Discovering and Learning Recurring Structures in Building Floor Plans

Andreas Sedlmeier and Sebastian Feld

Abstract Autonomous mobile robots show promising opportunities as concrete use cases of location-based services. Such 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 these perceptions 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. Combining these topics will enable the creation of new and better location-based services which have a deep awareness of their surroundings. This paper presents a framework to create a data set containing 2D isovist measures calculated along geospatial trajectories that traverse a 3D simulation environment. Furthermore, we show 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 is able to identify meaningful structures like rooms, hallways and doorways. These labeled data sets can further be used for neural network based supervised learning. The models generated this way do generalize and are able to identify structures in different environments.

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*

A. Sedlmeier · S. Feld (✉)
Mobile and Distributed Systems Group, LMU Munich, Munich, Germany
e-mail: sebastian.feld@ifi.lmu.de

A. Sedlmeier
e-mail: andreas.sedlmeier@ifi.lmu.de

P. Kiefer et al. (eds.), *Progress in Location Based Services 2018*, Lecture Notes
in Geoinformation and Cartography, https://doi.org/10.1007/978-3-319-71470-7_8

(Weiser et al. 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), 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 simple tasks like the transportation of packets in storehouses. Further examples of research are mobile robots that lead tourist groups through an airport (Triebel et al. 2016) or serving 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 are numerous empirical and experimental studies 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 (Smith et al. 2007). The most basic term in this context is *Space Syntax*, summarizing mostly the acquisition of topological structures of an environment without geometric measurements (Hillier and Hanson 1984). A further concept in this area called *Isovist* has been introduced in (Tandy 1967) and describes the set of points in space that are visible from a specific vantage point. Based on this idea there have been presented a formal definition of isovists together with some analytical measures enabling a quantitative description of a spatial environment (Benedikt 1979).

Even if 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 problem area in which huge progress was made in the last few years thanks to the advances in the field of deep learning with neural networks. Deep neural networks excel at the recognition of recurring structures in large data sets and the inference of underlying functions generating these structures.

The main idea of this paper is to investigate, whether the recurring structures inside buildings also have recurring isovist measures and whether such numerical features can be used to learn a model of such structures. Specifically this means that we incorporate unsupervised machine learning techniques of visual perception features to label a dataset consisting of geospatial trajectories through floor plans. This labeled dataset is utilized by supervised machine learning techniques to predict labels, thus structures, in unknown environments. The general use case of this idea is to create advanced spatial context for location-based services. A more 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 idea presented in this paper, a mobile robot would be able to independently learn a model of recurring structures inside buildings like, for example, rooms, hallways, and doorways. This model can then be reused in unknown buildings to recognize learned structures straight away. Alternative use cases are the off-line analysis or annotation of floor plans or the incorporation in computer games, such that non-player characters gain an additional understanding of altering surroundings.

The contributions of this paper are twofold. First, we present a framework that is able to automatically generate input data for learning a model of recurring structures inside buildings based on floor plans. The framework builds on the game engine *Unity* developed by Unity Technologies (Unity 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 of the data set. Second, we present a framework of machine learning techniques that can be used to train a model of recurring structures inside floor plans and to recognize such structures in unknown floor plans. We built 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 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: Sect. 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, and prediction of recurring structures in floor plans. In Sect. 4 we present our experimental results and present a detailed discussion. Section 5 concludes this paper.

2 Background and Related Work

This section contains the technical background for the further understanding of this paper together with related work. First, techniques for the analysis of visual perception as well as machine learning techniques are described. These are the main ingredients for the automatic generation of a model of recurring structures inside buildings. Furthermore, related work with respect to semantic annotation of floor plans are illustrated.

2.1 Analysis of Visual Perception

As already mentioned in Sect. 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 in Tandy (1967) and more formally defined in 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 in 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 in Benedikt (1979) as the set of isovists along a trajectory, or more complete, the set of isovists at all places of an investigated environment. Since a human is 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 calculate 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 potentially get complicated, since all corner points of visible walls and objects have to be determined and connected. Feld et al. (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 the generation of knowledge from experience. During a training phase a systems learns from examples and is able to make generalizations 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 infer 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 modified until a certain error function is minimized. *k-means* is a widely used 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). The algorithm has got two parameters: 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.

