International Journal of Social Robotics (2022) 14:733–745
https://doi.org/10.1007/s12369-020-00744-8

Stitching Videos from a Fisheye Lens Camera and a Wide-Angle Lens Camera for Telepresence Robots

Yanmei Dong[1,2] · Mingtao Pei[2] · Lijia Zhang[2] · Bin Xu[2] · Yuwei Wu[2] · Yunde Jia[2]

Accepted: 31 December 2020 / Published online: 11 September 2021

Abstract

Many telepresence robots are equipped with a forward-facing camera for video communication and a downward-facing camera for navigation. In this paper, we propose to stitch videos from the FF-camera with a wide-angle lens and the DF-camera with a fisheye lens for telepresence robots. We aim at providing a compact and efficient user interface for telepresence robots with user-friendly interactive experiences. To this end, we present a multi-homography-based video stitching method which stitches videos from a wide-angle camera and a fisheye camera. The method consists of video image alignment, seam cutting, and image blending. We directly align the wide-angle video image and the fisheye video image based on the multi-homography alignment without calibration, distortion correction, and unwarping procedures. Thus, we can obtain a stitched video with shape preservation in the non-overlapping regions and alignment in the overlapping area for telepresence. To alleviate ghosting effects caused by moving objects and/or moving cameras during telepresence robot driving, an optimal seam is found for aligned video composition, and the optimal seam will be updated in subsequent frames, considering spatial and temporal coherence. The final stitched video is created by image blending based on the optimal seam. We conducted a user study to demonstrate the effectiveness of our method and the superiority of user interfaces with a stitched video.

Keywords Video stitching · Telepresence robot · User interface · Fisheye lens camera · Wide-angle lens camera · Image alignment

1 Introduction

A telepresence robot is a form of a video conferencing device mounted on a mobile robot, which allows a remote operator to teleoperate the robot as his/her embodiment to actively telecommunicate with local persons [21]. In recent years, telepresence robots are increasingly common in various everyday contexts, such as office environments [39], remote education [38], technical mediation [1], elderly people support [6], and residential care [42]. Many existing telepresence robots [11,27,50] are equipped with a forward-facing camera (FF-camera) for video communication and a downward-facing camera (DF-camera) for navigation. The two cameras provide two live videos displayed on two corresponding windows in the GUI for operators perceiving a local environment. However, in testing a user interface with these two video windows, we found that two video windows could introduce some confusion over a local environment [21]. For example, an operators often feels missing some views and context of the local environment, and distracts the attention due to frequently switching the two video windows. Fortu-

✉ Yuwei Wu
wuyuwei@bit.edu.cn

Yanmei Dong
dongyanmei@bit.edu.cn

Mingtao Pei
peimt@bit.edu.cn

Lijia Zhang
zhanglijia@bit.edu.cn

Bin Xu
xubinak47@bit.edu.cn

Yunde Jia
jiayunde@bit.edu.cn

[1] Tus College of Digit, Guangxi University of Science and Technology, Liuzhou 545006, People's Republic of China

[2] Beijing Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, People's Republic of China

nately, there is a great deal of overlap between two live videos owing to the wide-angle lens of cameras. Therefore, we propose to stitch the two live videos from the FF-camera and the DF-camera into a stitched video, and we aim at providing a more compact and efficient user interface for telepresence robots with user-friendly interactive experiences.

Video stitching has been widely used in video surveillance [37], virtual reality (VR) [30], and digital entertainment [46]. Existing methods mainly address two challenging issues: very high computational cost and visual artifacts (e.g., jitters, causing by the lack of a spatial and temporal coherence stitching model between successive frames). In our robotic telepresence application scenario, the FF-camera uses a wide-angle lens for video communication, and the DF-camera uses a fisheye lens to obtain a full view of the telepresence robot and its surroundings for navigation. So we have to face the strong distortion of fisheye videos and non-ideal inputs (e.g., the optical centers of the cameras are not exactly at the same location, the scene is non-planar, and/or dominant foreground objects move across cameras).

A straightforward scheme to handle these challenging issues is to perform image stitching on each pair of video images. There are some works on stitching wide-angle images [5,25] and fisheye images [20,52], which stitch the distorted images by correcting the distortion or unwarping the distorted images with the equirectangular projection. However, the distortion correction and the unwarping process may produce unnatural effects on regions near image edges. And if being designed to improve stitching quality, these methods often suffer from a high computational cost. Directly employing image stitching algorithms for video stitching also introduces noticeable visual artifacts (e.g., jitters).

In this paper, we develop and implement a multi-homography-based video stitching algorithm to create the stitched video from a wide-angle camera and a fisheye camera for telepresence robots. Our method consists of video image alignment, seam cutting, and video image blending. To provide a stitched video without shape distortion caused by stitching, we directly align the wide-angle video image and the fisheye video image based on the multi-homography alignment without distortion correction, unwarping with equirectangular projection, or other pre-processes. To alleviate ghosting effects caused by moving objects and/or moving cameras during telepresence robots driving, we use an enhanced dynamic programming algorithm to find an optimal seam for warped video image composition. The selected optimal seam will be updated in subsequent video images, considering spatial and temporal coherence. The final stitched video is created through image blending on the basis of the optimal seam. We conducted a user study on a telepresence robot equipped with a wide-angle lens camera and a fisheye lens camera to demonstrate the effectiveness of our method.

Our contributions are three-fold:

- We propose to stitch videos captured from a FF-camera and a DF-camera to provide a more compact and efficient user interface for telepresence robots, and provide remote operators with user-friendly interactive experiences.
- We present a multi-homography-based video stitching method to stitch videos from a wide-angle camera and a fisheye camera. Without calibration, distortion correction, and unwarping procedures, we can obtain a stitched video with shape preservation in the non-overlapping regions and alignment in the overlapping area.
- The user study results demonstrate the effectiveness of our method and the superiority of the user interfaces with a stitched video.

The remainder of this paper is organized as follows. Section 2 briefly reviews the related work. Section 3 describes the multi-homography-based video stitching algorithm for telepresence. The user study settings, procedures, results, and corresponding analyses are discussed in Sect. 4, and we conclude this work in Sect. 5.

2 Related Work

In this section, we make a review on image alignment and video stitching.

2.1 Image Alignment

Image alignment is essential for video stitching, and has attracted a lot of attentions in the past decades [48]. Conventional methods typically estimate a global transformation to bring an image pair into alignment [3,32,49], making an assumption that the scene is a roughly planar, or images are captured by purely rotating the camera about its optical center. Such imaging conditions are rarely met in practice, resulting in misalignments and ghosting effects in alignment results.

To address these problems, many efforts have been devoted to estimating multiple transformation. Gao et al. [15] estimated dual-homography for the image alignment when the scene can be divided into a distant plane and a ground plane. Lin et al. [34] proposed a smoothly varying affine transformation, according to the smoothly varying depth of the scene. Similarly, Zaragoza et al. [53] presented an as-projective-as-possible method (APAP) to estimate multiple homographies for better alignment. Lou and Gevers [35] described a piecewise planar region matching method to calculate multiple affine transformations, and they used multiple planes to approximate the image. These methods improve the alignment quality but heavily depend on keypoint detection and feature matching algorithms to offer sufficient and uniformly distributed keypoint correspondences. Additionally, keypoint detection, feature matching, and transformation estimation are time-consuming for real-time applications.

