deepRGBXYZ: Dense Pixel Description Utilizing RGB and Depth with Stacked Dilated Convolutions

Ramy Battrawy[1], René Schuster[1], Oliver Wasenmüller[1,2], Qing Rao[3], Didier Stricker[1,4]

***Abstract*— In this paper, we propose deepRGBXYZ – a feature descriptor to represent pixels for robust dense pixel matching. To this end, we concatenate RGB image (appearance) information with the depth (geometric) information represented as XYZ in order to build a robust descriptor which is more invariant to photometric and geometric changes. Both information (RGB and depth) are embedded as an early fusion into one neural network which is based on stacked dilated convolutions for enlarging the receptive field. We alleviate the limitations of image-only descriptors especially within ill-conditioned light regions or textureless objects. Additionally, we overcome the difficulty of using depth-only information which show less descriptive details compared to image-only. We demonstrate the superior accuracy of our deepRGBXYZ descriptor against the state-of-the-art image-only descriptors and we verify our design decision. In addition, we investigate the superior robustness of our deepRGBXYZ descriptor by bringing it into the application of optical flow and scene flow estimation on the established data sets KITTI and FlyingThings3D.**

I. Introduction

Pixel-wise matching is one of the fundamental requirements in many computer vision problems such as image retrieval, object recognition and flow estimation. The task of flow estimation presents more difficulty for aligning scenes with dynamic objects. More attention was paid recently in this task which can increase the robust perceptual information of the surroundings and the dynamic changes for autonomous driving systems. Here, robust local feature descriptors play a significant role for finding dense accurate matches through comparing the distance of their local descriptors (i.e. feature maps) [1], [2].

Many applications show a widespread use of handcrafted descriptors in the past decade such as SIFT [5], DAISY [6] and HOG [7]. However, the recent advances in deep neural networks lean themselves to compute patch-based descriptors by pruning the image contents into rich patches with strong features (i.e. keypoints) and obtaining less norm distances between the local descriptors of the similar patches (i.e. correspondences) [8], [9], [10].

Furthest size of these patches increases the receptive field of the feature space and encodes more details into the feature maps. In this context, SDC [3] showed recently a promising robustness and fast training approach based on triplet-based similarity patches. It aims to enrich the textural information

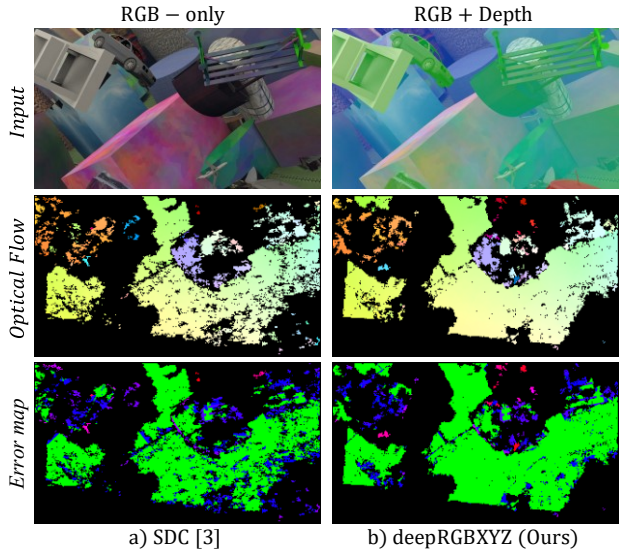

Fig. 1: Our deepRGBXYZ feature descriptor embeds photometric and geometric information for improving dense pixel-level matching. In the context of optical flow estimation, our deepRGBXYZ achieves more density and lower error rate (green color in Error map encodes inliers) compared to image-only descriptor like SDC [3]. The comparison is performed by CPM [1] on FlyingThings3D [4].

of the computed descriptor by increased receptive field. However, the robustness of the aforementioned descriptors depends on the provided details within the view of the receptive field. Thus, insufficient details within the receptive field due to the ill-conditioned light or textureless regions can limit the accuracy of any image-only descriptors and cause inaccurate matching. Such regions can be quite often faced in outdoor scenes especially in the perception applications of autonomous vehicles and inaccuracy in these areas can cause dangerous situations.

With the wide availability of depth sensors (such as LiDAR or RGB-D), many approaches aim to go further toward using the depth and geometric clues of the observed scene. However, the descriptors from only 3D information provide normally less dense feature description compared to image-only descriptors.

The fusion of both modalities; appearance (represented as RGB image) and geometric (produced by 3D sensors); proves a very impressive robustness in semantic segmentation [11], odometry [12], object detection [13] and many other fields. Here, the key challenge is to utilize the geo-

[1]DFKI – German Research Center for Artificial Intelligence, Germany: firstname.lastname@dfki.de
[2]University of Applied Sciences Mannheim, Mannheim, Germany: o.wasenmueller@hs-mannheim.de
[3]BMW Group, Germany: firstname.lastname@bmw.de
[4]University of Kaiserslautern - TUK, Kaiserslautern, Germany

metric information in a proper way and to complement the information of RGB image.

