Interactive Syntactic Modeling With a Single-Point Laser Range Finder and Camera

Thanh Nguyen* Raphael Grasset† Dieter Schmalstieg‡ Gerhard Reitmayr§

Institute for Computer Graphics and Vision
Graz University of Technology

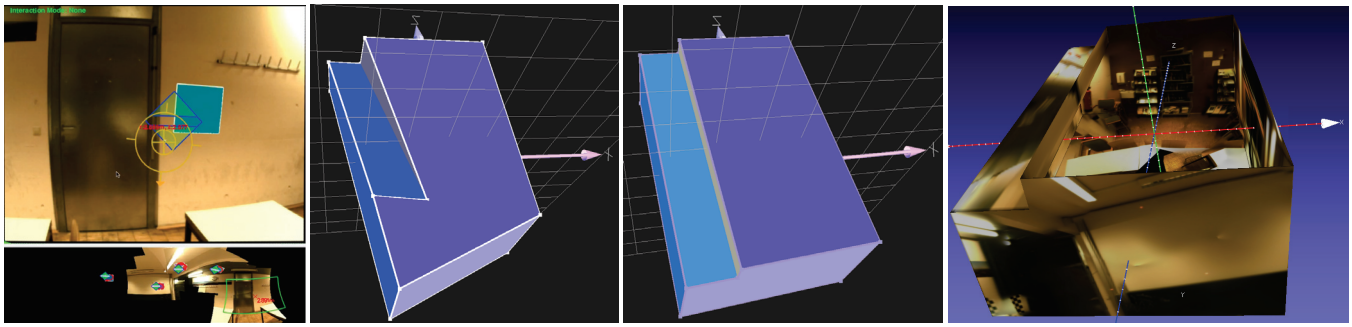

Figure 1: Our method consists of capturing a panorama of the environment and entering geometric primitives (far left). Accumulated errors from panoramic tracking allow only for an inaccurate model (left). Therefore, we incorporate the input into an underlying constrained reconstruction pipeline in order to enhance the final model (right). Consequently, our method enables producing highly qualitative and meaningful models quickly (far right).

Abstract

In-situ 3D Modeling becomes increasingly prominent in current Augmented Reality research, particularly for mobile scenarios. However, real-time performance and qualitative modeling remain highly challenging. In this work, we propose a new interactive 3D modeling approach for indoor environments, combining an assistive user interface and constrained reconstruction with a device consisting of a single-point laser range finder and a camera. Using our system, a user pans around capturing a panorama of the environment, while simultaneously measuring the distance to a single point per frame. An automatic detection process estimates planes from these sparse 3D measurements. The user can highlight specific geometric features in the environment, such as 2- or 3-way corners, with simple gestures, adding more 3D points to the estimation. The segmented planes are refined using a constrained optimization, enforcing orthogonality and parallel constraints as well as minimizing the number of planes used in the reconstruction. Finally a volumetric space-carving approach determines the geometry of the environment. Our reconstruction approach can output highly accurate models built only from simple, clean geometry. To examine the quantitative performance of our approach, we run evaluations on both synthetic and real data.

1 Introduction

Reconstructing 3D models of the environment has become an important topic in recent Augmented Reality (AR) research. Such models can help to improve user experience with AR environments concerning tracking, visualization, and interaction. Yet, there is still a large gap towards real-time reconstruction of 3D models, particularly with meaningful structural information (*syntactic model*). Current techniques generally produce a large set of geometric primitives, which is often the result of fully automatic methods and have limited long-term usage. As AR has shifted to mobile situations, creating in-situ, efficient 3D modeling tools is a prime research goal.

In this paper, we propose a new modeling approach that combine affordable technology and user assisted algorithms to reduce this gap. The new approach overcomes current difficulties in reconstructing syntactic models whilst offering an intuitive user interface. Our proposed system employs a visual panoramic SLAM approach for tracking from a single camera, combined with a single-point laser range finder. The system enables a user to interactively input simple geometric primitives such as planes and intersections of planes, using a set of simple gestures combined with visual guidance. Both the sparse 3D point measurements from the single laser-range finder and appearance from the camera are used to estimate a set of planes bounding the environment. Through constrained optimization we refine the model, enforcing orthogonality and parallel planes. At the same time the number of primitives used in the model is kept small. A volumetric space carving approach extracts the final geometry of the environment. The result is a minimal model consisting of planes with maximal extent observing the typical constraints of man-made environments. Figure 1 illustrates our method and resulting models.

The main contributions of this paper are:

New modeling approach Our proposed approach uses a simple hardware platform, combining inexpensive devices, making it accessible to a large group of potential users. It enables the user to reconstruct qualitative and meaningful 3D models in real-time.

Constrained reconstruction Our system automatically discovers geometric constraints and iteratively incorporates these constraints

*e-mail: thanh@icg.tugraz.at
†e-mail:raphael.grasset@icg.tugraz.at
‡e-mail:schmalstieg@icg.tugraz.at
§e-mail:reitmayr@icg.tugraz.at

into the underlying structure-from-motion reconstruction process. This allows us to reconstruct highly accurate models in limited conditions where the baseline between camera poses is very small.

To assess our work, we run synthetic and empirical evaluation to demonstrate accuracy, reliability and robustness of our approach as well as presenting qualitative results.

2 Prior Work

Our motivation for syntactic models is inspired by work of Sinha et al. [17], who described a system where the reconstructed model is formed from planes. The authors proposed an interactive editor for making 3D models from images in multiple views. A user needs to trace polygons, in order to input planes to the underline pipeline of making models, which can be a time-consuming task. Unlike the above approach, modeling tools in AR usually focus on real-time performance aspect (immersive 3d modeling).

Baillot et al. [1] was one of the early work in AR 3D modeling, proposing a mobile modeling platform for outdoor scenarios, using manual input for iteratively creating geometric primitives. Lee et al. [10] describe an immersive modeling system for virtually copying real world objects, combining a HMD for real time visualization and a tracked stylus for specifying manually geometric primitives. Piekarski et al. [13, 14], propose the usage of planes for creating geometries, using a glove based interface with a wearable platform for outdoor situations.

For indoor scenarios, Freeman et al. [6] described a a video based modeling solution for rapidly building small models using computer vision based tracking and a primitive based matching tool. Langlotz et al. [9] relies on a panorama tracker and a mobile device for sketching simple 3D geometric models in outdoor or indoor contexts. Sankar and Seitz [15] describe a system for generating building models using the sensors of a smartphone.

All of these methods requiring manual input are physically demanding. Another range of work in AR modeling explored how computer vision techniques can be used to simplify the modeling process. Bunnun et al. [4] presented a technique relying on a 3D pointer and epipolar geometry constraint to create basic wireframe 3D models. Hengel et al. [18] introduced an interesting hybrid solution using offline modeling techniques, as proposed in their previous work in VideoTrace [7], and a real-time SLAM system to improve the modeling experience. Furthermore, the authors later proposed a more intuitive interaction tool for modeling [2]. In the more recent approach, users only need to briefly input region of interest on object they want to model. Nevertheless, the approach is considerably limited to small objects and depends on the visual hull for reconstruction, which requires large coverage of the reconstructing object.

