Recurrent Spatial Pyramid CNN for Optical Flow Estimation

Ping Hu, Gang Wang, *Senior Member, IEEE*, and Yap-Peng Tan, *Senior Member, IEEE*

***Abstract*—Optical flow estimation plays an important role in many multimedia and computer vision tasks. Although great progress has been made in applying convolutional neural networks (CNNs) to estimate optical flow in recent works, it is still difficult for CNNs to generate optical flow with the desired effectiveness and efficiency. Compared to CNN-based methods, conventional variational methods normally perform to optimize an energy function and produce optical flow with more precise details. Inspired by the effectiveness of variational methods and deep CNNs, we propose a recurrent spatial pyramid (RecSPy) network for optical flow estimation. To deal with large displacements and to decrease the number of parameters, we formulate the spatial pyramid as a recurrent process, and adopt a CNN to refine optical flow at each spatial scale. Furthermore, to improve the results with more precise details, we propose an energy function that encodes structure and constancy constraints to help refine the optical flow at each spatial scale. The combination of the proposed RecSPy network and the proposed energy-based refinement enables our system to estimate optical flow effectively and efficiently. Experimental results on the benchmarks validate the effectiveness and efficiency of the proposed method.**

***Index Terms*—Optical flow estimation, convolutional neural network, coarse-to-fine refinement.**

I. Introduction

THE wide spread of multimedia systems with video displays accentuates the importance of several computer vision problems. Optical flow estimation, which provides the fundamental visual information, plays a key role in these applications. For example, optical flow is widely applied in tasks like action recognition [1]–[3], video content analysis [4]–[6], video compression [7], video index and retrieval [8], [9], 2D-3D conversion [10], video editing [11], and so on.

The basic definition of optical flow is the displacement of intensity patterns. This notion is based on the assumption that the intensities of moving pixels remain unchanged during motion. However, directly estimating the correspondences by matching pixels of similar intensity is a classical ill-posed problem since the constraints cannot guarantee a unique solution. To address the problem, in 1981 Horn and Schunck [12] firstly proposed to impose additional constraints on the optical flow, and solved it with a variational method. Since then, variational energy minimization has become one of the most popular frameworks for motion field estimation, and many subsequent extensions and improvements have been made [13]–[19]. Variational methods succeed because they allow additional assumptions to constrain the ill-posed problem. However, the optimization cannot guarantee the optimal solution and this type of models cannot handle well large displacements since they adapt optical flow locally. Hence, hierarchical schemes are adopted to improve the optical flow field in a coarse-to-fine way based on the spatial pyramid [17], [20]–[24]. The hierarchical matching scheme has been shown to be effective in avoiding poor local minima for large displacements and efficient at convergence. At each scale, variational optimization or approximate nearest neighbor fields (ANNF) can be applied to refine the coarse flow produced by the previous scale. However, both of them rely on the initial flow field and the discriminative feature matching. If the optical flow is inaccurate at coarse scales or the features for matching cannot discriminate different image patches, this kind of methods may fail.

Recently, deep convolutional neural networks (CNN) has succeeded in many computer vision and multimedia tasks [25]. In the field of optical flow estimation, several works [24], [26]–[28] also have explored CNN, and their performance shows that applying CNN to learn optical flow is a promising direction. Compared to traditional methods, CNN itself is a hierarchical structure and has the superior ability to learn features of different scales from data. However, the CNN in previous methods usually have a large number of parameters and are typically trained via gradient descent with L1 or L2 loss, which treat pixels in the dense flow field as independent samples from an identical distribution [29]. While in the context of optical flow estimation, pixels in the images are highly correlated with each other both spatially and temporally; therefore optical flow directly generated by CNN may lose details of the motion structures. An example is shown in Fig. 1(c).

In this work, to alleviate these limitations and train a CNN to produce more accurate optical flow, we propose a recurrent spatial pyramid CNN, which is denoted as $RecSPyNet$. The network has two components, a Siamese network that generates initial optical flow at a small spatial scale, and a refinement CNN that upscales and refines optical flow at each scale of the

Manuscript received August 17, 2017; revised November 26, 2017 and January 13, 2018; accepted February 18, 2018. Date of publication March 15, 2018; date of current version September 18, 2018. This research was carried out at the Rapid-Rich Object Search (ROSE) Lab at the Nanyang Technological University, Singapore. The ROSE Lab is supported by the National Research Foundation, Singapore, and the Infocomm Media Development Authority, Singapore. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Fatih Porikli. *(Corresponding author: Ping Hu.)*

P. Hu and Y.-P. Tan are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: phu005@ntu.edu.sg; eyptan@ntu.edu.sg).

G. Wang is with the Alibaba Group, Hangzhou 310052, China (e-mail: gangwang6@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2815784

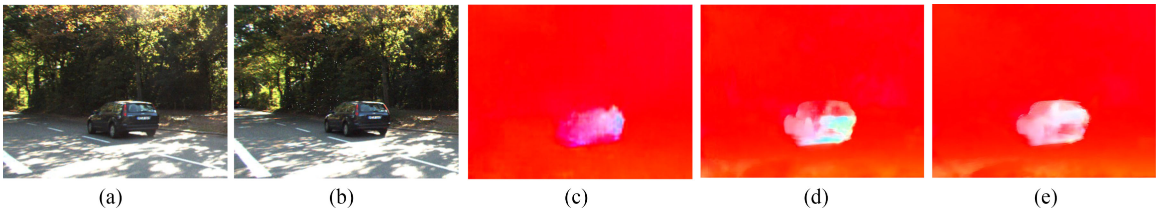


Fig. 1. Comparison between methods. (a) and (b) are the image pair. (c) The result of the CNN network *FlowNetS* [26]. (d) The result of our recurrent spatial pyramid CNN, denoted as RecSPy. (e) The result of RecSPy with proposed refinement.

pyramid and finally generates a full-resolution output. Since the initial flow plays a very important role, we use a Siamese network to extract features from input images separately and then fuse these features to estimate the initial flow. The initial flow is used to warp the input, and then the refinement CNN is utilized to process the warped images and generates refined optical flow which is then used as initialization at next scale of the spatial pyramid. In the Siamese network and the recurrent CNN, parameters are reused and thus our model is very small and efficient.

To improve the optical flow with more details, we also propose an energy function that encodes constancy constraint and structure constraint. Refinement based on this energy function is performed at each scale of the spatial pyramid. Constancy constraint is a widely adopted method for optical flow estimation [12], [20], [21]. For fast computation, we only use data term and smoothness term [20]. Another challenge faced by both variational methods and CNNs is the motion discontinuity. Both of them find difficulty in estimating optical flow accurately at the motion boundaries. Since the motion discontinuity usually occurs at object boundaries, the edges have been used as an extra information to improve the optic flow [24], [30]. In this paper we propose to impose a structure regularity on the refinement. Inspired by the bilateral filter [31], [32], we include an edge-preserving term into the energy function. The edge-preserving term serves as a local constraint on the optical flow. It adaptively utilizes the input images and avoids extracting edges explicitly. The overall energy can be easily optimized with existing techniques. Results of the proposed network and refinement are shown in Fig. 1(d) and (e)

In summary, we make the following three following contributions:

- We formulate the hierarchical optical flow estimation as a recurrent spatial pyramid CNN. The proposed network has a small number of parameters and runs efficiently.
- We proposed an energy function that imposes constancy and structure constraints to refine the optical flow. The energy function can be readily applied to improve other algorithms for motion field estimation.
- By combining variational method with deep learning, we present a recurrent spatial pyramid CNN that estimates optical flow effectively and efficiently.

