The Visual Computer manuscript No.
(will be inserted by the editor)

Effective mobile mapping of multi-room indoor structures

Giovanni Pintore · Enrico Gobbetti

Received: date / Accepted: date

Abstract We present a system to easily capture building interiors and automatically generate floor plans scaled to their metric dimensions. The proposed approach is able to manage scenes not necessarily limited to the *Manhattan World* assumption, exploiting the redundancy of the instruments commonly available on commodity smartphones, such as accelerometer, magnetometer and camera. Without specialized training or equipment, our system can produce a 2D floor plan and a representative 3D model of the scene accurate enough to be used for simulations and interactive applications.

Keywords Sensors fusion · Mobile graphics · Mobile mapping · Scene analysis · Indoor reconstruction

1 Introduction

Automatic 3D reconstruction of architectural scenes is a challenging problem in both outdoor and indoor environments. Fully automated approaches exist for the reconstruction of urban outdoor environments [26,5,19, 29], furthermore user-assisted methods have long proven effective for facade modeling [7,16,28,25]. Compared to building exteriors, 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, in addition interiors are often dominated by surface that are barely lit or texture-poor walls. Approaches range from 3D laser scanning [17,18] to image-based methods [11,21]. These methods 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. 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* [13]. 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 the definition of thermal zones for energy simulation, or, in the field of security management and building protection, to enable non-technical people to create models with enough geometric features for simulations and enough visual information to support location recognition.

Approach. We capture the scene by walking between rooms, ideally drawing the wall upper or lower boundary aiming the phone camera at it (Fig. 1 left). During the acquisition a video of the environment is recorded and every frame spatially indexed with the phone's sensors data, storing thousands of samples for every room. Through a specialized method based on statistical indicators, we merge all individual samples exploiting their strong redundancy and obtaining the walls direction and position in metric coordinates. With a further refinement pass we calculate the complete shape of the single rooms and the whole aligned floor plan(Fig. 1 center).

Contributions. In contrast to previous work, see Sec. 2, no *Manhattan World* assumption is needed to reconstruct the geometry of the walls, allowing our method to reconstruct rooms with irregular shapes – i.e., with corners that do not form right angles. Moreover, the output of the mathematical model adopted returns both wall

G. Pintore and E. Gobbetti
CRS4 Visual Computing, POLARIS Ed. 1, 09010 Pula, CA, Italy
E-mail: {giovanni.pintore,enrico.gobbetti}@crs4.it
www: http://www.crs4.it/vic/

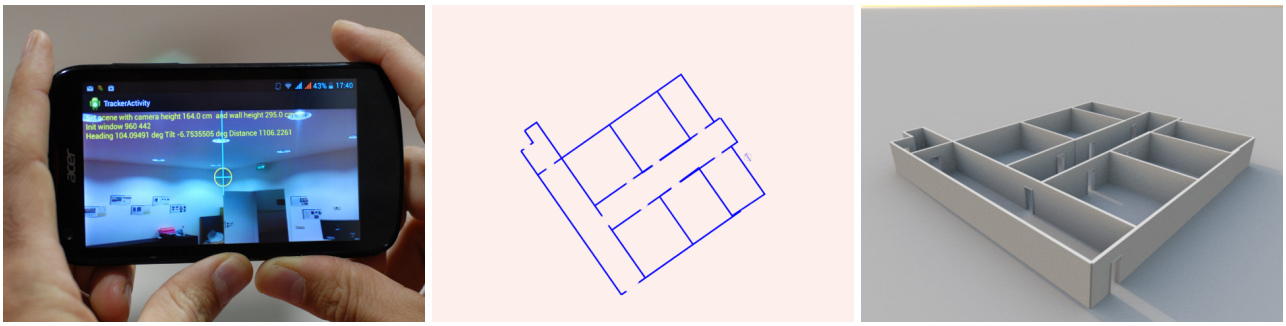

Fig. 1 The system captures an indoor environment through a mobile device, automatically producing a floor plan and a 3D model with its metric dimensions.

directions and corner positions in real world units, returning a room shape ready for the following automatic floor merging step, which is able to compose multi-room models without manual interventions.

Advantages and limitations. Our approach combines and extends state-of-the-art results in order to support acquisition using low-end mobile devices and reduces user interaction and editing time when building multi-room models in a non-Manhattan world. In order to achieve its goals, the systems makes a number of assumptions on the acquired shapes. In particular, the floor plane must be parallel to the ceiling plane, since the wall height is supposed constant during the acquisition of a single room. Moreover, narrow corridors with complex shapes can not be managed properly (see Sec. 5).

2 Related Work

Standard methods for reconstructing building structures usually focus on creating visually realistic models [9,10], rather than structured ones. Even though some of these 3D reconstruction algorithms extract planar patches from data [1,27], these usually have the goal of finding simplified representations of the models, rather than identifying walls, ceilings, and floors. Furukawa et al. [11] reconstruct the 3D structure of moderately cluttered interiors by fusing multiple depth maps (created from images) using the heavily constrained *Manhattan World* [6] 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. SLAM-based reconstruction has been shown to work on smartphones by Shin et al. [24] who used sensor data to model indoor environments, similarly Kim et al. [14] 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. Exploiting modern mobile devices capabilities, commercial solutions as *MagicPlan* [22] 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. With a similar acquisition approach Sankar et al. [20] 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. In contrast to these related methods, our new approach is not limited by the Manhattan World assumption and it can reconstruct the model without manual scaling or manual assembly of the floor plan.

