2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

LiDAR-SGM: Semi-Global Matching on LiDAR Point Clouds and their Cost-based Fusion into Stereo Matching

Bianca Forkel and Hans-Joachim Wuensche

***Abstract*— Stereo matching can be used to estimate dense but inaccurate depth information for each pixel of a camera image. A LiDAR can provide accurate but sparse depth measurements. The fusion of both can combine their advantages.**

We propose an efficient method for fusing stereo and LiDAR at the cost level of Semi-Global Matching. It significantly improves density and accuracy of the estimated disparities while remaining real-time capable. Based on a LiDAR point cloud projected into the camera image costs are calculated for each possible disparity. These costs are added to the costs from stereo matching.

Our LiDAR-SGM outperforms other real-time capable fusion approaches evaluated on the KITTI Stereo 2015 dataset. In addition to this real data, synthetic datasets are created (and made available) for a detailed analysis of the benefit of stereo LiDAR fusion as well as the evaluation of different sensors.

I. Introduction

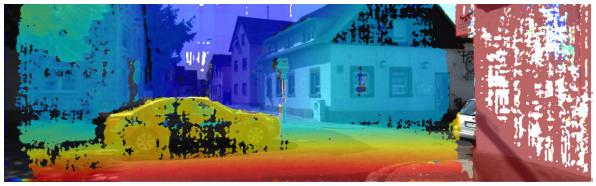
Stereo matching is a way of using cameras to obtain not only color, but also depth information. For this, correspondences are found between the images of two, usually horizontally displaced cameras, and a disparity is calculated. The resulting point cloud offers a high spatial resolution. However, the errors in the estimated depth increase quadratically with the distance to the camera. Furthermore, disparities often cannot be measured for objects with little or periodic texture.

To overcome the last issue, in previous work [1], we proposed adding a third, vertically displaced camera in the same plane. However, this is an unusual setup and often difficult to install. Instead, in automotive research, there is often a LiDAR sensor available. It has the benefit of more accurate depth measurements almost independently of an object's texture. However, the measurements are very sparse and cannot provide depth information for each pixel of a camera image. This is why we are looking into the fusion of LiDAR and stereo to combine the advantages of both sensors.

For stereo, we are using Semi-Global Matching (SGM) [2] on a horizontal camera pair. Analog to [1], we fuse it with LiDAR point clouds not on object or point level but on cost level, using the LiDAR information directly for stereo matching. This way, a more dense and more accurate stereo disparity image is gained in real-time.

Figure 1 shows three main advantages of the fusion: Compared to traditional Stereo-SGM, our new LiDAR-SGM improves the disparity image on objects with little texture and in areas occluded in one of the cameras. Moreover, the disparities are more accurate. Our LiDAR-SGM may also be used independently of stereo to add depth information to a single camera image.

All authors are with the Institute for Autonomous Systems Technology (TAS) of the Universität der Bundeswehr München, Neubiberg, Germany. Contact author email: bianca.forkel@unibw.de



(a) Stereo-SGM ($\alpha = 0$): 76.66 % correct, RMSE 1.00 px

(b) LiDAR-SGM ($\alpha = 0.7$): 93.12 % correct, RMSE 0.58 px

Fig. 1: Example of how the fusion of LiDAR information into stereo can improve the estimated disparities. (**a**) shows the disparities estimated by stereo only, (**b**) the result of our fusion. The scene is part of the KITTI dataset. Note, that the upper third of the image is not captured by the LiDAR sensor and thus not improved.

II. Related Work

Different methods for the fusion of stereo and LiDAR have already been proposed. An early approach was introduced by Badino et al. [3]. Also using SGM, they reduce its disparity search space based on LiDAR measurements. Further, an additional penality for deviations from the LiDAR measurements is added. Unfortunately, quantitative and timing evaluations are missing. Maddern and Newman [4] also use the LiDAR points to reduce the disparity search space. Moreover, they fuse priors from sparse stereo matching and LiDAR using the stereo framework of [5]. However, it does not offer a high accuracy, especially with missing LiDAR information. Courtois and Aouf [6] extend [5] similar to [4], but using a bilateral filter for interpolation. While being more accurate, their approach is not real-time capable.

Processing times are also the main problem of the recently popular Convolutional Neural Networks (CNN) for the fusion of stereo and LiDAR. The first to utilize a CNN for disparity fusion were Park et al. [7]. Based on SGM, a stereo disparity map is calculated and then fused with a LiDAR disparity map using a deep convolutional neural network. However, the whole pipeline takes 10 times the computation time of its baseline SGM. The CNNs of Wang et. al [8], Cheng et al. [9], and Choe et al. [10] all take more than one second for a single scene. Zhang et al. [11] and Mai et al. [12] do not give inference times for their CNNs. Only Meng et al. [13] recently managed to achieve real-time capability by using binarized neural networks for feature extraction, while falling back on SGM for cost aggregation.

