Int J Computer Vision manuscript No.
(will be inserted by the editor)

SymStereo: Stereo Matching using Induced Symmetry

Michel Antunes · João P. Barreto

Received: date / Accepted: date

Abstract Stereo methods always require a matching function for assessing the likelihood of two pixels being in correspondence. Such functions, commonly referred as *matching costs*, measure the photo-similarity (or dissimilarity) between image regions centered in putative matches. This article proposes a new family of stereo cost functions that measure symmetry instead of photo-similarity for associating pixels across views. We start by observing that, given two stereo views and an arbitrary virtual plane passing in-between the cameras, it is possible to render image signals that are either symmetric or anti-symmetric with respect to the contour where the virtual plane meets the scene. The fact is investigated in detail and used as cornerstone to develop a new stereo framework that relies in symmetry cues for solving the data association problem. Extensive experiments in dense stereo show that our symmetry-based cost functions compare favorably against the best performing photo-similarity matching costs. In addition, we investigate the possibility of accomplishing *Stereo Rangefinding* that consists in using passive stereo to exclusively recover depth along a pre-defined scan plane. Thorough experiments provide evidence that stereo from induced symmetry is specially well suited for this purpose.

Keywords dense stereo matching · matching cost · symmetry · stereo rangefinder · SRF

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

1 Introduction

Passive methods for stereo correspondence invariably require a metric for assessing the likelihood of two image locations being a match. Typically, the first step of a dense stereo algorithm is the evaluation of this matching function across all possible disparities and pixel locations. The result is the so-called Disparity Space Image (DSI) [30], over which is carried either local aggregation or global optimization with the objective of finding the correct depth map [27]. Local stereo methods aggregate the matching function over a support region for obtaining a spatially coherent DSI [15, 33]. This is usually followed by a Winner-Takes-All (WTA) procedure along the disparity dimension that leads to the final depth assignment. In global stereo methods, the pixel correspondence between views is formulated as a global optimization problem over the DSI that is solved using an energy minimization framework for obtaining the final disparity map [31]. There is still a third strategy called Semi-Global Matching (SGM) that minimizes a 2D energy function defined over the DSI by performing path-wise optimization along multiple 1D directions [16].

The present article revisits the construction of the DSI using a suitable matching function. We focus exclusively in this initial step that is common to any stereo algorithm independently of using local or global optimization. A new family of matching costs is proposed, studied, and evaluated for the first time. The functions described so far in the stereo literature rely, in one way or the other, in measuring the photo-consistency between two image locations. We show in this paper that, given a calibrated stereo pair, it is possible to render image signals that are either symmetric or anti-symmetric around the projection of the contour where an arbitrary virtual cut plane intersects the scene. This allows using symmetry instead of photo-consistency for quantifying the likelihood of two pixels being a match. We show through extensive comparative experiments that symmetry-based metrics outperform photo-similarity for the purpose of data association in dense stereo. Moreover, and since the symmetries are induced using virtual cut planes, these new matching functions are particularly well suited for recovering depth along pre-defined scan planes. As discussed in [2], this is an effective way of probing into the 3D structure resulting in profile cuts of the scene that resemble the ones obtained with a 2D Laser Rangefinder (LRF) [4]. The independent estimation of depth along a scan plane will be referred as *Stereo Rangefinder (SRF)* in order to be distinguished from conventional dense stereo. To the best of our knowledge this is also the first work that discusses and benchmarks the concept of SRF.

1.1 Related Work

Dense stereo matching is a mature research topic and the literature reports a large number of matching functions. We provide below a non-exhaustive account of representative cost functions organized according to the taxonomy used in [17]:

- *Pixel-wise matching cost*, like Absolute Differences (AD), measure the dissimilarity between single pixels, being popular because of their simplicity and fast computation. However, pixel-wise metrics tend to be ambiguous even when used in conjunction with local aggregation methods, e.g. Sum of Absolute Differences (SAD). Since pixel-wise matching functions do not make implicit assumptions about the image neighborhood surrounding the pixel, they are broadly used for evaluating the DSI in global stereo approaches. In this case, the sampling-insensitive metric proposed by Birchfield-Tomasi (BT) is usually preferred to a straightforward AD implementation. BT computes the absolute difference between the pixel of interest in one view and a linear interpolation of the neighborhood of the hypothesized match in the other view [7]. A pre-processing step that significantly improves the stereo matching performance of BT is Bilateral Background Subtraction (BBS) that smoothes the images without blurring the depth discontinuities [1].
- *Window-based matching cost* evaluate the similarity (or dissimilarity) between 2D regions in the stereo images. Normalized Cross-Correlation (NCC) is an example of this type of matching functions that is widely used because of its good trade-off between accuracy and computational efficiency. Zero-mean Normalized Cross-Correlation (ZNCC) is a variant of NCC that compensates for gains and offsets across stereo images in order to achieve more accurate and robust matching results [27].
- *Non-parametric matching costs* use the ordering of image intensities in a local neighborhood around the pixels of interest. The most popular metric of this type is probably the Census filter introduced in [36]. The approach consist in constructing a bit string where each bit corresponds to a pixel in a local neighborhood around the pixel of interest $\mathbf{q}$. The bit is set *iff* the pixel intensity value is lower than the intensity of $\mathbf{q}$. The filtered images are compared by computing the Hamming distance between corresponding bit strings.
- *Mutual Information* computed from the entropy of the input images can also be used as a stereo matching cost, as discussed in [16]. The idea is to transform views according to the disparity assignment such that the mutual information between the transformed stereo images is maximized.

Several works in stereo have benchmarked not only competing matching costs [14, 27, 6, 10, 12, 17], but also cost aggregation methods [27, 10, 34, 15, 25, 33] and global optimization schemes [27, 10, 31]. In this article, we are only interested in the formers, among which the work of Hirschmuller and Scharstein [17] is of special relevance because of its systematic methodology and thorough evaluation using images of the Middleburry dataset [27, 26]. In their evaluation each cost function gives rise to a DSI that leads to a final disparity map after using local aggregation, SGM, or a straightforward Markov Random Field formulation with Graph-Cut (GC) optimization. The results show that BT with BBS, ZNCC, and Census are, respectively, the top-performers among pixel-wise, window-based, and non-parametric matching costs. In absolute terms, Census proved to have the best matching performance throughout the evaluation. In Sections 5 to 7 we use the exact same methodology of [17] for comparing our symmetry-based matching costs against BT with BBS, ZNCC, and Census, in an effort to show that symmetry can be more effective than photo-similarity for solving the stereo data association problem.

To the best of our knowledge the usage of induced symmetries for the purpose of stereo matching has never been reported in the literature [1]. The only exceptions are our preliminary conference papers that use symmetry-based SRF for the detection and reconstruction of planar surfaces [5, 3], for robotic applications with strict time requirements [2], and for mimic a LRF [4]. However, these prior works focus more in showing that stereo from symmetry can be helpful for solving specific problems rather than in providing a thorough discussion and evaluation of the framework.

[1] In [29] , the term symmetry is employed with a completely different meaning, referring to the equal treatment of left and right views.

1.2 Article Overview

Section 2 provides an intuitive description of the *mirroring effect* that is induced by a virtual plane intersecting the baseline. The mirroring effect is the cornerstone of our SymStereo framework because it enables the rendering of image signals that are either symmetric or anti-symmetric with respect to the contour where the virtual plane cuts the scene. The stereo matching is achieved by finding the image of this contour in the two views using symmetry cues. Section 3 refers to the geometric analysis of the framework. We provide a rigorous formal proof of the mirroring effect, discuss singular configurations, and show how to select the virtual cut planes for generating a complete DSI. Section 4 derives suitable symmetry metrics for quantifying the likelihood of a certain image pixel being locally symmetric and/or anti-symmetric. We propose three different symmetry-based matching costs: (i) *SymBT* that is a modification of BT for measuring symmetry instead of similarity [7]; (ii) *SymCen* that is a non-parametric symmetry metric inspired in the Census transform [36]; and (iii) *logN* that uses a bank of log-Gabor wavelets for quantifying symmetry, inspired by the work of Kovesi in [19].

Sections 5, 6, and 7 describe several experiments that validate the SymStereo framework and compare the accuracy of symmetry-based matching costs against state-of-the-art photo-similarity cost functions. Section 5 reports experiments in dense stereo using 15 images of the Middlebury data set [27, 26] and the Oxford Corridor stereo pair. We follow the methodology described in [17], with the parameters of competing methods being tuned using the four standard Middlebury images [27] that are not considered for the final evaluation. The results show that symmetry-based costs outperform the corresponding photo-similarity counterparts, with *SymBT* and *SymCen* systematically beating BT and Census [7, 36]. Section 6 repeats the tests of Section 5 for the case of Stereo Rangefinder (SRF) [5, 2]. While dense stereo estimates the depth of the entire viewed scene, SRF recovers the depth exclusively along a pre-defined virtual plane, giving rise to a so-called *profile cut* of the scene. Since SRF does not evaluate the entire DSI, neither local 2D aggregation, nor standard stereo optimization methods can be employed. The experiments show that, under such circumstances, the symmetry-based cost *logN* is clearly the top-performer with 4% less errors than the second ranked. Finally, Section 7 runs tests in the images of the Fountain-p11 dataset [28] providing evidence that the conclusions above generalize for the case of wide-baseline stereo.

1.3 Notation and Terminology

We represent scalars in italic, e.g. s, vectors in bold characters, e.g. $\mathbf{p}$, $\mathbf{\Pi}$, matrices in sans serif font, e.g. M, image signals in typewriter font, e.g. $\mathtt{I}$, and curves in calligraphic symbols, e.g. $\mathcal{C}$. 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 a scale. Finally, $[\mathbf{v}]_\times$ denotes the skew symmetric matrix defined by the 3-vector $\mathbf{v}$, and $\mathsf{I}_{3\times3}$ refers to the 3×3 identity matrix.

Although SymStereo can be used with any stereo pair, the article assumes rectified stereo for most derivations and experiments. Thus, a generic 1-D line of the image signal $\mathtt{I}$ is denoted by $\mathtt{I}(p_1)$, with p_1 being the free coordinate along the horizontal axis. The 1-D signal $\mathtt{I}(p_1)$ has a local symmetry about a point q_1 in its domain *iff* the following holds:

$$\mathtt{I}(q_1+\delta) \;=\; \mathtt{I}(q_1-\delta),\, \forall_{\delta\,\in\,\mathcal{N}}$$

with $\mathcal{N}$ being an interval centered in zero. In a similar manner, $\mathtt{I}(p_1)$ is said to be anti-symmetric in a local neighborhood around q_1 *iff*

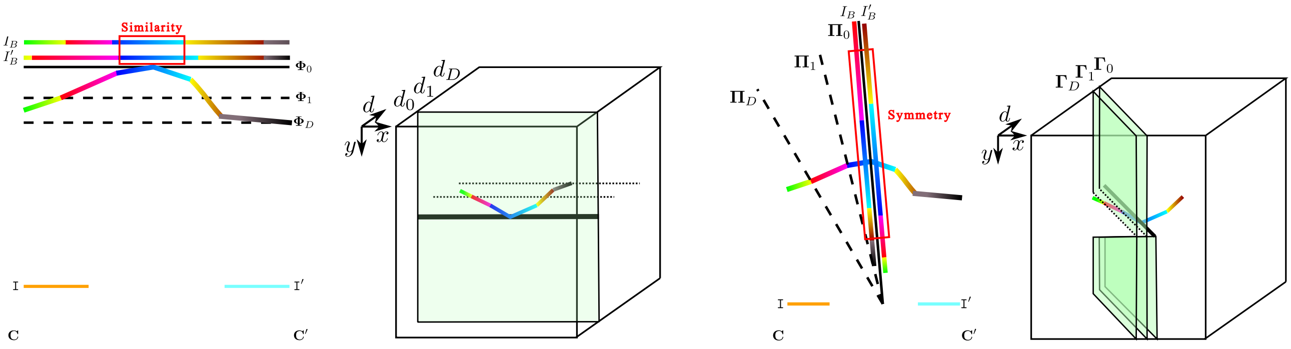
$$\mathtt{I}(q_1)-\mathtt{I}(q_1+\delta) \;=\; -(\mathtt{I}(q_1)-\mathtt{I}(q_1-\delta)),\, \forall_{\delta\,\in\,\mathcal{N}}$$

The stereo matching will be carried by quantifying 1-D signal symmetry and anti-symmetry in successive pixel locations along epipolar lines.

We will often refer to a matching function as being a ”matching cost” or a ”cost function” without distinguishing if the function measures photo-similarity, photo-dissimilarity, local symmetry, or lack of local symmetry. We will also employ the term ”similarity-based matching cost” to designate matching functions that use conventional photo-consistency metrics, as opposed to the new stereo functions that exploit induced symmetry cues.

