2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Maximally Informative Surface Reconstruction from Lines

Jonas Witt and Gerhard Mentges

***Abstract*— In this paper, we propose a novel multi-view method for surface reconstruction from matched line segments with applications to robotic mapping and image-based rendering. Starting from 3D line segments, plane hypotheses are formed for all non-collinear and sufficiently coplanar segment pairs. The surface that is spanned by two segments is used to immediately prune hypotheses that do not pass a sight line occlusion check to keep the initial plane number tractable. After further merging, exhaustive intersections are computed in an efficient way to yield a maximally informative surface representation. Finally, robustified visibility constraints are used to recover a dense surface mesh that is a pessimistic representation of the free space, which is desirable for path planning applications. The presented system is a complete and automatic solution suitable for mapping an environment in real-time scenarios like robotic exploration. We demonstrate the performance of our algorithm on several indoor scenes with varying complexity.**

I. INTRODUCTION

In many applications, geometric reconstruction from multiple views is an important problem. For image-based rendering and 3D scanning, accurate and visually pleasing models are desired, whereas higher level robotic planning requires concise geometrical representations of an environment. While great progress has been made with feature point based structure from motion (SfM) techniques in recent years, the reconstruction of scenes containing non-textured surfaces is still a major challenge. Many man-made environments do not even provide sufficient feature points to recover the camera motion [1], not to mention surface reconstruction. Even partial lack of texture quickly gives rise to holes with conventional techniques [2]. For robotic planning, this is usually not acceptable, which is why active vision systems (e.g. laser scanners) are mostly employed. However despite the lack of point features, the structural information of sparsely or non-textured scenes is still visually deducible from edges (by which we mean intensity gradient maxima). Using piecewise straight line segments as a compact edge representation, we aim to reconstruct completely untextured scenes which are only partially visible in each frame as is the case during indoor exploration.

This work builds upon the Iterative Closest Multiple Lines (ICML) algorithm from [1] extended by bundle adjustment to yield a complete and fully automatic Line-SLAM system. The resulting map of line segments along with their visibility graph is the input for the presented algorithm. Our surface reconstruction approach tries to leverage the maximal information content from the input data by even incorporating planes that derive from only two line segments (in contrast to statistical consensus of multiple primitives). We generate plane hypotheses from coplanar line segment pairs and immediately check visibility of other line segments in the quad area that the candidate segments span. This early pruning is done to keep the plane count tractable. The end point covariances that are obtained from bundle adjustment are used to assign line segments to hypotheses and subsequently merge planes based on common children and a coplanarity criterion. The intersection of the remaining planes yields a three-dimensional grid for which every face is initially assumed to be solid. The free space is determined by rubustified visibility constraints that the sight lines from different camera poses to the line segments impose.

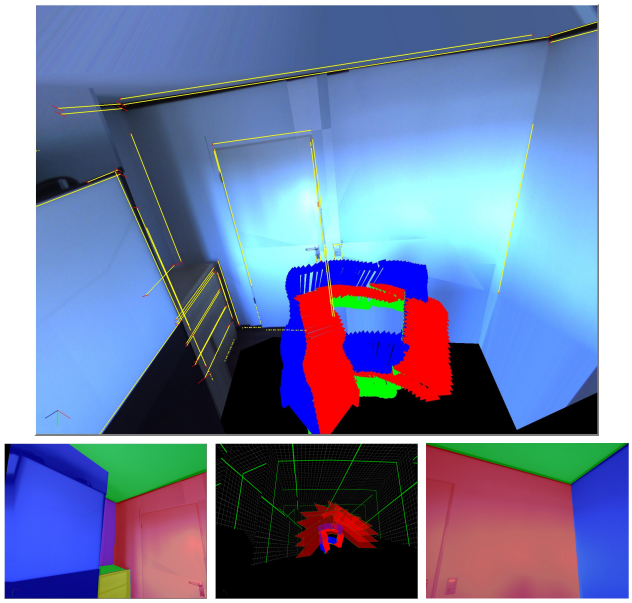
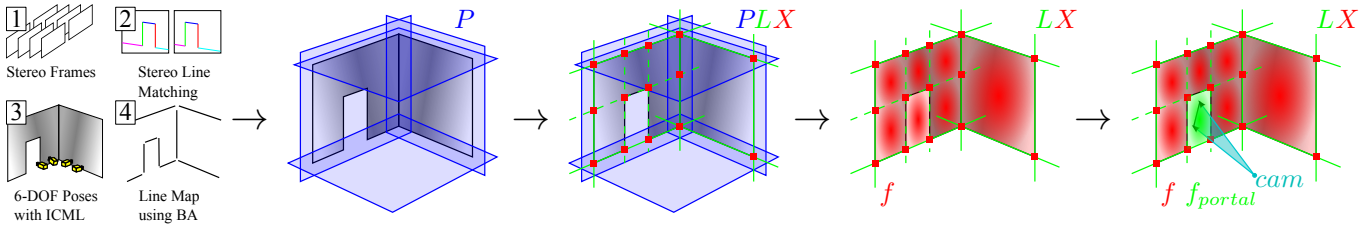


Fig. 1. The Room sequence has virtually no texture, large areas with specularities and all frames have only partial scene visibility. The image row shows the pose/frustum plot (center) and two input images that have been colored based on the normal direction of reconstructed planes. The upper image shows a rendering of the complete reconstructed scene with overlayed camera poses (blue arrow is the look at direction) and detected line segments (with error cuboids in red).

Our main contributions are three-fold. First, we propose a novel method for generating plane hypotheses from line segments in N views that is capable of recovering planes from just two coplanar line segments while not creating an intractable number of false hypotheses. This enables the reconstruction of completely untextured scenes for which surfaces may only be partially visible. Secondly, an efficient plane intersection scheme based on a scene graph is defined by several simple rules: 1. intersections are created for all

J. Witt and G. Mentges are with the Institute for Reliability Engineering, Hamburg University of Technology, 21073 Hamburg, Germany, email: jonas.witt@tuhh.de, gerhard@mentges.eu



(a) Stereo Line SLAM ⇒ ICML [1] is used for inter-frame line association. Bundle adjustment creates the final maps.

