Learning Indoor Space Perception

Andreas Sedlmeier and Sebastian Feld

Mobile and Distributed Systems Group, LMU Munich, Munich, Germany

ARTICLE HISTORY
Compiled October 15, 2018

ABSTRACT
Human perception of location and space forms the basis upon which the interaction with location-based services (LBS) takes place. As most of these LBS services are ultimately built for humans, a shared awareness and common understanding of location and space, between machines, systems and their users, will lead to better services. Already, mobile robots are able to perform various tasks in buildings using a wide array of sensors to perceive their surroundings. A connected area of research which forms the basis for a deeper understanding of perception is the numerical representation of visual perception of space. Different structures in buildings like rooms, hallways and doorways form different, corresponding patterns in these representations. Thanks to recent advances in the field of deep learning with neural networks, it now seems possible to explore the idea of automatically learning these recurring structures using machine learning techniques. We presents a complete framework, starting from the collection of isovist measures along geospatial trajectories on indoor floor plans, over the statistical analysis of the data, the extraction of meaningful structure using unsupervised machine learning techniques, up to the training of models using supervised machine learning that generalize to previously unseen environments. We show that the isovist measures do reflect the recurring structures found in different buildings, that these recurring patterns are encoded in the data in a way that unsupervised machine learning is able to identify them and that the identified structures are meaningful as they represent human relatable concepts like rooms, hallways and doorways. Furthermore, we propose to use cluster similarity analysis as a promising concept for quantifying visual perception similarity.

KEYWORDS
visual perception; recurring structures; semantic annotation; machine learning; isovist measures

1. Introduction

Location-based services form a very interdisciplinary field of research ranging from Electrical Engineering over Computer Science to Social Science. Technological progress, especially the increase in computational power and the miniaturization of electronic devices and sensors, enabled the ideas of *Ubiquitous Computing* (Weiser 1991) and *Context Awareness* (Dey and Abowd 1999; Chen and Kotz 2000), both of which lead to the integration of location-based services into the daily life of many people. Built on this, mobile devices like smartphones or wearables contain several sensors for measuring movement (accelerometer), brightness (camera), volume (microphone),

CONTACT Andreas Sedlmeier. Email: andreas.sedlmeier@ifi.lmu.de

air pressure (barometer), position (GPS), and others. Thus, location-based services are basically context-aware services that incorporate spatial information (Küpper 2005).

Mobile robots can also be regarded to represent location-based services. Equipped with sensors like laser scanners, optical cameras, or tactile sensors they perceive and process their environment, resulting in the execution of tasks like the transportation of packets in warehouses. Further examples of research are mobile robots that lead tourist groups through an airport (Triebel et al. 2016) or serve as an assistance in housekeeping and everyday tasks (Rashidi and Mihailidis 2013). Due to recent advances in the fields of big data and machine learning, mobile robots get increasingly autonomous. Recent research focuses on cooperation, competition, and communication in order to solve more complicated tasks (Lowe et al. 2017; Mordatch and Abbeel 2017).

A related field of study deals with the visual perception of space. Since the end of the 1960s there were numerous empirical and experimental studies conducted on the perception of architectural space. An early example is the work of Hayward and Franklin, who analyzed the influence of bordering elements like walls or trees on the perception of openness (Hayward and Franklin 1974). Today, there are different theories and tools to analyze spatial arrangements (De Smith, Goodchild, and Longley 2007) often summarized under the term *Space Syntax*. The focus of this field is mostly on the analysis of topological structures of an environment without geometric measurements (Hillier and Hanson 1984). *Isovists*, introduced by Tandy (1967) describe the set of points in space that are visible from a specific vantage point and can be used for visibility analysis. Based on this idea a formal definition of isovists was developed, together with some analytical measures enabling a quantitative description of a spatial environment (Benedikt 1979).

Although we employ Isovists as a tool from the field of Space Syntax, we use only the "raw" measures that form the Isovist and do not focus on topology or higher-level aspects which can be based on Isovists like integration, navigation or the prediction of human activity. Instead, we are interested in the visual perception generated by buildings as spatial structures and to find general similarity or differentiation in and among them.

Even though every building is different, one can still observe structures that recur constantly. Examples for such structural recurrences together with some semantics are rooms (small enclosed areas, often rectangular), corridors (long areas connecting rooms), or doorways (gaps in walls connecting rooms and corridors). Further exemplary structures are halls, staircases, or patios. The interesting part of such structures is that every room, corridor, and the like looks different, but they contain similarities that enable a (not necessarily distinct) recognition. Interestingly, this is a general problem area in which huge progress was made in the last few years thanks to the advances in the field of deep learning with artificial neural networks. Deep neural networks excell at the recognition of recurring structures in large data sets and the discovery of underlying functions generating these structures. Feld et al. (2018) for example just recently showed that an unsupervised learning approach based on a neural autoencoding is able to learn semantically meaningful continuous encodings of spatio-temporal trajectory data. This learned encoding can then be used to generate prototypical representations of the visual perception of different kinds of movement.

Sedlmeier and Feld (2018) have shown that recurring structures inside buildings also have recurring isovist measures and that the numerical features can be used to learn models of these structures. In this work, we extend and build upon these findings by investigating whether the presence or absence of the identified structures can be used to compare different environments. In section 4.3 and 4.4 we show that this can be done visually by comparing map- and feature-space visualizations, enabling a qualitative judgement of the similarity. In section 4.5 we present first evidence that besides the qualitative comparison, it is also possible to use cluster similarity measures based on the learned structures, to quantify the visual perception difference between environments. To the best of our knowledge, such a numerical quantification of perceptual floor plan similarity based on unsupervised learning has not been shown before. We regard this numerical quantification as an important advance from the strictly qualitative evaluations performed in previous work.

The general use case of these ideas is to create advanced spatial context for location-based services. A concrete use case would be the problem of *Simultaneous Localization and Mapping* (SLAM), where a mobile robot has to build a map of its environment and estimate its pose simultaneously (Leonard and Durrant-Whyte 1991). Using the ideas presented in this paper, a mobile robot would be able to autonomously learn a model of recurring structures inside buildings, like rooms, hallways, and doorways. This model could then be reused in unknown buildings to recognize and label the learned structures straight away. By further mapping these conceptual labels to natural language, i.e. the words *room* or *door*, this perceptional awareness would allow the robot to communicate it's intentions and plans more naturally with humans, as well as understand tasks containing these spacial concepts. By making use of the previously mentioned similarity measures, the robot would also be able to judge the similarity or difference of the environments. Alternative use cases are the off-line analysis or annotation of floor plans or the incorporation of our findings in computer games, such that non-player characters gain an additional awareness of altering surroundings.

This paper builds upon the work done by Sedlmeier and Feld (2018), extends the evaluation of the framework by including a larger amount of floor plans and presents new ideas enabling the evaluation of similarity or dissimilarity between floor plans. Together with the extensions presented in this paper, the contributions are as follows:

First, we present a framework that is able to automatically generate data for learning models of recurring structures inside buildings based on floor plans as input. The framework builds on the game engine *Unity* developed by Unity Technologies (Unity Technologies 2017) and uses the included navigation and route finding procedures to create a set of geospatial trajectories. Furthermore, our framework contains a custom isovist implementation in C# that is able to calculate isovist measurements for each time step of the trajectories in the data set. Second, we present a framework which combines several machine learning techniques that can be used to extract and visualize, as well as analytically analyze the identified recurring structures. Third, we show how supervised machine learning can be used for the training of models that recognize such structures in unknown floor plans. Fourth, we we show that being able to learn and recognize structures of building floor plans, also makes it possible to judge the similarity or dissimilarity of the floor plans. This can be done qualitatively using map- and PCA feature-space visualizations or quantitatively via cluster similarity analysis. First evaluations show that the results match our human judgement of the floorplans' similarities.

All of the before mentioned parts are released under an open-source licence and made available at *https://github.com/sedand/isovista*. We build upon existing scientific computing libraries written in the Python programming language (Pedregosa et al. 2011; Jones et al. 2001–) as well as the open source neural network library Keras (Chollet et al. 2015), which in turn uses TensorFlow (Abadi et al. 2015), a low level machine learning library developed by Google.

The remainder of this article is structured as follows: Section 2 describes the technical background for the further understanding of this paper together with related work. Section 3 incorporates the methodology for generating isovist measures along geospatial trajectories as well as the discovery, learning, prediction, visualization and comparison of recurring structures in floor plans. In Section 4 we present our experimental results and present a detailed discussion. Section 5 concludes this paper.

2. Background and Related Work

This section presents some technical background to enable a deeper comprehension of this paper together with related work. First, techniques for the analysis of visual perception are described, which allow for a quantitative description of spatial environments. Next, several data analysis and machine learning techniques are presented which form the basis of the automatic discovery of recurring structures inside buildings as well as the learning of models. Furthermore, related work with respect to the semantic annotation of floor plans are illustrated.

2.1. *Analysis of Visual Perception*

As already mentioned in Section 1, we utilize techniques that analyze the visual perception of space. There are numerous studies in the sector of cognitive psychology that address the behavior of people in typical buildings like hospitals (Haq and Zimring 2003), malls (Dogu and Erkip 2000), or airports (Raubal 2002).

Isovist Analysis is a concrete technique of *Space Syntax* that is used in many cases. As originally introduced by Tandy (1967) and more formally defined by Benedikt (1979), an isovist is the set of points in space that is visible from a specific vantage point.

The six isovist measures as defined by Benedikt (1979) are as follows:

(1) A_x: the **area** describes the surface area of the isovist. The higher the value, the more space is visible from the vantage point. At the same time, this means that the vantage point can be observed from a large space.
(2) P_x: the **real-surface perimeter** describes the length of the isovist's circumference that lies on visible obstacle surfaces, for example walls.
(3) Q_x: the **occlusivity** describes the length of the isovist's circumference that lies in free space. With other words, these are the concealed radial borderlines that can be imagined as rays passing an obstacle and traversing through free space.
(4) $M_{2,x}$: since the set of points in space that is visible from a specific vantage point can be calculated using rays sent out radially from the vantage point (Benedikt 1979), the **variance** is the second central moment of the rays' length.
(5) $M_{3,x}$: the **skewness** is the third central moment of the rays' length.
(6) N_x: the **circularity** is an isoperimetric quotient and evaluates the area against the perimeter. Basically, this is a numerical value that describes how similar a figure is in comparison to a circle. Circularity is calculated using $N_x = |\partial V_x|^2 / 4\pi A_x$, with $|\partial V_x|$ indicating the isovist's perimeter.

Isovist fields are likewise described by Benedikt (1979) as the set of isovists along a trajectory, or more complete, the set of isovists at all places of an investigated environment. As humans are moving through an environment in a continuous manner, the isovist measures are also changing continuously. Thus, one can observe gradual changes in the isovist measures. This is the underlying idea of our approach: we generate geospatial trajectories through floor plans, calculate isovist measures at every time step and analyze both, the absolute as well as the delta values to previous steps.

Although the framework proposed in this paper uses a 3D environment, the agent navigating through the building only walks on a 2D plane and thus creates 2D trajectories and corresponding 2D isovist measures. Nevertheless, there is literature that analyzes isovists in 3D space (Emo 2015).

The calculation of an isovist or rather the calculation of the isovist measures, as described above, is constrained by the environment's geometry and can get potentially complicated, since all corner points of visible walls and objects have to be determined and connected. Feld, Werner, and Linnhoff-Popien (2016) showed that at least *area*, *variance*, *skewness*, and *circularity* can easily be approximated using a simple ray-scan algorithm. The authors' motivation was to receive a preferably simple equivalent of isovist measures that can be applied on floor plans represented as occupancy grids via bitmaps. White pixels stood for walkable free space, black pixels represented obstacles like walls or other objects. Their experiments showed that there is a systematic error regarding the approximated and exact isovist measures, however, they show a strong correlation.

The ideas and solutions presented in this paper are using a similar ray-based approach.

2.2. *Machine Learning*

Machine learning can basically be regarded as a generic term for computer systems that are able to learn, i.e. improve their performance with data. During a training phase, such a systems learns from examples and is able to make predictions afterwards. Exactly this behavior will be utilized by the approach presented in this paper: we want to learn recurring structures inside floor plans of buildings that are as generic as possible in order to reuse the generated model on new and unknown floor plans. As our approach uses isovist measures for training, a necessary precondition is the assumption that recurring structures in buildings also have recurring structures in their isovist measures.

Generally, machine learning can be divided into several categories. **Unsupervised learning** methods use a set of unlabeled input data in order to learn a function that describes the data's inherent structure. In our case the input data consists of a large set of isovist measures forming time series that have been calculated along geospatial trajectories. As the input data is unlabeled, the algorithm has no explicit target values to learn and instead tries to determine a function that reflects patterns in the data. A popular example of unsupervised learning is clustering, that is the automatic segmentation of data into groups of "similar" observations. Partitioning clustering techniques subdivide data into a predetermined number of k clusters. The assignment of observations to clusters will be adjusted in a way that minimizes a certain error function. An example for this kind is the widely used *k-means* partitioning clustering technique (Lloyd 1982).

