The HCI Stereo Metrics: Geometry-Aware Performance Analysis of Stereo Algorithms

Katrin Honauer[1] Lena Maier-Hein[2] Daniel Kondermann[1]

[1]HCI, Heidelberg University
firstname.lastname@iwr.uni-heidelberg.de

[2]German Cancer Research Center (DKFZ)
l.maier-hein@dkfz-heidelberg.de

Abstract

Performance characterization of stereo methods is mandatory to decide which algorithm is useful for which application. Prevalent benchmarks mainly use the root mean squared error (RMS) with respect to ground truth disparity maps to quantify algorithm performance.

We show that the RMS is of limited expressiveness for algorithm selection and introduce the HCI Stereo Metrics. These metrics assess stereo results by harnessing three semantic cues: depth discontinuities, planar surfaces, and fine geometric structures. For each cue, we extract the relevant set of pixels from existing ground truth. We then apply our evaluation functions to quantify characteristics such as edge fattening and surface smoothness.

We demonstrate that our approach supports practitioners in selecting the most suitable algorithm for their application. Using the new Middlebury dataset, we show that rankings based on our metrics reveal specific algorithm strengths and weaknesses which are not quantified by existing metrics. We finally show how stacked bar charts and radar charts visually support multidimensional performance evaluation. An interactive stereo benchmark based on the proposed metrics and visualizations is available at: http://hci.iwr.uni-heidelberg.de/stereometrics

1. Introduction

Disparity maps computed from stereo image pairs often serve as crucial input for higher level vision tasks such as object detection, 3D reconstruction, and image based rendering, which are in turn used in applications such as driver assistance [31] and computer assisted surgery [24].

Fueled by the renowned Middlebury benchmark [34], stereo matching algorithms have made tremendous progress in the past decade. Since then, stereo benchmarks have become increasingly challenging, diverse and realistic with datasets such as the new Middlebury dataset [33], KITTI [10], HeiSt [20] and the new SINTEL stereo data [5]. Top ranking algorithms on these benchmarks have long left behind purely pixel-based approaches. Instead, they hypothesize on local geometry, including segment-wise plane fitting [16], explicit support for slanted and curved surfaces [2, 39] as well as integrating sophisticated shape priors and object recognition [3, 4, 12]. Even though this evolution

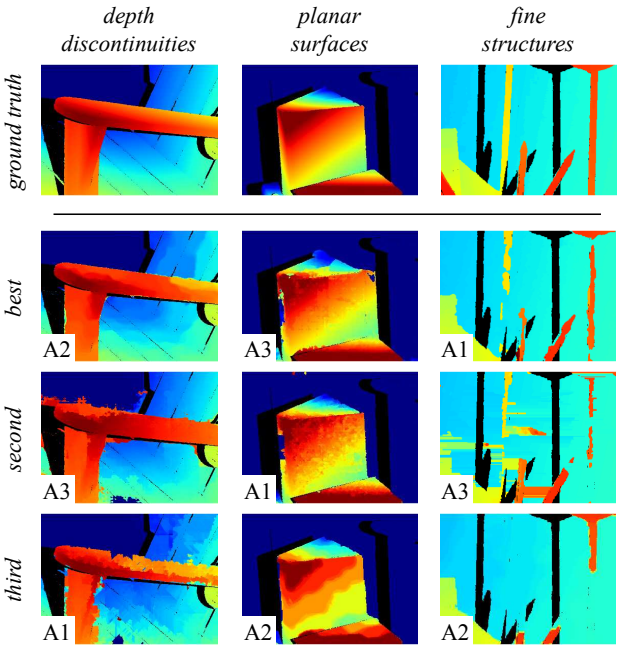


Figure 1: The same three algorithms A1-A3 rank differently, depending on which of our proposed performance metrics is used. For example, A1 is "the best algorithm" according to the widely used *RMS* measure. Yet, A1 yields the lowest performance at depth discontinuities. The column rankings show that our metrics allow for a more expressive and semantically intuitive assessment of stereo results with respect to depth discontinuities, planar surfaces, and fine structures. (Black denotes occluded regions.)

towards higher-level reasoning started more than ten years ago, performance evaluation in the stereo community still mainly works with purely pixelwise comparison of disparity differences. The two prevalent metrics are 1) *RMS*, which denotes the root mean squared pixelwise disparity difference to a given ground truth disparity (GT) and 2) *BadPix*, the fraction of pixels whose disparity error exceeds a certain threshold, commonly set to 1 or 2 pixels.

Given this situation, our goal is to let stereo evaluation catch up with the progress of the stereo algorithms it is supposed to assess. Yet, introducing novel metrics for stereo evaluation is only justified if these metrics foster new valuable insights and complement the established metrics *RMS* and *BadPix*. On the one hand, the established metrics already fulfill many requirements for good performance metrics as they are widely applicable, easy to compute, independent of image dimensions, and commonly accepted. On the other hand, metrics which average over all image pixels cannot account for the fact that input pixels for stereo applications are neither spatially independent nor equally important or equally challenging.

In the Middlebury Stereo Evaluation v.3[1], Scharstein and Hirschmüller address this issue by using binary masks for occluded pixels and linear image weights for the overall ranking. We build upon this idea and further flesh out the information given in existing GT disparity maps. We automatically extract GT pixel subsets of geometric structures at semantically meaningful image regions such as planar surfaces. These subsets can be extracted from different GT datasets and applied to dense depth maps generated by stereo or other reconstruction methods.

Our contribution is threefold:

1. We propose the *HCI Stereo Metrics*, a novel set of nine semantically intuitive metrics which characterize stereo performance at depth discontinuities, planar surfaces, and fine structures (Section 3).
2. We re-evaluate recent Middlebury submissions, reveal previously unquantified algorithm properties, and demonstrate how metric combinations and multidimensional visualizations can be used to optimize for application-specific requirements (Section 4).
3. We provide source code for our evaluation framework and publish an interactive benchmarking website[2].

