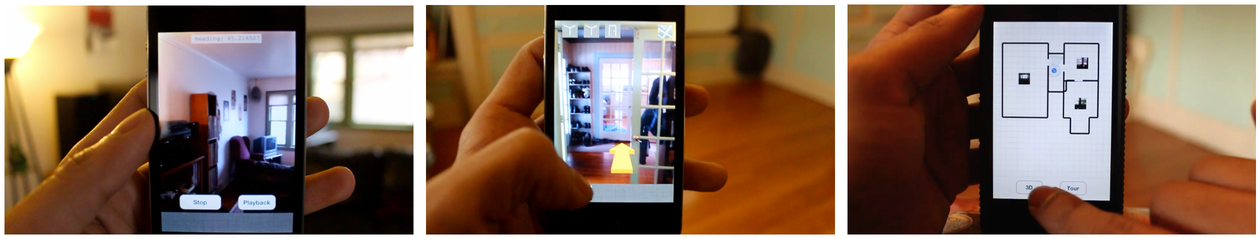
Capturing Indoor Scenes with Smartphones

Aditya Sankar[1]
[1]University of Washington
Seattle, WA, USA
{aditya, seitz}@cs.washington.edu

Steven M. Seitz[1,2]
[2]Google Inc.
Seattle, WA, USA



(a) Capture data (b) Playback interactive tour (c) Generate indoor floor plan

Figure 1: Our system captures an interactive visual tour and a floor plan of an indoor scene, in real time, on a smartphone.

ABSTRACT
In this paper, we present a novel smartphone application designed to easily capture, visualize and reconstruct homes, offices and other indoor scenes. Our application leverages data from smartphone sensors such as the camera, accelerometer, gyroscope and magnetometer to help model the indoor scene. The output of the system is two-fold; first, an interactive visual tour of the scene is generated in real time that allows the user to explore each room and transition between connected rooms. Second, with some basic interactive photogrammetric modeling the system generates a 2D floor plan and accompanying 3D model of the scene, under a Manhattan-world assumption. The approach does not require any specialized equipment or training and is able to produce accurate floor plans.

Author Keywords
Handheld Devices and Mobile Computing; Visualization; Virtual Reality; Augmented Reality

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces - Graphical user interfaces (GUI); H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems - Artificial, augmented, and virtual realities

General Terms
Algorithms; Design; Human Factors

INTRODUCTION
In 1979, Lippman et al. [12] used gyrostabilized cameras mounted on top of a car to create an interactive visualization of downtown Aspen. This early hypermedia system, a forerunner of Google's Street View [1], pioneered the use of spatially indexed imagery for generating interactive virtual tours. Their system allowed users to interactively explore an environment and presented the user with a sense of *'being there'*. The desire to enhance and improve the notion of virtual presence has subsequently fueled a sizable body of work around interactive visual tours. However, most of these approaches [24, 21, 1, 9] rely on specialized data acquisition equipment and have complex and time-consuming offline processing pipelines, making them inaccessible to the casual user.

In this work, we present a system to easily create and view interactive visual tours, in real-time, entirely on a smartphone. To create a tour, the user first captures video of an indoor scene by following a few simple guidelines. This video data is then spatially and temporally indexed using sensor readings from the smartphone's gyroscope, accelerometer and magnetometer. Our system then generates an image-based rendering of the scene that allows a viewer to interactively playback a virtual tour in which he/she may explore each room and 'walk' between connected rooms. Once the data capture is complete, we use an interactive photogrammetric modeling scheme to recover room features such as the location of room corners and doors from within the image data. With this information we estimate the position of camera and generate an approximate floor-plan representation of the environment. Our method is designed to capture *Manhattan World* [5] scenes, i.e., scenes wherein all the surfaces are aligned with three dominant directions. The computed floor plans correctly capture the layout of the scene and are used to augment the interactive tour, enabling the user to understand where they are in the scene.

There are several challenges in making such a system work robustly, in real-time, on a mobile device. First, phones lack accurate localization information or odometry (GPS does not work well indoors [26]), making it hard to estimate camera position. Second, the monocular video data lacks depth information about room dimensions or floor plan layout. Extracting this information from video data in real-time is a difficult task. Finally, working within the computational and memory constraints of an embedded device pose additional engineering challenges.

The main contributions of this work are a framework for capturing and spatially indexing visual data by using a combination of measurements from the gyroscope, accelerometer and magnetometer, and a novel interactive technique to estimate room geometry and floor plan layout from monocular video. At a system level, the novelty lies in the implementation of an end-to-end smartphone application capable of capturing and reconstructing indoor scenes. We believe that our application is the first of its kind to deliver an interactive tour experience coupled with a floor plan on-the-fly, entirely on a phone. We foresee the use of our system in areas such as real estate, interior design, indoor localization and mobile robot navigation.

In the remainder of the paper we discuss related work, present an overview of the application from a user's perspective, describe novel components of our system, present floor plan results and evaluate them against ground truth. We conclude with future research directions.

RELATED WORK

The idea of indoor interactive video-based tours has been explored by several authors in graphics, vision and HCI. Brooks [2] in 1986 was one of the first to propose a system to build rapid visual prototypes of buildings for architectural use. More recently Uyttendaele et al. [24] used omnidirectional video to create indoor virtual tours. Similar approaches are used in Google's Streetview [1] and Art Project [9]. Our work is similar, in that the user can can interactively look around an environment, move freely along predefined paths, and branch to different areas at decision points. These prior approaches required sophisticated omnidirectional camera rigs and several hours of offline processing, whereas our method is implemented end-to-end on a commodity smartphone and is able to generate tours in a few of minutes. Our video-based tour approach is similar to that of Quiksee [15] but differs in that they used a hand-held camcorder and an offline processing pipeline. Their method requires manual spatial registration (as camcorders lack gyroscopic sensors) and does not model the geometry of the scene. Quiksee has since been acquired by Google Inc. and no published information is available on their method.

