DOI: 10.1111/cgf.14512

COMPUTER GRAPHICS *forum*
Volume 00 (2022), number 0 pp. 1–16

Quad-fisheye Image Stitching for Monoscopic Panorama Reconstruction

Haojie Cheng,[1,2] Chunxiao Xu,[1,2] Jiajun Wang[1,2] and Lingxiao Zhao[1,2]

[1]Division of Life Sciences and Medicine, School of Biomedical Engineering (Suzhou), University of Science and Technology of China, Hefei, China
[2]Suzhou Institute of Biomedical Engineering and Technology, Chinese Academy of Sciences, Suzhou, China

Abstract

Monoscopic panorama provides the display of omnidirectional contents surrounding the viewer. An increasingly popular way to reconstruct a panorama is to stitch a collection of fisheye images. However, such non-planar views may result in problems such as distortions and boundary irregularities. In most cases, the computational expense for stitching non-planar images is also too high to satisfy real-time applications. In this paper, a novel monoscopic panorama reconstruction pipeline that produces better quad-fisheye image stitching results for omnidirectional environment viewing is proposed. The main idea is to apply mesh deformation for image alignment. To optimize inter-lens parallaxes, unwarped images are firstly cropped and reshuffled to facilitate the circular environment scene composition by the seamless ring-connection of the panorama borders. Several mesh constraints are then adopted to ensure a high alignment accuracy. After alignment, the boundary of the result is rectified to be rectangular to prevent gapping artefacts. We further extend our approach to video stitching. The temporal smoothness model is added to prevent unexpected artefacts in the panoramic videos. To support interactive applications, our stitching algorithm is programmed using CUDA. The camera motion and average gradient per video frame are further calculated to accelerate for synchronous real-life panoramic scene reconstruction and visualization. Experimental results demonstrate that our method has advantages in respects of alignment accuracy, adaptability and image quality of the stitching result.

Keywords: monoscopic panorama, panorama reconstruction, image stitching, mesh deformation, GPU acceleration

CCS Concepts: • Computing methodologies → Image processing; Computational photography

1. Introduction

The monoscopic panorama is a centre-of-projection image and normally presents as a single equirectangular panoramic photo [TBLG16]. It is typically a cost-effective choice for the omnidirectional photography, by which people can view monoscopic contents on any device. A panoramic image can be used to display the surrounding environment directly and efficiently [dSJ19]. The traditional way that cumbersomely reconstructs all 3D models in the space greatly limits the complexity and lacks of adaptability in computer graphics applications. Besides, the monoscopic panorama can be used as the input for other tasks, e.g. omnidirectional depth estimation [ZKZD18], image-based rendering [EKD*17], 6 degree-of-freedom (DoF) virtual reality (VR) [SKC*19], etc.

One promising way to reconstruct a professional panoramic scene is to use a quad-fisheye camera with four wide-angle lenses. It can have a field-of-view (FOV) larger than 180° per lens (Figure 1) and acquire images simultaneously for four orthogonal viewing directions. These images have large-proportional overlapping areas with each other [JKOK15] and together cover the full 360° viewing range of surroundings [LKK*16]. Compared to other more professional (e.g. multi-camera arrays) or commercial (e.g. dual-fisheye cameras) panorama cameras, the quad-fisheye camera offers an optimal trade-off between hardware cost, image resolution and FOV coverage. After fisheye images have been acquired, they are stitched in a row with an appropriate order to form a panorama. The stitched panoramic image can then be mapped onto a 3D spherical surface and the virtual scene is rendered at user's viewing position in this way to form an omnidirectional display of the environment. It has to present a rectangular boundary [GRE*16] so that the left and right borders can be seamlessly connected. This guarantees that no obvious gaps are visible on the displayed surrounding environment and ensures a good quality of the panoramic view.

Figure 1: *Image acquisition using a quad-fisheye camera. Each lens has a FOV of* 190° *and two adjacent lenses have an overlapping FOV of* 100°.

Image stitching is crucial to the panoramic scene reconstruction. In recent years, related research has been carried out on perspective image stitching [Sze06]. However, not many efforts have been conducted for non-planar image stitching. Perspective image stitching methods based on feature point matching can be extended to align fisheye images. Traditional image stitching methods aim to minimize alignment errors, such as finding global parametric warps to align images [YHL*16]. Global warp models work well for ideal cases when the camera translation is negligible or the scene is nearly planar [BL07]. However, for stitching images of viewpoints of non-planar scenes, undesired artefacts may not be easily disposed.

The spatially varying warp model has been proposed recently to improve the stitching quality [GKB11]. Different spatially varying warp models for image stitching can be classified into two categories, i.e. the alignment model using multiple local homographs [LLM*11] and the mesh deformation model based on the global homograph [ZCT*14]. Although the former improves the alignment accuracy, the non-overlapping image region is always incorrectly stretched, and multiple images with large parallax cannot be stitched accurately [WRHS13]. The latter optimizes the image alignment quality by adding constraint factors to prevent distortions in non-overlapping regions [CSC14].

Hardware profiling of a quad-fisheye camera, e.g. lens aperture, focus, assembly errors, etc., is known by the manufacturer or can be calibrated. Image stitching algorithms applied in software programs offered by camera vendors are strictly dependent on these measurements and are normally non-public. In our research, we developed a camera-independent quad-fisheye image stitching approach, which is preferable for the universal application development without binding to a limited selection of cameras and enclosed commercial software programs. Camera-dependent image stitching approaches are normally based on the rigid image alignment using geometric transformation, while our approach makes use of the deformable registration mechanism based on correlating image characteristics.

This paper proposes a novel monoscopic 360° scene reconstruction pipeline based on our camera-independent quad-fisheye image stitching approach. Acquired fisheye images are firstly unwarped to represent planar views. Then energy functions with constraint factors are applied to the alignment based on deformable meshes to bring homologous points of two images as close as possible. The image stitching order is optimized to preserve a high quality of the omnidirectional viewing. Undistorted images are cropped to optimize inter-lens parallaxes and avoid inconsistent artefacts. To extend our method for stitching videos, the temporal smoothness constraint is used to improve the spatio-temporal smoothness of the deformed mesh. To accelerate the computation, parameters of meshes and textures are calculated on GPU using CUDA. For synchronous panoramic viewing, resulting monoscopic videos are mapped onto a uniform 3D spherical surface to describe omnidirectional information of the scene. Experimental results demonstrate that our approach has advantages of the high alignment accuracy, robustness and adaptability, by efficiently delivering a high image quality of the stitched panorama.

Main contributions of our work are outlined as follows:

- We develop a new mesh-based approach to stitch quad-fisheye images. After inter-lens parallax optimization, several new or revised mesh constraints are applied to eliminate distortions and artefacts of the resulting panoramic image with a regular boundary.
- We extend our fisheye image stitching method for the efficient video stitching to create the synchronous panoramic viewing. Accumulative errors of shared video frames are bounded by calculating camera motion and average gradient per video frame.
- Our camera-independent approach supports interactive applications well. The entire pipeline of our approach can be implemented on GPU using CUDA and achieve a high FPS for the real-time panorama stitching and rendering.

2. Related Research

The monoscopic panorama is created by stitching multiple images with the narrow FOV. Image stitching aims to build the correspondence between multi-view images, while parallaxes and lens distortions are overcome. A comprehensive survey of research for image stitching is given by Szeliski [Sze06].

Global alignment models. Most traditional image stitching methods make use of global warps, e.g. global affine and homography transformation, to align images [Sze06]. Conventional methods based on global alignment models normally are feature-based. Lin *et al*. [LLM*11] proposed the smoothly varying affine (SVA) warp that maintains a global affinity for adaptive image stitching. Due to the higher DoF, the homography model is more flexible to align images from different perspectives [BL07]. Brown *et al*. [BL07] proposed an automatic stitching method using a global homography. The global homography model works well for images of planar views. In recent years, deep learning methods have shown their potentials to stitch images with weak constraints. DeTone *et al*. [DMR16] leveraged the deep convolutional neural network (CNN) to estimate relative homography between a pair of images. Nie *et al*. [NLL*20] presented the view-free image stitching network based

on a global homography. Current networks can only be trained using a dataset of parallax-free and synthetic images. Existing data-driven methods cannot provide desired stitching results for real-world scenes.

Feature-based adaptive homography models. Decomposing a single global warp model into multiple local warp models can achieve a better alignment accuracy. Gao *et al*. [GKB11] proposed a dual-homography warp to process scenes with two dominant planes using a weighted sum of two homographies.

Some research focused on superpixel segmentation methods, Chaurasia *et al*. [CDSHD13] performed local alignment warps on each superpixel individually. Lee *et al*. [LS20] estimated multiple homographies and found their inlier feature matches between two images. Some other research focused on mesh-based methods. Zaragoza *et al*. [ZCT*14] tackled the limited performance for extrapolating the local affine model and proposed the as-projective-as-possible (APAP) warp for the accurate local adaptation with a good extrapolation. Inspired by the effectiveness of APAP, Chang *et al*. [CSC14] proposed a shape-preserving half-projective (SPHP) warp, which spatially combines a homography warp with a similarity warp. Lin *et al*. [LPRA15] proposed an adaptive as-natural-as-possible (AANAP) warp that uses a smooth stitching field derived from the local homography or its linearized version and a global similarity transformation. Chen *et al*. [CC16] proposed a global-similarity-prior (GSP) warp and designed a scheme to automatically select the proper global scale and rotation for each image to be aligned. Lee *et al*. [LKK*16] also split a panorama into multiple cells and designed a deformable spherical projection model to minimize stitching artefacts caused by large parallax in the views.

Panoramic image generation. In recent years, imaging applications have been pursuing the one-time acquisition of a panoramic scene. A single camera integrated with multiple wide-angle lenses or camera arrays can satisfy this demand. Ho *et al*. [HSRB17] made use of a dual-fisheye image stitching method based on rigid moving least squares. Huang *et al*. [HCR*17] enhanced the standard structure-from-motion (SfM) method to work with dual-fisheye 360° videos. Perazzi *et al*. [PSZ*15] used unstructured camera arrays to generate panoramic videos. They computed pairwise warps for robust video stitching with minimal parallax artefacts. Lai *et al*. [LGG*19] proposed a novel pushbroom stitching network to generate video panoramas. However, these methods suffer from undesired distortions when more images are stitched.

