Globally Optimal Manhattan Frame Estimation in Real-time

Kyungdon Joo* Tae-Hyun Oh* Junsik Kim In So Kweon
Robotics and Computer Vision Lab., KAIST, South Korea
{kdjoo369, thoh.kaist.ac.kr, mibastro}@gmail.com, iskweon@kaist.ac.kr

Abstract

Given a set of surface normals, we pose a Manhattan Frame (MF) estimation problem as a consensus set maximization that maximizes the number of inliers over the rotation search space. We solve this problem through a branch-and-bound framework, which mathematically guarantees a globally optimal solution. However, the computational time of conventional branch-and-bound algorithms are intractable for real-time performance. In this paper, we propose a novel bound computation method within an efficient measurement domain for MF estimation, i.e.*, the extended Gaussian image (EGI). By relaxing the original problem, we can compute the bounds in real-time, while preserving global optimality. Furthermore, we quantitatively and qualitatively demonstrate the performance of the proposed method for synthetic and real-world data. We also show the versatility of our approach through two applications: extension to multiple MF estimation and video stabilization.*

1. Introduction

Most man-made structures, such as urban and indoor scenes, consist of a set of parallel and orthogonal planes. These structures are commonly approximated by the Manhattan World (MW) assumption [5] in the fields of computer vision and robotics. Under the MW assumption, three orthogonal directions are used to represent a scene structure, which are referred to as the Manhattan Frame (MF) in previous studies [27, 9]. Recent studies have proposed a variety of MF estimation methods for scene representation [8, 25, 13, 21]. In addition to scene representation, an accurate estimation of MF is important as a key module for many computer vision applications, such as scene understanding [24, 4, 11], SLAM [6, 30], focal length estimation [27, 2], and 3D reconstruction [8]. Since MF estimation is typically used as an early phase module in such a wide range of applications, its overall performance is critical for the viability of the application as a whole.

In order to ensure the versatile applicability to a broad range of applications, a given MF estimation method requires two properties: *stability* and *efficiency*. For stability, the robustness against noise and outliers is essential, as well as the estimation method's insensitivity to its initialization. As for efficiency, the computational complexity of the MF estimation has to remain reasonable. Even when its stability is guaranteed, an MF estimation with a high order complexity is undesirable for time critical applications, *e.g.*, SLAM.

In this research, we propose a robust and real-time Manhattan Frame (MF) estimation approach that guarantees a globally optimal solution with high stability and efficiency. We pose the MF estimation problem as that of a consensus set maximization, and solve it through a branch-and-bound (BnB) framework [15]. Typically, the bound computation is the most time-consuming part in a conventional BnB framework [19, 12]. To combat this, we suggest a relaxation of the original problem and a new bound definition that can be efficiently computed on a 2D domain (*i.e.*, EGI). This allows the bounds to be computed using a few simple arithmetic operations with linear complexity, while still preserving the global optimality and the convergence property of the BnB framework. The proposed framework is illustrated in Fig. 1. Our method is quantitatively and qualitatively validated with real and synthetic data. We also demonstrate the flexibility of our method by using it in two applications: multiple MF estimation (a mixture of Manhattan Frames) of a scene and video stabilization. In summary, the contributions of this work are as follows:

- We propose a branch-and-bound based, real-time MF estimation. Our approach can process around $300,000$ measurements in real-time.
- We relax the problem and present a new and efficient bound computation with linear-time complexity, while guaranteeing a globally optimal solution.
- Our method has been validated through systematic experiments and real world data. To show extensibility, we present multiple MF estimation and video stabilization as possible applications.
- We make the source code of the proposed method available for future use.

[*] The first and the second authors provided equal contributions.

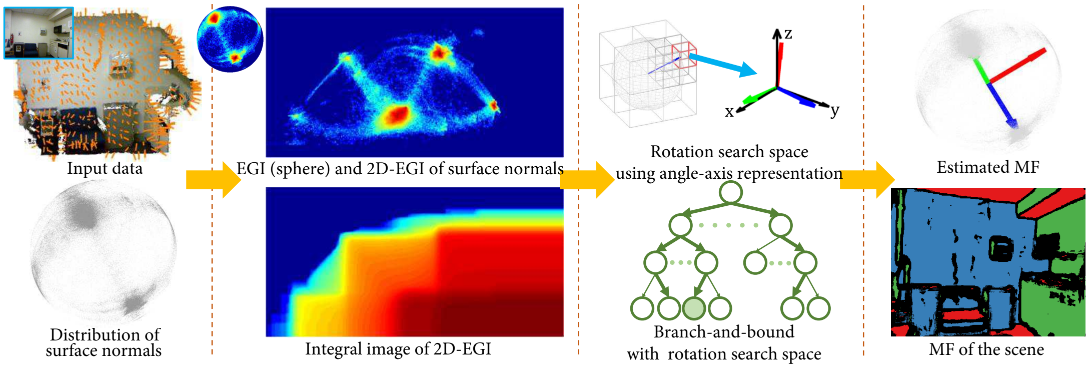

Input data
Distribution of surface normals
EGI (sphere) and 2D-EGI of surface normals
Integral image of 2D-EGI
Rotation search space using angle-axis representation
z
x
y
Branch-and-bound with rotation search space
Estimated MF
MF of the scene

Figure 1: Overview of the proposed MF estimation approach. *First column*: Input data and distribution of surface normals of a scene from NYUv2 dataset [24]. *Second column*: Its surface normal histogram, *i.e.*, EGI, the 2D-EGI and its integral image to efficiently calculate bounds. *Third column*: Illustration of the efficient bound based BnB framework using rotation search space. *Last column*: The estimated globally optimal MF.

2. Related Work

MF estimation is commonly used as a key module for high-level vision tasks, such as scene understanding and 3D reconstruction. Understanding man-made scene structures can be boiled down to either estimating the orthogonal vanishing points (VP) in the image domain or estimating the three dominant orthogonal directions (or rotation matrix) with 3D information, such as the depth or surface normals. Image-based approaches fundamentally rely on the perspective cues of an image and lines, as in [8, 13, 20, 4]. On the other hand, recent studies have focused on accurately estimating dominant orthogonal directions using 3D information [24, 28, 11]. As long as the related approaches are designed to reveal the Manhattan structure of a scene, we will refer to them as MF estimation approaches for the sake of clarity.