More recently, deep convolutional neural networks have been exploited to handle the problems of low efficiency and sparse keypoint correspondences for image alignment. DeTone et al. [10] designed a HomographyNet to directly estimate a homography between two images in an end-to-end manner. With the success of the HomographyNet, several deep learning-based homography estimation networks have been presented. Nowruzi et al. [13] proposed a hierarchy network using a twin convolutional network, while Chang et al. [8] presented a cascade Lucas-Kanade network by combining the Lucas-Kanade algorithm with the convolutional neural network. Apart from these supervised learning-based methods, an unsupervised homography estimation network was introduced by Nguyen et al. [40] for UAV image alignment.

The alignment of wide-angle images and/or fisheye images is more challenging, and suffers from heavy radial distortion [52]. In order for bringing wide-angle images into alignment, Jin [23] and Byröd et al. [5] estimated jointly the lens distortion and the alignment transformation, assuming that all images share the same distortion factors. For cameras in different radial distortions, Ju and Kang [24] estimated the lens distortion factor for each image, and then computed a homography for the alignment of synthetic images, whereas Kukelova et al. [28] estimated a homography and different distortion factors to bring images of real scene into alignment. In addition, Ho and Budagavi [20] proposed to align two images captured by a dual-fisheye lens camera. They unwarped the fisheye images into spherical 2-Dimensional space, and then employed a two-step alignment to register the unwarped images. Due to the unwarping process, the regions near edges of original images are stretched, leading to shape distortions in alignment results.

In our application scenario, we need to bring a wide-angle video image and a fisheye video image into alignment. To obtain an alignment with shape preservation in both the fisheye video image and the wide-angle video image for telepresence, we also need to bring the image pair into alignment without any distortion correction and unwarping processes.

2.2 Video Stitching

There are several commercial video stitching softwares, such as VideoStitch Studio[1] and AutoPano.[2] These softwares usually compute a 2D transformation relating two cameras, and then bring all pairs of video images into alignment for the post-production of stitched videos. To improve the quality of the stitched video, Li et al. [31] found double-seam to eliminate intensity misalignment, and similar work has been presented for designing a content-aware adaptive blending [26]. Some works were presented to obtain better alignment results. Lee and Sim [29] stitched videos by projecting the background plane and the foreground objects separately, while Jiang and Gu [22] stitched videos using spatial-temporal content-preserving warps. For videos captured by handheld cameras, Su et al. [41,47] and Guo et al. [17] combined the stitching and stabilization techniques together into a unified optimization framework for video stitching, whereas Lin et al. [33] stitched videos by reconstructing the 3D scene using the recovered 3D camera paths and the 3D scene points. These methods stitched videos in an iterative manner with low computational efficiency.

Besides, some work was designed for real-time processing or time-critical applications. For video surveillance applications, He and Yu [19] employed a background modeling algorithm and a change-detection-based optimal seam selection approach to stitch videos captured by fixed cameras. A Multi-UAV-based video surveillance system, SkyStitch [37], was designed and implemented for real-time aerial surveillance, employing flight information (e.g., the UAV attitude and GPS location) got from the flight controller as assistance. Okumura et al. [43] introduced a real-time video stitching method by implementing and improving a feature-based algorithm on a field-programable gate array (FPGA). Apart from hardware acceleration, EI-Saban et al. [12] developed a real-time method to stitch independent videos streamed by different mobile phones, while Silva et al. [46] stitched several live videos into a 360^o field of view and spread the stitched video based on GPU.

Since most existing approaches are designed for stitching videos from conventional cameras, they can not handle videos with heavy lens distortions captured by the wide-angle lens camera or fisheye lens camera. Considering the distortion, a simple method is to undistort the video images through a rectilinear projection, and then stitch the undistorted videos frame-by-frame [45]. Nevertheless, the undistortion may incur unnatural stretches on the regions near the borders of video images, particularly in video images captured by fisheye lens cameras.

Different from the existing methods, we stitch two live videos captured by a wide-angle lens camera and a fisheye lens camera mounted on a telepresence robot, and provide a more compact and efficient user interface for the operators to obtain friendly interactive experience. Without calibration, distortion correction, and unwarping procedures, we can obtain a stitched video with shape preservation in the non-overlapping regions and alignment in the overlapping area for telepresence.

[1] https://www.orah.co/software/videostitch-studio/.

[2] http://www.kolor.com/autopano/.

3 Video Stitching for Telepresence

In this section, we describe the algorithm that used to stitch the videos captured from a wide-angle camera and a fisheye camera mounted on a telepresence. We combine video image alignment, seam cutting and updating, image blending together to stitch the videos for telepresence, considering spatial and temporal coherence to alleviate the jitters. The pipeline of the video stitching algorithm is depicted in Fig. 1.

3.1 Video Image Alignment

Without calibration, distortion correction, and unwarping procedures, we align the wide-angle video image and the fisheye video image using the multi-homography alignment method proposed in our previous work [52]. A keypoint detector and descriptor (e.g., SUFT [2]) is used to obtain feature points from the two video images. After feature matching, inliers can be selected from the point correspondences by using a multi-homography inlier selection method. A global projective transformation and multiple local homographies are then estimated from the inliers. The final multi-homography warps are constructed by weighting between the global homoggraphy and local homography, in which local homographies are exploited for local region alignment. As a result, we can achieve a good alignment accuracy in the overlapping area and shape preservation in non-overlapping regions.

3.1.1 Multi-homography Inlier Selection

Feature matching can produce point correspondences from all the feature points, and there may include many mismatched points (i.e., outliers). To remove outliers from the point correspondence set, an inlier selection method can be employed. The Random Sample Consensus (RANSAC) [14] is popularly used to select inliers by generating multiple hypotheses for homography estimation.

In our work, the heavy lens distortion of the wide-angle video image I and the fisheye video image I' should be taken into consideration, since the homography is a plane transformation. We employ a multi-homography inlier selection method which can select more inliers for the alignment. A conditional sampling strategy [9] is used to generate multiple homography hypotheses. Given a set of point correspondences $F = \{(f_i, f_i')\}_{i=1}^{\hat{N}}$, we generate M homography hypotheses $\{h_1, \ldots, h_M\}$ by randomly sampling M minimal subsets of point correspondences from F, where (f_i, f_i') (denoted as F_i) is a point correspondence between the wide-angle video image and the fisheye video image. For each point correspondence F_i, its corresponding residuals to all homography hypotheses are calculated and ranked in a non-descending order. According to the residual order, a new list of homography hypotheses for F_i can be acquired by $h^i = \{h_1^i, \ldots, h_M^i\}$. F_i is more likely to be an inlier of the hypothesis with a lower residual.

