Near-Regular Texture Analysis and Manipulation

Yanxi Liu Wen-Chieh Lin James Hays
Carnegie Mellon University*

Figure 1: Top row: The regularity of the input texture is being manipulated along color and geometry dimensions, regularities decrease from left to right. Bottom row: Texture replacement where extracted geometric and lighting deformation fields from an input texture are applied to new textures.

Abstract

A near-regular texture deviates geometrically and photometrically from a regular congruent tiling. Although near-regular textures are ubiquitous in the man-made and natural world, they present computational challenges for state of the art texture analysis and synthesis algorithms. Using regular tiling as our anchor point, and with user-assisted lattice extraction, we can explicitly model the deformation of a near-regular texture with respect to geometry, lighting and color. We treat a deformation field both as a function that acts on a texture and as a texture that is acted upon, and develop a multi-modal framework where each deformation field is subject to analysis, synthesis and manipulation. Using this formalization, we are able to construct simple parametric models to faithfully synthesize the appearance of a near-regular texture and purposefully control its regularity.

Keywords: near-regular texture, deformation field, texture analysis, texture replacement, texture synthesis, texture manipulation

1 Introduction

Near-regular textures are ubiquitous. They can be readily observed in man-made environments: buildings, wallpapers, floors, tiles, windows, fabric, pottery and decorative arts; and in nature: the arrangement of atoms, honeycomb, animal fur, gait patterns, feathers, leaves, waves of the ocean, and patterns of sand dunes. To simulate the real world on computers faithfully, near-regular textures deserve special attention. The commonality behind the varied appearance of near-regular patterns is their strong tendency towards regularity or symmetry, even though the regularity is often imperfectly presented and intertwined with stochastic signals and random noise.

*e-mail:{yanxi,wclin,jhhays}@cs.cmu.edu

We view a near-regular texture (NRT) as a statistical distortion of a regular, wallpaper-like congruent tiling, possibly with individual variations in tile shape, size, color and lighting. Near-regular textures can depart from regular tiling along different axes of appearance, and thus could have (1) a regular structural layout but irregular color appearance in individual tiles (top-left input texture in Figure 1); (2) distorted spatial layout but topologically regular alterations in color (bottom-left texture input in Figure 2); or (3) deviations from regularity in both structural placement and color intensity (bottom-left texture input in Figure 1). We call these **Type I, II** and **III** near-regular textures respectively (Table 1).

Table 1: A Categorization of Near-regular Textures

Type	**G**eometry	**C**olor	Symbols
0	**R**egular	**R**egular	GRCR
I	**R**egular	**I**rregular	GRCI
II	**I**rregular	**R**egular	GICR
III	**I**rregular	**I**rregular	GICI

Using regular tiling [Grünbaum and Shephard 1987] as our anchor point, we are able to define and capture these statistical departures from regularity using a multi-modal (geometry, lighting, color), multi-dimensional mapping, which we call a *deformation field*. Whereas Liu and Lin [2003] treat the deformation field as a geometric deformation of 2D patterns, we extend the concept of the deformation field and its treatment in several ways in this paper: (1) **Multi-modal deformation fields**: extending the concept of "deformation field" to include color and lighting as well as geometry so that we can treat NRT on curved surfaces (Figure 1 bottom); (2) **Deformation field duality**: treating a multi-modal deformation field both as a functional mapping and as a texture that is subject to analysis, synthesis and manipulation; (3) **Deformation fields analogies**: capturing and synthesizing the associative appearance of geometry and lighting deformation fields; (4) **Deformation field manipulation**: actively controlling the magnitudes of the deformation fields such that the regularity of the output texture can vary according to users' specifications (Top of Figure 1).

Our work is composed of two major components. The first is analysis of near-regular textures (NRT), where we study how to define,

represent and extract three different types of deformation fields that relate near-regular textures to their regular counterparts (Section 3). User assistance is needed to initiate this process by identifying a coarse lattice structure (Section 3.1). The second component is synthesis and manipulation of NRT through multi-model deformation fields (Section 4). These two components are connected by the idea of deformation field duality: treating a deformation field both as a function that acts on a texture and as a texture that is acted upon.

2 Related Work

Texture has long been a fascinating topic in human perception. Rao and Lohse [1993] show that regularity plays an important role in human texture perception. Texture synthesis has been an active research area in computer graphics for many years. *Procedural texture synthesis* [Perlin 1985; Witkin and Kass 1991; Turk 1991; Fleischer et al. 1995; Worley 1996; Walter and Fournier 1998] is based on generative models of texture. Many near-regular textures can be generated using this method. However, the cost of model-specific parameter tuning is a limiting factor for covering a larger set of textures. On the other hand, the *sample-based* approach does not require any previous texture model, yet has the capability of reproducing textures with similar appearance as the input sample [Heeger and Bergen 1995]. Existing work on image sample-based texture synthesis has achieved impressive results for a variety of textures [De Bonet 1997; Efros and Leung 1999; Ashikhmin 2001; Efros and Freeman 2001; Hsu and Wilson 1998; Wei and Levoy 2000; Hertzmann et al. 2001; Xu et al. 2001; Liang et al. 2001]. These texture-synthesis algorithms share a common theme of local neighborhood-based statistical analysis. In particular, non-parametric estimation of textures has become popular [Efros and Leung 1999; Wei and Levoy 2000; Efros and Freeman 2001; Liang et al. 2001; Kwatra et al. 2003] due to their simplicity in implementation, fast speed [Wei and Levoy 2000; Liang et al. 2001] and the ability to cover a large variety of textures.

