# High-Resolution Stereo Matching based on Sampled Photoconsistency Computation

Chloe LeGendre[1]
legendre@ict.usc.edu
Konstantinos Batsos[2]
kbatsos@stevens.edu
Philippos Mordohai[2]
mordohai@cs.stevens.edu

[1] Institute for Creative Technologies
University of Southern California
Los Angeles, CA, USA

[2] Stevens Institute of Technology
Hoboken, NJ, USA

**Abstract**

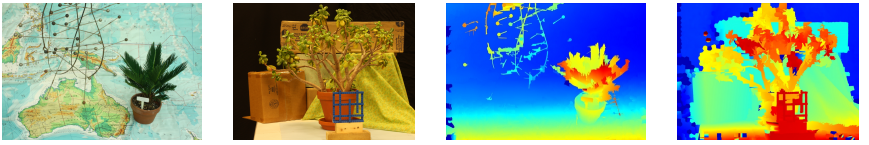
We propose an approach to binocular stereo that avoids exhaustive photoconsistency computations at every pixel, since they are redundant and computationally expensive, especially for high resolution images. We argue that developing scalable stereo algorithms is critical as image resolution is expected to continue increasing rapidly. Our approach relies on oversegmentation of the images into superpixels, followed by photoconsistency computation for only a random subset of the pixels of each superpixel. This generates sparse reconstructed points which are used to fit planes. Plane hypotheses are propagated among neighboring superpixels, and they are evaluated at each superpixel by selecting a random subset of pixels on which to aggregate photoconsistency scores for the competing planes. We performed extensive tests to characterize the performance of this algorithm in terms of accuracy and speed on the full-resolution stereo pairs of the 2014 Middlebury benchmark that contains up to 6-megapixel images. Our results show that very large computational savings can be achieved at a small loss of accuracy. A multi-threaded implementation of our method [1] is faster than other methods that achieve similar accuracy and thus it provides a useful accuracy-speed tradeoff.

## 1 Introduction

An inspection of the relevant benchmarks [14, 25, 27, 30] shows that substantial progress has been made in stereo matching, and the effects of many of the inherent challenges, such as lack of texture, repetitive patterns and occlusion, have been mitigated to a large degree. A challenge that has yet not been fully addressed is that of estimating disparity maps of even moderately high resolution images. While the accuracy of modern approaches on these data is remarkable, the resolution of the images is a small fraction of that produced by even the cheapest current cameras. The images of the first two versions of the Middlebury Stereo Evaluation benchmark [25] do not exceed 200,000 pixels, those of the KITTI stereo benchmark [14] are roughly 0.5 megapixels, and the multi-view stereo benchmark of Seitz et al. [27] contains 640×480 images. The resolution of the images in the inactive benchmark of Strecha et al. [30] is 6 megapixels, but virtually all authors have downsampled them.



[1] Available at https://github.com/chloelle/SPS



(a) (b) (c) (d)

Figure 1: (a,b) Left image of the Australia and Jadeplant stereo pair, part of the Middlebury test and training set [26] respectively. (c,d) The disparity maps computed by our algorithm.

Recently, the third version of the Middlebury Stereo Evaluation benchmark was released [26]. It contains mostly 6 megapixel images and allows processing at three resolutions: full (F), half (H) and quarter (Q). The images are more challenging and realistic than those of previous Middlebury benchmarks due to miscalibration, exposure and illumination variations, the materials depicted and the complexity of the scenes. Results of Semi-Global Matching (SGM) [16] on all resolutions are provided as a baseline. This leads to a surprising observation: the percentage of pixels with a disparity error greater than 2 is: 21.2, 18.7 and 25.3% for Q, H and F resolution, respectively. Counter-intuitively, SGM achieves the lowest accuracy at the highest resolution. In fact, most top performing results are in H resolution. To us, this is an indication of the lack of effective high-resolution stereo algorithms. We speculate that this is partially due to miscalibration effects that are more pronounced at higher resolutions and that most methods are tuned for lower resolution images.

A bottleneck for the majority of conventional stereo methods is the computation of a photoconsistency measure for all possible disparity assignments to all pixels. (The number of computations grows cubically with the width of the image, since the disparity range must grow with image resolution for a fixed depth range.) This is true for both local methods that rely mainly on the photoconsistency measure to assign disparities to pixels and for global methods that use these computations for the data terms of their objective functions. Storing the photoconsistency (matching cost) volume is an additional bottleneck for methods that require it. For example, the cost volume for the Vintage stereo pair [26] is over 16 GB at single or integer precision (for $2912\times1924$ pixels and 740 possible disparities per pixel).