(b) Plane Search ⇒ Planes P for every surface containing ≥ 2 coplanar input segments.

(c) Plane Intersection ⇒ Interconnected grid of intersection lines L and points X.

(d) Face creation ⇒ All solid mesh of faces f across intersection grid, free space may be filled up.

(e) Occlusion Check ⇒ Classification of faces into solids and permeable portals. Output: surface mesh (polygons).

Fig. 2. The shown reconstruction pipeline visualizes all steps, starting from the preceding Line SLAM which outputs 3D line segments (with error estimates), camera poses and sight lines. Subsequently, planes are found among the input segments, for which an intersection grid is created based on line segment relations. Finally, faces are computed from the intersections and free space faces are culled based on the visibility constraint (using sight lines).

common line segments between planes, 2. line segments have to be enclosed by intersection points which is enforced by auxiliary intersections that 3. are found in planes that fulfill a spatial or graph-based adjacency criterion. Thirdly, we propose a robust occlusion score for convex faces and the sight triangle that is spanned by a camera origin and line segment end points. This score considers lateral and depth penetration severity to determine the navigable free space from imperfect measurements.

Accordingly, due to the resulting concise geometric representation, the high efficiency that enables real-time use and pessimistic assumptions about the free space, our method is especially useful for robotic exploration.

II. RELATED WORK

A frequently chosen approach to planar reconstruction is plane sweeping [3], [4], [5], [6], where plane hypotheses are generated at locations of high feature density along a limited and more or less constrained set of sweeping directions.

In [3] the camera image plane is swept across the scene and at every depth location where features were found, a Delaunay triangulation is performed. Because of the triangulation, reconstructed geometry is only indirectly related to the scene geometry and more importantly reconstruction of every plane in the scene requires a large number n of different perspectives, convergence is shown for $n \to \infty$.

Another approach is to sweep across principal directions of the scene. One way is to assume a Manhattan world as in [5], [4], [7] with obvious limitations considering scene geometry.

More flexibly, non-orthogonal vanishing directions serve as sweeping directions in [6]. However, underrepresented vanishing directions may be challenging. Furthermore plane hypotheses defined only by parallel lines are ignored, which can lead to problems in some environments (e.g. corridors).

Generally speaking, plane sweeping techniques impose rather strong restrictions to the solution space and it is difficult to determine meaningful sweeping directions while not missing relevant but under-implied ones (e.g. untextured surfaces of which only few edges are detectable).

In [8], junction points are searched among input line segments to generate plane hypotheses from connected line pairs. Faces are only created across looping sequences of junctions and thus many faces of the scene may be missing in the reconstruction, since not every junction or even line may be visible or detectable.

In [9], input images are divided in planar and non planar areas. For planar reconstruction, a RANSAC based scoring method is employed to fit arbitrarily oriented plane hypotheses to dense depth input point data. However, such input may not be available in untextured scenes.

An approach that leverages image-space neighborhood of edge pixels for generating plane hypotheses is presented in [10]. Seed points are distributed across the input image according to a Voronoi diagram of the detected edges, for which 3D coordinates have been determined in a previous SLAM step. For each seed point, RANSAC is used to select the best hypothesis for its supporting edge pixel neighborhood. However, it seems that a fixed threshold irrespective of error estimates is used for hypothesis scoring which is problematic for surfaces in a larger range of distances. Multiple views are reconstructed individually and then stitched together. This makes a reconstruction of textureless regions at image borders difficult.

Our approach leverages information from multiple views while considering individual line segment error covariances. Plane intersections provide consistent and concise geometry without unnecessary holes. The use of line segments as sparse input allows our method to be fast enough for real-time applications.

III. ALGORITHM

The motivation behind our algorithm is that line segments are concise and geometrically expressive features with which many man-made scenes can be described. The number of planes that can be derived from n_S line segments is $O(n_S^2)$ while from n_p points, $O(n_p^3)$ planes are possible. Pairs of line segments even allow to immediately assess coplanarity as a plane likelihood criterion, whereas any 3 points are coplanar. Another beauty lies in the hinting towards 3D edges. We utilize these hints to compute plane intersections based on common line segments between different planes to extract volumes. Fig. 2 gives an overview of our pipeline

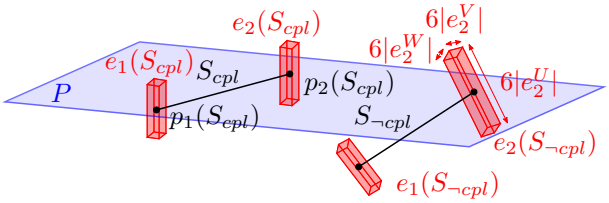


Fig. 3. A line segment S is considered coplanar to a plane P, if both error volumes $e_{1,2}(S)$ of its end points $p_{1,2}(S)$ intersect with P.

from generating plane hypotheses, over plane intersection (based on line segments), the extraction of faces from the intersection grid and finally culling faces that do not pass our robustified visibility constraint.

A. Line Segments and Coplanarity

Each line segment S detected by the Stereo-SLAM is defined by its end points $p_1(S)$ and $p_2(S)$ in Euclidean coordinates and corresponding error description. For simplification, the errors of the segment end points are represented by bounding cuboids $e_{1,2}(S)$ around the Gaussian error ellipsoids, spanned by the error bases $\{e_{1,2}^U, e_{1,2}^V, e_{1,2}^W\}$ which correspond to the eigenvectors of the point covariance matrices estimated during the SLAM optimization procedure. The resulting error cuboids are scaled to 3σ in each direction to obtain volumes, which contain the ideal points to a high certainty of 99.7%.

