Efficient High-Resolution Stereo Matching using Local Plane Sweeps

Sudipta N. Sinha
Microsoft Research
sudipsin@microsoft.com

Daniel Scharstein
Middlebury College
schar@middlebury.edu

Richard Szeliski
Microsoft Research
szeliski@microsoft.com

Abstract

We present a stereo algorithm designed for speed and efficiency that uses local slanted plane sweeps to propose disparity hypotheses for a semi-global matching algorithm. Our local plane hypotheses are derived from initial sparse feature correspondences followed by an iterative clustering step. Local plane sweeps are then performed around each slanted plane to produce out-of-plane parallax and matching-cost estimates. A final global optimization stage, implemented using semi-global matching, assigns each pixel to one of the local plane hypotheses. By only exploring a small fraction of the whole disparity space volume, our technique achieves significant speedups over previous algorithms and achieves state-of-the-art accuracy on high-resolution stereo pairs of up to 19 megapixels.

1. Introduction

As imaging and processor systems continue to increase in resolution and power, the need for more efficient stereo matching algorithms is becoming more acute. Increasing the image resolution not only increases the number of pixels that must be processed, it also increases the number of disparity levels that must be considered. For example, the full-size 2005 Middlebury stereo pairs [15] average 1.4 megapixels (MP) and have a disparity range of 200 pixels; the recent Disney/ETH datasets [12] are as large as 19 MP with disparity ranges up to 1000 pixels.

While great advances have been made in the last decade, most algorithms (with the exception of seed-and-grow and "PatchMatch" approaches, which we discuss in the next section) still evaluate the complete *disparity space image* (DSI), either explicitly, or by doing a local correspondence search over the full range of disparities.

In this paper, we remove this full search using sparse feature correpondences to propose local planes along which we perform small-disparity plane sweeps. This has the advantage of handling highly slanted surfaces without requiring many disparity hypotheses and without any bias toward fronto-parallel orientations. The local plane sweep not only performs subpixel registration, it also deals gracefully with curved surfaces, which a single plane would fail to model.

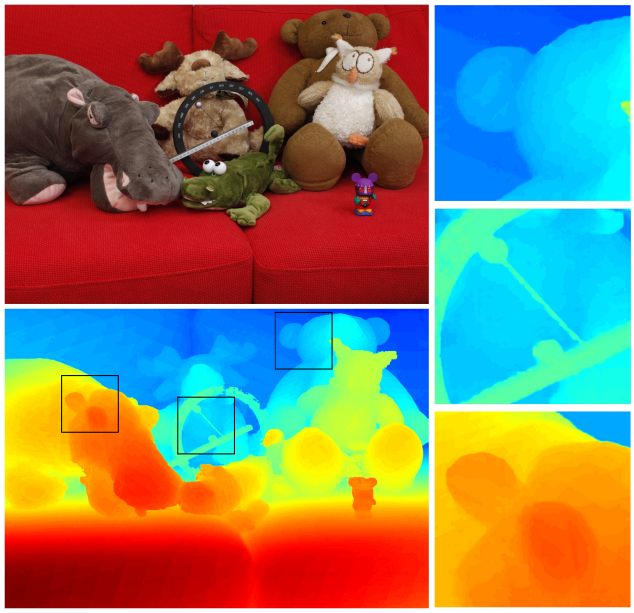


Figure 1. A disparity map computed in 15 seconds by our method for the 11-megapixel *Couch* stereo pair selected from the multi-view datasets released by [12] (best seen in color). More than 98.5% of pixels agree to within 2.0 disparities with their result, which was computed from 100 densely-spaced input images.

The key to our approach is its ability to efficiently propose and evaluate local plane-sweep hypotheses independently. Our algorithm is also able to propagate promising hypotheses into adjoining image regions that have not yet been adequately modeled. We merge the candidate surfaces from the local plane sweeps in a final optimization step using a variant of semi-global matching [10]. The resulting algorithm exhibits high efficiency since it only evaluates a small fraction of potential disparity hypotheses. It also maintains high accuracy due to its subpixel registration and edge-aware global optimization components. We present an experimental comparison with several state-of-the-art techniques that demonstrates the superior accuracy and high efficiency of our approach on 20 high-resolution stereo pairs, including seven new 5-6 MP datasets with ground truth.

2. Previous work

While stereo matching is one of the longest-studied problems in computer vision [9, 16], the last decade has seen an explosion of new algorithms, with dramatic increases in both accuracy and computational efficiency. For example, the Middlebury Stereo Evaluation page [15] lists 140 new algorithms published (and benchmarked) since 2002. The strong interest in this problem has been driven both by the practical utility of such algorithms and the availability of challenging benchmarks such as Middlebury and the KITTI Vision Benchmark Suite [7].

Much of the recent focus in this area has been directed towards achieving pixel-accurate results on the Middlebury benchmarks, often using sophisticated global optimization techniques and Markov Random Field (MRF) formulations. Stereo methods such as ours that employ slanted planes date back to [1]; more recent algorithms include [3, 4, 23]. Most of these techniques, however, take minutes or even hours to run on images of the size we consider in this paper.

