Visual Localization with Environment Outline Prior

Vojtěch Pánek[1,2]

[1]Czech Institute of Informatics, Robotics and Cybernetics, Czech Technical University in Prague
[2]Faculty of Electrical Engineering, Czech Technical University in Prague

Abstract

Visual localization is the task of estimating the camera position and orientation relative to a given environment representation, which is used in applications such as mobile robotics, autonomous driving, and augmented reality. Currently, the environment has to be scanned by a human operator or explored by an autonomous system to create a precise and detailed map, often taking a large amount of storage space. Our approach aims to circumvent the scanning phase and decrease the memory footprint by using a publicly available lightweight environment representation in the form of building floor plans or human-readable digital maps. We extract the outline of the environment visible in images and use hierarchical localization on top of the layout representation, using image retrieval for coarse pose estimate and refining the pose by ICP algorithm. We also try to entirely replace the images in the retrieval step with the extracted outline geometry. The initial experiments show that the approach might achieve reasonable performance while keeping the storage needed for environment and image representation at a minimum.

1 Introduction

The visual localization task aims at camera pose estimation based on given image data. Thanks to mobile robotics, autonomous driving, and augmented reality applications, the research problem is currently fairly popular. Apart from input image data, each visual localization system needs prior knowledge about the deployment space, i.e., an environment representation. Such representation can have a form of a point cloud, mesh, image set, can be implicitly stored in a neural network, or have a form similar to a human-readable map. The last-mentioned approach is particularly interesting as it is often already available and can be shared and easily adjusted by both humans and machines. We, therefore, try to develop a visual localization system that could work on top of the human-readable representations in the form of floor plans and maps of urban areas.

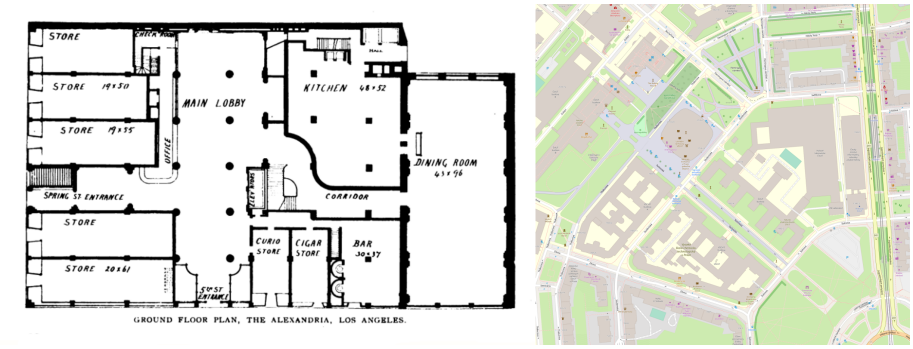

Figure 1: Examples of human-readable maps: left - floor plan, right - outdoor map of an urban area (Open Street Maps).

1.1 Related work

The current state-of-the-art visual localization approaches are often based on structure-based hierarchical pipelines [23]. The environment is represented by a set of database images with known poses and a triangulated point cloud with appended local images features. First, a coarse pose estimate is obtained, *e.g.*, by using image retrieval based on image-level descriptors [1, 15] to find similar database images with known poses. The coarse estimate is then refined in the second step by performing local image features matching [17, 22, 25] between the query image and the triangulated points followed by camera pose estimation. The pose estimate is done by solving a minimal geometric problem in RANSAC [4, 11] scheme.

To get the database images, the environment has to be visited before the deployment or well covered by publicly available imagery. The creation of a feature point cloud by an SfM pipeline [24] can become highly time-consuming, especially for larger scenes. There exist several works which try to overcome these issues by using other scene representations, such as floor plans or building outlines, which are often readily available without the need for scene capturing. Moreover, the pure geometry-based representations are much more robust relative to image-based representations, which are subject to illumination, weather, and occlusions.

