International Cartographic Association
Association Cartographique Internationale

Towards Urban Environment Familiarity Prediction

Lukas Gokl[a], Marvin Mc Cutchan[a], Bartosz Mazurkiewicz[a], Paolo Fogliaroni[a, b], Ioannis Giannopoulos[a]*

[a] *Research Group Geoinformation, Vienna University of Technology, Austria, [firstname.lastname]@geo.tuwien.ac.at*
[b] *ESRI R&D Center Vienna, Austria, pfogliaroni@esri.com*

* Corresponding Author

Abstract:
Location Based Services (LBS) are definitely very helpful for people that interact within an unfamiliar environment, but also for those that already possess a certain level of familiarity with it. In order to avoid overwhelming familiar users with unnecessary information, the level of details offered by the LBS shall be adapted to the level of familiarity with the environment: providing more details to unfamiliar users and a lighter amount of information (that would be superfluous, if not even misleading) to the users that are more familiar with the current environment. Currently, the information exchange between the service and its users is not taking into account familiarity. Within this work, we investigate the potential of machine learning for a binary classification of environment familiarity (i.e., familiar vs unfamiliar) with the surrounding environment. For this purpose, a 3D virtual environment based on a part of Vienna, Austria was designed using datasets from the municipal government. During a navigation experiment with 22 participants we collected ground truth data in order to train four machine learning algorithms. The captured data included motion and orientation of the users as well as visual interaction with the surrounding buildings during navigation. This work demonstrates the potential of machine learning for predicting the state of familiarity as an enabling step for the implementation of LBS better tailored to the user.

Keywords: environment familiarity, machine learning, virtual environment

1. Introduction

Location Based Services (LBS) are influencing the way people interact with each other as well as with their surrounding environment and there are still several challenges that have to be overcome (Huang et al., 2018). For instance, knowing if a human is familiar with her surrounding environment is a very relevant topic as it enables to improve the quality of the provided service. Since it is practically impossible to ask every user explicitly about her familiarity with her surrounding environment, it is important to be able to predict it based on objectively measurable factors in an implicit manner. Geospatial data, such as user location and orientation, can be easily captured through mobile devices such as smart phones which are used for LBS. Rising computational and data transfer capabilities, especially in hand-held devices, enable to process data in real-time in the background without disturbing the users. The information generated can then be used by an LBS on the device in order to obtain a more accurate understanding of the user and her intentions, improving the quality of the provided service. For instance, a navigation system could adapt the visualizations of the route or the flow of information according to whether the user is familiar with the surroundings or not.

In this work, it is analyzed how the state of familiarity can be predicted based on the motion of a user as well as based on the interaction with the surrounding environment. For this purpose, a 3D-model was created using data from the regional government of Vienna, Austria, obtained through their geodataviewer[1], and the software *CityEngine* from Esri. The CityEngine software was utilized to create the shapes of the buildings according to the actual ground shape and height. As no facade graphics of the actual buildings were available, textures provided by CityEngine were used to produce a realistic appearance. This model was integrated into the *Unity* game engine in order to enable navigation. A user experiment with 22 participants was performed in order to collect ground truth data which were used to train four machine learning algorithms.

This work demonstrates the potential of familiarity prediction based on machine learning even with basic measures. The paper is structured as follows: we continue with related work and introduce the applied methodology followed by the results. Next, the findings are discussed and we close with a conclusion and objectives for future work.

2. Related work

Virtual environments have often been employed for empirical experiments for many different types of research questions. Experiments in laboratory environments are often favored due to the experimental control that can be achieved. Furthermore, several studies, e.g., (Anderson and Bushman, 1997, Kuliga et al., 2015), have demonstrated that laboratory studies externalize quite well. In a related experiment Chao Li (Li, 2006) implemented an immersive urban virtual reality environment to analyze behavior and information preferences of participants. The experiment setup was similar to the setup used in our work. He created a virtual environment that was based on the data of a real location, in this case a traditional market town in the United Kingdom. In his study, participants also used a joystick to move within the virtual environment. They

[1] https://www.wien.gv.at/ma41datenviewer/public/

were required to wear stereo glasses and had a PDA (Personal Digital Assistant) for accessing routing information. None of the participants was familiar with the test environment, nor did they get the chance to explore it beforehand. During the experiment, the time, location and rotation of each participant was automatically tracked. This specific experiment was focusing on the behavior of the participants and therefore the use of the PDA device was recorded. The collected datasets were later integrated to form a common time series over all different data types. This time series was then processed to gain information about how often, when and where the participants used the PDA and which information type they accessed. The users were additionally classified based on what kind of information they mostly accessed on the PDA into three groups: The first was dominated by the use of text route information, the second by low use of map information and the third by high use of map information. For each of the three groups a density map of the PDA usage was created. Contrary to our work presented in this paper no machine learning algorithms were applied to classify the participants of the experiment. Instead Ward's method, an algorithm for hierarchical cluster analysis, was applied to find clusters in the data. Nevertheless, this work served as a basis for our experimental setup.