2 Mirroring Effect and Stereo from Induced Symmetry

Let $\mathtt{I}$ and $\mathtt{I}'$ be a pair of rectified stereo images acquired by two cameras with projection centers $\mathbf{C}$ and $\mathbf{C}'$. The scheme of Fig. 1(a) is a top-view of this situation, where the two cameras observe a concave surface S with five regions of different colors. The 3D volume of Fig. 1(b) is the corresponding DSI, with each point $(\mathbf{p}, d)$ representing the disparity hypothesis d for the pixel location $\mathbf{p} = (x, y)$ [30]. We can understand the stereo matching cost as a scalar function with domain $(\mathbf{p}, d)$, and the DSI as the result of evaluating this function across the entire domain. Ideally, the cost function should be such that, for each image point $\mathbf{p}$ there is one, and only one, extremum along the disparity axis that signals the correct disparity value d. In this case, the set of all extrema define a surface in the DSI that enables the accurate 3D reconstruction of the scene. In practice, several ambiguities arise, and the evaluation of the matching cost



(a) Plane Sweeping (b) DSI Plane Sweeping (c) SymStereo (d) DSI SymStereo

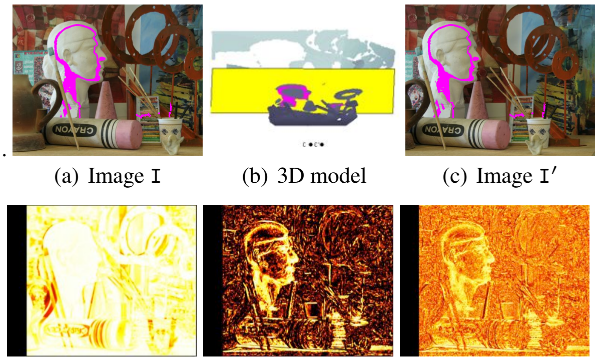
Fig. 1 Plane Sweeping vs SymStereo. (a) and (b): Conventional stereo matching can be understood as a particular instance of plane sweeping [11]. The DSI is evaluated for increasing values of disparity d_i. Each disparity hypothesis d_i is associated with a virtual plane $\mathbf{\Phi}_i$ that is fronto-parallel. The chosen matching cost implicitly measures the photo-similarity between $\mathtt{I}_B$ and $\mathtt{I}'_B$, that are the results of back-projecting $\mathtt{I}$ and $\mathtt{I}'$ onto $\mathbf{\Phi}_i$; (c) and (d) - In SymStereo the virtual planes $\mathbf{\Pi}_i$ pass between the cameras, and the back-projection images are reflected with respect to the curve where $\mathbf{\Pi}_i$ intersects the scene structure (mirroring effect). This enables to perform stereo matching using symmetry instead of photo-similarity. In the same manner that each plane $\mathbf{\Phi}_i$ in (a) is associated with a constant disparity plane in (b), each plane $\mathbf{\Pi}_i$ in (c) corresponds to an oblique plane $\mathbf{\Gamma}_i$ in (d). Thus, the entire DSI domain can be fully covered by carefully choosing the set of virtual cut planes $\mathbf{\Pi}_i$.

usually leads to multiple incorrect extrema. The steps of local aggregation and/or global optimization over the DSI aim to overcome this problem by refining the matching surface taking into account spatial consistency criteria.

It is well known that, for the case of rectified stereo, the image pairs of points lying in a fronto-parallel plane $\mathbf{\Phi}_0$ are related by the same disparity amount d_0. Thus, the disparity plane d_0 in the DSI can be evaluated by back-projecting the two input views, $\mathtt{I}$ and $\mathtt{I}'$, onto the virtual plane $\mathbf{\Phi}_0$, followed by comparing the results $\mathtt{I}_B$ and $\mathtt{I}'_B$ using some type of photo-similarity metric. As shown in the scheme of Fig. 1 (a), the back-projected images $\mathtt{I}_B$ and $\mathtt{I}'_B$ overlap in the points where $\mathbf{\Phi}_0$ intersects the scene surface and, consequently, the quantification of photo-similarity tends to highlight these image locations enabling a correct disparity assignment. This way of addressing the problem was first introduced by Collins that suggested to find matches across multiple views by sweeping the 3D space with a pre-defined set of parallel virtual planes [11]. The computation of the DSI in rectified stereo can be understood as a particular instance of *plane sweeping*, with the sweeping direction being parallel to the camera axis, and each plane $\mathbf{\Phi}_i$ corresponding to a constant disparity d_i (see Fig. 1(a)-(b)).

SymStereo relates with plane sweeping in the sense that it also samples the 3D space by a set of virtual planes. However, there are two major differences: (i) the virtual planes must pass in between the cameras, which is considered to be a degenerate configuration in plane sweeping [13]; and (ii) the pixel association between views is achieved using symmetry cues instead of photo-similarity metrics.

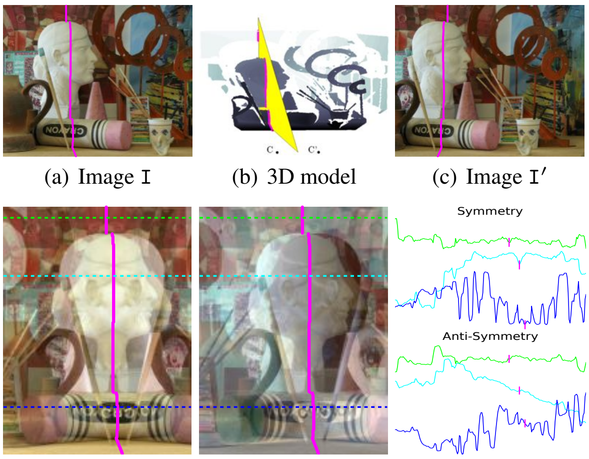
Consider the scheme of Fig. 1(c), with $\mathbf{\Pi}_0$ being a plane that passes between the cameras, and $\mathtt{I}_B$ and $\mathtt{I}'_B$ being the result of back-projecting views $\mathtt{I}$ and $\mathtt{I}'$ onto $\mathbf{\Pi}_0$. Remark that, while in Fig. 1(a) the back-projection images correlate in the pixel locations where the virtual plane meets the 3D surface, in Fig. 1(c) the images $\mathtt{I}_B$ and $\mathtt{I}'_B$ are mirrored with respect to the curve $\mathcal{C}$ where $\mathbf{\Pi}_0$ intersects the scene structure. SymStereo explores this *mirroring effect* for accurately reconstructing the contour $\mathcal{C}$ (the *profile cut*) using image symmetry analysis. The strategy is effective, not only for recovering depth along a pre-defined virtual cut plane (SRF), but also for achieving dense stereo reconstruction. It can be proved that the mirroring effect holds for any plane $\mathbf{\Pi}_i$ intersecting the baseline, corresponding an oblique plane $\mathbf{\Gamma}_i$ in the DSI domain. Thus, and in a similar manner to plane sweeping, it is possible to carefully select the virtual cut planes such that the DSI is fully evaluated and the correct disparity surface is recovered (Fig. 1 (d)).



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

(d) SAD9 (e) ZNCC9 (f) Census1

Fig. 2 Conventional stereo matching costs based in photo-similarity. $\mathtt{I}$ and $\mathtt{I}'$ are stereo views of the 3D scene shown in (b). The virtual plane $\mathbf{\Phi}_0$ (yellow) corresponds to a constant disparity d_0 in the DSI domain. Let $\widehat{\mathtt{I}}$ be the result of mapping $\mathtt{I}'$ into $\mathtt{I}$ using the plane-homography. The disparity hypothesis d_0 is evaluated by measuring the photo-similarity between $\mathtt{I}$ and $\widehat{\mathtt{I}}$, such that the image of the regions where $\mathbf{\Phi}_0$ intersects the scene structure becomes highlighted (d)-(f).



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

(d) $\mathtt{I}^S = \mathtt{I} + \widehat{\mathtt{I}}$ (e) $\mathtt{I}^A = \mathtt{I} - \widehat{\mathtt{I}}$ (f) Epipolar Lines

Fig. 3 SymStereo: The *virtual cut plane* $\mathbf{\Pi}_0$ in yellow intersects the scene structure in a non-continuous 3D curve $\mathcal{C}$ marked in magenta (the profile cut). Let $\widehat{\mathtt{I}}$ be the result of warping $\mathtt{I}'$ by the plane-homography induced by $\mathbf{\Pi}_0$. 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 $\mathcal{C}$ (d)-(e). In (f) we show the pixel intensities of $\mathtt{I}^S$ and $\mathtt{I}^A$ along three distinct epipolar lines (green, cyan and blue). Remark that the intersections with the locus where $\mathcal{C}$ is projected can be identified with almost no ambiguity by searching common pixel locations for which the top and bottom 1D-signals are respectively locally symmetric and anti-symmetric.

Fig. 2 aims to illustrate the evaluation of the disparity hypothesis d_0 using a conventional stereo matching cost such as SAD, ZNCC, or Census. The plane $d = d_0$ in the DSI domain (Fig. 1(b)) corresponds to a fronto-parallel virtual plane $\mathbf{\Phi}_0$ that is marked in yellow in the 3D model of Fig. 2(b). Let $\widehat{\mathtt{I}}$ be the warping result of mapping the right view $\mathtt{I}'$ into the left reference view using the plane-homography induced by $\mathbf{\Phi}_0$. For the particular case of rectified stereo, the warping is a simple image shift by d_0 pixels along the horizontal axis. The DSI values of the points lying in the plane $d = d_0$ is determined by measuring the similarity between images $\mathtt{I}$ and $\widehat{\mathtt{I}}$ using a specific metric. As shown by the results of Fig. 2(d)-(f), this enables depth recovery by highlighting the pixel locations corresponding to the regions where $\mathbf{\Phi}_0$ intersects the scene structure (magenta marks in Fig. 2(a)-(c)).

In this paper, the DSI is evaluated using a radically different strategy. Consider the virtual cut plane $\mathbf{\Pi}_0$ that intersects the scene surfaces in the profile cut $\mathcal{C}$ marked with magenta in the model of Fig. 3(b). Let H be the plane-homography associated with $\mathbf{\Pi}_0$ that maps the right image into the reference view. If $\widehat{\mathtt{I}}$ is the warping result of mapping $\mathtt{I}'$ by H then, it comes from the mirroring effect, that $\mathtt{I}$ 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 $\mathcal{C}$ is projected (Fig. 3(d)). 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 (Fig. 3 (e)). 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 (Fig. 3 (f)). This provides an implicit manner of recovering depth along $\mathbf{\Pi}_0$ and achieving data association across views. Since $\mathbf{\Pi}_0$ is mapped into an oblique plane $\mathbf{\Gamma}_0$ in the DSI domain, the joint symmetry and anti-symmetry metric assigns a matching cost to every point $(\mathbf{p}, d)$ lying on $\mathbf{\Gamma}_0$. Thus, and as stated above, the DSI can be fully evaluated by stacking the results of a set of planes $\mathbf{\Pi}_i$, such that the corresponding planes $\mathbf{\Gamma}_i$ cover the entire $(\mathbf{p}, d)$ domain (Fig. 1(d)).

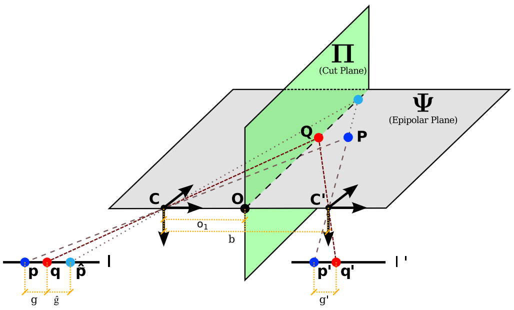


Fig. 4 Geometric analysis of SymStereo. The analysis is carried in an arbitrary epipolar plane $\mathbf{\Psi}$ assuming that the images are rectified. The camera centers $\mathbf{C}$ and $\mathbf{C}'$ are separated by a distance $b > 0$ (the stereo baseline), and the world frame is coincident with the coordinate system of the left view (the reference view). For the sake of graphical clarity the image points are projected behind the optical centers.

3 Geometric Analysis of SymStereo

This section derives the conditions for a generic 3D plane $\mathbf{\Pi}$ to intersect the baseline, proves that the mirroring effect holds for any virtual plane passing between the cameras *iff* corresponding image pixels have the same order in both views, and discusses the mapping of planes $\mathbf{\Pi}_i$ in 3D space into planes $\mathbf{\Gamma}_i$ in the DSI domain.

3.1 Necessary and sufficient condition for a virtual plane $\mathbf{\Pi}$ to intersect the baseline

Consider a rectified stereo pair that is acquired by two cameras with centers in $\mathbf{C}$ and $\mathbf{C}'$ as shown in Fig. 4. Since the camera reference frames are aligned, the transformation T, that maps right view coordinates into left view coordinates, is

$$\mathsf{T} = \begin{pmatrix} \mathsf{I}_{3\times3} & \mathbf{t} \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix}, \tag{1}$$

with

$$\mathbf{t} = \begin{pmatrix} b \\ 0 \\ 0 \end{pmatrix}.$$

We assume, without loss of generality, that the world coordinate system is coincident with the reference frame centered in $\mathbf{C}$. The virtual cut plane $\mathbf{\Pi}$, that passes between the cameras, is represented by the following homogeneous vector

$$\mathbf{\Pi} \sim \begin{pmatrix} \mathbf{n} \\ -h \end{pmatrix} , \tag{2}$$

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

$$\mathbf{n} \sim \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} .$$

In addition, the centers $\mathbf{C}$ and $\mathbf{C}'$ define a line $\mathbf{L}$ that contains the baseline and has Plücker coordinates [22]

$$\mathbf{L} \sim \begin{pmatrix} \mathbf{t} \\ \mathbf{0} \end{pmatrix} .$$

The intersection of the virtual cut plane with the baseline can be efficiently computed by multiplying the 4-vector $\mathbf{\Pi}$ with the Plücker matrix of the dual of $\mathbf{L}$ [24]. It follows that the homogeneous coordinates of the intersection point $\mathbf{O}$ are

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

Using β to denote the ratio between the signed distances $\mathbf{CO}$ and $\mathbf{CC}'$, it comes that the plane $\mathbf{\Pi}$ passes between the cameras *iff* the following condition holds

$$0 < \left(\beta = \frac{O_1}{b}\right) < 1 \quad \Longleftrightarrow \quad \frac{b\, n_1}{h} > 1. \tag{3}$$

3.2 Proof of the Mirroring Effect

Consider a generic 3D point $\mathbf{P}$ that is projected into points $\mathbf{p}$ and $\mathbf{p}'$ in the stereo views as shown in Fig. 4. Since we are assuming rectified stereo, then the non-homogeneous coordinates p_2 and p'_2 must have the same value y. In a similar manner, consider a point $\mathbf{Q}$ that lies in the intersection of the same epipolar plane $\mathbf{\Psi}$ with the virtual plane $\mathbf{\Pi}$. Since the image points $\mathbf{p}$, $\mathbf{q}$ in the left view, and $\mathbf{p}'$, $\mathbf{q}'$ in the right view, only differ in terms of the first coordinates, then we can define the following pair of signed distances:

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

Remark that g and g' have the same signal *iff* the points $\mathbf{P}$ and $\mathbf{Q}$ are imaged with the same order in the two views. We assume henceforth that this condition holds.