The main issue for geometry-based visual localization systems comes with alignment between the two different data modalities. The scene representation is often purely geometry-based, but the input is still an image whether RGB only [2, 3, 8, 9, 12, 13, 16, 20, 27] or with depth measurements [9, 18, 19, 28]. Different solutions to the modality issue comprise usage of line features in the image domain [2,3,8,27], aligning depth measurements to the scene layout [14, 19, 28] or applying SfM on RGB image sequences and aligning the extracted geometry [9, 12, 20]. We also try to align the 3D ge-

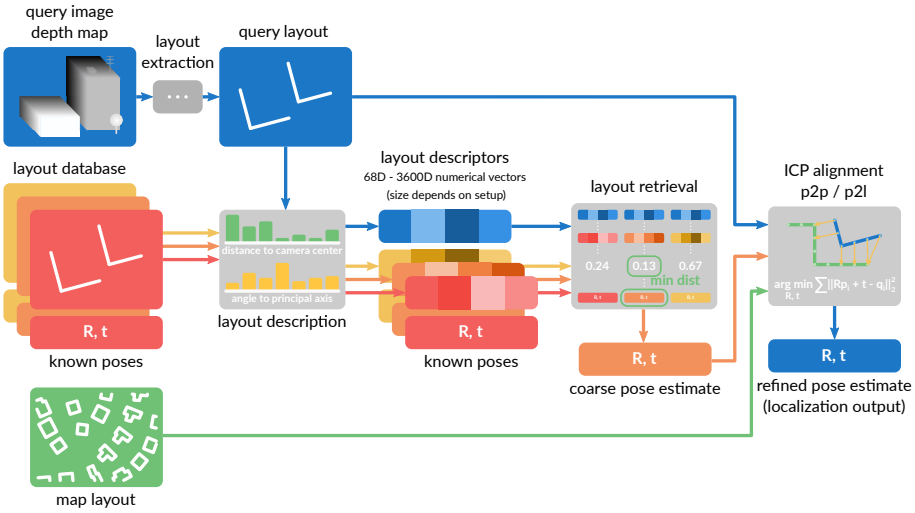

Figure 2: Architecture of the whole localization pipeline with layout retrieval. See Fig. 3 for the description of layout extraction procedure.

ometry extracted from the images to the scene, but just after distillation to get the same modality for both image and environment representation.

Some approaches are using semantic scene understanding to extract only relevant geometry from the scene [2, 3, 5] or to get higher-level features such as doors, windows, or text labels, which might be present in the scene representation [16, 19, 27]. The potential camera poses can be tested all in a grid-like manner [2, 3, 13], or the space of candidates can be filtered out, *e.g.*, by Monte Carlo Localization approach [6, 14, 18, 19, 28]. Our approach reduces the potential space by using the retrieval to find a coarse pose estimate.

2 Localization Using Environment Layout

This section describes the developed pipeline, composed of layout extraction from depth images, retrieval step for obtaining coarse pose estimate, and layout alignment step for pose refinement.

2.1 Layout Extraction from Images

To match a query image to a given layout map, we need to extract the layout visible in the image. The first idea was to extract the 3D geometry visible in the image by employing a state-of-the-art monocular depth estimator [21]. The following part of our layout extraction relies on planar region extraction from the depth estimates. Unfortunately, the initial experiments on

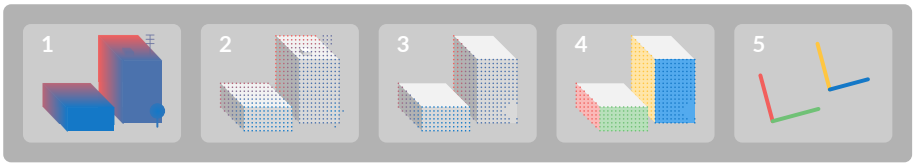

Figure 3: The process of layout extraction from an image: **1** The depth map is either directly obtained from a stereo camera or estimated from an RGB image, **2** depth map is reprojected to 3D, **3** points belonging to horizontal planes and non-planar structures are filtered out, **4** the remaining points are divided into individual planar segments, and the plane parameters are computed, **5** the outlines in the camera coordinate frame are generated by intersecting the extracted plane segments with a selected ground plane.

the depth extractors showed large geometry distortion, which would make the plane extraction impossible. Therefore we abandoned the idea of layout generation from a single RGB image until we find another reliable way of geometry extraction and we implemented and tested the rest of our pipeline on RGB-D data, which already contains high-quality geometry measurements.

Once we have the scene geometry in the form of a depth image, we can transform it to a point cloud, compute point normals and estimate gravity direction. We assume that the majority of the visible scene adheres to the Manhattan world assumption, i.e., it is composed of planes of three major, mutually perpendicular, orientations. This assumption is reasonable as the shape of most urban buildings is composed of a set of cuboids, and the same applies to the majority of indoor spaces. The second assumption is that the images are taken in an approximately horizontal direction, and therefore, the majority of the visible planes in the scene are walls spanning through two major Manhattan orientations. The third perpendicular orientation corresponds to the ground or ceiling. The normal of the found ground plane can be used as an initial estimate of the gravity direction vector, which is then iteratively refined.

