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

Abstract. Cost volume is an essential component of recent deep models for optical flow estimation and is usually constructed by calculating the inner product between two feature vectors. However, the standard inner product in the commonly-used cost volume may limit the representation capacity of flow models because it neglects the correlation among different channel dimensions and weighs each dimension equally. To address this issue, we propose a *learnable cost volume* (LCV) using an elliptical inner product, which generalizes the standard inner product by a positive definite kernel matrix. To guarantee its positive definiteness, we perform spectral decomposition on the kernel matrix and re-parameterize it via the Cayley representation. The proposed LCV is a lightweight module and can be easily plugged into existing models to replace the vanilla cost volume. Experimental results show that the LCV module not only improves the accuracy of state-of-the-art models on standard benchmarks, but also promotes their robustness against illumination change, noises, and adversarial perturbations of the input signals.

Keywords: Optical Flow; Cost Volume; Cayley Representation; Inner Product

1 Introduction

Optical flow estimation is a fundamental computer vision task and has broad applications, such as video interpolation [2], video prediction [21], video segmentation [36,6], and action recognition [22]. Despite the recent progress made by deep learning models, it is still challenging to accurately estimate optical flow for image sequences with large displacements, textureless regions, motion blur, occlusion, illumination changes, and non-Lambertian reflection.

Most deep optical flow models [33,23,12] adopt the idea of coarse-to-fine processing via feature pyramids and construct *cost volumes* at different levels of the pyramids. The cost volume stores the costs of matching pixels in the source image with their potential matching candidates in the target image. It is typically

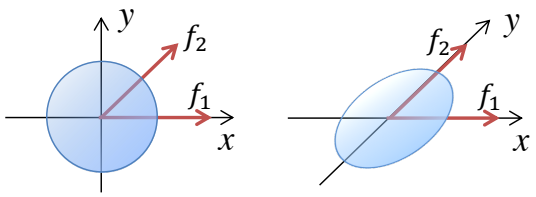


Fig. 1: Standard inner product space v.s. elliptical inner product space.

constructed by calculating the inner product between the convolutional features of one frame and those of the next frame, and then regressed to the estimated optical flow by an estimation sub-network. The accuracy of the estimated optical flow heavily relies on the quality of the constructed cost volume.

While the standard Euclidean inner product is widely used to build the cost volume (a.k.a., vanilla cost volume) for optical flow, we argue that it limits the representation capacity of the flow model for two reasons. First, the correlation among different channel dimensions is not taken into consideration by the standard Euclidean inner product. As shown in Fig. 1, we use a simple 2D example for illustration. Given two feature vectors $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ with positive correlation in the standard inner product space, we are able to find a proper elliptical inner product space to make these two feature vectors orthogonal to each other, which gives a zero correlation. Therefore, the specific choice of the inner product space influences the values of the matching costs, and thus should be further exploited. Second, each feature dimension contributes equally to the vanilla cost volume, which may give a sub-optimal solution to constructing the cost volume for flow estimation. Ideally, dimensions corresponding to noises and random perturbations should be suppressed, while those containing discriminative signals for flow estimation should be kept or magnified.

To address these limitations, we propose a *learnable cost volume* (LCV) module which accounts for the correlation among different channel dimensions and re-weighs the contribution of each feature channel to the cost volume. The LCV generalizes the Euclidean inner product space to an elliptical inner product space, which is parameterized by a symmetric and positive definite kernel matrix. The spectral decomposition of the kernel matrix gives an orthogonal matrix and a diagonal matrix. The orthogonal matrix linearly transforms the features into a new feature space, which accounts for the correlation among different channel dimensions. The diagonal matrix multiplies each transformed feature by a positive scalar, which weighs each feature dimension differently. From a geometric perspective, the orthogonal matrix rotates the axes and the diagonal matrix stretches the axes so that the feature vectors are represented in a learned elliptical inner product space, which generates more discriminative matching costs for flow estimation.

However, directly learning a kernel matrix in an end-to-end manner cannot guarantee the symmetry and positive definiteness of the kernel matrix, which is required by the definition of inner product. To address this issue, we perform

spectral decomposition on the kernel matrix and represent each component via the Cayley transform. Specifically, the special orthogonal matrices that exclude -1 as the eigenvalue can be bijectively mapped into the skew-symmetric matrices, and the diagonal matrices can be similarly represented by the composition of the Cayley transform and the arctangent function. In this way, all parameters of the learnable cost volume can be inferred in an end-to-end fashion without explicitly imposing any constraints.

The proposed learnable cost volume is a general version of the vanilla cost volume, and thus can replace the vanilla cost volume in the existing networks. We finetune the existing architectures equipped with LCV by initializing the kernel matrix as the identity matrix and restoring other parameters from the pre-trained models. Experimental results on the Sintel and KITTI benchmark datasets show that the proposed LCV significantly improves the performance of existing methods in both supervised and unsupervised settings. In addition, we demonstrate that LCV is able to promote the robustness of the existing models against illumination changes, noises, and adversarial attacks.

To summarize, we make the following contributions:

1. We propose a learnable cost volume (LCV) to account for correlations among different feature dimensions and weight each dimension separately.
2. We employ the Cayley representation to re-parameterize the kernel matrix in a way that all parameters can be learned in an end-to-end manner.
3. The proposed LCV can easily replace the vanilla cost volume and improve the accuracy and robustness of the state-of-the-art models.

