Homography-guided stereo matching for wide-baseline image interpolation

Yuan Chang^{1,2}, Congyi Zhang^{1,2,3}, Yisong Chen^{1,2}, and Guoping Wang^{1,2} (✉)

© The Author(s) 2021.

Abstract Image interpolation has a wide range of applications such as frame rate-up conversion and free viewpoint TV. Despite significant progresses, it remains an open challenge especially for image pairs with large displacements. In this paper, we first propose a novel optimization algorithm for motion estimation, which combines the advantages of both global optimization and a local parametric transformation model. We perform optimization over dynamic label sets, which are modified after each iteration using the prior of piecewise consistency to avoid local minima. Then we apply it to an image interpolation framework including occlusion handling and intermediate image interpolation. We validate the performance of our algorithm experimentally, and show that our approach achieves state-of-the-art performance.

Keywords image interpolation; view synthesis; homography propagation; belief propagation

1 Introduction

Image interpolation is a process that generates a new image using available images, which is useful for frame rate-up conversion [1], view synthesis [2], etc. In some applications, the available images have a wide baseline. Here, baseline means the translation and rotation that a camera undergoes to capture image pairs. For example, in virtual street roaming applications, users can teleport themselves from one

street spot to another by clicking the directional arrow. In order to make the transition between discrete views smooth, it is important to realistically interpolate intermediate views between wide-baseline image pairs since the sampled street views are usually far from each other.

Nie et al. [2] discussed the definition of various kinds of baselines, and divided them into three categories based on the median distance between the KITTI images [3]: small-baseline, medium-baseline, and wide-baseline. The basic idea of most image interpolation algorithms is to estimate the motion field of the input views and map them to the desired position. Traditional interpolation methods were usually designed for small baseline images [4], and recent large displacement optical flow methods [5] can be regarded as medium-baseline algorithms. Due to the large translations and rotations involved, it is still a challenging problem to estimate the motion field for wide-baseline image pairs.

One classical approach to motion estimation is to consider it as a labeling problem, which can be formulated to a global optimization problem in a Markov random field. In other words, we need to select the best motion vector from the set of potential motion vectors for each pixel in the source image, to minimize the energy defined using some prior assumptions such as brightness constancy and spatial smoothness. However, since the space of all possible motion vectors is usually too large, employing global optimization over the full image grid in this space has excessive computational requirements. To reduce the amount of computation, some approaches use a search window as the candidate label set [6]. However, for wide-baseline image pairs, the window size should be very large to avoid falling into local minima, which makes the optimization prohibitively slow. Other

1 Peking University, Beijing 100871, China. E-mail: Y. Chang, changyuan@pku.edu.cn; Y. Chen, chenysisong@pku.edu.cn; G. Wang, wgp@pku.edu.cn (✉).

2 Beijing Engineering Technology Research Center of Virtual Simulation and Visualization, Peking University, Beijing 100871, China.

3 The University of Hong Kong, Hong Kong, China. E-mail: cyzh@hku.hk.

Manuscript received: 2021-01-20; accepted: 2021-03-14

approaches use approximate nearest neighbors in feature space to prune the set of potential motions [5]. But the proposed set is still excessive, because it needs to maintain a high recall of the target motions. So they have to perform optimization on a sampled image grid, and use interpolation [7] to get the motion field of the full image grid.

An alternative strategy to estimate the motion is to compute parametric transformation models locally, which can transform each pixel to its target position in the target image [2]. It is an efficient strategy to deal with wide-baseline image pairs. However, this strategy cannot guarantee the estimated motion field to be piecewise smooth, which may lead to some artifacts of stretching, overlapping, holes, etc. Therefore, methods using this strategy usually need an extra global optimization stage to further eliminate artifacts.

In this paper, we propose a novel method of motion estimation, which combines the advantages of both global optimization and local parametric transformation model based algorithms. We formulate the problem in terms of global optimization in a Markov random field. Rather than using a constant set of candidate motions like previous methods [5, 6], we adjust the candidate set iteratively guided by homography fitting and propagation. More specifically, we first initialize the set of candidate motions for each pixel by approximate nearest neighbor search in feature space. Unlike DiscreteFlow [5], where the candidate set is excessive, the size of our candidate set can be very small. Then, we perform global optimization over the full image grid with the proposed candidate set. As the small candidate set may not include the target motion, we propose a novel strategy to update the candidate set iteratively through local refinement under a piecewise parametric model. Our approach requires neither a large candidate set to guarantee that the target motion is included, nor a coarse-to-fine scheme to gradually refine the estimated motions.

In summary, the main contributions of this paper are as follows:

- a novel optimization framework for motion estimation based on homography guided belief propagation,
- application of the proposed motion estimation method to an image interpolation framework,
- experiments to show that our approach is able to deal well with the wide-baseline image interpolation problem, and
- a demonstration that our approach also performs well for traditional small-baseline image pairs too, through experiments on a typical optical flow dataset.

The rest of this paper is organized as follows: we first review the related work in Section 2. Then in Section 3 we introduce our approach including candidate set initialization, the inference algorithm, and the modification strategy for the candidate set. In Section 4, our algorithm is validated and compared to other approaches experimentally. Finally, we conclude and discuss the limitations of this paper in Section 5.

2 Related work

As mentioned above, the basic idea of image interpolation algorithms is motion estimation. In other words, image interpolation is a high-level application of motion estimation techniques. So we first review relevant low-level motion estimation algorithms, and then we mainly review related work on image interpolation including frame rate-up conversion and view synthesis.

