Interactive mapping of indoor building structures through mobile devices

Giovanni Pintore*, Marco Agus† and Enrico Gobbetti‡

CRS4 Visual Computing

Polaris Ed. 1, 09010 Pula, CA, Italy

*Email: *giovanni.pintore@crs4.it, †marco.agus@crs4.it, ‡enrico.gobbetti@crs4.it*

Abstract—We present a practical system to map and reconstruct multi-room indoor structures using the sensors commonly available in commodity smartphones. Our approach combines and extends state-of-the-art results to automatically generate floor plans scaled to real-world metric dimensions and to reconstruct scenes not necessarily limited to the *Manhattan World* assumption. In contrast to previous works, our method introduces an interactive method based on statistical indicators for refining wall orientations and a specialized merging algorithm for building the final rooms shape. The low CPU cost of the method makes it possible to support full execution by commodity smartphones, without the need of connecting them to a compute server. We demonstrate the effectiveness of our technique on a variety of multi-room indoor scenes, achieving remarkably better results than previous approaches.

Keywords-Computer aided engineering; Mobile computing; Application software

I. INTRODUCTION

Acquiring and reconstructing the indoor built environment is an important, but challenging, task in many real world applications. Original blueprints are often hard to find, especially for older buildings, and even where they exist, they often do not represent the current layout. Recovering the built structure often requires onerous manual modeling sessions. Even though automated approaches exist for the reconstruction of urban outdoor environments [3], [16], [22], [29], and user-assisted methods have been successfully applied for facade modeling [12], [21], [25], the reconstruction of interiors is complicated by a number of factors. For instance, visibility reasoning is more problematic since a floor plan may contain several interconnected rooms. Moreover, interiors have usually texture-poor walls that complicate image-based processing. Current high-performance approaches, which range from 3D laser scanning [13] to image-based methods [2], [18], produce high resolution 3D models which are often an overkill for a large branch of applications, especially those focused on the structure of a building rather than the details of the model. In practice, contractors and interior designers use point-to-point laser measurement devices to acquire a set of distance measurements. Based on these measurements, CAD experts create a floor plan that respects these measurements and represents the layout of the building. There is now a growing interest in semi-automatic methods that simplify the scene capture, making it possible for casual users, and eliminate the manual post-processing necessary for reconstructing the layout of walls, making the process faster and cost-effective. The use of modern mobile devices to create a 3D map of an indoor environment is a growing and promising approach, as highlighted by the recent presentation of *Google Project Tango* [9]. In this context, we propose a method to enable any user to reconstruct building interiors with the aid of a mobile phone and without requiring the assistance of computer experts, 3D modelers, or CAD operators. This kind of multi-room mapping is useful in many real-world applications, such as in the field of security management and building protection, to enable non-technical people to create models with enough geometric features for simulations or enough information to support interactive visual tours [5], [17].

Approach. Similarly to other recently introduced methods [15], [17], the user captures the scene by walking in the indoor environment, acquiring 360 degrees videos of each visited rooms. These videos are captured by targeting with the phone camera the intersection of the wall with the floor or with the ceiling. In this process, we store together with the video frames, all the data coming from the inertial measurement unit (IMU). Unlike previous work [15], [17], we don't calculate the final room shape only from this initial data, but we exploit it to automatically build a coarse reconstruction of each room, estimating the goodness of fit of each wall through statistical indicators. We make then available to the user interface this coarse room representation (Fig. 1 center), highlighting potential unreliable wall estimations and asking the user to acquire a specific orientation measurement for them (Fig. 2). Once the interactive capture is completed, we automatically combine the whole information through a specialized method, trying to fit the best room shape that combines constrained corner angles with the wall intersections. As a result, we obtain rooms represented in metric coordinates with respect to each room's center. When the user moves through the rooms, a graph of the interconnections between doors and rooms is maintained. Once all room shapes are calculated, we automatically assemble the floor plan exploiting the graph information (Fig. 1 right).

Contributions. Our approach combines and extends state-of-the-art results in order to support the acquisition on an indoor environment using low-end mobile devices and

Figure 1. System pipeline. Left: Capturing of an indoor environment through a mobile device. Center: A coarse reconstruction of the room shape is available to the user interface immediately, highlighting potential unreliable wall estimations (red lines). Right: as a final result a floor plan scaled to real-word metric dimensions is returned.

reduces user interaction and editing time when building multi-room models in a non-Manhattan world. We improve over recent work on reconstruction of multi-room models in a non-Manhattan world [15] by introducing a method for exploiting the strong redundancy between samples and images to evaluate the goodness of fit of each segment representing a wall, completing the initial data with additional orientation measurements in the case of uncertain results. We also proposed an improved merging algorithm to recover the definitive shape through the minimization of a custom distance metric, achieving better results than in previous approaches (see Sec. II).