2 Related Work

Supervised Learning of Optical Flow. Inspired by the success of convolutional neural networks (CNNs) on per-pixel predictions such as semantic segmentation and single-image depth estimation, Dosovitski *et al.* propose FlowNet [8], the first end-to-end deep neural network capable of learning optical flow. FlowNet predicts a dense optical flow map from two consecutive image frames with an encoder-decoder architecture. FlowNet2.0 [15] extends FlowNet by stacking multiple basic FlowNet modules for iterative refinement and its accuracy is fully on par with those of the state-of-the-art methods at the time. Motivated by the idea of coarse-to-fine refinement in traditional optical flow methods, SpyNet [29] introduces a compact spatial pyramid network that warps images at multiple scales to deal with displacements caused by large motions. PWC-Net [33] extracts feature through pyramidal processing and builds a cost volume at each level from the warped and the target features to iteratively refine the estimated flow. VCN [39] improves the cost volume processing by decoupling the 4D convolution into a 2D spatial filter and a 2D winner-take-all (WTA) filter, while still retaining a large receptive field. HD^3 [40] learns a probabilistic matching density distribution at each scale and merges the matching densities at different scales to recover the global matching density.

Unsupervised Learning of Optical Flow. The advantage of unsupervised methods is that it can sidestep the limitations of the synthetic datasets and exploit the large number of training data in the realistic domain. In [17] and [31], the flow guidance comes from warping the target image according to the predicted flow and comparing against the reference image. The photometric loss is adopted to ensure brightness constancy and spatial smoothness. In some work [37,25], occluded regions are excluded from the photometric loss. As pixels occluded in the target image are also absent in the warped one, enforcing matching of the occluded pixels would misguide the training. Wang *et al.* [37] obtain an occlusion mask from the range map inferred from the backward flow, while UnFlow [25] relies on the forward-backward consistency to estimate the occlusion mask. Unlike these two methods that predict the occlusion map in advance with certain heuristic, Back2Future [16] estimates the occlusion and optical flow jointly by introducing a multi-frame formulation and reasoning the occlusion in a more advanced manner. DDFlow [23] performs knowledge distillation by cropping patches from the unlabeled images, which provides flow guidance for the occluded regions. SelFlow [24] hallucinates synthetic occlusions by perturbing super-pixels where the occluded regions are guided by a model pre-trained from non-occluded regions.

Correspondence Matching. Typically, stereo matching algorithms [32,11] involve local correspondence extraction and smoothness regularization, where the smoothness regularization is enforced by energy minimization. Recently, handcrafted features are replaced by deep features and minimization of the matching cost is substituted by training convolutional neural networks [42,19]. Xu *et al.* [38] construct a 4D cost volume using an adaptation of the semi-global matching, and Yang *et al.* [39] reduce the computation overhead of processing the 4D matching volume by factorizing into two separable filters.

Different from these approaches where the correspondence is represented by a hand-crafted matching cost volume, we propose a learnable cost volume that can capture the correlation among different channels by adapting the features to an elliptical inner product space. Such a correlation is automatically learned by optimizing the kernel matrix using the Cayley representation, which is more flexible and effective in optical flow estimation and can be easily plugged into the existing architectures. To our knowledge, this paper is the first one to use the Cayley representation for learning correspondence in optical flow.

3 Learnable Correlation Volume

3.1 Vanilla Cost Volume

Let $\boldsymbol{F}^1, \boldsymbol{F}^2 \in \mathbb{R}^{c \times h \times w}$ be the convolutional feature of the first frame and the warped feature of the second frame, respectively. The vanilla cost volume is defined as the inner product between the query feature $\boldsymbol{F}^1_{i,j}$ and the potential

match candidate $\boldsymbol{F}^2_{k',l'}$, *i.e.,*

$$C(\boldsymbol{F}^1, \boldsymbol{F}^2)_{k,l,i,j} = \boldsymbol{F}^{1\top}_{i,j} \boldsymbol{F}^2_{k',l'}, \tag{1}$$

which maps from the space $\mathbb{R}^{c\times h\times w} \times \mathbb{R}^{c\times h\times w}$ to $\mathbb{R}^{u\times v\times h\times w}$. Here, u and v are usually odd numbers, indicating the displacement ranges in horizontal and vertical directions, (i, j) denotes the spatial location of the feature map $\boldsymbol{F}^1$, and $(k', l') = (i-(u-1)/2+k, j-(v-1)/2+l)$ denotes that of $\boldsymbol{F}^2$. For each location (i, j) of the query feature $\boldsymbol{F}^1$, the matching is performed against pixels of $\boldsymbol{F}^2$ within a $u\times v$ search window centered by the location (i, j). Then, the cost volume is either reshaped into $uv \times h \times w$ and post-processed by 2D convolutions [33], or kept as a 4D tensor on which the separable 4D convolutions [39] are applied.

3.2 Learnable Cost Volume

We generalize the standard Euclidean inner product to the elliptical inner product, where the matching cost is computed as follows:

$$C(\boldsymbol{F}^1, \boldsymbol{F}^2)_{k,l,i,j} = \boldsymbol{F}^{1\top}_{i,j} \boldsymbol{W} \boldsymbol{F}^2_{k',l'}. \tag{2}$$

Here, $\boldsymbol{W} \in \mathbb{R}^{c\times c}$ is a learnable kernel matrix that determines the elliptical inner product space, and other notations are the same as those in Eq. (1). According to the definition of inner product, $\boldsymbol{W}$ should be a symmetric and positive definite matrix. By spectral decomposition, we obtain

$$\boldsymbol{W} = \boldsymbol{P}^\top \boldsymbol{\Lambda} \boldsymbol{P}, \tag{3}$$

where $\boldsymbol{P}$ is an orthogonal matrix, and $\boldsymbol{\Lambda}$ is a diagonal matrix with positive entries, *i.e.,* $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_c)$ with $\lambda_i > 0$, $\forall i \in \{1, \cdots, c\}$. The orthogonal matrix $\boldsymbol{P}$ actually rotates the coordinate axes and the diagonal matrix $\boldsymbol{\Lambda}$ re-weights different dimensions, which directly address the two limitations mentioned in Sec. 1.

3.3 Learning with the Cayley Representation

In the proposed LCV module, the entries of the kernel matrix $\boldsymbol{W}$ are the only learnable parameters. However, the constraints of symmetry and positive-definiteness hinders the gradient-based end-to-end learning of $\boldsymbol{W}$. To address this issue, we propose to optimize $\boldsymbol{P}$ and $\boldsymbol{\Lambda}$ instead of $\boldsymbol{W}$.

