On the innovation and application of texture synthesis technique in image processing

Jing Lu and Ming-Tao Sun†
School of Economics and Management,Bei Hang University ,
Bei Jing, 100191, China
E-mail: lujing@buaa.edu.cn, Sunmt@buaa.edu.cn

Texture synthesis skill is widely used in fields of Computer Graphics, Computer Vision, Image Processing and so on. It can efficiently avoid noticeable seams between texture patches and minimize the stretch and distortion of the pattern when tiling a texture on surfaces.This paper focuses on two problems aroused in innovation and application of texture synthesis. Firstly, A. Criminisi, P. P´erez in the University of Cambridgeonce raised a arithmetic which had performed well when removing objects in an image or picture. In this paper, during the process of carrying out this arithmetic, we raised a deeper discussionand comprehension of the arithmetic.Secondly, we described a texture synthesis method based on multiple samples, which provides a general texture synthesis method.

Keywords: Exemplar; Texture synthesis; Image inpainting; User interaction.

1.Introduction

In the past few years, the image-based rendering technique has developed well, it can get novel results by applying images and show details which the geometric model cannot reflect, and cover the shortage of geometry-based rendering. Texture synthesis skill plays an important role in photorealistic rendering, especially in recent years, the texture synthesis skill has achieved significant progress and drawn great attention from researchers all around the world.

The texture synthesis technique can overcome the shortages of traditional texture mapping method, and avoid the tedious adjustment of procedural texture synthesis, therefore, it has obtained increasing attention from researchers. It has become a focus in the fields of computer graphics, computer vision and image processing. By using texture synthesis technique, we can also achieve texture inpainting (e.g. to patch cracking picture for original effect), texture transfer (e.g. to post a picture texture into another one), it can use a piece of video image to generate non-repetitive video animations of random lengths to expand to time domain. That's why the texture synthesis technique has a wide application

prospect in image editing, data compression, burst transmission of network data, large scale scene generation and realistic and non-realistic renderings and so on.

2.Image Inpainting from Samples

2.1 Principles

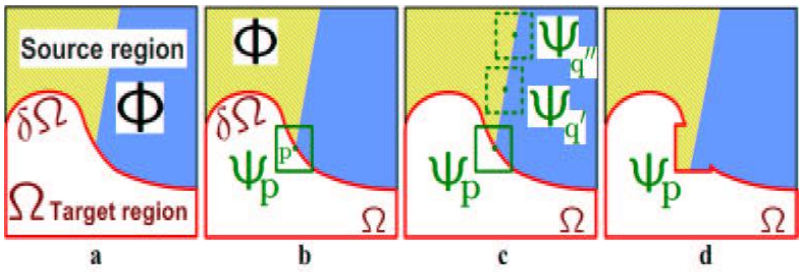


Fig. 2.1 texture synthesis deals with structural breeding and derivative

The above picture illustrates how texture synthesis from samples deals with structural breeding and derivative. (a) is the original picture with the target region Ω, its contour line δΩ and sample source region Φ is clearly marked; (b) we hope that we can use the image information ΨP of point P on the contour line of the target region to delimit expected synthetic regional limit; (c) the most likely matching candidate comes from the point information which locates in the boundary between two textures in the sample source region, such as Ψq'and Ψq" in the picture; (d) we can see that the best matching image information has been copied to the region once occupied by ΨP and the padding of partly target region is realized. In fact, the target region has already finished shrink to some extent.

2.2 Arithmetic Description

The current inpainting method takes pixel blocks as inpainting units, that is, the pixel blocks repair pixels on the edge of the region. The whole inpainting process includes three steps:

1. The priority of each patch on the "fill front" should be calculated, and the patch with the highest priority is obtained.
2. The optimal fill sample should be selected.
3. The confidence value should be updated

2.2.1Priority Calculation and Patch Confirmation

The formula of priority is:

1.Priority definitionP(p)=C'(p)*D(p)

In which C'(p) represents the confidence value of the patch

$$C'(p)=\sum C(q)/ |\Psi p|$$

$|\Psi p|$ represents the area of Patch Ψp, it shows the patch size, C(p) represents the confidence value of a certain pixel inside a patch, we define that every pixel in the image has two values: confidence value and color value.

In the beginning the confidence value of the pixel in the inpainting region is assumed as C(q)=0;

Those outside the inpainting region as C(q)=1.

Therefore, the greater the number of pixels in the sample region, the more confidence value we get!

