by David Marr.

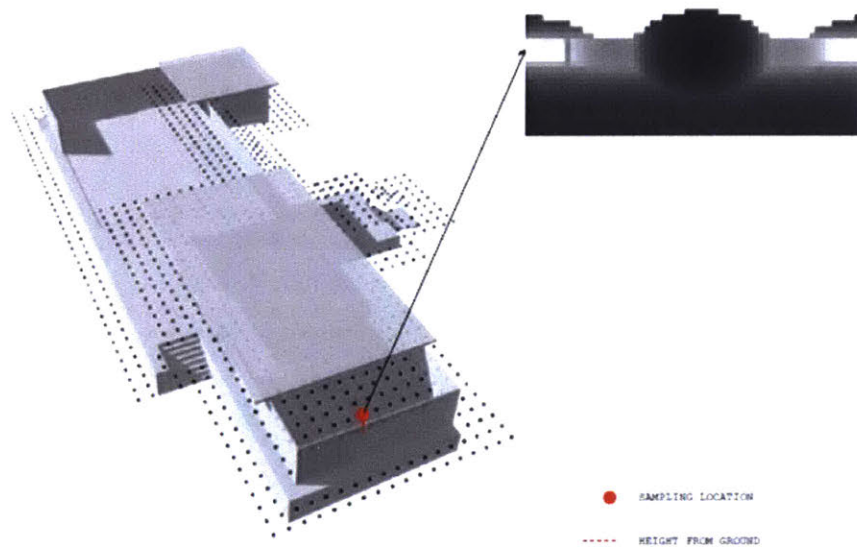
For simplicity, this thesis explores the behavior of human space perception in the aspect of geometric boundaries. Only depth information is kept in this space sampling process and in the experiments to be introduced in the following sections. A panoramic depth sampling method based on the concept of 3D isovist is proposed. By projecting the space depth information onto a sphere centered on the observation point, the spherical depth values are projected to a 2D equirectangular "image" with a resolution of 60 x 30. In the experiments introduced in this thesis, only the values of the distances from the surface of the objects to the observation point, or depth, is kept as the criteria of each pixel, which makes up a one channel image. If values other than distance (like the brightness of the environment/the RGB value of the surrounding space, and so forth) are included, they can be added as additional channels to this "image" to be utilized in the matrix calculation. The maximum sampling distance is 10 meters, which is also the radius of the sampling sphere. This sampling distance threshold ignores geometry information that is too far from the viewpoint. After the sampling process, the original spatial composition (from the observation point) can be represented as this one-channel image that contains its necessary spatial boundary (distance). This sampling process is illustrated in the figure below.



**Figure 3.2-1:** 3D Isovist Sampling of a given space from observation point P, resulting in a panoramic depth image.

The projected representation is a distorted image from the original view. Since it is using equirectangular projection, the region close to the top and bottom of the sphere is stretched

horizontally. The orientation of the equirectangular representation might also need some attention since different sampling orientations create horizontally shifted representations. However, the systems developed (introduced in the next sections) can learn the distortion and orientation of the invariant features. Also, in the experiments that orientation is not a critical factor, the image samples are shifted horizontally by centralizing the average darkest column, making sure that the closest boundary is centralized so that the most prominent features to the observation point are captured.



**Figure 3.2-3:** Panoramic depth sampling of the model of Barcelona Pavilion

### **3.3 Training Data Acquisition**

Preparing training dataset is one of the most critical steps in the training of a supervised machine learning system. It is also one of the most expensive and time-consuming parts of building a machine learning model. A training dataset is a set of example data used to fit a machine learning model using a supervised learning method. A training dataset often consists of pairs of an input vector (can be in either one dimensional, or multidimensional) and a corresponding label vector or value scalar. This label or value is also known as a target or ground truth. Within the training process, the model is run with the training dataset and produce a result in the format of the target. By comparing the result and the ground truth target, a loss is being computed; that is then used by the learning algorithm to adjust the model.

The quality of the training dataset can be crucial in training a machine learning system. Features of a suitable training dataset include:

- (1) The right quantity: If the training set has a limited amount of data, the system may be trained un-converged or overfitted. However, a dataset too large may take more time to collect yet slows down the training process.
- (2) A good variety: Training dataset with a good variety can help to prevent overfitting problem, allowing better performance on more input scenarios.
- (3) A high quality of each sample: The machine learning algorithm will learn for whatever data fed in. Typically, the samples fed in need to possess two essential qualities – independence and identical distribution.
- (4) A good relevance of data and label: If the sample and the label have irrelevant distributions, the system will not perform well in learning, as there is no feature to be learned.

Sometimes there is already a significant amount of historical data and a precise ground truth knowledge about each data sample, in which case the training dataset is already , and all that is retained to do is clean, normalize, sub-sample, analyze, and train and iterate a model until it achieves a good prediction rate. However, more often, there is only a large amount of raw unlabeled data, and the process of manually building a consistent ground truth might be the most painful phase of the entire workflow.

The quality and amount of training dataset can influence the accuracy and generalization of a machine learning system dramatically. ImageNet<sup>27</sup>, for instance, is a popular image dataset utilized broadly in the field of image classification. It consists of over 14 million images that are hand-annotated by users on crowdsourcing platform Amazon Mechanical Turk. The publish of the ImageNet dataset, and a corresponding ImageNet image classification challenge has been widely considered to be the beginning of the deep learning revolution of the 2010s. The dramatic influence of the ImageNet dataset can be a sound proof of how vital a training dataset is in developing a machine learning system.

With the 3D Isovist sampling method in hand, it is easy to sample any given space and construct a training set using the sampled panoramic depth image. However, depending on the specific needs and purposes of the machine learning systems, different training sets should be utilized. In the experiments of machines' perception of space, three dataset sources are mainly employed, including generated dataset, public open source dataset, and crowdsourced dataset. The acquisition of each is explained below in Section 3.3.1, Section 3.3.2, and Section 3.3.3.

### **3.3.1 Generated Dataset**

Using a computer to generate a large dataset is the easiest way to acquire training datasets. It is effortless to sample multiple compositions of space from various observation points with digital models. In every iteration of a generating process, the computer program first generates a random model along with a random viewpoint in that space. Based on this randomly generated combination, the panoramic depth sampling method is applied and collect a panoramic depth image of the surroundings from the specified viewpoint. This panoramic depth image is then stored with the ground truth labels formatted from the current model.

Compared to datasets collected manually, since this generating process is purely automated, the construction of a generated dataset is much more efficient, makes no mistake if properly programmed, and can be easier organized and formatted.

On the other hand, a generated dataset may also have its drawbacks and thus can be utilized in limited situations. To begin with, the variance of a generated space composition is highly dependent on the programmed rules of the generative system (completely randomize

---

<sup>27</sup> Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.



the result will result in a lousy performance in regular tasks), and that can limit the flexibility of results and may lead to overfitting<sup>28</sup> in the later training process. Additionally, the labels and values of the dataset must be insufficient and low level, as they have to be directly computed from the digital models. For instance, high-level sentiment labels such as emotional ratings of space can hardly be generated, as these values by themselves cannot be computed. Because of that, generated datasets can only be used mostly in specific form related prediction tasks. In other words, it can be only trained on prediction tasks when the correlated labels can be directly recorded.

According to the experiments conducted, generated datasets can be helpful in 3D reconstruction and space composition classification tasks. Since the computer already registered the complete 3D structure of a scene, the sampled panoramic depth images can be directly attached to its labels derived from the 3D structure.

### **3.3.2 Public Depth Dataset**

Using public datasets can also be a proper way to start with. These datasets are well organized, mostly free of charge, have a good variety of contents, and even provide their own data management tools.

The most famous public datasets that include image depth information so that can be used in developing machines' perception of space systems is the NYU Depth Dataset (V1 and V2)<sup>2930</sup>. The dataset includes frames comprised of a video sequence from various indoor scenes (over 40 different scene labels) that recorded both the RGB and Depth values using Microsoft Kinect. Recently there is also a 2D-3D-S dataset<sup>31</sup> released, and it includes more labels and information such as mesh, semantics in 2D and 3D, and surface normals.

---

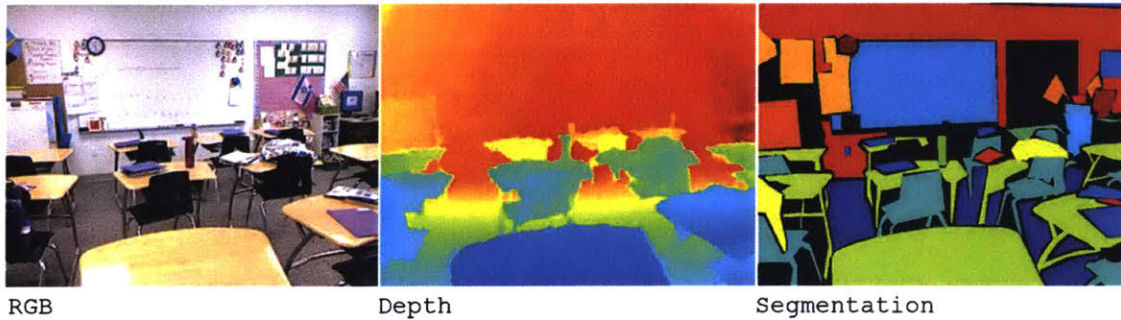
<sup>28</sup> Hawkins, Douglas M. "The problem of overfitting." *Journal of chemical information and computer sciences* 44.1 (2004): 1-12.

<sup>29</sup> Silberman, Nathan, et al. "Indoor segmentation and support inference from rgb-d images." *European Conference on Computer Vision*. Springer, Berlin, Heidelberg, 2012.

<sup>30</sup> Silberman, Nathan, and Rob Fergus. "Indoor scene segmentation using a structured light sensor." *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011.

<sup>31</sup> Armeni, Iro, et al. "Joint 2D-3D-Semantic Data for Indoor Scene Understanding." *arXiv preprint arXiv:1702.01105* (2017).

"Classroom"



**Figure 3.3.2-1:** NYU Depth Dataset. (Source: [https://cs.nyu.edu/~silberman/datasets/nyu\\_depth\\_v2.html](https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html))

However, public datasets are constructed in specific ways that they are dedicated to particular training purposes. Take the NYU dataset as an example; it is mostly designed to train machine learning models for an image to depth estimation task. It does have the scene labels, which also allows the training of scene classification from single depth images. However, in the experiments introduced in this thesis, the input is panoramic depth images. Additional work and several compromises should be made to take advantage of the NYU Depth dataset in training networks aimed for panoramic depth image inputs. Details about the approaches will be explained in section 3.5. Further, since the data from NYU datasets are real depth images taken by depth cameras, they may include many small details ranging from texture bumps to furniture pieces. The system trained with these datasets may not end up with an optimized performance on space sampled from digital models, as the models lack the details as exists in real world.

### 3.3.3 Crowdsourced Dataset

Training dataset can also be constructed through crowdsourcing. Unlike tasks that can be trained using generated datasets, building datasets for recognition tasks requiring higher level labels or values, such as rate the interestingness of a space, or tell the type of a scene, can be more challenging tasks. In that case, only human intelligence can be reliable in solving such high-level labeling and evaluation tasks, but usually with very low efficiency. With a vast amount of unlabelled data, building a consistent ground truth manually might be the most painful phase of the entire workflow.

However, online crowdsourcing makes the task much faster and feasible. With the internet and a modern browser in almost every people's pockets, running a survey is becoming more

accessible and cheaper. By hiring a large number of human intelligence from the internet, space datasets can be labeled with better efficiency and comparatively good quality.

Mechanical Turk is an online crowdsourcing platform dedicated to human intelligence service. It is a platform where requesters can easily post surveys and get results from the internet with comparatively little payment. Papers show that the Mechanical Turk participants are diverse, efficient, inexpensive, and the survey quality is no less reliable than those obtained via traditional methods<sup>32</sup>. By designing customized survey tools for specific tasks, data for almost any task can be collected through the Mechanical Turk platform.

---

<sup>32</sup> Buhrmester, Michael, Tracy Kwang, and Samuel D. Gosling. "Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data?." *Perspectives on psychological science* 6.1 (2011): 3-5.

## **3.4 Space Composition Classification**

### **3.4.1 Problem Description**

Architecture generally consists of elements such as columns, walls, shades, and windows. Although these elements are simple and common components, they can be combined to create complex and specific compositions of space. Different spatial compositions define different spatial boundaries, and therefore produce different feelings to the observers inside the space. Against a single wall, enclosed by an L-shaped wall, surrounded by columns, and with or without a shade; all these situations create unique local spatial experiences for the observers. Architects usually consider these local spatial experiences, and their sequence critical to architectural design.

While multiple representation methods are practiced in the field of architecture, there is a lack of compelling ways to capture and identify local spatial experience. Therefore, it can be challenging for architects to describe spatial experience quantitatively and efficiently.

The proposed method formulates this recognition problem as a discriminative classification task: with several predefined local spatial compositions, or what is called Seed-Spaces, a classifier is trained to predict the type of a given space in the form of a panoramic depth sample. The methodology can be applied to different settings of predefined compositions.

### **3.4.2 Workflow**

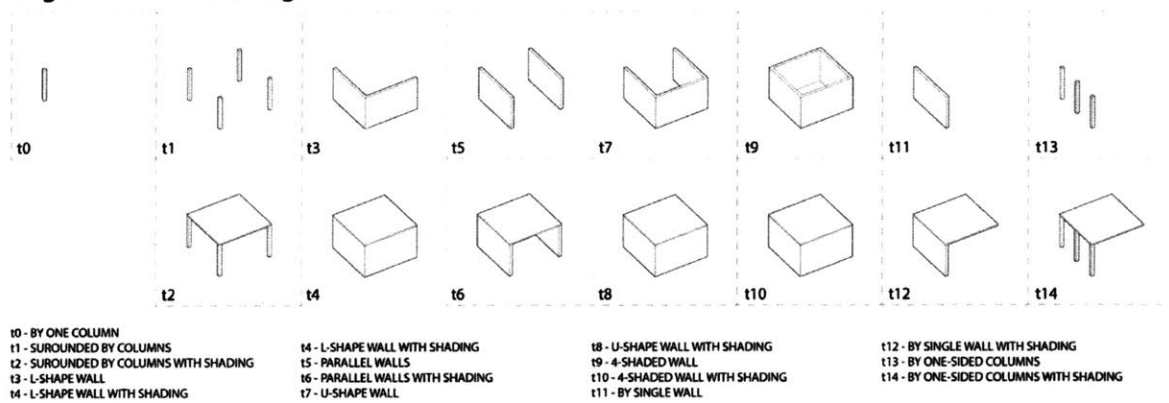
The framework of the spatial composition classification system is identical to the space machine learning system introduced in section 3.0. It consists of two phases, training, and service. In each phase, the system first performs spatial sampling, and then runs the result through the network, to either train the network or use the pre-trained network to acquire a prediction result.

In the training phase, data sampled from pre-defined Seed-Spaces are utilized to train the network. The labels of these sampled datasets correspond to the originated Seed-Spaces. In the service phase, the network presumes the most similar SeedSpace for any input data sampled from a given space using the same methodology.

### 3.4.3 Collection of Local Spatial Compositions

As stated earlier by Ching<sup>33</sup>, space can be composed of elements. Limited elements can create unlimited possibilities of space. Horizontal planes (including base plane, elevated base plane, depressed base plane, and overhead plane), vertical linear elements (like columns), and vertical planes (walls) are considered as the primary architectural space-defining elements.

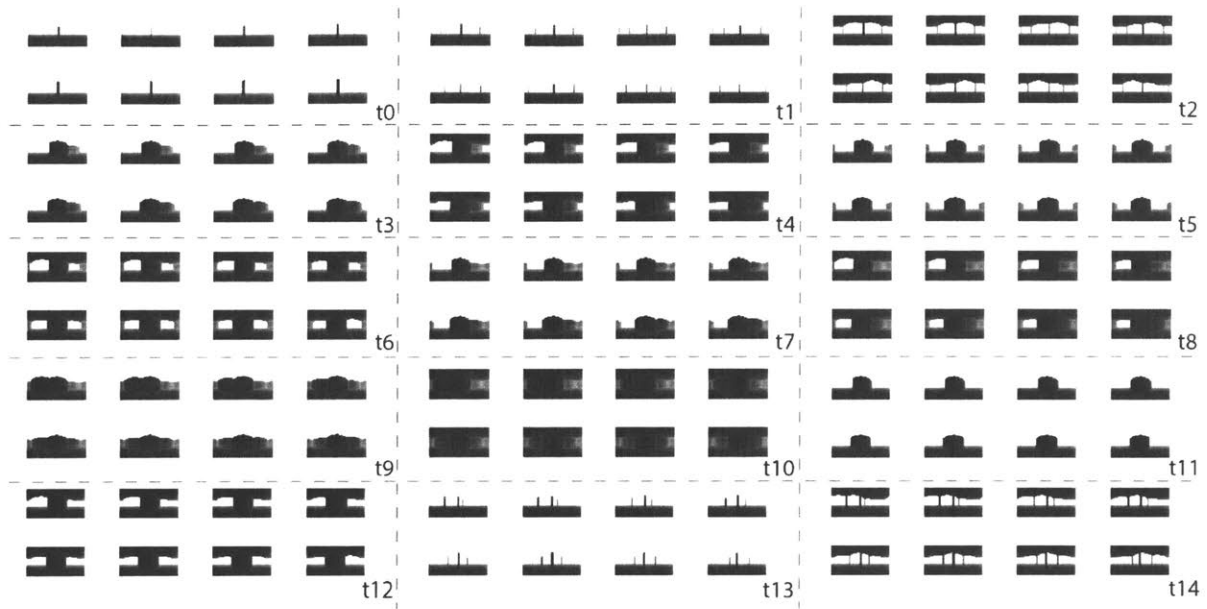
In this research, fifteen basic types of element compositions are selected to describe the local spatial conditions of one-story buildings. This collection of Seed-Spaces can be customized when dealing with different spatial conditions. These Seed-Spaces are considered to be the primary space types taken as elements to compose other spaces to be analyzed. Therefore, the selected Seed-Spaces are made up of columns, walls, and overhead planes in various ways, and they are listed in Figure 3.4.3-1, which depicts them using isometric drawings.



**Figure 3.4.3-1:** The 15 different space types, or Seed-Spaces, labeled as t0, t1, ..., t14.

Figure 3.4.3-2 shows a part of the samples of the 15 Seed-Spaces. Only the area enclosed by the space elements are sampled, as it is believed that those are the areas that best represent the local spatial condition defined by the surrounding elements. Samples of the Seed-Spaces are labeled accordingly to be constructed as the training set of the network in the training phase of the system. Meanwhile, in the service phase, the same sampling method is applied to incoming spaces, and the sampled image is used to obtain predictions from the pre-trained network.

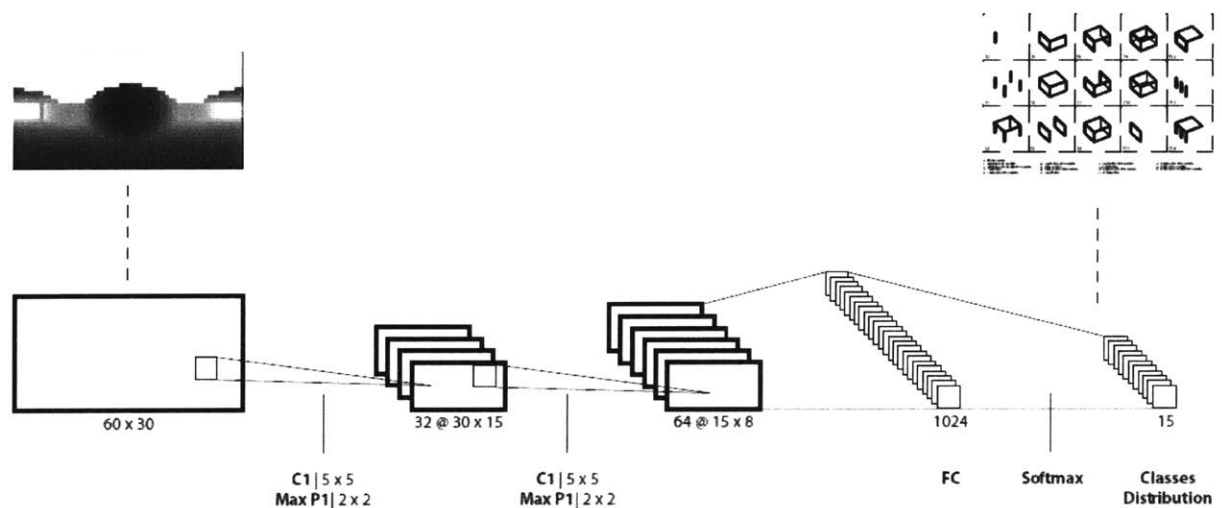
<sup>33</sup> Ching, Francis DK. *Architecture: Form, space, and order*. John Wiley & Sons, 2014.



