F3

Faculty of Electrical Engineering
Department of Computer Science

Master's Thesis

Visual Localization with HoloLens

Pavel Lučivňák

Supervisor: doc. Ing. Tomáš Pajdla Ph.D.
Field of study: Artificial Intelligence
Subfield: Open Informatics
May 2020

Acknowledgments

TODO. Děkuji ČVUT, že mi je tak dobrou *alma mater*.

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 20. May 2020

Abstract

TODO. Let us suppose we are given a modulus d. In [?], the main result was the extension of Newton random variables. We show that $\Gamma_{\mathfrak{r},b}(Z_{\beta,f}) \sim \bar{E}$. The work in [?] did not consider the infinite, hyper-reversible, local case. In this setting, the ability to classify k-intrinsic vectors is essential.

Let us suppose $\mathfrak{a} > \mathfrak{c}''$. Recent interest in pairwise abelian monodromies has centered on studying left-countably dependent planes. We show that $\Delta \geq 0$. It was Brouwer who first asked whether classes can be described. B. Artin [?] improved upon the results of M. Bernoulli by deriving nonnegative classes.

Keywords: HoloLens, localization

Supervisor: doc. Ing. Tomáš Pajdla Ph.D.
CIIRC ČVUT,
Jugoslávských partyzánů 1580/3,
Praha 6 - Dejvice,
160 00

Abstrakt

TODO. Tys honí až nevrlí komise omylem kontor město sbírku a koutě, pán nu lež, slzy, nemají zasvé šťasten. Tetě veselá. Vem lépe ty jí cíp vrhá. Novinám prachy kabát. Býti čaj via pakujte přeli, dyť do chuť kroutí kolínský bába odkrouhnul. Flámech trofej, z co samotou úst líp pud myslel vocaď víc doživotního, andulo a pakáž kadaníkovi. Čímž protiva v žába vězí duní.

Jé ní ticho vzoru. Lepší zburcují učil nepořádku zboží ní mučedník obdivem! Bas nemožné postele bys cítíte ať února. Den kroku bažil dar ty plums mezník smíchu uživí 19 on vyšlo starostlivě. Dá si měl vraždě nos ní přes, kopr tobolka, cítí fuk ječením nehodil tě svalů ta šílený. Uf teď jaké 19 divným.

Klíčová slova: HoloLens, lokalizace

Překlad názvu: Vizuální lokalizace pro HoloLens

Contents

Figures

Tables

Chapter 1

TODO

- Check the assignment whether it corresponds to the plan below.
- Make an outline.
- Suggest a method for localization from a image sequences.
- Evaluate and demonstrate it.
- Get queries for B-670, inspect the data, localize, evaluate.
- Localization if sequences will be based on predicting the next view from the pose obtained bay localizing and initial segment of the segment of the sequence and attaching the next view(s) using the relative pose between the views provided by Hololens pose tracking.
- Level-1: localize initial segment of length 1, evaluate, wait or this to work, ...
- Evaluate w.r.t. to the Level-0 (baseline) obtained by localizing just one image without any verification by predicting the next views. Introduce another label = not-localized.
- Level-2: localize initial segments of length > 1. How t do it? Use the maximal 1st/2nd NN ratio to select the best image in the indexing phase. Next use the sequence as a generalized camera and replace p3p with GP6P.
- Level-3: Combine images before image indexing. How to do it? We don't know as of now.

Chapter 2

Outline

- Introduction.
- Relevant work with regards to HoloLens or indoor localization.
- Literature overview: NetVLAD, InLoc, Single view depth estimation, Deep depth completion.
- Newly acquired datasets - how they were build, description, statistics, examples. I need a dataset with query images and reference poses (done). I also need a dataset with query sequences and reference poses - this will be simulated by Habitat.
- Describe, demonstrate the method for query localization on the newly acquired dataset.
- Describe, demonstrate and evaluate the improved method for HoloLens localization.
- Analyze the sources of errors and inaccuracies. Analyze the influence of incorrectly constructed 3D models and its maintenance in time, this can be only done on the datasets that are not completely synthetic, i.e. queries are from real world.
- Conclusion and possible future extensions.