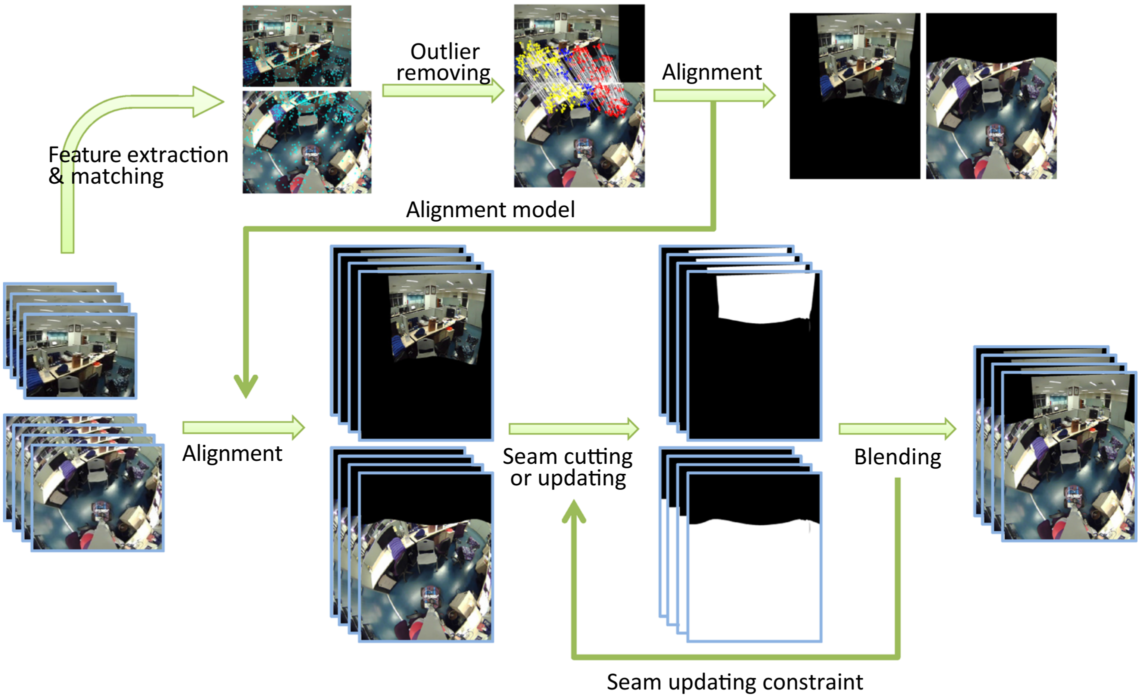


Fig. 1 Pipeline of the video stitching of a wide-angle video and a fisheye video

There may be many common hypotheses shared by h^i and h^j at the top place of the list, especially when F_i and F_j are in the same local area. A conditional inlier probability is computed to guide the inlier selection after selecting F_i as an inlier:

$$f(F_i, F_j) = \frac{1}{m}|h^i_{1:m} \cap h^j_{1:m}|, \tag{1}$$

where $h_{1:m}$ specifies the first-m hypotheses in the list, $|\cdot|$ is the counting operator, and $\cap$ is the intersection operator. Given the first selected inlier, the probability of that inlier with the rest point correspondences of F is used to select the second inlier.

In the experiment, the size of the minimal subset s, the outlier threshold ε_o and ε_r, the number of homography hypotheses M_0 and M are set to 4, 1, 0.01, 10, and 500, respectively.

3.1.2 Global Homography Estimation

Given an inlier set $P = \{p_i, p_i'\}_{i=1}^N$ between I and I', the global homography $H_g \in \mathbb{R}^{3\times 3}$ is defined by

$$p' \sim H_g p, \tag{2}$$

where $\sim$ denotes an equality up to a scale. p and p' are represented in 2D homogenous coordinates, and H_g is also in homogeneous. Omitting the scale term, Eq. (2) can be rewritten as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \tag{3}$$

Through a cross product on both side, Eq. (3) becomes $\mathbf{0}_{3\times 1} = p' \times H_g p$, and can be linearized as

$$\mathbf{0}_{3\times 1} = \begin{bmatrix} \mathbf{0}_{1\times 3} & -p^\top & y'p^\top \\ p^\top & \mathbf{0}_{1\times 3} & -x'p^\top \\ -y'p^\top & x'p\top & \mathbf{0}_{1\times 3} \end{bmatrix} [h_1, \dots, h_9]^\top = A\mathbf{h}, \tag{4}$$

and only two rows of A are linearly independent. Using the Direct Linear Transformation (DLT) [18], a global homography to fit all inliers can be solved by

$$\mathbf{h}^* = \arg\min_{\mathbf{h}} \sum_{i=1}^{N} \|\tilde{A}_i \mathbf{h}\|^2 = \arg\min_{\mathbf{h}} \|\tilde{A}\mathbf{h}\|^2, \tag{5}$$
$$s.t. \quad \|\mathbf{h}\|^2 = 1,$$

where $\tilde{A}_i$ is the first two rows of A for the i-th inlier, and $\tilde{A} \in \mathbb{R}^{2N\times 9}$ is the stack of all $\tilde{A}_i$.

Equation (5) can be solved through singular value decomposition (SVD), and global homography H_g is obtained by reshaping $\mathbf{h}^*$ into a 3×3 matrix.

3.1.3 Multi-homography Estimation

Due to strong distortions, wide-angle video image and fisheye video image alignment with a global homograhy will introduce large misalignment [52]. To increase the alignment quality, multiple local homographies are estimated by performing Moving DLT [53] on the inlier set P through

$$\mathbf{h}^* = \arg\min_{\mathbf{h}} \sum_{i=1}^{N} \|w_i \tilde{A}_i \mathbf{h}\|^2 = \arg\min_{\mathbf{h}} \|W\tilde{A}\mathbf{h}\|^2, \tag{6}$$
$$s.t. \quad \|\mathbf{h}\|^2 = 1,$$

for each position p^* in image I, and $W \in \mathbb{R}^{2N\times 2N}$ takes the form as $W = diag\,([w_1, w_1, \dots, w_N, w_N])$. The scalar weight w_i is defined as

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

where σ is a scale parameter, and $\gamma \in [0, 1]$ is used to avoid numerical issues. The inlier closer to p^* is given a higher weight, assuming that pixels in a local area share a homography.

Due to a lack of point correspondences, local homographies in non-overlapping regions are also calculated by inliers (i.e., some point correspondences in the overlapping region). To alleviate the artifact, we integrate the local homography $\mathbf{h}^*$ and the global homography H_g into a new homography, taking advantage of both homographies for local alignment and shape preservation. The integration formulation is given by

$$H = wH_l + (1 - w)H_g, \tag{8}$$

where w is the integration parameter, and H_l is the 3×3 matrix form of the local homography $\mathbf{h}^*$.

Since the deformation caused by the global homography increases along the positive u-axis from the overlapping region to the non-overlapping regions [7], H is smoothed from local homography to global homography along u-axis using w,

$$w = (u - u_m)/(u_M - u_m), \tag{9}$$

where (u, v) is a new coordinate obtained by rotating the original coordinate (x, y) of the warped image of I. u_m and u_M are the minimum and the maximum u coordinate of all pixels, respectively. The rotation angel is $\theta = \arctan(h_8/h_7)$.

Note that image I' is warped by using $R = H(H_l)^{-1}$ to compensate the local homography warping effects on the overlapping region.

3.2 Optimal Seam Cutting and Updating

To alleviate ghosting effects caused by moving objects and/or moving cameras, an optimal seam is found for aligned video composition, and the optimal seam will be updated in subsequent frames, considering spatial and temporal coherence. The final stitched video image, called forward-downward-facing video image (FDF-video image), is created by using the multi-band blending to provide a smooth transition of lighting from the downward-facing video image (DF-video image) to the forward-facing video image (FF-video image).

