Learning to Fuse Proposals from Multiple Scanline Optimizations in Semi-Global Matching

Johannes L. Schönberger[1,2(✉)], Sudipta N. Sinha[1], and Marc Pollefeys[1,2]

[1] Microsoft, Redmond, USA
[2] Department of Computer Science, ETH Zürich, Zürich, Switzerland
jsch@inf.ethz.ch

Abstract. Semi-Global Matching (SGM) uses an aggregation scheme to combine costs from multiple 1D scanline optimizations that tends to hurt its accuracy in difficult scenarios. We propose replacing this aggregation scheme with a new learning-based method that fuses disparity proposals estimated using scanline optimization. Our proposed SGM-Forest algorithm solves this problem using per-pixel classification. SGM-Forest currently ranks 1st on the ETH3D stereo benchmark and is ranked competitively on the Middlebury 2014 and KITTI 2015 benchmarks. It consistently outperforms SGM in challenging settings and under difficult training protocols that demonstrate robust generalization, while adding only a small computational overhead to SGM.

1 Introduction

Semi-Global Matching (SGM) is a popular stereo matching algorithm proposed by Hirschmüller [15] that has found widespread use in applications ranging from 3D mapping [17,34,39,40], robot and drone navigation [19,38], and assisted driving [8]. The technique is efficient and parallelizable and suitable for real-time stereo reconstruction on FPGAs and GPUs [2,9,19]. SGM incorporates regularization in the form of smoothness priors, similar to global stereo methods but at lower computational cost. The main idea in SGM is to approximate a 2D Markov random field (MRF) optimization problem with several independent 1D scanline optimization problems corresponding to multiple canonical scanline directions in the image (typically 4 or 8). These 1D problems are optimized exactly using dynamic programming (DP) by aggregating matching costs along the multi-directional 1D scanlines. The costs of the minimum cost paths for the various directions are then summed up to compute a final aggregated cost per pixel. Finally, a winner-take-all (WTA) strategy is used to select the disparity with the minimum aggregated cost at each pixel.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-01261-8_45) contains supplementary material, which is available to authorized users.

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11217, pp. 758–775, 2018.
https://doi.org/10.1007/978-3-030-01261-8_45

Summation of the aggregated costs from multiple directions and the final WTA strategy are both ad-hoc steps in SGM that lack proper theoretical justification. The summation was originally proposed to reduce 1D streaking artifacts [15] but is ineffective for weakly textured slanted surfaces and also generally inadequate when multiple scanline optimization solutions are inconsistent.

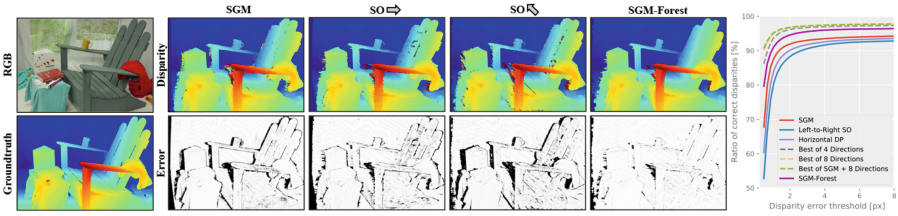

Fig. 1. Fusing Multiple Scanline Proposals. *Left:* Visualization of disparity maps from SGM, two (out of 8) scanline optimizations (SO) and our proposed SGM-Forest method. While SGM is more accurate than each SO on the whole image, each SO solution is better in some specific areas. SGM-Forest identifies the best SO proposal at each pixel and produces the best overall result. *Right:* Error plots for SGM, SO and SGM-Forest solutions (solid line) and upper bounds for oracles making optimal selections (dotted line). In this example, SGM-Forest gets close to the upper bounds.

Our main motivation in this work is to devise a better strategy to fuse 1D scanline optimization costs from multiple directions. We argue that the scanline optimization solutions should be considered as independent disparity map proposals and the WTA step should be replaced by a more general fusion step. Figure 1 shows two of the eight scanline optimization solutions for the ADIRONDACK pair from the Middlebury 2014 dataset [35]. While both solutions suffer from directional bias due to their respective propagation directions, each solution is accurate in certain image regions where the other one is inaccurate. For example, the horizontal pass produces accurate disparities near the left occlusion boundaries of the chair, whereas the diagonal pass performs better on the right occlusion edges. In those regions, the final SGM solution is slightly worse. The error plot in Fig. 1 quantifies this observation for the entire image. Whereas SGM is more accurate than each scanline optimization individually, the *joint accuracy* of all scanlines is much higher than SGM. Here, joint accuracy refers to a theoretical upper bound of the achievable accuracy of an oracle, which has access to ground truth and selects the best out of all the scanline solution proposals.

Based on this insight, we formulate the fusion step as the task of selecting the best amongst all the scanline optimization proposals at each pixel in the image. We propose to solve this task using supervised learning. Our method, named *SGM-Forest*, uses a per-pixel random forest classifier. As shown in Fig. 1, it gets close to the theoretical upper bound and significantly outperforms SGM.

The per-pixel classifier in SGM-Forest is trained on a low-dimensional input feature that encodes a sparse set of aggregated cost samples. Specifically, these

cost values are sampled from the cost volumes computed during the scanline optimization passes. The sampling locations correspond to the disparity candidates for all scanline directions at each pixel. In fact, the proposals need not be limited to the usual scanline directions. Including the SGM solution and two horizontal scanline optimization solutions from the right image as additional proposals improves accuracy further. We train and evaluate the forest using ground truth disparity maps provided by stereo benchmarks [35,37,41]. At test time, the random forest predicts the disparity proposal to be selected at each pixel. Inference is fast and parallelizable and thus has small overhead. The forest automatically outputs per-pixel posterior class probabilities from which suitable confidence maps are derived, for use in a final disparity refinement step.

Thus, the main contribution in this paper is a new, efficient learning-based fusion method for SGM that directly predicts the best amongst all the 1D scanline optimization disparity proposals at each pixel based on a small set of scanline optimization costs. SGM-Forest uses this fusion method instead of SGM's sum-based aggregation and WTA steps and our results shows that it consistently outperforms SGM in many different settings. We evaluate SGM-Forest on three stereo benchmarks. Currently, it is ranked 1st on ETH3D [41] and is competitive on Middlebury 2014 [35] and KITTI 2015 [10]. We run extensive ablation studies and show that our method is extremely robust to dataset bias. It outperforms SGM even when the forests are trained on datasets from different domains.