Many of the best-performing methods use deep learning and are applied on H resolution images. The most representative among them is the MC-CNN approach [38, 39]. It requires 12 GB of GPU memory, which is the maximum that is currently available. Applying MC-CNN at full-resolution would take a GPU with roughly 100 GB of memory or new techniques for splitting the computation. Image resolution is likely to continue growing faster than storage and computational requirements. In order for stereo matching to be applicable to high resolution imagery, computation and storage requirements must be drastically reduced.

Algorithms that do not perform exhaustive photoconsistency computations have been reported in the literature (see Section 2). PatchMatch stereo [8] is based on a random search and propagation scheme that is much faster than the naive baseline of considering every possible plane orientation at every possible depth for each pixel, but it is not a fast algorithm overall. This is due to the increased dimensionality of the search space and the fact that *all pixels are visited* during propagation. Algorithms that do not visit all pixels have also been proposed. Geiger et al. [13] compute the photoconsistency of pixels on a regular grid and connect them via Delaunay triangulation to approximate the surface. Sinha et al. [29] match interest points and use them to estimate planes which in turn define multiple local plane sweep problems that require photoconsistency computations only locally around each plane. Compared to these approaches, the use of superpixels allows our algorithm to reason

on regions that are very likely to respect object boundaries. Wang et al. [34] eliminate uninformative photoconsistency computations by testing whether the true disparity is likely to have already been found. After the disparity range of each pixel has been reduced to a small subset, however, these algorithms [13, 29, 34] still evaluate disparity hypotheses at *every pixel*.

We propose to significantly reduce photoconsistency computations with a sampling routine, and we analyze the impact of such sampling on the accuracy of the final disparity maps. Our superpixel-based stereo algorithm initially fits planes to superpixels and subsequently propagates plane hypotheses among neighboring superpixels. The algorithm requires photoconsistency computations in two stages: for the generation of pixel-wise disparity estimates that are used to fit planes to the superpixels and for assessing the quality of different plane hypotheses for each superpixel. Both of these steps involve a lot of redundant computations, since nearby pixels often have similar, but not identical, disparities. We claim that *performing a fraction of these computations is sufficient* to achieve accuracy close to that of an exhaustive version of the algorithm, at a smaller computational cost. Errors due to incorrect plane estimation or lack of texture are corrected in the plane propagation stage. Since propagation occurs on the superpixel adjacency graph, plane hypotheses reach much longer ranges than PatchMatch stereo [8] and other similar algorithms. Figure 1 shows two images of the test set and the disparity maps generated by our algorithm.

In summary, the contributions of this paper are:

- an approach to stereo matching that does not rely on exhaustive photoconsistency computation for either the hypothesis generation or the hypothesis evaluation step,
- regularization and long range hypothesis propagation through the use of superpixels, which are very likely to respect image boundaries,
- a flexible approach that can be parameterized to achieve a speed-accuracy tradeoff in accordance to a user-specified utility function and is also highly parallelizable enabling a fast implementation that obtains competitive results on the Middlebury 2014 benchmark.

## 2 Related work

In this section, we review approaches to stereo matching that do not require exhaustive photoconsistency computations. PatchMatch stereo [8] caused a paradigm shift by showing that it is possible to obtain state of the art accuracy relying on randomized search and propagation. Algorithms inspired by PatchMatch include the work of Heise et al. [15], who integrate the PatchMatch concept in a variational formulation that explicitly regularizes disparity and normal gradients, instead of smoothing them. Besse et al. [4] propose to use PatchMatch to compute the unary terms of an energy function optimized using particle belief propagation (PBP). PatchMatch-based algorithms [4, 8, 15] compute photoconsistency for all pixels, but not exhaustively since this is impractical due to the increased dimensionality of the search space that includes surface normals in addition to disparity.

Wang et al. [33] observe that it is redundant to evaluate all disparity candidates for all pixels due to the smoothness of natural images. Their approach relies on bilateral filtering for propagating disparities from seeds to their neighborhoods and is thus limited to piecewise constant disparity maps. This work was extended by integrating a sequential probability ratio test to determine when the sampling of photoconsistency is sufficient and by modeling slanted surfaces [34]. Propagation, however, is still local, between neighboring pixels.

