

Fast and Robust Edge-Guided Exemplar-Based Image Inpainting

Yun Wu and Chun Yuan

Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems Graduate School at Shenzhen, Tsinghua University

Abstract. A fast and robust edge-guide exemplar-based method of image inpainting is proposed in this paper. Unlike traditional exemplar-based methods, we introduce an edge-reconstruction procedure before inpainting textures. The edge reconstruction procedure exploits different properties of edges, such as the curvature similarity, color similarity, and other estimate of how well two edges connect to each other. Guided by the reconstructed edge lines, the improved exemplar-based method is used to restore the textures and remaining structures. Moreover, we redesign the random search strategy to make it more suitable for our framework to solve the time-consuming problem caused by exhaustive search in most exemplar-based methods. After the match patch is chosen, color transfer is used to propagate the match patch information to further improve the visual quality and perceptual reasonability. Comprehensive experiments are performed to compare the proposed method with other well-known methods on synthetic images and natural images. The results show the proposed method can not only greatly reduce the computing time of exemplar-based methods, but also behave better on visual plausibility and continuity.

Keywords: image inpainting, edge, exemplar-based, PatchMatch, image completion.

1 Introduction

Image inpainting is this kind of technology whose goal is to remove some objects or restore the missing information from the source image in a way the observer can not notice the artificial trace.

In the literature, most inpainting methods can be categorized into two types, exemplar-based and diffusion-based. The diffusion-based method is often called PDE (partial differential equation)-based method because the inpainting image is often modeled as a PDE. These methods source from the heat diffusion, it propagates the information of the known area into the missing area along some direction. Bertalmio et al. [1] took the isophote direction as the propagating direction, and iteratively propagated the Laplace smooth information into the missing region until the image didn't change. Chan et al. [2] developed the Total Variation (TV) model for the image inpainting. To solve the discontinuity

problem in the TV model, curvature-driven diffusion (CDD) [3] model was introduced. These diffusion-based methods can deal the image of small damaged area without texture very well, but when it comes to the image of large damaged area or area with textures, they often result in some blurs.

On the contrary, the exemplar-based method can handle the image of large damaged area with texture well. The exemplar-based method copies the information from the known region by finding the most similar patch. However, directly copying and pasting the known information can't get a plausible result. Criminisi et al. [4] developed the patch priority, which is determined by the flux of the isophote flowing into the inpainting contour. Using the patch priority to determine the filling order, the texture information and the linear structure information can be well reconstructed. Several variants were proposed after article [4]. Cheng [5] reformed the priority function of article [4] to get more robust performance, Tao et al. [6] introduced the patch shifting scheme. However, most of these exemplar-methods still result in global structure inconsistency and use exhaustive search to find the best match patch which is very time-consuming. These two points are the main points we focus on in this paper.

As to the time cost, Kwok et al. [7] used the DCT to decompose the patches into frequency coefficients to speed up the process of finding the best match patch. Barnes et al. [8] developed the PatchMatch method, which can effectively reduce the heavy time cost. Tao et al. [9] redesigned the PatchMatch method to make it suit to their framework. But these acceleration algorithms don't consider about the inconsistency problem of the global structures. As we know, only article [10] did similar work as us, but they still use the method in article [4] without any change to fill the texture, that leads huge time cost and some visual discontinuity.

In this paper, we proposed a fast and robust edge-guided exemplar-based image inpainting method. To solve the global structure inconsistency problem, we introduce an edge-reconstruction procedure into the exemplar-based method. Exploiting different properties of edges, the fractured global edge lines are well connected. Guided by the reconstructed edge lines, the improved exemplar-based method is used to restore the textures and remaining structures. Moreover, we redesign the random search strategy to solve the time-consuming problems caused by exhaustive search. To further improve the visual quality and perceptual reasonability, we use color transfer to propagate the match patch information to the target patch. Comprehensive experiments are performed to compare the proposed method with other well-known methods on synthetic images and natural images. The results show the proposed method can gain promising improvements both on visual plausibility and computation time.

The rest of the paper is organized as follows. In section 2, we introduce edge reconstruction procedure in detail. The redesigned exemplar-based method is described in section 3. The comparison results over the typical inpainting method are presented in the section 4. We conclude this paper in section 5.

