Rectangling irregular videos by optimal spatio-temporal warping

Jin-Liang Wu¹, Jun-Jie Shi², and Lei Zhang² (✉)

© The Author(s) 2021.

Abstract Image and video processing based on geometric principles typically changes the rectangular shape of video frames to an irregular shape. This paper presents a warping based approach for rectangling such irregular frame boundaries in space and time, i.e., making them rectangular again. To reduce geometric distortion in the rectangling process, we employ content-preserving deformation of a mesh grid with line structures as constraints to warp the frames. To conform to the original inter-frame motion, we keep feature trajectory distribution as constraints during motion compensation to ensure stability after warping the frames. Such spatially and temporally optimized warps enable the output of regular rectangular boundaries for the video frames with low geometric distortion and jitter. Our experiments demonstrate that our approach can generate plausible video rectangling results in a variety of applications.

Keywords rectangling; warping; content-preserving

1 Introduction

In recent years, geometry techniques have been widely used in image and video processing, such as image resizing or retargeting [1], perspective editing [2, 3], video stabilization [4], and panoramic stitching [5–9]. Unlike traditional image and video processing based on pixel operations, geometry-driven methods typically adopt a mesh grid structure in the image plane and manipulate the grid points to drive the processing of the enclosed patches, enabling more flexible and coherent processing of the image and video content [10]. Generally, these methods achieve the

desired balance between efficiency and effectiveness, but they usually result in warping of the boundary of images or video frames when directly operating on grid points, changing the rectangular shape to an irregular boundary. Figure 1 shows two examples with irregular boundaries resulting from fish-eye video correction and panoramic video stitching. Because most display screens are rectangular, it is necessary to make images and videos with irregular boundaries rectangular for normal display on common screens. We call this process rectangling. This paper mainly addresses the problem of rectangling video frames with irregular boundaries.

Obviously, the most direct solution to video rectangling is to crop a rectangle from the input video, but this will lose information. Other methods use image and video completion or inpainting methods to fill the gaps between the irregular boundaries of frames and the output rectangles [11–14]. However, existing image and video completion methods are too brittle to deal with irregular videos in general, especially for fish-eye or stitched videos with a large field of vision, which are prone to disturbing artifacts after completion (see Fig. 1).

He et al. [15] proposed the use of image warping to rectangle panoramic images with irregular boundaries. Such a method can well fill the gaps along the irregular boundaries by warping the whole image, with content preservation; it is able to generate more natural transitions than image completion methods. This rectangling strategy has also been adopted for processing stereoscopic panoramas [7, 16], which has fewer requirements for temporal consistency involving feature correspondence and motion. In this paper, we further extend this method to irregular videos by rectangling frames in a spatially and temporally coherent manner. More importantly,

1 The 54th Research Institute of CETC, Shijiazhuang 050050, China. E-mail: jinliangwuzju@gmail.com.

2 School of Computer Science, Beijing Institute of Technology, Beijing 100081, China. E-mail: J.-J. Shi, shijunjie@outlook.com; L. Zhang, leizhang@bit.edu.cn (✉).

Manuscript received: 2021-01-11; accepted: 2021-03-01

Fig. 1 Irregular videos. Above: shape-corrected fish-eye video frame using Ref. [3]. Below: panorama video frame using Ref. [6]. 3rd column: rectangling results using our approach. 4th column: results using a video completion method.

our approach can not only deal with panoramic or stitched video, but also irregular video from other video processing tasks like perspective editing of fish-eye video.

The main contribution of our work lies in a warping based approach for rectangling irregular videos. In particular, the motion-aware deformation of the underlying mesh grid is constrained across frames and optimized in space and time with line structure preservation. This can significantly mitigate the shape distortion in rectangling irregular video. Our experiments demonstrate the effectiveness and efficiency of our approach using a variety of irregular videos generated in different applications.

2 Related work

In this section, we briefly review the work most closely related to our method, in the areas of image completion, video completion, and warping.

2.1 Image and video completion

Image completion methods can be employed to fill the gaps along the irregular boundaries during rectangling. Generally, these methods may be broadly classified into three categories: statistical, partial differential equation (PDE) based, and exemplar-based. Statistical methods are mostly used in texture synthesis, where statistical models like histograms [17], wavelet coefficients [18], etc., are employed to describe the color or structure of the images and fill the holes. But these methods are only good at filling the holes with simple textures.