Other methods restrict initial processing to a subset of the pixels that are used to guide disparity estimation for their neighbors in a deterministic way. ELAS [13] explores the full disparity range of pixels on the vertices of a regular grid. Delaunay triangulation is applied on these support points, and the resulting triangles cover all pixels. Sinha et al. [29] propose the Local Plane Sweep (LPS) algorithm that clusters matched interest points to form disparity plane hypotheses. Local plane sweep problems are then defined around each plane and solved by a generalized version of SGM [16].

Superpixel or segmentation-based approaches to stereo matching are also relevant to our research. The paper of Birchfield and Tomasi [5] is a milestone since it relaxed the fronto-parallel assumption and proposed a practical algorithm for global optimization with plane memberships as labels. This work was extended to non-planar segments by Lin and Tomasi [20]. Segmentation-based approaches have also been very successful on the previous [6, 7, 9, 17, 31, 35] and current [19, 40] Middlebury benchmarks, as well as the KITTI benchmark [10, 32, 36, 37], but they visit all pixels in all iterations. Recently, Duan and Lafarge [12] addressed 3D reconstruction from satellite imagery relying on segmentation and priors on buildings.

A different class of algorithms that generates piecewise planar reconstruction are those that use the Total Generalized Variation (TGV) as the regularization term [18, 23, 24]. As opposed to minimizing total variation which favors piecewise constant solutions, minimizing TGV of order 2, along with a data fidelity term, leads to piecewise affine solutions.

# 3 Method

The main steps of our method are the following. We consider the first two preprocessing.

1. Vertical alignment of the images to correct residual calibration errors,
2. Oversegmentation of the reference image into superpixels using SLIC [2],
3. Photoconsistency estimation for a random sample $S$ of the pixels within each superpixel,
4. Plane-fitting for each superpixel using RANSAC,
5. Propagation of planes among neighboring superpixels for $n$ iterations, evaluating photoconsistency on a random sample $V$ of the pixels.

## 3.1 Vertical Alignment

Based on the observation of Scharstein et al. [26] that conventional calibration procedures are likely to leave some residual vertical disparities in high resolution stereo pairs, we refine the rectification by estimating a transformation as in [29]. We extract upright SURF features [3] in both images and detect potential correspondences. We then use RANSAC to fit a global linear model of vertical displacement $d_y(x,y) = ay + b$. This is estimated for each stereo pair and leads to small refinements of the rectification. While it is visually imperceptible, we verified experimentally that a misalignment of the epipolar lines by even one pixel makes a noticeable difference in the accuracy of stereo matching.

## 3.2 Superpixel Segmentation

We segment the left image of the stereo pair, which will be used as the reference image throughout, into superpixels using the SLIC algorithm of Achanta et al. [2]. SLIC is a fast adaptation of k-means for image oversegmentation in which the pixel dissimilarity metric depends on color and image coordinate distances. The relative weights of the two distances can

be adjusted to obtain more or less compact superpixels. SLIC has two additional parameters that control the average superpixel size and the size in pixels below which a connected component is eliminated in postprocessing. We have evaluated the effects of the parameters of SLIC on our algorithm, but have done so non-exhaustively since we consider segmentation as preprocessing. The main objective is to ensure an oversegmentation, so that superpixels do not straddle multiple surfaces, while keeping the number of superpixels manageable. To avoid handling the small blank areas that may appear after vertical alignment, we segment the original images and warp the segment labels according to the aligning transformation.

## 3.3 Sparse Photoconsistency Estimation and Plane-Fitting

Our research is motivated by the observation that exhaustive photoconsistency computation is redundant since, in most cases, neighboring pixels have similar disparities and similar local maxima in their photoconsistency curves. Since we approximate the superpixels with planar surfaces, a small number of correctly reconstructed points is sufficient to estimate each plane. Of course, relying on more points is advantageous as it increases robustness against wrong matches and allows a more precise fit based on all inliers after RANSAC. This tradeoff between sampling density and accuracy is our focus in this paper.

We select pixels for which to estimate photoconsistency by randomly choosing a fraction $s$ of all pixels in each superpixel. For the selected pixels $S$, we evaluate the photoconsistency of all disparity candidates using normalized cross correlation (NCC). We apply a simple Winner-Take-All (WTA) disparity selection on the sampled pixels and assign to them the disparity with the maximum NCC value.