II. Related Work

Research on optical flow estimation has a long history over 30 years since the pioneering work of Horn and Schunck [12] as well as Lucas and Kanade [33]. In this section, we briefly overview the recent developments.

Dense optical flow estimation is an ill-posed problem since the flow at a pixel has to two components. Variational methods tackle this by optimizing an objective function composed of extra constraints. The earliest original model [12] only has a data term that imposes intensity consistency and a smoothness term that encodes global smoothness of the optical field. Based on this model, many extensions have been made. Non-quadratic regularisers are introduced to deal with discontinuities and occlusions [13], [14]. Gradient consistency [15], [16] and photometric invariant model [34] are applied to tackle the change of pixel intensity. Dense descriptor matching is adopted into the energy function for robustness [17], [18], [20], [21]. Local constraints are imposed for motion details [35], [36] . Optimization on regular grids [37] is proposed to efficiently generate a good initial flow for further refinement. MRF-based methods [38]–[41] also treat optical flow estimation as an optimization problem and solve it using probability models. Optimization-based models are effective but they still face the difficulty of finding an optimal solution. Another popular way to compute optical flow is utilizing the nearest neighbor field (NNF) [22], [42]–[45]. NNF methods target at searching for the nearest neighbors from the next image for given regions in the reference image. When applied to dense optical flow estimation, despite the computation complexity, NNF methods have the challenge of mismatching and outliers. To avoid poor local minima and deal with large displacement, the hierarchical matching scheme has been widely adopted in optical flow estimation. When the optical flow is passed from the coarsest scale to finest scale, variational methods [16], [17], [20], [21], [23], [35] or NNF methods [22], [44], [46] can be applied at each scale to refine the flow fields coming from coarser scale. Traditional methods achieved a great development in recent years; however, researchers still need to balance between high effectiveness and high efficiency when designing an algorithm.

Along with the success of deep learning in other computer vision tasks, several attempts have been made in the field of optical flow estimation. Dosovitskiy *et al.* [26] present the Flownet which is the first deep CNN model for optical flow estimation. The Flownet is trained on the artificial Flying Chairs dataset and achieves competitive results. Ilg *et al.* [27] propose a well-designed network based on Flownet to accurately estimate both large displacements and small displacements. The network comprises several sub-networks and achieves state-of-the-art performance. Bai *et al.* [47] adopt CNN to separate vehicles and background and then estimate optical flow for them separately. Jason *et al.* [48] make use of differentiable warping [49] to impose brightness constancy and motion smoothness so that the network is able to learn end-to-end unsupervised optical

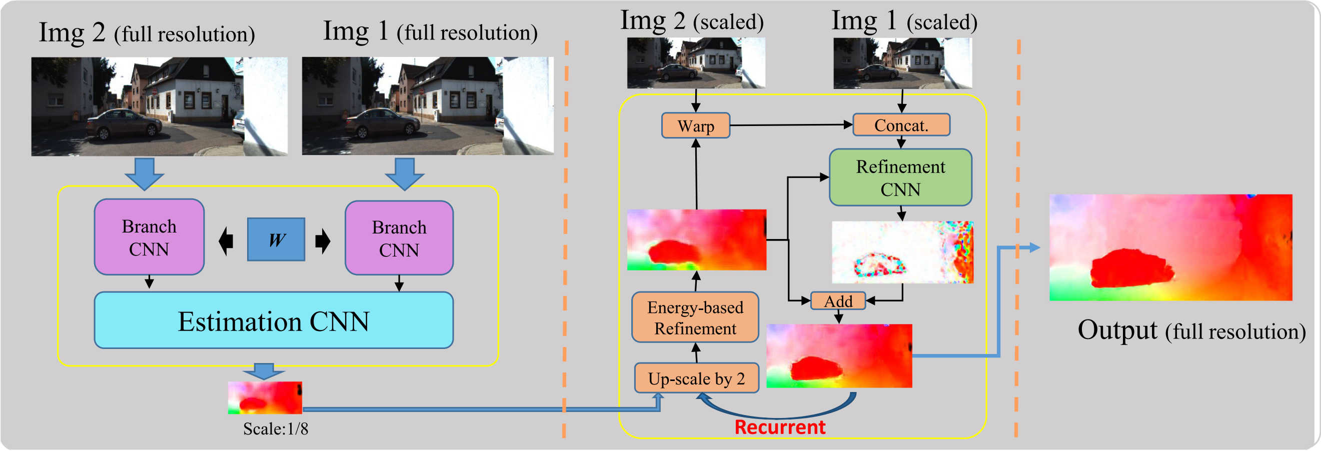


Fig. 2. System overview. The proposed network is composed of a Siamese network and a recurrent spatial pyramid CNN. The Siamese network takes full-resolution images as inputs and generates an initial optical flow of 1/8 full resolution. The initial flow was upscaled and refined in the spatial pyramid recurrently. At each scale of the pyramid, the flow from the previous scale is first scaled up and refined by our proposed energy function. Then the second image in the pair is warped by the flow and the image pair is processed by a CNN to compute a residual flow. The input flow and the residual flow are added together to be the refined output of current scale. With this recurrent operation on the spatial pyramid, the initial optical flow generated by the Siamese network is finally converted to be the refined flow of full resolution.

flow. Spatial pyramid network is also utilized by Ranjan *et al.* [24]. Bailer *et al.* [50] and Schuster *et al.* [51] both perform CNN based patch matching to estimate the optical flow. Zweig *et al.* [52] utilize a fully convolutional network to interpolate sparse optical flow into dense one. In this paper, we further formulate the spatial pyramid network with recurrent convolution to greatly decrease the model size. We also propose a novel energy function that encodes constancy constraint and structure constraint to improve the results. Compared to conventional methods for optical flow, deep neural networks are promising due to the superior ability to learn representative features from data and the efficiency of computation.

III. Network for Optical Flow

In this section, we first introduce the proposed recurrent spatial pyramid network, and then discuss the proposed energy function that encodes constancy constraint and structure constraint to help refine optical flow. The architecture of the proposed network is shown in Fig. 2.

A. Recurrent Spatial Pyramid Network

Spatial pyramid has been frequently applied in optical flow estimation [20], [23], [24] to cope with large displacements. The idea is to downsample the input images to different spatial scales. At a coarse spatial scale, large displacements become small and can be estimated more easily. For each scale, optical flow generated by previous coarser scale is applied to warp the input images, and then a small motion increment is estimated based on the warped image pair. As a result, the optical flow is iteratively upscaled and refined from the smallest scale to full resolution. In this work, we combine this framework with deep learning to achieve an effective and efficient model.