Early mobile robot navigation systems such as those proposed by Ishiguro [10] and Yagi [25] utilized omnidirectional camera systems combined with odometry measurements to reconstruct environments for mobile robot navigation. Taylor [21] also estimated camera position and environment geometry from video data. Their approach required the user to specify several point-and-line correspondences in key-frames of the omnidirectional video, whereas our method requires users to mark each wall corner once in an interactive panorama of the room. The visual SLAM and computer vision communities have developed automatic approaches [6, 7, 19, 4, 20] to reconstruct indoor scenes from images. While computer vision-based 3D reconstruction has demonstrated potential, we avoid explicit reconstruction with computer vision as it tends to be brittle, computationally expensive and does not work well on texture-poor surfaces, e.g. painted walls, which dominate interiors. SLAM-based reconstruction has been shown to work on smartphones by Shin et al. [18] who used sensor data to model indoor environments. Their approach is restricted to modeling only corridors while our method describes rooms, their volumes and also presents an interactive visualization of the space. A Manhattan-world assumption was employed by Kim et al. [11], to acquire indoor floor plans in real-time. Their approach is hardware-intensive, requiring the user to carry a Kinect camera, projector, laptop and a special input device while capturing data around the house.

The recent shift of imaged-based systems to the mobile phone platform is exemplified by the mobile Photosynth application [14] that creates panoramic images in real-time on a smartphone. MagicPlan [17] is a commercial floor plan generation app available for the iPhone. By marking floor corners in the room via an augmented reality interface, their system is able to estimate dimensions of the room and generate a corresponding floor plan. While their algorithms are proprietary and unpublished, they likely estimate and use the height of the observer (in this case the phone camera) and the tilt of the camera to estimate scene depth. By estimating depth in this manner, MagicPlan is able to capture non-Manhattan world scenes. This approach is susceptible to error when floor corners are occluded by furniture, requiring the user to guess at their positions. Our approach uses wall edges (the lines formed where two walls meet), which typically run from floor to ceiling and are often visible in rooms regardless of furniture placement and other occlusions. MagicPlan reconstructs rooms individually and then has a user manually assemble them to form a complete floor plan, whereas in our approach, we solve for correspondences between rooms and assemble the entire floor plan automatically. Finally, the augmented reality approach employed by MagicPlan is quicker and allows users to mark floor corners and doors in a single step. In contrast our two-step approach requires users to first capture imagery of the room and thereafter mark room features over the captured imagery. However, MagicPlan focuses solely on floor plan generation, whereas our goal is to produce a sense of virtual presence, to which end we use the captured imagery to visually reconstruct an immersive tour of the scene. A comparison of floor plans generated by both the approaches is presented in the results section.

APPLICATION OVERVIEW

This section describes the function of our application at a user-level. Our application is implemented on an iPhone 4, running iOS 5.1. However, the technique is designed to work on any device that is equipped with a camera, gyroscope, accelerometer and magnetometer; features that are now standard in the latest smartphones and tablets. We encourage readers to view the accompanying supplementary video [3]