Given a sparse set of points in $(x, y, d)$ space, we fit a plane using RANSAC. We enforce the disparity gradient limit [22] and reject plane hypotheses that have a disparity gradient larger than one with respect to either camera, and those that cause an inversion of pixel ordering between the two images. The hypothesis with the most inliers, defined as reconstructed points whose distance to the plane is at most one disparity level, is selected. The final plane equation is assigned to the superpixel after applying SVD to all inliers to obtain the plane normal. We tried re-using the matched SURF features from Section 3.1 to augment the set of reconstructed points, but no benefits were observed, so this step was eliminated for simplicity. Figure 3 (c) shows the disparity map for MotorcycleE after plane-fitting. Notice that many superpixels are visually correct, but others appear to be entirely wrong.

## 3.4 Plane Propagation

Most pixels in the disparity maps generated after plane fitting have correct disparities. Errors are mostly due to occlusion or lack of texture. While the former is partially mitigated if the superpixels have a sufficiently large visible part in which pixels can be matched accurately, it is clear that some form of regularization would be beneficial.

We implemented regularization in the form of plane propagation where each superpixel receives plane hypotheses from its neighboring superpixels. To facilitate this step, we record the superpixel adjacency graph and propagate planes among superpixels connected by edges in the graph. Propagation starts from left to right, then right to left, top to bottom and finally bottom to top. Edges in the graph are assigned to the nearest of the four directions, based on the relative location of the superpixel centroids. We define an iteration as a full cycle over the four propagation directions. Each superpixel collects on average six plane hypotheses at each iteration and compares their photoconsistency with that of its current plane. Due to the

size of the superpixels, plane hypotheses can be propagated much faster from one part of the image to another compared to pixel-based algorithms, such as PatchMatch Stereo.

Following a similar argument as that in Section 3.3, evaluating the photoconsistency of plane hypotheses on all pixels is redundant. Therefore, we propose to restrict evaluation only to a sampled subset of pixels. Since only one disparity per pixel for each plane hypotheses must be evaluated, this step is much cheaper computationally than the previous one. Due to this difference and to avoid biasing the entire solution based on the previous sample, we select a different subset of pixels $V$ for these computations at a new rate $v$.

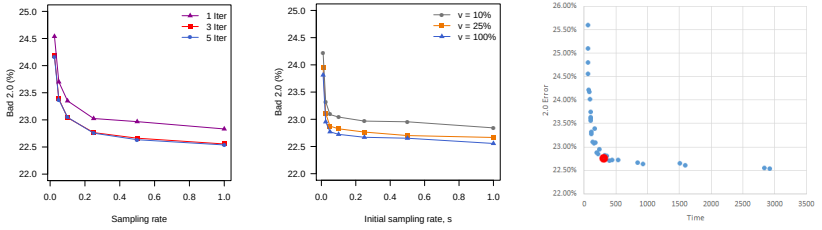
First, for the subset of pixels $V$, we aggregate the NCC scores for the initial plane estimate produced from RANSAC. Then, at each iteration, all adjacent plane candidates that are propagated to a superpixel are scored by accumulating new NCC scores for the subset $V$. NCC windows are smaller here than the initial photoconsistency estimation step. This was determined experimentally. If the new plane hypothesis is more photoconsistent than the current one, it replaces it. (Relying on the number of plane inliers to score plane hypotheses led to inferior results.) Propagation is performed for $n$ iterations, with $n$ being one of the main parameters of our algorithm along with the sampling rates $s$ and $v$. For the same values of $v$ and $s$, the plane propagation step is cheaper computationally since on average only six plane hypotheses have to be evaluated as opposed to the entire disparity range.

# 4 Experimental Results

We perform a comprehensive set of experiments on Version 3 of the Middlebury Stereo Evaluation [26] using only the full resolution images. As noted in the introduction, it is possible to achieve higher accuracy on half-resolution images using sophisticated optimization techniques. Our focus in this paper, however, is on high resolution data. Version 3 of the benchmark consists of a training set of 15 stereo pairs with publicly available ground truth and a test set of 15 stereo pairs, the ground truth for which has not been released. The image resolution varies between 1.5 and 5.9 megapixels with an average of 5.2 megapixels and the disparity range varies between 256 and 800 with an average of 387. These data are available at three resolutions: full (F), half (H) and quarter (Q). This version of the benchmark is more challenging because most stereo pairs have imperfect rectification, except those with a suffix 'P' in their filename, while several others contain images taken under different exposure or lighting, denoted by 'E' and 'L' respectively [26]. The ranking in the new tables is determined by weighted averages of the selected metric. (We explicitly state whether the tables below show weighted or un-weighted averages.) The default error metric is the percentage of "bad" pixels with disparity errors above a certain threshold.