Density-based clustering techniques arrange objects into groups which are close to each other, separated by areas with lower density. An example for such a technique is *DBSCAN* (Ester et al. 1996). This algorithm is parameterized by two variables: ϵ representing the distance up to which two observations are reachable and *minPts* representing the minimal number of reachable observations that make an observation a cluster point.

As the input data used in unsupervised learning is unlabeled, it is not straightforward to judge the quality of a certain clustering result. Still, there exist some techniques like the computation of silhouette coefficients as proposed by Rousseeuw (1987) which try to aid in the interpretation and validation of clustering results. Silhouette analysis works by computing the silhouette value $s(i)$ defined as

$$s(i) = \frac{b(i) - a(i)}{max(a(i), b(i))} \tag{1}$$

for all samples i. $a(i)$ being the average dissimilarity of i to all other samples in cluster A. $b(i)$ being the minimum of the average dissimilarity of i to all objects of any cluster $C \neq A$. It is then possible to compute the silhouette coefficient as the mean of the silhouette values of all samples belonging to a single cluster or the complete dataset. Possible values are in the range of $[-1, 1]$ with -1 being the worst possible score, 0 indicating overlapping clusters and 1 being the best possible score, i.e clearly defined clusters.

Supervised learning algorithms, by contrast, try to learn a function based on given pairs of input and corresponding known output labels. Applications of supervised learning can further be divided into *classification* problems, where the output is one of a finite number of discrete categories and *regression* problems, where the output is one or more continuous variables. In our case, we perform classification where the input data again consists of a large set of time series of isovist measures calculated along geospatial trajectories and the output is a corresponding known ground truth category, representing for example: "at this point the agent resides inside a room" or "at this point the agent traverses a doorway". A popular use case for supervised learning, in which a lot of progress has been made in the last years, is the automatic classification of images using deep learning techniques with neural networks (Deng, Yu et al. 2014). Given enough input data and the right network structure, neural networks are able to learn arbitrary functions from labeled data sets. Using images of known classes for example, a model can be trained that learns a function determining the class membership of previously unseen images. Or in other words: Observations that have not been used during the training phase. In our case a model is trained on isovist measures of floor plans where the categories representing e.g. rooms, hallways, and doorways are known. Afterwards, this model can be used on unlabeled floor plans where no such semantic information is available.

2.3. *Statistical Data Analysis*

The field of statistical data analysis is often subdivided into descriptive, exploratory and confirmatory data analysis. For the purposes of this paper, we mostly focus on exploratory data analysis (EDA). For a more in depth discussion of the field, see for example Ratner (2017). An often used method from the field of EDA is Principal component analysis (PCA), which was first proposed by Pearson (1901). It is a specific method from the subfield of dimensionality reduction, which performs feature projection. Using it, it is possible to transform data from a high-dimensional space to a lower-dimensional space which can then be plotted and analyzed visually. PCA works by iteratively selecting the component with the highest variance which is orthogonal to the previous identified component. This way, the i original features are replaced by $p < i$ uncorrelated linear combinations of the original features. Each original feature contributes to a varying degree to each of these p principal components. It is then possible to chose for example those two principle components that best explain the variance in the input data and by this reduce the dimensionality to 2D.

2.4. *Cluster Performance and Similarity Analysis*

While there is a common consensus on how to evaluate the performance of supervised learning algorithms (e.g. by counting the errors or measuring precision and recall), measuring the performance of clustering algorithms is not as easy and is consequently still an active field of research. Nevertheless, several different algorithms have been proposed over the years. Two important categories are measures based on counting pairs and measures based on mutual information.

All measures of the first category count pairs of objects that are classified in the same way in both clusterings, i.e. they are in the same cluster (or in different clusters, respectively) under both clusterings. Examples of this category are the *Chi Squared Coefficient* (Pearson 1900), *General Rand Index* (Rand 1971) and *Adjusted Rand Index* (ARI) (Hubert and Arabie 1985).

Measures of the mutual information category arose from the field of information theory and are based on the concept of entropy. Examples of this category are *Normalized Mutual Information* (NMI) (Strehl and Ghosh 2002) and the more recently proposed *Adjusted Mutual Information* (AMI) (Vinh, Epps, and Bailey 2010) which like the ARI measure corrects for chance.

For an extensive survey and in-depth comparison of the advantages and disadvantages of the individual algorithms, see Wagner and Wagner (2007).

2.5. *Semantic Annotation of Floor Plans*

Map representations of spatial environments are an essential foundation for most location-based services. Even if the positioning of an object works without a map representation, further benefit can only be created using a map. Examples are road maps, touristic maps or floor plans of buildings.

Such map representations can include logical subdivisions. Road maps involve country roads, highways, crossroads, turns and more. Buildings, for example, can be subdivided into rooms, zones, units, and levels (Weber et al. 2010). Besides that, there are semantic subdivisions like rooms, hallways, and doors. These are the focus of the paper at hand.

There is extensive related work regarding semantic annotation of architectural floor plans. Samet and Soffer (1994) perform automatic interpretation of floor plans using statistical pattern recognition. Their work is distinct from ours as we do not detect concrete objects like tables or bathtubs explicitly marked in architectural plans. Ah-Soon and Tombre (1997) analyze architectural drawings using geometric analysis, symbol recognition, and spatial analysis. Again, our approach is not geometrical, but instead uses the numerical representation of visual perception. Dosch et al. (2000) aim to reconstruct the building in 3D, based on architectural drawings. Using graphic recognition for image processing and feature extraction, the authors are able to recognize graphic layers, text layers, thick and thin lines as well as marked doorways, stair cases and more. Summarized, they try to identify marked semantics and transform this into 3D. In contrast, we try to identify semantics that are not explicitly marked. Lu et al. (2007) is a further work that tries to recognize typical structural objects and architectural symbols. Our approach works on floor plans that can be used by robots and not on architectural drawings. Weber et al. (2010) presents a system where a user can draw schematic abstractions of floor plans. Afterwards, the system searches for plans that are structurally similar. This is quite related to our approach, since they also seek for semantic relations. However, our focus is not on searching in databases, but on learning a model.

Further related work originates in the research field of mobile robots. What this work has in common, is that the ideas can be used for the problem of *Simultaneous Localization and Mapping* (SLAM) (Leonard and Durrant-Whyte 1991). This means, an autonomous robot has to examine an unknown area and try to create a corresponding floor plan. Concurrently, the robot has to position itself. Thus, it makes sense to enrich the map just created with additional semantic information. The basic assumption is that the robot's perception, in most cases laser range scans, contains enough information about the environment. In a way, we follow this approach as well, since we use isovist measures based on rays. Buschka and Saffiotti (2002) describe a virtual sensor that can be used to detect rooms and to recognize already visited rooms in order to create a topological map of the environment. Our focus is wider than just detecting rooms, although we do not address topology. Anguelov et al. (2004) present a probabilistic framework for detecting and modeling doors. They use 2D laser range finders, but also panoramic cameras. Mozos and Burgard (2006); Mozos (2010) extract the topology of buildings from geometric maps created by mobile robots using range data. The authors use supervised learning techniques in order to subdivide all points of the map into semantic classes. For this, they use the labels *room*, *corridor*, and *hallway* as the ground truth. This approach is very similar to the one presented in this paper, but the authors work only with supervised learning techniques and with different, yet similar features. Goerke and Braun (2009) is also a similar related work that semantically annotates maps using laser range measurements of mobile robots. The authors follow two basic approaches. First, they use supervised learning techniques with the labels *doorway*, *corridor*, *freespace*, *room*, and *unknown*. Second, they use unsupervised learning techniques, but state that this approach did not produce satisfying results. Furthermore, the authors only work on a single floor plan, while our paper in particular addresses the aspect of generalization, which is why multiple maps are used. Chen et al. (2014) use deep learning techniques to identify doors, so that autonomous mobile robots are able to approach targets more accurately. Their focus is only on detecting doors visually, using cameras.

There is further related work on analyzing architectural space using isovist analysis. Bhatia et al. (2012) use 3D isovists in order to estimate salient regions in architectural and urban environments. Thus, the authors are able to detect regions that posses strong visual characteristics. Our approach focuses on recurring and not on salient structures. Feld, Werner, and Linnhoff-Popien (2016) approximate four out of six isovist measures using a simple ray-casting approach while showing that the resulting error is systematically yet small, and the exact and approximated values show a strong correlation. Furthermore, they show with a few examples on a single map that trajectories of isovist measures potentially provide clues to identifying doors. The paper at hand goes much further and creates a model to recognize such structures. Feld, Lyu, and Keler (2017) calculate isovist measures on 2D floor plans, cluster the values using archetypal analysis and interpret the results afterwards. They show that the identified clusters correspond to regions like streets, rooms, hallways, and the like. However, their approach is unsupervised learning with interpretation of relations, thus, they do not learn a specific model using which predictions can be made. Sedlmeier and Feld

(2018) presented the first parts of the framework, on which this paper extends. Their framework is able to create a data set containing 2D isovist measures calculated along geospatial trajectories that traverse a 3D simulation environment. Results showed that these isovist measures do reflect the recurring structures found in buildings and the recurring patterns are encoded in a way that unsupervised machine learning was able to identify meaningful structures like rooms, hallways and doorways. Neural network based supervised learning was then successfully used to train models that generalize and are able to identify structures in different environments.

3. Methodology

This section is split into multiple parts: (1) It describes our framework for generating isovist measures along geospatial trajectories in a map-based simulation environment. These measures provide the input for the following steps, (2) data preprocessing, (3) the discovery of recurring structures in floor plans, (4) the generation of temporal features, (5) the learning of models and prediction of structures on other floor plans, all of which enable (6) a qualitative visual data analysis in map- and feature-space, and (7) the calculation of measures which quantify the visual perception similarity between environments.

Unsupervised learning techniques are employed in the discovery phase, while supervised machine learning is performed for the modeling and prediction tasks. The output of both are then used as input for visualization and cluster comparison techniques. Details regarding the exact implementation of these aspects can be found in Section 4.

3.1. *Input Generation*

Basic input for the framework is supplied as bitmap files representing building floor plans. Walkable space is represented as white pixels, while black pixels depict obstacles like walls or furniture. Note that doors are excluded. In a first step, these bitmaps are vectorized using a common vector graphics editor. The vector files are then imported into Blender, an open-source 3D computer graphics software (Blender Foundation 2017), where a 3D-Extrusion is performed in order to generate a 3D map of the building. These 3D maps serve as the basic asset for Unity, a 3D game engine and development environment (Unity Technologies 2017). For each map, a navigation mesh (Snook 2000) is generated in Unity to enable automatic navigation and pathfinding. Custom built C# scripts then enable a player object (non-player character, NPC) to automatically select a random point on the navigable area inside the map and move towards it using Unity's built-in navigation algorithm. For each step of the NPC, another custom C# script was developed, which performs isovist measure calculations and logs the results to disk. In order to generate the isovist, a configurable amount of rays are cast, originating from the current position of the NPC, as can be seen in Figure 1. Points in space, where the rays intersect with the map's mesh colliders (hitpoints) are detected and used to calculate the different isovist measures.

The isovist measures calculated are based on Benedikt (1979), as previously described in Section 2.1. As a discrete, ray-based isovist calculation is used, the calculated measures are only an approximation of the true isovist measures. The accuracy of the calculation can be adjusted, as the amount of rays cast is configurable.

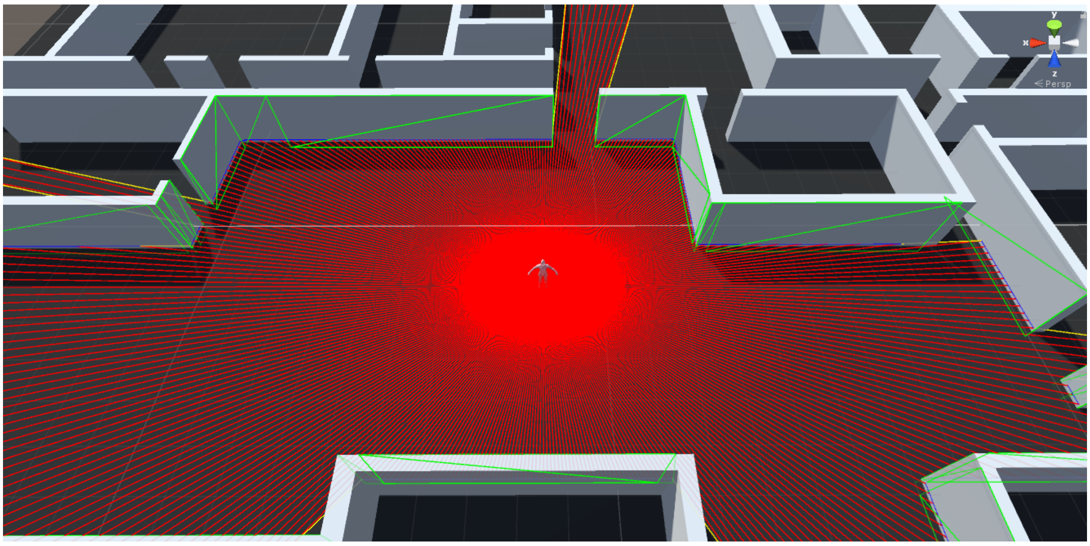


Figure 1. 3D view of a utilized floor plan, showing the non-player character casting 360 rays (red lines) from it's current vantage point.