The plane $\mathbf{\Pi}$ defines a homography H that can be used to map points from the right view into the left view. Given the relative camera pose of Equation 1 and the homogeneous plane representation of Equation 2, it comes that [22]

$$\mathsf{H} \sim \left(\mathsf{I}_{3\times3} + \frac{\mathbf{t}\,\mathbf{n}^\mathsf{T}}{h}\right)^{-1} \sim \begin{pmatrix} 1 + \frac{bn_1}{h-bn_1} & \frac{bn_2}{h-bn_1} & \frac{bn_3}{h-bn_1} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5}$$

Using H to map $\mathbf{p}'$ in the right view onto $\widehat{\mathbf{p}}$ in the left view yields

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

with k_y depending on the second coordinate y and being a constant for points sharing the same epipolar line. From Equation 4 it comes that $p'_1 = g' + q'_1$ and the expression above can be re-written as

$$\widehat{p}_1 = \left(1 + \frac{bn_1}{h - bn_1}\right) q'_1 + k_y + \left(1 + \frac{bn_1}{h - bn_1}\right) g'. \tag{6}$$

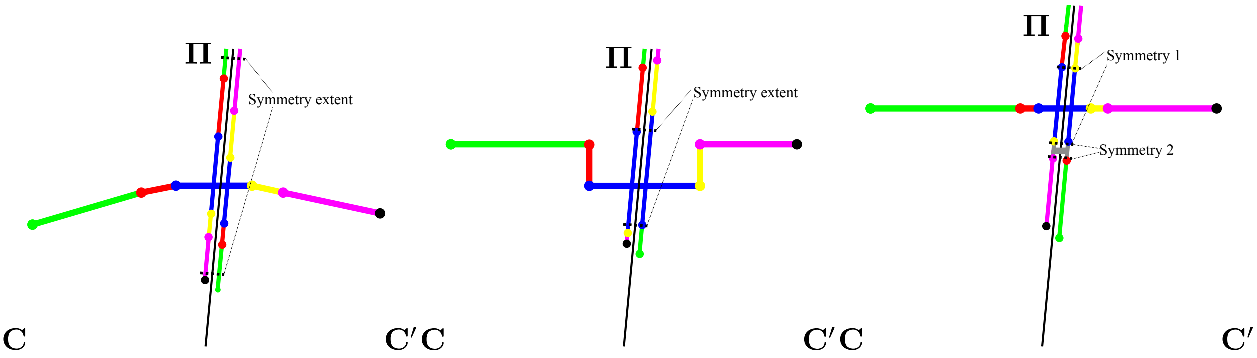
In a similar manner let $\widehat{\mathbf{q}}$ be the mapping result of $\mathbf{q}'$ such that $\widehat{\mathbf{q}} \sim \mathsf{H}\,\mathbf{q}'$. Since $\mathbf{Q}$ lies in the cut plane $\mathbf{\Pi}$ that defines the homography, then point $\widehat{\mathbf{q}}$ must be coincident with $\mathbf{q}$ and the following holds

$$q_1 = \left(1 + \frac{bn_1}{h - bn_1}\right) q'_1 + k_y .$$

Replacing the result above in Equation 6 comes that the signed image distance between $\mathbf{q}$ and $\widehat{\mathbf{p}}$ is

$$\widehat{g} = \widehat{p}_1 - q_1 = \left(1 - \frac{bn_1}{h}\right)^{-1} g' . \tag{7}$$

For the case of the virtual plane $\mathbf{\Pi}$ passing between the cameras, the condition of Equation 3 holds, which means that g' and $\widehat{g}$ have opposite signs. Thus, and assuming that distances g and g' have always the same signal, we have just proved that points $\mathbf{p}$ and $\widehat{\mathbf{p}}$ must be on opposite sides of $\mathbf{q}$, and that the mirroring effect holds for any plane $\mathbf{\Pi}$ that intersects the baseline. Nothing is said about the modulus of the distances g and $\widehat{g}$ that must be equal in order for the image symmetry of Fig. 3(d) to be geometrically accurate. It can be analytically shown that in general $|g| \neq |\widehat{g}|$, leading to a deviation in the rendered symmetry that depends both on the point where $\mathbf{\Pi}$ intersects the baseline, and on the position and slant of the imaged 3D surface. The present paper does not pursue the topic further, however we can advance that this deviation has limited practical impact, as proved by the experimental results of Sections 5 to 7.



(a) Continuous surface (b) Occlusion (c) Double Nail Illusion

Fig. 5 (a) In the case the virtual cut plane $\mathbf{\Pi}$ intersects the scene in a continuous surface, most of the back-projected image regions contribute for the mirroring effect. (b) In the presence of occlusions (the yellow region is occluded in the left view and the red region is occluded in the right view), the symmetry extend is reduced and limited by the depth occlusion boundaries. (c) In the presence of double nail illusion, the virtual cut plane intersects two surfaces, in which case the mirroring effect occurs in two distinct regions - one corresponding to the surface in front (grey) and one corresponding to the surface in the back (blue).

3.3 Singular Configuration

We have proved that the homography associated with a cut plane causes a reflection *iff* the scene points are projected in the two views in the same order. For most stereo applications, the spatial order of corresponding points in the two views is the same, and the mirroring effect is verified (refer to Fig. 5 (a) and (b)). However, there is a singular configuration for which the ordering constraint is not verified. This configuration, known as *double nail illusion*, typically arises in scenes with thin foreground objects or narrow holes [29]. Consider the scheme of Fig. 5 (c), in which case the thin foreground object (grey) causes a double nail illusion - the grey region is projected to the right of the blue region in the left view, while to the left in the right view. In this case, the virtual cut plane $\mathbf{\Pi}$ intersects the scene in two distinct regions (grey and blue) visible by both cameras. The mirroring effect occurs in both regions and two different symmetries are induced using SymStereo, each one precluding the detection of the other. Since the double nail illusion arises seldom in practice, we will ignore it for the rest of the paper, and consider that the mirroring effect is always verified, with the cut plane intersecting the scene in a single point per epipolar line.

3.4 Mapping $\mathbf{\Pi}$ into a plane $\mathbf{\Gamma}$ in the DSI domain.

In the same manner that a fronto-parallel plane $\mathbf{\Psi}$ induces a constant disparity d, a virtual cut plane $\mathbf{\Pi}$ defines a pixel association between views that corresponds to a particular surface $\mathbf{\Gamma}$ in the DSI domain (see Fig. 1). Let's consider the inverse of the plane homography given by Equation 5. The transformation H^{-1} enables to map points $\mathbf{q}$ in the left image into points $\mathbf{q}'$ in the right image, such that

$$q'_1 = (1 + \frac{bn_1}{h})\, q_1 + \frac{bn_2}{h}\, q_2 + \frac{bn_3}{h}\,. \tag{8}$$

It can be verified that the cut plane $\mathbf{\Pi}$ defines for each point $\mathbf{q}$ in the reference view a putative stereo disparity $d = q'_1 - q_1$ given by

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

The equation above specifies a plane surface in the 3D space parametrized by (q_1, q_2, d). Thus, the matching hypothesis implicitly defined by $\mathbf{\Pi}$ (Equation 2) correspond to a plane $\mathbf{\Gamma}$ in the DSI domain, with homogeneous representation

$$\mathbf{\Gamma} \sim \begin{pmatrix} \frac{bn_1}{h} \\ \frac{bn_2}{h} \\ -1 \\ \frac{bn_3}{h} \end{pmatrix}. \tag{9}$$

3.5 Sweeping the scene by a pencil of vertical virtual planes bisecting the baseline

As stated previously, dense stereo matching with SymStereo requires using multiple virtual cut planes $\mathbf{\Pi}_i$ such that the corresponding planes $\mathbf{\Gamma}_i$ completely sweep the DSI domain. Let's assume that the planes $\mathbf{\Pi}_i$ belong to a vertical pencil with the axis intersecting the baseline in its middle point. In this case, the homogeneous representation of each plane is given by

$$\mathbf{\Pi}_i \sim \begin{pmatrix} 1 \\ 0 \\ -2\tan(\theta_i) \\ \frac{b}{2} \end{pmatrix},$$

with θ_i denoting the rotation angle around the vertical axis, and the plane homography of Equation 2 becomes

$$\mathsf{H}_i \sim \begin{pmatrix} -1 & 0 & 2\tan(\theta_i) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} .$$

Consider now that the image points $\mathbf{q}$ and $\mathbf{q}'$ are expressed in pixel coordinates, and that both cameras have the same intrinsic parameters

$$\mathsf{K} \sim \begin{pmatrix} f & 0 & c_1 \\ 0 & f & c_2 \\ 0 & 0 & 1 \end{pmatrix} .$$

The homography mapping $\mathbf{q} \sim \mathsf{K}\,\mathsf{H}_i\,\mathsf{K}^{-1}\,\mathbf{q}'$ defines a possible pixel association between images that can be written as

$$q_1 = \underbrace{2\,c_1 - q_1'}_{flip} + \lambda_i \,, \tag{10}$$

with

$$\lambda_i = 2\,f\,\tan(\theta_i) \,.$$

Moreover, and from the discussion of Section 3.4, each virtual cut plane $\mathbf{\Pi}_i$ corresponds to a plane $\mathbf{\Gamma}_i$ in the DSI domain with homogeneous coordinates

$$\mathbf{\Gamma}_i \sim \begin{pmatrix} 2 \\ 0 \\ -1 \\ -2\,c_1 - \lambda_i \end{pmatrix} . \tag{11}$$

Two important conclusions can be drawn from the analysis above. The first is that the range of disparities in the DSI domain is fully covered by a set of planes $\mathbf{\Gamma}_i$ such that the parameters λ_i take successive integer values. This enables to choose the angles θ_i that define a suitable set of virtual planes $\mathbf{\Pi}_i$ in the 3D scene space. The second is that the homography mapping of Equation 10 considerably simplifies the rendering of images $\widehat{\mathtt{I}}_i$ required for generating the symmetries and anti-symmetries (see Fig. 3). The warping can be efficiently achieved by flipping the original image $\mathtt{I}'$ around the vertical axis passing through the principal point, followed by shifting the result by an integer amount λ_i along the horizontal image direction. Henceforth, and since the use of a vertical pencil of planes $\mathbf{\Pi}_i$ is specially convenient for sweeping the scene in rectified stereo, the article will only address this particular configuration.

4 Measuring Local Symmetry and Anti-Symmetry

As shown in Fig. 3, the objective of SymStereo is to associate pixels across views by jointly using symmetry and anti-symmetry measurements. This section discusses techniques for quantifying local signal symmetry and anti-symmetry at every image pixel location of $\mathtt{I}^S$ and $\mathtt{I}^A$. We describe three alternative metrics: the *SymBT* that adapts the famous BT matching cost for measuring signal asymmetry instead of dissimilarity [7]; the *SymCen* that is a non-parametric symmetry metric inspired in the Census transform [36]; and the *logN* that has been originally proposed by Kovesi in [19] and uses a bank of N log-Gabor wavelets for evaluating local symmetry. Please note that the detection of symmetry in images has been extensively studied in the past, with [21] constituting an excellent survey of existing techniques. However, these methods typically concern perceptual symmetry and target tasks like detecting symmetric objects in images, which is substantially different from our objective of quantifying low-level signal symmetry in different pixel locations.