The class of textures that yield good results for texture-synthesis algorithms remains an open question. Lin et al. [2004] compare several texture-synthesis algorithms. Their results show that general-purpose synthesis algorithms fail to preserve the structural regularity on more than 40% of the tested Type I NRT samples. An example of a synthesized brick wall texture is shown in Figure 2. Figure 2 also shows a synthesized Type II NRT sample where the color pattern regularity of the input texture is not preserved. These results demonstrate that faithful near-regular texture synthesis remains a challenging problem.

Some effort has been made for faithful texture synthesis of NRT to preserve both regularity and randomness of its appearance. Liu et al. [2004b] propose a tiling-based synthesis algorithm aimed at Type I near-regular textures only. The method is composed of two steps: (1) identifying the underlying lattice of the input texture either automatically or by user selection of two translation vectors, thus providing crucial information on the size, shape and orientation of a canonical tile of the input texture. (2) using a synthesis process similar to [Efros and Freeman 2001] except that customized tiles are matched on or around their lattice points.

At least two other pieces of work in computer graphics also use the idea of tiling. Cohen et al. [2003] use Wang Tiles for fast on-line texture synthesis where local boundary conditions between tiles are considered but global near-regularity is not addressed. The goal of Escherization [2000] is to produce a regular pattern with translational symmetry by generating tiling boundaries from a given closed planar contour.

The texture transfer problem described in Image Analogies [Hertzmann et al. 2001] and Texture Quilting [Efros and Freeman 2001] is related to our effort in this paper on replacement of near-regular

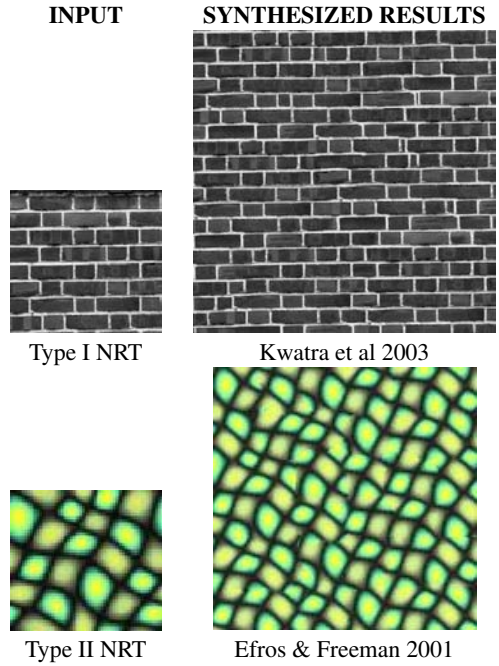


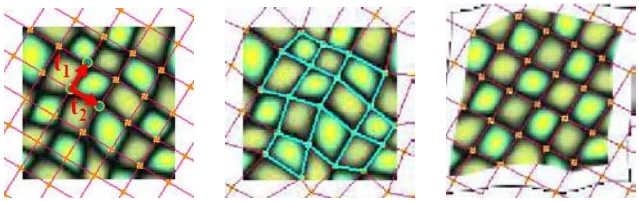
Figure 2: Two examples where near-regular texture (NRT) regularity is **not** preserved in the synthesized results. Top: The Type I input texture's geometric regularity (small-large-small-large brick pattern) is not faithfully preserved in the synthesized texture. Bottom: The Type II input texture's color alternation (yellow-green-yellow-green color pattern) is not faithfully preserved in the synthesized texture.

textures. The difference is (1) we explicitly treat both distorted geometry and lighting effects as deformation fields so that we can synthesize the deformation fields separately or jointly (Section 4), while they model data association in intensity only; and (2) our method preserves the implied underlying (distorted) regular tiling of a near regular texture, rather than its local appearance alone.

Tsin et al. [2001] proposed a texture replacement algorithm that can replace regular texture patterns on a plane. Important assumptions of the algorithm include: the surface is planar, the texture is Type I NRT and a color/intensity model of the texture can be constructed by sampling multiple texture elements from a "normal" lighting region. This model is used to detect the lighting changes and non-texture regions across the image. We have used a similar lighting and color model in our work (Section 3.2). The major difference is that we do not assume planar surfaces nor Type I NRT.

Oh et al. [2001] decouple illuminance and texture using a non-linear filtering technique. They assume that large-scale variations are due to lighting and small scale details are due to texture, which is a reasonable assumption without any knowledge of the texture. Our work differs from the work of Oh et al. in that we take advantage of a given canonical tile on the near-regular texture, and can thus extract finer shadows on sharply curved surfaces. Median filtering is used for local smoothing in our method to mitigate falsely detected lighting effects. On the other hand, since we do not have a 3D model, we can not change views or light sources. Instead of using depth information for 3D modeling, our method solely depends on characterizing relative deformations from a well-defined regularity, be it geometry, lighting or color.

3 Near-Regular Texture Analysis



(a) (b) (c)

Figure 3: (a) Regular lattice placed over the input texture based on user input of two translation vectors $\vec{t}_1, \vec{t}_2$. (b) Near-regular lattice resulting from user adjustment of lattice points. (c) The straightened lattice.