In contrast, Simon [16] introduced direct in-situ interaction techniques using with a SLAM approach. Pan et al. [12] also explored direct in-situ modeling for small objects, using an interactive approach where the end-user has to rotate a model in front of the camera.

Our use of the combination of a single-point laser range finder and a camera is closely related to work of Wither et al. [19]. They developed a reconstruction tool which employs panoramic tracking together with single-point laser range finder in order to automatically reconstruct a depth map of outdoor environment. Even though they offered a very simple to use tool, the accuracy and quality of reconstructed depth map in their approach was rather limited, especially for indoor environments. Our work rather aims at creating a highly simplified model, using minimal user input to denote planes in the environment.

In this work, we aim to build an easy to use modeling tool which can produce highly accurate, qualitative, and meaningful models. Our approach uses the combination of single-point laser range finder and camera in an egocentric situation. This enables end user to quickly and easily make a 3D model while standing in a single spot. Similar to the work of Wither et al., we use panoramic tracking to initialize the modeling procedure. Moreover, we exploit user input of geometric primitives in order to improve accuracy in both pose estimation and resulting models, which is neglected in other approaches.

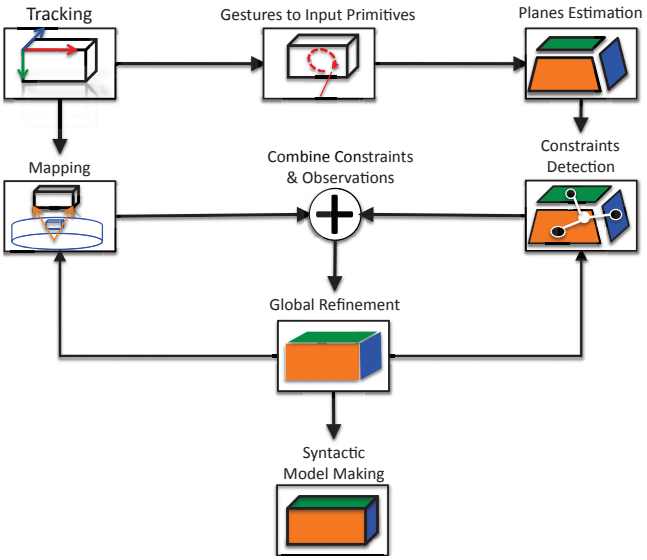

Figure 2: Overview of the proposed constrained reconstruction pipeline.

3 Overall Approach

Consider the following simple scenario: *To obtain the 3D model of her office, Roberta stands somewhere close to the center of the room. Equipped with a mobile AR system, she proceeds, through simple gestures, to mark-up geometric features that she wants to capture in the final model. These features include a single plane, a corner between two or three planes, or an edge of a plane. After marking up all the important features, a 3D model is automatically computed and visualized on her AR screen. She can pursue to the next room to create a complete model of her building.*

Figure 2 gives an overview of the technical steps of our approach. The main idea is to generate 3D model consisting of full bounding polygons, defined by intersecting the planes at the specified locations and constrained to follow typical relationships in man-made environments.

During the interaction phase, our system captures 3D range measurements at each video frame using a specific AR measurement device (section 4.1). We combine it with a panoramic SLAM method (section 4.2) to estimate the relative directions of these range measurements resulting in a very sparse point cloud (section 4.3). From this point cloud, plane hypotheses are created, using both depth and appearance information (section 5.1). Then a common model of planes is estimated by jointly optimizing the plane parameters, camera poses and point locations as well as plane/point associations (section 5.3). Finally, the extents of the planes are computed from a volumetric representation (section 5.4). Figure 2 gives an overview of the technical steps in our approach.

While a random set of 3D points might suffice for this process, we explicitly rely on user input to make the system more robust. The panoramic tracking and mapping provides direct interaction to let the user highlight important parts of the geometry using simple gestures (section 4.4). Specifically, we use two types of gesture: a single scanning gesture that performs roughly a large circle around the room, and a circular local gesture that sweeps over all parts of a

more complex geometric configuration. The later is used to mark a single wall or corners between two or three walls. The input from a single gesture is a local 3D point cloud that is used to create initial plane hypotheses for the following reconstruction steps.

4 Modeling Platform

This section describes the user interface and the available interactions to create a model in more detail.

4.1 User Interface

Our mobile AR interface combines a handheld input device and a mobile display (Figure 3). The input device combines a single camera, a single-point laser range finder, an IMU and a wireless touch button (Figure 4 top). The camera operates with a 720×480 input image at $30Hz$, and the laser range finder is accurate to less then $1mm$ at distances below $10m$.

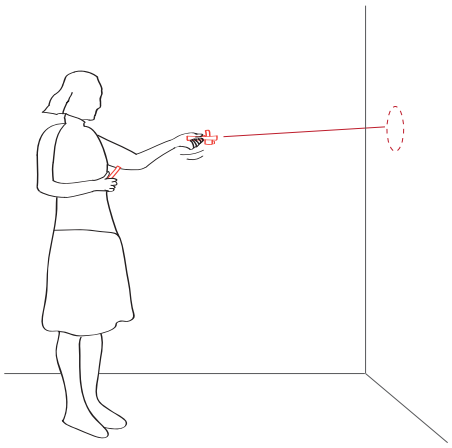

Figure 3: User Interface Concept: A user executes a gesture with our AR system, using a handheld device (left hand) for input control and a mobile display (right hand) for information display. Bottom: the handheld input device with its different components.

The mobile display is used to interactively visualize the reconstruction with an AR view, a VR view and additional control view providing an overview of the captured environment (Figure 4 bottom). The user can naturally choose any direction and proceed to a modeling step, while all three views are continuously updated and aligned. At any direction, the user can initiate an interaction to input a geometric feature.

The different views are synchronized during execution. The VR view, presenting the current reconstructed model, is synchronized with an AR view that provides feedback about the current gesture and feature estimation. The control or tracking view is also synchronized with the AR view to provide a global overview of the environment and show the current viewing direction.

The laser range finder is mounted close to the camera and pointing roughly along the optical axis. Therefore the measurement point (laser dot) is always displayed within the video frame but can be sometimes hard to perceive (color contrast, camera auto adjustments). To ensure that it is always visible to the user, a 3D cursor positioned at the measured location is projected into the AR view and the current distance is displayed.

