2012 Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization & Transmission

Semi-dense piecewise planar stereo reconstruction using SymStereo and PEARL

Michel Antunes and João P. Barreto
Institute of Systems and Robotics,
Faculty of Sciences and Technology,
University of Coimbra,
3030 Coimbra, Portugal
(michel, jpbar)@isr.uc.pt

***Abstract*—We propose an algorithm for two-frame stereo reconstruction that generates an accurate semi-dense piecewise planar model of the 3D scene. First, we use the SymStereo framework [1], [2] for obtaining a semi-dense stereo reconstruction, which exploits symmetry cues for estimating depth along virtual cut planes. Next, the energy-based multi-model fitting algorithm PEARL [3] is applied for estimating the plane models, and labeling the lines of intersection between the virtual and epipolar planes. This procedure involves three steps: (i) generate an initial set of plane hypotheses by extracting line cuts from each virtual cut plane, (ii) inlier classification, and (iii) a plane model refinement step.**

We provide experiments of our approach on a variety of challenging indoor and outdoor stereo pairs. The results show that we successfully detect and reconstruct the planar surfaces contained in the scene, as well as are able to obtain plausible piecewise planar descriptions of non-planar objects. In contrast to existing approaches, the accuracy of our plane hypotheses are not limited by an initial 3D reconstruction, but are iteratively refined after each plane labeling.

***Keywords*-Piecewise planar reconstruction; SymStereo; Semi-dense reconstruction; PEARL**

I. Introduction

Piecewise planar models have recently become popular for the reconstruction of man-made environments [4], [5], [6], [7], [8]. The strong planarity assumption is useful as a prior for stereo reconstruction, helping to overcome difficulties caused by poorly textured surfaces and non-Lambertian reflections. In addition, the computed 3D models are perceptually pleasing and geometrically simple and, thus, their rendering, storage and transmission is computationally less complex. It is important to emphasize that by Piecewise Planar Reconstruction (PPR) we mean detecting and reconstructing dominant plane surfaces in the scene, which is different from approximating surfaces by small planes, as typically done in several dense stereo methods (e.g. [9], [10]).

PPR is in a large extent a ”chicken-and-egg” problem. If the dominant planes in the scene are known ’a priori’, then accurate stereo matching and depth estimation can be easily achieved (e.g. the stereo aggregation can be carried along the known plane orientations [9]). On the other hand, if we have correct depth maps, then the problem of detecting, segmenting, and estimating the pose of dominant planes is relatively trivial. Unlike other methods that treat stereo matching and plane detection sequentially, the algorithm herein proposed accomplishes both goals in a simultaneous and integrated manner. We show how to determine the planar surfaces in the scene while accurately reconstructing the line segments where these surfaces are intersected by a set of pre-defined virtual cut planes (semi-dense PPR). This can be accomplished using only two calibrated images, and without making any assumption about number or relative pose of planes in the scene. In addition, the method can be tuned to be more or less strict about what is considered to be a planar surface, which is a useful feature that enables approximating entire scenes by piecewise planar models.

A. Related Work

Several works in PPR start by obtaining a sparse 3D reconstruction of the scene (e.g. point clouds, edge lines, etc), then establish plane hypothesis 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 [4], [11], [5]. Furukawa et al [6] proposes to perform PPR assuming a Manhattan-world model. They reconstruct 3D patches in textured image regions from multiple views using [12], and use the patches' normals to establish plane surface hypotheses. These hypotheses are then used in a MRF formulation for pixel-wise plane labeling. In [7], Sinha et al. propose a probabilistic framework for assigning plane hypotheses to pixels with the evidences of planar surfaces being provided by point cloud reconstruction, estimation of vanishing lines, and sparse reconstruction of edges. All these works use multiple cues from multiple views to robustly detect the dominant planes in the scene. As discussed in [1], these approaches are not meant for two-view stereo, that would hardly provide enough sparse 3D evidence for supporting the plane hypotheses.

An alternative strategy is to over-segment the stereo images based on color information and fit a 3D plane to each segment region. The number of planes to be considered is defined by the segmentation result, acting as a smoothness prior during global optimization. This segmentation information can be either used as a hard minimization

