Wang Tiles And Its Applications

Institut für Graphik und Simulation (ISG)
Prof. Dr. Thomas Strothotte
Fakultät für Informatik (FIN)
Otto-von-Guericke-Universität Magdeburg

von

Maik Schulze
Computervisualistik
3. Fachsemester
171377
maik@oneder.de

Seminar:	Virtual Communities
Betreuer:	Timo Götzelmann
Semester:	WS 2005/2006
Tag der Abgabe:	28. 11. 2005

Abstract

The generation of good-looking large textures for arbitrary surfaces as well as the distribution of many objects within an interactive ecosystem are very important. The restrictions of graphics memory size limit the desired dimensions of information content. In order to create more and more realistic looking environments, computer graphics has to make use of all algorithmic capabilities. Wang tiles provide the possibility to create pseudo-chaotic and aperiodic content at low cost.

This paper presents the history of Wang tiles and approaches to utilize them in computer graphics. Two- and three-dimensional applications will be addressed as well as propositions about utilizations of Wang tiles in virtual communities.

Contents

1 Introduction

The demands of high quality computer graphics are growing. The faster the computers are, the better is the visualization of complex content. However, the complexity of computer graphics is limited to the size of the graphics memory and other parameters. In order to obtain high performance, it is important to make use of very efficient algorithms and limit the memory needs of the application. Unfortunately, the content for realistic looking and complex environments is very large. The geometry has to be simplified as well as the textures have to be downsized to assure real-time capability and other criteria.

If the size of a texture is too large to cover an object, the surface of the object can be tiled. The texture is repeatedly put on the object to fill the surface. This method is called periodic tiling. If only one texture is used several times to fill a surface, visual patterns will appear. Such repetitions are problematic as they produce an artificial appearance and mar the immersive effect of the application.

The contrary of periodic tiling is the aperiodic tiling. This type of tiling can assure that no visual repetitions appear. The look of objects with aperiodic textures is more pleasing as nature itself is aperiodic and chaotic to some extend. Unfortunately it is not known if a single tile of specific geometry can tile a plane aperiodically. A single rectangular tile certainly tiles a plane periodically. The smallest known set of aperiodic tilings consists of two non-rectangular shapes and follows the rules described by R. Penrose and R. Ammann. The Penrose tiling was quite theoretical until physicists discovered materials where the placement of molecules follows these rules in 1982.

However, computer graphics cannot utilize this type of aperiodic tiling because of the non-rectangular shapes. Another way of achieving aperiodic tiling is described by Wang tiling. A Wang tile is a square with colored edges. It is not allowed to rotate the tile. If a surface is to be filled with Wang tiles, the colors of incident edges have to match. A full set of Wang tiles consists of

$$K^2_{horizontal} \times K^2_{vertical}$$

tiles. *K* represents the number of horizontal and vertical colors used within the tiles. In the following sections, an example with two horizontal and two vertical colors is used. Thus, the full set of this configuration consists of 16 tiles. However, the size of the set can be reduced to eight and still avoid periodic patterns. Yet, strong aperiodicity cannot be assured.

The rectangular shape of Wang tiles allows the utilization in computer graphics. Besides using them for texturing, it is also possible to create object distributions. The following sections address the related work to Wang tiles and methods of content creation. Furthermore, the mapping of Wang tiles with hardware acceleration is presented. Additionally, applications of computer graphics that make use Wang tiles are shown and the utilization of Wang tiles in virtual communities is discussed.

2 Related Work

The theory of Wang tiles is a quite old idea. In 1961 Hao Wang, a Chinese born mathematician and philosopher, proposed an algorithm which decides if an finite, arbitrarily chosen set of tiles tiles a plane. In order to prove the correctness of his algorithm, he made the assumption that every set, that is able to tile a plane, would also periodically tile it [WWW1]. An example for this, is a repeatedly drawn wallpaper.

In 1966, Robert Berger proved that Wang's presumption was wrong. He presented a set of 20,426 different tiles which fill a plane aperiodically. The basis for his work was the use of rules of Penrose tiling and the Golden Rule [WWW2]. He presumed that smaller sets, also subsets of his own, would work as well. In 1986, Grünbaum and Shepherd presented rules of how to tile a plane aperiodically. Ten years later, Karel Culik published a set of only thirteen tiles that always tiles a plane aperiodically. In 1997, Jos Stam [STA97] used sixteen tiles to generate water textures and caustic maps. This work was the basis for Cohen et al. [COH03] who introduced algorithms for texture generation and object distributions. One year later Li-Yi Wei extended this work and presented solutions to issues concerning GPU's.

