KalmanFlow 2.0: Efficient Video Optical Flow Estimation via Context-Aware Kalman Filtering

Wenbo Bao, *Student Member, IEEE,* Xiaoyun Zhang, *Member, IEEE,*
Li Chen, *Member, IEEE,* and Zhiyong Gao

***Abstract*—Recent studies on optical flow typically focus on the estimation of the single flow field in-between a pair of images but pay little attention to the multiple consecutive flow fields in a longer video sequence. In this paper, we propose an efficient video optical flow estimation method by exploiting the temporal coherence and context dynamics under a Kalman filtering system. In this system, pixel's motion flow is first formulated as a second-order time-variant state vector, and then optimally estimated according to the *measurement* and *system* noise levels within the system by maximum *a posteriori* criteria. Specifically, we evaluate the measurement noise according to the flow's temporal derivative, spatial gradient, and warping error. And we determine the system noise based on the similarity of contextual information, which is represented by the compact features learned by pre-trained convolutional neural networks. The context-aware Kalman filtering helps improve the robustness of our method against abrupt change of light and occlusion/dis-occlusion in complicated scenes. Experimental results and analyses on the MPI Sintel, Monkaa and Driving video datasets demonstrate that the proposed method performs favorably against the state-of-the-art approaches.**

***Index Terms*—Video Optical Flow, Kalman Filter, Temporal Coherence, Convolutional Neural Networks**

I. Introduction

THE estimation of optical flow, which describes the dense correspondences occurred in dynamic scenes [1], has been a challenging problem in computer vision. It serves numerous related tasks including object tracking [2], video segmentation [3], frame interpolation [4], and video compression [5], to name a few. To determine the optical flow field in-between two sequentially captured images, considerable research has been devoted in literature [1, 6–9].

Horn and Schunck [1] first formulate the optical flow estimation with optimizing an energy function defined over the optical flow field. The function includes a data term that constrains brightness constancy of moving pixels and a spatial term that regularizes the smoothness of flow fields. Based on the variational framework [1], other techniques such as TV-L1 regularization [7, 10], coarse-to-fine architecture [7], feature matching [11, 12] are proposed to improve the overall performance of flow estimation. Besides these algorithms, with the challenging datasets and quantitative evaluations [13, 14], significant progress has been made ever since. On the Middlebury benchmark [14], MDPFlow2 [12] and NNF-Local [15] are among the best methods for their near error-free performance. However, it is noticeable that in this dataset, the sequences are mostly indoor scenes and the typical motion is less than 10 pixels, which cannot cover actual scenes with large motions.

To motivate further research on challenging problems in optical flow estimation, Butler *et al.* propose the MPI Sintel [16] dataset by computer rendering. Compared to the Middlebury dataset [14], this new dataset comes with long video sequences that contain sufficient large displacements at an average of about 20 pixels. It also incorporates motion blur, non-rigid motion, specular reflections, and atmospheric effects to mimic the complex natural scenes. To address these problems, new approaches have been recently proposed [8, 17–21]. The PatchMatch based algorithms [8, 17, 18] exploit the visual correspondence by approximate nearest-neighbor fields [19] for large displacement. The learning based methods, especially the ones that use convolutional neural networks (CNNs), also show effectiveness in dealing with correspondence matching problems. FlowNet [20] trains deep CNNs that are supervised by ground truth flow labels, and its successor FlowNet2.0 [21] obtains comparable results with those of conventional variational optimization based algorithms.

Although consecutive image sequences are provided in the MPI Sintel dataset, most of the existing approaches only use every two of them as a pair to estimate a flow field in-between. Consequently, the temporal correlation that motions in a scene tend to be coherent over time is neglected. However, it has been discovered that motion's temporal coherence can be profitable for estimating optical flow [22–24]. Volz *et al.* [25] and Zimmer *et al.* [26] modeled temporal coherence for optical flow in the variational framework. In Volz *et al.*'s method [25], five image frames are utilized to make an estimation for the flow field of the center one. Sun *et al.* [27] propose a layered motion model where spatial smoothness and temporal coherence are considered in segmentation for layers. They show that image sequences with more frames are demanded to resolve ambiguities in the layer's ordering at occlusion boundaries. Another work by Kennedy and Taylor [28] presents a temporal information enhanced framework by using *inertial estimates* of the flow.

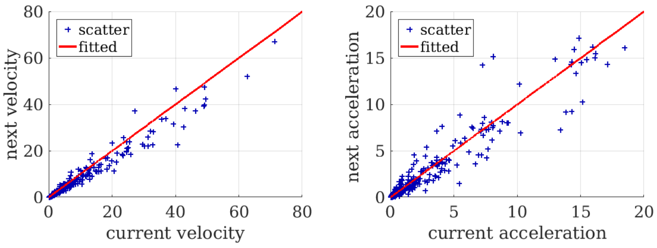
In Figure 1, we make statistics for the optical flow fields of the MPI Sintel dataset to explain the temporal coherence in videos. In (a), the scatter points with plus sign markers illustrate the correlation of velocity magnitudes between current and next flow fields. The fitted line exhibits a positive correlation between the temporally consecutive flow fields. Furthermore, we look into the difference between consecutive

The authors are with Institute of Image Communication and Network Engineering, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China (e-mail: baowenbo@sjtu.edu.cn; xiaoyun.zhang@sjtu.edu.cn; hilichen@sjtu.edu.cn; zhiyong.gao@sjtu.edu.cn). (Corresponding author: Xiaoyun Zhang; Li Chen)

flows, which represents acceleration. The fitted curve shown in Figure 1 (b) inspires us that the second-order motion term also contains temporal coherence though with higher noise distraction. The reason why temporal coherence of motion is rarely exploited in previous works may be summarized in several aspects. First, more frames are required, and it will lead to a complicated objective function based on the existing framework [24, 25, 29]. As a consequence, the computational cost for the numerical optimization of such objection function is prohibitively high. Second, since the variational framework has already penalized flow fields' spatial smoothness, the temporal coherence assumption will lead to the trade-off between temporal and spatial terms. The last but not the least, large displacements, abrupt changes might influence the temporal correlations of flow fields and challenge the effectiveness of the temporal coherence prior [30].

In this paper, we propose a novel method derived from our early conference version [31] to exploit temporal coherence towards robust and accurate optical flows for videos. The proposed method contains substantial improvements in the algorithmic enhancements on context-aware system noise, the better performances on evaluated datasets, and the analysis on parameter sensitivity. We refer to the new method as KalmanFlow2.0 as distinguished from the previous KalmanFlow. In the proposed video flow estimation system, the pixel's motion flow is formulated as a second-order time-variant state vector consisting of velocity and acceleration components. The state vector can be predicted by the transition process as suggested by previous states and also measured by optimization on newly emerged video frames. Since the prediction and measuring processes are noisy with outliers due to the complicity of the real-world object motion, an optimal estimation can be obtained by maximum *a posteriori* criteria according to the measurement and system noise levels. Specifically, the measurement noise covariance is evaluated according to the flow's temporal derivative, spatial gradient, and warping error. Namely, we impose the temporal coherence prior by introducing it into the measurement noise.

Furthermore, considering that there exist complicated scenes such as abrupt change of light and occlusions/dis-occlusions which violate the temporal coherence assumption, we introduce a context-aware algorithm for determining the system noise of our Kalman filter. By computing the contextual similarity of pixel patches to adjust system noise level dynamically, the temporal filters are able to recover from the inconsistency of flow fields. The contextual information is extracted from the learned features of convolutional neural networks, which follows the idea of the feature embedding in [32]. The network is pre-trained to transform image patches into a compact feature representation. It improves the robustness of our Kalman filtering to the abrupt change of light and occlusion/dis-occlusion in complicated scenes. Figure 2 presents the visual comparison of the flow results by DFAuto [33], the proposed KalmanFlow method, and the ground truth. The DFAuto algorithm is recently proposed by Monzon *et al.* [33]. It is a variational optimization based method that considers the regularization strategy for discontinuity preserving. Our method is capable of filtering out the flow outliers on the wall as well as the occluded back of the head.



(a) (b)

Fig. 1: Scatter and fitted plot of the absolute magnitudes of (a) velocity and (b) acceleration on the MPI Sintel dataset [16].

Comparing to the other optical flow estimation methods, the contributions of this paper can be summarized from the following aspects:

1) We model the motion with both velocity and acceleration, and make use of the temporal coherence of video sequences by the Kalman filtering tool for the optical flow estimation. The proposed method is more accurate and efficient than optimizing the classical multi-frame objective variational function.
2) In our filtering system, the noise levels of measurements are evaluated through the consideration of flow's spatial smoothness, temporal derivative, and pixel warping error. And the system noise is designed according to the contextual information extracted from the learned convolutional neural networks. The seamless incorporation of these factors into the temporal filtering system makes our method more robust.
3) With extensive experiments, our approach is demonstrated to be effective in generating temporally coherent and quantitatively accurate results against the state-of-the-art optical flow estimation methods. And it is noted that the upcoming optical flow algorithms in the future may also be readily plugged into our proposed framework for better performances.

The remainder of this paper is organized as follows. Section II discusses the mostly related works in literature. Section III introduces the proposed Kalman filtering method, including the preliminaries and the formulation of our Kalman filtering system. Section IV gives details on the propagation of the Kalman filter from frame to frame, the evaluation of the measurement noise and the context-aware system noise. Experiments and analyses are conducted in Section V and conclusions are drawn in Section VI.

II. Related Work

We introduce the mostly related works to the method in this paper. The interested reader can refer to the literature by Sun *et al.* [34] and Fortune *et al.* [30] for a full review of optical flow estimation.

A. Variational Optimization

The variational optimization method is pioneered by Horn and Schunck [1], and Lucas and Kanade [35]. It optimizes an objective function that generally contains a data term and

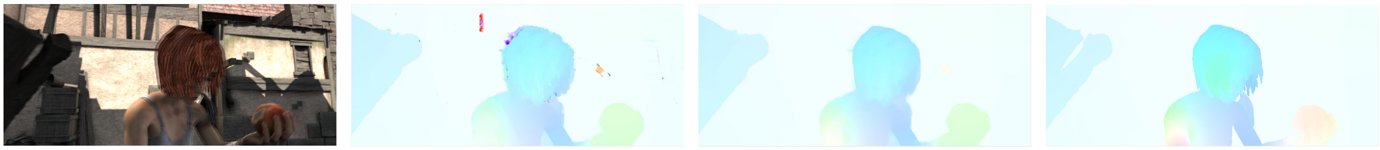

(a) Input images (b) DFAuto [33] (c) KalmanFlow (d) Ground Truth

Fig. 2: Illustration of the effectiveness of KalmanFlow.

a spatial term. The data term penalizes the warping error specified by a flow field from the search space, while the spatial term constrains the search space by using the prior information such as the spatial smoothness of the flow field. Other priors such as the object boundary are more likely to be a motion boundary [36], or objects of a scene should be classified into multiple layers [27] may also be engaged in the literature. For instance, to preserve the motion discontinuity, some studies adopt the gradient guided regularization for the smoothness term [12]. The variational energy function is often optimized by numerical algorithms such as the SOR iterations [7]. We in this paper use the data term and spatial term as indicators for the noise levels in our measurement process. Besides, we introduce the temporal derivatives as a new prior term for better temporal consistency in-between consecutive flow fields.