One way to optimize $\boldsymbol{P}$ is to employ the Riemann gradient descent on the Stiefel manifold, which is defined as

$$V_k(\mathbb{R}^n) = \{\boldsymbol{A} \in \mathbb{R}^{n\times k} | \boldsymbol{A}^\top \boldsymbol{A} = \boldsymbol{I}_k\}. \tag{4}$$

All orthogonal matrices lie in the Stiefel manifold. Specifically, $\boldsymbol{P} \in V_c(\mathbb{R}^c)$. Therefore, we can apply the Riemann gradient descent on the Stiefel matrix manifold, where the projection and retraction formula [1] are given by

$$\mathcal{P}_{\boldsymbol{X}}(\boldsymbol{Z}) = (\boldsymbol{I} - \boldsymbol{X}\boldsymbol{X}^\top)\boldsymbol{Z} + \boldsymbol{X} \cdot \mathrm{skew}(\boldsymbol{X}^\top \boldsymbol{Z}) \tag{5}$$

$$\mathcal{R}_{\boldsymbol{X}}(\boldsymbol{Z}) = (\boldsymbol{X} + \boldsymbol{Z})(\boldsymbol{I} + \boldsymbol{Z}^\top \boldsymbol{Z})^{-\frac{1}{2}}, \tag{6}$$

where $\text{skew}(\boldsymbol{X}) := (\boldsymbol{X} - \boldsymbol{X}^\top)/2$. However, to perform the Riemann gradient descent, the projection and retraction operations are required in each training step, and the matrix multiplication brings considerable computational overhead.

We can address this issue in a more elegant way using the Cayley Representation [5]. First, we define a set of matrices:

$$\text{SO}^*(n) := \{\boldsymbol{A} \in \text{SO}(n) : -1 \notin \sigma(\boldsymbol{A})\}, \tag{7}$$

where $\sigma(\boldsymbol{A})$ denotes the spectrum, *i.e.,* all eigenvalues, of $\boldsymbol{A}$. $\text{SO}^*(n)$ is a subset of the special orthogonal group $\text{SO}(n)$ and the spectrum of its elements excludes -1. Then, we have the following theorems:

Theorem 1 (Cayley Representation) *Given any matrix $\boldsymbol{P} \in \text{SO}^*(n)$, there exists a unique skew-symmetric matrix $\boldsymbol{S}$,* i.e., $\boldsymbol{S}^\top = -\boldsymbol{S}$, *such that*

$$\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{S})(\boldsymbol{I} + \boldsymbol{S})^{-1}. \tag{8}$$

Theorem 2 *The set of matrices $\text{SO}^*(n)$ is connected.*

By Theorem 1, we can initialize the matrix $\boldsymbol{P}$ in Eq. (3) as an identity matrix $\boldsymbol{I} \in \text{SO}^*(c)$, and update $\boldsymbol{S}$ so as to update $\boldsymbol{P}$ using gradient-based optimizer. Let $\boldsymbol{P}^*$ be the optimal orthogonal matrix, and we claim that it is possible to reach $\boldsymbol{P}^*$ from initializing as the identity matrix $\boldsymbol{P} = \boldsymbol{I}$. This because $\text{SO}^*(c)$ is a connected set (Theorem 2), so there exists a continuous path joining $\boldsymbol{I} \in \text{SO}^*(c)$ and any $\boldsymbol{P} \in \text{SO}^*(c)$, including $\boldsymbol{P}^*$.

Due to the positive definiteness of $\boldsymbol{W}$, the constraint of the diagonal matrix $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_c)$ is $\lambda_i > 0$, $\forall i = 1, \ldots, c$. Thus, we map $\mathbb{R}$ to $\mathbb{R}^+$ by applying the composition of the Cayley transform and the arctangent function, *i.e.,*

$$\lambda_i = \frac{\pi + 2\arctan t_i}{\pi - 2\arctan t_i}, \tag{9}$$

where $t_i \in \mathbb{R}$ is free of constraint.

The above re-parameterization trick enables us to update the kernel matrix $\boldsymbol{W}$ in an end-to-end manner using the SGD optimizer or its variants, which alleviates the heavy computation brought by the projection and retraction and makes the training process much easier.

3.4 Interpretation

To better understand the learnable cost volume, we analyze several cases here.

1. $\boldsymbol{W} = \boldsymbol{I}$. This degenerates into the vanilla cost volume, in which the standard Euclidean inner product is adopted.

2. $\boldsymbol{W = \Sigma^{-1}}$. Let $\boldsymbol{\Sigma}$ be the covariance matrix, *i.e.,* Gram matrix, of the convolutional feature, then the learnable cost volume is essentially a whitening transformation. Let $\boldsymbol{Q} = \boldsymbol{\Lambda}^{1/2}\boldsymbol{P}$, and then Eq. (2) can be formulated as

$$C(\boldsymbol{F}^1, \boldsymbol{F}^2)_{k,l,i,j} = \boldsymbol{F}^{1\top}_{i,j}\boldsymbol{P}^\top\boldsymbol{\Lambda}^{1/2}\boldsymbol{\Lambda}^{1/2}\boldsymbol{P}\boldsymbol{F}^2_{k',l'} = (\boldsymbol{Q}\boldsymbol{F}^1_{i,j})^\top(\boldsymbol{Q}\boldsymbol{F}^2_{k',l'}), \tag{10}$$

where $\boldsymbol{Q}\boldsymbol{F}^1_{i,j}$ represents the transformed feature of $\boldsymbol{F}^1_{i,j}$ after PCA [18] whitening. Similarly, letting $\boldsymbol{R} = \boldsymbol{P}^\top\boldsymbol{\Lambda}^{1/2}\boldsymbol{P}$, we can have

$$C(\boldsymbol{F}^1, \boldsymbol{F}^2)_{k,l,i,j} = \boldsymbol{F}^{1\top}_{i,j}\boldsymbol{P}^\top\boldsymbol{\Lambda}^{1/2}\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{\Lambda}^{1/2}\boldsymbol{P}\boldsymbol{F}^2_{k',l'} = (\boldsymbol{R}\boldsymbol{F}^1_{i,j})^\top(\boldsymbol{R}\boldsymbol{F}^2_{k',l'}), \tag{11}$$

where $\boldsymbol{R}\boldsymbol{F}^1_{i,j}$ is the transformed feature of $\boldsymbol{F}^1_{i,j}$ after ZCA [3] whitening. It has been shown that the high-level styles can be removed with the contextual structures remained by whitening the convolutional features [20].

