Towards a Machine Learning Framework in Spatial Analysis

Cecilia Ferrando
School of Architecture
Carnegie Mellon University
Pittsburgh, PA 15213
cferrand@andrew.cmu.edu

Abstract

How can we build a Machine Learning model of learnable spatial rules? How would a Machine Learning framework prove a useful tool in the analysis of architectural qualities? Inspired by the long-open question whether it is possible to build an AI design assistant, this thesis researches a Machine Learning framework for spatial analysis of floor plans. It shows that Machine Learning algorithms trained on large datasets of plan configurations have the potential to characterize abstract architectural qualities in terms of quantifiable spatial features.

Two commonly adopted techniques used in spatial analysis are isovists, introduced by Michael Benedikt (1979), and graph theory, explored in architecture by Christopher Alexander, but dating back to XVIIIth century France. These are contextualized within Space Syntax, the set of theories and methods studying spatial configurations and their cultural implications. Such techniques convey different types of information on architectural space - visual connectivity on one side, hierarchies and accessibility on the other. In this thesis, I aim at relating spatial features to architectural qualities through a Machine Learning algorithm. Specifically, I look at the quality of architectural privacy in homes.

One obstacle in building a Machine Learning framework for the analysis of architectural space is that a large amount of labeled data is needed. In order to prove the feasibility of building a large dataset of floor plans labeled according to a set of spatial features, a software extracting spatial features out of image data is outlined in its structure. Finally, the technical aspects of this newly proposed Neural Network framework for spatial analysis are presented and discussed. From a proof-of-concepts experiment, it emerges that when statistical analysis is run on a relatively small dataset of spaces in house floor plans, patterns relating Space Syntax features and the level of intimacy of different rooms in a house floor plan are found. At the same time, the limits in these results reinforce the need for a more complex function approximator (such as a Neural Network) to detect spatial patterns.

By presenting a novel AI approach to spatial analysis on the floor plan, this thesis opens the floor to both analytic and generative applications of Machine Learning in architecture.

Acknowledgements

I wish to express here my gratitude to the people who supported my work in different ways.

First and foremost, I want to thank my advisor, Daniel Cardoso Llach, for his guidance, insightful feedback, and encouragement throughout all the phases of this work; Daniel's expertise and knowledge were a constant inspiration to think critically and to improve my research skills. Thank you for helping me shape my work.

I would also like to thank Peter Scupelli, for his insightful advice on Space Syntax theories and methods.
Thank you to Simon Lucey and Brian Okorn, for offering their advice on Computer Vision and Image Processing matters in this thesis.
Thank you to Susan Finger, who kindly provided precious feedback on my Matlab code.

I wish to thank all the experts who provided feedback on my work at my public presentations, their advice was fundamental and helped me better frame my questions and goals.

Thank you to my cohort in the MS in Computational Design program, for sharing with me the best and the hardest moments into this program.

I am especially grateful to the Fulbright Program, for giving me the necessary support to embark into the most precious opportunity of growth and enrichment I have ever had.

My deepest gratitude goes to Nic Dalmasso, for many reasons. For being my Data Science guru in this thesis. But first and foremost for being with me in so many important ways.

Finally, all my gratitude goes to my family, who believed in my dream and supported my graduate studies at Carnegie Mellon University. None of this would have been possible without your support and love.

This thesis is dedicated to the memory of my uncle Carlo

Contents

Part I

First part

1 Introduction

1.1 Can a machine learn spatial quality?

When designing the layout of different spaces in a floor plan, an architect implicitly adopts spatial rules. These rules relate to the accessibility of spaces, the functions that each room is designed to have. Depending on the function - which can be more or less private - the architect takes decisions over the location, connections and visual openness of each space in a building. This results in each room having precise characteristics or qualities.
Considering the long-open question whether it is possible to build an automated system to assist architectural design, it is particularly important to inquiry into whether computational processes can get to the point of learning these qualitative choices, and reproduce them in the context of an AI program supporting design decisions.
Machine Learning has recently enabled pattern recognition in datasets of different kinds. I want to show that introducing automated learning of architectural information can open new analytic and design possibilities in architecture.
The idea of introducing AI in architectural design is not new. In fact, the MIT CAD project (1959-1970) was one of the first projects expressing a desire for introducing an automated assistant in design and architecture[1]. More recently, with the fast advancements of computational capabilities, these possibilities have become closer realities.
What are the prerequisites to build an intelligent architectural design assistant? This thesis starts from the consideration that there are two possible ways of building design intelligence. First, architects could hardcode their decision rules in a computer program. The limitation of this setting is that it would result in a deterministic intelligence, biased by the fact that each designer has some level of subjectivity in his/her design decisions. Hardcoding those would result in a replica of those rules in an automated context - raising the question whether *this* should be called intelligence rather than automatism. Machine Learning takes a different - more promising - approach to the problem. By observing large amounts of data, an algorithm can progressively learn what the patterns in a certain context are, and therefore what rules are most likely to produce a good result. This approach has the potential of learning architectural rules from scratch, with no need for an *a priori* enforcement of what those rules are. This statistical approach to intelligence comes with one important *caveat*: the high dependence of the inferred results from the input data. A biased dataset results in a non-generalizable result. However, this dependence can be positively leveraged in architecture by forcing bias in a dataset (for instance, analyzing a dataset of Parisian villas from the XVIII century, as opposed to a generic randomly sampled dataset). This would lead to the discovery of genotype specific rules and characteristics.
The question, at this point, is whether or not there can be a sufficiently large labeled datasets to allow for automatic learning. The answer is no. This is mainly due to the fact that labeling, for example, an architectural floor plan not only requires large amounts of time and expertise, but also when it comes to the qualities of architectural spaces these do not have an objective definition, and they are left to the subjective qualitative consideration of experts in the field.
This thesis takes a look at the most commonly used document encoding architectural information - the plan. It provides a proof-of-concept of the possibility of extracting a set of spatial measures of rooms in a plan starting from a plan image as an input. It shows that a large labeled dataset of architectural plans encoding spatial qualities can be used as input in a Machine Learning framework to train a machine to detect the patterns relating spatial features and architectural qualities. A Neural Network system will be sketched that takes as an input raw architectural plans of houses and targets patterns in spatial privacy and openness based on a matrix of spatial measures. By implementing a statistical analysis on a sample dataset, I will show that some qualitative aspects of spatial configurations can be statistically learned.
In short, this thesis will address the following research questions: "How can we build a Machine

[1] For reference, see Daniel Cardoso Llach, *Builders of the Vision: Software and the Imagination of Design*, 2005

Learning model capable of learning spatial qualities? How would a Machine Learning framework prove a useful tool in the analysis of architectural space?".

1.2 Outline

The thesis is subdivided in the following parts:

- Part I - This part provides the theoretical background to spatial analysis of architectural floor plans, with an emphasis on the techniques and practices of Space Syntax. Specifically, the graph representation and the isovist techniques are presented in detail, together with examples of Space Syntax analysis. The concept of architectural privacy will be discussed, and how it can be related to Space Syntax features.
- Part II - This part tests our hypotheses on spatial the quality of privacy through a preliminary experiments. It addresses the problem of learning from large datasets and the obstacle represented by the lack of substantial labeled datasets to use in a Machine Learning framework. To overcome this obstacle, this part presents a demo program to process image floor plans so to extract the target spatial measures. Guidelines for a fully-working program are provided.
- Part III - This part describes the Machine Learning workflow allowing for spatial quality learning from labeled datasets of floor plans. In the context of this thesis, the focus is on floor plans of houses and living units. A simple example of statistical analysis is presented. Results from this example confirm the existence of patterns relating privacy to spatial features and further motivate a Machine Learning framework. Finally, this part discusses the advantages of a Machine Learning framework for Space Syntax analysis in the context of an AI design assistant, as well as the limitations, potential extensions and applications of this work.

2 Background

2.1 Space Syntax

Space Syntax is set of theories and techniques for analyzing architectural space in its functional characteristics. Space Syntax has the ambition to build a theoretical model of human space, its structure and its social implications.
Space Syntax sees architecture as configuration - no matter the scale. A building, as well as a city, are configurations because they are composed of parts and relationships.
The '70s and '80s were prolific in Space Syntax research. It is worth acknowledging that Space Syntax took different nuances in different schools of thought. Christopher Alexander is interested in the abstract hierarchies and patterns of spatial arrangements [1]. American mathematicians George Stiny and James Gips (1971) formalized a computational theory of design based on form. Their theoretical system - Shape Grammars - takes a view of designed objects as a visual algebra where shapes are the object of calculation [2]. While taking this mathematical view of shape, Stiny and Gips actually aim at decoupling computation from computers, bringing the computation of Shape Grammars to a level of higher abstraction. In applying Shape Grammars to the study of Palladian villas, Stiny and Gips are interested in addressing the following questions: can we characterize the common traits among different Palladian buildings? [3] Can we learn the generative rules of a Palladian villa?
The University of College London has also been a very active center of research in Space Syntax. The approach to Space Syntax at UCL is slightly different from both Christopher Alexander's theory and from Stiny's Shape Grammars. As Bill Hillier himself claims in the introduction to *The Social Logic of Space* [2], his approach gives up the high mathematical rigor of Shape Grammars and formalizes "syntactic generators", which can describe space relations independently from shape - which is instead extremely important in Stiny's work. Shape Grammars are in fact computational frameworks for studying (and applying) the generation of architectural form. Bill Hillier and the UCL school aim at a different intellectual goal, which is rather analytic. While Shape Grammar researchers are interested in the question "how can we compute with shapes?", the UCL school is more concerned with the social and cultural meaning of spatial configurations, and in its practical validation.
In this thesis, my goal is to focus on the automation of the analytic practices of Space Syntax[4]. For this reason, Bill Hillier and the UCL Bartlet School of Architecture research are kept as main references to develop a discourse on the meaning of spatial quality.

2.2 Graph representation

One important analytic practice in Space Syntax leverages the representation of spatial components as nodes in a graph, so to analyze their relationships and connectivity.
The origin of Graph Theory is traditionally dated back to the seventeenth-century paradox of the Bridges of Königsberg (see Figure 2). This is a mathematical problem concerning seven bridges separating four landmasses and a Knight's desire to cross each bridge only once while moving in a continuous sequence [3].
Although Graph Theory is a broad set of theories in mathematics, we are concerned with the use of graph representations of spaces and their connections. It is not until the 1960s and 1970s that we can observe a substantial growth in interest in the applications of Graph Theory to a variety of analytic problems in architecture and geography (Harary 1969). In such context, the work by Christopher Alexander in the 1970s plays a fundamental role. In Alexander's mathematical view of space, design problems can be represented as graphs and trees. Alexander proposed a simple computational model of architectural design (1964) [4], soon followed by an application of graph theory to the analysis of urban connectivity [5]. A few years later, Alexander defined a pattern-based (rather than "graph based") approach to design (Alexander et al., 1977) [1].

[2] Daniel Cardoso Llach, *Data as substrate / Data as interface: the poetics of machine learning in design*, in: Machine Learning. Medien, Infrastrukturen und Technologien der Künstlichen Intelligenz (Machine Learning. Media, Infrastructures and technologies of Artificial Intelligence.) Edited by Christoph Engemann and Andreas Sudmann. Transcript 2017

[3] This question is similar to the problem of genotypes in Hillier and Hanson's work at UCL

[4] Although this is the focus of my thesis, I believe that a Machine Learning framework for Shape Grammars would be a very interesting research field. Brief comments on the generative applications of a Machine Learning framework for spatial analysis are also sketched in Chapter 11.

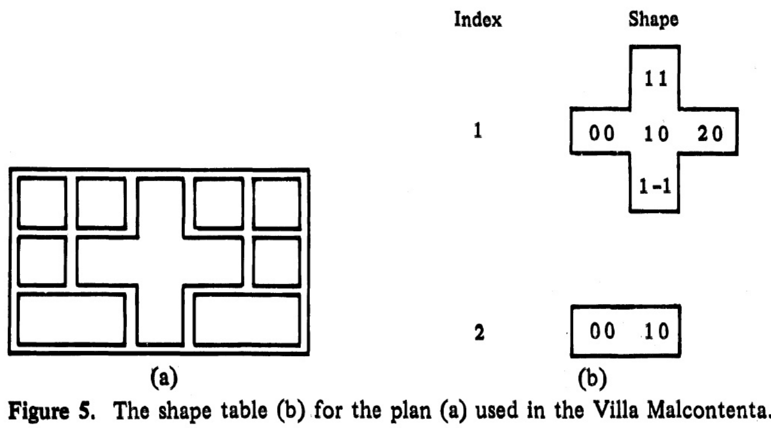


Figure 1: A figure from Stiny and Gips (1978)

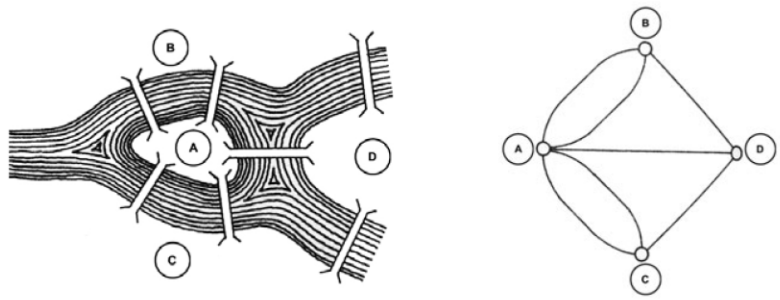


Figure 2: The Seven Bridges of Königsberg and their graph representation [3]

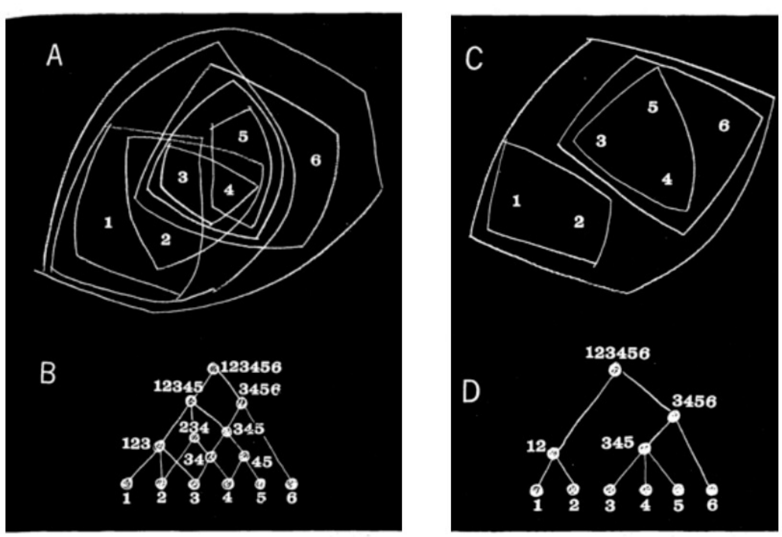


Figure 3: A semi-lattice and a tree. From Christopher Alexander, "A City Is Not a Tree, Part 1," Architectural Forum 122, no.4 (1965), p. 59

While the 1970s saw the emergence of a graph theory of architecture and space, the practice of studying the graph properties of spaces implicitly dates back to the French intellectual debate on architecture in the XVIIIth century, when Jacques-François Blondel wrote *De la distribution des*

maisons de plaisance and the historically famous *Cours d'Architecture*. The XVIIIth century in France was characterized by the emergence of the need for different levels of intimacy in the rooms of mansions and palaces [5]. This brought French architects of that time to study architectural distribution in depth, and design according to spatial hierarchies [6].
Space Syntax often uses justified plan graphs as a way to represent spatial hierarchies. The very first step in the process of analyzing a space is the extraction of a convex map (see Figure 4). The convex map translates a spatial plan (either architectural or urban) into a diagram that represents its configuration. Citing Hillier and Tzortzi (2006) [6],

> *Spatial layouts are first represented as a pattern of convex spaces, lines, or fields of view covering the layout (or ... some combination of them), and then calculations are made of the configurational relations between each spatial element and all, or some, others.*

A convex map is essentially a tool to identify spaces and connections from plans (see Figure 4).

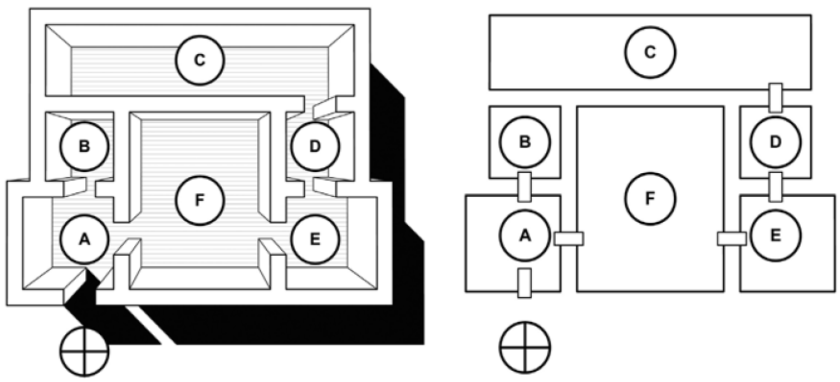


Figure 4: Example of convex map of a plan for Villa Alpha, Ostwald (2011) [3]

However, the convex map does not convey any graphic information on the hierarchy of the configuration. This is instead conveyed through justified plan graphs ("JPG", or simply "justified graphs"). A justified graph is a graph in which a particular space is selected as root, and the others are displayed above (or below) in a ramified structure in levels according to how many spaces one must pass through to reach them (see Figure 7 and Figure 6). If, in order to reach room X, we need to traverse several spaces, room X is considered segregated, while conversely if it can be reached by traversing few spaces, than it is considered integrated. A graph representation makes it possible to assign values to represent these spatial properties. In fact, graphs are not merely a representation tool to clarify the visualization of a configuration, they are rather a key analytic tool, used to study 'depth' and 'rings' in architectural space configurations.
Some meaningful graph measures that can be used in spatial analysis are:

- depth, or integration: a space is at depth 1 from another if it is directly accessible to it, depth 2 if it is necessary to pass through one intervening space in order to move from one to the other, at depth 3 if a minimum of two spaces must be passed through, and so on[7]
- betweenness centrality: counts the number of shortest paths passing through a given node
- degree centrality: counts the number of edges (connections) a node has
- eigenvector centrality: defines important nodes based on the connection to other important nodes

[5] The quality of intimacy will be further discussed and will be object of further study in this thesis

[6] "[The French] great achievement was to perfect the apartment as a sequence of spaces of ever increasing comfort and intimacy [...]. Even in the grandest of seventeenth-century mansions, rooms had been used indiscriminately by many members of a family and by passing servants. There was an easy promiscuity. But in the eighteenth century, with the opening up of the realm of feeling and especially individual sentiment, privacy took on a new value" From Robin Middleton's introduction to The Genius of Architecture, Nicholas Le Camus de Mezieres, The Getty Center, 1992

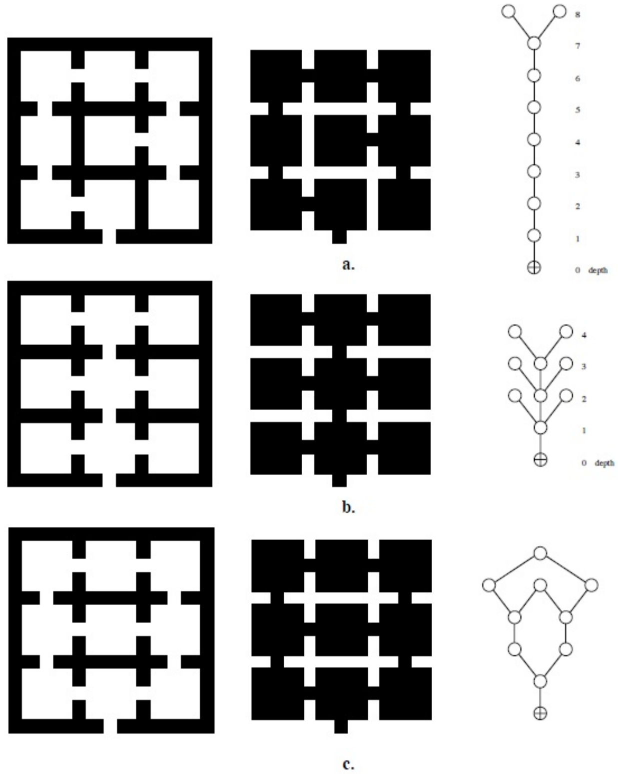


Figure 5: A graph analysis from Bill Hillier's *Space is the machine*

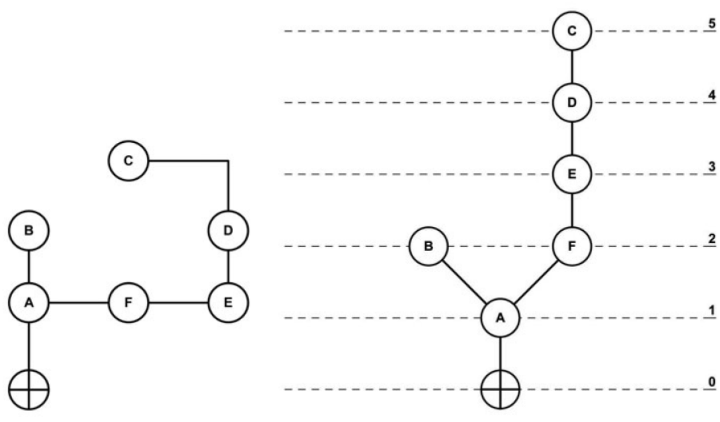


Figure 6: Examples of justified graph for Villa Alpha, Ostwald (2011) [3]

At the UCL school, integration has been extensively used in the analysis of the cultural implications of spatial configurations. In *Decoding Homes and Houses*, Julienne Hanson highlights how, in cases where analysis was run over statistically reliable samples of real house configurations in various vernacular traditions, different functions and activities were systematically assigned to spaces with different levels of integration. This way, functions became increasingly intertwined with the properties of their assigned space, forming what could be called a 'cultural fingerprint' of house configuration. This crystallization of cultural meaning in specific spatial configurations is referred to by Hillier and Hanson as 'genotype'.

Beyond the cultural differences, integration of the spaces with respect to the exterior (room 0) is

engineered so to filter and control the access to certain spaces, or on the opposite side so to open and invite. On the base of these nuances of privacy (which will be discussed in deeper detail in the next Chapter), different space-types can be adopted, such as the terminal spaces, the bi-permeable spaces, rings, enfilades, intersections [8].
Graphs are important tools in the understanding of spatial use as well. As highlighted by Peponis

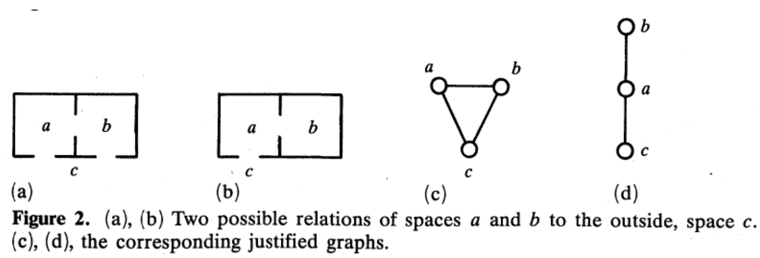


Figure 7: Examples of justified graph, from "Ideas are in things", B. Hillier et al. (1986) [7]

and Wineman in [9], spatial structure does influence behavior. For example, spaces that are highly accessible (in graph terms, this could be a highly central node) also have higher probability of being used for movement. This is true for buildings of different nature, as well as for the built environment in general. In fact, Hillier and associates (1987) registered a 0.75 Pearson correlation between integration and the square root of the number of pedestrians in four London squares. In office environment, for example, high integration was found to be highly correlated with human interaction.

2.3 Visual Fields

In Space Syntax, it is fundamentally important to take into consideration spatial properties from the point of you of a situated observer. This is made possible through the use of axial lines, convex spaces and isovists. Isovists (or viewsheds) are polygon representations of the space visible from a determined point in space. It is typically represented on a plan view. They were theorized by Michael Benedikt at the University of Texas at Austin in 1979 [10], and have been object of interdisciplinary research since then, especially at the University College London. In particular, Sophia Psarra presents an extensive work on isovists in *Architecture and Narrative: The Formation of Space and Cultural Meaning* [11].
Although we will discuss isovists for two-dimensional polygonal spaces, the same ideas can be generalized to non-polygonal spaces and to three-dimensional spaces.
The isovist of point x consists of all the points y in polygon P that are visible from x (see Figure 8). Isovists are dependent on the observer's location in space. In order describe the visual characteristics

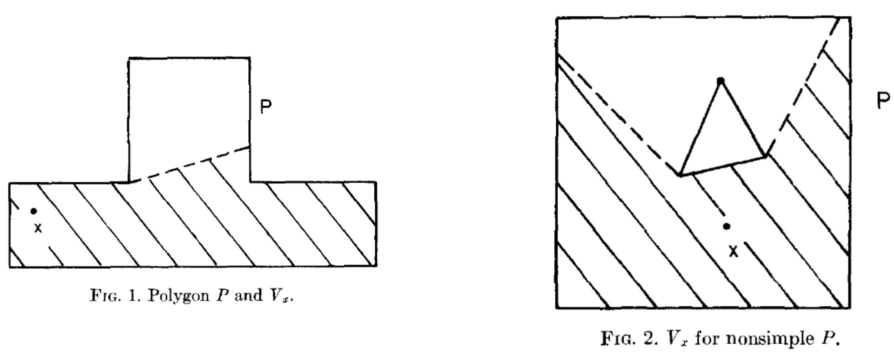


Figure 8: Example of isovist V_x in polygon P [10]

of an environment as a whole, Turner et al. (2001) proposed the technique of visibility graph analysis. Inspired by the graph-based representations adopted in social theories of networks, and by the small worlds analysis of Watts and Strogatz (1998), Turner and colleagues used isovists to derive a visibility graph of the environment - essentially a graph of mutually visible locations in a spatial layout that has been discretized in a grid of points. Through this discrete representation, they defined a set of measures of local and global spatial characteristics conveying information that well relates to our perception of the built environment (see Figure 9) [12]. Once we derived a visibility graph, a number

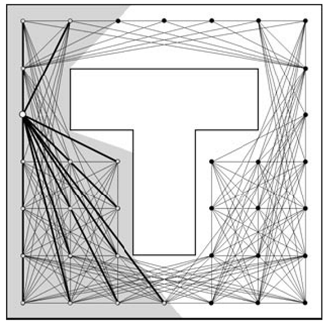
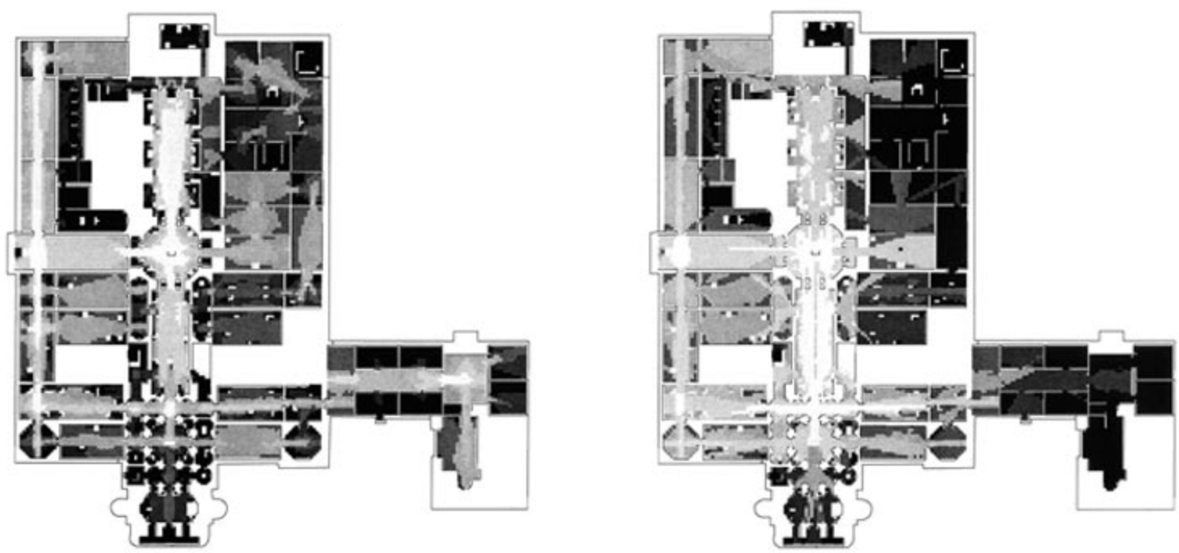


Figure 9: Visibility graph for a simple configuration, Turner et al. (2001) [12]

of different measures can be extracted that investigate the graph properties. A major reference for graph properties is Wilson and Beineke (1979). Turner and colleages (2001) focus on three measures of graph structural properties in particular:

- Neighbourhood size - The neighbourhood of a vertex is the set of vertices immediately connected to the vertex via one edge. Mathematically,
  $N_i = \{v_j | e_{ij} \in E\}$
  Notice that "if the set of generating locations covers the entire space (at some uniform resolution, so that for our purposes it fully describes the space), then this set can be thought of as equivalent to the isovist itself. Hence there is a one-to-one correspondence between the neighbourhood of a vertex in a visibility graph and the isovist from the location represented by that vertex."[12] In other words, the neighborhood size of a vertex in a discrete field of points in space corresponds to the isovist from that point, provided that the density of the grid is adequate.
- Clustering coefficient - The clustering coefficient is the number of edges between all the vertices in the neighbourhood of the generating vertex. This represents the number of lines of sight between all the points forming the isovist, divided by the total number of possible connections with that neighbourhood size. In isovist terms, this would be the mean area of intersection between the generating isovist and all the isovists visible from it [12]. The meaning of the clustering coefficient can be interpreted as the extent to which the neighborhood of a vertex is convex; if the neighbourhood of a vertex approximates a convex polygon, then the clustering coefficient will tend to one. The clustering coefficient is therefore a measure of the proportion of intervisible space within the visibility neighbourhood of a point, over the total possible intervisibility connections among its points. Mathematically,
  $C_i = \frac{|\{e_{ij} : v_j, v_i \in N_i \wedge e_{ij} \in E\}|}{k_i(k_i - 1)}$
- Mean shortest path length - The shortest path between two vertices in a graph is the minimum amount of edges that a visitor would have to traverse in order to get from one vertex to another. The mean shortest path length of a vertex is obtained by averaging the shortest paths length from that vertex to all other vertex in the system. In other words, it represents the average amount of steps required for all possible journeys starting at that point. The journey (path) between vertex A V_i and vertex B V_j is more formally defined by a sequence of vertices $(V_i, ..., V_n, ..., V_j)$, such that all consecutive vertices are connected by an edge in the graph. We define d_{ij} to be the shortest path, then the mean shortest path between V_i and V_j is
  $\bar{L}_i = \frac{1}{|V|} \sum_j^{V_j \in V} d_{ij}$
  In Hillier and Hanson's work, a similar measure is identified as "visual accessibility" of different spaces; in fact, the mean shortest path extends this idea to continuous spaces and gives an idea of how many space steps are required to gain visual accessibility between a start and an end point.

