Supplementary Material: Learnable Cost Volume Using the Cayley Representation

Taihong Xiao[1][0000−0002−6953−7100], Jinwei Yuan[2], Deqing Sun[2][0000−0003−0329−0456], Qifei Wang[2] Xin-Yu Zhang[3], Kehan Xu[4], and Ming-Hsuan Yang[1,2][0000−0003−4848−2304]

[1] University of California, Merced {txiao3,mhyang}@ucmerced.edu
[2] Google Research {jinwei,deqingsun,qfwang,minghsuan}@google.com
[3] Nankai University xinyuzhang@mail.nankai.edu.cn
[4] Peking University yurina@pku.edu.cn

1 Proofs of Theorems

1.1 Proof of Theorem 1

The Theorem of Cayley representation was first given by Cayley in his paper [1]. However, the old paper is not available online. For convenience, we give a simple proof here.

Proof. First, we validate that the $\boldsymbol{P}$ is a orthogonal matrix. The condition that $\boldsymbol{S} \in \mathrm{SO}^*(n)$ ensures that $(\boldsymbol{I}+\boldsymbol{S})$ is invertible. Since $\boldsymbol{S}$ is skew-symmetric, so $\boldsymbol{S}^\top \boldsymbol{S} = -\boldsymbol{S}^2 = \boldsymbol{S}\boldsymbol{S}^\top$. Hence we have

$$\begin{aligned}
\boldsymbol{P}^\top \boldsymbol{P} &= (\boldsymbol{I}+\boldsymbol{S})^{-\top}(\boldsymbol{I}-\boldsymbol{S})^\top(\boldsymbol{I}-\boldsymbol{S})(\boldsymbol{I}+\boldsymbol{S})^{-1} && (1)\\
&= (\boldsymbol{I}+\boldsymbol{S}^\top)^{-1}(\boldsymbol{I}-\boldsymbol{S}^\top)(\boldsymbol{I}-\boldsymbol{S})(\boldsymbol{I}+\boldsymbol{S})^{-1} && (2)\\
&= (\boldsymbol{I}-\boldsymbol{S})^{-1}(\boldsymbol{I}-\boldsymbol{S}^\top-\boldsymbol{S}-\boldsymbol{S}^\top\boldsymbol{S})(\boldsymbol{I}+\boldsymbol{S})^{-1} && (3)\\
&= (\boldsymbol{I}-\boldsymbol{S})^{-1}(\boldsymbol{I}-\boldsymbol{S}^\top-\boldsymbol{S}-\boldsymbol{S}\boldsymbol{S}^\top)(\boldsymbol{I}+\boldsymbol{S})^{-1} && (4)\\
&= (\boldsymbol{I}-\boldsymbol{S})^{-1}(\boldsymbol{I}-\boldsymbol{S})(\boldsymbol{I}-\boldsymbol{S}^\top)(\boldsymbol{I}+\boldsymbol{S})^{-1} && (5)\\
&= (\boldsymbol{I}-\boldsymbol{S}^\top)(\boldsymbol{I}+\boldsymbol{S})^{-1} && (6)\\
&= (\boldsymbol{I}+\boldsymbol{S})(\boldsymbol{I}+\boldsymbol{S})^{-1} && (7)\\
&= \boldsymbol{I}. && (8)
\end{aligned}$$

Next, we need to show the uniqueness of the Cayley representation.

$$\begin{aligned}
& \boldsymbol{P} &&= (\boldsymbol{I}-\boldsymbol{S})(\boldsymbol{I}+\boldsymbol{S})^{-1} && (9)\\
\Longleftrightarrow\ & \boldsymbol{P}(\boldsymbol{I}+\boldsymbol{S}) &&= \boldsymbol{I}-\boldsymbol{S} && (10)\\
\Longleftrightarrow\ & \boldsymbol{P}+\boldsymbol{P}\boldsymbol{S} &&= \boldsymbol{I}-\boldsymbol{S} && (11)\\
\Longleftrightarrow\ & \boldsymbol{S}+\boldsymbol{P}\boldsymbol{S} &&= \boldsymbol{I}-\boldsymbol{P} && (12)\\
\Longleftrightarrow\ & (\boldsymbol{I}+\boldsymbol{P})\boldsymbol{S} &&= \boldsymbol{I}-\boldsymbol{P} && (13)\\
\Longleftrightarrow\ & \boldsymbol{S} &&= (\boldsymbol{I}+\boldsymbol{P})^{-1}(\boldsymbol{I}-\boldsymbol{P}) && (14)
\end{aligned}$$

Therefore, the skew-symmetric matrix $\boldsymbol{S}$ is uniquely represented by $\boldsymbol{P}$, which concludes the proof. □

1.2 Proof of Theorem 2

Before giving the proof, we would like to recall the definition of connectedness.