Advantages and limitations. Our approach allows casual users to rapidly acquire complex multi-room models in a non-Manhattan world using only commodity devices. The output of the system returns room shapes scaled to real-world metric units and employs a fully automatic floor merging step. Compared to similar approaches our method overcomes some of their limits, being more effective in corridors, big rooms and generally reducing the inevitable human error associated with this type of measurement (Sec. V). The method supports walls that do not intersect at right angles, but, due the particular assumptions of the acquisition model, the method doesn't recover curved walls, sloped ceiling and structures that don't belong to the perimeter of the rooms.

II. RELATED WORK

Many approaches have been proposed to automatically recover the architecture of an indoor scene, ranging from 3D laser scanning to image-based methods. Devices such as laser scanners represent often the most effective but also expensive solution for a fast and accurate acquisition [27], for this reason their use is often reserved to specific applications such as in cultural heritage field. Starting from the output of these devices the reconstruction methods depend on the purpose, sometimes creating visually realistic models [6], [7], [26], architectural floor plans [24] or semantic 3D models [28]. Even though some of these 3D reconstruction algorithms extract planar patches from

data [1], [13], [23], these usually have the goal of finding simplified representations of the models, rather than identifying rooms, walls, ceilings, and floors. On the other hand many 3D reconstruction methods based on images have surfaced, whose accuracy compares to laser range sensor systems at a fraction of the cost [18]. Furukawa et al. [8] reconstruct the 3D structure of moderately cluttered interiors by fusing multiple depth maps (created from images) using the heavily constrained *Manhattan World* [4] assumption, through the solution of a volumetric Markov Random Field. These image-based methods have demonstrated potential, but they tend to be computationally expensive and do not work well on texture-poor surfaces such as painted walls, which dominate interiors. Recently Cabral et al. [2] extend the work of Furukawa et al. [8] by extracting depth cues also from the single images and considering the floor plan reconstruction as a shortest path problem on a specially crafted graph, enforcing piecewise planarity and managing simple cases of *Non Manhattan World* scenes. SLAM-based reconstruction has been shown to work on smartphones by Shin et al. [20], who used sensor data to model indoor environments. Although limited to short-range indoor scanning, Kim et al. [10] showed how to acquire indoor floor plans in real-time, but under the constraints imposed by the Manhattan World assumption. 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 presentation of *Google Project Tango* [9] confirms there is a growing interest in mobile devices to recover 3D indoor scenes in an easy and cheap way, with the goal of enabling non-technical users to obtain a quick even if approximative reconstruction of an indoor environment, just exploiting commodity devices. In this field commercial solutions as *MagicPlan* [19] reconstruct floor plans by marking floor corners visible from the room's center via an augmented reality interface. This approach manages also *non-Manhattan world* scenes, but requires manual editing for room assembly and is susceptible to error when floor corners are occluded by furniture, requiring the

user to guess their positions. Systems as *Locometric* [11] acquire the walls orientation by placing the smartphone on the wall surface and tracking approximatively the user position with a step tracker. The captured data is then exploited to support a sketch-based framework to manual edit and scale the floor plan. With an acquisition approach similar to *MagicPlan* [19], Sankar et al. [17] reconstruct individual room shapes geometrically calculated using only the horizontal heading of the observer, assuming they must be *Manhattan-world* shapes. The resulting rooms have arbitrary dimensions and need manual scaling and manual identification of correspondences between doors before the floor plan can be assembled. Exploiting the redundancy between the instruments available on modern mobile devices Pintore et al. [15] obtain indoor building layouts non necessary limited to *non-Manhattan world* assumptions and scaled to metric dimensions. This approach has demonstrated to be potential but lacks in accuracy in several environments as corridors, big halls and generally when rooms have far or dark corners. We enhance and extend this approach by improving both the capture and reconstruction subsystems, obtaining more accurate results (see Sec. V), and supporting the reconstruction of environments where usually previous methods fail, as big halls with dark corners, very narrow rooms or corridors.

III. OVERVIEW

The method is implemented as a single mobile application and it can be summarized in two blocks: **interactive scene capture** and **automatic scene processing**.

A. Interactive scene capture

We capture the scene by walking between rooms, ideally drawing the wall upper or lower boundary aiming the phone camera at it. During the acquisition a video of the environment is recorded and every frame spatially indexed using the phone's sensors data, storing angular measures coupled with images for every room. Exploiting this initial information a coarse 2D layout of the room is displayed on the device screen, highlighting potential wrong estimations of the walls position, according to statistical indicators (see Fig. 1 center). An additional measure procedure is then performed for the unreliable walls, as illustrated in IV-B. When the single room acquisition is completed the user moves to the next one aiming the phone camera to the exit door while the application tracks his direction and updates a graph with the interconnections between the rooms.