2 Related Work

In this section, we review SGM and learning-based methods for stereo. We then compare and contrast our proposed SGM-Forest to closely related works.

SGM was built on top of earlier methods such as 1D scanline optimization [29,37,50] and dynamic programming stereo [46] with a new aggregation scheme to fix the lack of proper 2D regularization in those methods. However, a proper derivation of the aggregation step remained elusive until Drory et al. [6] showed its connection to non-loopy belief propagation on a special graph structure. Veksler [47] and Bleyer et al. [3] advanced dynamic programming stereo to tree structures connecting all pixels, but those methods have not been widely adopted. SGM has been extended to improve speed and accuracy [1,2,7,9,13,14,16,19], reduce memory usage [18,19,23], and to compute optical flow [45,49].

Scharstein and Pal [36] were one of the first to use learning in stereo. They trained a conditional random field (CRF) on Middlebury 2005–06 datasets to model the relationship between the CRF's penalty terms and local intensity gradients in the image. The KITTI and Middlebury 2014 [10,35] benchmarks encouraged much work on learning. In particular, CNNs have been trained to compute robust matching costs [5,25,48]. Zbontar and Lecun were the first; they proposed MC-CNN [48] and reported higher accuracy when using MC-CNN in conjunction with SGM for regularization and additional post-processing steps. Newer methods combined MC-CNN with better optimization but as a

result are much slower. The method of Taniai et al. [44] uses iterative graph cut optimization and MC-CNN-acrt [48] and is the current state of the art on Middlebury.

End-to-end training of CNNs is nowadays popular on KITTI [11,21,27,30] but is almost never tested on Middlebury. In one rare case, moderate results were reported [22]. In contrast, our method generalizes across three benchmarks [10,35,41] on which it consistently outperforms baseline SGM. Furthermore, we train three separate models on Middlebury 2005–06, KITTI, and ETH3D. All three outperform SGM when tested on the Middlebury 2014 training set. SGM-Net [42] is a CNN-based method for improving SGM. SGM-Net performs more accurate scanline optimization by using a CNN to predict the parameters of the underlying scanline optimization objective. In contrast, we use regular scanline optimization but propose a learning-based fusion step using random forests.

Stereo matching has been solved by combining multiple disparity maps using MRF fusion moves [4,24,44]. Fusion moves are quite general, but computationally expensive and need many iterations. This makes them slow. Alternatively, multiple disparity maps can also be fused using learning, based on random forests [43] and CNNs [32]. Other methods first predict confidence maps [20], often via learning [12,26,31,33], and then use the predicted confidence values in a greedy fashion to combine multiple solutions. Drory et al. [6] proposed a different uncertainty measure for SGM but do not show how to use it. Unlike MRF fusion moves [24], our fusion method is not general. It combines a specific number and specific type of proposals but does so in a single efficient step.

Michael et al. [28] and Poggi and Mattoccia [33] (SGM-RF) proposed replacing SGM's sum-based aggregation with a weighted sum, setting smaller weights in areas with 1D streaking artifacts. The former work [28] proposes using global weights per scanline direction. SGM-RF [33] is more effective as it predicts per-pixel weights for each scanline direction using random forests based on disparity-based features. However, SGM-RF was not evaluated on the official test sets of the Middlebury 2014 and KITTI 2015 benchmarks. Mac Aodha et al. [26] also used random forests to fuse optical flow proposals using flow-based features.

Our SGM-Forest differs from these methods in several ways. First, it avoids predicting confidence separately for each proposal [26,33] but instead directly predicts the best proposal at each pixel. The forest is invoked only once at each pixel and has information from all the scanline directions. This makes inference more effective. Furthermore, the features used by our forest are directly obtained by sampling the aggregated cost volumes of each scanline optimization problem at multiple selective disparities. This is much more effective than handcrafted disparity-based features [33,43]. Finally, our confidence maps derived from posterior class probabilities are normalized and hence better for refining the disparities during post-processing. Haeusler et al. [12] aim to detect unreliable disparities and suggest adding SGM's aggregated (summed) costs to their handcrafted disparity-based features. In contrast, we focus on fusing multiple proposals and propose to sample all the cost volumes for each independent scanline optimization at multiple disparities to better exploit contextual information.

3 Semi-Global Matching

We now review SGM as proposed by Hirschmüller [17] for approximate energy minimization of a 2D Markov Random Field (MRF)

$$E(D) = \sum_{\mathbf{p}} C_{\mathbf{p}}(d_{\mathbf{p}}) + \sum_{\mathbf{p},\mathbf{q}\in\mathcal{N}} V(d_{\mathbf{p}}, d_{\mathbf{q}}), \tag{1}$$

where $C_{\mathbf{p}}(d)$ is a unary data term that encodes the penalty of assigning pixel $\mathbf{p} \in \mathbb{R}^2$ to disparity $d \in \mathcal{D} = \{d_{\min}, \ldots, d_{\max}\}$. The pairwise smoothness term $V(d, d')$ penalizes disparity differences between neighboring pixels $\mathbf{p}$ and $\mathbf{q}$. In SGM, the term V is chosen to have the following specific form

$$V(d, d') = \begin{cases} 0 & \text{if } d = d' \\ P_1 & \text{if } |d - d'| = 1 \\ P_2 & \text{if } |d - d'| \geq 2\,, \end{cases} \tag{2}$$

which favors first-order smoothness, *i.e.*, has a preference for fronto-parallel surfaces. Minimizing the 2D MRF is NP-hard. Therefore, SGM instead solves multiple scanline optimization problems, each of which involves solving the 1D version of Eq. 1 along 1D scanlines in 8 cardinal directions $\mathbf{r} = \{(0, 1), (0, -1), (1, 0), ...\}$. For each direction $\mathbf{r}$, SGM computes an aggregated matching cost

$$L_{\mathbf{r}}(\mathbf{p}, d) = C_{\mathbf{p}}(d) + \min_{d' \in \mathcal{D}} (L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}\,, d') + V(d, d')). \tag{3}$$

The definition of $L_{\mathbf{r}}(\mathbf{p}, d)$ is recursive and is typically started from a pixel on the image border. An aggregated cost volume $S(\mathbf{p}, d)$ is finally computed by summing up the eight individual aggregated cost volumes

$$S(\mathbf{p}, d) = \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d)\,. \tag{4}$$

The final disparity map is obtained using a WTA strategy by selecting per-pixel minima in the aggregated cost volume

$$d_{\mathbf{p}} = \arg\min_{d} S(\mathbf{p}, d). \tag{5}$$

The steps in Eqs. 4 and 5 are accurate when the costs from different scanline directions are mostly consistent wrt. each other. However, these steps are likely to fail as the scanlines become more inconsistent. To overcome this problem, we propose a novel fusion method to robustly compute the disparity $d_{\mathbf{p}}$ from the multiple scanline costs $L_{\mathbf{r}}(\mathbf{p}, d)$.