As the stitching result, a panoramic image should have a rectangular boundary so that the 3D panoramic scene can be reconstructed directly by circular connecting and spherical mapping [MCE*17]. Otherwise, gapping artefacts can be introduced and destroy the visual consistency. He *et al*. [HCS13] proposed a content-aware warping method to fix panoramas with irregular boundaries. Although their method works well for images of planar views, their two-step warping strategy is based on line detection and not suitable for stitching non-planar images. Zhang *et al*. [ZLZ20] proposed a content-preserving image stitching method that setups piecewise rectangular boundary constraints to restore the naturalness of stitched image contents. Won *et al*. [WRL20] presented large-scale panorama datasets of omnidirectional multi-view images for training and testing data-driven methods. Fisheye images can leverage data-driven methods to generate panoramic images with rectangular boundaries.

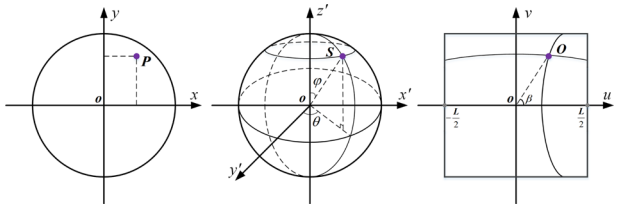


Figure 2: *Fisheye image unwarping. Left: a position P in the fisheye image; middle: P is mapped to S on the 3D unit sphere; right: S is projected to the 2D orthogonal coordinate system O.*

3. Quad-fisheye Image Stitching

The pipeline of our quad-fisheye image stitching method is elaborated in this section. The input is a set of four fisheye images captured simultaneously using a quad-fisheye camera. Each fisheye image is fitted with a deformable mesh and divided into small patches by grid cells. These fisheye images are aligned by deforming fitting meshes and stretching the image correspondingly with constraints defined as energy functions. The output of the stitching process is a rectangular panoramic image that correctly stitches all four fisheye images together with minimum distortions.

3.1. Fisheye image unwarping

According to the optical principle of quad-fisheye camera lenses, the acquired fisheye image initially represents a non-planar view with tremendous distortions in its periphery [HSRB17]. It is highly necessary to unwarp the fisheye image before stitching to eliminate distortions. The centre of the circular content area in the fisheye image should be identified first. It normally coincides with the image focus position. The minimum circumscribed circle is computed to extract the circular content area and the circle centre is taken as the focus position. This circular area is then bounded by a fitting square to crop the original fisheye image.

To unwarp the cropped fisheye image of an edge length L, each image point $P(x, y)$ is firstly mapped onto a 3D unit sphere as spherical coordinates $S(\theta, \varphi, 1)$ (Figure 2). In practice, due to the design or manufacturing deviation, the centerline of the fisheye camera lens inevitably has a pitch angle $\rho \neq 0$ to the camera horizontal plane. The unwarping may suffer from ignoring this pitch angle ρ and result in serious distortions along the image horizontal middle line [RPZSH13]. ρ can be calibrated [Zha00] or learned from the manufacturer, and cartesian coordinates $S'(x', y', z')$ of $S(\theta, \varphi, 1)$ are calculated as the following:

$$\begin{aligned} x' &= \sin\theta \cdot \sin\varphi \\ y' &= (\cos\theta \cdot \sin\varphi \cdot \cos\rho) - (\cos\varphi \cdot \sin\rho) \\ z' &= (\cos\theta \cdot \sin\varphi \cdot \sin\rho) + (\cos\varphi \cdot \cos\rho) \end{aligned} \tag{1}$$

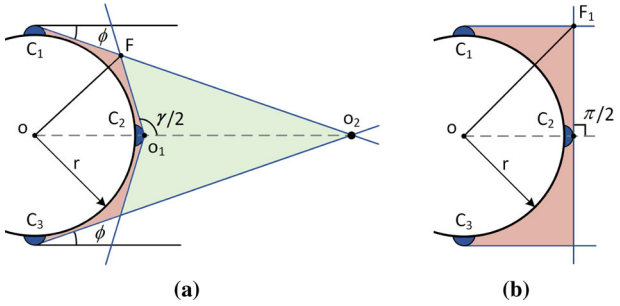


Figure 3: *Inter-lens parallax optimization between adjacent camera lenses $\{C_1, C_2, C_3\}$. Green and red areas represent blind regions of camera lenses.*

Then point S' is projected to a 2D orthogonal coordinate system using spherical projection as $O(u, v)$:

$$u = L\left(\frac{1}{\gamma}\tan^{-1}\left(\sqrt{(x')^2 + (z')^2}/y'\right)\right)\cos\beta$$
$$v = L\left(\frac{1}{\gamma}\tan^{-1}\left(\sqrt{(x')^2 + (z')^2}/y'\right)\right)\sin\beta \quad (2)$$

where γ indicates the FOV of each camera lens and $\beta = \tan^{-1}(z'/x')$. Instead of rectifying distortions after stitching by a 2D correction transformation [HZ04], our approach unwarps input fisheye images using the spherical projection in advance. This also guarantees that the panorama suits for the 360° viewing when it is rendered on a spherical surface. Before practical application, we just need one calibration to obtain parameters of fisheye image unwarping for a given camera. These precomputed parameters can then be reused from a lookup table (LUT) for each image stitching.

3.2. Inter-lens parallax optimization

The image stitching approach aims to convert views with inter-lens parallaxes into a consistent stitching result. The object closer to the camera makes a larger parallax between adjacent images. Note that an object that is too close to the camera might be observed by only one lens [SBSH18]. The feature correspondence between matched images cannot be correctly built in such a case. This fact can lead to serious artefacts in the stitched image. As shown in Figure 3a, the minimum parallax-tolerant depth d is defined as:

$$d = \|O_1O_2\| = \frac{r}{\tan\phi} - r \quad (3)$$

where r is the camera rig radius and $\phi = (\gamma - \pi)/2$. γ indicates the FOV of each camera lens. $d_1 = \|OF - r\|$ is the minimum visible depth of the camera. In other words, if the distance between an object and the camera is less than d_1 (i.e. in the red region), all lenses cannot observe this object. Parallaxes will be a minor issue if the camera is used to capture environments with far away objects [LS20]. Therefore, inter-lens parallaxes can be optimized by increasing d. We crop each unwarped image to convert the horizontal FOV from γ to π (i.e. $\phi = 0$ and $d = +\infty$), as shown in Figure 3b. Although the new minimum visible depth $d_2 = \|OF_1 - r\|$ is slightly larger than d_1, artefacts in the stitched image can be effectively reduced. Intuitive experiments are discussed in Section 5.2. In our method, as long as the distance between the object and the camera is greater than d_2, adjacent images can be stitched more accurately.

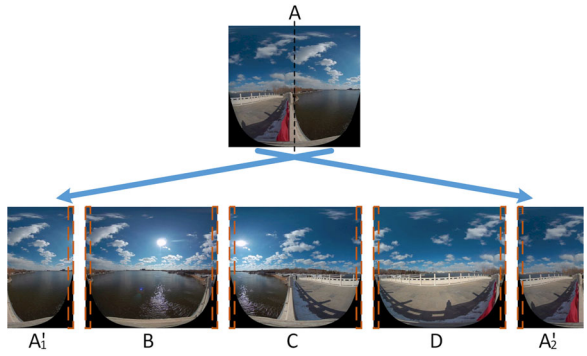


Figure 4: *Reshuffle quad-fisheye images: the first image* **A** *is split into two equal sub-images* $\mathbf{A}'_1$ *and* $\mathbf{A}'_2$*, which are stitched with fisheye image B, C and D sequentially. The dash box represents cropped regions according to the minimum parallax-tolerant depth.*

As in a looped sequence of fisheye images, the left part of the first image overlaps with the right part of the last image. To reconstruct the panoramic view, the left part of the panorama is connected to its right part. A simple and direct way is to crop or merge duplicated areas in the stitching result. However, important contents are very likely to be cut off by mistake and gaps are possibly introduced. Ghosting artefacts are also easily generated by the merging in such a way.

In our approach, input quad-fisheye images are reshuffled before stitching. The first source image is split into two equal sub-images, which are then stitched, respectively, with the remaining three images as shown in Figure 4. This facilitates connecting the left and right sides of the resulting image to form a circular panorama and ensures its display without distortions and gaps.

3.3. Alignment based on matching points

Our fisheye image stitching approach makes use of deformable meshes fitted to source images. The estimation of how fitting meshes are deformed for the image alignment is based on correspondences between matching points, which are pairs of associated image feature points. Each of two paired matching points is identified on either of the two source images to be aligned.

Let I_i and I_j denote two adjacent unwarped images. $i, j \in [1, 5]$ indicate image indices. To ensure a minimized elapsed time of processing, it can be programmed with hardware-acceleration using GPU. In our experiment, CUDA-based ORB [RRKB11] is used to calculate matching points for I_i, I_j. Then the global alignment transformation between image I_i and I_j is measured by calculating the global alignment transformation $H \in \mathbb{R}^{3\times3}$ [HZ04] between matching points on them. Given the kth pair of matching points $p = [x, y]^T \in I_i$ and $p' = [x', y']^T \in I_j$, $k \in [1, N]$ and N is the

number of matching point pairs on images I_i and I_j, the correspondence between homogeneous coordinates p and p' is a projective homography represented as a planar transformation $\tilde{p}' = H\tilde{p}$. Let $h = [h_1 h_2 \dots h_9]^T$ reform H, where $\|h\| = 1$, its items can be estimated using DLT [ZCT*14] to solve the following equation:

$$\hat{h} = \arg\min_{h} \sum_{k=1}^{N} \|a_k h\|^2 \tag{4}$$

where $a_k \in \mathbb{R}^{2\times 9}$ is defined as the following:

$$a_k = \begin{bmatrix} -x & -y & -1 & 0 & 0 & 0 & xx' & yx' & x' \\ 0 & 0 & 0 & -x & -y & -1 & xy' & yy' & y' \end{bmatrix} \tag{5}$$

H can be obtained by reshaping $\hat{h}$ to a 3×3 matrix. The pixel per position p_* in image I_i is aligned to position p'_* in image I_j using $\tilde{p}'_* = H\tilde{p}_*$.