One of the more challenging aspects to calculate is the differentiation between *real-surface* and *occlusivity* of the isovist. Benedikt states in Benedikt (1979) that the occlusivity of an isovist "measures the length of the occluding radial boundary R_x of the isovist V_x and indicates [...] the depth to which environmental surfaces are partially covering each other as seen from the vantage point.".

In order to be able to differentiate occlusion from real-surface in our simulation's engine, we developed an algorithm which performs calculations based on the triangles that form the mesh of the environment. For every ray cast in a clockwise manner, a comparison with the previous ray's hitpoint on the environment's surface is performed. If the previous ray hit a triangle which shares none of it's edge coordinates with the currently hit triangle, we define the current ray to have hit an *un-connected triangle* (in respect to the previous triangle). The length of the line connecting the previous and current ray hitpoint in space is then counted towards the occlusion value of the isovist. If a *connected triangle* was hit, the length of the connecting line is counted towards the real-surface perimeter of the isovist. Figure 2 shows the resulting lines calculated by our algorithm inside the Unity engine. Red lines are the rays cast from the current vantage point, green lines visualize the hit triangles of the meshes, blue lines denote real-surface, while yellow lines denote occlusion.

3.2. *Data Preprocessing*

In a first step of preprocessing, the logged isovist measures calculated during simulation time are vectorized in order to retrieve a suitable data set for unsupervised learning. The vectors created this way are 6-dimensional, with each dimension representing one of the 6 measured isovist features as defined by Benedikt (1979) and explained in section 2.1.

As most machine learning estimators require standardized datasets to perform well, we apply a scaling and normalizing procedure to standardize the features. The reason for this is that when performing multidimensional clustering, the distance between the units in the data is often important (Everitt and Hothorn 2011). By standardizing, we

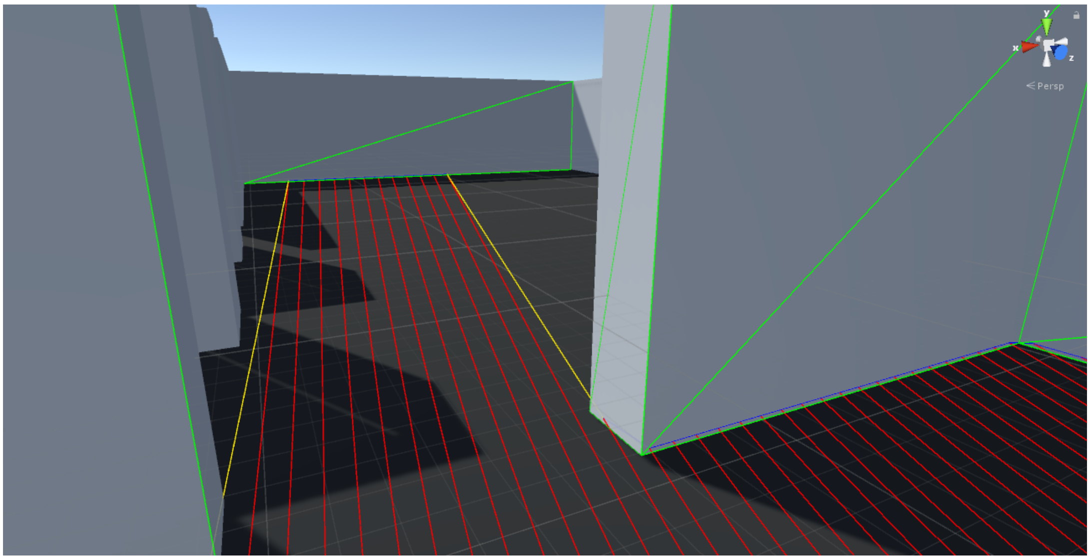


Figure 2. In-engine view of the custom built algorithm's results for real-surface and occlusion isovist measure calculation. Red lines are the rays cast from the current vantage point, green lines visualize the meshes' triangles hit by the rays, blue lines denote real-surface while yellow lines denote occlusion.

make sure that by default, all of the isovist features like *area* or *occlusivity* are treated equally and none has a higher impact just because it's values are scaled differently. The process is performed by removing the mean value of each feature, which centers the dataset. Afterwards, the data is also scaled by dividing the features by their standard deviation.

This procedure has a positive secondary implication. It removes the effects caused by differently scaled map representations from the isovist measures. After the data is normalized, it does no longer matter whether the data was collected on floor plans of different scale.

3.3. *Unsupervised Learning of Unknown Floor Plan Structures*

The first part of the learning framework is responsible for discovering hidden structures contained in the isovist measures. The goal of this step is to group input data into meaningful clusters, each representing a human-relatable concept, for example "isovists recorded in rooms" versus "isovists recorded in corridors". This can be achieved using unsupervised machine learning techniques. We employ k-means (Lloyd 1982), a centroid-based clustering algorithm, as well as DBSCAN (Ester et al. 1996), a density-based clustering algorithm, both of which are implemented in the scikit learn python library (Pedregosa et al. 2011). For k-means, we make use of the calculation of silhouette coefficients. This allows us the get a first insight into the internal structure of the data as well as choose appropriate values for the amount of clusters with which we perform the evaluation.

3.4. *Temporal Feature Calculation*

It is important to keep in mind that the input data given to the clustering algorithm as described above is static in nature. That is to say, each data point contains only the isovist measures of a single position along the trajectory trough the map. As there is no temporal component involved, the concept of movement and the dynamic change of isovists while moving along a path is not reflected in this analysis. The overarching idea of this step, the inclusion of time, is to not only reflect the perception of "space" but the "changing of space perception" as caused by movement.

In order to tackle this idea, an additional data processing step was developed which reflects the temporal dimension of the data. For every feature of each data point, the delta of the current data point's feature x_c and the simple moving average (SMA) (Balsamo, Knottenbelt, and Marin 2013) – a method commonly employed in the statistical analysis of time series – of n previous data points' features, is calculated:

$$x_c - \frac{1}{n}\sum_{i=1}^{n} x_{c-i}$$

This way, the amount of features available to the machine learning algorithms is doubled from 6 to 12.

3.5. *Supervised Learning of Known Floor Plan Structures*

Using the method described in the previous section, labeled input data can be generated, which forms the basis for a following supervised machine learning step. The goal of this step is to learn a model representing the structures discovered in the data, by learning a function which maps new unlabeled input data points to the respective cluster categories. This model can then be used, for example, in mobile robots as a lightweight component enabling the robot to infer the type of room it currently resides in, or whether it has just passed a doorway, by feeding it's current and previous isovist measures into the model. For our framework, we chose to implement a multi-layer feed-forward neural network using the open source neural network library Keras (Chollet et al. 2015), which in turn uses TensorFlow (Abadi et al. 2015), a low level machine learning library developed by Google, to execute it's calculations. Training data is provided by the labeled input data, as output from the unsupervised learning step. In order to verify the validity of the model generated using supervised learning methods, it is important to separate validation from training data. For this, we split the data into a left half and a right half. The split is performed depending on which half of the floor plan (left / right) the location of the vantage point falls in. This ensures that no part of the building in the validation data has been used in the training data. Training was performed on the right half of the data, while validation was performed on the left half. As part of our evaluation of different neural network architectures, we found a rather small network of 5 fully-connected layers to be sufficient for our purposes. The input layer contains 12 neurons (one for each feature), connected to 3 hidden layers, each containing 64 neurons, followed by an output layer containing 4 neurons. Softmax activation is used on the 4-neuron output layer in order to build a classifier representing the 4 cluster labels, while rectified linear unit (ReLU) activation (LeCun, Bengio, and Hinton 2015) is used on all other layers. Categorical cross entropy is employed as the loss function while Adam (Kingma and Ba 2014) is used as the stochastic gradient descent algorithm. All in all, the network contains over 9,000 trainable parameters. After the training step, the best model is selected based on the model accuracy score. In order to test the generalization capacity of the trained model even further, the model is then used to predict values from data collected on a different floor plan. The question to be answered by this is whether the model learned general abstractions (e.g. a concept of "doors"') that capture underlying basic principles of the data which are independent of the specific floor plan layout.

3.6. *Visual Data Analysis*

Sedlmeier and Feld (2018) were able to show that the previous steps of the framework allow the successful learning of recurring structures in building floor plans. We now show that being able to learn and recognize these structures also makes it possible to judge the similarity or dissimilarity of building floor plans. In a first step, we show that by visually comparing different models' cluster membership predictions, it is possible to reach a deeper understanding of the data's structure and differences between the floor plans. For this, we performed a Principal Component Analysis (PCA) (Pearson 1901). This step is necessary, as the isovist data is high-dimensional and cannot be directly visualized in a meaningful way. We replace the original isovist features by 3 uncorrelated linear combinations of the original features and plot them using the TensorBoard tool which is part of the TensorFlow Suite (Abadi et al. 2015). The resulting plots show the data in so called *feature-space*, i.e. without the floor plan's X- and Y-coordinates, using only the 3 principal components as axes. By coloring the datapoints according to their cluster- or prediction-labels, a visual comparison of the results becomes possible. Additionally, we compare the *feature-space* visualizations with the *map-space* visualizations, i.e. by plotting the data points using the X- and Y-coordinates and coloring according to the labels again.

3.7. *Quantifying the Visual Perception Similarity of Environments*

The previous step allows a qualitative comparison of the visual perception similarity of building floor plans. We now propose to apply cluster comparison measures in order to quantify this similarity of visual perception. By performing an algorithmic cluster similarity analysis, we show that it is a promising concept for quantifying visual perception similarity/dissimilarity. We compute multiple cluster similarity scores between the direct clustering results of the data points on the floor plans and the cluster memberships predicted by the models trained on different floor plans. By comparing the similarity scores with the results of the visual data analysis, we show that the scores match our human judgement of the floorplans' similarity.

4. Evaluation and Discussion

For the evaluation, four distinct floor plans of different buildings were chosen: A section of a university building of the Ludwig-Maximilians-Universität München (LMU), part of the Queen's University School of Medicine Clinical Simulation Centre (QueensU), the main hall and connected rooms of the Technische Universität München (TUM) and the ground floor of a hotel building in Amsterdam (Hotel). The floor plans feature different amounts of repeating structures like corridors and rooms. Overall, the LMU and QueensU plan appear more structured and uniform when compared to the more irregular structure formed by large halls and hallways of the TUM and Hotel floor plan. Using the input generation framework described in section 3.1, more than 300,000 isovists were recorded along random trajectories on each floor plan.

We compared two different unsupervised machine learning methods: k-means, a

Table 1. Silhouette scores of k-means based clustering on all four floor plans with different amount of clusters (k).

k	QueensU	Hotel	LMU	TUM
2	0.405	0.442	0.690	0.315
3	0.325	0.584	0.478	0.383
4	0.388	0.489	0.507	0.396
5	0.364	0.454	0.446	0.413
6	0.361	0.437	0.380	0.386
7	0.331	0.388	0.336	0.393
8	0.329	0.404	0.316	0.372
9	0.331	0.408	0.319	0.325

centroid-based clustering algorithm and DBSCAN, a density-based clustering algorithm. As it is possible to define the amount of clusters to be found when using k-means, the results of using different values were compared. The computation of the silhouette score as the mean silhouette coefficient of all samples was used to analyze the quality of the clusterings for different amounts of clusters.