Machine learning has widely been utilized in the domain of Geographic Information Science (GIScience), highlighting the promising potential for finding relationships between spatial and non-spatial data. For instance, in (Yan et al., 2018) they classified types of places by using a convolutional neural network(CNN) using geospatial as well as auxiliary data to perform this classification task. They concluded that CNNs hold great potential for performing this type of prediction task. In (Mc Cutchan and Giannopoulos, 2018) another type of machine learning algorithm was utilized, namely association analysis. They analyzed the connection of different types of land covers and the geo-objects they maintain. In (Stenneth et al., 2011), the possibilities of detecting the mode of transportation of a human using mobile phones and GIS information was investigated. They collected GPS traces of six transportation modes (stationary, walking, bike, bus, car and above ground train) over a time span of three weeks. After preprocessing the GPS data, it was fused with three different GIS datasets (real time bus locations, rail lines and bus stops). Based on this, they trained five different machine learning algorithms (Naive Bayes, Bayesian Network, Decision Trees, Random Forest and Multilayer Perceptron) and compared the precision and recall accuracy. Random Forest on average reached 93.70% precision accuracy and 93.80% recall accuracy, which was the best result of all five algorithms, therefore the Random Forest model was used as the final classification model. Based on that, they used the model on three slightly different datasets: once with and once without transportation network data as well as once only with the top five classification features. They found, that there was a significant change in accuracy depending on whether transportation network data was used or not (75% versus 94%). Pruning all but the top five features (which were in order of importance: average speed, average rail line closeness, average acceleration, average bus closeness and candidate bus closeness) hardly changed the accuracies (93%) and therefore they concluded that those are the most important features. Taking a closer look at the feature ranking, they argued that their model could be easily adapted for various regions of the world especially since widely available network data such as rail line closeness is higher ranked (number 2 of 8) and therefore more important than less available data like bus stop closeness (number 7 of 8). Moreover, they proposed to prune the network data according to the zip-code of the users location and found that this is a very good method to reduce computation time, especially for real-time, mobile systems. Contrary to the work in this paper they also deployed the developed system to the real world, with new test individuals under everyday use. The results achieved showed that the proposed system works under everyday conditions and is also very robust to traffic condition changes. These works served as an inspiration and guide for our selected methods.

Next to the LBS application, the prediction or detection of familiarity with the surrounding environment is also very important when evaluating LBS. Very often, different types of LBS systems are evaluated through a user study. Environmental familiarity can strongly bias the results of such experiments. For instance, when evaluating a navigation assistance system (Gartner et al., 2011, Kerber et al., 2014, Giannopoulos et al., 2015, Schirmer et al., 2015) if the users in one of the conditions are familiar with the environment (without explicitly stating it), they would have to rely less on the assistance, thus introducing a bias in the captured data that can alter the final results of the experiment.

3. Methodology

Within this section the creation of the 3D model, the setup of the experiment as well as the final processing and machine learning of the data is described. Navigation was enabled in the 3D model in order to perform the experiments. While the experiments were carried out, user behavior data was collected. This data was then processed and used to determine if the corresponding participant is familiar or not with the surrounding environment.

3.1 The 3D-Model

The regional government of Vienna, Austria provided a collection of data about the city. It ranges from spatial data over administrative data to demographic data. For the purpose of this paper, the *multi-purpose area map*, the *road graphs* as well as the *official trees register* were utilized. Next, the data were preprocessed and filtered. All the datasets used were imported into CityEngine and *rule-based modeling* and *Computer Generated Architecture* (CGA) rules were applied to calculate the heights of the buildings. This was done by calculating the distance between the lowest and the highest point of each building. Furthermore, the pre-existing *Building Mass Texturizer* rule file by Esri was used to apply randomly generated textures to the facades of the buildings. This method achieves a realistic look, however does not resemble the facades of the actual buildings (see Figure 1).