2. Related Work

The state-of-the-art performance evaluation method for stereo algorithms clearly consists of comparing *RMS* scores achieved on the Middlebury [33, 34] and KITTI [10] datasets with the published scores on the respective benchmark websites. Both benchmarks provide scores computed on full, non-occluded and occluded pixel subsets. Middlebury v.2 additionally provides scores for pixel subsets at depth discontinuities.

Looking from a broader perspective, performance evaluation for correspondence problems tends to be either very theoretical or very application-specific [8, 19].

On the theoretical side, Barnard and Fischler defined a comprehensive set of characteristics ranging from accuracy and reliability to domain sensitivity and computational complexity [1]. Maimone and Shafer analyzed which performance characteristics can be assessed on test setups ranging from empirical uncontrolled environments over engineered test data to pure mathematical analysis [25]. Haralick suggested sound statistical performance characterization with random perturbations of the algorithm input [13]. Despite their mathematical universality, most of these evaluation methods are hardly feasible for stereo evaluation in current research and real-world scenarios because they often require exact and comprehensive models of the algorithms, problem domains, and input data.

On the application-oriented side, a variety of evaluation methods has been proposed, such as for pedestrian or lane detection in driver assistance scenarios [9, 17, 27, 31]. Maier-Hein et al. proposed evaluation metrics for stereo accuracy, robustness, point density and computation time in laparoscopic surgery [23, 24]. Further specialized evaluation methods were proposed with regard to immersive visualization for tele-presence [29], video surveillance systems [37], and imaging parameter dependence on Mars missions [18]. Those methods accomplish their specific purpose very well but the problem-specific insights are often not easily transferable to other domains.

Our goal is to find a good trade-off between those two areas of research. We aim at developing theoretically sound general purpose metrics which are nonetheless easily applicable to existing benchmark datasets and parameterizable to suit the specific needs of different applications.

In the stereo community, Kostková et al. reasoned that performance evaluation should take the algorithm purpose into account and showed that evaluation must not be limited to basic pixel averaging [21]. Instead, they discriminate matching errors such as the false negative rate and occlusion boundary inaccuracy. Furthermore, we borrow ideas from the segmentation and object detection communities to include higher level reasoning about the image structure: Margolin et al. proposed evaluation metrics for foreground maps which incorporate the fact that pixels are neither spatially independent nor equally important [26]. Özdemir et al. developed performance metrics for object detection evaluation which are sensitive to boundary and fragmentation errors [30]. Yasnoff et al. state that good metrics for scene segmentation should incorporate error categories for different picture elements and have adjustable costs [38].

[1] http://vision.middlebury.edu/stereo/eval3
[2] http://hci.iwr.uni-heidelberg.de/stereometrics

3. Novel Metrics for Stereo Evaluation

In this Section, we introduce theoretical principles for the quantitative evaluation of stereo performance at depth discontinuities, planar surfaces, and fine structures. For each of these geometric entities, we first motivate their relevance for stereo applications, then briefly explain how we obtain the respective ground truth subsets, and finally propose distinct metrics to formally quantify stereo performance. For each proposed metric, 0 denotes a perfect result and higher values indicate lower performance. The methods to obtain the relevant pixel subsets are only briefly outlined in this Section. Further details are given in the supplemental material.

3.1. Depth Discontinuities

Depth discontinuities are defined as image regions where the disparity differences between adjacent pixels exceed a certain threshold. Sharp and accurate *disparity edges* are important for applications such as object detection and tracking [15]. Yet, depth discontinuity areas are challenging and error-prone due to occlusion effects and either the smoothness terms of global stereo algorithms or the local support windows of local algorithms.

We propose metrics to quantify three phenomena at depth discontinuities: foreground fattening, foreground thinning, and fuzziness. Figure 2 depicts schematic illustrations of these phenomena together with actual disparity maps and visualizations of our metrics.

To quantify the described characteristics, we define $\Omega \subset \mathbb{N}^2$ as the set of pixels of a given image. We then define $\mathcal{M}_d \subset \Omega$ as the subset of pixels which are located at high gradients of the ground truth disparity map D_{gt}. By linearly following local gradient directions on both sides of

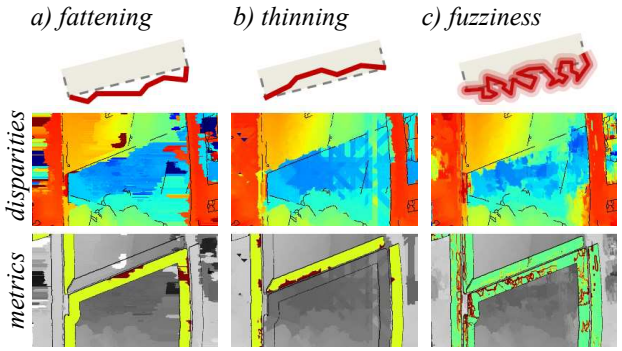


Figure 2: Stereo algorithms yield very different performance at depth discontinuities (*middle row*). With our metrics (*bottom row*), we quantify the degree of a) edge fattening, b) thinning and c) fuzziness using geometric clues extracted from GT disparity maps. The GT disparity and pixel subsets used for the evaluation are illustrated in Figure 3.

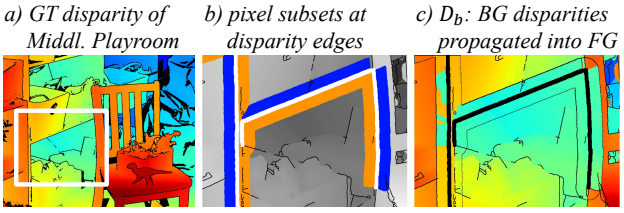


Figure 3: To quantify edge thinning and fattening, we automatically extract ground truth subsets (b) for depth discontinuities (white), nearby foreground objects (blue), and the adjacent background (orange). We further create extrapolated disparity maps where nearby background disparities are propagated into the foreground (c) and vice versa.

the discontinuity and applying median filtering, we obtain the pixel subsets $\mathcal{M}_f$ and $\mathcal{M}_b$ (shown in Figure 3.b). They denote the foreground and background areas on either side of the discontinuity. We further introduce the extrapolated disparity maps D_f and D_b. For D_b, those disparities of $\mathcal{M}_b$ which are closest to the discontinuity are propagated into $\mathcal{M}_f$ along the local gradient directions (see Figure 3.c).