3. $\boldsymbol{W = P^\top\Lambda P}$. The learnable cost volume shares a similar formula as the whitening process, but $\boldsymbol{W}$ is learned over the whole training dataset rather than statistics of two inputs, thus contains certain holistic information of the entire training dataset. Because it has been verified that the certain holistic characteristics of the underlying image can be captured by the Gram matrix along the channel dimension [9,20]. The learnable cost volume performs as "whitening" features using the common information learned from all frames. Specifically, the orthogonal matrix $\boldsymbol{P}$ re-arranges the information across the channel dimension, while the diagonal matrix $\boldsymbol{\Lambda}$ filters out insignificant signals, making the correlation more robust to the illumination changes and noises. (See Sec. 4.4.)

It should also be pointed out that the whitening matrix $\boldsymbol{R}$ in Eq. (11) could be viewed as a 1×1 conv functioning on the feature, but directly applying a 1×1 conv with learnable parameters on features before computing the standard cost volume cannot replace the proposed learned cost volume. Because $\boldsymbol{R}^\top\boldsymbol{R}$ only gives a positive semi-definite matrix even when $\boldsymbol{R}$ is full-rank, which does not meet the positive definiteness property of an inner product.

3.5 Relation with the Weighted Sum of Squared Difference

The learnable cost volume can be also formulated by re-thinking the simplest matching criterion for comparing two features, *i.e.,* the weighted sum of squared difference (WSSD):

$$\sum_i \lambda_i \left(G_i(\boldsymbol{F}^2) - G_i(\boldsymbol{F}^1)\right)^2, \tag{12}$$

where $G : \mathbb{R}^c \to \mathbb{R}^c$ denotes a transformation function on the features $\boldsymbol{F}^i \in \mathbb{R}^c$, $i = 1, 2$, and $G_i(\boldsymbol{F})$ indicates the i^{th} element of $G(\boldsymbol{F})$. By the Taylor series expansion, we have

$$\sum_i \lambda_i \left(G_i(\boldsymbol{F}^2) - G_i(\boldsymbol{F}^1)\right)^2 \approx \sum_i \lambda_i \left(\nabla G_i(\boldsymbol{F}^1)^\top \Delta\boldsymbol{F}\right)^2 = \Delta\boldsymbol{F}^\top\boldsymbol{W}\Delta\boldsymbol{F}, \tag{13}$$

where $\Delta \boldsymbol{F}=\boldsymbol{F}^2-\boldsymbol{F}^1$ is the feature difference and $\boldsymbol{W}=\sum_i \lambda_i \nabla G_i(\boldsymbol{F}^1)\nabla G_i(\boldsymbol{F}^1)^\top$ is the auto-correlation matrix. Here, $\boldsymbol{W}$ coincides with the kernel matrix of the proposed LCV module in Eq. (2). When $\lambda_i = 1(i = 1, \ldots, c)$ and G is an identity map, then $\boldsymbol{W} = \boldsymbol{I}$, which corresponds to the vanilla cost volume. If we further expand Eq. (13), we can see the connection with the proposed learnable correlation volume as follows:

$$\begin{aligned} \Delta \boldsymbol{F}^\top \boldsymbol{W} \Delta \boldsymbol{F} &= (\boldsymbol{F}^2 - \boldsymbol{F}^1)^\top \boldsymbol{W} (\boldsymbol{F}^2 - \boldsymbol{F}^1) \\ &= (\boldsymbol{F}^{2\top} \boldsymbol{W} \boldsymbol{F}^2 + \boldsymbol{F}^{1\top} \boldsymbol{W} \boldsymbol{F}^1) - 2\boldsymbol{F}^{1\top} \boldsymbol{W} \boldsymbol{F}^2, \end{aligned} \tag{14}$$

where the last term shares the same formula with the proposed learnable cost volume. This implies that the proposed learnable cost volume is inversely correlated with WSSD. As WSSD measures the discrepancy between two features, the learnable cost volume characterizes a certain kind of similarity between them.

4 Experiments

In this section, we present the experimental results of optical flow estimation in both supervised and unsupervised settings to demonstrate the effectiveness of the proposed learnable cost volume. Also, we carry out ablation studies to show that the LCV module performs favorably against other counterparts. Moreover, we analyze the behavior of LCV and find it beneficial to handling three challenging cases. More results can be found in the supplementary material and the source code and trained models will be made available to the public.

Training Process. It is well-known that the deep optical flow estimation pipeline consists the following stages in the supervised settings [34]: 1) train the model on the FlyingChairs [7] dataset; 2) finetune the model on the FlyingThings3D [28] dataset; and 3) finetune the model on the Sintel [4] and KITTI [27,26] training sets. Besides, there are lots of tricks such as data augmentation and learning rate disruption, making the training process more complicated.