Silberman *et al.* [24] generate all possible MF hypotheses of a scene with measured perspective cues and surface normals. Then, all the hypotheses are exhaustively scored by the number of measurements that support the MF, and then the best one is taken. Taylor *et al.* [28] estimate a gravity vector, which corresponds to the floor normal, from RGB-D segmentation in order to sequentially estimate the other orthogonal normal vectors from wall plane segments. This approach is based on the assumptions that the vertical axis should be aligned with the gravity vector and that large planar regions should be placed near the bottom of the image. Similar to Taylor *et al.*, Gupta *et al.* [11] estimate and refine a single gravity vector from the y-axis of a RGB-D image as the initial gravity vector, without extending it to MF. Ghanem *et al.* [9] has recently proposed a non-convex MF estimation that exploits the inherent sparsity of the membership of each measured normal vector. Unfortunately, all the algorithms mentioned above are all sub-optimal.

As the importance of MF has grown, high stability and efficiency of MF estimation are desirable for general purposes. To guarantee stability, Bazin *et al.* [1, 3] propose an orthogonal VP estimation based on the BnB framework, which guarantees a global optimal solution. In [1], they present an interval analysis-based BnB framework. This strategy is improved in [3] by proposing an efficient bound computation on the rotation search space [12]. However, BnB frameworks are usually too slow for real-time applications [19]. For this purpose, Parra *et al.* [19] propose a fast BnB rotation search method that uses an efficient bound function computation, which can register up to 1000 points in 2 seconds. However this method is still inadequate for real-time applications. Straub *et al.* [26] propose a GPU-supported MF inference method that operates in real-time, but does not guarantee global optimality. In contrast to previous studies, the proposed MF estimation method with BnB framework guarantees a globally optimal solution, as well as real-time efficiency.

Straub *et al.* [27] suggest a new perspective on MF, which they call a mixture of Manhattan Frames (MMF). The motivation behind the MMF is to represent general, real-world scenes that the conventional MW assumption fails to depict. Inspired by this work, we also extend our method to multiple MF estimation as an application.

3. Problem Statement

The type of input for our problem can either be the 3D surface normals (from the depth map or 3D point cloud) or the VPs. For simplicity, we only consider the case in which the surface normals[1] are the input. Given a set of surface normals $\mathcal{N} = \{\mathbf{n}_i\}_{i=1}^{N}$, our goal is to estimate the MF of a scene, which consists of three orthogonal directions.

By virtue of the orthogonal property of an MF, the process of MF estimation becomes equivalent to that of estimating the proper rotation matrix $\mathbf{R} \in SO(3)$ that transforms the standard basis of a coordinate to a new three or-

[1] Recently, Straub *et al.* [26] show that estimating surface normals in real-time (about $15ms$) on a single GPU is possible.

thogonal basis that are aligned to the dominant surface normals, up to a sign. Since the direction vector and its flipping vector indicate the same structural support, we incorporate both the basis vectors and their flipping vectors into a set $\mathcal{E} = \{\mathbf{e}_j\}_{j=1}^{6}$[2] of six canonical vectors. To estimate the optimal rotation matrix $\mathbf{R}^*$, we formulate an optimization problem that maximizes the number of inliers as:

$$\underset{\mathbf{R}\in SO(3)}{\arg\max} \sum_{i=1}^{N} \sum_{j=1}^{6} [\![\angle(\mathbf{n}_i, \mathbf{R}\mathbf{e}_j) \leq \tau]\!], \tag{1}$$

where $\angle(\mathbf{a}, \mathbf{b})$ is the angle between the vectors $\mathbf{a}$ and $\mathbf{b}$, τ is the inlier threshold, and $[\![\cdot]\!]$ is the indicator function. Thus, the problem is to find the optimal rotation on the rotation manifold (*i.e.*, *solution search space*) by counting the number of inlier normals in the input set (*i.e.*, *measurement space*). However, Eq. (1) is not easy to handle in a numerical optimization. Similar to the approach of Li [17], we introduce an auxiliary variable $y_{ij} \in \{0, 1\}$ indicating whether the i-th surface normal is an inlier ($y_{ij} = 1$) or an outlier ($y_{ij} = 0$). Hence, we can reformulate Eq. (1) into an equivalent integer programming problem:

$$\begin{aligned} \underset{\{y_{ij}\},\mathbf{R}\in SO(3)}{\arg\max} \quad & \sum_{i=1}^{N} \sum_{j=1}^{6} y_{ij} \\ \text{s.t.} \quad & y_{ij}\angle(\mathbf{n}_i, \mathbf{R}\mathbf{e}_j) \leq y_{ij}\tau, \\ & y_{ij} \in \{0, 1\}, \forall i = 1 \cdots N,\, j = \{1, \cdots, 6\}. \end{aligned} \tag{2}$$

Solving Eq. (2) is still challenging due to the non-linearities within the geodesic distance measure, the rotation manifold parameterization, and the engagement between the two aforementioned families of unknowns. Consequently, it constitutes a challenging type of non-convex problem. Fortunately, this can be dealt with the BnB framework described in later sections.

4. Branch-and-Bound with Rotation Search

Branch-and-bound (BnB) is a general framework for global optimization [15]. The basic idea of BnB is to recursively divide a solution space into smaller sub-spaces and test each sub-space with a feasibility test to see whether it contains a globally optimal solution. The feasibility test is conducted by values called a *bound* of sub-space. The sub-spaces proven to be infeasible from the test are excluded from the search space and the remaining sub-spaces are subdivided for further searches until an optimal solution or a desired accuracy is reached. The key parts of the BnB framework are the determination of the appropriate search space and the bound function.

4.1. Branching Part

The first key part of the BnB is to define the appropriate search space. The search space in our MF estimation problem is the rotation space. We employ an angle-axis parameterization to represent a rotation matrix $\mathbf{R}$, which is formed by a three-dimensional vector $\boldsymbol{\beta}$ in a ball B_π of radius π, whose direction and norm specify the axis $\boldsymbol{\beta}/\|\boldsymbol{\beta}\|$ and angle $\|\boldsymbol{\beta}\|$ [12]. In the angle-axis parameterization, any rotation can be represented by a point in the ball B_π, which is equivalent to the search space in this problem. Let D_{init} be the initial cube that tightly encloses the ball B_π. We divide the rotation search space into smaller sub-spaces via octal subdivision for branching in BnB, as illustrated in the top half of the third column in Fig. 1.