B. Cost Volume Search

In contrast to the continuous optimization [7] of a variational framework, the discrete optimization based methods, such as PatchMatch and cost volume search, assume integer flow candidates to each pixel and perform an iterative pixel-wise search for optimal vectors. As a representative of discrete algorithms in recent years, the cost volume search approach originated from the visual correspondence [37] is introduced to optical flow estimation [32, 38]. FullFlow [38] presents a global optimization method on the regular 2-dimensional label space of flow fields and achieves state-of-the-art results. DCFlow [32] implements a 4-dimensional cost-volume to estimate optical flow. It calculates the cost through learned feature representations of contextual patches. In contrast to the normalized correlation cost in FullFlow [38], the learned feature is not only robust geometric and radiometric distortions but also helps to construct cost volume at a fast speed thanks to its compact vectorized formulation. In the proposed Kalman filtering system of this paper, we inherit the same spirit by using the contextual similarity represented by learned features to determine dynamic system noises.

C. Convolutional Neural Networks

Convolutional Neural Network (CNN) has stimulated a growing interest in applying learning based methods to computer vision tasks especially after its success on image classification [39]. DeepFlow [40] trains a CNN model for extracting the multi-stage features of patches that are blended into a variational framework. PatchBatch [8] and FlowFieldsCNN [41] propose to extract CNN based image features for the PatchMatch based methods [19]. FlowNet [20] and FlowNet2.0 [21] make significant progress by showing that with pure convolutional neural networks the optical flow can be estimated at a comparable accuracy with the state-of-the-art ones by non-learning based methods. It is also noted that there are some semi-supervised [9] or unsupervised learning methods [42–44]. However, the state-of-the-art performance with neural networks is achieved by Sun *et al.* [45]. Their method called as PWC-Net [45] outperforms all the other methods on records according to the MPI Sintel Optical Flow benchmark of its time. PWC-Net shows us that the insights originated from classical methods including image feature pyramid, frame warping, cost volume can be gracefully combined with regular neural layers, which finally lead to a compact end-to-end trainable network model. Our new method KalmanFlow2.0 has some internal connections with PWC-Net in that the image features and cost volume are considered. But the difference lies in that we use these techniques in a filtering process for more coherent optical flow fields in temporal domain.

Several most recent works [46–49] also pay efforts to employ the temporal coherence within deep learning or variational optimization frameworks. Ren *et al.* [46] train a network to fuse the multiple pre-fetched optical flow fields into a new one. Based on the pipeline in EpicFlow [36], Maurer and Bruhn [47] propose the ProFlow algorithm that uses online learning to make better predictions for unreliable estimations, especially in the occluded area, and achieves the state-of-the-art performance on Sintel benchmark [16]. Janai *et al.* [48] apply unsupervised learning to explicitly reason about occlusions within a multi-frame architecture. Directional priors to the classical variational framework by Maurer *et al.* [49] extend the two-frame approaches to the multi-frame domain. Noticeably, all these four works above use three frames as short term information source, as compared to our long term utilization of frames via recursive Kalman filters.

III. Proposed Method

In this section, we present the framework of the proposed video optical flow estimation method. We first start with the conventional formulation of variational and cost volume search frameworks that only considers brightness constancy and temporal smoothness. And then we describe the proposed Kalman filtering system that takes advantages of temporal coherence.

A. Preliminaries

Variational Framework. Let $\mathbf{I}(\mathrm{t}, \mathbf{x}) : (\mathrm{T}, \Omega) \to \mathbb{R}$ be a video sequence, of which the temporal domain is $\mathrm{T} \subset \mathbb{Z}$ and pixel's spatial domain is $\Omega \subset \mathbb{Z}^2$. Flow field $\mathbf{v}(\mathrm{t}, \mathbf{x})$ between every two consecutive frames represents the motion vector of pixels

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2903656, IEEE Transactions on Image Processing

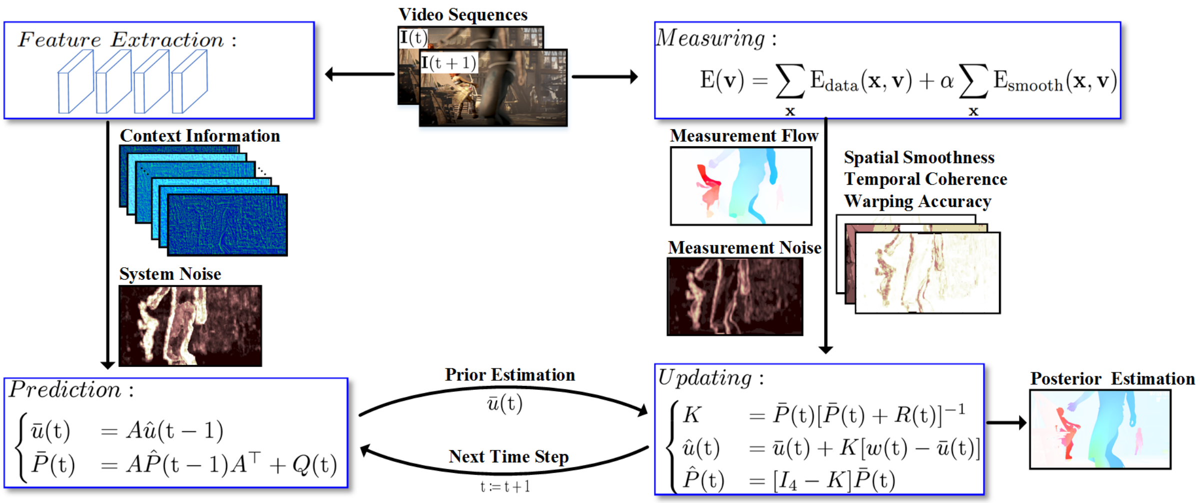

Video Sequences
$\mathbf{I}(\mathrm{t})$
$\mathbf{I}(\mathrm{t}+1)$
Feature Extraction :
Measuring :
$\mathrm{E}(\mathbf{v}) = \sum_{\mathbf{x}} \mathrm{E}_{\mathrm{data}}(\mathbf{x},\mathbf{v}) + \alpha \sum_{\mathbf{x}} \mathrm{E}_{\mathrm{smooth}}(\mathbf{x},\mathbf{v})$
Context Information
System Noise
Measurement Flow
Spatial Smoothness
Temporal Coherence
Warping Accuracy
Measurement Noise
Prediction :
$\bar{u}(\mathrm{t}) = A\hat{u}(\mathrm{t}-1)$
$\bar{P}(\mathrm{t}) = A\hat{P}(\mathrm{t}-1)A^{\top} + Q(\mathrm{t})$
Prior Estimation
$\bar{u}(\mathrm{t})$
Next Time Step
$\mathrm{t}:=\mathrm{t}+1$
Updating :
$K = \bar{P}(\mathrm{t})[\bar{P}(\mathrm{t}) + R(\mathrm{t})]^{-1}$
$\hat{u}(\mathrm{t}) = \bar{u}(\mathrm{t}) + K[w(\mathrm{t}) - \bar{u}(\mathrm{t})]$
$\hat{P}(\mathrm{t}) = [I_4 - K]\bar{P}(\mathrm{t})$
Posterior Estimation

Fig. 3: Flowchart of the proposed video optical flow estimation method. At time step t, we extract the context information by extracting input frames' convolutional features. With the posterior flow field and noise covariance of previous time step $\mathrm{t}-1$, we calculate use the *Prediction* equations to provide a prior estimation of current flow field and evaluate the system noise. Besides, with the given frame $\mathbf{I}(\mathrm{t})$ and $\mathbf{I}(\mathrm{t}+1)$, we use measuring tools that solve the energy function to obtain a measurement flow field. The measurement noise is calculated according to the spatial smoothness, temporal coherence and warping accuracy. We engage the Kalman *Updating* equations to obtain a posterior estimation of the flow field as well as its posterior noise covariance. For the next time step $\mathrm{t} := \mathrm{t}+1$, the above procedures will be recursively executed.

moving from t-th to t+1-th frame. To determine the 2-D optical flow field $\mathbf{v}(\mathrm{t},\mathbf{x}) := (\mathrm{u},\mathrm{v})$, an energy cost function is minimized in a Horn-Schunck-type variational framework [1],

$$\mathrm{E}(\mathbf{v}) = \sum_{\mathbf{x}} \mathrm{E}_{\mathrm{data}}(\mathbf{x},\mathbf{v}) + \alpha \sum_{\mathbf{x}} \mathrm{E}_{\mathrm{smooth}}(\mathbf{x},\mathbf{v}), \quad (1)$$

where

$$\mathrm{E}_{\mathrm{data}}(\mathbf{x},\mathbf{v}) := \Phi(|\mathbf{I}(\mathrm{t}+1,\mathbf{x}+\mathbf{v}) - \mathbf{I}(\mathrm{t},\mathbf{x})|^2), \quad (2)$$

and

$$\mathrm{E}_{\mathrm{smooth}}(\mathbf{x},\mathbf{v}) := \Phi(|\nabla \mathbf{v}_{\mathrm{u}}|^2 + |\nabla \mathbf{v}_{\mathrm{v}}|^2). \quad (3)$$

In the objective function, data term $\mathrm{E}_{\mathrm{data}}(\mathbf{x},\mathbf{v})$ minimizes pixel's warping error pointed by $\mathbf{v}(\mathrm{t},\mathbf{x})$. And regularization term $\mathrm{E}_{\mathrm{smooth}}(\mathbf{x},\mathbf{v})$ is derived from the prior that motion fields tend to be spatially smooth and is calculated according to flow's spatial gradient. α is a parameter tuning the balance between the two terms. The penalty function $\Phi(s) = \sqrt{s^2 + k^2}$ (k is a small constant 0.001) is to avoid singularity problem. By a continuous optimization such as SOR [7] of the function, the flow field $\mathbf{v}(\mathrm{t},\mathbf{x})$ can be obtained.

Cost Volume Search. Another paradigm for estimating optical flow is the discrete optimization of a high structured cost volume. The cost volume is made up of the costs of the integer flow vectors in a 2-dimensional search space. Typical use of the pixel/patch difference as in equation (1) or the normalized cross-corelation as in [38] may not be robust to the visual appearance change of objects. Therefore, a more compact and computationally efficient way is used by Xu [32] and Sun [45]. They train a convolutional neural networks to transform image patches (27 × 27) into a 64-dimensional feature space so as to measure the patch distances in a more compact Euclidean space. The CNN is trained to tell whether a given pair of patches are similar or not according to their transformed features. With the transformed pixel-wise features of two images at time step t and $\mathrm{t}+1$, the 4-dimensional cost volume is evaluated by

$$\mathrm{C}(\mathbf{x},\mathbf{v}) = 1 - \big(\mathbf{F}(\mathrm{t},\mathbf{x})\big)^{\top}\big(\mathbf{F}(\mathrm{t}+1,\mathbf{x}+\mathbf{v})\big) \quad (4)$$

where $\mathbf{F}(\mathrm{t},\mathbf{x})$ is the convolutional features of frame $\mathbf{I}(\mathrm{t},\mathbf{x})$. A higher similarity of the features will lead to a smaller cost value. With the cost volume constructed, a discrete energy function over the integral flow candidates is established. It takes the cost values as the data term of equation (1) and the spatial smoothness between neighboring flow vectors. Finally, the optical flow field can be estimated by discrete optimization such as the SGM algorithm [32] of this regular cost volume structure.