979-8-3503-2365-8/23/$31.00 ©2023 IEEE

III. LIDAR-SGM

Without utilizing a resource-demanding and expensive to train neural network, our LiDAR stereo fusion hooks directly into SGM [2] in an extremely simple way.

A. Semi-Global Matching

To give a short summary of [2]: In Semi-Global Matching, a disparity image is estimated by

1) pixelwise cost calculation,
2) (global) smoothness constraints, and
3) the search for the minimum cost.

First, for each pixel $\mathbf{p}$, matching costs $C(\mathbf{p}, d)$ for possible disparities d are calculated. A commonly used measure is the census transform [14] (also see [1]). It is evaluated independently for every pixel. Directly using the resulting costs for disparity estimation is prone to error due to noise or ambiguities. Thus, SGM adds global smoothness constraints by introducing penalties for disparity changes between neighboring pixels. To avoid the need of solving an NP-complete optimization problem when considering the neighborhoods of every pixel, SGM dissects the problem into single 1D paths in the image. For a path $\mathbf{r}$ (see [2]), the costs

$$\begin{aligned} L_{\mathbf{r}}(\mathbf{p}, d) = & C(\mathbf{p}, d) + \min\big(L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d), \\ & L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d-1) + P_1, L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, d+1) + P_1, \\ & \min_i L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, i) + P_2\big) - \min_k L_{\mathbf{r}}(\mathbf{p}-\mathbf{r}, k) \end{aligned} \tag{1}$$

are calculated, where P_1 penalizes neighboring pixels with small disparity changes, while P_2 penalizes disparity jumps. The optimal disparity of a pixel is then obtained by minimizing the costs aggregated along different image paths:

$$\mathbf{D}(\mathbf{p}) = \arg\min_d \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d)\,. \tag{2}$$

To obtain a sub-pixel disparity estimate, a parabola is fitted through the optimal disparities of a pixel and its two neighbors. Finally, a left-right consistency check is performed (see [2]).

B. LiDAR-SGM: LiDAR Matching Cost

For LiDAR-SGM, we want to apply the cost aggregation and smoothing of Semi-Global Matching on LiDAR point clouds. Hence, we need to calculate pixelwise matching costs from the LiDAR data. Therefore, the LiDAR point cloud is projected into an image (taking into account the intrinsic and extrinsic calibration of the camera to the LiDAR). Due to different view points, one pixel may get several associated LiDAR points. To increase the density of the sparse LiDAR points, not only the depth points projected directly into the corresponding pixel are considered, but also that of direct or diagonal neighbors (neighborhood of 9, comparable to a median filter with a window size of 3 px). An example projection can be seen in Fig. 2.

To apply SGM, for each pixel $\mathbf{p}$, a cost for each possible disparity d must be specified. Thus, for every LiDAR point P, a virtual disparity

$$d_P = \frac{fb}{z_P} \tag{3}$$

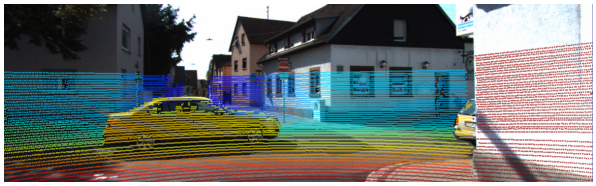


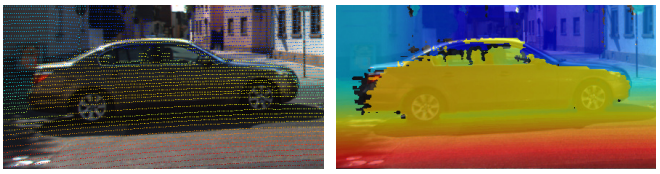
Fig. 2: Disparities calculated from a LiDAR point cloud by projecting it into an image considering a neighborhood of 9 pixels. The scene is part of the KITTI dataset. Note that the point cloud does not cover the upper part of the image.

is calculated based on its depth z_P, focal length f and a baseline b. In case of the later fusion with stereo, f and b should correspond to the values of the respective rectified camera. We then determine the LiDAR matching cost as

$$C_L(\mathbf{p}, d) = \min_P \left(\frac{64}{\sigma^2}\left(d - d_P\right)^2\right), \tag{4}$$

considering all associated LiDAR points. This results in quadratic costs corresponding to a Gaussian probability distribution of the disparity, with mean d_P and standard deviation σ. We set σ to 2 px. Instead of 64, another constant less than 128 could be used. We choose 64 since it is in the middle of the possible value range of the cost.

SGM could be directly applied to this cost. That way, a dense depth image can be interpolated from the sparse LiDAR point cloud, as the example in Fig. 3 shows. The disadvantage is that the depths are discretized in form of disparities. However, although the result is quite good, our main goal is not to interpolate LiDAR data without any additional information, but to fuse the LiDAR information into stereo matching to get the best of both worlds.



(a) Point Cloud **(b)** LiDAR-only-SGM ($\alpha = 1.0$)

Fig. 3: The dense depth image in (**b**) can be interpolated from the sparse LiDAR point cloud in (**a**) using Semi-Global Matching (without additional information).