The advantage of adopting the mean shortest path to measure accessibility, *in lieu* of axial lines and convex spaces, is that these latter are not useful to identify variation across open-plan layouts. Moreover, the mean shortest path gives information about the global visual connectivity of an environment in its whole, and not merely locally.



(a) (b)

Figure 10: From Turner et al. (2001) [12]. a) Neighborhood size analysis of Tate Gallery. b) Pattern of the mean shortest paths

2.4 "Ideas are in things": Space, Function and Culture

As previously seen, over the past decades UCL researchers have dedicated a remarkable amount of effort to studying the social implications of spatial configurations [7], and to inquiring into the functioning of architecture at various scales - buildings and cities. A fundamental publication treating this subject is The Social Logic of Space, a book collecting a decade of work and research by Bill Hillier and Julienne Hanson [2]. The book presents a new set of theories and tools for the analysis and understanding of buildings and cities as products of a social logic. It defines a method of analysis of spatial patterns, which is applied both to the analysis of human settlement at the urban level and for the decoding of building interiors. The book also presents a new theory of the social dimension of spatial systems, addressing the question of what it is that leads different cultural systems to adopt different spatial forms.
The pioneering publication that opened a new way in the statistical study of house configurations is *Ideas are in things* (Hillier, 1987) [7]. In this fundamental publication, Hillier uncovers a quantitative and statistical approach to the decoding of "genotypical" similarities between houses which have apparently different plans, but share similar patterns of spatial integration/segregation. A study of this kind was run on a series of seventeen plans of Normandy farm-houses. Hillier's procedure followed these steps:

- Justified adjacency graphs were drawn for the minimum living complexes of the Normandy farm-houses in the dataset. The exterior was selected as root node. Examples of these graphs are reported in Figure 11.
- Second, without considering the functions assigned to each space, syntactic spatial patterns were studied.
- In a third phase, the spatial patterns were further analyzed to check how different functions were arranged within the spatial pattern as a whole.

Each of these stages of analysis shaped two kinds of considerations: the first are geographical statements, which convey information about the sample as a whole; the latter type are 'phenotypical' statements, which concern the individual dwellings in their peculiar characteristics. What emerged from this study was that in half of the plans the *salle commune* was found to be the most integrated space of the house. In the other half, that role was taken by either the *vestibule* or by transition spaces. Although there existed no obvious single house type within the sample, it is however evident that there existed at least one underlying spatial-functional 'genotype', which is common to the majority of the cases despite being concealed under different 'phenotypical' arrangements.
Following the same method, Hanson [8] analyzes the arrangements in vernacular settlements in different parts of the world, coming to similar conclusions. This suggests that spaces are formed to accommodate human activities under the conditions imposed by a certain social and cultural environment.

[7] As seen in work by Hillier and Hanson

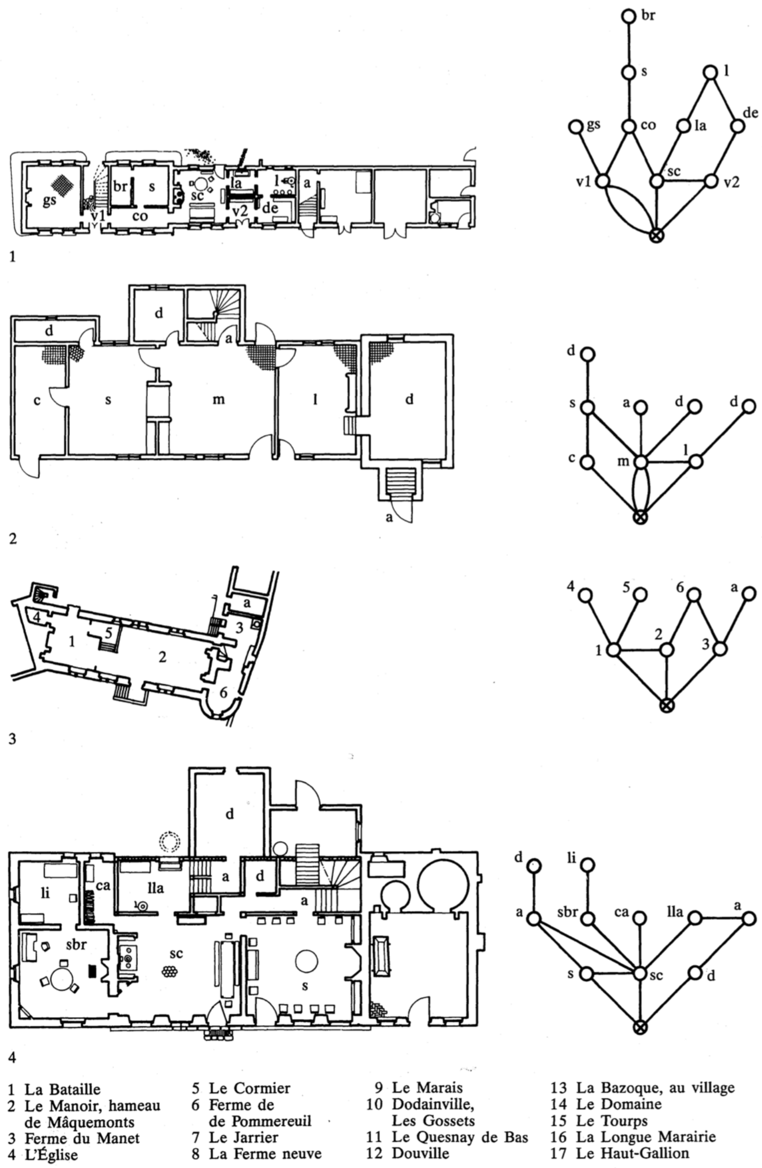


Figure 11: Excerpt of Hillier's dataset of Normandy farm-houses and related justified graphs

Another study applying this Space Syntax analysis methon was carried out by Bustard in 1999 [13] on a dataset of historical Anasazi houses in Chaco Canyon, New Mexico (see Figure 12. It emerged that the spaces where families gathered to eat their meals was also the most highly integrated space of the house 13. In this study, Space Syntax was used to interrogate the remaining of these historical buildings with the purpose of studying temporal differences in the use of space within that culture.

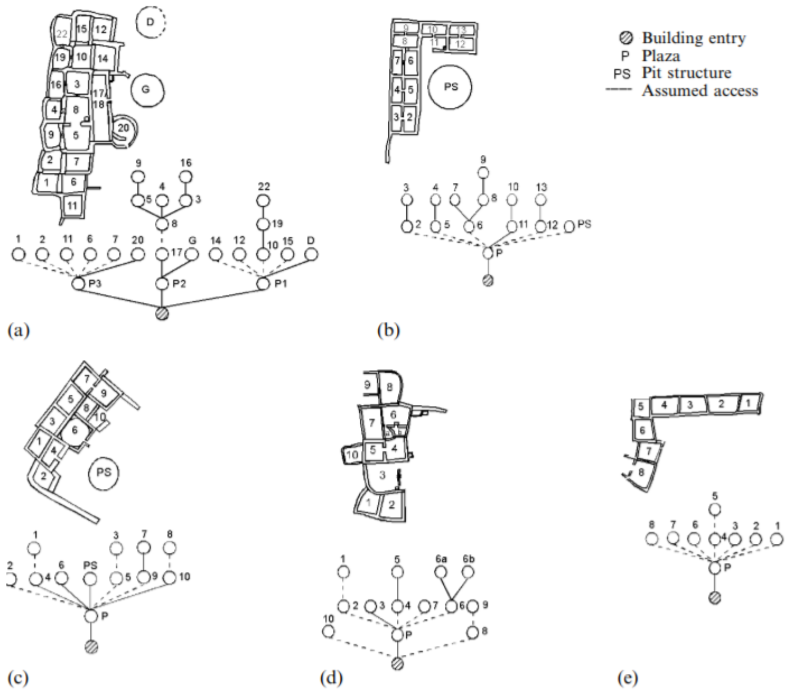


Figure 1. Access graphs for Classic Bonito Phase small houses: (a) 29SJ 627; (b) East Ruin; (c) Smith Ranch Ruin; (d) Turkey House; (e) Ruin 3.

Figure 12: Image from Bustard's study on Anasazi houses of Chaco Canyon, New Mexico

P > R > M > F > PS > N > S.

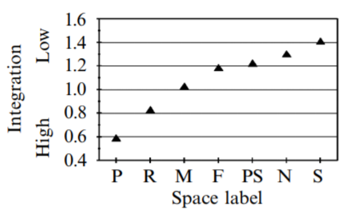


Figure 8. Ordering of mean integration values by space label for all small houses, $N = 307$ (P, plaza; R, rooftop; M, mealing bin; F, firepit; PS, pit structure; N, no floor features; S, storage facility).

Figure 13: Image from Bustard's study on Anasazi houses of Chaco Canyon, New Mexico

2.5 The limitations of Space Syntax

We saw how Space Syntax is a useful theory and method for investigating spatial patterns and their socio-cultural implications. I now want to put Space Syntax in critical perspective, and highlight the limitations we naturally incur into when attempting to assess spatial quality under the lenses of Space Syntax. The main limitation of this set of theories and techniques can be summarized in the incompleteness of the information encoded in a plan. This limitation manifests at different levels. First of all, any Space Syntax analysis relies on the fact that it is possible to identify a plan's convex spaces (rooms) and connections in an unequivocal way. The reality is of course more complex than this. Identifying the boundaries between convex spaces is often subject to the subjective evaluation of the plan reader. As a matter of fact, using Space Syntax methods on a modern, open-layout floor plan is particularly difficult due to the ambiguity in defining the boundaries between spaces. These might be determined by furniture or by function, rather than architectural barriers. In the context of a study or dataset analysis, this limitation can be overcome by pre-deciding a set of rules in the separation of convex spaces. Of course, such rules would result as axiomatic in the model.

One interesting limit that emerged from the study presented in Part III of this thesis is the problems associated with the assignment of a value to represent the connection between a pair of spaces. In the context of the graph analysis of a plan, a link is represented by a binary variable - a zero if no connection exists, a one if a connection does exist. However, there are pairs of spaces connected by more than one door. This of course creates a different spatial relation between the two spaces, which is entirely flattened out by a binary variable. Such nuances would deserve to be captured by an *ad hoc* measure, especially in studies that take as an object architectural types where traversal is the central activity (for example, a museum).

Another level of incompleteness is given by the limit in the type of information that a plan can convey. There are in fact spatial and cultural factors that are not captured by a plan alone - such as lighting and 3-dimensional appearance of the space. Edmund Leach (1978), for example, pushes this argument to extreme conclusion arguing

> *From my point of view the syntactic argument is meaningful and interesting, but I do not believe that one can immediately infer the generative syntax simply by looking at the lay-out of settlement patterns on the ground, and even if one could be sure of what the generative syntactic rules have been, one cannot infer anything at all about the society that makes use of the resultant settlement.*

I distance myself from the second statement in this argument. In fact, although it is unreasonable to think that cultural meanings can be inferred directly from a Space Syntax analysis, we do claim that Space Syntax is an effective and useful opportunity for exploratory studies that highlight the cultural differences in spatial use. This has been highlighted in the previous Chapter. Part III will also demonstrate this idea more in the detail.

2.6 Motivation

Literature on Space Syntax studies is fairly extended in the world of Architecture and Urban studies. So we might ask ourselves the question why it is worth exploring requisites and potentials of a Machine Learning framework in the context of the analysis of spatial patterns on large datasets. Here follow the main motivations for pursuing this type of inquiry.

First, it is worth noting that the studies on syntactic genotypes conducted by Hillier and colleagues were based on the availability of labeled floor plan datasets. When discussing the existence of a configuration genotypes in vernacular buildings, the UCL researchers base their consideration on the statistical analysis of the available samples. If patterns of spatial use are expression of social and cultural form of human activity organization, then our considerations must be based on statistical observations. And here is where the first advantage of a Machine Learning workflow becomes evident: by leveraging the statistical analysis of large datasets, as opposed to small datasets, Machine Learning algorithms can be efficiently trained to identify patterns of correlation in contexts with a high number of features and complex relation function. In the era of massive data availability, Space Syntax has not yet achieved its fullest potential for a simple reason: lack of large labeled datasets describing the permeability and visibility properties of spaces. A second point is that the automation of feature extraction from documents representing space - such as architectural plans - has not achieved a complete state, meaning that although software for Space Syntax analysis already exists, it either performs only one specific task, or requires the input to be in a specific format (typically DXF). A complete and integrated software allowing for the identification of spatial properties starting from raster images does not seem to be available within the Space Syntax community. However, given that the amount of available image data on the internet largely surpasses the amount of DWG or DXF data, it would be worth exploring the possibility of a software designed to perform feature extraction on image documents.

Thanks to existing CV and OCR algorithms, the automatic labeling of room functions, combined with the possibility of extracting graphical and visibility properties from a plan, would open the opportunity to build a very large dataset of labeled plans that can be read and interpreted by Machine Learning algorithms. These could outperform humans at detecting patterns in the relationship between space use and spatial properties, and potentially help researchers in defining cultural and historical genotypes - as introduced by Hillier.

Finally, and in a future perspective, a Machine Learning Space Syntax framework could be turned from analytic tool to generative tool and become a step in a larger AI-based design assistant supporting design choices in new projectsComments on possible generative applications are left to the Applications section at the end of the thesis.

The goals sketched here are of course ambitious. In the context of this thesis, I hope to help clarify how a Machine Learning workflow can be conceived and implemented.

3 The quality of privacy in homes

3.1 Architectural privacy

In the previous Chapter, we discussed how Space Syntax researchers leverage configuration analysis and analysis of the visual fields to explore what we would define as the experiential dimension of built space.
It is necessary, at this point, to refine the distinction between what graph analysis and isovists analysis can convey. In other words, we need to elaborate on the relationship between permeability and visibility. The permeability structure of a system is the relationship between the articulations of space as experienced by someone moving across them. It defines where the user can go, in how many ways he or she can get there and how costly it is to get there. On the other hand, visibility belongs to the narrative of what sight makes accessible to a user standing at a specific location. It tells us something about the visual power and possibilities of a room in a plan.
Space Syntax has been used in Design research to study architectural privacy [14]. Privacy, as referred to space, can be defined as the quality of being apart from company, traversal or observation. Irwin Altman and Westin (Altman, 1975, Westin, 1970) claim, from an environmental psychology point of view, that privacy has the major role of reinforcing self-identity by creating personal boundaries.

> *Privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others. Viewed in terms of the relation of the individual to social participation, privacy is the voluntary and temporary withdrawal of a person from the general society through physical or psychological means, either in a state of solitude or small-group intimacy or, when among larger groups, in a condition of anonymity or reserve. The individual's desire for privacy is never absolute, since participation in society is an equally powerful desire. Thus each individual is continually engaged in a personal adjustment process in which he balances the desire for privacy with the desire for disclosure and communication of himself to others.*[8]

This behavioral idea of privacy as a control mechanism reinforcing self-identity has a parallel with spatial privacy, in the sense that space can work as a filtering mechanism identifying intimate spaces as spaces for the self, as opposed to spaces for the community and for the interaction.
Architectural privacy can be in fact defined as the capacity of space to regulate the information which is passed from a specific space unit to the surrounding environment [14]. The filtering property is assigned to boundaries. Walls, doors, furniture and other barriers, but as well buffer spaces and the configuration of space itself serve as privacy builders. As pointed out in [14], boundaries take different forms and roles in different cultures. For example, in the Japanese culture sliding walls can create different levels of inclusion and exclusion depending on the chosen configuration, which itself is modulated on the base of the time of the day [9]. Differently, "Arabs avoid partitions and since there is no physical privacy, they use other means to be alone. The form of the home is such as to hold the family together into a single protective shell" [10]. In the XVII-XVIII century France, the *enfilade* (sequence of spaces connected in linear ordering) was used as a spatial device to increasingly filter the access from the reception spaces of palaces and *hôtels* up to the private cabinets of the owners. The contemporary western architecture tends to use buffer spaces as filters that distribute access to bedrooms, and the boundary function is again assigned not only to physical barriers, but also to the distribution itself.
Having defined architectural privacy, we can now identify two different qualities a space can have based on its level of privacy - intimacy and openness.

3.2 Intimacy gradients

Intimate spaces are spaces with the characteristic of being significantly segregated with respect to the other functions and spaces in a plan. Christopher Alexander defines the concept of intimacy gradient in *A Pattern Language: Towns, Buildings, Construction* [1],

[8] Westin, 1967
[9] Hall, 1969
[10] Ibidem

Pattern 127 - Intimacy Gradient:

Conflict: Unless the spaces in a building are arranged in a sequence which corresponds to their degrees of privateness, the visits made by strangers, friends, guests, clients, family, will always be a little awkward.

Resolution: Lay out the spaces of a building so that they create a sequence which begins with the entrance and the most public parts of the building, then leads into the slightly more private areas, and finally to the most private domains.

The gradient of privacy described by Alexander can be commonly found in Modern and Contemporary Western architecture. An example is reported in Figure 14. The plan in figure represent the renovation of the top floor apartment in a historic Italian building (Whyassociati). [11]. As pointed out by the architects, the areas of this house are organized hierarchically in communal meeting zones and the more intimate, cozy spaces.

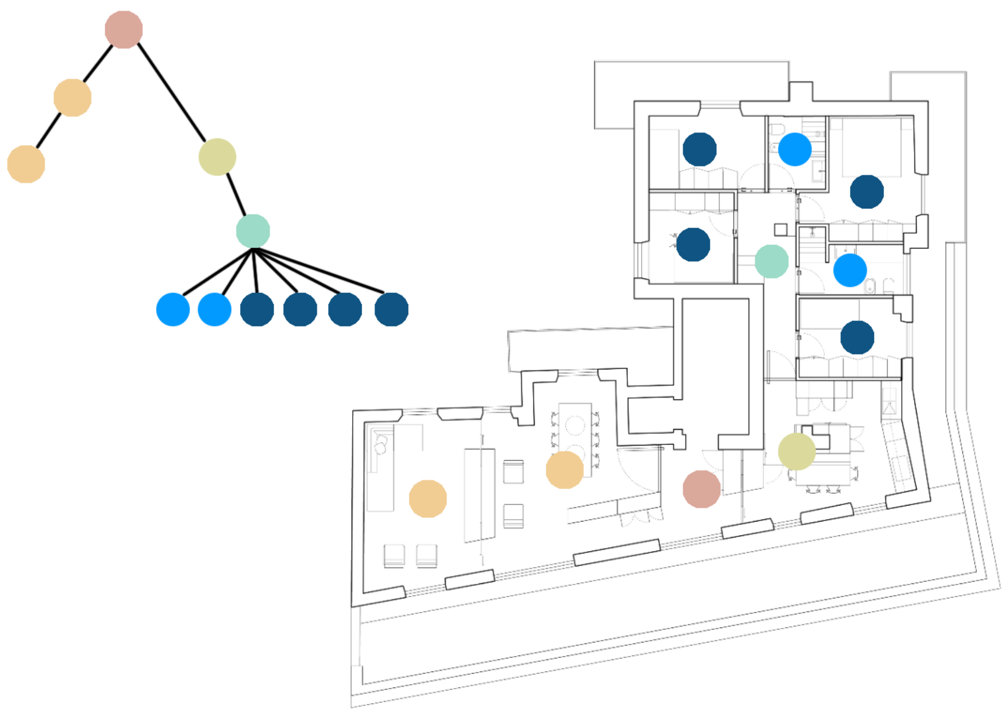


Figure 14: A representation of the intimacy gradient in Multiplicity House by Whyassociati. The warm color represent the less intimate spaces, and the colder colors the more intimate

When considering the most intimate spaces, we notice that they are located at the very periphery of the graph representing the spatial connections. In fact, "peripherality" is a shared property of intimate spaces in different cultural contexts. "Peripherality" can therefore be considered an intimacy-building factor in spatial contexts.
When considering the isovists, intimacy can be intended as the quality of spaces that are away from the visibility axes generated in other rooms. A room with a low number of visual neighbors is visually more intimate than a room with a high number of visual neighbors.
When coming to the qualities of spaces for interaction and openness, the quality-builders spatial characteristics are in opposite relation to the ones just seen for intimacy. These tend to be spaces with a high level of graph centrality, as well as a high number of visual neighbors. Part II and Part III in this thesis will further analyze these patterns.

[11] Whyassociati, Multiplicity House, Seregno, Italy

3.3 Privacy in homes

The dimension of the house offers the perfect ground for studying gradients of privacy. A home is a place for a broad spectrum of activities and states requiring different levels of intimacy. Some of the rooms in a house are spaces for the interaction and common living - the living room, the dining room, the kitchen. These have a low privacy level, and their quality is openness, rather than intimacy. On the opposite side of the spectrum, there are spaces devoted to the intimate sphere of the individual - the bedroom, the bathroom, the dressing room. We also find a range of intermediate-privacy spaces. For example, study rooms, reading rooms, music rooms, small libraries. These are not necessarily exclusive to an individual; however, they are mainly for individual use, or at least they are not designed for communication or interaction, rather for reflective work. We can as well have service rooms, which we would not classify as private nor as open, but somewhere in between these two - for example, laundries, play rooms. Corridors and other distribution spaces, which serve the purpose of connecting other spaces, can be more or less private.

The nuances of intimacy in a house are multiple and can be ambiguous. The French culture, in particular, has a tradition for the study of the spatial hierarchies in homes. As previously mentioned, Jacques-François Blondel authored a systemic and broad discussion around the layout of French mansions [12] and focused on architectural distribution in the second part of his *Cours d'Architecture* (1777). The French aristocratic mansions of the XVIII century were often designed with a specific attention to the hierarchical properties of their layout. For this reasons, they represent interesting case studies when inquiring into architectural privacy. In these buildings, the *salon* was the main reception space, open to the interaction with hosts from outside the family. It was located topologically before the private apartments. An *antichambre* was typically located between the *salon* and the more private *chambre* and served the purpose of hosting visitors who were waiting to be admitted to meet the lord of the house in the *chambre*. Only the closest friends were admitted in the *chambre*. This sequence of filters ended in the *cabinet*, the most intimate space of the house, were visitors were not admitted.

Although French aristocratic mansions are a very specific category of buildings, this way of modulating the privacy of different rooms is more widely diffused and characterizes several building typologies across cultures. Patterns of correlation between Space Syntax properties - such as the graph relations and the visibility - will be identified through a validation experiment in the Part II. While the experiment will be useful to prove that relationships do exist, identifying what those relationships are is object of a more refined analysis, and motivates a Machine Learning framework, as we will discuss in Part III.

[12] Jacques-François Blondel, *De la distribution des maisons de plaisance et de la décoration en général*, 1737-1738

4 Why Machine Learning?

4.1 Statistical approaches to spatial pattern discovering

In Bill Hillier's research, one core idea is that it is possible to detect the presence of cultural ideas in the physical forms of architectural space. By analyzing, for instance, the justified plan graphs of traditional French houses, Hillier observes that despite the apparent geometrical differences in the plans, there are in fact strong similarities in the graph configurations [6]. Specifically, it emerged that the *salle commune*, the main living space, was not only a direct link to the exterior, but it had in all cases a very high level of integration with respect to the whole plan configuration, meaning that the number of spaces to be traversed to get from the *salle commune* to all other spaces in the plan is minimal with respect to other rooms.

> *In the three French houses, for example, we find that there is a certain order of integration among the spaces where different functions are carried out, always with the salle commune as the most integrated, as can be seen in the j-graphs beside each plan. If all the functions of the three houses are set out in order of the integration values of the spaces in which they occur, beginning with the most integrated space, we can read this, from left to right, as: the salle commune is more integrated (i.e. has less depth to all other spaces) than the corridor, which is more integrated than the exterior, and so on. To the extent that there are commonalities in the sequence of inequalities, then we can say that there is a common pattern to the way in which different functions are spatialised in the house. We call such common patterns 'inequality genotypes', because they refer not to the surface appearances of forms but to deep structures underlying spatial configurations and their relation to living patterns.*[6]

Besides this small example, UCL researchers conducted statistical studies over larger datasets of plans, finding patterns of similarity among French farmhouses [7] as well as among family houses designed by architects in post-war London [8]. The dataset of Space Syntax features used for the analysis of London houses is reported in Figure 15.

house number	year built	location	integration focus	area, square metres	convex total	exterior spaces	interior spaces	bounded spaces	boundary : convex ratio	integration with exterior	integration interior only	external rings	internal rings	function spaces	transitions	space : transition ratio
1	1960	suburb	trans	163	27	4	23	12	0.522	1.342	1.690	5	1	9	14	0.643
2	1961	city	trans	146	20	6	14	7	0.500	1.434	1.579	1	0	5	9	0.556
3	1964	city	trans	144	26	9	17	6	0.353	1.297	1.391	1	1	6	11	0.545
4	1964	city	trans	183	32	8	24	16	0.667	1.313	1.484	3	1	11	13	0.846
5	1964	city	trans	117	30	9	21	8	0.381	1.379	1.598	4	1	10	11	0.909
6	1966	suburb	space	370	40	9	31	16	0.516	1.292	1.771	1	2	14	17	0.824
7	1976	city	space	350	35	10	25	10	0.400	1.038	1.859	12	1	11	14	0.786
8	1976	suburb	space	220	31	6	26	4	0.154	1.197	1.377	7	9	11	15	0.733
9	1977	city	trans	184	27	4	23	7	0.304	1.842	1.933	0	0	9	14	0.643
10	1983	city	trans	225	42	8	34	8	0.255	1.912	1.952	0	0	18	16	1.125
11	1984	city	space	252	37	9	28	13	0.464	1.449	1.708	5	1	14	14	1.000
12	1985	city	trans	70	17	7	10	4	0.400	1.294	1.491	1	1	6	4	1.500
13	1986	city	trans	150	25	6	19	10	0.526	1.490	1.412	0	1	10	9	1.100
14	1987	city	trans	225	35	4	31	14	0.452	1.667	1.662	0	2	16	15	1.067
15	1989	city	trans	180	29	3	26	8	0.308	0.815	1.056	7	1	11	15	0.733
16	1989	suburb	space	448	40	8	32	17	0.531	1.280	1.287	2	4	9	23	0.391
17	1989	city	trans	156	34	5	29	12	0.414	1.646	1.597	2	1	14	15	0.933
18	1989	city	trans	369	30	3	27	9	0.333	1.880	1.771	0	1	9	18	0.500

Figure 15: Dataset in Bill Hillier and Julienne Hanson's London houses study [8]

In this study, several measures are taken into consideration in order to research culture-imprinted patterns of London post-war houses. Metric area and number of convex spaces are the first variable to be included. A first, basic statistical consideration is that the metric area is positively correlated with the number of convex spaces within a house [13]. The two UCL researchers classify convex spaces in use-spaces, which are rooms that are devoted to everyday activity, static spaces designed mainly for static occupation and transition/distribution spaces. A strong positive correlation was found that relates the metric area and the number of transition spaces [14]; this means that the larger the house is the more likely it is that architects chose to give increased emphasis to buffer spaces. Hillier and Hanson work out several statistical considerations and conclude that the large majority of the samples in the dataset had a tree-like structure, despite secondary ring structures. Differently from the experiment done on vernacular French farm-houses, a rank-order of the integration of the different function in the house is harder to detect. This is due to the fact that architect's houses do not share a particular genotype.

