Multi-frame stereo matching with edges, planes, and superpixels

Tianfan Xue[a], Andrew Owens[b], Daniel Scharstein[c], Michael Goesele[d], Richard Szeliski[e]

[a]*Massachusetts Institute of Technology, United States of America*
[b]*University of California, Berkeley, United States of America*
[c]*Middlebury College, United States of America*
[d]*Technische Universität Darmstadt, Germany*
[e]*Facebook Research, United States of America*

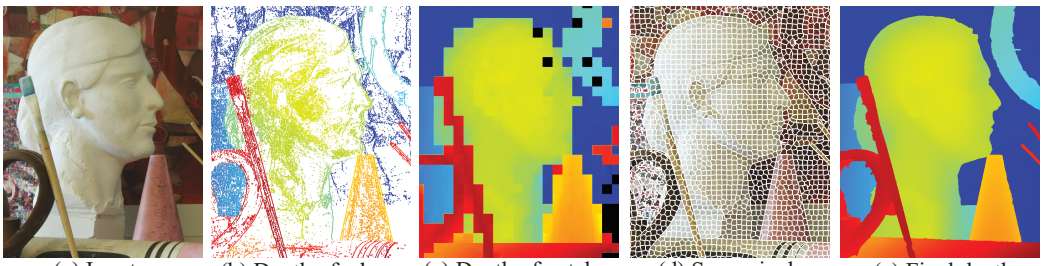
Abstract

We present a multi-frame narrow-baseline stereo matching algorithm based on extracting and matching edges across multiple frames. Edge matching allows us to focus on the important features at the very beginning, and deal with occlusion boundaries as well as untextured regions. Given the initial sparse matches, we fit overlapping local planes to form a coarse, over-complete representation of the scene. After breaking up the reference image in our sequence into superpixels, we perform a Markov random field optimization to assign each superpixel to one of the plane hypotheses. Finally, we refine our continuous depth map estimate using a piecewise-continuous variational optimization. Our approach successfully deals with depth discontinuities, occlusions, and large textureless regions, while also producing detailed and accurate depth maps. We show that our method outperforms competing methods on high-resolution multi-frame stereo benchmarks and is well-suited for view-interpolation applications.

Keywords: stereo, edge matching, superpixels, plane fitting

1. Introduction

While binocular (pairwise) stereo matching remains a challenging open problem, the availability of ubiquitous video recording and computing devices such as cell phones makes it timely to re-examine the benefits of multi-frame stereo matching [1, 2, 3] and edge-based stereo approaches [4, 5]. In this paper, we focus on multi-frame narrow-baseline stereo matching. We show that an edge-based



(a) Input (b) Depth of edges (c) Depth of patches (d) Superpixels (e) Final depth

Figure 1: We match intensity edges in a multi-frame sequence (a) to obtain a sparse depth map (b), from which we derive overlapping slanted plane hypotheses (c). Using an MRF defined over superpixels (d), we obtain a piecewise-planar dense depth estimate, which we refine using continuous optimization (e).

approach is particularly well-suited for multi-frame analysis, since it allows us to focus the computation on the important features, to identify object boundaries, and to defer region-based reasoning as well as handling untextured areas, resulting in dramatically improved performance in regions that are only visible in some frames.

Our approach has four main stages (Fig. 1). The first stage matches edges, which correspond to locations where reliable depth information can be inferred and potential locations of depth discontinuities and occlusion boundaries. The second stage estimates a coarse description of the scene by fitting overlapping slanted planes to the matched edges. In the third stage, we use a Markov random field to infer a dense piecewise-planar depth estimate of the scene over a set of superpixels. Finally, we refine the depth map using piecewise-continuous variational optimization. Decomposing the model into multiple stages makes it easier to test each component individually and to perform experiments on different variants. Each of the proposed stages is optimized to infer the most reliable information available at that point, building up a hierarchy of successively more detailed, dense, and accurate shape estimates.

This approach mainly solves two issues of previous stereo matching algorithms: 1) the foreground objects are sometimes enlarged in estimated depth maps due to occlusion, which is also known as foreground fattening, and 2) depth in flat regions are sometimes inaccurate due to insufficiently matched features. In this work, we solve these two problems by jointly using edge-based matching [4, 5, 6, 7, 8, 9, 10] and plane representation [2, 11, 12, 13]. Both these two ideas have been used separately in previous works, but we have demonstrated in this work that by combining them in a single optimization framework, we can

reduce foreground fattening and improve the accuracy of stereo matching in flat regions and occlusion boundaries.

More specifically, multi-frame *edge matching* produces a semi-dense, reliable, and rich representation of 3D shape. Since edges can be extracted to sub-pixel precision, the resulting disparity estimates can attain very high precision. Furthermore, the features of edges usually changes more slowly than the photometric (texture) appearance. In particular, at depth discontinuities where the background colors can change, the edge orientation and foreground colors usually remain unchanged. Overlapping *slanted planes* (layered depth models) provide a compact set of structure hypotheses that aggregate local low-level evidence in a way that naturally supports discontinuities [14]. Layered models also support efficient per-pixel depth inference, due to the small number of hypotheses being considered at each pixel [15]. Finally, *superpixels* [16] are an efficient way to obtain dense reconstructions that naturally align with discontinuities and textureless regions, removing the need for spatially shiftable matching windows. They also reduce the number of variables over which global inference is performed, leading to more efficient and less error-prone algorithms.

2. Related Work

Our approach revisits some of the earliest approaches to stereo, including edge-based and line-based matching [4, 5, 6, 7, 8, 9, 10], multi-baseline approaches [2, 11, 12, 13], and epipolar-plane analysis [1, 3]. Many of these early ideas have gained new popularity over the years; e.g., gradients are used as a robust matching primitive for stereo [17, 18, 19], and optical flow [20]. Multi-frame analysis has also been successful for occlusion reasoning [11, 12, 21, 22]. Most work in stereo, however, has focused on binocular (two-frame) methods for producing dense disparity maps [23].

Depth discontinuities pose a challenge for dense per-pixel stereo methods that utilize window-based matching. A number of solutions have been proposed to reduce the resulting foreground-fattening effect, such as shiftable windows and temporal selection [12, 23, 24], but these methods still suffer at depth boundaries, especially when the background region lacks texture. We address this problem by tracking and robustly combining intensity edges over multiple frames. This approach is similar to methods that track keypoints [25] and also fine-to-coarse models that detect lines in epipolar planes [3]. However, since we match edges (rather than keypoints), our initial reconstruction is relatively dense, and it contains many complete object boundaries (Fig. 1b).

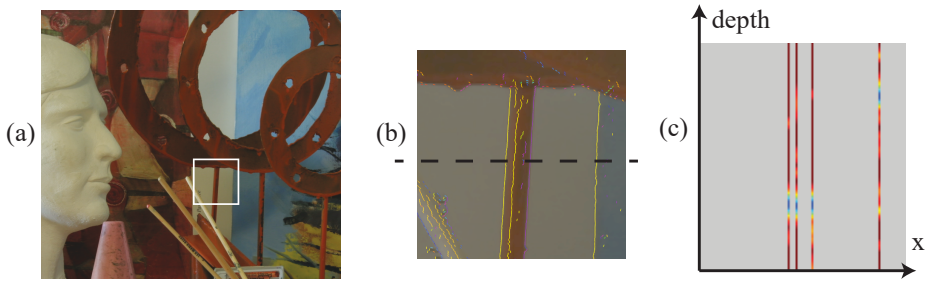


Figure 2: Edge extraction and matching: (a) input frame; (b) detail of extracted edges for the marked region; (c) summed edge matching cost function for the scanline in (b), where blue indicates lower costs, red indicates higher costs, and gray indicates no edges.

