

Image Completion Based on Partial Region

Mang Xiao¹, Guangyao Li¹, Yunlan Tan^{1,2}, Lei Peng¹, Yangjian Lv¹ and Li Xie¹

¹*College of Electronic Information and Engineering, Tongji University,
Shanghai 201804, China*

²*School of Electronic Information and Engineering, Jingtangshan
University, Ji'an, Jiangxi 343009, China
mangxiao@163.com*

Abstract

The traditional image completion algorithms based on patches, are to iteratively search the most similar patches in the known region of the whole image to fill the target region. Generally, it is difficult to use these algorithms in practice, due to the big computation. To overcome this problem, in this paper we propose a novel image completion algorithm based on partial regions. Firstly we adapt a randomized correspondence algorithm to search the pixels in the known region that are similar with the pixels in the border of the target region. Then the regions that include the main structures and Texture can be determined by these pixels, and thus we can limit the search space. Secondly we optimize the method of computing filling priorities that is based confidence factor and edge information. Consequently, the correctness of the order of the repairing patches and the success of structures propagation have been enhanced. Finally we improve the method of calculating the most similar patches. A variety of experiments show that, in general, our method yields better results and is 5-10 times faster than the state-of-the-art methods.

Keywords: *Image completion; approximate nearest neighbor; texture synthesis*

1. Introduction

Image completion is the technology generally applied to the problem of automatic filling-in the missing region of an image in a visually plausible way. In the recent years, it has been one of the most important technologies in the field of image processing and computer vision.

Image completion can be divided into two categories: structure-based and texture-based. The structure-based [1] scheme complete the missing region by the data fusion techniques, which the linear structures can be preserved and propagated in the target region, it is effective repair for small missing region but is prone to blurring for a larger area.

The texture-based technique [2] generates the target region with existing sample textures from the known region. It is very useful for the larger missing texture area, however the critical problem is hard to keep the main structure of image. Therefore in order to solve this problem, Criminisi, *et al.*, [3] inspired from the artificial way of repairing the image, based on the structural information to determine the filling order. This approach took both the texture and the structure into account, and significantly improved the quality of image completion. Cheng, *et al.*, [4] discover the problem that the value of filling orders would be quickly decrease to zero, which make it like a random filling order. So proposed a better method to count the value of filling order that made the algorithm robust.

However, the conventional texture synthesis methods are usually one by one to search the

similar exemplar patches from the known region of image, because the exemplar patch is a small block, and the input image is typically a very big area, so this search process is become a big and critical problem that consuming too much time.

Therefore one of the most effective methods to handle this problem is compressing the space of searching exemplars. Various speedups for this search have been proposed, Ashikhmin [5] proposed a local propagation technique exploiting local coherence in the synthesis process by limiting the search space for a patch to the source location of its neighbors in the exemplar texture. The k-coherence technique [6] combines the propagation idea with a precomputation stage in which the k nearest neighbors of each patch is cached, and later searches take advantage of these pre-computed sets. Based at He [8] proposed a new quick method which utilized propagation-assisted kd-trees to reduce the exemplar searching space. While Barnes [7, 9] used a randomized algorithm for quickly finding approximate nearest neighbor matches between image patches, because some good patch matches can be found via random sampling, and natural coherence in the imagery allows it to propagate such matches quickly to surrounding areas.

We observed based on texture synthesis methods are based on a key assumption that the texture of a damaged area has the same texture in the known region. Therefore, if we can find these region, then we only need to search similar exemplars in these region to repair the image. Inspired by this assumption^[9], we proposed based on partial region image completion algorithm. This algorithm has the following advantages that compare with other greedy image completion's algorithms.

(1) It can quickly search the most similar exemplar patch for filling patch via randomized correspondence algorithm.

(2) Searching exemplar patches in the local region could maintain the continuity of the structure and improve the similarity between the patches.

(3) Optimized the calculation similar exemplar patch method, compare the difference between the patches has the same value of Sum of Square Differences (SSD), then get the optimum patch to further enhancing the robust of the algorithm.

