Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Research Article

Homography-guided stereo matching for wide-baseline image interpolation

Yuan Chang[1,2], **Congyi Zhang**[1,2,3], **Yisong Chen**[1,2], **Guoping Wang**[1,2] (✉)

Abstract Image interpolation techniques have 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 local parametric transformation model. We perform the optimization over dynamic label sets, which are modified after each iteration using the prior of piecewise consistency to avoid the 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 we 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] and view synthesis [2] *etc.* In some applications, the available images are wide-baseline spaced. 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 street spot by clicking the directional arrow. In order to make the transition between discrete views smooth, it is important to interpolate the intermediate views between wide-baseline image pairs realistically since the set of 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 estimating 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 between wide-baseline image pairs, 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 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 full image grid in this space always needs dramatically high computational complexity. To reduce the amount of computation, some approaches use a search window as

1 Peking University, Beijing, 100871, China. E-mail: Yuan Chang, changyuan@pku.edu.cn, Yisong Chen, chenyisong@pku.edu.cn, Guoping Wang, wgp@pku.edu.cn.

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

3 the University of Hong Kong, Hong Kong. E-mail: Congyi Zhang, cyzh@hku.hk.

Manuscript received: 2014-12-31; accepted: 2015-01-30.

the candidate label set [6]. However, for wide-baseline image pairs, the window size should be very large to avoid falling into the local minima, which makes the optimization prohibitively slow. Other approaches use approximate nearest neighbors in feature space to prune the set of potential motions [5]. But the proposed set is still superfluous, because it needs to maintain a high recall of the target motions. So they have to perform the optimization on the sampled image grid, and use the interpolation method [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 can not guarantee the estimated motion field to be piecewise smooth, which may lead to some artifacts of stretching, overlapping and holes, *etc.* Therefore, methods using this strategy usually need an extra global optimization to further eliminate the 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 to a global optimization in Markov Random Field. Different from using a constant set of candidate motions as previous methods did [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. Different from DiscreteFlow [5], where the candidate set is superfluous, the size of our candidate set can be very small. Then, we perform global optimization over full image grid with the proposed candidate sets. Considering that 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. First, we propose a novel optimization framework for motion estimation based on homography guided belief propagation. Second, we apply the proposed motion estimation method to an image interpolation framework, and we show experimentally that our approach is able to deal well with the wide-baseline image interpolation problem. We demonstrate that our approach also performs well for traditional small-baseline image pairs too, through experiments on typical optical flow dataset.

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

2 Related Work

As we 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 the relevant low-level motion estimation algorithms in this section. Then we mainly review the related work about image interpolation including frame rate-up conversion and view synthesis.

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 have been a huge body of literature on optical flow [9–12]. One typical approach is to consider it as a labeling problem like we mentioned in Sec. 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 were proposed to reduce the label set. The simplest way is using a searching window centered at the initial label [6]. But it is easily prone 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 neighbors search and random sampling around the reference pixel. Veksler *et al.* [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 the 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. There have been some methods proposed 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 different from 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 [23, 24].

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 the most similar block in the following frame for each block. On pixel level, Mahajan *et al.* [26] move the image gradients to a given time step and solve a Poisson equation to reconstruct the interpolated frame. Stich *et al.* [27] find edges and homogeneous regions in images for matching, yielding a dense motion field between images. Meyer *et al.* [28] propose propagating phase information across oriented multi-scale pyramid levels for video interpolation. Moreover, CNN-based methods also showed good performance in this application. Long *et al.*train a deep CNN to directly predict the interpolated frames [29], but the results are usually blurry. Then some methods take advantage of the accurate estimated pixel-wise optical flow to improve the performance [1, 4]. Besides, some methods formulate frame interpolation as convolution operations over local patches and estimate the convolutional kernels for each output pixel [30, 31]. However, these methods are designed for small-baseline image pairs, and they are not effective for wide-baseline image interpolation.

View synthesis. View synthesis is the process of generating a new view using existing views taken by multiple cameras. In this situation, there may be large displacement because of large transition 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 approximate nearest neighbor field [32], which help 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] obtain large displacement optical flow by increasing the smoothness of PatchMatch [34]. But still, these methods do not perform very well for wide-baseline image interpolation. Image-based rendering techniques [35–38] are proposed to get better result of wide-baseline view synthesis. Chaurasia *et al.* [37] reconstruct 3D model for a scene, and compensated for the errors of the reconstruction by depth synthesis. However, sometimes we may fail to reconstruct the 3D scene for some reasons like insufficient number of images. Some researchers try to apply deep learning methods to view synthesis problem [39–42]. For example, Zhou *et al.* [39] train a convolutional neural network to generate an appearance flow vector that specifies which pixels in the input image could 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 sourse image into superpixels, and estimate for each superpixel a homography, which transforms each superpixel to the target position. However, without enforcing spatial smoothness constraint explicitly, there may be some artifacts because of the discontinuity between different superpixels. Although there is a mesh warping framework to further eliminate the artifacts, there are still some artifacts like stretching and holes. Our method is similar to [2], since we both use the assumption that each superpixel represents a small plane, and our method also includes the homography fitting and propagation. But unlike them, we formulate the whole process of motion estimation as an energy minimization problem, which explicitly enforcing spatial smoothness constraint and achieve better performance than that of [2].