Semi-global matching (SGM) [10] is a faster optimization technique that approximates 2D MRF inference by performing cost aggregation along various 1D paths in the image, which allows high-resolution images to be efficiently handled. Recent work on dense scene reconstruction from 3D light fields [12] demonstrates that extremely detailed geometry can be recovered from high resolution images.

Local methods for stereo matching have had a recent renaissance, as adaptive support weights or bilateral filtering can yield results competive with global approaches for scenes with sufficient texture [11]. Unfortunately, both SGM and local methods require evaluating the full disparity space image (DSI), i.e., computing matching costs at each pixel for all disparities under consideration.

Techniques that avoid evaluating the full search space have been proposed both for global and for local methods. Methods for reducing the search space for global optimization include [21, 22, 23]; however, they still require an exhaustive DSI computation, which makes them quite slow.

The seed-and-grow method [20] does in fact only explore part of the search space, but it produces only a partial disparity map, leaving large image regions unlabeled. The more recent highly optimized "Libelas" method [8] builds a triangulation over sparsely matched keypoints and uses this to explore the disparity search space. Finally, PatchMatch stereo [2] also avoids exploring the full disparity space by propagating good disparities from an initial set of guesses to neighbors. We compare our technique with both Libelas and PatchMatch stereo in Section 8.

3. Overview

Unlike most prior work on plane+parallax representations in stereo matching, where a pre-estimated depth map is used to identify which pixels should be associated with dominant planes, in our work, the planes are estimated from sparse feature matches. Each plane π and a narrow range of parallax $\pm T$ defines a local stereo problem in a certain region R of the image. We refer to (R, π, T) as a *local plane sweep* problem and solve it using SGM [10]. This generally only produces a partially valid surface proposal s, as the true disparity of some pixels in R may be outside the range. In the final stage of our algorithm, all image pixels are assigned to individual surface proposals using an efficient discrete optimization framework.

While avoiding the evaluation of the whole DSI is one motivation for our work, our two-staged approach decomposes the original problem into multiple local problems, each of which has a lower range of uncertainty. The restricted range of each local problem further aids regularization and encourages locally smooth surfaces. As we explain in Section 5, the local solutions provide additional discriminative features for the final assignment stage.

Our final stage involves a discrete multi-label pixel labeling problem, where pixels must be assigned to surface proposals. Such problems have traditionally been addressed with global energy minimization methods. In this paper, we show that SGM is also an effective technique for general multi-label assignment problems, which, to our best knowledge, is also a novel contribution. Our proposed approach thus goes significantly beyond simply reducing the label set for a standard SGM stereo method, which operates on ordered label sets and cannot model second-order smoothness (i.e., slanted planes).

4. Hypothesis generation

Feature matching. We establish sparse correspondences by extracting Harris corner keypoints and upright DAISY descriptors [5] in both images. We match these features along epipolar lines and allow small vertical offsets to increase robustness to minor vertical misalignments. After an initial set of matches have been selected using the ratio test heuristic [13], more matches are found in a second round where the horizontal search range is reduced using local estimates of the disparity range at each pixel, which are computed from the initial set of matches.

Vertical alignment. We correct for small vertical misalignments from errors in rectification, which are often more pronounced in high-resolution images, by fitting a global linear model $d_y(x, y) = ay + b$ to the vertical offsets d_y. This involves a few RANSAC iterations of matching features obtained from the previous stage followed by a global least-squares fitting step. The overhead of this correction is negligible since it is applied at the same time as warping the right image during plane-based resampling described in Section 5.

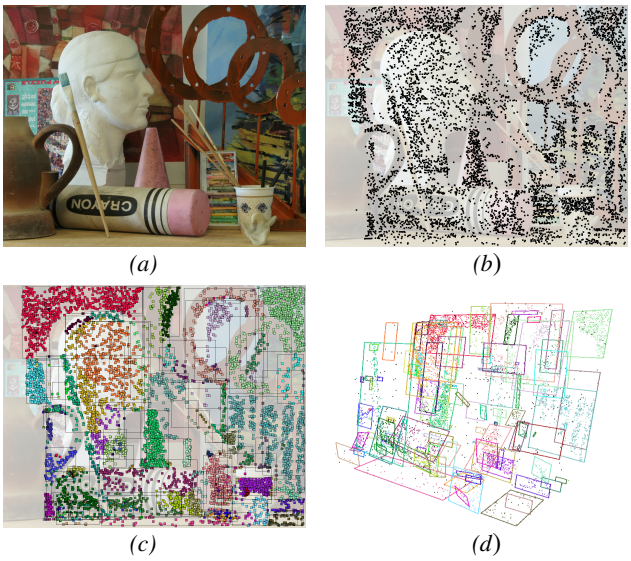


Figure 2. Plane hypothesis generation: (a) input image; (b) extracted and matched keypoints; (c) clustered points (color-coded); (d) initial estimated planes visualized in 3D.

Disparity plane estimation. Although previous stereo matching approaches have exploited the idea of using local planes, they typically require at least an initial disparity map and often also pixel grouping via color segmentation, which tends to make these methods computationally expensive. Instead, in our method, we directly estimate multiple disparity planes from sparse feature matches. We denote the matches $\mathcal{M} = \{x_i, y_i, d_i\}_{i=1}^n$, where (x_i, y_i) and $(x_i - d_i, y_i)$ are matched pixels in the left and right images. A plane $\pi = (\pi_a, \pi_b, \pi_c)$ induces a disparity equal to $\pi(x, y) = \pi_a x + \pi_b y + \pi_c$ at the left pixel (x,y).

