GENERATING A SEAMLESS TILING OF A NATURE IMAGE

Budi Hartanto
Informatic Engineering, University of Surabaya
Jl. Raya Kalirungkut Surabaya – 60292
Email: budi@ubaya.ac.id

ABSTRACT: Tiling is a process to create a repetitive – larger size – image. However if the picture on the left side of the original image is not continuous to the picture on its right side, the tiling image will have a virtual vertical line called seam. Meanwhile the virtual horizontal line will appear if the picture on the top side of the original image is not continuous to the picture on its bottom side. The research performed here tries to generate a seamless tiling image by finding the closest match of the fractional source image to the partially build resulted image. From the experiment, it can be shown that the quality of the resulted image are affected by the number of similar elements in the source image, the number of fractional images created from the source image, and the width of the overlap area. Source image with a lot of similar element, high number of fractional images, and wider size of the overlap area have been proven to yield a seamless tiling image.

Keywords: tiling, seamless, fractional image, overlap area.

INTRODUCTION

Large size image can be created by performing tiling of a small size image. Tiling is an effect generated by placing the same images side by side, horizontally and/or vertically. Unfortunately, if the picture on the left side of the image is not continuous to the picture on its right side then the tiling image will have an unpleasantly vertical seam. This vertical seam actually appears from the differences between the colors of the pictures on the left side of the image from its right side. The same phenomenon will occur if the picture on the bottom side of the image is not continuous to the picture on its top side. In this case, the generated image will have a horizontal seam.

In some applications, these vertical and horizontal seams may not irritate the users at all. However in some other applications such as in the game application or in the 3D object texture mapping, the tiling image is expected to appear as natural (seamless) as the source image. To get a seamless tiling image, one should provide a continuous source image, horizontally and vertically. Unfortunately, sometimes it is difficult to find source image that meet this requirement.

Most nature images such as grass, stone, cloud, etc are composed from similar elements that are dispersed to the entire image. Since the elements of the image is similar, human usually can not distinguish easily the modification made to any element of this images, as long as the modification does not producing any seams. What is meant by the modification in this case is the transformation of the image element. Therefore the image element can be translated, rotated, and even scaled in such a way and the resulted image can still be seen natural. The opposite condition occurs for the image without any similar elements. In this kind of image, any transformation to the image element will make the image becomes unnatural and even becomes weird.

Based on this concept, a seamless tiling image from the nature image can be created by placing side by side, the similar image element from the source image. The image elements taken from the source image are not necessarily taken consecutively. As long as the image elements have a high similarity to the partially build resulted image, the image element can be taken as a new part of the resulted image. The consideration of the similarity in this case is only applied to the border part of the image element. When the resulted image is built from left to right – top to bottom, the left border of the image element must be similar to the right side of the partial resulted image. Additionally the top border of the image element must also be similar to the bottom side of the partial resulted image.

TILING

According to Wikipedia.com, an online encyclopedia, tiling is a process of creating a resulted image that consist of a collection of other image that precisely cover the resulted image [1]. Sometimes people called tiling with other terms such as tessellation due to the same characteristic of objects repetition [2]. However, the object repetition in tessellation usually applies to an object element and there is not any space between each object element (see Figure 1). On the other hand, the tiling image usually has looser criteria. As long as the resulted

image is generated from the repetition of the other images, then it can be said that the resulted image is a tiling image.

Tiling image is usually used in the game development to create a patterned landscape or a repetitive background [3]. Applying tiling image for that purpose will save a lot of memory in the game application and make it possible to run the game in real time. The other usage of the tiling image can be found in the two or three dimensional object texture mapping [4] to increase the realism of the object.

A simple method that can be used to tile an image is to place the source image side by side to build the resulted image [5]. One example of the resulted tiling image can be seen in Figure 2. However, this process may yield seams on the resulted image if the pictures on the source image are not continuous on their borders. One simple solution is to mirror the source image on its vertical and horizontal axis before the tiling is performed (see Figure 3). Since the picture on the left side of the image becoming the same to the picture on the right side of the image (the same condition applies to the top and bottom image), the resulted tiling image can be guaranteed will be seamless.