2.1 Motion estimation

Optical flow methods are typical motion estimation algorithms, most of which are designed for small-baseline image pairs. Since the original work of Horn and Schunck [8], there has been a huge body of literature on optical flow [9–12]. One typical approach is to consider it as a labeling problem as mentioned in Section 1. The motion field can be estimated by solving an energy minimization problem based on brightness constancy and spatial smoothness [13–15]. Since the space of all possible labels is usually too large or even infinite [16, 17], some strategies have been proposed to reduce the label set. The simplest way is to use a search window centered at the initial label [6], but it is prone to converging to local minima, especially when there are large displacements between image pairs. Discrete Flow [5] pruned the label set by proposing a diverse set of candidate labels using approximate K -nearest-neighbor search and random sampling around the reference pixel. Veksler [18] decreased the computational cost of the

graph cuts stereo correspondence technique efficiently using the results of a simple local stereo algorithm to limit the disparity search range. The particle belief propagation technique [19] applied Markov chain Monte Carlo sampling to the current belief estimation using a Gaussian proposal distribution. Besse et al. [20] defined a new family of algorithms, called PMBP, which combines the best features of both PatchMatch and particle belief propagation; they leveraged PatchMatch to produce particle proposals effectively. Other methods are based on PMBP [21, 22]. Li et al. [21] proposed a method called SPM-BP to tackle the computational bottleneck of PMBP. Hornáček et al. [22] showed that optimization over high-dimensional, continuous state space can be carried out using an adaptation of PMBP. We use belief propagation as the base algorithm to optimize the objective function too. But instead of using PatchMatch, we utilize homography estimation to propose new labels, which performs better than PMBP-based methods.

There are also many other types of optical flow estimation algorithms. For example, the recent advances in deep learning have significantly influenced the literature on optical flow estimation. However, it is beyond the scope of this paper to review the entire literature. For a more detailed survey of optical flow estimation, please refer to Refs. [23, 24].

2.2 Frame rate-up conversion

Frame rate-up conversion is a typical application of image interpolation, where one can interpolate intermediate frames between adjacent video frames to increase the frame rate of a video. In this situation, objects undergo very small displacements, since sequential video frames are very similar. Owing to their simplicity, block matching algorithms are commonly used in frame rate-up conversion [25]. These methods divide a frame into non-overlapping blocks and search for the most similar block in the following frame. At the pixel level, Mahajan et al. [26] moved the image gradients through a given time step and solve a Poisson equation to reconstruct the interpolated frame. Stich et al. [27] found edges and homogeneous regions in images for matching, yielding a dense motion field between images. Meyer et al. [28] proposed propagating phase information across oriented multi-scale pyramid levels for video interpolation. CNN-based methods

also show good performance for this application. Long et al. [29] trained a deep CNN to directly predict the interpolated frames, but the results are usually blurred. Some methods take advantage of accurately estimated pixel-wise optical flow to improve performance [1, 4]. Other methods formulate frame interpolation as convolution over local patches and estimate the convolution kernels for each output pixel [30, 31]. However, these methods are designed for small-baseline image pairs, and they are ineffective for wide-baseline image interpolation.

2.3 View synthesis

View synthesis is the process of generating a new view using existing views taken from multiple cameras. In this situation, there may be large displacement because of large translation or rotation of a camera. Recently, large-displacement optical flow methods have been proposed. Some methods initialize the variational model by sparse feature correspondences or an approximate nearest neighbor field [32], which helps to escape from the local minima. These methods are improved by proposing more sophisticated feature matching algorithms [7]. From a different angle, Bao et al. [33] obtained large displacement optical flow by increasing the smoothness of PatchMatch [34]. However, these methods do not perform very well for wide-baseline image interpolation. Image-based rendering techniques [35–38] have been proposed to get better results in wide-baseline view synthesis. Chaurasia et al. [37] reconstructed a 3D model for a scene, and compensated for reconstruction errors by depth synthesis. However, sometimes they may fail to reconstruct the 3D scene, e.g., if there are too few images. Some researchers have applied deep learning methods to view synthesis problem [39–42]. For example, Zhou et al. [39] trained a convolutional neural network to generate an appearance flow vector that specifies which pixels in the input image can be used to reconstruct the output. However, learning based methods require a large amount of training data and much training time. Nie et al. [2] proposed a method that only needs two images as input. They oversegment the source image into superpixels, and estimate for each superpixel a homography, which transforms each superpixel to the target position. However, without explicitly enforcing a spatial smoothness constraint, artifacts may occur because of the discontinuity between different superpixels.

Although there is a mesh warping framework to further eliminate artifacts, some artifacts still remain like stretching and holes. Our method is similar to Ref. [2], since we both use the assumption that each superpixel represents a small plane, and our method also includes homography fitting and propagation. But unlike them, we formulate the whole process of motion estimation as an energy minimization problem, which explicitly enforces spatial smoothness and achieves better performance than Ref. [2].

3 Approach

3.1 Background

Our aim is to generate intermediate images between two given images I_1 and I_2 . To that end, we compute a forward displacement vector from I_1 to I_2 for each pixel in I_1 and a backward displacement vector from I_2 to I_1 for each pixel in I_2 . Our approach considers this to be a labeling problem, where the label here is the displacement vector for each pixel. We solve this problem by minimizing an energy function in a Markov random field (MRF) over dynamic candidate label sets. Inspired by belief propagation (BP) [43], we propose a novel optimization scheme guided by homography fitting and propagation to avoid local minima. The pipeline is shown in Fig. 1. First of all, for each pixel, we generate an initial candidate label set whose size is very small: see Section 3.3. Then, to tackle the problem of insufficient candidates caused by the limited size of the label set, we propose new labels using homography estimation, and modify the candidate label sets after each iteration of the optimization: see Section 3.4.

Before presenting the details of the algorithm, we first introduce the formulation of our motion

estimation approach and some essential concepts of BP in Section 3.2.

3.2 Formulation of motion estimation

Without loss of generality, we only consider estimation of forward displacement vectors from I_1 to I_2 , since the backward displacement from I_2 to I_1 can be obtained in exactly the same way. Our goal is to estimate the motion field w for I_1 , where $w(p) = (u(p), v(p))$ is the displacement vector at pixel p and $p = (x, y)$ represents pixel coordinates in image I_1 . Since we formulate this problem as global optimization in an MRF, we can also consider $w(p)$ to be a label for pixel p . The energy function to be minimized is formulated as Eq. (1); it includes a data term E_d and a smoothness term E_s . The data term represents the similarity between the matched pixels corresponding to the motion field, and the smoothness term constrains the labels of adjacent pixels to be similar. Here, ϵ is a set containing all neighborhoods on a four-connected image grid, and λ weights the smoothness term.

$$E(w) = \sum_p E_d(w(p)) + \lambda \sum_{(p,q) \in \epsilon} E_s(w(p), w(q)) \quad (1)$$

Let $C(p) = \{w_1^p, \dots, w_L^p\}$ be the candidate label set of each pixel p in image I_1 , which contains L candidate labels. For simplicity, here we set the size of every pixel's label set to be the same, L , although they can differ in our algorithm.