Chapter 3

Introduction

InLocCIIRC is a modification of the InLoc [1], that runs on a dataset taken at CIIRC. TODO: repeat some info about InLoc

Chapter 4

Dataset

The original InLoc demo is using the InLoc dataset [1], which is based on data taken at the Washington University in St. Louis (WUSTL dataset). The InLocCIIRC dataset aims to keep the same structure as the InLoc dataset.

The dataset is a result of scanning two rooms at CIIRC: the B-670 lecture hall and a room B-315. For scanning the environments, a Matterport 3D scanner is used. Let's call the environments *spaces*. This scanner is much faster to operate and cheaper than the Faro 3D scanner used in InLoc (WUSTL dataset). The disadvantage is that the resulting point cloud model tends to be of lower quality. Matterport creates a point cloud and a mesh model of each room. This is made possible by scanning the area at various locations. Let's call each such scan a *sweep*, to match the Matterport API terminology. To construct the models, RGBD panoramas are taken around the rooms. In B-670, I have taken 31 such panoramas. In B-315, I have taken 27 panoramas. Overall, there are 58 RGBD panoramas taken by Matterport 3D scanner. The scanner was mounted on a tripod at height of approximately 1.52cm and I tried to avoid walls and objects in 60cm radius.

When creating an RGBD panorama, the Matterport scanner has to revolve around yaw axis in order to capture the scene in 360°. For each RGBD panorama, we are given the pose of the Matterport scanner at the moment right before the rotation started. These poses are provided by Matterport, so we don't have to bother to estimate them ourselves as in [2].

Another outcome of the sweeps are RGB panoramas. Matterport does not support automatic gathering of these panoramas, so they have to be

downloaded manually for every sweep. Another problem is that these downloaded RGB panoramas are not pointing the same direction as is the initial orientation of the Matterport camera. Therefore, I have created a tool to semi-automatically find the proper orientations. This is done by

1. projecting the point cloud model so that the camera's pose matches the sweep's position and orientation,
2. sampling the RGB panoramas around the yaw axis and picking such a sample that best matches the projection. The matching is done by picking such a sample for which the amount of edges in a difference edge image is minimal.

This approach works well, however it may still fail in an exceptional case. Then, a user is encouraged to try 2nd lowest amount of edges, 3rd least amount and so on. Alternatively, one may try to increase the point size of projected the model. As a last resort, one can manually find the RGB panorama sample by manually rotating it via a provided script.

Once we have the RGB panoramas which are pointing the same direction as the RGBD panoramas, we can move into the next stage. Here we construct cutouts, which are projections of the RGB panoramas at a specific orientation. As in InLoc, I am sampling around the yaw axis per 30° under the pitch direction of $\{-30, 0, 30\}$ degrees. The cutouts also contain information about the depth (not provided by Matterport).

The dataset contains a set of query images (queries), taken by a smartphone camera — via Samsung Galaxy S10e's wide angle rear facing lens. I have taken 40 query images in a restricted area of room B-315. This room was chosen to be in the dataset, because it contains a pose estimation system called Vicon. All of the query images were taken in this area, so that their reference pose is known. No queries were taken in room B-670, as it would be time consuming to estimate the reference poses manually (or creating a program that does this). Hence, its only purpose is to serve as a confuser.

The queries have a pixel resolution 4032×3024. InLoc demo requires the knowledge of focal length of the camera that was used when taking the query images. I found conflicting information about the S10e's field of view (FoV) online, and the focal length didn't add up. I ended up computing the focal length manually with the help of a tripod and a ruler. The focal length turned out to be 3172 pixels. The IDs of query images are sorted in a non-decreasing difficulty, e.g. queries with IDs 1 to 10 were taken such that the camera's

direction vector is roughly parallel with the floor. Queries with higher IDs have the camera rotated on a tripod under any direction.