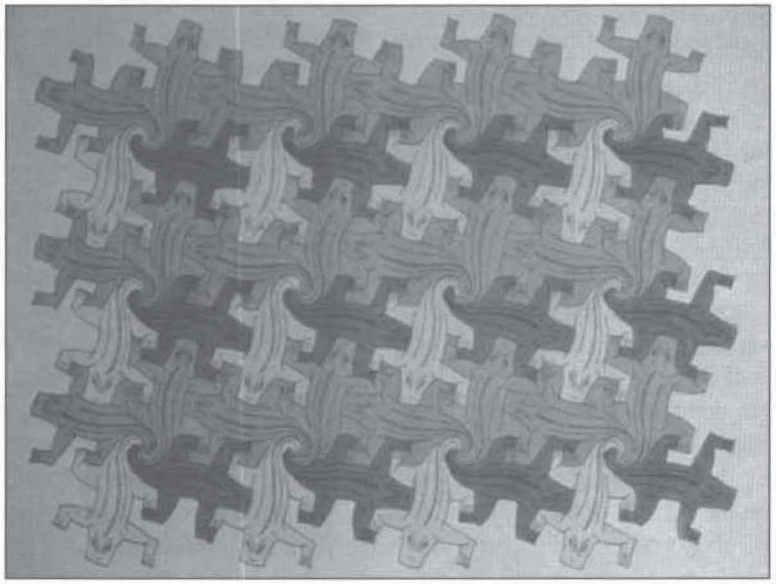


Figure 1.Tessellation image

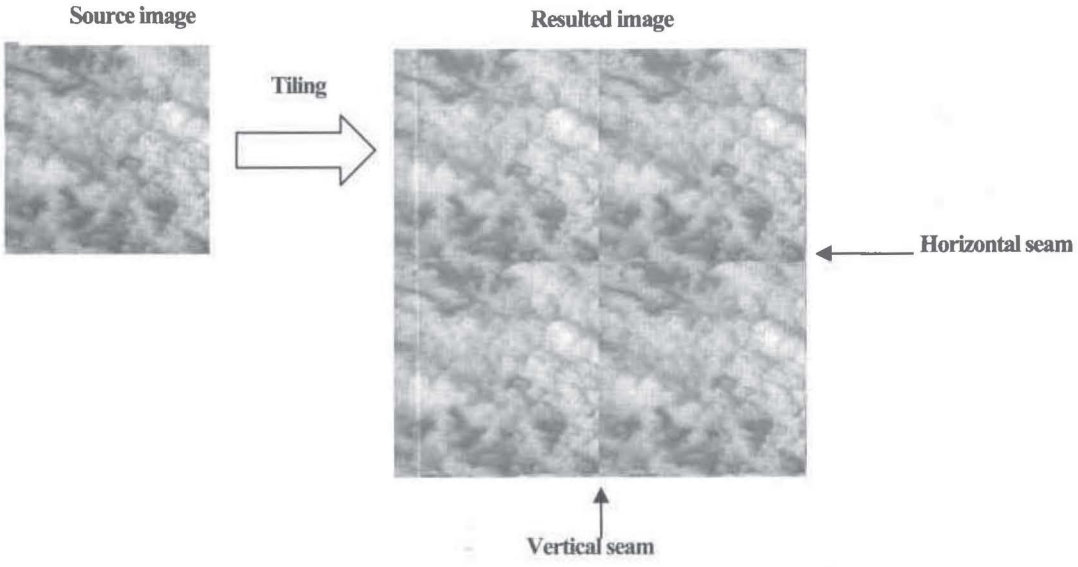


Figure 2. Tiling a source image

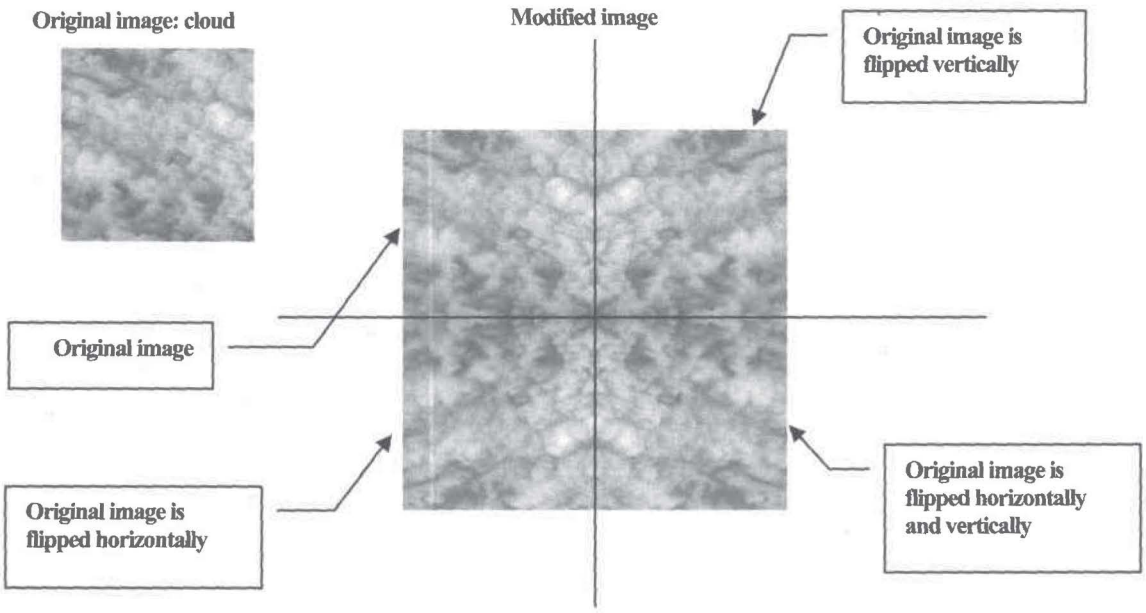


Figure 3. Modifying the original image before the tiling is performed

Unfortunately, performing tiling to the modified image sometimes yield a new image that does not have the same theme as the original image. For example, the cloud that is used as the source image in Figure 3, becoming an abstract image after the source image get mirrored and tiled (see Figure 4)

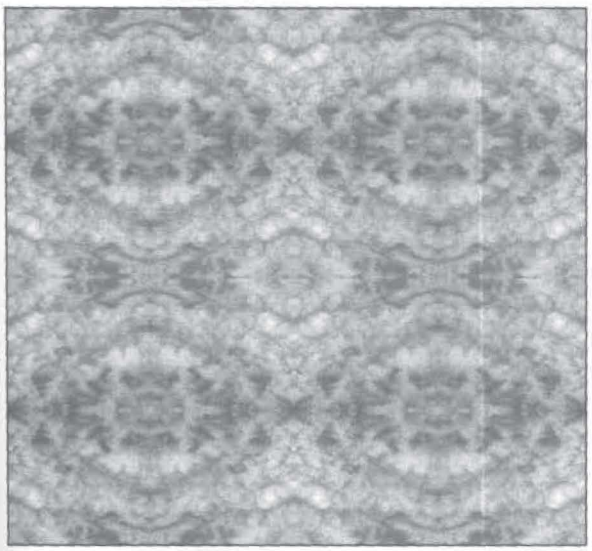


Figure 4. The resulted image after the original image is mirrored and tiled

SEAMLESS TILING IMAGE

In order to generate a seamless tiling image, it is necessary to check the border color of the source image that will be added to the resulted image. For example to add another source image (call it image A) to the right column of the resulted image (call it image B), it is necessary to check the colors similarity on the left part of the image A to the right part of the image B. If the colors are similar, then image A can be placed on the right side of image B. However, this process can not be performed if the whole part of the source image should be added to the resulted image.

Therefore the source image will be divided into fractions. The construction of the resulted image is started by selecting a random fraction of the source image to be placed in the top-left part of the resulted image (see Figure 5). The next fraction that will be placed in the resulted image is chosen from the most similar border of the fractional image to the partial resulted image. To disguise the seams that still possibly occur due to the unavailability of a fractional source image with the same color as the partial resulted image, the selected fractional source image can be placed overlap to the partial resulted image. The seams can be disguised by averaging the overlapped border color from both images. All of the process can be repeated until the desired size resulted image is built.

The algorithm to build the seamless tiling image can be described as:

1. Select a random fraction from the source image as a starting block in the resulted image. Put the selected random fraction on the top–left part of the resulted image.
2. Find another fraction from the source image with the smallest differences to the already existed block in the resulted image.
3. Put the selected fraction beside the existing block in the resulted image. Some part of the fraction can be placed overlap to the resulted image (see figure 6). The overlap area can be averaged to even more decrease the occurrence of the seams.

