Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Generalized Content-Preserving Warp: Direct Photometric Alignment beyond Color Consistency

KAI CHEN[1], (Student Member, IEEE), JINGMIN TU[1], JIAN YAO[1], (Member, IEEE) and JIE LI[2]
[1]School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, Hubei, P.R.China
[2]School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan, Hubei, P.R.China

Corresponding author: Jingmin Tu (e-mail: jingmin.tu@whu.edu.cn).

This work was partially supported by the National Natural Science Foundation of China (Project No. 41571436), the National Key Research and Development Program of China (Project No. 2017YFB1302400), the Hubei Province Science and Technology Support Program, China (Project No. 2015BAA027).

ABSTRACT Motion estimation is vital in many computer vision applications. Most existing methods require high quality and large quantity of feature correspondence, and may fail for images with few textures. In this paper, a photometric alignment method is proposed to obtain better motion estimation result. Since the adopted photometric constraints are usually limited to required illumination or color consistency assumption, a new Generalized Content-Preserving Warp (GCPW) framework therefore is designed to perform photometric alignment beyond color consistency. Similar to conventional Content-Preserving Warp (CPW), GCPW is also a mesh-based framework, but it extends CPW by appending a local color transformation model for every mesh quad, which expresses the color transformation from a source image to a target image within the quad. Motion-related mesh vertexes and color-related mapping parameters are optimized jointly in GCPW to get more robust motion estimation result. Evaluation of tens of videos reveals that the proposed method achieves more accurate motion estimation results. More importantly, it is robust to significant color variation. Besides, this paper explores the performance of GCPW in two popular computer vision applications: image stitching and video stabilization. Experimental results demonstrate GCPW's effectiveness in dealing with typical challenging scenes for these two applications.

INDEX TERMS Motion Estimation, Photometric Constraint, Color Difference, Image Stitching, Video Stabilization.

I. INTRODUCTION

MOTION estimation between two images, is carried out to find the corresponding pixel in a target image for each pixel in a source image. It is essential in many computer vision applications, such as image stitching and video stabilization. Generally, motion estimation methods can be classified into two categories: non-parametric [1]–[5] and parametric methods [6]–[10].

A typical and the most popular non-parametric motion estimation method is the optical flow [1], which directly estimates a $2D$ offset for each pixel that indicates the motion vector from a source image to a target image. Conventional optical flow estimation approaches usually assume that pixel intensities between two images keep constant during motion, base on which the target of optical flow estimation is to compute a motion field that minimizes intensity differences between two images. A lot of methods [2], [11] are proposed to achieve this target, but the brightness constancy assumption actually is a main drawback of these methods. Brox et al. [3] therefore resorted to high-order constancy (e.g., image gradient) to overcome this problem, but image gradient is sensitive to noise, and the relevant L_1-norm penalty is not easy to be optimized. Alternatively, in [4], [12], they decomposed images into cartoon and texture components and applied texture images that were less affected by illumination to estimate optical flow. Mileva et al. [13] achieved an illumination-robust estimation method by transforming color images into some photometric-invariant color space. Demetz et al. [14] directly learned a brightness transfer function from training data to handle the intensity variation. Above methods however, are limited and not robust enough to complex color variation. Recent deep learning

based methods consist of purely supervised [5], [15], semi-supervised [16], and unsupervised [17], [18] methods. These methods either fail to cope with images with color variation or require sufficient labeled data to train neural networks for some specific scenes, which restrict their application. More importantly, this non-parametric motion estimation method is time-consuming.

Under these circumstances, parametric motion model is a suitable alternative. There are many methods that use a parametric model to express motions between two images, in which motion estimation is formulated into efficient model parameters optimization. According to the applied model type, parametric methods can be classified as global parametric methods [6], [19], [20] and local parametric methods [7]–[10], [21].

In global parametric methods, motions of the entire image are parameterized by a single model. Image homography is the most widely used global parametric model that has an eight degrees of freedom (DoF) [22]. Brown and Lowe [6] proposed to estimate such a global homography by Direct Linear Transformation (DLT) using matched feature points in image overlapping region. Li et al. [9] estimated it with a similar manner but they utilized point correspondence as well as line segment correspondence to achieve a more accurate and more robust result. Most recently, some deep learning based methods [20], [23], [24] also are proposed to robustly compute homography parameters by convolutional neural networks (CNN). However, such a global parametric model is too coarse to represent motions for complex scenes. It works well only when image scenes are located on a single plane or images are captured by a camera that is under pure rotation.

Local parametric methods use spatially varying models to represent motions for different image areas. Compared with global methods, the higher DoF makes them more flexible to handle motions in complex scenes, but it makes the model estimation more difficult either. Therefore, two typical kinds of constraints are adopted to guide the model estimation process: geometric constraints [7]–[9] and photometric constraints [10].

Geometric constraints include point constraints and line segment constraints. There are many methods that estimate the local model from matched feature points in image overlapping region. Gao et al. [21] proposed a dual homography model to stitch images with two-plane scenes. More generally, Lin et al. [25] and Zaragoza et al. [8] developed a spatially-varying motion model to locally align image content. Besides, inspired by the technique of mesh-based image manipulation [26], Liu et al. [7] proposed content-preserving warp (CPW) for video stabilization, which approximates a motion field from an original video frame to a stabilized video frame. Liu et al. [27] introduced efficient MeshFlow. It estimates motions between adjacent two video frames from feature point correspondence that are obtained by Kanade-Lucas-Tomasi (KLT) tracker [28]. These methods have a common limitation: their performances are highly relied on the quality and quantity of extracted feature points, which is difficult to ensure when feature points are distributed with spatial bias or image scene is lack of texture. Recently, many researches [9], [29]–[31] therefore resorted to line segment constraints to breakthrough this limitation. Joo et al. [29] proposed a line guided moving DLT (L-mDLT) method, which estimated a spatially-varying homography model with line segment correspondences. Similarly, Li et al. [9] combined point and line segment constraints into the CPW framework. It achieves a state-of-the-art motion estimation performance on low-textured images.

Although the combination of point constraints and line segment constraints has significantly improved motion estimation quality, line segment constraints are not yet robust enough. For one thing, when images contain many small structures or tiny gradients, the extraction of line segments is difficult for most existing methods (e.g., LSD [32], ED-Line [33], and CannyLine [34]). For another, although line matching has obtained a lot of attention [35]–[38] in recent years, nevertheless, robust line matching algorithm is still a problem that has not yet been completely resolved. Any mismatched line segment would destroy final motion estimation result. Lin et al. [10] therefore developed mesh-based photometric alignment (MPA), which introduced photometric constraints into the CPW framework to estimate motions between two images better. This idea mainly stems from the problem of optical flow but is formulated as mesh deformation, which has a higher efficiency than conventional optical flow estimation. However, at the same time, MPA inherits the same drawback as optical flow estimation: the required color consistency assumption. Actually, this assumption is easily violated in practice in case of abrupt changes of illumination sources, existing shadows or influenced by noise in acquisition process [39].

In this paper, we present a novel model called generalized content-preserving warp (GCPW) for motion estimation between two images. GCPW is a mesh-based motion model. It considers the inherent color transformation between images by appending a local color mapping function for every mesh quad. There are many works [40], [41] that focus on an approximation of color transformation between two images. We propose to adopt the simple but effective affine mapping function that has been widely used in many tasks for the purpose of overcoming color variation, such as visual tracking [40] and image enhancement [41]. We then design a photometric constraint beyond color consistency, which is workable even for images with significant color variation. In order to maintain image content and reduce image distortion, the proposed photometric constraint is combined with three smoothness terms to form our final energy function. Finally, mesh vertexes and color affine parameters are optimized jointly to produce a better motion estimation result. GCPW has higher DoF than CPW to handle color difference, which makes it more difficult to optimize GCPW. We therefore design a three-step optimization method to achieve a robust estimation for such a high DoF model. Experiments on both real and synthetic data demonstrate that our proposed method

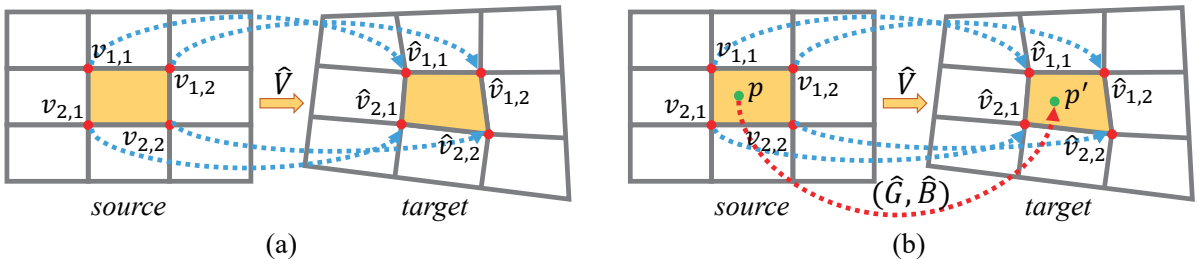


FIGURE 1. A comparison between GCPW and CPW [7]. (a) The conventional CPW framework only estimates warped mesh vertexes $\hat{V}$. (b) Our GCPW consists of two parts: motion-related mesh vertexes $\hat{V}$ and color-related mapping function $(\hat{G}, \hat{B})$. These two parts are optimized jointly in GCPW. $(\hat{G}, \hat{B})$ expresses the color transformation from a source image to a target image (*e.g.*, from p to p').

achieves more accurate motion estimation result than state-of-the-are methods, and it is also robust to significant color variations.

In the computer vision community, there are many problems that are relevant to motion estimation between two images. For the problem of image stitching, multiple images are stitched into a panorama by combining corresponding pixels from different images together, which actually is a process of motion estimation. As for application of video stabilization, whose main target is to produce a stable video from a shaky video. Camera path is usually extracted by motion estimation between every adjacent two video frames before it is stabilized by some advanced techniques. Motion estimation plays an important role in this problem. In this paper, we explore the performance of our motion estimation method in above two applications. Abundant experimental results demonstrate that the proposed GCPW is effective in these two popular application scenarios.