Regular textures are defined in this paper as wallpaper-like, congruent 2D tiling whose structural regularity can be completely characterized by the 17 wallpaper groups [Grünbaum and Shephard 1987; Schattschneider 1978]. The underlying lattice structure of each regular texture can thus be represented and generated by a pair of linearly independent translations $\vec{t}_1, \vec{t}_2$. Under the translational subgroup of each 2D regular texture p_r, the smallest bounded region that produces simultaneously a covering (no gaps) and a packing (no overlaps) of the texture pattern on the 2D plane is called a *tile* [Grünbaum and Shephard 1987]. A *near-regular* texture is defined as $p = d(p_r)$ where $d = d_{geo} \times d_{light} \times d_{color}$ is a pixel-wise mapping representing the composition of geometric transformations, lighting changes and color alterations, respectively.

3.1 Geometric Deformation Field

For any near-regular texture p, there exists an underlying lattice L that is a geometric distortion of a regular lattice L_r from a regular texture p_r. Though there may be many potential regular lattices that a near-regular lattice L can deform to, there is a well-defined regular lattice L_r that has a minimum distance from L [Kazhdan et al. 2002; Liu et al. 2004a]. The *Geometric deformation field* d_{geo} is defined as a function that maps L to L_r **and** warps all the input texture pixels to a geometrically regular texture.

The process of finding the geometric deformation field d_{geo} is as follows:

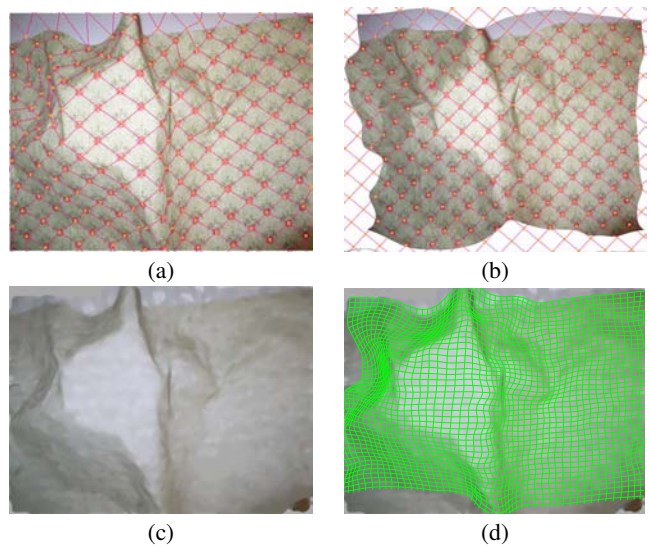
Step 1. The user gives a pair of lattice generating vectors, $\vec{t}_1, \vec{t}_2$, by clicking 3 points on p;
Step 2. The computer overlays an extrapolated 2D lattice with generators $\vec{t}_1, \vec{t}_2$ on the input texture p (Figure 3 (a));
Step 3. The user adjusts some of the misplaced lattice points to form the distorted lattice L by dragging each lattice point on the computer screen (Figure 3(b)); We have carried out a small user study on ten users, 7 of them were novices. The mean time $\pm$ one standard deviation to complete this step was 59 ± 6 seconds, 87 ± 11 seconds and 18 ± 3.3 minutes for the brick-wall (Figure 1), snake skin (Figure 3) and the cloth (Figure 4) textures, respectively.
Step 4. The computer takes the user specified L and automatically finds L_r, a mapping between L and L_r, and a pixel-wise warping function d_{geo} between p and p_r. To determine the regular lattice L_r computationally, we formulate the process as a minimization problem:

$$\min_{\|\vec{t}_1\|,\|\vec{t}_2\|,\theta} E = \sum_{i=1}^{N_i}(l_i - \|\vec{t}_1\|)^2 + \sum_{j=1}^{N_j}(l_j - \|\vec{t}_2\|)^2 + \sum_{k=1}^{N_k}(l_k - \|\vec{t}_1 + \vec{t}_2\|)^2 + \sum_{m=1}^{N_m}(l_m - \|\vec{t}_1 - \vec{t}_2\|)^2 \quad (1)$$

where l_i, l_j, l_k, and l_m, are the lengths of the links in lattice L 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 respectively, and 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 $\vec{t}_1 + \vec{t}_2$. Similar mesh models are used in [Terzopoulos and Vasilescu 1991] and [Liu and Lin 2003].

We use the multilevel free-form deformation (MFFD) algorithm proposed in [Lee et al. 1995] to capture a 1-to-1 warping field. Using corresponding lattice points between L and L_r as control points, this method generates a bijective warping function, which induces a pixel-wise warping function between the input texture p and the Type I NRT p_r (Figure 3(c)). This bijective warping function is the geometric deformation field d_{geo}.

3.2 Lighting Deformation Field



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

Figure 4: (a) Input texture (Figure 9) with overlaid, user-specified lattice (b) Straightened lattice. (c) Extracted light map. (d) Overlay of geometry and lighting deformation fields.

Relating a near-regular texture to its regular origin has many computational benefits. One is the ability to estimate lighting effects on the texture. By indicating a portion of the texture that is regular and *canonically* lighted, we are able to develop an algorithm that simultaneously finds the lighting map (shadows, occlusion) while segmenting out the texture region in the image.

Recall that the geometric deformation field defined in Section 3.1 is capable of warping an arbitrary near-regular texture back to a Type I texture. Our algorithm for extracting the lighting deformation field works as follows: (1) straighten the lattice of the input NRT using the geometric deformation field d_{geo}; (2) apply Tsin et al. [2001]'s algorithm for lighting map extraction in the plane; and (3) apply the inverse geometric deformation field to map the lighting deformation field back to the original input texture. This method works well for surface textures with minimal occlusion.