[13] The correlation factor as identified by Hillier and Hanson is 0.67 [8]

[14] Correlation 0.83 [8]

4.2 From small to large datasets

Studies such as the ones we just analyzed were based on relatively small datasets, and leveraging statistics to uncover underlying relations between a relatively small set of variables. Automation of document understanding as enabled by advances in computer vision and image processing now allows to segment and interpret images in their semantic parts. Part II in this thesis provided a proof-of-concept of the feasibility of extracting the graph representation and properties of a plan starting from raw images. Similarly, by detecting interest points and boundaries on the segmented plan and by using flood-fill techniques we could obtain the isovists of each space. We can therefore realistically imagine a program that takes in the image of an architectural plan and outputs a matrix of the numerical properties for each space. While the full implementation of such a program is left as a future extension of this work, we now want to highlight the fact that such a system would enable us to process large amounts of publicly accessible floor plan images from the internet and automatically label their spaces. In short, this would open up the possibility of building very large datasets of spaces labeled according to their physical properties as well as their assigned function [15].

There are several advantages in shifting Space Syntax analysis from a small-dataset, few-variables setting to a large-dataset, many-variables setting. The first and most evident is that it would provide a larger statistical base to infer patterns. Not only this, but when dealing with the problem of cultural genotypes, having the possibility of working with large datasets of floor plans of houses from a specific socio-cultural context would make it possible to compare spatial patterns across cultures at a higher level of precision. Moreover, there is the problem of bias. For example, the dataset of French farmhouses plans used by Hillier and analyzed above was pulled out of the same source - a museum. It can therefore be biased towards specific cases of farmhouse. By working with very large datasets pulled out from the larger base of the Internet, we would limit the risk of bias in our data.

4.3 Machine Learning and architectural design

> *Art has always existed in a complex, symbiotic and continually evolving relationship with the technological capabilities of a culture. Those capabilities constrain the art that is produced, and inform the way art is perceived and understood by its audience.*[16]

In this section, we provide an overview of the state of the art of Machine Learning techniques as applied in the context of architectural design problems - clustering of plans on the base of types, plan generation, style classification and others. In particular, we present a brief literature review on the very few pioneering research projects applying Neural Networks non-trivially to core problems in architecture.

The idea of automating the process of design is not new. In fact computer-aided design was born decades ago. The oldest CAD system is the Sketchpad developed by Ivan Sutherland at MIT in 1963. Sutherland is considered the father of computer graphics and won the Turing prize in 1988 thanks to his contribution in this area. During the 60s and 70s, the intellectual discourse around human-machine collaboration in architectural design was very fertile and led to visionary research such as the work by Steve Coons and Nicholas Negroponte. "The Architecture Machine" by Nicholas Negroponte (1969) outlines the characteristics, potentials and limits of a human-machine collaboration to solve design problems. In Negroponte's vision, human intelligence and machine intelligence are combined in such a way that artificial intelligence is leveraged as a "design assistant", but also goes beyond this level and is imagined as a design intelligent agent itself.

The advances in computational capabilities over the past decades have closed many of the gaps between design intelligence and reality, and has made it possible today to practically experiment with ideas that date back to the '60s-'70s intellectual debate.

The explosion of research in Machine Learning has inspired designers to expand the intersection between AI and design tools. There are several examples of design and research projects that demonstrate the potential applications of Machine Learning systems in different stages of the design process.

One sample project that exemplifies a Machine Learning application in Design is Paul Harrison's

[15] To detect room function, two ways are possible: a OCR function reading the character-label of rooms, where available, or a neural network trained to recognize the furniture symbols on the plan, and consequently assigning the respective label.

[16] Blaise Aguera y Arcas, Artists and Machine Intelligence, 2016

“What Bricks Want: Machine Learning and Iterative Ruin” [17]. In this paper, the author leverages Machine Learning to generate unique structural arrangements through rigid-body simulations of building collapses. Another example is the deployment of Machine Learning techniques to understand space usage at WeWork. The WeWork Product Research team used Machine Learning to assist in predicting how often meeting rooms would be booked. Other examples include Autodesk Research work in building automation and floor plan generation.

4.4 CNNs for isovist analysis

One paper of special interest in our case deploys Deep Neural Network for the classification of spaces based on their isovists.
In their very recent paper "Machines’ Perception of Space", Wenzhe Peng, Fan Zhang and Takehiko Nagakura at the MIT Design Computing group explore a similar inquiry and present a threefold result. In order to set up the conditions for an algorithm to quantify spatial experiences, they present a system for representing 3D isovists in 2D. Staring from a catalogue of basic 3D spatial situations, they create a corresponding catalogue of 2D isovist fingerprints and let a CNN model learn to classify each space base of its fingerprint. In test phase, they then employ the system to recognize and classify the category of each space based on its 3D isovists.
By training the CNN with a 5000-element dataset of artificially generated samples, the model achieves 99% accuracy on the validation set.
The model is tested on two real world case studies, the Barcelona Pavilion, Exhibition House Berlin 1931, and Paviljoen van Aldo van Eyck. The result is critically analyzed in its potential but also on its limits. Potential extensions and improvement of this work include the use of a real dataset during training.
This work is of particular relevance in the context of this thesis because it appears to be the first attempt to use Deep Neural Networks to classify rooms and spaces based on their isovists.

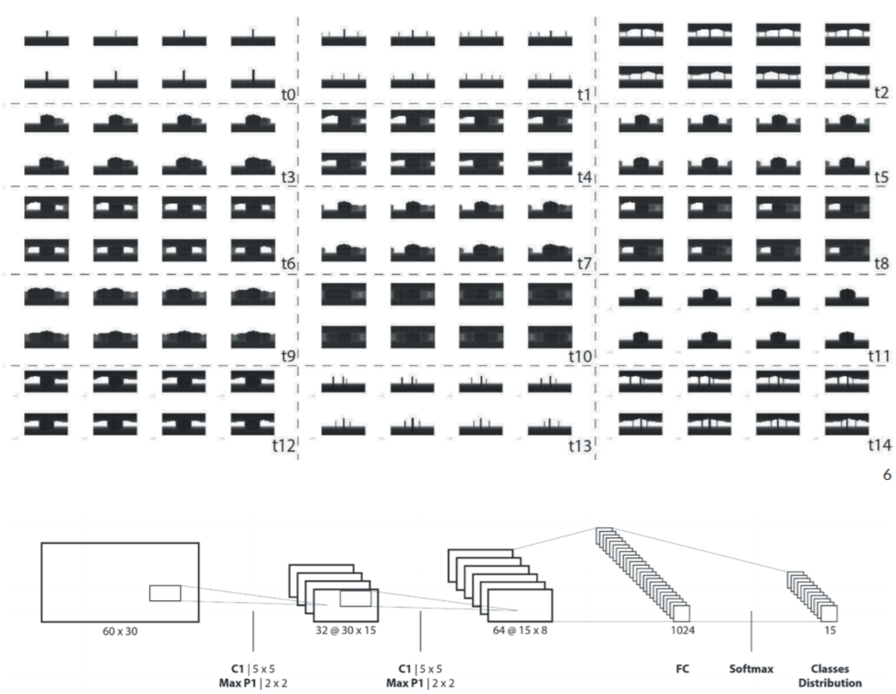
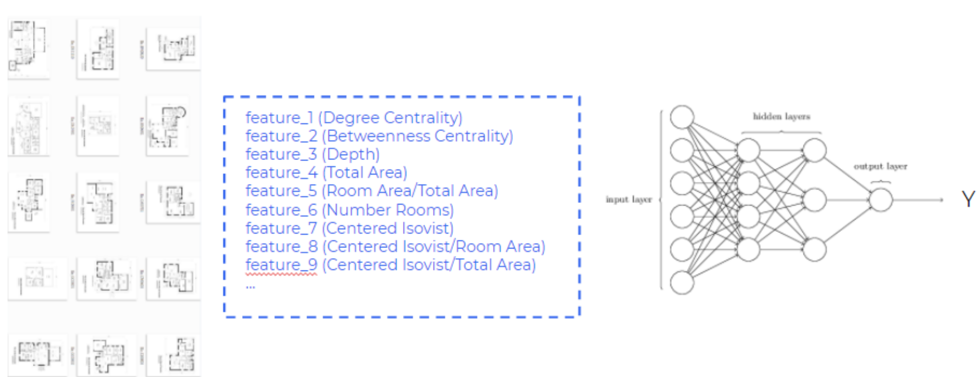


Figure 16: A part of the space samples of the 15 different Seed-Spaces in Peng’s paper, and CNN architecture. From Peng et. al (2017)

[17] Presented at ACADIA 2016

4.5 Why Machine Learning in spatial analysis?

We have overviewed Space Syntax and its main methods and tools. Why would Machine Learning represent a useful method in Space Syntax?
Machine Learning algorithms are powerful in recognizing patterns in contexts where many variables are involved that could influence some variable of interest. In particular, neural networks have become a pervasive and effective tool for pattern recognition.
We have also just considered how a large dataset of architectural plans labeled in their spatial properties and functions is realistically achievable. When large labeled datasets are available, supervised learning with neural networks becomes a feasible, very effective method for performing classification or regression analysis with the purpose of pattern discovery.
To better highlight the advantages of a Machine Learning framework is Space Syntax, we will go back to Bill Hillier's *Ideas are in things*. This study, as seen, applied Space Syntax techniques to statistically investigate traditional French farmhouses. Specifically, graph depth and integration were used to identify spatial genotypes, which can be defined as specific types of ordering of access among spaces with different function. The main limit in Hillier's study is in the dimension of his dataset. Being the dataset small, we can ask ourselves how general the found results can be claimed to be. While in the physical sciences mathematics has proved to be the most elegant and effective way to explain phenomena, in the social sciences, but in general in any problem that involves human behavior at any level, we cannot define a simple equation describing a complex phenomenon. As Halevy, Norvig and Pereira claim [15], when it comes to social problems, we can't start to understand what patterns and rules are until we have a lot of data to look at - and there is where the "unreasonable effectiveness of data" emerges. And with big data, Deep Neural Networks outperform other models at classification and regression tasks in several problems. Moreover, the deeper Deep Learning models are in their structure, the more capable they are of modeling complex functions of the input features. Therefore, if we suspect that our variable of interest y is a complex function of a set of features X, and we have sufficient number of examples to learn on, than Deep Neural Networks are the best that we can do to gain insight and knowledge.
This thesis looks at rooms in a floor plan (specifically house plans) under the lenses of their privacy. Starting from the consideration that different levels of privacy characterize spaces with different functions, and starting from the reasonable (and validated in Part II) assumption that quantitative Space Syntax measures do contribute in explaining privacy, we want to show how a Machine Learning model learning to correctly classify the level of privacy of a room in a floor plan based on a larger vector of measures than the one adopted in the Space Syntax studies cited above.



Part II

Second part

In this Part, we will overview a validation experiment to provide a ground-level demonstration of how we can leverage the spatial information encoded in architectural floor plans to extract measures for the space quality of privacy/openness in rooms in a plan.
Measuring the intimacy/openness quality of a space is *per se* a complex task. Because the appreciation of quality of architectural spaces is experiential, it is hard to express it in numbers. In addition to this, architectural space is a three-dimensional object, and is composed of materials, voids, light, paths, levels. This thesis focuses on the spatial information carried by architectural plans, to test how much information about the quality of spaces can be extracted from this type of architectural document.
The goal of this Part is double. On one side, we will present the results from a validation experiment. On the other, the skeleton of an image processing program will be built to perform the extraction of the graph structure out of plan images.

5 Preliminary experiment

In this section, we consider the results from a preliminary expert choice experiment in which a group of scholars with Design backgrounds was asked to express a preference for the assignment of two different functions to a space in an unlabeled floor plan. This experiment was run with the intention to validate the existence of patterns of choice based on intrinsic qualities of the spatial configuration (integration, segregation, openness,...).
In this preliminary experiment, we tested what qualities, implicitly, experts in the field of architecture look for from spaces with different functions. Specifically, two floor plans of significant architectures were chosen, one historic and one contemporary. Labels indicating the function of each room were digitally removed. A group of nine graduate students from the Master of Science in Computational Design at Carnegie Mellon University and their instructor prof. Daniel Cardoso Llach were asked to consider the set of qualities described above - visibility, connectivity, peripherality. They were then asked to analyze both floor plans, and highlight which room they would pick as their own bedroom and which room they would pick as reception room, if that were their own apartment.

5.1 Hôtel Biron

The first plan analyzed in this expert choice experiment is the Hôtel Biron in Paris. This is a XVIIIth century aristocratic mansion designed with specific architectural design principles and distribution constraints, that will be illustrated later. The participants in the experiment were all designers with technical expertise, however they did not have access to specific information regarding the building. In fact, the experts were asked to work on the unlabeled version on this plan.
The experts were prompted to select the room on the floor plan that they considered more suitable to become their bedroom. The results are reported in Figure 4, listing the number of people who chose the selected room. Unlabeled rooms received no vote.
From these raw results, some considerations emerged that confirmed part of the assumptions. First, regarding the choice of a bedroom, it is interesting to highlight that all experts in the group chose spaces that are located at a terminal node in the graph representation. This is clear from the comparison of Figure 4 and Figure 7. This might be because experts highly value the fact that these spaces have only one access point, and therefore are not passage spaces, which is a quality a user can be reasonably looking for in a bedroom. However, it is also interesting to observe that the majority of the experts opted for the only terminal-node room that also had three windows. This suggests that lighting conditions matter in the choice. Also, the most highly selected bedroom was rectangular, as opposed to rounded[18].
As far as the choice of a reception space is concerned, we again notice that it is polarized, this time favoring the large central room with terrace. This room as several doors connecting to other spaces

[18]While lighting and shape will not be included as features in the analysis that follows, an interesting extension of this work could include these factors has contributing to the perceived privacy of a room

on the same floor. It is therefore a highly connected space. A minority of the group opted for the round room at one end-side of the *enfilade*.

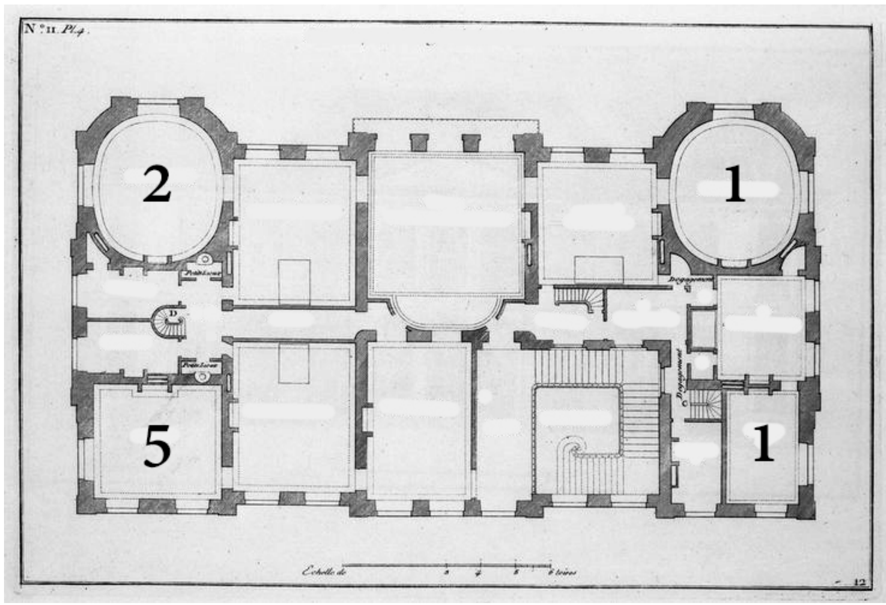


Figure 17: Number of experts who selected the room as their bedroom, if any

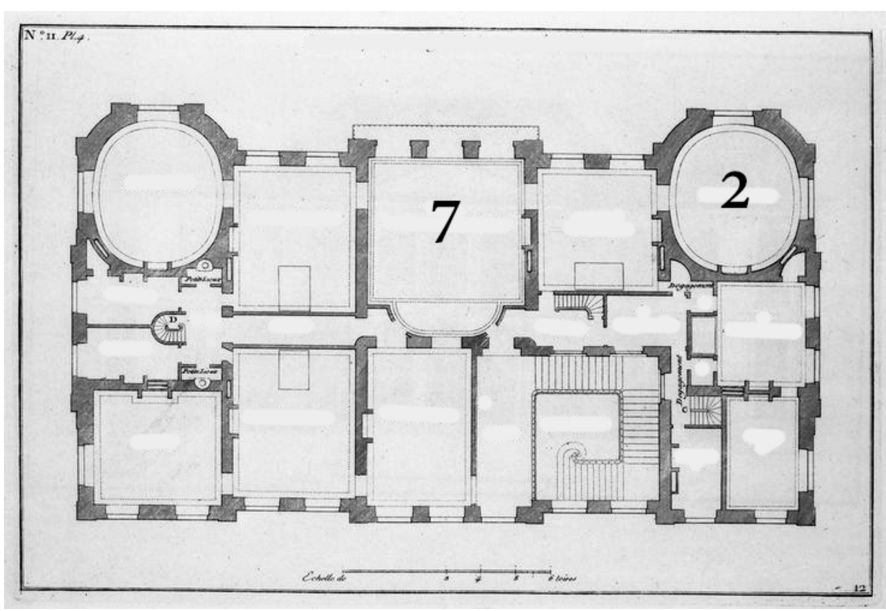


Figure 18: Number of experts who selected the room as reception, if any

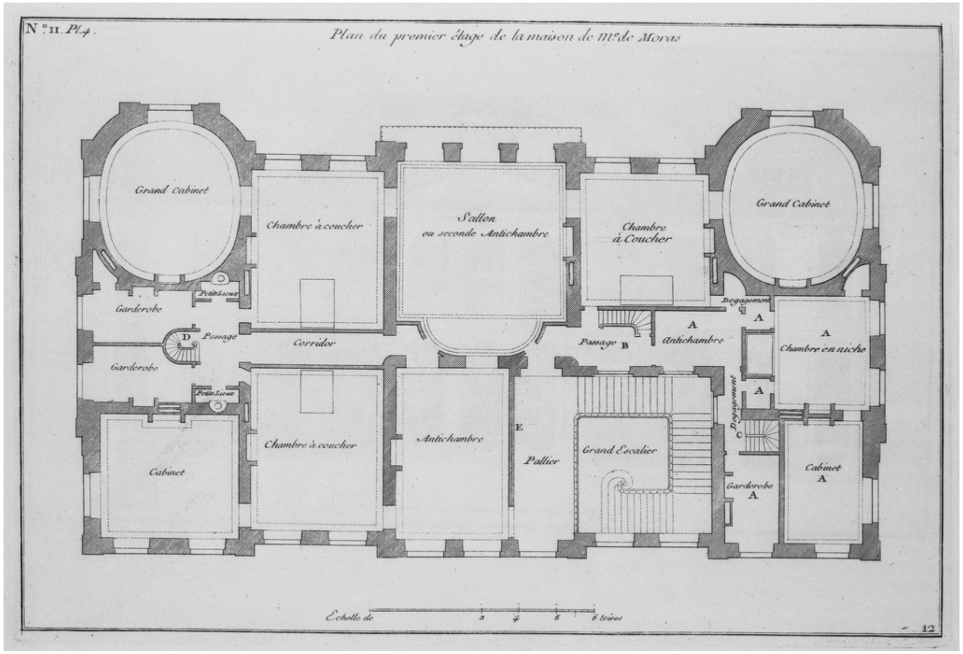


Figure 19: Original functions

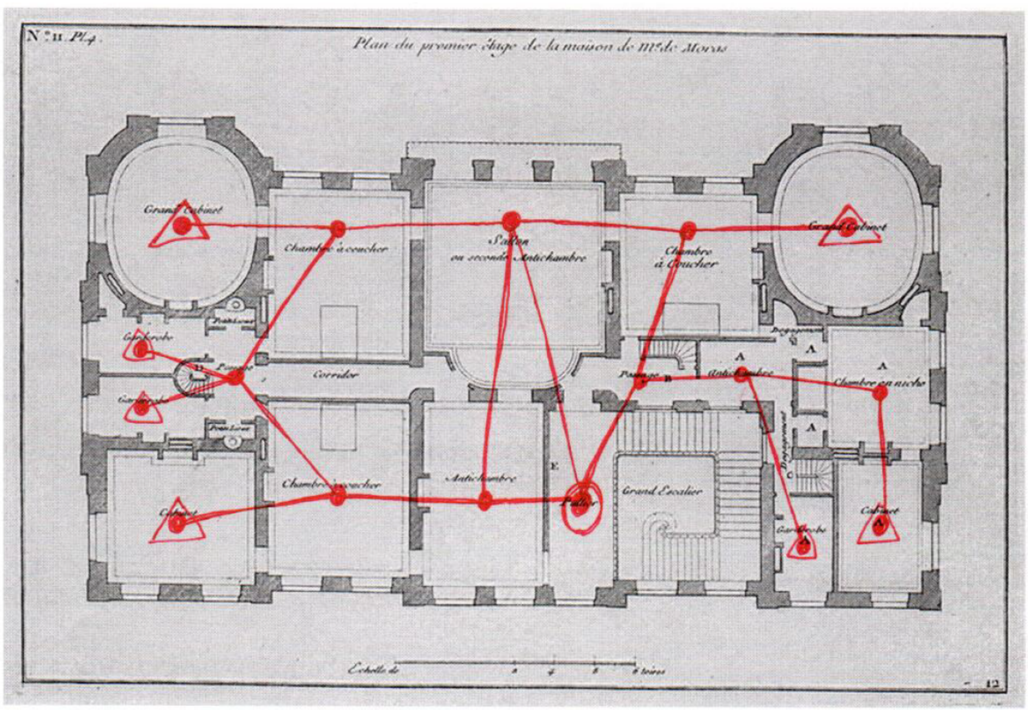


Figure 20: Graph connectivity

5.2 Casa Siza Vieira de Castro

The second example that the experts were asked to consider is the second floor of Casa Vieira de Castro, signed by the architect Alvaro Siza (1994). Again, the group was not informed over the details regarding this architecture, except that it was a second floor of a private house. The experiment followed the same rules as the Hôtel Biron experiment (see related section), except that this time the group was asked to focus on the choice of a private space only. This is because in the original project the spaces on this floor were mainly conceived as private rooms. This was helpful in clarifying the criteria of choice of intimate spaces.
From these raw results regarding the choice of a bedroom, it emerged that the majority of the participants (8/9) selected as bedroom a space with low visibility-to-room-area ratio (see results in Figure 22).

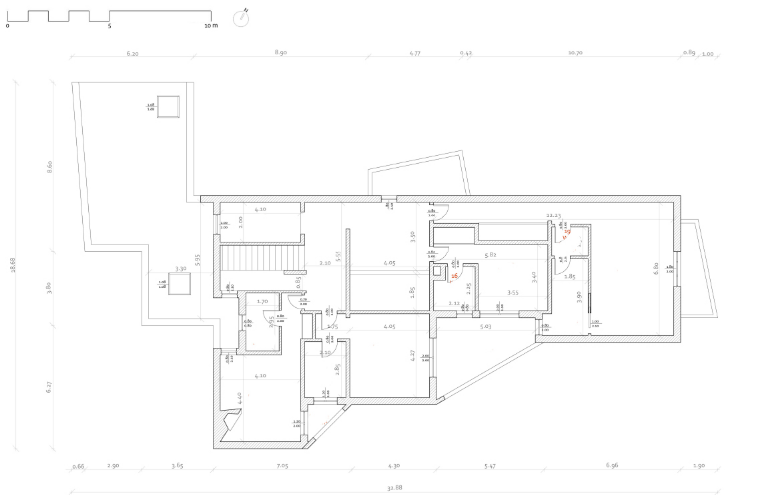


Figure 21: Plan of the second floor of Casa Siza Vieira de Castro

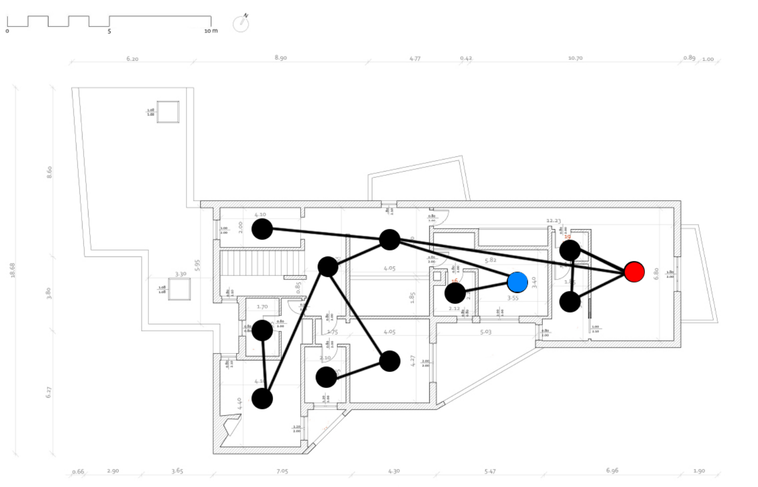


Figure 22: Graph representation of the plan of the second floor of Casa Siza Vieira de Castro. Eight participants over nine chose the room represented by the red node as bedroom, one person chose the blue.

6 Methods

In this section, we explain what methods were adopted to extract measures of spatial qualities from the floor plans in the experiment case studies, using a Network Analysis software to build the graph and traditional isovist methods to compute the visibility measures of interest. Also, a method to merge two distinct measures in one unique measure of spatial quality is described and proved coherent.
The objective of this first experiment was to collect data to explore whether the measures related to isovists and graphical representation of the plan could be jointly interpreted to form a measure indicating the suitability of the single room for private functions (intimacy) rather than "public" functions (openness). Having collected the results from these two experiment, the next steps were to:

- formalize the graphical representation of the floor plan for the two cases
- manually draw the isovists from the center of the room of the relevant rooms
- use the isovists to define a numeric measure to represent the visibility
- define a mathematical formula to express the combined effect of graphical properties and visibility
- verify if the so found unique measure could be verified to be meaningful

6.1 Graph analysis

The analysis of the graphical properties was implemented in Python 3.6 with the use of the network analytic package NetworkX.
The location of the nodes was manually hardcoded, but it is worth noting that it does not actually impact the analysis in any way, as the graph is an abstraction of the topological properties of a set of nodes and their connections.
The measure that better represents an architectural meaning useful for the purpose of this analysis is degree centrality, which represents how many connections one node has relative to the number of other nodes in the graph. Specifically,

$$C_D(v) = \frac{N_{connections}(v)}{|V| - 1}$$

where $|V|$ is the number of vertexes in the graph.
Here follow the graphs and the tables of degree centrality for each of the examples in the preliminary experiment.