We aim in this paper to concatenate RGB image and depth together in a CNN-based dense descriptor with large receptive field. In this paper, our contributions are the following:

- We propose deepRGBXYZ – the concatenation of rich RGB information (generated by image sensor) and depth (produced by 3D sensor) for learning local image descriptor.
- We investigate several representations of depth with RGB fusion.
- We verify our design decision compared to the architectures of recent learning-based descriptors.
- We show the superior accuracy of deepRGBXYZ descriptor in pixel-wise matching for optical flow and scene flow estimation.

II. Related Work

Traditionally, local image features are the common types of information which play a crucial role for matching purpose. Several handcraft descriptors, such as SIFT [5], DAISY [6] and HOG [7] encode the local features into representative vectors. They consider defined patches to form invariant and descriptive information. However, they fail to represent high robustness in many cases. With the recent advances in deep learning, numerous researches focus on the powerful tools of neural networks to replace the handcrafted descriptors. Indeed, they show high accuracy compared to the handcrafted ones for matching purpose [14]. Many state-of-the-art learning-based approaches apply the patch-based similarity for computing descriptors. In this context, MatchNet [15], DeepDesc [9] and DeepCompare [16] employ a Siamese network [17] and learn (non-) similar patches using non-linear distance metrics. PN-Net [18], Hard-Net [19] and L2-Net [20] go further by using triplet-based similarity network [21] and propose losses to separate the distribution of matching and non-matching pairs. Different from L2-Net [20] architecture, GeoDesc [22] follows each convolutional layer by batch normalization except the last layer. Moreover, it adds geometric constraints for training patches with strong geometric similarities. To this end, it takes advantages from 3D information to measure the similarity between patches by considering camera position and surface normal. UCN [23] optimizes a deep metric learning to directly learn a feature space that preserves either geometric or semantic similarity, it shows more ability for extracting dense correspondences. SDC [3] proposes a novel method to increase the receptive field. It stacks multiple dilated convolutions in parallel and combines each output by concatenation to form one SDC layer. It enhances the robustness against photometric variations for dense matching especially for flow estimation. Although all of the aforementioned CNN-based descriptors show impressive accuracy for matching purpose compared to handcrafted designs, they require sufficient details in their receptive fields for more robust computation.

Alternative to 2D descriptors, series of 3D conventional descriptors [24], [25], [26] are used for 3D matching. Unlike to these descriptors, CSHOT [27] and BRAND [28] combine texture and geometry for 3D registration, however they are not suitable for dense correspondences registration. Following the CNN-based solutions, 3DMatch [29] introduces 3D CNN-based descriptor by training 3D patches using volumetric representation. The method uses 3D convolutional networks and presents a superior accuracy compared to the conventional ones. However, using 3D-only descriptors cannot generate robust descriptors for matching co-planar surfaces in which insufficient feature details are expected. Some conventional designs of 3D descriptors introduce a combination between texture and geometry for 3D matching [27], [28]. They are useful for sparse matching purposes, but they seem not suitable for dense use.

In this paper, we introduce deepRGBXYZ for involving depth information efficiently with appearance knowledge in order to enhance the dense matching on image domain. To serve our goal, we follow the concept of SDC [3], but we consider the additional information of depth in our descriptor.

III. Our Dense Feature Descriptor

Our deepRGBXYZ is based on combining RGB image and depth information on image domain. 3D sensors can perceive measurements as depth maps on image domain using pattern projection, Time-of-Flight or as point clouds using LiDAR sensors. In case of capturing RGB and depth by two different sensors, synchronization and well-calibration are the basic demand for this combination. Additionally, we assume that intrinsic calibration, i.e. the principal points of 2D image sensor and focal length are known. Thereby the alignment and projection of depth information into image plane are possible.

A. Geometric Representation and Fusion Strategy

A proper selection of geometric representation of depth is the anchor of many fusion designs. A direct use of depth representation is inspired as an early fusion with RGB for semantic segmentation [30]. We verified different approaches for our needs.

One representation, called HHA [31], encodes depth into three channels; horizontal disparity, height above ground, and the angles between local surface normal and the gravity direction. Estimating the gravity direction, surface normal and ground are the main components for high quality.

Other approaches work completely on 3D domain using the Voxel representation which shows a strong potential in 3D object detection [32].

Among these representations, we verify that involving the 3D Cartesian location XYZ can complement the image information to generate a robust accuracy. Learning 3D Cartesian coordinates in one-hot pixel by channel-wise concatenation to image tensor is an advanced research which allows the network to learn completely or partially the translation invariance [33]. We investigate this design decision in our experiments in Section IV-A.