The following parameters were held constant for the results shown in this section. The upright SURF features were used with the default OpenCV parameters. NCC was computed on grayscale versions of the images, in $15 \times 15$ windows in the initial photoconsistency estimation phase and in $5 \times 5$ windows during plane propagation. For SLIC, region size (distance between adjacent segment centroids) was set to 60, the regularization parameter to 200 and the minimum region size to 800 pixels. The objective of our experiments was to assess the speed-accuracy tradeoff as a function of three parameters:

- the sampling rate $s$ for the initial plane fitting step,
- the sampling rate $v$ for comparing propagated plane hypotheses, and
- the number of propagation iterations $n$.




(a) (b) (c)

Figure 2: (a) Error rate as a function of sampling rate, keeping $s$ and $v$ equal. The purple, red and blue curves correspond to 1, 3 and 5 propagation iterations respectively. (The latter two are almost indistinguishable.) Results are averages over 10 trials. (b) Error rate as a function of the initial sampling rate $s$ for $n$=3 iterations. The blue, orange and gray curves correspond to $v$ being 100%, 25% and 10% respectively. Results are averages over 10 trials for each sampling rate. (c) Error rate over execution time (in sec) for various settings of the parameters. The red point corresponds to the settings used for the submitted results.

| Sampling rate ($s$/$v$) | 2.5% | 5% | 10% | 25% | 100% |
|---|---|---|---|---|---|
| NCC-RANSAC time (sec) | 66.7 | 133.0 | 265.6 | 663.4 | 2661.6 |
| Plane propagation time (sec) | 1.7 | 3.0 | 5.7 | 13.8 | 54.8 |

Table 1: Average execution times on the training set for NCC computation and plane-fitting using RANSAC (Section 3.3) and for one iteration of plane propagation and selection (Section 3.4) for different values of the corresponding sampling rate, $s$ or $v$. The execution time of the relevant part of the *single-threaded* code scales approximately linearly with $s$, $v$ and $n$. An overhead of 24.8 sec, that includes the time needed for SLIC, should be added to the above times to obtain the total execution time.

## 4.1 Speed-Accuracy Analysis

We begin with results on the effects of the three parameters on the accuracy of the algorithm, while also aiming to reveal potential redundancies and opportunities for accuracy improvements at low computational cost. All results in this subsection are *un-weighted averages over the training set with the error threshold set at 2.0 pixels.* Execution times are for a single-threaded C++ implementation on an Intel Core i7-4900MQ processor at 2.8GHz with 16 GB of RAM. A faster implementation will be discussed later.

Figure 2 (a) shows how the error varies for a fixed number of propagation iterations $n$ as the sampling rates $s$ and $v$, which are kept equal to each other in these tests, vary. Not surprisingly, more iterations are beneficial but diminishing returns are observed. Execution time grows approximately linearly with the number of iterations for a given $v$. Knowledge of the expected improvements due to each additional iteration on the training set allows the user to make an informed decision on the value of the parameters that fit a particular task.

Next, we show error rates as both $s$ and $v$ vary, while the number of iterations is set to 3 in Figure 2 (b). A conclusion that can be drawn from these results is that setting $s$ to very small values leads to poor accuracy regardless of $v$. Table 1 shows timing results of the various steps of our algorithm using the single-threaded C++ implementation. Its purpose is to show how much can be saved by sampling photoconsistency computations.

Given data on accuracy and execution time, a fraction of which has been presented above, users can select parameter values that meet their needs. What is missing is either a constraint on error rate or processing time or a utility function that combines the two. The Middlebury

| Algorithm | $NCC_5$ | $NCC_5$ | $NCC_{11}$ | $NCC_{15}$ | $SPS_{100}$, $n$=0 | $SPS_{100}$, $n$=3 | SPS, $n$=0 | SPS, $n$=3 |
|---|---|---|---|---|---|---|---|---|
| bad 1.0 | 72.6 | 52.7 | 48.1 | 43.5 | 35.5 | 32.5 | 36.8 | 33.0 |
| bad 2.0 | 68.6 | 47.1 | 42.0 | 36.5 | 26.0 | 22.6 | 27.2 | 22.9 |

Table 2: *Un-weighted* average error rates on the Middlebury training set at two values of the threshold. SPS is our algorithm using $s$=5%, $v$=25% and $n$=3, while $SPS_{100}$ uses $s$=100%, $v$=100% and $n$=3. Run times for *single-threaded* implementations of NCC in $15 \times 15$ windows, $SPS_{100}$ and SPS are 2635.2, 2834.3 and 194.9 sec respectively. The latter shows that a 14-fold reduction in the number of operations only results in a minimal loss of accuracy.

