Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

PERCEPT-V: Vision-Based Indoor Localization and Navigation via 3D Spatial Map Generation

HAO DONG[1], SUSHMA SURESH BABU[2], MARCO F. DUARTE[2], (Senior Member, IEEE), AND AURA GANZ[2], (Fellow, IEEE)

[1]Google, San Francisco, CA 94105 USA (e-mail: haodong@google.com)
[2]Department of Electrical and Computer Engineering, University of Masachusetts, Amherst, MA 01003 USA (e-mail: {ssureshbabu,mduarte,ganz}@ecs.umass.edu)

Corresponding author: Hao Dong (e-mail: haodong@google.com).

This work was supported in part by the National Science Foundation under Grant ECCS-1645737.

ABSTRACT We present an indoor navigation system that relies on a 3D spatial map of the navigable area generated from a set of carefully controlled video captures of the environment. The video-to-spatial map translation is commonly performed using structure from motion methods, which typically rely on opportunistic collection of images and do not leverage the presence of temporal or spatial correlations known a priori. In contrast, a controlled capture of the images can provide rich diversity in spatial and view orientation coverage. However, it can also strain the computational and storage burdens of spatial map generation. We therefore introduce a structure-from-motion approach that is tailored to leverage the presence of spatial and temporal correlations in the input image dataset while mitigating the impact of large-scale image collection. We also propose a set of data structures that characterize the generated spatial map and enable a framework for an indoor navigation system that is designed for blind and visually impaired users and that can be deployed in a smartphone platform without requiring retrofit of the environment. We describe the relationship between the data structures and the indoor localization and navigation algorithms. Experimental results on the performance and complexity of spatial map generation for indoor navigation verify the improvements enabled by our proposed framework.

INDEX TERMS Image motion analysis, indoor environments, inertial navigation, spatial map generation, structure from motion, indoor navigation, localization.

I. INTRODUCTION

Rendering high-resolution spatial maps of large-scale areas has become feasible thanks to advances in image understanding and the availability of high-quality, large scale databases of images. Toolboxes that derive structure from motion (SFM) such as Bundler [**?**], [**?**], COLMAP [**?**], [**?**], [**?**], VisualSFM [**?**], among others, now see widespread use. While in many cases the images used for SFM are obtained opportunistically, there are cases where instead an operator can perform a controlled survey of the space to be mapped, optimizing the coverage in locations and points of view involved in the image capture process. For example, SFM has found wide applicability in the design of navigation algorithms for indoor spaces targeted to blind and visually impaired users and based on image registration [**?**], [**?**], [**?**], [**?**].

While such structured captures are desirable and show improvement in spatial map generation (SMG), SFM tools do not exploit the inherent correlations and redundancy between highly structured image datasets. In most SFM implementations, the images are not assumed to have any temporal structure, and there are no assumptions regarding the specific structure of the overlap present between different images. In contrast, in controlled captures there are inherent correlations between cameras with overlapping fields of view that can inform the SFM algorithm. Furthermore, controlled captures can provide image datasets of size much larger than those available from opportunistic captures, which can stress the resources available for the SFM process.

To address settings where structured captures are possible and helpful, we propose in this paper an adaptation of existing SFM methods to multi-camera settings (in our specific

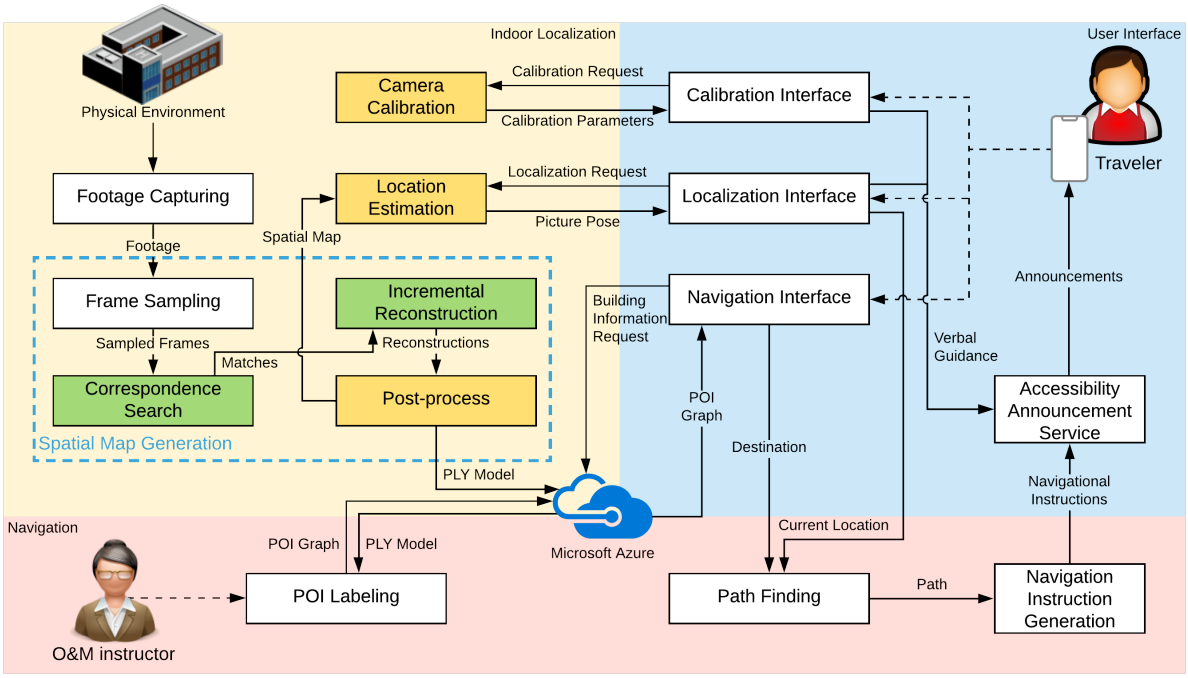


FIGURE 1: *Block diagram of PERCEPT-V indoor localization, navigation, and user interface modules.*

case, COLMAP) that exploits these known correlations and alleviate the computational and storage burdens of the SMG. We showcase these behaviors for our proposed method by evaluating them over a set of controlled image captures in indoor environments. Furthermore, we leverage the SMG output as the cornerstone of an indoor, real-time wayfinding assistant system that is tailored for blind and visually impaired (BVI) users.

More specifically, our real-time wayfinding assistant system provides guidance during the user's physical visit in the navigable area. This system is designed under the following considerations: (*i*) the system should be easily deployable to a new environment with minimum human involvement and infrastructure requirements; (*ii*) the system must be available to the user via a widely available interface, such as a smartphone; and (*iii*) the system must provide instructions to help users understand the path to the selected destination and their perimeter. Therefore, we exploit a vision-based indoor localization approach that requires no intervention to the physical environment.

Our system uses the 3D spatial map to register and localize the user based on their point of view, as captured by a photo taken from their smartphone. The 3D spatial map is also used to generate a novel multi-layer graph-based data structure that organize the building's information, including points of interest (POIs), paths, and regions. The graph is also used to calculate the navigation path from the user's location to the destination, and to generate navigational instructions from the path. Figure **??** shows an overview of the system's architecture. Our description of the system in this paper focuses on the indoor localization and navigation components, with a summarized description of the user facing interface for these components included for the sake of completeness. For a more detailed description, we refer the readers to [**?**], [**?**], [**?**], [**?**], [**?**].

The rest of this paper is organized as follows. Sections **??** and **??** summarize the literature in indoor localization and navigation, respectively. Section **??** describes our multi-camera SMG setup, focusing on the challenges of the multicamera setup and our solutions. Section **??** provides a quantitative assessment of the SMG. Section **??** overviews the image-based localization procedure, which is anchored on the SMG, while Section **??** presents a SMG-based data structure that provides the core component of an indoor navigation algorithm. Sections **??** summarize the user interfaces, and Section **??** provides a brief summary of the human subject trials completed for the system. Finally, Section **??** presents some conclusions.

II. INDOOR LOCALIZATION: STATE OF THE ART

With the pervasive deployment of radio frequency (RF) technology in infrastructures and user devices for communication purposes, RF signals are also considered for indoor localization purposes. There are various types and protocols used in recent research. Wireless local area networks (WLANs) are commonly used due to their widely deployed infrastructure [**?**], [**?**]. Received signal strength is collected from access points and used for localization with triangulateration [**?**], [**?**], [**?**] or fingerprinting [**?**], [**?**], [**?**]. Similar methods also rely on Bluetooth technology [**?**],

[?], [?]. The recently developed Bluetooth Low-Energy (BLE) within Bluetooth 5 provide more flexible, portable, and long-lasting alternatives to WLAN [?], [?]. Given that RF Identification Device (RFID) is used for tracking and identifying objects, it is also exploited for indoor localization of users. Localization systems using passive RFID [?], [?], [?] have lower deployment cost but lag in spatial coverage and localization resolution, compared to alternatives that use active RFID [?], [?]. A variation of passive RFID is indoor localization using near field communication [?], [?]. Furthermore, the location estimation can be performed on the mobile device instead of a server. Ultra-wideband has also received attention in indoor localization due to its greater penetration of obstacles such as walls [?], [?]. Another category of indoor localization uses magnetic signals. The first localization system using magnetic signals was proposed in 1979 [?], requiring the deployment of magnetic transmitters within the building and the mounting of magnetic receivers on the user. Several improvements [?], [?], [?] to this design were proposed. More recent magnetic-based localization systems are based on fingerprinting of the distinctive nature of magnetic field variations in indoor environments [?] and disturbances of the Earth's magnetic field on structural steel elements in a building [?], [?]. These approaches have also been extended to applications on mobile devices.

Two types of acoustic wave are also explored for indoor localization. Audible sound waves are used in multiple indoor positioning systems [?], [?], [?] that usually consist of mobile transmitters (speakers), stationary receivers (microphones), a central server, and wireless connections between them. It has the advantages of low cost on components and high accuracy but also suffers from the possible interference between users and with ambient noise. Other similar systems use ultrasound waves instead [?], [?], [?]. These solutions show good accuracy and low-cost at room level. However, the high path loss limits the spatial coverage and further makes it not scalable for large spaces.

Similar to acoustic solutions, researchers also investigated the potential use of optical signals. Infrared (IR) signals were used in early solutions [?]. Although IR sensors are inexpensive and have a good battery life, to deploy such systems for localization purposes is still expensive. Positioning systems based on visible light communication emerged in recent years. Light-emitting diode (LED) arrays are deployed on the ceiling to transmit modulated signals to the receiver. Then the locations are estimated from the combination of them The LED array is very energy efficient, but due to the nature of design, it becomes expensive for the dense distribution. A detailed survey of positioning systems using this approach can be found in [?].

Another main category is indoor localization using vision-based approaches. In contrast to the optical approaches, vision-based approaches observe artificial markers or natural features that exist in the environment, precluding the need to deploy active transmitters. These approaches are beneficial due to their low cost for deployment and maintenance, while incurring larger real-time computation cost. Artificial markers used in indoor localization include Vuforia [?], QR codes [?], ArUco [?], and Color Targets combined with barcodes [?]. Natural features are extracted by various computer vision algorithms and used in more sophisticated algorithms, e.g., Simultaneous Localization and Mapping, Vision Odometry, or Structure from Motion (SFM). The user's location can be estimated by having them capture a photograph from their point of view, and then searching in the 3-D spatial map generated from previously captured images. A comprehensive survey of this type of localization solutions can be found in [?].