We follow this principle and concatenate depth as 3D Cartesian coordinates with RGB image. So that we have an

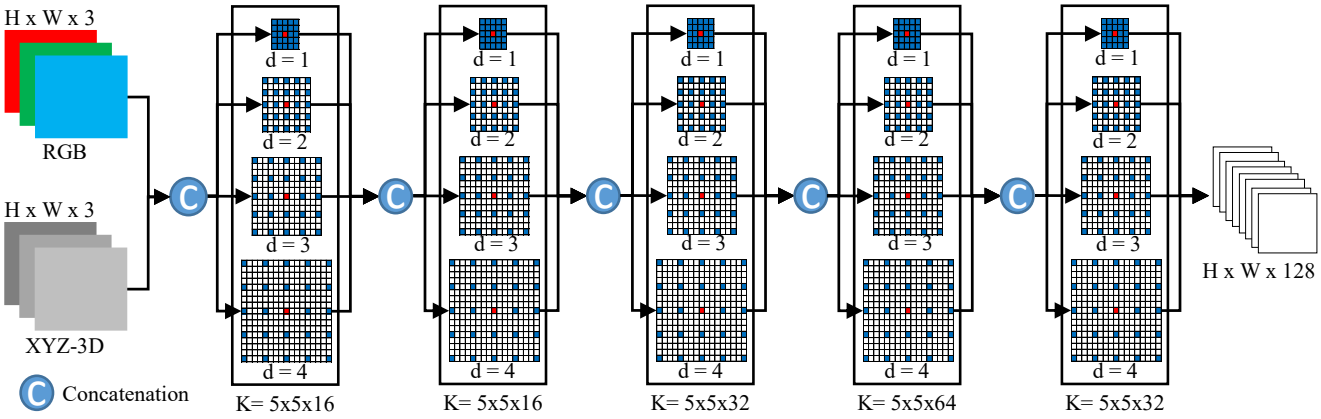

Fig. 2: Our deepRGBXYZ architecture consists of 5 layers. Each layer applies 4 convolutions with kernels (K) and dilation rates (d). The output feature map is dense with 128 channels.

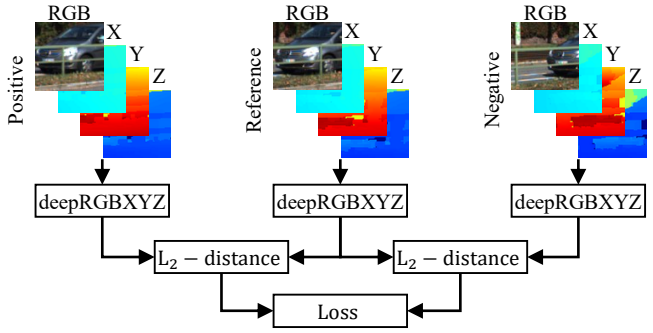

Fig. 3: Patch-based triplet training utilized in our deepRGBXYZ network.

early fusion architecture with 6 channels image tensor as shown in Fig. 2.

B. Network Architecture

CNN-based solutions result in promising accuracy compared to the handcrafted systems. Many of the recent networks employ multiple of max-pooling and stride convolutions to present the spatial coherence of the nearest pixels into the resulted feature maps. They perform efficiently for image classification tasks but they reduce significantly the spatial resolution and generate sparse feature maps. Some architectures follow the resulted feature maps by bilinear interpolation to recover the full resolution of feature responses [34] and others offer deconvolutional layers [35]. Tendency of using dilated or Atrous convolutions shows a better choice for maintaining the full resolution of feature responses. They differ from the standard convolutional kernels by alternating the dilation rates of convolutional kernels. This principle shows strong potential for semantic segmentation [36] as well as for learning descriptors [3]. Both approaches are stacking layers with increasing rates of dilation.

We take the advantages of this design and originate our fusion to improve the distinctiveness of the feature maps as much as possible and to support the image regions which lack some details in their receptive fields with depth information. Here, we increase at the same time the receptive field for 3D information represented as XYZ and we support the contextual information of 2D patches with 3D clues. By early fusion, both 2D and 3D information share the same receptive field. We consider the same parameters of the architecture SDC [3]; we stack 5 layers; each applies 4 parallel convolutions with 5×5 kernels and with dilation rates 1, 2, 3 and 4. The size of receptive field is 81 pixels as shown in Fig. 2.

C. Training Details and Loss Function

We apply in our training the triplet-based network [21] as shown in Fig. 3. The core of this approach encodes the similar feature maps to be closer than the dissimilar ones. To this end, the deepRGBXYZ training network accepts three parallel patches – reference, positive and negative patches – with shared weights. Sampling the images into patches follows the process of [3]; where the reference and positive patches are supposed to be with strong similarity and the negative one is considered to be with large distance to the reference one.

Hence, we aim to infer 3D information as an input for our training, we select the data sets which offer depth data and optical flow ground truth to support sampling the images into the needed patches. Such requirements are available in the established KITTI 2015 [37] and FlyingThings3D [4] data sets. Thus, the optical flow ground truth facilitates the sampling from second view the positive patch which is strongly correlated to the reference patch in the first view. The negative patch is obtained also from the second view but with altered displacement which can be semi-correlated to the positive patch.

We use the thresholded hinge embedding loss function for training the aforementioned patches [38]. It tries to minimize the L_2 distance between reference and positive patches and to increase the L_2 between reference and negative patches.