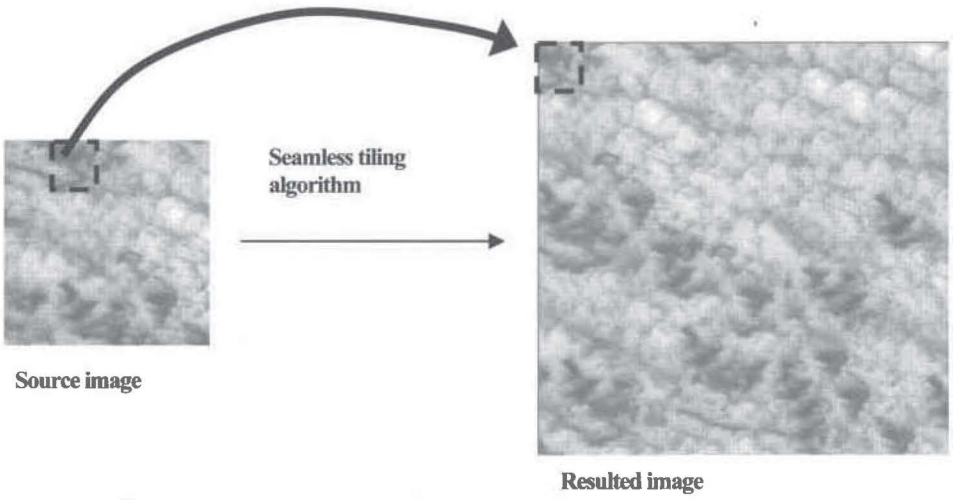


Figure 5. Building the resulted image by placing fractional source image

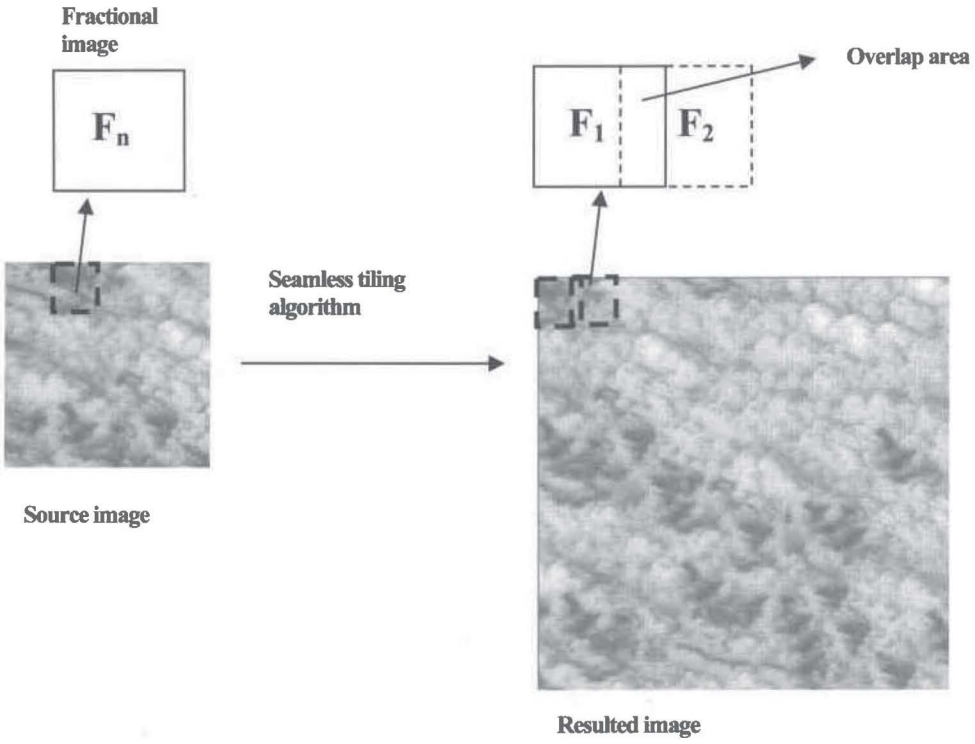


Figure 6. Decreasing seams by placing fractional image overlapped

The difference of the color on the overlap area can be calculated by:

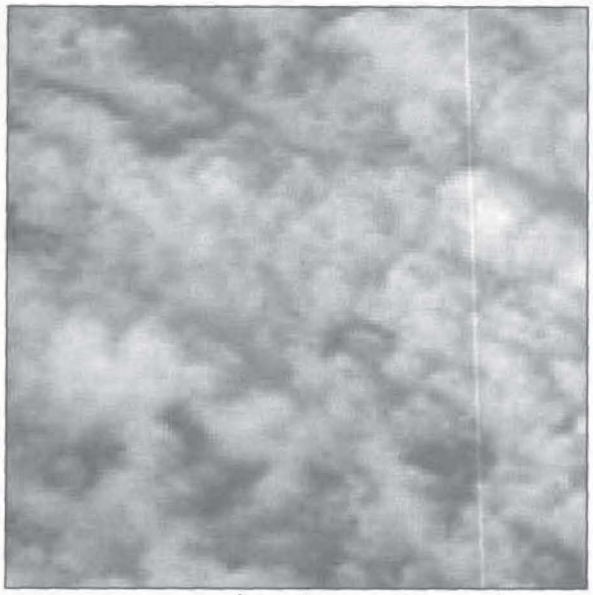
$$D(N_1, N_2) = \sum_{p \text{ in } N} \{(R_1(p)-R_2(p))^2 + (G_1(p)-G_2(p))^2 + (B_1(p)-B_2(p))^2\} \quad (1)$$

where:

N1 = the overlap area from the fractional source image
N2 = the overlap area from the partial resulted image
D = difference of color between the source and the result image
R = the red color component from a pixel P
G = the green color component from a pixel P
B = the blue color component from a pixel P

To find the minimum differences of colors, it is necessary to scan the entire possible fractions from the source image (see Figure 7). The scanning can be performed orderly from the upper-left corner of the

source image to the lower-right. In order to get all possible fractional images, the next fraction taken from the horizontal scanning is just one pixel away from the previous fraction. When the fraction is already at the most right of the source image, the next fraction will be taken from the most left of the source image with just one pixel below the previous row.



Source image

Figure 7. Scanning the source image

In order to decrease the occurrence of the seams at the overlap area, the color components from each picture can be averaged first before they are placed in the resulted image. The averaging process can be performed by:

$$C = (C_1^{ov} + C_2^{ov}) / 2 \qquad (2)$$

where:

C = the averaged color

C_n^{ov} = color of the n^{th} image at the overlap area

There are two conditions that are thought will make the resulted image becoming more seamless. These conditions are:

1. The number of the similar elements in the source image.
   If a source image has a lot of similar elements in it, the algorithm will have a lot of alternatives in selecting the fraction image to build the resulted image.
2. The number of the fractional images that can be generated from the source image.
   The combination of the image source element to the partial resulted image is performed at the size of the fractional image. Therefore it is necessary to guarantee that the numbers of fractional images are big enough to increase the possibility of getting the most similar fractional image to the partial resulted image. The number of fractional images from a source image can be found by:

$$N_f = (S_w - F_w + 1) \times (S_H - F_H + 1) \qquad (3)$$

where:

N_f = The number of fractional images

S_w = Width of the source image

S_H = Height of the source image

F_w = Width of the fractional image

F_H = Height of the fractional image

RESULT

Figure 8 shows several images that were tiled using the modified tiling algorithm. For comparison, the resulted images from the standard tiling algorithm were shown as well. From all the images shown in Figure 8, it can be seen that all the seams occur in the standard tiling images are not visible in the modified tiling images.

The dimension of all the resulted images in Figure 8 is 256x256, meanwhile the dimension of the source image is 128x128. The overlap area width to create the modified tiling images is 3 pixels, while the fractional image dimension is 20x20. The quality of all of the modified tiling images on Figure 8 can be considered good.

Figure 9 shows the effect of the usage of various fractional image dimensions. Figure 9a was created by 40x40 fractional image dimensions, while Figure 9b and 9c by 50x50 and 100x100 fractional image dimension. The quality of all images in the Figure 9 can be considered worse than the image in Figure 8 due to the occurrence of the seams on the image. As the fractional image dimension getting bigger, the number of fractional images that can be built is getting decreased. Therefore the possibility to get the most similar fractional image to the partial resulted image is getting decreased as well.

The number of fractional images generated from 128x128 source image for each fractional image dimension shown in Figure 8 and Figure 9 can be seen in Table 1. From the table, it can be seen that the number of fractional images with 40x40 image dimension are only 7.921. This number is far fewer than the number of fractional image with 20x20 image dimension. Since the number of fractional image is decreased, the possibility of having the similar fractional image to the partial resulted image is decreased as well. Therefore smaller size fractional image will yield a better resulted image than the bigger one.