3 Proposed Approach

Our aim is to generate intermediate images between two given images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$. To that end, we compute a forward displacement vector from $\boldsymbol{I}_1$ to $\boldsymbol{I}_2$ for each pixel in $\boldsymbol{I}_1$ and a backward displacement vector from

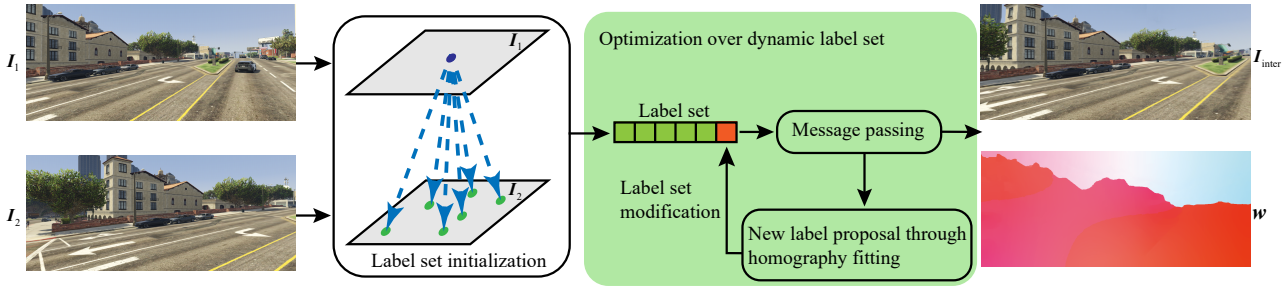


Fig. 1 The pipeline of our approach. The label set initialization phase composes label set using N nearest neighbor in feature space. And in addition to iterative optimizing the objective function, the optimization phase marks the worst candidate in label set by their cost and replaces it by new label proposal in each iteration.

$\boldsymbol{I}_2$ to $\boldsymbol{I}_1$ for each pixel in $\boldsymbol{I}_2$. Our approach considers it as a labeling problem, where the label here means the displacement vector for each pixel, and we solve this problem by minimizing an energy function in the 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 the local minima. The pipeline is shown in Fig. 1. First of all, we propose an initial candidate label set whose size is very small for each pixel. We introduce this part in Sec. 3.2. Then, to tackle the problem of insufficient candidates caused by 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. The details of these processes will be introduced in Sec. 3.3.

Before presenting the details of the algorithm, we first introduce the formulation of our motion estimation approach and some essential knowledge of BP in Sec. 3.1.

3.1 Formulation of motion estimation

Without loss of generality, we only introduce the estimation of forward displacement vectors from $\boldsymbol{I}_1$ to $\boldsymbol{I}_2$, since the backward displacement from $\boldsymbol{I}_2$ to $\boldsymbol{I}_1$ can be obtained using exactly the same way. Our goal is to estimate the motion field $\boldsymbol{w}$ for $\boldsymbol{I}_1$, where $\boldsymbol{w}(\boldsymbol{p}) = (u(\boldsymbol{p}), v(\boldsymbol{p}))$ is the displacement vector at pixel $\boldsymbol{p}$ and $\boldsymbol{p} = (x, y)$ represents the grid coordinate of image $\boldsymbol{I}_1$. Since we formulate this problem as a global optimization in MRF, we can also see $\boldsymbol{w}(\boldsymbol{p})$ as a label of pixel $\boldsymbol{p}$. The energy function to be minimized is formulated as Eq. (1), including 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 contains all the neighborhoods on a four-connected image grid, and λ is a weight coefficient of the smoothness term.

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

Let $\boldsymbol{C}(\boldsymbol{p}) = \{\boldsymbol{w}_1^p, \boldsymbol{w}_2^p, \cdots, \boldsymbol{w}_L^p\}$ be the candidate label set of each pixel $\boldsymbol{p}$ in image $\boldsymbol{I}_1$, containing L candidate labels. For clarity, we set the size of every pixel's label set to the same L, although they can be different in our algorithm.

Belief propagation is an inference algorithm working by passing message around the 4-connected image grid iteratively [43]. It updates a L-dimensional message $\boldsymbol{m}_{p\to q}^t(\boldsymbol{w}_i^q), 1 \leqslant i \leqslant L$, sent from each pixel $\boldsymbol{p}$ to its neighbor $\boldsymbol{q}$ at each iteration t from$[0, T]$. The messages are computed in the following way, where $\mathcal{N}(p)\backslash q$ denotes the neighbors of $\boldsymbol{p}$ other than $\boldsymbol{q}$.

$$\begin{aligned}\boldsymbol{m}_{p\to q}^t(\boldsymbol{w}_i^q) = \min_{1\leqslant j\leqslant L}(E_s(\boldsymbol{w}_j^p, \boldsymbol{w}_i^q,) + E_d(\boldsymbol{w}_j^p) \\ + \sum_{s\in\mathcal{N}(p)\backslash q} \boldsymbol{m}_{s\to p}^{t-1}(\boldsymbol{w}_j^p)).\end{aligned} \quad (2)$$

Then, with the obtained $\boldsymbol{m}_{p\to q}^t$, we can compute a belief vector $\boldsymbol{b}_p^t(\boldsymbol{w}_i^p)$ for each pixel $\boldsymbol{p}$ at each iteration t using

$$\boldsymbol{b}_p^t(\boldsymbol{w}_i^p) = E_d(\boldsymbol{w}_i^p) + \sum_{(\boldsymbol{p},\boldsymbol{q})\in\epsilon} \boldsymbol{m}_{q\to p}^t(\boldsymbol{w}_i^p). \quad (3)$$

The value of $\boldsymbol{b}_p^t(\boldsymbol{w}_i^p)$ represents an approximation to the probability that the correct label for $\boldsymbol{p}$ is $\boldsymbol{w}_i^p$. After T iterations, the final belief vector $\boldsymbol{b}_p^T(\boldsymbol{w}_i^p)$ can be calculated for each pixel, and we can select the best label $\boldsymbol{w}^*(\boldsymbol{p})$ for every pixel $\boldsymbol{p}$ from its label set $\boldsymbol{C}(\boldsymbol{p})$ by minimizing $\boldsymbol{b}_p^T(\boldsymbol{w}_i^p)$ pixel-wisely.