C. LiDAR-SGM: Cost Fusion

Inspired by [1], where horizontal stereo is fused with vertical stereo, our fusion of LiDAR and stereo is done at cost level. So, we are calculating fused matching cost

$$C(\mathbf{p}, d) = \alpha C_L(\mathbf{p}, d) + (1-\alpha) C_S(\mathbf{p}, d) \tag{5}$$

as a linear combination of LiDAR matching cost C_L and stereo matching cost C_S. The LiDAR impact factor α adjusts the weighting of both sensors. Section IV-B examines the choice of α. While $\alpha = 0$ reflects traditional Stereo-SGM and $\alpha = 1$ LiDAR-only-SGM, fusing both costs can not only help for a higher density but also to resolve ambiguities and increase the accuracy, as the example in Fig. 4 shows.

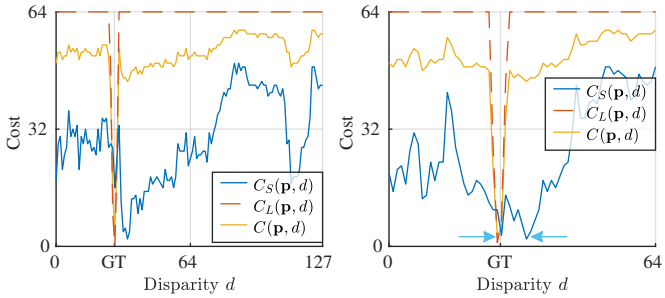


Fig. 4: Exemplary fusion of the stereo matching cost (blue) with our LiDAR cost (red) to the fused matching cost (yellow), with $\alpha = 0.7$. GT marks the ground truth disparity. In (**a**), the disparity is determined more accurately, while in (**b**) the LiDAR matching cost resolves an ambiguity (arrows) in the stereo matching cost.

IV. Experimental Results

To evaluate LiDAR-SGM, we first need to investigate the impact of fusing LiDAR point clouds into stereo matching and determine a suitable LiDAR impact factor α. Then, we want to compare the approach not only with traditional horizontal stereo and other approaches fusing LiDAR into stereo, but also with stereo matching with three cameras.

A. Setup

Qualitative and quantitative results in this paper are based on three different datasets including real-world data and synthetic data. Synthetic data is needed, as a simulation can give accurate ground truth disparity information (GT) for the whole image without using the LiDAR point cloud itself. Further, additional sensors can be added easily. Example images of every dataset can be seen in Table III.

a) Real-World Data: The real-world dataset consists of 142 RGB image pairs from the training data of the KITTI stereo 2015 benchmark [15] and the associated LiDAR point clouds from the raw KITTI vision benchmark suite [16] (for the other images from the stereo evaluation no LiDAR scan is provided). The ground truth disparities provided with the benchmark were created from accumulated LiDAR point clouds, which is why the results of our LiDAR-SGM are biased in this dataset (since we use the LiDAR point cloud as measurement). However, we want to include this dataset in our evaluation, as it is used in the evaluation of previous work ([4], [6]–[9]). Like in the official benchmark, we are using the occluded ground truth. Besides cars, for which CAD models are used to fill the disparity map, the ground truth is very sparse and does not cover the upper part of the image. The baseline of the two rectified cameras is 53.27 cm.

b) Simulated Data: For the synthetic data, two sets of 500 static scenes were rendered using the CARLA urban driving simulator [17]. Two different weather settings were used, 'ClearSunset' and 'HardRainNoon'. Unfortunately, these weather settings only affect the images. The ground truth is calculated from the results of an undisturbed depth camera. For the RGB cameras with a horizontal baseline of 80 cm, post-processing effects like jitter, bloom or lens flair are activated. The settings for the LiDAR sensor are inspired by the Velodyne HDL-64E used in the KITTI dataset (64 channels, 1 300 000 points per second, maximum range 120 m). For the evaluations in Section IV-D and Section IV-D, an additional LiDAR scanner with 128 channels, 2 400 000 points per second and a maximum range of 245 m (similar to a Velodyne Alpha Prime), as well as a third camera 20 cm above the left camera were added. The datasets are available at `https://www.mucar3.de/icra2023-lidar-sgm`.

c) Implementation: Our Stereo- and LiDAR-SGM implementation is an evolution of the Triple-SGM implementation [1], such that the compared approaches share the same stereo matching core. Using the CUDA implementation of libSGM[1], the (fused) matching cost is aggregated along four paths with $P_1 = 10$ and $P_2 = 100$. A median filter with a window size of 5 px is applied to the resulting disparity image.

d) Metrics: Like in [1], the quantitative results are based on two metrics, the rate of pixels (with GT) for which a correct disparity is estimated (correctness), and the root-mean-square error (RMSE) of those correct pixels. Inspired by the KITTI benchmark [15], a pixel is considered to be correct if its disparity error is < 3 px or $< 5\,\%$ of the GT disparity.