4 Learning to Fuse Scanline Optimization Solutions

We start by analyzing some difficult examples for scanline optimization in order to motivate our fusion method and then describe the method in detail.

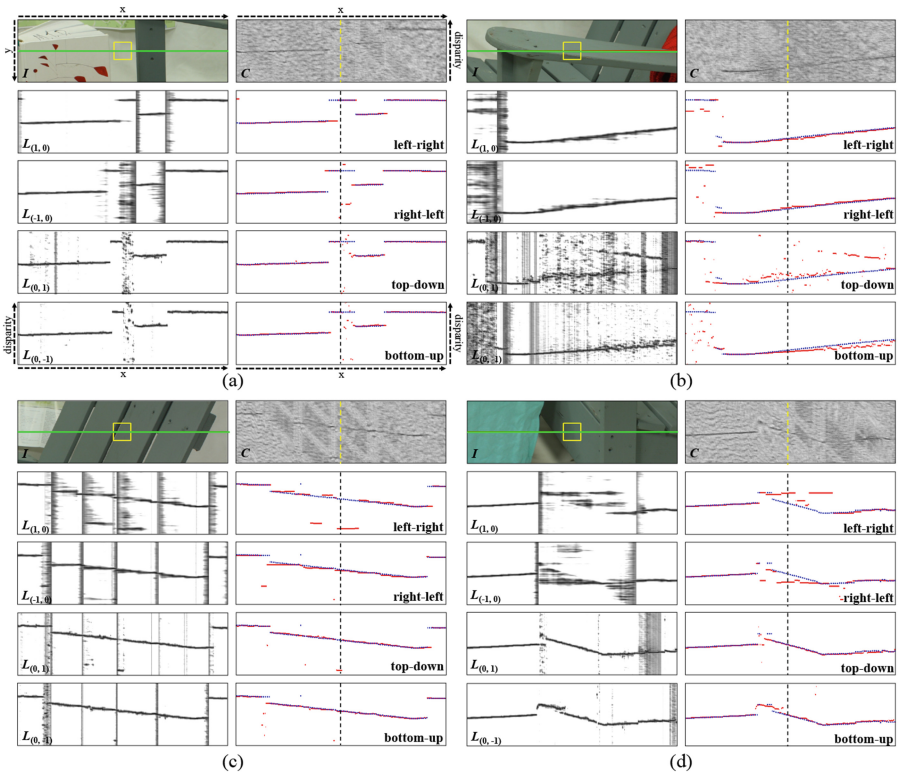

Fig. 2. 1D Scanline Optimization Costs. Each of the four subfigures shows the following – *Top Left:* Image and reference scanline section in green centered around yellow patch. *Top Right:* x–d slice of unary cost volume C along the reference scanline and ray of reference patch center in yellow. *Bottom:* Aggregated costs $L_{\mathbf{r}}$ for four scanline directions on the left and the corresponding disparities on the right. The WTA solution is shown in red whereas the ground truth disparity is in blue. (Color figure online)

4.1 Scanline Optimization Analysis

Figure 2 shows four scanlines from the left ADIRONDACK image with the corresponding x–d slices of the the unary cost C and the four horizontal and vertical aggregated scanline costs $L_{\mathbf{r}}$ alongside their respective WTA solutions. Notice the patterns in the $L_{\mathbf{r}}$ cost slices for the different passes. When the smoothness prior is effective, the noisy unary costs get filtered, producing strong minima at the correct disparities. However, when the unary costs are weak and the prior is ineffective, multiple noisy minima are present or the minimum is at an incorrect location. We now investigate these problematic cases in further detail.

Weak Texture. Figure 2(a)–(d) focus on weakly textured image patches. Whenever the unary cost is weak, the smoothness prior in the 1D optimization favors

propagating several equally likely disparity estimates along the propagation direction. This effect is seen clearly on the vertical wooden plank in Fig. 2(d) in the horizontal passes. Here, the left-right propagation continues the solution from the left occlusion boundary to the right, while the right-left solution continues from the corner of the chair to the left. In contrast, the two vertical passes are in agreement at the correct disparity as the surface *along that propagation direction* is indeed fronto-parallel.

Slanted Surface. Figure 2(b), (c), (d) show examples of weakly textured slanted surfaces, where the 1D scanline solutions are typically biased and jump at random pixel locations, leading to inconsistent solutions in different scanlines. A prominent example is the arm rest in Fig. 2(b), where the left-right pass underestimates the disparity, whereas the right-left and bottom-up passes overestimate the disparity. In this case, there is no clear outlier in the solution but final cost summation leads to a biased estimate. Notice also the asymmetry in the two vertical passes where the bottom-up direction has a much more consistent solution while the top-down solution jumps at random locations. On weakly textured slanted surfaces, adjacent scanlines solutions are mostly inconsistent leading to noisy disparity maps and well-known streaking artifacts.

Occlusion. Figure 2(a) is centered around a region which is occluded in the right image. In this case, the unary cost is invalid and the only pass producing a correct prediction is the left-to-right direction. Here, the occluded surface is fronto-parallel and the smoothness prior is likely to propagate the correct disparity to the occluded region. Typically, only a small subset of scanlines results are correct in occluded areas, whereas SGM's standard cost summation is not robust and therefore produces gross outliers (see Fig. 1).

Repetitive Structure. The wooden planks on the chair's backrest in Fig. 2(c) are repetitive and produce multiple ambiguous local cost minima. In this example, the solutions of the left-right and top-down directions are incorrectly estimated, since the centered patch is almost identical to the symmetric patch on the right-most wooden plank. Notice also that the right-left and bottom-up directions are much less susceptible to this specific ambiguity problem.

These examples show that the joint distribution of aggregated costs over the disparity range at each pixel appears to provide strong clues about which scanline proposal or which subset of proposals are likely to be correct. This insight forms the basis of our fusion model which is described next.