2 Edge Reconstruction Procedure

2.1 Overview of the Proposed Method

The proposed method mainly includes two important procedures, the global structure restoration procedure and the texture inpainting procedure. Given an image with a missing area, we first generate the edge line using the segmentation and edge detection algorithm. As shown in Fig. 1(b), $L_1L_2...L_m$ are the generated interceptive edges. By analyzing the local and global features of these lines, the missing lines in the missing region are reconstructed by our edge reconstruction method. Then by using the restored lines as guideline, the information near the restored lines is first inpainted by our improved exemplar-based method. Finally, the left region is inpainted by the exemplar-based method. Figure 1 shows the procedure demonstration of the proposed algorithm. The following will describe the first procedure-edge reconstruction in detail.

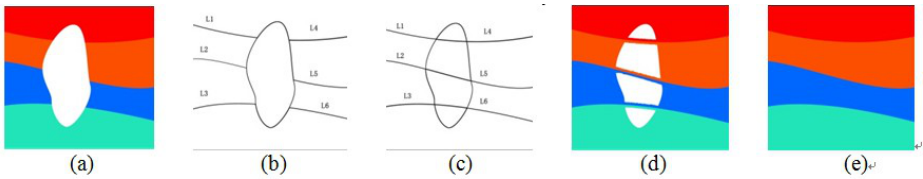


Fig. 1. Overview of the proposed algorithm. (a) The source image with white mask. (b) Edge image after segmentation and edge detection. (c) Reconstructed edge-line image. (d) Intermediate result of inpainting guided by edge image. (e) Final result of the proposed method.

2.2 Edge Generation and Reconstruction

In Edge generation the goal is to get the contour lines of the image, and itemize the lines into each segmented part. In this paper, Mean Shift algorithm [11] is adopted to segment the source image because of its relatively fast speed and efficiency. As to the edge detection, we adopt the canny edge detection algorithm [12].

After getting the edge image, we extract all the edges hitting on the missing region contour. All these edges are defined as lines $L_1L_2...L_m$ (e.g. Fig. 1(b)). Then we need to determine which two edge lines should connect to each other, which is the most important part of edge reconstruction procedure. We define as the score to estimate how well two edge lines can connect to each other. The score is calculated according to the following equation:

$$C_{i,j} = CS_{i,j} + S_{i,j} + I_{i,j} + D_{i,j} \quad (1)$$

Where $CS_{i,j}$ represents the average curvature similarity of the pair of lines, $S_{i,j}$ represents the direction consistency of the pair of lines, $I_{i,j}$ represents the color

similarity surrounding the edge lines, $D_{i,j}$ represents the distance of the pair of lines. All the four parameters have values between 0 and 1. The better the two lines match, the closer the variants are to 1. The calculation details are described in the following.

1. The average curvature similarity $CS_{i,j}$:

$$CS_{i,j} = \left| \frac{\min(\text{average}(k_{i1}, k_{i2} \dots k_{in}), \text{average}(k_{j1}, k_{j2} \dots k_{jn}))}{\max(\text{average}(k_{i1}, k_{i2} \dots k_{in}), \text{average}(k_{j1}, k_{j2} \dots k_{jn}))} \right| \quad (2)$$

Where k_{it} represents the curvature t of point i on the edge line from the crossing point with the boundary to the other direction. n is the number of points to compute the average curvature on the edge line, we here set it as 5 to 10. When the average curvatures of edge i and edge j are equivalent, the value is 1. The bigger difference the average curvatures have, the smaller the values is.

2. The color similarity surrounding the edge $I_{i,j}$:

$$I_{i,j} = \frac{\min(d_{1,2}, d_{1,3} \dots d_{2,3} \dots d_{m-1,m})}{d_{i,j}} \quad \text{with } d_{i,j} = \sum_{t=1}^m (\text{patch}_{it} - \text{patch}_{jt}) \quad (3)$$

Where the $d_{i,j}$ means the color difference of edge i and edge j , m is the number of the edges, patch_{it} is the patch center on the point t on the edge i , its size is set to be 5 to 9. $\text{patch}_{it} - \text{patch}_{jt}$ is the sum of the corresponding color difference of the two patches. The smaller the difference is, the surrounding pixels of the edge are more alike, the bigger $I_{i,j}$ is.

