Piecewise-Planar Reconstruction using Two Views

Michel Antunes, João P. Barreto and Urbano Nunes

Abstract

The article describes a reconstruction pipeline that generates piecewise-planar models of man-made environments using two calibrated views. The 3D space is sampled by a set of virtual cut planes that intersect the baseline of the stereo rig and implicitly define possible pixel correspondences across views. The likelihood of these correspondences being true matches is measured using signal symmetry analysis [1], which enables to obtain profile contours of the 3D scene that become lines whenever the virtual cut planes intersect planar surfaces. The detection and estimation of these *lines cuts* is formulated as a global optimization problem over the symmetry matching cost, and pairs of reconstructed lines are used to generate plane hypotheses that serve as input to PEARL clustering [2]. The PEARL algorithm alternates between a discrete optimization step, which merges planar surface hypotheses and discards detections with poor support, and a continuous optimization step, which refines the plane poses taking into account surface slant. The pipeline outputs an accurate semi-dense Piecewise-Planar Reconstruction of the 3D scene. In addition, the input images can be segmented into piecewise-planar regions using a standard labeling formulation for assigning pixels to plane detections. Extensive experiments with both indoor and outdoor stereo pairs show significant improvements over state-of-the-art methods with respect to accuracy and robustness.

Keywords: Piecewise-planar reconstruction, SymStereo, Stereo-Rangefinding, Semi-dense reconstruction

1. Introduction

Stereo cameras are becoming increasingly popular because of the recent advent of 3D visualization and display. A few years ago they were considered special purpose devices that could only be found in research laboratories and high-end equipments, but nowadays they are a consumer electronics product being available either as standalone hand-held cameras (e.g. Fujifilm Finepix 3D and Sony Bloggie 3D), or integrated into smart-phones (e.g. HTC Evo 3D). Our work is motivated by this proliferation of stereo cameras that is expected to create an urge for robust algorithms able to render complete, photo-realistic 3D models in an automatic manner.

Stereo reconstruction is a classical problem in computer and robot vision that deserved the attention of thousands of authors [3, 4]. Despite the many advances in the field, situations of poor texture, variable illumination, severe surface slant or occlusion are still challenging for most stereo matching methods, making it difficult to find a tuning that provides good results under a broad variety of acquisition circumstances [5]. Since man-made environments are dominated by planar surfaces, several authors suggested to overcome the above mentioned difficulties by using the planarity assumption as a prior for stereo reconstruction [6, 7, 8, 9, 10]. These approaches have the advantage of providing piecewise-planar 3D models of the scene that are perceptually pleasing and geometrically simple, and, thus, their rendering, storage and transmission is computationally less complex. This article proposes a pipeline for two-view *Piecewise-Planar Reconstruction* (PPR) understood as the detection and reconstruction of dominant planar surfaces in the scene.[1].

PPR is in a large extent a *chicken-and-egg* problem. If there is accurate 3D evidence about the scene, such as points, lines, vanishing directions, etc, then the problem of detecting, segmenting, and estimating the pose of dominant planes can be potentially solved using standard model fitting techniques [13, 2]. On the other hand, if there is a prior knowledge about dominant planar surfaces in the scene, then the matching process can be constrained to improve the accuracy of the final 3D reconstruction, e.g. the known plane orientations can be used to guide the stereo aggregation [11]. Existing methods for PPR typically comprise three steps that are executed sequentially:

1. *3D Reconstruction:* The objective is to collect 3D evidence about the scene from multiple views. This evidence can either be obtained from *sparse stereo* that matches a sparse set of features across views (e.g. [8, 9]), or from *dense stereo* that performs dense data association between frames by assigning to each pixel a disparity value (e.g. [10]).

2. *Plane Hypotheses Generation:* Given the 3D data, the objective is to detect and estimate the pose of planar surfaces using some sort of multi-model fitting approach.

3. *Plane Labeling:* The goal is to assign to each pixel one of the plane hypotheses generated in the previous step. This is usually done using a *Markov Random Field* (MRF) formulation with photo-consistency being used as data term.

[1] We mean by PPR something that is different from approximating surfaces by small planes, as typically done in several dense stereo methods (e.g. [11, 12])

While most methods were originally designed to receive multiple views [6, 14, 7, 8, 9, 10], we propose a pipeline that uses only two views and makes no assumptions about the scene other than the fact of being dominated by planar surfaces. The novelty is mainly in the steps of *3D Reconstruction* and *Plane Hypothesis Generation*, and the contributions can be summarized as follows:

- *Reconstruction of line cuts using Stereo from Induced Symmetry (SymStereo):* Establishing dense stereo correspondence is computationally expensive specially when dealing with high-resolution images. On the other hand, two-view sparse stereo tends to provide insufficient 3D data for establishing accurate plane hypotheses. Thus, we propose to carry a semi-dense reconstruction of the scene by independently recovering depth along a set of pre-defined virtual planes using SymStereo [1]. Since the intersections of the virtual scan planes with the planar surfaces in the scene are lines, we extract line segments from the profile cuts and use these *line cuts* to generate plane hypotheses.
- *Improving SymStereo accuracy in the case of surface slant:* In a similar manner to what happens in conventional stereo, surface slant affects the depth estimation obtained from SymStereo. In this case, the line cuts are poorly reconstructed and the plane surface estimation is inaccurate. We study the problem of surface slant in the context of the SymStereo framework and devise a simple solution that enables the reconstruction of highly slanted planes.
- *Global plane fitting:* Most methods for PPR treat stereo matching and plane detection in a sequential manner [6, 14, 7, 8, 9, 10]. This is problematic because the accuracy of the plane hypotheses is inevitably limited by the accuracy of the initial 3D reconstruction that does not take into account the fact of the scene being dominated by planar surfaces. We carry the 3D reconstruction and the plane fitting in a simultaneous and integrated manner using the recent PEARL framework proposed in [2]. The algorithm alternates between a global discrete optimization step, which merges plane hypotheses and discards spurious detections, and a continuous optimization step over the symmetry energy, which refines the plane pose estimation taking into account surface slant. The output is a set of plane hypotheses and a semi-dense PPR of the 3D scene, where the reconstructed line cuts are labeled according to the plane detections.

1.1. Related Work

Several works in PPR start by obtaining a sparse 3D reconstruction of the scene (e.g. point clouds, lines, etc), then establish plane hypotheses by applying multi-model fitting to the reconstructed data, and finally use these hypotheses to guide the dense stereo process and/or perform a piecewise-planar segmentation of the input images. In [6], Werner and Zisserman rely in several cues and assumptions to find the dominant surface orientations and use plane-sweeping along the detected normal directions to reconstruct the scene. Bartoli [14] obtains an initial sparse point reconstruction from multiple views and applies a RANSAC-like algorithm for generating and scoring the plane hypotheses. In a similar manner, Pollefeys et al. [7] propose to detect planar surfaces in urban environments from sparse 3D point features and use the estimated normals for guiding plane-sweep stereo. Furukawa et al. [8] reconstruct 3D patches in textured image regions from multiple views using [15], and use the normals of these patches to establish plane hypotheses assuming a Manhattan-world model. These hypotheses are then used in a MRF formulation for pixel-wise plane labeling. In [9], Sinha et al. introduce a probabilistic framework for assigning plane hypotheses to pixels with the evidences of planar surfaces being provided by point cloud reconstruction, matching of line segments, and estimation of vanishing points. Gallup et al. [10] propose a stereo method capable of handling both planar and non-planar objects contained in the scene. A robust procedure based on RANSAC is used for fitting plane hypotheses to dense depth maps, followed by a MRF formulation for plane labeling of the input images.

These pipelines were originally designed to work with multiple images. Moreover, depth estimation and plane fitting are carried in a sequential and decoupled manner that, as discussed previously, has the drawback that errors in 3D evidence affect the accuracy of plane pose estimation, and the inferred planar surfaces are not used for refining the initial depth estimates

An alternative strategy is to over-segment the stereo images based on color information and fit a 3D plane to each non-overlapping region. The number of planes to be considered is defined by the segmentation result, which acts as a smoothness prior during the global optimization. This segmentation information is either used as a hard minimization constraint [16, 17, 18] or as a soft constraint [19]. The main weakness of this type of strategy is the assumption that planar surfaces in the scene have different colors, which is often not the case in most man-made environments (e.g. walls, doors and windows).

There are a few approaches [20, 21, 22] that perform PPR by carrying stereo matching and 3D plane fitting iteratively. The strategy consists in alternating between segmenting the input images into non-overlapping regions and estimating the plane parameters for each region. However, and as stated by the authors of [21], these type of algorithms can become easily stuck in a local minimum whenever they face challenging surface structures e.g. surfaces with low and/or repetitive texture.

1.2. Article Overview and Notation

Section 2 reviews three background concepts that are used throughout the article, namely Stereo from Induced Symmetry [1], energy-based multi-model fitting using PEARL [2], and a global formulation for pixel-wise plane labeling. Section 3 provides an overview of the pipeline for PPR. Section 4 proposes an algorithm for reconstructing line cuts along a single virtual cut plane, while Section 5 shows how these line cuts can be refined in case there is prior slant information available. Then, we present in Section 6 an algorithm for semi-dense PPR that uses the line cuts for posing plane hypotheses and combines SymStereo and PEARL for the final label assignment. Finally, Section 7 reports experiments in PPR, where the accuracy of the plane estimation and pixel labeling is evaluated with respect to ground truth data, and the performance of our pipeline is compared with two different strategies.

We represent scalars in italic, e.g. s, vectors in bold characters, e.g. $\mathbf{p}$, matrices in sans serif font, e.g. M, and image signals in typewriter font, e.g. $\mathtt{I}$. Unless stated otherwise, we use homogeneous coordinates for points and other geometric entities, e.g. a point with non-homogeneous image coordinates (p_1, p_2) is represented by $\mathbf{p} \sim (p_1\, p_2\, 1)^\mathsf{T}$, with $\sim$ denoting equality up to scale.

2. Background

This section briefly reviews background concepts that are used throughout the article, namely *Stereo-Rangefinding* (SRF) using SymStereo (Section 2.1), the energy-based multi-model fitting framework called PEARL (Section 2.2), and a global pixel-wise plane labeling formulation (Section 2.3).

2.1. Stereo-Rangefinding using SymStereo

The SymStereo framework [1] was proposed for matching pixels across stereo views using symmetry analysis instead of traditional photo-consistency. Let $\mathtt{I}$ and $\mathtt{I}'$ be a pair of rectified stereo images and consider a virtual cut plane $\mathbf{\Pi}$ (see Figure 1). The orientation of the virtual plane is arbitrary being the only requirement that it intersects the baseline of the stereo rig. Under such circumstances, the left and right back-projections become reflected one with respect to the other at the locations where the virtual plane intersects the scene. Thus, the sum of both back-projections gives rise to an image signal that is locally symmetric around the *profile cut*, while the subtraction results in a signal that is anti-symmetric. These symmetries are usually not *strict symmetries* due to perspective distortion, surface slant and occlusions, but can be used as cues to recover the profile cut where the virtual plane meets the scene.

