

Simultaneous Localization and Layout Model Selection in Manhattan Worlds

Armon Shariati¹, Bernd Pfrommer¹, Camillo J. Taylor¹

Abstract—In this paper, we will demonstrate how Manhattan structure can be exploited to transform the Simultaneous Localization and Mapping (SLAM) problem, which is typically solved by a nonlinear optimization over feature positions, into a model selection problem solved by a convex optimization over higher order layout structures, namely walls, floors, and ceilings. Furthermore, we show how our novel formulation leads to an optimization procedure that automatically performs data association and loop closure and which ultimately produces the simplest model of the environment that is consistent with the available measurements. We verify our method on real world data sets collected with various sensing modalities.

I. INTRODUCTION

This paper describes a novel approach to Simultaneous Localization and Mapping (SLAM) that leverages the Manhattan structure of the environment and reformulates the reconstruction task in terms of a model selection problem. Importantly, the key subproblem of establishing long range correspondences and loop closures is incorporated into the reconstruction procedure and solved automatically as part of the optimization procedure.

While there exist several SLAM approaches that try to incorporate the rectilinear structure of indoor man-made environments within their models by tracking more semantically meaningful features such as lines and planes [1], [2], to our knowledge, we are the first to frame the mapping aspect of the problem entirely as one of model selection. Our ideal model is one that would resemble an architect’s blueprint which outlines the location of all large static *layout structures*, namely walls, floors, and ceilings. Such a generative model, even partially complete, could not only enable a robot to track its position and orientation within the environment with greater precision, but could also serve as a strong prior for occupancy inference, as well as object detection and completion. We demonstrate on real data that our novel formulation, based on the principle of Occam’s razor (i.e. select the simplest model of those that best describe the data), produces an optimized trajectory and a compact map representation, jointly.

II. RELATED WORK

This work aims to simultaneously address several problems which collectively intersect a few broader research topics in robotics and computer vision. At the heart of our system is an optimization over a series of robot poses and landmarks, which clearly places it squarely in the SLAM

domain. On the other hand, our contributions to automated layout-model selection extends its relevance to loop closure and automated floor plan generation.

A. Simultaneous Localization and Mapping

The SLAM problem has been a central concern in robotics research for well over two decades. While space constraints prohibit a comprehensive review, the dichotomy highlighted in [3] between non-structural and structural SLAM systems is quite useful when considering where the contributions of this paper best fit.

The more popular and mature non-structural SLAM systems such as [4], [5], [2] prioritize generalizability across environments at the cost of accuracy as they refrain from introducing any hard constraints between landmarks and poses, and among landmarks themselves. Meanwhile, the recently popular structural SLAM systems such as [6], [1], [7], [8] leverage structural cues in order to provide geometric constraints, which can be used to improve accuracy at the cost of universality. Our present work falls in the latter category.

The works most closely related to our own among this body of literature would be that of Kim *et al.* [9], [10]. The first tries to use structural regularities to reduce drift by decoupling estimates for rotation and translation, much like we do, while the second involves a nearly identical initial approach and output, and focuses on the computational limitations of previous approaches using a Bayesian filtering based technique. However, neither algorithm seeks to address the issues surrounding data association and automatic model selection. Similarly, the work of [1] also bears some resemblance to our own in terms of an approach and the tools they use, yet they too overlook the challenge of compact map synthesis and loop closure.

Semantic SLAM [11] is a relatively new body of work which shares many of our own objectives in a more general context. However, a key component of our approach is the assumption of underlying rectilinear structure which we explicitly exploit to improve the quality of our reconstruction.

B. Loop Closure

As the results in this paper are also relevant to visual loop closure, it is worth visiting at least a few approaches to the problem thus far. However, a more in depth review of the subject may be found in [12].

What distinguishes our work here from many of the other key-frame based SLAM systems that also perform loop detection and closure, such as [5], is that our system is

¹GRASP Lab, University of Pennsylvania, Philadelphia PA 19104. {armon, pfrommer, cjtaylor}@seas.upenn.edu

able to solve the loop detection problem and the belief-update problem in a single optimization framework instead of relying on separate modules. While this combinatorial optimization problem is NP-hard we are able to reformulate the search as a convex optimization problem which makes this approach feasible.

Latif *et al.* [13] also observed that sparsity can be leveraged in the context of loop closure. However, while their approach is effectively a search through a set of correspondences to find one sparse set of basis vectors for reconstruction, we invert the problem by trying to instead maximize the total number of correspondences, which allows us to solve the more general problem of data association across temporal frames in addition to loop closure.

C. Automated Floor Plan Generation

Constructing a meaningful representation of an indoor environment is at the heart of automated floor plan generation [14]. While we may share many of the motivations behind the works within this body of literature, our problem domain is that of robotic exploration. As a result, the complexity of many of the models used in these approaches are still greater than is necessary, and furthermore, entirely overlook the localization aspect of the SLAM problem.