3. The spatial distance of the two edges $D_{i,j}$

$$D_{i,j} = \frac{\min(\text{dis}_{1,2}, \text{dis}_{1,3}, \dots, \text{dis}_{2,3}, \dots, \text{dis}_{m-1,m})}{\text{dis}_{i,j}} \quad (4)$$

with $\text{dis}_{i,j} = \sqrt{(p_{ix} - p_{jx})^2 + (p_{iy} - p_{jy})^2}$

Where $\text{dis}_{i,j}$ is the coordinate distance of the cross point with the boundary of the edge i and edge j . The smaller the distance is, the bigger is.

As shown in Fig. 2(a), $L1L2L3L4$ have similar curvatures, and if textures surrounding these edges are also similar, then as the method in article [10] states, the spatial distance would decide which two edges should connect. Finally, the result of Fig. 2(b) is obtained. We can notice the unnaturalness very easily. So we introduce a new feature-direction consistency to solve the problem in Fig. 2(a). We first compute the distance of the centers of curvature circles. As shown in Fig. 2(c), the distance of $R1$ and $R4$ are much nearer, so we are more prone to connect $L1$ and $L4$, and so do with $L2$ and $L3$. The detail definitions are given in the following.

4. Direction Consistency $S_{i,j}$:

$$cd_{i,j} = d(\text{center}_i, \text{center}_j), \text{center}_i = g(\text{center}_{i1}, \text{center}_{i2} \dots \text{center}_{in})$$

$$S_{i,j} = \frac{\min(cd_{1,2}, cd_{1,3}, \dots, cd_{2,3}, \dots, cd_{n-1,n})}{cd_{i,j}} \quad (5)$$

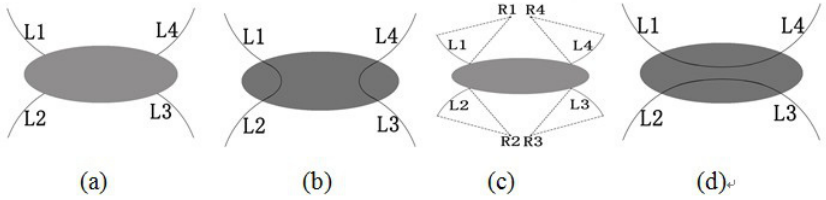


Fig. 2. (a) The situation method in article [10] can not deal with. (b) Unnatural result from method in article [10]. (c) Centers of curvature circles. (d) Natural result from the proposed method.

Where g returns the geometric center, $center_{ik}$ is the center of curvature circle of k th points on the edge i , n is the number of points, n is usually set to be 3 to 7, $cd_{is_{ij}}$ is the coordinate distance of the two centers. The smaller the distance is, the bigger $S_{i,j}$ is.

After computing all the scores of the every pair of edge lines, we choose the pair with the highest score every time. Then we use *NURBS* curve [13] to fit the two lines to connect the pair of edge lines. *NURBS* curve is a powerful method to design curves. It is geometrically invariant and affine invariant. It is very convenient to joint two curves. Taking into consideration these advantages, we decide to use it to fit the broken edges.

Until all the lines are connected or all the pairs' scores are less than a threshold TH , we stop the connecting procedure. It would be appropriate to set the threshold TH as values between 0.3 and 1.5, we here set it as 0.7.

3 Improved Edge-Guided Exemplar-Based Inpainting Procedure

3.1 Redesigned Generalized Random Search Strategy

Here we first review the PatchMatch algorithm introduced by the Barnes [8], which is a very efficient method to solve problems like this: given an image I , for every square patch center on the pixel of I , find out the approximate best match patch in an image S , with SSD (sum-squared difference) between corresponding pixels as the measure.

The original PatchMatch algorithm includes three steps, initialization, propagation and random search. For patch T center on pixel (x, y) , define $f : I \rightarrow R^2$ as the nearest neighbor function. Then $f(x, y)$ returns the corresponding patch offset of $T(x, y)$.

In the initialization step, all the patches are assigned to their corresponding patch randomly. Of course, you can adopt other strategies to initialize, such as basing on the prior knowledge or up-sampling the corresponding patch of the low resolution level.

