SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2023, Vol. 66 112106:1–112106:16
https://doi.org/10.1007/s11432-021-3323-2

# Model-guided 3D stitching for augmented virtual environment

Zhong ZHOU[1], Ming MENG[1], Yi ZHOU[1*], Zhe ZHU[3] & Jingdi YOU[2]

[1]*State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing* 100191*, China;*
[2]*Beijing BigView Technology Co., Ltd., Beijing* 100088*, China;*
[3]*Department of Radiology, Duke University, Durham* NC 27708-0187*, USA*

Received 11 January 2021/Revised 1 June 2021/Accepted 30 June 2021/Published online 29 December 2022

**Abstract** Image and video stitching have made tremendous progress in the construction of wide field-of-view (FOV). However, some long-term challenges still exist, including wide baselines between cameras, large parallaxes, and low texture in overlapping areas. The augmented virtual environment (AVE) captures videos as live textures of 3D models in a virtual environment, and provides another 3D solution to overcome the aforementioned challenges. Existing AVE methods primarily follow from video projection, and cannot produce satisfactory stitching results compared with image stitching. In this paper, we propose a novel model-guided 3D stitching algorithm for AVE. The algorithm recovers an approximate 3D model for each video streaming and optimizes the warping of the models to meet the requirements of feature point matching of the 3D models from adjacent videos. Compared with previous state-of-the-art methods, experiment results illustrate that our method significantly improves the stitching quality.

**Keywords** 3D stitching, video model, multiple video visualization, augmented virtual environment

**Citation** Zhou Z, Meng M, Zhou Y, et al. Model-guided 3D stitching for augmented virtual environment. Sci China Inf Sci, 2023, 66(1): 112106, https://doi.org/10.1007/s11432-021-3323-2

## 1 Introduction

Recently, the use of digital images and videos has increased remarkably in surveillance and social networking. The quantity of media places a cognitive burden on users, particularly in tasks such as monitoring videos captured from massive camera networks. The most widely used method to display massive surveillance videos is to arrange them in a monitor matrix, which is not effective. Researchers have investigated alternative ways to effectively organize and visualize the videos captured from such networks. In previous decades, image and video stitching techniques have been studied since they can construct wide field-of-view (FOV) videos by stitching multiple images or videos. Some studies have applied a registered camera array [1–3], requiring intrinsic and extrinsic parameters of the cameras obtained in advance. This reduces reliance on feature correspondences and requires fixed relative positions of the cameras. Recent studies employ 2D meshes for optimizing alignment [4, 5], performing content-aware warping and stitching to reduce artifacts due to the parallax. Here, the input videos can be captured using unregistered camera arrays. First, an optimized warp [6–9] is calculated, and then mesh-based deformation [10, 11] is performed to improve the quality of alignment. However, some long-term challenges still exist, including wide baselines between cameras, large parallaxes, and low texture in overlapping areas.

Augmented virtual environment (AVE) techniques composite multiple videos to a 3D virtual environment and have a larger FOV [12–15]. Their major advantage is fusing multiple real-time video streaming using 3D models to have depth-consistent views. However, existing AVE methods mainly use video projection, which relies on 3D model accuracy and cannot produce satisfactory stitching results compared with image stitching.

To overcome these challenges, we present a novel approach beyond the previous image and video stitching methods in 2D canvas. Our model-guided 3D stitching algorithm develops single image-based

* Corresponding author (email: zhouyibuaa@buaa.edu.cn)

 

**Figure 1** (Color online) Left: simultaneous frames from 3 input videos. Center, right: stitched results rendered from 3 different viewpoints.

modeling [16] to recover 3D models of the scene. Then, these models are stitched in 3D using an energy minimization-based warp. Next, an optimal seam between adjacent models is calculated, and the mapping between each 2D video streaming and the 3D model is established. Finally, the stitched view can be produced using the original video as texture, as shown in Figure 1. Our work primarily aims at city surveillance and can be applied to other scenarios. Notably, the foreground object motion between different stationary cameras is not considered in AVE systems. Therefore, as with image stitching, we are concerned about stitching the video background frames in a frame-by-frame fashion.

Our algorithm requires the input of multiple synchronized videos of a scene captured from static cameras whose views overlap to some degree. A ground image, typically an aerial image from Google maps, provides a base ground for modeling. Suppose the scene includes several dominant planar structures (roads, buildings) and moving objects (vehicles and pedestrians) that are relatively small compared to the previous objects. Our approach is flexible since it does not require a registered camera array. The output can be written in a new format called image-based modeling type (IBMT), which can be rendered in real time and supports augmented reality applications and viewing.

Our approach projects 2D feature points in the regions where adjacent videos overlap into 3D spaces for matching. These matched feature points in 3D guide optimization-based warping for stitching. The application of 3D stitching instead of the 2D one used in existing methods makes our approach more robust for challenging setups with large parallax, as our experiments later demonstrate. Since our approach provides a high-quality output without registered camera arrays, it can use existing surveillance camera networks.

## 2 Related work

### 2.1 Image and video stitching

Traditional image-stitching methods align the input images by estimating a global 2D transformation and have been comprehensively discussed in [17, 18]. The underlying assumption is that either the image is captured from a fixed viewpoint or the scene is far from the viewpoint, i.e., the entire scene is considered approximately planar. However, such methods cannot handle cases when the input images have significant parallax. Several methods have been proposed to solve this problem; however, mesh-based optimization is the most promising. The advanced mesh-based methods are mainly divided into two categories: the local-adaptive warping and seam-driven blending methods.

Local-adaptive warping methods use local homography transformations instead of a global one. Early studies by Zaragoza et al. [4] split the input into a grid of cells and applied an as-projective-as-possible (APAP) warp to optimize alignment by estimating a homography that varies smoothly from cell to cell. Chang et al. [5] proposed the shape-preserving half-perspective method (SPHP), which reduces perspective distortion but is affected by unnatural rotation artifacts. Chen et al. [8] proposed the global similarity prior method (GSP) to improve the stitching result by combining local and global similarities. Li et al. [9] proposed a parallax-tolerant method from robust elastic warping (REW) for accurate and

efficient image stitching. Zheng et al. [19] proposed a novel projective-consistent plane-based image stitching method (PCPS) that achieves stitching results with few seams and projective distortion by dividing the overlapping regions of the input image into several projective-consistent planes using the normal vector orientation and reprojection error. Zhang et al. [20] proposed an efficient approach for content-preserving stitching with regular boundary constraints. However, it is nonfunctional on images with many lines in local regions. Lee et al. [21] proposed an image stitching algorithm that uses the warping residuals for images with large parallax. It partitions the input image into superpixels and adaptively warps each superpixel using the optimal homography to alleviate the parallax artifacts. Li et al. [22] proposed a local-adaptive image alignment method from a triangular facet approximation (TFA) to achieve accurate and efficient alignment of perspective and non-perspective images. However, it does not obtain appreciable stitching results for input images containing serious occlusions. Li et al. [23] explored the importance of semantic planar structures under perspective geometry and proposed an image-stitching method using semantic planar region consensus to obtain accurate local alignments and maintain transition naturalness. Although these methods can handle input images with moderate parallax, they fail for images with large parallax. Furthermore, a more common problem of these methods is ghosting artifacts. These artifacts are the consequence of moving objects appearing in different positions in neighboring images.

Seam-driven blending methods hides artifacts by selecting an optimal seam for stitching. Zhang et al. [7] proposed the parallax-tolerant image stitching (PTIS), which uses content-preserving warping to solve parallax. Zhang et al. [10] constructed a multi-viewpoint panorama from wide-baseline (WB) images, which eliminates incorrect matching using a direct linear transformation and estimates the optimal seam using a different map. Lin et al. [11] proposed a seam-guided local alignment method (SEAGULL), which accomplishes plausible warping by iteratively optimizing the seam via a structure-preserving strategy. Although these methods reduce ghosting artifacts, the output quality of these methods on images with large parallax remains poor.

Video stitching techniques, an extension of image stitching, have gained widespread research interest. Since our surveillance scenario has cameras in fixed positions, we mainly review video-stitching methods from static cameras. Jiang et al. [24] applied for the latest advances in parallax-tolerant image-stitching and video stabilization to perform spatiotemporal local warping and seam finding and solve the temporal coherence in video stitching. Perazzi et al. [25] combined global prealignment with warping adjustments to generate a panoramic video from unstructured camera arrays having a certain degree of parallax.

Although the aforementioned methods have made progress in image and video stitching, two major problems still exist. The first problem is the challenge associated with processing inputs having a small degree of overlap, particularly for real video surveillance, using these methods. The second problem is the challenge associated with reduced stitching quality as parallax in the input videos increases. Our approach targets these problems using the single image-based modeling [16] to recover a 3D model of the scene, which greatly affects the final stitching results. Our approach produces better stitching results than existing methods in the presence of small overlap and large parallax.

### 2.2 Augmented virtual environment

Various situation awareness techniques have been proposed for exploring video collections. They can be divided into two categories. The first one is the methods relying on associations between the videos, e.g., semantic relations [26] or corresponding locations in a 2D map [27]. The other is AVE methods, which fuse multiple videos in a 3D virtual environment to obtain a larger FOV [12–15, 28]. We propose that the latter methods provide more useful results since they can be viewed from a broad range of viewpoints, providing a more user-friendly viewing experience.