In which D(p) represents the isophote intensity of patch P on the "fill front".

$$D(p)=|I(p) \cdot np|/\alpha$$

I(p) represents the isophotedirection and intensity of P

α is the normalizing factor (full gray scale picture α=255)

np represents the normal vector of P on the "fill front".

Therefore, the bigger the isophote intensity and the smaller the include angle between the isophote and the normal, the greater calculated data value we get. It helps the linear structure of the image get repaired in priority.

The patch with the highest priority can be confirmed by the priority of the patches around the front filled contour line. We can conclude from the above definition that the priority is used to decide the sequence of image padding inpainting, therefore the image sampling process could be well organized. The consistency of image inpainting can be ensured at the same time.

2.2.2 Image Data Copying of the Optimal Exemplar Block

We search one certain block (the same size of the patch block) in the sample region to obtain the optimal exemplar block. We assume the patch with the highest priority as Ψp, then the optimal exemplar block needs to be:

$$\Psi q = \arg\min d(\Psi p,\Psi q')$$

In which Ψq is a randomly chose exemplar block, all of its pixels locate in the sample region.

$d(\Psi p,\Psi q')$ represents the distance between block Ψp and block Ψq, it refers to the quadratic sumof the difference between the color values of the two filled pixels in two blocks. Optimal exemplar block Ψq represents the block which makes the distance value $d(\Psi p,\Psi q')$ the minimum in the distance ensemble. After finding the optimal exemplar block, all we need to do is filling the pixel information into corresponding places of the patch.

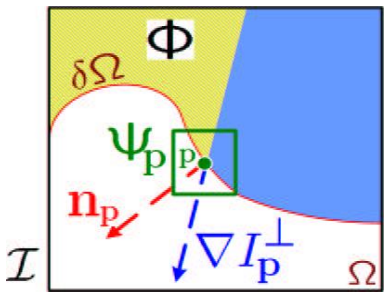


Fig. 2.2.2 the image information of Ψp

np represents the normal of contour line of target region, ▽p⊥ represents the equipotential point of P in direction and luminance. In order to copy the optimal exemplar block, the whole picture can be marked as I indicates.

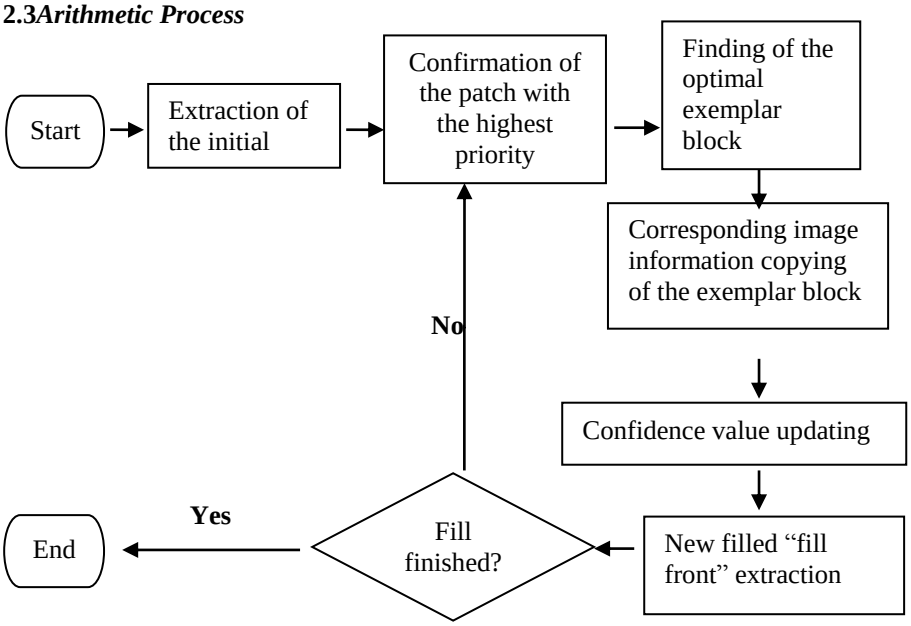
2.2.3 Confidence Value Updating

After filling the patch, the filled pixels has changed from damaged region to sample region

$$C(p) = C(p')$$

We can get a new "fill front" after confidence value updating, it means the damaged region is well repaired if it shows no pixel in the new area, otherwise repeat the above steps.

2.3*Arithmetic Process*



3.Texture Synthesis of User Interaction

3.1*Ashikhmin Fast Texture Synthesis Arithmetic*

3.1.1Arithmetic Description

Based on the basic synthesis arithmetic, Ashikhmin[1] fast texture synthesis arithmetic takes advantage of the spatial correlation between pixels, so that the overall arithmetic which found the matching before only needs to find a few points, and the arithmetic efficiency raises greatly.