III. INDOOR NAVIGATION: STATE OF THE ART

Using the Global Navigation Satellite System (GNSS) for outdoor localization, navigation applications such as Google Maps facilitate wayfinding tasks for both vehicle and pedestrian use. However, due to attenuation and scattering of GNSS signals in indoor environments, outdoor navigation applications cannot be used in indoor environments. Although there are several promising solutions for indoor navigation that have been successfully tested with human subjects [?], [?], [?], [?], [?], [?], [?], [?], none of them have been widely adopted in indoor venues. Existing approaches can be grouped according to the following criteria:

- Localization technology: The type of technique and technology used to localize the user in the physical environment. For example, dead reckoning uses inertial sensors to measure the rate of motion of the localization target, updating the location estimate accordingly.
- Sensor deployment and maintenance efforts: Sensor deployment effort correlates with the localization technology. For example, BLE-based localization requires the installation of BLE sensors throughout the physical space. In addition to the deployment effort, the sensor infrastructure requires maintenance such as tags' replacement due to vandalism, malfunction and/or battery depletion. Therefore, sensor-based infrastructure leads to an increased cost to the venue owners. On the other hand, vision-based and inertial localization does not require any deployment of sensors in the environment lowering the deployment cost. Both sensor and vision-based approaches require "digital maintenance" (change in the digital representation of the physical environment) in case the physical environment undergoes renovations (similarly, outdoor GIS systems need to be updated when roads are added/removed, or traffic regulations are changed).
- User device: Most systems leverage the sensors, display, computation, and communication modules embedded in the Smartphone. Some systems require specialized devices to facilitate localization and navigation tasks. Use of specialized devices increases the cost to the user and may hinder the adoption of this technology.

- Navigation instructions: The navigation instructions guide the user's journey to the chosen destination through the physical environment. We can first classify the instructions in three groups based on the time the instructions are generated and when they are delivered to the user:
  1) Off-line instructions: Generated during the system deployment phase. or just before the user's trip. These instructions are stationary and only available between POIs. During the navigation, they do not change as the user location changes in the venue. No real-time localization technology is required. It is assumed that the users know their location.
  2) User-solicited instructions: Generated during the preparation process or before the trip. The instructions are updated following the users' location and presented to the users when they prompt the system. Real-time localization is required.
  3) Continuous instructions: Dynamically generated based on the user's current location (similar to outdoor navigation apps as Google maps). Real-time localization is required.

  We can also classify the instructions based on the granularity of their contents:
  1) Shoreline: Instructions describe trailing movements, which is commonly used as an orientation and mobility skill for blind and visually impaired (BVI) travelers. An example is "follow the wall on your right and turn right at the next opening." [**?**].
  2) Turn-by-turn: Instructions include a set of turns such as turning direction, the distance to turn, and contextual information at the turn. For example: "please turn right into an intersecting hallway about 20 steps ahead." [**?**]
  3) Additional information: Some systems also provide additional information about the environment surrounding the user during the journey such as points of interest or description of specific areas.
- User trials: Whether the system was tested with human subjects and how many subjects participated. We emphasize that testing with blindfolded subjects cannot reflect the sentiment of BVI users. This is due to the fact that BVI users have undergone orientation and mobility training as well as possess very strong hearing sense not usually found in sighted people.
- Environment structure and complexity: Whether the system was designed and/or tested in specific environments. The navigation strategy in corridor-based environments and in open areas is quite different for BVI travelers.

IV. MULTICAMERA SMG FOR INDOOR NAVIGATION

The spatial map used in the localization and navigation processes is generated via incremental SFM, a standard computer vision technique for estimating three-dimensional structures from two-dimensional image sequences [**?**]. Several off-the-shelf implementations are available in the literature, including [**?**], [**?**], [**?**], [**?**], [**?**], [**?**]. We focus on a general-purpose pipeline for SFM and Multi-View Stereo called COLMAP [**?**], [**?**], [**?**]. COLMAP provides a well-structured code basis with both graphical and command-line interfaces, a wide range of features for the spatial map of ordered and unordered images, and code maintenance from the authors. Furthermore, unlike approaches like SLAM and Project Tango [**?**], [**?**], [**?**], COLMAP can be implemented using generic smartphone cameras, providing low overhead for deployment. However, we found several shortcomings to its use for public indoor environments that cause incorrect spatial maps to be generated. Fortunately, these shortcomings can be addressed by leveraging spatiotemporal information present when COLMAP is used on sequences of images from multiple video camera recordings. In this section, we describe the functionality of COLMAP and propose additional functionalities and requirements on its inputs that suffice to improve its performance for controlled multi-camera capture settings.

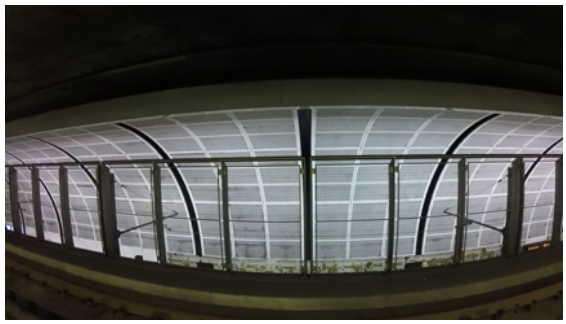


FIGURE 2: *Environment with extensive repetitive visual features.*

A. STRUCTURE FROM MOTION VIA COLMAP

COLMAP [**?**], [**?**], [**?**] provides a complete SFM pipeline based on three steps: feature extraction, correspondence search, and incremental reconstruction (or mapping). COLMAP is designed to handle arbitrary collections of images, making it widely applicable. However, this can cause issues in the correspondence search when there are separate regions of the area being observed that have visual similarity or repetitiveness. An example of visual repetitiveness is shown in Figure **??**. It is easy to see that many possible image sets can be obtained from different poses that have matching visual appearances, which can negatively affect the SMG.

B. SEQUENTIAL MATCHING IN CORRESPONDENCE SEARCH

The correspondence search process finds pairs of images that are capturing overlapping regions of the 3D scene and estimates their relative poses against one another. The correspondence is determined by extracting image features from each image and matching feature descriptors from different source images that are observing the same corresponding

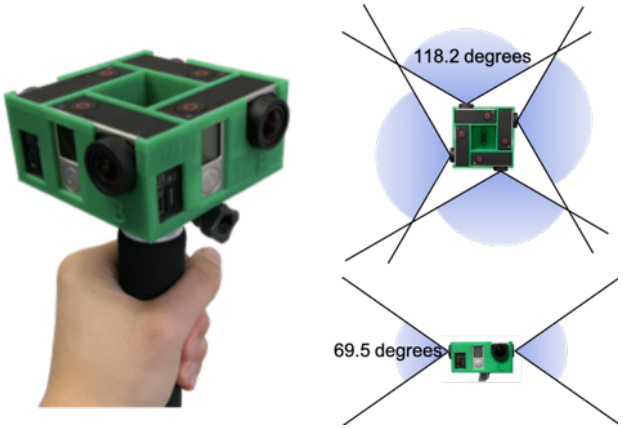


FIGURE 3: *Multiple Camera System used in Capturing Process.*

spatial location. Once feature matches are determined, the 2D locations for the matched features in each pair of images are used to obtain relative pose estimates, i.e., the translation and rotation applied to the camera between the captures of the images considered. When the input is given as a sequence of images, COLMAP takes neighboring images in a temporal sequence as correspondence candidates, based on the assumption that temporal neighbor images are more likely to capture views of the same scene [**?**].

C. MULTI-CAMERA SEQUENTIAL MATCHING

To generate an accurate spatial map, we need to capture a set of images that cover different angles of the environment, while maintaining enough overlap between the images. Since this process can be very time consuming, we design a data collection process that minimizes the human effort using a custom video camera array, shown in Fig. **??**.

We extend the original matching strategy of COLMAP (using its custom matching functionality) by introducing inter-camera matching, which further attempts to match each image with its neighboring images in other cameras. The new inter-camera sequential matching considers the following images as candidates:

- Neighboring images: The next $n-1$ images in the sequential order which are captured by the same camera, where n is the length of the matching window (e.g., sequential matching in COLMAP).
- Sibling images: The images that are captured by the other cameras in the array at the same time.
- Sibling neighboring images: The neighboring images of each sibling image, with the same matching window length n.

These addition allow for the matching of images obtained from different cameras, which can provide for significant improvements in the performance of SMG due to the acquisition of images with overlapping views from different perspectives - see Figure **??** for an illustration.

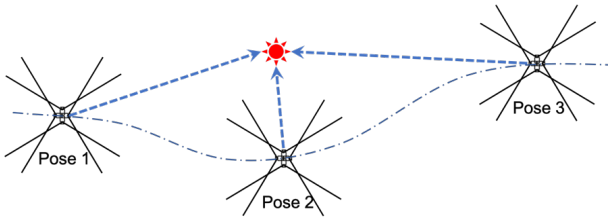


FIGURE 4: *An example of 3D scene features shared by inter-camera frames.*

D. INCREMENTAL SPATIAL MAP GENERATION

The incremental reconstruction process uses the correspondences obtained from the matching process as an input, starting from a given image pair, and proceeding by registering one additional matched image and triangulating the additional feature points sequentially. The output of this image registration will be a new spatial map, an estimate of the camera position and pose, and estimates of the camera's intrinsic parameters. Thus, the size of the spatial map (i.e., the number of 3D points included) grows incrementally during the generation process, and the incremental nature of this process can lead to accumulating errors. Therefore, after the number of registered images in the spatial map grows by a certain percentage, a global bundle adjustment (GBA) is performed to minimize the reprojection error and optimize the poses and map point positions. We find that this SMG process has several challenges that are discussed and addressed in the remainder of this section.

Computational cost: The GBA computation has a cubic time complexity [**?**]. In addition, a convergence strategy is applied to this process so that the bundle adjustment will be repeated until the cost improvement is less than a certain value or a maximum number of repetitions is reached. These convergence criteria often renders GBA very slow when a large number of images is involved (e.g., in the thousands).

To alleviate the computational burden of GBA, we propose a segmented generation strategy. First, the entire sampled image sequence is portioned into small segments while maintaining a certain amount of overlap between adjacent segments. The spatial maps for these segments are computed separately; then, the spatial maps of adjacent segments are merged incrementally using COLMAP's "model merger" function until one spatial map remains. The size of each segment and the size of the overlap between segments can be configured as algorithm parameters that provide the following tradeoff: the more images that are overlapped in the segments, the more likely merging will be successful, while incurring a correspondingly higher cost in the generation of each segment of the spatial map. Figure **??** shows the described process with an example containing four segments.

Incorrect registration: For some challenging visual environments (e.g., a subway station with insufficient lighting), some images are registered at incorrect positions. Usually, these images can be detected and filtered out of the map generation process by evaluating the estimated camera

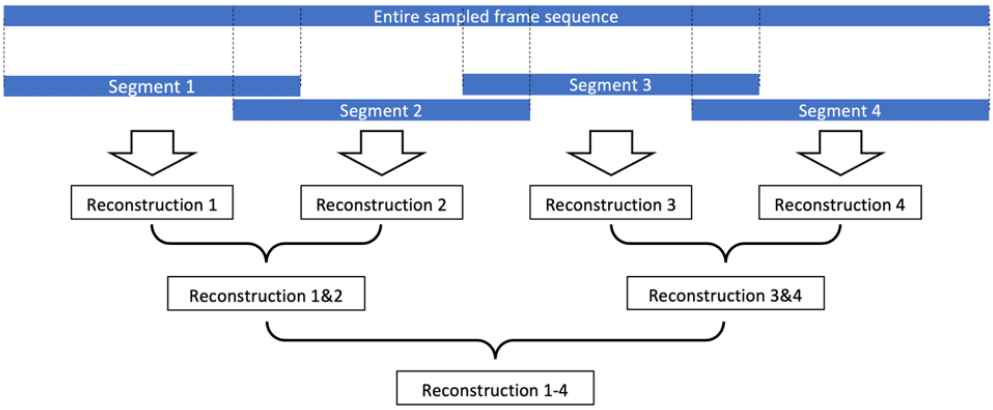


FIGURE 5: *Segmented spatial map generation process.*

intrinsic parameters, which are often incorrect or unfeasible. Common incorrectly estimated parameters include (i) an inaccurate focal length; (ii) a principal point located outside of the image borders; and (iii) invalid distortion parameters; COLMAP tests for the presence of any of these conditions in the camera parameter estimates to detect incorrectly registered images that will be excluded. Since the device variation between the cameras used in practical settings is very limited, we have found that using more specific conditions can improve the filtering of incorrectly registered images. More specifically, we empirically observed that incorrectly registered images have principal points estimates located far from the image center, yet within the image borders.