Within the modeled area, a suitable test route was chosen to accommodate several requirements: (1) It should contain sections with a high as well as sections with a low intersection density, (2) different types of intersections (T- and

[2]https://www.esri.com/en-us/arcgis/products/esri-cityengine

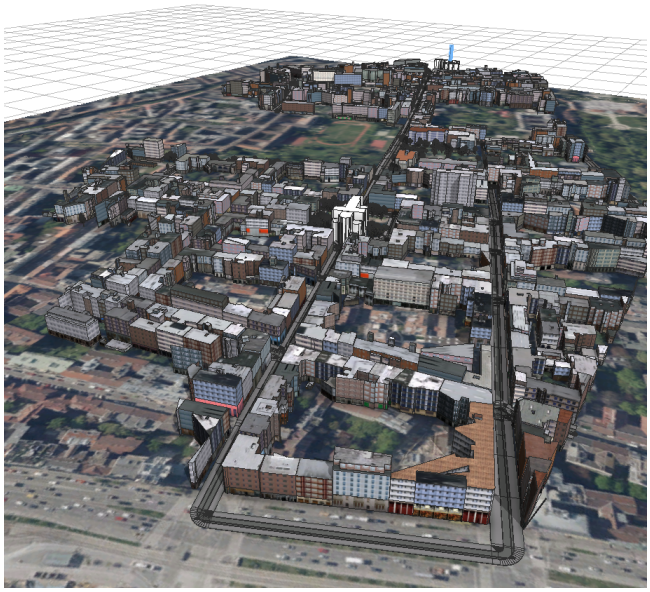

Figure 1. Overview of the 3D model scenery in City Engine[2]. The ground is overlaid with a satellite image.

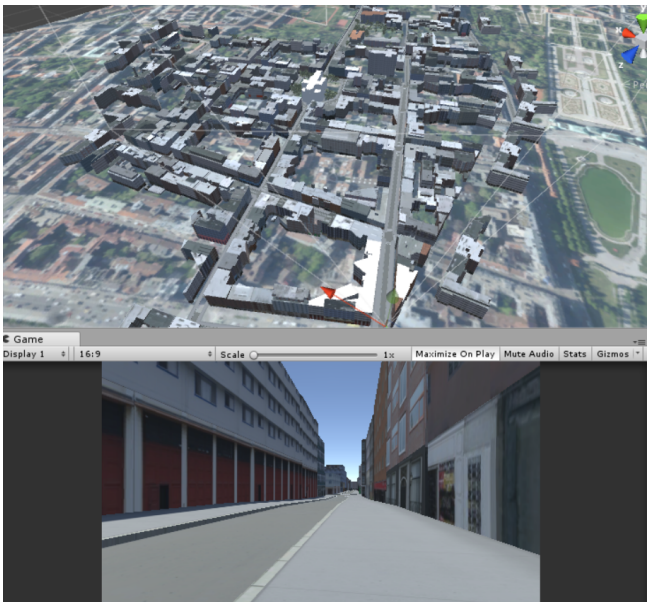

Figure 2. The 3D city model in Unity. The overview at the top is very similar to the CityEngine view and at the bottom, the first-person view is illustrated, which the users saw during the experiment. The participants were able to navigate through this environment using a joystick.

Y-Intersections and Crossroads (Fogliaroni et al., 2018)) should be present, and (3) there should be enough unique, distinguishable facades which can act as landmarks. Landmarks were used to define a route, however, not all of the selected or placed landmarks were later used in the selected route. Some of the landmarks acted as decoys in order to prevent participants from learning to detect the distinguishable textures. Finally, the models developed with the CityEngine software were exported using the *Autodesk FBX* format. Furthermore, a python script was used in order to export the IDs of the multi-purpose area map in order to ensure that each building will be uniquely identifiable during the post-processing.

Before the experiment started, the 3D-model was imported into the Unity game engine (see figure 2). During the import, a C# script was applied in order to ensure that every Unity *GameObject* will be connected with the correct multi-purpose area map ID.

The "Rigidbody First Person Controller" (avatar) was imported from the *Standard Assets* provided by Unity and added in the scene editor at the position the user should start with the experiment. In order to ultimately be able to control the avatar, a joystick (Logitech Extreme 3D Pro) was configured as an input device. The speed settings of the avatar were set during run-time via a menu before the start of the actual virtual environment scene. For this purpose two C# scripts were implemented. One script for changing the values from the start menu and one script to store the values and pass them through from the start menu to the actual scene. A further script was added as a component to the avatar GameObject in order to read the position and orientation of the avatar. This script applied ray tracing in order to find the IDs of all the buildings the user could possibly see from her current location. The script collected this data once every frame during the entire run-time of the experiment and stored the resulting strings into a comma-separated values file. This collected data was later used as input to train a machine learning algorithm.

In order to capture the environment of the avatar, ray tracing was used. An essential aspect regarding ray tracing is the interval in which the rays are cast. In this experiment a step size of 15° was used (see figure 3). Setting this interval too high, i.e. higher degree between the ray casts, would increase the likelihood of missing important objects. However, setting the interval to low would increase the computational effort.