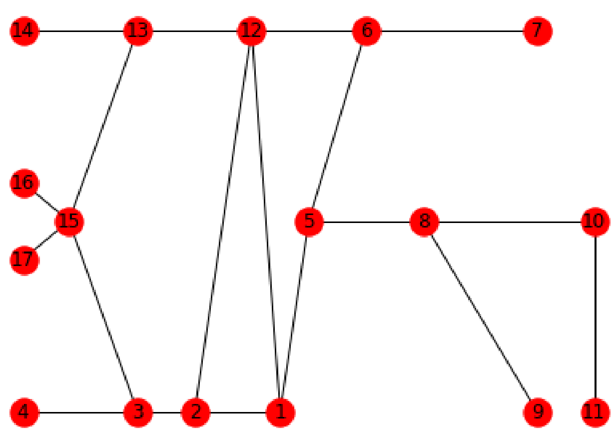


Figure 23: Graph representation of the second floor of the Hôtel Biron

```
Degree centrality:
```

```
{1: 0.1875,
 2: 0.1875,
 3: 0.1875,
 4: 0.0625,
 5: 0.1875,
 6: 0.1875,
 7: 0.0625,
 8: 0.1875,
 9: 0.0625,
 10: 0.125,
 11: 0.0625,
 12: 0.25,
 13: 0.1875,
 14: 0.0625,
 15: 0.25,
 16: 0.0625,
 17: 0.0625}
```

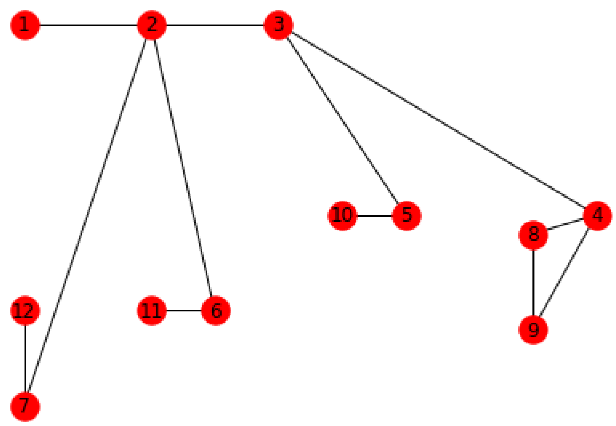


Figure 24: Graph representation of the second floor of Casa Vieira de Castro

```
Degree centrality:
{1: 0.091,
 2: 0.364,
 3: 0.273,
 4: 0.273,
 5: 0.182,
 6: 0.182,
 7: 0.182,
 8: 0.182,
 9: 0.182,
 10: 0.091,
 11: 0.091,
 12: 0.091}
```

6.2 Isovists study in AutoCAD

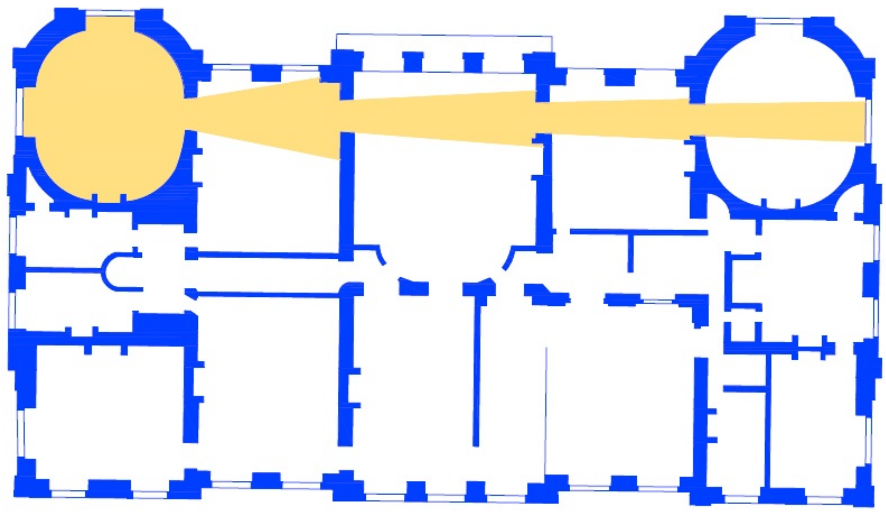


Figure 25: Isovists from center of the room, room 14. Visible area ratio: 0.15. Degree centrality: 0.0625

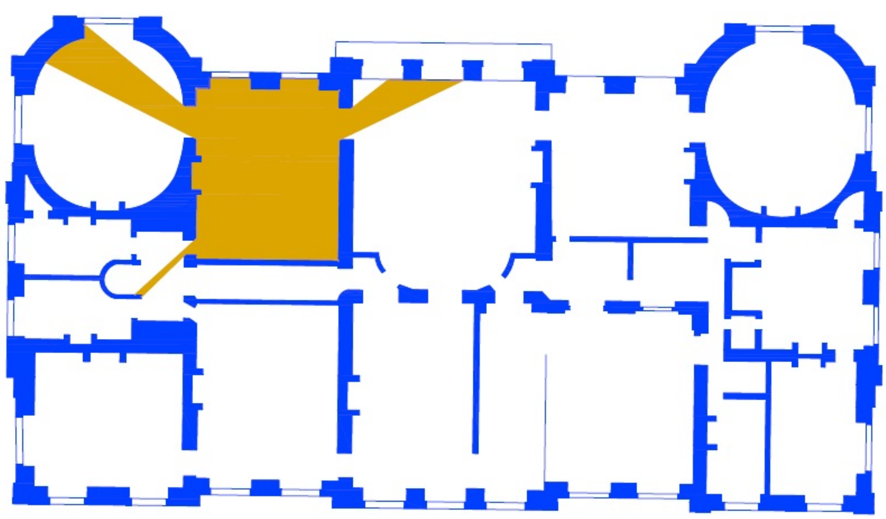


Figure 26: Isovists from center of the room, room 13. Area: $33m^2$. Visible area ratio: 0.1. Degree centrality: 0.1875

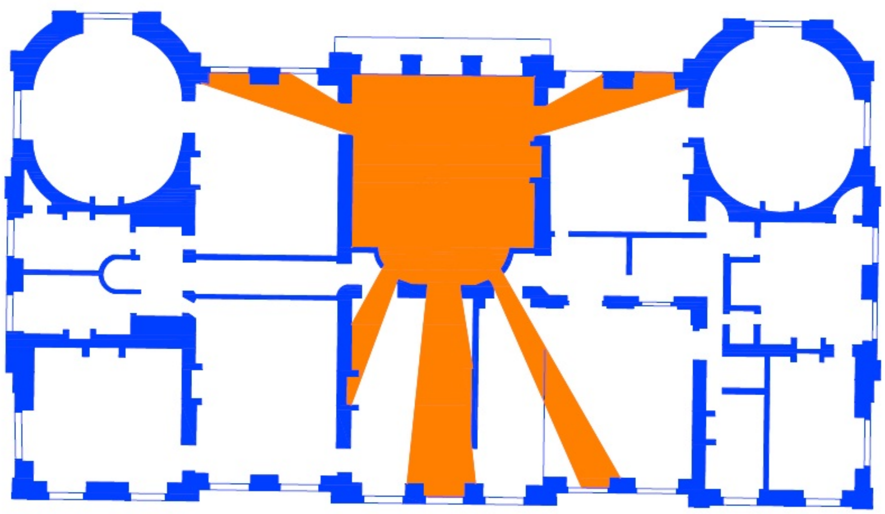


Figure 27: Isovists from center of the room, room 12. Area: $61m^2$. Visible area ratio: 0.19. Degree centrality: 0.25. Hypothetical measure of "gravity": 0.19+0.25 = **0.46**

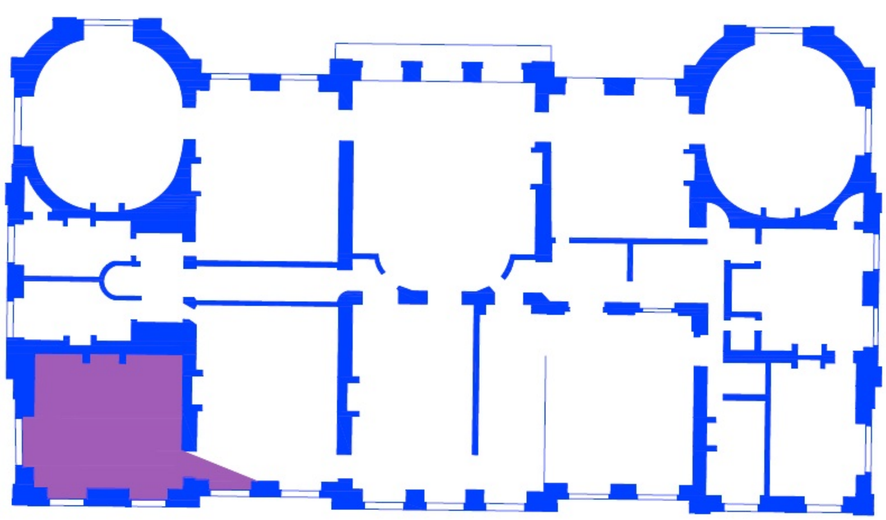


Figure 28: Isovists from center of the room, room 4. Area: $22m^2$. Visible area ratio: 0.067. Degree centrality: 0.0625. Hypothetical measure of "gravity": 0.067+0.0625 = **0.1295**

Floor visibility measures - Hôtel Biron			
Room	Isovists to Floor ratio	Degree centrality	Sum
Room 14	0.15	0.0625	0.2125
Room 13	0.1	0.1875	0.2875
Room 12	0.19	0.25	0.44
Room 4	0.067	0.0625	0.1295

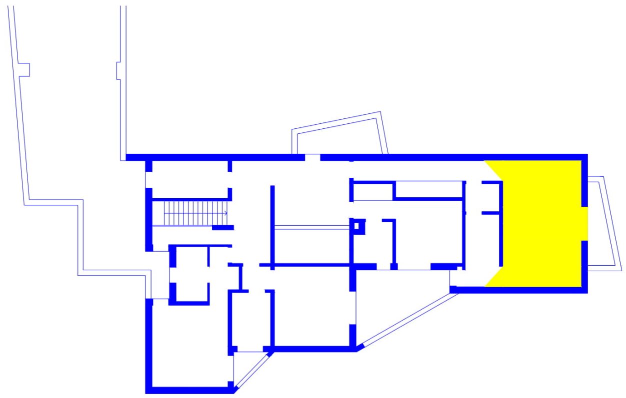


Figure 29: Isovists from center of the room, room 6. Visible area to room area ratio: 0.808. Degree centrality: 0.25. Hypothetical measure of "gravity": 0.808+0.25 = **1.058**

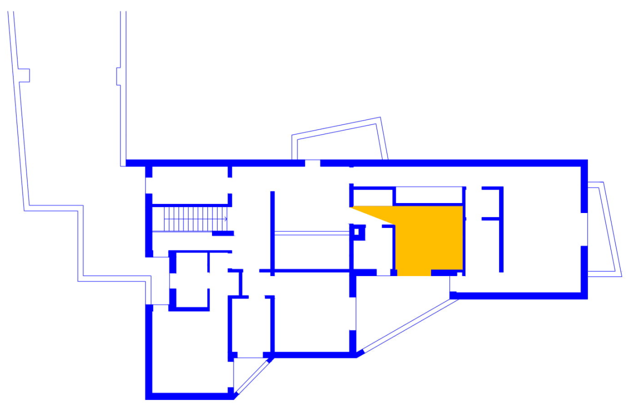


Figure 30: Isovists from center of the room, room 4. Visible area to room area ratio: 0.972. Degree centrality: 0.167. Hypothetical measure of "gravity": 0.808+0.167 = **1.139**

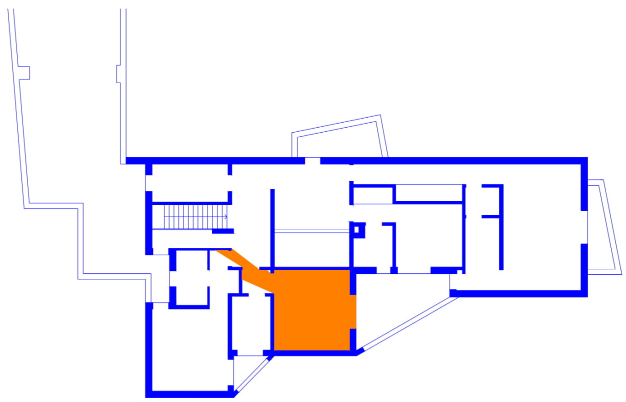


Figure 31: Isovists from center of the room, room 9. Visible area to room area ratio: 1.000. Degree centrality: 0.167. Hypothetical measure of "gravity": 1.000+0.167 = **1.167**

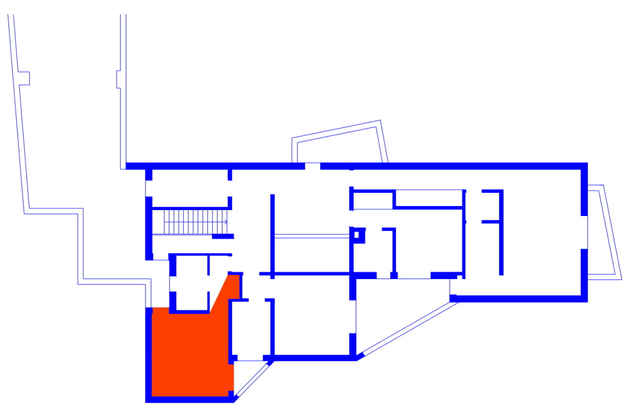


Figure 32: Isovists from center of the room, room 11. Visible area to room area ratio: 0.936. Degree centrality: 0.167. Hypothetical measure of "gravity": 0.936+0.167 = **1.103**

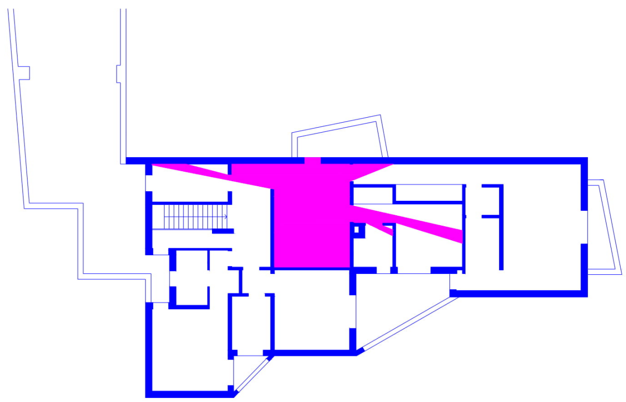


Figure 33: Isovists from center of the room, room 2. Visible area to room area ratio: 2.232. Degree centrality: 0.167. Hypothetical measure of "gravity": 2.232+0.167 = **2.399**

Floor visibility measures - Casa Vieira de Castro			
Room	Isovists to Room Area ratio	Degree Centrality	Sum
Room 6	0.808	0.25	1.058
Room 4	0.972	0.167	1.139
Room 9	1.000	0.167	1.167
Room 11	0.936	0.167	1.103
Room 2	2.232	0.167	2.399

6.3 Considerations

From the results of the preliminary experiment, it emerged that the spaces selected as intimate private spaces by the participants in the first example scored low in degree centrality and in visibility-to-floor ratio. At first glance, the private spaces selected in the second example do not follow this trend. We observe that Room 6 has in fact a high degree centrality score. However, we notice that the reason for that is a loop in the graph connecting the bedroom with its own closet and bathroom. If we consider the loop as a whole (merging rooms 6, 7, and 8), the degree centrality for room 6 becomes 0.167 - lower, and the same as all the other bedrooms on this floor. This leads to a sum score of 0.335. We also observe that the space with higher sum score is also the only space on the floor plan that is not private (room 2).

The sum score is more significant in the historic example of the Hôtel Biron in Paris. The distribution machine of this building was engineered precisely to progressively filter visual and physical intrusion up to the most intimate spaces of the house. While in this historic example the *enfilade* dominates, in the modern house designed by Alvaro Siza the intimate spaces are organized as sub-branches of a star-shape distribution, where the intimate spaces are terminal nodes (or groups) that are not intended for visual or physical traversal.

In this example, visibility seems again to be a separating factor between intimate spaces and open spaces, as the isovist to room floor area ratio of the only one non-intimate space is also the highest value by large (2.232). However, the degree centrality for the most highly chosen intimate space is

higher than expected (0.25). This is because that room is linked in a cycle with two satellite rooms (a bathroom and dressing), which belong to the same system. If we considered this loop system as a whole, the degree centrality of this room would go down to 0.167, the same as all the other bedrooms on the floor plan. Even without retouching the degree centrality, we notice that the sum measure is the lowest for the most highly selected intimate space and the highest for the only non-intimate space on the floor plan.
This considerations can be summarized as follow: spaces perceived as intimate tend to have lower values in degree centrality and isovist-to-floor-area/isovist-to-room-area ratio, while spaces perceived as open tend to have higher values in both variables. As a result, the sum of the visibility measure and the graph measure was in both examples a good proxy for the level of intimacy. This fact validates the idea that there exist discoverable patterns between Space Syntax characteristics and the quality of perceived intimacy of a room as seen on a floor plan. Of course, a longer list of variables needs to be taken into account for a more refined analysis, as well as a larger set of data. This justifies the inquiry into a more advanced analytic tool for the classification of the intimacy quality of architectural spaces. Before going deeper in sketching a Machine Learning framework for spatial analysis, we will present the requirements for a software extracting the necessary spatial features automatically.

7 Extracting graphs and their properties

The technical goal of this part is to build a demonstrative program in Matlab to automatically perform the extraction, representation and analysis of graph representation from floor plan images. The single phases of the graph extraction, starting from the raw image of a floor plan of a building, are described in detail. The algorithmic strategies used to overcome the main obstacles are also detailed. Graphical documentation of the technical steps is provided, as well as the code script.

7.1 The input

The input of this program will be raster floor plan images. While working on vectorial plans would be easier, there is higher public availability of raster images that CAD drawings of floor plans. In the perspective of using this program in the context of the creation of a large dataset, of labeled plans, the program itself must have the potential to run spatial analyis on image plans.
In the context of this thesis, the main step in this phase was to produce a Matlab program that could first of all run all the desired tasks on a sample image. This step was necessary to assess the major challenges and obstacles in building a generalized model.

7.2 First challenge: image pre-processing

The first task is the pre-processing of the floor plan image. The reason why this is necessary is that the only semantic elements that we need to extract from the image are walls, the door voids, the window voids and background. There are other semantic elements, such as text labels, symbols to represent architectural elements and furniture, and other drawing symbols that we need to rule out in the first place.
An OCR system is therefore used to remove text labels.
In order to control for background noise, the image is then processed with a morphological blur filter and finally binarized to a black-and-white image.

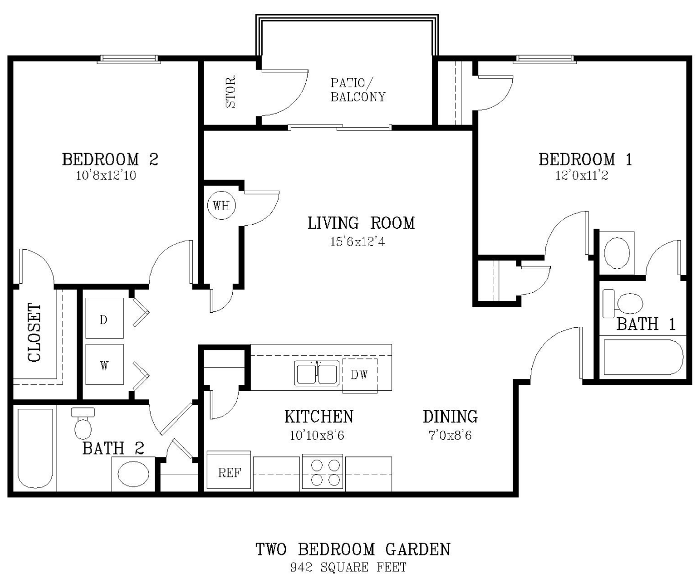


Figure 34: Sample floor plan image in JPG format

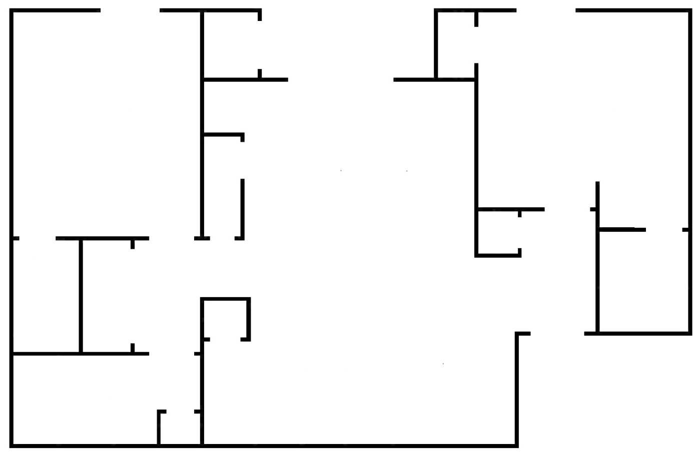


Figure 35: Pre-processed binary image,

7.3 Second challenge: detecting doors

Following the pre-processing phase, we are now left with the extracted wall segments (1 valued pixels) over background (0 valued pixels). Note that this is still a raster image, therefore the interest points such as corner points defining the doors are not stored in any data-structure yet. The challenge now is to detect the location of the doors. This task is achieved through the following procedure.
I first of all want to find the corner points using the Harris corner detection algorithm. This leaves us with a large number of corners detected on the plan, many of which do not define a door. We want to focus on the Harris corner that define the location of doors. The Harris corner detection returns corner points as in Figure 36.
Each door is framed by two edges, each of which has two very close Harris corners. My objective is to fill the door space with a full color, so to create a clear division between different room, a step that is necessary for all the subsequent tasks. In order to do so, an algorithm first isolates the point pairs that belong to a door, then finds the midpoint between each pair of close points defining a door edge (see Figure 37). The first objective is achieved by imposing a pixel condition on a mask centered at the door edge, and checking for the percentage of 0-pixels over 1-pixels.
Now that we recovered the edge points of each door, we would like to proceed to connecting these to form lines. In order to select the correct door point pairs on the drawing, we can leverage the fact that door edges usually share either the same x-coordinate or the same y-coordinate. This is of course a simplification, as there are exceptions where doors are neither horizontal nor vertical with respect to the drawing. To overcome this, another possibility would be to set conditions for the dimension on the door on the plan an find the pairs that satisfy those conditions. However, in the context of this demonstrative program, we assume that all doors will be defined by two edge points that either share the same x-coordinate or y-coordinate.
Once selected all the pairs that share one coordinate, these are connected [19].
We have now closed the doors, which makes it possible to subdivide the image in its closed regions. To do so, we use the connected components and region properties function in Matlab, obtaining the room segmentation as in Figure 39. The background is considered as room 0.

[19] Imposing extra condition to avoid multiple connection of doors on the same coordinate

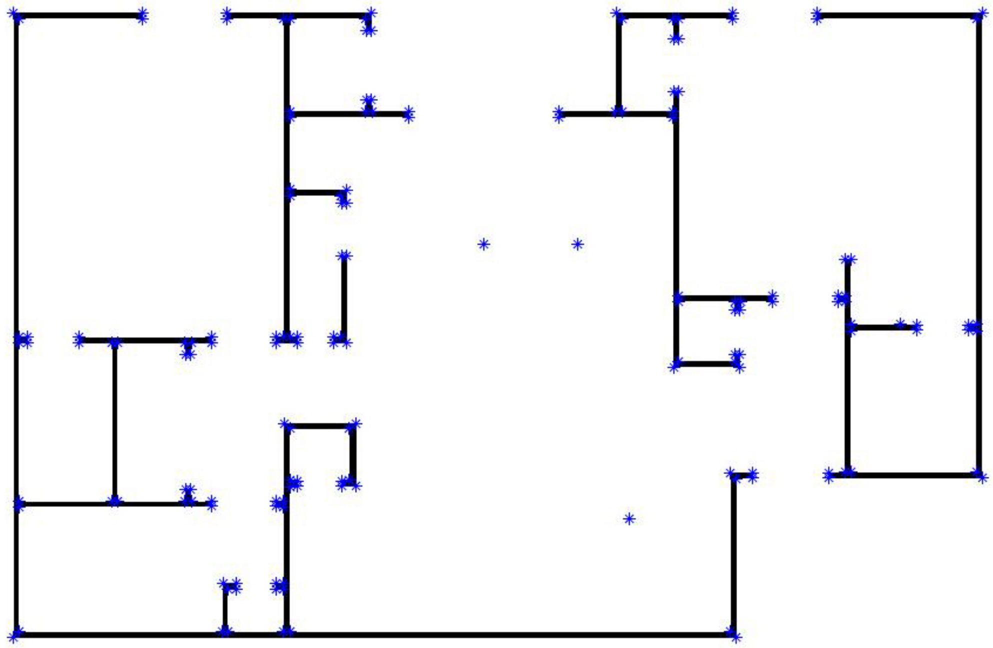


Figure 36: Harris corners

(a) Harris corners

(b) Mid-points between Harris corner pairs on each door edge

Figure 37: Reconstructing doors

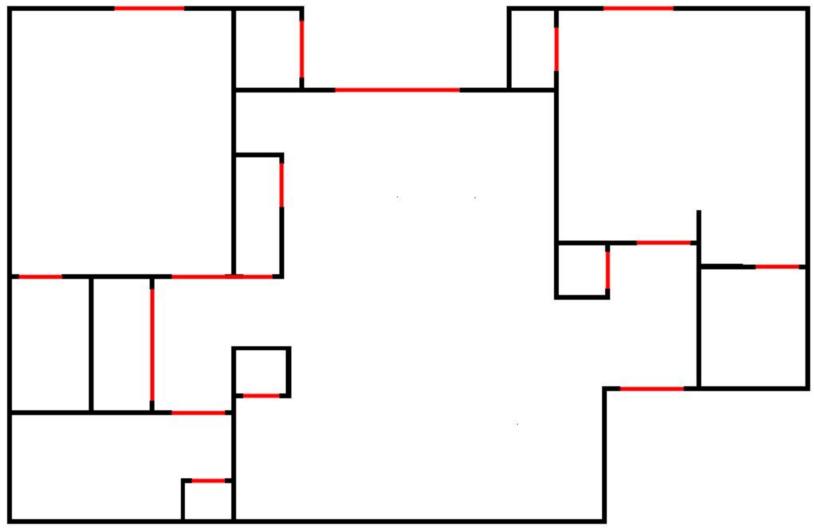


Figure 38: Closed doors

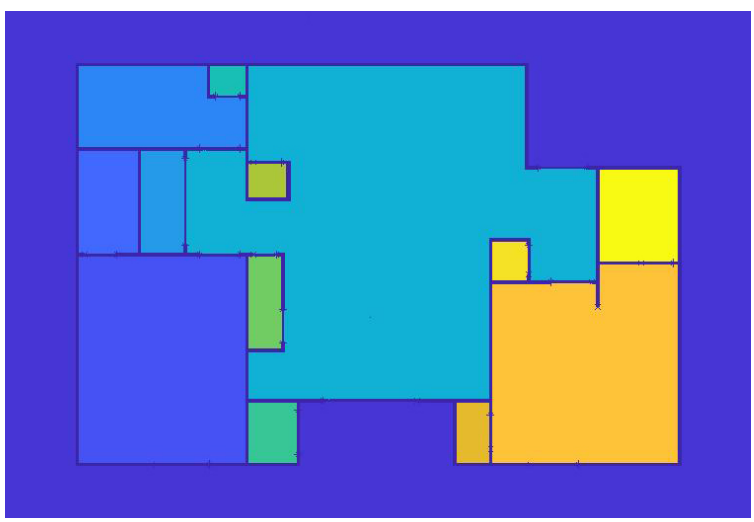


Figure 39: Room segmentation (the background is considered as room 0)

To summarize so far, in order to obtain a room segmentation from a plan image, we can adopt the following strategy:

- Finding corner points using the Harris corner detection algorithm. This leaves us with a large number of corners detected on the plan, many of which do not define a door. We want to focus on the Harris corners that define the location of doors.
- Selecting the point pairs that belong to a door edge using pixel conditions.
- Finding the midpoint for each of these pairs and close the doorgap with a segment

- Using connected components and region properties to store the rooms (and their properties) in a cell structure

7.4 Third challenge: connect the spaces

Once the rooms and doors are stored in their respective data-structures, the challenge was to find, describe and store the correct graph representing the configuration of the plan.
After several other attempted strategies, this was achieved by adopting a shortcut:

- As previously, doors were assumed to be either vertical or horizontal, which is a reasonable assumption on the majority of the plans (but remains a limitation that opens opportunity for further work)

- For each door, its centroid was computed and temporarily stored

- For each centroid, the coordinates of the two extremes of a short diagonal segment centered at the centroid were stored (see Figure 41

- If the two sets of coordinates were found in the pixel list of two different rooms, none of which was the background, then a connection between the centroid of the two rooms was created and stored

In such a way, a data-structure of a graph object was created which contained the rooms (nodes) and connections (doors). Since the graph itself is built as a Matlab object, all sort of graphical measures can now be extracted using built-in algorithms.

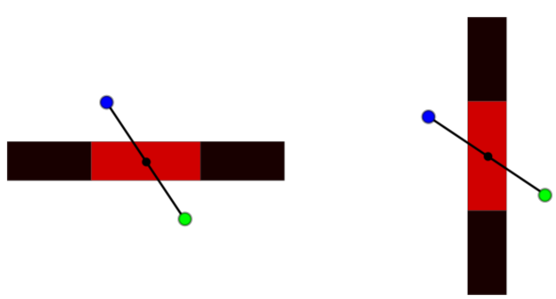


Figure 40: Trick to identify room connections through pixel conditions at the extreme of a segment centered at a door. Doors are in red, different colors at the extremes of the segment signal that the two points belong to different rooms.

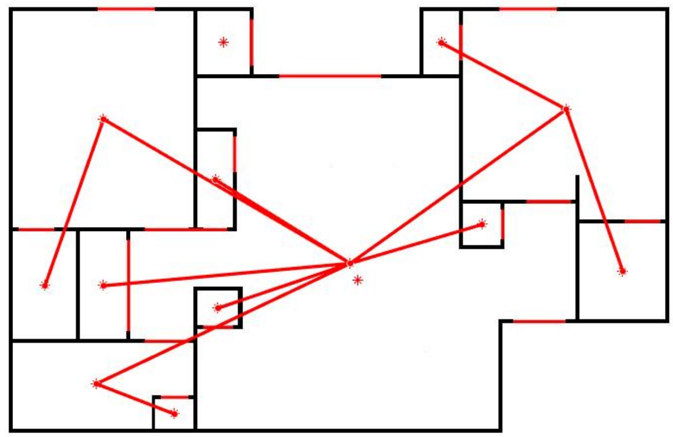


Figure 41: Representation of the final graph

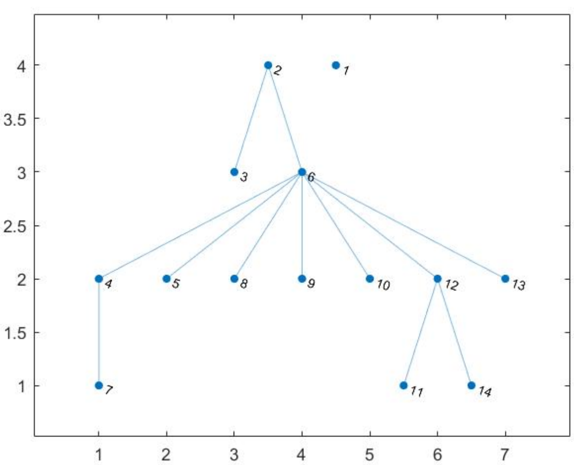


Figure 42: Final graph as justified with respect to room 2

Part III

Third part

8 A Machine Learning workflow for Space Syntax

In this section, we sketch the structure and functioning of a Neural Network system for the regression analysis of spatial characteristics as predictors of suitable functions for different rooms in a floor plan. In Appendix, I present an essential introduction to Machine Learning and Neural Networks, overviewing the necessary pre-requisite technicalities.
The input of the Machine Learning system will be a vector of spatial features as extracted by an image processing software as presented in Part II. The architecture of the Neural Network will include multiple layers of neurons and non-linearities.

