

## Article

# SC-AOF: A Sliding Camera and Asymmetric Optical-Flow-Based Blending Method for Image Stitching

Jiayi Chang<sup>1,2,3,\*</sup>, Qing Li<sup>2</sup>, Yanju Liang<sup>3</sup> and Ligu Zhou<sup>4,\*</sup><sup>1</sup> Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100049, China<sup>2</sup> University of Chinese Academy of Sciences, Beijing 101408, China<sup>3</sup> Wuxi Iot Innovation Center Co., Ltd., Wuxi 214135, China; liangyanju@wiot.tech<sup>4</sup> School of Computation, Information and Technology, Technical University of Munich, 85748 Garching bei München, Germany

\* Correspondence: changjiayi@ime.ac.cn (J.C.); liguo.zhou@tum.de (L.Z.)

**Abstract:** Parallax processing and structure preservation have long been important and challenging tasks in image stitching. In this paper, an image stitching method based on sliding camera to eliminate perspective deformation and asymmetric optical flow to solve parallax is proposed. By maintaining the viewpoint of two input images in the mosaic non-overlapping area and creating a virtual camera by interpolation in the overlapping area, the viewpoint is gradually transformed from one to another so as to complete the smooth transition of the two image viewpoints and reduce perspective deformation. Two coarsely aligned warped images are generated with the help of a global projection plane. After that, the optical flow propagation and gradient descent method are used to quickly calculate the bidirectional asymmetric optical flow between the two warped images, and the optical-flow-based method is used to further align the two warped images to reduce parallax. In the image blending, the softmax function and registration error are used to adjust the width of the blending area, further eliminating ghosting and reducing parallax. Finally, by comparing our method with APAP, AANAP, SPHP, SPW, TFT, and REW, it has been proven that our method can not only effectively solve perspective deformation, but also gives more natural transitions between images. At the same time, our method can robustly reduce local misalignment in various scenarios, with higher structural similarity index. A scoring method combining subjective and objective evaluations of perspective deformation, local alignment and runtime is defined and used to rate all methods, where our method ranks first.



**Citation:** Chang, J.; Li, Q.; Liang, Y.; Zhou, L. SC-AOF: A Sliding Camera and Asymmetric Optical-Flow-Based Blending Method for Image Stitching. *Sensors* **2024**, *24*, 4035. <https://doi.org/10.3390/s24134035>

Academic Editor: Yun Zhang

Received: 29 April 2024

Revised: 16 May 2024

Accepted: 19 June 2024

Published: 21 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** image stitching; sliding cameras; asymmetric optical flow; image blending

## 1. Introduction

Image stitching is a technology that can align and blend multiple images to generate a high-resolution, wide field-of-view and artifact-free mosaic. It has broad and promising applications in many fields such as virtual reality, remote sensing mapping, and urban modeling. The calculation of the global homography, as an important step in image stitching [1,2], directly determines the image alignment accuracy and the final user experience. However, global homography only works for planar scenes or rotation-only camera motions. For non-planar scenes or when the optical centers of cameras do not coincide, homography tends to cause misalignment, resulting in blurring and ghosting in the mosaic. It can also cause perspective deformation, making the final mosaic blurred and severely stretched at the edges. Many solutions have been proposed to solve the problems of parallax and perspective deformation in image stitching, so as to improve the quality of stitched images. But most state-of-the-art mesh-based [3–5] and multi-plane [6–8] methods are time-consuming and vulnerable to false matches.

In this work, an innovative image stitching method combining sliding camera (SC) and asymmetric optical flow (AOF), referred to as the SC-AOF method, is proposed to

reduce both perspective deformation and alignment error. In the non-overlapping area of the mosaic, the SC-AOF method manages to keep the viewpoint of the mosaic the same as or one rotation around the camera Z axis from those of the input images. In the overlapping area of the mosaic, the viewpoint is changed from one input image viewpoint to another, which can effectively solve the perspective deformation at the edge. A global projection plane is estimated to project input images onto the mosaic. After that, an asymmetric optical flow method is employed to further align the images. In the blending, the softmax function and alignment error are used to dynamically adjust the width of the blending area to further eliminate ghosting and improve the mosaic quality. This paper makes the following contributions:

- The SC-AOF method innovatively uses an approach based on sliding camera to reduce perspective deformation. Combined with either a global projection model or a local projection model, this method can effectively reduce the perspective deformation.
- An optical-flow-based image alignment and blending method is adopted to further mitigate misalignment and improve the stitching quality of the mosaic generated by a global projection model.
- Each step in the SC-AOF method can be combined with other methods to improve the stitching quality of those methods.

This article is organized as follows. Section 2 presents the related works. Section 3 first introduces the overall method of this article, then an edge stretching reduction method based on sliding camera and a local misalignment reduction method based on asymmetric optical flow are elaborated in detail. Section 4 presents our qualitative and quantitative experimental results compared with other methods. Finally, Section 5 summarizes our method.

## 2. Related Works

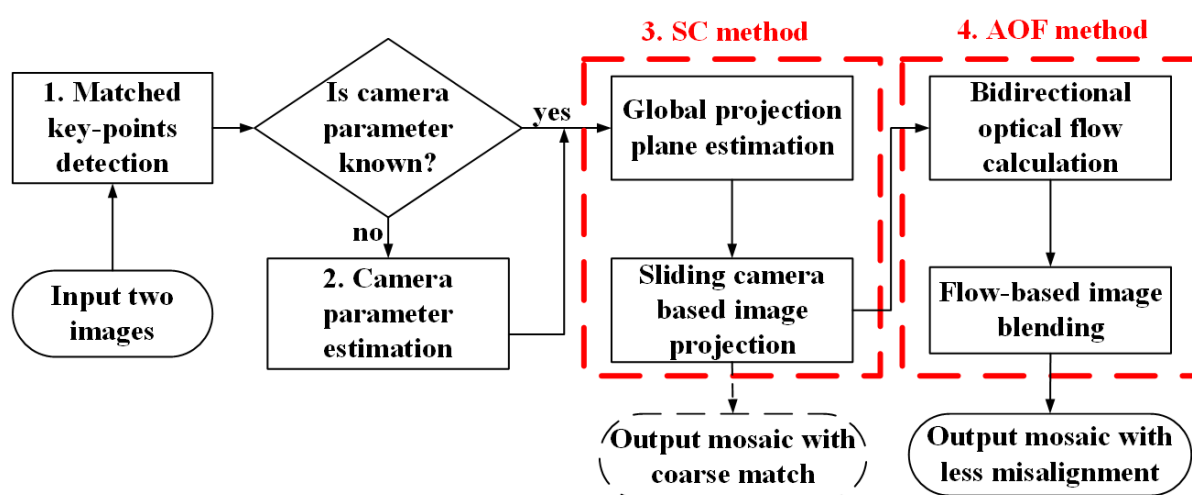
For the local alignment, APAP (as-projective-as-possible) [8,9] uses the weighted DLT (direct linear transform) method to estimate the location-dependent homography and then eliminate misalignment. However, if some key points match incorrectly, the image areas near these key points may have incorrect homography, resulting in serious alignment errors and distortion. APAP needs to estimate homography using DLT for each image cell, and therefore APAP runs much slower than the global homography warping. REW (robust elastic warping) [10,11] uses the TPS (thin-plate spline) interpolation method to convert discrete matched feature points into a deformation field, which is used to warp the image and achieve accurate local alignment. The estimation of TPS parameters and the deformation field is fast, so REW has excellent running efficiency. TFT (triangular facet approximation) [6] uses the Delaunay triangulation method and the matched feature points to triangulate the mosaic canvas, and the warping inside each triangle is determined by the homography calculated based on the three triangle vertices, so the false matches will lead to serious misalignment. TFT estimates a plane for every triangle instead of a homography for every cell, so TFT depends on the number of triangular facets in efficiency and runs faster than APAP generally. The warping-residual-based image stitching method [7] first estimates multiple homography matrices, and calculates warping residuals of each matched feature point using the multiple homography matrices. The homography of each region is estimated using moving DLT with the difference of the warping residuals as weight, which means the method can handle larger parallax than APAP, but is less robust to the incorrectly estimated homography and runs slower than APAP. The NIS (natural image stitching) [12] method estimates a pixel-to-pixel transformation based on feature matches and the depth map to achieve accurate local alignment. In [13], by increasing feature correspondences and optimizing hybrid terms, sufficient correct feature correspondences are obtained in the low-texture areas to eliminate misalignment. The two methods require additional runtime to enhance robustness, but also are susceptible to the uneven distribution and false matches of feature points.

For perspective deformation, SPHP (shape preserving half projective) [14,15] spatially combines perspective transformation and similarity transformation to reduce deformation. Perspective transformation can better align pixels in overlapping areas, and similarity transformation preserves the viewpoint of the original image in non-overlapping areas. AANAP (adaptive as-natural-as-possible) [16] derives the appropriate similarity transform directly based on matched feature points, and uses weights to gradually transit from perspective transform to similarity transform. The transitions from the homography of the overlapping area to the similarity matrix of the non-overlapping area adopted by SPHP and AANAP are artificial and unnatural, and can generate some “strange” homography matrices, causing significant distortion in the overlapping area. Both SPHP and AANAP require the estimation of homography or similarity matrices for each cell, and thus have the same efficiency issue as APAP. GSP (global similarity prior) [17,18] adds a global similarity prior to constrain the warping of each image so that it resembles a similarity transformation as a whole and avoids large perspective distortion. In SPW (single-projective warp) [19], the quasi-homography warp [20] is adopted to mitigate projective distortion and preserve the single perspective. SPSO (Structure Preservation and Seam Optimization) [4] uses a hybrid warping model based on multi-homography and mesh-based warp to obtain precise alignment of areas at different depths while preserving local and global image structures. GES-GSP (geometric structure preserving-global similarity prior) [21] employs deep learning-based edge detection to extract various types of large-scale edges, and further introduces large-scale geometric structure preservation to GSP to preserve the curves in images and protect them from distortion. GSP, SPW, SPSO and GES-GSP are based on content preserving warping and require constructing and solving a linear equation with  $m$  variables and  $n$  equations to acquire the corresponding coordinates after mesh warping, in which  $m$  is the number of cell vertices multiplied by 2,  $n$  is the number of alignment constraints, structural preservation constraints, and other constraints. Both  $m$  and  $n$  are generally larger, therefore more runtime is required.

Based on the above analysis, generating a natural mosaic quickly and robustly remains a challenging task.

### 3. Methodology

The flow chart of the SC-AOF algorithm is illustrated in Figure 1. The details on each of its steps are described below.



**Figure 1.** Flow chart of SC-AOF method. After the detection and matching of feature points, the camera parameters are obtained in advance or estimated. Then the two warped images are calculated using SC method, and the mosaic that is coarsely aligned can be obtained. Finally, the AOF method is used to further align the two warped images to generate a blended mosaic with higher alignment accuracy.

Step 1: Feature point detection and matching. SIFT (scale-invariant feature transform) and SURF (speed-up robust feature) methods are generally used to detect and describe key points from two input images. Using the KNN (k-nearest neighbor) method, a group of matched points is extracted from the key points and used for camera parameter estimation in step 2 and global projection plane calculation in step 3.

Step 2: Camera parameter estimation. The intrinsic and extrinsic camera parameters are the basis of the SC method, and can be obtained in advance or estimated. When camera parameters are known, we can skip step 1 and directly start from step 3. When camera parameters are unknown, they can be estimated by minimizing the epipolar and planar errors, as described in Section 3.3.

Step 3: Sliding camera-based image projection. In this step, we estimate the global projection plane first, then adjust the camera projection matrix and generate a virtual camera in the overlapping area by interpolation, and obtain the warped images by global planar projection, as detailed in Section 3.1. Misalignment can be found in the two warped images obtained in the current step. Therefore, we need to use the AOF method in step 4 to further improve the alignment accuracy.

Step 4: Flow-based image blending. In this step, we first calculate the bidirectional asymmetric optical flow between the two warped images, then further align and blend the two warped images to generate a mosaic using the optical flow (see Section 3.2 for more details).

### 3.1. SC: Viewpoint Preservation Based on Sliding Camera

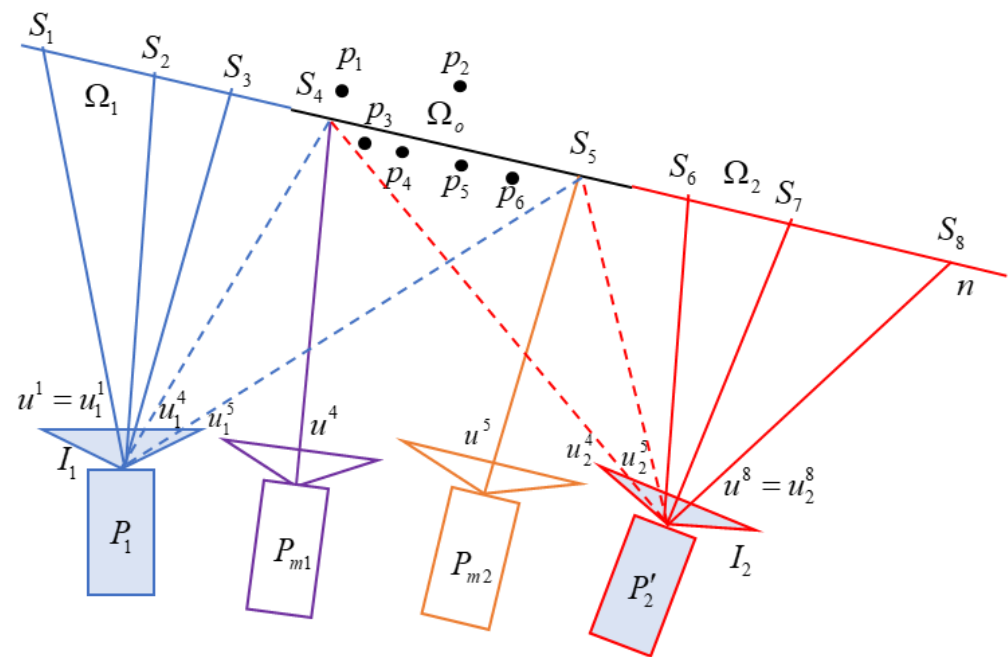
The sliding camera (SC) method is proposed for the first time to solve perspective deformation, and is the first step in the SC-AOF method. For this reason, this section will first introduce the stitching process of this method, and then detail how to calculate the global projection plane and the sliding projection matrix required by this method.

#### 3.1.1. SC Stitching Process

In order to ensure that the mosaic can maintain the perspective of the two input images, the SC method is used. That is, in the non-overlapping area, the viewpoints of the two input images are preserved. In the overlapping area, the viewpoint of the camera is gradually transformed from  $I_1$  to  $I_2$ .

As shown in Figure 2, the image  $I_1$  and  $I_2$  are back-projected onto the projection surface  $n$ , so that the corresponding non-overlapping areas  $\Omega_1, \Omega_2$  and overlapping area  $\Omega_o$  are obtained. Assume that the pixels in the mosaic  $I$  are  $u^1, u^2, \dots, u^8$ , which correspond to the sampling points  $S_1, S_2, \dots, S_8$  on the projection surface  $n$ . When the sampling points are within the projection area  $\Omega_1$  of image  $I_1$ , the mosaic is generated from the viewpoint of  $I_1$ .  $S_1, S_2, S_3$  are the intersection points of the backprojection lines of  $u^1, u^2, u^3$  in  $I_1$  and the projection surface  $n$ . Therefore,  $u^i = P_1 S_i (i = 1, 2, 3)$ , where  $P_1$  is the projection matrix of  $I_1$ . When the sampling points are within the projection area  $\Omega_2$  of image  $I_2$ , the mosaic is generated from the camera viewpoint of  $I_2$ . Similarly, we obtain  $S_i$  and  $u^i = P_2 S_i$ , where  $i = 6, 7, 8$ . In the overlapping area  $\Omega_o$  of  $I_1$  and  $I_2$ , the SC method is used to generate a virtual camera, whose viewpoint gradually transitions from the viewpoint of  $I_1$  to that of  $I_2$ .  $S_4$  and  $S_5$  are the intersection points of the back-projection lines of  $u^4, u^5$  in the visual camera and projection plane  $n$ , respectively. The virtual camera's image is generated from images  $I_1$  and  $I_2$  using perspective transformation. For example, pixel  $u^4$  of the virtual camera corresponds to pixel  $u_1^4$  in  $I_1$  and pixel  $u_2^4$  in  $I_2$ , and are generated by blending the latter two pixels.





**Figure 2.** Image stitching based on sliding cameras.  $n$  is the projection surface, which is fitted by scene points  $p_1, p_2, \dots, p_6$ . Stitched image  $I$  can be generated by projection of sampling points  $S_1, S_2, \dots, S_8$ . The points  $S_1, S_2, S_3$  in the area  $\Omega_1$  are generated by back-projection of pixels in  $I_1$ . Similarly, the points  $S_6, S_7, S_8$  in the area  $\Omega_2$  are generated by back-projection of pixels in  $I_2$ . The points  $S_4, S_5$  in the area  $\Omega_0$  are generated by back-projection of pixels in virtual cameras. The pixel values of  $S_4, S_5$  correspond to the fused pixel values of projection in  $I_1$  and  $I_2$ .  $P_1$  and  $P_2$  are the camera projection matrices of images  $I_1$  and  $I_2$ . To unify the pixel coordinates of  $I_1$  and  $I_2$ ,  $P_2$  is adjusted to  $P'_2$  using the method in Section 3.1.3.