4.2 Panoramic Tracking

We use a panoramic tracking and mapping approach as described by Kim et al. [8]. Our tracking simultaneously estimates the camera rotation $\mathbf{T} \in \mathbb{SO}(3)$ from live video input while constructing a panoramic map consisting of keyframes. We add a new keyframe

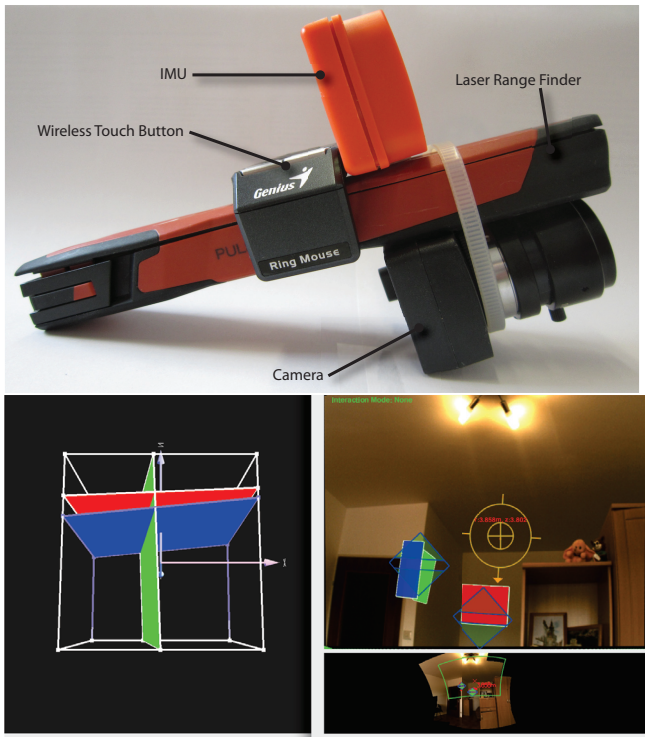

Figure 4: User Interface Components: input and display. Top: the handheld input device with its different components. Bottom: The visual interface presented on the mobile display includes three main window views, the AR view (top-right) demonstrating current live video input with overlaid local planes, the VR view (top-left) showing corresponding reconstructed planes in common panorama coordinates, and the panorama view showing captured panorama map with highlighted gesture inputs.

when the smallest angle between the current camera direction and all existing keyframe directions exceeds a certain threshold (e.g., half of the field of view). For each new keyframe, we detect keypoints and represent them in world frame as $W_j = \mathbf{T}^{-1} \cdot [x_j, y_j, 1]^T$, where $[x_j, y_j]$ is the location of a keypoint in the new keyframe with rotation $\mathbf{T}$. The pose estimation for each input video frame then minimizes the reprojection of these key points

$$C(\mathbf{T}_i) = \sum_j \| p_j - \mathrm{Proj}(\mathbf{T_i} \cdot W_j) \|_2 \qquad (1)$$

where p_j is observation on camera plane of point W_j in frame pose $\mathbf{T}$. $\mathrm{Proj}(\cdot)$ is the camera projection function including radial distortion and a standard 3×3 camera calibration matrix.

4.3 Recording 3D Measurements

The single point laser range finder (LRF) provides a single, highly accurate distance measurement with a frequency between 5 and 0.1 Hz. The measurement frequency depends on the target surface, distance and motion while measuring. The LRF is mounted close to the camera and pointing along its optical axis. To map the distance measurement precisely into the camera coordinate frame, we calibrated both the origin of the laser ray C and the direction D with respect to the camera coordinate frame, using the method described by Ngyuen et al. [11]. A laser measurement l corresponds to the 3D point $\mathbf{P} = C + l \cdot D$ in the local camera coordinate frame.

For any range measurement l, the corresponding local 3D point $\mathbf{P}$, the current camera rotation $\mathbf{T}$ and video frame is stored for further processing. The laser is measuring continuously, but measure-

ments are only stored during a gesture operation as described in the following section.

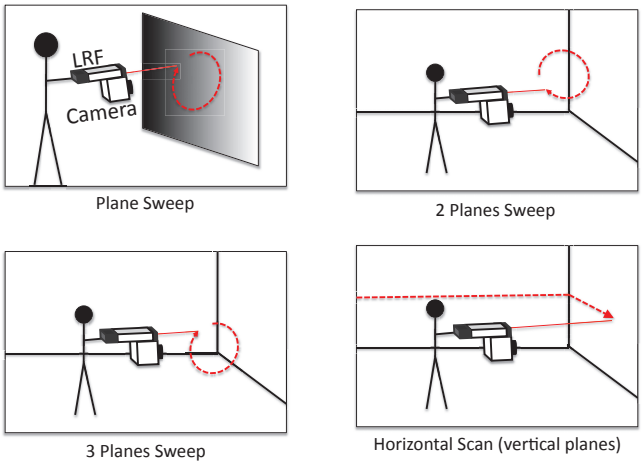

Figure 5: Sweep and scan gestures are used to input primitives to the system.

4.4 Gestures

Modeling is operated through pointing the handheld input device at features to be reconstructed and collecting 3D measurements during motion of the handheld device over the feature. However, instead of completely automatic detection of features, we rely on the user to tell the system when an interesting feature should be recorded through gestures. After examining various gestures for mobile devices and interaction at a distance on large screens, we selected a simple set of intuitive gestures: sweep and scan. The sweep gesture (SWEEP) and scan gesture (SCAN) are built on natural hand motions that one can naturally execute with a laser pointer (Figure 5).

SWEEP: This gesture is the main input mode for defining plane(s) in the system. The user is only required to sweep back and forth over an area of interest a couple of times to create a tight cluster of 3D measurements. The features can be a single plane, but also the intersections of 2 or more planes (see Figure 5c) and d)). Instead of sweeping over every plane of a rectangle individually, only two sweeps in two opposite corners are necessary to enter the whole geometry. By supporting the capturing of multiple planes, we offer an efficient method to determine the geometry of an environment.

SCAN: This gesture is an extension of sweep to input the overall geometry of an environment in a single motion, for covering a large extent. Instead of creating a local cluster of measurements, the user turns around while scanning, capturing a wide angle of possible scanning directions. This is more efficient than the SWEEP mode for quickly capturing the outline of a room. While such a 1D line of measurements may not fully constrain the geometry, we add specific assumptions to estimate the geometry measured in this mode (see section 5.2).

Inspired by the approach described by Bau and Mackay [3], we provide visual hints for gesture prediction and completion after the user triggers the start of a new gesture. The visual hint is visualized as a diamond consisting of two triangles (top and bottom) and one hexagon (middle) (see Fig. 6).

The two triangles (second row in Fig. 6) are highlighted in blue when the laser dot enters them. This tells the user that the system expects a SWEEP gesture. The gesture is recognized and validated when the laser dot leaves the two blue triangles and is outside of the bounding diamond. The hexagon (third row in Fig. 6) is colored green when the laser dot enters this region. Similar to SWEEP, a SCAN gesture is recognized when the laser dot leaves the hexagon and is outside of the bounding diamond.

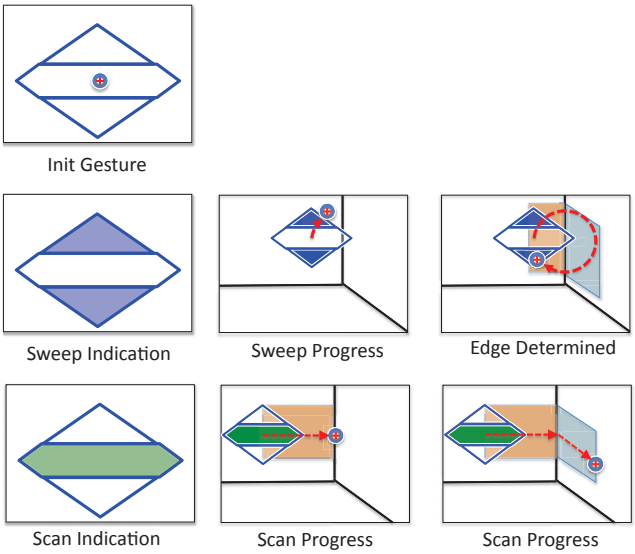

Figure 6: The top row shows the visual hint presented to the user upon triggering a gesture. If the user exits through the top or bottom triangle, a sweep gesture is recognized (second row). If the user exits through either side of the hexagon, a scan gesture is recognized (third row).

