# The Promise and The Perils of Near-regular Texture*

Yanxi Liu and Yanghai Tsin
The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave. Pittsburgh, PA 15213
{yanxi,ytsin}@cs.cmu.edu

**Abstract**

*In this work, we demonstrate the promise and perils of texture analysis and texture synthesis applied to near-regular patterns. We propose a novel view of texture as statistical departures from regular patterns. We shall show that a true understanding of near-regular texture structures based on their translation symmetries can enhance existing methods of texture synthesis. Our texture synthesis result shows the promise of faithfully preserving the regularity as well as the randomness presented in a texture sample.*

## 1 Motivation

Near-regular textures are common in our daily life. They can be observed in man-made (machine-made) environments ranging from buildings to fabrics, as well as in nature [7, 2, 20, 26, 9]. Humans have an innate ability to perceive and take advantage of symmetry [12] in everyday life, but it is not obvious how to automate this powerful insight. [18] shows, in particular, that regularity plays an important role in human texture perception.

Mathematically speaking, *regular* texture refers to those patterns that present some kind of translation, rotation, or reflection *symmetry* [17, 4, 8]. *Periodic patterns* are referring to those images that present non-trivial translational symmetry. When studying periodic patterns, a useful fact from mathematics is the answer to Hilbert's 18th problem: there is only a finite number of symmetry groups for all possible periodic patterns in dimension $n$ [1]. In computer vision and graphics, the application of classic mathematics to near-regular pattern analysis has yet to be fully explored. Only recently, symmetry group classification algorithms have been developed for periodic patterns under Euclidean [15] and affine transformations [16], where the basic tile shape and size of periodic patterns are utilized extensively. One interesting recent work in computer graphics [11] is to find Escher-like tilings from a given single closed planar figure.

*This work is partially funded by an NSF research grant # IIS-0099597.

In the real world, however, rarely anything is strictly regular. Our research interest is to combine the mathematical theory of regular patterns with statistical modeling of data in texture analysis and synthesis.

Existing work on texture synthesis has achieved impressive results for a variety of textures e.g. [6, 5, 13, 10, 24, 25, 27]. While the evaluation on most synthesized results is hard to quantify, when the texture sample is near-regular, the structural regularity of the pattern becomes an objective measurement. However, we have observed from existing work on texture synthesis that when the original texture sample is near-regular, the regularity is usually not preserved in the synthesized texture. To the best of our knowledge, we have not yet seen an existing texture synthesis algorithm that preserves the regularity in an input brick wall sample (Figure 1).

This situation motivates us to first apply texture analysis for texture synthesis, such that the recovery of near-regular texture can be achieved with higher fidelity. Figure 2 demonstrates two sample results from our texture analysis/synthesis algorithm in comparison with the texture synthesis results in Figure 1 and results from naive direct tiling.

## 2 Our Method for Texture Synthesis

The essential element in our method is to acknowledge the regularity or periodicity in a near-regular texture by locating the generating "tile" explicitly. In [15], we have formulated the problem of periodic pattern perception based on the theory of wallpaper groups. A 2D *periodic* pattern has the following property: there exists a finite region bounded by two linearly independent translations, which when acted upon by the translation generators of its symmetry group produces simultaneously a *covering* (no gaps) and a *packing* (no overlaps) of the original image [19, 8]. We call the smallest such bounded region a **tile** of the pattern. For a given periodic pattern its tiles are unique in shape/size/orientation but not unique in location or content (i.e. its pixel intensity/color values). In order to find such tiles we developed an algorithm [15], called *region of dominance*, for locating the underlying lattice of a given pattern. Any offset of the lattice on a pattern carves the pattern into a set of similar tiles, any one of them can potentially tile the whole 2D plane.

The key insight to our method is to treat a set of tiles carved by the detected lattice as multiple samples of the same tile. This gives us the promise of capturing statistical density/color variations from the input data, which can be used to give the generated texture more natural appearance, while reproducing its regularity.

**Algorithm:**

*Step 1 (analysis)*: Given a sample texture pattern, first determine its translational symmetry and its underlying lattice. Find all the *minimum tiles* $t_i$ carved by the lattice structure (no gaps and no overlapping). For each $t_i$ construct a corresponding *maximum tile* $T_i$ by enclosing each rhombic shaped tile with a rectangular one (Figure 3). Each tile in $T_i$ overlaps with four other tiles in $T_i$ (Figure 3).

