Stereo estimation of depth along virtual cut planes

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

Abstract

Stereo vision is broadly employed in robotics and intelligent vehicles for recovering the 3D structure of the environment. The scene depth is typically estimated by triangulation after associating pixels between views using a dense stereo matching approach. In the last few years, the image resolution has steadily increased due to the advances in camera technology. Unfortunately, achieving real-time stereo using large size images is difficult because of the computational cost of dense matching. An obvious solution is to re-sample the acquired input images, but this implies decreasing the accuracy of depth estimates. We propose an alternative that consists in performing the stereo reconstruction of the contour $\mathcal{C}$ where a pre-defined virtual cut plane intersects the scene. This approach enables a trade-off between runtime and 3D model resolution that does not interfere with depth accuracy. The profile cuts $\mathcal{C}$ are independently recovered using the SymStereo *framework that has been recently introduced in [1]. It is proved through comparative experiments that SymStereo is particularly well suited for recovering depth along virtual cut planes, outperforming state-of-the-art stereo cost functions both in terms of accuracy and runtime.*

1. Introduction

Stereo reconstruction consists in recovering the 3D structure of a scene by associating pixels in two calibrated images acquired from different viewpoints. The stereo approaches can be coarsely divided into two groups: *sparse stereo*, that sparsely extracts features from the images and then searches for corresponding locations in the other views [13]; and *dense stereo*, that performs dense data association between images by assigning to each pixel a disparity value [11]. The matching process typically uses some type of similarity measure to determine how likely pixels in different views correspond [7].

Many autonomous systems employ stereo vision for reconstructing the surrounding environment in order to accomplish navigation and detection tasks [6]. The chosen stereo approach must run in real-time and provide depth estimates that are accurate enough to build an useful 3D map. The accuracy of the depth estimates is mainly limited by the computational cost of the matching, with the acquired input images being often re-sampled to keep execution tractable. As shown in Fig. 1, the runtime of dense stereo can be reduced by increasing the disparity sampling intervals, i.e. by selecting integer disparities to be larger than unitary pixel shifts. Unfortunately, this means decreasing the metric accuracy of the final depth estimates, which is prejudicial for many applications. Thus, and in order perform stereo over high resolution images in real-time, we propose to sample the 3D space by N virtual planes and detect the images of the contours $\mathcal{C}$ where each cut plane intersects the scene structure (see Fig. 2(b)). The reconstruction of the N profile cuts $\mathcal{C}$ gives raise to a sparse 3D model of the scene with the largest possible depth resolution (unitary pixel steps). The runtime can be reduced by decreasing the number of virtual cut planes, leading to a coarser scene model but with the

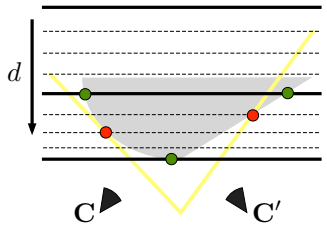
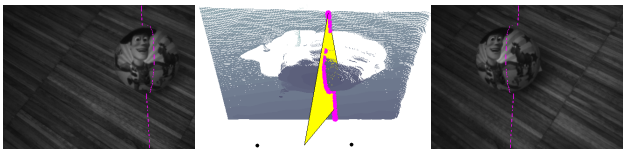


Figure 1. Estimation of scene depth from rectified stereo. The dashed lines are the depths corresponding to successive disparity values (unit steps). The runtime can be decreased by considering disparity steps that are greater than one (solid lines). Unfortunately, in this case the red pixels will receive an inaccurate depth label. A possible alternative is to sample the 3D space by a set of virtual cut planes intersecting the baseline (yellow), and reconstruct the curves where they meet the structure. This leads to a 3D model that is coarser (not all image pixels are reconstructed), but with the maximum possible depth resolution along the cut planes.

same depth resolution.



(a) Image $\mathtt{I}$ (b) 3D model (c) Image $\mathtt{I}'$

Figure 2. The figure shows the stereo pair, $\mathtt{I}$ and $\mathtt{I}'$, and a 3D model of the scene structure. In (b) the *virtual cut plane* Π (yellow) passes between the cameras, and intersects the structure in a non-continuous 3D curve $\mathcal{C}$ (magenta). The SymStereo approach is used for determining the image of the *profile cut* $\mathcal{C}$ that is overlaid in magenta in (a) and (c).

Let's consider a particular virtual plane $\mathbf{\Pi}$ and the corresponding profile cut $\mathcal{C}$. The profile cut can be easily reconstructed by finding the contours where $\mathcal{C}$ is projected in the two views (see Fig. 2). Remark that, since $\mathbf{\Pi}$ induces an homography relation, then each image pixel has a unique disparity hypothesis. Thus, the image of $\mathcal{C}$ can be determined by looking for extrema in a stereo cost function defined across the possible disparities.

