Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China, 2-4 Nov. 2007

FAST TEXTURE TRANSFER THROUGH THE USE OF WAVELET-BASED IMAGE FUSION

JIWON KIM

Electrical Engineering and Computer Science, Massachusetts Institute of Technology, MA 02139, USA
E-MAIL: jiwon@mit.edu

Abstract

This paper describes a new texture transfer scheme based on wavelet image fusion. Texture transfer applies the pattern of the texture image to the source image. While most of the current state-of-the-art methods perform texture transfer in space domain, the scheme introduced in this paper processes images using image fusion in wavelet domain. Image fusion is a popular method to process information from multiple images and combine the images into a smaller number of images. In the application to texture transfer, the approximation of the source image is extracted and used for the synthesized image. In addition, edges of the source and texture images are extracted and preserved in the synthesized image. This is done through the use of 2-D wavelet decomposition. Comparison to the current state-of-the-art methods is provided in the paper. It is concluded that wavelet-based texture transfer performs a reasonable texture transfer, and it is relatively fast.

Keywords: Texture transfer; information fusion; image fusion; wavelet image processing;

1 Introduction

Texture is a ubiquitous visual experience. It can describe a wide variety of surface characteristics such as surfaces of flowers, skin, and terrain. Since reproducing the visuals of the physical world is a major goal for computer graphics, there have been many researchers studying properties of texture and synthesizing and transferring it.

In many applications, it is useful to have the ability to create a texture of arbitrary size given a small sample texture. *Texture synthesis* techniques perform this operation. For texture synthesis, a clear criterion of success exists: the local regions of the result have to look like the input sample. There have been many papers on texture synthesis, and those include [3], [2], and [1]. The method introduced in [3] successfully performs a very fast texture synthesis using tree-structured vector quantization.

Texture transfer, on the other hand, does not have a clear goal. The degree of similarity with the origianl source image is usually adjusted based on user preferences. In a similar way, different people have different preferences over intensities of transferred texture. Especially, the case of artistic style transfer, one of texture transfer applications, illustrates this point well. The definition of artistic style is subjective. Therefore, users would take experiments with different texture transfer algorithms and parameter values to find what they want.

In this paper, a new texture transfer technique based on wavelet techniques is introduced. This algorithm runs in linear time, and it performs a reasonable texture transfer. One difficulty of using this method is that it presumes the existence of a texture image whose size is the same as or larger than the source image. Achieving a large texture image can be done using one of the texture synthesis algorithms introduced above. By keeping a database of large texture images, texture transfer can be done without running synthesis algorithm as a part of texture transfer process. From now on, the texture image is assumed to have the same size as the source image.

The technique introduced in this paper gives a variety of artistic effects depending on parameter values. In addition, these results have different visual characteristics from other texture transfer algorithms. The decision of which algorithm to use solely depends on visual preference of an user. Aside from the quality of synthesized images, one advantage of using the technique in this paper is that it performs texture transfer very fast.

Section 2 introduces the basic concepts of image fusion techniques. This is the fundamental technique used for our texture transfer method. In Section 3, a wavelet-based image fusion technique is applied to texture transfer. This technique is extended to perform progressive texture transfer in Section 4. Finally, our texture transfer method is compared to some of the existing texture transfer methods in