3 Overview

We divide the pipeline of our system in two blocks: **scene capture** and **scene processing**, resulting in two different applications (see Sec. 5 for implementation details). The scene capture application stores the acquired data in a distributed database, whereas the scene processing application remotely accesses this database to reconstruct the environment. In the mobile version (e.g., Android) of the system the two applications are implemented as two *Activity* components which can be both hosted on the same device.

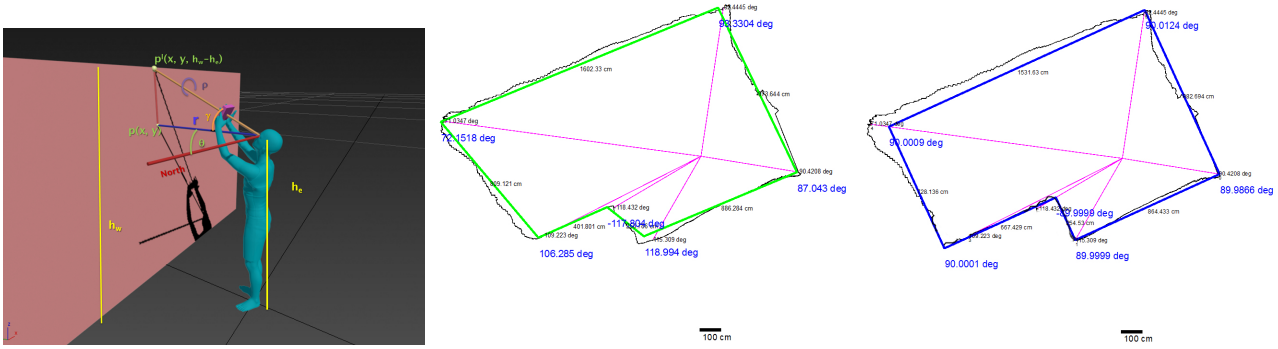

Fig. 2 Left: the adopted model for data capturing. Center: Acquired samples in Cartesian coordinates (black line interpolation). Green lines show the 2D fit of the samples with a standard M-estimator (Huber distance). Right: The walls geometry estimated with our method (blue). Note as the outliers points (upper left corner) have been discarded in the wall computation thanks to their images feedback (see Sec. 4.2).

3.1 Scene capture

Starting approximately from the center of the room, the user acquires a 360 degrees video of the environment, targeting with the phone camera the upper or lower boundaries of the walls, with a movement which traces an ideal trajectory along the intersection of the wall with the floor or with the ceiling. Once a room corner or a door is reached the user records the event with a click on the screen, storing their azimuth angles, then he continues acquiring the following wall, until he/she completes the whole perimeter of the room. For each wall segment delimited by two corner angles we automatically acquire a set of angular coordinates, taken at the maximum rate allowed by the device (order of thousands samples for room), establishing an association with the video frames acquired. When a room acquisition is completed the user moves to the next one aiming the phone camera to the exit door, whereas the application tracks his direction and updates a graph with the interconnections between the rooms.

3.2 Scene processing

From the samples acquired for each wall segment we estimate a 2D line representing the observer's direction and position in real-world metric coordinates. Assuming a model where the coordinates of the wall's upper and lower boundary depend only on the azimuth θ (defining the heading of the current target point, Eq. 1) and the tilt γ (defining the distance of the point from the observer), we observe that these couples of angular coordinates must be linked by a relation when they individuate points on the wall boundaries. Due to model approximation (see Sec. 4) and measurement error, fitting the associated coordinates (eq. 1 and 2) of the samples S_i directly with a conventional method results in an inaccurate estimation (see Sec. 5). In our approach, to achieve a more accurate result we introduce a method exploiting the associated image data and the redundancy between samples, in order to achieve a better linear fitting, thus obtaining a more accurate estimation of the wall. Once all the single wall lines have been estimated in real-world metric coordinates, we build the room shape through a merging algorithm, which considers the wall approximations with their reliability without restricting the problem space to the *Manhattan World* assumption nor requiring camera or scene calibration methods. Exploiting the interconnection graph generated during the scene capture we calculate through doors matching all the transformations to generate the whole floor in real world metric units, assuming as origin of the coordinates system the room with the best fit values (see Sec. 4.2).Besides a 3D model is extruded using the floor plan with the walls height (Fig. 1 right).