Among the more complex and expressive models, are those described in [15] and [16], both of which lift the Manhattan World assumption in order to output a set of watertight polyhedra representing the boundaries of rooms in the environment. These techniques also focus on volumetric labeling of the space.

While both [17] and [18] present solutions for use on mobile platforms, Liu *et al.* take a learning approach to the problem, whereas the authors of [18] leverage the potential of user intervention.

Another approach, relying on image panoramas alone can be found in [19], which formulates the floor plan reconstruction as a shortest path problem.

III. TECHNICAL APPROACH

In this section we describe the main features of our structural analysis procedure. To ground our subsequent discussion we begin with a brief description of the input data and the hardware systems used to acquire it in our subsequent experiments. Figure 1 shows an example of one of our sensor rigs. Despite slight differences among sensor configurations, every rig features a stereo pair of cameras, hardware synchronized to an inertial measurement unit (IMU), as well as a depth sensor that captures low-resolution depth images up to a range of about 6 meters. The data from the stereo cameras and IMU are used to drive a stereo based MSCKF Visual-Inertial Odometry (VIO) system [20]. Further details surrounding each sensor are provided in Section IV.

The end result is that our sensor suites provide the analysis algorithm with a set of depth maps along with initial estimates for the relative motion of the sensor rig over time

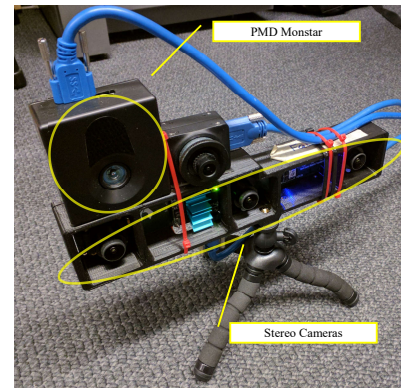


Fig. 1: Sensor rig used to acquire data. Annotated in yellow are the PMD Monstar depth sensor and a custom stereo pair. The camera to the right of the Monstar and the center stereo camera are not used.

and an estimate for the direction of the gravity vector in each frame.

We note that similar datasets could be acquired using other means. One could acquire depth maps using a LIDAR sensor like the Velodyne puck or from a passive stereo system. Similarly pose information could be derived from monocular Visual-Inertial Odometry or from wheel encoders on a moving platform. The proposed analysis would still be applicable in all of these cases.

The first stage in the analysis involves processing each frame in the depth map separately to extract salient axis-aligned planar fragments of layout structure, called *layout segments*. The first step in this process involves projecting the depth points into the plane defined by the measured gravity vector and then rotating the resulting 2D point set in one degree increments to find a yaw orientation that minimizes the entropy of the resulting point distribution. This algorithm is described in a number of previous works including [21], [22], and [23] where it is referred to as an entropy compass. Upon completion, this procedure recovers the orientation of the frame with respect to the prevailing Manhattan structure. We note that one could also use other means for Manhattan frame estimation such as [24], [25].

Once this has been done, the system labels each pixel in the depth map according to the axis alignment of the surrounding $k \times k$ patch, and then groups them together using a connected components procedure as shown in Figure 2. Finally, using an inverse perspective projection, each cluster of pixels is projected into the aligned sensor frame as a point cloud where we fit an orientation-constrained planar model using RANSAC. Planar segments with insufficient extent are discarded to favor the detection of dominant structures.

In addition to the entropy compass procedure which is applied to each frame individually, the system has an estimate for the relative orientation between each frame derived from the visual-inertial odometry system. These two sources of information are fused to provide a final estimate of the orientation of each frame in the sequence. The relative yaw estimates from the VIO system are used to constrain the

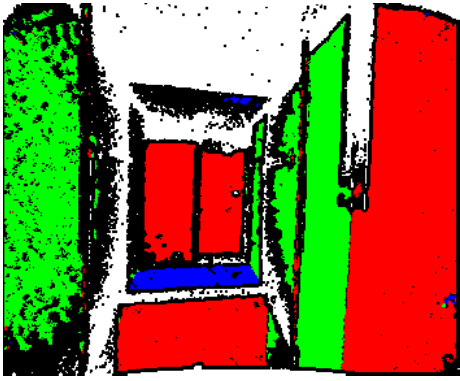


Fig. 2: Depth map broken into salient surfaces. Red, green, and blue pixels represent x , y , and z -axis alignment. A pixel p is assigned to the major axis which maximizes the number of pixels in its $k \times k$ neighborhood that would reside on the plane centered at p with the given major axis orientation. If no axis can be assigned with sufficient confidence or no depth information is recorded at p , it is colored black and white respectively.