The searching idea for optimal point matching is shown as below (Fig.3.1): according to the L-shaped neighbor midpoint of the current synthesis point (e.g. A), we find the matching point in the input sample image (e.g. A_1), then shift for corresponding amount for a waiting point (e.g. P_1). We compare the distances between each waiting point and L2, the L-shaped neighbor of P, and choose the distance minimum as the optimal matching point.

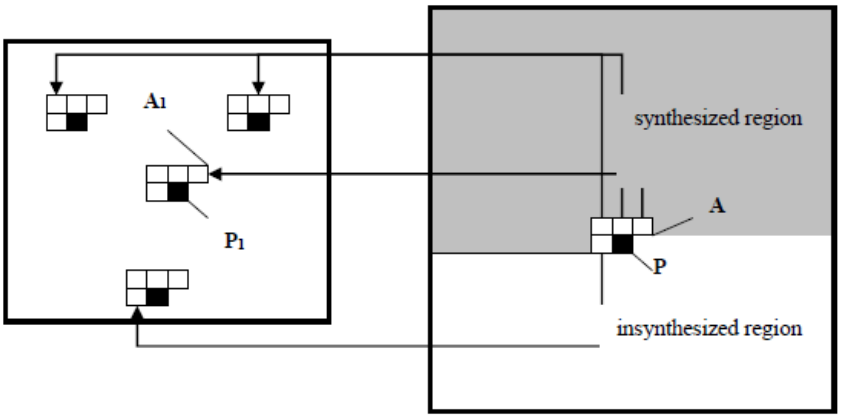


Fig. 3.1 Optimal matching point searching in the neighborhood

Assume that we need to synthesize the current pixel P, because the four pixels in the L-region of P are already synthesized, we could find the four points A1 A2 A3 A4 in the input sample image, and shift the coordinate of A1, then we get the waiting point P1 of P, P1 is obtained by A1+(-1，-1). By such analogy, the test waiting points are obtained as P1 P2 P3.

The main process of Ashikhmin fast texture synthesis arithmetic is shown as below:

At the beginning, we record the location array of matching point of each pixel in the synthesized image, and set them as the random sampling point of the input image;

for output image, we synthesize each pixel P according to the sequence of scanning line:

- The current L-region of P should be considered. For each point in the neighbor region, we shift the corresponding location based on that of the matching points in the array and select it as a waiting point in the sample image;
- The repeated waiting points should be eliminated;
- The waiting point which has the minimum distance with the L-region of P should be selected and copied to the current point in the output image, the location should be recorded.

3.1.2 *Synthesis Result*

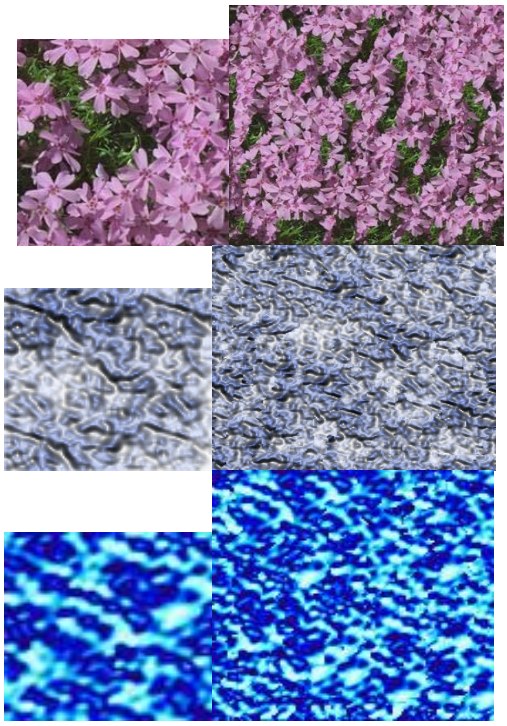


Fig.3. 2 Result of Ashikhmin fast texture synthesis arithmetic

3.2*Interaction*

Ashikhmin fast texture synthesis arithmetic can achieve good effect. During the process, if the user can control the output result to some extent, then the user satisfaction could be raised. When discussing the neighborhood, we have already knew that the square neighborhood window has non-causality, it can add

the pixel value effect of the region to be synthesized into the pixel for synthesis. Therefore, if we preset certain texture structure into the image to be synthesized, meanwhile, if we change the L-shaped window into a square window when calculating the L2-distance of neighborhood window, the preset texture structure could influence the synthesis result. Fig. 3.3 is the situation in which the neighborhood window is changed into a square window in Fig. 3.1. It is worth noting that when selecting the waiting point, only the L-shaped subregion synthesized in the square window is considered.

