Pedestrian-Aware Panoramic Video Stitching Based on a Structured Camera Array

ANQI ZHU, LIN ZHANG, and JUNTAO CHEN, School of Software Engineering, Tongji University, China
YICONG ZHOU, Department of Computer and Information Science, University of Macau, China

The panorama stitching system is an indispensable module in surveillance or space exploration. Such a system enables the viewer to understand the surroundings instantly by aligning the surrounding images on a plane and fusing them naturally. The bottleneck of existing systems mainly lies in alignment and naturalness of the transition of adjacent images. When facing dynamic foregrounds, they may produce outputs with misaligned semantic objects, which is evident and sensitive to human perception. We solve three key issues in the existing workflow that can affect its efficiency and the quality of the obtained panoramic video and present Pedestrian360, a panoramic video system based on a structured camera array (a spatial surround-view camera system). First, to get a geometrically aligned 360° view in the horizontal direction, we build a unified multi-camera coordinate system via a novel refinement approach that jointly optimizes camera poses. Second, to eliminate the brightness and color difference of images taken by different cameras, we design a photometric alignment approach by introducing a bias to the baseline linear adjustment model and solving it with two-step least-squares. Third, considering that the human visual system is more sensitive to high-level semantic objects, such as pedestrians and vehicles, we integrate the results of instance segmentation into the framework of dynamic programming in the seam-cutting step. To our knowledge, we are the first to introduce instance segmentation to the seam-cutting problem, which can ensure the integrity of the salient objects in a panorama. Specifically, in our surveillance oriented system, we choose the most significant target, pedestrians, as the seam avoidance target, and this accounts for the name *Pedestrian360*. To validate the effectiveness and efficiency of Pedestrian360, a large-scale dataset composed of videos with pedestrians in five scenes is established. The test results on this dataset demonstrate the superiority of Pedestrian360 compared to its competitors. Experimental results show that Pedestrian360 can stitch videos at a speed of 12 to 26 fps, which depends on the number of objects in the shooting scene and their frequencies of movements. To make our reported results reproducible, the relevant code and collected data are publicly available at https://cslinzhang.github.io/Pedestrian360-Homepage/.

CCS Concepts: • **Computing methodologies** → **Graphics input devices**; **Camera calibration**; *Scene understanding;*

This study was supported in part by the National Key Research and Development Project under Grant 2020YFB2103900, in part by the National Natural Science Foundation of China under Grant 61973235, in part by the Shanghai Science and Technology Innovation Plan under Grant 20510760400, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by the Fundamental Research Funds for the Central Universities.
Authors' addresses: A. Zhu, L. Zhang (corresponding author), and J. Chen, School of Software Engineering, Tongji University, 1239 Siping Road, Shanghai, 200092, China; emails: {1931555, cslinzhang, 1931554}@tongji.edu.cn; Y. Zhou, Department of Computer and Information Science, University of Macau, Taipa University Road, Macau, China; email: yicongzhou@um.edu.mo.

1551-6857/2021/11-ART136 $15.00
https://doi.org/10.1145/3460511

Additional Key Words and Phrases: Panoramic video stitching, extrinsic calibration, photometric alignment, seam-cutting, instance segmentation

ACM Reference format:
Anqi Zhu, Lin Zhang, Juntao Chen, and Yicong Zhou. 2021. Pedestrian-Aware Panoramic Video Stitching Based on a Structured Camera Array. *ACM Trans. Multimedia Comput. Commun. Appl.* 17, 4, Article 136 (November 2021), 24 pages.
https://doi.org/10.1145/3460511

1 INTRODUCTION

In the fields of surveillance and space exploration, the 360° observation toward the surrounding environment plays a vital role [3, 13, 21]. To help the viewer quickly perceive the environmental visual information, a panorama stitching system that aligns the surrounding images on a plane and fuses them naturally is necessarily called for. Such a system presents a consistent horizontal view and allows the viewer to be unaware of the transition between different images.

Different from image stitching, video panorama stitching requires more strict real-time processing ability and consideration of dynamic foregrounds to obtain a consistent wide **field-of-view (FOV)** video efficiently. Three key issues need to be addressed when building a panorama system. The first one is how to align the images captured under different poses. The second one is the photometric alignment among the images with varying levels of brightness and colors. The last one is how to find a seam that bypasses sensitive areas for the **human visual system (HVS)**.

It is necessary to align the images to make them consistent in the overlapping area because the images that make up the panorama are the 2D projections of the scene in different camera coordinate systems. Directly stitching the images will cause serious image distortion and cannot meet the visual consistency. There are mainly two ways to perform image alignment. One is to align the source image with the target image by image warping [3, 13, 50], which often includes feature point matching or local mesh-based deformation. The other is to map the images to be stitched to a standard surface. Compared with the warping-based method, it is more efficient to map all images to a standard surface with a mapping table, which is suitable for structured camera arrays. Hence, the panorama video stitching system constructed in this article is also based on standard surface mapping. Specifically, a unified cylindrical coordinate system to project all images to the cylindrical surface is established, with which a horizontal 360° view with consistent visual appearance can be synthesized.

To build a unified cylindrical coordinate system for a structured camera array, the poses of the cameras should be pre-calibrated to estimate the transformation among different images. The intrinsic and extrinsic parameters of the cameras on a structured camera array are estimated in the calibration step. Note that since intrinsic calibration techniques are now quite mature [55], we focus on the extrinsic calibration procedure in this article. In addition to estimating the relative pose between adjacent cameras, we also perform joint optimization to eliminate accumulated errors and coordinate mapping with the cylindrical model to build a unified cylindrical coordinate system.

In addition to the geometric alignment based on camera calibration, the photometric alignment is also an essential issue for obtaining a visually consistent panorama. Images taken by different cameras may have different brightness levels and colors due to different shooting angles and lighting conditions, which leads to noticeable differences in luminosity on both sides of the seam. Photometric alignment has been investigated in the literature for multi-camera image stitching [4] and stereoscopic 3D image rendering [12]. The linear model [4], which only uses a photometric parameter to adjust the brightness of the images, is not flexible enough to correct the color difference completely. Spatial neighborhood filtering-based methods [43, 46] are computationally intensive.

Fig. 1. Comparison of different seam-cutting algorithms. The upper images are the result of a visual saliency-based scheme [29] and corresponding visual saliency maps of images to be stitched. The visual saliency strategy failed in such a panorama scene, and the seam cannot avoid the pedestrians. The bottom images are the result of our instance segmentation-based scheme, which can accurately segment the pedestrians and ensure their integrity.

We introduce a bias to the baseline linear model, which not only improves the robustness but also ensures the efficiency of the algorithm.

After the cylindrical projection step for image alignment, the dynamic foreground may cause misalignment and the ghost phenomenon. To avoid such a ghost phenomenon, the overlapping area in adjacent images is divided into two parts with a seam. Each part comes from one image to be stitched. The seam-cutting algorithm amounts to finding a seam in the overlapping area of the source image and the target image by minimizing the energy function, which is designed to make the seam pass through the pixel with the smallest difference between the two images. Then the image fusion is performed around the seam. Approaches solving the problem of optimal seam search roughly fall into two categories: **dynamic programming (DP)**-based ones [8, 18] and **graph-cut (GC)** based ones [21, 24, 29, 30]. Compared to GC-based solutions, DP-based ones have the merit of low complexities. Thus, our panoramic video system also resorts to a DP-based framework for seam-cutting to ensure its low computational complexity.

Seam-cutting can be regarded as a pixel-level optimization problem, but the HVS is more sensitive to high-level semantic objects, such as pedestrians and vehicles. Although the existing methods have introduced visual saliency [29], they may fail in complex panorama scenarios. As shown in Figure 1, a visual saliency-based method [29] failed in such a panorama scene, where the seam cannot avoid the pedestrians. Instead, we notice that the fields of semantic segmentation and instance segmentation are developing rapidly in recent years. Their results are more accurate and robust in complex scenes compared with visual saliency prediction methods. Based on the preceding considerations, we design a novel seam-cutting algorithm that integrates the results of instance segmentation into the framework of DP, which can make seams avoid high-level semantic objects. In our surveillance-oriented system, we choose the most significant target, the pedestrian, as the avoidance target for the seam, because for most end users, when there are broken pedestrians in

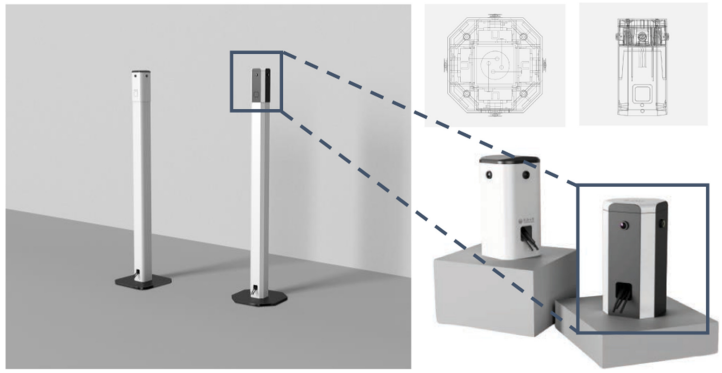

Fig. 2. The appearance and the detailed internal structure of our panoramic video system Pedestrian360. The two kinds of color schemes share the same internal structure. Four fisheye cameras are mounted on a column facing four directions. Each camera has a 180° FOV. Adjacent cameras share overlapping FOVs, and four fisheye cameras cover the column's whole surrounding area. There are two installation modes for Pedestrian360: one is the floor-standing mode (left picture), and the other is the desktop mode (right picture).

the panoramic video, they will deem that the video is of poor quality. For other applications, such as wildlife-oriented surveillance, we only need to change the segmentation target of the semantic segmentation module to obtain the mask of wildlife. Similarly, we can also achieve simultaneous consideration of multiple objects via multi-target segmentation. Our system ensures the integrity of the pedestrians when they pass through the overlapping area of adjacent cameras, and accordingly, it is named as *Pedestrian360*.