Definition 1 (Connectedness) *A set of matrices $\mathcal{G}$ is said to be connected if for all $\boldsymbol{A}$ and $\boldsymbol{B}$ in $\mathcal{G}$, there exists a continuous path $\boldsymbol{A}(t),\ 0 \leq t \leq 1$, lying with $\boldsymbol{A}(0) = \boldsymbol{A}$ and $\boldsymbol{A}(1) = \boldsymbol{B}$.*

The above definition of connectedness is actually path connectedness in topology. Now we begin our proof of Theorem 2.

Proof. Since the identity matrix $\boldsymbol{I} \in \mathrm{SO}^*(n)$, it suffices to prove that for any $\boldsymbol{X} \in \mathrm{SO}^*(n)$, there exists a continuous path $\boldsymbol{A}(t),\ 0 \leq t \leq 1$, such that $\boldsymbol{A}(0) = \boldsymbol{I}$ and $\boldsymbol{A}(1) = \boldsymbol{X}$. For any $\boldsymbol{X} \in \mathrm{SO}^*(n)$, we have its spectral decomposition

$$\boldsymbol{X} = \boldsymbol{P}^\top \mathrm{diag}(K_1, \ldots, K_q, 1, \ldots, 1)\boldsymbol{P}, \tag{15}$$

where the $\boldsymbol{P} \in \mathrm{O}(n)$ and $0 \leq q \leq n/2$, and

$$K_\lambda = \begin{pmatrix} \cos(\theta_\lambda) & -\sin(\theta_\lambda) \\ \sin(\theta_\lambda) & \cos(\theta_\lambda) \end{pmatrix}, \quad \theta_\lambda \in [-\pi, \pi), \quad \lambda = 1, \ldots, q. \tag{16}$$

If we put

$$K_\lambda(t) = \begin{pmatrix} \cos(t\theta_\lambda) & -\sin(t\theta_\lambda) \\ \sin(t\theta_\lambda) & \cos(t\theta_\lambda) \end{pmatrix}, \tag{17}$$

then the path required is

$$\boldsymbol{A}(t) = \boldsymbol{P}^\top \mathrm{diag}(K_1(t), \ldots, K_q(t), 1, \ldots, 1)\boldsymbol{P}. \tag{18}$$

□

2 More Results

2.1 Ranking on Sintel and KITTI 2015 Benchmark

The ranking results on the Sintel and KITTI 2015 benchmark can be found at `http://sintel.is.tue.mpg.de/results` and `http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=flow`. Here we capture the screenshot of the ranking results by March 8, 2020.

Final Clean

	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
ScopeFlow [2]	4.098	1.999	21.214	4.028	1.689	1.180	0.725	2.589	24.477	Visualize Results
MaskFlownet [3]	4.172	2.048	21.494	3.783	1.745	1.310	0.592	2.389	26.253	Visualize Results
Anonymous872 [4]	4.200	2.099	21.330	4.276	1.738	1.259	0.933	2.592	24.297	Visualize Results
FlowElements [5]	4.224	1.956	22.704	3.288	1.479	1.419	0.646	1.897	27.596	Visualize Results
SelFlow [6]	4.262	2.040	22.369	4.083	1.715	1.287	0.582	2.343	27.154	Visualize Results
MaskFlownet-S [7]	4.384	2.120	22.840	3.905	1.821	1.359	0.645	2.526	27.429	Visualize Results
VCN [8]	4.404	2.216	22.238	4.381	1.782	1.423	0.955	2.725	25.570	Visualize Results
LiteFlowNet3 [9]	4.448	2.089	23.681	3.873	1.755	1.344	0.754	2.503	27.471	Visualize Results
GCA-Net [10]	4.494	2.168	23.464	3.926	1.702	1.545	0.800	2.593	27.422	Visualize Results
ContinualFlow_ROB [11]	4.528	2.723	19.248	5.050	2.573	1.713	0.872	3.114	26.063	Visualize Results
LiteFlowNet3-S [12]	4.529	2.120	24.162	3.952	1.720	1.398	0.795	2.502	27.949	Visualize Results
MFF [13]	4.566	2.216	23.732	4.664	2.017	1.222	0.893	2.902	26.810	Visualize Results
IRR-PWC [14]	4.579	2.154	24.355	4.165	1.843	1.292	0.709	2.423	28.998	Visualize Results
PWC-Net+ [15]	4.596	2.254	23.696	4.781	2.045	1.234	0.945	2.978	26.620	Visualize Results
PPAC-HD3 [16]	4.599	2.116	24.852	3.521	1.702	1.637	0.617	2.083	30.457	Visualize Results
CompactFlow [17]	4.626	2.099	25.253	4.192	1.825	1.233	0.845	2.677	28.120	Visualize Results
PCF-F [18]	4.630	2.197	24.465	3.410	1.737	1.744	0.603	2.131	30.652	Visualize Results
RichFlow-ft-fnl [19]	4.634	2.152	24.886	4.187	1.815	1.377	0.802	2.519	28.780	Visualize Results
HD3-Flow [20]	4.666	2.174	24.994	3.786	1.719	1.647	0.657	2.182	30.579	Visualize Results
LiteFlowNet2 [21]	4.686	2.248	24.571	4.048	1.899	1.473	0.811	2.433	29.375	Visualize Results