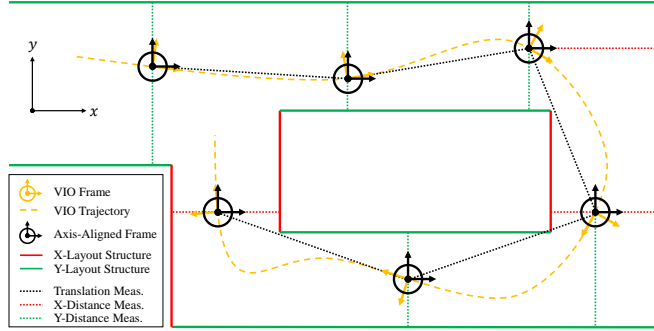


Fig. 3: A 2-dimensional geometric representation of our model which illustrates a sensor moving through a Manhattan environment making periodic range measurements to various layout structures. Solid lines correspond to layout structures, while dotted lines correspond to measurements. Each distance measurement to a particular layout structure corresponds to the distance computed to the visible layout segment within the depth map captured at that frame.

range of angles considered in the entropy compass phase and to provide orientation estimates during periods where no axis-aligned surfaces are visible.

The end result of the procedure is described in Figure 3, which shows a top down view of a set of camera frames. Each frame is associated with two coordinate frames of reference, one which indicates the actual orientation of the sensor head and the other indicating an axis-aligned frame derived from the entropy analysis. Each layout measurement, denoting an estimate for the minimum distance between the camera frame and the corresponding layout segment observed at that frame, is depicted by a red or green dotted line.

This system of measurements can be abstracted into the factor-graph [26] shown in Figure 4. Here the circular nodes on top correspond to axis-aligned frame positions while the triangular and rectangular nodes on the bottom correspond

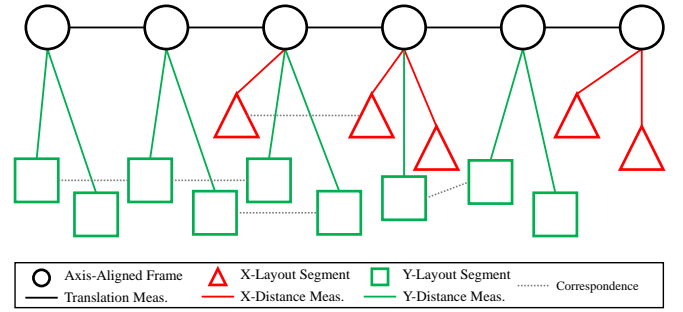


Fig. 4: A functional representation of our model, in two dimensions, as a factor-graph. Circles correspond to robot locations, while triangles and squares correspond to x and y aligned layout segments respectively. Solid lines correspond to measurement factors derived from the VIO, entropy analysis, and depth map processing. We extend the traditional factor-graph formulation by including binary correspondence edges, represented by dotted lines. Initially generated by a temporal analysis, the set of hypothetical correspondence edges is also augmented by a user defined heuristic. Our sparse optimization procedure ultimately determines which of these constraints to enforce and discard.

to layout segments. The links between frames correspond to the estimates for inter-frame motion while the links between the frames and the layout segments correspond to the distance measurements described in Figure 3. Expressing each estimate of inter-frame translation, provided by the VIO subsystem, with respect to the previous estimate of the Manhattan frame, provided by the orientation estimation procedure, yields an axis-aligned and – in principle – drift-reduced trajectory.

In the sequel we will use the following notation to describe the elements of the model shown in Figure 4. Let $\mathbf{p}_i \in \mathbb{R}^3$ denote the position of frame i in the axis-aligned trajectory while $R_i \in \mathbf{SO}(3)$ denotes the orientation of the axis-aligned frame with respect to the corresponding sensor frame. Each of the layout segments that we observe will ultimately be associated with a structural supporting *layout plane*, which is modeled as an axis-aligned surface with infinite extent. Each such layout plane will be modeled with a single parameter. More specifically we will let m_j^x denote the x coordinate of an infinite layout plane with index j that is perpendicular to the x -axis of the model, similarly m_k^y denotes the y coordinate of a y aligned layout plane with index k and m_l^z denotes the z coordinate of a z aligned layout plane with index l .

Correspondences between layout segments are denoted by the dotted lines in Figure 4. These correspondences amount to asserting that two extracted segments lie on the same axis-aligned layout plane. Note that these correspondences would typically link layout segments extracted in different frames but could also link two segments extracted in the same frame. At this stage of the analysis procedure a simple temporal analysis procedure is used to establish correspondences between segments seen in one frame and segments seen in the subsequent frame that have sufficient overlap. This

initial set of correspondences will be augmented with longer range correspondences that are automatically discovered in a subsequent step of the process.

We will let the vector $\mathbf{t}_i \in \mathbb{R}^3$ denote the estimate for the translation between subsequent axis-aligned frames in the sequence that is derived from the visual odometry system and corrected by the orientation estimation procedure. That is \mathbf{t}_i denotes an estimate for the quantity $\mathbf{p}_{i+1} - \mathbf{p}_i$.

We will let ξ denote a vector formed by stacking the free parameters of our model, that is \mathbf{p}_i , m_j^x , m_k^y , and m_l^z , for all i, j, k , and l . Note that we assume that the camera orientations that align the frames with the Manhattan model, R_i , have been estimated using the entropy compass procedure described previously.

