Deformable Texture: the Irregular-Regular-Irregular Cycle

Yanxi Liu and Wen-Chieh Lin

CMU-RI-TR-03-26

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This research is supported in part by an NSF research grant IIS-0099597.

ABSTRACT

Departures from a regular texture pattern can happen in many different dimensions. Previous related work has focused on faithful texture synthesis for near-regular texture departing along the color and intensity axes while the underlying geometric regularity is well preserved. In this paper, we address the issue of faithful texture synthesis for textures that have both the structural and color/intensity deformations. We propose a framework that treats irregular texture as a deformation from regular texture by first deducing a deformation field between the input irregular texture and its corresponding regular counterpart. The novel view in this work is to treat the deformation field itself as a texture that is both visual and functional. As a result, we can handle faithful texture synthesis for a much larger variety of near-regular textures.

1 Introduction

A large amount of texture synthesis work has been done in computer graphics. Some of the representative work includes [2, 1, 3, 6, 12, 5, 8, 13]. Most existing work shares two common themes: (1) texture is a stochastic, random phenomena (non-regular) and (2) the texture synthesis is a local process. Alternatively, we view textures as different forms of departures from regularity, and we are exploring a method that combines both local intensity and global structural information.

Departures from a regular texture pattern can happen in many different dimensions, including: color, intensity, geometry (global versus local, rigid versus affine versus perspective etc.), resolution, occlusion or non-linear distortion caused by viewing media (Figure 1). Previous related work has focused on faithful texture synthesis for near-regular texture with color and intensity variations only [10] while maintaining underlying geometric structural regularity. Examples include brick walls, tiled floor and woven straw sheet. In this paper,

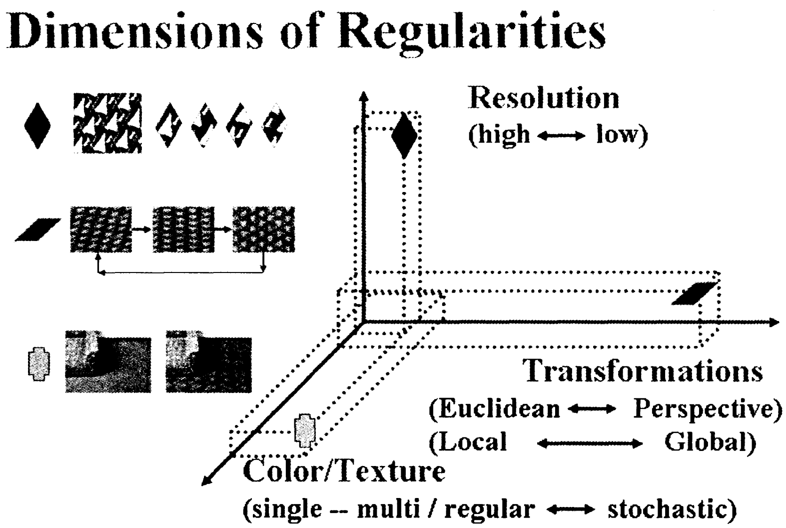


Figure 1: Regularity of texture patterns can span a multi-dimensional space.

we address the issue of faithful texture synthesis for textures that have both structural and color/intensity deformations. We propose a framework that captures the geometric deformation of irregular texture by deducing a deformation field between the input texture and its corresponding regular texture. We treat the deformation field both as a 2D vector field and as a texture that can be synthesized. Therefore, the deformation field has a dual property that is both visual and functional. As a result, we can handle faithful texture synthesis for a much larger variety of near-regular textures. Our long term goal is to construct a computational model bridging regular, near-regular, irregular and stochastic textures.

For the convenience of clarity in this paper, we shall use the term *regular* texture to refer to periodic wallpaper patterns [9]; *near-regular* texture to refer to textures with little geometric structural distortions but considerable statistical color and intensity departures from regular texture [10]; and *irregular* texture to those textures that have both structural and color/intensity deformations from regular texture, i.e. the underlying lattice of a texture is irregular. See Figure 2 for examples of each of these texture types. See Figure 3 for an irregular texture sample and its near-regular version (its underlying lattice is straightened out).