Without knowing the surface geometry, straightening the lattice alone does not truly flatten a folded texture surface to a planar surface, especially in regions where geometric distortions are severe (such as sharp ridges of a cloth). Therefore, some tile patterns may differ significantly in shape and size in the straightened texture. These residual geometric distortions need to be removed, otherwise they will be mistakenly identified as part of the lighting effects. To

solve this problem, we use an affine registration algorithm between each tile and a reference tile, allowing local deformations to be corrected. The light map is then computed and warped back to the original lattice space (before straightening the lattice). We also apply a median filter to the light map to further reduce falsely detected lighting effects. The result on a piece of cloth with a substantial amount of folding is shown in Figure 4.

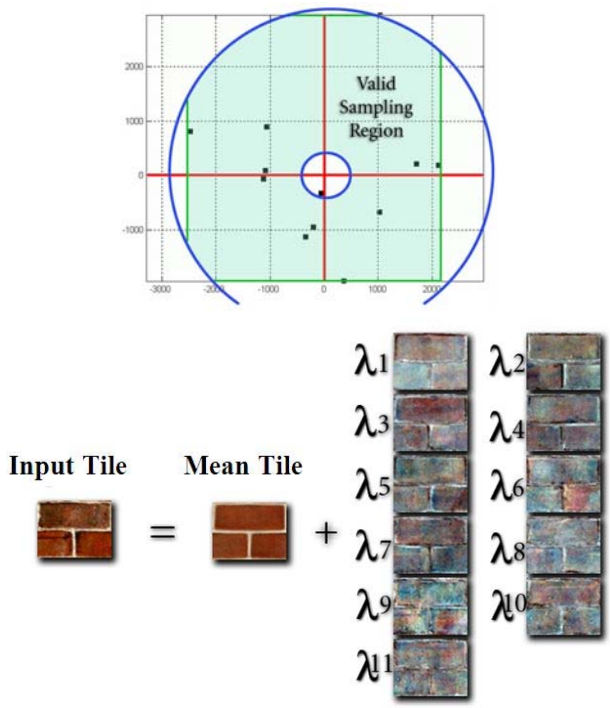


Figure 5: Principal component analysis of tiles from the brick texture in Figure 1. Top: The plot shows tile distributions in the space spanned by the top two principal components. Bottom: Each input tile can be represented as a linear combination of its mean tile with the top 11 PCA bases. The blue colors on PCA bases reflect negative values.

3.3 Color Deformation Field

The implied regularity in NRT suggests that tiles with varied appearances are actually generated by a similar process. Therefore it is meaningful to explore the generative process as well as model the color space variations between tiles. If we attempt to model the space of color deformations naively, the degrees of freedom (DOF) are prohibitive: a 100×100 pixel RGB tile has 30,000 dimensions. In reality, there are far fewer independent dimensions along which a tile's color can vary. We can estimate these reduced data dimensions automatically in terms of their orthogonal bases using principal component analysis (PCA).

PCA is a classical statistical method designed to capture the variance in a dataset [Morrison 1990]. Given a set of 2D tiles, we rearrange them into a 2D matrix with each column representing a tile and each row representing the corresponding pixels from each tile. The mean tile is the column average of this matrix. Using the method described in [Turk and Pentland 1991], we compute eigenvectors of this input tile stack. Each eigenvector can be thought of as a basis tile. The original tiles can then be reconstructed as the sum of the mean tile and some linear combination of basis tiles (Figure 5). This set of PCA bases and associated coefficients forms the color deformation field for the input texture, capturing departures from color regularity, represented in this case by the mean tile.

Figure 5 shows a sample result of PCA analysis on a brick wall texture sample (top-left input texture in Figure 1). The distribution of input sample tiles in the space formed by the top two principal components is especially relevant, since it demonstrates the range of the most predominant color irregularity in the input texture samples. The further away from the origin (mean tile) we sample, the larger the departures from color regularity. From this analysis, we are able to sample the PCA space in a controlled manner: either sample near the input distribution for faithful reproduction of the input texture, or away from the input in the PCA space for generating textures of varied appearance. This topic will be explored further in Section 4.

3.4 A Pair of Regularity Measurements

Using the same symbolic and geometric convention for a lattice structure as in Equation (1) of Section 3.1, we define a pair of quantitative measurements for near-regular textures. This pair of statistical measures characterizes the regularity of a near-regular texture from two different perspectives, G for **g**eometric structure and A for **a**ppearance of color intensity variation, with the understanding that the color variation can be caused by both inherent tile color differences and external lighting changes.

Geometric regularity:

$$G = \sum_{i=1}^{N_i} \frac{(l_i - \|\vec{t}_1\|)^2}{\|\vec{t}_1\|^2} + \sum_{j=1}^{N_j} \frac{(l_j - \|\vec{t}_2\|)^2}{\|\vec{t}_2\|^2} + \sum_{k=1}^{N_k} \frac{(l_k - \|\vec{t}_1 + \vec{t}_2\|)^2}{\|\vec{t}_1 + \vec{t}_2\|^2} + \sum_{m=1}^{N_m} \frac{(l_m - \|\vec{t}_1 - \vec{t}_2\|)^2}{\|\vec{t}_1 - \vec{t}_2\|^2}$$

Appearance regularity: $A = \frac{1}{m}\sum_{i=1}^{m} std([T_1(i), T_2(i), ..., T_n(i)])$ where T_i is a reshaped tile (a column vector) of a Type I texture, m is the number of pixels within a tile, and n is the number of tiles.