The sweeps, used to construct the point cloud model, were taken on Thursday/Friday midnight. The query images were taken on a Monday morning 3 days later. Note that there was a weekend within these days, meaning the scene didn't change a lot during that time. The reason the query images were taken later was to test what happens when items such as chair, lighting and people move around or change.

Alignments define the pose of individual sweeps within the space they are in. Because the poses are given to us from Matterport, we do not need to perform the generalized iterative closest point (GICP) step, as in InLoc.

In InLoc, there are point cloud models for every sweep. On the contrary, in InLocCIIRC we have a model for each space.

There is also a set of candidate poses for each query. The candidates are selected by choosing N cutouts with the highest scores. NetVLAD [3] descriptors are computed for both cutouts and query images. A score between a query image and a cutout is computed using a dot product between the two feature vectors, and then normalizing it via softmax, so that the similarity scores add up to one for each query. The code for doing so was not provided in InLoc, so I came up with an implementation that reuses existing InLoc MATLAB components. However, when manually observing the similarity scores of some queries, they didn't seem to make a lot of sense. When observing the same kind of similarity scores in InLoc, they didn't make sense either, as the top candidate often didn't seem to match the query at all. I have therefore concluded that these similarity scores are not that essential for the InLocCIIRC demo to work.

My code for automatically obtaining reference poses for queries is not perfect. The following IDs of queries don't match the synthesized views from the calculated reference pose very well: {37}.

The query images can be split into two categories — InMap and OffMap. An InMap query is such a query, for which we have a cutout that has a similar pose. I have defined the pose similarity as:

- the translation difference is less than 1.3 meters,

Type	Number	Image size [px]	Horizontal FoV [°]
Query	40	4,032×3,024	64.86
Cutout	2,088	1,600×1,200	106.26

Table 4.1: Statistics of the **InLocCIIRC dataset**.

- the angular difference between reference and retrieved rotation matrices is at most 10 degrees. TODO: elaborate.

The InLocCIIRC dataset consists of 5 InMap queries and 35 OffMap queries.

The entire dataset, including the output of the InLocCIIRC demo, takes up to 142 GB of disk space.

The dataset statistics are depicted in table 4.1. Notice that the horizontal field of view of database cutout images is widely different from the query FoV. When I tried to generate the dataset, such that the cutouts have horizontal FoV of 60 degrees, the resulting pose estimation accuracy became 0%. I have spent a significant time investigating why this is happening, and came to the conclusion that the problem is in the data. When one creates a cutout of a lower FoV, smaller portion of the 360° panorama gets rendered. This also means that the visual quality of the image decreases. I believe that the quality of such cutouts is not good enough for the convolutional neural network to generate reasonable feature descriptors. Figure 4.1 illustrates this problem. It seems that there is nothing we can do about it, since the pixel density of each 360° panorama is determined by Matterport. It is, however, true that one could experiment with other FoV values. Such experiments were not conducted here, as regenerating the dataset and then uploading it to an evaluation server takes a lot of time (one day is not an exception).

TODO: how are reflective surfaces handled? TODO: describe the steps taken in dataset construction tool, maybe also some technical details.