Our method for producing a dense depth map from edge reconstructions is based on the assumption that a complicated scene can be approximated by a set of planar surfaces. Similar assumptions are also used in previous work, in which surfaces are detected either via matched keypoints [15], or based on the photometric consistency of matched regions [26, 27, 28, 29]. In contrast, we detect the planes in the scene by fitting planes to matched edges, which are denser than matched keypoints, and more robust than photometric appearance on occlusion boundaries.

To obtain final depth maps, we follow several recent methods and estimate a piecewise planar reconstruction with an MRF defined over superpixels. This is closely related to the work of Yamaguchi et al. [30, 31], since our MRF uses pairwise terms between superpixels to model occlusion relationships. In that work, the piecewise-planar model is used to refine depth estimates originating from semi-global matching. In contrast, our MRF takes edge matches directly as input, and creates depth maps with crisp edges, using a novel edge-superpixel compatibility function to model the relationship between edge depths and superpixel depths.

Besides, in this work, we focus on narrow baseline stereo. Wide baseline multiview stereo algorithm used in the literature of image-based 3D reconstruction and structure-from-motion [32, 33, 34, 35] is beyond the scope of this work.

3. Edge Matching

We first extract and match edges in all images individually. The goal of this stage is to provide sparse but reliable disparity estimates for later stages, e.g., the creation of local plane hypotheses, and to signal potential locations for depth discontinuities.

While our approach could be extended to arbitrary motion, in this paper we focus on equally-spaced multi-baseline sequences with purely horizontal camera motion [1, 2, 3]. We typically use sequences with 7–9 equally-spaced frames and choose the central frame as the reference frame. Given this setup, we restrict edge detection to the horizontal direction. Instead of using regular rotationally-invariant edge detectors [36, 37, 38], we identify local maxima of the squared horizontal gradients and refine their location using a parabola fit. We control edge density using a detection threshold (we use 0.25 pixel in the experiment) in the reference frame. To ensure the detected edges are distributed more evenly, we only keep all the edges whose magnitude is maximum in the 5×5 local patch around that point. We also keep strong edges whose gradient is larger than a higher threshold (set to 2.64 intensity levels) even though they are not local maximums. Besides, we use a smaller threshold (half of that in the reference frame) in the matching frames to increase the chance of matching. We also estimate the edge orientation from the local 2D gradient (Fig. 2b), which we use for edge matching.

After detecting all the edges in our source images, we match them to obtain a sparse estimate of 3D scene geometry for each edge in the reference image. We represent this geometry (and the final depth map) in terms of *disparity*, i.e., the horizontal displacement (in pixels) between neighboring frames.

To find corresponding features, we define a cost function to score the fitness of a particular match using a weighted sum of three distance terms: location distance, orientation difference, and color difference.

The *location distance* D_l between the predicted location $x_p = x_r - bd$ and its actual location x_e is $D_l(d) = |x_p - x_e|$, where x_r is the edge location in the reference frame, $b \in [-4, 4]$ is the integer baseline between the reference and matched frame, and d is the disparity.

We define the *orientation difference* D_o as the absolute value of the cross product between the normal vectors of the reference edge and the putative matching edge,

$$D_o = |\mathbf{n}_r \times \mathbf{n}_e| / |b|, \tag{1}$$

where $\mathbf{n}_r$ and $\mathbf{n}_e$ are the edge normals in the reference and matching frames. Since edges on slanted surfaces change their orientation as both a function of slant and distance from the reference image, we divide the cross product by the frame difference from the reference frame.

The *color difference* D_c is measured as the smaller of the two differences (in

RGB space) between corresponding pixels adjacent to the edge location,

$$D_c = \min(\|\mathbf{c}_{r-} - \mathbf{c}_{e-}\|, \|\mathbf{c}_{r+} - \mathbf{c}_{e+}\|), \tag{2}$$

where $\mathbf{c}_{*\pm}$ are the colors of the pixel just to the right and left of a detected edge location. Because the background color may change at depth discontinuities, we take the smaller of the two color differences.

The three squared differences are weighted by their inverse expected variances and summed to produce a composite weighted squared distance (variances are chosen empirically: $\sigma_l = 0.4$, $\sigma_o = 0.25$, and $\sigma_c = 10$),

$$D_m^2 = \sigma_l^{-2} D_l^2 + \sigma_o^{-2} D_o^2 + \sigma_c^{-2} D_c^2, \tag{3}$$

which is then passed through a robust penalty function to obtain the cost of matching a given reference edge at a disparity d to its nearest edge in a different image,

$$E_m(d) = \phi(D_m^2). \tag{4}$$

We use the robust penalty function suggested by Zach [39],

$$\phi(d^2) = w^2 d^2 + \tau^2 (1 - w^2)^2 / 2, \tag{5}$$

where $w^2 = \max(0, 1 - d^2/\tau^2) \in [0, 1]$ is an *inlier weight* and τ is an outlier threshold, which we set to 3.

Since we operate in a multi-frame setting, we need to consolidate all pairwise matches between a reference frame and all other frames. One approach is to simply sum the cost functions E_m across all frames evenly and hope that the robust outlier function takes care of missing edges (e.g., due to occlusions). Another alternative is to just take the best fraction (e.g., the top half) of matches, i.e., sum up the $N/2$ lowest scores [13],

$$E_m(e, d) = \sum_{i=1 \ldots N/2} E_m^{(i)}(d), \tag{6}$$

where the $E_m^{(i)}$ are sorted in increasing value and e denotes the edge under consideration. Experimentally, we have found that the second approach produces more reliable estimates.

Fig. 2c shows the cost function $E_m(e, d)$ for a set of edges e on a given scanline as a function of their disparities d. This is a sparse version of the *disparity space image* (DSI) [24, 23] that is used by many stereo algorithms to form a cost volume. One can see a clear trend where the dominant surfaces appear as continuous slowly-varying minima (blue regions) in this cost function.

4. Plane hypothesis generation

While multi-frame matching often produces correct matches by independently matching each edge, i.e., selecting the value of d for each edge e that minimizes $E_m(e, d)$, the reliability can be greatly improved by aggregating the matching information spatially. One can imagine several ways to do this, including approaches that use the cost volume to directly integrate local support [23, 40]. In our work, we use fixed-size overlapping square image patches as the main mechanism to simultaneously perform spatial aggregation and to estimate local plane hypotheses. We then extend the local hypotheses to cover larger regions.

Patch-based aggregation and plane fitting. To obtain an initial set of plane hypotheses, we divide the image into 32×32 patches with 16 pixels overlap and perform an independent *plane sweep* [8, 23]. We then refine each patch to obtain a slanted plane hypothesis. Once all of the hypotheses have been generated, we assign each edge to the overlapping plane whose fit has the highest confidence.

We start by first matching each reference edge to all the other edges (within the disparity range) and forming a single sampled cost function per edge, as shown in Fig. 2c. Once we have computed $E_m(e, d)$ for the permissible range of disparity values d, we find all local minima with inlier weights w^2 greater than 0.5 and keep their sub-pixel locations (obtained using a parabolic fit) and cost value. In practice, we only keep the top 5 (or fewer) minima and throw away the rest.

Next, we aggregate matches over the whole patch to compute the best-fitting plane. We do this in two steps. First, we compute a weighted histogram of the top three disparities for all edges, where the weight is a bilinear tent function centered over the patch. We then find the maximum histogram value and use a three-point parabolic fit to find a refined estimate for the best (fronto-parallel) patch disparity value d_{fp}. Next, we fit a robust weighted 3-parameter plane model to the inlier matches, i.e., those matches that are either within 0.5 disparities of the winning disparity d_{fp} or whose disparity gradient is below 0.5, again using a bilinear spatial weighting funtion.

