Deformable Texture: the Irregular-Regular-Irregular Cycle

Yanxi Liu and Wen-Chieh Lin
The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave. Pittsburgh, PA 15213
{yanxi,wclin}@cs.cmu.edu

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 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 near-regular counterpart. The novel idea 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, 6, 12, 1, 3, 5, 9, 13, 7]. Most existing work shares two common themes: (1) texture is a stochastic, random phenomena (non-regular) and (2) 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 different dimensions, including: color, intensity, geometry (global versus local, rigid versus affine versus perspective etc.), resolution, occlusion or nonlinear distortion caused by viewing media. Previous related work has focused on faithful texture synthesis for near-regular texture with color and intensity variations [11] while maintaining underlying geometric structural regularity. Examples include brick walls, tiled floor and woven straw sheet. In this paper, we address the issue of faithful texture synthesis for textures that have both structural and color/intensity deformations.

For the convenience of clarity in this paper, we shall use the term *regular* texture to refer to periodic wallpaper patterns [10]; *near-regular* texture to refer to textures with little geometric structural distortions but considerable statistical color and intensity departures from regular texture [11]; 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[1]. See Figure 1 for examples of each of these texture types. See Figure 2 for an irregular texture sample and its near-regular version (its underlying lattice is straightened out).

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.

2 Our Approach

Our texture synthesis algorithm is illustrated in Figure 3 using one simple irregular texture example. When given an input irregular texture p, we first identify its underlying lattice interactively. The texture lattice is warped into its "nearest" regular version based on an energy minimization function. Correspondingly, texture p turns into a near-regular texture p_r. A deformation field between p_r and p can then be computed. The deformation field and p_r are synthesized respectively to D and P_r of equal dimension. Finally, the synthesized texture is produced by deforming P_r using D. In the following, we discuss each step in our algorithm.

---

[1] Note, within this paper, we are making a distinction between the terms "irregular" (deformation from regularity) and "non-regular" (no regularity).

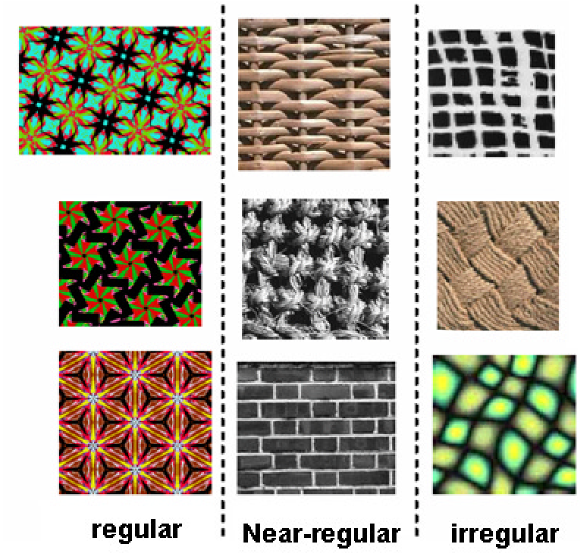


Figure 1: A sampler of different types of textures.

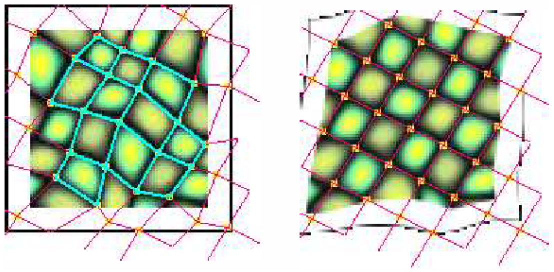


Figure 2: Left: An irregular texture overlaid with its lattice. Right: its near-regular counterpart.

