

# LiDAR-Flow: Dense Scene Flow Estimation from Sparse LiDAR and Stereo Images

Ramy Batrawy<sup>1</sup>, René Schuster<sup>1</sup>, Oliver Wasenmüller<sup>1</sup>, Qing Rao<sup>2</sup>, Didier Stricker<sup>1</sup>

**Abstract**— We propose a new approach called LiDAR-Flow to robustly estimate a dense scene flow by fusing a sparse LiDAR with stereo images. We take the advantage of the high accuracy of LiDAR to resolve the lack of information in some regions of stereo images due to textureless objects, shadows, ill-conditioned light environment and many more. Additionally, this fusion can overcome the difficulty of matching unstructured 3D points between LiDAR-only scans. Our LiDAR-Flow approach consists of three main steps; each of them exploits LiDAR measurements. First, we build strong seeds from LiDAR to enhance the robustness of matches between stereo images. The imagery part seeks the motion matches and increases the density of scene flow estimation. Then, a consistency check employs LiDAR seeds to remove the possible mismatches. Finally, LiDAR measurements constraint the edge-preserving interpolation method to fill the remaining gaps. In our evaluation we investigate the individual processing steps of our LiDAR-Flow approach and demonstrate the superior performance compared to image-only approach.

## I. INTRODUCTION

Robust perception is an essential task for reliable autonomous driving platforms. Achieving this goal requires more awareness of the dynamic changes of the environment. For this purpose, dense scene flow estimation became an active research area. It seeks to compute the 3D geometry as well as the 3D motion field and serves as comprehensive representation of a dynamic environment. In the past, this was often approximated by optical flow in 2D space.

Scene flow is usually computed based on dense pixel matches in stereo images. Dense pixel matching is well established [1], [2], [3] and achieves reasonable quality in many scenarios. However, in image regions with textureless objects, shadows, ill-conditioned light environment, etc., image-based matching is extremely challenging. However, exactly these regions are of high importance for autonomous vehicles and inaccuracies can anticipate accidents scenarios.

Unlike image-based technologies, LiDAR sensors are much less sensitive to the aforementioned environmental conditions. Thus, LiDAR is a core technology for mapping 3D surroundings of autonomous vehicles. However, dense LiDAR sensors are expensive and matching unstructured 3D point clouds is challenging [4]. Our approach fuses sparse LiDAR values and stereo images (see Fig. 1) in order to overcome the deficiencies of each technology which leads to increase in the robustness of scene flow estimation. To the best of our knowledge, LiDAR-Flow is the first attempt for this fusion in the context of scene flow.

<sup>1</sup>DFKI – German Research Center for Artificial Intelligence, Germany: [firstname.lastname@dfki.de](mailto:firstname.lastname@dfki.de)

<sup>2</sup>BMW Group, Germany: [firstname.lastname@bmw.de](mailto:firstname.lastname@bmw.de)

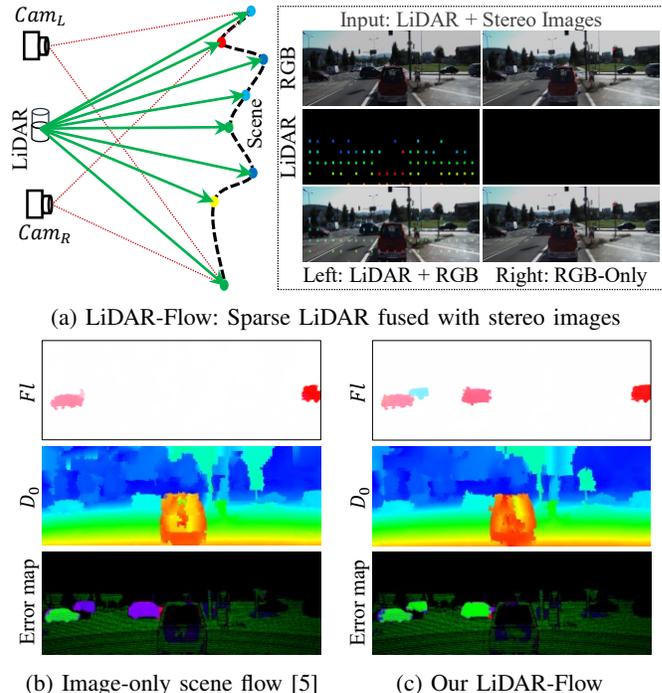


Fig. 1: We introduce LiDAR-Flow which fuses a sparse LiDAR and stereo images for dense scene flow. Compared to image-only approach [5], we achieve high robustness in challenging image regions with our fusion. The green pixels of the error map represent the inliers of scene flow compared to the ground truth.

Our LiDAR-Flow utilizes LiDAR measurements as anchor points to support each processing step across the framework. We utilize a sparse-to-dense approach for our LiDAR-Flow, which is established for image-only scene flow already [5], [6], [7], [8]. We initialize disparity with the LiDAR measurements, estimate flow based on image matching and iteratively propagate these measurements with random search into the neighborhood. During propagation process, each scene flow pixel will be optimized by minimizing a matching cost function. Mismatches will be removed through consistency check with the help of LiDAR measurements afterwards. The sparse set of matches is robustly interpolated with the LiDAR-support. Fig. 2 illustrates an overview of our LiDAR-Flow approach.

In a summary, our contributions are the following:

- We propose LiDAR-Flow – the fusion of sparse LiDAR and stereo images for scene flow estimation.

