

An Interactive Perspective Scene Completion Framework Guided by Complanate Mesh

Chuanyan Hao¹, Zilong Jin², Zhixin Yang^{3*} and Yadang Chen²

¹ School of Education Science and Technology, Nanjing University of Posts and Telecommunications
Nanjing, Jiangsu 210023 - China

[e-mail: hcy@njupt.edu.cn]

² School of Computer and Software, Nanjing University of Information Science and Technology
Nanjing, Jiangsu 210044 - China

[e-mail: zljn85@163.com, cyd4511632@gmail.com]

³ Faculty of Science and Technology, University of Macau
Taipa, Macao 999078 - China

[e-mail: zxyang@umac.mo]

*Corresponding author: Zhixin Yang

*Received October 14, 2019; revised December 12, 2019; accepted January 3, 2020;
published January 31, 2020*

Abstract

This paper presents an efficient interactive framework for perspective scene completion and editing tasks, which are available largely in the real world but rarely studied in the field of image completion. Considering that it is quite hard to extract perspective information from a single image, this work starts from a friendly and portable interactive platform to obtain the basic perspective data. Then, in order to make this interface less sensitive, easier and more flexible, a perspective-rectification based correction mechanism is proposed to iteratively update the locations of the initial points selected by users. At last, a complanate mesh is generated by the geometry calculations from these corrected initial positions. This mesh must approximate the perspective direction and the structure topology as much as possible so that the filling process can be conducted under the constraint of the perspective effects of the original image. Our experiments show the results with good qualities and performances, and also demonstrate the validity of our approaches by various perspective scenes and images.

Keywords: image completion, image editing, exemplar-based synthesis, interaction, perspective rectification

This work is supported by NSFC 61702278, 61802197 and 61602252, and Natural Science Foundation of Jiangsu Province of China (No.BK20160902, BK20160964 and BK20160967) and the Startup Foundation for Introducing Talent of Nanjing University of Posts and Communications(NJUPT)(NO. NY217015).

1. Introduction

Image completion has been widely used in many applications, varying from computer graphics and vision algorithms, like background recovery, image reshuffling, to industrial productions, like photo editing in Photoshop, animation and video post-processing. It is a well known technique to reconstruct incomplete images or movie frames by filling target regions in a visual plausibly way. Although so much advances have been made, perspective scene completion still remains a hard problem. As a matter of fact, perspective effects are broad available and can be readily observed in our real and man-made artistic worlds. Therefore, the research work on image filling technology for perspective phenomena is of great significance for realistic simulation and solutions of image editing jobs in this article have their own special challenges that distinguish from many previous image completion problems.

Perspective is a drawing method that expresses 3D scene as realistically as possible on two-dimensional image from the view of observers. Artists have used perspective drawings to depict three-dimensional visual effects on two-dimensional images for a long time, as dating to the Renaissance. It can be seen that perspective scene is an indispensable element to simulate the real world, and the research on it has high value in theory and application. In fact, many research fields, such as image-based modeling and animation, must follow the principle of perspective imaging. Recently, one of the typical tasks is the 3D reconstruction technology based on camera calibration. This technique has achieved good results, but generally requires the input of multiple images to reconstruct the real three-dimensional information [1-2]. This is a complex, tedious and resource-consuming process, not suitable for the perspective scenario just in a single image. Thus another type of work is a better choice, which preserves perspective effects through interactive system where basic perspective information is provided by simple and easy user operations.

The most popular image completion algorithms are the fragment-based synthesis methods [3-5] with a range of improvements. The core of this kind of approaches is to measure the similarity of block overlapping regions on the defined search space, and then put the most matched block in the appropriate target position. Here, the similarity measurement is used to control the selection of the most matching candidate blocks in the source image region of the known content, but the metric is only based on the color of the block pixels or simple structure information. In addition, various skills for defining search space have only improved performance to a certain extent or only retained some specific image features. Moreover, most filling algorithms only use simple translation, rotation and scaling operations when placing candidate blocks. Now the challenges in this study lies on that the above progresses do not give good solutions for perspective features. On the other hand, the general completion algorithms always assume that the source block is small enough to be coplanate in the scene, but such assumptions often ignore the real three-dimensional information, occlusion effects and other factors, which make them impossible to maintain the perspective effect consistent with the original image. In hence, with the help of user interaction, this paper freshly establishes perspective mesh to guide the transmission of perspective information and structural features so as to correct the possible mistakes caused by the non-coplanar issue.

The primary contributions demonstrated in this paper contain a portable and friendly interactive platform, a perspective-based correction method to adjust user input and an automation approach to reconstruct the background images. **Fig. 1** gives an example of our work. **Fig. 1 (a)** is the input image. The magenta region in **Fig. 1 (b)** is the area to be filled. **Fig.**

1 (c) shows the recovery result obtained by our method while **Fig. 1 (d)** is the result from the typical fragment-based completion algorithm [3] without perspective constraint. Obviously, our method successfully preserves the perspective structures. In Section 2, we review the related work in image completion tasks. Section 3 elaborates our approach and the interactive system. We then discuss the experiments and results in Section 4, followed by drawing the conclusion and future work in Section 5.