How we choose the label set $\boldsymbol{C}(\boldsymbol{p})$ is very important. The set can't be too large because the optimization will be prohibitively slow. But a fixed small candidate label set may lead to local minima easily. Therefore, our approach uses a compact dynamic candidate label

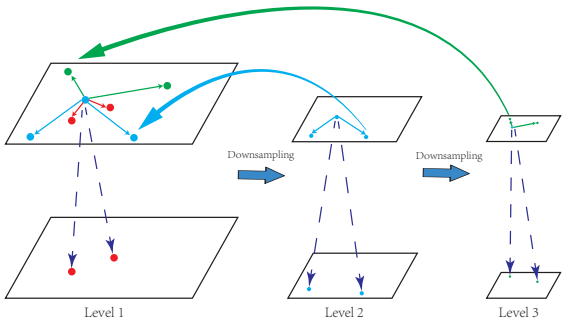


Fig. 2 Illustration of label set initialization.

set. We initialize a very small label set for each pixel, and we modify the label sets iteratively during BP to avoid the local minima.

3.2 Initialization

We use a multi-scale K nearest neighbor search strategy to initialize the candidate label sets, which is shown in Fig. 2. First, we construct image pyramids with N_L levels, where $N_L = 4$ in our experiments, for both $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ by downsampling from the original images using bilinear interpolation . Let $\boldsymbol{I}_i^\ell (i = 1, 2)$ be the downsampled image of $\boldsymbol{I}_i$ at each pyramid level ℓ. We compute a feature descriptor for each pixel in $\boldsymbol{I}_1^\ell$ and $\boldsymbol{I}_2^\ell$ to help finding correspondences. Because for wide-baseline image pair, the brightness of an object may change during the transition between views, and a feature descriptor is more robust to find nearest matches. To overcome local scale and rotation changes in wide baseline scenario, we use per-pixel Scale-Invariant Feature Transform (SIFT) descriptor [6] as the dense feature descriptor. After we get the feature maps $\boldsymbol{D}_1^\ell$ for $\boldsymbol{I}_1^\ell$ and $\boldsymbol{D}_2^\ell$ for $\boldsymbol{I}_2^\ell$, we search K_ℓ nearest neighbors in $\boldsymbol{D}_2^\ell$ for every descriptor in $\boldsymbol{D}_1^\ell$ under L_1 distance. Then we get K_ℓ labels corresponding to the K_ℓ nearest neighbors for each pixel in $\boldsymbol{I}_1^\ell$ at level ℓ, and we upsample it to the original scale of image $\boldsymbol{I}_1$ to propose K_ℓ initial labels for each pixel in $\boldsymbol{I}_1$. We collect the initial labels proposed from each level ℓ to get the initial candidate label set of each pixel in $\boldsymbol{I}_1$ with size $N = \sum_\ell K_\ell$. In our experiment, we search $K_\ell = 2$ labels for each level ℓ to get 8 candidates for each pixel. Note that the multi-scale scheme is only used in the initialization step. The following optimization does not require a coarse-to-fine scheme to prevent local minima, since we use the homography guided modification strategy, which is introduced in the next section.

3.3 Optimization

We first introduce the specific data term and smoothness term we use in our experiment. We use the truncated L_1 distance between the SIFT descriptors, which are computed in the initialization phase, to be matched 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 account for motion discontinuities. They are shown in Eq. (4) and Eq. (5), where $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are the feature maps of the original input images $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$, and τ_d and τ_s are the truncation threshold of the data term and the smoothness respectively.

$$E_d(\boldsymbol{w}(\boldsymbol{p})) = \min\left(\|\boldsymbol{D}_1(\boldsymbol{p}) - \boldsymbol{D}_2(\boldsymbol{p} + \boldsymbol{w}(\boldsymbol{p}))\|_1, \tau_d\right) \quad (4)$$

$$E_s(\boldsymbol{w}(\boldsymbol{p}), \boldsymbol{w}(\boldsymbol{q})) = \min\left(\|\boldsymbol{w}(\boldsymbol{p}) - \boldsymbol{w}(\boldsymbol{q})\|_1, \tau_s\right) \quad (5)$$

With the specific energy function, the optimization can be performed now. As we mentioned in Sec. 3.1, 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 to prevent local minima. In order to conduct the homography check and the label set modification, we first over-segment image $\boldsymbol{I}_1$ into superpixels $S = \{S_1, S_2, \cdots, S_K\}$ employing the method of [44], and we regard each superpixel as a small plane, which corresponds to somewhere in $\boldsymbol{I}_2$ by a homography, as they did in [2] since it is small and it usually has homogeneous color.

Homography check. As introduced in Sec. 3.1, we compute a belief vector $\boldsymbol{b}_{\boldsymbol{p}}^t(\boldsymbol{w})$ for every pixel $\boldsymbol{p}$ after each iteration t, and select the current best label $\boldsymbol{w}_t^*(\boldsymbol{p})$ from $\boldsymbol{C}(\boldsymbol{p})$ for $\boldsymbol{p}$. With the prior knowledge of the plane approximation in 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 will help to generate new labels while modifying the label sets, which we will introduce later. To ensure the validity of labels guessed by homographies, we need to identify whether a homography is reliable or not first.