1) Generating Initial Optical Flow: When estimating optical flow with the spatial pyramid, the initial flow at the coarsest scale plays an important role. To generate an accurate initial flow, we utilize a Siamese network as shown in the left part of Fig. 2. Instead of estimating optical flow from the original images, the branch CNN in Siamese network first extracts features from the original inputs separately, and then a decision CNN operates on the extracted features and estimates the initial flow. To effectively extract features for optical flow estimation and reduce the model size, the branch CNN is built up with the initial block and bottleneck block as used in [53]. As shown in Table I, the branch CNN contains three scales. The initial scale is a single block as presented in Fig. 3(a). The second and third scales share the same structure, which is composed of four bottlenecks. As the structure shown in Fig. 3(b). There are three convolution layers in the bottleneck. The first and third 1 × 1 convolutional layers respectively reduce and expand the dimensionality. The second convolution layer has different types. In this work, regular convolution and asymmetric convolution (a sequence of 5 × 1 and 1 × 5 convolution) are used and the respective bottlenecks are denoted by $regular$ and $asymmetric$. All convolutional layers are followed by Batch Normalization and PReLU. The branch CNN converts the input of $3 \times H \times W$ to be a feature matrix of $32 \times \frac{H}{8} \times \frac{W}{8}$.

TABLE I
Architecture of the Branch CNN

Name	Type	Output Scale
Input		$384 \times 512 \times 3$
Initial Block		$192 \times 256 \times 16$
bottleneck1.0	downsampling	$96 \times 128 \times 32$
bottleneck1.1	regular	$96 \times 128 \times 32$
bottleneck1.2	asymmtetric 5	$96 \times 128 \times 32$
bottleneck1.3	regular	$96 \times 128 \times 32$
bottleneck2.0	downsampling	$48 \times 64 \times 32$
bottleneck2.1	regular	$48 \times 64 \times 32$
bottleneck2.2	asymmtetric 5	$48 \times 64 \times 32$
bottleneck2.3	regular	$48 \times 64 \times 32$

Output size are given for an input of $384 * 512$.

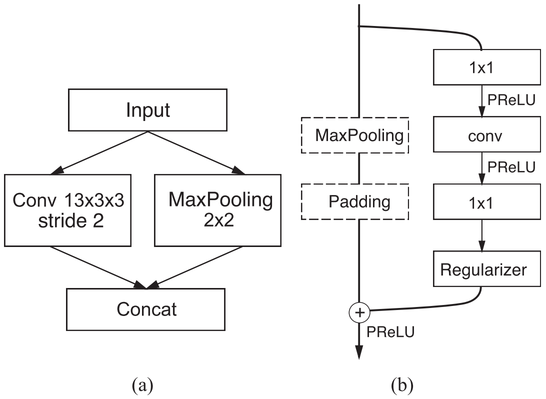


Fig. 3. (a) Initial block. It takes inputs of 3 channels and out 16 feature maps with size scaled by 0.5. (b) Bottleneck. $conv$ is the type of bottleneck.

TABLE II
ARCHITECTURE OF THE ESTIMATION CNN (LEFT) AND REFINEMENT CNN (RIGHT)

Estmation CNN	
Layer	Output Scale
Input	$48 \times 64 \times 64$
conv $7 \times 7 \times 64$	$48 \times 64 \times 64$
conv $7 \times 7 \times 32$	$48 \times 64 \times 32$
conv $7 \times 7 \times 32$	$48 \times 64 \times 32$
conv $7 \times 7 \times 16$	$48 \times 64 \times 16$
conv $7 \times 7 \times 2$	$48 \times 64 \times 2$
Refinement CNN	
Layer	Output Scale
Input	$\mathrm{w} \times \mathrm{h} \times 8$
conv $7 \times 7 \times 64$	$\mathrm{w} \times \mathrm{h} \times 64$
conv $7 \times 7 \times 32$	$\mathrm{w} \times \mathrm{h} \times 32$
conv $7 \times 7 \times 32$	$\mathrm{w} \times \mathrm{h} \times 32$
conv $7 \times 7 \times 16$	$\mathrm{w} \times \mathrm{h} \times 16$
conv $7 \times 7 \times 2$	$\mathrm{w} \times \mathrm{h} \times 2$

After applying branch CNN, the extracted features of the input images are concatenated to be $64 \times \frac{H}{8} \times \frac{W}{8}$ and processed by an estimation CNN. As shown in the left part of Table II, the estimation CNN contains five convolutional layers with a kernel size of 7 × 7 and stride of 1. The number of feature maps output by each layer is 64, 32, 32, 16, 2 respectively. PReLU is placed after each convolutional layer except for the last one, whose output is the initial optical flow. At last, the estimation CNN outputs optical flow at the scale of $\frac{H}{8} \times \frac{W}{8}$, and is used as the initial flow for subsequent Recurrent Spatial Pyramid network.

The branch CNN and estimation CNN are trained jointly in a supervised setting. During training, we utilize the two-branch structure to connect the two input images to the initial optical flow as shown in the left part of Fig. 2. Since inputs go through the same feature encoding process, we only need one branch CNN. To achieve this, the two branches are constrained to be the same model, and any updates are applied to the shared parameters for both sides.

2) Recurrent Refinement With CNN: At first, we downsample the input image pair by a factor of 2 to build a spatial pyramid. The optical flow generated by the Siamese network is utilized as the initial flow at the coarsest scale of the spatial pyramid. As aforementioned that the initial flow is 1/8 the size of the original input; hence there are four scales of the pyramid and it needs 3 times of refinement to obtain optical flow of full resolution. Specifically, refinement here means to find the residual flow based on the initial optical flow generated by the previous coarser scale. We adopt deep network to estimate the motion increments. Different from [24] which trains different CNNs for different spatial levels, we propose to use one single CNN to recurrently estimate the residual flow from coarse to fine scale.

At each scale of the spatial pyramid, the optical flow generated by the last coarser scale is first resized to the current scale. Then, the second image is warped by the coarse optical flow. The warping is performed by $I_{\mathrm{warp}}(x + u_e, y + v_e) = I_{\mathrm{ori}}(x, y)$, where I_{ori} and I_{warp} are the input image and warped result respectively, $\omega = (u_e, v_e)$ is the coarse optical flow. Then the refinement CNN is applied on the warped image pair to estimate the residual flow field. And finally, the initial flow and the residual flow are added together to form the output of the current spatial scale. With the initial flow of the coarsest scale, we perform the refinement with the refinement CNN scale by scale, until reaching the full resolution. This process is depicted in the middle part of Fig. 2 by ignoring the $Energy\ based\ Refinement$ component which will be presented in the next subsection. Some results for different scales are shown in Fig. 4.

