In Seventh European Conference on Computer Vision (ECCV 2002),
volume 2, pages 525–540, Copenhagen, Denmark, May 2002

Symmetric Sub-Pixel Stereo Matching

Richard Szeliski[1] and Daniel Scharstein[2]

[1] Microsoft Research, Redmond, WA 98052, USA
[2] Middlebury College, Middlebury, VT 05753, USA

Abstract. Two central issues in stereo algorithm design are the matching criterion and the underlying smoothness assumptions. In this paper we propose a new stereo algorithm with novel approaches to both issues. We start with a careful analysis of the properties of the continuous *disparity space image* (DSI), and derive a new matching cost based on the reconstructed image signals. We then use a symmetric matching process that employs visibility constraints to assign disparities to a large fraction of pixels with minimal smoothness assumptions. While the matching operates on integer disparities, sub-pixel information is maintained throughout the process. Global smoothness assumptions are delayed until a later stage in which disparities are assigned in textureless and occluded areas. We validate our approach with experimental results on stereo images with ground truth.

1 Introduction

The last few years have seen a dramatic improvement in the quality of dense stereo matching algorithms [14]. A lot of this improvement can be attributed to better optimization algorithms and better smoothness constraints [6, 4, 16]. However, a remarkable amount of the improvement has also come from better matching metrics at the input [3]. In fact, Birchfield and Tomasi's sampling-insensitive dissimilarity measure is used by a number of today's best performing algorithms [6, 4].

Using something better than just pixel-sampled intensity differences is not a new idea. For example, Matthies *et al.* interpolated scanlines by a factor of 4 using a cubic interpolant before computing the SSD score [11]. Tian and Huhns wrote an even earlier survey paper comparing various algorithms for sub-pixel registration [17]. In fact, some stereo and motion algorithms have always evaluated displacements on a half-pixel grid, but never mentioned this fact explicitly (P. Anandan, personal communication).

The set of initial matching costs that are fed into a stereo matcher's optimization stage is often called the *disparity space image* (DSI) [19, 5]. However, while the concept of stereo matching as finding an optimal surface through this space has been around for a while [19, 2, 5], relatively little attention has been paid to the proper sampling and treatment of the DSI.

In this paper, we take a more careful look at the structure of the DSI, including its frequency characteristics and the effects of using different interpolators in sub-pixel registration. Among the questions we ask are: What does the DSI look like? How finely do we need to sample it? Does it matter what interpolator we use?

We also propose a number of novel modifications to the matching cost that produce a better set of initial high-quality matches, at least in textured, unoccluded areas. It is

our contention that filling in textureless and occluded areas is best left to a later stage of processing [5, 6, 4, 16].

In the second half of the paper, we show how the local structure of the DSI can be used to select *certain* matches, i.e., matches that are correct with high probability. (Intille and Bobick [5] call such points *ground control points*.) We also develop an iterative algorithm that adds more matches, using a combination of uniqueness enforcement (filling the DSI with high costs in already matched columns), and doing more aggregation for pixels with multiple possible matches. We present the final dense matching results of our approach, which are comparable in quality to recent stereo matching algorithms, but which do not require any global optimization algorithm.

The remainder of the paper is structured as follow. Section 2 presents our analysis of the DSI and discusses minimal sampling requirements. Section 3 develops some novel matching costs based on our analysis. The utility of these novel costs is validated experimentally in Section 4. Section 5 presents our algorithm for establishing certain matches and for iteratively adding more matches to this set. We conclude with some ideas for future research.

2 Matching costs

In this section, we look at how matching costs are formulated. In particular, we analyze the structure of the DSI and its sampling properties and propose some improvements to commonly used matching costs.

Given two input images, $I_L(x, y)$ and $I_R(x, y)$, we wish to find a disparity map $d_L(x, y)$ such that the two images match as closely as possible

$$I_L(x, y) \approx I_R(x - d_L(x, y), y). \tag{1}$$

(We assume that the images have been rectified to have a horizontal epipolar geometry [12, 9]. We will also want to impose some smoothness or other prior constraints on the function $d_L(x, y)$.)

Define the 3D *signed difference image* (SDI) as the intensity (or color) difference between the shifted left and right images,

$$SDI_L(x, y, d) = I_L(x, y) - I_R(x - d, y). \tag{2}$$

Let the raw *disparity space image* (DSI) be the squared difference (summed over all the color bands),

$$DSI_L(x, y, d) = \|SDI_L(x, y, d)\|^2. \tag{3}$$

Why do we use squared differences? The analysis for this case is simpler, and it also has some other advantages we will discuss shortly. In the ideal (continuous, noise-free) case with no occlusions, we expect $DSI_L(x, y, d_L(x, y))$ to be zero.

Unfortunately, we do not actually have access to continuous, noise-free versions of $I_L(x, y)$ and $I_R(x, y)$. Instead, we have sampled noisy versions, $\hat{I}_L(x_i, y_i)$ and $\hat{I}_R(x_i, y_i)$,

$$\hat{I}_L(x_i, y_i) = [I_L * h](x_i, y_i) + n_L(x_i, y_i) \tag{4}$$

$$\hat{I}_R(x_i, y_i) = [I_R * h](x_i, y_i) + n_R(x_i, y_i), \tag{5}$$

where $h(x, y)$ is the combined point-spread-function of the optics and sampling sensor (e.g., it incorporates the CCD fill factor [18]), and n_L is the (integrated) imaging noise.