Wang tiles are also interesting to scientists of other disciplines. Aperiodic sets of Wang Tiles have been used for the simulation of DNA computers as they can simulate any Turing machine [COH03]. Furthermore Moore et al. examined in what way regions with boundary conditions can tile planes aperiodically [FUL05].

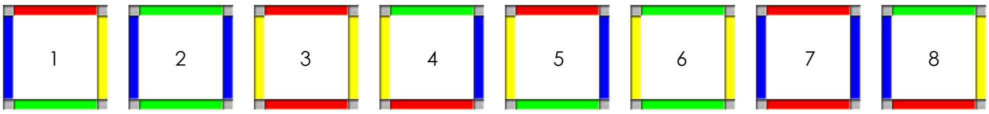

Figure 1: Eight Wang tiles that tile a plane.

3 Process Of Tiling

The following subsections present the process of the creation of a set of Wang Tiles. A distinction between Wang tiles as containers for texture content and object distribution is made. Arising problems concerning these uses of Wang tiles are addressed. Additionally, specific arising hardware problems are discussed as well as the correct mapping once a set of Wang tiles is obtained.

3.1 Textures

The next paragraphs address the different approaches to the generation of texture content for a set of Wang tiles. The tiles can be either crafted interactively by the user or with the help of computer programs. Both variants generate a non-repetitive set of Wang tiles which later can be applied to arbitrary objects.

Interactive Tile Design
One possibility of creating a set of Wang tiles is to make use of an interactive program. This application allows the placement of geometric primitives, that can be imported from any given drawing, within the set. The main advantage is that placing a primitive very close to border of a tile, so that it overlaps the boundaries, results in the automatic appearance of the primitive and its rest within all other tiles of the set having the same colored edge as well. This way, the opposing edges of the same color will match the placed primitive. Therefore, the user will not have to worry about the creation of matching incident edges. This solution comes with a graphical preview function, which immediately shows the changes of the set as well as the overall appearance of the placed primitive within a large scale texture. This immediate feedback eases the interactive tile design distinctively.

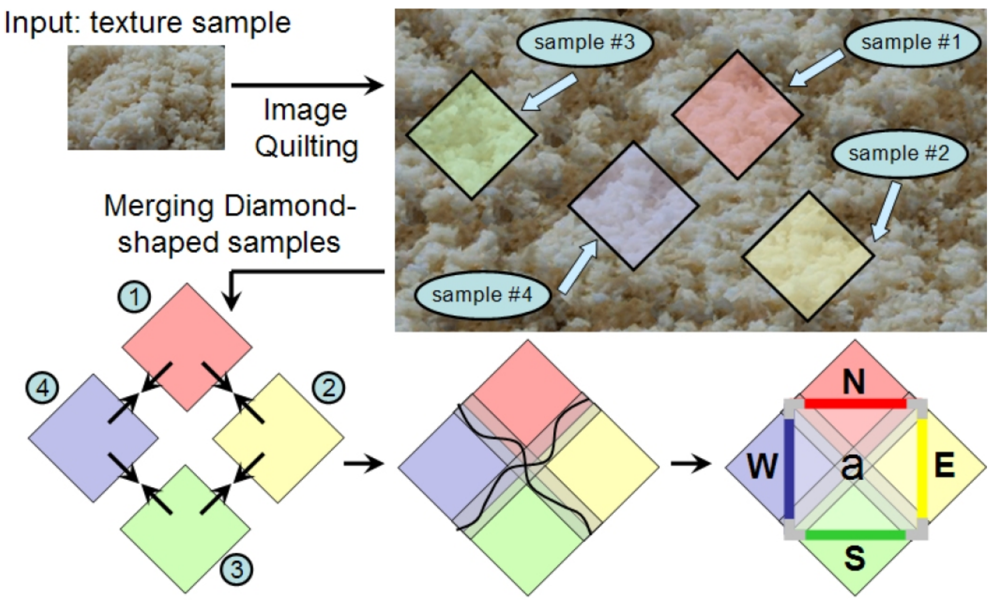

Figure 2: Showing the image quilting. Taken from Fu and Leung [FUL05].