**Global projection surface calculation.** In order to match the corresponding pixels  $u_1^4$  of  $I_1$  and  $u_2^4$  of  $I_2$ , the projection surface  $n$  needs to be as close as possible to the real scene point; we can use the moving plane method [7–9] or the triangulation method [6] to obtain a more accurate scene surface. Since the SC-AOF method will use the optical flow to further align the images, for the stitching speed and stability, only the global plane is calculated as the projection surface. Section 3.1.2 will calculate the optimal global projection surface using the matched points.

**Sliding camera generation.** Generally, since the pixel coordinates of  $I_1$  and  $I_2$  are not uniform, in the mosaic  $I$ , when  $I(\tilde{u}) = I_1(P_1S)$  in the non-overlapping area of  $I_1$ ,  $I(\tilde{u}) = I_2(P_2S)$  is false in the non-overlapping area of  $I_2$ , where  $S$  is the sampling point on the projection surface. It is necessary to adjust the projection matrix of  $I_2$  to  $P'_2$ , so that  $I(\tilde{u}) = I_2(P'_2S)$ . The red camera is shown in Figure 2. Section 3.1.3 will deduce the adjustment method of the camera projection matrix, and interpolate in the overlapping area to generate a sliding camera, and obtain the warped images of  $I_1$  and  $I_2$ .

### 3.1.2. Global Projection Surface Calculation

The projection matrices of cameras  $C_1$  and  $C_2$  corresponding to images  $I_1$  and  $I_2$  are:

$$P_1 = K_1[I_{3 \times 3}|0] \quad P_2 = K_2R[I_{3 \times 3}|-t] \quad (1)$$

where  $K_1$  and  $K_2$  are the intrinsic parameter matrices of  $C_1$  and  $C_2$  respectively;  $R$  is the inter-camera rotation matrix; and  $t$  is location of the optical center of  $C_2$  in the coordinate system  $C_1$ .

The relationship between the projection  $u_1$  in  $I_1$  and the projection  $u_2$  in  $I_2$  of a 3D point  $p$  on plane  $n$  is:

$$\tilde{u}_2 \sim K_2 R (I_{3 \times 3} + t n^T) K_1^{-1} \tilde{u}_1 = H \tilde{u}_1 \quad (2)$$

where  $\tilde{u}_1$  and  $\tilde{u}_2$  are the homogeneous coordinates of  $u_1$  and  $u_2$ , respectively. The intersection point  $p$  satisfies  $n^T p + 1 = 0$ .  $\sim$  means that  $\tilde{u}_2$  is parallel to  $H \tilde{u}_1$ .

If camera parameters  $K_1, K_2, R$  and  $t$  are known, then we can deduce the following Equation (3) from Equation (2)

$$n^T \tilde{y}_1 = - \frac{(R^T \tilde{y}_2 \times t)^T (R^T \tilde{y}_2 \times \tilde{y}_1)}{(R^T \tilde{y}_2 \times t)^T (R^T \tilde{y}_2 \times t)} = b \quad (3)$$

where  $\tilde{y}_1 = K_1^{-1} \tilde{u}_1$  and  $\tilde{y}_2 = K_2^{-1} \tilde{u}_2$  are the normalized coordinates of  $\tilde{u}_1$  and  $\tilde{u}_2$ , respectively.

We use Equation (3) of all matched points to construct an overdetermined equation and obtain the fitted global plane  $n$  by solving this equation. Since the optical-flow-based stitching method will be used to further align the images, the RANSAC method is not used here to calculate the plane with the most inliers. Instead, the global plane that fits all feature points as closely as possible is selected, misalignment caused by global plane projection will be better solved during optical flow blending.

### 3.1.3. Projection Matrix Adjustment and Sliding Camera Generation

To preserve the viewpoint in the non-blending area of  $I_2$ , it is only required to satisfy  $I(\tilde{u}) = I_2(N\tilde{u}) = I_2(\tilde{u}_2)$ , where  $\tilde{u}$  is the homogeneous coordinate of a pixel in the mosaic,  $\tilde{u}_2$  is the homogeneous coordinate of a pixel in  $I_2$ ,  $N$  is a similarity transformation between  $I_2$  and  $I$ , and can be obtained by fitting the matched feature points:

$$N_* = \min_S \sum_{j=1}^n \|N\tilde{u}_1^j - \tilde{u}_2^j\|^2 \quad (4)$$

where  $\tilde{u}_1^j$  and  $\tilde{u}_2^j$  are the homogeneous coordinates of pixels in  $I_1$  and  $I_2$  respectively.

Therefore, in the non-overlapping area of  $I_2$ ,  $\tilde{u} = N_*^{-1} \tilde{u}_2 = N_*^{-1} K_2 R_2 (S - t)$ , where  $S$  is the corresponding 3D point of  $\tilde{u}_2$  on plane  $n$ . So we get the projection matrix  $P'_2 = N_*^{-1} K_2 R_2$ .

By RQ decomposition, the internal parameter matrix  $K'_2$  and rotation  $R'_2$  are extracted from  $P'_2$ :

$$N_*^{-1} K_2 R_2 = K_2 (K_2^{-1} N_*^{-1} K_2) R_2 = K_2 (K_* R_*) R_2 = K'_2 R'_2 \quad (5)$$

where  $K'_2$  and  $R'_2$  are upper triangular matrix and rotation matrix respectively; and the third line of both matrices is  $(0 \ 0 \ 1)$ .

Compared with  $P_2$ ,  $P'_2$  has a different intrinsic parameter matrix, and its rotation matrix only differs by one rotation around Z axis, and its optical center  $t$  is not changed.

$$t_m = (1 - m) 0_{3 \times 1} + m t \quad (6)$$

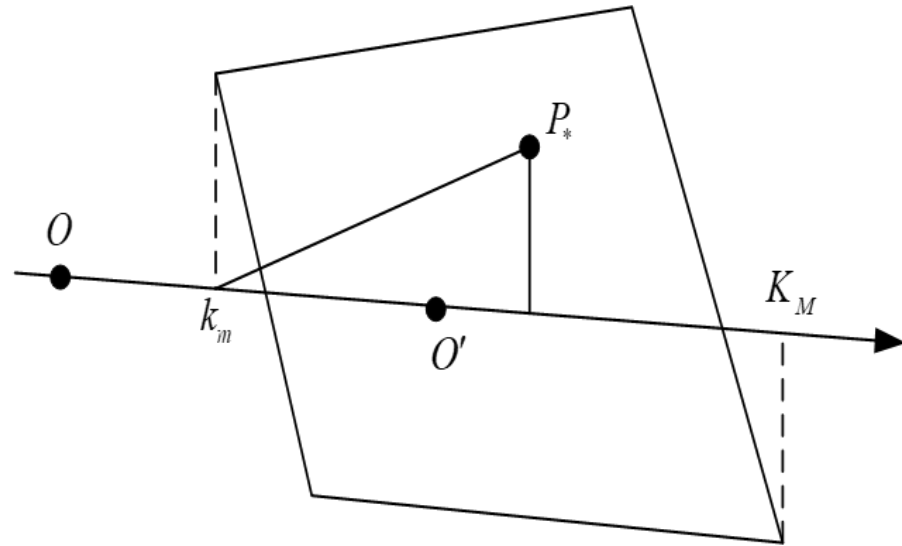
$$K_m = (1 - m) K_1 + m K'_2 \quad (7)$$

$$q_m = \frac{\sin((1 - m)\theta)}{\sin(\theta)} q_1 + \frac{\sin(m\theta)}{\sin(\theta)} q_2 \quad (8)$$

where  $q_1, q_2, q_m$  represent the quaternions corresponding to  $I_{3 \times 3}, R'_2$  and  $R_m$ ,  $\theta$  is the angle between  $q_1$  and  $q_2$ , and  $m$  is the weighting coefficient.

As depicted by Figure 3, the weighting coefficient  $m$  can be calculated by the method in AANAP [16]:

$$m = \left\langle \vec{k}_m P_*, \vec{k}_m K_M \right\rangle / |\vec{k}_m K_M|^2 \quad (9)$$



**Figure 3.** The diagram of gradient weight. The quadrilateral is the boundary of the overlapping area of  $I_1$  and mapped image of  $I_2$  using  $H^{-1}$ , where  $O$  is the center of  $I_1$  and  $O'$  is the warped point of the center point of  $I_2$  using  $H^{-1}$ .  $k_m$  and  $K_M$  are the projection points closest to  $O$  and  $O'$  on the line  $OO'$  of the quadrilateral vertices, respectively.  $P_*$  indicates the pixel coordinates within the overlapping area that need to calculate weighted parameter  $m$ .

In the overlapping area, if  $u$  corresponds to sliding camera  $(K_m, R_m, t_m)$ , then the relation between  $u$  and  $u_i$  in  $I_i$  ( $i = 1, 2$ ) can be expressed as:

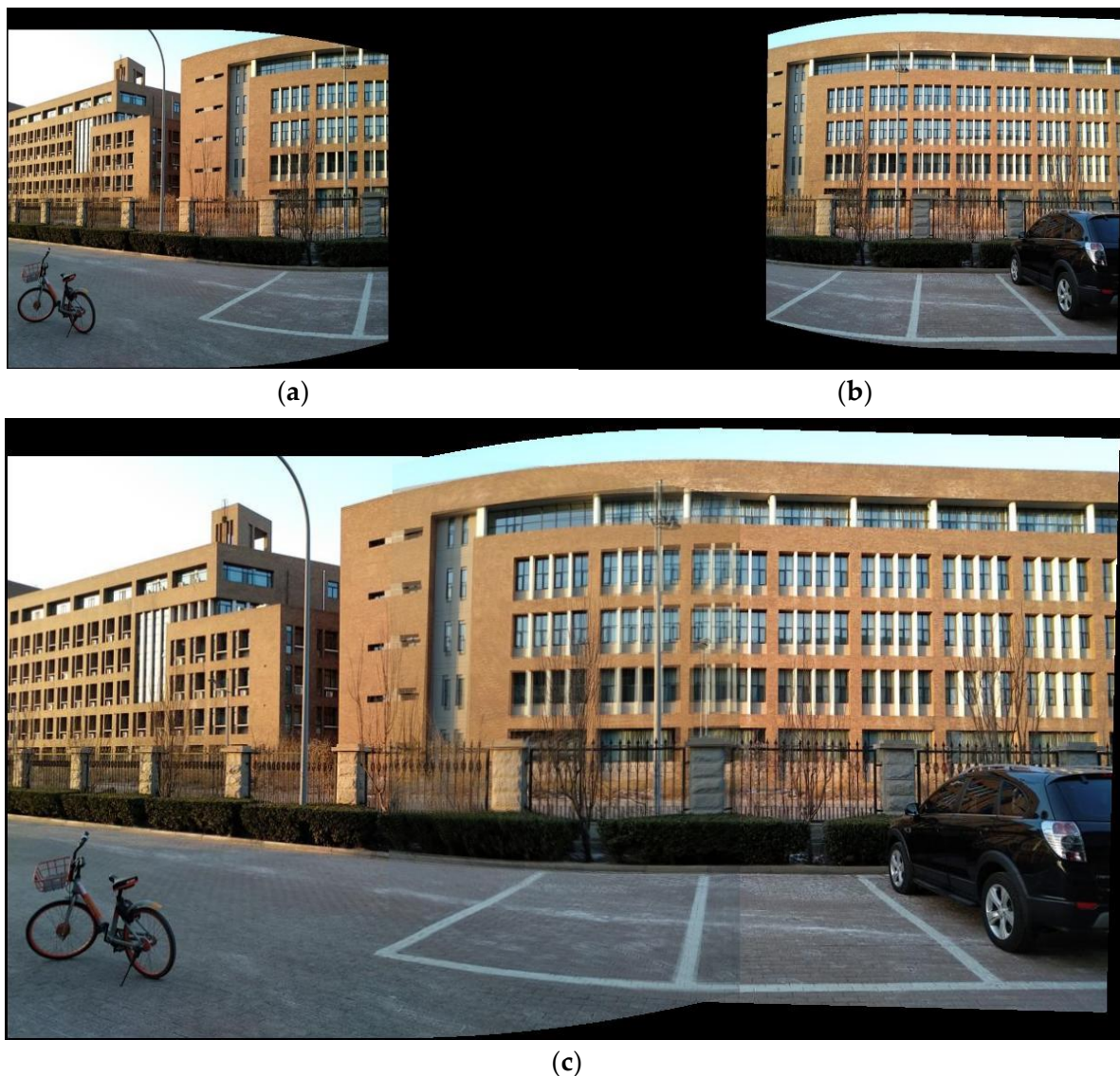
$$\tilde{u} = K_m R_m (I + t_m n^T / d) K_1^{-1} \tilde{u}_1 = H_m^1 \tilde{u}_1 \quad (10)$$

$$\tilde{u} = H_m^1 H^{-1} \tilde{u}_2 = H_m^2 \tilde{u}_2 \quad (11)$$

Equations (10) and (11) are also applicable to the non-overlapping area. Projecting  $I_1$  and  $I_2$  through  $H_m^1$  and  $H_m^2$  onto the mosaic, respectively, to get warped images  $I'_1$  and  $I'_2$ . Obviously:

$$I'_i(\tilde{u}) = I_i \left( (H_m^i)^{-1} \tilde{u} \right) (i = 1, 2) \quad (12)$$

Figure 4 shows the experiment result on two school images used in [10]. Due to the parallax between  $I_1$  and  $I_2$ , blending  $I'_1$  and  $I'_2$  will cause ghosting. Therefore, the next section will use an optical-flow-based blending method (AOF) to further align the images.



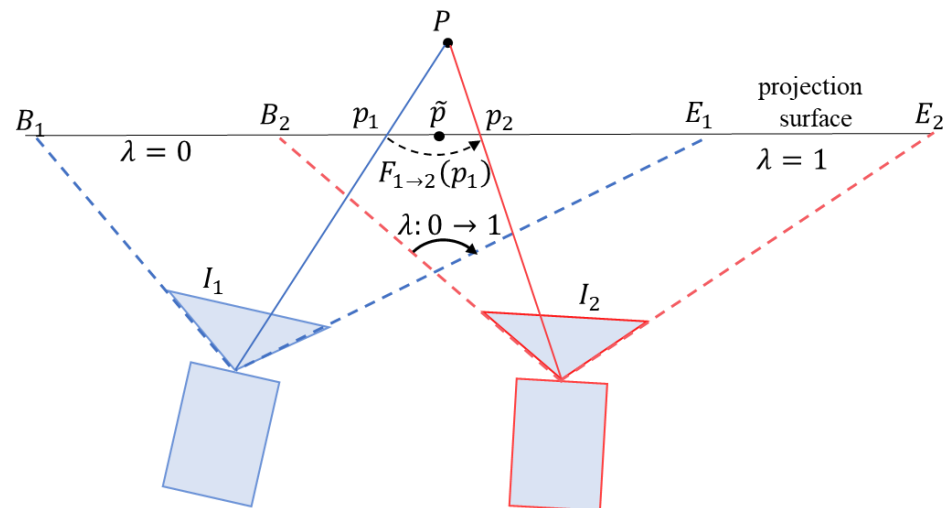
**Figure 4.** Image stitching based on sliding cameras and global projection plane. (a,b) show the warped images  $I_1'$  and  $I_2'$  of the input images of a school; (c) shows the average blending images of  $I_1'$  and  $I_2'$ . That is, in the overlapping area, the blended value is  $(I_1' + I_2')/2$ .

### 3.2. AOF: Image Alignment Based on Asymmetric Optical Flow

The mosaic generated by the SC method will inevitably have misalignment in most cases. So, the optical-flow-based method is further employed to achieve more accurate alignment. This section firstly introduces the image alignment process based on asymmetric optical flow (AOF), and then details the calculation method of AOF.

#### 3.2.1. Image Blending Process of AOF

$I_1$  and  $I_2$  are projected onto the custom projection surface to obtain warped images  $I_1'$  and  $I_2'$ , which are then blended to generate the mosaic  $I$ . As the 3D points of the scene are not always on the projection plane, ghosting artifacts can be seen in the mosaic, as shown in Figure 4 in the previous section. Direct multi-band image blending will lead to artifacts and blurring. As shown in Figure 5, point  $P$  is projected to two points  $p_1$  and  $p_2$  in the mosaic, resulting in duplication of content. To solve the ghosting problem in the mosaic, the optical-flow-based blending method in [22] is adopted.



**Figure 5.** Image blending based on optical flow.  $B_1E_2$  is the projection surface of the mosaic. In the overlapping areas (denoted by  $B_2E_1$ ) of  $I_1$  and  $I_2$ , we need to blend  $I_1'$  and  $I_2'$ . The 3D point  $P$  is outside the projection surface. When  $P$  is projected onto the projection surface, ghosting points  $p_1$  and  $p_2$  appear. Through the weighted blending of asymmetric optical flow,  $p_1$  and  $p_2$  are merged into point  $\tilde{p}$ , which solves the ghosting problem of stitching.