Instead of relying in a standard matching cost, we apply the SymStereo framework [1] for detecting the image of the profile cut $\mathcal{C}$ using exclusively symmetry analysis. This is possible because, since the virtual plane $\mathbf{\Pi}$ is assumed to pass in between the cameras, the homography mapping of one view into the other gives raise to a warped image that is mirrored with respect to the locus where $\mathcal{C}$ is projected. As discussed in [1], the symmetry cue has a global character that is advantageous in handling low textured regions where photo-consistency based metrics are often non-discriminative. According to experiments using the Middlebury dataset, SymStereo is significantly better for estimating depth along independent cut planes than state-of-the-art stereo cost functions [7].

It is important to refer that SymStereo was first introduced in [1] for reconstructing the "line cuts" where virtual planes meet planar surfaces in the scene. In here, the SymStereo framework is used for recovering arbitrary profile cuts. Moreover, we provide a formal proof of the mirroring effect, and discuss how the virtual cut planes relate with the disparity space image (DSI) [14]. Finally, we present new experiments using the Middlebury dataset, that show for the first time that symmetry outperforms photo-consistency for the purpose of sparse stereo reconstruction.

1.1. Structure and Notation

The paper is organized as follows: Section 2 explains the rendering of symmetric images, provides a formal proof for the mirroring effect, and discusses how the virtual cut planes are mapped into the DSI [14]. Section 3 introduces the log-Gabor wavelets used for quantifying signal symmetry [8], and demonstrates how the detection of more than one profile cut can be implemented efficiently. Section 4 applies the SymStereo framework in the reconstruction of sparse profile cuts of the scene. The results are compared against state-of-the-art matching costs.

We denote scalars by italics, e.g. s, vectors by bold characters, e.g. $\mathbf{p}$, $\mathbf{P}$, matrices in sans serif font, e.g. M, image signals in typewriter font, e.g. $\mathtt{I}$, and curves by calligraphic symbols, e.g. $\mathcal{C}$. Unless otherwise stated, we use homogeneous coordinates for points and other geometric entities, e.g. an image point with non-homogeneous coordinates (p_1, p_2) is represented by $\mathbf{p} \sim \begin{pmatrix} p_1 & p_2 & 1 \end{pmatrix}^\mathsf{T}$, with $\sim$ denoting equality up to a scalar factor. $[\mathbf{v}]_\times$ refers to the skew symmetric matrix defined by the 3-vector $\mathbf{v}$, and $\mathsf{I}_{3\times3}$ is the 3×3 identity matrix.

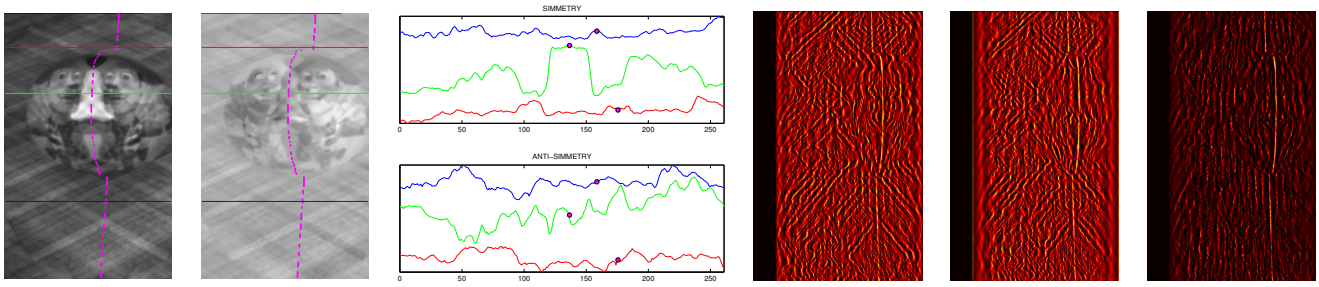
2. Stereo From Induced Symmetry

The plane sweeping algorithm was first introduced by Collins [3] for finding matches across multiple images without the need of rectification. It has been widely used in dense depth estimation due to its simplicity and computational efficiency. The basic idea consists in sampling the 3D space by a family of parallel virtual planes, back-project the images onto these planes, and find the locations where the back-projections are most similar. Ideally, these locations correspond to the intersection points of the plane with the imaged surfaces, which enables depth recovery. Stereo matching over rectified stereo [11] can be understood as a particular instance of plane-sweeping, with the virtual planes being fronto-parallel to the cameras, and each plane corresponding to a disparity hypothesis. The SymStereo approach for stereo reconstruction relates with plane-sweeping in the sense that it also samples the 3D space by virtual planes. However, there are two major differences: (i) exclusively virtual planes that intersect the baseline in a point between the cameras are considered; (ii) the pixel association, instead of being performed using direct photo-similarity, is implicitly achieved based on symmetry cues.

2.1. The Mirroring Effect

Works using plane sweeping consider virtual planes passing between the cameras a degenerate configuration to be avoided [5]. The reason is as follows: Let $\mathtt{I}$ be the left stereo image and $\widehat{\mathtt{I}}$ be the result of warping the right image $\mathtt{I}'$ by the plane homography of the virtual cut plane (see Fig.2). Since the virtual plane crosses the baseline, $\mathtt{I}$ and $\widehat{\mathtt{I}}$ are mirrored one with respect to the other around a contour $\mathcal{C}$, which corresponds to the projection of the 3D curve where the plane cuts the scene (the *profile cut*).