Automatic Tile Design For Texture Synthesis
The manual creation of a set of Wang tiles can be very difficult for contents which are more complex. In order to avoid this problem, a set of Wang tiles can also be generated automatically. Since the content of the tiles can largely vary, a specific application which meets the needs of the solution has to used. One application of Wang tiles are textures which will be addressed more detailed later. Here, the focus lies on the process of tile design. The first solution is represented by Cohen et al. [COH03]. For the creation of a continuous texture generated by Wang tiles it is necessary to find tiles that visually match across the incident edges. This algorithm extracts sample diamonds, squares rotated by 45 degrees, of the original source images. Each of the four diamond-shaped sources represents a color of an edge of the future Wang tile (see first and second step in Fig. 2). If the creation a set of eight Wang tiles (see Fig. 1) is intended, four sample images each representing a color are required.

Sometimes, the available source image can be too small to extract multiple suitable samples. In order to avoid such problematic cases, Efros and Freeman [EFR01] developed a method to automatically generate a large texture out of one source image. Small sample squares are randomly chosen out of the original image and subsequently added to the created texture. This texture would contain unwanted visible artifacts since it is unlikely that the randomly chosen samples will fit together properly. Thus, it is necessary to balance the color differences along the cutting paths once a new sample should be added to the existing texture. Only if these optimizations were able to degenerate the threshold of the color differences between the regions, the sample is added to the texture. This guarantees the quality of the output texture.

Making use of this method, each tile is now created by combining four diamonds to one larger diamond (see step four and five on Fig. 2). This requires more optimizations along the cutting paths between the four samples. Then a square is constructed out of it with its center where the four different diamonds (colors) meet. Since optimizations were applied no artifacts are any more visible along the cutting path of the single diamonds. The resulting square represents a Wang tile of the set (see step five on Fig. 2). The edges have colors corresponding to the four different diamonds it was made of. Further Wang tiles of the set are generated with the same image samples. Each tile consists of four half diamond samples. The other halves are used for equally colored opposing edges (see the unselected halves in step five on Fig. 2). This guarantees that the entire set of Wang tiles will fit together properly.

Unfortunately some artifacts and distortions may still remain. Cohen et al. [COH03] state that firstly the use of too few sample patches of the source texture can lead to such unwanted effects. Secondly, the introduced algorithm by Efros and Freeman cannot completely prevent the creation of artifacts. By increasing the number of the tiles within the set of Wang tiles, the first issue can be overcome. Yet, it increases the cost of the computation process and the memory need. Another way of reducing the number of artifacts without enlarging the set of Wang tiles, is to compute the occurring overall quilting error, the sum of the color errors along the cutting paths. If the error is too high, four new sample patches are chosen and another set of Wang tiles is created. This evaluation can be repeated until the overall quilting error is low enough.

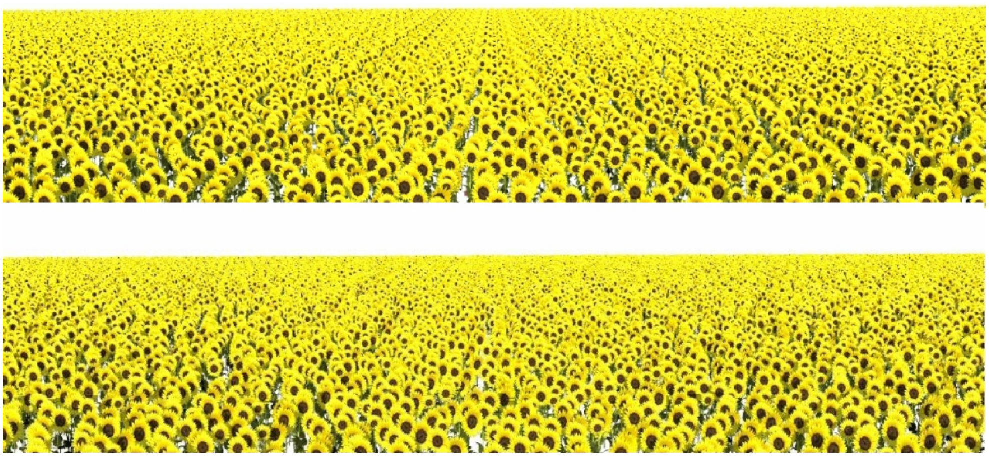

Figure 3: Repetitive patterns in the upper image are visible if only one tile is used. Eight tiles were used in the lower image. Taken from Cohen et al. [COH03].