**Figure 3.4.3-2:** A part of the space samples of the 15 different Seed-Spaces, in equirectangular format. The labels t0, t1, ..., t14 represent each individual Seed-Space.

### 3.4.4 Network

Learning data representation is the fundamental part of pattern recognition and classification tasks. In this case, learning features for a local space-type classification task is more challenging due to the inherent complexity of space and human perceptions of it. To address this issue, a CNN is employed. This method is expected to simulate humans' perceptions to particular spatial compositions in this study.



**Figure 3.4.4-1:** CNN architecture for space composition classification

Considering the situation of this task, the CNN architecture is designed based on the configuration of the input—the equirectangular dataset—as Figure 3.4.4-1 shows. The input panoramic depth image can be considered as a  $60 \times 30 \times 1$  matrix. After conducting two convolutional layers and two max poolings, a fully connected layer is extracted as the feature vector, and this vector is then used to classify the input image with a fifteen-label softmax classifier with dense connections. The network is set up with TensorFlow (Abadi et al. 2016) in the first experiment, and in developing API for the web-platform, Pytorch<sup>34</sup> is utilized.

This neural network is designed so that it takes a sampled image as input, and outputs a presumption, or class distribution, among the fifteen predefined Seed-Spaces. Once the CNN is properly trained, utilizing the training set built upon space samplings of the Seed-Spaces, it can be used to make a judgment upon a given space sample, i.e., classify the given space. The network is trained with a training set of over 5000 images for the fifteen predefined Seed Spaces. It achieved higher than 99% “Top-1” prediction accuracy on validation sets.

<sup>34</sup> Team, Pytorch Core. "Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration." (2017).

This space composition classification system has been employed and tested in the interactive modeling interface introduced in section 4.1, and the building analysis in section 4.2.



## 3.5 Scene Classification

### 3.5.1 Problem Description

Different scenes may possess different spatial boundaries. As a human, he or she can presume the type of a scene by feeling the atmosphere created by the geometric components in the scene. A bedroom, a lobby, a corridor, every type of scene has its way of organizing its geometric components, resulting in different spatial boundaries for an observer inside of the scene.

This task is to classify the type of a scene by its panoramic depth sample. In other words, given a space sample, the machine learning system predicts a scene type or several scene types that this space may belong to.

### 3.5.2 Workflow

Similar to the method introduced in Section 3.4.2, the scene classification system is also developed with a training phase and a testing phase. What makes a difference is that the training dataset utilized in this task are labeled with scene categories.

A training dataset with scene labels is hard to construct since there is a lack of well-organized open-source architecture datasets. However, public depth datasets can be an alternative to tackling this issue. Here, the NYU Depth Dataset V2, introduced in section 3.2.2, is utilized in training a scene classification system. It has collections of depth images that are labeled by scenes. The scene labels of the NYU dataset are listed below in Table 3.5.2-1.

**Table 3.5.2-1:** Scene labels of the NYU Depth V2 Dataset

office	bathroom	bedroom	kitchen	hotel_room
dining_room	living_room	office_kitchen	corridor	coffee_room
rest_space	home_office	discussion_area	laundromat	stairs
home	lab	printer_room	study_space	dancing_room
classroom	study	basement	bookstore	library
lobby	playroom	reception_room	cafeteria	reception

storage_room	indoor_balcony	computer_room	conference_room	exhibition
dinette	gym	furniture_store	recreation_room	music_room
office_dining	lecture_theatre	dining_area	mail_room	idk

However, since the depth images in the NYU Depth Dataset are non-panoramic depth images, they can only be used in training scene classification systems with non-panoramic depth image inputs. The space samples are panoramic so that it is required to resize and subsample the space samples to get multiple non-panoramic depth images and fuse the result for each subsample to get the prediction for the whole panoramic image input.

### 3.5.3 Network

In order to train a system that can predict scenes based on panoramic image inputs, first, a regular depth image based scene classification system is trained with the NYU Depth V2 Dataset. This network takes an input of a regular image and outputs a label distribution over the 45 categories.

The network used here is a variant of Resnet<sup>35</sup> which has 34 residual modules. The input has a size of 112x112x1, representing the x, y location of the pixel in the image, and the grayscale value of each pixel represents the depth in the scene. The original width-height-ratio of the images in the NYU Depth Dataset is 3:4 since they are all collected using the Microsoft Kinect. Before going through this network, they are first resized to the size of 112x112 to fit the network. After the 34 residual layers, the network outputs a distribution with a 45 sized fully connected layer. So the resulting 45-dimensional array can be computed using softmax<sup>36</sup> to get a probability presumption over the 45 labels.

### 3.5.4 Subsample and Fuse

As described in section 3.5.2, the trained network only takes regular depth images for scene classification. That is why the system requires additional subsample and fuse process to work with panoramic inputs.

---

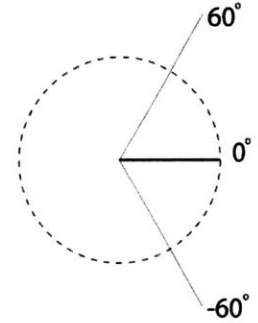
<sup>35</sup> He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

<sup>36</sup> "Softmax function - Wikipedia." [https://en.wikipedia.org/wiki/Softmax\\_function](https://en.wikipedia.org/wiki/Softmax_function). Accessed 21 May. 2018.

In this experiment, each panoramic image is sampled between a pitch angle range of  $-60^\circ$  to  $60^\circ$  with a window width-height ratio of 4:3. In width, the sampling window covers an  $80^\circ$  region, while in height, the window covers  $60^\circ$ . The sampling windows cover the full input panoramic image with a stride less than the size of the window in height and width, allowing some overlap in between. As a result, the sampling process produces 4 (in height) by 9 (in width) subsamples.

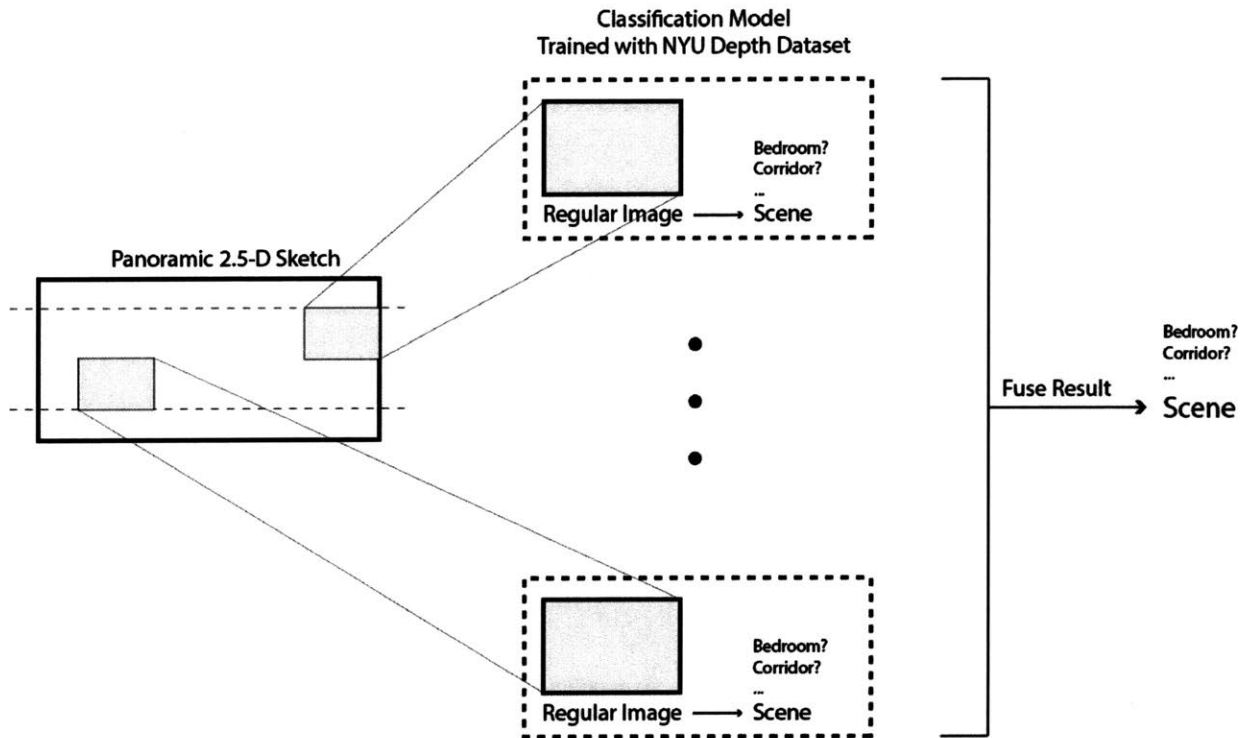
The subsamples are ran through the pre-trained scene classification network, and each produces a prediction array, then fused together to compute the final probability distribution for the labels. If the number of subsamples is denoted as  $n$ , and the number of labels denoted as  $m$ , then each subsample  $S_i$  creates a result  $[x_{i,1}, \dots, x_{i,m}]$ . If the final probability distribution is written as  $[P_1, \dots, P_m]$ , then the final probability distribution can be computed by computing each item  $P_j$ :

$$P_j = \frac{e^{\sum_{i=1}^n x_{i,j}}}{\sum_{j=1}^m e^{\sum_{i=1}^n x_{i,j}}}$$



**Figure 3.5.2-1:** Range of Sub-sampling in a panoramic image.

The illustration below shows the idea of how the system subsamples the input panoramic depth image, process individually by the neural network, and fuse the result.



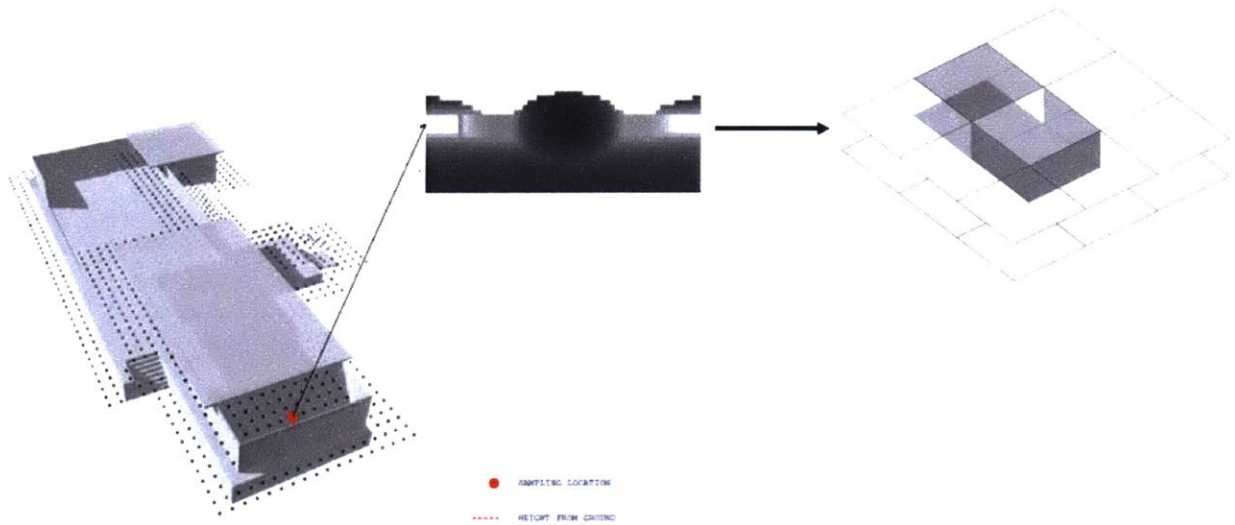
**Figure 3.5.4-1:** Illustration of the "subsample and fuse" process.

The system introduced in this section is further tested with the interactive modeling software introduced in section 4.1.

## 3.6 3D Reconstruction of Space through Element Segmentation

### 3.6.1 Problem Description

It is fundamental for a human to understand the 3D structure of a scene with vision. No matter where the viewpoint is, a semantic 3D model can be inferred in a human brain. The brain not only understands the 3D structure of the scene surrounding the viewpoint, but also the type of each element composing the 3D structure.



**Figure 3.6.1-1:** Sampling and 3D reconstruction of a space.

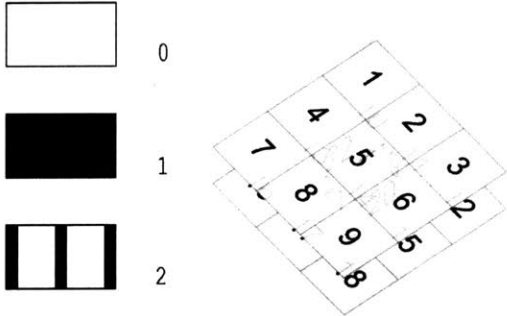
In this task, a system is proposed so that it reconstruct a scene in 3D from its panoramic depth input. In solving this reconstruction problem, for simplicity, it is formalized as a multi-target classification task. In other words, each possible location surrounding the viewpoint in space is considered as a target to be predicted by the system, and the labels to be classified for the targets include all possible element types. In this case, the element labels are arbitrarily defined, and may not apply to most space structures. The reconstructed structure is only an approximation of space structure with the designated elements. The resolution of the targets surrounding the viewpoint can be customized and fits different use scenarios.

One thing to note is, the 3D structure associated with a panoramic depth image is closely related to the orientation of that panoramic image. Horizontally shifted space depth sampling undoubtedly results in rotated 3d Structure. In concern of that, the panoramic

depth images utilized in this task are not centralized by its darkest column as in the other tasks (introduced in section 3.1).

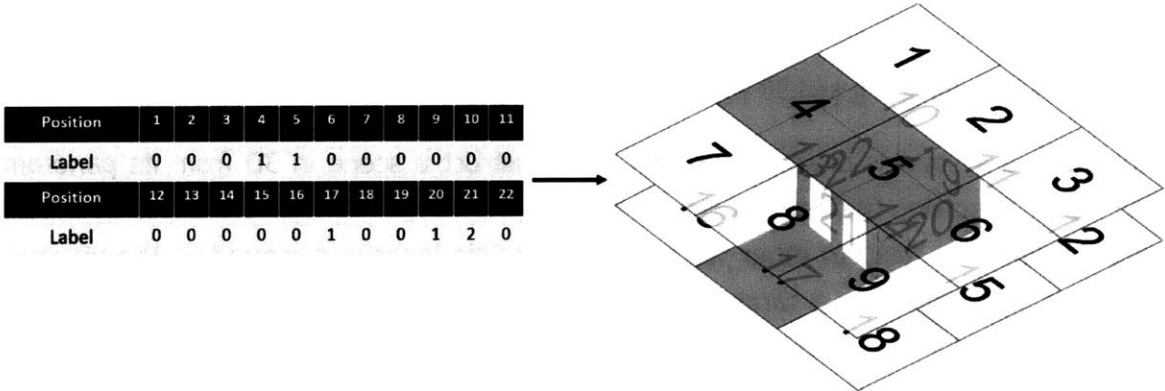
### 3.6.2 Workflow

The types of the 22 grids around the viewpoint are considered as the targets of this multi-target classification task. They are 9 shadings, numbered from 1 to 9, 9 floors, numbered from 10 to 18, and 4 vertical elements surrounding the viewpoint, numbered from 19 to 22. The viewpoint locates under shading 5, and on top of floor 14.



**Figure 3.6.2-1:** The three target states (left) and the 22 targets surrounding a viewpoint (right).

Three element states are defined, namely "none," "solid," and "frames". In this case, shadings and floors can only have two possible states namely "none" or "solid," while the vertical elements may have one of all three states.



**Figure 3.6.2-2:** An example of a 3D structure represented by a 22-dimensional vector

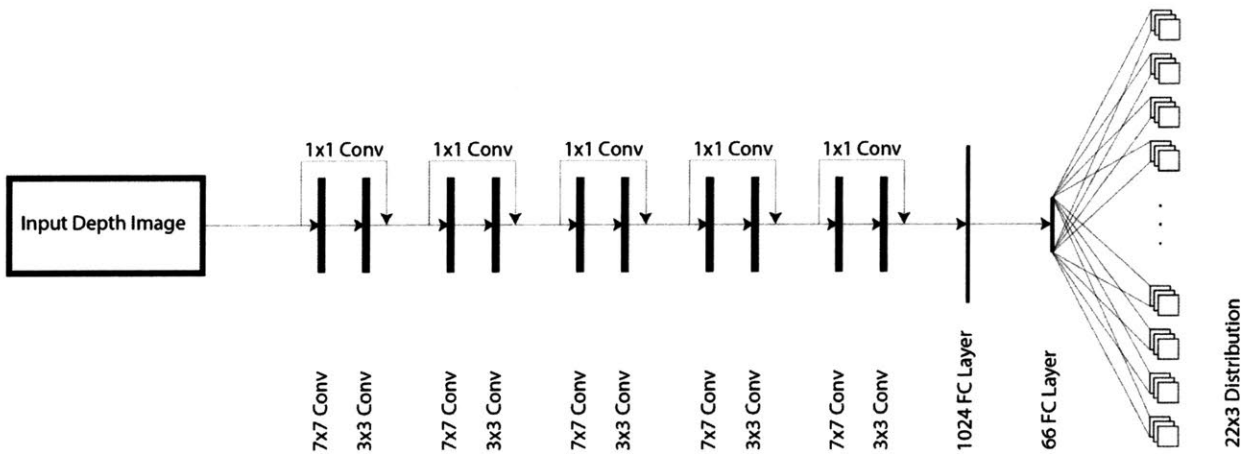
The table above lists the states of all the 22 targets, and number 0, 1, 2 denotes state "none," "solid," and "frames" respectively. In that sense, the table represents the 3D structure shown on the right: position 4, 5, 17, and 20 are walls or shades, position 21 is "frames," and all the other positions have no element.

With this task formatted like above as a 22 target three label classification problem, a training dataset dedicated to this problem can be generated. In one iteration of the training data generation process, the program randomizes elements on the 22 designated locations and samples the 3D structure from the center with a random minor shift to get a panoramic depth image, while the program records the types of the 22 elements as the labels.

### 3.6.3 Network

The figure below illustrates the architecture of the network. The network takes the panoramic depth image as its input, and pass the input through several convolutional residual modules to extract its features. Eventually, it outputs a result label distribution matrix for the 22 labels. The output matrix is of the size 22 by 3, representing the probability of 3 labels for each of the 22 targets.

The output is considered as a 22 by one 2D image output so that in the optimization process, a 2D CrossEntropyLoss function is utilized to compute the loss of the network, and the loss function computes its point-wise segmentation loss. The CrossEntropyLoss values of all 22 targets are calculated, so all targets are trained in a single process.



**Figure 3.6.3-1:** Network Architecture for 3D Reconstruction

### 3.7 Space Calculation Based on Autoencoder

According to the paper "Efficient Estimation of Word Representations in Vector Space," vector representation of words can be acquired by training a network with a large dataset. The vector representations, known as latent vectors, also provide state-of-the-art performance for measuring syntactic and semantic word similarities, and the computation of these representational vectors reflects the calculation of the meaning of the original words. The resulting vectors can be used to answer subtle semantic relationships between words, such as a city and the country it belongs to, or a word and its parent category, e.g., France is to Paris as Germany is to Berlin. If this can be written as a vector calculation or algebraic operation, it is like:

$$\text{Vector}(\text{"Paris"}) - \text{Vector}(\text{"France"}) = \text{Vector}(\text{"Berlin"}) - \text{Vector}(\text{"Germany"})$$

This calculation allows for the computation of semantic meanings through the computation of their representative vectors. Additionally, the similarity of the semantic meanings is represented quantitatively by the distance of the vector representations. With such a vector representation system, a constant relationship vector can be computed that represents the relationship between the meanings of two words, just like a "capital-country" relationship can be represented by:

$$\text{Vector}(\text{Paris}) - \text{Vector}(\text{"France"})$$

With this relationship vector, any country name can be computed from its capital name and vice versa. And it is certain that for this to work, acquiring the vector representation of semantic meaning is critical.