PDE-based methods use a diffusion process to propagate hole boundary information to its interior. The diffusion process is usually described by a Laplacian equation [19], Poisson equation [20], or Navier–Stokes equation [21], which are unsuitable for processing large holes in images. Exemplar-based methods borrow some compatible patches from the input image itself, and fill the holes by aligning the patches with appearance consistency between neighboring patches [22–24]. Generally, exemplar-based methods are more capable of generating high-quality completion results than statistical and PDE-based methods, but can result in semantic ambiguity, especially for natural images.

It has been observed that introducing semantic cues to exemplar-based image completion helps to obtain plausible results. Such cues can be specified by user interaction [25, 26] or extracted from some large-scale dataset like Internet images [27–30]. Then exemplar patches are selected and aligned to satisfy the cues such that the completed holes are a good match to the context of the entire image.

The above image completion methods can be directly applied to video completion by filling the holes in each frame individually. However, this may lead to temporal inconsistency between successive frames, especially for videos with dynamic scenes. To obtain good completion in space and time, motion information needs to be considered when filling holes in different frames. Jia et al. [11] used motion tracking to ensure that consistent fragments fill regions at the same positions in neighboring frames, so generating

visually smooth completion results. Alternatively, the motion field can also be used when selecting the candidate patches to fill the holes [12, 31]. Although these methods demonstrate good behavior in filling small holes, failure can still occur when processing video with more complex scenes.

2.2 Image and video warping

Recent image warping methods typically use geometric principles by deforming an embedded mesh grid for the target shape, which then drives change in the image content. To keep the original video content, the warping is usually required to be shape-preserving [32, 33]. This idea has been widely used in many applications like image resizing [34], perspective editing [2, 35], etc. Image warping can also be used for rectangling images with irregular boundaries [15], which provides the effect of image completion, and has been extended to process stereoscopic panoramas [7, 16]. Actually, the method of Ref. [7] claims to be the first to deal with rectangling stereoscopic stitched images, while the method of Ref. [16] extends it to stereoscopic panoramic video. However, these two methods focus on disparity preservation, and lack explicit constraints for motion consistency or line structure correspondence between frames, to preserve the original motion in the warping. This can still result in distortion after warping frames. For video warping, temporal consistency should be considered when warping each frame, as done for video stabilization [4], fish-eye video correction [3], and video stitching [6].

In this paper, we extend the image rectangling method of Ref. [15] to process irregular videos based on the spatio-temporal optimization of the frame warps, which purports to “fill” the holes between the irregular boundaries and rectangles in the video frames for rectangling. To ensure spatio-temporal consistency, we propose the use of line structure preservation and motion preservation between adjacent frames for rectangling video frames. Here, the line preservation enables common line orientations by line matching between adjacent frames, while motion preservation avoids motion jitter when warping the original frames. These are the key ingredients of our method that differ from those for single image rectangling.

3 Building a spatio-temporally consistent grid

Our warping-based frame rectangling approach requires a consistent mesh grid structure for all frames, i.e., a mesh grid in each frame with the same number of grid points and connectivity, to drive the frame warping with spatial and temporal coherence. Here, we use a quad mesh to build the mesh grid structure and drive the warping of frames.

Because the input video has irregular frame boundaries, we have to construct a virtual domain fit to a rectangle, in which the mesh vertices can be correctly placed and used to embed the frame content. We adopt the mesh placement scheme in Ref. [15] to set up the quad mesh in each frame. Concretely, a set of extra seams is inserted into the irregular frames to construct a virtual rectangular domain (see Fig. 2(above)). Then, the method employs a local warping technique to deform the rectangular domain, where the quad mesh is deployed and warped back to the original irregular frame. This procedure can guarantee a valid quad mesh displacement in the irregular frame. For an input video $\mathcal{F} = \{I^t\}_{t=1}^T$, we denote the quad mesh in the t -th frame as $\mathcal{M}^t = \{\mathcal{V}^t, \mathcal{E}^t\}$, where $\mathcal{V}^t = \{V_{i,j}^t\}$ is the set of mesh vertices and $\mathcal{E}^t = \{< V_{i,j}^t, V_{i\pm 1, j\pm 1}^t >\}$ is the set of mesh edges in the t -th frame.