Suppose  $F_{2 \rightarrow 1}(p_2)$  represents the optical flow value of  $p_2$  in  $I_2'$  and  $F_{1 \rightarrow 2}(p_1)$  represents the optical flow value of  $p_1$  in  $I_1'$ . If the blending weight of pixel  $\tilde{p}$  in the overlapping area is  $\lambda$  (from the non-overlapping area of  $I_1'$  to the non-overlapping area of  $I_2'$ ),  $\lambda$  gradually transitions from 0 to 1, as shown in Figure 5, then after blending, the pixel value of image  $I$  at  $\tilde{p}$  is:

$$I(\tilde{p}) = (1 - \lambda)I_1'(\tilde{p}_1) + \lambda I_2'(\tilde{p}_2) \quad (13)$$

where  $\tilde{p}_1 = \tilde{p} + \lambda F_{2 \rightarrow 1}(\tilde{p})$  represents the corresponding value of  $\tilde{p}$  in  $I_1'$ , and  $\tilde{p}_2 = \tilde{p} + (1 - \lambda)F_{1 \rightarrow 2}(\tilde{p})$  represents the corresponding value of  $\tilde{p}$  in  $I_2'$ . That is, for any pixel  $\tilde{p}$  in the overlapping area of the mosaic, its final pixel value can be obtained by a weighted combination of its corresponding values in the two warped images using optical flow.

To achieve get a better blending effect, following the method presented by Meng and Liu [23], a softmax function is used to facilitate the mosaic transition quickly from  $I_1'$  to  $I_2'$ , narrowing the blending area. Furthermore, if the optical flow value of a warped image is larger, the salience is higher, and the blending weight of the warped image should be increased accordingly. Therefore, the following blending weight  $\beta$  can be employed:

$$\beta = \frac{\exp(\alpha_s \lambda (1 + \alpha_m M_2))}{\exp(\alpha_s (1 - \lambda) (1 + \alpha_m M_1)) + \exp(\alpha_s \lambda (1 + \alpha_m M_2))} \quad (14)$$

where  $M_1 = \|F_{2 \rightarrow 1}(\tilde{p})\|$  and  $M_2 = \|F_{1 \rightarrow 2}(\tilde{p})\|$  represents the optical flow magnitude;  $\alpha_s$  is the shape coefficient of the softmax function; and  $\alpha_m$  denotes the enhancement coefficient of the optical flow. The larger  $\alpha_s$  and  $\alpha_m$  are, the closer  $\beta$  is to 0 or 1, and the smaller the image transition area becomes.

Also, similar to multi-band blending, a wider blending area is used in smooth and color-consistent areas, and a narrower blending area is used in color-inconsistent areas. And the pixel consistency is measured using  $D_c$ :

$$D_c = \|I_1(\tilde{p}_1) - I_2(\tilde{p}_2)\| \quad (15)$$



The final blending parameter  $\alpha$  is obtained:

$$\lambda_d = \tanh(c_d D_c) \quad (16)$$

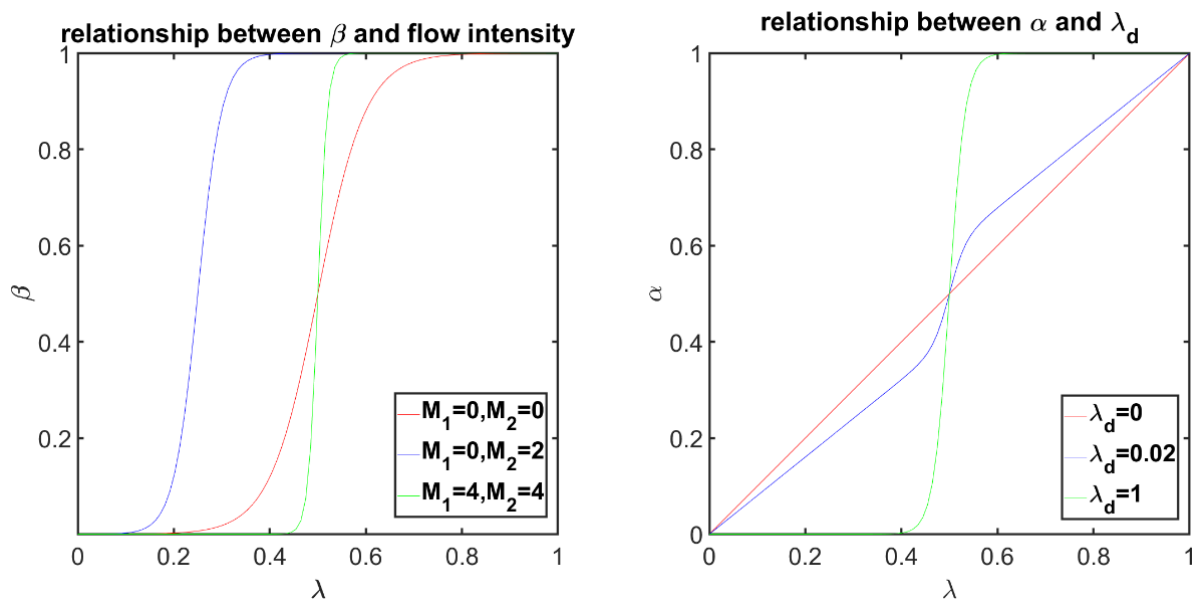
$$\alpha = (1 - \lambda_d)\lambda + \lambda_d\beta \quad (17)$$

$\beta$  corresponds to a fast transition from  $I'_1$  to  $I'_2$ ,  $\lambda$  corresponds to a linear transition from  $I'_1$  to  $I'_2$ . When the color differs slightly, the transition from  $I'_1$  to  $I'_2$  is linear, and when the color difference is large, we tend to have a fast transition from  $I'_1$  to  $I'_2$ .

Then the pixel value of the mosaic is:

$$I(\tilde{p}) = (1 - \alpha)I_1(\tilde{p}_1) + \alpha I_2(\tilde{p}_2) \quad (18)$$

The curve in the left panel of Figure 6 shows the curve of  $\beta$  with respect to  $\lambda$  under different optical flow intensities.  $\beta$  can be used to achieve quick transition of the mosaic from  $I'_1$  to  $I'_2$ , narrowing the transition area. In the case of a large optical flow, the blending weight of the corresponding image can be increased to reduce the transition area. The curve in the right panel of Figure 6 shows the influence of  $\lambda_d$  on the curve of  $\alpha$  as a function of  $\lambda$ . When  $\lambda_d$  is small, a wider fusion area tends to be used; otherwise, a narrower fusion area is used, which is similar to the blending of different frequency bands in a multi-band blending method.



**Figure 6.** Blending parameter curves. The figure on the left shows the  $\beta$  curves at different optical flow intensities. The right figure shows the  $\alpha$  curve at different  $\lambda_d$  values.

### 3.2.2. Calculation of Asymmetric Optical Flow

The general pipeline of the optical flow calculation is to construct an image pyramid, calculate the optical flow of each layer from coarse to fine, and use the estimated current-layer optical flow divided by the scaling factor as the initial optical flow of the finer layer until the optical flow of the finest layer is obtained [23–26]. Different methods are proposed to achieve better solutions that satisfy brightness constancy assumptions, solve large displacements and appearance variation [27,28], and address edge blur and improve temporal consistency [29–31]. Recently, some deep learning methods have been proposed. For example, RAFT (recurrent all-pairs field transforms for optical flow) [32] extracts per-pixel features, builds multi-scale 4D correlation volumes for all pairs of pixels, and iteratively updates a flow field through a recurrent unit. FlowFormer (optical flow Transformer) [33] is based on a transformer neural network architecture with a novel

encoder which effectively aggregates cost information of correlation volume into compact latent cost tokens, and a recurrent cost decoder which recurrently decodes cost features to iteratively refine the estimated optical flows.

In order to improve the optical flow calculation speed, we use the method based on optical flow propagation and gradient descent adopted in Facebook surround360 [34] to calculate the optical flow. When calculating the optical flow of each layer, first calculate the optical flow of each pixel from top to bottom and from left to right. From the optical flow values of the current-layer left and top pixels and upper-layer same-position pixel, the value with minimum error represented by Equation (19) is selected as the initial value of the current pixel. Then a gradient descent method is performed to update the optical flow value of the current pixel, and is then spread to the right and bottom pixels, as a candidate for the initial optical flow of the right and bottom pixels. After completing the forward optical flow propagation from top to bottom and from left to right, perform a reverse optical flow propagation and gradient descent from bottom to top and from right to left to obtain the final optical flow value.

When calculating the optical flow value  $F(u)$  of pixel  $u$ , the error function  $E(F(u))$  used is:

$$E(F(u)) = E_I + \alpha_S E_S + \alpha_T E_T \quad (19)$$

$$E_I(F(u)) = \|\nabla I_1(u) - \nabla I_2(u + F(u))\| \quad (20)$$

$$E_S(F(u)) = \|F(u) - G(u; \sigma) * F(u)\| \quad (21)$$

$$E_T(F(u)) = \|F(u) * D(1/W, 1/H)\| \quad (22)$$

where  $E_I$  denotes the optical flow alignment error of the edge image (which is Gaussian filtered to improve the robustness);  $E_S$  denotes the consistency error of the optical flow;  $G(u; \sigma) * F(u)$  denotes the Gaussian-filtered optical flow of pixel  $u$ ;  $E_T$  denotes the magnitude error after normalization of optical flow, with excessively large optical flow being penalized;  $W$  and  $H$  are the width and height of the current-layer image, respectively,  $D(1/W, 1/H)$  denotes the diagonal matrix with diagonal elements  $1/W$  and  $1/H$ .

### 3.3. Estimation of Image Intrinsic and Extrinsic Parameters

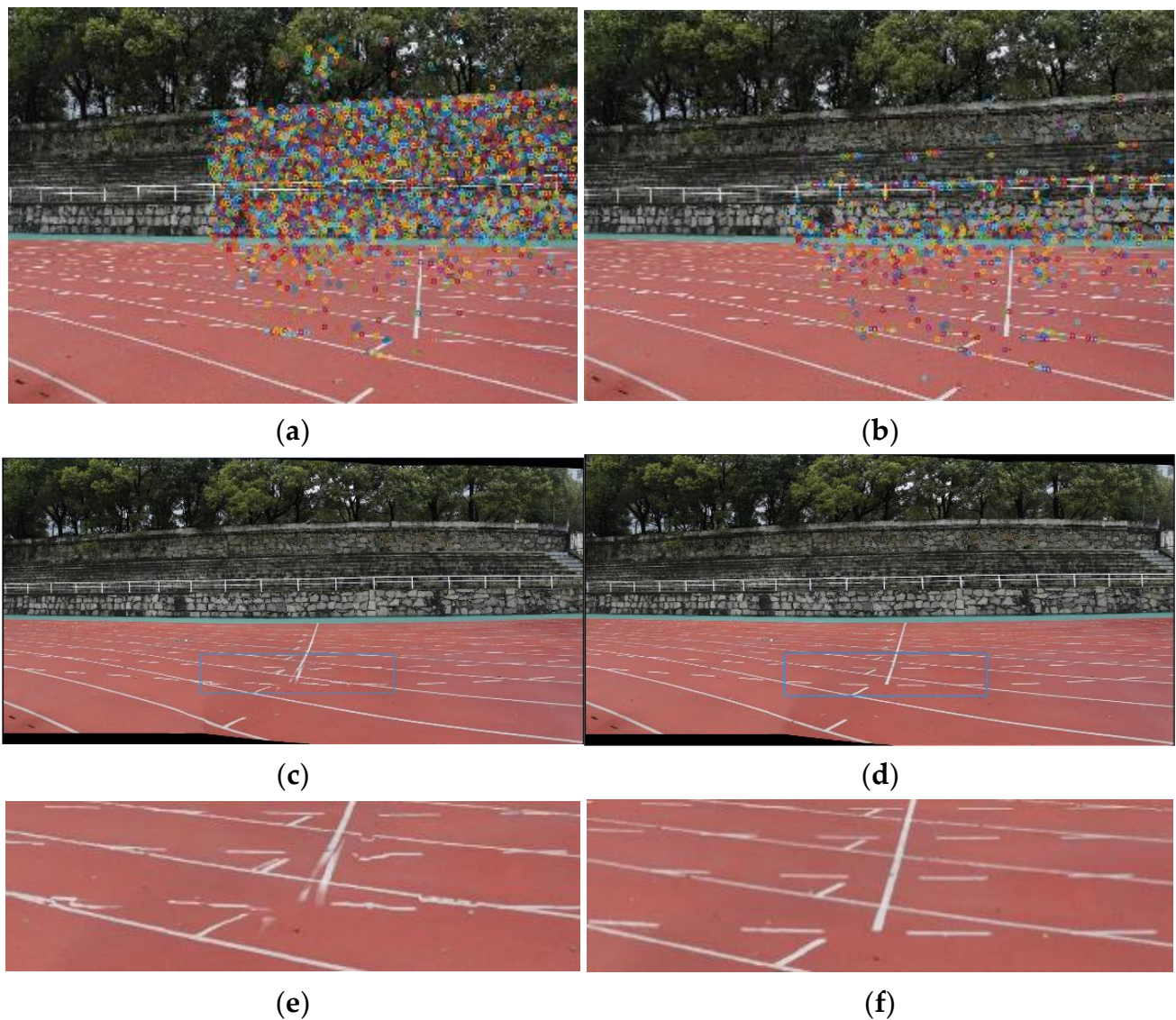
The SC-AOF method requires known camera parameters of images  $I_1$  and  $I_2$ . When only the intrinsic parameters  $K_1$  and  $K_2$  of an image are known, the essential matrix  $[t]_{\times} R$  between two images can be obtained by feature point matching, and the rotation matrix  $R$  and translation vector  $t$  between images can be obtained by decomposing the essential matrix. When both intrinsic and extrinsic parameters are unknown, the intrinsic parameters can be estimated by calibration [35,36] firstly, and then the extrinsic parameters of the image can be estimated accordingly. In these cases, both intrinsic and extrinsic parameters of image  $I_1$  and  $I_2$  can be estimated robustly.

When none of the above methods is feasible, it is necessary to calculate the fundamental matrix from the matched feature points and restore the camera internal and external parameters.

When the camera has zero skew, the known principal point and aspect ratio, then each intrinsic parameter matrix has only one degree of freedom (focal length of the camera). The total degree of freedom of the camera parameters is 7 (where  $t$  has 2 degrees of freedom due to the inability to recover scale,  $R$  has 3 degrees of freedom, and each camera has 1 degree of freedom), which is equal to the fundamental matrix  $F$ 's degree of freedom. The internal and external parameters of the image can be recovered using a self-calibration method [37]. But even if these constraints are met, the camera parameters by solved [37] suffer from large errors when the scene is approximately planar and the matching error is large. Therefore, we use the method of optimizing the objective function in [6] to solve the internal and external parameters of the camera.

To obtain an accurate fundamental matrix, firstly, the feature points need to be distributed more evenly in the image. As shown in Figure 7, a uniform and sparse distribution

of feature points can both reduce the computation time and obtain more robust intrinsic and extrinsic camera parameters and global projection planes, which will lead to improved stitching results.



**Figure 7.** The impact of feature point distribution on stitching results. The feature points are marked by small color circles, and the blue boxes indicate the regions where the enlarged images are located in the mosaics. The feature points in (a) are concentrated in the grandstand. The corresponding mosaic (c) is misaligned in the playground area. The feature points in (b) are evenly distributed within a  $2 \times 2$  grid. Although the total number of feature points is smaller, the mosaic (d) has better quality. (e,f) show the detail of mosaics.

Secondly, it is necessary to filter the matched feature points to exclude the influence of outliers. Use the similarity transformation to normalize the matched feature points. After normalization, the mean value of the feature points is 0, and the average distance to the origin is  $\sqrt{2}$ .

Thirdly, multiple homographies are estimated to exclude outlier points. Let  $F_{cond}$  and  $n$  denote all matched feature points and the total number of matched feature points. In  $F_{cond}$ , the RANSAC method with threshold  $\eta = 0.01$  is applied to compute homography  $H^1$  and its inlier set  $F_{inlier}^1$ , and the matches of isolated feature points which have no neighboring points within a 50 pixel distance are removed from  $F_{inlier}^1$ . A new candidate set is generated

by removing  $F_{inlier}^1$  from  $F_{cond}$ . Repeat the above steps to calculate  $m$  homography matrices  $H^m$  and corresponding inlier set  $F_{inlier}^m$  until  $\|F_{inlier}^m\| < 20$  or  $|F_{inlier}^m| < 0.05n$ . The final inlier set is  $F_{inlier} = \cup_{i=1}^m F_{inlier}^i$ . If  $m = 1$ , then there is only one valid plane. In this case, apply the RANSAC method with threshold  $\eta = 0.1$  to recalculate homography  $H^1$  and the corresponding inlier set  $F_{inlier}^1$ .

After excluding the outliers, for any matched points  $\{x_1, x_2\}$  in the inlier set  $F_{inlier}$ , the cost function is:

$$E(x_1, x_2) = (1 - \lambda)h(r_e, \sigma_e) + \lambda h(r_p, \sigma_p) \quad (23)$$

where  $\lambda$  balanced epipolar constraint and the infinite homography constraint, and generally take  $\lambda = 0.01$ .  $h$  is a robust kernel function, which can mitigate the effect of mis-matched points on the optimization of camera internal and external parameters.  $r_e$  and  $r_p$  denote the projection errors of the epipolar constraint and of the infinite homography constraint, respectively:

$$r_e = \frac{1}{\rho} x_2^T K_2^{-T} [t]_{\times} R K_1^{-1} x_1 = \frac{1}{\rho} x_2^T F x_1 \quad (24)$$

$$r_p = x_2 - \frac{1}{w} K_2 R K_1^{-1} x_1 \quad (25)$$

where  $\rho$  denotes the length of the vector composed of the first two components of  $Fx_1$ . That is, assuming  $Fx_1 = (a, b, c)^T$ , then  $\rho = \sqrt{a^2 + b^2}$ .  $w$  represents the third component value of the vector  $K_2 R K_1^{-1} x_1$ .