B. Automatic scene processing

We consider a 2D model of the floor plan where the center of each room is the observer position, the walls coordinates depend on the azimuth θ (defining the heading of the current target point, Eq. 1) and d is the distance between the wall and the observer (see Fig. 2 left). We calculate d from the angle γ according to eq. 2 or, alternatively, we can obtain this measurement directly from the mobile device if the feature is supported (for instance with the *Google Tango* device). By combining the acquired points and the associated image data we estimate a 2D line for each wall segment, obtaining an initial room representation. Due to the model approximation and measurement error (mainly human error during the acquisition) the shape produced is still a coarse representation of the room. To improve it we use a merging algorithm, introducing the corrected orientation measurements stored at acquisition time and iteratively refining the scale of the corner positions, obtaining as result a room shape already represented in real world metric coordinates, without restricting the problem space to the *Manhattan World* assumption. Exploiting the interconnection graph generated during the scene capture we calculate through doors matching all the transformations to generate the whole floor plan, assuming as origin of the coordinates system the room with the best fit values (see Sec. IV-B).

IV. SYSTEM COMPONENTS

A. Scene acquisition

To capture the scene we adopt a model based on [15], illustrated in Fig. 2 (left and center) and summarized as follows. The first goal of this method is to obtain a set of boundary points between the wall and the ceiling, from which the wall position and the corner points can be estimated. We assume that the observer's position is the origin of room's coordinates and each targeted point p is represented with *metric Cartesian coordinates* as

$$p(x, y) = (d * \cos\theta, d * \sin\theta) \tag{1}$$

where the angle θ is the heading of the targeted point with respect to the magnetic North and d the distance from the observer to the wall. Assuming the height of the eye h_e and the height of the wall h_w are constant and known in metric units, the distance d is estimated from the angle γ through the equation

$$d = \begin{cases} (h_e / \sin\gamma_f) * \cos\gamma_f \\ ((h_w - h_e) / \sin\gamma_c) * \cos\gamma_c \end{cases} \tag{2}$$

where γ_f is the tilt angle if the observer is aiming to the intersection between wall and floor and γ_c is the tilt angle if the observer is aiming to the intersection between wall and ceiling (Fig. 2 center). Alternatively, since next generation mobile devices are beginning to integrate depth sensors exploiting computer vision techniques or dual camera technologies to estimate the focus distance (e.g., *Google Tango* [9], as well as some new smartphones such as the HTC One M8 and other Android 4.4 devices), the value d could be obtained directly without the need of Eq. 2. However, at the time of this writing devices with depth
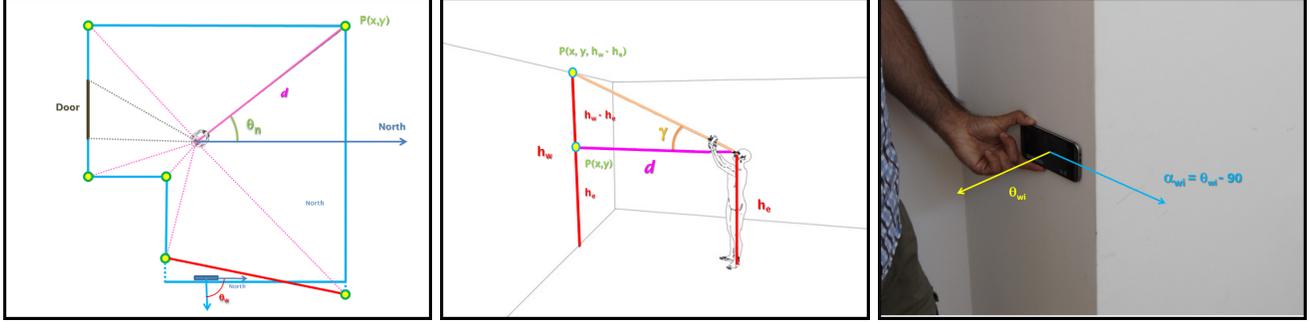
Figure 2. Left: top view of the acquisition model. When a wall is marked as *unreliable* (red line) we perform a further acquisition step to estimate its orientation θ_w in respect to the Magnetic North. Center: side view of the acquisition model with $\gamma = \gamma_c$. As we can evince from eq. 2 the points can be acquired indifferently aiming at the floor or at the ceiling, overcoming a typical problem of previous systems such as *MagicPlan* [19] which are prone to considerable errors when corners on the floor are occluded. Right: unreliable walls are fixed through the easy procedure described in IV-A.