Existing AVE approaches are primarily based on video projections. Neumann et al. [13, 14] first elaborated AVE and proposed a prototype system. DeCamp et al. [15] applied the fisheye camera for the AVE system to build an indoor immersive system named as HouseFly. Li et al. [28] proposed a fast, topology-accounting method for multi-video fusion with 3D geographic information system scenes to ensure efficient fusion with 3D scenes. However, these methods face several long-term problems. One such difficulty is associated with the virtual-real alignment, which makes video boundary consistency with 3D models or other overlapping videos in the environment difficult. The other is the distortion problem in fusion results [29], consisting of several artifacts.

To tackle these problems, we proposed a single image-based modeling [16], where a single frame is

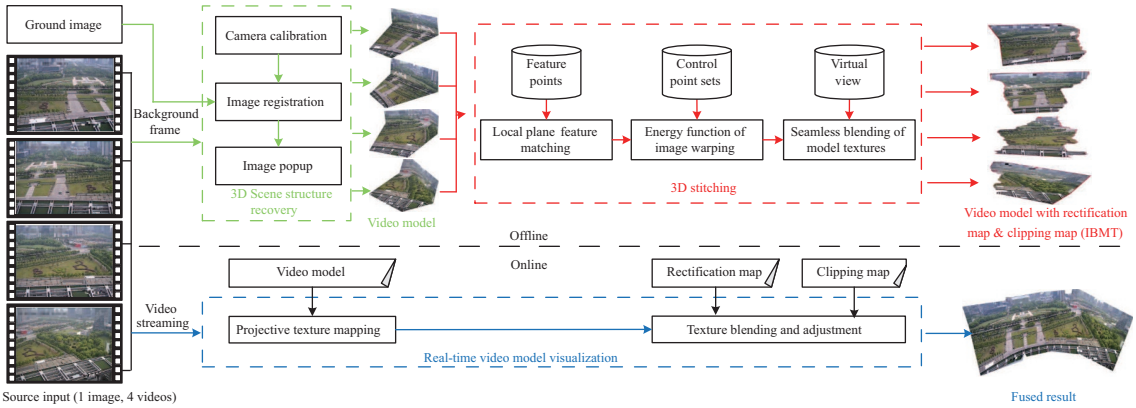


**Figure 2** (Color online) Model-guided 3D stitching for augmented virtual environment. Given a ground image and multiple video streamings with overlapping views (source input), first, we recover the 3D planar structures in their backgrounds (3D scene structure recovery), which are then stitched in 3D using mesh-based warping and seam-driven blending (3D stitching). The output video is rendered in real time from the stitched 3D model and the input video textures (real-time video rendering), guaranteeing high-quality fusion results (fused video).

selected from the video to recover a 3D model of the scene. Our approach generates a mixed virtual environment with video fusion using the recovered 3D models. For complex real-world scenes, where automatic modeling can be inaccurate and manual refinement is needed, the method in [30] provides efficient interactive modeling.

## 3 Overview

The input comprises multiple video streamings from fixed cameras with partly overlapping views and a ground image, which is an aerial image of a simple terrain without height in a 3D scene. Suppose the scene in each video streaming is dominated by several planes and that all moving objects are relatively small. Our goal is to seamlessly stitch these videos to provide a 3D model of the scene, which is rendered to provide a stitched output video from an arbitrary viewpoint in real time. From Figure 2, our approach comprises three basic steps: per video 3D scene structure recovery, 3D stitching, and real-time video rendering. The 3D scene structure recovery and 3D stitching are performed using only background frame, created by a classical fast background extraction method ViBe [31]. Furthermore, the 3D structure of the scene in each camera is determined by recovering its dominant planes, as with the approach in automatic photo pop-ups [32]. However, our approach differs because it is interactive, allowing us to deal with complex scenes in the real world for which automated methods fail.

For 3D stitching, we calculate the relative positions of different cameras by matching key points between their background frames. The matched feature point pairs in 2D are projected into the recovered 3D structure calculated in the previous step and used to guide an optimal warp of the recovered 3D model of each background frame for better alignment. Once aligned, adjacent models with a certain degree of overlap are evaluated. We calculate optimal seams between adjacent models to reduce artifacts during texturing. Then, the video model with rectification and clipping maps (stored in the IBMT format) is generated. Finally, the video content provides the texture of each model to achieve high-quality fusion results with large FOV. Since we have a 3D model of the whole scene and textures, we can render the scene from an arbitrary viewpoint.

## 4 Model-guided 3D stitching algorithm

### 4.1 3D scene structure recovery

The 3D background structure recovery process is performed separately for each input video. We conduct three steps to do so: vanishing point (VP)-based camera calibration, image registration and image popup. In order to handle complex real world scenes and ensure high modeling quality, when needed, we use an interactive refinement tool [30] in the image registration and image popup (3D model creation from an image) [33] steps.

**Camera calibration.** The essence of camera calibration is to calculate the camera intrinsic and extrinsic parameters. A simplified camera model is used in our camera calibration process, where the intrinsic parameter is focal length and extrinsic parameter includes rotation and translation matrix. The latter refers to the rotation and translation of the local coordinate system composed of the unit vector corresponding to the vanishing point with respect to the camera coordinate system. The procedure of our camera calibration method mainly consists of three steps: line set optimization, VP estimation, focal length and rotation matrix estimation. We adopt vanishing point-based calibration [34] to calculate focal length, rotation and translation matrix.

**Image registration.** The positions of different cameras in the unified world coordinate system are unknown, so we introduce the ground image as the unified 3D environment. On the basis of camera calibration, different cameras are registered into ground image through 2D-3D registration technology [16] to achieve the positioning of the space-time relationship between cameras. In other words, the captured images are registered into the 3D environment under the same perspective. The 3D scene is represented by as an existing 3D models with geographic information provided by the ground image (using an aerial image as a simple terrain without height in 3D scene). The images are automatically registered by matching 2D image line segment with 3D line segment from the 3D scene, including starting point matching, direction vector matching, and length matching.

**Image popup.** We employ single frame-based modeling method which recovers the 3D scene structure the background frame through image interactive popup to improve the quality of fusion results. Based on the camera pose, the depth of each point can be obtained by way of viewray intersection. We define the plane primitive and add it to our 3D scene through incremental interactive operations, which includes adding point and push/popup planes. In the process of creating a 3D plane (composed of A, B, C and D corresponding to $a$, $b$, $c$, and $d$ in the image), we first select the starting point $a$ in image editor, and calculate point A by the intersection of viewray at point $a$ and the 3D scene. Then, the diagonal point $c$ is selected based on the given normal $\boldsymbol{n}$ of this plane, and the point C is determined by the intersection of the viewray at point $c$ and the plane $\langle \mathrm{A}, \boldsymbol{n} \rangle$. In addition, points B and D are obtained by plane constraints and orthogonal relations. When we continue to add primitive, the background frame modeling is completed, defined as the video model. In this process, we try to make the modeled primitives cover the entire image.

### 4.2 3D stitching

Given the recovered 3D models for adjacent input videos, our next goal is to stitch them. We start by matching the feature points of the two models based on local planes, generating a guiding control point set per plane. We then align the adjacent models by mesh-based warping driven by these guiding control point sets. Finally, seam-driven blending is performed based on the constructed virtual view.

**Local plane feature matching.** We extract and match feature points from the input texture images for adjacent video models by combining global matching under the local-plane constraint. Our local matching strategy helps to reduce the matching error and refine the matching results. Based on this, we calculate their guiding control points for warping.

(1) Feature point matching. As in many previous methods, we first use SIFT [35] to detect the feature points. We match them between the two original images ($I_{\mathrm{A}}$, $I_{\mathrm{B}}$), to produce an initial matching result. Next, several planes are separately recovered from $I_{\mathrm{A}}$ and $I_{\mathrm{B}}$ by image popup (shown in thin green lines in Figure 3(a)), generating the corresponding video model $M_{\mathrm{A}}$, $M_{\mathrm{B}}$. The corresponding local plane sets are $\mathrm{LP_A}$=$\{\mathrm{lp}_1, \ldots, \mathrm{lp}_i, \ldots, \mathrm{lp}_m\}$ and $\mathrm{LP_B}$=$\{\mathrm{lp}_1, \ldots, \mathrm{lp}_j, \ldots, \mathrm{lp}_n\}$, where $m$ and $n$ are the numbers of local planes in video models $M_{\mathrm{A}}$, $M_{\mathrm{B}}$, respectively. Finally, we find the corresponding local plane pairs (shown in red quadrilaterals in Figure 3(a)) in the two sets, and reject mismatched points for each local plane pair by RANSAC [36], generating the dense, uniformly distributed, refined matching feature points for image warping.