4.2. Bounding Part

For rotation search, a useful and efficient bound computation is suggested by Bazin *et al.* [3]. We directly re-state the result of Bazin *et al.* to show its connection to our problem formulation in Eq. (2).

Proposition 1 (Bazin *et al.* [3])**.** *Given a cube D with the half side length σ and the rotation $\bar{\mathbf{R}}$ corresponding to the center of the cube D, the solutions of the following systems are the valid lower and upper bounds, L_B and U_B, of the inlier cardinality for any rotation in the cube D, respectively.*

$$\begin{aligned} L_B = \max_{\{y_{ij}\}} \quad & \sum_{i=1}^{N} \sum_{j=1}^{6} y_{ij} \\ \textit{s.t.} \quad & y_{ij}\angle(\mathbf{n}_i, \bar{\mathbf{R}}\mathbf{e}_j) \leq y_{ij}\tau, \\ & y_{ij} \in \{0, 1\}, \forall i = 1 \cdots N,\, j = \{1 \cdots 6\}. \end{aligned} \tag{3}$$

$$\begin{aligned} U_B = \max_{\{y_{ij}\}} \quad & \sum_{i=1}^{N} \sum_{j=1}^{6} y_{ij} \\ \textit{s.t.} \quad & y_{ij}\angle(\mathbf{n}_i, \bar{\mathbf{R}}\mathbf{e}_j) \leq y_{ij}(\tau + \sqrt{3}\sigma), \\ & y_{ij} \in \{0, 1\}, \forall i = 1 \cdots N,\, j = \{1 \cdots 6\}. \end{aligned} \tag{4}$$

Their proof can be directly followed in [3]. In Proposition 1, the solutions of Eqs. (3) and (4), L_B and U_B, are simply obtained by exhaustively checking the inlier constraint for each normal with respect to the rotation corresponding to the center of the given cube. We refer to this heuristic method as the *exhaustive BnB*, and will be compared with the proposed method.

According to Proposition 1, a single evaluation of either the lower or upper bound has $O(N)$ complexity, as each sample must go through a bound computation. The evaluation is linear to the number of input normals, but with an exponentially increasing number of cubes in the branching step, an efficient BnB is hard to realize. To overcome this inefficiency, we relax the problem defined in Eq. (2) and propose an efficient bound computation with $O(1)$ complexity, which is effective for MF estimation.

5. Efficient Bound Computation

To compute the bound values for a cube, we count the number of normal vectors that come within the given threshold. During the bound computation in Proposition 1, an inlier domain can be represented as a region on the measurement space. We call this region the *inlier region* (see

[2] *i.e.*, $\mathbf{e}_1 = [1\ 0\ 0]^\top$, $\mathbf{e}_2 = [0\ 1\ 0]^\top$, $\mathbf{e}_3 = [0\ 0\ 1]^\top$, $\mathbf{e}_4 = -\mathbf{e}_1$, $\mathbf{e}_5 = -\mathbf{e}_2$ and $\mathbf{e}_6 = -\mathbf{e}_3$.

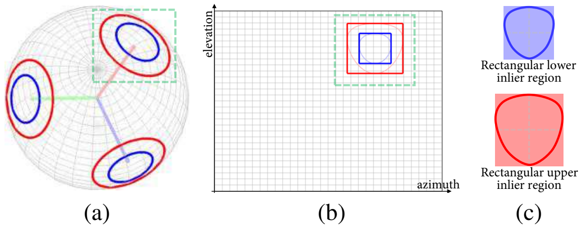

elevation
azimuth
Rectangular lower inlier region
Rectangular upper inlier region
(a)
(b)
(c)

Figure 2: Illustration of efficient inlier regions. We visualize only three direction vectors for illustration purposes. (a) Boundary point set of inlier region for lower and upper bounds on a spherical domain ($\mathcal{X}_L$ and $\mathcal{X}_U$). Blue and red indicate the boundaries of the lower and upper inlier regions, respectively. (b) An example of the transferred boundaries on the 2D-EGI ($\hat{\mathcal{X}}_L$ and $\hat{\mathcal{X}}_U$) and the rectangular inlier regions, which enclose the transferred one (this is nothing more than geometrical understanding). (c) Illustration of the rectangular inlier region on the 2D-EGI.

Fig. 2a). Thus, the bound computation counts the number of inliers within the inlier region.

Traditionally, a bound computation is the main computational bottleneck in BnB frameworks. For an efficient bound computation, we represent a set of surface normals as a surface normal histogram, which is an approximation of the extended Gaussian image (EGI) representation [14]. Based on the EGI representation, we relax the original problem in Eq. (2) and propose a new set of efficient bound functions. Throughout this paper, we will refer to the EGI representation and the surface normal histogram interchangeably.

5.1. Efficent Measurement Space on 2D-EGI

For the EGI representation, we discretize a unit sphere (*i.e.*, measurement space) by simply tessellating by its azimuth and elevation, which have the ranges of $0 \sim 360°$ and $0 \sim 180°$, respectively. A direction vector in the discretized domain is approximated and represented as its elevation and azimuth angles in discrete units. By accumulating the direction vectors (surface normals) as a histogram in each discrete unit, we can obtain the 2D-EGI (see second column of Fig. 1). We denote the 2D-EGI space as $\texttt{EGI} \in \mathbb{R}_+^{m \times n}$, where m and n are the height and width defined as $180 \times s$ and $360 \times s$, respectively, and s denotes the EGI resolution parameter, meaning that the angle unit is $1/s°$. There is a trade-off between accuracy and computation time by adjusting the EGI resolution (we empirically set $s = 2$ in our experiments). Although it may seem like a simple accumulation of surface normals, the EGI provides several powerful advantages, as it allows for a direct 2D representation [18].