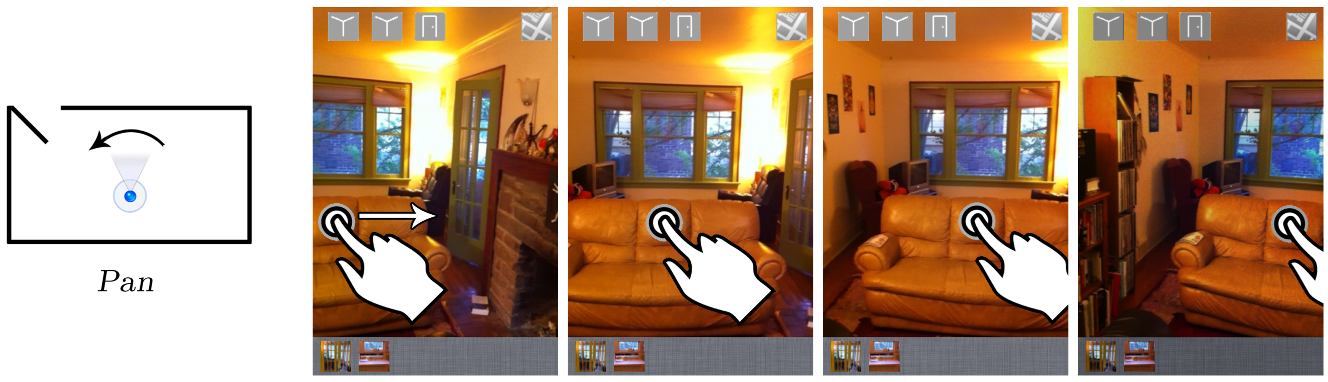


Figure 2: The user may pan left or right in the panorama by dragging

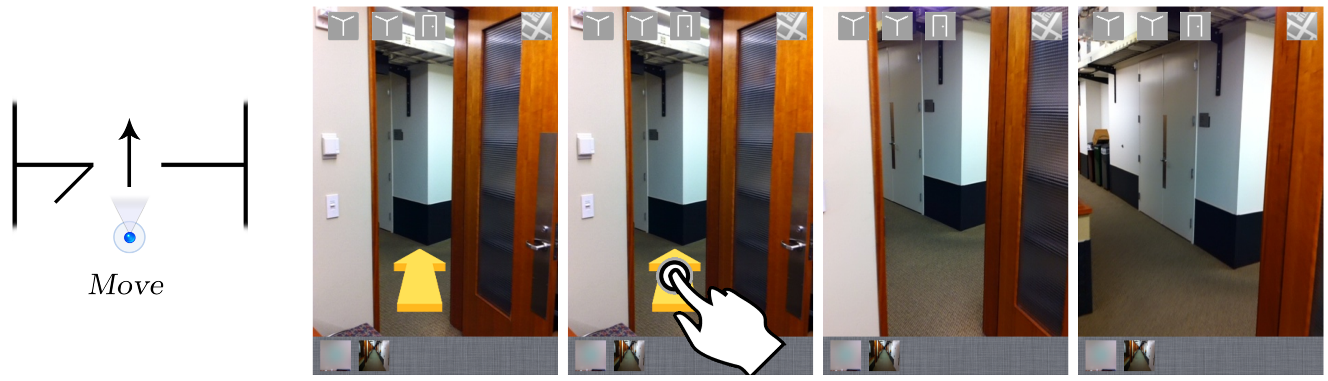


Figure 3: By tapping on the motion arrow, the user may move between rooms

in order to gain a better understanding of the application's interface. The basic steps in the approach are outlined below and in Figures 1, 2 & 3:

1. **Data acquisition**: (Figure 1a) To capture a room, the user stands near the center of the room, holding the phone upright and aiming the phone camera at a wall. He/she then proceeds to rotate 360° to capture a video of the entire room. The application automatically detects when a full rotation has been completed and prompts the user to turn and face the next room. Once in position, the user indicates that he/she is ready to move to the next room by performing a *swipe up* gesture and subsequently proceeds to walk to the next room. Our system records this transition as a video. On reaching the center of the next room, the user performs a *swipe down* gesture and begins to capture the current room. The same process is repeated for every room. If a room is re-visited, the user has an option to indicate this in the interface. Captured rooms appear as thumbnails on the bottom of the screen.

2. **Interactive Tour Playback**: (Figure 1b) Once capture is complete, the user can instantly view a virtual tour of the scene. For each room, the system generates a 360° video panorama, in which the user may pan left of right to explore the scene (Figure 2). The user is afforded two modalities to interact with the panorama. They may drag left or right on the touchscreen or physically move the device in 3D space to '*look around*' the room, in a manner reminiscent of mobile virtual/augmented reality applications [16, 23]. When a door appears in the users current field of view, a motion arrow [8] appears on the screen, indicating that there is a path leading to an adjacent room. Upon tapping the arrow, a transition video is played back giving the user the effect of 'walking' into the next room (Figure 3). At the end of the transition, the system presents the 360° video panorama of the current room, wherein the user may once again explore interactively.

3. **Embedding Close-up Images and Text**: The user has the ability to embed close-up photographs and text in the scene, during playback. To add an embedded object, the user double taps a point of interest on the screen. This instantiates a modal dialog that prompts the user to take a high-resolution photograph of the point of interest and optionally add some descriptive text. Upon completing this task and closing the dialog, the user is returned to the tour and the embedded object now appears as an icon overlaid on top of the panorama. The icon is positioned at the location where the user originally performed the double tap gesture and is pinned to that location as the user pans the panorama left or right. Tapping the icon spawns a modal

popup that shows the hi-res photo along with the annotation text. The embedded objects are saved along with the virtual tour data for future viewing.

4. **Marking Room Features**: In playback mode, the user can mark room features such as wall edges (where two walls meet) and doors via an intuitive touch interface. To mark a wall edge the user first brings the edge into view by panning left or right in the panorama. Once the edge is visible, the user drags a marker to align with the room corner in the image. Similarly, doors are indicated by placing two markers on the extremities of the door. Note that each door is visible from both the rooms that it connects. This door correspondence is also specified by the user, (i.e., which room does a specific door lead to), by tapping the appropriate room thumbnail.
5. **2D Floor Plan**: (Figure 5c) An optimization algorithm uses the corner and door information and other room constraints to generate a 2D floor plan of the space. Alignment errors in the reconstructed floor plan can be manually corrected via a touch interface. To fix an alignment error, the user first taps a wall that requires re-alignment and next taps a wall to which the previously selected wall should be aligned. The walls can be in the same room or different rooms, but they must be parallel. The floor plan is rendered as a top-down map of the environment and indicates the users current position in the tour.
6. **3D Floor Plan**: (Figure 5d) We extrude the 2D floor plan vertically and render a 3D model of the scene. The 3D rendering can be manipulated via the touch interface and users can explore novel views of the environment (e.g., a $45°$ bird's-eye view).

SYSTEM COMPONENTS

Spatial Video Indexing

Indexing video spatially with camera pose information is a simple yet effective method of creating an interactive viewing experience [12]. During playback, by interactively selecting a desired viewpoint and viewing direction, we can retrieve the closest view from the previously captured video sequence. Interactive video playback has advantages over a traditional composited panorama in that it captures dynamic exposure changes and motion in the environment, which are essential to the richness and realism of the visualization [24]. However, the disadvantage of using a video sequence is the inability to zoom-out. While the idea of spatial video indexing has been explored before [21, 24], we present a few novel developments that help adapt it to a smartphone.