The last stage of our edge-matching algorithm assigns each edge to its best overlapping plane hypothesis by choosing the plane (within ± 1 disparities) with the lowest matching cost. This filters out spurious bad matches that are not associated with any nearby planes.

Plane merging and extension. In the previous stage, small planar patches are locally fit to matched edges. We now merge patches that are roughly coplanar to reduce the number of plane hypotheses and improve the accuracy of estimated plane equation using an efficient greedy algorithm.

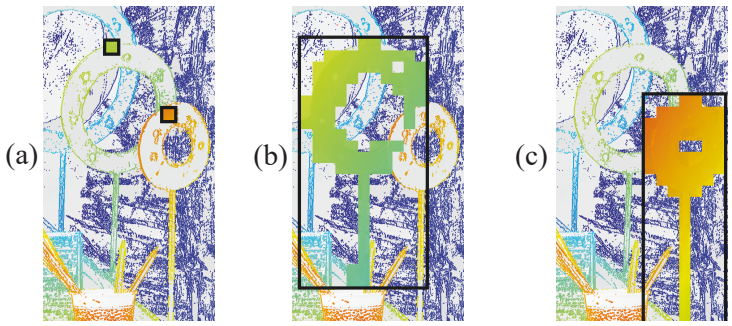


Figure 3: Illustration of plane merging. (a) Two plane patches fitted to depth of edges. The sparse points show the depth of matched edges, and the colors inside the two black boxes encode the depth of the plane patches. (b,c) The results of plane merging, using the two patches in (a) as seeds.

We first select the patch with the most edges as the initial seed plane and grow it by merging other roughly coplanar patches until there are no patches left that can be merged. We remove all inlier edges that belong to this plane (within 0.2 disparities of the plane equation). We then pick the next patch that contains the most edges as the initial seed plane. We repeat this process until we cannot pick a patch that contains more than 3 edges to be a seed plane.

To grow a seed plane, we repeatedly select among the neighboring patches the one that best fits the plane. We evaluate how good the patch fits to the plane by counting the number of edges in the patch that lie on the extension of the plane (also within 0.2 disparity of the plane). We refit the plane equation to all inlier edges after merging, and repeat this process until none of neighboring patches contain more than 2 inlier edges. We set the extent of the final merged plane as the bounding box of all constituent patches, as shown in Fig. 3. For robustness, we also extend the plane extent by $\lfloor \frac{W}{20} \rfloor$ pixels, where W is the width of the input image.

5. Superpixel MRF

After the previous stage, each pixel is covered by one or more planes. In the next step, we decide which plane a pixel actually belongs to. To reduce the search space and accelerate the algorithm, we segment the reference frame into superpixels using SLIC [16] and assign all pixels within a superpixel to the same plane.

Our superpixel assignment is based on two sources of information, namely *photo consistency*, as used in most stereo matching algorithms [23, 15], and *edge*

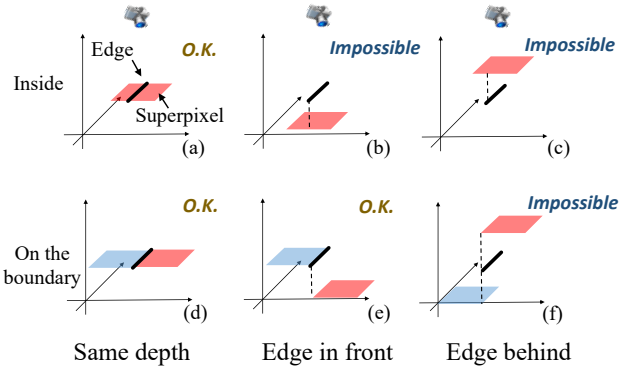


Figure 4: Edge consistency. (a–c) When an edge (black line) is inside a superpixel (red square), their depths must agree. (d–f) An edge on the boundary between superpixels cannot be behind either of them.

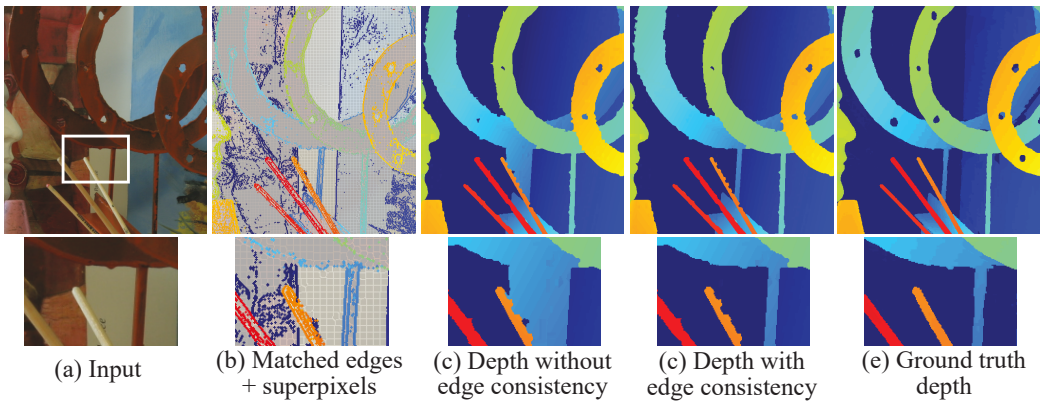


Figure 5: Edge consistency reduces foreground fattening. See text below for more details.

consistency, which requires that the depth of edges should be consistent with the depth of the corresponding plane.

The assignment problem is modeled as an MRF with a data term consisting of the photo-consistency term E_{photo} and an edge-consistency term E_{edge} as described below.

Photo consistency. Let P_j be the j-th plane after plane merging and L_i the index of the plane that superpixel S_i is assigned to. Let t^* be the index of the reference frame and t be another frame in the sequence, and let I^t and ${I_x}^t$ be the color image and its x-gradient of the t-th frame respectively. Also, let $\mathbf{x}$ be the location of a pixel in the i-th superpixel S_i and $W^t(\mathbf{x}, P_{L_i})$ be the corresponding pixel in frame t assuming $\mathbf{x}$ is on plane P_{L_i}. The photo-consistency term penalizes both color

and gradient difference [18] between the reference frame and frame t:

$$E^t_{photo}(L_i) = \lambda_c \sum_{\mathbf{x}\in S_i} \phi_{T_c}\Big(I^{t^*}(\mathbf{x}) - I^t(W^t(\mathbf{x}, P_{L_i}))\Big) + \lambda_g \sum_{\mathbf{x}\in S_i} \phi_{T_g}\Big({I_x}^{t^*}(\mathbf{x}) - {I_x}^t(W^t(\mathbf{x}, P_{L_i}))\Big), \quad (7)$$

where $\phi_T(\mathbf{v}) = \sum_j \min(|\mathbf{v}_j|, T)$ is the truncated l_1-norm. In all experiments we use the following weights and thresholds for color and gradient terms: $\lambda_c = 0.1, \lambda_g = 0.9, T_c = 0.03, T_g = 0.004$.

Edge consistency. The second data term E_{edge} measures the consistency between the depth of superpixels and the depth of edges. If an edge is inside a superpixel or on its boundary, the depth of that superpixel should agree with the depth of the edge. There are two cases to be considered. First, when an edge is inside a superpixel (the top row of Figure 4), their depths should be the same. Second, when an edge is on the boundary of a superpixel (the bottom row of Figure 4), the edge can either lie at the depth of the superpixel or in front of it, since it might belong to another, closer superpixel.