Additionally a start and pause menu was developed for convenience. During the experiment this menu was used to change some preferences without having to open the editor. It also gives the participant time to prepare before the virtual environment was shown and the first audio instruction were played back. After this setup, all the software preparation was done and the virtual environment was able to be experienced while the actions were recorded.

3.2 Experiment

A between subjects design was employed for this experiment. Each participant from both groups had to navigate along the the same route in the same virtual environment. The only difference between the two groups was familiarity with the experimental area.

3.2.1 Participants

In total 22 participants took part in the experiment. They were split into two equal sized groups: one group was familiar with the experiment area (F) and one group was unfamiliar (U). The distinction between two groups was favoured over a more graded scheme, such as one including a "slightly familiar" class, as it would be almost impossible to gather a clean ground truth for it. The F group consisted of 8 males and 3 females. They had a mean age of 29.18 years with a standard deviation of 11.24. Their mean score on the Santa Barbara Sense-of-Direction Scale (Hegarty et al., 2002) was 4.47 points with a standard deviation of 1.33, the mean gaming experience was 4.27 points

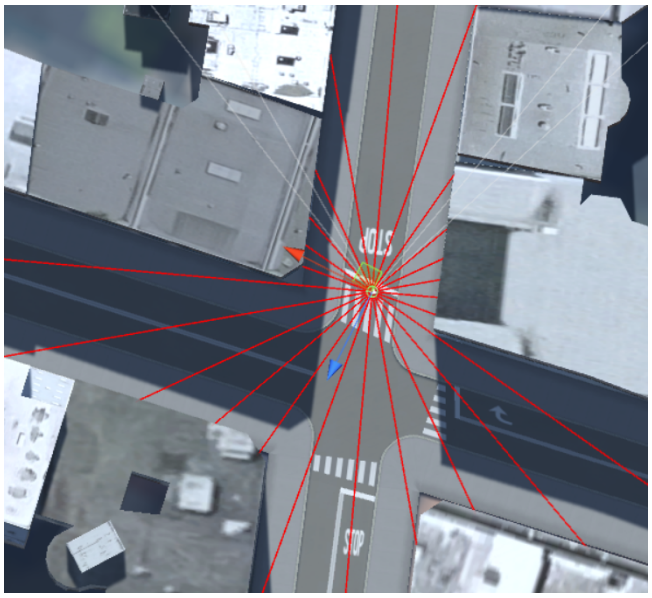

Figure 3. Top-down view of the rays cast by the avatar with a step size of 15°.

with a standard deviation of 2.28 and the mean joystick experience was 5.45 points with a standard deviation of 1.80. The U group consisted of 6 males and 5 females. They had a mean age of 23.72 years with a standard deviation of 5.15. Their mean score on the Santa Barbara Sense-of-Direction Scale was 5.38 with a standard deviation of 0.73, the mean gaming experience was 3.72 with a standard deviation of 2.45 and the mean joystick experience was 5.59 with a standard deviation of 1.70. These descriptive statistics differ for each group, however, not in a significant way. Therefore no significant impact on the comparability of both groups can be expected.

3.2.2 Setup

The experiments took place in a laboratory environment. The virtual environment was displayed on a projection wall (see figure 4). A joystick was placed on a table in front of the projection. The projector was 5.5m away from the screen and the table was set at the fixed distance of 2.5m from the screen. This setup was never changed throughout the entire experiment.

3.2.3 Procedure

All participants, from both groups, received the same information sheet and questionnaires. Next to demographic information, all participants were asked to fill in the *Santa Barbara Sense-of-Direction Scale*, a self-assessment of spatial abilities, as well as answer questions regarding their gaming and joystick experience. All questionnaires are based on a 7-Likert scale (1-7), with higher numbers indicating higher experience.

The participants in the F group got some time to accustom to the surroundings and look at the generated textures before the actual experiment started. It was assumed that if a user was already familiar with the geometry, the generated textures for the facades could be learned easily without losing the level of familiarity with the surrounding environment. During this activity the participants were free to explore the environment until they confirmed that they were familiar with it. In the event that a test person missed an important part of the model, the experimenter would suggest to further investigate the missed area and gave directions to it.

The actual experiment started after the exploration phase for the F group and immediately for the U group. First, the experimenter shortly recapped what was written on the information sheet to make sure that the participants understood the task. All participants were told to look at the middle of the screen and to look around the virtual environment only with the joystick controlled avatar and not by looking to the sides of the screen. Although the experimenter also reminded the participants during the experiment, it was noted that many participants kept looking around from time to time. This problem will be discussed later in detail. Next, the participants were told to follow the audio instructions played back by the experimenter. The instructions always consisted of a landmark and a turning direction, for example *"Turn right at the hotel"*. In the event that a test person overlooked a landmark, the supervisor would tell him to go back and have a closer look around the last intersections. As soon as it was obvious that the correct decision was made by the participant the next instruction was played back. After the experimenter closed the program, the collected data was transferred, together with the filled in questionnaires, to a participant specific folder.