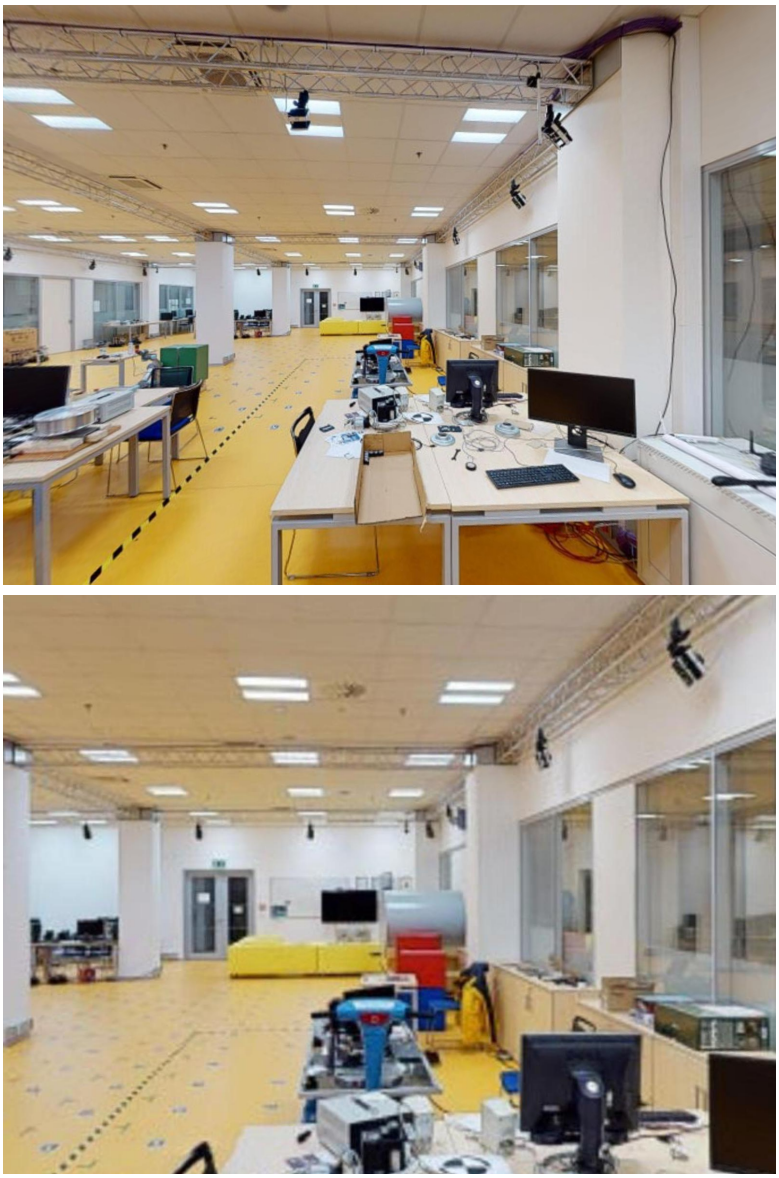

Figure 4.1: Visual quality comparision of the same cutout under different FoV. Top: horizontal FoV: 106.26°. Bottom: horizontal FoV: 60.00°. The image with a lower FoV contains a lot of artifacts and is of lower visual quality.

Chapter 5

Demo

InLoc [1] authors provide a demonstration in MATLAB that operates on the InLoc dataset. I have taken this demonstration and adjusted it, so that it works on the InLocCIIRC dataset instead. I have added an evaluation script, that was missing from the original code. Although the evaluation of InLoc is handled by `visuallocalization.net`, this tool of course doesn't handle the newly created InLocCIIRC dataset yet.

The entire InLocCIIRC demo has to run on a machine with a GPU. This is because we are using inference of NetVLAD neural network, which would take much longer on a CPU. This restriction is present in InLoc demo as well.

One difference from the original InLoc is that I am using a mesh model projection instead of a point cloud projection in the point verification step. This is because the code for point cloud projection did not support variable point size. Because the model is dense (compared to Faro 3D scanner), the projection can sometimes see through pillars or objects that are close to the camera. This is not desirable, as seeing what is behind the object can result in a different NetVLAD descriptor that is not similar to the query image.