(a) Sintel Final

Final Clean

	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
MaskFlownet [2]	2.521	0.989	15.032	2.742	0.908	0.291	0.361	1.285	16.261	Visualize Results
MR-Flow [3]	2.527	0.954	15.365	2.866	0.710	0.420	0.446	1.715	14.826	Visualize Results
ProFlow_ROB [4]	2.709	1.013	16.549	2.843	0.723	0.518	0.485	1.586	16.470	Visualize Results
MaskFlownet-S [5]	2.771	1.077	16.608	2.901	0.996	0.342	0.419	1.404	17.777	Visualize Results
VCN [6]	2.808	1.108	16.682	3.267	0.867	0.418	0.646	1.669	16.302	Visualize Results
ProFlow [7]	2.818	1.027	17.428	2.892	0.751	0.496	0.469	1.626	17.369	Visualize Results
Anonymous872 [8]	2.833	1.093	17.039	3.210	0.875	0.400	0.607	1.607	16.860	Visualize Results
SfM-PM [9]	2.910	1.016	18.357	2.797	0.756	0.479	0.559	1.732	17.431	Visualize Results
FlowFields++ [10]	2.943	0.850	20.027	2.550	0.603	0.403	0.560	1.859	17.401	Visualize Results
GCA-Net [11]	2.947	1.032	18.585	2.900	0.830	0.456	0.602	1.645	17.753	Visualize Results
LiteFlowNet3 [12]	2.994	1.148	18.077	3.000	0.985	0.498	0.559	1.670	18.302	Visualize Results
LiteFlowNet3-S [13]	3.028	1.173	18.182	3.079	0.996	0.527	0.574	1.646	18.566	Visualize Results
FlowFields+ [14]	3.102	0.820	21.718	2.340	0.616	0.373	0.593	1.865	18.549	Visualize Results
DIP-Flow [15]	3.103	0.881	21.227	2.574	0.681	0.419	0.548	1.801	18.979	Visualize Results
PST [16]	3.110	0.942	20.809	2.759	0.664	0.378	0.635	2.069	17.919	Visualize Results
MPIF [17]	3.111	1.134	19.218	3.070	0.939	0.523	0.616	1.980	18.220	Visualize Results
PGM-C [18]	3.234	0.929	22.045	2.724	0.659	0.424	0.567	1.999	19.467	Visualize Results
CPM2 [19]	3.253	0.980	21.812	2.663	0.751	0.416	0.615	1.954	19.503	Visualize Results
L0-norm-Flow2 [20]	3.298	1.069	21.502	2.512	0.826	0.600	0.711	1.940	19.449	Visualize Results

(b) Sintel Clean

Fig. 1: Ranking results on Sintel and KITTI 2015 benchmark.