Figure 4. A picture taken during the experiment. The participant can view the virtual environment on the screen and navigates through it with a joystick.

3.3 Data Processing

After the experiments were finished, features were extracted from the raw data and formatted in sparse vectors. The extracted features can be divided into two general groups: data related directly to the avatar and data related to the surroundings of the avatar. Features directly related to the avatar were further divided into two groups: position and orientation related features. Positional features are time, traveled distance, as well as the number and duration of stops. Orientation features are the rotation measured in degrees as well as the number of stops during the rotation. For each feature, descriptive statistics were computed, such as the maximum, minimum, mean, median

and variance. The second group, the features concerning the surrounding environment of the avatar, consisted of the identifiers (IDs) of the buildings that were in sight (see later in this section for details on what is considered to be in sight) of the avatar.

In order to be able to compute the descriptive statistics for the features, the data was aggregated over sections of the route the avatar traveled. Thus, measurements which were collected along a section were then used to compute the descriptive statistics of the features. Three different definitions of section where defined: (1) **Original section**, which is bounded within two consecutive intersections, (2) **Combination A** which is the same as the original definition but ignoring extremely close intersections, and (3) **Combination B** defines a section as the street segment between two intersections where a direction change occurred.

The algorithm we implemented computes the number of moves and rotations for every section. A move or a rotation is a single continuous motion, as soon as this motion is interrupted, a stop is introduced. This is done by checking if the position in this frame equals the position in the next. However, this does not apply in practice, as the data is captured with a higher frequency (same as the projected frame rate) than the motion of the avatar describes a change of location. Thus, as the data is captured and the avatar is in motion, it introduces artificial stops as no motion is recognized due to the higher frequency data is captured with. To overcome this problem, the position in one frame was tested against the position three frames later. If the coordinates did not differ, then the continuous motion was interrupted and a stop was introduced. Instead of calculating duration and distance of the motion from the first and the last frame, they were calculated once every three-frames and added up, in order to reduce potential noise. Rotations were processed in the same manner.

Once rotation as well as position related features were computed for each section, they were aggregated using descriptive statistics, such as minimum, maximum, average, median, standard deviation, and put into the yet empty feature vector for this section. Next, the IDs of all the visible objects found by ray casting were processed. They were then separated into three different groups: (1) IDs of objects which are within a full circle around the avatar. (2) Objects in the field of view (ranging from -30° to 30°), and (3) objects directly in front of the avatar. Afterwards, the number of unique IDs as well as the number of frames in which every unique ID was visible were computed for each of the three ranges and statistical values were derived from the resulting datasets. These computed values were then added to the feature vector of the corresponding section.

The entire feature extraction process resulted in three tables per user: one for the original sections, one for combination A and one for combination B. Lastly, the tables for each combination were merged into one table for all users. This merged table was passed to RapidMiner, which was used in order to apply machine learning. There, a sequence of operations was carried out (see figure 5). First, the data was preprocessed and subsequently carried out a parameter optimization for finding the best parameter for the used machine learning procedure . The preprocessing operator labeled the data and deleted those features which exhibited a significant correlation. The optimization aims at detecting the best parameters for machine learning by applying a grid based search strategy. Thus, it carries out the machine learning procedure multiple times and finds the set of parameters which score the best prediction results. Additionally, we performed a *10-fold cross validation*. Therefore, 10% of the table was used as test data and 90% of the table as training data. A 10-fold cross validation repeats the training and testing procedure 10 times. Each time, a different subset (10 % of the data) was used as test data. This yields ten different accuracy assessments for each of the ten iterations, which consist of a confusion matrix and an overall accuracy. In order to get an overall accuracy assessment, values from each iteration were averaged. This entire process was repeated four times. Each time a different machine learning algorithm was used, namely: Logistic Regression (Kleinbaum et al., 2002), Support Vector Machine (Hsu and Lin, 2002), Gradient Boosted Trees (Elith et al., 2008) and Random Forest (Breiman, 2001). Logistic regression uses a fitted logistic function to predict the class of the highest likelihood. SVM introduces support vectors into the feature space in order to create a hyper plane which aims at optimally separating the classes in the feature space. Gradient Boosted Trees based prediction aims at improving decision tree predictors gradually in a steepest descent manner. The random forest classifier introduces a set of decision trees which exhibit a minimal correlation to each other for predicting.