Though the variational framework [7, 28] and the cost volume search [32, 45] are widely used in different literature, both of them are essentially built on the similarity of data as well as the spatial smoothness of the flow field. Meanwhile, the flow field's temporal coherence has been little studied in literature [22, 23, 25, 50]. In this paper, with the help of Kalman filtering tool, we propose to exploit the temporal coherence to obtain video's flow. The filtering process is performed in a recursive way that only two video frames and the predicted flow fields are required for each time step, which makes our method efficient.

B. *Kalman Filtering*

Kalman filtering [51] is a high-efficiency optimal estimation tool applied in many areas including auto control [52], signal processing, *etc*. Particularly, in image/video processing, it has also been widely adopted in object tracking [53, 54], video error concealment [55], *etc*. For a time-variant dynamic system, its *state vector* follows a transition function over time.

To get the value of target state vector, a measuring process is usually executed. The measurement result may be derived from state vector directly or indirectly, which is formulated by a measurement function. Since the transition and measuring process are both noisy in real-world applications, the optimal estimation of state vectors is required given its measurements and transition process, which compose the idea of Kalman filtering.

In our filtering system for flow field, a second-order approximation for motion is adopted, *i.e.*, motion is modeled as an accelerated process. Thus, the state vector to be filtered for each pixel location in our system is defined as

$$u(\mathrm{t}) = [\mathbf{v}(\mathrm{t}), \mathbf{a}(\mathrm{t})]^\top . \tag{5}$$

We here omit the coordinate index $\mathbf{x}$ of velocity $\mathbf{v}(\mathrm{t}, \mathbf{x})$ and acceleration $\mathbf{a}(\mathrm{t}, \mathbf{x})$ for convenience. Since previous works [22, 23] made use of temporal coherence only with a constant assumption, we use an acceleration term for higher precision of motion. This acceleration assumption for motion is also implicitly adopted by [25] with a second-order regularization for flow field. With an accelerated motion model, the state vectors transited over time can be formulated by

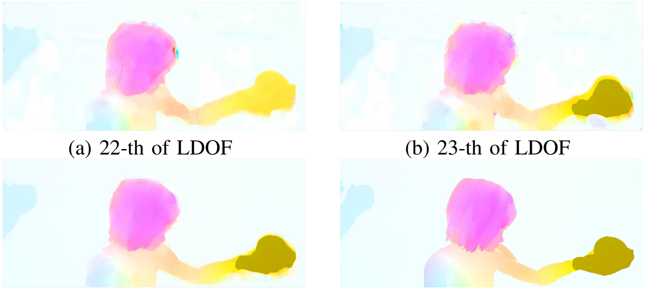
$$\begin{aligned} u(\mathrm{t}) &= \begin{bmatrix} \mathbf{v}(\mathrm{t})^\top \\ \mathbf{a}(\mathrm{t})^\top \end{bmatrix} = \begin{bmatrix} \mathbf{v}(\mathrm{t}-1)^\top + \mathbf{a}(\mathrm{t}-1)^\top \\ \mathbf{a}(\mathrm{t}-1)^\top \end{bmatrix} \\ &\triangleq Au(\mathrm{t}-1) + \varepsilon(\mathrm{t}) \end{aligned} \tag{6}$$

where the transition matrix A is a 4×4 square matrix as $\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \otimes I_2$, where $\otimes$ represents a Kronecker product and I_n is an n-dimension identity matrix. By this transition function, the horizontal and vertical component of motion vectors are assumed to be independent. The system noise $\varepsilon(\mathrm{t})$ follows an independent identical Gaussian distribution with covariance matrix $Q(\mathrm{t})$.

Then, in the measuring process, the measurement vector is directly derived from state vector.

$$w(\mathrm{t}) = u(\mathrm{t}) + \eta(\mathrm{t}) \tag{7}$$

Similarly, the noise term $\eta(\mathrm{t})$ is assumed to be Gaussian distributed with covariance matrix $R(\mathrm{t})$. Measurement of velocity vector $\mathbf{v}(\mathrm{t})$ in $w(\mathrm{t})$ is obtained by solving the energy function (1) by continuous or discrete optimization algorithms that are off-the-shelf in literature. We compute the acceleration vector $\mathbf{a}(\mathrm{t})$ through an indirect way. It is given by $\mathbf{a}(\mathrm{t}) = \mathbf{v}(\mathrm{t}) + \mathbf{v}^{\mathrm{b}}(\mathrm{t})$ where $\mathbf{v}^{\mathrm{b}}(\mathrm{t})$ is backward motion vector of t-th frame to $\mathrm{t}-1$-th frame. Note that the backward velocity is negatively proportional to time increasing direction, thus the forward and backward velocity are summed up to approximate the acceleration vector.


(a) 22-th of LDOF (b) 23-th of LDOF

(c) 23-th of KalmanFlow (d) 23-th of Groundtruth

Fig. 4: (a) and (b) are the 22-th and 23-th flow field generated by LDOF [56] algorithm where outliers exist in various regions including motion boudaries, flat areas, *etc*. (c) and (d) are the 23-th flow field generated by KalmanFlow and the groudtruth flow.

According to the theory of Kalman filtering [51], the prediction and updating equations can be derived.

$$Prediction: \\ \begin{cases} \bar{u}(\mathrm{t}) &= A\hat{u}(\mathrm{t}-1) \\ \bar{P}(\mathrm{t}) &= A\hat{P}(\mathrm{t}-1)A^\top + Q(\mathrm{t}) \end{cases} \tag{8}$$

$$Updating: \\ \begin{cases} K &= \bar{P}(\mathrm{t})[\bar{P}(\mathrm{t}) + R(\mathrm{t})]^{-1} \\ \hat{u}(\mathrm{t}) &= \bar{u}(\mathrm{t}) + K[w(\mathrm{t}) - \bar{u}(\mathrm{t})] \\ \hat{P}(\mathrm{t}) &= [I_4 - K]\bar{P}(\mathrm{t}) \end{cases} \tag{9}$$

In these equations, $\bar{u}$ and $\hat{u}$ represent the prior and posterior estimation for the state vector u, respectively. Correspondingly, $\bar{P}$ and $\hat{P}$ are the prior and posterior noise covariances. *Prediction* equations (8) calculate the predicted state vector and its system noise covariance, while *Updating* equations (9) fuse the prior prediction and measurement with maximum a posteriori estimation of u according to the noise covariances.

By the above prediction and updating equations, for current time step t, the filtering process is as follows. We first obtain a prior estimation $\bar{u}(\mathrm{t}+1)$ without $\mathbf{I}(\mathrm{t}+1)$ by the prediction equations (8). Simultaneously, its noise covariance $\bar{P}(\mathrm{t}+1)$ is determined according to previous optimal estimation's noise covariance $\hat{P}(\mathrm{t})$. Then, once given the frame $\mathbf{I}(\mathrm{t}+1)$, we calculate the forward and backward optical flow, and a measurement $w(\mathrm{t})$ can be obtained. We then evaluate its noise by some criterion to get $R(\mathrm{t})$. By determining a Kalman Gain K according to $\bar{P}(\mathrm{t})$ and $R(\mathrm{t})$, the posterior estimation $\hat{u}(\mathrm{t}+1)$ of time step t is obtained. The whole process of KalmanFlow2.0 is illustrated in the Figure 3. Figure 4 (a) and (b) illustrate two consecutive flow fields by LDOF algorithm [56]. Figure 4 (c) and (d) illustrate the results of KalmanFlow and the ground truth. The outliers on the moving head and underarm are eliminated by KalmanFlow.

IV. Implementation Details

In the Kalman filtering process, three issues are critical to the solution of the optimal estimation. The first issue is how we get the previous estimations so as to keep a continuous tracking of the same object. Another issue lies in that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2903656, IEEE Transactions on Image Processing

evaluation of measurement and system noise plays a significant role in balancing the prediction and measurement. Moreover, the third issue is that how we keep a robust estimation for complicated scenarios including abrupt change of light, occlusion/dis-occlusion, *etc*.

A. Propagation of Kalman Filters

In the filtering system, we assign each pixel location a Kalman filter to do the optimal estimation correspondingly. When it comes to a new time step, due to the object motions, these filters should also be propagated to their new locations so as to maintain the historical information for correct estimation. These information includes the previous optimal state vector and its posterior noise covariance. However, the problem is that there may exist occlusions and dis-occlusions near motion boundaries. Hence, we need to drop off the filters for those to-be-occluded pixel locations and initialize new filters for newly emerged object pixels.

Given flow field $\mathbf{v}(\mathrm{t}, \mathbf{x})$ of time t, the Kalman filters will propagate to their new locations at $\mathbf{y} = \mathbf{x} + \mathbf{v} \in \Omega_{\mathrm{t}+1}$. The subscript $\mathrm{t}+1$ for Ω represents a new coordinate space of time step $\mathrm{t}+1$. However, because of non-translational motion or occlusion, multiple pixels may propagate to a same location. Denote that set $\mathcal{S}(\mathbf{y}) = \{\mathbf{x}_\mathrm{i} : \mathbf{x}_\mathrm{i} + \mathbf{v}_\mathrm{i} = \mathbf{y}, \forall \mathbf{x}_\mathrm{i} \in \Omega_\mathrm{t}, \mathbf{y} \in \Omega_{\mathrm{t}+1}\}$ contains pixels of $\mathbf{x}_\mathrm{i}$ mapped to the same location $\mathbf{y}$. The unique active pixel in set $\mathcal{S}(\mathbf{y})$ that will not be occluded is determined by

$$\mathcal{A}(\mathbf{y}) = \{\mathbf{x}^\star = \underset{\mathbf{x}_\mathrm{i} \in \mathcal{S}(\mathbf{y})}{\arg\min}\, \mathrm{E}_{\mathrm{data}}(\mathbf{x}_\mathrm{i}, \mathbf{v}_\mathrm{i})\} \qquad (10)$$

Here, we use the data term $\mathrm{E}_{\mathrm{data}}(\mathbf{x}_\mathrm{i}, \mathbf{v}_\mathrm{i})$ to decide which filter will win over others and is active in next frame. Then, the propagated Kalman filter for each pixel location $\mathbf{y} \in \Omega_{\mathrm{t}+1}$ is chosen from $\mathcal{A}(\mathbf{y})$. For the location that have no previous Kalman filter, namely $\mathcal{S}(\mathbf{y}) = \varnothing$ and $\mathcal{A}(\mathbf{y}) = \varnothing$, a fresh new Kalman filter will be initialized as needed. The initialized filter will set its noise covariance of prior estimation $\bar{u}(\mathrm{t})$ to be infinite, namely $\bar{P}(\mathrm{t}) = \mathbf{inf}$, representing that the prior estimations are untrusted.

