Blocks-World Cameras

Jongho Lee
University of Wisconsin-Madison
jongho@cs.wisc.edu

Mohit Gupta
University of Wisconsin-Madison
mohitg@cs.wisc.edu

Abstract

For several vision and robotics applications, 3D geometry of man-made environments such as indoor scenes can be represented with a small number of dominant planes. However, conventional 3D vision techniques typically first acquire dense 3D point clouds before estimating the compact piece-wise planar representations (e.g., by plane-fitting). This approach is costly, both in terms of acquisition and computational requirements, and potentially unreliable due to noisy point clouds. We propose Blocks-World Cameras, a class of imaging systems which directly recover dominant planes of piece-wise planar scenes (Blocks-World), without requiring point clouds. The Blocks-World Cameras are based on a structured-light system projecting a single pattern with a sparse set of cross-shaped features. We develop a novel geometric algorithm for recovering scene planes without explicit correspondence matching, thereby avoiding computationally intensive search or optimization routines. The proposed approach has low device and computational complexity, and requires capturing only one or two images. We demonstrate highly efficient and precise planar-scene sensing with simulations and real experiments, across various imaging conditions, including defocus blur, large lighting variations, ambient illumination, and scene clutter.

1. The 3D Revolution

We are in the midst of a 3D revolution. Robots enabled by 3D cameras are beginning to drive cars, explore space, and manage our factories. While some of these applications require high-resolution 3D scans of the surroundings, several tasks do not explicitly need dense 3D point clouds. Imagine a robot navigating an indoor space, or an augmented reality (AR) system finding surfaces in a living room for placing virtual objects. For such applications, particularly in devices with limited computational budgets, it is often desirable to create *compact*, memory- and compute-efficient 3D scene representations. For example, in piece-wise planar indoor scenes, a popular approach is to *first* capture 3D point clouds with a depth or an RGBD camera, *and then* estimate a piece-wise planar representation (Fig. 1).

Historically, point clouds have been the canonical representation for 3D scenes in the computer vision and robotics communities. This is not surprising because almost all depth imaging modalities capture 3D point clouds as the raw data. Indeed, there are several applications which do require dense 3D representations (e.g., CAD modeling, facial motion retargeting), for which points clouds are a good fit. However, point clouds also have limitations: First, dense point clouds are memory, compute and bandwidth intensive. Second, acquisition of point clouds by depth cameras is prone to errors in non-ideal imaging conditions including defocus, multi-path [23, 46, 43] and multi-camera interference [10, 63, 39], and ambient illumination [24, 3]. Finally, extracting piece-wise planar representation by fitting planes to a point cloud requires global reasoning, which may result in inaccurate plane segmentation, especially if the underlying point-clouds are noisy to begin with (Fig. 1).

This raises a natural question: Why capture high-resolution and noisy 3D point clouds at large acquisition costs, only to compress it later into planar representations at large computational cost? If we are going to perform downstream reasoning in terms of planes, can we design imaging modalities that *directly* capture compact and accurate *plane-centric geometric representations* of the world?

We propose *Blocks-World Cameras*, a class of imaging systems which directly recover dominant plane parameters for Blocks-World [57] (piece-wise planar) scenes without creating 3D point clouds, enabling fast, low-cost and accurate reconstructions (Fig. 1). The Blocks-World Cameras are based on a structured-light system consisting of a projector which projects a *single* pattern on the scene, and a camera to capture the images. The pattern consists of a sparse set of cross-shaped features (each with two line-segments) which get mapped to cross-shaped features in the camera image via homographies induced by scene planes. If correspondences between image and pattern features can be established, the plane parameters can be estimated simply by measuring the deformation (change of angles of the two segments) between these features [28].

For scenes with high geometric complexity (e.g., a large number of distinct dominant planes), the projected pattern must have a sufficiently high feature density, requiring multiple features on each epipolar line, leading to ambiguities. Resolving these ambiguities would require correspondence matching via computationally intensive global reasoning,

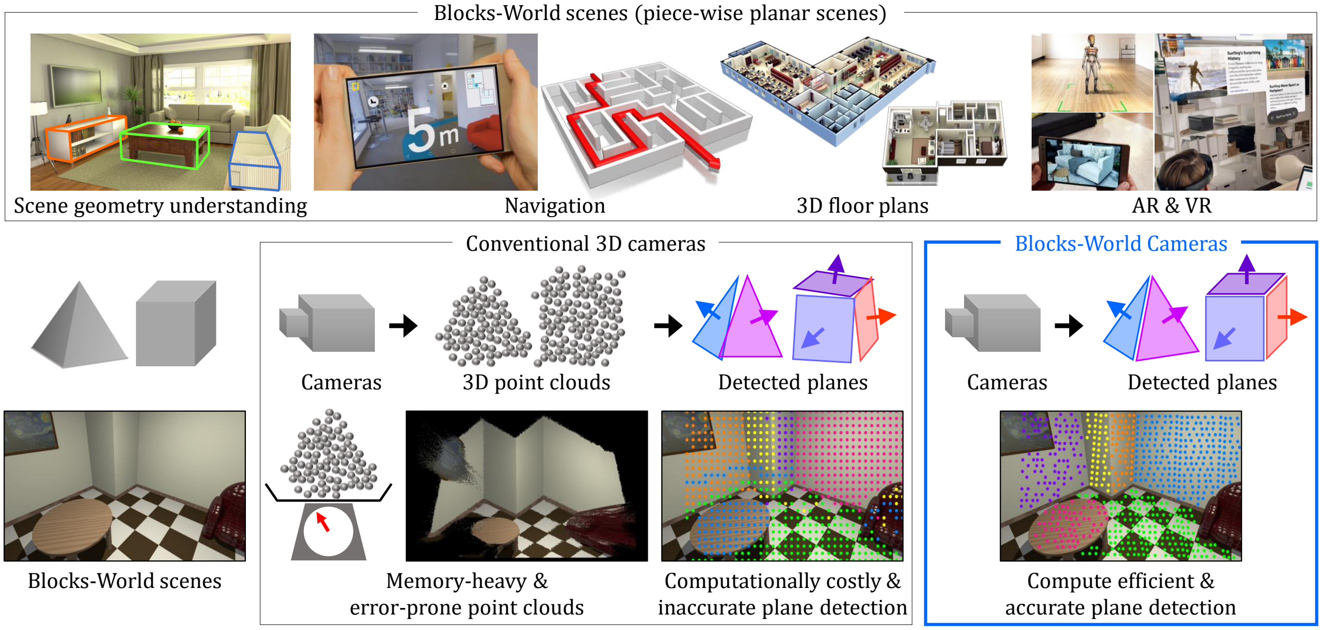


Figure 1. **Blocks-World Cameras.** (top) Several applications require compact 3D representations of piece-wise planar scenes. (bottom) Even for such blocks-World scenes, conventional approaches first recover dense 3D point clouds, followed by estimating planar scenes via plane-fitting. This process has large acquisition and computational costs, and is often error-prone. We propose Blocks-World Cameras for recovering dominant planes **directly** without creating 3D point clouds, enabling fast, low-cost and accurate Blocks-World reconstructions.

thus defeating the purpose of Blocks-World Cameras. Is it possible to perform reconstruction while maintaining both high feature density and low computational complexity?