#### 4. Experiment

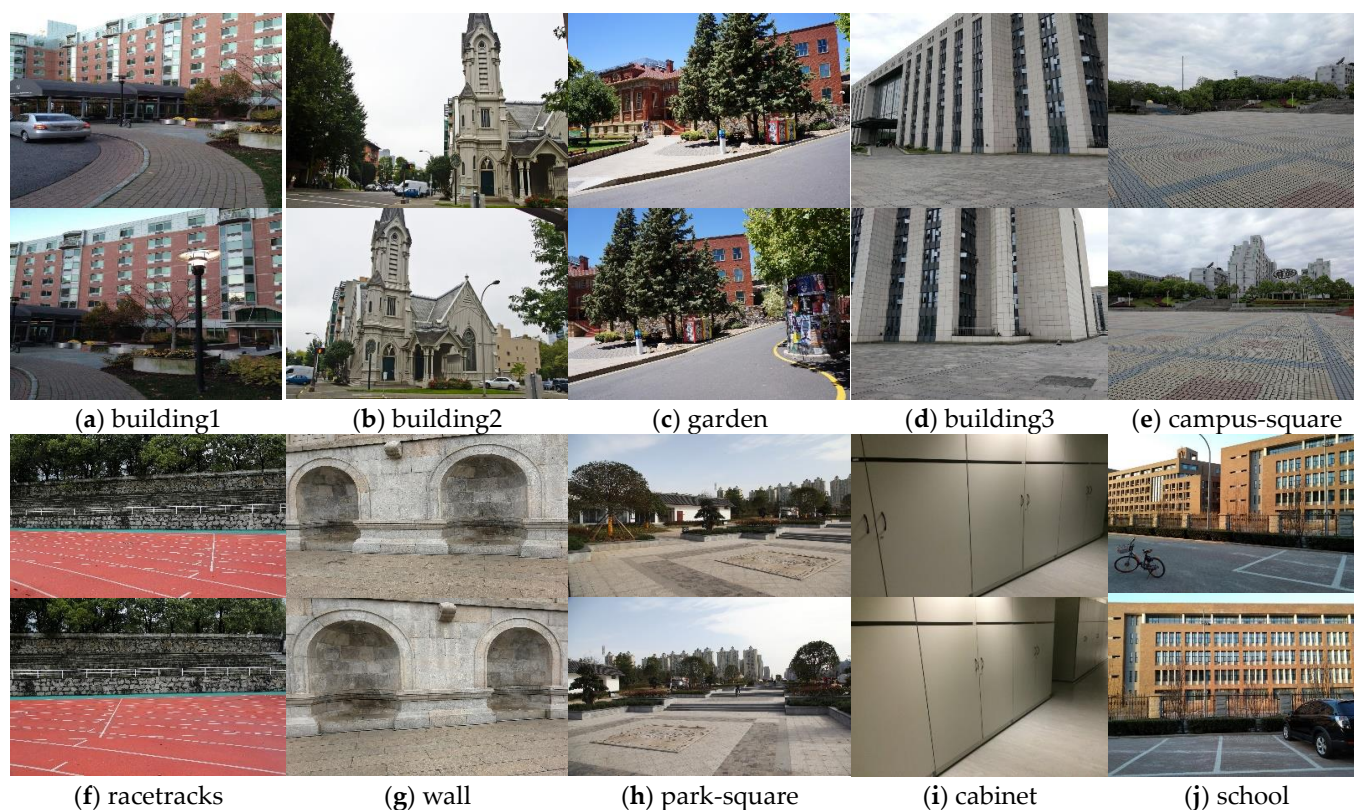
To verify the effectiveness of the SC-AOF method, the mosaics generated by our method and the existing APAP [4], AANAP [16], SPHP [14], TFT [7], REW [10] and SPW [18] methods are compared on some typical datasets used by others to verify the feasibility and advantages of the SC-AOF method in solving deformation and improving alignment. Next, the SC-AOF method is used together with other methods to demonstrate its compatibility. The image pairs used in the comparison experiment are shown in Figure 8.

##### 4.1. Effectiveness Analysis of SC-AOF Method

In this section, various methods of image stitching are compared and analyzed based on three indicators: perspective deformation, local alignment and running speed. The experimental setup is as follows.

- The first two experiments compare typical methods for solving perspective deformation and local alignment, respectively, and all the methods in the first two experiments are included in the third experiment to show the superiority of the SC-AOF method in all aspects.
- Since the averaging methods generally underperform compared to linear blending ones, all methods to be compared adopt linear blending to achieve the best performance.
- All methods other than ours use the parameters recommended by their proposers. Our SC-AOF method has the following parameter settings in optical-flow-based image blending:  $\alpha_s = 10$ ,  $\alpha_m = 100$ , and  $c_d = 10$ .





**Figure 8.** The image dataset for comparative experiments. The image pairs are initially used by stitching methods such as APAP, AANAP, and REW.

#### 4.1.1. Perspective Deformation Reduction

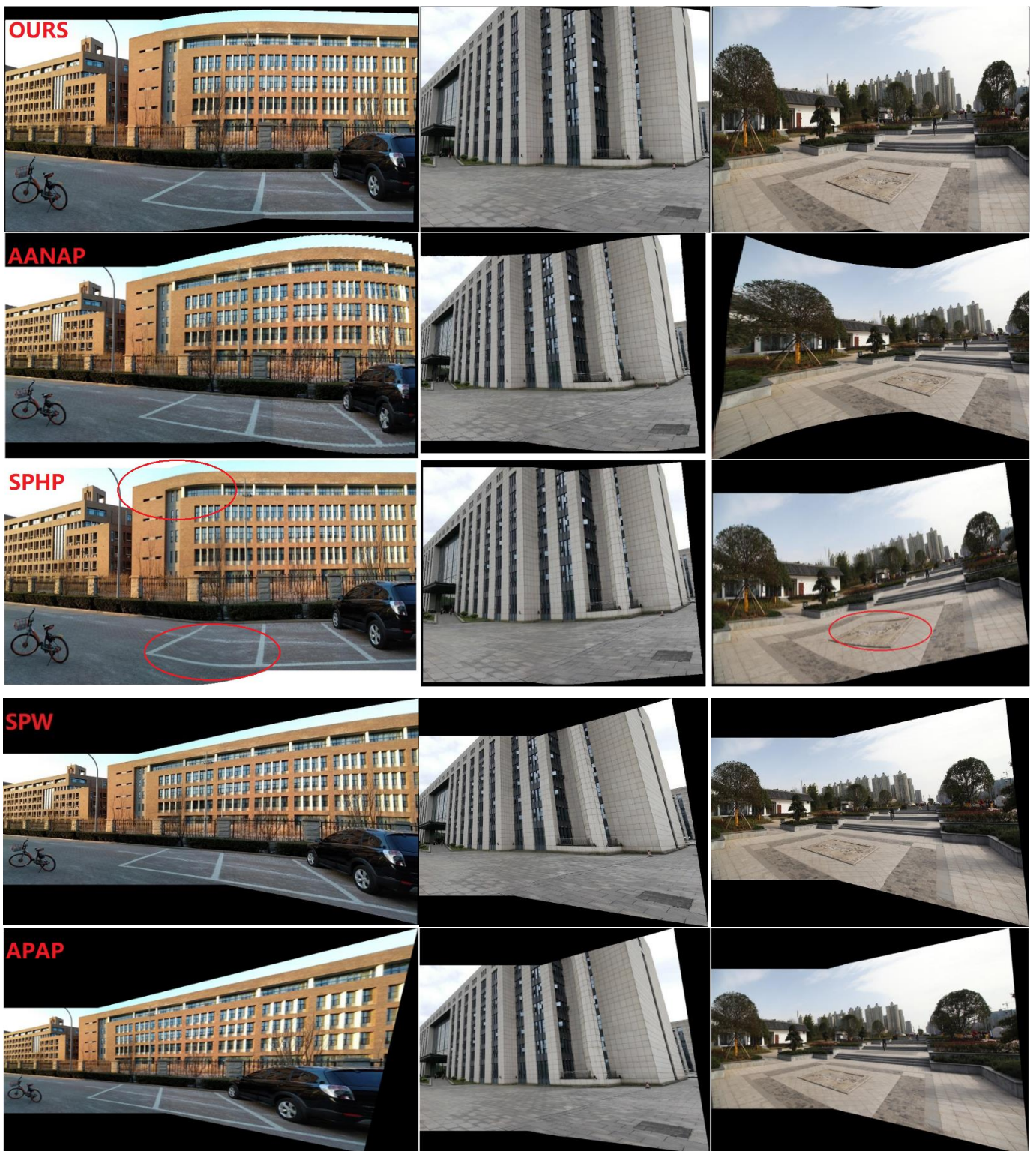
Figure 9 shows the results of the SC-AOF method versus the SPHP, APAP, AANAP and SPW methods for perspective deformation reduction in image stitching. School, building and park square datasets were used in this experiment. We can see from Figure 9 that, compared with the other methods, our SC-AOF method changes the viewpoint of the stitched image in a more natural manner and effectively eliminates perspective deformation. As explained below, all other methods underperform compared to our SC-AOF method.

The image stitched using the APAP method has its edges stretched to a large extent. This is because it does not process perspective deformation. This method only serves as a reference to verify the effectiveness of perspective-deformation-reducing algorithms.

The AANAP algorithm can achieve a smooth transition between the two viewpoints, but results in severely “curved edges”. And there is even more severe edge stretching for the park square dataset than that of the APAP method. This is because, when the AANAP method extrapolates from homographies, it linearizes the homography in addition to similarity transformation, causing affine deformation in the final transformation.

Compared with the APAP method, the SPW method makes no significant improvement in perspective deformation, except for the image in the first row. SPW preserves perspective consistency, so a multiple-viewpoint method excels in solving perspective deformation compared to single-viewpoint method.





**Figure 9.** Comparison of perspective deformation processing. From the first row to the last row, the mosaics generated by our method, AANAP, SPHP, SPW and APAP on the datasets are presented, respectively. The red elliptical boxes indicate the unnatural transitions in the mosaics.

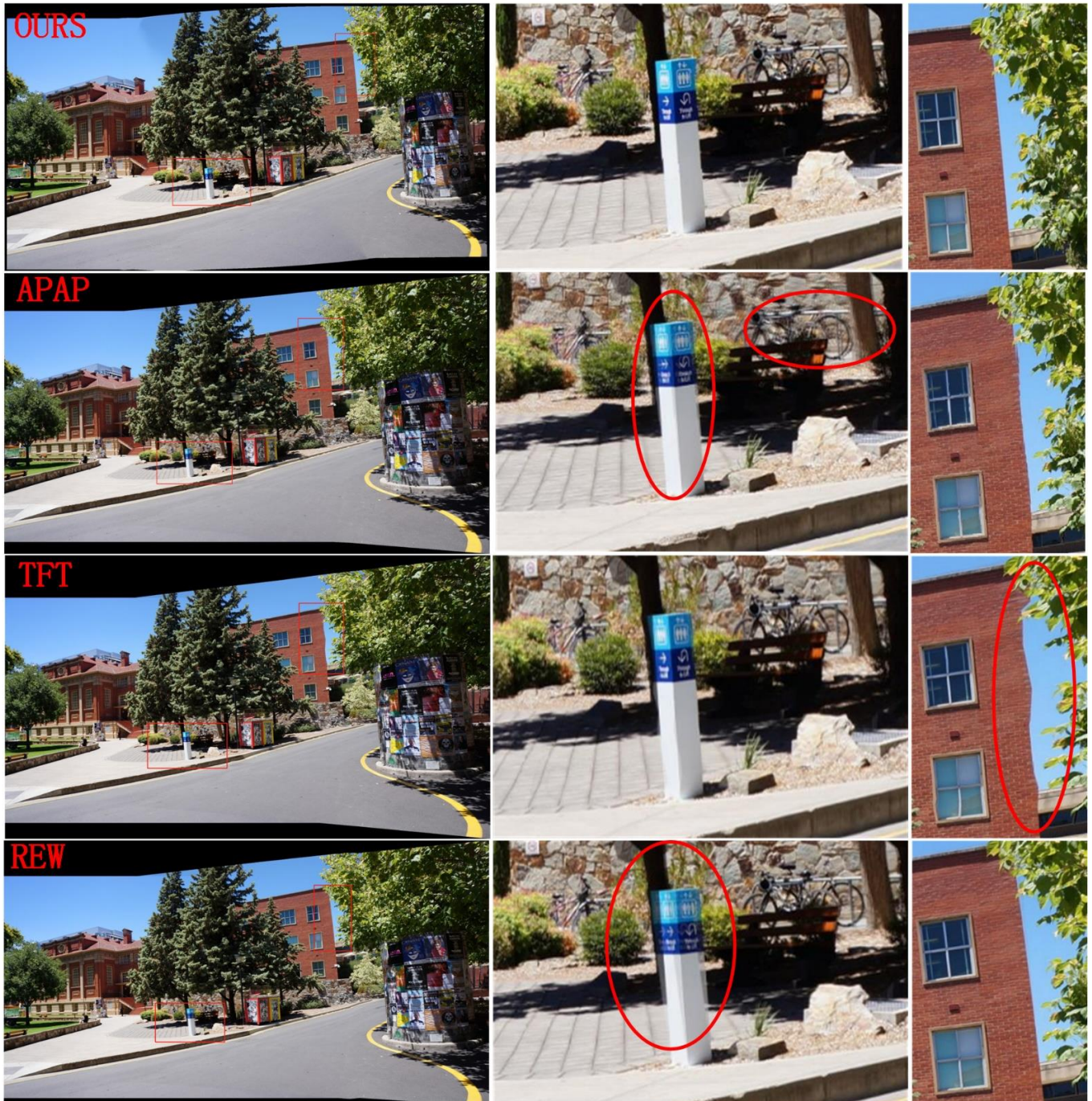
The SPHP algorithm performs well overall. However, it causes severe distortions in some areas (red circles in Figure 8c) due to the rapid change of viewpoints. This is because the SPHP method estimates the similarity transformation and interpolated homographies from global homography. As a result, the similarity transformation cannot reflect the real



scene information and the interpolated homographies may deviate from a reasonable image projection.

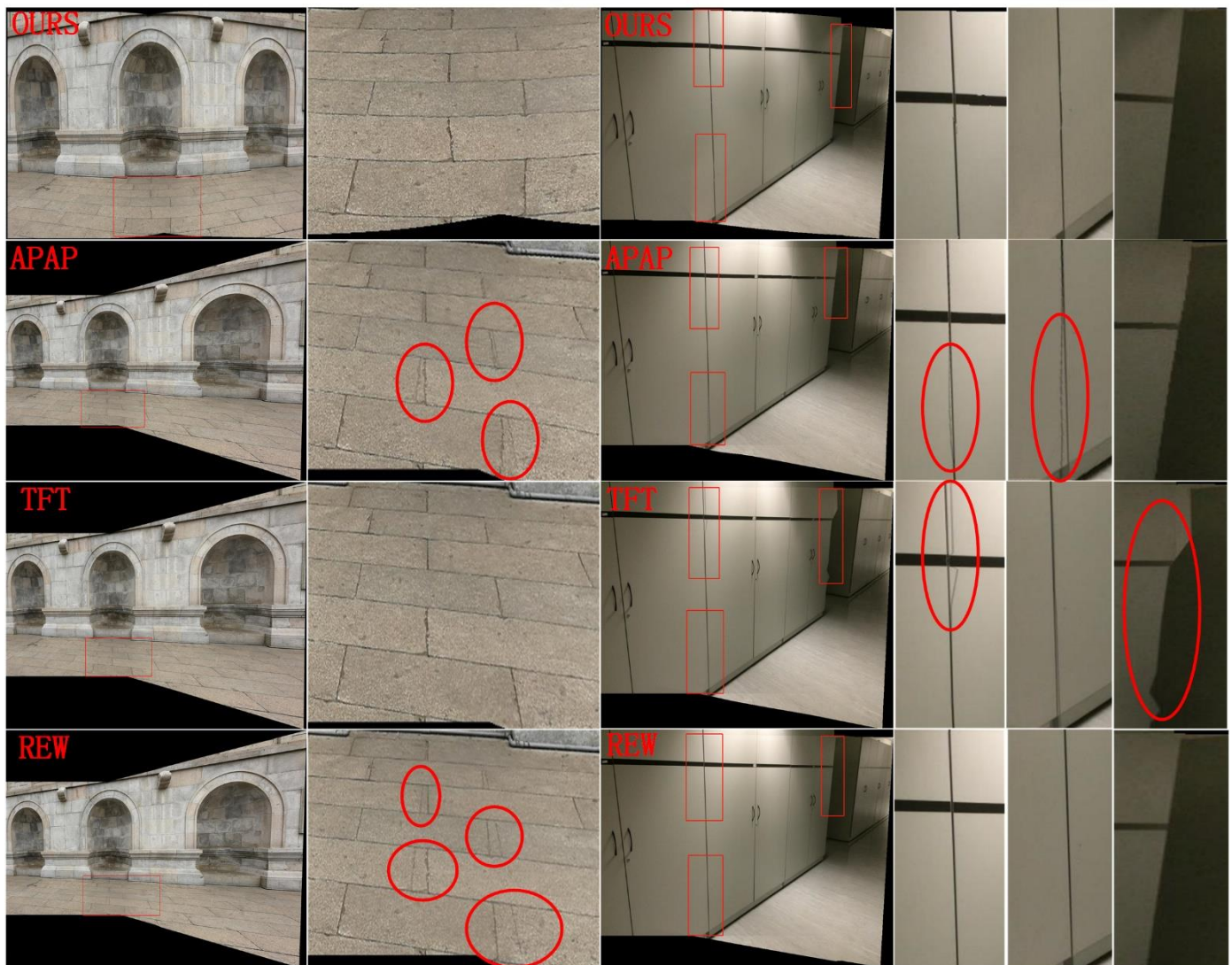
#### 4.1.2. Local Alignment

Figures 10 and 11 show the results of the SC-AOF method versus APAP, TFT and REW methods for local alignment in image stitching. It can be seen that SC-AOF performs well in all scenes, showing the effectiveness of our method in local alignment.