Assuming that the world coordinate system is coincident with the reference frame of the left view, the virtual cut plane $\mathbf{\Pi}$ can be represented by the homogeneous vector

$$\mathbf{\Pi} \sim \begin{pmatrix}\mathbf{n}^\mathsf{T} & -h\end{pmatrix}^\mathsf{T}, \qquad (1)$$

where $\mathbf{n}$ indicates the direction orthogonal to the plane

$$\mathbf{n} \sim \begin{pmatrix}n_1 & n_2 & n_3\end{pmatrix}^\mathsf{T}.$$

The homogenous coordinates of the intersection point $\mathbf{O}$ of the virtual cut plane with the baseline is given by [1]:

$$\mathbf{O} \sim \begin{pmatrix}\frac{h}{\mathbf{n}_1} & 0 & 0 & 1\end{pmatrix}^\mathsf{T}.$$

Using β to denote the ratio between O_1 and the baseline length b comes that the plane $\mathbf{\Pi}$ passes between the cameras *iff* the following condition holds

$$0 < (\beta = \frac{O_1}{b}) < 1. \qquad (2)$$

For efficiency purposes, the images do not need to be explicitly back-projected onto the virtual plane $\mathbf{\Pi}$, but instead the homography H induced by $\mathbf{\Pi}$ can be used to map points from the right view into the left view [1]:

$$\mathsf{H} \sim \begin{pmatrix} 1+\frac{bn_1}{h-bn_1} & \frac{bn_2}{h-bn_1} & \frac{bn_3}{h-bn_1} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad (3)$$

Assuming that $\widehat{\mathtt{I}}$ is the warping result of mapping $\mathtt{I}'$ using H, then it comes from the mirroring effect described in [1] that $\mathtt{I}$ and $\widehat{\mathtt{I}}$ are reflected around the image of the profile cut (see Figure 1). Thus, the sum of $\mathtt{I}$ and $\widehat{\mathtt{I}}$ yields an image signal $\mathtt{I}^S$ that is symmetric around the locus where the profile cut is projected. In a similar manner, the difference between $\mathtt{I}$ and $\widehat{\mathtt{I}}$ gives rise to an image signal $\mathtt{I}^A$ that is anti-symmetric at the exact same location. SymStereo detects the image of the profile cut by jointly evaluating symmetry and anti-symmetry in $\mathtt{I}^S$ and $\mathtt{I}^A$. This provides an implicit manner of recovering depth along the scan plane $\mathbf{\Pi}$ that is called *Stereo-Rangefinding* (SRF), in analogy to Laser-Rangefinding that provides depth readings along a scan plane. From [1] it comes that the log$N$ matching cost is the top-performing metric for the purpose of SRF. This cost relies on local frequency analysis for locating symmetric structures by employing a bank of N log-Gabor wavelets (we set $N=10$ in this article). The output of log10 is the joint energy E, where the image of the profile cut is highlighted (see Figure 1).

2.2. Energy-based multi-model fitting using PEARL

Isack and Boykov argued in [2] that methods that greedily search for models with most inliers while ignoring the overall classification of data (e.g. Hough Transform or sequential RANSAC) are a flawed approach to multi-model fitting, and that formulating the fitting as an optimal labeling problem with a global energy function is preferable. In the follow-up of this conclusion, they described the PEARL algorithm that consists in three main steps:

1. Propose an initial set of models (labels) $\mathcal{L}_0$ from the observations
2. Expand the label set for estimating its spatial support (inlier classification)
3. Re-estimate the inlier models by minimizing some error function.

Given the initial model set $\mathcal{L}_0$, the multi-model fitting is cast as a global optimization where each model in $\mathcal{L}_0$ is interpreted as a particular label f. Consider that $d \in \mathcal{D}$ is a data point and that f_d is a particular label in $\mathcal{L}_0$ assigned to d. The objective is to compute the labeling $\mathbf{f} = \{f_d | d \in \mathcal{D}\}$ such that the following energy is minimized:

$$E(\mathbf{f}) = \underbrace{\sum_{d\in\mathcal{D}} D_d(f_d)}_{\text{data term}} + \underbrace{\lambda_S \sum_{d,e\in\mathcal{N}} V_{d,e}(f_d, f_e)}_{\text{smoothness term}} + \underbrace{\lambda_L \cdot |\mathcal{F}_f|}_{\text{label term}}, \qquad (4)$$

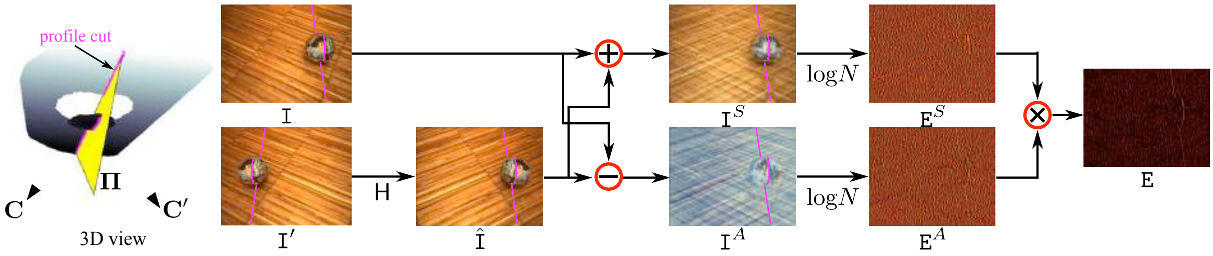


Figure 1: The virtual cut plane $\mathbf{\Pi}$ (yellow) passes between the cameras and intersects the 3D scene in a non-continuous 3D curve (magenta). Let $\widehat{\mathtt{I}}$ be the result of warping $\mathtt{I}'$ by the homography induced by $\mathbf{\Pi}$. The images $\mathtt{I}^S$ and $\mathtt{I}^A$ are, respectively, symmetric and anti-symmetric around the image of the profile cut (magenta). The output of the $logN$ joint symmetry and anti-symmetry quantification method is the energy map $\mathtt{E}$ that highlights the image of the profile cut.

where $\mathcal{N}$ is the neighborhood system considered for d, $D_d(f_d)$ is some error that measures the likelihood of point d belonging to model f_d, and $V_{d,e}$ is the spatial smoothness term that encourages piecewise smooth labeling by penalizing configurations $\mathbf{f}$ that assign to neighboring nodes d and e different labels. The label term is used for describing the data points using as few unique models as possible, with $\mathcal{F}_f$ being the subset of different models assigned to the nodes d by the labeling $\mathbf{f}$ (see [2] for further details). In order to handle outlier data points in $\mathcal{D}$, the outlier label $f_\emptyset$ is added to $\mathcal{L}_0$. Any point d to which is assigned the label $f_\emptyset$ is considered an outlier, and has usually a constant likelihood measure $D_d(f_d = f_\emptyset) = \tau$. The energy of Equation 4 is efficiently minimized using α-expansion [2].

The third step of PEARL consists in re-estimating the model labels f in $\mathcal{L}_0$ given the non-empty set of inliers $\mathbf{D}(f) = \{d \in \mathcal{D} | f_d = f\}$. Let $\mathbf{m}_f$ be the model associated to the label f. Each model $\mathbf{m}_f$ is refined by minimizing the error cost over its parameters:

$$\mathbf{m}_f^* = \min_{\mathbf{m}_f} \sum_{d \in \mathbf{D}(f)} D_d(f).$$

The models with non-empty set in $\mathcal{L}_0$ are replaced with the refined models $\mathbf{m}_f^*$, and the labels with empty set are discarded. The new set of labels $\mathcal{L}_1$ is then used in a new expand step, and we iterate between discrete labeling and plane refinement until the α-expansion optimization does not decrease the energy of Equation 4.

2.3. Pixel-wise Plane Labeling

Given a set of plane hypotheses in the scene, the objective is to assign one of these planes to each pixel of the input images. For this purpose, we use a standard MRF formulation that minimizes an energy involving only data and smoothness terms (the label term in Equation 4 is not considered). The nodes $d \in \mathcal{D}$ are the image pixels, and the labels $f \in \mathcal{P}$ are the plane hypotheses. A 4×4 neighborhood $\mathcal{N}_4$ is assumed for neighboring pixels $\mathbf{d}$ and $\mathbf{e}$, and the data term is defined as

$$D_d(f) = \begin{cases} \min(\rho_d(f), \rho_{max}) & \text{if } f \in \mathcal{P} \\ \gamma \rho_{max} & \text{if } f = f_\emptyset \end{cases} \tag{5}$$

where $\rho_d(f)$ is the photo-consistency between the pixels in the two views put into correspondence by the plane associated to label f. For measuring the photo-consistency we use Zero-mean Normalized Cross-correlation, ρ_{max} is used for handling poorly matching surfaces and γ is a constant parameter. The smoothness term is defined as:

$$V_{d,e}(f_d, f_e) = g \cdot \begin{cases} 0 & \text{if } f_d = f_e \\ T & \text{if } (f_d \vee f_e) = f_\emptyset \\ D' & \text{otherwise} \end{cases}, \tag{6}$$

where

$$D' = \min(D, T) + t$$

and

$$g = \frac{1}{\nabla I^2 + 1}.$$

D is the 3D distance between neighboring points according to their plane labels f_d and f_e, respectively. The parameter t is used for preventing spurious transitions between planes, while T makes the cost robust to depth discontinuities. The measure

$$\nabla I = |I(d) - I(e)|.$$

is the image gradient.

Our formulation is largely standard with the data and the smoothness terms being similar to the ones used in the graph-cut labeling of Gallup et al. [10]. The global assignment herein described will be used in Section 7 for obtaining a dense plane labeling from a semi-dense PPR.

3. Overview of the PPR Pipeline

This section provides an overview of our pipeline for PPR, each element is further described in more detail in the following sections. As depicted in Figure 2, the pipeline receives as input a rectified stereo pair and comprises four parts (the highlighted second and third steps are the main contributions of the paper):

1. The use of SRF, briefly described in Section 2.1, along M virtual cut planes $\mathbf{\Pi}_i$ for computing M joint energies $\mathtt{E}_i$. Each $\mathtt{E}_i$ contains the matching cost of pairs of pixels that are reconstructed on a particular plane $\mathbf{\Pi}_i$.

2. **Detection in each energy $\mathtt{E}_i$ of *line cuts***, which are lines likely to be the intersection between the virtual cut plane and planar surfaces in the scene (refer to Section 4). This is accomplished by first obtaining multiple hypotheses using a very inclusive Hough Transform, followed by PEARL optimization that aims at selecting and refining the position of the most likely line cuts.

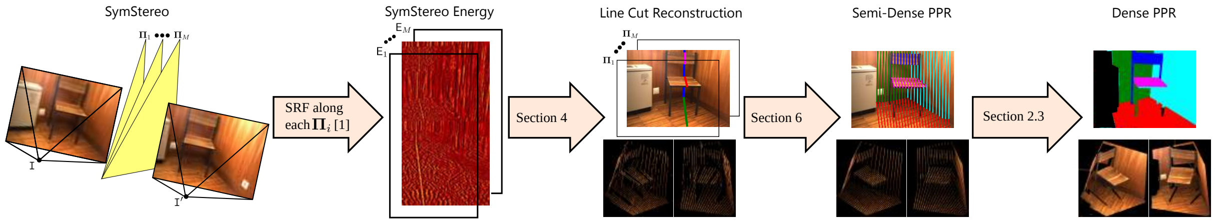


Figure 2: Pipeline for PPR using a pair of calibrated images.

3. **Detection and estimation of 3D planes from line cuts** (see Section 6). We start by establishing multiple planar hypotheses from pairwise combinations of line cuts. A PEARL step is then employed to select the most likely planes and refine their pose taking into account the way slant influences the measurement of the SymStereo energy $\mathtt{E}_i$ (refer to Section 5).

4. Finally, the strongest plane hypotheses are used as input in a standard MRF labeling step that assigns the detected planes to image pixels (refer to Section 2.3). The explanation for doing this is that from the previous step we obtain a semi-dense PPR, and a dense 3D model is a much more pleasant way for showing the reconstruction results. The planes parameters are kept unchanged in this step.

First, it is important to refer that the main contributions of this paper are in the second and third steps of the pipeline (described in Sections 4, 5 and 6). The theory described in the first step was introduced in [1] and is briefly discussed in Section 2.1. The last step concerns a standard MRF that is used for dense plane labeling 2.3, and is only employed for showing the superiority of the obtained 3D reconstructions with respect to two competing approaches (the method **SS** proposed by Sinha et al. [9], and the method **DS** proposed by Gallup et al. [23], refer to Section 7). Both the second and third steps of the pipeline deal with multi-model fitting problems in the sense that they establish model hypotheses, line cuts and planes, respectively, from subsets of the input data. The objective is then to assign to each observation a particular model label such that the joint labeling minimizes some objective function. This is solved using the PEARL framework, described in Section 2.2, that alternates between discrete labeling and continuous model refinement. These multi-model fitting problems could have been solved using a standard Hough transform or sequential RANSAC as done in our previous work for PPR [24]. However, the recent developments (refer to Section 2.2) clearly show that a global approach such as PEARL provide superior results in multi-model fitting problems than greedy clustering techniques. The PEARL framework is important for the quality of the presented results, but the contributions of this paper are not limited to its application.