Scene representation with plane parameter space: We develop a novel geometric method which enables plane estimation even with *unknown* correspondences. For a given image feature, the set of all the candidate pattern feature correspondences vote for a set of plane hypotheses (in the 3D plane parameter space), called the *plane parameter locus.* Our key observation is that if the pattern features are spaced *non-uniformly* on the epipolar line, then the plane parameter loci for multiple image features lying on the same world plane will intersect at a *unique* location in the parameter space. The intersection point corresponds to the parameters of the world plane, and can be determined by simple peak finding, without determining correspondences.

Implications: Based on this observation, we design a pattern, and a fast algorithm that simultaneously recovers depths and normals of Blocks-World scenes. We demonstrate, via simulations and experiments, capture of clean and clutter-free 3D models, for a wide range of challenging scenarios, including texture-rich and texture-poor scenes, strong defocus, and large lighting variations. The computational complexity of the proposed approach is low, and remains largely the same regardless of the geometric complexity of the scene, enabling real-time performance on high-resolution images. The method requires capturing only 1 or 2 images, and can be implemented with simple and low-cost *single-pattern* projectors with a static mask. Furthermore, the sparsity of the projected pattern makes it robust to interreflections, a challenging problem which is difficult to solve with dense patterns.

Scope: Blocks-World Cameras are specifically tailored to piece-wise planar scenes, in applications requiring compact 3D representations consisting of a small set of planes. It is not meant to be a general-purpose technique that can replace conventional approaches. Indeed, for scenarios requiring dense geometry information for complex scenes, existing 3D imaging approaches will achieve better performance. However, the proposed technique can facilitate fast and robust dominant plane extraction, with applications in robotic navigation [66, 56], indoor scene modeling and AR.

2. Related Work

Piece-wise planar scene constraint: There is a long tradition of piece-wise planar 3D scene reconstructions, starting from the Blocks-World [57] and Origami-World [37] works nearly five decades ago. Since then, piece-wise planarity has been widely used as a prior for accurate 3D modeling [55, 49, 15, 31, 18, 7, 69], and scene understanding [29, 54, 76, 20]. In Multi-View Stereo, the planar scene constraint has been used to overcome lack of texture, repetitive structures, and occlusions [18, 64, 44, 7, 69]. Planes are popular scene primitives in SLAM [59, 66, 12, 74, 38, 36] as well, having been used for efficient and accurate 3D registration between frames [56, 66]. The planar scene constraint

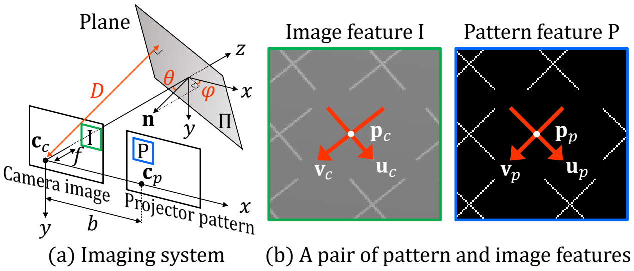


Figure 2. **Imaging principle.** (a) The Blocks-World Cameras are based on a structured-light system consisting of a projector to project a single pattern on the scenes and a camera to capture the images. (b) The pattern consists of a sparse set of cross-shaped features, which get mapped to cross-shaped features in the image via homographies induced by scene planes.

has been used for detecting junctions of indoor scenes or wireframes of urban scenes to recover scene layouts from a single RGB image [54, 76]. The Manhattan world constraint [13] which assumes the scenes to be made of axis-aligned planes has been exploited to reconstruct indoor environments such as floor-plans and room layouts [11, 40].

Plane-fitting to point clouds: A piece-wise planar scene representation can be created from the dense, and often noisy, 3D point clouds captured by conventional depth cameras, by fitting planes. For example, Hough transform [32] is a method for detecting parameterized objects such as lines and circles in images, and is easily extended to 3D planes [8, 33]. The RANdom SAmple Consensus (RANSAC) [16] has also been widely used for plane detection due to its robustness to outliers [19, 7, 67]. Other approaches for plane-fitting include region growing [52, 30, 48], as well as energy-based multi-model fitting [35, 55, 69]. These approaches can be computationally intensive especially for cluttered scenes, often requiring complex global reasoning. In contrast, Blocks-World Cameras infer the parameters of the piecewise planar scenes *directly* using lightweight computational algorithms, without capturing 3D point clouds.

Scene planarity in learning-based approaches: Recently, scene planarity has been used in learning-based approaches for recovering scene geometry from a single RGB image [42, 73, 41, 71]. While these learning-based approaches have started producing promising results, their generalization abilities are not well understood. Our work leverages geometric multi-view cues from a structured-light setup, and can be used in a complementary manner to improve the generalization abilities of learning-based approaches.

3. Mathematical Preliminaries

Two-view geometry of structured-light: The Blocks-World Camera is based on a structured-light system, which typically consists of a projector and a camera [45], as shown in Fig. 2 (a). We assume a pinhole projection model for both the camera and the projector, and define the camera and projector coordinate systems (CCS and PCS) centered at $\mathbf{c}_c$ and $\mathbf{c}_p$, the optical centers of the camera and the projector, respectively. $\mathbf{c}_c$ and $\mathbf{c}_p$ are separated by the projector-camera baseline b along the x axis. The world coordinate system (WCS) is assumed to be the same as the CCS centered at $\mathbf{c}_c$, i.e., $\mathbf{c}_c = [0, 0, 0]^T$ and $\mathbf{c}_p = [b, 0, 0]^T$ in the WCS. Without loss of generality, both the camera and the projector are assumed to have the same focal length f. We further assume a rectified system such that the epipolar lines are along the rows of the camera image and projector pattern. These assumptions (same focal length, rectified setup) are made only for ease of exposition, and are relaxed in practice by calibrating the projector-camera setup and rectifying the captured images to this canonical configuration [45].

Plane parameterization: A 3D plane can be characterized by three parameters: $\Pi = \{D, \theta, \varphi\}$, where $D \in [0, \infty)$ is the shortest distance from $\mathbf{c}_c$ to Π, $\theta \in [0, \pi]$ is the polar angle between the plane normal and the $-z$ axis, and $\varphi \in [0, 2\pi)$ is the azimuthal angle from the x axis to the plane normal (clockwise), as shown in Fig. 2 (a). The plane normal is given by: $\mathbf{n} = [\sin\theta\cos\varphi, \sin\theta\sin\varphi, -\cos\theta]^T$.

4. Single-Shot Blocks-World Camera

Structured-light (SL) systems can be broadly classified in two ways. Multi-shot methods such as line striping [62, 4, 14], binary Gray coding [34, 61] or sinusoid phase-shifting [65] require projecting multiple patterns on the scenes. These techniques can achieve high depth-precision, but are not suitable for dynamic scenes. In contrast, single-shot methods [75, 72, 58, 60] require projecting only a single pattern, enabling them to handle scene/camera motion. Furthermore, these methods can be implemented with low-cost single-pattern projectors using a static mask or a diffractive optical element, instead of a full projector that can dynamically change the projected patterns.

In this section, we present *single-shot* Blocks-World Cameras that can estimate *both* depths and surface normals of piece-wise planar scenes with a *single* projected pattern. These cameras have low complexity, both computationally (low-cost algorithms) and for hardware (single-shot).