Following this, and as shown in Fig.3, the sum of $\mathtt{I}$ and $\widehat{\mathtt{I}}$ yields an image signal $\mathtt{I}_s$ that is locally symmetric around the cut contour $\mathcal{C}$. In a similar manner, the subtraction of $\widehat{\mathtt{I}}$



(a) $\mathtt{I}_s = \mathtt{I} + \widehat{\mathtt{I}}$ (b) $\mathtt{I}_a = \mathtt{I} - \widehat{\mathtt{I}}$ (c) 1-D Signal (Epipolar Line) (d) $\mathtt{E}_s$ (e) $\mathtt{E}_a$ (f) $\mathtt{E} = \mathtt{E}_s \cdot \mathtt{E}_a$

Figure 3. $\mathtt{I}_s$ and $\mathtt{I}_a$ are symmetric and anti-symmetric image signals generated from the stereo pair of Fig. 2 (just the overlapping part between the two views is shown). They are rendered by adding and subtracting $\mathtt{I}$ with $\widehat{\mathtt{I}}$, that is the result of warping $\mathtt{I}'$ by the homography induced by $\mathbf{\Pi}$. Fig. 3(c) shows the intensity level of $\mathtt{I}_s$ and $\mathtt{I}_a$ for three distinct epipolar lines (blue, green and red). The intersections with the contour $\mathcal{C}$ can be identified with almost no ambiguity by searching the pixel locations for which the top and bottom 1D-signals are respectively symmetric and anti-symmetric. The convolution of $\mathtt{I}_s$ and $\mathtt{I}_a$ with the log-Gabor wavelets yields $\mathtt{E}_s$ and $\mathtt{E}_a$. The final energy $\mathtt{E}$ is computed by pixel-wise multiplication of $\mathtt{E}_s$ and $\mathtt{E}_a$ to highlight pixel locations for which both symmetry and anti-symmetry arise.

from $\mathtt{I}$ gives raise to an image signal that is anti-symmetric at the exact same location. Thus, we propose to detect $\mathcal{C}$, and implicitly recover the depth information, by searching for common pixel locations where $\mathtt{I}_s$ and $\mathtt{I}_a$ are respectively symmetric and anti-symmetric. As shown in Fig.3, this seems to be a highly discriminative cue for stereo.

2.2. Formal geometric proof

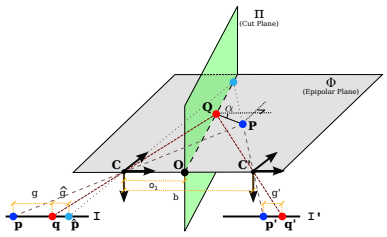


Figure 4. Geometric derivation of SymStereo. We assume rectified stereo with the entire analysis being carried in an arbitrary epipolar plane. The camera centers $\mathbf{C}$ and $\mathbf{C}'$ are separated by a distance $b > 0$ (the baseline length), and the world coordinate system is assumed to be coincident with the reference frame of the left view. For the sake of graphical clarity the points are projected behind the optical centers.

This section derives and analyzes the proposed symmetry cue for the case of rectified stereo [4]. As shown in Fig. 4, the reference frame centers $\mathbf{C}$ and $\mathbf{C}'$ of the two cameras are aligned. Thus, the relative camera rotation is $\mathsf{R} = \mathsf{I}_{3\times 3}$ and the translation is

$$\mathbf{t} = \begin{pmatrix} b & 0 & 0 \end{pmatrix}^{\mathsf{T}}.$$

Consider that the virtual cut plane is represented by the homogeneous vector

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

The plane $\mathbf{\Pi}$ intersects the 3D line going through $\mathbf{C}$ and $\mathbf{C}'$ that can be parametrized using the so-called Plücker coordinates [9]. Knowing that the line direction and momentum are respectively $\mathbf{u} = \mathbf{t}$ and $\mathbf{v} = \mathbf{0}$ (the line goes through the origin), it comes that the intersection point $\mathbf{O}$ is

$$\mathbf{O} \sim \begin{pmatrix} -[\mathbf{0}]_\times & \mathbf{t} \\ -\mathbf{t}^{\mathsf{T}} & 0 \end{pmatrix} \mathbf{\Pi} \sim \begin{pmatrix} \frac{h}{\mathbf{n}_1} & 0 & 0 & 1 \end{pmatrix}^{\mathsf{T}}.$$

Let β be the ratio of the distances $\mathbf{CO}$ and $\mathbf{CC}'$. The cut plane $\mathbf{\Pi}$ passes between the cameras *iff* the following condition holds

$$0 < \left(\beta = \frac{O_1}{b}\right) < 1 \iff \frac{b\, n_1}{h} > 1. \qquad (1)$$