For the first case (an edge inside a superpixel), we penalize the difference in depth between edges and superpixels:

$$E_{in}(L_i) = \sum_{e\in\mathcal{I}_{S_i}} \psi\left(d_e - D(\mathbf{x}_e, P_{L_i})\right), \quad (8)$$

where d_e is the depth of edge e predicted in edge matching, $\mathbf{x}_e$ is the 2D spatial coordinate of that edge, $\mathcal{I}_{S_i}$ is the set of edges inside the superpixel S_i, $\psi(x) = \min(x^2, T_d)$ is the squared truncation function, and function $D(\mathbf{x}_e, P_{L_i})$ returns the predicted depth by the plane equation P_{L_i} at the location $\mathbf{x}_e$. For the second case (an edge on the boundary of a superpixel), we penalize the difference in depth only if an edge is behind a superpixel:

$$E_{bound}(L_i) = \sum_{e\in\mathcal{B}_{S_i}} \psi\Big(\max(D(\mathbf{x}_e, P_{L_i}) - d_e), 0)\Big). \quad (9)$$

The edge-consistency term is then the sum of these two terms

$$E_{edge}(L_i) = E_{in}(L_i) + E_{bound}(L_i). \quad (10)$$

Figure 5 demonstrates how the edge-consistency term reduces foreground fattening. Without edge consistency, the light region behind the dark vertical bar (Figure 5a) is grouped with the foreground in the recovered depth map (Figure 5c).

The depth of the background region should be dark blue, but it is light blue (indicating the foreground depth) in Figure 5c. If the edge-consistency term is included, the light region is correctly grouped with the background (Figure 5d). This is because the matched edges on the left (dark blue dots in Figure 5b) push their neighboring superpixels to the background, as the edge consistency term does not allow a superpixel in front of its neighboring edges. Some superpixels in that light region are not directly neighbors of matched edges on the right, but they are still assigned to the background because of the smoothness term discussed below.

Smoothness term. The smoothness term $E_b(L_i, L_j)$ checks whether the labels of two neighboring superpixels are consistent. It is a product of two terms:

$$E_b(L_i, L_j) = \lambda_{smooth} \cdot w(S_i, S_j)\psi(P_{L_i}, P_{L_j}, B_{i,j}). \tag{11}$$

The first term $w(S_i, S_j)$ is an affinity term that measures the similarity of the appearance (mean color) of superpixels:

$$w(S_i, S_j) = \exp(-(C_i - C_j)^2/\sigma_c^2). \tag{12}$$

The second term $\psi(P_{L_i}, P_{L_j}, B_{i,j})$ measures the average disparity difference of two superpixels along their shared boundary, and it is truncated in the similar way as in Eqs. 8 and 9.

At last, we solve the MRF using alpha expansion [41].

Refinement. At this point, our pipeline yields a piecewise planar reconstruction. In order to faithfully reconstruct smoothly varying surfaces, we add a refinement step that uses the depth information at edges located close to the piecewise planar reconstruction and performs a variational piecewise-smooth 2D interpolation [42] (Fig. 6).

The piecewise planar reconstruction provides the location of discontinuities and serves as a weak data term. We combine this with a second, stronger data term consisting of robust weighted edges (using again the smoothly truncated quadratic (5) by Zach [39]), where such edges are available close to the surface. In practice, we use a cut-off of 1.0 disparity levels and perform only a single iteration, since a reweighted least-squares approach yielded worse results. This yields a smoothly-interpolated discontinuity-preserving reconstruction that is closer to the (locally more reliable) original edges than the planes.

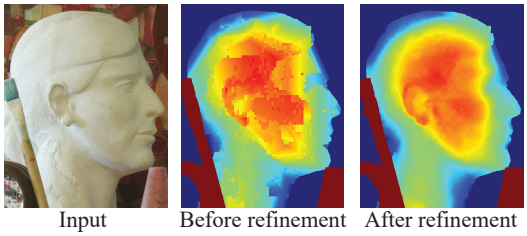


Figure 6: Effect of depth refinement. The model introduces in Section 4 can only model piece-wise planar surfaces. Through the depth refinement, our model can also model the curved surfaces, like human faces, and most depth "staircasing" is removed after the refinement.

6. Experimental Results

The focus of our work is high-resolution multi-baseline matching. Our method utilizes multiple frames taken from a narrow baseline and produces a depth map for the reference (center) frame. Unfortunately, there are no established benchmarks for this scenario. The Middlebury and KITTI stereo benchmarks [43, 44] are restricted to two input views, while existing multiview benchmarks [45, 46] use a wide-baseline scenario and evaluate 3D surface meshes. Thus, providing a fair comparison with published results is difficult. Here, we compare with existing two-view results. In addition, we implemented a multi-frame extension of SGM [47] to provide a baseline comparison for our method when utilizing all frames. Finally, we show a qualitative comparison with Kim et al.'s approach [3], which uses a multi-baseline setup with a large number of views. All results are also available at `http://people.csail.mit.edu/tfxue/edgestereo/`.

6.1. Datasets and Baselines

We evaluate our algorithm on two sets of high-resolution multi-baseline sequences. The first one, *Midd-F*, consists of 7 full-resolution sequences from the Middlebury 2003–2006 datasets [43], with 7–9 frames each at a resolution of 1.4–2.7 megapixels (MP). While our main focus is high-resolution matching, we also report results on quarter-resolution sequences *Midd-Q*, which include the *Teddy* and *Cones* images used in the Middlebury stereo benchmark version 2. The second group, *Disney*, is from the high-resolution multi-baseline dataset used in Kim et al. [3]. Each sequence contains 101 frames, from which we pick 9 frames in 5-frame increments.

The disparity range in both datasets varies from 200 to 330 pixels. All sequences contain either 7 or 9 frames, and we always select the middle frame as the reference frame. For the Middlebury datasets, we create ground-truth disparities for the center frame by warping the the provided left and right disparity

Sequence	Libelas (2 frames)		LPS (2 frames)		SGM (3 frames)		SGM (all frames)		MCCNN (all frames)		Ours (3 frames)		Ours (all frames)	
	nonocc	all	nonocc	all	nonocc	all	nonocc	all	nonocc	all	nonocc	all	nonocc	all
Aloe	2.83	5.47	1.73	7.81	1.87	3.70	2.27	3.85	2.12	4.21	2.30	3.50	**1.06**	**2.38**
Art	7.28	17.61	4.43	20.02	5.57	9.64	4.77	8.41	8.56	11.79	2.06	3.54	**1.93**	**3.32**
Cloth3	1.67	7.24	0.56	6.39	0.45	1.38	0.44	1.26	1.07	1.34	0.71	1.45	**0.42**	**1.01**
Cones	4.13	11.04	1.59	9.21	2.25	5.11	1.73	4.25	2.69	4.15	3.40	5.82	**1.06**	**2.59**
Dolls	4.81	12.52	4.02	13.10	4.46	7.76	6.25	9.21	3.42	**4.71**	2.68	6.15	**2.25**	4.84
Rocks2	1.51	5.47	0.82	6.55	0.85	1.70	0.82	1.63	1.07	1.26	0.71	1.34	**0.59**	**1.06**
Teddy	12.00	18.46	7.55	16.40	10.30	13.31	7.47	9.68	7.27	7.84	6.61	8.69	**3.84**	**5.46**
Avg.	4.89	11.12	2.96	11.35	3.68	6.09	3.39	5.47	3.74	5.04	2.64	4.36	**1.59**	**2.95**

Table 1: Errors (%) for three methods on *Midd-F* (we use threshold t=2.0 for all full-resolutions experiments).

