Indoor positioning with the Microsoft Hololens

MSc Geomatics Thesis Proposal

Laurens Oostwegel
Student Number 4284887
l.oostwegel@student.tudelft.nl

1st supervisor: Stelios Vitalis
2nd supervisor: Ken Arroyo Ohori
external supervisor CGI: Robert Voûte

January 13, 2020

Contents

1 Introduction **3**

2 Related work **4**
2.1 Emergency Response 4
2.2 Indoor Positioning 4
2.3 Positioning with the Microsoft Hololens 5
2.4 Marker-based spatial matching 5
2.5 Marker-less spatial matching 6

3 Research objectives **7**
3.1 Research questions 7
3.2 Scope 8

4 Methodology **8**
4.1 Input floor plans 8
4.2 Wall extraction 9
4.3 Spatial Matching Algorithms 9
4.3.1 Iterative Closest Points 10
4.3.2 Local quadratic approximant and instantaneous kinematics 10
4.3.3 Hough Transform 11
4.4 Positioning 12
4.5 Evaluation 12
4.6 Preliminary Results 13

5 Time planning **13**

6 Tools and datasets used **13**

Acronyms

AR Augmented Reality

BIM Building Information Model

ER Emergency Response

GPS Global Positioning System

HGPS High-sensitivity Global Positioning System [GPS]

ICP Iterative Closest Points

IMU Inertial Measurement Unit

MH Microsoft Hololens

MSE Mean Square Error

SLAM Simultaneous Localization and Mapping

1 Introduction

It is difficult to imagine the world nowadays without a GPS. Navigation apps are omnipresent and every mobile phone has the ability to tell you how to go from point A to B. That is to say, concerning outdoor navigation. Reliable indoor positioning and localization is still a challenging task. Traditional indoor services require pre-installed infrastructure or a priori information, such as Bluetooth beacons or Wi-Fi fingerprinting. These systems have their challenges, such as high costs and multipath effects by obstacles (e.g. walls and furniture Yang and Shao, 2015). While they suffice the need for indoor positioning in many scenarios, sometimes it is considered infeasible to start with any existing infrastructure or information (Rantakokko et al., 2011). In the case of Emergency Response [ER], the location of impact is not known beforehand and there is little time available to act. Alternative systems have been proposed for these kind of situations, that often require synergies of multiple sensors (see for example Rantakokko et al., 2011).

One potential technique for positioning that does not need any infrastructure is the Simultaneous Localization and Mapping [SLAM] algorithm. SLAM makes use of sensors, such as RGB cameras, depth cameras or laser range finders, to continuously map a space and find its own location inside the mapped space. With the emergence of Augmented Reality [AR], SLAM becomes a viable option for indoor positioning. Not only does AR make use of the positioning method, it also gives immediate feedback through holograms (Brito et al., 2019). There is however a problem with the method. It only computes its position within its own mapped space, not within an existing map that can be used for navigation. While manually placing the position of an AR device on the map at initialisation can partly solve this problem, over time the device will suffer from drift (Hübner et al., 2019). Drift is an error that gets bigger over time, therefore making the algorithm less reliable. The AR device used in this thesis, the Microsoft Hololens [MH], has the same problem. One needs to update the location of the MH in relation to the map, to account for the drift error. This thesis will use the SLAM technique of the MH, in combination with various registration techniques, to position itself.

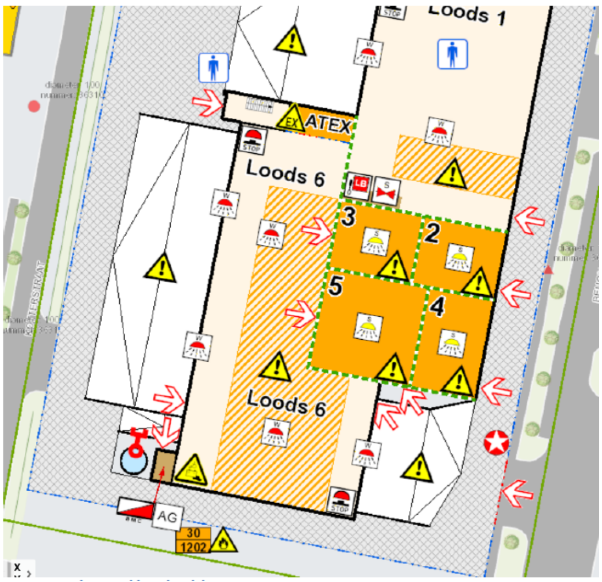

Figure 1: Mobile Operational Information System of The Netherlands (van der Meer, 2016)

In order for a positioning system to be useful, it should relate to a context that is apprehensible for a person, such as a map or a 3D model. While a 3D Building Information Model [BIM] is usually not at hand, 2D alternatives are better accessible (Zlatanova et al.). In ER, some organizations have web applications with building information. For example, the project *Mobiel Operationeel Informatiesysteem* (Mobile Operational Information System) is used by first responders in the Netherlands. This system contains both outdoor information and indoor information of buildings with a higher risk of hazards to scale (see fig 1). In this thesis, some similar floor plan to scale is used to position the MH on.