2. Analysis of Traditional Image Completion

The source region is indicated by Φ . Target region is denoted Ω , and its contour is denoted $\delta\Omega$. Suppose that the square Ψ_p centra at the point p will be filled. The most similar patch Ψ_q from the source region Φ is most analogous to those parts that are already filled in Ψ_p . The patch Ψ_p priority $P(p)$ is defined as the product of two terms:

$$P(p)=C(p)D(p) \tag{1}$$

The $C(p)$ is confidential term and $D(p)$ is data term, and they are defined as follows:

$$C(p) = \frac{\sum_{q \in \Psi_p \cap (I-\Omega)} C(q)}{|\Psi_p|} \tag{2}$$

$$D(p) = \frac{|\nabla I_p^\perp * n_p|}{\alpha} \tag{3}$$

Where $|\Psi_p|$ is the area of Ψ_p , α is a normalization factor. n_p is a unit vector orthogonal to the front $\delta\Omega$ in the point p and \perp denotes the orthogonal operator. During initialization, the function $C(p)$ is set to $C(p) = 0 \quad \forall p \in \Omega$, and $\forall p \in I-\Omega$, $C(q)=1$. Find the most similar patch Ψ_q of Ψ_p and copy pixels in Ψ_q to fill the missing pixels in Ψ_p and update.

$$C(p) = C(q) \quad \forall p \in \Psi_q \cap \Omega \quad (4)$$

We observe this greedy algorithm have three principal problems.

- (1) This algorithm needed much time to search the most similar patch in the whole image.
- (2) Using function (1) to calculate patch priority may cause enormous errors.
- (3) The calculation of the most similar patch is not accurate enough, which just calculate the similarity between Ψ_p and Ψ_q via simply compare the SSD.

$$d(\psi_p, \psi_q) = \sqrt{\sum_{i=1}^m \sum_{j=1}^m (p_{ij}^R - q_{ij}^R)^2 + (p_{ij}^G - q_{ij}^G)^2 + (p_{ij}^B - q_{ij}^B)^2} \quad (5)$$

$p_{ij}^R, p_{ij}^G, p_{ij}^B, q_{ij}^R, q_{ij}^G, q_{ij}^B$ Respectively denotes the pixels value of red, green and blue in Ψ_p and Ψ_q . d denotes the similarity while the smaller d 's value the more similar patches.

$$\Psi_{q^*} = \arg \min_{\Psi_q \in \Phi} d(\psi_p, \psi_q) \quad (6)$$

There may have more than one different patch that have the same least d value, and if we don't choose the best patch that could result in the accumulation of error.

3. Our Algorithm

In order to solve the above three problems, this algorithm is based on image characteristics that the feature distribute sparsely and the texture and structure are local coherence, so we adopt a randomized correspondence algorithm to search the most similar patch of every filling patch on the edge of target region, find the needed main structure and texture region, in other words we just need to find the useful partial region, reduce the redundant searching space, optimize the method of computing patch priority that based confidence factor and edge information, enhance the correct of be repairing patches' order, and make the image's structures are more inherence, Finally optimize the method of calculating the most similarly patches, when the similar patches have the same value of sum of square differences.

3.1. Initialization Partial Region

To increase the speed of image completion, we must reduce the search time for the most similar patch, the key is to narrow the searching exemplar area.

We determine the image filling region and pixels in contour $\delta\Omega$, since the contour is continuous, so adjacent pixels can be effectively propagate information, which we use randomized correspondence method to search nearest-neighbor for each pixel in $\delta\Omega$ and record its location.

We define a nearest-neighbor field (NNF) as a function $f: \delta\Omega \rightarrow \Phi$, defined over all possible patch coordinates (location of patch centers) in $\delta\Omega$, for some location function of two patches. Given patch coordinate (x, y) in $\delta\Omega$ and its corresponding nearest neighbor (x', y') in source region Φ , $f(x, y) = (x', y')$, and these values are stored in an array.