This vector representation oriented operation can also be useful for space. Similarity of two spaces can be represented by their distance in feature vector space; The relationship of two spaces also can be computed in advance as a constant, and utilized as a style-transition vector for computation in the future (such as the transition from columns to walls); Ideally, certain style-transition can be applied to a base space through calculation in feature vector space (such as convert the base space from a domino<sup>37</sup> style space to a Mies<sup>38</sup> style space); This is hard for a regular computer aided design system, even rule based generative design

---

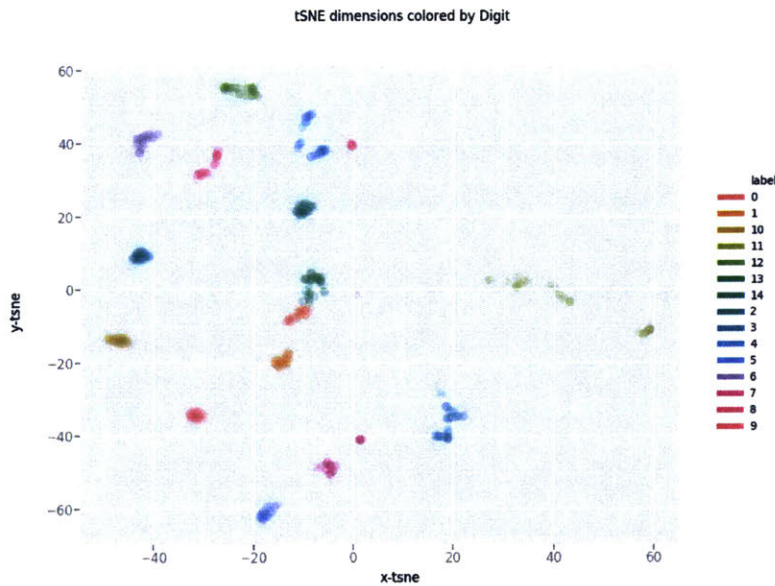
<sup>37</sup> Von Moos, Stanislaus. *Le Corbusier: elements of a synthesis*. 010 Publishers, 2009.

<sup>38</sup> "Ludwig Mies van der Rohe | American architect | Britannica.com." 25 Apr. 2018, <https://www.britannica.com/biography/Ludwig-Mies-van-der-Rohe>. Accessed 21 May. 2018.



systems. Since such high-level features can hardly be described, and too nuanced to be simulated with hand-crafted rules.

For the methods described in section 3.4, 3.5, and 3.6, space panoramic depth images are converted to feature vectors (as introduced in section 2.4.2) for classification tasks. The feature vectors are indeed the vector representations of the space sampling inputs. The vector representations for space samplings, however, can be influenced when generated with different network architectures, or the same network architecture yet different training sets, as the feature extraction process is closely related to the network, and the training procedural of the network influences the generated feature vectors. For example, the system trained for space composition classification in section 3.4 (denoted as network 3.4), is dedicated for the 15 space composition types or Seed-Spaces; Yet the network trained for 3D reconstruction in section 3.6 (denoted as network 3.6), is less specialized in the previous 15 compositions. In dealing with cases involving only simple space compositions included in the 15 types, network 3.4 is more reliable in generating a feature vector representing its input. However, in dealing with more general cases, network 3.6 will generalize better compared to network 3.4.



**Figure 3.7-1:** t-SNE plot (in 2 dimensions) of the vector representations. Each space type ( $t_0, t_1, \dots, t_{14}$ ) is marked with a corresponding color shown in the legend on the right.

By sampling the "Seed-Spaces" and ran them through "network 3.4", vector representations for each "Seed-Space" can be acquired. That makes it possible to find the similarities of the "Seed-Spaces" by comparing the corresponding vectors.

As described above, the similarity of different spaces can be considered as the distance between these feature vectors. By applying the t-SNE algorithm (Section 2.4.6) to reduce the dimension of feature vectors from 50 (in the case of network 3.3) to 2, it allows for the plotting of these vector representations in a 2-dimensional space. The visualization of the similarities between the 15 "Seed-Spaces" is shown in Figure 3.7-1. Each dot represents a projected vector, and the dots are color-coded and suggests which "Seed-Spaces" they belong to.

The visualization gives a good sense of the differences between walls, columns, spaces with shading, and space without shading. The closer two "Seed-Space" are, the more similar they are. Since the vector representations are computed utilizing features extracted by the network, they have an even distribution in the feature vector space, as the network learned the differences between different labels, and balances the distance between clusters of labels.

With a network adequately trained, it becomes possible to generate vector representations for any inputs, which is a conversion from content to vector. However, to calculate the specific content, this "forward approach" from content to vector is not enough. A "backward approach" to convert vector back to content is also required to close the loop.

In computing the previous example,

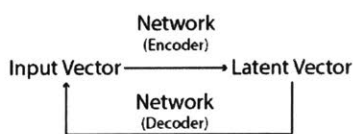
$$\text{"France"} - \text{"Paris"} + \text{"Germany"} = ?$$

As discussed in section 3.7, it can be computed in their vector space, meaning:

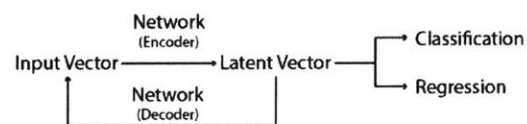
$$\text{Vector}(\text{"France"}) - \text{Vector}(\text{"Paris"}) + \text{Vector}(\text{"Germany"}) = \text{Vector}(?)$$

With a "forward approach" network, the vectors of the inputs "France," "Paris," and "Germany" all can be generated by the network, so that  $\text{Vector}(?)$  can be calculated. However, since the final target is not a vector, but the word which the vector represents, a critical process is to convert  $\text{Vector}(?)$  back to "?", and is indeed through the "Backward approach."

In order to develop a system that has both “Forward” and “Backward”, the method of autoencoder (Section 2.4.5) is introduced. The idea of an autoencoder is to train a “forward” network along with a “backward” network. The original purpose of an autoencoder is dimension reduction, meant to represent an input vector with another vector which has fewer dimensions. It is by default an unsupervised method trained with no labels, and the feature vectors learned from the unsupervised learning process are very little biased to specific tasks. In the meantime, an autoencoder can also be trained concerning labels as a supervised training task.



**Figure 3.7-2:** Unsupervised training of an autoencoder



**Figure 3.7-3:** Supervised training of an autoencoder

Figure 3.7.2 illustrates a regular autoencoder network, unsupervised trained so that it results in an encoder and a decoder, meant to be applied to the “forward” and “backward” process respectively. The encoder and decoder are trained as a whole, and the loss of training is computed by the input vector and the output vector of the Decoder. If the input vector is denoted as  $x = [x_1, \dots, x_n]$ , the output vector of the decoder is denoted as  $X = [X_1, \dots, X_n]$ , the loss function can be written as:

$$Loss = \sum_{i=1}^n (x_i - X_i)^2$$

On the other hand, Figure 3.7-3 shows a supervised training process of an Autoencoder. Unlike the unsupervised training process, this process also computes the loss of the prediction. The loss function can be written as:

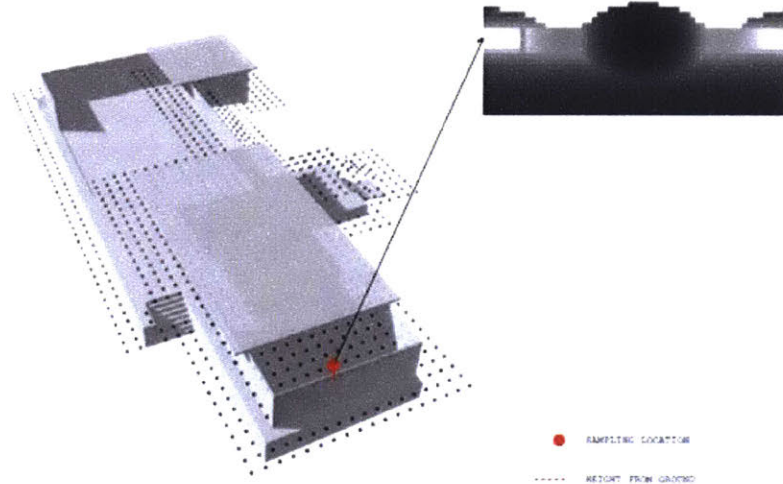
$$Loss = Loss_{Autoencoder} + \sigma Loss_{Prediction}$$

This network is trained with regard to both the autoencoder and a prediction task. The parameter  $\sigma$  adjusts the weight of the training, as it is biased towards minimizing either the autoencoder loss, or the prediction loss. In real practice, the network was trained as a two-step process: it was first trained with regard to only the prediction task, as the  $Loss$  is

computed by  $Loss_{prediction}$ . With the network pre-trained, it then gets fine-tuned on the overall  $Loss$ , which had the auto-encoder part trained as well. In the second training process, the weights for the encoder network can be locked, and only update the decoder weights. This two-step training process helps the network better trained for the two objectives.

## 3.8 Space Rating System

### 3.8.1 Problem Description



How **interesting** is this space?  
How **spacious** is this space?  
How **public** is this space?  
. . .

**Figure 3.8.1-1:** acquire the rating of a space through a panoramic depth image

The volume of space is deterministic, yet the experience about space is not. People tend to have subjective experience, and subtle differences can influence the nuance of experience. How interesting, how public, how spacious ... tiny spatial adjustments can influence all these different adjectives about space. Also, the same space can create a different experience for different subjects, and the same adjective about an experience is likely to mean something different for different subjects as well. Architects may see the experience of space more logically, while novices may react to space more intuitively, people with different age and gender may treat space slightly different, yet, the experience of all subject groups is equally essential for design.

If a computer can quantify human's feeling of space, a computer-aided system can be much more powerful and helpful to designers. However, the nuance of spatial experience makes it a challenge to quantify spatial experience. With the expertise of vector representation

(Section 3.7), if a network is trained so that experience-related features of space can be extracted, it is likely that the feeling of a space can be computed using its feature vector.

In this task, trained on a crowdsourced dataset and employing the idea of vector representation, a system is developed so that it rates a space panoramic depth sampling in three aspects through value interpolation in its feature vector space: interestingness, publicness, and spaciousness. Besides, the system can simulate the rating of space for different subject groups, such as architects and novices, young and old, male and female.

### **3.8.2 Data Crowdsourcing**

The biggest challenge of developing a space rating system is data acquisition. The amount and quality of data are crucial for any machine learning system. However, space rating data is sentimental and high level, which ensures that it cannot be easily generated from scratch. The only source of space rating ground truth is human intelligence.

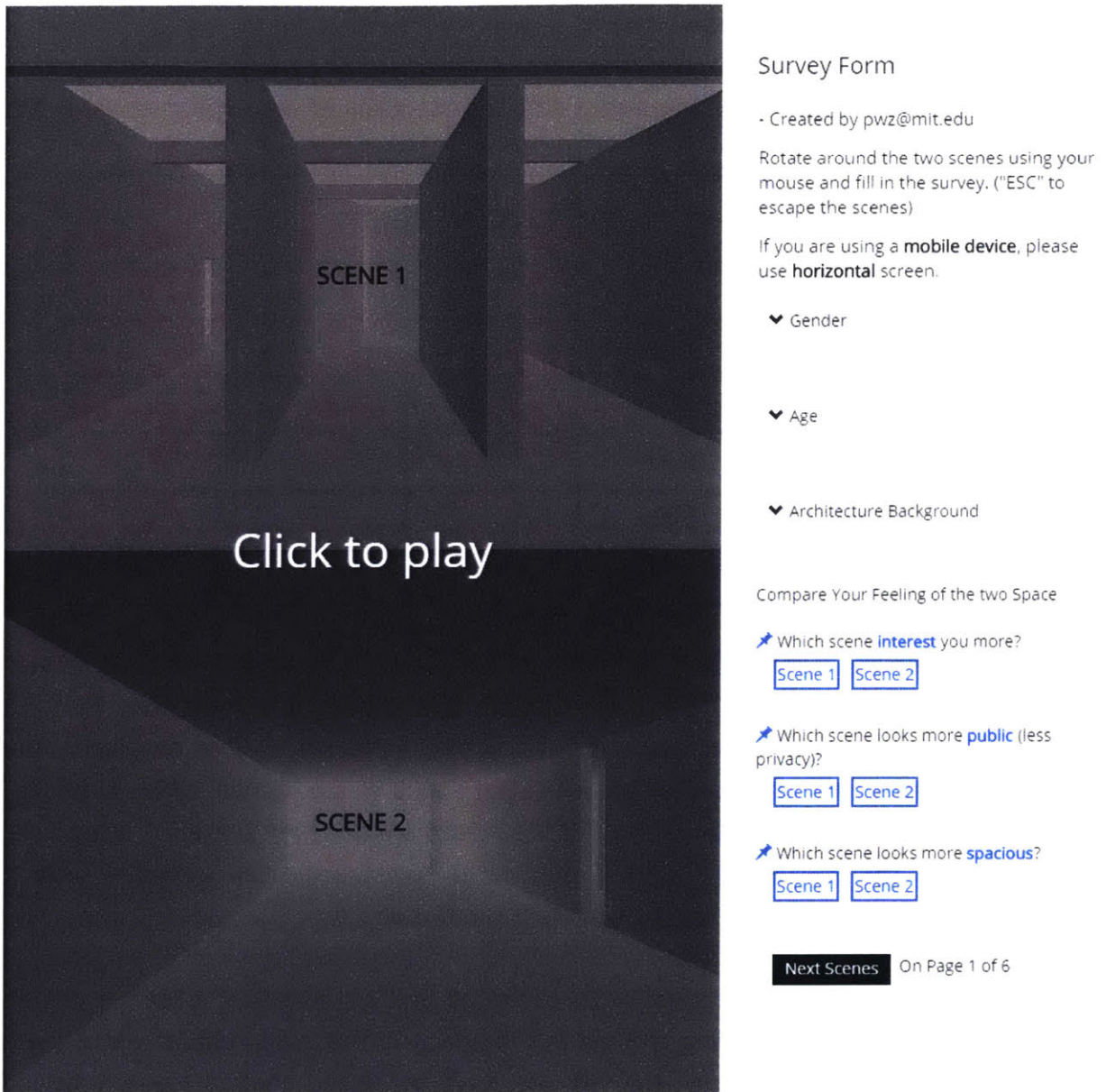
A customized survey is developed to collect sentimental space data from human intelligence. This survey is posted on crowdsourcing platform Mechanical Turk, which is briefly introduced in Section 3.3.3.

The screenshot below shows the interface of the survey. Although the space model samples utilized in building a training set can be of any random structures, existing architectural designs are mostly utilized in this survey. This selection tries to bias the system towards regular building-like space over entirely randomized space, and the dataset is believed to train systems in a certain way so that they have better performance on regular building-like space inputs. The selected models for sampling include Barcelona Pavilion and Exhibition House Berlin designed by Mies, Paviljoen van Aldo van Eyck and the 15 primary spatial compositions utilized in Section 3.4.

Since the rating of a space may be subjected to time, place, participant and many other factors, other than asking the participants to rate space directly, it can be more credible to acquire the comparison of a pair of spaces, which can then be converted to ranked scores<sup>39</sup>. With that in mind, the survey is designed in the way that it initially collects comparisons between pairs of scenes from surveyees.

---

<sup>39</sup> Naik, Nikhil, et al. "Streetscore-predicting the perceived safety of one million streetscapes." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014.



**Figure 3.8.2-1:** Interface of the survey

When the survey is loaded in a browser, the page shows up two random scenes in the canvas on the left in parallel; the two scenes are denoted as "scene 1" and "scene 2". Since the survey requires the participants to compare the two scenes in panoramic views while the scenes are shown in a regular format, they will need to control the view using their mouse or touchscreen to see the scene around. On the right side of the page, there is a survey form that collects the gender, age, and architecture background of the participant, along with the comparison results of the two scenes by the surveyee. The comparisons include "which scene interest you more," "which scene looks more public," and "which

scene looks more spacious.” The table shows the options for each field and the value of each option saved in the database.

**Table 3.8.2-1:** The options for the survey fields and the value of each option saved in the database.

<b>Value saved in DB</b>	0	1	2	3	4
<b>Gender</b>	Male	Female			
<b>Background</b>	Architect	Novice			
<b>Age</b>	0-12	12-18	18-30	30-50	50+
<b>Interesting</b>	Scene 1	Scene 2			
<b>Public</b>	Scene 1	Scene 2			
<b>Spacious</b>	Scene 1	Scene 2			

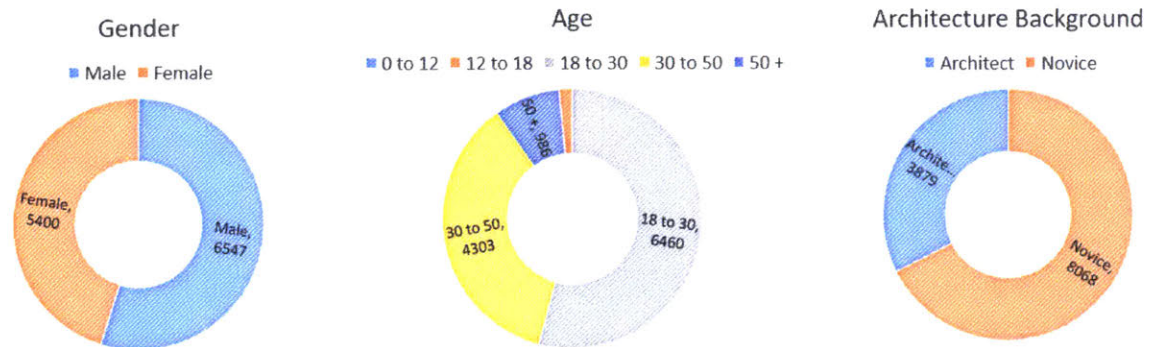
The survey collecting system tracks the status of the survey as it is being taken. It records the completion status of each field and ensures that all submissions have all fields filled. To ensure that the surveyees have compared the scenes thoroughly before completing the survey, the system will also monitor the activation status of the scenes, which is tracked when the user rotates around the scenes. If the scenes are not even activated, a reminder will pop up on the interface saying “It is required to compare the two scenes by looking around them thoroughly.”, and reject the current submission.

Besides, since there are two scenes to be controlled and compared by a user, it can be tricky and complicated regarding the control interface if the user controls each scene individually, and that also deepens the learning curve by adding another scene switching procedure. Because of this, the survey proposed in this experiment allows participants to control both scenes when moving the mouse or touching the screen, providing a parallel comparing experience.

Once a surveyee takes the survey and submit a result, the three comparisons “Interestingness”, “Publicness”, and “Spaciousness” will be saved together with the information of the two scenes being compared, along with the participant’s gender, architecture background, and age.



The survey was posted on the Mechanical Turk system as well as other social media such as Facebook and Wechat. In total, it collected over 10,000 scene comparisons in about two weeks. Here are some of the statistics about the surveyees:



**Figure 3.8.2-2:** Surveyee Stats

It can be seen that there is a similar amount of male and female participants. Regarding age, most participants are of age 18 to 30, followed by age 30 to 50. Further, there are about one third architect participants. The ratio of architect participants is slightly over the expectation, but since the survey is posted on the author's social media which might have a much higher architect-ratio, this result is considered credible.