*Step 2 (synthesis)*: 1. For each lattice point, randomly select a tile from $\{T_i\}$ and center it on the lattice point. 2. Blend all the overlapping tiles (two layers) using a feathering technique [21]. The synthesis time is less than 10 seconds in Matlab code, on a 800Mhz laptop computer.

Figure 3 shows both the minimum and the maximum tiles used in the brick example (Figure 2 (a)). It may be counter intuitive to some people that the basic tile for this pattern is NOT the size and shape of a single physical brick. The reason we are using overlapping maximum tiles for feathering is for a smoother transition on the tile boundaries. Figure 2 demonstrates the difference between simple direct tiling and our random selection method. Our goal is to preserve the near-regular nature of the input texture *as well as* the variations among and within the tiles[1].

## 3 Discussion

The reason that current texture synthesis algorithms appear to work on certain near-regular patterns (patterns of dots, for example) is due to a judicious choice of the window size and shape. Conversely, the wrong choice of window size and shape usually causes their failure. It is pointed out in [5]: "Determining precisely what are the patches for a given texture and how they are put together is still an open problem." For near-regular patterns, the window (patch) size/shape/orientation is a crucial parameter. It should be realized by now that the regularity preservation problem can not be solved by adjusting window size alone. One advantage of our approach is that the tile orientations (not necessarily horizontal/vertical), and the shape and size are determined up front, explicitly and customized to each near-regular input texture pattern (Figure 4).

One of the perils when approaching near-regular texture is the temptation to use direct tiling (of a unit tile) to fill the whole 2D image. Though tiling is the central theme and appropriate means for many artistic and design tasks [23, 8], it is usually NOT suited for providing natural visual effects in the context of texture synthesis. The results from simple tiling are overly regular, usually more so than the original input sample. When one really understands the making of a periodic pattern and its generating regions [19], modifications can be made to direct tiling such that more natural appearance can be achieved. In particular, we only used translational symmetry in this paper, rotation and reflection symmetries can also be used to produce much smaller tiles. This means that a much larger sample set of observed statistical variations can be obtained in a principled way.

## 4 Limitations and Research Directions

In this paper, we provide a new method for near-regular texture synthesis. Our method differs from most local-neighborhood approaches to texture synthesis in that it first does a texture structure analysis. Also it separates the treatment of spatial layout regularity (tiles) from the intensity/color regularity (the content of a tile). A special treatment for near-regular texture in texture synthesis has been a missing piece in the texture synthesis puzzle.

One obvious limitation of this paper is its focus on near-regular texture alone. There are many ways to combine our approach with existing local-neighborhood methods. One way is to build a near-regular texture classifier $F$. Given a sample texture $T$, if $F(T) = 1$ or larger than a certain threshold, use our near-regular texture algorithm, otherwise resort to one of the local-neighborhood methods. People have already experimented with such classifiers, e.g. [3] provides a score for a textured pattern that seems to be consistent with human perception. Our lattice detection algorithm [15] can also serve as a periodicity measure.

We have constructed the basic framework of our method to enhance the existing treatment of near-regular texture. However, the quality of current results is limited by the simple synthesis technique used

[1] More results can be seen on our website http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/yanxi/www/images/Texture/index.html

as a quick feasibility study of our ideas. The simple feathering technique used in our synthesis step can produce blurring and ghosting effects when the tiles are not well registered (Figure 3 (d)). We are developing more sophisticated synthesis methods, e.g. dynamic programming or deformable registration, to produce high quality, more visually convincing outputs.

Our long term goal is to treat near-regular patterns as part of the chaotic-symmetry continuous spectrum [14]. Dimensions of near-regular texture include resolution (scale), spatial local or global transformations (rigid, affine, perspective, random ...) [15, 16], color (single, multi), and density (regular, stochastic) [22].

## 5 Acknowledgement

We highly appreciate the constructive comments given by two anonymous reviewers. Dr. Liu thanks Professor W. T. Freeman, A.A. Efros and R.T. Collins for helpful discussions.

## References

[1] L. Bieberbach. Über die Bewegungsgruppen der n-dimensional en Euklidischen Räume mit einem endlichen Fundamental bereich. *Göttinger Nachrichten*, pages 75–84, 1910.

[2] D.A. Chambers, editor. *DNA: the double helix: Perspective and Prospective at Forty Years.* New York Academy of Sciences, New York, 1995.