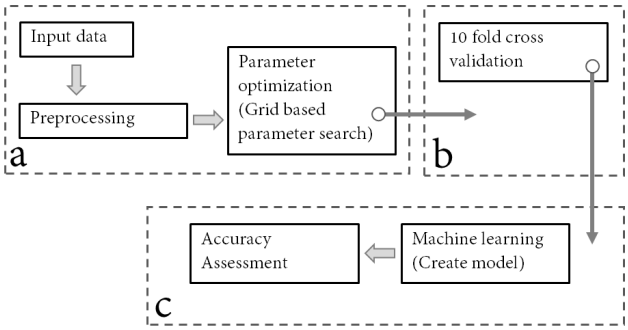

Figure 5. The overall machine learning process. The thinner arrows indicate, that the following part is inside the operator.
a: The outermost process, the data is retrieved, preprocessed and optimized.
b: Inside the optimization, the data is used the cross validation.
c: Inside the cross validation the training and testing is performed.

4. Results

The four machine learning algorithms were evaluated by confusion matrices and overall accuracy. The overall accuracy was additionally expressed by a percentage describing the fluctuation of the overall accuracy exhibited during the 10-fold cross validation. The values of the confusion matrix and the overall accuracy correspond to the average of every iteration of the 10-fold cross validation. Tables 1 to 4 show the prediction results for each machine learning algorithm using the original set of sections, Tables 5 to 8 show the prediction results using the section combination A, and Tables 9 to 12 show the prediction results using section combination B.

4.1 Results using the original sections

Table 1. Confusion matrix using a 10-fold cross validated Logistic Regression with the original sections splitting.

overall accuracy: 51.87% +/- 6.73%

	true familiar	true unfamiliar	class precision
predicted familiar	136	130	51.13%
predicted unfamiliar	51	59	53.64%
class recall	72.73%	31.22%	
Kappa(κ)	0.04		

Table 2. Confusion matrix using a 10-fold cross validated Support Vector Machine with the original sections splitting.

overall accuracy: 56.97% +/- 6.45%

	true familiar	true unfamiliar	class precision
predicted familiar	86	61	58.50%
predicted unfamiliar	101	128	55.90%
class recall	45.99%	67.72%	
Kappa(κ)	0.14		

Table 3. Confusion matrix using a 10-fold cross validated Gradient Boosted Trees with the original sections splitting.

overall accuracy: 55.08% +/- 6.54%

	true familiar	true unfamiliar	class precision
predicted familiar	68	50	57.63%
predicted unfamiliar	119	139	53.88%
class recall	36.36%	73.54%	
Kappa(κ)	0.10		

Table 4. Confusion matrix using a 10-fold cross validated Random Forest algorithm with the original sections splitting.

overall accuracy: 59.82% +/- 9.53%

	true familiar	true unfamiliar	class precision
predicted familiar	114	78	59.38%
predicted unfamiliar	73	111	60.33%
class recall	60.96%	58.73%	
Kappa(κ)	0.20		

4.2 Results using the section combination (a)

Table 5. Confusion matrix using a 10-fold cross validated Logistic Regression with the sections combination (a).

overall accuracy: 55.52% +/- 8.82%

	true familiar	true unfamiliar	class precision
predicted familiar	101	85	54.30%
predicted unfamiliar	42	58	58.00%
class recall	70.63%	40.56%	
Kappa(κ)	0.11		

Table 6. Confusion matrix using a 10-fold cross validated Support Vector Machine with the sections combination (a).

overall accuracy: 54.52% +/- 10.23%

	true familiar	true unfamiliar	class precision
predicted familiar	78	65	54.55%
predicted unfamiliar	65	78	54.55%
class recall	54.55%	54.55%	
Kappa(κ)	0.09		

Table 7. Confusion matrix using a 10-fold cross validated Gradient Boosted Trees with the sections combination (a).

overall accuracy: 52.80% +/- 4.42%

	true familiar	true unfamiliar	class precision
predicted familiar	110	102	51.89%
predicted unfamiliar	33	41	55.41%
class recall	76.92%	28.67%	
Kappa(κ)	0.07		

Table 8. Confusion matrix using a 10-fold cross validated Random Forest algorithm with the sections combination (a).

overall accuracy: 61.16% +/- 8.06%

	true familiar	true unfamiliar	class precision
predicted familiar	82	50	62.12%
predicted unfamiliar	61	93	60.39%
class recall	57.34%	65.03%	
Kappa(κ)	0.22		

4.3 Results using the section combination (b)

Table 9. Confusion matrix using a 10-fold cross validated Logistic Regression with the section combination (b).

overall accuracy: 53.24% +/- 9.71%

	true familiar	true unfamiliar	class precision
predicted familiar	59	54	52.21%
predicted unfamiliar	25	31	55.36%
class recall	70.24%	36.47%	
Kappa(κ)	0.07		