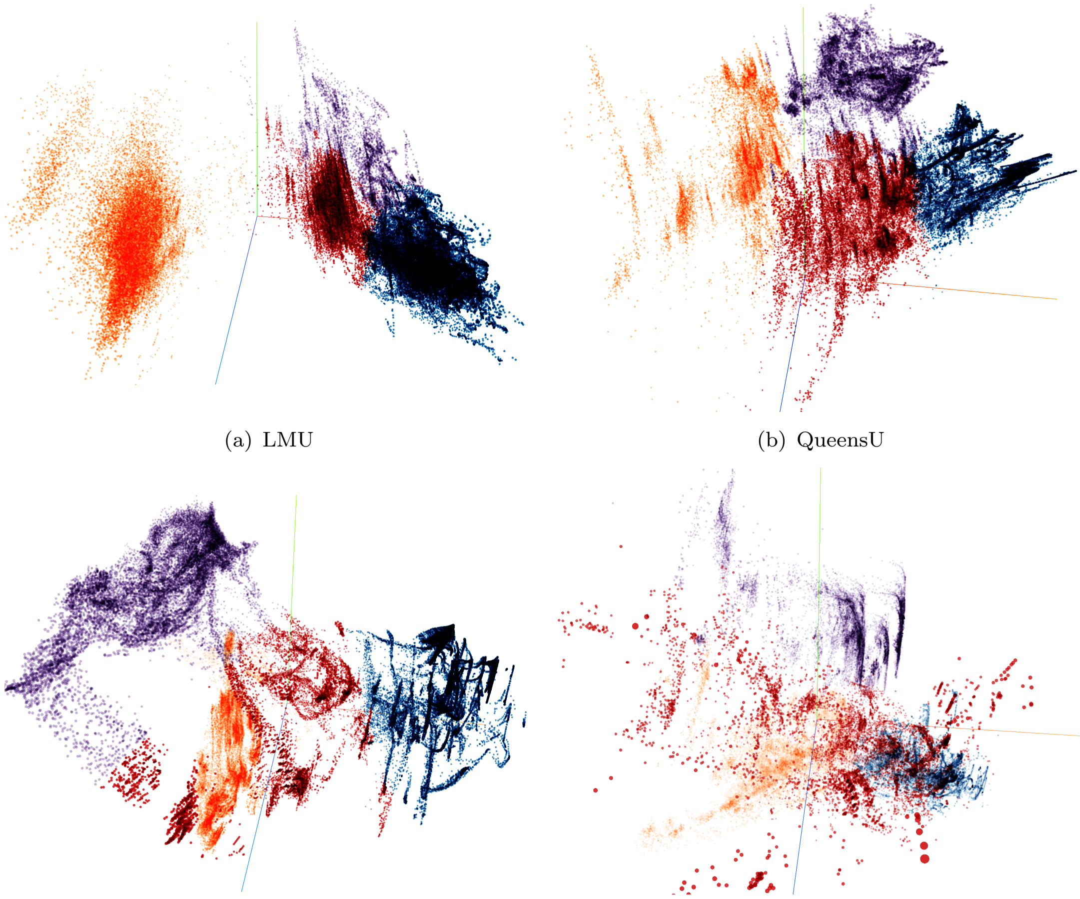
As can be seen in table 1, there is no common best value of k for all maps. This is why we decided to continue the first part of the following visual data analysis with varying amounts of clusters. For the model building steps in section 4.3, we chose to use a common amount of 4 clusters for all maps, as it is the second best value for each map and might provide more insight into the structural differences and similarities of the floor plans than using a very low value like $k = 2$ would.

4.1. *Visual Data Analysis via PCA Feature Extraction*

In order to get an intuition on how the unsupervised learning methods reach their results, as well as on the similarity of the floor plans, we performed a visual data analysis on the data sets. By visualizing the results of the different cluster configurations, it is possible to reach a deeper understanding of the data's structure. For this, we performed a Principal Component Analysis (PCA) (Pearson 1901). During the process, the i original isovist features are replaced by $p < i$ uncorrelated linear combinations of the original features. Each original feature contributes to a varying degree to each of these p principal components. In our case, we chose those 3 principle components that best explain the variance in the input data. When plotting these 3 dimensions, the internal structure of the dataset becomes apparent. In an additional step, each data point in the resulting 3D-plot can be colored according to it's cluster label.

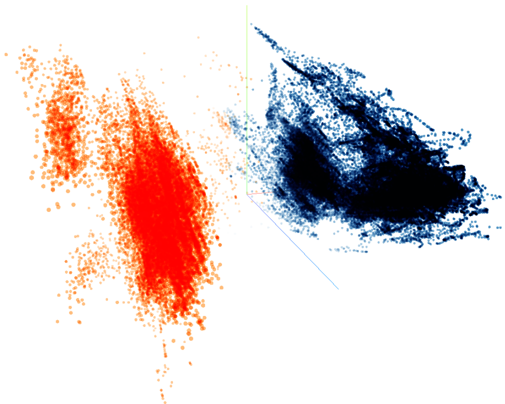
As can be seen in figure 3, the different floor plans show varying amounts of visible structure in the 3-dimensional PCA decomposition, plotted using the TensorBoard tool which is part of the TensorFlow Suite (Abadi et al. 2015).

Figure 4 shows the results of these steps for different cluster configurations on the LMU floor plan. When viewed in feature-space, i.e. without X,Y map-coordinates, instead using the PCA-Components as axes, the clear differentiation between the left (orange) cluster and the other clusters becomes apparent. This clear separation matches the fact that the LMU floor plan had it's highest silhouette score (0.69) for the configuration $k = 2$. When plotted in map-space, i.e. using the original X,Y map-coordinates of the data points as axes, the meaning of this left cluster becomes apparent: All isovist measures forming these data points were collected at locations in the large corridor as can be seen in Figure 5. The differentiation of the right part of the data points for $k >= 3$ is not as clear when viewed in feature-space. Still, meaningful concepts are

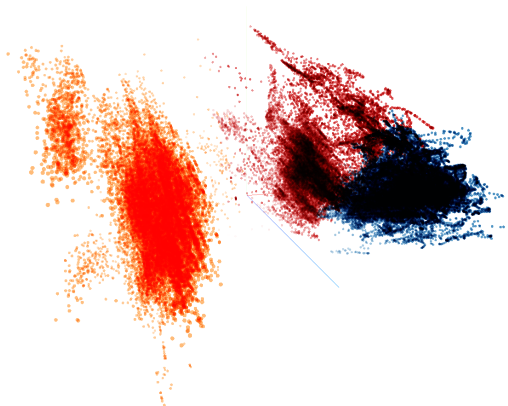


(a) LMU (b) QueensU (c) Hotel (d) TUM

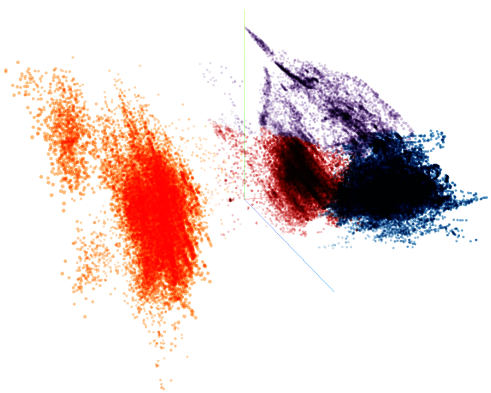
Figure 3. Results of the PCA decomposition of the different floor plan data sets, using the 3 principal components that best explain the variance in the data as axes. Data points are colored according to a k-means based clustering with $k = 4$. An interactive online version is available via *https://github.com/sedand/isovista* and can be used to view these plots in 3D.



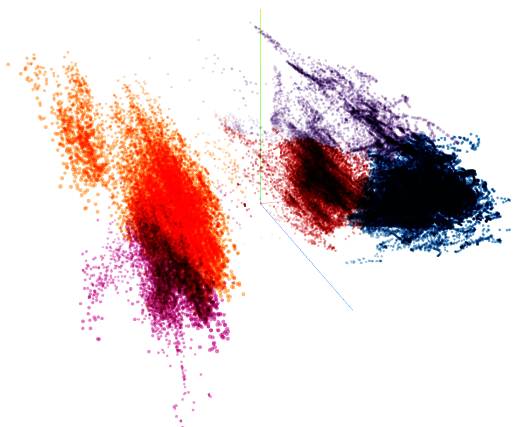
(a) $k = 2$



(b) $k = 3$



(c) $k = 4$



(d) $k = 5$

Figure 4. PCA decomposition of the LMU Isovist data set, showing data points colored according to a k-means based clustering with varying configurations of parameter k.

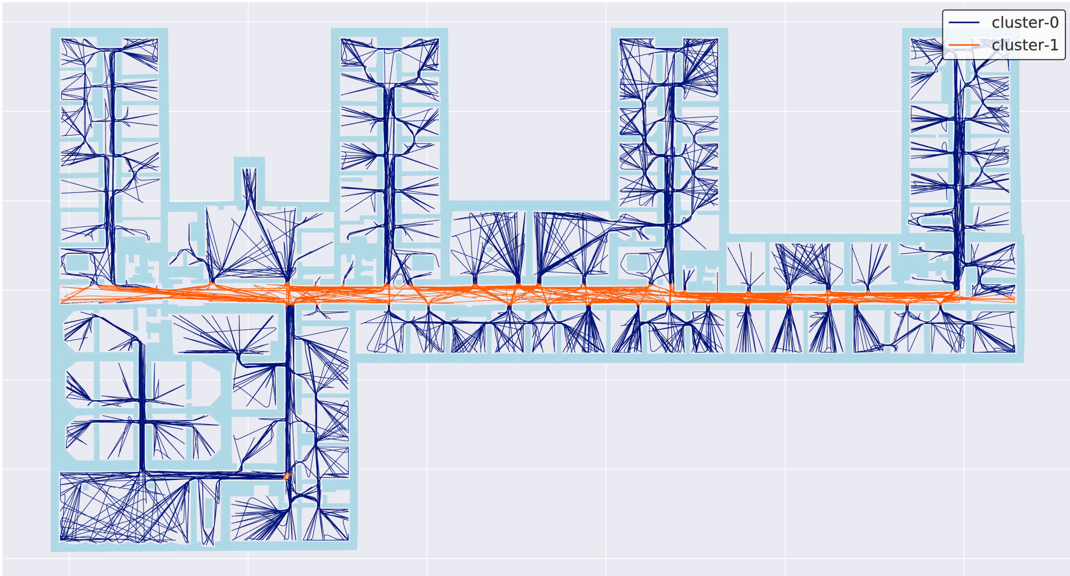


Figure 5. Results of a k-means based clustering with $k = 2$ of static isovist measures on the LMU floor plan. Cluster labels/colors are the same as in Figure 4(a).

captured in these clusters that become visible when plotted in map-space as can be seen in Figure 6. For $k = 5$ the left cluster is further split into two separate clusters. From a human point of view, this seems rather arbitrary and is not easily interpretable when visualized in map-space. Map-space plots for different values of k are available in the appendix.

It is important to note that a human interpretation of the formed clusters can be misleading. The performed analysis cannot give a direct high-level, conceptual answer to the question of *why* these clusters were formed. Still, the general characteristics of Isovists, for example the fact that they are rotation, but not scale invariant, may provide a hint at which aspects can or cannot play a part in the cluster formation. For example, the size and circularity of a room together with the amount of occlusion may play a part, but not the room's rotation in map-space.

4.2. *Cluster-Result Comparison in Map-Space*

We chose to continue our evaluation with a value of $k = 4$, as we think it provides a good tradeoff between high silhouette score and usefulness from the point of view of an application. For visualizations of other clustering configurations and different floor plans, see the appendix. As can be seen in Figure 6, for this cluster configuration, four different structures of the LMU floor plan are separated into different clusters. A clear separation between the large (orange) corridor, the smaller (red) corridors, and two different room structures (purple and blue) becomes apparent. It is important to keep in mind, that human concepts are not necessarily reflected by the clusters, which is why the meaning of a cluster is always subject to interpretation.

Besides the centroid-based clustering algorithm k-means, we also evaluated the density-based clustering algorithm DBSCAN. As the amount of clusters to be found is not to be specified in DBSCAN and can only be indirectly influenced by configuring two density parameters ϵ and *minPts*, it is a lot harder to produce a sensible amount of human interpretable clusters. For our data set, an ϵ value of 3 and *minPts* values

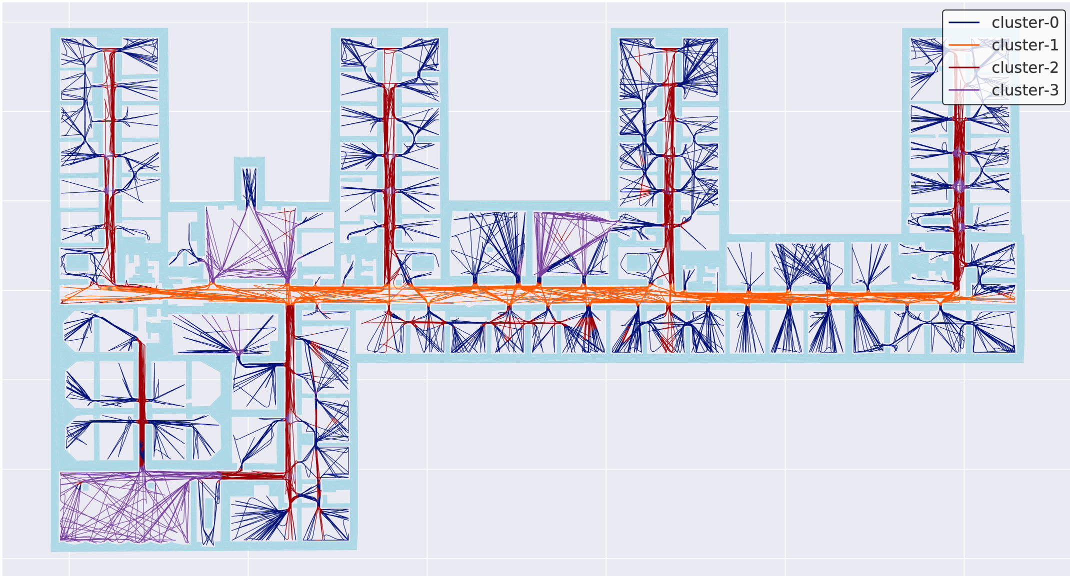


Figure 6. Results of a k-means based clustering with $k = 4$ of static isovist measures on the LMU floor plan. Cluster labels/colors are the same as in Figure 4(c).

between 1000 and 2500 produced meaningful results. Compared to the clusters produced by the k-means algorithm, the resulting structures were less interpretable (see figure 7). This is why we decided to continue our analysis using the k-means based clustering.