Let $\mathcal{X}_L$ and $\mathcal{X}_U$ be the sets of densely sampled boundary points of the lower and upper inlier regions around the six direction vectors (blue and red points of Fig. 2a) on the spherical manifold of the measurement domain. Once $\mathcal{X}_L$ and $\mathcal{X}_U$ are mapped onto the 2D-EGI, the transferred boundaries $\hat{\mathcal{X}}_L$ and $\hat{\mathcal{X}}_U$ on the 2D-EGI domain appear as inlier regions with a curved boundary (light blue and red curves in Fig. 2b). The lower and upper bounds, L_B and U_B, can be computed by summing the histogram values within the transferred boundaries on the 2D-EGI domain, but instead we further speed up the bound function evaluation by relaxing the original problem.

5.2. Rectangular Lower and Upper Bounds

Regardless of the tightness of the bounds, any valid bound guarantees a global optimum in the BnB framework with a breadth-first-search strategy [15]. By slightly relaxing the inlier condition, we can improve the computational efficiency significantly while preserving the global optimality, if its new bounds are still valid.

Since the transferred boundaries $\hat{\mathcal{X}}_L$ and $\hat{\mathcal{X}}_U$ on the 2D-EGI have non-linear shapes, exhaustive traversal within the transferred boundaries is mandatory for computing the bounds. We instead relax the original problem defined in Eq. (2) by replacing the original constraint with a set of new axis-aligned inlier constraints as:

$$
\begin{aligned}
&\underset{\{y_{ij}\},\mathbf{R}\in SO(3)}{\arg\max} \quad \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{6} y_{ij} \\
&\quad \text{s.t.} \quad y_{ij}\phi(\mathbf{n}_i, \mathbf{R}\mathbf{e}_j) \le y_{ij}\tau_{\text{el}}, \\
&\qquad\quad\; y_{ij}\theta(\mathbf{n}_i, \mathbf{R}\mathbf{e}_j) \le y_{ij}\tau_{\text{az}}, \\
&\qquad\quad\; y_{ij} \in \{0,1\}, \forall i = 1\cdots N,\, j = \{1\cdots 6\},
\end{aligned} \tag{5}
$$

where $\phi(\cdot,\cdot)$ and $\theta(\cdot,\cdot)$ are the angle distances between two vectors along the elevation and the azimuth axes of EGI respectively, and τ_{el} and τ_{az} are the inlier thresholds for each axis. We will discuss how to choose these inlier thresholds later. These constraints form the box constraint.

We can formulate the relaxed problem in Eq. (5) into BnB by defining new valid lower and upper bound functions, similar to those in Proposition 1. For the new bound functions to be closer to those in the original problem in Eq. (2), we find the tightest circumscribed rectangles of $\hat{\mathcal{X}}_L$ and $\hat{\mathcal{X}}_U$. We call these inlier regions the *rectangular inlier regions*, as shown in Fig. 2. When we restrict the boundaries to be axis-aligned and rectangular, from the transferred point sets $\hat{\mathcal{X}}_{L,U}$, the new boundaries are uniquely defined by finding a circumscribed rectangle with the left-most, right-most, highest, and lowest points along the elevation and azimuth axes, as shown in Fig. 2c. We observe that the shape of the rectangular boundaries on the 2D-EGI varies depending on the location of $\mathbf{r}_i$. While the height of the rectangular boundary remains constant[3], the width varies according to the elevation angle of $\mathbf{r}_i$ on the 2D-EGI. Then, the bound functions of the relaxed problem in Eq. (5) can be defined as follows:

[3] Actually, this does not hold in the polar regions of the 2D-EGI due to the range limit of the 2D-EGI map. However, by cropping the rectangular bound regions that exceed the map (or zero padding), we can equally measure the bound functions near the polar regions. This indeed provides the correct number by the structural property of the 2D-EGI.

$$
\begin{aligned}
L_R = \max_{\{y_{ij}\}} \quad & \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{6} y_{ij} \\
\text{s.t.} \quad & y_{ij}\phi(\mathbf{n}_i, \bar{\mathbf{R}}\mathbf{e}_j) \leq y_{ij}\tau_{\mathrm{el}}, \\
& y_{ij}\theta(\mathbf{n}_i, \bar{\mathbf{R}}\mathbf{e}_j) \leq y_{ij}\tau_{\mathrm{az}}^{L}(\phi(\mathbf{e}_2, \bar{\mathbf{R}}\mathbf{e}_j)), \\
& y_{ij}{\in}\{0,1\}, \forall i{=}\{1\cdots N\},\ j{=}\{1\cdots 6\},
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
U_R = \max_{\{y_{ij}\}} \quad & \sum\nolimits_{i=1}^{N} \sum\nolimits_{j=1}^{6} y_{ij} \\
\text{s.t.} \quad & y_{ij}\phi(\mathbf{n}_i, \bar{\mathbf{R}}\mathbf{e}_j) \leq y_{ij}(\tau_{el} + \sqrt{3}\sigma), \\
& y_{ij}\theta(\mathbf{n}_i, \bar{\mathbf{R}}\mathbf{e}_j) \leq y_{ij}\tau_{\mathrm{az}}^{U}(\phi(\mathbf{e}_2, \bar{\mathbf{R}}\mathbf{e}_j)), \\
& y_{ij}{\in}\{0,1\}, \forall i{=}\{1\cdots N\},\ j{=}\{1\cdots 6\},
\end{aligned} \tag{7}
$$

where $\mathbf{e}_2$ is the basis of the y-axis of the 3D measurement space used to measure the elevation angle. $\tau_{\mathrm{az}}^{L}(\cdot)$ and $\tau_{\mathrm{az}}^{U}(\cdot)$ are the inlier threshold functions for the azimuth axis, which can be computed in advance by finding the circumscribed rectangles of $\hat{\mathcal{X}}_L$ and $\hat{\mathcal{X}}_U$ and storing them into a look-up table, as will be explained in Sec. 5.5.

The feasibility constraints in the aforementioned rectangular inlier regions are interpreted as whether a pixel in the 2D-EGI map is inside the rectangular regions defined by the inlier thresholds. Thus, the bound computation can be done by summing up the values in the rectangular regions, which can be efficiently computed using the *integral image* [29] (*i.e.*, through simple add and subtract operations with the four corner values of the rectangular inlier region on the integral image). The proposed rectangular bound has the following property.