(2) Generation of guiding control point sets. Based on the refined matching feature points, we interpolate the spatial locations of the matched feature points to generate guiding control point sets. They are used to guide the underlying warping (Figure 3(b)). Given one matched feature point pair ($p$, $q$) in the original images, we compute their projection positions $P$ and $Q$. They are obtained by intersecting the projection rays with $M_{\mathrm{A}}$ and $M_{\mathrm{B}}$, where the rays pass the corresponding $p$ and $q$ starting from $C_1$ and $C_2$, respectively. Then the final projection position $T$ of the ($p$, $q$) is generated by intermediate interpolation $P$ and $Q$ with a weight $w = 0.5$. $w$ denotes the distances from the midpoint $T$ to $P$ and

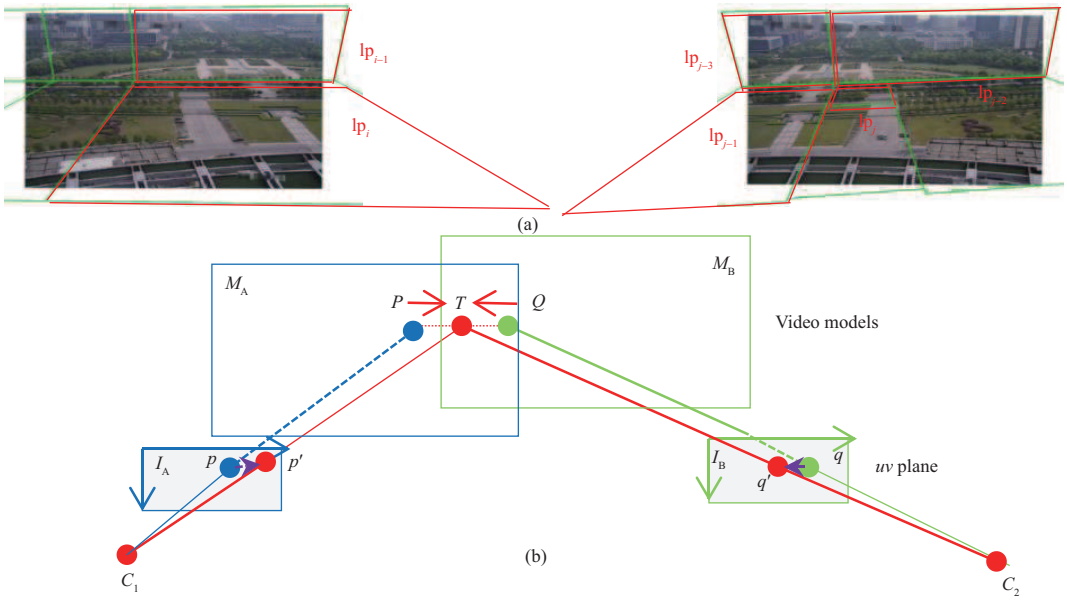


**Figure 3** (Color online) Generation of guiding control point sets based on local plane feature matching. (a) Video models of original images $I_{\mathrm{A}}$ and $I_{\mathrm{B}}$, denoted as $M_{\mathrm{A}}$ and $M_{\mathrm{B}}$, composed of multiple local planes (marked by thin green lines). We refine the matched feature points by the chosen local plane pairs, such as ($\mathrm{lp}_i$, $\mathrm{lp}_j$), ($\mathrm{lp}_i$, $\mathrm{lp}_{j-1}$) (marked by red quadrilaterals). (b) The guiding control point sets of the refined matching feature points are generated by projection and back-projection.

$T$ to $Q$ are equal. Finally, we separately project back $T$ to the original images $I_{\mathrm{A}}$ and $I_{\mathrm{B}}$, obtaining the corresponding guiding control point $p'$ and $q'$. They are added to the guiding control point sets used as the local alignment term for warping.

**Energy function of image warping.** The local plane feature matching provides control point sets to guide the warping. We use a mesh-based warping algorithm with a carefully chosen energy function. We generate an initial regular grid $V = [x_1 y_1, \ldots, x_m y_m]^{\mathrm{T}}$ for each input background image, where $m$ is the index number of grid vertices. Since a control point can be represented by its 4 neighborhood vertices, we define an energy function of 4 terms using the positions of the control points as constraints.

(1) Local alignment term. A local alignment term is used to ensure alignment quality by back-projecting matched feature points in 2D to the same point in 3D. Note that our algorithm performs local alignment using the guiding control point sets from two model textures separately. Next, we align $p$ to its guiding control point $p'$ in $I_{\mathrm{A}}$, and $q$ to $q'$ in $I_{\mathrm{B}}$ (the purple arrow in Figure 3(b)) after warping. We define the local alignment term on $I_{\mathrm{A}}$ as (1), and the same for $I_{\mathrm{B}}$.

$$E_a(V) = \sum_{(p,q)\in S_{(\mathrm{A,B})}} (\|\Phi(p) - p'\|^2 + \|\Phi(q) - q'\|^2), \tag{1}$$

where $S_{(\mathrm{A,B})}$ is a set containing matched feature points of image pair $(I_{\mathrm{A}}, I_{\mathrm{B}})$, and $\Phi(p)$ and $\Phi(q)$ give the warped position of matched feature points $p$ and $q$, respectively. $\Phi(p) = \sum_{t=1}^{4} w_t v_t$, where $v_t \in V$ are the four vertices of the warped grid cell containing $p$ and $w_t$ denotes the corresponding bilinear weights found by bilinear interpolation on the original grid. The position of the corresponding control point $p'$ for a given $p$ is defined as

$$p' = \Pi_{C_1}(\mathcal{F}(\Pi_{C_1}^{-1}(p), \Pi_{C_2}^{-1}(q))), \tag{2}$$

where $C_1$ and $C_2$ are camera locations of cameras 1 and 2. $\Pi$ ($\mathbb{R}^3 \longmapsto \mathbb{R}^2$) is the camera projection model from 3D space to image space, and $\Pi^{-1}$ (image space to 3D space) is its inverse. $\mathcal{F}(\Pi_{C_1}^{-1}(p), \Pi_{C_2}^{-1}(q))$ returns the linear interpolation of two points, which refers to the corresponding 3D point obtained by projecting the two matched feature points ($p$, $q$) into 3D space.

(2) Similarity transformation term. The local alignment term only constrains grid cells with matching feature points. According to [11], we adopt a similarity transformation term to ensure the naturalness of the warped image by ensuring similar transformations at adjacent vertices of the grid. $(v_1, v_2, v_3, v_4)$ are the four vertices of the initial grid cell that the matched feature point sits in. $(\hat{v}_1,\hat{v}_2,\hat{v}_3,\hat{v}_4)$ and $(v'_1,v'_2,v'_3,v'_4)$ are the four vertices of the actual and desired warped grid cell, respectively. Each grid cell

is divided into two triangles. Each vertex can be represented by a local coordinate system constructed by a vector between the other two vertices and the $R_{90}$. We calculate the local coordinates $(u, v)$ of $v_1$ in the local coordinate system defined by $v_2$ and $v_3$:

$$v_1 = v_3 + u(v_2 - v_3) + vR_{90}(v_2 - v_3), \tag{3}$$

where $R_{90} = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$. Given $u, v$, $\hat{v}_2$ and $\hat{v}_3$, the desired location for $\hat{v}_1$ is defined as

$$v_1' = \hat{v}_3 + u(\hat{v}_2 - \hat{v}_3) + vR_{90}(\hat{v}_2 - \hat{v}_3), \tag{4}$$

where $v_2'$ and $v_3'$ can be defined similarly. Then we estimate the error associated with the triangle based on $v_1'$, $v_2'$ and $v_3'$ as

$$C_{\text{tri}} = \sum_{i=1,2,3} \left\| v_i' - \hat{v}_i \right\|^2. \tag{5}$$

Finally, we take the error sum of all triangles as the error of the entire grid, and then the total similar transformation term is defined as

$$E_{\text{st}}(V) = \sum_{i=\text{A}}^{\{\text{A,B}\}} \sum_{j=1}^{T_i} C_{\text{tri}}^{i,j}, \tag{6}$$

where $T_i$ is the number of triangles of $I_i$. $C_{\text{tri}}^{i,j}$ represents the error value of the $j$th triangle in image $I_i$.

(3) Scale-preserving term. The scale-preserving term is defined in the same way as in [10]. It is used to constrain the minimum global scale change of the image and ensure no local perspective distortion through calculating the ideal scale of each image, measured by the four edges of the image. First, we calculate the relative scaling factor for matched image pair $(I_{\text{A}}, I_{\text{B}})$ according to the feature points:

$$\mathfrak{R}_{\text{AB}} = \frac{\mathcal{P}_{\text{A}}}{\mathcal{P}_{\text{B}}}, \tag{7}$$

where $\mathcal{P}_{\text{A}}$ is the perimeter of convex polygon, built on the feature points from $I_{\text{A}}$. $\mathcal{P}_{\text{B}}$ is the perimeter of convex polygon in $I_{\text{B}}$ corresponding to the convex polygon in $I_{\text{A}}$. Next we estimate the absolute scaling factor for $I_{\text{A}}$ and $I_{\text{B}}$ by minimizing

$$\arg\min_{\mathcal{K}} \sum_{(\text{A,B}) \in S_I} \left| \mathfrak{R}_{\text{AB}} \mathcal{K}_{\text{B}} - \mathcal{K}_{\text{A}} \right|^2 \qquad \text{s.t.} \sum_{\text{A} \in I} \mathcal{K}_{\text{A}} = N_I, \tag{8}$$

where $S_I$ is the set of the matched image pairs. $I$ is the set of images, and $N_I$ is the number of images. $\mathcal{K}_{\text{A}}$ and $\mathcal{K}_{\text{B}}$ are the absolute scaling factors of $I_{\text{A}}$ and $I_{\text{B}}$, respectively. For this conditional extreme value problem, we adopt the Lagrangian multiplier method to solve it. Finally, based on the absolute scaling factor, the scale-preserving term is defined as

$$\begin{aligned} E_{\text{sp}}(V) &= \sum_{I_{\text{A}} \in I} \left\| \delta(\hat{I}_{\text{A}}) - \mathcal{K}_{\text{A}} \delta(I_{\text{A}}) \right\|^2, \\ \delta(I_{\text{A}}) &= \begin{bmatrix} \|e_{\text{t}}\| & \|e_{\text{b}}\| \\ \|e_{\text{l}}\| & \|e_{\text{r}}\| \end{bmatrix}, \end{aligned} \tag{9}$$

where $\delta(I_{\text{A}})$ and $\delta(\hat{I}_{\text{A}})$ represent scale matrices of the original image $I_{\text{A}}$ and the warped image $\hat{I}_{\text{A}}$, respectively. $e_{\text{t}}$, $e_{\text{b}}$, $e_{\text{l}}$ and $e_{\text{r}}$ are the four (top, bottom, left and right) edges of $I_{\text{A}}$.