Therefore, we replace the second condition to (ii) a principal point located farther than a maximum distance d from the image center. This range distance d is configurable as a percentage of the image size. In our empirical results (cf. Section **??**), the revised criteria shows a significant improvement in rejecting the badly registered images.

Spatial map merging and post-processing: Unlike standard SFM applications, where a database of images is collected for processing, in localization practice the input images will be collected via multiple sequential captures and processed as batches. Thus, we need to merge spatial maps obtained from each of the sequences. The spatial map files based on different recordings are held in their own separate databases. We use COLMAP's "database merge" functionality to merge spatial maps from different sequential captures in an incremental manner, i.e., merging one new spatial map at a time.

Once we merge the spatial maps of all recordings in the target environment, we need to align the model coordinates to the real world coordinate system. COLMAP's model orientation alignment step will rotate the virtual representation of the environment in the spatial map so that the gravity vector always points downward (i.e., it points in the positive Y axis in its coordinate system). This step uses vanishing point estimation under the Manhattan world assumption [**?**], which holds for most indoor environments.

V. MULTICAMERA SMG PERFORMANCE

We show the performance of the multicamera SMG approach in terms of accuracy, processing time, and computational resource usage, and compare it against that of the standard COLMAP method. We recorded video sequences for motion paths within two buildings on the UMass Amherst campus (Whitmore Building and Campus Center) and a transportation hub - North Station in Boston. The recording paths in all three environments are shown in Fig **??**. The structural characteristics and the lengths of recording paths are presented in Table **??**. The spatial maps of all three environments are generated on a desktop computer with an Intel Core i7-4770K 8-core 3.5 GHz processor, 16GB RAM, and a GeForce GTX760 GPU running 64-bit Ubuntu 18.04.2 and CUDA 10.1.

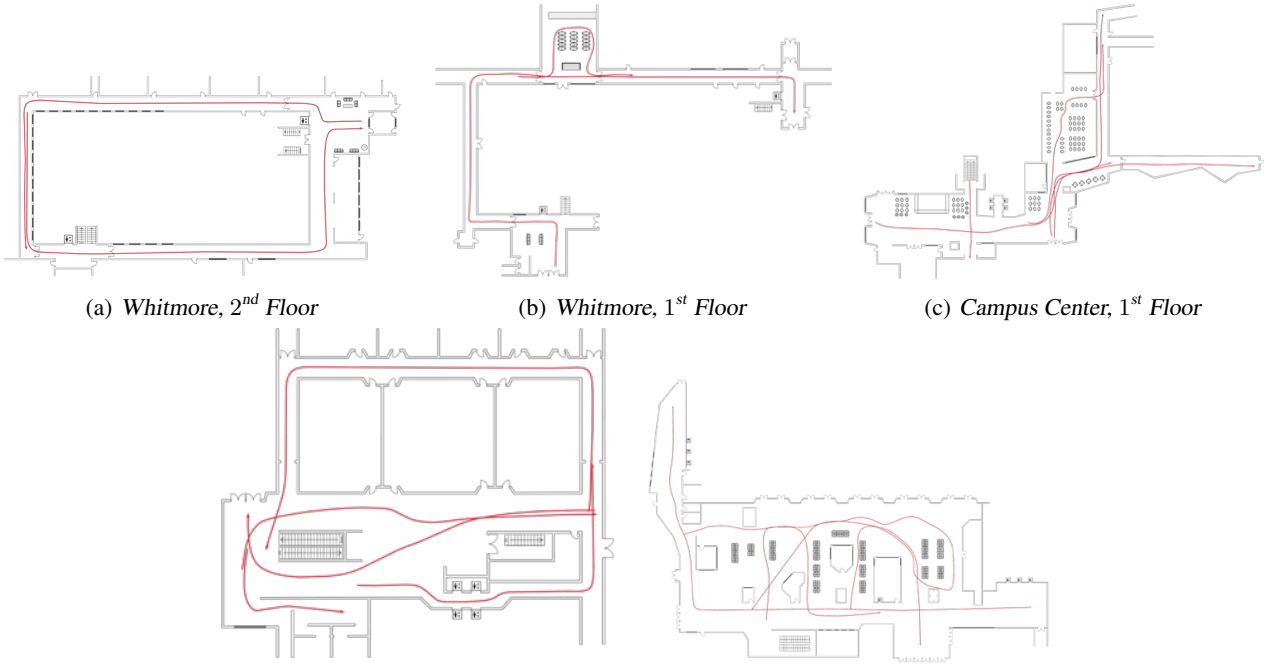
A. DATA COLLECTION

The system consists of 4 GoPro Hero3+ Silver cameras that capture a FOV of 360 degrees horizontally and about 70 degrees vertically and a custom 3-D printed mount that allows for full spatial coverage around the location of the operator. All recordings are captured at a resolution of 1920-by-1080 and frame rate of 60 frame per second (fps). All of our numerical experiments used the same configuration for the SMG. The correspondence search matching window size is set to $n = 10$ frames; for the image sequence segmentation, we denote the segment size (number of frames) by S and the overlap between segments (number of frames) by O.

A set of videos are first recorded from the target environment using the multiple cameras in the array. The recording operator only needs to hold this camera system above their head and traverse the target environment (e.g., all high traffic areas in the indoor environment) during the recording.

B. FRAME SAMPLING

Since COLMAP does not directly support video recordings as input, we first extract frames from the recordings into an image sequence. The original footage is recorded at a 60 fps rate, which results in a very large number of frames to process with high redundancy in the corresponding



(a) *Whitmore, 2nd Floor* (b) *Whitmore, 1st Floor* (c) *Campus Center, 1st Floor*

(d) *Campus Center, Basement* (e) *North Station*

FIGURE 6: *Recording paths in each test environment for SMG performance evaluation (in red), overlaid over building blueprints.*

Environment	*Structure*	*Covered Area (m^2)*	*Recording Path (m)*
Whitmore	*Two floors; corridor-based with opening areas (<100m^2)*	*1st floor: 386; 2nd floor: 500*	*1st floor: 148; 2nd floor: 173*
Campus Center	*Two floors; large opening areas (<1000 m^2) with long extended corridors*	*Basement: 938; 1st floor: 1247*	*Basement: 250; 1st floor: 282*
North Station	*One floor; a very large opening area (2500 m^2) with a short-extended corridor*	*2700*	*230*

TABLE 1: *Test environments of SMG performance.*

neighboring images. Thus, only a subsample of the video recording frames are used as the input images for COLMAP. Given the performance of SMG during experimental tests in different types of environments, we determined that a subsampled frame rate of 3 fps results in a sufficient number of informative images for SMG in most indoor environments. One exception to this rule of thumb is the set of environments with extensive repetitive visual features, such as fences and concrete hallways (e.g., Fig. **??**); in such cases, a higher sampling rate (e.g., 5 fps) is sufficient to avoid incorrect "foldings" in the SMG.

C. SMG ACCURACY AND COMPUTATIONAL COMPLEXITY

Figure **??** provides the generated spatial maps for original COLMAP vs. the proposed multicamera SMG. It is evident that in four of the five environments tested the amount of detail available in the spatial maps is significantly higher for the proposed SMG. More significantly, the maps from the proposed SMG have a larger range of spatial locations that are accurately mapped out in comparison to the original COLMAP result; this can be evaluated by comparing these point clouds to the blueprints from Figure **??**.

Tables **??**, **??** and **??** show the processing time, the peak RAM usage and the size of each file contained in the spatial map for all the testing environments, respectively. Figure **??** shows the processing time of the SMG against the coverage area in the physical space. One can conclude that the preparation time of the spatial map takes less than one minute for every 3 square meters in area.

We also test the performance of our proposed multicamera SMG method and the original COLMAP formulation. We select several runs from those illustrated in Figure **??** to cover a variety of sequence lengths. We examine the computation time and final database size for the SMG process as performance metrics. Figure **??** confirms that the computation time for SMG is reduced when processing longer sequences, while for shorter sequences the overhead of segment merging cancels or mitigates the savings from segmentation. Furthermore, Figure **??** verifies that there is no penalty to segmentation on the size of the resulting database. Finally, Table **??** provides a summary of properties of the spatial maps generated in our experiments. The table shows several improvements for environments involving long

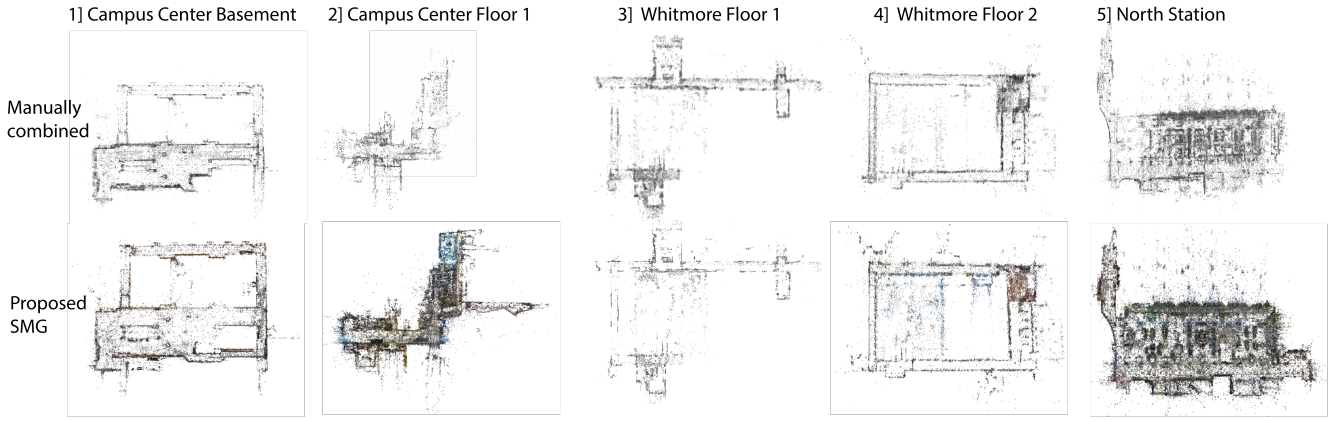


FIGURE 7: *Spatial maps generated by COLMAP and the proposed multicamera SMG. The top row corresponds to original COLMAP, where we manually combined the multiple independent and separated partial spatial maps generated by COLMAP; the bottom row corresponds to the proposed multicamera SMG.*

Process	Whitmore Floor 1 (min)	Whitmore Floor 2 (min)	Campus Center Basement (min)	Campus Center Floor 1 (min)	North Station (min)
Frame sampling	3.86	4.68	9.21	8.3	10.48
Feature extraction	1.003	1.186	2.159	2.29	3.10
Frame matching	6.607	10.567	10.565	21.865	52.68
Spatial map gen.	26.248	39.402	78.645	82.941	215.47
Merging	26.965	36.91	192.52	270.33	531.78
Aligning	9.43	12.62	23.33	23.5	32.55
Indexing	1.24	1.635	3.39	3.47	5.55
Total	74.653	107	319.819	412.696	854.13

TABLE 2: *Processing time of proposed multicamera SMG.*

Process	Whitmore Floor 1 (Mb)	Whitmore Floor 2 (Mb)	Campus Center Basement (Mb)	Campus Center Floor 1 (Mb)	North Station (Mb)
Frame sampling	1861	1894	3277	2922	2899
Feature extraction	144	143	144	143	140
Frame matching	203	151	145	165.4	217
Spatial map gen.	844.3	987.1	1003	1347	2491
Merging	1181	1390	1789	2126	4368
Aligning	132.8	174	269.4	337	591
Indexing	369.7	400.3	472.5	543.8	786
Total	4735.8	5139.4	7099.9	7584.2	11492

TABLE 3: *Peak RAM usage of proposed multicamera SMG.*

Process	Whitmore Floor 1 (Mb)	Whitmore Floor 2 (Mb)	Campus Center Basement (Mb)	Campus Center Floor 1 (Mb)	North Station (Mb)
Database	355.6	511.7	1009	1298	2550
Spatial map gen.	54.4	75.8	112.4	168.5	324
Indexing File	180.8	202.9	250.5	301.4	467
Total	590.8	790.4	1371.9	1797.9	3341

TABLE 4: *Size of generated files in proposed multicamera SMG.*

corridors (Campus Center, Whitmore): (i) increased numbers of images matched, (ii) reduced number of redundant points, and (iii) higher number of frames preserved in the SMG as measured by the map size. The table also shows that the proposed SMG retains good performance for environments featuring large open spaces (North Station).