### 3.8.3 Comparison to Rating

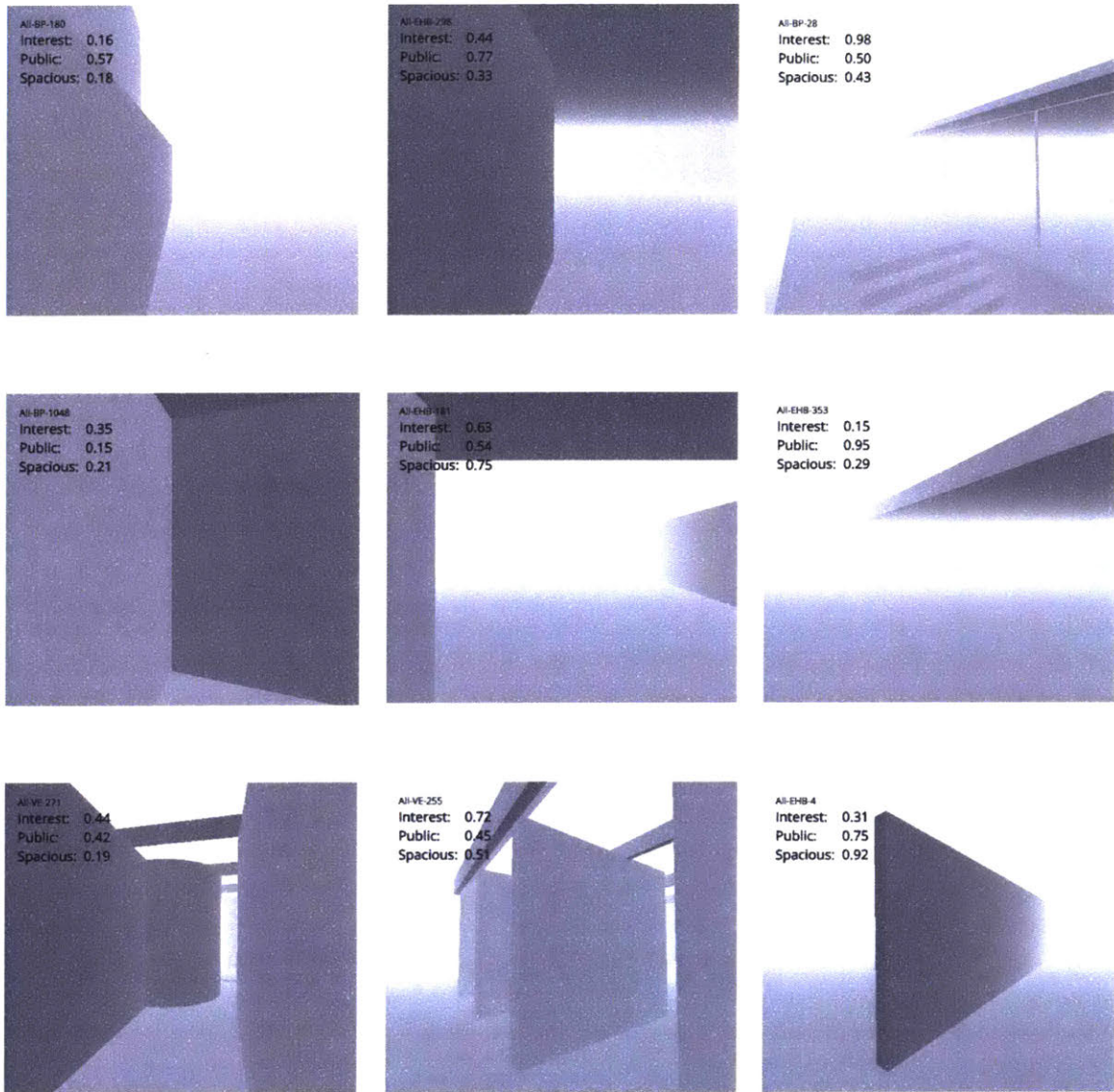
Since the data acquired is based on comparisons, it is required to convert the comparison data to absolute normalized values, or ranked scores to train a machine learning regression system. To do that, the "TrueSkill" algorithm<sup>40</sup> is employed that converts the result to absolute normalized scores.

TrueSkill algorithm is a skill-based ranking system developed by Microsoft. Each participant's skill can be represented as a normal distribution, with a mean value of  $\mu$  and a variance of  $\sigma$ . Every time a participant wins,  $\mu$  gets increased, and decreases when loss. Computed based on a bayesian graphical model, how much each skill score gets updated is determined by how surprising the comparison result is. In an unbalanced comparison, for example, the algorithm will update the values very little when the favorite wins, or will result in huge updates when the favorite loses.

<sup>40</sup> Herbrich, Ralf, Tom Minka, and Thore Graepel. "TrueSkill™: a Bayesian skill rating system." *Advances in neural information processing systems*. 2007.

In this case, each comparison of two spaces is considered a two-player contest. All spaces are first set to default values  $\mu = 25$ ,  $\sigma = 25/3$ , and the values get updated by the algorithm according to the comparison results collected from the survey. The final  $\mu$  scores compose a ranking list of all scenes. By normalizing the  $\mu$  values to the range of 0 to 1, a rating for each scene is then acquired. A rating, let's say interestingness of the scene, close to 1 means that this scene is more interesting than almost all the other scenes, and vice versa.

In a two-player contest, Trueskill algorithm converges after taking about 12 to 36 contests. In this case, to get a fully converged result, by calculation at least 20,000 comparisons are required. Besides, when calculating categorized ratings, fewer data samples can be taken from each category. For that, specific data augmentation processes should be conducted. In this case, an extra step was conducted to increase the number of comparisons between the selected scenes. Each panoramic depth image of a sampled scene is run through (or encoded by, the methodology of Auto-encoder is introduced in Section 3.7) a pre-trained neural network introduced in section 3.4, and resulted in a 50-dimensional latent vector. Since the network is trained through supervised learning, the resulting latent vectors encode the features of the panoramic depth image. In another word, the closer the distance between two latent vectors is, the more similar the two corresponding scenes are. Additional comparisons are generated for pairs when one of the two is very similar to another scene by comparing the distance in feature vector space.



**Figure 3.8.3-1:** A part of space rating results of the survey after converted comparisons to ranking scores.

### 3.8.4 Rating System

A rating regression system can be developed on the basis of the collected rating data. The general idea is that the system has rating scores for a collection of scenes, and given a new scene input, the system calculates a new rating score concerning the scores of existing surveyed scenes. This score can be calculated by interpolating the value in feature vector space (as introduced in section 3.7.2) or can be computed through another regression network.

Using the 3D Isovist sampling methodology introduced in section 3.2, each scene that has run through the surveys is converted to a panoramic depth image. This process is also applied to new input scenes to be rated by the system. On the basis of all the panoramic depth image samplings, a pre-trained network is utilized to convert the samplings to their vector representations.

This pre-trained network can be network trained for different purposes or with different training datasets, but different network results in different feature extraction process when an input sample is being run through. In this experiment, the network introduced in section 3.4 is utilized, and it can be seen that the system will have better performance when dealing with regular walls and column composed space, similar to the 15 space composition types featured in section 3.4.

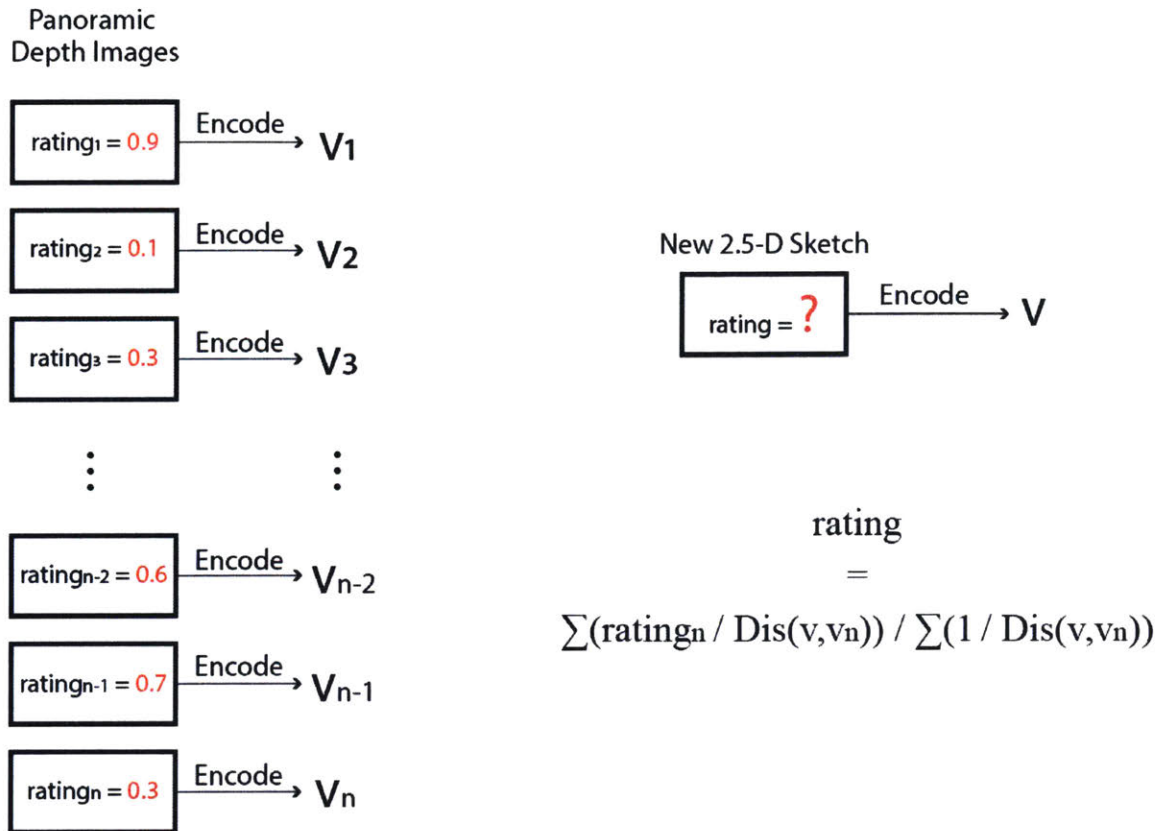
Once all the space samplings have been "encoded" to their vector representations, the rating of the input scene can then be computed in this feature vector space. In this case, vector interpolation method is utilized for computing the rating.

The formula for interpolation of the rating can be written as:

$$Rating = \frac{\sum_{i=1}^n \frac{Rating_n}{Dis(v, v_i)}}{\sum_{i=1}^n \frac{1}{Dis(v, v_i)}}$$

In the formula, vector  $v$  denotes the encoded vector representation of a new input scene, and  $v_i$  is one of the  $n$  vectors for the  $n$  surveyed scenes.

This whole computation system is illustrated below:



**Figure 3.8.4-1:** illustration of the computation of the rating system

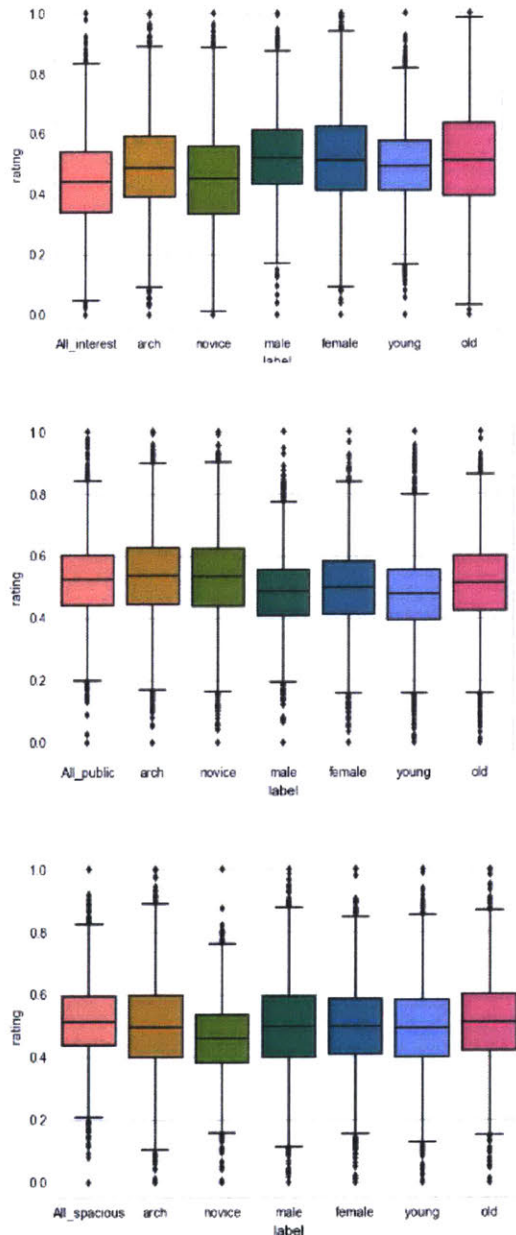
As can be seen in the formula, when computing the value of a new input, the system iterates through all the existing samples and interpolate the final result. This will work well with limited existing samples, but if the amount of existing samples is too significant, a regression neural network can be a better option, since it is lighter in computation.

The system introduced in this section is further tested with the interactive modeling software introduced in section 4.1 and 4.2, and additional results are shown in the appendix section.

### 3.8.5 Rating by Subject Groups

The data collected in the survey can be categorized by the backgrounds of the participants. The categories are made by gender (male or female), architecture background (architect or novice), and age (young-aged 0 to 30 or old-aged 30 or more). For data collected from different categories of participants, the ratings of the surveyed scenes may differ. So the data can also be grouped by category, and each generates a rating collection for the

surveyed scenes. The Rating distributions of the different groups are plotted below with box plot in Figure 3.8.5-1. They show the interestingness, publicness and spaciousness rating distributions respectively for different categories.

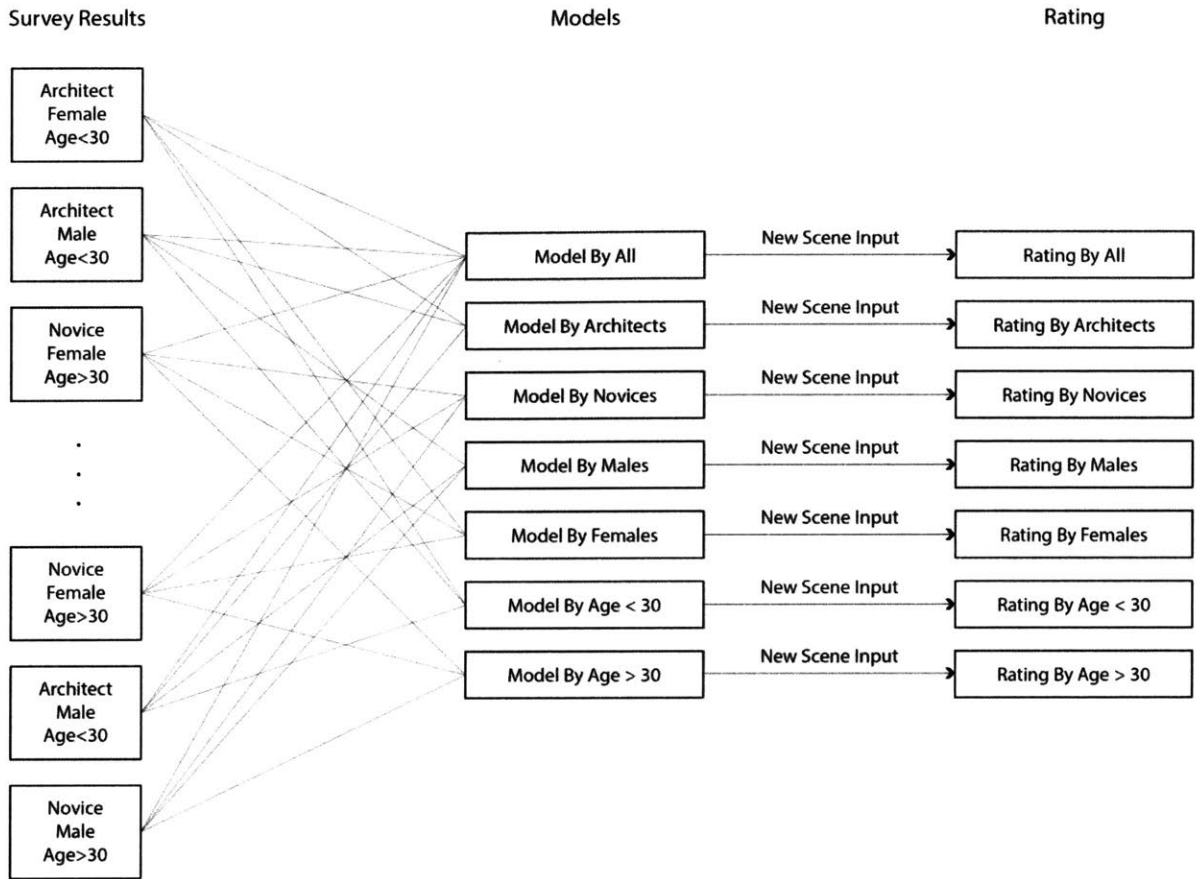


**Figure 3.8.5-1:** From top to bottom: Interestingness, publicness and spaciousness rating distribution box plot for different groups. Young means age smaller than 30 here, and old represents age larger than 30.

Based on these rating distribution plots, comparisons can be made about different categories of participants. Although the data amount is limited so it may be biased by different factors, it still shows opportunities in studying spatial awareness and its correlation to subject backgrounds.

For instance, female participants' rating about spatial interestingness and publicness tends to be a bit more extreme than male participants, as the ratings from female participants have a more substantial variance. A similar comparison can also be made in the spaciousness figure for architects and novices, or in the publicness figure for young and old. One fun fact to notice is that architects and novices have more diverged opinions about the interestingness and spaciousness of space compared to publicness.

Further, with the data categorized, instead of computing the rating by referencing all the survey results, the system can first filter out the participants and their survey data by subject groups and compute ratings only using the filtered results. In another word, the system can simulate a rating of space as a human of different categories. This process can be illustrated as the figure below:



**Figure 3.8.5-2:** Compute rating by subject groups through filtering comparison results by category and compute results separately.

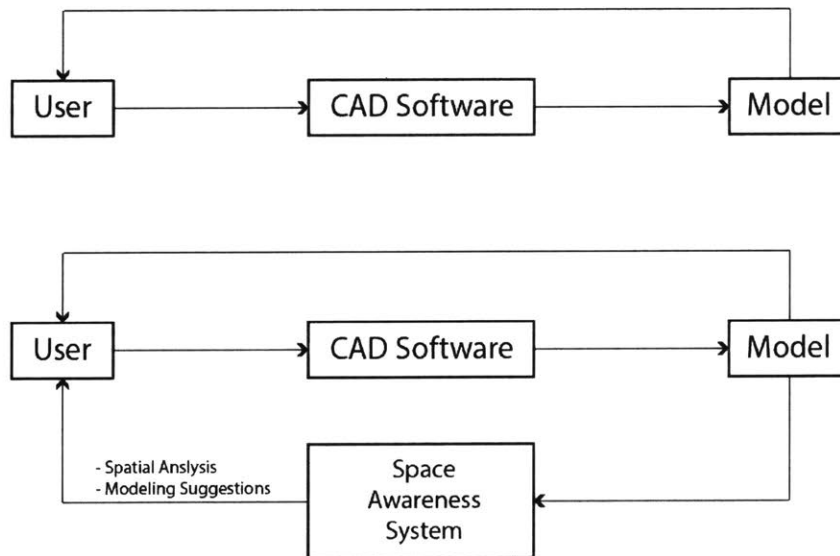
## 4 APPLICATIONS

To demonstrate the systems introduced in the previous section, several use cases are conducted. Those mainly include two parts: Firstly, they are applied to a real-time space evaluation modeling interface, which gives a user prompt insights about the scene being constructed; Secondly, they are also utilized in the spatial analysis of existing architectural designs, namely small designs by Mies van der Rohe and Aldo van Eyck. The case studies conducted validate that this methodology works well in understanding local spatial conditions, and it can be helpful either as a design aid tool or in spatial analysis.

These two parts are described in the sections below.

### 4.1 Interactive Modeling Based on Space Awareness

#### 4.1.1 Introduction



**Figure 4.1.1-1:** Comparison of a regular CAD operating workflow and a space awareness modeling system

Current Computer Aided Design (CAD) tools are similar to pens and paper; they follow users' commands. As a result, a design process involving CAD tools is more of a one-way process - human to computer. The system stores what the user's input as it is and follows orders. What's more, the geometric elements are discrete and cannot make sense as a whole.



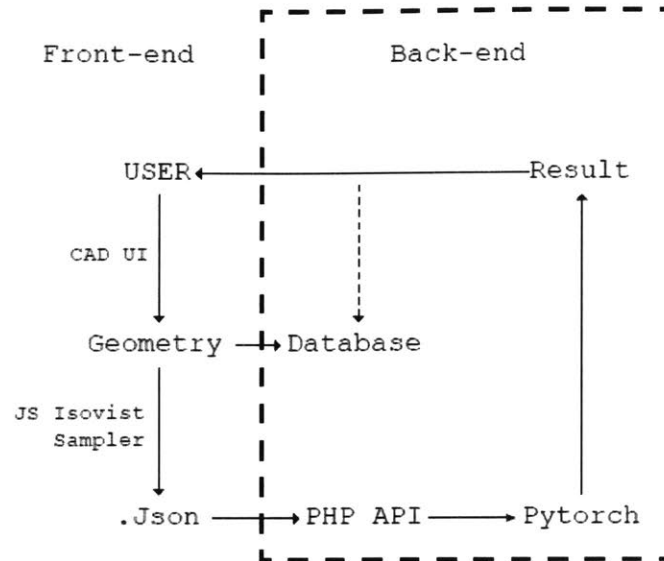
With space awareness systems implemented, CAD software can be more helpful and intelligent for designers to use. The systems can record not only how each element is placed, but attempt to understand what the user is trying to do, and yield spatial insights to user operations on the fly. Possible operations can also be suggested to the user based on the insights and thus better engaged in the design workflow (as illustrated in figure X).