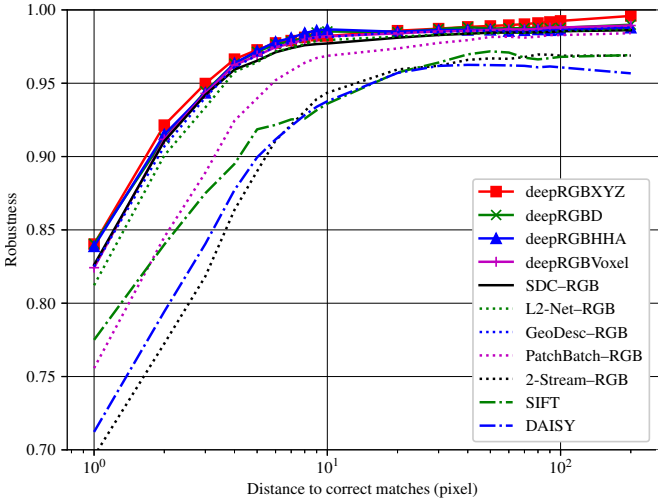

Fig. 4: Robustness curve of our deep descriptor by using RGB and various geometric representation. Our deepRGBXYZ descriptor presents more robustness compared to other geometric representations.

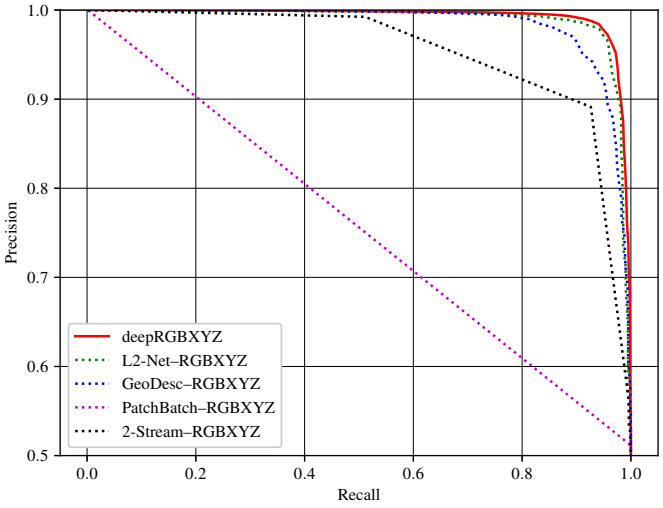

Fig. 6: Recall-precision curve compared to the state-of-the-art architectures with RGB-XYZ concatenation.

IV. Experiments

We conduct our deepRGBXYZ descriptor to versatile test environments to show the superior performance among other tested architectures. Firstly, we verify different types of geometric representations and compare to other CNN-based descriptors. Secondly, we justify our design decision. Finally, we integrate our deepRGBXYZ into optical flow and scene flow algorithms and we compare to an image-only descriptor [3].

We utilize KITTI 2015 train set [37] in the Sections IV-A and IV-B. In this context, we split KITTI into two groups; one for training purpose and another for validation and test. KITTI offers a beneficial alignment of depth information (generated by LiDAR) to image plane. However, the resulted projection does not fill the complete image with depth. To overcome this issue, we fill these pixels with depth by using a robust interpolation method [39]. Additional to KITTI, we use FlyingThings3D (FT3D) [4] data set in the last section of our experiments. We train our deepRGBXYZ and the other methods for comparison on each data set separately and we verify also each data set based on its trained model.

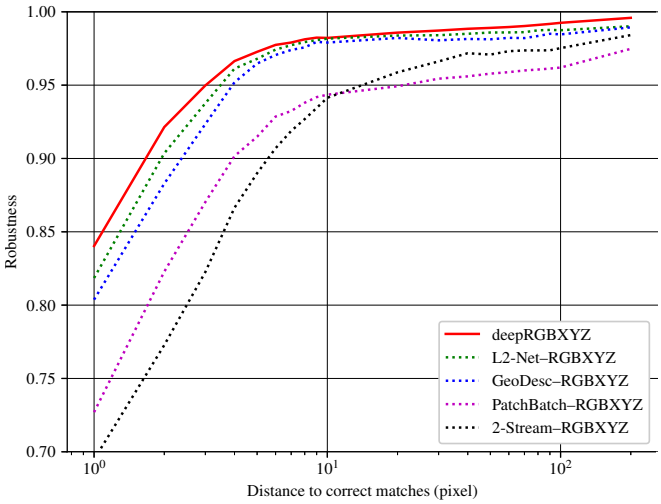

Fig. 5: Robustness curve compared to the state-of-the-art architectures with RGB-XYZ concatenation.

A. Comparison to other 3D Representations and Image-only Descriptors

In this section, we verify the robustness of our deep descriptor by concatenating different representations of depth to RGB. Hence, we concatenate channel-wise RGB with depth map, HHA map [31] and XYZ coordinates to compute (deepRGBD), (deepRGBHHA) and (deepRGBXYZ) respectively as an early fusion fashion. Thus, the input tensor with (deepRGBD) has 4 channels and each of (deepRGBHHA) and (deepRGBXYZ) has 6 channels. We verify also Voxel representation by partitioning into 3D patches and learning in 3D-CNN stream beside the 2D one then concatenating the feature maps in a late fusion fashion (deepRGBVoxel).