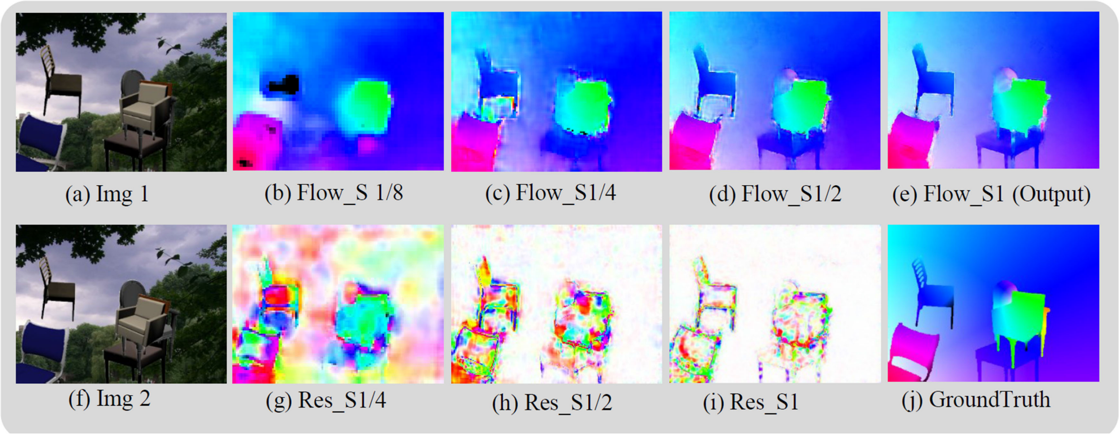
The structure of the refinement CNN is shown in the right part of Table II. It is composed of five convolutional layers with a kernel size of 7 × 7 and stride of 1. The first four convolutional layers are followed by PReLU. The refinement CNN takes the warped image pair and coarse flow as input and the numbers of feature maps for the convolutional layers are 32, 64, 32, 16, 2 respectively. To estimate residual flow at different scales, the refinement CNN was trained at different scales alternatively.

B. Refinement With Constancy and Structure Constraints

We now present the $Energy\ based\ Refinement$ component in Fig. 2 that helps to further improve the optical flow estimation with constancy constraint and structure constraint. Variational method combined with spatial pyramid is one of the popular ways to cope with large displacement. At each scale of the pyramid, an energy function that encodes matching errors and other constraints is minimized to search for an optimal motion displacement. We also adopt a similar way to further improve estimated optical flow at each scale with the proposed energy function.

Given a pair of consecutive images ($I_1, I_2 : \Omega \rightarrow \Re^c$), which are defined on Ω with c channels,we propose to estimate a 2-dimension motion field $\omega = (u, v) : \Omega \rightarrow \Re^2$ by minimizing the energy:

$$E(\omega) = \underbrace{\int_{\Omega} \delta\Psi(E_{\mathrm{int}}) + \gamma\Psi(E_{\mathrm{grad}}) + \alpha\Psi(E_{\mathrm{smooth}}) d\mathbf{x}}_{\mathrm{Constancy}} + \underbrace{\int_{\Omega} E_{\mathrm{stru}} d\mathbf{x}}_{\mathrm{Structure}} \tag{1}$$


(a) Img 1 (b) Flow_S 1/8 (c) Flow_S1/4 (d) Flow_S1/2 (e) Flow_S1 (Output)

(f) Img 2 (g) Res_S1/4 (h) Res_S1/2 (i) Res_S1 (j) GroundTruth

Fig. 4. Results at different scales of the pyramid. (b) is the initial flow produced by the Siamese network. (c)–(e) are the outputs of different scales from coarse to fine. (g)–(i) are the motion incremental estimated by the CNN from the coarsest scale to the finest. It should be noted that the flow maps and residual flow maps are resized to the same size for viewing.

where the first integration is constancy constraint and the second integration is structure constraint imposed on the flow.

1) Constancy Constraint: It is a simplification of the variational model in [21]. $\Psi(s) = \sqrt{s^2 + \epsilon^2}$ with $\epsilon = 0.001$ is a robust penalizer that suppress outlier and benefits the minimisation process. Given the brightness constancy assumption:$(\nabla_3^\top I)\omega = 0$ with $\nabla_3 = [\partial x, \partial y, \partial t]^\top$, the intensity data term is built as $E_{\text{int}} = \omega^\top \bar{J}_0 \omega$ where $\bar{J}_0 = \frac{(\nabla_3 I)(\nabla_3^\top I)}{\|\nabla_2 I\|^2 + 0.01}$. The normalization term in $\bar{J}_0$ helps to enforce brightness constancy. In a similar way, the gradient data term that penalizes gradient constancy is presented as $E_{\text{grad}} = \omega^\top \bar{J}_{xy} \omega$ with $\bar{J}_{xy} = \frac{(\nabla_3 I_{dx})(\nabla_3^\top I_{dx})}{\|\nabla_2 I_{dx}\|^2 + 0.01} + \frac{(\nabla_3 I_{dy})(\nabla_3^\top I_{dy})}{\|\nabla_2 I_{dy}\|^2 + 0.01}$, I_{dx} and I_{dy} are the derivatives of I with respect to x and y respectively. The smoothness term is defined as the gradient flow norm, which is $E_{\text{smooth}} = \|\nabla u\|^2 + \|\nabla v\|^2$. The energy function is optimised using the numerical optimization algorithm as in [15], [21] with 2 fixed-point iterations and 10 iterations of Successive Over Relaxation (SOR) algorithm [54].

2) Structure Constraint: It is based on the fact that motion discontinuities usually occur at the boundary of object edges. Therefore, refining optic flow with more accurate structure details means to smooth the optical flow while preserving edges. Inspired by bilateral filter [31], [32], we assume that optical flow ω is linearly related to the first input image I within a square window d_k centered at pixel k

$$\omega_i = a_k I_i + b_k, \forall i \in d_k \tag{2}$$

where a_k and b_k are constants adaptively decided by the window d_k. The linear relationship ensures that the motion discontinuities occur at the edges of I with $\nabla \omega = a \nabla I$. To drive the optical flow has more structure details, we minimize the structure energy,

$$E_{\text{stru}} = \sum_{i \in \omega_k} \left((a_k I_i + b_k - \omega_i)^2 + \tau a_k^2 \right) \tag{3}$$

where τa_k^2 is a regularization that constrains on the value of a_k. To minimize this energy, guided filtering [32] is adopted. By applying linear regression, we obtain $a_k = \frac{\frac{1}{|d|} \Sigma_{i \in d_k} I_i \omega_i - \frac{\mu_k}{|d|} \Sigma_{i \in d_k} \omega_i}{\sigma_k^2 + \epsilon}$ and $b_k = \frac{1}{|d|} \Sigma_{i \in d_k} \omega_i - a_k \mu_k$, where μ_k and σ_k^2 are the mean and variance of I in window d_k, $|d|$ is the number of pixels in d_k. Then by applying (2) to the entire image and taking average at each pixel, the optimised optical flow can be written as, $\omega_i' = \frac{1}{|d|} \Sigma_{k: i \in d_k} a_k I_i + b_k$. As proved in [32], the expression can be finally rewritten as,

$$\omega_i' = \Sigma_j W_{ij}(I) \omega_j \tag{4}$$

with

$$W_{ij}(I) = \frac{1}{|d|^2} \sum_{k:(i,j) \in d_k} \left(1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon} \right) \tag{5}$$

where μ_k and σ_k^2 are the mean and variance of I in window d_k, $|d|$ is the number of pixels in d_k.

When applying the proposed energy function to refine optical flow, we perform the optimization only once on current scale. The constancy energy and structure energy are optimized alternatively.