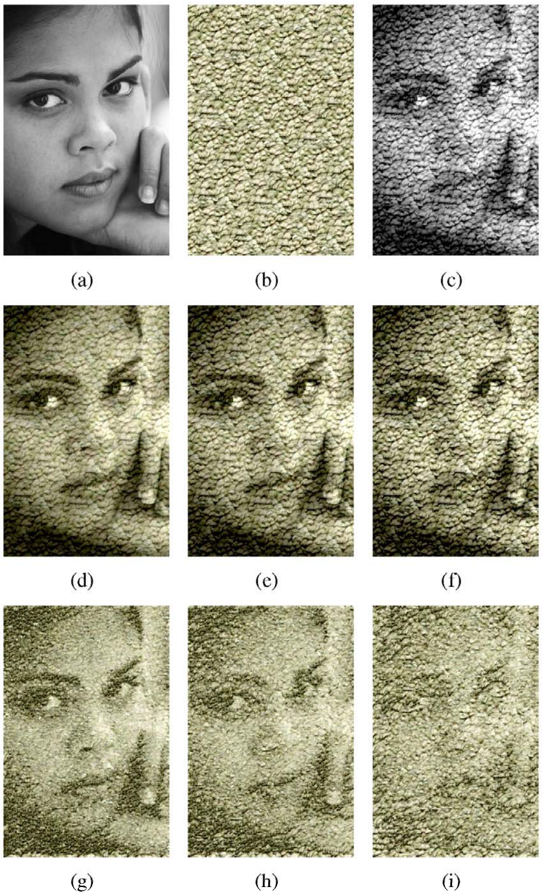


Figure 1. Texture transfer result. (a) : source, (b) : rug texture, (c), (d), (e), and (f) : texture transfer using the method developed in this paper (color components of (c) from (a) and (d),(e), and (f) from (b)). The parameter values used in $Fusion(Source, t \cdot Texture)$ are $t = 1, 0.6, 0.8, 1$, respectively. (e), (f) and (g) : results from the existing method with different parameters (Image Analogies http://mrl.nyu.edu/projects/image-analogies/tt.html) [1].

Section 5.

2 Image Fusion

In general, image fusion is defined as the process of combining multiple input images into a smaller number of images, usually a single one, which retrieves the relevant information from the inputs, to understand the scene better in various ways. The images to be combined are referred to as *input* or *source* images, and the resultant combined image as *fused* image.

Image fusion can be used to improve reliability by having redundant information from multiple sources, and improve capability by having complementary information that is not easy to achieve with only one source of information. Related research fields of image fusion include computer vision, image processing, and robotics.

Since an image is 2-D signal, the wavelet transform is a powerful tool for multiresolution analysis of an image. Recently, with the development of wavelet theory, people began to apply wavelet multiscale decomposition to image fusion techniques to achieve both spatial and frequency domain localization. In this section, a brief introduction to basic concepts of wavelet-based image fusion is provided.

2.1 Basic Concepts of Wavelet-based Image Fusion

In the following, the review of 2-D discrete wavelet transform (DWT) is provided for readers who are not familiar with it. If you are familiar with 2-D DWT and the pyramid structure arising from it, you can skip the following sub-section.

2.1.1 2-D DWT

Since an image is 2-D signal, I will mainly focus on the 2-D wavelet transforms. Figure 2(a) shows an one-level of the wavelet decomposition. After one level of decomposition, there will be four frequency bands, namely Low-Low (LL), Low-High (LH), High-Low (HL), and High-High (HH). The next level of decomposition is only applied to the LL band of the current stage. This procedure is recursively defined.

Figure 2(b) shows the pyramid structure of 2-D wavelet decomposition coefficients with 3 decomposition levels. An N-level decomposition results in $3N+1$ different frequency bands. Among these $3N+1$ bands, only one band to which only low-low filter is applied is defined to be the *approximation* of the image, and other $3N$ high frequency bands are defined to be the *details* of the image. Due to the energy compaction property of images, the approximation band contains overall characteristics of the image.

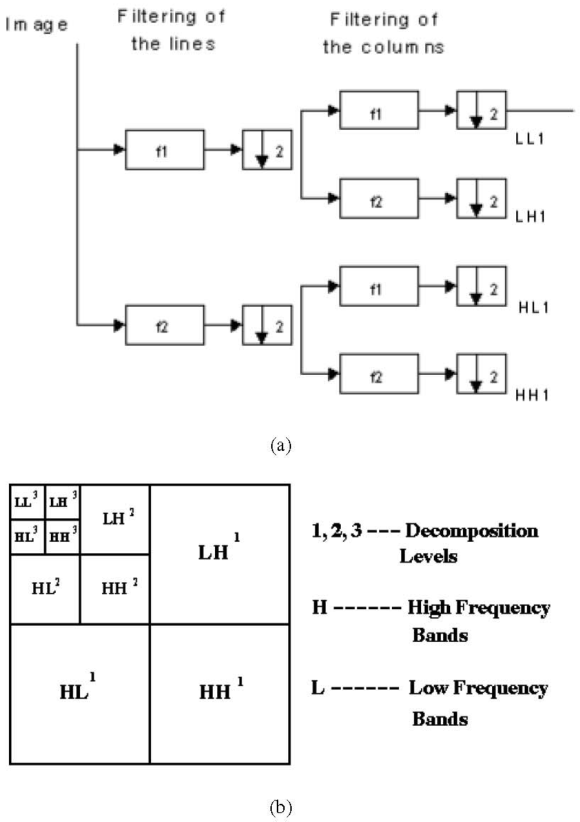


Figure 2. 2-D wavelet decomposition. (a) : one-level of wavelet decomposition, (b) : the pyramid structure of wavelet decomposition [5].