Evaluation ground truth All pixels Evaluation area All pixels

	Method	Setting	Code	Fl-bg	Fl-fg	Fl-all	Density	Runtime	Environment	Compare
1	UberATG-DRISF			**3.59 %**	10.40 %	**4.73 %**	100.00 %	0.75 s	CPU+GPU @ 2.5 Ghz (Python)	
W. Ma, S. Wang, R. Hu, Y. Xiong and R. Urtasun: Deep Rigid Instance Scene Flow. CVPR 2019.										
2	ACOSF			4.56 %	12.00 %	5.79 %	100.00 %	5 min	1 core @ 3.0 Ghz (Matlab + C/C++)	
3	PCF-F			6.05 %	**5.99 %**	6.04 %	100.00 %	0.08 s	GPU @ 2.5 Ghz (Python)	
4	PPAC-HD3			5.78 %	7.48 %	6.06 %	100.00 %	0.14 s	NVIDIA GTX 1080 Ti	
5	MaskFlownet			5.79 %	7.70 %	6.11 %	100.00 %	0.06 s	NVIDIA TITAN Xp	
6	ISF			5.40 %	10.29 %	6.22 %	100.00 %	10 min	1 core @ 3 Ghz (C/C++)	
A. Behl, O. Jafari, S. Mustikovela, H. Alhaija, C. Rother and A. Geiger: Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios?. International Conference on Computer Vision (ICCV) 2017.										
7	Anonymous 872			5.75 %	8.80 %	6.25 %	100.00 %	0.26 s	1 core @ 2.5 Ghz (Python)	
8	VCN		code	5.83 %	8.66 %	6.30 %	100.00 %	0.18 s	Titan X Pascal	
G. Yang and D. Ramanan: Volumetric Correspondence Networks for Optical Flow. NeurIPS 2019.										
9	Mono expansion			5.83 %	8.66 %	6.30 %	100.00 %	0.25 s	GPU @ 2.5 Ghz (Python)	
10	Stereo expansion			5.83 %	8.66 %	6.30 %	100.00 %	2 s	GPU @ 2.5 Ghz (Python)	
11	NSNR			5.92 %	8.78 %	6.40 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
12	FlowElements		code	6.22 %	8.06 %	6.52 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
13	HD^3-Flow		code	6.05 %	9.02 %	6.55 %	100.00 %	0.10 s	NVIDIA Pascal Titan XP	
Z. Yin, T. Darrell and F. Yu: Hierarchical Discrete Distribution Decomposition for Match Density Estimation. CVPR 2019.										
14	MPSD-ft			6.09 %	9.46 %	6.65 %	100.00 %	0.1 s	1 core @ 2.5 Ghz (Python + C/C++)	
15	PRSM		code	5.33 %	13.40 %	6.68 %	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	
C. Vogel, K. Schindler and S. Roth: 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. ijcv 2015.										
16	RichFlow			6.62 %	6.98 %	6.68 %	100.00 %	0.09 s	NVIDIA P5000	
ERROR: Wrong syntax in BIBTEX file.										
17	MaskFlownet-S			6.53 %	8.21 %	6.81 %	100.00 %	0.03 s	NVIDIA TITAN Xp	
18	ScopeFlow			6.72 %	7.36 %	6.82 %	100.00 %	-1 s	Nvidia GPU	
A. Bar-Haim and L. Wolf: ScopeFlow: Dynamic Scene Scoping for Optical Flow. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2020.										
19	OSF+TC			5.76 %	13.31 %	7.02 %	100.00 %	50 min	1 core @ 2.5 Ghz (C/C++)	
M. Neoral and J. Šochman: Object Scene Flow with Temporal Consistency. 22nd Computer Vision Winter Workshop (CVWW) 2017.										
20	DCTF			7.22 %	6.47 %	7.09 %	100.00 %	tbd	GPU @ 2.5 Ghz (Python + C/C++)	