| Algorithm | Dense | Training dense | | | Test dense | | |
|---|---|---|---|---|---|---|---|
| | | bad 1.0 | bad 2.0 | time/MP | bad 1.0 | bad 2.0 | time/MP |
| SPS | y | **30.0** | 21.1 | 4.33 | 29.1 | **19.6** | 4.77 |
| LPS [29] | y | 33.7 | 26.2 | 7.14 | **27.6** | 20.3 | 5.28 |
| SGM [16] | n | 31.7 | 22.1 | 10.3 | 35.8 | 25.3 | 13.4 |
| SGBM11 [1] | n | 36.5 | 27.4 | 2.79 | 38.0 | 28.4 | 3.69 |
| PFS [11] | y | 30.3 | **19.9** | 5.66 | 43.0 | 32.2 | 10.2 |
| ELAS [13] | n | 38.5 | 26.6 | 0.56 | 44.4 | 32.3 | 0.56 |
| TSGO [21] | y | 55.0 | 31.3 | 11.4 | 55.9 | 39.1 | 8.26 |
| ICSG [28] | n | 47.9 | 37.7 | 31.9 | 55.2 | 45.6 | 36.2 |

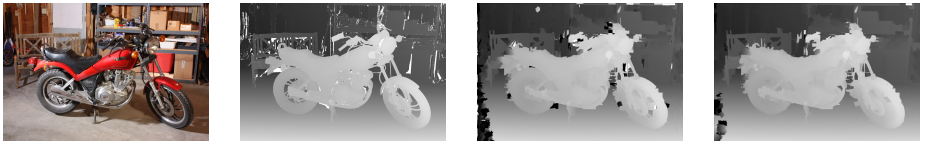
Table 3: *Weighted* average error rates on the full-resolution Middlebury data. The table includes only published full-resolution results and ours. It is sorted according to bad 2.0 on the test set, the default criterion. Our algorithm is the fastest in terms of time per megapixel among the dense algorithms that produce disparities for all pixels.

benchmark does not provide any of these additional bits of information; the most common goal is high accuracy. We feel, however, that setting $s$ and $v$ to 100% and performing several propagation iterations is against the spirit of our algorithm. In Figure 2 (c), we plot the error rate over execution time for several settings of the three parameters approximating the frontier points of our algorithm's ROC curve. Since we explicitly wanted to avoid arbitrarily setting the cost of each additional second of run time in terms of a number of bad pixels, we applied several other criteria to select the operating point. These included finding the point that is nearest to the origin after normalizing time and error rate by their standard deviation, with and without trimming extreme values, and picking the point with the highest f-score. The majority of these tests selected the point that is marked in red in the figure, which corresponds to $s$=5%, $v$=25% and $n$=3. The choice of a large value for $v$ is not surprising, since plane evaluation is computationally cheap and leads to accuracy gains. These parameters were used for our submission to the benchmark. Note that this is neither the most accurate nor the fastest configuration of our algorithm.

## 4.2 Middlebury Evaluation Results

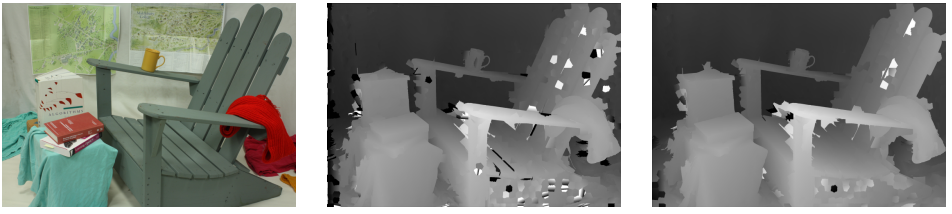
Here, we compare our results on the dense training and test datasets of the new benchmark with those of baseline algorithms, namely Winner-Take-All NCC and our algorithm with fully dense sampling, as well as with submitted results by seven other algorithms. All algorithms implemented by us have been applied after calibration refinement (Section 3.1). We will refer to our algorithm as the **Sampled-Photoconsistency Stereo (SPS)** algorithm and denote by $SPS_{100}$ the variant that performs exhaustive photoconsistency computations and three plane propagation iterations.