A generic point $\mathbf{P}$ is projected onto the stereo images in points $\mathbf{p}$ and $\mathbf{p}'$. Since we are assuming rectified stereo, then the non-homogeneous coordinates p_2 and p'_2 must have the same value y. Moreover, it can be proved [4] that

$$\mathbf{P} \sim \begin{pmatrix} \frac{b}{d_p} p_1 & \frac{b}{d_p} p_2 & \frac{b}{d_p} & 1 \end{pmatrix}^{\mathsf{T}}, \qquad (2)$$

where d_p denotes the disparity between left and right views

$$d_p = p_1 - p'_1. \qquad (3)$$

In addition to the generic point $\mathbf{P}$, consider the point $\mathbf{Q}$, that lies on the same epipolar plane and also belongs to $\mathbf{\Pi}$. Let $\mathbf{q}$ and $\mathbf{q}'$ be the projections of $\mathbf{Q}$ onto the stereo pair. The signed distances between the images of the two points are defined as

$$\begin{aligned} g &= p_1 - q_1 \\ g' &= p'_1 - q'_1 \end{aligned} \qquad (4)$$

In general, the order of corresponding points in the two views is the same and the distances g and g' have the same sign. However, there are singular stereo situations for which

the ordering constraint is not verified. In this case we are in the presence of a *double nail illusion*, that typically arises in scenes with thin foreground objects or narrow holes [12].

The virtual cut plane $\mathbf{\Pi}$ defines an homography H that can be used to map points $\mathbf{p}'$ of the right image into points $\widehat{\mathbf{p}}$ in the left image. Given the relative camera pose and the planar surface coordinates, it follows that [9]

$$\mathsf{H} \sim \left(\mathsf{I}_{3\times 3} + \frac{\mathbf{t}\,\mathbf{n}^{\mathsf{T}}}{h}\right)^{-1} \sim \begin{pmatrix} 1+\frac{bn_1}{h-bn_1} & \frac{bn_2}{h-bn_1} & \frac{bn_3}{h-bn_1} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{5}$$

with $\mathbf{n} = \begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix}^{\mathsf{T}}$. By making $\widehat{\mathbf{p}} \sim \mathsf{H}\,\mathbf{p}'$, we easily conclude that the first non-homogeneous coordinate of the map result is

$$\widehat{p}_1 \;=\; \left(1 + \frac{bn_1}{h - bn_1}\right) p'_1 \;+\; k_y,$$

where k_y is a constant for points sharing the same epipolar line y. From Eq.4 it yields that $p'_1 = g + q_1$. By replacing in the expression above, it comes that

$$\widehat{p}_1 \;=\; \underbrace{\left(1 + \frac{bn_1}{h - bn_1}\right) q'_1 \;+\; k_y}_{\widehat{q}_1 = q_1} + \left(1 + \frac{bn_1}{h - bn_1}\right) g'.$$

The homography H transforms $\mathbf{q}'$ into $\mathbf{q}$ because the 3D point $\mathbf{Q}$ lies in the cut plane $\mathbf{\Pi}$. Thus, the signed distance between the image point $\mathbf{q}$ and the mapped point $\widehat{\mathbf{p}}$ is

$$\widehat{g} = \widehat{p}_1 - q_1 = \left(1 - \frac{bn_1}{h}\right)^{-1} g' \tag{6}$$

For the case of the virtual plane $\mathbf{\Pi}$ going between the stereo views, the condition of Eq.1 holds, and the distances g' and $\widehat{g}$ have always opposite signs. Thus, whenever g and g' have the same sign, the points $\mathbf{p}$ and $\widehat{\mathbf{p}}$ are located on opposite sides of $\mathbf{q}$, which leads to the mirroring effect described previously. The only cases for which the homography map does not induce a reflection with respect to the cut contour $\mathcal{C}$ are the situations of *double nail illusion* [12]. This is a singularity of the SymStereo framework, that rarely happens and henceforth will be ignored.

2.3. Interpreting the cut planes in terms of the DSI

The DSI can be understood as a function from $\mathbb{R}^3$ into $\mathbb{R}$ that assigns to each pixel $(q_1,\, q_2)$ and possible disparity d_q a scalar matching cost that reflects the likelihood of the hypothesized disparity being correct [14]. Let us now discuss how a virtual cut plane $\mathbf{\Pi}$ is related to the DSI. The homography defined by a particular cut plane $\mathbf{\Pi}$ implicitly establishes a range of possible disparity values for each image pixel. It can be proved from Eq.5 that, for the images of a point $\mathbf{Q}$ lying on $\mathbf{\Pi}$, the following holds

$$q'_1 \;=\; (1 + \frac{bn_1}{h})\, q_1 + \frac{bn_2}{h}\, q_2 + \frac{bn_3}{h}\,.$$