In this case the measurement system takes on a particularly simple linear form. Namely for each measurement from a frame to an x -aligned layout segment we have an equation of the form

$$m_j^x - p_i^x = d_{ij} \quad (1)$$

where m_j^x denotes the x coordinate associated with the x -aligned layout plane associated with the layout segment, p_i^x denotes the x coordinate of the position of frame i , and d_{ij} denotes the measured offset between the layout segment and the camera as depicted in Figure 3. Note d_{ij} can be signed depending upon where the frame is relative to the layout segment.

For layout segments aligned with the y axis and z axis we would have exactly analogous equations

$$m_k^y - p_i^y = d_{ik} \quad (2)$$

$$m_l^z - p_i^z = d_{il} \quad (3)$$

As previously discussed, the measurements of interframe motion derived from the VIO system and entropy analysis can be modeled as follows

$$\mathbf{p}_{i+1} - \mathbf{p}_i = \mathbf{t}_i \quad (4)$$

Given this system of measurements the task of finding the optimal estimate for the structure of the scene and the trajectory of the sensor based on the factor-graph simply amounts to solving a sparse linear system $A\xi = \mathbf{b}$ in a least squares sense.

$$\underset{\xi}{\text{minimize}} \quad \|A\xi - \mathbf{b}\|_2 \quad (5)$$

This is simply the system formed by stacking the measurement equations, namely Equations 1, 2, 3, and 4, into a single sparse system. The vector \mathbf{b} aggregates the right hand sides of the equations including the distance measurements, d_{ij} , d_{ik} , d_{il} , and the translation estimates \mathbf{t}_i . This sparse system can be solved extremely efficiently even for relatively large systems of measurements.

Running this procedure yields the result shown in the second column of Figure 7. Each entry shows a result that captures the overall structure of the hallway but also exhibits the kind of drift typically associated with SLAM solutions; an artifact further highlighted in Figure 5. These

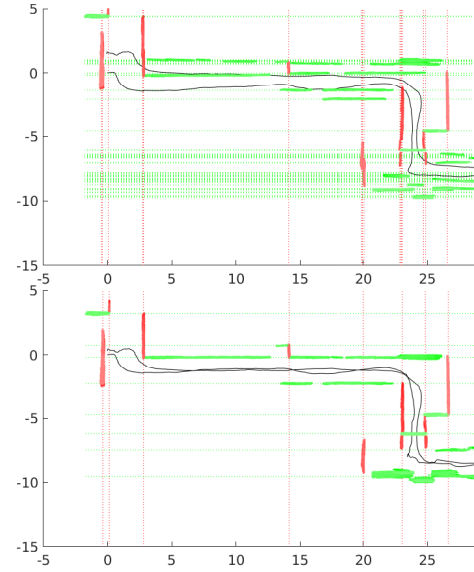


Fig. 5: A birds-eye illustration of how our convex solution (below) can improve reconstruction by eliminating the drift still present in the least-squares solution (above). Notice the reduction in the total number of layout planes, which are denoted with red and green denoted lines corresponding to each axis-alignment.

reconstruction errors stem from the fact that the initial set of correspondences derived from the stream of depth frames is necessarily incomplete. While correspondences derived from frame to frame analysis are typically correct they fail to capture salient long term matches. For example, when one enters then exits a room it is important to encode the fact that walls in the hallway were in fact previously seen and are not new features in the map. Similarly it is entirely possible to encounter a structural wall, then an opening, and then an entirely new section of the same wall. This problem of establishing long range correspondences is exacerbated by the fact that layout structures, unlike visual point feature landmarks, are extended structures and are rarely visually distinctive. Different sections of the same structure can have different appearances in different locations which can frustrate simple techniques that attempt to establish these long range correspondences.

We note that this problem of establishing long range correspondences subsumes the problem of loop closure which also revolves around the issue of deciding that one structure, a wall in this case, corresponds to one that was observed previously.

We propose a novel method that allows us to solve this problem by re-imagining this problem as one of model selection where our goal is to derive the simplest model that is consistent with our observations.

We begin by noting that solutions to Equation 5 suffer from having too many wall surfaces. This is because when a layout structure is encountered again after an intervening break it will be entered again in the map as a new structural layout plane. Our goal then is to discover which of the segments in our overly large model could actually be coincident.

Identifying two or more layout structures with each other effectively reduces the number of parameters associated with the model since all of the displacement parameters associated with that set are collapsed to a single value. In this way we effectively compress the model leading to a simpler solution.

We begin by encoding all of the possible or suspected equivalences between layout segments in a set of equations of the following form

$$m_a^x - m_b^x = 0 \quad (6)$$

As you would expect Equation 6 encodes the idea that the x aligned layout plane with index a and the one with index b are in fact the same. Analogous equations are defined for y and z aligned layout planes.

These possible equivalences can be readily accumulated into a single sparse linear system $E\xi = 0$. Where E is a sparse matrix encoding the relationship and ξ is the vector of model parameters described earlier and used in Equation 5.