The propagation procedure tries to improve the best match using the known best match above or the left. Let $f(x, y)$ denotes the patch difference between patch centers on (x, y) and $(x, y) + v$. So the candidate best patch comes from $\min D(f(x, y)), D(f(x - 1, y) + (1, 0)), D(f(x, y - 1) + (0, 1))$. Through this propagation, if (x, y) has a good match, then all the patches below and right to (x, y) will have good matches. Besides, on the even iteration, scan the pixel on the reverse order, propagate the good match of down and right.

The propagation procedure converges very fast, but it may trap in the local mini-mal. So the random search strategy is used: candidate is sampled from an window randomly, and $f(x, y)$ is updated if any of the candidate has smaller distance. Let v_0 be the current best offset of $f(x, y)$. Then the candidate is randomly chosen by search around v_0 at an exponentially decrease radius: $u_i = v_0 + w\alpha^i R_i$, where w is the maximum search radius, α is the ratio decreasing the window size, R_i is a random in $[1, -1] \times [-1, 1]$. The index i increases from 1 to n until the window size is less than 1.

Tae-o-sot et al. [9] changed the random search procedure. As the radius decreases, the search center no longer remains unchanged. Every time the radius decrease, it lets the previous best patch be the search center: $u_i = u_{i-1} + w\alpha^i R_i$. Obviously, this change makes it more accurately to find the better patch. But it brings another problem: if the final best patch is far from u_i , this change may end up with a local optimum, figure 3(a) demonstrates this situation. Suppose q_i is the match patch, and the smaller the index is, the better the patch matches. When $w=2$ (the red box), if q_9 is not selected, for example, q_8 is selected, then the best match in the top right corner would be missed. But we here save k best match patches, then q_9 is very likely to be saved in the list, and the best match patch will be also more likely to be selected. The following is the details.

We save k best match patches p_0, p_1, \dots, p_{k-1} in the search list. p_0 is initialized as v_0 , then act as the best patch match, and p_{k-1} is the k th best match patch. Every time inserting one better patch into the list and delete the worst patch from the list, we changed the random search strategy as follows:

$$u_i = \min(p_0 + w\alpha^i R_i, p_1 + w\alpha^i R_i, \dots, p_{k-1} + w\alpha^i R_i) \quad (6)$$

If u_i is better than p_k , we insert it into the list at appropriate position, and delete p_k from the list. That strategy makes the random search not be trapped into local optimum. And because the k is a constant and the most time-consuming procedure in the PatchMatch is not random search, so our redesigned scheme almost behaves as fast as the original scheme, and gets more consistent results than the original one. Figure 3(b) shows the running time of the original PatchMatch [8], kd-tree [14] and our scheme on different images with different number of pixels.

3.2 Color Transfer

In the typical exemplar-based methods, once the best match patch is selected, the pixels of the patch are copied to the target patch without any change.

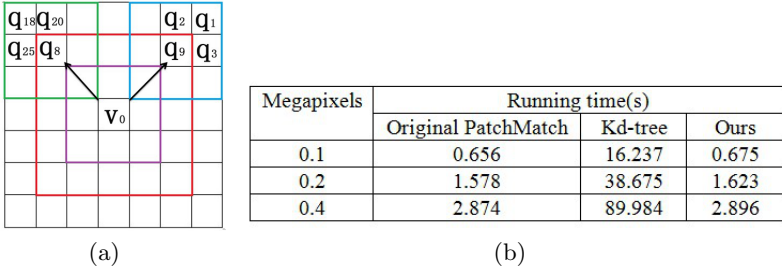


Fig. 3. (a) The situation the method in [9] fails. (b) Running time contrast of different schemes.

Although the selected patch is the best match patch in the known region, it is not always in the same distribution with the target patch, which may lead some unnatural result in the end. Here we use some simple statistical feature to transfer the selected patch to fit the target patch.