The randomized correspondence algorithm has three main components, include initiation, propagation and random search, illustrated in Figure 1.

3.1.1. Initiation: The nearest neighbor field can be initialized by assigning random values to the field, as shown in Figure 1(a), there has three adjacent solid line patches that center pixel is in contour, corresponding to three random given dashed line patches distributed in different locations.

3.1.2. Iteration: This iteration process is aimed to improve the similarity with patches after initialization. Each iteration is in accordance with a fixed scan order to find the most similar patch. There are even scan order from left to right, top to bottom and odd scan order from right to left, bottom to up. If W_i and V_i denote, respectively, propagation and random search at patch i , then we proceed in the order: $W_1, V_1, W_2, V_2, \dots, W_n, V_n$.

(1) propagation: If we search the similar patch $f(x, y)$ to the patch w centered (x, y) , we can use the coordinates of similar patches $f(x-1, y)$ and $f(x, y-1)$, which are the adjacent patches of patch w . $D(w, v)$ denote the SSD value between patch w and patch v , so the smallest value in $\{D(w, f(x,y)), D(w, f(x-1,y)), D(w, f(x,y-1))\}$ mean the patch most similar with patch w , and assign this patch's coordinate to $f(x, y)$.

The principal advantage of this method is that if several patches can correctly map to similar patches in a continuous area, the other patches can be propagated through these patches to obtain the corresponding highly similar patches. As shown in Figure 1(b), when search the similar patch of blue patch, we can check whether the red patch has a correct mapping, if it has correct one, then we can quickly to find a good similar patch without searching all patches over the image.

(2) Random search: To further improve the similar accuracy of patches, as shown in figure 1(c), define a coordinate mapping function that search the similar patch in a large region centered $f(x, y)$.

$$z_i = f(x, y) + r\alpha^i R_i \quad (7)$$

Where R_i is a uniform random in $[-1,1] \times [-1,1]$, r is a large maximum search radius, and α is a fixed ratio between search window sizes. We search patches for $i=0, 1, 2, \dots$ until the current search radius $r\alpha^i$ is below less than 1. Empirical value $\alpha=0.5$.

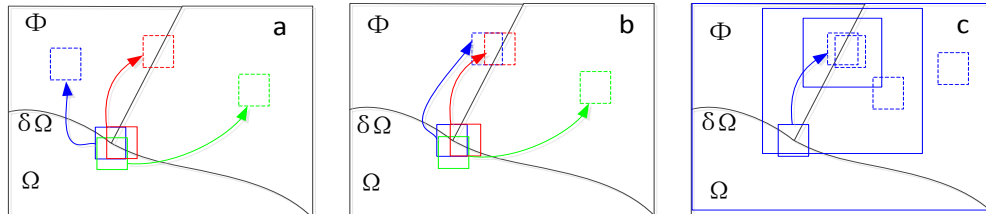


Figure 1. Random Mapping Exemplar Algorithm

We divided image into many overlap cell that size is $m \times m$ pixels, statistics the number of every cell mapped to find the most similar patches, choose the k cells of the largest mapping number. Because there have been some small overlap areas that like the red boxes in figure 2(a), so finally we should merge these areas that shown in Figure 2(b). Empirical value k is in the range 10 to 15. As shown in Figure 3, we statistics the first 9 cells of highest mapping number of Figure 2(a) sized 330×250 , here $m=60$.

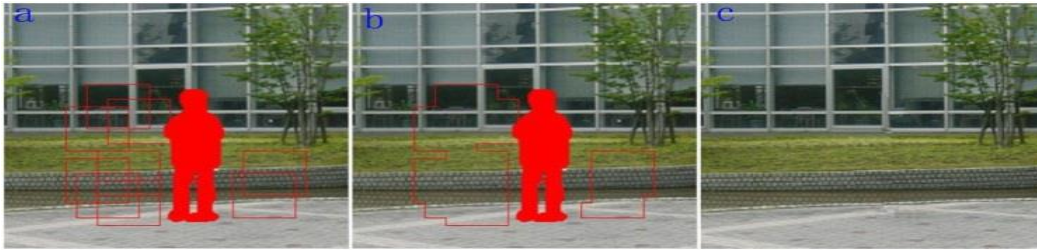


Figure 2. The Region of All Red Rectangles in the Figure (a) are Original Areas for Searching Exemplar, After Merger the Red Region are the Final Searching Areas in the Figure (b), Figure (c) is the Completion Image used our Algorithm