For a line segment S to be considered coplanar to a plane P, the condition

$$cpl(S,P) = (e_1(S) \cap P \neq \emptyset) \wedge (e_2(S) \cap P \neq \emptyset) \qquad (1)$$

has to be fulfilled, i.e. both of the described error cuboids $e_{1,2}(S)$ have to intersect with P. Accordingly, two segments $S_{1,2}$ are called coplanar, if

$$cpl(S_1,S_2) = \exists P : cpl(S_1,P) \ \wedge \ cpl(S_2,P). \qquad (2)$$

When searching for plane hypotheses among combinations of segments, first, a provisional normal $n' = dir(S_1) \times dir(S_2)$ is computed, where $dir(S)$ denotes the normalized segment direction. The error base vectors of each segment point are then projected onto n' to obtain the error $e^{n'} = |e^{U,n'}| + |e^{V,n'}| + |e^{W,n'}|$ in direction of n'. The reciprocal square of the projected errors $e^{n'}$ is now used to perform a weighted least squares fit of a plane to the four segment points of the combination. If the fitted plane intersects all error cuboids, a coplanar segment combination and thus a valid plane hypothesis is found.

Prior to the coplanarity check, segment combinations are checked for collinearity using a similar projection of the error bases onto the relevant end point distance vectors, and checking the error projections for overlap, which implies collinearity. If a combination S_1, S_2 is collinear, the condition $cln(S_1,S_2)$ holds true. Collinear segments are not considered when checking for coplanarity to ensure geometrical stability of the constructed plane hypotheses.

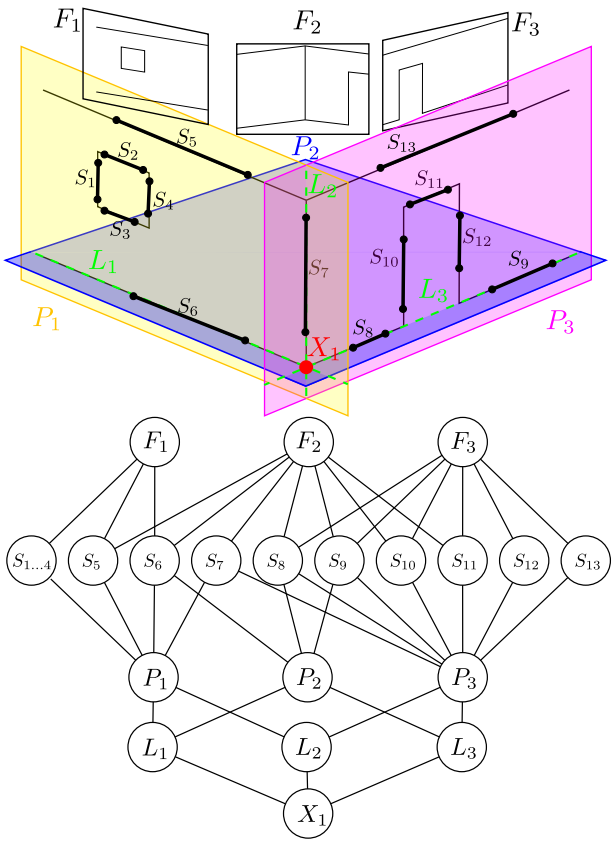


Fig. 4. A simple scene with three camera frames $F_{1...3}$ from which 13 line segments $S_{1...13}$ are visible. The planes $P_{1...3}$ generate the intersection lines $L_{1...3}$ and intersection point X_1 (note that the ceiling plane that S_5 and S_{13} would generate has been omitted for visual clarity). The graph represents the resulting relations of visibility and geometry, which are created and assessed during reconstruction.

Line segments S coplanar to a plane P are called the *child segments* of P, represented by the set

$$cld_S(P) = \{S \mid cpl(S,P)\}. \qquad (3)$$

Correspondingly, for each segment S a set $pnt_P(S)$ of *parent planes* is defined, containing all planes to which S is coplanar.

B. Frames and Visibility

For each frame F, $cld_S(F)$ defines a set of the visible segments as obtained from the Line-SLAM. Analogously, for each segment S a set $pnt_F(S)$ of *parent frames* is defined, which includes all frames F, wherein S has been observed. From this, we derive a set of parent frames

$$pnt_F(P) = \bigcup_{S \in cld_S(P)} pnt_F(S) \qquad (4)$$

for each plane P, which is the union of the child segments' parent frames.

The sets $cld_S(F)$ and $pnt_F(P)$ form a graph (Fig. 4), which is used to constrain interference of geometry by visibility. $pnt_F(P)$ has the important property of providing information about overlapping geometry between frames and thus about regions where frame-wise geometry has to be seamed (Fig. 4), which is e.g. used to restrict merging and

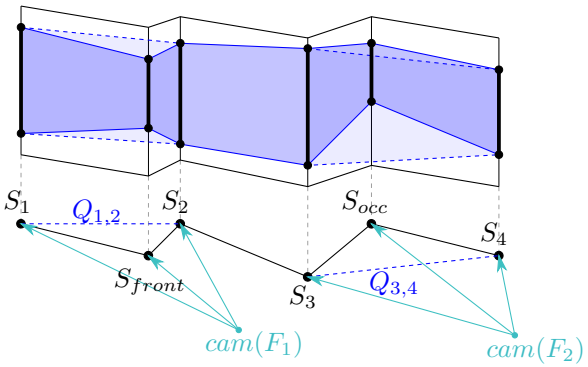


Fig. 5. Immediate occlusion and contour check for plane hypotheses. The false hypothesis defined by S_1, S_2 crosses solid volume and fails the contour test because its spanning quad $Q_{1,2}$ gets occluded by S_{front} and $cpl(S_1, S_{front}) \wedge cpl(S_{front}, S_2)$. This pairwise coplanarity requirement ensures the discarded hypothesis will get enclosed by the resulting planes. The false hypothesis between S_3, S_4 crosses free space and will be discarded since its spanning quad $Q_{3,4}$ occludes S_{occ}, which may be of arbitrary orientation. Note that most of the 10 possible false hypotheses between the shown parallel line segments have been left out for clarity.

intersecting of two planes P_1 and P_2 based on the presence of *common parent frames* $pnt_F(P_1) \cap pnt_F(P_2)$.