In summary, in this paper, we make the following three contributions:

- We introduce a new motion model GCPW, which is more flexible than the widely used CPW and can cope with images with color variation.
- We proposed a photometric constraint to perform photometric alignment beyond color consistency. A three-step optimization scheme is designed to achieve a robust estimation for GCPW.
- We apply the proposed GCPW to two popular computer vision applications. Experimental results demonstrate its effectiveness when it is used to handle challenging scenes in these two applications.

The rest of this paper is organized as follows. In Section II and Section III, the proposed motion estimation method is presented. Specifically, the general description of GCPW model is given in Section II-A, the adopted photometric constraint is described in Section II-B, and the three-step optimization pipeline is presented in detail in Section III. We evaluate our motion estimation result on various videos in Section IV, where Section IV-A compares our method with other state-of-the-art methods to demonstrate the estimation accuracy, and Section IV-B evaluates the robustness of GCPW to images with different degrees of color variation. Section V explores the performance of GCPW on two typical applications: image stitching and video stabilization. Finally, conclusion is drawn in Section VI.

II. GENERAL DESCRIPTION OF GCPW

In this section, we describe the main idea of the proposed GCPW model to estimate motions from a source image I_s to a target image I_t. We assume that I_s and I_t are roughly aligned (e.g., two consecutive video frames or two images that have been roughly aligned by a global homography). First, a general introduction of proposed GCPW framework is given. Next, we describe that how to measure photometric error of two images under the GCPW framework.

A. FRAMEWORK OF GCPW

The proposed GCPW is a mesh-based motion model. Fig. 1 gives a comparison between GCPW and the conventional CPW [7]. For one thing, they are similar. Because both of them divide an image into an $m \times n$ uniform grid mesh, and formulate the motion estimation into a problem of mesh deformation. For another, they are different. CPW only estimates warped mesh vertexes while GCPW optimizes mesh vertexes and color model parameters simultaneously.

As shown in Fig. 1, instead of using a global model to express color transformation from I_s to I_t, we express it locally. Specifically, we assume that pixels within a quad share the same affine color mapping relation and use a set of affine models to fit the color mapping relation between two images. Such a local color model is simple, but it is flexible enough to approximate a variety of complicated color variations. Let $V = \{v_{i,j}\}, i = 0, 1, ..., m; j = 0, 1, ..., n$ denote mesh vertexes. $G = \{g_{i,j}\}, i = 1, ..., m; j = 1, ..., n$ and $B = \{b_{i,j}\}, i = 1, ..., m; j = 1, ..., n$ represent gain and bias of the affine color model respectively. For a pixel p in I_s and its matched pixel p' in I_t, if p is located in the quad $[v_{i-1,j-1}, v_{i-1,j}, v_{i,j-1}, v_{i,j}]$, the mapping relation between p and p' is formulated as:

$$I_t(p') = I_s(p) \times g_{i,j} + b_{i,j}, \tag{1}$$

where $I_s(p)$ and $I_t(p')$ denote their pixel intensities. GCPW estimates V and $\{G, B\}$ jointly to obtain the optimal mesh vertexes $\hat{V}$ and the optimal local color model $\{\hat{G}, \hat{B}\}$. As a motion model, the main purpose of GCPW is to estimate $\hat{V}$ accurately, but $\{\hat{G}, \hat{B}\}$ play an important role in this process,

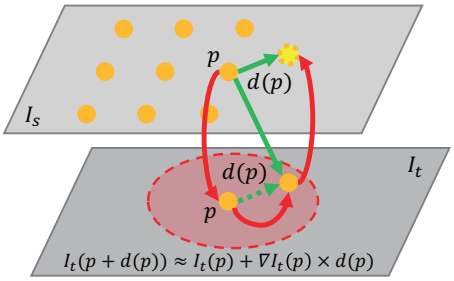


FIGURE 2. An illustration of photometric error measurement between I_s and I_t. Uniform points are sampled in the overlapping region. $I_t(p + d(p))$ is approximated by Taylor expansion within a small local region. The object of photometric alignment is to find optimal $d(p)$ that minimize the difference between $I_t(p + d(p))$ and mapped $I_s(p)$.

especially when we resort to photometric constraint to cope with low-texture images.

B. PHOTOMETRIC ERROR METRIC

Photometric error metric measures pixel intensity difference between two images. It is a widely used metric in optical flow estimation, and Lin et al. [10] first introduce it into CPW to estimate motions by mesh deformation. Their required color consistency condition however, is not always met in practice. Therefore, we modify it into a photometric error metric beyond color consistency based on the proposed GCPW framework.

As shown in Fig. 2, we first uniformly sample a set of points at a fixed interval both horizontally and vertically in image overlapping region. For each sampled point p in I_s, we want to find a $2D$ offset $d(p)$ that minimizes the photometric difference between p in I_s and $p+d(p)$ in I_t. Considering the color variation from I_s to I_t, we measure their photometric error as:

$$\|G(p) \times I_s(p) + B(p) - I_t(p + d(p))\|^2, \qquad (2)$$

where $I_s(p)$ is the intensity of I_s at p, and $I_t(p+d(p))$ is the intensity of I_t at $p + d(p)$. $G(p)$ and $B(p)$ denote the gain and bias of the quad that p is located in. $\|\cdot\|^2$ denotes the Euclidean distance. In our practice, for each sampled point in a typical RGB image, we measure its photometric error on three image channels separately. Since we assume that I_s and I_t are roughly aligned, which is similar to [10], we think that $d(p)$ should be small and set its initial value as $(0, 0)$. Then by using the first-order Taylor expansion of $I_t(p + d(p))$, the photometric error defined by Eq. 2 can be simplified as:

$$e_p = \|G(p)\times I_s(p)+B(p)-(I_t(p)+\nabla I_t(p)\cdot d(p))\|^2, \quad (3)$$

where $\nabla I_t(p)$ is image gradient of I_t at p. The error metric defined by Eq. 3 typically has two advantages: (1) it can measure photometric error normally even when I_s and I_t have apparent color difference; (2) it retains a quadratic property and is easy to be minimized. In next section, we will combine it with other constraints to estimate parameters of GCPW.

III. GCPW OPTIMIZATION

Compared with CPW, due to higher degree of freedom (DoF), GCPW is more flexible to estimate motions based on photometric constraint for images with noticeable color difference. In this section, we propose a three-step optimization pipeline to achieve a robust estimation for such high DoF model. Fig. 3 presents the general optimization process. First, the color model is initialized according to initial sampled point sets. Next, we jointly estimate mesh vertexes and affine color models by minimizing overall photometric errors of sampled points. Finally, we revise the model by eliminating effects from moving objects and occlusion regions. Algorithm 1 outlines the main optimization process. More details are described as follows.

Algorithm 1 Joint Optimization

Input:
Image gaussian pyramid $\{I_s^{(k)}\}$, $\{I_t^{(k)}\}$, $k = 1, 2, 3$
Initialized grid mesh $\mathcal{M}_{m\times n}\{V, G_0, B_0\}$
Output:
Optimal grid mesh $\hat{\mathcal{M}}_{m\times n}\{\hat{V}, \hat{G}, \hat{B}\}$

$\hat{\mathcal{M}} \leftarrow \mathcal{M}$
for k=1,2,3 **do**
  $I_s \leftarrow I_s^{(k)}$, $I_t \leftarrow I_t^{(k)}$
  For I_s, generate uniform grid mesh $\mathcal{M} \leftarrow \hat{\mathcal{M}}$
  Sample points in image overlapping region S_p^k and mark them as *inlier*
  $C_{outlier} \leftarrow$ INT_MAX
  while $C_{outlier} > \tau_c$ **do**
    Estimate current model parameters $\widetilde{\mathcal{M}}\{\widetilde{V}, \widetilde{G}, \widetilde{B}\}$ with inlier sampled points(Eq. 20)
    Set $C_{outlier} = 0$
    for all $p \in S_P^k$ **do**
      if p is *inlier* **then**
        compute residual photometric difference of p to get e_p(Eq. 21)
        if $e_p > \tau$ **then**
          mark p as an *outlier*, $C_{outlier} \leftarrow C_{outlier}+1$
        end if
      end if
    end for
  end while
  Propagate estimation result from current layer to next layer $\hat{\mathcal{M}} \leftarrow \widetilde{\mathcal{M}}$
end for

A. INITIALIZATION STEP

As shown in Fig. 3(a), for each sampled point p in I_s, q is the corresponding point in I_t, which has the same coordinates with p. We take three factors into consideration to initialize the color model of GCPW. First, the initial overall photometric difference between p and q should be small, which is achieved by minimizing the following equation:

$$E_p^{init}(G, B) = \sum_p \|G(p) \times I_s(p) + B(p) - I_t(q)\|^2. \quad (4)$$

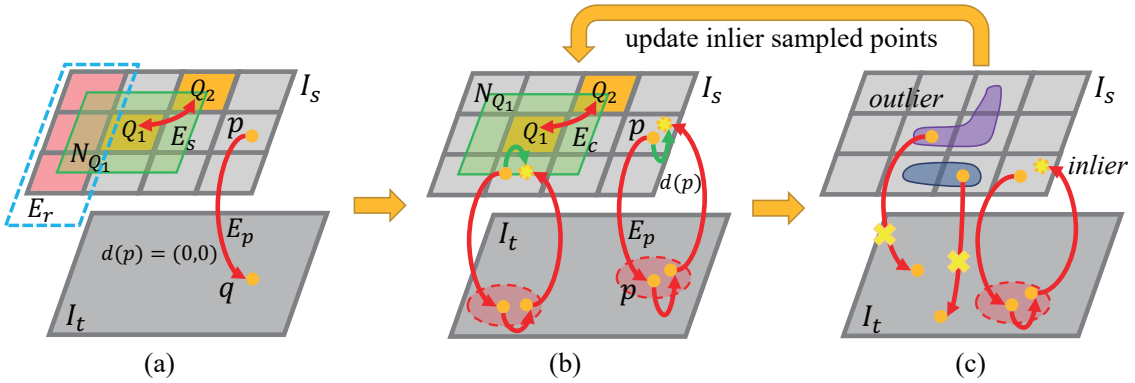


FIGURE 3. The three-step optimization pipeline of the proposed method. (a) Initialization step. $d(p)$ is fixed at $(0, 0)$. The blue rectangular marks out the non-overlapping region of I_s and I_t, in which the color model is required to be close to the identity affine transformation. The green marks out the eight-connected neighborhood of quad Q_1. (b) Joint optimization step. The optimal value of $d(p)$ is searched within a small local region (denoted by red circle in I_t). (c) Model revision step. Sample points located in moving objects (blue area) and occlusion region (purple area) are marked as outliers and are abandoned.