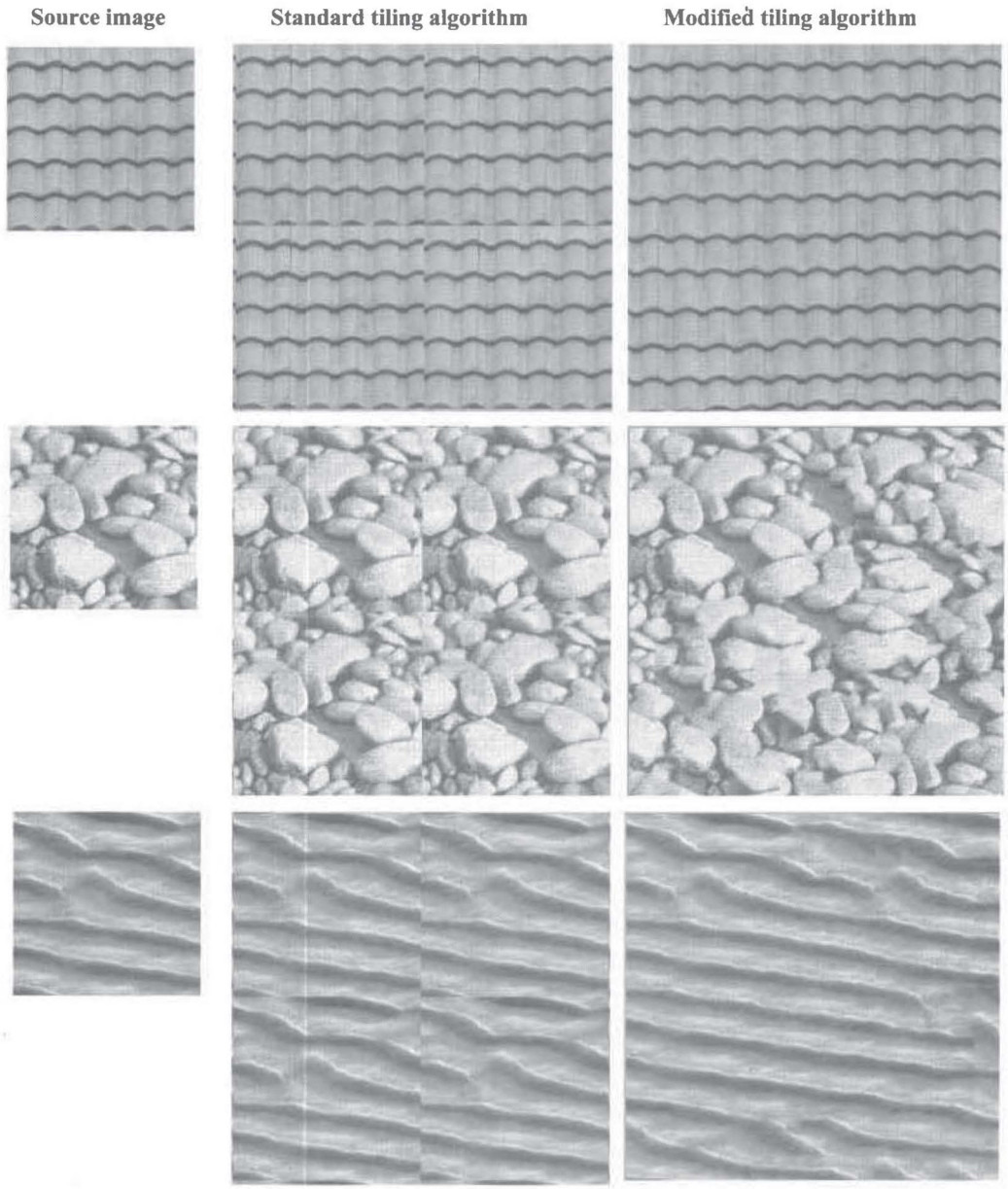


Figure 8. Images generated by standard and modified tiling algorithm

a. 40x40 fractional image dimension

b. 50x50 fractional image dimension

c. 100x100 fractional image dimension

Figure 9. Modified tiling algorithm with different fractional image dimension

Table 1. The number of fractional image generated from 128x128 source image

No	Fractional image dimension	The number of fractional image
1	20x20	11.881
2	40x40	7.921
3	50x50	6.241
4	100x100	841

The effect of the different overlap area width is shown in Figure 10. The three images in that figure are created using 10, 20, and 25 overlap area width. From the figure, it can be seen that wider overlap area will create fewer seams on the resulted image. The reason behind this phenomenon is that wider overlap area will create wider similar area between two fractional images. In addition, since the overlap area will be averaged, the resulted image with a bigger overlap area will be more seamless.