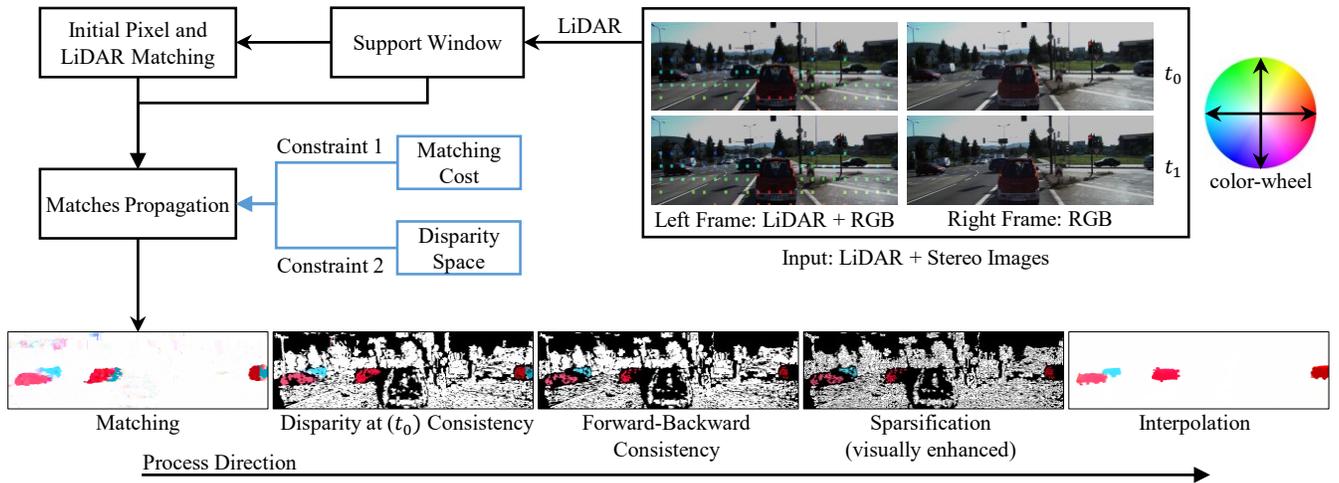


Fig. 2: Overview of our LiDAR-Flow. We show the optical flow ( $FI$ ) estimation after each step. The color-wheel encodes the motion direction and the black color maps the removed outliers after consistency check and sparsification.

- Using LiDAR measurements, we increase the constancy of pixel-based matches for untextured regions.
- We propose an edge-preserving interpolation constrained by LiDAR.
- We show the impact of LiDAR-Flow in each step and compared to the state-of-the-art image-based scene flow on KITTI.

## II. RELATED WORK

The initial approaches for dense scene flow were based on a stereo camera setup. Vedula et al. [9] combine optical flow estimation with the first-order approximations of depth maps as variational optimization framework. Wedel et al. [10] use stereo images for scene flow estimation and decouple the motion from geometry for real time use. In contrary, other methods [11], [12] couple the motion and geometry consistency to improve the scene flow accuracy under stereo use. In the same context, [2], [13] add the cross constancy and an assumption of gradient constancy [13], [14] increase the robustness of matches against illumination change. Further contribution of [15] assumes local rigid regularization instead of variational regularization in using stereo images. In that way, a method of Menze et al. [1] segments the object scene into a set of rigid objects and uses CRF to estimate the association between them. That method does not consider the non-rigidly moving objects like pedestrians or bicyclists. To handle non-rigidly objects, Schuster et al. [8] builds the concept of sparse matching and uses an edge-preserving interpolation method [16] to spread the sparse matches into the entire image. The reliability of generating these matches against light change, perspective deformations and occlusions is increased more by considering multiple frames of stereo [5] and using a robust interpolation method [6]. The existing stereo-based methods achieve impressive results for seeking correspondences across the images. However, they are not reliable with matching textureless objects or challenging illumination.

Alternatively, some methods use active sensors for scene flow estimation. All approaches relying on RGB-D cameras – such as Kinect [17] – provide commendable results [18], [19], [20] but perform poorly in outdoor scenarios with large distances. Thus, LiDAR approaches were used mainly for our use-case. Dewan et al. [21] associate correspondences across LiDAR scans on point cloud space. They formulate the scene flow estimation as an energy minimization problem based on matching signature of histograms SHOT feature descriptors [22]. However, such descriptors can be subjected to errors with sensor noise and they do not overcome the sparse nature of LiDAR data. Following the use of LiDAR, learning-based solutions exist to predict matches from point cloud. Some algorithms train neural network models from unstructured LiDAR point clouds [23], [24]. However, training from that domain is a challenging task. Occupancy grids are used to structure LiDAR data [25], [26] for building network models for feature learning on 3D space. Vaquero et al. [27] estimate dense optical flow for automotive applications from LiDAR scans only, but do not estimate scene flow. They build dense ground truth of the optical flow from the imagery part and they facilitate structuring LiDAR measurements by an alignment on the image domain. Overall, LiDAR-based solutions result in competitive accuracy compared to stereo and RGB-D approaches. However, they are not generating dense estimation of scene flow such as image-based solutions (e.g. stereo systems).

The fusion of a sparse LiDAR and the imagery part has not yet been investigated in the context of scene flow. However, this fusion was applied to other related components. Ma et al. [28] propose a deep regression network for dense depth estimation using one single camera. They use RGB-D raw data to train their model due to the lack of large-space data set with dense LiDAR. The fusion of LiDAR and stereo is suggested to speed up the disparity map computation for outdoor scenes [29], [30]. They interpolate LiDAR measurements on the image domain

and utilize them as a prior information for computing the space of stereo disparity map. Park et al. [31] utilize deep convolutional neural network (CNN) architecture to fuse sparse LiDAR into dense stereo. Our work is the first among the aforementioned fusion methods to fuse a sparse LiDAR and stereo images for computing dense scene flow.