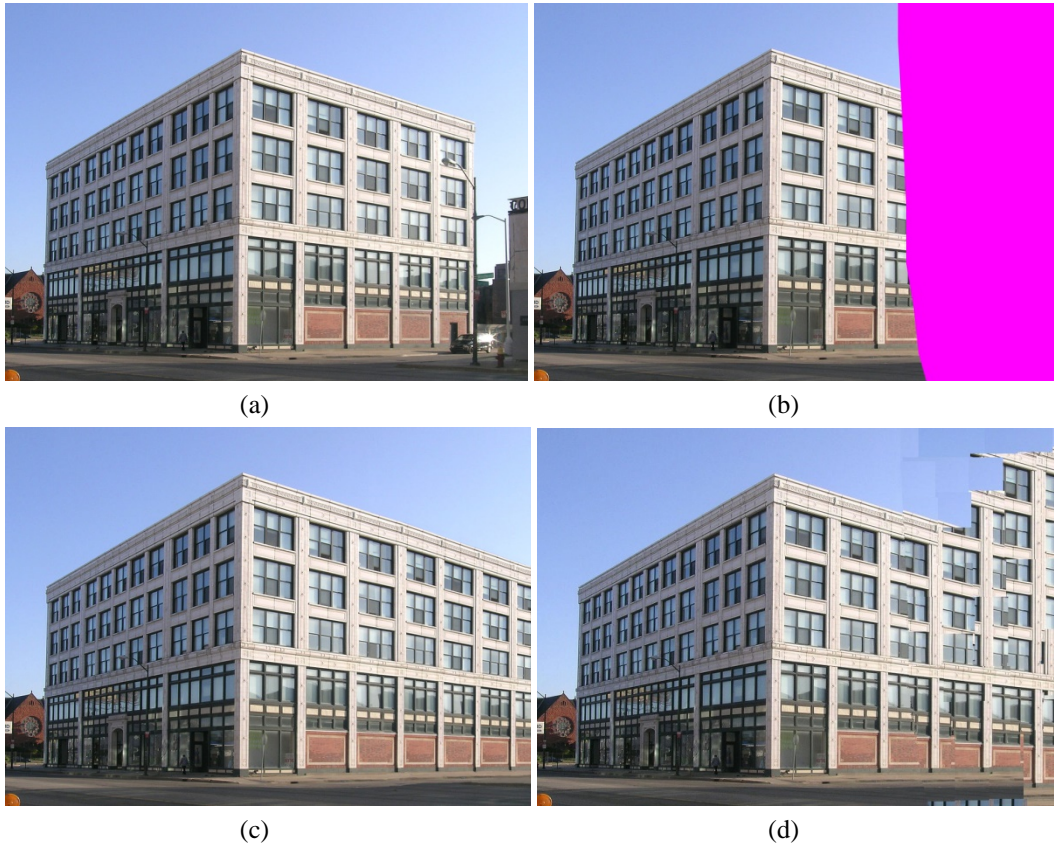


Fig. 1. An example of perspective constraint image completion. (a) is the source image. (b) is the masked image. (c) is the result from our method and (d) is from the work in [3].

2. Related Work

Image completion originated from image inpainting technology [6-7], but the latter is more suitable for narrow linear structures such as lines, small gaps, stains and so on, such as repairing scratches from old films. When image inpainting is used to repair large missing areas, it is easy to lead to blurred artificial traces, so image completion is proposed for filling a large part of missing areas. On the other hand, the development of image completion techniques is closely related to the promotion of texture synthesis technologies [8-14].

Texture synthesis is a process of generating large texture images from small texture samples, in which texture samples and generated texture images need to keep visual consistency. At present, the common texture synthesis algorithms are based on a non-parametric sampling process, which can be roughly divided into two categories: pixel-based algorithms [8-10] and patch-based algorithms [11-14]. The milestone work is the Markov Random Field (MRF) model proposed by Efros and Leung et al. [8] who introduced the non-parametric approach,

turning the parametric model to a neighborhood search problem. Then some improvements are proposed by modifying search and sampling strategies. Wei et al. accelerated the search process through multiple frequency bands and vector quantization [10]. Work in [9] shows a stronger consistent search method, greatly reducing the search space and achieving interactive speed. However, patch-based method is the main trend of modern texture synthesis techniques because of its advantages in speed and quality. The most important one is also from the work [11] by Efros et al., which creates the new texture by directly copying the patches from the source texture. Kwatra et al. [12] introduced the graph cutting algorithm into texture synthesis to optimize the stitching of block boundaries. Besides textures, more researches concentrate on images [12], videos [14], gradient domain operations [15-16] and so on. Barnes et al. [17] proposed the PatchMatch algorithm, successfully speeding up this kind of work through a fast random search scheme, and further extended to a variety of vision applications, named the Generalized PatchMatch [18]. Most recently, based on the PatchMatch method, manifold skills on image editing is collected in [19] and an excellent work back to texture synthesis is shown in [20].

Similarly, exemplar-based image completion algorithms develop through two stages, from pixel-based [6-7] to fragment-based [3-5]. Also, the fragment-based methods obtain better quality and performance. They usually augment earlier image inpainting algorithms by applying texture and structure inpainting simultaneously to fill the missing regions such as the work in [3-4]. In order to preserve strong oriented structures, Criminisi et al. [3] made an improvement on the order of the example-based texture synthesis algorithms, which is determined by the strength of the boundary gradient of the missing regions. A related approach is proposed in [5] which uses interactive framework to indicate where structures should be propagated. Additional similar techniques include those from [21-22]. These typical image completion methods generally fill the missing regions by borrowing small pieces from the original image. Alternatively, Hays et al. [23] presented a novel algorithm which takes a large number of images such as a large enough image database sharing from the website or the Internet as the source and selects the candidate source image by comparing image contexts and boundary compatibility. Recently, work in [24-25] demonstrate some outstanding results in image completion jobs.

This work is also somewhat related to image-based modeling and editing techniques, which convey the perspective information by drawing quads. Instructively, Pavic et al. [26] presented an quad metaphor method to generalize the feature curves to arbitrary feature points. Liu et al. [27] discover the texture regularity of near regular textures in the form of quad-like lattice. Most recently, transformation based skills [28-29] are also introduced to image completion tasks. However, these methods are not perspective specifically and always offer a complex quads commutation process while our approach supplies a simpler workflow to achieve the three dimensional visual effect.

3. Our Approach

3.1 Problem and Solution Design

As motivated by the problem shown in Fig. 1, our work focus on how to preserve the perspective structures. It is easy to think about making use of some special type of medium to guide image completion to do synthesis along perspective directions. Perhaps a perspective mesh is a good choice, which is supposed to not only reveal the direction of perspective but also restrict the shape and size of patches according to the perspective effect, as shown in Fig.

2, the green mesh. And the last image Fig. 2 is used to illustrate the synthesis process by our perspective constrained image completion approach stated in Section. Furthermore, our experiments prove that a perspective mesh is indeed a good solution to this problem.

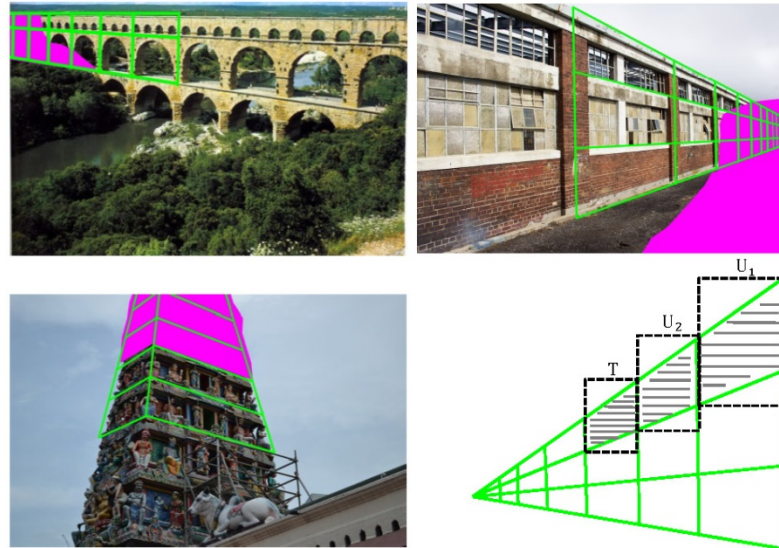


Fig. 2. Examples of the complete perspective meshes.