A **supervised learning** algorithm, by contrast, tries to infer a function based on given pairs of input and corresponding known output labels. The idea is to train the system in order to create associations. In our case the input data again consists of a large set of time series of isovist measures calculated along geospatial trajectories. However, for each data point, there is a corresponding known ground truth, for example: "at this point the agent resides inside a room" or "at this point the agent traverses a doorway". A popular use case of supervised learning, where 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 and yu et al. 2014). Given enough input data and the right structure, neural networks are able to learn arbitrary functions from labeled data sets. Using images of known classes, a model is trained that infers a function determining class boundaries. Afterwards, one can use this model to predict the classes of unknown images, or in other words: Observations that have not been used during the training phase. In our case a model is trained on floor plans where rooms, hallways, and doorways are known. Afterwards, this model can be used on unlabeled floor plans where no such information is available.

2.3 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. This is 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 tries to create a corresponding floor plan. Concurrently, the robot has to position itself. Thus, it makes sense to enrich the map just created with semantic information. The basic assumption is that the robot's perception, in most cases laser range scans, contains enough information about the environment. Basically, we indirectly 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) and 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 in 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 et al. (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 et al. (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.

3 Methodology

This section is split in two 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 step, (2) the discovery, learning, and prediction of recurring structures in floor plans. Unsupervised learning techniques are employed in the discovery phase, while supervised machine learning is performed for the modeling and prediction tasks. Details regarding the exact implementation of these aspects can be found in Sect. 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 (blender.org 2017), an open-source 3D computer graphics software, 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 (2017), a 3D game engine and development environment. 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 Fig. 1.

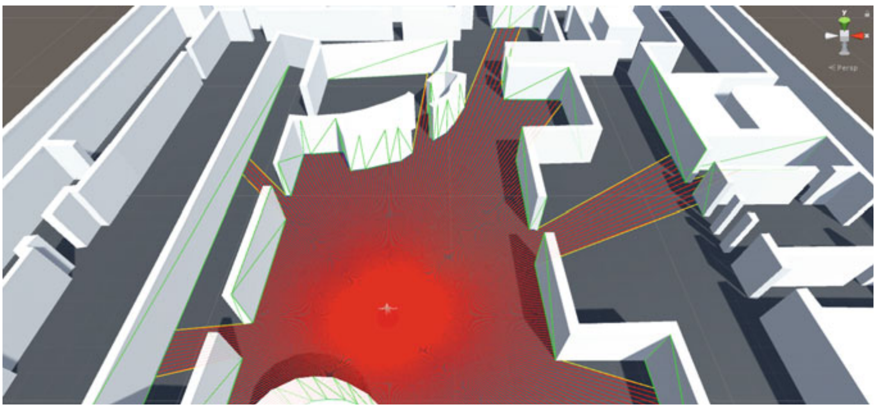

Fig. 1 3D view of a utilized floor plan, showing the non-player character casting 360 rays (red lines) from it's current vantage point

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 Sect. 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.

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

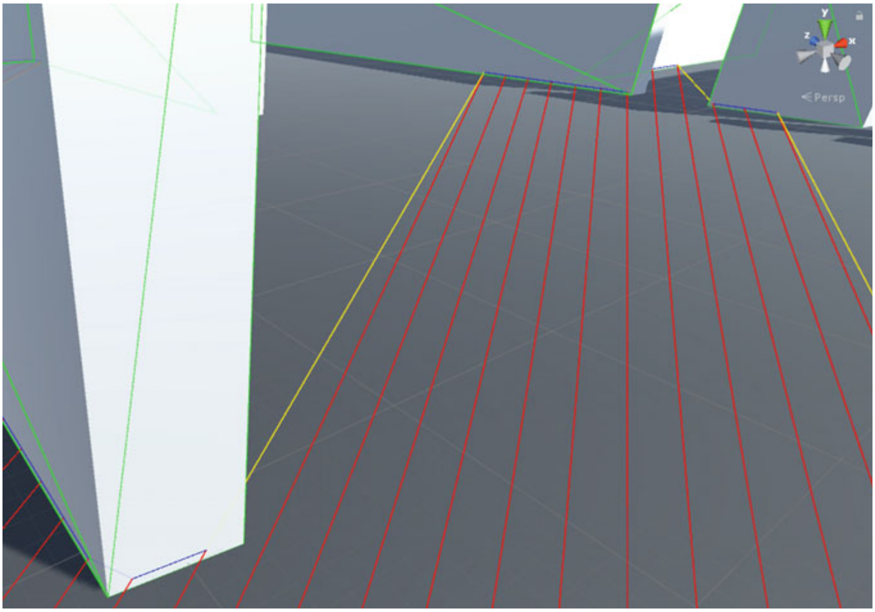

Fig. 2 In-engine view of the custom built algorithm's results for real-surface and occlusion isovist measure calculation. Red lines are the rays cast from the current vantage point, green lines visualize the meshes' triangles hit by the rays, blue lines denote real-surface while yellow lines denote occlusion

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 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. 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.