As described in Section 2.2, the PEARL algorithm consists in three main steps. In the first step, an initial set of model hypotheses is computed, then it involves a discrete labeling step and a continues optimization step. The energy used for the discrete labeling is shown in Equation 4 and consists of a data cost and two regularizers, namely the smoothness and the label costs. The smoothness term is a standard regularizer in computer vision and encourages solutions that are spatially smooth. The insertion of the label cost is explained by the fact that the smoothness term prefers spatially coherent segments, but has no encouragement to combine non-adjacent segments and, thus, to avoid redundant labels in the final assignment. It is this label term that allows to use discrete labeling for purposes of clustering and multi-model fitting. Considering our pipeline depicted in Figure 2, if we would for example skip the third step that involves the PEARL optimization, and directly feed multiple plane hypothesis from pairwise combinations of line cuts into the MRF in step 4, then we would end up with an over-segmented dense PPR and much less accurate plane detections. This occurs because the MRF formulation by itself is unable to change the input label space by either merging planes or refining their pose.

Regarding the continuous refinement of the PEARL algorithm (see Section 2.2), it is necessary because the initial set of model hypotheses is computed from a small set of random data points, and using a larger set of inlier points in an optimization step generally provides better solutions. While in Section 4 the continuous optimization is mostly trivial (regular *Levenberg-Marquardt* (LM) [25]), in Section 6 we use a new type of optimization that is explained in Section 5 and that complements the work of [1].

4. Reconstruction of lines along a single cut plane

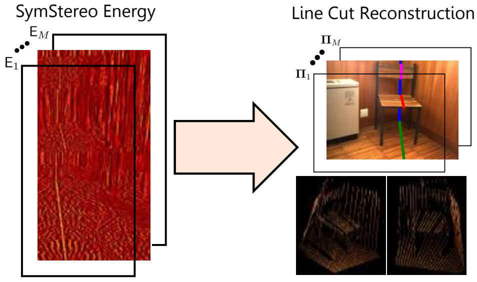


Figure 3: Reconstruction of line cuts. The input is the SymStereo energy along M virtual cut planes (left), refer to Section 2.1, and the output is a sparse 3D line reconstruction (right; top shows the labeling result along one virtual cut plane, where each color identifies a different line cut, and at the bottom are the reconstructed lines along the M cut planes).

There are several algorithms that reconstruct lines in the scene by matching salient image edges across views [9]. In this

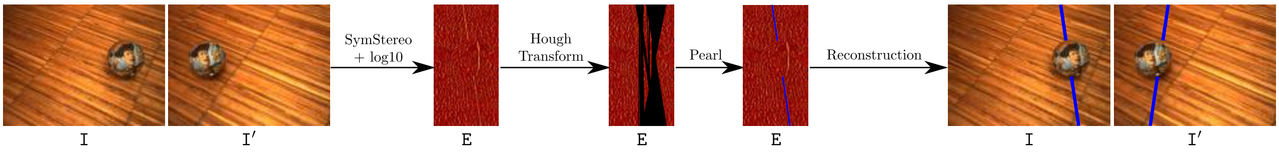


Figure 4: Reconstruction of 3D line cuts from a stereo pair along a virtual cut plane $\mathbf{\Pi}$. We use SymStereo and employ the log10 symmetry-based matching cost for obtaining the joint energy $\mathtt{E}$. The energy $\mathtt{E}$ is used as input to a weighted Hough transform for extracting line cuts (black lines), from which the most appropriate hypotheses (in this example one line cut (blue) is detected) are selected using a global framework constituted by data, smoothness and label costs.

section, we go one step further and show how to use SRF to reconstruct lines that are not distinguishable in the input images. Let $\mathbf{\Pi}$ be a virtual cut plane that intersects the planar surfaces in the scene into different lines, henceforth referred as *line cuts* (e.g. in Figure 4 the cut plane $\mathbf{\Pi}$ meets the floor plane in the blue line cut). The 3D line cuts are projected onto the stereo views into line segments that, in general, cannot be perceived from the input images alone (no visible edges). However, and as discussed in this section, these segments can be reliably detected and estimated from the joint energy $\mathtt{E}$ (see Figure 4).

The algorithm for detection and estimating line cuts from the energy images (refer to Figure 3) needs to be such that:

- there is at most one line cut detection per epipolar line because of the visibility constraint
- long and short lines must be equally well detected to enable the 3D reconstruction of both large and small planes (e.g. the vertical faces of the sidewalks in Figure 19)
- the accuracy in position must be high, otherwise none of the generated plane hypotheses will be close enough to the real 3D planes for step 3 of the pipeline presented in Section 3 to converge correctly.

4.1. Line cut detection using Hough and PEARL

As shown in Figure 1, we use the SymStereo framework along a virtual cut plane $\mathbf{\Pi}$ and employ the log10 symmetry metric for computing the joint energy $\mathtt{E}$. Each pixel in $\mathtt{E}$ provides the matching likelihood of a particular pair of pixels in the stereo views, being an indirect measurement of the occupancy probability in 3D along $\mathbf{\Pi}$. Referring to Figure 4, the energy $\mathtt{E}$ is then used as input to a weighted Hough transform for extracting a set of line cut hypotheses $\mathcal{L}_0$. This is accomplished by selecting the N_H local maxima in the Hough voting space. After obtaining $\mathcal{L}_0$, we formulate the line cut detection as a global labeling problem in a PEARL framework where the objective is to assign to each epipolar line (image row) a line cut hypothesis in $\mathcal{L}_0$. Following the notation of Section 2.2, the data points d of the graph are the epipolar lines, with the size of the set $\mathcal{D}$ being equal to the number of image rows, and the goal is to assign a line segment label f to each epipolar line d. The data term is defined as

$$D_d(f) = \begin{cases} \min(1 - \mathtt{E}(d, x_f), \tau) & \text{if } f \neq f_\emptyset \\ \alpha_\emptyset \tau & \text{otherwise} \end{cases}$$

where $\mathtt{E}(r, c)$ is the joint energy value for row r and column c. The coordinate x_f corresponds to the intersection between the epipolar line d and the line segment $\mathbf{l}_f$ associated to label f. Remark that the truncation parameter τ is used for handling poorly matching surfaces e.g. containing low and/or repetitive textures, while the discard label $f_\emptyset$ indicates that no satisfactory line cut hypothesis can be assigned to d. In this case, the virtual cut plane $\mathbf{\Pi}$ has high probability of not intersecting a planar surface along the epipolar plane associated to d. The smoothness term of neighboring nodes d and e is given by

$$V_{de}(f_d, f_e) = \begin{cases} 0 & \text{if } f_d = f_e \\ \lambda_\emptyset & \text{if } (f_d \vee f_e) = f_\emptyset \\ \frac{1}{\nabla I^2 + 1} & \text{otherwise} \end{cases}$$

where

$$\nabla I = |I(d, x_{f_d}) - I(e, x_{f_e})|$$

is the image gray-scale gradient. No penalization is assigned to neighboring image rows d and e receiving the same label, while in the case one node receives the label $f_\emptyset$, then a non-zero cost $\lambda_\emptyset$ is added to $\mathbf{f}$. The smoothness term V prefers label transitions at locations of larger image gradient (lower smoothness cost), which usually occurs at the boundaries of two different surfaces. We use a constant label term λ_L in Equation 4 for favoring line cut assignments $\mathbf{f}$ with fewer labels.

Finally, and after computing an initial labeling solution $\mathbf{f}$ for nodes d, the line cuts $\mathbf{l}$ are refined by minimizing their parameters over the energies $\mathtt{E}$ via LM:

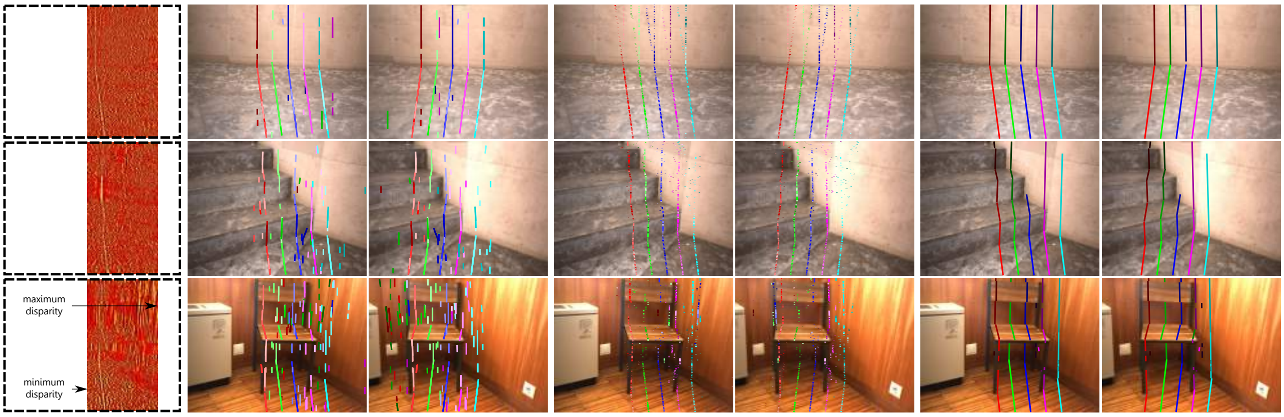
$$\mathbf{l}_f^* = \min_{\mathbf{l}_f} \sum_{d \in \mathbf{D}(f)} (1 - \mathtt{E}(d, x_f)), \tag{7}$$

where $\mathbf{D}(f)$ is a subset of image rows d to which the label f was assigned. Remark that at each solver iteration, the point x_f on d is recomputed according to the current line cut hypothesis $\mathbf{l}_f$. The new set of line cuts $\mathbf{l}_f^*$ are then used in a new global line cut assignment (expand) step, and we iterate between discrete labeling and line cut refinement until the energy of Equation 4 stops decreasing.

4.2. Experiments in line cut detection

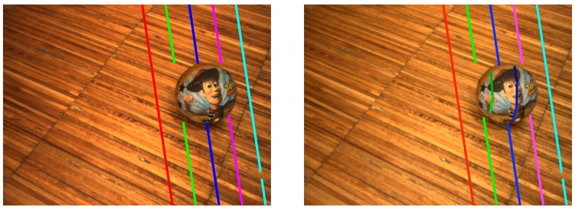
We performed experiments of our line cut detection approach[2] on indoor scenes acquired using a Bumblebee stereo camera from PointGrey, which has a baseline of 24 cm and image resolution of 1024×768 pixels, and compared it against two different strategies (refer to Figure 5).

[2] We used for all the experiments the same parameters: $N_H = 200$, $\lambda_S = 1$, $\tau = 0.8$, $\alpha_\emptyset = 0.7$, $\lambda_\emptyset = 0.9$ and $\lambda_L = 20$, which were empirically selected without much effort.



(a) Energy $\texttt{E}_3$ (b) Ridge detection + HT (c) Weighted HT + Max. per Epipolar line (d) Proposed

Figure 5: Comparison of three line cut detection algorithms: (b) ridge detection over the SymStereo energy $\texttt{E}_i$, followed by Hough Transform (HT) voting (as done in [24]); (c) weighted HT voting using $\texttt{E}_i$ as input, and then assigning to each epipolar line the line cut intersection with highest energy; and (d) proposed algorithm that combines SymStereo and PEARL. Three examples are shown (rows), and each algorithm performs the detections along 5 different virtual cut planes independently. For each example, we show the SymStereo energy computed from the middle virtual cut plane (blue), and the left and right views with the detected line cuts overlaid. For comparing the matching accuracy, note that different colors indicate different cut planes, while different shades identify different line cuts.



(a) $\alpha_\emptyset = 0.7$ and $\lambda_L = 20$ (b) $\alpha_\emptyset = 0.8$ and $\lambda_L = 10$

Figure 6: Results for two different settings of $\alpha_\emptyset$ and λ_L. By varying these parameters, we can control the algorithm to be more permissive with respect to what is considered a line cut (b), while for lower values of $\alpha_\emptyset$ and higher values of λ_L the algorithm only detects line segments with high probability of belonging to planar surfaces (a).