8.1 Target variable: level of privacy

In sketching this Machine Learning framework, we can imagine several different target variables a researcher might want to address. Among these, we are interested in the problem of detecting patterns that link spatial features and privacy. The concept of privacy has been discussed in Part I, and we have seen how a gradient of intimacy results from mixing spatial variables (such as degree centrality or centered isovists) in different ways. The level of privacy of the spaces in a house can be analyzed either by categorizing the different functions and top-down assigning a value to their level of privacy/openness, or rather, and more interestingly, by taking a close look at the spatial features and using them to infer the level of privacy. In *Community and Privacy* (1963), Chermayeff and Alexander describe the social agenda for a well-designed house as follows:

> Irrespective of their function and size, the diverse domains of the modern world are multiplying and are susceptible to rapid change and to a variety of conflicts between them. These domains can not only be abstractly explained in terms of function and need, but an be precisely described in terms of physical properties, which can be directly perceived. [16]

8.2 Deep Neural Network model

Given all the background and considerations so far, in order to build a Deep Neural Network classifying rooms in a plan on the base of their level of privacy, we need to synthesize the information of a floor plan into a vector of meaningful spatial features. There is no single answer to what this list should be, however one possible input vector could include:

- Total floor area
  The area of a house floor plan does realistically impact the layout. It is a relevant variable. Moreover, other variables, such as the ones expressing relative visibility, do relate to the total floor area
- Room area
  The area of a room may to some extent explain its function. For this reason, it is important to include it. Moreover, once again some visibility features are defined in relation to the room area.
- Centered isovist
  The area of visible space from the center of the given room in a floor plan. It gives an idea of how much we could see from that room, or conversely from how many points in space the center of the room would be visible.
- Centered isovist to room floor area
  The ratio of centered isovist to room floor area. Provides a slightly different information than the absolute isovist inasmuch it relates the isovist to the room area itself, therefore representing the proportion of visible area and floor area in a room.
- Centered isovist to total floor area
  The ratio of centered isovist to total floor area. It represents the proportion of visible area

from a room and total floor area. It gives an idea of the visibility "power" of a room, the larger the proportion of visible space, the stronger the room is in terms of visibility.

- Number of visual neighbors
  The number of other rooms reached by the centered isovist of a room. In other words, how many other spaces are visible from one space.

- Degree centrality
  Degree centrality measures how many connections a node has, as related to the total number of other nodes in the graph. This measure is useful to quantify the probability of a node for "catching whatever is flowing through the network"[20]. It is a measure of integration of the node in the graph.

- Betweenness centrality
  Betweenness centrality measures the extent to which a vertex lies on the shortest paths between other vertices. High betweenness centrality can be interpreted as the node having influence and control over information passing between others. They are also the ones whose removal from the network will most disrupt communications between other vertices because they lie on the largest number of paths taken by messages.[21]

$$b_i = \sum_{s,t} w^i_{s,t} = \sum_{s,t} \frac{n^i_{s,t}}{n_{s,t}}$$

  where $n^i_{s,t}$ is total number of shortest paths from node s to node t passing from node i and $n_{s,t}$ is total number of shortest paths from node s to node t.

Additional features can include

- Geographic location of the house
- Historical label of the house
- Building type of the house
- Number of windows of the room
- Width of windows of the room
- Number of doors to access a room
- Number of other wall-adjacent rooms

We defined what the input of the neural network should be. Now we need to decide how many categories of privacy we would like the neural network to model the classification on. In the simple example of statistical analysis that follows, we used a binary classification in the first case (either public or private spaces) and a ternary classification in the second (public, intermediate, private).
Having a matrix of n datapoints (rooms) and their d features, and a number c of output classes, we can feed the input to a multi-layer neural network with a d-dimensional first layer of neurons, a c-dimensional output layer (fully connected layer and softmax layer), and a number of intermediate layers with non-linear activation function.

[20] Wikipedia, "Degree Centrality"
[21] https://www.sci.unich.it/ francesc/teaching/network/betweeness.html

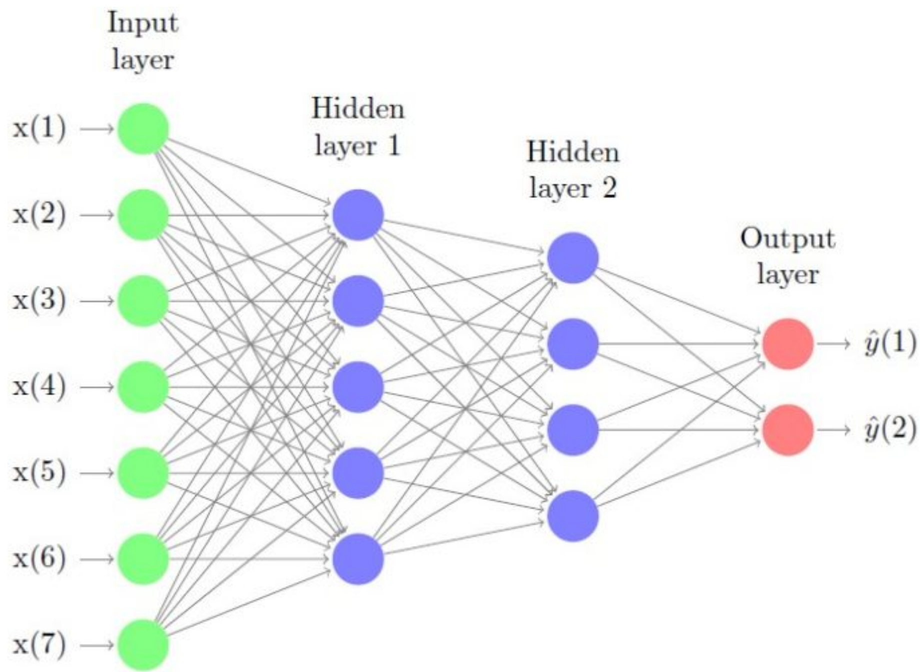


Figure 1: An illustrative example of a feed-forward neural network with two hidden layers, seven features and two output states. Deep learning network classifiers typically have many more layers, use a large number of features and several output states or classes. The goal of learning is to find the weight on every edge that minimizes the out-of-sample error measure.

Figure 43: Example of Neural Network for classification with two classes, from VW, Classification-Based Financial Markets Prediction Using Deep Neural Networks

The dataset would need to be split in training data, validation data and test data [22]. For every data point, we know an assigned label y. In order to train the neural network, we must propagate forward the input feature values through the layers of neurons (model parameters). Once we obtain an outpout, we are able to compare that predicted label $\hat{y}$ with the real label y and compute a loss function $L(y, \hat{y})$. Through backpropagation, we then propagate backwards by adjusting the model parameters so to make them a better fit for the task. This process is repeated on the whole set of datapoints for a number of cycles (epochs), so to decrease the loss function. The training time must be fine tuned by comparing the results on a validation set. We can go on decreasing the training loss forever. However, at some point the model will overfit (learn to identically replicate the patterns, which is bad for generalization). In order to avoid overfitting, we must stop training at the epoch where the validation loss stops decreasing.
Once the optimal model parameters are computed, we can feed in test data and compute the model accuracy on prediction on unseen data. If the accuracy is satisfactory, we can use the model on new unseen data for classification.

[22] for instance, 60% training data, 20% validation data, 20% test data

9 A simple example

Given enough data, deep neural networks have been proved to outperform other algorithms at several tasks. We did sketch the requirements for a software extracting both graph representation and measures and isovist measures. However, the full implementation of such software was outside the scope of this thesis. To demonstrate how a neural network system can learn architectural patterns and become a performing tool for Space Syntax analysis, we will first run a sample analysis on a relatively small dataset of 97 room datapoints. The hypothesis in this experiment is that plans with different cultural profiles share a common pattern relating the level of intimacy and some selected spatial features. We will represent and comment the relationships among the different features, and we will run a set of statistical algorithms on this data for classification. We will comment the performance of these different algorithms, and observe how a deep neural network can outperform such models on the base of its complexity and its ability to approximate more subtle separator functions between classes.

9.1 A sample dataset

A small dataset of nine architectural plans of houses, for a total of 97 rooms, was selected from the archive in Divisare, a web-based architecture portal [23]. These plans where selected randomly from the category of projects in Divisare that included the architectural drawing details, as we needed the plan drawings. Although the choice was random, the fact that these drawings belonged to the same category of projects whose architectural documents were available might induce bias to some extent, in the sense that Divisare might have a selection bias on the projects it published [24]. However, for further analysis we will assume that the sample of plans is unbiased.
The plans represent living units of various building types (apartments, villas, renovated historic apartments), size (from four to seventeen rooms) and geographical locations (United States, Europe, Japan). Two of them are on multiple levels, while the others are on the same level.
Here's the list of sample plans as numbered in the Appendix:

- Plan n.1 - Contemporary American house
- Plan n.2 - Renovation of apartment in XVII century Roman building, Italy
- Plan n.3/4 - Renovation of apartment in XIX century Lisbon building, Portugal
- Plan n.5 - Renovation of apartment in XIX century Lisbon building, Portugal
- Plan n.6 - Apartment in contemporary apartment building, Italy
- Plan n.7 - Apartment in contemporary apartment building, Portugal
- Plan n.8 - Apartment in contemporary apartment building, Germany
- Plan n.9 - Contemporary villa, Japan

From these plans, the graph representation was extracted together with the degree centrality and betweenness centrality. For each convex space in the plan, its area, and room area to total area ratio were computed. Also, the centered isovists were computed for each room, as well as the ratio between isovist area and room area/total area [25]. Each room was labeled according to its function so to have a privacy value. Two possible gradients of privacy were chosen - binary and ternary privacy. In the binary case, privacy = 0 corresponds to spaces for the common living and interaction, such as living rooms, dining rooms, kitchens, connection spaces. The case privacy = 1 corresponds to bedrooms, bathrooms, dressings and other spaces for individual use. In the ternary case, privacy = 0 corresponds again to spaces for the common living and interaction, such as living rooms, dining rooms, kitchens, connection spaces. The case privacy = 2 corresponds to bedrooms, bathrooms, dressings and other spaces for individual use. The case privacy=1 is a hybrid category including studios, reading rooms, playrooms and other spaces that are neither explicitly private nor open to interaction. These spaces where categorized as 0-privacy spaces in the binary classification scenario.

[23] Divisare is an independent ad-free archive of contemporary architectural designs. It can be found at the link https://divisare.com/

[24] For example, Divisare might collect only those projects that appeared in other publications or archives, or apply specific rules in their portfolio selection

[25] These features correspond to the ones listed in Section 9.5

We collected a total of 97 room datapoints with respective features. The set of measures for each room and each plan is reported in an Excel spreadsheet in Figures below.

Sample	Room	Room Area	Total Floor Area	Room to Total Area	Centered Isovist	Centered Isovist to Room Floor	Centered Isovist to Total Floor	Number of visual neighbors	Function	Privacy	Degree Centrality	Betweennes s Centrality
1	Room 1	79.6	222.1	0.358	122	1.53	0.55	6	Living	0	4	37
1	Room 2	28.0	222.1	0.126	40.9	1.46	0.18	3	Bedroom	1	2	18
1	Room 3	14.7	222.1	0.066	21	1.43	0.09	2	Bathroom	1	2	10
1	Room 4	6.2	222.1	0.028	9.5	1.53	0.04	1	Closet	1	1	0
1	Room 5	8.7	222.1	0.039	9.8	1.13	0.04	1	Laundry	0.5	1	0
1	Room 6	5.4	222.1	0.024	21.4	3.96	0.10	1	Storage	0	1	0
1	Room 7	11.9	222.1	0.054	32.5	2.73	0.15	4	Distribution	0	5	39
1	Room 8	5.1	222.1	0.023	8	1.57	0.04	2	Dressing	0.5	2	10
1	Room 9	4.7	222.1	0.021	6.5	1.38	0.03	1	Bathroom	1	1	0
1	Room 10	15.3	222.1	0.069	19.9	1.30	0.09	2	Bedroom	1	1	0
1	Room 11	21.2	222.1	0.095	24.2	1.14	0.11	1	Bedroom	1	1	0
1	Room 12	21.3	222.1	0.096	25.3	1.19	0.11	2	Bedroom	1	1	0
2	Room 1	11.5	162	0.071	10.7	0.93	0.07	1	Distribution	0	2	28
2	Room 2	12.0	162	0.074	15.6	1.30	0.10	2	Kitchen	0	2	24
2	Room 3	14.2	162	0.088	17.5	1.23	0.11	2	Dining	0	2	18
2	Room 4	15.0	162	0.093	17.2	1.15	0.11	2	Studio	0.5	2	10
2	Room 5	20.5	162	0.127	22.3	1.09	0.14	1	Living	0	1	0
2	Room 6	11.9	162	0.073	13.9	1.17	0.09	1	Playroom	0.5	2	30
2	Room 7	12.0	162	0.074	15.6	1.30	0.10	2	Bedroom	1	2	30
2	Room 8	9.1	162	0.056	24.5	2.69	0.15	5	Dressing	0.5	3	31
2	Room 9	2.6	162	0.016	4.1	1.58	0.03	1	Bathroom	1	1	0
2	Room 10	14.2	162	0.088	17.5	1.23	0.11	3	Boudoir	1	3	19
2	Room 11	2.0	162	0.012	13.6	6.80	0.08	2	Bathroom	1	1	0
2	Room 12	20.5	162	0.127	22.3	1.09	0.14	1	Bedroom	1	1	0
3	Room 1	16.5	108.3	0.152	39.9	2.42	0.37	2	Studio	0.5	1	0
3	Room 2	27.9	108.3	0.258	47.1	1.69	0.43	2	Living	0	2	2
3	Room 3	31.8	108.3	0.294	53.1	1.67	0.49	3	Kitchen	0	2	2
3	Room 4	32.1	108.3	0.296	36.9	1.15	0.34	2	Bedroom	1	1	0
4	Room 1	21.0	94	0.223	31.6	1.50	0.34	3	Living	0	2	9
4	Room 2	20.4	94	0.217	31.3	1.53	0.33	2	Bedroom	1	2	8
4	Room 3	21.7	94	0.231	35.3	1.63	0.38	4	Closet	1	2	5
4	Room 4	6.1	94	0.065	6.6	1.08	0.07	1	Bathroom	0	1	0
4	Room 5	9.8	94	0.104	23.4	2.39	0.25	3	Kitchen	0	2	8
4	Room 6	8.2	94	0.087	30.3	3.70	0.32	5	Storage	0	2	5
4	Room 7	6.8	94	0.072	11.7	1.72	0.12	2	Studio	0.5	1	0
5	Room 1	3.7	125.4	0.029	19.9	5.44	0.16	4	Distribution	0	2	13
5	Room 2	8.1	125.4	0.065	17.628	2.18	0.14	8	Distribution	0	8	56
5	Room 3	11.0	125.4	0.087	21.244	1.94	0.17	3	Kitchen	0	1	0
5	Room 4	16.5	125.4	0.132	22.9	1.39	0.18	5	Bedroom	1	2	12
5	Room 5	4.4	125.4	0.035	8.1	1.84	0.06	1	Bathroom	1	1	0
5	Room 6	4.5	125.4	0.036	6.6	1.47	0.05	2	Bathroom	1	1	0
5	Room 7	1.9	125.4	0.015	11	5.82	0.09	5	Distribution	0	1	0
5	Room 8	8.0	125.4	0.064	24.7	3.09	0.20	3	Closet	1	2	3

Sample	Room	Room Area	Total Floor Area	Room to Total Area	Centered Isovist	Centered Isovist to Room Floor	Centered Isovist to Total Floor	Number of visual neighbors	Function	Privacy	Degree Centrality	Betweennes s Centrality
5	Room 9	19.5	125.4	0.155	30.4	1.56	0.24	7	Living	0	4	10
5	Room 10	11.0	125.4	0.088	21.9	1.99	0.17	6	Dining	0	3	13.5
5	Room 11	9.9	125.4	0.079	12	1.21	0.10	2	Reading/Studio	0.5	1	0
5	Room 12	10.5	125.4	0.084	22.1	2.10	0.18	5	Bedroom	1	2	2
5	Room 13	9.1	125.4	0.073	20.9	2.29	0.17	4	Closet	1	2	3
5	Room 14	7.4	125.4	0.059	17.5	2.36	0.14	4	Closet	1	2	6.5
6	Room 1	45.6	172.6	0.264	67.4	1.48	0.39	4	Living	0	3	35
6	Room 2	1.8	172.6	0.010	21.4	12.09	0.12	3	Distribution	0	3	21
6	Room 3	3.6	172.6	0.021	4.6	1.28	0.03	1	Bathroom	1	1	0
6	Room 4	10.7	172.6	0.062	12.8	1.20	0.07	2	Bedroom	1	1	0
6	Room 5	21.3	172.6	0.123	40.8	1.92	0.24	4	Kitchen	0	1	0
6	Room 6	14.4	172.6	0.083	15.6	1.08	0.09	2	Distribution	0	7	51.5
6	Room 7	11.0	172.6	0.064	16	1.45	0.09	3	Reading/Studio	0.5	2	4.5
6	Room 8	17.4	172.6	0.101	18.7	1.07	0.11	1	Bedroom	1	2	0.5
6	Room 9	5.2	172.6	0.030	6.6	1.27	0.04	1	Bathroom	1	3	5.5
6	Room 10	5.7	172.6	0.033	9.3	1.63	0.05	1	Laundry	0.5	2	0
6	Room 11	4.4	172.6	0.025	5.3	1.20	0.03	1	Bathroom	1	1	0
6	Room 12	15.8	172.6	0.092	17.22	1.09	0.10	1	Bedroom	1	1	0
6	Room 13	15.7	172.6	0.091	16.8	1.07	0.10	2	Bedroom	1	1	0
7	Room 1	50.2	170.8	0.294	60.2	1.20	0.35	2	Living	0	2	8
7	Room 2	11.7	170.8	0.069	54.2	4.63	0.32	2	Reading/Studio	0.5	1	0
7	Room 3	13.2	170.8	0.077	35.7	2.70	0.21	3	Distribution	0	7	34
7	Room 4	3.5	170.8	0.020	5.4	1.54	0.03	1	Bathroom	1	1	0
7	Room 5	14.8	170.8	0.087	16	1.08	0.09	1	Bedroom	1	1	0
7	Room 6	14.5	170.8	0.085	16.1	1.11	0.09	1	Bedroom	1	1	0
7	Room 7	16.4	170.8	0.096	22.3	1.36	0.13	3	Bedroom	1	2	8
7	Room 8	7.1	170.8	0.042	10.84	1.53	0.06	1	Bathroom	1	1	0
7	Room 9	12.2	170.8	0.071	15.3	1.25	0.09	2	Kitchen	0	1	0
7	Room 10	27.2	170.8	0.159	25.9	0.95	0.15	0	Dining	0	1	0
8	Room 1	8.6	98.7	0.087	32.8	3.81	0.33	5	Distribution	0	5	12.5
8	Room 2	14.1	98.7	0.143	17.8	1.26	0.18	3	Bedroom	1	1	0
8	Room 3	16.5	98.7	0.167	27.4	1.66	0.28	3	Bedroom	1	1	0
8	Room 4	6.8	98.7	0.069	18.1	2.66	0.18	3	Distribution	0	2	2
8	Room 5	8.0	98.7	0.081	28.7	3.59	0.29	3	Kitchen	0	2	2
8	Room 6	9.1	98.7	0.092	18.2	2.00	0.18	3	Bathroom	1	1	0
8	Room 7	35.6	98.7	0.361	41.3	1.16	0.42	2	Living	0	2	0.5
9	Room 1	6.8	144.3	0.047	18.5	2.72	0.13	3	Distribution	0	2	15
9	Room 2	11.4	144.3	0.079	21	1.84	0.15	5	Kitchen	0	1	0
9	Room 3	17.4	144.3	0.121	47.1	2.71	0.33	5	Dining	0	5	89
9	Room 4	21.1	144.3	0.146	41.6	1.97	0.29	4	Living	0	1	0
9	Room 5	4.3	144.3	0.030	31.3	7.28	0.22	8	Distribution	0	3	36
9	Room 6	4.1	144.3	0.028	16.9	4.12	0.12	4	Bathroom	1	2	15
9	Room 7	3.5	144.3	0.024	12.6	3.60	0.09	4	Bathroom	1	1	0
9	Room 8	6.2	144.3	0.043	18.3	2.95	0.13	4	Laundry	0.5	2	3
9	Room 9	10.0	144.3	0.069	20.2	2.02	0.14	5	Bedroom	1	2	4
9	Room 10	6.3	144.3	0.044	13	2.06	0.09	5	Distribution	0	5	50
9	Room 11	11.3	144.3	0.078	19.4	1.72	0.13	2	Bedroom	1	1	0
9	Room 12	9.2	144.3	0.064	15.5	1.68	0.11	2	Storage	0	1	0
9	Room 13	1.7	144.3	0.012	9.7	5.71	0.07	2	Bathroom	1	1	0
9	Room 14	4.9	144.3	0.034	17.6	3.59	0.12	3	Distribution	0	4	42
9	Room 15	11.4	144.3	0.079	12.4	1.09	0.09	1	Bedroom	1	1	0
9	Room 16	1.8	144.3	0.012	5.2	2.89	0.04	2	Bathroom	1	1	0
9	Room 17	12.9	144.3	0.089	13.8	1.07	0.10	1	Studio	0.5	1	0

Figure 44: Table of the collected data

9.2 First look at the data

To gain a first insight into the data, we can observe the pair plot of each feature as related to each of the other features, as well as the statistical distribution of each feature for the different levels of privacy. The pair plot for a two-class privacy is reported in Figure 53. The pair plot for a three-class privacy is reported in Figure **??**.

From the first pair plot we observe, first of all, that for several features the statistical distributions for privacy level 0 and privacy level 1 do decouple. For example, this is clearly visible for the variable number_of_visual_neighbors. The distribution for this variable in the case privacy=0 has larger mass on higher values, whereas in the case privacy=1, the distribution (in blue) has larger mass on lower values. We can interpret this as the private rooms to be visually connected to fewer visual neighbors than the more public rooms, on average. Similar considerations are true for the variables related to the centered isovist (centered isovist, centered isovist to room floor ratio, centered isovist to total floor ratio) (see Figure 53). Another relevant consideration is that the degree centrality of a room tends to have higher values for the non-private spaces then for the private spaces, which confirms the hypothesis that the spaces devoted to interaction tend to be more central in the graph representation and tend to be more highly integrated [26]. Betweenness centrality is also higher for non-private spaces and lower for private spaces, which again is a confirmation of the tendency to place private functions at locations of the house corresponding to less integrated nodes, which are very often terminal nodes, rather than connecting.

It is interesting to observe a few changes when we consider the pair plot for three levels of privacy (0,1,2) (see Figure **??**). Again, in several cases the distributions of the features decouple for different levels of privacy, which is evident for the number of visual neighbors as well as the centered isovist to total floor ratio and degree centrality. Betweenness centrality takes different values in the three cases: it is in fact evident from the box plot that privacy=0 spaces tend to have much higher values in betweenness centrality, with a median value of around 10; privacy=2 spaces, on the other side of the spectrum, tend to have very low values in betweenness centrality, with a median of 0; the spaces with intermediate privacy, which are hybrid spaces intended for individual use but not explicitly private, also have a median betweenness centrality value of 0, however the average value is between the other two categories. Another interesting consideration is that when looking at the variables room area, room to total area, centered isovist, centered isovist to total floor, number of visual neighbors, we observe spikes on the low values for the intermediate privacy spaces. For these spaces, the distributions seem to be skewed towards the low values. This can be interpreted as these spaces being more similar to the privacy=2 spaces than the privacy=0. However, the median degree centrality for spaces of privacy=1 is the exact same as the spaces of privacy=2. These facts can be interpreted as the hybrid spaces sharing a mix of characteristics of the namely private spaces and the namely open, public spaces. Also, intermediate privacy spaces make for most of the low values in the room area distribution.

In general, it is also worth noting how degree centrality is positively correlated with the number of visual neighbors (correlation is 0.459 for the spaces with privacy=0, 0.609 for the spaces with privacy=1 and 0.474 for the spaces with privacy=2), as well as betweenness centrality (correlation is 0.389 for the spaces with privacy=0, 0.369 for the spaces with privacy=1 and 0.346 for the spaces with privacy=2). This can be interpreted as a tendency of central spaces[27] to also be spaces of high visual connectivity and of high physical connectivity. As seen in Part III, these spaces are perceived as good spaces for interaction, such as living rooms. Spaces with low degree centrality, on the other side, tend to have lower visual connectivity and betweenness centrality, and as verified in Part II they tend to be more easily perceived as intimate spaces, thanks to the fact that they result more segregated.

[26] Just like in Bill Hillier's results presented in *Ideas are in things*, 1981

[27] spaces that are central to the graph

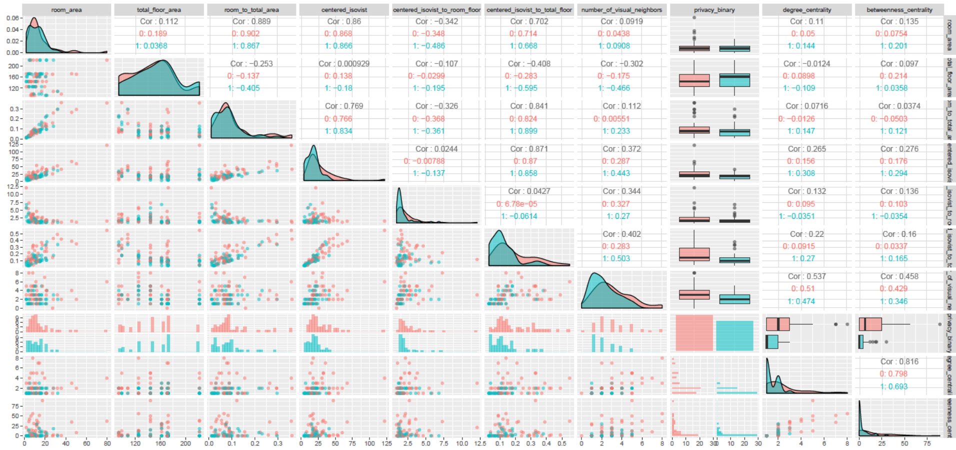


Figure 45: Pairplot - binary privacy

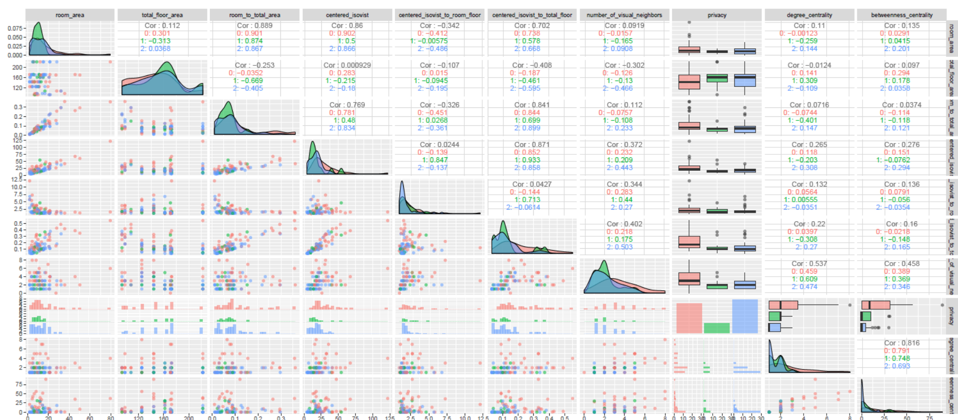


Figure 46: Pairplot - gradient (3-class) privacy

9.3 Analysis and discussion

Several statistical algorithms for classification were trained to learn how to correctly label the privacy of rooms based on their Space Syntax characteristics. A dataset of 29 datapoints (30% of the original dataset) was held-out as test data. The list of the models and the related accuracy on test data is reported in Table 1 (binary classification) and in Table 2 (ternary classification).
It emerged that in the binary classification task the accuracy on the privacy=0 class was always higher than the accuracy on the privacy=1 class. This can be interpreted as the non-private spaces having a simpler to detect relationship with the variables in the model. The highest accuracy for the privacy=0 class (92.86%) was achieved in the Support Vector Machine model with Radial Basis Function, Gradient Boosted Trees, Gradient Boosted Trees with optimization, Random Forest with scaled features. The best accuracy for the privacy=1 class is achieved in the Optimized Nearest Neighbors, Support Vector Machine with feature scaling and Support Vector Machine with Radial Basis Function and scaled features, as well as Gradient Boosted Trees with scaled features. The overall best performance is achieved by the Gradient Boosted Trees with scaled features, which was able to correctly classify 12/14 cases of privacy=0 rooms and 9/15 cases of privacy=1. The interpretation for this is that this particular algorithm was able to approximate the separator function separating private from public spaces in such a way to obtain very good accuracy on the public spaces, and satisfactory but lower accuracy in detecting private spaces.
When looking at the results for the 3-class classification models, we observe that the best overall accuracy is achieved by the Logistic Regression with L1 Penalty. What is extremely interesting to observe is that no model was able to correctly predict the class of privacy=1 rooms, which are the hybrid, intermediate privacy rooms. Despite achieving high accuracy on the other two classes, none of the models here presented was able to beat the 0% accuracy. This fact is interesting because it can be interpreted as the inability, with this small dataset and with these algorithms, to model a complex-enough separator function able to capture class 1. The algorithms classified class 1 rooms as either straight-private or straight-public spaces. This result not only reinforces the idea of the complexity in studying the relationship between spatial features and qualities, but it also fully justifies the need for a more complex function approximator, such as a Deep Neural Network, which given enough labeled data would more likely achieve better classification performance.
Another reason why these results are interesting is that they show how even with a relatively small dataset, given enough spatial features, models were able to detect significant patterns of relations between Space Syntax features and the level of privacy of different rooms in a floor plan.