978-0-7695-4873-9/12 $26.00 © 2012 IEEE
DOI 10.1109/3DIMPVT.2012.49

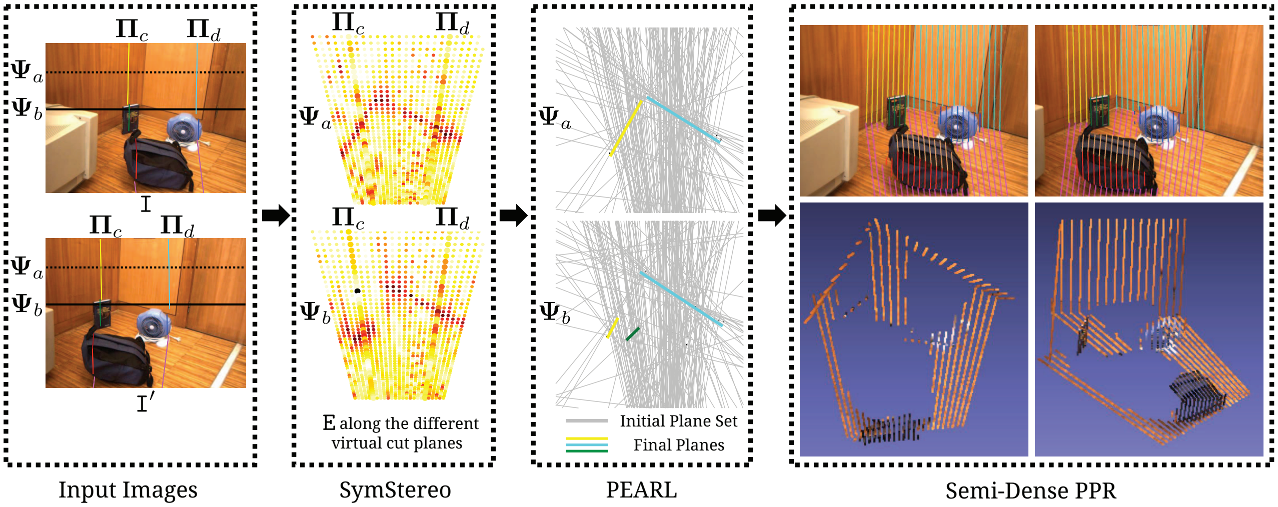


Figure 1. Overview of the algorithm. The input images I and I′ denote, respectively, the left and the right views. The SymStereo framework is applied for obtaining a discrete set of energies E which quantify the occupancy likelihood in 3D in a sparse manner. We show the resulting sparse energy maps for points lying on the epipolar planes $\mathbf{\Psi}_a$ and $\mathbf{\Psi}_b$ (see overlay in the input images). The highlighted energies correspond to the points of intersection with the virtual cut planes $\mathbf{\Pi}_c$ and $\mathbf{\Pi}_d$. The PEARL approach is used for estimating the different planes contained in the scene. We show in grey the lines of intersection between the initial plane label space for PEARL and the epipolar planes $\mathbf{\Psi}_a$ and $\mathbf{\Psi}_b$. A MRF-based regularization framework is used to assign plane labels to each back-projection ray $\mathbf{b}_{q,r}$ of the cyclopean eye, corresponding to the intersection of the virtual cut plane $\mathbf{\Pi}_q$ and the epipolar plane $\mathbf{\Psi}_r$. After the plane labeling, the inlier planes are re-fined based on the SymStereo energies E. We iterate between discrete plane labeling and continuous plane optimization until the energy of Equation 1 stops decreasing. The output of our algorithm is an accurate semi-dense piecewise planar description of the 3D scene. We show on the right the input images with the final line cuts overlaid, where each color corresponds to a different plane; and two views of the final semi-dense 3D reconstruction.

constraint [13], [14], [15] or as a soft constraint [16]. The main weakness of this 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, corridors, doors, windows, etc).

Gallup et al [8] propose a stereo method capable of handling both planar and non-planar objects contained in the scene. It uses a robust procedure for fitting planes hypotheses to dense depth maps, that minimizes the chances of a wrong plane labeling. Although the approach was originally meant for multi-view stereo, we tested it in two-view stereo obtaining reasonable results. The main drawback is the fact that depth estimation and plane fitting are carried in a sequential, decoupled manner. The errors in the stereo matching affect the accuracy of plane pose estimation, and the plane surface assumption is not used to refine the initial depth estimates.

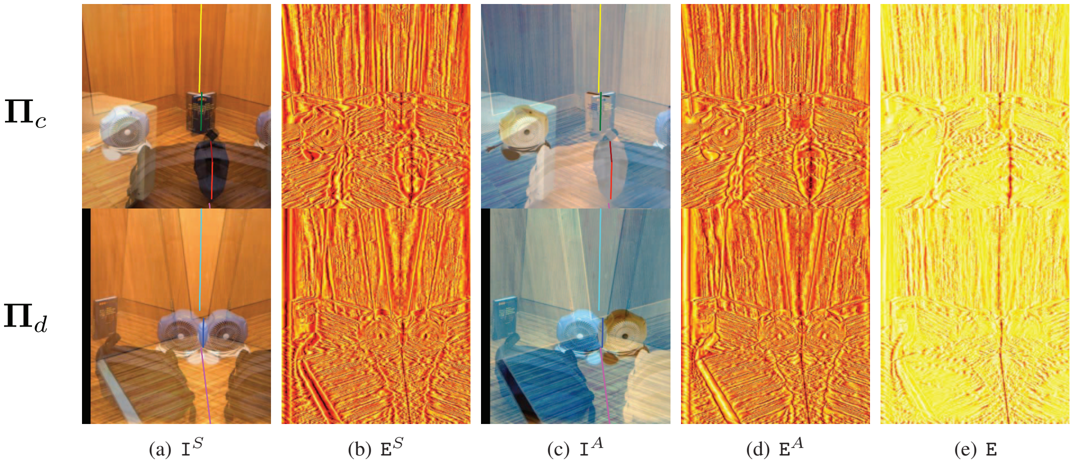
Antunes et al. have recently proposed a stereo approach that is specially effective for recovering depth along arbitrary virtual planes passing in-between the cameras [2]. The framework, named SymStereo, has been employed in [1] for the purpose of PPR using just two images. The strategy consists in probing the work volume by a discrete set of virtual planes, and reconstructing the contours where these planes intersect the scene structure. Since in the case of planar surfaces these curves are straight lines, they proceed by extracting line segments with Hough transform, followed by clustering the reconstructed line cuts into planes using a RANSAC-like approach. The method has the advantage of avoiding dense disparity estimation, which helps to keep the computation tractable whenever handling high-resolution images [2]. However, and in a similar manner to [8], the depth estimation and the planar surface detection are carried in a sequentially, decoupled manner.