B. Temporal Coherence Enhanced Measurement Noise

Since Kalman filter is a recursive optimal filter, the quality of results is highly dependent on the estimation of measurement noise $\eta(\mathrm{t})$ and system noise $\varepsilon(\mathrm{t})$. For the measurement noise $\eta(\mathrm{t})$, a straightforward idea may come to mind that it is correlated with flow field's data term and smoothness term as used in the energy function (1). However, the methods minimizing the two terms may occasionally produce incorrect motion vectors but with low cost. And it is also found that these outliers do not occur at the same location because they are mainly caused by the complex motions, the local minimum in energy optimization, *etc*. Thus, we propose a temporal coherence term to enhance the evaluation of vector's noise.

In our method, the pixel-wise noise variance for a flow field $\mathbf{v}$ is given as follows,

$$\sigma^2(\mathbf{x}, \mathbf{v}) = C - \exp^{-\gamma \mathrm{E}_{\mathrm{data}}(\mathbf{x}, \mathbf{v})} - \exp^{\beta \mathrm{E}_{\mathrm{smooth}}(\mathbf{x}, \mathbf{v})} - \exp^{-\tau \mathrm{E}_{\mathrm{temporal}}(\mathbf{x}, \mathbf{v})}. \qquad (11)$$

Algorithm 1 The KalmanFlow 2.0 for Video Sequences

```
Input: The input video sequence $\{\mathbf{I}(\mathrm{t}) : \mathrm{t} \in (1, \mathrm{T})\}$.
for $\mathrm{t} = 1$ to $\mathrm{T} - 1$ do
    if $\mathrm{t} = 1$ then
        Initialize $\bar{P}(\mathrm{t}) = \mathbf{inf}$.
        Initialize $\bar{u}(\mathrm{t}) = \mathbf{0}$.
    else
        Compute $\bar{u}(\mathrm{t})$.
        Compute $Q(\mathrm{t})$ by equation (14) or (15).
        Compute $\bar{P}(\mathrm{t})$ by equations (8).
    end if
        /*Calculation for current time step*/
    Compute $w(\mathrm{t})$ by solving equation (1).
    Compute $R(\mathrm{t})$ by equation (13).
    Compute $\hat{u}(\mathrm{t})$ and $\hat{P}(\mathrm{t})$ by equations (9).
        /*Preparation for next time step*/
    Propagate filters according to equation (10).
end for
Output: Video's optical flow $\{\hat{u}(\mathrm{t}) : \mathrm{t} \in (1, \mathrm{T})\}$.
```

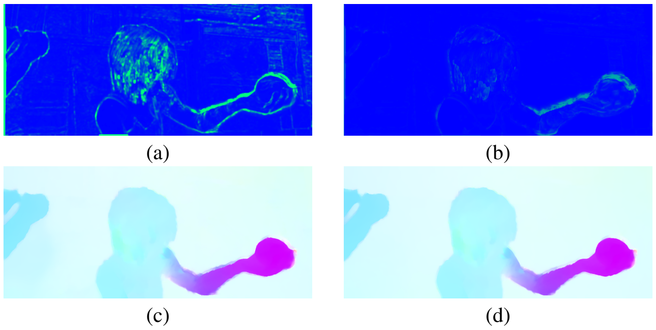

(a) (b) (c) (d)

Fig. 5: Illustration of the (a) first main component of measurement noise $R(\mathrm{t})$, (b) first main component of prediction noise $\bar{P}(\mathrm{t})$, (c) measurement optical flow by DCFlow and (d) optimal estimation by KalmanFlow2.0.

In this equation, the noise is related to three terms, including warping error $\mathrm{E}_{\mathrm{data}}(\mathbf{x}, \mathbf{v})$, spatial smoothness error $\mathrm{E}_{\mathrm{smooth}}(\mathbf{x}, \mathbf{v})$ and temporal coherence error $\mathrm{E}_{\mathrm{temporal}}(\mathbf{x}, \mathbf{v})$. They are fused by coefficient γ, β and τ. The constant C is set to 3.0, constraining the noise variance to the range of 0.0 and 3.0. Moreover, the temporal coherence term $\mathrm{E}_{\mathrm{temporal}}(\mathbf{x}, \mathbf{v})$ is given by

$$\mathrm{E}_{\mathrm{temporal}}(\mathbf{x}, \mathbf{v}) = \Phi(||\mathbf{v}(\mathrm{t}, \mathbf{x}) - \mathbf{v}'(\mathrm{t}, \mathbf{x})||_2), \qquad (12)$$

where $\mathbf{v}'$ is the propagated flow vector of previous time step, *i.e.*, $\mathbf{v}'(\mathrm{t}, \mathbf{x}) = \mathbf{v}(\mathrm{t}-1, \mathbf{x}_\mathrm{j} + \mathbf{v}_\mathrm{j})$, similar to the propagation of filters in Section IV-A. When the noise covariances of $\mathbf{v}$ and $\mathbf{v}^b$ are calculated, the measurement noise $R(\mathrm{t})$ is obtained by

$$R(\mathrm{t}) = diag\Big(\sigma^2(\mathbf{x}, \mathbf{v}), \sigma^2(\mathbf{x}, \mathbf{v}^b) + \sigma^2(\mathbf{x}, \mathbf{v})\Big) \otimes I_2, \qquad (13)$$

where $diag(\cdot)$ is the operator to compose a diagonal matrix with given elements.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2903656, IEEE Transactions on Image Processing

C. Context-Aware System Noise

Generally, the system noise covariance matrix Q can be set as a constant matrix,

$$Q = \kappa I_4, \tag{14}$$

where we empirically set κ to a small constant $\kappa = 0.001$. For most cases where motion keeps a good continuity, this constant assumption for the system noise covariance is reasonable.

In the recent benchmarks [16], optical flow estimation is faced with challenges like large displacement, appearance change and occlusion/dis-occlusion. To make the Kalman filter robust to these challenges, we propose to adaptively adjust the filter according to the cost of the predicted flow. We inherit the evaluation of the cost volume that extract the contextual information to determine the system noise. Specifically, the constant parameter κ in equation (14) is replaced with a context-aware factor $\kappa(\mathrm{t})$, leading to a time-variant system noise as follows:

$$\begin{aligned} Q(\mathrm{t}) &= \kappa(\mathrm{t}) I_4 \\ &= \left(1 - \exp^{-\mathrm{C}(\mathbf{x}, \bar{\mathbf{v}})}\right) I_4, \end{aligned} \tag{15}$$

where the $\bar{\mathbf{v}}$ is extracted from the predicted state vector $\bar{u}$. When contexts between the pixel $\mathbf{x}$ of step t and $\mathbf{x} + \bar{\mathbf{v}}$ of step $\mathrm{t}{+}1$ change sharply, a large cost value $\mathrm{C}(\mathbf{x}, \bar{\mathbf{v}})$ of the predicted optical flow vector is obtained. Consequently, we get a large system noise covariance $Q(\mathrm{t})$. The prior noise covariance $\bar{P}(\mathrm{t})$ will grow large if the previous $Q(\mathrm{t})$ has been consistently high in the past time steps. Thus according to the calculation of Kalman Gain K in the *Updating* equations (9), the filter will get optimal estimation closer to the measurement result. Inversely, when the variation of context correlation is low, $\kappa(\mathrm{t})$ will keep at a small value, so that the flow field will be refined with high reliability. We list the detailed procedures of the proposed method in Algorithm (1).

We present an example in Figure 5 for a visualized concept of the measurement and prediction noise. It is noticed that for most areas, both the measured and predicted flow fields are satisfying. However, for the occluded areas such as the static background above the moving arm, the cost volume search algorithm cannot generate good results. But according to the history information that the background is mostly static, the predicted optical flow is more accurate than the measured one. Finally, the outliers for the static background will be corrected in the optimal estimation.

V. Experimental Results

In recent years, the most popular datasets to evaluate optical flow algorithms are Middlebury [14], KITTI [57], and MPI Sintel [16]. Among the three datasets, we use the MPI Sintel that provides video sequences of various scenes including large motions, specular reflections, motion blur, *etc*. In this dataset, the *Clean* pass is rendered with shading effect but no image degradations, whereas the *Final* pass additionally includes motion blur, defocus blur, and atmospheric effects.

We evaluate our methods against eight different optical flow estimation algorithms. First of all, the DFAuto [33] algorithm is a variational optimization based method that considers the regularization strategy for discontinuity preserving. Then, three remarkable algorithms aimed at resolving large displacements including MDP-Flow2 [12], LDOF [56], and SIFTflow [11] are also brought in for comparisons. Besides, the adopted EpicFlow [36] calculates the dense flow field by edge-preserved interpolation of the sparse field initialized by feature descriptors, while FullFlow [38] optimizes the classical flow objective over the full space of mappings between discrete grids without the use of descriptors. We also use the DCFlow [32] and PWC-Net [45], both of which are the recent state-of-the-art algorithms on the MPI Sintel benchmark.

For each of these eight algorithms, the KalmanFlow or KalmanFlow2.0 filtering process is implemented as follows. By the provided source codes of the algorithms including FullFlow [38], MDPFlow2 [12], LDOF [56], SIFTFlow [11], EpicFlow [36], DFAuto [58], DCFlow [32], and PWC-Net [45], initial measurements of the flow fields for video sequences are obtained. Then we perform Kalman filtering on these measurements and denote the filtered results as MDPFlow2+KF (KF is short for KalmanFlow), LDOF+KF, SIFTFlow+KF, EpicFlow+KF, DFAuto+KF, DCFlow+KF, and PWC-Net+KF. We implement the KalmanFlow2.0 on DCFlow [32] and PWC-Net [45] to make full understandings of how contextual information for dynamic system noise helps improve the robustness of filtering. We note them as DCFlow+KF2 and PWC-Net+KF2 respectively. The controlling parameters required by KalmanFlow and KalmanFlow2.0 are kept the same for all baseline algorithms and evaluated datasets. In the evaluation of flow field's noise by equation (11), parameter γ, β, and τ are set to 0.1, 0.30, and 0.02 respectively. The parameter sensitivity will be discussed in the subsection V-D.

All the experiments are performed on a machine with Intel Core i7 3.5GHz CPU and 32GB RAM. On average, for each image pair (1024×436) in MPI Sintel dataset, our KalmanFlow2.0 implemented with Matlab consumes about 20 seconds. The speed of Kalman filtering can be much faster thanks to its full parallelism of pixel-wise filters.