In this demonstration, a real-time space evaluation modeling interface is developed. In using the modeling interface, a user can quickly create a scene by resizing a base cube and place it anywhere in the interface as he or she desires. After each operation, the system makes sense of the geometrical elements in the scene through the space awareness systems introduced in section 3, and provide insights of the designed space. Those insights include the space's panoramic depth image, composition classification result, scene classification result, 3D reconstruction result, and ratings of interestingness, publicness, and spaciousness.

Once the system can generate spatial insights for a given viewpoint in the scene, thanks to the efficiency and economy of computers, the system can sample and analyze the whole scene from every possible viewpoint easily. The insights allow the user to have a better understanding of the designed space from much greater perspectives.

Additionally, this interactive modeling tool also works as a testing toolkit for the space awareness systems developed in section 3. With the ability to quickly model a scene and run it through the systems, it also serves as a perfect trial and error platform for the development of the space awareness systems.

#### **4.1.2 Software Structure**



**Figure 4.1.2:** Software structure

The interactive modeling tool is a web-based application. It consists of a front-end for client-side user interaction, and a back-end dedicated to insights computation and data storage. On the front-end, a user can operate the geometris in the scene using the CAD toolkit, as a regular modeling software; The model, in real time, gets sampled and generates a .json format panoramic depth image, which is sent to API for back-end computation; In the meantime, the model also gets stored in the database, and that ensures that each step of the modeling process is kept track of; Once the back-end computation is done, the result will be sent back to the front-end to the user, and also saved to the database along with its geometry data.

The front-end is developed in Javascript<sup>41</sup>, with THREE.js<sup>42</sup> and D3.js<sup>43</sup> as major toolkits for 3D visualization, UI design or 2D visualization in the browser, and CSG.js<sup>44</sup> library for 3D model Boolean operations. Additionally, TreeModel.js<sup>45</sup> is used for model's tree graph management. The back-end mainly has two purposes, data storage, and computation. The database is developed with Firebase<sup>46</sup> and keeps track of the overall space prediction results, and the dimension, position, and time of each element. The computation function is developed as a PHP API: the API accepts data from the front-end (.json format) with a

41 "JavaScript." <https://www.javascript.com/>. Accessed 22 May. 2018.

42 Cabello, Ricardo. "Three.js." URL: <https://github.com/mrdoob/three.js> (2010).

43 Bostock, Michael. "D3.js." *Data Driven Documents* 492 (2012): 701.

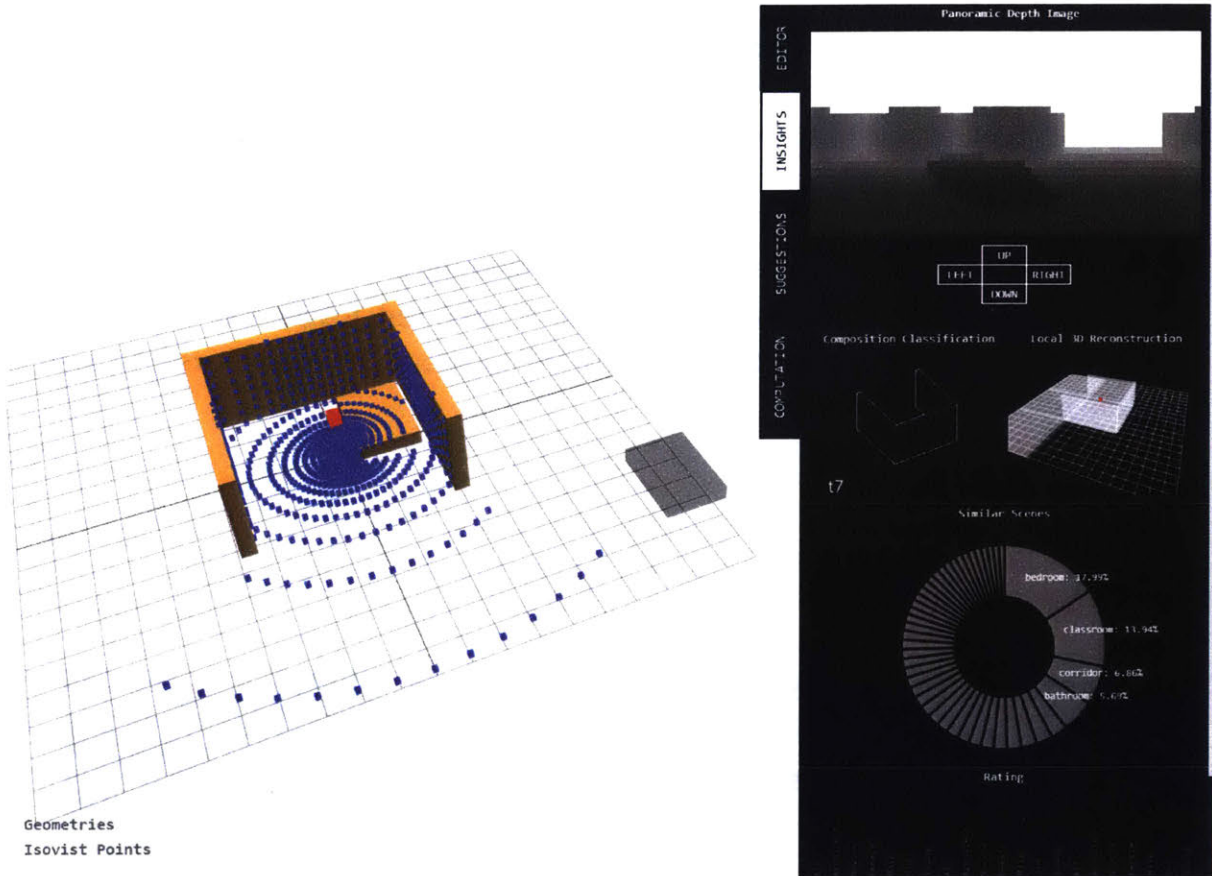
44 "csg.js." <http://evanw.github.io/csg.js/docs/>. Accessed 22 May. 2018.

45 "TreeModel." <http://jnuno.com/tree-model-js/>. Accessed 22 May. 2018.

46 "Firebase." <https://firebase.google.com/>. Accessed 22 May. 2018.

POST request, and run space data through a pre-defined python program running neural network systems (Introduced in section 3) developed in Pytorch. The printed result of the python program is then returned to front-end as the queried result of the POST request.

### 4.1.3 Interface



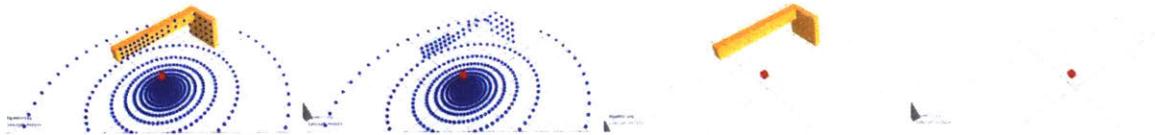
**Figure 4.1.3-1:** the interface screenshot.

The interface is designed similar to regular modeling software. It has a canvas on the left for 3D geometry previewing and editing, and a panel on the right with four tabs, namely "editor," "insights," "suggestions" and "computation," dedicated to different usage scenarios.

#### 4.1.3.1 Canvas

The canvas is mainly used for 3D geometry previewing and editing. It will show a horizontal reference plane as an operation desk for the user, and geometries are to be added on top of the reference plane. In the scene, there is also a red cube which locates initially in the middle above the reference plane. This red cube is an indication of viewpoint, and the

software will sample space in real time from this viewpoint and generate corresponding panoramic depth images (Section 3.2).



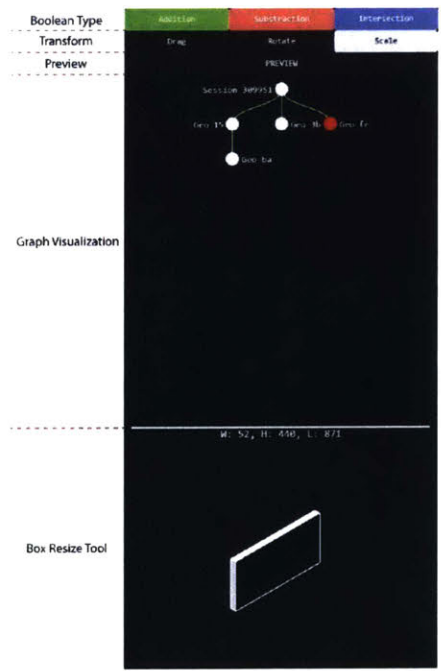
**Figure 4.1.3.1-1:** different display preview options in the canvas. From left to right are: geometry on, isovist points on; geometry off, isovist points on; geometry on, isovist points off; geometry off, isovist points off.

On the bottom left of the canvas, there are two layer marks, namely “Geometries” and “Isovist Points.” By clicking these layer marks, a user can toggle the display of the geometries and the sampled isovist points from the viewpoint. An example can be seen in the figure below.

### 4.1.3.2 Editor

The software is meant to be a user-friendly modeling tool. The editing operation procedurals are designed simple enough so that a user can model a scene easily by intuition. Users can construct a scene easily by placing cubes into the scene.

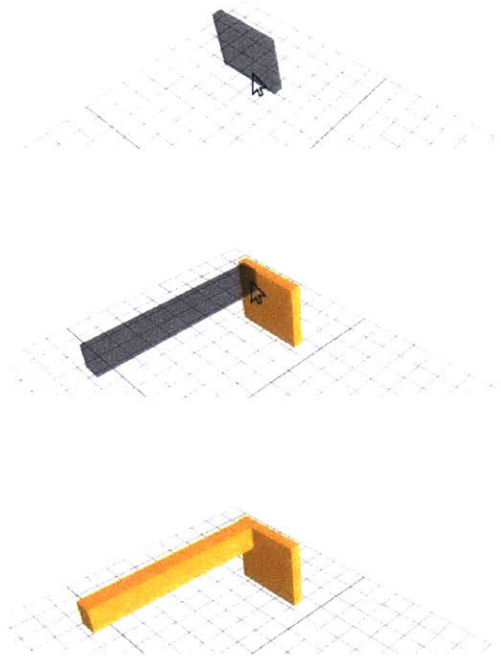
The “editor” tab is where the majority of the modeling tools are. On the bottom of this section, there is a box resize tool, where users can drag the faces of the cube by the mouse to resize it. (Figure 4.1.3.2-1) Once a box size is set, in the modeling canvas, the cube can be placed attaching to where the mouse targets, which will be acquired by intersecting the geometries in the scene, and offset the cube so that it is ideally attached to the pointed object. (Figure 4.1.3.2-2)



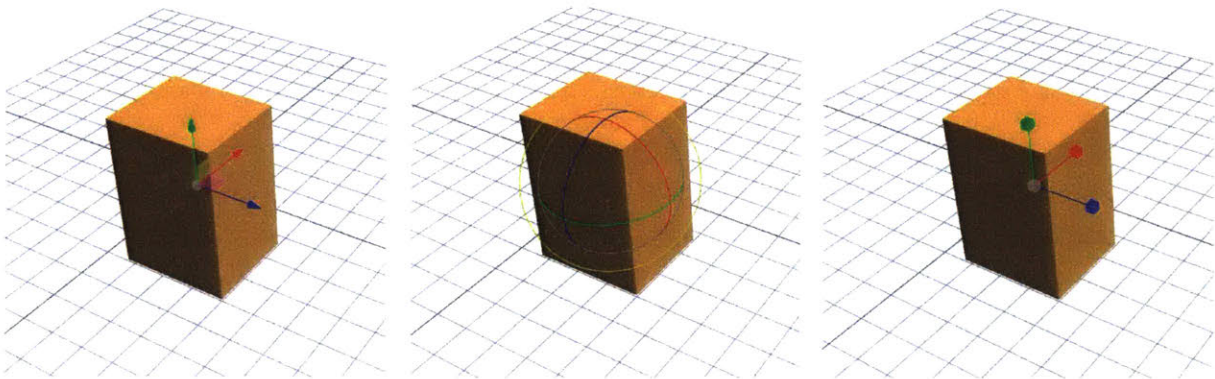
**Figure 4.1.3.2-1:** Interface of the “Editor” panel.

A user can also manipulate a geometry in the scene by clicking one of the options in section "transform". Once one transform mode is selected, geometries in the scene can be selected by mouse, and a control gumball will show up in the scene when a geometry is selected. "Drag", "Rotate", and "Scale" will show different gumballs that allows users to move around the object, rotate the object, and scale the object respectively in the scene. (Figure 4.1.3.2-3)

In section "graph visualization," (Figure 4.1.3.2-1) the software shows each cube as a node which is affiliated to another. By default, when adding a new cube, the new cube will be added as the child of the previous cube. Alternatively, by selecting a node in the "graph visualization" panel, parent of the new cubes can also be specified manually. This parent-child relationship will make no difference for "addition" mode, as all volumes are accumulated regardless of the relationship. However, it does matters when some cubes are added as a "subtraction" or "intersection." By default, the software will use "addition" mode for the current edit, and show the resulting model directly by adding. Alternatively, users can also select "subtraction" or "intersection" in the "Boolean Type" section. Instead, such operation will be applied to the parent of the current cube, not the whole geometry. Also,

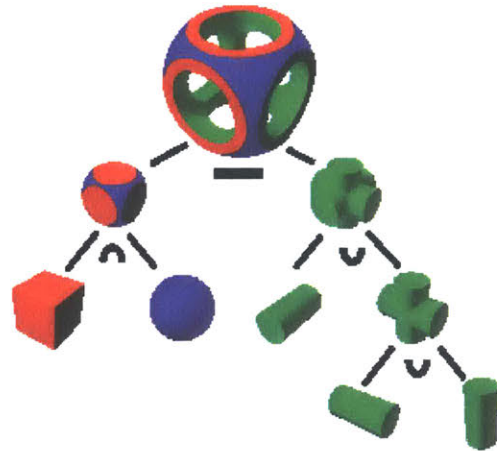


**Figure 4.1.3.2-2:** Geometry placement. The mouse intersects with a geometry (including the reference plane) in the scene and computes a placement position so that the object will be attached to the pointed object position.



**Figure 4.1.3.2-3:** Gumballs. From left to right: "dragging", "rotation", and "scaling"

the operation type will be color coded in “graph visualization” panel, namely green for addition, red for subtraction, and blue for intersection. The final result will be computed using the Constructive solid geometry (CSG) algorithm, and previewed when the user clicks the “preview” button, otherwise, the viewer shows all elements as they initially are.

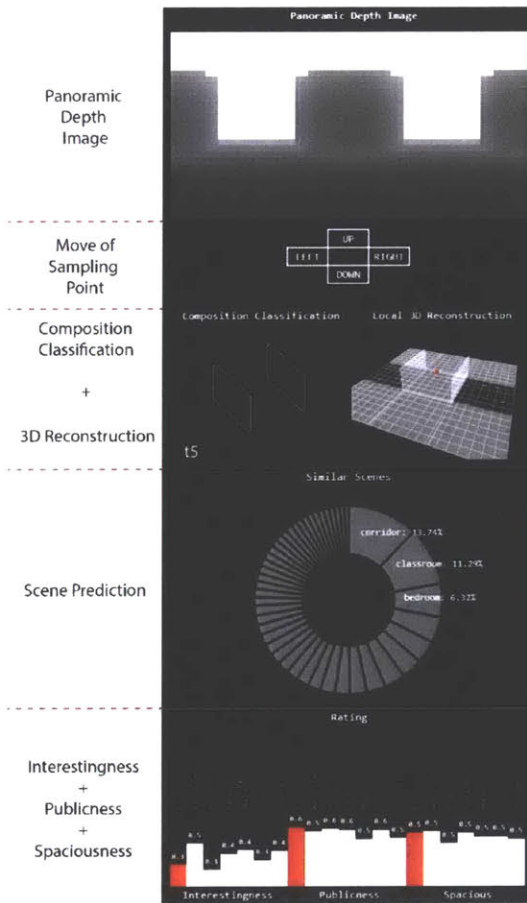


**Figure 4.1.3.4:** CSG operation by CSG tree. In this figure, the nodes are labeled “∩” for intersection, “∪” for addition (union), and “−” for subtraction. (Source: [https://en.wikipedia.org/wiki/Constructive\\_solid\\_geometry](https://en.wikipedia.org/wiki/Constructive_solid_geometry))

### 4.1.3.3 Insights

The “insights” panel is a visualization panel for the sampled panoramic depth image and its machines’ perception results returned from the API. Every time the user makes a new operation in the scene, a further sampling will be made and sent to the API for computation.

On the top of the “insights” panel, the software shows the panoramic depth image sampled from the specified viewpoint in the scene. It is already an unwrapped grayscale image with a resolution of 60 by 30. Below the panoramic depth image, there is a section with control buttons. Users can change the location of the sampling viewpoint with these buttons. The composition classification section, positioned below, shows the top-1 prediction label for the current input, it is using the system introduced in section 3.4. The 3D reconstruction section, to the right of the



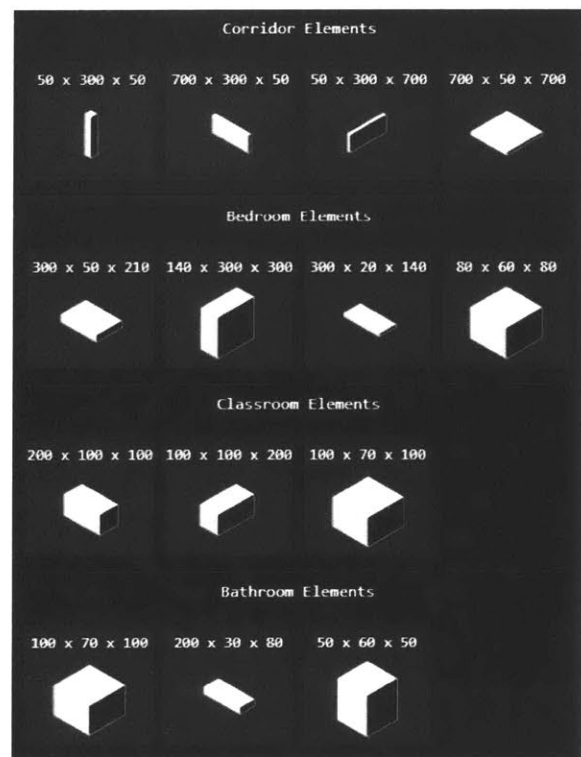
**Figure 4.1.3.3-1:** the “insights” section, which shows the space awareness results

classification section, shows a reconstructed 3D structure computed with the system introduced in section 3.6. It is a model in 3D which users can pan, zoom, or scale. Below this section, the scene prediction section shows the result for similar scenes predicted by the system in section 3.5. The results are shown with a donut chart, and the area of each region represents the probability of the correlated scene type. On the bottom of the whole panel, there is a histogram showing the ratings for measurements including interestingness, publicness, and spaciousness. The result also includes ratings by different categories of people, as introduced in section 3.8.5, which are filled in white, whereas the overall ratings are highlighted with red fillings.

#### 4.1.3.4 Suggestions

The “suggestion” panel is a section where system recommended predefined shapes are listed, and a user can easily apply by selecting. The recommendations are made by the scene classification (Section 3.5) results of the current model, and a probability of a scene larger than 5% will activate the suggestion for that specific scene type. The system lists the most popular shapes for the predicted scenes. For example, if the scene is predicted as a bedroom (probability > 5%), the suggested shapes will include a cube of the size of a bed.