Each gesture session includes four stages: Starting a session, determining gesture type, evaluating gesture progress and completion and ending the session. Starting and ending a session is triggered by a button click. We only record laser points of current gesture session for planes estimation when the coverage of the recorded laser points is well conditioned.

5 Geometry Estimation

The output of the user interaction are multiple sets of 3D measurements, each set corresponding to one gesture interaction. To estimate the full environment geometry, we start with a local estimation of the planar geometry for each individual set. The overall model is globally optimized to obtain a consistent geometry. Finally, the full geometry is generated through computing the free space around the user.

5.1 Plane hypotheses from local gestures

For each gesture, we obtain a set of 3D points defined by the laser distance measurements and camera rotation. In this first step, we segment the set into points belonging to the same planes and estimate the plane parameters. This is a hard problem, because we neither know the number of planes nor the parameters of the planes. Therefore, we use an Expectation-Maximization optimization approach to group laser points into dominant planes, while simultaneously estimating the 3D planes. Furthermore, at each iteration steep, we explicitly prune replicated solutions by merging duplicate planes.

Initial plane hypotheses are generated by using graph-based segmentation [5] separately on both the color information and depth information of the 3D points (see Fig. 7). Each 3D point is projected into the corresponding video frame to generate a 2D location and to compute the average RGB color of a surrounding patch.

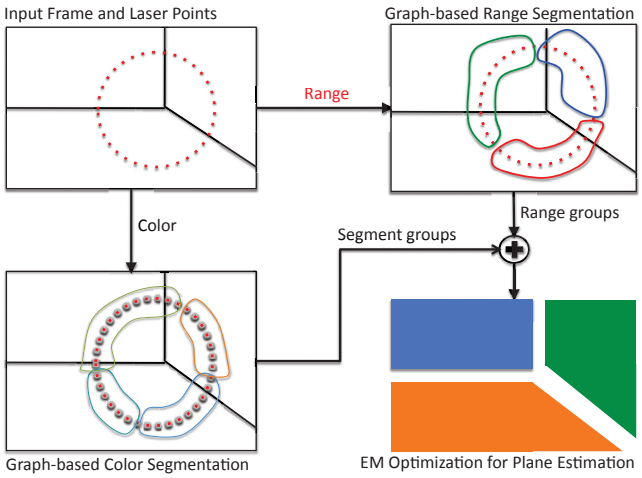

Figure 7: Overview of the flow in our planes segmentation approach.

Then both the color as well as the depth information in the frame are segmented into groups.

For each modality of input (color and depth), we form a graph with the 3D points as nodes. For each 3D point, the five closest neighbors in 3D space are connected with edges. The edges are weighted either with difference in color or distance in 3D space. Then the graph-based segmentation [5] is run on both the color and the depth graph. Each segment in the two graphs is finally used to create a plane hypothesis.

5.2 Optimization for Plane Estimation

The segments from color segmentation and range segmentation are used to form candidate co-planar sets of 3D points. We apply Expectation Maximization (EM) optimization in order to estimate 3D planes while determining the minimum number of planes that best fit the given 3D points. 3D planes are estimated from the 3D point sets and represented as a unit normal vector $\mathbf{n}_k = (a_k, b_k, c_k)$ and the distance to the origin $-d_k$; thus a plane is given as $\Pi_k = (\mathbf{n}_k, d_k)$ or alternatively as a 4-vector $\Pi_k = (a_k, b_k, c_k, d_k)$.

The EM optimization alternates between assigning 3D points to co-planar groups in the E-step and estimating 3D planes in the M-step over several iterations until convergence.

In the Expectation step, we assign 3D points to co-planar groups by computing the likelihood for a point $\mathbf{P}_j$ to belong to one plane Π_i as

$$p(\mathbf{P}_j \in \Pi_i \mid \mathbf{P}_j, \Pi_i) \propto exp\left(\frac{-(\Pi_i \cdot \mathbf{P}_j)^2}{2\sigma^2}\right). \quad (2)$$

At the beginning, we initialize the likelihoods by setting $p(\mathbf{P}_j \in \Pi_i \mid \mathbf{P}_j, \Pi_i) = 1$ if the point $\mathbf{P}_j$ is in the group corresponding to Π_i. After each E-step, the likelihoods are normalized such that

$$\sum_i p(\mathbf{P}_j \in \Pi_i \mid \mathbf{P}_j, \Pi_i) = 1. \quad (3)$$

In the Maximization step, we minimize the objective function in Equation 4 as to maximize the likelihood function in Equation 5. Now, the assignment probability is not changed. Additionally, we explicitly merge duplicate 3D planes in each iteration step of the M-step.

$$O(\Pi) = \sum_{i,j} p(\mathbf{P}_j \in \Pi_i \mid \mathbf{P}_j, \Pi_i) \left\| \Pi_i \cdot \mathbf{P}_j \right\|^2 \quad (4)$$

$$L(\Pi) = \prod_{i,j} p(\mathbf{P}_j \mid \Pi_i) \quad (5)$$

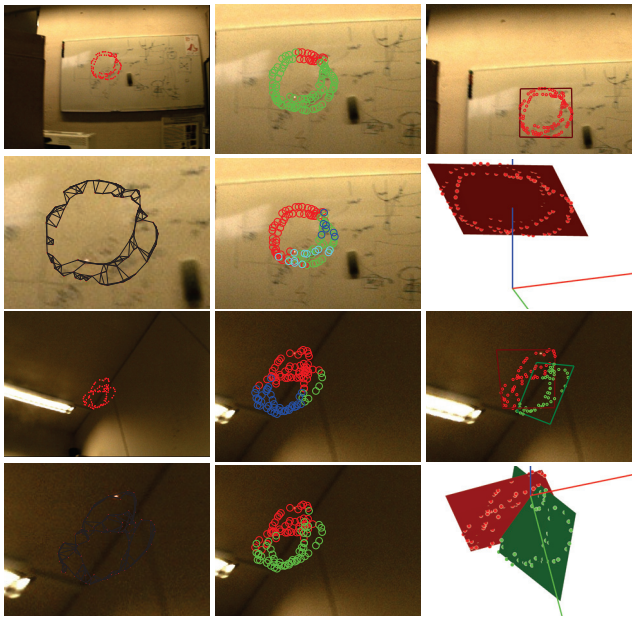

Figure 8: Results of intermediate steps in planes estimation. The first two rows present results of estimation on a plane, while the last two rows are estimation on an edge. The first column displays input laser point observations with corresponding neighborhood network. The second column displays intermediate results of color segmentation (row 1, row 3) and range segmentation (row 2, row 4). The final column shows results of segmented groups and corresponding estimated 3D planes.