(4) Line-preserving term. To help constrain all lines to be straight after warping, we define a line-preserving term. Firstly, we employ the LSD algorithm [37] to detect the line segments and generate the corresponding lines set $L = \{l_1, \ldots, l_n\}$. Then we sample $m$ discrete points for each $l$ in $L$, denoted by $l = \{\tau_1, \ldots, \tau_m\}$. These points must be in different grid cells and represented by the grid vertices via bilinear interpolation to facilitate subsequent optimization. The corresponding warped points set is $l^* = \{\tau_1^*, \ldots, \tau_m^*\}$. Finally, the deviation of each line segment from the corresponding line is accumulated as the line-preserving term, defined as

$$E_l(V) = \sum_{l \in L} \sum_{i=1}^{m-1} \left( [a_l, b_l]_{\perp} \cdot (\Phi(\tau_i) - \Phi(\tau_{i+1})) \right), \tag{10}$$

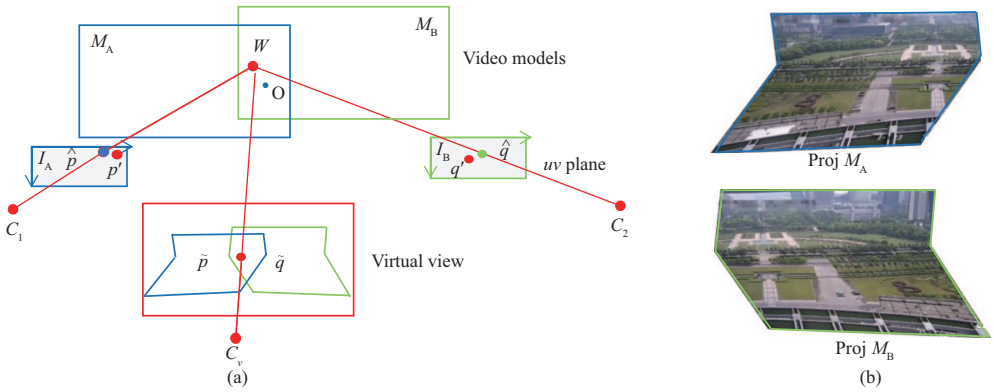


**Figure 4** (Color online) Construction of the virtual view for seam-driven blending. (a) Warped images are rendered from a new virtual view using the image-to-video model projection transformation. The corresponding pixel positions $\hat{p}$ and $\hat{q}$ in warped images are projected to their world space position $W$ by $C_1$ and $C_2$. Then $W$ is back-projected to the virtual view by $C_v$ to generate $\widetilde{p}$ and $\widetilde{q}$. (b) Warped images in the virtual view.

where $[a_l, b_l]_\perp$ is the orthogonal vector, calculated by $\tau_1^*$ and $\tau_m^*$. $\Phi(\tau_i)$ gives the warped position of the sampled point $\tau_i$. $\Phi(\tau_i) = \sum_{t=1}^{4} w_t v_t$, where $v_t \in V$ are the four vertices of the warped grid cell.

The total energy is defined as

$$E(V) = E_a(V) + \lambda_1 E_{\mathrm{st}}(V) + \lambda_2 E_{\mathrm{sp}}(V) + \lambda_3 E_l(V), \tag{11}$$

where $\lambda_1$, $\lambda_2$ and $\lambda_3$ weight corresponding terms, controlling their importance. In our experiments, we empirically set them to 1. Since the scale-preserving and line-preserving term (i.e., $E_{\mathrm{sp}}$ and $E_l$) are non-quadratic in the energy function $E(V)$, we replace these terms with their linear approximations and then update the results iteratively like the method in [10]. Next, the energy function $E(V)$ can be optimized by conjugate gradients, and the calculated grid is then used to guide warping [38].

**Seamless blending of model textures.** After warping, artifacts may still exist, caused by geometric and photometric errors in the overlapping regions. To reduce them, we conduct optimal seam-driven blending to refine the stitching result in the overlapping regions. Blending has two steps: virtual view construction and optimal seam generation. After that, we further blend images based on the smooth strip with alpha blending to reduce artifacts. We only consider blending two models. For multiple overlapping images, we have a default input order. Each step only stitches one input image to the already stitched image and performs seamless blending, and maintains a current image with a mask for the next input.

(1) Virtual view construction. We construct a virtual view as a reference view for seam-driven blending (Figure 4). We first calculate the center of the bounding box containing the two models, denoted as $O$, and then construct a line through $O$ that is the angle bisector of the two lines $C_1O$ and $C_2O$. We choose that point on the angle bisector with the minimum distance required to just capture the union of the two mapped images as the virtual camera position. $C_1$ and $C_2$ are the camera positions of the two input images. The warped images are projected to the constructed 3D models, and then matched 3D points are back-projected onto the corresponding sub-region of the virtual view for seamless blending.

(2) Optimal seam generation. Due to the lack of structural information, previous image composition methods usually find the optimal seam by measuring color differences on both sides of the seam. In our scenario, we have recovered the scene geometry, so we can measure both structural and textural differences to improve the seam quality. We consider the alignment error and color difference as follows.

Alignment error. Given two overlapping images $I_{\mathrm{A}}$ and $I_{\mathrm{B}}$, we first calculate the alignment errors for $S_{(\mathrm{A,B})}$ in virtual view and map them to [0, 1] via the Gaussian of

$$s_{(p,q)} = \exp\left(-\frac{\|\widetilde{p} - \widetilde{q}\|^2}{\sigma_1^2}\right), \tag{12}$$

where $(p, q) \in S_{(\mathrm{A,B})}$. $\widetilde{p} = \Pi_{C_v}(\Pi_{C_1}^{-1}(\Psi_{\mathrm{A}}(p)))$ and $\Psi_{\mathrm{A}}(p)$ is the warping function corresponding to $I_{\mathrm{A}}$. Similarly, we can figure out $\widetilde{q}$ in terms of $\Psi_{\mathrm{B}}(q)$ and $C_2$. $\sigma_1$ is a constant, empirically set to $0.003D$, where $D$ is the diagonal length of the image. When the alignment error is greater than $0.01D$, the corresponding feature point pair is considered unreliable and is discarded. Then, we compute the contribution of feature

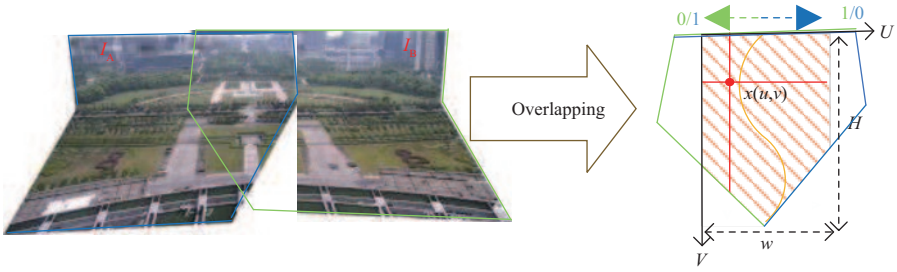


**Figure 5** (Color online) Mask calculation in the smooth strip with alpha blending. The orange curved line indicates the optimal seam. The mask value of smooth strip with alpha blending for $I_{\mathrm{A}}$ decreases from left to right, while for $I_{\mathrm{B}}$ it increases.

points to the overlapping region pixel $x$, defined as the weight coefficient:

$$w_{(p,x)} = \exp\left(-\frac{\|\Pi_{C_v}(\Pi_{C_1}^{-1}(p)) - \Pi_{C_v}(\Pi_{C_1}^{-1}(x))\|^2}{\sigma_2^2}\right), \tag{13}$$

where $\sigma_2 = 0.4D \cdot s_{(p,q)}$. Similarly, we can figure out $w_{(q,x)}$ in terms of $q$ and $C_2$. Next, we define the alignment error of $I_{\mathrm{A}}$ and $I_{\mathrm{B}}$, respectively, taking $I_{\mathrm{A}}$ as an example:

$$S_a^{I_{\mathrm{A}}}(x) = \frac{\sum_{p\in S_{(\mathrm{A},\mathrm{B})}} w_{(p,x)}^2 s_{(p,q)}}{\sum_{p\in S_{(\mathrm{A},\mathrm{B})}} w_{(p,x)}}. \tag{14}$$

Similarly, we repeat the above process for $I_{\mathrm{B}}$ to produce $S_a^{I_{\mathrm{B}}}(x)$. Finally, the final alignment error is defined as

$$S_a(x) = \frac{1}{2}(S_a^{I_{\mathrm{A}}}(x) + S_a^{I_{\mathrm{B}}}(x)). \tag{15}$$

Color difference. The color difference is calculated as the Euclidean distance of pixel RGB value in the overlapping region of the virtual view. The color difference value is scaled to $[0, 1]$ using a Gaussian function:

$$S_c(x) = \exp\left(-\frac{\|\Upsilon_{\mathrm{A}}(x'_{\mathrm{A}}) - \Upsilon_{\mathrm{B}}(x'_{\mathrm{B}}) - \mu\|^2}{\sigma^2}\right), \tag{16}$$

where $x'_{\mathrm{A}} = \Pi_{C_v}(\Pi_{C_1}^{-1}(x))$. $\Upsilon_{\mathrm{A}}$ and $\Upsilon_{\mathrm{B}}$ are pixel colors in the overlapping. $\mu$ and $\sigma$ are the mean and standard deviation of the pixel distance in the overlapping region.