To avoid the tedious training procedure over multiple datasets, we adopt a more efficient way to train the model equipped with LCV. As mentioned in Sec. 3.4, the vanilla cost volume is a special case of the learnable cost volume when $\boldsymbol{W} = \boldsymbol{I}$, which means that the learnable cost volume is more general and backward compatible with vanilla cost volume. Therefore, we initialize the kernel matrix $\boldsymbol{W}$ as the identity matrix and other parameters are directly restored from the pre-trained models without using LCV. After that, we finetune the model with LCV on the Sintel or KITTI datasets using the same loss function. This training process not only significantly reduces training time but also plays a crucial role in the success under the unsupervised settings. (See Sec. 4.2.) This approach can also be viewed as fixing the kernal matrix as $\boldsymbol{W} = \boldsymbol{I}$ in the first three training stages, and let $\boldsymbol{W}$ be learnable in the final stage.

Table 1: Results of the supervised methods on the MPI Sintel and KITTI 2015 optical flow benchmarks. All reported numbers indicate the average endpoint error (AEPE) except for the last two columns, where the percentage of outliers averaged over all groundtruth pixels (Fl-all) are presented. "-ft" means finetuning on the relative MPI Sintel or KITTI training set and the numbers in the parenthesis are results that train and test on the same dataset. Missing entries (-) indicate that the results are not reported for the respective method. The best result for each metric is printed in bold.

Methods	Sintel				KITTI 2015		
	Clean		Final		AEPE	Fl-all (%)	
	train	test	train	test	train	train	test
FlowNet2 [15]	2.02	3.96	3.14	6.02	10.06	30.37	-
FlowNet2-ft [15]	(1.45)	4.16	(2.01)	5.74	(2.30)	(8.61)	10.41
DCFlow [38]	-	3.54	-	5.12	-	15.09	14.83
MirrorFlow [13]	-	-	-	6.07	-	9.93	10.29
SpyNet [29]	4.12	6.69	5.57	8.43	-	-	-
SpyNet-ft [29]	(3.17)	6.64	(4.32)	8.36	-	-	35.07
LiteFlowNet [12]	2.52	-	4.05	10.39	-	-	-
LiteFlowNet+ft [12]	(1.64)	4.86	(2.23)	6.09	(2.16)	-	10.24
PWC-Net [33]	2.55	-	3.93	-	10.35	33.67	-
PWC-Net-ft [33]	(2.02)	4.39	(2.08)	5.04	(2.16)	(9.80)	9.60
PWC-Net+-ft [34]	(1.71)	3.45	(2.34)	4.60	(1.50)	(5.30)	7.72
IRR-PWC-ft [14]	(1.92)	3.84	(2.51)	4.58	(1.63)	(5.30)	7.65
HD^3 [40]	3.84	-	8.77	-	13.17	23.99	-
HD^3-ft [40]	(1.70)	4.79	(1.17)	4.67	(1.31)	(4.10)	6.55
VCN [39]	2.21	-	3.62	-	8.36	25.10	8.73
VCN-ft [39]	(1.66)	2.81	(2.24)	4.40	(1.16)	(4.10)	6.30
RAFT [35]	1.09	2.77	1.53	3.61	(1.07)	(3.92)	6.30
RAFT (warm start) [35]	1.10	**2.42**	1.61	3.39	-	-	-
VCN+LCV	(1.62)	2.83	(2.22)	4.20	(1.13)	(3.80)	**6.25**
RAFT+LCV	**(0.94)**	2.75	**(1.31)**	3.55	**(1.06)**	**(3.77)**	6.26
RAFT+LCV (warm start)	(0.99)	2.49	(1.47)	**3.37**	-	-	-

4.1 Supervised Optical Flow Estimation

First, we incorporate the learnable cost volume in the VCN [39] and RAFT [35] framework, and compare them with other existing methods. As shown in Table 1, our method performs favorably against other state-of-the-art methods on the Sintel Clean/Final pass and the KITTI 2015 benchmark.

The proposed LCV module improves the performance of VCN and RAFT by transforming the features of video frames to a whitened space to obtain a clean and robust matching correlation. This could account for the performance improvement on the Sintel Final pass, where the scenarios are much harder. As shown in Fig. 2, the flow estimation error for the snow background at the right side is smaller than other methods. This is a challenging case because the front person's arm renders occlusion to part of the snow background and the

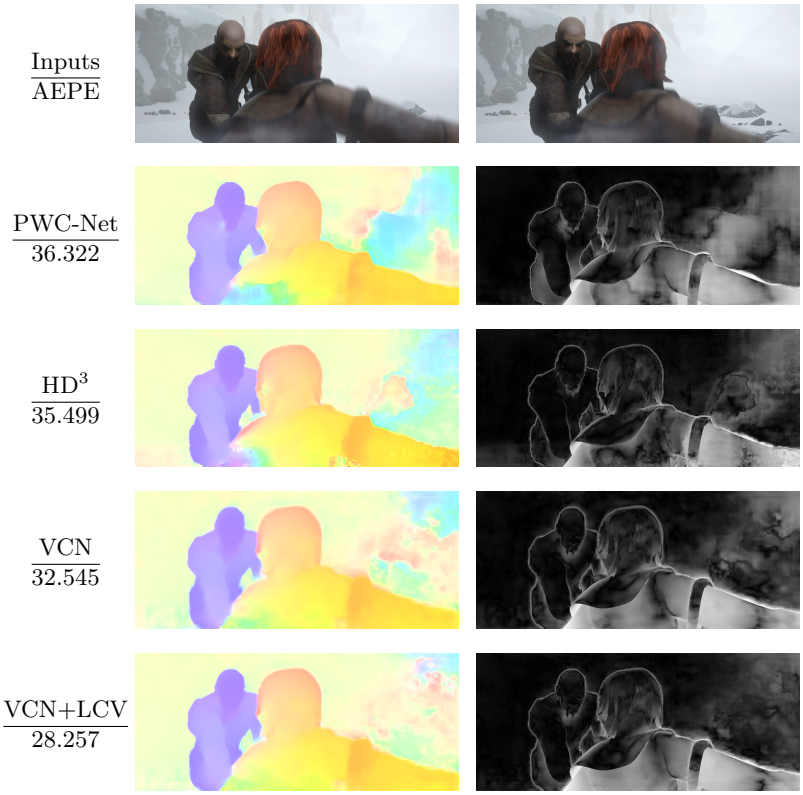


Fig. 2: Visual results on "Ambush 1" from the Sintel test final pass. The number under each method denotes the average end-point error (AEPE). Left: estimated flow; right: error map (increases from black to white).

background is nearly all white, providing few clues for matching. However, the LCV module exploits more information from the correlation among different channels, which assists in obtaining the coherent flow estimation in the snow background. The LCV module also has an edge over the vanilla cost volume under the circumstance of light reflection and occlusion. As shown in Fig. 3, the prediction error of our method is smaller around the light reflection region and the rightmost traffic sign.