D1. Foreground Fattening. We quantify foreground fattening by defining $\mathcal{M}_{fat}$ as the subset of pixels, whose estimated disparity $D_a(\vec{x})$ is closer to the extrapolated foreground $D_f(\vec{x})$ than to the actual background $D_{gt}(\vec{x})$, i.e.:

$$\mathcal{M}_{fat} = \{\vec{x} \in \mathcal{M}_b \colon |D_a(\vec{x}) - D_{gt}(\vec{x})| > |D_a(\vec{x}) - D_f(\vec{x})|\} \tag{1}$$

The degree of foreground fattening $D_{fat} \in [0, 1]$ is then defined as the cardinality of $\mathcal{M}_{fat}$ normalized by the total number of considered pixels:

$$D_{fat} = |\mathcal{M}_{fat}| \,/\, |\mathcal{M}_b| \tag{2}$$

D2. Foreground Thinning. Similarly, we quantify foreground thinning by defining the subset of pixels whose estimated disparity $D_a(\vec{x})$ is closer to the extrapolated background $D_b(\vec{x})$ than to the actual foreground $D_{gt}(\vec{x})$, i.e.:

$$\mathcal{M}_{thin} = \{\vec{x} \in \mathcal{M}_f \colon |D_a(\vec{x}) - D_{gt}(\vec{x})| > |D_a(\vec{x}) - D_b(\vec{x})|\} \tag{3}$$

The normalized $D_{thin} \in [0, 1]$ is then defined as:

$$D_{thin} = |\mathcal{M}_{thin}| \,/\, |\mathcal{M}_f| \tag{4}$$

D3. Fuzziness. Algorithm results with sharp edges yield strong disparity gradients close to depth discontinuities and smaller gradients at more distant pixels. Thus, we compute $G = \|\nabla D_{gt}\| - \|\nabla D_a\|$, the differences of absolute disparity gradient magnitudes between the GT and the algorithm disparity map. We penalize the differences weighted by their distance to the depth discontinuity. We use the common distance metric $dist(\vec{x}, \mathcal{M}) = \min_{\vec{x}_i \in \mathcal{M}} \|\vec{x} - \vec{x}_i\|$ to

find the closest element in the set of edge area pixels $\mathcal{M}_e = \mathcal{M}_d \cup \mathcal{M}_f \cup \mathcal{M}_b$. Furthermore, we define:

$$f(\vec{x}) = \begin{cases} |G(\vec{x})| \cdot dist(\vec{x}, \mathcal{M}_d), & \text{if } G(\vec{x}) < 0 \\ G(\vec{x}) \cdot dist(\vec{x}, \Omega \setminus \mathcal{M}_e), & \text{otherwise} \end{cases} \quad (5)$$

which penalizes overly strong gradients by their distance to discontinuities and overly soft gradients by their closeness. Finally, we quantify the fuzziness of discontinuities as:

$$D_{fuz} = \frac{1}{|\mathcal{M}_e|} \sum_{\vec{x} \in \mathcal{M}_e} f(\vec{x}) \quad (6)$$

3.2. Planar Surfaces

Reconstructed planar surfaces are used with very different requirements among stereo applications like image-based rendering or driver assistance. While some applications care about the correct principal orientation, others require the exact distance or prefer smooth but slightly tilted planes over more accurate yet uneven planes with artifacts.

A common strategy among many stereo algorithms is to fit local planes or splines to some sort of superpixels [16, 35, 39]. Their parametrization often is a trade-off between locally accurate fits with jumps between the superpixels or smoother yet less accurate results.

We propose three metrics to quantify the described characteristics of planar surfaces: bumpiness, offset, and local misorientation (compare Figure 4). To quantify the proposed characteristics, we use RANSAC to robustly fit planes to connected regions of homogeneous gradient directions in D_{gt}. With $\mathcal{P} = \{p_0, ..., p_m\}$, we denote the set of m fitted planes in disparity space, defined in point-normal form $p_i = (\vec{n}_i, P_i)$. The set of pixels whose disparity values belong to the fitted planes is denoted as $\mathcal{M}_p$.

P1: Bumpiness. Disparity maps at planar surfaces should ideally have homogeneous gradients and hence a constant second derivative. To quantify bumpiness, we therefore compute the second derivative of the algorithm result D_a using the Laplacian Δ and denote the metric as:

$$P_{bump} = \frac{1}{|\mathcal{M}_p|} \sum_{\vec{x} \in \mathcal{M}_p} |\Delta D_a(\vec{x})| \quad (7)$$

P_{bump} is 0 if all gradients of the estimated disparity map are smooth and bigger than 1 for strong bumpiness.

P2: Offset. To quantify the offset, we consider all elements in $\mathcal{M}_p$ and compute the Euclidean distance $d(\vec{X}, p)$ of each 3D point $\vec{X} = (x, y, D_a(x, y))$ to its corresponding plane $p = (\vec{n}, P)$:

$$P_{off} = \frac{1}{|\mathcal{M}_p|} \sum_{p_i \in \mathcal{P}} \sum_{\vec{x} \in \mathcal{M}_{p_i}} d(\vec{X}, p_i) \quad (8)$$

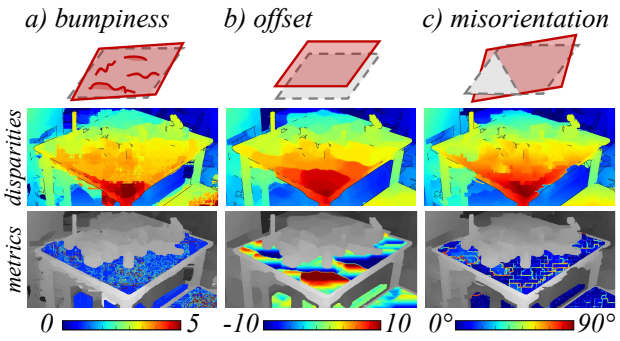


Figure 4: Reconstructing planar surfaces such that they are smooth as well as correctly located and oriented is challenging for stereo algorithms. Our metrics quantify that the stereo result on the Middlebury v.3 *Playtable* displayed in b) has locally smooth areas which suffer from inaccurate orientation leading to locally increasing offsets from the true plane. In c) the local orientation is only slightly off for some patches but their relative offset to the plane leads to significant jumps between them.