measurement capabilities are not on the market yet; further, currently available devices return uncalibrated depth values at short range (maximum 2 meters) and without a real spatial reference. To acquire a room the observer rotates 360 degrees, following an ideal trajectory with the phone camera with the goal of keeping boundary of the wall (upper or lower) on the target in the middle of the screen (see Fig. 1 left). During this phase we automatically store a set of *samples* for each wall containing the angles θ, γ, ρ, respectively the azimuth, tilt and roll identifying the targeted point at each instant within the illustrated model, and a time index expressed in milliseconds identifying the corresponding frame in the video sequence. Once a room corner or a door extremity is reached the user records the event with a click on the screen (for doors, one click for each extremity) storing its azimuth θ, then he proceeds acquiring the next wall.

Unlike [15], after the user completes the entire perimeter of the room a coarse reconstruction of the acquired layout is displayed by the application interface (see Fig. 1 center) highlighting potentially wrong wall representations (see Sec. IV-B). The user is then invited to acquire a measurement of the wall orientation θ_{wi} in respect to the magnetic North by placing the smartphone on its surface (see Fig. 2 right). Considering that the x axis in our room coordinates system is also the heading of the magnetic North, for each wrong wall segment $\widetilde{W}_i^*$ we store an additional angle

$$\alpha_{wi} = \theta_{wi} - 90 \qquad (3)$$

expressed in degrees and representing the new orientation of the wall with respect to the room coordinate system. This information is stored together with the other measurements and used for the final reconstruction (see Sec. IV-C). Once the room is completely acquired the user moves to the next one aiming the phone camera to the exit door. The passage to the next room is automatically identified by tracking the direction of the movement. This information is then stored in a graph of rooms interconnections (see Sec. IV-D).

B. Coarse room reconstruction

We assume our room model is a closed polygon bounded by a set of segments (walls) $\{\widetilde{W}_0 \cdots \widetilde{W}_n\}$. Since we acquired the wall boundaries by imposing a linear trajectory, we expect the slope and position of each wall $\widetilde{W}_i$ to be identified by the 2D line that best fits the *samples* $S_i = \{s_0 \cdots s_m\}$, which were acquired between two corner angles θ_{i-1} and θ_i (order of a thousand samples per wall). Each sample s_k is identified by its metric Cartesian coordinates according to Eq. 1 and by a time index t_k identifying its associated image inside the video sequence. As demonstrated in previous work [15], the exploitation of the additional information provided by the image data considerably improves the quality of the reconstruction. To integrate this information in our system we adopt a weighted linear regression method, calculating for each sample s_k its weight w_k by considering the associated frame f_k at the instant t_k as proposed in [15].

Once we have calculated the weights a linear equation

$$y(x) = a + bx \qquad (4)$$

is recovered by the minimization of the following quantity:

$$\chi^2 = \sum_{i=0}^{m} w_i (y_i - a - bx_i)^2 \qquad (5)$$

The Cartesian coordinates of the samples $s_i = (x_i, y_i)$ are calculated from Eq. 1 and depend only on the inertial measurement unit (IMU), while the weights w_i are computed from the edges detected in the associated images f_i. Since the residuals of the trajectory follow a normal distribution, the goodness of fit can be computed by employing the standard coefficient of determination R^2 [14], defined as

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \qquad (6)$$

where $SS_{res} = \sum_{i=0}^{m} w_i (y_i - a - bx_i)^2$ is the weighted sum of residuals, while $SS_{tot} = \sum_{i=0}^{m} w_i (y_i - \bar{y})$ is the weighted

variance. According to this definition, values of R^2 close to 1 indicate that a perfect linear fit has been recovered for a given wall. Given the estimated walls $\{\widetilde{W}_0 \cdots \widetilde{W}_n\}$, their intersections $\{\tilde{p}_0 \cdots \tilde{p}_n\}$ identify the room's corners. Each wall is marked as *reliable* or *unreliable* according to the following criteria:

$$\widetilde{W}_i \begin{cases} reliable & R^2(\widetilde{w}_i) > 0.95 \\ unreliable & R^2(\widetilde{w}_i) < 0.95 \end{cases} \quad (7)$$