Belief propagation is an inferencing algorithm which works by passing messages around the 4-connected image grid iteratively [43]. It updates an L -dimensional message $m_{p \rightarrow q}^t(w_i^q)$, $1 \leq i \leq L$, sent from each pixel p to each neighbor q at each iteration t from $[0, T]$. The messages are computed in the following way, where $\mathcal{N}(p) \setminus q$ denotes the neighbors

Fig. 1 Pipeline of our approach. Label set initialization composes the label sets using N nearest neighbors in feature space. In addition to iterative optimizing the objective function, the optimization phase marks the worst candidate in each label set by cost and replaces it by a new label proposal in each iteration.

of p other than q .

$$\begin{aligned} m_{p \rightarrow q}^t(w_i^q) = & \min_{1 \leq j \leq L} (E_s(w_j^p, w_i^q) + E_d(w_j^p)) \\ & + \sum_{s \in \mathcal{N}(p) \setminus q} m_{s \rightarrow p}^{t-1}(w_j^p) \end{aligned} \quad (2)$$

Then, with the obtained $m_{p \rightarrow q}^t$, we can compute a belief vector $b_p^t(w_i^p)$ for each pixel p at each iteration t using

$$b_p^t(w_i^p) = E_d(w_i^p) + \sum_{(p,q) \in \epsilon} m_{q \rightarrow p}^t(w_i^p) \quad (3)$$

The value of $b_p^t(w_i^p)$ represents an approximation to the probability that the correct label for p is w_i^p . After T iterations, the final belief vector $b_p^T(w_i^p)$ can be calculated for each pixel, and we can select the best label $w^*(p)$ for every pixel p from its label set $C(p)$ by minimizing $b_p^T(w_i^p)$ pixelwise.

How we choose the label set $C(p)$ is very important. The set cannot be too large because optimization will be prohibitively slow. But a fixed small candidate label set may easily cause convergence to local minima. Therefore, our approach uses a compact dynamic candidate label set. We initialize a very small label set for each pixel, and modify the label sets iteratively during BP to avoid local minima.

3.3 Initialization

We use a multi-scale K -nearest-neighbor search strategy to initialize the candidate label sets, as shown in Fig. 2. First, we construct image pyramids with N_L levels, where $N_L = 4$ in our experiments, for both I_1 and I_2 , by downsampling the original images using bilinear interpolation. Let I_i^ℓ ($i = 1, 2$) be the downsampled image of I_i at each pyramid level ℓ . We compute a feature descriptor for each pixel in I_1^ℓ and I_2^ℓ to help finding correspondences: for a wide-baseline image pair, the brightness of an object may change during the transition between

views, so a feature descriptor is more robust when finding nearest matches. To overcome local scale and rotation changes in the wide baseline scenario, we use per-pixel scale-invariant feature transform (SIFT) descriptors [6] as the dense feature descriptor. After we get the feature maps D_1^ℓ for I_1^ℓ and D_2^ℓ for I_2^ℓ , we search K_ℓ nearest neighbors in D_2^ℓ for every descriptor in D_1^ℓ under L_1 distance. Then we get K_ℓ labels corresponding to the K_ℓ nearest neighbors for each pixel in I_1^ℓ at level ℓ , and we upsample it to the original scale of image I_1 to propose K_ℓ initial labels for each pixel in I_1 . We collect the initial labels proposed from each level ℓ to get the initial candidate label set of each pixel in I_1 with size $N = \sum_\ell K_\ell$. In our experiments, we consider $K_\ell = 2$ labels for each level ℓ to get 8 candidates for each pixel. Note that the multi-scale scheme is only used during initialization. The optimization stage does not require a coarse-to-fine scheme to prevent local minima, since we use the homography guided modification strategy, as introduced in the next section.

3.4 Optimization

3.4.1 objective function

We first introduce the specific data term and smoothness term used. We use the truncated L_1 distance between the matched SIFT descriptors (computed in the initialization phase) along with the displacement as the data term to account for matching outliers, and we use the truncated L_1 distance between labels of neighboring pixels as the smoothness term to account for motion discontinuities. These are shown in Eqs. (4) and (5), where D_1 and D_2 are the feature maps of the original input images I_1 and I_2 , and τ_d and τ_s are the truncation thresholds of the data term and the smoothness respectively.

$$E_d(w(p)) = \min(\|D_1(p) - D_2(p + w(p))\|_1, \tau_d) \quad (4)$$

$$E_s(w(p), w(q)) = \min(\|w(p) - w(q)\|_1, \tau_s) \quad (5)$$

Given this specific energy function, optimization can be performed. As mentioned in Section 3.2, a small candidate label set may lead to local minima easily, so we propose a novel optimization scheme to tackle the problem. Inspired by BP [43], we also solve the minimization problem by passing messages. But after message passing at each iteration, we perform a homography check and a label set modification

Fig. 2 Label set initialization.

to prevent local minima. In order to conduct the homography check and the label set modification, we first over-segment image I_1 into superpixels $S = \{S_1, \dots, S_K\}$ following Ref. [44], and we regard each superpixel as a small plane, which corresponds to somewhere in I_2 by a homography, as in Ref. [2], since it is small and it usually has homogeneous color.

3.4.2 Homography check

As introduced in Section 3.2, we compute a belief vector $\mathbf{b}_p^t(\mathbf{w})$ for every pixel $\mathbf{p}$ after each iteration t , and select the currently best label $\mathbf{w}_t^*(\mathbf{p})$ from $\mathbf{C}(\mathbf{p})$ for $\mathbf{p}$. With prior knowledge of the plane approximation for each superpixel, we can fit a homography H_i for each superpixel S_i from the best labels of all the pixels in S_i using RANSAC [45]. The homographies help to generate new labels while modifying the label sets, as explained later. To ensure the validity of labels suggested by homographies, we first need to identify whether a homography is reliable or not.