3.2 Object Distributions

Instead of using Wang tiles for generating large textures, they also be used for geometric objects. The following paragraphs address methods for the automatic tile design as well as the Poisson disc distributions and Lloyd's method. The objective is the finding of a uniformly object distribution across the set of Wang tiles.

Automatic Tile Design
Nature appears aperiodic and chaotic, therefore Wang tiles are ideal for generating large complex ecosystems. Fig. 3 shows the difference between periodic and aperiodic plant placement within a huge field. For such environment thousands of objects have to be distributed. Instead of completely modeling the entire plant field, instances of the objects can be created. Nonetheless, the memory needs for large distributions can be very high. Instead of saving the position of each flower in the field, only the position for a few flowers within a Wang tile have to be saved. The full population can be generated analogously to the creation of textures by filling the plane with the appropriate tiles. The only difference is that the Wang tiles do not contain image information but geometric information. Naturally, by randomly distributing flowers on a plane aperiodicity can be achieved as well. The problem is that certain properties, like minimum distance to the next flower, cannot be considered. Therefore, methods that deal with object distribution will be presented.

Poisson disc distributions
The goal is to find a set of Wang tiles with appropriate point positions, representing the geometric position of an object. When combining the tiles to fill a plane, certain properties have be to regarded also beyond a tile's edge. In the prior example, the distance of a flower to its closest possible neighbor can be described as a circle with its center in the position of the flower. According to Cohen et al. [COH03], dart throwing is usually used to generate Poisson disc distributions. Random positions are sequentially generated and added to the point set if they do not interfere with existing positions in the point set. As several Wang tiles are used, it is important that tile edges are regarded. If a dart hits close to a tile's edge, it is possible that the equally colored opposing edge already claims that area due to existing points close to the boundary of the tile. Thus, the new point position would interfere with an existing one. In order to avoid this, it is necessary to immediately add the surrounding disc along all tiles as soon as a point is added to the set. Unfortunately this leads to less points added near the borders [COH03]. A visible distortion in the overall tilings can be noticed.

Lloyd's method
Retrieving a better result can be achieved by using the Lloyd's method. This variant has the advantage that points can interact with each other. Firstly, a Poisson distribution point set is generated for each tile independently. Then, one tile of the set is chosen and surrounded by eight matching tiles from the set. Across the boundaries of these nine tiles, the euclidean distances between the points are equalized. Further information on this variant can be found in Hiller et al. [HIL01].

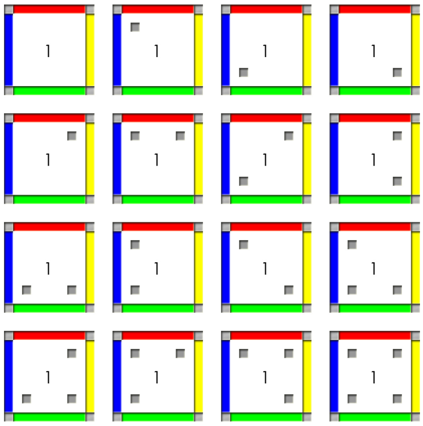

Figure 4: Showing one tile with all possible combinations of two density levels.

3.3 Introducing Inhomogeneity

The presented methods generate quite homogeneous textures or distributions. Yet, there are cases, where the content of the set of Wang tiles requires inhomogeneous properties. An example is the creation of a scratched metal texture. Naturally image quilting is used to obtain aperiodicity along a plane filled by a set of Wang tiles. However, adding local attributes like rusty spots is not possible. If rust is added to the source image, containing the scratched metal texture, it will be placed uniformly on each tile, depending on the affected tile boundaries to ensure matching edges. This would lead to unwanted repetitive patterns. Local changes in the texture content can be accomplished by using two source images for the texture synthesis. These images will have to potentially fit together, so that the quilting algorithm mentioned earlier will not produce visible artifacts. In this example, one source image contains the scratched metal surface and the other one contains the rusty spots.