These two measures provide a more precise vocabulary to define different types of textures (Table 1). Strictly speaking, we call a texture **regular** if $G = A = 0$; **Type I** near-regular if $G = 0$ and $A > 0$; **Type II** if $G > 0$ and $A = 0$; and **Type III** if $G > 0$ and $A > 0$. In reality, however, these different types of textures form a continuous texture spectrum where it is difficult to find clear-cut boundaries. However, these measurements provide a quantitative guideline for texture regularity classification (Figure 6) and texture synthesis results evaluation (Figure 8).

Table 2 presents a statistical evaluation of the computed regularity scores from the user-assisted extracted lattices (Section 3.1). The consistency of the mean and the median for each measure shows that the distribution of the results is balanced and stable. The relatively low standard deviations show the consistency of lattice extracted by different users.

4 Near-Regular Texture Manipulation

In addition to modeling deformation fields explicitly, another central idea behind our approach is the realization of the *duality* of a deformation field. On the one hand, a deformation field is a multimodal, multidimensional mapping that warps a regular texture into its near-regular form or vice versa (Section 3). On the other hand, **a deformation field itself is a texture** that can be subjected to texture synthesis and manipulation, as demonstrated in the rest of this paper. Our approach to manipulation of near-regular textures is achieved through manipulation of their deformation fields.

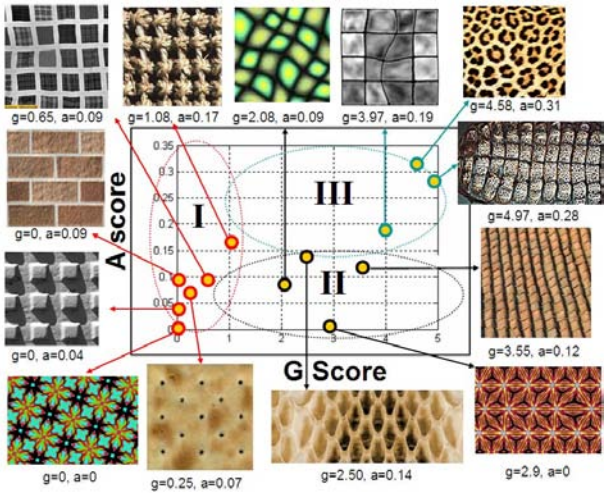


Figure 6: Textures plotted using our geometric (G) and appearance (A) regularity measures defined in Section 3.4. The higher the scores, the more irregular the texture. Notional classification regions for Type I,II and III near-regular textures are indicated.

Table 2: We report the quality of extracted lattices on three different textures from ten users. 7 out of the 10 are first time users. We compute median and the robust standard deviation (std) for each set of measurements. The low stds show consistency in lattices extracted by different users.

G-A SCORES	mean	median	robust std
Brick wall G	0.25	0.27	0.05
Brick wall A	0.12	0.12	0.01
Snake skin G	2.6	2.5	0.37
Snake skin A	0.1	0.1	0.01
Cloth G	11.6	8.9	1
Cloth A	0.11	0.11	0

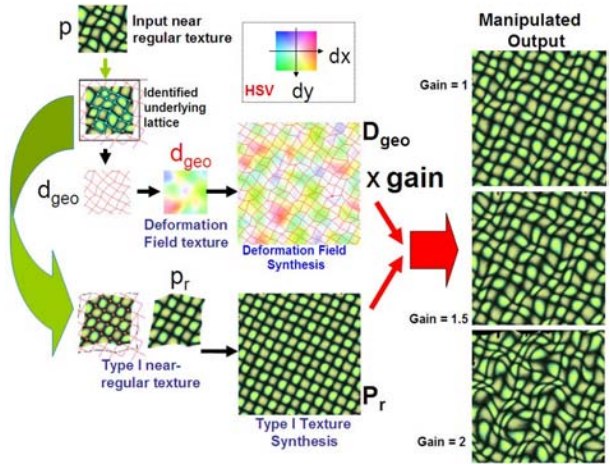


Figure 7: Overview of geometric deformation field synthesis and manipulation. Starting with an input near-regular texture (NRT) p, its geometrically regular version p_r is obtained by "straightening out" its underlying lattice L into the nearest regular lattice such that $L = d_{geo}(L_r)$ and $p = d_{geo}(p_r)$. p_r is a Type I NRT by definition and d_{geo}, a geometric deformation field treated as a texture, is usually not a regular or near-regular texture. We synthesize p_r to P_r using a Type I NRT synthesis algorithm developed in [Liu *et al* 2004b], and synthesize the deformation field d_{geo} to D_{geo} based on the algorithm proposed in [Efros and Leung 1999]. Finally, the NRT manipulation is achieved by applying different gains to D_{geo}. When gain = 1, the output texture is most similar to the input texture.

4.1 Geometry Deformation Field Manipulation

From the geometric deformation field analysis in Section 3.1, we obtain a mesh-like bijective mapping representing the deformation field d_{geo} of the input near-regular texture p, where $d_{geo}(p) = p_r$ (Figure 7). Using the deformation field duality, we visualize the geometric deformation field d_{geo} in the hue, saturation, and value (HSV) color space. The hue and saturation components represent the direction angle and the length of a 2D displacement vector, respectively, and the value component is set to 1. A pure white color means zero movement, and a red color means a movement in the positive x direction (Figure 7).