Model	0 Class Accuracy	1 Class Accuracy
Nearest Neighbors	78.57 % (11 / 14)	53.33 % (8 / 15)
Nearest Neighbors (Scaled)	78.57 % (11 / 14)	46.67 % (7 / 15)
Nearest Neighbors (Optimized)	78.57 % (11 / 14)	60.0 % (9 / 15)
SVM	78.57 % (11 / 14)	53.33 % (8 / 15)
SVM (Scaled)	71.43 % (10 / 14)	60.0 % (9 / 15)
SVM (Optimized)	78.57 % (11 / 14)	53.33 % (8 / 15)
SVM (RBF Kernel)	92.86 % (13 / 14)	20.0 % (3 / 15)
SVM (RBF Kernel) (Scaled)	78.57 % (11 / 14)	60.0 % (9 / 15)
SVM (RBF Kernel) (Optimized)	92.86 % (13 / 14)	20.0 % (3 / 15)
Logistic Regression	85.71 % (12 / 14)	53.33 % (8 / 15)
Logistic Regression (Scaled)	78.57 % (11 / 14)	46.67 % (7 / 15)
Gradient Boosted Trees	92.86 % (13 / 14)	46.67 % (7 / 15)
Gradient Boosted Trees (Scaled)	85.71 % (12 / 14)	60.0 % (9 / 15)
Gradient Boosted Trees (Optimized)	92.86 % (13 / 14)	46.67 % (7 / 15)
Logistic Regression - L1 Penalty	85.71 % (12 / 14)	53.33 % (8 / 15)
Logistic Regression - L1 Penalty (Scaled)	78.57 % (11 / 14)	46.67 % (7 / 15)
Random Forest	71.43 % (10 / 14)	46.67 % (7 / 15)
Random Forest (Scaled)	92.86 % (13 / 14)	26.67 % (4 / 15)
Random Forest (Optimized)	71.43 % (10 / 14)	46.67 % (7 / 15)

Table 1: Privacy Labels 0 and 1

Model	0 Class Accuracy	1 Class Accuracy	2 Class Accuracy
Nearest Neighbors	41.67 % (5 / 12)	0.0 % (0 / 4)	61.54 % (8 / 13)
Nearest Neighbors (Scaled)	33.33 % (4 / 12)	0.0 % (0 / 4)	53.85 % (7 / 13)
Nearest Neighbors (Optimized)	50.0 % (6 / 12)	0.0 % (0 / 4)	53.85 % (7 / 13)
SVM	66.67 % (8 / 12)	0.0 % (0 / 4)	61.54 % (8 / 13)
SVM (Scaled)	66.67 % (8 / 12)	0.0 % (0 / 4)	53.85 % (7 / 13)
SVM (Optimized)	58.33 % (7 / 12)	0.0 % (0 / 4)	53.85 % (7 / 13)
SVM (RBF Kernel)	83.33 % (10 / 12)	0.0 % (0 / 4)	30.77 % (4 / 13)
SVM (RBF Kernel) (Scaled)	66.67 % (8 / 12)	0.0 % (0 / 4)	69.23 % (9 / 13)
SVM (RBF Kernel) (Optimized)	83.33 % (10 / 12)	0.0 % (0 / 4)	30.77 % (4 / 13)
Logistic Regression	75.0 % (9 / 12)	0.0 % (0 / 4)	69.23 % (9 / 13)
Logistic Regression (Scaled)	66.67 % (8 / 12)	0.0 % (0 / 4)	61.54 % (8 / 13)
Gradient Boosted Trees	66.67 % (8 / 12)	0.0 % (0 / 4)	23.08 % (3 / 13)
Gradient Boosted Trees (Scaled)	66.67 % (8 / 12)	0.0 % (0 / 4)	53.85 % (7 / 13)
Gradient Boosted Trees (Optimized)	66.67 % (8 / 12)	0.0 % (0 / 4)	30.77 % (4 / 13)
Logistic Regression - L1 Penalty	83.33 % (10 / 12)	0.0 % (0 / 4)	69.23 % (9 / 13)
Logistic Regression - L1 Penalty (Scaled)	66.67 % (8 / 12)	0.0 % (0 / 4)	69.23 % (9 / 13)
Random Forest	50.0 % (6 / 12)	0.0 % (0 / 4)	61.54 % (8 / 13)
Random Forest (Scaled)	58.33 % (7 / 12)	0.0 % (0 / 4)	38.46 % (5 / 13)
Random Forest (Optimized)	58.33 % (7 / 12)	0.0 % (0 / 4)	46.15 % (6 / 13)

Table 2: Privacy Labels 0, 1 and 2.

10 Contributions, applications and future work

The contribution of this thesis consists of three parts. First, the introduction of the possibility of a Machine Learning Space Syntax tool itself; it is, in fact, a novel idea. The requirements of such a system were detailed, and a second contribution consists in introducing a software for the extraction of the graph representation from a floor plan image, rather than vector drawing. The program is not robust to style, but it represents a first step towards a fully-functional feature extraction software from architectural images, which *per se* has not been attempted yet. Although in the context of this thesis it was practically unfeasible to build a large enough data set to run a Deep Neural Network model, we did verify that other statistical models where able to achieve satisfactory binary classification results for the level of privacy when trained and tested on a dataset of 97 architectural rooms. This represents a third contribution, as the statistical analysis included variables that have not been found in previous studies, such as the number of visual neighbors and the isovist-to-total-floor ratio. Since the three-class classification worked well on very private or very open spaces, but not for hybrid spaces, these results also reinforce the idea that a Deep Neural Network algorithm, being a more refined function approximator, could better capture the relationship linking spatial features and abstract qualities (such as privacy).

Once a Deep Neural Network has been trained to learn spatial patterns (for instance the level of privacy of the base of spatial features), we might ask ourselves what its main applications would be. First of all, we can think of using such Machine Learning workflow as an analytic tool in the context of design research. Different architectural qualities can be targeted through Machine Learning techniques. Of course, depending on the type of dataset that is taken as an input, different goals and results can be achieved. With a randomly selected and generic large dataset, the algorithm could detect generalized patterns. By specifically pre-selecting the dataset, for example targeting a specific architectural type or culture, the system would output type/culture-specific rules. Analytic applications can be used by Space Syntax researchers, architects and designers interested in a computational approach to spatial qualities.

After training Deep Neural Network, the model can be conversely used in a generative way - to recover the best layout to fit specific qualitative requirements, such as privacy. In this sense, the new tool will also benefit designers in their creative work [28]. Moreover, by working out the feature extraction software in such a way to run plan image processing in almost real time we can envision potential applications in the context of the real estate search tools, as a mobile app able to render the privacy levels of different rooms in a plan would likely represent a useful resource in this field.

Through the discussion and results here presented, we explored a yet unexplored matter - a Machine Learning workflow in Space Syntax analysis. We showed how processing spatial data from a Space Syntax analysis through a Neural Network can improve our ability to describe the relationship between quantitative spatial features and qualitative characteristics, such as privacy. This thesis opens the way to further work towards a Machine Learning framework in Space Syntax. Because of the nature of this inquiry, several open questions are left to future work.

First of all, a fully-robust software achieving the extraction of visibility and graphical measures from plan images is yet to be achieved. This resulted in the impossibility to create a large dataset of plans labeled according to their spatial characteristics in the context and constraints of this thesis. However, building the fully functional software performing these tasks was out of the scope of this inquiry, and is left as an important future milestone to achieve, in order to train Neural Networks to learn spatial patterns from large-base datasets.

Second, and most important, we saw the limitations we incur into when researching spatial quality under the lenses of Space Syntax analysis on architectural plans. The main reasons for extending this work is that despite the plan being a very information-dense document, it does not encode *all* the information about space. It leaves out height, light, color, materiality, smell, sound. An attempt to include sound and smell in the Space Syntax analysis of floor plan has been authored by Michael Georgiou [14]. It would be useful to include the related features in a Machine Learning framework as well. In general, a very interesting extension on this path would be to include in the feature vector a more extended set of sensory inputs, which could be collected, for example, by a robotic agent traversing spaces in a building. This would allow to keep record of sensory stimuli of different kind, including light, color and the other aspects that a simple Space Syntax analysis would not capture. Of course, the time required to capture this type of information for a large dataset of buildings is

[28] Such generative Machine Learning framework represents a very interesting and rich research field, and is left as the author's future doctoral work

considerable, and this direction seems to be relatively unexplored.
Another extension worth exploring is the collection of subject-based evaluation of the privacy of rooms in a building. The way rooms were labeled in this study was on the base of their function, which was directly transformed into a category (for example, all kitchens were considered privacy=0 spaces). However, another way of quantifying privacy would be through accessing the individual perception of a sample of participants reacting to spaces in different rooms. Such individual-based experiment would of course allow for a more complex definition of privacy levels.

11 Conclusions

In this thesis, we introduced a Machine Learning framework as an approach to inferring architectural qualities as a function of quantitative (or quantifiable) spatial features. After an overview of the traditional Space Syntax techniques and methods, we acknowledged previous studies targeting architectural qualities through graph and visibility properties. We took as an object of further study the quality of intimacy and privacy of different rooms in a house plan. Through a validation experiment, we observed patterns between spatial characteristics and the level of spatial intimacy as perceived by a group of experts. While statistical studies already exist in Space Syntax, such as the studies by Bill Hillier, we found that these are typically based on small datasets and a small set of features. We therefore suggested that a Machine Learning working on larger data would allow for more accurate results in the prediction of the level of intimacy of spaces in a floor plan.
Further development of this work includes building a larger dataset of architectural spaces labeled according to a long enough vector of spatial features, so to train a Deep Neural Network for the task of privacy classification. The impact of the introduction of a Neural Network framework in spatial analysis is multifaceted: depending on what dataset is taken as an input, a Machine Learning algorithm can output different kinds of new knowledge. In the analytic applications, a randomly selected dataset would allow for inference of general rules and qualities, that hold true across cultures. On the opposite side, a culture or type specific dataset will result in gaining specialized knowledge about the culture or type under consideration. The interpretation of the results of a Machine Learning model must therefore attentively take into account the nature of the dataset. The impact of these new paths on the analytic side will mainly benefit researchers, architects and designers dealing with spatial analysis. In the longer term, we can imagine this framework to be integrated in the context of an intelligent design assistant. Finally, two important future steps in the direction of a more comprehensive understanding of the quality of privacy. First, the inclusion of 3D information in the analysis. As we saw, it would be interesting to have 3D data automatically collected by a robotic device traversing a building (in real life or in simulation). This would allow for a richer set of descriptors in the model. Second, it can be interesting to collect survey feedback on the perceived privacy of rooms in a plan from a large-base sample. by substituting the single expert evaluation with the evaluation by a larger base sample of people, we could allow for a less deterministic method for labeling the level of privacy of each room in a sample plan. An appendix presents a Google survey that has been designed as a source of data to cross-validate the results of the statistical analysis in Part III.
Coming back to our initial research question, we can state that a Machine Learning workflow does have the potential to improve spatial analysis on the floor plan and can help researchers to characterize abstract architectural qualities in terms of quantifiable spatial features. Architecture remains one of the fields in which introducing Machine Learning frameworks is the least intuitive. This is due to the fact that large labeled datasets of architectural information are rare, and that many features in architecture are abstract and qualitative in nature. The Machine Learning framework we propose would impact the core itself of how spatial analysis work is done in architecture - and has been done for decades. It will realistically open new possibilities in the study of qualitative aspects in architecture, and bridge spatial analysis to the era of learning machines.

Part IV

Appendices

12 Appendix: Code

```
%%%WALL AND DOOR SEGMENTATION


im = imread('thesisPlans/ex3rem.png');
imWhite = ones(size(im));
im = im2bw(im, 0.5);

sizeW = size(im, 2);
sizeH = size(im, 1);

%find Harris corners
corners = corner(im,'harris');
imshow(im)
hold on
plot(corners(:,1), corners(:,2), 'r*');

nearestNeighbors = cell(length(corners),1);

%build a list of the nearest neighbor of each interest point
for i=1:length(corners)
    minDist = 10000;
    minDistIndex = 10000;
    for j=1:length(corners)
        if i~=j
            p1 = corners(i,:);
            p2 = corners(j,:);
            p1X = p1(2);
            p1Y = p1(1);
            p2X = p2(2);
            p2Y = p2(1);
            distance = sqrt((p1X-p2X)^2 + (p1Y-p2Y)^2);
            if distance < minDist && (abs(p1X-p2X)<=10 || abs(p1Y-p2Y)<=10)
                minDistIndex = j;
                minDist = distance;
                minDist;
            end
        end
    nearestNeighbors{i} = minDistIndex;
    end
end

%for points that are on the same extreme of a wall, we are interested in
%their midpoint
midpoints = cell(length(corners),1);

for i=1:length(corners)
    point1 = corners(i,:);
    neighborIndex = nearestNeighbors{i};
    point2 = corners(neighborIndex,:);
```

```
    distance_p1_p2 = abs(pdist2(point1,point2));
    if distance_p1_p2 <= 100   %%%%%was 15
        midPoint = [(point1(:) + point2(:)).'/2];
        midpoints{i} = midPoint;
        plot(midpoints{i}(:,1), midpoints{i}(:,2), 'r*');
    end
end

imshow(im)
hold on
plot(corners(65,1), corners(65,2), 'r*');
indexneigh = nearestNeighbors{65}
plot(corners(indexneigh,1), corners(indexneigh,2), 'b*');

point1red = corners(65,:);
point2blue = corners(indexneigh,:);
distance_p1_p2 = abs(pdist2(point1red,point2blue));
currmidpoint = [(point1red (:) + point2blue(:)).'/2];
plot(currmidpoint(:,1), currmidpoint(:,2), 'y*')


cleanedMidpoints = round(cell2mat(midpoints));

imshow(im)
hold on
for i=1:length(cleanedMidpoints)
    viscircles([cleanedMidpoints(4,:)],2)
end

%only pick those corners that seem to be door/window corners
openIndexes = [];

%select only those points that appear to be on the extreme point of a wall,
%as this means that they delimit either a door or window or other opening
for i=1:length(cleanedMidpoints)
    mask = imcrop(im,[cleanedMidpoints(i,1)-5,cleanedMidpoints(i,2)-5,10,10]);
    %if the mask is white enough
    if sum(mask(:)==1)/numel(mask) > 0.3
        %openIndexes = cat(openIndexes, i)
        openIndexes = [openIndexes;i];
    end
end


finalPoints = cleanedMidpoints(openIndexes,:);

lines = {};

background = ones(sizeH,sizeW);
imshow(im)
hold on
for i=1:length(finalPoints)
    viscircles([finalPoints(i,:)],2)
end
finalPoints = finalPoints(1:2:length(finalPoints),:);

imshow(im)
hold on
%% Find aligned points
```

```
for i=1:length(finalPoints)
    %in this loop we leverage the alignment property of a floor plan
    candidates = cell(length(finalPoints));
    matchPoint = 0;
    minDist = 10000;
    point1 = finalPoints(i,:);
    candidates = []
    for j=1:length(finalPoints)
         point2 = finalPoints(j,:);

         if ~isequal(point1,point2)
            %check whether the points have same X or same Y and their
            %distance is close enough to create the conditions for it to be
            %an opening
            if (abs(point1(:,1)-point2(:,1))<=3 || abs(point1(:,2)-point2(:,2))<=3)
                candidates = horzcat(candidates, [j])
            end
         end
    end

    for candidate=1:length(candidates)
        point1
        candidates(candidate)
        finalPoints(candidates(candidate),:)
        distance = abs(pdist2(point1,finalPoints(candidates(1,candidate),:)));
        if distance <= minDist
            matchPoint = finalPoints(candidates(candidate),:);
            minDist = distance;
        end
    end

    %plot the found lines, unless there is no match
    if matchPoint~=0
        plot([point1(1), matchPoint(1)],[point1(2), matchPoint(2)],'r','LineWidth',3);
        hold on
    end
end


%%%GRAPH


%% Open skeleton image, convert, bw, regions
im_closed_0 = imread('closedWalls.jpg');
im_open = imread('walls.jpg');
h_pixels = size(im_closed_0,1);
w_pixels = size(im_closed_0,2);

im_doors = im_closed_0;

redChannel = im_doors(:, :, 1);
greenChannel = im_doors(:, :, 2);
blueChannel = im_doors(:, :, 3);

% Find where color is door color
mask = (redChannel == 255);
[rows, columns] = find(mask);  % Note [rows, columns] = [y, x], NOT [x, y]
```

```
%% Open image segmenter to segment the doors

doors = imread('doorsOnly.jpg');
doors = im2bw(doors, 0.5);
doors = imcomplement(doors);

doorsRegions = bwconncomp(doors, 4);
rpDoors = regionprops(doorsRegions);
centroidDoors = regionprops(doorsRegions,'Centroid');
doorLabelMatrix = labelmatrix(doorsRegions);

displayRegions = imagesc(doorLabelMatrix)


extremeA = cell(length(centroidDoors), 1);
extremeB = cell(length(centroidDoors), 1);

for i=1:length(centroidDoors)
    point = centroidDoors(i).Centroid
    x = point(:,1);
    y = point(:,2);
    A = [round(x-5), round(y-5)];
    extremeA{i,1} = A;
    B = [round(x+5), round(y+5)];
    extremeB{i,1} = B;
    plot(x,y, 'b*');
    hold on
end



%% Label spaces
im_closed = im_closed_0;
%convert to a binary image
BW = im2bw(im_closed, 0.5);

%label the semantic regions
L = bwlabel(BW);

%display regions
displayRegions = imagesc(L);

%% Find regions centroids and place them in a cell structure
rp = regionprops(L);
centroids = regionprops(L,'centroid');
centroids_cell = struct2cell(centroids);
%find list of pixels in every subregion and place them in a cell structure
pixels_in_region = regionprops(L,'PixelList');
pixels_in_region_cell = struct2cell(pixels_in_region);

imshow(im_closed)
hold on

%plot centroids
for i=1:length(centroids)
    point = centroids(i).Centroid
    point
    x = point(:,1)
```

```
    y = point(:,2)
    plot(x,y, 'r*');
    hold on
end


%% Construct and draw edges of the graph
edges = {}

for i=1:length(centroidDoors)
    roomA = 0;
    roomB = 0;
    for r=1:length(pixels_in_region_cell)
        pixelList = pixels_in_region_cell{r};
        for row=1:size(pixelList,1)
            currentVector = [pixelList(row,1), pixelList(row,2)];
            if currentVector == extremeA{i,1}
                roomA = r;
            end
            if currentVector == extremeB{i,1}
                roomB = r;
            end
        end
    end
    if roomA ~= roomB && roomA~=1 && roomB~=1
        pointA = round(centroids_cell{roomA})
        xA = pointA(2);
        yA = pointA(1);
        viscircles([yA,xA],2);
        pointB = round(centroids_cell{roomB});
        xB = pointB(2);
        yB = pointB(1);
        viscircles([yB,xB],2);
        plot([yA,yB],[xA,xB],'Color','r','LineWidth',2)
        hold on
        %edges{end+1} = line(centroids_cell{roomA},centroids_cell{roomB});
    end
end



%% Room boundaries search
%[c, h] = contour(L)
%plot(C(1,(1:C(2,1))+1),C(2,(1:C(2,1))+1))
%boundaries = visboundaries(L);
boundaries = bwboundaries(BW);
boundariesList_0 = cell(length(boundaries),1);
countNonZeros = 0;
for k=1:length(boundaries)
   b = boundaries{k};
   boundariesList_0{k} = b;
   if size(b, 1) > 10
       countNonZeros = countNonZeros+1;
       plot(b(:,2),b(:,1),'g','LineWidth',3);
   end
end

%delete empty rows from cell and create final list of boundaries
boundariesList = cell(countNonZeros, 1);
```

```
k=1
for i=1:length(boundaries)
    boundaryElement = boundariesList_0{i};
    if size(boundaryElement,1)>10
        boundariesList{k} = boundaryElement;
        k = k+1;
    end
end
```

13 Appendix: Machine Learning essentials

13.1 What is Machine Learning?

Machine Learning is the field of computer science that studies of computers can exploit statistics to learn patterns out of data. With the increasing volume and availability of data, Machine Learning has grown over the past two decades to become a pervasive data analysis tool, and is applied to range of computing tasks where designing and programming explicit algorithms with good performance is difficult or unfeasible, but data availability is large enough to allow machines to automatically learn patterns.
Machine Learning tasks are subdivided into two broad categories, depending on whether or not there is a feedback available as a term of comparison for the learning system:

- Supervised learning. The computer is presented with a series of sample inputs and their desired outputs. These outputs are determined by an expert. The objective is to process these inputs and related outputs to learn a general rule mapping the first to the latter.
- Unsupervised learning. In this case, the computer is presented with only the sample inputs, and no labels. The objective of the learning system is therefore discovering hidden patterns in data.

There are different types of problems Machine Learning can solve:

- Classification. Inputs belong to two or more classes $\mathbf{y} = \{y_1, ..., y_n\}$. Each input has a series of features $\mathbf{x} = \{x_1, ..., x_n\}$ and the learning system must produce a model that assigns unseen inputs to one or more of these classes. One example of classification problem is image classification [29].
- Regression. This is another large set of Machine Learning problems. In this case the goal is to estimate a real-valued variable $y \in \mathbb{R}$ which the model relates as dependant on a function of the features $\mathbf{x} = \{x_1, ..., x_n\}$ and some parameter θ (see Figure 47):
$$y \approx f(\mathbf{x}, \theta)$$
A loss function is used to quantify the difference between our predicted y and the true y.
- Clustering. This is similar to a multi-class classification, except that the classes are not known beforehand. This makes the problem of clustering an unsupervised problem.

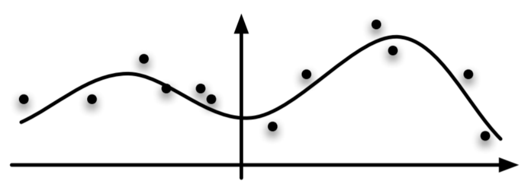


Figure 47: Example of regression. We are given a set of observations (the black dots) and would like to model a function f that maps the observations X to R such that f(x) is close to the observed values. From Alex Smola's *Introduction to Machine Learning* book draft.

13.2 Probability Theory essentials

Machine Learning relies on Probability Theory and its language. Here follows a brief, essential overview of Probability concepts.

- Random variable
A random variable - usually written X - is a variable whose possible values are numerical outcomes of a random phenomenon or process. Depending on the nature of the numerical

[29] Such as the well known ImageNet Large Scale Visual Recognition Challenge

outcomes, a random variable can be discrete or continuous. For example, when tossing a coin, the random variable X representing the outcome is $X = \{Head, Tail\}$. The number of possible outcomes is finite, therefore the random variable is discrete. In a different example, if we consider the weight of newborn babies, the random variable can take infinitely many values, as $X \in \mathbb{R}$, therefore it is continuous.

- Probability distribution
The most important way to characterize a random variable is to look at a function that matches possible outcomes with the probabilities at which they occur. In case of a discrete random variable, this assignment of probabilities is called a probability mass function (PMF). By definition of probability, PMF must be non-negative and sum to one. In the case described above, if the coin is a fair coin with equal probability of resulting in a head or tail, then the random variable X described above takes on values of +1 (head) and 1 (tail) with probability 0.5:
$$p(X = x) = \begin{cases} 0.5 & \text{if } X = +1, \\ 0.5 & \text{if } X = -1, \end{cases}$$
In a countinous case, we talk about Probability Density Function, or PDF. One very common PDF is the Gaussian Distribution, also called Normal, represented in Figure 48. The probability density for outcome x is represented by
$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}$$

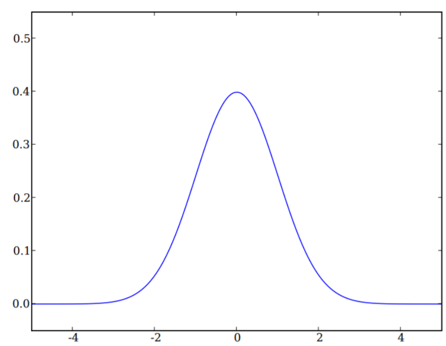


Figure 48: Example of Gaussian distribution (Normal). From Alex Smola's *Introduction to Machine Learning* book draft.

13.3 Neural Networks and Deep Learning

What is a neural network? To explain what a neural network is and how it works, we need to start from the concept of perceptron. It was scientist Frank Rosenblatt, inspired by earlier work by Warren McCulloch and Walter Pitts, who introduced perceptron in the 1950s and 1960s. The perceptron was a primitive model of the human neuron [30]. A perceptron is a processor unit that takes several binary inputs $x_1, x_2, ...$ and returns a single binary output y.

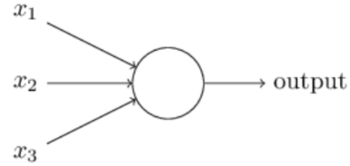


Figure 49: Scheme representing the primitive perceptron. From Michael Nielsen's Neural Network and Deep Learning Book, Chapter 1.

[30] Today it is more common to use other models of artificial neurons, such as the sigmoid neuron

A simple rule to compute the output involves assigning weights $w_1, w_2, \ldots$ to the inputs [31]. Weights are real numbers that represent the relative importance of the respective inputs to the output. The output is computed by taking the weighted sum $\sum_i w_i x_i$ and checking if it is smaller than or greater than some real-valued threshold t. In the first case, the output will be 0, in the latter, it will be 1:

$$y = \begin{cases} 0 & \text{if } \sum_i w_i x_i < t, \\ 1 & \text{if } \sum_i w_i x_i \geq t \end{cases}$$

In short, a perceptron weighs up different kinds of given information to make a decision.
Of course, a single perceptron cannot define very complex decision boundaries. That can be achieved, instead, by combining perceptrons in a larger network, as in Figure 50.

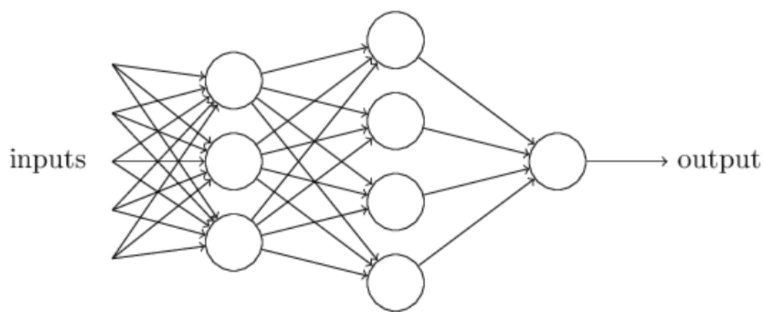


Figure 50: Scheme representing a network of perceptrons. From Michael Nielsen's Neural Network and Deep Learning Book, Chapter 1.

In this network, the first layer of perceptrons is weighing the input information and making three simple decisions. Their output (which is single) is sent out to the next layer of perceptrons, each making a decision by weighing up the results from the previous layer. Each layer advances the complexity of decision that is made possible. By stacking layers of perceptrons, the advanced layers can make a decision at a more complex level than the preceding perceptron layers. A multi-layer network of perceptrons can therefore outcome sophisticated decisions. This is the principle of a Deep Neural Network.
How does a multi-layer neural network work? The core idea is that we can design a learning algorithm to automatically adjust the weights of a Neural Network in order to adapt it to solve specific decision-making problems (for example, handwritten digit recognition and classification). This tuning happens in response to external stimuli, without direct intervention by a human. That is why it is called "Machine" Learning.
For the purpose of further analysis, we will transform the perceptron notation so that the threshold term t is moved to the right hand side of the inequalities and it is called "bias" (b). We will also adopt matrix notation and express $\sum_i w_i x_i$ as vector product $\mathbf{w} \cdot \mathbf{x}$.
The core idea in neural learning is that a small change in the weights $\mathbf{w}$ of the network will propagate forward in the computation and result in a small change in the output (see Figure 51).

[31] Introduced by Rosenblatt

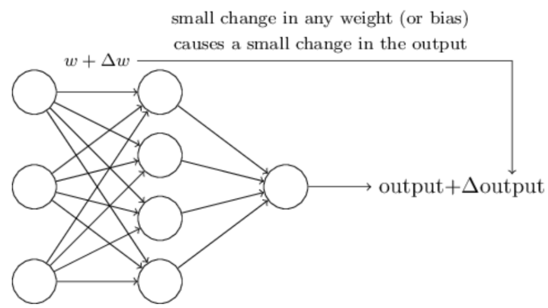


Figure 51: Forward propagation of a small change in the model weights. From Michael Nielsen's Neural Network and Deep Learning Book, Chapter 1.