**Figure 10.** Qualitative comparison on the garden image pairs. From the first row to the last row, the mosaics and detail views generated by our method, APAP, TFT, REW are presented, respectively. The red boxes indicate the regions where the enlarged images are located in the mosaics. The red circles highlight errors and distortions.





**Figure 11.** Comparison of image alignment on the wall and cabinet image pairs. From the first row to the last row, the mosaics and detail views generated by our method, APAP, TFT, REW are presented, respectively. The blue boxes indicate the region where the enlarged images are located. The red circles highlight errors and distortions.

- The APAP method performs fairly well in most images, though with some alignment errors. This is because the moving DLT method smooths the mosaics to some extent.
- The TFT-generated stitched image is of excellent quality in planar areas. But when there is a sudden depth change in the scene, there are serious distortions. This is because large errors appear when calculating planes using three vertices of a triangle in the area with sudden depth changes.
- The REW method has large alignment errors in the planar area and aligns the images better than the APAP and TFT method in all other scenes. This is because the fewer feature points in the planar area might be filtered out as mismatched points by the REW method.

The SSIM (structural similarity) [38] is employed to objectively describe the alignment accuracy of different methods. SSIM measures the similarity between two images  $J_1$  and

$J_2$  to be blended in the overlapping area. For our two-step alignment method,  $J_1(u) = I_1'(u + \lambda F_{2 \rightarrow 1}(u))$ ,  $J_2(u) = I_2'(u + (1 - \lambda)F_{1 \rightarrow 2}(u))$ . The structural similarity is defined as:

$$SSIM(J_1, J_2) = \frac{(2\mu_1\mu_2 + C_1)(2\sigma_{12} + C_2)}{(\mu_1^2 + \mu_2^2 + C_1)(\sigma_1^2 + \sigma_2^2 + C_2)} \quad (26)$$

where  $\mu_1$  and  $\sigma_1$  represent the mean and standard deviation of pixel values within the overlapping area  $O$  of  $J_1$ , respectively.  $\mu_2$  and  $\sigma_2$  are the corresponding mean and standard deviation of  $J_2$ , respectively.  $\sigma_{12}$  is the covariance of pixel values in the overlapping area of  $J_1$  and  $J_2$ .  $C_1 = (k_1L)^2$  and  $C_2 = (k_2L)^2$  are constants used to maintain stability, where  $k_1 = 0.01$ ,  $k_2 = 0.03$ , and  $L$  is the dynamic range of pixel values (for 8-bit grayscale images,  $L = 255$ ).

The scores of all methods on the datasets building1, building2, garden, building, school, park-square, wall, cabinet, campus-square and racetracks are listed in Table 1. The best SSIM value is highlighted in bold.

**Table 1.** Comparison of SSIM.

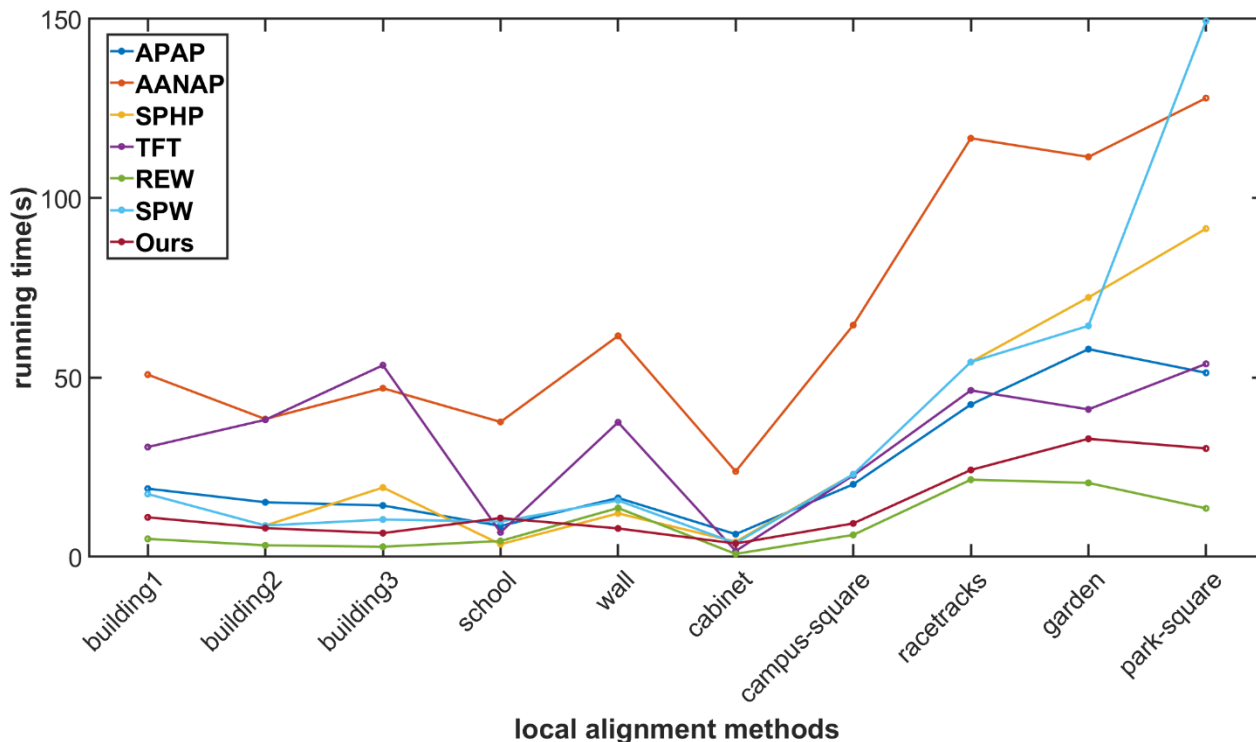
	APAP	AANAP	SPHP	TFT	REW	SPW	Ours
building1	0.88	0.87	0.75	0.88	0.89	0.86	<b>0.90</b>
building2	0.82	0.82	0.75	0.92	0.76	0.81	<b>0.93</b>
garden	0.90	0.92	0.81	0.82	<b>0.95</b>	0.92	0.93
building3	0.93	0.94	0.89	0.70	<b>0.96</b>	0.90	<b>0.96</b>
school	0.89	0.91	0.67	0.90	0.91	0.87	<b>0.93</b>
wall	0.83	0.91	0.68	0.90	0.82	0.81	<b>0.92</b>
park-square	0.95	0.96	0.80	0.97	<b>0.97</b>	0.95	<b>0.97</b>
cabinet	0.91	0.91	0.87	0.89	<b>0.98</b>	0.92	0.96
campus-square	0.92	0.94	0.84	0.95	<b>0.98</b>	0.93	0.97
racetracks	0.74	0.79	0.68	<b>0.86</b>	0.83	0.70	0.85

- APAP and AANAP have high scores on all image pairs, but the scores are lower than our method and REW, proving that APAP and AANAP blur mosaics to some extent.
- When SPHP is not combined with APAP, only the global homography is used to align the images, resulting in lower scores compared to other methods.
- TFT has higher scores on the datasets except for the building dataset. TFT can improve alignment accuracy but also bring instability.
- SPW combines quasi-homography and content-preserving warping to align images, which add other constraints while also reducing the accuracy of alignment, resulting in lower scores compared to REW and our method.
- Both REW and our method use a global homography matrix to coarsely align the images. Afterwards, in REW and our method, a deformation field and optical flow are applied to further align the images, respectively. Therefore, both methods have higher scores and robustness than other methods.

#### 4.1.3. Stitching Speed Comparison

The running speed is a direct reflection of the efficiency of each stitching method. Figure 12 shows the speed of the SC-AOF method versus the APAP, AANAP, SPHP, TFT, REW and SPW methods. The same image pairs as in the SSIM comparison are used in this experiment. It can be seen that the REW algorithm has the fastest stitching speed. The reason is that it only needs to calculate TPS parameters based on feature point matching and then compute the transformations of grid points quickly. Our SC-AOF method ranks

second in terms of stitching speed, and the AANAP algorithm requires the longest running time. Both the APAP and AANAP methods calculate the local homographies based on moving DLT, and the AANAP method also needs to calculate the Taylor expansion of anchor points.



**Figure 12.** Comparison on elapsed time. Our method is second only to REW in speed and is superior to other methods.

#### 4.1.4. Overall Scoring for All the Methods

In order to comprehensively and quantitatively evaluate our method and other methods in improving local alignment and reducing perspective deformation, we define a scoring method that assigns an integer score ranging from 0 to 10 to estimate the effectiveness and efficiency of stitching each image pair using each method. The total score is obtained by adding up the scores from four aspects:

1. The subjective scoring of perspective deformation reduction. The scores from 0 to 2, respectively, indicate severe deformation, slight relief of deformation, and less deformation.
2. The subjective scoring of local alignment. The score ranges from 0 to 2, where 0 indicates obvious ghosting in many regions, 1 indicates few or mild mismatches, and 2 indicates no apparent alignment errors.
3. The objective scoring of local alignment. The score ranges from 0 to 3. We define the mean and standard deviation of the SSIM values of different methods on the same image pair as  $\mu$  and  $\sigma$ , the SSIM of current method is  $x$ , the score of the method is 0, 1, 2 and 3, respectively, when  $x$  satisfies  $x - \mu < -\sigma$ ,  $x - \mu \in (-\sigma, 0)$ ,  $x - \mu \in (0, -\sigma)$  and  $x - \mu > \sigma$ .
4. The scoring of running time. Like the objective scoring for local alignment, we score 0 when the running time of the method is greater than the mean plus standard deviation. When the time is less than the mean plus standard deviation and greater than the mean, the score is 1. The score is 2 when the time is less than the mean and greater than the mean minus standard deviation. Otherwise, the score is 3.



The scoring results of these methods on the image pairs are shown in Table 2. The image pairs in Table 2 include those used in SSIM and runtime comparison, as well as the test image pairs in the Appendix B (specific comparison of the mosaics generated by different methods are shown in the Appendix B). Every scoring is displayed in the format as “the score of perspective deformation reduction + the subjective score of local alignment + the objective score of local alignment + the score of running time = the overall score”. The highest score is bolded and highlighted. Our SC-AOF method has the highest scores in all the image pairs except the worktable image pair. Given that our method and REW all scored highly and have the same scores on some image pairs, in order to prove that our method is indeed ahead of REW, rather than due to statistical bias, we performed a Wilcoxon test using MATLAB 2018b on all scores of our method and REW. The resultant  $p$ -values of 0.0106 and  $h = 1$  prove that the scores of REW and our method come from different distributions, our method has the better overall performance, and our method can maintain a desirable operation efficiency while guaranteeing the final image quality. Our method can have broad applications and promotion significance.

#### 4.2. Compatibility of SC-AOF Method

The SC-AOF method can not only be used independently to generate stitched image with reduced perspective deformation and low alignment error, but also be decomposed (into SC method and image blending method) and combined with other methods to improve the quality of the mosaic.

##### 4.2.1. SC Module Compatibility Analysis

The sliding camera (SC) module in the SC-AOF method can not only be used in the global alignment model, but also be combined with other local alignment models (e.g., APAP and TFT) to solve perspective deformation while maintaining the alignment accuracy. The implementation steps are as follows.

1. Use the global similarity transformation to project  $I_2$  onto the  $I_1$  coordinate system to calculate the size and mesh vertices of the mosaic;
2. Use Equations (6)–(9) to calculate the weights of mesh vertices and the projection matrix, replace the homography  $H$  in (2) with the homography matrix in local alignment model, and bring them into (12) to compute the warped images and blend them.

Figure 13 presents the stitched images when using the TFT algorithm alone vs. using the TFT algorithm combined with the SC method. The combined method is more effective in mitigating edge stretching, and it generates more natural images. This shows that the SC method can effectively solve perspective deformation suffered by the local alignment method.

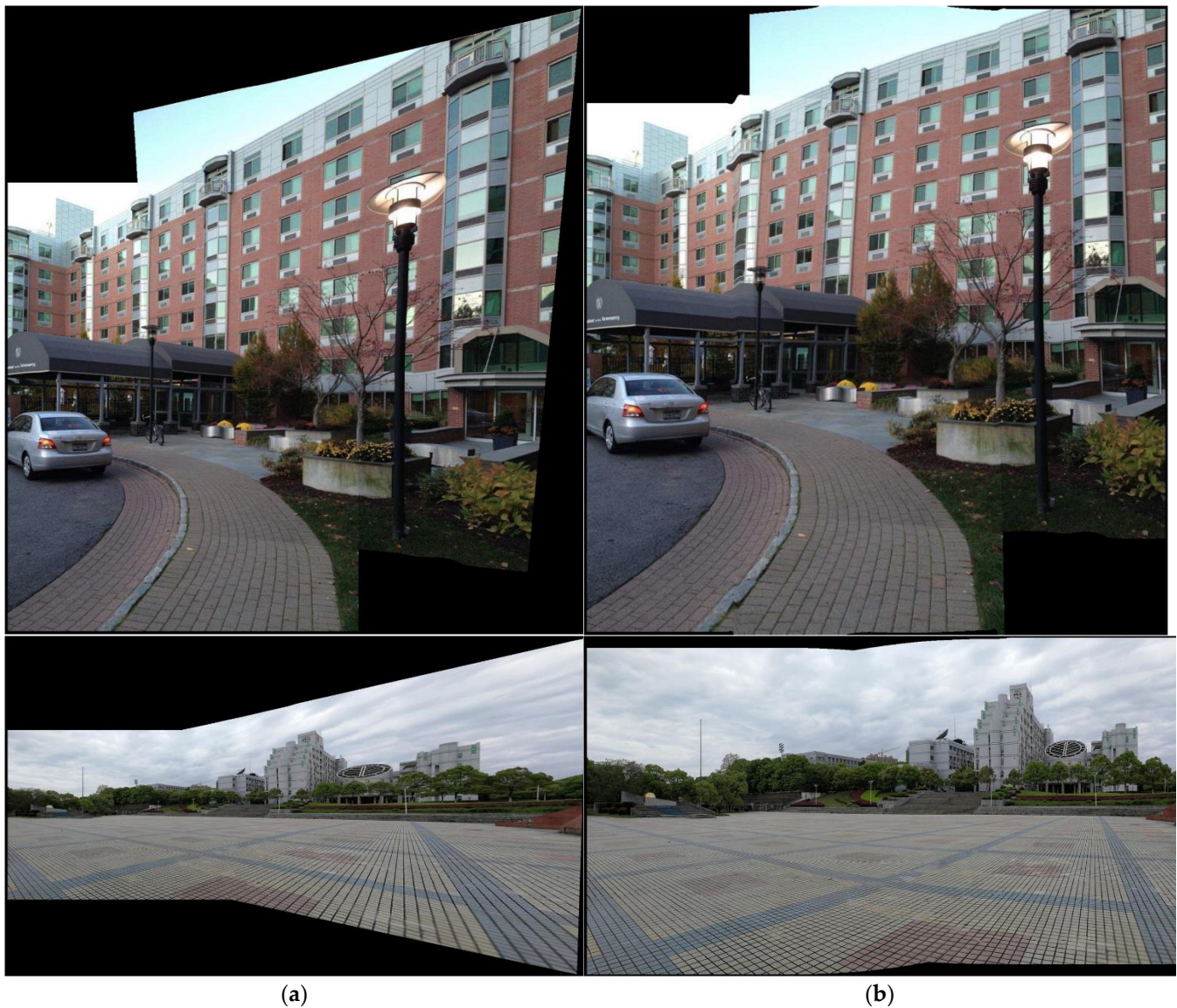
##### 4.2.2. Blending Module Compatibility Analysis

The asymmetric optical-flow-based blending in the SC-AOF method can also be used in other methods to enhance the final stitching effect. The implementation steps are as follows.

1. Generate two projected images using one of the other algorithms and calculate the blending parameters based on the overlapping areas;
2. Set the optical flow value to be 0, replace linear blending parameter  $\lambda$  with  $\alpha$  in Equation (17) to blend warped images, preserve the blending band width in the low-frequency area and narrow the blending width in the high-frequency area to obtain a better image stitching effect.

Table 2. The soring results on the image pairs.