After clustering the static data features and finding clusters that could be interpreted as rooms and corridors, the delta of every isovist measure and the SMA of the previous measures was calculated, in order to capture the temporal dimension of the data. Using these "delta-features" as input to the k-means clustering, a completely different picture becomes visible: As can be seen in Figure 8, a cluster now formed around passage ways, especially doorways.

This intuitively makes sense, as doorways are components in a building, often connecting structures of different shape, which is why movement through them leads to changes in the perception of space, in turn reflected in high changes of isovist measures.

By combining these static-data and dynamic-data clusters, we generated a merged set of data-labels containing five different clusters as shown in Figure 9.

As an example, we list our interpretations of the clusters on the LMU floor plan:

Cluster-0 (blue): small rooms
Cluster-1 (orange): large corridor(s)
Cluster-2 (red): small corridors
Cluster-3 (purple): large rooms
Cluster-4 (green): passage ways (e.g. doors)

The clusters formed on different floor plans might be interpreted differently, depending on the structures present in the respective building. See the appendix for map-space visualizations of the clustering results of the other floor plans.

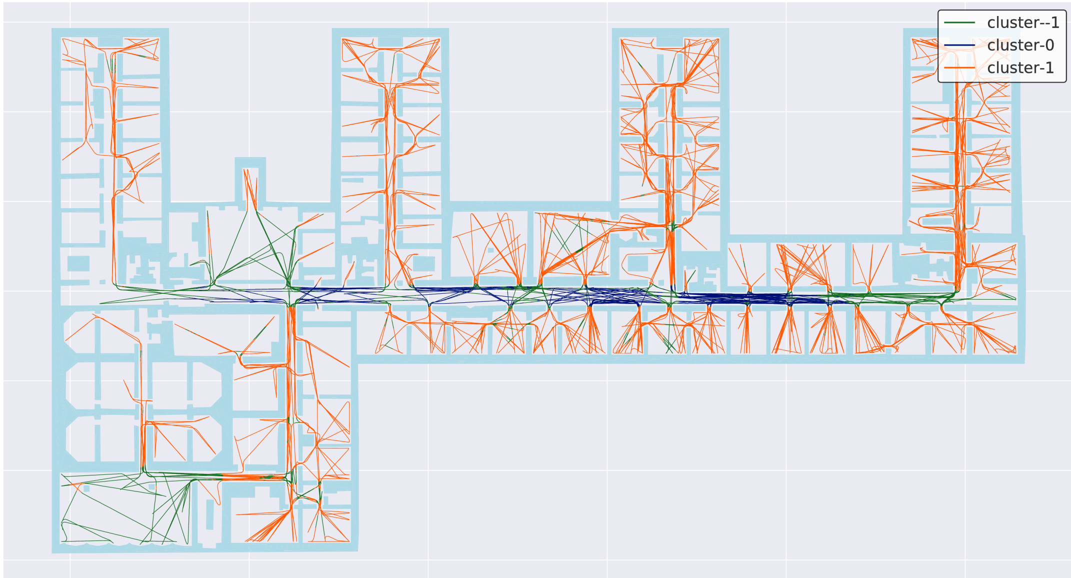


Figure 7. Results of a DBSCAN based clustering with $\epsilon = 3$, $minPts = 2500$ of static isovist measures on the LMU floor plan. We were unable to cluster all 300000 datapoints for performance reasons and restricted the DBSCAN clustering to the first 100000 measures.

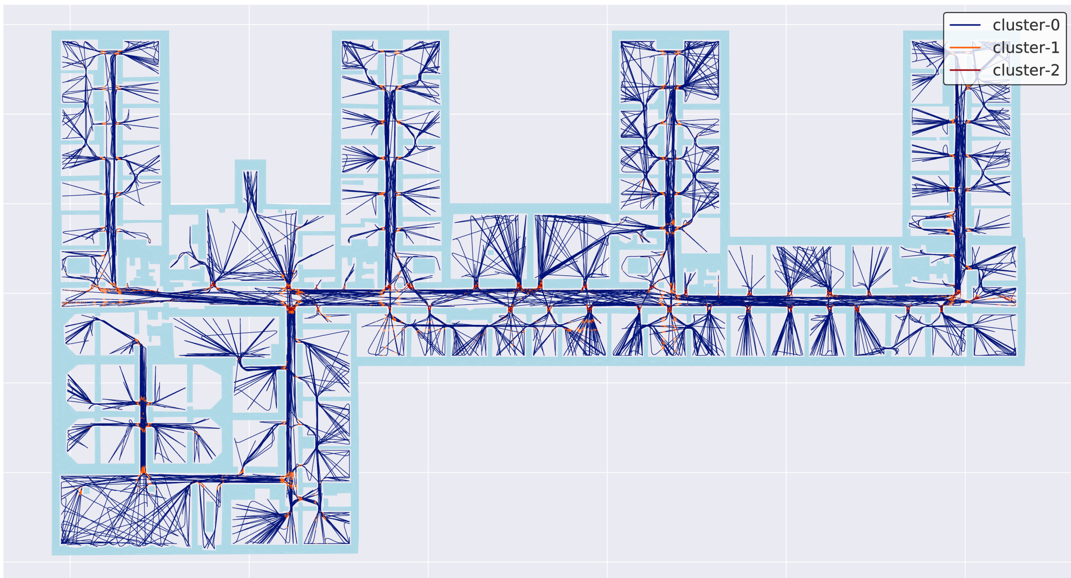


Figure 8. Results of k-means based clustering of dynamic, SMA based isovist measures ($k = 3, c = 5$).

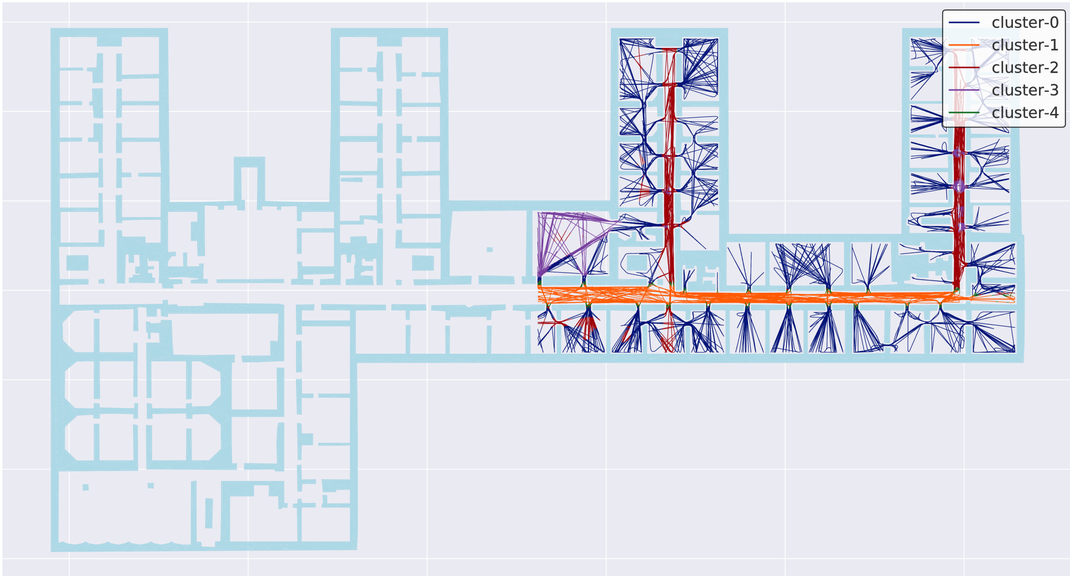


Figure 9. Visualization of training/validation data split after merging static and dynamic features and labels. Only training data is visualized in this figure.

4.3. *Supervised Learning Results*

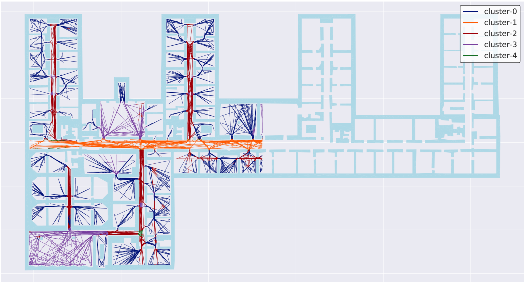
Figure 9 also shows the training/validation split that was performed on the data. Training was performed on the right half of the data, while validation was performed on the left half. Good results could be achieved when training a 5-layer fully connected feedforward neural network using the 12-dimensional feature vector (static & SMA deltas) as input and the 4 cluster labels described above as targets. Training showed very good train and validation set accuracy values for all floor plans, as can be seen in table 2.

Figure 10 shows that the predictions produced by the models on the validation data match our interpretations of the clusters in the training data (this can be regarded as *intra-map model generalization*). This shows, that the models were able to learn a function representing the structures bundled in the respective clusters. As the models were able to predict meaningful results on previously unseen parts of the floor plans, it becomes obvious that the models generalize to new data.

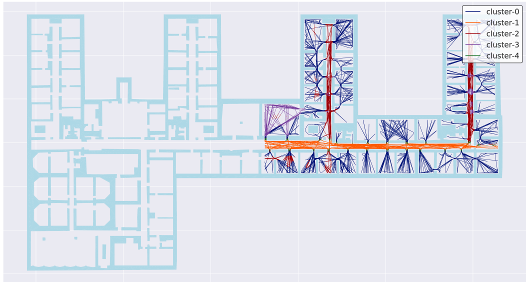
In order to test the performance of our models even further, we analyze the *inter-map generalization* capacity by using the models to predict the cluster memberships on the respective other floor plans. Table 3 shows prediction accuracy values of models which were trained on the complete data set of each map, by comparing the predictions of each model to the clustering labels of each floor plan. Overall high values of the LMU Model for example show that the structures found in this floor plan can be used to predict the structures of the other floor plans to a large extent. We present a

Table 2. Intra-map model accuracy and loss values.

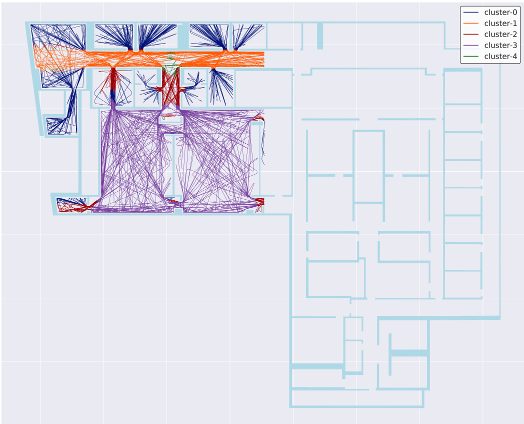
Metric	QueensU	Hotel	LMU	TUM
Training accuracy	0.925	0.972	0.975	0.940
Validation accuracy	0.942	0.958	0.977	0.932
Training loss	0.233	0.103	0.092	0.195
Validation loss	0.200	0.144	0.099	0.211



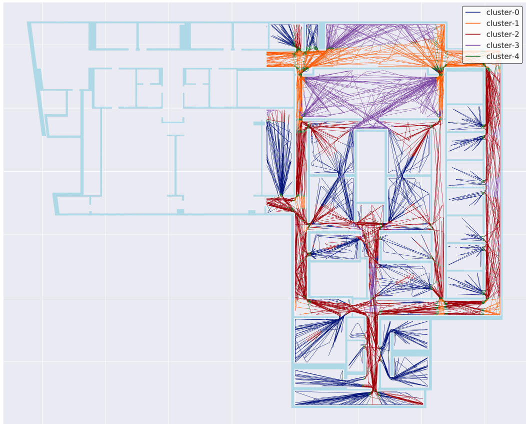
(a) LMU Model Predictions



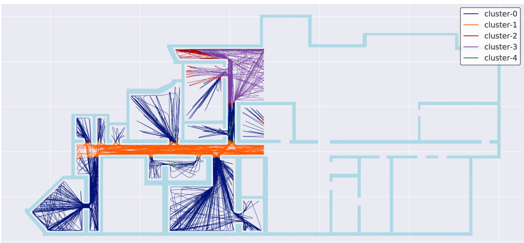
(b) LMU Model Training



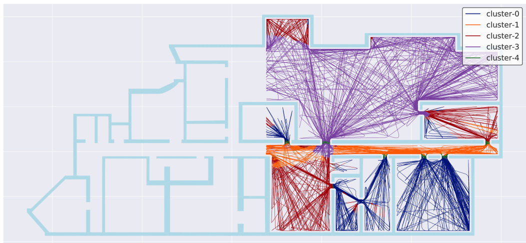
(c) Queens University CSC Model Predictions



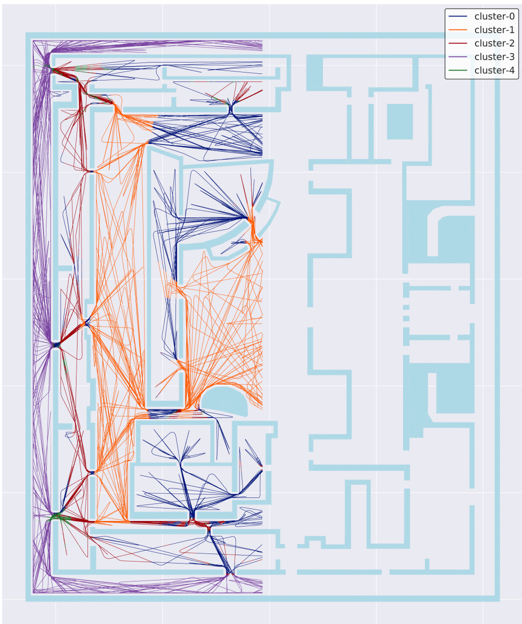
(d) Queens University CSC Model Training