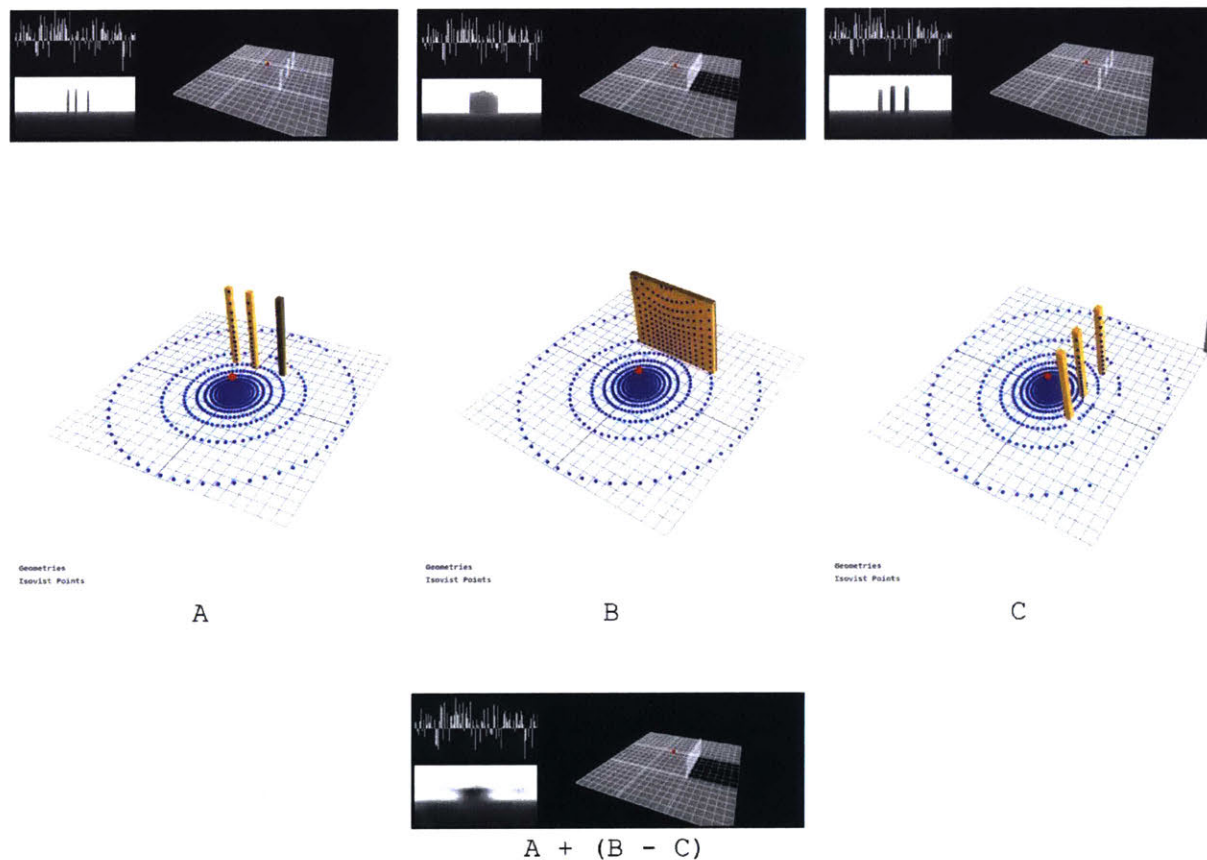
For the current time being, the suggestion system is only built on the scene classification results, and the suggestions are limited to affecting the modeling cube’s dimension. It can be anticipated that it can be more powerful as more spatial insights are contributing and a more vast geometry database is built for the suggestive system.



**Figure 4.1.3.4-1:** “suggestion panel”, show system recommended shapes to apply. Recommendation is made by scene classification results.

#### 4.1.3.5 Computation

The last panel of this system is "computation," and it provides functions for space computation using the autoencoder system introduced in section 3.7. It has four scene placeholders, namely "A," "B," "C," and "A + (B - C)." A user can model a scene using the editing tools, and set the current space to either "A," "B," or "C" in this panel. Once a placeholder is set, it shows the panoramic depth image of the specified scene; the network encoded vector representation visualization and the 3D reconstruction of the current scene. The vector representation, which has 50 dimensions, is the feature vector encoded by the auto-encoder system. The vector visualization shows the value in each dimension as one vertical segment.



**Figure 4.1.3.5-1:** Scene A, Scene B, Scene C, and the computation panel which shows the result of computing "A + (B - C)".

Once all three of the scenes "A", "B", and "C" are specified, by clicking the button "calculate A + (B - C)", the system sends the data of scenes "A", "B", and "C" to the back-end and run through the auto-encoder system. The system first computes the vector representation of each scene, and conduct algebraic operations of "A + (B - C)" in vector space. This new



vector is seen as the vector representation of the resulting space; With the new vector in hand, the system uses the vector as feature vector and run it through the segmentation network introduced in section 3.6 to get the 3D reconstruction result of the resulting space; Additionally, by decoding the new vector, the autoencoder system gets a panoramic depth image of the resulting space. The new vector, 3D reconstruction result, and decoded panoramic depth image are then visualized in the last placeholder "A + (B - C)."

### 4.1.4 Use cases

For every scene, a panoramic depth image is sampled from the viewpoint represented by the red cube. It was then sent to the backend of the software for computation, and the system yields the results for composition classification, 3D-Reconstruction, Similar Scenes, and ratings by different subject groups. The table below shows some results of the interactive modeling interface running on some very basic geometries. On these fundamental scenes, the system works decently, and every function is working as expected.

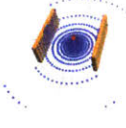
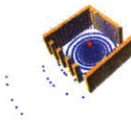
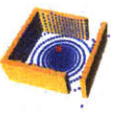

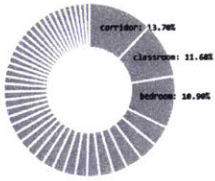
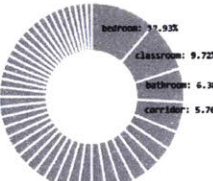
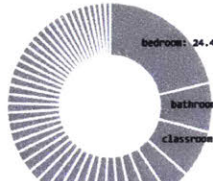
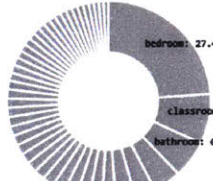








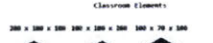


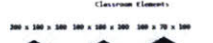

**Table 4.1.4-1:** results running on simple geometries

Geometry						
Point Cloud						
Panoramic Depth Image						
Composition Classification						
3D-Reconstruction						
Similar Scenes	<p>- Corridor - Classroom</p>	<p>- Corridor - Classroom</p>	<p>- Corridor - Classroom</p>	<p>- Corridor - Classroom</p>	<p>- Bedroom - Corridor</p>	<p>- Bedroom - Bathroom</p>

	- Discussion area	- Discussion area	- Bedroom - Bathroom	-Bedroom	- Classroom - Bathroom	
Interestingness						
Publicness						
Spaciousness						

Further, the system is tested in a specific use case when a user tries to model a bedroom. A record of the process is listed below in Table 4.1.4-2. Initially the scene is blank. The user starts by placing walls in the scene, like in step 1 below. When two parallel walls are added to the scene, it was seen as more of a corridor like space, followed by classroom and bedroom. The system also suggested some predefined sizes to choose from for these scenes. The system suggested some columns and walls at this stage, as these are elements selected more according to statistics in a corridor-like scene. As it is more like a corridor, for now, the user continues to place elements to enclose the space and get step 2. It is a room with one side made of frames, and the other three sides are walls when one of it has a door-like gap. At this stage, space is seen more like a room. The top 2 predictions have been updated to bedroom, and classroom, while the difference between the two is very limited. The user may be happy with the current enclosing, but if not since it is too similar to a classroom, he can easily modify the frames side of the room and make it more private into step 3. In step 3, a wall with only a smaller window replaced the previous frames, and the prediction become bedroom and bathroom when bedroom now has a probability boost from below 13% to 24.49%. The user can continue the modeling process by adding interior elements, let's say a bed. The user can directly choose the bed from the suggested shapes, and add it to the room. Once the bed is added, as shown in step 4, it can be seen that the bed prediction probability continues to increase and reached 27.40%.

**Table 4.1.4-2:** A modeling record of a bedroom.

Step	1	2	3	4
Scene				
Scene Prediction	 <p>Corridor; Classroom; Bedroom</p>	 <p>Bedroom; Classroom; Bathroom; Corridor</p>	 <p>Bedroom; Bathroom; Classroom</p>	 <p>Bedroom; Classroom; Bathroom</p>
Suggested Shapes	<p>Corridor Elements</p>  <p>Bedroom Elements</p>  <p>Classroom Elements</p> 	<p>Corridor Elements</p>  <p>Bedroom Elements</p>  <p>Classroom Elements</p>  <p>Bathroom Elements</p> 	<p>Bedroom Elements</p>  <p>Classroom Elements</p>  <p>Bathroom Elements</p> 	<p>Bedroom Elements</p>  <p>Classroom Elements</p>  <p>Bathroom Elements</p> 

### 4.1.5 Discussion

Although the interactive modeling software is limited in function and can only accommodate the modeling of simple forms, the employment of the space awareness system suggests new possibilities in computer-aided design and shows great potential for making a modeling process more efficient. The results generated by the space awareness system are as expected, and the system shows excellent efficiency that it yields results promptly in real time. The efficiency allows for tight integration of human-machine interaction.

In this early phase of the application, only scene classification results are taken to generate modeling suggestions, and limited predefined geometries are recorded. But even now, the system already demonstrates the idea of how the machine reads a scene constructed by

human and provides feedback. It can be seen that as a more massive database is constructed with a greater variety of elements, and are labeled with more spatial features, the system can be much more powerful and useful. The features attached to geometric elements can include not only scene types, but interestingness, publicness, spaciousness, and a lot more. Other than the features of a static geometric state, if the system can predict how the features are influenced by a specific element before adding it, the trial and error process of a modeling procedural can be reduced and thus the design process can be more efficient.

## **4.2 Analysis of existing buildings**

### **4.2.1 Introduction**

Simple and common architectural elements can be combined to create complex spaces. Different spatial compositions of elements define different spatial boundaries, and each produces a unique local spatial experience to observers inside the space. Therefore, an architectural style brings about a distinct spatial experience.

However, it turns out that the spatial experience of architecture can hardly be quantitatively studied, as it requires the personal engagement of the space. As Tadao Ando stated that in designing, architects would have to think about how people will approach the building and experience that space. That fact ensures that traditionally if one wants to study the style and spatial quality of an architecture statistically, it demands a significant amount of efforts from the professionals.

If the space perception systems can be applied to a quantitative study of architectural designs, researchers and designers can understand the local spatial conditions of architecture from a new perspective, acquire the pattern and frequency of their appearance in designs, and study the peculiar spatial experiences embedded in an architectural style.

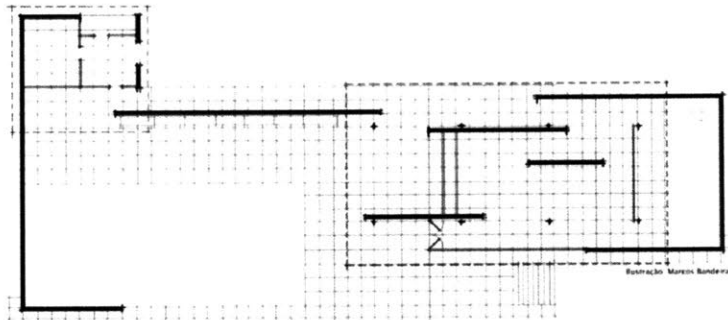
In this case study, several existing architecture projects are sampled using their digital models and ran through the systems. With their samplings and the networks trained ahead of time, spatial analysis of the sampled models can be acquired. In this analysis, the systems introduced in Section 3.4 and 3.8 are utilized to get space composition classification results and space rating results for each space sample. Further, the concept of vector representation introduced in section 3.7.2 is employed in the spatial analysis of the selected buildings and allows for the computation of results such as vector representation distribution, centroid representation vector, and representation vector variance. The analysis of these selected buildings leads to style related insights. The results are shown in section 4.2.2, section 4.2.3, and section 4.2.4.

Same as the samples acquired for training, utilizing the panoramic depth sampling method introduced in section 3.2, the three models are sampled with a resolution of 1 x 1 m from the height of 1.6 m from the floor, and only in areas close enough to the buildings (either inside of the building or within a distance of 4 m) and human accessible (pools are not included). Glass and windows are removed from the models, as only visual geometric boundaries are considered in this experiment.

## 4.2.2 Selected Buildings

Several architectural design projects are sampled and ran through the systems using their digital models. These projects are selected considering mainly two reasons: (1) The selected projects are famous for their spatial design. (2) The selected projects are one-story buildings, and their space compositions are similar to the training datasets utilized in training the space perception systems. The selected projects are shown in the list below, and they include two Mies projects and one Aldo van Eyck project:

- (1) Barcelona Pavilion, Mies van der Rohe

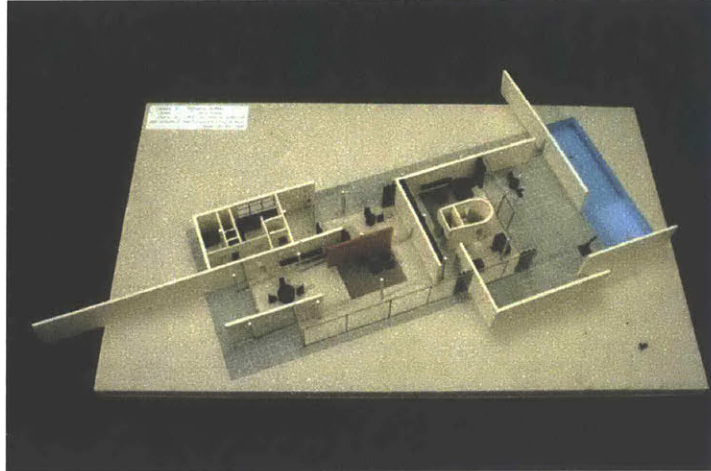


**Figure 4.2.2-1:** Plan of the Barcelona Pavilion.

(Source: <http://a2d-architecture.com/post/25012108227/to-german-pavilion-barcelona-spain-by-mies-van>)

The Barcelona Pavilion was designed by Mies van der Rohe. It was originally designed as the German National Pavilion for the Barcelona International Exhibition, but it soon became an important building in the history of modern architecture, known for its form and use of materials.

- (2) Exhibition House, Berlin, 1931, Mies van der Rohe



**Figure 4.2.2-2:** The Exhibition House, Berlin, 1931.

(Source: <https://www.pinterest.com/pin/373376625332801494/?lp=true>)

The Exhibition House, Berlin is another spatial experiment conducted by Mies in 1931, after the Barcelona Pavilion.

### (3) Paviljoen van Aldo van Eyck



**Figure 4.2.2-3:** Paviljoen van Aldo van Eyck.

(Source: <http://data.collectienederland.nl/page/aggregation/kroller-muller/73454>)

Paviljoen van Aldo van Eyck was designed in 1965-1966 for the 5th International Sculpture Exhibition and it heavily featured circles and curves as key elements of Van Eyck's "humane architecture."

## 4.2.3 Space Composition Classification

### 4.2.3.1 Results

The three selected buildings are sampled and ran through the space composition classification system. The statistical result of each space type from the system is shown in Table 4.2.3.1-1 for Barcelona Pavilion, in Table 4.2.3.1-2 for Exhibition House Berlin, and in Table 4.2.3.1-3 for Paviljoen van Aldo van Eyck; The sampling locations for the three designs are illustrated in Figure 4.2.3.1-1.

**Table 4.2.3.1-1:** Barcelona Pavilion Stats (1071 samples in total)

	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14
Sample Number	6	5	20	115	44	0	41	46	12	0	10	445	42	115	170
Sample Percentage	0.6%	0.5%	1.9%	10.7%	4.1%	0%	3.8%	4.3%	1.1%	0%	0.9%	41.5%	3.9%	10.7%	15.9%

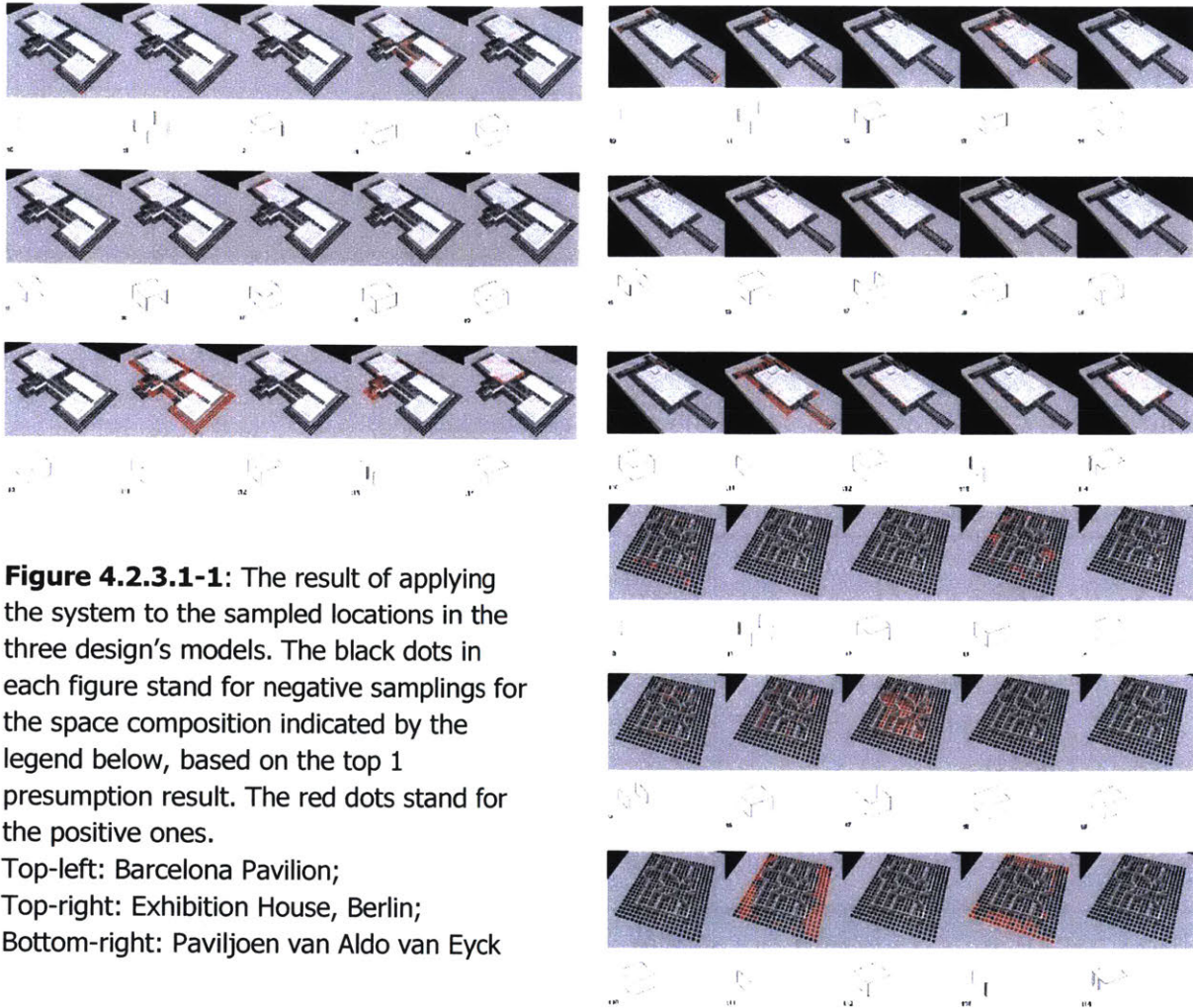
**Table 4.2.3.1-2:** Exhibition House, Berlin Stats (1140 samples in total)

	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14
Sample Number	29	23	4	153	83	5	108	36	5	0	73	322	97	29	173
Sample Percentage	2.5%	2.0%	0.4%	13.4%	7.3%	0.4%	9.5%	3.2%	0.4%	0%	6.4%	28.2%	8.5%	2.5%	15.2%

**Table 4.2.3.1-3:** Paviljoen van Aldo van Eyck Stats (436 samples in total)

	t0	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	t11	t12	t13	t14
Sample Number	16	0	0	40	3	17	31	107	3	1	0	115	0	102	1
Sample Percentage	3.7%	0%	0%	9.2%	0.7%	3.9%	7.1%	24.5%	0.7%	0.2%	0%	26.4%	0%	23.4%	0.2%

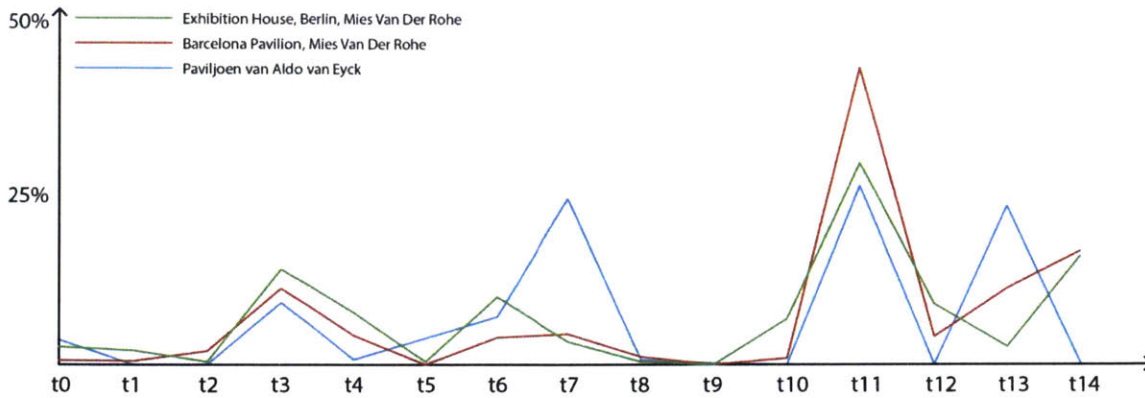




**Figure 4.2.3.1-1:** The result of applying the system to the sampled locations in the three design's models. The black dots in each figure stand for negative samplings for the space composition indicated by the legend below, based on the top 1 presumption result. The red dots stand for the positive ones.  
 Top-left: Barcelona Pavilion;  
 Top-right: Exhibition House, Berlin;  
 Bottom-right: Paviljoen van Aldo van Eyck

### 4.2.3.2 Result Analysis

These results show that the system can provide reasonable space-type presumptions for new sampled spaces. Though the results are not perfect, in most cases, the generated presumptions were close to human perception of the spaces. In addition, statistic reports for the whole buildings can be acquired by running through samplings of them. The results can be utilized to distinguish the building or its style. Figure 4.2.3.2-1 and Table 4.2.3.2-1 show the accumulated results acquired from the tests of the three buildings.