By definition, matching points are normally distributed non-uniformly when the image texture information is insufficient. Since adjacent fisheye images do not differ purely by rotation, ghosting artefacts are generated if two images are aligned only based on the globally uniform transformation H. To address this problem, local image structures are taken into account for the alignment. The local alignment transformation H_* is used to replace the global H at p_* and H_* varies at different image positions. H_* is calculated locally based on the given H by using DLT to solve the equation:

$$\hat{h}_* = \arg\min_{h} \sum_{k=1}^{N} \left\| w_*^k a_k h \right\|^2 \tag{6}$$

where a_k is defined for the kth pair of matching points (p, p') in Equation (5), $\hat{h}_*$ is the reshaped form of H_*, the weighted factor w_*^k is calculated based on the distance from p_* to a kth matching point p in the same unwarped fisheye image:

$$w_*^k = \max\left(\exp\left(-\frac{\|p_* - p\|^2}{\sigma^2}\right), \gamma\right) \tag{7}$$

where σ is a scalar factor and $\gamma \in [0.0, 1.0]$ is used to offset the weight w_*^k. As indicated by Equation (7), higher weights are assigned to local structures closer to p_*. H_* pays more attention to the local context than H of the overall image.

3.4. Mesh registration

For image alignment using localized transformation $\tilde{p}'_* = H_*\tilde{p}_*$, it is unnecessary to compute H_* per image pixel position p_*. A more efficient way is to guide the alignment using a mesh-based method [SMW06]. Each of two adjacent unwarped images (I_i, I_j) to be stitched is fitted with a mesh grid. Each mesh grid cell contains a sub-image patch. H_* is only estimated at mesh grid vertices. Then it is used to relocate corresponding mesh vertices fitted in I_i and I_j to align pre-identified matching points. In such a way, I_i and I_j are aligned along with the deformation of the two meshes fitted to overlapping areas of them [CC16].

Given the mesh cell deformation guided by local transformation H_* at each cell vertex, pixels of the sub-image patch within the mesh cell are aligned from source image to target image using bilinear interpolation [ZCT*14]. Due to the content complexity of the source image, the fitting mesh cell may have too much deformation flexibility for the image alignment. Therefore, mesh-based method may generate overstretching artefacts and introduce undesired distortions in the stitching result. To eliminate overstretch and restore naturalness of image structures, constraints are added to suppress the mesh deformation uncertainty.

In our approach, after mesh vertices $V = [v_1 v_2 \dots v_n]$ on the source image have been transformed to $W = [w_1 w_2 \dots w_n]$ on the target image using H_*, the *total mesh registration energy function* $Q_T(W)$ is applied to adjust W for the image alignment. $Q_T(W)$ composes of *image registration function* $Q_R(W)$, *boundary rectification function* $Q_b(W)$ and *temporal smoothness function* $Q_s(W)$. $Q_R(W)$ offers the trade-off between the alignment accuracy and the resulting image naturalness. It includes *feature alignment function* $Q_a(W)$, *local structure preserving function* $Q_l(W)$ and *global similarity function* $Q_g(W)$. $Q_b(W)$ is used to prevent image contents from being discontinuous. $Q_s(W)$ aims to prevent the mesh jittering effect and maintain the temporal consistency of stitching results for stitching fisheye frames of a video. $Q_T(W)$ and $Q_R(W)$ are defined as the following:

$$\begin{aligned} Q_T(W) &= Q_R(W) + \beta_b Q_b(W) + \beta_s Q_s(W) \\ Q_R(W) &= \beta_a Q_a(W) + \beta_l Q_l(W) + \beta_g Q_g(W) \end{aligned} \tag{8}$$

where β_a, β_l, β_g, β_b and β_s are weighting coefficients that reflect the proportional significance of each constraint to the mesh deformation for image stitching. These coefficients need to satisfy the condition: $\beta_a + \beta_l + \beta_g + \beta_b + \beta_s = 1.0$. In practice, these constraints can affect each other. More details are described in following sub-sections. In our experiments, a setting of $\beta_a = 0.2$, $\beta_l = 0.2$, $\beta_g = 0.1$, $\beta_b = 0.3$ and $\beta_s = 0.2$ yielded the best result. $Q_a(W)$ is directly related to the alignment accuracy. Therefore, β_a cannot be set very small. Besides, if $Q_b(W)$ is not given with the highest weight among the five constraint factors, gaps may appear when the panoramic scene is reconstructed.

Note that these energy functions are quadratic. Their intentions are to minimize the distance of each pair of matching points to be aligned and restrict the variation of a fitting mesh shape before and after the alignment. The input of Equation (8) are vertices W of the transformed mesh after pairing identified matching points. $Q_T(W)$ should be satisfied with least square solutions for all its constraints and optimized mesh vertices $\hat{V}$ can, therefore, be obtained by recursively tuning W. For this purpose, a sparse linear solver [LL19] is applied:

$$\hat{V} = \arg\min_{W} Q_T(W) = \arg\min_{W} \left\| K^T W - B^T \right\|^2 \tag{9}$$

where $K = [k_a\, k_l\, k_g\, k_b\, k_s]$ is the slope. $B = [b_a\, b_l\, b_g\, b_b\, b_s]$ is the intercept of the linear correlation and corresponds to all constraints of $Q_T(W)$. After $\hat{V}$ is calculated, each pixel in the unwarped image is then transformed using the bilinear interpolation according to its bounding mesh cell deformation. Energy functions and constraint

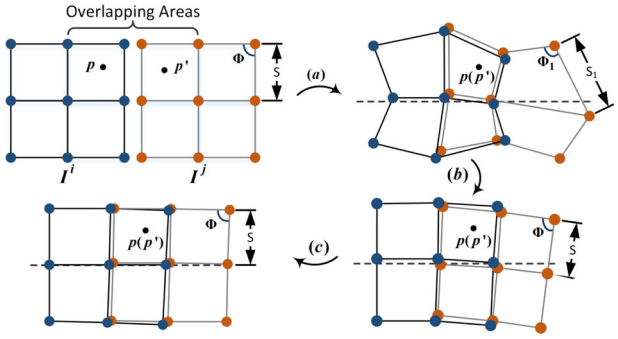


Figure 5: *The registration function $Q_R(W)$ used in our fitting mesh alignment: (a) two meshes fitted to images (I_i, I_j) are aligned using $Q_a(W)$ to guarantee that every pair of matching points (p, p') is aligned accurately in overlapping areas. (b) $Q_l(W)$ restricts the mesh deformation by maintaining geometric properties of the mesh, e.g. the inner angle Φ and the edge length s. (c) $Q_g(W)$ ensures that the resulting image centerline is horizontal to the camera.*

factors used for our fitting mesh alignment are discussed in following paragraphs.

3.4.1. *Feature alignment*

The feature alignment function ensures that matching points are correctly aligned according to their correspondences after the mesh deformation, as shown in Figure 5a. The constraint energy function for feature alignment is defined as:

$$Q_a(W) = \sum_{i \in I} \sum_{j \in \mathcal{N}(I_i)} \sum_{k \in M_{ij}} \left\| \alpha_{ik} \cdot w_{ik} - \alpha_{jk} \cdot w_{jk} \right\|^2 \tag{10}$$

where α_{ik} and α_{jk} denote the weighted factor dependent on how a transformed mesh vertex, e.g. w_{ik} or w_{jk}, is located within an original fitting mesh cell on the target image I_j or I_i. For more details, please refer to Lin *et al.* [LPRA15]. M_{ij} consists of all k matching vertices between images I_i and I_j. $\mathcal{N}(I_i)$ represents all adjacent images that overlap with the image I_i. By resolving the least square solution of the multivariable first-order equation (10), the distance between each of paired matching points is guaranteed to be as short as possible in the overlapping image areas.

3.4.2. *Local structure preserving*

The local structure preserving function limits the degree of the mesh deformation to prevent overstretching artefacts. Let $E = [e_{i1} e_{i2} \dots e_{in}]$ denote original fitting mesh edges on image I_i and $F = [f_{i1} f_{i2} \dots f_{in}]$ denote deformed mesh edges, respectively. n is the number of mesh edges in each image. The constraint energy function for local structure preserving is defined as:

$$Q_l(W) = \sum_{i \in I} \sum_{e \in E, f \in F} \sum_{k \in M_{ef}} \alpha_l(e_k) \| f_k - T e_k \|^2 \tag{11}$$

where M_{ef} are matched edges in the image I_i. T is the similarity transformation matrix [II11] that takes all neighbouring mesh vertices around edge e_k into account. As shown in Figure 5b, serious overstretching artefacts can be eliminated in overlapping image areas with less texture information and non-overlapping areas.

Note that the computation to obtain the circular content area of the fisheye image inevitably has errors as described in Section 3.1, particularly for pixels in the image periphery. Although $Q_a(W)$ can be adopted to accurately align pixels far away from the centre of a fisheye image, $Q_l(W)$ limits the flexibility of mesh grids. Therefore, a relatively flexible restriction should be provided to the mesh deformation. A novel weighted factor $\alpha_l(e_k)$ is defined to control the local structure preserving factor:

$$\alpha_l(e_k) = \begin{cases} \mathcal{D}(y) / \left(\sum_{c \in \mathcal{N}(e_k)} G_c + 1 \right), & if\ e_k \in \Omega \\ \mathcal{D}(y), & if\ e_k \notin \Omega \end{cases} \tag{12}$$

where $\mathcal{D}(y) = \sqrt{1 - |H - 2y|/H}$. H is the height and y is the e_k midpoint ordinate in the unwarped image. Neighbouring cells of the edge e_k can be expressed with a neighbourhood-operator $\mathcal{N}$. In overlapping areas Ω, $\alpha_l(e_k)$ not only is affected by y, but also depends on G_c, which denotes the amount of matching points in $\mathcal{N}(e_k)$. If y is a constant, $\alpha_l(e_k)$ complies with the fact that the richer local texture information is, the smaller $\alpha_l(e_k)$ is, and vice versa. In non-overlapping areas, $\alpha_l(e_k)$ increases non-linearly while the $|H - 2y|$ gradually decreases. Equation (11) is resolved to get the least square solution, which guarantees that the position correspondence between a specific mesh vertex and its neighbouring vertices should keep their relative positions as consistent as possible after the mesh deformation for the image alignment.