The next question is how to construct the perspective mesh. The first important thing is to obtain the basic perspective information such as the vanishing point. Although there is a lot of work aimed at exploring the perspective phenomenon of a single image, it is still difficult to find a completely automatic method to obtain accurate perspective information from a single image. For example, for a single image with any feature, it is almost impossible to find a completely automatic method to accurately determine the position of the vanishing point. In other words, many automatic methods are proposed for images with a certain class of features. In contrast, human vision can easily identify rough perspective information and help machine systems through interactive interfaces. In fact, the interactive system not only can produce a more reasonable visual effect than the complex automation method, but also can more better feedback the ideas of the users. In our case, after the deep investigation of both two strategies, an interactive system and an automatic approach, it is turn out that combination of user interaction could lead to much more plausible results.

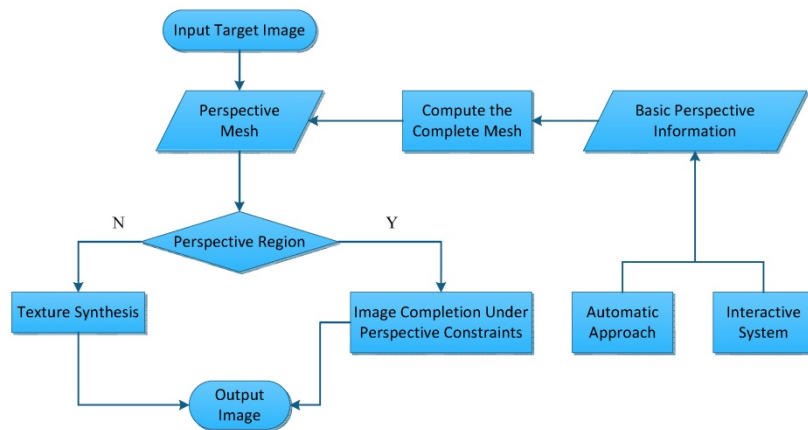


Fig. 3. The workflow of our system.

So far, we can draw the blueprint of our perspective constrained system as shown in Fig. 3, a system diagram. After having the complete perspective mesh in hand, image completion can be performed under the constraint of the perspective mesh. The details about mesh calculation and the mesh-based synthesis procedure will be elaborated in Section 3.4 and Section 3.5.

3.2 User Interface

Our interactive system offers a friendly interface to users to specify the basic perspective information. Instead of drawing perspective lines or locating the vanishing point, users only need to pick up several points in an intuitive way, that is, clicking on those points along the perspective direction, as shown in Fig. 4. Such simplicity sacrifices the accuracy of the input points and likely leads to distorted perspective meshes. Therefore, we must briefly discuss the principle of perspective projection first before solving this problem.



Fig. 4. User interface. The left image shows the points (red and green) picked up by users with their intuitive perspective direction. The right image illustrates the possible errors caused by the rough operations, like the three vanishing points (black crosses) generated by any two sets of points.

3.2.1 Perspective Projection

Generally speaking, perspective is divided into parallel projection and perspective projection. The latter is our focus, which converges the projection point along the horizon line to the vanishing point, and produces the visual effect that the object narrows gradually along the line of sight, thus reflecting the three-dimensional sense on the two-dimensional image plane, as shown in the left image in Fig. 5.

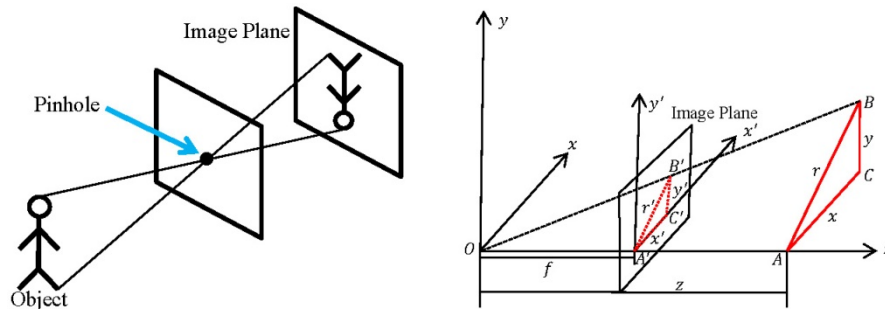


Fig. 5. Perspective projection. The left image shows the principle of pinhole camera imaging. The right image is an explanatory diagram for perspective projection transformation.

Based on the right image in Fig. 5, we first let the center of projection be on the world origin, the camera axis on the z -axis of the world system and the image plane be in front of the projection center, then from the similar triangle theory, the new position (x', y', z') for any three dimensional point (x, y, z) after the perspective projection could be computed as follows:

1. From $\triangle OA'B' \cong \triangle OAB$: $\frac{f}{z} = \frac{r'}{r}$

2. From $\Delta A'B'C' \cong \Delta ABC$: $\frac{x'}{x} = \frac{y'}{y} = \frac{r'}{r}$
3. For 3D point (x, y, z) , its 2D correspondence after projection is:

$$x' = \frac{x \times f}{z}, y' = \frac{y \times f}{z}, z = f$$
4. The matrix notation is as following in the form of homogeneous coordinates:

$$\begin{bmatrix} x' \\ y' \\ z' \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & f & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (1)$$

Formula (1) ensures that scaling and projection reduction are proportional to image size. This shows that the reduction of the quadrangle block along the perspective direction obeys the invariance of the intersection ratio, that is to say, the change ratio between the distances of the line segment is consistent. Besides, we also notice that although implicit three dimensional information is exploited for this system, a projective transformation of a plane is essentially a two dimensional transform, that is a homograph transform which can be defined by a 3×3 matrix using homogeneous coordinates. In hence, it is natural to think that the easiest way to represent a homograph transformation is to make use of quadrilaterals in image space because the correspondences among the four two-dimensional points provide the eight constraints by which the nine entries of the 3×3 matrix can be figured out.

3.2.2 User Interactions

So in our system, users just need to select the four points of a quadrilateral (see the small yellow boxes in Fig. 6) through which the quadrilateral is connected as the green patches shown in Fig. 6. After rectifying these points, we will figure out the cross ratio according to which the complete perspective mesh (see Fig. 6 (e) as an example) is constructed by spreading out these initial quads.