Given that we can only evaluate the DSI at the integral grid positions (x_i, y_i), we have to ask whether this sampling of the DSI adequate, or whether there is severe aliasing in the resulting signal. We cannot, of course, reconstruct the true DSI since we have already band-limited, corrupted, and sampled the original images. However, we can (in principle) reconstruct continuous signals from the noisy samples, and then compute their continuous DSI. The reconstructed signal can be written as

$$\overline{I}_L(x, y) = \sum_i \hat{I}_L(x_i, y_i) g(x - x_i, y - y_i) \tag{6}$$

$$= \tilde{I}_L(x, y) + \tilde{n}_L(x, y), \tag{7}$$

where $g(x, y)$ is a reconstruction filter, $\tilde{I}_L(x, y)$ is the sampled and reconstructed version of the *clean* (original) signal, and $\tilde{n}_L(x, y)$ is an interpolated version of the noise. This latter signal is a band-limited version of continuous Gaussian noise (assuming that the discrete noise is i.i.d. Gaussian).

We can then write the interpolated SDI and DSI as

$$\overline{SDI}_L(x, y, d) = \overline{I}_L(x, y) - \overline{I}_R(x - d, y) \quad \text{and} \tag{8}$$

$$\overline{DSI}_L(x, y, d) = \|\overline{SDI}(x, y, d)\|^2. \tag{9}$$

What can we say about the structure of these signals?

The answer can be found by taking a Fourier transform of the SDI. Let us fix y for now and just look at a single scanline,

$$\mathcal{F}\{\overline{SDI}\} = H_L(f_x) - H_R(f_x) e^{j2\pi(f_x - f_d)}, \tag{10}$$

where H_L and H_R are the Fourier transforms of $\overline{I}_L$ and $\overline{I}_R$.

Figure 1 shows the SDIs and DSIs for two scanlines taken from the 38th and 148th row of the image in Figure 1a, along with their Fourier transforms. The first term in (10) corresponds to the horizontal line in the SDI's Fourier transform (second column of Figure 1), while the second term, which involves the disparity, is the slanted line.

Squaring the SDI to obtain the DSI (third column in Figure 1) is equivalent to convolving the Fourier transform with itself (fourth column in Figure 1). The resulting signal has twice the bandwidth in x and d as the original SDI (which has the same bandwidth as the interpolated signal). It is also interesting to look at the structure of the DSI itself. The thin diagonal stripes are spurious bad matches (dark-light transitions matching light-dark transitions), while the horizontal stripes are good matching regions (the straighter and darker the better).

What can we infer from this analysis? First, the continuous DSI has significant frequency content above the frequencies present in the original intensity signal. Second, the amount of additional content depends on the quality of the interpolator applied to the signal. Thus, when perfect band-limited reconstruction (a sinc filter) is used, the resulting DSI signal only has twice the frequency of the image. It is therefore adequate (in theory) to sample the DSI at 1/2 pixel intervals in x and d. When a poorer interpolant such as piecewise linear interpolation is used, the sampling may have to be much