It is necessary to enlarge the set of Wang tiles (see Fig. 4. Doing so, it is possible to create different density levels [COH03]. The scratched metal can be regarded as a high density level and the rusty spots as a low density level. The main disadvantage is the increased time exposure for the generation of the Wang tiles set. Additionally the memory consumption is increased, due to the larger size of the set. Object distributions with different density levels can be produced likewise.

3.4 Layered Depth Image Tiles

The work of Cohen et al. [COH03] presents layered depth image tiles utilizing Wang tiles. Instead of rendering the geometry of objects at runtime, the rendered output is precomputed. The geometry of objects is converted to two-dimensional image samples. In order to allow an object to rotate, several images have to be precomputed. Furthermore, additional samples are required to support level-of-detail. This allows to reduce the memory needs of distant objects and draw them less detailed. All these precomputed samples are saved in a hierarchy.

Cohen et al. [COH03] rendered a plant field with thousands of flowers. Wang tiles are used to place the flowers aperiodically across the plane. This approach allows the creation of complex natural environments at interactive speeds.

Unfortunately, many samples are needed to appropriately cover every viewing angle to an object. This leads to an increment of the memory needs and slows down the application. As the geometry of objects is not present anymore, no more computations can be applied. Thus, layered depth image tiles are a compromise between performance and quality.

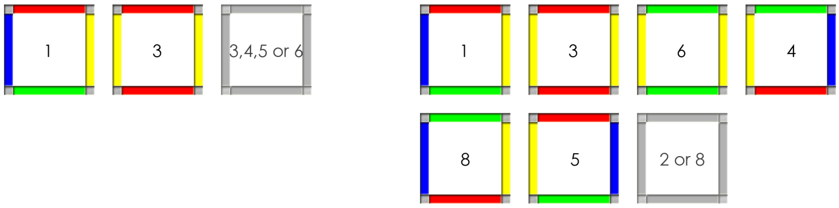

Figure 5: Showing two steps of the tiling process.

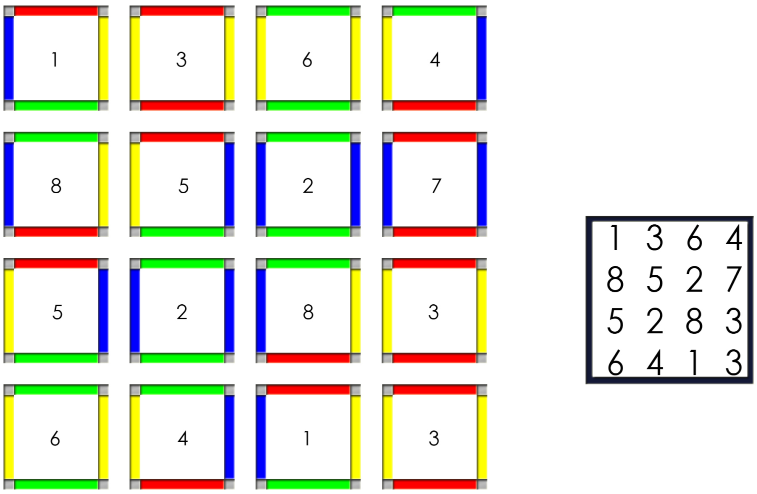

Figure 6: The final result of a 4x4 tiling plane on the left. The right matrix represents the index map.

3.5 Mapping Of The Tiles

The computation of a set of Wang tiles consumes the greatest share of time. It can take a few seconds up to several minutes, depending on the complexity of the source textures or object distribution and the utilized algorithms. Fortunately, they all have in common that the later memory needs are acceptable, especially if a single large texture would have been the alternative. The combination of single Wang tiles to fill a plane with textures or popularize it with a certain object distribution is capable of real-time rendering. Fig. 5 and 6 show how a plane is sequentially filled with Wang tiles. The upper left tile is randomly chosen and then the plane is filled row by row. The next tile of course has to have the appropriate opposing colored horizontal edge to match the conditions. If more than one Wang Tile is possible to fill the current gap, a random selection is made to ensure the aperiodicity. After having filled the first row, it will be advanced to the second. Here the selections are more limited, but it is still possible to choose between several tiles to fill a gap. Two edges of the tile have to match, namely the eastern and northern edge. In a set of eight Wang tiles it can be chosen between two. To assure the consistence of the texture or object distribution, such an allocation of Wang tiles can be saved (see Fig. 6). For example, the allocation can be saved in a two dimensional array of numbers, indicating the placement of the Wang tiles. If the rendering of such a scene is intended, the texture coordinates of the visual object parts and the correct texture tiles have to be determined. Naturally this also applies to object distributions.