The main contributions of this work can be summarized as follows:

- To build a unified cylindrical coordinate system for a structured camera array, we establish a novel refinement approach by introducing the closed-loop optimization for the calibration and perform coordinate mapping with the cylindrical model. Such a unified multi-camera coordinate system can build a geometrically aligned 360° view in the horizontal direction.
- To eliminate the brightness and color difference of images taken by different cameras, we propose a photometric alignment approach for the closed-loop four-image alignment task by introducing a bias to the baseline linear adjustment model and solving it with two-step least-squares. The resulting model not only improves the robustness of the photometric alignment but also has the advantage of high computational efficiency.
- We consider the high-level semantic information that is sensitive to HVS by incorporating the results of instance segmentation into the framework of DP in the seam-cutting step. To our knowledge, we are the first to introduce instance segmentation to the seam-cutting pipeline, which ensures the integrity of salient objects in a panorama.
- By integrating the aforementioned three contributions, Pedestrian360, a panoramic video system based on a structured camera array, is established, whose appearance and internal structure are illustrated in Figure 2. To corroborate its effectiveness and efficiency, extensive experiments have been conducted. All of the codes and the collected dataset used in our studies related to camera calibration and panoramic stitching have been released to the community, which will facilitate the relevant studies.

2 RELATED WORK

In this section, we will first introduce the related work of both image and video panorama stitching, then present the studies in subtasks of panorama stitching, including camera calibration,

photometric alignment, and seam-cutting. In addition, since Pedestrian360 involves an instance segmentation model, we will also introduce representative works in the field of semantic and instance segmentation and alternative visual saliency schemes.

2.1 Image and Video Panorama Stitching

There are mainly two ways to perform image alignment for image stitching. One is to apply image warping to align the source image with the target image, such as feature point matching and local mesh-based deformation. The other is to map all of the images to be stitched to a standard surface.

The earliest image warping approaches were based on global transformation, and then the approaches based on the multi-plane assumption were derived. Today, most state-of-the-art image warping approaches are based on mesh deformation. Global transformation deforms and aligns the images by adopting the same transformation models (e.g., projective, or affine). Typical studies depending on global transformation include the work of Brown and Lowe [3, 4]. They estimate a global warp for alignment between the source image and the target image. This global warp is devoted to minimizing the alignment errors between overlapping pixels via one uniform global transformation (mainly homography), which is often not flexible enough. To increase the deformability of the warp, Gao et al. [10] and Lin et al. [33] divided an image plane into multiple planes, each plane corresponding to a transformation. These methods perform well on images with simple scene structures but may fail in more complex situations. The mesh-based alignment scheme with better local alignment capability is a breakthrough to the image stitching in the early days. Specifically, the images are divided into uniform meshes, each mesh corresponding to a transformation. Zaragoza et al. [50] first introduced mesh-based alignment into image stitching, and the resulting solution is referred to as "APAP" (As-Projective-As-Possible), which locally preserves the salient structures in the non-overlapping regions and reduces the distortions caused by warping in the overlapping region. Due to the excellent performance of APAP, numerous later research has presented different optimization strategies based on various prior constraints [6, 31, 52] for mesh-based alignment. Image warping research currently focuses more on image alignment and naturalness, lacking real-time performance due to the involved time-consuming feature point matching or mesh optimization operations. Thus, they are not suitable for time-critical panoramic video stitching. Learning-based approaches [22, 28] model the image alignment as an end-to-end problem. Their application scenarios are limited by the training datasets, implying their lack of generalization ability.

Mapping the images to be stitched to a standard surface is another way to perform image alignment. The methods of this branch are suitable for structured camera arrays, where a suitable geometric model is selected for the projection of the unified coordinate system. Geometric models can be designed or selected according to deployment environments, such as the bowl-shaped model [51], the boat-shaped model [11], the burger model [53], the spherical model [25], and the cylindrical model [49]. Among them, the cylindrical model is the most commonly used in panoramic stitching. After the images to be stitched are mapped to the geometric model, the stitching result can be obtained via image fusion [5]. Note that geometric model mapping can generate the mapping table directly. Each frame of the panoramic video can be generated according to this mapping table during the stitching process, which is very helpful for the time performance of the final system.

Video stitching can be regarded as image stitching for every individual frame. Liu et al. [34] simply used the stitching model derived from the first few frames to stitch the following frames and did not consider the moving foregrounds. In contrast, Tennøe et al. [44] and Hu et al. [18] updated the seam in every frame, which is computationally inefficient. To balance between suppressing the artifacts and the real-time requirement, He and Yu [13] proposed to first detect changes around

the previous seams and only perform seam update when there are moving objects across seams. Kang et al. [21] proposed a seam consistency module to prevent sudden movement between consecutive stitched frames by penalizing the horizontal offset of seams in adjacent frames, which will however lead to inflexibility of the seam. Lee et al. [25] proposed a stitching method that employs a deformable spherical projection surface where calibrated videos are projected with minimal parallax artifacts. However, with the scheme of Lee et al. [25], it is time consuming to generate the optimized projection surface.

2.2 Calibration and Photometric Alignment of a Multi-Camera System

In structured camera arrays, the relative pose between any pair of cameras is fixed and can be pre-calibrated to estimate the transformation among different images. The intrinsic and extrinsic calibration constitute the calibration of a multi-camera system. This section draws focus on extrinsic calibration. According to the types of the features used, the existing methods can be divided into two categories: interest point-based ones and pattern-based ones.

The interest point-based approaches estimate the camera parameters using the interest points extracted from real scene images [7, 16, 17]. However, interest points are hard to track to obtain point correspondences between images from adjacent cameras because overlapping areas are severely distorted due to the use of fisheye lenses. Thus, interest point-based approaches are not suitable for obtaining accurate calibration parameters.

The pattern-based approaches estimate camera parameters using special patterns, including corners, circles, or lines. Since the configurations of these precisely drawn patterns are known, it is possible to estimate the camera parameters accurately. This kind of calibration scheme relies on placing calibration patterns in overlapping FOVs of the cameras [15, 38, 45, 54]. However, no further refinement is performed to guarantee high-precision of calibration parameters, which consequently leads to the accumulated errors in calibration results.

Photometric alignment has been investigated in the literature for multi-camera image stitching [4] and stereoscopic 3D image rendering [12]. Brown and Lowe [4] exploited a linear model to correct the photometric misalignment for panoramic image stitching. This linear model uses only one photometric adjustment parameter for each image, namely the overall gain, to adjust the brightness of the images, which is not flexible enough to correct the difference completely. Liu et al. [35] matched the intensity histograms of two images to compensate for the brightness difference. However, this technique does not generalize well to the cases where adjacent cameras share limited common FOVs. Suen et al. [43] and Uylttendaele et al. [46] proposed locally adaptive photometric correction methods. Since such methods require spatial neighborhood filtering at each pixel, they are computationally intensive and thereby are not suitable for time-critical systems.

2.3 Seam-Cutting

Seam-cutting, also called *optimal seam search*, is to find a seam in the overlapping area of the source image and the target image. Usually, it is achieved by minimizing the energy function, which is designed to make the seam pass through the pixel with the smallest difference between the two images. Optimal seam search strategies can be roughly divided into two categories: DP-based ones [8, 18] and GC-based ones [21, 24, 29, 30].

DP-based seam-cutting treats each row (column) as one step and searches for the optimal path in a step-by-step manner, which is memoryless and can be easily implemented [8]. Considering that the slope of the seam calculated by the traditional DP algorithm is limited—that is, the stitching points of two adjacent lines can only be offset by 1 pixel—Hu et al. [18] proposed a discontinuous seam-cutting algorithm so that the stitching points of two adjacent lines in the final optimal seam can be offset arbitrarily. Although DP-based schemes can only find horizontal or vertical seams,

they are still applicable in specific situations such as horizontal image stitching due to their high computational efficiency. For the overlapping area with n pixels, the complexity of the DP-based algorithms is $O(n)$.

The core idea of GC-based seam-cutting pipelines to find the optimal seam [24] is to construct an undirected graph based on the image. The vertices in the graph correspond to pixels in the overlapping area, and the edges correspond to the relationships between adjacent pixels. The cost of each edge is calculated according to the definition of the energy function. The most straightforward energy function is a measure of color difference between the pairs of pixels. The more similar the two pixels connected by the edge are, the better the final stitching visual effect is. The GC algorithm is adopted to minimize the energy function and solve the pathfinding problem. Compared with the DP-based algorithms, the seams identified by GC-based seam-cutting are not limited to horizontals or verticals. Li et al. [29] introduced pixel saliency to the traditional GC-based seam-cutting framework, hoping to avoid the seams that pass through the significant areas to human perception. As expected, the efficacy of this method highly depends on the accuracy of saliency prediction. Nonetheless, the performance of modern saliency prediction algorithms for complex scenes is still far from satisfactory, especially for images like panoramic ones that are usually not considered when training saliency prediction models.