A comparison of our algorithm using the selected parameters and the baselines imple-

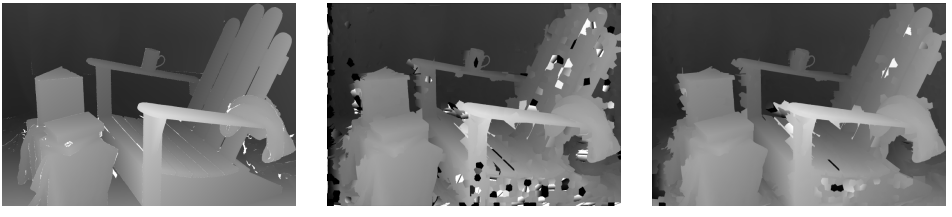


(a) (b) (c) (d)

Figure 3: The MotorcycleE example. (a) Left image. (b) Ground Truth disparity map. (c) The disparity map after plane-fitting, but before plane propagation. (d) The final disparity map after plane propagation.



(a) Left image (b) $SPS_{100}$, $n$=0 (c) $SPS_{100}$, $n$=3



(d) Ground truth (e) SPS, $n$=0 (f) SPS, $n$=3

Figure 4: Comparison of exhaustive photoconsistency computation ($SPS_{100}$) with a sampled version using the selected parameters (SPS). The middle column shows results before plane propagation, while the rightmost column shows the final disparity maps. The error rates at the 2.0 error threshold are: (b) 18.8%, (c) 13.9%, (e) 21.4% and (f) 15.1%. While the low sampling rate in the initial stage leads to errors, plane propagation corrects many of them.

mented by us can be seen in Table 2. NCC is clearly less accurate than the single-threaded SPS, and it is also slower. Three iterations of plane propagation have a large impact on the error rate and enable the faster version of our algorithm to reach the accuracy of the exhaustive one. See Figs. 3 and 4 for example results on two stereo pairs from the training set. Figure 4 includes a comparison of exhaustive and sampled photoconsistency computations, where plane propagation corrects initial errors of the latter to reach approximately the same accuracy at a fraction of the time.

In order to demonstrate the parallelizability of SPS, we accelerated it using the OpenMP framework and vector instructions provided by Intel's MMX. SLIC was also accelerated using the same techniques. Results produced by this version of the algorithm executed on a 6-core Intel Core i7-5820K processor at 3.3 GHz were submitted to the benchmark.

Table 3 shows the *weighted* average error rates at two different thresholds on the full-resolution data of the dense training and test sets. *Among all full-resolution submissions by published methods, our algorithm ranks first or second in accuracy on both the test and training set when the error threshold is 1.0 or 2.0.* (It also ranks no worse than third when the threshold is set to 0.5 or 4.0 on both sets. Please see the evaluation table on the Middlebury website.) Moreover, SPS is the fastest among all dense algorithms that estimate a disparity for every pixel, using total time or time per megapixel as the criterion.

# 5 Conclusion

We have presented an approach that addresses the challenge of balancing accuracy and processing time for stereo matching. We envision users of our approach operating under a time budget and configuring the algorithm to obtain the highest expected accuracy in the allotted time. In other cases, a utility function that takes into account speed and accuracy may be available. Our analysis would allow users to select the operating point in parameter space to maximize their utility. We believe that this is a more flexible and intuitive approach to stereo than turnkey, black box systems.

We focused our efforts on high-resolution images due to the challenges they present to many current stereo methods, both local and global. We strongly believe that processing higher-resolution stereo images is inevitable and the research community is not fully prepared for it. The use of superpixels provides a basis for regularizing the solution and preserves region boundaries, while the piecewise planar approximation holds for small enough superpixels as shown by the high accuracy of our results. Since it is easily parallelizable, a multi-threaded, vectorized implementation of SPS is also the fastest among all dense algorithms evaluated on the full-resolution data.

# References

[1] OpenCV 2.4.8, 2013.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11):2274–2282, 2012.

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *ECCV*, pages 404–417, 2006.

[4] Frederic Besse, Carsten Rother, Andrew Fitzgibbon, and Jan Kautz. PMBP: Patchmatch belief propagation for correspondence field estimation. *IJCV*, 110(1):2–13, 2014.

[5] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, pages 489–495, 1999.

[6] Michael Bleyer and Margrit Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal of Photogrammetry and Remote Sensing*, 59(3):128–150, 2005.

[7] Michael Bleyer, Carsten Rother, and Pushmeet Kohli. Surface stereo with soft segmentation. In *CVPR*, pages 1570–1577, 2010.