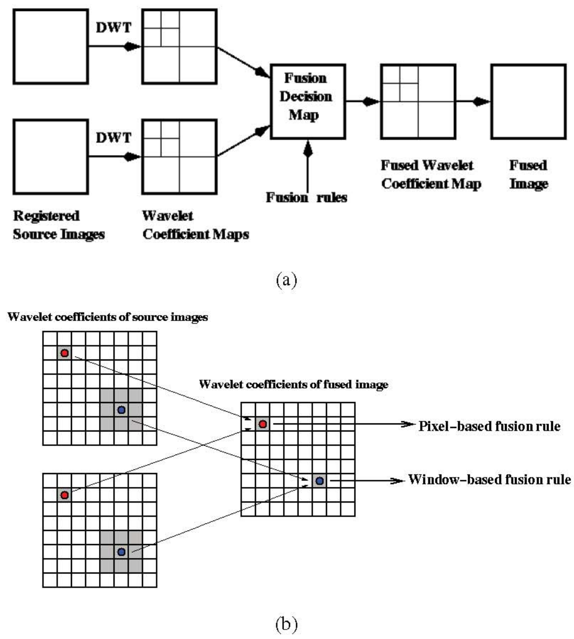


Figure 3. Image fusion scheme. (a) : Block diagram of a generic wavelet-based image fusion approach, (b) : Popular fusion decision rules [5].

2.1.2 Image Fusion Scheme

In this section, the generic approach of wavelet-based image fusion is introduced. The block diagram of the approach is shown in Figure 3(a). First, wavelet decomposition is performed on each of source images. Then, we use a fusion decision map. The input to the map is the set of wavelet coefficients of source images, and the fused wavelet coefficients are constructed from the input according to the fusion decision rules. Finally, the fused image is obtained by performing the inverse wavelet decomposition.

From the block diagram, it is obvious that the fusion rules play a very important role during the fusion process. There are many fusion rules developed and choosing one of them to use depends on which application the fusion process is applied to.

Figure 3(b) shows some popular fusion decision rules. Pixel-based fusion rule is very simple, but powerful. When constructing each wavelet coefficient for the fused image, we combine two coefficients of the sources located at the same position to produce the fused coefficient. There are some common ways to combine two coefficients such as taking average, maximum, or minimum of the two. We can extend this pixel-based rule to consider neighboring coefficients. This window-based fusion rule takes the advantage of the fact that neighboring pixels are highly correlated.

The fusion rule will be only pixel-based throughout this paper since it is straight-forward to design and implement, and yet very powerful. Deciding appropriate filters and decomposition levels to use highly depends on the application. This design decision will be revisited later when the fusion scheme is applied to texture transfer.

3 Texture Transfer Scheme

In this section, the image fusion technique introduced in Section 2 is applied to texture transfer. For texture transfer, we have two images to fuse: source image and texture image. Only luminance values are used during transfer and color components of images are unchanged. For the color components of the synthesized image, those of either the source image or the texture image can be used. In Figure 1(f), the color components of the texture image are used,

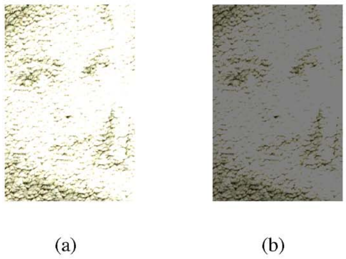


Figure 4. Averaging two images. (a) : sum of pixel values of two images, (b) : average of pixel values of two images.

while those of the source image are used in Figure 1(c).

To give some idea of the difficulty to perform texture transfer, one simple experiment has been done. Pixel values of two images at the same postion are averaged to construct the corresponding pixel value of the synthesized image. Figure 4 illustrates the result. It is difficult to recognize either the face or the texture. The sharpness of edges is not preserved well in both of images. In contrast, the results in Figure 1 successfully preserve characteristics of both source and texture.