4 System components

4.1 Scene acquisition

We acquire the scene according with the model illustrated in Fig. 2 *left*. The height h_e of the eye is assumed constant, considering included in the angle γ all height variations. The angle θ is the heading of the targeted point respect to the magnetic North, and is defined as a rotation around the ideal axis between the Sagittal and Coronal planes of the observer. The angle γ (observer tilt) is the rotation around the axis between the Coronal and Transverse planes, and the angle ρ is the rotation around the intersection between the Sagittal and Transverse planes. We assume that the Transverse

plane of the observer is parallel to the floor and parallel to the ceiling, and the walls are perpendicular to the floor (but not necessarily perpendicular to each other). For the specific world constraints (e.g. floor or ceiling visibility, etc.) refer to Sec. 5. Assuming that the observer position is the origin of room's coordinates, a target point p can be represented with *metric Cartesian coordinates* as

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

where r is the distance from the observer to the wall

$$r = \begin{cases} (h_e / \sin\gamma_f) * \cos\gamma_f & \text{floor point} \\ ((h_w - h_e) / \sin\gamma_c) * \cos\gamma_c & \text{ceiling point} \end{cases} \tag{2}$$

In order to estimate h_e and h_w we perform a quick geometric calibration standing in front of the wall at a known distance r_n (3 meters in all the test cases), and aiming the phone at the wall intersections with the floor and then the ceiling and thus measuring the eye tilt angles γ_f and γ_c, respectively

$$\begin{cases} h_e = r_n * \tan\gamma_f \\ h_w = r_n * \tan\gamma_c + h_e \end{cases} \tag{3}$$

To acquire a room the observer stands approximately in the middle of the room and rotates 360 degrees to capture a video of the entire environment. While rotating, he/she should follow an ideal trajectory with the phone camera, in order to target in the middle of the screen the boundary of the wall (upper or lower) (see Fig. 1 left). During this phase we automatically store a set of *samples* for each wall, with the maximum frequency allowed by the device (see Tab. 1). Every *sample* contains 3 angles θ, γ, ρ, individuating the instant targeted point according to the illustrated model, and a time index expressed in milliseconds identifying the frame inside the video sequence. Once a room corner is reached the user records the event with a click on the screen storing its azimuth θ, then he proceeds acquiring the next wall, until he completes the whole perimeter of the room. Since points can be acquired indifferently aiming at the floor or at the ceiling, depending on the visibility from the observer's point of view, we overcome a typical problem of systems such as *MagicPlan* [22] which are prone to considerable errors when corners on the floor are occluded, with the only constraint that every single wall must be acquired along the same upper or lower boundary. In addition, like corners, doors are recorded by indicating their two azimuths with a click. Once a room is acquired the user moves to the next one aiming the phone camera to the exit door. We automatically identify the passage's door to the next room tracking the direction of the movement and storing this information in a graph of the rooms interconnections (see Sec. 4.3). A connection between two room is defined as a couple 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). In case the user should happen to visit the same room more than once (e.g., a corridor) we provided the acquisition application with an interface to manually update the room id.

4.2 Room reconstruction

We describe a room as a set of segment lines (walls) $\{\mathbf{l_1} \cdots \mathbf{l_{n+1}}\}$ whose intersections $\{\mathbf{p_0} \cdots \mathbf{p_n}\}$ define a closed polygon in metric Cartesian coordinates. A single wall l_i is identified as the 2D line that best fit the *samples* $S_i = \{\mathbf{s_0} \cdots \mathbf{s_m}\}$, acquired between the two azimuth angles θ_{i-1} and θ_i (angles marked as corners during the scene capture). To recover the linearity imposed by our acquisition method we need to decompose the problem in order to exploit the additional information provided by our integrated device, since fitting directly the samples cartesian coordinates with conventional methods (eg. RANSAC [4]) results in bad approximations (see Sec. 5). We start from the assumption that in our model the coordinates of a sample depend only by the azimuth θ, defining the heading of the current target point p (eq. 1) on the wall, and the tilt γ, defining the distance r from the observer to the wall (eq. 2). Since these samples have been acquired imposing a linear trajectory on the boundary, the coordinates of the point p are linked by a linear relation

$$y(x) = a + bx \tag{4}$$

The maximum likelihood estimate of the parameters is obtained by minimizing the quantity

$$\chi^2 = \sum_{i=0}^{m} \frac{(y_i - a - bx_i)^2}{\sigma_{y_i}^2 + b^2\sigma_{x_i}^2} \tag{5}$$

where σ_{y_i} and σ_{x_i} are the standard deviations of y and respectively x for the i sample. The occurrence of b in the denominator makes equation 5 for the slope $\partial\chi^2/\partial b = 0$ non linear. To simplify our relation we introduce the image data acquired during the room capture. Assuming θ varies linearly inside the interval between two corners, we group our data quantizing the angular interval between θ_{i-1} and θ_i into finite intervals (eg. 0.5 degrees steps) $\{\theta_0 \cdots \theta_m\}$. For each discretized θ_d we considered the *samples* $S_d = \{\mathbf{s_0} \cdots \mathbf{s_k}\}$ which lie inside the quantization interval, then we calculate

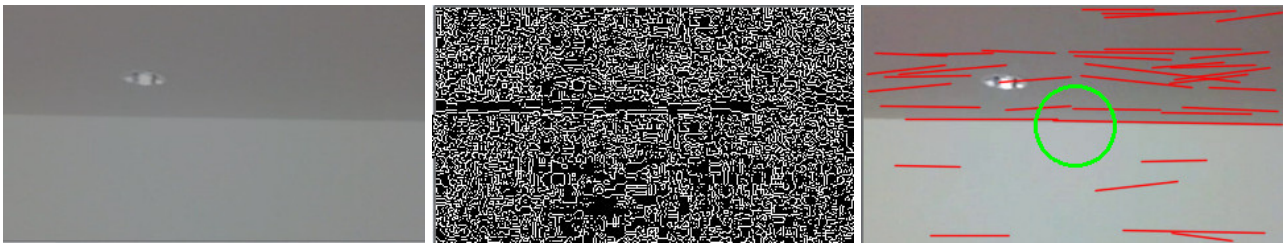

Fig. 3 Left: A video frame associated with a sample (image enhanced for printing). Despite for an human eye the line between ceiling and wall looks clear, it could not be the same for an automatic edge detector (see central figure). Center: The noise of the frame without filtering, evidenced by a Canny edge detector. Right: The edge line detected after filtering and Hough transform.