In the handwritten digit example, the inputs to the network can be the raw pixel data from handwritten digit images. We build a multi-layer network and we would like to automatically learn the weights and biases of the model so that the output from the network correctly classifies the digit. If we make a small change in some weight or bias, this should cause only a small corresponding change in the output from the network. In fact, this property is what makes the learning process possible [32]. When the network is presented with a handwritten digit, suppose a 5, and the current set of weights and bias results in a computation that mis-classifies the 5 as a 6, we can slightly modify some weights and/or bias terms so to adjust the network to correctly label the input. We would have to repeat this procedure for every example the network is presented with, and for several epochs. Repeating this adjustment over and over characterizes the learning, and if correctly done would result in a better model.
However, this small change trick does not work as well if the network is formed out of simple perceptron, as a small change in a perceptron network can actually completely change the final outcome of the network. A better type of neuron is one that involves the use of non-linear activation functions in order to compute their local outcome. A sigmoid neuron is a neuron that receives inputs $(x_1, x_2, ...)$ with weights $(w_1, w_2, ...)$, just like the perceptron. The difference is that the output is not a binary output 0 or 1, but it is computed as $\sigma(\mathbf{w} \cdot \mathbf{x} + b)$, where σ is a sigmoid function defined by

$$\sigma(z) := \frac{1}{1 + e^{-z}}$$

The output of a sigmoid neuron would be:

$$y = \frac{1}{1 + e^{-(\sum_i w_i x_i + b)}}$$

This corresponds to replacing the step function activation in the perceptron with a smoother, continuous activation function in the sigmoid neuron (see Figure 52).

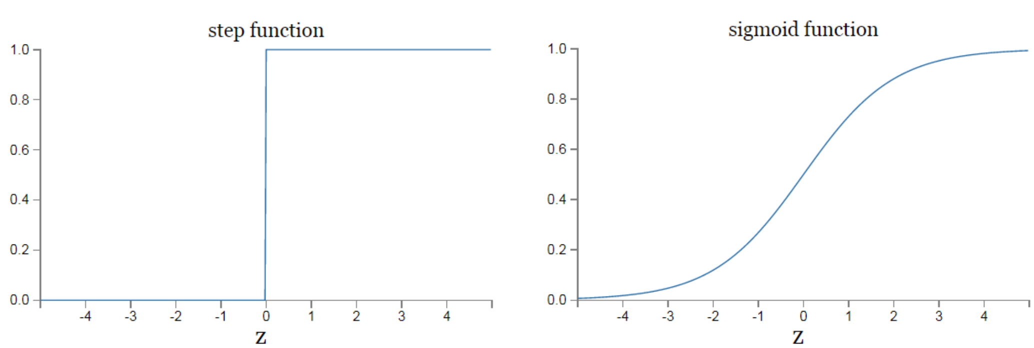


Figure 52: Comparison between perceptron's step activation functiona and sigmoid activation function. From Michael Nielsen's Neural Network and Deep Learning Book, Chapter 1.

[32]Michael Nielsen's Neural Network and Deep Learning Book, Chapter 1

When sigmoid activation neurons are used to form the neural network, we will observe that [33]

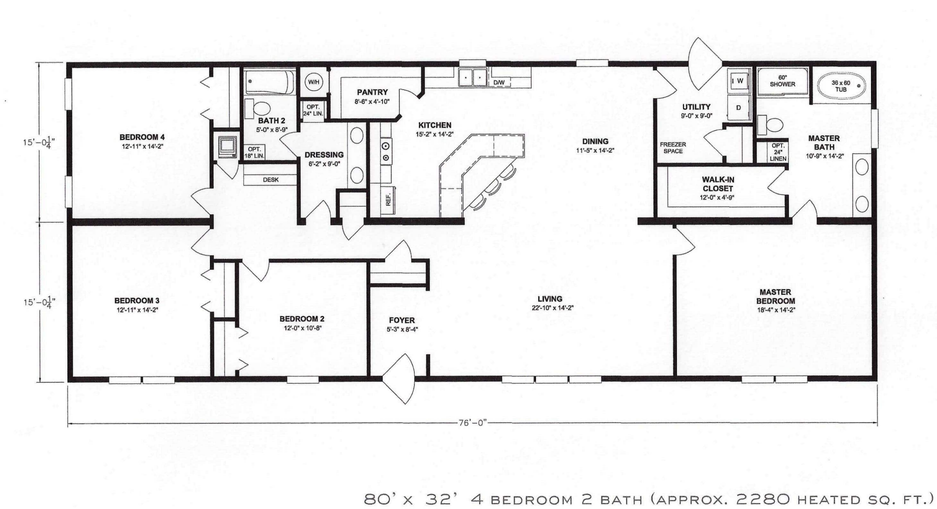
$$\Delta y \approx \sum_i \frac{\partial y}{\partial w_i} \Delta w_i + \frac{\partial y}{\partial b} \Delta b$$

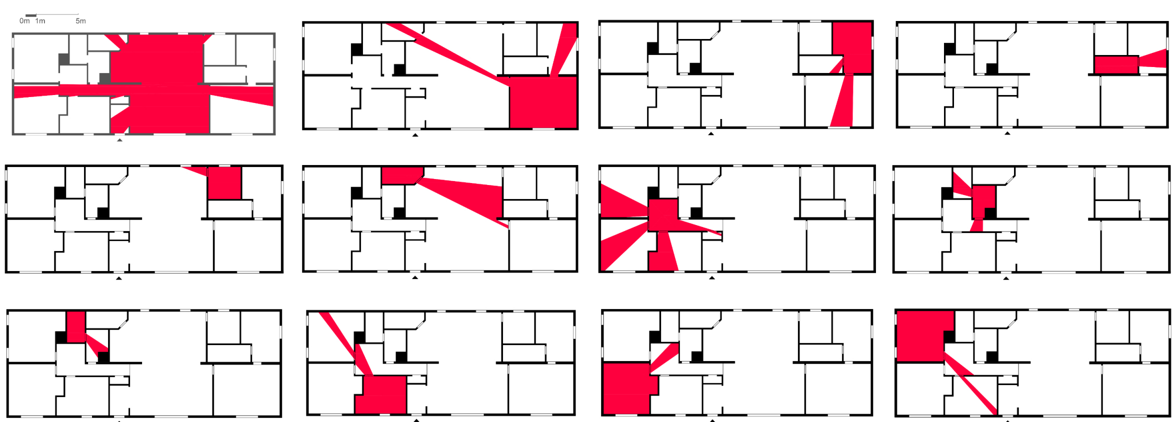
which makes the learning possible.

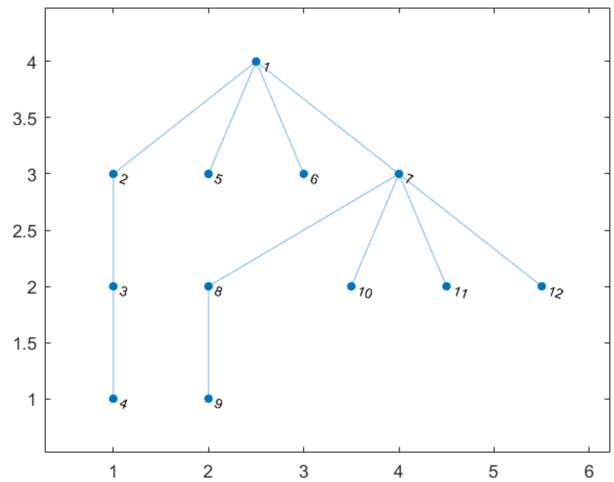
[33] Michael Nielsen's Neural Network and Deep Learning Book, Chapter 1

14 Appendix: Dataset

14.1 Sample Plan n.1

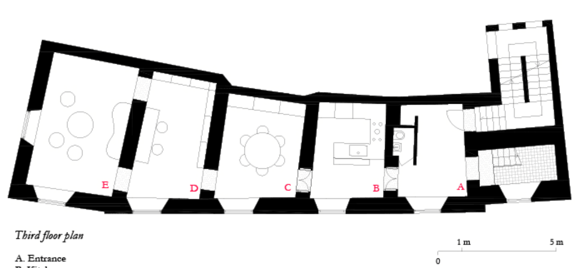






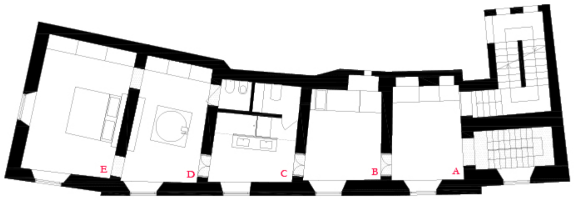
Sample	Room	Room Area	Total Floor Area	Room to Total Area	Centered Isovist	Centered Isovist to Room Floor	Centered Isovist to Total Floor	Number of visual neighbors	Function	Privacy	Degree Centrality	Betweennes s Centrality
1	Room 1	79.6	222.1	0.358	122	1.53	0.55	6	Living	0	4	37
1	Room 2	28.0	222.1	0.126	40.9	1.46	0.18	3	Bedroom	1	2	18
1	Room 3	14.7	222.1	0.066	21	1.43	0.09	2	Bathroom	1	2	10
1	Room 4	6.2	222.1	0.028	9.5	1.53	0.04	1	Closet	1	1	0
1	Room 5	8.7	222.1	0.039	9.8	1.13	0.04	1	Laundry	0.5	1	0
1	Room 6	5.4	222.1	0.024	21.4	3.96	0.10	1	Storage	0	1	0
1	Room 7	11.9	222.1	0.054	32.5	2.73	0.15	4	Distribution	0	5	39
1	Room 8	5.1	222.1	0.023	8	1.57	0.04	2	Dressing	0.5	2	10
1	Room 9	4.7	222.1	0.021	6.5	1.38	0.03	1	Bathroom	1	1	0
1	Room 10	15.3	222.1	0.069	19.9	1.30	0.09	2	Bedroom	1	1	0
1	Room 11	21.2	222.1	0.095	24.2	1.14	0.11	1	Bedroom	1	1	0
1	Room 12	21.3	222.1	0.096	25.3	1.19	0.11	2	Bedroom	1	1	0

14.2 Sample Plan n.2



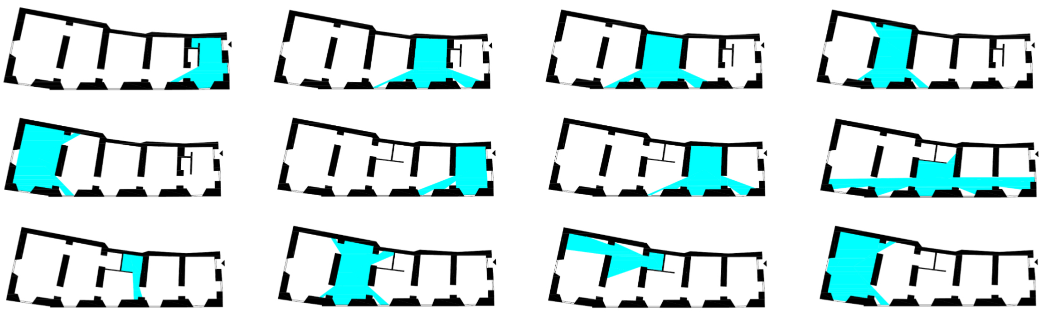
Third floor plan

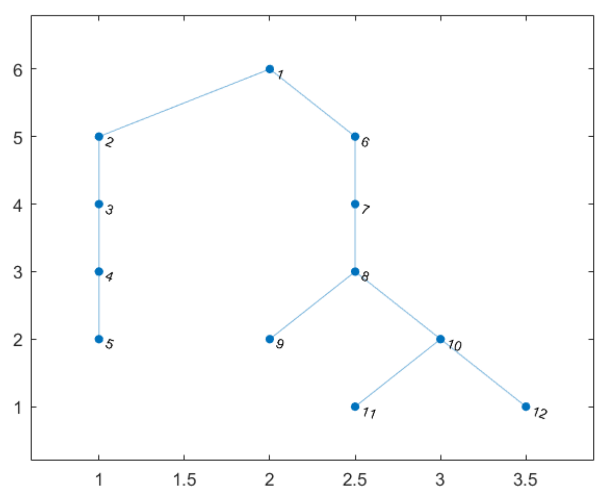
A. Entrance
B. Kitchen
C. Dining room
D. Study
E. Living room



Fourth floor plan

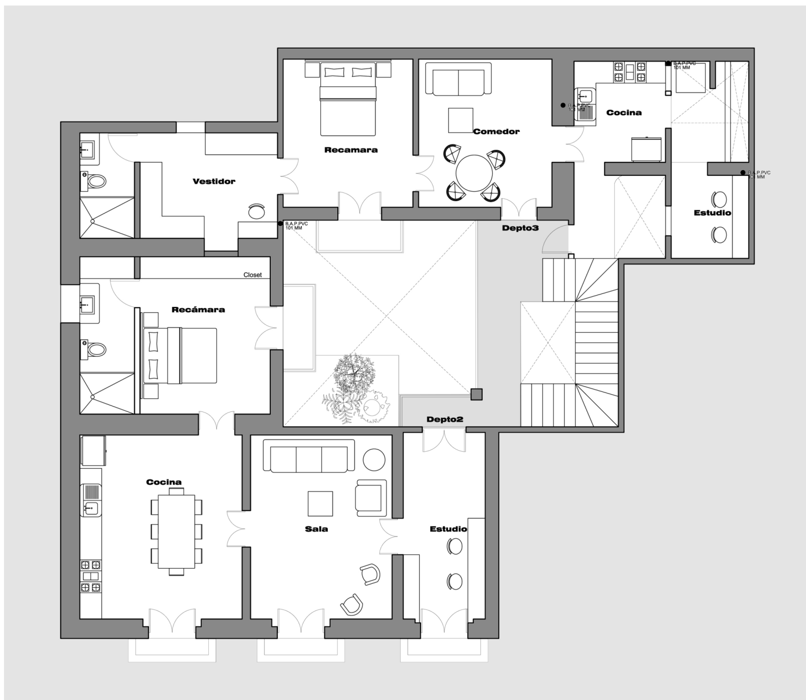
A. Play room
B. Bedroom
C. Bathroom
D. Boudoir
E. Master bedroom

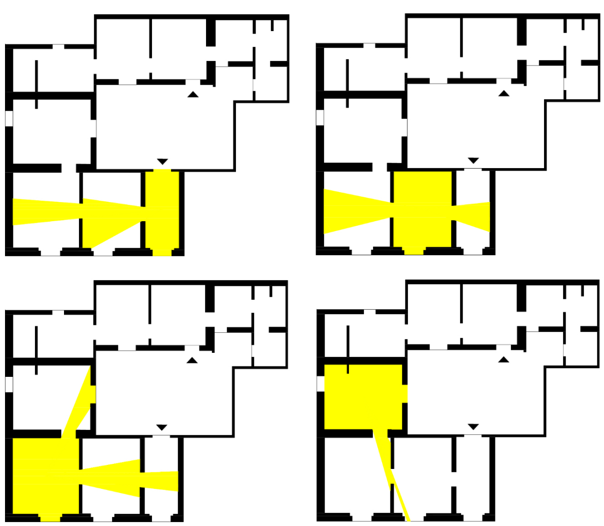


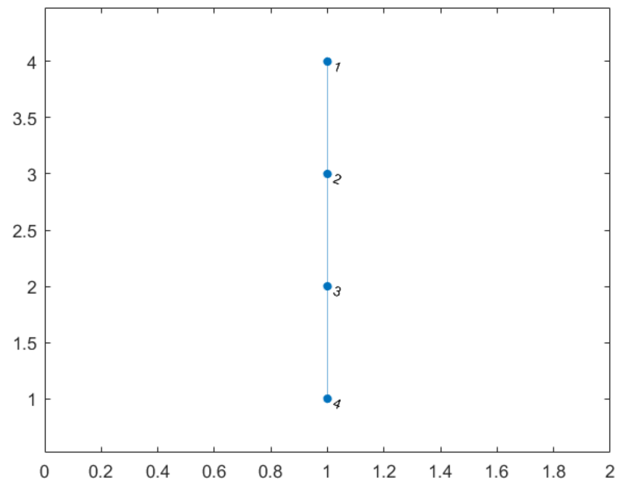


Sample	Room	Room Area	Total Floor Area	Room to Total Area	Centered Isovist	Centered Isovist to Room Floor	Centered Isovist to Total Floor	Number of visual neighbors	Function	Privacy	Degree Centrality	Betweennes s Centrality
1	Room 1	79.6	222.1	0.358	122	1.53	0.55	6	Living	0	4	37
1	Room 2	28.0	222.1	0.126	40.9	1.46	0.18	3	Bedroom	1	2	18
1	Room 3	14.7	222.1	0.066	21	1.43	0.09	2	Bathroom	1	2	10
1	Room 4	6.2	222.1	0.028	9.5	1.53	0.04	1	Closet	1	1	0
1	Room 5	8.7	222.1	0.039	9.8	1.13	0.04	1	Laundry	0.5	1	0
1	Room 6	5.4	222.1	0.024	21.4	3.96	0.10	1	Storage	0	1	0
1	Room 7	11.9	222.1	0.054	32.5	2.73	0.15	4	Distribution	0	5	39
1	Room 8	5.1	222.1	0.023	8	1.57	0.04	2	Dressing	0.5	2	10
1	Room 9	4.7	222.1	0.021	6.5	1.38	0.03	1	Bathroom	1	1	0
1	Room 10	15.3	222.1	0.069	19.9	1.30	0.09	2	Bedroom	1	1	0
1	Room 11	21.2	222.1	0.095	24.2	1.14	0.11	1	Bedroom	1	1	0
1	Room 12	21.3	222.1	0.096	25.3	1.19	0.11	2	Bedroom	1	1	0