2 Related work

2.1 Emergency Response

ER is the acute phase within emergency management that starts immediately after a disaster happens (Zlatanova and Holweg, 2004, p. 1). It is the third of four phases in emergency management, after the mitigation and preparedness phases, and before the recovery phase. ER is characterized by the little amount of time that is available to act (Zlatanova and Holweg, 2004). When used appropriately, spatial data can be used to make faster and more adequate decisions within ER (Abdalla, 2016). Often, such geographical systems involve determining positions and movement (van der Meer, 2018; Abdalla, 2016), such as the tracking of personnel during a deployment (van der Meer, 2018). Therefore, indoor positioning and localization is one major area of interest within the field of ER (Rantakokko et al., 2011). End users that could benefit from positioning information are navigators and (head) officers. A navigator is a currently non-existing person, proposed by (van der Meer, 2018), that accompanies a team and helps with indoor wayfinding. A (head) officer manages one or several teams inside and outside the building (van der Meer, 2018). The officer is interested in the positions of its teams.

2.2 Indoor Positioning

There exists some ambiguity on the definition of positioning, as opposed to localization. The paper of Sithole and Zlatanova (2016) will be leading on this matter (see Table 1). A position *"defines a pin point in space"* that is set within a reference frame (Sithole and Zlatanova, 2016, p.91). Its position is absolute, with a specificity that depends on the device that gives the position. A location is defined as *"the smallest physically defined space in a building"* (Sithole and Zlatanova, 2016, p.92). The location is certain, defined by physical borders (e.g. someone is in the study room). The location is always with a context (someone is inside of some place that has a meaning).

	Position	**Location**
Reference	Absolute (coordinate system)	Absolute (room number)
Specificity / Uncertainty	Depends on device providing position	Certain, defined by the physical borders (walls)
Scope	Defined by a reference frame	Contains places
Context	No context	Context

Table 1: Position and location, adapted from Sithole and Zlatanova (2016)

A position of a person or device is always related to some space that the subject is in. This could be globally defined, e.g. the position of the subject in the world. GPS uses global coordinates to find its position on earth. Position can also be defined locally, such as the position in an existing model of a building. A number of methods exist to position or localize someone within a building. More traditional positioning and localization methods that involve Bluetooth, Wi-Fi or similar wireless systems, need pre-installed infrastructure, such as beacons, or a priori information, like fingerprinting. This is considered infeasible for many scenarios, as there is no time to install such systems. Approaches that do not need a priori information or pre-existing infrastructure rely on integration of multiple sensors. These include High-sensitivity GPS [HGPS], foot-mounted inertial systems, imaging sensors, ultrasonic sensors and more (Rantakokko et al., 2011, 12).

2.3 Positioning with the Microsoft Hololens

One device that offers possibilities in ER and indoor positioning, is the MH (Khoshelham et al., 2019; Sharma et al., 2019). Using the device for cartographic purposes is a highly new and developing research field. The positioning of the MH relies on SLAM. It relates its position to a virtual model of the environment that is mapped in real-time. The MH has a built-in proprietary SLAM algorithm to track its position with regard to its environment (Newcombe et al., 2014; Cyrus et al., 2019). This so-called 'Head-pose Tracking' combines a depth camera, RGB cameras and an Inertial Measurement Unit [IMU] to accurately calculate its stand (see fig. 2). The accuracy of the spatial mapping gives promising results with regard to positioning in small indoor spaces (Khoshelham et al., 2019). However, when it comes to positioning in larger indoor spaces, the MH shows it has major drift effects (Hübner et al., 2019).

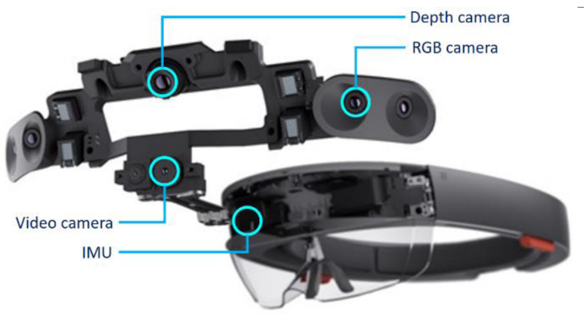

Figure 2: Microsoft Hololens with its sensors (Khoshelham et al., 2019)

Hence, it is not easy to track the MH inside an existing model, like a 2D floor plan or a BIM-model. As the virtual model is directly laid over the real world, the visualization of this model needs to be very accurately placed, because any deviation of the virtual model with the real world will be observed easily (Hübner et al., 2018). The placement of the existing model could be determined manually, by using the gesture system of the MH. However, this brings two problems: (1) a small rotational error at time of placing the virtual model can result in huge errors in other parts of the building (Fonnet et al., 2017) and (2) as already explained before, mapping large indoor spaces will result in drift effects (Hübner et al., 2019). There has been some effort to minimize these errors. Solutions could be found in two main directions: spatial matching with and without markers. The following sections will elaborate on this.