For point sets measured in SCAN mode, we add an additional constraint $C(\Pi_i)$ to the Maximization step in equation (4). In this mode, the 3D points are recorded close to an arc on the sphere, resulting in the points being close to a single plane already. As we assume a horizontal scan, we additionally require the planes to be vertical and the plane normals therefore to be normal to the gravity vector G. Thus the constraint is

$$C(\Pi_i) = \left\| \mathbf{n}_k \cdot G \right\|^2, \quad (6)$$

and the overall objective function becomes

$$O(\Pi) = \sum_{i,j} p(\mathbf{P}_j \in \Pi_i \mid \mathbf{P}_j, \Pi_i) \left\| \Pi_i \cdot \mathbf{P}_j \right\|^2 + \lambda \sum_i C(\Pi_i) \quad (7)$$

5.3 Global Nonlinear Refinement

The estimates obtained from the local gesture-driven point sets are describing the local geometry, but are not accurately registered in a global frame due to errors in the panoramic tracking approach. Any translation motion of the cameras is translated into additional rotation leading to a linear drift of the rotation estimation with increasing angle from the start orientation. Furthermore, the geometry estimation did not take any of the 2D-2D measurements between camera frames into account. To correct for any errors induced by the panoramic tracking approach and to make use of all information in the system, we further apply a global nonlinear refinement step that combines all available observation data and global geometric constraints on the plane estimates.

We integrate the following information into our global estimation:

- 3D point measurements which must lie on estimated planes

- 2D-2D observations between keyframes from orientation only tracking
- Constraints between estimated planes including orthogonality and parallelism

We estimate the following parameters simultaneously using a global cost function:

- The 6DOF $\mathbf{T_i} \in \mathbb{SE}(3)$ pose of all keyframe cameras and of all frames where a 3D point was measured
- The 3D locations $\mathbf{W}_j$ of all 2D points observed between keyframes
- The plane parameters Π_k for all planes

The overall cost function thus consists of a data term and a constraints term:

$$C_{all} = C_{data} + C_{constraints}. \tag{8}$$

The data term describes the re-projection error for the unknown 3D locations $\mathbf{W}_j$ and the plane-point distances for the measured 3D points $\mathbf{P}_l$ from the assigned 3D planes Π_k:

$$\begin{aligned} C_{data}(\mathbf{T}_i, \mathbf{W}_j, \Pi_k) = \sum_{i,j} \|p_{ij} - \text{Proj}(\mathbf{T}_i * \mathbf{W}_j)\|^2 \\ + w_{laser} \sum_{\mathbf{P}_{lk} \in \Pi_k} \|\Pi_k \cdot \mathbf{T}_i^{-1} \mathbf{P}_{lk}\|^2 \end{aligned} \tag{9}$$

The weight for 3D point measurements w_{laser} was set to 1000 to account for the difference in number of 2D-2D observations vs 3D measurements.

The constraints term comprises parallel plane pairs, orthogonality between plane pairs and co-planarity of estimated 3D points $\mathbf{W}_j$ with planes:

$$C_{constraints} = w_{planes}(C_{parallelism} + C_{orthogonality}). \tag{10}$$

The constraints are determined through comparing the plane normals of all planes found in the local plane estimation. We assume man-made environments with strong preferences for parallel and orthogonal walls and elements. Therefore we use a wide threshold on the plane normals to create pairs for parallel and orthogonal constraints.

Again, we weight constraints on plane pairs w_{planes} with a factor of 1000 to account for the difference in number of 2D-2D observations vs plane constraints.

For parallel planes, the cross product of associated normal vectors should be zero length:

$$C_{parallelism}(\Pi_i, \Pi_j) = \sum_{i,j} \|\mathbf{n}_i \otimes \mathbf{n}_j\|^2. \tag{11}$$

For orthogonal planes, the dot product of the associated normal vectors should be zero:

$$C_{orthogonality}(\Pi_i, \Pi_j) = \sum_{i,j} \|\mathbf{n}_i \cdot \mathbf{n}_j^T\|^2. \tag{12}$$

We do the global nonlinear refinement through applying constraints on planes and on the estimated 3D points. This allows to adjust camera poses and planes to reasonably correct locations; the estimated 3D point features $\mathbf{W}_j$ are optimized as well to represent the epipolar constraints between cameras.

5.4 Space carving

After having points, laser points, and planes determined accurately, we now can derive the syntactic model by a simple space carving approach on the simplexes defined by the estimated planes. In our approach, each plane is defined as an infinite plane, thus dividing 3D space into two half-spaces: a front half-space that contains the camera, and back half-space. Then we enumerate all possible volumes created by intersecting all combinations of front and back half-spaces. For each volume, we decide if it forms part of the free space around the camera or not (see Figure 9).

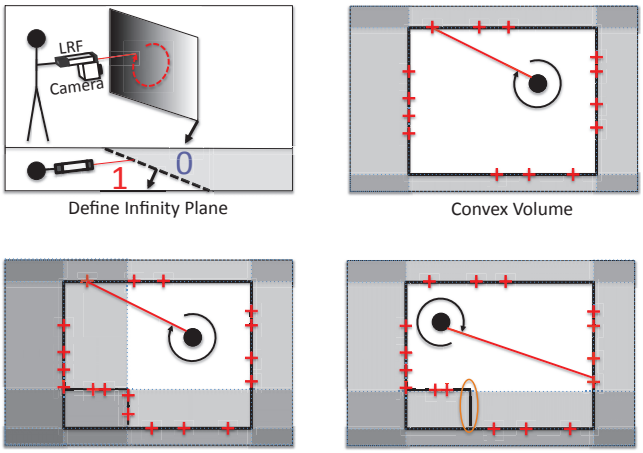

Figure 9: Infinity planes with convex volumes and non-convex volumes.

A volume is valid and part of the free space, if the following conditions hold:

- It is the intersection of at least one front half space. This is equivalent to the camera being in front of at least one plane.
- At least one front facing plane contains 3D points measured by the laser range finder. A robust measure is used to account for inaccuracies in the estimation.

We can enumerate all possible half-space intersections with a simple table where each column corresponds to one plane and can take values of 0 or 1, corresponding to one bit, where 1 represents a front half-space and 0 a back half-space. Then all possible numbers created by the n bits correspond to one volume. Figure 10 shows this concept for 3 planes Π_1, Π_2, Π_3. In brief, the lookup table presents all possible combinations of front and/or back half-spaces, which is analogous to all possible volumes in space defined by the intersections of infinite planes.

This representation allows us to do exhaustive search through all

Π_1	Π_2	Π_3	Volume
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

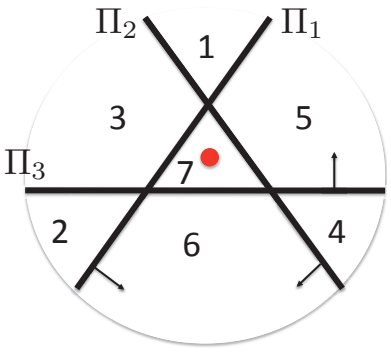

Figure 10: The right side illustrates an arrangement of 3 infinity planes surrounding observing point (red dot). The left side representation of infinity planes (half-spaces) as a lookup table with index column named *Volume*.

Table 1: Results of the synthetic evaluation with plane errors are represented in normal angle errors and mid-point distance errors. The table shows the mean and standard deviation for each error.