IV. Experiments

A. Network Training

The architecture of the proposed system is shown in Fig. 2. The model is composed of two components which are the Siamese network for initial flow at the smallest scale and a CNN for recurrent refinement. These two networks are pre-trained separately on the Flying Chair dataset [26]. The two-tower Siamese network is constrained to share parameters at the two branches and trained jointly. During training at each scale, we first perform mean subtraction for input images and then directly drive the network to minimize the endpoint error. The loss function is $L(w_e, w_g) = \sqrt{(u_e - u_g)^2 + (v_e - v_g)^2}$, where $w_e = (u_e, v_e)$ and $w_g = (u_g, v_g)$ are the estimated optical flow and groundtruth respectively. To train the Siamese network, Adam [56] method is adopted for optimization with $\beta_1 = 0.9$ and $\beta_2 = 0.99$. We use a batch size of 16 with 1000 iterations per epoch. The learning is set as 2×10^{-4} for the first 10 epochs, 1×10^{-4} for the 11th to the 60th epochs, and 1×10^{-5} for the rest epochs until converge. The CNN for recurrent refinement is

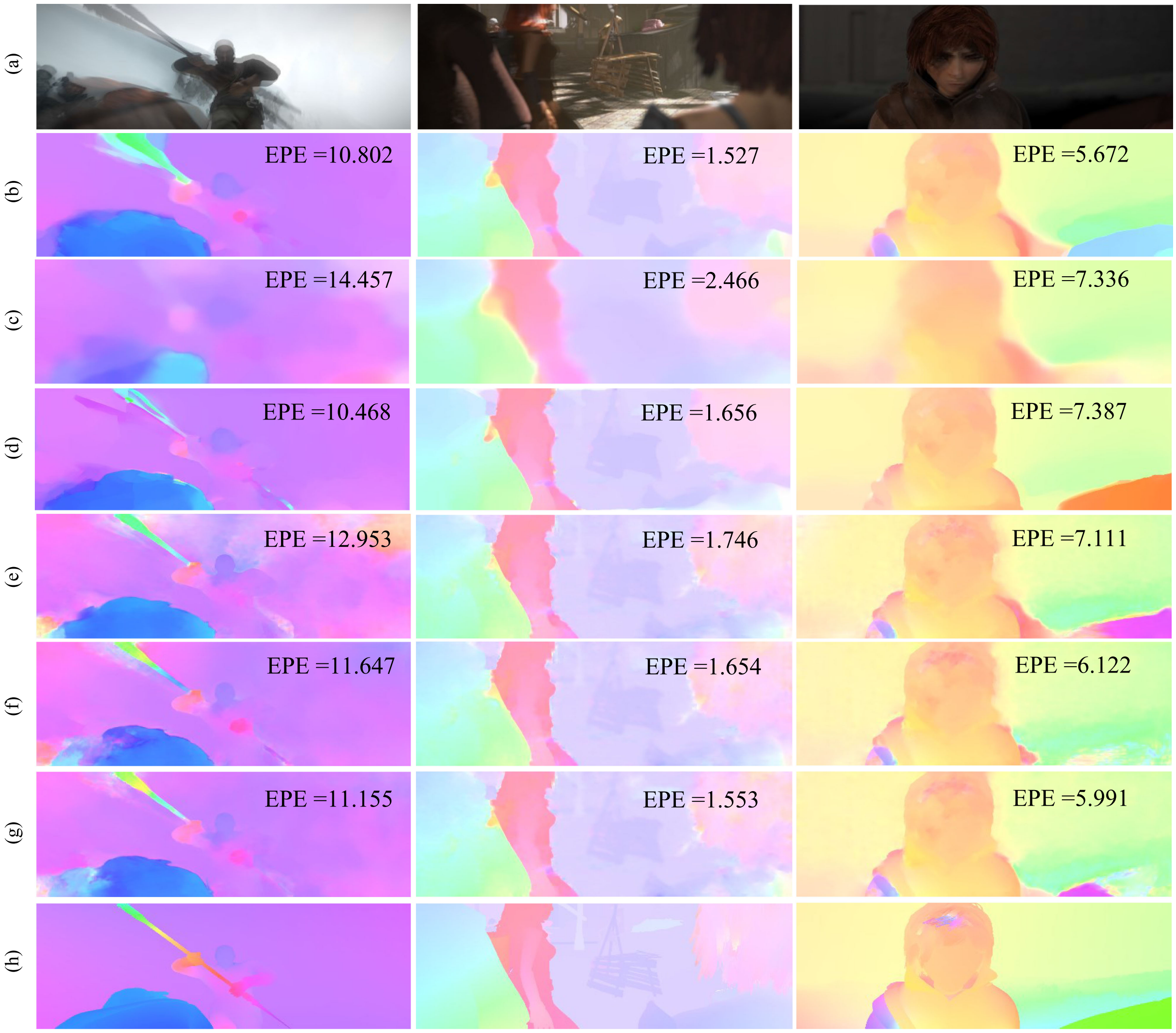


Fig. 5. Vsiual comparison of optical flow estimations.

trained iteratively from small to large resolution scale. We first train the CNN at a coarser scale, and then utilize the trained model as initialization for training at a finer scale. During training at a finer scale, we fix the CNN for estimation at coarser scales and only optimize it for the current scale. In experiments, we find that finetuning at the scale of full resolution blemishes the accuracy, and finetuning at the scale of 1/4 full resolution performs best. This is because that the same CNN works at different scales, and training too much on the fine scale will affect its ability to deal with coarse ones. In our method there is an energy-based refinement step that is applied alternatively with the refinement CNN. During training, we have also tried with applying the energy-based refinement to the coarse optical flow, yet don't find improvement of performance but the training time is greatly increased. Therefore, we don't apply the energy-based refinement during training. To train the refinement CNN, at each scale we apply Adam optimization with a batchsize of 16, and a learning rate that is 1×10^{-4} for the first 20 epochs, 1×10^{-5} for the rest. To increase the variety of flow fields, we also apply data augmentation like translation, cropping, rotation, noise addition and color jittering as in [24], [26]. The system is implemented using Torch framework, and experiments are performed on a Nvidia Geforce Titan X GPU card.

B. Benchmarks and Evaluation Metric

In our experiments, we adopt two metrics, which are average endpoint error (EPE) and percentage of optical flow outlier (Fl), for evaluation and analysis. The EPE measures the absolute error between the estimated optical flow (u_e, v_e) and groundtruth (u_g, v_g) by $EPE = \sqrt{(u_e - u_g)^2 + (v_e - v_g)^2}$. The Fl is defined as percentage of pixels with estimation error $>$ 3pixels and $>5\%$ of the true disparity. We perform experiments on three benchmarks: Flying Chair Dataset [26], KITTI2012 [57], and KITTI2015 Dataset [58], and compared with deep learning based methods FlowNetS [26], FlowNetC [26], SPynet [24] (all are trained using the same dataset as us) , spatial pyramid based methods DeepFlow [55], DIS-Fast [23], approximate nearest neighbor fields(ANN) based method FlowFields [46], and forward-backward consistency refinement based method EpicFlow [30]. Some visual results are shown in Fig. 5. The proposed network is denoted as $RecSPy$, and the system with the proposed refinement is denoted as $RecSPy^+$.