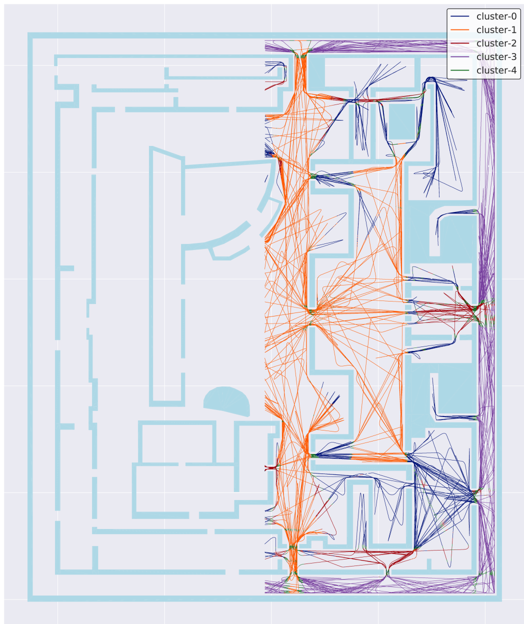
(e) Hotel Model Predictions



(f) Hotel Model Training



(g) TUM Model Predictions



(h) TUM Model Training

Figure 10. Intra-map model generalization. Left column shows predicted labels (cluster memberships) of all data points in the validation data set along random trajectories trough the floor plans. Training was performed using the 5-layer feedforward neural network on the right half of each map (right column) using the k-means cluster results as labels and the 12 combined static and dynamic isovist features as input.

Table 3. Inter-map model accuracies of all models for all floor plans in respect to clustering results of the floor plan. Columns specify models (which were trained on the complete dataset of the respective floor plan), rows specify the floor plan on which predictions were made.

Floor Plan	Model-QueensU	Model-Hotel	Model-LMU	Model-TUM
QueensU	0.940	0.653	0.910	0.547
Hotel	0.642	0.972	0.711	0.408
LMU	0.941	0.747	0.978	0.651
TUM	0.357	0.295	0.409	0.941

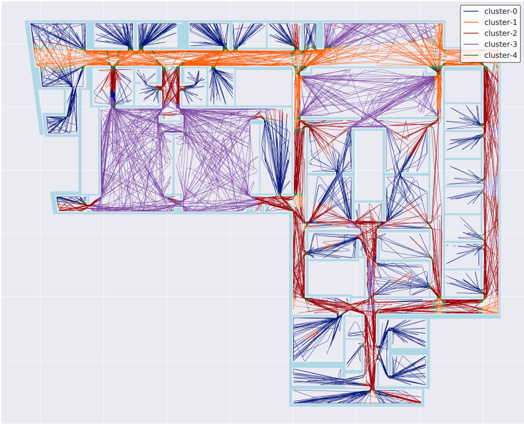
more in-depth comparison and discussion of the maps' similarities as well as problems inherent to this approach in section 4.5.

It is also possible to analyze the prediction results visually. Figure 11 shows the predictions of the LMU model when applied to the Queens University CSC floor plan. As can be seen in figures 11(b) and 11(c) by visually comparing the cluster assignments in map-space, the model predictions match the clustering assignments to a large degree. This result confirms our previous statement, which was based on the accuracy values, that the two buildings, LMU and QueensU, are structurally very similar. Figures 11(e) and 11(f) show that the TUM model by contrast does not generalize as well to the QueensU floor plan. It is not able to differentiate between the large corridor and the large rooms. Consequently, the prediction results of these data points differ, which in turn is reflected in the lower accuracy score of 0.547. Nevertheless, small rooms and the large corridor are assigned to the same clusters as is the case for the clustering result.

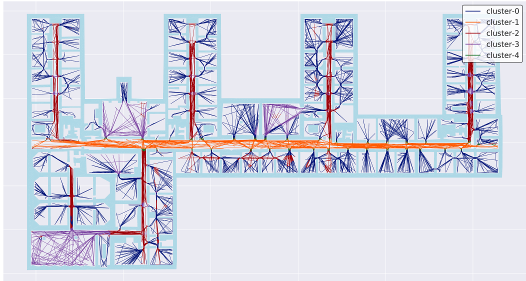
Next, we focus only on the prediction results of cluster-4 which is based on the temporal dimension of the data, the SMA features. Figure 12 shows the predictions of this cluster of the QueensU and TUM models on the LMU floor plan. Both models predict those points in the floor plan that can be understood as doors and passage ways or more generally: Points where a different structural part of the building becomes visible. This structural part can, but need not be explicitly separated by a door. So here again, the models do generalize very well to a different floor plan.

4.4. *Feature-Space Inter-Map Cluster Prediction Evaluation*

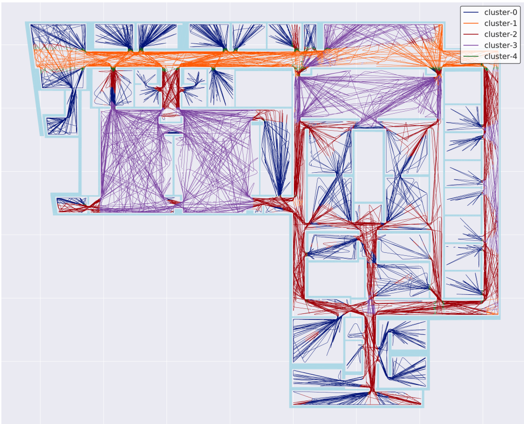
We now compare the prediction results of the different models in feature-space, i.e. without X,Y map-coordinates, instead using the PCA-Components as axes. Figure 13 shows the PCA decomposition of the different floor plan data sets, using various clusterings and models for labeling. As can be seen in the figures, the LMU and QueensU models assign almost all of the data points to the same clusters as the k-means algorithm on the respective floor plan does. This result again confirms our view that the two buildings, LMU and QueensU, are structurally very similar. When predicting the LMU or QueensU floor plan using the TUM model, the results 13(e),13(f) look different. On the QueensU floor plan for example, the TUM model does not split the data into the orange and violet cluster like the LMU model does. As we saw before when plotted in map-space, this results in the TUM model not being able to differentiate between the large corridor and the large rooms on the QueensU floor plan.



(a) K-Means clustering on QueensU

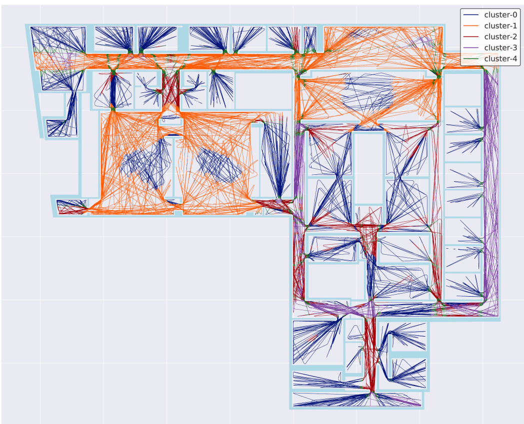


(b) LMU Model Training

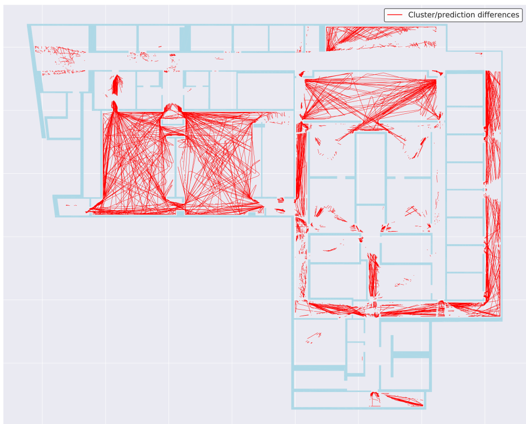


(c) Predictions on QueensU using LMU Model

(d) Differences between k-means clustering and prediction using the LMU Model

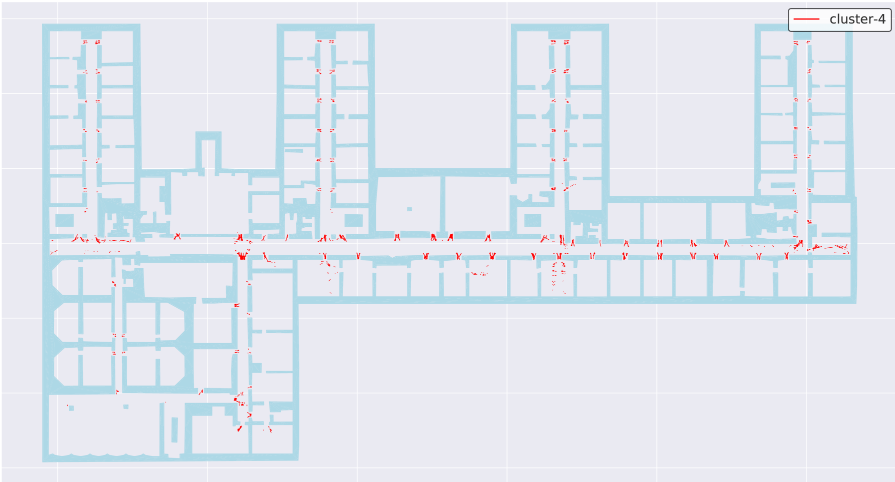


(e) Predictions on QueensU using TUM Model

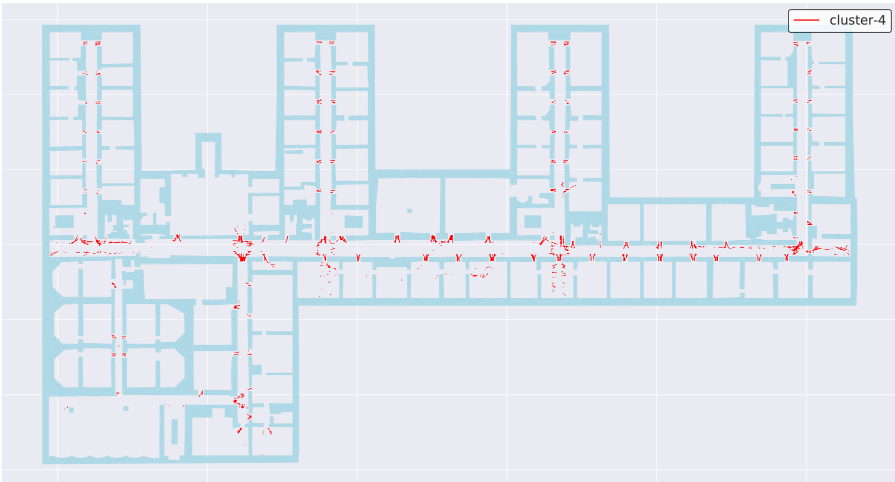


(f) Cluster membership differences between k-means clustering and prediction using the TUM Model

Figure 11. Inter-map model generalization. Figure 11(b) and 11(c) show a comparison between results produced when clustering the QueensU dataset and predicted cluster membership by the LMU Model when used on the Queens University CSC floor plan. The LMU Model *correctly* predicts the majority of the cluster membership of the data points (i.e. it assigns the same labels as the k-means clustering does). Figure 11(d) shows the exceptions, where the LMU model predicts a different cluster membership. Figures 11(e) and 11(f) show that the TUM model does not generalize as well to the QueensU floor plan. Nevertheless, small rooms and the large corridor are assigned to the same clusters.