Setup	Step	Angle Error	Distance Error
Square	Pre-opt	3.4334 ± 2.5988degree	0.1656 ± 0.1867m
	After-opt	0.2726 ± 0.2195degree	0.0585 ± 0.0606m
T-Shape	Pre-opt	1.4955 ± 1.2260degree	0.1414 ± 0.1533m
	After-opt	0.553 ± 0.320degree	0.1112 ± 0.1246m
T-Shape*	Pre-opt	1.4683 ± 0.9968degree	0.1293 ± 0.1511m
	After-opt	0.5780 ± 0.1873degree	0.0989 ± 0.0724m

Table 2: Results of the empirical evaluation with plane errors are represented in normal angle errors and mid-point distance errors. The table shows the mean and standard deviation for each error.

Setup	Step	Angle Error	Distance Error
Square	Pre-opt	3.2525 ± 3.9310degree	0.5534 ± 0.5442m
	After-opt	0.000307 ± 0.00022degree	0.4836 ± 0.5679m
L-Shape	Pre-opt	5.0664 ± 4.1303degree	0.4437 ± 0.48204m
	After-opt	2.4027 ± 4.1528degree	0.4088 ± 0.5588m

possible convex volumes in space given the number of estimated planes. The final volume is the union of all valid volumes.

6 Evaluation

To evaluate the performance of our system, we devised experiments using both synthetic and real data. For synthetic data, we wish to investigate accuracy of our proposed method under ideal conditions. Additionally, we also want to examine our method in an empirical setup in order to demonstrate real-world robustness in practical use.

6.1 Synthetic Data

To experiment with the proposed method in a controlled environment, we create a synthetic setup to simulate the operation of our system. We avoid errors induced by image processing and rotation tracking by generating a structured point cloud, known camera motion and re-projecting the point clod to obtain correct 2D-3D correspondences. We add Gaussian noise with standard deviation 1 pixel to the 2D observations to simulate simple noise in the system. This allows us to study the performance of the method for different room shapes and measurement methods.

Figure 11 shows an overview of the simulated room-like structures from our simulation. We model a room using a floor plane as polygon in a defined world coordinates, wall height, number of 3D point features per wall, the center of camera rotation in the world coordinates, offset from the camera to the center of rotation to simulate the approximation errors induced by rotation-only tracking, and set of gestures for measurement. Laser measurements were computed by intersecting the single-point laser with the visible wall. The laser center was defined to lie at the camera center and the laser direction is the same as the camera principal axis.

For instance, Figure 11 shows simulated square room and an inverted T-Shape room where the center of rotation is at the center of the rooms with a height of $1.7m$ from the ground plane. In both simulated rooms, we set the camera offset to center of rotation to $0.2m$. New keyframes are added when the camera rotated by more than a preset threshold (30 deg) from all known keyframes. 3D points are randomly distributed on each wall of the rooms. We sample laser points on the walls corresponding to the given sets of scan gestures in the input.

In these simulated rooms, we use either only vertical walls SCAN gestures or only SWEEP gestures to test the different measurement methods. Although the structures look radically simplified, the problem of reconstructing 3D structures from narrow baseline still remains challenging.

To compare the estimated geometry with the ground truth, we computed the following error measures between the walls in the estimation and ground truth. We transform the ground-truth geometry to the reference frame of the estimated geometry and associate each wall in the estimated geometry with a ground-truth wall. Then we define an angular error as the angle between the plane normals of the estimated and ground-truth wall, and a distance error as the distance from the mid-point of the ground-truth plane to the estimated plane along the ground-truth normal.

Table 1 shows the results of our estimation for the three test environments, both as angular and distance errors. The rows show errors before and after global optimization. While the estimates based on the local plane estimation still have errors beyond a few degrees, after global refinement we obtain good results below 1 degree and $10cm$.

6.2 Real Data

To validate the synthetic results and study the performance under real-world conditions, we tested the system with two real rooms. In this evaluation, we selected two room shapes: a rectangular room to compare with the simulation results and an L-shaped room as a more complicated test case. We manually measured the ground-truth geometry of the selected rooms using a the same accurate laser range finder device as in the setup.

We compute plane normal angle errors and mid-point distance errors through first generating ground-truth geometry from the manual measurements of the selected rooms. Then, we transform the ground-truth geometry to the corresponding estimated geometry in order to compute plane errors. To compute the transformation, we map the three best estimated planes that form a corner of the room. Given the corner and the mapped planes, we then compute a 6DOF transformation from the coordinate frame of the ground-truth geometry to the coordinate frame of the estimated geometry.

Figure 13 shows the results of these reconstructions, and our models of the final global refinement converge to the correct geometry. The plane normal angle errors as shown in Table 2 are significantly better than using only panoramic tracking and local estimates. The mid-point distance errors are also reduced after global refinement.

6.3 Qualitative results

Figure 14 presents qualitative results for different rooms, showing the initial model, the constraint result and a texture-mapped rendering. The reconstructed room models represent the correct geometry. This is also confirmed earlier in the quantitative evaluation results. For texture mapping, we first warp keyframe RGB images onto corresponding planes' bounding rectangles. Then we synthesize plane textures with OpenCV blending functions. This results in a visually pleasing textured model, as shown in the right column of Figure 14.

We employ a straightforward texture synthesizing approach through projecting keyframes to corresponding planes and did blending (exposure compensation, seam carving, blending) all together. This raises challenges from duplicate misaligned regions. The misalignment comes from small errors in camera translation estimation which is scale misalignment between real-world unit (from laser-point measurements) and point features (from corner features in video frames). The translation errors come partly from limitation of our selected panoramic tracking approach.

7 Conclusion and Future work

We presented a simple system to quickly capture the geometry and layout of an indoor environment using only a camera and single-point laser range finder. The system rests on several important observations: the combination of a simple tracking method producing drift with global estimation can yield accurate results, by refining

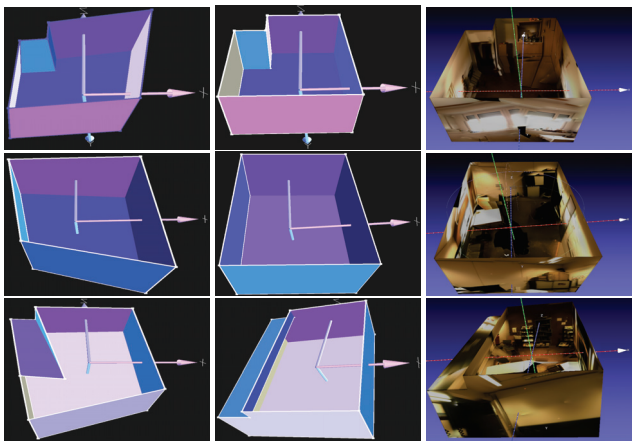

Figure 14: Reconstructed models of three indoor scenarios including: normal living room (row 1), office room (row 2), and seminar room (row 3). The left column shows results that use only the panoramic tracking as input. The middle column shows results using our proposed approach, and the right column displays texture mapped results based on the reconstructed models in the middle column.

the initially perturbed measurements. User input can provide important model information, but the detailed extraction of the model information is left to an automatic optimization. Using background information such as angle constraints in man-made structures leads to more accurate models.