3.2.1 Seam Cutting

Seam cutting is utilized to select an optimal pixel-based continuous curve (seam) for image blending, which can alleviate ghosting effects caused by moving objects and/or moving cameras. As demonstrated in Fig. 2, the seam cutting task is to find an optimal seam, e.g., the red line from point A to point B over the overlapping area. We use an enhanced dynamic programming approach [16] that holds search directions as shown in Fig. 3 to find the optimal seam.

The enhanced seam is defined as

$$C_{i,j} = e_{i,j} + \min\left(C_{i-1,j-1}, C_{i-1,j}, C_{i-1,j+1}, C_{i,j-1}, C_{i,j+1}\right), \quad (10)$$

where (i, j) is a pixel coordinate, C and e indicate the cumulative cost and the gradient cost, respectively. To find a seam without gradient difference and visible artifacts, we define the gradient cost by gradient smoothness S_m and gradient similarity S_d:

$$e = S_m + S_d. \quad (11)$$

Given the overlapping areas I_s and I_t of the two warped images, S_m and S_d are defined as

$$S_m = \|\nabla(I_s + I_t)\|/mean(\|\nabla(I_s + I_t)\|), \quad (12)$$

$$S_d = \|\nabla(I_s - I_t)\|/mean(\|\nabla(I_s - I_t)\|), \quad (13)$$

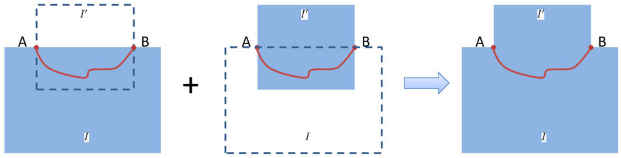


Fig. 2 Seam cutting. The task is to find an optimal seam from point A to point B, such as the red line

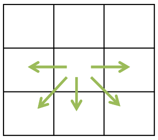


Fig. 3 Search directions of the enhanced dynamic programming algorithm

where $\|\cdot\|$ and ∇ are the L2-norm and the gradient operator, respectively.

3.2.2 Selected Seam Updating

The selected seam will be dynamically updated in successive video images. Avoiding to introduce noticeable artifacts caused by large drift of the optimal seams between successive video images, we employ a seam updating method based on temporal propagation constraint [44] to gain stable seams.

The temporal propagation constraint is constructed based on location information of the optimal seam in the previous video image. It is represented by a matrix $\mathbf{D}^{t-1}_{w\times h}$, where t is the index of the video image. Each element of $\mathbf{D}^{t-1}_{w\times h}$ is a penalty for each point in the overlapping area, which equals to horizontal distance between the corresponding point and the optimal seam of the previous video image. In other words, the penalty increases with the distance.

For the current video image, we can get a cost matrix $\mathbf{C}^{t}_{w\times h}$ using Eq. (10). Combining with the temporal propagation constraint, the final cost matrix $\tilde{\mathbf{C}}^{t}_{w\times h}$ is calculated using

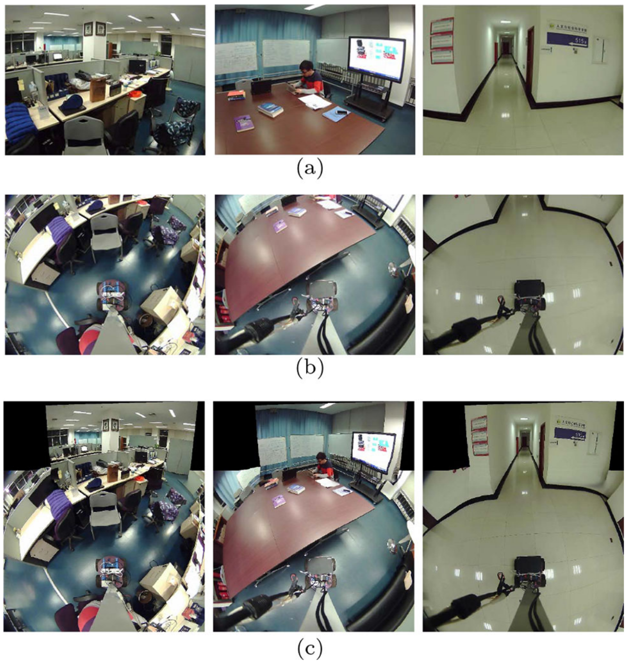
$$\tilde{\mathbf{C}}^{t}_{w\times h} = \mathbf{C}^{t}_{w\times h} + \mathbf{D}^{t-1}_{w\times h}. \quad (14)$$

We can update the selected seam by performing the enhanced dynamic programming algorithm again.

3.3 Video Image Blending

Due to the distinct orientations of the DF-camera and the FF-camera, there exist lighting inconsistencies between the same scenes of the DF-video image and the FF-video image. Video image blending can be used to achieve a smooth transition of lighting from one image to the other. We utilize the multi-band blending [4] which is widely used and relatively insensitive to misalignment [54], for warped video image composition.

We build a Laplacian pyramid on each warped video image, and the blending becomes a solution of feather blending on each pyramid level. To obtain the weights used to perform feather blending, each weight image built on the optimal seam is converted into a Gaussian pyramid, and each Gaussian pyramid level is a weight map for corresponding level of the Laplacian pyramid. The FDF-video image is reconstructed by interpolating and merging all the blended pyramid levels.


(a)

(b)

(c)

Fig. 4 Stitching results on wide-angle images and fisheye images of different scenes. **a** Original wide-angle video images captured by the FF-camera. **b** Original fisheye video images captured by the DF-camera, corresponding to the original wide-angle video images in **a**. **c** Stitching results (the FDF-video images)

3.4 Stitching Results

In general, each image pair should be aligned using the alignment algorithm. However, alignment on every newcome image pair will make the stitching slow due to the low computational efficiency of keypoint detection, feature matching, and inlier selection. Fortunately, the DF-camera and the FF-camera are fixed on the vertical post of the telepresence robot. We can assume that the configuration of the cameras are not changed in a matter of seconds, so we do not need to estimate the alignment model on every image pair.

Some stitched video images and the corresponding original wide-angle video images captured by the FF-camera and fisheye video images captured by the DF-camera are shown in Fig. 4. These images consists of complex scenes and simple scenes, and are captured by a telepresence robot with a wide-angle lens camera and a fisheye lens camera.

4 User Study

We conducted a user study to compare user interfaces with one video (the FDF-video) and two videos (the DF-video and the FF-video) for telepresence robots. For simplicity, we denote the user interface with the FDF-video as 'FDF', while the user interface with the FF-video and the DF-video as 'FF+DF'.

4.1 Experimental Platform

The experimental platform consists of a telepresence robot in a local environment, user interfaces used by an operator in a remote environment, and wireless communication networks for connecting the two environments. An operator can use a tablet in a remote environment to drive a telepresence robot to perceive a local environment.

The telepresence robot used in our user study was developed in our lab, called Mcisbot [21], as depicted in Fig. 5. It uses the Pioneer 3-AT as a mobile robot base equipped with a special designed robot head for telepresence. The robot head contains a light LCD screen, a forward-facing camera (FF-camera), a downward-facing camera (DF-camera), and a speaker & microphone, and all together are mounted on a pan-tilt platform hold up by a vertical post. The FF-camera with a wide-angle lens can provide a live video for clear watching of targets or persons in front. The DF-camera with a fisheye lens provides a complete watching of the ground around the robot for navigation.