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. This means that X and Y coordinates are removed so that a 6-dimensional vector remains, whereas each dimension represents one of the measured isovist features. 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). The algorithm determines a configurable amount of cluster centers and assigns the data points to the nearest cluster center, by minimizing the squared distances from the clusters.

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 was not reflected in the analysis. The overarching idea of the next 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 et al. 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.

3.3 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 inferring a function which maps new unlabeled input data points to the respective cluster categories. This model can then be used, for example, in robots as a lightweight component enabling the robot to deduce 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 feedforward neural network using the open source neural network library Keras (Chollet 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, based on the data points' coordinates. 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 et al. 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 9,412 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 captured 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.

4 Results and Discussion

For the evaluation, two distinct floor plans were chosen. The first one is a section of a university building of the Ludwig-Maximilians-Universität München (LMU). It features repeating structures of corridors and similar rooms. The second floor plan features the main hall and connected rooms of the Technische Universität München (TUM). By comparison, it contains a more irregular structure formed by large lecture halls and connecting hallways. Because of it's distinct and repeating structures, the

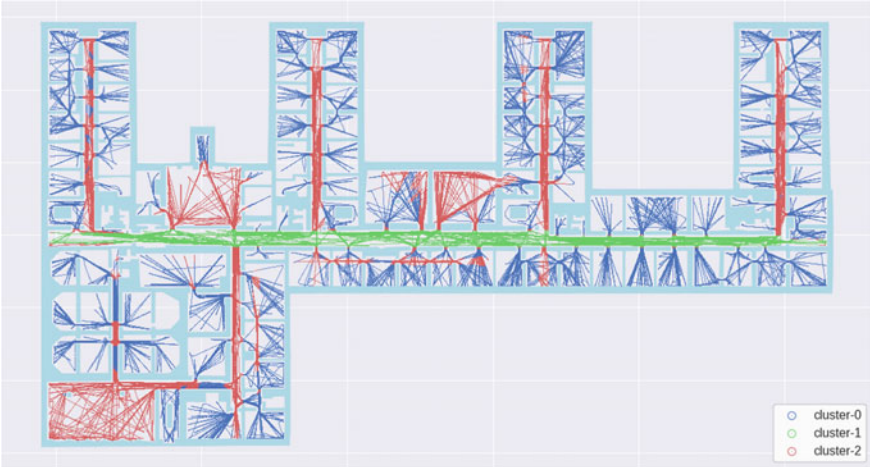

Fig. 3 Results of a k-means based clustering with $k = 3$ of static isovist measures on the LMU floor plan

LMU floor plan was chosen for training. We recorded more than 370,000 isovists along random trajectories on the LMU map and more than 220,000 isovists on the TUM map.

We compared two different unsupervised learning methods: k-means, a 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. From all the values tested, we found a value of $k = 3$ to produce the most meaningful results on the LMU floor plan. As can be seen in Fig. 3, three different structures of the floor plan are separated into different clusters. A clear separation between the large horizontal corridor, the smaller vertical corridors, and rooms became 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 between 1000 and 2500 produced meaningful results.

Compared to the clusters produced by the k-means algorithm, the resulting structures were less interpretable. 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 floors, the delta of the current data point and the SMA of the isovist measures was calculated, in order to capture the temporal dimension of the data.