To estimate a set of planes and their parameters, we adopt an iterative approach for clustering points in $\mathcal{M}$ such that a single plane provides a good fit to points within each cluster (Figure 2). Our method is inspired by a variational approach used for mesh simplification [6].

We construct a graph $\mathcal{G}$ over $u \in \mathcal{M}$ by connecting each node u in $\mathcal{G}$ to its ten 2D Euclidean nearest neighbors. Our clustering technique then uses this graph to compute k disparity planes by minimizing the overall objective $\sum_i^n e(u_i, \pi_{l_i})$, where the index l_i indicates the plane that u_i is assigned to. The function $e(u_i, \pi) = (d_i - \pi(x_i, y_i))^2$ measures the error between the true disparity at u_i and the disparity induced by the plane.

In the first iteration, random nodes $u \in \mathcal{G}$ are selected as seeds and fronto-parallel planes π_u are computed using the mean disparity at u and its neighbors $\{v_j\}$. Then, elements (v_j, π_u) are inserted into a priority queue Q using the fitting error $e(u, \pi)$ as the priority value. Each plane π_u is propagated via breadth-first traversal as long as the error is within a threshold κ that controls the resolution of the planar approximation (we use $\kappa = 3$ pixels). Nodes are then assigned to planes by popping elements (u, π) from Q and assigning u to π if u is currently unassigned. Once all vertices are assigned, the plane parameters are re-estimated using least squares from the points assigned to it.

In the next iteration, the clustering from the previous step is used to select a point that best fits each plane and these serve as new seed nodes for computing the assignment from scratch. The label propagation from these seeds to all other nodes then proceeds in the same way as described before. This method converges quite quickly so we perform only three iterations in all our experiments.

At this point, we have estimated all plane parameters, but their true 2D spatial extents remain unknown. Extending the planes to the whole image [18] can be slow and is feasible only for piecewise-planar scenes where a small number of planes are needed. In general, obtaining the true extents a-priori is difficult. Therefore, we start by initializing the extents of the planes to the 2D bounding rectangles of their respective clustered points and progressively update them after each round of local plane sweeps as more information becomes available.

5. Local plane sweeps

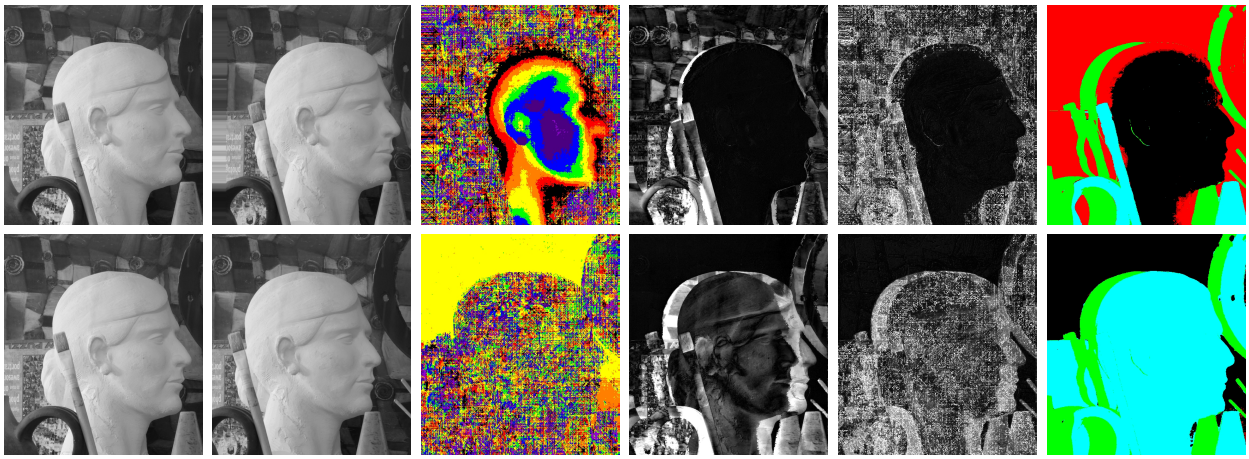
In this section we describe how each local plane sweep problem is solved. We assume that we are given a plane π, a rectangular region R in the left image and a range of $\pm T$ pixels of parallax from π. The desired output is a local surface proposal s together with a *cost map* U indicating how well pixels could be matched within the given range.