3.4.3. *Global similarity*

The global similarity constraint is crucial to prevent source images to be oblique during stitching and maintain the naturalness of the stitched panorama, as shown in Figure 5c. Since a fisheye image presents a non-planar view, the barrel-shaped distortion inevitably exists. The closer an image object is to the centre of the lens, the more natural its structure looks, and vice versa. The energy function constrained for global similarity is defined as:

$$\begin{aligned} Q_g(W) &= \sum_{i \in I} \sum_{e_k \in E} \alpha_g(e_k) F(T) \\ F(T) &= \| T_x - s_i \cos\theta_i \|^2 + \left\| T_y - s_i \sin\theta_i \right\|^2 \end{aligned} \tag{13}$$

where s_i is the scaling factor and θ_i is the rotating factor for each source image. They can be calculated using the *bundle adjustment method* [CC16] that solves the unification of parameters for all lenses. Assuming all lenses of a quad-fisheye camera have identical structures, s_i and θ_i can be pre-calculated and used for stitching all fisheye images. T_x and T_y are components of T indicated by Equation (11) on x and y axes.

The weighted factor $\alpha_g(e_k)$ is used to smooth the deformable mesh transformation across overlapping and non-overlapping areas of unwarped images. A more computationally expensive definition of $\alpha_g(e_k)$ is given in Chen and Chuang [CC16]. In our quad-fisheye

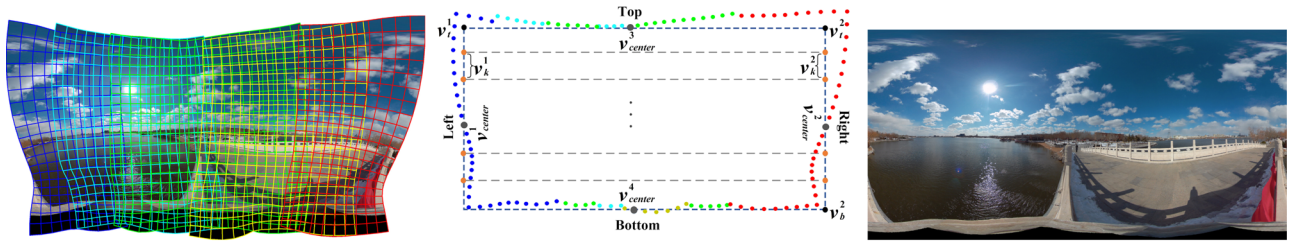


Figure 6: *The proposed quad-fisheye image stitching approach generates rectangular panoramas. (a) Images are aligned using deformable meshes, then (b) boarders of the stitching result are rectified to be regularly straight, finally (c) occluded areas are replenished to form a rectangular panorama.*

image stitching algorithm, it is simplified to reduce the computational complexity:

$$\alpha_g(e_k) = \delta + \frac{d(e_k, \Omega)}{L} \tag{14}$$

where $d(e_k, \Omega)$ calculates the distance between the edge e_k and the overlapping areas Ω of images (I_i, I_j). L indicates the width of the source image. δ is used as a compensation coefficient to control the global similarity constraint and $\delta = 5.0$ offers the best result in our experiment. $\alpha_g(e_k)$ plays a less important role in overlapping regions than non-overlapping regions.

3.4.4. *Boundary rectification*

Irregular boundaries normally present as jagged or tortuous edges of the stitching result of most existing image stitching methods. The 3D display of the panoramic image may strongly suffer from irregular boundaries when it is mapped to the 3D spherical surface. To fix this drawback frequently generated during the deformable mesh alignment (Figure 6a), a novel boundary rectification approach is added to our pipeline as a helpful addition. *Boundary rectification function* composes of *boundary regularity function* $Q_{br}(W)$, *content continuity function* $Q_{bc}(W)$ and *shape constraint function* $Q_{bs}(W)$:

$$Q_b(W) = Q_{br}(W) + Q_{bc}(W) + Q_{bs}(W) \tag{15}$$

Vertices w^c_{center} of deformed meshes at centres of left ($c = 1$), right ($c = 2$), upper ($c = 3$) and lower ($c = 4$) borders B_c of the stitching result are labelled as references, respectively (Figure 6b). Remaining vertices w^c_k of deformed mesh borders are aligned with reference centres along the corresponding panoramic image border. The constraint function for boundary regularity is defined as:

$$Q_{br}(W) = \sum_{c=1}^{4} \sum_{w^c_k \in B_c} \left\| \vartheta \left[w^c_k - w^c_{center} \right] \right\|^2 \tag{16}$$

where $\vartheta \in \mathbb{R}^{2\times 1}$ is used to constrain horizontal and vertical alignments of the mesh border vertex v'^c_k. If $c = 1$ or $c = 2$, then $\vartheta = [1 \quad 0]^T$. Otherwise $\vartheta = [0 \quad 1]^T$.

Omnidirectional content continuity is interrupted by above mentioned energy functions. This is due to the fact that there are no overlapping regions near the left and right borders of the stitched panorama. In addition, vertices of deformed meshes are not aligned vertically. The omnidirectional content continuity is protected by the following function:

$$Q_{bc}(W) = \sum_{w^c_k \in B_c} \left\| \vartheta \left[w^2_k - w^1_k \right] \right\|^2 \tag{17}$$

The stitched rectangular panorama usually has an aspect ratio of 2:1 in our case. If the lower border of the panorama is refined by cropping, contents at the bottom of the panoramic image may be cut off. To avoid this, the shape constraint function is defined as:

$$Q_{bs}(W) = \left\| \vartheta \left[w^2_b - w^2_t \right] - H \right\|^2 + \left\| \vartheta \left[w^2_t - w^1_t \right] - W \right\|^2 \tag{18}$$

where w^c_b and w^c_t are located on the bottom and top mesh borders, respectively. H and W are the height and width of the stitched panorama, respectively. Because the pitch angle exists and the bottom area is occluded to camera lenses, contents at the bottom of the stitched panorama are inevitably missing. This area is replenished with a default black background as shown in Figure 6c.

3.4.5. *Temporal smoothness*

We extend our image stitching method to address the problem of stitching dynamic panoramas. Given a set of input videos (each with T frames), even if the real scene does not rapidly change, matching points detected by the feature detector have subtle differences in each frame. It may cause jittering effects of fitted meshes between adjacent fisheye image frames of the video. In addition, the instability of matching points can change the resolution of the resulting panorama. To solve these problems, the *temporal smoothness function* used to the tth frame ($t \in [2, T]$) is defined as:

$$Q_s(W) = \sum_{i \in I} \sum_{w_k \in W_i} \alpha_s \left\| w^t_k - w^{t-1}_k \right\|^2 \tag{19}$$

The sparse feature point matching may establish incorrect feature correspondences in textureless regions. Furthermore, the stitched panorama can have serious distortions. To avoid these problems, a weighted factor $\alpha_s = N^{t-1}/N^t$ is defined to improve the temporal smoothness in textureless images. N^t indicates the number of feature points at the t th frame. Normally, adjacent video frames of

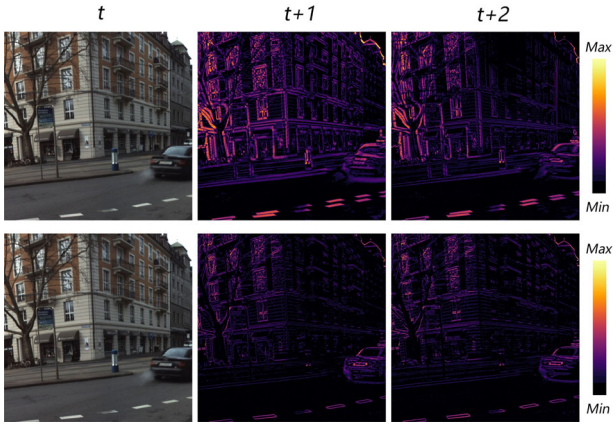


Figure 7: *Comparison results over three consecutive frames obtained with (bottom) and without (top) the temporal smoothness function $Q_s(W)$. Two right columns represent differences maps between the current frame and the previous frame.*

Figure 8: *Our fisheye image blending to eliminate the visual inconsistency in the stitching result. (a) Normal blending method does not consider the interference of dark background colour in unwarped images marked in the pink box. (b) Our image blending method can improve naturalness of aligned overlapping areas.*

the same scene usually only have subtle differences. As shown in Figure 7, $Q_s(W)$ greatly helps prevent jittering effects and improve the visual consistency between adjacent frames of the video stitching result.

3.5. Image blending

To better stitch images of viewpoints of non-planar scenes, existing blending schemes can be applied to reduce the inconsistent artefact caused by the inaccurate pixel alignment [Sze06]. But, due to the existence of the non-zero pitch angle and the nature of fisheye camera lens in our case, black pixels in the background of the unwarped image (Figure 4) may destroy the visual consistency in a single stitched panorama (Figure 8a).