TABLE III
AVERAGE ENDPOINT ERROR(EPE) AND TESTING TIME ON THE TEST SPLIT OF THE THE FLYINGCHAIR DATASET

	EpicFlow [30]	FlowFields [46]	DeepFlow [55]	DIS-Fast [22]	FlowNetS [26]	FlowNetC [26]	SPyNet [24]	RecSPy	RecSPy$^+$
EPE	2.94	2.45	3.53	5.03	2.71	2.19	2.63	2.50	2.48
Time(s)	16	12	17	0.06	0.08	0.15	0.07	0.07	0.16

TABLE IV
RESULTS ON THE KITTI DATASETS

		EpicFlow	FlowFields	DeepFlow	DIS-Fast	FlowNetS+ft	FlowNetC+ft	SPyNet+ft	RecSPy	RecSPy+ft	RecSPy+ft$^+$
kitti12	$train$	3.09	3.33	4.48	11.01	7.52	8.79	4.13	10.2	3.12	2.76
(EPE)	$test$	3.8	3.5	5.8	14.4	9.1	–	4.7	13.7	4.5	3.6
kitti15	$train$	27.18%	24.43%	26.52%	53.73%	–	–	–	41.43%	25.12%	23.02%
(Fl-all)	$test$	27.10%	–	29.18%	–	–	–	–	40.90%	26.93%	25.32%

On the KITTI2012 average endpoint error(EPE) is evaluated. On the KITTI2015 ratio of pixel with optical flow estimation error $> 3\ \mathrm{pixels}$ and $> 5\%$ (Fl-all)

TABLE V
AVERAGE ENDPOINT ERROR(EPE) ON THE SINTEL DATASET

	EpicFlow	FlowFields	DeepFlow	DIS-Fast	FlowNetS+ft	FlowNetC+ft	SPyNet+ft	RecSPy	RecSPy+ft	RecSPy+ft$^+$
$train$	3.56	3.06	3.57	6.31	4.44	5.28	4.32	6.79	4.27	4.07
$test$	6.29	5.81	7.21	10.13	7.76	8.51	8.36	9.38	8.36	8.03

The Flying Chair Dataset [26] is a large synthetic dataset created by applying affine transformations to color images and a set of rendered 3D chair models. This dataset is composed of 22,232 training and 640 testing image pairs of resolution 512 × 384 as well as corresponding groundtruth. We pretrain our model on the training split and the results are shown in Table III. Our model achieves a better or comparable performance with the pure learning based methods FlownetS [26] and SPyNet [24], and the traditional methods FlowFields [46], EpicFlow [30], DeepFlow [55], and DIS-Fast [23] with a fast speed. This shows that our method is effective in learning the correspondence of pixels from data. The FlowNetC [26] outperforms our method with a large margin; this is because FlowNetC performs the feature correlation which compares two feature maps pixel by pixel. This operation works well on FlyingChair dataset due to the small resolution and simple content of testing images. We can see that on other datasets with complex and large images, FlowNetC may fail.

The KITTI2012 and KITTI2015 Dataset [57], [58]. These two datasets consist of realistic image pairs collected by a wide-view camera fixed on a driving car. The KITTI2012 dataset contains 194 training and 195 testing image pairs. The KITTI2015 comprises 200 training and 200 testing images. The sparse groundtruth flow of the training image pairs in these two datasets are also provided. Since the type of objects and motion are very different from those in the Flying Chair dataset, fine-tuning is performed on the combination of training sets from these two datasets. As shown in Table IV, our final result achieves a comparable performance to FlowFields and outperforms other methods on both datasets. While it should be noted that the ANN-based FlowFields is post refined via EpicFlow [46] and runs near 75 times slower than our methods. On the KITTI2012, our method and other learning based methods, which are FlowNetS, FlowNetC, and SPyNet, all have a high endpoint error on the training set after finetuning. This shows the difficulty of learning optical flow with deep networks. Especially for FlowNetC, the high estimation error on training set shows the correlation operation performs badly. Among these methods, our method achieves both low training error and testing error, which shows the effectiveness of our model to learn optical flow. There are also several other methods which are worth mentioning, like PatchBatch [45] that adopts CNNs for feature encoding and PH-Flow [18] that designs an energy function for piecewise homography to refine optical flow. Although they achieve better performance on KITTI2012 (PatchBatch EPE:3.3 and PH-Flow EPE:2.9) than ours (EPE: 3.6), their running speed is much slower. To estimate optic flow for one image pair PatchBatch takes about 50 seconds and PH-Flow takes more than 265 seconds, yet our method only takes about 0.35 seconds.

The MPI-Sintel Dataset [59] is built from computer-animated films. The image sequences in this dataset contain large/rapid motion, which makes the dataset challenging. The images in the 'final' subset of this dataset are generated with rendering effects like defocus blur, motion blur, and atmospheric effect. We finetune our model with the 'final' version training samples and evaluate on this subset. The results are shown in Table V. In this dataset, traditional methods DeepFlow achieve a better performance than deep learning based methods. This could be due to the fact that the outputs of deep networks are

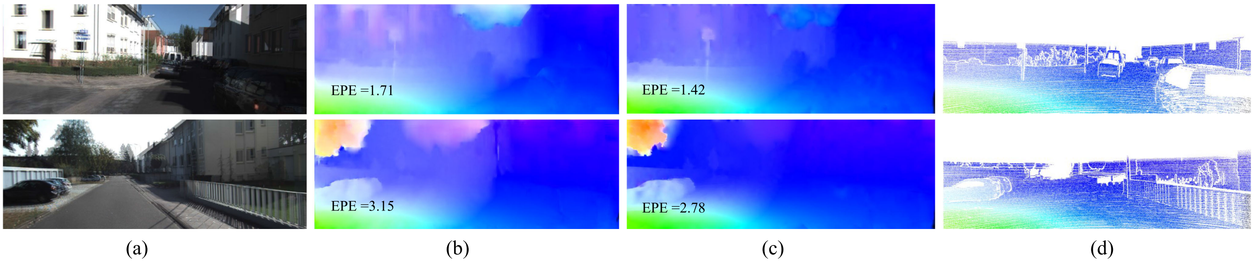


Fig. 6. Results of the proposed energy based refinement. (a) Image overlay. (b) RecSPy_ft. (c) RecSPy_ft $^{+}$. (d) GroundTruth.

always noisy and non-smooth. As shown in Fig. 5(b) and (f), although our network can estimate optical with more small motion details, the endpoint error is still higher.

C. Method Analysis

1) Running Time: Efficiency is very important for optical flow estimation. Since the running time is related to the resolution of input images, we perform the comparison of running time on the FlyingChair dataset and present the results in Table III. In the FlyingChair dataset, the input images are of resolution 512 × 384. Our method takes on average about 0.07 seconds and 0.16 seconds to process a pair of input images with and without the refinement step respectively. As shown in the table, comparing to traditional coarse-to-fine or ANN based methods, our approach achieves a comparable performance at a much faster speed, while comparing to other CNN based approaches like FlowNetS, FlowNetC and SPyNet our method can learn from data more effectively.

2) Model Size: The proposed system is composed of a Siamese network and a Refinement CNN. In the Siamese network, the two-branch CNN share the same parameters to extract features from the image. The extracted features are then converted to be an initial flow by an estimation CNN. Using the two-tower structure Siamese network enables the system to estimate the initial optical flow effectively with a small number of parameters. In the second part, formulating the spatial pyramid refinement as a recurrent CNN helps to further reduce the number of parameters. As compared in Table VII, our model further decreases the number of parameters but achieves better performance compared with SPyNet [24]. If converting the parameters in the model into half-precision floating point number, the model can be faster and more economical in terms of memory.