(c) KITTI 2015 - Supervised

70	intpflow			12.04 %	20.50 %	13.45 %	100.00 %	0.08 s	1 core @ 2.5 Ghz (C/C++)	
71	PR-Sceneflow		code	11.73 %	24.33 %	13.83 %	100.00 %	150 s	4 core @ 3.0 Ghz (Matlab + C/C++)	
C. Vogel, K. Schindler and S. Roth: Piecewise Rigid Scene Flow. ICCV 2013.										
72	Anonymous 872			12.98 %	19.83 %	14.12 %	100.00 %	0.1 s	GPU @ 2.5 Ghz (Python)	
73	MPSD-noft			7.72 %	46.40 %	14.16 %	100.00 %	.1 s	1 core @ 2.5 Ghz (Python + C/C++)	
74	SelFlow			12.68 %	21.74 %	14.19 %	100.00 %	0.09 s	GPU @ 2.5 Ghz (Python)	
P. Liu, M. Lyu, I. King and J. Xu: SelFlow: Self-Supervised Learning of Optical Flow. CVPR 2019.										
75	DDFlow			13.08 %	20.40 %	14.29 %	100.00 %	0.06 s	GPU @ >3.5 Ghz (Python + C/C++)	
P. Liu, I. King, M. Lyu and J. Xu: DDFlow: Learning Optical Flow with Unlabeled Data Distillation. AAAI 2019.										
76	Devon			13.28 %	19.49 %	14.31 %	100.00 %	0.04 s	1 core @ 2.5 Ghz (C/C++)	
77				14.11 %	18.28 %	14.80 %	100.00 %			
78	DCFlow		code	13.10 %	23.70 %	14.86 %	100.00 %	8.6 s	GPU @ 3.0 Ghz (Matlab + C/C++)	
J. Xu, R. Ranftl and V. Koltun: Accurate Optical Flow via Direct Cost Volume Processing. CVPR 2017.										
79	ProFlow			13.86 %	20.91 %	15.04 %	100.00 %	112 s	GPU+CPU @ 3.6 Ghz (Python + C/C++)	
D. Maurer and A. Bruhn: ProFlow: Learning to Predict Optical Flow. BMVC 2018.										
80	FlowNetC-MD			15.98 %	10.96 %	15.14 %	100.00 %	TBD s	1 core @ 2.5 Ghz (C/C++)	

(d) KITTI 2015 - Unsupervised

Fig. 1: Ranking results on Sintel and KITTI 2015 benchmark.

2.2 More Visualization Results

As shown in Fig. 2 and 3, we compare our method with other methods under the supervised settings. We can observe that the flow boundary of the dragon in Fig. 2, which is predicted by VCN+LCV, is better than the other methods and the flow prediction near the tree (in front of the car) and fence by our method in Fig. 3 is more accurate compared with those of the others. The flow prediction for these pixels are are challenging due to the occlusion. LCV explores more information among channel dimensions, which could help alleviate the problem of occlusion to some extent.

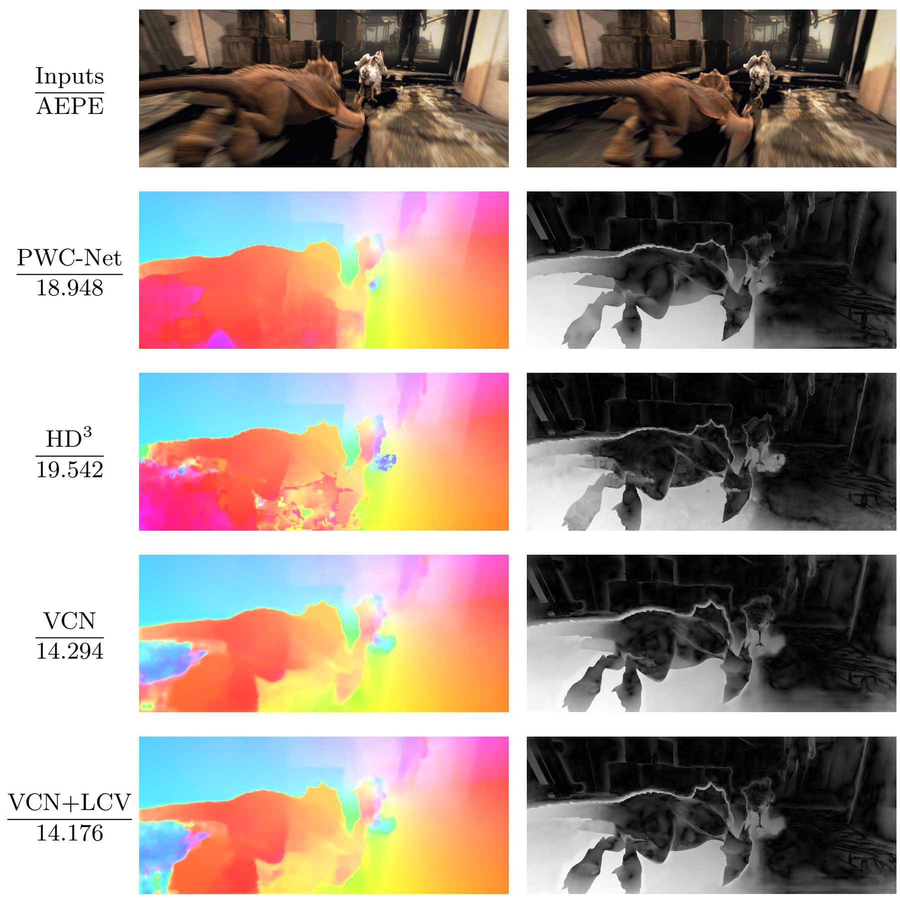


Fig. 2: More visualization results on “Market 4” from the Sintel test final pass. The number under each method name denotes the average end-point error (AEPE) on the given frames. The estimated flow and error maps are presented on the left and right sides, respectively. In the error map, the error of the estimated flow increases from black to white.