	Aloe	Art	Cloth3	Cones	Dolls	Rocks2	Teddy
Nonocc	1.57	2.56	0.75	1.02	2.69	1.08	2.16
All	3.51	4.45	1.48	2.56	4.41	1.78	3.43

Table 2: Errors (%) for our method on *Midd-Q* (t=1.0). Errors of the top-performing two-view algorithms on *Teddy* and *Cones* are around 2–3% (in *all* regions).

maps. For *Disney*, ground-truth disparities are not available, so we only provide a qualitative comparison.

6.2. Quantitative Results

We compare our method against four stereo algorithms. Libelas [25] and Local Plane Sweeps (LPS) [15] are two competitive high-resolution 2-view algorithms. As mentioned, these methods do not utilize the additional input frames, so they face a greater challenge especially in occluded regions. For a multi-frame baseline comparison, we extend the OpenCV implementation of the semi-global matching (SGM) algorithm [47] from two frames to multiple frames by replacing the original matching cost with a summation of k lowest matching costs between the reference frame and other frames. We also extend one of the pioneer neural-network-based stereo algorithm, MCCNN [48], to multiple frames using the similar strategy.

Following established practice, we report the error rate for *non-occluded* pixels visible in both input frames as well as for *all* pixels. To obtain the occlusion mask for our method, we warp the left and right occlusion masks to the center frame.

Table 1 compares numerical results for the four methods on the full-resolution dataset *Midd-F*. The table shows that in non-occluded regions, our method beats Libelas and LPS both in non-occluded and all regions. It also beats the baseline SGM method by a comfortable margin in terms of average errors, as well

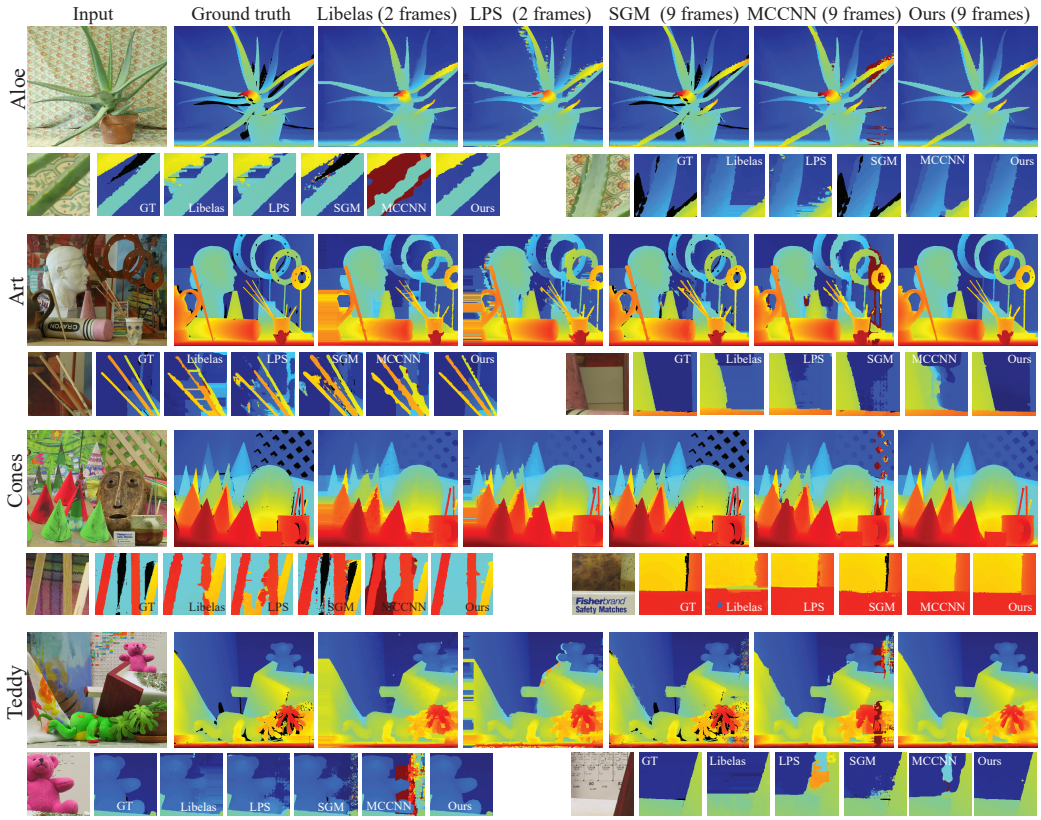


Figure 7: Reconstruction results on *Midd-F*. Corresponding close-up views are shown below each sequence (please see supplementary material for the full results).

as on most individual sequences, and our algorithm with 3 frames as input even beats SGM and MCCNN with 9 frames as input. The largest improvements of our method over the baseline SGM method and MCCNN are on sequences such as *Teddy* and *Art* containing textureless regions, where our edge-based approach successfully prevents foreground fattening. This demonstrates the power of our edge-based multi-frame approach in the presence of occlusions.

Table 2 shows our numerical results on the quarter-resolution sequences *Midd-Q*, which are comparable to the best two-view results listed in the Middlebury stereo benchmark version 2, for both *nonocc* and *all* regions.

6.3. Qualitative Results

A side-by-side comparison on a subset of the *Midd-F* and *disney* is shown in Figures 7 and 8. (See the supplementary materials for the full results and error maps.) Note that LPS and Libelas use the left frame (the third frame in 9-frame

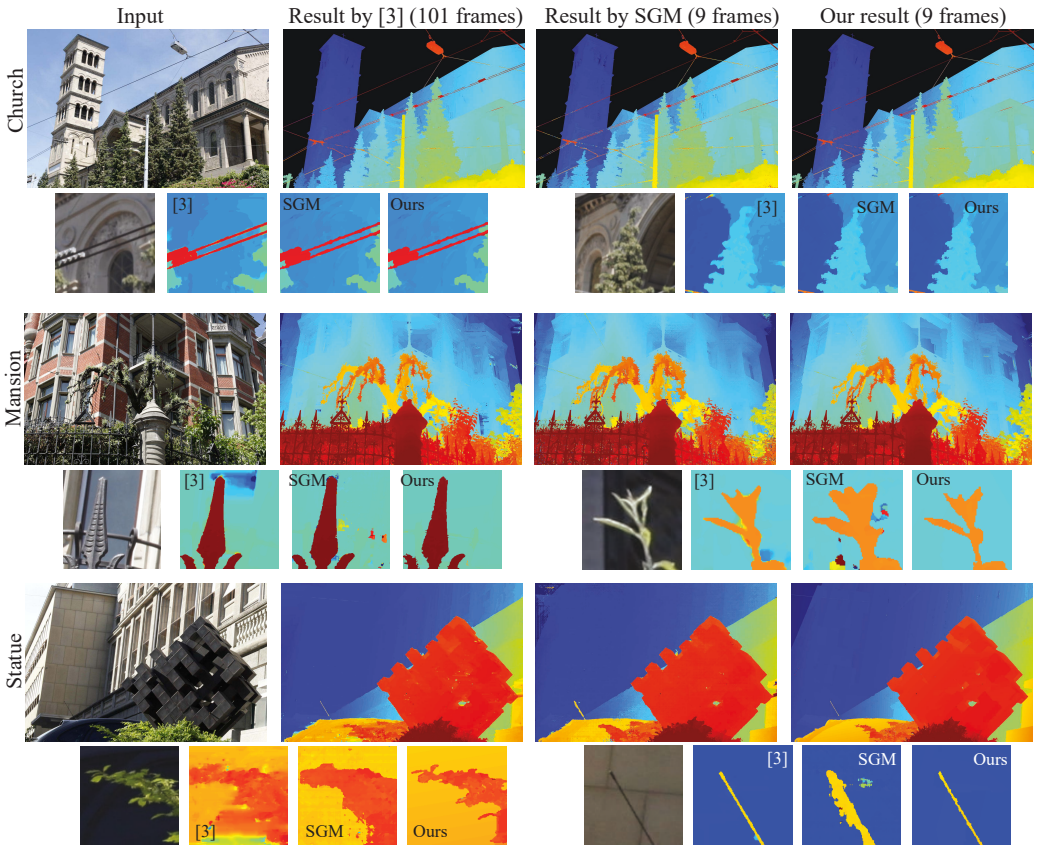
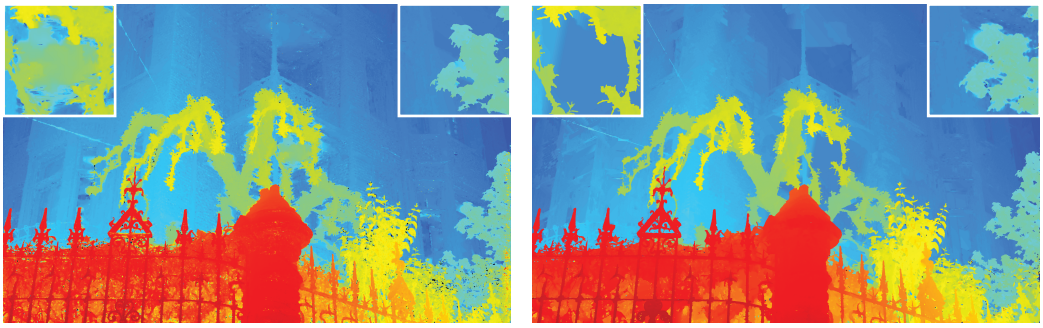