Liao et al. [30] proposed an iterative algorithm, in which the search of the optimal seam is guided by quality evaluation of the current seam. Such an iterative approach is doomed to be extremely time consuming.

Finding the best cut for a graph can have a worst-case $O(n^2)$ cost for a graph with n nodes and its average timings appear to be $O(nlog(n))$ [24]. Thus, the complexity of DP-based methods is lower than that of GC-based ones.

2.4 Visual Saliency Models and Semantic Segmentation

Considering that the HVS is more sensitive to high-level semantic information and deep learning has become a powerful tool to help people understand images [9, 14, 36], here we compare two types of methods for extracting high-level semantics from images.

Visual saliency models mimic the behavior of human beings and capture the most salient regions from images or scenes. Heuristic-based visual saliency models rely on various priors, such as center-prior [19], backgroundness-prior [48], and objectness prior [20]. These priors do not necessarily hold in large-view panoramic images containing a large number of objects. Learning-based visual saliency models [9, 37, 47, 56] with training data are utilized to find salient regions from images with complex backgrounds. The data-driven training scheme greatly limits the generalization capabilities of these models. Consequently, these learning-based schemes usually perform quite well on test images satisfying the conditions on which they were trained. On the contrary, their performance deteriorates significantly once these conditions are not met, such as the cases of large-view panoramic images. In the work of Li et al. [29], visual saliency is introduced to the seam-cutting pipeline to avoid the seams that pass through the significant areas to human perception. But its effectiveness depends on the performance of saliency prediction, which is not trained or optimized on panoramic data.

Semantic segmentation, also called *scene labeling*, refers to the process of assigning a semantic label (e.g., car, person, and road) to each pixel of an image. It is an essential data processing step for robots and other unmanned systems to understand the surrounding scene. According to the current research focus, existing methods can be roughly divided into three main categories: hand-engineer feature-based ones [9, 39], learned feature-based ones [1, 41], and weakly or semisupervised ones [42]. Instance segmentation distinguishes individual information based on semantic segmentation. The representative method in this field is Mask R-CNN [14], which uses a relatively

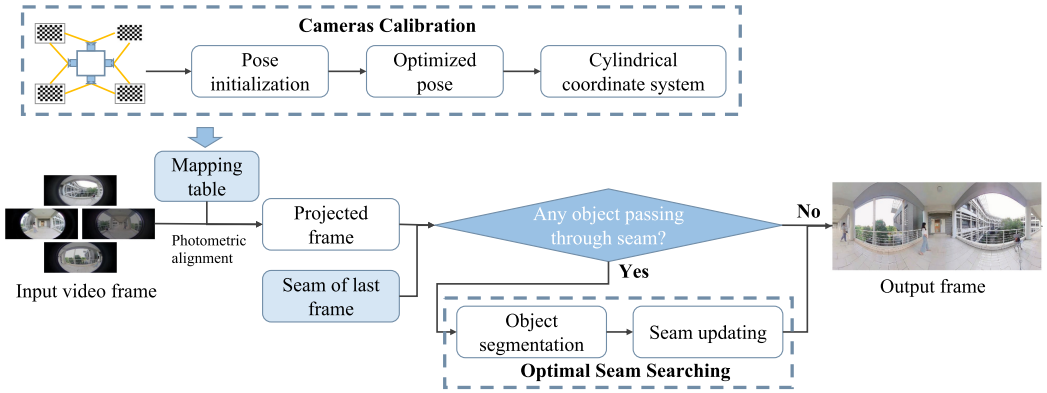

Fig. 3. The video stitching pipeline of Pedestrian360. It consists of three steps: calibration, cylindrical projection with photometric alignment, and seam update. Before video stitching, the spatial surround-view camera system calibration is conducted to establish a unified cylindrical coordinate system. The current frames to be stitched are uniformly projected onto the cylindrical coordinate system for photometric alignment. Whether or not to perform the optimal seam search depends on the changes around the seam of the previous frame.

simple mask predictor to extend the Faster R-CNN [40] detection model. Compared with the field of visual saliency prediction, the field of semantic segmentation is more mature. The existing segmentation methods are more robust to the understanding of complex scenes and more accessible for us to extract the information we want. Our Pedestrian360 is the first to introduce instance segmentation into the stitching pipeline, hoping to provide more accurate and robust guidance for the stitching in large-view panoramic images.

3 PEDESTRIAN360 STITCHING PIPELINE

In this section, we briefly describe the video stitching pipeline of Pedestrian360, as well as the purpose and process of each step. In Sections 4 and 5, the mathematical details of projection and seam-cutting are described, respectively.

Our panoramic video system Pedestrian360 is based on a structured camera array, which consists of four wide-angle, small-sized 1080P cameras. Its appearance and detailed internal structure are illustrated in Figure 2. Four fisheye cameras are mounted on a column facing four directions. Each camera has a 180° FOV. Adjacent cameras share overlapping FOVs and four fisheye cameras together cover the whole surrounding area of the column.

As illustrated in Figure 3, the video stitching pipeline of Pedestrian360 consists of three steps: calibration, cylindrical projection with photometric alignment, and seam update. First, the spatial surround-view camera system should be calibrated to establish a unified cylindrical coordinate system. Next, the four images of the current frame are uniformly projected onto the established cylindrical coordinate system for photometric alignment. Then, similar to the work of He and Yu [13], whether there is any object passing through the seam is judged according to the changes near the seam of the previous frame. If there is an object passing through the seam, the seam needs to be updated by optimal seam searching. Otherwise, the seam of the previous frame will continue to be used. Finally, the panoramic image of the current frame is obtained by image blending around the seam.

To combine the views from multiple cameras, which are tightly structured, it is reasonable to perform the calibration in advance and reuse the mapping table. Considering that the cameras are horizontally outward, the cylinder is selected as the projection model to obtain a 360° horizontal

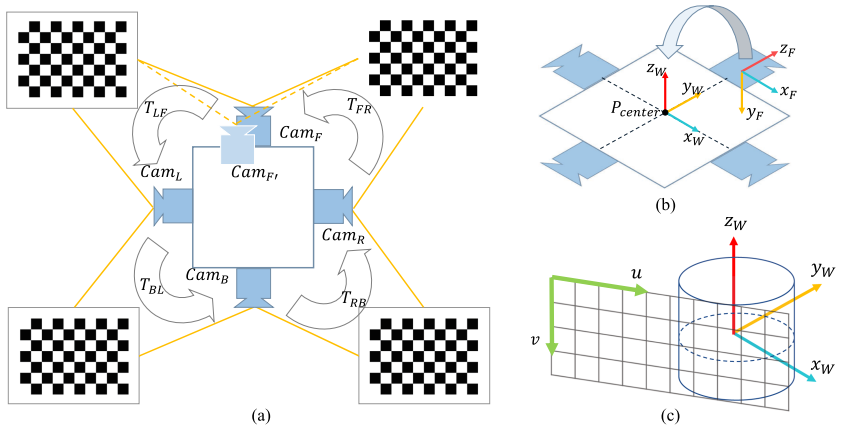

Fig. 4. Calibration of the spatial surround-view camera system and cylindrical projection. (a) Configuration setup for camera poses calibration. (b) Relationship between the front camera coordinate system and the cylindrical coordinate system. (c) Mapping relationship between cylindrical coordinates and pixel coordinates of a panoramic image.

view. The geometric relationship between the front camera coordinate system and the cylindrical coordinate system is characterized by translation and rotation.

In this cylindrical coordinate system, the images to be stitched are projected to the cylinder surface. The photometric alignment can achieve consistent luminosity and color among these projected images, which is conducive to the subsequent seam-cutting and blending.

The seam-cutting step intends to find an unobservable seam in the overlapping region of the aligned images. It can effectively relieve the artifacts generated by local misalignment. Typically, the seam-cutting approach is usually formulated as an energy minimization problem. Here we consider the human perception of specific objects by integrating the results of instance segmentation into the DP framework in the seam-cutting step. The choice of the DP-based method to solve the energy minimization problem is for the consideration of computational efficiency.

4 CYLINDRICAL PROJECTION WITH CONSISTENT BRIGHTNESS AND COLORS

First, the calibration of the spatial surround-view camera system is conducted to establish a unified cylindrical coordinate system for a combination of the views from multiple cameras. Then the photometric alignment is performed in an attempt to achieve consistent luminosity and color among projected images. The geometric alignment and photometric alignment can effectively facilitate subsequent seam-cutting and blending.

4.1 Calibration of the Spatial Surround-View Camera System and Cylindrical Projection

The procedure of getting cylindrical projected images mainly consists of three steps: pose initializing and joint optimizing of the spatial surround-view camera system, establishing the camera-cylinder relationship, and mapping table formulation from the cylindrical projected image to the original fisheye image.

The core idea of our calibration strategy proposed in this article is illustrated in Figure 4(a). By placing the chessboard pattern in the overlapping FOV between adjacent cameras, the relative pose between them can be estimated by making use of epipolar constraints [26]. Here we denote

the relative pose between the front and left cameras by T_{LF}. Similarly, the other three relative poses are denoted by T_{BL}, T_{RB}, and T_{FR}.