One key contribution is to show how simple user input is turned into syntactic information and numeric estimates. While the user only has to mark or paint a certain structure of interest, this gesture collects the required data to estimate the underlying local geometry from the input. In contrast to other modeling techniques, there is no requirement to trace the outlines or borders of polygons or use explicit modeling operations and modes. At the same time, the interactive input ensures that the user is in control of the features to be modeled, in contrast to a fully automatic system. Overall, we believe that this combination of high-level user input and low-level automated estimation is a promising approach to build systems that are usable while producing realistic results and data.

For future work, we will investigate how to extend the system to more general shapes and geometric relationships. Currently, planes need to be visible at some point, but we plan to extend the volumetric check to include planes that are only inferred from the available data. Furthermore, we would like to be able to model clutter in some way, for example through estimating simple bounding volumes around areas marked-up by the user.

In contrast to heavy computation demand depth sensors based approaches, our current setup enables to perform qualitatively and quantitatively live modeling on affordable consumer mobile platforms. However, rising accessibility to recent depth sensors generates great interests to many enthusiasts including us. Therefore, we also plan to expand our approach to use depth sensors such as Kinect, where depth measurements are densely available. Availability of dense depth maps leads to easier geometric primitives estimation and finer details tuning. Consequently, it also poses hard challenges for clutter handling and redundancies.

REFERENCES

[1] Y. Baillot, D. Brown, and S. Julier. Authoring of physical models using mobile computers. In *Proc. ISWC '01*, pages 39–46, Washington, DC, USA, 2001. IEEE Computer Society.

[2] J. Bastian, B. Ward, R. Hill, A. van den Hengel, and A. Dick. Interactive modelling for AR applications. In *Proc. ISMAR '10*, pages 199–205. IEEE, Oct. 2010.

[3] O. Bau and W. E. Mackay. Octopocus: a dynamic guide for learning gesture-based command sets. In *Proc. UIST '08*, pages 37–46, New York, NY, USA, 2008. ACM.

[4] P. Bunnun and W. W. Mayol-Cuevas. Outlinar: an assisted interactive model building system with reduced computational effort. In *Proc. ISMAR '08*, pages 61–64, Washington, DC, USA, 2008. IEEE Computer Society.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[6] R. Freeman and A. Steed. Interactive modelling and tracking for mixed and augmented reality. In *Proc. VRST '06*, pages 61–64, New York, NY, USA, 2006. ACM.

[7] A. V. D. Hengel, A. Dick, T. Thorm, W. Philip, A. van den Hengel, T. Thormählen, B. Ward, and P. H. S. Torr. VideoTrace : Rapid interactive scene modelling from video. *ACM Transactions on Graphics*, 26(3):86, July 2007.

[8] H. Kim, G. Reitmayr, and W. Woo. IMAF: In-situ indoor modeling and annotation framework on mobile phones. *Personal and Ubiquitous Computing*, 17(3):571–582, April 2013.

[9] T. Langlotz, S. Mooslechner, S. Zollmann, C. Degendorfer, G. Reitmayr, and D. Schmalstieg. Sketching up the world: in situ authoring for mobile augmented reality. *Personal Ubiquitous Comput.*, 16(6):623–630, Aug. 2012.

[10] J. Lee, G. Hirota, and A. State. Modeling real objects using video see-through augmented reality. *Presence: Teleoper. Virtual Environ.*, 11(2):144–157, Apr. 2002.

[11] T. Nguyen and G. Reitmayr. Calibrating setups with a single-point laser range finder and a camera. In *Proc. IROS '13*. IEEE, November 2013.

[12] Q. Pan, G. Reitmayr, and T. W. Drummond. Interactive model reconstruction with user guidance. In *Proc. ISMAR '09*, pages 209–210, Washington, DC, USA, 2009. IEEE Computer Society.

[13] W. Piekarski and B. H. Thomas. Interactive augmented reality techniques for construction at a distance of 3d geometry. In *Proceedings of the workshop on Virtual environments 2003*, EGVE '03, pages 19–28, New York, NY, USA, 2003. ACM.

[14] W. Piekarski and B. H. Thomas. Augmented reality working planes: A foundation for action and construction at a distance. In *Proc. ISMAR '04*, pages 162–171, Washington, DC, USA, 2004. IEEE Computer Society.

[15] A. Sankar and S. Seitz. Capturing indoor scenes with smartphones. In *Proc. UIST '12*, pages 403–412, New York, NY, USA, 2012. ACM.

[16] G. Simon. In-Situ 3D Sketching Using a Video Camera as an Interaction and Tracking Device. In *Proc. Eurographics '10*, Norrk{ö}ping, Sweden, May 2010.

[17] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3D architectural modeling from unordered photo collections. *ACM Transactions on Graphics*, 27(5):1, Dec. 2008.

[18] A. van den Hengel, R. Hill, B. Ward, and A. Dick. In situ image-based modeling. In *Proc. ISMAR '09*, pages 107–110, 2009.

[19] J. Wither, C. Coffin, J. Ventura, and T. Hollerer. Fast annotation and modeling with a single-point laser range finder. In *Proc. ISMAR '08*, pages 65–68, Washington, DC, USA, 2008. IEEE Computer Society.

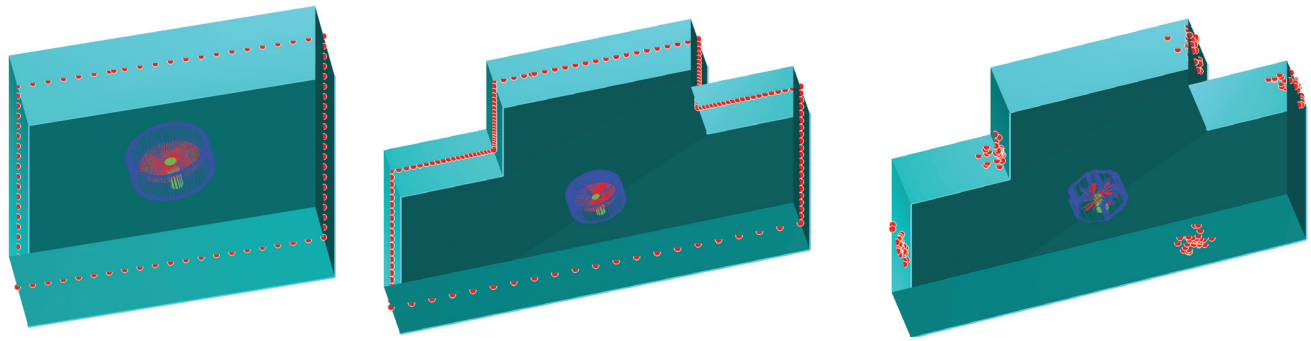

Figure 11: We simulate full panoramic tracking for 3 indoor scenarios: a square room, and an inverted T-shape room. The center ring in blue shows the camera poses in the simulated environment. Red crosses on the side walls of the rooms are laser points at the intersections of the single-point laser beam with the corresponding wall. In the left and middle figure, a SCAN gesture was used to measure the room, while in the right figure the input consists of individual SWEEPs at some corners of the room.