C. Finding Meaningful Planes

1) Frame-wise Plane Search: First, for each frame F, coplanar line segment combinations are searched among all $S \in cld_S(F)$. Especially the presence of many parallel segments in the scene can lead to large numbers $O(|cld_S(F)|^2)$ of false hypotheses, since every pair of parallels defines another plane. On the other hand, parallel segments cannot be ignored, since in many situations, we found parallel contours to be the only source of information, such as for untextured corridor floors. To make use of parallels without generating too many false hypotheses, each combination found to be coplanar has to pass a pair of tests to immediately discard hypotheses spanned across free space or solid volume.

To discard hypotheses crossing free space, the spanning quad $Q_{3,4}$ of a combination $S_{3,4} \in cld_S(F)$ is checked for occluding other visible line segments $S_{occ} \in cld_S(F)$ (Fig. 5). For each occlusion, a score is computed from length, position and average back distance of the occluded part of S_{occ}, which is accumulated across frames F with $S_{3,4} \in cld_S(F)$. Face occlusion checks and scoring will be explained in detail within section III-E. If the accumulated score exceeds a threshold, the hypothesis will be discarded. By this approach we achieve robustness against sporadic false occlusions, caused by segment outliers (e.g. reflections).

A similar *contour check* is performed to detect hypotheses spanned across solid volume. Now, it will be checked for $Q_{1,2}$, if it *gets occluded* by a segment $S_{front} \in cld_S(F)$ (Fig. 5). This frontal occlusion of $Q_{1,2}$ is scored and processed analogously to the back occlusion case. The potentially dangerous process of discarding hypotheses because of features *in front of* the spanning quad $Q_{1,2}$ is made robust by an additional requirement. Such hypotheses may only be discarded, if

$$cpl(S_{front}, S_1) \ \wedge \ cpl(S_{front}, S_2), \tag{5}$$

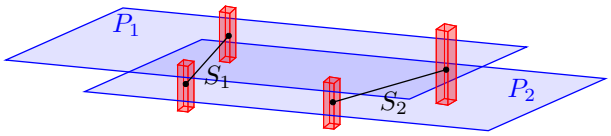


Fig. 6. Plane Merging. A combination of planes $P_{1,2}$ will get merged, if they have at least one non-collinear segment combination $S_{1,2}$ in common and do not lose any child segments after refitting to the united child segments.

that is the occluding line segment S_{front} has to be coplanar with *both*, S_1 and S_2. This means, the discarded hypothesis will be *enclosed* by the two resulting hypotheses between S_1,S_{front} and S_2,S_{front}.

False hypotheses may remain undetected, if e.g., $Q_{1,2}$ of a coplanar segment combination is of small extent compared to average distance between segments visible across the scene. However, at this stage, it is not required to discard every false hypothesis, since the final main occlusion check of the algorithm will eliminate non-solid faces generated within false planes.

2) Frame-wise Plane Merging: After plane hypotheses $P \in cld_P(F)$ passing the early occlusion and contour check were found, for each frame F all segments $S \in cld_S(F)$ are checked for coplanarity with the generated hypotheses $\in cld_P(F)$ and added to the respective sets of plane child segments $cld_S(P)$.

Now, that we obtained complete information about segment coplanarity for the current frame, we can make use of this to perform a specific merging of planes, based on common child segments between them. A combination of planes $P_1, P_2 \in cld_P(F)$ will get merged, if it fulfills the condition

$$\begin{aligned} mrg(P_1,P_2) = \\ \exists S_1, S_2 \in cld_S(P_1) \cap cld_S(P_2) : \neg cln(S_1, S_2) \\ \wedge \ \forall S \in cld_S(P_1) \cup cld_S(P_2) : cpl(S, P_{mrg}), \end{aligned} \tag{6}$$

i.e. if they have at least one *non-collinear* combination of segments in common and the resulting merged plane P_{mrg} still contains all of the united child segments, where P_{mrg} is obtained from performing a weighted least squares fit across the united child segments. When performing merging on a subset of planes, those are sorted by the number of possible merge candidates and merged with descending candidate count. This procedure is iterated until no more mergings are possible.

3) Frame-overlapping Plane Merging: After plane hypotheses have been found and merged frame-wise, a further assignment of segments to planes and merging of planes is performed, based on the requirement they own common parent frames. First, segments S for which $pnt_F(S) \cap pnt_F(P) \neq \emptyset \ \wedge \ cpl(S, P)$ are added to the child segments $cld_S(P)$ of the respective planes P. Second, planes P_1, P_2 having common parent frames $pnt_F(P_1) \cap pnt_F(P_2) \neq \emptyset$ get merged if they fulfill $mrg(P_1, P_2)$. Through this approach, we obtain geometry continuously seamed across adjacent frames, but do not perform any merging of unrelated ge-

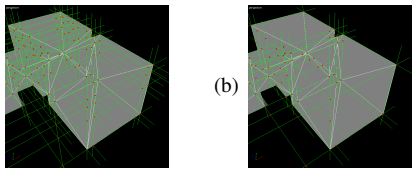


Fig. 7. Comparison of (a) exhaustive intersection and (b) under requirement of common child segments between intersecting planes in the ideal case of observed segments for every scene edge.

ometry, which could lead to an unnecessary loss of local detail at higher computational cost.

The sets of common parent frames between planes $pnt_F(P_1) \cap pnt_F(P_2)$ and segments $pnt_F(S_1) \cap pnt_F(S_2)$ can be determined efficiently by searching them among the intersections between the sets of child segments $cld_S(F)$ and child planes $cld_P(F)$ of each frame F. In general, a frame based approach is more efficient, instead of complexities $O(n_P^2)$ for plane merging and $O(n_P \cdot n_S)$ for child segment assignment, we achieve $O(n_F \cdot |cld_P(F)|)$ and $O(n_F \cdot |cld_P(F)| \cdot |cld_S(F)|)$ respectively, where n_F is the number of processed frames F. n_P and n_S are the total number of known planes and line segments, where usually $n_P \gg |cld_P(F)|$ and $n_S \gg |cld_S(F)|$.