Table 10. Confusion matrix using a 10-fold cross validated Support Vector Machine with the section combination (b).

overall accuracy: 54.41% +/- 11.56%

	true familiar	true unfamiliar	class precision
predicted familiar	54	47	53.47%
predicted unfamiliar	30	38	55.88%
class recall	64.29%	44.71%	
Kappa(κ)	0.09		

Table 11. Confusion matrix using a 10-fold cross validated Gradient Boosted Trees with the section combination (b).

overall accuracy: 61.58% +/- 7.36%

	true familiar	true unfamiliar	class precision
predicted familiar	53	34	60.92%
predicted unfamiliar	31	51	62.20%
class recall	63.10%	60.00%	
Kappa(κ)	0.23		

Table 12. Confusion matrix using a 10-fold cross validated Random Forest algorithm with the section combination (b).

overall accuracy: 65.70% +/- 6.22%

	true familiar	true unfamiliar	class precision
predicted familiar	54	28	65.85%
predicted unfamiliar	30	57	65.52%
class recall	64.29%	67.06%	
Kappa(κ)	0.31		

5. Discussion

The results revealed that the Random Forest achieved the best overall accuracy of 65.70% with a variance of 6.22% when using the data captured at street segments between direction changes (i.e., combination B). Both classes (familiar and unfamiliar) got similar class recall (64.29% and 67.06%) and precision (65.85% and 65.52%) values. The most important attributes for the algorithm were the rotation time and rotation distance. This makes sense, since people unfamiliar with their surrounding environment will most likely look around and turn their head more than people familiar with their surroundings. Since the walking speed is relatively slow, the participants had enough time to look around even without stopping. Therefore the average moving time did not differ significantly between the two test groups making this attribute not as important for the machine learning algorithm. Other important features for the algorithm were the building IDs that were in the avatars field of view. This shows whether a participant looked at a single building for a long time or not. The IDs tended to be constant in a couple of different scenarios: a participant kept her head fixed at a building while walking past it, or while walking in a straight line to the next intersection.

The combination A combined a part in the route where many short sections were in a row. In this area there was the special case that the landmark, which is placed at the end, is already highly visible from the start. It can be assumed that if a participant, familiar or unfamiliar, spotted the landmark in the back she would stop searching for the landmark and walk straight toward it. Having too many short sections, the participant would be able to pass through several sections while walking straight ahead. The features from these sections would not characterize the behavior sufficiently in order to be able to distinguish it. Some test participants did not exclusively use the joystick controlled avatar to look around but instead turned their head to the left or the right of the screen. This implies lost information on the rotation related features.

One of the difficulties encountered before the experiment started was the separation of participants between familiar and unfamiliar ones. When can someone be considered familiar with her surrounding environment? In our case, we explicitly asked the participants about their familiarity with the environment and also let them navigate in the virtual environment, allowing them to explore it. Although this is not comparable to someone who has lived for several years in the area and actually interacted with many of the buildings several times. Nevertheless, we considered the training session as an adequate approximation. Furthermore, next to the familiarity level, factors such as spatial abilities, gaming and joystick experience are also very important. Although the reported differences between the two groups were not significantly different, still, there might be an impact which needs to be further investigated.

A problem encountered during the experiment was related to the tracked behavioral data. Although the experimenter instructed every participant to keep their head steady towards the middle of the screen many participants occasionally moved their head to the sides, changing their field of view. This change should actually be reflected in the collected data as a joystick turn. In an extreme case, the data may show that the participant was walking in a straight line, looking straight ahead while in reality she was constantly moving her head searching for the landmark. There are several possible solutions: One solution would be to limit the field of view. This would at the same time make the whole experiment less immersive and would not represent the way humans observe and interact with the real environment. Another approach could be to make the image blurry, except of the center, e.g., introduce a spotlight effect. This way it would be gradually harder to read a text written on a facade but the participant would still be able to notice that there is something that could be interesting. A different solution would be to measure the actions of the participant in the real world. This could be done by recording the gaze (Kiefer et al., 2017) or the head movements via an external device like an eye tracker or motion trackers. Depending on the actual method, this could lead to very precise data on where the participant focused her gaze during the experiment, regardless whether the avatar is turned in this direction or not. Further research has to be conducted to find practical solutions to this problem.

6. Conclusion and Outlook

In this work, a virtual environment was created in order to let 22 participants navigate in it. Their movement was recorded and gathered in order to extract features. These features were than used by four different machine learning algorithms, to determine if the participants are familiar, or unfamiliar with their environment.