4.2 Definition of Fusion Model

The disparities of the different scanline solutions are often inconsistent, especially in areas of weak data cost. Yet, in almost all cases there is at least one scanline that is either correct or is very close to the correct solution. The main challenge for robust and accurate scanline fusion is to identify the scanlines which agree on the correct estimate. In our proposed approach, we cast the fusion of scanlines as a classification problem that chooses the optimal estimate from the given set

of candidate scanlines. Typically, the pattern at which specific scanlines perform well is consistent and repeatable. We aim to encode these patterns into rules that can identify the correct solution from a given set of candidate solutions. However, manually hand-crafting these rules is unfeasible and error-prone, which is why we resort to automatically learning these rules from training data in a supervised fashion. To facilitate the learning of these rules, we provide the model with discriminative signals that allow for a robust and efficient disparity prediction. Our proposed model takes sparse samples from a set of proposal cost volumes $K_n(\mathbf{p}, d)$ (*e.g.*, the optimized scanline costs $L_{\mathbf{r}}(\mathbf{p}, d)$) and concatenates them into a per-pixel feature vector $\mathbf{f_p}$. This feature vector is then fed into a learned model that predicts a disparity estimate $\hat{d}_{\mathbf{p}}$ together with a posterior probability $\hat{\rho}_{\mathbf{p}}$, which we use as a confidence measure for further post-processing.

More specifically, our model is defined as $(\hat{d}_{\mathbf{p}}, \hat{\rho}_{\mathbf{p}}) = F(f_{\mathbf{p}})$ with $d_{\mathbf{p}} \in \mathbb{R}_0^+$, $\rho_{\mathbf{p}} \in [0, 1]$, and $\mathbf{f_p} \in \mathbb{R}^{N+N^2}$, where N is the number of proposal costs $K_n(\mathbf{p}, d)$. For all $n = 1...N$ proposals $K_n(\mathbf{p}, d)$, the feature $\mathbf{f_p}$ stores the location of its per-pixel WTA solution $d^*_{\mathbf{p}}(n) = \arg\min_d K_n(\mathbf{p}, d)$ and the corresponding costs $K_m(\mathbf{p}, d^*_{\mathbf{p}}(n))$ in all proposals $m = 1...N$. Overall, the feature is composed of N WTA solutions and the N^2 sparsely sampled costs. For each disparity proposal $d^*_{\mathbf{p}}(n)$, we thereby encode its relative significance wrt. the other proposals in a compact representation. The intuition is that when multiple proposals agree, their minima $d^*_{\mathbf{p}}(n)$ are close and their respective costs $K_m(\mathbf{p}, d^*_{\mathbf{p}}(n))$ are low.

Note that the naïve approach of concatenating the per-pixel costs of all proposals into a feature vector is not feasible for two reasons. First, we want a lightweight feature representation and model with small runtime overhead wrt. regular SGM. However, the naïve approach would result in a very high-dimensional feature representation of size $N \cdot |\mathcal{D}|$ (*e.g.*, $8 \cdot 256 = 2048$ for 256 disparity candidates and 8 scanlines), which would require a complex model and eliminate the computational efficiency of SGM. In contrast, our proposed feature vector is only $8+8^2 = 72$-dimensional in case of 8 scanline proposals. Second, we strive to learn a generalizable model, which uses a fixed-size feature representation during training and inference even though the disparity range $\mathcal{D}$ may vary between scenes. In summary, our proposed feature encodes discriminative signals for our classification task without sacrificing efficiency, compactness, or accuracy.

4.3 Random Forests for Disparity and Confidence Prediction

Given ground truth disparities, there are many ways to learn the model $F(\mathbf{f_p})$ using supervised learning. The first principal design decision is whether to pose the problem as a classification or regression task. Arguably, classification problems are often considered as easier to solve. As shown in Fig. 1, at least one of the different scanline solutions is often accurate. We therefore chose to formulate a N-class classification task that predicts the best solution from the set of candidates $d^*_{\mathbf{p}}(n)$. This approach gave much better results than modeling the problem as a regression task. The second principal design decision is the specific type of classifier to use, *e.g.*, k-NN, support vector machines, decision trees,

Table 1. Validation performance for non-occluded pixels on the Middlebury 2014 training set (15 half resolution pairs). Rows 1–5 show results for SGM baselines. Rows 6–14 report ablation studies for SGM-Forest. Bottom three rows show results for the best SGM-Forest setting, trained on different datasets. Letters M, K, and E refer to Middlebury 2005–06, KITTI, and ETH3D, respectively. The matching cost is always MC-CNN-acrt. Runtimes exclude matching cost and timed on same CPU.

Method	Left View Scanlines	Right View Scanlines	Filtering	Training Dataset	bad 0.5px [%]	bad 1px [%]	bad 2px [%]	bad 4px [%]	Time [s]
SGM	all			–	50.85	23.04	8.89	5.16	**3.0**
SGM – $\min_d L_{\mathbf{r}}(\mathbf{p}, d)$	all			–	52.18	25.45	11.81	7.79	3.1
SGM – $\min_d \text{median}_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d)$	all			–	63.25	31.81	9.90	8.24	3.2
SGM-SVM	all			M	48.68	21.88	8.57	5.09	323.7
SGM-MLP	all			M	47.77	21.83	8.53	5.08	21.0
SGM-Forest	horiz+vert			M	47.36	21.30	8.49	4.93	5.7
	top-down			M	47.45	21.20	8.38	4.94	5.8
	bottom-up			M	47.65	21.54	8.54	4.98	5.8
	all			M	46.67	20.85	8.40	4.89	6.1
	all	•		M	46.49	20.81	8.23	4.72	6.3
	all	•	•	E	46.80	20.32	8.17	4.79	8.2
	all	•	•	K	46.48	20.45	8.09	4.81	8.2
	all	•	•	M	**46.08**	**19.99**	**7.78**	**4.41**	8.2

neural nets, etc. In our experiments, random forests provided the best trade-off between accuracy and efficiency (see Sect. 5.2 and Table 1).