[3] D. Chetverikov. Pattern regularity as a visual key. *Image and Vision Computing*, 18:975–986, 2000.

[4] H.S.M. Coxeter and W.O.J. Moser. *Generators and Relations for Discrete Groups.* Springer-Verlag, New York, fourth edition, 1980.

[5] A.A. Efros and W.T. Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, pages 35–42, 2001.

[6] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, 1999.

[7] R.P. Feynman. *Six Not-so-easy pieses: Einstein's relativity, Symmetry, and Space-Time.* Perseus Press, Cambridge, Mass, 1998.

[8] B. Grünbaum and G.C. Shephard. *Tilings and Patterns.* W.H. Freeman and Company, New York, 1987.

[9] István Hargittai and Magdolna Hargittai. *In Our Own Image: Personal Symmetry in Discovery.* Kluwer Academic Publishers, 2000.

[10] T.I. Hsu and R. Wilson. A two-component model of texture for analysis and synthesis. *IEEE Trans. on Image Processing*, 7(10):1466–1476, October 1998.

[11] C.S. Kaplan and D.H. Salesin. Escherization. In *the 27th International Conference on Computer Graphics and Interactive Techniques*, July 2000.

[12] M. Leyton. *Symmetry, Causality, Mind.* The MIT Press, Cambridge,Massachusetts, 1992.

[13] L. Lin, C. Liu, Y. Xu, B. Guo, and H.Y. Shum. Realtime texture synthesis by patch-based sampling. *ACM Transactions on Graphics (TOG)*, 20(3):127–150, July 2001.

[14] Y. Liu. *Computational Symmetry*, volume 80, chapter 21, pages 231–245. Wenner-Gren International Series, Portland, London, ISBN I 85578 149 2, 2001.

[15] Y. Liu and R. T. Collins. A Computational Model for Repeated Pattern Perception using Frieze and Wallpaper Groups. In *Computer Vision and Pattern Recognition Conference (CVPR'00)*, pages 537–544, Hilton Head,SC, June 2000. IEEE Computer Society Press. (http://www.ri.cmu.edu/pubs/pub_3302.html).

[16] Y. Liu and R. T. Collins. Skewed Symmetry Groups. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01)*, Kauai,HI, December 2001. IEEE Computer Society Press. (http://www.ri.cmu.edu/pubs/pub_3815.html).

[17] W. Miller Jr. *Symmetry Groups and Their Applications.* Academic Press, New York, 1972.

[18] A.R. Rao and G.L. Lohse. Identifying high level features of texture perception. *CVGIP: Image Processing*, 55:218–233, 1993.

[19] D. Schattschneider. The plane symmetry groups: their recognition and notation. *American Mathematical Monthly*, 85:439–450, 1978.

[20] M. Senechal. *Quasicrystals and Geometry.* Cambridge Universy Press, 1995.

[21] H. Shum and R. Szeliski. Construction of panoramic mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, February 2000.

[22] Y. Tsin, Y. Liu, and V. Ramesh. Texture replacement in real images. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'01). (http://www-2.cs.cmu.edu/ ytsin/research/texture_replacement/index.html)*, Kauai, December 2001. IEEE Computer Society Press.

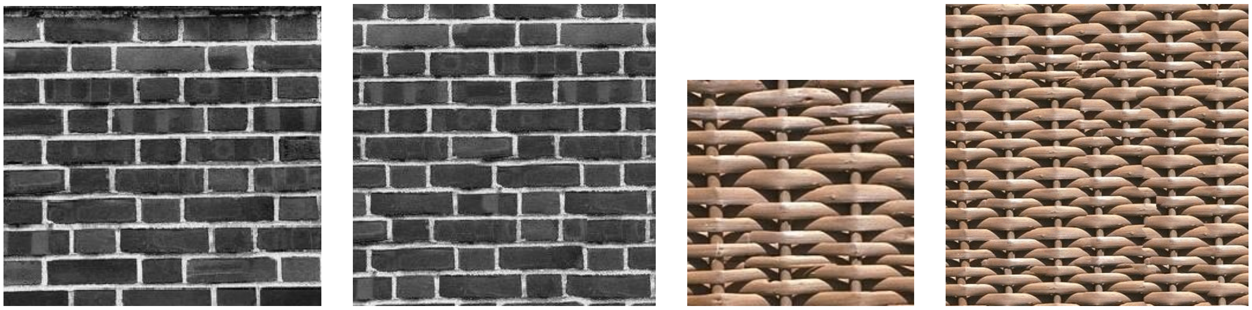
[23] D.K. Washburn and D.W. Crowe. *Symmetries of Culture: Theory and Practice of Plane Pattern Analysis.* University of Washington Press, 1991.