It is equivalent to fixing the $2D$ offset $d(p)$ of Eq. 2 at $(0, 0)$ for all sampled points. Secondly, Since for quads that are located in the image non-overlapping region, there is not any sampled point. So we directly require color models of these quads to be close to the identity affine transformation:

$$E_r^{init}(G, B) = \sum_{Q \cap \Omega = \varnothing} (\|G(Q) - 1.0\|^2 + \|B(Q) - 0.0\|^2), \tag{5}$$

where $\Omega = I_s \cap I_t$ denotes the image overlapping region. $G(Q)$ and $B(Q)$ represent the color gain and bias of quad Q. Thirdly, color models of neighboring quads should be similar. We therefore design a smoothness term to limit color model smoothness within an eight-connected neighboring region. We select a set of intensity values in normalized intensity range at a fixed interval, which can be denoted as $X = \{x_1, x_2, ..., x_k\}$, and we require these sampled intensities to remain close after two neighboring color mappings:

$$E_s^{init}(G, B) = \sum_{x \in X} \sum_{Q_2 \in N_{Q_1}} \sum_{Q_1} \|\mathcal{F}_{Q_1}(x) - \mathcal{F}_{Q_2}(x)\|^2, \tag{6}$$

where N_{Q_1} denotes eight-connected neighboring quads of Q_1, and $\mathcal{F}_{Q_1}(\cdot)$ and $\mathcal{F}_{Q_2}(\cdot)$ represent neighboring two affine color mapping functions. Considering above three aspects, the color model is initialized as:

$$\{G_0, B_0\} = \arg\min_{G,B}(E_p^{init} + E_r^{init} + E_s^{init}), \tag{7}$$

where G_0 and B_0 are initialized color gain and bias respectively, and we convey them to next step to estimate $\hat{V}$ and $\{\hat{G}, \hat{B}\}$ jointly.

B. JOINT ESTIMATION STEP

Original CPW [7] uses matched feature points in image overlapping region to guide motion estimation process, and its objective function can be expressed as:

$$E_{cpw} = E_f + \alpha E_s, \tag{8}$$

where E_f is the feature point term, and E_s is the similarity transformation term. We add three terms on the basis of Eq. 8

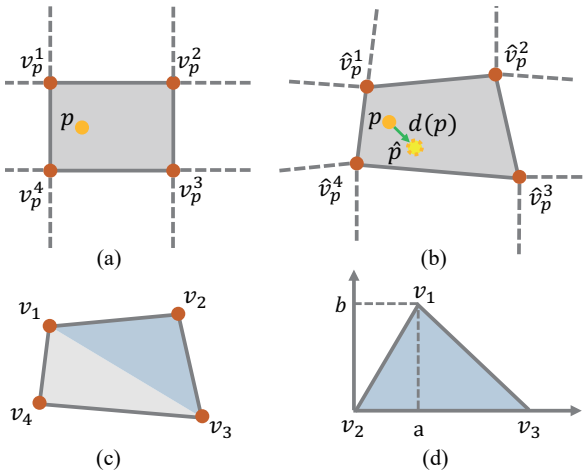


FIGURE 4. An illustration of a quad in the grid mesh. In (a) and (b), p and $\hat{p}$ are expressed by related vertexes using the same interpolation coefficients. In (c) and (d), the quad is first divided into two triangulations. After that, one vertex is expressed in the local coordinate system defined by other two vertexes.

to obtain our objective function for three purposes: (1) we add a photometric term E_p to cope with low-texture images; (2) a color similarity term E_c is added to ensure the smoothness of the color model in GCPW; (3) a line collinearity term E_l is designed to preserve image content better. In general, our objective function can be expressed as:

$$E^{joint} = \lambda_1 E_f + \lambda_2 E_s + \lambda_3 E_p + \lambda_4 E_c + \lambda_5 E_l, \tag{9}$$

where λ_1, λ_2, λ_3, λ_4, and λ_5 are associated balancing weights. Next, we first introduce a parameterization method that connects our adopted constraints with mesh deformation. After that, a detailed description about above five terms is given.

As shown in Fig 4, p is a point in I_s, and it is enclosed by a quad whose four vertexes are denoted as v_p^1, v_p^2, v_p^3, and v_p^4. The described parameterization method consists of two principles: First, p is parameterized by expressing it as a

weighted sum of these four vertexes:

$$p = \sum_{k=1}^{4} w_k \times v_p^k, \tag{10}$$

where w_k denote four bilinear interpolation weights. Secondly, the interpolation weights are unchanged after warping the quad. Let $\hat{p}$, $\hat{v}_p^1$, $\hat{v}_p^2$, $\hat{v}_p^3$ and $\hat{v}_p^4$ denote the warped positions of p, v_p^1, v_p^2, v_p^3, and v_p^4. The second principle assumes that $\hat{p}$ can be expressed by $\hat{v}_p^1$, $\hat{v}_p^2$, $\hat{v}_p^3$ and $\hat{v}_p^4$ using the same interpolation weights that we computed in the original quad:

$$\hat{p} = \sum_{k=1}^{4} w_k \times \hat{v}_p^k. \tag{11}$$

As demonstrated in [42], this assumption is reasonable especially when size of a quad is small.

- **Feature point term** E_f. This term provides mesh warping guidance with matched feature points. Let p be a feature point in I_s, and let p' be its matching point in I_t. We restrict the warped position of p to being close to p', which is achieved by minimizing their Euclidean distance of $2D$ image coordinates. The feature point term is formulated as follows:

$$E_f = \sum_{(p,p')\in M_f} \| \sum_{k=1}^{4} w_k \times v_p^k - p' \|^2, \tag{12}$$

  where M_f represents matched feature point set. Each feature point in I_s is further parameterized by above parameterization method.
- **Similarity transformation term** E_s. This term measures the deviation of each warped mesh quad from a similarity transformation of its shape before warping. Specifically, as shown in Fig. 4, each mesh quad $[v_1, v_2, v_3, v_4]$ is first divided into two triangulations $\Delta v_1 v_2 v_3$ and $\Delta v_1 v_3 v_4$. Taking $\Delta v_1 v_2 v_3$ as an example, we express v_1 in a local coordinate system defined by v_2 and v_3 as follows:

$$v_1 = v_2 + a(v_3 - v_2) + b\mathbf{R}_{90}(v_3 - v_2), \mathbf{R}_{90} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{13}$$

  where a and b are computed from original positions of v_1, v_2 and v_3. We restrict that v_1 can be represented by v_2 and v_3 using the same local coordinates (a, b) before and after mesh warping. The overall similarity transformation term is defined as:

$$E_s = \sum_{i=1}^{C_t} \| v_1^i - (v_2^i + a(v_3^i - v_2^i) + b\mathbf{R}_{90}(v_3^i - v_2^i)) \|^2, \tag{14}$$

  where C_t is the total count of triangulations in the grid mesh. v_1^i, v_2^i and v_3^i are three vertexes of the i-th triangulation.
- **Photometric term** E_p. For each sampled point p, we measure its corresponding photometric error according to Eq. 3, which is supposed to be minimized in our joint optimization step. In order to associate it with mesh deformation, we parameterize the $2D$ offset $d(p)$ by:

$$d(p) = \hat{p} - p, \quad \hat{p} = \sum_{k=1}^{4} w_k \times v_p^k, \tag{15}$$

  where $\hat{p}$ is the warped position of p, and w_k are bilinear interpolation coefficients computed by our described parameterization method. E_p is computed by summing up e_p over all sampled points:

$$E_p = \sum_{p\in M_s} e_p, \tag{16}$$

  where M_s denotes the sampled point set.
- **Color similarity term** E_c. This term is designed to constrain the smoothness of color models within an eight-connected neighboring region, which is also taken into account in our initialization step, and we just define it in the same way. Specifically, we sample a set of discrete intensity values $X = \{x_1, x_2, ..., x_k\}$ in normalized intensity range with a fixed interval, and let these selected intensities remain close after two neighboring color mappings:

$$E_c = \sum_{x\in X} \sum_{Q_2\in N_{Q_1}} \sum_{Q_1} \| \mathcal{F}_{Q_1}(x) - \mathcal{F}_{Q_2}(x) \|^2. \tag{17}$$