Figure 2: A sampler of different types of textures.

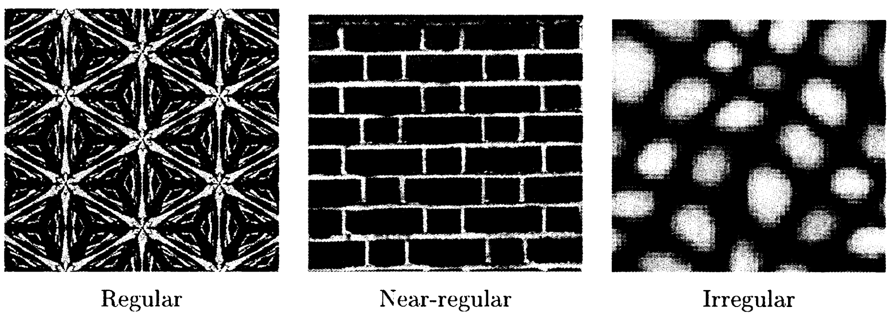
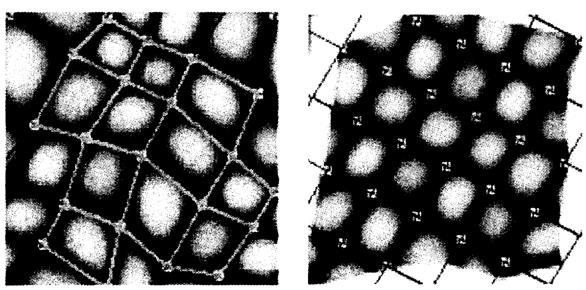


Figure 3: An irregular texture, its lattice and its near-regular counterpart.



2 Our Approach

Our texture synthesis algorithm is illustrated in Figure 4 using one simple irregular texture example. When given an input irregular texture p, we first identify its underlying lattice