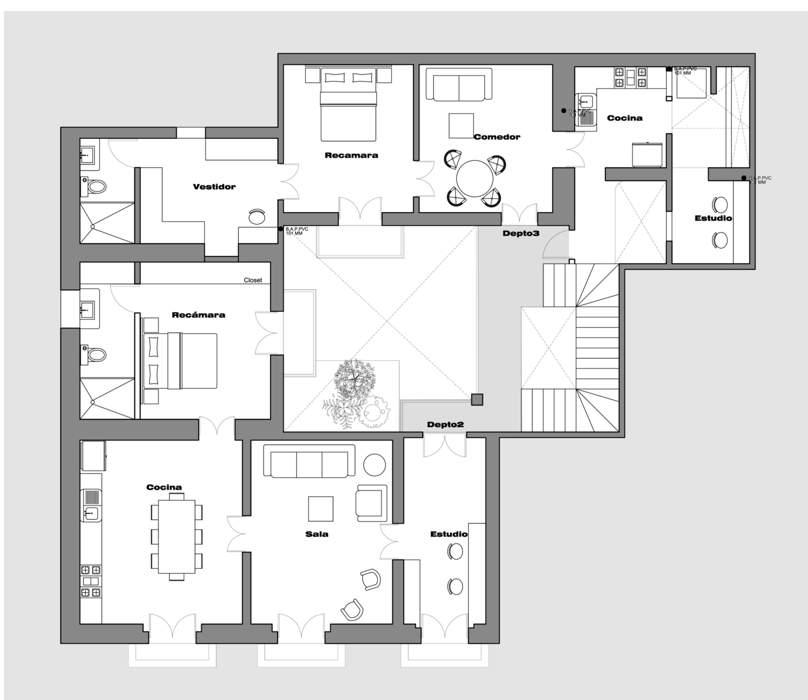
14.3 Sample Plan n.3

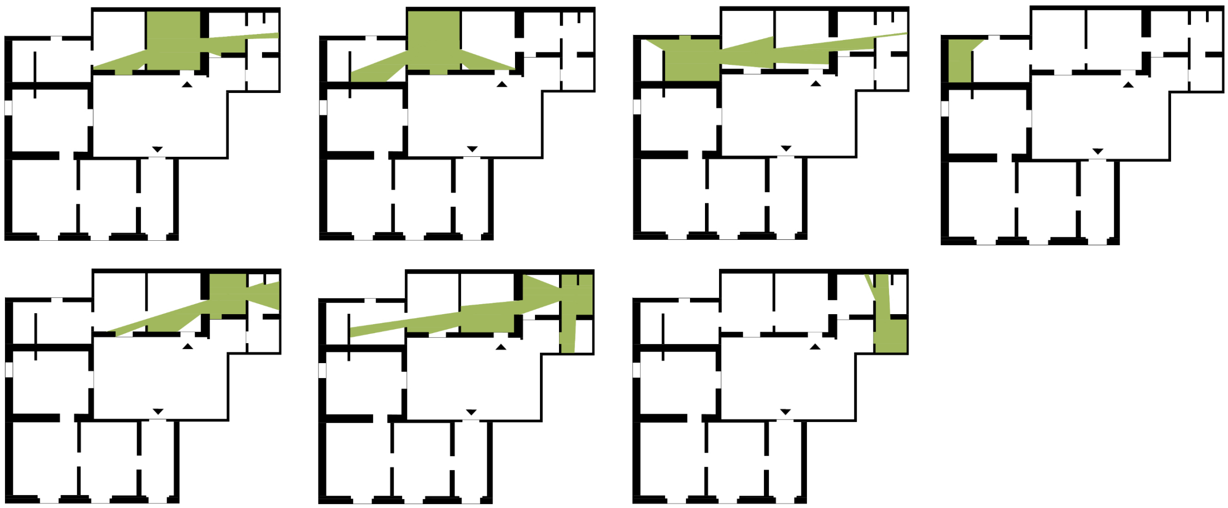


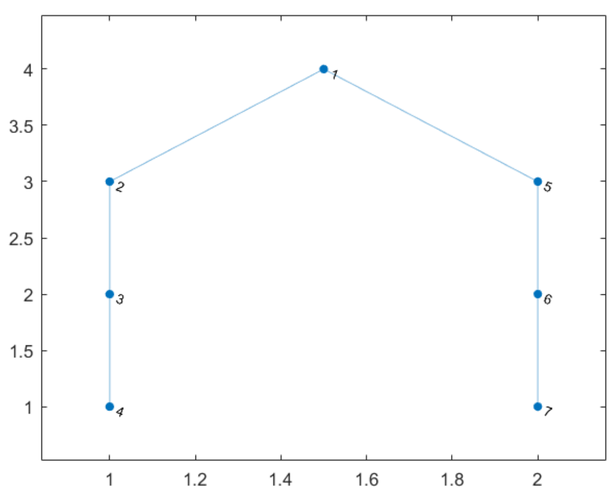




14.4 Sample Plan n.4

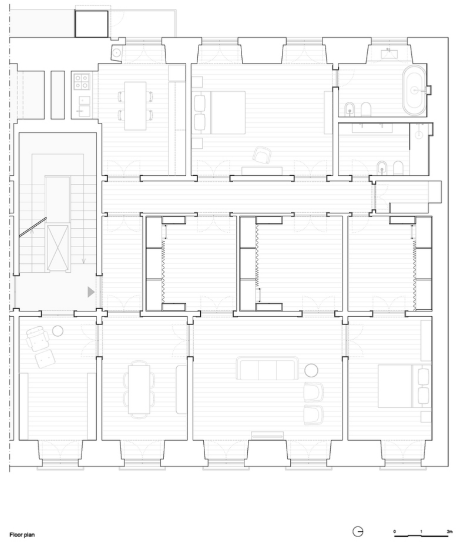


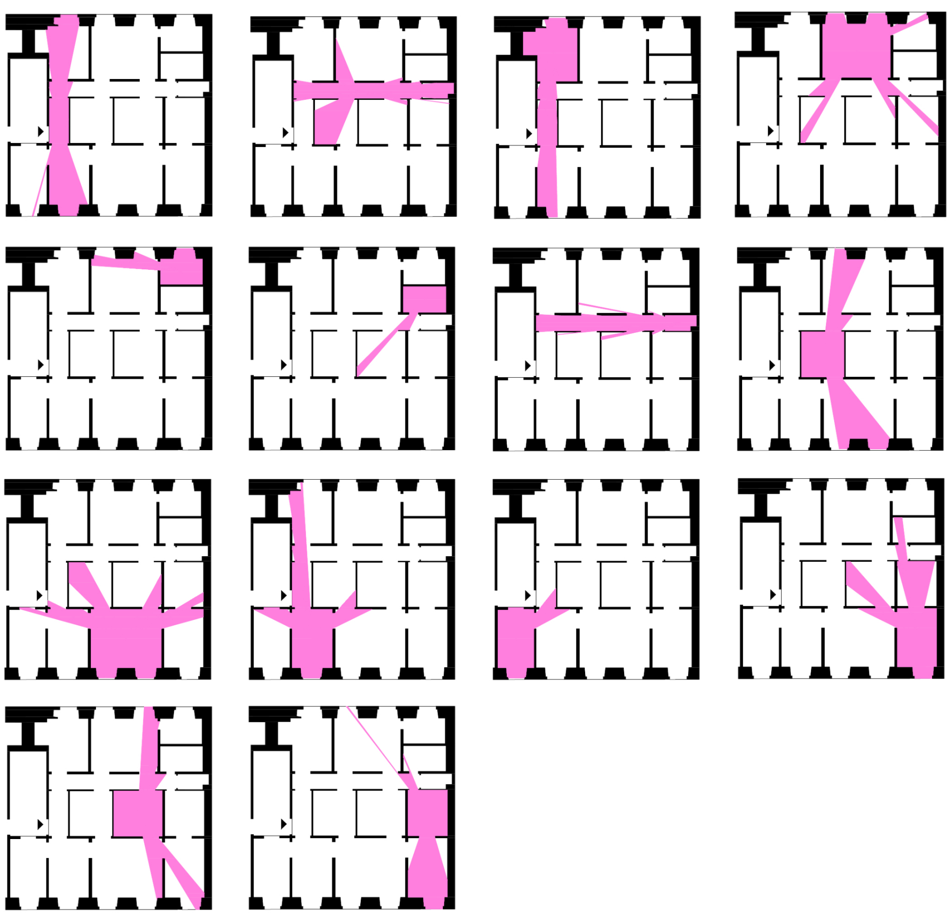


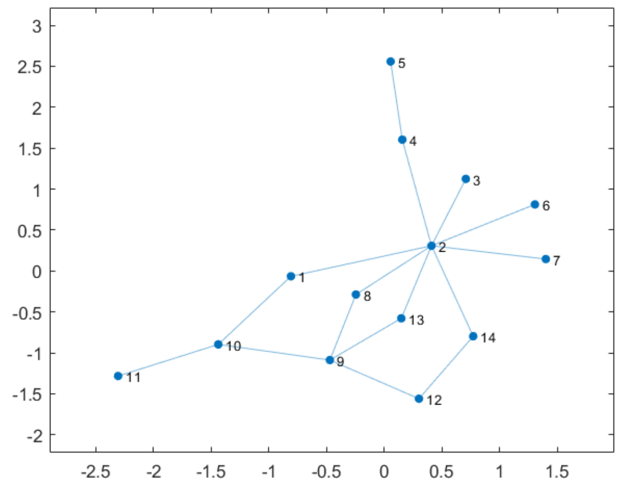


4
3.5
3
2.5
2
1.5
1
1
1.2
1.4
1.6
1.8
2
1
2
3
4
5
6
7

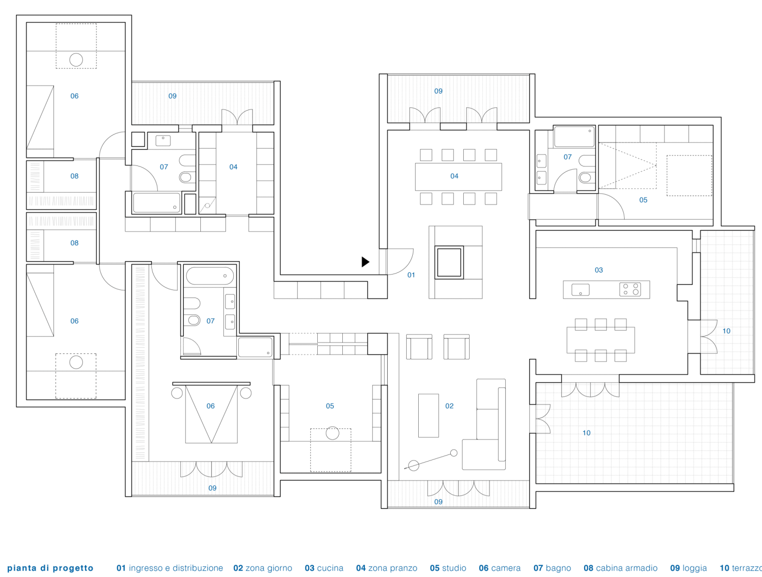
14.5 Sample Plan n.5

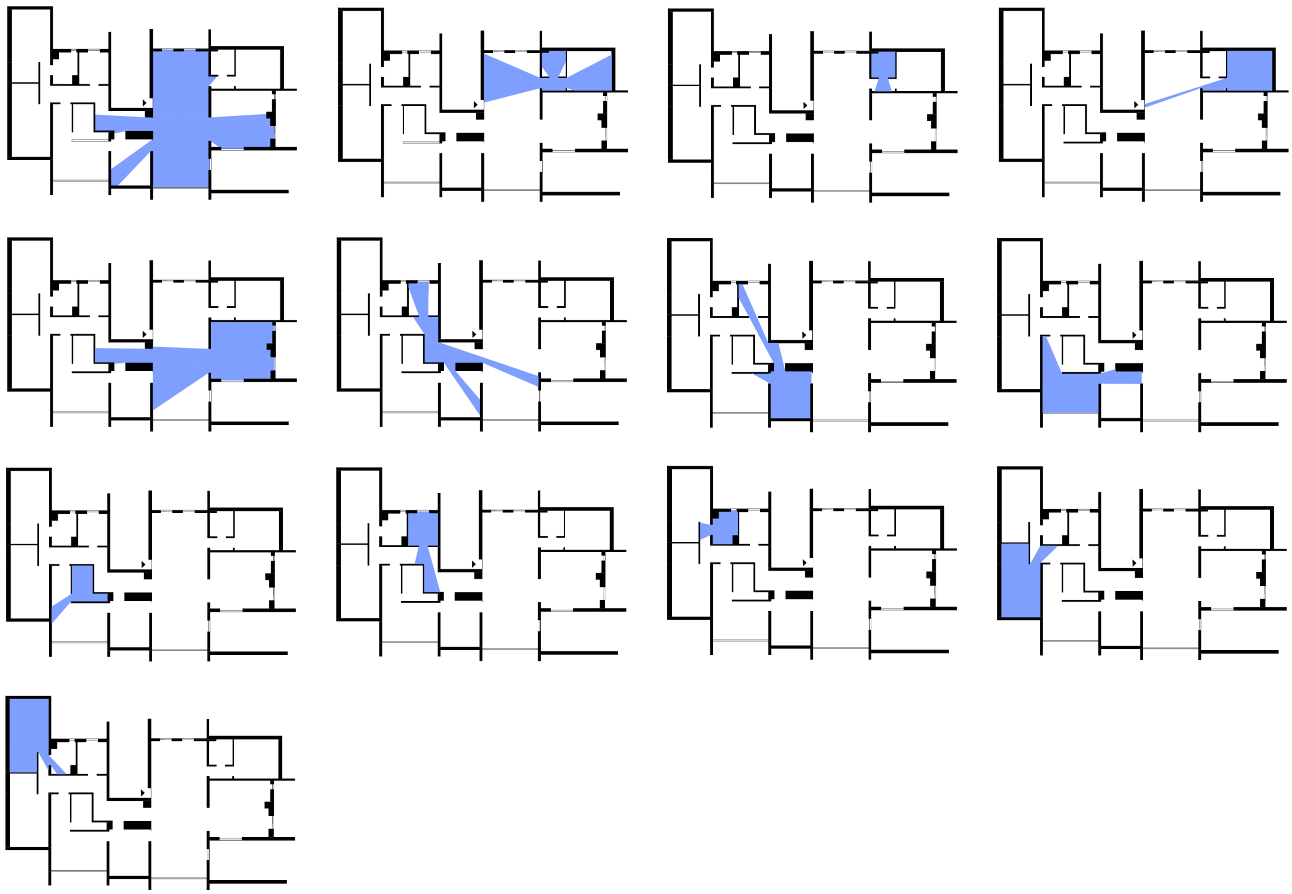


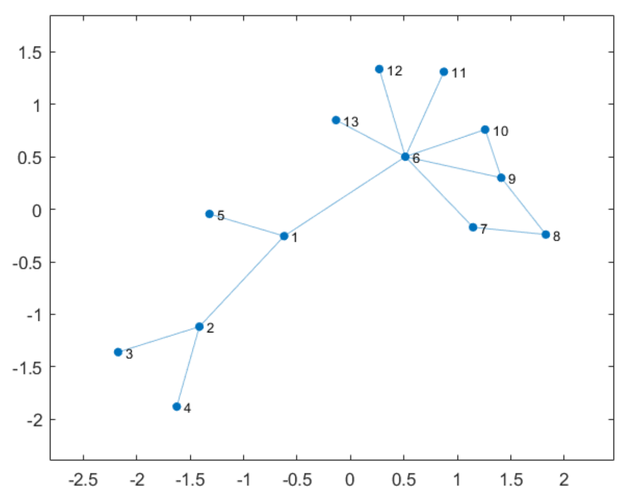




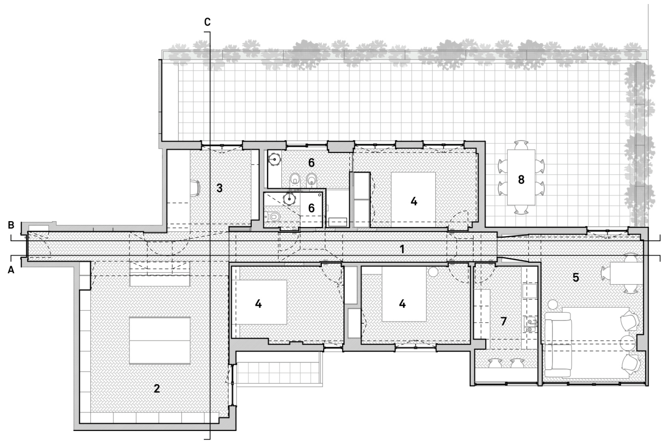
14.6 Sample Plan n.6



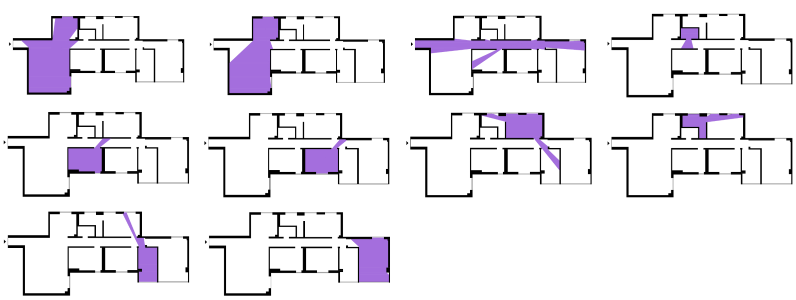


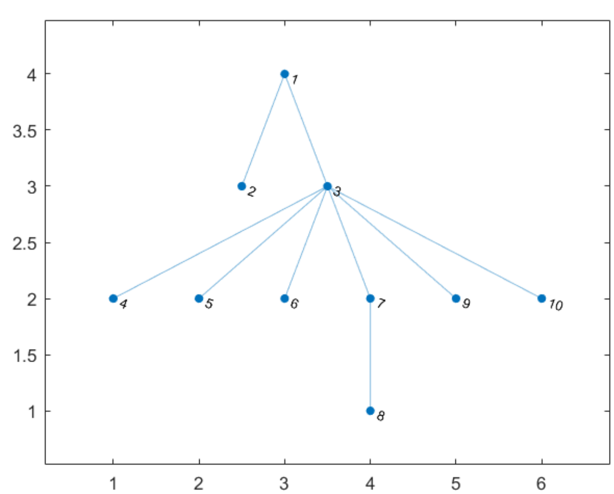

1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
1
2
3
4
5
6
7
8
9
10
11
12
13

14.7 Sample Plan n.7

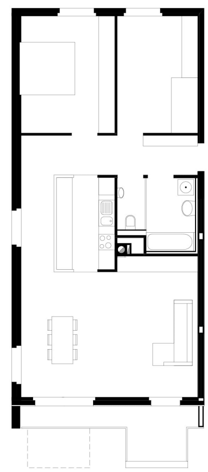


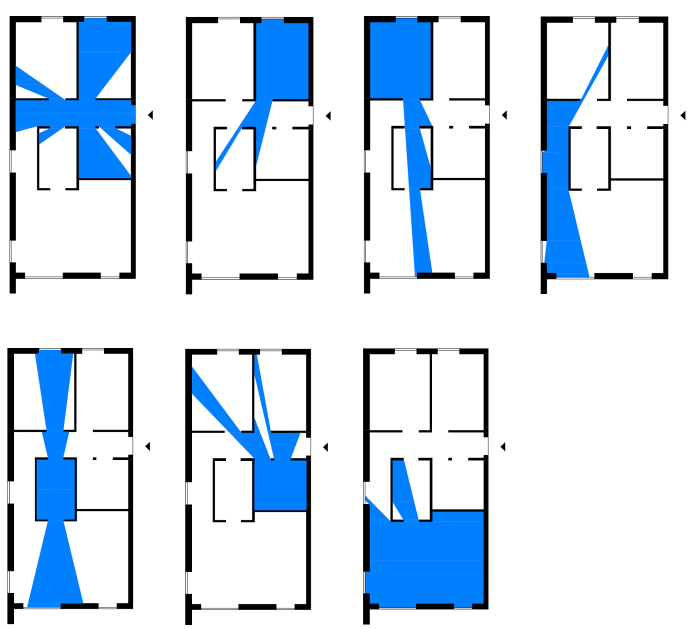
PLAN
1 corridor **2** library **3** office **4** bedroom **5** living-room
6 bathroom **7** kitchen **8** terrace **9** balcony

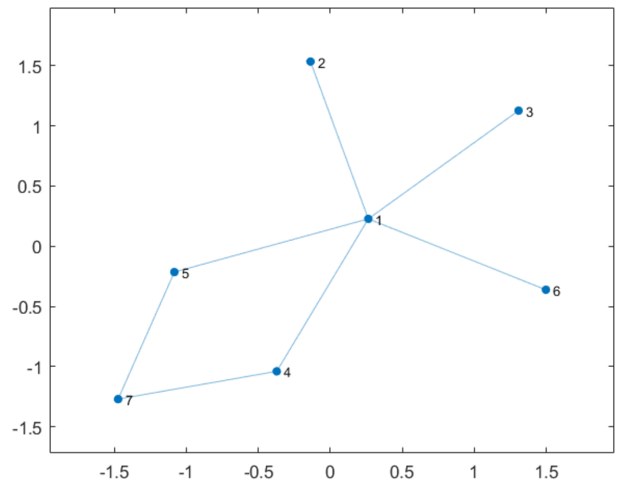




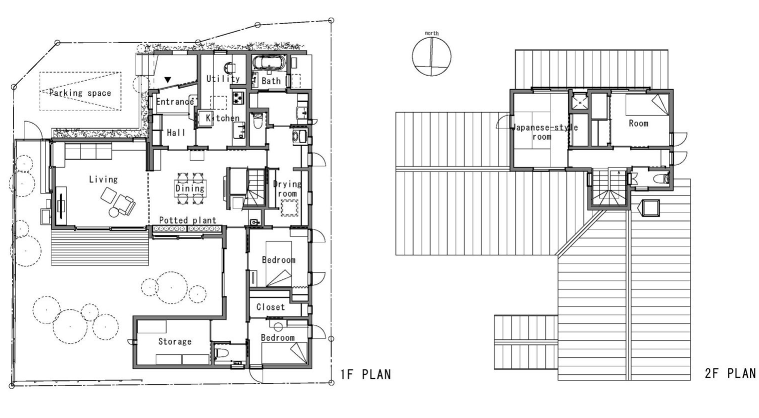
14.8 Sample Plan n.8

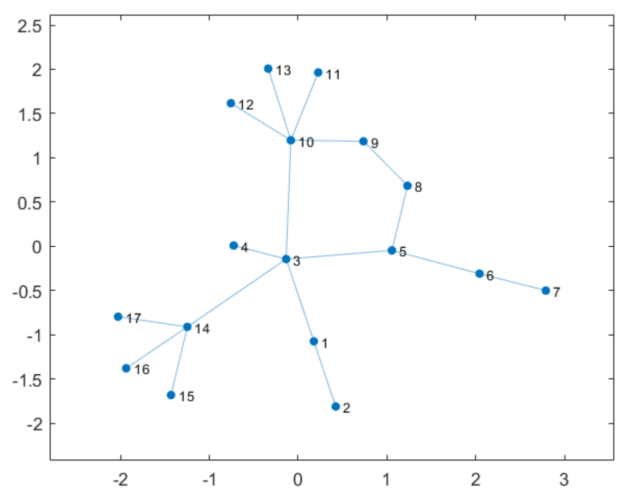






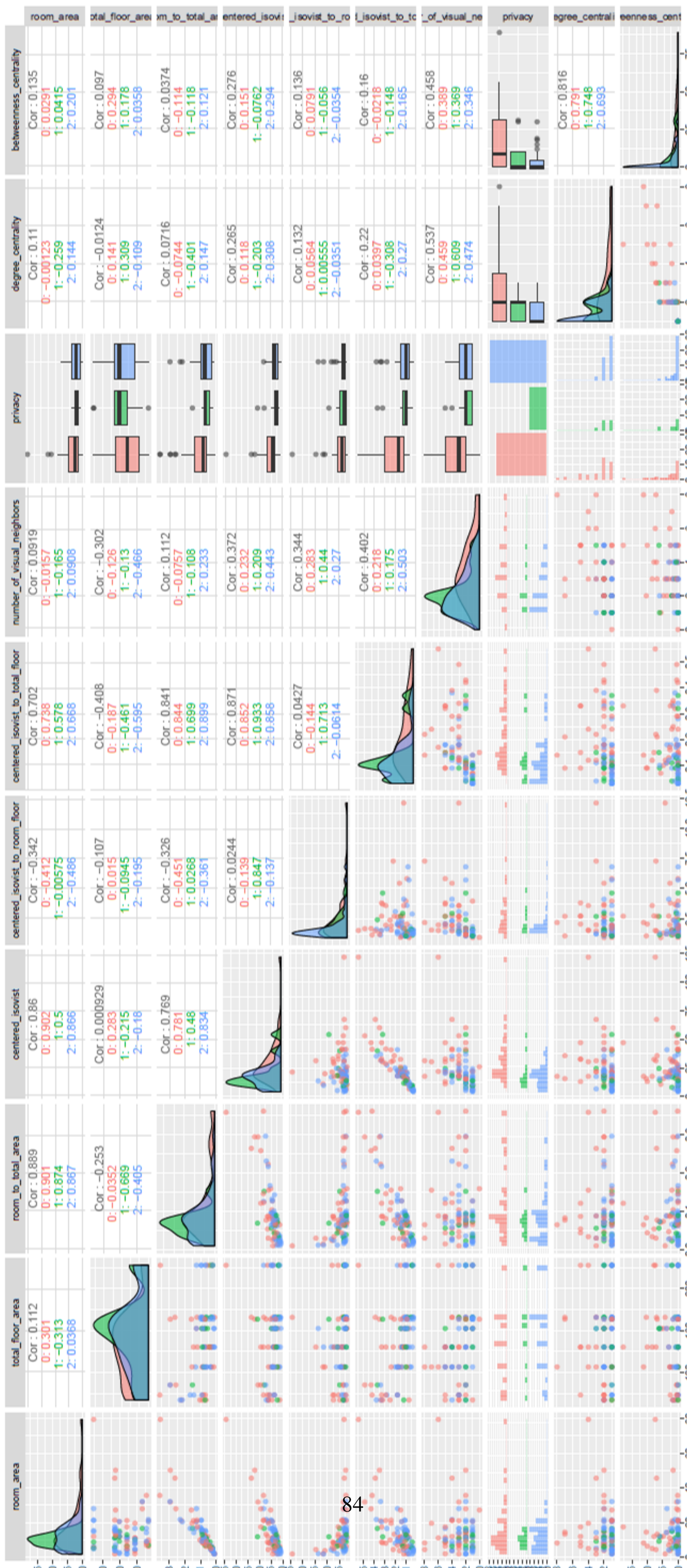
14.9 Sample Plan n.9




1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1
0
1
2
3

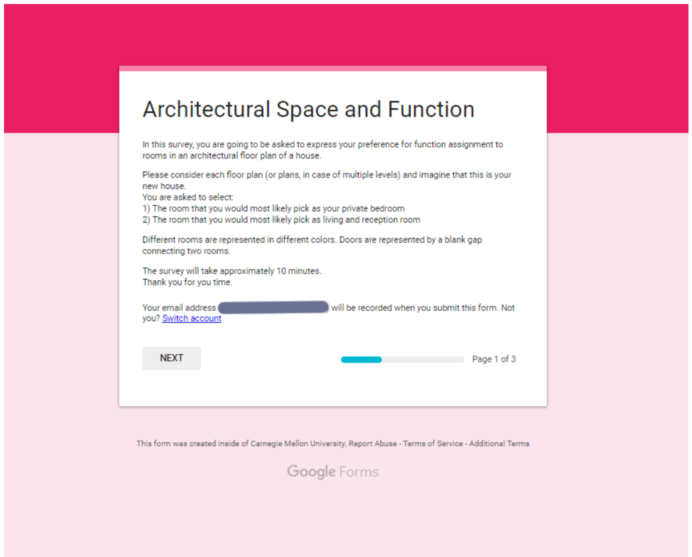
15 Appendix: results

Figure 53: Pairplots



room_area
total_floor_area
room_to_total_area
centered_isovist
centered_isovist_to_room_floor
centered_isovist_to_total_floor
number_of_visual_neighbors
privacy
degree_centrality
betweenness_centrality

16 Appendix: survey



Architectural Space and Function

In this survey, you are going to be asked to express your preference for function assignment to rooms in an architectural floor plan of a house.

Please consider each floor plan (or plans, in case of multiple levels) and imagine that this is your new house.
You are asked to select:
1) The room that you would most likely pick as your private bedroom
2) The room that you would most likely pick as living and reception room

Different rooms are represented in different colors. Doors are represented by a blank gap connecting two rooms.

The survey will take approximately 10 minutes.
Thank you for you time.

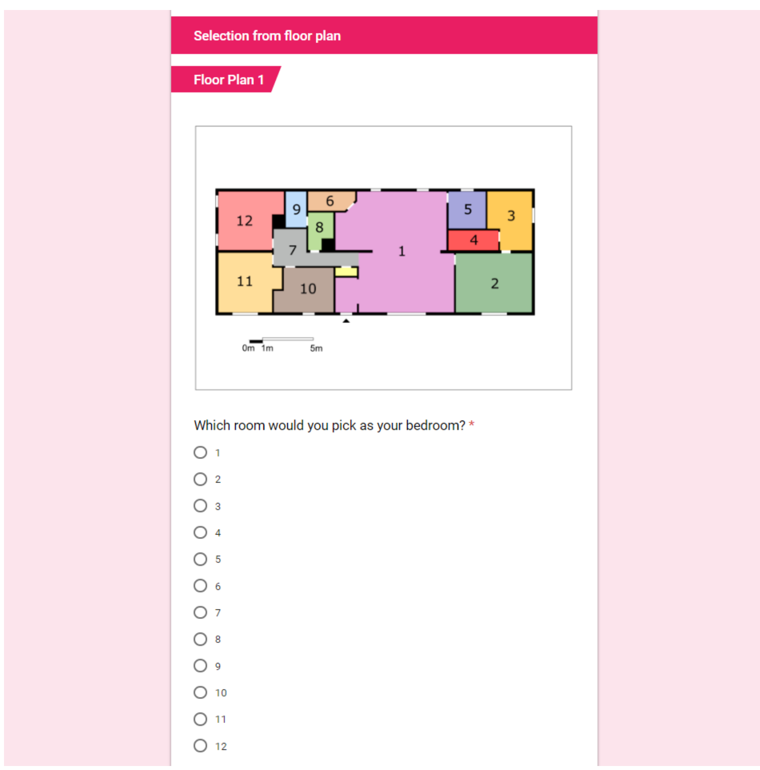
Your email address [redacted] will be recorded when you submit this form. Not you? Switch account

NEXT

Page 1 of 3

This form was created inside of Carnegie Mellon University. Report Abuse - Terms of Service - Additional Terms

Google Forms

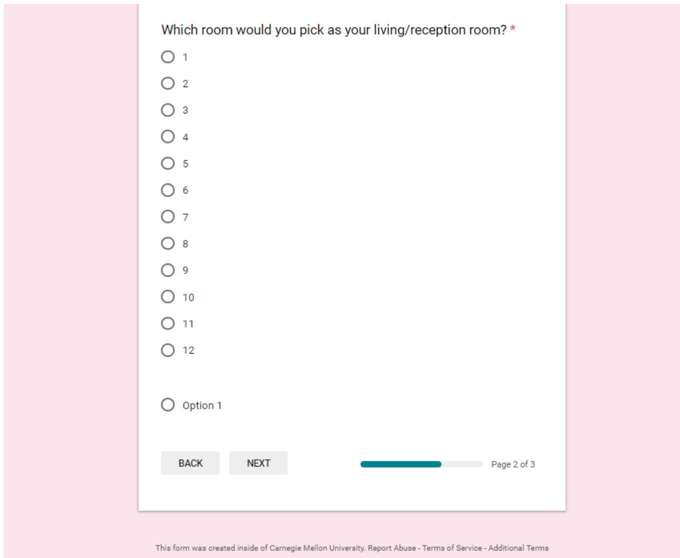


Selection from floor plan

Floor Plan 1

Which room would you pick as your bedroom? *

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12



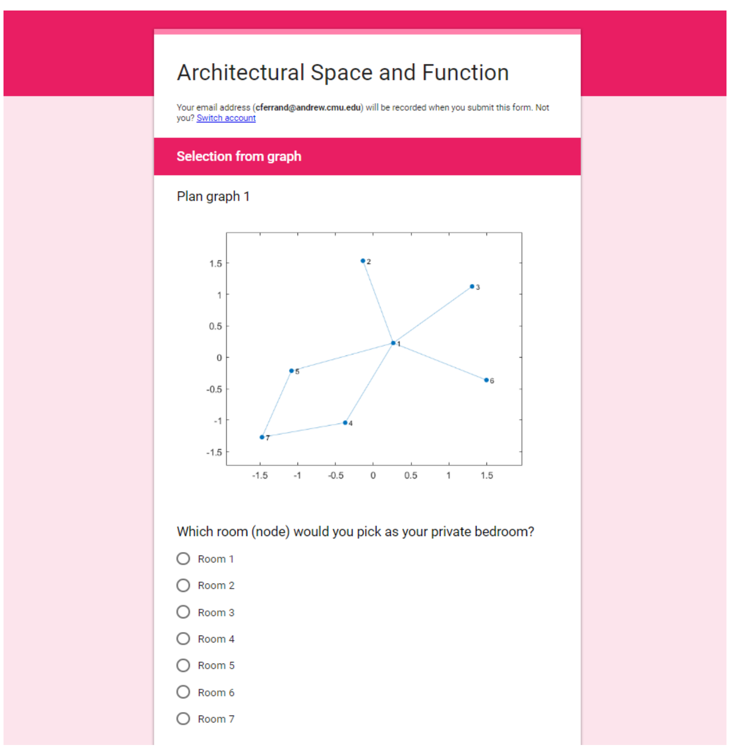
Which room would you pick as your living/reception room? *

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

- Option 1

BACK NEXT Page 2 of 3

This form was created inside of Carnegie Mellon University. Report Abuse - Terms of Service - Additional Terms



Architectural Space and Function

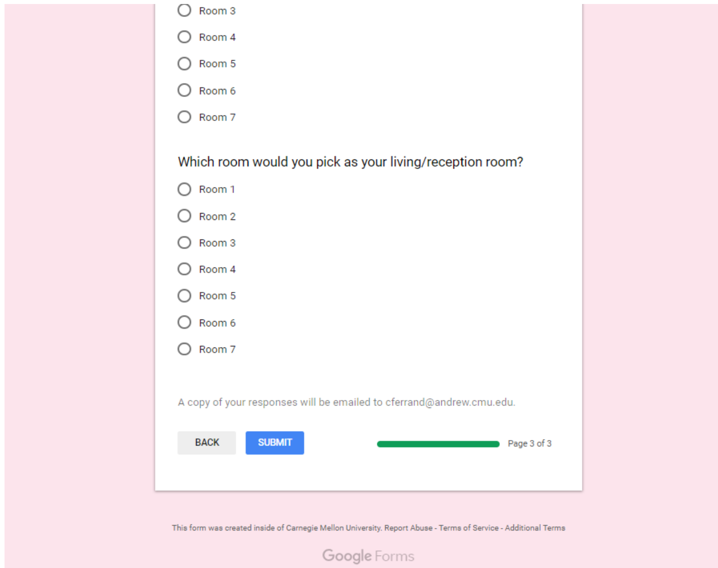
Your email address (**cferrand@andrew.cmu.edu**) will be recorded when you submit this form. Not you? Switch account

Selection from graph

Plan graph 1

Which room (node) would you pick as your private bedroom?

- Room 1
- Room 2
- Room 3
- Room 4
- Room 5
- Room 6
- Room 7



- Room 3
- Room 4
- Room 5
- Room 6
- Room 7

Which room would you pick as your living/reception room?

- Room 1
- Room 2
- Room 3
- Room 4
- Room 5
- Room 6
- Room 7

A copy of your responses will be emailed to cferrand@andrew.cmu.edu.

BACK SUBMIT Page 3 of 3

This form was created inside of Carnegie Mellon University. Report Abuse - Terms of Service - Additional Terms

Google Forms

References

[1] Christopher Alexander. *A pattern language: towns, buildings, construction*. Oxford university press, 1977.

[2] Bill Hillier and Julienne Hanson. *The social logic of space*. Cambridge university press, 1989.

[3] Michael J Ostwald. The mathematics of spatial configuration: revisiting, revising and critiquing justified plan graph theory. *Nexus Network Journal*, 13(2):445–470, 2011.

[4] Christopher Alexander. *Notes on the Synthesis of Form*, volume 5. Harvard University Press, 1964.

[5] Christopher Alexander. A city is not a tree. *1965*, 1964.

[6] Bill Hillier and Kali Tzortzi. Space syntax: the language of museum space. *A companion to museum studies*, pages 282–301, 2006.

[7] Bill Hillier, Julienne Hanson, and Hillaire Graham. Ideas are in things: an application of the space syntax method to discovering house genotypes. *Environment and Planning B: planning and design*, 14(4):363–385, 1987.

[8] Julienne Hanson. *Decoding homes and houses*. Cambridge University Press, 2003.

[9] BUILT SPACE. Spatial structure of. *Handbook of environmental psychology*, page 271, 2003.

[10] Larry S Davis and Michael L Benedikt. Computational models of space: Isovists and isovist fields. *Computer graphics and image processing*, 11(1):49–72, 1979.

[11] Sophia Psarra. *Architecture and Narrative: The formation of space and cultural meaning*. Routledge, 2009.

[12] Alasdair Turner, Maria Doxa, David O'sullivan, and Alan Penn. From isovists to visibility graphs: a methodology for the analysis of architectural space. *Environment and Planning B: Planning and design*, 28(1):103–121, 2001.

[13] Wendy Bustard. Space, evolution, and function in the houses of chaco canyon. *Environment and Planning B: Planning and design*, 26(2):219–240, 1999.

[14] Michael Georgiou. *Architectural privacy: A topological approach to relational design problems*. PhD thesis, UCL (University College London), 2006.

[15] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

[16] Serge Chermayeff and Christopher Alexander. *Community and privacy: Toward a new architecture of humanism*. Doubleday, 1963.

[17] George Stiny and J Gips. An evaluation of palladian plans. *Environment and planning B: Planning and design*, 5(2):199–206, 1978.

[18] Samuel Dodge, Jiu Xu, and Björn Stenger. Parsing floor plan images. In *Machine Vision Applications (MVA), 2017 Fifteenth IAPR International Conference on*, pages 358–361. IEEE, 2017.

[19] Sébastien Macé, Hervé Locteau, Ernest Valveny, and Salvatore Tabbone. A system to detect rooms in architectural floor plan images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 167–174. ACM, 2010.

[20] Qamer Uddin Sabri, Johannes Bayer, Viktor Ayzenshtadt, Syed Saqib Bukhari, Klaus-Dieter Althoff, and Andreas Dengel. Semantic pattern-based retrieval of architectural floor plans with case-based and graph-based searching techniques and their evaluation and visualization. In *ICPRAM*, pages 50–60, 2017.

[21] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1605.06211, 2016.

[22] Lluis-Pere de las Heras, OriolRamos Terrades, Sergi Robles, and Gemma Sanchez. Cvc-fp and sgt: a new database for structural floor plan analysis and its groundtruthing tool. *International Journal on Document Analysis and Recognition*, 2015.

[23] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE transactions on visualization and computer graphics*, 17(11):1624–1636, 2011.

[24] Zhaoliang Lun, Matheus Gadelha, Evangelos Kalogerakis, Subhransu Maji, and Rui Wang. 3d shape reconstruction from sketches via multi-view convolutional networks. *arXiv preprint arXiv:1707.06375*, 2017.

[25] Gedas Bertasius, Lorenzo Torresani, Stella X Yu, and Jianbo Shi. Convolutional random walk networks for semantic image segmentation. *arXiv preprint arXiv:1605.07681*, 2016.

[26] Paul Martin. A spatial analysis of the jba headquarters in splinter cell: Double agent. *Entertainment Computing*, 5(1):71–82, 2014.

[27] Lluís-Pere de las Heras, David Fernández, Ernest Valveny, Josep Lladós, and Gemma Sánchez. Unsupervised wall detector in architectural floor plans. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1245–1249. IEEE, 2013.

[28] Colin Day, Anthony Hauck, John Haymaker, and Diana Davis. Space plan generator.

[29] Eugénio Rodrigues, David Sousa-Rodrigues, Mafalda Teixeira de Sampayo, Adélio Rodrigues Gaspar, Álvaro Gomes, and Carlos Henggeler Antunes. Clustering of architectural floor plans: A comparison of shape representations. *Automation in Construction*, 80:48–65, 2017.

[30] Alexander Koutamanis and Vicky Mitossi. Computer vision in architectural design. *Design Studies*, 14(1):40–57, 1993.

[31] Nicolas Le Camus De Mezieres. *The Genius of Architecture: Or, the Analogy of that Art with Our Sensations*. Getty Publications, 1992.