First, we use the plane equation (along with the known *corrected* epipolar geometry) to warp the right image into R using bi-cubic interpolation. We then compute normalized cross-correlation over 3×3 patches to obtain a correlation score $NCC(x, y, d)$ in $[-1, 1]$ for each disparity hypothesis, while adding a term ϵ to the denominator in the NCC expression to downweight textureless regions with low variance.[1] We clip and invert the NCC score to turn it into a per-pixel matching cost $C(x, y, d) = 1 - \max(0, NCC(x, y, d))$. Next, we use optimization to refine these raw cost (DSI) estimates in ambiguous (repetitive-texture or low-texture) areas. As with most MRF regularization approaches [19], we minimize the sum of the local unary matching costs and spatially-varying smoothness costs, for which we use a penalty of the *clipped monomial form*,

$$V_{pq}(l_p, l_q) = \begin{cases} 0 & \text{if } l_p = l_q \\ 1 & \text{if } |l_p - l_q| = 1 \\ w_s(I_p - I_q) & \text{otherwise} \end{cases}, \quad (1)$$

where (l_p, l_q) are integer disparities at adjacent pixels, and $w_s(\Delta I) = 1 + \alpha e^{-|\Delta I|/\sigma_I}$ is a gradient-dependent

[1] See the project webpage [17] for full equations and all parameter values used in our implementation.



Left *RightW* D^* R^* U GT

Figure 3. Local plane sweeps for two different scene depths, using disparity planes that align the statue face (top row) and the background (bottom row). *RightW* is the right image warped by the respective plane. D^* is the estimated local disparity map (8 disparity levels color-coded in rainbow colors + black). R^* is the absolute intensity residual corresponding to D^*, which is combined with matching costs and a disparity jump map into a final cost map U (Equation 3). To show that U adequately segments out the "good" pixels within range, we also show the ground-truth segmentation GT (with black = in-range, red = too far, cyan = too close, and green = occluded).

smoothness weight that decays to 1 as the inter-pixel intensity difference gets large. I_p and I_q lie in the range $[0, 255]$; we use $\sigma_I = 8$ and $\alpha = 10$.

Semi-global optimization. While many global methods can compute good approximations of the optimal disparity assignments [19], we use the semi-global matching (SGM) algorithm by Hirschmüller [10] because of its efficiency and good performance. SGM works by computing running sums of local matching costs together with neighbor smoothness costs along several 1D paths in the image. The left-to-right running sum is computed as

$$\begin{aligned} S_0(x, y, d) &= C(x, y, d) + M(x - 1, y, d), \\ M(x, y, d) &= \min_{d'} V_{pq}(d, d') + S_0(x, y, d'), \end{aligned} \quad (2)$$

where $p = (x, y)$ and $q = (x+1, y)$. Because of the special form of the smoothness potential (1), the entries in a new column of summed disparity values can be computed using just 3 add/subtract and 4 minimum operations per disparity level [10]. We compute these running sums in the 8 cardinal directions and sum them to obtain a total aggregated cost volume. The complexity of each mini plane sweep is thus $O(whT)$, where (w, h) is the size of the rectangle R and $2T$ is the parallax range. We experimented with values of T in the range of 2–5 and settled on $T = 3$. The choice is not critical; smaller values speed up the local sweeps but require generating more plane hypotheses.

Once the running sums have been computed, we choose the local disparity d^* at each pixel that minimizes the total aggregated cost and add this value to the plane equation to obtain the base disparity map D^* representing the surface proposal s.

Identifying in-range disparities. In order to identify which pixels could be confidently matched within the given disparity range, we compute three additional quantities from the winning disparities D^*: the intensity residuals $R^*(p)$, gradient-weighted NCC costs $C^*(p) = \|\nabla I(p)\| C(p, d^*)$, and a disparity jump map $J^*(p)$ encoding disparity changes greater than 1 in D^*. We combine these three features into a per-pixel cost map

$$U(p) = \lambda_R R^*(p) + \lambda_C C^*(p) + \lambda_J J^*(p), \quad (3)$$

where $\lambda_R = 0.25$, $\lambda_C = 0.25$, and $\lambda_J = 0.5$ control the relative contribution of each map after normalizing them to the same range. As shown in Figure 3, the cost map $U(p)$ can be used to identify the pixels whose disparities are within range. The intuition is that the three components of U provide complementary evidence for pixels out of range—either via high residuals R^*, high NCC costs at high gradients C^*, or at locations of high (and frequent) disparity jumps J^*.

The cost map U is used in the next two stages, for generating new proposals and for computing the final disparity map using global optimization.

Parameter tuning. We determined default values for all parameters empirically using all available stereo pairs with ground truth. The SGM parameters were tuned in our baseline implementation (see Section 8). We designed the terms in the cost map U (3) and tuned their scalar coefficients using ground-truth solutions to the local plane sweep problems. Ideally these parameters could be learned if a sufficient amount of ground-truth training data was available.

6. Proposal generation

We have developed a greedy approach for efficiently exploring the disparity search space while avoiding unnecessary local plane sweeps as much as possible. Our algorithm divides the left image into a 2D grid of $t \times t$ tiles. We use $t = 256$ pixels and an overlap of 1 pixel vertically and $2T$ pixels horizontally to avoid boundary issues. In our method, local plane sweeps are performed in nR rounds and new proposals are added in batches in all but the last round.

Initially, given k planes and their point clusters in $\mathcal{M}$ (Section 4), we find the planes with associated points within each tile. These form the initial set of proposals $\mathcal{P}_0$ for each tile in the first round of local plane sweeps.

We then iteratively generate additional *online proposals* as follows. After each round r of plane sweeps, we update a winner-take-all (WTA) label map L_r^* at all pixels. Although L_r^* is noisy, it provides an indication of where each plane has support, as frequent labels in a particular tile in L_r^* are preferred candidates for neighboring tiles as well. We then propagate well-supported plane proposals into adjoining tiles. At each tile, we collect the new proposals from the four neighbors, discard planes that were already swept, and add the plane with the most support to the set of planes for the next round. In this manner, promising parts of the disparity space get explored even if they were not among the set of initial proposals based on matched features.