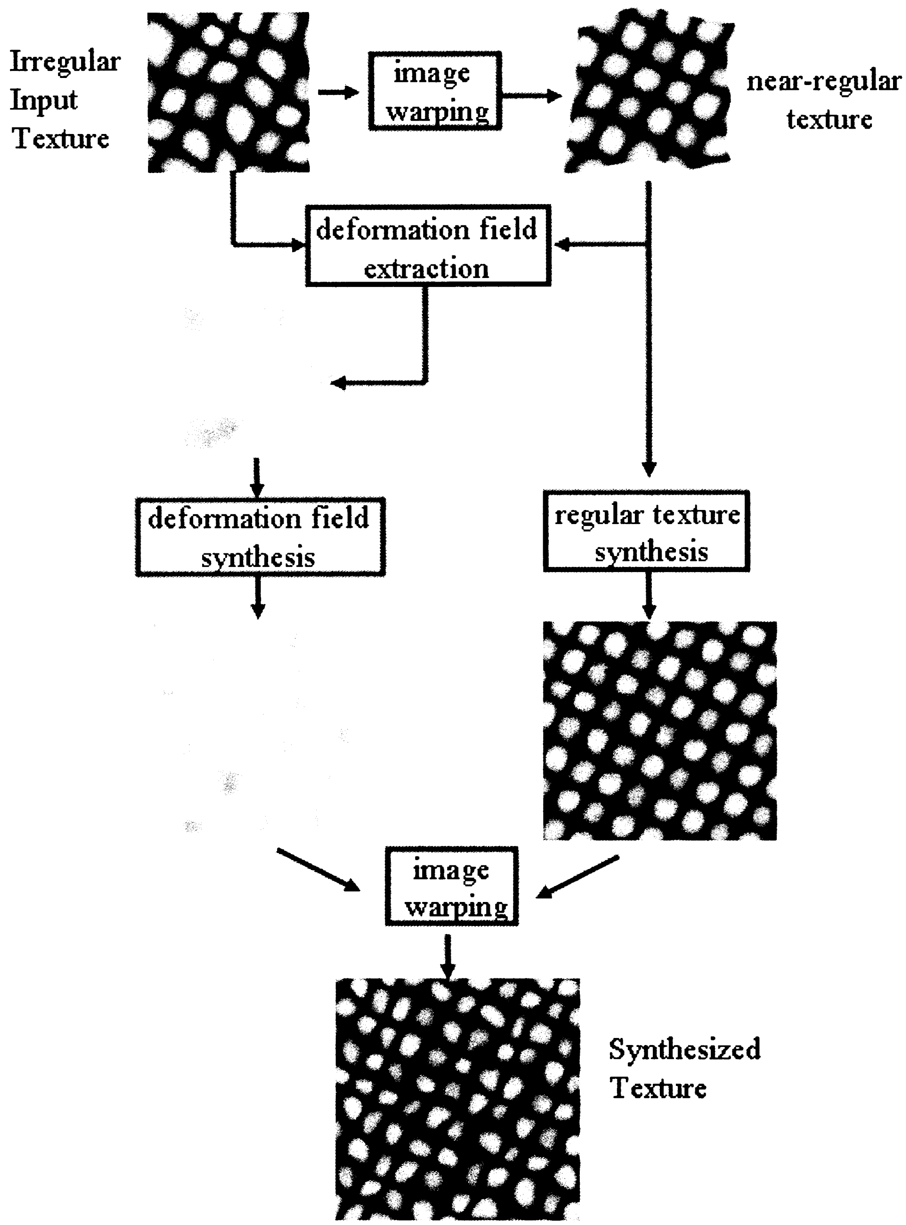


Figure 4: This is an overview of our approach. Starting with an input irregular texture p, its near-regular version is obtained by straightening out its underlying lattice L_{ir} into its nearest regular lattice L_r (Figure 3). Correspondingly, a near-regular texture p_r is obtained from p and can be synthesized into P_r. Then a deformation field between L_r and L_{ir} is computed and further synthesized into D with the same size as P_r. Finally, applying D to P_r a synthesized irregular texture is achieved.

interactively. A deformation is computed such that the texture lattice becomes its "nearest" regular version, and correspondingly p turns into a near-regular texture p_r. We then compute the deformation field between from p_r to p. The deformation field and p_r are synthesized respectively to D and P_r of the same dimension. Finally, the synthesized texture is produced by deforming P_r using D. In the following, we discuss each step in our algorithm.

2.1 From Irregular to Near-Regular Texture Structure

Each regular texture has an underlying 2D lattice L_r that is generated by two linearly independent vectors $\vec{t_1}, \vec{t_2}$. An irregular texture, as a departure from a regular texture, also has an underlying lattice L_{ir} which is a geometric distortion of L_r. See Figure 3 for an example. There may be many potential regular lattices that L_{ir} can deform to. Our goal is to determine the generating vectors $\vec{t_1}, \vec{t_2}$ of an irregular texture p's corresponding regular lattice L_r such that the total amount of deformation between L_{ir} and L_r is minimal. The total deformation is defined as the sum of the length variations between all corresponding link pairs between lattices L_{ir} and L_r. We formulate this as a minimization problem:

$$\min_{||\vec{t_1}||,||\vec{t_2}||,\theta} E = \sum_{i=1}^{N_i}(l_i - ||\vec{t_1}||)^2 + \sum_{j=1}^{N_j}(l_j - ||\vec{t_2}||)^2$$
$$+ \sum_{k=1}^{N_k}(l_k - ||\vec{t_1} + \vec{t_2}||)^2 + \sum_{m=1}^{N_m}(l_m - ||\vec{t_1} - \vec{t_2}||)^2$$

where l_i, l_j, l_k, and l_m, are the lengths of the links in lattice L_{ir} corresponding to links in L_r along the directions of $\vec{t_1}, \vec{t_2}, \vec{t_1} + \vec{t_2}$, and $\vec{t_1} - \vec{t_2}$, respectively. N_i, N_j, N_k and N_m are the total number of links in L_{ir} corresponding to each direction. θ is the angle between $\vec{t_1}$ and $\vec{t_2}$ which can be deduced from the lengths of $\vec{t_1}, \vec{t_2}$ and $\vec{t_1} + \vec{t_2}$.