2.4 Marker-based spatial matching

One approach to overcome the problem of positioning the MH inside an existing model is to use fixed markers in the real world as reference points, that are positioned in a BIM as well. The pose of the MH with regard to the mapped marker is combined with the pose of the marker with regard to model (Hübner et al., 2018). This could be extended by using computer vision to find existing features in a building, such as text on signs (e.g. room numbers). If the BIM contains the same text on the correct place, the text could be used equivalent to the markers.

Some problems occur when using marker-based spatial matching. In the case of ER, it is not feasible to stick markers and assign them to an exact location in a virtual model. There is a limited time-frame and these kind of procedures are too time-consuming. The text-based spatial matching overcomes that problem. It uses already existing features of a building to estimate its position. Nonetheless, a 3D BIM is needed to accurately reference the text markers and thus estimate the position of the MH. During hazards, it is not likely that a 3D BIM is at hand.

2.5 Marker-less spatial matching

Other approaches to match the existing and real-time virtual models do not have to rely on markers or extensive BIM models. These methods bring together an existing virtual model of a building and an on-the-fly created mesh. In essence is this the same as shape registration. Some of these matching techniques have been used with the MH already, such as the Iterative Closest Points [ICP] algorithm. Research performed in this field will shortly be elaborated on. Other algorithms have not yet been tested on the MH, but are proposed as alternatives to ICP.

The ICP algorithm is a method for registration of a 3D shape. It tries to find a rotation and translation vector, based on the closest points between input points (in this case the MH acquired mesh) and a shape (the existing floor plan) and iterates until a local minimum is reached. It always converges to the nearest local minimum of a mean-square distance metric (Besl and McKay, 1992, p. 586). The algorithm works best if the initial position of the input dataset is similar to the source. It is a useful technique for the MH, since the position of the input data can already be approximated by the built-in SLAM method. Therefore, convergence can be reached relatively fast.

Kopsida and Brilakis (2016) use Kinect Fusion to match a BIM file with the current view on the Kinect camera. While they use a different device, it has similar characteristics to the MH. It uses the 6DoF pose estimation from KinectFusion to follow the movement of the camera. To deal with possible errors, the ICP is used to correctly align the BIM model with the scene. While the results were positive, the scene that was reconstructed merely consisted of someone's workspace. On top of that, the camera needed to move slow, to prevent loss of tracking.

A full-scale application is proposed by Fonnet et al. (2017). Their approach is to manually position a BIM roughly on the site. The position of the model can be refined with use of the ICP algorithm. Brito et al. (2019) try to realize the proposed method, but fail to do so. They give as reasons that (1) ICP is resource intensive, which is not feasible with the limited hardware capabilites of the MH, (2) the spatial mapping data is noisy and (3) the spatial mapping is very local, since only a small portion of the building data is available at start of the mapping. They do conclude that these problems could be overcome with custom designed algorithms.

The following method is based on the local quadratic approximants of the squared distance function and instantaneous kinematics, and is proposed as an alternative to ICP. The method should be faster than (non-optimized) ICP on small differences between two point clouds, while it will perform similar on point clouds that are less aligned. This is mainly due to ICP exhibiting a linear convergence, while the proposed method has a non-linear (squared) convergence (Pottmann et al., 2004). The algorithm follows three steps.

1. Compute the local quadratic approximant F_i on surface ϕ for every point x_i
2. Attach a velocity vector $v(x_i)$ to each point, such that the approximated distance between surface ϕ and the displaced point $x_i + v(x_i)$ is minimized. This gives velocity field $v(x)$.
3. Compute the Euclidean displacement of all points based on the velocity field $v(x)$.

Ni et al. (2013) propose the use of Hough transform as an alternative to ICP to register a 3D point cloud of multiple buildings to a 2D map image, such as Google Maps. While the scale of the use case is bigger in their research, the idea is very similar. An input point cloud is matched with a 2D map. The algorithm makes use of the fact that rotation is independent of translation in the Hough spectrum. The research concluded that the algorithm was slower than ICP. The parameter that was hardest to determine was the scale (Ni et al., 2013). The workflow is as following:

1. Find planes by by transforming the points into Hough domain and using a kernel to find centres of the clusters.
2. Find the up-vector, or the normal of the ground plane.
3. Find the rotation γ by making use of the characteristic of the Hough space that rotation is independent of translation and scale.
4. Correct the scale.
5. For the translation, the Euclidean space and a cross-correlation method is used.