Our complete method is summarized in Algorithm 1. Note that the iterative propagation of proposals can be skipped by setting $nR = 1$, resulting in a simpler and faster method with slightly reduced accuracy (see Section 8).

Compared to previous methods, e.g., [3, 4], our generated proposals are often more targeted. Furthermore, the local plane sweep solutions are efficiently merged using global optimization in a final stage, rather than using expensive MRF fusion moves after every proposal.

7. Global optimization

Given several surface proposals and their associated cost maps, we formulate the final disparity map estimation as a pixel labeling problem that involves assigning each pixel p to one of the candidate surfaces s, or equivalently, to one of the planes π. The optimal assignment L is computed by minimizing an energy function defined on a 4-connected pixel grid,

$$E(L) = \sum_p U_p(l_p) + \sum_{p,q} V_{pq}(l_p, l_q). \qquad (4)$$

For the unary term, we use the truncated costs $U_p = \min(\tau_u, U(p))$, where $U(p)$ is the cost map corresponding to surface l_p (3) and $\tau_u = 40$. The pairwise term V_{pq} is similar to (1) defined in Section 5, but we omit the term for label pairs that differ by one, since our labels no longer form an ordered set. Instead, we now use a pure contrast-sensitive Potts model, and weigh the term for different labels by $w = 25$.

Algorithm 1 *Local-Plane-Sweep-Stereo* (I_l, I_r)

```
M ← match-keypoints (I_l, I_r)
Π ← estimate-disparity-planes (M)      // see Section 4
P_0 ← generate-initial-proposals (Π, t)
S ← ∅      // the set of candidate surfaces
for r = 1 → nR do
  for all proposals p ∈ P_{r−1} do
    s ← local-plane-sweep (p)      // see Section 5
    S ← S ∪ s      // update set of candidate surfaces
  end for
  if r < nR then
    P_r ← generate-online-proposals (Π, S)   // see Section 6
  end if
end for
D ← global optimization (S)      // see Section 7
return D
```

We use semi-global matching [10] to efficiently obtain an approximate solution to this optimization problem. Even though the formulation is conceptually similar to the one described in Section 5, there are some important differences.

First, although cost aggregation is now done on the whole image, pixels within each tile are restricted to a smaller subset of planes. Second, these subsets may vary from tile to tile. We extend the cost aggregation technique used by SGM to exploit the fact that most tiles have a compact subset of labels. The update rules within a tile remain identical to Equation 2 but operate only on the subset of labels pertaining to that tile. However, across tile boundaries, we need to find which local label indices in adjoining tiles correspond to the same plane. By precomputing bidirectional lookup tables for sets of labels in all adjoining tiles, this mapping can be retrieved in constant time during running sum computations.

Final disparity selection. Instead of just selecting the surface with the minimum aggregated cost as in the original method, we compute the final disparity by first selecting multiple (top–m) candidates per pixel. At each pixel, we then compute the median of all multi-valued disparity hypotheses within a $\tau \times \tau$ window centered on that pixel and select this as the final disparity. To select the multiple candidates, we first find the minimum cost c^* at a pixel, and then select the top m planes that have costs less than or equal to λc^* [2]. If a pixel has fewer than m candidates, we replicate its winning candidate to ensure that it has m hypotheses.

[2] We use $m = 2$ and $\lambda = 1.25$. We use $\tau = 5$ for resolutions of 3 MP or less; otherwise we use $\tau = 7$.

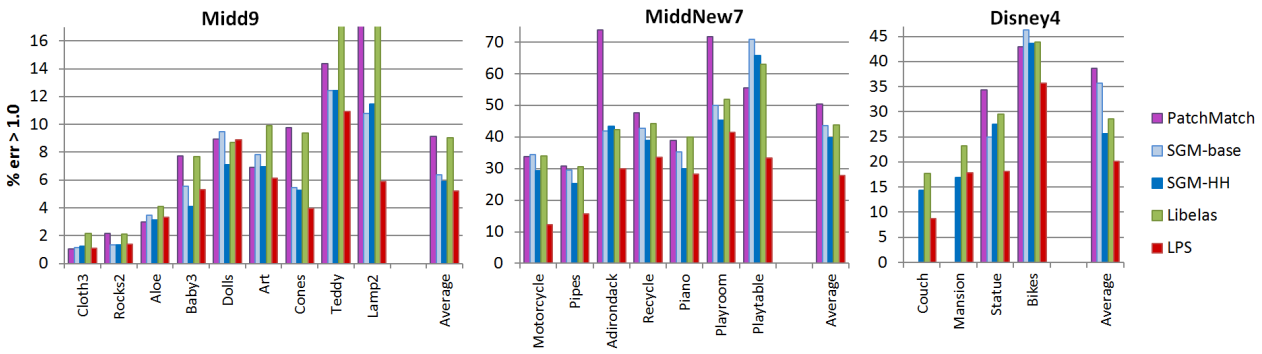


Figure 4. Error rates (% bad pixels with error > 1.0) on the three sets of test images. Our method yields the lowest average errors on all three sets. Note that PatchMatch and SGM-base could not be run on the two largest Disney4 datasets.