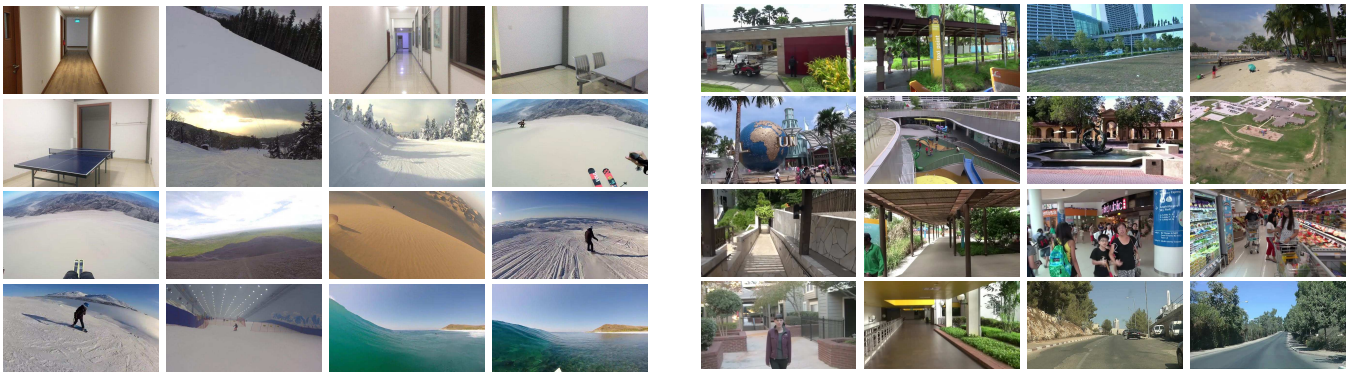
  N_{Q_1} denotes eight-connected neighboring quads of Q_1, and $\mathcal{F}_{Q_1}(\cdot)$ and $\mathcal{F}_{Q_2}(\cdot)$ represent neighboring two affine color mapping functions.
- **Line collinearity term** E_l. The similarity transformation term E_s preserves image content within a mesh quad by restrict the deviation of each quad from a similarity transformation, but it is usually not sufficient to reduce the distortions for structures larger than the mesh quad. Therefore, similar to [31], [43], we add a line collinearity term to preserve image content better, which is achieved by maintaining the straightness of linear structures in I_s as much as possible. Specifically, we detect line segments in I_s using LSD detector [32]. For each line segment l, whose two endpoints are denoted as l_s and l_e, we sampled key points along it at a fixed interval. For each sampled key point l, we compute its coordinate c in the local coordinate system defined by l_s and l_e:

$$l = l_s + c \times (l_e - l_s). \tag{18}$$

  We require each key point can be expressed by the same local coordinate before and after mesh warping. So the line collinearity term is defined as:

$$E_l = \sum_{i=1}^{C_l} \sum_{j=1}^{C_p^i} \| l^{i,j} - (l_s^i + c \times (l_e^i - l_s^i)) \|^2, \tag{19}$$

  where C_l is the count of detected line segments and C_p^i is the number of sampled key points of the i-th line segment. $l^{i,j}$, l_s^i and l_e^i are parameterized by mesh vertexes using the same parameterization method that described above.



Low-texture videos: 01-16 Ordinary videos: 17-32

FIGURE 5. Two groups of videos used in our method evaluation. They include ordinary as well as low-texture scenes, cover different types of motion modes (e.g., zooming, rotation, rolling shutter, running and driving), and contain thousands pairs of video frames.

The final joint optimization result is obtained by solving:

$$\{\widetilde{V}, \widetilde{G}, \widetilde{B}\} = \arg\min_{V,G,B} E^{joint} \tag{20}$$

C. REVISION STEP

In previous steps, we uniformly select points in image overlapping region and refer them as photometric constraints to estimate motions from I_s to I_t. It might result in a biased estimation result if we blindly require two points from different objects to have similar intensity values, which occurs when sampled points locate on moving objects or image occlusion regions. Therefore, we further revise above joint optimization result by eliminating these effects based on the following observation: $\widetilde{V}$ is affected slightly because applied strong regularization terms and points located on those bad regions usually have larger photometric errors after our joint optimization.

In our practice, for each sampled point p, we compute its warped position p' based on $\widetilde{V}$. Then its corresponding photometric error value is computed as:

$$e_p = \|\widetilde{G}(p) \times I_s(p) + \widetilde{B}(p) - I_t(p')\|^2, \tag{21}$$

where $\widetilde{G}(p)$ and $\widetilde{B}(p)$ are optimized color gain and bias obtained from joint optimization step. Sampled point whose e_p is larger than a predefined threshold τ is marked as an outlier, otherwise, it is marked as an inlier. We discard outliers and use inlier sampled points to perform joint optimization step again. After that, photometric errors of current inlier points are re-computed based on newly optimized model parameters, and points with e_p larger than τ are marked as outliers and are abandoned. This process performs iteratively until only a few new outliers can be picked out.

In order to handle large displacement between I_s and I_t, a coarse-to-fine scheme that is widely used in many motion estimation problems is also adopted in this paper. We build a three layer Gaussian pyramid for both I_s and I_t. The initialization step (III-A) is only performed on the coarsest layer. Then for each pyramid layer, we use the same mesh resolution and perform joint optimization (III-B) and model revision (III-C) iteratively to obtain optimal estimation result. The result from current layer serves as initial values to propagate to next pyramid layer. The result of the finest layer is our final model estimation result $\{\hat{V}, \hat{G}, \hat{B}\}$. We can use $\hat{V}$ to warp I_s to I_t by mesh warp technology.

IV. QUANTITATIVE EVALUATION

In this section, we evaluate our proposed GCPW quantitatively on various videos with 640×360 resolution. On the one hand, we compared our method with three geometric-based methods to demonstrate that the proposed GCPW produces more accurate motion estimation results. On the other hand, we generated experimental inputs with different degrees of color variation and performed experiments on them. A comparison with the state-of-the-art methods shows that our model can handle images with color difference and is more robust to different degrees of color variation. During the quantitative evaluation, we fix the balancing weights in Eq. 9 as: $\lambda_1 = 1.0$, $\lambda_2 = 0.5$, $\lambda_3 = 100.0$, $\lambda_4 = 1.0$, and $\lambda_5 = 1.0$. The photometric threshold τ is set to 0.05, and the mesh resolution is fixed at: $m = 16, n = 16$.

A. EVALUATION OF ESTIMATION ACCURACY

1) Experimental Data

As shown in Fig. 5, 32 videos are carefully prepared by collecting from publicly available datasets [10], [44]–[46] and YouTube [1], or by capturing by ourselves. Typically, the collected 32 videos can be categorized into two classes: low-texture videos and ordinary videos. Both of them consist of videos with different typical motion modes, such as: rotation, zooming, rolling shutter, running, etc.

2) Evaluation Metric

We describe the metric that is used to evaluate experimental results quantitatively. For a video with k frames, we take each pair of adjacent video frames (f_t, f_{t+1}) as input. f_{t+1} is warped to obtain $\hat{f}_{t+1}$, which is aligned to f_t by using the estimated motion from f_{t+1} to f_t. The accuracy of the estimated motion is measured based on the alignment error between f_t and $\hat{f}_{t+1}$, which is computed by calculating the

[1] https://www.youtube.com/

TABLE 1. Average RMSE results on 32 videos. *Global*: global homography [6]. *CPW*: content-preserving warp [7]. *DFW*: dual-feature warp [9].

Video No.	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Global	0.616	5.574	0.775	0.281	0.836	3.063	1.806	9.155	4.268	1.690	1.593	11.096	10.808	1.598	1.958	2.959
CPW	0.625	4.038	0.652	0.361	1.090	1.111	0.544	6.232	2.966	1.372	1.320	7.468	7.344	0.804	1.605	2.407
DFW	0.657	4.128	0.717	0.402	1.173	1.141	0.575	6.201	2.927	1.357	1.270	7.327	7.270	0.782	1.594	2.391
GCPW	**0.394**	**3.552**	**0.515**	**0.205**	**0.658**	**0.751**	**0.474**	**5.506**	**2.456**	**1.265**	**0.953**	**4.201**	**4.054**	**0.426**	**1.467**	**2.070**
Video No.	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Global	3.716	8.6702	2.978	2.291	5.221	1.586	3.758	2.322	3.420	5.490	3.686	3.318	2.310	1.587	9.443	18.759
CPW	2.993	6.262	2.808	**1.547**	**3.384**	1.054	2.427	0.344	2.011	4.081	2.510	2.259	2.026	1.118	5.307	10.705
DFW	3.059	6.246	2.826	1.800	3.490	1.646	3.154	0.618	2.076	4.288	2.624	2.508	1.812	1.217	5.473	10.837
GCPW	**2.467**	**5.164**	**2.466**	1.594	3.479	**0.857**	**2.250**	**0.232**	**1.642**	**3.588**	**2.277**	**1.957**	**1.497**	**1.038**	**3.941**	**8.773**

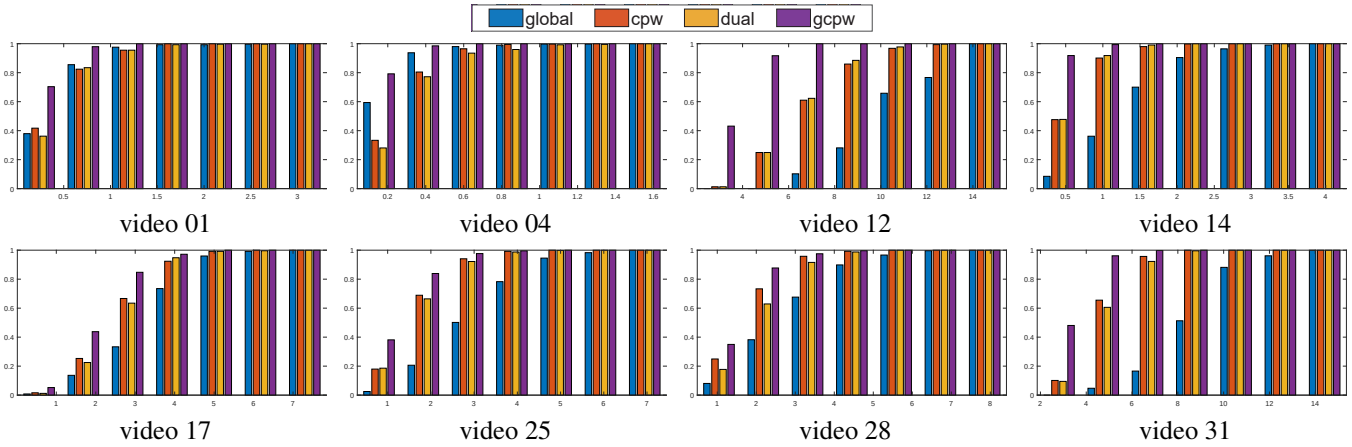


FIGURE 6. Cumulative distribution histograms of eight selected videos. Compared with other three methods, the results produced by GCPW has larger proportion of frames that have low alignment errors.