Although we do not report the model parameters in the table, the proposed LCV module only makes a very slight increase in the model size. The additional parameters come from the kernel matrices $\boldsymbol{W} \in \mathbb{R}^{c \times c}$ at different pyramid levels. Taking VCN+LCV as an example, there are five kernel matrices in total, whose channel dimensions are 64, 64, 128, 128, and 128, respectively. The LCV module only takes up $64^2 \times 2 + 128^2 \times 3 = 57,344$ parameters, which is negligible compared with the entire VCN model of around 6.23M parameters.

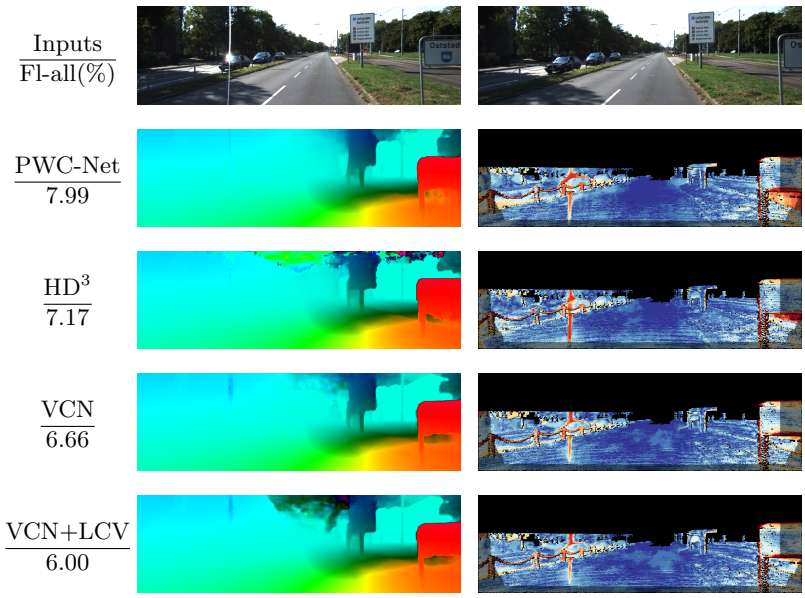


Fig. 3: Visual results on the KITTI 2015 test set. The number under each method name denotes the Fl-all score on the given frames. Left: estimated flow; right: error map (increases from blue to red).

4.2 Unsupervised Optical Flow Estimation

We also test the LCV module in unsupervised settings on the KITTI 2015 benchmark. We replace the vanilla cost volume with the LCV module in the DDFlow [23] model, and compare it with other unsupervised methods. As shown in Table 2, our model outperforms the DDFlow baseline, and even performs favorably against SelFlow [24], an improved version of DDFlow.

The training process is crucial to the success of the LCV module in the unsupervised methods. Different from the supervised training of optical flow models, there is no ground truth for direct supervision. Instead, most unsupervised methods use the photometric loss as a proxy loss. Specifically, the training of DDFlow consists of two stages: 1) pre-train a non-occlusion model with census transform [10], and 2) train an occlusion model by distillation from the non-occlusion model. If we directly follow the same procedure, the training of DDFlow+LCV will run into trivial solutions, as the photometric loss does not give a strong supervision for the correspondence learning, especially when the LCV module increases the dimension of the solution space. To prevent from trivial solutions, we fix the kenrel matrix as $\boldsymbol{W} = \boldsymbol{I}$ in the pre-train stages, and update $\boldsymbol{W}$ in the distillation stage.

Table 2: Results of the unsupervised methods on the KITTI 2015 optical flow benchmark. Missing entries (-) indicate that the results are not reported for the respective method. The best result for each metric is printed in bold.

Methods	KITTI 2015			
	train	test		
	AEPE	Fl-bg (%)	Fl-fg (%)	Fl-all (%)
DSTFlow [31]	16.79	-	-	39
GeoNet [41]	10.81	-	-	-
UnFlow [25]	8.88	-	-	28.95
DF-Net [43]	7.45	-	-	22.82
OccAwareFlow [37]	8.88	-	-	31.20
Back2FutureFlow [16]	6.59	22.67	24.27	22.94
SelFlow [24]	**4.84**	**12.68**	21.74	14.19
DDFlow [23]	5.72	13.08	20.40	14.29
DDFlow+LCV (Ours)	5.15	12.98	**19.83**	**14.12**

Table 3: Ablation study of different variants of VCN on the KITTI 2015 dataset.