Replacing q'_1 in the computation of the disparity d_q yields

$$d_q \;=\; \frac{bn_1}{h}\, q_1 + \frac{bn_2}{h}\, q_2 + \frac{bn_3}{h}$$

The equation above specifies a plane in the 3-dimensional space parametrized by $(q_1,\, q_2,\, d_q)$. Thus, each virtual cut plane $\mathbf{\Pi}$ gives raise to a planar surface $\mathbf{\Gamma}$ in the DSI domain, that has homogeneous representation

$$\mathbf{\Gamma} \sim \begin{pmatrix} \frac{bn_1}{h} & \frac{bn_2}{h} & -1 & \frac{bn_3}{h} \end{pmatrix}^{\mathsf{T}}.$$

In our experiments we will assume a pencil of virtual cut planes $\mathbf{\Pi}_\theta$ that intersect in a vertical axis going through the midpoint of the baseline (θ indicates the rotation around the axis). The corresponding plane surfaces in the DSI are parametrized by

$$\mathbf{\Gamma}_\lambda \sim \begin{pmatrix} 2 & 0 & -1 & -\lambda \end{pmatrix}^{\mathsf{T}},$$

with $\lambda = 2\tan(\theta)$. λ is chosen to be integer valued, so that for given a plane $\mathbf{\Gamma}_\lambda$, each pixel p in the left view will correspond to a particular integer disparity hypothesis d_p, and hence to a particular pixel p'.

3. Evaluating Signal Symmetry

This section shows how to use log-Gabor wavelets for the quantification of symmetry and anti-symmetry along image rows, and how this task can be implemented efficiently if more than one virtual cut plane are considered.

3.1. Symmetry analysis using log-Gabor wavelets

The localization of the contour $\mathcal{C}$ requires quantifying the symmetry and anti-symmetry of $\mathtt{I}_s$ and $\mathtt{I}_a$ along the epipolar lines. This is achieved using the approach proposed by Kovesi [8], that applies log-Gabor wavelets with pre-specified scales k for measuring the image signal symmetry and anti-symmetry at every pixel location. The local spectral information is computed by using two filters in quadrature.

Let us consider a row $\mathbf{r}$ of the symmetric image $\mathtt{I}_s$. We must quantify the signal symmetry at every pixel location i in order to find the point belonging to the contour where $\mathcal{C}$ is projected. The amount of symmetry can be measured by using a similar energy function to that used by Kovesi [8]

$$\mathtt{E}_s(i)^{\mathbf{r}} \;=\; \frac{\sum_{k=0}^{n-1} \mid \mathtt{s}_k(i) \mid - \mid \mathtt{a}_k(i) \mid}{\sum_k A_k(i)} \tag{7}$$

with $\mathtt{s}_k(i)$ and $\mathtt{a}_k(i)$ being the real and imaginary parts of the convolution of the image row $\mathbf{r}$ with the 1-D log-Gabor

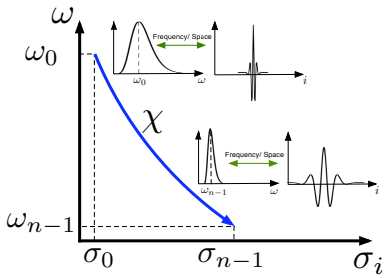


Figure 5. (Qualitative) space-frequency behavior of the log-Gabor wavelet for a shape-factor χ. σ_i represents the filter spatial extent, and ω_0 is the center frequency of the mother wavelet. Low wavelet scales k are well suited for the analysis of high-frequency signals (texture) with short spatial extent; on the contrary, high scales k are appropriate for low-textured regions with large spatial extent.

wavelet $\mathtt{logG_k}$, and n is the number of different wavelet scales used. The energy is computed as the sum of the difference between even and odd responses across wavelet scales. The normalization by the sum of the magnitudes A_k of the filter responses provides invariance to changes in illumination. Note that in a location i of symmetry the response of $\mathtt{s}_k$ is high and the response of $\mathtt{a}_k$ is low. By following the same reasoning, an anti-symmetry energy $\mathtt{E}_a$ is defined by summing the differences between odd and even responses across wavelet scales. Fig. 5 provides an intuition of the way the wavelet scale k relates with the space-frequency behavior of the log-Gabor filter for the case of a constant shape-factor χ.

As shown in Fig. 3, the convolution of the image $\mathtt{I}_s$ with the log-Gabor wavelets gives raise to a symmetry energy $\mathtt{E}_s$. An equivalent procedure is followed for generating an anti-symmetry energy $\mathtt{E}_a$ from $\mathtt{I}_a$. It can be observed that in both cases there are several local maxima that preclude a correct detection of the relevant contour using a single type of energy. The pixel-wise multiplication of $\mathtt{E}_s$ and $\mathtt{E}_a$ leads to $\mathtt{E}$ where the pixel locations with both types of energy are clearly highlighted. This joint energy $\mathtt{E}$ is the output of the SymStereo pipeline that will be often referred in the subsequent sections.