In [17], Birchfield and Tomasi describe a method for carrying stereo matching and 3D plane fitting in simultaneous. The strategy consists in alternating between segmenting the input images into non-overlapping regions and finding the affine parameters describing the disparity function of each region. As acknowledged by the authors, the algorithm can become easily stuck in a local minimum whenever it faces low-textured surfaces.

In this article, we combine the semi-dense stereo reconstruction described in [1], [2], which uses symmetry energy for estimating the depth along virtual cut planes, with the energy-based model fitting recently proposed in Isack and Boykov [3]. The main advantage of our algorithm is that it enables to refine the plane hypothesis after each plane labeling, which means that, unlike [8] or [1], the accuracy of the PPR is not limited by the initial dense disparity maps or line cut reconstruction, respectively.

B. *Overview of the algorithm*

The algorithm presented in this paper is briefly summarized in Figure 1. The input is a pair of rectified stereo images. We use the SymStereo framework (Section I-B) and



(a) $\mathtt{I}^S$ (b) $\mathtt{E}^S$ (c) $\mathtt{I}^A$ (d) $\mathtt{E}^A$ (e) $\mathtt{E}$

Figure 2. SymStereo: Let $\widehat{\mathtt{I}}$ be the result of warping $\mathtt{I}'$ by the plane-homography induced by a virtual cut plane $\mathbf{\Pi}$ intersecting the baseline (see $\mathbf{\Pi}_c$ and $\mathbf{\Pi}_d$ in Figure 1). The image signals $\mathtt{I}^S$ and $\mathtt{I}^A$, obtained by adding and subtracting $\mathtt{I}$ with $\widehat{\mathtt{I}}$, are respectively symmetric and anti-symmetric around the image of the profile cut (colored line segments in (a) and (c)). Remark that the intersections with the locus where the profile cut is projected can be identified with almost no ambiguity by searching common pixel locations for which $\mathtt{I}^S$ and $\mathtt{I}^A$ are respectively locally symmetric and anti-symmetric along the epipolar lines. (b) The symmetry energy $\mathtt{E}^S$, (d) the anti-symmetry energy $\mathtt{E}^A$, and (e) the final joint energy $\mathtt{E}$ is computed by pixel-wise multiplication of $\mathtt{E}^S$ and $\mathtt{E}^A$ in order to highlight pixel locations for which both symmetry and anti-symmetry arise (low to high energy varies from white, yellow to shades of red and black).

employ a sparse set of N virtual cut planes intersecting the baseline in its midpoint for obtaining the energy $\mathtt{E}$ for each virtual plane. The approach can be understood as follows: consider a cyclopean eye located between the cameras that scans the 3D scene by a discrete set of virtual cut planes. For a particular virtual scan plane, the likelihood of each point location being occupied is quantified by an energy $\mathtt{E}$ obtained using the SymStereo framework [1], [2]. In order to detect and locate planar surfaces in the scene, we take advantage of the prior that two planes always intersect into a line, meaning that the intersection of the virtual plane with a scene plane is always a *line cut*. We formulate the multi-model plane fitting as an optimization problem using the PEARL algorithm [3] (Section III). The initial plane labels are proposed by first extracting line segments from each energy $\mathtt{E}$ using the weighted Hough transform, and then each set of two lines provides a plane hypothesis for the initial label set. The objective in the expand step is to assign to each back-projection ray of the cyclopean eye, that is the intersection of a virtual cut plane and an epipolar plane, a plane label of the initial plane set. This is accomplished using an energy regularization framework and α-expansion optimization. In the third step, we re-estimate the parameters of each plane model with non-empty set of inliers by minimizing over the energies $\mathtt{E}$ via Levenberg-Marquardt [18]. All labels with no inliers are discarded, and the new plane set is used in the expand step. We iterate between inlier classification and re-estimation steps until a new α-expansion does not decrease the global energy. The results (Section IV) show that our approach successfully recovers accurate semi-dense piecewise planar descriptions of challenging scenes containing multiple planar and non-planar surfaces.

II. SymStereo

The SymStereo framework was presented in [1], [2] and relates with plane-sweep stereo [19] in the sense that it also samples the 3D space by a family of virtual planes and back-projects the input images onto these planes for searching corresponding image locations. However, SymStereo exclusively considers virtual planes that intersect the baseline in a point between the cameras. In this case, corresponding image locations are not photo-consistent, but instead are reflected one with respect to the other around the curve where the virtual plane intersects the scene structure (the profile cut). SymStereo explores this mirroring effect for reconstructing the profile cut using symmetry analysis.

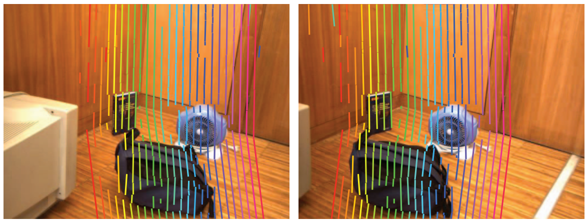
A. *Processing each virtual cut plane*

We show in Figure 2 the processing of two virtual cut planes, where the final detection results of our algorithm are overlaid in the input images in Figure 1. Let $\widehat{\mathtt{I}}$ be the result of warping the right image $\mathtt{I}'$ by the plane-homography associated with a virtual cut plane. Since the virtual plane intersects the baseline, then $\mathtt{I}$ and $\widehat{\mathtt{I}}$ are reflected around the image of the profile cut. 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 the symmetry and anti-symmetry of $\mathtt{I}^S$ and $\mathtt{I}^A$ at every image pixel location. This provides an implicit manner of recovering depth along a virtual cut plane and achieving data association across views.