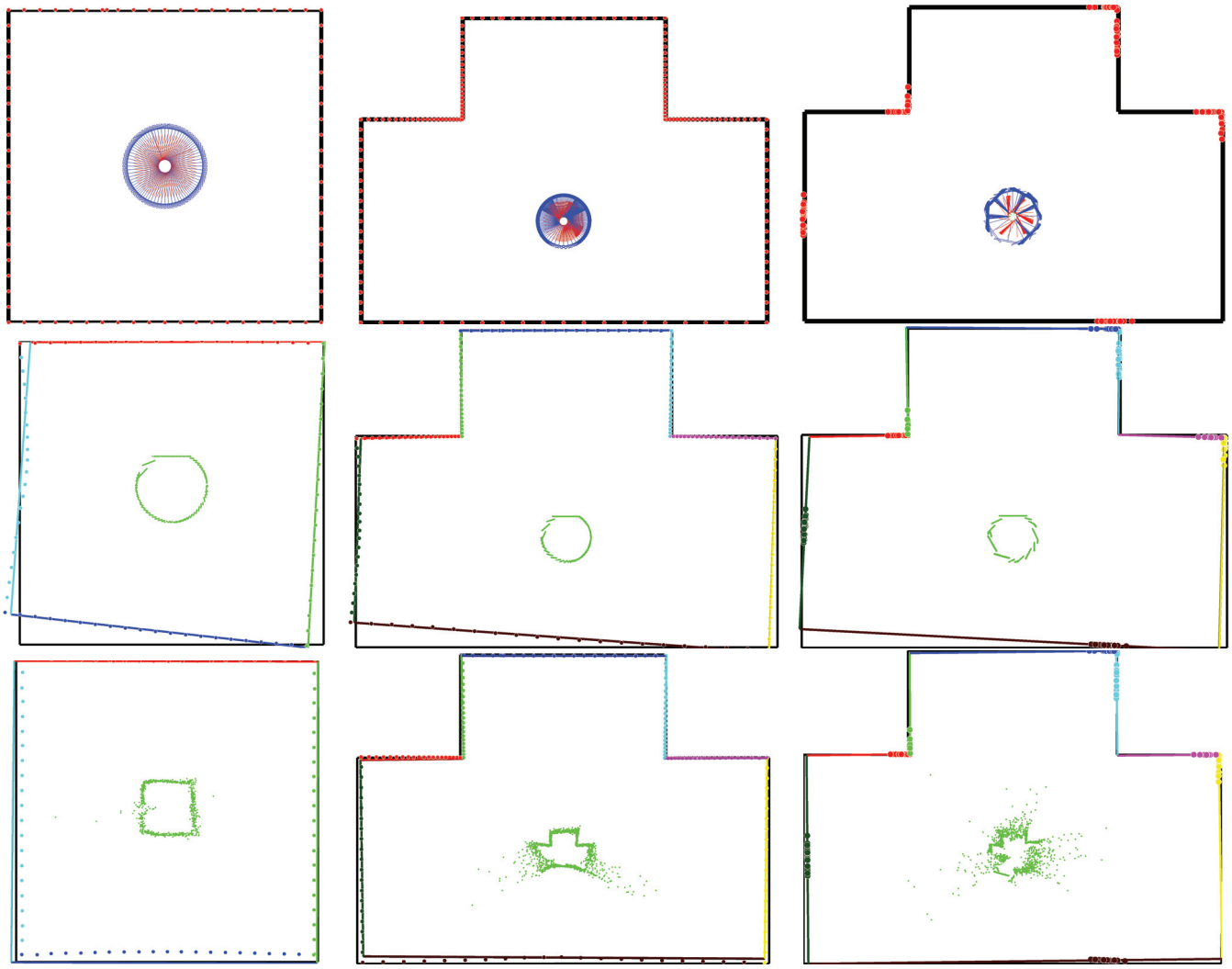

Figure 12: The first row shows the ground-truth geometry from top-down view with laser points in red, reconstructed geometry using purely panoramic tracking (second row) and the final after global refinement (third row). In the second and third row, the colored lines illustrate the estimated geometry. Measured laser points are colored according to the corresponding plane they are associated with. The columns show the results for the square room (left), the inverted T-shape with SCAN measurements (middle) and inverted T-shape with SWEEP measurements (right).

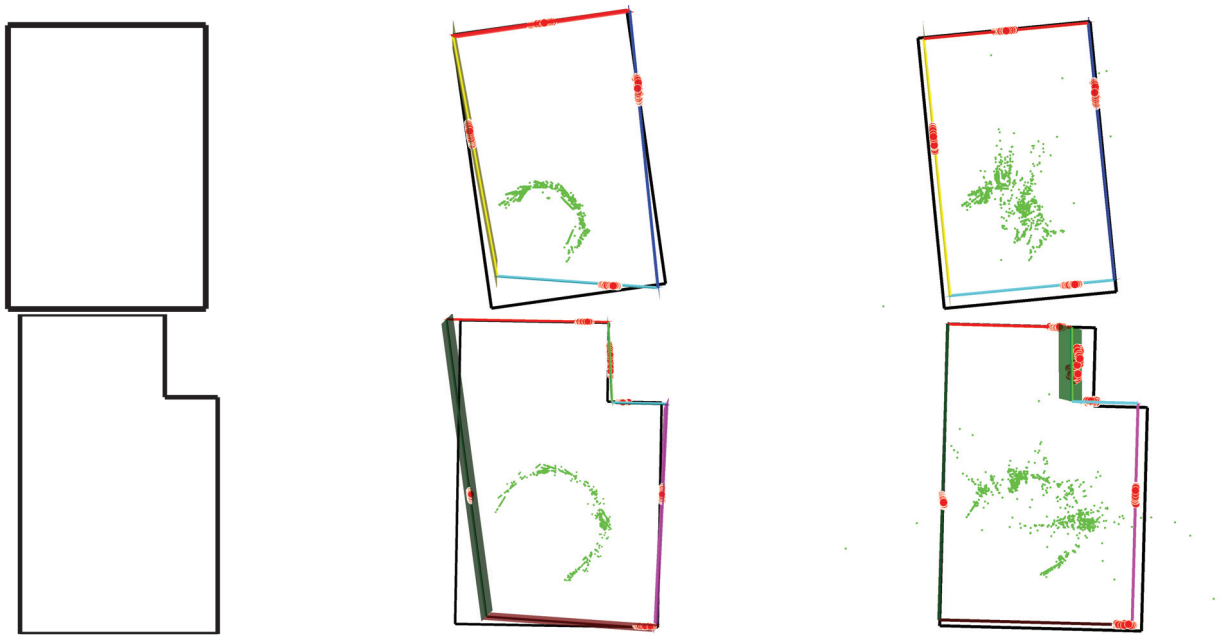

Figure 13: Similarly to the synthetic evaluation, he columns show the ground-truth geometry from top-down view (left) with laser points in red, reconstructed geometry using purely panoramic tracking (middle) and the final after global refinement (right). In the middle and right column the colored lines illustrate the estimated geometry. Measured laser points are colored according to the corresponding plane they are associated with. The rows show the results for the rectangular room (top), and the L-shaped room (bottom).