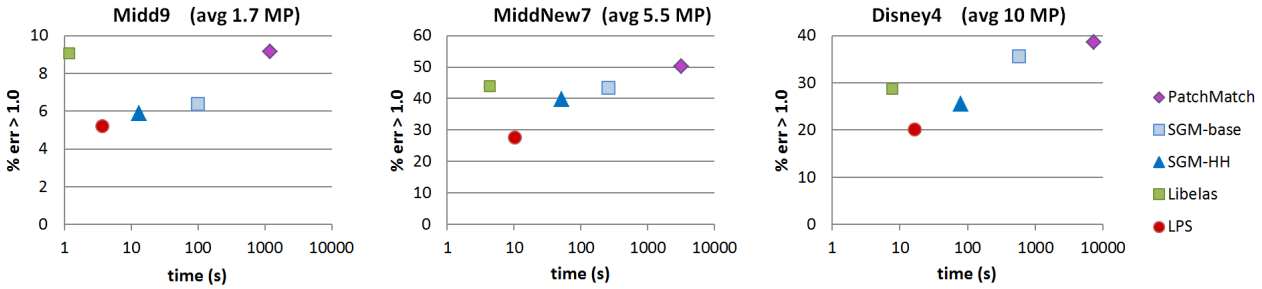


Figure 5. Average error vs. log runtime for the three test sets. Our method yields the lowest errors and the second-lowest runtimes.

8. Experiments

Datasets. We have evaluated our technique on 20 high-resolution images with ground-truth disparities, organized in three groups. The first group, *Midd9*, consists of nine "full-resolution" stereo pairs from the Middlebury benchmark [15] that were used in recent work on high-resolution stereo matching [8, 23]. These images range from 1.4 megapixels (MP) to 2.7 MP in size and include pixel-accurate integer ground-truth disparities. The second group, *MiddNew7*, consists of seven new challenging datasets taken from the 2014 Middlebury public stereo test set [14] (Figure 7 depicts a subset). These images range in size from 5.0 to 5.9 MP and include subpixel-accurate floating-point disparities. The third group, *Disney4*, consists of 4 stereo pairs manually selected from the high-resolution multi-baseline datasets used in [12]. Here, the resolution is between 4.5 MP and 18.9 MP; we treat the disparities computed by [12] from the full sets of 50–100 images each as the ground truth. The disparity range in all 20 pairs varies from 200 to 330 pixels.

Comparison methods and evaluation. We compare our Local Plane Sweep (LPS) method to several state-of-the-art methods able to handle the large image sizes of our test suite: PatchMatch, an implementation of [2] provided by the authors; SGM-HH, an optimized implementation of SGM [10] from the author; SGM-base, our own baseline implementation of SGM; and Libelas, the implementation of [8] publicly shared by the author. All five implementations are in C++ and the reported timings are from runs on a system with a 2.7GHz Quad Core i7 CPU and 32GB RAM.

SGM-base runs a single plane sweep minimizing the same costs as LPS (Section 5), but densely evaluates the full disparity range on the full image. The implementation is unoptimized and single-threaded. In contrast, both SGM-HH and Libelas are heavily optimized using SSE intrinsics and single-threaded, while PatchMatch is multi-threaded. Our LPS implementation is also multi-threaded but could be further optimized; in particular it trivially parallelizes to many-core architectures and GPUs.

Following standard practice, we use the fraction of incorrect disparities at non-occluded pixels for error thresholds $t = 1.0$ and $t = 2.0$ pixels as the accuracy metric. For datasets whose ground truth is only available as integers, we round floating-point disparities to integers in order to allow a fair comparison with integer-based methods. In this paper we present a selection of the important results; the complete results can be found on the project webpage [17].

Results. The accuracy of the five methods on all 20 datasets for $t = 1.0$ is shown in Figure 4. Table 1 lists the average accuracy by test set for both error thresholds as well as runtimes. It can be seen that our LPS method is the most accurate not only in terms of test set averages, but also on all but three of the individual datasets. To examine the relation be-

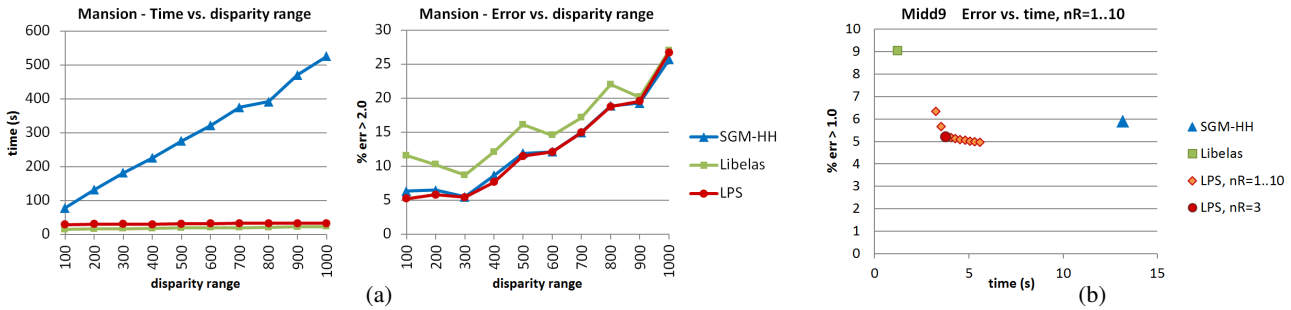


Figure 6. (a) Runtime and accuracy as a function of disparity range on the 19 MP Disney Mansion sequence. (b) Accuracy vs. runtime of our method on the Midd9 images as the number of rounds nR is varied from 1 to 10. We use $nR=3$ for all results reported.