2.2 Deformation Field Extraction

Once the optimal regular lattice L_r is obtained, we are able to compute a unique deformation field between the input texture and its near-regular version (structurally regular with color/intensity irregularities) by deforming the underlying lattice of input texture L_{ir} to L_r or vice versa. We use the MFFD algorithm proposed in [7], where a 1-1 warping field is computed. The basic idea is to use the corresponding lattice points between L_{ir} and L_r as corresponding point features (control points). The problem then turns into an interpolation problem: to establish warping vectors for all pixels of the texture in addition to those pre-specified lattice points. At the same time, we keep the warped field one-to-one and as smooth as possible. As a result, we obtain a deformation field between an irregular texture and its near-regular counterpart.

2.3 Near Regular Texture Synthesis

As the input texture is rectified into a near-regular texture, the structural deformation of the texture becomes minimal while color and intensity variations among different tiles remain. Here we use the term "tile" to indicate the smallest parallelogram-shaped 2D region on a regular texture that can reproduce the texture patterns under the texture's translation subgroup [10]. For regular texture, only one tile is needed. For near-regular texture, we can collect a set of sample tiles, chosen in a principled manner [10], with roughly the same size and shape but varying color and intensity. We apply synthesis algorithm to the near-regular texture generated in Section 2.1. The synthesis approach is similar to the one reported in [10], where we can take advantage of the structural regularity and color/intensity variations of near-regular textures by randomly selecting tiles from a customized tile set.

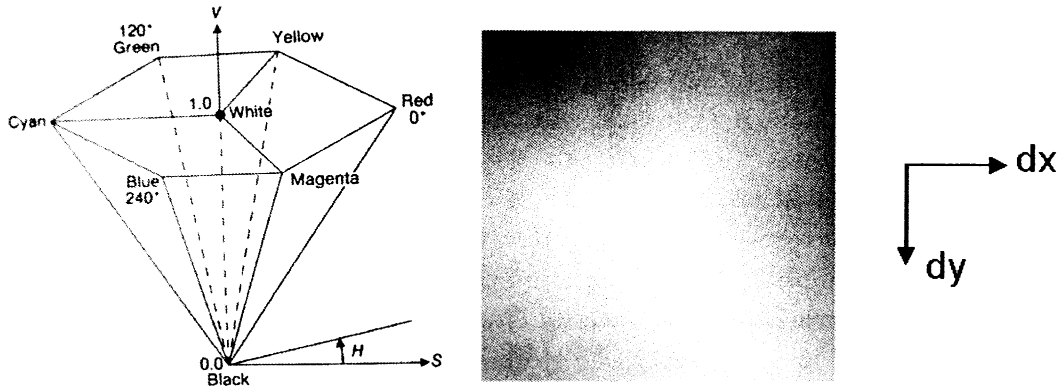
2.4 Deformation Field Synthesis

A central intuition behind our approach is the realization of the *duality* of a deformation field. On the one hand, a deformation field is a vector field that can take a texture (any image actually) into its warped version. On the other hand, **a deformation field itself can be viewed as a texture**. Therefore it can be subject to texture synthesis as well. So, our idea is to synthesize a deformation field between near-regular and irregular textures, and then apply this synthesized deformation field to a synthesized near-regular texture to achieve the effect of a synthesized irregular texture (Figure 4).

Deformation fields are usually non-regular, stochastic textures. We can apply any existing texture synthesis algorithm to generate a new deformation field of any size. Some differences from standard texture synthesis are:

First, the MFFD algorithm [7] actually outputs a warping function that transforms the pixel location of the source image to the target image in X and Y directions separately. To be able to synthesize the deformation field, we have to convert the absolute global displacements into relative local displacements $\delta(x), \delta(y)$ and treating them as a pair of color channels at each pixel location. For better visualization and evaluation, we use hue, saturation, and value (HSV) color space to represent a deformation field (Figure 5). *Hue* is the color type (such as red, blue, or yellow); measured in values of 0-360 by the central tendency of its wavelength. *Saturation* is the 'intensity' of the color (or how much grayness is present), measured in values of 0-100% by the amplitude of the wavelength, *Value* is the brightness of the color, measured in values of 0-100% by the spread of the wavelength. HSV is a non-linear

Figure 5: We use hue, saturation, and value (HSV) color space (left) to represent a deformation field. In our representation (right), the value component is set to 1. A pure white color means zero movement, and a pure red color means a movement in positive X direction, and etc..