Figure 11 shows the modified tiling algorithm that is applied to a source image with just a few similar elements. Figure 11a shows a tiling of a rock

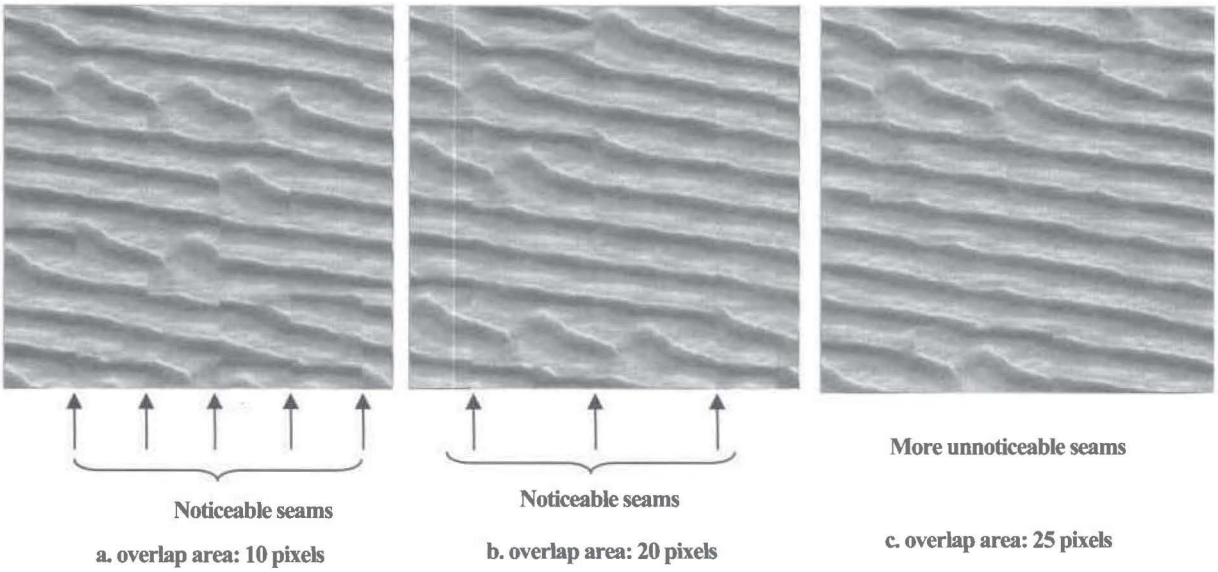


Figure 10. Modified tiling with different overlap area width

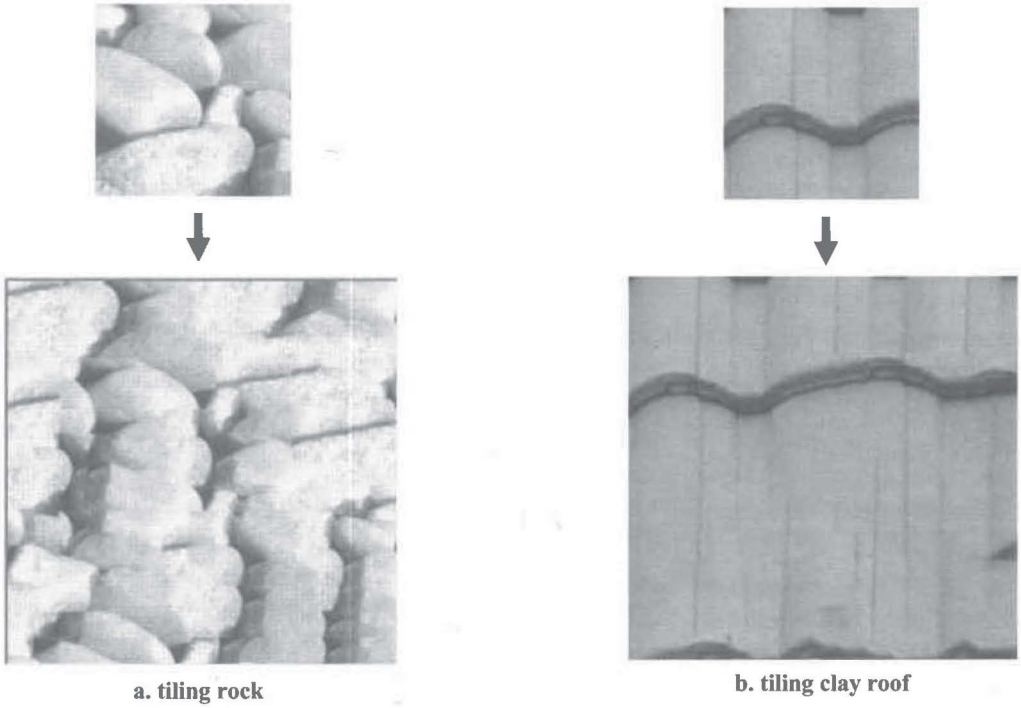


Figure 11. Modified tiling algorithm for image source with few similar elements

image. The resulted image can be said unexpected although it still appear natural and may be still acceptable. Meanwhile the resulted image in Figure 11b can be said odd and unacceptable. Therefore, to increase the naturalism of the resulted image, users should provide source image with a lot of similar elements on it.

CONCLUSIONS

From the experiment it can be concluded that the modified tiling algorithm can generate a good result when the number of similar element in the source image is large enough. If this condition can not be fulfilled the resulted image will become unexpected although some resulted image can still be considered natural. Increasing the number of fractional images is another key point to get a good resulted image. The number of fractional images will increase if the fractional image dimension is decreased. However decreasing the fractional image dimension will increase the time required to build the resulted image. This is the trade off. A good image dimension for a certain source image can be found by trial and error. The other parameter that influences the resulted image is the overlap area width. A wider overlap area will make the seams becoming more unnoticeable although for some source image the seams may not be able to be removed completely.