Calculating Camera Pose:
The on-board gyroscope on the iPhone 4 provides fine-grained information about the angular velocity of the device; however integrating this data to calculate relative orientation results in drift and requires calibration. The accelerometer and magnetometer conversely provide more reliable orientation information, but at a lower resolution and refresh rate. We use a simple sensor fusion approach to calculate pose by combining data from the gyroscope, accelerometer and magnetometer as follows. The rotation rate of the device around three axes is obtained by querying the gyroscope at regular intervals. The major axis of rotation is determined by querying the accelerometer to determine the direction of gravitational acceleration and hence the device orientation (portrait v/s landscape). Once the major axis is known, we bootstrap the camera pose with the actual orientation of the device relative to true north, as obtained from the magnetometer. Thereafter current orientation is determined as follows:

$$\theta_t^G = \theta_{t-1}^G + \Delta t \times \omega^{majoraxis} \tag{1}$$

Where θ_t^G represents current device orientation, as calculated by adding the delta change in orientation $\Delta t \times \omega^{majoraxis}$ to the last known orientation θ_{t-1}^G. This value is updated at approximately 100Hz, but suffers from gyroscopic drift of as much as $\pm 10°$ in a full $360°$ rotation. We then spatially index the current image with the calculated device orientation value.

Countering Gyroscopic Drift:
In order to counter gyroscopic drift, we rely on the magnetometer to inform us when a full $360°$ rotation has been completed. While the intermediate magnetometer readings are noisy and not useful for updating device orientation, we have found that the resolution is sufficient to indicate when a rotation has been completed. Once we know that the rotation is complete, we can close the "gap" caused due to drift. For this we use the initial and current gyroscopically calculated orientation values and eliminate the drift by uniformly distributing it across the entire rotation.

User Interaction to Mark Room Elements

Once the video and orientation data have been captured, the tour is played back interactively. The playback interface is similar to previous work such as Street View [1] and VideoPlus [21] which allow the user to interactively pan within a 'viewing bubble' and branch to different bubbles at decision points.

During playback, we also present an interface to mark room elements such as corners and doors. In this interface the user is presented with a draggable marker which they can place over the current image to indicate the presence of a corner or door. Marking a corner requires only one marker, whereas marking the extremities of a door requires two. The relative heading of each marker is calculated from the indicated x-coordinate of the marker within an image, by the following equation:

$$\theta_{marker} = \theta_t^G + fov/w_{img} \times (P_{marker} - w_{img}/2) \tag{2}$$

The spatial index, θ_t^G, of the image on-screen corresponds to the heading of the center of the image. The one-dimensional pixel-offset of the marker from the center of the image of width w_{img} is calculated by subtracting the position of the center pixel, $w_{img}/2$, from the position of the marker, P_{marker}. The pixel-offset is then converted into a heading-offset by multiplying by the field of view of the image, fov, and dividing by the total pixel width, w_{img}. Finally, the estimated heading of the marker, θ_{marker}, is determined by adding the heading offset to the current spatial index, θ_t^G.

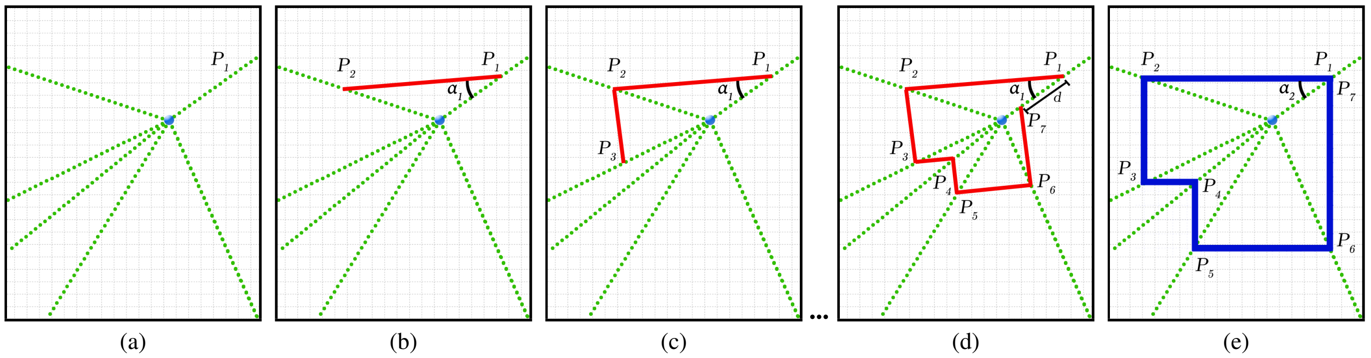


Figure 4: Floor plan reconstruction algorithm: (a) Top-down view of the scene showing the corner rays (dotted green) emanating from the camera. P_1 is unit distance from camera along first ray. (b) Hypothesized wall at angle α_1 intersects second ray at P_2. (c) The next wall is rotated 90 degrees, and intersects at P_3. (d) Incorrect hypothesis results in a distance error d between first and last points. (e) Optimal wall configuration with starting wall angle α_2, minimizes d.

During tour playback, the following equation (2) is used to calculate position, $P_{feature}$, of a feature with a known heading $\theta_{feature}$ within an image.

$$P_{feature} = w_{img}/2 + w_{img}/fov \times (\theta_{feature} - \theta_t^G) \quad (3)$$

This allows us to embed UI elements such as motion arrows and annotations into the scene.