D. Efficient Robust Plane Intersection

After plane hypotheses have been constructed, it has to be determined, in which way planes get intersected with each other to form a geometrical and topological grid of intersection points X connected along intersection lines L.

1) Regular intersections: A natural choice for an intersection criterion is the presence of a *common child segment* among two planes, since such a segment corresponds to an observed real edge of the environment geometry. Thus, a combination of planes P_1, P_2 will get intersected, if the condition

$$\begin{aligned} cmn(P_1, P_2) &= (cld_s(L_{1,2}) \neq \emptyset) \\ \text{with} \quad cld_S(L_{1,2}) &= cld_S(P_1) \cap cld_S(P_2) \end{aligned} \tag{7}$$

is fulfilled, where $L_{1,2}$ denotes the intersection line between P_1 and P_2 and $cld_S(L_{1,2})$ is the set of child segments of $L_{1,2}$, generally defined by

$$cld_S(L) = \bigcap_{P \in pnt_P(L)} cld_S(P). \tag{8}$$

Through this, every *observed* environment edge will get represented by an intersection line L within the constructed grid. After planes have been intersected according to (7), intersections points X are calculated plane-wise from combinations of lines $L_{1,2}$. Note that restricting the plane intersection leads to the existence of intersection points X with less than 3 parent planes $pnt_P(X)$ and lines $pnt_L(X)$ connected to it, while they still intersect geometrically in X. This leads to an intersection grid, only containing intersection points, colinear to observed scene edges represented by segments (Fig. 7).

Line segments S_{pln} for which $|pnt_P(S_{pln})| = 1$, i.e. which only have a single parent plane, are called *planar segments*. Such segments stem from planar contours part of observed planes and define borders between different untextured planar areas. To represent those borders, artificial intersection lines L_{pln} are constructed from the projection of those planar segments onto their parent plane, which get otherwise intersected regularly with all other intersections lines of their parent plane.

2) Line Segment Enclosement: Since we do not assume perfect line detection for every environment edge, we cannot rely on the presence of line segments as cues for plane intersections. This section explains the fundamental rule that we use to determine the need for auxiliary intersections (*after* performing regular intersections according to (7)). The main idea is the enforcement of *line segment enclosement* by intersection points X along all intersection lines L (see Fig. 8). This is motivated by the fact, that the final mesh is a 3D-grid of connected intersection point pairs and only these sections of intersection lines can make up the face edges of the final reconstruction. Accordingly, since we assume that every line segment is relevant, we have to enforce enclosement by intersection points to assure that all line segments are represented in the resulting reconstruction.

Only the outermost segment end points $p_{min,max}(S \in cld_S(L))$ along each intersection line L need to be checked for enclosement, since the line segments in between are naturally enclosed as well. Therefore, an intersection line L is *enclosed* if and only if the condition

$$\begin{aligned} enc(L) \; &= \exists X_1 \in cld_X(L) : \mu_L(X_1) \geq \mu_L(p_{max}) \\ &\wedge \exists X_2 \in cld_X(L) : \mu_L(X_2) \leq \mu_L(p_{min}) \\ &\wedge |pnt_L(X_{1,2})| \geq 3 \;\wedge\; |pnt_P(X_{1,2})| \geq 3 \end{aligned} \tag{9}$$

is true, where $\mu_L(p)$ is the scalar line parameter of a point p that corresponds to the closest point on L, $cld_X(L)$ denotes the set of child intersection points along L and $pnt_L(X)$ the set of parent intersection lines crossing in an intersection point X.

The sub predicate $|pnt_L(X_{1,2})| \geq 3 \;\wedge\; |pnt_P(X_{1,2})| \geq 3$ requires an enclosing intersection point to stem from the intersection of at least 3 planes, instead of just from intersections between 2 or more intersection lines of a single plane (e.g. of planar line segments). The reason for this (as shown in Fig. 8) is that only one of the L-adjacent faces f would be closed if an intersection point X is not a three-dimensional, but a planar intersection. However, an auxiliary intersection of the parent planes $pnt_P(L)$ with a further plane P_{aux} always yields enclosing intersection points X_{aux} fulfilling this requirement.

3) Auxiliary Intersections: If $enc(L)$ evaluates to false for any line L, the best plane P_{aux} for an auxiliary intersection has to be found. In general, we want to generate the tightest possible enclosement that minimizes the distance between X_{aux} and an unenclosed segment endpoint $p(S)$. For this, we search among the planes of the following groups:

1) common parent frame: $pnt_F(L) \cap pnt_F(P_{aux}) \neq \emptyset$,
2) spatially close: $\exists P \in pnt_P(L) : close(L, P, P_{aux})$,
3) artificial scene bounding planes.

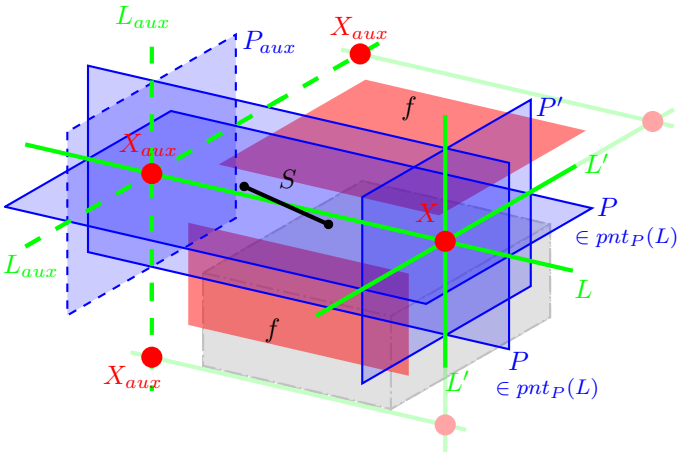


Fig. 8. To obtain a consistent 3D intersection line grid that represents all segments S, it is essential to enforce intersection points X on both segment sides. The lack of intersection points is usually a result of undetected line segments (e.g. due to lighting or occlusion). We solve this problem by searching for an auxiliary plane P_{aux} to generate the missing intersection lines L_{aux} and points X_{aux} (dashed). The requirement for intersection points to have 3 parent lines and planes guarantees that all S-adjacent faces f can be closed and thus 3D volumes are terminated by planes to all sides.