**Figure 4.2.3.2-1:** Plot of Seed-Space distribution in all three buildings.

**Table 4.2.3.2-1:** Categorized sampling result for all three buildings.

	Wall Elements (t3, t4, t5, t6, t7, t8, t9, t10, t11, t12)	Column Elements (t0, t1, t2, t13, t14)	Shaded Walls (t4, t6, t8, t10, t12)	Not Shaded Walls (t3, t5, t7, t9, t11)	Shaded Columns (t2, t14)	Not Shaded Columns (t0, t1, t13)
Barcelona Pavilion	70.4%	29.6%	13.8%	56.6%	17.8%	11.8%
The Exhibition House, Berlin	77.3%	22.6%	32.1%	45.2%	15.6%	7%
Paviljoen van Aldo van Eyck	72.7%	27.3%	8.5%	64.2%	0.2%	27.1%

For the case of the Paviljoen van Aldo van Eyck (PVAVE), it is interesting to notice that although the only compositional elements of the building are walls, it's still presumed to have 25.9% column like space. By checking the distribution mapping of space in Figure 10, it can be seen that the one-sided columns (t13) are primarily situated against the ends of walls, where the local experience is more similar to columns. Similarly, in the PVAVE case, pocket-walls (t7) is shown in many of the samplings, where the observer is surrounded by curved walls on one side. The fact that one type of space can be composed using other types of elements is reasonable, though it may not be obvious when only considering the type of the original elements. Spatial experience is too obscure to be described merely using

drawings or models, but the proposed methodology suggests a possible solution. Additionally, the system can be improved by adding these new space types, “against walls” or “enclosed by curved walls,” to the Seed-Spaces. The system will be able to identify these new spatial compositions with an adjusted network so that the result will describe the PVAVE’s design more precisely.

Comparing the Barcelona Pavilion case and the Exhibition House Berlin case, the Mies designs both have walls and columns in the models. Mies van der Rohe is famous for designing free-flowing spaces, and it can be seen that the less open space types, such as t7, t8, t9, and t10, rarely appear in these two buildings. The difference between the two also can be identified from the sampling results. The BP has more outdoor space compared to the EH, yet the EH has a greater proportion of walls.

The main distribution of Seed-Spaces displays similar trends for the EH and the BP. The PVAVE case shows different proportions of Seed-Spaces compared with the previous two, although all three designs are one-story buildings composed of mainly walls (t11). The PVAVE case returns more pocket walls (t7) and one-sided columns (t13), but no shaded one-sided columns (t14).

Last but not the least, it’s also important to point out that for each sampling, the network produces a probability distribution of the Seed-Spaces. In this perspective, each spatial composition can be considered as a “Hybrid Space” of the SeedSpaces. For example, if the result indicates that the input has a 60% probability to be space t1 and a 40% probability to be space t2, this input space T` is 60% similar to t1 and 40% similar to t2. It can be considered as a hybrid space ( $0.6t1 + 0.4t2$ ) of the two. This soft assignment allows the system to be applied to analyze space in more flexible scenarios, allowing for the description of transitional space types.

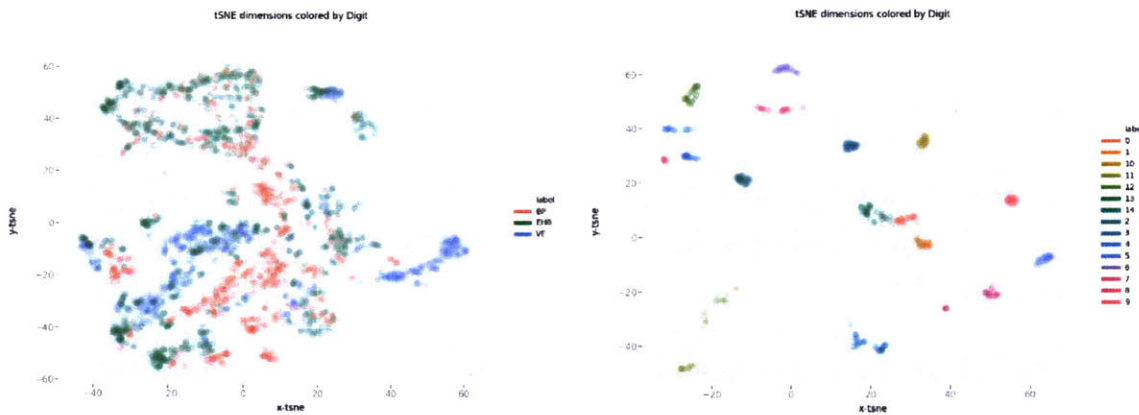
#### **4.2.4 Vector Representations**

As seen in section 3.6.2, the similarity of different spaces can be considered as the distance between the vector representations of these spaces. After a thorough sampling of the selected architectural designs, the samplings are then encoded by network 3.4, resulting in vector representations, which are 50-dimensional feature vectors. By applying the t-SNE algorithm (Section 2.4.6) to reduce the dimension of the feature vectors from 50 to 2, it allows for plotting of these vector representations in a 2-dimensional space. The vector dimension reduction process (t-SNE) of the “Seed-Spaces” and the three buildings are run in

a single e-SNE process so that they share the same reprojection procedure, and that ensures the 2-D vectors can be compared with each other across the source.

#### 4.2.4.1 Results

The vector representations plots of the three designs is shown below. Each dot represents a vector which has been projected to 2-D, and the vectors from the three buildings are color-coded by which building a specific vector belongs to. Also, the plot of the 15 "Seed-Spaces" is shown again for comparison. Besides the plot, data about the vector representations of the three buildings are listed in Table 4.2.3.1-1.

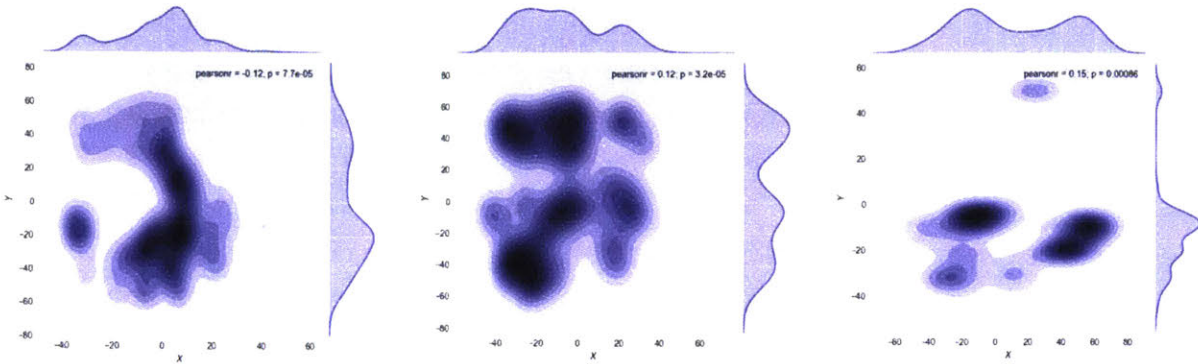


**Figure 4.2.3.1-1:** 2-D vector representation plots for samplings in the three buildings and the 15 "Seed Spaces." Left: plot for the three designs, with red for Barcelona Pavilion, blue for Paviljoen van Aldo van Eyck, and green for The Exhibition House, Berlin. Right: Plot for the 15 "Seed Spaces."

**Table 4.2.3.1-1:** The distribution values for the three designs

	Count	X Range	Y Range	X Centroid	Y Centroid	X Variance	Y Variance	Variance
Barcelona Pavilion	1071	[ -37.78, 52.81 ]	[ -55.03, 58.90 ]	-2.11	-5.10	278.70	1000.62	1279.32
The Exhibition House, Berlin	1140	[ -43.43, 60.09 ]	[ -55.19, 60.14 ]	-7.70	5.73	433.16	1283.73	1716.90
Paviljoen van Aldo van Eyck	460	[ -43.45, 60.70 ]	[ -45.60, 52.08 ]	11.64	-11.85	1044.86	325.29	1370.14

In order to have a better comparison of the vector representations between the three designs, distribution heatmaps are created for them. The heatmaps are shown below in Figure 4.2.3.1-2.



**Figure 4.2.3.1-2:** Vector Representation Distribution Heat Map of Barcelona Pavilion (left), The Exhibition House, Berlin (middle) and Paviljoen van Aldo van Eyck (right)

#### 4.2.4.2 Result Analysis

From the results shown in the previous section, we have a better idea of the similarities between each design project, and how they are similar. For example, by overlapping the plots in Figure 4.2.3.1-1, it can be seen that the most prominent difference of Paviljoen van Aldo van Eyck against the two Mies designs is it has a lot more t5 and t7 space. The locations of t5 and t7 vector dots in Figure 4.2.3.1-1 overlaps mostly with the dots representing Paviljoen van Aldo van Eyck. Also from Table Figure 4.2.3.1-1, the values of variance suggest that The Exhibition House Berlin seems to have more variety of space than the other two. Besides, the centroid values show a general impression that by average, Barcelona Pavilion is in the middle of t13, t14, t12 and t11, suggesting it is about the mix of columns and walls with shade. In comparison, Exhibition House Berlin is similar to Barcelona Pavilion, just that its centroid has a more significant y value, and that is due to the accumulation of samplings belonging to quadrant 1 and 2, which are mainly for walls (t6, t7, t12).

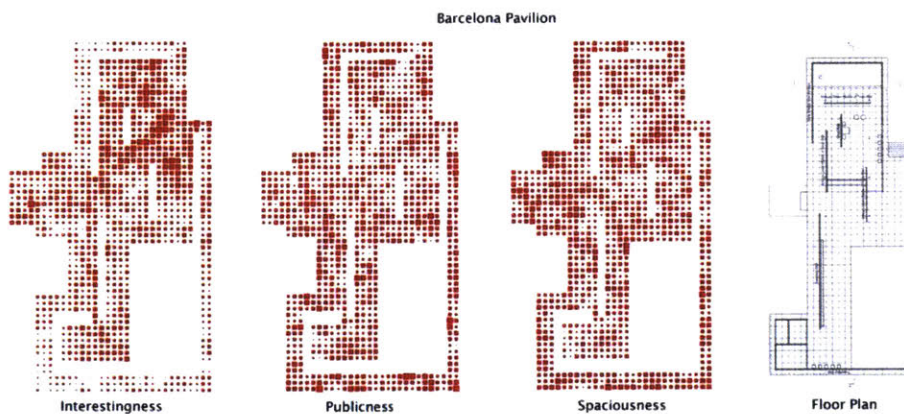
Also, from the vector representation distribution heat map (Figure 4.2.3.1-2), it can be seen that the two Mies projects have a comparatively similar distribution, yet Paviljoen van Aldo van Eyck is more unique. Almost all vectors of Paviljoen van Aldo van Eyck locate in quadrant 3 and 4, and its X-Variance is much larger than Y-Variance. It is likely that this vector representation data is style related.

## 4.2.5 Ratings

### 4.2.5.1 Results

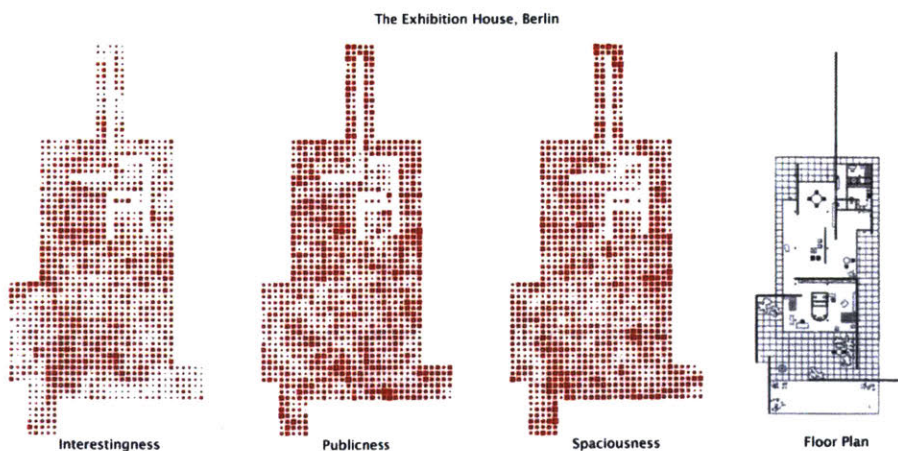
The space rating system (Section 3.8) has been applied to the samples of the three selected models. Since the survey collected data concerning the interestingness, publicness, and spaciousness comparison of scenes, these are also the three feasible measurements of the space rating system. Ratings computed with all data are shown in this section below, and rating results by individual categories are attached in the appendix section.

The figures below show rating fields of the three buildings. Each dot represents its sampling location in the model, and the size of a dot represents the intensity of that measurement.



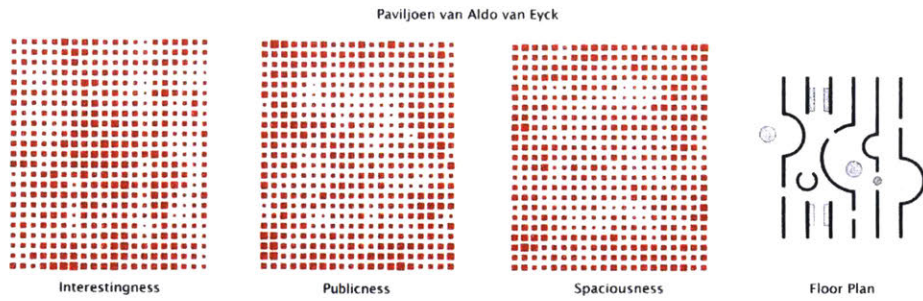
**Figure 4.2.5.1-1:** Rating results for Barcelona Pavilion.

Left: Interestingness, middle, publicness, right: spaciousness.



**Figure 4.2.5.1-2:** Rating results for the Exhibition House Berlin.

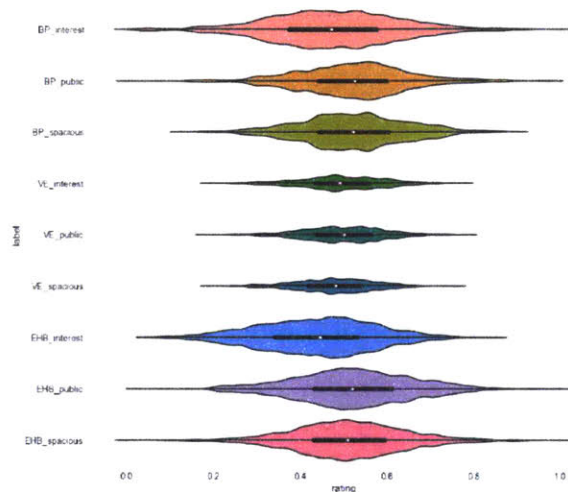
Left: Interestingness, middle, publicness, right: spaciousness.



**Figure 4.2.5.1-3:** Rating results for the Exhibition House Berlin.

Left: Interestingness, middle, publicness, right: spaciousness.

Besides the rating fields, a violin graph is shown on the right (Figure 4.2.5.1-4), which shows the rating value distributions of each measurement, in each building. In parallel to the horizontal axis, it consists of 9 plots, representing the three measurements for the three designs respectively; The horizontal location of a plot stands for the rating value ranging from 0 to 1, and the height of the plots on different rating values means the count of the specific rating value. Also, the white dot on each plot stands for the mean of that rating, and the range of the horizontal black thick line suggests the value of the first and third quartile.



**Figure 4.2.5.1-4:** violin graph showing the rating value distributions of each measurement, from top to bottom: Interestingness, publicness, spaciousness of Barcelona Pavilion, Interestingness, publicness, spaciousness of Paviljoen van Aldo van Eyck, Interestingness, publicness, spaciousness of Exhibition House Berlin.

#### 4.2.5.2 Result Analysis

From the rating fields, we can see that the rating system is producing some convincing results for the three designs. People tend to have more interest in space where there are more variety of elements, such as the interior parts and the stairs of Barcelona Pavilion. In the Paviljoen van Aldo van Eyck interestingness field, a clear boundary can be seen between the inside and outside of the building, when people see more of a complex mix of elements

versus only one-sided walls are seen in the scene. The publicness results and spaciousness results are undoubtedly more complex. Comparing the publicness and spaciousness of the same project, the overall trend is close, but there are subtle differences, mostly in regions such as corners, parallel walls, and other interior space. It seems although the two adjectives both have a similar meaning as they are all more or less related to the "openness" of space, there are also some nuances. A human understands space not only by form but also by the affordance of space. Publicness suggests that space leads to some traffic activity, whereas spacious suggests more of a space where different activities can be done inside. Traffic-related space such as stairs and corridors comparatively have higher publicness rating, as it can be seen as a major node or pass way for traffic. Spacious space is more closely related to the concept of "openness," as can be seen that the exterior regions mostly have higher spaciousness ratings. But in some areas where space is enclosed well, they also have high spaciousness scores since the spaces are integral and suggest that they have a right capacity for activities. Such cases can be seen in big interior spaces of both the three designs.

The violin graph, on the other hand, suggests a parallel comparison for the three designs. It can be seen that in the eye of the participants, Paviljoen van Aldo van Eyck is a little more interesting than the other two, as the mean and the third quartile values are higher. The spaciousness rating, on the opposite, is the lowest for Paviljoen van Aldo van Eyck. One interesting thing to notice is, the variances of Paviljoen van Aldo van Eyck ratings are much smaller than the two Mies designs, whereas the two Mies designs have very similar rating distribution, in both the mean values and the variances.





## 5 DISCUSSION

As initially discussed in Section 2.1, the “physicality” and “spatiality” of space are two closely integrated parts that cannot be separated and are both concrete features of space. The two parts of space are bridged harmoniously by human perception. With the attempt to develop machine systems, “spatiality” may also be connected to “physicality” through artificial neural networks.

Although the systems are mostly trained with small and artificial datasets (except for scene classification) and applied to limited topics, we can see the potential of this methodology in architectural design studies. The results of both the tests on the modeling software and the analysis of existing buildings show that the systems extract features from space samplings and produce results that are similar to human perception. The following paragraphs discuss the pros and cons of this method I observed as they were developed, followed by possible future improvements, developments, and the contribution of this thesis.

As proposed in Section 1, training machine learning systems to get “perceptions” of space is no more than a simulation of the behaviors of human perception. Current artificial intelligence systems still work on well-defined tasks, which makes little difference compared to traditional computation methods. The actual concepts of perception are far beyond what can be covered by several prediction problems. Human perception is non-deterministic. From perception, emerging understandings and predictions, even imaginations can be inferred by human brains. In that sense, a machine’s perception system is not true perception, but simulations of perception in several aspects.