The final energy function combines alignment error and color difference as follows:

$$S(x) = \frac{(S_a(x) + S_c(x)) - \min}{\max - \min}, \tag{17}$$

where max and min are the maximum and minimum values of $S_a(x) + S_c(x)$, respectively. $S_a(x)$ and $S_c(x) \in [0, 1]$. $S(x)$ denotes the consistency of $(I_{\mathrm{A}}, I_{\mathrm{B}})$ at pixel $x$, while the accumulation of $S(x)$ over the seam pixels gives the total consistency. We solve this optimization to give the seam using graph-cut [39].

After obtaining the optimal seam, illumination differences may still exist along the seam. To further reduce artifacts, we use the smooth strip with alpha blending arranged along the seam to blend the images (Figure 5). We calculate masks for the two images with mask values in $[0, 1]$. Pixels in the non-overlapping region of the mask are set to 1, while in the smooth strip with alpha blending their values are calculated as

$$D_{\mathrm{A}}(\tilde{x}) = \frac{(W - \mathcal{G}_u(\tilde{x})) \cdot (H - \mathcal{G}_v(\tilde{x}))}{W \cdot H}, \tag{18}$$

where $\tilde{x}$ indicates a pixel in the smooth strip with alpha blending, $\mathcal{G}_u(\tilde{x})$ and $\mathcal{G}_v(\tilde{x})$ calculate coordinates of $\tilde{x}$ in the smooth strip with alpha blending in $x$ and $y$ directions, respectively. $W$ and $H$ are the width and height of the belt, respectively. The mask $D_{\mathrm{B}}(\tilde{x})$ of the image $I_{\mathrm{B}}$ is calculated in the same way. The color of each pixel in the smooth strip with alpha blending is a linear combination of the color values in the two images:

$$C(\tilde{x}) = D_{\mathrm{A}}(\tilde{x}) \cdot \Upsilon_{\mathrm{A}}(\tilde{x}) + (1 - D_{\mathrm{A}}(\tilde{x})) \cdot \Upsilon_{\mathrm{B}}(\tilde{x}), \tag{19}$$

where $\Upsilon_{\mathrm{A}}(\tilde{x})$ and $\Upsilon_{\mathrm{B}}(\tilde{x})$ are the colors of $\tilde{x}$ in $I_{\mathrm{A}}$ and $I_{\mathrm{B}}$.

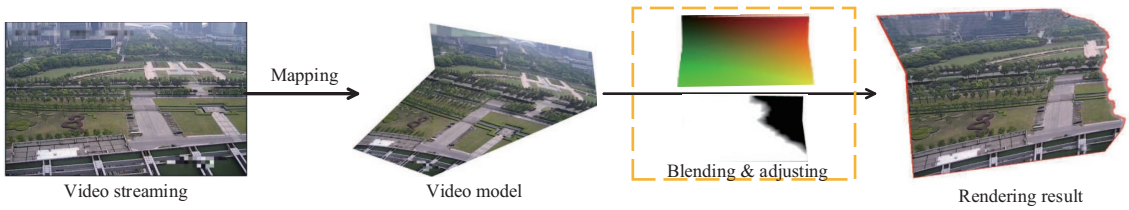


**Figure 6** (Color online) Rendering the output video. The input is the video captured in real time. First, the video content is mapped directly to the video model as a texture for each model. Then the texture is blended and adjusted based on the rectification map (storing the warped result, shown above) and the clipping map (storing the blended result, shown below) from the 3D stitching algorithm to achieve a high-quality rendering result.

## 5 Real-time video model visualization

Using the recovered 3D models as well as the stitched textures, we can now generate a 3D model with superimposed video textures in real time. The results are provided in a XML-described IBMT format file. It is designed for fast rendering of the video model.

For video model rendering, we first perform occlusion detection and calculate the occluded surface through shadow mapping [40]. This method can solve the problem that there may be multiple points on the path of light emission and the same image may be projected more than one in the scene. Then the projective texture mapping [41] is conducted to achieve real-time rendering. This strategy works well for a model recovered from a single input video. For models recovered from multiple input videos, ambiguity exists in overlap regions when there are moving objects. In this situation alpha blending is used to blend the textures of the two models in the overlapping region. We also make minor adjustments to the textures, including rectification and clipping, to further improve rendering results, as shown in Figure 6.

## 6 Experiments and discussion

One assumption of our approach is that input videos are captured by stationary cameras. Thus, we make comparisons with state-of-the-art image stitching methods. Synthetic datasets were gathered and used to evaluate our model-guided 3D stitching algorithm including 12 sets of images taken by us from video surveillance systems and 20 sets from publicly available images and videos in [4, 5, 7, 9, 11].

### 6.1 Evaluation metrics

We evaluated the following aspects of our 3D stitching quality: alignment quality, warping quality, seam quality and subjective user evaluations. We employed root mean squared error (RMSE) of an estimated warping $\Phi$ to quantify the alignment quality on the matched feature points set $S_{(\mathrm{A,B})}$, defined as

$$\mathrm{RMSE}_{\Phi} = \sqrt{\frac{\sum_{(p,q)\in S_{(\mathrm{A,B})}} \left(\|\Phi(p) - p'\|^2 + \|\Phi(q) - q'\|^2\right)}{N_{S_{(\mathrm{A,B})}}}}, \tag{20}$$

where $N_{S_{(\mathrm{A,B})}}$ is the number of matched feature point pairs in $S_{(\mathrm{A,B})}$. Warping quality is measured using the root mean squared distance (RMSD) of the estimated warping $\Phi$ on the matched feature points set $S_{(\mathrm{A,B})}$, defined as

$$\mathrm{RMSD}_{\Phi} = \sqrt{\frac{\sum_{(p,q)\in S_{(\mathrm{A,B})}} \left(\|\Phi(p) - p\|^2 + \|\Phi(q) - q\|^2\right)}{N_{S_{(\mathrm{A,B})}}}}. \tag{21}$$

In the blending stage, a visible seam may produce structure inconsistency visual artifacts. We adopt the assessment strategy in [11] to quantitatively measure seam quality. Similarity in the blending region is calculated by zero-base normalized cross correlation (ZNCC) on the optimal stitching seam with $N$ pixels, defined as

$$\mathrm{SQ} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\mathrm{ZNCC}(x_i) + 1}{2}\right), \tag{22}$$

**Table 1** Surveillance video data used in Figures 7–9[a)]

| Scene name | Basic parameters | | | | | Quantitative results | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Videos | Resolution (P) | Average baseline length (m) | Field of view (Mean/Max/Min) | Overlap (Mean/Max/Min) (%) | RMSE (Mean/Max/Min)↓ | RMSD (Mean/Max/Min)↓ | Modeling time (Avg.) (min) ↓ | Stitching time (Avg.) (min) ↓ | Stitching score (Avg.) ↑ |
| 'square' | 4 | 1080 | 46.2 | 37°/43°/30° | 23.7/33/17 | 0.379/0.498/0.161 | 0.145/0.351/0.041 | 4 | 7 | 7.75 |
| 'junction' | 4 | 1080 | 48.9 | 38°/43°/35° | 25.5/48/7 | 0.036/0.051/0.026 | 0.114/0.26/0.032 | 6 | 9 | 8.10 |
| 'street' | 33 | 1080 | 21.6 | 55°/56°/49° | 10/20/5 | 0.152/0.486/0.033 | 0.166/0.37/0.035 | 4 | 13 | 8.05 |

a) ↓: lower values denote better performance; ↑: higher values denote better performance.

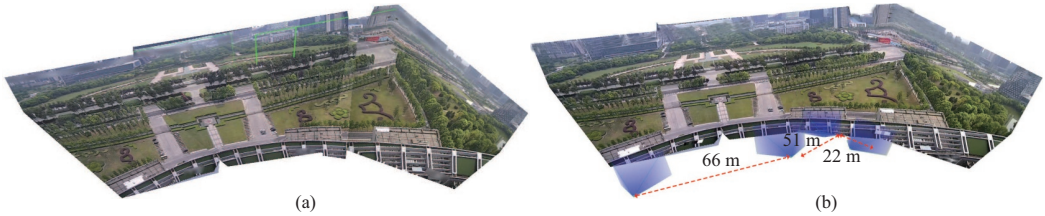


**Figure 7** (Color online) Stitching results for the 'square' scene. (a) Stitching result without blending. The total number of local planes is 45. (b) Final stitching result with blending. The baseline between adjacent cameras is marked as a red dotted line.

where SQ is the mean of the blending texture similarity over all seam pixels, in the interval $[0, 1]$. Smaller values indicate higher seam quality.

We also conducted a user evaluation to subjectively evaluate the results of our approach and other methods. 20 participants were involved. Half of them were in-lab students while the rest were students in other majors. 32 scenes were selected randomly and each time the stitching results from 8 methods were presented to the participants in random order who rated their quality from 1–10 (stitching score, larger meaning better perceptual quality). For each scene the scores of all the participants were averaged for each method to give a score for that method.

### 6.2 Performance analysis of our model-guided 3D stitching method

We validate the effectiveness and robustness of our algorithm in two ways. Firstly, we employed our stitching algorithm on challenging real world videos. Then, we performed a flexibility analysis on our approach, to verify that it can handle scenes with complex structures as well.