We assume the frames have fixed boundaries in the temporal sequence in this paper. Then, the mesh edges $\mathcal{E}^t$ ensure a common topology across the frames by their connectivities: the corresponding vertices sharing the same edge connections. However, the temporal differences between adjacent frames possibly make the corresponding vertex positions suffer from slight movements, i.e., $V_{i,j}^t \neq V_{i,j}^{t+1}$. To enable a consistent grid structure between adjacent frames, we need to further rectify the vertex positions to build a unified grid, in which each of its vertices has the same position across the frames. Here, we use the average mesh vertices of adjacent frames to construct the unified grid: $\sum_{t=1}^T V_{i,j}^t / T$, where T is the total number of video frames. Then, the corresponding vertices have the same positions in the space.

Consequently, by carefully setting the grid size, we can obtain a valid mesh deployed over all the frames (see Fig. 2(bottom)). In the following sections, we

Fig. 2 Above: per-frame quad mesh placement by the method of Ref. [15]. Red lines: inserted seams. Green lines: mesh edges. Below: the consistent quad mesh grid through four successive frames (7th–10th frames).

still use $\{V_{i,j}^t\}$ to denote the unified mesh vertices in every frame I^t , by which the irregular boundaries are warped to the rectangular shape.

4 Motion-aware content-preserving frame rectangling

4.1 Approach

With the embedded grid, we next find the optimal image warps that deform each frame to a rectangle as well as preserving the content of the input video. We denote the mesh vertices in the deformed frame as $\hat{V}^t = \{\hat{V}_{i,j}^t\}$. Their positions determine the image appearance and distortion after rectangling irregular frames. In our setup, the preferred content-preserving frame warp refers has two goals: (i) deformation should result in low geometric distortion (ii) the original video motion should be preserved with reduced jitter frame warping. Hence, we propose a novel energy function to formulate an optimal motion-aware content-preserving frame warping for optimisation to rectangle the irregular boundaries. We next elaborate the details of the energy function terms and its optimization.

4.2 Energy function

The overall energy function of our approach contains the following four terms to drive frame warping towards rectangling the boundaries in a spatially and temporally consistent manner.

4.2.1 Shape preservation

To preserve the local content of the original frame, we require the warping to induce as little geometric distortion as possible after rectangling.

Here, we follow the idea of as-similar-as-possible transformation, and define the shape preservation term as

$$E_s(\hat{V}^t) = \sum_{p \in Q_{i,j}} \|\hat{V}_{i,j}^t - (\hat{V}_{i,j+1}^t + u_p(\hat{V}_{i+1,j+1}^t - \hat{V}_{i,j+1}^t) + v_p R_{90}(\hat{V}_{i+1,j+1}^t - \hat{V}_{i,j+1}^t))\|^2 \quad (1)$$

where u_p and v_p are scales in the local coordinates system of quad $Q_{i,j}$ enclosed by $\{V_{i,j}^t, V_{i+1,j}^t, V_{i+1,j+1}^t, V_{i,j+1}^t\}$ in the original frame, and R_{90} is a 2×2 anti-clockwise rotation matrix, defined by $R_{90} = [0, -1; 1, 0]$ (see Fig. 3). Derivation of Eq. (1) can be found in Ref. [4], which ensures that a similarity transformation on the frame has the minimal geometric distortion.

4.2.2 Line preservation

As the human visual system is very sensitive to linear structures, we also add a line preservation term as in Ref. [15] to keep the orientations of lines after warping. But our approach differs from theirs by considering line correspondences between adjacent frames: corresponding lines in different frames should be warped in a temporally consistent manner.

Concretely, we first detect corresponding lines across multiple frames by using line matching techniques like Ref. [36] (see Fig. 4). We denote the lines as $L_h = \{l_h^t\}$, where h is the line index

Fig. 3 Shape preservation by as-similar-as-possible transformation with the same local coordinates.

Fig. 4 Line structure and matching between two successive frames. Corresponding lines have the same indices.

in the t -th frame. Each line of L_h should have the same orientation after frame warping. The line orientation range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ is quantized into $M = 50$ bins, and the lines are assigned to the corresponding bins to obtain $\{\theta_m\}_{m=1}^M$. Here, the corresponding lines between adjacent frames should share a common orientation when warping each frame individually, which is determined by line matching as shown in Fig. 4. This enables lines to preserve their common orientation after frame warping. Hence, the line preservation term is defined as

$$E_l^t(\hat{V}, \{\theta_m\}) = \frac{1}{N_L^t} \sum_j \|C_j(\theta_{m(j)}) \mathbf{e}_{q(j)}\|^2 \quad (2)$$

where N_L^t is the total number of lines in the t -th frame, $q(j)$ indicates the quad containing the line segment, $\mathbf{e}_{q(j)}$ is the difference vector between the end points of the line segment, and C_j is the rotation matrix corresponding to the orientation angle $\theta_{m(j)}$.