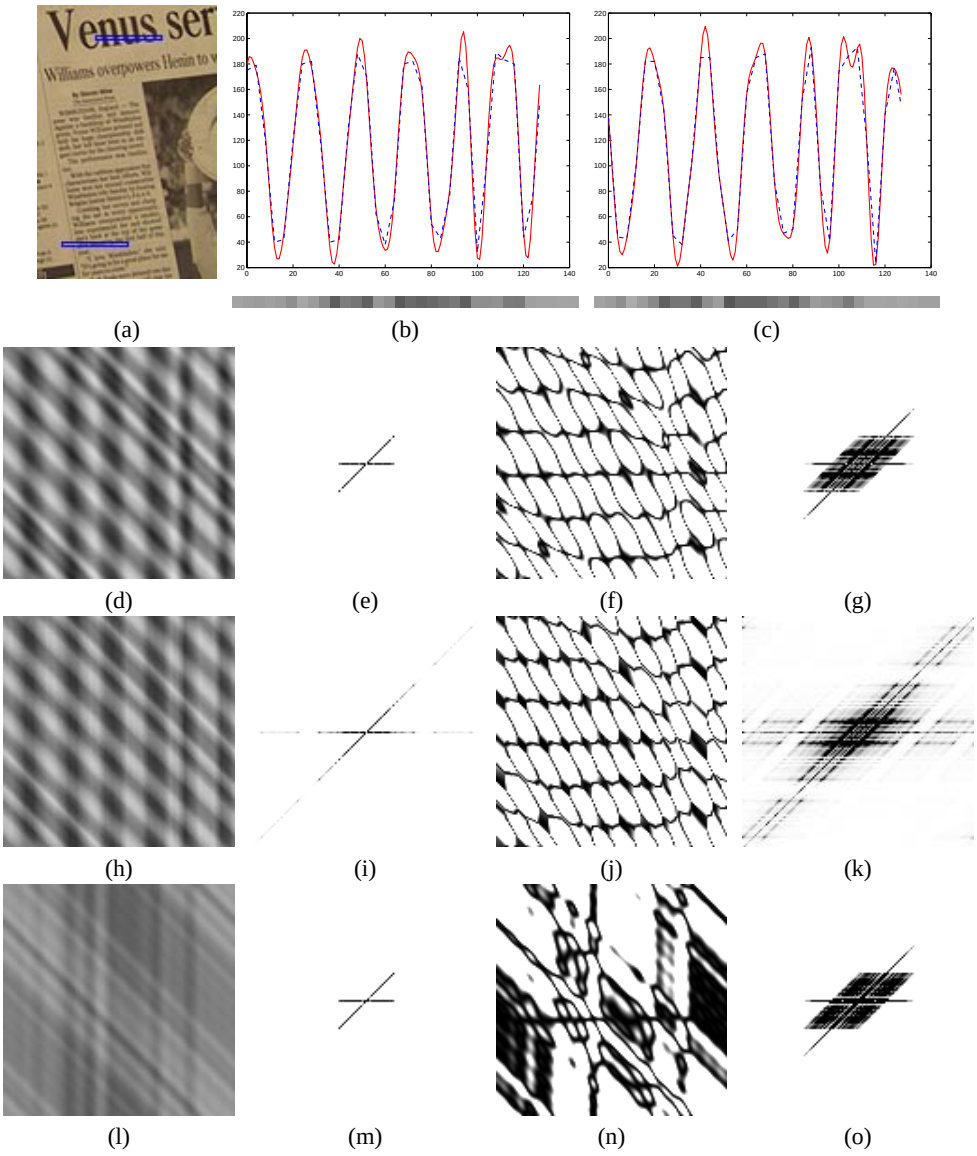


Fig. 1. Sample SDIs and DSsI and their Fourier transforms. (a) Original color image with two selected scanlines; (b–c) profiles of second selected scanline (L148); notice how the sinc-interpolated signals (red, solid) are more similar than the linearly interpolated ones (blue, dashed). (d–g) Signed Difference Image (SDI) and its transform, and Disparity Space Image (DSI) and its transform for L38, using perfect (sinc) interpolation; (h–k) same images using piecewise linear interpolation; (l–o) same images for L148 and perfect interpolation. (See the electronic version of this paper for color images.)

higher. The same is true when a different non-linearity is used to go from the SDI to the DSI, e.g., when absolute differences or robust measures are used. This is one of the reasons we prefer to use squared difference measures. Other reasons include the statistical optimality of the DSI as the log likelihood measure under Gaussian noise, and the ability to fit quadratics to the locally linearized expansion of the DSI.

We can summarize these observations in the following Lemma:

Lemma 1: *To properly reconstruct a Disparity Space Image (DSI), it must be sampled at* at least *twice the horizontal and disparity frequency as the original image (i.e., we must use at least 1/2 pixel samples and disparity steps).*

It is interesting to note that if a piecewise linear interpolant is applied between image samples before differencing and squaring, the resulting DSI is piecewise quadratic. Therefore, it suffices in principle to simply compute one additional squared difference between pixels, and to then fit a piecewise quadratic model. While this does reconstruct a continuous DSI, there is no guarantee that this DSI will have the same behavior near true matches as a better reconstructed DSI. Also, the resulting minima will be sensitive to the original placement of samples, i.e., a significant bias towards integral disparities will exist [15].

For example, if the original signal is a fairly high-frequency chirp (Figure 2a), then applying a piecewise linear interpolant will fail to correctly match the signal with a fractionally shifted version. Figure 2b and c show the results of aggregating the original raw DSIs with a 7-pixel filter (see Section 3). Clearly, using the linear interpolant will result in the wrong disparity minimum being selected in the central portion (where the central horizontal line is weak). One might ask whether such high-frequency signals really exist in practice, but it should be clear from Figure 1b and c that they do.

3 Improved matching costs

Given the above analysis, how can we design a better initial matching cost? Birchfield and Tomasi [3] and Shimizu and Okutomi [15] have both observed problems with integral DSI sampling, and have proposed different methods to overcome this problem.

Birchfield and Tomasi's *sampling-insensitive dissimilarity measure* compares each pixel in the reference image against the linearly interpolated signal in the matching image, and takes the minimum squared error as the matching cost. It then reverses the role of the reference and matching images, and takes the minimum of the resulting two cost measures. In terms of our continuous DSI analysis, this is equivalent to sampling the DSI at integral x locations, and computing the minimum value vertically and diagonally around each integral d value, based on a piecewise linear reconstruction of the DSI from integral samples. We generalize Birchfield and Tomasi's matching measure using the following two ideas:

Symmetric matching of interpolated signals First of all, we interpolate both signals up by a factor s using an arbitrary interpolation filter. In this paper, we study linear ($o = 1$) and cubic ($o = 3$) interpolants. We then compute the squared differences between *all* of the interpolated and shifted samples, as opposed to just between the original left (reference) image pixels and the interpolated and shifted right (matching)

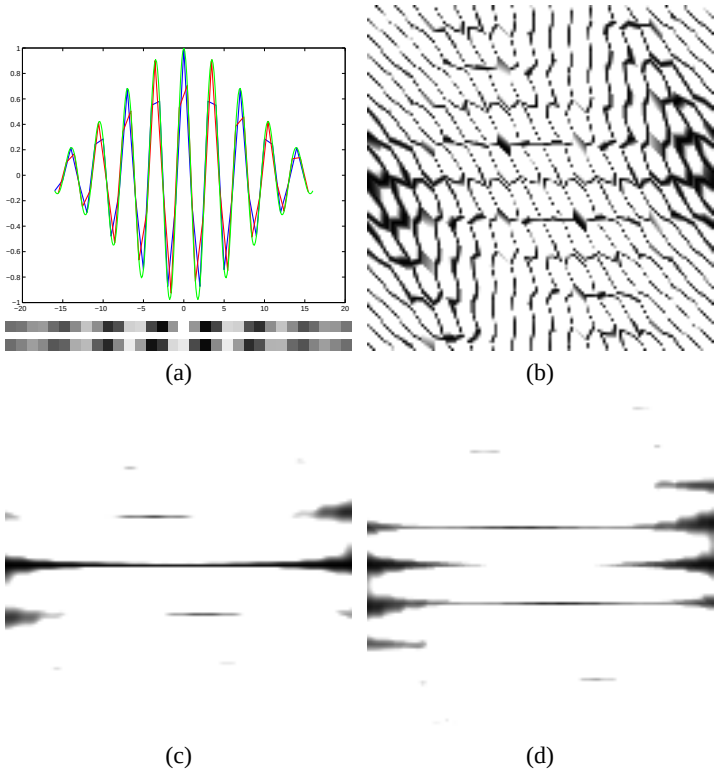


Fig. 2. Chirp signal matching: (a) the continuous green signal is sampled discretely to obtain the red and blue signals; (b) Disparity Space Image (DSI) for linear interpolation; (c) horizontally aggregated DSI for sinc interpolation, showing correct minimum; (c) horizontally aggregated DSI for linear interpolation, with incorrect minima near the center.

image samples. This difference signal is then reduced back to the original horizontal image sampling using a symmetric filter of width s and then downsampling. A higher-order filter could potentially be used, but we wish to keep discontinuities in depth sharp in the DSI, so we prefer a simple box filter.