The following sub-section explains the details of how image fusion scheme is applied to texture transfer.

3.1 Design Specification

Let $x^{(1)}_{i,j}$ and $x^{(2)}_{i,j}$ denote the source and texture images, respectively. First, the wavelet decomposition is applied to the images. The filter used for the decomposition is Cohen-Daubechies-Feauveau 9/7 (CDF 9/7), the FBI fingerprint standard. The reason why CDF 9/7 is chosen is that it is currently known to be one of the best filters for analyzing images. Let $c^{(1)}_{i,j}$ and $c^{(2)}_{i,j}$ denote the wavelet coefficients of $x^{(1)}_{i,j}$ and $x^{(2)}_{i,j}$. According to the image fusion scheme, $c^{(1)}_{i,j}$ and $c^{(2)}_{i,j}$ are combined to produce $c^{(3)}_{i,j}$, the wavelet coefficients of the synthesized image. The synthesized image, $y_{i,j}$, is the inverse wavelet decomposition of $c^{(3)}_{i,j}$.

Since the pixel-based fusion rule is used, the fused wavelet coefficient at the position (k_1, k_2) is $c^{(3)}_{k_1,k_2} = f(c^{(1)}_{k_1,k_2}, c^{(2)}_{k_1,k_2}, b_{k_1,k_2})$, where $b_{i,j}$ is a boolean variable defined as follows:

$$b_{i,j} = \begin{cases} 1 & \text{if position } (i,j) \text{ in the LL}^N \text{ band,} \\ 0 & \text{if } (i,j) \text{ in high frequency band.} \end{cases}$$

The following sub-section explains the fusion decision rule $f(c_1, c_2, b)$.

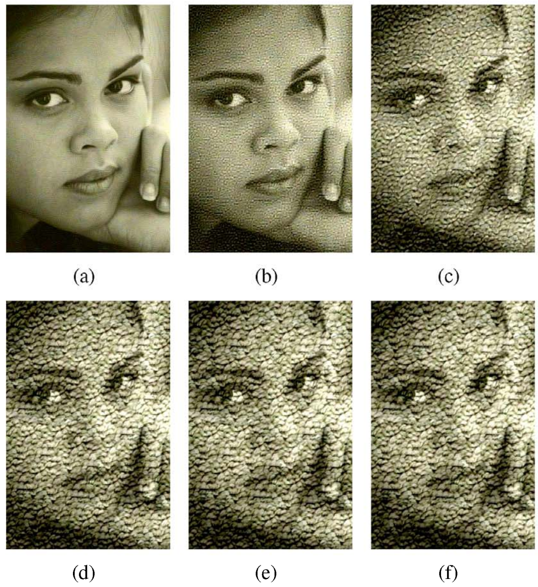


Figure 5. Different decomposition levels while the filter and fusion rule used remain unchanged. (a), (b), (c), (d), (e), and (f) : 1, 2, 3, 4, 5, and 10 decomposition levels, respectively.

3.1.1 Fusion Decision Rule

It is very important to come up with an appropriate fusion rule. Since evaluation on texture transfer result is very subjective, there is no perfect answer for fusion rule. There are two basic principles to consider. First, the synthesized image should look similar to the source overall. Especially when looking at the fused image at a far distance, the feeling of the image should be similar to that of the source. Second, when inspecting the synthesized image closely, it should show the pattern of texture vividly. The following fusion rule is derived from the two principles described above.

$$f(c_1, c_2, b) = \begin{cases} c_1 & \text{if b = 1,} \\ max(c_1, c_2) & \text{otherwise (i.e if b = 0).} \end{cases}$$

What this rule does is that for the approximation of the image, it takes the coefficients of the source image. In this way, the synthesized image will look like the source overall. For the details, the maximum of two coefficents is used. By taking the maximum, the fusion preserves local characteristics of both source and texture. The edges of source and texture still remain sharp after fusing.