The robustness of the aforementioned representations is justified to each other and compared as well to image-only

TABLE I: Mean Average Precision (mAP) (%) for state-of-the-art architectures by feeding them once using RGB tensor and another with RGBXYZ.

Architecture	Patch Size	RGB	RGB-XYZ
SDC [3]	81	98.55	–
PatchBatch [40]	51	97.79	51.04
2-Stream [16]	64	96.96	91.64
GeoDesc [22]	32	98.41	98.09
L2-Net [20]	32	98.42	98.63
deepRGBXYZ	81	–	**99.14**

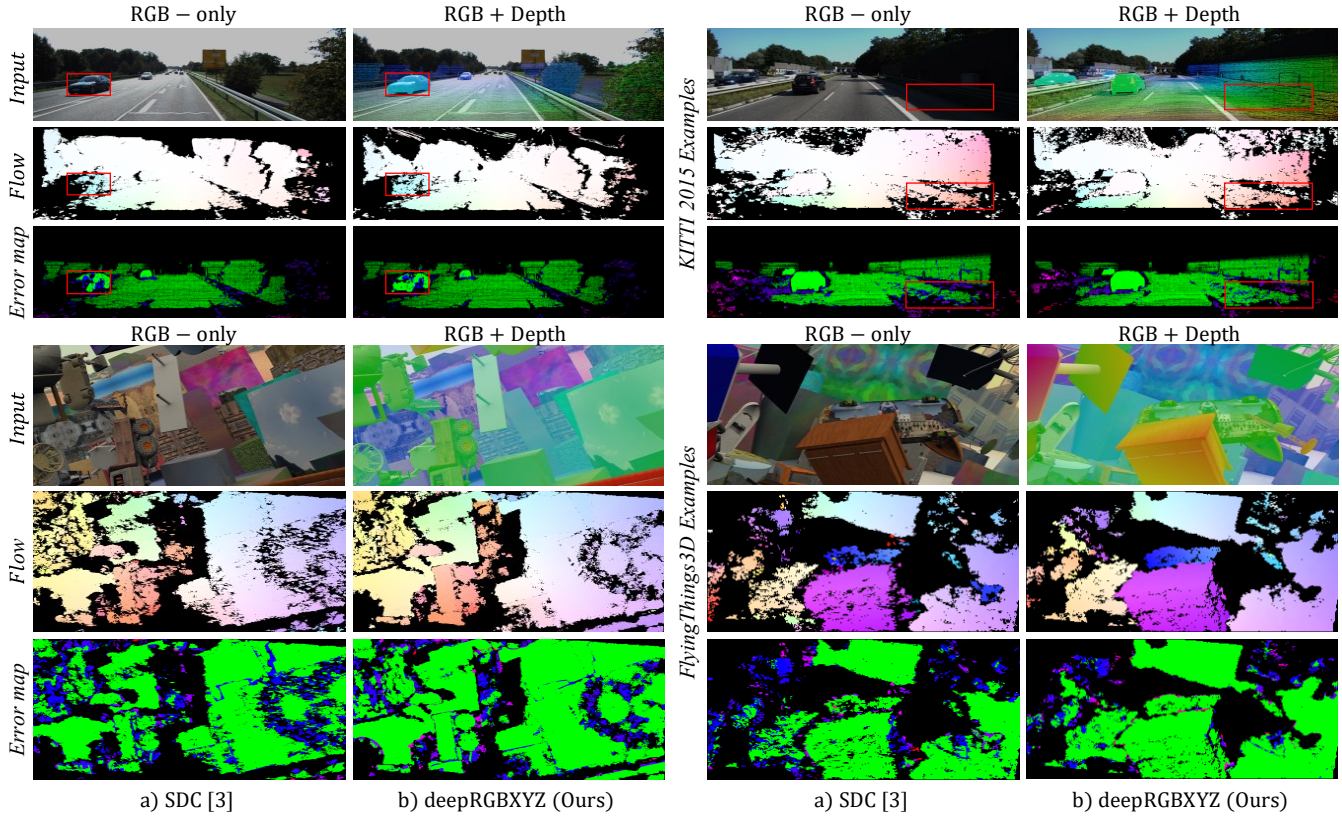

Fig. 7: The optical flow after consistency check of FF++ [2] (first and second columns) and optical flow from CPM [1] (third and last columns) are computed once using SDC [3], and another with our deepRGBXYZ. The visual comparison as well as the quantitative results in Table II and III show that our deepRGBXYZ descriptor increases the density of matches and increases the inliers (encoded as green in Error map). The results are impressive in the low-illuminated areas (i.e. textureless) marked as red rectangles in KITTI examples and near to depth discontinuities areas in FT3D examples. Note that the upper side of KITTI images appears less dense by using our deepRGBXYZ descriptor because it is not covered by depth and the interpolation in those areas is not accurate enough.

TABLE II: Comparison between image-only SDC [3] descriptor and our deepRGBXYZ in the context of flow estimation using FF++ [2].