P3: Local Misorientation. To quantify the misorientation in D_a, we estimate the local surface orientation at each element in $\mathcal{M}_p$ by fitting a plane to its 5×5 neighborhood using standard least squares. With $\vec{n}_a(\vec{x})$ denoting the estimated unit surface normal of D_a at $\vec{x}$, we compute the average angle difference to the GT unit normal $\vec{n}_i$ as:

$$P_{orient} = \frac{1}{|\mathcal{M}_p|} \sum_{p_i \in \mathcal{P}} \sum_{\vec{x} \in \mathcal{M}_{p_i}} \sphericalangle(\vec{n}_a(\vec{x}), \vec{n}_i) \quad (9)$$

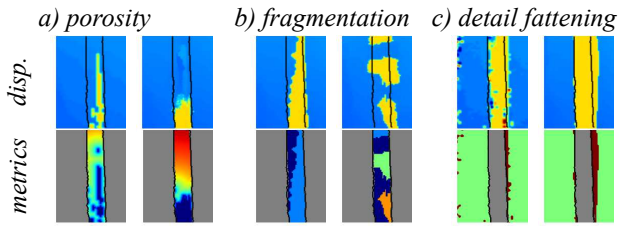
Values for P_{orient} range from $0°$ for perfect normals to $90°$ for surfaces which are orthogonal to the GT plane.

3.3. Fine Structures

Reconstructing depth at fine, elongated structures of small horizontal extent is challenging for stereo algorithms. In the trade-off between minimizing artifacts and preserving fine structures, the latter are often sacrificed for smooth disparities at larger objects. But reconstructing fine structures is essential for obstacle detection in autonomous navigation and medical instrument detection in laparoscopic surgery.

Metrics averaging over the entire image are very tolerant against such errors, as the structures make up just a small fraction of the image. We propose three metrics to quantify algorithm performance at fine structures: porosity, fragmentation, and detail fattening (see Figure 5).

To quantify algorithm performance at fine structures, we define the subset $\mathcal{M}_s$ denoting all pixels which belong to vertical fine structures. We obtain this subset by shifting positive and negative gradients of D_{gt} towards each other and by keeping regions with high overlap. Since many stereo applications primarily care about the detection of



RMS	33.86	34.15	23.50	36.04	21.07	23.10
F_{por}	**1.87**	**2.66**	0.53	1.06	0.35	0.00
F_{frag}	0.50	0.00	**0.00**	**0.67**	0.00	0.00
F_{fat}	0.00	0.00	0.03	0.09	**0.06**	**0.11**

Figure 5: *Top:* Stereo algorithms produce very different results at fine structures. For these regions, our metrics visualize (*Middle*) and quantify (*Bottom*) porosity, fragmentation and detail fattening (lower values are better). Both results in a) detect a comparable amount of the structure. Yet, the left result is distributed over the entire structure, yielding a better value for the sampling metric F_{por}.

fine structures but are rather tolerant about their exact distance, we further define the pixel subset $\mathcal{M}_a$ for correctly detected fine structure elements in D_a. This set includes all pixels whose disparity differences are within a given error tolerance to D_{gt}.

F1: Porosity. Given a fixed number of correctly detected structure pixels, their spatial distribution can make a big difference when estimating the shape of a structure. As shown in Figure 5.a small fragments which are distributed over the entire structure may be preferred over a connected block which misses half of the structure. We quantify this characteristic by penalizing big missing parts of fine structures. For each missing structure element in $\mathcal{M}_m = \mathcal{M}_s \setminus \mathcal{M}_a$, we compute the logarithmic distance to the closest correct structure element in $\mathcal{M}_a$:

$$F_{por} = \frac{1}{|\mathcal{M}_s|} \sum_{\vec{x} \in \mathcal{M}_m} \log(1 + dist(\vec{x}, \mathcal{M}_a)) \qquad (10)$$

F2: Fragmentation. Fine structures which are fragmented into multiple substantial parts can be misleading for applications like object recognition. We quantify the fragmentation of $\mathcal{M}_a$ for each structure by computing the amount of 8-connected components. Normalized by the number of GT structures, fragmentation is quantified as:

$$F_{frag} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} (1 - \frac{1}{|\mathcal{F}_s|}) \qquad (11)$$

where $\mathcal{S}$ is the set of ground truth structures and $\mathcal{F}_s$ the set of estimated fragments for each structure $s \in \mathcal{S}$. F_{frag} is 0, if the algorithm produces a single component per structure and closer to 1 with an increasing number of fragments.

F3: Detail Fattening. Similarly to D_{fat} for edge fattening, we quantify the extent to which pixels left and right of fine structures, denoted as $\mathcal{M}_n$, are erroneously closer to the extrapolated structure D_n than to the background D_{gt}. This is particularly relevant as fine structures often appear as part of grids which tend to be estimated as solid objects.With:

$$\mathcal{M}_{dfat} = \{\vec{x} \in \mathcal{M}_n \colon |D_a(\vec{x}) - D_{gt}(\vec{x})| > |D_a(\vec{x}) - D_n(\vec{x})|\} \qquad (12)$$

the degree of detail fattening is defined as:

$$F_{fat} = |\mathcal{M}_{dfat}| \,/\, |\mathcal{M}_n| \qquad (13)$$

4. Experimental Validation

In this Section we perform a threefold validation of our proposed evaluation metrics. After describing the experimental setup, we first test the expressiveness and specificity of our individual metrics on recent submissions to the Middlebury benchmark. We then introduce visualization methods to demonstrate the feasibility of multidimensional performance analysis. Finally, we validate whether our metrics are orthogonal to the established *RMS* and *BadPix* metrics.