Figure 8: Reconstruction results on *Disney* for Kim at al. [3], SGM, and our algorithm. Kim et al. [3] use 101 frames as input, while SGM and our method use only 9 frames. Corresponding close-up views are shown below each sequence.

sequences) as the reference frame, while our algorithm and SGM use the middle frame.

On *Midd-F*, our algorithm produces much cleaner depth maps compared to the other methods. The boundaries of objects are cleaner with fewer fuzzy edges (e.g., the leaves in *Aloe*). In the depth maps created by either SGM or MCCNN, foreground objects with untextured background are often enlarged (e.g., *Cones* and the right edge of the *Teddy*). These are typical examples of foreground fattening that our algorithm can avoid using edge reasoning. Moreover, in the flat regions, like the white background besides the teddy bear, SGM or MCCNN generate incorrect depth, while our approach correctly predict depth in those regions using our planar representation.



Result by [3] (11 frames) Our result (11 frames)

Figure 9: Comparison to Kim et al. [3] on *Disney*'s *Mansion* sequence. Both algorithms use 11 frames as input. Corresponding close-up views are shown on the right.

On *Disney*, both our algorithm and SGM use only 9 frames as input and achieve a similar performance as the method by Kim et al. [3], which uses 101 frames. Many thin structures are recovered in the depth maps, like the power lines in *Church* and the leaves in *Mansion*. Compared with SGM, our algorithm again reduces the foreground fattening for objects with untextured background, e.g., the leaves in *Mansion* and *Statue*, and the car antenna in *Statue*.

Fig. 9 compares our method with the method by Kim et al. [3] when both methods are run on only 11 frames (thus handicapping the latter), and our depth map has much cleaner boundaries and again avoids foreground fattening.

6.4. Analysis

To understand how each step in our algorithm contributes to its performance, we conduct several evaluations on the *middF* dataset. All the error rate, if not specified, are calculated using $T = 2.0$.

Edge detection. First, to evaluate the edge matching, we compare our results with an oracle that predicts the actual depth of an edge using the ground truth depth map. We evaluate the performance of edge detection using two criteria: edge accuracy, which is the percentage of edges whose estimated depth is within 0.5 disparities of the ground truth, and percentage of missing edges, which is the percentage of edges that cannot find matches (recall that we reject matches that do not agree with locally fitted planes).

Table 3 shows the quantitative evaluation of edge matching. We compare our edge matching algorithm with two variants: only using the orientation term (set the weight for the color difference term defined Equation (2) in to 0) and only using the color term (set the weight for the orientation difference term defined

Equation (1) in to 0). The results shows that both color term and the orientation term improves accuracy of edge matching: using both terms out-perform the algorithm only using the color term by 7% and out-performs the algorithm only using the orientation term by 4%.

Measurement	Matching cost	Aloe	Art	Cloth3	Cones	Dolls	Rocks2	Teddy	Avg.
Accuracy	Only orientation term	66.44	46.83	79.99	34.30	53.29	73.07	29.97	54.84
	Only color term	73.11	55.32	84.59	51.53	61.07	75.99	45.36	63.85
	Both	**76.33**	**60.33**	**87.23**	**55.13**	**65.49**	**81.22**	**47.64**	**67.62**
Percentage of missing edges	Only orientation term	23.61	40.37	15.35	51.46	33.26	21.42	52.60	34.01
	Only color term	9.27	19.71	6.54	**16.45**	13.49	12.60	**22.33**	14.34
	Both	**8.31**	**18.73**	**5.21**	16.98	**12.83**	**9.44**	23.64	**13.59**

Table 3: Evaluation of matching algorithms under two evaluation metrics, accuracy and percentage of missing edges. Two variants of the original matching algorithm (*both*) are tested: only using the orientation term and only using the color term.

Furthermore, to demonstrate that the edge matching finds better sparse matches than keypoint matching, we also replace the edge matching step in our algorithm by the keypoint matching used in Libelas algorithm [25], and feed the matched to keypoint to the rest of the pipeline (including plane detection, superpixel assignment, and depth map refinement). The original keypoint matching algorithm in Libelas only takes two frames as input, and we extend it to also allow multiple frames as input. Table 4 shows the accuracy of estimated depth maps using two different sparse matches as input (our edge matching and keypoint matching used in Libelas), and our edge matching our-performs the keypoint matches by Libelas in all sequences.

Plane estimation. To demonstrate the importance of the plane merging in the plane estimation, we compare estimated depth from the original planes fitted to 32×32 patches and larger planes after the plane merging. Table 5 shows that plane merging significantly improves the quality of reconstructed depth, especially on the sequences with large untextured regions, such as *Teddy* and *Art*. This is because the initially estimated plane equations from local untextured regions are inaccurate. The plane merging step increases the number of edges in each plane, resulting in more accurate plane estimation.

Superpixel assignement. Figure 5 already qualitatively shows that the edge consistency reduces the foreground fattening. To quantitatively evaluate that, we run the superpixel assignment with three different energy functions: only using the edge consistency, only using the photo consistency, and using both consistencies.

Matching algorithms	Aloe	Art	Cloth3	Cones	Dolls	Rocks2	Teddy	Avg.
Libelas: keypoint matching	3.53	4.61	1.41	4.47	6.42	1.37	13.40	5.03
Ours: edge matching	**2.38**	**3.32**	**1.01**	**2.59**	**4.84**	**1.06**	**5.46**	**2.95**

Table 4: Errors (%) of estimated depth using two different algorithms to find sparse matches: our edge matching algorithm the keypoint matching algorithm used in Libelas [25]

Source of planes	Aloe	Art	Cloth3	Cones	Dolls	Rocks2	Teddy	Avg.
Without plane merging	3.33	7.52	1.18	4.76	5.23	1.32	12.87	5.17
With plane merging	**2.38**	**3.32**	**1.01**	**2.59**	**4.84**	**1.06**	**5.46**	**2.95**

Table 5: Errors (%) of estimated depth with/without plane merging

Table 6 shows that even by fitting a smooth surface to matched edge, the algorithm already can recover decent depth maps (that is the result by only using the edge consistency, whose the average error rate is 5.88%) and using the appearance information from input sequences further improves it to 2.91%.