3.1.2 Decompostion levels

How many decomposition levels should be used is discussed in this section. Figure 5 shows the results of using different decomposition levels while the filter and fusion rule remain unchanged. With one, two, or three levels, texture is not correctly transferred. As most of wavelet decomposition applications to images uses four or five levels, texture transfer also looks reasonable starting from four levels. From four to ten levels, the results look very similar even though there exist slight differences between results of four and five levels. Therefore, five levels are chosen to be used for the fusion scheme.

3.2 Space and Time Complexity

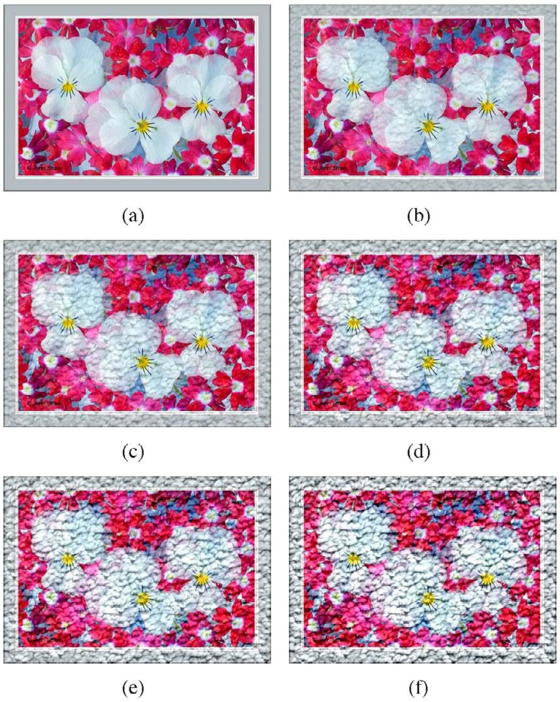
For the source and texture images of size $m \times n$, the wavelet-based texture transfer uses only $O(mn)$ addtional space. This is used for storing wavelet coefficients. The time complexity of the transfer is also $O(mn)$ since both wavelet decomposition and fusion of coefficients can be done in $O(mn)$. It is concluded that space and time required for texture transfer are linear in the size of images.

All experiments performed in the paper used a 2.8GHz Pentium D PC, and the system was implemented in MATLAB. Runtime of an experiment never exceeded one second.

4 Progressive Texture Transfer

It is often the case that users do not want to have too much of texture in the synthesized image so that texture overwhelms the entire image. In Figure 1(f), it is not easy to see some of details in the face. To resolve this problem, intensities of the texture image is multiplied by some constant less than one, and we still use the method described in Section 3. Let $Fusion(Source, t \cdot Texture)$ denotes this texture transfer where the effect of texture transfer can be reduced with $t < 1$. If t is close to 0, no texture is transferred. If t is close to 1, texture looks dominant in the synthesized image. Figure 1 and 6 show results of this progressive transfer. By letting t vary from 0 to 1, it is possisble to create a very artistic animation. In such an animation, the pattern of texture is progressively transferred. An interesting observation is that even though the very first and last frames look totally different, adjacent frames look almost the same. Therefore, visual changes in these animations look very smooth giving a pleasure of watching. Some of movie files for these animations are uploaded online[1].

[1] http://jwkim.com/academic/projects/texture_transfer



(a) (b) (c) (d) (e) (f)

Figure 6. Progressive Texture Transfer. (a) : Source. The rug texture in Figure 1(b) is used for texture. (b), (c), (d), (e), and (f) : $Fusion(Source, t \cdot Texture)$, where $t = 0.2, 0.4, 0.6, 0.8$, respectively.

5 Comparison to the Existing Methods

Image Analogies, the framework for processing images by example, is known to be one of state-of-the-art methods for texture transfer [1]. This method, introduced by Hertzmann et al., performs reasonable texturization as shown in Figure 1.

The figure contains three sub-figures from *Image Analogies*. Notice that all three images of *Image Analogies* have diminished the intensities of the face, and the rug texture is not easily recognizable in Figure 1(g) and 1(h). In Figure 1(i), the pattern of the rug texture is partially recognizable, but the face is not recognizable anymore. In contrast, Figure 1(f) is more contrasty, and the rug texture is easily recognizable. The feeling of the rug texture is better preserved in Figure 1(f).