After H_i is obtained, we can project each pixel $\mathbf{p}$ in S_i to a new location $\mathbf{p}'$ in I_2 using H_i .

$$\mathbf{p}' = H_i \mathbf{p} \quad (6)$$

Let $\mathbf{q} = \mathbf{p} + \mathbf{w}_t^*(\mathbf{p})$ be the location corresponding to the current best label. Then we can define a delta function using the Euclidean distance $\text{Dis}(\mathbf{p}', \mathbf{q})$ between $\mathbf{p}'$ and $\mathbf{q}$:

$$\delta(\mathbf{p}) = \begin{cases} 1, & \text{if } \text{Dis}(\mathbf{p}', \mathbf{q}) < r \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

to determine whether a pixel is an inlier ($\delta(\mathbf{p}) = 1$) or an outlier ($\delta(\mathbf{p}) = 0$), where r is a threshold. We show the process in Fig. 3.

Then we can compute the reliability $\text{Re}(S_i)$ of the fitted homography H_i of a superpixel S_i , which calculates the proportion of inlier pixels in

Fig. 3 Inlier / outlier pixel discrimination. $\mathbf{p}_1$ represents an inlier while $\mathbf{p}_2$ represents an outlier.

a superpixel:

$$\text{Re}(S_i) = \frac{\sum_{\mathbf{p} \in S_i} \delta(\mathbf{p})}{|S_i|} \quad (8)$$

where $|S_i|$ is the number of pixels in S_i . Therefore, we can identify whether the fitted homography H_i of a superpixel S_i is reliable using a threshold ζ , and find the set $\mathcal{R}$ of superpixels whose fitted homographies are reliable.

$$\mathcal{R} = \{S_i | \text{Re}(S_i) > \zeta\} \quad (9)$$

The remaining superpixels constitute the set $\mathcal{U} = \mathcal{S} \setminus \mathcal{R}$ of superpixels whose fitted homographies are unreliable.

3.4.3 Label set modification

After determining all superpixels to be reliable $\mathcal{R}$ or unreliable $\mathcal{U}$, we modify the candidate label set by substituting new labels. Substituting a label $\mathbf{w}$ here means replacing the worst label in the current candidate label set for each pixel by the new proposed label $\mathbf{w}$. Here, similar to the definition of the currently best label, we select the current worst label by maximizing $\mathbf{b}_p^t(\mathbf{w}_i^p)$.

We use different ways to propose new labels for pixels for reliable and unreliable superpixels. In the first case, if a pixel $\mathbf{p}$ belongs to a reliable superpixel S_i , we directly use the homography H_i fitted in the homography check to generate a new label using Eq. (10) since we consider the reliable homography to be a good estimate of the transformation of $\mathbf{p}$ from I_1 to I_2 .

$$\mathbf{w}_{\text{new}} = H_i \mathbf{p} - \mathbf{p} \quad (10)$$

If $\mathbf{p}$ is a pixel of superpixel S_i whose fitted homography H_i is unreliable, we can not use H_i directly to generate a new label. Instead, we utilize other superpixels whose homographies are reliable to help generate new labels. To that end, we construct an undirected graph whose nodes are all superpixels and edges connecting the superpixels with shared boundaries, as shown in Fig. 4. The weight of each edge is defined as the color similarity between the connected superpixels. Following Ref. [2], we create a normalized color histogram for each superpixel, and compute the χ^2 distance between histograms of pairs of adjacent superpixels as the color similarity. Using the graph structure, we define the similarity between any two superpixels as the shortest path connecting them on the graph, which can be easily computed using Dijkstra's algorithm.

Fig. 4 The superpixel graph. Red lines: boundaries of superpixels. Yellow points (graph nodes) and black lines (graph edges) illustrate the graph structure.

Then we generate the M new labels based on the similarity between any two superpixels. We search for the M most similar superpixels from R for $S_i \in U$, and we project $\mathbf{p}$ using the M corresponding homographies H_i^j , $j = 1, \dots, M$, to generate the M new labels $\mathbf{w}_j$ as below:

$$\mathbf{w}_j = H_i^j \mathbf{p} - \mathbf{p} \quad (11)$$

We show this process in Fig. 5. For the unreliable superpixel S_i (yellow in Fig. 5(b)), we consider the M superpixels (blue) in R most similar to S_i . Note that we do not use the neighboring superpixels directly to propose new labels for S_i , as some neighboring superpixels may not belong to the same object as S_i when S_i is near the boundary of an object. Moreover, unlike reliable superpixels where we propose one

new candidate for each pixel, we propose M new candidates for each pixel in the case of unreliable superpixels, to improve the chance that we propagate the correct homography to the unreliable superpixel.

Since we use the same homography to generate new labels for pixels in the same superpixel in both two cases, these labels will share consistency between neighboring pixels so that the smoothness term may be reduced dramatically even if these labels are incorrect. Therefore, in practice, to avoid this issue, during each iteration, we uniformly sample 30% pixels from the outlier pixels of reliable superpixels and 30% pixels from unreliable superpixels for modification.

3.5 Occlusion handling

Since we do not consider occlusion explicitly, the computed displacement vectors of occluded pixels may be incorrect. Therefore, we remove outliers from our result using forward-backward consistency checking, i.e., we compute forward displacement vectors from I_1 to I_2 and backward vectors from I_2 to I_1 and discard inconsistent ones. Then we use a state-of-the-art interpolation scheme [46] to interpolate the discarded regions.

3.6 Interpolation

With the computed displacement vectors $\mathbf{w}_1$ for I_1 and $\mathbf{w}_2$ for I_2 , we can smoothly interpolate any intermediate image I_t at time $t \in (0, 1)$ between I_1 and I_2 using a patch-based reconstruction scheme [47]. For any pixel $\mathbf{p}$ in I_1 , its motion vector to I_t is $t \cdot \mathbf{w}_1(\mathbf{p})$. Thus, we can map each pixel $\mathbf{p}$ in I_1 to

Fig. 5 Generating new labels for unreliable superpixels. (a) Original input image I_1 . (b) Searching for similar superpixels in R . Dark regions: unreliable superpixels. Yellow: unreliable superpixel to be processed. Blue: most similar superpixels found in R to the yellow one.

its new location $\mathbf{p} + t \cdot \mathbf{w}_1(\mathbf{p})$ in $\mathbf{I}_t$ to render the intermediate image. Likewise, we can also render the intermediate image using $\mathbf{I}_2$.

After obtaining the intermediate image $\mathbf{I}_t^1$ warped from $\mathbf{I}_1$ and $\mathbf{I}_t^2$ from $\mathbf{I}_2$, we blend them using the multiband blending method [48] to get the final interpolation result $\mathbf{I}_t$.