Color scheme used | Displacement Map

transformation of the RGB color space. In our representation, the value component is set to 1. A pure white color means zero movement, and a pure red color means a movement in positive x direction, etc..

Second, we need to define a distance function specifically for the deformation field to measure the difference and smoothness of the vector field. Two versions of DF texture synthesis algorithms are experimented. Method one uses patch-based texture synthesis [10, 3] on the original deformation field. We define a distance function between two 2D regions a, b as following:

$$DIST(a, b) = \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{\alpha \cdot \theta_{ij} + (1 - \alpha) \cdot (abs(\|\vec{a}_{ij}\| - \|\vec{b}_{ij}\|))}{MN}$$

where M, N are the numbers of row and columns of a or b, θ_{ij} is the angle between the two displacement vectors at position i, j and $\alpha \in [0, 1]$ is a weight coefficient. In addition, the smoothness in the boundary of the tiles is more demanding than that of a normal texture synthesis task. The image feathering algorithm [11] is applied to the synthesized deformation field to stitch the overlapping tiles.

The method two for DF texture synthesis uses pixel-based texture synthesis [4, 7] on the control points of DF. Here we treat DF control-point texture as if it is a color texture. We have found the synthesized results of the second method to be more faithful to the input. Therefore, the results using the method two are presented here (Figures 7 - 8).

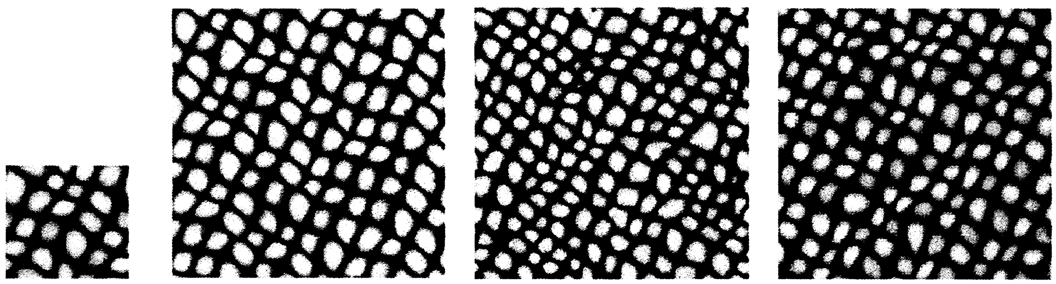
2.5 From Near-regular Texture to Irregular Texture

The final stage of irregular texture synthesis is straightforward: simply apply the synthesized deformation field to the synthesized near-regular texture.

3 Experimental Results

We have applied our texture synthesis algorithm to a set of textures with varying degrees of structural deformation. Figure 6 shows the texture synthesis results on the same texture sample by our method and two other texture synthesis algorithms [3, 12]. One can observe on close inspection that alterations in the shape and color of the input texture sample are more faithfully preserved by our method.

Figure 6: Texture synthesis results compared with others. One can observe on close inspection that alterations in color and shape of the input texture sample is more faithfully preserved by our method.



sample input | Efros&Freeman[3] | Wei & Levoy[12] | Our Method

The intermediate and the final results of sample texture synthesis are shown in Figures 7-9. From the deformation field texture, one can observe that the departure motions from regularity in each of the three textures are quite different (refer to Figure 5): Figure 7 shows a texture with many local motions toward the lower left quadrant. The texture in Figure 8, on the other hand, departs from its regular version via large left-right motions (positive X or Y directions). Figure 9 shows a texture departs from its regularity along diagonal directions. Through these texture synthesis results, we are able to gain a deeper understanding of how irregular textures move from their regular counterparts. This can be used as a basis for texture categorization.