3) Effectiveness of Recurrent Spatial Pyramid CNN: As shown in Tables III–V, compared with FlowNet, our RecSPy without refinement achieves a better performance on the realistic dataset KITTI and a comparable performance on the animated datasets FlyingChair and Sintel with a much smaller model size and a faster speed. Furthermore, our RecSPy outperforms SPyNet on all three datasets with a smaller model size. The performance shows our RecSPy is more effective to learn optical flow from training data. We begin with analyzing the effectiveness of the Siamese network. We first estimate the initial flow using the Siamese network, and then train several refinement CNNs for different spatial scales separately as in SPyNet [24]. On the testing set of the FlyingChair dataset, this network with multiple refinement CNNs achieves an EPE of 2.42 and significantly outperforms SPyNet(EPE: 2.63), which shows that a good initial estimation is important and the Siamese network is effective considering its small model size. To evaluate the effectiveness of the refinement CNN, we present the performance for outputs of different scales in our RecSPy in Table VI. As we can see, the single refinement CNN is able to refine the coarse optical flow at different scales. It should be noted that comparing to our network with multiple refinement CNNs (with an $EPE = 2.42$), our RecSPy with a single refinement CNN halves the model size with only a slight decrease in EPE of 0.08.

TABLE VI
PERFORMANCE OF DIFFERENT LEVELS IN THE PYRAMID

Level	Scale	EPE	Time(s)
Initial	1/8	0.44	0.03
Level 1	1/4	0.75	0.03
Level 2	1/2	1.37	0.04
Level 3	1/1	2.50	0.07

TABLE VII
SIZES OF DIFFERENT MODELS

Method	FlowNetS	FlowNetC	SPyNet	RecSPy(ours)
#Param	38M	38M	1.2M	0.42M

4) Energy Based Refinement: The proposed energy function that encodes structure constraint and constancy constraint is applied as an optional refinement component in the final system. With this component in the proposed coarse-to-fine method, we can easily combine traditional methods to further improve the results produced by the CNN. The energy function can be optimized efficiently with existing methods as described in Section III-B. To keep the system efficient, we perform the refinement only once at each scale. As shown in Table III, this refinement only increases about 0.1 seconds when dealing with an input with a resolution of 512 × 384. To evaluate the the proposed refinement, we compare the performance last two columns in Tables III–V. On all the benchmarks except for the Flying Chair testing set, the refinement produces a significant improvement in accuracy for the estimated flow field. Visual results in Figs. 6 and 5 also show that the proposed refinement can improve the optical flow with more motion details. We notice that our energy based refinement only has a minor improvement

TABLE VIII
ANALYSIS OF ENERGY-BASED REFINEMENT USING ENDPOINT ERROR(EPE)

	KITTI12	KITTI15	SINTEL
w/o Refinement	3.12	6.02	4.27
Constancy	2.85	5.93	4.15
Structure	2.91	5.97	4.21
Constancy+Structure	2.76	5.89	4.07

on the Flying Chair dataset. This could be caused by the unnatural objects and motion fields in the dataset which are not suitable for the variational methods. By comparing the performance of FlowNetS with FlowNetS+v on this dataset in Table III, we can find that the conventional variational refinements even lead to worse performance. We also analyze the contribution of different terms in the proposed energy function for refinement. As shown in Table VIII, both the constancy constraint and structure constraint help to improve the estimated flow field. Applying these two constraints together can further improve the results.

V. Conclusion

In this paper, we have proposed a Recurrent Spatial Pyramid CNN for effective and efficient optical flow estimation. The system is composed of a Siamese network and a Recurrent CNN. The Siamese network extracts features from images in the input pair separately with the same branch CNN and then converts the features to be initial optical flow at a small spatial scale. The initial flow is then refined by a CNN recurrently form a small scale to the full resolution on the spatial pyramid. We have also proposed an energy function that imposes structure constraints and constancy constraints to optical flow to help refinement at each scale of the pyramid. The combination of the Recurrent Spatial Pyramid CNN and the energy based refinement endow our optical flow estimation system with effectiveness, efficiency, and a very small model size.

References

[1] Y. Shi, Y. Tian, Y. Wang, and T. Huang, "Sequential deep trajectory descriptor for action recognition with three-stream CNN," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1510–1520, Jul. 2017.

[2] W. Xu, Z. Miao, X.-P. Zhang, and Y. Tian, "A hierarchical spatio-temporal model for human activity recognition," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1494–1509, Jul. 2017.

[3] J. Liu, A. Shahroudy, D. Xu, A. K. Chichung, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2017.2771306.

[4] G. Wu and W. Kang, "Robust fingertip detection in a complex environment," *IEEE Trans. Multimedia*, vol. 18, no. 6, pp. 978–987, Jun. 2016.

[5] L. Chen, J. Shen, W. Wang, and B. Ni, "Video object segmentation via dense trajectories," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2225–2234, Dec. 2015.

[6] Y. Wan, Z. Miao, X.-P. Zhang, Z. Tang, and Z. Wang, "Illumination robust video foreground prediction based on color recovering," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 637–652, Apr. 2014.

[7] J. Xiong, H. Li, Q. Wu, and F. Meng, "A fast HEVC inter CU selection method based on pyramid motion divergence," *IEEE Trans. Multimedia*, vol. 16, no. 2, pp. 559–564, Feb. 2014.

[8] C.-W. Su *et al.*, "Motion flow-based video retrieval," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1193–1201, Oct. 2007.

[9] V. E. Liong, J. Lu, Y.-P. Tan, and J. Zhou, "Deep video hashing," *IEEE Trans. Multimedia*, vol. 19, no. 6, pp. 1209–1219, Jun. 2017.

[10] R. Phan and D. Androutsos, "Robust semi-automatic depth map generation in unconstrained images and video sequences for 2D to stereoscopic 3D conversion," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 122–136, Jan. 2014.

[11] S.-P. Lu, B. Ceulemans, A. Munteanu, and P. Schelkens, "Spatio-temporally consistent color and structure optimization for multiview video color correction," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 577–590, May 2015.

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

[13] M. J. Black and P. Anandan, "Robust dynamic motion estimation over time," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 1991, pp. 296–302.

[14] E. Memin and P. Perez, "Dense estimation and object-based segmentation of the optical flow with robust techniques," *IEEE Trans. Image Process.*, vol. 7, no. 5, pp. 703–719, May 1998.

[15] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2004, pp. 25–36.

[16] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *Int. J. Comput. Vis.*, vol. 67, no. 2, pp. 141–158, 2006.

[17] C. Liu, J. Yuen, and A. Torralba, "SIFT flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.

[18] J. Yang and H. Li, "Dense, accurate optical flow estimation with piecewise parametric model," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2015, pp. 1019–1027.

[19] J. Yang, H. Li, Y. Dai, and R. T. Tan, "Robust optical flow estimation of double-layer images under transparency or reflection," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2016, pp. 1410–1419.

[20] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, Mar. 2011.