The quantification of the signal symmetry and anti-symmetry along the epipolar lines in $\mathtt{I}^S$ and $\mathtt{I}^A$, respectively, is accomplished by analyzing the local frequency information using a bank of log-Gabor wavelets [20]. This allows to obtain the symmetry energies $\mathtt{E}_S$ and $\mathtt{E}_A$ from $\mathtt{I}_S$ and $\mathtt{I}_A$, respectively. In order to highlight the pixel locations that are both symmetric and anti-simetric in $\mathtt{I}_S$ and $\mathtt{I}^A$, respectively, the joint energy $\mathtt{E}$ is computed by pixel-wise multiplication of $\mathtt{E}_S$ and $\mathtt{E}_A$. As can be observed in Figure 2, this operation enables to discard many spurious local maxima that arise in both the symmetry and anti-symmetry energies.

III. Piecewise planar reconstruction using SymStereo and PEARL

This section describes the algorithm that combines the SymStereo framework with the geometric multi-model fitting algorithm PEARL [3] for semi-dense piecewise planar reconstruction of the scene. The PEARL algorithm consists in three main steps: (i) *propose* an initial set of plausible models (labels) from the observations, (ii) *expand* the label set for estimating its spatial support (inlier classification), and (iii) *re-estimate* the inlier models by minimizing some error function. We assume that we have N energies $\mathtt{E}_q$ obtained using SymStereo from a set of N virtual cut planes $\mathbf{\Pi}_q$ that belong to a vertical pencil with the axis intersecting the baseline in its middle point.



(a) Left View $\mathtt{I}$ (b) Right View $\mathtt{I}'$

Figure 3. Detection of line cuts from 25 virtual cut planes. Line cuts belonging to the same virtual plane have the same color. The independent detection for each virtual plane of the image of the profile cut contains holes, poor estimations and mis-detections.

A. Initial plane set for PEARL

Following an approach similar to [1], each energy $\mathtt{E}_q$ is used as input to a weighted Hough transform for extracting line cuts (Figure 3). Each 3D line segment is a possible location of intersection of a virtual cut plane with a planar surface contained in the scene. In order to propose an initial set of plane models $\mathcal{P}$ for PEARL, we generate all possible planes that can be obtained from two 3D lines. Remark that only line cuts from different virtual planes generate a valid plane hypothesis for $\mathcal{P}$.

B. Graph-Cut labeling and plane refinement

Given the initial set of plane hypotheses $\mathcal{P}$, the objective is to expand the models and estimate their spatial support. Let $\mathbf{\Pi}_q$, with $q = 1, 2, ..., N$, be the set of vertical virtual cut planes that meet the baseline in its midpoint, and $\mathbf{\Psi}_r$, with $r = 1, 2, ..., M$, the set of epipolar planes (one epipolar plane per image row). Consider also line $\mathbf{b}_{q,r}$ where $\mathbf{\Pi}_q$ intersects $\mathbf{\Psi}_r$, which can be understood as a back-projection ray of the cyclopean eye. The objective of our approach is to estimate the point on $\mathbf{b}_{q,r}$ that most likely belongs to a planar surface. This problem can be cast as a labeling problem, in which the nodes of the graph are the back-projection rays $\mathbf{b}_{q,r} \in \mathcal{B}$, and to each $\mathbf{b}_{q,r}$ we want to assign a plane label $f_{\mathbf{b}_{q,r}}$. The set of possible labels is $\mathcal{F} = \{\mathcal{P}, non-planar\}$, with $non-planar$ meaning that no point on $\mathbf{b}_{q,r}$ belongs to a plane surface. Please note that we use $\mathbf{b}$ instead of $\mathbf{b}_{q,r}$ whenever the virtual and epipolar plane specification is not strictly necessary. The objective energy E is defined as

$$E(\mathbf{f}) = \underbrace{\sum_{\mathbf{b}\in\mathcal{B}} D_{\mathbf{b}}(f_{\mathbf{b}})}_{\text{data term}} + \lambda \underbrace{\sum_{\mathbf{b},\mathbf{c}\in\mathcal{N}} V_{\mathbf{b},\mathbf{c}}(f_{\mathbf{b}}, f_{\mathbf{c}})}_{\text{smoothness term}} + \underbrace{\beta \cdot |\mathcal{F}_f|}_{\text{label term}}, \qquad (1)$$

where $\mathbf{f}$ is a particular labeling being considered, the neighborhood $\mathcal{N}$ for $\mathbf{b}_{q,r}$ is defined by the four back-projection rays $\mathbf{b}_{q-1,r}$, $\mathbf{b}_{q+1,r}$, $\mathbf{b}_{q,r-1}$ and $\mathbf{b}_{q,r+1}$ (see Figure 4), and $V_{\mathbf{b},\mathbf{c}}$ is the spatial smoothness term that encourages piecewise smooth labeling by penalizing configurations $\mathbf{f}$ that assign to neighboring nodes $\mathbf{b}$ and $\mathbf{c}$ different labels. The label term is used for describing the 3D scene using as few unique plane surfaces as possible, with $\mathcal{F}_f$ being the subset of different plane models assigned to the nodes $\mathbf{b}$ by the labeling $\mathbf{f}$ (see [3] for further details). The data and smoothness terms are further described below.