In the first example of Figure 5, the proposed algorithm detected for each virtual scan plane two distinct line cuts: one corresponding to the intersection with the floor, and the other to the intersection with the wall. Comparing these results with the ones obtained using the approaches (b) and (c), we clearly identify the advantages of using a global optimization strategy such as PEARL. First, by formulating the problem as a epipolar line labeling, we implicitly handle the visibility constraint and only assign one detection per image row. Second, the smoothness term of Equation 4 enforces spatial consistency such that much less fragmented detections are obtained, while the label term avoids redundant labels and only two different line cuts are computed for each cut plane. Note that both the approaches (b) and (c) have line cut estimations that are very close to each other. This could be avoided by increasing the suppression neighborhood of the hough peak selection, but this would involve that close planes are discarded (e.g. chair back and wall behind it in the last example). The label cost of PEARL takes care of this issue, and merges models that do not show well separated ridges in $\texttt{E}_i$. These facts are further evidenced in the next examples, where the combination of SymStereo and PEARL shows improved accuracy and consistency in line segment matching when compared to the naive approaches (b) and (c). Concerning the runtime, both methods b) and c) in Figure 5 take about $7-9$ seconds for processing 5 virtual cut planes, while the proposed approach using PEARL takes usually $20-25$ seconds (Matlab implementations). Eventually, we could have selected the lesser accurate methods b) or c) with the hope that the errors are corrected in the third step of the pipeline, which detects and estimates 3D planes from the line cuts. We tested it, but in many cases, the third step cannot improve the precision of the line cuts, and even in cases it is able to achieve it, the time economy in this second step is exceeded by the complexity of the third step that needs to deal with more line cut hypotheses and/or hypotheses with much less precision. Following this, we decided to do the best in both steps.

Besides some minor spurious detections, Figure 5 shows some failures cases: one line cut segment is undetected in the second example (blue cut plane), one line cut in the last example (green cut plane), and two line cuts are computed from the intersection of the same virtual cut plane with the wall in the last example. This mainly occurs because the SymStereo energy $\texttt{E}_i$ is unable to provide well defined energy ridges whenever the cut plane intersects the scene in low-textured regions and slanted surfaces. In these cases, and since there is a low confidence about the location of the image of the profile contour, the algorithm tends to assign the $f_\emptyset$ label or separate the energy contributions. We will show in Section 6 that most of these difficulties are easily handled by our semi-dense PPR pipeline that estimates plane hypotheses from multiple virtual cut planes simultaneously.

Finally, Figure 6 shows an example containing a non-planar object on the top of the floor plane. The algorithm can be either tuned to only detect the line cuts corresponding to the intersection with strict plane surfaces (example (a)), or to approximate the intersection with non-planar surfaces by an appropriate set of line cuts (example (b)). The different tunings are accomplished by manipulating the weighting factor $\alpha_\emptyset$ and the label

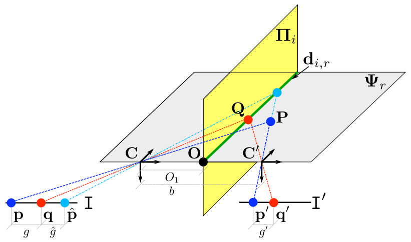


Figure 7: Geometric overview of SymStereo. The camera centers $\mathbf{C}$ and $\mathbf{C}'$ are separated by a distance b (the baseline). The 3D point $\mathbf{Q}$ on $\mathbf{d}_{i,r}$ is detected using the mirroring effect induced by the virtual cut plane $\mathbf{\Pi}_i$ (yellow) intersecting the baseline.

cost λ_L. Using low values of $\alpha_\emptyset$ and high values of λ_L implies that only line cuts belonging to planar surfaces are reconstructed, while higher values of $\alpha_\emptyset$ and low values of λ_L enable to approximate non-planar surfaces by various plausible line cuts. This feature is useful to control the complexity of the final 3D model, by either being strict about the scene geometry, or by enforcing non-planar objects to be reconstructed as a set of planar surfaces.

5. Refinement of line cuts under surface slant

SymStereo was first introduced in [24], with the reader being referred to [1] for a thorough geometric analysis of the framework. This section extends this geometric analysis by studying how surface slant affects the accuracy of symmetry-based matching costs. It is shown that, in a similar manner to multi-directional plane-sweeping [26] where the sweeping direction can be aligned with the surface normal to handle slant [27, 6], in SymStereo it is possible to use prior knowledge about surface orientation to carefully choose the virtual cut planes that render perfect signal symmetries and improve the overall accuracy and robustness of the approach.

Consider the generic points $\mathbf{P}$ and $\mathbf{Q}$ that lie on the same epipolar plane $\mathbf{\Psi}$, as depicted in Figure 7, and assume a virtual cut plane $\mathbf{\Pi}$ that goes through $\mathbf{Q}$. Let $\mathbf{p}$, $\mathbf{p}'$, and $\mathbf{q}$, $\mathbf{q}'$ be the projections of $\mathbf{P}$ and $\mathbf{Q}$ in the left and right views. Since the stereo pair is rectified, the signed distances between the images of the two points are defined as follows:

$$g = p_1 - q_1, \quad g' = p'_1 - q'_1. \tag{8}$$

From the derivations in [1], it comes that, if $\widehat{\mathbf{p}}$ is the result of mapping $\mathbf{p}'$ into the left view using the homography H induced by $\mathbf{\Pi}$, then the following relation holds

$$\widehat{g} = \widehat{p}_1 - q_1 = \left(\frac{\beta}{\beta - 1}\right) g', \tag{9}$$

with β being the ratio of Equation 2. Consider now the stereo disparities $\Delta_p = p_1 - p'_1$ and $\Delta_q = q_1 - q'_1$ of points $\mathbf{P}$ and $\mathbf{Q}$ and define

$$\Delta \, = \, \Delta_p - \Delta_q \, .$$

From Equation 8 it follows that $g' = g - \Delta$, which means that Equation 9 can be re-written as

$$\widehat{g} = \left(\frac{\beta}{\beta - 1}\right) (g \, - \, \Delta). \tag{10}$$

It is important to keep in mind that the symmetry around $\mathbf{q}$ is perfect *iff* the distances g and $\widehat{g}$ are equal with opposite signs ($\widehat{g} = -g$) [1]. In general, this condition is not satisfied, and the energy E tends to spread around the image of the profile cut rather than defining a sharp ridge that enables accurate detection (see Figure 9). The result of Equation 9 suggests that it is possible to enforce $\widehat{g}$ to be equal to g by choosing a suitable ratio β or, in other words, by controlling the location where the virtual cut plane intersects the baseline as a function of the difference in stereo disparity. Assuming that $\mathbf{P}$ and $\mathbf{Q}$ are close points, the difference Δ is directly related with the depth variation in the neighborhood of the 3D profile cut (the surface slant).

5.1. *Slant prior for enhancing SymStereo*

Assume that the points $\mathbf{P}$ and $\mathbf{Q}$ also lie on the same scene plane $\mathbf{\Omega} \sim \begin{pmatrix}\mathbf{m} & -l\end{pmatrix}^\mathsf{T}$, which defines a homography M similar to Equation 3. Using the inverse homography, it can be shown that

$$\Delta_q = \frac{bm_1}{l} q_1 + \frac{bm_2}{l} q_2 + \frac{bm_3}{l}.$$

Since $\mathbf{p}$ is also the projection of the same planar surface, a similar expression can be obtain for Δ_p. Given that $q_2 = p_2$, then Δ_p differs from Δ_q by

$$\Delta = \alpha_1 (p_1 - q_1),$$

where

$$\alpha_1 = \frac{bm_1}{l} \tag{11}$$

encodes the slant of the plane along the horizontal direction. Replacing in Equation 10 comes that

$$\widehat{g} = \left(\frac{\beta}{\beta - 1}\right) (1 - \alpha_1) g. \tag{12}$$

The virtual cut plane $\mathbf{\Pi}$ only affects the symmetry in terms of the intersection point with the baseline. For similar conditions of relative depth variation, any cut plane going through the same point $\mathbf{O}$ generates symmetries with equivalent quality, regardless of its orientation. The conclusion that can be drawn is that having prior knowledge about the position and orientation of the surface to be reconstructed, we can determine the point of intersection between the virtual plane $\mathbf{\Pi}$ and the baseline that grants perfect induced symmetry. The image signals are perfectly symmetric whenever $\widehat{g} = -g$, so that solving with respect to β in Equation 12 yields

$$\beta \; = \; \frac{1}{2 - \alpha_1}. \tag{13}$$

From the analysis above, we propose a simple approach that uses slant information to refine the SymStereo depth estimates

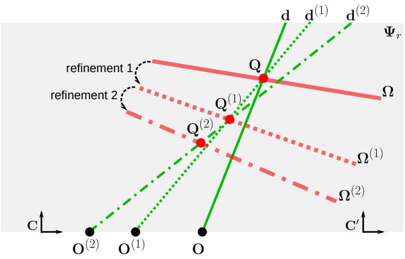
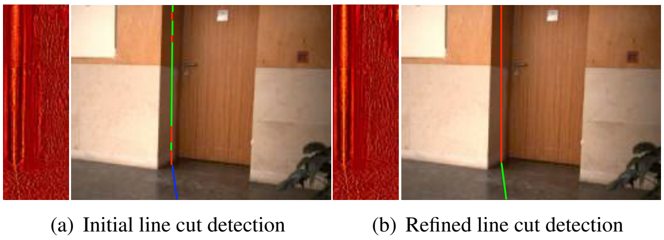


Figure 8: Refinement using slant prior (top view of scene in Figure 7). Assume that $\mathbf{Q}$ lies on the plane $\mathbf{\Omega}$. Then, we can determine the position on the baseline $\mathbf{0}^{(1)}$ (see Equation 13) that improves the induced symmetries. Using the vertical virtual cut plane defined by $\mathbf{0}^{(1)}$ and $\mathbf{Q}$, it is possible to induce new symmetries from which the refined point $\mathbf{Q}^{(1)}$ is estimated.



(a) Initial line cut detection (b) Refined line cut detection

Figure 9: SymStereo refinement. Example (a) shows the energy $\mathtt{E}$ (left) and the line cut detection results overlaid in $\mathtt{I}$ (right), considering a virtual cut plane that intersects the baseline ind its midpoint. Example (b) shows the same experiment, but considering a different virtual cut plane, with β being given by Equation 13 and using an approximate plane estimation of the intersected wall surface. As can be seen, the refined $\mathtt{E}$ in (b) presents a much better defined ridge of the image of the profile cut on the vertical wall.

(see Figure 8). The first step consists in computing the line cuts for a set of virtual scan planes $\mathbf{\Pi}_i$ that go through the midpoint of the baseline ($O_1 = 0.5b$). The line reconstruction tends to be inaccurate in the presence of surface slant, however, and as discussed in the next section, it can be used to obtain a first estimate for the value of α_1. From this estimate, we determine the point $0_1^{(1)} = \beta^{(1)} b$ in the baseline, with $\beta^{(1)}$ being given by Equation 13. A new virtual cut plane $\mathbf{\Pi}_i^{(1)}$ is defined by joining the 3D line cut with $\mathbf{O}^{(1)}$, the corresponding energy E is computed, and finally the line cut is re-estimated. Since the scan plane is chosen such that the induced symmetry is enhanced (see Figure 9), the accuracy of the reconstruction tends to improve. The procedure is repeated till there is no significant change in the positioning of the point in the baseline.

6. PPR using SymStereo and PEARL

This section describes the algorithm for semi-dense PPR that uses the line cuts computed in the previous section for posing plane hypotheses in the scene (refer to Figure 10). The output is a discrete set of planar surfaces and a semi-dense 3D reconstruction. The plane hypotheses can then be used as labels in a global optimization step for obtaining a dense piecewise planar model (Section 2.3).