	APAP	AANAP	SPHP	TFT	REW	SPW	Ours
<b>building1</b>	1+2+2+2=7	2+2+2+0=6	2+2+0+2=6	1+2+2+1=6	<b>1+2+2+3=8</b>	1+2+1+2=6	<b>2+2+2+2=8</b>
<b>building2</b>	0+1+1+2=4	1+2+1+0=4	2+0+0+2=4	1+0+3+0=4	1+2+0+3=6	1+1+1+2=5	<b>2+2+3+2=9</b>
<b>garden</b>	1+2+2+1=6	2+2+2+0=6	2+1+0+1=4	1+1+0+2=4	<b>1+2+3+3=9</b>	2+2+2+1=7	<b>2+2+3+2=9</b>
<b>building3</b>	1+1+2+2=6	2+1+2+0=5	2+2+1+2=7	0+0+0+0=0	<b>1+2+2+3=8</b>	1+2+2+2=7	<b>2+2+2+2=8</b>
<b>school</b>	0+2+2+2=6	2+2+2+0=6	2+2+0+2=6	0+2+2+2=6	0+2+2+2=6	1+2+2+2=7	<b>2+2+2+2=8</b>
<b>wall</b>	0+1+1+2=4	2+2+2+0=6	2+0+0+2=4	1+2+2+1=6	1+1+1+2=5	0+1+1+2=4	<b>2+2+3+2=9</b>
<b>park-square</b>	1+2+2+2=7	1+2+2+0=5	2+0+0+1=3	1+2+2+2=7	<b>1+2+2+3=8</b>	1+2+2+0=5	<b>2+2+2+2=8</b>
<b>cabinet</b>	1+1+1+2=5	2+1+1+0=4	2+2+0+2=6	2+0+1+2=5	<b>2+2+3+2=9</b>	1+2+2+2=7	<b>2+2+3+2=9</b>
<b>campus-square</b>	0+2+1+2=5	1+2+2+0=5	2+2+0+2=6	0+2+2+2=6	<b>0+2+3+3=8</b>	0+2+1+2=5	<b>2+2+2+2=8</b>
<b>racetracks</b>	2+1+1+2=6	2+1+2+0=5	2+0+0+1=3	<b>2+2+3+2=9</b>	<b>2+2+2+3=9</b>	1+1+0+1=3	<b>2+2+3+2=9</b>
<b>roundabout</b>	<b>2+2+2+2=8</b>	2+2+2+0=6	2+2+0+1=5	2+1+2+1=6	<b>2+2+2+2=8</b>	2+2+0+2=6	<b>2+2+2+2=8</b>
<b>fence</b>	1+2+2+2=7	2+2+2+0=6	2+1+0+2=5	1+1+2+1=5	1+2+2+2=7	<b>2+2+2+2=8</b>	<b>2+2+2+2=8</b>
<b>railtracks</b>	2+1+1+2=6	2+2+2+0=6	2+0+0+2=4	2+2+2+2=8	2+2+2+2=8	2+1+1+0=4	<b>2+2+3+2=9</b>
<b>temple</b>	1+2+2+2=7	2+2+2+0=6	2+2+0+2=6	1+1+2+2=4	1+2+2+2=7	1+1+1+2=5	<b>2+2+2+2=8</b>
<b>corner</b>	2+2+2+1=7	2+1+2+0=5	2+1+1+2=6	2+0+0+2=4	<b>2+2+2+2=8</b>	2+1+2+2=7	<b>2+2+2+2=8</b>
<b>shelf</b>	2+2+2+1=7	2+2+2+0=6	2+2+0+2=6	2+1+2+2=7	<b>2+2+2+3=9</b>	2+2+2+1=7	2+2+2+2=8
<b>standing-he</b>	1+1+2+2=6	2+1+1+1=5	2+1+1+2=6	0+0+1+2=3	1+1+2+2=6	1+1+2+0=4	<b>2+1+3+2=8</b>
<b>foundation</b>	1+2+2+2=7	2+2+2+0=6	2+2+0+2=6	1+1+3+2=7	<b>2+2+2+2=8</b>	1+2+1+0=4	<b>2+2+2+2=8</b>
<b>guardbar</b>	1+1+2+1=5	2+1+2+0=5	2+1+0+2=5	1+1+3+2=7	1+1+2+2=6	1+1+1+1=4	<b>2+1+2+2=7</b>
<b>office</b>	1+1+2+2=6	2+1+2+0=5	2+1+0+2=5	0+1+2+1=4	1+1+1+3=6	0+1+2+1=4	<b>2+2+3+2=9</b>
<b>plantain</b>	1+2+2+2=7	1+2+2+0=5	2+2+1+2=7	0+0+0+1=1	<b>2+1+2+3=8</b>	1+2+2+2=7	<b>2+2+2+2=8</b>
<b>building4</b>	1+1+2+1=5	2+1+2+0=5	2+1+0+2=5	1+1+2+2=6	1+2+2+2=7	1+2+1+2=6	<b>2+2+3+2=9</b>
<b>potberry</b>	<b>2+2+2+2=8</b>	<b>2+2+2+2=8</b>	2+2+0+2=6	2+2+2+1=7	<b>2+1+2+3=8</b>	1+1+1+0=3	<b>2+2+2+2=8</b>
<b>lawn</b>	1+2+2+2=7	1+2+2+2=7	2+2+0+0=4	1+2+2+1=6	<b>1+2+2+3=8</b>	1+2+2+1=6	<b>2+2+2+2=8</b>
<b>worktable</b>	2+1+2+2=7	2+1+1+0=4	1+1+0+2=4	2+0+2+2=6	<b>2+2+2+2=8</b>	2+1+2+0=5	<b>2+2+2+2=8</b>



**Figure 13.** The combination of TFT and moving cameras method. (a) The mosaics created using TFT. (b) The mosaics obtained by adding the moving camera method to TFT.

Figure 14 shows the image stitching effect of the APAP algorithm when using linear blending vs. when using our blending method. It can be seen that the blurring and ghosting in the stitched image are effectively mitigated when using our blending method. This shows that our blending algorithm can blend the aligned images better.





**Figure 14.** The combination of APAP and our blending method. (a) The mosaic and detail view generated by the APAP using linear blending. (b) The results of APAP combined with our blending method. The red elliptical boxes indicate the regions where the enlarged images are located.

## 5. Conclusions

In this paper, to solve the perspective deformation and misalignment in image stitching using homographies, a SC-AOF method is proposed. In image warping, a new virtual camera and a projection matrix are generated as the observation perspective in the overlapping area by interpolating between two projection matrices. The overlapping area transitions gradually from one viewpoint to another to achieve preservation of the viewpoint and the smooth transition of the stitched image, and thus solve the perspective deformation problem. In image blending, the optical-flow-based blending algorithm is proposed to further improve alignment accuracy. The width of the blending area is automatically adjusted according to the softmax function and alignment accuracy. Finally, extensive comparison experiments are conducted to demonstrate the effectiveness of our algorithm in reducing perspective deformation and improving alignment accuracy. In addition, our algorithm had broad applicability, as its component modules can be used with other algorithms to mitigate edge stretching and improve alignment accuracy.

However, the proposed local alignment method may fail if the input images contain large parallax, which cause severe occlusion to prevent us from obtaining the correct optical flow. The problem of local alignment failure caused by large parallax also exists in other local alignment methods. Exploring more robust optical flow calculation and occlusion processing methods to reduce misalignment in a large parallax scene is an interesting research direction for future work.

**Author Contributions:** Conceptualization, Q.L. and J.C.; methodology, J.C.; software, J.C.; validation, J.C.; formal analysis, Q.L.; investigation, Q.L. and J.C.; resources, Y.L.; data curation, Y.L.; writing—original draft preparation, J.C.; writing—review and editing, L.Z.; visualization, L.Z.; supervision, Q.L.; project administration, Q.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** Author Jiayi Chang and Yanju Liang were employed by the company Wuxi Iot Innovation Center Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Appendix A

Given that the abundance of symbols and abbreviations in this paper can lead to reading confusion, the symbols and abbreviations are listed and explained in Tables A1 and A2.

**Table A1.** The symbols and their explanations.

Symbols	Description
$I_1, I_2, I$	the source image pair and the final mosaic
$n$	the global projection plane
$\Omega_1, \Omega_2$	the non-overlapping area of $I_1, I_2$
$\Omega_0$	the overlapping area of $I_1$ and $I_2$
$P_1, P_2$	the camera projection matrix of $I_1, I_2$
$P_{im}, P_m$	the projection matrix of virtual sliding camera
$P'_2$	the adjusted projection matrix of $I_2$ to unify the pixel coordinates of $I_1$ and $I_2$
$u^i, u$	the non-homogeneous coordinate of the pixel in $I$
$\tilde{u}^i, \tilde{u}$	the homogeneous coordinate of the pixel in $I$
$u_1^i, u_1, u_2^i, u_2$	the non-homogeneous coordinate of the pixel in $I_1, I_2$
$\tilde{u}_1^i, \tilde{u}_1, \tilde{u}_2^i, \tilde{u}_2$	the homogeneous coordinate of the pixel in $I_1, I_2$
$S, S_i$	the sampling points on the plane $n$ which are projected onto $u$ and $u^i$ in $I$
$p, p_i$	the 3D scene points
$C_1, C_2$	the cameras corresponding to $I_1$ and $I_2$
$K_1, K_2$	the internal parameter matrices of $C_1$ and $C_2$
$I_{3 \times 3}$	the 3 by 3 identity matrix
$R, t$	the rotation matrix and the location of the optical center of $C_2$ in $C_1$ 's coordinate system
$\tilde{y}_1, \tilde{y}_2$	normalized coordinates of $\tilde{u}_1$ and $\tilde{u}_2$
$N^*$	the similarity matrix between $I_1$ and $I_2$
$K'_2, R'_2$	the rotation matrix and translation vector corresponding to $P'_2$
$K_m, R_m, t_m$	the internal parameter matrix, the rotation matrix and the translation vector corresponding to $P_m$
$q_1, q_2, q_m$	the quaternions corresponding to $I_{3 \times 3}, R'_2$ and $R_m$
$H_m^i (i = 1, 2)$	the homography between $I_i$ and $I$
$I'_i (i = 1, 2)$	the warped image of $I_i$ using $H_m^i$
$F_{i \rightarrow j}(p_i)$	the optical flow of $p_i$ in $I'_j$ which makes $I'_j(p_i) = I'_j(p_i + F_{i \rightarrow j}(p_i))$



Table A1. Cont.

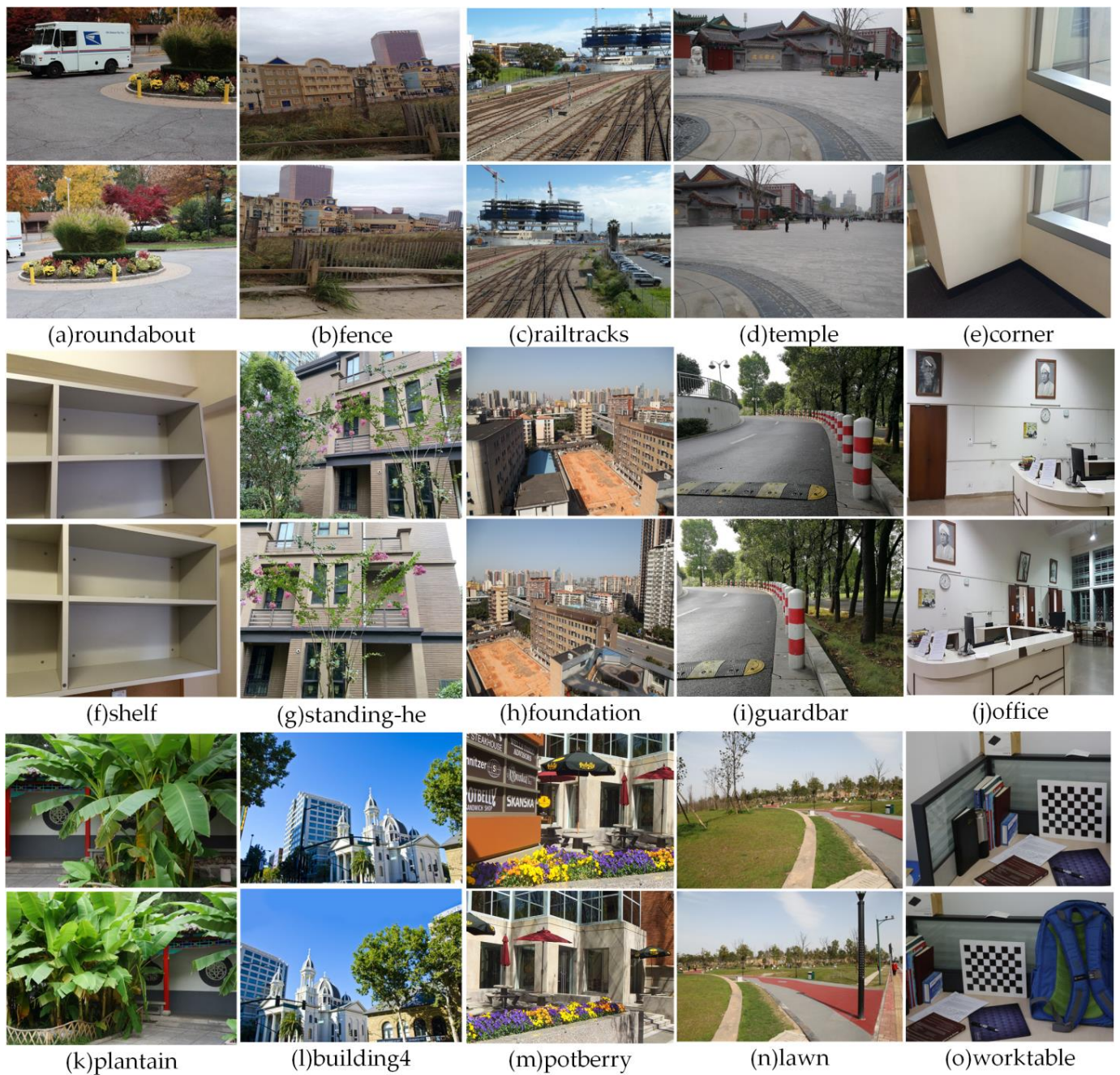
Symbols	Description
$\tilde{p}, \tilde{p}_1, \tilde{p}_2$	the pixel in $I$ and corresponding pixel in $I'_1, I'_2$
$M_1, M_2$	the optical flow magnitude of $F_{2 \rightarrow 1}(\tilde{p})$ and $F_{1 \rightarrow 2}(\tilde{p})$
$\alpha_s, \alpha_m$	the softmax function's shape coefficient and the optical flow's enhancement coefficient
$\lambda$	the weight which makes $I$ transition from $I'_1$ to $I'_2$ linearly
$\beta$	the softmax weight to transition $I$ from $I'_1$ to $I'_2$ fastly
$\alpha$	the linear combination of $\lambda$ and $\beta$ which makes the transition of $I$ from $I'_1$ to $I'_2$ depends on the color difference
$D_c, \lambda_d$	the color difference and the hyperbolic tangent function of the color difference
$F(u)$	the optical flow of $u$
$E(F(u))$	the error function of $F(u)$ used for solving the optimal optical flow
$E_I, E_S, E_T$	the optical flow's alignment error, consistency error and penalty for large value
$F_{inlier}^i, H^i$	the $i$ -th homography transforming $I_1$ to $I_2$ and the corresponding inlier set
$F_{cond}, F_{inlier}$	the initial set and the final inlier set of matched feature points
$r_e, r_p$	the projection errors of the epipolar constraint and of the infinite homography constraint
$h(r, \sigma)$	the robust kernel function to reduce the impact of false matches to optimization

Table A2. The abbreviations and their explanations.

Abbreviation	Meaning
SC	sliding camera, proposed by us to solve the perspective deformation
AOF	asymmetric optical flow, proposed by us to solve the local alignment
APAP	as-projective-as-possible, used to solve the local alignment by location-dependent homography warping
DLT	direct linear transform, used for estimating the parameters of the homography
REW	robust elastic warping, used to improve the local alignment using deformation fields
TPS	thin-plate spline, used to compute deformation fields corresponding to matched feature points
TFT	triangular facet approximation, using scene triangular facet estimating to improve the local alignment
NIS	natural image stitching, a local alignment method using the depth map
SPHP	shape preserving half projective, solving perspective deformation by gradually changing the resultant warp from projective to similarity
AANAP	adaptive as-natural-as-possible, a method to solve perspective deformation
GSP	global similarity prior, used to align images and reduce deformation
SPW	single-projective warp, which adopts the quasi-homography warp to mitigate projective distortion and preserve single perspective
SPSO	structure preservation and seam optimization, a method can obtain precise alignment while preserving local and global image structures.
GES-GSP	geometric structure preserving-global similarity prior, based on GSP to further protect the large-scale geometric structure from distortion
SIFT	scale-invariant feature transform, a feature detection and description method
SURF	speed-up robust feature, a feature detection and description method, faster than SIFT
KNN	k-nearest neighbor, a feature matching method
RAFT	recurrent all-pairs field transforms, estimating optical flow based on deep learning
RANSAC	random sample consensus, used to filter outliers and estimate model parameters

## Appendix B

In this section, some supplementary experiments about perspective deformation reduction and local alignment are added. The image pairs used in the supplementary experiments are shown in Figure A1.



**Figure A1.** The image dataset for supplementary comparative experiments. The image pairs are initially used by stitching methods such as APAP, AANAP, REW. Four yellow Chinese artistic characters are shown on the temple's wall in the upper image of (d).

Figures A2–A4 show the comparisons of perspective deformation reduction of our method, AANAP, SPHP, SPW and APAP.

The comparisons of local alignment of our method, APAP, TFT and REW are shown in Figures A5–A12. The detail images inside the red rects are the image regions with misalignment and shown directly right the mosaics.





Figure A2. Qualitative comparisons of perspective deformation reduction on the building1, fence and building4 image pairs.