3 Research objectives

The end goal of this research is to develop a method to estimate the position of the Microsoft Hololens (MH) on a 2D floor plan, without any pre-existing infrastructure. To develop such a method, a spatial matching technique that is used in the method should be chosen. The positioning method can be used in situations such as ER, when there is no time available for installing infrastructure and when there is no 3D BIM at hand. The method should communicate the precision of the given position and the probability that the device is actually in the room that the method suggests it is in. The developed approach can ultimately help to improve and speed up decision making in the case of ER.

3.1 Research questions

Based on the the objectives, a main research question could be developed. The main research question is:

How can the Microsoft Hololens improve indoor positioning, using the on-the-fly produced mesh with an existing 2D floor plan?

The following sub-questions will accommodate the main research question:

1. What spatial matching techniques are feasible to match a 3D mesh acquired real-time with the Microsoft Hololens with an existing 2D floor plan?
2. What is the most promising spatial matching technique in terms of accuracy and speed?
3. How can the indoor positioning of the Microsoft Hololens be improved by making use of the researched positioning method?
4. In which cases does the researched positioning method fail to estimated the position on a 2D floor plan?

3.2 Scope

This thesis will focus on estimating the position of the MH on a 2D floor plan. A system will be developed that works on the MH and shows its location on the 2D floor plan. While many 2D floor plans are in reality not to scale, in this thesis it is assumed that they are. This can be justified by the fact that first responders already use tools like the previously explained *Mobile Operational Information Systems*, that do have a correct scale. Evaluation of the method will be on accuracy, speed and robustness. The method should be tested on a general case, as well on particular cases that involve more complex forms, such as stairs; doors; windows; doors; stairs; windows; hallways without corners; large rooms; incongruence between floor plan and real world. While the proposed method should test in which of the particular cases the method fails, it is not meant to deal with these failed cases.

One issue that is heard when working with the MH is the limited processing power (Brito et al., 2019). If the MH appears to have problems with the processing power, it suffices to establish a real-time connection with a computer and use that device for the computations. Although this is considered infeasible for real life situations, it does not detract from the method itself. It is more justified by the fact that the first generation of MH is used in this thesis, that has less processing power than the already on the market second version of the MH.

4 Methodology

The objective of this thesis is to estimate the position of the MH on an existing 2D floor plan. There are two general approaches to this goal. The SLAM algorithm of the MH could be adjusted, so that the acquired meshes relate to the floor plan. This will likely be a very complicated task, since the algorithm is proprietary and therefore very hard to change. Another method is to find local transform parameters of the acquired meshes to the floor plan and apply these to the position within the floor plan. There is a dual system that has one position within the on-the-fly acquired model and one on the existing floor plan. The dual system is made up of the Hololens data and the SLAM algorithm on the one side and the floor plan with the position on the floor plan on the other side. The position that is computed by the SLAM algorithm maps to the position on the floor plan by the transform parameters found in the spatial matching algorithm. The proposed workflow of this dual system is shown in figure 3. In an initialization step, the first acquired mesh is placed on the estimated start position that is manually placed by the Hololens carrier. The walls are extracted from the mesh with use of surface normals. Afterwards, the transform parameters of the mesh to align with the 2D floor plan are estimated through the spatial matching algorithms. These transform parameters are not used to change the position the mesh or the floor plan itself, but to map the position of the SLAM algorithm to the position on the floor plan. This process repeats every ten seconds.

4.1 Input floor plans

The tests in the research will be located in multiple buildings. There are some requirements to the floor plans of the buildings, to reduce the scope of the research. Based on these requirements, an appropriate data format will be chosen. They are as follows:

- The floor plan should have the correct scale.
- The floor plan should have a vector format.
- There should be some semantics in the floor plan. These should not be extensive. Walls, doors, stairs and windows will be sufficient.