Data set		Matching				Consistency Check						Interpolation			
		SDC [3]		**deepRGBXYZ**		SDC [3]			**deepRGBXYZ**			SDC [3]		**deepRGBXYZ**	
		EPE	KIO	EPE	KIO	EPE	KIO	Density	EPE	KIO	Density	EPE	KIO	EPE	KIO
KITTI	bg	28.76	24.87	21.55	24.00	3.81	8.51	–	3.31	7.47	–	6.98	14.37	6.57	14.29
	fg	9.77	16.79	6.99	15.91	2.20	9.98	–	2.11	8.91	–	4.58	12.61	5.49	13.49
	all	27.74	25.00	20.52	24.06	3.54	8.91	78.78	3.07	7.78	78.50	7.06	15.03	6 .93	15.17
FT3D		37.10	32.55	36.26	27.30	3.41	9.85	68.96	2.99	7.60	73.85	9.36	16.72	9.62	15.69

descriptors: SDC [3], L2-Net [20], GeoDesc [22], PatchBatch [40], 2-Stream [16], SIFT [5] and DAISY [6]. The mentioned learning-based descriptors are trained using triplet-based principle with the same epochs, batch size, hyper-parameters and threshold of the hinge loss function. The patch size of each learning-based descriptor is selected based on the receptive field size of its neural network design as shown in Table I. The robustness of fusing 3D Cartesian locations with RGB (deepRGBXYZ) overcomes other geometric representations and outperforms the image-only descriptors as shown in Fig. 4. We measure the robustness curve in Fig. 4 and 5 by computing the possibility of obtaining smaller L_2 distances between reference and positive patches (i.e. correct matches) compared to those between reference and negative ones. More negative patches are generated with various distances (in pixels) to the positive ones (i.e. correct matches).

B. *Verification of deepRGBXYZ Architecture*

In this part of experiments, we verify the design of deepRGBXYZ against other recent neural networks. We replace the image-only tensor of the aforementioned image-only descriptors in Section IV-A, with the concatenated input

TABLE III: Comparison between image-only SDC [3] descriptor and our deepRGBXYZ in the context of flow estimation using CPM [1].

Data set		SDC [3]			**deepRGBXYZ**		
		EPE	KIO	Density	EPE	KIO	Density
KITTI	bg	3.40	8.87	–	3.18	8.01	–
	fg	2.38	10.97	–	2.18	9.76	–
	all	3.24	9.45	8.74	3.01	8.56	8.77
FT3D		3.63	11.35	7.64	2.81	8.01	8.06

tensor (i.e. RGB + XYZ). We train with the same strategy of deepRGBXYZ and we validate the results. Among the tested architectures, the robustness curve in Fig. 5 shows a superior accuracy of deepRGBXYZ architecture over all distances to correct matches. We compare by computing the precision, recall curve in Fig. 6 which presents another confidence of the superior accuracy with deepRGBXYZ descriptor using stacked dilated convolution method.

Moreover, we compute mean Average Precision (mAP) once using the image-only tensor and another with RGBXYZ tensor as in Table I. The use of RGBXYZ tensor decreases significantly the accuracy with PatchBatch [40] and shows also negative influence on the other neural networks except L2-Net [20] which shows marginal improvement by using RGBXYZ tensors. Among them, the architecture of our deepRGBXYZ presents a superior accuracy and outperforms all of other architectures using RGBXYZ tensors.

C. Accuracy of Matching-based Optical Flow and Scene Flow Estimation

For the final part of our experiments, we verify the computed deepRGBXYZ descriptor for dense matching in the context of optical flow and scene flow estimation. To this end, we conduct our analysis to optical flow algorithms of Flow-Fields (FF++) [2] and CPM [1] and scene flow algorithm of LiDAR-Flow [41] by replacing their original descriptors with our deepRGBXYZ. These algorithms follow coarse-to-fine pyramid images for seeking the matches, however, FF++ and LiDAR-Flow are dense matching approaches and the matching phase in each of them is followed by consistency check for removing mismatches and then sparse-to-dense interpolation [42] for filling the gaps after consistency check.

Since SDC [3] in its nature is developed for dense matching and outperforms the aforementioned CNN-based descriptors in Section IV-A and IV-B, we compare against it to show the influence of the depth knowledge as 3D Cartesian representation. We utilize the over all 200 images in KITTI train set and 199 frames from FT3D train set; selected one frame each 100 frames (i.e. [0, 100, 200, ...]). We use the common metrics of KITTI represented as the average of end point error (EPE) and outliers rate (KIO)[%] for those pixels whose EPE is bigger than 3 pixels and their relative error $>$ 5% compared to ground truth. Additional to over all pixels evaluation (all), the dynamic parts called as foreground (fg) and static parts called as background (bg) in KITTI train set are evaluated. Green color in quantitative results marks better accuracy of our deepRGBXYZ compared to SDC [3] and the red color marks the less accuracy; where the saturated colors describe an absolute difference of bigger than 2.