If a wall is marked as *unreliable*, we follow the additional measurement step illustrated in Sec. IV-A and define a new corrected wall W_i by the line with direction α_{wi} (Eq. 3) and passing through the midpoint between $\widetilde{p}_i$ and $\widetilde{p}_{i+1}$. On the other hand, for reliable walls $W_i \equiv \widetilde{W}_i$.
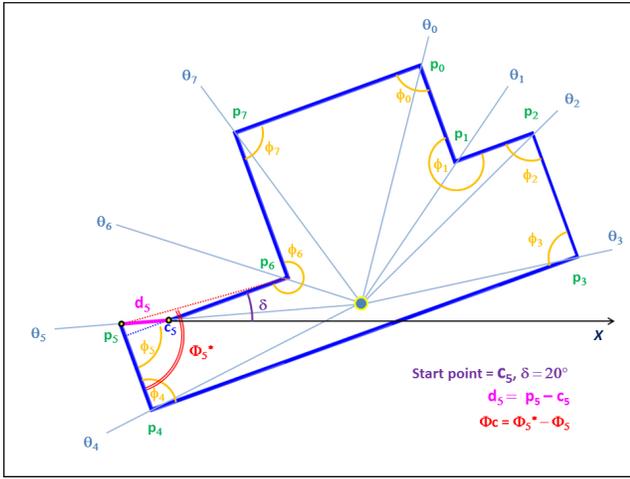

Figure 3. *Room shape optimization* An example with the error intentionally increased: the polygon $P(\delta, c_5)$ starting from the corner c_5 along the angle δ 20 degrees. The last intersection results in the point p_5 and an associated angle ϕ_5. In the ideal case the last intersection p_i should almost coincide with a starting point c_i; likewise, the angle ϕ_5 should coincide with the angle ϕ_5^*.

C. Room shape refinement

Once we have calculated the set of walls $\{W_0 \cdots W_n\}$, we use their intersection points $\{p_0 \cdots p_n\}$ to calculate their relative corner angles $\{\phi_0 \cdots \phi_n\}$ (see Fig. 3). Since we require that the system run almost in real-time on mobile devices with limited resources we decided to employ a fast custom method to recover the definitive shape of the room. We consider all the possible closed polygons $P(\delta, c_i)$, starting from each c_i corner (see Sec. IV-B) and having δ as starting angle, with δ varying between 0 and 360 degrees. Every polygon $P(\delta, c_i)$ must have the angles $\{\phi_0 \cdots \phi_n\}$ as internal constant angles; further, its corners must be on the points of intersection with the constant rays $ray(origin, \theta_i)$ – going from the room's origin along the heading θ_i to the point p_i. We iterate for each $c_i \in \{c_0 \cdots c_n\}$ and δ between 0 and 360 degrees (i.e. 0.2 degree steps), tracing

the polygons intersecting the rays $ray(origin, \theta_i)$ with segments along directions $\delta + \phi$ (see example in Fig. 3), searching for the polygon that minimizes the quantity

$$d(d_c, \phi_c) = (1 - \lambda)d_c + \lambda \cdot \phi_c \quad (8)$$

where: $d_c = \|p_i - c_i\|$ is the distance between the current starting point c_i and its estimation after the last segment's intersection p_i (in the ideal case it should coincide with the starting point); ϕ_c is the difference between the angle ϕ_i at the starting point c_i and its estimation ϕ_i^* after closing the candidate polygon; λ a weight factor balancing scale error (each c_i imposes its scale to the polygon) and angular error (ϕ_i at the starting corner i ideally must close the room shape). In our case we use $\lambda = 0.25$ (more weight on the scale error).
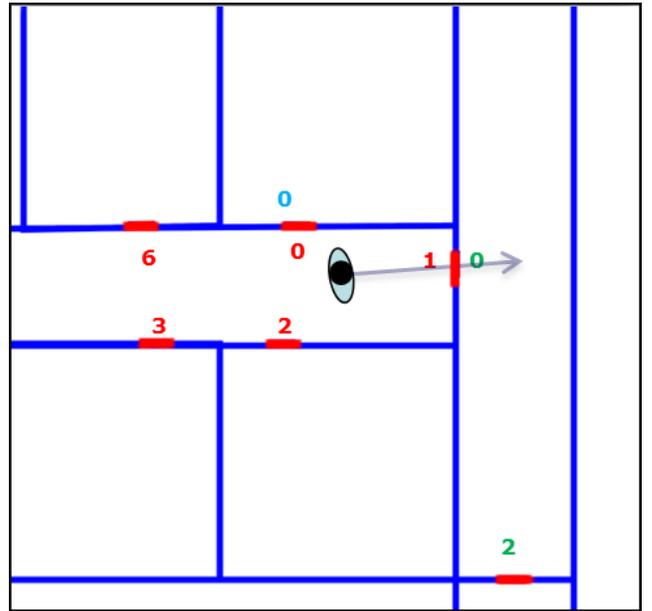
D. Floor plan generation

Figure 4. *Floor plan generation* We choose the starting room r_0 with the best goodness of fit value (Eq. 6). We then align each other room to r_0, calculating the path to reach the starting room as a set of transforms representing the passages encountered while moving from the aligned room to $\mathbf{r_0}$.