1) Data term: The data term $D_{\mathbf{b}_{q,r}}$ for the back-projection ray $\mathbf{b}_{q,r}$ is defined as

$$D_{\mathbf{b}_{q,r}}(f) = \begin{cases} \min(1 - \mathtt{E}_q(r, x_f), \tau) & \text{if } f \in \mathcal{P} \\ \tau & \text{if } f = non-planar \end{cases} \qquad (2)$$

where $\mathtt{E}_q$ is the energy associated with $\mathbf{\Pi}_q$, r is the row corresponding to $\mathbf{\Psi}_r$, x_f is the column defined by the plane hypothesis f (intersection of $\mathbf{b}_{q,r}$ with the plane indexed by f), and τ is a constant. Note that similarly to [8], the $non-planar$ label indicates that no satisfactory plane hypothesis can be assigned to $\mathbf{b}_{q,r}$. In this case, the back-projection ray $\mathbf{b}_{q,r}$ has high probability of not intersecting the scene in a planar surface.

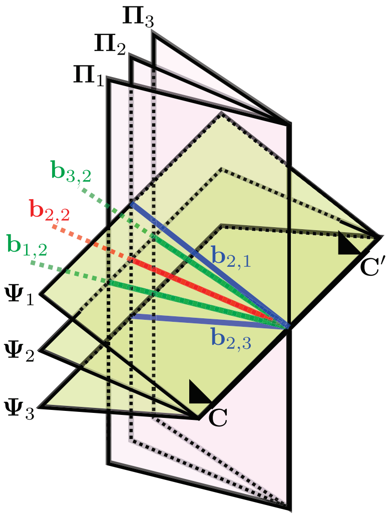
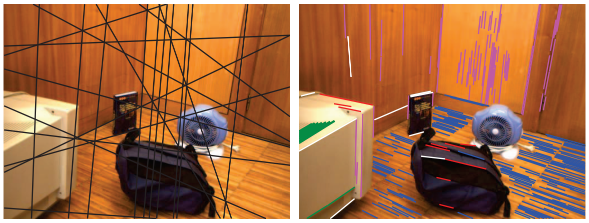


Figure 4. We use the PEARL algorithm for assigning to each back-projection ray $\mathbf{b}_{q,r}$, corresponding to the intersection between a virtual cut plane $\mathbf{\Pi}_q$ and an epipolar plane $\mathbf{\Psi}_r$, a plane surface label f_b. $\mathbf{C}$ and $\mathbf{C}'$ represent, respectively, the left and the right cameras. The neighborhood $\mathcal{N}$ of the red line $\mathbf{b}_{2,2}$ is constituted by the green lines $\mathbf{b}_{1,2}$ and $\mathbf{b}_{3,2}$ that are on the same epipolar plane, but belong to neighboring virtual planes; and by the blue lines $\mathbf{b}_{2,1}$ and $\mathbf{b}_{2,3}$, which are on the same virtual plane, but are intersected by neighboring epipolar planes.



(a) Crease Edges (b) Detected Line Segments

Figure 5. Crease edges arise from the intersection of planar surfaces in the scene. We show in (a) the crease edges obtained from our piecewise planar 3D reconstruction. The location of occlusion edges is usually coincident with visible 2D line segments. We show in (b) the result of the clustering of concurrent lines, such that each group of lines provides a possible vanishing point location. The white line segments did not received any vanishing point label.

2) Smoothness term: Inspired by [7], the smoothness term of neighboring nodes $\mathbf{b}$ and $\mathbf{c}$ is given by

$$V_{\mathbf{bc}}(f_{\mathbf{b}}, f_{\mathbf{c}}) = \begin{cases} 0 & \text{if } f_{\mathbf{b}} = f_{\mathbf{c}} \\ \lambda_1 & \text{if } (f_{\mathbf{b}} \vee f_{\mathbf{c}}) = non{-}planar \\ \lambda_2 & \text{if } (\mathbf{b}, \mathbf{c}, f_{\mathbf{b}}, f_{\mathbf{c}}) \in \mathsf{S}_1 \\ \lambda_3 & \text{if } (\mathbf{b}, \mathbf{c}, f_{\mathbf{b}}, f_{\mathbf{c}}) \in \mathsf{S}_2 \\ \lambda_4 & \text{if } (\mathbf{b}, \mathbf{c}) \in \mathsf{S}_3 \\ 1 & \text{else} \end{cases} \quad (3)$$

where $0 < \lambda_2 < \lambda_3 < \lambda_4 < 1$, and the content of the sets S_1, S_2 and S_3 is shortly described. No penalization is assigned to neighboring nodes receiving the same plane label, while in the case of one node obtaining the $non{-}planar$ label, a non-zero cost is added to the plane configuration $\mathbf{f}$.