Floor Plan Generation

Computing Room Geometry:
We treat the marked corners and doors as rays emanating from the camera center and reconstruct the room geometry that fits the given set of rays (as shown in Figure 4), based on the following assumptions. First, we assume that the scene satisfies the Manhattan world assumption, i.e., the walls are planar and the room corners are right-angled. Many architectural scenes approximately fit the Manhattan World assumption, and it has previously enabled very high quality reconstruction results across a wide range of scenes [7]. We also assume that the camera is located in a position within the room from which all corners are visible, or if this is not possible, that the user is able to estimate and mark the location of occluded room features. Finally, we assume that the corner ray directions are determined from the spatial indexing data, as described in the previous section.

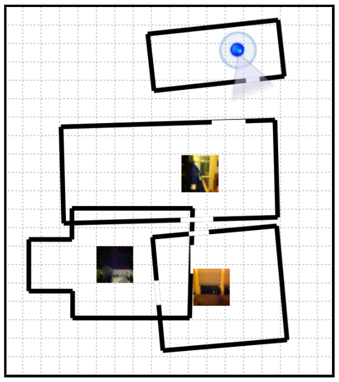
Search Algorithm:
Our search algorithm over the possible room configurations is described below and in Algorithm 1.

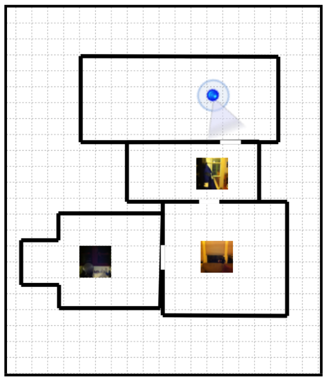
Without loss of generality, assume the camera is located at the scene origin, and the first room corner is at unit distance from the origin (we solve for the room only up to scale). Now suppose we also specify the orientation α (relative to the first corner ray) of the first wall. Under the Manhattan World Assumption, the orientations of all other walls are determined (since they meet at right angles). It also turns out that their lengths can also be inferred from the known corner ray directions (as we will describe shortly). The approach works as follows. Let P_1 be the point at unit distance from the origin along the ray direction of the first corner (Figure 4a). The intersection of the ray from P_1 at angle α with the ray from the origin along the ray direction of the second corner defines the position P_2 of the second corner in the room (Figure 4b). Rotate 90 degrees and repeat this procedure to intersect the ray from P_2 with the ray from the origin along the direction of the third corner to define P_3(Figure 4c), and so on. This procedure will place all of the corners in the plane. The key insight is that for the correct value of α, the final ray from P_N should intersect P_1 (Figure 4e), assuming N corners. An incorrect value of α will generally not satisfy this criterion (Figure 4d). Our approach is therefore to do an exhaustive search over all angles (0 - 360, sampled at 0.5 degree increments), to solve for an α that yields the closest intersection with P_1 (see Algorithm 1). Our algorithm finds an optimal solution in $O(Nd)$ running time where N is the number of corners in the room and d is the number of angles sampled in our search.

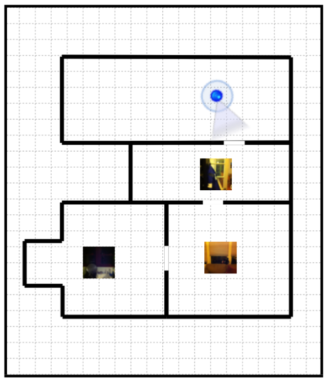
Algorithm 1 Calculating optimal wall configuration

```
1: $P_0$ = Origin
2: $P_1$ = Unit distance from $P_0$ along the first ray
3: $N$ = Number of corners
4: $minDistance = MAX\_FLOAT$
5: for $\alpha = 0 \rightarrow 360$ step 0.5 do
6:     for $i = 2 \rightarrow N$ do
7:         $\theta_i$ = direction of $i^{th}$ corner        ▷ from (2)
8:         $P_i \leftarrow intersect(P_{i-1},\ \alpha + (i-1) \times 90,\ P_0,\ \theta_i)$
9:     end for
10:    $distance = getDistance(P_1, P_N)$
11:    if $distance < minDistance$ then
12:        $minDistance \leftarrow distance$
13:        $minAngle \leftarrow angle$
14:    end if
15: end for
16: Return $minAngle$
```

Note that in Algorithm 1, $intersect(P_A,\ \theta_A,\ P_B,\ \theta_B)$ finds the intersection point of two lines A and B defined in point-


(a) Floor layout as estimated by odometry, with current camera position indicated by the camera frumstum


(b) Solving for the door constraint translates, scales, and rotates rooms to align properly


(c) Layout after manual edits have been made via a touch interface

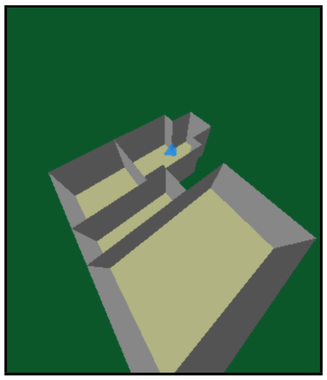

(d) Reconstructed 3D model of home.

Figure 5: Stages of solving floor layout.

slope form and $getDistance(P_A, P_B)$ calculates the distance between points P_A and P_B in a euclidean space.

Aligning Rooms:
Since our goal is to generate a complete floor plan of an indoor scene, we must calculate relative positions and orientations of rooms to each other. We make an initial best guess with the help of some approximate odometry. If we assume that the user walks at a constant pace between rooms during capture, we can estimate relative positions of rooms, using the user's movement direction and length of the transition video sequence. However this method is prone to error (as is seen in Figure 5a) due to the fact that users may move with a non-uniform velocity and change direction mid-flight.