a representative sample s_d having θ_d as azimuth angle and a weighted γ_d mean angle

$$\gamma_d = \frac{\sum_{i=0}^{k}(\gamma_i w_i)}{\sum_{i=0}^{k} w_i} \tag{6}$$

with an individual weighted standard variance

$$\sigma^2_{\gamma_d} = \frac{\sum_{i=0}^{k} w_i \left(\gamma_i - \gamma_d\right)^2}{\sum_{i=0}^{k} w_i} \tag{7}$$

and with the weights $\{w_0 \cdots w_k\}$ calculated from the k frames associated with the samples $S_d = \{\mathbf{s_0} \cdots \mathbf{s_k}\}$ as follow.

We consider the frame f_i of the sample s_i (Fig. 3 left). The walls intersections with the ceiling or with the floor are usually dark and noisy in an image acquired with a mobile phone, reason for standard features detectors (see for example [12,23]) often consider these regions as *background*, failing to find information to perform camera calibration (i.e. an adequate number of vanishing points). Since for the scope of our analysis we are not interested in a full image calibration and furthermore we already roughly know where to search for the edge line, we focus our attention only on estimating the reliability of our measures in image space. To attenuate the typical noise originated by the compression and by the sensor limitations (see Fig. 3 center) we apply a *Non Local Mean* filter based on [2] on the luminance channel, then we run a standard *edge detector* (i.e.*Canny* [3]) with a low threshold, isolating the break in the shadows due to the wall edge geometry. We transform the filtered image in a *Hough Space* [15] having as origin the center of the image, obtaining several lines (see Fig. 3 right)

$$d_i = x\cos\alpha + y\sin\alpha \tag{8}$$

with d distance of the line to the center and alpha the angle defining the direction of the line. Since the center of the image is also the projection in image space of the sample with the values θ_i and γ_i, we choose the line with the lowest value d_i and we assign the weight w_i to the sample s_i

$$w_i = 1 - d_i/max_d \tag{9}$$

with max_d the maximum distance admitted to consider valid the line (i.e. 10 pixels). Samples with lines found outside the max_d circle are marked as *outliers* and discarded. Once we have externally calculated the representative *samples* $\{\bar{\mathbf{s_0}} \cdots \bar{\mathbf{s_m}}\}$ we can simplify equation 5 as follows

$$\chi^2 = \sum_{i=0}^{m} \frac{(y_i - a - bx_i)^2}{\sigma_i^2} \tag{10}$$

with $\sigma_i \approx \sigma_{\gamma_i}$. Equation 10 can be minimized to determine a and b. At its minimum derivatives of $\chi^2(a,b)$ with respect to a and b vanish, resulting in a system of two equations in two unknowns, giving the solution for the best-fit model parameters a and b.

$$\begin{cases} a*(\sum_{i=0}^{m}\frac{1}{\sigma_i^2}) + b*(\sum_{i=0}^{m}\frac{x_i}{\sigma_i^2}) = \sum_{i=0}^{m}\frac{y_i}{\sigma_i^2} \\ a*(\sum_{i=0}^{m}\frac{x_i}{\sigma_i^2}) + b*(\sum_{i=0}^{m}\frac{x_i{}^2}{\sigma_i^2}) = \sum_{i=0}^{m}\frac{x_i*y_i}{\sigma_i^2} \end{cases} \tag{11}$$

From the derivatives of a and b with respect to y_i we can also evaluate the variances σ_a^2 and σ_b^2, that allow us to estimate the probable uncertainties of our fitting (see Sec. 5, respectively for scale and direction. We estimate the goodness-of-fit of the samples to the wall model through the *incomplete gamma function*

$$Q = gamma(\frac{m-2}{2}, \frac{\chi^2}{2}) \tag{12}$$

We adopt this estimator considering its values can be precomputed and tabulated, thus facilitating the use on low-end devices. For each intersection (corner) p_i of the estimated walls l_i and l_{i+1} we store an angle ϕ_i

$$\phi_i = \begin{cases} angle(l_i, l_{i+1}) & Q(l_i) \text{ and } Q(l_{i+1}) > 0.1 \\ sign(l_i, l_{i+1}) * 90 & Q(l_i) \text{ and } Q(l_{i+1}) < 0.1 \end{cases} \tag{13}$$

resulting in the set of corner angles $\{\phi_\mathbf{0} \cdots \phi_\mathbf{n}\}$, with ϕ calculated from the walls intersection if its estimation is considered reliable, otherwise approximated to 90 degrees (with the sign of the lines intersection). In a similar way we define a rank between corners, combining σ_a^2 and σ_b^2 for each intersection. Due to angles approximation the resulting polygon defining the room shape is not always perfectly closed as we expect. We perform a further optimization step to refine our shape. For every possible direction β we trace an intersection ray from the corner with the best ranking p_0 and direction β to intersect the *ray(origin,θ_i)*, traced from the origin of the room along the heading of the corner θ_i (see Figure 4). We continue the procedure for each corner

$$P_i = intersect(ray(origin, \theta_{i+1}), ray(P_i, \alpha_i)) \qquad (14)$$

with $\alpha_i = \beta + \phi_i$, until we complete all intersection and and we close the polygon. As last point we obtain an estimated $p_0{}^*$, which in an ideal case must coincide with the starting point p_0. Then we calculate the distance between $p_0{}^*$ and p_0 and we choose the β that minimizes this distance, estimating the definitive shape. The generated room is already scaled in real world metric units, without the need for further manual editing in contrast to [11,20,22].