The Mcisbot was specifically designed to evaluate the usability of the Touchable live video Image based User Interface (TIUI). The most notable feature of the TIUI is that there are no explicit graphical buttons, arrow keys, and/or menus, compared to traditional touchscreen GUIs. The TIUI allows operators to drive the robot by directly touching the live video images with finger touch gestures. Naturally, we use the TIUI to test the effectiveness of the stitched video for convenience. Moreover, since traditional touchscreen GUIs are widely used to support pilot operators to teleoperate telepresence robots [11,36,50,51], we also conducted the user study on the GUIs to evaluate the effectiveness of the proposed method. Figure 6a shows the TIUI containing two separate videos (the FF-video and the DF-video) and the one with a

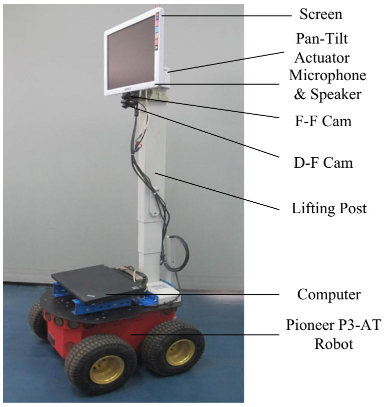
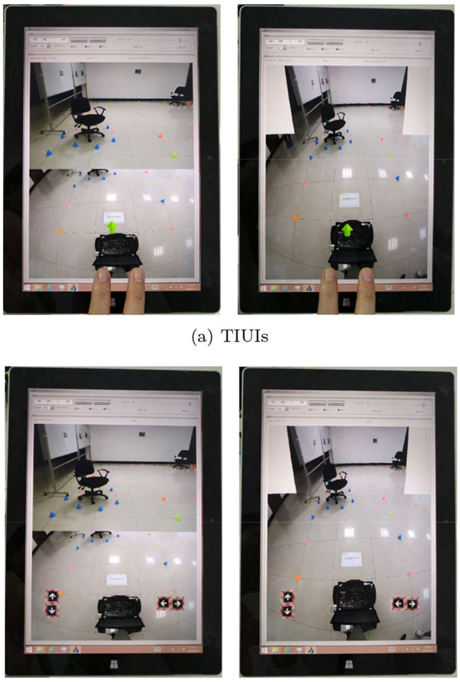


Fig. 5 The Mcisbot robot



(a) TIUIs

(b) GUIs

Fig. 6 Two kinds of user interfaces used in our user study. **a** The GUI with two videos (left) and the GUI with a FDF-video (right). **b** The TIUI with two videos (left) and the TIUI with a FDF-video (right)

stitched video (the FDF-video). The GUIs used in our experiments are shown in Fig. 6b.

During the experiment, we resized both the wide-angle video image and the fisheye video image to 640×480 for stitching. Using a laptop computer with an I5-2410M Intel 2.30 GHz CPU and 4 GB RAM, we achieved a rate of video stitching up to 15 fps (frames per second). Considering system delay and video stitching efficiency, we directly show the stitching result in the user interfaces, without any post-processing.

4.2 Participants

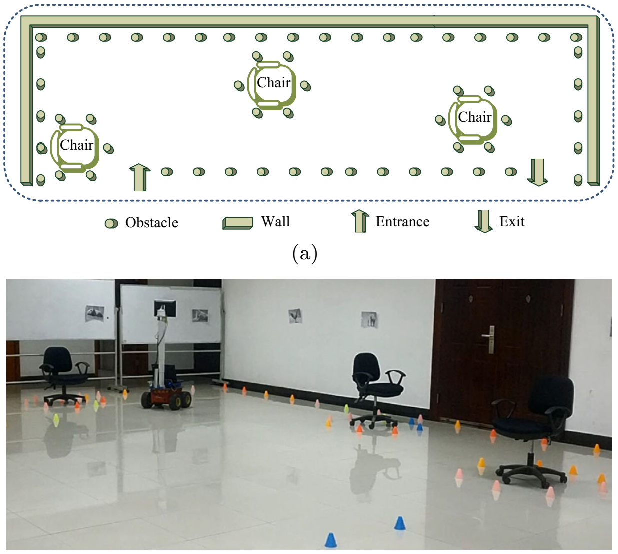
We recruited 18 participants from the local university for the user study, whose ages vary from 17 to 28 years (M = 21.28, SD = 2.803), where M and SD indicate the mean value and the standard deviation, respectively. All participants use computers in their daily life. With a five-point scale for familiarity, ranging from “1 = not at all familiar” to “5 = very familiar”, all participants reported their familiarity with telepresence robots (M = 1.72, SD = 0.752). A few of them heard of telepresence robots, but had no experience in telepresence robot operation. The others expressed that they had no idea about telepresence robots. Similarly, all the participants reported how familiar they were with video chat on the same five-point scale, and most of them had video chat experiences (M = 3.83, SD = 1.200).

4.3 Environment Setup

We constructed an experimental room in our lab as a local environment to simulate a complex environment, such as a museum or a meeting room that contains some obstacles, pictures, and chairs. The physical arrangement of the local environment is shown in Fig. 7a. The obstacles and chairs offered participants the direction to drive the robot to walk around the local environment, and also played roles as anti-collision objects for safely driving. Operators were located in another room, specified as a remote environment, to remotely drive the robot using the user interface, and a picture of an operator driving the robot to walk around the experimental room is shown in Fig. 7b.

4.4 Tasks

For comparing the FDF and the FF+DF, we assigned each participant with tasks of animal picture recognition and robot driving. When performing these tasks by using the FF+DF, a remote operator has to perceive the local environment from two videos, such that, he/she needs to frequently switch his/her attention between the FF-video and the DF-video to watch different parts of the local environment, i.e., look at the animal pictures on the wall through the FF-video and look at the floor through the DF-video for robot navigation. Besides, a remote operator has to spend more time to become familiar with the FF-video and the DF-video, and find the scene relationship between the FF-video and the DF-video. Instead, by



(a)

(b)

Fig. 7 The local environment. **a** The physical arrangement of the local environment. **b** The robot was driven to walk around the local environment

using the FDF, a remote operator can directly perceive the local environment from a single video.

Using each user interface, a participant was required to drive the Mcisbot robot to walk through the local environment in safe. Meanwhile, he/she should recognize all the animal pictures on the wall and tell the experimenter what animal he/she has saw. There were 8 animal pictures on the wall and with different recognition difficulty, e.g., different contrast between the foreground and the background.

The two cameras are designed for different purposes, the DF-camera for navigation and the FF-camera for communication. To keep the robot safe from collision with other objects, operators needed to focus on the DF-video (or the lower part of the FDF-video) all the time when they were driving the robot. However, to recognize pictures on the wall, participants needed to watch the FF-video (or the upper part of the FDF-video). For safely driving, participants may stop driving when they could not watch the ground around the robot.

4.5 Measurements and Analyses

We investigated objective and subjective measurements for evaluating the FDF and the FF+DF.

The objective measurements measure participants' task performance, including task completion time, total number, correct number. The task completion time is timing from the robot entering the entrance to it coming out from the exit. The total number indicates the number of pictures seen by a participant, and the correct number is the number of correctly recognized pictures.