B. Stereo-SGM, LiDAR-only-SGM and their Fusion

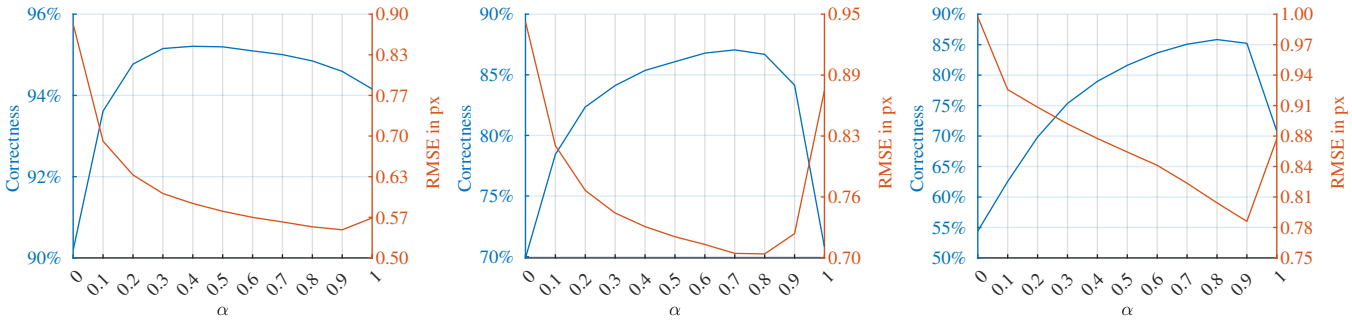
First of all, we want to quantitatively evaluate the result of Semi-Global Matching on LiDAR point clouds and how the cost fusion with stereo matching affects the disparity estimation result depending on the cost weighting. Therefore, Fig. 5 shows correctness and RMSE of all three datasets in dependence of the LiDAR impact factor α, where $\alpha = 0$ corresponds to traditional camera-only horizontal stereo matching, and $\alpha = 1$ means LiDAR-only-SGM.

Looking at the stereo-only results, one can see that they are disturbed by bad weather. Further, stereo performs worse on the CARLA datasets than on KITTI. On the one hand, this is because the synthetic CARLA images have less texture. On the other hand, unlike KITTI, CARLA provides ground truth disparities for the whole image, including the sky and also for very small objects, which are both difficult to estimate.

This is also why in CARLA ClearSunset, LiDAR-only-SGM does not have a much higher correctness rate than Stereo-SGM, since the LiDAR point cloud does not contain any information about the upper parts of the image. In CARLA HardRainNoon and on KITTI, however, LiDAR-only-SGM estimates significantly more disparities correctly than stereo. Although more pixels are considered to be correct, also their RMSE is lower. However, KITTI's ground truth is based on the LiDAR data and in CARLA the LiDAR point cloud is not affected by the bad weather. All in all, LiDAR(-only)-SGM is a good way to obtain depth information for camera images and may outperform stereo matching.

Yet the clearly best correctness (and for CARLA also RMSEs) can be achieved by fusing stereo and LiDAR costs for SGM. Which weighting of the costs is best differs between the datasets. While KITTI has a peak in the correctness at $\alpha = 0.4$, the peaks for CARLA are at $\alpha = 0.7$ and $\alpha = 0.8$.

[1] `https://github.com/fixstars/libSGM`



(a) KITTI **(b)** CARLA ClearSunset **(c)** CARLA HardRainNoon

Fig. 5: Median rate of correctly estimated disparity pixels (blue) and their RMSE (red) in dependence on the weighting α of the LiDAR matching costs (and $(1-\alpha)$ of the stereo matching costs) over all images for the three datasets. Note the different axes for correctness (left) and RMSE (right).

TABLE I: Median rates of correctly estimated disparity pixels and their RMSE for traditional stereo matching ($\alpha = 0$), LiDAR-only-SGM ($\alpha = 1$) and our fused LiDAR-SGM ($\alpha = 0.7$) for the three datasets KITTI, CARLA ClearSunset and CARLA HardRainNoon.

	KITTI		ClearSunset		HardRainNoon	
	Correct	RMSE	Correct	RMSE	Correct	RMSE
$\alpha = 0$	90.19 %	0.88 px	69.90 %	0.94 px	54.40 %	1.00 px
$\alpha = 1$	94.16 %	0.57 px	70.79 %	0.87 px	70.83 %	0.87 px
$\alpha = 0.7$	95.00 %	0.56 px	87.05 %	0.70 px	85.07 %	0.83 px

We decide to use a LiDAR impact factor of $\alpha = 0.7$ for LiDAR-SGM from now on. For better reading, the values for $\alpha = 0.7$ are given again in Table I. The median rate of wrong pixels is reduced by up to 67 % (CARLA HardRainNoon), while the RMSE can be lowered by at least 17 % (CARLA HardRainNoon) up to 36 % for KITTI. Qualitative results can be seen in Table III.