Let $p_{\mathbf{b},f_b}$ be the projection onto the left view $\mathtt{I}$ of the point on the ray $\mathbf{b}$ intersected by the plane surface associated to f_b. Following the same reasoning as in [7], the assignment of different plane labels f_b and f_c to neighboring nodes $\mathbf{b}$ and $\mathbf{c}$ is less penalized if $p_{\mathbf{b},f_b}$ and $p_{\mathbf{c},f_c}$ are separated by either a crease or an occlusion edge. A crease edge (Figure 5(a)) corresponds to the projection of the 3D line of intersection between a pair of planes onto $\mathtt{I}$. For this purpose, we store in the set S_1 the quadruples $(\mathbf{b}, \mathbf{c}, f_{\mathbf{b}}, f_{\mathbf{c}})$ whose points $p_{\mathbf{b},f_b}$ and $p_{\mathbf{c},f_c}$ are located on different sides of the crease edge defined by f_b and f_c.

Occlusion edges often coincide with visible 2D line segments in the input images (Figure 5(b)). In order to find possible occlusion edges, we detect 2D line segments $\mathbf{s}$ in $\mathtt{I}$ using the Line Segment Detector [21]. Then, single-view multi-vanishing point fitting is performed for clustering concurrent lines. This is achieved using a simple PEARL fitting, whose data term is given by

$$\sum_{\mathbf{s}} d(\mathbf{s}_s, \mathbf{v} \times \mathbf{s}_m) + d(\mathbf{s}_e, \mathbf{v} \times \mathbf{s}_m), \quad (4)$$

where $\times$ indicates the vector cross product, and $d(\mathbf{p}, \mathbf{l})$ represents the euclidean distance between point $\mathbf{p}$ and line $\mathbf{l}$; $\mathbf{s}_s$, $\mathbf{s}_m$ and $\mathbf{s}_e$ are the start, middle and end points of the line segment $\mathbf{s}$, respectively, and $\mathbf{v}$ is a vanishing point. A constant label cost is used for explaining the line segments using as few unique vanishing point labels as possible (see Figure 5(b)), and no smoothness penalization is applied. We store in the set S_2 the quadruples $(\mathbf{b}, \mathbf{c}, f_{\mathbf{b}}, f_{\mathbf{c}})$, with points $p_{\mathbf{b},f_b}$ and $p_{\mathbf{c},f_c}$ being located on different sides of a line segment $\mathbf{s}$ clustered to the vanishing point $\mathbf{v}$, whose vanishing direction is orthogonal either to the plane associated to f_b or f_c. Finally, the set S_3 contains the remaining pairs $(\mathbf{b}, \mathbf{c})$ whose points $p_{\mathbf{b},f_b}$ and $p_{\mathbf{c},f_c}$ in $\mathtt{I}$ are on different sides of a line segment $\mathbf{s}$ to which no vanishing point was assigned (white line segments in Figure 5(b)). Remark that different from [7], we do not perform any line matching between the stereo views.

3) Plane refinement: The third step of the PEARL algorithm is to re-estimate the model parameters using the inliers of the discrete labeling. Let $\mathbf{\Omega}_\alpha$ be the plane associated with the label f_α, that has been assigned to a non-empty set of inliers $\mathbf{I}(f_\alpha) = \{\mathbf{b} \in \mathcal{B} | f_{\mathbf{b}} = f_\alpha\}$. Each plane $\mathbf{\Omega}_\alpha$ is refined by minimizing its plane parameters over the energies $\mathtt{E}$ via Levenberg-Marquardt [18]:

$$\mathbf{\Omega}_{LM} = \min_{\mathbf{\Omega}} \sum_{\mathbf{b}_{q,r} \in \mathbf{I}} \left(1 - \mathtt{E}_q(r, x_{\mathbf{\Omega}})\right), \quad (5)$$

where $x_{\mathbf{\Omega}}$ is the column defined by the intersection of $\mathbf{b}_{q,r}$ with $\mathbf{\Omega}$. The new set of labels $\mathcal{P} = \{\mathbf{\Omega}_{LM}\}$ 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 1.

IV. Results

We performed experiments of our semi-dense piecewise planar approach on various challenging indoor and outdoor scenes (see Figure 6) acquired using a Bumblebee stereo camera from PointGrey, with a baseline of 24 cm and image resolution of 1024×768 pixels. The scenes contain planar and non-planar surfaces, as well as a variety of complicated situations to traditional stereo methods e.g. low and/or repetitive textures, high surface slant, specular structures. As discussed previously, we apply the SymStereo framework using a vertical pencil of $N = 25$ virtual cut planes with the axis intersecting the baseline in its middle point. The average runtime using a straightforward Matlab implementation takes about 2 minutes for the processing of a stereo pair. Most of the time is spent computing the symmetry energies $\mathtt{E}$, which can be parallelized in the GPU in further versions of the algorithm.

For all the experiments, we used the same parameters $\{\lambda, \beta, \lambda_1, \lambda_2, \lambda_3, \lambda_4\}$=$\{0.04, 10, 0.5, 0.5, 0.55, 0.75\}$, which were empirically selected without much effort. Concerning the parameter τ, we tested two different values, namely 0.8 and 0.6. The parameter τ works as a control for the rigor of the plane selection. Higher values means more permissive with respect to what is considered a planar surface, while for lower values the algorithm only outputs strict planes.