Nonetheless, several problematic issues arise when utilizing Wang tiles and specifically making use of hardware acceleration. The work of Li-Yi Wei [WEI04] can be regarded as an extension of Cohen et al. [COH03], which only attached importance to the creation of a set of Wang tiles. Wei presents algorithms for assembling these tiles in order to create large virtual textures on-the-fly.

After having retrieved a set of Wang tiles, the tiles are packed into a single texture map. When a texture request has been made, the relative offset within the input tile is determined and the correct texel can be retrieved. The advantages of this variant are the capability of using the texture memory of the graphics hardware and the support of filtering and mip mapping. Additionally the entire algorithm can be implemented for the GPU and used like the traditional interface [WEI04]. Yet, guaranteeing random access for the texture tiles requires more tiles with redundant information. As mentioned before, it is possible to compute an index map and store it for the rendering process. However, even such an index map can be very large and in any case the consistency can be assured by utilizing a hash table. Unfortunately, the use of a hash table increases time consumption of the computation. Therefore, Wei proposes the utilization of the faster hash algorithm MD5. Another problem is bilinear and anisotropic filtering. If a texel is close to the border of a tile, the adjacent tiles are required to be touched as well, so that a correct output can be computed. Since the entire set of Wang tiles is packed into a single texture map directly on graphics hardware, it is possible to receive the convenient texel information.

His benchmarks have shown that Wang tiles can be implemented in an efficient way. Compared to the traditional textures, the approach of Wang tiles is slower. Nevertheless, with increasing texture sizes stressing the graphics memory, Wang tiles can be the favored solution and surmount the performance of traditional texture mapping.

4 The Uses Of Wang Tiles In 3D Applications

Chapter 3 described how to create a set of Wang tiles and mapping them correctly. A few examples were given, to help understanding the previous section. They gave an insight of the various applications of Wang tiles. The following paragraphs discusses the uses of Wang tile in three-dimensional applications.

As computers become more and more powerful in visualizing content, the demands for visually attractive solutions have largely grown. Natural environments are often presented in computer games and other applications. As nature appears chaotic and non-periodic, the attraction can suffer when repetitive patterns are perceived. Wang tiles can create pseudo-chaos with little needs of memory and computation.

The presented algorithm for object distribution however, is limited to object distance. The work of Andrew A. Cove [COV05] extends this work, so the object distribution can be computed in respect to further parameters like the plant's movement caused by wind. He presents methods that prevent plant geometry to overlap each other at strong wind levels without thinning the density of the interactive ecosystem.

Furthermore, a paper by Stam [STA97] presented the use of Wang tiles to generate caustic maps and water surfaces. It was possible to create quite beautiful effects without having unwanted periodicity at low costs. Such details can deepen the immersive effect of an application. Many other utilizations can be imagined using Wang tiles as containers for texture or object distribution information.

The next section addresses additional conceivable applications of Wang tiles in computer graphics.

5 Discussion And Future Work

The presented algorithms have shown how to generate large virtual textures and object distributions. Scientists were able to formulate algorithms for the automation of the image quilting. Thus, computers can autonomously create images without visual artifacts out of small samples. The automatic texture synthesis could be automated as well, so the users will not have to worry about the time taking processes of creating matching texture tiles. The work of Wei [WEI04] has shown that it is possible to make use of hardware acceleration and create attractive real-time environments with Wang tiles.

Nonetheless, it seems that Wang tiles are surprisingly unused in computer graphics. Even though, it is almost ten years ago, that Karel Culik found a set of Wang tiles with only thirteen tiles, the most important research concerning computer graphics, took place in the last three years. One reason might be the much smaller texture sizes used in older computer graphics software. Additionally, programmable GPU's had not widely spread and prevented the extensive use of Wang tiles in real-time environments. New hardware however, will offer more capabilities of utilizing Wang tiles.