Energy functions	Aloe	Art	Cloth3	Cones	Dolls	Rocks2	Teddy	Avg.
Only edge consistency	4.81	7.17	2.36	6.20	9.43	2.25	8.93	5.88
Only photo consistency	2.20	2.76	1.04	2.84	5.90	**1.03**	**5.12**	2.98
Both	**2.14**	**2.75**	**1.02**	**2.74**	**5.30**	1.10	5.32	**2.91**

Table 6: Errors (%) of estimated depth using different energy functions in the superpixel assignment (Note that in this experiment, we do not run the refinement, so the number of *both* is slightly different than the last column of Table 1)

Refinement. At last, Table 7 shows how the refinement affects the performance of the algorithm. The refinement algorithm is mostly designed to correct small error on curved surfaces (before refinement, the algorithm only estimates a piecewise planar depth map). Therefore, there is a large improvement on sequences with many curved structures, like *Cones* and *Dolls*, especially when using a lower threshold in calculating the accuracy. Even though, the accuracy under the threshold 2.0 slightly decrease when using refinement, it still makes the estimated depth map much smoother, as shown in Figure 6.

Number of frames. Figure 10 shows the effect of reducing the number of input frames on our algorithm. As already discussed in the quantitative analysis, our algorithm still works reasonably well when fewer frames are provided. Even from 3 frames, clean depth maps can still be derived, and errors decrease when more

Threshold	Algorithms	Aloe	Art	Cloth3	Cones	Dolls	Rocks2	Teddy	Avg.
T = 1.0	w.o. refinement	**4.13**	**4.36**	1.87	5.44	11.34	2.29	9.64	5.58
	with refinement	4.20	5.01	**1.56**	**3.81**	**9.26**	**1.66**	**8.88**	**4.91**
T = 2.0	w.o. refinement	**2.14**	**2.75**	1.02	2.74	5.30	1.10	**5.32**	**2.91**
	with refinement	2.38	3.32	**1.01**	**2.59**	**4.84**	**1.06**	5.46	2.95

Table 7: Errors (%) of estimated depth with/without refinement

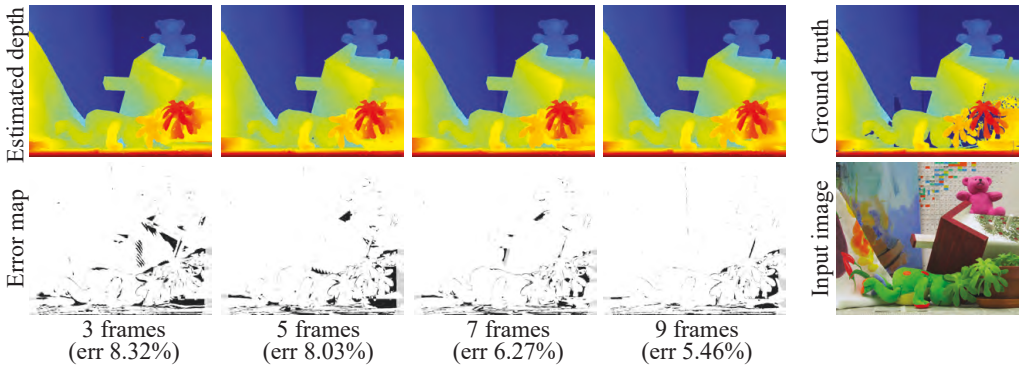


Figure 10: Recovered depth maps using the proposed method with different number of input frames. Errors are shown below each sequence.

frames are used (we do not show the result of using two-frames as input, as it cannot check the consistency in occluded regions).

6.5. View interpolation

One of major usage of depth map is image-based rendering. To demonstrate the quality of our depth, we generate view interpolation result as follows. We pick two frames in the sequence as the left and right reference frames. For each of these two reference frames, we pick 5 to 9 frames around them and use them to recover the depth map in the reference frame. We then render any view points between the left and right frames by warping both color images with the recovered depth maps of those two frames and blending the two warped images using a z-buffer and proportional weighting. The synthesized sequences are available in the supplementary material.

One big advantage of our algorithm is that the recovered depth map has sharp boundaries and is also smooth within objects. It therefore creates fewer artifacts when synthesizing a viewpoint, as shown in Figure 11. Due to foreground fattening, the interpolated image by SGM has missing pieces in the background. For example, both the digit 2 in the top patch and the digit 4 in the bottom patch

Two input frames | Interpolation by SGM (PSNR=28.89dB) | Interpolation by ours (PSNR=30.03dB) | Ground truth

Figure 11: Interpolation results using depth map estimated by ours algorithm and SGM on *middF* dataset. The ground truth image is the actual image captured by the camera located at interpolated viewpoint.

Resolution of Middlebury Dataset	Runtime (seconds)		
	SGM	MCCNN	Ours
Quad resolution (100–200 kP)	**8.2**	16.4	10.1
Half resolution (200–500 kP)	68.1	139.2	**35.8**
Original resolution (1–2 MP)	458.7	1905.4	**170.1**

Table 8: Runtime of three different algorithms: SGM [47] (using all frames), MCCNN [48], and ours. We evaluated on three different resolutions: original, half (the width and height are 1/2 of the original), and quad.

are missing. Such error does not exists in our results, as ours depth has a clear foreground-background boundary. The interpolated image by our depth is almost identical as the ground truth, and has higher PSNR than the interpolated image by SGM depth.

6.6. Runtime

At last, we evaluate the runtime of our algorithm and compare it with two multi-frame stereo algorithms: SGM [47] and MCCNN [48]. All three algorithms are evaluated on a Intel i7 CPU with a GeForce GTX 1080 Ti GPU. Both SGM and ours are running on a single-thread CPU. For MCCNN, the matching costs are calculcated on a single-thread CPU and depth estimation from matching cost are calculated on GPU (there is no CPU implementation for this part). Table 8 shows the runtime of our algorithm on Middlebury dataset with three different resolutions. Our algorithm is much faster than SGM and MCCNN on the original resolution. Morever, our runtime increases linearly with the number of pixels, while others increase superlinearly. This also demonstrates the scalability of our algorithm.

7. Conclusion

In this paper, we have developed a multi-frame stereo algorithm based on matching edges, inferring local slanted plane hypotheses, and computing dense per-pixel disparities using a superpixel-based MRF followed by piecewise continuous relaxation. Our multi-stage pipeline computes the most reliable estimates (edge correspondences and local planar hypotheses) first, and then produces a dense estimate that better preserves depth discontinuities and deals with semi-occluded and textureless regions.

Our experiments demonstrate the importance of edge matching. Edge matching reduces foreground fattening, is more robust to color variation, and provides better initial sparse matches compared with keypoint matching. With matched edges, our approach outperforms alternative approaches when recovering depth from high-resolution multi-frame sequences, particularly when we evaluate errors on *all* the pixels, including semi-occluded regions. This increased accuracy is particularly important in applications such as view interpolation and video editing.

In future work, we plan to extend our approach in several directions. The first is to associate color distribution models with planar depth hypotheses, as is done by Bleyer et al. [27]. The second is to implement a coarse-to-fine approach, which can significantly reduce the computational complexity of the initial edge correspondence stage by re-using local planar hypotheses from coarser levels.