As shown in articles [15] [16], image color characteristics affects human visual sen-sibility heavily. So we use the skewness and the kurtosis to transfer the selected patch to make it more similar with the target patch. Skewness is to measure uniformity of data distribution. Its value is M_3/σ^3 , M_3 is the third moment of the distribution, σ is the standard deviation. We use the following equation [17] to transfer the skewness:

$$Y_i(\lambda) = \begin{cases} \frac{(x_i+t)^\lambda - 1}{\lambda g_1^\lambda} & \lambda \neq 0 \\ g_1 \ln(x_i + t) & \lambda = 0 \end{cases} \tag{7}$$

g_1 is the geometric mean of source data, we should ensure $(x_i + t) > 0$, and adjust λ to change the skewness.

We use the following equation [17] to transfer the kurtosis, which equals to M_4/σ^4 , M_4 is the fourth moment of the distribution:

$$Z_i(\lambda) = \begin{cases} \text{sign}\left(\frac{(|x_i-b|+1)^{\lambda^2}-1}{\lambda g_2^{\lambda^2-1}}\right) & \lambda \neq 0 \\ \text{sign}(g_2 \ln(|x_i - b| + 1)) & \lambda = 0 \end{cases} \tag{8}$$

$$g_2 = \left(\prod_1^n (|x_i - b| + 1)\right)^{1/n}, \text{ sign}=\text{sgn}(x_i - b)$$

Where b is the geometric or arithmetic mean, we also transfer the target through ad-justing λ . When the difference of the skewness is bigger than that of kurtosis, we mainly transfer the skewness, else transfer the kurtosis. After the transfer, we copy the transferred data to the target patch. And that returns more robust results.

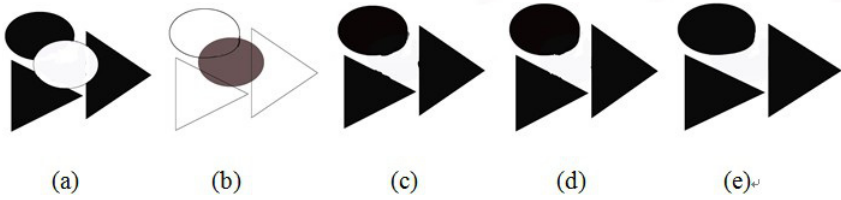


Fig. 4. Restore the synthesis image. (a) The source image with an ellipse missing region. (b) Reconstructed sketch line image. (c) The result of the method in article [4]. The method can only deal with linear structure, so the reconstructed curves are obviously inconsistent. (d) The result of method in article [7] (e) Final result of the proposed method. The curve of the ellipse and the truncated straight lines are well reconstructed.



Fig. 5. Restore the natural scene image. (a) Source image with a man. (b) Image after removing the man and his reflection in the water. (c) The result from the method in article [4]. The grass lines have obvious artifact, the mountain lines are also not so well reconstructed. (d) The result of method in article [7] (e) The result of the proposed method. The structures and textures are well restored in a way human almost can't detect the artifact.

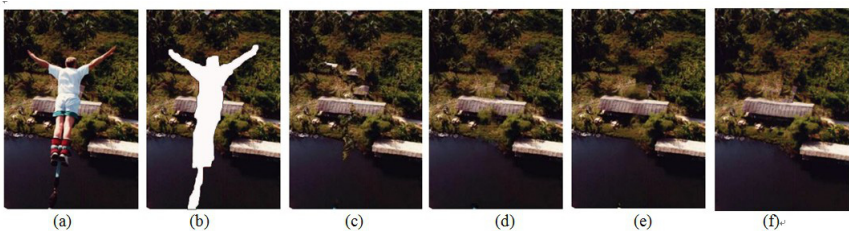


Fig. 6. Removing large object. (a) Source image. (b) Removing the man from source image. (c) Result of method in article [4]. (d) Result of method in article [7]. (e) Result of method in article [9]. (f) Result of the proposed method. Notice that the structures of the roof are well restored.

4 Experiment Results

The proposed method is experimented on many kinds of images, from the natural pictures to synthesis images. The final results show the robustness and the effective-ness of the proposed method.

We first test the proposed algorithm on a synthetic image with two triangles and one ellipse to show how the algorithm reconstructs the structure of the synthetic im-age. As shown in Fig.4, the curve can be well reconstructed; the texture also can be filled appropriately guided by the reconstructed edge lines. However, the method in article [4] can only restore some linear structure and may lead to some inconsistency. The method in article [7] also results in some artifact.