In addition, the experiments and methodologies proposed in this thesis were conducted using simplified settings. Unlike a real architectural space, which involves aspects such as lighting and material, only the geometric aspects of space were captured. Currently, many of the systems proposed work as proofs of concept within laboratory settings and have not yet been utilized in practice. However, such space perception simulation systems show great potential in assisting with spatial design and its analysis. The methods may not cover all aspects of a human’s perception of space, but once a system is built so that a specific aspect can be simulated, thanks to the efficiency of computers, results can be acquired promptly and in great detail considering historical datasets once adequately collected and organized.

As seen in the application of the interactive modeling software, the system yields insights in

real time as a designer operates in the modeling interface. Furthermore, insights such as ratings by different subject groups are beyond what a regular designer can imagine by himself or herself. These insights also can be a part of the design pipeline. The interface can suggest, as seen in the interactive modeling software, predefined shapes to the users that can be directly applied to the current model, and the suggestions are made based on the scene classification results. In addition, counting on the efficiency of the perception simulation systems not only results from a single sampling location, insights of samplings covering every corner of a model can be computed in no time. Thus, a designer can get an overall quality report of the entire design down to every part of the operation “on the fly.” This application can be seen in the analysis reports of the three selected buildings. Such reports can be generated in real time and consider every subtle change of a design process. The system’s involvement in design can then go beyond the current shape suggestion once a larger dataset is constructed and more perception aspects are introduced.

As for the system itself, the network may not be deep enough, meaning there might still be potential to acquire better feature extraction performance with better-crafted networks. A larger training set with greater variety can also help in the generalization of the networks and would also help to avoid overfitting. Considering the experiments on existing designs, evenly distributed samplings are utilized in the models, acquiring a distribution of “seed-spaces.” In reality, architects may design a building based on the sequences of spatial experience. It could also be interesting to run sampling on a designated walk-through in the space, along with analyzing the transformations of space in vector representation space. This kind of analysis can not only yield a space-type distribution, but the variation of different spaces along the sequence. This could lead to interesting research topics in circulation design.

Due to the difficulty and comparatively higher cost of real building sampling, the systems are trained with artificial data generated from digital models. In this way, huge datasets can be acquired easily; however, compared to real space, artificial data encapsulates limited features and simplified shapes, which may lead to overfitting and other issues. With the aid of 3D-scanning techniques and photogrammetric sampled models, potentially training data based on real space data can be constructed, resulting in more convincing and better-generalized prediction models.

Additionally, for the space rating system, a limited amount of survey data was collected, which more or less prevented the system from converging. This issue is acceptable when

computing general ratings; however, it can be less applicable when computing ratings by individual subject groups. In the extreme case, category “architect” only has about one-third of the overall data. In cases where overlapping categorical restrictions are applied, such as “male architects who are under 30 years old,” the data shortage can be critical. For that, the construction of an extensive dataset can be the only solution.

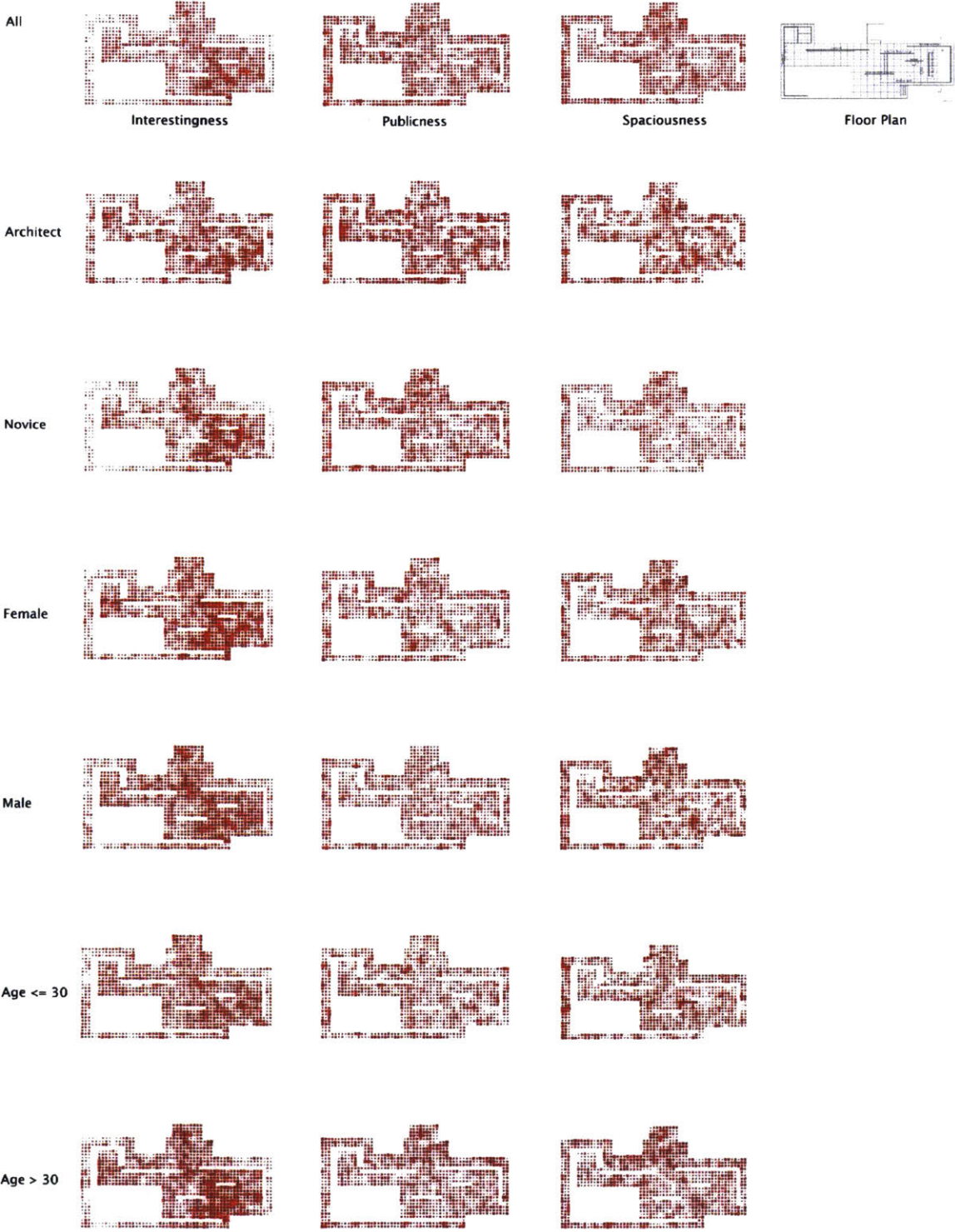
Another problem with the current space rating survey is that participants are viewing the scenes in their browsers or on mobile devices by either dragging the mouse or touching the screen. This subjective engagement of space may bias the perceptions as the actual space engagement is more of an objective experience: situated inside of a space and immersed in the space as it is. A VR survey may be a better alternative to resolve this issue, although such a survey may have some demands in aspects such as equipment and accessibility.

Comparing a machine’s perception and a human’s perception would also be interesting and would provide feedback for improving the systems. One approach would be to use VR equipment to test a human subject with an identical space and compare the result of the human subject with the presumed result of the system. A VR experiment would provide an immersive experience for a subject, yielding a more convincing spatial feeling.

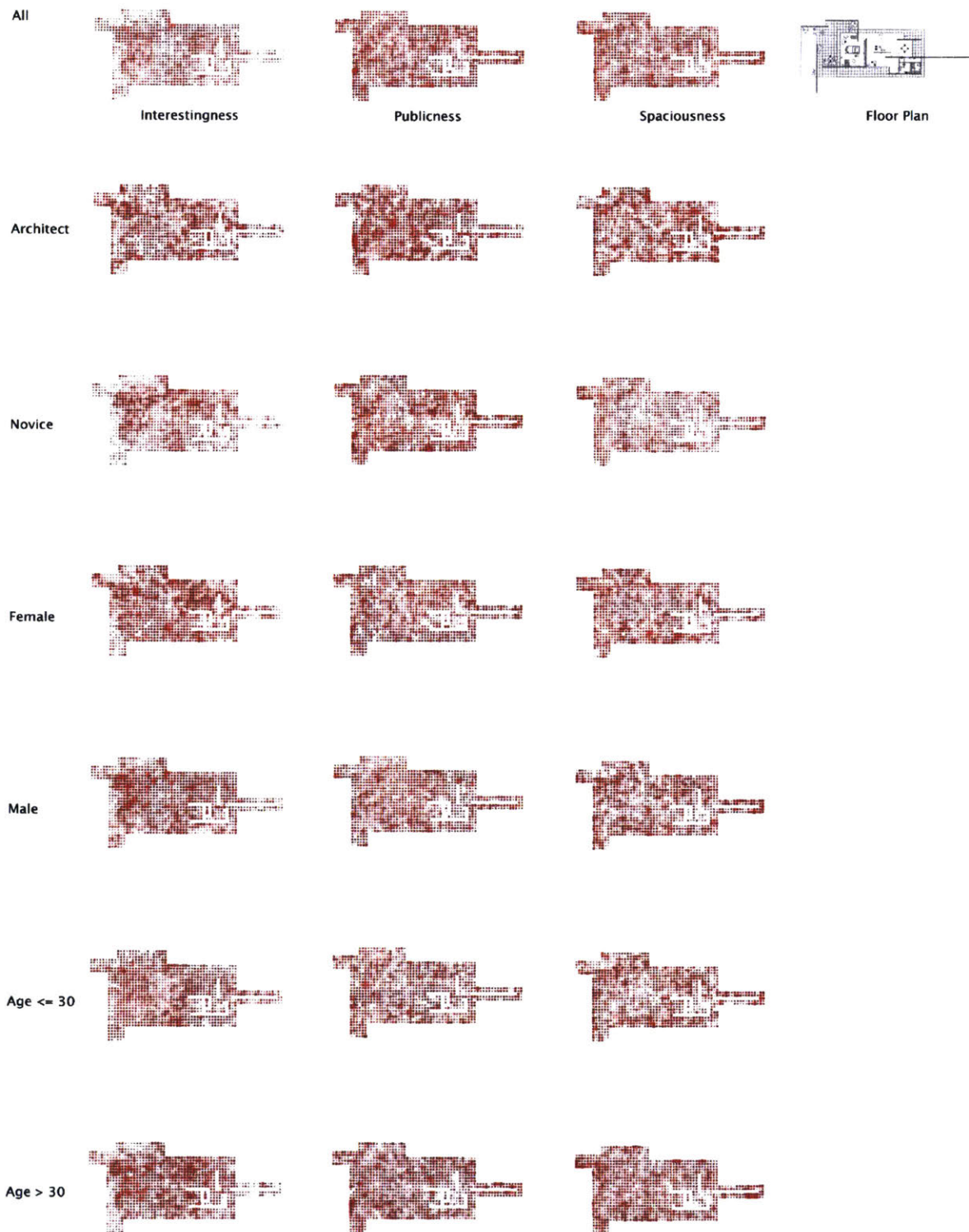
The contribution of this thesis is fourfold. First, it introduces a sampling method based on 3D isovists that generates a panoramic depth image that can be used to represent a 3D space from a specific observation point. Second, it employs machine learning and artificial neural networks to extract features from the space samples, which is then utilized for prediction. Third, it proposes a survey method that can collect space sentiment data. Fourth, it demonstrates a few applications where this space prediction method is applied and how it helps with design and its analysis.



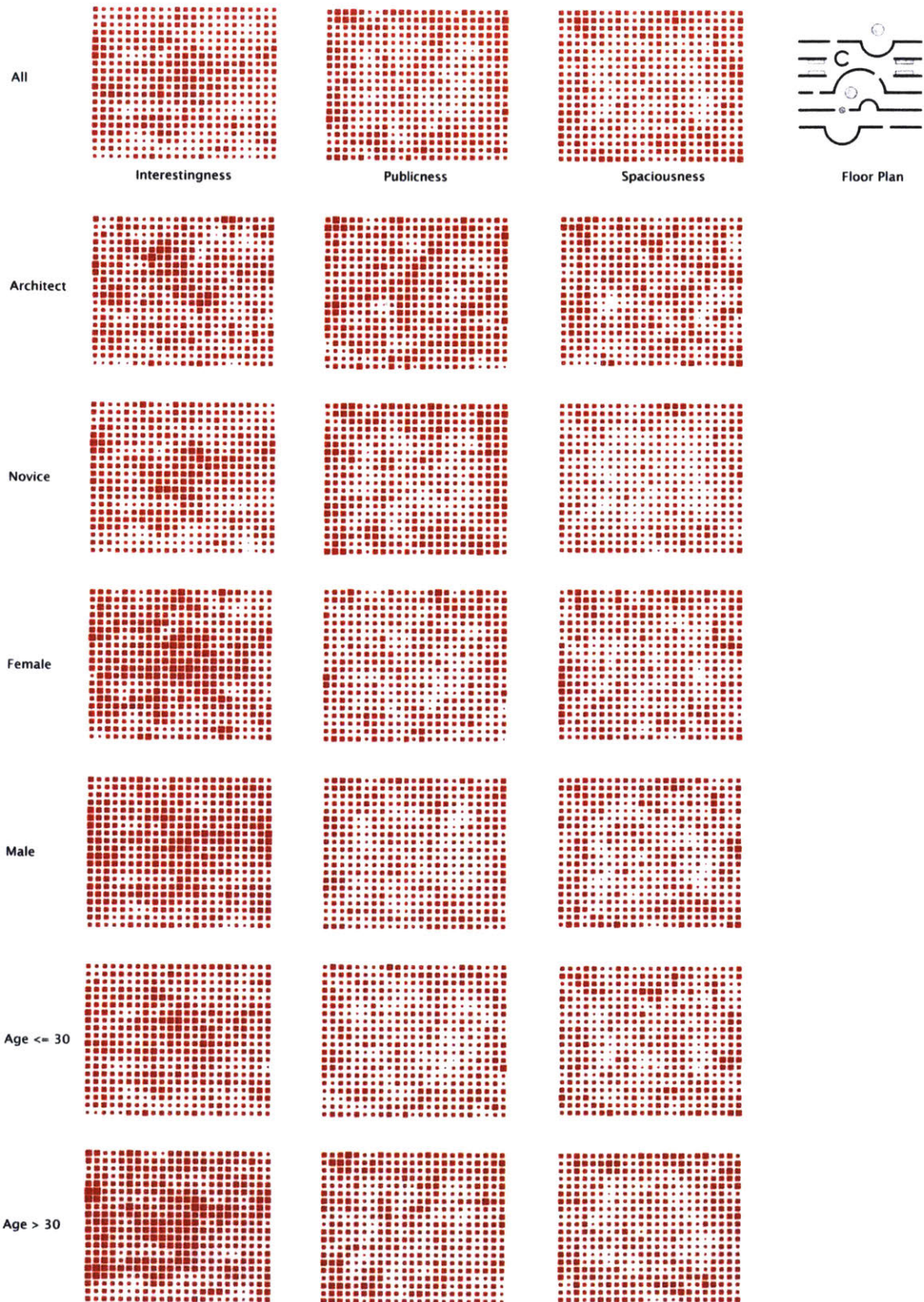
# 6 APPENDIX



**Figure 6-1:** Space rating of Barcelona Pavilion by different subject groups. Calculation based on system introduced in Section 3.8.



**Figure 6-2:** Space rating of Exhibition House, Berlin by different subject groups. Calculation based on system introduced in Section 3.8.



**Figure 6-3:** Space rating of Paviljoen van Aldo van Eyck by different subject groups. Calculation based on system introduced in Section 3.8.





## **BIBLIOGRAPHY**

"The History of Blueprints - PlanGrid Construction Productivity Blog." 12 Apr. 2016, <https://blog.plangrid.com/2016/04/the-history-of-blueprints/>. Accessed 20 May. 2018.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

Kheradpisheh, Saeed Reza, et al. "Deep networks can resemble human feed-forward vision in invariant object recognition." *Scientific reports* 6 (2016): 32672.

Schmarsow, August. "The essence of architectural creation." *Empathy, Form and Space—Problems in German Aesthetics* 1893 (1873): 125-148.

Forty, Adrian, and Adrian Forty. *Words and buildings: A vocabulary of modern architecture*. Vol. 268. London: Thames & Hudson, 2000.

Palladio, Andrea. *The four books of architecture*. Vol. 1. Courier Corporation, 1965.

"Spotlight: Tadao Ando | ArchDaily." 13 Sep. 2017, <https://www.archdaily.com/427695/happy-birthday-tadao-ando>. Accessed 20 May. 2018.

Ching, Francis DK. *Architecture: Form, space, and order*. John Wiley & Sons, 2014.

Stiny, George. "Introduction to shape and shape grammars." *Environment and planning B: planning and design* 7.3 (1980): 343-351.

Marr, David. "Vision: A computational investigation into the human representation and processing of visual information. MIT Press." *Cambridge, Massachusetts* (1982).

Jackendoff, Ray. *Consciousness and the computational mind*. The MIT Press, 1987.

Benedikt, Michael L. "To take hold of space: isovists and isovist fields." *Environment and Planning B: Planning and design* 6.1 (1979): 47-65.

Steadman, Philip. "The contradictions of Jeremy Bentham's Panopticon penitentiary." *Journal of Bentham Studies* 9 (2007): 1-31.

De Boer, Pieter-Tjerk, et al. "A tutorial on the cross-entropy method." *Annals of operations research* 134.1 (2005): 19-67.

Oquab, Maxime, et al. "Learning and transferring mid-level image representations using convolutional neural networks." *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014.

- Donahue, Jeff, et al. "Decaf: A deep convolutional activation feature for generic visual recognition." *International conference on machine learning*. 2014.
- LeCun, Yann, et al. "Backpropagation applied to handwritten zip code recognition." *Neural computation* 1.4 (1989): 541-551.
- Zhou, Bolei, et al. "Learning deep features for scene recognition using places database." *Advances in neural information processing systems*. 2014.
- Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks." *science* 313.5786 (2006): 504-507.
- Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." *Advances in neural information processing systems*. 2013.
- Antol, Stanislaw, et al. "Vqa: Visual question answering." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.
- Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2018): 834-848.
- Chang, Angel X., et al. "Shapenet: An information-rich 3d model repository." *arXiv preprint arXiv:1512.03012* (2015).
- Harwath, David, Antonio Torralba, and James Glass. "Unsupervised learning of spoken language with visual context." *Advances in Neural Information Processing Systems*. 2016.
- Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." *Journal of machine learning research* 9.Nov (2008): 2579-2605.
- Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.
- Hawkins, Douglas M. "The problem of overfitting." *Journal of chemical information and computer sciences* 44.1 (2004): 1-12.
- Silberman, Nathan, et al. "Indoor segmentation and support inference from rgb-d images." *European Conference on Computer Vision*. Springer, Berlin, Heidelberg, 2012.
- Silberman, Nathan, and Rob Fergus. "Indoor scene segmentation using a structured light sensor." *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011.
- Armeni, Iro, et al. "Joint 2D-3D-Semantic Data for Indoor Scene Understanding." *arXiv preprint arXiv:1702.01105* (2017).

Buhrmester, Michael, Tracy Kwang, and Samuel D. Gosling. "Amazon's Mechanical Turk: A new source of inexpensive, yet high-quality, data?." *Perspectives on psychological science* 6.1 (2011): 3-5.

Team, Pytorch Core. "Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration." (2017).

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

"Softmax function - Wikipedia." [https://en.wikipedia.org/wiki/Softmax\\_function](https://en.wikipedia.org/wiki/Softmax_function). Accessed 21 May. 2018.

Von Moos, Stanislaus. *Le Corbusier: elements of a synthesis*. 010 Publishers, 2009.

"Ludwig Mies van der Rohe | American architect | Britannica.com." 25 Apr. 2018, <https://www.britannica.com/biography/Ludwig-Mies-van-der-Rohe>. Accessed 21 May. 2018.

Naik, Nikhil, et al. "Streetscore-predicting the perceived safety of one million streetscapes." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2014.

Herbrich, Ralf, Tom Minka, and Thore Graepel. "TrueSkill™: a Bayesian skill rating system." *Advances in neural information processing systems*. 2007.

"JavaScript." <https://www.javascript.com/>. Accessed 22 May. 2018.

Cabello, Ricardo. "Three.js." URL: <https://github.com/mrdoob/three.js> (2010).

Bostock, Michael. "D3.js." *Data Driven Documents* 492 (2012): 701.

"csg.js." <http://evanw.github.io/csg.js/docs/>. Accessed 22 May. 2018.

"TreeModel." <http://jnuno.com/tree-model-js/>. Accessed 22 May. 2018.

"Firebase." <https://firebase.google.com/>. Accessed 22 May. 2018.