4.1 The SymBT metric

Consider a pair of corresponding epipolar lines in the stereo images $\mathtt{I}$ and $\mathtt{I}'$, and let d be a putative disparity value that associates pixel q_1 in $\mathtt{I}$ with pixel $q_1 - d$ in $\mathtt{I}'$. The matching likelihood can be inferred by measuring the dissimilarity between $\mathtt{I}(q_1)$ and $\mathtt{I}'(q_1 - d)$. In order to avoid sampling issues, Birchfield and Tomasi (BT) suggest to compare the intensity value $\mathtt{I}(q_1)$ in the reference view against a brightness interval $[m',\, M']$ around the putative image correspondence $\mathtt{I}'(q_1 - d)$ in the second view [7]. This is illustrated in Fig. 6(a), where the boundaries of the intensity range are

$$m' = \min\big(\mathtt{I}'(q_1 - d);\ \mathtt{I}'_-;\ \mathtt{I}'_+\big)$$
$$M' = \max\big(\mathtt{I}'(q_1 - d);\ \mathtt{I}'_-;\ \mathtt{I}'_+\big) \,,$$

with $\mathtt{I}'_-$ and $\mathtt{I}'_+$ being interpolated brightness values at the sub-pixel locations around $q_1 - d$. The dissimilarity between $\mathtt{I}(q_1)$ and $\mathtt{I}'(q_1 - d)$ is quantified by

$$C = \max\big(0;\ \mathtt{I}(q_1) - M';\ m' - \mathtt{I}(q_1)\big).$$

Considering now that $\mathtt{I}'$ is the reference view, it comes in a similar manner that

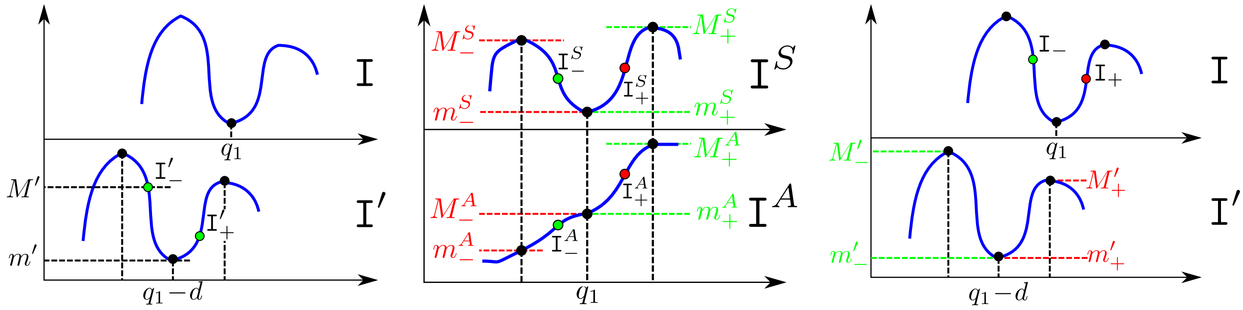
$$C' = \max\big(0;\ \mathtt{I}'(q_1 - d) - M;\ m - \mathtt{I}'(q_1 - d)\big),$$

where

$$m = \min\big(\mathtt{I}(q_1);\ \mathtt{I}_-;\ \mathtt{I}_+\big)$$
$$M = \max\big(\mathtt{I}(q_1);\ \mathtt{I}_-;\ \mathtt{I}_+\big) \,,$$

The final BT score handles the two views symmetrically and is given by

$$C_{BT}(q_1,\, d) = \min\big(C;\ C'\big)$$



(a) BT (b) SymBT (c) Efficient SymBT

Fig. 6 The SymBT metric: In (a) the standard BT cost compares value of pixel q_1 in the reference view against the intensity range $[m', M']$ around the putative match $q_1 - d$. The scheme (b) illustrates how SymBT quantifies the symmetry and anti-symmetry along the epipolar lines of $\mathtt{I}^S$ and $\mathtt{I}^A$. Given a particular pixel location q_1, the idea is to use the BT metric to compare the interpolated intensity value on one side against the intensity interval on the other side. Finally (c) shows how the SymBT metric can be efficiently implemented without requiring the explicit rendering of the image signals $\mathtt{I}^S$ and $\mathtt{I}^A$.

4.1.1 Modifying BT to measure asymmetry

Inspired by the BT cost function, we can define a metric for measuring asymmetry along the epipolar lines of the image signal $\mathtt{I}^S$ that is invariant to sampling issues. Let $\mathtt{I}^S_-$ and $\mathtt{I}^S_+$ be interpolated image values in the neighborhood of a particular pixel location q_1 in $\mathtt{I}^S$ (see Fig. 6(b)). The 1-D image signal symmetry can be evaluated by verifying if the sub-pixel image value in one side of q_1 is within the brightness interval in the opposite side. Thus, we propose to quantify the asymmetry of the image signal $\mathtt{I}^S$ about the pixel location q_1 by

$$D^S_{BT} = \max\left(0, \mathtt{I}^S_- - M^S_+;\ m^S_+ - \mathtt{I}^S_-;\ \mathtt{I}^S_+ - M^S_-;\ m^S_- - \mathtt{I}^S_+\right),$$

with

$$\begin{aligned} m^S_\pm &= \min\left(\mathtt{I}^S(q_1);\ \mathtt{I}^S(q_1 \pm 1)\right) \\ M^S_\pm &= \max\left(\mathtt{I}^S(q_1);\ \mathtt{I}^S(q_1 \pm 1)\right). \end{aligned}$$

A similar approach can be used for scoring the anti-symmetry of the image signal $\mathtt{I}^A$ at particular pixel locations. Consider the scheme in the bottom of Fig. 6(b), where $\mathtt{I}^A_-$, $\mathtt{I}^A_+$ are the interpolated image values at sub-pixel locations, and $[m^A_-,\ M^A_-]$, $[m^A_+,\ M^A_+]$ are the brightness intervals defined above. It is easy to understand that, if the image signal is anti-symmetric about q_1, then the following must hold:

$$\begin{aligned} &\mathtt{I}^A(q_1) + (\mathtt{I}^A(q_1) - \mathtt{I}^A_-) \in [m^A_+,\ M^A_+] \\ &\mathtt{I}^A(q_1) + (\mathtt{I}^A(q_1) - \mathtt{I}^A_+) \in [m^A_-,\ M^A_-]. \end{aligned}$$

Thus, we can modify the asymmetry score defined above for quantifying lack of signal anti-symmetry about q_1

$$\begin{aligned} D^A_{BT} = \max\big(0;\ &2\mathtt{I}^A(q_1) - \mathtt{I}^A_- - M^A_+;\ m^A_+ - 2\mathtt{I}^A(q_1) + \mathtt{I}^A_-; \\ &\ldots 2\mathtt{I}^A(q_1) - \mathtt{I}^A_+ - M^A_-;\ m^A_- - 2\mathtt{I}^A(q_1) + \mathtt{I}^A_+\big). \end{aligned}$$

Finally, the SymBT score for finding pixel locations that are simultaneously symmetric in $\mathtt{I}^S$ and anti-symmetric in $\mathtt{I}^A$ is defined as:

$$D_{BT}(q_1) = \max\left(D^S_{BT};\ \mathsf{D}^A_{BT}\right). \tag{12}$$

4.1.2 Efficient Implementation

The SymBT metric described in the previous section has the inconvenient of requiring the explicit rendering of the image signals $\mathtt{I}^S$ and $\mathtt{I}^A$ for each considered virtual cut plane. As discussed in section 3.5, a particular choice of cut plane implicitly assigns points q_1 in the reference view $\mathtt{I}$, to points $q_1 - d$ in the secondary view $\mathtt{I}'$. It is now shown how to compute the SymBT score for a particular matching hypothesis $(q_1,\ d)$ without having to explicitly render the image signals $\mathtt{I}^S$ and $\mathtt{I}^A$. Let's consider the scheme of Fig. 6(c) where $\mathtt{I}_-$, $\mathtt{I}_+$ are interpolated image values in the neighborhood of the pixel location q_1 in the reference view $\mathtt{I}$, and $[m'_-,\ M'_-]$, $[m'_+,\ M'_+]$ are the brightness intervals on the sides of the putative correspondence $q_1 - d$ in the secondary view $\mathtt{I}'$. The metric S evaluates till which extent $\mathtt{I}_-$ and $\mathtt{I}_+$ are within the ranges $[m'_-,\ M'_-]$ and $[m'_+,\ M'_+]$, respectively.

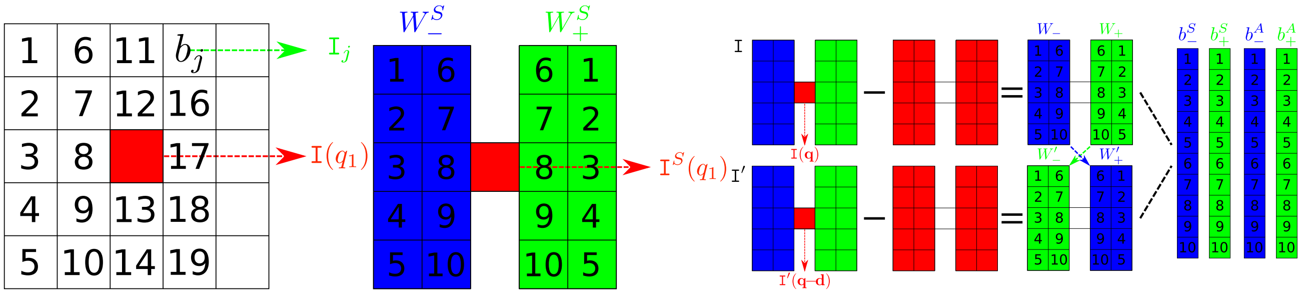
$$\begin{aligned} S_- &= \max\left(0, \mathtt{I}_- - M'_-;\ m'_- - \mathtt{I}_-\right) \\ S_+ &= \max\left(0, \mathtt{I}_+ - M'_+;\ m'_+ - \mathtt{I}_+\right) \\ S\ &= S_- + S_+. \end{aligned}$$

Considering now that $\mathtt{I}'$ is the reference view, it comes in a similar manner that

$$\begin{aligned} S'_- &= \max\left(0, \mathtt{I}'_- - M_-;\ m_- - \mathtt{I}'_-\right) \\ S'_+ &= \max\left(0, \mathtt{I}'_+ - M_+;\ m_+ - \mathtt{I}'_+\right) \\ S'\ &= S'_- + S'_+. \end{aligned}$$

It can be shown that the score D_{BT} given in Equation 12 is well approximated by

$$S_{BT}(q_1, d) = \max(\mathsf{S}, \mathsf{S}'). \tag{13}$$



(a) Census trasnform (b) SymCen (c) Efficient SymCensus

Fig. 7 The SymCensus transform. In (a) the standard Census transform defines a bit string $\mathbf{b}$ for each image point q_1, with each bit b_j corresponding to a particular pixel in a local patch centered in q_1. In (b) SymCen is used to quantify the signal symmetry in $\mathtt{I}^S$ by comparing the regions W_-^S and W_+^S on both sides of q_1. In (c) the SymCen is implemented without requiring the explicit rendering of $\mathtt{I}^S$ and $\mathtt{I}^A$. The bit strings b_-^S, b_+^S, b_-^A and b_+^A are computed by performing simple operations over W_-, W_+, W'_- and W'_+.

This cost function avoids the rendering of $\mathtt{I}^S$ and $\mathtt{I}^A$ and has a computational complexity that is close to the original BT dissimilarity metric.

4.2 The SymCen metric

The Census transform is a non-parametric filter that analyzes the differences between image intensity values in a $m \times n$ neighborhood around the pixel of interest. For illustration purposes consider a 5×5 patch centered in a pixel location denoted by q_1, and let $\mathtt{I}_j$ be the image intensity values for the entries j in this patch ($j = 1, \ldots, 24$) as shown in Fig. 7(a)). The output of the Census transform is a string $\mathbf{b}$, with 24 bits, where each bit b_j is set as follows:

$$b_j = \begin{cases} 1 & \text{if } \mathtt{I}(q_1) > \mathtt{I}_j \\ 0 & \text{if } \mathtt{I}(q_1) \leq \mathtt{I}_j \end{cases} . \tag{14}$$

Considering that the pixel q_1 in image $\mathtt{I}$ corresponds to pixel $q_1 - d$ in image $\mathtt{I}'$, we build a second bit string $\mathbf{b}'$ encoding the intensity values around $q_1 - d$ and compute the Census dissimilarity as

$$C_C(q_1, d) = H(\mathbf{b}; \mathbf{b}'),$$

with H denoting the Hamming distance.

4.2.1 Modifying Census to measure dissymmetry

Fig. 7(b) shows how the Census transform can be used to quantify image symmetry instead of image dissimilarity. In this case the 5×5 neighborhood is divided into two 5×2 regions, W_-^S and W_+^S, that are respectively in the left and right sides of the pixel of interest. The intensity values of the two patches are encoded in the bit strings $\mathbf{b}_-^S$ and $\mathbf{b}_+^S$ using Equation 14, and a new bit string is computed which describes the symmetry of the image signal $\mathtt{I}^S$ about the pixel location q_1

$$\mathbf{b}^S = (\mathbf{b}_-^S == \mathbf{b}_+^S),$$

where $==$ is the bitwise equality operator. The anti-symmetry in image $\mathtt{I}^A$ can be encoded in a similar manner by

$$\mathbf{b}^A = (\mathbf{b}_-^A == \bar{\mathbf{b}}_+^A),$$

where $\mathbf{b}_-^A$ is the bit string of the left side region W_-^A, and $\bar{\mathbf{b}}_+^A$ is the binary complement of the bitstring of the right side patch W_+^A. The final SymCen score for the pixel q_1 is obtained by comparing corresponding symmetry and anti-symmetry bits $\mathbf{b}_j^S$ and $\mathbf{b}_j^A$, and then summing all the bit responses:

$$S_C(q_1) = \sum_j \mathbf{b}_j^S \& \mathbf{b}_j^A, \tag{15}$$

where & is the bitwise *and* operator. Remark that different from the Census metric, larger values of the SymCen cost correspond to higher matching likelihood.