In conclusion it can be said that through the presented user experiment we accomplished to collect data that enables a machine learning algorithm to predict to a certain degree the local familiarity of a human user. Furthermore, the analysis showed that the rotation data was the most important feature.

The results show that using a random forest classifier, it is possible to predict with an overall accuracy of 65.7%. Therefore, this machine learning algorithm is the most promising candidate when pursuing further research on this work.

One should look into using real facade textures for the virtual environment to make the experience for the participants even more immersive. It should also be considered to use head or eye tracking devices for capturing the actions of the participants while controlling the avatar. Because these improvements would closer resemble the real world as well as provide data about previously not captured actions they are likely to increase the overall accuracy. Additionally, instead of separating features into rotations and linear movements, an integrated representation can be used for future work. Considering the setup of the 3D-virtual environment, CityEngine provided an easy to use interface which enables one to create a 3D model of an urban environment from existing ground plots and heights. CityEngine also provides capabilities to extend the model if more data becomes available at a later point.

Clearly, additional improvements can be made to the presented approach. However, the results presented in this work look promising and illustrate that it holds potential for further research.

References

Anderson, C. A. and Bushman, B. J., 1997. External validity of "trivial" experiments: The case of laboratory aggression. *Review of General Psychology* 1(1), pp. 19–41.

Breiman, L., 2001. Random forests. *Machine learning* 45(1), pp. 5–32.

Elith, J., Leathwick, J. R. and Hastie, T., 2008. A working guide to boosted regression trees. *Journal of Animal Ecology* 77(4), pp. 802–813.

Fogliaroni, P., Bucher, D., Jankovic, N. and Giannopoulos, I., 2018. Intersections of Our World. In: S. Winter, A. Griffin and M. Sester (eds), *10th International Conference on Geographic Information Science (GIScience 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 114, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 3:1–3:15.

Gartner, G., Huang, H., Millonig, A., Schmidt, M. and Ortag, F., 2011. Human-centred mobile pedestrian navigation systems. *Mitteilungen der Österreichischen Geographischen Gesellschaft* 153, pp. 237–250.

Giannopoulos, I., Kiefer, P. and Raubal, M., 2015. Gazenav: gaze-based pedestrian navigation. In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ACM, pp. 337–346.

Hegarty, M., Richardson, A. E., Montello, D. R., Lovelace, K. and Subbiah, I., 2002. Development of a self-report measure of environmental spatial ability. *Intelligence* 30(5), pp. 425 – 447.

Hsu, C.-W. and Lin, C.-J., 2002. A comparison of methods for multiclass support vector machines. *IEEE transactions on Neural Networks* 13(2), pp. 415–425.

Huang, H., Gartner, G., Krisp, J. M., Raubal, M. and de Weghe, N. V., 2018. Location based services: ongoing evolution and research agenda. *Journal of Location Based Services* 12(2), pp. 63–93.

Kerber, F., Krüger, A. and Löchtefeld, M., 2014. Investigating the effectiveness of peephole interaction for smartwatches in a map navigation task. In: *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, ACM, pp. 291–294.

Kiefer, P., Giannopoulos, I., Raubal, M. and Duchowski, A., 2017. Eye tracking for spatial research: Cognition, computation, challenges. *Spatial Cognition & Computation* 17(1-2), pp. 1–19.

Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M. and Klein, M., 2002. *Logistic regression*. Springer.

Kuliga, S. F., Thrash, T., Dalton, R. C. and Hölscher, C., 2015. Virtual reality as an empirical research tool—exploring user experience in a real building and a corresponding virtual model. *Computers, Environment and Urban Systems* 54, pp. 363–375.

Li, C., 2006. User preferences, information transactions and location-based services: A study of urban pedestrian wayfinding. *Computers, Environment and Urban Systems* 30, pp. 726–740.

Mc Cutchan, M. and Giannopoulos, I., 2018. Geospatial Semantics for Spatial Prediction (Short Paper). In: S. Winter, A. Griffin and M. Sester (eds), *10th International Conference on Geographic Information Science (GIScience 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 114, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 45:1–45:6.

Schirmer, M., Hartmann, J., Bertel, S. and Echtler, F., 2015. Shoe me the way: a shoe-based tactile interface for eyes-free urban navigation. In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ACM, pp. 327–336.

Stenneth, L., Wolfson, O., Yu, P. S. and Xu, B., 2011. Transportation mode detection using mobile phones and gis information. In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, ACM, New York, NY, USA, pp. 54–63.

Yan, B., Janowicz, K., Mai, G. and Zhu, R., 2018. xNet+SC: Classifying Places Based on Images by Incorporating Spatial Contexts. In: S. Winter, A. Griffin and M. Sester (eds), *10th International Conference on Geographic Information Science (GIScience 2018)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 114, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 17:1–17:15.