### III. THE PROPOSED APPROACH

For scene flow estimation, we require consecutive scans at times  $t_0$  and  $t_1$  from synchronized LiDAR and stereo frames ( $D_l^0, D_l^1, I_l^0, I_l^1, I_r^0, I_r^1$ ), where  $D$  denotes sparse LiDAR measurements and ( $I_l, I_r$ ) are the left and right of the stereo images (see Fig. 2). We assume the LiDAR measurements and left images are aligned (i.e. all sensors are calibrated). In other words,  $D_l^0$  and  $D_l^1$  are aligned to the images of  $I_l^0$  and  $I_l^1$  respectively. The stereo images are rectified and the baseline  $B$  is known.

Scene flow is represented by the optical flow components  $(u, v)^T$  and the disparity values  $(d_0, d_1)^T$  at a given pixel  $p$  on image  $I_l^0$  (i.e. the reference frame).

$$SF(p) = (u, v, d_0, d_1)^T \quad (1)$$

We are denoting the image space information as well as LiDAR frames with capital letters and the pixel parameters with small letters.

#### A. LiDAR-based Support Window

A prior smoothness assumption is made to improve the matching algorithm. A constant LiDAR measurement is used within a support window to facilitate the initialization and the propagation. In detail, within one support window,  $d_0$  is initialized on the coarsest level with the value of the closest LiDAR measurement of  $D_l^0$  to support the propagation of robust LiDAR measurements into regions with less reliable image matching cost (i.e. minimum matching cost does not necessarily meet optimal scene flow in untextured regions). Even with accurate optical flow components  $(u, v)^T$ , it is unlikely to match a measurement of  $D_l^1$  because of the sparsity. This problem can be solved using the proposed support window also. Increasing the size of the support window has a big advantages in occluded areas (i.e. not visible in the right view) as long as the windows do not overlap. However, the support window can lose its advantages if a strong depth discontinuity occurs within the support window.

To compensate this drawback, a consistency check for the geometry within the support window is applied. The reliable LiDAR measurements of  $D_l^0$  are not subjected to the consistency check.

#### B. LiDAR-Supported Matching

We follow the coarse-to-fine approach of [7] for seeking the matches across the images. Starting with the coarsest level of the reference view,  $d_0$  for each pixel will be initialized with the nearest LiDAR measurement. The proposed support window limits the search area. The other scene flow components are initialized with an image-based method. This

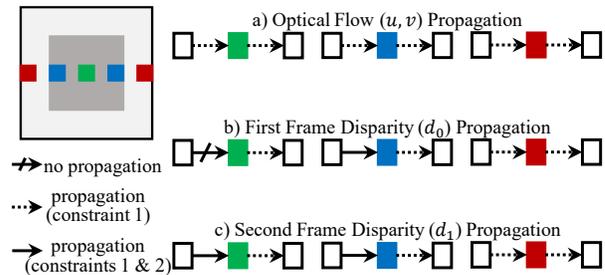


Fig. 3: Our constrained propagation: Pixel with LiDAR measurement (green), pixels within LiDAR-based support window (blue) and image-based matches (red). Propagation into pixels should minimize matching cost (constraint 1) and yield geometry matches within the disparity space (constraint 2) defined by LiDAR measurement. Propagation of  $(u, v)$  into all pixels as well as  $d_0$  and  $d_1$  into red pixels is applied only by constraint 1. Propagating  $d_1$  into green and blue pixels as well as  $d_0$  into blue pixels follows two constraints 1 & 2. Propagation of  $d_0$  into green pixel is not admitted.

is also used for  $d_0$  if no LiDAR measurement is met. On each pyramid level, the initial matches will be propagated into neighboring pixels with random search for a fixed number of iterations. The results of the finest level yields our final matches.

The initialization and the propagation algorithm follow [8] but with further constraints to account for the additional sparse LiDAR measurements.

Since we are considering LiDAR of  $D_l^0$  as exact, propagation of  $d_0$  into LiDAR seeds will not be allowed. Propagation from LiDAR seeds into the neighbor pixels within the LiDAR support window will be possible under a constraint (called constraint 2). This constraint reduces the possibilities of the disparity values coming from propagation or random search. Fig. 3 illustrates the initialization and the propagation constraints in our algorithm. Optimal scene flow matching is obtained by minimizing the matching cost (called constraint 1) for each pixel. The cost function is the sum of Euclidean norms between SIFT Flow features [32] over a patch window. If pixel  $p_A$  in image  $I_A$  matches  $p_B$  in image  $I_B$ , then matching cost yields:

$$E(I_A, I_B, p_A, p_B) = \sum_{\Omega} \|\phi(I_A(p_A)) - \phi(I_B(p_B))\| \quad (2)$$

Where  $\Omega$  is  $7 \times 7$  patch window and  $\phi(I(p))$  is the SIFT Flow vector of pixel  $p$  in image  $I$ . Depending on whether a LiDAR measurement in  $D_l^0$  or  $D_l^1$  is hit during matching, four different cases for the matching are considered. The first case is purely image-based, i.e. no LiDAR measurement:

$$C_I = E_{I_l^0, I_l^1} + E_{I_l^0, I_r^0} + E_{I_l^0, I_r^1} \quad (3)$$

where

$$E_{I_l^0, I_l^1} = E(I_l^0, I_l^1, p, p + (u, v)^T), \quad (4)$$

$$E_{I_l^0, I_r^0} = E(I_l^0, I_r^0, p, p - (d_0, 0)^T), \quad (5)$$

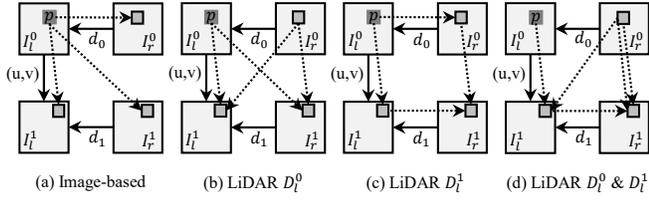


Fig. 4: Matching cost computation: The scheme of pure image-based constancy (a), the constancy if a LiDAR measurement in  $D_r^0$  is met (b), if a LiDAR measurement in  $D_r^1$  is met (c) and if both LiDAR  $D_r^0$  &  $D_r^1$  measurements are met (d).