With the knowledge of gravity direction, we can filter out parts of point cloud belonging to horizontal planes, which do not have any equivalent in 2D map representations. The rest of the points are divided into the two remaining Manhattan directions, and DBSCAN [10] algorithm is used to extract individual planar regions. The regions are filtered by their size to eliminate small patches, which usually correspond to non-planar geometry, and plane parameters are computed for the remaining regions. In the end, the planes are intersected by a horizontal plane at the height of the camera center to get the layout lines, which are limited to line segments based on the planar region boundaries along the line. The layout can be further refined by merging near colinear line segments. Note that the layout extracted from

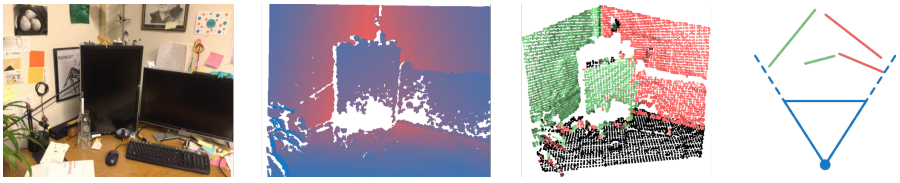

Figure 4: A simple example of layout extraction on real data. From left: RGB image, depth map (from a stereo camera), reprojected point cloud with color coding based on the three major Manhattan directions, estimated 2D layout in camera coordinate frame.

an image is represented in the corresponding camera coordinate frame.

As the field of view of a single image can be especially indoors relatively small, capturing only a limited part of the scene geometry and making the alignment difficult or impossible, we also tested integrating the geometry over a sequence of consecutive image frames. The point clouds extracted from a sequence of depth images can be registered to create a single point cloud with geometry visible in the images along the sequence. This joint point cloud can then be used to extract the layout for the whole sequence. Of course, this method can be used only when the consecutive images have an overlap large enough, so the registration does not fail.

2.2 Layout Maps of Environment

To align the layout extracted from an image to the correct location in the map, we need to represent the environment in the same way, i.e., a set of line segments with known parameters. Some input formats such as 2D CAD drawings, publicly available maps, or other vector graphics formats can be used directly. In the case of rasterized formats like images, hand drawings, or occupancy maps, we have to vectorize the map to get precise line segment parameters usable for alignment. Line detection in images is a problem with well-matured solutions [7], as the task is reliably solvable even with standard image processing methods (without deep learning).

Environment representations in usable format are usually available, reducing the need for scanning or manual creation of the maps before system deployment. The majority of buildings have some construction documentation, often containing CAD drawings of the interior layout. Most urban areas are well covered by publicly available maps, such as Open Street Map (OSM). OSM also provides an API that can easily be used for filtering and downloading the building outlines right in the vectorized format.

2.3 Retrieval Stage

Image retrieval is the task of searching for the most similar database image to the query image given as the input; specifically, our requirement is to find an image capturing the scene from as similar pose as possible. As the database images have known camera poses, we can use the pose of similar database images as a rough estimate for the query image camera pose.

Direct comparison of pixel values of two images is not robust enough because of illumination, seasonal and weather changes, and occlusions caused by objects such as people, vehicles, or furniture. The standard solution is to create a representation of the image in the form of a numeric vector, which would be robust enough and make the computation of similarity between two images easy. We are using NetVLAD [1] deep-learning-based image-level descriptor, which is known to work reasonably well for the localization task both indoors and outdoors. The dissimilarity between two images can be simply computed as the Euclidean distance between the two descriptors of the images.

Using the image retrieval for rough pose estimation has the disadvantage of higher storage space consumption as the database images and potentially their high-dimensional descriptor have to be stored. On the other hand, the layout extracted from the images are very compact as we are storing just endpoint coordinates of a small set of line segments, and therefore the idea to use the extracted layouts for retrieval task sounds compelling.