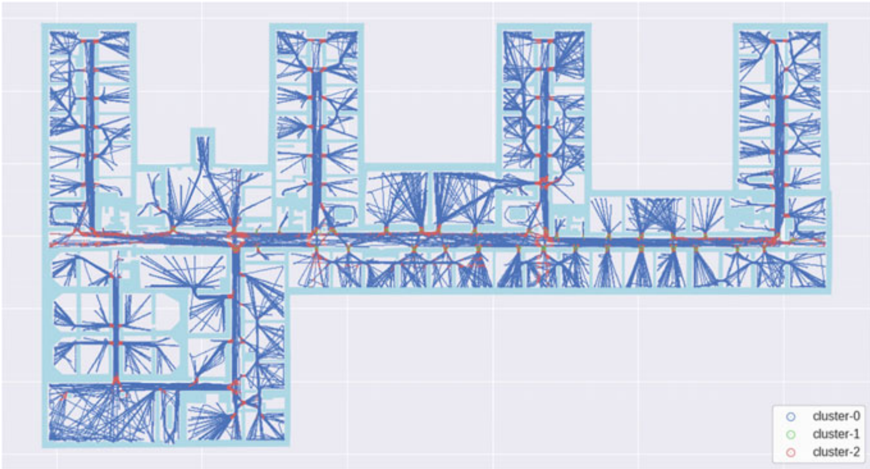

Fig. 4 Results of k-means based clustering of dynamic, SMA based isovist measures ($k = 3, c = 5$)

Using these "delta-features" as input to the k-means clustering, a completely different picture became visible: As can be seen in Fig. 4, 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 four different clusters as shown in Fig. 5.

We interpret the clusters as follows:

Cluster-0 (blue): rooms
Cluster-1 (green): horizontal corridors
Cluster-2 (red): small vertical corridors and large rooms
Cluster-3 (purple): passage ways (e.g. doors)

Figure 5 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 and SMA deltas) as input and the 4 cluster labels described above as targets. The best model showed an accuracy value of 0.9856 and validation accuracy value of 0.9188.

As can be seen in Fig. 6, the predictions produced by this model on the validation data match our interpretations of the clusters in the training data. This means, that the model was able to learn a function representing the structures bundled in the

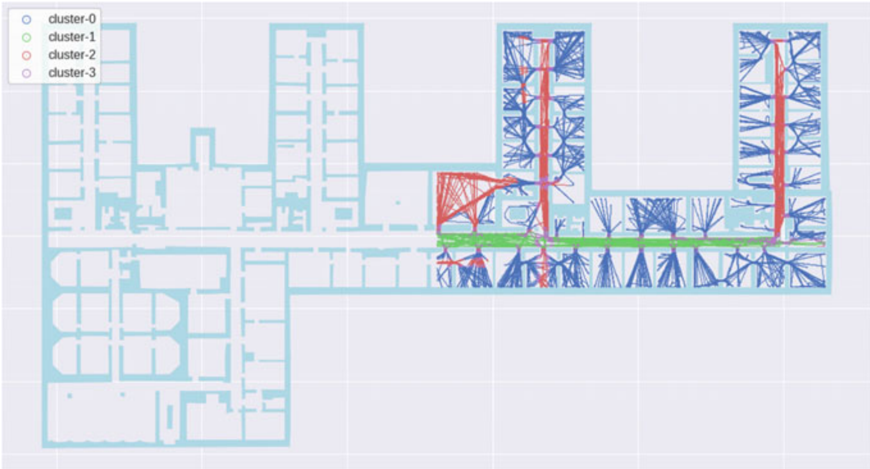

Fig. 5 Results of training/validation data split after merging static and dynamic cluster features. Only training data is visualized in this figure

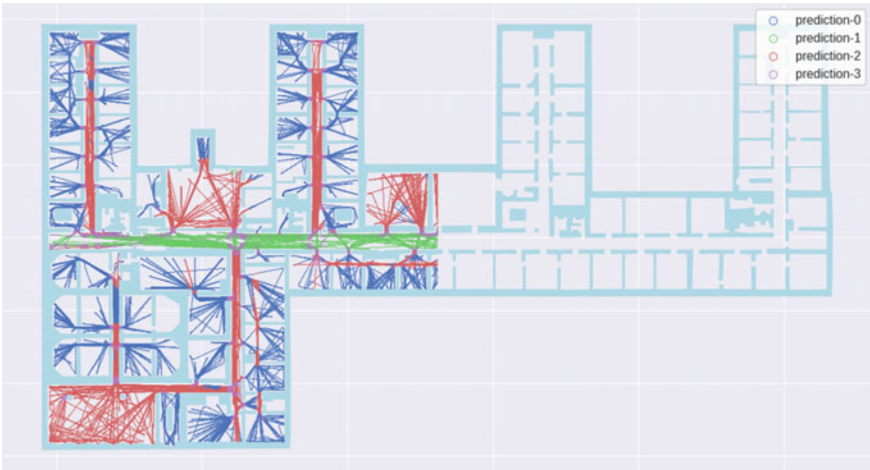

Fig. 6 Predictions of the 5-layer feedforward neural network on the validation data set

respective clusters. As it was able to predict meaningful results on previously unseen parts of the floor plan, it became obvious that the model generalizes to new data.