$$E_{I_r^0, I_r^1} = E(I_r^0, I_r^1, p, p + (u - d_1, v)^T). \quad (6)$$

The second case is applied when a LiDAR measurement is available in  $D_r^0$ . In that case, the term  $E_{I_r^0, I_r^0}$  in Eq. 3 will be replaced with two other constancy terms:

$$C_{D_r^0} = E_{I_r^0, I_r^1} + E_{I_r^0, I_r^1} + E_{I_r^0, I_r^1} + E_{I_r^0, I_r^1} \quad (7)$$

where

$$E_{I_r^0, I_r^1} = E(I_r^0, I_r^1, p - (d_0, 0)^T, p + (u - d_1, v)^T), \quad (8)$$

$$E_{I_r^0, I_r^1} = E(I_r^0, I_r^1, p - (d_0, 0)^T, p + (u, v)^T). \quad (9)$$

The third case covers the availability of a LiDAR measurement in  $D_r^1$ . Two other constancy terms replace the term of  $E_{I_r^0, I_r^1}$  in Eq. 3:

$$C_{D_r^1} = E_{I_r^0, I_r^1} + E_{I_r^0, I_r^0} + E_{I_r^0, I_r^1} + E_{I_r^1, I_r^1} \quad (10)$$

$$E_{I_r^1, I_r^1} = E(I_r^1, I_r^1, p + (u, v)^T, p + (u - d_1, v)^T) \quad (11)$$

The last case involves LiDAR measurements in both frames  $D_r^0$  and  $D_r^1$ :

$$C_{D_r^0 D_r^1} = E_{I_r^0, I_r^1} + E_{I_r^0, I_r^1} + 2 \cdot E_{I_r^0, I_r^1} + E_{I_r^1, I_r^1} \quad (12)$$

The optical flow constancy of the right camera (Eq. 8) is used twice in this case, to avoid an improper deviation of the optical flow to match sparse LiDAR measurements in  $D_r^1$ . The proposed schemes of the matching cost are visualized in Fig. 4 and the overall process of matching is shown in Fig. 2.

### C. LiDAR-Supported Consistency Check

Since we have sparse LiDAR measurements not covering the entire image, a consistency check is required to remove mismatches caused by image matching. Removing outliers is performed through two stages, each followed by a clustering algorithm. During all filtering stages sparse LiDAR measurements will never be filtered. Moreover, LiDAR measurements are employed as reference values during the clustering algorithm.

The first stage involves checking the consistency of the disparity at  $t_0$  (see Fig. 2). To this end, the  $d_0$  values of

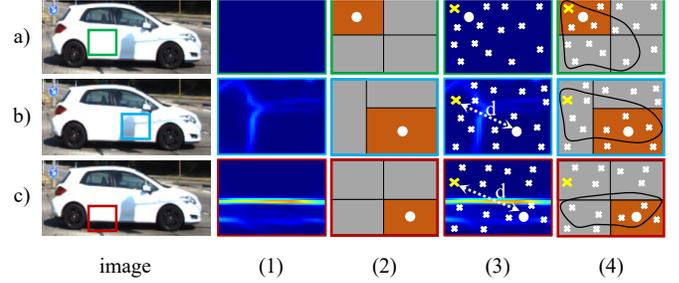


Fig. 5: Constraints of the piecewise model computation for a superpixel (orange); whose center (solid white) in different regions (green, blue and red) in an image. Edge map (1), associated superpixels (e.g. 4 segments) to each region (2), associated matches to each region (crosses) (3) and consistent matches within the sketch (4). LiDAR match (yellow cross) initializes a piecewise model and assists in selecting more support matches for a final model (a), the same if the distance  $d$  from LiDAR match (yellow cross) to superpixel center is less than a defined threshold (b), otherwise a closest pure image-based match (closest white cross) is the best choice for initialization and estimating consistent matches (c).

the scene flow matching are compared to a disparity map of  $(I_r^0, I_r^0)$  computed with SGM [33]. Values within the support windows are middle rank confident values and will not be subjected to filtering in this first stage.

The second stage compares the scene flow matches using a forward-backward scene flow consistency check (see Fig. 2). In the backward scene flow, the correspondences will be computed with respect to  $I_r^1$  by the matching framework proposed in the previous section. Scene flow matches that yield differences bigger than a defined threshold will be removed. During the second stage, scene flow matches within the support window, as well as the components  $(u, v, d_1)$  of LiDAR pixels, are also subjected to filtering.

The clustering algorithm segments the remaining scene flow matches into similar regions. The scene flow of LiDAR pixels will be used as reference values in the clustering algorithm. Small clusters which are having less pixels than a given threshold, will be removed.