[8] Michael Bleyer, Christoph Rhemann, and Carsten Rother. PatchMatch stereo-stereo matching with slanted support windows. In *BMVC*, 2011.

[9] Michael Bleyer, Carsten Rother, Pushmeet Kohli, Daniel Scharstein, and Sudipta Sinha. Object stereo - joint stereo matching and object segmentation. In *CVPR*, pages 3081–3088, 2011.

[10] Ayan Chakrabarti, Ying Xiong, Steven J Gortler, and Todd Zickler. Low-level vision by consensus in a spatial hierarchy of regions. In *CVPR*, pages 4009–4017, 2015.

[11] Cevahir Cigla and A Aydın Alatan. Information permeability for stereo matching. *Signal Processing: Image Communication*, 28(9):1072–1088, 2013.

[12] Liuyun Duan and Florent Lafarge. Towards large-scale city reconstruction from satellites. In *ECCV*, pages 89–104. Springer, 2016.

[13] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010.

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237, 2013.

[15] Peter Heise, Sebastian Klose, Bjoern Jensen, and Aaron Knoll. PM-Huber: PatchMatch with Huber regularization for stereo matching. In *ICCV*, pages 2360–2367, 2013.

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

[17] Andreas Klaus, Mario Sormann, and Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, pages III:15–18, 2006.

[18] Georg Kuschk and Daniel Cremers. Fast and accurate large-scale stereo reconstruction using variational methods. In *ICCV Workshops*, pages 700–707, 2013.

[19] Ang Li, Dapeng Chen, Yuanliu Liu, and Zejian Yuan. Coordinating multiple disparity proposals for stereo computation. In *CVPR*, pages 4022–4030, 2016.

[20] M.H. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *CVPR*, pages I: 710–717, 2003.

[21] Mikhail G Mozerov and Joost van de Weijer. Accurate stereo matching by two-step energy minimization. *IEEE Trans. on Image Processing*, 24(3):1153–1163, 2015.

[22] Stephen B Pollard, John EW Mayhew, and John P Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 4:449–470, 1985.

[23] Rene Ranftl, Stefan Gehrig, Thomas Pock, and Horst Bischof. Pushing the limits of stereo using variational stereo estimation. In *Intelligent Vehicles Symposium*, pages 401–407, 2012.

[24] Rene Ranftl, Thomas Pock, and Horst Bischof. Minimizing TGV-based variational models with non-convex data terms. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 282–293. Springer, 2013.

[25] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[26] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition (GCPR)*, 2014.

[27] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, pages 519–528, 2006. ISBN 0-7695-2597-0.

[28] M Shahbazi, G Sohn, J Théau, and P Ménard. Revisiting intrinsic curves for efficient dense stereo matching. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 123–130, 2016.

[29] Sudipta N Sinha, Daniel Scharstein, and Richard Szeliski. Efficient high-resolution stereo matching using local plane sweeps. In *CVPR*, pages 1582–1589, 2014.

[30] C. Strecha, W. von Hansen, L.J. Van Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008.

[31] Tatsunori Taniai, Yasuyuki Matsushita, and Takeshi Naemura. Graph cut based continuous stereo matching using locally shared labels. In *CVPR*, pages 1613–1620, 2014.

[32] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a piecewise rigid scene model. *IJCV*, 115(1):1–28, 2015.

[33] Yilin Wang, Enrique Dunn, and Jan-Michael Frahm. Increasing the efficiency of local stereo by leveraging smoothness constraints. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 246–253, 2012.

[34] Yilin Wang, Ke Wang, Enrique Dunn, and Jan-Michael Frahm. Stereo under sequential optimal sampling: A statistical analysis framework for search space reduction. In *CVPR*, pages 485–492, 2014.

[35] Zeng-Fu Wang and Zhi-Gang Zheng. A region based stereo matching algorithm using cooperative optimization. In *CVPR*, 2008.

[36] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous markov random fields for robust stereo estimation. In *ECCV*, pages V: 45–58, 2012.

[37] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, 2014.

[38] Jure Žbontar and Yann LeCun. Computing the stereo matching cost with a convolutional neural network. In *CVPR*, 2015.

[39] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *The Journal of Machine Learning Research*, 17 (65):1–32, 2016.

[40] Chi Zhang, Zhiwei Li, Yanhua Cheng, Rui Cai, Hongyang Chao, and Yong Rui. Meshstereo: A global stereo model with mesh alignment regularization for view interpolation. In *ICCV*, pages 2057–2065, 2015.