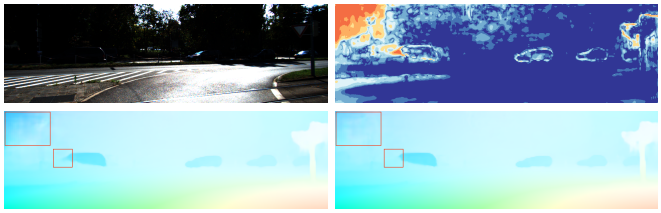
Methods	VCN	VCN (ct)	VCN ($\boldsymbol{W}$, ct)	VCN ($\boldsymbol{\Lambda}$, ct)	VCN ($\boldsymbol{P}$, ct)	VCN(1x1 conv)	VCN + LCV
AEPE/Fl-all	3.9/1.144	4.2/1.204	4.1/1.193	3.8/1.136	3.9/1.129	3.9/1.163	3.8/1.132

4.3 Ablation Study

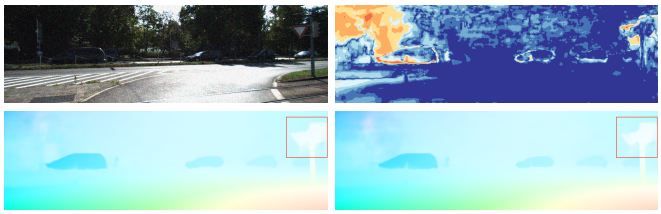
We evaluate multiple variants of the LCV module based on the VCN baseline:

- VCN: the original VCN baseline.
- VCN (ct): continue training the existing VCN using a small learning rate for more epochs.
- VCN ($\boldsymbol{W}$, ct): remove the symmetry and positive definiteness constraint of $\boldsymbol{W}$, *i.e.,* , not using the Cayley representation. We restore the weights from the pre-trained VCN and continue training the model with free $\boldsymbol{W}$.
- VCN ($\boldsymbol{\Lambda}$, ct): fix $\boldsymbol{P}$ to be an identity matrix and make the diagonal matrix $\boldsymbol{\Lambda}$ learnable.
- VCN ($\boldsymbol{P}$, ct): fix $\boldsymbol{\Lambda}$ to be an identity matrix and make the orthogonal matrix $\boldsymbol{P}$ learnable.
- VCN (1x1 conv): replace the positive definite $\boldsymbol{W}$ with $\boldsymbol{R}^\top \boldsymbol{R}$, where $\boldsymbol{R}$ is a 1×1 conv operating on features with input and output dimensions equal. $\boldsymbol{R}^\top \boldsymbol{R}$ is only a positive semi-definite matrix.
- VCN+LCV: employ the Cayley representation to ensure the symmetry and positive definiteness of $\boldsymbol{W}$.

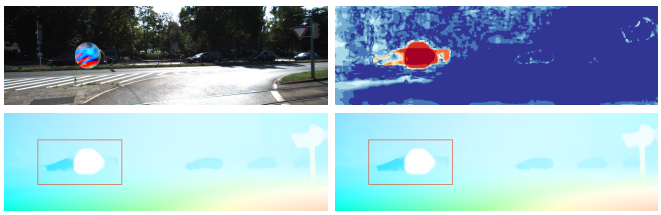
We randomly split the 200 images with ground truth from the KITTI 2015 training set into the training and validation set by a ratio of 4:1. As shown in Table 3, we report the AEPE/Fl-all scores on the validation set. We observe that continuing training of the VCN model does not bring any benefit, which



(a) Illumination change ($\gamma = 0.5$)



(b) Noise (std=0.001)



(c) Adversarial patch (radius=50)

Fig. 4: Visual results of three challenging cases, *i.e.,* illumination change, noise, and adversarial patch. Top left: the first input frame; bottom left/right: flow by VCN / VCN+LCV; top right: flow difference between two methods.

indicates that the best VCN model is not obtained at the very end of the training. Another interesting observation is that VCN ($\boldsymbol{W}$, ct) performs better than VCN (ct), showing the benefit of increasing the model capacity. However, it does not outperform VCN, not even VCN+LCV, confirming the importance of using a valid inner product space. Comparing the result of VCN (1x1 conv), we can further conclude that ensuring the positive definiteness via the Cayley representation is crucial to the performance. We can also find that VCN ($\boldsymbol{\Lambda}$, ct) gets a lower AEPE and VCN ($\boldsymbol{P}$, ct) gets a lower Fl-all compared with vanilla VCN. VCN+LCV combines the advantages of both axis rotation and re-weighting, aiming to address two limitations mentioned in the paper.

4.4 Robustness Analysis

To further understand the effect of the LCV module, we evaluate the flow estimation performance under three challenging cases, *i.e.,* 1) illumination changes:

Table 4: Results on three challenging cases (numbers: AEPE/Fl-all scores).

(a) Illumination change

γ	0.2	0.3	0.4	0.5	0.7	1.0	2.0	3.0
VCN	16.8/3.240	9.9/1.891	5.9/1.306	3.8/0.995	2.7/0.834	2.5/0.805	2.6/0.819	2.6/0.826
VCN+LCV	17.1/3.232	9.8/1.866	5.9/1.273	3.7/0.967	2.6/0.804	2.4/0.775	2.4/0.790	2.5/0.804

(b) Noise

Standard deviation	0.0001	0.001	0.01	0.1
VCN	2.6/0.816	2.9/0.868	5.0/1.157	19.6/3.213
VCN+LCV	2.4/0.785	2.7/0.838	4.7/1.107	18.9/3.043

(c) Adversarial patch

Patch size	50	100	150	200
VCN	3.5/0.981	5.6/1.419	8.5/2.048	11.9/2.880
VCN+LCV	3.4/0.949	5.5/1.384	8.3/2.004	11.6/2.801

we adjust the illumination of the input frames by changing the value of γ, where $\gamma = 1.0$ is the original image, $\gamma < 1.0$ is for a darker image, and $\gamma > 1.0$ is for a brighter image. 2) adding noises: we adjust the standard deviation to control the noise magnitude. and 3) inserting adversarial patches: we borrow the universal adversarial patch [30] that can perform a black-box attack for all optical flow models, and insert patches of different sizes to the input frames.

We compare the VCN model and its variant equipped with LCV. Both two models are trained on the KITTI 2015 training set. For qualitative comparison, we perform the above three types of processing on 194 images with the flow groundtruth from the KITTI 2012 as our test set. As shown in Table 4(a), VCN+LCV consistently outperforms the VCN baseline in all three challenging cases. For better illustration, we visualize the effect on an image from KITTI 2015 test set as shown in Fig. 4. It can be seen that the LCV module can help stabilize the flow prediction around the background trees at the top left corner of the frame under the cases of dark illumination and random noise injection. In the third example, the outline of the car body near the patch circle is better preserved by our model. (See the difference map for details.)