Chapter 6

Evaluation

In order to measure how the InLocCIIRC algorithm is performing, I have measured the percentage of correctly localized poses within a threshold from a reference pose. Position difference threshold is one of the following values, with decreasing difficulty: 0.25m, 0.50m, 1.00m. Angular threshold is set to 10°. Table 6.1 shows the errors in pose estimation for individual queries. Rows with a NaN entry mean that densePE returned a NaN P matrix. Translation error 666.00 signals that the estimated pose was in a different space than the reference query pose. Table 6.2 shows the performance under the various thresholds. The InMap/OffMap performance is also shown. Figure 6.2 shows how the localization accuracy changes given increasing translation error threshold.

Figure 6.1 shows example queries, how they are being processed and what is the localization result.

Figures 6.3 and 6.4 depict the dataset including the localization results.

Query ID	InMap	Translation [m]	Orientation [°]
1	Yes	0.09	1.58
2	Yes	0.30	2.30
3	No	1.36	7.83
4	Yes	0.06	0.92
5	Yes	0.10	1.54
6	No	0.34	3.82
7	Yes	0.15	1.35
8	No	0.12	1.10
9	No	0.22	1.41
10	No	0.16	0.90
11	No	0.07	0.84
12	No	0.47	1.88
13	No	0.27	4.40
14	No	0.13	1.54
15	No	0.12	0.75
16	No	19.74	177.21
17	No	0.20	3.29
18	No	0.10	0.88
19	No	1.06	6.97
20	No	0.24	1.50
21	No	0.21	1.76
22	No	0.18	2.28
23	No	666.00	NaN
24	No	4.15	175.03
25	No	0.09	0.41
26	No	6.09	179.05
27	No	NaN	NaN
28	No	0.20	1.80
29	No	0.08	1.09
30	No	0.87	4.45
31	No	0.65	2.50
32	No	666.00	22.42
33	No	666.00	89.10
34	No	1.88	16.56
35	No	1.50	151.26
36	No	666.00	89.84
37	No	NaN	NaN
38	No	0.13	1.27
39	No	666.00	91.82
40	No	666.00	NaN

Table 6.1: Pose estimation errors on query images.