To address this problem, a linear weighted method is used to blend adjacent unwarped images after mesh registration [HB17] (Figure 8b). The alpha value of the dark background in unwarped images is assigned zero so that it does not disturb the blending result. Strong exposure differences between lenses also need to be handled. Gain compensation method [BL07] is applied to eliminate colour or

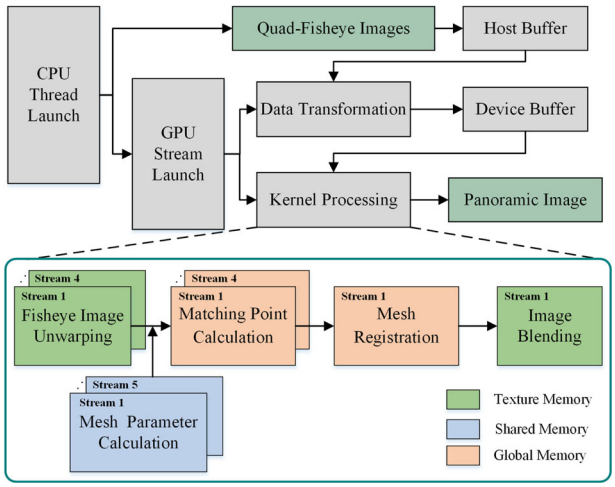


Figure 9: *The CUDA-based GPU implementation of our quad-fisheye image stitching pipeline. Fisheye images are firstly transferred into the GPU memory. Our pipeline is divided into multiple subtasks allocated in different streams and threads.*

brightness differences between source fisheye images. The compensation factor g_{ij} between images I_i and I_j is defined as:

$$g_{ij} = \frac{N_{ij}/\sigma_g^2 + N_{ij}\mathcal{I}_{ij}\mathcal{I}_{ji}/\sigma_N^2}{N_{ij}/\sigma_g^2 + N_{ij}\mathcal{I}_{ij}^2/\sigma_N^2} \tag{20}$$

where N_{ij} indicates the pixel amount of the area in image I_i that overlaps with image I_j. $\mathcal{I}_{ij}$ and $\mathcal{I}_{ji}$ represent average intensities of overlapping areas in image I_i and I_j, respectively. σ_g and σ_N are the gain and the standard deviation of intensity errors, respectively.

4. Real-time 360° Panorama Reconstruction

Panorama reconstruction algorithms applied in software programs offered by camera vendors are strictly dependent on hardware parameters and are normally non-public. If we want to leverage a random quad-fisheye camera for interactive applications [dSJ19], e.g. real-time rendering and VR viewing, the camera-independent image stitching approach with higher FPS is necessary. With the GPU-accelerated implementation, the real-time reconstruction of monoscopic 360° allows to display the captured scene conveniently right after the start of the recording session.

4.1. GPU-accelerated implementation using CUDA

Our quad-fisheye image stitching approach involves intensive computations. The execution of the entire pipeline is computationally expensive. A pure CPU implementation can barely satisfy interactive applications in real time. The GPU implementation architecture of our pipeline is shown in Figure 9.

In our GPU implementation, different graphics memories are used to support various data structures adapted to fit with the GPU architecture, including shared memory, global memory and texture

Figure 10: *The stitched panorama is mapped onto the environment sphere to create a 3D interactive omnidirectional view.*

memory. In addition, independent tasks can be performed in multiple GPU streams asynchronously.

First, input quad-fisheye camera images are placed in four separate streams to perform fisheye image unwarping (Section 3.1) using the LUT. After inter-lens parallax optimization (Section 3.2), parameters of five fitting meshes are calculated per fisheye image in five new different streams. More specifically, geometric parameters of meshes, i.e. position and shape, are calculated and prepared for the subsequent mesh registration. Then, another four streams are created to identify matching points for four pairs of two adjacent source images (Section 3.3). This task is the most time-consuming step in our pipeline. Finally, all image data and fitting meshes are merged into one stream to perform the mesh registration for stitching (Section 3.4) and then blend resulting images (Section 3.5). The mesh registration is the core task, but its computational complexity is not higher than other tasks. Due to the limited number of mesh grids, some atomic operations (such as sum operations in all energy functions indicated by Equation 8) do not need to be accelerated using the GPU parallel processing. To speed up the sparse linear solution indicated by Equation (9), the multi-thread mechanism offered by CUDA can be used to improve the computational efficiency.

4.2. Synchronous panoramic viewing

To produce a 3-DoF 360° viewing, the conversion from 2D equirectangular image coordinates $p(x, y)$ to 3D unit spherical surface coordinates $P(\varphi, \theta)$ is defined as $\theta = 2\pi x/W$ and $\varphi = \pi y/H$, where W and H are width and height of the stitched panoramic image, respectively (Figure 10). After the coordinate system transferring, users can then perform panoramic viewing by adjusting the rendering camera direction. In such a way, the integrative visual effect of the stitched panorama created using our fisheye image stitching algorithm can be demonstrated.

In our experiment, our GPU-accelerated quad-fisheye image stitching is extended for stitching quad-fisheye videos and applied for the synchronous scene reconstruction in real time. The input is a time-sequential series of quad-fisheye images, each of which represents an instant real-life environment scene. Such image series form panoramic videos that can be transmitted using HDMI or Wi-Fi in real time.

To synchronously capture quad-fisheye images from four camera lenses or extract frames from a quad-fisheye video, a parallel method based on CPU multi-threading is implemented. In our pipeline, parameters used in fisheye image unwarping and mesh registration are invariable for image frames per camera lens. These parameters are pre-estimated once and applied throughout the entire video stitching process to avoid redundant computational costs.

Although the matching point calculation (Section 3.3) and mesh registration (Section 3.4) are accelerated using GPU, it is hard to satisfy high computational efficiency especially for high-resolution images. This tremendously annoys the real-time application that processes time-sequential image series. For this reason, matching points on a set of four unwarped source images successfully stitched for one panoramic frame are shared with a few following frames. Since relative positions of different camera lenses are fixed, all frames can theoretically use same mesh registration parameters. However, sparse matching points cannot provide comprehensive correspondences of all pixels. The temporal difference inevitably leads to accumulative errors in the dynamic scene.

To solve this issue, the movement distance of the camera $\mathcal{L}$ and the average gradient per frame $\mathcal{G}$ are calculated to determine the number of shared frames. In our experiments, if $\mathcal{L} > 5$ m or $\mathcal{G} > 10$ pixels, mesh registration parameters should be updated, and vice versa. $\mathcal{G}$ is used to evaluate the alignment quality of the stitching result [PR17]. The motion per lens is synchronized with the camera. To efficiently calculate $\mathcal{L}$, a specific lens of the quad-fisheye camera $\mathbf{C}$ is tracked. First, the intrinsic parameter matrix $\mathbf{K}$ of the lens $\mathbf{C}$ should be pre-computed using the camera calibration method [Zha00]. Then, the feature correspondence between adjacent frames is built to generate the essential matrix $\mathbf{E}$. Finally, the SfM algorithm [SZFP16] is leveraged to recover the extrinsic parameter matrix $\mathbf{M}$ and calculate the $\mathcal{L}$.

$$\mathbf{M} = [R \mid T] = \mathbb{D}(\mathbf{E}, \mathbf{K}) \tag{21}$$

where $R \in \mathbb{R}^{3\times3}$ and $T \in \mathbb{R}^{3\times1}$ indicate the rotation matrix and the translation matrix respectively. $\mathbb{D}(\cdot)$ is the decomposition function of the essential matrix. The movement distance $\mathcal{L}$ is directly related to T and is equal to the Frobenius norm of T: $\mathcal{L} = \|T\|_F$.

In this way, accumulative errors are bounded by the number of frames that share the same set of matching points. This not only improves the video stitching robustness, but also speeds up the processing. The stitched panorama per frame can be mapped to a spherical surface and then visualized in the 3D space. As shown in Figure 11, compared to the static scene reconstruction using just a single fisheye image stitching result, instant video stitching supports the displaying of a dynamic real-life scene.

5. Evaluations

This section documents results of our comprehensive evaluations for verifying the effectiveness and applicability of the proposed approach. Fisheye images captured in various real-life environments were used. The high image alignment quality, robustness, adaptability and computational efficiency of our GPU implementation are demonstrated with a comparison between our approach and existing state-of-the-art methods.

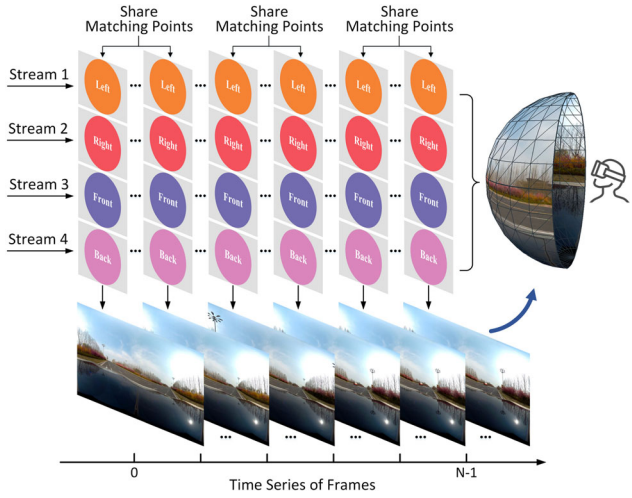


Figure 11: *Interactive panoramic viewing by stitching quad-fisheye video frames in real time. Matching points calculated on one set of quad-fisheye video frames for stitching are shared in a few following frames.*

5.1. Experiments

Current applications show more and more interests on realistic and interactive panoramic environment view reconstruction and rendering in real time [LKK*16]. It requires not only the high quality of fisheye image stitching results, but also fast computing to provide sufficient interactivity with the high FPS [IR15]. This section discusses our experiments and verifications conducted to assess the performance of the proposed method using fisheye images of real-life scenes. Results of our approach are compared with some mesh-based methods, i.e. APAP [ZCT*14], SPHP [CSC14], AANAP [LPRA15], GSP [CC16] and Rich360 [LKK*16], as well as several non-mesh-based methods, i.e. AutoStitch [BL07] and OmniMVS [WRL20]. Since these existing methods cannot directly take the original fisheye images as the input, fisheye images are unwarped before applying the stitching algorithm as in our pipeline.

Our experiments were performed using a PC with the moderate computing capability of a NVIDIA GeForce GTX1060 GPU with 3GB DDR3, an Intel quad-core i7-6700 CPU and 16GB system memory. Proposed algorithms were programmed in C++ and using CUDA ver. 9.2, with a UI developed using Qt. Various fisheye images were captured for verifications using the Detu F4 camera, which is a commercial HD panoramic photo camera. This camera has four wide-angle fisheye lenses. Each lens can capture 4K video at 30FPS. To balance the computational efficiency with the image alignment quality, the resolution of the unwarped fisheye image was adjusted to 1000×1000. The dimension of deformable meshes for image alignment is set to a 10×10 grid in our experiments.