Where (x, y) denotes the center coordinates of the cell. We can see the final searching region are the red boxes in Figure 2(b). It includes the almost main structure and texture. Therefore, we just need to search partial region for image completion quickly and well.

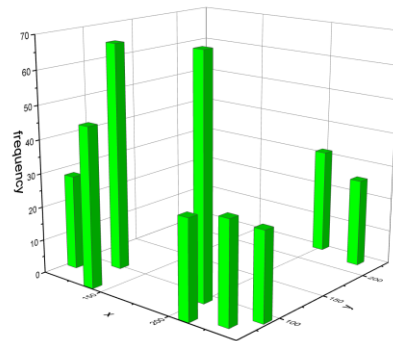


Figure 3. The Number of Cells Mapped

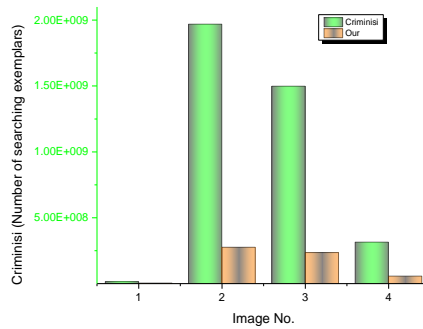


Figure 4. The Number of Patches in our Algorithm Compared with Criminisi's Algorithm

3.2. Computing Patch Filling Priorities

We define the robust priority function as follows.

$$P(p) = w_c \bullet G_c(p) + w_d \bullet D(p), \quad 0 \leq w_c, w_d \leq 1 \quad (8)$$

Where w_c and w_d are respectively the component weights of the confidence and the data terms. Note that $w_c + w_d = 1$. $G_c(p)$ denote a regularizing function to smooth the decreasing curve of the confidence term to match that of the data term as follows.

$$G_c(p) = (1-h) \times C(p) + h, \quad 0 \leq h \leq 1 \quad (9)$$

Where h is the regularizing factor for controlling the decreasing curve smoothness. Without loss of generality, h is empirically set to 0.7. Therefor the value range of $G_c(p)$ is regularized to $[h, 1]$, which is very beneficial to keep the strong structure propagating in image completion.

3.3. Searching Most Similar Patch

There may have more than one distinct patch that have the same least SSD value, and if we don't choose the most similar patch that could result in the accumulation of error. So we compare the variance of these patches and choose the smallest value patch to be the most similar patch. We first define the function to calculate the pixels mean both known area p of filling patch and the corresponding area of the similar patch as follows.

$$M = \frac{\sum_{p \in \Phi \cap \Psi} l}{\#\{p \mid p \in \Phi \cap \Psi\}} \quad (10)$$

Where l denote pixel value and $\#$ is the number of pixels. Then we define the function to calculate the variance of the similar patch area that corresponded to the filling patch unknown area q as follows.

$$S = \frac{\sum_{q \in \Phi - \Psi} (l_{q \in \Phi - \Psi} - M)^2}{\#\{q \mid q \in \Phi - \Psi\}} \quad (11)$$

4. Experimental Results and Analysis

To verify the effectiveness of the proposed algorithm, we conduct some experiments and compare the so-obtained results with the conventional approach. All experiments were run on an Intel dual-core 2.8GHZ with 4GB of RAM.