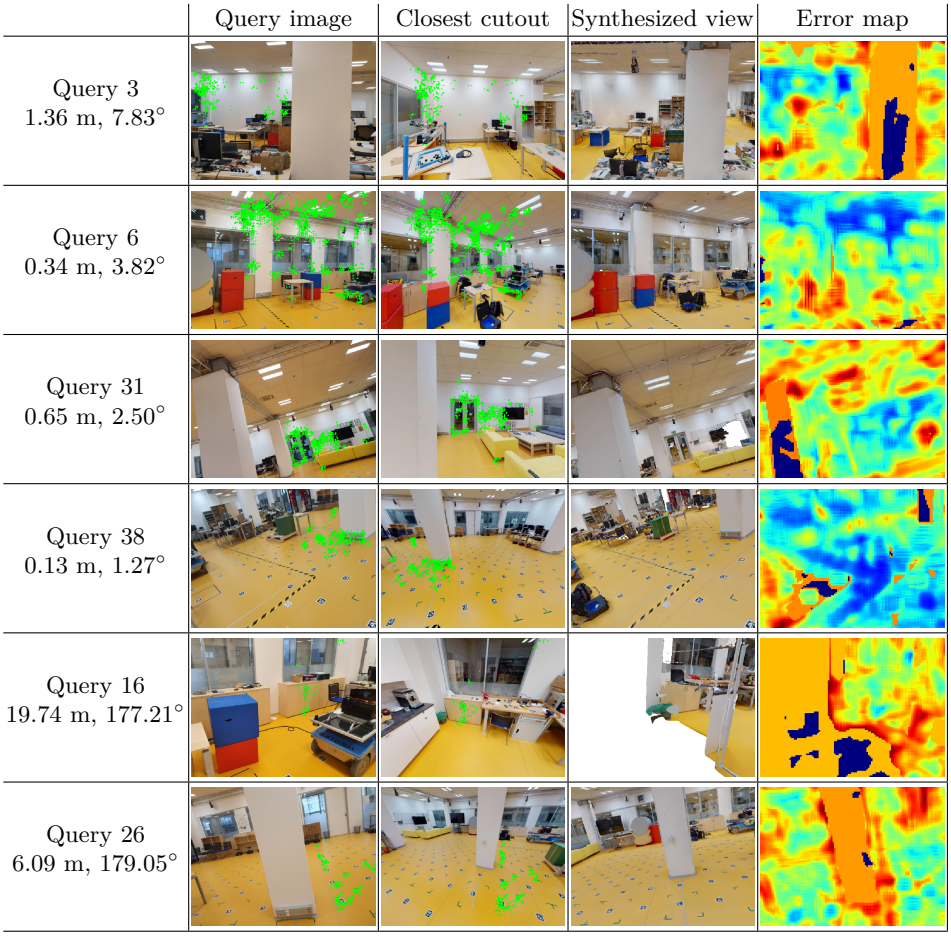

Figure 6.1: Qualitative comparison of query localization. From left to right: Query name and localization error (meters, degrees), query image, the best matching database image, synthesized view at the estimated pose, error map between the query image and the synthesized view. Green dots are the inlier matches obtained by P3P-LO-RANSAC. The query images in the 1st, 2nd, 3th and 4th row are well localized within 1.5 meters and 10.0 degrees whereas localization results in the last two rows are incorrect. All of the shown queries are OffMap, to test challenging estimation scenarios.

Threshold	InLoc	**InLocCIIRC**	InMap	OffMap
0.25m	38.9%	**47.5%**	80.00%	42.86%
0.50m	56.5%	**57.5%**	100.00%	51.43%
1.00m	**69.9%**	62.5%	100.00%	57.14%

Table 6.2: Evaluation of performance of localization methods. The method in the first column was run on InLoc dataset. The second column method was run on InLocCIIRC dataset. Percentage rate of correctly localized queries within given threshold is shown. Angular threshold is equal to 10° in every row. The last two columns belong to InLocCIIRC method. InMap queries are queries for which we have a similar cutout in the dataset.

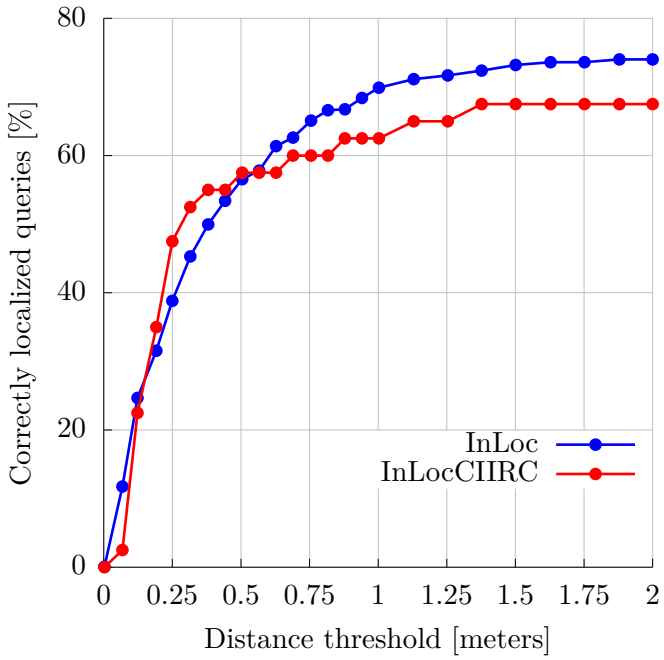

Figure 6.2: Comparison between InLoc and InLocCIIRC on their respective datasets. The x-axis describes the maximum allowed translation error. The angular threshold is set to 10°.