After H_i is obtained, we can project each pixel $\boldsymbol{p}$ in S_i to a new location $\boldsymbol{p}'$ in $\boldsymbol{I}_2$ using H_i.

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

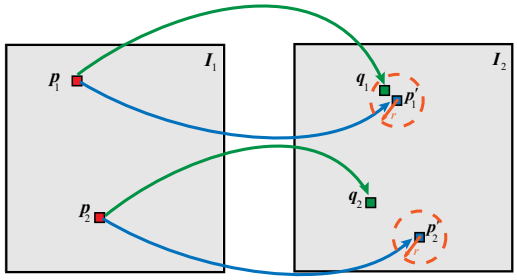


Fig. 3 Illustration of inlier/outlier pixel discrimination. $\boldsymbol{p}_1$ represents an inlier pixel while $\boldsymbol{p}_2$ represents an outlier pixel.

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

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

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

Then we can compute the reliability $Re(S_i)$ of the fitted homography H_i of a superpixel S_i, which calculates the percentage of inlier pixels in a superpixel:

$$Re(S_i) = \frac{\sum_{\boldsymbol{p} \in S_i} \delta(\boldsymbol{p})}{|S_i|}, \tag{8}$$

where $|S_i|$ is the number of pixels of 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 $\mathcal{R}$ which contains superpixels whose fitted homographies are reliable.

$$R = \{S_i | Re(S_i) > \zeta\} \tag{9}$$

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

Label set modification. After we divide all the superpixels into reliable ones $\mathcal{R}$ and unreliable ones $\mathcal{U}$, we modify the candidate label set by substituting new labels. Substituting a label $\boldsymbol{w}$ here means replacing the current worst label in the current candidate label set of each pixel with the new proposed label $\boldsymbol{w}$. Here, similar to the definition of the current best label, we select the current worst label by maximizing $\boldsymbol{b}_p^t(\boldsymbol{w}_i^{\boldsymbol{p}})$.

We use different ways to propose new labels for pixels in reliable superpixels or in unreliable superpixels.

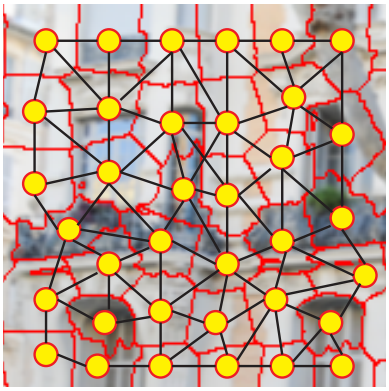


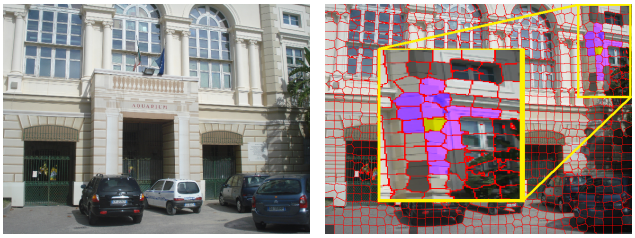
Fig. 4 Illustration of the superpixel graph. We use red lines to represent the boundaries of superpixels, and we use yellow points(graph nodes) and black lines(graph edges) to illustrate the graph structure.

As the first case, if a pixel $\boldsymbol{p}$ belongs to a reliable superpixel S_i, we directly use the homography H_i fitted in homography check to generate a new label by Eq. (10) since we consider the reliable homography as a good estimation of the transformation of $\boldsymbol{p}$ from $\boldsymbol{I}_1$ to $\boldsymbol{I}_2$.

$$\boldsymbol{w}_{new} = \boldsymbol{H}_i \boldsymbol{p} - \boldsymbol{p} \tag{10}$$

If $\boldsymbol{p}$ 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 generating new labels. To that end, we construct an undirected graph whose nodes are all superpixels and edges connecting the superpixels with shared boundaries, which is shown in Fig. 4. In this paper, the weight of each edge is defined as the color similarity between the connected superpixels. As [2] did, we create normalized color histogram for each superpixel, and we compute the χ^2 distance between two histogram of two adjacent superpixels as the color similarity. With 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 shortest path algorithm.

Then we propose the M new labels based on the similarity between any two superpixels. We search the M most similar superpixels from R for $S_i \in U$, and we project $\boldsymbol{p}$ using the M corresponding homographies H_i^j, $j = 1, 2, \cdots, M$, to generate the M new labels $\boldsymbol{w}_j$ using Eq. (11). We show this process in Fig. 5. For the unreliable superpixel S_i, which is marked as yellow in image (a) of Fig. 5, we search M superpixels(shown as



(a) (b)

Fig. 5 Illustration of generating new labels for unreliable superpixels. Image (a) shows The original input image $\boldsymbol{I}_1$, while image (b) shows the process of searching similar superpixels in R. The dark region in (b) shows the unreliable superpixels, and the yellow one represents the unreliable superpixel to be processed. The blue ones are the most similar superpixels of the yellow one we searched from R.

blue) in R which are most similar to S_i. Note that we do not use the neighboring superpixels directly to propose new labels for S_i, because 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 candidates for each pixel, we propose M new candidates for each pixel in unreliable superpixels to ensure as much as possible that we propagate the correct homography to the unreliable superpixel.

$$\boldsymbol{w}_j = \boldsymbol{H}_i^j \boldsymbol{p} - \boldsymbol{p} \tag{11}$$

Since we use the same homography to generate new labels for pixels in the same superpixel in both two cases, these labels will share good consistency between neighboring pixels so that the smoothness term may be reduced dramatically even these labels are not correct. Therefore, to avoid such case, during each iteration, we uniformly sample 30% pixels from the outlier pixels of reliable superpixels and 30% pixels from unreliable superpixels to be modified in practice.