The subjective measurements were gained through three questionnaires and consist of the perceived task success, the participants' preference between the FDF and the FF+DF. For the participants' preference, we compared the number of persons for each option. There was a five-point scale for other questions, and the larger was the better.

A one-way fixed-effects analysis of variance (ANOVA) was conducted to test the effects of the two cases upon measurements of task completion time and perceived task success. For tests of statistical significance, we used a cut-off value of $p < 0.05$.

4.6 Procedure

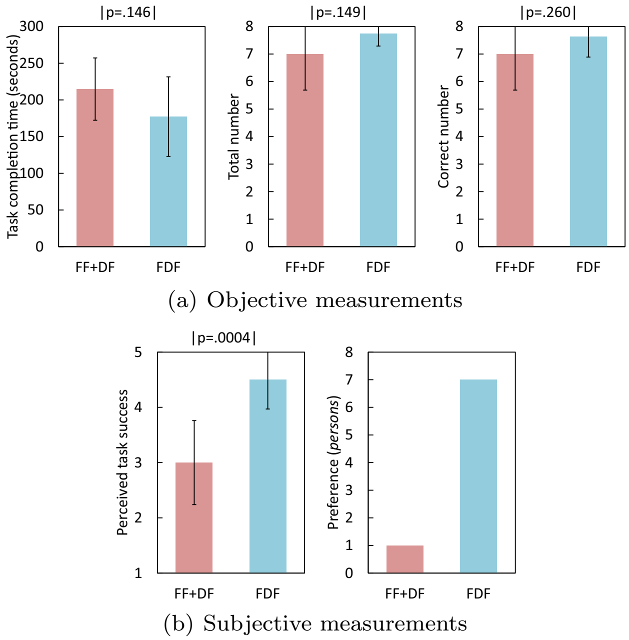
A mixed between- and within-subject user study was conducted. The user interface type (the TIUI and the GUI) was the between-subject variable, and all participants were divided into two groups, one group using the TIUI and the other group using the GUI. The number of videos, i.e., one video (the FDF-video) and two videos (the FF-video and the DF-video) was the within-subject variable such that a better comparison between the two manners can be obtained. To counterbalance the possible order effect, we permuted the order of using the FDF and the FF+DF, i.e., the odd-numbered participants first used the FDF, and the even-numbered participants first used the FF+DF. We changed the locations of animal pictures and furniture after the first trial and told the participants that the physical layout of the environment may have been changed. Besides, we also told participants that the total number of animal pictures may have been changed, and animal pictures may be different from those used in the first trial. As a result, participants need to re-explore the environment to complete the task in the second trial.

In preparation, a participant was given an overview of the experimental task, and then an experimenter provided the instructions on how to use the TIUI or the GUI to drive the Mcisbot robot. All participants were given 10 minutes to practice remotely driving the robot with the help of an experimenter in the training room. This training environment contained several different animal pictures for participants to practice the recognition task. Most of participants completed the practices in less than 10 minutes. And most of participants spent practice time with the TIUI exceeding the GUI because the GUI is simple and intuitive while skilled operation with touchscreen gestures on the TIUI need to be obtained. But using the TIUI to remotely drive the robot is more flexible than the GUI, particularly changing the direction and speed of the robot.

After the experimenter drove the robot to entrance of the local environment, a participant started to operate through the FDF (or the FF+DF), and the timer was started until the robot came out from the exit. The number of the correctly recognized pictures and the total number of the saw pictures were accounted by the experimenter. The participant was asked to fill in the first questionnaire and prepared to using the FF+DF (or the FDF) to start the task again. Before the task, the experimenter changed the locations of the pictures and furniture, and told the participant that the layout of the environment may have been changed. Then, the participant started to re-explore the local environment again to complete the task. After the task, the participant was required to fill in the second questionnaire, and the same objective measurements were recorded. Finally, the third questionnaire designed for capturing the participant's preference on the FDF and the FF+DF was also required to be filled in.

4.7 Results

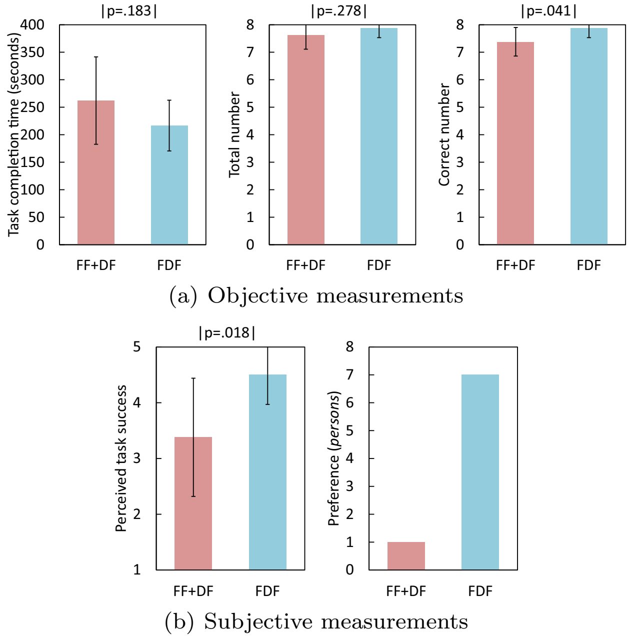
We obtained 16 sets of effective experiment data, 8 sets for each type of user interfaces (the TIUI and the GUI). Figures 8 and 9 show the results of the user study using the TIUI and the GUI, respectively. Herein, the experiment focused on comparing the FDF and the FF+DF, so we run data analysis on the TIUI and the GUI, respectively, and the comparison of the TIUI and the GUI can be found in [21].



(a) Objective measurements

(b) Subjective measurements

Fig. 8 Evaluations on the TIUI. Note that the error bars show the standard deviations of the data



(a) Objective measurements

(b) Subjective measurements

Fig. 9 Evaluations on the GUI. Note that the error bars show the standard deviations of the data

The objective measurements include mean task completion time (Task-completion time), the total number of pictures seen by a participant (Total number), and the number of correctly recognized pictures (Correct number). The subjective measurements are composed of perceived task success and the participants' preference between the FDF and the FF+DF.

The average task completion time using the TIUI with the FDF was 177.25 s (SD = 54.22), and 214.75 s (SD = 42.48) for the FF+DF, as shown in Fig. 8. The effect on the task completion time was not significant, F(1,14) = 2.37, $p = 0.146$, $\eta^2 = 0.145$, possibly because participants were not skilled in changing the moving speed of the robot. For the pictures that had been seen, the average number was 7.75 (SD = 0.46) and 7 (SD = 1.31) for the FDF and the FF+DF, respectively. We did not found a significant effect of video stitching on this measurement, F(1,14) = 2.33, $p = 0.149$, $\eta^2 = 0.143$. Besides, the average number of correctly recognized pictures using the FDF (M = 7.63, SD = 0.74) was larger than using the FF+DF (M = 7, SD = 1.31), and F(1,14) = 1.38, $p = 0.260$, $\eta^2 = 0.090$. There was a significant effect on perceived task success with F(1,14) = 21, $p = 0.0004$, $\eta^2 = 0.6$, (M = 4.5, SD = 0.53) for the FDF and (M = 3, SD = 0.76) for the FF+DF. Additionally, almost all participants preferred using the FDF than the FF+DF, as shown in Fig. 8b.