Interval matching If we wish to still apply the idea of a sampling-insensitive dissimilarity measure [3], we can still do this on the interpolated signals before downsampling. However, rather than treating the reference and matching images asymmetrically and then reversing the roles of reference and matching (as in [3]), we have developed the following related variant that is based on interval analysis.

Figure 3 shows two signals that have been interpolated to yield the set of discrete intensity samples shown as vertical lines. The original Birchfield-Tomasi measure compares a pixel in the reference image with the interval in the matching image defined by the center pixel and its two 1/2-sample interpolated values (rectangular boxes in Figure 3b). It then performs this same computation switching the reference and match

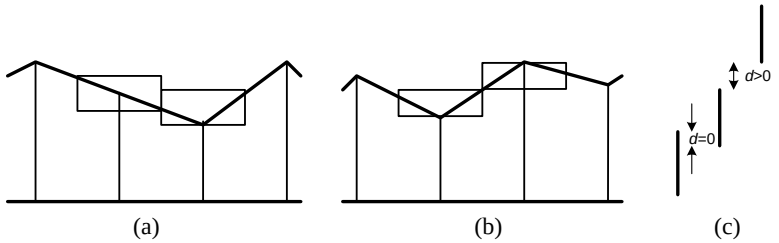


Fig. 3. Interval analysis: (a–b) two signals with their corresponding half-sample intervals; (c) three intervals being compared (differenced).

images, and takes the minimum of the resulting two costs. Our version of the algorithm simply compares the two *intervals*, one from the left scanline, the other from the right, rather than comparing values against intervals. The unsigned difference between two intervals is trivial to compute: it is 0 if the intervals overlap (Figure 3c), else it is the gap between the two intervals. A signed difference could also be obtained by keeping track of which interval is higher, but in our case this is unnecessary since we square the differences after computing them.

When working with color images, we currently apply this interval analysis to each color band separately. In principle, the same sub-pixel offset should be used for all three channels, but the problem then becomes a more complicated quadratic minimization problem instead of simple interval analysis.

Local minimum finding (quadratic fit) An alternative to doing such interval analysis is to directly compute the squared differences, and to then fit a parabola to the resulting sampled DSI. This is a classic approach to obtaining sub-pixel disparity estimates [17, 1, 11], although applying it directly to integer-valued displacements (disparities) can lead to severe biases [15].

When the DSI has been adequately sampled, however, this is a very useful alternative for estimating the analytic minimum from the (fractionally) sampled DSI. In order to reduce the noise in the DSI before fitting, we apply spatial aggregation first. In this paper, we study both fixed and shiftable square windows, as these perform quite well [14], at least in textured areas. (We will deal with untextured areas in Section 5.)

Collapsing the DSI Finally, once the local minima in the DSI at all pixels have been adequately modeled, we can collapse the DSI back to an integral sampling of disparities. This step is often not necessary, as many stereo matchers do their optimization at sub-pixel disparities. It does, however, have several potential advantages:

- For optimization algorithms like graph cuts [6] where the computation complexity is proportional to the square of the number of disparity level, this can lead to significant performance improvements.
- Certain symmetric matching algorithm such as dynamic programming and the technique developed in Section 5 require an integral sampling of disparity to establish two-way optima.

To collapse the DSI, we find the lowest matching score within a $\frac{1}{2}$ disparity from each integral disparity, using the results of the parabolic fitting, if it was used. We also store the relative offset of this minimum from the integral disparity for future processing and for outputting a final high-accuracy disparity map. Alternately, sub-pixel estimates could be recomputed at the end around each winning disparity using one of the techniques described in [17], e.g., using a Lucas-Kanade gradient-based fit [10] to nearby pixels at the same disparity.

4 Experimental evaluation of matching costs

Since there are so many alternatives possible for computing the DSI, how do we choose among them? From theoretical arguments, we know that it is better to sample the DSI at fractional disparities and to interpolate the resulting surface when looking for local minima. However, real images have noise and other artifacts such as aliasing and depth discontinuities. So, how do we choose?

In order to answer this question, we apply the techniques introduced in this section to the four test sequences described in [14], which are also available on the Web (http://www.middlebury.edu/stereo/). Two of these sequences are shown in Figure 4a. (The other two are omitted due to space limitations.) At this point, we are only interested in the accuracy of these techniques in unoccluded textured areas; techniques for estimating the disparities of the remaining pixels will be presented in the next section.

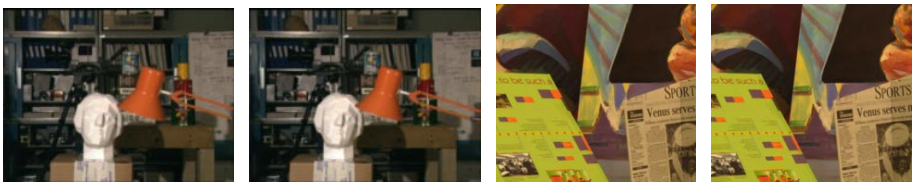
For the analysis in this section, we select textured pixels as follows: compute the squared horizontal gradient at each pixel (averaging the left and right values to remain symmetrical), and average these values in a 3×3 neighborhood. Then, threshold the averaged squared value to obtain the textureless masks shown in Figure 4c. We currently use a threshold of 9 gray levels squared.