### D. LiDAR-Supported Robust Interpolation

The proposed consistency check algorithm generates a sparse set of matches. To speed up the interpolation, the remaining scene flow pixels, except LiDAR measurements, will be subjected to a sparsification process as in [8] (see Fig. 2). It selects within non-overlapping  $3 \times 3$  blocks the scene flow match with the highest consistency. After sparsification, we need to interpolate the remaining matches across the entire image into a dense scene flow (see Fig. 2). The geometry and motion matches of LiDAR pixels will initialize piecewise models of the edge-preserving interpolation approach [6] during which the remaining outliers due to image matching are rejected. The reference view image  $I_r^0$  is over-segmented into a set of superpixels and edges. The initial anchor point for each superpixel is its center of mass. Each superpixel is

associated with the closest LiDAR measurement and LiDAR match for interpolation of geometry and motion respectively. If the closest LiDAR point lies within the superpixel or is in its proximity (defined by a certain threshold), it will be used as new anchor point for this superpixel. The edge-aware neighborhoods for each superpixel are computed with respect to its anchor point. A geodesic distance transformation, which is built with respect to an input edge map, assists in finding the edge-aware neighborhoods. We use the result of the Structured Edge Detector framework [34] to compute our edge map of the reference image. Fig. 5 visualizes the use of LiDAR matches for interpolation.

Based on the anchor points and their neighborhoods, each superpixel will be initialized with piecewise models for superpixel flow. These models are refined with random search and propagation between superpixels. During model refinement, hypotheses for geometry and motion are directly rejected if they are inconsistent with the associated LiDAR measurement or match.

#### IV. EXPERIMENTS AND EVALUATION

We perform a series of experiments to verify the results of our new LiDAR-Flow algorithm. First, we quantify the impact of fusing a sparse LiDAR into each step of the proposed pipeline (overview in Fig. 2). Then, we follow with quantitative and qualitative comparison against a state-of-the-art image-only approach. Finally, the impact of using more samples of LiDAR measurements is evaluated.

##### A. Evaluation Data

**Evaluation Data:** The first aim of our evaluation is emphasizing the capability of LiDAR-Flow under challenging conditions. To the best of our knowledge, KITTI [1], [35] is the only data set that provides stereo images in a combination with LiDAR measurements for real traffic scenarios. The train set of KITTI-2015 consists of 200 consecutive frames; each with ground truth of the scene flow components  $(u, v, d_0, d_1)$ . The ground truth has been generated by synchronizing the stereo with a high dense LiDAR and the augmentation with 3D CAD models for all vehicles in motions [1]. The evaluation on KITTI computes the average of percentage pixels whose the endpoint error of at least one scene flow component exceeds 3 pixels or 5%. The results of scene flow are quantified as an average over all frames in terms of endpoint error and outlier rate for the disparity maps  $(D_0, D_1)$ , the optical flow  $(FL)$  and the scene flow  $(SF)$ .

**LiDAR-Flow Input Data:** To carry out our LiDAR-Flow approach, we take the advantages of KITTI-2015 train set to build the input frames (mentioned in Section III). Since the LiDAR measurements of KITTI are semi-dense and have been aligned into the reference view  $I_l^0$ , we have to preprocess the ground truth over all 200 frames. The preprocessing employs the optical flow ground truth to de-warp each geometry pixel of  $D_l^1$  into  $I_l^1$ . That mimics the real capture of the second LiDAR frame. The de-warping process is supported with an occlusion-handling algorithm. Finally,

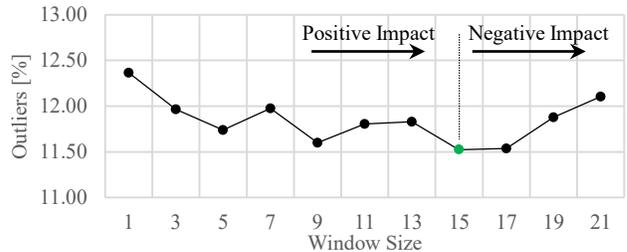


Fig. 6: Impact of the window support size on the final scene flow outliers.

we conduct a dense-to-sparse algorithm on each LiDAR frame  $D_l^0$  and the de-warped  $D_l^1$  to remain only 88 sparse LiDAR measurements (i.e. around 0.1% of ground truth) as an average over all train set frames. Each sparse LiDAR map is generated by moving a  $5 \times 5$  window with regular steps over the image and picking a LiDAR measurement within the defined window which is the nearest one to its center pixel. After de-warping and sparsification, the correspondences of the two sparse measurements  $(D_l^0, D_l^1)$  are completely dissolved in our input data.

##### B. LiDAR-Flow Pipeline Evaluation and Comparison to an Image-only Approach

Table I quantifies the results after each fusion step in our LiDAR-Flow pipeline. Since LiDAR-Flow incorporates LiDAR measurements and stereo images into dense scene flow estimation on image domain; [21], [23], [24] can not be conducted to our evaluation. They exploit the full resolution of LiDAR on point cloud domain and they remove LiDAR points on grounds which is inconsistent to our algorithm. As mentioned in related work, there is no directly related algorithm utilizing a sparse LiDAR and stereo images for scene flow computation. Thus, we compare to the image-only scene flow algorithm SFF++ [5] in its dual-frame version. We compare against SFF++ [5], since it follows a similar concept of sparse-to-dense interpolation; although it does not utilize any LiDAR data. The comparison to SFF++ [5] involves the main steps of matching, consistency check and interpolation in Sections III-B, III-C and III-D respectively. Fig. 7 visualizes also the superior performance of our LiDAR-Flow against SFF++ [5] under three challenging scenarios in terms of  $(FL)$ ,  $(D_0)$  and error map which maps the inliers of scene flow with green color.

**Support Window:** Various sizes of the proposed support window are tested to check the impact into the final accuracy. We assume a symmetrical support window in our investigation. Fig. 6 illustrates that increasing the size performs better accuracy until  $15 \times 15$ . However, the use of bigger sizes can drop the final accuracy down for the reasons mentioned in Section III-A. The following results are performed using the  $15 \times 15$  support window.