4 Experiments

In this section, we first analyze the performance of our approach experimentally, and validate the claims made in Section 4.1. Then we evaluate our method by comparing it to prior work in Section 4.2.

4.1 Performance analysis

4.1.1 Validation for label set modification

Since we use a very small candidate label set for each pixel, the initial label set may not include the correct label at all. Therefore, if we perform optimization over constant label sets, it is easy to fall into local minima. However, our strategy of label set modification can help avoid local minima without enlarging the label sets. To validate this claim, we first perform experiments on image pairs with ground truth optical flow. We show two cases from the MPI Sintel dataset [49] with and without large displacements respectively.

To evaluate a pixel's candidate label set, we select the label nearest to the ground truth label from the label set. If the endpoint error (EPE) between the selected label and the ground truth label is less than γ pixels, where γ is an threshold, the pixel's candidate label set is considered to be a good label set. Pixels without good label sets tend to stick in local minima more often than pixels with good label sets. Therefore, we expect more pixels to have good label sets after label set modification. To show the quality of all pixels' label sets clearly, we draw a pixel in black if its label set is good, and otherwise white: see Fig. 6 and Fig. 7. We compare the ratio of pixels with good label sets before optimization to the ratio after 10 iterations of optimization, to evaluate the effectiveness of our label set modification strategy, using a threshold $\gamma = 5$. It is shown that our modification strategy effectively improves the ratio of pixels with good label sets. For an image pair with small displacement (see Fig. 6), 93.1% pixels' initial label sets are good, while 99.1% pixels' modified label

Fig. 6 Effects of our label set modification strategy for an image pair with small displacement. Pixels having a good label set are black, and others are white. (c) and (d) visualize label set quality before and after optimization. The percentage of the black pixels increases from 93.1% to 99.1%.

Fig. 7 Effects of our label set modification strategy for an image pair with large displacements. Pixels having a good label set are black, and others are white. (c) and (d) visualize label set quality before and after optimization. The percentage of the black pixels increases from 53.6% to 78.5%.

sets are good. For a more challenging image pair with large displacement (see Fig. 7), label set modification increases the ratio from 53.6% to 78.5%.

We further validate the effectiveness of our strategy by comparing the energy convergence with and without label set modification. We performed experiments on image pairs with large pixel displacements (≈ 200 px) and small pixel displacements (< 10 px). Figure 8 shows the change of energy during iteration. We can see that in both cases, the energy decreases dramatically after employing our dynamic label set framework.

We also compare the results visually and quantitatively. Figure 9 shows the visual comparison between the interpolated images from wide-baseline image pair with and without using the modification process. We can see that there are more artifacts in the result without using our label set modification strategy. The quantitative comparison on the Middlebury dataset [23] is shown in Table 2 and Table 1. All these results demonstrate the effectiveness of our

Fig. 8 Energy change during optimization. (a) Image pair with large displacement. (b) Image pair with small displacement.

Fig. 9 Comparison between (a) image interpolated using the baseline method with a constant label set, (b) generated by our method with label set modification.

label set modification strategy for introduce more correct labels to the candidate label set.

4.1.2 Validation of homography check

In Section 3.4, we use a homography check to divide superpixels into reliable and unreliable ones in order to guide the process of label set modification. Here we experimentally validate the effects of the homography check.

We performed an extra set of experiments, in which we do not conduct the homography check, i.e., we consider all fitted homographies to be reliable. As in Section 4.1.1, we first compare the energy change during iteration. Figure 8(b) shows that, for cases whose pixel displacements are small, there is not much difference in performance between the methods with and without the homography check process. The reason is that for these relatively easier cases, there are sufficient inlier pixels in each superpixel to fit a reliable homography, because there

are sufficient pixels whose initial candidate label sets are good enough (as shown in Fig. 6). However, the homography check process is more useful for wide-baseline image pairs. As we can see in Fig. 8(a), in more challenging cases whose pixel displacements are much larger, the energy is lower when we conduct the homography check process. Moreover, comparing interpolated images in Fig. 11, we can easily see that with the homography check process, our method generates many fewer artifacts such as distortion and holes.

We also compare the performance quantitatively on the Middlebury dataset [23]. See Table 2 and Table 1. Conducting the homography check process improves the accuracy of both the estimated motion fields and interpolated images.

4.2 Comparison to prior work

In this section, we first compare our method with prior work by evaluating images interpolated for

wide-baseline image pairs by Refs. [50] and [37], qualitatively showing the effectiveness of our method for handling large displacements. In addition, we quantitatively compare our method with other algorithms by evaluating the estimated motion fields and interpolated images on the Middlebury benchmark database [23]. We show that our method also achieves good performance on image pairs containing small motions, demonstrating the robustness of our method.

4.2.1 Qualitative evaluation

Nie et al. proposed a wide-baseline image interpolation algorithm, which is the state of the art for our problem. The second column of Fig. 10 shows the results of Ref. [2], while the last column shows our results. We can see that the method of

Ref. [2] generates more artifacts such as distortion and blurring than ours. Our method handles these cases much better, due to the spatial smoothness constraint which we enforce explicitly during optimization.

Since optical flow methods can also be used to interpolate images between image pairs, we also compare our method with Maskflownet [9], the state-of-the-art optical flow method based on deep learning, and some variational model methods similar to ours. In our experiments, we computed optical flow between image pairs using these optical flow methods and interpolated the intermediate images using the interpolation method in Section 3.6. The first column of Fig. 10 shows the results of Maskflownet. We use the pre-trained model trained on Flying Chairs [51], Flying Things3D [52], and MPI

Fig. 10 Comparison between Refs. [2], [9], and our method. There is less distortion in our results.

Fig. 11 Comparison of images interpolated by our approach (a) without the homography check, (b) with the homography check.

Sintel [49], provided by the authors of Ref. [9], to infer optical flow. As shown in Fig. 10, Maskflownet generates more artifacts than our method when interpolating between wide-baseline images. The performance of Maskflownet dramatically reduces when the displacement between the image pair is too large, as shown in the third row of Fig. 10; our method can handle these wide-baseline cases very well. One possible reason is the lack of training data suitable for many amateur datasets, which are more common. Our method takes only two images as input, which making our method more flexible.