4.3 Floor plan generation

Once all shapes have been calculated we choose the room $\mathbf{r_0}$ with the global best fit values (i.e. eq. 12, σ_a^2 , σ_b^2) as origin of the whole floor coordinates system, and then we exploit the information stored in the graph to align the other rooms $\{\mathbf{r_1} \cdots \mathbf{r_N}\}$ to the first one. For each connection we calculate the affine transform $M_{j,j+1}$ representing the transform from the room r_{j+1} to the room r_j coordinates. Since a connection between two rooms is defined as a couple of doors that fundamentally are the same door expressed in different coordinates, we obtain $M_{j,j+1}$ applying a standard least squares method to the corresponding door extremities. From the rooms connectivity graph we calculate for each room $\mathbf{r_p} \in (\mathbf{r_1} \ldots \mathbf{r_N})$ the path to $\mathbf{r_0}$ as a set of transforms $\{\mathbf{M_1} \cdots \mathbf{M_p}\}$, representing the passages encountered to reach $\mathbf{r_0}$ and the whole transformation from $\mathbf{r_p}$ to $\mathbf{r_0}$ coordinates. At the end of this passage we have obtained a floor plan fully aligned and scaled, rather than in [20] where the user manually indicates the matching doors and the different scale factors for each room and in [22] where the alignment is manual. Moreover the generated scene graph and the image data acquired can be exploited as input for systems like [8], for example to enable interactive photo-realistic tours of the indoor scene.

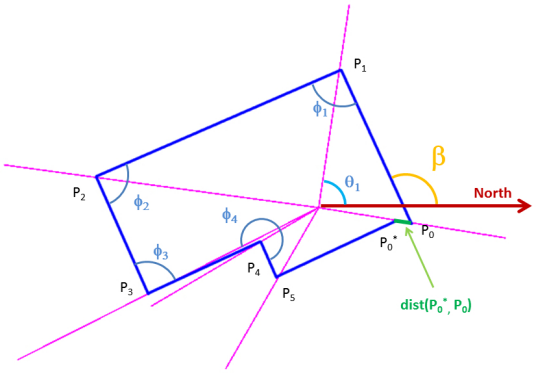

Fig. 4 *Room shape optimization* Illustrative room with the error intentionally increased. For every possible direction β we trace an intersection ray from the corner with the best ranking p_0 and direction β to intersect the *ray(origin,θ_i)*, traced from the origin of the room along the heading of the corner θ_i. Then we calculate the distance between $p_0{}^*$ and p_0 and we choose the β that minimizes this distance, estimating the definitive shape.

5 Results

We developed a framework including two distinct applications: *scene capture* and *scene processing*. The *scene capture* application has been developed as a simple and intuitive tool working on any Android device with camera, accelerometer and magnetometer, features currently available on commodity smart-phones and tablets. The *scene processing* part is implemented both for Android and for PC desktop applications (Windows and Linux), in fact the peculiar solution proposed makes the method scalable for mobile devices, avoiding long iterations and hardware consuming routines. For the mobile version we propose an optional image filtering based on a *Gaussian kernel*, since this second filter is less time consuming than *Non Local Means* filter. The loss in quality of the reconstruction is strictly related to the quality level of the video and the phone's camera, the discussion about is beyond our purpose. To capture the environments we intentionally employed a mid-low end device, an Acer Liquid E2 Duo, with 1.2 GHZ quadcore processor MediaTek MT6589, RAM 1 GB, a PowerVR SGX 544MP GPU, an accelerometer Bosch BMA050, a magnetometer Bosch BMM050 and 8 Mpixels camera. We present 7 significant test cases of office and residential environments, divided in scenes **MW** (Manhattan