Lemma 1. *For a cube D, let c_D^* be the optimal inlier cardinality of the relaxed problem (Eq. (5)). Then the bounds L_R and U_R obtained by the proposed method satisfy $L_R \leq c_D^* \leq U_R$. Also, when the maximum half side length of D,* i.e.*, σ, goes to zero, then $U_R - L_R \leq \epsilon$.*

Lemma 1 asserts that the proposed bound functions of Eqs. (6) and (7) are valid, and is useful for further theoretical analysis on the behavior of BnB using rectangular bounds, which will be discussed in a later section.

5.3. Algorithm Procedure

The BnB procedure is formalized in Alg. 1. The algorithm reduces the search space iteratively by rejecting subspaces with the feasibility test until it converges to a globally optimal value or reaches a desired accuracy level. At first, the cube-list $\mathcal{L}$ is initialized with the cube D_{init} that encloses the rotation ball B_π. At every iteration, each cube in the cube-list is subdivided into octal sub-cubes with the half length size of its parent cube and stored in the cube-list while removing the parent cubes from the list. For each subcube, the rotation center and the lower and upper bounds are computed. Then, a feasibility test is conducted based on the rectangular bounds (*c.f*. Sec. 5.2). Cubes with an upper bound smaller than the maximum lower bound are excluded from the cube-list, as they are guaranteed to not contain the globally optimal solution. This procedure repeats iteratively until a single cube, of which the lower bound and the upper bound are the same, remains or a desired accuracy is achieved. The solution is the rotation of the cube center that has the highest cardinality, *i.e., the rotation guaranteed to be globally optimum.*

Algorithm 1 BnB on the Efficient Measurement Space

```
Initialize the cube list $\mathcal{L}$ with the rotation ball $B_\pi$.
  repeat
      Subdivision($\sigma \leftarrow \sigma/2$) of each cube $D_i$ of $\mathcal{L}$.
      for each cube $D_i$ of $\mathcal{L}$ do
          Calculate the rotation $\mathbf{R}_{D_i}$ of the cube center.
          Compute the rectangular lower $L_{R_i}$ and upper $U_{R_i}$ bounds.
          (c.f. Sec. 5.2).
      end for
      $L^* = \max_i L_{R_i}$, $i^* = \arg\max_i U_{R_i}$,
      $U^* = U_{R_{i^*}}$, $\mathbf{R}^* = \mathbf{R}_{D_{i^*}}$.
      Remove all the cubes from $\mathcal{L}$ such that $U_{R_i} < L^*$.
  until $\exists i$, such that $L_{R_i} = U^*$ (i.e., at least one cube whose lower bound
  is $U^*$) or it reaches the desired accuracy level.
Output: $\mathbf{R}^*$ (i.e., the rotation matrix maximizing the number of inliers).
```

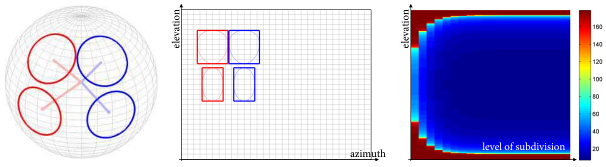
5.4. Analysis

Computational Complexity Compared to the related studies [12, 3], Proposition 1 tells us that each bound computation has $O(N)$ complexity. For simplicity, let C be the number of cubes that should be evaluated in the BnB framework until convergence. Then, the computation complexity of the whole procedure is $O(CN)$.

In our framework, constructing the EGI by accumulating surface normals and the integral image take $O(N)$ and $O(B)$, respectively, only at the initial stage, where B denotes the number of bins, *i.e.*, mn of EGI. Each bound computation with the 2D-EGI representation takes $O(1)$ on the integral image. Since the initial stage exhibits a linear complexity with respect to the resolution of EGI and the BnB procedure on the proposed problem shows a linear complexity with respect to the number of evaluated cubes C, the overall algorithm complexity is $O(C + B)$ which is still linear.

We can see that the proposed method is much faster than the exhaustive BnB method [3], which has a quadratic complexity $O(CN)$. In practice, given a single depth image with a resolution of 640 × 480, N is around $300,000$ samples, and C is in the range of hundreds of thousands to millions. This gives a sense of what makes the proposed algorithm real-time.

Convergence Since it is already shown that the rectangular bounds are valid bounds by Lemma 1, the proposed method with the rectangular bounds is guaranteed to compute the globally optimal solution [15]. However, we utilize a discrete EGI representation, where a certain sub-division level exists such that subsequent upper bound values are quantized into the same value. Albeit with this discretization, our method still guarantees to converge to the bounded globally optimal solution.



(a) (b) (c)

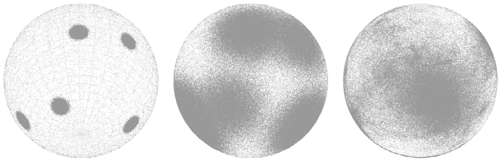
Figure 3: Illustration of the properties of the rectangular inlier region in the 2D-EGI domain for LUT. (a) The four direction vectors and their inlier regions on the sphere domain. The same color direction vectors have the same azimuth value, but a different elevation value. (b) The rectangular inlier regions on the 2D-EGI. (c) The example of LUT, whose column axis indicates the elevation angle, row axis indicates the level of subdivision, and the values of entries encode the half width size of the corresponding rectangular inlier region.

5.5. Further Speed Up

Search Space Rejection As mentioned before, given a rotation matrix $\mathbf{R} = [\mathbf{r}_1\ \mathbf{r}_2\ \mathbf{r}_3]$ of a cube, a direction $\mathbf{r}_i$ and its flipping vector $-\mathbf{r}_i$ (for $i = \{1, 2, 3\}$) indicate the same vector, *i.e.*, the rotation matrices $\mathbf{R}$ and $-\mathbf{R}$ indicate the same solution. Hence, we do not need to search within the other half-sphere in the rotation solution space.