**Matching:** Matching between stereo images only results in outlier rate 39.09%. With pushing the LiDAR measurements directly into matching algorithm, we constraint the space

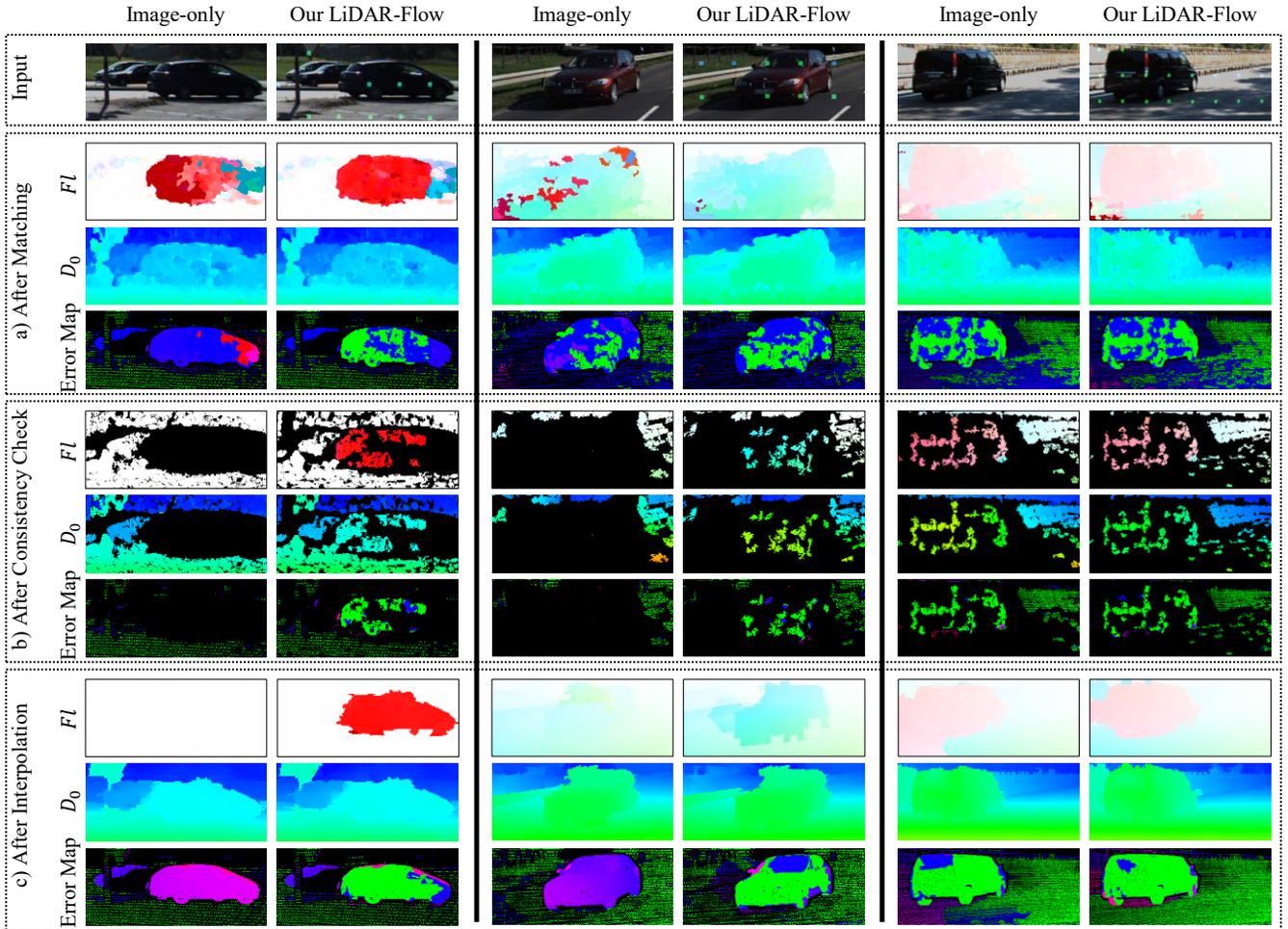


Fig. 7: Three examples visualize the impact of our LiDAR-Flow compared to an image-only SFF++ [5]. Our LiDAR-Flow performs superior in challenging image regions with shadows and low illumination. Exactly these regions are of high importance for applications like autonomous driving or mobile robots.

TABLE I: Quantitative evaluation for each single processing in Fig. 2, and comparison of LiDAR-Flow against image-only SFF++ [5] across the main steps: matching, consistency check and interpolation.

		EPE [px]			Outliers [%]				Density
		$D_0$	$D_1$	$Fl$	$D_0$	$D_1$	$Fl$	$SF$	[%]
Matching	im-only	7.30	10.98	38.64	12.48	28.86	32.02	39.09	100
	im+LiDAR	6.90	14.19	36.91	12.48	28.70	32.67	39.55	100
	im+LiDAR+win	6.48	12.20	37.72	11.66	27.95	32.15	38.89	100
	im+LiDAR+win+const1	<b>6.56</b>	10.65	35.11	11.67	27.62	31.82	38.62	100
	<b>LiDAR-Flow: im+LiDAR+win+const1+const2</b>	6.57	<b>10.61</b>	<b>34.81</b>	<b>11.65</b>	<b>27.59</b>	<b>31.70</b>	<b>38.46</b>	100
SFF++ [5]		7.30	11.14	38.02	12.48	28.82	31.95	39.04	100
Consist. Check	Disp+LiDAR	0.85	6.50	23.49	2.01	16.00	20.15	22.71	63.75
	<b>LiDAR-Flow: disp+LiDAR &amp; backward+LiDAR</b>	<b>0.71</b>	<b>0.93</b>	1.46	<b>1.21</b>	<b>2.46</b>	2.97	<b>4.21</b>	37.30
	SFF++ [5]	0.89	1.01	<b>1.11</b>	2.66	3.39	<b>2.37</b>	5.04	40.89
Interpolation	Spars+LiDAR	0.75	1.36	2.98	1.50	3.69	4.40	6.01	5.78
	Spars+LiDAR & Interp-only	1.17	1.69	5.05	5.30	8.55	10.32	12.27	100
	<b>LiDAR-Flow: Spars+LiDAR &amp; Interp+LiDAR</b>	<b>1.14</b>	<b>1.62</b>	<b>4.46</b>	<b>5.02</b>	<b>8.07</b>	<b>9.61</b>	<b>11.52</b>	100
	SFF++ [5]	1.33	1.88	5.32	6.44	9.67	12.24	14.51	100