4.2.3 Motion preservation

The process of warping frames inevitably causes jitter by changing the original inter-frame motion inconsistently. Hence, we should introduce a motion preservation term to follow the original motion as much as possible; this is a major difference from the method of Ref. [15]. Here, we use trajectories detected by pyramidal Lucas–Kanade tracking methods like Refs. [37–39] to collect feature points and represent the motion based on the corresponding trajectories that start in the t -th frame and end in the $(t+s)$ -th frame, denoted by $T_j = \{P_j^k\}_{k=t}^{t+s}$. Here, P_j^k indicates the feature point of the j -th trajectory in the k -th

frame. Then, we want the inter-frame transformation of feature points to be rigid, so that motion structure is preserved after frame warping. Hence, we have the following motion preservation term based on constraining the configuration of feature points:

$$\begin{aligned} E_m^k(\hat{V}) &= \sum_j \|(P_j^k - P_j^{k-1}) - R_t^k \cdot (\hat{P}_j^k - \hat{P}_j^{k-1})\|^2 \\ &= \sum_j \|(P_j^k - P_j^{k-1}) - R_t^k \cdot (\mathcal{B}_j^k(\hat{V}^k) - \mathcal{B}_j^{k-1}(\hat{V}^{k-1}))\|^2 \end{aligned} \quad (3)$$

where $\mathcal{B}_j^k(\cdot)$ is the bilinear interpolation operator to represent each trajectory point enclosed by a quad with its four vertices, and R_t is a rotation matrix that preserves the relative positions of feature points. Intuitively, the motion preservation term imposes a rigid motion structure when warping the frames, such that the inter-frame motion can follow the original motion to avoid the jitter.

4.2.4 Boundary constraints

The aim of frame warping is to obtain a rectangular boundary for each frame. So we have to add the boundary constraints to the transformed vertices; we use the same form as in Ref. [15]. Let $\hat{V}_i^B = (x_i, y_i)$ be the vertices along the boundary of the rectangular frame, then we have

$$E_B(\hat{V}) = \sum_{\hat{V}_i \in L} x_i^2 + \sum_{\hat{V}_i \in R} (x_i - w)^2 + \sum_{\hat{V}_i \in T} y_i^2 + \sum_{\hat{V}_i \in B} (y_i - h)^2 \quad (4)$$

where $\{L, R, T, B\}$ is the left, right, top, and bottom boundary of the target rectangle, and w and h are the width and height of the rectangle, respectively.

With the above four terms, our frame rectangling energy function is defined as

$$E(\hat{V}) = E_S + \alpha E_L + \beta E_M + \gamma E_B \quad (5)$$

where α , β , and γ are weighting factors to control the trade-off between the terms. Typically, γ is set to a large value to obtain a rectangular boundary shape. In the experiments of this paper, we set $\alpha = 5$, $\beta = 10$, and $\gamma = 50$. Then, the warped frames are determined by changing the vertices to the new positions and driving the deformation of the corresponding quads, to make the resultant videos having rectangular boundaries.

4.3 Optimization

The energy function of Eq. (5) is non-linear with respect to its variables $\{V_{i,j}^t\}$, $\{\theta_{m(k)}\}$, and $\{R_t^k\}$. These variables should be determined in a unified

manner to obtain the optimal frame warps, which are then used to drive the deformation of grid positions for the rectangular boundary.

Here, to optimize the energy function, we resort to a two-step scheme with respect to $\{V_{i,j}^t\}$, and $\{\theta_{m(k)}\}$, and $\{R_t^k\}$ separately. Concretely, we iteratively compute the optimal solution for one variable by fixing the other variables as follows:

1. Fixing the values of line orientation $\{\theta_{m(k)}\}$ causes Eq. (5) to become a quadratic function with respect to $\{V_{i,j}^t\}$. We may compute its normal equation by setting the gradient to zero.