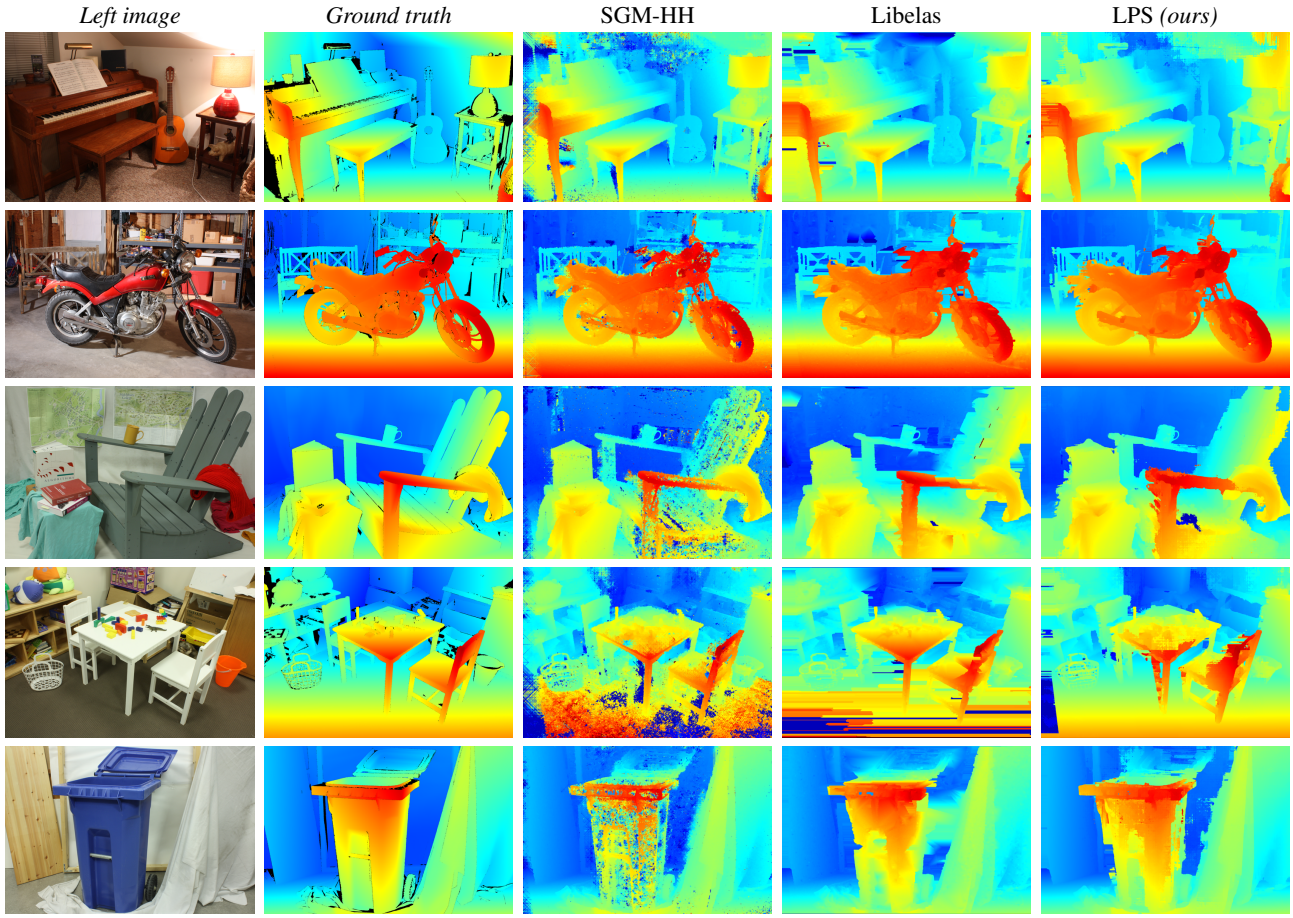


Figure 7. Results on a subset of the MiddNew7 test images with ground truth (*Piano*, *Motorcycle*, *Adirondack*, *Playtable*, and *Recycle*).

tween accuracy and runtime, Figure 5 plots the average errors of all five methods against log runtimes for each of the three test groups. The same trend is observed in all cases: Libelas is the fastest, our method LPS is the most accurate, and SGM-HH is the second most accurate. SGM-base is often comparable to SGM-HH in accuracy, but slower, while PatchMatch is neither competitive in speed nor accuracy.