A. Effectiveness of KalmanFlow

Table I and II present quantitative results on the Test and Training set of the MPI Sintel benchmark respectively. We use the EPE (EndPoint Error) to evaluate the quantitative performance of optical flow estimation. A smaller EPE value represents better performance. The tables list six different pixel regions of the average EPE for *all* (all pixels), *non* (non-occluded pixels), *occ* (occluded pixels), *d0-10* (within 10 pixels of an occlusion boundary) and *s40+* (displacements larger than 40 pixels).

In Table I, for all the variational based algorithms such as MDPFlow2, LDOF, and SIFTFlow, the improvement of our KalmanFlow is significant. The improvements on EPE for *all* pixel regions of the three algorithms are about $0.4 \sim 0.6$ in *Final* pass, and $0.1 \sim 0.9$ in *Clean* pass. For particular pixel regions such as *non* and *occ*, the improvements are also evident between $0.1 \sim 1.0$ and $0.5 \sim 2.0$ respectively. Overall, our KalmanFlow can effectively improve the performance of baseline algorithms. Meanwhile, we notice that there are cases

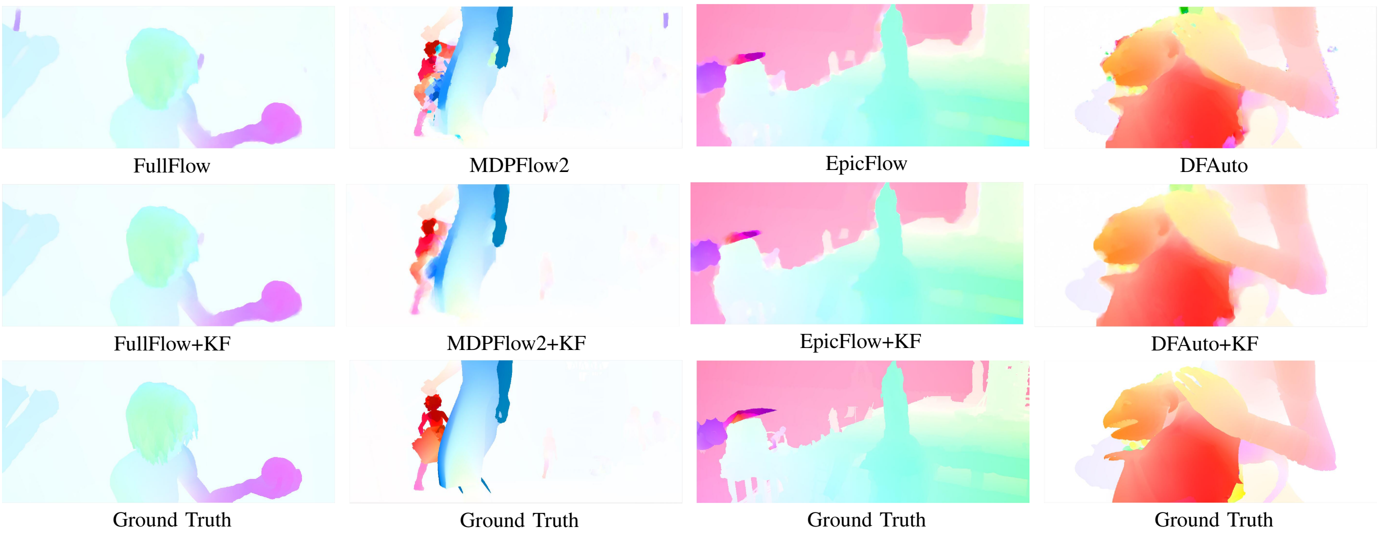
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2903656, IEEE Transactions on Image Processing

TABLE I: AVERAGE ENDPOINT ERROR OF DIFFERENT ALGORITHMS ON THE MPI SINTEL TEST SET. *all* = OVER THE WHOLE IMAGE. *noc* = NON-OCCLUDED PIXELS. *occ* = OCCLUDED PIXELS. *d0-10* = WITHIN 10 PIXELS OF AN OCCLUSION BOUNDARY. *s40+* = DISPLACEMENTS LARGER THAN 40 PIXELS.

TEST set	*Final* pass						*Clean* pass					
	all	*non*	*occ*	*d0-10*	*d10-60*	*s40+*	*all*	*non*	*occ*	*d0-10*	*d10-60*	*s40+*
FullFlow+KF	**5.802**	2.761	30.58	4.866	2.438	35.51	**3.598**	1.247	22.77	2.957	0.977	21.21
FlowFields [18]	5.810	2.621	31.79	4.851	2.232	33.89	3.748	1.056	25.70	2.784	0.878	23.60
FullFlow [38]	5.895	2.838	30.79	4.905	2.506	35.59	3.601	1.296	22.42	2.944	1.023	20.61
MDPFlow2+KF	**8.078**	3.938	41.78	5.642	3.691	49.31	**5.711**	1.842	37.20	3.370	1.846	39.39
MDPFlow2 [12]	8.445	4.150	43.43	5.703	3.925	50.50	5.837	1.869	38.15	3.210	1.913	39.45
LDOF+KF	**8.751**	4.298	44.99	6.086	4.084	54.83	**6.592**	2.449	40.30	4.248	2.351	44.25
LDOF [56]	9.116	5.037	42.34	6.849	4.928	57.29	7.563	3.342	41.17	5.353	3.284	51.69
SIFTFlow+KF	**9.317**	4.527	48.30	6.366	4.405	60.48	**8.280**	3.524	46.96	5.511	3.554	58.25
SIFTFlow [11]	9.941	4.987	50.34	7.164	4.791	61.53	8.898	4.008	48.85	6.490	3.943	58.95

TABLE II: AVERAGE ENDPOINT ERROR OF DIFFERENT ALGORITHMS ON THE MPI SINTEL TRAINING SET.

TRAINING set	*Final* pass						*Clean* pass					
	all	*non*	*occ*	*d0-10*	*d10-60*	*s40+*	*all*	*non*	*occ*	*d0-10*	*d10-60*	*s40+*
FullFlow+KF	**3.744**	2.229	22.97	3.820	1.894	26.78	**2.456**	1.166	18.83	2.559	0.836	18.38
FullFlow [38]	3.816	2.273	23.41	3.837	1.947	26.96	2.498	1.184	19.17	2.533	0.868	18.34
EpicFlow+KF	**3.708**	2.235	22.39	3.917	1.884	26.19	**2.220**	0.994	17.78	2.505	0.578	14.69
EpicFlow [38]	3.760	2.266	22.72	3.956	1.916	26.19	2.263	1.009	18.18	2.540	0.585	14.64
MDPFlow2+KF	**5.266**	3.229	31.12	4.649	2.814	36.27	**3.065**	1.331	25.07	2.811	1.020	21.38
MDPFlow2 [12]	5.728	3.468	34.40	4.827	3.091	37.62	3.299	1.376	27.70	2.749	1.117	22.23
LDOF+KF	**5.149**	3.067	31.57	4.877	2.650	35.47	**3.375**	1.587	26.07	3.334	1.175	23.56
LDOF [56]	6.205	3.595	39.34	5.702	3.124	39.41	4.099	1.944	31.45	4.013	1.423	26.31
SIFTFlow+KF	**5.373**	3.140	33.71	5.240	2.830	39.13	**4.389**	2.389	30.71	4.381	1.954	33.30
SIFTFlow [11]	5.943	3.543	36.41	5.890	3.154	40.59	4.959	2.728	33.27	5.071	2.284	34.59
DFAuto+KF	**7.004**	4.827	34.63	7.220	4.734	56.15	**6.677**	4.481	34.54	7.099	4.442	56.63
DFAuto [33]	7.375	5.084	36.45	7.705	4.977	57.16	7.021	4.700	36.48	7.530	4.625	57.47


FullFlow MDPFlow2 EpicFlow DFAuto

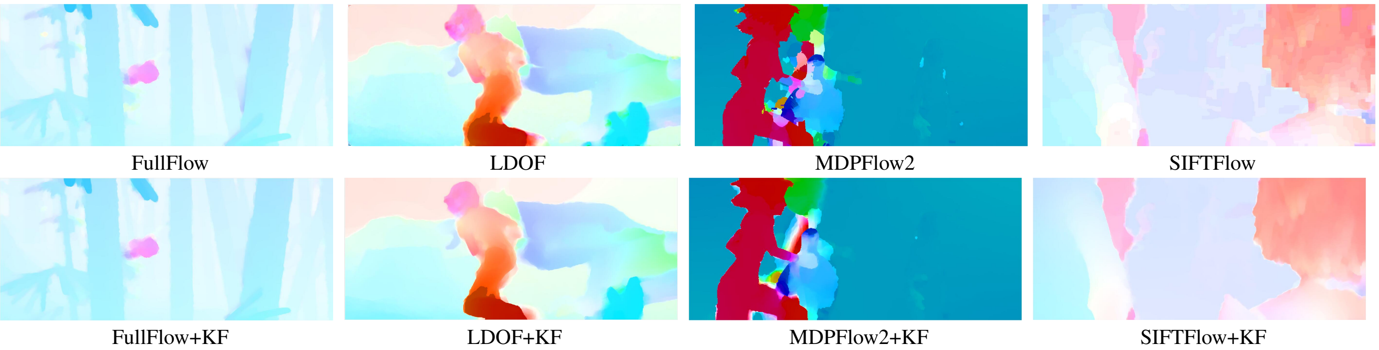
FullFlow+KF MDPFlow2+KF EpicFlow+KF DFAuto+KF

Ground Truth Ground Truth Ground Truth Ground Truth

Fig. 6: Visual comparison of different algorithms on the MPI Sintel TRAINING set. The flow fields from left to right are from *alley_1*, *market_2*, *temple_2* and *bandage_1* respectively.

when the results are worsening by KalmanFlow. For example, the *occ* region of *Final* pass by LDOF and LDOF+KF are 42.34 and 44.49. This problem is mainly due to the long-term error of some large displacements in complex scenes. However, it does not affect the overall advantage of KalmanFlow. The *non* region by the two methods are 5.037 and 4.29 respectively, and the result for *all* region are 9.116 and 8.751, demonstrating that our algorithm can provide better flow fields.

Moreover, the FullFlow are improved by KalmanFlow since the *Final* pass in *all* region is refined from 5.895 to 5.802. The margin 0.093 may not be as large as what KalmanFlow has achieved over MDPFlow2, but it helps the FullFlow algorithm outperform FlowFields [18]. The reason is that with EPE getting closer to zeros, the improvement of it becomes difficult.