For the GUI, the average task completion time of using the FDF (M = 216.63, SD = 46.11) was similar with the FF+DF (M = 262.13, SD = 79.47), and F(1,14) = 1.96, $p = 0.183$, η^2=.123. The average number of pictures that had been seen using the FDF (M = 7.88, SD = 0.35) and the FF+DF (M = 7.63, SD = 0.52) were almost the same. The average number of correctly recognized pictures for FDF and FF+DF are 7.88 (SD = 0.35) and 7.38 (SD = 0.52), respectively. We found a significant effect on this measurement, F(1,14) = 5.09, $p = 0.041$, $\eta^2 = 0.267$. We found a significant effect on perceived task success, F(1,14) = 7.18, $p = 0.018$, $\eta^2 = 0.339$, and the FDF had a mean value of 4.5 (SD = 0.53) while the FF+DF was 3.38 (SD = 1.06). Similar to the user study using the TIUI, almost all participants preferred to using the GUI with the FDF rather than the FF+DF, as shown in Fig. 9b. For a clear comparison, we list the results in Table 1.

Several participants reported that the FDF is a more compact and high efficient user interface than the FF+DF. Using the GUI, six participants said that the FDF was more convenient in perceving the local environment, and they felt easier in distance perception because the scene is continuous. Five participants using the TIUI also gave us the same opinion. Additionally, five participants using the GUI with two videos described that they had to frequently switch between the FF-video and the DF-video, and three operators using the TIUI with two videos noticed the same problem. This problem was not existing on the FDF, since the scene in the FDF was continuous.

In summary, both the user study results on the TIUI and the GUI have demonstrated the effectiveness of our method. The telepresence robot incorporating with our video stitching algorithm can provide more friendly interactive experiences.

Table 1 Experimental results of the user study using the TIUI and the GUI

User interface	Measurement	Video	Mean	Standard deviation	F-value	P-value	Effect size
TIUI	Task-completion time	FF+DF	214.75	42.48	2.37	0.146	0.145
		FDF	177.25	54.22			
	Total number	FF+DF	7	1.31	2.33	0.149	0.143
		FDF	7.75	0.46			
	Correct number	FF+DF	7	1.31	1.38	0.260	0.090
		FDF	7.63	0.74			
	Perceived task success	FF+DF	3	0.76	21	0.0004	0.6
		FDF	4.5	0.53			
	Preference	FF+DF	1	–	–	–	–
		FDF	7	–			
GUI	Task-completion time	FF+DF	262.13	79.47	1.96	0.183	0.123
		FDF	216.63	46.11			
	Total number	FF+DF	7.63	0.52	1.27	0.278	0.083
		FDF	7.88	0.35			
	Correct number	FF+DF	7.38	0.52	5.09	0.041	0.267
		FDF	7.88	0.35			
	Perceived task success	FF+DF	3.38	1.06	7.18	0.018	0.339
		FDF	4.5	0.53			
	Preference	FF+DF	1	–	–	–	–
		FDF	7	–			

5 Conclusions

This paper has proposed to stitch two live videos to provide a more compact and high efficient user interface for remote operators of telepresence robots with friendly interactive experiences. The two live videos can be captured by a forward-facing camera with wide-angle lens and a downward-facing camera with fisheye lens for video communication and navigation in robotic telepresence systems, respectively. A multi-homography-based video stitching algorithm, consisting of video image alignment, seam cutting, and image blending, can stitch these videos without calibration, distortion correction, and unwarping procedures. The user study on a telepresence robot was conducted and results demonstrated the effectiveness of our method and the superiority of the user interface with a stitched video.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

Funding This work was supported by the Natural Science Foundation of China (NSFC) under Grants No. 61773062 and No. 61702037.

References

1. Bae I (2018) Public acceptance of fundamental rights via a telepresence robot and a video call stand in south korea. Int J Social Robot 10(4):503–517
2. Bay H, Tuytelaars T, Gool LV (2006) Surf: speeded up robust features. In: Proceedings of the European conference on computer vision, pp 404–417
3. Brown M, Lowe DG (2007) Automatic panoramic image stitching using invariant features. Int J Comput Vis 74(1):59–73
4. Burt PJ, Adelson EH (1983) A multiresolution spline with application to image mosaics. ACM Trans Graph 2(4):217–236
5. Byröd M, Brown MA, Åström K (2009) Minimal solutions for panoramic stitching with radial distortion. In: Proceedings of British machine vision conference, pp 1–11
6. Cesta A, Cortellessa G, Orlandini A, Tiberio L (2016) Long-term evaluation of a telepresence robot for the elderly: methodology and ecological case study. Int J Soc Robot 8(3):421–441
7. Chang CH, Sato Y, Chuang YY (2014) Shape-preserving half-projective warps for image stitching. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3254–3261
8. Chang CH, Chou CN, Chang EY (2017) Clkn: cascaded Lucas–Kanade networks for image alignment. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2213–2221
9. Chin T, Yu J, Suter D (2012) Accelerated hypothesis generation for multistructure data via preference analysis. IEEE Trans Pattern Anal Mach Intell 34(4):625–638
10. DeTone D, Malisiewicz T, Rabinovich A (2016) Deep image homography estimation. arXiv preprint arXiv:1606.03798
11. Double (2017) Double robotics. http://www.doublerobotics.com