5.2. Stitching quality assessment

To evaluate the stitching quality of our approach, we compared results of our algorithm and existing methods, in terms of both visual inspection and quantitative assessment. In addition, we conducted a user study to evaluate how our method improves user's subjective perception.

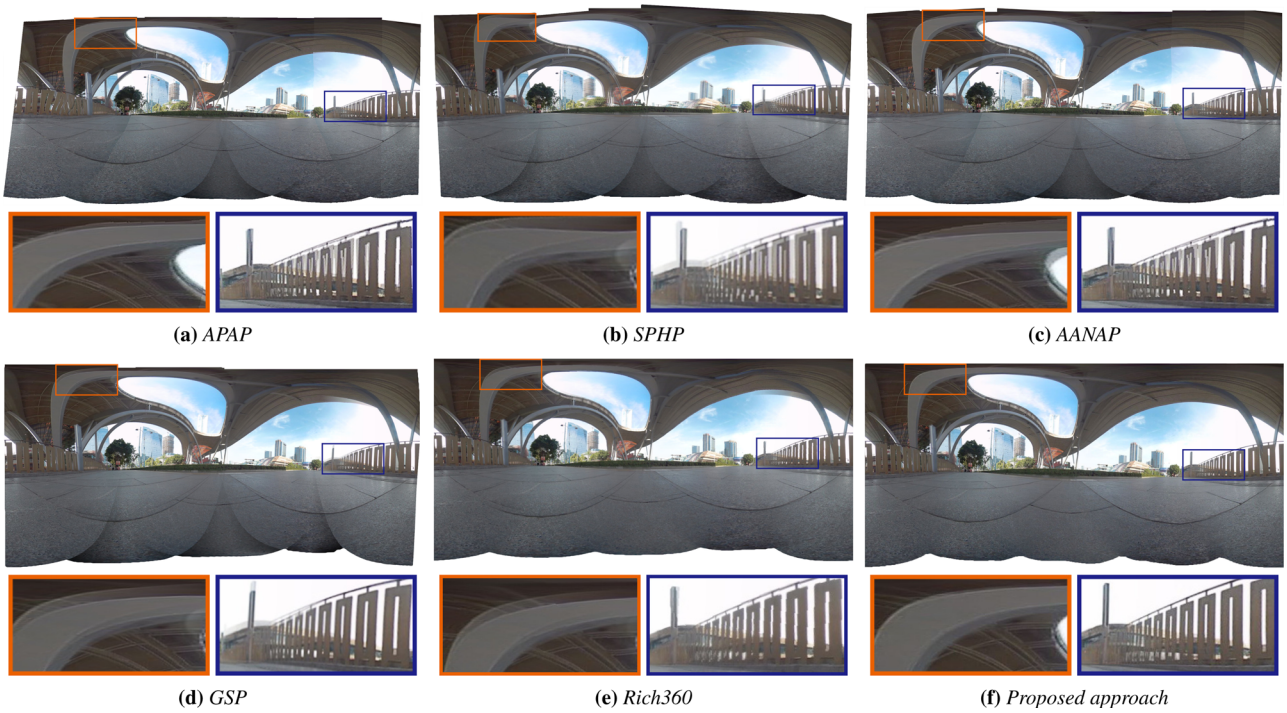
5.2.1. *Visual inspection of stitching results*

A visual comparison between results of our approach and existing mesh-based methods is shown in Figure 12. Due to the simple extrapolation of the projection transformation to non-overlapping regions, the result of APAP [ZCT*14] shows projection distortions in the non-overlapping regions (Figure 12a). SPHP [CSC14] pays more attention to mitigating the perspective distortion but not the alignment accuracy. The shape structure is preserved but parallax errors still exist in the result of SPHP (Figure 12b). There are visible local distortions in the result of AANAP [LPRA15]. Unnatural rotation artefacts are still difficult to eliminate (Figure 12c). GSP [CC16] uses the prior global similarity and the local warp model, which help reduce the shape distortion and provide a higher alignment accuracy. However irregular boundaries and ghosting artefacts are not completely removed (Figure 12d). Note that we reproduced Rich360 [LKK*16] because its original codes or executable files are not publicly available. Compared to other methods, Rich360 generates regular boundaries. Due to the low DoF of mesh grids, top and bottom regions of the panorama stitched using Rich360 still have serious artefacts.

We further compared our approach with several non-mesh-based methods, as shown in Figure 13. To verify whether the proposed method can be applied to other quad-fisheye cameras, fisheye images were collected from a public dataset [WRL20]. These testing images were captured using a camera that consists of four 220° FOV fisheye lenses. AutoStitch [BL07] uses the global homography model for the alignment. Structures are stretched in the horizontal direction and ghosting artefacts are visible (Figure 13a). OmniMVS [WRL20] leverages a CNN model to learn the global and local context information of fisheye images. But OmniMVS is hard to handle large inter-camera parallaxes and leads to serious artefacts in the stitched panoramic image (Figure 13b). These drawbacks are better solved in our method (Figures 12f and 13c). It demonstrates that our camera-independent method gives the better visual quality with fewer distortions and artefacts, as well as a better boundary regularity.

To visually evaluate the adaptability and robustness, our approach was tested on quad-fisheye images of real-life scenes in different circumstances, e.g. various illuminations, environment contexts, complexities, visual field broadnesses and depths, as shown in Figure 14. Quad-fisheye images captured in wide open outdoors (Figure 14a), enclosed indoors of large space (Figure 14b), narrow rooms (Figure 14c) with more complex arrangements, and outdoors with darker illumination (Figure 14d) were adopted in our evaluations. It shows that our approach is capable to handle all these testing cases and give desired stitching results.

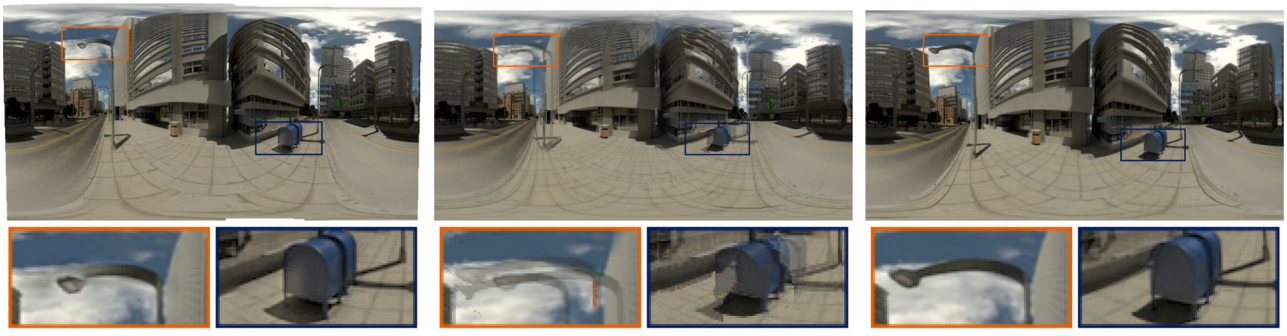
Figure 15 illustrates the effect of the inter-lens parallax optimization (Section 3.2). A classical 3D model, i.e. Lucy, was placed in front of a quad-fisheye camera. Fisheye images were captured by adjacent lenses. Since the model was quite close to the camera, different lenses captured images with large parallaxes. Right after the weighted blending, the stitched image clearly showed transparent



(a) *APAP* **(b)** *SPHP* **(c)** *AANAP*

(d) *GSP* **(e)** *Rich360* **(f)** *Proposed approach*

Figure 12: *Visual comparison of stitching results between our algorithm and existing mesh-based approaches. Details of local alignment quality are highlighted in the lower two images per result.*



(a) *AutoStitch* **(b)** *OmniMVS* **(c)** *Proposed approach*

Figure 13: *Visual comparison of stitching results between our algorithm and existing non-mesh-based approaches. Details of local alignment quality are highlighted in the lower two images per result.*

(a) *The outdoor space* **(b)** *The large indoor space* **(c)** *The narrow indoor space* **(d)** *The night outdoor space*

Figure 14: *The proposed quad-fisheye image stitching approach was tested in different real-world scenes.*

Figure 15: *Comparison results obtained without (left) and with (right) the inter-lens parallax optimization.*

artefacts, which can be effectively eliminated using the inter-lens parallax optimization.

5.2.2. *Quantitative evaluation of stitching quality*

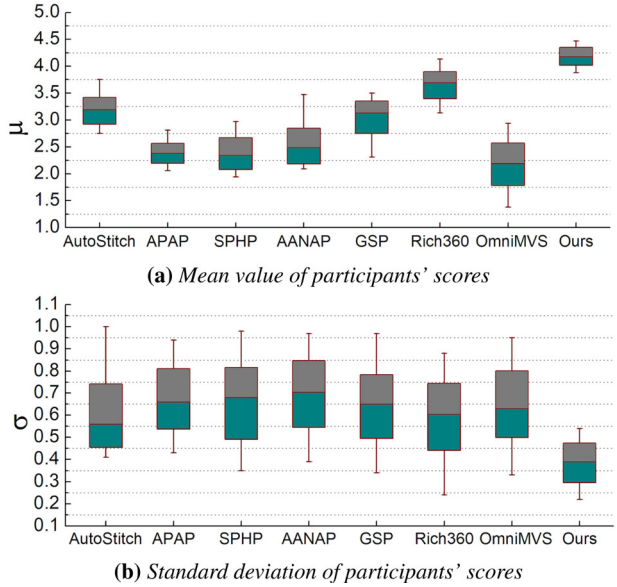
Three evaluators, i.e. *Natural Image Quality Evaluator* (NIQE) [MSB13], *Blind/Referenceless Image Spatial QUality Evaluator* (BRISQUE) [MMB12] and *Average Gradient* (AG) [PR17], were employed to quantitatively assess the resulting image quality of the proposed method in our evaluation. NIQE and BRISQUE are blind image quality assessment models to evaluate the image quality without knowledge of anticipated distortions or subjective opinions of humans. The smaller NIQE or BRISQUE is the better visual quality of the stitched panoramic image we get and vice versa. AG can be used to estimate whether the stitching result is free of ghosting artefacts. A larger AG reflects a higher image alignment accuracy and vice versa.