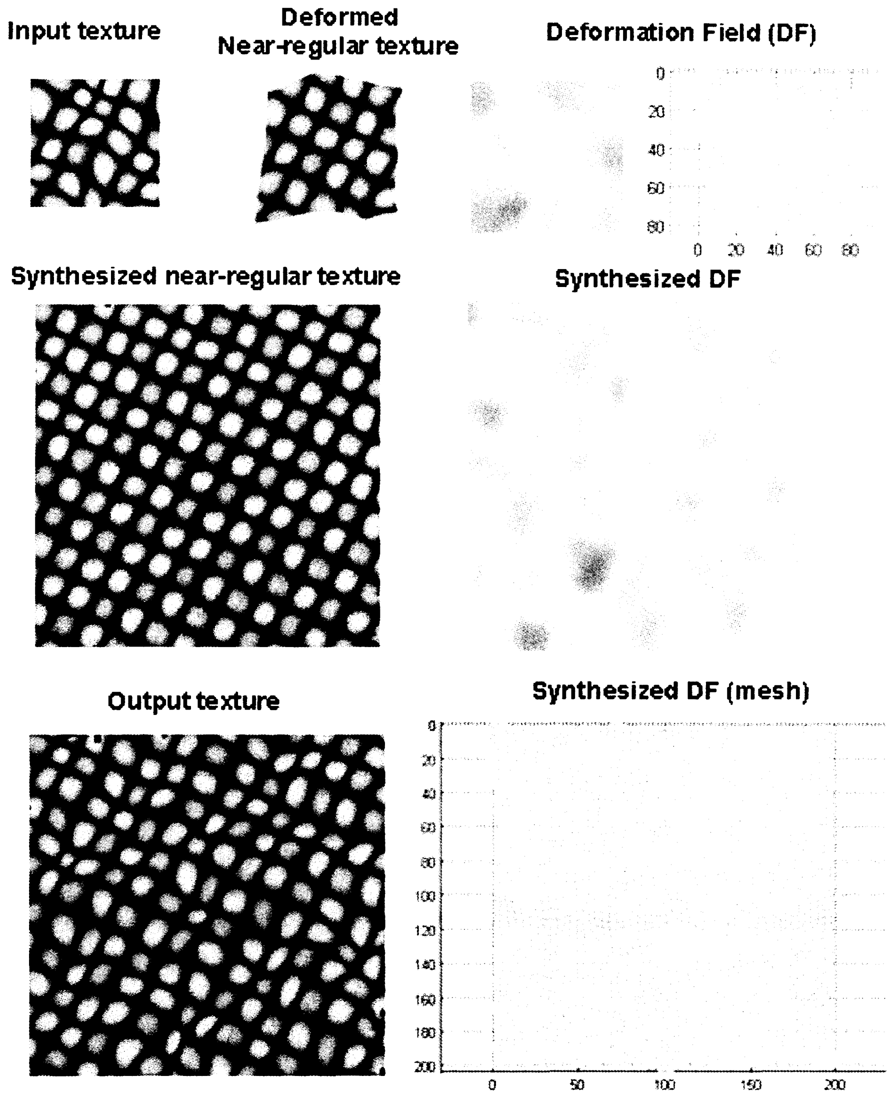


Figure 7: Texture synthesis results (intermediate and final) are shown for an irregular texture that has many local motions toward the lower left quadrant (Figure 5). The color textured deformation field indicates movements from regular to irregular lattices.