[21] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Deepmatching: Hierarchical deformable dense matching," *Int. J. Comput. Vis.*, vol. 120, no. 3, pp. 300–323, 2016.

[22] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patch match for large displacement optical flow," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2016, pp. 5704–5712.

[23] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool, "Fast optical flow using dense inverse search," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 471–488.

[24] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proc. IEEE Comput. Vis. Pattern Recogniti.*, 2017, pp. 4161–4170.

[25] G.-J. Qi, H. Larochelle, B. Huet, J. Luo, and K. Yu, "Guest editorial: Deep learning for multimedia computing," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1873–1874, Nov. 2015.

[26] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2758–2766.

[27] E. Ilg *et al.*, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2017, pp. 2462–2470.

[28] Y. Zhu, Z. Lan, S. Newsam, and A. G. Hauptmann, "Guided optical flow learning," arXiv:1702.02295, 2017.

[29] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010, pp. 177–186.

[30] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2015, pp. 1164–1172.

[31] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1998, pp. 839–846.

[32] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 1–14.

[33] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, 1981, pp. 674–679.

[34] Y. Mileva, A. Bruhn, and J. Weickert, "Illumination-robust variational optical flow with photometric invariants," in *Proc. Joint Pattern Recognit. Symp.*, 2007, pp. 152–162.

[35] L. Xu, J. Jia, and Y. Matsushita, "Motion detail preserving optical flow estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1744–1757, Sep. 2012.

[36] A. Wedel, D. Cremers, T. Pock, and H. Bischof, "Structure-and motion-adaptive regularization for high accuracy optic flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 1663–1668.

[37] Q. Chen and V. Koltun, "Full flow: Optical flow estimation by global optimization over regular grids," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2016, pp. 4706–4714.

[38] R. Szeliski *et al.*, "A comparative study of energy minimization methods for Markov random fields with smoothness-based priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1068–1080, Jun. 2008.

[39] B. Glocker, N. Paragios, N. Komodakis, G. Tziritas, and N. Navab, "Optical flow estimation with uncertainties through dynamic MRFs," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[40] D. Sun, S. Roth, J. Lewis, and M. J. Black, "Learning optical flow," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 83–97.

[41] L. Bao, Q. Yang, and H. Jin, "Fast edge-preserving patchmatch for large displacement optical flow," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2014, pp. 3534–3541.

[42] K. He and J. Sun, "Computing nearest-neighbor fields via propagation-assisted KD-trees," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2012, pp. 111–118.

[43] S. Korman and S. Avidan, "Coherency sensitive hashing," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1607–1614.

[44] Z. Chen, H. Jin, Z. Lin, S. Cohen, and Y. Wu, "Large displacement optical flow from nearest neighbor fields," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2013, pp. 2443–2450.

[45] D. Gadot and L. Wolf, "PatchBatch: A batch augmented loss for optical flow," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2016, pp. 4236–4245.

[46] C. Bailer, B. Taetz, and D. Stricker, "Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4015–4023.

[47] M. Bai, W. Luo, K. Kundu, and R. Urtasun, "Exploiting semantic information and deep matching for optical flow," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 154–170.

[48] J. Y. Jason, A. W. Harley, and K. G. Derpanis, "Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 3–10.

[49] M. Jaderberg *et al.*, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[50] C. Bailer, K. Varanasi, and D. Stricker, "CNN-based patch matching for optical flow with thresholded hinge embedding loss," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2017, pp. 2710–2719.

[51] T. Schuster, L. Wolf, and D. Gadot, "Optical flow requires multiple strategies (but only one network)," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2017, pp. 6921–6930.

[52] S. Zweig and L. Wolf, "InterpoNet, a brain inspired neural network for optical flow dense interpolation," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2017, pp. 6363–6372.

[53] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "ENet: A deep neural network architecture for real-time semantic segmentation," arXiv:1606.02147, 2016.

[54] D. Young, *Iterative Solution of Large Linear Systems*. New York, NY, USA: Elsevier, 1971.

[55] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "DeepFlow: Large displacement optical flow with deep matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1385–1392.

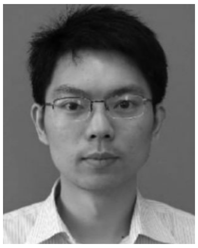
[56] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980, 2014.

[57] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[58] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2015, pp. 3061–3070.

[59] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 611–625.

Ping Hu received the B.Eng. degree from Sichuan University, Chengdu, China, in 2013, and the M.Eng. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2016. He is currently studying at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include multimedia analysis, computer vision, and machine learning.



Gang Wang received the B.Eng. degree in electrical engineering from the Harbin Institute of Technology, Harbin, China, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA. He was an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. From 2010 to 2014, he had a joint appointment with the Advanced Digital Science Center, Singapore, as a Research Scientist. He is currently a Researcher with Alibaba and a Chief Scientist with the Alibaba AI Labs.

Prof. Wang was a recipient of MIT Technology Review Innovator under 35 Awards (Asia). He is currently an Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and an Area Chair of International Conference on Computer Vision 2017 and Conference on Computer Vision and Pattern Recognition 2018.

Yap-Peng Tan received the B.S. degree from the National Taiwan University, Taipei, Taiwan, in 1993, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, USA, in 1995 and 1997, respectively, all in electrical engineering. From 1997 to 1999, he was with the Intel Corporation, Chandler, AZ, USA, and the Sharp Laboratories of America, Camas, WA, USA. In November 1999, he joined Nanyang Technological University, Singapore, where he is currently a Professor and an Associate Chair (Academic) with the School of Electrical and Electronic Engineering. His research interests include image and video processing, content-based multimedia analysis, computer vision, pattern recognition, machine learning, and data analytics. He was the Chair of the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society from 2012 to 2014, Chair of the Membership and Election Subcommittee of the Multimedia Systems and Applications Technical Committee of the IEEE Circuits and Systems Society from 2013 to 2017, Chair of the Nominations and Elections Subcommittee of the Multimedia Signal Processing Technical Committee of the IEEE Signal Processing Society from 2012 to 2013, a voting member of the IEEE International Conference on Multimedia and Expo (ICME) Steering Committee from 2011 to 2012, and the Chairman of the IEEE Signal Processing Singapore Chapter from 2009 to 2010. He was also an Associate Editor for the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE SIGNAL PROCESSING LETTERS, and the IEEE ACCESS, an Editorial Board Member for the *EURASIP Journal on Advances in Signal Processing* and *EURASIP Journal on Image and Video Processing*, Guest Editor for special issues of several journals including the IEEE TRANSACTIONS ON MULTIMEDIA, a member of the Multimedia Systems and Applications Technical Committee and Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society, and a member of the Image, Video and Multidimensional Signal Processing Technical Committee of the IEEE Signal Processing Society. He is the Technical Program Co-Chair of IEEE International Conference on Multimedia and Expo (ICME 2018) and the 2019 IEEE International Conference on Image Processing (ICIP 2019), the Chair of the ICME Steering Committee from 2018 to 2019, and was the Finance Chair of ICIP 2004, General Co-Chair of ICME 2010, Technical Program Co-Chair of ICME 2015, and General Co-Chair of the 2015 IEEE International Conference on Visual Communications and Image Processing 2015.