For each connection (Fig. 4) stored in the acquisition graph (see Sec. IV-A) we calculate the affine transform $M_{j,j+1}$ representing the transform from the coordinates of room r_{j+1} to room r_j. A connection between two room is defined as a pair of doors: the exit door from the previous room (identified by tracking the movement direction) and the entrance door to the next room (by convention, the first one acquired). The user may happen to visit the same room more than once (e.g., a corridor); for these cases, we provided the acquisition application with an interface to manually update the room id. Since a connection between

Table I

COMPARISON BETWEEN THE RECONSTRUCTION BASED ON SAMPLES AND IMAGES ONLY AND OUR FINAL RECONSTRUCTION.

	Scene		Samples and images only error			Our method error			
Type	Area mq	Rooms	Area %	Corner angle max	Corner pos max	Area %	Corner angle max	Corner pos max	Unrel. walls
Hotel F11	515	7	5.52	7.5 deg	85 cm	1.9	0.8 deg	20 cm	8
Residential 1	90	6	2.75	5.0 deg	22 cm	1.82	0.9 deg	8 cm	1
Residential 2	136	8	5.12	7.0 deg	27 cm	2.10	0.8 deg	12 cm	4
Office Block H	560	27	2.25	2.1 deg	36 cm	1.75	1.1 deg	15 cm	7
Office Block B	326	15	1.62	1.2 deg	42 cm	1.02	0.5 deg	10 cm	4

two rooms is defined as a pair of doors that are really the same door expressed in different coordinates, we obtain $M_{j,j+1}$ by applying a standard least squares method to the corresponding door extremities. After selecting the starting room r_0 – chosen for having the best goodness of fit value (Eq. 6) – we align the other rooms to it. For each aligned room we calculate the path to $\mathbf{r_0}$ as a set of transforms representing the passages encountered to reach $\mathbf{r_0}$ and the whole transformation to the origin room coordinates. The end result of the entire procedure is a floor plan that is automatically aligned and scaled, without manual editing or intervention.

V. RESULTS

We implemented the method as an Android application (2.2 or higher compatible). To capture the scenes we employed an HTC One M8, with Quad-core 2.3 GHz CPU, Qualcomm MSM8974AB Snapdragon 801 chipset, 2 GB RAM, Adreno 330 GPU, 4 Mpixels dual camera. Even though the HTC One M8 has innovative capabilities respect to other commodity devices (i.e. dual camera, focus distance estimation, etc.) we intentionally prefer to use only its standard features, currently available also on low-end devices, such as accelerometer, magnetometer, gyroscope and single camera. We present results for 5 building floors, distributed between commercial and residential buildings (see Fig. 5 and Table I), with both *Manhattan World* and *Non-Manhattan World* scenes. We consider as ground truth the provided blue prints and a manual measuring and modeling session when they are not available, assuming the error illustrated in Tab. I respect to these measures (actually the true measures do not always coincide with the building layout supplied by the architects. For the purpose of this work the difference is negligible). We show in Tab. I a comparison between the reconstruction based on an implementation of [15] exploiting the samples and the images only (light blue) and our method final result (blue). The area error is calculated as the ratio of area incorrectly reconstructed to the total ground truth area, the corners position maximum error is the maximum depth error observed for each corner, whereas the corner angle error is relative to the final result for each room. In contrast to previous methods where uncertain intersections between walls are approximated as *Manhattan World* angles (90 degrees), our method never try to ap-

proximate the corner angles, overcoming the typical limits of previous methods in presence of corridors, large halls with dark corners, etc. We notice that a large part of the error in the coarse reconstruction is due to the user handle of the device. This error is fixed partially by the images feedback and manly by the angle constraints imposed in the final room shape reconstruction. Moreover since fixing a wall orientation affects the coordinates of its extremities, it adjusts also their heading θ (see model Fig. 2), correcting the error that inevitably appears when a corner is manually marked on the screen (see IV-A). Otherwise in previous approaches as for example [17] this kind of error is not managed, thus affecting the quality of the reconstruction.

As for other methods sloped ceilings, curved walls, structures that don't belong to perimeter of the rooms are not handled by our method as illustrated in Fig. 5, where the *room3* of the conference hall isn't been reconstructed due to a curved glass wall.