One possible approach to generating equivalence hypotheses, is to simply enumerate all possible equivalences between segments which face the same direction (north, east, south, west). However, this approach leads to an unnecessarily large E matrix that contains numerous spurious hypotheses; the effects of which we discuss more thoroughly in Section IV. For now, we adopt the heuristic of enumerating all possible equivalences between segments facing in the same direction that are within some distance of each other. Depending on the length of the path, the expected amount of drift, and the initial number of planes detected, this value can vary between 0.5-3 meters.

At this point we are not sure which of the equivalences are correct and which are false. This leads to a model selection problem. If there are k possible equivalence relations then there are in principle 2^k possible models depending on which of the equivalence relations are enforced, modulo independence issues related to transitive closures among the equivalence relations.

How then can we go about selecting which relations are correct from this exponentially large set of possibilities?

We begin by using the original reconstruction problem as a system that defines a set of possible solutions. We do this by considering the set of ξ values that satisfy:

$$\|A\xi - \mathbf{b}\|_2 \leq \delta \quad (7)$$

where δ encodes the discrepancy between a proposed solution and the available measurements.

One way to choose delta is simply by setting it to

$$(1 + \epsilon)\|A\xi_{\text{lin}}^* - \mathbf{b}\|_2,$$

where ξ_{lin}^* is the optimal value of ξ after solving Equation 5. Alternatively, one can relate δ to the error that one expects in the measurements based on the sensor model. We could also imagine replacing the ℓ_2 norm with the ℓ_1 or ℓ_∞ norms. In each case this inequality defines a convex set in parameter space corresponding to solutions that are sufficiently consistent with the original set of measurements.

We then view our problem as finding the point in the set that maximizes the number of equivalence relations we can satisfy. Note that maximizing the number of equivalences is equivalent to minimizing the number of parameters in the final model, so our goal is to effectively apply the principle of Occams razor to find the simplest model that explains our data.

Formally we can state our goal as follows

$$\begin{aligned} & \underset{\xi}{\text{minimize}} \quad \|E\xi\|_0 \\ & \text{subject to} \quad \|A\xi - \mathbf{b}\|_2 \leq \delta \end{aligned} \quad (8)$$

In this expression, the ℓ_0 norm of a vector simply counts the number of non-zero entries in its input. This problem formulation is reminiscent of the kinds of problems one encounters in compressed sensing.

While this formulation is what we would ideally like to tackle, the discontinuous nature of the ℓ_0 norm makes it intractable so we resort instead to the ℓ_1 norm which we can view as a convex relaxation of our original problem.

Our new goal then can be stated as follows

$$\begin{aligned} & \underset{\xi}{\text{minimize}} \quad \|E\xi\|_1 \\ & \text{subject to} \quad \|A\xi - \mathbf{b}\|_2 \leq \delta \end{aligned} \quad (9)$$

Many may notice the similarity between our formulation and the LASSO procedure [27]. LASSO performs subset selection over model coefficients by forcing as many of them to zero by bounding the sum of the absolute values of regression coefficients. Our approach to model simplification is different as our procedure reduces model complexity by enforcing equivalence relations encoded in the E matrix.

At this point we note that the optimization problem stated in Equation 9 involves minimizing a convex function subject to a convex constraint which places us squarely in the domain of convex optimization. The resulting problem can be reformulated as solving for the optimal value of a linear objective function subject to a set of linear and convex quadratic constraints. We note that we can solve problems involving hundreds of variables in a matter of seconds due to the sparseness of the underlying systems. In our current implementation we formulate and solve this problem in Matlab using CVX.

Once the problem has been solved we examine the resulting vector $E\xi$ and apply a threshold μ to decide which of the equivalences should be enforced. We then re-solve the optimization problem enforcing these equivalences

$$\begin{aligned} & \underset{\xi}{\text{minimize}} \quad \|A\xi - \mathbf{b}\|_2 \\ & \text{subject to} \quad E'\xi = 0 \end{aligned} \quad (10)$$

where E' denotes the reduced set of enforced equivalences. The extent of the new layout structures are determined by computing the boundary around the individual corresponding layout segments residing on the same plane.

We also introduce an additional hard constraint on the trajectory of the sensor relative to the layout segments. For

instance, a measurement d_{ij} to layout segment m_j^x also introduces the following linear inequality constraint

$$-\text{sign}(d_{ij})(m_j^x - p_i^x) \leq 0 \quad (11)$$

This constraint ensures that the recovered model is topologically consistent with the range observations. Accumulating these inequalities yields an additional convex constraint $D\xi \leq 0$, which is added to Equations 9 and 10.