3.4 Occlusion handling

Since we do not consider occlusions explicitly, the computed displacement vectors on occlusion pixels may be incorrect. Therefore, we remove the outliers from our result using the forward-backward consistency checking, i.e., we compute the forward displacement vectors from $\boldsymbol{I}_1$ to $\boldsymbol{I}_2$ and the backward vector from $\boldsymbol{I}_2$ to $\boldsymbol{I}_1$ and discard inconsistent ones. Then we use the state-of-the-art interpolation scheme [46] to interpolate the discarded regions.

3.5 Interpolation

With the computed displacement vectors $\boldsymbol{w}_1$ for $\boldsymbol{I}_1$ and $\boldsymbol{w}_2$ for $\boldsymbol{I}_2$, we can smoothly interpolate any intermediate image $\boldsymbol{I}_t$ at time $t \in (0, 1)$ between $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ using the patch-based reconstruction scheme [47]. For any pixel $\boldsymbol{p}$ in $\boldsymbol{I}_1$, its motion vector to $\boldsymbol{I}_t$ is $t \cdot \boldsymbol{w}_1(\boldsymbol{p})$. So we can map each pixel $\boldsymbol{p}$ in $\boldsymbol{I}_1$ to its new location $\boldsymbol{p} + t \cdot \boldsymbol{w}_1(\boldsymbol{p})$ in $\boldsymbol{I}_t$ to render the intermediate image. Likewise, we can render the intermediate image using $\boldsymbol{I}_2$ too.

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

4 Experiments

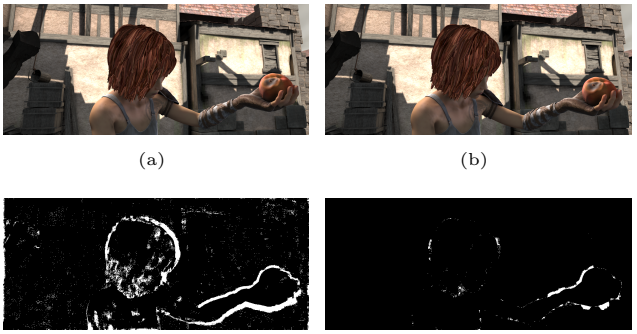
In this section, we first analyze the performance of our approach experimentally, and validate the claims we made before in Sec. 4.1. Then we evaluate our method by comparing to prior work in Sec. 4.2.

4.1 Performance analysis

4.1.1 Validation for label set modification

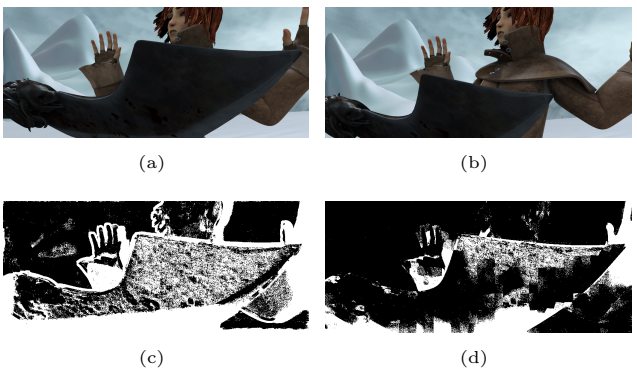
Since we use a very small candidate label set for each pixel, the initial label set may not include the correct label at all. Therefore, if we perform the optimization over the constant label sets, it is easily prone to local minima. However, our strategy of label set modification can help avoiding the 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 in 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 "fine label set". Pixels without fine label sets tend to stuck in local minima more likely than pixels with fine label sets. Therefore, we expect more pixels having fine label sets after label set modification. To demonstrate the quality of all pixels' label set clearly, we mark a pixel as black if its label set is "fine", otherwise we mark it as white as shown in Fig. 6 and Fig. 7. We compare the ratio of pixels with "fine label sets" in all pixels before the optimization process with the ratio after 10 iterations of the optimization, to see the effectiveness of our label set modification strategy, and we set the threshold γ to be 5 here. It is shown that



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

Fig. 6 Effects of our label set modification strategy with small displacements. The pixels with "fine label set" is in black and otherwise they are in white. (c) and (d) show the visualization of label set quality before and after our optimization respectively. The percentage of the black pixels increases from 93.051% to 99.107%.

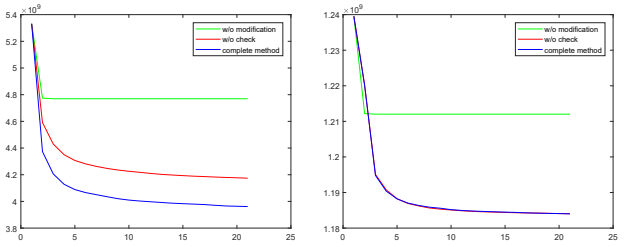


(a) (b) (c) (d)

Fig. 7 Effects of our label set modification strategy in image pairs with large displacements. The pixels with "fine label set" is in black and otherwise they are in white. (c) and (d) show the visualization of label set quality before and after our optimization respectively. The percentage of the black pixels increases from 53.640% to 78.529%.

our modification strategy improves the ratio of pixels with "fine label set" effectively. For image pair without large displacements shown in Fig. 6, 93.051% pixels' initial label sets are "fine label sets", while 99.107% pixels' modified label sets are "fine label sets". For more challenging image pair with large displacement shown in Fig. 7, the label set modification process increases the ratio from 53.640% to 78.529%.