3.2. Efficient implementation for the case of a vertical pencil of cut planes

As described previously, a set of N virtual cut planes intersecting the baseline in its midpoint will be used to reconstruct sparsely the 3D scene. We will show in this section that SymStereo can be implemented efficiently without requiring the rendering of the images $\mathtt{I}_s$ and $\mathtt{I}_a$ for each cut plane.

From Eq.5, and assuming a virtual cut plane belonging to the vertical pencil and whose axis intersects the baseline in its midpoint, comes that

$$\mathsf{H}_\lambda \sim \begin{pmatrix} -1 & 0 & -\lambda \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{8}$$

This particular homography corresponds to a reflection (mirrorring) about the origin and a horizontal shift of λ (proportional to the rotation θ of the virtual cut plane).

In Section 3.1 we referred that the signals $\mathtt{s}_k$ and $\mathtt{a}_k$ in Eq.7 are obtained by a 1D convolution of an image row $\mathbf{r}$ of $\mathtt{I}_s = \mathtt{I} + \widehat{\mathtt{I}}$ with the log-Gabor kernels $\mathtt{logG}_k$ at scale k [1]. Since the convolution is a linear operator, we can first perform the convolution of the rows of $\mathtt{I}$ and $\widehat{\mathtt{I}}$ with $\mathtt{logG}_k$, and then the addition. Moreover, as $\widehat{\mathtt{I}}$ is computed using Eq. 8, $\widehat{\mathtt{I}}$ is simply a reflected and shifted version of $\mathtt{I}'$. This means that we only need to perform one convolution for the N virtual cut planes, namely

$$\begin{aligned} \left[\mathtt{R}_k, \mathtt{I}_k\right] &= \mathtt{I} * \mathtt{logG}_k \\ \left[\widehat{\mathtt{R}}_k, \widehat{\mathtt{I}}_k\right] &= \widehat{\mathtt{I}}_0 * \mathtt{logG}_k \end{aligned}$$

where $\mathtt{R}_k$ ($\widehat{\mathtt{R}}_k$) and $\mathtt{I}_k$ ($\widehat{\mathtt{I}}_k$) are the even and odd responses of the 1D convolution, and $\widehat{\mathtt{I}}_0$ is a mirrored version of $\mathtt{I}'$ (corresponding to the plane $\mathbf{\Pi}_0$). Finally, $\mathtt{s}_k$ and $\mathtt{a}_k$ of the various virtual cut planes are simply different combinations of $\mathtt{R}_k$ ($\widehat{\mathtt{R}}_k$) and $\mathtt{I}_k$ ($\widehat{\mathtt{I}}_k$) that depend on λ.

4. Experiments

This section evaluates the use of induced symmetry as a matching cost. As described previously, SymStereo is able to exclusively recover depth along a pre-specified cut plane, which provides a new controlled manner for probing into the 3D structure, useful for problems like piecewise-planar reconstruction [1]. However, this feature can also be achieved by adapting other methods in the literature. This section evaluates the usage of induced symmetry as a matching cost for reconstruction a virtual contour $\mathcal{C}$ against other matching costs, namely Zero-mean Normalized Cross-Correlation (ZNCC), Census filtering [16], and the sampling-insensitive absolute difference of Birchfield and Tomasi [2] in conjunction with background subtraction by bilateral filtering [15] (BilSub/BT), that were rated as top performers in a recent evaluation [7].

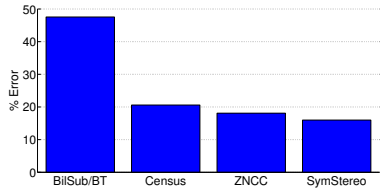
4.1. Quantitative evaluation

The methodology described in [7] is used as guideline for our experiments that compare SymStereo against

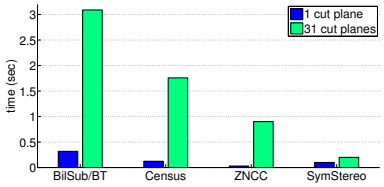
[1] It is important to refer that log-Gabor filters are analytical signals. This implies that instead of directly convolving the images with the pre-defined set n of wavelets, the filtering is achieved by taking the *DFT* of the rows of $\mathtt{I}_s$ and $\mathtt{I}_a$, multiply by the log-Gabor kernels and then take the *IDFT* to obtain the signals $\mathtt{s}_k$ and $\mathtt{a}_k$. This operations are performed using the Fast Fourier Transform (FFT), which is very efficient.

Figure 6. The experimental evaluation of the matching costs was carried on these 15 Middlebury stereo images [10, 7] (only the left image is shown), containing a wide variety of possible 3D scenarios e.g. slanted surfaces, different textures, depth discontinuities.

.



(a) Mean Errors



(b) Mean Runtime

Figure 7. Comparison of BilSub/BT, Census, ZNCC and SymStereo for the detection of the image of the profile contour $\mathcal{C}$. The evaluation was carried on 15 stereo pairs (see Fig.6), and for each stereo pair 31 virtual cut planes ware considered. The graphics show for each matching cost the mean percentage of incorrectly assigned disparities over the virtual cut planes and the mean runtimes for processing 1 and 31 virtual planes.