VI. REAL-TIME LOCALIZATION

In this section, we describe the procedure for image-based user localization, which follows the diagram shown in Figure **??**. Given that errors and minor implementation defects are ubiquitous in modern manufacturing processes for digital cameras, there are no two camera sensors that have the exact same projection features. Using the same camera intrinsic parameters universally on the same type of cameras will exacerbate the impact of the differences between the cameras on the algorithms' performance and, more specifically, lead to inaccurate estimations of the camera poses. To calibrate the camera on the user device, we rely on the camera calibration functionality of COLMAP, which is intrinsic in its image registration process. The user is asked to take a number of pictures of a stationary scene

Environment	# models generated by COLMAP	# matched images/ # images		Total # points		Spatial map size (Localization), Mb		Spatial map size (Reconstruction), Mb		Area in m^2
		Proposed	COLMAP	Proposed	COLMAP	Proposed	COLMAP	Proposed	COLMAP	
Campus Center Basement	18	3019 / 3208	2835 / 3208	191666	244293	23.4	24.6	98	95.4	938
Campus Center Floor 1	24	2977 / 3068	2863 / 3068	262068	283336	31.3	28.4	136.8	135.0	1247
Whitmore Floor 1	8	1276 / 1436	991 / 1436	80904	66008	8.1	6.0	46.4	36.7	386
Whitmore Floor 2	16	1711 / 1808	1309 / 1808	128212	10772	12.9	9.7	62.8	50.8	500
North Station	11	4020 / 4100	3999 / 4100	531195	742076	63.5	72.3	261.1	532.1	2700

TABLE 5: *Numerical properties of 3-D point models generated by SMG.*

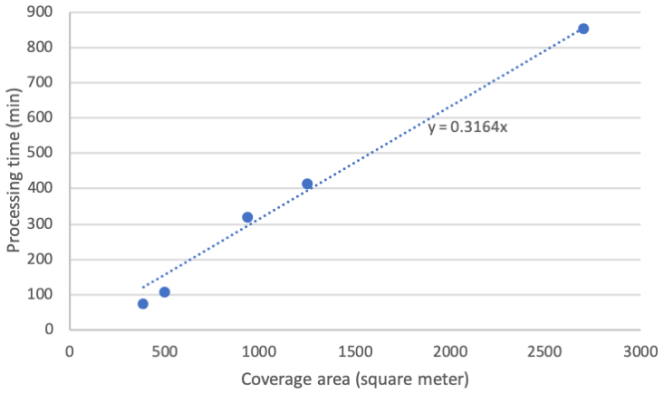


FIGURE 8: *Trend of proposed multicamera SMG time to the coverage area.*

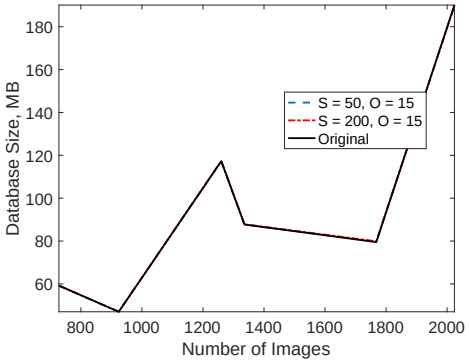


FIGURE 10: *Database size for proposed multicamera SMG vs. original COLMAP. Note that the three plots essentially overlap one another.*

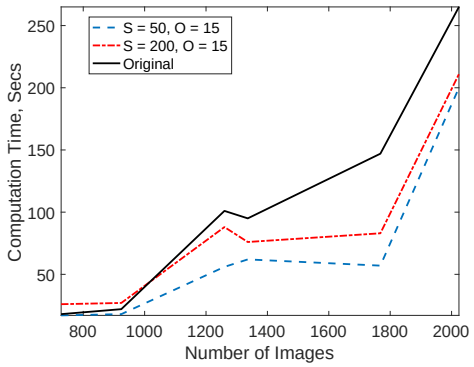


FIGURE 9: *Computation time for proposed multicamera SMG vs. original COLMAP.*

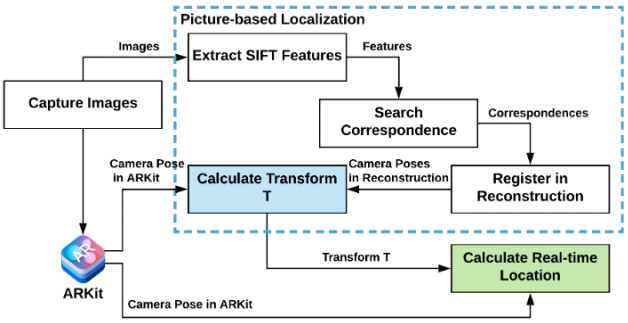


FIGURE 11: *Block diagram for image-based localization process.*

the first time the navigation app is used. If COLMAP can register this small set of images, it will return the set of camera parameters that can be used for all future captures. If COLMAP fails to register them, then we ask the user repeat the image collection in a different region of the navigation space, until COLMAP succeeds.

To begin the process, we ask the user to capture a few pictures using a smartphone client at different orientations. From these pictures, the localization system extracts SIFT features and matches them with the frames registered in the spatial map of the target area using the COLMAP vocabulary tree. A successful matching provides the navigation system with an initial localization for the point of view from the user.

Figure **??** provides a histogram of localization times (e.g., the latency of the localization process) using the proposed multicamera SMG system for a set of 100 trials corresponding to 10 randomly chosen images from the Campus Center (Basement and First Floor) dataset and from the North station dataset, with each image being used in 10 trials to assess the variability of the localization time. The graph also includes a smooth curve by using a kernel density estimate which is obtained by normalizing the statistics

computed within each bin. The average localization time is 0.011 seconds for the Campus Center and 0.103 seconds for the North Station, which can be explained by the higher complexity of the latter environment. For comparison, the average localization time for a training image using standard COLMAP for these two environments was found to be 0.063 seconds and 0.020 seconds, respectively.

After the initial localization of the user, we rely on inertial odometry (also known as dead reckoning) that is usually included in smartphone platform APIs. For example, Apple's iOS includes these functionalities in ARKit, an API allowing third-party developers to build augmented reality applications [**?**]. One of ARKit's essential functions is to track the device pose in real-time using inertial odometry based on the smartphone's inertial sensors. We leveraged this functionality to get the user's relative movement from the start of the navigation (e.g., the location where the user takes the pictures). For each successfully registered picture in the previous stage, we save its estimated camera pose in the inertial odometry coordinate system and in the spatial map coordinate system. Next, by comparing the camera pose in these two coordinate systems (cf. Fig. **??**), we can calculate the transformation T from the inertial odometry coordinate system to the spatial map coordinate system as

$$T = \mathrm{Avg}(\mathrm{Pose}_{\mathrm{Rec}} \cdot \mathrm{Pose}_{\mathrm{AR}}^{-1}). \tag{1}$$

The transformation is then used to get the user's real-time location in the spatial map coordinate system based on the pose tracked in inertial odometry as

$$\mathrm{Pose}_{\mathrm{Rec}} = T \cdot \mathrm{Pose}_{\mathrm{AR}}. \tag{2}$$

However, the accuracy of the location calculated in (**??**) has been observed to degrade as the length of the user's inertial odometry track increases, because the inertial odometry tracking is not globally optimized for large scale environments. When the localization error exceeds a certain (environment-dependent) threshold, the validity of the navigation instructions will be affected. Therefore, to preserve accurate localization, the image-aided localization should be repeated at regular intervals during normal use (e.g., at predetermined intervals of time or spatial displacement; after a given number of trips is completed; or when the user desires to confirm their current location).

The localization in multiple story environments is slightly different in some cases. For multi-story environments, a separate spatial map needs to be prepared for each floor. When the user performs the initial visual-aided localization, the system will perform a search for the best match among all available spatial maps (i.e., all floors). The floor in which the most pictures are successfully registered and their estimated positions are the most coherent will be selected. Subsequently, any further localization searches will be only performed in this floor while the elevation change (monitored using the smartphone's pressure sensors, for example) is less than 2 meters. When such an elevation change is detected, another all-floor-localization will be performed.

VII. NAVIGATION ALGORITHM AND DATA STRUCTURES

In this section, we describe a new multi-layer graph-anchored data structure that represents the information captured by the 3D spatial map that is relevant for user navigation. The design purpose of the proposed graph and data structure is to organize all the information leveraged in the navigation tasks uniformly. Foltz [**?**] suggests a set of principles that develop a basic vocabulary of spatial features that assist wayfinding and imageability — the traveler's ability to form a mental map of the space, including identifiable places, landmarks, paths, and regions. Here, we leverage similar categories of features and organize them in a three-layer data structure described over the next three subsections, respectively.

A. BACKBONE LAYER

The backbone layer contains the basic structure and information of the places and paths in the environment. This layer has the following components:

- *Landmarks* denote locations considered as a destination in a wayfinding task. Each landmark is represented by a node in the backbone layer graph and contains a name, a verbal description, and a coordinate in the spatial map. For places which are not distinct in their functionality, e.g., restrooms, exits, or floor transitions, a category field is assigned. For example, some environments will contain multiple accessible restrooms; if the user selects "closest accessible restroom" as the destination, all of these restrooms will be considered as candidates. Subsequently, only the closest one will be selected for navigation.
- *Waypoints* are decision points between landmarks, and are also represented by a node in the backbone layer graph. Each waypoint can connect in the graph to both landmarks and other waypoints. For example, a waypoint may correspond to the intersection of two corridors, or just outside the corner of a structure that is blocking a direct path to a landmark.
- *Edges* will be placed in the graph between a waypoint node and all nodes that match one of the following two properties: (i) the node corresponds to a landmark that is directly within line-of-sight of the waypoint, or (ii) the node corresponds to a second waypoint that is within line-of-sight of the first waypoint. More generally, a link between nodes in the graph represents that these two nodes correspond to two waypoints or a waypoint and a landmark that are in line-of-sight to each other.
- *Paths* between physical locations are described as a list of connected nodes in the graph that starts and end at the corresponding landmark nodes.

While a waypoint can be linked to another waypoint or landmark via an edge, a landmark can only link to a waypoint, since landmarks are always the destinations of the paths. A path defined in the graph is a list of the waypoints leading the user to the destination. To follow the path, the user can always walk straight to the next waypoint until the destination is reached.