Figure 6 also shows that our previous interpretation of cluster-2 to be mainly comprised of small vertical corridors no longer holds. The small red colored corridor formed by the doors of interconnected rooms in the lower right part of the floor plan is clearly horizontal in nature. Apart from that, our previous interpretations of the clusters still hold.

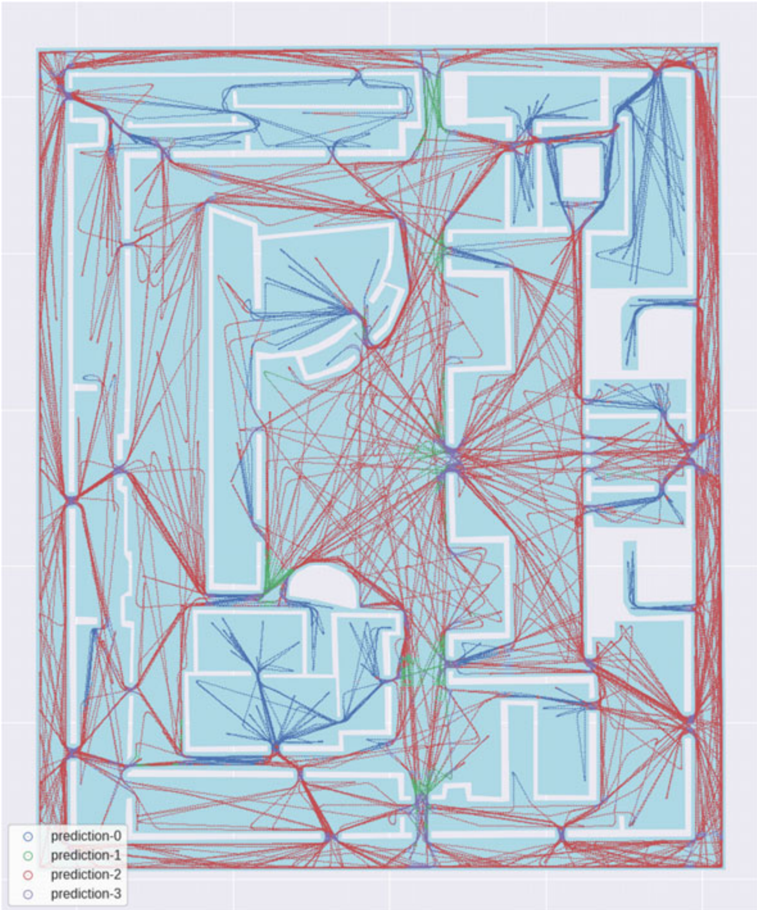

Fig. 7 Predicted cluster memberships of all data points along random trajectories trough the TUM floor plan using the model trained on the LMU floor plan cluster membership dataset with 12 features (static and dynamic)

In order to test the generalization performance of our model even further, we used it to predict the cluster memberships on a completely different floor plan. As the room and corridor structures in the TUM plan are completely different from the LMU plan, on which the model was trained, it is a much more difficult task for the model to perform. Figure 7 shows the predicted cluster-memberships of all recorded points along the random trajectories.

Even though the maps have completely different layouts, a visual comparison with our previous cluster definitions provided a good match. Points on the trajectories in smaller rooms are almost completely predicted to belong to cluster-0 (blue), as they did on the LMU floor plan. Large rooms and corridors predicted to be cluster-2 (red) also match our expectations. As there is no directly corresponding structure to the