We tried to develop a simple descriptor for the layouts, which would capture the essence of line segment set geometry and allow direct similarity comparison between two layouts. The descriptor takes the distances of the line segments from the camera center and their angles from the optical axis and creates histograms of the values with a selected number of bins. Multiple tricks can be applied on top of the histograms to improve retrieval performance. We tested a various number of bins for both histograms, smoothing of counts over neighboring bins, soft assignment of values, binarization of the bin values, normalization of values over histograms, and creation of joint histogram for distance and angle. Euclidean distance is again used to compute the dissimilarity score between two layouts.

2.4 Layout Alignment

The alignment of the image layout to the map layout is done by ICP algorithm. The basic idea of ICP is to iteratively find the nearest point from point cloud B to each point from point cloud A and apply a proper rigid transformation on point cloud A so that the distance between corresponding points from A and B is minimized. In our case, we do not have point clouds but line segment sets, but we can easily transform the line segments to point clouds by sampling points along the lines. We also tried to apply the point

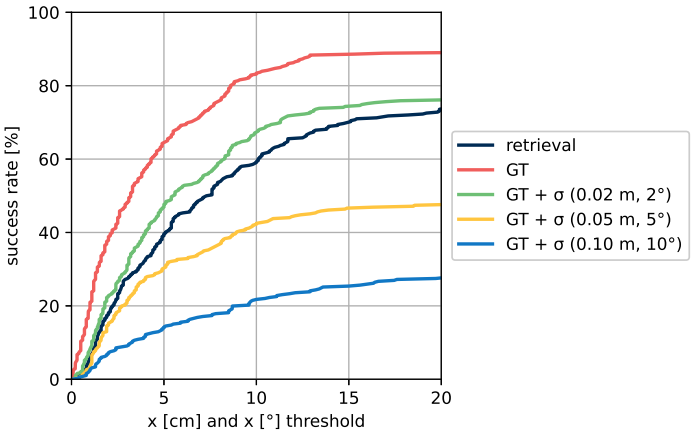

Figure 5: We show the localization error of pipeline with point-to-point ICP alignment, initialized from ground truth camera pose, and for three levels of random error added to the ground truth pose. The first (dark blue) curve shows ICP performance when initialized by image pose estimate from NetVLAD-based image retrieval. Evaluated on office1/gates362 scene.

sampling only on one of the sets and using point-line or point-line-segment distance metrics in the ICP loop.

As stated above, we store the image layout in the local coordinate frame of the camera, and therefore, once we align the image layout to the map, we can easily derive the camera pose relative to the map coordinate frame.

3 Experimental Evaluation

We evaluated the localization pipeline on the 12 Scenes dataset [26], capturing 12 room-scale scenes by an RGB-D sensor. As the 12 Scenes dataset does not provide a layout of the rooms, we extracted the layout from room point clouds in a similar manner as we do with images. The performance of localization is evaluated by plotting localization success rate curves at given error thresholds, i.e., the percentage of query images that have localization error (Euclidean distance and rotational difference) under given thresholds.

The first set of ablation experiments focused on the evaluation of the ICP alignment algorithm with a point-to-point distance metric. We initialized the alignment in the ground truth pose and gradually increased a random initialization error to see how the ICP would be able to cope with decreasing initialization quality. The results in Fig. 5 show that the alignment performance significantly falls together with the increasing initialization error, and even small initialization error results in a significant decrease in alignment

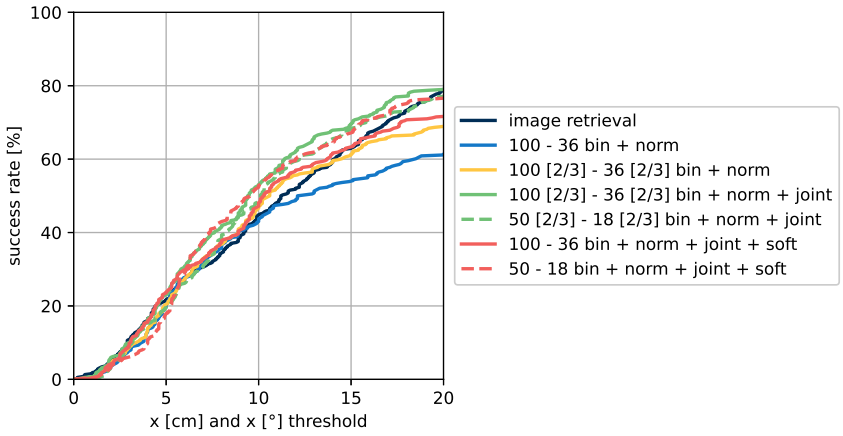

Figure 6: Performance evaluation of pipeline local layout retrieval stage. The first value marks the number of distance bins, the second number of angular bins, both might have a bin overlap (in square brackets), or the values can be assigned in the soft manner (`soft`). The bins can be structured to form a single joint histogram over all possible distance-angle combinations (`joint`). The values in the histogram can be binarized (`bin`) and normalized to unit sum (`norm`). Evaluated on office1/gates362 scene.