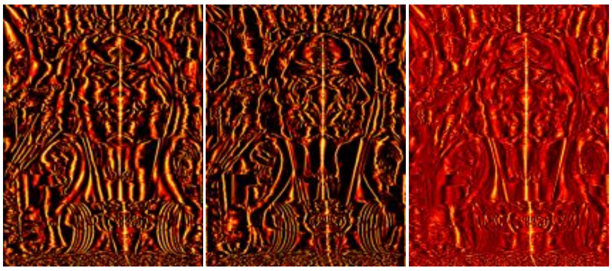
4.2.2 Efficient Implementation

The bit strings b_-^S, b_+^S, b_-^A, and b_+^A, that are required for evaluating the SymCensus cost of Equation 15, can be directly computed from the stereo pair $\mathtt{I}$ and $\mathtt{I}'$ as shown in Fig. 7(c). Let W_- and W_+ be the patches on both sides of pixel q_1 in the reference view $\mathtt{I}$, and W'_- and W'_+ be the patches around the putative correspondence $q_1 - d$ in the secondary view $\mathtt{I}'$. Subtract $\mathtt{I}(q_1)$ to the intensity values in regions W_- and W_+. Repeat the procedure in the secondary view using $\mathtt{I}'(q_1 - d)$. It can be proved that the bit strings for evaluating the score S_C can be determined as follows:

$$\begin{aligned} \mathbf{b}_-^S &= T(W_-; -W'_-) \\ \mathbf{b}_+^S &= T(W_+; -W'_+) \\ \mathbf{b}_-^A &= T(W_-; W'_+) \\ \mathbf{b}_+^A &= T(W_+; W'_-) \end{aligned}$$

with T being an operator that compares the intensity values of corresponding pixels in two patches W and W', generating a bit string with the j^{th} bit being given by

$$T_j(W; W') = \begin{cases} 1 & \text{if } \mathtt{I}_j > \mathtt{I}'_j \\ 0 & \text{if } \mathtt{I}_j \leq \mathtt{I}'_j \end{cases} .$$



(a) $\mathtt{E}^S$ (b) $\mathtt{E}^A$ (c) $\mathtt{E} = \mathtt{E}^S \cdot \mathtt{E}^A$

Fig. 8 The logN metric: (a) is the symmetry energy $\mathtt{E}^S$ of the image signal $\mathtt{I}^S$, while (b) is the anti-symmetry energy $\mathtt{E}^A$ of image $\mathtt{I}^A$. The final joint energy $\mathtt{E}$ in (c) is obtained by pixel-wise multiplication of $\mathtt{E}^S$ and $\mathtt{E}^A$.

This alternative scheme for computing the SymCensus score has the obvious advantage of avoiding the explicit rendering of image signals $\mathtt{I}^S$ and $\mathtt{I}^A$, which substantially decreases the computational complexity.

4.3 The logN metric

Kovesi shows that an intensity distribution that is symmetric about a particular pixel location gives rise to specific phase patterns in the Fourier series of the image signal [19]. Thus, he proposes to detect symmetry and anti-symmetry based on frequency information obtained using a bank of log-Gabor filters. This section describes the joint application of Kovesi's algorithms with the SymStereo framework, leading to a new stereo matching cost that is referred as *logN*, with N standing for the number of wavelet scales that is considered for the signal analysis.

Since the log-Gabor wavelets are analytical signals, the image filtering must be carried in the spectral domain. Let $\mathcal{G}_k$, with $k = 1, \ldots N$, be the frequency response of the pre-selected wavelet scales, and $\mathcal{I}^S$ be the spectrum of a generic epipolar line $\mathtt{I}^S(q_1)$ in the symmetry image (see Fig. 3(d)). The filtering result is the following 1D complex signal

$$s_k^S(q_1) + \mathbf{i}\, a_k^S(q_1) = F^{-1}(\mathcal{I}^S \cdot \mathcal{G}_k)\,, \tag{16}$$

with F denoting the Fourier transform and $\mathbf{i}^2 = -1$. It can be shown that, if the image is symmetric about the pixel location q_1, then the real component $\mathtt{s}_k^S$ typically takes high values, while the imaginary component $\mathtt{a}_k^S$ takes small values [19]. Therefore, and given the N wavelet scale responses, we can establish the following energy of symmetry:

$$\mathtt{E}^S(q_1) = \frac{\sum_{k=1}^{N} \mid s_k^S(q_1) \mid - \mid a_k^S(q_1) \mid}{\sum_k \sqrt{\left(s_k^S(q_1)\right)^2 + \left(a_k^S(q_1)\right)^2}}, \tag{17}$$

where the normalization by the sum of the magnitudes provides invariance to changes in illumination [19]. Fig. 8(a) shows the result of stacking the lines $\mathtt{E}^S(q_1)$ arising from each row of image $\mathtt{I}^S$ of Fig. 3(d). It can be observed that the highlights correspond to pixel locations where the image signals is symmetric along the horizontal direction.

Considering now the anti-symmetric image $\mathtt{I}^A$ of Fig. 3(e), we can use the different wavelet scales and compute

$$s_k^A(q_1) + \mathbf{i}\, a_k^A(q_1) = F^{-1}(\mathcal{I}^A \cdot \mathcal{G}_k)\,. \tag{18}$$

Apply a similar approach for deriving an energy of anti-symmetry yields

$$\mathtt{E}^A(q_1) = \frac{\sum_{k=1}^{N} \mid a_k^A(q_1) \mid - \mid s_k^A(q_1) \mid}{\sum_k \sqrt{\left(s_k^A(q_1)\right)^2 + \left(a_k^A(q_1)\right)^2}}\,. \tag{19}$$

The resulting energy $\mathtt{E}^A$ is depicted in Fig. 8(b), with the locations of image anti-symmetry being clearly emphasized.

Both $\mathtt{E}^S$ and $\mathtt{E}^A$ have several local maxima along the horizontal lines, which preclude a straightforward detection of the image of the profile cut $\mathcal{C}$, that is overlaid in Fig. 3(d) and Fig. 3(e). Since points in $\mathcal{C}$ must be simultaneously local maxima in $\mathtt{E}^S$ and $\mathtt{E}^A$, the pixel-wise multiplication of the two energies enables to discard most spurious detections. Thus, we consider the following joint energy $\mathtt{E}$

$$\mathtt{E} = \mathtt{E}^S \cdot \mathtt{E}^A \tag{20}$$

where the image of the contour $\mathcal{C}$ is clearly distinguishable as shown in Fig. 8(c)

4.3.1 *Efficient implementation*

The joint energy $\mathtt{E}$ is computed from the images $\mathtt{I}^S$ and $\mathtt{I}^A$, which are rendered for a particular virtual cut plane $\mathbf{\Pi}$. As discussed in section 3.5, each plane $\mathbf{\Pi}_i$ in the scene gives rise to a plane $\mathbf{\Gamma}_i$ in the DSI that is function of an integer parameter λ_i (see Equation 11). As discussed in this section, the energy $\mathtt{E}$ can be computed without explicitly rendering the image signals $\mathtt{I}^S$ and $\mathtt{I}^A$, and the evaluation of *logN* across the entire DSI domain can be carried in a very efficient manner.

Let $\mathtt{I}^S(q_1)$ be the 1D signal arising from a generic epipolar line in the symmetry image $\mathtt{I}^S$. If $\mathtt{I}(q_1)$ and $\mathtt{I}'(q_1)$ are the corresponding lines in the rectified stereo pair, then it follows from Equation 10 that:

$$\begin{aligned} \mathtt{I}^S(q_1) &= \mathtt{I}(q_1) + \hat{\mathtt{I}}(q_1) \\ &= \mathtt{I}(q_1) + \mathtt{I}'_f(q_1 - \lambda)\,, \end{aligned}$$

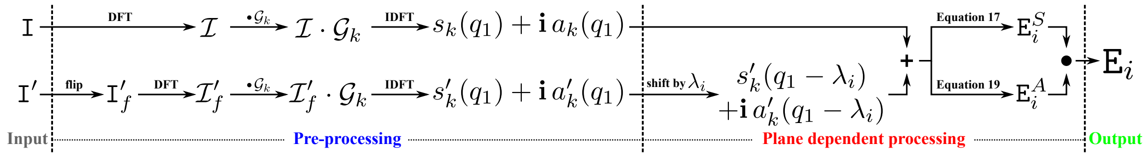


Fig. 9 Efficient implementation of the *logN* stereo matching cost. In a first step the rectified stereo pair is filtered by the considered wavelet scales $\mathcal{G}_k$ in order to obtain the left and right complex signals $s_k(q_1)+\mathbf{i}\,a_k(q_1)$ and $s'_k(q_1)+\mathbf{i}\,a'_k(q_1)$ with $k=1,2\ldots N$. In a second stage, and for each scale k, the right-side signal is shifted by an amount λ_i, which depends on the virtual cut plane $\mathbf{\Pi}_i$, and the result is added and subtracted to the left-side signal. The operation provides the input coefficients for computing the symmetry and anti-symmetry energies of equations 17 and 19, ultimately leading to the energy $\mathtt{E}_i$.

where λ is a shift amount depending on the choice of the virtual plane $\mathbf{\Pi}$, and $\mathtt{I}'_f$ is a horizontally flipped version of the right side image

$$\mathtt{I}'_f(q_1) \;=\; \mathtt{I}'(2c_1-q_1)\,.$$

From the reasoning above, and exploring the linear properties of the Fourier transform, it comes that Equation 16 can be re-written as:

$$\begin{aligned} s_k^S(q_1)+\mathbf{i}\,a_k^S(q_1) &= (s_k(q_1)+s'_k(q_1-\lambda)) \\ &\quad+\mathbf{i}\;(a_k(q_1)+a'_k(q_1-\lambda))\,, \end{aligned}$$

with

$$\begin{cases} s_k(q_1)+\mathbf{i}\,a_k(q_1) \;=\; F^{-1}(\mathcal{I}\cdot\mathcal{G}_k) \\ s'_k(q_1)+\mathbf{i}\,a'_k(q_1) \;=\; F^{-1}(\mathcal{I}'_f\cdot\mathcal{G}_k) \end{cases},$$

where $\mathcal{I}$ and $\mathcal{I}'_f$ stand for the Fourier transform of $\mathtt{I}(q_1)$ and $\mathtt{I}'_f(q_1)$, respectively. The response of Equation 18 for the anti-symmetric image signal $\mathtt{I}^A(q_1)$ can be computed in a similar manner by

$$\begin{aligned} s_k^A(q_1)+\mathbf{i}\,a_k^A(q_1) &= (s_k(q_1)-s'_k(q_1-\lambda)) \\ &\quad+\mathbf{i}\;(a_k(q_1)-a'_k(q_1-\lambda))\;. \end{aligned}$$

Fig. 9 is a schematic of the computation pipeline for obtaining the energy $\mathtt{E}_i$ for a particular choice $\mathbf{\Pi}_i$ of virtual cut plane. The new formulation not only avoids the explicit rendering of the symmetric and anti-symmetric image signals, but also enables to efficiently evaluate the entire DSI by simply varying the shifting amount λ_i with $i=1,2\ldots M$. Moreover, and despite of not done in this article, the computations can be easily parallelized using GPGPU techniques.

4.3.2 Selection of wavelet scales

The choice of the log-Gabor wavelets for filtering the input images has a strong influence in the final stereo results. Despite of the fact that log-Gabor filters are analytical signals with no real representation in the space domain, the scheme of Fig. 10 tries to provide an intuition about how the wavelet parameters relate with the space-frequency response of the filter. The horizontal axis refers to the space extent or support of the filter kernel, while the vertical axis concerns the frequency components of the image signal to which $\mathcal{G}_k$ responds. If the image region is very textured, then it is advisable to operate in the top-left corner of the $(\omega,\,\sigma)$ plane, and choose filters with high-frequency response and small space extent. On the other hand, if the image region is textureless, then we must consider wavelets that respond to low-frequency components, but that have a larger support which tends to diminish the pixel accuracy of the analysis.

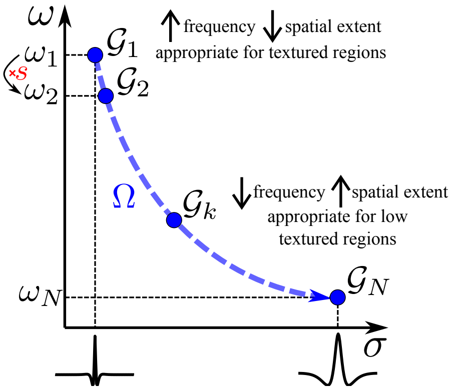


Fig. 10 (Qualitative) space-frequency behavior of the log-Gabor wavelets $\mathcal{G}_k$. The horizontal axis refers to the spatial support σ of the filter kernel, while the vertical axis concerns the response frequency ω.

As discussed in [20], the bank of log-Gabor wavelets $\mathcal{G}_k$ is usually parametrized by the shape-factor $\mathit{\Omega}$, the center frequency of the mother wavelet ω_1, the scaling step s, and the total number N of wavelets. The shape-factor $\mathit{\Omega}$ can be related with the filter bandwidth, and defines a contour in the $(\omega,\,\sigma)$ domain containing the wavelets that can be selected (see Fig. 10). The center frequency ω_1, together with the shape factor $\mathit{\Omega}$, defines uniquely the first wavelet scale $\mathcal{G}_1$. The scaling step s sets the distance between the center frequencies of successive wavelet scales k and $k+1$ along the contour. In this article we have manually set $\mathit{\Omega}=0.55$, $\omega_1=0.25$, and $s=1.05$, and kept these values constant throughout the entire set of experiments. The only parameter that is allowed to vary is the number of scales N that controls the ability of obtaining response in low textured image regions by using filters with a larger spatial support.

5 Experiments in Dense Stereo Reconstruction

Until now we proposed three matching costs - SymBT, SymCen, and logN - that use symmetry instead of photo-consistency for accomplishing pixel data association. The described approach is new and original, but an important question remains: what are the effective advantages with respect to existing stereo cost functions? This section tries to answer the question by running an extensive set of experiments in dense stereo reconstruction. The results enable to characterize the performance of symmetry-based stereo and empirically show the advantages with respect to state-of-the-art matching costs. The conclusions are further confirmed in Sections 6 and 7 that run additional tests in SRF and wide-baseline stereo.