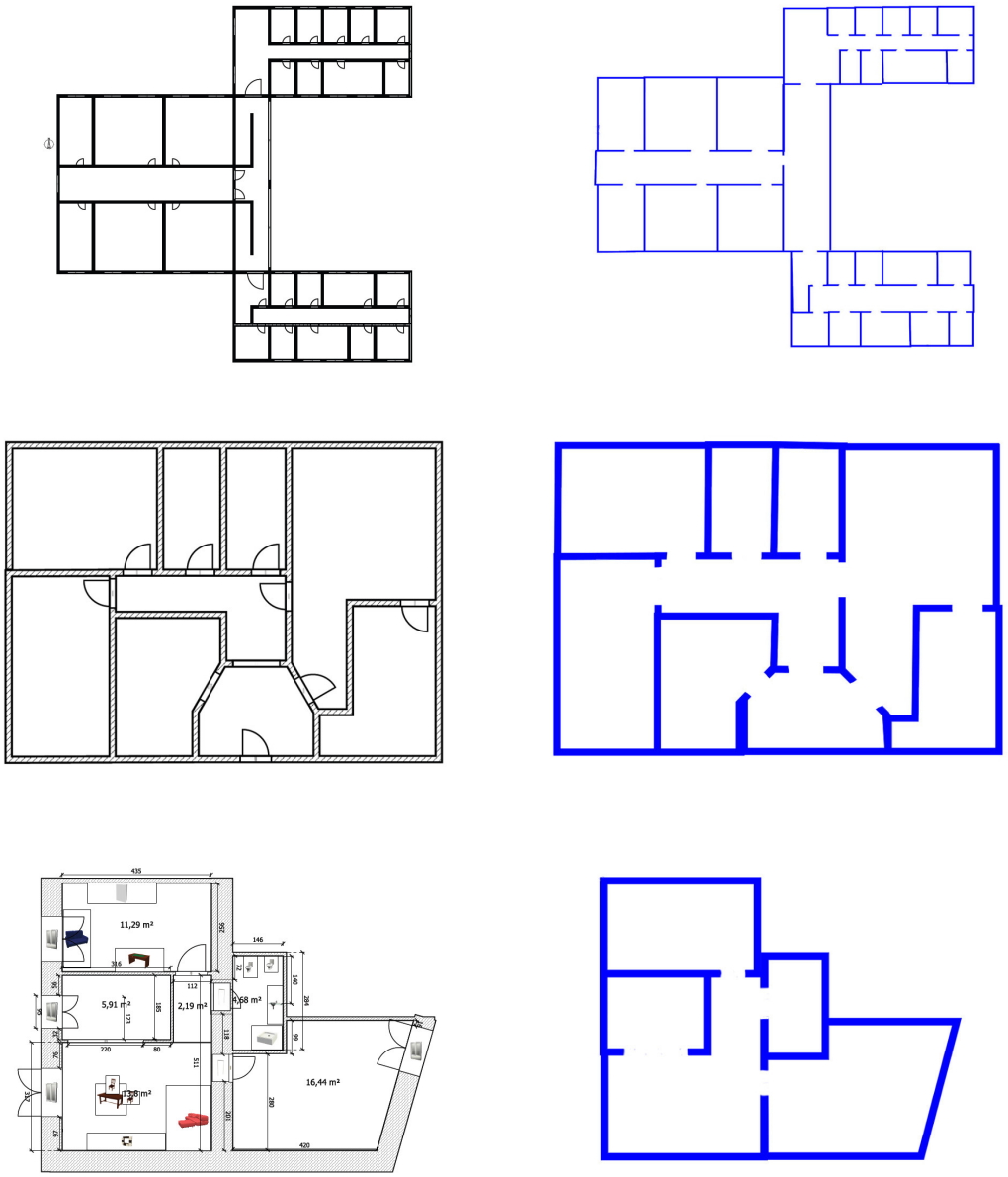

Fig. 5 Comparison between the building map (right) and our reconstruction (left blue) of three of seven cases presented. Since our method does not adopt any specific *Manhattan World* constraint, the two groups are threatened exactly in the same way. In reality the true measures do not always coincide with the building layout supplied by the architects. For the purpose of this work the differences are negligible, so we assume the provided floor plan data as ground truth and we consider as error the differences with respect to our reconstruction.

World) and **NMW** (Non Manhattan World)(see Fig. 5 and Table 1). Since our method does not adopt any specific *Manhattan World* constraint, the two groups are treated exactly in the same way. We perform the initial calibration step from a distance r_n of 3 meters (see Sec.4.1). For almost all rooms we acquired the samples on the ceiling intersection, since the visibility of the floor was often occluded by the furniture. We acquired instead the samples on the floor when the upper part of the wall contains awnings, ducts for ventilation,etc. It is important to note that, similarly to other systems, stairs, sloped ceilings and walls that do not belong to the perimeter of the room can not be managed by the system. Moreover, in reality, the true measures do not always coincide with the building layout supplied by the architects. For the purpose of this work the differences are negligible, so we assume the provided floor plan data as ground truth and we consider as error the

Scene	Features		Samples per room		Errors				MagicPlan
	Area mq	Rooms	Avg number	Valids %	Area %	Corner angle max	Corner pos max	Unrel. corners	editing time
MW Office F1	875	29	2289	64	1.78	1.2 degrees	25 cm	8	41m23s
MW Office F2	875	25	2426	60	1.58	1.6 degrees	20 cm	12	37m11s
MW Office F3	320	8	1852	69	1.16	0.7 degrees	15 cm	0	17m35s
NMW Office F1	185	12	2046	72	1.74	4.2 degrees	12 cm	3	28m30s
NMW Office F2	180	7	2224	86	1.42	1.1 degrees	10 cm	0	18m30s
NMW House 1	70	5	2312	54	2.12	3.7 degrees	23 cm	1	23m30s
NMW House 2	138	9	2128	57	3.67	4.5 degrees	18 cm	1	34m10s