RMSE of one minus normalized cross correlation (NCC) over a local $w \times w$ window for pixels within the image overlapping region:

$$RMSE(f_t, f_{t+1}) = \sqrt{\frac{1}{N}\sum_{p\in\Omega}(1.0 - NCC(f_t(p), \hat{f}_{t+1}(p)))}, \tag{22}$$

where Ω denotes overlapping region of f_t and $\hat{f}_{t+1}$, and N is total pixel number within Ω. We take the average RMSE value of all frame pairs as the final measurement of current video:

$$Eval = \frac{1}{k-1}\sum_{t=1}^{k-1}(RMSE(f_t, f_{t+1})). \tag{23}$$

3) Comparative Results

We compared our proposed GCPW with three geometric-based motion estimation methods: global homography [6], CPW [7], and DFW [9]. The comparative results are presented in Table 1, from which we can see that the global homography usually produced the largest alignment error because such a global motion model is not flexible enough to express motions between adjacent video frames. The CPW performed better than the global homography (produced smaller alignment error) as it is a local motion model and it uses matched feature points to compensate displacements between two frames. The DFW resorts to line segments to estimate motions and it got better performance than the CPW in some videos (e.g., video 08 and video 29). Nevertheless, in our experimental results, DFW's improvement is inconspicuous and unstable (e.g., in video 20 and video 22). This is because the performance of DFW is highly relied on good results of line segment detection as well as line segment matching, which can be easily destroyed by any wrong line segment correspondence. In contrast, our proposed GCPW produced the smallest alignment error values on most videos. Fig. 6 further presents eight cumulative distribution histograms, which indicate the accumulated distributions of alignment errors of eight selected video sequences. We can observe that in GCPW, more video frames tend to have lower alignment errors compared with other three methods, which also can demonstrate that motions estimated by our GCPW have higher accuracy.

B. EVALUATION OF ROBUSTNESS TO COLOR VARIATION

1) Data Preparation

In this experiment, we still utilized our collected 32 videos as experimental data. For each pair of video frames (f_t, f_{t+1}), we processed f_{t+1} by some color transformation to generate 16 pairs of videos frames, which can be denoted as $\{(f_t, f_{t+1}^1), (f_t, f_{t+1}^2), ..., (f_t, f_{t+1}^{16})\}$. These 16 frame pairs are controlled to explicitly have different degrees of color difference that pair (f_t, f_{t+1}^1) has the smallest color difference

TABLE 2. 16 groups of parameter combinations used to synthesize frame pairs with different degrees of color difference.

a	1	2	3	4	5	6	7	8
$\beta(Y, Cb, Cr)$	(1.00,1.00,1.00)	(0.95,1.00,1.00)	(0.90,0.99,1.00)	(0.85,0.99,0.99)	(0.85,0.97,0.99)	(0.80,0.97,0.98)	(0.80,0.96,0.97)	(0.75,0.96,0.97)
a	9	10	11	12	13	14	15	16
$\beta(Y, Cb, Cr)$	(0.70,0.96,0.97)	(0.70,0.96,0.96)	(0.65,0.96,0.97)	(0.65,0.95,0.96)	(0.60,0.95,0.96)	(0.55,0.95,0.96)	(0.55,0.95,0.95)	(0.50,0.95,0.95)

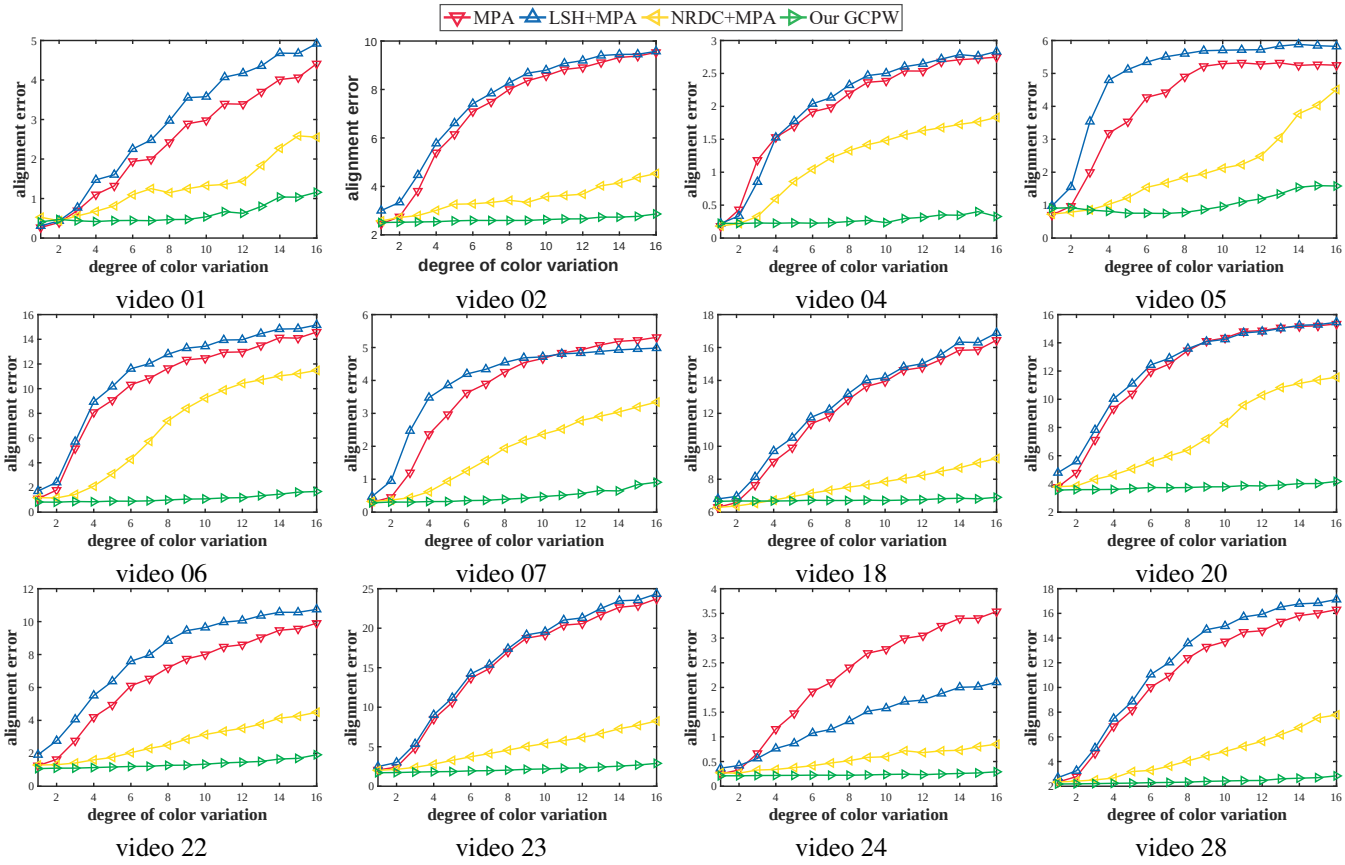


FIGURE 7. Comparison with MPA [10], LSH+MPA [40], and NRDC+MPA [41] on synthetic frame pairs. 12 typical videos are selected. When the color difference is small, the performance of all four methods are close. With the color difference gets larger, the average alignment error of other three methods increase significantly. In contrast, the proposed GCPW stably produces the best alignment quality.

and pair (f_t, f_{t+1}^{16}) has the largest color difference.

In order to avoid the effect of channel correlation, we processed f_{t+1} in the $YCbCr$ color space, and the color model that we adopted is similar to the one of [47], which can be presented as:

$$I' = I^{\gamma}, \tag{24}$$

where I is the original frame and I' is the processed frame. $(\cdot)^{\gamma}$ is the non-linear gamma mapping function. Moreover, in order to simulate the local color variation, for each frame pixel, we computed its γ value based on its $2D$ image coordinates:

$$\gamma = \beta \times (1 + \frac{a-1}{N})^{d/s}, \tag{25}$$

where $N = 16$ denotes the total number of frame pairs that we generate from each original frame pair. $a = 1, 2, ..., 16$ is current pair index. d denotes x (or y) coordinate of frame pixel and s is the width (or height) of a video frame. We controlled the degree of color variation from $a = 1$ to $a = 16$ by assigning different β for different a, and the parameter combination that we used in our experiment is listed in Table 2.

2) Evaluation Manner

Assume that $(f_t, f_{t+1}^{a_1})$ and $(f_t, f_{t+1}^{a_2})$ are two frame pairs generated from (f_t, f_{t+1}). They have different color variations and in order to compare their motion estimation results fairly, we evaluated their results as follows: First, we obtained their estimated motions $\mathcal{M}^{a_1}$ and $\mathcal{M}^{a_2}$ that aligned $f_{t+1}^{a_1}$ and $f_{t+1}^{a_2}$ to f_t respectively; Secondly, f_{t+1} was warped by $\mathcal{M}^{a_1}$ to get $\hat{f}_{t+1}^{a_1}$ and was warped by $\mathcal{M}^{a_2}$ to get $\hat{f}_{t+1}^{a_2}$ respectively; Finally, we computed the alignment error between f_t and $\hat{f}_{t+1}^{a_1}$ as well as between f_t and $\hat{f}_{t+1}^{a_2}$ based on Eq. 22 and referred to the error values as their evaluation metrics.

For a video with k frames, for each degree of color variation (e.g., $a = 1$), we averaged alignment errors of $k-1$ frame pairs to get the final evaluation metric of this video under current degree of color variation.