The parameters that we vary in our experiments are as follows:

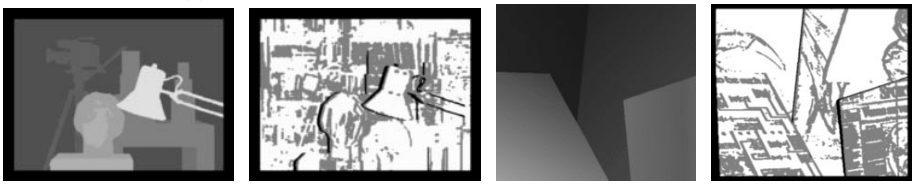
- $s = 1, 2, 4$ interpolation rate (inverse of fractional disparity)
- $o = 1, 3$ interpolation order (linear or cubic)
- i: symmetric matching of interpolated scanlines (on or off)
- d: dissimilarity metric (squared differences SD or interval differences ID)
- f: sub-pixel fit of disparities after aggregation (on or off)

The parameters that we hold fixed in the algorithm are the matching criterion (squared differences), the window size (7×7), and the fact that the windows are shiftable. We also collapse the DSI to an integer-valued sampling, so that the final winner-take-all is performed on integer disparities (with the stored sub-pixel minimum estimates used to compute the final sub-pixel disparity).

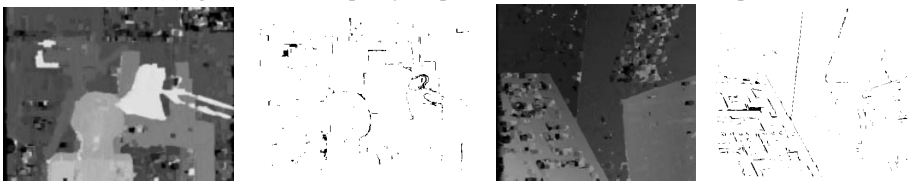
The statistics that we gather are the number of pixels that are "bad" matches, i.e., whose floating point disparity differs from the ground truth by more than 1.5 pixels. (We use 1.5 instead of 1 in order to tolerate small disparity deviations due to possible vertical misregistration.) Table 1 shows the percentage of bad matches for each of the four data sets as a function of our variable parameters. We only show results using cubic interpolation ($o = 3$); linear interpolation gives comparable, but (on average) slightly inferior results.



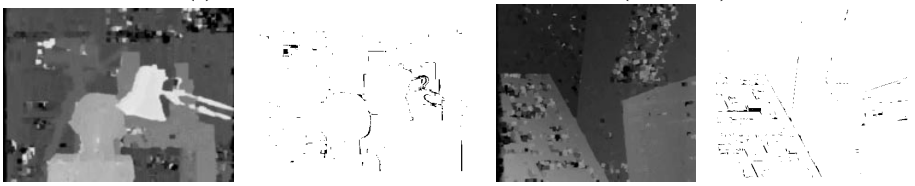
(a): Tsukuba Venus



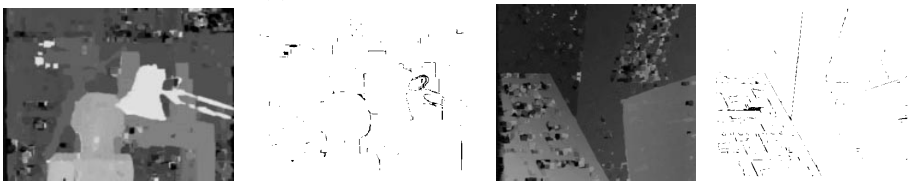
(b): ground truth disparity maps and occluded/textureless maps



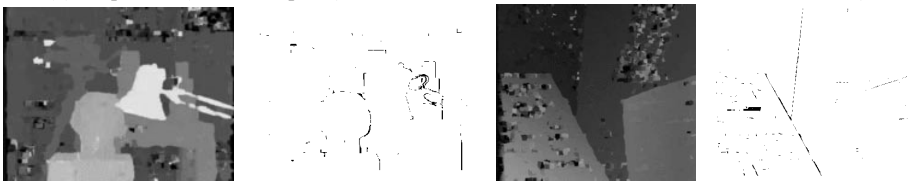
(c): traditional shiftable-window SSD results (SD, $s=1$)



(d): interval difference results (ID, $s=1$)



(e): improved (fractional pixel) shiftable-window SSD results (SD, $s=4$, $f=0$, $i=1$)



(f): improved (fractional pixel) interval difference results (ID, $s=4$, $f=0$, $i=1$)

Fig. 4. Test images and associated maps: (a) input images; (b) true disparity maps and occluded (black) and textureless (white) regions (the gray regions are the ones for which we collect statistics); (c) traditional shiftable SSD results (disparity map and error map); (d) interval difference (asymmetric Birchfield-Tomasi dissimilarity); (e) fractional disparities with symmetric matching; (f) fractional disparities with symmetric matching and interval difference.

Image	d	$s=1$	$s=2$				$s=4$			
		f:	0		1		0		1	
		i:	0	1	0	1	0	1	0	1
Sawtooth	SD	0.37	0.36	0.30	0.38	0.39	0.35	0.29	0.37	0.42
	ID	0.22	0.20	**0.12**	0.18	0.16	0.22	0.20	0.25	0.26
Venus	SD	1.33	1.10	1.23	1.21	1.34	**1.08**	1.19	1.16	1.29
	ID	4.04	1.33	4.68	1.43	4.82	1.26	1.52	1.38	1.65
Tsukuba	SD	4.38	3.69	3.51	3.73	6.84	3.72	3.43	3.77	3.37
	ID	6.54	3.25	**3.15**	3.62	3.22	3.67	3.17	3.65	3.26
Map	SD	4.72	3.85	3.25	3.84	3.24	3.66	3.20	3.64	3.22
	ID	3.15	5.77	2.38	3.05	2.55	2.92	**2.19**	2.95	2.28

Table 1. Percentage of bad matching pixels for various matching cost options. The numbers highlighted in boldface are the best matching variants for each dataset.