We also compared our method with two variational model optical flow methods, DiscreteFlow [5] and SPM-BP [21], which are similar to our optimization scheme. DiscreteFlow is a representative large displacement optical flow method, which considers large-displacement optical flow from a discrete point of view. It proposes a diverse candidate label set which is quite large for each pixel, and performs optimization on this constant label set. Since their candidate label set is much larger than ours, optimization has to been performed on a sampled image grid and they need to get the final flow field by interpolation, while our method performs optimization directly on the full image grid. Our

method outperforms DiscreteFlow visually: Fig. 12 shows a comparison. The second column shows the results of DiscreteFlow while the third column shows our results. We can see that our approach produces fewer artifacts such as distortion.

Like our method, PMBP [20] uses the idea of dynamic label set update, but they utilizes PatchMatch to propose new labels. SPM-BP takes advantage of efficient edge-aware cost filtering to speed up PMBP and improves the performance. The first column of Fig. 12 shows results from SPM-BP. We can see that our method perform much better than theirs, due to our strategy of homography guided label proposal, which is more effective than SPM-BP's approach based on the idea of PatchMatch [34].

4.2.2 Quantitative evaluation

We now quantitatively compare our method with other work by evaluating results on two kinds of different datasets. Our method is designed for wide-baseline image interpolation. However, the baseline between pairs of images in commonly used optical flow datasets, such as KITTI [3] and MPI Sintel [49], is not wide enough [2], so we use wide-baseline synthetic image pairs photo-realistically rendered from virtual scenes to quantitatively evaluate our method. MVS-Synth [53] is a photo-realistic synthetic

Fig. 12 Comparison with Refs. [5] and [21]. There are fewer artifacts in our results.

Table 1 Interpolation error (PSNR) for the Middlebury benchmark. Image pairs: B: Beanbags, D: Dimetrodon, DD: DogDance, G2: Grove2, G3: Grove3, H: Hydrangea, M: MiniCooper, R: RubberWhale, U2: Urban2, U3: Urban3, V: Venus, W: Walking

Method	B	D	DD	G2	G3	H	M	R	U2	U3	V	W	Average
PMBP [20]	25.01	30.58	25.67	26.02	23.12	29.11	22.14	29.00	30.80	27.37	26.74	28.93	27.04
Nie et al. [2]	26.27	30.40	28.35	31.47	27.46	31.72	17.22	27.76	34.89	30.75	29.25	26.09	28.47
Maskflownet [9]	29.68	36.51	29.85	28.62	26.82	33.90	27.94	34.01	34.40	33.26	31.50	32.19	31.56
SPM-BP [21]	27.29	38.13	30.23	32.11	28.75	34.60	26.10	27.15	37.19	34.40	33.42	30.81	31.68
Discrete flow [5]	28.27	38.57	30.77	32.27	28.77	35.39	30.19	40.87	37.44	34.41	33.78	31.66	33.53
Ours w/o modification	29.02	38.57	30.92	32.53	28.41	35.43	30.21	41.90	37.49	35.63	34.28	31.01	33.78
Ours w/o check	29.29	38.59	31.05	32.53	28.25	35.43	30.25	41.90	37.72	36.04	34.31	32.00	33.95
Ours	29.52	38.59	31.05	32.54	29.02	35.43	30.25	41.90	37.78	36.04	34.31	32.00	34.03

Table 2 Motion error (EPE) for the Middlebury benchmark

Method	Dimetrodon	Grove2	Grove3	Hydrangea	RubberWhale	Urban2	Urban3	Venus	Average
PMBP [20]	0.5868	1.3295	2.6422	0.5478	0.2535	2.0244	3.8433	2.2079	1.8020
Nie et al. [2]	0.1759	0.2810	1.1288	0.2595	0.2487	0.5111	1.8042	1.6309	0.7617
Maskflownet [9]	0.2236	0.3309	0.9592	0.2591	0.2630	0.4474	0.9361	0.3279	0.5078
SPM-BP [21]	0.1744	0.2750	0.5872	0.2733	0.2195	0.4727	0.5638	0.2338	0.3752
Discrete flow [5]	0.1399	0.2421	0.7246	0.2231	0.1828	0.3405	0.4260	0.3078	0.3432
Ours w/o modification	0.0829	0.1791	0.8264	0.2114	0.1250	0.5780	0.7761	0.4465	0.4349
Ours w/o check	0.0815	0.1830	0.8834	0.2154	0.1217	0.5371	0.8271	0.4273	0.4440
Ours	0.0807	0.1500	0.6274	0.1601	0.1029	0.2934	0.7623	0.3760	0.3420

dataset that provides ground truth depth maps and the camera parameters for each rendered RGB image. Therefore, we can generate ground truth motion fields between image pairs using the provided ground truth geometry. We compare our method with previous works using wide-baseline image pairs rendered from 20 different scenes, where the average ground truth pixel displacement is about 300 pixels. We give the average end-point error (EPE) of the motion fields estimated by different methods in Table 3, which shows that our method quantitatively outperforms these previous methods.

The Middlebury dataset [23] is a widely used dataset for traditional optical flow method evaluation. Since it provides ground truth for the intermediate images, we also make comparisons using them although the average ground truth pixel displacement is only about 10 pixels. In Table 1, we list the peak signal to noise ratio (PSNR) between the interpolated images and the ground truth for different methods.

Table 3 Motion error (EPE) for the MVS-Synth dataset

Method	Average EPE
PMBP [20]	109.0240
Maskflownet [9]	44.7966
SPM-BP [21]	44.5387
Discrete flow [5]	30.4418
Nie et al. [2]	27.4805
Ours	26.6020

We also compute the average EPE of estimated motion fields on image pairs with ground truth motion fields for different algorithms, as shown in Table 2. Our method quantitatively outperforms these previous algorithms.

5 Conclusions

We have proposed a novel method of image interpolation, based on a motion estimation algorithm using homography guided optimization. We combine the advantage of both global optimization and a local parametric transformation model. Optimization is performed over very small candidate label sets, which are iteratively modified to avoid local minima, using piecewise consistency priors with superpixel as the bridge. We show experimentally that the proposed method improves the accuracy of both estimated motion fields and interpolated images.

Our method also has limitations. First, our strategy for new label proposal based on homography fitting and propagation uses superpixels as a fundamental structure. Therefore, our method's performance relies on the quality of superpixel segmentation. In addition, corresponding areas in image pairs representing different scenes may not be associated with a homography: our approach does not handle matching between different scenes very well, which is also a target of our future work.