The testing images are carefully selected from both natural and synthetic images. We can see the results of the Criminisi repair algorithm that have many discontinuous situation such as in Figure 7-10. While the result in the last row, it shows abundant mountain texture in the sky. It may be a big mistake in human eyes. As increasing the number of iterations of the repair process, the filling patch confidence value was so smaller that increased the probability of error, therefore chose the wrong match patches and made the incorrect structure and texture. While using Komodakis [10] approach the repaired results are obvious blur such as in the first row and last row, and using Wexler [11] also have many obvious blur where we can in the second row and the last row. The mainly reasons of this phenomenon were not always able to find the global optimum solution via optimized the global energy function and the pixels values obtained by the weighted neighbor pixel values. Our algorithm optimized the patches priority and enhanced to search most similar patch, reduced the searching region of exemplar, thus we greatly accelerated the speed of image completion and decreased the error rate of inpainting to achieve better visual effect.

The size of all the red boxes is approximately one third of the image size in Figure 2(b), and the combination number of 9*9 pixels patches of the red boxes may be about 10% of the

entire image. Therefore, it is more effective to reduce precious time of the High Definition (HD) image, subtract the time of initialization partial region, overall, we can save nearly 85% time.

We compare the number of searching patches of figure 7-10 by our algorithm with Criminisi algorithm as shown in Figure 4. It is easy to know that the time of our algorithm grows more slowly as the increasing of image size and has faster convergence.

We compare the time and speed of our algorithm with Criminisi's algorithm as shown in Figure 5, where 333*250*3.6% denoted that the image size is 333*250 and 3.6% is the target region. As we can see our method can save more time of HD image than a small image.

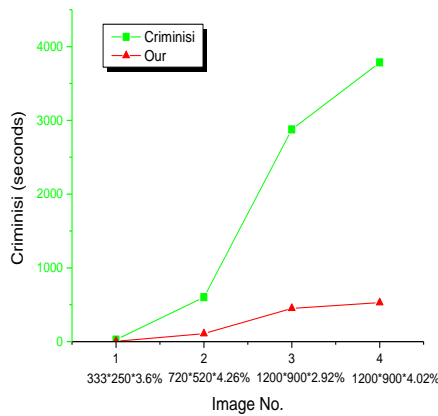


Figure 5. Our Algorithm Compared with Criminisi's Algorithm on Time

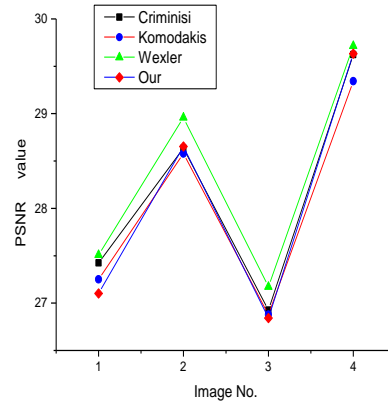


Figure 6. Image Quality Assessment with PSNR

In order to evaluate the impletion effect of our algorithm with other algorithm, we use Peak Signal to Noise Ratio (PSNR) to detect the image inpainting quality of Figure 7-10 as shown in Figure 6, which PSNR1, PSNR2, PSNR3, PSNR4 respectively denote the PSNR value by Criminisi, Komodakis, Wexler and our algorithm. It is easy to know that the repaired images by our algorithm are better than other two algorithms in image consistency and coherence from human eyes. However, it is not obvious different compared by PSNR value, because PSNR is not same with human visual system, therefore PSNR cannot completely evaluate the true effect of image completion, while image completion pursuit the effect on the human visual system to achieve a reasonable consistency and continuity.