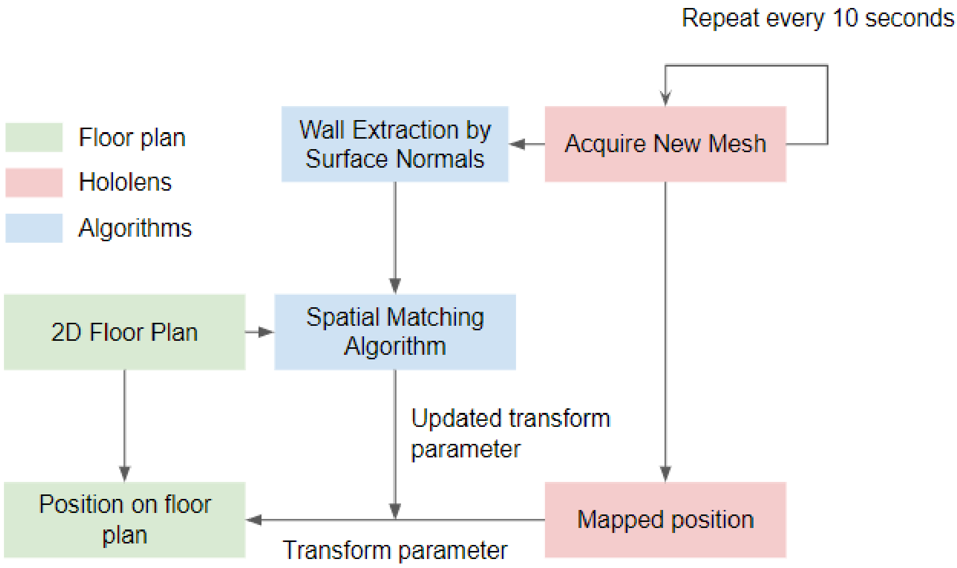

Figure 3: Workflow of positioning the MH inside a 2D floor plan

- The floor plan should reduce its information to the semantic information defined in the previous item, in order that the algorithm is not diffused with unnecessary information.

4.2 Wall extraction

Since the reference data is a 2D floor plan, the only data that is needed from the acquired mesh are the walls that are mapped. Therefore, these should be filtered. The MH optionally gives surface normals along with the acquired mesh. On top of that, the ground plane is automatically aligned, therefore making it relatively easy to extract all meshes that are perpendicular to the ground. The remaining data does not only contain the walls, but also furniture. Therefore, another filter should be applied, to remove the meshes that aren't walls. The SLAM algorithm of the MH performs very well on small spaces. As the position of the MH on the floor plan updates every 10 seconds, the newly proposed method only has to account for a small error correction every time. Therefore, using all points within a certain buffer distance from the walls on the floor plan will sufficiently remove all meshes that are of no use. The exact buffer distance needs to be defined during testing.

4.3 Spatial Matching Algorithms

Three spatial matching algorithms have been selected as feasible options. These are ICP, the hough transformation and instantaneous kinematics. The workflow of all are shortly explained in section 2. The inputs of the algorithms are the wall meshes and the 2D floor plan. The output is a transformation matrix. The next sections will give a more elaborate explanation of all algorithms, together with a notion where they need to be customized in order to be fit for deployment on the MH. All algorithms take a 3D mesh and 2D lines as input. Both the ICP algorithm as the local quadratic approximant and instantaneous kinematics work with either 2D or 3D data (but not a mix of both). Therefore, for those algorithms, a decision should be made to either extract surfaces of the 2D floor plan lines, or remove the Z component of the 3D meshes. The hough transform algorithm works with a 2D map and a 3D point cloud, so there is no need to convert the input to either fully 2D or 3D for this algorithm. Lastly, the output of the MH is a mesh, rather than a point cloud. All algorithms work with a point cloud as input. Since the mesh of the MH is derived from a point cloud, the mesh will be dealt with as a sparse point cloud.

4.3.1 Iterative Closest Points

The following section is adapted from Besl and McKay (1992) and describes the workflow of the ICP. The objective of ICP is to find a rotation parameter R and a translation t so that a data shape P is best aligned with model shape X. Therefore, the distance between displaced points of data shape P and model shape X should be minimized. The optimization function is as follows, where $q(P)$ are the points displaced by R and t:

$$\underset{\vec{q}(P)}{\arg\min}\, d(M, \vec{q}(P)) \tag{1}$$

Every iteration it follows three steps. First, for every point in data shape P, the closest point to model shape X is computed:

$$d(\vec{p}, X) = \min_{\vec{x} \in X} \|\vec{x} - \vec{p}\| \tag{2}$$

Then, the displacement vector $\vec{q}(P)$ should be computed. This can be found by minimizing the mean square objective function:

$$\underset{\vec{q}}{\arg\min}\, \frac{1}{N_p} \sum_{i=1}^{N_p} \|]\vec{x}_i - R\vec{q}_R\|^2 \tag{3}$$

In this equation, $\vec{q}_R$ is the rotational component of $\vec{q}$ and $\vec{q}_T$ the translational. R is the 3x3 rotation matrix. In the third step, the data shape P is displaced by $\vec{q}$ and the algorithm iterates until equation 1 is below a certain threshold.

4.3.2 Local quadratic approximant and instantaneous kinematics

The first step of this algorithm is to find the closest distance of every x_i in data shape P to the tangent plane of the nearest point y_i on the surface ϕ. Every point x_i is rewritten as a combination of nearest point y_i, normal vector n_i and distance d_i

$$x_i = y_i + d_i \mathbf{n_i} \tag{4}$$

The object that is used in the research of Pottmann et al. (2004) research to register a point cloud has a lot of curvature. This is different from the object in the current study: the surfaces of buildings are usually planar. The tangent plane of planar surfaces is the plane of the surface itself. Therefore, the first step in the algorithm is arguably not performing better in the case of building point cloud registration and could possibly not yield the same results as found in Pottmann et al. (2004).