Figure A3. Qualitative comparisons of perspective deformation reduction on the foundation and office image pairs.





Figure A4. Qualitative comparisons of perspective deformation reduction on the standing-he and lawn image pairs.



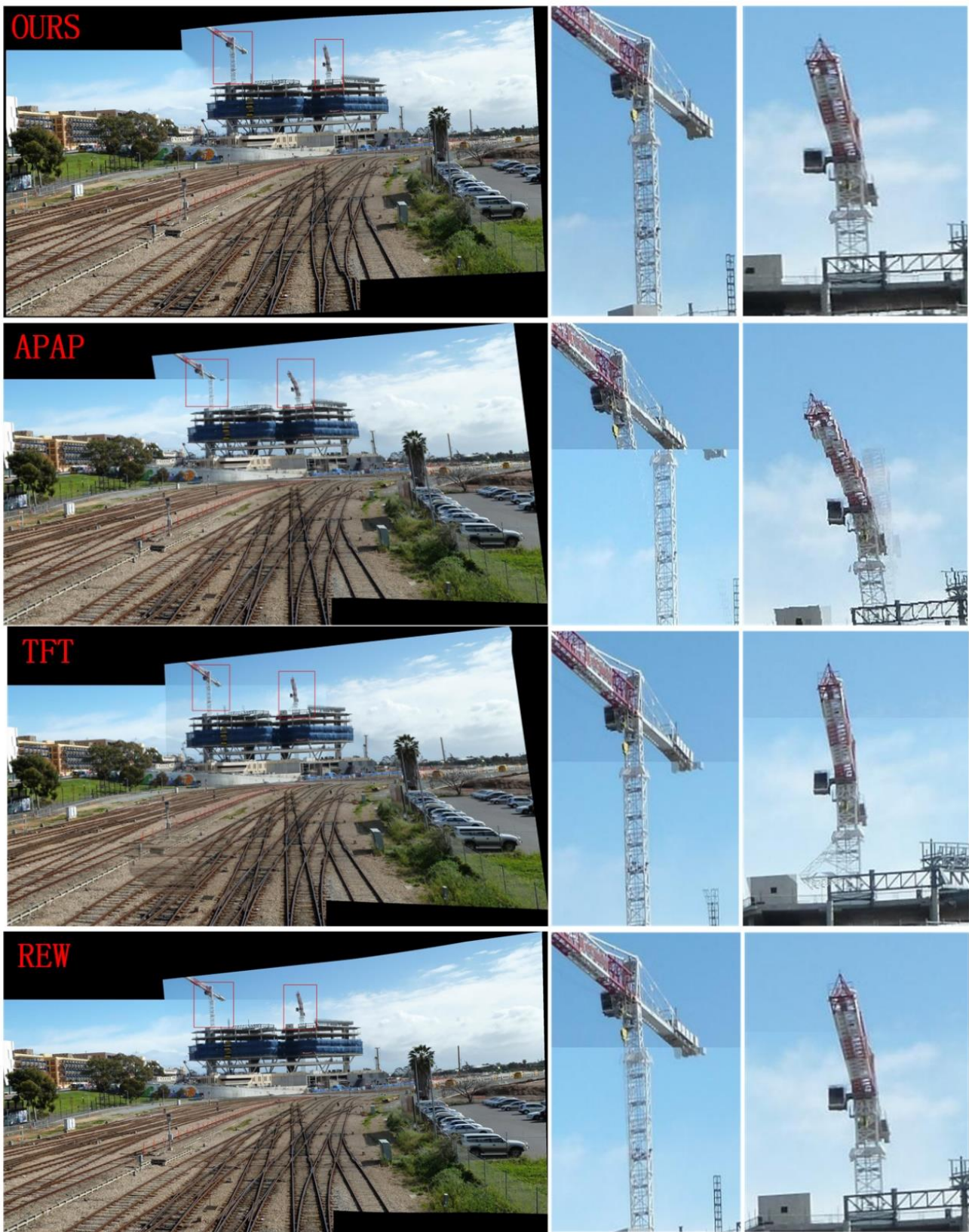
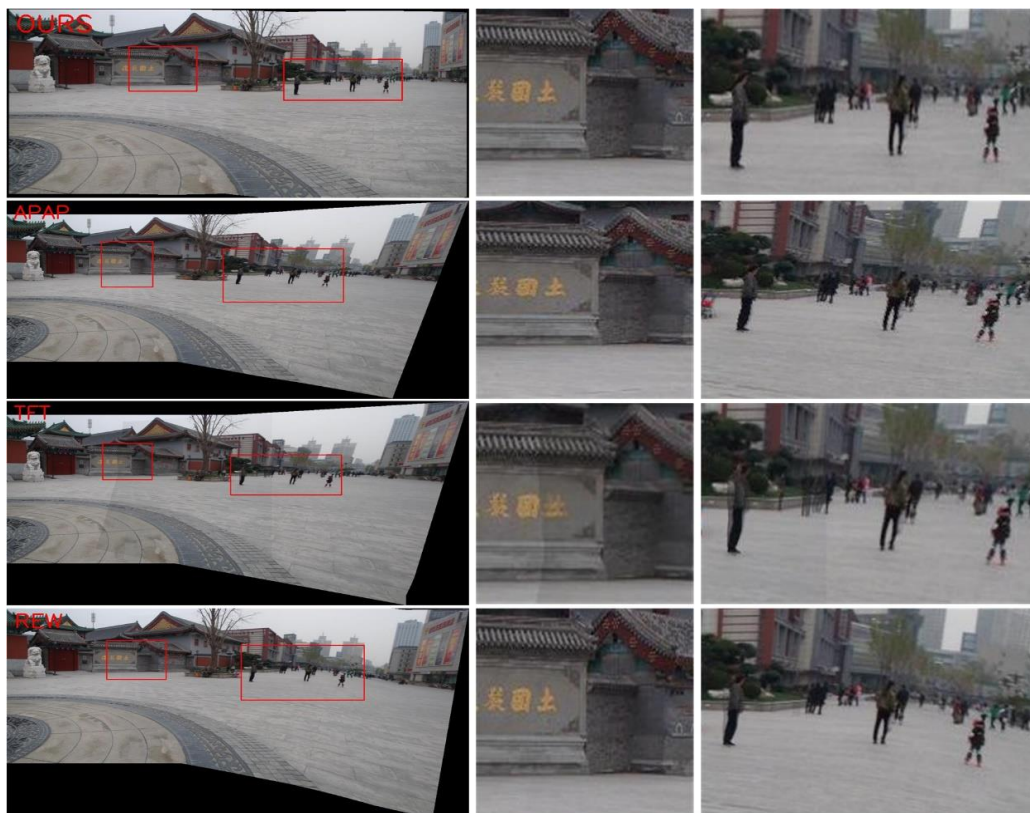


Figure A5. Qualitative comparisons of local alignment on the railtracks image pair. The red boxes indicate the regions where the enlarged images are located in the mosaics.



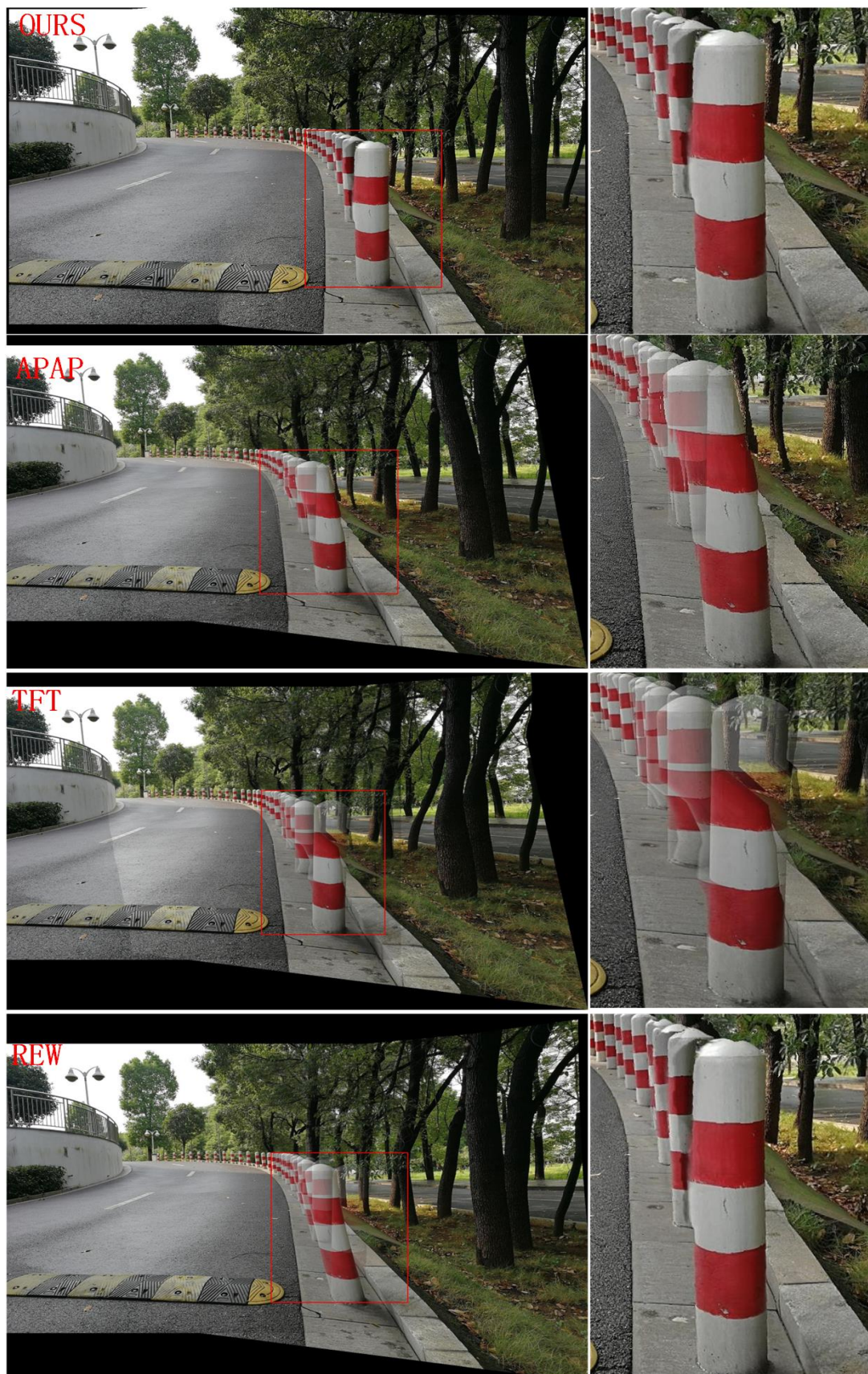


**Figure A6.** Qualitative comparisons of local alignment on the worktable image pair. The red boxes indicate the regions where the enlarged images are located in the mosaics.



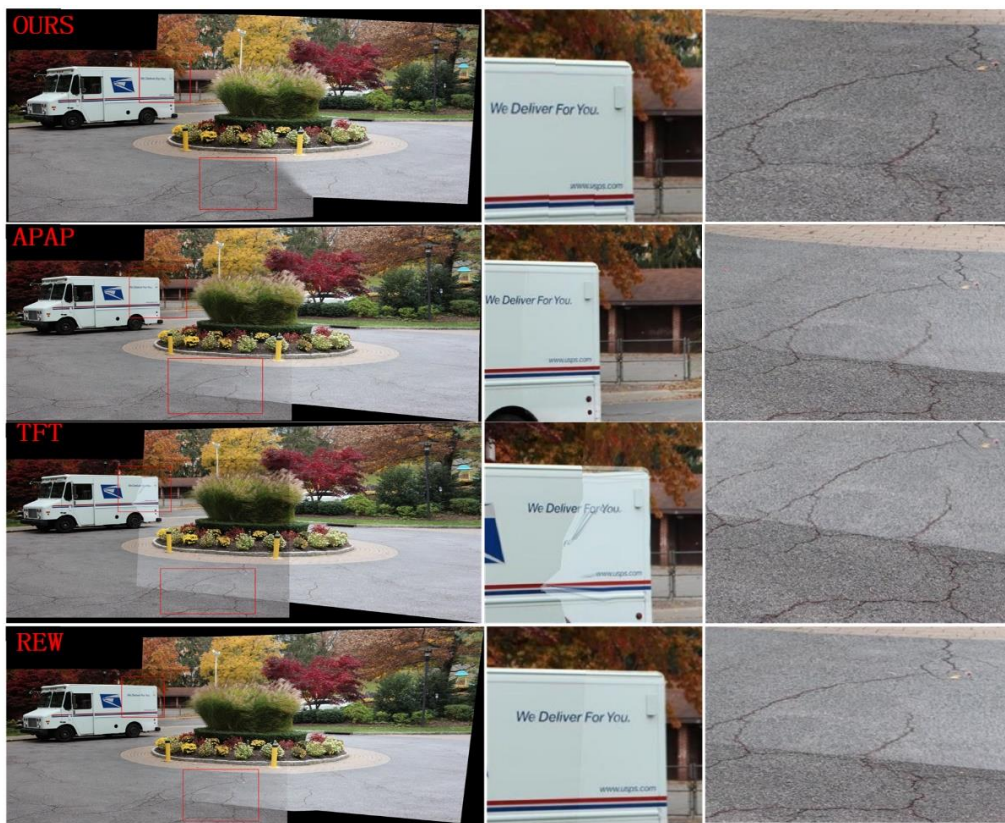
**Figure A7.** Qualitative comparisons of local alignment on the temple image pair. The red boxes indicate the regions where the enlarged images are located in the mosaics.





**Figure A8.** Qualitative comparisons of local alignment on the guardbar image pair. The red boxes indicate the regions where the enlarged images are located in the mosaics.



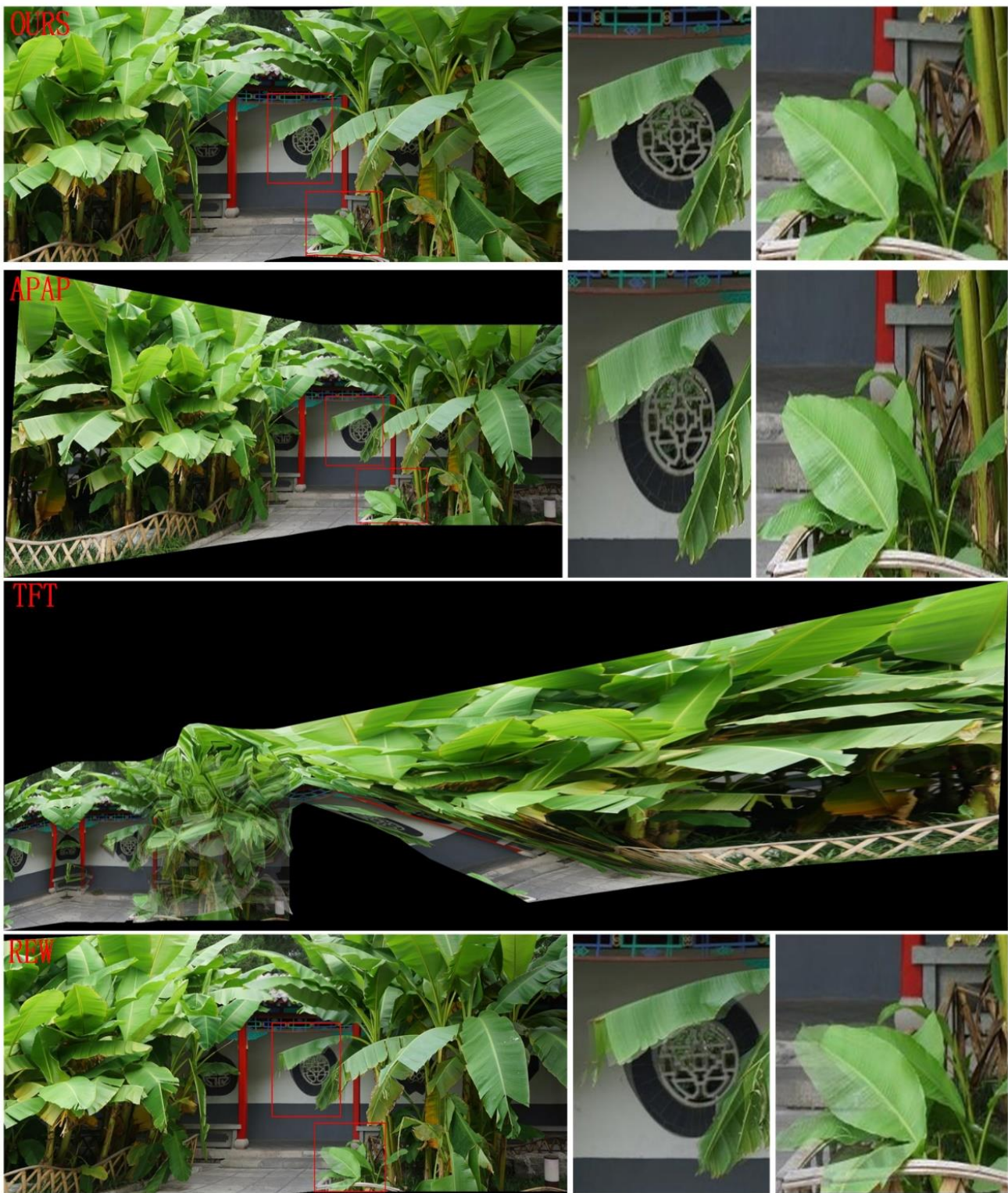


**Figure A9.** Qualitative comparisons of local alignment on the roundabout image pair. The red boxes indicate the regions where the enlarged images are located in the mosaics.