of finding other matches of  $(u, v, d1)$  on the untextured regions. However, we affect badly reaching the optimal flow components in neighbor pixels during the propagation. One of the neighbor pixels within the proposed support window

can reach the optimal matches with minimum matching cost. In addition, the use of matching cost constraint (const1) in Fig. 4 and the disparity space constraint (const2) optimize the outlier rate to 38.46%. The improvement is performed not

TABLE II: Quantitative evaluation of the final scene flow results for the LiDAR seeds, the  $15 \times 15$  support window and overall pixels compared to SFF++ [5].

	SF Outliers [%]		
	Seeds	Window	Dense
SFF++ [5]	16.05	13.89	14.51
<b>LiDAR-Flow</b>	<b>7.88</b>	<b>8.70</b>	<b>11.52</b>
Average no. of evaluated points	88	6820	91875

only for the disparity map of  $D_0$  but also for the matches of optical flow components ( $Fl$ ) and the disparity map of ( $D_1$ ). The quantitative values of our LiDAR-Flow approach show improvement overall scene flow components compared to SFF++ [5]. The error map shows in Fig. 7 more inliers with low illuminated regions of the cars, in which the matching algorithm of SFF++ [5] fails.

**Consistency Check:** The consistency check of the disparity map at  $t_0$  using sparse LiDAR measurements (disp+LiDAR) removes the inaccurate geometry matches in which no contribution of LiDAR measurements is. That decreases the density of scene flow to 63.75%, but remains the reliable geometry matches. The forward-backward consistency (backward+LiDAR) removes unreliable matches of optical flow ( $u, v$ ) and disparity  $d_1$ . That decreases the density of matches to 37.30%. Together, consistency check of LiDAR results in more inliers to the critical regions (e.g. untextured) where consistency check of SFF++ [5] fails (see Fig. 7). The quantitative comparison compared to SFF++ [5] shows that less outlier rate of scene flow ( $SF$ ) and disparity maps ( $D_0, D_1$ ) but not with ( $Fl$ ). The reason of that, the contribution of LiDAR measurements in estimating the optical flow ( $Fl$ ) components is less than that with disparity maps ( $D_0, D_1$ ).

**Interpolation:** The sparsification (spars+LiDAR) continues removing less confident scene flow components and decreases the density of scene flow results to 5.7%. The interpolation before fusing LiDAR measurements (interp-only) is compared to the interpolation algorithm constrained by LiDAR measurements (interp+LiDAR). Overall components, LiDAR-Flow interpolation algorithm achieves a better outlier rate (11.52 %) than the rate of SFF++ [5] of 14.51%. Not only with the disparity ( $D_0$ ), but even with the optical components ( $Fl$ ) and disparity ( $D_1$ ), the use of 88 LiDAR measurements only, can resolve the challenging regions. Beside the quantitative comparison, Fig. 7 shows the substantial improvement under the challenging regions of shadowed areas and low illuminated areas compared to an image-only approach. In the aforementioned evaluation, error metrics are computed on average over the whole image. Table II indicates the final scene flow results in different image regions. The results are given for the position of the 88 input LiDAR measurements only, indicating a halved outlier rate compared to the image-only approach. In addition, the outliers in the  $15 \times 15$  support window are substantially reduced. This indicates that the utilized sparse LiDAR measurements

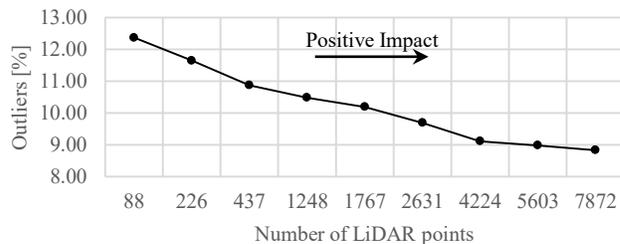


Fig. 8: Impact of number of LiDAR measurements on the final scene flow outliers.

serve as anchor points for these regions.

**Run Time:** Estimating scene flow for a consecutive frame requires 78 seconds on a single 4 cores CPU @ 3.5 GHz base frequency with C++. Parallelism is utilized for a small part of the algorithm.

### C. Impact of Number of LiDAR Measurements

We test the utilization of more LiDAR measurements and their impact on the final scene flow outlier rate. The proposed support window in Section III-A is neglected in order to avoid overlapping between LiDAR support windows. Fig. 8 illustrates that the accuracy tends to be better with more LiDAR measurements. However, even with very sparse LiDAR measurements (88 points only), we achieve already significant improvement. Such sparse LiDARs are comparably cheap and lead – in combination with our LiDAR-Flow – to the desired increased robustness in challenging image regions.