In this paper we extend the idea of treating a deformation field as a texture beyond synthesis [Liu and Lin 2003] to manipulation. The synthesized deformation field D_{geo} provides a parameterized approximation of geometric regularity similar to the input sample texture. We can now multiply the pixel-wise magnitude of the deformation field by a *gain factor*: when $gain = 1$ the deformation field is faithful to the input; when $|gain| < 1$ or $|gain| > 1$ the magnitude of the deformation field departs from the input texture's regularity.

One can observe the consistency between the gain factor, the regularity measures, and the visual effects in the texture manipulation results of Figure 7. Figure 8 compares our texture-synthesis result with four other texture-synthesis algorithms [Efros and Leung 1999; Efros and Freeman 2001; Wei and Levoy 2000; Kwatra et al. 2003]. One can observe on close inspection that only our algorithm preserves the color alterations in the input texture sample (checkerboard-like yellow-green pattern). In the 2D regularity space (geometry versus color variations) defined in Section 3.4, our synthesis result is the closest to the input texture (Figure 8).

4.2 Texture Replacement

Once both the geometric and lighting deformation fields of a given near-regular texture are extracted as described in Section 3.2, we can apply this pair of deformation fields on a completely new texture P, regardless of whether P is regular or not. The goal is to create the same visual effects in appearance using a different texture, given only a single input image. Figures 1 (bottom row) and 9 show the results from our texture replacement algorithm.

Figure 13 shows another texture replacement example from an outdoor photo. Figure 10 explains how the underlying lattice of the outdoor scene is extracted. This building photo, as a texture, has only three large vertical tiles (each is three-floors high). These are tiles with curved boundaries, captured by the user interface using cubic spline interpolation. The control points can be adjusted by the user for a better fit. The application of this type of manipulation includes architecture, interior design, and cloth manipulation.

4.3 Deformation Field Analogy

Given a near-regular texture placed in a realistic environment, geometry and lighting deformations of the texture are strongly associated. We have shown that a joint representation of these two deformation fields can be captured under our framework (Section 3.2). To achieve realistic synthesis of a near-regular texture, we need to

Figure 9: The rose patterned cloth (top-left) is the input image. The rest is texture replacement results of our method. From the input texture and its associated geometric lattice, our method computes the geometric and lighting deformation fields as shown in Figure 4.

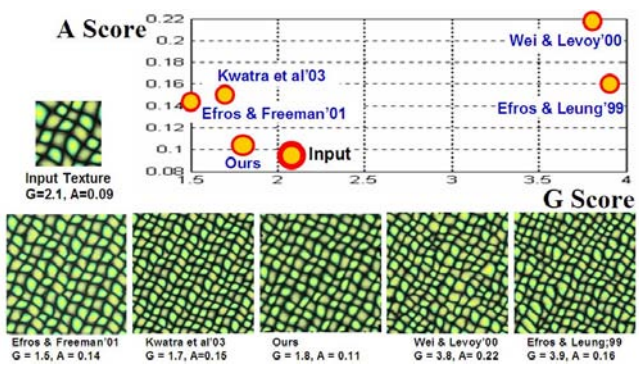


Figure 8: One of our texture-synthesis results compared with other methods. One can observe on close inspection the regular alterations in color (... yellow, green, yellow, green, ...) of the input texture. This regularity is more faithfully preserved by our texture synthesis method (Figure 7, when gain =1). In the quantified geometry-appearance regularity space represented by the *G-A* regularity scores (Section 3.4), our output is the closest to the input texture.

Figure 10: Three complete vertical tiles, each tile is three-floors high, are identified in an outdoor photo. Internal correspondences are added (each large tile is divided into three portions) for an extra assurance on global as well as local geometric consistency during deformation. Less than 4 minutes are needed for the lattice extraction process, initiated by the user (Section 3.1).

synthesize the two deformation fields simultaneously. Given an image of a near-regular texture, we are able to distill its geometry and lighting deformation field pair (Section 3.2), let us call them A and A'. Based on our deformation field duality, both the geometric deformation field and the lighting field can now be treated as a pair of directly related textures. Without loss of generality, given $A = d_{geo}$ (Section 3.1), the inferenced $A' = d_{light}$ (Section 3.2) and a faithfully synthesized $B = D_{geo}$ from d_{geo} (Figure 7), we have a well defined "deformation field analogies" problem at hand: that is to find $B' = D_{light}$ that bears the same relation to B as A' to A (Figure 11).

There is an inherent symmetry in this problem, that is, we can either (1) synthesize A to B then look for B' using analogy, or (2) synthesize A' to B' then look for B using analogy (Figure 11). Theoretically, following either route we should be able to achieve a new geometry/lighting deformation field pair as the result of deformation field analogy. In our experiments, the first route is more favorable: we apply a geometric deformation field synthesis algorithm [Liu and Lin 2003] to generate D_{geo} first, then apply image analogies to synthesize the lighting deformation field corresponding to D_{geo}. The geometric deformation field provides the necessary causal constraints on the lighting deformation field. See Figure 12 for an example of deformation field analogies.