Note that by multiplying these transformation matrices, we define a new front camera pose after a closed-loop transformation as

$$T'_{FF} = T_{FR}T_{RB}T_{BL}T_{LF}T_{FF}, \tag{1}$$

where $T_{FF} = I$. In ideal conditions, T'_{FF} should be equal to T_{FF}. However, due to accumulated errors, $T'_{FF} \neq T_{FF}$.

To improve the initial estimation accuracy of camera poses, all camera parameters are refined by bundle adjustment using a graph-based joint optimization method [23]. The triangulated chessboard corners and camera poses are jointly optimized with the aim of minimizing the reprojection errors. The camera poses are represented by the Lie algebra ξ to convert the pose estimation into an unconstrained optimization problem. For each camera pose ξ_i, each triangulated chessboard corner $\boldsymbol{P}_{ij}$ in the ith camera and its corresponding pixel coordinates $\boldsymbol{u}_{ij}$, we sum all reprojection errors up and build up a least-squares minimization problem,

$$\boldsymbol{P}^*, \xi^* = \arg\min_{\boldsymbol{P},\xi} \frac{1}{2} \sum_{i=1}^{5} \sum_{j=1}^{N_i} \left(\boldsymbol{u}_{ij} - \frac{1}{s_{ij}} K_i \exp\left(\xi_i^\wedge\right) \boldsymbol{P}_{ij} \right), \tag{2}$$

where K denotes camera intrinsics and s is the depth. During optimization, the camera poses are updated to conform to $T'_{FF} = T_{FF}$—that is, the accumulated errors are reduced as much as possible. Now we can obtain four optimized camera poses in the front camera coordinate system, which is selected as the reference coordinate system.

To merge the views of multiple cameras and facilitate observation, it is necessary to establish a unified cylindrical coordinate system. As shown in Figure 4(b), the cylindrical coordinate system is established by translating the front camera coordinate system to the center of four cameras and rotating it 90° around the x-axis so that its z-axis is upward. The center coordinates P_{center} of four cameras in the front camera coordinate system can be computed as

$$\begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix} = \frac{1}{4} \left(T_{FF}^{-1} \cdot \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix} + T_{LF}^{-1} \cdot \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix} + T_{BF}^{-1} \cdot \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix} + T_{RF}^{-1} \cdot \begin{matrix} 0 \\ 0 \\ 0 \\ 1 \end{matrix} \right). \tag{3}$$

The transformation matrix from cylindrical coordinates to front camera coordinates, which combines the rotation matrix and the translation vector, is

$$T_{FW} = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 0 & -1 & P_y \\ 0 & 1 & 0 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

At this point, the relationship between the other three cameras coordinates and cylindrical coordinates can be determined as

$$\begin{aligned} T_{LW} &= T_{LF}T_{FW} \\ T_{BW} &= T_{BL}T_{LF}T_{FW} \\ T_{RW} &= T_{RB}T_{BL}T_{LF}T_{FW}. \end{aligned} \tag{5}$$

The last step is to form the mapping table from the cylindrical projected image to the original fisheye image. As shown in Figure 4(c), the pixel coordinates (u, v) in the panorama stitching result

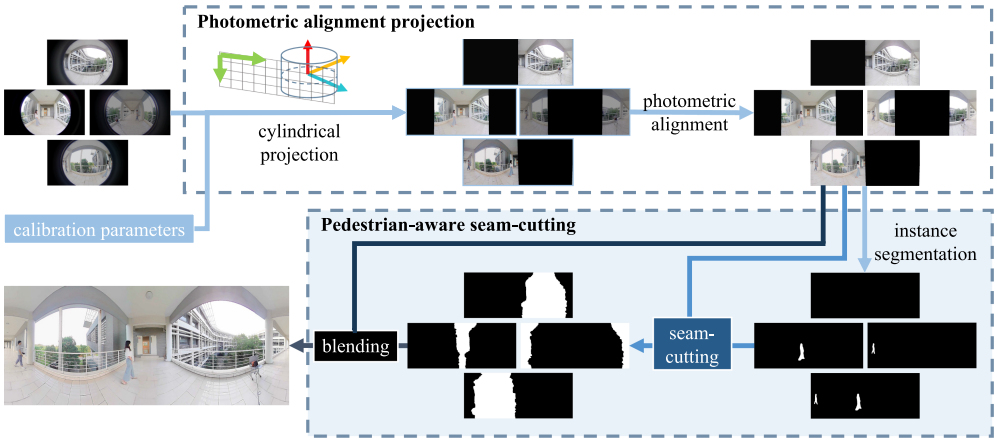

Fig. 5. The image stitching pipeline of Pedestrian360. The input images are mapped to the cylindrical coordinates according to the calibration parameters, and the images with the consistent exposure are obtained by photometric alignment. In seam-cutting, the mask is obtained by instance segmentation to guide the seam to avoid the pedestrians. Finally, the panoramic image is generated by image fusion.

are mapped to cylindrical coordinates (x_w, y_w, z_w) as

$$\begin{aligned} x_w &= \cos(u \cdot w/2\pi) \cdot r \\ y_w &= \sin(u \cdot w/2\pi) \cdot r \\ z_w &= (h/w - v) \cdot h_{scale}, \end{aligned} \tag{6}$$

where r is the radius of the cylinder and h_{scale} is the scale factor in the z-axis. The transformation from the cylindrical coordinates to the original fisheye image pixel coordinates (u_F, v_F) can be expressed as

$$\begin{bmatrix} u_F \\ v_F \end{bmatrix} = K_F \cdot T_{FW} \cdot \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{7}$$

where u_F and v_F are the original fisheye image pixel coordinates. Note that Equation (7) contains two dehomogeneous operations. The coordinates under the other three cameras can be computed similarily. As shown in Figure 5, after camera calibration and cylindrical projection, geometrically consistent cylindrical projected images are produced for subsequent steps.

4.2 Photometric Alignment

Different shooting angles or lighting conditions of different cameras may bring photometric misalignments in the projected images. This will lead to noticeable differences in luminosity on both sides of the seam. We decompose the images into HSV channels and perform photometric alignment on each channel to achieve the purpose of unifying the brightness and color of projected images. Next, the proposed adjustment approach of a single channel is described.

First of all, the intensity adjustment model is required to be determined. The photometric alignment approach proposed in the work of Brown and Lowe [4] is widely used due to its extensible model and low computational complexity. Considering its merits, we choose the intensity adjustment model proposed by Brown and Lowe [4] as the baseline to improve it. Specifically, we introduce a bias term b to the baseline adjustment method based on the gain g. The resulting model is $I' = g \cdot I + b$. With this model, the degree of freedom of image intensity adjustment is increased,

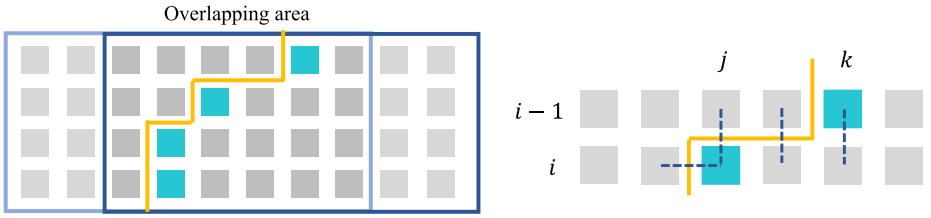

Fig. 6. A seam in the overlapping area of two images and its cost calculation in DP.

so additional constraints can be introduced to the baseline model [4], which only aligns mean intensity values of overlapping areas in adjacent images. Specifically, in this work, the variance alignment is selected as an additional type of constraint. In other words, in addition to the mean values, the variances of overlapping parts of adjacent images are calculated and aligned using the adjustment model.

Denote the mean value and standard deviation of the image i in the overlapping area of image i and image j by m_{ij}, σ_{ij}, and those of the image j by m_{ji}, σ_{ji}. The alignment equations of mean values and standard deviations of overlapping regions are

$$\begin{aligned} g_i \cdot m_{ij} + b_i &= g_j \cdot m_{ji} + b_j \\ g_i \cdot \sigma_{ij} &= g_j \cdot \sigma_{ji}, ij \in \{FR, RB, BL, LF\}, \end{aligned} \tag{8}$$

a total of eight equations. We design a two-step least-squares-based approach to solve this system of equations. The four equations only containing the gains $\{g_i\}$ are solved first. Considering there is no non-zero solution for these equations, it is reasonable to compute the least-squares solution and perform normalization by dividing the mean value of the gains. Then, substituting the result into Equation (8), the following equations about b_i can be obtained,

$$\begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} b_F \\ b_R \\ b_B \\ b_L \end{pmatrix} = \begin{pmatrix} g_R m_{RF} - g_F m_{FR} \\ g_B m_{BR} - g_R m_{RB} \\ g_L m_{LB} - g_B m_{BL} \\ g_F m_{FL} - g_L m_{LF} \end{pmatrix}, \tag{9}$$

which do not have an exact solution. The approximate solution can be obtained by SVD decomposition. After the parameters of the intensity adjustment model are solved, the four projected images are adjusted accordingly to obtain the projection results with the consistent brightness levels and colors for subsequent seam-cutting and blending. As shown in Figure 5, after photometric alignment, photometrically consistent cylindrical projected images are available for seam-cutting.