Table 1 The approximate acquisition time has been the same needed by *MagicPlan*. However no editing time is required by our method. The number of total samples acquired is proportional to the time spent by the user in the room and the refresh time of the sensors. This rate is adapted dynamically during the acquisition (eg. compass not reliable, etc.)and can range from 10Hz to 50 Hz, affecting also the video spatial indexing. MW Office was built in 2002, NMW Office in 1998, NMW House 1 in 1934 and NMW House 2 in 1957.

differences with respect to our reconstruction. We show in Tab. 1 the *direction* error and the *scale* error. The maximum error in degrees showed in Tab. 1 is relative to *reliables* corners. Corners marked as *unreliable* have been instead approximated as indicated in Sec. 4; their number grows with the number of the rooms visited and with the presence of corridors or narrow rooms. For these specific cases our method does not work properly and we adopt a *Manhattan World* backup solution to complete the reconstruction. The *scale* error is shown by the columns *Area* and *Corner position*. The area error is calculated as the ratio of area incorrectly reconstructed to the total ground truth area, whereas the corners position max error is the max depth error observed for each corner. Although the results between the area and the absolute corner positions may seem contradictory we evince that the overall shape and surface of every room is maintained, thanks to a good wall direction estimation. In Figure 2 we highlight the difference between a robust *M-estimator* applied to the samples of a room with only right angles (samples in black, estimated lines in green), showing the resulting wrong approximation. In the same figure we see instead the same samples fitted with our method, where we can evince how outlier points (upper left corner) have been discarded from the the wall computation thanks to their images feedback. In Fig. 6 we show an example of *Non Manhattan World* room reconstruction. We show in black the average trajectory of the camera tracking the boundary and in red the estimated wall directions. Almost all the samples in the corners 1,2 and 4 have been discarded through the images weights. We found 2 not right angles in corner 5 (97 degrees) and 4 (83 degrees) and we approximate to 90 degrees the others with a variance of ± 0.5 degrees. The time needed to acquire the scene has been the same needed by *MagicPlan*(time to walk through the scene targeting the features on the wall). We show also the editing time required by *MagicPlan* to obtain the same results produced by our automatic method. In many cases (example NMW House 1) *MagicPlan* was unable to approximate 3 of 5 rooms due to the furniture on the floor occluding the view, needing for additional editing time.

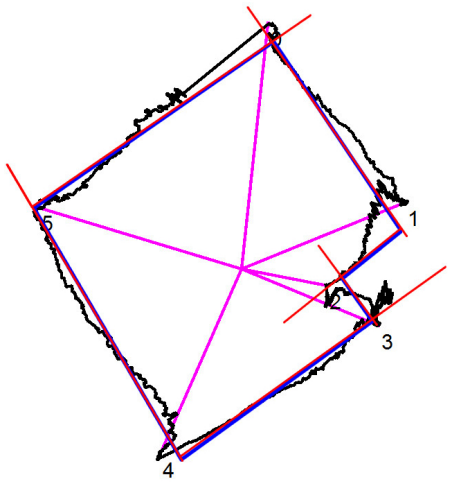

Fig. 6 *Non Manhattan World* room example. We show the samples tracked by the phone as a trajectory in black, the estimated directions in red and the reconstructed shape in blue. Almost all samples in corners 1,2 and 4 have been discarded through the images weights. The estimated NMW corners are 97 degrees for corner 5 and 83 degrees for corner 4

6 Conclusions

We presented a framework for mobile devices enabling non-technical people to automatically map and reconstruct multi-room indoor structures. To achieve our results we don't need to impose *Manhattan World* constraints as previous methods did, taking advantage of the redundancy of the modern smartphones instruments. Our computational solution was conceived to be scalable on the mobile devices hardware, as well as the re-

sults obtained can be useful in many real-world applications. The recent unveiling of *Google Project Tango* highlights the interest for such kind of applications, especially those focused on the structure of a building rather than the details of the model. These applications can be for example the definition of thermal zones for energy simulation or the support for evacuation simulations and estimation for circulation of people in commercial/public/office buildings, without requiring any specialized equipment or training to model the scenes. For the future we plan to extend our methods exploiting the new instruments available on the next generation mobile devices, as integrated depth sensors or tools as *Google Glasses*, with the intent to capture and manage more complex features as furniture, transparent windows, curved walls, and offer support to new and more advanced visual and physical simulations.

Acknowledgements This research is partially supported by EU FP7 grant 607737 (VASCO). We also acknowledge the contribution of Sardinian Regional Authorities.