**Stitching quality.** We present 3D stitching results of our approach on 3 scenes captured from a video surveillance system. The 'square' scene and 'junction' scene are wide scenes taken from a high building, while the 'street' scene is a long narrow roadside scene viewed from light poles. Key characteristics of the scenes are provided in Table 1. Note that further details of the modeling, stitching time and results of each scene can be found in Appendixes A–E.

Figure 7 shows the stitching result for the 'square' scene. Our approach generates visually pleasing results for this case. Figure 8 shows a fused result for the 'junction' scene stitched from 4 videos. From left to right we see background frames of the input videos with modeling lines, the recovered models for each video with textures, stitched models with alpha blending and a comparison between our fused result without (above) and with (below) 3D stitching. Figure 9 shows another long scene example stitched from 33 input videos; the cameras used to capture these videos were mounted on one side of the street. Our approach works well in this case—the rendered result is shown in Figure 9(a). Details of the fused result for part of the scene are shown in Figure 9(b), with close-up views in Figures 9(c)–(e). We quantitatively evaluated our stitching results using both objective and subjective metrics, as shown in Table 1. Objective evaluation results indicate that our approach achieves high alignment accuracy while subjective evaluation results show that our approach can generate visually satisfying results.

**Modeling flexibility.** Our modeling method approximates the scene geometry using local planes. Appropriately choosing the number of planes can balance the quality and complexity of the modeling process. We explored the effect of increasing the number of planes used in the modeling process for a case with two scenes (denoted as scene1 and scene2), and gave the alignment results as well as their corresponding RMSE and RMSD. Results are shown in Figure 10. We first limited the number of planes for each scene to 3; the alignment result is shown in Figure 10(a). In this case there is serious misalignment and structural distortion (marked by a red rectangle). Then we increased the number of local plane for scene1 to 9 while decreasing the number of local plane for scene2 to 2; the alignment results are shown in Figure 10(b). This slightly improved the alignment quality, giving a lower alignment error (RMSE = 4.168) and a smaller warping degree (RMSD = 1.882), but misalignments still exist (marked by the red

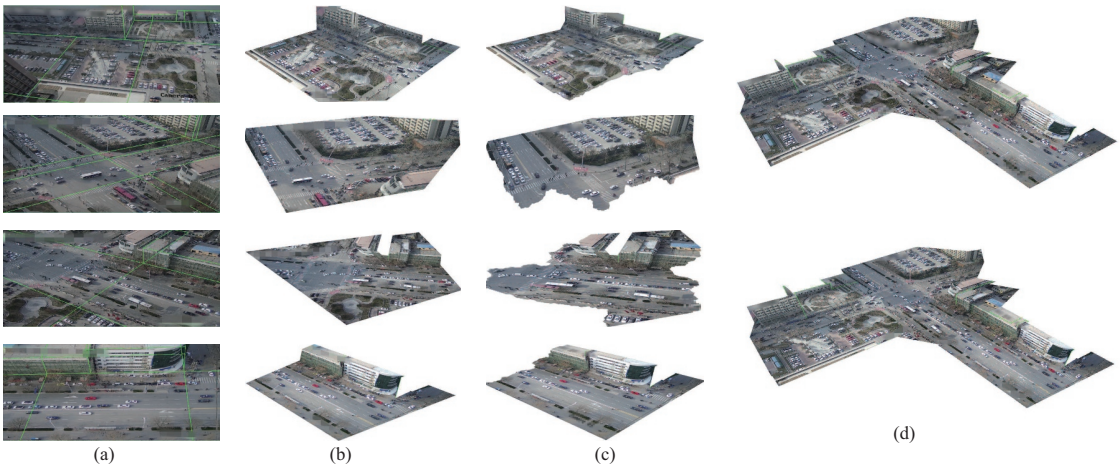


**Figure 8** (Color online) Modeling and stitching performance for the 'junction' scene. (a) Input frames with line drawings. (b) Image-based modeling results. The total number of local planes is 45. (c) Blending results after stitching adjacent frames. (d) The fused results without 3D stitching (above), and with 3D stitching (below).

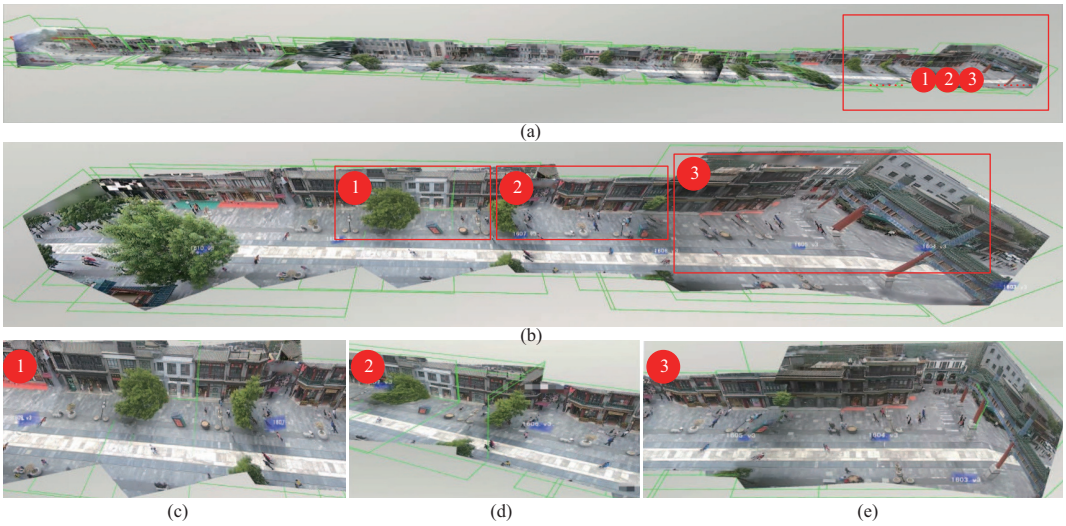


**Figure 9** (Color online) (a) The fused results for the 'street' scene using 33 videos. The total number of local planes is 70. (b) Close-up view of 'street' scene using 7 videos. (c–e) Zoom-in details of three parts of (b).

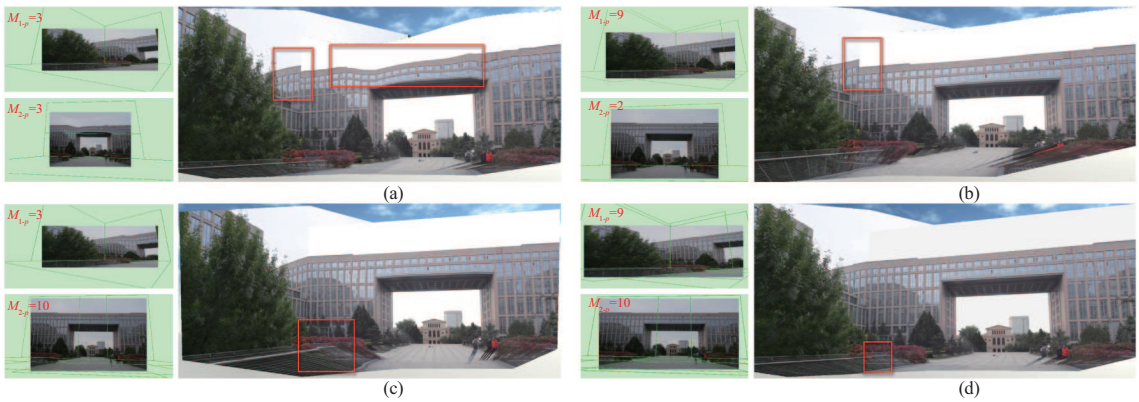


**Figure 10** (Color online) Alignment results with differing numbers of local planes. Each case includes two recovered video models ($M_{1\text{-}p}$ and $M_{2\text{-}p}$) and their warped model, evaluated by RMSE and RMSD. The misalignment and perspective distortion are indicated by red rectangles respectively. '$M_{1\text{-}p} = 3$' indicates the number of local planes of the first video model is 3. (a) RMSE = 6.554, RMSD = 2.726; (b) RMSE = 4.168, RMSD = 1.882; (c) RMSE = 3.926, RMSD = 1.508; (d) RMSE = 0.242, RMSD = 0.095.