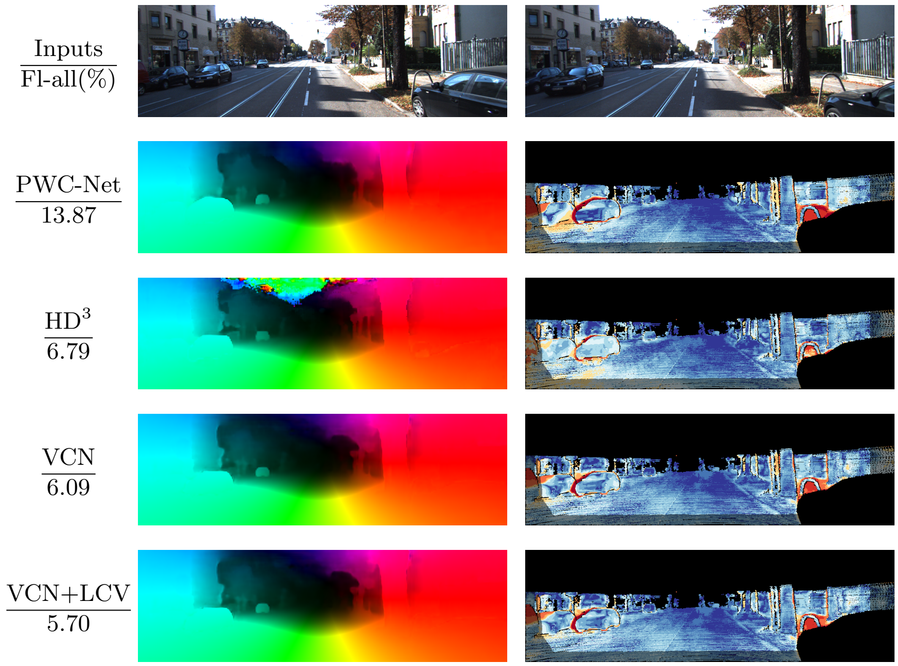


Fig. 3: More visualization results on the KITTI 2015 test set. The number under each method name denotes the Fl-all score on the given frames. The estimated flow and error maps are presented on the left and right sides, respectively. From blue to red, the error of the estimated flow increases in the error map.

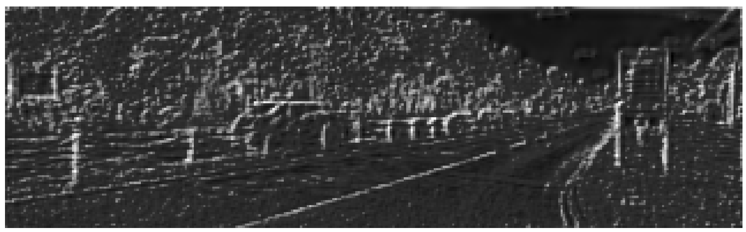
2.3 More Visualization Result on Challenging Cases

We provide three videos showing the effectiveness of our method in three types of challenging cases: 1) illumination change; 2) noise; and 3) adversarial patches. The test videos are from the training set in the KITTI tracking benchmark. We compare the flow results under the normal setting with those under challenging settings. We can observer that the flow results of our model in either one of three challenging cases are temporarily consistent and reasonably good.

2.4 Visualization of the Learned Features

We visualize the feature maps for different eigenvalues in Fig. 4. We find that boundaries of (moving) objects are salient in the feature map corresponding to the max eigenvalue while the min eigenvalue mainly corresponds to background information, which results in more discriminative cost volume and more accurate flow estimation.

(a) frame



(b) feature (max eigenvalue)

(c) feature (min eigenvalue)

Fig. 4: The feature maps corresponding to the largest the smallest eigenvalues.

References

1. Cayley, A.: About the algebraic structure of the orthogonal group and the other classical groups in a field of characteristic zero or a prime characteristic. Reine Angewandte Mathematik **32**, 1846 (1846) 1