[24] L.Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, pages 479–488, July 2000.

[25] Y.Q. Xu, S.C. Zhu, B.N. Guo, and H.Y. Shum. Asymptotically admissible texture synthesis. In *International Workshop on Statistical and Computational Theories of Vision*, July 2001.

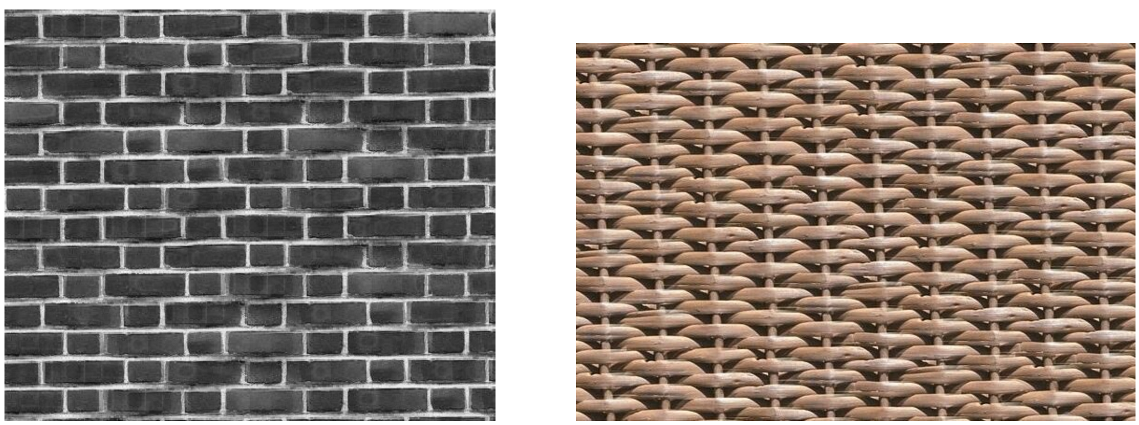
[26] Anthony Zee. *Fearful Symmetry.* Princeton University Press, 1999.

[27] S.C. Zhu, X. Liu, and Y. Wu. Exploring texture ensembles by efficitn markov chain monte carlo. *IEEE Transaction on PAMI*, 22(6), 2000.

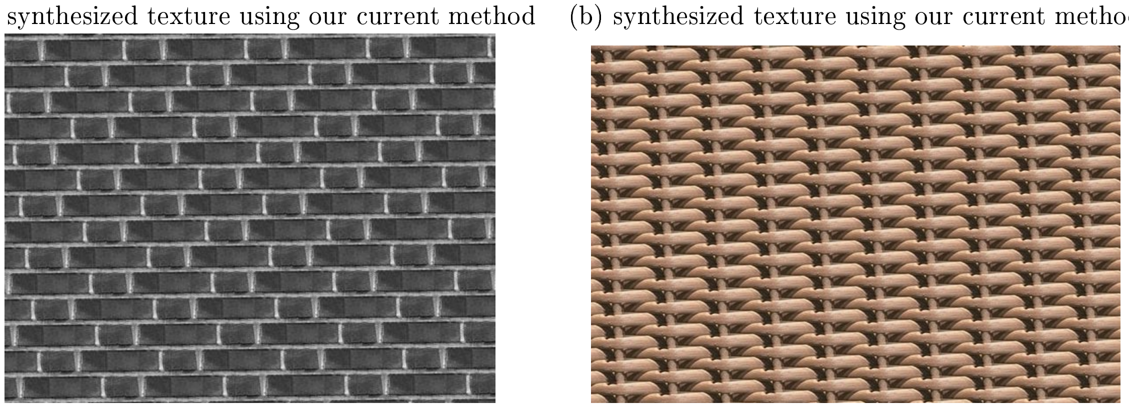


(a) (b) (c) (d)

Figure 1: (a),(c): input texture samples. (b),(d): texture synthesis results from [5]. This is one of the best results on brick wall texture synthesis that we can find. However, the regularity in the input texture samples is not faithfully preserved in the synthesized texture (brick example in (b): two short bricks are stacked together. straw pattern in (d): one vertical line is terminated midway).