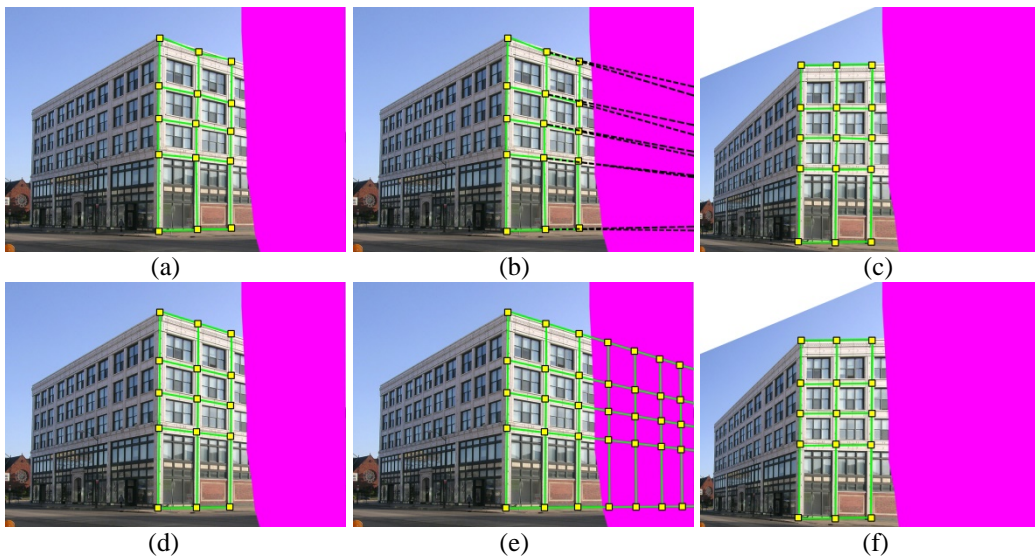


Fig. 6. User interactions. (a) shows the rough initial points by user input. (b) illustrates that the inaccurate inputs cause multiple perspective directions. (c) is the non-coplanar and discontinuous orthogonal view. (d) demonstrates the initial points after correction. (e) gives an example of the complete mesh after spreading out along the consistent perspective direction. (f) is the coplanar and continuous orthogonal view.

Specifically speaking, after loading the input images, users can first optionally paint a target region where the unwanted contents will be removed and reconstructed later by our image manipulations, as the magenta area shown in **Fig. 6 (a)**. Then users can feed several initial points which represent the rough perspective and structure information to our system by picking up several pixels in the input image along the perspective directions. For example, these small yellow boxes in **Fig. 6 (a)** indicate the initial choices from the user input. Since this operation does not require any specific knowledge on the theory of perspective projection and users only need to finish this step in terms of their intuitions and daily experiences, these initial points cannot guarantee the accurate and consistent perspective directions, such as those dashed black lines (**Fig. 6 (b)**) which stretch to different directions rather than gathering in one point (the vanishing point). In addition, coarse initial points also connect a mesh which cannot make sure the coplanarity of adjacent rectified regions and the continuity at the boundaries of them. For instance, it can be seen from **Fig. 6 (c)** that discontinuous mesh generates a distorted and interlaced rectified regions. It can be imagined that these inaccurate initial points must incur the failure of image completion. Therefore, in Section 3.3, we use a perspective transformation between the original image and the rectified image to iteratively update the initial points.

3.3 Correction for User Input

As mentioned above, our interactive system is intuitive and user-friendly. Even though it captures implicit three-dimensional information, the user interactions are strictly two-dimensional operations. However, such simplicity offered to users is achieved at the price of losing the accuracy of the input points. As a result, users may pick up points multi-times if the previous selections of points are too bad. In order to address this issue, we propose a perspective-based correction method to help our interactive system to obtain initial points as accurate as possible by adjusting the coarse user inputs and keeping the friendship with users at the same time.

3.3.1 Correction of Initial Points

The basic idea behind this correction process is pretty simple and easy to understand, which states that if the initial points are precise enough (like **Fig. 6 (d)** presented), then after rectification, all the rectified quads together should yield an unfolded and continuous orthogonal plane of the original image, as **Fig. 6 (f)** shown. We already have known that while our interface allows untrained users to quickly select a rough approximation of the four points of a consistent quad piece, it turns out to be quite difficult for users to pick up points sufficiently precise that the rectified image is a coplanar and continuous deformation of the original input image. The reason why so difficult lies in that even if two quads are defined on the same vertices, it does not mean that the two neighboring edges evolving the common vertices can coincide on a same line. As illustrated in **Fig. 7 (a)**, quads $ABED$ and $BCFE$ share edge \overline{BE} , but the common vertex B cannot guarantee that edges \overline{AB} and \overline{BC} are on the same line and alike, the common vertex E cannot ensure edges \overline{DE} and \overline{EF} on the same line. Similar cases may also happen in other edges which share the same vertices. In order to make user interactions less sensitive, we consider the deviations of the point positions before and after the rectification and derive a perspective-rectification-based iteration process to adjust the locations of the initial points by users to a nearby configuration such that edges evolving the common vertices can be on the same line or the rectified quad can be a rectangle.

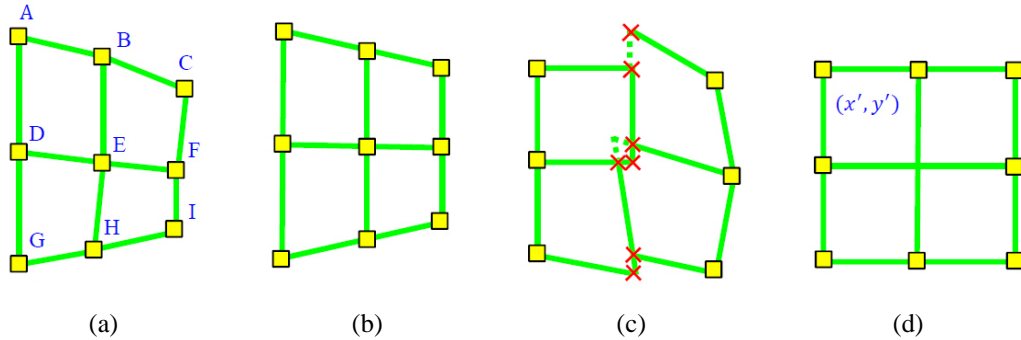


Fig. 7. Correct the initial points by user input. (a) shows a rough approximation of the initial constellation of a quad. (b) is the ideal locations of the initial points. (c) visualizes the discontinuities after the homograph transform. (d) demonstrates the ideal positions after correction.