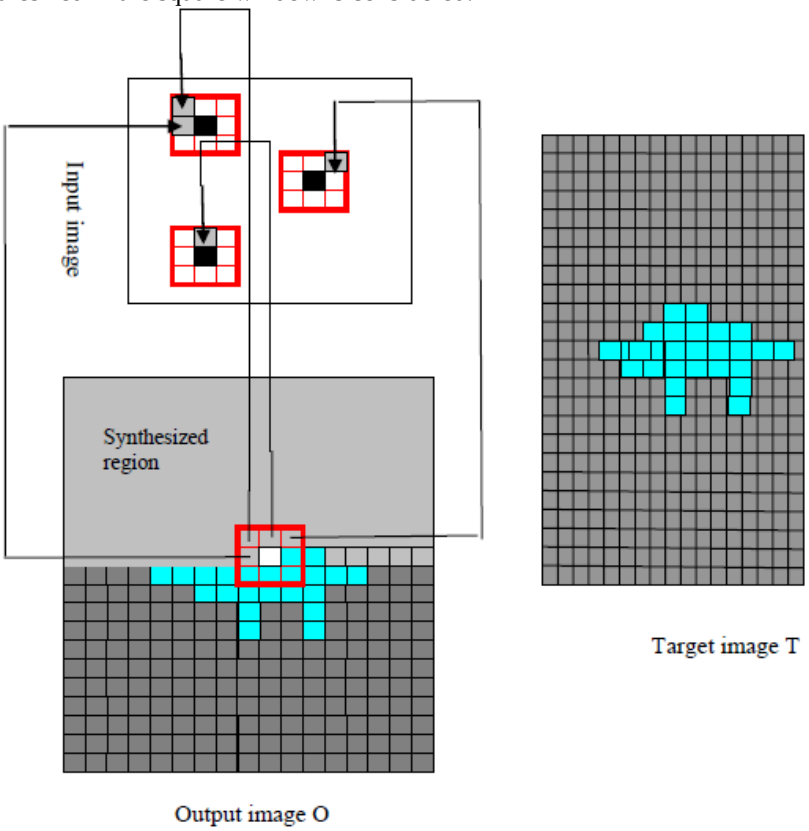


Fig. 3.3 Waiting point finding under the square window

Fig. 3.4 is one of the synthetic results. Fig. 3.4 (b) is the preset texture in control, if we hope for more texture of red flower in a certain area, we should preset red pixel there. If we hope for a pattern of red flower in the synthesized image, we could use red pixel to preset expected pattern structure.

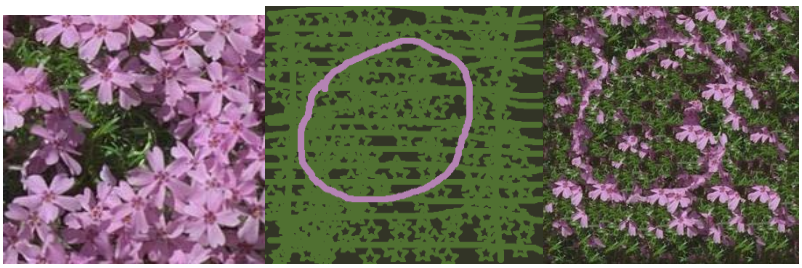


Fig. 3.4 Synthesis result image by user interaction. (c)is the target image by synthesizing (a) and (b)

4. Application

4.1*Current Application*

We design a software to show the innovation and application of texture synthesis technique in image processing based on the image inpainting arithmetic and the fast texture synthesis arithmetic with interaction, considering the development trend of synthesis technique and our resource and capability. The software is much easily operated than others, such as Photoshop. The process is totally automatic, and handled easily. It also works well.

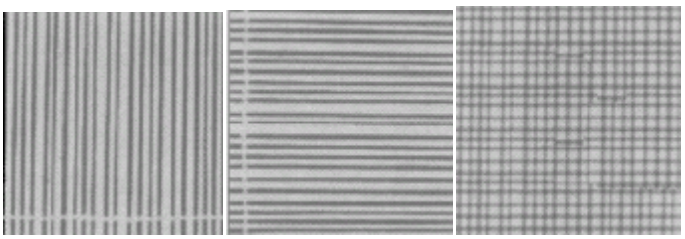
The extension based on the arithmetic proposed in the current study can be applied to computer animation, film special effect, visual reality technique and so on.