In our quantitative assessment of the result quality, our method was compared with other existing methods in Figures 12 and 13. Repeated experiments were conducted on fisheye images captured in eight different environments for 20 times and mean values of statistics of all image pixels are listed as in Table 1. It demonstrates that the proposed approach outperforms all other existing methods with all testing scenes.

5.2.3. *User study*

Due to the significant difference between the visual perception and the image-based quality (e.g. foveation, ventral metamers, spatio-temporal retinal sampling, etc.), a formal subjective user study was conducted to evaluate the stitching quality, in terms of the artefact existence, robustness and general visual perception. Thirty participants aging from 20 to 50 years old were invited in this user study. Besides, participants were engaged in technical or non-technical occupations, with or without image stitching experiences.

In our comparison, we selected eight real-world scenes and eight algorithms as outlined in Table 1. First, all resulting images were randomly ordered for each participant to avoid cueing him or her. After watching each reconstructed panorama for 30 s, the participant was then asked to give a score from 1 to 5 (from the worst to the best). Every participant repeated this experiment three times. In addition, we designed several evaluation criteria, e.g. alignment accuracy, boundary regularity and blending consistency, to control the individual bias of each participant. All tested algorithms were able to process selected real-world scenes and all participants evaluated all resulting images. In such a way, the conclusion of this user study can be mainly affected by the subjective opinions of these participants. Original scores made by all participants are given in the supplementary material. Figure 16 shows statistics of the mean value μ and standard deviation σ of scores given by the 30 participants. The proposed method obtained the average score of $\mu = 4.18$ and $\sigma = 0.38$ and outperformed other methods. This user study demonstrates that our method is favoured by most involved participants.



(a) *Mean value of participants' scores*

(b) *Standard deviation of participants' scores*

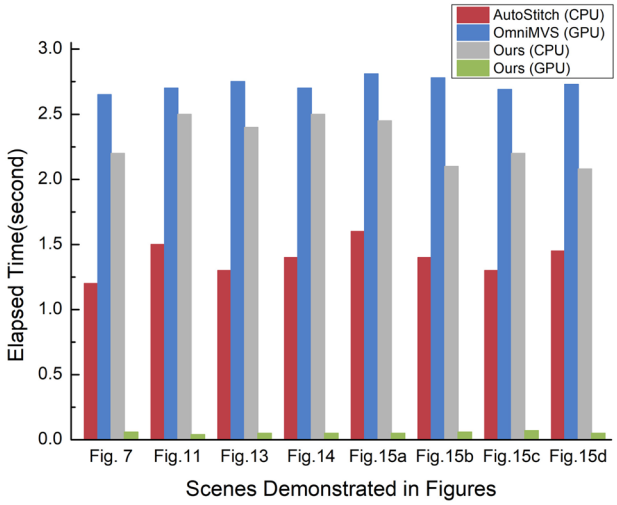
Figure 16: *User study. Participants were invited to evaluate the stitching quality between our approach and existing methods.*

5.3. Evaluation of computational efficiency

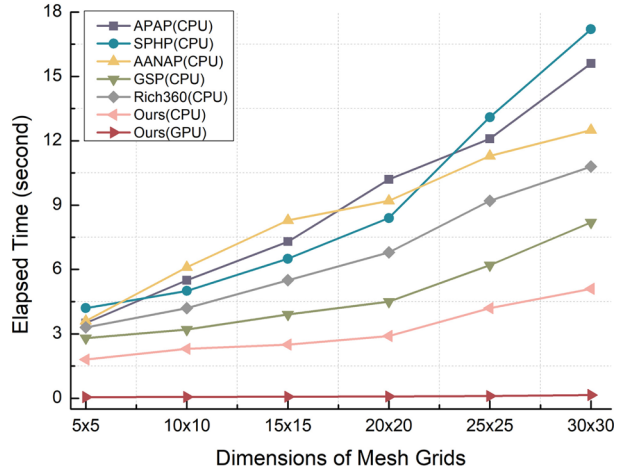
The speed of environment scene reconstruction is crucial for interactive real-time applications. The computational efficiency of our

Table 1: *Quantitative evaluation of stitching quality (average of all pixels).*

Scene in Fig#		AutoStitch [BL07]	APAP [ZCT*14]	SPHP [CSC14]	AANAP [LPRA15]	GSP [CC16]	Rich360 [LKK*16]	OmniMVS [WRL20]	**Ours**
Figure 6	NIQE (↓)	3.29	3.39	3.58	3.83	3.16	2.65	5.22	**2.53**
	BRISQUE (↓)	16.14	27.26	35.66	27.64	24.73	14.10	38.67	**12.85**
	AG (↑)	6.83	6.56	6.24	6.53	6.10	6.95	5.89	**6.97**
Figure 10	NIQE (↓)	3.52	4.32	4.23	4.20	3.92	3.01	5.31	**2.94**
	BRISQUE (↓)	29.25	34.01	59.67	34.05	28.70	13.96	40.58	**11.14**
	AG (↑)	4.93	4.54	4.21	4.55	4.92	5.13	4.77	**5.32**
Figure 12	NIQE (↓)	2.78	2.91	3.43	2.88	2.85	2.57	4.69	**2.55**
	BRISQUE (↓)	23.50	29.92	45.87	29.17	35.83	17.85	42.33	**15.08**
	AG (↑)	6.53	6.42	6.09	6.52	6.67	6.88	5.91	**6.95**
Figure 13	NIQE (↓)	3.01	4.16	4.88	4.23	3.58	2.97	6.44	**2.76**
	BRISQUE (↓)	19.19	32.21	37.74	35.65	27.88	13.49	38.81	**8.93**
	AG (↑)	7.02	5.34	5.26	5.32	7.55	7.68	4.91	**7.81**
Figure 14a	NIQE (↓)	4.89	5.15	5.22	4.59	4.36	2.35	4.96	**2.21**
	BRISQUE (↓)	48.34	42.26	58.45	47.91	30.89	7.68	40.97	**7.38**
	AG (↑)	5.92	4.99	4.32	4.92	5.33	7.42	4.36	**7.44**
Figure 14b	NIQE (↓)	3.36	3.79	3.74	3.45	3.52	3.26	5.83	**3.18**
	BRISQUE (↓)	13.08	27.92	45.01	25.14	16.27	13.01	30.87	**12.89**
	AG (↑)	7.04	6.34	6.21	6.52	6.68	6.99	5.74	**7.13**
Figure 14c	NIQE (↓)	2.89	3.12	3.47	2.89	2.68	2.30	4.66	**2.20**
	BRISQUE (↓)	12.78	21.01	25.86	17.57	17.31	8.74	20.64	**8.65**
	AG (↑)	7.66	6.12	7.22	7.47	6.96	7.55	4.43	**7.71**
Figure 14d	NIQE (↓)	5.31	4.78	5.04	4.82	3.73	2.62	6.25	**2.59**
	BRISQUE (↓)	30.08	23.90	22.59	22.74	20.62	11.76	29.78	**10.83**
	AG (↑)	6.05	4.39	4.91	4.86	5.37	7.65	4.11	**7.78**



(a) *Compare our method with a non-mesh-based method*



(b) *Compare our method with mesh-based methods*

Figure 17: *Comparison of elapsed time measures between our quad-fisheye image stitching approach and other state-of-the-art image stitching methods.*

approach was compared with other publicly available methods. Our comparison was still based on the same computer hardware and software configuration as described in Section 5.1. The Detu F4 camera was used as the only device to capture fisheye images.

Figure 17a illustrates measures of elapsed time for comparing our method with two non-mesh-based methods, i.e. AutoStitch and OmniMVS. Figure 17b compares our method with existing mesh-based methods, i.e. APAP, SPHP, AANAP, GSP and Rich360. Since AutoStitch uses the global warp model of much lower computational complexity, our method is slower than AutoStitch when the algorithm runs on the CPU. To pursue higher accuracy, OmniMVS consists of a complex network architecture. It inevitably leads to low computational efficiency. The execution speed of our program was highly boosted with a GPU implementation. In the comparison between our method and other existing mesh-based methods, only GSP, Rich360 and our approach were implemented in C++. The programs of other methods obtained from their official internet resources were implemented in MATLAB. Instead of comparing the absolute performance regardless of the implementation platform, we tried to evaluate impacts on these approaches and their implementations due to the increase of data scales and fitting mesh complexities. In our experiments, we set different mesh sizes to compare the speed regression of all methods. The measured elapsed time included operations of feature point detection, mesh deformation, texture mapping and blending. It is demonstrated that stitching speeds of all methods drop as the fitting mesh size increases and our approach has less speed regressions than others. If programs run on CPU, our program is a little faster than GSP and much faster than implementations of other approaches. If implemented on GPU using CUDA, our method can be remarkably accelerated by the GPU parallel processing power. Our execution performance on GPU is very stable almost without regression as the mesh complexity increases.

The GPU-accelerated implementation greatly facilitates achieving real-time panoramic image stitching and the display of 360° real-world scenes. We evaluated our stitching algorithm and CUDA implementation for synchronous panoramic environment reconstruction by stitching the pre-recorded quad-fisheye video clip, as described in Section 4.2. We performed a repeated experiment of 20 times for stitching a fisheye video clip of the resolution 1000×1000 per unwarped fisheye image frame. Our end-to-end system, e.g. from capture, reconstruction to display, supports an average 30.2FPS on a NVIDIA GTX1060 GPU with 3GB of video RAM. On a NVIDIA GTX1080ti GPU with 8GB of video RAM, it can achieve an average 39.7FPS.

6. Conclusions and Future Work

In this paper, a novel monoscopic panorama reconstruction pipeline based on quad-fisheye image stitching is proposed. We developed a new camera-independent quad-fisheye image stitching approach. It applies a mesh-based image alignment to stitch quad-fisheye images. To optimize inter-lens parallaxes, unwarped fisheye images are cropped and reshuffled before stitching. To better constrain the mesh deformation for the image alignment and avoid overstretching artefacts, a new integrative energy function is applied. It has five constraint factors assigned with weighting coefficients. Compared with existing methods, our approach offers a better stitching quality for non-planar views. It provides the higher image alignment accuracy and the higher robustness for real-life scenes in various circumstances. In addition, the entire pipeline of our method is implemented on GPU using CUDA to achieve higher FPS and interactivity. To further improve the computational efficiency of the video stitching, pre-computed mesh parameters are shared with a few video frames. Accumulative errors of mesh registration are controlled by taking the camera motion and average gradient per video frame into account. The panoramic viewing can be visualized synchronously along with the real-life environment using our extended panorama video stitching method.