A qualitative analysis of the disparity maps (Figure 7) reveals that our method excels at recovering slanted surfaces even with weak texture, which are difficult for SGM (for instance the Adirondack chair). Libelas, on the other hand, often makes gross errors if the initial triangulation of feature points fails to include parts of the scene. LPS also can compensate for vertical misalignment which causes problems especially in areas with high-frequency texture (such as the floor in *Playtable*). Completely untextured regions, such as the ceiling in *Piano*, are problematic for all techniques.

To evaluate performance on larger disparity ranges, we selected ten pairs with increasing baseline from the 19 MP *Mansion* dataset [12], yielding disparity ranges from 100 to

		PatchM.	SGM-base	SGM-HH	Libelas	LPS
Time (s)	Midd9	1216	101.1	13.2	**1.2**	3.8
	MiddNew7	3218	257.2	51.2	**4.4**	10.3
	Disney4	7371*	568.3*	78.9	**7.8**	16.6
% Error	Midd9	9.2	6.4	5.9	9.0	**5.2**
$t=1.0$	MiddNew7	50.3	43.5	39.7	43.7	**27.7**
	Disney4	38.6	35.6	25.6	28.6	**20.1**
% Error	Midd9	7.3	3.6	3.7	6.1	**3.2**
$t=2.0$	MiddNew7	44.2	28.8	28.0	29.0	**18.7**
	Disney4	23.0	16.5	13.0	11.9	**7.5**

Table 1. Average runtimes and errors for the each of the five methods. Our LPS method yields the highest accuracy at the second-lowest runtime. *PatchMatch and SGM-base cannot handle the two largest Disney pairs, so the runtimes are extrapolated.

1000 pixels. As shown in Figure 6a, the runtimes of our our LPS method and Libelas remain constant while the runtime of SGM-HH, as expected, is linear in the number of disparities. At the maximum disparity range of 1000 pixels, our method is 16 times faster than SGM-HH. LPS and SGM-HH are comparable in accuracy on all ten pairs and consistently more accurate than Libelas. (The rising errors at larger baselines are partly due to inconsistencies in the disparities computed by [12], which we treat as ground truth.)

Finally, we evaluate the iterative generation of proposals in our method. By default, we use $nR=3$ rounds of proposals. It is possible, however, to trade speed for slightly higher accuracy using additional rounds. Figure 6b shows the results of varying nR from 1 to 10. For $nR=10$, the accuracy of our method improves by 4.9% on Midd9, while the runtime increases by 50%. On the other datasets (not shown here) the accuracy gains are lower: 1.5% for MiddNew7, and 3.7% for Disney4, with similar increases in runtime. In the future, we plan to investigate an automatic criterion for adaptively selecting a smaller number of proposal rounds.

9. Conclusion

We have presented a new highly efficient stereo matching technique that exploits sparse stereo correspondences to perform local plane sweeps without exploring the full disparity search space. We have compared our method with four other state-of-the-art techniques on public benchmark images as well as 7 new challenging high-resolution images with ground truth. The experiments show that our technique is accurate in textured areas, recovers accurate depth discontinuities, and can handle strongly slanted surfaces without bias. Among the efficient methods that are feasible for stereo matching on high resolution images, our technique produces the most accurate results, while being on average only a factor of 2-3 slower than the fastest technique. In the future, we would like to further improve the accuracy of our method, in particular in untextured regions. We also plan to explore a multi-resolution approach with local plane sweeps at multiple resolutions to handle large untextured areas as well as high spatial frequencies within a unified framework.

References

[1] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, 1999.

[2] M. Bleyer, C. Rhemann, and C. Rother. PatchMatch stereo – stereo matching with slanted support windows. In *BMVC*, 2011.

[3] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, 2010.

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

[5] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *TPAMI*, 33(1):43–57, 2011.

[6] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *ACM ToG*, 23(3):905–914, 2004.

[7] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012.

[8] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010.

[9] M. J. Hannah. *Computer Matching of Areas in Stereo Images*. PhD thesis, Stanford University, 1974.

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

[11] A. Hosni, M. Bleyer, and M. Gelautz. Secrets of adaptive support weight techniques for local stereo matching. *CVIU*, 117(6):620–632, 2013.

[12] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM ToG (SIGGRAPH 2013)*, 32(4):73:1–73:12, 2013.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[14] D. Scharstein et al. High-resolution subpixel-accurate stereo datasets. http://vision.middlebury.edu/stereo/data/2014/, Manuscript, 2014.

[15] D. Scharstein and R. Szeliski. Middlebury stereo vision page. http://vision.middlebury.edu/stereo/.

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

[17] S. Sinha et al. LPS stereo project webpage. http://research.microsoft.com/en-us/um/redmond/groups/IVM/LPS/.

[18] S. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, 2009.

[19] R. Szeliski et al. A comparative study of energy minimization methods for Markov random fields with smoothness-based priors. *TPAMI*, 30(6):1068–1080, 2008.

[20] J. Čech and R. Šàra. Efficient sampling of disparity space for fast and accurate matching. In *BenCOS*, 2007.

[21] O. Veksler. Reducing search space for stereo correspondence with graph cuts. In *BMVC*, 2006.

[22] L. Wang, H. Jin, and R. Yang. Search space reduction for MRF stereo. In *ECCV*, 2008.

[23] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous Markov random fields for robust stereo estimation. In *ECCV*, 2012.