4.1. Experimental Setup

Our experiments are based on the new Middlebury benchmark v.3 which is split into 15 test and 15 training images. Upon submission to the evaluation page, algorithm results on the training images are made publically available in full resolution. For our experiments, we drop images with intentionally imperfect illumination or rectification (*PianoL, Playtable, MotorcycleE*) and keep the remaining 12 images. As stereo results we use the highest resolution submission of each of the 13 available algorithms, namely *BSM* [40], *Cens5* [15], *ELAS* [11], *IDR* [22], *LCU*[3], *LDSM* for *LAMC_DSM* [36], *LPS* [35], *MeshS*[3], *SGBM1*[4], *SGBM2*[4], *SGM* [14], *SNCC* [7], and *TSGO* [28]. In line with Middlebury, we use dense stereo results and evaluate on full resolution. We also exclude occluded pixels, an image boundary of $max(20, 0.01 * imgwidth)$ pixels and those pixels where D_{gt} was marked as invalid.

4.2. Qualitative Metric Evaluation

In this Section, we exemplarily test how much the ranking defined by our metrics correlates with the intuitive ranking of the respective characteristics at test.

Depth Discontinuities Figure 6 depicts three stereo results for the *Adirondack* image ranked by their performance at edge thinning and fuzziness. The ranking by thinning corresponds well to the intuitive assessment of the disparity edges, particularly at the left side of the back rest. Similarly, edge fuzziness corresponds with the amount of artefacts in all three stereo results, particularly at the arm rest.

[3] anonymous submission

[4] www.opencv.org - implementation of SGM [14]

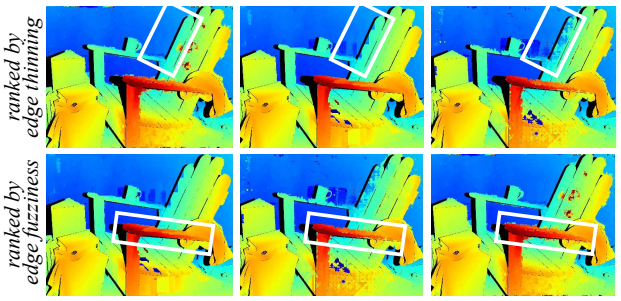


Figure 6: *Top:* Analyzed for edge thinning, the algorithms *LDSM*, *SNCC*, *SGM* rank best from left to right with 0.01, 0.05, and 0.13 for D_{thin}. *Bottom:* Analyzed for edge fuzziness, their relative order changes to *SNCC*, *SGM*, *LDSM* with 0.63, 0.72, and 0.75 for D_{fuz}.

Planar Surfaces The disparity maps depicted in Figure 4 show that stereo results at planar surfaces indeed perform well at one surface metric whilst performing lower at another. From left to right the stereo results are *LPS*, *TSGO*, and *MeshS*. With an average angle difference of 9.52° on the entire subset $\mathcal{M}_p$, *LPS* achieves more accurate surface orientations than *TSGO* with 20.03°. Yet in terms of surface bumpiness, *TSGO* ranks better with a value of 0.45 as compared to 1.82 for *LPS*.

Fine Structures From left to right, Figure 5 depicts disparity maps of the algorithms *LCU*, *LDSM*, *IDR*, *MeshS*, *BSM* and *Cens5*, which are taken from the *Pipes* image. Below, we show pairwise visualizations of the metrics F_{por}, F_{frag}, and F_{fat}, together with the metric scores for each algorithm. Clearly, *Cens5* has the best sampling which is correctly quantified by $F_{por} = 0$. Interesting to note are the first two algorithm results. With 33.86 and 34.15 they have very similar *RMS* values but the first result supports a much better reconstruction of the structure, which is correctly quantified by the lower F_{por} metric of 1.87 as compared to 2.66. Similarly, the values for F_{frag} and F_{fat} correspond well to the intuitive ranking of the displayed disparity maps.

4.3. Comparison of Algorithm Performance

Combined algorithm performance is ideally evaluated on a range of representative images. Since images have different content and difficulty, benchmarks such as Middlebury v.3 apply weights to normalize metric values across test images. Our metrics are naturally normalized across images as they only consider specific subsets on each image. The upper bar chart in Figure 7 illustrates linearly combined performance metrics for three algorithms averaged over all test images. *Cens5* shows the best overall performance and is best at planar surfaces. *SGBM1* has a lower overall performance but it is more sensitive to detecting fine structures.

Weights for individual metrics can easily be adjusted to meet the priorities of specific application domains. For instance, augmented reality applications in computer assisted minimally-invasive surgery require accurate reconstruction of the poses of medical instruments [6]. As shown in Figure 8, this includes detecting fine structures such as sutures, which are very challenging for stereo algorithms [23].

The lower chart in Figure 7 illustrates how relative rankings change when performance at fine structures is given a higher weight. According to the new ranking on our test data, *SGBM1* would be a better choice for applications in computer-assisted surgery than *BSM* or *Cens5*.

A *multidimensional analysis* is useful in situations where algorithm performance must be thoroughly assessed; in such cases a single combined performance scalar is often insufficient. For instance, researchers publishing a new stereo algorithm with particular focus on depth discontinuities would ideally be able to show quantitatively that their algorithm performs better at discontinuities and maintains good scores at the remaining characteristics. Using radar charts as depicted in Figure 9, different algorithms can be compared with regard to multiple performance characteristics based on their relative ranking and their absolute scores. In our case, lower values in the center represent the highest performance and algorithms further outside rank lower.