The quantitative results of optical flow from FF++ [2] are shown in Table II. The comparison states that the image-based matching with our deepRGBXYZ descriptors reduces the outliers rate (KIO) by $\sim$1% and $\sim$5% on KITTI and FT3D respectively. In terms of EPE, our deepRGBXYZ shows also superior accuracy but significantly appears with KITTI data set over all KITTI components (fg), (bg) and all pixels. Additionally, our deepRGBXYZ descriptor contributes also the improvement with consistency check (inverse matching process) and outperforms SDC [3] in terms of EPE and KIO but decreases slightly the density with KITTI data set only. The results of interpolation shows that our deepRGBXYZ descriptor outperforms SDC [3] but not over all terms in KITTI data set. With FT3D, our deepRGBXYZ shows better accuracy in terms of outliers rate compared to SDC [3] but a little bite worse in terms of EPE. We have to mention here that the interpolation algorithm of FF++ [2] is completely independent of using any descriptors.

The quantitative comparison on CPM algorithm [1] is shown in Table III. Our deepRGBXYZ outperforms SDC [3] on both data sets KITTI and FT3D.

The qualitative results visualize also the superior accuracy compared to SDC [3] in Fig. 7. KITTI examples show that our deepRGBXYZ can resolve more inliers in low-illuminated areas within the red rectangles. FT3D examples also present more accuracy especially to the areas near to depth discontinuities.

In the context of scene flow estimation using LiDAR-Flow [41], we adapt our deepRGBXYZ to the algorithm and compare the results to SDC [3] in terms of estimating matches of disparities D_0 and D_1, optical flow terms Fl and SF as shown in Table IV. The matching step shows high accuracy over all terms but significantly in terms of EPE on KITTI data set and KIO on FT3D. This reflects also more accuracy after consistency check. After interpolation, we see some negative behavior in some terms but mostly our deepRGBXYZ outperforms the image-only descriptor SDC [3]. Like FF++ [2], the interpolation method in LiDAR-Flow [41] is completely independent of using any descriptors.

In the context of run time, our deepRGBXYZ requires 549.5 milliseconds for a 0.5 megapixel image on GPU GeForce RTX-2080 Ti using C++.

V. Conclusion

In this paper, we proposed our deepRGBXYZ descriptor which concatenates RGB image with 3D Cartesian locations of the observed scene to build a robust dense pixel descriptor for dense matching on image domain. The concatenation was embedded with dilated convolutions in order to increase the contextual information of RGB image and the depth at the same time. This recovered the lack of the information in some regions of the scene due to low illuminated regions or textureless objects, in which the image-only descriptors

TABLE IV: Comparison between image-only SDC [3] descriptor and our deepRGBXYZ in the context of scene flow estimation using LiDAR-Flow [41].

Data set			Matching				Consistency Check				Interpolation			
			SDC [3]		**deepRGBXYZ**		SDC [3]		**deepRGBXYZ**		SDC [3]		**deepRGBXYZ**	
			EPE	KIO	EPE	KIO	EPE	KIO	EPE	KIO	EPE	KIO	EPE	KIO
KITTI	D_0	bg	7.64	7.75	5.18	7.73	0.68	0.95	0.67	0.91	1.04	4.37	1.04	4.35
		fg	6.34	9.41	5.18	8.99	0.77	1.84	0.75	1.64	1.50	7.73	1.52	7.74
		all	8.44	8.77	5.91	8.70	0.70	1.11	0.68	1.04	1.14	4.98	1.14	5.00
	D_1	bg	16.49	24.50	10.88	23.72	0.88	2.07	0.85	1.98	1.42	6.55	1.42	6.65
		fg	10.19	15.31	5.89	14.69	0.99	3.12	1.34	2.86	1.95	11.75	1.89	10.85
		all	16.78	24.46	10.73	23.62	0.89	2.17	0.86	2.05	1.58	7.62	1.55	7.50
	Fl	bg	39.50	25.26	29.49	24.38	1.36	2.48	1.28	2.22	3.13	7.52	3.37	7.61
		fg	18.02	18.11	12.79	17.02	1.38	3.81	1.45	3.48	6.11	15.93	5.49	15.42
		all	38.84	25.65	28.71	24.67	1.30	2.50	1.22	2.17	4.11	9.24	3.98	9.09
	SF	bg	–	29.89	–	28.97	–	3.38	–	3.12	–	9.30	–	9.38
		fg	–	25.48	–	24.42	–	6.21	–	5.69	–	18.90	–	18.33
		all	–	30.94	–	29.96	–	3.62	–	3.28	–	11.19	–	11.05
FT3D	D_0	all	23.61	23.13	22.67	18.11	0.52	1.31	0.45	0.98	2.37	7.55	2.22	6.73
	D_1	all	32.78	30.49	38.64	27.34	0.83	1.86	0.67	1.38	3.24	9.49	3.23	9.03
	Fl	all	44.58	34.32	40.82	27.56	1.02	1.57	0.72	0.96	12.62	18.45	19.54	17.89
	SF	all	–	44.52	–	38.10	–	2.71	–	1.89	–	20.23	–	19.41

could be inaccurate. Exactly, these regions are very important for autonomous vehicles applications and accuracy here can increase the safety of these vehicles.