Look-Up Table (LUT) We observed a few properties regarding the shape of the rectangular inlier region (*c.f.* Sec. 5.2). Firstly, the half height of the rectangular inlier region remains constant, *i.e.*, τ and $\tau + \sqrt{3}\sigma$ for the lower and the upper rectangular inlier regions, respectively. That is, τ_{el} in the 2D-EGI is equal to τ in the original domain. Secondly, the half width of the rectangular inlier region only depends on the elevation angle of the mapping point of $\mathbf{r}_i$ (or $-\mathbf{r}_i$) onto the 2D-EGI. Lastly, the shapes of the rectangular inlier regions on the same elevation angle are equal, regardless of azimuth angles, as illustrated in Fig. 3b. Therefore, we generate a look-up table (LUT), which describes $\tau^L_{\text{az}}(\cdot)$ and $\tau^U_{\text{az}}(\cdot)$, by precomputing the half widths of the rectangular inlier region along the elevation angles and along the level of subdivision (*i.e.*, threshold $\tau + \sqrt{3}\sigma_k$) in order to reduce redundant and repetitive computations.

By virtue of the subdivision rule of the rotation search space [12], we can pre-compute a series of bound thresholds as $\tau + \sqrt{3}\sigma_k$, where $\sigma_k = \sigma_0/2^k$ and σ_0 is the half side length of the initial cube. At the beginning of the algorithm, with the pre-computed bound thresholds, we construct an M-vector LUT for each level of subdivision, where M denotes the user specified deepest level. Entries of each vector store the calculated half width sizes of the rectangular inlier regions corresponding to the elevation angles[4]. By concatenating each vector-form LUT, we can generate a matrix-form LUT (see Fig. 3c). By using the LUT matrix, we can obtain a rectangular inlier region without any computation while running the algorithm.



(a) $\kappa^{-1} = 0.0012$ (b) $\kappa^{-1} = 0.08$ (c) Real data

Figure 4: Distributions of surface normals. (a) and (b) are synthetic data distribution according to κ^{-1} of vMF, 0.0012 and 0.08, respectively. (c) A sample distribution of real data from the NYUv2 dataset [24].

6. Experimental Result

In this section, we present our experimental results to answer the following questions:

- Sensitivity: How sensitive is our method to the parameters (EGI resolution s and inlier threshold τ)?
- Robustness: Is our method robust to noise and outliers?
- Convergence: How many sub-divisions (iterations) are typically required until convergence?
- Speed: How much faster is our method?
- Practicality: Does our method robustly and efficiently work with real-world data?

6.1. Simulation

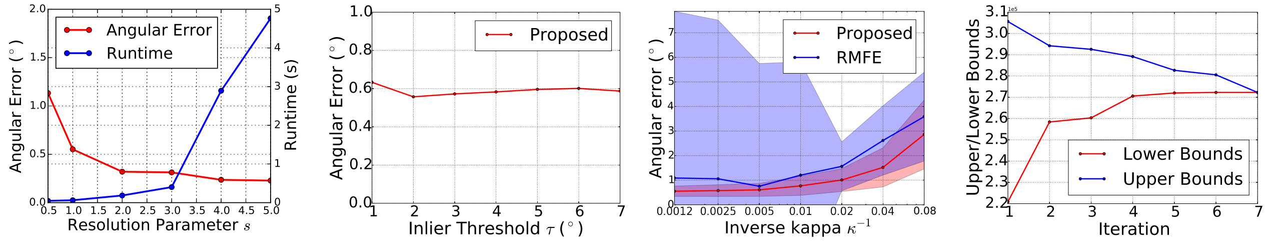
In synthetic simulations, we perform various experiments to demonstrate stability (accuracy and convergence) and efficiency (time profile) of the proposed BnB method. For synthetic data, we randomly selected three orthogonal direction vectors that correspond to the ground truth MF and two additional direction vectors to generate outlier directions. We then sampled $400,000$ surface normals on the von-Mises-Fisher (vMF) distribution [7][5], which is an isotropic distribution over the unit sphere. We also sampled $10,000$ surface normals on a uniform distribution to generate sparse noise (see Fig. 4). Unless specifically mentioned otherwise, we fixed the inlier threshold τ as $5°$, the EGI resolution parameter s as 2, and the inverse kappa κ^{-1} as 0.01, for each experiment. We ran each experiment 100 times on MATLAB and measured the mean of the max angular error.

Accuracy We tested the trade-off between the accuracy and runtime of the proposed method, with respect to the change in EGI resolution parameter s. As shown in Fig. 5a, the accuracy improves as the resolution parameter s increases, but the ratio of the increased runtime is critical. In all other experiments, we empirically chose the resolution parameter $s = 2$.

Our method shows stable accuracy regardless of the inlier threshold τ (see Fig. 5b). We also evaluated the change

[4] The width calculation is done by finding the tightest rectangle enclosing the transferred boundaries $\hat{\mathcal{X}}_L$ or $\hat{\mathcal{X}}_U$.

[5] The von Mises-Fisher distribution for a p-dimensional unit vector $\mathbf{x}$ is defined by $f_p(\mathbf{x}; \mu, \kappa) = Z_p(\kappa)\exp(\kappa\mu^T\mathbf{x})$, where μ is the mean direction, κ is the concentration and normalization constant $Z_p(\kappa) = \kappa^{p/2-1}(2\pi)^{-p/2}\mathbf{I}_{p/2-1}(\kappa)^{-1}$, where $\mathbf{I}_v$ denotes the modified Bessel function of the first kind at order v.



(a) Effects of EGI resolution (b) Accuracy vs. inlier threshold τ (c) Accuracy vs. noise variance (d) Convergence graph

Figure 5: Simulation results for observing the behaviors of the proposed method.

Method	MPE [28]	MMF [27]	ES [24]	RMFE [9]	Exhaustive BnB [3]	Proposed
θ_x	26.3°	8.1°	2.3°	2.3°	2.9°	3.0°
θ_y	18.1°	19.6°	5.6°	4.7°	1.8°	2.0°
θ_z	18.2°	9.8°	2.9°	2.8°	2.8°	2.9°
Avg.	20.87°	12.50°	3.60°	3.27°	**2.5**°	**2.63**°
Runtime (s)	2.8	0.1	21.4	0.9	117.06	**0.07**

Table 1: Comparisons of average angular error and runtime for the ground truth of NYUv2 dataset [9].