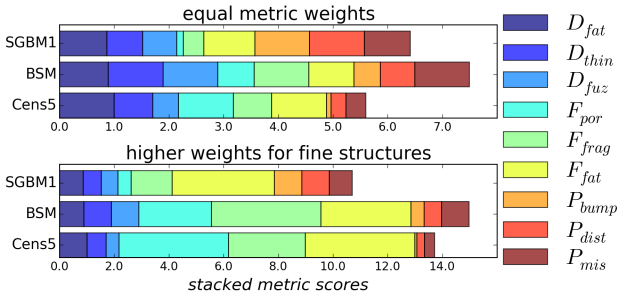


Figure 7: *Top:* Among the depicted algorithms *Cens5* has the best overall performance (shortest bar). It is particularly good at planar surfaces. *Bottom:* For applications where fine structures are important, relative metric weights differ such that other algorithms like *SGBM1* are better suited.

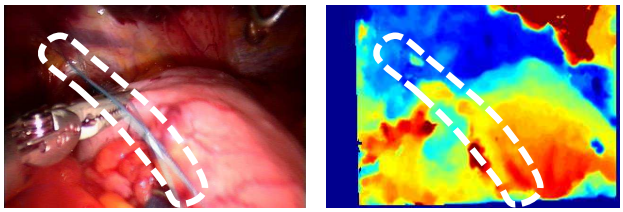


Figure 8: Reconstructing fine structures is both essential and challenging for stereo applications in computer assisted surgery. Even state-of-the-art algorithms [32] suffer from poor reconstruction of sutures and medical instruments.

Figure 9 depicts algorithm performance for the image *ArtL* using *RMS*, *Bad1.0*, *Bad4.0*, and the three proposed metrics for fine structures. Interestingly, the algorithms *SGM*, *LCU*, and *TSGO* rank similar in *RMS* but show very different performance at fine structures. *SGM* achieves the best *BadPix* percentages, the lowest detail fattening and little porosity but it suffers from relatively strong fragmentation. By contrast, *LCU* yields no fragmentation at all and yields good performance at all the other metrics. Hence, *SGM* would be the best choice for applications which are robust against fragmentation of fine structures whereas *LCU* would be the better overall choice. It is further interesting to note that algorithms like *SGBM1* and *LPS* have a much higher *RMS* on the *ArtL* image but are among the best performing algorithms for sampling fine structures.

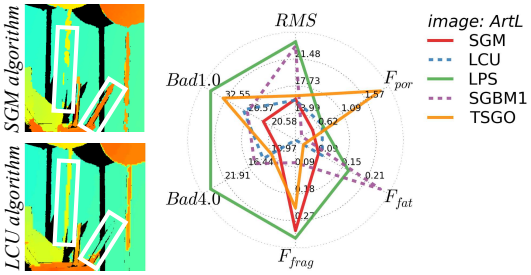


Figure 9: With radar charts, multiple performance dimensions can be evaluated at the same time. *SGM* and *LCU* yield very similar *RMS* scores (lower values in the center are better). Yet, as shown on the disparity maps and quantified by F_{frag} and F_{por} on the chart, *LCU* features less fragmentation whereas *SGM* yields better structure sampling.

4.4. Orthogonality of Metrics

To evaluate whether our metrics are complementary to the existing metrics, we check to what extent the metrics are mutually correlated on the 12 training images and 13 algorithm results. Figure 10 plots algorithm performance with different metrics against each other. The transparency and direction of the lines denote the degree and orientation of linear correlations. We use the *Jadeplant* image as it features discontinuities, fine structures, and planar surfaces. For a more comprehensive evaluation with all images and algorithms we refer to Figure 6 in the supplemental material. The top row of Figure 10 shows that algorithm performance measured by *RMS* and *BadPix* is correlated on the *Jadeplant* image. By contrast, our metrics show little correlation with the *RMS*. The table in Figure 11 further denotes r^2, the coefficient of determination, for each metric paired with *RMS*, *Bad1.0* and three of our metrics, computed on the full dataset. Most of our metrics are highly uncorrelated. The higher correlation between F_{fat} and D_{fat} is acceptable as both metrics measure similar stereo inaccuracies but are justified by having different scopes.

As a further experiment, we compute *RMS* scores on each pixel subset of the metrics in order to separately test the influence of our subset selection and of the metric functions applied to these sets. Rankings based on the subset *RMS* scores are more similar to those defined by our metrics yet not identical[5]. We conclude that it is the combination of subset selection and metric function that makes our metrics specific about their meaning. This is nicely illustrated by the stereo results in Figure 5. Understandably, the *RMS* is more specific about fine structure performance when being applied only to pixels at fine structures. Yet, the expressiveness of metrics which incorporate spatial pixel distributions such as F_{frag} cannot be achieved by the *RMS* metric.

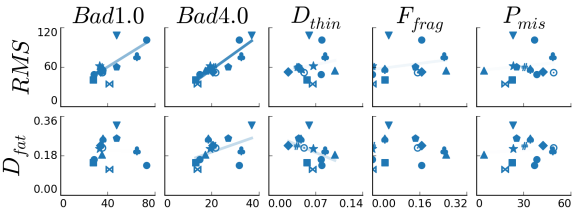


Figure 10: *RMS* scores on the *Jadeplant* image are more correlated with *BadPix* than with our more specific metrics. (Algorithms at the lower left corner perform best).

	Bad1.0	*Bad4.0*	D_{fat}	D_{thin}	D_{fuz}	F_{por}	F_{frag}	F_{fat}	P_{bump}	P_{dist}	P_{mis}
RMS	0.14	0.26	0.04	0.08	0.71	0.00	0.00	0.01	0.14	0.04	0.25
Bad1.0	-	0.66	0.14	0.05	0.03	0.05	0.07	0.35	0.06	0.10	0.26
D_{fat}	0.14	0.35	-	0.03	0.00	0.07	0.12	0.73	0.10	0.10	0.07
F_{por}	0.05	0.06	0.07	0.01	0.00	-	0.62	0.13	0.09	0.02	0.00
P_{bump}	0.06	0.27	0.10	0.05	0.14	0.09	0.00	0.02	-	0.68	0.61

Figure 11: The coefficient of determination for linear fits between scores across images and algorithms is very low for most pairs of metrics. The pairs ($Bad1.0$, $Bad4.0$) and (D_{fat}, F_{fat}) seem to be correlated on the Middlebury data.