VI. CONCLUSION

We presented a mobile system to capture a built indoor environment and automatically generate floor plans scaled to their metric dimensions, without the need of heavy *Manhattan World* constraints. Without specialized training or equipment, our system can produce a 2D floor plan (and eventually an extruded 3D model) just exploiting the redundancy of the instruments commonly available on commodity smartphones. Our solution extends previous state-of-the-art approaches to achieve more accurate results and to overcome some of their limits. The results obtained can be useful in many real-world applications, as for example those focused on the structure of a building rather than the details of the model, or as support for an interactive image-based navigation. Since the trend for the future is the integration of even more instruments on mobile devices, such a kind of system can be extended to capture and manage more complex features, allowing any user to real-time capture, render and elaborate a 3D indoor environment just exploiting commodity tools as smartphone and tablet.

References

[1] M. Arikan, M. Schwärzler, S. Flöry, M. Wimmer, and S. Maierhofer. O-snap: Optimization-based snapping for modeling architecture. *ACM Trans. Graph.*, 32(1):6:1–6:15, Feb. 2013.

[2] R. Cabral and Y. Furukawa. Piecewise planar and compact floorplan reconstruction from images. *The IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[3] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool. 3d urban scene modeling integrating recognition and reconstruction. *Int. J. Comput. Vision*, 78(2-3):121–141, 2008.

[4] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proc. ICCV*, volume 2, pages 941–947, 1999.

[5] M. Di Benedetto, F. Ganovelli, M. Balsa Rodriguez, A. Jaspe Villanueva, R. Scopigno, and E. Gobbetti. Exploremaps: Efficient construction and ubiquitous exploration of panoramic view graphs of complex 3d environments. *Computer Graphics Forum*, 33(2), 2014. Proc. Eurographics 2014.

[6] S. F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin. System for indoor 3D mapping and virtual environments. In S. F. El-Hakim, editor, *Videometrics V*, volume 3174 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 21–35, July 1997.

[7] C. Frueh, S. Jain, and A. Zakhor. Data processing algorithms for generating textured 3d building facade meshes from laser scans and camera images. *International Journal of Computer Vision*, 61(2):159–184, 2005.

[8] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Reconstructing building interiors from images. In *Proc. ICCV*, 2009.

[9] Google inc. Google project tango, 2014.

[10] Y. M. Kim, J. Dolson, M. Sokolsky, V. Koltun, and S. Thrun. Interactive acquisition of residential floor plans. In *Proc. IEEE ICRA*, pages 3055–3062, 2012.

[11] Locometric. Locometric, 2014.

[12] P. Müller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. In *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, pages 614–623, New York, NY, USA, 2006. ACM.

[13] C. Mura, O. Mattausch, A. J. E. Gobbetti, and R. Pajarola. Automatic room detection and reconstruction in cluttered indoor environments with complex room layouts. *Computers & Graphics*, 2014.

[14] N. J. Nagelkerke. A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692, 1991.

[15] G. Pintore and E. Gobbetti. Effective mobile mapping of multi-room indoor structures. *The Visual Computer*, 30, 2014. Proc. CGI 2014.

[16] M. Pollefeys and et al. Detailed real-time urban 3d reconstruction from video. *Int. J. Comput. Vision*, 78(2-3):143–167, 2008.

[17] A. Sankar and S. Seitz. Capturing indoor scenes with smartphones. In *Proc. ACM UIST*, pages 403–412, 2012.

[18] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. CVPR*, volume 1, pages 519–528, 2006.

[19] Sensopia. Magicplan, 2011.

[20] H. Shin, Y. Chon, and H. Cha. Unsupervised construction of an indoor floor plan using a smartphone. *IEEE Trans. Systems, Man, and Cybernetics*, 42(6):889–898, 2012.

[21] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3d architectural modeling from unordered photo collections. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pages 159:1–159:10, New York, NY, USA, 2008. ACM.

[22] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3d. *ACM TOG*, 25(3):835–84, 2006.

[23] I. Stamos, G. Yu, G. Wolberg, and S. Zokai. 3d modeling using planar segments and mesh elements. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 599–606, June 2006.

[24] E. Turner and A. Zakhor. Watertight as-built architectural floor plans generated from laser range data. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 316–323, Oct 2012.

[25] J. Xiao, T. Fang, P. Tan, P. Zhao, E. Ofek, and L. Quan. Image-based facade modeling. In *ACM SIGGRAPH Asia 2008 Papers*, SIGGRAPH Asia '08, pages 161:1–161:10, New York, NY, USA, 2008. ACM.

[26] J. Xiao and Y. Furukawa. Reconstructing the world's museums. In *Proceedings of the 12th European Conference on Computer Vision*, ECCV '12, 2012.

[27] X. Xiong, A. Adan, B. Akinci, and D. Huber. Automatic creation of semantically rich 3d building models from laser scanner data. *Automation in Construction*, 31(0):325 – 337, 2013.

[28] X. Xiong and D. Huber. Using context to create semantic 3d models of indoor environments. In *Proceedings of the British Machine Vision Conference*, pages 45.1–45.11. BMVA Press, 2010. doi:10.5244/C.24.45.