FullFlow LDOF MDPFlow2 SIFTFlow

FullFlow+KF LDOF+KF MDPFlow2+KF SIFTFlow+KF

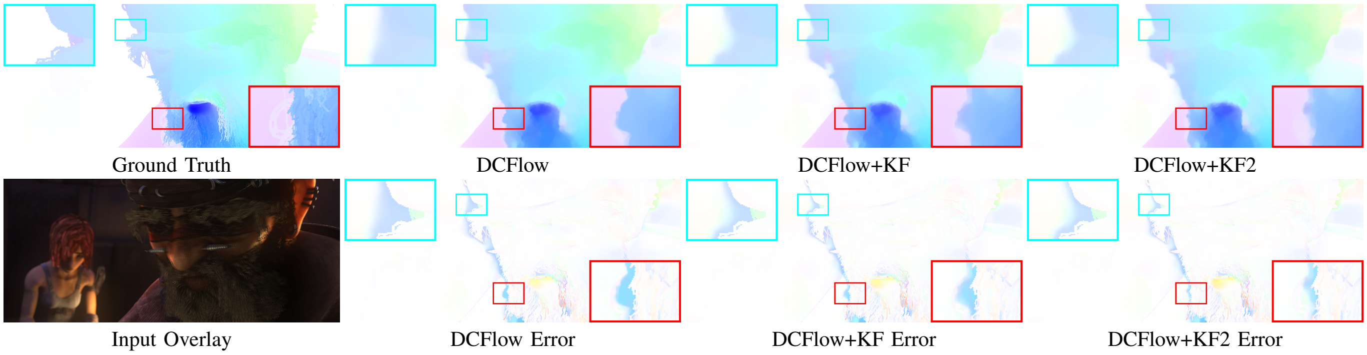
Fig. 7: Visual comparison of different algorithms on the MPI Sintel TEST set. The flow fields from left to right are from *bamboo_3*, *cave_3*, *market_1* and *PERTURBED_market_3* respectively.

TABLE III: AVERAGE ENDPOINT ERROR OF DCFLOW WITH OR WITHOUT THE PROPOSED KALMANFLOW AND KALMANFLOW2.0 ALGORITHMS ON THE MPI SINTEL TRAINING AND TEST SET.

Dataset	*Method*	*Final* pass						*Clean* pass					
		all	*non*	*occ*	*d0-10*	*d10-60*	*s40+*	*all*	*non*	*occ*	*d0-10*	*d10-60*	*s40+*
TRAINING	DCFlow+KF2	**3.249**	1.842	21.09	3.409	1.478	22.02	**2.069**	0.881	17.15	2.263	0.498	14.10
	DCFlow+KF	3.382	1.866	22.61	3.450	1.500	23.04	2.256	0.983	18.41	2.360	0.631	14.65
	DCFlow [32]	3.353	1.899	21.81	3.473	1.528	22.31	2.114	0.908	17.42	2.263	0.526	13.88
TEST	DCFlow+KF2	**5.067**	2.195	28.47	4.652	1.948	29.57	3.645	1.149	23.99	3.037	0.932	22.86
	DCFlow+KF	5.120	2.245	28.55	4.747	1.981	29.51	3.585	1.142	23.50	3.036	0.921	22.20
	DCFlow [32]	5.119	2.283	28.22	4.665	2.108	29.35	**3.537**	1.103	23.39	2.897	0.868	21.29

TABLE IV: AVERAGE ENDPOINT ERROR OF PWC-NET WITH OR WITHOUT THE PROPOSED KALMANFLOW AND KALMANFLOW2.0 ALGORITHMS ON THE MPI SINTEL TRAINING AND TEST SET.

Dataset	*Method*	*Final* pass						*Clean* pass					
		all	*non*	*occ*	*d0-10*	*d10-60*	*s40+*	*all*	*non*	*occ*	*d0-10*	*d10-60*	*s40+*
TRAINING	PWC-Net+KF2	**2.275**	1.326	14.32	2.789	0.988	14.76	**1.749**	0.881	12.76	2.23	0.475	11.35
	PWC-Net+KF	2.357	1.331	15.38	2.799	0.992	15.55	1.830	0.884	13.84	2.244	0.476	12.15
	PWC-Net [45]	2.302	1.367	14.16	2.858	1.020	14.64	1.814	0.956	12.70	2.375	0.535	11.30
TEST	PWC-Net+KF2	4.979	2.430	25.78	4.571	2.078	30.35	**3.753**	1.588	21.44	3.657	1.298	24.70
	PWC-Net+KF	**4.964**	2.438	25.56	4.603	2.066	30.46	3.850	1.595	22.27	3.664	1.302	25.28
	PWC-Net [45]	5.042	2.445	26.22	4.636	2.087	31.07	4.386	1.719	26.16	4.282	1.657	28.79

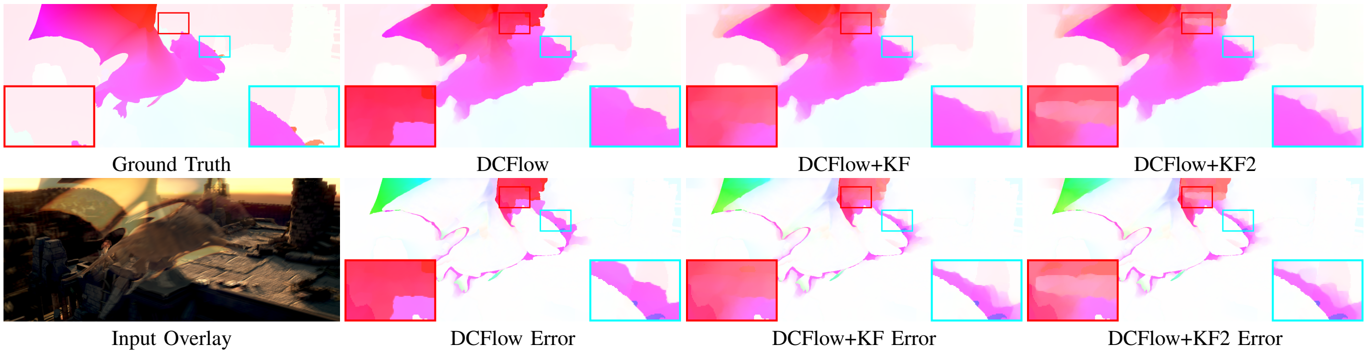

Ground Truth DCFlow DCFlow+KF DCFlow+KF2

Input Overlay DCFlow Error DCFlow+KF Error DCFlow+KF2 Error

Fig. 8: Visual comparison on the 18-th optical flow field of *shaman_2* sequence.

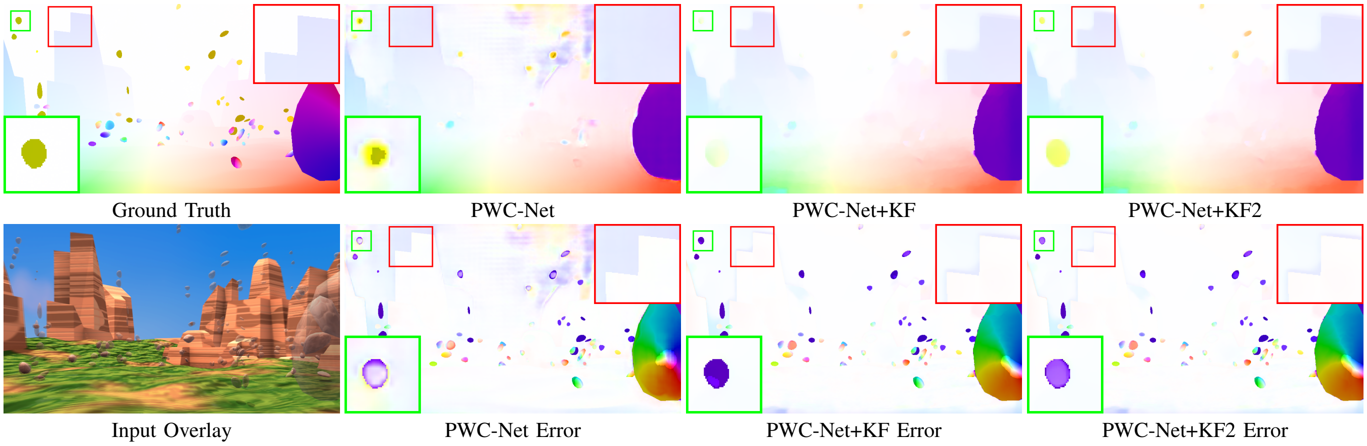
We note that the EPEs of the top-ranked methods are very close to each other. More results for the TRAINING set of MPI Sintel are provided in Table II. The EpicFlow and DFAuto are added to assess the KalmanFlow's capability further. In this table, almost all of the pixel regions by the six algorithms gain improvements from KalmanFlow.

Figure 6 and 7 depict some of the visual images of coded optical flow. In brief, the tones of the coded flow field image represent the directions of motions, and the saturations are for magnitudes of the motions. In Figure 6, some estimated flow fields in the TRAINING set of MPI Sintel is provided. The four flow fields of the first row are generated by FullFlow, MDPFlow2, EpicFlow, and DFAuto. The second and third rows are KalmanFlow's results and the ground truth respectively. It can be found that outliers frequently occur around motion boundaries, but can be robustly eliminated or mitigated by KalmanFlow. The incorrect motions of occluded regions can be refined by the predictive nature of Kalman filter. And


Ground Truth DCFlow DCFlow+KF DCFlow+KF2

Input Overlay DCFlow Error DCFlow+KF Error DCFlow+KF2 Error

Fig. 9: Visual comparison on the 24-th optical flow field of *temple_2* sequence.


Ground Truth PWC-Net PWC-Net+KF PWC-Net+KF2

Input Overlay PWC-Net Error PWC-Net+KF Error PWC-Net+KF2 Error

Fig. 10: Visual comparison on the 48-th optical flow field of *a_rain_of_stones_x2* sequence from the Monkaa dataset [59].

TABLE V: AVERAGE ENDPOINT ERROR ON THE MONKAA AND DRIVING DATASETS.

Method	Monkaa		Driving	
	Clean	*Final*	*Clean*	*Final*
PWC-Net+KF2	**2.804**	**3.319**	**11.92**	**12.66**
PWC-Net+KF	2.827	3.485	12.24	12.81
PWC-Net [45]	3.077	3.508	12.92	13.72

the performance in *occ* column of Table II and Table I also validates this improvement.

Figure 7 shows the results of *bamboo_3*, *cave_3*, *market_1* and *PERTURBED_market_3* from the TEST set, without ground truth flow fields that are reserved for benchmarking though. Similarly, the first row illustrates the results of existing methods while the second row provides results by KalmanFlow. In the first column of *bamboo_3*, the flow at the boundary of the right bamboo tree is affected by the shadow effect. However, KalmanFlow can recover it from filter's previous estimations, thus producing a satisfactory flow field. The rest three columns in Figure 4 also shows the advantage of KalmanFlow, such as the area over the head of the girl in *cave_3*, the leg of the running girl in *market_1*.

B. Effectiveness of Context-Aware KalmanFlow2.0

Our improved method KalmanFlow2.0 dynamically adjusts the system noise according to the contextual similarity of neighboring frames. The quantitative results for DCFlow+KF2 and PWC-Net+KF2 are presented in Table III and IV. The DCFlow+KF generally performs slightly inferior to the original DCFlow method. For example, on the *Final* pass of the TRAINING set, DCFlow has an EPE of 3.353 for *all* region while DCFlow+KF gets higher error of 3.382. When we look into the *occ* and *non* regions, the problems turns out that the occluded regions are not well filtered by our KalmanFlow method that it deteriorate the EPE from 21.81 to 22.61. Similar phenomenon can be found in the *Clean* pass of the TRAINING set. However, when we perform the context-aware Kalman filtering for optical flow, we make substantial improvements in both the occluded and non-occluded regions. The EPEs of *occ* and *non* regions are reduced from 21.81 to 21.09 and 1.899 to 1.842 respectively. Other regions such as *d0-10*, *d10-60*, *s40+* are also refined. Consequently, KalmanFlow2.0 improves the *all* region of TRAINING set's *Final* pass from 3.353 to 3.249. In the *Final* pass of TEST set, DCFlow+KF2 also reduces the EPE of *all* region from 5.119 to 5.067.