Acknowledgements

This project was supported by the National Key Technology Research and Development Program of China (No. 2017YFB1002601), PKU–Baidu Fund (No. 2019BD007), and National Natural Science Foundation of China (NSFC) (No. 61632003).

References

- [1] Jiang, H. Z.; Sun, D. Q.; Jampani, V.; Yang, M. H.; Learned-Miller, E.; Kautz, J. Super SloMo: High quality estimation of multiple intermediate frames for video interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 9000–9008, 2018.
- [2] Nie, Y.; Zhang, Z.; Sun, H.; Su, T.; Li, G. Homography propagation and optimization for wide-baseline street image interpolation. *IEEE Transactions on Visualization and Computer Graphics* Vol. 23, No. 10, 2328–2341, 2017.
- [3] Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3354–3361, 2012.
- [4] Bao, W. B.; Lai, W. S.; Ma, C.; Zhang, X. Y.; Gao, Z. Y.; Yang, M. H. Depth-aware video frame interpolation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3698–3707, 2019.
- [5] Menze, M.; Heipke, C.; Geiger, A. Discrete optimization for optical flow. In: *Pattern Recognition. Lecture Notes in Computer Science, Vol. 9358*. Gall, J.; Gehler, P.; Leibe, B. Eds. Springer Cham, 16–28, 2015.
- [6] Liu, C.; Yuen, J.; Torralba, A. SIFT flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 33, No. 5, 978–994, 2011.
- [7] Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; Schmid, C. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1164–1172, 2015.
- [8] Horn, B. K. P.; Schunck, B. G. Determining optical flow. *Artificial Intelligence* Vol. 17, Nos. 1–3, 185–203, 1981.
- [9] Zhao, S. Y.; Sheng, Y. L.; Dong, Y.; Chang, E. I. C.; Xu, Y. MaskFlowNet: Asymmetric feature matching with learnable occlusion mask. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6277–6286, 2020.
- [10] Sun, D. Q.; Yang, X. D.; Liu, M. Y.; Kautz, J. PWC-net: CNNs for optical flow using pyramid, warping, and cost volume. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 8934–8943, 2018.
- [11] Hui, T. W.; Tang, X. O.; Loy, C. C. A lightweight optical flow CNN- revisiting data fidelity and regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* doi: 10.1109/TPAMI.2020.2976928, 2020.
- [12] Yang, G.; Ramanan, D. Volumetric correspondence networks for optical flow. In: Proceedings of the 33rd Conference on Neural Information Processing Systems, 794–805, 2019.
- [13] Chen, Q. F.; Koltun, V. Full flow: Optical flow estimation by global optimization over regular grids. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4706–4714, 2016.
- [14] Chen, J.; Cai, Z.; Lai, J.; Xie, X. A filtering-based framework for optical flow estimation. *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 29, No. 5, 1350–1364, 2019.
- [15] Chen, J.; Cai, Z. M.; Lai, J. H.; Xie, X. H. Fast optical flow estimation based on the split bregman method. *IEEE Transactions on Circuits and Systems for Video Technology* Vol. 28, No. 3, 664–678, 2018.
- [16] Bleyer, M.; Rhemann, C.; Rother, C. PatchMatch stereo-stereo matching with slanted support windows. In: Proceedings of the British Machine Vision Conference, 14.1–14.11, 2011.
- [17] Tanai, T.; Matsushita, Y.; Naemura, T. Graph cut based continuous stereo matching using locally shared labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1613–1620, 2014.
- [18] Veksler, O. Reducing search space for stereo correspondence with graph cuts. In: Proceedings of the British Machine Vision Conference, 709–718, 2006.
- [19] Kothapa, R.; Pacheco, J.; Sudderth, E. Max-product particle belief propagation. Master project report. Dept. of Computer Science, Brown University, 2011. Available at <http://cs.brown.edu/research/pubs/theses/masters/2011/kothapa.pdf>.
- [20] Besse, F.; Rother, C.; Fitzgibbon, A.; Kautz, J. PMBP: PatchMatch belief propagation for correspondence field estimation. *International Journal of Computer Vision* Vol. 110, No. 1, 2–13, 2014.
- [21] Li, Y.; Min, D. B.; Brown, M. S.; Do, M. N.; Lu, J. B. SPM-BP: Sped-up PatchMatch belief propagation for continuous MRFs. In: Proceedings of the IEEE International Conference on Computer Vision, 4006–4014, 2015.
- [22] Hornáček, M.; Besse, F.; Kautz, J.; Fitzgibbon, A.; Rother, C. Highly overparameterized optical flow using PatchMatch belief propagation. In: *Computer Vision - ECCV 2014. Lecture Notes in Computer Science, Vol. 8691*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 220–234, 2014.