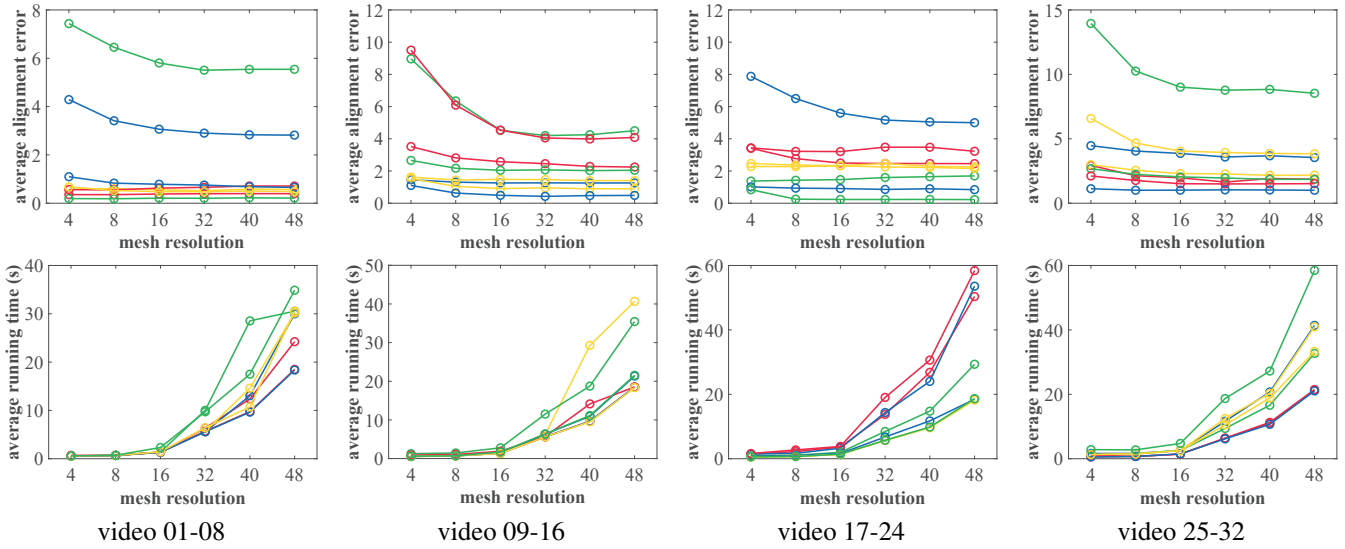


FIGURE 8. Effect of different mesh resolutions on motion estimation accuracy and average running time. 32 videos are separated into 4 groups. The effect on motion estimation accuracy is presented in the first row, and the effect on average running time is reported in the second row.

3) Comparative Results

We compared the proposed GCPW with MPA [10], LSH+MPA and NRDC+MPA to demonstrate the robustness of our method. The MPA that is proposed in [10] utilizes photometric constraints in the traditional CPW framework to estimate motions between two adjacent video frames, but it requires images to obey color consistency assumption. We therefore further tested the MPA cooperated with two pre-processing operations. For the first operation, we combined the MPA with the locality sensitive histogram (LSH) [40], which offers invariant image features, based on which MPA was adopted to perform motion estimation. We referred to this method as LSH+MPA. For the second operation, we combined the MPA with NRDC [41], which performs image color consistency correction to compensate color difference between two video frames. We referred to this method as NRDC+MPA. Fig. 7 typically presents 12 groups of comparative results. We can observe that when color difference is modest, all four methods produced similar estimation results. However, errors of the MPA, LSH+MPA and NRDC+MPA became larger as the color difference increased. The growth of the error curve of NRDC+MPA is usually slower than the one of MPA and LSH+MPA, but it is still higher a lot than the one of GCPW when the color variation becomes huger and more complicated (e.g., a=16). In contrast, as the degree of color variation varied from the smallest to the largest, errors of proposed GCPW were stable and were usually the smallest, demonstrating that GCPW is robust to different degrees of color variation.

C. EFFECT OF THE MESH RESOLUTION

The proposed GCPW actually approximates the $2D$ offset of each sampled point by mesh deformation. Parameters m and n, that are related to mesh resolution, indeed play an important part in GCPW method. On the one hand, higher mesh resolution can achieve more accurate motion approximation, which leads to lower alignment error. On the other hand, higher mesh resolution means larger amount of parameters that require longer time to be solved. Therefore, we tested GCPW with different mesh resolutions. Specifically, mesh resolution was controlled to vary from 4×4 to 48×48. The average alignment error and average running time of all 32 videos are reported in Fig. 8. We can observe that the use of higher m and n values usually can produce lower alignment errors, which means higher motion estimation accuracy, but it always costs longer time. Besides, when the mesh resolution reaches to 16×16, continuing to increase the mesh resolution can bring only a little improvement in accuracy, but it enlarges the average time cost significantly. We find that a mesh resolution of 16×16 achieves a good trade off between accuracy and efficiency.

V. EVALUATION ON APPLICATIONS

In this section, we validate GCPW in two typical computer vision applications: image stitching and video stabilization.

A. IMAGE STITCHING

The objective of image stitching is to combine multiple images into a panorama, which has a wider field of view. Typically, there are two challenging scenes in the task of image stitching: low-texture image stitching and large-parallax image stitching. We demonstrate that our proposed GCPW can be applied to cope with these two challenging scenes effectively.

Many methods use matched feature points [8], [25] or line segments [9], [29], [31] to perform image stitching. But these methods are highly relied on the quantity and quality of feature extraction and feature matching, which are difficult to guarantee for images with low texture. In contrast, our GCPW resorts to dense photometric constraints, which stably offers sufficient guidance to the stitching process. More importantly, the utilized photometric constraints do

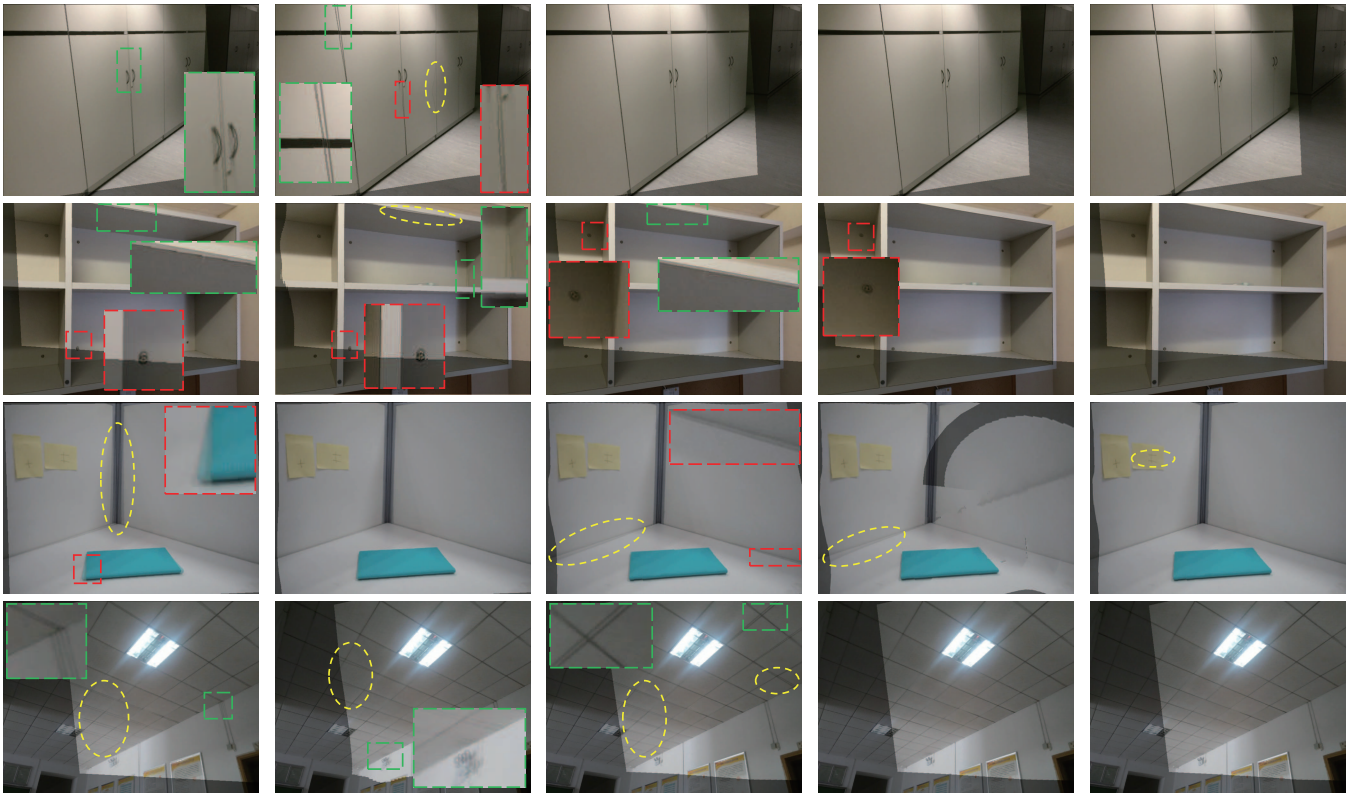


FIGURE 9. An illustration of comparative results for low-texture image stitching. (a) Global homography [6]. (b) APAP [8]. (c) CPW [7]. (d) DFW [9]. (e) Proposed GCPW.