While the first group of planes is close in terms of their scene graph relation, the spatial closeness of planes is has to be defined. Since a plane P is spatially related to its child segments, we define the spatial adjacence of planes by their segment bounding boxes

$$\begin{aligned} &close(L, P \in pnt_P(L), P_{aux}) = \\ &\quad ||(L \cap P_{aux}) - C(P)|| < \lambda \cdot R(P) \\ &\quad \wedge\ ||(L \cap P_{aux}) - C(P_{aux})|| < \lambda \cdot R(P_{aux}), \end{aligned} \tag{10}$$

where $C(P)$ denotes the center and $R(P)$ the radius of the bounding box of all child segment end points $p_{1,2}(S \in cld_S(P))$ and λ serves as a safety factor (chosen dependent on the observed/expected line segment density).

Often, surfaces like untextured walls extend beyond the field of view and cannot be enclosed by planes that where generated from the visible geometry. If those faces shall be reconstructed (as is desirable in most cases, especially robotics), artificial bounding planes in all 6 principal directions need to be introduced. We parameterize these planes to form a bounding box around all detected line segments, scaled by a safety factor.

4) Face Creation: After obtaining an intersection grid from the found plane hypotheses, convex faces f are searched along the planar sub grids of intersection lines by employing a left turning search.

E. Main Occlusion Check And Solid Surface Extraction

Now, we have obtained a 3D-intersection-grid spanned with faces, that subdivides the reconstructed geometry into separate volumes. Since we initially assume that all faces are solid, we recover the free space by enforcing the visibility constraint. However, hard culling of faces that are penetrated by sightlines can be problematic with real world data. Reflections, inaccuracies or outliers could quickly destroy geometry. We use a robustified occlusion score $occ(f)$ that

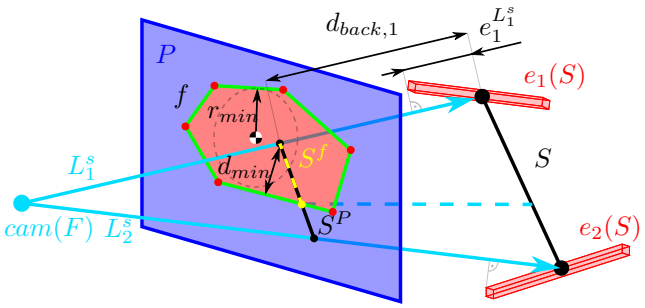


Fig. 9. Every face f is checked for occlusions of line segments. Our occlusion score uses the border distance d_{min} and penetration depth d_{back} to compute an occlusion severity $\in [0, 1]$. The border distance d_{min} is normalized by the maximal inner face circle radius r_{min} and d_{back} by the segment error in sightline direction.

factors in the severity in lateral and depth direction to assess the likelihood that a face f actually needs to be culled. After all faces have been scored in this way, the free volume is obtained by thresholding. By this approach we obtain a pessimistic estimation of the navigable space that surrounds the observer.

1) Segment Sightline Projection: For every parent frame $F \in pnt_F(P)$ of a plane P all visible child segments $S \in cld_S(F)$ with at least one end point behind P are projected onto P, with the exception of plane child segments $S \in cld_S(P)$. The sightline projection S^P of a segment S with end points $p_{1,2}(S, F)$ is defined by the segment between the projected end points $p^P_{1,2}(S, F)$, which are obtained by intersecting the sight lines

$$L^s_{1,2}(\lambda) = cam(F) + \lambda(p_{1,2}(S, F) - cam(F)), \quad \lambda \in [0; 1] \tag{11}$$

with P, where $cam(F)$ is the camera position for frame F. Segments intersecting P are trimmed to the part behind P before sight line intersection. Note, that every segment S can have different and multiple sets of end points $p_{1,2}(S, F)$ per parent frame F, since different parts of a segment can be visible in each frame, e.g. due to occlusions or unfavorable lighting.

Once the projection S^P of a segment S onto P is computed, it will be used to perform the occlusion check for all faces $f \in cld_f(P)$, i.e. all faces among the planar sub grid of intersection lines $\in cld_L(P)$.

2) Occlusion Score: To perform the check for a single face f, first, the face-overlapping sub segment S^f is determined by intersecting S^P with the border edges of f. If S^f exists, an occlusion score

$$occ(f, S) = \frac{d_{overlap}(S^f, f) \cdot d_{back}(S^f, S, f)}{|pnt_F(f)|} \tag{12}$$

is computed, where $d_{overlap}$ is an *overlap score* $\in [0; 1]$ weighting the face border distance of the planar overlap of S^f with f and d_{back} is a *back distance score* $\in [0; 1]$ weighting the average back distance of the part of S behind S^f with respect to the camera $cam(F)$. The score is normalized by the number of parent frames $|pnt_F(f)|$ of f, i.e. the number

of frames the face has been visible in, where $pnt_F(f)$ is defined by

$$pnt_F(f) = \bigcap_{L \in L(f)} \underbrace{\bigcap_{P \in pnt_P(L)} pnt_F(P)}_{pnt_F(L)}, \tag{13}$$

wherein $L(f)$ denotes the set of lines defining the border of f and $pnt_F(L)$ the parent frames of a line L.