References

1. Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., Maierhofer, S.: O-snap: Optimization-based snapping for modeling architecture. ACM Trans. Graph. **32**(1), 6:1–6:15 (2013). DOI 10.1145/2421636.2421642. URL http://doi.acm.org/10.1145/2421636.2421642
2. Buades, A., Coll, B., Morel, J.M.: Non-Local Means Denoising. Image Processing On Line **1** (2011)
3. Canny, J.: A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on **PAMI-8**(6), 679–698 (1986). DOI 10.1109/TPAMI.1986.4767851
4. Capel, D.: An effective bail-out test for ransac consensus scoring. In: Proc. BMVC, pp. 629–638 (2005)
5. Cornelis, N., Leibe, B., Cornelis, K., Gool, L.V.: 3d urban scene modeling integrating recognition and reconstruction. Int. J. Comput. Vision **78**(2-3), 121–141 (2008)
6. Coughlan, J.M., Yuille, A.L.: Manhattan world: Compass direction from a single image by bayesian inference. In: Proc. ICCV, vol. 2, pp. 941–947 (1999)
7. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '96, pp. 11–20. ACM, New York, NY, USA (1996)
8. Di Benedetto, M., Ganovelli, F., Balsa Rodriguez, M., Jaspe Villanueva, A., Scopigno, R., Gobbetti, E.: Exploremaps: Efficient construction and ubiquitous exploration of panoramic view graphs of complex 3d environments. Computer Graphics Forum **33**(2) (2014). Proc. Eurographics 2014
9. El-Hakim, S.F., Boulanger, P., Blais, F., Beraldin, J.A.: System for indoor 3D mapping and virtual environments. In: S.F. El-Hakim (ed.) Videometrics V, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 3174, pp. 21–35 (1997)
10. Frueh, C., Jain, S., Zakhor, A.: 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)
11. Furukawa, Y., Curless, B., Seitz, S.M., Szeliski, R.: Reconstructing building interiors from images. In: Proc. ICCV (2009)
12. Guru, D., Dinesh, R.: Non-parametric adaptive region of support useful for corner detection: a novel approach. Pattern Recognition **37**(1), 165 – 168 (2004). DOI http://dx.doi.org/10.1016/S0031-3203(03)00234-6
13. inc., G.: Google project tango (2014). URL http://www.google.com/atap/projecttango/
14. Kim, Y.M., Dolson, J., Sokolsky, M., Koltun, V., Thrun, S.: Interactive acquisition of residential floor plans. In: Proc. IEEE ICRA, pp. 3055–3062 (2012)
15. Matas, J., Galambos, C., Kittler, J.: Robust detection of lines using the progressive probabilistic hough transform. Computer Vision and Image Understanding **78**(1), 119 – 137 (2000). DOI http://dx.doi.org/10.1006/cviu.1999.0831
16. Müller, P., Wonka, P., Haegler, S., Ulmer, A., Van Gool, L.: Procedural modeling of buildings. In: ACM SIGGRAPH 2006 Papers, SIGGRAPH '06, pp. 614–623. ACM, New York, NY, USA (2006)
17. Mura, C., Jaspe Villanueva, A., Mattausch, O., Gobbetti, E., Pajarola, R.: Reconstructing complex indoor environments with arbitrary wall orientations. In: Proc. Eurographics Posters. Eurographics Association (2014)
18. Mura, C., Mattausch, O., Jaspe Villanueva, A., Gobbetti, E., Pajarola, R.: Robust reconstruction of interior building structures with multiple rooms under clutter and occlusions. In: Proc. 13th International Conference on Computer-Aided Design and Computer Graphics (2013)
19. Pollefeys, M., et al.: Detailed real-time urban 3d reconstruction from video. Int. J. Comput. Vision **78**(2-3), 143–167 (2008)
20. Sankar, A., Seitz, S.: Capturing indoor scenes with smartphones. In: Proc. ACM UIST, pp. 403–412 (2012)
21. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Proc. CVPR, vol. 1, pp. 519–528 (2006)
22. Sensopia: Magicplan (2011). URL http://www.sensopia.com
23. Shi, J., Tomasi, C.: Good features to track. In: Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on, pp. 593–600 (1994). DOI 10.1109/CVPR.1994.323794
24. Shin, H., Chon, Y., Cha, H.: Unsupervised construction of an indoor floor plan using a smartphone. IEEE Trans. Systems, Man, and Cybernetics **42**(6), 889–898 (2012)
25. Sinha, S.N., Steedly, D., Szeliski, R., Agrawala, M., Pollefeys, M.: Interactive 3d architectural modeling from unordered photo collections. In: ACM SIGGRAPH Asia 2008 Papers, SIGGRAPH Asia '08, pp. 159:1–159:10. ACM, New York, NY, USA (2008)
26. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. ACM TOG **25**(3), 835–84 (2006)
27. Stamos, I., Yu, G., Wolberg, G., Zokai, S.: 3d modeling using planar segments and mesh elements. In: 3D Data Processing, Visualization, and Transmission, Third International Symposium on, pp. 599–606 (2006). DOI 10.1109/3DPVT.2006.5
28. Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., Quan, L.: Image-based facade modeling. In: ACM SIGGRAPH

Asia 2008 Papers, SIGGRAPH Asia '08, pp. 161:1–161:10. ACM, New York, NY, USA (2008)
29. Zebedin, L., Bauer, J., Karner, K., Bischof, H.: Fusion of feature- and area-based information for urban buildings modeling from aerial imagery. In: Proc. ECCV, pp. 873–886 (2008)

Giovanni Pintore is a researcher in the Visual Computing group at the CRS4 research center, Italy. He holds a Laurea (M. Sc.) degree (2002) in Electronics Engineering from the University of Cagliari, Italy. His research interests include multiresolution representations of large and complex 3D models, light-field displays, reconstruction and rendering of architectural scenes through new generation mobile devices.

Enrico Gobbetti Director of Visual Computing at the CRS4 research center, Italy. His research focuses on the creation and application of efficient techniques for acquisition, processing, distribution, and exploration of complex 3D scenes. Enrico holds an Engineering degree (1989) and a Ph.D. degree (1993) in Computer Science from the EPFL, Switzerland.