4.1. What Pattern should be Projected?

The performance of a single-shot SL system is determined by the projected pattern. There are several single-shot SL patterns such as 1D color De Bruijn codes [75, 72], multiple sets of 1D stripes for all-round 3D scanning [17], sparse 2D grid of lines [60, 53], 2D color encoded grids [9, 58], grid patterns with spacings that follow a De Bruijn sequence [68], 2D pseudo-random binary code [70], and 2D random dots (e.g., MS Kinect V1). While these patterns have been designed for explicitly recovering scene depths, our goal is different: directly estimate the plane parameters

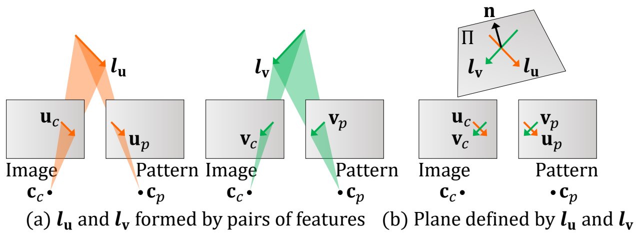


Figure 3. **Plane estimation from a known feature correspondence.** (a) Line segments $\mathbf{u}_p$ and $\mathbf{u}_c$ from image and pattern features create a pair of planes which meet at a 3D line $\mathbf{l}_u$. Similarly, $\mathbf{v}_p$ and $\mathbf{v}_c$ create $\mathbf{l}_v$. (b) $\mathbf{l}_u$ and $\mathbf{l}_v$ define a 3D plane which can be estimated from known image and pattern feature correspondence.

without recovering dense depth maps. Next, we describe the design of a new pattern optimized for achieving this goal.

Pattern design principles: There are two key considerations when designing the pattern. First, for piece-wise planar scenes, a pair of corresponding patches in the projected pattern and the captured images are related via a homography (assuming the patches lie on a single plane). The homography contains sufficient information to uniquely recover the parameters of the 3D scene plane [27], and it preserves straight lines and their intersections. Second, a pattern with a sparse set of features (a small fraction of the projector pixels are on) enables robust and fast correspondence matching, potentially reduced source power with diffractive optical elements and robustness to multi-path interference, a critical issue in SL imaging with dense patterns [22, 21]. On the other hand, sparse single-shot patterns have a trade-off in that for general scenes, they can achieve only sparse 3D reconstructions. However, for piece-wise planar scenes with a relatively small set of dominant planes, scene geometry can be recovered even with sparse patterns.

Based on these two considerations, we design a pattern consisting of a sparse set of identical features distributed spatially. Each feature is cross-shaped, consisting of two intersecting line-segments. For optimal performance, the segments make angles of 45° and 135° with the epipolar line (Fig. 2 (b)). See supplementary report for a detailed discussion. For sufficiently small line segments, the image features in the camera image also have cross shapes (Fig. 2 (b)). These cross-shaped features facilitate robust localization and efficient plane parameter estimation with computationally light-weight algorithms, as discussed next.

4.2. Plane from a Known Correspondence

Consider a pattern feature $\mathrm{P} = \{\mathbf{u}_p, \mathbf{v}_p, \mathbf{p}_p\}$, where $\mathbf{v}_p$ and $\mathbf{u}_p$ are two line vectors and $\mathbf{p}_p$ is the intersection of $\mathbf{v}_p$ and $\mathbf{u}_p$ as shown in Fig. 2 (b). Let the corresponding image feature I be described by $\mathrm{I} = \{\mathbf{u}_c, \mathbf{v}_c, \mathbf{p}_c\}$, where $\mathbf{v}_c$ and $\mathbf{u}_c$ are line vectors corresponding to $\mathbf{v}_p$ and $\mathbf{u}_p$, and $\mathbf{p}_c$ is the intersection of $\mathbf{v}_c$ and $\mathbf{u}_c$. We assume that P lies within a single scene plane, and is completely visible to the camera.

The elements in P and I are described in their own coordinate systems (PCS and CCS, respectively), i.e., for the pattern feature $\mathrm{P} = \{\mathbf{u}_p, \mathbf{v}_p, \mathbf{p}_p\}$,

$$\mathbf{u}_p = [u_{px}, u_y, 0]^T, \mathbf{v}_p = [v_{px}, v_y, 0]^T, \mathbf{p}_p = [p_{px}, p_y, f]^T. \tag{1}$$

For the corresponding image feature $\mathrm{I} = \{\mathbf{u}_c, \mathbf{v}_c, \mathbf{p}_c\}$,

$$\mathbf{u}_c = [u_{cx}, u_y, 0]^T, \mathbf{v}_c = [v_{cx}, v_y, 0]^T, \mathbf{p}_c = [p_{cx}, p_y, f]^T. \tag{2}$$

Then, *if the correspondence is known*, i.e., if pairs of corresponding P and I can be identified, the plane parameters can be recovered analytically by basic geometry, as illustrated in Fig. 3. Specifically, each cross-shaped feature correspondence provides two line correspondences $\{\mathbf{u}_c, \mathbf{u}_p\}$ and $\{\mathbf{v}_c, \mathbf{v}_p\}$, which can be triangulated to estimate two 3D line vectors $\mathbf{l}_\mathbf{u}$ and $\mathbf{l}_\mathbf{v}$, respectively. The plane Π can be estimated from the estimates of $\mathbf{l}_\mathbf{u}$ and $\mathbf{l}_\mathbf{v}$. In particular, the surface normal $\mathbf{n}$ of Π is given as:

$$\mathbf{n} = \frac{((\mathbf{p}_p \times \mathbf{v}_p) \times (\mathbf{p}_c \times \mathbf{v}_c)) \times ((\mathbf{p}_p \times \mathbf{u}_p) \times (\mathbf{p}_c \times \mathbf{u}_c))}{\| ((\mathbf{p}_p \times \mathbf{v}_p) \times (\mathbf{p}_c \times \mathbf{v}_c)) \times ((\mathbf{p}_p \times \mathbf{u}_p) \times (\mathbf{p}_c \times \mathbf{u}_c)) \|}. \tag{3}$$

The shortest distance D from $\mathbf{c}_c$ to Π is:

$$D = \frac{b\mathbf{n}^T \mathbf{p}_p}{p_{px} - p_{cx}} - \mathbf{n}^T \mathbf{c}_p. \tag{4}$$

Given $\mathbf{n}$ and D, depth of $\mathbf{p}_c$ can be computed. See the supplementary report for details and measurable plane space.

Avoiding degenerate solutions: If line correspondences $\{\mathbf{u}_c, \mathbf{u}_p\}$ or $\{\mathbf{v}_c, \mathbf{v}_p\}$ are collinear with epipolar lines, it gives a degenerate solution. To avoid this, the line segments of the features should not be aligned with the epipolar lines.

5. Plane from Unknown Correspondences

As described above, if the feature correspondences are known, the plane parameters can be estimated using Eqs. 3 and 4. One way to achieve this is to place a *single* feature on each epipolar line of the pattern. In this case, for each image feature, the correspondence can be computed trivially. However, this limits the maximum number of pattern features by the number of rows of the pattern. In order to maximize the likelihood of each scene plane being illuminated by a feature, we need to have a sufficiently large density of pattern features, which requires placing *multiple* pattern features on each epipolar line. While this approach increases the feature density, the pattern now consists of multiple *identical* features on each epipolar line, leading to ambiguities. Without additional information or complex global reasoning, it is challenging to find the correct feature correspondences. This presents a tradeoff: Is it possible to perform reconstruction while maintaining both high feature density and low computational complexity?