In Table 1, we have highlighted in boldface the lowest score for each of the four data sets. As one can see, there is no single setting that consistently outperforms the others, although sub-pixel fitting usually leads to slightly worse results, probably because it is sensitive to noise in the DSI. Figure 4 shows the results corresponding to the first ($s = 1$) and seventh ($s = 4$, $f = 0$, $i = 1$) columns. Note how the seventh column consistently outperforms both the original SD (squared difference) and ID (asymmetric interval difference) algorithms. Using interval analysis instead of sub-pixel fitting seems to usually result in lower errors, although it can lead to problems both during min-finding (this Section) and when establishing certain matches (Section 5) because many reasonable matches can yield a matching cost of 0.

Carefully examining the error maps in Figure 4 shows that the main effects of using the fractional disparity matching scores seems to be getting better results at strong intensity discontinuities. The remaining errors seem to be a combination of the classic "fattening" effect seen near disparity discontinuities (which is characteristic of local analysis), and some errors in repeated textures such as the video tapes in the Tsukuba image, which could be ameliorated with more aggregation or global optimization.

In summary, while there are no clear winners among the different cost variants, it can be seen that symmetric interpolated matching ($i=1$ and $s=2$ or $s=4$) usually outperforms traditional, integer-based matching. The benefit of interval matching depends on the winner selection strategy.

5 Symmetric matching process

We now turn to the second part of our method: the symmetric matching process. Using an interpolated matching cost that is insensitive to aliasing, we can determine the quality of matches based on the cost distribution in the DSI after a small amount of initial aggregation. Throughout this section we use symmetric matching with $\frac{1}{2}$ pixel interpolation (SD, $s = 2$, $f = 0$, $i = 1$). We are not using interval matching (ID) since this tends to "round down" good matching costs to 0, making it more difficult to draw conclusions from the cost value distributions.

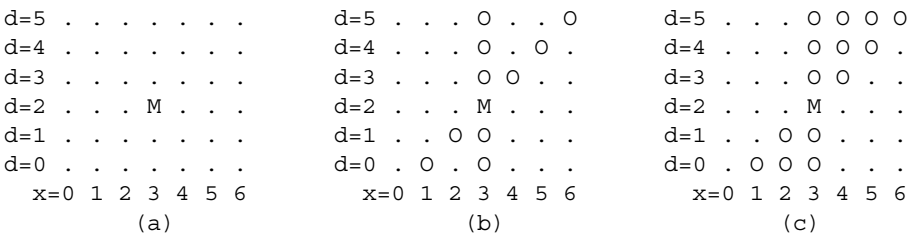


Fig. 5. Illustration of uniqueness and ordering constraints. The figures symbolize part of an x-d slice through the DSI. (a) A proposed match M for $x = 3$ and $d = 2$. (b) Other matches O ruled out by the uniqueness constraint. The vertical line eliminates other matches for the reference (left) pixel; the diagonal line eliminates other matches for the matching (right) pixel. Many asymmetric algorithms only enforce the former. (c) Some algorithms (not ours) enforce the ordering constraint, and disallow all matches in the two triangular regions.

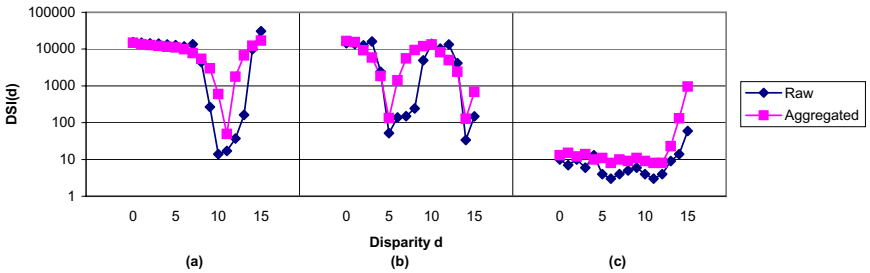


Fig. 6. Example cost value distributions (vertical DSI columns) for three locations in the Tsukuba images. We show both initial (raw) values and after aggregation with a 5×5 window. (a) Locally constant image region near the nose of the bust. Aggregation recovers the correct minimum. (b) Repetitive texture (video tapes on shelf), yielding two local minima. (c) Textureless region (shadow under the table), resulting in many low-cost values.

Our algorithm starts by selecting a subset of high-confidence matches, and then aggregates the DSI with increasingly larger windows to disambiguate matches in untextured areas. (A related technique that starts with high-confidence corner matches has been proposed by Zhang and Yang [20].) For easier reasoning about visibility, we collapse the DSI to integer sampling, as discussed in Section 3.

5.1 Selecting certain matches

There are two basic tests to determine whether a match is *certain*, i.e., correct with high probability. First, there should be a clear minimum among all candidate cost values. Second, the minima for both left-to-right and right-to-left matching should agree [7, 5], which can be checked by examining the vertical and diagonal columns in disparity space (Figure 5b).

While the second test is easy to implement, the definition of a clear minimum is less straightforward. Using one of the improved matching costs developed in the previous

section ensures that a correct match yields a low matching cost, even in high-frequency image regions. Thus, a minimum corresponding to a correct match should have a low cost value. Conversely, a large minimum value indicates an occluded pixel that has no correct match. At many pixels, however, there will be multiple low cost values that cannot always be disambiguated. Figure 6 illustrates three different cost distributions, which help motivate the following definition:

We define a match (x, y, d) with cost $C = DSI(x, y, d)$ to be *certain* if

1. C is the minimum cost in both columns, i.e.,

$$\begin{array}{lll} C \;\leq\; DSI(x, y, d') & \forall d' \neq d, \text{ and} \\ C \;\leq\; DSI(x - d + d', y, d') & \forall d' \neq d; \end{array} \tag{11}$$

2. C is a *strong* minimum in at least one column, i.e.,

$$\begin{array}{lll} C \;\leq\; m\; DSI(x, y, d') & \forall d' \neq d, \text{ or} \\ C \;\leq\; m\; DSI(x - d + d', y, d') & \forall d' \neq d \end{array} \tag{12}$$

for a given *winner margin* $m < 1$.