At test time, we first perform 1D scanline optimization to construct the proposal cost volumes $K_n(\mathbf{p}, d)$, from which we build the per-pixel feature vectors $\mathbf{f}_\mathbf{p}$. In the second stage, we simply feed the feature vectors $\mathbf{f}_\mathbf{p}$ of all pixels $\mathbf{p}$ through our model to obtain a posterior probability $\rho_\mathbf{p}(n)$ for each proposal n. We select the proposal with the maximum posterior probability $n^*_\mathbf{p} = \arg\max_n \rho_\mathbf{p}(n)$ as our initial disparity estimate $d^*_\mathbf{p}(n^*)$ for pixel $\mathbf{p}$. To further refine this initial estimate, we find the subset of disparity proposals close to the initial estimate and their corresponding posteriors:

$$\mathcal{D}^*_\mathbf{p} = \{(d^*_\mathbf{p}(k), \rho_\mathbf{p}(k)) \,|\, k = 1...N \wedge |d^*_\mathbf{p}(k) - d^*_\mathbf{p}(n^*)| < \epsilon_d\} \tag{6}$$

When multiple scanlines agree on a solution, the inlier set $\mathcal{D}^*_\mathbf{p}$ contains multiple elements, even for small disparity thresholds ϵ_d. The final per-pixel disparity estimate $\hat{d}_\mathbf{p}$ and confidence measure $\hat{\rho}_\mathbf{p}$ are computed as

$$\hat{d}_\mathbf{p} = \frac{\sum_k \rho_\mathbf{p}(k)\, d^*_\mathbf{p}(k)}{\sum_k \rho_\mathbf{p}(k)} \quad \text{and} \quad \hat{\rho}_\mathbf{p} = \sum_k \rho_\mathbf{p}(k) \tag{7}$$

Note that the final disparity estimate has sub-pixel precision. Moreover, all steps are fully parallelizable on the pixel level and therefore suitable for real-time FPGA implementations (see Sects. 5.2 and 5.5). Next, we will describe our spatial edge-aware filtering scheme for disparity refinement.

4.4 Confidence-Based Spatial Filtering

The random forest produces a per-pixel estimate for disparity and confidence. In a final filtering step, we now enhance the spatial smoothness of the disparity and confidence maps. Towards this goal, we define the adaptive local neighborhood

$$\mathcal{N}_{\mathbf{p}} = \{\mathbf{q} \,|\, \|\mathbf{q}-\mathbf{p}\| < \epsilon_{\mathbf{p}} \wedge \hat{\rho}_{\mathbf{q}} > \epsilon_{\rho} \wedge |I(\mathbf{p}) - I(\mathbf{q})| < \epsilon_I\} \tag{8}$$

centered around each pixel $\mathbf{p}$, where $I(\mathbf{q})$ is the image intensity at pixel $\mathbf{q}$. The filtered disparity and confidence estimates are finally given as $\bar{d}_{\mathbf{p}} = \text{median}\, \hat{d}_{\mathbf{q}}$ and $\bar{\rho}_{\mathbf{p}} = \text{median}\, \hat{\rho}_{\mathbf{q}}$ with $\mathbf{q} \in \mathcal{N}_{\mathbf{p}}$. The filter essentially computes a median on the selective set of neighborhood pixels $\mathcal{N}_{\mathbf{p}}$ which have high confidence and similar color as the center pixel $\mathbf{p}$.

5 Experiments

We report a thorough evaluation of SGM-Forest on three stereo benchmarks – Middlebury 2014, KITTI 2015, and ETH3D 2017 [10,35,41]. Our evaluation protocol contrasts to most top-ranked stereo methods which often evaluate only on one benchmark [11,21,27,30,42,44]. In all our experiments, SGM-Forest outperforms SGM by a significant margin and ranks competitively against the state-of-the-art learning-based and global stereo methods, which are computationally more expensive. It also robustly generalizes across different dataset domains.

5.1 Implementation Details

Scanline Optimization and SGM. To facilitate an unbiased comparison, we use the same SGM implementation throughout all experiments. We compare three different matching costs (NCC, MC-CNN-fast [48], MC-CNN-acrt [48]) as the unary term C, which is quantized to 8 bits for reduced memory usage using linear rescaling to the range $[0, 255]$. Image intensities are given in the range $[0, 255]$. For NCC, we use a patch size of 7×7. We follow standard procedure and improve the right image rectification using sparse feature matching before computing the matching cost. The smoothness term $V(d, d')$ uses the constant parameters $P_1 = 100$ and $P_2 = P_1(1 + \alpha e^{-|\Delta I|/\beta})$, where $\alpha = 8$, $\beta = 10$, and ΔI is the intensity difference between neighboring pixels.

SGM-Forest. In all our experiments, we train random forests with 128 trees, a maximum depth of 25, and the Gini impurity measure to decide on the optimal data split. We set $\epsilon_d = 2$, $\epsilon_\rho = 0.1$, $\epsilon_{\mathbf{p}} = 5$, and $\epsilon_I = 10$. These optimal parameters were decided using parameter grid search and 3-fold cross validation on the Middlebury 2014 training scenes. For generalization across different disparity ranges between training and test datasets, we normalize to relative disparities prior to the extraction of the feature $\mathbf{f}_{\mathbf{p}}$ using the average of the input disparity proposals $d^*_{\mathbf{p}}(n)$. The relative disparity estimates are then denormalized to achieve absolute disparities. To showcase the generalization robustness of our

approach, we train and evaluate our SGM-Forest on different dataset combinations. In all settings, the training and test scenes are non-overlapping and we provide a detailed list of training/test splits in the supplementary material. For learning our SGM-Forest model, we sample a maximum of 500 K random pixels with ground-truth disparity uniformly in each training image.

5.2 Ablation Study

We now evaluate several aspects of our algorithm using an extensive ablation study summarized in Tables 1 and 2 (full tables in the supplementary material).

Table 2. This table shows the validation performance for non-occluded pixels using 3-fold cross-validation for different matching costs and datasets at different error thresholds. Our method (SGM-F.) outperforms baseline SGM in all settings.

		Middlebury 2014				KITTI 2015				ETH3D 2017			
Datacost	**Method**	0.5px	1px	2px	4px	0.5px	1px	2px	4px	0.5px	1px	2px	4px
NCC	SGM	54.15	28.59	15.23	10.14	59.70	32.28	13.09	6.17	30.94	14.78	8.62	5.67
	SGM-F.	**50.06**	**25.29**	**12.55**	**8.08**	**51.61**	**24.74**	**9.22**	**4.17**	**21.14**	**10.28**	**5.59**	**3.67**
MC-CNN-fast	SGM	51.22	23.49	10.58	6.85	57.53	29.82	11.28	4.80	24.70	8.56	4.14	2.57
	SGM-F.	**48.73**	**22.24**	**9.55**	**5.91**	**50.25**	**22.98**	**7.88**	**3.28**	**16.31**	**6.08**	**3.04**	**1.94**
MC-CNN-acrt	SGM	50.85	23.04	8.89	5.16	56.27	26.90	7.41	3.00	37.46	14.44	7.17	4.72
	SGM-F.	**46.08**	**19.99**	**7.78**	**4.41**	**46.16**	**18.82**	**5.76**	**2.56**	**26.26**	**11.05**	**6.56**	**4.71**

SGM Baseline. We compare our SGM baseline against two simple methods that robustify Eqs. 4 and 5 (see Table 1): SGM – $\min_d L_{\mathbf{r}}(\mathbf{p}, d)$ selects the scanline solution with minimum cost as the disparity estimate, while SGM – $\min_d \text{median}_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d)$ uses the robust median instead of summation for aggregating the costs from multiple scanlines. Both methods perform worse than baseline SGM, underlining the need for a more sophisticated fusion approach.