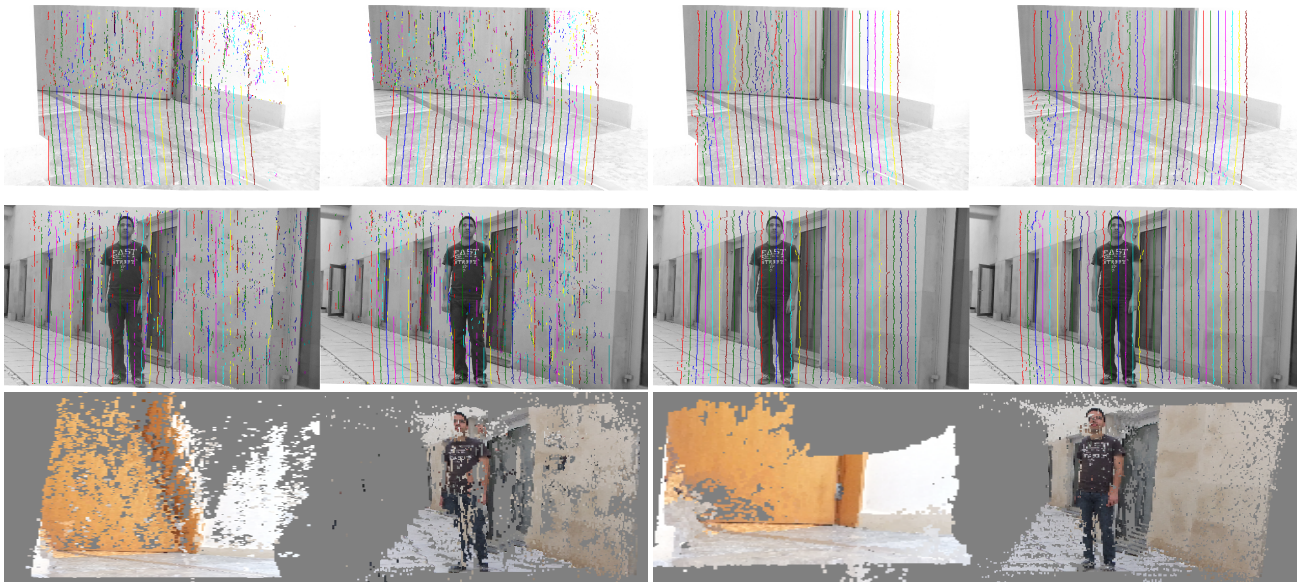
ZNCC, Census and BilSub/BT. The choice of ZNCC is justified by its popularity, whereas BilSub/BT because it is one of the best parametric matching costs of [7]. The Census filter was selected because it proved to be the top similarity measure for dense disparity estimation [7]. We decided to use a local stereo method for the final disparity selection because: (i) local aggregation is better suited for comparing stereo cost functions because it is more straight forward than global methods with many tuning parameters and sophisticated inference from priors; and (ii) it is better suited for real-time applications. Since we only analyze the 3D space along a virtual cut plane, the aggregation of the DSI is accomplished by summing over a 9×1 window (no horizontal disparity aggregation is used). The points with lowest cost are selected as being the image location of the profile cut in the reference view.

For each matching cost, we manually tune their parameters using the 4 standard evaluation images of the Middleburry data set [11]: (i) in [7] the Census filtering is accomplished using a 9×7 pixels window, we decided to use the same window height and to tune the width; (ii) for ZNCC we tuned the square window size; (iii) a similar procedure is followed for choosing the log-Gabor scales for the symmetry detection [8]; and (iv) the BilSub/BT parameters are the same as in [7]. After the tuning, the parameters are kept constant for the rest of the experiment using the more recent and challenging Middleburry stereo pairs [10, 7], see Fig. 6.

We assume a disparity range of 80 pixels for gray-level images with an approximate size of 450×370 pixels. For each input stereo pair we sample the 3D space by 31 virtual cut planes with the angles θ being chosen such that the distance between consecutive corresponding parallel planes $\mathbf{\Gamma}_\lambda$ in the DSI is maintained constant. As described in Section 2.3, each pixel on the image of the profile contour $\mathcal{C}$ implicitly corresponds to a disparity hypothesis. This allows us to evaluate the estimations by counting the number of pixels of the image of $\mathcal{C}$ whose disparity differs by more than 1 with respect to the ground truth. In this counting we exclude occluded image regions.

The average results in the detection of the profile contour $\mathcal{C}$ are depicted in the graphics of Fig. 7. From Fig. 7(a) we observe that, although SymStereo is a 1D metric and ZNCC and Census are 2D metrics, the former is the top-performer in terms of accuracy, with 2.1% less errors compared to ZNCC, and 4.6% less errors compared to Census. Surprisingly, the performance of Census is worst than ZNCC for the detection of the contour $\mathcal{C}$, which is in contrast with the evaluation results of [7]. This means that ZNCC has more discriminative power along a virtual cut plane than Census. Concerning BilSub/BT, since it only analyzes a very small 1D neighborhood, there is not enough support to locate the correct matches along the virtual plane. From the evaluation we might conclude that SymStereo is specially well suited for estimating depth along virtual cut planes, outperforming state-of-the-art matching costs.