4.2*In-depth Exploration for Application*

4.2.1*Texture mixture*

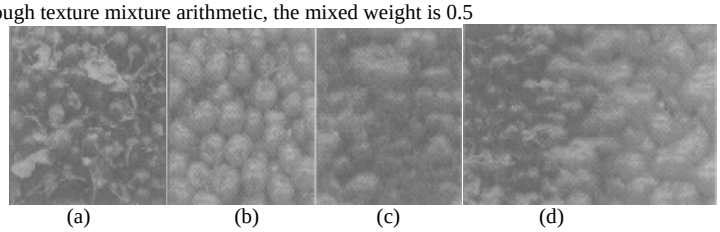
Texture mixture refers to the operation of mixing different texture characters to generate a new texture, it is the application of various texture arithmetic. There are many examples of texture mixture, for example, a pattern texture mixes with another one of different background color and generates a new texture. It can also be two different pattern textures mixing together and generating a new seamless mixed texture sample. We can also combine different textures of blue sky and white cloud t get a sample image of cloudy sky. Texture mixture can capture the characters of several texture samples at the same time, which is useful for new texture generation.

There are many methods for texture mixture, and they are mainly applied in parameter spaces of different texture mixtures. For example, we can synthesize texture mixture in steerable pyramid domain, or we can use statistic learning to mixed synthesize the sample image.



(a) (b) (c)

Fig. 4.1 Sample effect of texture mixture.Line (a) and Line(b)are sample textures, Line (c) is the result through texture mixture arithmetic, the mixed weight is 0.5



(a) (b) (c) (d)

Fig. 4.2 Different mixed result under different weights.Line (a) and Line(b)are sample textures, Line (c) is the result with a mixed weight of 0.5, Line (d) is the result with a mixed weigh of 0.4

Fig. 4.1 shows the result through texture mixture arithmetic, the mixed weight is 0.5. If we change the mixed weight, the effect changes as well. Fig. 4.2 specifically shows the results of different weights.

4.2.2 *Texture Transmission*

Texture transmission and texture synthesis of double samples have similar arithmetics. Double sample-based texture synthesis finds the optimal matching point in two samples to generate new texture images, texture transmission, however, transmits or transfers a sample texture to target image to achieve the expected effect.

As shown in Fig. 4.3, sample image S is the texture to be transmitted, T is the target image, O is the mixed image after texture transmission. The widths and heights of O and T are the same and O totally reserves the basic information of T and it mixes the texture of S, visually the texture of S becomes the background texture of T.

(a)Input sample image S (b)Target image T (c)Target image T

Fig. 4.3 A typical sample of texture transmission

References

1. Michael Ashikhmin. *Synthesizing natural textures*. In Proceedings of 2001 ACM Symposium on Interactive 3D Graphics, pages 217--226, March 2001.
2. Marcelo Bertalmio, Guillermo Sapiro, Vicent Caselles, and Coloma Ballester. *Image inpainting. Proceedings of SIGGRAPH 2000*, pages 417-424, July 2000.
3. Paul Billault. *Texture Synthesis Algorithms.*http://w3.impa.br/~billault/.
4. P Harrison. *Patchwork texture synthesis*. Technical Report 2002/119, Monash University School of Computer Science and Software Engineering.
5. Li-Yi Wei, *Marc Levoy.Fast Texture Synthesis using Tree-structured Vector Quantization*. SIGGRAPH '2000
6. *Object Removal by Exemplar-Based Inpainting*，Microsoft Research Ltd., Cambridge, UK Microsoft Corporation, Redmond, WA, USA
7. Li-Yi Wei, *Marc Levoy.Texture Synthesis over Arbitrary Manifold Surfaces.* SIGGRAPH 2001, Computer Graphics Proceedings
8. G. Turk. *Genereating Textures on Arbitrary Surfaces Using Reaction-Diffusion.* In Computer Graphics (SIGGRAPH ’91 Proceedings), volume 25, pages 289–298, July 1991.
9. H. Igehy and L. Pereira. *Image replacement through texture synthesis. In Proceedings of the 1997 IEEE International Conf*. on Image Processing, 1997
10. Alexei A. Efros and William T. Freeman. *Image quilting for texture synthesis and transfer*. In Proceedings of SIGGRAPH 2001, pages 341--346, August 2001. 2