We can more accurately calculate relative room positioning by using the supplied door rays. Once the walls have been calculated via Algorithm 1, we can find the projected coordinates of the door by intersecting the door rays with the walls. Since each door is visible from exactly two rooms (as indicated in the interface by the user), we can proceed to calculate a 2D rigid body transformation, to align them. The transform is written as:

$$\begin{pmatrix} x'_i \\ y'_i \end{pmatrix} = \begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \qquad (4)$$

Here $c,\ s,\ t_x$ & t_y are unknowns; by substituting the values of corresponding doors in two rooms, we can solve for the transformation that correctly aligns the doors, and hence also aligns both rooms. The matrix equation can be solved by standard least squares fitting methods.

We solve for the transformations between every pair of connected rooms and are able to generate a complete floor plan as shown in Figure 5b. This intermediate solution still contains errors, such as misaligned walls and incorrectly scaled room dimensions. The errors are caused due to several factors; gyroscopic drift not being eliminated completely, human error in marking room features (typically off by a few pixels), error due to discretization of the search angle α and the fact that the physical walls may not be perfectly axis aligned. There may also be error caused due to parallax as the user may slightly move the camera center while capturing the scene.

To alleviate errors, we allow users to manually correct wall alignment using a simple touch interface, as described in the Application Overview section. The end result of this process is depicted in Figure 5c, which more accurately describes the room layout in the house. In this particular example, five corrections were required to produce the final result.

3D Rendering
3D reconstructions of a scene allow for generating novel views and a more immersive spatial visualization. Once we have a suitable 2D floor plan rendering, we extrude it vertically and render a floor plane to obtain a representative 3D model of the environment as shown in Figure 5d. To provide the user with additional spatial context, we render a small camera frustum that represents their current position and orientation in the tour.

RESULTS AND DISCUSSION
We tested our application in four indoor environments. For each environment we generated an interactive visual tour and floor plan. The resulting tour experience has been described previously in the Application Overview section and a demonstration of the interface is available in the supplementary video [3]. In this section we will present the results of floor plan generation.

Recovering Individual Room Geometry
In Figure 6 we present the results of individual room reconstruction by our approach and that of MagicPlan. Since our floor plan reconstruction is not to scale, for the overlay we manually scale and register our result over ground truth. Room 1 took 1 minute and 12 seconds to capture with our approach (44s capture + 28s marking room features) and 53

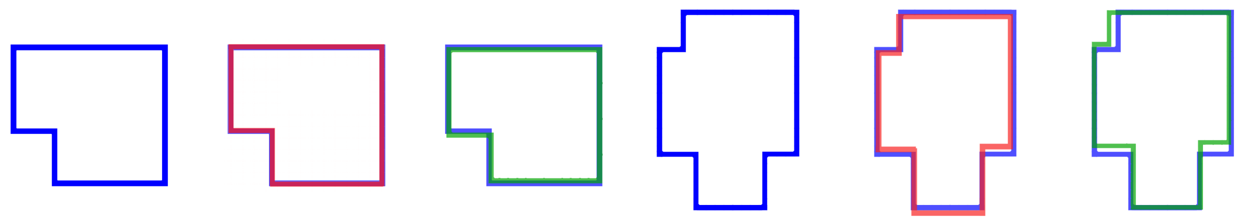


Figure 6: Room geometry recovered by our approach (red) and by MagicPlan (green), when overlaid on ground truth (blue)

seconds to capture with MagicPlan. Room 2 took 1 minute and 34 seconds with our approach (46s capture + 48s marking room features) and 1 minute and 4 seconds with MagicPlan. The running time for both reconstruction algorithms was negligible. Our image capture time remains roughly the same regardless of room configuration, but more complex rooms require longer to mark room features. While the total time taken by MagicPlan is less, our approach also captures imagery of the room for creating an immersive visual tour experience, whereas MagicPlan outputs only a floor plan. The examples in Figure 6 show that our approach is able to capture the shape of the room with similar accuracy to MagicPlan. Note, however that our reconstruction does not provide metric scale, while MagicPlan does.

Generating Floor Plan Layout

In Table 1 and Figure 7 we present results from four indoor environments:

		Features			No. of	Total Time	Error
Env.	*Type*	R	C	D	Frames	Taken	(%)
A	House	5	26	4	1356	7m16s	5.66
B	House	6	42	5	2107	9m42s	13.38
C	House	5	26	5	1571	8m40s	8.76
D	Office	7	28	6	3422	12m35s	13.98
Avg.						9m33s	10.45

Table 1: Experimental results for four environments tested. (Legend: R = Rooms, C = Corners, D = Doorways)

The reconstructed floor plans for each environment are shown in Figure 7, along with manually measured ground truth. Some reconstruction errors can be observed in environments (A) and (D), where an individual room has been misaligned. In environment (C) our system approximates a bay window to be flat, in accordance with the Manhattan-world assumption. We calculate a measure of reconstruction error from the overlay, as the ratio of area incorrectly reconstructed (sum of both overestimated and underestimated areas) to the total ground truth area. The results indicate that our system is able to reconstruct the dimensions and overall floor plan of an indoor scene with an average error of 10.45% for the environments tested. Aside from the previously mentioned failure cases, the floor plan correctly captures the shape, relative position and orientation of most rooms. This meets our goal of capturing an accurate layout that is useful for the purposes of visualization and navigation in the interactive tour.