[29] L. Zebedin, J. Bauer, K. Karner, and H. Bischof. Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In *Proc. ECCV*, pages 873–886, 2008.
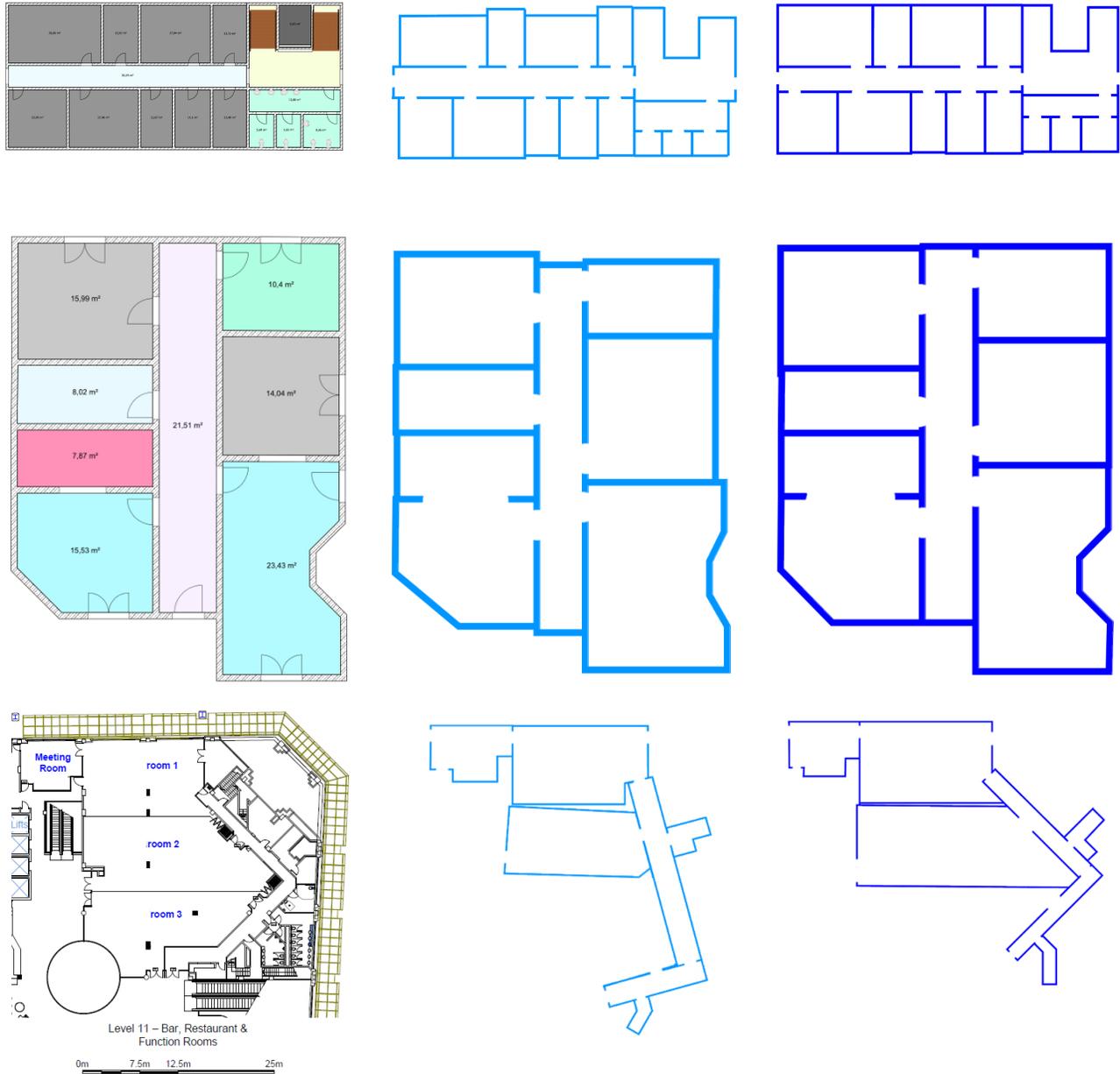
Figure 5. Comparison between blue prints/ground truth (right), a reconstruction based only on IMU measurements and images (light blue - center) and our method (blue - left). Top: office floor with corridors and 15 rooms. Center: Residential building with 8 rooms and a long corridor. Bottom: Hotel conference room. The acquisition covers the rooms accessible to visitors and attenders. We manage the corridor turn splitting it in two separated rooms, considering the split line as a door in order to align correctly the two parts. We intentionally highlight the limits of our method showing as *room3* isn't been reconstructed due to a curved glass wall.