2. With the computed values of mesh vertices $\{V_{i,j}^t\}$, we update the line orientation by computing the new $\{\theta_{m(k)}\}$ and rotation matrix R . The best line orientation $\{\theta_{m(k)}\}$ can be computed by optimizing Eq. (2) with an iterative Newton's method. The best rotation matrix R can be computed by singular value decomposition (SVD) of Eq. (3).

The above two steps can be iterated until the changes in positions of mesh vertices is below a prescribed threshold. Finally, we obtain the new vertex positions which change the frames to be with the rectangular shape.

5 Experiments

5.1 Setting

We have implemented our algorithm and tested it on a variety of irregular videos that have the nearly fixed boundaries through their frame sequences. The purpose of these tests is to verify the effectiveness of our algorithm, especially for the spatio-temporal consistency between frames. The test videos were

produced by perspective editing of the fish-eye videos [2], panoramic stitching of videos captured by multiple unstructured cameras [6], etc. These usually have irregular boundary shapes that need to be rectangled for display adaption (see Figs. 1, 6, and 7).

All experiments were performed on a machine with an Intel Core i5-2400 3.1 GHz CPU and 8 GB RAM. We show some results on rectangling irregular videos and make a comparison with other methods. In our first two examples, we adopt distortion-corrected fish-eye videos as input, and then compare the video completion method and naive cropping method with our approach. In the second two examples, the inputs are panoramic videos from stitching multiple videos. Both have irregular boundaries that need to be rectified for the boundary shape of the rectangle. We encourage readers to watch the video demonstration in the Electronic Supplementary Material (ESM) to appreciate temporal differences in results.

5.2 Rectangling results

The irregular boundaries of Fig. 6 are generated by editing the fish-eye videos to correct spherical distortion, so are usually smooth within the irregular frames. Our approach can well recover a regular rectangle shape as well as preserving the structure, especially salient lines, after rectangling the boundaries.

The irregular boundaries of Fig. 7 are generated by stitching regular videos captured by multiple cameras, which usually leads to piecewise straight lines in the panoramic videos. In this case, our approach can also align the irregular boundaries to the rectangle without distortion of the structure,

Fig. 5 Rectangling results using frame-by-frame rectangling (above) and our algorithm (below). Flickering occurs in the frame-by-frame rectangling results, especially for the car (green ellipse).

Fig. 6 Comparison of rectangling distortion-corrected fish-eye videos. Top to bottom: input frames of two examples; rectangling results using our approach, DPR [16], a video completion method [12], and naive cropping.

especially near the boundaries. Our approach usually takes about 2.5 s to process one frame, which involves 4 iterations to obtain good rectangling results. Most of the computation time is consumed in quad mesh placement and iterative optimization to solve Eq. (5).

The use of spatially and temporally optimized warping achieves good temporal consistency in the transition between adjacent frames, which avoids the jitter resulting if we simply use frame-by-frame rectangling based on the method of Ref. [15] (see Fig. 5). It can be seen that the frame-by-frame rectangling results have obvious jitter, while our algorithm provides temporally consistent rectangling results.

5.3 Comparison with other rectangling methods

To demonstrate the superiority of our approach, we also compared our results with ones obtained by other video rectangling methods like disparity-preserving image rectangularization (DPR) [16], the classical video completion method [12], and naive cropping. For the DPR method, we ignore the

disparity constraint in the warping energy terms. Because DPR has fewer constraints on the line structure as well as line correspondences between frames, it can result in distortion: see Fig. 6. The video completion method of Ref. [12] is also able to generate regular frames with rectangular shape by filling the gaps. Typically, such methods attempt to find a set of patches or volumetric blocks from the video itself to cover the gaps with spatial-temporal coherence of color and structure. Therefore, the rectangled video can suffer from block repetition in the gaps, especially for regions with salient structure (see Figs. 6(a) and 7(a)). Naive cropping is simple to realise, and can avoid distortion in rectangling the irregular boundaries. But it usually incurs over-cropping especially for videos with very irregular boundaries, resulting in loss of visual information.

Instead, our approach directly deforms the frames to the rectangular boundaries with both spatial and temporal coherence to enable a smooth transition between successive frames. From the results of Figs. 6 and 7, it can be seen that the irregular boundaries

Fig. 7 Comparison of rectangling panoramic videos. Top to bottom: input frames of two examples; rectangling results using our approach, DPR [16], a video completion method [12], and naive cropping results.

are well rectangled with good shape preservation, providing appealing results after frames rectangling.