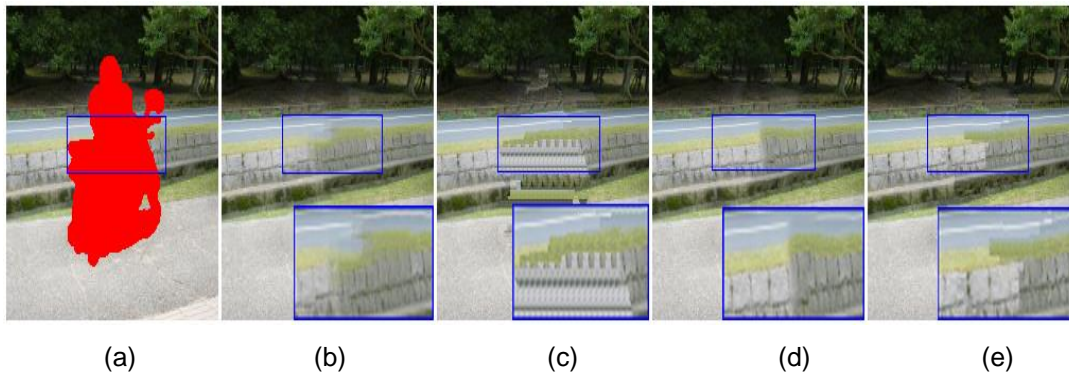


Figure 7.

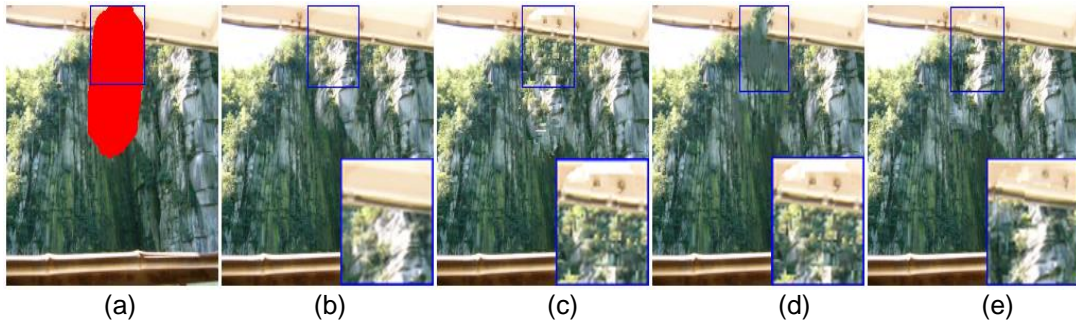


Figure 8.

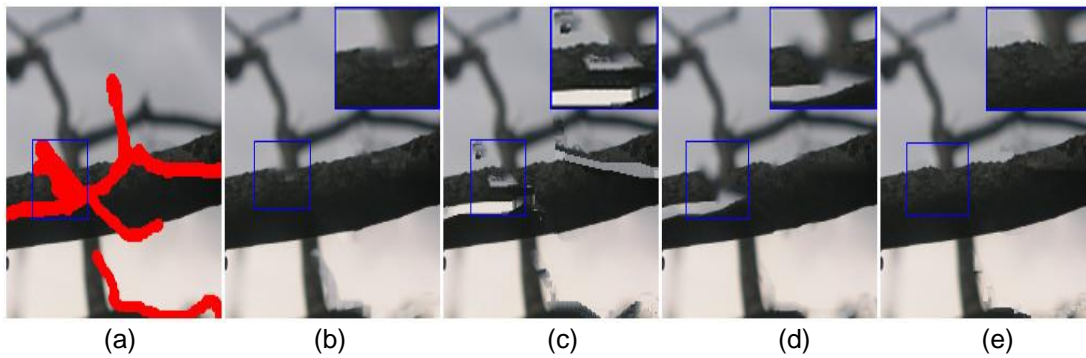


Figure 9.

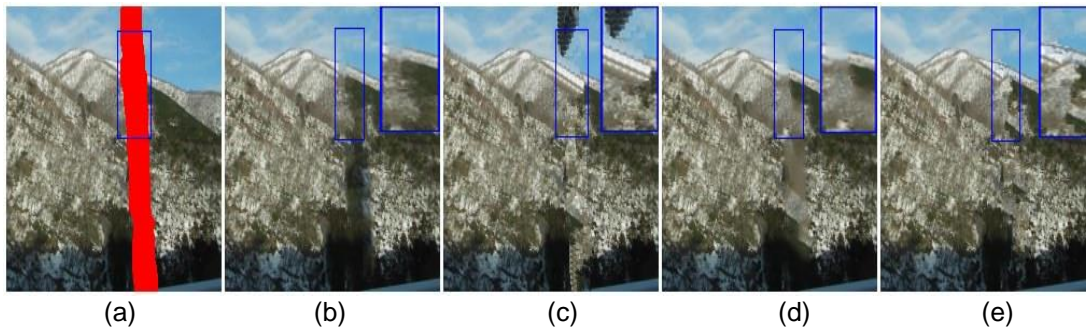


Figure 10.