quality. On the other hand, the initialization by image retrieval is working reasonably well and is not as far behind the ground truth initialization as expected. Note that the quality of rough pose estimate of retrieval is greatly influenced by the spatial distribution of database camera poses relative to the query camera poses, as the best possible query pose estimate is always the pose of the nearest database camera.

The second set of experiments focused on searching for optimal layout descriptor parameters. As the number of tested parameter values is large, Fig. 6 shows only a selected subset with reasonable localization performance. The curves mark localization error of retrieval part of the pipeline only, i.e., without the ICP pose refinement. A surprising result is that layout-based retrieval can achieve similar performance as image-based retrieval, which needs more storage space by several orders of magnitude. The best-performing layout descriptor setup used 100 line distance and 36 line angle bins in the joint setup; therefore, the final histogram had 3600 bins in total. Both the distance and angular bin values have 2/3 overlap, which means that a single value will be assigned to three neighboring distance and three neighboring angular bins. This approach is an alternative to soft assignment, where the values assigned to bins are inverse to the relative distance from the two neighboring bin centers. The last two tested parameters were bin

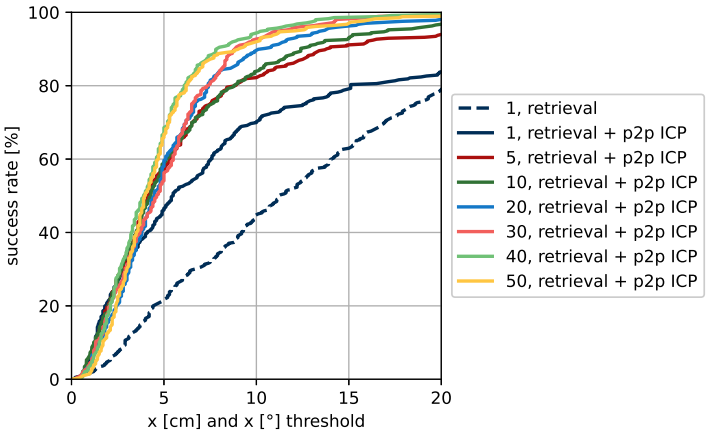

Figure 7: Comparison of pure image retrieval vs. hierarchical pipeline on top of an image sequence with increasing length. As the 12 Scenes dataset [26] is captured as a sequence of video frames, consecutive frames were used for temporal geometry integration. Evaluated on office1/gates362 scene.

value binarization to either zero or one and normalization, so the values over the whole histogram sum to one. The other selected parameter sets do not fall far behind the best performing, but we can see a minor advantage of soft assignment or bin overlapping over the case when none of these techniques is used.

The last experiment we show in this paper was performed to find out how the localization performance depends on the size of the part of the scene visible for the pipeline. The results in Fig. 7 show that increasing the visible part leads to a decrease in ambiguity both during the retrieval and alignment steps. The first step from using only a single image to the usage of 5 consecutive frames adds around ten percent, and integrating the geometry over more frames further increases the performance. The increasing trend seems to stop at 40 frames, and merging geometry from 50 frames does not increase the success rate anymore. A similar effect could be achieved by using a camera with a wider field of view lens.

4 Conclusion

Our paper deals with visual localization using environment layouts. We developed a prototype of a hierarchical localization system, which uses image or layout retrieval for rough pose estimate and an ICP algorithm for precise pose refinement. The performed experiments show that the pipeline is capable of pose estimation at reasonable precision. On top, the developed layout

descriptor achieves similar retrieval performance to the current state-of-the-art image-based approach while having much lower storage space consumption.

As the state-of-the-art in neural scene understanding is moving fast forward, we would like to get rid of dependence on depth sensors and employ the monocular depth estimation or monocular depth extraction stage in the pipeline. Another part of the pipeline where we would like to employ higher-level image understanding is the filtering of movable objects, which cannot be used for localization.