C. Choice of the LiDAR Sensor

While the KITTI dataset [16] is recorded using a LiDAR scanner with 64 laser diodes, modern LiDAR scanners offer 128 laser diodes and thus nearly twice as many points (and also a higher range). The effects of using such a LiDAR for LiDAR-SGM compared to the sensors with 64 diodes is examined in Fig. 6. It shows the distributions of correctness and RMSE over all images for Stereo-SGM, LiDAR-SGM using a LiDAR with 64 laser diodes, and LiDAR-SGM with 128 diodes. Figure 6 shows the use of the better sensor to further increase the rate of correct pixels while also reducing the RMSE. Since the standard is still 64 beams, we nevertheless used it for all other evaluations.

D. Using a Third Camera with Triple-SGM

In [1], where our idea of cost fusion originates from, a third vertically displaced camera is added to the traditional setup with two horizontally displaced cameras. This not only improves recognition of horizontal structures, but also the estimated disparities significantly.

In order to compare the benefit of a third camera to that of a LiDAR or even both, Fig. 6 also shows the distribution of correctness and RMSE for Stereo-SGM, LiDAR-SGM, Triple-SGM [1] and Triple-SGM fused with LiDAR costs. In both datasets, Triple-SGM performs worse than LiDAR-SGM in terms of correctness and also has a significantly higher RMSE. This is since the LiDAR works according to a different measurement principle than stereo matching, and therefore can compensate well for the problems of stereo, such as low or periodic texture. It further offers a higher accuracy. Thus, the use of an additional LiDAR sensor is more beneficial than that of a third, vertically displaced camera.

The additional fusion of the LiDAR cost into Triple-SGM can slightly improve the results of Triple-SGM in bad weather conditions. However, the results of Stereo-LiDAR-SGM are still better than those of Triple-LiDAR-SGM, as fusing three cost tensors reduces the individual cost gradients too much. Qualitative results can be seen in Table III.

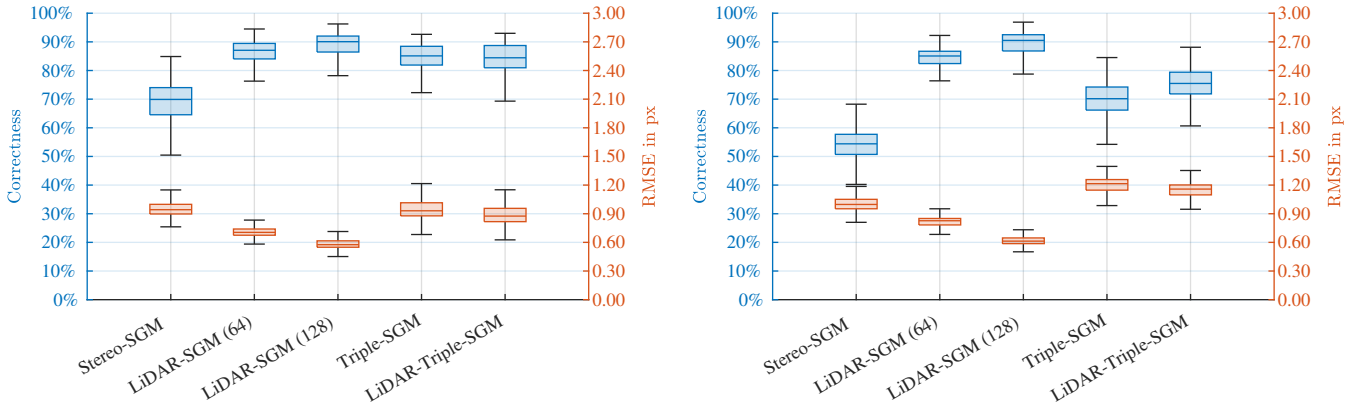
E. Other Approaches

The use of the KITTI dataset allows for a comparison of our approach with the related work of [4], [6]–[8] and [9] that use the same 142 scenes of KITTI stereo 2015 [15] and the corresponding LiDAR scans from the raw dataset [16]. [13] unfortunately creates the LiDAR data differently.

We used the evaluation script delivered with the KITTI-devkit to fill Table II with separate error rates for background pixels, foreground pixels, and total pixels with a ground truth. In the background, our LiDAR-SGM outperforms not only Stereo-SGM, but also the related work of [4] and [6]. In the foreground, fusing the LiDAR information into the stereo matching worsens the result, as the different view points have a stronger effect here, and small objects may not be estimated accurately. Nevertheless, the total error rate of our fused LiDAR-SGM undercuts that of the approaches of Maddern and Newman [4] and Park et al. [7]. It is slightly worse than that of Courtois et al. [6]. However, the approach of [6], as well as the even better CNN-based fusions suggested in [8] and [9], are with processing times of 680 ms, 1 s, and 2 s far from real-time capable.