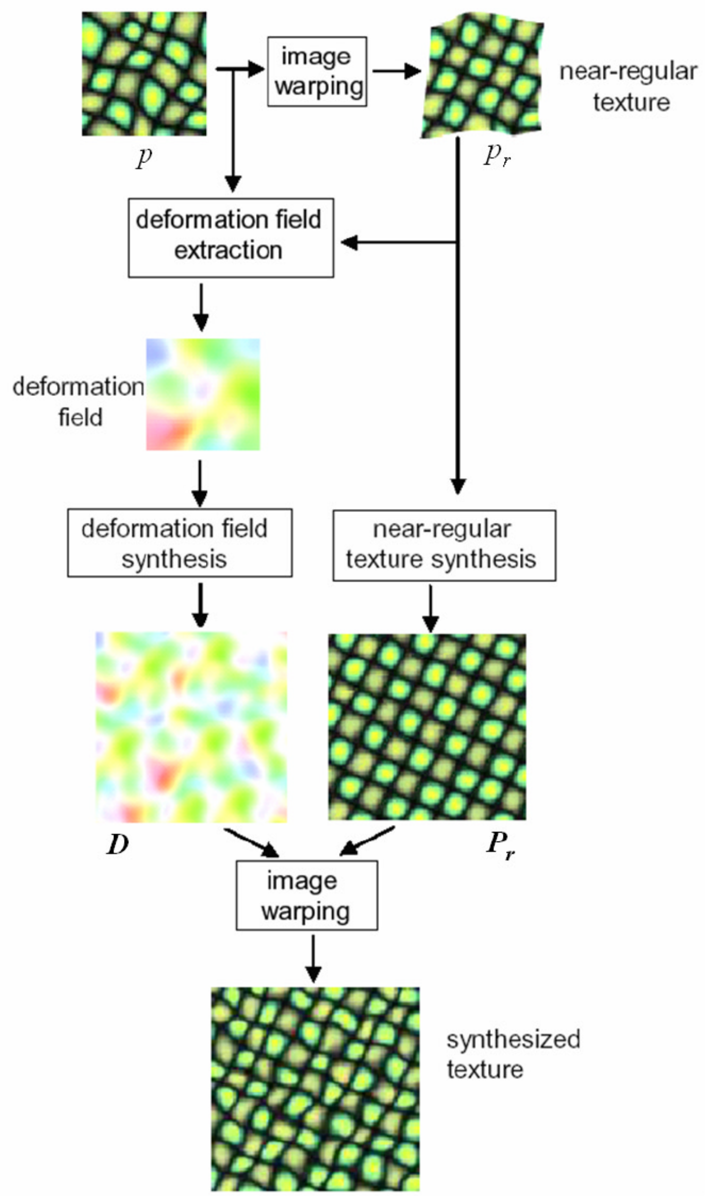


Figure 3: This is an overview of our approach. Starting with an input irregular texture p, its near-regular version is obtained by straightening out its underlying lattice L_{ir} into its nearest regular lattice L_r (Figure 2). Correspondingly, a near-regular texture p_r is obtained from p and synthesized into P_r; and a deformation field between L_r and L_{ir} is computed and synthesized into D. Finally, a synthesized irregular texture is achieved by applying D to P_r.

2.1 From Irregular to Near-Regular Texture

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 2 for an example. There may be many potential regular lattices that an irregular lattice L_{ir} can deform to. However, we are looking for that particular regular lattice L_r with generating vectors $\vec{t_1}, \vec{t_2}$ such that the total amount of deformation between L_{ir} and L_r is minimal. The process can thus be formulated 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 $\vec{t_1} + \vec{t_2}$.

The result of this step is a near-regular texture generated from the input irregular texture.

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 multilevel free-form deformation (MFFD) algorithm proposed in [8], 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 control points (corresponding point features). The MFFD algorithm uses these point features as constraints to generate a bijective warping function $\overrightarrow{W}(x, y)$ at all pixel location in addition to the prespecified lattice points. A deformation field $\overrightarrow{\delta W}(x, y)$ is computed from the warping function by converting the global position mapping to a local displacement vector. 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 [11]. For regular texture, only one tile is needed for reproduction and it can be chosen in a principled manner for perception purposes [11]. For near-regular texture, a set of sample tiles with roughly the same size and shape but varying color and intensity are randomly selected and synthesized [11], where we can take advantage of the structural regularity while preserving color/intensity variations of near-regular textures.