For simplicity, let us first assume that the rectified quads are squares, if the initial points can ensure the continuity of quads, then it will generate a mesh consisting of coplanar squares after performing a piecewise homograph transform, as **Fig. 7 (d)** shown. Unfortunately, the locations of these initial points are just a rough approximation. In terms of projective geometry, the true case is that the vertices evolved in the adjacent edges will deviate from itself after the homograph transforms, for instance, the red crosses in **Fig. 7 (c)** indicate the offsets of the common vertices. Just as observed this deviation, our iteration process updates the locations of the initial points by measuring the inconsistency of how strongly the two mappings deviate at the vertices of the common edge and minimizing the sum of vertices deviations.

Mathematically, let quad $ABED$ be $Q(x, y)$ where (x, y) is the coordinates of vertex A , then the neighboring quads are $Q(x + 1, y)$, $Q(x, y + 1)$ and $Q(x + 1, y + 1)$ and the homograph transform defined on $Q(\cdot)$ can be noted as $F(Q(\cdot))$. Because the deviation appears in the image space, it is better to use the inverse homograph transform $F^{-1}(Q(\cdot))$ which can map the ideal unit squares (see **Fig. 7 (d)**) $(x', x' + 1) \times (y', y' + 1)$ back to the image quads $(x, x + 1) \times (y, y + 1)$, that is $\{Q(\cdot)\}$. Then the deviation on the common vertex $(x + 1, y)$, that is vertex B , can be computed by L_1 norm as:

$$\|F^{-1}(Q(x, y)) - F^{-1}(Q(x + 1, y))\|_{(x+1,y)} \quad (2)$$

The new homograph F' will be yielded because of the updated locations of common vertices and thus the above procedure repeats under the new F' until the best matched locations are determined through minimizing the sum of all the deviations. This iteration traverses each common vertices to adjust them to their proper positions. In addition, the boundary vertices are kept fixed as much as possible to enforce the initialization and end conditions. At last, this algorithm always works robustly and converge fast within a few iterations in practice.

3.4 Figuring of Perspective Mesh

Based on the precise initial mesh (see **Fig. 7 (b)**), the complete perspective mesh can be calculated according to the theory of projective geometry. As illustrated in **Fig. 8**, what we want to solve are the cyan mesh vertices and the purple vanishing point V . Since the initial vertices have been corrected in the last section, edges \overline{AB} and \overline{BC} are on the same line and converge to the vanishing point V . The same principle is applied to edges \overline{DE} and \overline{EF} , \overline{GH} and \overline{HI} . Then vanishing point V is the intersection of above any pair of edges and the angle β between \overline{VA} and \overline{AG} or \overline{AD} is again determined from the law of cosines as following shown:

$$\sin \beta = 1 - \left(\frac{\|\overline{VA}\|^2 + \|\overline{AG}\|^2 - \|\overline{VG}\|^2}{2 \times \|\overline{VA}\| \times \|\overline{AG}\|} \right)^2 \quad (3)$$

where $\|\cdot\|$ denotes the length of line segments \overline{VA} and \overline{AG} or \overline{VG} that are measured by the Euclidean distance.

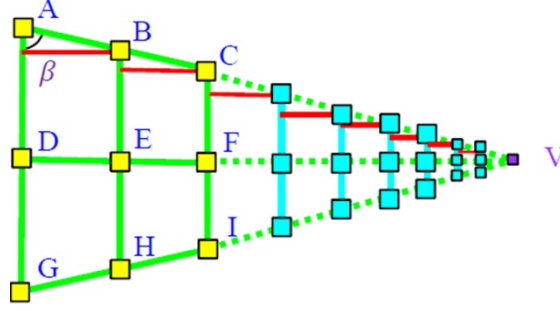


Fig. 8. Illustration for mesh computation.

Similarly, vertices sets (A, D, G) , (B, E, H) and (C, F, I) have the property of being collinear. And the aspect ratio q among quads can be defined as $q = \frac{\|C \perp \overline{BH}\|}{\|B \perp \overline{AG}\|}$, which means that q is the ratio of the vertical distance from vertex B to \overline{AG} (the first red line in **Fig. 8**) and from vertex C to \overline{BH} (the second red line in **Fig. 8**). Now, let the coordinates of any cyan vertices in line \overline{VA} be $Z_i(x, y)$, then it can be computed as:

$$Z_i \begin{bmatrix} x \\ y \end{bmatrix}_{i=0}^k = \begin{bmatrix} C_x + \|B \perp \overline{AG}\| \times q^{i+2} \\ C_y + (1 - (\sin \beta)^2) \times \|\overline{AB}\| \times q^{i+2} \end{bmatrix} \quad (4)$$

where i is the index of the cyan vertices, k is the total number of the cyan vertices in line \overline{VA} , $i, k \in N$, and (C_x, C_y) are the coordinates of vertex C . Making using of the same computation, cyan vertices in line \overline{VG} can be determined and thus cyan structure lines and cyan vertices in \overline{VD} can be figured out. All of these elements finally construct the complete perspective mesh and optionally, we also keep the interface to feedback the complete mesh to users for possible adjustment.

3.5 Image Completion under Perspective Mesh

With the constraint of perspective grid, the completion process can reasonably transmit perspective information and structural characteristics of targets. It is first to be mentioned that the perspective grid divides the original filled target area (the user-given magenta area) into two parts: the area covered by the perspective grid and the background area that is not covered. For the background area with no perspective and structural features, traditional patch-based image synthesis method is adopted as seen in the system diagram in **Fig. 3**.

Secondly, for the perspective feature area, the image filling work is mainly composed of two stages under the guidance of perspective grid. The first stage is search and comparison. This stage differs from the conventional method in that the search is performed in the perspective direction, constrained by the perspective grid. The metric to compare patches is still the sum squared distance (SSD), but the effective comparison content is no longer a regular square or rectangular block, but a perspective grid constrained quadrangle block area, indicated by the shading part in the last image of **Fig. 2**, also as the quad $ABED$ and $BCFE$ in **Fig. 8** shown. In addition, considering the structural characteristics, the gradient factor is introduced into the calculation of search and comparison. Now, let the quads in **Fig. 8** as Q , the set of the source patches be $S = \{Q_j | j = 1, 2, \dots, n\}$, n for the number of the source patches,

and the current patch be T , $G(\cdot)$ denote the distance of the image gradient, $C(\cdot)$ represent the distance of the image color, and w_g and w_c be the weights of the gradient and the color respectively, M_α be the mask of the target area, then the best matched patch $S' \in \{S\}$ for T could be defined as:

$$S' = \operatorname{argmin}_{j=1}^n \{M_\alpha \times [w_g \times G(T, Q_j) + w_c \times C(T, Q_j)]\} \quad (5)$$

The second stage is to put the best matched patch S' to the correct position. In order to comply with the constraint of the perspective grid, the transformation of the block position is subject to the cross ratio q , that is, $P(T) = \varphi(q) \cdot P(Q')$. Here, P defines the transfer of block positions, which is constrained by $\varphi(q)$ and calculated by bilinear interpolation.