## V. CONCLUSIONS

In this paper, we proposed LiDAR-Flow to estimate dense scene flow from a fusion of sparse LiDAR and stereo images for resolving the challenges of textureless objects, shadows and poor illumination. We tightly integrated highly accurate LiDAR measurements into each processing step of our pipeline. They initialized image-based matching as well as served as anchor points for the consistency check in removing mismatches. In addition, we incorporated the sparse LiDAR measurements as constraints to increase the robustness of the sparse-to-dense interpolation. In our comprehensive evaluation on KITTI, we demonstrated the impact of each design decision of our LiDAR-Flow. Compared to image-only approaches, LiDAR-Flow resolved robustly the lack of information in challenging image regions and improves the overall accuracy of scene flow. Especially in critical scenarios – see Fig. 7 with oncoming cars, low illumination, shadows – our LiDAR-Flow performed superior.

## REFERENCES

- [1] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [2] J. Čech, J. Sanchez-Riera, and R. Horaud, “Scene flow estimation by growing correspondence seeds,” in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

- [3] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, and D. Cremers, "Stereoscopic scene flow computation for 3d motion understanding," *International Journal of Computer Vision (IJCV)*, 2011.
- [4] Q. M. Thomas, O. Wasenmüller, and D. Stricker, "Delio: Decoupled lidar odometry," in *IEEE International Conference on Intelligent Vehicles Symposium (IV)*, 2019.
- [5] R. Schuster, O. Wasenmüller, C. Unger, G. Kusch, and D. Stricker, "SceneFlowFields++: Multi-frame matching, visibility prediction, and robust interpolation for scene flow estimation," *arXiv preprint arXiv:1902.10099*, 2019.
- [6] Y. Hu, Y. Li, and R. Song, "Robust interpolation of correspondences for large displacement optical flow," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [7] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [8] R. Schuster, O. Wasenmüller, G. Kusch, C. Bailer, and D. Stricker, "Sceneflowfields: Dense interpolation of sparse scene flow correspondences," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018.
- [9] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *IEEE International Conference on Computer Vision (ICCV)*, 1999.
- [10] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers, "Efficient dense scene flow from sparse or dense stereo data," in *European Conference on Computer Vision (ECCV)*, 2008.
- [11] R. Li and S. Sclaroff, "Multi-scale 3d scene flow from binocular stereo sequences," *Computer Vision and Image Understanding (CVIU)*, 2008.
- [12] T. Basha, Y. Moses, and N. Kiryati, "Multi-view scene flow estimation: A view centered variational approach," *International Journal of Computer Vision (IJCV)*, 2013.
- [13] C. H. Hung, L. Xu, and J. Jia, "Consistent binocular depth and scene flow with chained temporal profiles," *International Journal of Computer Vision (IJCV)*, 2013.
- [14] F. Huguët and F. Devernay, "A variational method for scene flow estimation from stereo sequences," in *IEEE International Conference on Computer Vision (ICCV)*, 2007.
- [15] C. Vogel, K. Schindler, and S. Roth, "3d scene flow estimation with a rigid motion prior," in *IEEE International Conference on Computer Vision (ICCV)*, 2011.
- [16] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *IEEE International conference on computer vision and pattern recognition (CVPR)*, 2015.
- [17] T. Yoshida, O. Wasenmüller, and D. Stricker, "Time-of-flight sensor depth enhancement for automotive exhaust gas," in *IEEE International Conference on Image Processing (ICIP)*, 2017.
- [18] E. Herbst, X. Ren, and D. Fox, "Rgb-d flow: Dense 3-d motion estimation using color and depth," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.
- [19] J. Quiroga, T. Brox, F. Devernay, and J. Crowley, "Dense semi-rigid scene flow estimation from rgb-d images," in *European Conference on Computer Vision (ECCV)*, 2014.
- [20] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers, "A primal-dual framework for real-time dense rgb-d scene flow," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [21] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard, "Rigid scene flow for 3d lidar scans," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [22] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *European Conference on Computer Vision (ECCV)*, 2010.
- [23] X. Liu, C. R. Qi, and L. J. Guibas, "Learning scene flow in 3d point clouds," *arXiv preprint arXiv:1806.01411*, 2018.
- [24] A. Behl, D. Paschalidou, S. Donné, and A. Geiger, "Pointflownet: Learning representations for 3d scene flow estimation from point clouds," *arXiv preprint arXiv:1806.02170*, 2018.
- [25] A. K. Ushani, R. W. Wolcott, J. M. Walls, and R. M. Eustice, "A learning approach for real-time temporal scene flow estimation from lidar data," in *IEEE International Robotics Conference on Automation (ICRA)*, 2017.
- [26] A. K. Ushani and R. M. Eustice, "Feature learning for scene flow estimation from lidar," in *Conference on Robot Learning (CoRL)*, 2018.
- [27] V. Vaquero, A. Sanfeliu, and F. Moreno-Noguer, "Hallucinating dense optical flow from sparse lidar for autonomous vehicles," in *IEEE International Conference on Pattern Recognition (ICPR)*, 2018.
- [28] F. Ma and S. Karaman, "Sparse-to-dense: Depth prediction from sparse depth samples and a single image," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [29] W. Maddern and P. Newman, "Real-time probabilistic fusion of sparse 3d lidar and dense stereo," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [30] D. Huber, T. Kanade, et al., "Integrating lidar into stereo for fast and improved disparity computation," in *IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, 2011.
- [31] K. Park, S. Kim, and K. Sohn, "High-precision depth estimation with the 3d lidar and stereo fusion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [32] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2011.
- [33] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on pattern analysis and machine intelligence (TPAMI)*, 2008.
- [34] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *IEEE international conference on computer vision (ICCV)*, 2013.
- [35] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving?" in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.