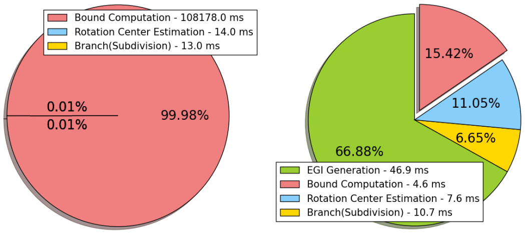
in accuracy as the variation parameter of the vMF distribution changed, namely κ^{-1}, and compared it with the state-of-the-art robust MF estimation (RMFE) method [9][6]. For this experiment, we tested κ^{-1} within a range from 0.0012 to 0.08 on a log-scale. In Fig. 5c, the blue and red lines indicate the mean angular error, and the light blue and light red regions represent the standard deviation of angular error according to κ^{-1}. While RMFE is easily biased toward an outlier and shows an unstable error with a large standard deviation for small κ^{-1} values, the proposed method shows a stable and precise accuracy, as shown in Fig. 5c.

Convergence The lower and upper bounds of the proposed BnB method converge to a specific value, demonstrating the convergence property of the proposed algorithm. It commonly converges within 7 iterations and shows that the efficient bounds are valid, as seen in Fig. 5d.

Time Profiling To show improvements in the computational efficiency, we compared the time profiles of the exhaustive BnB [3] and the proposed BnB (see Fig. 6). Both methods have three steps in common: branch (subdivision), rotation center estimation, and bound computation. In addition, the proposed BnB has an EGI generation step for efficient bound estimation. In the case of the exhaustive BnB, bound computation takes $108.178s$, which is 99.98% of the entire computational time. On the other hand, the proposed BnB takes only $4.6ms$ to compute the bounds. This reduces the bound computation time by more than $20,000$ times, compared to that of the exhaustive BnB.

6.2. Real-World Experiment

To evaluate the performance of our method on real-world data, we used the NYUv2 dataset [24], which contains 1449 RGB-D images of various indoor scenes. In particular, we used the recently introduced ground truth benchmark [9] of the NYUv2 dataset for a quantitative evaluation.



(a) Exhaustive BnB [3] (b) Proposed BnB

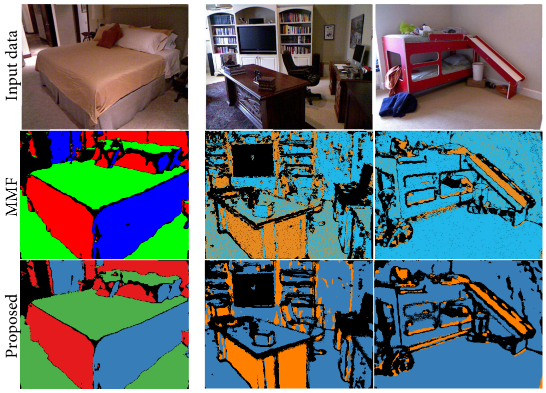
Figure 6: Time profiles of the exhaustive BnB [3] and the proposed BnB approach.

We compare the exhaustive BnB and the proposed BnB with MPE [28], MMF [27], ES [24] and RMFE [9]. Except for the exhaustive BnB and the proposed BnB, we directly quote the results from Ghanem *et al*. [9]. We tested the average angular errors of the exhaustive BnB and the proposed BnB on similar hardware configurations (*i.e*., a 3GHz workstation on MATLAB), as done by Ghanem *et al*. Since MPE is based on the assumption that a large portion of the scene consists of the floor plane, it is sensitive to scene clutter and outliers. MMF also shows less accurate results than those of ES and RMFE. We deduce the reason for MMF's poor performance in angular errors as the absence of noise/outlier handling. ES and RMFE show comparable results, but their runtimes are inefficient for real-time applications. Exhaustive BnB shows the most accurate results, but its runtime is intractable in terms of efficiency. On the other hand, the proposed BnB performs stably while achieving real-time efficiency, as shown in Table 1. Accuracy differences between the exhaustive BnB and the proposed one comes from the relaxation in Eq. (5).

6.3. Applications

Extension to Multiple Manhattan Frames Since conventional MW assumptions cannot represent general real-world indoor and urban scenes, Straub *et al*. [27] introduces a more flexible description of a scene, consisting of multiple MF, namely a mixture of Manhattan Frames (MMF). As an application, we extended the proposed method to MMF estimation by sequentially finding different MFs.

[6] For a fair comparison, we used the publicly available code and set the balancing parameter λ of the original paper [9] to be equal to that in our implementation.

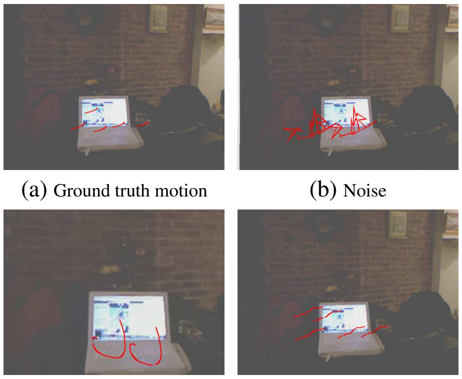


(a) 1 MF (b) 2 MFs

Figure 7: Extension to Multiple Manhattan Frames. We show the RGB images, the original MMF [27] and the estimated MMF by our method on various indoor scenes in the first, second, and last row, respectively. (a) 1 MF. Each color indicates an MF axis. (b) 2 MFs. Each color (orange and blue) indicates different MFs.

To estimate MMF, we applied a greedy-like algorithm. For a given input data, we estimated the optimal MF, and updated the normal data by excluding the set of normals that corresponds to the inliers of the optimal MF. We then sequentially estimated the next optimal MF for the updated normal data. As in [27], we only considered MFs with inlier normals that account for more than 15% of all valid normals, to deal with poor depth measurements. We qualitatively demonstrated our extension, *i.e.*, MMF inference for the NYUv2 dataset [24], by comparing it with the original MMF approach [27]. In Fig. 7, the proposed MMF inference shows comparable results with that of the original MMF, as well as the theoretical guarantees.