3) Overlap Score: Let $e_i(f)$ be the edges of a face f spanned between intersection points X , then the score

$$d_{overlap}(S^f, f) = \min\{\ 1;\ d_{min}(S^f, f)/r_{min}\ \} \tag{14}$$

with

$$d_{min}(S^f, f) = \min_{e \in e_i(f)} \left\{ \max_{p \in \{p_{1,2}(S^f)\}} \{d(p, e)\} \right\} \tag{15}$$

relates border distance of S^f to the radius $r_{min}(f)$ of the biggest circle around the centroid $ctr(f)$ of f. The value $d_{min}(S^f, f)$ is the minimal distance of S^f to any of the face edges $e_i(f)$, where $d(p, e)$ is the distance of an end point p of S^f to an edge $e \in e_i(f)$. By relating to $r_{min}(f)$, d_{max} is weighted evenly along every direction for faces of varying aspect ratio.

4) Back Distance Score: By linear interpolation between the end points of a segment S, the back distances $d_{back}(S^f)$ and the sightline parallel errors $e_{back}(S^f)$ of the segment part behind S^f (i.e. the part occluded by f) are computed from the back distances $d_{back,1,2}$ and sight line parallel errors $e_{1,2}^{L_{1,2}^s} = \sum_{R=U,V,W} |e_R^{L_{1,2}^s}|$ of S. Let $\bar{d}_{back}(S^f)$ be the mean back distance and $\bar{e}_{back}(S^f)$ the mean error behind S^f, then the back distance score defines as

$$d_{back}(S^f, S, f) = \min\left\{1; \frac{\bar{d}_{back}(S^f)}{\bar{e}_{back}(S^f)}\right\}. \tag{16}$$

Visually spoken, $occ(f, S)$ expresses the depth error relative back distance, weighted by the amount of overlap between truncated segment projection S^f and face f, averaged over the frames the face f has been visible in. By this averaging, $occ(f, S)$ gains robustness against occurrences of implausible sight lines, e.g. caused by segment outliers.

After having $occ(f, S)$ computed for each face f, we can now robustly determine non-solid portals f_{portal} by thresholding with occ_{max}. In our experiments $occ_{max} = 0.25$ proved to deliver good results, meaning in every $4th$ frame, a face has been visible in, an ideal occlusion with $occ(f, S) = 1$ needs to occur.

5) Complexity: With the present implementation, the check is performed by iterating linearly over all faces of a plane, yielding an execution time $O(|cld_f(P)|)$ per plane. This could be reduced to $O(\log|cld_f(P)|)$ by employing a binary space partitioning of the planar grids, ideally using the existing intersection lines as the partitioning hyperplanes. While point queries appeared trivial, a solution to handling line segment queries with respect to occlusions still has to be found.

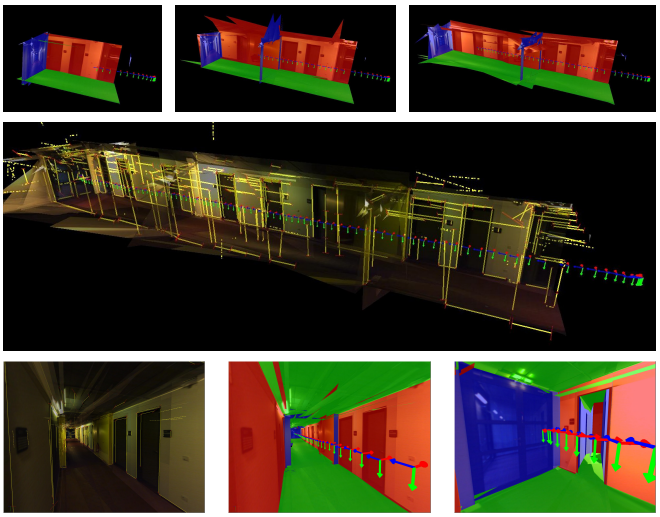


Fig. 10. The first row shows reconstructions of the Corridor sequence for increasing numbers of frames, as one would compute during robotic exploration. The middle plot shows the texture-mapped result with overlaid line segments (backfaces culled, as in the top row). The bottom row shows renderings from novel viewpoints inside the corridor (coloring shows plane normal direction). Note the open door in the bottom right rendering.

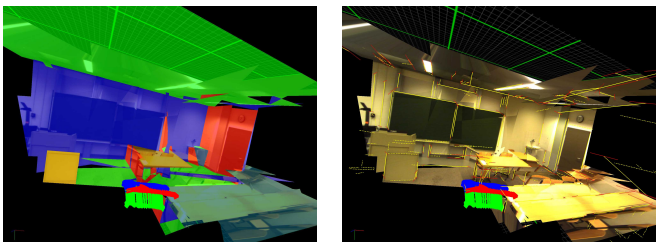


Fig. 11. The left rendering visualizes the different plane normal directions in the reconstructed Blackboard sequence. The free space is not entirely accurate, as some non-solid faces are not violating the visibility constraint from the given frames. However, the reconstruction is still useable for robotic planning, and as a robot continues to explore, it will generate more sightlines to carve out the free volume.

IV. EXPERIMENTAL RESULTS

The complete SLAM and reconstruction system that we developed is fully line-based and functions down to only two visible line segments per frame – even in complete absence of point features. For visual odometry, line segments are stereo-matched in every frame. The triangulated lines from the previous frame are iteratively reprojected and matched to the image-space lines of the current frame, as proposed in [1] (similar to the Iterative Closest Points algorithm). The pose updates are used to bootstrap line bundle adjustment, which outputs refined camera poses and world-space line segments along with their corresponding covariances. This system allows us to reconstruct challenging datasets such as the Room sequence that is virtually textureless and never shows a long shot of the complete scene geometry (Fig. 1).