There are plenty of promising research avenues in our future work towards an interactive and complete real-life scene visualization technique, including the realistic rendering of 3D virtual objects. We

will expand our panoramic scene reconstruction pipeline and integrate it with more techniques to improve the immersive interaction between virtual entities and surroundings in reality. In our current work, the proposed approach can only generate the 3-DoF viewing. We will leverage the monoscopic panorama to predict the omnidirectional depth and then develop a 6-DoF VR technique [SKC*19]. In addition, our approach will also be integrated into an interactive VR system for the further experiments and improvements.

Acknowledgements

This work was supported in part by the China National Key R&D Plan under Grant 2020YFC2003900, and the Provincial Key Technology R&D Program under Grant BE2019710 of Jiangsu, China.

References

[BL07] Brown M., Lowe D. G.: Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision 74*, 1 (Aug. 2007), 59–73.

[CC16] Chen Y.-S., Chuang Y.-Y.: Natural image stitching with the global similarity prior. In *Proceedings of the European Conference on Computer Vision* (Amsterdam, Sep. 2016), pp. 186–201.

[CDSHD13] Chaurasia G., Duchene S., Sorkine-Hornung O., Drettakis G.: Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics 32*, 3 (June 2013), 1–12.

[CSC14] Chang C.-H., Sato Y., Chuang Y.-Y.: Shape-preserving half-projective warps for image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Columbus, June 2014), pp. 3254–3261.

[DMR16] DeTone D., Malisiewicz T., Rabinovich A.: Deep image homography estimation. *arXiv preprint arXiv:1606.03798* (2016).

[dSJ19] da Silveira T. L. T., Jung C. R.: Dense 3D scene reconstruction from multiple spherical images for 3-DoF+ VR applications. In *Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces* (Osaka, Mar. 2019), pp. 9–18.

[EKD*17] Eilertsen G., Kronander J., Denes G., Mantiuk R. K., Unger J.: HDR image reconstruction from a single exposure using deep CNNs. *ACM Transactions on Graphics 36*, 6 (Dec. 2017), 1–15.

[GKB11] Gao J., Kim S. J., Brown M. S.: Constructing image panoramas using dual-homography warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Colorado Springs, June 2011), pp. 49–56.

[GRE*16] Gaddam V. R., Riegler M., Eg R., Griwodz C., Halvorsen P.: Tiling in interactive panoramic video: Approaches and evaluation. *IEEE Transactions on Multimedia 18*, 9 (June 2016), 1819–1831.

[HB17] Ho T., Budagavi M.: Dual-fisheye lens stitching for 360-degree imaging. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing* (New Orleans, Mar. 2017), pp. 2172–2176.

[HCR*17] Huang J., Chen Z., Research A., Ceylan D., Hailin U.: 6-DOF VR videos with a single 360-camera. In *Proceedings of the IEEE Virtual Reality* (Los Angeles, Mar. 2017), pp. 37–44.

[HCS13] He K., Chang H., Sun J.: Rectangling panoramic images via warping. *ACM Transactions on Graphics 32*, 4 (July 2013), 1–10.

[HSRB17] Ho T., Schizas I. D., Rao K. R., Budagavi M.: 360-degree video stitching for dual-fisheye lens cameras based on rigid moving least squares. In *Proceedings of the IEEE International Conference on Image Processing* (Beijing, Sep. 2017), pp. 51–55.

[HZ04] Hartley R., Zisserman A.: *Multiple View Geometry in Computer Vision* (2nd edition). Cambridge University UK. Press, 2004.

[II11] Igarashi T., Igarashi Y.: Implementing as-rigid-as-possible shape manipulation and surface flattening. *Journal of Graphics, GPU, and Game Tools 14*, 1 (Jan. 2011), 17–30.

[IR15] Iorns T., Rhee T.: Real-time image based lighting for 360-degree panoramic video. In *Proceedings of the PSIVT 2015 Workshops on Image and Video Technology* (Auckland, Nov. 2015), vol. *9555*, pp. 139–151.

[JKOK15] Joo K., Kim N., Oh T.-H., Kweon I. S.: Line meets as-projective-as-possible image stitching with moving DLT. In *Proceedings of the IEEE International Conference on Image Processing* (Quebec, Sep. 2015), pp. 1175–1179.

[LGG*19] Lai W., Gallo O., Gu J., Sun D., Yang M., Kautz J.: Video stitching for linear camera arrays. In *Proceedings of the British Machine Vision Conference* (Cardiff, Sep. 2019), pp. 1–12.

[LKK*16] Lee J., Kim B., Kim K., Kim Y., Noh J.: Rich360: Optimized spherical representation from structured panoramic camera arrays. *ACM Transactions on Graphics 35*, 4 (July 2016), 1–11.

[LL19] Liao T., Li N.: Single-perspective warps in natural image stitching. *IEEE Transactions on Image Processing 29* (Aug. 2019), 724–735.

[LLM*11] Lin W.-Y., Liu S., Matsushita Y., Ng T.-T., Cheong L.-F.: Smoothly varying affine stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Colorado Springs, June 2011), pp. 345–352.

[LPRA15] Lin C.-C., Pankanti S. U., Ramamurthy K. N., Aravkin A. Y.: Adaptive as-natural-as-possible image stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, June 2015), pp. 1155–1163.

[LS20] Lee K.-Y., Sim J.-Y.: Warping residual based image stitching for large parallax. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (Seattle, June 2020), pp. 8195–8203.

[MCE*17] Matzen K., Cohen M. F., Evans B., Kopf J., Szeliski R.: Low-cost 360 stereo photography and video capture. *ACM Transactions on Graphics 36*, 4 (July 2017), 1–12.

[MMB12] Mittal A., Moorthy A. K., Bovik A. C.: No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing 21*, 5 (Dec. 2012), 4695–4708.

[MSB13] Mittal A., Soundararajan R., Bovik A. C.: Making a "Completely Blind" image quality analyzer. *IEEE Signal Processing Letters 20*, 3 (Mar. 2013), 209–212.

[NLL*20] Nie L., Lin C., Liao K., Liu M., Zhao Y.: A view-free image stitching network based on global homography. *Journal of Visual Communication and Image Representation 73* (Nov. 2020), 102950.

[PR17] Plötz T., Roth S.: Automatic registration of images to untextured geometry using average shading gradients. *International Journal of Computer Vision 125* (June 2017), 65–81.

[PSZ*15] Perazzi F., Sorkinehornung A., Zimmer H., Kaufmann P., Wang O., Watson S., Gross M.: Panoramic video from unstructured camera arrays. *Computer Graphics Forum 34*, 2 (June 2015), 57–68.

[RPZSH13] Richardt C., Pritch Y., Zimmer H., Sorkine-Hornung A.: Megastereo: Constructing high-resolution stereo panoramas. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Portland, June 2013), pp. 1256–1263.

[RRKB11] Rublee E., Rabaud V., Konolige K., Bradski G.: ORB: An efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision* (Barcelona, Nov. 2011), pp. 2564–2571.

[SBSH18] Schroers C., Bazin J.-C., Sorkine-Hornung A.: An omnistereoscopic video pipeline for capture and display of real-world VR. *ACM Transactions on Graphics 37*, 3 (Aug. 2018), 1–13.

[SKC*19] Serrano A., Kim I., Chen Z., DiVerdi S., Gutierrez D., Hertzmann A., Masia B.: Motion parallax for 360° RGBD video. *IEEE Transactions on Visualization and Computer Graphics 25*, 5 (Mar. 2019), 1817–1827.

[SMW06] Schaefer S., McPhail T., Warren J.: Image deformation using moving least squares. *ACM Transactions on Graphics 25*, 3 (July 2006), 533–540.

[Sze06] Szeliski R.: Image Alignment And Stitching: A Tutorial. *Publication: Foundations and Trends®in Computer Graphics and Vision 2*, 1. Jan. 2006, pp. 1–104.

[SZFP16] Schönberger J. L., Zheng E., Frahm J.-M., Pollefeys M.: Pixelwise view selection for unstructured multi-view stereo. In *Proceedings of the European Conference on Computer Vision* (Amsterdam, Sep. 2016), pp. 501–518.

[TBLG16] Thatte J., Boin J.-B., Lakshman H., Girod B.: Depth augmented stereo panorama for cinematic virtual reality with head-motion parallax. In *Proceedings of the IEEE International Conference on Multimedia and Expo* (Seattle, Aug. 2016), pp. 1–6.

[WRHS13] Weinzaepfel P., Revaud J., Harchaoui Z., Schmid C.: DeepFlow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision* (Sydney, Dec. 2013), pp. 1385–1392.

[WRL20] Won C., Ryu J., Lim J.: End-to-end learning for omnidirectional stereo matching with uncertainty prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence 43*, 11 (May 2020), 3850–3862.

[YHL*16] Yan W., Hou C., Lei J., Fang Y., Gu Z., Ling N.: Stereoscopic image stitching based on a hybrid warping model. *IEEE Transactions on Circuits and Systems for Video Technology 27*, 9 (May 2016), 1934–1946.

[ZCT*14] Zaragoza J., Chin T.-J., Tran Q.-H., Brown M. S., Suter D.: As-projective-as-possible image stitching with moving DLT. *IEEE Transactions on Pattern Analysis and Machine Intelligence 36*, 7 (July 2014), 1285–1298.

[Zha00] Zhang Z.: A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 11 (Nov. 2000), 1330–1334.

[ZKZD18] Zioulis N., Karakottas A., Zarpalas D., Daras P.: OmniDepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision* (Munich, July 2018), pp. 448–465.

[ZLZ20] Zhang Y., Lai Y., Zhang F.: Content-preserving image stitching with piecewise rectangular boundary constraints. *IEEE Transactions on Visualization and Computer Graphics PP*, *99* (Jan. 2020), 1.