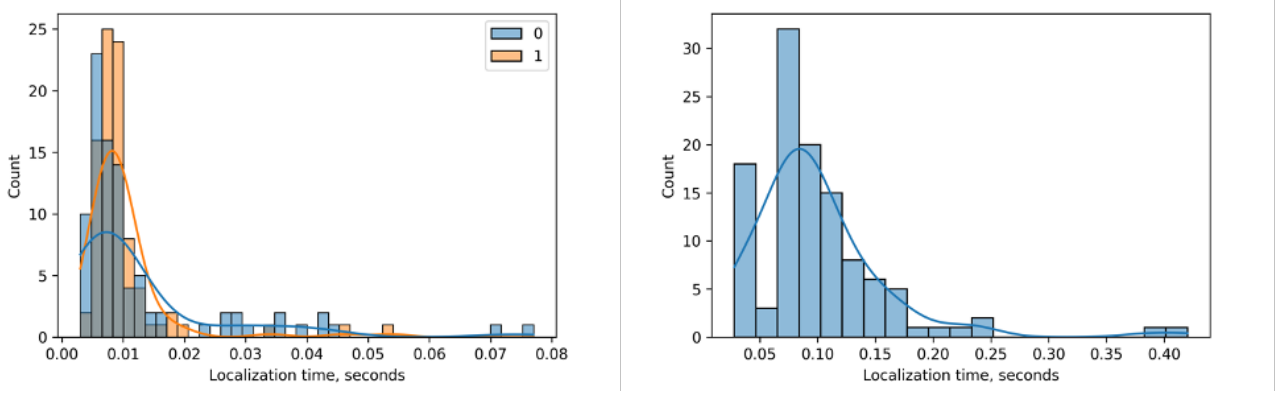


FIGURE 12: *Histogram of localization times, in seconds, from 100 trials in each environment. Left: Campus Center (legend denotes floors); Right: North Station. The curves show corresponding kernel density estimates.*

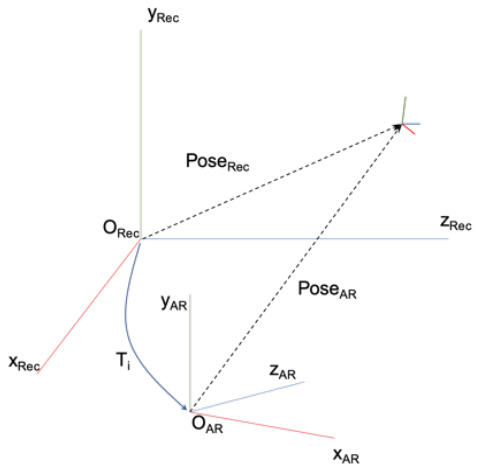


FIGURE 13: *Transformation T_i from the spatial map coordinate system to the inertial odometry coordinate system for captured picture i.*

B. REGION LAYERS

Parallel to the backbone data and graph layer, there are two region data layers, which contain information about different regions of the navigable area. A region consists of a set of informational labels and a set of nodes from the backbone layer graph whose locations are within the region.

- *Map regions* describe convex-shaped walkable areas. Each map region's informational labels include the region's name and the topological relation to the landmarks and waypoints included in it. Map regions will be used in the pathfinding algorithm (c.f. Section **??**).
- *Announced regions* contain the information used in the navigation instructions. Each announced region's information includes a name of the area and the type of area, which can be a general purpose open area or a corridor, e.g., usually a narrow passageway that provides access to and from open areas. For corridor areas, the central axis of the corridor will be automatically defined, which is used to determine which side of the corridor each landmark contained in the region is located on. The algorithm to automatically determine the corridor's central axis is introduced in Section **??**.

For each landmark and waypoint node, the node's data will include the labels of the map region and announced region that include that landmark or waypoint.

C. TILE LAYER

Note that the region layer presents region descriptions that are qualitative rather than quantitative, i.e., they do not define the geometry of the region such as its shape or contours. Unlike landmarks or waypoints, the spatial definition of a region is much more complex than a single coordinate. For most buildings, the shape of each region can be reduced to a simple shape, e.g., a rectangle. However, with the consideration of obstacles in the environment such as furniture or decorations, a simple shape may not be elaborate enough to describe a region. In such cases, the use of poly-line contours or polygons will raise the so-called point-in-polygon (PIP) problem.

Instead, we introduce a tile-based description of regions as the basis of another layer in the data structure holding this geometry information. The tile layer is constructed as follows. First, the whole navigable area is divided into tiles of small size (e.g., 1-4 square feet). This provides approximations for any shape in the region as a finite set of tiles. The tile size can be customized based on the complexity of the structure and the precision requirement.

Tiles are also used to integrate a user's position into the graph via a user node assignment. Since the user's location is not contained in the backbone graph, the shortest path to the destination cannot be directly calculated from that graph using a graph-based pathfinding algorithm, such as Dijkstra's algorithm [**?**]. Instead, we first collect a list of waypoints and landmarks in the map regions contained in the same tile as the user's location, and link the user's node to the nodes for the aforementioned waypoints and landmarks. The

details will be provided in Section **??**.

Tiles are also used to characterize corridor regions in terms of their central axis. We use Algorithm **??** to determine the corridor's central axis using a tile-based area definition of the corridor region. In words, our algorithm identifies the coordinates of the four outermost/corner tiles in a corridor area to sketch a perimeter of the region, and then identifies the two mutually farthest corners in this perimeter as the ends of the central axis of the corridor.

Algorithm 1 *Computing central axis of a corridor area.*

Input: Set of tiles in the region T
Output: : 2D-coordinates of two ends of the central axis $C = [c_1, c_2]$

Initialize sets of tiles $T_\mathrm{l}, T_\mathrm{r}, T_\mathrm{t}, T_\mathrm{b} \leftarrow [\cdot]$
for each row of tiles in T **do**
  Add left-most tile in T_l
  Add right-most tile in T_r
end for
for each column of tiles in T **do**
  Add top-most tile in T_t
  Add bottom-most tile in T_b
end for
$t_\mathrm{lt} \leftarrow$ the top-most tile in T_l
$t_\mathrm{lb} \leftarrow$ the bottom-most tile in T_l
$t_\mathrm{bl} \leftarrow$ the left-most tile in T_b
$t_\mathrm{br} \leftarrow$ the right-most tile in T_b
$t_\mathrm{rb} \leftarrow$ the bottom-most tile in T_r
$t_\mathrm{rt} \leftarrow$ the top-most tile in T_r
$t_\mathrm{tr} \leftarrow$ the right-most tile in T_t
$t_\mathrm{tl} \leftarrow$ the left-most tile in T_t
Initialize $M \leftarrow [\cdot]$
$T_\mathrm{corner} \leftarrow$ a list $[t_\mathrm{lt}, t_\mathrm{lb}, t_\mathrm{bl}, t_\mathrm{br}, t_\mathrm{rb}, t_\mathrm{rt}, t_\mathrm{tr}, t_\mathrm{tl}, t_\mathrm{lt}]$
for each adjacent pair $[t_1, t_2]$ in T_corner **do**
  if $t_1 \neq t_2$ **then**
    Add midpoint of t_1 and t_2 to M
  end if
end for
Remove redundant points in M.
$d_\mathrm{max} \leftarrow 0$
for each pair of different points $p_1, p_2 \in M$ **do**
  if $\mathrm{dist}(p_1, p_2) > d_\mathrm{max}$ **then**
    $d_\mathrm{max} \leftarrow \mathrm{dist}(p_1, p_2)$
    $c_1 \leftarrow p_1$
    $c_2 \leftarrow p_2$
  end if
end for

D. LABELING TOOL

To encode the information for the navigation data structures, we developed a desktop-based labeling tool using the Unity engine. The engine can render the 3D model obtained from the SMG when provided in PLY format. The users can download, edit and upload the graphs described in this section through its user interface.

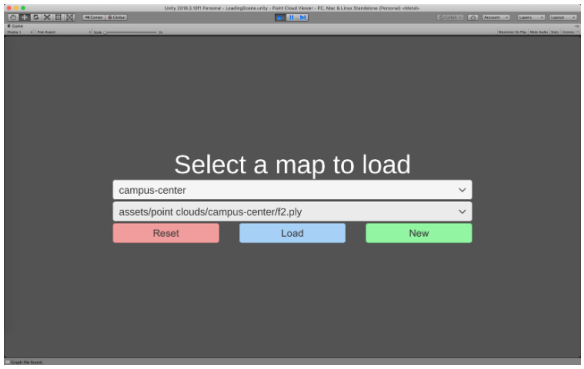


FIGURE 14: *Screenshot of labeling tool - select building and area.*

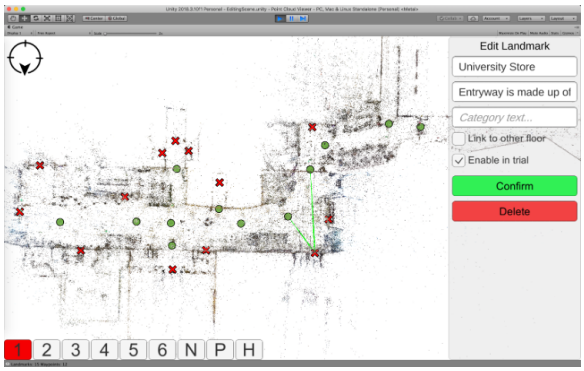


FIGURE 15: *Screenshot of labeling tool - edit landmarks dialog.*

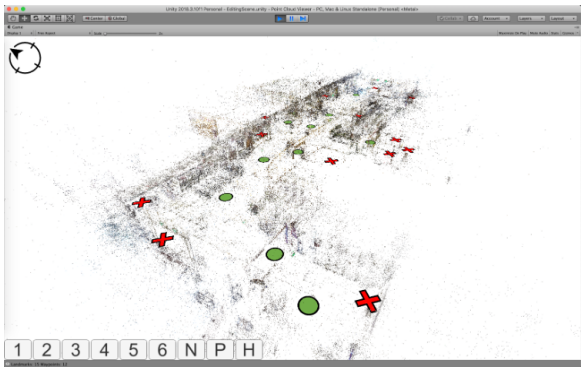


FIGURE 16: *Screenshot of labeling tool - 3D free view.*

The tool starts up by browsing a container on an Azure Blob storage for all available SMG reconstructions. Then the user can select the building and area to load (cf. Figure **??**). Once the PLY model is loaded, the top view of the reconstruction will be rendered on the main view. The user can move around using the `W/A/S/D` keys, rotate using the `Q/E` keys, and zoom using the `F/C` keys in top-down view (cf. Figure **??**). The tool can also be switched into a 3D free view (cf. Figure **??**), in which the arrow keys are used to rotate the view. A side panel shown on the right edge of the screen provides options to edit the different components on the map.

A landmark add/edit panel is available by clicking or

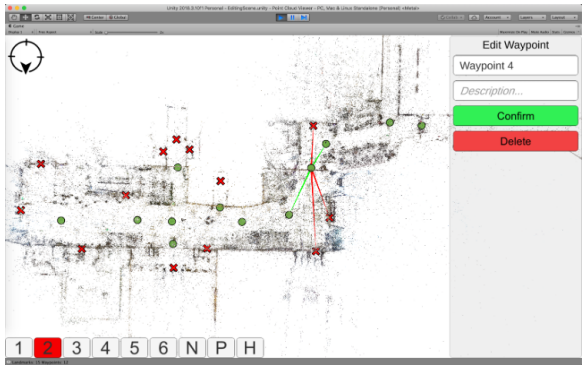


FIGURE 17: *Screenshot of labeling tool - edit waypoint dialog.*

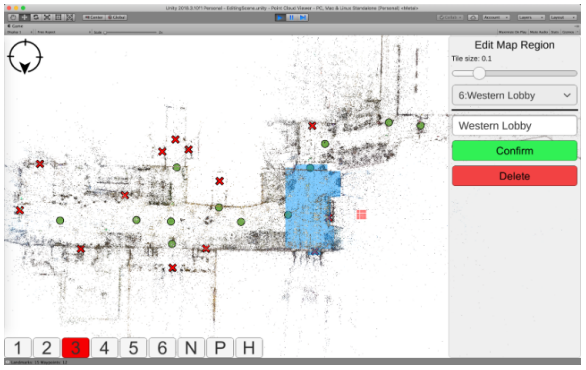


FIGURE 18: *Screenshot of labeling tool - edit map region dialog.*

pressing the `1` button or key, respectively (cf. Figure **??**), or by clicking on the landmark location. The user can edit its name and description by entering in the fields and indicate its position by directly click on the map. Landmarks can be linked to waypoints by clicking on them. Similarly, a waypoint add/edit panel is available by clicking or pressing the `2` button/key (cf. Figure **??**), or by clicking on the waypoint location.