Even though the rug texture is better expressed in the result of our method, it is not possible to conclude which one produces the better result since copying the texture exactly might not be the objective. Since there can be different artistic effects people are interested in, I will leave the preference to readers.

Image Analogies requires intensive computations that take up from several minutes to several hours [4]. In contrast, my method takes less than one second. This is a huge gain in speed. This speed gain enables user interactive texture transfer. User can tweak the parameter values easily while looking at the corresponding preview in real-time.

There exists a fast texture transfer method from previous literature [4]. However, this method is not as popular as *Image Analogies* since the method requires complicated user-defined metrics and well-chosen parameters, thereby harder to use.

One can argue that some of existing blending techniques can achieve the same goal. I performed some experiments using some of famous graphics editors. The existing graphics editors performed reasonable texture transfer in some local regions, but the pattern of texture was not well expressed in many regions. This is due to the fact that the low-frequency components and high-frequency components should be processed separately. Blending techniques are not well suited for the purpose of texture transfer.

Implementing the method introduced in this paper is very straight-forward. There are standard implementations of wavelet and inverse wavelet decomposition available in popular computer languages, and fusion of coefficients can be done by a couple of matrix operations.

For comparison purpose, more results are shown in Figure 7. The pattern of texture is transferred well in general. For future benchmark, high-quality version of images used in the paper are posted online[1].

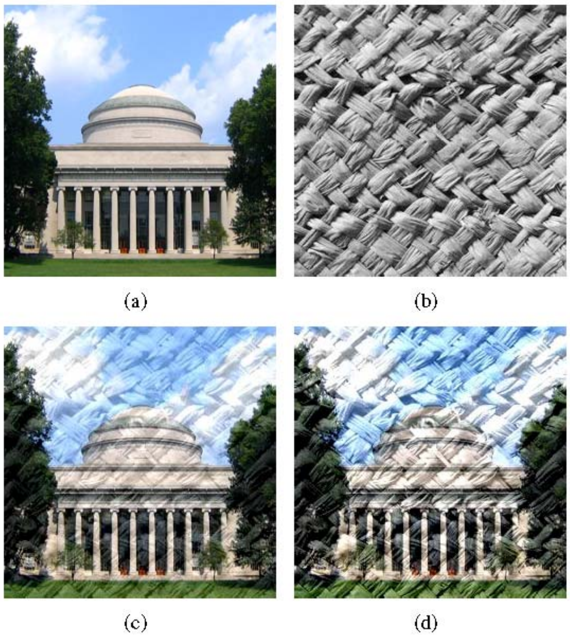


Figure 7. (a) : source, (b) : texture, (c) : $Fusion(Source, 0.4 \cdot Texture)$**, (d) :** $Fusion(Source, 0.8 \cdot Texture)$**.**

6 Discussion and Future Work

In this paper, I have described a texture transfer algorithm based on wavelet techniques. I have shown that the algorithm runs relatively fast, and texture is successfully transferred. There is still much work to be done. By using different fusion rules, it is possible to achieve different artistic effects. Trying window-based or region-based fusion rules would give very different results from the results of the current method. In addition, it is possible to speed up the current method by discarding small wavelet coefficients of texture images, and iterating over only non-zero coefficients that are left for fusion. Finally, performing texture synthesis in wavelet domain can be a very interesting direction of research. I am encouraged by my early results, and look forward to improving the method and finding more applications.

Acknowledgements

This paper is written for the final project of the wavelet signal processing class at MIT. I want to mention thanks to Professor Vivek Goyal for lecturing and guiding throughout the semester.

References

[1] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, D. Salesin, “Image Analogies,” ACM SIGGRAPH Conference Proceedings, 2001.

[2] A. Efros, T. Leung, “Texture Synthesis by Non-parametric Sampling,” ICCV, 1999.

[3] L. Wei, M. Levoy, “Fast Texture Synthesis using Tree-structured Vector Quantization,” ACM SIGGRAPH Conference Proceedings, 2000.

[4] M. Ashikhmin, “Fast Texture Transfer,” IEEE Computer Graphics and Applications, July/August 2003.

[5] R. Blum et al., “Investigations of Image Fusion,” `http://www.eecs.lehigh.edu/SPCRL/IF/image_fusion.htm` (current May 13, 2007).

[1] `http://jwkim.com/academic/projects/texture_transfer`