4.4 Texture Regularity Manipulation

From PCA analysis of the input texture (Section 3.3), we obtain a compact representation of the texture color distribution. This

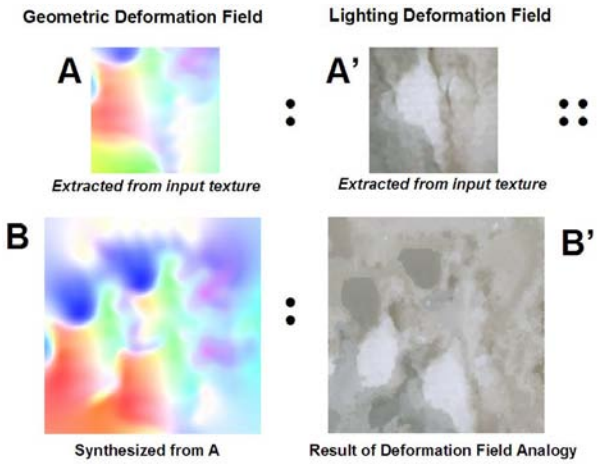


Figure 11: Deformation field analogies, based on teh idea of image analogies [Hertzmann et al 2001]. We synthesize the geometric deformation field B from A using Efros and Leung's [1999] algorithm, then extract the lighting deformation field A' (Section 3.2), and finally use the image analogies approach to find lighting deformation field B'. We experimented with different parameter settings of the image analogy algorithm, and found that a smaller coherence parameter and larger neighborhood width provide better results in terms of the smoothness of the lighting deformation field. The best result we obtained is coherence parameter k=0, and neighborhood width n = 25.

PCA model provides us with several "knobs" to control the appearance regularity of the synthesized texture. The tiles provided to the texture synthesis algorithm are sampled uniformly within a hyper-volume in the space of the PCA coefficients. This sampling space is bounded by the magnitude of the PCA coefficients computed from the input texture tiles (Figure 5). The dimensionality of this hyper-volume is equal to the number of PCA bases chosen to capture a desirable amount of the variance in the data for reproduction. In the case of the brick wall sample in Figure 1, for example, 10 bases are used. Even though the dimensionality of the resulting PCA space is only one less than the number of the input tiles in this case, the reduction in the DOFs from the original color-pixel tile space is enormous (more than 10 thousand-fold reduction) with the additional benefit of having a generative PCA model.

This generative model allows us to produce many new tiles that are not in the input texture. We are coupling a statistical texture synthesis method, used to create individual tiles, with our image based near-regular texture synthesis method [Liu et al. 2004b]. An arbitrarily large number of new tiles can be generated during the sampling process, allowing the texture synthesis algorithm to find tiles that can match seamlessly together. In our video presentation, we show a synthesized brick wall texture using 1400 unique, synthetic tiles (both as a stand alone synthesis result and as a texture replaced onto a building), where both the underlying regularity and color variations of the input texture are faithfully preserved.

Figure 1 shows the effect of increasing color gain and geometry gain for the brick texture. The color gain is a coefficient to scale the range from which new tiles can be sampled. A color gain of 0 results in sampling the mean tile only, and color gain of 1 is sampling tiles consistent with the input. A larger gain allows large coefficients for the PCA bases, and thus larger amounts of irregularity in the sampled tiles. One can observe a migration of the synthesized textures from regular to irregular along both color and geometry axes when raising the gain values (from 0 to 2).

5 Discussion and Conclusion

In this paper, we view near-regular textures as statistical departures from a regular texture along different dimensions. Using regular tiling as our anchor point, the concept of a deformation field becomes natural, intuitive and versatile. More importantly, we show that with reasonable user assistance, the computation of deformation fields is feasible and effective for many different applications. Furthermore, the deformation field duality idea facilitates the computational advantage of applying existing texture synthesis and analogy algorithms to functions, treated as textures. Using our proposed formulation, we are able to construct simple parametric models from input texture samples to purposefully manipulate the regularity of a variety of real-world near-regular textures.

Automatic lattice generation for a given distorted near-regular texture is an unsolved segmentation problem. In our current implementation, the following user assistance is required: (1) identifying the underlying distorted lattice of an input near regular texture for geometric deformation field analysis; (2) circling a small, canonically lighted region for lighting map extraction; and (3) controlling two slider bars to set the levels of geometry and color regularity (gain). The first two interventions are necessary to initiate the automatic analysis and synthesis process, but are done only once for a given texture. Our user study on ten subjects with three types of textures shows that (a) 1 to 20 minutes are needed for lattice adjustment, depending on the regularity of the input texture; and (b) the texture geometry and appearance regularity measurements, computed from the lattices extracted by different users, have small variance. The third intervention allows the user to interactively manipulate the desired output textures, and thus can be adjusted multiple times.

Our exploration of regularity-sensitive texture analysis and manipulation algorithms only touches the tip of the near-regular texture iceberg. Many challenging problems remain. For example, how to treat deformation fields when input near-regular textures have self occlusion remains an open problem. One important limitation of the lighting deformation field is that only those attached shadows, directly caused by surface geometry, can be captured, modeled and synthesized using image analogies. Cast shadows from non-local regions may cause un-reasonable (unexplainable) synthesized results. Modeling texture space with PCA is also limited by two major factors. First, PCA is sensitive to the proper registration of input tiles. That is, PCA is only possible on tiles that are geometrically aligned; however, if the tiles have complicated internal structure that can not be aligned under global transformations, PCA may produce unsatisfactory, blurred results. Second, in general, our assumption of a uniform space of legal tiles across the PCA coefficients is not true. However, given the limited number of tiles in the input samples (on the order of 10), there are few alternatives. If the number of input tiles were orders of magnitude higher, attempts could be made to more precisely model the manifold on which valid tiles lie.

6 Acknowledgement

We would like to thank Yanghai Tsin, Chenyu Wu and Aaron Hertzmann for sharing their code. This paper benefited greatly from constructive suggestions given by the members of CMU and Georgia Tech graphics groups. In particular, we thank Vivek Kwatra for providing the graph-cut texture synthesis result in Figure 2. We thank anonymous reviewers, especially referee #1, for their valuable comments, and Jing Xiao and Robert T. Collins for proofreading our paper. We also thank volunteers from CMU for testing our user interface. This research is supported in part by an NSF grant IIS-0099597.