5.4 Evaluation

There is no standard to evaluate the performance of different rectangling methods. All of the above methods can generate videos with rectangular boundaries. For a more comprehensive evaluation of the rectangling results produced by different methods, we conducted a user study to evaluate their visual quality, following Ref. [7]. We asked 37 participants to grade the rectangling results of different methods using a score from 0 to 5. We collected 7 examples as the cases in the user study, to ensure a suitable watching time for the participants that did not cause visual fatigue. Four cases are from the examples in Figs. 6 and 7, and the other three cases are from

examples in the ESM. We asked participants about (i) visual comfort when watching the rectangled videos and (ii) freedom from artifacts that affect perception of video content. Here, the visual comfort includes a subjective opinion about the consistency of adjacent frames, which is important for the video rectangling results. Average scores for different methods are shown in Table 1. It can be seen that our method can generate results with better visual comfort and fewer artifacts.

5.5 Limitations and discussion

Although our approach is useful, it is not without limitations. Our approach requires spatially and temporally consistent grids to drive coherent frame warping. At present, we employ the simple average position of the quad mesh to set the grids. Although

Table 1 User study on the rectangling results from our method, DPR [16], video completion [12], and naive cropping. Numbers indicate average scores for visual comfort and freedom from artifacts respectively

	Fig. 6(a)	Fig. 6(b)	Fig. 7(a)	Fig. 7(b)	Suppl.1	Suppl.2	Suppl.3
Our method	4.61/4.79	4.81/4.78	4.51/4.71	4.53/4.56	4.64/4.72	4.22/4.25	4.57/4.62
DPR	3.65/4.01	3.99/4.13	4.25/4.03	4.14/4.11	4.02/4.09	4.19/4.21	4.11/4.15
Completion	1.43/1.41	3.12/3.43	1.23/1.50	2.52/2.85	2.11/2.21	3.56/3.62	1.89/1.93
Naive cropping	2.91/4.77	2.61/4.47	3.81/4.21	3.57/4.45	2.78/2.82	3.67/3.72	3.72/3.78

this can provide consistent mesh placement for videos with small motion between adjacent frames, it can fail for videos with large motions. In this case, we have to manually correct the position of the quad mesh for final consistent mesh placement. Furthermore, in the case of dynamic boundaries during the frame sequence, e.g., as arising in frames generated by video stabilization methods, our approach may fail to place the expected grids, which can cause disturbing artifacts after frame rectangling. In this case, a potential solution is to use a cross parameterization technique from geometry processing like Ref. [40], to insert extra vertices for consistent grid topology in space and time.

6 Conclusions

We have presented a warping based approach to transform irregular video frames generated by geometry-based image and video processing into the ones with rectangular boundaries. Our approach enables spatio-temporal consistency in warping frames towards rectangular boundaries due to the use of shape and motion preservation terms. Our experiments demonstrate the efficacy of our approach for rectangling irregular videos.

In future work, we plan to extend our approach to deal with videos with time-varying boundaries, e.g., videos generated by applying video stabilization algorithms. We also hope to accelerate the speed of quad mesh placement and iterative optimization by using parallel GPU programming techniques.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported by the National Natural Science Foundation of China under Grant Nos. 61922014 and 61772069.

Electronic Supplementary Material Supplementary

material is available in the online version of this article at <https://doi.org/10.1007/s41095-021-0222-z>.