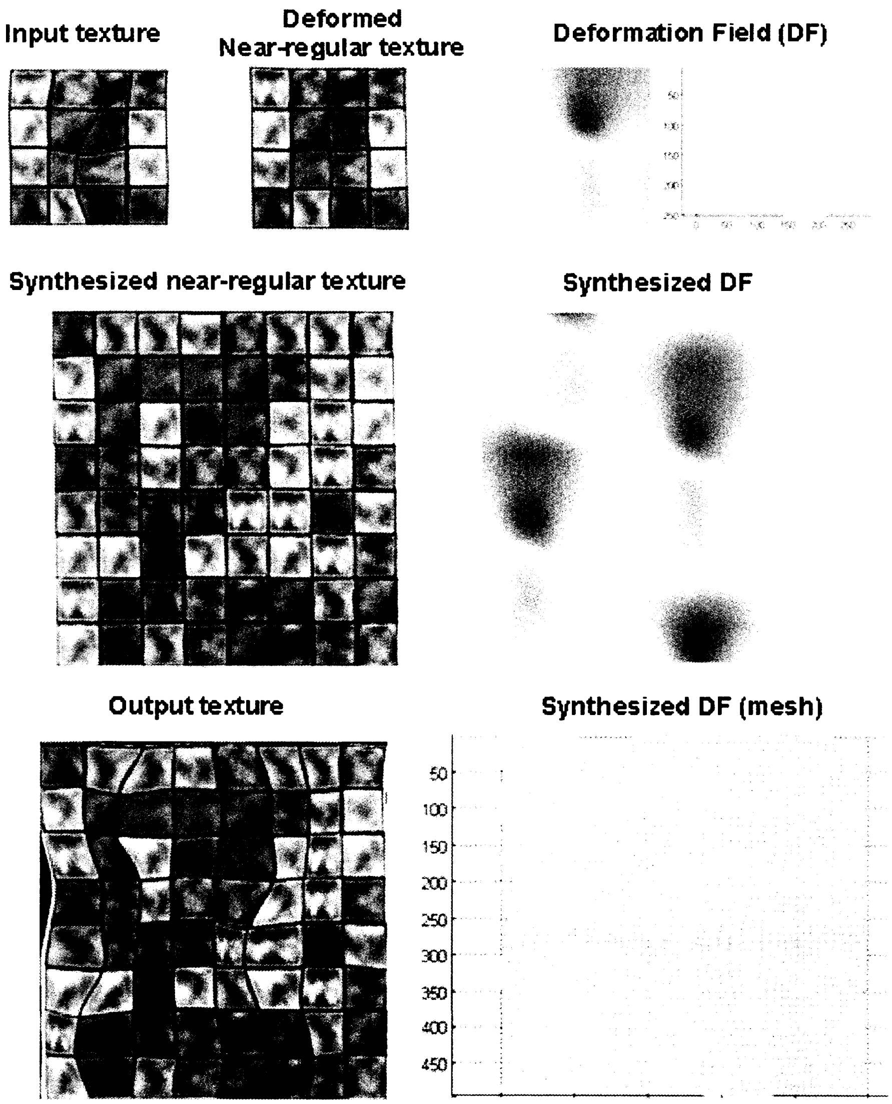


Figure 8: Texture synthesis results (intermediate and final) are shown for another irregular texture that has large horizantal motions. The color textured deformation field indicates movements from regular to irregular lattices.