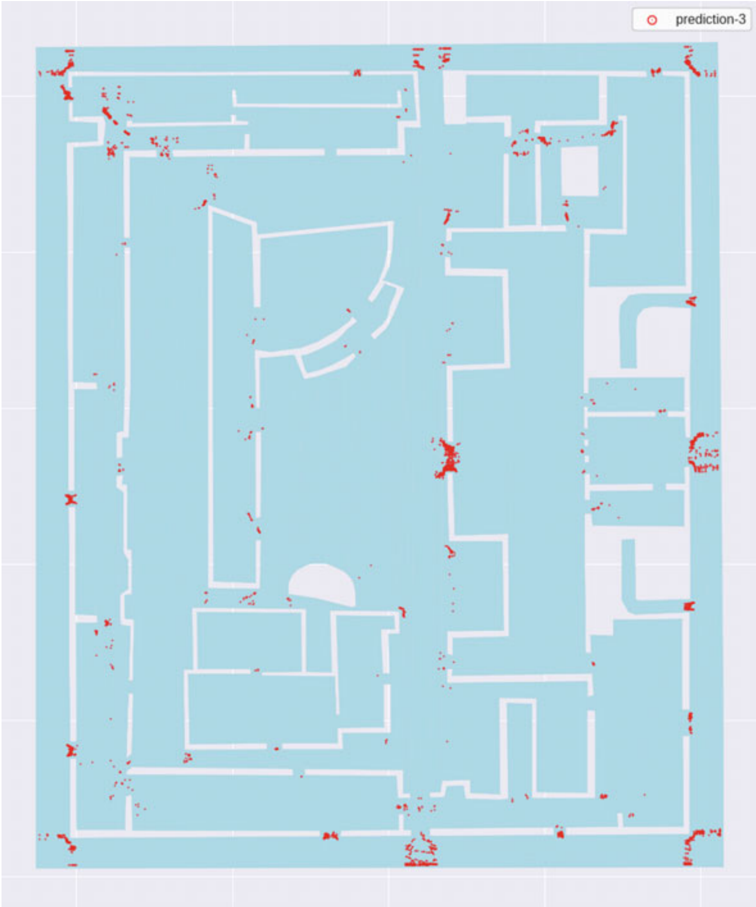

Fig. 8 Predicted membership of cluster-3 (passage ways and doors) using a confidence threshold of 99.99% on the TUM floor plan using the model trained on the LMU floor plan cluster membership dataset with 12 features (static and dynamic)

single large corridor (cluster-1: LMU) on the TUM floor plan, it is no surprise that labelings of cluster-1 (green) do not lend themselves to intuitive interpretations.

Most interestingly, structures labeled as cluster-3 (purple) match our definition of passage ways and doors almost perfectly. This becomes even more apparent when combined with a prediction confidence threshold. This is possible, because the output layer of the neural network does not produce binary label decisions but instead numeric values denoting the confidence that the current data point belongs to the respective label.

After increasing the threshold to 99.99%, doors and passage ways are marked largely correct, as can be seen in Fig. 8. What becomes apparent is that our definition of cluster-3 to contain only passage ways and doors might need to be expanded when

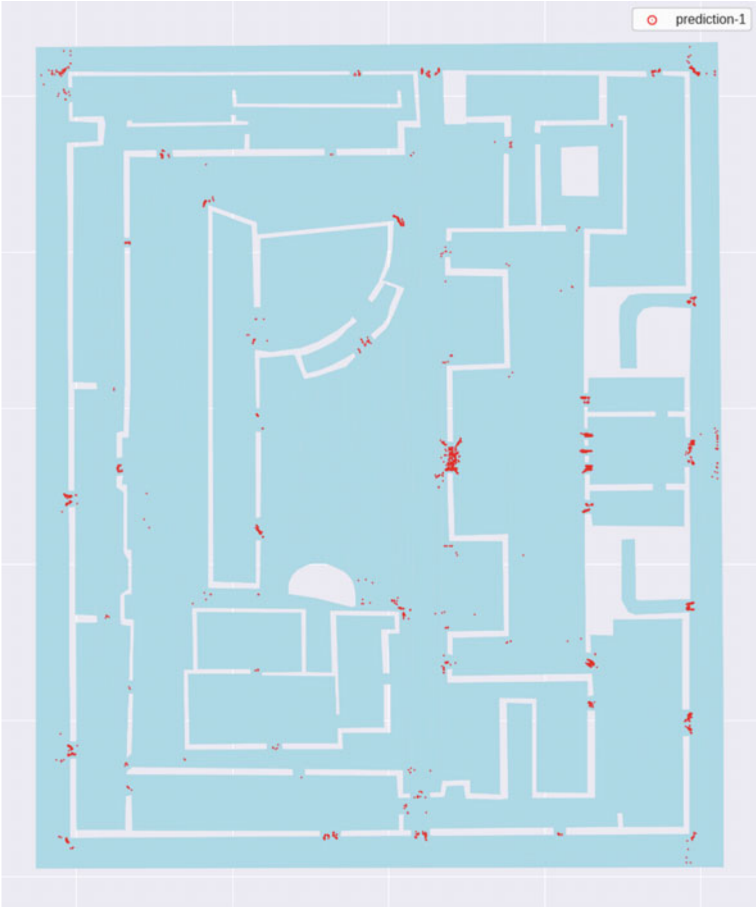

Fig. 9 Predicted cluster membership with confidence threshold of 99.95% on the TUM floor plan by a model trained using only the 6 dynamic SMA features on the LMU floor plan

applied to this floor plan. A more accurate description could be: Points where a different structural part of the building is entered. This structural part can, but need not, be explicitly separated by a door.