5 PEDESTRIAN-AWARE SEAM-CUTTING

With four geometrically and photometrically aligned horizontal images, it is required to find four vertical seams. For each vertical seam, the seam-cutting cost for pixels can be computed row by row from one endpoint to the other in a DP manner. As illustrated in Figure 6, one pixel in each row of the overlapping area is selected as the dividing point. The final vertical seam can be obtained by connecting these dividing points. The extraction of a horizontal seam is similar but will operate in a column-by-column fashion. Similar to the work of Hu et al. [18], the state transition function of DP in row i and column j is as follows:

$$M(i, j) = \lambda Sem(i, j) + \min_k (M(i - 1, k) + Spa(i, j, k)), \tag{10}$$

Fig. 7. The segmentation result of the person category by Mask R-CNN [14].

where λ is the weight factor of the proposed semantic cost and k covers all possible column values j in the previous row. In our implementation, for the purpose of saving time, we set $j-6 < k < j+6$. $Sem(i, j)$ is the semantic cost of selecting the pixel (i, j) as the dividing point, whereas $Spa(i, j, k)$ is the spatial cost of the path from $(i-1, k)$ to (i, j). By DP, a path with the minimum total cost can be found.

The calculation of the spatial cost $Spa(i, j, k)$ is illustrated by the dotted line in Figure 6, which actually reflects the differences of each group of pixels that the seam of adjacent rows of pixels pass through. In this work, $Spa(i, j, k)$ is defined as the following form:

$$Spa(i, j, k) = edg([i, j-1], [i, j]) + \sum_{p=j}^{k} edg([i-1, p], [i, p]), \tag{11}$$

where $edg([i_1, j_1], [i_2, j_2])$ describes the difference in two pixels (i_1, j_1) and (i_2, j_2) of the two images as

$$edg([i_1, j_1], [i_2, j_2]) = \frac{(I_1(i_1, j_1) - I_2(i_2, j_2))^2 + I_1(i_2, j_2) - I_2(i_1, j_1))^2}{2}. \tag{12}$$

The semantic cost $Sem(i, j)$ is the binary mask of the semantic object, as illustrated by the instance segmentation result in Figure 5. Here we choose Mask R-CNN [14] pre-trained on the COCO dataset [32] as the segmentation model and adopt the mask of person category. Figure 7 presents two examples of the person category segmentation result. The results of instance segmentation are integrated into the state transition function of DP in an attempt to provide semantic guidance for the seam. So far, the instance segmentation mask is incorporated into the seam-cutting algorithm, resulting in the seams that avoid the pedestrians.

To enable the reader to have a clear and overall understanding of our pedestrian-aware seam-cutting, the algorithmic details are summarized in Algorithm 1. The four input geometrically and photometrically aligned cylindrical projected images of the algorithm can be obtained by offline calibration of the cameras' pose parameters and online photometric alignment. Photometric alignment is carried out at each frame to ensure that the alignment parameters can adapt to the scene's changes. The frequency of photometric alignment can be appropriately reduced to make a tradeoff between time cost and performance.

Our proposed seam-cutting pipeline mainly comprises three stages. First, before the DP, it is determined whether the seams need to be updated according to adjacent frames' difference. If not, the previous frame's seams can be directly inherited as the result of the current frame. The second

ALGORITHM 1: Algorithm of pedestrian-aware seam-cutting

Input: The current frame number t. $t = 0$ denotes the first frame; Four seams of the last frame $seam_{t-1}[i], i = 1, 2, 3, 4$ if $t > 0$; Four geometrically and photometrically aligned cylindrical projected images $I_t[i], i = 1, 2, 3, 4$; Four overlapping area masks $overlap[i], i = 1, 2, 3, 4$, where $overlap(p)$ is the overlapping area of $I_t(p)$ and $I_t(p + 1)$.

Output: Four seams of the current frame $seam_t[i], i = 1, 2, 3, 4$.

for $p = 0 : 4$ **do**
  1. Determine whether or not seam update is required.
  if $t > 0$ and the changes between the last frame and the current frame near $seam_{t-1}(p)$ are smaller than the threshold **then**
    $seam_t(p) \leftarrow seam_{t-1}(p)$;
    continue;
  end if
  2. Forward DP.
  2.1 Generate the instance segmentation masks m_1 and m_2 of $I_t(p)$ and $I_t(p+1)$, respectively. Generate the semantic cost $Sem \leftarrow m_1 \cup m_2$.
  2.2 Compute the state transition function values $M(i, j)$ of the pixels in the first row of $overlap(p)$.
  for all $(i, j) \in overlap(p)$ where $i = 0$ **do**
    $M(i, j) \leftarrow \lambda Sem(i, j) + edg([i, j], [i, j + 1])$, where $edg()$ is defined in Equation (12);
  end for
  2.3 Compute the state transition function values $M(i, j)$ of the pixels in the remaining rows of $overlap(p)$.
  for all $(i, j) \in overlap(p)$ where $i > 0$ **do**
    Search the minimum temporary cost as
    $mt \leftarrow \min_k(M(i - 1, k) + Spa(i, j, k)), where\ k = \max(0, j - 5) : \min(j + 6, c)$;
    Update the last step table as $laststep(i, j) \leftarrow \arg\min_k(M(i - 1, k) + Spa(i, j, k))$, where $Spa()$ is defined in Equation (11);
    Update the state transition function value $M(i, j) \leftarrow \lambda Sem(i, j) + mt$;
  end for
  3. Backward DP.
  3.1 Search the destination end of the optimal path in the last row that minimizes the total cost as $end \leftarrow \arg\min_j M(r - 1, j)$, where $(r - 1, j) \in overlap(p)$.
  3.2 Add the destination $(r - 1, end)$ into $seam_t(p)$.
  3.3 Trace back to find the points passed by the optimal path.
  for $i = r - 1 : 1$ **do**
    $end \leftarrow laststep(i, end)$;
    Add the point $(i - 1, end)$ into $seam_t(p)$;
  end for
end for
return $seam_t[i], i = 1, 2, 3, 4$;

stage, forward DP, is mainly to calculate the state transition function Equation (10). The last stage, backward DP, is to infer the optimal seam from the state transition function.

The main computational complexity lies in the second stage. At step 2.3 in Algorithm 1, each pixel in the overlapping area needs to be traversed. From Figure 5, it can be observed that the

Fig. 8. Stitching examples for five different scenes in our dataset by our Pedestrian360.

overlapping area of adjacent images is about $\frac{1}{4}MN$, where M and N are the number of rows and columns of the panorama, respectively. When traversing each pixel, it is required to traverse k values to calculate the spatial cost from the last row to the current row. In our implementation, there are at most 10 cases for k's values. The preceding operations need to be performed four times for the four overlapping regions, and the final complexity is $O(4 \times \frac{1}{4}MN \times 10)$. After omitting the coefficient, it is $O(MN)$. The time complexity is proportional to the image size, implying that the algorithm can be accelerated by downsampling the image.

6 EXPERIMENTAL RESULTS AND DISCUSSIONS

6.1 Experiment Setup, Benchmark Dataset, and Evaluation Metrics

We evaluated the proposed Pedestrian360 in both indoor and outdoor scenarios by placing the column equipped with the spatial surround-view camera system consisting of four fisheye cameras (refer to Figure 2 for details) in various scenes.

To facilitate the study of panorama stitching, we have established and released a large-scale benchmark dataset. The dataset provides videos with pedestrians in five scenes, including two indoor scenes and three outdoor scenes, as shown in Figure 8. The indoor scenes include the cases of multiple-person walking, up to 4 moveable persons. In the outdoor scenes, up to 10 walking pedestrians appear in the video simultaneously. To increase the diversity and facilitate the storage, we consciously set the frame rate to 10 fps when capturing videos and covered as much variety as possible in short videos. The sample of each scene contains fisheye videos taken in four directions, each with 200 frames. In total, 4,000 ($5 \times 4 \times 200$) images were collected. The resolution of the fisheye cameras is 1920×1080. This dataset was utilized in the evaluation and comparison experiments of photometric alignment and seam-cutting.

In addition, to make the calibration results reproducible, we also provide experimental data for extrinsic calibration. The data was collected by placing a chessboard with 9×6 squares in the common FOVs of adjacent cameras. Each square of the chessboard is 10 cm in length. In each common FOV, two images were taken from adjacent cameras respectively, and eight images were collected from four common FOVs.

Pedestrian360 is implemented with C++, except that Mask R-CNN is implemented in PyTorch and runs on GPU. The two modules communicate with each other through a socket connection. All experiments were carried out on a workstation with a 3.0-GHz Intel Core i7-5960X CPU and an Nvidia GeForce GTX 3080 GPU. In all experiments, we set $\lambda = 1,000$ to make the seam avoid pedestrians as much as possible. The relevant code and collected data are publicly available at https://cslinzhang.github.io/Pedestrian360-Homepage/.

Extensive experiments were carried out on each stage of Pedestrian360 to verify its efficacy, including calibration of the spatial surround-view camera system, photometric alignment, and

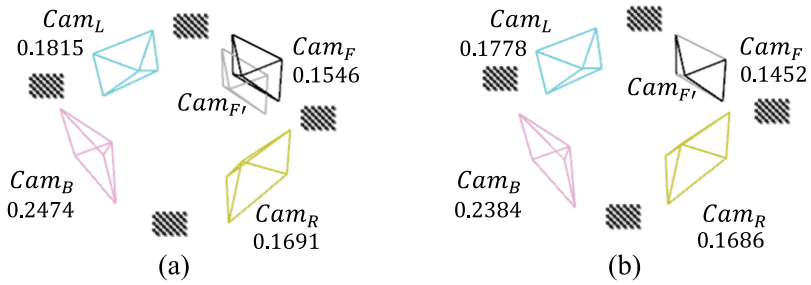

Fig. 9. Camera poses and reprojection errors before/after refinement. (a) Initialized poses. (b) Poses after refinement.

seam-cutting. In addition, Pedestrian360 was compared with competing systems in a system view. Finally, the limitations of Pedestrian360 are discussed for further possible improvement.