The winner margin m needs to be chosen low enough to avoid false positives (i.e., incorrect matches labeled certain). On the other hand, a higher margin results in a higher fraction of pixels being matched. Table 2a demonstrates this trade-off using the Tsukuba images from Figure 4. It shows the error rate among the certain matches and the total fraction of pixels matched as a function of winner margin m.

For lowest error rates, a small amount of aggregation is necessary. Here we aggregate the initial cost values with a 5×5 window. Note that unlike in Section 4, we do not need to explicitly label pixels as textureless, since this is subsumed by our test for match certainty.

5.2 Reasoning about visibility and occlusions

Before discussing how we can propagate certain matches to ambiguous regions, we briefly address how visibility constraints enter the matching process. Since we eventually need to assign matches even where there is no clear minimum among the cost values, we need to ensure that the uniqueness constraint is enforced. We can achieve this by altering the cost values "shadowed" by certain matches. Every time a new certain match has been assigned, we set the cost values for other matches eliminated by the new match (i.e., the O's in Figure 5b) to a large value $C_{\max}$. This prohibits the future assignment of low-cost matches in the diagonal DSI column. Altering the cost values can also help disambiguate multiple good matches, especially on the perimeter of textureless regions.

After the certain matches have been found and the costs have been altered as described, new certain matches may emerge where competing low cost values have been changed to $C_{\max}$. The process is thus repeated until no new certain matches can been found. Typically, this process yields an increase in certain matches of about 5–10%.

We have also experimented with enforcing the ordering constraint by assigning $C_{\max}$ to all O's in Figure 5c. We found, however, that this yields too few additional

Margin m	Bad	Matched
1.0	12.1%	96%
0.9	9.5%	90%
0.8	6.0%	81%
0.7	4.0%	73%
0.6	3.2%	66%
0.5	2.8%	59%
0.4	2.4%	53%
0.3	2.1%	45%

(a)

Final w	Pass	Bad	Matched
5	1	2.8%	59%
9	2	3.8%	83%
13	3	4.0%	90%
17	4	4.0%	91%
21	5	4.0%	91%
21	5*	4.9%	100%

(b)

Table 2. (a) Percent of bad certain matches (disparity error > 1) and fraction of pixels matched as a function of winner margin m for the Tsukuba image pair. Lower margins result in fewer errors but leave more pixels unmatched. (b) Performance of our matching algorithm while aggregating with increasing window size w (see Section 5.3 and Figure 7) using a constant margin $m = 0.5$. The last row shows the percentage of bad pixels in unoccluded regions after the remaining unmatched regions have been filled in.

certain matches to warrant the inability to deal with narrow occluding objects imposed by the ordering constraint.

The second advantage of eliminating matches by setting their costs to $C_{\max}$ is that occluded pixels can be detected more easily. We currently label all pixels as occluded when their minimum cost value is more than 10 times the average cost value of all certain matches. This works very well in textured areas with many certain matches, whose $C_{\max}$ values effectively rule out *all* disparities for pixels that must be occluded. The disparities of occluded areas can be estimated fairly reliably by filling them with the nearest background disparity on each scanline. This is the last stage in our matching algorithm. First, however, the disparities in ambiguous regions need to be recovered, which is discussed next.

5.3 Propagating matches

To propagate certain matches into ambiguous matching regions, we propose an iterative algorithm similar in spirit to adaptive-windows techniques [8, 13, 20]. Our algorithm interleaves the selection of certain matches with further aggregation of the DSI, using successively larger windows. After each aggregation step, new certain matches are selected among the previously uncommitted pixels. The algorithm converges quickly because the large cost values $C_{\max}$ are aggregated as well, which helps to rule out many ambiguous matches.

Figure 7 shows the results of our algorithm on the Tsukuba image pair. We start by aggregating with a 5×5 window, and then increase the window size by 4 after each iteration. We use a constant winner margin $m = 0.5$. After 5 iterations, disparities have been assigned to virtually all uncertain regions. The remaining occluded (high-cost) regions are filled in as described in the previous section. Finally, we restore the sub-pixel estimates computed before collapsing the DSI to integer disparities.