As a last step, we evaluated whether the model performance for this specific prediction of passage ways and doors could be further improved by excluding possibly irrelevant features for this task from the training data. A separate model having a smaller, 6-neuron input layer to take in only the 6 dynamic SMA features was trained on the LMU floor plan. The resulting predictions of now only two clusters are shown in Fig. 9. We think that these predictions fit our expectations even better, as the placement of labels is now more accurately inside the doors.

5 Conclusion

This paper is based on the idea that recurring structures inside buildings also show recurring structures in the numerical representation of the visual perception when traversing them. We presented a framework that contains three main functionalities. First, a 3D environment can be used to create a data set containing geospatial trajectories that traverse the floor plan together with 2D isovist measures calculated at each time step along the trajectories. Second, unsupervised learning techniques can be used to group the data set containing geospatial trajectories into meaningful clusters, based on visual perception features. Third, the now labeled data set can be utilized by supervised learning techniques to automatically create a model of recurring structures in the floor plan. This model can then be used to identify structure in unlabeled floor plans.

Our results show that isovist measures recorded along trajectories through the building do reflect the recurring structures found in buildings. These recurring patterns are encoded in the isovist measures in a way that unsupervised machine learning is able to identify meaningful clusters. Further, we were able to show that these clustered data sets can also be used for neural network based supervised learning in order to create a re-usable model which is able to identify structures in previously unknown environments. Good model accuracy results show, that the neural network is able to learn a function which represents the underlying structure of the training data. The validation score in turn shows that the network does not simply remember a 1:1 mapping from input to output, but abstracts general structures from the isovist measures that also fit the validation data. This becomes obvious in the validation step, where labeling was performed on the map of a completed different environment, as the network was able to correctly label previously unseen inputs.

As future work we envision a deeper analysis of the generalization capacity of the models to new floor plans with different characteristics, containing e.g. curved walls. We also plan to replace the ray-based isovist measures by exact isovist measures following the definition given in Benedikt (1979) in order to increase our data accuracy. Furthermore, we would like to analyze 3D floor plans of buildings using 3D isovist measures. Finally, extensive feature engineering can be conducted and different neural network architectures explored in order to improve model accuracy and generalization performance.

References

blender.org (2017) Home of the blender project—free and open 3d creation software. https://www.blender.org/. Accessed 23 July 2017

Unity (2017) Game engine. http://unity3d.com. Accessed 22 July 2017

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Mané D, Monga R, Moore S, MurrayD, Olah C, Schuster M, Shlens J, Steiner