Some Results. The First Column Shows Unknown Region (red). The Column 2 to Column 5 are Denoted the Results Completed by Criminisi, Komodakis, Wexler and our Algorithm, Respectively

5. Conclusions

This paper presents a fast image completion based on partial region, which can quickly well repair the structure and texture of the image. The main contribution of this work is finding the most small region including main structure and texture of the image, which uses the randomized correspondence algorithm, and optimize the method of patch priority and most similar patch. Experimental results show that the proposed algorithm is effective in both the speed greatly improvement and the visual quality improvement. However, once we selected a wrong patch, then could accumulate minor errors to a serious error in the end, this

is one big fault of the greedy algorithm. Therefore, we could further investigate how to utilize additional auxiliary information to reduce the error rate and to enhance the accurate of curved structures propagation.

Acknowledgements

This work was supported by National Natural Science Foundation of China under grant.

References

- [1] M. Bertalmio, G. Sapiro, V. Caselles and C. Ballester, "Image in painting", Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, New Orleans, Louisiana, USA, (2000), pp. 417-424.
- [2] A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling", IEEE International Conference on Computer Vision, Greece, (1999), pp. 1033-1038.
- [3] P. P. Criminisi and K. Toyama, "Region filling and object removal by exemplar-based image in painting", IEEE Transactions on Image Processing, vol. 13, no. 9, (2004), pp. 1200-1212.
- [4] W. H. Cheng, C. W. Hsieh, S. K. Lin, C. W. Wang and J. L. Wu, "Robust Algorithm for Exemplar-Based Image in painting", The International Conference on Computer Graphics, Imaging and Vision, Beijing, China, (2005), pp. 64-69.
- [5] M. Ashikhmin, "Synthesizing natural textures", Proceedings of the 2001 Symposium on Interactive 3D Graphics, ACM Press, New York, USA, (2001), pp. 217-226.
- [6] X. Tong, J. Zhang and L. Liu, "Synthesis of bidirectional texture functions on arbitrary surfaces", ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH, (2002), pp. 665-672.
- [7] E. S. Barnes, A. Finkelstein and D. B. Goldman, "Patch Match: A randomized correspondence algorithm for structural image editing", ACM Transactions on Graphics, Proceedings of ACM SIGGRAPH, vol. 28, no. 3, (2009), pp. 1-8.
- [8] K. M. He and J. Sun, "Computing nearest-neighbor fields via propagation-assisted kd-trees", IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, (2012), pp. 111-118.
- [9] E. S. Barnes and D. Goldman, "The Generalized Patch Match correspondence algorithm", European Conference on Computer Vision ECCV, (2010), pp. 29-43.
- [10] N. Komodakis, G. Tzirtas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning", IEEE Transactions on Image Processing, vol. 16, no. 11, (2007), pp. 2649-2661.
- [11] Y. Wexler, E. Shechtman and M. Irani, "Space-time completion of video", Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 29, no. 3, (2007), pp. 463-476.

Authors



Mang Xiao, he born in 1982, Ph.D., his main research interests include image processing, virtual reality.

Guangyao Li, born in 1965, Ph. D., Professor, Ph. D. , supervisor. His main research interests include graphics and image research work, virtual reality.

Yunlan Tan, born in 1972, Ph. D., her main research interests include virtual reality, visualization in scientific computation.

Lei Peng, born in 1977, Ph. D., his main research interests include image processing, virtual reality.

Yangjian Lv, born in 1990, master, his main research interests include image processing.

Li Xie, born in 1989, master, his main research interests include image processing, virtual reality.