4.5. Limitations

We identified two limitations of our proposed evaluation framework. First, our metrics are not homogenously normalized. Just like for the *RMS*, this is not an issue by itself. Yet, in combination, the different ranges make it difficult to get a good grasp of the relative differences between algorithms across multiple metrics. To address this issue, we provide heuristic score distributions in Figure 12. With these histograms, individual metric scores can be put in context when assessing algorithm performance. As a further solution, our metrics could be rewritten to denote respective percentages of bad pixels, e.g. the percentage of surface normals which are off by more than $5°$.

[5]Quantitative results on all images and algorithms are provided in Section 3.2. and the second column in Figure 6 of the supplemental material.

As a second limitation, our pixel extraction methods are not completely parameter-free. We publish our source code such that our results can be reproduced and comparable metric scores can be computed for further disparity maps.

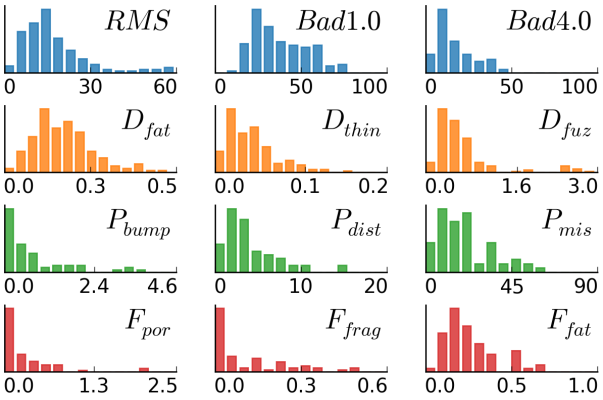


Figure 12: The histograms illustrate the relative distributions of metric scores on the Middlebury dataset. With these scores, individual stereo results can be evaluated in context.

5. Further Benefits and Applications

Beyond the assessment of algorithm performance on academic benchmark datasets, our geometry-based evaluation also supports blackbox tuning of algorithm parameterization and makes performance evaluation more tolerant against dataset bias and ground truth deficiencies.

Parameter tuning of stereo algorithms often is a difficult and rather subjective process, all the more if the respective implementation details are inaccessible. Combined with a coarse parameter sweep, our metrics can be used for application-specific parameter optimization.

Even carefully composed datasets are not perfectly representative for the proportions of ordinary and variously challenging pixels on test images. For instance, common benchmarks feature large areas with flat objects and therefore tend to favor smooth disparity maps. By computing metrics for specific, semantically meaningful image regions, our approach avoids to disadvantage algorithms which perform well on less frequent yet equally important regions such as fine structures. As a second issue, the term "representative" is highly application-specific which lead to specialized benchmarks such as KITTI [10]. Our approach allows to generalize and re-combine multiple datasets. Algorithm performance on different image regions may be composed such that it complies with the proportions and priorities of a given application.

Our proposed metrics may further be applied to data with missing GT disparities. For example, the metric for surface bumpiness can be computed on disparity maps with roughly segmented planar image regions.

6. Conclusion and Outlook

We proposed and carefully justified the *HCI Stereo Metrics*: nine semantically intuitive performance measures for three geometric categories of stereo ground truth. Our metrics can be applied to various benchmark datasets and to dense algorithm results generated by two-frame stereo or other reconstruction methods. By combining our proposed metrics, automated benchmarks or parameter tunings can be carried out taking into account a variety of application-specific requirements.

The presented metrics and evaluation methods are available online[6]. On this interactive benchmark website, researchers and engineers may thoroughly assess and compare state-of-the-art stereo algorithms. We thereby hope to help engineers identify "their best" stereo algorithm and to foster progress in those stereo applications where existing methods still yield insufficient quality.

Future work will focus on performance with respect to radiometric challenges such as specular highlights or transparency. These concepts will further be applied to optical flow and multi-view stereo evaluation.

References

[1] S. T. Barnard and M. A. Fischler. Computational stereo. *ACM Computing Surveys (CSUR)*, 14(4):553–572, 1982. 2

[2] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo-stereo matching with slanted support windows. In *BMVC*, volume 11, pages 1–11, 2011. 1

[3] M. Bleyer, C. Rhemann, and C. Rother. Extracting 3d scene-consistent object proposals and depth from stereo images. In *ECCV 2012*, pages 467–481. Springer, 2012. 1

[4] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha. Object stereo - joint stereo matching and object segmentation. In *CVPR 2011*, pages 3081–3088. IEEE, 2011. 1

[5] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV 2012*, pages 611–625. Springer, 2012. 1

[6] T. R. dos Santos, A. Seitel, H.-P. Meinzer, and L. Maier-Hein. Correspondences search for surface-based intraoperative registration. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 660–667. Springer, 2010. 6

[7] N. Einecke and J. Eggert. A two-stage correlation method for stereoscopic depth estimation. In *DICTA*, pages 227–234. IEEE, 2010. 5

[8] W. Förstner. Diagnostics and performance evaluation in computer vision. In *Performance versus Methodology in Computer Vision, NSF/ARPA Workshop Seattle*, pages 11–25, 1994. 2

[9] J. Fritsch, T. Kuehnl, and A. Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems*, volume 28, pages 38–61, 2013. 2

[6] http://hci.iwr.uni-heidelberg.de/stereometrics

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR 2012*, pages 3354–3361. IEEE, 2012. 1, 2, 8

[11] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *ACCV 2010*, pages 25–38. Springer, 2011. 5

[12] F. Güney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *CVPR 2015*, pages 4165–4175, 2015. 1