Table 1. Performance illustration

Image Resolution(MP)	Completion	Number of Initial Points	Perspective Mesh
(0.1-0.25)	73s	≤ 6	1.05s
(0.25-0.5)	110s	6-10	3.53s
(0.5-0.75)	154s	10-20	7.33s
(0.75-1.0)	240s	> 20	13.27s

4. Experimental Results and Discussion

Our tests for this correction system on a number of images are conducted on a platform with Intel Core i7 CUP 2600K @3.40GHz and 8GB memory, implemented using Matlab R2016a. The interaction consumes little time of several seconds to several minutes, varying according to different structures. In the stage of filling, mesh correction and generation just occupy a small part of the total time consuming since its generation is through the geometrical calculation and has nothing to do with image resolutions. Instead, synthesis for recovering the background areas which do not contain perspective information costs more. The details of the performance are shown in **Table 1** where the unit of image is megapixel (MP) and the unit of time consuming is seconds.

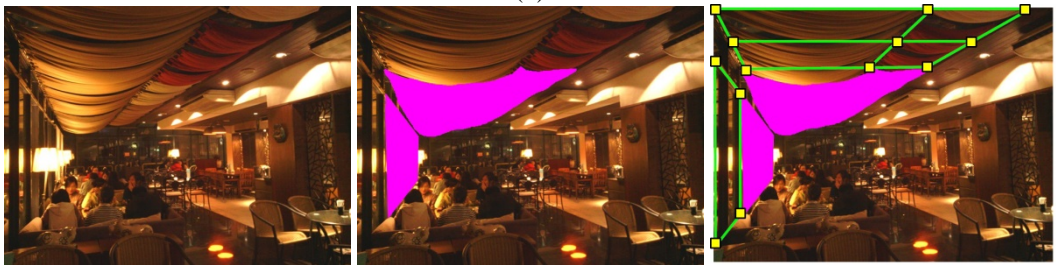
Nine groups of our experiments from (a) to (i) and their comparisons with several typical exemplar-based image completion methods are presented in **Fig. 9** to **Fig. 11**, in which images in each group are respectively the source image, target region to be replaced (marked by magenta), initial mesh after correction, the complete perspective mesh, result from our algorithm and the result by other three image completion approaches. In **Fig. 9**, we compared our method with the classical fragment-based algorithm [3] while comparing with the other two approaches, image melding [19] and planar-guided complete [24] in **Fig. 10** and **Fig. 11** separately. It is obvious that with the assistance of our perspective mesh, the recovered images can preserve the original perspective effects without artifacts which happen in the results produced by the compared methods. On the other hand, our mesh correction mechanism not only makes user interactions be easier, less sensitive and more flexible, but also guarantees the accuracy of the final perspective mesh and can handle images with some complex structures, such as the cases in group (b) in **Fig. 9** and group (i) in **Fig. 11**. Besides, results in group (c) and (d) in **Fig. 9** show that our algorithm has the ability to work well in scenes that have more than one vanishing points. At last, the rest four groups of (e) to (h) in **Fig. 10** and **Fig. 11** also reveal that our system is flexible and versatile for cases with various directions and shapes.



(a)



(b)



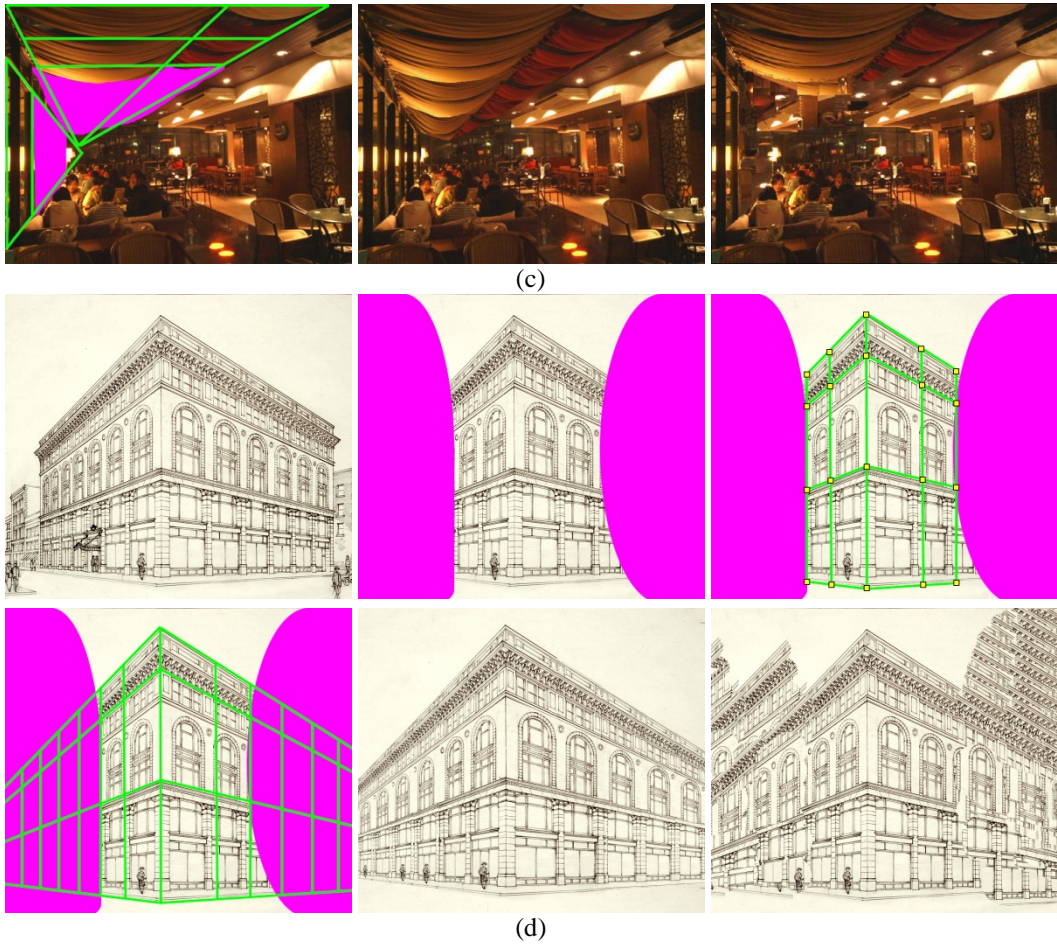
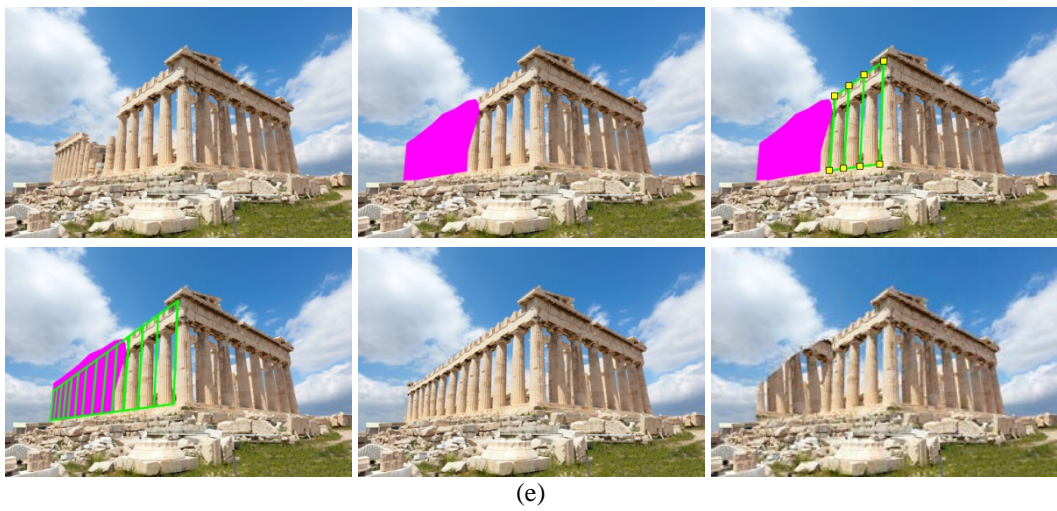