2.4 Deformation Field Synthesis

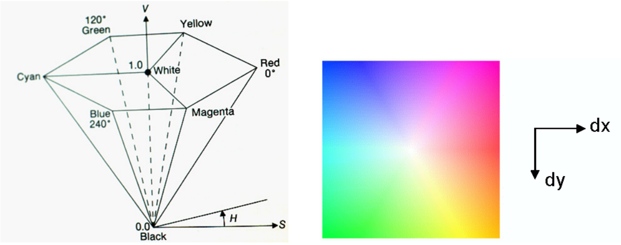
A central intuition behind our approach is the realization of the *duality* of a deformation field. On the one hand, a deformation field is a vector field that can take a texture into its warped version. On the other hand, **a deformation field itself can be viewed as a texture**. Therefore it can be 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 3).

For better visualization and evaluation, we use hue, saturation, and value (HSV) color space to represent a deformation field (Figure 4). *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, 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 transformation of the RGB color space. In our representation, the hue and saturation components are used to represent the direction angle and the length of a 2D displacement vector, respectively, and the value component is set to 1. Thus, a pure white color means zero movement, and a red color means a movement in the positive x direction, etc..

Deformation fields (DF) are usually non-regular, stochastic textures (Figure 6). There are potentially many existing texture synthesis algorithms for this type of textures. Some differences from standard texture synthesis are:

(1) we need to define a distance function specifically for the deformation field to measure the difference and smoothness of the vector field;

Figure 4: 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 the positive X direction, and etc..



Color scheme used Displacement Map

(2) The smoothness of the synthesized field is more demanding because an artifact edge in the synthesized field means a rapid change of the displacements of the pixels, which usually results in a discontinuous warping function.

We have chosen Efros and Leung's pixel-based texture synthesis algorithm [4] for DF texture synthesis to be applied on the lattice points (control points) of DF. The synthesized deformation field is totally determined by the synthesized movements of the lattice points. We can therefore synthesize at the coarsest resolution level and then use the MFFD algorithm[8] to compute the movements in finer resolution levels. We treat the movements of the lattice points as a 2D vector field and apply Efros and Leung's pixel-based synthesis algorithm on the vector field. As a result, the smoothness of the field is guaranteed by the MFFD algorithm and the size and shape of the color blobs in the synthesis field well resembles those in the input DF. Figure 6 shows the deformation field synthesis results.

Note that the computational speed problem in the pixel-based approach is not an issue here because the number of the control points, depending on the number of the tiles in the input lattice, is small. They are usually less than 50 and the number of synthesized control points is in hundreds. These numbers are far less than the number of pixels in an input texture and synthesized texture. For example, the number of control points in the input DF and synthesized DF in Figure 6(A) is 20 and 396, respectively. The running time for synthesizing the DFs in Figure 6, is about 19, 30, 21 and 32 seconds, respectively (on a Pentium 4, 2.66GHz machine with non-optimized Matlab code).

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 5 shows the texture synthesis results on the same texture sample by our method and four other texture synthesis algorithms [4, 3, 12, 7]. 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.

The intermediate and the final results of sample texture synthesis are shown in Figure 6. From the deformation field texture, one can observe that the departure from regularity in each of the four textures is quite different (refer to Figure 4): Figure 6(A) shows a texture with many local motions toward the lower left quadrant. Figure 6(B) shows a texture that departs from its regularity along a diagonal direction (positive X). Figure 6(C)'s texture, on the other hand, departs from its regular version via large left-right motions (positive or negative X directions). Texture in Figure 6(D) moves toward the other diagonal direction (negative X) than the one in example (B). 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.