5.1 Methodology and tuning of parameters

Since the stereo literature is vast, it is virtually impossible to compare SymStereo against every possible method and approach. Thus, and in order to assure a rigorous and conclusive study, the evaluation herein presented follows the methodology and takes into account the results of the recent benchmark work of Hirschmuller and Scharstein [17]. We compare the three symmetry-based matching costs against the cost functions that, for one reason or the other, were considered to be top-performers in [17]. These stereo cost functions are:

1. *Birchfield-Tomasi (BT):* It quantifies pixel dissimilarity by comparing 1-Dimensional (1D) neighborhoods defined along the epipolar lines [7]. According to [17], the BT metric combined with Bilateral Background Subtraction (BBS) [1], provides the best matching results among pixel-wise parametric costs.
2. *Zero-mean Normalized Cross-Correlation (ZNCC):* It is a broadly used cost function, that considers a 2-Dimensional (2D) support region for quantifying photo-similarity, and proved to be the a top-performer among window-based parametric matching costs.
3. *Census Filter:* It is a window-based non-parametric cost function [36], that consistently proved to be the top similarity measure for dense disparity estimation.

The evaluation is carried using stereo pairs with ground truth disparity that include challenging situations, e.g. slanted surfaces, low and repetitive textures, depth discontinuities. Like in [17], most experiments in this section are performed using the Middlebury dataset [27, 26, 17] but, while they run the benchmarking in 6 image pairs, we consider a set of 15 examples that covers a wider range of situations (see Fig. 13). For each cost function under analysis, we build the DSI of the different image pairs, estimate the corresponding disparity maps using a particular stereo method, and score the estimation result by counting the number of pixel locations in non-occluded regions with a disparity error greater than one. The matching costs under benchmark are ranked by averaging the error score across all stereo pairs in the test set. Since the focus is in evaluating the performance of matching costs, the disparity estimation must be carried by the exact same stereo method for all costs in order to assure fair comparison. As in [17], we present results using three distinct possibilities:

1. *Local Aggregation :* The DSI is aggregated by summing the costs over a window and each image pixel is assigned with the disparity value that has the lowest cost.
2. *Semi-Global Matching (SGM):* It is an approach in-between local and global matching that minimizes a 2D energy by solving multiple 1D minimization problems [16].
3. *Graph-Cut (GC):* The disparity map is estimated by global minimization of an energy function defined in the DSI using graph-cuts [9, 18, 8, 32].

GC and SGM are formulated in the standard manner, and post-processing steps, e.g. left-right consistency check or sub-pixel interpolation, are not considered.

It can be argued that using local aggregation is better suited for comparing different matching costs than using SGM or GC. It is a fact that global and semi-global methods, being more sophisticated minimization techniques, can eventually hide issues and weaknesses in the stereo cost function. Although we agree that local aggregation provides the most relevant benchmarking information, this section also presents the scores obtained with SGM and GC for the sake of completeness and to assure full compliance with the methodology and results described in [17].

It can also be argued that choosing adaptive-weight aggregation [35], instead of basic window aggregation, is likely to improve the disparity estimation in image regions that are close to discontinuities or lack strong texture. This is true but it is important to keep in mind that such improvements are transversal to all matching costs and do not necessarily change the relative disparity scores. Moreover, and as stated above, advanced stereo methods are more likely to overcome issues that are inherent to the considered cost function, which can bias the results of the benchmark.

Finally, for every matching cost under study, the computation of the DSI is carried in C++ assuming input images with size 460×370 and disparity range of 64 pixels. The C++ implementations are straightforward, do not involve parallel processing, and only use the standard code optimizations described in stereo literature. This is proved by the fact that BT, ZNCC, and Census, present execution times that are consistent with what has been reported by other authors. All the runtimes presented in this article were

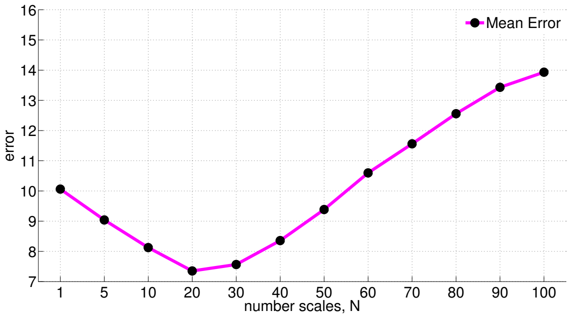


Fig. 11 Tuning the number of wavelets scales N for dense stereo using the standard Middleburry dataset. The figure plots the average error in disparity estimation using local aggregation when N increases.

Table 1 Summary of the parameters used in experiments throughout the article in Dense Stereo (DS), Stereo Rangefinder (SRF), and wide-baseline (WB) images.

	DS (Sec. 5)	**SRF** (Sec. 6)	**DS-WB** (Sec. 7)	**SRF-WB** (Sec. 7)
ZNCC<b>M</b>	9×9	15×15	7×7	9×9
Census<b>H</b>	9×7	9×19	9×19	9×23
SymCen<b>H</b>	9×7	9×19	9×9	9×23
log<b>N</b>	20	40	50	70

measured on the same machine in order to assure a fair comparison between competing matching costs.

5.1.1 Tuning of parameters

Like in [17], all parameters are manually tuned using the standard Middleburry dataset [27], that comprises the images *Tsukuba*, *Venus*, *Teddy*, and *Cones*. These pairs are not considered latter in the benchmark to avoid bias effects. Whenever applicable, we use the optimal values reported in [17], meaning that for the dense stereo experiments the local aggregation window is 9×9, the ZNCC window is 9×9, and the Census window is 9×7. In order to allow a direct comparison between Census and SymCen we also consider a window of 9×7 for the second. As shown in Fig. 11, the number of wavelet scales to be used with logN is set to $N = 20$. As expected, increasing N does not necessarily improve the performance because low frequency wavelets have wider space support that decreases the accuracy of the disparity estimation (see Fig. 10 in section 4.3.2). For the case of BT and SymBT, we always apply bilateral filtering and consider a 3 pixel neighborhood. Table 1 summarizes the choice of parameters for this and the following sections. For the latter experiments in SRF and wide-baseline stereo, we will re-tune the window-size of ZNCC, the horizontal window-size of Census and SymCen, and the number of scales of logN.

After tuning the cost functions assuming local aggregation, we move to the setting of the parameters of SGM and GC that will be used with each matching cost. The tuning is carried by selecting the parameter values that provide the smallest percentage of disparity errors in the images of the standard dataset. These errors are plotted in Fig. 12 where it can be observed that the results for BT, ZNCC, and Census are close to the ones reported in [17][2]. This indicates that the choice of parameters is optimal, and that our symmetry-based matching costs will be effectively compared against top-performing metrics. We do not provide scores for the case of ZNCC combined with SGM or GC because ZNCC is by definition a local method and, as also referred in [17], the experiments showed that the global and semi-global minimizations often lead to poorer results that the ones obtained with simple aggregation.

5.2 Tests in Middleburry Images

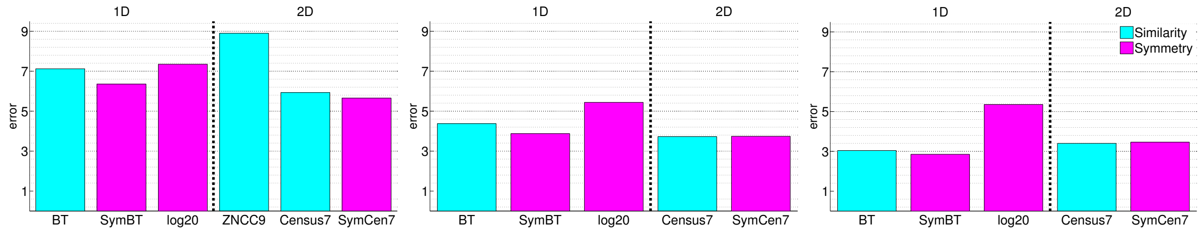
The 6 matching costs are now compared by analyzing the errors in dense disparity estimation in the Middlebury images of Fig. 13. Fig. 14 shows the mean of the percentage of pixels with incorrect disparity label for a particular combination of matching cost and stereo method. The first observation is that pixel-based 1D metrics tend to perform worse than window-based 2D costs. This is to expect because most surfaces in the Middlebury dataset have moderate or no slant. More important is the fact that the symmetry-based metrics, SymBT and SymCen, consistently beat their similarity-based counterparts, BT and Census. Thus, the experimental evidence clearly suggests that the symmetry cues are more effective than the standard photo-consistency measurements for matching pixels across views.

It can also be observed that log20 has an erratic behavior ranking differently according to the stereo method that is considered. For the case of local aggregation it is the most inaccurate metric among the 1D matching costs, although it performs significantly better than ZNCC. Apparently the use of global minimization changes the ranking of relative performances, with log20 becoming respectively the best and second best pixel-based cost function when combined with SGM and GC. The reasons for this behavior require a more detailed analysis of the experimental data. For this purpose the input set is divided into two subsets:

1. *Set I:* It comprises the images with many objects and surface discontinuities (marked with yellow in Fig. 13).
2. *Set II:* It contains the images that are dominated by large sized surfaces that often present poor or repetitive texture (marked with green in Fig. 13).

The estimation in the two sets is analyzed using the criterion introduced in [23] that tests the ability of a matching cost to rank the matches according to their reliability.

[2] We obtain slightly worse results with SGM but, on the other hand, the results accomplished with GC are slightly better



(a) Local Aggregation (b) SGM (c) GC

Fig. 12 Result after tuning the parameters: the figure plots the percentage of errors in dense disparity estimation across the images of the standard Middlebury dataset.

After using local aggregation for the dense disparity labeling, the pixel locations are sorted in ascending order of cost, and a semi-dense disparity map is obtained by selecting the first $L\%$ pixels for which the matching confidence is higher. Fig. 15 shows the mean percentage of errors in the semi-dense disparity estimation for increasing values of L. Looking to the scores for $L = 100$, that correspond to the errors in the dense disparity map, it can be seen that all matching costs perform worse in Set II than in Set I, suggesting that the former dataset is a more challenging than the latter. It can also be observed that SymBT and SymCen behave equal or better than BT and Census, respectively, for all levels of completeness L. The most striking difference between the two plots is the fact that log20 has the second worse reliability performance in the images of Set I, but it is clearly the most accurate matching cost for a completeness up to $L = 85\%$ in Set II, only loosing the advantage in the disparity labeling of the last 15% of pixels with highest cost scores. It happens that these pixels are usually located close to discontinuities and/or occlusion regions, suggesting that log20 is very effective in estimating the disparity along the continuous surfaces with low or repetitive texture, but has more difficulty than other matching costs in handling the depth discontinuities. This can also explain the improvements of log20 in the ranking of relative performances that were observed in Fig. 14. Since the pixels in the continuous surfaces have lower cost values at the correct disparities, they have a stronger regularization effect during the SGM and GC minimizations that leverages the depth estimation close to the discontinuities.

Fig. 16 shows, for each stereo pair and matching cost, the error normalized by the mean error over all matching functions [17]. The objective of the plots is to provide a perspective about the relative performance of the different matching cost in a particular input image. The results show that log20 always compares well for the images of Set II confirming the hypothesis that, despite of being a 1D matching cost, it is specially effective in scenes dominated by large surfaces with low and/or repetitive texture. It can also be seen that SGM and GC boost the relative accuracy of log20 in Set II but not in Set I, which is in accordance with the interpretation that the improvements in the ranking of Fig. 14 are because of the low cost values at correct pixel disparities observed in Fig. 15(b).

Table 2 Runtime for evaluating the DSI assuming 375×450 images and a disparity range of 64 pixels.

Match. Cost	Time (ms)	Match. Cost	Time (ms)
BT (+BBS)	120 (+296)	*SymBT (+BBS)*	170 (+296)
Census7	160	*SymCen7*	185
ZNCC9	3200	*log20*	3900

Fig. 17 shows the disparity errors in the *Laundry* example. It is interesting to observe that SymBT and SymCen tend to outperform BT and Census in the continuous regions, while presenting similar performance close to discontinuities. In general the log20 is very accurate in the continuous surfaces, providing to be resilient to low and repetitive textures, but the error regions are considerably larger close to depth discontinuities and occlusions.

Table 2 summarizes the runtimes for evaluating the DSI of the *Teddy* stereo pair using the different matching functions. Since no special effort has been made in optimizing the codes, the times are merely indicative of the relative computational complexity. The slowest matching cost is log20 because it requires the computation of 2 FFTs and 40 IFFTs. However, and since the pipeline of Fig. 9 has a high degree of parallelism, the runtime can be easily decreased by a GPGPU implementation. In general the symmetry-based matching functions have higher computational complexity, but the magnitude of additional effort does not preclude the possibility of real-time dense disparity estimation, largely justifying the observed improvements in accuracy.