F. Timing

On the full resolution (1242 px × 375 px) KITTI images, our implementation of LiDAR-SGM takes an average time of 24 ms per scene, compared to 21 ms for the baseline Stereo-SGM. Fusing the LiDAR into SGM takes only 2.4 ms. Thus,



(a) CARLA ClearSunset **(b)** CARLA HardRainNoon

Fig. 6: Distributions of correctness and RMSE over all images of the two CARLA datasets, comparing Stereo-SGM ($\alpha = 0$), LiDAR-SGM using a LiDAR with 64 beams ($\alpha = 0.7$), LiDAR-SGM using a LiDAR with 128 beams ($\alpha = 0.7$), Triple-SGM [1] ($\alpha = 0$), and Triple-SGM fused with LiDAR costs ($\alpha = 0.7$, 64 beams).

TABLE II: Comparison of our LiDAR-SGM ($\alpha = 0.7$) and our baseline Stereo-SGM ($\alpha = 0$) with related work using the KITTI stereo 2015 benchmark. The numbers are taken from the respective papers. $Error_{bg}$, $Error_{fg}$, and $Error_{tot}$ give the rate of wrong pixels in the background, in the foreground, and in total in the non-interpolated disparity images. The density gives the number of pixels (with a GT), for which a disparity is estimated at all.

Method	Error_{bg}	Error_{fg}	$\text{Error}_{\text{tot}}$	Density
Stereo-SGM	3.64 %	9.78 %	4.54 %	92.23 %
LiDAR-SGM	2.06 %	14.47 %	3.87 %	97.46 %
Combined+Pyr [4]	4.55 %	13.18 %	5.91 %	99.62 %
Tri_BF [6]	2.62 %	9.01 %	3.60 %	98.59 %
Park et al. [7]	-	-	4.84 %	99.8 %
Wang et al. [8]	-	-	3.35 %	-
Cheng et al. [9]	-	-	1.98 %	100.00 %

LiDAR-SGM is even faster than the fusion of a third camera in [1], as Triple-SGM takes 30 ms on CARLA images of the same size. For all these results, a single core of an Intel© Core™ i7-8700K CPU running at 3.70 GHz and an Nvidia GeForce GTX 1050Ti GPU were used.

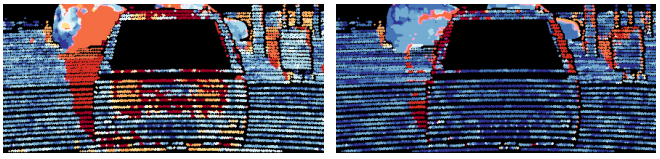
For even faster processing, the input images may be scaled down. At an image scale of 0.5, LiDAR-SGM takes only 7 ms, while correctness and RMSE are only very slightly worsened to 94.35 % and 0.58 px (still significantly better than Stereo-SGM on the full resolution).

G. Qualitative Results

While all previous evaluations gave numbers to demonstrate the performance of LiDAR-SGM, we finally give qualitative examples to show how and where the fusion of LiDAR information into stereo matching is beneficial.

One effect can be seen in Fig. 7: Due to the horizontal displacement of the second camera, there are hidden areas behind/next to objects where the second camera cannot see and no disparity can be calculated (see red parts in Fig. 7a). The LiDAR can provide depth information for these areas, and the disparity images are filled (see Fig. 7b).

Further effects can be seen in Table III, which shows one example scene from each dataset. The left image border and the bottom left corner are not visible in the right camera. Thus, Stereo-SGM cannot estimate a disparity there, while LiDAR-SGM can (visible in all datasets). Further, as expected, difficult texture like dark areas (KITTI), bright areas due to reflections from the sun (CARLA ClearSunset), a wet ground (CARLA HardRainNoon), or polished surfaces (bus in CARLA ClearSunset) cause difficulties in traditional stereo and benefit from additional LiDAR information. However, small structures like street lights or traffic light posts in higher distances are slightly smoothed away.



(a) Stereo-SGM ($\alpha = 0$) **(b)** LiDAR-SGM ($\alpha = 0.7$)

Fig. 7: Visualization of the disparity errors for Stereo-SGM and LiDAR-SGM in the same scene of the KITTI dataset. Dark blue corresponds to a low error, dark red to a high error.

V. Conclusion

In this paper, we show a real-time capable fusion of LiDAR point clouds into stereo processing based on Semi-Global Matching. It improves the rate of pixels with a correctly estimated disparity as well as their RMSE compared to SGM based on stereo-only or LiDAR-only costs. It also outperforms related, real-time capable work evaluated on the training data of the KITTI Stereo 2015 benchmark. We further show, that fusing a LiDAR is more beneficial than adding a third, vertically displaced camera. In future work, the weighting of stereo and LiDAR could be adjusted depending on the range.

Acknowledgment

This research paper is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr [project MORE]. dtec.bw is funded by the European Union – NextGenerationEU.