IV. EXPERIMENTAL RESULTS

Several reconstruction experiments were carried out in different areas of our campus. Three different sensor rigs were used in these experiments in order to demonstrate robustness. All of them have in common a custom stereo monochrome camera running at 20 Hz with a $97^\circ \times 81^\circ$ field of view (FOV) [28]. Together with a hardware synchronized inertial measurement unit (IMU), they provide data to drive the stereo VIO algorithm. The primary difference between rigs is the choice of depth sensor and its frame rate. Rig A (used for Areas 1,2, and 3) hosts a PMD Monstar time-of-flight depth sensor, which captures 352×287 resolution frames at 10 Hz and has a FOV of $100^\circ \times 85^\circ$. Rig B (used for Areas 4 and 5) features the same depth sensor, but run at 5Hz. Lastly, Rig C (used for Area 6) replaces the Monstar with an Orbbec Astra structured light camera, which runs at 30Hz and has a resolution of 640×400 and a FOV of $60^\circ \times 49.5^\circ$.

With the exception of the threshold used for enumerating possible equivalences, all computations were carried out using the same set of optimization parameters agnostic to sensor choice or environment. The entropy compass explored a radius of 0.3° around the estimate of angular displacement provided by the VIO system. The value of ϵ used to determine δ was empirically set to 2%. Finally, the threshold value of μ used to conclude equivalence between two segments was set to 30 centimeters. All of our computations were carried out using an Intel i7-8700K CPU with 32GB of RAM.

Figure 7 illustrates a composite of the reconstruction results in each environment using our convex approach, which we compare to reconstructions based on registering layout segments to the axis-aligned frame position they were observed in, as well as the results of the least-squares optimization described in Equation 5.

Different environments presented different challenges. For instance, Area 3 is rather large and contains many classrooms along its eastern wing. As a result, redundant candidate layout structures were generated as we encountered the same walls multiple times upon entering and exiting each classroom, which our optimization then had to consider. Area 6 features a modern architecture with many glass surfaces (embedded even in doors), large open areas, and exposed structural I-beams oriented at various angles. As a result, not only was the entropy analysis and layout segment detection confounded by the actual layout itself, but also by missing and corrupted depth measurements. Importantly, almost all of the examples involve situations where the robot needs to perform loop closures to account for situations where the

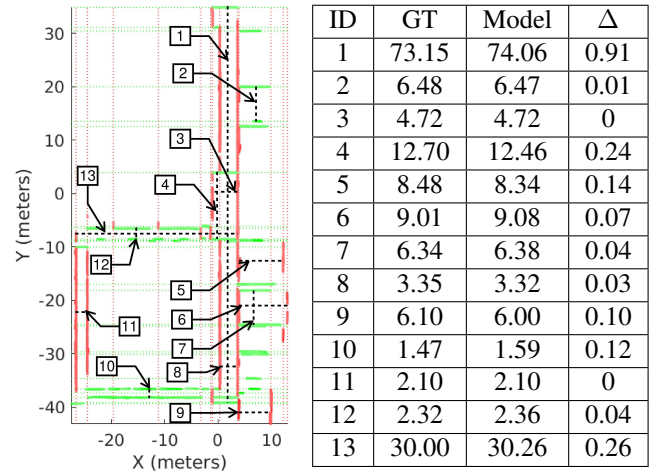


Fig. 6: Comparison of surface to surface distances against ground truth measurements collected with a laser range finder.

same surface is encountered again after a significant interval of time. These loop closures are automatically detected and factored into the reconstruction as part of our procedure.

For each environment, table I shows the number of equivalences considered, the number that were accepted, the number of layout structures in the original model and the number in the final simplified model. It also indicates the computational time required for the analysis and the length of the robot's trajectory. Note that the final optimized model contains far fewer layout planes than the original model.

Table II provides a quantitative analysis of the effects different types of optimization have on the trajectory drift. As the sensor rig is carried back to the starting location after each exploration, the values reported are the distances between the starting point and ending point of the trajectory after reconstruction. Note that in all cases but one, the convex optimization significantly reduced the drift in the reconstructed trajectory.

We do note however that our system is not entirely immune to false data associations due to the use of a hard merge threshold. This can be problematic particularly in more complicated environments, such as Area 6, where multiple independent planes may occur within that threshold in a particular region and the error in the reconstructed trajectory may not be large enough to prevent an incorrect assignment. We are currently pursuing ideas such as weighted hypotheses and adaptive thresholds to address this issue.

Figure 6 provides a quantitative analysis of the distance between selected surfaces in the recovered model compared to ground truth measurements of these distances taken with a laser range finder in Area 3. As these results demonstrate the average reconstruction error in this set of measurements is 1.5%.

V. CONCLUSION

In conclusion, we have demonstrated an approach for generating compact reconstructions of Manhattan environments.