We compared several geometric representations of depth with early and late neural network fusion strategies and verified the robustness to our deepRGBXYZ descriptor. In addition, we justified our design decision compared to the recent CNN-based descriptors and showed that our deepRGBXYZ descriptor introduced a superior accuracy over all experiments. In our comprehensive evaluation on KITTI and FlyingThings3D, we demonstrated the impact of our deepRGBXYZ descriptor in the context of optical flow and scene flow estimation. Compared to image-only descriptors, our deepRGBXYZ resolved the lack of information in challenging image regions and improved the overall accuracy of matching-based optical flow and scene flow algorithms.

REFERENCES

[1] 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.

[2] R. Schuster, C. Bailer, O. Wasenmüller, and D. Stricker, "FlowFields++: Accurate Optical Flow Correspondences Meet Robust Interpolation," in *IEEE International Conference on Image Processing (ICIP)*, 2018.

[3] R. Schuster, O. Wasenmüller, C. Unger, and D. Stricker, "SDC-Stacked Dilated Convolution: A Unified Descriptor Network for Dense Matching Tasks," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *IEEE International Conference on Computer Vision (ICCV)*, 1999.

[6] E. Tola, V. Lepetit, and P. Fua, "DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo," *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2009.

[7] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[8] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[9] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer, "Discriminative Learning of Deep Convolutional Feature Point Descriptors," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[10] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua, "LIFT: Learned Invariant Feature Transform," in *European Conference on Computer Vision (ECCV)*, 2016.

[11] L. Deng, M. Yang, T. Li, Y. He, and C. Wang, "RFBNet: Deep Multimodal Networks with Residual Fusion Blocks for RGB-D Semantic Segmentation," *arXiv preprint arXiv:1907.00135*, 2019.

[12] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, "Fast Odometry and Scene Flow from RGB-D Cameras based on Geometric Clustering," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[13] C. Premebida, J. Carreira, J. Batista, and U. Nunes, "Pedestrian Detection Combining RGB and Dense LiDAR Data," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[14] J. L. Schonberger, H. Hardmeier, T. Sattler, and M. Pollefeys, "Comparative Evaluation of Hand-Crafted and Learned Local Features," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[16] S. Zagoruyko and N. Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[17] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature Verification using a" Siamese" Time Delay Neural Network," in *Advances in Neural Information Processing Systems (NIPS)*, 1994.

[18] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk, "PN-Net: Con-

joined Triple Deep Network for Learning Local Image Descriptors," *arXiv preprint arXiv:1601.05030*, 2016.

[19] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, "Working hard to know your neighbor's margins: Local descriptor learning loss," in *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[20] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] E. Hoffer and N. Ailon, "Deep metric learning using Triplet network," in *International Workshop on Similarity-Based Pattern Recognition (SIMBAD)*, 2015.

[22] Z. Luo, T. Shen, L. Zhou, S. Zhu, R. Zhang, Y. Yao, T. Fang, and L. Quan, "GeoDesc: Learning Local Descriptors by Integrating Geometry Constraints," in *European Conference on Computer Vision (ECCV)*, 2018.

[23] C. B. Choy, J. Gwak, S. Savarese, and M. Chandraker, "Universal Correspondence Network," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[24] A. E. Johnson and M. Hebert, "Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes," *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 1999.

[25] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning Point Cloud Views Using Persistent Feature Histograms," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2008.

[26] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[27] F. Tombari, S. Salti, and L. Di Stefano, "A Combined Texture-Shape Descriptor for Enhanced 3D Feature Matching," in *IEEE International Conference on Image Processing (ICIP)*, 2011.

[28] E. R. Nascimento, G. L. Oliveira, M. F. Campos, A. W. Vieira, and W. R. Schwartz, "BRAND: A Robust Appearance and Depth Descriptor for RGB-D Images," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[29] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, "3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor Semantic Segmentation Using Depth Information," *arXiv preprint arXiv:1301.3572*, 2013.

[31] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning Rich Features from RGB-D Images for Object Detection and Segmentation," in *European Conference on Computer Vision (ECCV)*, 2014.

[32] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[33] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, "An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution," in *Advances in Neural Information Processing Systems (NIPS)*, 2018.

[34] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[35] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *European Conference on Computer Vision (ECCV)*, 2014.

[36] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," in *International Conference on Learning Representations (ICLR)*, 2016.

[37] M. Menze and A. Geiger, "Object Scene Flow for Autonomous Vehicles," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[38] C. Bailer, K. Varanasi, and D. Stricker, "CNN-based Patch Matching for Optical Flow with Thresholded Hinge Embedding Loss," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[39] R. Schuster, O. Wasenmüller, C. Unger, G. Kuschk, and D. Stricker, "SceneFlowFields++: Multi-frame Matching, Visibility Prediction, and Robust Interpolation for Scene Flow Estimation," *arXiv preprint arXiv:1902.10099*, 2019.

[40] D. Gadot and L. Wolf, "PatchBatch: A Batch Augmented Loss for Optical Flow," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[41] R. Battrawy, R. Schuster, O. Wasenmüller, Q. Rao, and D. Stricker, "LiDAR-Flow: Dense Scene Flow Estimation from Sparse LiDAR and Stereo Images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[42] 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.