References

[1] Arandjelović, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR (2016)

[2] Armagan, A., Hirzer, M., Lepetit, V.: Semantic segmentation for 3D localization in urban environments. 2017 Joint Urban Remote Sensing Event (JURSE) pp. 1–4 (2017)

[3] Armagan, A., Hirzer, M., Roth, P.M., Lepetit, V.: Learning to Align Semantic Segmentation and 2.5D Maps for Geolocalization. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4590–4597 (2017)

[4] Baráth, D., Matas, J.: MAGSAC: Marginalizing Sample Consensus. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 10189–10197 (2019)

[5] Blum, H., Stiefel, J., Cadena, C., Siegwart, R.Y., Gawel, A.: Precise Robot Localization in Architectural 3D Plans. ArXiv **abs/2006.05137** (2021)

[6] Boniardi, F., Valada, A., Mohan, R., Caselitz, T., Burgard, W.: Robot Localization in Floor Plans Using a Room Layout Edge Extraction Network. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 5291–5297 (2019)

[7] Canny, J.F.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-8**, 679–698 (1986)

[8] Cham, T.J., Ciptadi, A., Tan, W.C., Pham, M.T., Chia, L.: Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map. 2010 IEEE Computer Society

Conference on Computer Vision and Pattern Recognition pp. 366–373 (2010)

[9] Chu, H., Kim, D.K., Chen, T.: You are Here: Mimicking the Human Thinking Process in Reading Floor-Plans. 2015 IEEE International Conference on Computer Vision (ICCV) pp. 2210–2218 (2015)

[10] Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: KDD (1996)

[11] Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**, 381–395 (1981)

[12] Herbers, P.M., König, M.: Indoor Localization for Augmented Reality Devices Using BIM, Point Clouds, and Template Matching. Applied Sciences (2019)

[13] Howard-Jenkins, H., Ruiz-Sarmiento, J.R., Prisacariu, V.A.: LaLaLoc: Latent Layout Localisation in Dynamic, Unvisited Environments. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 10087–10096 (2021)

[14] Ito, S., Endres, F., Kuderer, M., Tipaldi, G.D., Stachniss, C., Burgard, W.: W-RGB-D: Floor-plan-based indoor global localization using a depth camera and WiFi. 2014 IEEE International Conference on Robotics and Automation (ICRA) pp. 417–422 (2014)

[15] Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR (2010), `http://lear.inrialpes.fr/pubs/2010/JDSP10`

[16] Liu, C., Schwing, A.G., Kundu, K., Urtasun, R., Fidler, S.: Rent3D: Floor-plan priors for monocular layout estimation. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 3413–3421 (2015)

[17] Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. IJCV (2004)

[18] Maffei, R., Pittol, D., Mantelli, M., e Silva, E.P., Kolberg, M.L.: Global Localization Over 2D Floor Plans with Free-Space Density Based on Depth Information. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 4609–4614 (2020)

[19] Mendez, O., Hadfield, S., Pugeault, N., Bowden, R.: SeDAR: Reading Floorplans Like a Human—Using Deep Learning to Enable Human-Inspired Localisation. International Journal of Computer Vision **128**, 1286–1310 (2019)

[20] Noonan, J., Rivlin, E., Rotstein, H.: FloorVLoc: A Modular Approach to Floorplan Monocular Localization. Robotics **9**, 69 (2020)

[21] Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-Shot Cross-Dataset Transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**, 1623–1637 (2022)

[22] Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: An efficient alternative to SIFT or SURF. In: ICCV (2011)

[23] Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From Coarse to Fine: Robust Hierarchical Localization at Large Scale. In: CVPR (2019)

[24] Schönberger, J.L., Frahm, J.M.: Structure-From-Motion Revisited. In: CVPR (2016)

[25] Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)

[26] Valentin, J., Dai, A., Niessner, M., Kohli, P., Torr, P., Izadi, S., Keskin, C.: Learning to Navigate the Energy Landscape. In: 3DV (2016)

[27] Wang, S., Fidler, S., Urtasun, R.: Lost Shopping! Monocular Localization in Large Indoor Spaces. 2015 IEEE International Conference on Computer Vision (ICCV) pp. 2695–2703 (2015)

[28] Winterhalter, W., Fleckenstein, F.V., Steder, B., Spinello, L., Burgard, W.: Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 3138–3143 (2015)