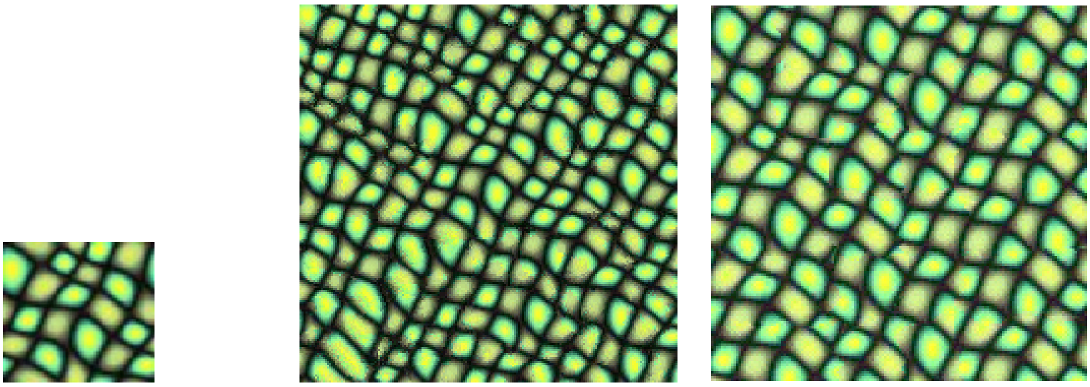
4 Conclusion and Future Work

We have proposed a novel method for irregular texture synthesis by synthesizing both the deformation field and near-regular textures (Figure 3). The initial results show the feasibility of the proposed texture synthesis approach, especially for those textures that have both geometric and color/intensity distortions from regularity (Figure 6). For certain irregular textures, our method produces more faithful synthesized results than existing methods (Figure 5). The method is simple and flexible in modeling and synthesizing the deformation field between an irregular texture and its near-regular counterpart. We are investigating the robustness of the current algorithm while systematically increasing the structural and color irregularities in the input textures.

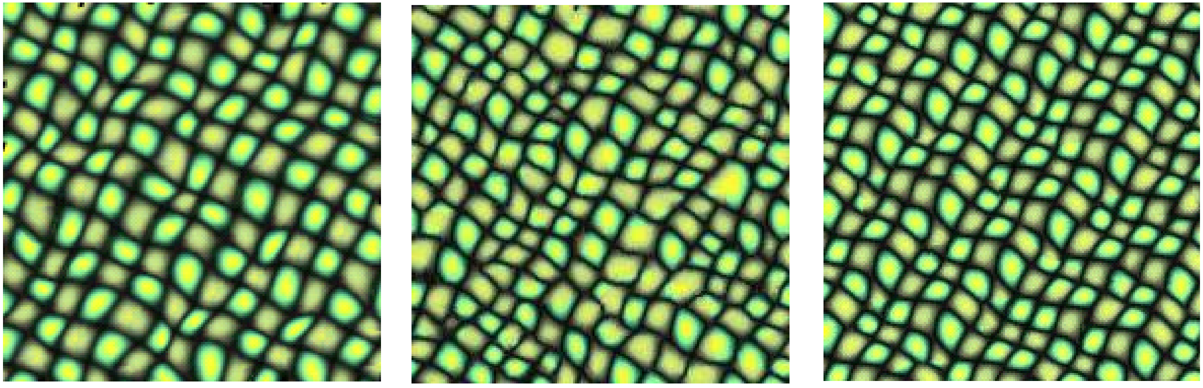
5 Acknowledgement

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

Figure 5: One of our texture synthesis results compared with others. One can observe on close inspection that alterations in color (... yellow, green, yellow, green, ...) and shape of the input texture sample is more faithfully preserved by our method.



sample input Efros&Leung[4] Efros&Freeman[3]



Our Method Wei & Levoy[12] Kwatra *et al* [7]

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] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, SIGGRAPH 2003*, pages 277–286, July 2003.

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

[9] 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.

[10] 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*, In Press, 2003.

[11] Y. Liu, Y. Tsin, and W. C. Lin. The promise and perils of near-regular texture. *International Journal of Computer Vision*, M. Chantler and L. Van Gool (guest Eds.), Accepted for publication. 2003.

[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 efficient markov chain monte carlo. *IEEE Transaction on PAMI*, 22(6), 2000.

Figure 6: Our texture synthesis results (intermediate and final) are shown for four different textures. The color textured deformation field indicates movements from regular to irregular lattices, refer to Figure 4 for the direction and amount of deformation indicated by the color-map representation of the deformation field.