CONCLUSION AND FUTURE WORK

We presented a novel end-to-end system to capture and reconstruct indoor scenes for the purpose of creating an interactive visual tour. While there is a large body of prior work in this area, our system is the first to allow casual users to quickly and easily create immersive tours on a smartphone. Our method also enables the creation of indoor floor plans and 3D renderings of the scene, without the need to remove furniture or physically measure the environment. In order to achieve our design goals, we have devised a data capture framework, an interactive photogrammetric modeling scheme and an indoor floor plan generation algorithm, designed to run independently on a commodity smartphone in real-time.

While the floor plan generation algorithm is currently unable to model curved walls or non right-angled corners, we believe that the system is sufficient to capture a qualitative sense of most indoor environments.

Several aspects of this system may be improved upon in future work. A hand-held device is susceptible to shake and the resulting imagery is sometimes motion blurred. The gyroscope data from the non-major axes can be used to select frames with minimal shake, thus stabilizing the image data. Another improvement would be to better model complex indoor scenes that contain non-Manhattan world elements. One approach would be to use computer vision techniques to estimate the depth of room features to assist the floor plan generation algorithm. The creation of more detailed 3D renderings of the scene can be enabled by adding textures to the walls and modeling furniture and other items within a room. Finally, to enhance the effect of immersion, it should be possible to capture aspects of physical interaction in the scene, for example operating appliances or turning on/off light switches. These can be captured with our video-based approach but require explicit modeling to incorporate them into the interactive tour.

We believe that real estate is perhaps the most compelling application for our system. Realtors and homeowners can use our smartphone application to generate tours and floor plans of houses without the need for any extra equipment. Mapping is another compelling application area. Commercial map-

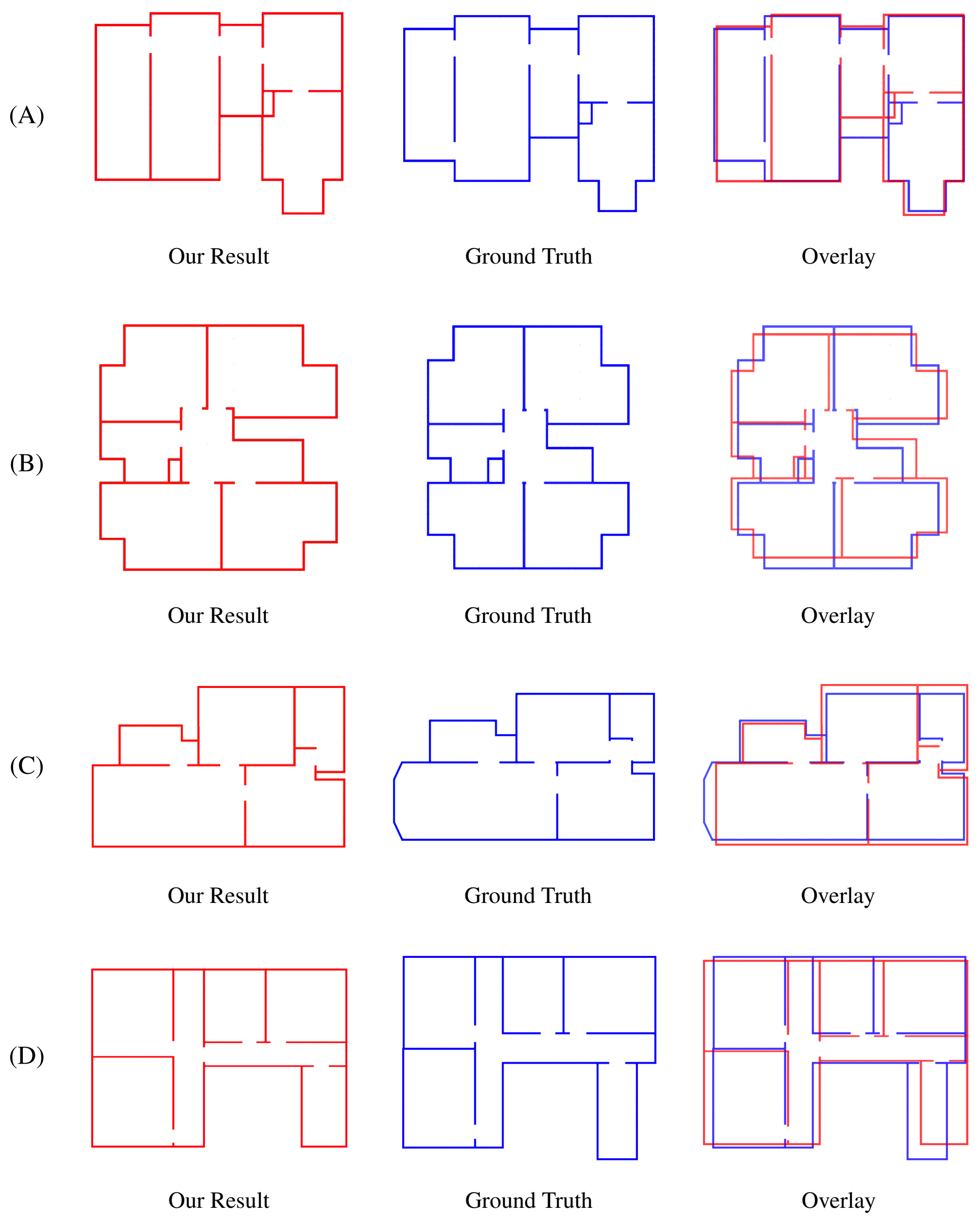


Figure 7: Side-by-side comparisons of floor plans generated by our system to ground truth

ping services such as StreetView [1] and Bing Maps [13] are increasingly incorporating indoor maps into their products. Small-business owners or other operators can use our system to create immersive tours of their establishments (cafés, restaurants, hotels, etc.) and share them with customers through such mapping services. Aspects of our approach may also be useful for other applications, in particular, the floor plans can be used to bootstrap systems for indoor localization (e.g., in a mall) and mobile robot navigation. Finally, the interactive tours give a sense of virtual presence, allowing individual users to share their experience of a space with friends and family.