References

- [1] Wang, Y. S.; Tai, C. L.; Sorkine, O.; Lee, T. Y. Optimized scale-and-stretch for image resizing. In: Proceedings of the ACM SIGGRAPH Asia 2008 Papers, Article No. 118, 2008.
- [2] Du, S. P.; Hu, S. M.; Martin, R. R. Changing perspective in stereoscopic images. *IEEE Transactions on Visualization and Computer Graphics* Vol. 19, No. 8, 1288–1297, 2013.
- [3] Wei, J.; Li, C. F.; Hu, S. M.; Martin, R. R.; Tai, C. L. Fisheye video correction. *IEEE Transactions on Visualization and Computer Graphics* Vol. 18, No. 10, 1771–1783, 2012.
- [4] Liu, F.; Gleicher, M.; Jin, H. L.; Agarwala, A. Content-preserving warps for 3D video stabilization. *ACM Transactions on Graphics* Vol. 28, No. 3, Article No. 44, 2009.
- [5] Levin, A.; Zomet, A.; Peleg, S.; Weiss, Y. Seamless image stitching in the gradient domain. In: *Computer Vision-ECCV 2004. Lecture Notes in Computer Science, Vol. 3024*. Pajdla, T.; Matas, J. Eds. Springer Berlin Heidelberg, 377–389, 2004.
- [6] Perazzi, F.; Sorkine-Hornung, A.; Zimmer, H.; Kaufmann, P.; Wang, O.; Watson, S.; Gross, M. Panoramic video from unstructured camera arrays. *Computer Graphics Forum* Vol. 34, No. 2, 57–68, 2015.
- [7] Zhang, Y.; Lai, Y. K.; Zhang, F. L. Stereoscopic image stitching with rectangular boundaries. *The Visual Computer* Vol. 35, Nos. 6–8, 823–835, 2019.
- [8] Zhang, F. L.; Barnes, C.; Zhang, H. T.; Zhao, J. H.; Salas, G. Coherent video generation for multiple hand-held cameras with dynamic foreground. *Computational Visual Media* Vol. 6, No. 3, 291–306, 2020.
- [9] Zhang, Y.; Lai, Y. K.; Zhang, F. L. Content-preserving image stitching with piecewise rectangular boundary constraints. *IEEE Transactions on Visualization and Computer Graphics* doi: 10.1109/TVCG.2020.2965097, 2020.