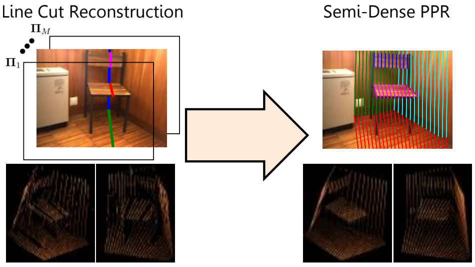


Figure 10: Semi-dense PPR. The input is a set of line cuts computed along M virtual cut planes (left), refer to Section 4, and the output is a semi-dense 3D reconstruction, where each line cut belongs to a particular plane (right top shows the labeling result, where each color corresponds to a different plane, and at the bottom is the semi-dense PPR).

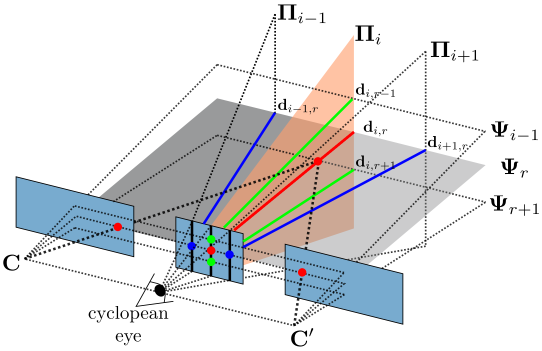


Figure 11: The scene is sampled by a discrete set of virtual cut planes $\mathbf{\Pi}_i$. This can be thought as an image created by a virtual camera that is located between the cameras (cyclopean eye), where each epipolar plane $\mathbf{\Psi}_r$ projects onto one row and each $\mathbf{\Pi}_i$ projects onto one column of the image. Each pixel of the cyclopean eye is originated from the back-projection ray $\mathbf{d}_{i,r}$ (red). The $\mathcal{N}_4$ neighbors of $\mathbf{d}_{i,r}$ are $\mathbf{d}_{i\pm1,r}$ (blue) and $\mathbf{d}_{i,r\pm1}$ (green).

6.1. Formulation of the global framework

Let's consider that the midpoint of the baseline is the projection center of a virtual camera henceforth referred as the cyclopean eye (see Figure 11). The height of the image of the cyclopean eye is equal to the number of epipolar planes $\mathbf{\Psi}_r$ with $r = 1, ..., R$ (one epipolar plane per image row), and the width is given by the number of virtual cut planes $\mathbf{\Pi}_i$ with $i = 1, ..., M$ (one cut plane for each column). Each pixel of the cyclopean eye is associated with a back-projection ray $\mathbf{d}_{i,r}$ that corresponds to the intersection between $\mathbf{\Pi}_i$ and $\mathbf{\Psi}_r$. The objective is to estimate the point on each $\mathbf{d}_{i,r}$ that is mostly like to belong to a planar surface. The problem is casted as a labeling problem following the PEARL framework (Section 2.2). The nodes of the graph are the back-projection rays $\mathbf{d}_{i,r}$ of the cyclopean eye, and the objective is to assign to each $\mathbf{d}_{i,r}$ a plane label f_d. The set of possible labels is $\mathcal{L}_0 = \{\mathcal{P}_0, f_\emptyset\}$, with $f_\emptyset$ meaning that no point on $\mathbf{d}_{i,r}$ belongs to a planar surface. Note that we use $\mathbf{d}$ instead of $\mathbf{d}_{i,r}$ whenever the virtual and epipolar plane specifications are not strictly necessary. We assume a $\mathcal{N}_4$ neighborhood for $\mathbf{d}_{i,r}$ that is defined by the four back-projection rays $\mathbf{d}_{i\pm1,r}$ and $\mathbf{d}_{i,r\pm1}$ (see Figure 11).

The action of PEARL is twofold in the case of semi-dense PPR: first is applied to downsize the number of plane hypotheses by either merging planes that are close to each other and refer to the same 3D surface, or by discarding planes that have little support from the data (discrete optimization with penalty term), and second to refine the pose of the most likely planes taking into account slant (continuous optimization).

6.2. Initial plane hypotheses

As discussed in Section 4, each line cut is a possible location of intersection of a virtual cut plane with a planar surface in the scene. The initial set of plane models $\mathcal{P}_0$ to be used in PEARL can eventually be generated by considering all possible hypotheses that can be obtained from two line cuts belonging to different scan planes $\boldsymbol{\Pi}$, as was originally proposed in [24]. However, and depending on the number of cut planes that are used, the set $\mathcal{P}_0$ can easily become very large. We noticed that using only pairs of line cuts from neighboring cut planes $\boldsymbol{\Pi}_{i\pm(1,2)}$ drastically decreases the size of $\mathcal{P}_0$ and it is typically enough for initializing the piecewise-planar labeling approach. Since it is unlikely that line cuts intersecting different epipolar planes correspond to the same planar surface, we further reduce $\mathcal{P}_0$, and only use pairs of line cuts that have a minimum of N_E epipolar lines of overlap ($N_E = 10$ in this article).

6.3. Data and smoothness term

The data term $D_{\mathbf{d}_{i,r}}$ for the back-projection ray $\mathbf{d}_{i,r}$ is defined as

$$D_{\mathbf{d}_{i,r}}(f) = \begin{cases} \min(1-\mathtt{E}_i(r,x_f),\tau) & \text{if } f\in\mathcal{P}_0 \\ \tau & \text{if } f=f_\emptyset \end{cases}$$

where $\mathtt{E}_i$ is the joint energy associated with the virtual cut plane $\boldsymbol{\Pi}_i$, r is the row corresponding to the epipolar plane $\boldsymbol{\Psi}_r$ and τ is a constant. The coordinate x_f is the column defined by the hypothesis f, corresponding to the intersection of $\mathbf{d}_{i,r}$ with the plane indexed by f. Note that similarly to [10], the non-planar $f_\emptyset$ label indicates that no satisfactory plane hypothesis can be assigned to $\mathbf{d}_{i,r}$.

Inspired by the work of Sinha et al. [9], the smoothness term for neighboring nodes $\mathbf{d}$ and $\mathbf{e}$ is given by

$$V_{\mathbf{de}}(f_\mathbf{d}, f_\mathbf{e}) = \begin{cases} 0 & \text{if } f_\mathbf{d} = f_\mathbf{e} \\ \lambda_1 & \text{if } (\mathbf{d},\mathbf{e},f_\mathbf{d},f_\mathbf{e}) \in \mathsf{S}_1 \\ \lambda_2 & \text{if } (\mathbf{d},\mathbf{e},f_\mathbf{d},f_\mathbf{e}) \in \mathsf{S}_2 \\ \lambda_3 & \text{if } (\mathbf{d},\mathbf{e}) \in \mathsf{S}_3 \\ \lambda_4 & \text{if } (f_\mathbf{d} \vee f_\mathbf{d}) = f_\emptyset \\ 1 & \text{else} \end{cases} \qquad (14)$$

where $0 < \lambda_1 < \lambda_2 < \lambda_3 < 1$, and the content of the sets S_1, S_2 and S_3 is described next. Remark that no penalization is assigned to neighboring nodes receiving the same plane label, while in the case of one node obtaining the discard label $f_\emptyset$, a non-zero cost λ_4 is added to the plane configuration $\mathbf{f}$.

Following a reasoning similar to [9], plane transitions between neighboring nodes $\mathbf{d}$ and $\mathbf{e}$ are more likely to occur in the presence of crease or occlusion edges. A crease edge corresponds to the projection of the 3D line of intersection between two different planes in the scene, while occlusion boundaries arise from spatially separated objects in 3D whose image projections interfere with each other.

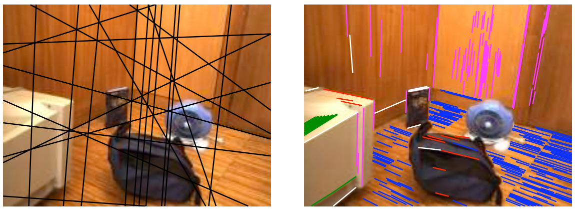

(a) Crease Edges (b) Detected Line Segments

Figure 12: We show in (a) some crease edges obtained from intersections of two different planes in $\mathcal{P}_0$, while in (b) the result of the clustering of concurrent lines is shown. Each group of lines (different groups have different colors) provides a vanishing point location. The white line segments did not receive any vanishing point label.

Let the point $\mathbf{p}_{\mathbf{d},f_\mathbf{d}}$ ($\mathbf{p}_{\mathbf{e},f_\mathbf{e}}$) be the projection of the intersection between the back-projection ray $\mathbf{d}$ ($\mathbf{e}$) and the plane associated to f_d (f_e). In order to encourage plane label transitions at crease edges, we store in the set S_1, the quadruples $(\mathbf{d},\mathbf{e},f_\mathbf{d},f_\mathbf{e})$ in which the points $p_{\mathbf{d},f_d}$ and $p_{\mathbf{e},f_e}$ are located on different sides of the crease edge defined by f_d and f_e. Whenever $\mathbf{f}$ contains assignments located in S_1, then it incurs a penalization λ_1. Figure 12(a) shows some crease edges that are estimated from real imagery.

Occlusion edges usually coincide with visible 2D line segments in the input views and are often aligned with the vanishing directions of scene planes (Figure 12(b)). In order to find possible occlusion boundaries, we detect 2D line segments in the left view $\mathtt{I}$ using the Line Segment Detector [28]. For clustering concurrent lines, we use the global vanishing point detection algorithm proposed by Antunes et al. [29]. The set S_2 contains the quadruples $(\mathbf{d},\mathbf{e},f_\mathbf{d},f_\mathbf{e})$ where the points $p_{\mathbf{d},f_d}$ and $p_{\mathbf{e},f_e}$ are located on different sides of a line segment that was clustered to a particular vanishing point, whose direction is orthogonal either to the planes associated to $f_\mathbf{d}$ or $f_\mathbf{e}$. Finally, S_3 contains the remaining pairs $(\mathbf{d},\mathbf{e})$ whose projections are on different sides of a line segment to which no vanishing point was assigned. Remark that in contrast to [9], we do not perform any line matching between the views, substantially decreasing the complexity of the algorithm.

6.4. Plane refinement

The third step of PEARL (Section 2.2) is to re-estimate the plane model parameters using the inliers of the discrete labeling $\mathbf{f}$. Let $\boldsymbol{\Omega}_f$ be the plane associated to f to which has been assigned a non-empty set of inliers $\mathbf{D}(f) = \{\mathbf{d} \in \mathcal{D} | f_\mathbf{d} = f\}$. Each plane $\boldsymbol{\Omega}_f$ is refined by minimizing its plane parameters over the energies $\mathtt{E}$ via LM:

$$\boldsymbol{\Omega}_f^* = \min_{\boldsymbol{\Omega}_f} \sum_{\mathbf{d}_{i,r}\in\mathbf{D}(f)} \left(1-\mathtt{E}_i(r,x_{\boldsymbol{\Omega}})\right), \qquad (15)$$

where $x_{\boldsymbol{\Omega}}$ is the column defined by the intersection of $\mathbf{d}_{i,r}$ with $\boldsymbol{\Omega}$. The new set of labels $\mathcal{P}_1 = \left\{\boldsymbol{\Omega}_f^*\right\}$ is then used in a new expand step, and we iterate between discrete labeling and plane refinement until the α-expansion optimization does not decrease the energy of Equation 4.

6.5. *Plane refinement after PEARL*

We have discussed in Section 5.1 that SymStereo can be enhanced in case there is slant information available. The output of the global algorithm described previously, is the labeling $\mathbf{f}$ that assigns to each back-projection ray $\mathbf{d}$ a plane $\mathbf{\Omega}$. The intersection of $\mathbf{d}$ with $\mathbf{\Omega}$ defines a 3D point $\mathbf{Q}$, and $\mathbf{\Omega}$ also defines α_1 that encodes the 3D slant in the neighborhood of $\mathbf{Q}$. Following this, the position $\mathbf{Q}$ can be refined by iteratively optimizing β.