In addition to the performance of depth estimation of the different matching costs, the runtime is also an important issue for many applications e.g. autonomous vehicles and robots. All matching costs were implemented in C++. The runtimes were measured on an Intel Core Q720 1.6GHz CPU laptop. Note that the runtime can vary on different CPU architectures (including their relative sizes), and that some implementation tricks can speed-up the matching processes. However, this evaluation provides an approximate idea about the computational effort. Fig. 7(b) illustrates the mean runtime of the different matching costs over the stereo data sets of Fig. 6, for one and for 31 virtual cut



(a) Results obtained using ZNCC (b) Results obtained using SymStereo

Figure 8. Qualitative evaluation of different matching costs for detecting the image of the profile contour $\mathcal{C}$. The first row concerns the *Door* dataset, the second row concerns the *Pedestrian* dataset, and the last row shows the 3D reconstructions obtained. In the first two rows the 3D space was sampled by 31 virtual cut planes, while in the last row 161 were employed.

planes. In the case of a single virtual cut plane, the fastest matching cost is ZNCC, approximately three times faster than SymStereo and four times faster than Census. This comes from the fact that before comparing directly matching hypotheses in the two views, SymStereo and Census need a pre-processing step: Census calculates a bit string for each pixel, which encodes the intensity distributions in the local neighborhood; while SymStereo starts by convolving the left and right views with the n log-Gabor wavelets (see Section 3.2). After the first cut plane, Census and SymStereo only need pixelwise comparisons, which are fast, while ZNCC continues to compare square windows in the two views. Remark that the processing of 31 virtual cut planes is more than 4 times faster using SymStereo than ZNCC. BilSub/BT is comparatively very slow due to the bilateral filtering. This aspect can eventually be improved using an approximate separable implementation.

4.2. Qualitative evaluation

In this section, we compare ZNCC and SymStereo in images of scenes with high surface slant and/or regions of low texture (see Fig. 8). We assume a disparity range of 350 pixels for gray-level images with an approximate size of 1280×1024 pixels. Since these images are roughly three times larger than the images in the previous section (Fig. 6), the parameters of the matching costs were tuned accordingly.

In order to access the reconstruction accuracy, each profile cut is projected in the two views (see Fig. 8). A correct cut gives raise to a pair of image contours going through corresponding image pixels.

The results are shown in Fig. 8. In the *Door* example it can be seen that both ZNCC and SymStereo do a good job in the floor surface. However, ZNCC clearly fails in reconstructing the door and the white wall. SymStereo manages that because of the global character of the symmetry cue. In the *Pedestrian* example, SymStereo also outperforms ZNCC. Note that using SymStereo the low-textured cloths of the pedestrian are accurately reconstructed, and the high slant of the wall causes almost no difficulties.

5. Conclusions

The article describes a new stereo framework, dubbed SymStereo, that uses symmetry for determining the scene depth along virtual cut planes. The virtual cut planes constitute a new manner of probing into the 3D structure, enabling a trade-off between computational effort and sparseness of reconstruction that preserves depth resolution. This is a useful feature for applications in robotics and autonomous vehicles that have simultaneous requirements in terms of time and accuracy. Moreover, we provide convincing evidence that symmetry is better suited than photo-consistency for the purpose of sparse stereo. The experiments in reconstructing profile cuts in Middlebury images clearly show that SymStereo outperforms state-of-the-art matching costs [7] both in terms of accuracy and computational overhead.

Acknowledgements

This work was supported in part by the Portuguese Foundation for Science and Technology (FCT) under the grant PTDC/EEA-AUT/113818/2009. Michel Antunes is grateful to the Portuguese Foundation for Science and Technology (FCT) by generous funding through the grant SFRH/BD/47488/2008.

References

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

[2] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[3] R. T. Collins. A space-sweep approach to true multi-image matching. In *CVPR*, 1996.

[4] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.

[5] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. In *CVPR*, 2007.

[6] D. B. Gennery. A stereo vision system for an autonomous vehicle. In *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, 1977.

[7] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *Trans. PAMI*, 2009.

[8] P. Kovesi. Symmetry and asymmetry from local phase. In *Australian Joint Conf. on Artificial Intelligence*, 1997.

[9] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. SpringerVerlag, 2003.

[10] D. Scharstein and C. Pal. Learning conditional random fields for stereo. *CVPR*, 2007.

[11] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 2001.

[12] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005.

[13] R. Szeliski. *Computer Vision : Algorithms and Applications*. Springer, 2010.

[14] R. Szeliski and D. Scharstein. Sampling the disparity space image. *Trans. PAMI*, 2004.

[15] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. ICCV '98, 1998.

[16] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV*, 1994.