In the first experiment, the performance of structured camera array calibration was assessed by a commonly used calibration metric: reprojection error. The reprojection error is the difference between the observed 2D position and the projected 2D position of the corresponding 3D point based on the estimated pose. It is valid in calibration evaluation in a multi-camera system because it can reflect the accuracy and the consistency of camera poses' estimation.

The performance of photometric alignment was assessed by the intensity differences between the overlapping areas of adjacent views. If the brightness levels and colors of adjacent images are well adjusted and unified, the intensity differences of their overlapping areas should be as slight as possible.

The quantitative metric for seam-cutting evaluation proposed in the work of Kang et al. [21] was adopted to demonstrate the advantages of Pedestrian360, which was conducted by counting the number of frames with broken objects. For each stitched panoramic frame, four volunteers were invited to judge whether any pedestrian or object was broken. For each video clip, the mentioned counting operations were performed to the results of every competing approach evaluated. To avoid the subjective bias, all of the stitched frames were randomized and were unknown to subjects. A lower ratio of broken frames means more appealing stitching results for human perception.

6.2 Performance of Structured Camera Array Calibration

In this experiment, we evaluated the performance of the structured camera array calibration and demonstrated the effectiveness of the proposed joint optimization approach. The pose initialization was achieved by making use of epipolar constraints using a chessboard placed in the common FOVs. Then the poses of four cameras were jointly optimized in a closed-loop manner.

Figure 9 shows poses of the structured camera array before and after joint optimization. In Figure 9(a), because of accumulated errors, the new front camera pose $Cam_{F'}$ obtained after a closed-loop transformation deviates from the original pose Cam_F by a large margin, whereas both cameras in Figure 9(b) almost coincide. This indicates that by the proposed joint optimization approach, the accumulated errors of the spatial surround-view camera system can be significantly reduced, and consequently the optimized poses are more accurate than the initial ones.

The quantitative results are presented in Figure 9, including the average reprojection errors of each camera using the initially estimated poses and the poses after joint optimization. A smaller reprojection error typically indicates more accurate pose estimation. It can be observed that the reprojection errors of all cameras noticeably decline and the average reprojection error of all

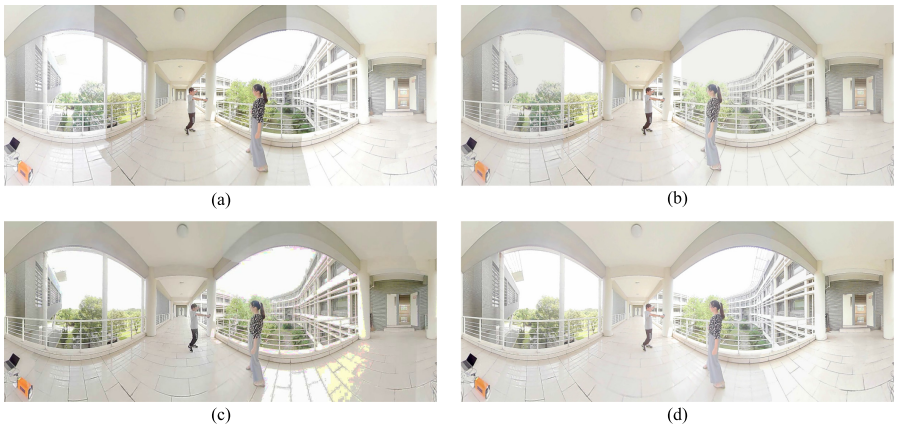

Fig. 10. Visual results of photometric alignment. (a) Result without using photometric alignment. (b) Result of Brown and Lowe [4]. (c) Result of Liu et al. [35]. (d) Our result.

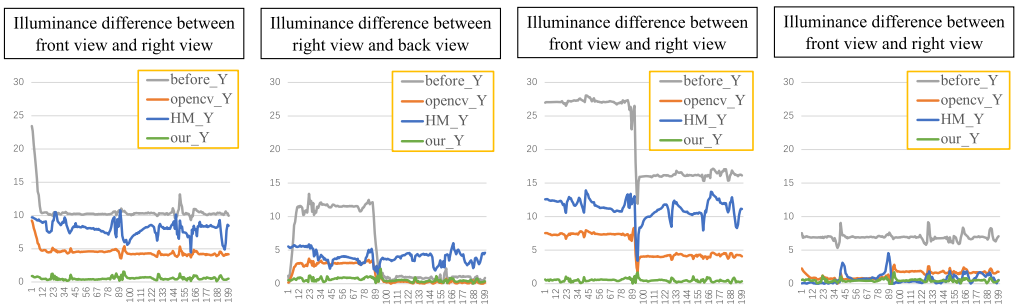

Fig. 11. Quantitative results of different photometric alignment schemes. The x-axis is the frame number in the test video. The y-axis is the intensity difference. In addition, "before_Y" is the result without using photometric alignment, "opencv_Y" is the result of Brown and Lowe [4], "HM_Y" is the result of Liu et al. [35], and "our_Y" is our result.

cameras decreases from 0.1881 to 0.1825, which demonstrates the effectiveness of the proposed joint optimization approach.

6.3 Photometric Alignment

In this experiment, we demonstrate the advantages of the proposed photometric alignment approach. To show the efficacy of our adjustment model, we compared our results with the results without using photometric alignment and the results of the baseline [4] qualitatively and quantitatively. In addition, we extended the histogram-matching-based photometric alignment method of two images in the work of Liu et al. [35] to four images and compared its results with our method.

We use an example in Figure 10 to illustrate the effectiveness of our adjustment model. Figure 10(a) is the result of direct stitching of geometrically aligned cylindrical projected images without performing photometric alignment. Figure 10(b) is the stitching result using the baseline photometric alignment approach [4], Figure 10(c) is the result of our implementation of Liu et al. [35], and Figure 10(d) is the result of Pedestrian360. In Figure 10(a), it can be noticed that the seam is very obvious, and the overall brightness transition is uneven, unlike images taken under

Fig. 12. Comparison of seam-cutting algorithms, including GraphCut [24], DP [8], Perception [29], and Iterative [30]. The stitching results of AutoStitch [4] without using seam-cutting are taken as the control group.

natural conditions. In Figure 10(b), although using the baseline [4] the unnaturalness is reduced, the overall brightness inconsistency on both sides of the seam can still be clearly perceived. In Figure 10(c), the histogram matching used in the work of Liu et al. [35] results in the distortion on the floor, and obvious light and dark transition can be observed on the column on the right part of the image. By contrast, the overall brightness transition in Figure 10(d) is relatively smooth and natural, and no obvious seam can be perceived. The apparent brightness differences in the sky in Figure 10(b) are not noticeable in Figure 10(d). The preceding results imply that our method has a stronger capability to make the images near the seam be blended more naturally.

In addition, the quantitative comparison on a test video in our dataset was conducted. Figure 11 presents the intensity differences between adjacent views of the three settings. It can be observed that when the original intensity difference (colored in gray) is large, our intensity difference (colored in green) is significantly lower than that of Brown and Lowe [4] (colored in red) and Liu et al. [35] (colored in blue). Furthermore, our method can maintain a relatively smooth result due to its multi-parameter adjustment model. From the preceding experimental results, the conclusion can be drawn that our approach can guarantee the robustness and stability of photometric alignment to a great extent.

6.4 Seam-Cutting

To verify the efficacy of our pedestrian-aware seam-cutting algorithm, we compared our scheme with four state-of-the-art seam-cutting algorithms, namely GraphCut [24], DP [8], Perception [29],

Table 1. Percentage of Broken Frames of the Competing Seam-Cutting Methods

Seam-Cutting	Indoor-1		Indoor-2		Outdoor-1		Outdoor-2		Outdoor-3		Average	
	Pedestrian	Object	Pedestrian	Object	Pedestrian	Object	Pedestrian	Object	Pedestrian	Object	Pedestrian	Object
DP [8]	55.5%	55.5%	6%	6%	21.5%	70%	16%	19.5%	7%	8%	21.2%	31.8%
GC [24]	26.5%	26.5%	1%	1	1.5%	**3.5%**	6.5%	6.5%	5.5%	5.5%	8.2%	8.6%
Perception [29]	46%	46.5%	0.5%	0.5%	7.5%	7.5%	13.5%	13.5%	3.5%	3.5%	14.2%	14.3%
Iterative [30]	22.5%	37.5%	3.5%	18%	23.5%	23.5%	4.5%	4.5%	8.5	9.5%	12.5%	18.6%
Pedestrian360	**4.5%**	**4.5%**	**0%**	**0%**	**0%**	5%	**0%**	**1.5%**	**1%**	**1.5%**	**1.1%**	**2.5%**

Table 2. Comparison of the Speeds of Different Seam-Cutting Algorithms on Different Scales

Scale	1.0×	0.5×	0.25×
DP [8]	140 ms	98 ms	67 ms
GC [24]	3,500 ms	730 ms	190 ms
Perception [29]	7,156 ms	2,040 ms	1,032 ms
Iterative [30]	22,180 ms	12,667 ms	5,368 ms
Ours (not update seams)	44 ms	40 ms	38 ms
Ours (update seams)	137 ms	96 ms	85 ms

and Iterative [30]. In addition, the stitching results of AutoStitch [4] without using seam-cutting were taken as the control group. We first adopted the same preprocessing pipeline of cylindrical projection and photometric alignment to ensure the fairness of the comparison. Then different seam-cutting algorithms were applied to the image blending stage.