Let $\mathbf{\Omega}$ be the plane associated to label f to which has been assigned a non-empty set of inliers $\mathbf{D}(f) = \{\mathbf{d}_{i,r} \in \mathcal{D} | f_{\mathbf{d}_{i,r}} = f\}$, and consider that $\mathbf{Q}_{i,r}$ is the intersection between the ray $\mathbf{d}_{i,r}$ and $\mathbf{\Omega}$ (refer to Figure 8). For each $\mathbf{d}_{i,r}$, we compute the corresponding *ideal* β_1 and obtain a new back-projection ray $\mathbf{d}_{i,r}^{(1)}$. The new ray $\mathbf{d}_{i,r}^{(1)}$ is located on the same epipolar plane, but on the virtual cut plane intersecting the point $\mathbf{O}^{(1)}$ and the previously reconstructed point $\mathbf{Q}_{i,r}$. Given the new plane $\mathbf{\Omega}^{(1)}$, a new homography mapping (see Equation 3) can be used for inducing improved symmetries, and from which the joint energy $\mathtt{E}_{i,r}^{(1)}$ is re-calculated. The new joint energies $\mathtt{E}_{i,r}^{(1)}$ are used in a new refinement step using LM (Section 6.4). We iterate between re-computing new back-projection rays $\mathbf{d}_{i,r}^{(n)}$ and refining $\mathbf{\Omega}^{(n)}$ for a pre-defined number of times (4 in this article).

6.6. *Experiments in semi-dense PPR*

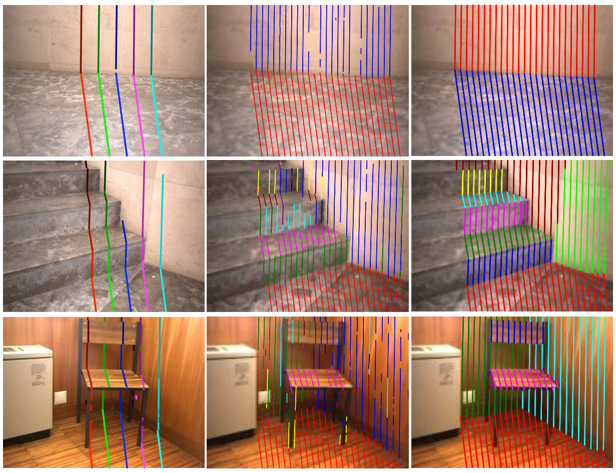

(a) Line Cuts (b) RANSAC (c) Proposed

Figure 13: Comparison between (a) independent line cut reconstruction, (b) sparse PPR using RANSAC (as previously suggested in [24]), and (c) the proposed semi-dense PPR using PEARL. For (a) we show the detection results along 5 virtual cut planes, while for both (b) and (c) the reconstructions were performed along 25 cut planes. Both approaches (b) and (c) receive the same set of line cuts as input. As additional accuracy indicator, we manually identified for all examples the planes that are mutually orthogonal and parallel. The mean angle of the orthogonal planes for (b) is 85.2° and for (c) is 89.1°, while for the parallel planes it is for (b) 1.1° and for (c) 0.5°.

We show in Figure 13 a brief comparison between the line cut reconstruction algorithm presented in Section 4, the sparse PPR approach proposed in [24], and the semi-dense PPR strategy described in this section. In contrast to the proposed PPR approach that uses a PEARL formulation, the algorithm in [24] uses a RANSAC procedure. The RANSAC search is carried in the dual 3D space, and pairs of line cuts generate a plane hypothesis. The inlier set is determined by calculating the Euclidean distance between the line cuts and the plane hypothesis.

Also in this case, the multi-model fitting results obtained using PEARL are superior to the ones obtained using a greedy approach such as RANSAC. First, RANSAC is only able to label the line cuts as being inliers or outliers, so that only a sparse reconstruction composed by line cuts can be obtained (e.g. first example of Figure 13). In our case, the objective is to label all pixels of the cyclopean eye, which enables to estimate a more complete semi-dense PPR. Second, the greedy model selection with largest consensus using RANSAC, independently of the global solution, generates in some cases random models (e.g. second and third example of Figure 13). Third, the smoothness term of PEARL enforces spatial consistency, so that the proposed approach is able to obtain much more coherence in the labeling results, as well as the transitions between planes are consistent and correctly over the image edges. Finally, the continuous optimization of PEARL improves dramatically the accuracy of the plane poses when compared to the ones obtained from pairs of line cuts.

As discussed previously, in cases where the virtual cut planes intersect planar surfaces with some texture and far from object discontinuities (e.g. first example of Figure 13), the independent reconstruction along single virtual planes provides accurate results. However, in scenarios containing multiple planes and complicated textures, the independent line cut reconstruction has some difficulties. These problems are solved using our semi-dense PPR pipeline that estimates planar surfaces in the scene along different virtual cut planes simultaneously and in a global manner.

7. Comparative Experiments in dense PPR

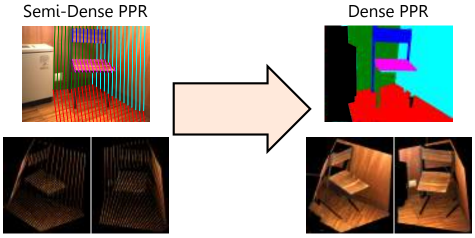


Figure 14: Dense PPR. The input is a set of plane hypotheses (left), refer to Section 6, and the output is a dense 3D reconstruction, where to each image pixel is assigned a particular plane (right; top shows the labeling result, where each color corresponds to a different plane, and at the bottom is the dense PPR).

In this section, we run a set of experiments in PPR with the objective of assessing the performance of the proposed pipeline and compare it against state-of-the-art methods [9, 10]. The evaluation is carried using two datasets: five stereo pairs from the Middlebury benchmark [3, 30, 31] and a new dataset comprising stereo pairs of both indoor and outdoor scenarios (see Figures 16 to 19). These images were acquired using a Bumblebee stereo camera from PointGrey, with a baseline of 24 cm and an image resolution of 1024×768 pixels. The scenes contain mostly planar surfaces, and include a variety of situations that are very challenging for traditional stereo methods e.g. low and/or repetitive textures, surface slant.

7.1. Methods and metrics for benchmarking

The proposed pipeline, henceforth referred as **SymS**, provides as output a set of plane hypotheses $\mathcal{P}^{SymS}$. We compare these plane hypotheses with the ones obtained using two different approaches:

1. *The method **DS** by Gallup et al. [10]:* In this work, the authors start by obtaining a dense depth map of the scene using a local stereo approach. Initial plane hypotheses are generated using a specific RANSAC procedure, which is followed by a linking step that merges planes that are close in distance and eliminates spurious estimates. The output of this algorithm is the discrete set of plane hypotheses $\mathcal{P}^{DS}$.

2. *The method **SS** by Sinha et al. [9]:* This approach starts by obtaining a sparse reconstruction of the scene based on point correspondences, matching line segments and vanishing points. The 3D data is used in successive histogram voting schemes and RANSAC procedures to generate the plane hypotheses. At the end the algorithm provides the discrete set of plane hypotheses $\mathcal{P}^{SS}$.

The experiments compare the pixel labeling results obtained with the MRF formulation described in Section 2.3 when the plane hypotheses are provided by SS, SymS or DS (refer to Figure 14 for a SymS example). Since this analysis is mostly qualitative, we decided to complement it with quantitative measurements of the accuracy of the plane pose estimation. Regarding the Middlebury experiments in Section 7.2, we compute the disparity map for each stereo pair from the estimated plane parameters and compare it with the provided *ground truth* (GT). Regarding the experiments using the new dataset containing real indoor and outdoor scenarios (see Section 7.3 and Section 7.4), it was difficult to obtain the GT model parameters for each stereo pair in the dataset. Thus, we decided to proceed as follows: First, for each stereo pair we manually select the image regions $\mathcal{R}_k$ corresponding to 3D planes $\mathbf{\Omega}_k$ in the scene. Second, given a particular set of plane hypotheses and the corresponding pixel-wise plane labeling $\mathbf{f}$, the accuracy of the pose estimations is evaluated using the following metric:

$$\mathrm{P}_k = \frac{\sum\limits_{\mathbf{p} \in \mathcal{R}_k} \rho_{\mathbf{p}}(f_{\mathbf{p}})}{\#\mathcal{R}_k}, \tag{16}$$

where $\#\mathcal{R}_k$ is the number of pixels in the region.

It is important to note that the accuracy measurements obtained with the above strategy must be interpreted with caution. Given two planes $\mathbf{\Omega}_k$ and $\mathbf{\Omega}_l$, the fact that $\mathrm{P}_k < \mathrm{P}_l$ does not necessarily mean that the former is better estimated than the latter. The proposed metric depends largely on the textures and illumination of the surfaces e.g. planar surfaces with low-texture and specularities will have a large P_k even if the corresponding plane model is well estimated. However, we are in the opinion that the metric P_k is well suited for comparing different estimations of the same plane $\mathbf{\Omega}_k$.

Table 1: Evaluation in Middlebury. The percentage of erroneous pixels in non-occluded regions are presented. We compare the state-of-the-art methods SS and DS, with the intermediate reconstruction results of our pipeline SymS (highlighted in gray). Since the algorithms for line cut detection (Section 4) and semi-dense PPR (Section 6) only recover depth along particular virtual cut planes, we only evaluate the pixels corresponding to the ground truth images of the profile cuts.

Data	Algorithm				
	SS	SymS			DS
		Line Cuts	Semi-dense PPR	Dense PPR	
Wood1	36.6%	8.2%	2.1%	**1.6%**	8.8%
Wood2	32.1%	16.1%	9.9%	**9.6%**	15.2%
Books	41.7%	25.8%	**18.3%**	21.4%	28.9%
Plastic	100%	57.8%	**37.5%**	39.2%	62.9%
Monopoly	51.3%	42.1%	**30.6%**	32.7%	39.8%

The parameters used in the different algorithms were manually tuned using the GT labeling on a subset of the dataset captured using the Bumblebee, whose results are not shown in the experimental comparison. These values were kept constant for all the remaining experiments (including for the experiments in Middlebury). Concerning the SymS algorithm, we decided to use $M = 25$ virtual cut planes to have a good trade-off between accuracy and runtime. The parameters of the final MRF labeling (see Section 2.3) were the same for the three plane hypotheses generators, namely $\rho_{max} = 0.8$, $\gamma = 0.6$, $t = 1$ and $T = 2$.

7.2. Middlebury Experiments

Figure 15 compares the performance of the state-of-the-art methods SS and DS, with the proposed pipeline SymS in generating plane hypotheses for five stereo pairs of Middlebury [3, 30, 31]. We show for each case the left view, the ground truth disparity map, and the estimated pixel-wise plane labeling result. Additionally, we show the disparity maps computed from the estimated plane parameters, and evaluate the numerical accuracy of the dense disparity maps in Table 1. The accuracy of the Line Cuts obtained using SymS is similar to the one obtained using DS and much better than SS. This confirms our first observation that SymStereo is better than sparse stereo for providing 3D evidence about the scene, and similar to dense stereo with lower computational cost, for computing plane models of the scene. The Semi-dense PPR improves the Line Cut results, and shows superior performance when compared to DS. This initial evaluation provides evidence that our PPR algorithm improves the state-of-the-art in reconstructing planar surfaces from two calibrated views. The next sections present the experimental results on real indoor and outdoor man-made environments, which are the scenes mainly targeted by our development.