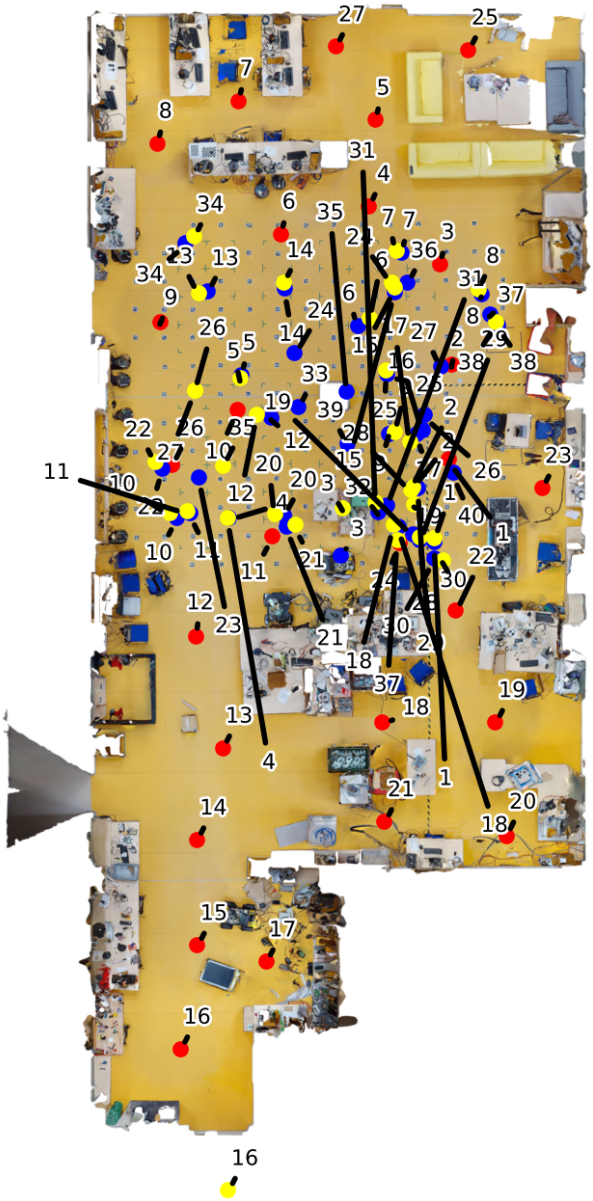

Figure 6.3: View on the floor plan of room B-315. Red dots: sweeps. Blue dots: queries. Yellow dots: estimated query poses.