To create a map region, the user can open a map region edit panel using the `3` button/key. The cursor will then become a small tile-based brush. The user will need to determine the size of the tiles to begin. Then, by selecting the "add new area" option from a dropdown list, the user can define a new map region by giving it a name and highlighting the tiles belonging to it (cf. Figure **??**). Announced regions can be created in a similar way using the announced region edit panel (cf. Figure **??**).

Another important step is to measure and define the scale of the reconstruction to the physical environment. Once the scale panel is opened (cf. Figure **??**), the user selects two points on the map by clicking them in order. The user then will be asked to enter the actual distance between the points in the physical environment, which is then used to calculate the required scaling ratio, which will then be attached to the graph as an auxiliary parameter. After editing is completed, the user can press the `ESC` key to bring up the main menu (cf. Figure **??**) and click the "Save Graph" button to save the graph as a JSON file on the Azure Blob storage.

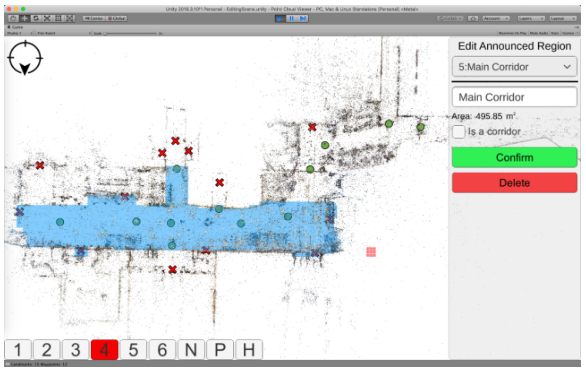


FIGURE 19: *Screenshot of labeling tool - edit announced region dialog.*

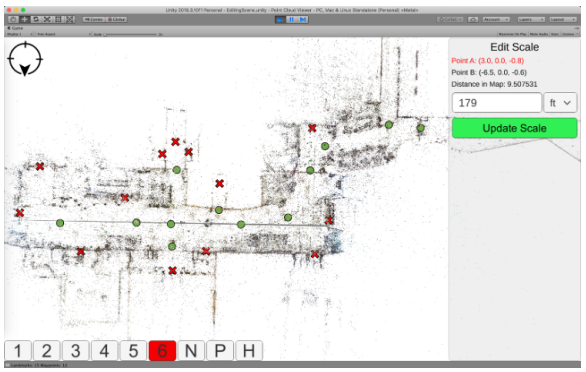


FIGURE 20: *Screenshot of labeling tool - edit scale.*

E. PATH FINDING ALGORITHMS

Algorithm **??** calculates an in-floor path using the proposed data structures. The user's current position will be mapped to a spatial tile — either the tile that contains the user's position or the closest tile to it — by linking a new node representing the user to the existing nodes corresponding to the waypoints and landmarks included within the selected tile using new edges. The algorithm will compute and return the shortest path that starts from the user node and ends at the destination node, where this choice takes into account the distances between spatially located nodes by using them as edge costs. The chosen path will contain a list of waypoints in traversal order. If the path does not include any waypoints, then the destination is in the line-of-sight to the user's current position. To compute a path to a destination on a different floor, we first use Algorithm **??** twice to calculate two in-floor paths, and then connect them with the floor transition landmark (e.g. stair, elevator, or escalator).

F. NAVIGATION INSTRUCTION GENERATION

The navigation instructions are generated only after the three previous processes have been successfully completed: user localization, destination landmark selection, and shortest path computation between the user's position and the destination.

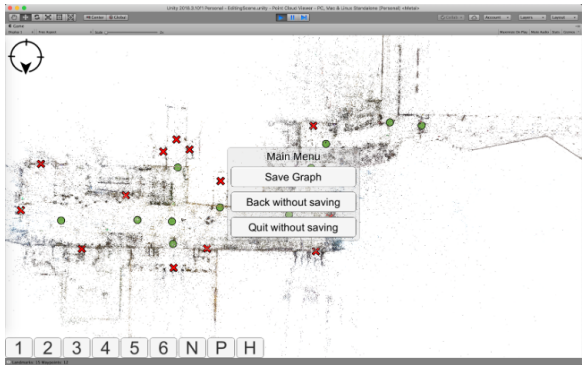


FIGURE 21: *Screenshot of labeling tool - pause menu.*

Algorithm 2 *Calculation of navigation paths using the proposed data structures.*

Input: Sets of tiles $\mathcal{T}$ and regions $\mathcal{R}$; backbone graph $\mathcal{G}$; user's location s; destination d
Output: Path as a list of waypoints p

Find the tile $t_s \in \mathcal{T}$ that contains or is closest to s.
Initialize a list of waypoints $W_s \leftarrow [\cdot]$
for each map region $r \in \mathcal{R}$ containing tile t_s **do**
  if r contains landmark d **then**
    Return an empty path p
  end if
  for each waypoint $w \in r$ **do**
    Add w to W_s
  end for
end for
Initialize a list of paths $W_p \leftarrow [\cdot]$
for each waypoint w in W_s **do**
  Add shortest path from w to d along $\mathcal{G}$ in W_p
end for
Initialize output path $p \leftarrow [\cdot]$ and its distance $d_p \leftarrow \infty$
for each path p_{cand} in W_p **do**
  $d_{\text{cand}} \leftarrow \text{dist}(p_{\text{cand}}) + \text{dist}(s, p_{\text{cand}}[1])$
  if $d_{\text{cand}} < d$ **then**
    $d_p \leftarrow d_{\text{cand}}$
    $p \leftarrow p_{\text{cand}}$
  end if
end for
Return path p

Once the navigation is started, the user will be instructed with a series of verbal announcements describing the relative position of the next waypoint (or destination if it is within line-of-sight), which we refer to as the anchor. The path is continuously updated based on the user's tracked location via inertial odometry. Thus, the user will always receive the timely updated instruction directing them to follow the path. Figure **??** illustrates the flowchart of the navigation instructions update.

Two types of announcements are posted in parallel with navigation instructions. One is the region change announcement. Once the device's position is changed from one announced region to another. A message in the form of "you have left ... and entered ..." will be queued for the announcement. Without any region change, the current region will be repeatedly announced at a low frequency (once per 12 seconds) in the form of "you are in ..." or "you are in the intersection area of ... and ...". This will help the user to understand different regions in the building structure. The other type of announcements contains the landmarks passing along the path. Any trip in the environment is a good opportunity for the user to learn what landmarks exist along the path. Once a landmark is in a certain range of user's current location, a message in the form of "... is at your ... o'clock" will be queued for the announcement. This can serve as an additional context to build a user's mental map efficiently.

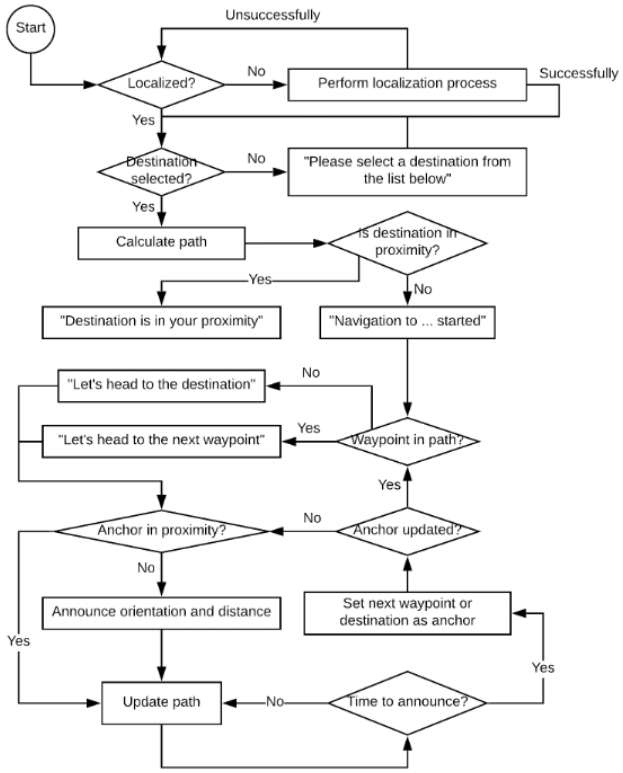


FIGURE 22: *Flowchart of navigation instruction generation.*

VIII. USER INTERFACE

In this section, we summarize the user interface of the two user-facing processes (the picture-based localization interface and the navigation interface) described in the prequel.

A. PICTURE-BASED LOCALIZATION INTERFACE

Recall that the user needs to take a few pictures with their smartphone in order to obtain their location in the physical environment. From the observation of preliminary trials with three BVI users, we noticed that the success of localization using vision-based technique is very sensitive to

the quality and angle of the picture taken by the user. We also observed that some of the BVI users find it difficult to take a feature-rich picture with the smartphone, and require specific feedback for this process. Therefore, we designed a verbally guided user interface for image capturing.

When the user triggers the localization function, they are instructed to take three pictures by panning the device while facing horizontally, receiving verbal guidance during the process. The interface uses ARKit to monitor the device relative movements. The smartphone will vibrate and prompt the user to "hold still" when the facing direction is more than 30 degrees different from the last picture. The user is asked to keep the device still for 2 seconds to avoid taking motion blur in the pictures.

B. NAVIGATION INSTRUCTIONS

Once the user is successfully localized, they can select the destination from a list of landmarks. Then, the application will start to announce the navigation instructions. During navigation, the instructions will be continuously announced to direct the user to the destination unless one of the following interactions occur:

- A different landmark is selected as the destination.
- Localization is triggered.
- Navigation is canceled.

C. ACCESSIBILITY ANNOUNCEMENT SERVICE

There are different functions announcing verbal instructions to the user; however, the audio output is in a linear form, which can only deliver instructions one after another, and each instruction may take several seconds to finish. In order to organize these announcements in the designated priority and order, we implement an extended announcement service in the application that uses the iOS built-in service VoiceOver, an accessible user interface for BVI users. VoiceOver reads the description of UI components on the screen as the user goes through them, and supports third-party application announcements through system calls. By default, if a new announcement is posted, any current announcements will be flushed immediately. This may cause interruptions in the user experience as they receive instructions.

Based on the purpose of different announcements generated by our application, we designed a queue-based announcement service with different mechanisms that allow for the navigation instructions, UI announcements, and interaction guidance to be easily managed based on the nature of their requirements.

1) Queue structure: Since the announcement needs to be spoken out, this linear operation takes time to finish. While announcing the current content, new announcements may be posted. They'll be put into a queue in the posting order and wait to be processed.
2) Lifetime: Some announcements contain time-sensitive information, such as the relative position of a landmark. If it's not announced in a short period of time, the user may move to a different position or face to another direction. Thus, each announcement has a fixed lifetime. An announcement that runs out of time will become stale and ignored during processing.
3) Repeatability: Any announcements generated by the application, e.g., navigation instructions, can be managed entirely within it, while announcements posted by the system, e.g., the user's interaction on UI, are not managed by our application. Thus, the application's announcements can interrupt and be interrupted by the system's announcement. Repeatability is added to ensure that an important application announcement is re-announced if interrupted.
4) Completion callback: Every announcement attaches an optional callback, which will be called once the announcement is spoken out successfully or interrupted. It gives the executability of processing depending on the completion of an announcement.
5) Interruption and Priority: Our accessibility announcement service can also provide intentional interruption of the application's announcements. For example, some optional announcement can be flushed by a more important and time-sensitive announcement. To scale the importance of an announcement, a priority value is assigned to each announcement. High priority announcements can interrupt low priority ones, but the latter will be pushed to the head of the queue if high priority announcements are being processed.

IX. IMPLEMENTATION AND HUMAN TRIALS

Since the vision-based localization algorithm is too computationally intensive to execute in a mobile device, we implemented the system in a client-server architecture as follows.

Client: An iOS application that leverages the built-in accessible user interface for the blind and visually impaired users (VoiceOver). The application includes all user interfaces, the real-time localization module, the path-finding module, the navigation instruction generation module, and the extended accessibility announcement service.

Server: A C++ REST API service that includes a camera calibration module, the picture-based localization module, and the module organizing the graphs of building information and sending the corresponding graph in JSON to the client by request.