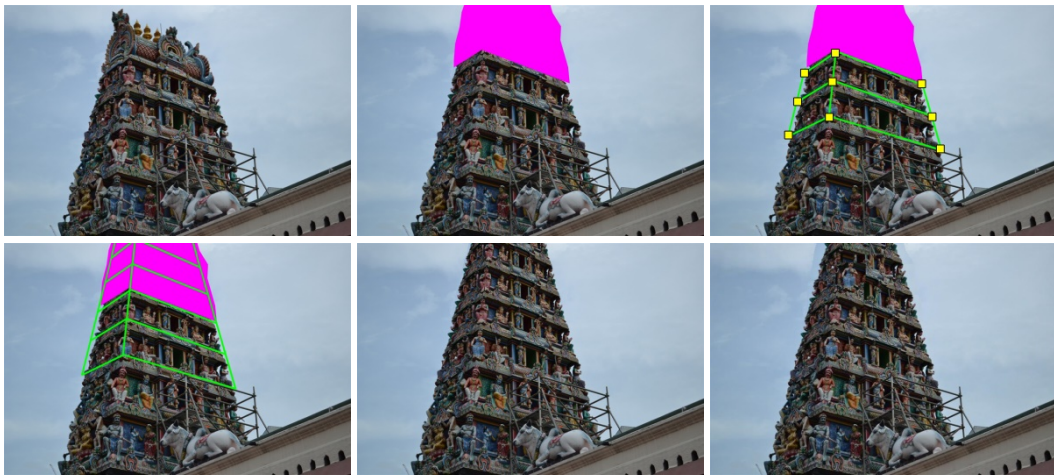
Fig. 9. Groups (a) to (d) of the nine groups of results by our corrected system, which are compared with method in [3].



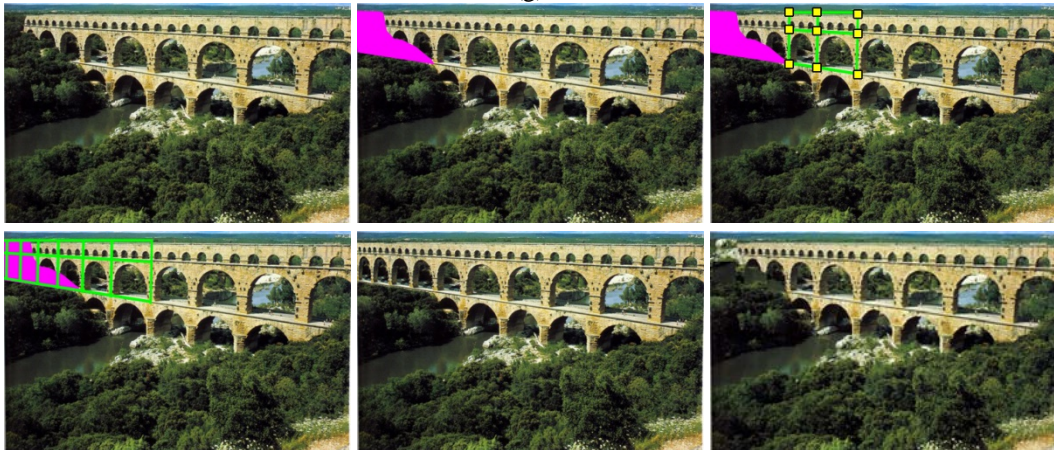


(f)

Fig. 10. Groups (e) and (f) of the nine groups of results by our corrected system, which are compared with method in [19].



(g)



(h)



Fig. 11. Groups (g) to (i) of the nine groups of results by our corrected system, which are compared with method in [24].

5. Conclusion

This paper presents an effective and friendly interactive framework for image completion of perspective scenes. The interactive operation of the system is simple and intuitive, does not need the professional knowledge of the users. Guided by the perspective mesh, it successfully conveys the perspective structure and generates the results with visually plausible quality. However, such simplicity cannot guarantee the accuracy of the points picked up by users and sometimes causes the redundant operations of users, that is the multiple selection of points, therefore, we augment this interactive system by a correction mechanism which uses a perspective-rectification-based iteration process to update the locations of the initial points so that they can match the proper positions as much as possible. This corrected system is proved to work well and can make our interface be less sensitive, more flexible and simpler to users.

Nevertheless, there is much further work demanded to improve our system. We would like to link it to broader range of structures and make a try to automatic one. But as well known, it is hard to propose an algorithm or mathematic formula which could extract perspective information from a single image with arbitrary features. Therefore, machine learning based methodology is a suggestive future direction with more features such as illumination, shading, reflections, clutters, texture, occlusion and so on being considered. In addition, it will be a good direction to utilize our algorithm on video editing and completion, animation and segmentation and so on.