Moreover, we further validate the effectiveness of our strategy by comparing the energy convergence using and not using the label set modification process. We perform experiments on image pairs with large pixel displacement($\sim$ 200px) and image pairs whose pixel displacements are small(<10px) respectively. Fig. 8 shows the change of energy during iterations. We can



(a) (b)

Fig. 8 The energy changing during optimization. (a) shows the energy changing of image pair with large displacement. (b) shows the energy changing of image pair with small displacement.

(a) (b)

Fig. 9 Comparison between interpolated image generated from the baseline method using a constant label set(a) and that generated from our method with label set modification process(b).

see that in both case, the energy decreases dramatically after employing our dynamic label set framework.

We also compare the results visually and quantitatively. Fig. 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 Tab. 2 and Tab. 1. All these results demonstrate the effectiveness of our label set modification strategy for introduce more correct labels to the candidate label set.

4.1.2 Validation for homography check

In Sec. 3.3, we use a homography check to divide superpixels into reliable and unreliable ones in order to guide the process of label set modification. Now we validate the effect of homography check experimentally.

We perform an extra set of experiments, where we do not conduct the homography check process. That means we consider all the fitted homographies as reliable ones. As we did in Sec. 4.1.1, we first compare the energy changing during iterations. The energy curves are shown in Fig. 8. As shown in Fig. 8 (b), for cases whose pixel displacements are small,

(a) result of Maskflownet [9] (b) result of Nie *et al.* [2] (c) result of ours

Fig. 10 Comparison between [2], [9] and our method. There are less distortions in our results.

there is not much difference of the performance between the methods using and not using 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 effective for wide-baseline image pairs. As we can see in Fig. 8 (a), for more challenging cases whose pixel displacements are much larger, the energy decreases after we conduct the homography check process. Moreover, we show the comparison of interpolated images too in Fig. 11. We can easily see that with the homography check process, our method generates much less artifacts such as distortions and holes.

We also compare the performance quantitatively on the Middlebury dataset [23], shown in Tab. 2 and Tab. 1. We can see that conducting the homography check process improves the accuracy of both the eitimated motion fields and the interpolated images.

4.2 Comparison to prior work

In this section, we first compare our method with prior work by evaluating the interpolated images of wide-baseline image pairs from [50] and [37], to show the effectiveness of our method for handling large displacement qualitatively. In addition, we quantitatively compare our method with other algorithms by evaluate the estimated motion fields and

(a) (b)

Fig. 11 Comparison between interpolated image generated from the baseline method without the homography check process(a) and that generated from our method with the homography check process(b).

the interpolated images on the Middlebury benchmark database [23]. We show that our method also achieves good performance on the image pairs containing small motions, which validates 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 of the problem that we focus on. The second column of Fig. 10 shows the results of [2], and the last column shows our results. We can see that the method of [2] generates more artifacts such as distortion and blur than ours. Our method handle these cases much better, and we owe it to the spatial smoothness constraint which we enforcing explicitly in the optimization.

Since optical flow methods can also be used to generate interpolated images between image pairs, we

(a) result of SPM-BP [21] (b) result of discrete flow [5] (c) result of ours

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

also compare our method with Maskflownet [9], the state-of-the-art optical flow method based on deep learning, and some variational model based methods which are 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 same interpolation method introduced in Sec. 3.5. The first column of Fig. 10 shows the results of Maskflownet. We use the pretrained model trained on Flying Chairs [51], Flying Things3D [52], and MPI Sintel dataset [49], which is provided by the authors of [9], to infer the optical flow. As shown in Fig. 10, Maskflownet generates more artifacts than our method when interpolating between wide-baseline images. And the performance of Maskflownet is dramatically reduced when the displacements between image pair are too large, as shown in the third row in Fig. 10, while our method can handle these wide-baseline cases very well. One possible reason is the lack of training data for many amateur datasets, which exist more widely. Our method takes only two images as input, which makes our method more flexible.

We also compared our method with two variational model based 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 looks at 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 an optimization on this constant label set. Since their candidate label set is much larger than us, the optimization has to been performed on the sampled image grid and they need to get the final flow field by interpolation, while our method perform optimization directly on the full image grid. Moreover, our method outperforms DiscreteFlow visually too. Fig. 12 shows the comparison. The second column shows the results of DiscreteFlow while the third column shows ours. We can see that our approach produces less artifacts like distortion. PMBP [20] uses the idea of dynamic label set update similar to ours, but they utilizes PatchMatch to propose new labels. SPM-BP takes advantages of efficient edge-aware cost filtering to speed up PMBP and improves the performance. The first column of Fig. 12 shows the results of SPM-BP. We can see that our method perform much better than theirs, and we owe it to our strategy of homography guided label proposal. Our strategy of label proposal is more effective than that of SPM-BP which is based on the idea of patchmatch [34].