In our current approach, we do not yet fully exploit all of the local occlusion cues that are available in multi-frame sequences. For example, we could use the larger of the two color differences adjacent to edges (Eqn. 2) as a local cue for occlusions and depth discontinuity events. We could also replace the edge-on-boundary edge consistency constraint in Eqn. 9 with one that requires at least one neighboring superpixel to be at the depth of the edge. However, this leads to an MRF formulation with non-submodular terms, requiring a more powerful solver, such as Thuerck et al. [49].

In the longer term, we would like to extend our approach to arbitrary camera motions, investigate alternative (non-patch-based) aggregation strategies such as bilateral filtering [40], and use our edge-based approach to deal with transparent motions and reflections. We believe that our approach can have significant advantages in all of these extended scenarios.

8. Acknowledgements

Part of this work was done when the authors were in Microsoft Research.

References

[1] R. Bolles, H. Baker, D. Marimont, Epipolar-plane image analysis: An approach to determining structure from motion, IJCV 1 (1987) 7–55.

[2] M. Okutomi, T. Kanade, A multiple baseline stereo, IEEE TPAMI 15 (4) (1993) 353–363.

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

[4] H. Baker, Edge based stereo correlation, in: Image Understanding Workshop, 1980, pp. 168–175.

[5] Y. Ohta, T. Kanade, Stereo by intra- and inter-scanline search using dynamic programming, IEEE TPAMI 7 (2) (1985) 139–154.

[6] R. Arnold, Automated stereo perception, Tech. Rep. AIM-351, AI Lab, Stanford University (1983).

[7] V. Venkateswar, R. Chellappa, Hierarchical stereo and motion correspondence using feature groupings, IJCV 15 (3) (1995) 245–269.

[8] R. Collins, A space-sweep approach to true multi-image matching, in: CVPR, 1996, pp. 358–363.

[9] R. Szeliski, R. Weiss, Robust shape recovery from occluding contours using a linear smoother, IJCV 28 (1) (1998) 27–44.

[10] C. Schmid, A. Zisserman, The geometry and matching of lines and curves over multiple views, IJCV 40 (3) (2000) 199–233.

[11] Y. Nakamura, T. Matsuura, K. Satoh, Y. Ohta, Occlusion detectable stereo—occlusion patterns in camera matrix, in: CVPR, 1996, pp. 371–378.

[12] S. B. Kang, R. Szeliski, J. Chai, Handling occlusions in dense multi-view stereo, in: CVPR, 2001, pp. I–103–110.

[13] S. B. Kang, R. Szeliski, Extracting view-dependent depth maps from a collection of images, IJCV 58 (2) (2004) 139–163.

[14] J. Wang, E. Adelson, Representing moving images with layers, IEEE TIP 3 (5) (1994) 625–638.

[15] S. Sinha, D. Scharstein, R. Szeliski, Efficient high-resolution stereo matching using local plane sweeps, in: CVPR, 2013.

[16] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Süsstrunk, SLIC superpixels compared to state-of-the-art superpixel methods, IEEE TPAMI 34 (11) (2012) 2274–2282.

[17] D. Scharstein, Matching images by comparing their gradient fields, in: ICPR, Vol. I, 1994, pp. 572–575.

[18] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, M. Gelautz, Fast cost-volume filtering for visual correspondence and beyond, in: CVPR, 2011, pp. 3017–3024.

[19] I. Jung, J. Sim, C. Kim, S. Lee, Robust stereo matching under radiometric variations based on cumulative distributions of gradients, in: ICIP, 2013, pp. 2082–2085.

[20] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: ECCV, 2004, pp. 25–36.

[21] H. Ishikawa, D. Geiger, Occlusions, discontinuities, and epipolar lines in stereo, in: ECCV, 1998, pp. 232–248.

[22] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, S. Seitz, Occluding contours for multi-view stereo, in: CVPR, 2014, pp. 4002–4009.

[23] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, IJCV 47 (1-3) (2002) 7–42.

[24] A. Bobick, S. Intille, Large occlusion stereo, IJCV 33 (3) (1999) 181–200.

[25] A. Geiger, M. Roser, R. Urtasun, Efficient large-scale stereo matching, in: ACCV, 2010.

[26] M. Bleyer, C. Rother, P. Kohli, Surface stereo with soft segmentation, in: CVPR, 2010, pp. 1570–1577.

[27] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, S. Sinha, Object stereo—joint stereo matching and object segmentation, in: CVPR, 2011, pp. 3081–3088.

[28] C. Vogel, K. Schindler, S. Roth, 3D scene flow estimation with a piecewise rigid scene model, IJCV (2015) 1–28.

[29] A. Chakrabarti, Y. Xiong, S. Gortler, T. Zickler, Low-level vision by consensus in a spatial hierarchy of regions, in: CVPR, 2015, pp. 4009–4017.

[30] K. Yamaguchi, D. McAllester, R. Urtasun, Efficient joint segmentation, occlusion labeling, stereo and flow estimation, in: ECCV, 2014.

[31] K. Yamaguchi, T. Hazan, D. McAllester, R. Urtasun, Continuous Markov random fields for robust stereo estimation, in: ECCV, 2012.

[32] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, A comparison and evaluation of multi-view stereo reconstruction algorithms, in: CVPR, 2006.

[33] A. Knapitsch, J. Park, Q.-Y. Zhou, V. Koltun, Tanks and temples: Benchmarking large-scale scene reconstruction, ACM Transactions on Graphics 36 (4) (2017) 78.

[34] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, A. Geiger, A multi-view stereo benchmark with high-resolution images and multi-camera videos, in: CVPR, 2017.

[35] K. Honauer, O. Johannsen, D. Kondermann, B. Goldluecke, A dataset and evaluation methodology for depth estimation on 4D light fields, in: Asian Conference on Computer Vision, 2016.

[36] J. Canny, A computational approach to edge detection, IEEE TPAMI 8 (6) (1986) 679–698.

[37] W. Freeman, E. Adelson, The design and use of steerable filters, IEEE TPAMI 13 (9) (1991) 891–906.

[38] P. Dollár, C. L. Zitnick, Structured forests for fast edge detection, in: ICCV, 2013, pp. 1841–1848.

[39] C. Zach, Robust bundle adjustment revisited, in: ECCV, 2014, pp. 772–787.

[40] J. Barron, A. Adams, Y. Shih, C. Hernández, Fast bilateral-space stereo for synthetic defocus, in: CVPR, 2015.

[41] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE TPAMI 23 (11) (2001) 1222–1239.

[42] R. Szeliski, Locally adapted hierarchical basis preconditioning, ACM TOG 25 (3) (2006) 1135–1143.

[43] D. Scharstein, R. Szeliski, Middlebury stereo vision page, `http://vision.middlebury.edu/stereo/`.

[44] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the KITTI vision benchmark suite, in: CVPR, 2012.

[45] S. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, A comparison and evaluation of multi-view stereo reconstruction algorithms, in: CVPR, 2006.

[46] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, U. Thoennessen, On benchmarking camera calibration and multi-view stereo for high resolution imagery, in: CVPR, 2008.

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

[48] J. Zbontar, Y. LeCun, Stereo matching by training a convolutional neural network to compare image patches, Journal of Machine Learning Research 17 (1-32) (2016) 2.

[49] D. Thuerck, M. Waechter, S. Widmer, M. von Buelow, P. Seemann, M. E. Pfetsch, M. Goesele, A fast, massively parallel solver for large, irregular pairwise markov random fields, in: Eurographics, 2016.