In the following step, the goal is to minimize the approximated distance between all points in X and the surface ϕ. Let F be the approximant of the squared distance function and $v(x_i)$ the velocity vector of point x_i. The optimization function that needs to be solved is:

$$\underset{v(x)}{\arg\min} \sum F_i(x_i + v(x_i)) \tag{5}$$

The velocity vector $v(x)_i$ is made up of two parameters: $\mathbf{c}$, the velocity vector of the origin (in this case point y), and $\bar{\mathbf{c}}$, the vector of angular velocity (or Darboux vector). It can be computed by:

$$v(x_i) = \bar{\mathbf{c}} + \mathbf{c} \times x_i \tag{6}$$

Using equation 4 and 6, the minimization function in equation 5 can be rewritten as:

$$F(C) := F(\bar{\mathbf{c}}, \mathbf{c}) = \sum_i (d_i + \mathbf{n_i} \cdot (\bar{\mathbf{c}} + \mathbf{c} \times x_i))^2 \tag{7}$$

Where d_i is the distance between x_i and y_i and n_i is the normal vector from the tangent plane. This can be rewritten into 8, that has a unique minimum found through 9 (See Pottmann et al., 2004, for a full explanation). A is a symmetric, positive definite six-by-six matrix, B is a column vector with six entries, C is the pair $(\bar{\mathbf{c}}, \mathbf{c})$ and D is just some scalar.

$$F(C) = D + 2B^T C + C^T AC \tag{8}$$

$$AC + B = 0 \tag{9}$$

The motion C $((\bar{\mathbf{c}}, \mathbf{c}))$ is a not a Euclidean rigid body motion, but an affine one. Therefore, C should be transformed in a rotation about an axis (G) of $\alpha = \arctan \|\mathbf{c}\|$ and a translation that is parallel ot G by a distance of $\rho \cdot \alpha$, where ρ is the pitch. Axis G $(g, \bar{g})$ can be found by:

$$g = \frac{\mathbf{c}}{\|\mathbf{c}\|}, \quad \bar{g} = \frac{\bar{\mathbf{c}} - p\mathbf{c}}{\|\mathbf{c}\|}, \quad p = \frac{\mathbf{c} \cdot \bar{\mathbf{c}}}{\mathbf{c}^2} \tag{10}$$

This results in a helical motion, as shown in fig. 4. The motion gives a better (quadratic) convergence behaviour than ICP for small residuals. Larger residuals give a convergence that is similar (linear) to ICP. The algorithm should account for small changes each time, therefore the algorithm is expected to work faster than ICP.

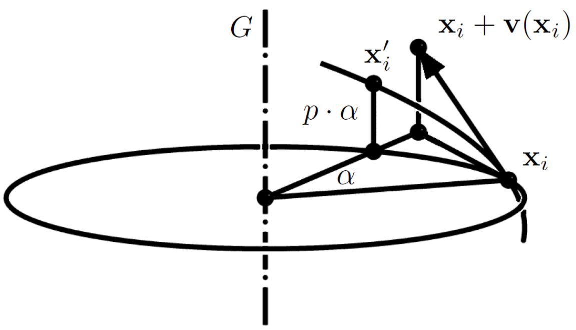

Figure 4: Transformation of x_i around axis G (Pottmann et al., 2004)

4.3.3 Hough Transform

The algorithm that is described by (Ni et al., 2013) finds the transformation matrix in five steps. In this research, two of these steps can be omitted. In the second step, they try to find the up-vector. The MH is automatically aligned to the ground plane, therefore already having the up-vector. Next to that, in the fourth step, the algorithm accounts for scale. The floor plan and the captured meshes both have the correct scale already. The remaining three steps are: (1) find planes in the captured point cloud (or mesh), making use of the continuous hough transformation; (2) Find rotation γ, making use of hough space; and (3) Find translation using cross-correlation.

For a finite point set X, that is analysed in continuous space, the Hough Transform equates:

$$H\{f_X(x)\}(\phi) = \sum_{x \in X} k(x, \phi) \tag{11}$$

In this equation, $k(x, \phi)$ is an exponential kernel function that relies on point-to-plane similarity. The multivariate estimate of planes that belong to the points can be found by:

$$f(\phi) = \frac{1}{N} \sum_{N} k_h(x, \phi) \tag{12}$$

Using an algorithm similar to the mean shift algorithm, this can be solved iteratively using the gradient of the function. The gradient is shown in equation 13. The first of the equation is the hough transformation, while the second part is the mean shift vector ($m_h(\phi^t)$). The algorithm computes the updated vector $m_h(\phi^t + 1)$ by $m_h(\phi^t + 1) = \phi^t + m_h(\phi^t)$.

$$\nabla_\phi f(\phi) = \frac{2}{CN} \left(\sum_i k_h(x_i, \phi) \right) \left[\frac{\sum_i x_i k_h(x_i, \phi)}{\sum_i k_h(x_i, \phi)} - \phi \right] \tag{13}$$