5.1. Geometric Approach to Correspondence-Free Plane Estimation

In order to address this tradeoff, we develop a novel, light-weight computational approach for estimating plane parameters *without* explicitly computing correspondences between image and pattern features. Let the set of pattern features on one epipolar line of the projected pattern be $\{\mathrm{P}_1, \ldots, \mathrm{P}_N\}$. A subset of these features are mapped to the camera image, resulting in the set of image features $\{\mathrm{I}_1, \ldots, \mathrm{I}_M\}$ $(M \leq N)$ (upper row of Figs. 4 (a) and (b)).

Consider one image feature, say I_1. All the N pattern features are candidate matching features. Each candidate pattern feature results in a plane hypothesis $\Pi = \{D, \theta, \varphi\}$ by triangulating with the image feature I_1. Accordingly, the set of all candidate pattern features $\{\mathrm{P}_1, \ldots, \mathrm{P}_N\}$ create a set of plane hypotheses $\Lambda_1 = \{\Pi_{11}, \ldots, \Pi_{1N}\}$, where Π_{1n} $(n \in \{1, \ldots, N\})$ is the plane parameters computed from I_1 and P_n. Each plane hypothesis can be represented as a point in the 3D *plane parameter space* (we call this the Π-space), as shown in the upper row of Fig. 4 (c). Therefore, the set of plane hypotheses $\Lambda_1 = \{\Pi_{11}, \ldots, \Pi_{1N}\}$ create a *plane parameter locus* in the Π-space. Similarly, we can create another plane parameter locus $\Lambda_2 = \{\Pi_{21}, \ldots, \Pi_{2N}\}$ by pairing I_2 and $\{\mathrm{P}_1, \ldots, \mathrm{P}_N\}$.

Observation 1. The *key observation* is if I_1 and I_2 correspond to scene points on the same scene plane, then two loci Λ_1 and Λ_2 must intersect. If they intersect at a unique location $\hat{\Pi}$ in the Π-space, then $\hat{\Pi}$ is the true plane parameters.

Voting in the plane parameter space: This is a simple, yet powerful observation, which motivates a computationally light-weight voting-based approach for plane estimation that does not require correspondence estimation. For each detected image feature, we compute its plane parameter locus as described above. The locus is the set of candidate planes that the feature votes for. We then collect votes from all the detected image features; the Π-space with loci from all the image features can be considered a likelihood distribution on scene planes. Fig. 5 (b) shows an example of Π-space. Finally, we estimate plane parameters of the dominant scene planes by identifying dominant local peaks in the Π-space. For a given local peak, all the image features that voted for the peak belong to the corresponding plane. For those image features, depth and surface normal values can be computed by plane-ray intersection (Fig. 5 (d)).

This approach is reminiscent of conventional Hough transform-based plane estimation, with two key differences: First, in conventional Hough transform, the planes are estimated from 3D points (each 3D point votes for candidate planes that pass through it), requiring first a 3D point cloud to be computed. In contrast, in our approach, 2D image features directly vote for candidate planes, thus avoiding the potentially expensive point cloud generation. Second, in the conventional approach, each 3D point votes for a dense set of potential planes. Coupled with a large number of 3D points, this can result in large computational and memory costs [47]. On the other hand, in the proposed approach, we use a sparse set of features, and each feature votes for a small, discrete set of candidate planes (e.g., we used < 10 in our experiments). This results in considerably, up to 2 orders of magnitude lower computational costs, especially in scenes with a small number of dominant planes.

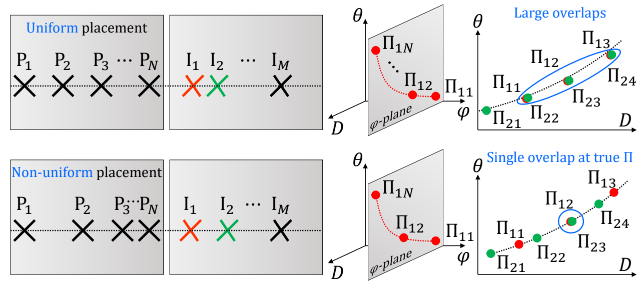


Figure 4. **Plane parameter space with uniformly and non-uniformly spaced pattern features.** (a), (b) N features are placed at (upper row) uniform and (lower row) non-uniform spacing on an epipolar line of the pattern. M of these are imaged as image features. (c) A plane parameter locus is created in the Π-space by pairing an image feature I_1 and all the pattern features on the corresponding epipolar line. The locus is on a plane parallel to the $(D - \theta)$ plane. (d, upper row) Loci corresponding to two different image features lying on the same scene plane have a large overlap with uniform pattern feature distribution, making it impossible to determine the true scene plane containing the features. (d, bottom row) However, for a pattern with non-uniform feature distribution, it is possible to uniquely determine the true scene plane.

5.2. Do Parameter Loci have Unique Intersections?

The voting-based algorithm described above relies on an important assumption: plane parameter loci for different image features corresponding to the same world plane intersect in a *unique* location. If, for example, the loci for all the features on a camera epipolar line overlap at *several* locations, we will not be able to identify unique plane parameters. This raises the following important questions: Does this assumption hold for general scenes? What is the effect, if any, of the pattern design (e.g., the spatial layout of the features)? In order to address these, we describe two key geometric properties of the plane parameter locus.

Property 1. The parameter locus $\Lambda_m = \{\Pi_{m1}, \ldots, \Pi_{mN}\}$ created by pairing an image feature I_m and a set of pattern features $\{\mathrm{P}_1, \ldots, \mathrm{P}_N\}$ on the same epipolar line always lies on a plane parallel to the $\varphi = 0$ plane in the Π-space.

Property 2. Let $\Lambda_m = \{\Pi_{m1}, \ldots, \Pi_{mN}\}$ be the parameter locus created in the same way as Property 1. Let P_μ $(\mu \in \{1, \ldots, N\})$ be the true corresponding pattern feature of I_m. Let $d_{\mu n}$ be the distance between pattern features P_μ and P_n on the epipolar line. Then, the locations of the elements of Λ_m are a function *only* of the *set* $D_\mu = \{d_{\mu n} \,|\, n \in \{1, \ldots, N\}\}$ of relative distances between the true and candidate pattern features.

See supplementary report for proofs. The first property implies that it is possible to recover the azimuth angle of the plane normal from a *single parameter locus*, without computing correspondences. An example is illustrated in the upper row of Figs. 4 (a-c). Since φ is constant across the locus, for the rest of the paper, we visualize parameter loci in 2D $D - \theta$ space, as shown in the upper row of Fig. 4 (d). Note that full 3D Π-space is necessary when differentiating between planes with the same D and θ, but different φ.