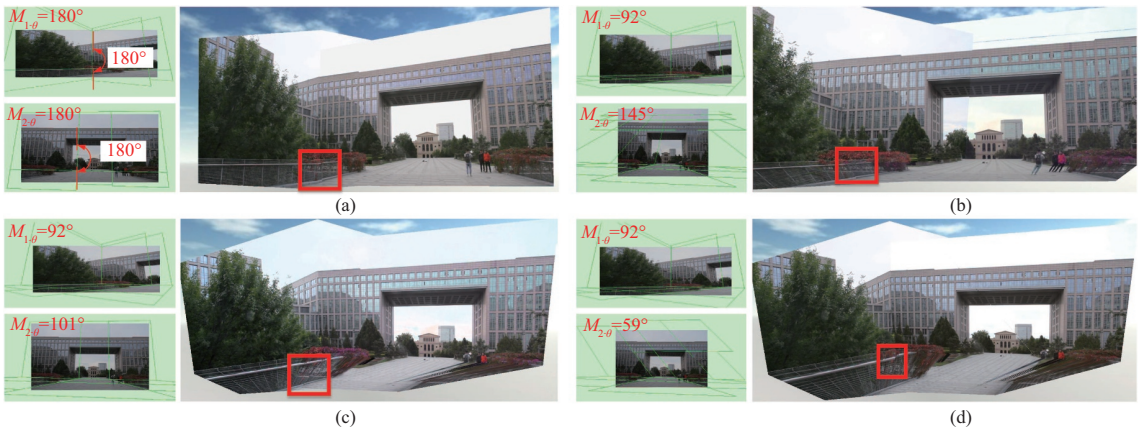


**Figure 11** (Color online) Alignment results with differing local plane orientations (an angle between the facade and the ground, marked by a red double arrow). Each case includes two recovered video models ($M_{1\text{-}\theta}$ and $M_{2\text{-}\theta}$) and their warped model, evaluated by RMSE and RMSD. The misalignment is indicated by a red rectangle. '$M_{1\text{-}\theta}$ = 180°' indicates the angle between the facade and the ground of the first video model is 180°. (a) RMSE = 0.244, RMSD = 0.164; (b) RMSE = 0.223, RMSD = 0.086; (c) RMSE = 0.213, RMSD = 0.073; (d) RMSE = 0.217, RMSD = 0.088.

rectangle). By setting the number of local plane of scene1 to 3 and increasing the number of local plane of scene2 to 10, we achieved similar result as shown in Figure 10(c). Finally we set the number of local plane for scene1 to 9 and for scene2 to 10, which lead to a perceptually satisfactory alignment result, as shown in Figure 10(d). This setting has low alignment error (RMSE = 0.242) and warping degree (RMSD = 0.095).

We can also choose different local plane orientations relative to the ground, the choice being parallel, obtuse angle, right angle or acute angle. Parallel is the most basic choice which always produces the worst result, as shown in Figure 11(a). We also show results using different angle combinations for two input scenes in Figures 11(b)–(d), and provide the corresponding RMSE and RMSD values. The minor misalignments are marked by red rectangles. These alignment choices produce different results and in practice choices should be made based on the scene structures.

### 6.3 Comparison with state-of-the-art image stitching methods

We compared our stitching algorithm with seven state-of-the-art image stitching methods on the synthetic datasets. The compared methods include such local-adaptive warping methods as APAP [4], SPHP [5], GSP [8], REW [9] and TFA [22], and such seam-driven blending methods as WB [10], PTIS [7], SEAGULL [11]. Since the source code of PTIS is not published, we only compare with its released stitching results. We also quantitatively evaluated the results using the methods in [10, 11].

**Results on our video dataset.** Figure 12(a) shows results for the 'square' scene with four input videos. Severe misalignment and perspective distortion (red rectangles) exist in the results from SPHP, APAP, GSP, REW, TFA, SEAGULL and WB. Additionally, the result of REW suffers from ghosting artifacts. Our approach works well on this scene (green rectangles), as there is no visible parallax error and shape distortion in the stitching result.

**Results on public datasets.** Figure 12(b) shows stitching results for seven methods for a scene from the SEAGULL dataset with two input images. Obvious misalignments exist in the results of SEAGULL and WB, and ghosting artifacts exist in the results of SPHP, APAP, GSP, REW, TFA and WB. In some results, structural objects such as the pillar are distorted. Our approach outperforms these methods, producing a better stitching result as shown in the last row.

**Quantitative evaluation.** We quantitatively evaluate the results of WB, SEAGULL and our approach as shown in Table 2. Twelve scenes were tested in this evaluation. Our approach achieves best results under the RMSE metric for all scenes, where our approach has the smallest alignment error. For the RMSD metric, we only evaluate SEAGULL and our approach, since WB employs the reference plane to generate the stitching, and original feature points in the corresponding image are not in the same coordinate system. Our approach has better RMSD than SEAGULL in all cases: our approach produces better warping quality than SEAGULL. The stitching score shows that participants preferred our results over the results of WB and SEAGULL. We also compared our approach with WB and SEAGULL on

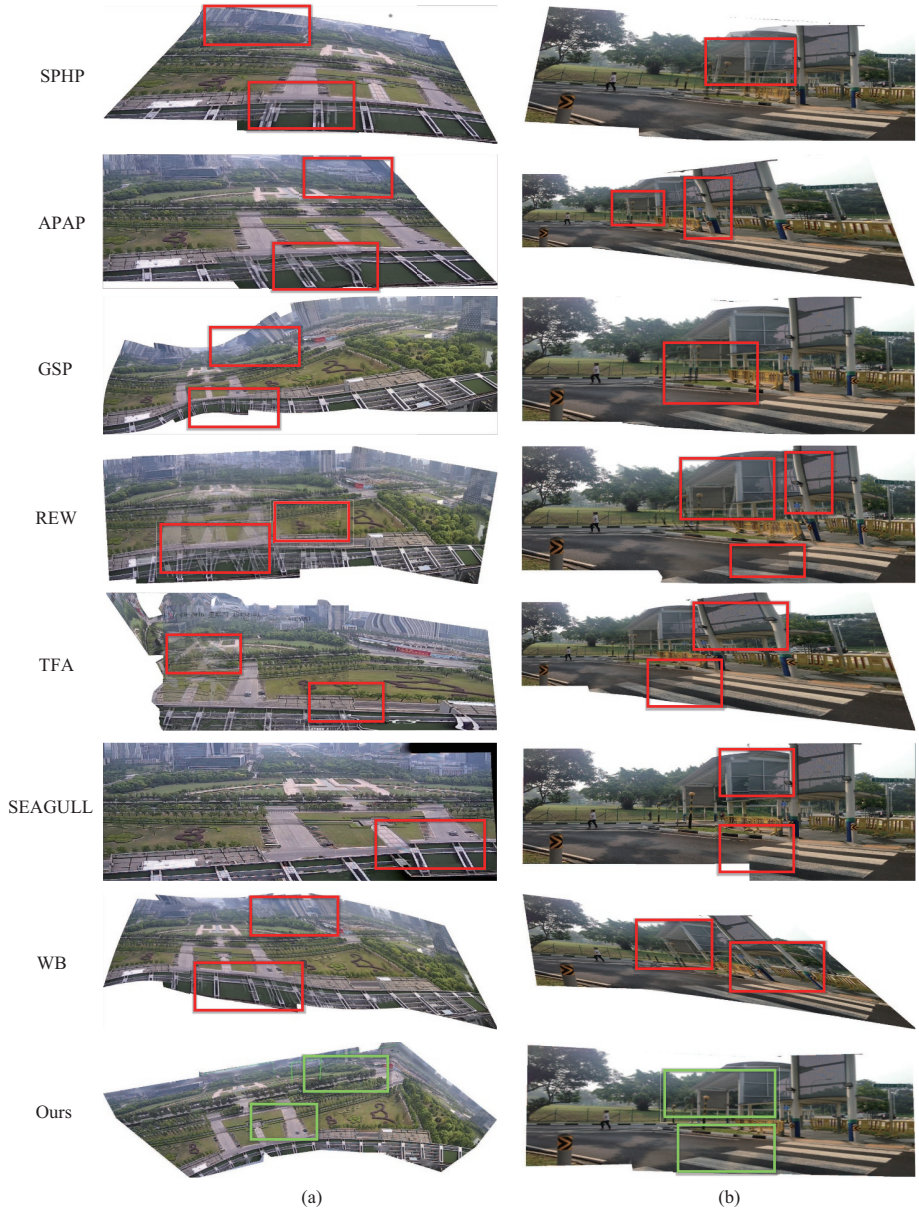


**Figure 12** (Color online) Comparison with state-of-the-art methods on our dataset and public datasets. (a) The 'square' scene from our dataset. (b) SEAGULL scene from a public dataset. Top to bottom: results of SPHP, APAP, GSP, REW, TFA, SEAGULL, WB and our approach. In each result, the misalignment, perspective distortion and ghosting artifacts are indicated by red rectangles. A green rectangle emphasises our plausible result. The total numbers of local planes in our results are (a) 45 and (b) 8.

these 12 scenes using the SQ metric. In all cases except for the third, our results have smaller SQ than SEAGULL and WB: our approach provides better seam quality than WB and SEAGULL.

### 6.4 Limitations

Our approach achieves plausible results with less distortion and misalignment for large parallax videos. The approach has shown its effectiveness and robustness. However, it also has some limitations. (i) The distance from the camera to moving objects needs to be far. Since moving objects are not modeled, rendering will give incorrect depths for them. If they are far, this effect is tolerable. However, when near, such objects will be rendered with the depths of certain planes, resulting in stretching distortion for viewpoints far from the original camera. (ii) Our approach cannot deal with videos from cameras with opposing orientations. If an object is near the camera, cameras with opposite orientations will capture the object from different directions, and stitching results in overlap areas will not be satisfactory. To