Finding the optimal rotation is an optimization problem. It can be solved through sweeping γ by shifting the Hough spectrum of the planes of the point cloud ($H^{(q)}$) and observing when it aligns the Hough spectrum of planes of the reference floor plan ($H^{(r)}$). The equation is as follows:

$$\arg\max_{\gamma} \sum_i \sum_{\rho_1} H^{(r)}(\theta_i, \rho_1) \sum_{\rho_2} H^{(q)}(\theta_i + \gamma, \rho_2) \tag{14}$$

In this equation, ρ_1 and ρ_2 are the offset from the center of the plane of the reference image and a plane in the point cloud respectively.

Lastly, the translation can be found by converting the rotation transformation back to euclidean space and using a conventional method to find the translational parameters. This can be done by a regular cross-correlation.

4.4 Positioning

Based on the characteristics of location and positioning defined by Sithole and Zlatanova (2016), it could be argued that this thesis is about positioning rather than localization. The method tries to estimate a point within a room, instead of the room itself. There are two positions that are calculated in the method. The SLAM of the MH computes the absolute coordinates that are defined by a Cartesian reference frame, relative to the origin of the mapped space. The proposed algorithm maps these coordinates to a position on the 2D floor plan. When the position on a 2D floor plan is found, this will remain a position rather than a location, since the output will be coordinates that refer to another Cartesian reference frame, relative to the floor plan itself. Not only does the method estimate the position of the device, it also finds the orientation. The position and the orientation together make the pose.

The position will be given together with an error metric, that estimates the probability that the device is on that given position. This will be based on the Mean Square Error [MSE] of the spatial matching algorithm output. Next to that, a probability that the MH is in another room should be available for the user at real-time. This will be based on MSE as well.

4.5 Evaluation

To evaluate the workflow, at certain places, a marker is placed on the ground that can also be found in the floor plan. This will serve as a reference point. At this point, the position of the Hololens will be saved. This position will be compared to the exact location of the marker. The same method will be used with and without the previously explained workflow, to compare the accuracy of both methods. Next to accuracy, the workflow will be tested on robustness. This will be examined for different kind of situations:

- New walls in the real world
- Old walls that do not exist anymore
- Walking to a different floor
- Walking through a door
- Big spaces without walls within 3m
- Rooms with many windows

4.6 Preliminary Results

The output of the SLAM algorithm can be seen in figure 5. The situation that is mapped is an office. The pink surfaces in the back are windows, while the purple surface on the right is a wall. While the device does give limited response around the windows, the meshes that are returned do show up as straight surfaces, indicating that they could be used for a spatial matching algorithm. The meshes of the wall are even better defined.

Figure 5: Microsoft Hololens SLAM

5 Time planning

See Attachment A for a Gantt chart with the time planning.

6 Tools and datasets used

The AR device used in this thesis is the Microsoft Microsoft Hololens. Deployment on the device will be done with help of `C#` and Unity. `C#` is also the environment that the algorithms will be written in, together with several modules. Floor plans will be modified with AutoCAD to fit the requirements. Various building sections of the TU Delft will be used as test cases, next to parts of the building of CGI, located in Rotterdam Alexander.

References

R. Abdalla. Evaluation of spatial analysis application for urban emergency management. *SpringerPlus*, 5(1):2081, Dec. 2016. ISSN 2193-1801. doi: 10.1186/s40064-016-3723-y. URL `http://springerplus.springeropen.com/articles/10.1186/s40064-016-3723-y`.

P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. pages 586–606, Boston, MA, Apr. 1992. doi: 10.1117/12.57955. URL `http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=981454`.

C. Brito, N. Alves, L. Magalhães, and M. Guevara. BIM MIXED REALITY TOOL FOR THE INSPECTION OF HERITAGE BUILDINGS. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W6:25–29, Aug. 2019. ISSN 2194-9050. doi: 10.5194/isprs-annals-IV-2-W6-25-2019. URL `https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-2-W6/25/2019/`.

J. Cyrus, D. Krcmarik, R. Moezzi, J. Koci, and M. Petru. Hololens Used for Precise Position Tracking of the Third Party Devices - Autonomous Vehicles. *Communications - Scientific letters of the University of Zilina*, 21(2), May 2019. URL `http://komunikacie.uniza.sk/index.php/communications/article/view/1464`.

A. Fonnet, N. Alves, N. Sousa, M. Guevara, and L. Magalhaes. Heritage BIM integration with mixed reality for building preventive maintenance. In *2017 24º Encontro Português de Computação Gráfica e Interação (EPCGI)*, pages 1–7, Guimaraes, Oct. 2017. IEEE. ISBN 978-1-5386-2080-9. doi: 10.1109/EPCGI.2017.8124304. URL `http://ieeexplore.ieee.org/document/8124304/`.