The second, perhaps more important, property implies that if the pattern features are *uniformly spaced* on the epipolar line, the resulting loci will *overlap* significantly. This is because of the following: for a uniformly spaced pattern, the set of relative distances (as defined in Property 2) for two distinct pattern features will share several common values. Since the elements of the parameter loci (of the corresponding image features) are determined solely by the set of relative distances, the loci will also share common locations. An example is shown in the upper row of Fig. 4 (d). This is *not* a degenerate case; for uniformly spaced patterns, regardless of the scene, the loci will always have large overlaps, making it impossible to find unique intersections. How can we ensure that different loci have unique intersections?

Patterns with non-uniform feature distribution: The *key idea* is to design patterns with features that are *non-uniformly spaced* across epipolar lines. The lower row of Fig. 4 (a) and (b) show an example, where N pattern features $\{\mathrm{P}_1, \ldots, \mathrm{P}_N\}$ are non-uniformly distributed on an epipolar line, and M of them are imaged as image features $\{\mathrm{I}_1, \ldots \mathrm{I}_M\}$. If this condition is met, the parameter loci do not overlap, except at the true plane parameters, as shown in the lower row of Fig. 4 (d). This enables estimation of the plane parameters even with unknown correspondences.

In our experiments, we placed 7 pattern features non-uniformly on each epipolar line. To ensure robustness against errors in epipolar line estimation, we place features on every k^{th} epipolar line on the pattern. See the supplementary report for details and the resulting patterns.

5.3. Image Feature Localization and Measurement

We localize cross-shaped image features by applying Harris corner detector [26] to the captured image, after thinning morphological operation. Although a single image is sufficient, for scenes with strong texture and lighting variations, we capture two camera frames in rapid succession, with and without the projected pattern, and take their difference. For each candidate feature location, the two line segments of the image feature ($\mathbf{u}_c$ and $\mathbf{v}_c$ in Fig. 2) are extracted. For robustness against projector/camera defocus blur, we extract two edges (positive and negative gradients) from each (possibly blurred) line segment, and compute their average. The line fitting computational routine is fast since it has a closed-form solution. Image feature $\mathrm{I} = \{\mathbf{u}_c, \mathbf{v}_c, \mathbf{p}_c\}$ is then estimated from the two line segments, and their intersection point $\mathbf{p}_c$.

5.4. Toward Higher Memory Efficiency

Blocks-World Cameras are memory-efficient since they do not require capturing and processing dense 3D point clouds. However, the plane parameter Π-space can occupy considerably amount of memory if very small bin sizes are used. We develop a memory-efficient version of Blocks-World Camera algorithm which does not explicitly create a plane parameter voting array. The key observation is that since the Blocks-World Cameras provide a pool of plane candidates with different confidence (e.g., larger number of plane candidates for dominant planes), it is possible to estimate scene planes by finding inliers via a RANSAC-like procedure, instead of voting in the Π-space. See the supplementary report for details of the algorithm and the results.

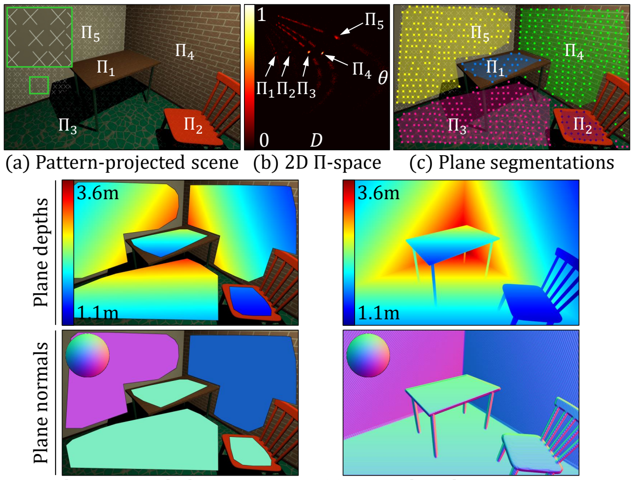
6. Experiments and Results

6.1. Validation by Simulations

We simulate the Blocks-World Camera imaging process with a ray tracing tool [1], using 3D models from an indoor dataset [2]. This allows us to compare the Blocks-World Camera reconstructions with the ground truth, as well as alternate approaches such as plane-fitting to point clouds.

Ground truth comparison: Fig. 5 (a) shows a pattern-projected scene with five dominant planes labeled as Π_1 to Π_5. Plane parameters for these planes are estimated from the Π-space (Fig. 5 (b)). The image features that voted for each dominant plane are identified and segmented to form the plane boundary by their convex hull (Fig. 5 (c)). The proposed approach accurately recovers 3D scene geometry in terms of both depths and surface normals (Fig. 5 (d)).

Comparison with plane-fitting: For evaluating conventional plane-fitting approaches, we simulate a structured-light system that captures a 3D point-cloud of the scene using sinusoid phase-shifting [65]. Fig. 6 (a) shows an example scene with six dominant planes. Fig. 6 (b) and the bottom center of Fig. 1 show the captured depth map and a point cloud. We use 3D Hough transform [8] and RANSAC, two approaches which have been widely used to extract planes from point clouds. We use the randomized version of the 3D Hough transform (RHT) [8] due to its computational efficiency. Figs. 6 (c), (d), and (e) show plane segmentation results by RHT, RANSAC, and Blocks-World Cameras, respectively. To ensure fair comparisons, for plane-fitting approaches, we down-sample the point cloud such that the



(a) Pattern-projected scene (b) 2D Π-space (c) Plane segmentations

(d) Recovered plane geometry (e) Ground truth scene geometry

Figure 5. **Ground truth comparison.** (a) A 3D scene with a projected pattern. (b) 2D Π-space with votes. Dominant planes illustrated at detected peak locations. (c) Plane boundaries formed by identifying image features that voted for the peaks. (d) Recovered plane depths and normals. (e) Ground truth depths and normals.

number of 3D points is the same as the number of image features captured by the Blocks-World Cameras.

For RHT (Fig. 6 (c)), it is challenging to extract small, distant or noisy planes because the votes for these planes are not reliably accumulated by random selection of points. Although RANSAC achieves better plane extraction, both RHT and RANSAC result in erroneous plane segmentation results (e.g., orange and blue points on the walls in Fig. 6 (c) and (d), respectively). This is a common issue with point cloud-based approaches since each 3D point does not have local plane information. In comparison, Blocks-World Cameras achieve accurate plane segmentation since each cross-shaped image feature contains partial information on the plane it belongs to, and does not need global reasoning. See the supplementary report for implementation details for RHT, RANSAC, and the Blocks-World Cameras.

Fig. 7 shows quantitative comparison between the Blocks-World Cameras and the conventional plane-fitting approaches in terms of (a) the accuracy of the extracted plane parameters, and (b) run-time of MATLAB implementations. We used a well-optimized implementation of MSAC (M-estimator sample and consensus) for RANSAC plane-fitting. In run-time comparison, we did not include time to create the point clouds for conventional approaches. RHT estimates the plane parameters accurately, but it fails to find all dominant planes and is slow in run-time. RANSAC is fast and finds all dominant planes robustly, but less accurate in plane parameter estimation. The Blocks-World Cameras can extract the plane parameters well in terms of both accuracy and run-time even without creating the point cloud. See the supplementary report for additional discussions on the trade-off between the run-time and plane estimation accuracy while varying the sampling rate of the 3D point clouds. Comparisons with other structured-light schemes as well as alternate 3D modalities are also discussed in the supplementary report.