(a) LMU using QueensU model

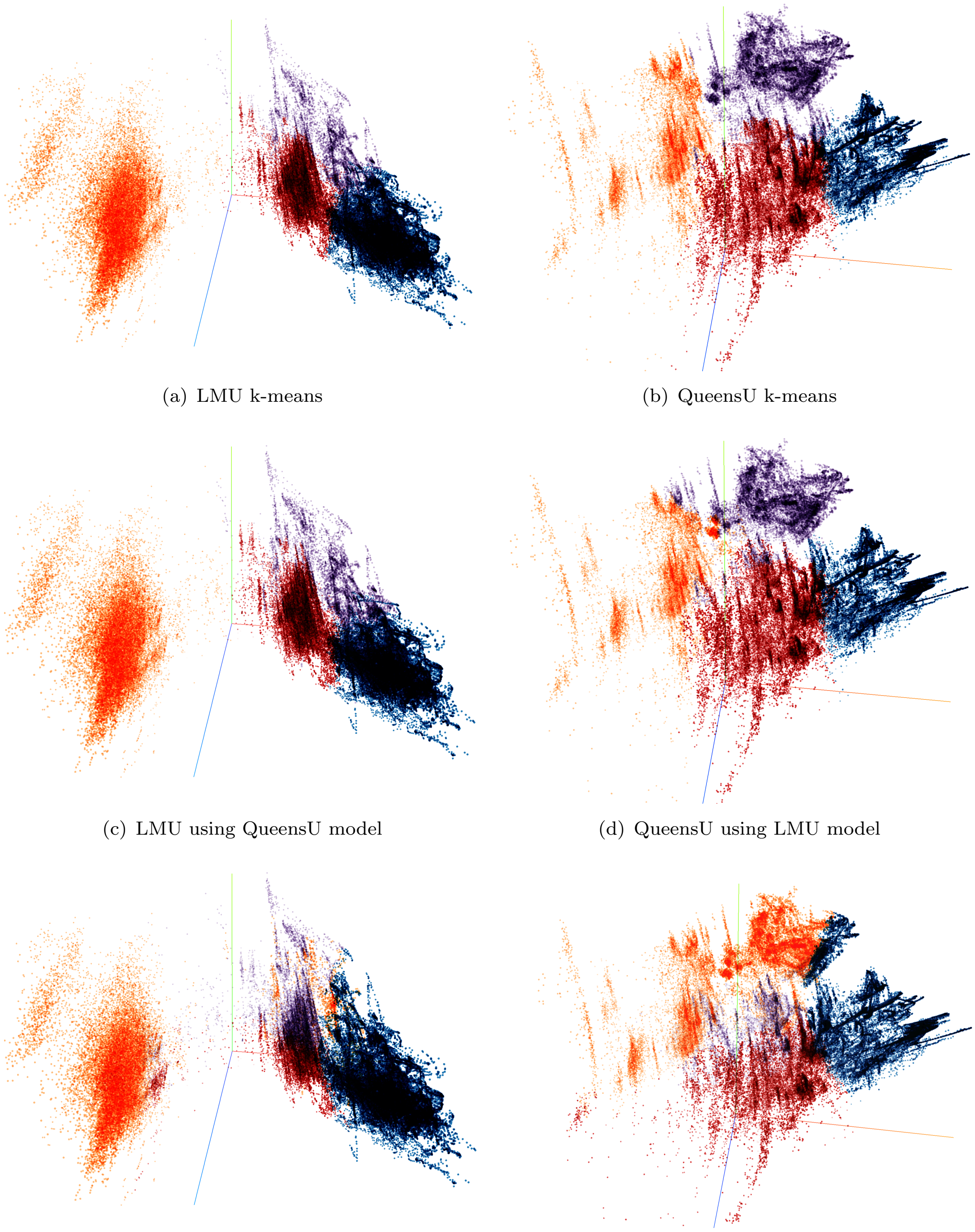


(b) LMU using TUM model

Figure 12. Predicted membership of cluster-4 on the LMU floor plan using models trained on the QueensU and TUM floor plan cluster membership dataset.

4.5. *Evaluation of Floor-Plan Similarity using Cluster Comparison Measures*

It is important to note that the calculation of accuracy values as shown in table 3 is not straight forward. The clustering algorithm k-means starts with a random set of seed points from which it starts building the clusters. As a consequence, the absolute value of the label itself (cluster-1, cluster-2, etc.) carries no semantic meaning. Multiple runs of the algorithm will lead to different outcomes. For example, points belonging to the large corridor on the LMU map will always be assigned to the same cluster, but this cluster will sometimes be labeled as cluster-1, sometimes as cluster-0, etc. This creates a problem when trying to compare the clustering results of different floor plans or even different models based on different clusterings of the same floor plan. The naive approach of trying to judge the quality of model predictions by calculating the accuracy of the predictions (in relation to a clustering outcome) will often fail when otherwise equal clusters are assigned different labels.



(a) LMU k-means (b) QueensU k-means

(c) LMU using QueensU model (d) QueensU using LMU model

(e) LMU using TUM model (f) QueensU using TUM model

Figure 13. Results of the PCA decomposition of the different floor plan data sets, using the 3 principal components that best explain the variance in the data as axes. Data points are colored either according to the k-means based clustering or according to model predictions as noted.

In section 3.7 we proposed to apply cluster comparison measures as a solution to this problem. By computing these comparison measures between the direct clustering results of the data points on the floor plans and the cluster memberships predicted by the different models, we are able to calculate a similarity score. We computed 6

Table 4. Adjusted Rand Index of cluster membership predictions of the different models against direct clustering results for each floor plan. An ARI score of 0.0 denotes random labeling, 1.0 shows identical cluster memberships.

Floor Plan	Model-QueensU	Model-Hotel	Model-LMU	Model-TUM	Mean score
QueensU	0.988	0.506	0.872	0.467	0.708
Hotel	0.754	0.995	0.743	0.796	0.822
LMU	0.931	0.763	0.99	0.683	0.842
TUM	0.57	0.422	0.42	0.996	0.602

different similarity scores for all combinations of floor plans and models. As we did not find any decisive difference between the results of the different measures, table 4 only shows the Adjusted Rand Index (ARI) (Hubert and Arabie 1985). The complete results for all measures are provided in the appendix.

As the ARI is symmetric, it does not matter whether the model predictions or clustering results are chosen as the *true clustering*. For reference, we also computed the similarity scores between the clustering results of a floorplan and a model trained on these results. As is to be expected, the similarity between the predictions and the cluster results is close to 1, e.g. 0.988 for the QueensU floor plan and the model trained on this floor plan. More interestingly, when comparing this ARI score to the score of the LMU model predictions (0.872) and the TUM model predictions (0.467) we find our previous visual similarity analysis to be confirmed. The PCA decomposition shown in figure 13 revealed a large overlap in the clustering results and LMU model predictions, but a lower overlap with the TUM model predictions. Both results match our perception that the LMU and QueensU floor plan are structurally more similar than the QueensU and TUM plans.

When comparing the aggregated ARI scores of all models, the LMU floor plan has the highest mean score of all plans: 0.842. In other words, the clusters found in the LMU building are predicted better by all models than the clusters on the other floor plans. This in turn can be interpreted as: the structures forming the LMU plan also appear in the other floor plans. We think this agrees well with our human judgement that of all analyzed floor plans, the LMU has the simplest recurring structures. The same reasoning also matches our perception in the inverse direction: The TUM floor plan has the lowest ARI mean score of 0.602, which tells us that the clusters predicted by the models differ most. This again matches our human perception that the TUM floor plan with it's irregular structures differs the most from the other floor plans.

5. Conclusion

We picked up and built upon the ideas of Sedlmeier and Feld (2018) who showed that it is possible to learn and recognize recurring structures in building floor plans. The presented framework allows the successful learning of recurring structures in building floor plans based on the numerical representation of visual perception. It contains three main functionalities. First, 3D environments can be used to create data sets containing geospatial trajectories that traverse the floor plans together with 2D isovist measures calculated at each time step along the trajectories. Second, unsupervised learning techniques can be used to group the data sets containing geospatial trajectories into meaningful clusters, based on visual perception features. Third, the now labeled data sets can be utilized by supervised learning techniques to automatically create models of recurring structures in the floor plans. These models can then be used to identify structures in unlabeled floor plans. Results show that isovist measures recorded along trajectories through the buildings do reflect the recurring structures found in them. These recurring patterns are encoded in the isovist measures in a way that unsupervised machine learning is able to identify meaningful clusters. Further, it was shown that these clustered data sets can also be used for neural network based supervised learning in order to create re-usable models which are able to identify structures in previously unknown environments. Good model accuracy results show, that the neural networks are able to learn functions which represent the underlying structures of the training data. The validation scores in turn show that the networks do not simply remember 1:1 mappings from input to output, but abstract general structures from the isovist measures that also fit the validation data. This becomes even more obvious in the inter-map validation step, where labeling was performed on floor plans of completely different environments, as the models were able to correctly label previously unseen inputs.

Besides extending the evaluation of the framework by including a larger amount of floor plans, this article extends the ideas by visual and numerical cluster similarity analysis. By doing this, we show that being able to learn and recognize structures of building floor plans, also makes it possible to judge the similarity or dissimilarity of the floor plans. By comparing the visualizations of cluster memberships predicted by models trained on different floor plans in map- and PCA feature-space, it is possible to reach a deeper understanding of the data's structure and differences between the floor plans. Besides this qualitative comparison of the visual perception similarity of building floor plans, we performed an algorithmic cluster similarity analysis, showing that it is a promising concept for quantifying visual perception similarity. The comparison of the similarity scores of different floor plan/model combinations with the results of the visual data analysis, showed that the scores match our human judgement of the floorplans' similarity.

To the best of our knowledge, such a numerical quantification of perceptual floor plan similarity based on unsupervised learning has not been shown before. It presents a valuable contribution making a shared awareness and common understanding of location and space, between machines, systems and their users, possible. This in turn can help in the development of improved location based services.

As future work we envision a deeper analysis of the generalization capacity of the models to new floor plans with different characteristics. Part of this would be to train "multi-map-cluster" models that combine the clustered data found on multiple floor plans. It will be interesting to see if these kind of models behave qualitatively differently from models trained only on a single floor plan. Furthermore, we would like to analyze 3D floor plans of buildings using 3D isovist measures. Besides this, it might be interesting to explore other neural network architectures like auto-encoders and recurrent neural networks to discover and learn further dynamic aspects encoded in the sequential nature of the isovist data.

References

Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2015. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." Software available from tensorflow.org, http://tensorflow.org/.

Ah-Soon, Christian, and Karl Tombre. 1997. "Variations on the analysis of architectural drawings." In *Document Analysis and Recognition, 1997., Proceedings of the Fourth International*

Conference on, Vol. 1, 347–351. IEEE.
Anguelov, Dragomir, Daphne Koller, Evan Parker, and Sebastian Thrun. 2004. "Detecting and modeling doors with mobile robots." In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, Vol. 4, 3777–3784. IEEE.
Balsamo, M.S., W. Knottenbelt, and A. Marin. 2013. *Computer Performance Engineering: 10th European Workshop, EPEW 2013, Venice, Italy, September 16-17, 2013, Proceedings.* Lecture Notes in Computer Science. Springer Berlin Heidelberg.
Benedikt, Michael L. 1979. "To take hold of space: isovists and isovist fields." *Environment and Planning B: Planning and design* 6 (1): 47–65.
Bhatia, Shashank, Stephan K Chalup, Michael J Ostwald, et al. 2012. "Analyzing architectural space: identifying salient regions by computing 3D isovists." In *Conference Proceedings. 46th Annual Conference of the Architectural Science Association (AN-ZAScA), Gold Coast, QLD*, .
Blender Foundation. 2017. *Blender - a 3D modelling and rendering package.* Blender Institute, Amsterdam: Blender Foundation. http://www.blender.org.
Buschka, Pär, and Alessandro Saffiotti. 2002. "A virtual sensor for room detection." In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, Vol. 1, 637–642. IEEE.
Chen, Guanling, and David Kotz. 2000. *A survey of context-aware mobile computing research.* Technical Report. Technical Report TR2000-381, Dept. of Computer Science, Dartmouth College.
Chen, Wei, Ting Qu, Yimin Zhou, Kaijian Weng, Gang Wang, and Guoqiang Fu. 2014. "Door recognition and deep learning algorithm for visual based robot navigation." In *Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on*, 1793–1798. IEEE.
Chollet, François, et al. 2015. "Keras." https://github.com/fchollet/keras.
De Smith, Michael John, Michael F Goodchild, and Paul Longley. 2007. *Geospatial analysis: a comprehensive guide to principles, techniques and software tools.* Troubador Publishing Ltd.
Deng, Li, Dong Yu, et al. 2014. "Deep learning: methods and applications." *Foundations and Trends® in Signal Processing* 7 (3–4): 197–387.
Dey, Anind K, and Gregory D Abowd. 1999. "Towards a better understanding of context and context-awareness." In *International Symposium on Handheld and Ubiquitous Computing*, 304–307. Springer.
Dogu, Ufuk, and Feyzan Erkip. 2000. "Spatial Factors Affecting Wayfinding and Orientation: A Case Study in a Shopping Mall." *Environment and Behavior* 32 (6): 731–755.
Dosch, Philippe, Karl Tombre, Christian Ah-Soon, and Gérald Masini. 2000. "A complete system for the analysis of architectural drawings." *International Journal on Document Analysis and Recognition* 3 (2): 102–116.
Emo, Beatrix. 2015. "Exploring isovists : the egocentric perspective." In *International Space Syntax Symposium*, 1–8.
Ester, Martin, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. "A density-based algorithm for discovering clusters in large spatial databases with noise." In *Kdd*, Vol. 96, 226–231.
Everitt, Brian, and Torsten Hothorn. 2011. *An introduction to applied multivariate analysis with R.* Springer Science & Business Media.
Feld, Sebastian, Steffen Illium, Andreas Sedlmeier, and Lenz Belzner. 2018. "Trajectory annotation using sequences of spatial perception." *26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* .
Feld, Sebastian, Hao Lyu, and Andreas Keler. 2017. "Identifying Divergent Building Structures Using Fuzzy Clustering of Isovist Features." In *Progress in Location-Based Services 2016*, 151–172. Springer.
Feld, Sebastian, Martin Werner, and Claudia Linnhoff-Popien. 2016. "Approximated Environment Features With Application to Trajectory Annotation." In *6th IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2016)*, .