Textures are a convenient application of Wang tiles. Memory needs of textures are always important and can substantially influence the rendering process and the overall performance on a computer, especially when several storage-consuming applications are being executed at the same time. In web applications another important factor is the bandwidth consumption. Wang tiles have the potential to create attractive visualizations with minimal needs. Instead of transferring large images over a network, it is possible to transfer only the set of Wang tiles and then create the final graphics on-the-fly on the client's computer.

If a user wants to create tiled background graphics for a website, visual artifacts will be the consequence. Enlarging the intended background image is also not a eligible solution, since bandwidth-consumption will increase and it is not possible to determine every user's screen resolution. Programs could make use of the presented algorithms for interactive and automatic tiling and present suitable results user-friendly. To what extent technologies such as Java and Macromedia Flash are capable of utilizing Wang tiles has to be determined.

Virtual communities are computer-mediated environments where people can meet and communicate over various ways. Often the shared resource, the knowledge which is distributed inside that environment, is the basis for valuable communication and lively longterm activity. The goal is to provide an environment where people are able to share ideas and knowledge and simultaneously work on certain objects. It should be possible to modify texts, geometric models and many other kinds of information. The visual appearance is a key feature and is able to influence the user's acceptance to last longer in such environments. Additionally, surfaces and well designed models are carriers of information.

Today's virtual reality environments often appear scanty and deserted. It is not possible to utilize huge textures and specifically model each object and its distribution within such a scene, due to limited graphics memory. The limited bandwidth, which does not allow extensive network traffic, narrows the possibility even more.

Wang tiles have the possibility to provide lively features and realistic details at low costs. The presented algorithms show opportunities to add very complex texture information without stressing the bandwidth of a virtual community. These textures can be applied to any kind of model and easily modified, as the texture information is independent of the geometric model.

Also volumetric fog and dust could be implemented. This has the advantage that without explicitly modeling these objects, they can enliven the environment and therefore strengthen the immersive effect of the virtual community. Clouds, dust and smoke could be generated with Wang cubes which carry information of density and randomly be distributed to a scene. They could also be used to represent further information. Imagine a less frequented room becoming more and more dirty with the time. Huge interactive ecosystems utilized by biologists to learn about the plant distribution at specific spots in nature and the frequency of a certain plant in a wood could be implemented as well. Traditional methods forbid such complex scenes, as the modeling effort and memory needs would be too intense. These were just a few examples of what Wang tiles are capable of to improve virtual communities. In what way it is possible to realize these propositions has to be discussed.

6 Bibliography

References

[COH03] Cohen, M. F., Shade, J., Hiller, S., and Deussen, O.: Wang Tiles for Image and Texture Generation. In ACM Transactions on Graphics 22, 3 (2003), pp. 287-294.

[COV05] Cove, A. : Dynamic Texturing of Botanical Environments. Carnegie Mellon University (2005).

[DEU02] Deussen, O., Colditz, C., Stamminger, M., and Drettakis, G.: Interactive visualization of complex plant ecosystems. In Proceedings of the conference on Visualization (2002), pp. 219-226.

[EFR01] Efros, A., and Freeman, W.: Image quilting for texture synthesis. In Proceedings of ACM SIGGRAPH (2001), pp. 341-346.

[FUL05] Fu,C.-W., and Leung, M.-K.: Texture Tiling on Arbitrary Topological Surfaces using Wang Tiles. In Proceedings of Eurographics Symposium on Rendering 2005 (2005), pp. 99-104.

[HIL01] Hiller, S., Deussen, O., and Keller, A.: Tiled blue noise samples. In Proceedings of Vision Modelling Visualization (2001), pp. 265-272.

[WEI04] Wei, L.-Y.: Tile-Based Texture Mapping on Graphics Hardware. In Proceedings of ACM SIGGRAPH/Eurographics conference on Graphics hardware (2004), pp. 55-63.

[SOM05] Somol, P., and Haindl, M.: Novel Path Search Algorithm for Image Stitching and Advanced Texture Tiling. In Proceedings of WSCG conference (2005), pp. 155-162.

[STA97] Stam, J.: Aperiodic Texture Mapping. ERCIM Research Report R046 from VTT(1997).

[WWW1] Wang tile - Wikipedia, the free encyclopedia; on the internet: http://en.wikipedia.org/wiki/Wang_tile, 8.11.05.

[WWW2] Robert Berger - Wikipedia, the free encyclopedia; on the internet: http://en.wikipedia.org/wiki/Robert_Berger, 8.11.05.