Video Stabilization The goal of video stabilization is to generate a visually stable and pleasant video from a video with jitters due to camera shakes. Depending on the information used for stabilization, the approaches can be grouped into two categories: 2D and 3D motion-based stabilization. 3D-based stabilization reflects more realistic motion information than 2D-based stabilization. Recently, Jia *et al.* [16] proposed a 3D video stabilization method that exploits the rotation of camera poses obtained from a built-in gyroscope in smartphones and tablet PCs. Instead of using the 3D rotations from the gyroscope, we apply the rotation matrices obtained by the proposed method and verify the applicability of the algorithm on video stabilization. For the experiment, we used the NYUv2 dataset [24] with synthetic rotation noise that mimic egocentric head motions applied to the image and depth sequences.



(a) Ground truth motion (b) Noise (c) YouTube video editor [10] (d) proposed

Figure 8: Video stabilization. We used the "NYUv2-living room part 1-living room 0009" dataset. Features are tracked between frames 265 and 285. The feature trajectories are overlaid as red polylines on frame 285.

We obtained the feature trajectories, shown in Fig. 8, only to visualize the stabilization performance. The initial features were detected by FAST [22], then tracked by KLT [23] for consecutive frames. We compared the feature trajectories of the video processed in two different ways: using the YouTube video editor [10] and the stabilization based on the proposed MF estimation. Fig. 8a shows the smooth feature trajectories of the original video that visualize the true camera motions. Fig. 8b shows the jittering feature trajectories of the synthetic rotation noise applied to the original video. Since the YouTube video editor uses 2D motion information, it generates a more smoothed trajectory, but it is far from the true camera motions, while the trajectory of our method shows similar tendencies to the original feature trajectory (see Fig. 8c and Fig. 8d). Note that the depth normals obtained from the NYUv2 dataset are very noisy and give inaccurate normal information. However, the stabilization using our method shows plausible results.

More qualitative experiments and analyses for the two applications are included in the supplementary materials.

7. Conclusion

In this paper, we propose a robust and real-time MF estimation that guarantees a globally optimal solution. This can be achieved by relaxing the original cardinality problem, so that the computational complexity of BnB is dramatically reduced, to a linear complexity. We prove the efficiency and stability of the proposed method through various synthetic and real-world experiments. The proposed method outperforms previous methods' speed with precise accuracy. We also apply the method on two applications: multiple MF estimation and video stabilization, and confirm the applicability of our work on an application level.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) and Korea Creative Content Agency (KOCCA) grant funded by the Korea government (MCST) (R0132-15-1006, Developing the technology of open composable content editors for realistic media).

References

[1] J.-C. Bazin, Y. Seo, C. Demonceaux, P. Vasseur, K. Ikeuchi, I. Kweon, and M. Pollefeys. Globally optimal line clustering and vanishing point estimation in manhattan world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 638–645, 2012. 2

[2] J.-C. Bazin, Y. Seo, R. Hartley, and M. Pollefeys. Globally optimal inlier set maximization with unknown rotation and focal length. In *European Conference on Computer Vision (ECCV)*, pages 803–817. Springer, 2014. 1

[3] J.-C. Bazin, Y. Seo, and M. Pollefeys. Globally optimal consensus set maximization through rotation search. In *Asian Conference on Computer Vision (ACCV)*, pages 539–551. Springer, 2012. 2, 3, 5, 7

[4] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 33–40, 2013. 1, 2

[5] J. M. Coughlan and A. L. Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 941–947, 1999. 1

[6] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the rgb-d slam system. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1691–1696, 2012. 1

[7] N. I. Fisher. *Statistical analysis of circular data*. Cambridge University Press, 1995. 6

[8] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Manhattan-world stereo. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1422–1429, 2009. 1, 2

[9] B. Ghanem, A. Thabet, J. Carlos Niebles, and F. Caba Heilbron. Robust manhattan frame estimation from a single rgb-d image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3772–3780, 2015. 1, 2, 7

[10] M. Grundmann, V. Kwatra, and I. Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 225–232. IEEE, 2011. 8

[11] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 564–571, 2013. 1, 2

[12] R. I. Hartley and F. Kahl. Global optimization through rotation space search. *International Journal of Computer Vision (IJCV)*, 82(1):64–79, 2009. 1, 2, 3, 5, 6

[13] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1849–1856, 2009. 1, 2

[14] B. K. Horn. Extended gaussian images. *Proceedings of the IEEE*, 72(12):1671–1686, 1984. 4

[15] R. Horst and H. Tuy. *Global optimization: Deterministic approaches*. Springer Science & Business Media, 2013. 1, 3, 4, 5

[16] C. Jia and B. L. Evans. Constrained 3d rotation smoothing via global manifold regression for video stabilization. *IEEE Transactions on Signal Processing*, 62(13):3293–3304, 2014. 8

[17] H. Li. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1074–1080. IEEE, 2009. 3

[18] A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3d point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1297–1304, 2006. 4

[19] A. J. Parra Bustos, T.-J. Chin, and D. Suter. Fast rotation search with stereographic projections for 3d registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3930–3937, 2014. 1, 2

[20] L. D. Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2719–2726, 2012. 2

[21] S. Ramalingam and M. Brand. Lifting 3d manhattan lines from a single image. In *IEEE International Conference on Computer Vision (ICCV)*, pages 497–504, 2013. 1

[22] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 1508–1515. IEEE, 2005. 8

[23] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593–600. IEEE, 1994. 8

[24] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*, pages 746–760. Springer, 2012. 1, 2, 6, 7, 8

[25] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1881–1888, 2009. 1

[26] J. Straub, N. Bhandari, J. J. Leonard, and J. W. Fisher III. Real-time manhattan world rotation estimation in 3d. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1913–1920, 2015. 2

[27] J. Straub, G. Rosman, O. Freifeld, J. J. Leonard, and J. W. Fisher. A mixture of manhattan frames: Beyond the manhattan world. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3770–3777, 2014. 1, 2, 7, 8

[28] C. Taylor and A. Cowley. Parsing indoor scenes using rgb-d imagery. In *Proceedings of Robotics: Science and Systems*, July 2012. 2, 7

[29] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–511–I–518 vol.1, 2001. 5

[30] 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*, 64(4):1364–1375, 2015. 1