TABLE I: Complexity results in each of the mapped environments

Area ID	1	2	3	4	5	6
Number of Equivalences Considered	151	2734	15631	454	1173	1704
Number of Equivalences Accepted	125	1431	10761	247	1092	1510
Number of Initial Layout Segments in Model	48	239	408	89	108	167
Number of Layout Structures After Analysis	14	34	40	28	25	16
Complexity Reduction %	70.8	85.8	90.2	68.5	76.9	90.4
Optimization Time (s)	0.12	2.02	50.67	0.23	0.79	0.59
Path Length (m)	53	200	249	69	113	67

TABLE II: Drift in meters after each type of optimization

Optim.	Raw VIO	Entropy Compass	Least-Squares	Convex
Area 1	0.67	0.82	0.62	0.16
Area 2	2.36	3.06	1.58	0.39
Area 3	2.24	2.98	1.63	0.19
Area 4	0.46	0.35	0.48	0.09
Area 5	1.32	0.81	0.46	0.15
Area 6	0.73	1.07	0.88	1.11

In scenarios where a reasonable estimate for one's rotation can be inferred from the visible Manhattan structure, we can solve the full SLAM problem using convex optimization. Furthermore, our sparse objective enables us to explore the vast combinatorial space of potential data associations and loop closures, which results in a more accurate trajectory alongside a compact representation of the map. We validate our mapping procedure on a set of representative indoor environments.

REFERENCES

- [1] M. Hsiao, E. Westman, and M. Kaess, "Dense Planar-Inertial SLAM with Structural Constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6521–6528.
- [2] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [3] H. Li, J. Yao, J.-c. Bazin, X. Lu, Y. Xing, and K. Liu, "A Monocular SLAM System Leveraging Structural Regularity in Manhattan World," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2518–2525.
- [4] H. Li, J. Yao, X. Lu, and J. Wu, "Combining Points and Lines for Camera Pose Estimation and Optimization in Monocular Visual Odometry *," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1289–1296.
- [5] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [6] F. Camposeco and M. Pollefeys, "Using vanishing points to improve visual-inertial odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2015-June, no. June, 2015, pp. 5219–5225.
- [7] J. Straub, R. Cabezas, J. Leonard, and I. Fisher, John W., "Direction-Aware Semi-Dense SLAM," *arXiv e-prints*, p. arXiv:1709.05774, Sep. 2017.
- [8] H. Zhou, D. Zou, L. Pei, R. Ying, P. Liu, and W. Yu, "StructSLAM: Visual SLAM with building structure lines," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1364–1375, 2015.
- [9] P. Kim, B. Coltin, and H. J. Kim, "Low-Drift Visual Odometry in Structured Environments by Decoupling Rotational and Translational Motion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7247–7253.
- [10] —, "Linear RGB-D SLAM for Planar Environments," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 1–16.
- [11] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.
- [12] S. Lowry, N. Sunderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual Place Recognition: A Survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.
- [13] Y. Latif, G. Huang, J. Leonard, and J. Neira, "Sparse optimization for robust and efficient loop closing," *Robotics and Autonomous Systems*, vol. 93, pp. 13–26, 2017.
- [14] B. Okorn, X. Xiong, B. Akinci, and D. Huber, "Toward Automated Modeling of Floor Plans," in *Symposium on 3D Data Processing, Visualization and Transmission*, 2010.
- [15] C. Mura, O. Mattausch, and R. Pajarola, "Piecewise-planar Reconstruction of Multi-room Interiors with Arbitrary Wall Arrangements," *Computer Graphics Forum*, vol. 35, no. 7, pp. 179–188, oct 2016. [Online]. Available: <http://doi.wiley.com/10.1111/cgf.13015>
- [16] S. Ochmann, R. Vock, R. Wessel, and R. Klein, "Automatic reconstruction of parametric building models from indoor point clouds," *Computers and Graphics (Pergamon)*, vol. 54, pp. 94–103, feb 2016.
- [17] C. Liu, J. Wu, and Y. Furukawa, "Floornet: A unified framework for floorplan reconstruction from 3d scans," *CoRR*, vol. abs/1804.00090, 2018. [Online]. Available: <http://arxiv.org/abs/1804.00090>
- [18] V. Angladon, S. Gasparini, and V. Charvillat, "Room floor plan generation on a project tango device," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10705 LNCS, 2018, pp. 226–238.
- [19] R. Cabral and Y. Furukawa, "Piecewise planar and compact floorplan reconstruction from images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [20] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight," *IEEE Robotics and Automation Letters (RA-L)*, 2017. [Online]. Available: <http://arxiv.org/abs/1712.00036>
- [21] J. C. Bazin, Y. Seo, and M. Pollefeys, "Globally optimal consensus set maximization through rotation search," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7725 LNCS, no. PART 2, Springer, Berlin, Heidelberg, 2013, pp. 539–551. [Online]. Available: http://link.springer.com/10.1007/978-3-642-37444-9_42
- [22] A. Taneja, L. Ballan, and M. Pollefeys, "Never get lost again: Vision based navigation using streetview images," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, Cham, 2015, vol. 9007, pp. 99–114. [Online]. Available: http://link.springer.com/10.1007/978-3-319-16814-2_7
- [23] A. Cowley, C. J. Taylor, and B. Southall, "Rapid multi-robot exploration with topometric maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, may 2011, pp. 1044–1049. [Online]. Available: <http://ieeexplore.ieee.org/document/5980403/>
- [24] J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher, "Real-time manhattan world rotation estimation in 3D," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. IEEE, sep 2015, pp. 1913–1920. [Online]. Available: <http://ieeexplore.ieee.org/document/7353628/>
- [25] K. Joo, T.-H. Oh, J. Kim, and I. S. Kweon, "Globally Optimal Manhattan Frame Estimation in Real-Time," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 1763–1771.
- [26] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, 2001.
- [27] R. Tibshirani, "Regression shrinkage and selection via the lasso:

A retrospective,” *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 2011.