Figure 5 demonstrates the results to inpaint the natural scene pictures using our al-gorithm. There are a variety of textures and structures near the missing region. As shown in (e), both the structures and the textures are restored in a plausible way by our algorithm. By contrast, the method in article [4] behaves not so well in reconstruct-ing the boundary of the different textures and re-sults in some artifacts. The proposed method also behaves better in structures restoration than the method in article [7] as shown in the red box.

The results of using different methods to remove large object are shown in Fig 6. Our method shows the superiority in restoring the structures and textures, especially in the roof (as marked in the red box), and returns comparative even better results over the typical and improved exemplar-based methods.

5 Conclusions

In this paper, we proposed a fast and robust edge-guided exemplar-based inpainting method basing on the basic truth that an image is a variety of textures organized by the global edge lines. Different from the method in article [4], we not only use the local structures to guide our texture inpainting, but also use the global structures restored by our algorithm, which makes both the structures and the textures be completed consistently. To solve the time-consuming problem of most exemplar-based meth-ods, we redesign the random search strategy to search the match patch. Besides, color transfer is used to further improve the visual consistency. In the future, we consider continuing to perfect our method of restoring edge lines, and test if we can apply the PDE method in our framework.

Acknowledgment. The work is supported by National Significant Science and Technology Projects of China under Grant No. 2013ZX01039001-002-003, National Natural Science Foundation of China Project under Grant No.61170253, Promotion Project of Shenzhen Key Laboratory on Information Science and Technology 2012.

References

1. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: ACM Comput. Graph (SIGGRAPH 2000), pp. 412–424 (2000)
2. Chan, T.F., Shen, J.: Mathematical models for local non-texture inpaintings. *SIAM J. Appl. Math.* 62, 1019–1043 (2001)
3. Chan, T.F., Shen, J.: Nontexture Inpainting by Curvature-Driven Diffusions. *Journal of Visual Communication and Image Representaion* 12, 436–449 (2001)
4. Criminisi, A., Perez, P., Toyama, K.: Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Transaction on Image Processing* 13, 1200–1212 (2004)
5. Cheng, W.H., Hsieh, C.W., Lin, S.K., Wu, J.L.: Robust Algorithm for Exemplar-based Image Inpainting. In: Proc. Int. Conf. Comput. Graphics, Imaging Vis (CGIV), pp. 64–69 (2005)
6. Tae-o-sot, S., Nishihara, A.: Exemplar-based image inpainting with patch shifting scheme. In: 17th International Conference on Digital Signal Processing, pp. 1–5 (2011)
7. Kwok, T.H., Sheung, H., Wang, C.C.: Fast Query for Exemplar-Based Image Completion. *IEEE Transaction on Image Processing* 19(12), 3106–3115 (2010)
8. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transaction on Graphics* 28(24) (2009)
9. Tae-o-sot, S., Nishihara, A.: Iterative gradient-driven patch-based inpainting. In: Ho, Y.-S. (ed.) PSIVT 2011, Part II. LNCS, vol. 7088, pp. 71–81. Springer, Heidelberg (2011)
10. Chen, Y., Luan, Q., Li, H.Q., Au, O.: Sketch-Guided Texture-Based Image Inpainting. In: Proc. IEEE International Conference on Image Processing, pp. 1997–2000 (2006)
11. Cheng, Y.: Mean Shift, Mode Seeking, and Clustering. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 17(8), 790–799 (1995)
12. Canny, J.: A Computational Approach to Edge Detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 8(6), 679–698 (1986)
13. Piegel, L., Tiller, W.: *The NURBS Book*. Springer (1995-1997)
14. Wexler, Y., Shechtman, E., Irani, M.: Space-Time Completion of Video. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 29(3), 463–476 (2007)
15. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. *IEEE Computer Graphics and Applications* 21, 34–41 (2001)
16. Zhao, G.Y., Xiang, S.M., Li, H.: Application of Higher Moments in Color Transfer between Images. *Journal of Computer-Aided Design & Computer Graphics* 16(1) (2004)
17. Gao, H.X.: *Statistic Computation [M]*. Beijing Peking University Press (1995) (in Chinese)