A. HUMAN TRIALS

The purpose of the study is to assess the usability of the proposed system. All aspects of the human-subject-trials have been approved by the UMass Institutional Review Board (IRB). Participants in the study must be:

- 18 years of age or older,
- legally blind or visually impaired, and
- have no further mental or physical impairment.

Each session includes three parts: orientation, trial, and a qualitative questionnaire.

Part I — Hands-On Orientation: In a sit-down orientation, the participant is introduced to the app by the experimenter, who goes over the application functionality and answer any questions the subject may have. On an on-site orientation, the participant uses the app in the UMass Campus Center to navigate to destinations that are not included in the actual trial in order to allow the participant to become familiar with the use of the testing mobile app. When the participant is comfortable, they move to the trial part.

Part II — PERCEPT-V Trial: During the trial, the experimenter will ask the participant to perform seven predetermined navigation tasks. For a navigation task, the participant is asked to navigate to a destination from a specific location in the UMass Amherst Campus Center. We say that a navigation task is successful when the participant reaches the destination independently, relying only on their mobility skills and the application feedback. When the participant believes they reached a destination, they inform the experimenter. The trial ends either when all destinations are reached, or the participant chooses to stop the trial.

Part III — Post Trial Questionnaire: After completion of the trial the experimenter collects participant's feedback and experience using a qualitative questionnaire.

B. RESULTS

Participant Baseline: There was a total of nine participants, of which seven were male and two were female. Two participants were blind, two had limited perception of light, and the remaining five participants had varying degrees of usable vision. Six of the participants use a white cane as a mobility aid, with one using it occasionally and not using it during the trial; one participant uses a guide dog.

Observations: The experimenter observed each participant while performing the set of seven pre-determined navigation tasks. All participants were successful at completing all navigation tasks. Four participants needed assistance in interacting with the application in order to reach the destination. In all cases, this interaction was at the beginning of the journey for either taking the picture or selecting the destination, and all these participants were unfamiliar with how to use VoiceOver on the iPhone. While these participants were given a tutorial during the orientation, they were simultaneously learning how to use the testing app and VoiceOver accessibility services. We also note that none of the participants asked for assistance due to localization failures. The app was able to evaluate the pictures taken by the participants and guided them for retakes as necessary.

From the collected feedback, participants felt that the mobile application for the wayfinding system was easy to learn and use. They thought that the user interface is clear and provided sufficient re-orientation information when they felt lost in the Campus Center. Participants were confident that they would reach their destination using the app. Each participant was asked if they would use PERCEPT-V in the future if it was available to them and they all responded affirmatively.

X. DISCUSSION AND CONCLUSIONS

We have presented a customized approach to SMG that leverages several types of structure available in a set of carefully constructed video captures of the environment of interest. The motivation for our custom SMG problem is the design of a wayfinding system for indoor environments; an SMG and computer-vision based design is very attractive for this system due to its low overhead, as it requires no retrofit of the navigable area and it can be deployed in a smartphone platform. Our approach only requires a careful design of a video capture protocol that provides adequate coverage and exhibits the leveraged structure. The generated spatial map then becomes the cornerstone of each of the components of the wayfinding system, i.e., localization and navigation.

Our system has been tested for the task of smartphone-aided indoor navigation for blind and visually impaired subjects, where computer-vision based approaches have attracted significant attention in recent years [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**], [**?**]. In this case, the SMG is executed and hosted by a centralized server, and the localization is jointly performed using a smartphone client and a spatial registration system in the server. Similarly, the navigation instructions are also computed in the server and are relayed to the user via the accessibility services available in the smartphone platform. Our usability tests for this setting show consistent satisfaction among the users and high levels of success in the test navigation tasks for varying levels of complexity and of user familiarity with the platform used [**?**], [**?**], [**?**], while providing a much simpler deployment strategy than with competing approaches for indoor navigation.

REFERENCES

[1] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, July 2006.

[2] S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building Rome in a day," *Commun. ACM*, vol. 54, no. 10, pp. 105–112, Oct. 2011.

[3] J. L. Schönberger and J. Frahm, "Structure-from-motion revisited," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, June 2016, pp. 4104–4113.

[4] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *European Conf. Computer Vision (ECCV)*, Amsterdam, Netherlands, Oct. 2016, pp. 501–518.

[5] J. L. Schönberger, T. Price, T. Sattler, J.-M. Frahm, and M. Pollefeys, "A vote-and-verify strategy for fast spatial verification in image retrieval," in *Asian Conf. Computer Vision (ACCV)*, vol. 1, Taipei, Taiwan, Nov. 2016, pp. 321–337.

[6] C. Wu, "VisualSFM: A visual structure from motion system," http://ccwu.me/vsfm/, Oct. 2019.

[7] V. Nair, C. Tsangouri, B. Xiao, G. Olmschenk, W. H. Seiple, and Z. Zhu, "A hybrid indoor positioning system for blind and visually impaired using Bluetooth and Google Tango," *J. Technology and Persons with Disabilities*, vol. 6, pp. 61–81, Mar. 2018.

[8] V. Nair, M. Budhai, G. Olmschenk, W. H. Seiple, and Z. Zhu, "ASSIST: Personalized indoor navigation via multimodal sensors and high-level semantic information," in *European Conf. Computer Vision (ECCV) Workshops*, Munich, Germany, Sep. 2018, pp. 128–143.

[9] X. Zhang, X. Yao, Y. Zhu, and F. Hu, "An ARCore based user centric assistive navigation system for visually impaired people," *Applied Sciences*, vol. 9, no. 5, Mar. 2019. DOI: 10.3390/APP9050989.

[10] H. Zhang and C. Ye, "An indoor wayfinding system based on geometric features aided graph SLAM for the visually impaired," *IEEE Trans. Neural Sys. Rehabilitation Eng.*, vol. 25, no. 9, pp. 1592–1604, Sep. 2017.

[11] A. Ganz, S. R. Gandhi, J. Schafer, T. Singh, E. Puleo, G. Mullett, and C. Wilson, "PERCEPT: Indoor navigation for the blind and visually impaired," in *Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBS)*, Boston, MA, Sep. 2011, pp. 856–859.

[12] A. Ganz, J. Schafer, S. Gandhi, E. Puleo, C. Wilson, and M. Robertson, "PERCEPT: Indoor navigation system for the blind and visually impaired: Architecture and experimentation," *Int. J. Telmed. Appl.*, vol. 1, Jan. 2012. DOI: 10.1155/2012/894869.

[13] A. Ganz, J. M. Schafer, Y. Tao, C. Wilson, and M. Robertson, "PERCEPT-II: Smartphone based indoor navigation system for the blind," in *Int. Conf. IEEE Engineering in Medicine and Biology Society (EMBS)*, Chicago, IL, Aug. 2014, pp. 3662–3665.

[14] A. Ganz, J. M. Schafer, Y. Tao, L. Haile, C. Sanderson, C. Wilson, and M. Robertson, "PERCEPT based interactive wayfinding for visually impaired users in subways," *J. Technology and Persons with Disabilities*, vol. 3, pp. 33–44, Oct. 2015.

[15] A. Ganz, J. Schafer, Y. Tao, Z. Yang, C. Sanderson, and L. Haile, "PERCEPT navigation for visually impaired in large transportation hubs," *J. Technology and Persons with Disabilities*, vol. 6, no. 1, pp. 336–353, 2018.

[16] Z. Farid, R. Nordin, and M. Ismail, "Recent advances in wireless indoor localization techniques and system," *J. Computer Networks and Comm.*, vol. 2013, Sep. 2013. DOI: 10.1155/2013/185138.

[17] S. He and S.-G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Comm. Surveys Tutorials*, vol. 18, no. 1, pp. 466–490, Jan. 2016.

[18] H. Wang and F. Jia, "A hybrid modeling for WLAN positioning system," in *Int. Conf. Wireless Comm., Networking and Mobile Computing (WiCom)*, Shanghai, China, Sep. 2007, pp. 2152–2155.

[19] S. Mazuelas, A. Bahillo, R. M. Lorenzo, P. Fernandez, F. A. Lago, E. Garcia, J. Blas, and E. J. Abril, "Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks," *IEEE J. Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 821–831, Oct. 2009.

[20] H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using rss measurements in indoor wireless networks," in *Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Sydney, Australia, Nov. 2012. DOI: 10.1109/IPIN.2012.6418856.

[21] K. Kaemarunsgi, "Design of indoor positioning systems based on location fingerprinting technique," Ph.D. dissertation, U. Pittsburgh, 2005.

[22] L. Jiang, "A WLAN fingerprinting based indoor localization technique," Master's thesis, U. Nebraska-Lincoln, 2012.

[23] J. Niu, B. Lu, L. Cheng, Y. Gu, and L. Shu, "Ziloc: Energy efficient wifi fingerprint-based localization with low-power radio," in *IEEE Wireless Comm. and Networking Conf. (WCNC)*, Shanghai, China, Apr. 2013, pp. 4558–4563.

[24] Y. Wang, Xu Yang, Yutian Zhao, Yue Liu, and L. Cuthbert, "Bluetooth positioning using RSSI and triangulation methods," in *IEEE Consumer Comm. Networking Conf. (CCNC)*, Las Vegas, NV, Jan. 2013, pp. 837–842.

[25] M. E. Rida, F. Liu, Y. Jadi, A. A. A. Algawhari, and A. Askourih, "Indoor location position based on Bluetooth signal strength," in *Int. Conf. Information Science and Control Engineering*, Shanghai, China, Apr. 2015, pp. 769–773.

[26] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh, "Indoor positioning in Bluetooth networks using fingerprinting and lateration approach," in *Int. Conf. Information Science and Applications (ICISA)*, Jeju Island, South Korea, Apr. 2011. DOI: 10.1109/ICISA.2011.5772436.

[27] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based indoor localization with Bluetooth Low Energy beacons." *Sensors (Basel)*, vol. 16, no. 5, May 2016. DOI: 10.3390/S16050596.

[28] F. Bergeron, K. Bouchard, S. Gaboury, S. Giroux, and B. Bouchard, "Indoor positioning system for smart homes based on decision trees and passive RFID," in *Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD)*, Auckland, New Zealand, Apr. 2016, pp. 42–53.

[29] S. S. Saab and Z. S. Nakad, "A standalone RFID indoor positioning system using passive tags," *IEEE Trans. Industrial Electronics*, vol. 58, no. 5, pp. 1961–1970, May 2011.

[30] D. Zhang, L. T. Yang, M. Chen, S. Zhao, M. Guo, and Y. Zhang, "Real-time locating systems using active RFID for Internet of things," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1226–1235, Sep. 2016.

[31] C. Wang, H. Wu, and N.-F. Tzeng, "RFID-based 3-D positioning schemes," in *IEEE Int. Conf. Computer Comm. (INFOCOM)*, Barcelona, Spain, May 2007, pp. 1235–1243.

[32] O. Al-Hammadi, A. Al-Hebsi, M. J. Zemerly, and J. W. P. Ng, "Indoor localization and guidance using portable smartphones," in *IEEE/WIC/ACM Int. Conf. Web Intelligence and Intelligent Agent Technology (WI-IAT)*, vol. 3, Macau, China, Dec. 2012, pp. 337–341.

[33] W. Sakpere, N. Mlitwa, and M. Oshin, "Towards an efficient indoor navigation system: A near field communication approach," *J. Engineering, Design and Technology*, vol. 15, no. 4, pp. 505–527, Aug. 2017.

[34] E. García, P. Poudereux, Á. Hernández, J. Ureña, and D. Gualda, "A robust UWB indoor positioning system for highly complex environments," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, Seville, Spain, Mar. 2015, pp. 3386–3391.

[35] A. De Angelis, J. Nilsson, I. Skog, H. Peter, and P. Carbone, "Indoor positioning by ultrawide band radio aided inertial navigation," *Metrology and Measurement Systems*, no. 3, pp. 447–460, Aug. 2010.