- [23] Baker, S.; Scharstein, D.; Lewis, J. P.; Roth, S.; Black, M. J.; Szeliski, R. A database and evaluation methodology for optical flow. *International Journal of Computer Vision* Vol. 92, No. 1, 1–31, 2011.
- [24] Hur, J.; Roth, S. Optical flow estimation in the deep learning age. In: *Modelling Human Motion*. Noceti, N.; Sciutti, A.; Rea, F. Eds. Springer Cham, 119–140, 2020.
- [25] Jeong, S. G.; Lee, C.; Kim, C. S. Motion-compensated frame interpolation based on multihypothesis motion estimation and texture optimization. *IEEE Transactions on Image Processing* Vol. 22, No. 11, 4497–4509, 2013.
- [26] Mahajan, D.; Huang, F. C.; Matusik, W.; Ramamoorthi, R.; Belhumeur, P. Moving gradients: A path-based method for plausible image interpolation. *ACM Transactions on Graphics* Vol. 28, No. 3, Article No. 42, 2009.
- [27] Stich, T.; Linz, C.; Wallraven, C.; Cunningham, D.; Magnor, M. Perception-motivated interpolation of image sequences. *ACM Transactions on Applied Perception* Vol. 8, No. 2, Article No. 11, 2011.
- [28] Meyer, S.; Wang, O.; Zimmer, H.; Grosse, M.; Sorkine-Hornung, A. Phase-based frame interpolation for video. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1410–1418, 2015.
- [29] Long, G. C.; Kneip, L.; Alvarez, J. M.; Li, H. D.; Zhang, X. H.; Yu, Q. F. Learning image matching by simply watching video. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9910*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 434–450, 2016.
- [30] Niklaus, S.; Mai, L.; Liu, F. Video frame interpolation via adaptive convolution. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2270–2279, 2017.
- [31] Niklaus, S.; Mai, L.; Liu, F. Video frame interpolation via adaptive separable convolution. In: *Proceedings of the IEEE International Conference on Computer Vision*, 261–270, 2017.
- [32] Chen, Z. Y.; Jin, H. L.; Lin, Z.; Cohen, S.; Wu, Y. Large displacement optical flow from nearest neighbor fields. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2443–2450, 2013.
- [33] Bao, L. C.; Yang, Q. X.; Jin, H. L. Fast edge-preserving PatchMatch for large displacement optical flow. *IEEE Transactions on Image Processing* Vol. 23, No. 12, 4996–5006, 2014.
- [34] Barnes, C.; Shechtman, E.; Finkelstein, A.; Goldman, D. B. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics* Vol. 28, No. 3, Article No. 24, 2009.
- [35] Penner, E.; Zhang, L. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics* Vol. 36, No. 6, Article No. 235, 2017.
- [36] Hedman, P.; Kopf, J. Instant 3D photography. *ACM Transactions on Graphics* Vol. 37, No. 4, Article No. 101, 2018.
- [37] Chaurasia, G.; Duchene, S.; Sorkine-Hornung, O.; Drettakis, G. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics* Vol. 32, No. 3, Article No. 30, 2013.
- [38] Wang, S.; Wang, R. G. Robust view synthesis in wide-baseline complex geometric environments. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2297–2301, 2019.
- [39] Zhou, T. H.; Tulsiani, S.; Sun, W. L.; Malik, J.; Efros, A. A. View synthesis by appearance flow. In: *Computer Vision – ECCV 2016. Lecture Notes in Computer Science, Vol. 9908*. Springer Cham, 286–301, 2016.
- [40] Xu, Z. X.; Bi, S.; Sunkavalli, K.; Hadap, S.; Su, H.; Ramamoorthi, R. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics* Vol. 38, No. 4, Article No. 76, 2019.
- [41] Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; Ng, R. NeRF: Representing scenes as neural radiance fields for view synthesis. In: *Computer Vision – ECCV 2020. Lecture Notes in Computer Science, Vol. 12346*. Vedaldi, A.; Bischof, H.; Brox, T.; Frahm, J. M. Eds. Springer International Publishing, 405–421, 2020.
- [42] Yu, A.; Ye, V.; Tancik, M.; Kanazawa, A. pixelNeRF: Neural radiance fields from one or few images. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.
- [43] Felzenszwalb, P. F.; Huttenlocher, D. R. Efficient belief propagation for early vision. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.
- [44] Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 34, No. 11, 2274–2282, 2012.
- [45] Fischler, M. A.; Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* Vol. 24, No. 6. 381–395, 1981.
- [46] Hu, Y.; Li, Y.; Song, R. Robust interpolation of correspondences for large displacement optical flow. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4791–4799, 2017.

- [47] Kaviani, H. R.; Shirani, S. Iterative mask generation method for handling occlusion in optical ow assisted view interpolation. In: *Proceedings of the IEEE International Conference on Image Processing*, 3387–3391, 2015.
- [48] Burt, P. J.; Adelson, E. H. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics* Vol. 2, No. 4, 217–236, 1983.
- [49] Butler, D. J.; Wulff, J.; Stanley, G. B.; Black, M. J. A naturalistic open source movie for optical flow evaluation. In: *Computer Vision – ECCV 2012. Lecture Notes in Computer Science, Vol. 7577*. Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; Schmid, C. Eds. Springer Berlin Heidelberg, 611–625, 2012.
- [50] Menze, M.; Geiger, A. Object scene flow for autonomous vehicles. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3061–3070, 2015.
- [51] Dosovitskiy, A.; Fischer, P.; Ilg, E.; Häusser, P.; Hazirbas, C.; Golkov, V.; Smagt, P.; Cremers, D.; Brox, T. FlowNet: Learning optical flow with convolutional networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, 2758–2766, 2015.
- [52] Mayer, N.; Ilg, E.; Häusser, P.; Fischer, P.; Cremers, D.; Dosovitskiy, A.; Brox, T. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4040–4048, 2016.
- [53] Huang, P. H.; Matzen, K.; Kopf, J.; Ahuja, N.; Huang, J. B. DeepMVS: Learning multi-view stereopsis. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2821–2830, 2018.

Yuan Chang is currently a Ph.D. student at Peking University. He received his B.S. degree from the School of Mathematical Sciences, University of Science and Technology of China, in 2016. His research interests include computer vision, computer graphics, and image computing.

Congyi Zhang is a postdoctoral fellow at the University of Hong Kong. He received his B.Sc. degree from the School of Mathematical Science, Fudan University, in 2012, and his Ph.D. degree from the School of Electronics Engineering and Computer Science, Peking University, in 2019. His research interests

include computer vision, human–computer interaction, and computer graphics.

Yisong Chen received his B.S. degree in information engineering from Xi'an Jiaotong University, China, in 1996, and his Ph.D. degree in computer science from Nanjing University, China, in 2001. From 2001 to 2003, he was a post-doctoral researcher with the Human–Computer Interaction and Multimedia Laboratory, Peking University. From 2003 to 2005, he was a research fellow with the Image Computing Group, City University of Hong Kong. From 2005 to 2006, he was a senior research fellow with the Heudiasyc Laboratory, Centre National de la Recherche Scientifique, University of Technology of Compiègne, France. In 2007, he joined the Department of Computer Science, Peking University, as an associate professor. His research interests include computer vision, image computing, and pattern recognition.

Guoping Wang is a professor of computer science, Peking University, and director of the Graphics and Interactive Technology Laboratory, Peking University. He obtained his Ph.D degree from the Institute of Mathematics, Fudan University in 1997. He became a full professor in Dept. of Computer Science, Peking University in 2002. He was awarded by the National Science Fund for Distinguished Young Scholars in 2009. His research interests include virtual reality, computer graphics, human–computer interaction and multimedia.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.