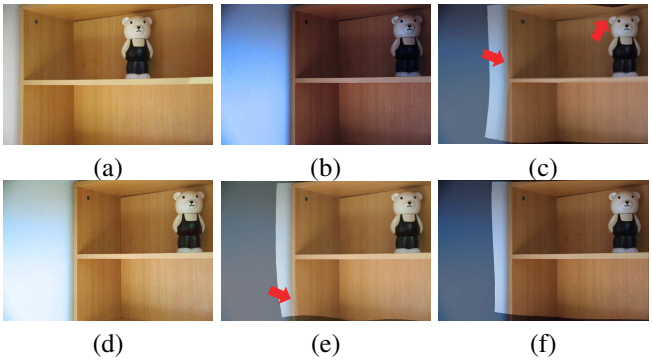


FIGURE 10. An intuitive example for stitching low-texture images with significant color differences. (a) target image. (b) source image. (c) result produced by MPA [10]. (d) color consistency correction result of (b). (e) result produced by MPA [10] based on corrected image pair. (f) result from GCPW.

not require images to obey color consistency assumption. In fact, this assumption can be violated easily in image stitching practice in case of changes of illumination sources, diversities of capturing devices or some other impact factors. Therefore, the proposed GCPW is more robust than MPA [10]. In order to evaluate the effectiveness of GCPW, we compared our method with four state-of-the-art methods: CPW [7], APAP [8], DFW [9], and MPA [10]. Fig. 9 gives an intuitive comparison with CPW, APAP, and DFW. The proposed GCPW usually achieves the best alignment quality. Furthermore, Fig. 10 shows an example that low-texture images to be stitched have apparent color differences. MPA suffers from these color variations and fail while the GCPW stably produce satisfactory result.

For images with large parallax, seamline-driven methods [43], [48], [49] usually perform better than methods that work in an alignment manner. Recent SEAGULL [43] combines the CPW-based image local alignment with the optimal seamline detection, and these two steps are conducted iteratively to obtain the state-of-the-art performance on parallax-tolerant image stitching. In order to prove the effectiveness of GCPW, we combined the proposed GCPW alignment method with the semaline-driven iterative procedure to stitch images with large parallax. The difference between our method with SEAGULL is that we perform image local alignment in the proposed GCPW framework, while SEAGULL locally align images in conventional CPW framework. We resort robust photometric constraints while SEAGULL highly relies on extracted feature points. Fig. 11 and Fig. 12 present our comparative results with Zhang and Liu's method [48] and SEAGULL [43], from which we can see that our method produces comparable stitching results with these two state-of-the-art methods on ordinary large-parallax scenes. Moreover, Fig. 13 shows an example that images to be stitched are lack of texture and have large parallax at the same time. As shown in Fig. 13, under this circumstance, GCPW cooperated with seamline-driven strategy produces better result than SEAGULL, which shows the superiority of GCPW.

FIGURE 11. Comparison with Zhang and Liu's method [48] for parallax-tolerant image stitching. Top: results from the proposed GCPW. Bottom: results from Zhang and Liu's method.

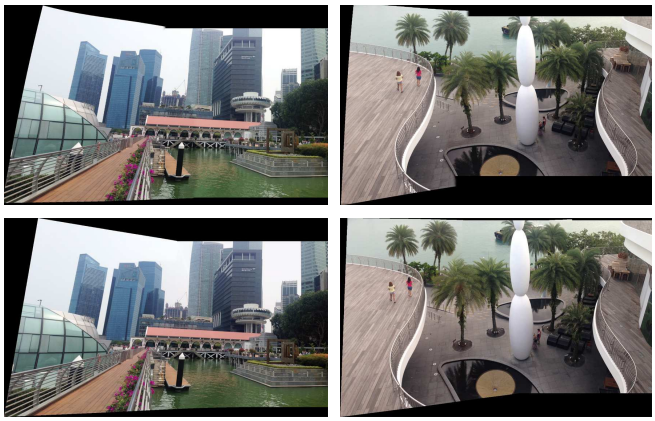


FIGURE 12. Comparison with SEAGULL [43] for parallax-tolerant image stitching. Top: results from the proposed GCPW. Bottom: results from SEAGULL.

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

FIGURE 13. An extremely challenging scene for image stitching, in which two challenges occur at the same time. Images to be stitched lack rich texture and have large parallax either. (a) initial stitching result produced by seamline searching. It suffers obvious misalignment (as yellow arrow indicates). (b) result produced by SEAGULL [43]. Misalignment artifacts are not effectively eliminated. (c) result produced by proposed method. (d) final stitching result after image blending.

B. VIDEO STABILIZATION

The technique of video stabilization aims to remove unwanted camera motion in shaky videos that are usually captured by some hand-held devices. Many methods [27], [44], [50] are therefore proposed and they often involve two steps: camera path estimation and camera path stabilization. Usually, camera path is represented by the motion between each pair of adjacent video frames, and it indeed plays an important role in video stabilization.

When referring to the first challenge, it is just similar to the one of image stitching. Once the captured scene is lack of sufficient features, previous methods may fail to obtain accurate motion estimation, which finally leads to unsatisfactory stabilization results. Fig. 14 gives two example videos that are less-textured. In order to demonstrate that the proposed GCPW is effective under such a challenging scene, we apply GCPW to estimation camera path, followed by a typical camera path optimization scheme [50] to produce final video stabilization result. We also stabilize the same video using other three popular methods: Subspace [51] (with implementation in Adobe After Effects), VirtualDub Deshaker [2](with offered software), and GeoStab [52](with offered executable file). These methods approximate camera path by either feature tracking or feature matching. In order to compare the video stabilization performance quantitatively,

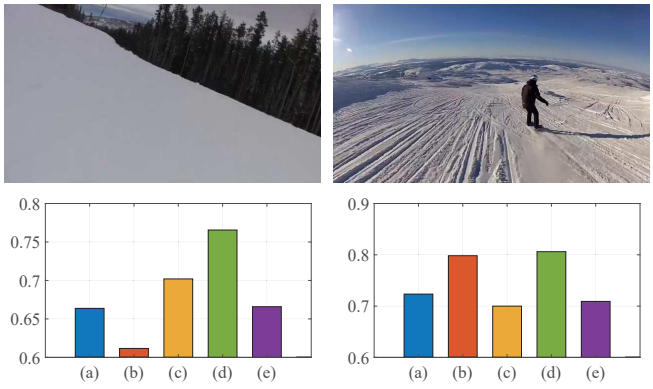


FIGURE 14. Comparison with three popular video stabilization methods. The evaluation value reported in the bottom row is produced by recent video stability assessment method [53]. (a) Results from original shaky video. (b) Results from Subspace [51]. (c) Results from Virtual Deshaker. (d) Results from proposed GCPW. (e) Results from GeoStab [52]. Higher value usually means better video stabilization result.

all stabilized results as well as the original video are evaluated by the intrinsic motion stability assessment [53]. Fig. 14 presents the reciprocal of applied evaluation metric, in which higher values mean better results. As we can see, other three compared methods may fail to produce good results, and sometimes, they may deteriorate video content and therefore obtain lower scores than the original shaky video. In contrast, our method outputs more stable results and win the highest score.

[2]http://www.guthspot.se/video/deshaker.htm

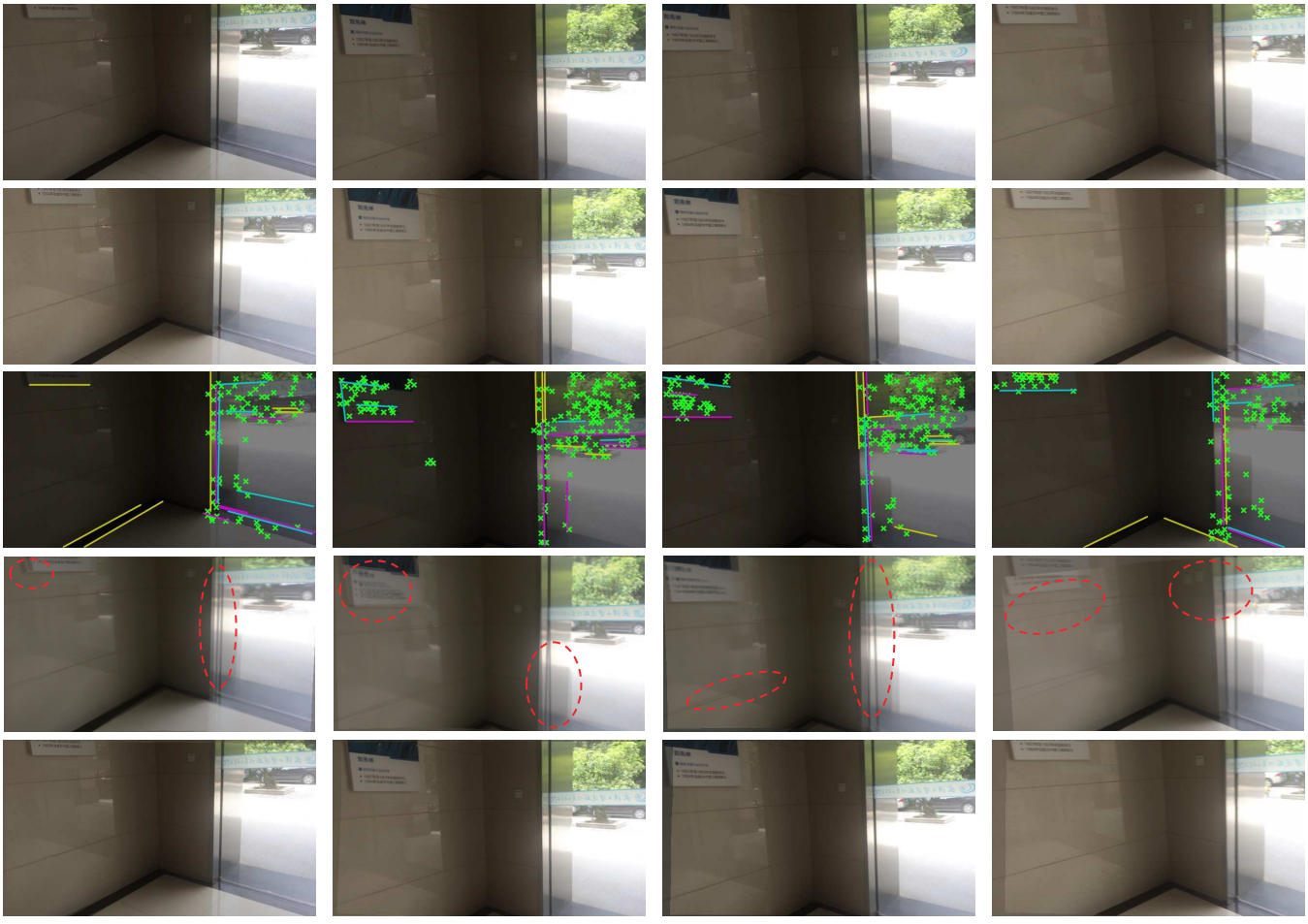


FIGURE 15. An illustration of motion estimation results on sampled video frame pairs. 1-st row: current frame. 2-nd row: next frame. 3-rd row: a visualization of feature matching result. 4-th row: alignment results from MPA [10]. 5-th row: alignment result from proposed GCPW.