Our stereo camera has a baseline of 16 cm, a resolution of 1280×1024 and a field-of-view of about $100° \times 80°$. The high resolution was used to preserve the sharpness during rectification. All datasets have 640×512 pixels resolution. The experiments were run on a single core of an Intel Core

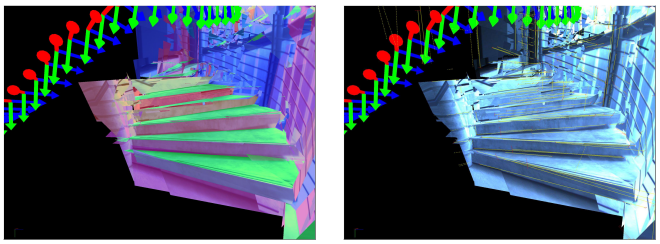


Fig. 12. Renderings of a reconstructed spiral staircase. Note the curved wall approximated by tangential planes. The coloring reflects the plane normal orientation.

i7 with 2.8 GHz. No GPU acceleration was used. See Tab. I for time measurements (excluding Line SLAM which usually takes 50-150ms per frame including bundle adjustment).

The Room sequence from Fig. 1 is the most extreme example of scene sparsity that we present in this evaluation. The white walls have large areas with bright moving specularities which may even provide false information for dense stereo or multi-view reconstruction approaches [2], [11]. Due to the small distance to the wall, the camera is only able to capture excerpts of the whole geometry (see the frustum plot that shows the trajectory from above). The plots with colored plane direction overlay exemplarily show two frames of the sequence. The large textureless border regions are especially challenging to reconstruct for other methods. The number of frames could be significantly reduced in this case, but we generally favor large numbers of sightlines to be able to cull more non-solid faces in our visibility test. Due to the low geometric complexity, the average computation time was only 0.5ms per frame.

The Corridor sequence consists of 90 frames from a 25m long trajectory in a moderately textured environment with reflections in the metallic ceiling and ceramic floor (see Fig. 10). The top image row illustrates how the reconstruction evolves during exploration and new sightlines continuously carve out the free volume. The reconstruction successfully recovers the two 20 cm ledges in the middle of the corridor (blue plane direction color) and the open door with portions of the dark room behind it (right image in bottom row).

The Blackboard scene in Fig. 11 is geometrically more complex. It is challenging that the line segment errors are relatively large due to the increased viewing distance and short camera trajectory (see the red error cuboids). However, this is a common situation for robots. Also, due to various occlusions and lighting, our Line-SLAM cannot track all line segments, which leads to an incomplete line map. While the resulting reconstruction is not entirely accurate, the correct planes are found and a robot could continue exploration while generating additional sightlines to improve its estimate of free space.

Finally, the Spiral Stairs sequence demonstrates the reconstruction of curved surfaces and various different plane orientations (Fig. 12). Of course, this is limited to curved surfaces for which a sufficient number of line segments can be detected. However, the sequence shows that the concept of our algorithm is general enough to be applied to such challenging cases.

TABLE I
EXECUTION TIMES PER FRAME BY PIPELINE STAGE

	Room	Corridor	Blackboard	Spiral Stairs
Frames	239	90	44	40
Plane Search	0.1ms	1.6ms	2.1ms	0.4ms
Intersection	0.1ms	1.6ms	1.1ms	9.2ms
Face Creation	0.1ms	1ms	0.1ms	7.2ms
Occlusion Check	0.2ms	3.1ms	2.8ms	53ms
Total per Frame	0.5ms	7.3ms	6.1ms	70ms
Total	109ms	655ms	265ms	2.73s

V. DISCUSSION

We presented a fast reconstruction method that can infer geometry from line segments with a pessimistic assumption about the free space. Plane hypotheses are efficiently generated and an intersection grid is created based on child line segment relations. The use of auxiliary plane intersections to close open contours allows reconstructions even when facing partial line detection failures. With this scheme, we try to leverage the maximal information content. However, the quality of the result still depends on the success of the Line-SLAM. Accordingly, highly textured objects have to be handled with conventional reconstruction methods. Another issue concerns the visibility constraint. The number of sightlines from line segments can be insufficient in some scenes and we may not be able to cull some non-solid faces that resulted from the intersection step. Future research will explore the addition of a simple dense stereo matching algorithm to generate supplementary sightlines for sufficiently textured regions. This will also allow to reduce the number of required frames significantly. The currently achieved reconstructions are already very useful for structured scenarios and could be even widened in scope, if point features are added to gain a fast hybrid approach.

REFERENCES

[1] J. Witt and U. Weltin, "Robust Stereo Visual Odometry Using Iterative Closest Multiple Lines," in *Proc. of IROS*. IEEE, 2013, pp. XX–XX.

[2] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis." *PAMI*, vol. 32, no. 8, pp. 1362–1376, Aug. 2010.

[3] A. Manessis, A. Hilton, P. Palmer, P. McLauchlan, and X. Shen, "Reconstruction of Scene Models from Sparse 3D Structure," in *Proc. of CVPR*, vol. 2. IEEE, 2000, pp. 2666–2673.

[4] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, "Manhattan-world stereo," in *Proc. of CVPR*. IEEE, Jun. 2009, pp. 1422–1429.

[5] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *Proc. of CVPR*, 2007, pp. 1–8.

[6] S. N. Sinha, D. Steedly, and R. Szeliski, "Piecewise planar stereo for image-based rendering," in *Proc. of ICCV*. IEEE, 2009, pp. 1881–1888.

[7] A. Flint, C. Mei, I. Reid, and D. Murray, "Growing semantically meaningful models for visual SLAM," in *Proc. of CVPR*. IEEE, 2010, pp. 467–474.

[8] P. McLauchlan, X. Shen, A. Manessis, P. Palmer, and A. Hilton, "Surface-Based Structure-from-Motion using Feature Groupings," in *Proc. of ACCV*. IEEE, 2000, pp. 699–705.

[9] D. Gallup, J. Frahm, and M. Pollefeys, "Piecewise Planar and Non-Planar Stereo for Urban Scene Reconstruction," in *Proc. of CVPR*, 2010, pp. 1418–1425.

[10] M. Tomono, "Image-based planar reconstruction for dense robotic mapping," in *Proc. of ICRA*. IEEE, 2012, pp. 3005–3012.

[11] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense Tracking and Mapping in Real-Time," in *Proc. of ICCV*. IEEE, 2011.