The research performed here has proved that the modified tiling algorithm can create a seamless tiling image, as long as some conditions are fulfilled. Applying the modified tiling algorithm for source image with a few repetitive elements may yield worse resulted image than the image generated by the original tiling algorithm. Further research can be made to improve the generated resulted image when the source image does not have enough similar elements.

REFERENCES

1. Wikipedia.com. *Tiling*. Taken from http://en.wikipedia.org/wiki/Tiling
2. Britton J. *Symmetry and Tessellations*. Dale Seymour Publications. 1999.
3. Estevao, M. *Tiling in DirectX*. Taken from, http://www.gamedev.net/reference/articles/article1.134.asp. 2000.
4. Watt. A. *3D Computer Graphics*. Addison Wesley. 1996
5. Hill, F.S. *Computer Graphics Using OpenGL*. Prentice Hall. 2001.

ISSN 1411-0105

JURNAL INFORMATIKA

Volume 8, Nomor 1 Mei 2007

Terakreditasi: SK Dirjen Dikti No. 56/DIKTI/Kep/2005

UNIVERSITAS KRISTEN PETRA - SURABAYA					
JURNAL INFORMATIKA	Vol. 8	No. 1	Hlm. 1-78	Surabaya Mei 2007	ISSN 1411-0105

ISSN 1411-0105

JURNAL INFORMATIKA

Volume 8, Nomor 1. Mei 2007

Alamat Sekretariat / Redaksi:
Pusat Penelitian Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236, Indonesia
Telp. (031) 8494830-31, psw. 3139-3147, Fax. (031) 8436418
E-mail : jurnal_informatika@petra.ac.id
http://puslit.petra.ac.id/journals/

Jurnal Informatika merupakan Jurnal Ilmiah untuk mengembangkan ilmu di bidang **Teknik dan Manajemen Informatika**

Jurnal Informatika diterbitkan oleh Jurusan Teknik Informatika dan Pusat Penelitian Universitas Kristen Petra. **Redaksi** mengundang para profesional dari dunia usaha, pendidikan dan peneliti untuk menulis mengenai perkembangan ilmu di bidang **Teknik dan Manajemen Informatika**.

Jurnal Informatika diterbitkan 2 (dua) kali dalam 1 tahun pada bulan **Mei** dan **Nopember**

ISSN 1411-0105

JURNAL INFORMATIKA

Volume 8, Nomor 1. Mei 2007

DAFTAR ISI