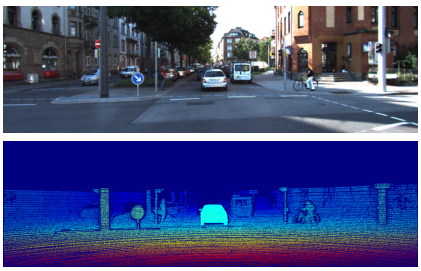
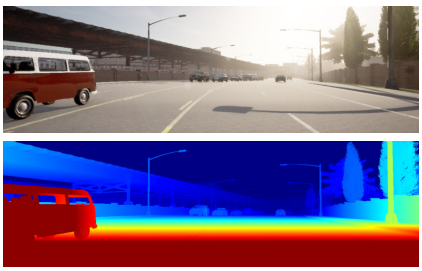
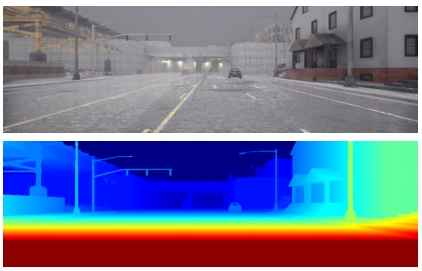
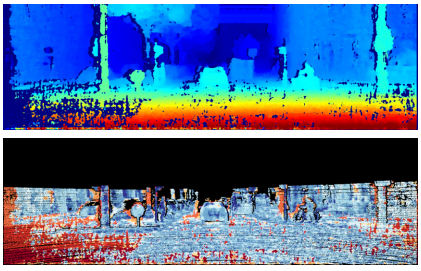
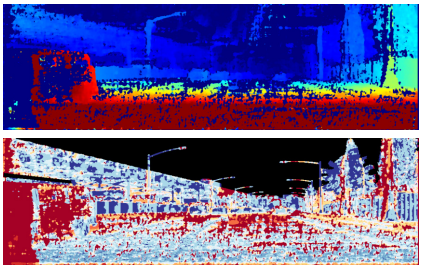
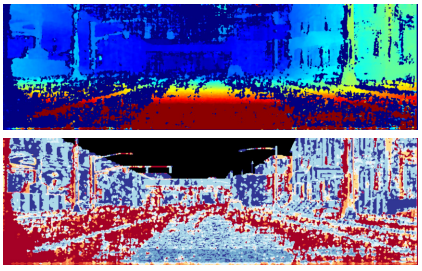
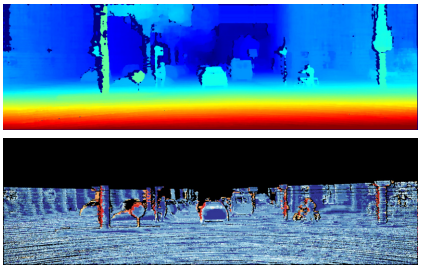
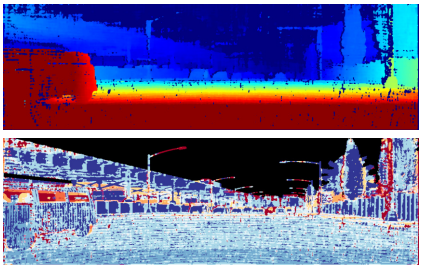
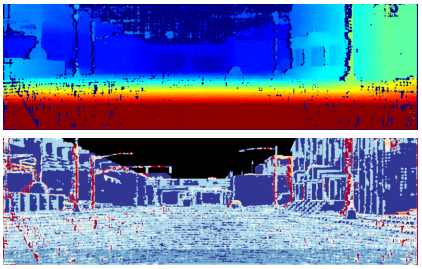
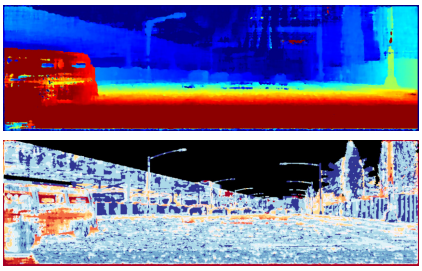
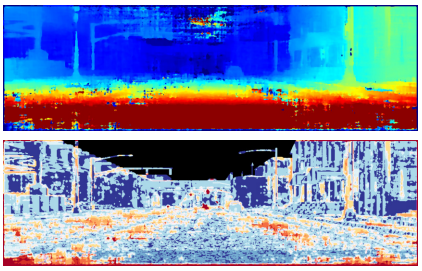
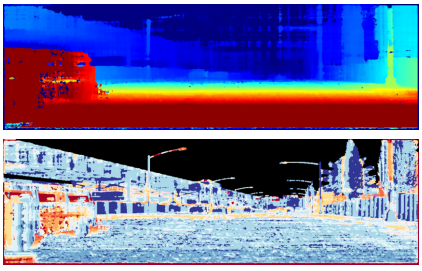
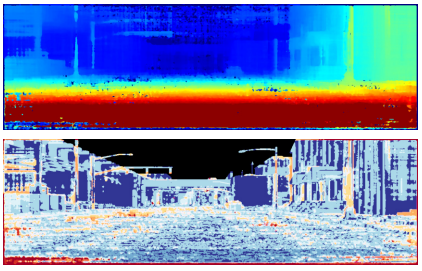
KITTI	CARLA ClearSunset	CARLA HardRainNoon
Input and ground truth		
		