References

- [1] G. F. Zhang, J. Y. Jia, T. T. Wong and H. J. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 31, pp. 974-988, 2009. [Article \(CrossRef Link\)](#).
- [2] Y. D. Chen, C. Y. Hao, Z. M. Cai, W. Wu and E. H. Wu, "Live accurate and dense reconstruction from a handheld camera," *Journal of Computer Animation and Virtual Worlds*, vol. 24, no. 3, pp. 387-397, 2013. [Article \(CrossRef Link\)](#).

- [3] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004. [Article \(CrossRef Link\)](#).
- [4] I. Drori, D. Cohen-Or, and H. Yeshurun, “Fragment-based image completion,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 303–312, 2003. [Article \(CrossRef Link\)](#).
- [5] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum, “Image completion with structure propagation,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 861–868, 2005. [Article \(CrossRef Link\)](#).
- [6] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proc. of the 27th Annual Conf. on Computer Graphics and Interactive Techniques*, pp. 417–424, July 23–28, 2000. [Article \(CrossRef Link\)](#).
- [7] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882–889, 2003. [Article \(CrossRef Link\)](#).
- [8] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proc. of the International Conf. on Computer Vision*, pp. 1033–1038, September 20–25, 1999. [Article \(CrossRef Link\)](#).
- [9] M. Ashikhmin, “Synthesizing natural textures,” in *Proc. of the 2001 Symposium on Interactive 3D Graphics*, pp. 217–226, March 26–29, 2001. [Article \(CrossRef Link\)](#).
- [10] L.-Y. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” in *Proc. of the 27th annual Conf. on Computer Graphics and Interactive Techniques*, pp. 479–488, July 23–28, 2000. [Article \(CrossRef Link\)](#).
- [11] A. A. Efros and W. T. Freeman, “Image quilting for texture synthesis and transfer,” in *Proc. of the 28th Annual Conf. on Computer Graphics and Interactive Techniques*, pp. 341–346, August 12–17, 2001. [Article \(CrossRef Link\)](#).
- [12] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: image and video synthesis using graph cuts,” *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 277–286, 2003. [Article \(CrossRef Link\)](#).
- [13] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum, “Real-time texture synthesis by patch-based sampling,” *ACM Transactions on Graphics*, vol. 20, no. 3, pp. 127–150, 2001. [Article \(CrossRef Link\)](#).
- [14] A. Schödl, R. Szeliski, D. H. Salesin, and I. Essa, “Video textures,” in *Proc. of the 27th Annual Conf. on Computer Graphics and Interactive Techniques*, pp. 489–498, July 23–28, 2000. [Article \(CrossRef Link\)](#).
- [15] P. Pérez, M. Gangnet, and A. Blake, “Poisson image editing,” in *Proc. of the 30th Annual Conf. on Computer Graphics and Interactive Techniques*, pp. 313–318, July 27–31, 2003. [Article \(CrossRef Link\)](#).
- [16] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, “Interactive digital photomontage,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 294–302, 2004. [Article \(CrossRef Link\)](#).
- [17] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patchmatch: a randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1–11, 2009. [Article \(CrossRef Link\)](#).
- [18] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, “The generalized patchmatch correspondence algorithm,” in *Proc. of the 11th European Conf. on Computer Vision*, pp. 29–43, September 5–11, 2010. [Article \(CrossRef Link\)](#).
- [19] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen, “Image melding: combining inconsistent images using patch-based synthesis,” *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–10, 2012. [Article \(CrossRef Link\)](#).
- [20] Z. Zhou, Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang, “Non-stationary texture synthesis by adversarial expansion,” *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–13, 2018. [Article \(CrossRef Link\)](#).

- [21] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007. [Article \(CrossRef Link\)](#).
- [22] M. Wilczkowiak, G. J. Brostow, B. Tordoff, and R. Cipolla, "Hole filling through photomontage," in *16th British Machine Vision Conf.*, pp. 492–501, September, 2005. [Article \(CrossRef Link\)](#).
- [23] J. Hays and A. A. Efros, "Scene completion using millions of photographs," *ACM Transactions on Graphics*, vol. 26, no. 3, pp. 1-7, 2007. [Article \(CrossRef Link\)](#).
- [24] J. -B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, "Image Completion Using Planar Structure Guidance," *ACM Transactions on Graphics*, vol. 33, no. 4, pp. 1-10, 2016. [Article \(CrossRef Link\)](#).
- [25] S. Lizuka, E. Simo-serra, and H. Ishikawa, "Globally and Locally Consistent Image Completion," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1-14, 2017. [Article \(CrossRef Link\)](#).
- [26] D. Pavić, V. Schönefeld, and L. Kobbelt, "Interactive image completion with perspective correction," *The Visual Computer*, vol. 22, no. 9, pp. 671–681, 2006. [Article \(CrossRef Link\)](#).
- [27] Y. Liu, W.-C. Lin, and J. Hays, "Near-regular texture analysis and manipulation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 368–376, 2004. [Article \(CrossRef Link\)](#).
- [28] J. -B. Huang, J. Kopf, N. Ahuja, and S. B. Kang, "Transformation Guided Image Completion," in *Proc. of IEEE International Conf. on Computational Photography*, pp. 1-9, April 19-21, 2013. [Article \(CrossRef Link\)](#).
- [29] K. He and J. Sun, "Statistics of patch offsets for image completion," in *Proc. of the 12th European Conference on Computer Vision*, pp. 16-29, October 7-13, 2012. [Article \(CrossRef Link\)](#)



Chuanyan Hao received the Ph.D. in Soft Engineering from University of Macau in 2015. She is currently an assistant professor in Nanjing University of Posts and Telecommunications. Her main research interests include texture synthesis and analysis, image and video processing and editing, image-based modeling and animation, data-driven approaches and so on.



Zilong Jin received the B.E. degree in computer engineering from Harbin University of Science and Technology, China, in 2009, and the M.S. and Ph.D. degrees in computer engineering from Kyung Hee University, Korea, in 2011 and 2016, respectively. He is currently an assistant professor of School of Computer and Software at Nanjing University of Information Science and Technology, China. His research interests include wireless sensor networks, mobile wireless networks, and cognitive radio networks.



Zhixin Yang obtained his PhD in Industrial Engineering and Engineering Management from the Hong Kong University of Science and Technology. He is currently an Associate Professor in the State Key Laboratory of Internet of Things for Smart City, Faculty of Science and Technology, and the Director of Research Service and Knowledge Transfer Office both at the University of Macau. His current research interests include fault diagnosis and prognosis, machine learning, and computer vision-based robotics.



Yadang Chen received the Ph.D. in Soft Engineering from University of Macau in 2015. He is currently an assistant professor in Nanjing University of Information Science and Technology, and also a Postdoctor in Michigan State University. His main research interests include video segmentation, video enhancement, video editing, augmented reality and so on.