5.3 Tests in Oxford Corridor

Fig. 18 shows the percentage of disparity errors for the *Oxford Corridor* that is exhibited in the bottom-right corner of Fig. 13. The disparity estimation is carried by a WTA strategy after local aggregation of the DSI using a 9×9 window. The relative performance of the matching functions differs from the one observed in the equivalent experiment using

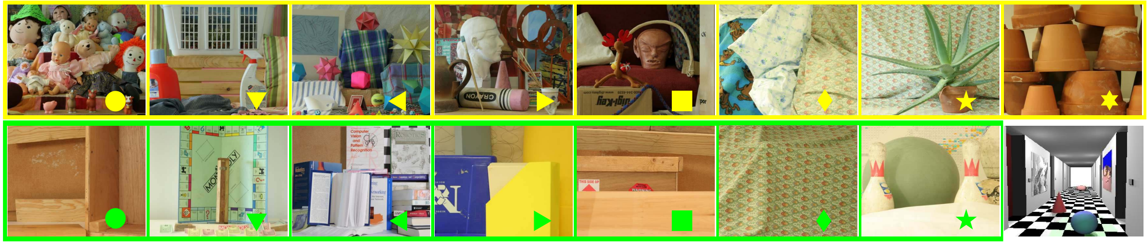
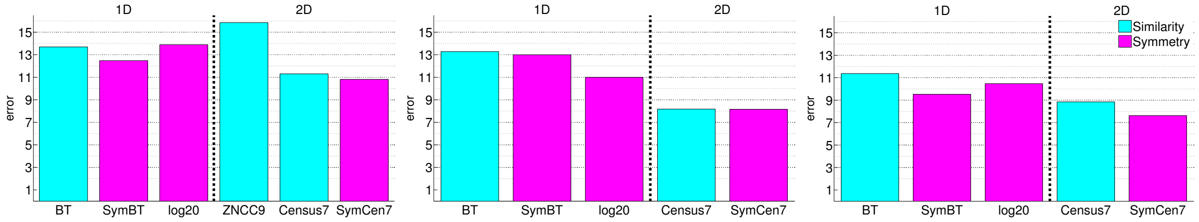
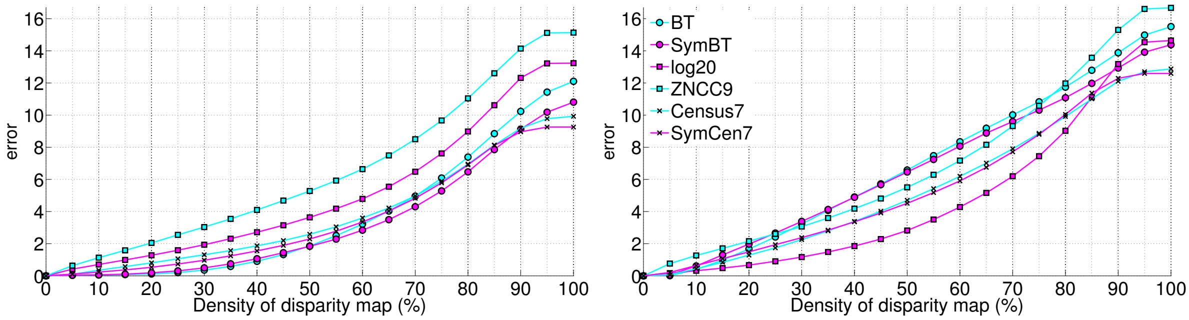


Fig. 13 The stereo pairs that are used as input for the experiments of Sections 5 and 6. The benchmark is carried in 15 images of the Middlebury dataset [26, 17]. The top row shows the Set I comprising frames with several objects and depth discontinuities. The bottom row exhibits the Set II consisting in scenes dominated by continuous surfaces with low or repetitive texture. The image in the bottom right corner refers to the Oxford Corridor that is used in Section 5.3 for evaluating the performance in case of strong surface slant.



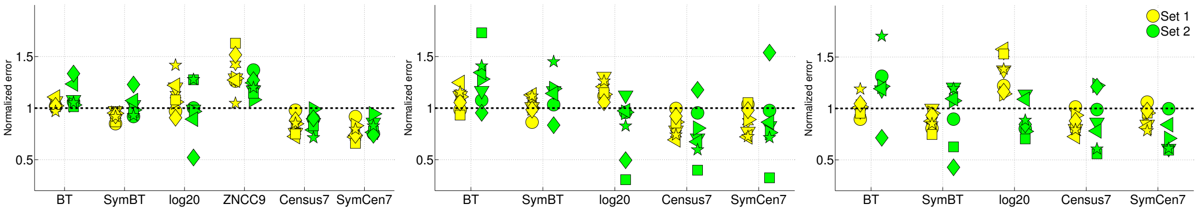
(a) Local Aggregation (b) SGM (c) GC

Fig. 14 Average percentage of disparity errors in the dense disparity maps of the 15 images of the Middlebury dataset (Set I + Set II).



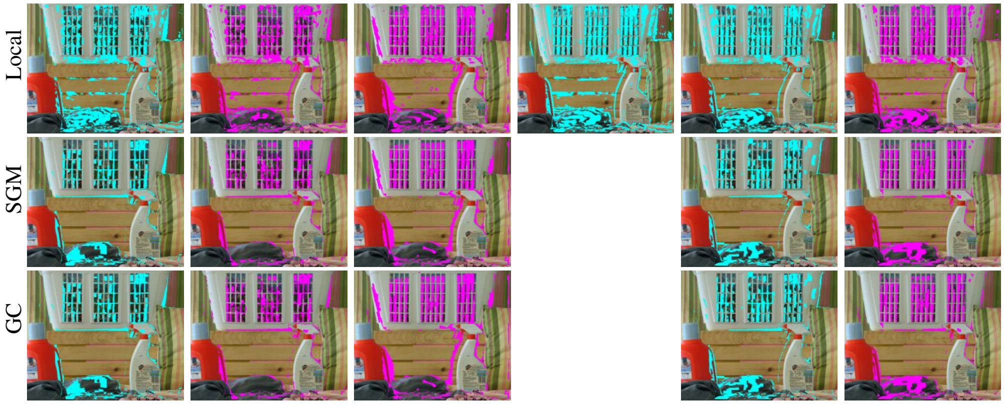
(a) Set I (b) Set II

Fig. 15 Average percentage of disparity errors in the semi-dense disparity maps of Set I (a) and Set II (b) obtained by selecting the first $L\%$ matches with lowest cost [23].



(a) Local Aggregation (b) SGM (c) GC

Fig. 16 The number of disparity errors for each input image normalized by the average number of errors across all matching costs [17].



(a) BT (b) SymBT (c) log20 (d) ZNCC9 (e) Census7 (f) SymCensus7

Fig. 17 Overlay of the disparity errors in the *Laundry* example for every possible combination of matching cost (columns) and stereo method (rows).

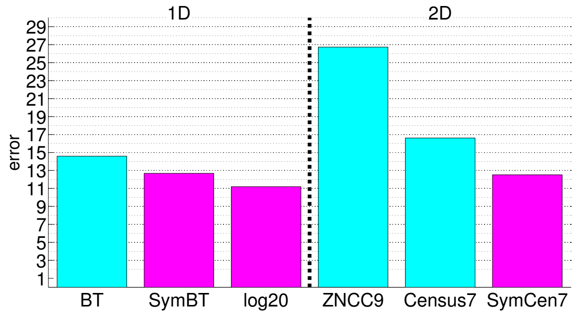


Fig. 18 Percentage of disparity errors in the dense disparity map of the *Oxford Corridor*. The estimation was carried after local aggregation with a 9×9 window.

.

the Middlebury dataset (see Fig. 14(a)). First, for the *Oxford Corridor* the 1D matching costs outperform the 2D functions because now the scene is dominated by highly slanted surfaces. Second, the differences in accuracy between symmetry and similarity-based matching functions are more striking in Fig. 18 than in Fig. 14(a). with the log20 being the top-performing metric. This is explained by the fact that most textures in the *Oxford Corridor* are either flat, e.g. the walls, or repetitive, e.g. the checkerboard pattern of the floor. Thus, the results of this experiment seem to confirm that the symmetry-based costs in general, and the logN metric in particular, are specially well suited for estimating the disparity in continuous regions with low or repetitive texture and high slant, clearly beating the similarity-based counterparts.

6 Experiments in Stereo Rangefinder (SRF)

Stereo Rangefinder (SRF) consists in using passive stereo for estimating depth along a virtual cut plane in order to reconstruct the contour $\mathcal{C}$ where the plane meets the scene. As discussed in [2], SRF enables a trade-off between runtime and 3D model resolution that does not interfere with depth accuracy, providing an effective way of probing into the 3D structure of the scene for applications like reconstruction of man-made environments [5, 3] and robot range-finding [4]. This section evaluates the performance of the 6 matching functions for the purpose of SRF. Henceforth, and due to space constraints, we will only present the disparity estimation results obtained using local aggregation.

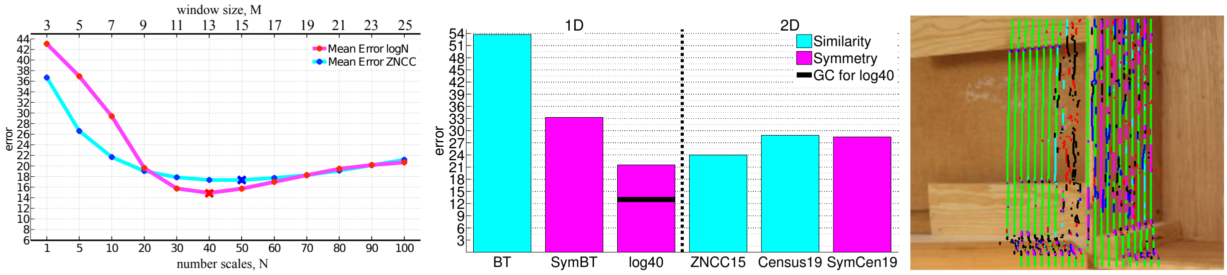
6.1 Methodology and tuning of parameters

From Section 3.5 it follows that a virtual cut plane $\mathbf{\Pi}_i$ going in-between the cameras corresponds to a plane $\mathbf{\Gamma}_i$ in the DSI domain. While dense stereo evaluates the matching function for the entire DSI, SRF only considers the disparity hypothesis corresponding to 3D points lying in $\mathbf{\Pi}_i$, meaning that the cost is exclusively evaluated along the plane $\mathbf{\Gamma}_i$ in the DSI. In our experiments, the scores in $\mathbf{\Gamma}_i$ are locally aggregated using a vertical 9×1 window (no horizontal aggregation), and a disparity label is assigned to each epipolar line using a WTA strategy. Since the winning labels must always occur in the pixel locations where the profile cut $\mathcal{C}$ is projected, the number of errors in SRF is determined by counting the winners that are more than 1 pixel apart from the ground truth image contour (see Fig. 3).

The performance of the 6 matching functions is benchmarked by averaging the results obtained in the 15 Middlebury images of Fig. 13. In each case the scene depth is independently estimated along 201 vertical cut planes $\mathbf{\Pi}_i$ with uniformly distributed rotation angles θ_i (see Section 3.5). The objective of using such a large number of cut planes is to cover a broad range of possible SRFsituations, with $\mathbf{\Pi}_i$ either intersecting the scene in a continuous surfaces or passing nearby a depth discontinuity. As in the dense stereo experiments, the parameters of the matching functions are manually tuned using the standard Middlebury dataset as input. Fig. 19(a) plots the average percentage of errors for logN and ZNCC in case of increasing number of scales and window-size, respectively. The choice of parameters is summarized in the second column of Table 1, where a comparison with dense stereo shows that SRF benefits from computing the matching costs across a wider pixel neighborhood. This is not surprising if we take into account that the larger image patches tend to compensate the fact that the aggregation is only carried in the 1D-vertical direction.

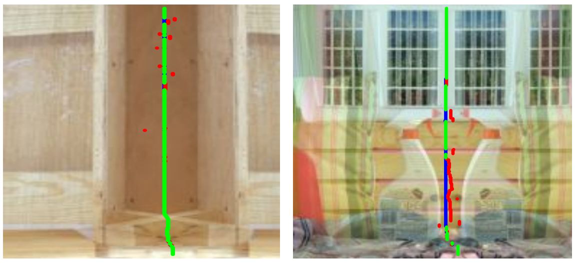
6.2 Tests in Middleburry Images

Fig. 19(b) shows the percentage of disparity errors averaged across the 15 image pairs of Fig. 13. Comparing with the dense stereo results of Fig. 14, it comes that the disparity estimation in SRF is less accurate for all matching functions. The higher percentage of errors is justified by the fact that SRF uses less information than dense stereo for the disparity labeling, because it only evaluates and aggregates the cost along a plane $\mathbf{\Gamma}_i$ in the DSI domain. The second observation is that symmetry-based matching costs still outperform their similarity-based counterparts, with SymBT and SymCen19 having less 20% and 1% of errors than BT and Census, respectively. The relative lower performance of the BT family is largely due to the fact that the scores are computed across a small 3-pixel neighborhood, which seems to be an insufficient image support for handling the lack of horizontal aggregation. Finally, ZNCC15 is the most accurate metric among the similarity-based matching functions, but it is beaten by log40 that presents 4% less errors. The figure also shows the accuracy of log40 when the local aggregation is replaced by global optimization using a standard GC formulation that enforces continuity in the mirroring contour.



(a) Manual tuning of parameters (b) Average percentage of disparity errors (c) log40 vs ZNCC15

Fig. 19 Benchmark of the cost functions for Stereo Rangefinder (SRF): (a) average percentage of errors in the standard Middlebury dataset for logN and ZNNC when the spatial support increases; (b) average percentage of disparity errors in the 15 Middlebury images of Fig. 13 for the 6 matching costs; (c) disparity errors in the *Wood1* example when using log40 and ZNCC15. The disparity labeling is independently carried for each virtual cut plane by a WTA approach after local aggregation using a 9×1 window. The overlay refers to the image of the mirroring contour where green is correct estimation of both (log40 and ZNCC15), black is wrong detection of both, magenta and blue means log40 is correct and ZNCC15 is wrong, respectively, whereas red and cyan means log40 is wrong and ZNCC15 is correct, respectively.