Tab. 1 comparison by interpolation error(PSNR) on Middlebury

Method	Beanbags	Dimetrodon	DogDance	Grove2	Grove3	Hydrangea	MiniCooper	RubberWhale	Urban2	Urban3	Venus	Walking	Average
PMBP [20]	25.0140	30.5751	25.6710	26.0227	23.1232	29.1122	22.1404	29.0011	30.7986	27.3659	26.7408	28.9333	27.0415
Nie *et al.* [2]	26.2718	30.3983	28.3529	31.4746	27.4603	31.7164	17.2192	27.7642	34.8911	30.7479	29.2531	26.0918	28.4701
Maskflownet [9]	**29.6818**	36.5061	29.8506	28.6162	26.8164	33.9030	27.9355	34.0114	34.4048	33.2649	31.4976	**32.1876**	31.5563
spm-bp [21]	27.2857	38.1278	30.2325	32.1130	28.7542	34.6010	26.0951	27.1484	37.1867	34.3967	33.4212	30.8079	31.6808
discrete flow [5]	28.2706	38.5731	30.7737	32.2749	28.7675	35.3917	30.1913	40.8717	37.4425	34.4117	33.7835	31.6599	33.5344
ours w/o modification	29.0207	38.5744	30.9202	32.5339	28.4101	35.4251	30.2115	41.9006	37.4949	35.6278	34.2800	31.0051	33.7837
ours w/o check	29.2890	38.5881	31.0511	32.5321	28.2469	35.4255	30.2533	41.9011	37.7202	36.0365	34.3069	31.9986	33.9458
ours	29.5231	**38.5883**	**31.0511**	**32.5368**	**29.0217**	**35.4257**	**30.2533**	**41.9011**	**37.7757**	**36.0365**	**34.3070**	31.9986	**34.0349**

Tab. 2 comparison by motion error(EPE) on Middlebury

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**

Tab. 3 comparison by motion error(EPE) on 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**

4.2.2 Quantitative evaluation

We compare our method with other works quantitatively by evaluating the results on two kinds of different datasets. Since our method is designed for wide-baseline image interpolation while the baseline between pairs of images in commonly used optical flow datasets, such as KITTI [3] and MPI Sintel [49], is not wide enough as discussed in [2], we use wide-baseline synthetic image pairs photo-realistically rendered from virtual scenes to evaluate our method quantitatively. MVS-Synth [53] is a photo-realistic synthetic dataset that provides the ground truth depth map and the camera parameters for each rendered RGB image. Therefore, we can generate the 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 list the average end-point error(EPE) of the motion fields estimated by different methods in Tab. 3, which shows that our method outperform these previous methods quantitatively.

The Middlebury dataset [23] is a widely used dataset for traditional optical flow methods evaluation. Since it provides the ground truth of the intermediate image, we also make comparisons on it although the average ground truth pixel displacement is only about 10 pixels. In Tab. 1, we list the Peak Signal to Noise Ratio(PSNR) between the interpolated images and the ground truth for different methods. We also compute the average EPE of estimated motion fields on image pairs with ground truth motion fields for different algorithms, which is shown in Tab. 2. As shown in Tab. 2 and Tab. 1, our method outperform these previous algorithms quantitatively as well.

5 Conclusion

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 local parametric transformation model. The optimization is performed over very small candidate label sets, and the label sets are iteratively modified to avoid the local minima using piecewise consistency prior with superpixel as the bridge. We show experimentally that the proposed method improves the accuracy of both estimated motion fields and interpolated images.

We also have limitations. First, our strategy for new label proposal based on homography fitting and propagation uses superpixel as a fundamental structure. Therefore, our method's performance relies on the quality of superpixel segmentation. In addition, corresponding areas in image pair representing difference scenes may not be associated with homography. So our approach does not handle matching across different scenes very well, which is also our interesting 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, Sun D, Jampani V, Yang M, Learned-Miller E, Kautz J. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 9000–9008.

[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*, Oct 2017, 23(10):2328–2341.

[3] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[4] Bao W, Lai W S, Ma C, Zhang X, Gao Z, Yang M H. Depth-aware video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[5] Menze M, Heipke C, Geiger A. Discrete optimization for optical flow. In Gall J, Gehler P, Leibe B, editors, *Pattern Recognition*, 2015, pp. 16–28.

[6] Liu C, Yuen J, Torralba A. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May 2011, 33(5):978–994.

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

[8] Horn B K P, Schunck B G. Determining optical flow. *Artif. Intell.*, August 1981, 17(1-3):185–203.

[9] Zhao S, Sheng Y, Dong Y, Chang E I C, Xu Y. Maskflownet: Asymmetric feature matching with learnable occlusion mask. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[10] Sun D, Yang X, Liu M, Kautz J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[11] Hui T W, Tang X, Loy C C. A lightweight optical flow cnn - revisiting data fidelity and regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, pp. 1–1.

[12] Yang G, Ramanan D. Volumetric correspondence networks for optical flow. In *Advances in neural information processing systems*, 2019, pp. 794–805.

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

[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*, 2019, 29(5):1350–1364.

[15] Chen J, Cai Z, Lai J, Xie X. Fast optical flow estimation based on the split bregman method. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018, 28(3):664–678.

[16] Bleyer M, Rhemann C, Rother C. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC*, January 2011.

[17] Taniai T, Matsushita Y, Naemura T. Graph cut based continuous stereo matching using locally shared labels. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1613–1620.

[18] Veksler O. Reducing search space for stereo correspondence with graph cuts. In *In: British Machine Vision Conf*, 2006, pp. 709–718.

[19] Kothapa R, Pacheco J, Sudderth E et al. Max-product particle belief propagation. *Master's project report, Brown University Dept. of Computer Science*, 2011.