[28] “Open vision computer,” <https://github.com/osrf/ovc>.

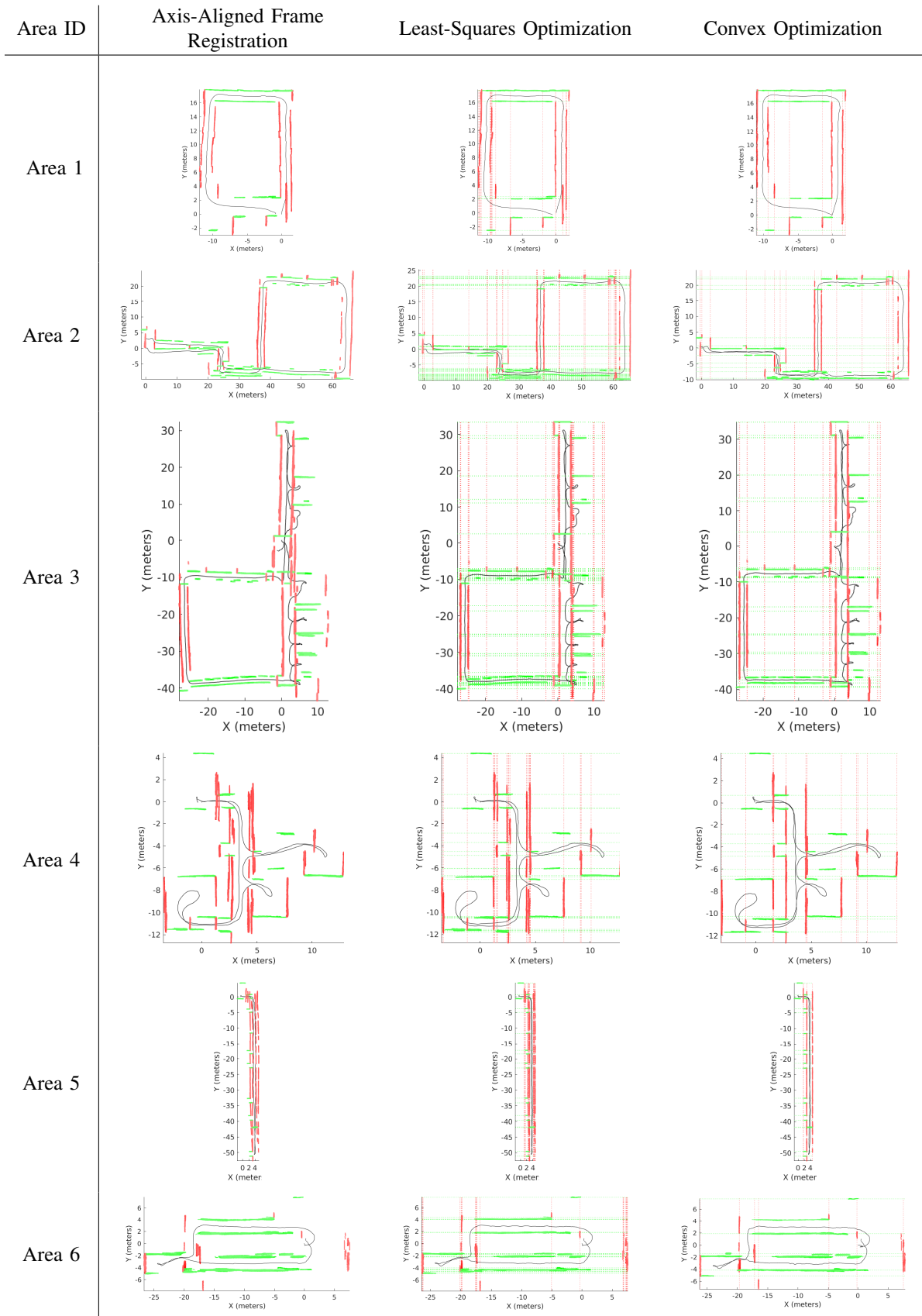


Fig. 7: Birds-eye view of the reconstruction results of our analysis in several Manhattan environments. Each column illustrates the effect of a different reconstruction method while each row corresponds to a different area. Red and green point clouds correspond to x and y -aligned layout segments, which reside on infinite layout planes denoted in each figure with red and green dotted lines. The black curve illustrates the sensor trajectory.