**Table 2** Quantitative comparison of [10, 11] and our approach on twelve datasets[a)]

| | | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Image/video number | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | Resolution | 1920 × 1080 | 1920 × 1080 | 1920 × 1080 | 1000 × 750 | 1000 × 750 | 1000 × 750 | 1000 × 750 | 1000 × 620 | 1000 × 620 | 1000 × 620 | 653 × 490 | 653 × 490 |
| RMSE ↓ | WB [10] | 24.14 | 80.90 | 1.02 | 1.84 | 1.63 | 7.73 | 10.55 | 2.32 | 0.68 | 2.20 | 2.16 | 0.36 |
| | SEAGULL [11] | 22.1 | Fail | 1.42 | 1.87 | 1.32 | 1.02 | 3.95 | 1.04 | 0.14 | 4.05 | 1.84 | 2.14 |
| | Ours | **0.18** | **0.22** | **0.03** | **0.16** | **0.44** | **0.10** | **0.10** | **0.11** | **0.16** | **0.10** | **0.18** | **0.02** |
| RMSD ↓ | SEAGULL [11] | 12.7 | Fail | 2.70 | 1.39 | 0.75 | 1.23 | 3.97 | 6.50 | 0.13 | 0.33 | 2.33 | 1.71 |
| | Ours | **0.10** | **0.13** | **0.03** | **0.10** | **0.44** | **0.10** | **0.12** | **0.11** | **0.10** | **0.10** | **0.18** | **0.02** |
| Stitching Score (Avg.) ↑ | WB [10] | 5.7 | 2.8 | 3.5 | 6.9 | 2.6 | 4.5 | 6.5 | 6.2 | 4.3 | 6.5 | 7.1 | 2.7 |
| | SEAGULL [11] | 5.3 | Fail | 3.8 | 4.5 | 5.5 | 5.6 | 7.2 | 6.1 | 7.6 | 5.0 | 5.8 | 4.0 |
| | Ours | **8.3** | **8.1** | **8.3** | **8.2** | **7.5** | **7.9** | **8.1** | **8.1** | **8.4** | **8.1** | **7.7** | **8.5** |
| SQ ↓ | WB [10] | 0.475 | 0.487 | **0.212** | 0.440 | 0.490 | 0.420 | 0.440 | 0.435 | 0.205 | 0.479 | 0.326 | 0.224 |
| | SEAGULL [11] | 0.424 | Fail | 0.478 | 0.510 | 0.480 | 0.470 | 0.420 | 0.538 | 0.271 | 0.496 | 0.229 | 0.413 |
| | Ours | **0.275** | **0.344** | **0.212** | **0.320** | **0.300** | **0.360** | **0.250** | **0.265** | **0.178** | **0.280** | **0.110** | **0.221** |

a) #1–#3 are our own scenes, #4–#7 are from [11], #8–#10 are from [7], #11 is from [5] and #12 is from [4]. 'Fail' indicates stitching failed.

obtain better results, real-time 3D object reconstruction needs to be considered; this is a topic for future work. (iii) In situations where a strong resolution difference exists in the stitching zone from different videos, the results may be blurred.

## 7 Conclusion and future work

We proposed a novel model-guided 3D stitching method for augmented virtual environment method for cityscapes. Our approach is robust and can generate plausible results. The effectiveness and robustness of our approach are validated on collected video datasets and several public datasets. We compared our approach with seven state-of-the-art stitching methods: qualitative and quantitative evaluations show that our approach outperforms other methods. In future, we plan to use semantic feature matching techniques to improve the automation process of the modeling of complex scenes.

**Acknowledgements** This work was supported by National Key Research and Development Program of China (Grant Nos. 2018YFB2100601, 2018YFB2100602) and National Natural Science Foundation of China (Grant No. 61872023).

**Supporting information** Appendixes A–E. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

**References**

1 Anderson R, Gallup D, Barron J T, et al. Jump: virtual reality video. ACM Trans Graph, 2016, 35: 1–13
2 Matzen K, Cohen M F, Evans B, et al. Low-cost 360 stereo photography and video capture. ACM Trans Graph, 2017, 36: 148
3 Zhu Z, Lu J, Wang M, et al. A comparative study of algorithms for realtime panoramic video blending. IEEE Trans Image Process, 2018, 27: 2952–2965
4 Zaragoza J, Chin T J, Brown M S, et al. As-projective-as-possible image stitching with moving DLT. In: Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2013. 2339–2346
5 Chang C H, Sato Y, Chuang Y Y. Shape-preserving half-projective warps for image stitching. In: Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014. 3254–3261
6 Lin C C, Pankanti S U, Ramamurthy K N, et al. Adaptive as-natural-as-possible image stitching. In: Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015. 1155–1163
7 Zhang F, Liu F. Parallax-tolerant image stitching. In: Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014. 3262–3269
8 Chen Y S, Chuang Y Y. Natural image stitching with the global similarity prior. In: Proceedings of European Conference on Computer Vision, 2016. 186–201
9 Li J, Wang Z, Lai S, et al. Parallax-tolerant image stitching based on robust elastic warping. IEEE Trans Multimedia, 2018, 20: 1672–1687
10 Zhang G, He Y, Chen W, et al. Multi-viewpoint panorama construction with wide-baseline images. IEEE Trans Image Process, 2016, 25: 3099–3111
11 Lin K, Jiang N, Cheong L F, et al. SEAGULL: seam-guided local alignment for parallax-tolerant image stitching. In: Proceedings of European Conference on Computer Vision, 2016. 370–385
12 Sawhney H, Arpa A, Kumar R, et al. Video flashlights: real time rendering of multiple videos for immersive model visualization. In: Proceedings of Eurographics Workshop on Rendering, 2002. 157–168
13 Neumann U, You S, Hu J, et al. Augmented virtual environments (AVE): dynamic fusion of imagery and 3D models. In: Proceedings of IEEE Virtual Reality, 2003. 61–67
14 Sebe I O, Hu J, You S, et al. 3D video surveillance with augmented virtual environments. In: Proceedings of ACM SIGMM Workshop on Video Surveillance, 2003. 107–112
15 DeCamp P, Shaw G, Kubat R, et al. An immersive system for browsing and visualizing surveillance video. In: Proceedings of ACM International Conference on Multimedia, 2010. 371–380
16 Zhong Z, Jingdi Y, Jin Y, et al. Method for 3D scene structure modeling and camera registration from single image. US 20160249041 A1, 2016
17 Szeliski R. Image alignment and stitching: a tutorial. FNT Comput Graph Vision, 2007, 2: 1–104
18 Lyu W, Zhou Z, Chen L, et al. A survey on image and video stitching. Virtual Reality Intell Hardware, 2019, 1: 55–83

19 Zheng J, Wang Y, Wang H, et al. A novel projective-consistent plane based image stitching method. IEEE Trans Multimedia, 2019, 21: 2561–2575
20 Zhang Y, Lai Y K, Zhang F L. Content-preserving image stitching with piecewise rectangular boundary constraints. IEEE Trans Visual Comput Graph, 2021, 27: 3198–3212
21 Lee K-Y, Sim J-Y. Warping residual based image stitching for large parallax. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020. 8198–8206
22 Li J, Deng B, Tang R, et al. Local-adaptive image alignment based on triangular facet approximation. IEEE Trans Image Process, 2020, 29: 2356–2369
23 Li A, Guo J, Guo Y. Image stitching based on semantic planar region consensus. IEEE Trans Image Process, 2021, 30: 5545–5558
24 Jiang W, Gu J. Video stitching with spatial-temporal content-preserving warping. In: Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015. 42–48
25 Perazzi F, Sorkine-Hornung A, Zimmer H, et al. Panoramic video from unstructured camera arrays. Comput Graph Forum, 2015, 34: 57–68
26 Huang H, Liu H, Zhang L. VideoWeb: space-time aware presentation of a videoclip collection. IEEE J Emerg Sel Top Circ Syst, 2014, 4: 142–152
27 Tompkin J, Kim K I, Kautz J, et al. Videoscapes: exploring sparse, unstructured video collections. ACM Trans Graph, 2012, 31: 1–12
28 Li C, Liu Z, Zhao Z, et al. A fast fusion method for multi-videos with three-dimensional GIS scenes. Multimed Tools Appl, 2021, 80: 1671–1686
29 Meng M, Zhou Y, Tan C, et al. Viewpoint quality evaluation for augmented virtual environment. In: Pacific-Rim Conference on Multimedia. Cham: Springer, 2018. 223–234
30 Zhou Y, Cao M, You J, et al. MR video fusion: interactive 3D modeling and stitching on wide-baseline videos. In: Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology, 2018. 17
31 Barnich O, van Droogenbroeck M. ViBe: a universal background subtraction algorithm for video sequences. IEEE Trans Image Process, 2011, 20: 1709–1724
32 Hoiem D, Efros A A, Hebert M. Automatic photo pop-up. ACM Trans Graph, 2005, 24: 577–584
33 Hoiem D, Efros A A, Hebert M. Recovering surface layout from an image. Int J Comput Vis, 2007, 75: 151–172
34 Guillou E, Meneveaux D, Maisel E, et al. Using vanishing points for camera calibration and coarse 3D reconstruction from a single image. Visual Comput, 2000, 16: 396–410
35 Lowe D G. Distinctive image features from scale-invariant keypoints. Int J Comput Vision, 2004, 60: 91–110
36 Fischler M A, Bolles R C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM, 1981, 24: 381–395
37 von Gioi R G, Jakubowicz J, Morel J M, et al. LSD: a fast line segment detector with a false detection control. IEEE Trans Pattern Anal Mach Intell, 2010, 32: 722–732
38 Zhu Z, Huang H Z, Tan Z P, et al. Faithful completion of images of scenic landmarks using internet images. IEEE Trans Visual Comput Graph, 2016, 22: 1945–1958
39 Kwatra V, Schödl A, Essa I, et al. Graphcut textures: image and video synthesis using graph cuts. ACM Trans Graph, 2003, 22: 277–286
40 Segal M, Korobkin C, Widenfelt R V. Fast shadows and lighting effects using texture mapping. In: Proceedings of ACM International Conference on Computer Graphics and Interactive Techniques, 1992. 249–252
41 Debevec P, Yu Y, Borshukov G. Efficient view-dependent image-based rendering with projective texture-mapping. In: Proceedings of the 9th Eurographics Rendering Workshop, 1998. 105–116