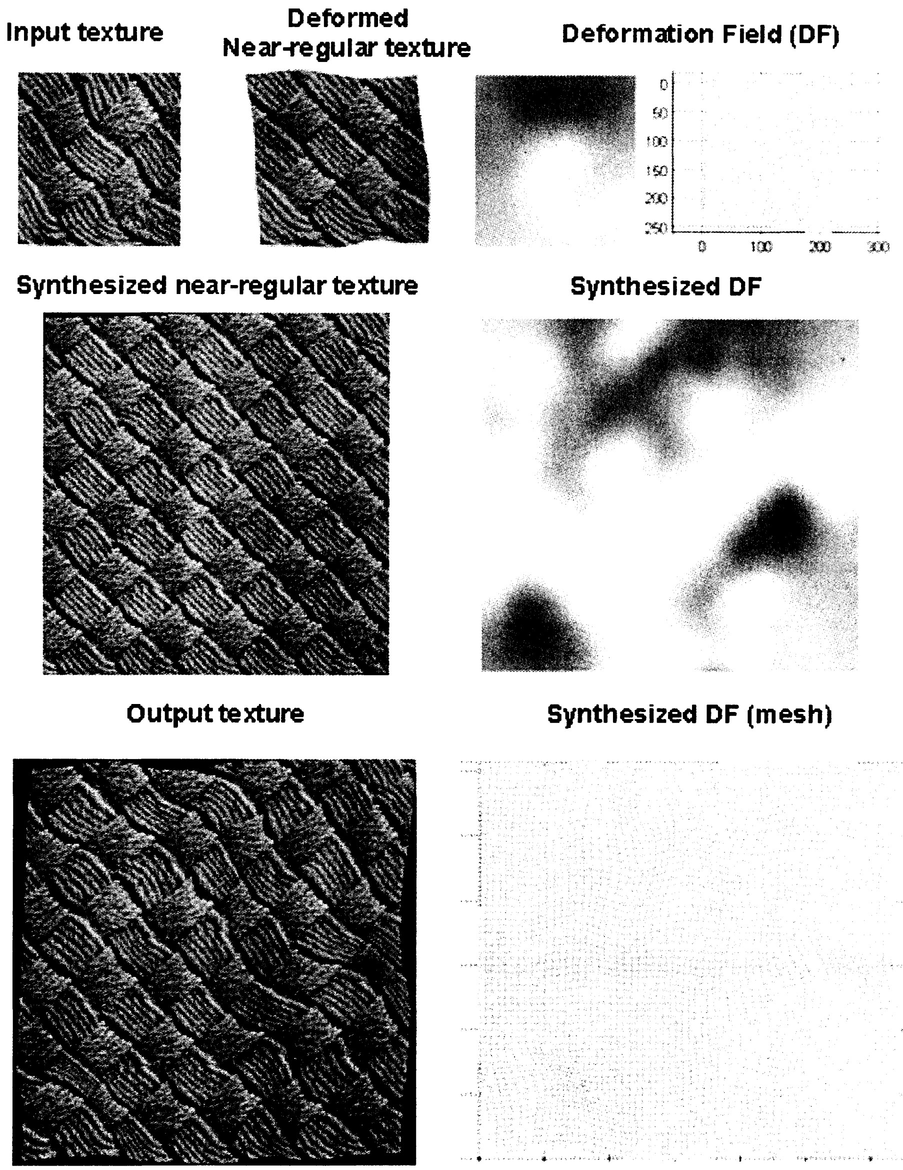


Figure 9: Texture synthesis results (intermediate and final) are shown for another texture that departs from it regular version towards the upper right corner (Figure 5). The color textured deformation field indicates movements from regular to irregular lattices.

4 Conclusion and Future Work

We have proposed a novel method for irregular texture synthesis by synthesizing both the deformation field and near-regular textures. The initial results show the feasibility of the proposed approach in faithful texture synthesis, especially for those textures that have both geometric and color/intensity distortions from regularity. For certain irregular textures, our method produces more faithful synthesized results than existing methods (Figure 6). The method is simple and flexible in modeling the deformation field between a regular texture and its structural and color/intensity variations through decomposition. We are also investigating the robustness of the algorithm with respect to different degrees of structural distortions.

5 Acknowledgement

Dr. Liu would like to thank Chenyu Wu, who carried out an initial implementation based on the idea of near-regular texture synthesis using deformable lattices when Dr. Liu was visiting Microsoft Asia in November 2002.

References

[1] M. Ashikhmin. Synthesizing natural textures. In *ACM Symposium on Interactive 3D Graphics*, pages 217–226, 2001.

[2] J.S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture image. In *SIGGRAPH Proceedings*, pages 361–368, 1997.

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

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

[5] A. Hertzmann, C.E. Jacobs, N. Oliver, B. Curless, and D.H. Salesin. Image analogies. *SIGGRAPH*, 2001.

[6] 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.

[7] S.Y. Lee, K.Y. Chwa, S.Y. Shin, and G. Wolberg. Image metamorphosis using snakes and free-form deformations. *SIGGRAPH*, 1995.

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

[9] Y. Liu, R.T. Collins, and Y. Tsin. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, to appear, 2003.

[10] Y. Liu, Y. Tsin, and W. C. Lin. The promise and perils of near-regular texture. *International Journal of Computer Vision*, In Press, 2003.

[11] 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.

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

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