7.3. Comparison Results on the Bumblebee images

Figure 16 compares the performance of SS, SymS and DS in generating plane hypotheses for several challenging stereo pairs. For each case, we show the left view, the pixel-wise plane

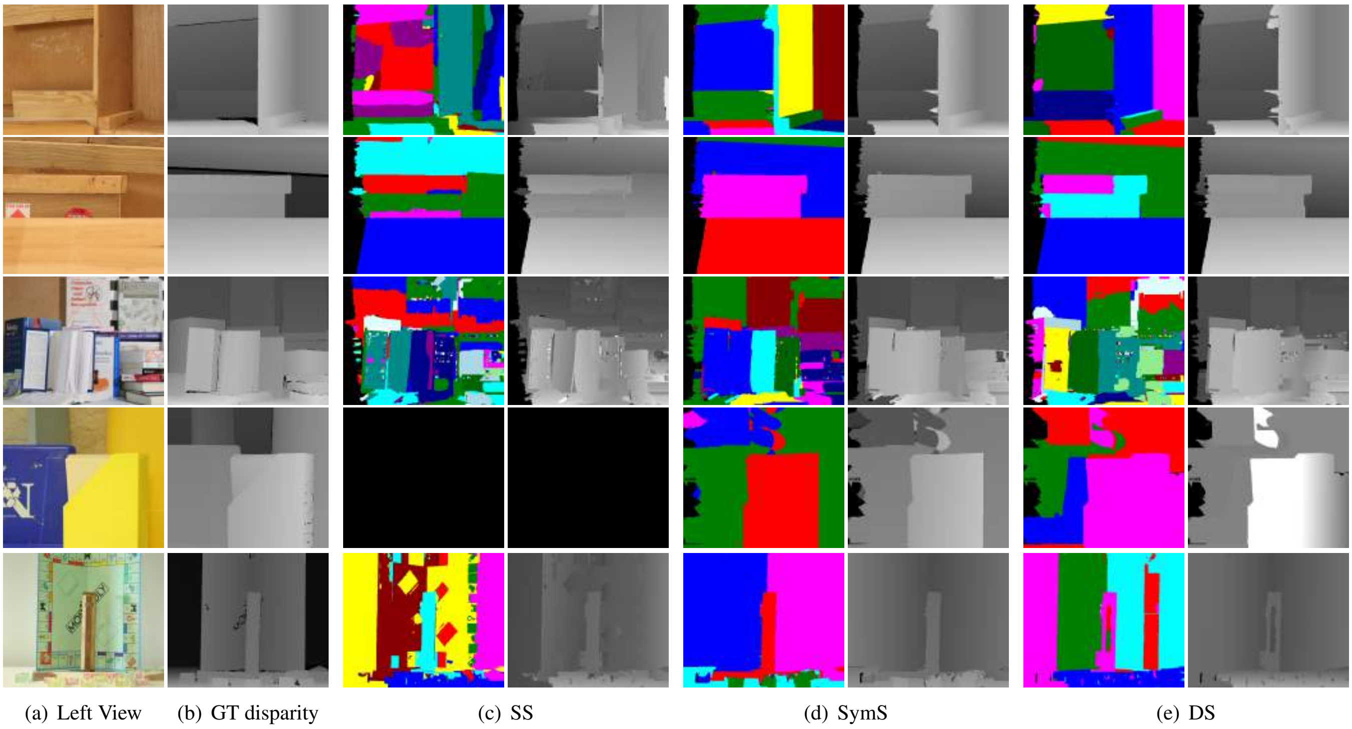


Figure 15: Middlebury results obtained from three different PPR algorithms: SS, SymS and DS. For each algorithm we show the plane labeling (left), where different colors correspond to different planes; and the disparity map (right) computed from the estimated plane parameters.

labeling result, and the P_k scores of the three approaches for different planes in the scene.

In the two first examples, the scene is composed by two and three planes, respectively, which are fronto-parallel to the cameras. It can be observed that the three methods work quite well and provide very similar results. SS shows some difficulty in distinguishing the vertical planes of example (b), which can be explained by the lack of features in the wall on the right.

The two examples in the second row present a highly slanted surface (blue and green planes in examples (c) and (d), respectively). Our algorithm is able to correctly detect and reconstruct these surfaces, whereas DS and SS have clear difficulties in handling such a large amount of slant. The two bottom rows show examples of scenes with difficult textures, slant and variable illumination conditions. SS and DS fail in some cases to provide acceptable plane hypotheses for the MRF labeling, so that no plane assignment is obtained. Our approach recovers all the planes, and can even separate surfaces that are at very close distance, as shown in example (g) where the floor and carpet are distinguished.

For demonstrating the effectiveness of the refinement strategy presented in Section 5, we shown in Figure 17 the improvements that can be achieved in plane pose estimation for an increasing number of refinement iterations. It can be observed that for fronto-parallel configurations the improvements are negligible, since the initial virtual cut planes already intersect the baseline near the optimal point. However, in the case of slanted surfaces, the iterative plane refinement strategy is effective and considerably improves the estimated surfaces.

Finally, and for the sake of completeness, we report the runtimes of each algorithm in these examples, without taking into account the final MRF labeling step. SymS takes between $1-2$ minutes, depending on the number of line cuts that are detected, DS runs in about 2 minutes, and SS takes approximately 3 minutes. All these times refer to straightforward, non-optimized Matlab implementations, with the exception of the α-expansion optimization used by PEARL that was performed using the publicly available C++ code [32, 33, 34, 35].

7.4. Two view piecewise planar models

Figure 18 and Figure 19 show additional piecewise-planar 3D models obtained using our pipeline. The stereo pairs show natural indoor and outdoor scenes that are typically targeted by PPR algorithms. While previous methods, such as the ones reported in [14, 8, 9, 10], require multiple views to obtain satisfactory models, our pipeline is able to reach competitive results using the information of only two views. As discussed in [36], the depth error in stereo relates to the image correspondence error by a multiplicative factor known as the geometric resolution that depends on the baseline and camera focal length. Taking into account our experimental setup, and assuming a maximum allowed error in relative depth of 2%, the maximum reconstruction depth is estimated in 16 meters. Thus, we do not present reconstruction results for surfaces and objects that are beyond this range. It is also important to emphasize that the pixel-wise labeling is exclusively performed based on photo-consistency and image pixel proximity, which largely explains that in some examples the region borders are poorly defined. This issue can be easily solved using a more sophisticated MRF formulation similar to the one used in Section 6 that incorporates crease and

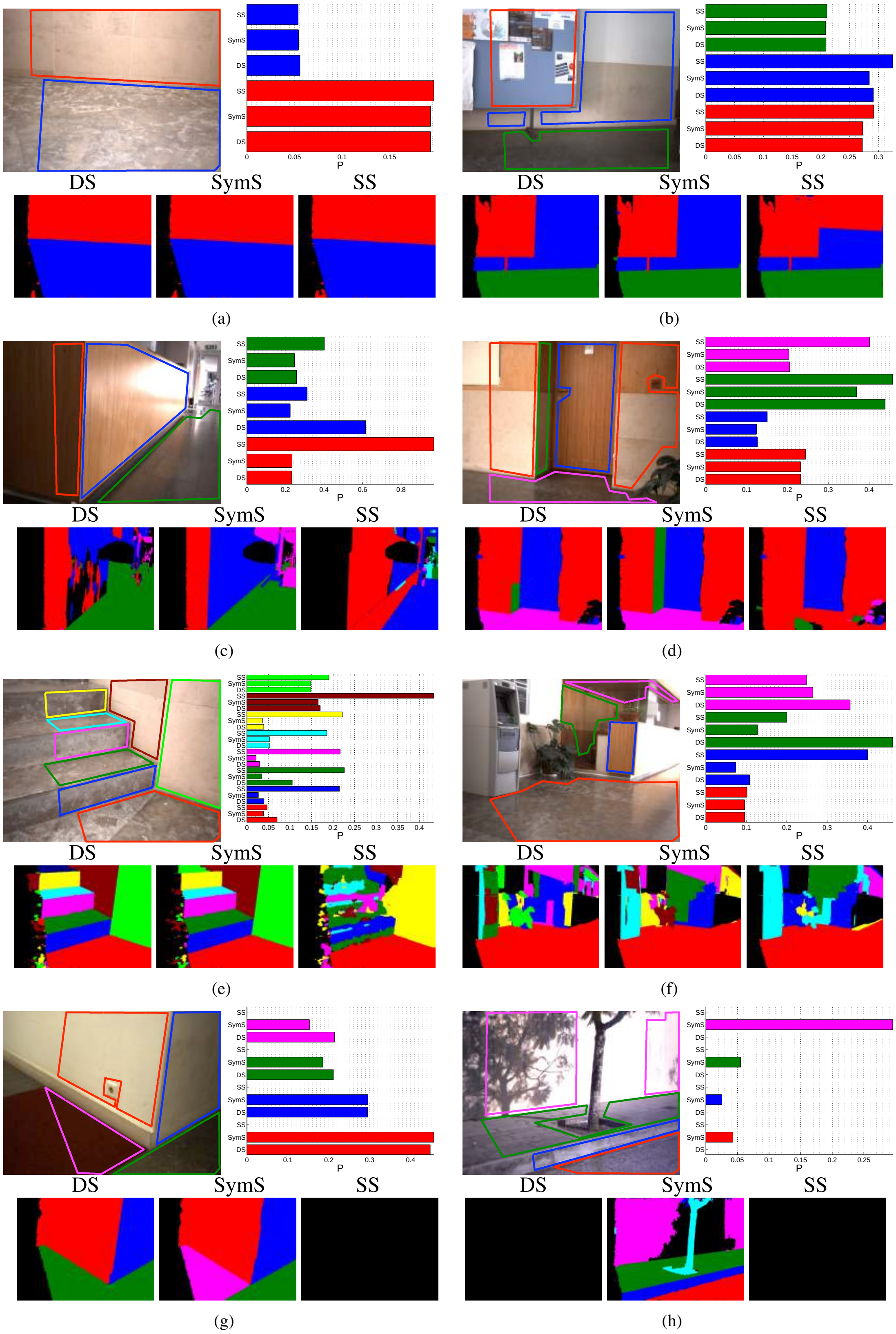


Figure 16: Comparison between DS, SymS and SS for PPR. For each example we show (top, left) I with GT labeling, different colors correspond to different planes; (top, right) mean photo-consistency P in the GT region for each algorithm, each color identifies a particular plane; and (bottom) pixel-wise plane assignment obtained using the different algorithms as plane hypotheses generators, different colors identify different planes. The black label refers to the discard label $f_{\emptyset}$.

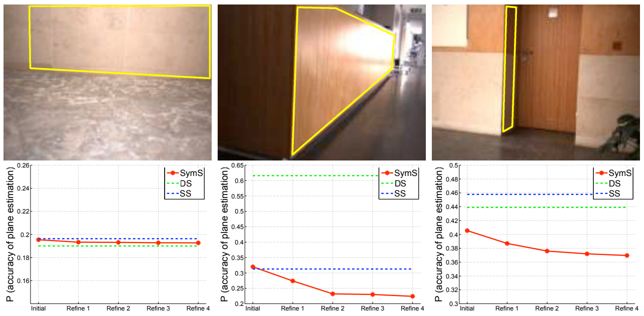


Figure 17: Plane pose refinement by iteratively adjusting the point of intersection of the virtual cut planes with the baseline. The top image shows the left view with the surface being analyzed, and the bottom plot presents the accuracy of the estimation for different SymStereo refinement steps.

occlusion edge information. We chose not to do so in order to better assess the accuracy of our plane pose estimation.

8. Conclusion

The paper presents an automatic piecewise planar reconstruction algorithm from two views. Unlike other existing approaches, the stereo depth estimation and the detection of planar surfaces are accomplished in a tight and coupled manner by combining SymStereo with PEARL [2]. This enables to take full advantage of the strong planarity prior, with the algorithm being able to accurately segment and reconstruct the planes contained in the scene. The effectiveness of the scheme is proved by comparison with two different state-of-the-art approaches in several challenging indoor and outdoor scenarios.