B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous systems. http://tensorflow.org/
Ah-Soon C, Tombre K (1997) Variations on the analysis of architectural drawings. In: Proceedings of the fourth international conference on document analysis and recognition, 1997, vol 1. IEEE, pp 347–351
Anguelov D, Koller D, Parker E, Thrun S (2004) Detecting and modeling doors with mobile robots. In: ICRA'04. 2004 IEEE international conference on robotics and automation, 2004. Proceedings, vol 4. IEEE, pp 3777–3784
Balsamo M, Knottenbelt W, Marin A (2013) Computer performance engineering: 10th European workshop, EPEW 2013, Venice, Italy, September 16–17, 2013, Proceedings. Lecture notes in computer science. Springer, Berlin
Benedikt ML (1979) To take hold of space: isovists and isovist fields. Environ Plan B: Plan Des 6(1):47–65
Bhatia S, Chalup SK, Ostwald MJ 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
Buschka P, Saffiotti A (2002) A virtual sensor for room detection. In: IEEE/RSJ international conference on intelligent robots and systems, 2002, vol 1. IEEE, pp 637–642
Chen G, Kotz D (2000) A survey of context-aware mobile computing research. Technical report TR2000-381, Dept of Computer Science, Dartmouth College
Chen W, Qu T, Zhou Y, Weng K, Wang G, Fu G (2014) Door recognition and deep learning algorithm for visual based robot navigation. In: 2014 IEEE international conference on robotics and biomimetics (ROBIO). IEEE, pp 1793–1798
Chollet F et al (2015) Keras. https://github.com/fchollet/keras
De Smith MJ, Goodchild MF, Longley P (2007) Geospatial analysis: a comprehensive guide to principles, techniques and software tools. Troubador Publishing Ltd
Deng L, Yu D et al (2014) Deep learning: methods and applications. Foundations and trends®. Signal Proces 7(3–4):197–387
Dey AK, Abowd GD (1999) Towards a better understanding of context and context-awareness. In: International symposium on handheld and ubiquitous computing. Springer, pp 304–307
Dogu U, Erkip F (2000) Spatial factors affecting wayfinding and orientation: a case study in a shopping mall. Environ Behav 32(6):731–755
Dosch P, Tombre K, Ah-Soon C, Masini G (2000) A complete system for the analysis of architectural drawings. Int J Doc Anal Recogn 3(2):102–116
Emo B (2015) Exploring isovists: the egocentric perspective. In: International space syntax symposium, pp 1–8
Ester M, Kriegel HP, Sander J, Xu X et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. Kdd 96:226–231
Feld S, Lyu H, Keler A (2017) Identifying divergent building structures using fuzzy clustering of isovist features. In: Progress in location-based services. Springer, pp 151–172
Feld S, Werner M, Linnhoff-Popien C (2016) Approximated environment features with application to trajectory annotation. In: 6th IEEE symposium series on computational intelligence (IEEE SSCI 2016)
Goerke N, Braun S (2009) Building semantic annotated maps by mobile robots. In: Proceedings of the conference towards autonomous robotic systems, pp 149–156
Haq S, Zimring C (2003) Just down the road a piece: the development of topological knowledge of building layouts. Environ Behav 35(1):132–160
Hayward SC, Franklin SS (1974) Perceived openness-enclosure of architectural space. Environ Behav 6(1):37–52
Hillier B, Hanson J (1984) The social logic of space. Cambridge University Press
Jones E, Oliphant T, Peterson P et al (2001) SciPy: open source scientific tools for python. http://www.scipy.org/. Accessed 23 July 2017

Kingma D, Ba J (2014) Adam: a method for stochastic optimization. arXiv:1412.6980
Küpper A (2005) Location-based services. Fundamental and operation. Willey,
LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
Leonard JJ, Durrant-Whyte HF (1991) Simultaneous map building and localization for an autonomous mobile robot. In: IEEE/RSJ international workshop on intelligent robots and systems' 91. Intelligence for mechanical systems, Proceedings IROS'91. IEEE, pp 1442–1447
Lloyd S (1982) Least squares quantization in pcm. IEEE Trans Inf Theory 28(2):129–137
Lowe R, Wu Y, Tamar A, Harb J, Abbeel P, Mordatch I (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. arXiv:1706.02275
Lu T, Yang H, Yang R, Cai S (2007) Automatic analysis and integration of architectural drawings. Int J Doc Anal Recogn 9(1):31–47
Mordatch I, Abbeel P (2017) Emergence of grounded compositional language in multi-agent populations. arXiv:1703.04908
Mozos ÓM (2010) Semantic labeling of places with mobile robots, vol 61. Springer
Mozos OM, Burgard W (2006) Supervised learning of topological maps using semantic information extracted from range data. In: 2006 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 2772–2777
Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in python. J Mach Learn Res 12:2825–2830
Rashidi P, Mihailidis A (2013) A survey on ambient-assisted living tools for older adults. IEEE J Biomed Health Inform 17(3):579–590
Raubal M (2002) Wayfinding in built environments: the case of airports. IfGIprints 14
Samet H, Soffer A (1994) Automatic interpretation of floor plans using spatial indexing. Prog Image Anal Process 3:233
Snook G (2000) Simplified 3d movement and pathfinding using navigation meshes. In: DeLoura M (ed) Game programming gems. Charles River Media, pp 288–304
Tandy C (1967) The isovist method of landscape survey. Methods of landscape analysis, pp 9–10
Triebel R, Arras K, Alami R, Beyer L, Breuers S, Chatila R, Chetouani M, Cremers D, Evers V, Fiore M et al (2016) Spencer: a socially aware service robot for passenger guidance and help in busy airports. In: Field and service robotics. Springer, pp 607–622
Weber M, Langenhan C, Roth-Berghofer T, Liwicki M, Dengel A, Petzold F (2010) a. SCatch: semantic structure for architectural floor plan retrieval. In: International conference on case-based reasoning. Springer, pp 510–524
Weiser M (1991) The computer for the 21st century. Sci Am 265(3):94–104