Input Proposals. The input to our algorithm is a set of proposal cost volumes $K_n(\mathbf{p}, d)$. As demonstrated in Fig. 1, a single scanline performs worse than SGM while the best of multiple scanlines is significantly better. In fact, our method is general and the input proposals to our system need not be limited to the canonical 1D scanline optimizations. We always consider the regular SGM cost volume $S(\mathbf{p}, d)$ as a proposal. Using only this proposal leads to a trivial 1-class classification problem and is equivalent to running baseline SGM (see Table 1). Adding the four horizontal and vertical scanlines from the left image as proposals improves the accuracy significantly, which is further boosted by adding the remaining 4 diagonal scanlines. Using only scanlines that propagate in the five top-down or five bottom-up directions degrades performance slightly but is still much better than regular SGM and enables real-time implementation of our algorithm on an FPGA [19]. We also experimented with running two horizontal scanline optimizations on the right image and warping the results to the left view to be used as two additional proposals. This is because the occluded pixels

in the left image are invisible in the right image and the left occlusion edges are usually more accurately recovered in the right disparity map. These additional proposals provide a small but consistent improvement.

Classification Model. In Sect. 4.3, we argued that, for our task, random forests provide the best trade-off in terms of accuracy and efficiency. We experimented with many different classification models, including k-NN search, SVMs, (gradient boosted) decision trees, AdaBoost, neural nets, etc. In Table 1, we show results for two other well-performing models: SGM-SVM uses a linear SVM classifier and SGM-MLP is a multi-layer perceptron using 3 hidden layers with ReLU activation and twice the neurons after each layer followed by a final softmax layer for classification. SGM-MLP outperforms the SGM baseline but has slightly lower accuracy and efficiency on the CPU than SGM-Forest.

Table 3. Middlebury Benchmark. *Left:* Official results for the top 10 performing methods using MC-CNN-acrt for our SGM-Forest. Our method achieves the best runtime among the top performing methods. *Right:* Inofficial results on the training scenes trained on Middlebury 2005–06 using MC-CNN-fast. SGM-Forest with MC-CNN-fast outperforms baseline SGM with MC-CNN-acrt but is an order of magnitude faster.

Middlebury 2014 (MC-CNN-acrt)					
Method	**non-occl.**		**all**		**Time**
LocalExp	5.43%	#1	11.7%	#1	881s
3DMST	5.92%	#2	12.5%	#3	174s
MC-CNN+TDSR	6.35%	#2	12.1%	#3	657s
PMSC	6.71%	#4	13.6%	#4	599s
LW-CNN	7.04%	#5	17.8%	#15	314s
MeshStereoExt	7.08%	#6	15.7%	#9	161s
FEN-D2DRR	7.23%	#7	16.0%	#11	121s
APAP-Stereo	7.26%	#8	13.7%	#5	131s
SGM-Forest	**7.37%**	**#9**	**15.5%**	**#8**	**88s**
NTDE	7.44%	#10	15.3%	#7	152s

Middlebury 2014 (MC-CNN-fast)					
Method	**non-occl.**		**all**		**Time**
LocalExp	6.52 %	#1	12.1%	#1	846s
3DMST	7.08 %	#2	12.9%	#2	167s
APAP-Stereo	7.53%	#3	14.3%	#6	117s
FEN-D2DRR	7.89%	#4	14.1%	#4	73s
...					
MC-CNN-acrt	10.1%	#12	19.7%	#20	106s
...					
SGM-Forest	**11.1%**	**#19**	**17.8%**	**#14**	**9s**
...					
MC-CNN-fast	11.7%	#21	21.5%	#27	1s

Filtering. The final step in our algorithm is the confidence-based spatial filtering of the disparity and confidence maps. While the biggest accuracy improvement stems from the initial fusion step (see Table 1), the final filtering further improves the results by eliminating spatially inconsistent outliers.

Efficiency. The reported runtimes in Table 1 show only a small computational overhead of SGM-Forest and our proposed filtering over baseline SGM, enabling a potential real-time implementation on the GPU or FPGA (see Sect. 5.5). Note that the runtimes exclude the matching cost computation, *i.e.*, the overhead of SGM-Forest becomes negligible if, for example, MC-CNN-acrt is used.

Generalization and Robustness. All results in Table 1 were obtained by training on Middlebury 2005–06 and evaluating on Middlebury 2014, which already demonstrates good generalization properties. Note that Middlebury 2014 images are much more challenging than those in Middlebury 2005–06. Moreover, we also evaluate cross-domain generalization by training on KITTI (outdoors) and ETH3D (outdoors and indoors) and evaluating on Middlebury 2014 (indoors). In both cases, our approach achieves almost the same performance as

compared to training on Middlebury. Table 2 shows that SGM-Forest improves over baseline SGM in every single metric irrespective of matching cost and dataset. In contrast to most learning-based methods, we demonstrate that our learned fusion approach is general and extremely robust across different domains and settings: SGM-Forest performs well outdoors when trained on indoor scenes, handles different image resolutions, disparity ranges and diverse matching costs, and consistently outperforms baseline SGM by a large margin.

5.3 Benchmark Results

Unlike most existing methods, we evaluate SGM-Forest on three benchmarks and achieve competitive performance wrt. the state of the art. For all benchmark submissions, we use the best setting found in our ablation study, *i.e.*, we include 8 (and 2) proposals from the left (and right) view and run disparity refinement.

Table 4. KITTI and ETH3D Benchmarks. *Left:* KITTI results over all pixels for all ranked SGM variants. Our SGM-Forest uses MC-CNN-fast as matching cost and achieves high accuracy at comparatively low runtime. *Right:* ETH3D results over non-occluded and all pixels for all ranked methods. Our SGM-Forest uses MC-CNN-fast as matching cost and achieves the best accuracy at comparatively low runtime.