6.2. Blocks-World Cameras in-the-Wild

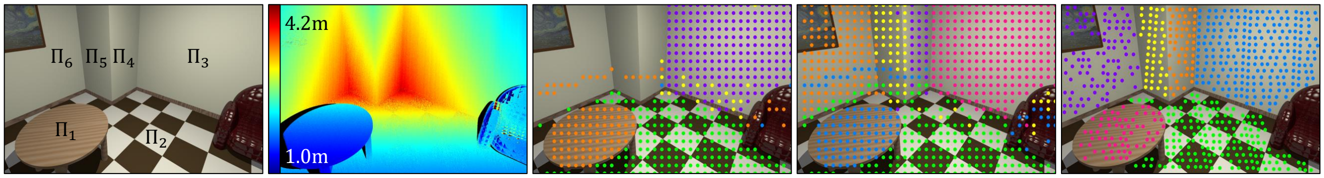
We prototype a Blocks-World Camera using a structured-light system consisting of an Epson 3LCD projector, and a digital SLR camera (Canon EOS 700D). The projector-camera baseline is $353\,\mathrm{mm}$. The system is rectified such that epipolar lines are aligned along the rows of the pattern and the captured image. Using this setup, we validate the performance of Blocks-World Camera with various challenging scenes in the real world.

Scene with large defocus blur: The ability to handle defocus blur is critical for the Blocks-World Cameras when imaging scenes with large depth variations. Our image feature detection algorithm averages the detected line segments for both positive and negative edges as mentioned in Section 5.3, thereby achieving robustness to defocus blur. Fig. 8 (a) shows a scene consisting of planar objects at different distances from the camera. The camera and the projector are focused on the corner between two walls to create a large blur on the rightmost wall just to demonstrate the performance over a wide range of blurs (Fig. 8 (b)). The Blocks-World Cameras can reliably estimate the planes even with blurred features, up to a certain blur size (Fig. 8 (c, d)). For scenes with huge depth variation, the blur size can be reduced by lowering the aperture, using extended depth-of-field approaches, and diffractive optical elements.

Performance under ambient light: Fig. 9 demonstrates the performance of the Blocks-World Cameras under different ambient lighting conditions. Since our approach is based on shape features instead of intensity features, it is robust to photometric variations (photometric calibration is not required) leading to stable plane estimation under different lighting. When ambient light completely overwhelms the projected pattern, the features may not be detected. This issue can be mitigated by narrow-band illumination, spatio-temporal illumination and image coding [25, 51, 50].

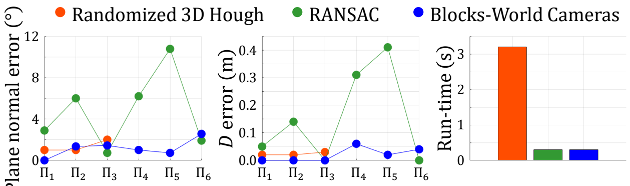
Scene with specular interreflections and strong textures: Fig. 10 (a) shows a scene with a metallic elevator door under strong, directional ambient light (upper), and a picture with complicated textures (lower). The Blocks-World Cameras use geometric features which encode the scene geometry through deformation of the feature shape, and are thus robust to challenging illumination conditions resulting in accurate geometry estimation (Fig. 10 (b, c)).

Non-planar scenes: Although Blocks-World Cameras are designed for piece-wise planar scenes, their performance degrades gracefully for non-planar scenes. Fig. 11 (a) shows a cylindrical object, and the piece-wise planar ap-



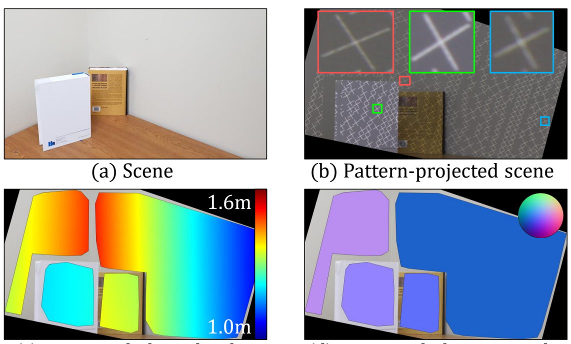
(a) Scene (b) Simulated depth map (c) Randomized 3D Hough (d) RANSAC (e) Blocks-World Cameras

Figure 6. **Comparison with plane-fitting.** (a) A 3D scene. (b) Depth map captured by a simulated structured-light system. (c, d, e) Plane segmentation results by randomized 3D Hough transform, RANSAC, and Blocks-World Cameras. The Blocks-World Cameras achieve more accurate plane segmentation than conventional approaches since each cross-shaped image feature contains local plane information.



(a) Plane parameters error comparison (b) Run-time comparison

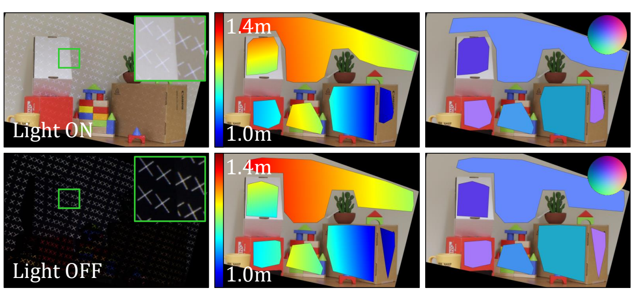
Figure 7. **Quantitative performance comparison.** (a) Plane parameters error comparison. (b) Run-time comparison. Blocks-World Cameras can extract the plane parameters well in terms of both accuracy and run-time even without creating the point cloud.



(a) Scene (b) Pattern-projected scene

(c) Measured plane depths (d) Measured plane normals

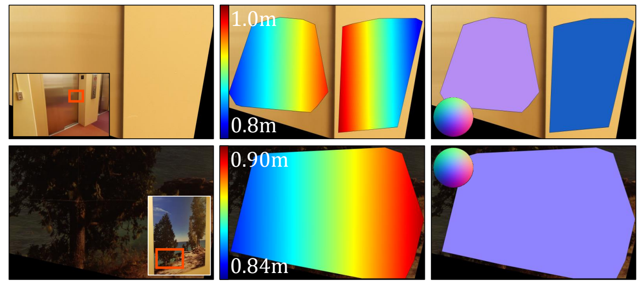
Figure 8. **Robustness to defocus blur.** (a, b) A scene with varying amounts of defocus blur. (c, d) Measured plane depths and normals. Our approach is robust to defocus blur.



(a) Pattern-projected scene (b) Plane depths (c) Plane normals

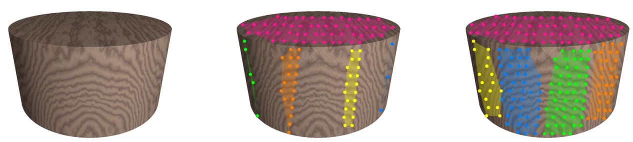
Figure 9. **Robustness to ambient light.** (a) A scene under different indoor lighting conditions. (b, c) Recovered plane depths and normals. Our shape features are robust to photometric variations.

proximation extracted by the proposed approaches. Although only perfectly or nearly planar scene geometry is extracted with relatively smaller bin sizes of Π-space (Fig. 11 (b)), non-planar portions of the scene is approximated with several planes with relatively larger bin sizes (Fig. 11 (c)).