[36] F. H. Raab, E. B. Blood, T. O. Steiner, and H. R. Jones, "Magnetic position and orientation tracking system," *IEEE Trans. Aerospace and Electronic Systems*, vol. 15, no. 5, pp. 709–718, Sep. 1979.

[37] D. Arumugam, J. Griffin, D. Stancil, and D. Ricketts, "Higher order loop corrections for short range magnetoquasistatic position tracking," in *IEEE Int. Symp. on Antennas and Propagation (APSURSI)*, Spokane, WA, July 2011, pp. 1755–1757.

[38] J. Blankenbach, A. Norrdine, and H. Hellmers, "A robust and precise 3D indoor positioning system for harsh environments," in *Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Sydney, Australia, Nov. 2012. DOI: 10.1109/IPIN.2012.6418863.

[39] G. De Angelis, V. Pasku, A. De Angelis, M. Dionigi, M. Mongiardo, A. Moschitta, and P. Carbone, "An indoor AC magnetic positioning system," *IEEE Trans. Instrumentation and Measurement*, vol. 64, no. 5, pp. 1267–1275, May 2015.

[40] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," in *Ubiquitous Positioning Indoor Navigation and Location Based Service*, Kirkkonummi, Finland, Oct. 2010. DOI: 10.1109/UPINLBS.2010.5653681.

[41] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *Int. Conf. Mobile Systems, Applications, and Services (MOBISYS)*, New York, NY, June 2011, pp. 141—154.

[42] R. Montoliu, J. Torres-Sospedra, and O. Belmonte, "Magnetic field based indoor positioning using the bag of words paradigm," in *Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Alcala de Henares, Spain, Oct. 2016.

[43] J. Moutinho, R. Araújo, and D. Freitas, "Indoor localization with audible sound - Towards practical implementation," *Pervasive Mob. Comput.*, vol. 29, pp. 1—16, July 2016.

[44] I. Rishabh, D. Kimber, and J. Adcock, "Indoor localization using controlled ambient sounds," in *Int. Conf. Indoor Positioning and Indoor Navigation (IPIN)*, Sydney, Australia, Nov. 2012. DOI: 10.1109/IPIN.2012.6418905.

[45] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling fine-grained smartphone localization via acoustic anchors," *IEEE Transactions on Mobile Computing*, vol. 15, no. 5, pp. 1144–1156, May 2016.

[46] O. J. Woodman and R. K. Harle, "Concurrent scheduling in the active bat location system," in *IEEE Int. Conf. Pervasive Computing and Communications Workshops (PERCOM)*, Mannheim, Germany, Apr. 2010, pp. 431–437.

[47] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Int. Conf. Mobile Computing and Networking (MOBICOM)*, Boston, MA, Aug. 2000, pp. 32—43.

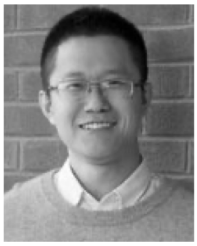
[48] M. Minami, Y. Fukuju, K. Hirasawa, S. Yokoyama, M. Mizumachi, H. Morikawa, and T. Aoyama, "Dolphin: A practical approach for implementing a fully distributed indoor ultrasonic positioning system," in *Int. Conf. Ubiquitous Computing (UBICOMP)*, Nottingham, United Kingdom, Sep. 2004, pp. 347–365.

[49] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, no. 1, pp. 91—102, Jan. 1992.

[50] Y. Zhuang, L. Hua, L. Qi, J. Yang, P. Cao, Y. Cao, Y. Wu, J. Thompson, and H. Haas, "A survey of positioning systems using visible LED lights," *IEEE Comm. Surveys Tutorials*, vol. 20, no. 3, pp. 1963–1988, July 2018.

[51] B. Al-Delail, L. Weruaga, M. J. Zemerly, and J. W. P. Ng, "Indoor localization and navigation using smartphones augmented reality and inertial tracking," in *IEEE Int. Conf. Electronics, Circuits, and Systems (ICECS)*, Abu Dhabi, United Arab Emirates, Dec. 2013, pp. 929–932.

[52] C. P. R. Raj, S. Tolety, and C. Immaculate, "QR code based navigation system for closed building using smart phones," in *Int. Mutli-Conf. Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, Kottayam, India, Mar. 2013, pp. 641–644.
[53] R. M. noz Salinas, M. J. Marín-Jimenez, E. Yeguas-Bolivar, and R. Medina-Carnicer, "Mapping and localization from planar markers," *Pattern Recognition*, vol. 73, pp. 158–171, Jan. 2018.
[54] J. Coughlan and R. Manduchi, "A mobile phone wayfinding system for visually impaired users," *Assistive Technology Research Series*, vol. 25, p. 849, 2009.
[55] N. Piasco, D. Sidibé, C. Demonceaux, and V. Gouet-Brunet, "A survey on visual-based localization: On the benefit of heterogeneous data," *Pattern Recognition*, vol. 74, pp. 90–109, Feb. 2018.
[56] I. Abu Doush, S. Alshatnawi, A. Al-Tamimi, B. Alhasan, and S. Hamasha, "ISAB: Integrated indoor navigation system for the blind," *Interacting with Computers*, vol. 29, no. 2, pp. 181–202, Mar. 2017.
[57] D. Ahmetovic, C. Gleason, C. Ruan, K. Kitani, H. Takagi, and C. Asakawa, "NavCog: A navigational cognitive assistant for the blind," in *Int. Conf. Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, Florence, Italy, Sep. 2016, pp. 90—99.
[58] D. Sato, U. Oh, K. Naito, H. Takagi, K. Kitani, and C. Asakawa, "Navcog3: An evaluation of a smartphone-based blind indoor navigation assistant with semantic features in a large-scale environment," in *ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, Baltimore, MD, Oct. 2017, pp. 270—279.
[59] S. A. Cheraghi, V. Namboodiri, and L. Walker, "GuideBeacon: Beacon-based indoor wayfinding for the blind, visually impaired, and disoriented," in *IEEE Int. Conf. Pervasive Computing and Communications (PerCom)*, Koha, HI, March 2017, pp. 121–130.
[60] J.-E. Kim, M. Bessho, S. Kobayashi, N. Koshizuka, and K. Sakamura, "Navigating visually impaired travelers in a large train station using smartphone and Bluetooth Low Energy," in *ACM Sympo. on Applied Computing (SAC)*, Pisa, Italy, Apr. 2016, pp. 604—611.
[61] N. A. Giudice, W. E. Whalen, T. H. Riehle, S. M. Anderson, and S. A. Doore, "Evaluation of an accessible, real-time, and infrastructure-free indoor navigation system by users who are blind in the Mall of America," *J. Vis. Impairment Blindness*, vol. 113, no. 2, pp. 140–155, Mar. 2019.
[62] S. Ullman and S. Brenner, "The interpretation of structure from motion," *Proc. Royal Soc. London Series B: Biological Sciences*, vol. 203, no. 1153, pp. 405–426, Jan. 1979.
[63] C. Wu, S. Agarwal, B. Curless, and S. M. Seitz, "Multicore bundle adjustment," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, June 2011, pp. 3057–3064.
[64] H. Xu, "OSM-Bundler," http://haiyangxu.github.io/osm-bundler/, Mar. 2015.
[65] P. Moulon, P. Monasse, R. Marlet, and Others, "OpenMVG: An open multiple view geometry library." https://github.com/openMVG/openMVG, July 2019.
[66] J. L. Schönberger, "Robust Methods for Accurate and Efficient 3D Modeling from Unstructured Imagery," Ph.D. dissertation, ETH Zürich, 2018.
[67] J. M. Coughlan and A. L. Yuille, "The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference," in *Neural Info. Proc. Sys. (NeuRIPS)*, Denver, CO, Dec. 2000, pp. 809–815.
[68] M. Buerli and S. Misslinger, "Introducing ARKit-augmented reality for iOS," in *Apple Worldwide Developers Conf. (WWDC 2017)*, San Jose, CA, June 2017, pp. 1–187.
[69] M. A. Foltz, "Designing navigable information spaces," Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1998.
[70] E. W. Dijkstra, "A note on two problems in connexion with graph," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
[71] E. Ko and E. Y. Kim, "A vision-based wayfinding system for visually impaired people using situation awareness and activity-based instructions," *Sensors*, vol. 17, no. 8, Aug. 2017. DOI: 10.3390/S17081882.
[72] Z. Yang and A. Ganz, "Egocentric landmark-based indoor guidance system for the visually impaired," *Int. J. E-Health Med. Commun.*, vol. 8, no. 3, pp. 55–69, July 2017.
[73] ——, "A sensing framework for indoor spatial awareness for blind and visually impaired users," *IEEE Access*, vol. 7, pp. 10 343–10 352, 2019.
[74] H. Dong, J.Schafer, Y. Tao, and A. Ganz, "PERCEPT-V: Integrated indoor navigation system for the visually impaired using vision-based localization and waypoint-based instructions," *J. Technology and Persons with Disabilities*, vol. 8, 2020.
[75] Y. Tao and A. Ganz, "Validation and optimization framework for indoor navigation systems using user comments in spatial-temporal context," *IEEE Access*, vol. 7, pp. 159 479–159 494, Nov. 2019.
[76] ——, "Simulation framework for evaluation of indoor navigation systems," *IEEE Access*, vol. 8, pp. 20 028–20 042, Jan. 2020.



HAO DONG received the B.Sc. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2011, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts, Amherst, MA, USA, in 2014 and 2019, respectively. He is currently a researcher at Google Inc., Mountain View, CA, USA. His research interests include virtual reality, augmented reality, assistive technologies, and computer vision.

SUSHMA SURESH BABU received the B.Sc. degree in Electronics and Communications Engineering from the People's Education Society University, Bengaluru, India, in 2019. She is currently a M. Sc. student in Electrical and Computer Engineering at the University of Massachusetts, Amherst, MA, USA. Her research interests include signal and image processing, assistive technologies, computer vision, and machine learning.

MARCO F. DUARTE (Senior Member, IEEE) received the B.Sc. degree (Hons.) in computer engineering and the M.Sc. degree in electrical engineering from the University of Wisconsin-Madison, Madison, WI, USA, in 2002 and 2004, respectively, and the Ph.D. degree in electrical and computer engineering from Rice University, Houston, TX, USA, in 2009.

He was an NSF/IPAM Mathematical Sciences Post-Doctoral Research Fellow of the Program of Applied and Computational Mathematics with Princeton University, Princeton, NJ, USA, from 2009 to 2010, and with the Department of Computer Science, Duke University, Durham, NC, USA, from 2010 to 2011. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA, USA. His research interests include machine learning, compressed sensing, sensor networks, and computational imaging.

Dr. Duarte is a member of Tau Beta Pi. He received the Presidential Fellowship and the Texas Instruments Distinguished Fellowship in 2004, and the Hershel M. Rich Invention Award in 2007 from Rice University. He was a recipient of the IEEE Signal Processing Society Overview Paper Award (with Y. C. Eldar) in 2017 and the IEEE Signal Processing Magazine Best Paper Award (with M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk) in 2020.

AURA GANZ (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from Technion, Haifa, Israel.

She is currently a Professor Emeritus with the Electrical and Computer Engineering Department and the Director of the 5G Mobile Evolution Laboratory, University of Massachusetts, Amherst, MA, USA. She has more than 25 years of experience in research, development, implementation and testing of wireless systems and systems related to healthcare settings, such as assistive technologies for the blind, disaster informatics mobile tele-medicine, and tele-surgery. This work has resulted in more than 250 journal and conference publications in highly respected refereed journals (IEEE, IEE, ACM, Kluwer, and Elsevier) with multiple best paper awards. The external recognition of her work is also evidenced by having been selected to serve on leading roles in many professional conferences and workshops and being elected as IEEE Fellow. Her research was continuously funded by federal agencies, such as NSF, NIH, ARO, AFOSR, and DARPA, state agencies, such as MassDOT, and industry, such as Microsoft, Intel, and EMC.