We find that this phenomenon is caused by that when the occluded regions undergo an abrupt change of local context, the filters are usually not suitable to be adapted to make prior estimation for the upcoming time steps. In such cases, it is better to reduce the contributions of the prior estimation. Kalman filtering is intrinsically able to adjust the system noise level as needed. The patches containing contextual information are represented with learned image features that are compact and invariant to the visual appearance. We show the visual images of the typical cases in Figure 8 and 9 to explain this phenomenon in an intuitive way. Figure 8 shows a man with heavy beard moving his head, the KalmanFlow and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2903656, IEEE Transactions on Image Processing

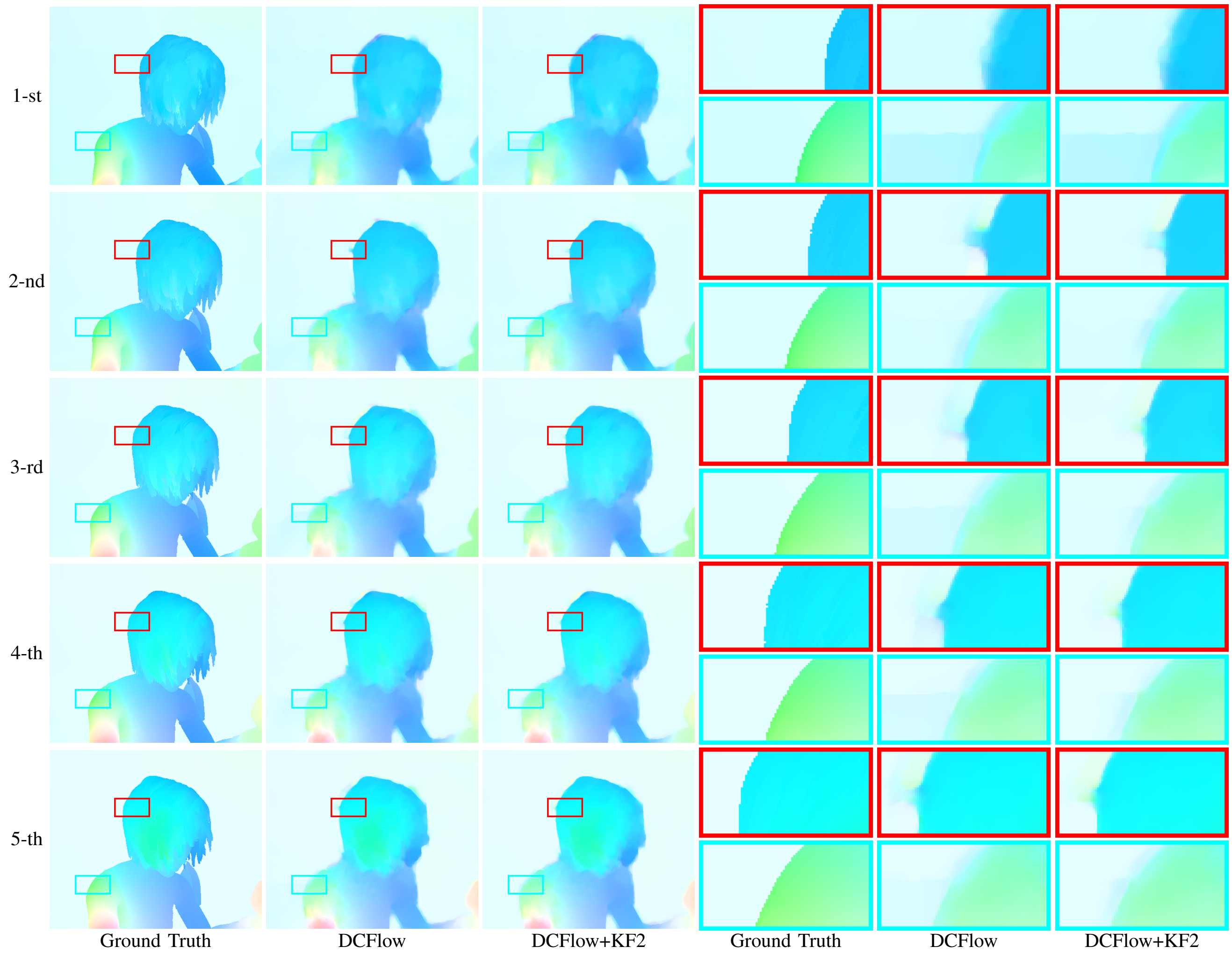


Fig. 11: Visual comparisons on the 1-st to 5-th optical flow field of *alley_1* sequence.

KalmanFlow2.0 can both filter out some of the outliers near the eyebrow (in the enlarged cyan box). But the outliers near the beard (in the enlarged red box) are not well eliminated by KalmanFlow. It's due to that the changed textures in the dark light are not well distinguished by KalmanFlow. But with the context information used in KalmanFlow2.0, the outliers are recognized and can be removed significantly. Similar cases are found in Figure 9, where the regions near the waving wing are taken good care of by the proposed context-aware Kalman filters.

We note that on the *Clean* images of the MPI Sintel TEST set, the DCFlow enhanced with our KF method performs inferior to original results. While we aim to maintain a unified Kalman filtering framework for all baseline algorithms and datasets, the controlling parameters including the γ for warping error, β for spatial smoothness and τ for temporal coherence are set to be fixed. However, the different rendering effects in *Final* and *Clean* passes may require us to give various settings for their best performances. Another intuition for the performance loss might be due to that the optimal parameter setting tuned on the TRAINING set may not works best for the TEST set because there exists a mismatch of the data distribution between the two sets, which has also been indicated by Mayer *et al.* [59]. We carry out experiments on the Monkaa, and Driving dataset [59] and present the quantitative results in the Table V. Moreover, we add Figure 10 to show the qualitative comparisons on the Monkaa dataset. Our method can remove some of the outliers in the estimated flow fields of this dataset. It shows the effectiveness of our approach for the other benchmarks with multi-frame optical flow annotations.

C. Video's Optical Flow Fields

We present a comprehensive comparison of the multiple consecutive flow fields processed by our methods. In Figure 11, there are optical flow fields from 5 consecutive time steps in the *alley_1* sequence. Each row represents a time step correspondingly. At $t = 1$, namely in the first row, both DCFlow and KalmanFlow2.0 generate some outliers above the head (in red box) and behind the arm (in cyan box). Since the KalmanFlow2.0 filters are initialized at this time step and take no prior information, the outliers are not removed. The enlarged views of the two regions are shown in the 4-th to 6-th column. Then at $t = 2$, DCFlow generates fewer outliers behind the arm, while our KalmanFlow2.0 eliminates them more thoroughly. For the outliers above the head, KalmanFlow2.0 also exclude them substantially. At time step $t = 3, 4, 5$, KalmanFlow2.0 produces better results than

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2903656, IEEE Transactions on Image Processing

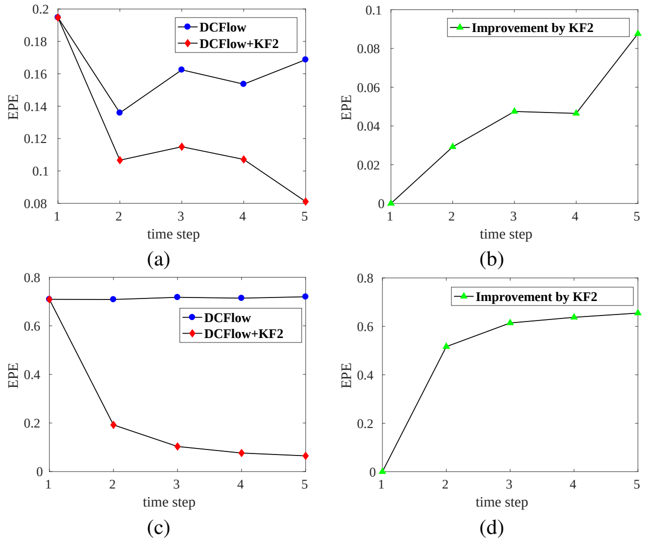


Fig. 12: The performance improvement by KalmanFlow2.0 that exploits temporal coherence. (a) and (c) presents the results of DCFlow and DCFlow+KF2 on the tracked regions of *alley_1*. (b) and (d) presents the improvements made by KF2.

TABLE VI: EndPoint Error with different parameter settings on the LDOF+KF2 algorithm.

Parameters			EPE		
γ	β	τ	*all*	*non*	*occ*
0.1	0.04	0.06	4.5953	1.9673	**27.3986**
0.1	0.04	0.18	4.7426	2.1168	27.5267
0.1	0.12	0.02	4.5697	1.9260	27.5094
0.1	0.12	0.06	4.6083	1.9662	27.5335
0.1	0.12	0.18	4.7137	2.0745	27.6146
0.1	0.30	0.02	**4.5651**	1.9096	27.6072
0.1	0.30	0.06	4.5712	1.9251	27.5319
0.1	0.30	0.18	4.6362	1.9839	27.6515
0.2	0.12	0.02	4.5832	1.9195	27.6962
0.2	0.12	0.06	4.6091	1.9524	27.6622
0.2	0.12	0.18	4.6917	2.0469	27.6413
0.2	0.30	0.02	4.5878	**1.9057**	27.8609
0.2	0.30	0.06	4.5919	1.9163	27.8087
0.2	0.30	0.18	4.6309	1.9635	27.7771
LDOF			5.9216	2.3004	37.3438

the original DCFlow method. The advantage is attributed to the use of temporal coherence, with the prior information passed from the past time steps, the outliers occurred behind the arms and the head is corrected effectively.

For the two regions in red and cyan boxes, we plot graphs to illustrate their EPEs with respect to time in Figure 12 (a) and (c). We also show the improvements made by KF2 across time in (b) and (d). With time increasing and more frames being captured, the performance improvements by our Kalman filter are becoming substantial and stable.

D. Parameter Sensitivity Analysis

We conduct experiments to test the parameter sensitivity in our KalmanFlow2.0. Since γ, β and τ balance the data warping error, spatial gradient and temporal derivative in the evaluation of noise variance, different settings are studied. Table VI presents some of the results based on the sequences of *alley_1*, *ambush_2* and *market_6* in the *Clean* set. And LDOF [56] algorithm is chosen as the measuring tool for Kalman filters. The three sequences have various motions such as small and large displacements, global motions which enable them to be good representatives for the whole set.