With almost half (49.7% [22]) of U.S. mobile subscribers now owning smartphones and similar rising trends across the world, we believe that our method will enable casual users to capture, visualize and reconstruct of a wide variety of interesting indoor scenes, from homes and offices to museums and heritage sites.

ACKNOWLEDGMENTS

This work was supported in part by NSF grants IIS-0811878 and IIS-0963657, SPAWAR, the University of Washington Animation Research Labs, and Google.

REFERENCES

1. Anguelov, D., Dulong, C., Filip, D., Frueh, C., Lafon, S., Lyon, R., Ogale, A., Vincent, L. and Weaver, J. Google Street View: Capturing the World at Street Level. In *Computer*, 43(6): pages 32–38, June 2010.
2. Brooks, F. Walkthrough – a dynamic graphics system for simulating virtual buildings. In *Proceedings of I3D'86*, pages 9–21, 1987.
3. Capturing Indoor Scenes with Smarthphones, project page. `http://grail.cs.washington.edu/videotour/`
4. Coorg, S. and Teller, S., Extracting textured vertical facades from controlled close-range imagery. In *Proceedings of CVPR*, pages 625–632, 1999.
5. Coughlan, J.M. and Yuille, A.L. Manhattan World: Compass Direction from a Single Image by Bayesian Inference. In *Proceedings of ICCV*, pages 941–947, 1999.
6. Flint, A., Murray, D. and Reid, I. Manhattan scene understanding using monocular, stereo, and 3D features. In *Proceedings of ICCV*, pages 2228–2235, 2011.
7. Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. Reconstructing Building Interiors from Images. In *Proceedings of ICCV '09*, pages 80-87, 2009.
8. Goldman, D. B., Curless, B., Salesin, D., and Seitz, S. M. Schematic storyboarding for video visualization and editing. In *Proceedings of SIGGRAPH*, pages 862–871, 2006.
9. Google Inc., Google Art Project, 2011. `http://www.googleartproject.com`
10. Ishiguro, H., Ueda, K. and Tsuji, S. Omnidirectional visual information for navigating a mobile robot. In *Proceedings of ICRA*, pages 799–804, 1993.
11. Kim, Y.M., Dolson, J., Sokolsky, M., Koltun, V., Thrun, S., Interactive Acquisition of Residential Floor Plans. In *Proceedings of ICRA*, pages 3055–3062, 2012
12. Lippman, A. Movie-maps: An application of the optical videodisc to computer graphics. In *Proceedings of SIGGRAPH*, pages 32–42, 1980.
13. Microsoft Bing Maps, 2012. `http://www.bing.com/maps/`
14. Microsoft Corporation, Photosynth, 2011. `http://photosynth.net/`
15. Quiksee, 2011. `http://www.quiksee.com/`
16. Reitmayr, G. and Drummond, T. Going out: robust model-based tracking for outdoor augmented reality. In *Proceedings of ISMAR*, pages 109–118, 2006.
17. Sensopia Inc., MagicPlan, 2011. `http://www.sensopia.com`
18. Shin, H., Chon, Y., Cha, H., Unsupervised Construction of an Indoor Floor Plan Using a Smartphone, In *IEEE Transactions on Systems, Man, and Cybernetics*, Volume PP, Issue 99, pages 1–10, 2011
19. Snavely, N., Seitz, S. M., and Szeliski, R., Photo tourism: exploring photo collections in 3D. In *Proceedings of SIGGRAPH*, pages 835–846, 2006.
20. Szeliski, R. and Shum, H. Creating full view panoramic image mosaics and environment maps. In *Proceedings of SIGGRAPH*, pages 251–258, 1997.
21. Taylor, C. J. VideoPlus: A Method for Capturing the Structure and Appearance of Immersive Environments. In *IEEE Transactions on Visualization and Computer Graphics*, 8(2):pages 171–182, 2002.
22. The Neilsen Company: Neilsen Mobile Insights, 2012. `http://bit.ly/H0J5XU`
23. Tsang, M., Fitzmzurice, G. W., Kurtenbach, G., Khan, A. and Buxton, B. Boom chameleon: simultaneous capture of 3D viewpoint, voice and gesture annotations on a spatially-aware display. In *Proceedings of SIGGRAPH*, pages 698–698, 2003.
24. Uyttendaele, M., Criminisi, A., Kang, S. B., Winder, S., Szeliski, R., and Hartley, R. Image-Based Interactive Exploration of Real-World Environments. In *IEEE Computer Graphics and Applications*, 24:pages 52–63, 2004.
25. Yagi, Y., Kawato, S. and Tsuji, S. Real-time omnidirectional image sensor (COPIS) for vision-guided navigation. In *IEEE Transactions on Robotics and Automation*, 10(1):pages 11–22, 1994.
26. Zandbergen, P. A. Accuracy of iPhone Locations: A Comparison of Assisted GPS, WiFi and Cellular Positioning. In *Transactions in GIS*, 13:pages 5–25, 2009.