5 Conclusions

In this work, we introduce a learnable cost volume (LCV) module for optical flow estimation. The proposed LCV module generalizes the standard Euclidean inner product into an elliptical inner product with a symmetric and positive definite kernel matrix. To keep its symmetry and positive definiteness, we use the Cayley representation to re-parameterize the kernel matrix for end-to-end training. The proposed LCV is a lightweight module and can be easily plugged into any existing networks to replace the vanilla cost volume. Experimental results show that the proposed LCV module improves both the accuracy and the robustness of state-of-the-art optical flow models.

Acknowledgement. This work is supported in part by NSF CAREER Grant 1149783. We also thank Pengpeng Liu and Jingfeng Wu for kind help.

References

1. Absil, P.A., Mahony, R., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2009) 5
2. Bao, W., Lai, W.S., Ma, C., Zhang, X., Gao, Z., Yang, M.H.: Depth-aware video frame interpolation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 1
3. Brox, T., Malik, J.: Large displacement optical flow: descriptor matching in variational motion estimation. IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI) **33**(3), 500–513 (2010) 7
4. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: European Conference on Computer Vision (ECCV) (2012) 8
5. 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) 6
6. Cheng, J., Tsai, Y.H., Wang, S., Yang, M.H.: Segflow: Joint learning for video object segmentation and optical flow. In: IEEE International Conference on Computer Vision (ICCV) (2017) 1
7. Dosovitskiy, A., Fischer, P., Ilg, E., Häusser, P., Hazırbaş, C., Golkov, V., v.d. Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: IEEE International Conference on Computer Vision (ICCV) (2015) 8
8. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015) 3
9. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 7
10. Hafner, D., Demetz, O., Weickert, J.: Why is the census transform good for robust optic flow computation? In: International Conference on Scale Space and Variational Methods in Computer Vision (SSVM). Springer (2013) 11
11. Horn, B.K., Schunck, B.G.: Determining optical flow. Artificial intelligence **17**(1-3), 185–203 (1981) 4
12. Hui, T.W., Tang, X., Change Loy, C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1, 9
13. Hur, J., Roth, S.: Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. IEEE International Conference on Computer Vision (ICCV) pp. 312–321 (2017) 9
14. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 9
15. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3, 9
16. Janai, J., Güney, F., Ranjan, A., Black, M.J., Geiger, A.: Unsupervised learning of multi-frame optical flow with occlusions. In: European Conference on Computer Vision (ECCV) (2018) 4, 12

17. Jason, J.Y., Harley, A.W., Derpanis, K.G.: Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In: European Conference on Computer Vision (ECCV). Springer (2016) 4
18. Jolliffe, I.T.: Principal components in regression analysis. In: Principal component analysis, pp. 129–155. Springer (1986) 7
19. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. In: IEEE International Conference on Computer Vision (ICCV) (2017) 4
20. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Universal style transfer via feature transforms. In: Neural Information Processing Systems (NeurIPS) (2017) 7
21. Li, Y., Fang, C., Yang, J., Wang, Z., Lu, X., Yang, M.H.: Flow-grounded spatial-temporal video prediction from still images. In: European Conference on Computer Vision (ECCV) (2018) 1
22. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: IEEE International Conference on Computer Vision (ICCV) (2019) 1
23. Liu, P., King, I., Lyu, M.R., Xu, J.: Ddflow: Learning optical flow with unlabeled data distillation. In: Association for the Advancement of Artificial Intelligence (AAAI) (2019) 1, 4, 11, 12
24. Liu, P., Lyu, M.R., King, I., Xu, J.: Selflow: Self-supervised learning of optical flow. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 4, 11, 12
25. Meister, S., Hur, J., Roth, S.: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In: Association for the Advancement of Artificial Intelligence (AAAI) (2017) 4, 12
26. Menze, M., Heipke, C., Geiger, A.: Joint 3d estimation of vehicles and scene flow. In: ISPRS Workshop on Image Sequence Analysis (ISA) (2015) 8
27. Menze, M., Heipke, C., Geiger, A.: Object scene flow. ISPRS Journal of Photogrammetry and Remote Sensing (JPRS) (2018) 8
28. N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, T.Brox: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 8
29. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3, 9
30. Ranjan, A., Janai, J., Geiger, A., Black, M.J.: Attacking optical flow. In: IEEE International Conference on Computer Vision (ICCV) (2019) 14
31. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: Association for the Advancement of Artificial Intelligence (AAAI) (2017) 4, 12
32. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal on Computer Vision (IJCV) **47**(1-3), 7–42 (2002) 4
33. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 1, 3, 5, 9
34. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Models matter, so does training: An empirical study of cnns for optical flow estimation. IEEE Transactions on Pattern Recognition and Machine Intelligence (PAMI) (2019) 8, 9

35. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. arXiv preprint arXiv:2003.12039 (2020) 9
36. Tsai, Y.H., Yang, M.H., Black, M.J.: Video segmentation via object flow. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 1
37. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Xu, W.: Occlusion aware unsupervised learning of optical flow. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4, 12
38. Xu, J., Ranftl, R., Koltun, V.: Accurate Optical Flow via Direct Cost Volume Processing. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 4, 9
39. Yang, G., Ramanan, D.: Volumetric correspondence networks for optical flow. In: Neural Information Processing Systems (NeurIPS) (2019) 3, 4, 5, 9
40. Yin, Z., Darrell, T., Yu, F.: Hierarchical discrete distribution decomposition for match density estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019) 3, 9
41. Yin, Z., Shi, J.: Geonet: Unsupervised learning of dense depth, optical flow and camera pose. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 12
42. Zbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. Journal of Machine Learning Research (JMLR) **17** (2016) 4
43. Zou, Y., Luo, Z., Huang, J.B.: Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In: European Conference on Computer Vision (ECCV) (2018) 12