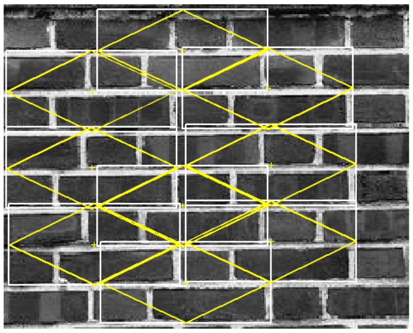


(a) synthesized texture using our current method (b) synthesized texture using our current method

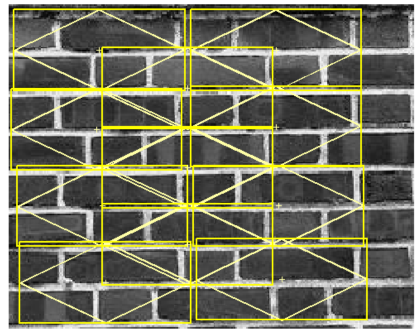


(c) texture from direct tiling (d) texture from direct tiling

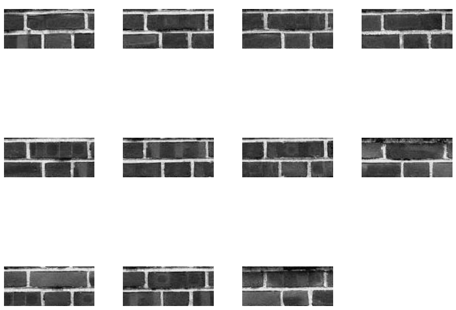
Figure 2: (a),(b): random sampling from tile sample sets plus feathering (our method). which preserve both the near-regular nature of the texture **and** the variations across tiles. The symmetry group of both patterns is classified as *cmm* containing translation, rotation, reflection and glide-reflection symmetries [15]. (c),(d): direct tiling results. Though the regularity of the input texture is preserved, the synthesized texture does not reflect the intensity variations in the input texture.



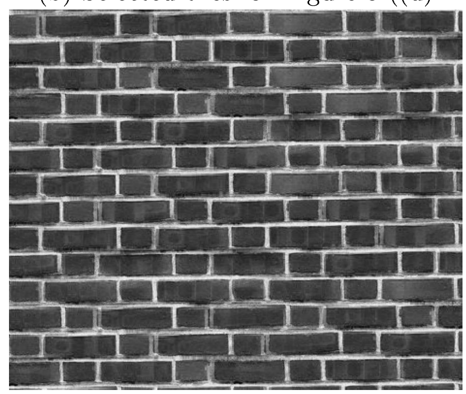
(a) Selected tiles for Figure 2 (a)



(b) Selected tiles for Figure 3 ((d)



(c)



(d)

Figure 3: The sample tiles (rhombic shaped tiles are *minimum tiles* $\{t_i\}$ and rectangle shaped tiles are *maximum tiles* $\{T_i\}$) are shown, they are carved from the input brick texture. (a) and (b) show two different lattice positions. (c) a sample set of maximum tiles. (d) a failed case of our current method, where double line segments present due to non-exact match.

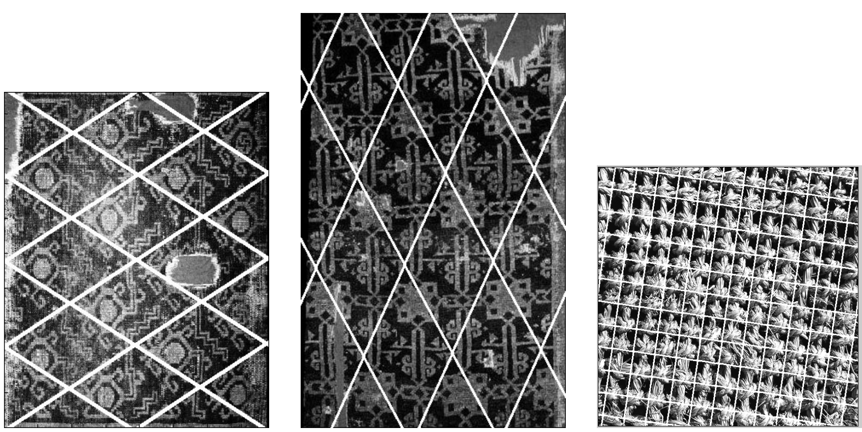


Figure 4: Examples of imperfect, real-world near-regular patterns overlayed with automatically detected underlying lattices.