KITTI 2015		
Method	**Error**	**Time**
CNNF+SGM	3.60% (#9)	71.0s
SGM-Net	3.66% (#11)	67.0s
MC-CNN-acrt	3.89% (#12)	67.0s
SGM-Forest	**4.38% (#14)**	**6.0s**
MC-CNN-WS	4.97% (#18)	1.4s
SGM_ROB [17]	6.38% (#27)	0.1s
SGM+C+NL	6.84% (#31)	270.0s
SGM+LDOF	6.84% (#32)	86.0s
SGM+SF	6.84% (#33)	2700.0s
CSCT+SGM+MF	8.24% (#35)	6.4ms

ETH3D 2017			
Method	**non-occl.**	**all**	**Time**
SGM-Forest	**5.40%**	**4.96%**	**5.21s**
SGM_ROB [17]	10.08%	10.77%	0.15s
MeshStereo	11.94%	11.52%	159.24s
SPS-Stereo	15.83%	15.04%	1.59s
ELAS	17.99%	16.72%	0.13s

Middlebury. Table 3 reports our results on Middlebury 2014. For the benchmark submission, we use MC-CNN-acrt matching costs and jointly train on Middlebury 2005–06 and the training scenes of Middlebury 2014. Our method ranks competitively among the top ten methods in terms of accuracy but is significantly faster. In addition to our official submission, we also report unofficial results for MC-CNN-fast evaluated on the training scenes[1]. The models for this submission were trained only on the Middlebury 2005–06 scenes. Using MC-CNN-fast, SGM-Forest outperforms SGM by two percentage points on non-occluded pixels. Evaluated on all pixels, SGM-Forest with MC-CNN-fast outperforms baseline SGM with MC-CNN-acrt by two percentage points but SGM-Forest is an order of magnitude faster.

KITTI. Table 4 lists all SGM-based methods evaluated on KITTI. We use MC-CNN-fast for this submission and are ranked right behind the original MC-CNN-acrt method [48], CNNF+SGM [51], and SGM-Net [42]. However, our method

[1] Only one submission per method is allowed on Middlebury 2014.

is an order of magnitude faster even though our scanline optimization and the proposed additional steps are implemented on the CPU while MC-CNN-WS runs on the GPU. Note that CNNF+SGM and SGM-Net report results only on KITTI whereas our method generalizes across domains and datasets.

ETH3D. On this fairly new benchmark with diverse indoor and outdoor images, SGM-Forest is currently ranked 1st with competitive running times (see Table 4). Our submission uses MC-CNN-fast which was surprisingly more accurate than MC-CNN-acrt on ETH3D (also see Table 2). Here, our SGM-Forest submission has almost half the error as the original SGM method [17].

5.4 Qualitative Results

Figure 3 shows qualitative results for Middlebury. Compared to baseline SGM, our SGM-Forest produces less streaking artifacts and performs significantly better in occluded areas. High confidence regions in general correspond to low errors. This is further confirmed by the monontonically decreasing precision-recall curves, which were produced by thresholding on the predicted confidences. For further qualitative results, *e.g.*, comparisons between raw predictions and filtered results, we refer the reader to the supplementary material.

5.5 Limitations and Future Work

Our current SGM and random forest implementation is CPU-based and is not real-time capable since we buffer all scanline cost volumes before fusion. The learned forests in this paper use 128 trees, so our method could be sped up easily by using fewer trees. In our experiments, even a single decision tree improved upon baseline SGM. An implementation of our method on the GPU would be straightforward, where SGM-MLP would probably outcompete SGM-Forest in

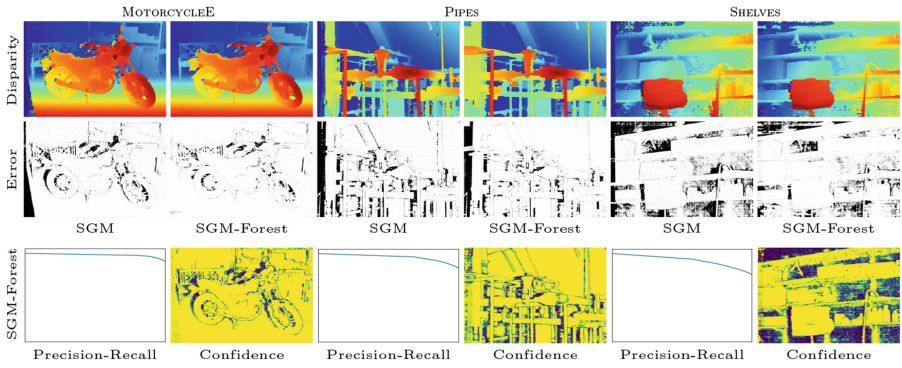

Fig. 3. Qualitative Middlebury results for SGM and SGM-Forest. Absolute error maps clipped to [0px, 8px]. Precision (Y) and Recall (X) in [0, 1]. Confidence maps log-scaled.

efficiency at the cost of a small degradation in accuracy. Real-time implementation on embedded systems [19] requires a one-pass, buffer-less algorithm prohibiting the use of all 8 scanline directions. In Table 1, we demonstrated that our idea also works well for top-down/bottom-up directions only.

6 Conclusion

We proposed a learning-based approach to fuse scanline optimization proposals in SGM, replacing the brittle and heuristic scanline aggregation steps in standard SGM. Our method is efficient and accurate and ranks 1st on the ETH3D benchmark while being competitive on Middlebury and KITTI. We have demonstrated consistent improvements over SGM on three stereo benchmarks. The learning appears to be extremely robust and generalizes well across datasets. Our method can be readily integrated into existing SGM variants and allows for real-time implementation in practical, high-quality stereo systems.

References

1. Semi-global stereo matching with surface orientation priors. In: International Conference on 3D Vision (3DV) (2017)
2. Banz, C., Blume, H., Pirsch, P.: Real-time semi-global matching disparity estimation on the GPU. In: International Conference on Computer Vision (ICCV) Workshops (2011)
3. Bleyer, M., Gelautz, M.: Simple but effective tree structures for dynamic programming-based stereo matching. In: International Conference on Computer Vision Theory and Applications (VISAPP) (2008)
4. Bleyer, M., Rother, C., Kohli, P., Scharstein, D., Sinha, S.: Object stereo—joint stereo matching and object segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
5. Chen, Z., Sun, X., Wang, L., Yu, Y., Huang, C.: A deep visual correspondence embedding model for stereo matching costs. In: International Conference on Computer Vision (ICCV) (2015)
6. Drory, A., Haubold, C., Avidan, S., Hamprecht, F.A.: Semi-global matching: a principled derivation in terms of message passing. In: Jiang, X., Hornegger, J., Koch, R. (eds.) GCPR 2014. LNCS, vol. 8753, pp. 43–53. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11752-2_4
7. Facciolo, G., De Franchis, C., Meinhardt, E.: MGM: a significantly more global matching for stereovision. In: British Machine Vision Conference (BMVC) (2015)
8. Franke, U., Pfeiffer, D., Rabe, C., Knoeppel, C., Enzweiler, M., Stein, F., Herrtwich, R.G.: Making bertha see. In: International Conference on Computer Vision (ICCV) Workshops (2013)
9. Gehrig, S.K., Eberli, F., Meyer, T.: A real-time low-power stereo vision engine using semi-global matching. In: Fritz, M., Schiele, B., Piater, J.H. (eds.) ICVS 2009. LNCS, vol. 5815, pp. 134–143. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04667-4_14
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)