[20] Besse F, Rother C, Fitzgibbon A, Kautz J. Pmbp: Patchmatch belief propagation for correspondence field estimation. *International Journal of Computer Vision*, Oct 2014, 110(1):2–13.

[21] Li Y, Min D, Brown M S, Do M N, Lu J. Spm-bp: Sped-up patchmatch belief propagation for continuous mrfs. In *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 4006–4014.

[22] Hornacek M, Besse F, Kautz J, Fitzgibbon A W, Rother C. Highly overparameterized optical flow using patchmatch belief propagation. In *ECCV*, 2014.

[23] Baker S, Scharstein D, Lewis J P, Roth S, Black M J, Szeliski R. A database and evaluation methodology for optical flow. *Int. J. Comput. Vision*, March 2011, 92(1):1–31.

[24] Hur J, Roth S. *Optical Flow Estimation in the Deep Learning Age*, pp. 119–140. Springer International Publishing, Cham, 2020.

[25] Jeong S G, Lee C, Kim C S. Motion-compensated frame interpolation based on multihypothesis motion estimation and texture optimization. *Trans. Img. Proc.*, November 2013, 22(11):4497–4509.

[26] Mahajan D, Huang F C, Matusik W, Ramamoorthi R, Belhumeur P. Moving gradients: A path-based method for plausible image interpolation. *ACM Trans. Graph.*, July 2009, 28(3):42:1–42:11.

[27] Stich T, Linz C, Wallraven C, Cunningham D, Magnor M. Perception-motivated interpolation of image sequences. *ACM Trans. Appl. Percept.*, February 2011, 8(2):11:1–11:25.

[28] Meyer S, Wang O, Zimmer H, Grosse M, Sorkine-Hornung A. Phase-based frame interpolation for video. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 1410–1418.

[29] Long G, Kneip L, Alvarez J M, Li H, Zhang X, Yu Q. Learning image matching by simply watching video. In *European Conference on Computer Vision*, 2016, pp. 434–450.

[30] Niklaus S, Mai L, Liu F. Video frame interpolation via adaptive convolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2270–2279.

[31] Niklaus S, Mai L, Liu F. Video frame interpolation via adaptive separable convolution. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 261–270.

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

[33] Bao L, Yang Q, Jin H. Fast edge-preserving patchmatch for large displacement optical flow. *IEEE Transactions on Image Processing*, Dec 2014, 23(12):4996–5006.

[34] Barnes C, Shechtman E, Finkelstein A, Goldman D B. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, July 2009, 28(3):24:1–24:11.

[35] Penner E, Zhang L. Soft 3d reconstruction for view synthesis. *ACM Trans. Graph.*, November 2017, 36(6).

[36] Hedman P, Kopf J. Instant 3d photography. *ACM Trans. Graph.*, July 2018, 37(4).

[37] Chaurasia G, Duchene S, Sorkine-Hornung O, Drettakis G. Depth synthesis and local warps for plausible image-based navigation. *ACM Trans. Graph.*, July 2013, 32(3):30:1–30:12.

[38] Wang S, Wang R. Robust view synthesis in wide-baseline complex geometric environments. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2297–2301.

[39] Zhou T, Tulsiani S, Sun W, Malik J, Efros A A. View synthesis by appearance flow. In *European conference on computer vision*, 2016, pp. 286–301.

[40] Xu Z, Bi S, Sunkavalli K, Hadap S, Su H, Ramamoorthi R. Deep view synthesis from sparse photometric images. *ACM Transactions on Graphics (TOG)*, 2019, 38(4):1–13.

[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 *ECCV*, 2020.

[42] Yu A, Ye V, Tancik M, Kanazawa A. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021.

[43] Felzenszwalb P F, Huttenlocher D R. Efficient belief propagation for early vision. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, June 2004, pp. I–I.

[44] Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, November 2012, 34(11):2274–2282.

[45] Fischler M A, Bolles R C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981, 24:381–395.

[46] Hu Y, Li Y, Song R. Robust interpolation of correspondences for large displacement optical flow. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 4791–4799.

[47] Kaviani H R, Shirani S. Iterative mask generation method for handling occlusion in optical flow assisted view interpolation. In *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 3387–3391.

[48] Burt P J, Adelson E H. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, October 1983, 2(4):217–236.

[49] Butler D J, Wulff J, Stanley G B, Black M J. A naturalistic open source movie for optical flow evaluation. In A Fitzgibbon et al (Eds), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, October 2012, pp. 611–625.

[50] Menze M, Geiger A. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4040–4048.

[53] Huang P H, Matzen K, Kopf J, Ahuja N, Huang J B. Deepmvs: Learning multi-view stereopsis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 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 the B.S. degree in information engineering from Xi'an Jiaotong University, Xi'an, China, in 1996, and the Ph.D. degree in computer science from Nanjing University, Nanjing, China, in 2001. From 2001 to 2003, he was a Post-Doctoral Researcher with the Human-Computer Interaction and Multimedia Laboratory, Peking University, Beijing, China. From 2003 to 2005, he was a Research Fellow with the Image Computing Group, City University of Hong Kong, 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, 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 Graphics and Interactive Technology Laboratory, Peking University. He got Ph.D from Institute of Mathematics, Fudan University in 1997. He got full professor position in Dept. of Computer Science, Peking University in 2002. He achieved the National Science Fund for Distinguished Young Scholars in 2009. His research interests include virtual reality, computer graphics, human-computer interaction and multimedia.