Figure 12 presents the results of seam-cutting comparison on two groups of images. There exist ghost phenomenons in the results of AutoStitch [4] without using a seam-cutting algorithm. In the results of GraphCut [24] or Perception [29], the resultant pedestrian's body is partially missing or misaligned when there are pedestrians in the common FOV. The root cause lies in that a certain area passed by the seam belongs to the human body area in one image, whereas this area belongs to the background area in another image. Similarly, DP [8] failed in the group of the first row, and Iterative [30] failed in the group of the second row. Such incompleteness or misalignment of faces and bodies will cause great confusion to human perception. By contrast, our method ensures the integrity of the pedestrians, contributing to the visually appealing results.

The results of quantitative comparison on five videos in our dataset are reported in Table 1, which lists the percentage of frames with broken pedestrians and objects. In each test scenario, our method can achieve a very low ratio of broken frames, especially for pedestrians. However, the pixel-level scheme DP [8] results in a large number of broken frames without considering semantic information, especially in video Indoor-1 with many pedestrians. In terms of the average percentage of broken frames, ours is also the lowest one, which demonstrates the superiority of our pedestrian-aware seam-cutting algorithm.

We recorded the average stitching time per frame using different seam-cutting algorithms. To speed up the algorithm, the cylindrical projected images were downsampled, and seam-cutting was performed on a small scale. Then the seam masks were enlarged to the original resolution for thereafter image blending. The results are presented in Table 2, in which the full resolution 1.0× is 500×1200. The recorded time was consumed to find four seams using each algorithm. Experimental results show that Pedestrian360 can stitch videos at a speed of 12 to 26 fps by selectively updating seams and accelerating based on scaling. The stitching speed depends on the number of objects in the shooting scene and their frequencies of movements. When there are not too many dynamic objects in the scene passing through the seams, the seams rarely need to be updated, and the stitching speed of 26 fps can be achieved. Conversely, if the scene is crowded with pedestrians and objects pass through the seams in every frame, the stitching speed will be reduced to

Table 3. Qualitative Comparison of Different Panoramic Stitching Systems

Panorama System	Stitching Model Calculation	Photometric Alignment	Seam-Cutting	Stitching Time
Surveillance [13]	Feature extraction and warping	No	Graph-cut	12 fps
FPGA [49]	Camera pose calibration and cylindrical projection	No	Direct image blending without searching seams	30 fps
Rich360 [25]	Camera pose calibration and deformed spherical projection	The baseline in the work of Brown and Lowe [4]	Graph-cut	0.034 fps
VRWorks360 [27]	Feature extraction and warping	No	Direct image blending without searching seams	18 fps
MineSurveillance [2]	Hybrid image feature detection and registration with projection transformation	Only image preprocessing for defogging	Direct image blending without searching seams	21 fps
PanoramicVideo [35]	Pre-alignment by vertical translation	Histogram matching	Spatio-temporal seam optimization	0.417 fps
Pedestrian360	Camera pose calibration and cylindrical projection	Linear adjustment model with bias solved by two-step least-squares	DP considering semantic information	12–26 fps

12 fps. Our system's bottleneck is instance segmentation, which takes 43 ms per frame running on GPU, occupying about 50% of the stitching time for each frame. Mask-RCNN is an instance segmentation framework that supports multiple categories of instances. Replacing it with a faster network is a choice for acceleration. The time costs of the variant of the GC [29] and the iterative method [30] are extremely high due to a series of extra time-consuming steps like image saliency prediction and multiple rounds of iterative optimization. In terms of the speed alone, DP [8] can achieve performance comparable to ours, but from Table 1, its ratio of broken frames is highest among all evaluated methods, implying that its stitching performance is far from satisfactory. The preceding analysis implies that compared with its counterparts, our seam-cutting algorithm not only can achieve better stitching results but also is more suitable for time-critical applications such as surveillance or driver-assistance systems.

6.5 Comparison of Panorama Systems and Failure Cases Discussions

In this section, Pedestrian360 is compared with six existing representative panorama systems from four aspects—stitching model calculation, photometric alignment, seam-cutting, and stitching time—and Table 3 lists the qualitative comparison results. Note that most of algorithms that are close to real time are implemented by generating mapping tables in advance, including those in other works [13, 27, 49]. However, only performing feature points extraction and matching on a single group of images without accurate calibration may lead to errors. The application of this inaccurate mapping relationship to subsequent video frames results in adverse effects of misalignments [13, 27]. Our camera calibration pipeline with the novel joint optimization guarantees the accuracy of camera poses and contributes to geometrically consistent projected images. In the work of Bai et al. [2], image preprocessing for defogging helps unify different mine images' brightness, but it is hard to be extended to daily surveillance. The spatio-temporal seam optimization proposed in the work of Liu et al. [35] is extremely time consuming. By contrast, our pipeline considers more robust photometric alignment and seam-cutting while maintaining satisfactory time performance. In addition, our consideration of the semantic information is more suitable for surveillance applications than the competitors. In Pedestrian360, pedestrians will not be broken or misaligned, which will facilitate subsequent computer vision tasks such as pedestrian recognition.

Figure 13 presents four examples where Pedestrian360 fails to produce visually compelling results. In Figure 13(a) and Figure 13(b), because the ground and ceiling areas do not conform to the assumption of a cylindrical geometric model, there is an obvious misalignment at the seam. In the case of the rich texture of the ceiling and floor, the cylindrical model is not fully applicable,

Fig. 13. Failure cases. (a, b) Failure cases caused by cylindrical projection. (c) A failure case of seam-cutting caused by motion blur. (d) A failure case of photometric alignment.

whereas in an outdoor environment where the ground is flat, the stitching results are better. To increase the adaptability of the geometric model, Lee et al. [25] designed a deformable spherical projection surface to minimize parallax artifacts. However, it is time consuming to generate the optimized projection surface. Thus, one of our future works is to design deformable models that can adapt to various environments while maintaining acceptable time costs.

Figure 13(c) is a failure case of seam-cutting caused by motion blur. When a pedestrian quickly passes near the camera, motion blur will affect the accuracy of instance segmentation, causing the seam to fail to avoid the pedestrian. Fortunately, Pedestrian360 is a plug-in system in which the instance segmentation module can be replaced. Considering that instance segmentation is a rapidly developing research field, Mask-RCNN can someday be replaced with a more advanced module that can handle motion blur better to further improve Pedestrian360's performance.

Figure 13(d) is a failure case caused by photometric misalignment. There is a mismatch on the floor behind the lady. This is because the adjacent cameras are oriented differently, one toward the wall and the other toward the well-lit sky, resulting in different lighting conditions. When the lighting conditions of adjacent cameras are extremely different, photometric alignment performance will be affected.

7 CONCLUSION

In this article, we presented Pedestrian360, a panoramic video system based on a structured camera array. Pedestrian360 provides a 360° horizontal panorama with geometric and photometric consistency. This is achieved by our novel refinement approach for extrinsic calibration and the proposed robust photometric alignment scheme. Being aware of the sensitivity of human perception to high-level semantic objects, we are the first to introduce the instance segmentation into the seam-cutting pipeline and propose a pedestrian-aware seam-cutting algorithm, ensuring that pedestrians in the panorama will not be split, which is meaningful for human observation and computer vision tasks. Through extensive experiments, the effectiveness and efficiency of Pedestrian360 are verified and guaranteed. Compared with competing systems, its core advantages lie in the generation of highly consistent panorama in 12 to 26 fps and the consideration of semantic information. Our future work is to improve the cylindrical projection and design deformable models that can adapt to various environments while maintaining acceptable time costs.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2017. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 12 (2017), 2481–2495.

[2] Zongwen Bai, Ying Li, Xiaohuan Chen, Tingting Yi, Wei Wei, Marcin Wozniak, and Robertas Damasevicius. 2020. Real-time video stitching for mine surveillance using a hybrid image registration Method. *Electronics* 9, 9 (2020), 1336.

[3] Matthew Brown and David G. Lowe. 2003. Recognising panoramas. In *Proceedings of the IEEE International Conference on Computer Vision*. 1218–1225.

[4] Matthew Brown and David G. Lowe. 2007. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision* 74, 1 (2007), 59–73.

[5] Peter J. Burt and Edward H. Adelson. 1983. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics* 2, 4 (1983), 217–236.

[6] Yu-Sheng Chen and Yung-Yu Chuang. 2016. Natural image stitching with the global similarity prior. In *Proceedings of the European Conference on Computer Vision*. 186–201.

[7] Kyoungtaek Choi, Ho Gi Jung, and Jae Kyu Suhr. 2018. Automatic calibration of an around view monitor system exploiting lane markings. *Sensors* 18, 9 (2018), 2956.

[8] Alexei A. Efros and William T. Freeman. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'01)*. 341–346.