(a) Scenes (b) Plane depths (c) Plane normals

Figure 10. **Robustness to specular reflections and strong textures.** (a) Scenes under challenging illumination conditions with specular reflections and strong textures. (b, c) Reconstructed plane depths and surface normals by Blocks-World Camera.



(a) Cylinder scene (b) Plane estimation with small & large bin sizes

Figure 11. **Approximating non-planar scene with piece-wise planar scene.** (a) Cylinder scene. (b) Plane estimation with relatively small and large bin sizes of Π-space, respectively.

7. Limitations and Future Work

Holes in reconstructions: Due to a sparse set of features in the pattern, the reconstructions have holes in regions where features are absent. An important next step is to develop sensor-fusion systems based on the proposed approach, by leveraging learning-based methods [42, 41] (that produce potentially inaccurate, but dense reconstructions) to generate dense, high-accuracy, hole-free reconstructions.

Non-planar geometric primitives: The proposed approach is designed for reconstructing planar surfaces. A promising line of future work is to design patterns and reconstruction algorithms for non-planar geometric primitives such as spheres, generalized cylinders [6] and geons [5]. Such a generalized Blocks-World Camera will find applications in a considerably broader set of scenarios.

Acknowledgement. This research was supported in parts by ONR grant N00014-16-1-2995, NSF CAREER Award 1943149, and Wisconsin Alumni Research Foundation.

References

[1] http://www.povray.org, 2021 (accessed March 27, 2021). 6

[2] http://www.ignorancia.org/index.php/technical/lightsys/, 2021 (accessed March 27, 2021). 6

[3] Supreeth Achar, Joseph R Bartels, William L'Red' Whittaker, Kiriakos N Kutulakos, and Srinivasa G Narasimhan. Epipolar time-of-flight imaging. *ACM Transactions on Graphics (ToG)*, 36(4):1–8, 2017. 1

[4] G. J. Agin and T. O. Binford. Computer description of curved objects. *IEEE Trans. Comput.*, 25(4):439–449, 1976. 3

[5] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987. 8

[6] T Binford. Visual perception by computer. In *IEEE Conference of Systems and Control*, 1971. 8

[7] András Bódis-Szomorú, Hayko Riemenschneider, and Luc Van Gool. Fast, approximate piecewise-planar modeling based on sparse structure-from-motion and superpixels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 469–476, 2014. 2, 3

[8] Dorit Borrmann, Jan Elseberg, Kai Lingemann, and Andreas Nüchter. The 3d hough transform for plane detection in point clouds: A review and a new accumulator design. *3D Research*, 2(2):3, 2011. 3, 6

[9] K.L. Boyer and AC. Kak. Color-encoded structured light for rapid active ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1):14–28, 1987. 3

[10] D Alex Butler, Shahram Izadi, Otmar Hilliges, David Molyneaux, Steve Hodges, and David Kim. Shake'n'sense: reducing interference for overlapping structured light depth cameras. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1933–1936, 2012. 1

[11] Ricardo Cabral and Yasutaka Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 628–635. IEEE, 2014. 3

[12] Alejo Concha and Javier Civera. Dpptam: Dense piecewise planar tracking and mapping from a monocular sequence. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5686–5693. IEEE, 2015. 2

[13] James M Coughlan and Alan L Yuille. Manhattan world: Compass direction from a single image by bayesian inference. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 941–947. IEEE, 1999. 3

[14] B. Curless and M. Levoy. Better optical triangulation through spacetime analysis. In *Proceedings of IEEE International Conference on Computer Vision*, 1995. 3

[15] Maksym Dzitsiuk, Jürgen Sturm, Robert Maier, Lingni Ma, and Daniel Cremers. De-noising, stabilizing and completing 3d reconstructions on-the-go using plane priors. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3976–3983. IEEE, 2017. 2

[16] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3

[17] R. Furukawa, R. Sagawa, H. Kawasaki, K. Sakashita, Y. Yagi, and N. Asada. One-shot entire shape acquisition method using multiple projectors and cameras. In *Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, pages 107–114, 2010. 3

[18] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Manhattan-world stereo. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1422–1429. IEEE, 2009. 2

[19] David Gallup, Jan-Michael Frahm, and Marc Pollefeys. Piecewise planar and non-planar stereo for urban scene reconstruction. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1418–1425. IEEE, 2010. 3

[20] Abhinav Gupta, Alexei A Efros, and Martial Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *European Conference on Computer Vision*, pages 482–496. Springer, 2010. 2

[21] Mohit Gupta, Amit Agrawal, Ashok Veeraraghavan, and Srinivasa G. Narasimhan. A practical approach to 3d scanning in the presence of interreflections, subsurface scattering and defocus. *International Journal of Computer Vision*, 102(1-3):33–55, 2013. 4

[22] Mohit Gupta and Shree K. Nayar. Micro phase shifting. In *Proc. IEEE CVPR*, 2012. 4

[23] Mohit Gupta, Shree K. Nayar, Matthias Hullin, and Jaime Martin. Phasor Imaging: A Generalization of Correlation Based Time-of-Flight Imaging. *ACM Transactions on Graphics*, 2015. 1

[24] Mohit Gupta, Qi Yin, and Shree K Nayar. Structured light in sunlight. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 545–552, 2013. 1

[25] Mohit Gupta, Qi Yin, and Shree K. Nayar. Structured Light in Sunlight. In *IEEE International Conference on Computer Vision (ICCV)*, pages 545–552, Sydney, Australia, Dec. 2013. IEEE. 7

[26] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 6

[27] R. Hartley and R. Gupta. Computing matched-epipolar projections. In *IEEE CVPR*, pages 549–555, 1993. 4

[28] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 1

[29] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 24(3):577–584, 2015. 2

[30] Dirk Holz and Sven Behnke. Fast range image segmentation and smoothing using approximate surface reconstruction and region growing. In *Intelligent autonomous systems 12*, pages 61–73. Springer, 2013. 3

[31] Thomas Holzmann, Michael Maurer, Friedrich Fraundorfer, and Horst Bischof. Semantically aware urban 3d reconstruction with plane-based regularization. In *Proceedings of the*

European Conference on Computer Vision (ECCV), pages 468–483, 2018. 2

[32] Paul VC Hough. Method and means for recognizing complex patterns, Dec. 18 1962. US Patent 3,069,654. 3

[33] Rostislav Hulik, Michal Spanel, Pavel Smrz, and Zdenek Materna. Continuous plane detection in point-cloud data based on 3d hough transform. *Journal of visual communication and image representation*, 25(1):86–97, 2014. 3

[34] S. Inokuchi, K. Sato, and F. Matsuda. Range imaging system for 3-d object recognition. In *International Conference Pattern Recognition*, pages 806–808, 1984. 3

[35] Hossam Isack and Yuri Boykov. Energy-based geometric multi-model fitting. *International journal of computer vision*, 97(2):123–147, 2012. 3

[36] M. Kaess. Simultaneous localization and mapping with infinite planes. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, page 4605–4611. IEEE, 2015. 2

[37] Takeo Kanade. A theory of origami world. *Artificial Intelligence*, 13:279–311, June 1980. 2