(a) Strength (b) Weakness

Fig. 20 Pros and cons of the logN symmetry-based matching function. Figures (a) and (b) show the symmetry images $\mathtt{I}^S$ for particular choices of $\mathbf{\Pi}_i$. The overlay refers to the image of the mirroring contour where blue is the ground truth, green is correct estimation, and red wrong detection. The logN matching function performs well in low textured, slanted surfaces (a) but fails in flat regions close to depth discontinuities (b). In (b) the edge of the foreground object induces an apparent symmetry that misleads the logN estimation.

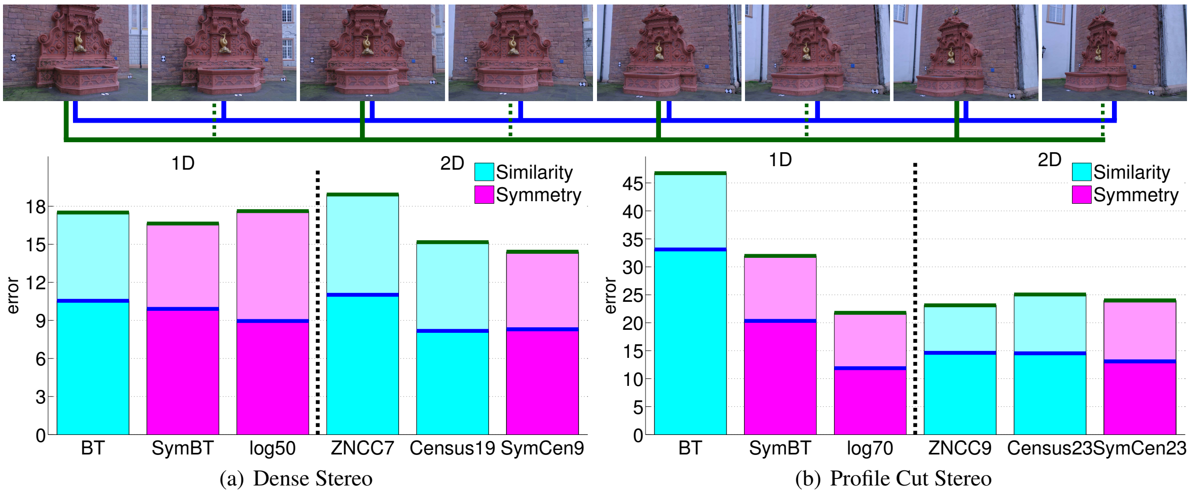
The error percentage becomes 13% which is about 5.8% more than the best result observed for dense stereo (SymCen7 with GC), and just 2% more than the best result accomplished with log20 (log20 with SGM).

Fig. 19(c) compares the performance of logN and ZNCC in the *Wood1* stereo pair by overlaying the results in detecting the mirroring contours for the 201 virtual cut planes. It can be observed that the latter, being a 2D metric with a large window support, has difficulties in handling depth discontinuities (e.g. errors in the horizontal depth transition at the top of the image, and in the occlusion region at the image center) and surface slant (e.g. errors in the boards lying on the floor). On the other hand, logN seems to combine the benefits of being a pixel-based matching cost, with a good discriminative power for pairing pixels in low textured regions. This is illustrated in Fig. 20(a) that shows the symmetry image $\mathtt{I}^S$ for a virtual cut plane that meets the scene in the vertical wooden board with significant slant. Since the pixel matching is accomplished using symmetry, the lack of local texture is partially compensated by nearby structures, such as edges and wood nodes, that contribute to successfully detect the image of the mirroring contour. Thus, the good performance in the presence of low texture is explained by the global character of the induced symmetry cue. However, and as exemplified by the situation of Fig. 20(b), such global character can become an issue whenever the contour passes in a flat region close to a depth discontinuity. In this case, the edge of the foreground object gives raise to an apparent image symmetry in the wrong location that, together with the absence of background texture, completely misleads the logN detection. It is also this phenomena that explains the poor performance of log20 close to discontinuities and occlusion regions during the dense stereo experiments (e.g. see the third column of Fig. 17). The problem can eventually be solved by using local texture information for selecting the wavelet scales at each pixel location, however the development of such a strategy is beyond the scope of this article.

Table 3 provides the average runtime for estimating the depth along a single virtual cut plane using SRF. Since the BBS filtering in BT and SymBT, and the spectral convolution in logN are executed only once independently of the number K of profile cuts, the workload required by these one time operations is accounted as an initialization overhead t_{oh}. The table shows that for $K = 1$ logN is about $10\times$

Table 3 Runtime of SRF measured in the *Teddy* stereo pair. The column t_{oh} refers to the initialization overhead whenever applicable, and the column t_{Π} reports the time for estimating disparity along a single virtual cut plane. The total time for processing K independent profile cuts is given by $t = t_{oh} + K.t_{\Pi}$.

Cost	t_{oh} (ms)	t_{Π} (ms)	**Cost**	t_{oh} (ms)	t_{Π} (ms)
BT	98	0.42	*SymBT*	98	0.60
Census19		32	*SymCen19*		33
ZNCC15		39	*log40*	378	13



(a) Dense Stereo (b) Profile Cut Stereo

Fig. 21 Mean errors on the fountain-P11 dataset [28]. The top row shows the 8 input images, while the bottom row shows the results of the different matching costs for dense stereo and SRF across the different stereo combinations (i) middle-baseline (blue), and (ii) wide-baseline (green).

slower than Census, SymCen, and ZNCC, but a quick calculation shows that for $K \geq 20$ the former becomes faster than the laters. Remark that there is no linear relationship between the runtimes of Tables 2 and 3 based on the number of image columns. The reasons are that the matching costs in SRF have larger window support and the scoring along a single plane in the DSI domain does not benefit from an efficient memory management.

7 Experiments in wide-baseline stereo

This section evaluates the performance of the 6 matching functions when the input image pairs have a wide-baseline. We consider the 8 frames of the fountain-P11 dataset [28] that are exhibited in the top row of Fig. 21. The sequence gives raise to 7 *medium-baseline* examples, corresponding to pairwise consecutive frames, and 6 *wide-baseline* examples obtained by pairing the frames with one image interval. We randomly select one of the stereo pairs for tuning the matching functions, and later discard the example for preventing bias effects during the benchmark. The selected parameters for dense stereo and SRF are shown in 3^{th} and 4^{th} columns of Table 1, respectively. The disparity range r is set by the minimum and maximum of the groundtruth disparity maps for images with size 440×640, and the threshold e for deciding about the correctness of the disparity labeling is chosen such that the ratio e/r is the same as in Section 5. Since the images are a bit larger than the dataset used in Section 6, for SRF the scene depth is independently estimated along 401 vertical cut planes.

The two plots at the bottom of Fig. 21 show the percentage of errors for dense disparity labeling (left) and SRF (right) in both *medium* and *wide-baseline* stereo pairs. The relative performance of the matching functions is in perfect accordance with the observed in Fig. 14(a) and Fig. 19(b), suggesting that all the conclusions drawn in Sections 5 and 6 hold for the case of wide-baseline imagery. It is interesting to see that in case of dense stereo, the window of SymCen is narrower than the window of Census, meaning that SymCen beats Census both in terms of accuracy and runtime.

8 Conclusions

The paper is the first work in the literature proposing to use symmetry instead of photo-similarity for assessing the likelihood of two image locations being a match. Stereo from symmetry is possible because of the *mirroring effect* that arises whenever one view is mapped into the other using the homography induced by a virtual cut plane that intersects the baseline. We provided a formal proof of this effect, studied the singularities, and investigated its usage for solving the data association problem in stereo. In the follow up of this effort we proposed three symmetry-based matching costs: *SymBT*, *SymCen*, and *logN*. The first two are closely related with the top-performing cost function BT [7] and Census [36], being in a large extent mere modifications for measuring symmetry instead of photo-similarity, while the later relies in wavelet transforms for detecting local signal symmetry. The new matching costs were benchmarked against the state-of-the-art metrics for accomplishing dense disparity labeling in both short and wide-baseline images. The results showed that the symmetry-based functions, SymBT and SymCen, consistently outperform their similarity-based counterparts, BT and Census, suggesting that symmetry is superior to standard photo-consistency as a stereo metric.

The log$N$ cost proved to be particularly effective in scenes with slanted surfaces and difficult textures, being the top-performer matching function in the Oxford Corridor dataset. The major weakness is its relative poor performance close to discontinuities and occlusion regions.We also investigated the use of passive stereo for estimating depth along a pre-defined scan plane. The technique, named Stereo Rangefinder (SRF), provides profile cuts of the scene similar to the ones that would be obtained by a LRF, enabling a tradeoff between runtime and 3D model resolution that does not interfere with depth accuracy [2]. The article described the first benchmark of SRF that showed that logN is undoubtedly the best performing matching cost.

As future work, we intend to develop local and global optimization techniques that take into account the specificities of symmetry-based cost functions, and solve the problem of lack of accuracy close to depth discontinuities and occlusion regions. We also want to investigate the joint use of symmetry and photo-similarity for improving stereo matching performance and extend the SymStereo framework to the case of multi-view stereo.

References

1. Ansar, A., Castano, A., Matthies, L.: Enhanced realtime stereo using bilateral filtering. In: 3D Data Processing, Visualization and Transmission (2004)
2. Antunes, M., Barreto, J.P.: Stereo estimation of depth along virtual cut planes. In: International Conference on Computer Vision Workshop (CVVT) (2011)
3. Antunes, M., Barreto, J.P.: Semi-dense piecewise planar stereo reconstruction using symstereo and pearl. In: 3DimPVT - 3D Data Processing, Visualization and Transmission, pp. 1–8 (2012)
4. Antunes, M., Barreto, J.P., Premebida, C., Nunes, U.: Can stereo vision replace a laserrangefinder? IEEE Int. Conf. in Intelligent Robot Systems pp. 1–8 (2012)
5. Antunes, M., Barreto, J.P., Zabulis, X.: Plane surface detection and reconstruction using induced stereo symmetry. In: British Machine Vision Conference (2011)
6. Banks, J., Corke, P.: Quantitative evaluation of matching methods and validity measures for stereo vision. The International Journal of Robotics Research (2001)
7. Birchfield, S., Tomasi, C.: A pixel dissimilarity measure that is insensitive to image sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence (1998)
8. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence (2004)
9. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence (2001)
10. Brown, M.Z., Burschka, D., Hager, G.D.: Advances in computational stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence (2003)
11. Collins, R.T.: A space-sweep approach to true multi-image matching. IEEE Conference on Computer Vision and Pattern Recognition (1996)
12. Fookes, C., Maeder, A., Sridharan, S., Cook, J.: Multi-spectral stereo image matching using mutual information. In: 3D Data Processing, Visualization and Transmission (2004)
13. Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q., Pollefeys, M.: Real-time plane-sweeping stereo with multiple sweeping directions. In: IEEE Conference on Computer Vision and Pattern Recognition (2007)
14. Gautama, S., Lacroix, S., Devy, M.: Evaluation of stereo matching algorithms for occupant detection. In: Proceedings of the International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems (1999)
15. Gong, M., Yang, R., Wang, L., Gong, M.: A performance study on different cost aggregation approaches used in real-time stereo matching. International Journal of Computer Vision (2007)
16. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)
17. Hirschmuller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. IEEE Transactions on Pattern Analysis and Machine Intelligence (2009)
18. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts? In: European Conference on Computer Vision (2002)
19. Kovesi, P.: Symmetry and asymmetry from local phase. In: Tenth Australian Joint Conference on Artificial Intelligence (1997)
20. Kovesi, P., Kovesi, P.: Image features from phase congruency. Tech. rep., Videre (1995)
21. Liu, Y., Hel-Or, H., Kaplan, C.S., Gool, L.J.V.: Computational symmetry in computer vision and computer graphics. Foundations and Trends in Computer Graphics and Vision (2010)
22. Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S.: An Invitation to 3-D Vision: From Images to Geometric Models. SpringerVerlag (2003)
23. Mordohai, P.: The self-aware matching measure for stereo. In: International Conference on Computer Vision (2009)
24. Ponce, J., McHenry, K., Papadopoulo, T., Teillaud, M., Triggs, B.: On the absolute quadratic complex and its application to autocalibration. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)

25. Sarkar, I., Bansal, M.: A wavelet-based multiresolution approach to solve the stereo correspondence problem using mutual information. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics (2007)
26. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. IEEE Conference on Computer Vision and Pattern Recognition (2007)
27. Scharstein, D., Szeliski, R., Zabih, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision (2002)
28. Strecha, C., von Hansen, W., Gool, L.J.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
29. Sun, J., Li, Y., Bing, S., yeung Shum, K.H.: Symmetric stereo matching for occlusion handling. In: IEEE Conference on Computer Vision and Pattern Recognition (2005)
30. Szeliski, R., Scharstein, D.: Sampling the disparity space image. IEEE Transactions Pattern Analysis and Machine Intelligence (2004)
31. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE Transactions on Pattern Analysis and Machine Intelligence (2008)
32. Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., Rother, C.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. IEEE Transactions on Pattern Analysis and Machine Intelligence (2008)
33. Tombari, F., Mattoccia, S., Stefano, L.D., Addimanda, E.: Classification and evaluation of cost aggregation methods for stereo correspondence. IEEE Conference on Computer Vision and Pattern Recognition (2008)
34. Wang, L., Gong, M., Gong, M., Yang, R.: How far can we go with local optimization in real-time stereo matching. In: 3D Data Processing, Visualization and Transmission (2006)
35. jin Yoon, K., Member, S., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Transactions Pattern Analysis and Machine Intelligence (2006)
36. Zabih, R., Zabih, R.: Non-parametric local transforms for computing visual correspondence. European Conference of Computer Vision (1994)