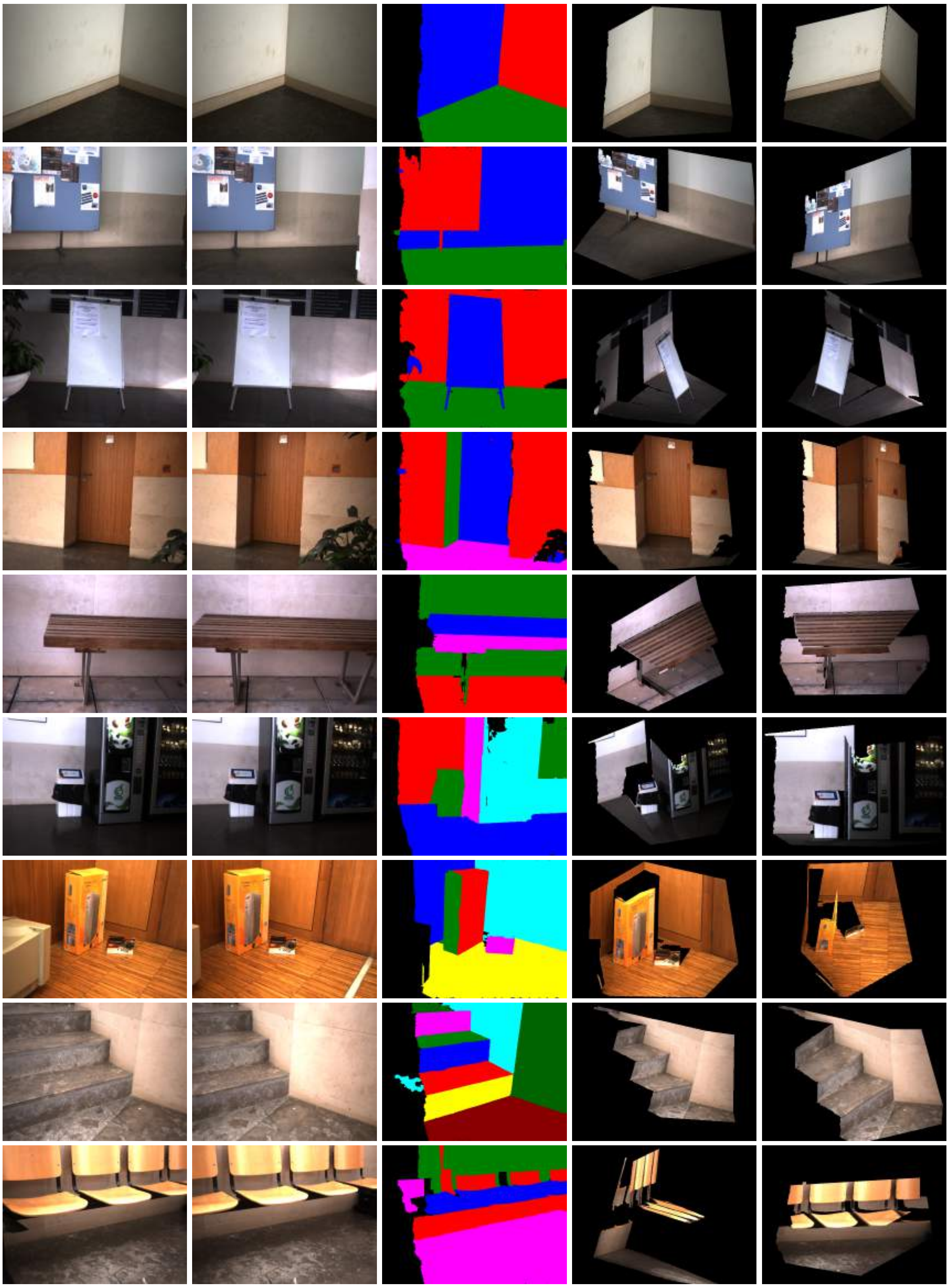
As a final comment, it can be claimed that the energy-based model fitting can either be applied to a dense stereo reconstruction or to a sparse point-cloud model. The former would substantially increase the computational complexity without bringing obvious benefits, while the latter would avoid the use of the smoothness term for regularizing the PEARL energy minimization. Thus, the symmetry-based semi-dense stereo approaches provides the best trade-off between the two, playing a key role in the success of the overall approach.

Acknowledgements

Michel Antunes acknowledges the Portuguese Science Foundation (FCT) that generously funded his work through the grant SFRH/BD/47488/2008. This work was also supported by FCT and the COMPETE program (co-funded by FEDER) under the project grant AMS-HMI12:RECI/EEI-AUT/0181/2012.

References

[1] M. Antunes and J. P. Barreto, “Symstereo: Stereo matching using induced symmetry,” *IJCV*, 2014.
[2] H. Isack and Y. Boykov, “Energy-based geometric multi-model fitting,” *IJCV*, 2012.
[3] D. Scharstein, R. Szeliski, and R. Zabih, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *IJCV*, 2002.
[4] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, “A comparison and evaluation of multi-view stereo reconstruction algorithms,” in *CVPR*, 2006.
[5] R. Klette, N. Kruger, T. Vaudrey, K. Pauwels, M. van Hulle, S. Morales, F. Kandil, R. Haeusler, N. Pugeault, C. Rabe, and M. Lappe, “Performance of correspondence algorithms in vision-based driver assistance using an online image sequence database,” *Vehicular Technology, IEEE Transactions on*, 2011.
[6] T. Werner and A. Zisserman, “New techniques for automated architectural reconstruction from photographs,” in *ECCV*, 2002.
[7] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles, “Detailed real-time urban 3d reconstruction from video,” *IJCV*, 2008.
[8] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, “Manhattan-world stereo,” in *CVPR*, 2009.
[9] S. Sinha, D. Steedly, and R. Szeliski, “Piecewise planar stereo for image-based rendering,” in *ICCV*, 2009.
[10] D. Gallup, J.-M. Frahm, and M. Pollefeys, “Piecewise planar and non-planar stereo for urban scene reconstruction,” *CVPR*, 2010.
[11] Y. Zhang, M. Gong, and Y.-H. Yang, “Local stereo matching with 3d adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy,” in *ICPR*, 2008.
[12] M. Bleyer, C. Rhemann, and C. Rother, “Patchmatch stereo - stereo matching with slanted support windows,” in *BMVC*, 2011.
[13] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, 1981.
[14] A. Bartoli, “A random sampling strategy for piecewise planar scene segmentation,” *CVIU*, 2007.
[15] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *PAMI*, 2010.
[16] A. Klaus, M. Sormann, and K. Karner, “Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure,” in *ICPR*, 2006.
[17] M. H. Lin and C. Tomasi, “Surfaces with occlusions from layered stereo,” *PAMI*, 2004.
[18] M. Bleyer and M. Gelautz, “A layered stereo algorithm using image segmentation and global visibility constraints,” in *ICIP*, 2004.
[19] M. Bleyer, C. Rother, and P. Kohli, “Surface stereo with soft segmentation,” in *CVPR*, 2010.
[20] S. Baker, R. Szeliski, and P. Anandan, “A layered approach to stereo reconstruction,” in *CVPR*, 1998.
[21] S. Birchfield and C. Tomasi, “Multiway cut for stereo and motion with slanted surfaces,” *ICCV*, 1999.
[22] H. Tao, H. S. Sawhney, and R. Kumar, “A global matching framework for stereo computation,” *ICCV*, 2001.
[23] D. Gallup, J.-M. Frahm, and M. Pollefeys, “Piecewise planar and non-planar stereo for urban scene reconstruction,” in *CVPR*.
[24] M. Antunes, J. P. Barreto, and X. Zabulis, “Plane surface detection and reconstruction using induced stereo symmetry,” in *BMVC*, 2011.
[25] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
[26] R. T. Collins, “A space-sweep approach to true multi-image matching,” in *CVPR*, 1996.
[27] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, “Real-time plane-sweeping stereo with multiple sweeping directions,” in *CVPR*, 2007.
[28] R. Grompone von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, “LSD: A Fast Line Segment Detector with a False Detection Control,” *PAMI*, 2010.
[29] M. Antunes and J. P. Barreto, “A global approach for the detection of vanishing points and mutually orthogonal vanishing directions,” in *CVPR*, 2013.
[30] D. Scharstein and C. Pal, “Learning conditional random fields for stereo,” in *CVPR*, 2007.
[31] H. Hirschmüller and D. Scharstein, “Evaluation of stereo matching costs on images with radiometric differences,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009.
[32] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimiza-



(a) Stereo pair (b) Plane labeling (c) Textured 3D reconstruction

Figure 18: Indoor results produced by our PPR algorithm.

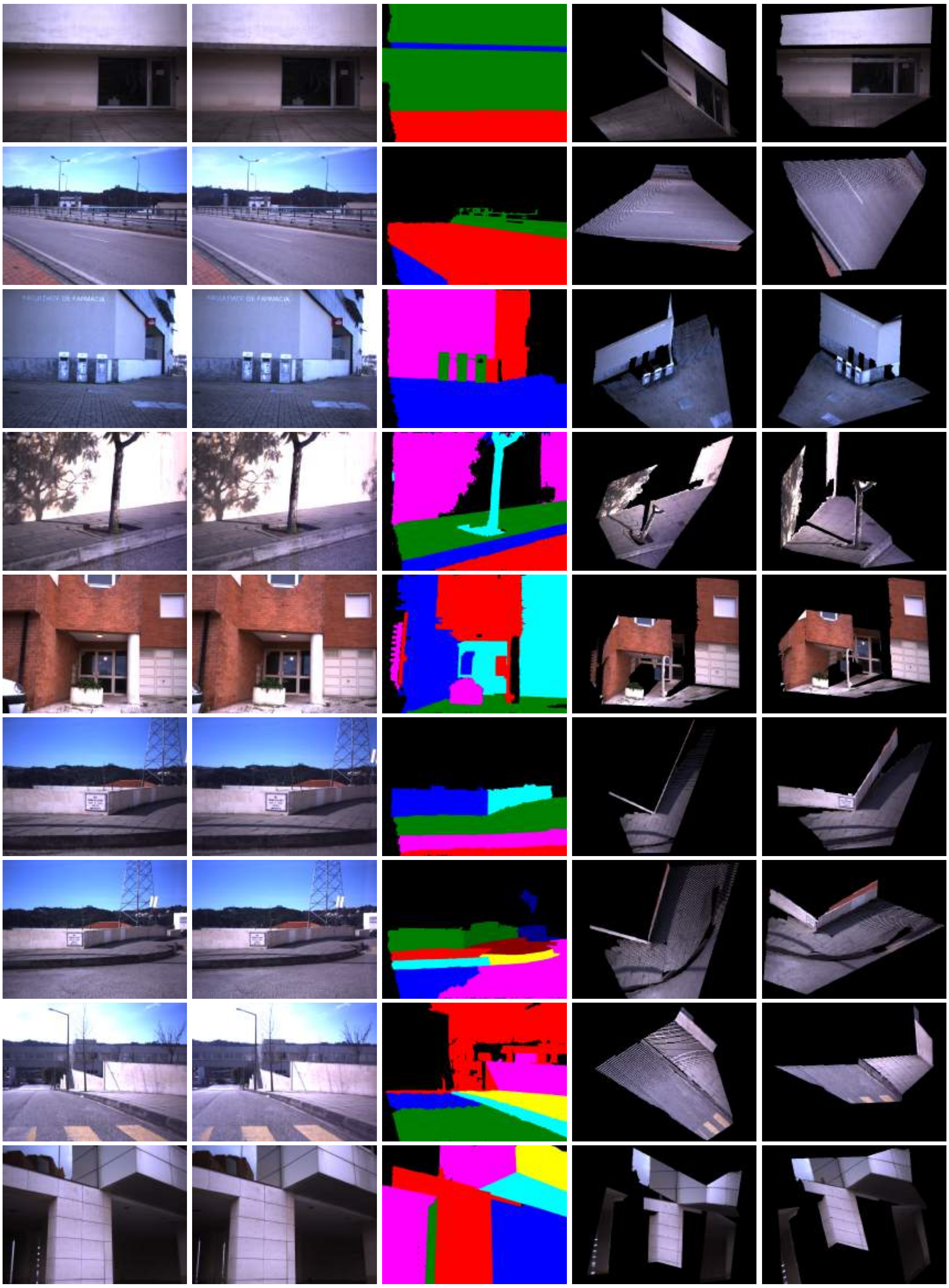
tion via graph cuts," *PAMI*, 2001.

[33] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *PAMI*, 2004.

[34] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *PAMI*, 2004.

[35] A. Delong, A. Osokin, H. N. Isack, and Y. Boykov, "Fast approximate energy minimization with label costs," *IJCV*, 2012.

[36] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *CVPR*, 2008.



(a) Stereo pair

(b) Plane labeling

(c) Textured 3D reconstruction

Figure 19: Outdoor results produced by our PPR algorithm.