Significant color difference is the second challenge for camera path estimation. Actually, a shaky video usually contains many high-frequency motions, such as rolling shutter or quick rotation. Due to these high-frequency motions, two consecutive video frames may have apparent color differences. It is challenging for recent MPA [10], since the required color consistency assumption has been violated. It should be noted that incorrect motion estimations finally result in unsatisfactory stabilization results with severe distortion or skew artifacts [9]. Fig. 15 gives an example that a shaky video is captured using an ordinary cellphone. On the one hand, the captured video lacks rich textures; On the other hand, some adjacent frames have noticeable color differences. We display the feature detection results of these frame pairs in Fig. 15. In the presented results, we can observe that the extracted features are usually so inadequate to estimate motions accurately, either for methods based on feature tracking [27] or for methods based on feature matching [44]. Besides, color variations between consecutive two video frames make MPA also fail to produce correct motion estimation results. In contrast, the proposed GCPW stably obtain satisfactory consequences, free from affections of low texture or color variation.

VI. CONCLUSION

Motion estimation plays an important part in many computer vision applications. Conventional feature-based methods usually fail for low-texture scenes. In this paper, we resort to photometric constraints to produce better motion estimation results. In order to make the applied photometric constraints robust to color variation, we propose the GCPW framework, in which the color transformation between images are modeled, and motion-related mesh vertexes and color-related mapping parameters are optimized jointly to obtain more accurate motion estimation results. We evaluate the proposed method on tens of videos. These videos contain ordinary as well as low-texture scenes, cover several typical motion modes, and include thousands of frame pairs. The results reveal that our method can estimate motion more accurately, both for ordinary videos and low-texture videos. Besides, a synthetic experiment is designed to estimate motions between images with different degrees of color variation. Experimental results prove that our method is robust to color difference. Finally, since motion estimation is the

basis of many computer vision applications, we further explore the possibility of GCPW being applied into two areas: image stitching and video stabilization. Some intuitive results demonstrate that our method is effective to handle some challenging scenes within above two application fields.

REFERENCES

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

[2] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger, "Probability distributions of optical flow," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1991.

[3] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in European Conference on Computer Vision (ECCV), 2004.

[4] D. Sun, S. Roth, and M. J. Black, "Secrets of optical flow estimation and their principles," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in IEEE International Conference on Computer Vision (ICCV), 2015.

[6] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," International Journal of Computer Vision, vol. 74, no. 1, pp. 59–73, 2007.

[7] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," in ACM Transactions on Graphics, vol. 28, p. 44, ACM, 2009.

[8] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving dlt," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[9] S. Li, L. Yuan, J. Sun, and L. Quan, "Dual-feature warping-based motion model estimation," in IEEE International Conference on Computer Vision (ICCV), 2015.

[10] K. Lin, N. Jiang, S. Liu, L.-F. Cheong, and M. D. J. Lu, "Direct photometric alignment by mesh deformation," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[11] B. D. Lucas, T. Kanade, et al., "An iterative image registration technique with an application to stereo vision," in Internation Joint Conference on Artificial Intelligence (IJCAI), 1981.

[12] A. Wedel, D. Cremers, T. Pock, and H. Bischof, "Structure-and motion-adaptive regularization for high accuracy optic flow," in IEEE International Conference on Computer Vision (ICCV), 2009.

[13] Y. Mileva, A. Bruhn, and J. Weickert, "Illumination-robust variational optical flow with photometric invariants," in Joint Pattern Recognition Symposium, 2007.

[14] O. Demetz, M. Stoll, S. Volz, J. Weickert, and A. Bruhn, "Learning brightness transfer functions for the joint recovery of illumination changes and optical flow," in European Conference on Computer Vision (ECCV), 2014.

[15] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[16] W.-S. Lai, J.-B. Huang, and M.-H. Yang, "Semi-supervised learning for optical flow with generative adversarial networks," in Advances in Neural Information Processing Systems (NIPS), 2017.

[17] A. Ranjan, V. Jampani, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Adversarial collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," arXiv preprint arXiv:1805.09806, 2018.

[18] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," arXiv preprint arXiv:1711.07837, 2017.

[19] Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H.-Y. Shum, "Full-frame video stabilization with motion inpainting," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 7, pp. 1150–1163, 2006.

[20] C.-H. Chang, C.-N. Chou, and E. Y. Chang, "Clkn: Cascaded lucas-kanade networks for image alignment," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[21] J. Gao, S. J. Kim, and M. S. Brown, "Constructing image panoramas using dual-homography warping," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[22] R. Hartley and A. Zisserman, Multiple view geometry in computer vision. Cambridge university press, 2003.

[23] N. Japkowicz, F. E. Nowruzi, and R. Laganiere, "Homography estimation from image pairs with hierarchical convolutional networks," in IEEE International Conference on Computer Vision Workshop (ICCVW), 2017.

[24] T. Nguyen, S. W. Chen, S. Skandan, C. J. Taylor, and V. Kumar, "Unsupervised deep homography: A fast and robust homography estimation model," IEEE Robotics and Automation Letters, 2018.

[25] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong, "Smoothly varying affine stitching," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[26] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," ACM Transactions on Graphics, vol. 24, no. 3, pp. 1134–1141, 2005.

[27] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng, "Meshflow: Minimum latency online video stabilization," in European Conference on Computer Vision (ECCV), 2016.

[28] J. Shi et al., "Good features to track," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1994.

[29] K. Joo, N. Kim, T.-H. Oh, and I. S. Kweon, "Line meets as-projective-as-possible image stitching with moving dlt," in IEEE International Conference on Image Processing (ICIP), 2015.

[30] T. Xiang, G.-S. Xia, L. Zhang, and N. Huang, "Locally warping-based image stitching by imposing line constraints," in International Conference on Pattern Recognition (ICPR), 2016.

[31] T.-Z. Xiang, G.-S. Xia, X. Bai, and L. Zhang, "Image stitching by line-guided local warping with global similarity constraint," Pattern Recognition, 2018.

[32] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 4, pp. 722–732, 2010.

[33] C. Akinlar and C. Topal, "Edlines: A real-time line segment detector with a false detection control," Pattern Recognition Letters, vol. 32, no. 13, pp. 1633–1642, 2011.

[34] X. Lu, J. Yao, K. Li, and L. Li, "Cannylines: A parameter-free line segment detector," in IEEE International Conference on Image Processing (ICIP), 2015.

[35] Z. Wang, F. Wu, and Z. Hu, "Msld: A robust descriptor for line matching," Pattern Recognition, vol. 42, no. 5, pp. 941–953, 2009.

[36] B. Fan, F. Wu, and Z. Hu, "Line matching leveraged by point correspondences," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[37] K. Li, J. Yao, and X. Lu, "Robust line matching based on ray-point-ray structure descriptor," in Asian Conference on Computer Vision (ACCV), 2014.

[38] K. Li, J. Yao, X. Lu, L. Li, and Z. Zhang, "Hierarchical line matching based on line–junction–line structure descriptor and local homography estimation," Neurocomputing, vol. 184, pp. 207–220, 2016.

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

[40] S. He, Q. Yang, R. W. Lau, J. Wang, and M.-H. Yang, "Visual tracking via locality sensitive histograms," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[41] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," ACM Transactions on Graphics, vol. 30, no. 4, p. 70, 2011.

[42] G. Zhang, Y. He, W. Chen, J. Jia, and H. Bao, "Multi-viewpoint panorama construction with wide-baseline images," IEEE Transactions on Image Processing, vol. 25, no. 7, pp. 3099–3111, 2016.

[43] K. Lin, N. Jiang, L.-F. Cheong, M. Do, and J. Lu, "Seagull: Seam-guided local alignment for parallax-tolerant image stitching," in European Conference on Computer Vision (ECCV), 2016.

[44] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," ACM Transactions on Graphics, vol. 32, no. 4, p. 78, 2013.

[45] M. Grundmann, V. Kwatra, and I. Essa, "Auto-directed video stabilization with robust l1 optimal camera paths," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[46] A. Goldstein and R. Fattal, "Video stabilization using epipolar geometry," ACM Transactions on Graphics, vol. 32, no. 5, 2012.

[47] J. Park, Y.-W. Tai, S. N. Sinha, and I. So Kweon, "Efficient and robust color consistency for community photo collections," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
[48] F. Zhang and F. Liu, "Parallax-tolerant image stitching," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[49] L. Li, J. Yao, X. Lu, J. Tu, and J. Shan, "Optimal seamline detection for multiple image mosaicking via graph cuts," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 113, pp. 1–16, 2016.
[50] S. Liu, L. Yuan, P. Tan, and J. Sun, "Steadyflow: Spatially smooth optical flow for video stabilization," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[51] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," ACM Transactions on Graphics, vol. 30, no. 1, p. 4, 2011.
[52] L. Zhang, X.-Q. Chen, X.-Y. Kong, and H. Huang, "Geodesic video stabilization in transformation space," IEEE Transactions on Image Processing, vol. 26, no. 5, pp. 2219–2229, 2017.
[53] L. Zhang, Q.-Z. Zheng, and H. Huang, "Intrinsic motion stability assessment for video stabilization," IEEE Transactions on Visualization and Computer Graphics, 2018.