From this table, we find the variation of parameter γ (0.1 to 0.2) for the data term has little effect on the EPE. By contrast, EPE is more sensitive to the parameter β and τ. A smaller τ tends to give better results on *occ* pixel regions while larger β is beneficial for the *non* pixel regions. As highlighted in the table, the best result is obtained for *all* pixel regions when $\gamma = 0.1$, $\beta = 0.30$, $\tau = 0.02$. Overall, by the given parameter settings that have been examined, the EPE changes within 0.2 pixels for *all* pixel regions and is overall lower than the baseline LDOF algorithm. The proposed KalmanFlow2.0 is robust to different parameter settings in a reasonable range. However, it is also noticed that with different baseline measurement methods and datasets, these parameters have to be adjusted to reach best performances.

VI. Conclusions

In this paper, we propose a novel optical flow estimation framework for video sequences. It employs the temporal coherence in videos through Kalman filtering. The proposed KalmanFlow and the context-aware KalmanFlow2.0 can improve existing state-of-the-art optical flow algorithms. In the evaluation for the flow field's noise, we introduce the temporal derivative term besides data and spatial smoothness terms, enabling the Kalman filter to produce more consistent results in time domain. We use the context-aware system noise to make the Kalman filters robust to the abrupt change and occlusion/dis-occlusions. We demonstrate that not only the variational framework based algorithms but also the recent state-of-the-art deep learning based ones can benefit from this temporal coherence prior.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China (61771306), Natural Science Foundation of Shanghai (18ZR1418100), Chinese National Key S&T Special Program (2013ZX01033001-002-002), Shanghai Key Laboratory of Digital Media Processing and Transmissions (STCSM 18DZ2270700).

References

[1] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *Acm computing surveys (CSUR)*, vol. 38, no. 4, p. 13, 2006.

[3] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2141–2148.

[4] L. L. Rakêt, L. Roholm, A. Bruhn, and J. Weickert, "Motion compensated frame interpolation with a symmetric optical flow constraint," in *International Symposium on Visual Computing*, 2012, pp. 447–457.

[5] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[6] A. Ayvaci, M. Raptis, and S. Soatto, "Sparse occlusion detection with optical flow," *International Journal of Computer Vision*, vol. 97, no. 3, pp. 322–338, 2012.

[7] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European Conference on Computer Vision*, 2004, pp. 25–36.

[8] D. Gadot and L. Wolf, "Patchbatch: a batch augmented loss for optical flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4236–4245.

[9] W.-S. Lai, J.-B. Huang, and M.-H. Yang, "Semi-supervised learning for optical flow with generative adversarial networks," in *Neural Information Processing Systems*, 2017, pp. 353–363.

[10] M. Werlberger, T. Pock, and H. Bischof, "Motion estimation with non-local total variation regularization," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2464–2471.

[11] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 978–994, 2011.

[12] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1744–1757, 2012.

[13] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[14] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.

[15] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu, "Large displacement optical flow from nearest neighbor fields," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2443–2450.

[16] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conference on Computer Vision*, 2012, pp. 611–625.

[17] Y. Li, D. Min, M. S. Brown, M. N. Do, and J. Lu, "SPM-BP: Sped-up PatchMatch Belief Propagation for Continuous MRFs," in *IEEE International Conference on Computer Vision*, 2015, pp. 4006–4014.

[18] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," in *IEEE International Conference on Computer Vision*, 2015, pp. 4015–4023.

[19] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 28, no. 3, pp. 1–11, 2009.

[20] A. Dosovitskiy, P. Fischery, E. Ilg, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, T. Brox *et al.*, "Flownet: Learning optical flow with convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[21] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1647–1655.

[22] T. M. Chin, W. C. Karl, and A. S. Willsky, "Probabilistic and sequential computation of optical flow using temporal coherence," *IEEE Transactions on Image Processing*, vol. 3, no. 6, pp. 773–788, 1994.

[23] P. Elad and A. Feuer, "Recursive optical flow estimation-adaptive filtering approach," in *Electrical and Electronics Engineers in Israel*, 1996.

[24] J. Weickert and C. Schnörr, "Variational optic flow computation with a spatio-temporal smoothness constraint," *Journal of mathematical imaging and vision*, vol. 14, no. 3, pp. 245–255, 2001.

[25] S. Volz, A. Bruhn, L. Valgaerts, and H. Zimmer, "Modeling temporal coherence for optical flow," in *IEEE International Conference on Computer Vision*, 2011, pp. 1116–1123.

[26] H. Zimmer, A. Bruhn, and J. Weickert, "Optic flow in harmony," *International Journal of Computer Vision*, vol. 93, no. 3, pp. 368–388, 2011.

[27] D. Sun, E. B. Sudderth, and M. J. Black, "Layered segmentation and optical flow estimation over time," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1768–1775.

[28] R. Kennedy and C. J. Taylor, "Optical flow with geometric occlusion estimation and fusion of multiple frames," in *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2015, pp. 364–377.

[29] D. Sun, E. B. Sudderth, and M. J. Black, "Layered image motion with explicit occlusions, temporal consistency, and depth ordering," in *Neural Information Processing Systems*, 2010, pp. 2226–2234.

[30] D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation: a survey," *Computer Vision and Image Understanding*, vol. 134, pp. 1–21, 2015.

[31] W. Bao, Y. Xiao, L. Chen, and Z. Gao, "Kalmanflow: Efficient kalman filtering for video optical flow," in *IEEE International Conference on Image Processing*, 2018, pp. 3343–3347.

[32] J. Xu, R. Ranftl, and V. Koltun, "Accurate Optical Flow

via Direct Cost Volume Processing," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1289–1297.

[33] N. Monzón, A. Salgado, and J. Sánchez, "Regularization strategies for discontinuity-preserving optical flow methods," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1580–1591, 2016.

[34] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 115–137, 2014.

[35] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Seventh International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[36] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Epicflow: Edge-preserving interpolation of correspondences for optical flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1164–1172.

[37] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, pp. 504–511, 2013.

[38] Q. Chen and V. Koltun, "Full flow: Optical flow estimation by global optimization over regular grids," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4706–4714.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012, pp. 1097–1105.

[40] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *IEEE International Conference on Computer Vision*, 2013, pp. 1385–1392.

[41] C. Bailer, K. Varanasi, and D. Stricker, "CNN-based patch matching for optical flow with thresholded hinge embedding loss," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3250–3259.

[42] A. Ahmadi and I. Patras, "Unsupervised convolutional neural networks for motion estimation," in *IEEE International Conference on Image Processing*, 2016, pp. 1629–1633.

[43] Z. Ren, J. Yan, B. Ni, B. Liu, X. Yang, and H. Zha, "Unsupervised deep learning for optical flow estimation." in *Association for the Advancement of Artificial Intelligence*, 2017, pp. 1495–1501.

[44] Y. Wang, Y. Yang, Z. Yang, L. Zhao, and W. Xu, "Occlusion aware unsupervised learning of optical flow," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4884–4893.

[45] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: Cnns for optical flow using pyramid, warping, and cost volume," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[46] Z. Ren, O. Gallo, D. Sun, M.-H. Yang, E. B. Sudderth, and J. Kautz, "A fusion approach for multi-frame optical flow estimation," in *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[47] D. Maurer and A. Bruhn, "Proflow: Learning to predict optical flow," in *British Machine Vision Conference*, 2018, pp. 1–13.

[48] J. Janai, F. Guney, A. Ranjan, M. Black, and A. Geiger, "Unsupervised learning of multi-frame optical flow with occlusions," in *European Conference on Computer Vision*, 2018, pp. 690–706.

[49] D. Maurer, M. Stoll, and A. Bruhn, "Directional priors for multi-frame optical flow," in *British Machine Vision Conference*, 2018, pp. 1–13.

[50] M. J. Black, "Recursive non-linear estimation of discontinuous flow fields," in *European Conference on Computer Vision*, 1994, pp. 138–145.

[51] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[52] X. Song, L. D. Seneviratne, and K. Althoefer, "A kalman filter-integrated optical flow method for velocity sensing of mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 3, pp. 551–563, 2011.

[53] Y. Motai, S. K. Jha, and D. Kruse, "Human tracking from a mobile agent: optical flow and kalman filter arbitration," *Signal Processing: Image Communication*, vol. 27, no. 1, pp. 83–95, 2012.

[54] S.-K. Weng, C.-M. Kuo, and S.-K. Tu, "Video object tracking using adaptive kalman filter," *Journal of Visual Communication and Image Representation*, vol. 17, no. 6, pp. 1190–1208, 2006.

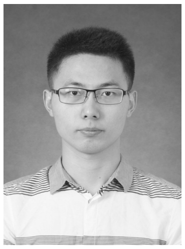
[55] W.-N. Lie and Z.-W. Gao, "Video error concealment by integrating greedy suboptimization and kalman filtering techniques," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 8, pp. 982–992, 2006.

[56] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.

[57] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

[58] N. Monzón, A. Salgado, and J. Sánchez, "Robust Discontinuity Preserving Optical Flow Methods," *Image Processing On Line*, vol. 6, pp. 165–182, 2016, https://doi.org/10.5201/ipol.2016.172.

[59] N. Mayer, E. Ilg, P. Fischer, C. Hazirbas, D. Cremers, A. Dosovitskiy, and T. Brox, "What makes good synthetic training data for learning disparity and optical flow estimation?" *International Journal of Computer Vision*, pp. 1–19, 2018.



Wenbo Bao recieved the B.S. degree in electronic information engineering from Huazhong University of Science and Technology, Hubei, China, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering with the Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include computer vision, machine learning, and video processing.

Xiaoyun Zhang received the B.S. and M.S. degrees in applied mathematics from Xian Jiaotong University in 1998 and 2001, respectively, and the Ph.D. degree in pattern recognition from Shanghai Jiao Tong University, China, in 2004. Her Ph.D. thesis has been nominated as National 100 Best Ph.D. Theses of China. Her research interests include computer vision and pattern recognition, image and video processing, digital TV system. Her current research focuses on image processing and video compression.

Li Chen received the B.S. and M.S. degrees from Northwestern Polytechnical University, Xian, China, in 1998 and 2000, respectively, and the Ph.D. degree from Shanghai Jiao Tong University, China, in 2006, all in electrical engineering. His current research interests include image and video processing, DSP and VLSI for image, and video processing.

Zhiyong Gao received the B.S. and M.S. degrees in electrical engineering from the Changsha Institute of Technology, Changsha, China, in 1981 and 1984, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1989. From 1994 to 2010, he took several senior technical positions in England, including a Principal Engineer with Snell and Wilcox, Petersfield, U.K., from 1995 to 2000, a Video Architect with 3DLabs, Egham, U.K., from 2000 to 2001, a Consultant Engineer with Sony European Semiconductor Design Center, Basingstoke, U.K., from 2001 to 2004, and a Digital Video Architect with Imagination Technologies, Kings Langley, U.K., from 2004 to 2010. Since 2010, he has been a Professor with Shanghai Jiao Tong University. His research interests include video processing and its implementation, video coding, digital TV, and broadcasting.