Figure 12: Given a near-regular texture (Type III), we show its texture synthesis and then texture replacement results.

Figure 13: Texture replacement on an outdoor photo (top left).

References

ASHIKHMIN, M. 2001. Synthesizing natural textures. In *ACM Symposium on Interactive 3D Graphics*, 217–226.

COHEN, M. F., SHADE, J., HILLER, S., AND DEUSSEN, O. 2003. Wang tiles for image and texture generation. *ACM Transactions on Graphics 22*, 3, 287–294.

DE BONET, J. 1997. Multiresolution sampling procedure for analysis and synthesis of texture images. In *Proceedings of SIGGRAPH 97*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 361–368.

EFROS, A., AND FREEMAN, W. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 341–346.

EFROS, A., AND LEUNG, T. 1999. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision*, 1033–1038.

FLEISCHER, K., LAIDLAW, D., CURRIN, B., AND BARR, A. 1995. Cellular texture generation. In *Proceedings of SIGGRAPH 95*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 239–248.

GRÜNBAUM, B., AND SHEPHARD, G. 1987. *Tilings and Patterns*. W.H. Freeman and Company, New York.

HEEGER, D. J., AND BERGEN, J. 1995. Pyramid-based texture analysis/synthesis. In *Proceedings of SIGGRAPH 95*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 229–238.

HERTZMANN, A., JACOBS, C., OLIVER, N., CURLESS, B., AND SALESIN, D. 2001. Image analogies. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 327–340.

HSU, T., AND WILSON, R. 1998. A two-component model of texture for analysis and synthesis. *IEEE Trans. on Image Processing 7*, 10 (October), 1466–1476.

KAPLAN, C., AND SALESIN, D. 2000. Escherization. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 499–510.

KAZHDAN, M., CHAZELLE, B., DOBKIN, D., FINKELSTEIN, A., AND T., F. 2002. A reflective symmetry descriptor. In *Proceedings of Seventh European Conference on Computer Vision*, Springer, 642–656.

KWATRA, V., SCHÖDL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics 22*, 3, 277–286.

LEE, S., CHWA, K., SHIN, S., AND WOLBERG, G. 1995. Image metamorphosis using snakes and free-form deformations. In *Proceedings of ACM SIGGRAPH 1995*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 439–448.

LIANG, L., LIU, C., XU, Y.-Q., GUO, B., AND SHUM, H.-Y. 2001. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics 20*, 3, 127–150.

LIN, W., HAYS, J., WU, C., KWATRA, V., AND LIU, Y. 2004. A comparison study of four texture synthesis algorithms on regular and near-regular textures. Tech. Rep. CMU-RI-TR-04-01, The Robotics Institute, Carnegie Mellon University.

LIU, Y., AND LIN, W. 2003. Deformable texture: the irregular-regular-irregular cycle. In *Texture 2003, The 3rd International Workshop on Texture Analysis and Synthesis*, 65–70.

LIU, Y., COLLINS, R., AND TSIN, Y. 2004. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transaction on Pattern Analysis and Machine Intelligence 26*, 3, 354–371.

LIU, Y., TSIN, Y., AND LIN, W. 2004. The promise and perils of near-regular texture. *International Journal of Computer Vision (In Press)*.

MORRISON, D. 1990. *Multivariate Statistical Methods*. McGraw Hill College Div, ASIN 0070431876.

OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. In *Proceedings of ACM SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 433–442.

PERLIN, K. 1985. An image synthesizer. In *Computer Graphics (Proceedings of SIGGRAPH 85)*, vol. 19, ACM, 287–296.

RAO, A., AND LOHSE, G. 1993. Identifying high level features of texture perception. *CVGIP: Image Processing 55*, 218–233.

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

TERZOPOULOS, D., AND VASILESCU, M. 1991. Sampling and reconstruction with adaptive meshes. In *IEEE Computer Vision and Pattern Recognition Conference*, 70–75.

TSIN, Y., LIU, Y., AND RAMESH, V. 2001. Texture replacement in real images. In *IEEE Computer Vision and Pattern Recognition Conference*, 539–544.

TURK, M., AND PENTLAND, A. 1991. Eigenfaces for recognition. *J. of Congnitive Neuroscience 3*, 1, 71–86.

TURK, G. 1991. Generating textures on arbitrary surfaces using reaction-diffusion. In *Computer Graphics (Proceedings of SIGGRAPH 91)*, vol. 25, ACM, 289–298.

WALTER, M., AND FOURNIER, A. 1998. Clonal mosaic model for the synthesis of mammalian coat patterns. In *Graphics Interface '98*, 82–91.

WEI, L.-Y., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 479–488.

WITKIN, A., AND KASS, M. 1991. Reaction-diffusion textures. In *Computer Graphics (Proceedings of SIGGRAPH 91)*, vol. 25, ACM, 299–308.

WORLEY, S. 1996. A cellular texture basis function. In *Proceedings of SIGGRAPH 96*, ACM Press / ACM SIGGRAPH, Computer Graphics Proceedings, Annual Conference Series, ACM, 291–294.

XU, Y., ZHU, S., GUO, B., AND SHUM, H. 2001. Asymptotically admissible texture synthesis. In *International Workshop on Statistical and Computational Theories of Vision*.