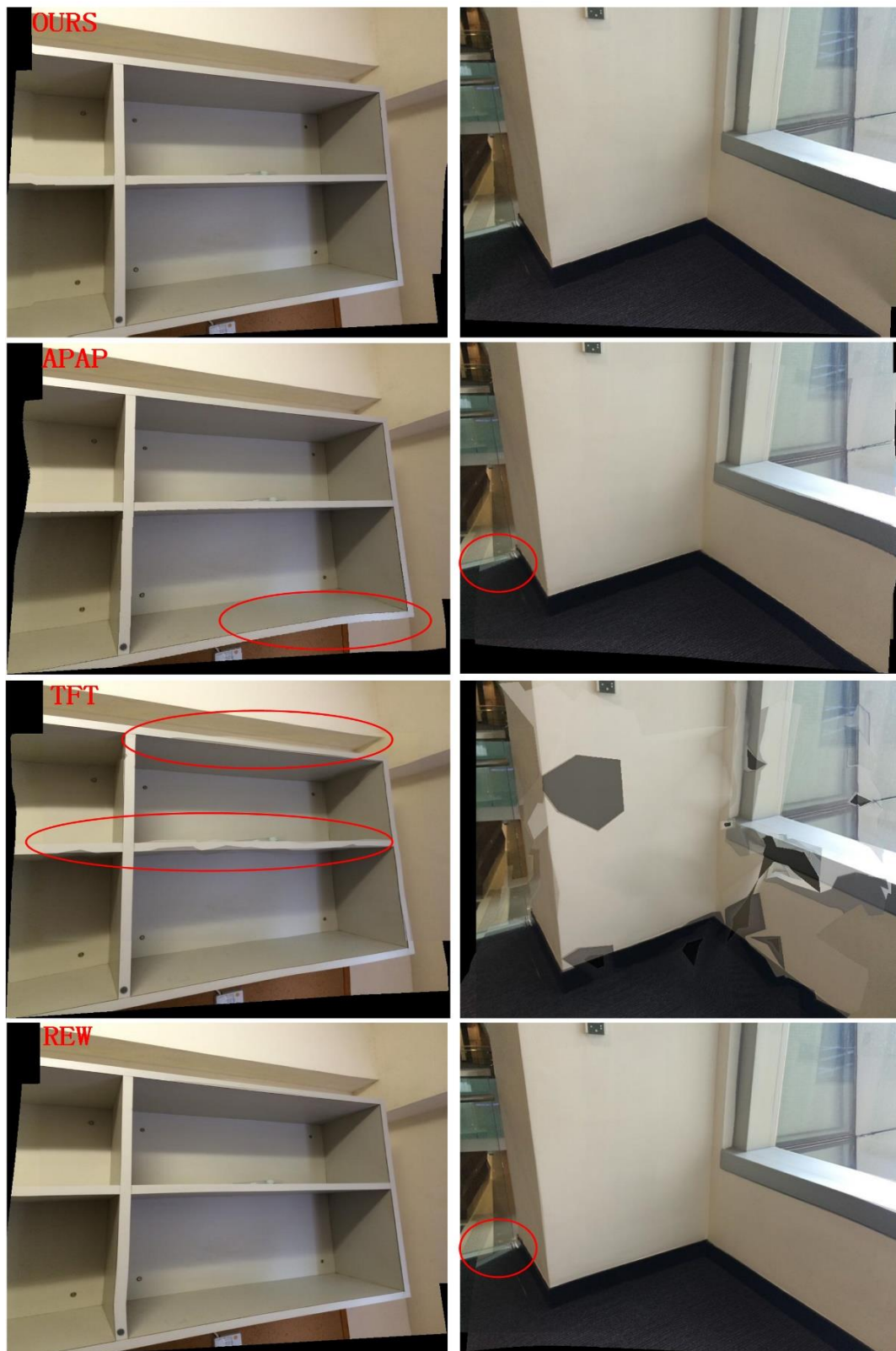


**Figure A10.** Qualitative comparisons of local alignment on the potberry image pair. The red boxes indicate the regions where the enlarged images are located in the mosaics.





**Figure A11.** Qualitative comparisons of local alignment on the plantain image pair. TFT failed to stitch this image pair. The red boxes indicate the regions where the enlarged images are located in the mosaics.



**Figure A12.** Qualitative comparisons of local alignment on the shelf and corner image pairs. TFT failed to stitch the corner image pair. The Red circles highlight errors and distortions.

The scores of all methods on the image pairs roundabout, fence, railtracks, temple, corner, shelf, standing-he, foundation, guardbar, office, plantain, building4, potberry, lawn and worktable are listed in Table A3. The best SSIM value is highlighted in bold.



Table A3. Comparison on SSIM.

	APAP	AANAP	SPHP	TFT	REW	SPW	Ours
roundabout	0.85	0.86	0.77	0.86	0.86	0.76	<b>0.87</b>
fence	0.93	<b>0.95</b>	0.81	<b>0.95</b>	<b>0.95</b>	0.93	<b>0.95</b>
railtracks	0.77	0.90	0.62	0.92	0.85	0.77	<b>0.94</b>
temple	0.90	0.91	0.73	0.95	0.94	0.85	<b>0.96</b>
corner	<b>0.98</b>	0.97	0.91	0.73	<b>0.98</b>	0.97	0.96
shelf	<b>0.98</b>	<b>0.98</b>	0.84	0.95	0.97	0.96	0.97
standing-he	0.72	0.77	0.64	0.35	0.78	0.71	<b>0.84</b>
foundation	0.75	0.78	0.58	<b>0.83</b>	0.76	0.71	0.80
guardbar	0.74	0.74	0.58	<b>0.79</b>	0.77	0.65	0.76
office	0.79	0.78	0.55	0.84	0.65	0.75	<b>0.88</b>
plantain	0.85	0.85	0.67	0.31	0.82	0.85	<b>0.90</b>
building4	0.71	0.72	0.58	0.73	0.74	0.70	<b>0.78</b>
potberry	0.89	0.89	0.68	<b>0.93</b>	0.87	0.82	0.91
lawn	0.92	<b>0.95</b>	0.79	<b>0.95</b>	<b>0.95</b>	0.93	<b>0.95</b>
worktable	0.87	0.84	0.59	0.86	<b>0.97</b>	0.85	<b>0.97</b>

Figure A13 shows the speed of the SC-AOF method versus the APAP, AANAP, SPHP, TFT, REW and SPW methods. The same image pairs as in the SSIM comparison are used in this experiment.

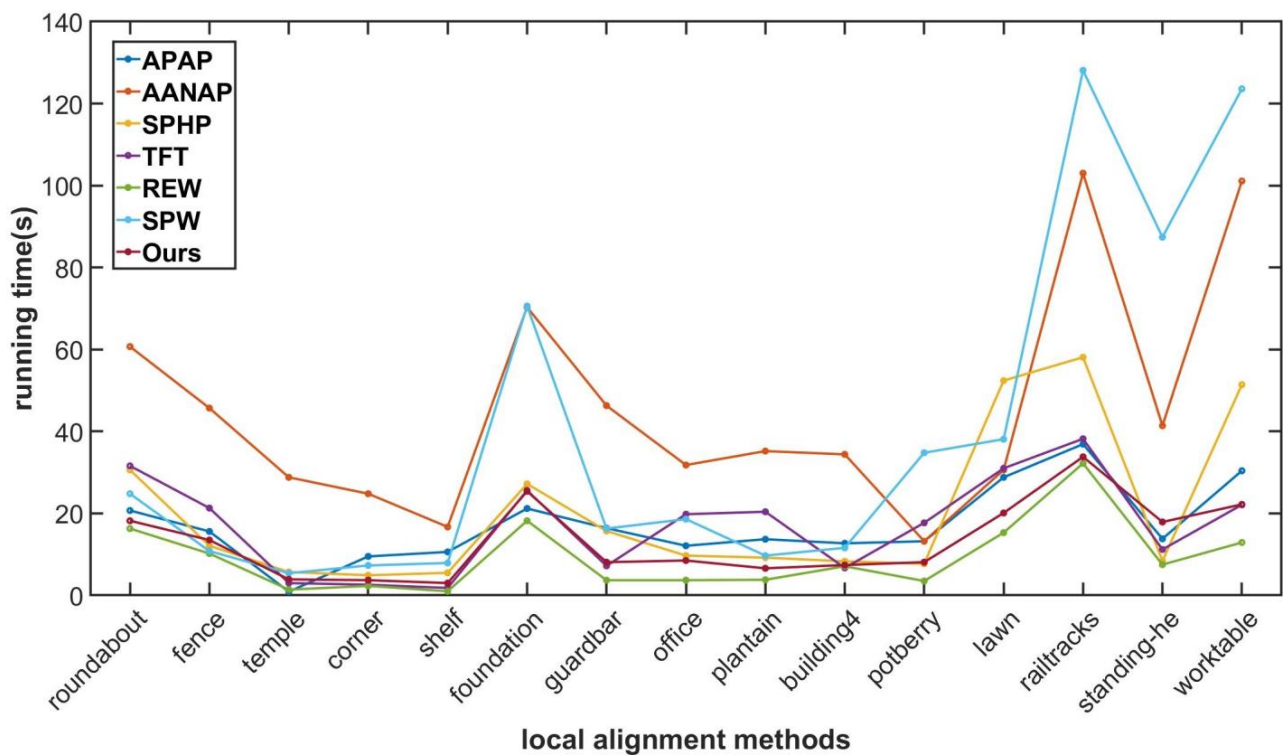


Figure A13. Comparison on elapsed time. Our method is only second to REW in speed and is superior to other methods.



## References

1. Abbadi, N.K.E.L.; Al Hassani, S.A.; Abdulkhaleq, A.H. A review over panoramic image stitching techniques. *J. Phys. Conf. Ser.* **2021**, *1999*, 012115. [[CrossRef](#)]
2. Gómez-Reyes, J.K.; Benítez-Rangel, J.P.; Morales-Hernández, L.A.; Resendiz-Ochoa, E.; Camarillo-Gomez, K.A. Image mosaicing applied on UAVs survey. *Appl. Sci.* **2022**, *12*, 2729. [[CrossRef](#)]
3. Xu, Q.; Chen, J.; Luo, L.; Gong, W.; Wang, Y. UAV image stitching based on mesh-guided deformation and ground constraint. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 4465–4475. [[CrossRef](#)]
4. Wen, S.; Wang, X.; Zhang, W.; Wang, G.; Huang, M.; Yu, B. Structure Preservation and Seam Optimization for Parallax-Tolerant Image Stitching. *IEEE Access* **2022**, *10*, 78713–78725. [[CrossRef](#)]
5. Tang, W.; Jia, F.; Wang, X. An improved adaptive triangular mesh-based image warping method. *Front. Neuroinformatics* **2023**, *16*, 1042429. [[CrossRef](#)]
6. Li, J.; Deng, B.; Tang, R.; Wang, Z.; Yan, Y. Local-adaptive image alignment based on triangular facet approximation. *IEEE Trans. Image Process.* **2019**, *29*, 2356–2369. [[CrossRef](#)] [[PubMed](#)]
7. Lee, K.Y.; Sim, J.Y. Warping residual based image stitching for large parallax. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–18 June 2020; pp. 8198–8206.
8. Zhu, S.; Zhang, Y.; Zhang, J.; Hu, H.; Zhang, Y. ISGTA: An effective approach for multi-image stitching based on gradual transformation matrix. *Signal Image Video Process.* **2023**, *17*, 3811–3820. [[CrossRef](#)]
9. Zaragoza, J.; Chin, T.J.; Brown, M.S.; Suter, D. As-Projective-As-Possible Image Stitching with Moving DLT. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 25–27 June 2013.
10. Li, J.; Wang, Z.; Lai, S.; Zhai, Y.; Zhang, M. Parallax-tolerant image stitching based on robust elastic warping. *IEEE Trans. Multimed.* **2017**, *20*, 1672–1687. [[CrossRef](#)]
11. Xue, F.; Zheng, D. Elastic Warping with Global Linear Constraints for Parallax Image Stitching. In Proceedings of the 2023 15th International Conference on Advanced Computational Intelligence (ICACI), Seoul, Republic of Korea, 6–9 May 2023; pp. 1–6.
12. Liao, T.; Li, N. Natural Image Stitching Using Depth Maps. *arXiv* **2022**, arXiv:2202.06276.
13. Cong, Y.; Wang, Y.; Hou, W.; Pang, W. Feature Correspondences Increase and Hybrid Terms Optimization Warp for Image Stitching. *Entropy* **2023**, *25*, 106. [[CrossRef](#)]
14. 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, OH, USA, 23–28 June 2014; pp. 3254–3261.
15. Chen, J.; Li, Z.; Peng, C.; Wang, Y.; Gong, W. UAV image stitching based on optimal seam and half-projective warp. *Remote Sens.* **2022**, *14*, 1068. [[CrossRef](#)]
16. Lin, C.-C.; Pankanti, S.U.; Ramamurthy, K.N.; Aravkin, A.Y. Adaptive as-natural-as-possible image stitching. In Proceedings of the Computer Vision & Pattern Recognition, Boston, MA, USA, 7–10 June 2015. [[CrossRef](#)]
17. Chen, Y.; Chuang, Y. Natural Image Stitching with the Global Similarity Prior. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Berlin/Heidelberg, Germany, 2016.
18. Cui, J.; Liu, M.; Zhang, Z.; Yang, S.; Ning, J. Robust UAV thermal infrared remote sensing images stitching via overlap-prior-based global similarity prior model. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *14*, 270–282. [[CrossRef](#)]
19. Liao, T.; Li, N. Single-perspective warps in natural image stitching. *IEEE Trans. Image Process.* **2019**, *29*, 724–735. [[CrossRef](#)]
20. Li, N.; Xu, Y.; Wang, C. Quasi-homography warps in image stitching. *IEEE Trans. Multimed.* **2017**, *20*, 1365–1375. [[CrossRef](#)]
21. Du, P.; Ning, J.; Cui, J.; Huang, S.; Wang, X.; Wang, J. Geometric Structure Preserving Warp for Natural Image Stitching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 3688–3696.
22. Bertel, T.; Campbell, N.D.F.; Richardt, C. Megaparallax: Casual 360 panoramas with motion parallax. *IEEE Trans. Vis. Comput. Graph.* **2019**, *25*, 1828–1835. [[CrossRef](#)]
23. Meng, M.; Liu, S. High-quality Panorama Stitching based on Asymmetric Bidirectional Optical Flow. In Proceedings of the 2020 5th International Conference on Computational Intelligence and Applications (ICCIA), Virtual, 19–21 June 2020; pp. 118–122.
24. Hofinger, M.; Bulò, S.R.; Porzi, L.; Knapitsch, A.; Pock, T.; Kotschieder, P. Improving optical flow on a pyramid level. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer International Publishing: Cham, Switzerland, 2020; pp. 770–786.
25. Shah, S.T.H.; Xiang, X. Traditional and modern strategies for optical flow: An investigation. *SN Appl. Sci.* **2021**, *3*, 289. [[CrossRef](#)]
26. Zhai, M.; Xiang, X.; Lv, N.; Kong, X. Optical flow and scene flow estimation: A survey. *Pattern Recognit.* **2021**, *114*, 107861. [[CrossRef](#)]
27. Liu, C.; Yuen, J.; Torralba, A. Sift flow: Dense correspondence across scenes and its applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 978–994. [[CrossRef](#)]
28. Zhao, S.; Zhao, L.; Zhang, Z.; Zhou, E.; Metaxas, D. Global matching with overlapping attention for optical flow estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 17592–17601.
29. Rao, S.; Wang, H. Robust optical flow estimation via edge preserving filtering. *Signal Process. Image Commun.* **2021**, *96*, 116309. [[CrossRef](#)]

30. Jeong, J.; Lin, J.; Porikli, F.; Kwak, N. Imposing consistency for optical flow estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 3181–3191.
31. Anderson, R.; Gallup, D.; Barron, J.T.; Kontkanen, J.; Snavely, N.; Hernández, C.; Agarwal, S.; Seitz, S.M. Jump: Virtual reality video. *ACM Trans. Graph. (TOG)* **2016**, *35*, 1–13. [[CrossRef](#)]
32. Teed, Z.; Deng, J. Raft: Recurrent all-pairs field transforms for optical flow. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part II 16. Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 402–419.
33. Huang, Z.; Shi, X.; Zhang, C.; Wang, Q.; Cheung, K.C.; Qin, H.; Dai, J.; Li, H. Flowformer: A transformer architecture for optical flow. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer Nature: Cham, Switzerland, 2022; pp. 668–685.
34. Available online: <https://github.com/facebookarchive/Surround360> (accessed on 1 January 2022).
35. Zhang, Y. Camera calibration. In *3-D Computer Vision: Principles, Algorithms and Applications*; Springer Nature: Singapore, 2023; pp. 37–65.
36. Zhang, Y.; Zhao, X.; Qian, D. Learning-Based Framework for Camera Calibration with Distortion Correction and High Precision Feature Detection. *arXiv* **2022**, arXiv:2202.00158.
37. Fang, J.; Vasiljevic, I.; Guizilini, V.; Ambrus, R.; Shakhnarovich, G.; Gaidon, A.; Walter, M.R. Self-supervised camera self-calibration from video. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 8468–8475.
38. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.