11. Gidaris, S., Komodakis, N.: Detect, replace, refine: Deep structured prediction for pixel wise labeling. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
12. Haeusler, R., Nair, R., Kondermann, D.: Ensemble learning for confidence measures in stereo vision. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
13. Hermann, S., Klette, R.: Iterative semi-global matching for robust driver assistance systems. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) ACCV 2012, Part III. LNCS, vol. 7726, pp. 465–478. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37431-9_36
14. Hermann, S., Klette, R., Destefanis, E.: Inclusion of a second-order prior into semi-global matching. In: Wada, T., Huang, F., Lin, S. (eds.) PSIVT 2009. LNCS, vol. 5414, pp. 633–644. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-92957-4_55
15. Hirschmüller, H.: Accurate and efficient stereo processing by semi-global matching and mutual information. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
16. Hirschmuller, H.: Stereo vision in structured environments by consistent semi-global matching. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2006)
17. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. Trans. Pattern Anal. Mach. Intell. **30**, 328–341 (2008)
18. Hirschmüller, H., Buder, M., Ernst, I.: Memory efficient semi-global matching. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences (2012)
19. Honegger, D., Oleynikova, H., Pollefeys, M.: Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU. In: International Conference on Intelligent Robots and Systems (IROS) (2014)
20. Hu, X., Mordohai, P.: A quantitative evaluation of confidence measures for stereo vision. Trans. Pattern Anal. Mach. Intell. **34**, 2121–2133 (2012)
21. Kendall, A., et al.: End-to-end learning of geometry and context for deep stereo regression. In: International Conference on Computer Vision (ICCV) (2017)
22. Knöbelreiter, P., Reinbacher, C., Shekhovtsov, A., Pock, T.: End-to-end training of hybrid CNN-CRF models for stereo. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
23. Lee, Y., Park, M.G., Hwang, Y., Shin, Y., Kyung, C.M.: Memory-efficient parametric semiglobal matching. Signal Process. Lett. **25**, 194–198 (2018)
24. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for markov random field optimization. Trans. Pattern Anal. Mach. Intell. **32**, 1392–1405 (2010)
25. Luo, W., Schwing, A.G., Urtasun, R.: Efficient deep learning for stereo matching. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
26. Mac Aodha, O., Humayun, A., Pollefeys, M., Brostow, G.J.: Learning a confidence measure for optical flow. Trans. Pattern Anal. Mach. Intell. **35**, 1107–1120 (2013)
27. Mayer, N., et al.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
28. Michael, M., Salmen, J., Stallkamp, J., Schlipsing, M.: Real-time stereo vision: Optimizing semi-global matching. In: Intelligent Vehicles Symposium (2013)
29. Ohta, Y., Kanade, T.: Stereo by intra-and inter-scanline search using dynamic programming. Trans. Pattern Anal. Mach. Intell. **2**, 139–154 (1985)

30. Pang, J., Sun, W., Ren, J., Yang, C., Yan, Q.: Cascade residual learning: A two-stage convolutional neural network for stereo matching. In: International Conference on Computer Vision (ICCV) Workshops (2017)
31. Park, M.G., Yoon, K.J.: Leveraging stereo matching with learning-based confidence measures. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
32. Poggi, M., Mattoccia, S.: Deep stereo fusion: combining multiple disparity hypotheses with deep-learning. In: International Conference on 3D Vision (3DV) (2016)
33. Poggi, M., Mattoccia, S.: Learning a general-purpose confidence measure based on O(1) features and a smarter aggregation strategy for semi global matching. In: International Conference on 3D Vision (3DV) (2016)
34. Rothermel, M., Wenzel, K., Fritsch, D., Haala, N.: SURE: Photogrammetric surface reconstruction from imagery. In: LC3D Workshop (2012)
35. Scharstein, D., et al.: High-resolution stereo datasets with subpixel-accurate ground truth. In: Jiang, X., Hornegger, J., Koch, R. (eds.) GCPR 2014. LNCS, vol. 8753, pp. 31–42. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11752-2_3
36. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2007)
37. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. J. Comput. Vis. **47**, 7–42 (2002)
38. Schmid, K., Tomic, T., Ruess, F., Hirschmüller, H., Suppa, M.: Stereo vision based indoor/outdoor navigation for flying robots. In: International Conference on Intelligent Robots and Systems (IROS) (2013)
39. Schönberger, J.L., Radenović, F., Chum, O., Frahm, J.M.: From Single Image Query to Detailed 3D Reconstruction. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2015)
40. Schönberger, J.L., Zheng, E., Frahm, J.-M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 501–518. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_31
41. Schöps, T., Schönberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
42. Seki, A., Pollefeys, M.: Sgm-nets: Semi-global matching with neural networks. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
43. Spyropoulos, A., Mordohai, P.: Ensemble classifier for combining stereo matching algorithms. In: International Conference on 3D Vision (3DV) (2015)
44. Taniai, T., Matsushita, Y., Sato, Y., Naemura, T.: Continuous 3D label stereo matching using local expansion moves. Trans. Pattern Anal. Mach. Intell. (2017). https://ieeexplore.ieee.org/document/8081755/
45. Taniai, T., Sinha, S.N., Sato, Y.: Fast multi-frame stereo scene flow with motion segmentation. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
46. Van Meerbergen, G., Vergauwen, M., Pollefeys, M., Van Gool, L.: A hierarchical symmetric stereo algorithm using dynamic programming. Int. J. Comput. Vis. **47**, 275–285 (2002)
47. Veksler, O.: Stereo correspondence by dynamic programming on a tree. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
48. Žbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. J. Mach. Learn. Res. **17**, 1–32 (2016)

49. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 756–771. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_49
50. Zach, C., Sormann, M., Karner, K.: Scanline optimization for stereo on graphics hardware. In: 3DPVT (2006)
51. Zhang, F., Wah, B.W.: Fundamental principles on learning new features for effective dense matching. Trans. Image Process. **27**, 822–836 (2018)