For all the stereo pairs of Figure 6 we show some crease edges that can be used as indicators of the accuracy of the plane estimation. Concerning $\tau = 0.8$, the first two scenes (rows) are only composed by planar surfaces that are accuractelly reconstructed, as proven by the crease edges. We added to the scenes of rows 3 and 4 non-planar objects, which are well approximated by one or more plane surfaces. Additionally, the scene of row 4 contains, besides large planar and non-planar surfaces, a small plane corresponding to the blue book, which is well estimated from only 4 virtual cut planes (please see the crease edge between the book and the floor). The last row of the office dataset contains and example of poor segmentation of the chair back (red plane). This is due to two reasons: (i) in order to overcome low-textured regions, we use a significant number of log-Gabor wavelets with large spatial support. In situations in which we intersect the scene close to depth discontinuities, the larger log-Gabor kernels favor the smaller object depth, and (ii) on the left of the chair back we have few virtual cut planes intersecting the wall, so that the smoothness term in this region prefers the red label.

In the first example of the stairs data set we detect 9 planar surfaces, from which 6 planes correspond to the steps, the red plane is the floor, and 2 planes refer to the right wall and the handrail. Remark that the reconstruction of the floor and steps is almost perfect. In the second example, the top steps are approximated by a single plane (red). This occurs because the image resolution is not sufficient for discriminating depth at such large distances, and SymStereo is not able to accurately guide the matching process. Additionally, we are able to detect a plane on the right of the stairs which apparently corresponds to the white wall (magenta). However and since the estimation is deceived by the handrail, the plane model is incorrect as proven by the crease edges.

The Lift stereo pair shows that our algorithm is able to handle specular surfaces (lift door and floor). We accurately estimate the 3 large planar surfaces, and our approach also attempts to estimate a very thin specular plane with almost no texture corresponding to the corner of the lift (blue).

Finally, in the outdoor dataset we show non-trivial 3D scenarios containing high slant and very low texture. In the first example, the green and blue planes are parallel (floor and roofing), while the red plane refers to the white wall being orthogonal. The three lines were the planes intersect (two visible crease edges and the vanishing line of the horizontal planes) perfectly intersect in a vanishing point, which proves high accuracy. In the top right corner, two small line cuts are correctly clustered as being in the vertical plane. There is one plane that is not detected, namely the thin vertical plane of the roofing. This occurs mainly due to lack of texture. In the last example, we accurately estimate the vertical blue plane of the roofing, as well as the small magenta plane near the window. Our approach is able to detect and distinguish the window, however the plane estimation is misled by strong specularities.

So far, most of the nodes $\mathbf{b}$ received a plane label even though belonging to non-planar objects (see rows 3 and 4). The control of labeling just *strict* planes can be achieved using the truncation parameter τ. We show on the two right columns of Figure 6 results on the same datasets, however decreasing τ from 0.8 to 0.6. In this case, the non-planar objects in rows 3 and 4, as well as the red plane in the second stairs example are not reconstructed, because the algorithm only outputs plane models with high probability of being *really* contained in the scene. However, this has the drawback of discarding planes, and erroneously labeling nodes $\mathbf{b}$ with the $non\text{--}planar$ label, in regions of low texture (e.g. white part of the walls in row 5) or containing specular reflections (e.g. lift door in row 8).

V. Conclusions

We presented an automatic algorithm for semi-dense piecewise planar reconstruction from just two views. Unlike other methods, the stereo depth estimation and the detection of planar surfaces are accomplished in a tight, coupled manner by combining SymStereo [1], [2] with PEARL clustering [3]. This enables to take full advantage of the strong planarity prior, with the algorithm being able to output very accurate 3D reconstructions of the line segments where the virtual cut planes meet the planes in the scene. No assumptions are made about the number or the pose of the 3D plane surfaces. The effectiveness of the scheme is proved by several experiments in challenging indoor and outdoor scenarios. It is also shown that, depending on the tuning of the parameter τ, it is possible to either obtain strict 3D plane detections or approximate the scene surfaces by a suitable piecewise-planar model. It is important to note that a vertical pencil of virtual planes was employed in SymStereo, so that using few planes can lead to entirely missing thin objects. This problem can be solved by applying oblique virtual planes for a better coverage of the 3D scene.

As a final comment, it can be claimed that the energy-based model fitting can either be applied to 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 provides a nice trade-off between the two, playing a key role in the success of the overall approach.

VI. Acknowledgments

Michel Antunes acknowledges the Portuguese Science Foundation (FCT) that generously funded his work through the grant SFRH/BD/47488/2008. This work has also been partially supported by Ministério da Ciência, Tecnologia e Ensino Superior (MCTES), under FCT project PDCS10:PTDC/EEA-AUT/113818/2009.

References

[1] M. Antunes and J. P. Barreto, “Plane surface detection and reconstruction using induced stereo symmetry,” in *BMVC*, 2011.

[2] ——, “Stereo estimation of depth along virtual cut planes,” in *ICCVW (CVVT)*, 2011.

[3] H. Isack and Y. Boykov, “Energy-based geometric multi-model fitting,” *IJCV*, 2012.

[4] T. Werner and A. Zisserman, “New techniques for automated architectural reconstruction from photographs,” in *ECCV*, 2002.

[5] 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.

[6] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski, “Manhattan-world stereo,” in *CVPR*, 2009.

[7] S. Sinha, D. Steedly, and R. Szeliski, “Piecewise planar stereo for image-based rendering,” in *ICCV*, 2009.

[8] D. Gallup, J.-M. Frahm, and M. Pollefeys, “Piecewise planar and non-planar stereo for urban scene reconstruction,” *CVPR*, 2010.

[9] 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.

[10] M. Bleyer, C. Rhemann, and C. Rother, “Patchmatch stereo - stereo matching with slanted support windows,” in *BMVC*, 2011.