Goerke, Nils, and Sven Braun. 2009. “Building semantic annotated maps by mobile robots.” In *Proceedings of the Conference Towards Autonomous Robotic Systems*, 149–156.

Haq, Saif, and Craig Zimring. 2003. “Just down the road a piece: the development of topological knowledge of building layouts.” *Environment and behavior* 35 (1): 132–160.

Hayward, Scott C, and Samuel S Franklin. 1974. “Perceived openness-enclosure of architectural space.” *Environment and Behavior* 6 (1): 37–52.

Hillier, Bill, and Julienne Hanson. 1984. *The social logic of space.* Cambridge university press.

Hubert, Lawrence, and Phipps Arabie. 1985. “Comparing partitions.” *Journal of classification* 2 (1): 193–218.

Jones, Eric, Travis Oliphant, Pearu Peterson, et al. 2001–. “SciPy: Open source scientific tools for Python.” [Online; accessed 2017-07-23], http://www.scipy.org/.

Kingma, Diederik, and Jimmy Ba. 2014. “Adam: A method for stochastic optimization.” *arXiv preprint arXiv:1412.6980* .

Küpper, Axel. 2005. “Location-based services.” *Fundamental and operation, John Willey & Sons, Ltd* .

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. “Deep learning.” *Nature* 521 (7553): 436–444.

Leonard, John J, and Hugh F Durrant-Whyte. 1991. “Simultaneous map building and localization for an autonomous mobile robot.” In *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, 1442–1447. Ieee.

Lloyd, Stuart. 1982. “Least squares quantization in PCM.” *IEEE transactions on information theory* 28 (2): 129–137.

Lowe, Ryan, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. “Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments.” *arXiv preprint arXiv:1706.02275* .

Lu, Tong, Huafei Yang, Ruoyu Yang, and Shijie Cai. 2007. “Automatic analysis and integration of architectural drawings.” *International Journal on Document Analysis and Recognition* 9 (1): 31–47.

Mordatch, Igor, and Pieter Abbeel. 2017. “Emergence of Grounded Compositional Language in Multi-Agent Populations.” *arXiv preprint arXiv:1703.04908* .

Mozos, Óscar Martínez. 2010. *Semantic labeling of places with mobile robots.* Vol. 61. Springer.

Mozos, Oscar Martinez, and Wolfram Burgard. 2006. “Supervised learning of topological maps using semantic information extracted from range data.” In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2772–2777. IEEE.

Pearson, Karl. 1900. “X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling.” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 50 (302): 157–175.

Pearson, Karl. 1901. “On lines and planes of closest fit to systems of points in space.” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2 (11): 559–572.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12: 2825–2830.

Rand, William M. 1971. “Objective criteria for the evaluation of clustering methods.” *Journal of the American Statistical association* 66 (336): 846–850.

Rashidi, Parisa, and Alex Mihailidis. 2013. “A survey on ambient-assisted living tools for older adults.” *IEEE journal of biomedical and health informatics* 17 (3): 579–590.

Ratner, Bruce. 2017. *Statistical and machine-learning data mining: Techniques for better predictive modeling and analysis of big data.* Chapman and Hall/CRC.

Raubal, Martin. 2002. “Wayfinding in built environments: The case of airports.” *IfGIprints* 14.

Rousseeuw, Peter J. 1987. “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." *Journal of computational and applied mathematics* 20: 53–65.

Samet, HANAN, and A Soffer. 1994. "Automatic interpretation of floor plans using spatial indexing." *Progress in Image Analysis and Processing* 3: 233.

Sedlmeier, Andreas, and Sebastian Feld. 2018. "Discovering and Learning Recurring Structures in Building Floor Plans." In *LBS 2018: 14th International Conference on Location Based Services*, 151–170. Springer.

Snook, Greg. 2000. "Simplified 3D Movement and Pathfinding Using Navigation Meshes." In *Game Programming Gems*, edited by Mark DeLoura, 288–304. Charles River Media.

Strehl, Alexander, and Joydeep Ghosh. 2002. "Cluster ensembles—a knowledge reuse framework for combining multiple partitions." *Journal of machine learning research* 3 (Dec): 583–617.

Tandy, CRV. 1967. "The isovist method of landscape survey." *Methods of Landscape Analysis* 9–10.

Triebel, Rudolph, Kai Arras, Rachid Alami, Lucas Beyer, Stefan Breuers, Raja Chatila, Mohamed Chetouani, et al. 2016. "Spencer: A socially aware service robot for passenger guidance and help in busy airports." In *Field and Service Robotics*, 607–622. Springer.

Unity Technologies. 2017. "Unity – Game Engine." http://unity3d.com. Accessed: 2017-07-22.

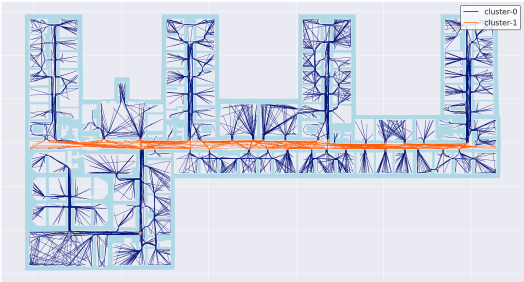
Vinh, Nguyen Xuan, Julien Epps, and James Bailey. 2010. "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance." *Journal of Machine Learning Research* 11 (Oct): 2837–2854.

Wagner, Silke, and Dorothea Wagner. 2007. *Comparing clusterings: an overview.* Universität Karlsruhe, Fakultät für Informatik Karlsruhe.

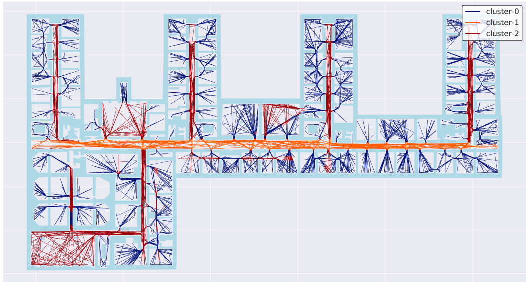
Weber, Markus, Christoph Langenhan, Thomas Roth-Berghofer, Marcus Liwicki, Andreas Dengel, and Frank Petzold. 2010. "a. SCatch: semantic structure for architectural floor plan retrieval." In *International Conference on Case-Based Reasoning*, 510–524. Springer.

Weiser, Mark. 1991. "The computer for the 21st century." *Scientific american* 265 (3): 94–104.

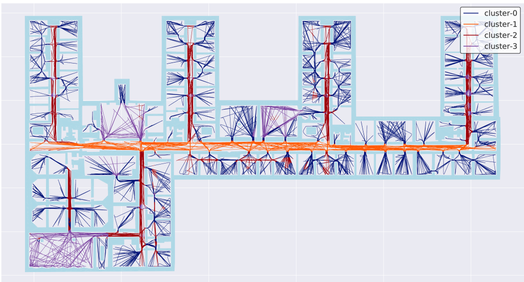
Appendix A. Map-space clustering results



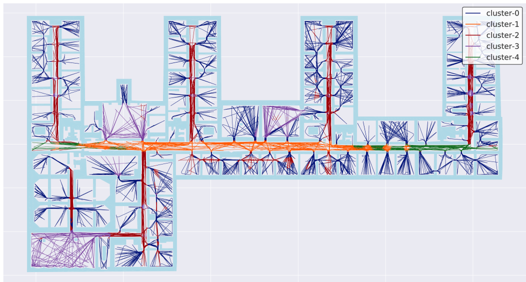
(a) $k = 2$



(b) $k = 3$

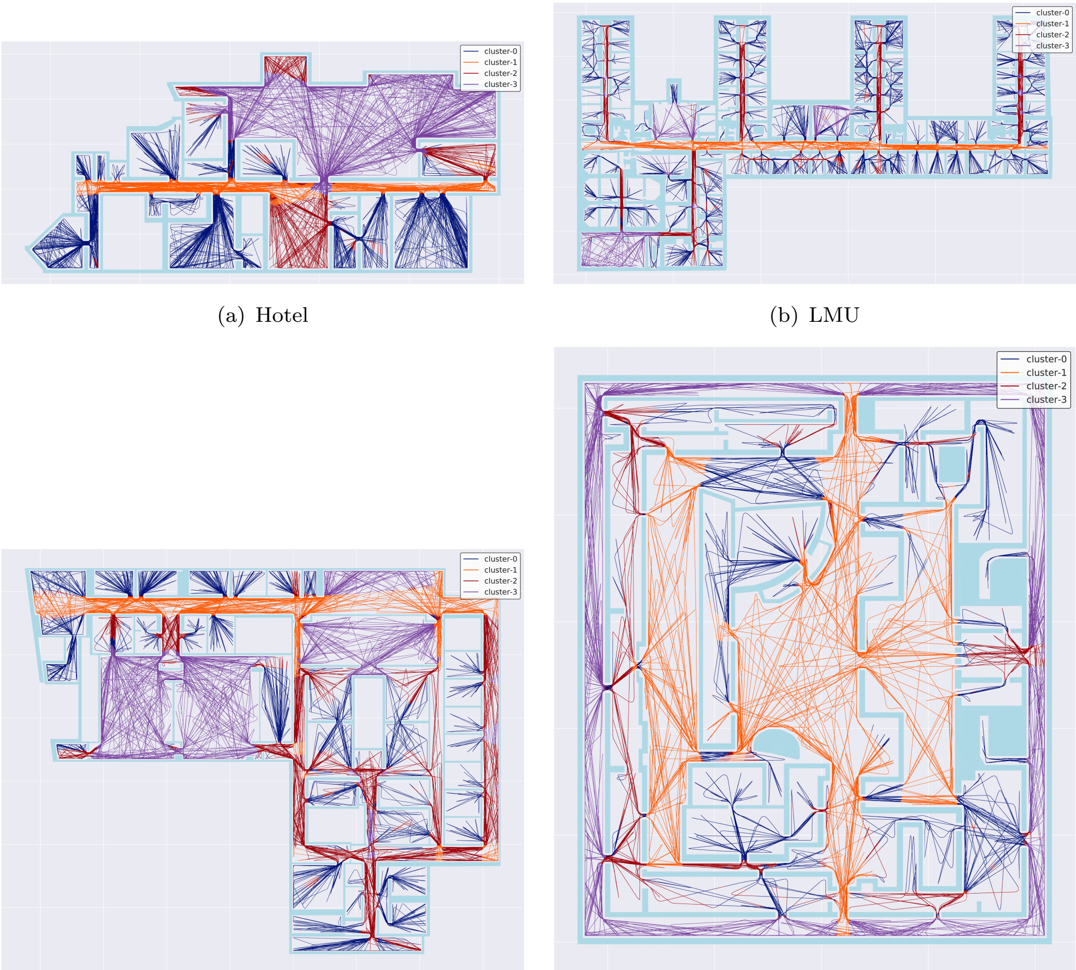


(c) $k = 4$



(d) $k = 5$

Figure A1. Results of k-means based clustering of static isovist measures on the LMU floor plan.



(a) Hotel

(b) LMU

(c) QueensU

(d) TUM

Figure A2. Results of k-means based clustering with $k = 4$ of static isovist measures on all floor plans.

Appendix B. Cluster Comparison Scores

Table B1. Floor plan: QueensU

Model	ARI	NMI	AMI	Homogeneity	Completeness	V-Measure
QueensU	0.987715	0.979196	0.987715	0.979493	0.9789	0.979196
Hotel	0.505622	0.558084	0.505622	0.555222	0.560961	0.558077
LMU	0.872009	0.833721	0.872009	0.832962	0.834481	0.833721
TUM	0.467313	0.539658	0.467313	0.524514	0.555239	0.539439

Table B2. Floor plan: Hotel

Model	ARI	NMI	AMI	Homogeneity	Completeness	V-Measure
QueensU	0.753856	0.752313	0.753856	0.734341	0.770726	0.752093
Hotel	0.995018	0.987989	0.995018	0.987918	0.988061	0.987989
LMU	0.742764	0.764612	0.742764	0.748026	0.781567	0.764429
TUM	0.795754	0.765496	0.795754	0.704661	0.831583	0.762879

Table B3. Floor plan: LMU

Model	ARI	NMI	AMI	Homogeneity	Completeness	V-Measure
QueensU	0.753856	0.752313	0.753856	0.734341	0.770726	0.752093
Hotel	0.995018	0.987989	0.995018	0.987918	0.988061	0.987989
LMU	0.742764	0.764612	0.742764	0.748026	0.781567	0.764429
TUM	0.795754	0.765496	0.795754	0.704661	0.831583	0.762879

Table B4. Floor plan: TUM

Model	ARI	NMI	AMI	Homogeneity	Completeness	V-Measure
QueensU	0.569887	0.6099	0.569887	0.593618	0.626629	0.609677
Hotel	0.421675	0.500498	0.421675	0.482868	0.518771	0.500176
LMU	0.419935	0.49122	0.419935	0.484377	0.498159	0.491172
TUM	0.996486	0.990757	0.996486	0.990625	0.990889	0.990757