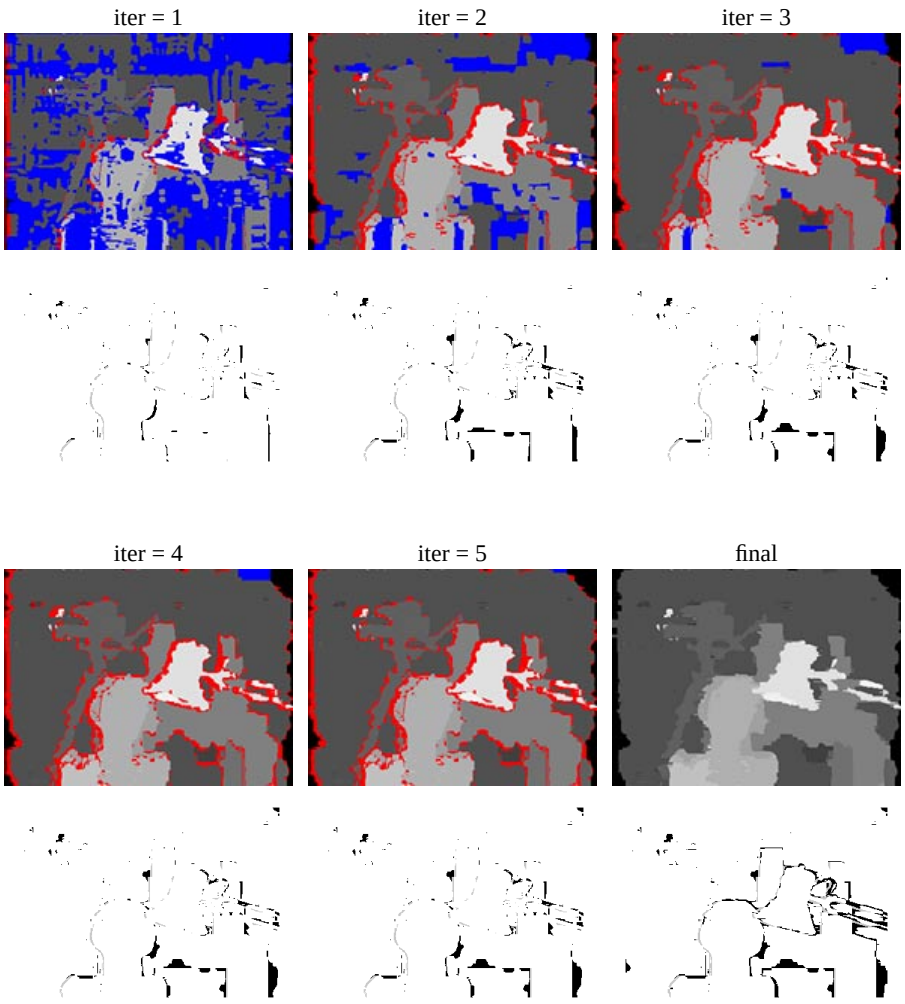


Fig. 7. Resolving unmatched areas using increasing amounts of aggregation. The figure shows a sequence of six (color) disparity maps as the matching process aggregates the DSI with windows of size 5, 9, 13, 17, and 21, and selects certain matches after each pass. Uncertain matches are shown in blue; high-cost (occluded) matches are shown in red. After the 5th pass, the remaining unmatched areas are filled in. Underneath each disparity map is the corresponding disparity error map (for certain matches only). Table 2b lists the statistics for each of the six disparity maps.

Table 2b lists the statistics for each of the five iterations. Note that the number of bad matched pixels increases only slightly, and the final numbers are quite good. The overall performance (4.9% bad unoccluded pixels) is comparable to most methods evaluated in [14], except for the graph-cut method [6]. The overall running time for this experiment is 4.7 seconds on a 750 MHz Pentium III machine.

6 Conclusion

In this paper we have presented both novel matching costs and a new symmetric matching algorithm. Our matching costs are based on interpolated image signals, and are motivated by a frequency analysis of the continuous disparity space image (DSI). We have explored several symmetric cost variants, including a generalized version of Birchfield and Tomasi's matching criterion [3]. While there is no clear winner among the different variants, we have shown that our new costs result in improved matching performance, in particular in high-frequency image regions. An added benefit is that the sub-pixel information derived during the initial cost computation can be restored at the end for the winning disparities, even if the intermediate matching process operates on integer disparities.

Our second contribution, the symmetric matching algorithm, utilizes visibility constraints to find an initial set of high-confidence matches, and then propagates disparity estimates into ambiguous image regions using successive aggregation of the DSI. Our initial experiments show competitive performance for a method that does not perform global optimization.

There are several issues we plan to address in future work. Relating to matching costs, we have started to explore the effect of certain asymmetries that occur when collapsing the subsampled DSI to an integer grid. We also want to evaluate our matching costs with further experiments, and compare them with the method developed by Shimizu and Okutomi [15]. Relating to disparity estimation, we plan to improve our current algorithm and to test its performance on image pairs with narrow occluding objects that violate the ordering constraint. In the longer term, we hope to achieve results competitive with the graph-cut algorithm [6] by exploring new ways of imposing global smoothness constraints in textureless areas.

References

1. P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, January 1989.
2. P. N. Belhumeur. A Bayesian approach to binocular stereopsis. *International Journal of Computer Vision*, 19(3):237–260, August 1996.
3. S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, April 1998.
4. S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Seventh International Conference on Computer Vision (ICCV'99)*, pages 489–495, Kerkyra, Greece, September 1999.

5. A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, September 1999.
6. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001.
7. P. Fua. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6(1):35–49, Winter 1993.
8. T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.
9. C. Loop and Z. Zhang. Computing rectifying homographies for stereo vision. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99)*, volume I, pages 125–131, Fort Collins, June 1999.
10. B. D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, pages 674–679, Vancouver, 1981.
11. L. H. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.
12. M. Okutomi and T. Kanade. A multiple baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363, April 1993.
13. D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *International Journal of Computer Vision*, 28(2):155–174, July 1998.
14. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.
15. M. Shimizu and M. Okutomi. Precise sub-pixel estimation on area-based matching. In *Eighth International Conference on Computer Vision (ICCV 2001)*, volume I, pages 90–97, Vancouver, Canada, July 2001.
16. H. Tao, H.S. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *Eighth International Conference on Computer Vision (ICCV 2001)*, volume I, pages 532–539, Vancouver, Canada, July 2001.
17. Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35:220–233, 1986.
18. Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of CCD imaging process. In *Eighth International Conference on Computer Vision (ICCV 2001)*, volume I, pages 480–487, Vancouver, Canada, July 2001.
19. Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 274–279, New York, New York, June 1993. IEEE Computer Society.
20. Z. Zhang and Y. Shan. A progressive scheme for stereo matching. In M. Pollefeys et al., editors, *Second European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE 2000)*, pages 68–85, Dublin, Ireland, July 2000.