[9] Keren Fu, Qijun Zhao, and Irene Yu-Hua Gu. 2018. Refinet: A deep segmentation assisted refinement network for salient object detection. *IEEE Transactions on Multimedia* 21, 2 (2018), 457–469.

[10] Junhong Gao, Seon Joo Kim, and Michael S. Brown. 2011. Constructing image panoramas using dual-homography warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 49–56.

[11] Yi Gao, Chunyu Lin, Yao Zhao, Xin Wang, Shikui Wei, and Qi Huang. 2017. 3-D surround view for advanced driver assistance systems. *IEEE Transactions on Intelligent Transportation Systems* 19, 1 (2017), 320–328.

[12] Seung-Ryong Han, Jongsul Min, Taesung Park, and Yongje Kim. 2012. Photometric and geometric rectification for stereoscopic images. In *Three-Dimensional Image Processing and Applications II*, Vol. 8290. SPIE, 829007.

[13] Botao He and Shaohua Yu. 2016. Parallax-robust surveillance video stitching. *Sensors* 16, 1 (2016), 7.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*. 2961–2969.

[15] Adam Hedi and Sven Lončarić. 2012. A system for vehicle surround view. *IFAC Proceedings Volumes* 45, 22 (2012), 120–125.

[16] Lionel Heng, Mathias Burki, Gim Hee Lee, Paul Furgale, and Marc Pollefeys. 2014. Infrastructure-based calibration of a multi-camera rig. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 4912–4919.

[17] Lionel Heng, Bo Li, and Marc Pollefeys. 2013. Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. 1793–1800.

[18] Jie Hu, Dong-Qing Zhang, Heather Yu, and Chang Wen Chen. 2015. Discontinuous seam cutting for enhanced video stitching. In *Proceedings of the IEEE International Conference on Multimedia and Expo*. 1–6.

[19] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. 2013. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2083–2090.

[20] Peng Jiang, Haibin Ling, Jingyi Yu, and Jingliang Peng. 2013. Salient region detection by UFO: Uniqueness, focusness and objectness. In *Proceedings of the IEEE International Conference on Computer Vision*. 1976–1983.

[21] Jeonho Kang, Junsik Kim, Inhong Lee, and Kyuheon Kim. 2019. Minimum error seam-based efficient panorama video stitching method robust to parallax. *IEEE Access* 7 (2019), 167127–167140.

[22] Lai Kang, Yingmei Wei, Jie Jiang, and Yuxiang Xie. 2019. Robust cylindrical panorama stitching for low-texture scenes based on image alignment using deep learning and iterative optimization. *Sensors* 19, 23 (2019), 5310.

[23] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. 2011. G2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*. 3607–3613.

[24] Vivek Kwatra, Arno Schödl, Irfan Essa, Greg Turk, and Aaron Bobick. 2003. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics* 22, 3 (2003), 277–286.

[25] Jungjin Lee, Bumki Kim, Kyehyun Kim, Younghui Kim, and Junyong Noh. 2016. Rich360: Optimized spherical representation from structured panoramic camera arrays. *ACM Transactions on Graphics* 35, 4 (2016), 1–11.

[26] Hongdong Li and Richard Hartley. 2006. Five-point motion estimation made easy. In *Proceedings of the IEEE International Conference on Pattern Recognition*, Vol. 1. 630–633.

[27] Jiangeng Li, Minjie Fan, Guangsheng Wang, Xiaoli Li, and Rihui Sun. 2018. Panorama video stitching system based on VR Works 360 video. In *Proceedings of the IEEE Chinese Automation Congress*. 715–720.

[28] Jia Li, Yifan Zhao, Weihua Ye, Kaiwen Yu, and Shiming Ge. 2019. Attentive deep stitching and quality assessment for 360° omnidirectional images. *IEEE Journal of Selected Topics in Signal Processing* 14, 1 (2019), 209–221.

[29] Nan Li, Tianli Liao, and Chao Wang. 2018. Perception-based seam cutting for image stitching. *Signal, Image and Video Processing* 12, 5 (2018), 967–974.

[30] Tianli Liao, Jing Chen, and Yifang Xu. 2019. Quality evaluation-based iterative seam estimation for image stitching. *Signal, Image and Video Processing* 13, 6 (2019), 1199–1206.

[31] Tianli Liao and Nan Li. 2019. Single-perspective warps in natural image stitching. *IEEE Transactions on Image Processing* 29 (2019), 724–735.

[32] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*. 740–755.

[33] Wen-Yan Lin, Siying Liu, Yasuyuki Matsushita, Tian-Tsong Ng, and Loong-Fah Cheong. 2011. Smoothly varying affine stitching. In *Proceedings of the IEEE International Conference on Computer Vision*. 345–352.

[34] Hanyu Liu, Chong Tang, Shaoen Wu, and Honggang Wang. 2011. Real-time video surveillance for large scenes. In *Proceedings of the IEEE International Conference on Wireless Communications and Signal Processing*. 1–4.

[35] Qiongxin Liu, Xiangyang Su, Lei Zhang, and Hua Huang. 2020. Panoramic video stitching of dual cameras based on spatio-temporal seam optimization. *Multimedia Tools and Applications* 79, 5 (2020), 3107–3124.

[36] Si Liu, Zhen Wei, Yao Sun, Xinyu Ou, Junyu Lin, Bin Liu, and Ming-Hsuan Yang. 2018. Composing semantic collage for image retargeting. *IEEE Transactions on Image Processing* 27, 10 (2018), 5032–5043.

[37] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. 2010. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 2 (2010), 353–367.

[38] Koba Natroshvili and Kay-Ulrich Scholl. 2017. Automatic extrinsic calibration methods for surround view systems. In *Proceedings of the IEEE Intelligent Vehicles Symposium*. 82–88.

[39] Nils Plath, Marc Toussaint, and Shinichi Nakajima. 2009. Multi-class image segmentation using conditional random fields and global classification. In *Proceedings of the International Conference on Machine Learning*. 817–824.

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*. 91–99.

[41] Abhishek Sharma, Oncel Tuzel, and Ming-Yu Liu. 2014. Recursive context propagation network for semantic scene labeling. In *Advances in Neural Information Processing Systems*. 2447–2455.

[42] Nasim Souly, Concetto Spampinato, and Mubarak Shah. 2017. Semi supervised semantic segmentation using generative adversarial network. In *Proceedings of the IEEE International Conference on Computer Vision*. 5688–5696.

[43] Simon T. Y. Suen, Edmund Y. Lam, and Kenneth K. Y. Wong. 2006. Digital photograph stitching with optimized matching of gradient and curvature. In *Digital Photography II*, Vol. 6069. SPIE, 60690G.

[44] Marius Tennøe, Espen Helgedagsrud, Mikkel Næss, Henrik Kjus Alstad, Håkon Kvale Stensland, Vamsidhar Reddy Gaddam, Dag Johansen, Carsten Griwodz, and Pål Halvorsen. 2013. Efficient implementation and processing of a real-time panorama video pipeline. In *Proceedings of the IEEE International Symposium on Multimedia*. 76–83.

[45] Toshio Ueshiba and Fumiaki Tomita. 2002. Calibration of multi-camera systems using planar patterns. *Sensors* 8 (2002), 4.

[46] Matthew Uyttendaele, Ashley Eden, and Richard Skeliski. 2001. Eliminating ghosting and exposure artifacts in image mosaics. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2. 509–516.

[47] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. 2015. Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3183–3192.

[48] Yichen Wei, Fang Wen, Wangjiang Zhu, and Jian Sun. 2012. Geodesic saliency using background priors. In *Proceedings of the European Conference on Computer Vision*. 29–42.

[49] Yuan Xu, Qinghai Zhou, Liwei Gong, Mingcheng Zhu, Xiaohong Ding, and Robert K. F. Teng. 2013. High-speed simultaneous image distortion correction transformations for a multicamera cylindrical panorama real-time video system using FPGA. *IEEE Transactions on Circuits and Systems for Video Technology* 24, 6 (2013), 1061–1069.

[50] Julio Zaragoza, Tat-Jun Chin, Michael S. Brown, and David Suter. 2014. As-projective-as-possible image stitching with moving DLT. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 7 (2014), 1285–1298.

[51] Buyue Zhang, Vikram Appia, Ibrahim Pekkucuksen, Yucheng Liu, Aziz Umit Batur, Pavan Shastry, Stanley Liu, Shiju Sivasankaran, and Kedar Chitnis. 2014. A surround view camera solution for embedded systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 662–667.

[52] Guofeng Zhang, Yi He, Weifeng Chen, Jiaya Jia, and Hujun Bao. 2016. Multi-viewpoint panorama construction with wide-baseline images. *IEEE Transactions on Image Processing* 25, 7 (2016), 3099–3111.
[53] Lin Zhang, Juntao Chen, Dongyang Liu, Ying Shen, and Shengjie Zhao. 2019. Seamless 3D surround view with a novel burger model. In *Proceedings of the IEEE International Conference on Image Processing*. 4150–4154.
[54] Liuxin Zhang, Bin Li, and Yunde Jia. 2007. A practical calibration method for multiple cameras. In *Proceedings of the IEEE International Conference on Image and Graphics*. 45–50.
[55] Zhengyou Zhang. 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 11 (2000), 1330–1334.
[56] Wenbin Zou and Nikos Komodakis. 2015. HARF: Hierarchy-associated rich features for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. 406–414.

Received November 2020; revised March 2021; accepted April 2021