Stereo-SGM ($\alpha = 0$)		
		
(78.45 % correct, RMSE 0.97 px)	(64.31 % correct, RMSE 1.12 px)	(60.87 % correct, RMSE 0.89 px)
LiDAR-SGM ($\alpha = 0.7$)		
		
(97.54 % correct, RMSE 0.53 px)	(86.85 % correct, RMSE 0.78 px)	(90.16 % correct, RMSE 0.69 px)
	Triple-SGM ($\alpha = 0$)	
		
	(78.66 % correct, RMSE 1.09 px)	(73.45 % correct, RMSE 1.04 px)
	LiDAR-Triple-SGM ($\alpha = 0.7$)	
		
	(79.77 % correct, RMSE 1.00 px)	(79.31 % correct, RMSE 0.96 px)

TABLE III: Exemplary qualitative results from the three datasets KITTI, CARLA ClearSunset, and CARLA HardRainNoon (columns). The first row shows the left input image and the corresponding ground truth disparities. Rows 2 and 3 show the disparities (top) along with their error to the GT (bottom) for traditional horizontal SGM (our baseline) and LiDAR-SGM (ours) with $\alpha = 0.7$. For the synthetic CARLA datasets, rows 4 and 5 also show the disparities (and their error) resulting from Triple-SGM [1] and Triple-SGM fused with our LiDAR-SGM ($\alpha = 0.7$). The disparity images are color-coded from dark blue for a disparity of 0 to dark red for a disparity of 63 or higher. The error images are color-coded from dark blue for a low error to dark red for a high error. The subfigure captions give the corresponding rate of pixels with a correct disparity (error < 3 px or < 5 %) and the RMSE of those pixels.

REFERENCES

[1] J. Kallwies, T. Engler, B. Forkel, and H.-J. Wuensche, "Triple-SGM: Stereo Processing using Semi-Global Matching with Cost Fusion," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 192–200.

[2] H. Hirschmüller, "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 807–814.

[3] H. Badino, D. Huber, and T. Kanade, "Integrating LIDAR into Stereo for Fast and Improved Disparity Computation," in *Proceedings of International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, 2011, pp. 405–412.

[4] W. Maddern and P. Newman, "Real-time probabilistic fusion of sparse 3D LIDAR and dense stereo," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2181–2188.

[5] A. Geiger, M. Roser, and R. Urtasun, "Efficient Large-Scale Stereo Matching," in *Proceedings of Asian Conference on Computer Vision (ACCV)*, 2010, pp. 25–38.

[6] H. Courtois and N. Aouf, "Fusion of Stereo and Lidar Data for Dense Depth Map Computation," in *Proceedings of Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, 2017, pp. 186–191.

[7] K. Park, S. Kim, and K. Sohn, "High-Precision Depth Estimation with the 3D LiDAR and Stereo Fusion," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2156–2163.

[8] T.-H. Wang, H.-N. Hu, C. H. Lin, Y.-H. Tsai, W.-C. Chiu, and M. Sun, "3D LiDAR and Stereo Fusion using Stereo Matching Network with Conditional Cost Volume Normalization," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5895–5902.

[9] X. Cheng, Y. Zhong, Y. Dai, P. Ji, and H. Li, "Noise-Aware Unsupervised Deep Lidar-Stereo Fusion," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 6332–6341.

[10] J. Choe, K. Joo, T. Imtiaz, and I. S. Kweon, "Volumetric Propagation Network: Stereo-LiDAR Fusion for Long-Range Depth Estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4672–4679, 2021.

[11] J. Zhang, M. S. Ramanagopal, R. Vasudevan, and M. Johnson-Roberson, "LiStereo: Generate Dense Depth Maps from LIDAR and Stereo Imagery," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7829–7836.

[12] N.-A.-M. Mai, P. Duthon, L. Khoudour, A. Crouzil, and S. A. Velastin, "Sparse LiDAR and Stereo Fusion (SLS-Fusion) for Depth Estimation and 3D Object Detection," in *Proceedings of International Conference of Pattern Recognition Systems (ICPRS)*, 2021, pp. 150–156.

[13] H. Meng, C. Zhong, J. Gu, and G. Chen, "A GPU-accelerated Deep Stereo-LiDAR Fusion for Real-time High-precision Dense Depth Sensing," in *Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 523–528.

[14] R. Zabih and J. Woodfill, "Non-parametric Local Transforms for Computing Visual Correspondence," in *Proceedings of European Conference on Computer Vision (ECCV)*, 1994, pp. 151–158.

[15] M. Menze and A. Geiger, "Object Scene Flow for Autonomous Vehicles," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3061–3070.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[17] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An Open Urban Driving Simulator," in *Proceedings of the Conference on Robot Learning (CoRL)*, 2017, pp. 1–16.