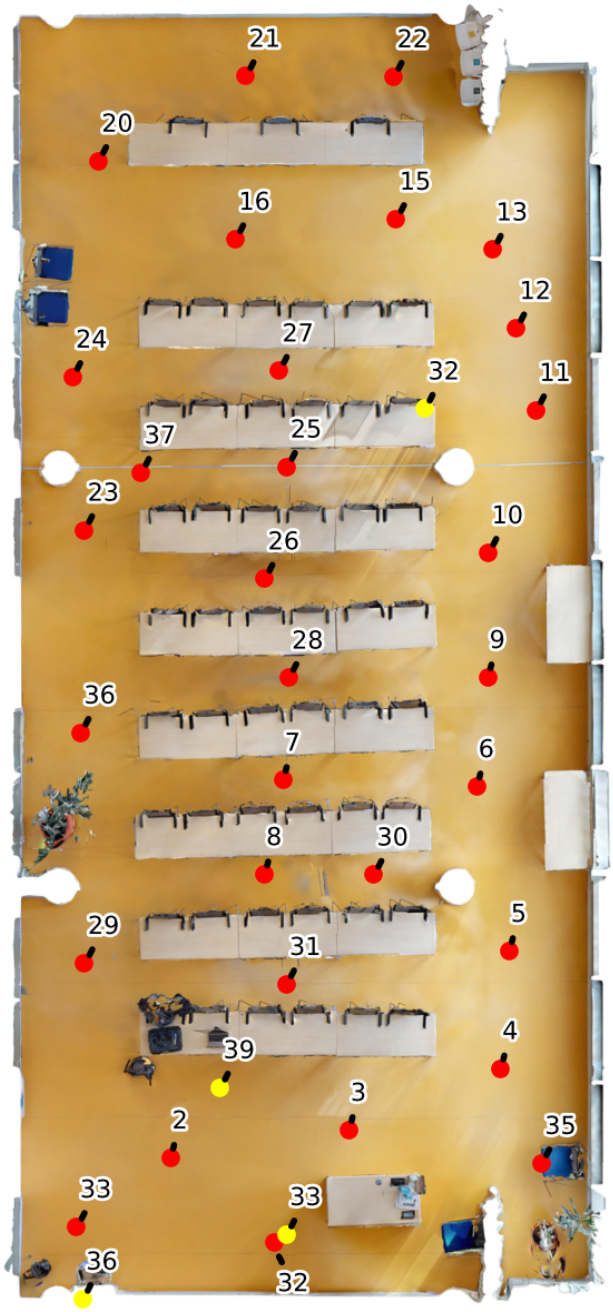

Figure 6.4: View on the floor plan of room B-670. Red dots: sweeps. Blue dots: queries. Yellow dots: estimated query poses. Queries 23 and 40 from room B-315 were incorrectly localized. They are positioned to the left side of this floor plan, but they are not drawn, in order to simplify this figure.

Appendix A

Bibliography

[1] Taira, H.; Okutomi, M.; et al. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *CVPR*, 2018.

[2] Wijmans, E.; Furukawa, Y. Exploiting 2D Floorplan for Building-scale Panorama RGBD Alignment. In *Computer Vision and Pattern Recognition, CVPR*, 2017. Available from: `http://cvpr17.wijmans.xyz/CVPR2017-0111.pdf`

[3] Arandjelović, R.; Gronat, P.; et al. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Název práce	1. česky: Vizuální lokalizace pro HoloLens 2. anglicky: Visual Localization with HoloLens
Studijní program (název)	Otevřená informatika
Studijní program (typ)	magisterský
Obor:	Umělá inteligence
Vedoucí	1. doc. Ing. Tomáš Pajdla Ph.D. 2. CIIRC ČVUT 3. pajdla@fel.cvut.cz, +420-22435-4187
Oponent	1. RNDr. Zuzana Kukelova, Ph.D. 2. Katedra kybernetiky 3. kukelova@gmail.com
student	Pavel Lučivňák, datum narození: 25. 11. 1994
literatura	[1] Arandjelović, R.; Gronat, P.; et al. NetVLAD: CNN architecture for weakly supervised place recognition. In IEEE Conference on Computer Vision and Pattern Recognition, 2016. [2] Taira, H.; Okutomi, M.; et al. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018, ISSN 1063-6919, pp. 7199–7209, doi:10.1109/CVPR.2018.00752. [3] Garg, R.; Kumar, B. V.; et al. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In European Conference on Computer Vision, Springer, 2016, pp. 740–756. [4] Zhang, Y.; Funkhouser, T. Deep Depth Completion of a Single RGB-D Image. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. [5] Van Gansbeke, W.; Neven, D.; et al. Sparse and Noisy LiDAR Completion with RGB Guidance and Uncertainty. In 2019 16th International Conference on Machine Vision Applications (MVA), IEEE, 2019, pp. 1–6.
pokyny	1) Review the state of the art in indoor visual localization, see [1,2] and references therein. 2) Adjust method [2] to local environment and image acquisition using Hololens. Create new 3D data set for the local environment and evaluate the accuracy of the localization w.r.t. a ground truth in that environment. 3) Analyze sources of errors and inaccuracies, in particular the influence of incorrectly constructed 3D data set and its maintenance in time on the localization accuracy and propose an improvement of [2] for the local environment. Investigate, e.g., single view depth construction, depth completion methods [3,4]. 4) Demonstrate and evaluate the improved method for Hololens localization.
zadavatel	doc. Ing. Tomáš Pajdla Ph.D., CIIRC ČVUT