[38] Pyojin Kim, Brian Coltin, and H Jin Kim. Linear rgb-d slam for planar environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 333–348, 2018. 2

[39] Jongho Lee and Mohit Gupta. Stochastic exposure coding for handling multi-tof-camera interference. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7880–7888, 2019. 1

[40] Cheng Lin, Changjian Li, and Wenping Wang. Floorplan-jigsaw: Jointly estimating scene layout and aligning partial scans. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5674–5683, 2019. 3

[41] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4450–4459, 2019. 3, 8

[42] Chen Liu, Jimei Yang, Duygu Ceylan, Ersin Yumer, and Yasutaka Furukawa. Planenet: Piece-wise planar reconstruction from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2579–2588, 2018. 3, 8

[43] Julio Marco, Quercus Hernandez, Adolfo Munoz, Yue Dong, Adrian Jarabo, Min H Kim, Xin Tong, and Diego Gutierrez. Deeptof: off-the-shelf real-time correction of multipath interference in time-of-flight imaging. *ACM Transactions on Graphics (ToG)*, 36(6):1–12, 2017. 1

[44] Branislav Micusik and Jana Kosecka. Piecewise planar city 3d modeling from street view panoramic sequences. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2906–2912. IEEE, 2009. 2

[45] Daniel Moreno and Gabriel Taubin. Simple, accurate, and robust projector-camera calibration. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 464–471. IEEE, 2012. 3

[46] Nikhil Naik, Achuta Kadambi, Christoph Rhemann, Shahram Izadi, Ramesh Raskar, and Sing Bing Kang. A light transport model for mitigating multipath interference in time-of-flight sensors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–81, 2015. 1

[47] Anh Nguyen and Bac Le. 3d point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics (RAM)*, pages 225–230. IEEE, 2013. 5

[48] Sven Oesau, Florent Lafarge, and Pierre Alliez. Planar shape detection and regularization in tandem. In *Computer Graphics Forum*, volume 35, pages 203–215. Wiley Online Library, 2016. 3

[49] Ali Osman Ulusoy, Michael J Black, and Andreas Geiger. Patches, planes and probabilities: A non-local prior for volumetric 3d reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3280–3289, 2016. 2

[50] Matthew O'Toole, Supreeth Achar, Srinivasa G. Narasimhan, and Kiriakos N. Kutulakos. Homogeneous codes for energy-efficient illumination and imaging. *ACM Transactions on Graphics (TOG)*, 34(4):1–13, July 2015. 7

[51] Matthew O'Toole, John Mather, and Kiriakos N Kutulakos. 3D Shape and Indirect Appearance by Structured Light Transport. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3246–3253, 2014. 7

[52] Charalambos Poullis. A framework for automatic modeling from point cloud data. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2563–2575, 2013. 3

[53] M. Proesmans, L. J. Van Gool, and A J. Oosterlinck. Active acquisition of 3d shape for moving objects. In *Proceedings of the International Conference on Image Processing*, volume 3, pages 647–650 vol.3, 1996. 3

[54] Srikumar Ramalingam, Jaishanker K Pillai, Arpit Jain, and Yuichi Taguchi. Manhattan junction catalogue for spatial reasoning of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3065–3072, 2013. 2, 3

[55] Carolina Raposo, Michel Antunes, and Joao P Barreto. Piecewise-planar stereoscan: structure and motion from plane primitives. In *European Conference on Computer Vision*, pages 48–63. Springer, 2014. 2, 3

[56] Carolina Raposo, Miguel Lourenço, Michel Antunes, and João Pedro Barreto. Plane-based odometry using an rgb-d camera. In *BMVC*, 2013. 2

[57] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 1, 2

[58] R. Sagawa, Yuichi Ota, Y. Yagi, R. Furukawa, N. Asada, and H. Kawasaki. Dense 3d reconstruction method using a single pattern for fast moving object. In *Proc. IEEE ICCV*, pages 1779–1786, 2009. 3

[59] Renato F Salas-Moreno, Ben Glocken, Paul HJ Kelly, and Andrew J Davison. Dense planar slam. In *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 157–164. IEEE, 2014. 2

[60] J. Salvi, J. Batlle, and E. Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, 19(11):1055 – 1065, 1998. 3

[61] K. Sato and S. Inokuchi. 3d surface measurement by space encoding range imaging. *Journal of Robotic Systems*, 2(1):27–39, 1985. 3

[62] Yoshiaki Shirai and Motoi Suwa. Recognition of polyhedrons with a range finder. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 80–87, 1971. 3

[63] Shikhar Shrestha, Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. Computational imaging with multi-camera time-of-flight systems. *ACM Transactions on Graphics (ToG)*, 35(4):1–11, 2016. 1

[64] Sudipta Sinha, Drew Steedly, and Rick Szeliski. Piecewise planar stereo for image-based rendering. 2009. 2

[65] V Srinivasan, Hsin-Chu Liu, and Maurice Halioua. Automated phase-measuring profilometry: a phase mapping approach. *Applied optics*, 24(2):185–188, 1985. 3, 6

[66] Yuichi Taguchi, Yong-Dian Jian, Srikumar Ramalingam, and Chen Feng. Point-plane slam for hand-held 3d sensors. In *2013 IEEE International Conference on Robotics and Automation*, pages 5182–5189. IEEE, 2013. 2

[67] Alexander JB Trevor, John G Rogers, and Henrik I Christensen. Planar surface slam with 3d and 2d sensors. In *2012 IEEE International Conference on Robotics and Automation*, pages 3041–3048. IEEE, 2012. 3

[68] A. O. Ulusoy, F. Calakli, and Gabriel Taubin. One-shot scanning using de bruijn spaced grids. In *IEEE ICCV Workshops*, pages 1786–1792, 2009. 3

[69] Cedric Verleysen and Christophe De Vleeschouwer. Piecewise-planar 3d approximation from wide-baseline stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3327–3336, 2016. 2, 3

[70] P. Vuylsteke and A Oosterlinck. Range image acquisition with a single binary-encoded light pattern. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(2):148–164, 1990. 3

[71] Peng Wang, Xiaohui Shen, Bryan Russell, Scott Cohen, Brian Price, and Alan L Yuille. Surge: Surface regularized geometry estimation from a single image. In *Advances in Neural Information Processing Systems*, pages 172–180, 2016. 3

[72] Shuntaro Yamazaki, Akira Nukada, and Masaaki Mochimaru. Hamming color code for dense and robust one-shot 3d scanning. In *Proceedings of the British Machine Vision Conference*, pages 96.1–96.9, 2011. 3

[73] Fengting Yang and Zihan Zhou. Recovering 3d planes from a single image via convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018. 3

[74] Shichao Yang, Yu Song, Michael Kaess, and Sebastian Scherer. Pop-up slam: Semantic monocular plane slam for low-texture environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1222–1229. IEEE, 2016. 2

[75] Li Zhang, Brian Curless, and Steven M. Seitz. Rapid shape acquisition using color structured light and multi-pass dynamic programming. In *IEEE International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 24–36, 2002. 3

[76] Yichao Zhou, Haozhi Qi, Yuexiang Zhai, Qi Sun, Zhili Chen, Li-Yi Wei, and Yi Ma. Learning to reconstruct 3d manhattan wireframes from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7698–7707, 2019. 2, 3