12. El-Saban MA, Refaat M, Kaheel A, Abdul-Hamid A (2009) Stitching videos streamed by mobile phones in real-time. In: Proceedings of the 17th ACM international conference on multimedia, pp 1009–1010
13. Erlik Nowruzi F, Laganiere R, Japkowicz N (2017) Homography estimation from image pairs with hierarchical convolutional networks. In: Proceedings of the IEEE international conference on computer vision workshops, pp 913–920
14. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
15. Gao J, Kim SJ, Brown MS (2011) Constructing image panoramas using dual-homography warping. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 49–56
16. Gu H, Yu Y, Sun W (2009) A new optimal seam selection method for airborne image stitching. In: Proceedings of the IEEE international workshop on imaging systems and techniques, pp 159–163
17. Guo H, Liu S, He T, Zhu S, Zeng B, Gabbouj M (2016) Joint video stitching and stabilization from moving cameras. IEEE Trans Image Process 25(11):5491–5503
18. Hartley R, Zisserman A (2003) Multiple view geometry in computer vision. Cambridge University Press, Cambridge
19. He B, Yu S (2015) Parallax-robust surveillance video stitching. Sensors 16(1):7
20. Ho T, Budagavi M (2017) Dual-fisheye lens stitching for 360-degree imaging. In: Proceedings of the IEEE international conference on acoustics, speech and signal processing, pp 2172–2176
21. Jia Y, Xu B, Shen J, Pei M, Dong Z, Hou J, Yang M (2015) Telepresence interaction by touching live video images. arXiv preprint arXiv:1512.04334
22. Jiang W, Gu J (2015) Video stitching with spatial-temporal content-preserving warping. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 42–48
23. Jin H (2008) A three-point minimal solution for panoramic stitching with lens distortion. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–8
24. Ju MH, Kang HB (2013) Panoramic image generation with lens distortions. In: Proceedings of the 20th IEEE international conference on image processing, pp 1296–1300
25. Ju MH, Kang HB (2014) Stitching images with arbitrary lens distortions. Int J Adv Rob Syst 11(1):2
26. Kim BS, Choi KA, Park WJ, Kim SW, Ko SJ (2017) Content-preserving video stitching method for multi-camera systems. IEEE Trans Consum Electron 63(2):109–116
27. Kristoffersson A, Coradeschi S, Loutfi A (2013) A review of mobile robotic telepresence. Adv Hum Comput Interact 2013:1–17
28. Kukelova Z, Heller J, Bujnak M, Pajdla T (2015) Radial distortion homography. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 639–647
29. Lee KY, Sim JY (2015) Robust video stitching using adaptive pixel transfer. In: Proceedings of the IEEE international conference on image processing, pp 813–817
30. Lee WT, Chen HI, Chen MS, Shen I, Chen BY et al (2017) High-resolution 360 video foveated stitching for real-time vr. Comput Graph Forum 36(7):115–123
31. Li J, Xu W, Zhang J, Zhang M, Wang Z, Li X (2015) Efficient video stitching based on fast structure deformation. IEEE Trans Cybern 45(12):2707–2719
32. Li Y, Monga V (2016) Siasm: sparsity-based image alignment and stitching method for robust image mosaicking. In: Proceedings of the IEEE conference on image proceeing, pp 1828–1832
33. Lin K, Liu S, Cheong LF, Zeng B (2016) Seamless video stitching from hand-held camera inputs. Comput Graph Forum 35(2):479–487
34. Lin WY, Liu S, Matsushita Y, Ng TT, Cheong LF (2011) Smoothly varying affine stitching. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 345–352
35. Lou Z, Gevers T (2014) Image alignment by piecewise planar region matching. IEEE Trans Multimed 16(7):2052–2061
36. Mayer S, Tschofen A, Dey AK, Mattern F (2014) User interfaces for smart things-a generative approach with semantic interaction descriptions. ACM Trans Comput Hum Interact 21(2):12
37. Meng X, Wang W, Leong B (2015) Skystitch: a cooperative multi-uav-based real-time video surveillance system with stitching. In: Proceedings of the 23rd ACM international conference on multimedia, pp 261–270
38. Meyer B (2016) Mobile devices and spatial enactments of learning: ipads in lower secondary schools. In: Proceedings of the 12th international conference on mobile learning, pp 3–10
39. Neustaedter C, Venolia G, Procyk J, Hawkins D (2016) To beam or not to beam: a study of remote telepresence attendance at an academic conference. In: Proceedings of the 19th ACM conference on computer-supported cooperative work and social computing, pp 418–431
40. Nguyen T, Chen SW, Skandan S, Taylor CJ, Kumar V (2018) Unsupervised deep homography: a fast and robust homography estimation model. IEEE Robot Autom Lett 3(3):2346–2353
41. Nie Y, Su T, Zhang Z, Sun H, Li G (2018) Dynamic video stitching via shakiness removing. IEEE Trans Image Process 27(1):164–178
42. Niemelä M, van Aerschot L, Tammela A, Aaltonen I, Lammi H (2019) Towards ethical guidelines of using telepresence robots in residential care. Int J Soc Robot 1–9
43. Okumura Ki, Raut S, Gu Q, Aoyama T, Takaki T, Ishii I (2013) Real-time feature-based video mosaicing at 500 fps. In: Proceedings of the IEEE/RSJ international conference on intelligent robots and systems, pp 2665–2670
44. Peng Y, Xu W, Li J, Zhang M (2016) A fast and stable seam selection algorithm for video stitching based on seam temporal propagation constraint. In: Proceedings of the 3rd international conference on systems and informatics, pp 572–577
45. Perazzi F, Sorkine-Hornung A, Zimmer H, Kaufmann P, Wang O, Watson S, Gross M (2015) Panoramic video from unstructured camera arrays. Comput Graph Forum 34(2):57–68
46. Silva R, Feijó B, Gomes PB, Frensh T, Monteiro D (2016) Real time 360^o video stitching and streaming. In: ACM SIGGRAPH 2016 posters, p 70
47. Su T, Nie Y, Zhang Z, Sun H, Li G (2016) Video stitching for handheld inputs via combined video stabilization. In: Proceedings of SIGGRAPH ASIA technical briefs, p 25
48. Szeliski R (2006) Image alignment and stitching: a tutorial. Found Trends® Comput Graph Vis 2(1):1–104
49. Szeliski R, Shum HY (1997) Creating full view panoramic image mosaics and environment maps. In: Proceedings of the 24th international conference on computer graphics and interactive techniques, pp 251–258
50. Technologies S (2017) Beam. http://suitabletech.com
51. Tsui KM, Norton A, Brooks DJ, McCann E, Medvedev MS, Allspaw J, Suksawat S, Dalphond JM, Lunderville M, Yanco HA (2014) Iterative design of a semi-autonomous social telepresence robot research platform: a chronology. Intel Serv Robot 7(2):103–119
52. Xu B, Jia Y (2017) Wide-angle image stitching using multi-homography warping. In: Proceedings of the IEEE international conference on image processing, pp 1467–1471
53. Zaragoza J, Chin TJ, Brown MS, Suter D (2013) As-projective-as-possible image stitching with moving dlt. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2339–2346

54. Zhu Z, Lu J, Wang M, Zhang S, Martin R, Liu H, Hu S (2016) A comparative study of algorithms for realtime panoramic video blending. arXiv preprint arXiv:1606.00103

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yanmei Dong received the B.S., M.S., and Ph.D. degrees in computer science from Beijing Institute of Technology (BIT) in 2013, 2015, and 2020, respectively. She is now a lecturer at Tus College of Digit, Guangxi University of Science and Technology (GXUST). Her research interests lie in computer vision and human-computer interaction.

Mingtao Pei received the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2004. He is an Professor with the School of Computer Science, Beijing Institute of Technology. From 2009 to 2011, he was a Visiting Scholar with the Center for Image and Vision Science, University of California at Los Angeles, Los Angeles, CA, USA. His current research interests include computer vision with an emphasis on event recognition and machine learning. Dr. Pei is a member of China Computer Federation.

Lijia Zhang received the M.S. degree in computer science from Beijing Institute of Technology (BIT) in 2019. Her research interests include visual object tracking and image stitching.

Bin Xu received the B.S. and M.S. degrees in computer science from Beijing Institute of Technology (BIT) in 2015 and 2018, respectively. His research interests include image processing and machine intelligence.

Yuwei Wu received the Ph.D. degree in computer science from Beijing Institute of Technology (BIT), Beijing, China, in 2014. He is now an Assistant Professor at School of Computer Science, BIT. From August 2014 to August 2016, he was a postdoctoral research fellow at Rapid-Rich Object Search (ROSE) Lab, School of Electrical & Electronic Engineering (EEE), Nanyang Technological University (NTU), Singapore. He received outstanding Ph.D. Thesis award from BIT, and Distinguished Dissertation Award Nominee from China Association for Artificial Intelligence (CAAI).

Yunde Jia (M'11) received the B.S., M.S., and Ph.D. degrees from the Beijing Institute of Technology (BIT) in 1983, 1986, and 2000, respectively. He was a visiting scientist with the Robotics Institute, Carnegie Mellon University (CMU), from 1995 to 1997. He is currently a Professor with the School of Computer Science, BIT, and the team head of BIT innovation on vision and media computing. He serves as the director of Beijing Lab of Intelligent Information Technology. His interests include computer vision, vision-based HCI and HRI, and intelligent robotics.