[13] R. M. Haralick. Performance characterization in computer vision. In *Computer Analysis of Images and Patterns*, pages 1–9. Springer, 1993. 2

[14] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *PAMI*, 30(2):328–341, 2008. 5

[15] H. Hirschmüller, P. R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3):229–246, 2002. 3, 5

[16] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *CVPR 2004*, volume 1, pages I–74. IEEE, 2004. 1, 4

[17] C. G. Keller, M. Enzweiler, and D. M. Gavrila. A new benchmark for stereo-based pedestrian detection. In *Intelligent Vehicles Symposium*, pages 691–696. IEEE, 2011. 2

[18] W. S. Kim, A. I. Ansar, R. D. Steele, and R. C. Steinke. Performance analysis and validation of a stereo vision system. In *Systems, Man and Cybernetics*, volume 2, pages 1409–1416. IEEE, 2005. 2

[19] D. Kondermann, S. Abraham, G. Brostow, W. Förstner, S. Gehrig, A. Imiya, B. Jähne, F. Klose, M. Magnor, H. Mayer, et al. On performance analysis of optical flow algorithms. In *Outdoor and Large-Scale Real-World Scene Analysis*, pages 329–355. Springer, 2012. 2

[20] D. Kondermann, R. Nair, S. Meister, W. Mischler, B. Güssefeld, K. Honauer, S. Hofmann, C. Brenner, and B. Jähne. Stereo ground truth with error bars. In *ACCV 2014*, pages 595–610. Springer International Publishing, 2015. 1

[21] J. Kostková, J. Čech, and R. Šára. Dense stereomatching algorithm performance for view prediction and structure reconstruction. In *Image Analysis*, pages 101–107. Springer, 2003. 2

[22] J. Kowalczuk, E. T. Psota, and L. C. Perez. Real-time stereo matching on cuda using an iterative refinement method for adaptive support-weight correspondences. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):94–104, 2013. 5

[23] L. Maier-Hein, A. Groch, A. Bartoli, S. Bodenstedt, G. Boissonnat, P. L. Chang, N. T. Clancy, D. S. Elson, S. Haase, E. Heim, J. Hornegger, P. Jannin, H. Kenngott, T. Kilgus, B. Muller-Stich, D. Oladokun, S. Rohl, T. R. Dos Santos, H. P. Schlemmer, A. Seitel, S. Speidel, M. Wagner, and D. Stoyanov. Comparative validation of single-shot optical techniques for laparoscopic 3-d surface reconstruction. *IEEE Trans Med Imaging*, 33(10):1913–1930, Oct 2014. 2, 6

[24] L. Maier-Hein, P. Mountney, A. Bartoli, H. Elhawary, D. Elson, A. Groch, A. Kolb, M. Rodrigues, J. Sorger, S. Speidel, et al. Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery. *Medical image analysis*, 17(8):974–996, 2013. 1, 2

[25] M. Maimone and S. A. Shafer. A taxonomy for stereo computer vision experiments. In *ECCV workshop on performance characteristics of vision algorithms*, pages 59–79. Citeseer, 1996. 2

[26] R. Margolin, L. Zelnik-Manor, and A. Tal. How to evaluate foreground maps. In *CVPR 2014*, pages 248–255. IEEE, 2014. 2

[27] N. Morales, G. Camellini, M. Felisa, P. Grisleri, and P. Zani. Performance analysis of stereo reconstruction algorithms. In *Procs. IEEE Intl. Conf. on Intelligent Transportation Systems*, pages 1298–1303, 2013. 2

[28] M. G. Mozerov and J. van de Weijer. Accurate stereo matching by two-step energy minimization. *IEEE Transactions on Image Processing*, 24(3):1153–1163, 2015. 5

[29] J. Mulligan, V. Isler, and K. Daniilidis. Performance evaluation of stereo for tele-presence. In *ICCV 2001*, volume 2, pages 558–565. IEEE, 2001. 2

[30] B. Özdemir, S. Aksoy, S. Eckert, M. Pesaresi, and D. Ehrlich. Performance measures for object detection evaluation. *Pattern Recognition Letters*, 31(10):1128–1137, 2010. 2

[31] D. Pfeiffer, S. Gehrig, and N. Schneider. Exploiting the power of stereo confidences. In *CVPR 2013*, pages 297–304. IEEE, 2013. 1, 2

[32] S. Röhl, S. Bodenstedt, S. Suwelack, H. Kenngott, B. P. Müller-Stich, R. Dillmann, and S. Speidel. Dense GPU-enhanced surface reconstruction from stereo endoscopic images for intraoperative registration. *Med Phys*, 39(3):1632–1645, 2012. 6

[33] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition*, pages 31–42. Springer, 2014. 1, 2

[34] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47:7–42, 2002. 1, 2

[35] S. N. Sinha, D. Scharstein, and R. Szeliski. Efficient high-resolution stereo matching using local plane sweeps. In *CVPR 2014*, pages 1582–1589. IEEE, 2014. 4, 5

[36] C. Stentoumis, L. Grammatikopoulos, I. Kalisperakis, and G. Karras. On accurate dense stereo-matching using a local adaptive multi-cost approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 91:29–49, 2014. 5

[37] P. L. Venetianer and H. Deng. Performance evaluation of an intelligent video surveillance system–a case study. *Computer Vision and Image Understanding*, 114:1292–1302, 2010. 2

[38] W. A. Yasnoff, J. K. Mui, and J. W. Bacus. Error measures for scene segmentation. *Pattern recognition*, 9(4):217–231, 1977. 2

[39] C. Zhang, Z. Li, R. Cai, H. Chao, and Y. Rui. As-rigid-as-possible stereo under second order smoothness priors. In *ECCV 2014*, pages 112–126. Springer, 2014. 1, 4

[40] K. Zhang, J. Li, Y. Li, W. Hu, L. Sun, and S. Yang. Binary stereo matching. In *ICPR 2012*, pages 356–359. IEEE, 2012. 5