P. Hübner, M. Weinmann, and S. Wursthorn. MARKER-BASED LOCALIZATION OF THE MICROSOFT HOLOLENS IN BUILDING MODELS. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-1:195–202, Sept. 2018. ISSN 2194-9034. doi: 10.5194/isprs-archives-XLII-1-195-2018. URL `https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-1/195/2018/`.

P. Hübner, S. Landgraf, M. Weinmann, and S. Wursthorn. Evaluation of the Microsoft HoloLens for the Mapping of Indoor Building Environments. page 10, 2019.

K. Khoshelham, H. Tran, and D. Acharya. INDOOR MAPPING EYEWEAR: GEOMETRIC EVALUATION OF SPATIAL MAPPING CAPABILITY OF HOLOLENS. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13:805–810, June 2019. ISSN 2194-9034. doi: 10.5194/isprs-archives-XLII-2-W13-805-2019. URL `https://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XLII-2-W13/805/2019/`.

M. Kopsida and I. Brilakis. Markerless BIM Registration for Mobile Augmented Reality Based Inspection. page 7, 2016.

R. Newcombe, S. Izadi, D. Molyneaux, O. Hilliges, D. Kim, J. D. J. Shotton, P. Kohli, A. Fitzgibbon, S. E. Hodges, and D. A. Butler. Mobile Camera Localization Using Depth Maps, 2014.

K. Ni, N. Armstrong-Crews, and S. Sawyer. Geo-registering 3d point clouds to 2d maps with scan matching and the Hough Transform. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1864–1868, Vancouver, BC, Canada, May 2013. IEEE. ISBN 978-1-4799-0356-6. doi: 10.1109/ICASSP.2013.6637976. URL `http://ieeexplore.ieee.org/document/6637976/`.

H. Pottmann, S. Leopoldseder, and M. Hofer. Registration without ICP. *Computer Vision and Image Understanding*, 95(1):54–71, July 2004. ISSN 10773142. doi: 10.1016/j.cviu.2004.04.002. URL `https://linkinghub.elsevier.com/retrieve/pii/S1077314204000475`.

J. Rantakokko, J. Rydell, P. Strömbäck, P. Händel, J. Callmer, D. Törnqvist, F. Gustafsson, M. Jobs, and M. Grudén. Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization. *IEEE Wireless Communications*, 18(2):10–18, Apr. 2011. ISSN 1536-1284. doi: 10.1109/MWC.2011.5751291. URL `http://ieeexplore.ieee.org/document/5751291/`.

S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pages 145–152, Quebec City, Que., Canada, 2001. IEEE Comput. Soc. ISBN 978-0-7695-0984-6. doi: 10.1109/IM.2001.924423. URL `http://ieeexplore.ieee.org/document/924423/`.

S. Sharma, S. T. Bodempudi, D. Scribner, J. Grynovicki, and P. Grazaitis. Emergency Response Using HoloLens for Building Evacuation. In J. Y. Chen and G. Fragomeni, editors, *Virtual, Augmented and Mixed Reality. Multimodal Interaction*, volume 11574, pages 299–311. Springer International Publishing, Cham, 2019. ISBN 978-3-030-21606-1 978-3-030-21607-8. doi: 10.1007/978-3-030-21607-8_23. URL `http://link.springer.com/10.1007/978-3-030-21607-8_23`.

G. Sithole and S. Zlatanova. POSITION, LOCATION, PLACE AND AREA: AN INDOOR PERSPECTIVE. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-4:89–96, June 2016. ISSN 2194-9050. doi: 10.5194/isprsannals-III-4-89-2016. URL `http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/III-4/89/2016/isprs-annals-III-4-89-2016.pdf`.

T. van der Meer. 3d-indoormodellen voor de brandweer – een verkennend onderzoek naar de toepassingsmogelijkheden binnen de veiligheidsketen. Technical report, TU Delft, 2016.

T. van der Meer. *Geovisualization for the Dutch fire brigade. A research about effective cartographic methods for assisting tactics choice and indoor deployments during building fires.* PhD thesis, Utrecht University, Utrecht, 2018.

C. Yang and H.-r. Shao. WiFi-based indoor positioning. *IEEE Communications Magazine*, 53(3):150–157, Mar. 2015. ISSN 0163-6804. doi: 10.1109/MCOM.2015.7060497. URL `http://ieeexplore.ieee.org/document/7060497/`.

S. Zlatanova and D. Holweg. 3d GEO-INFORMATION IN EMERGENCY RESPONSE: A FRAMEWORK. page 6, 2004.

S. Zlatanova, G. Vosselman, K. Koshelham, R. Peters, B. Beers, R. Voûte, and H. Djurrema. Slimme 3d Indoormodellen ter Ondersteuning van Crisismanagement in Grote Openbare Gebouwen. page 10.