[11] A. Bartoli, “A random sampling strategy for piecewise planar scene segmentation,” *CVIU*, 2007.

[12] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” *PAMI*, 2010.

[13] A. Klaus, M. Sormann, and K. Karner, “Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure,” in *ICPR*, 2006.

[14] M. H. Lin and C. Tomasi, “Surfaces with occlusions from layered stereo,” *PAMI*, 2004.

[15] M. Bleyer and M. Gelautz, “A layered stereo algorithm using image segmentation and global visibility constraints,” in *ICIP*, 2004.

[16] M. Bleyer, C. Rother, and P. Kohli, “Surface stereo with soft segmentation,” in *CVPR*, 2010.

[17] S. Birchfield and C. Tomasi, “Multiway cut for stereo and motion with slanted surfaces,” *ICCV*, 1999.

[18] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[19] R. T. Collins, “A space-sweep approach to true multi-image matching,” in *CVPR*, 1996.

[20] P. Kovesi, “Symmetry and asymmetry from local phase,” in *Australian Joint Conf. on Artificial Intelligence*, 1997.

[21] 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.

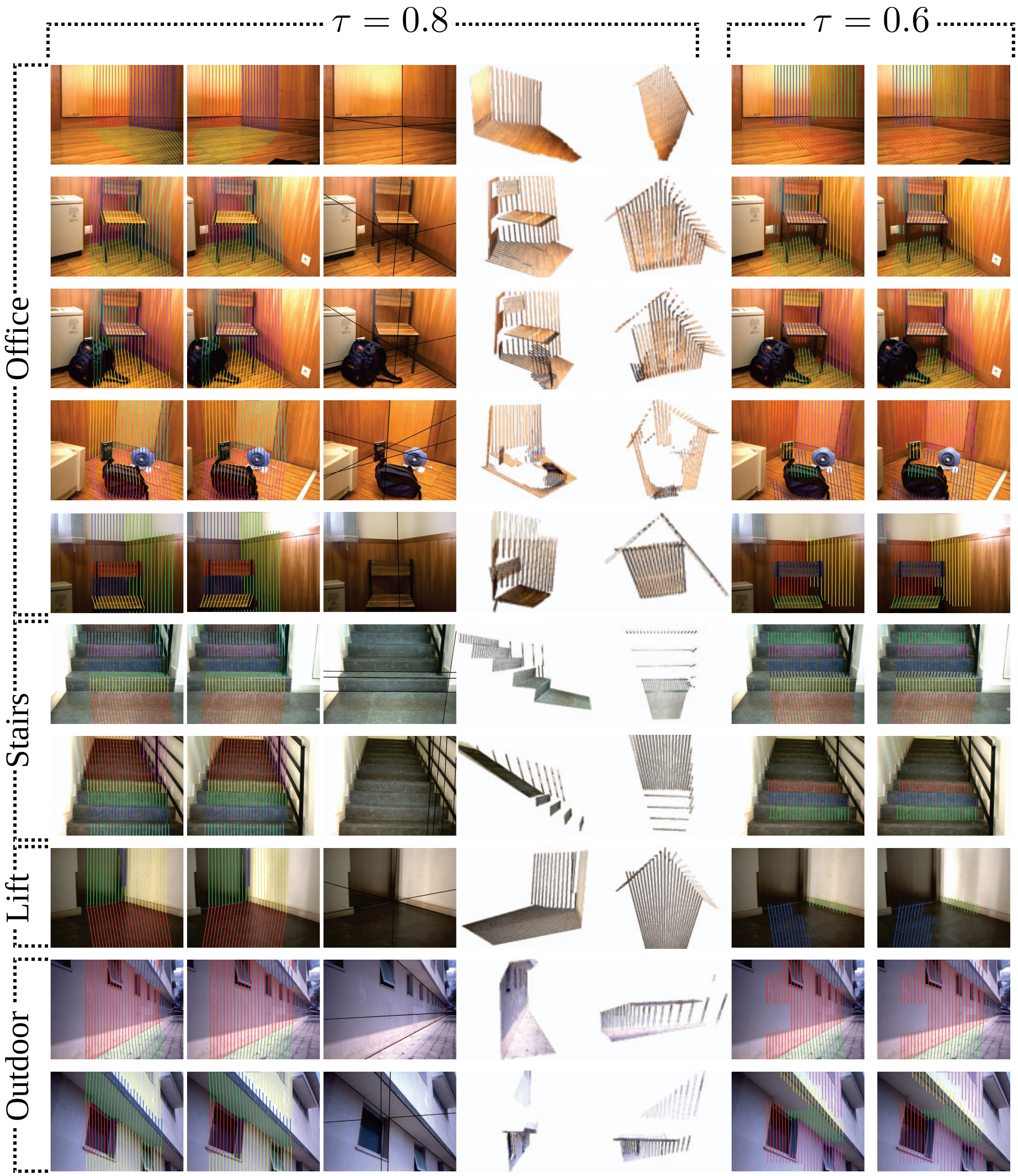


Figure 6. Results produced by our semi-dense piecewise planar algorithm (10 stereo pairs). From left to right: the left and right views with the images of the profile cuts overlaid (compare the matching between the views), different colors indicate different planes; the left view with crease edges (black) that can be used as indicators of the estimation accuracy; and two views of the textured 3D reconstruction rendered from different viewpoints. For the results on the right (two columns), we decreased the parameter τ.