- [10] Hu, S. M.; Chen, T.; Xu, K.; Cheng, M. M.; Martin, R. R. Internet visual media processing: A survey with graphics and vision applications. *The Visual Computer* Vol. 29, No. 5, 393–405, 2013.
- [11] Jia, Y. T.; Hu, S. M.; Martin, R. R. Video completion using tracking and fragment merging. *The Visual Computer* Vol. 21, Nos. 8–10, 601–610, 2005.
- [12] Wexler, Y.; Shechtman, E.; Irani, M. Space-time completion of video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 29, No. 3, 463–476, 2007.
- [13] Kopf, J.; Kienzle, W.; Drucker, S.; Kang, S. B. Quality prediction for image completion. *ACM Transactions on Graphics* Vol. 31, No. 6, Article No. 131, 2012.
- [14] He, K. M.; Sun, J. Image completion approaches using the statistics of similar patches. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 36, No. 12, 2423–2435, 2014.
- [15] He, K. M.; Chang, H. W.; Sun, J. Rectangling panoramic images via warping. *ACM Transactions on Graphics* Vol. 32, No. 4, Article No. 79, 2013.
- [16] Yeh, I. C.; Lin, S. S.; Hung, S. T.; Lee, T. Y. Disparity-preserving image rectangularization for stereoscopic panorama. *Multimedia Tools and Applications* Vol. 79, Nos. 35–36, 26123–26138, 2020.
- [17] Heeger, D. J.; Bergen, J. R. Pyramid-based texture analysis/synthesis. In: *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, 229–238, 1995.
- [18] Portilla, J.; Simoncelli, E. P. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision* Vol. 40, No. 1, 49–70, 2000.
- [19] Bertalmio, M.; Sapiro, G.; Caselles, V.; Ballester, C. Image inpainting. In: *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, 417–424, 2000.
- [20] Pérez, P.; Gangnet, M.; Blake, A. Poisson image editing. *ACM Transactions on Graphics* Vol. 22, No. 3, 313–318, 2003.
- [21] Bertalmio, M.; Bertozzi, A. L.; Sapiro, G. Navier-Stokes, fluid dynamics, and image and video inpainting. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.
- [22] Criminisi, A.; Perez, P.; Toyama, K. Object removal by exemplar-based inpainting. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.
- [23] Drori, I.; Cohen-Or, D.; Yeshurun, H. Fragment-based image completion. *ACM Transactions on Graphics* Vol. 22, No. 3, 303–312, 2003.
- [24] Komodakis, N.; Tziritis, G. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Transactions on Image Processing* Vol. 16, No. 11, 2649–2661, 2007.
- [25] Sun, J.; Yuan, L.; Jia, J. Y.; Shum, H. Y. Image completion with structure propagation. *ACM Transactions on Graphics* Vol. 24, No. 3, 861–868, 2005.
- [26] Pavić D.; Schönefeld, V.; Kobbelt, L. Interactive image completion with perspective correction. *The Visual Computer* Vol. 22, Nos. 9–11, 671–681, 2006.
- [27] Hays, J.; Efros, A. A. Scene completion using millions of photographs. *ACM Transactions on Graphics* Vol. 26, No. 3, Article No. 4, 2007.
- [28] Wang, M.; Lai, Y. K.; Liang, Y.; Martin, R. R.; Hu, S. M. BiggerPicture: Data-driven image extrapolation using graph matching. *ACM Transactions on Graphics* Vol. 33, No. 6, Article No. 173, 2014.
- [29] Zhu, Z.; Huang, H. Z.; Tan, Z. P.; Xu, K.; Hu, S. M. Faithful completion of images of scenic landmarks using Internet images. *IEEE Transactions on Visualization and Computer Graphics* Vol. 22, No. 8, 1945–1958, 2016.
- [30] Wang, M.; Shamir, A.; Yang, G. Y.; Lin, J. K.; Yang, G. W.; Lu, S. P.; Hu, S.-M. BiggerSelfie: Selfie video expansion with hand-held camera. *IEEE Transactions on Image Processing* Vol. 27, No. 12, 5854–5865, 2018.
- [31] Shiratori, T.; Matsushita, Y.; Tang, X. O.; Kang, S. B. Video completion by motion field transfer. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 411–418, 2006.
- [32] Igarashi, T.; Moscovich, T.; Hughes, J. F. As-rigid-as-possible shape manipulation. *ACM Transactions on Graphics* Vol. 24, No. 3, 1134–1141, 2005.
- [33] Schaefer, S.; McPhail, T.; Warren, J. Image deformation using moving least squares. *ACM Transactions on Graphics* Vol. 25, No. 3, 533–540, 2006.
- [34] Karni, Z.; Freedman, D.; Gotsman, C. Energy-based image deformation. *Computer Graphics Forum* Vol. 28, No. 5, 1257–1268, 2009.
- [35] Carroll, R.; Agarwala, A.; Agrawala, M. Image warps for artistic perspective manipulation. *ACM Transactions on Graphics* Vol. 29, No. 4, Article No. 127, 2010.
- [36] Werner, T.; Zisserman, A. New techniques for automated architectural reconstruction from photographs. In: *Computer Vision – ECCV 2002. Lecture Notes in Computer Science, Vol. 2351*. Heyden, A.; Sparr, G.; Nielsen, M.; Johansen, P. Eds. Springer Berlin Heidelberg, 541–555, 2002.
- [37] Brox, T.; Bruhn, A.; Papenberg, N.; Weickert, J. High accuracy optical flow estimation based on a theory for warping. In: *Computer Vision – ECCV 2004. Lecture Notes in Computer Science, Vol. 3024*. Pajdla, T.; Matas, J. Eds. Springer Berlin, 25–36, 2004.

- [38] Sun, D. Q.; Roth, S.; Black, M. J. Secrets of optical flow estimation and their principles. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2432–2439, 2010.
- [39] Li, Z. T.; Zhang, J.; Zhang, K. H.; Li, Z. Y. Visual tracking with weighted adaptive local sparse appearance model via spatio-temporal context learning. *IEEE Transactions on Image Processing* Vol. 27, No. 9, 4478–4489, 2018.
- [40] Xu, Y.; Chen, R. J.; Gotsman, C.; Liu, L. G. Embedding a triangular graph within a given boundary. *Computer Aided Geometric Design* Vol. 28, No. 6, 349–356, 2011.

Jin-Liang Wu received his Ph.D. degree in applied mathematics from Zhejiang University, Hangzhou, China, in 2012. He is a senior engineer in the 54th Research Institute of CETC, Shijiazhuang, China. His research interests include image and video processing, data analytics, artificial intelligence.

Jun-Jie Shi received his B.S. degree in computer science from the Beijing Institute of Technology, China, in 2017, where he is pursuing a master degree with the School of Computer Science. His research interest is image and video processing.

Lei Zhang received his B.S. and Ph.D. degrees in applied mathematics from Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively. He is a professor at the School of Computer Science, Beijing Institute of Technology. His research interests include image and video processing, and computer graphics.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.