EUROGRAPHICS 2005 / M. Alexa and J. Marks
(Guest Editors)

Volume 24 (*2005*), *Number 3*

Fractional Fourier Texture Masks: Guiding Near-Regular Texture Synthesis

A. Nicoll and J. Meseth and G. Müller and R. Klein

Institut für Informatik II, Universität Bonn, Germany

Abstract

Recently, the special kind of near-regular texture has drawn significant attention from researchers in the field of texture synthesis. Near-regular textures contain global regular structures that pose significant problems to the popular sample-based approaches, and irregular stochastic structures that can not be reproduced by simple tiling. Existing work tries to overcome this problem by user assisted modeling of the regular structures and then relies on regular tiling. In this paper we use the concept of fractional Fourier analysis to perform a fully automatic separation of the global regular structure from the irregular structure. The actual synthesis is performed by generating a fractional Fourier texture mask from the extracted global regular structure which is used to guide the synthesis of irregular texture details. Our new method allows for automatic and efficient synthesis of a wide range of near-regular textures preserving their regular structures and faithfully reproducing their stochastic elements.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – Color, shading, shadowing, and texture, I.4.7 [Image Processing and Computer Vison]: Feature Measurement – Texture

1. Introduction

During the last years, the area of texture synthesis has been devoted significant work from researchers in computer graphics and computer vision. The developed algorithms not only achieve impressive synthesis results for various kinds of textures but also provide general methods applicable to other interesting fields like image restoration or geometry completion. Nevertheless, not all kinds of textures can currently be handled in a pleasing manner. Especially textures referred to as near-regular textures [LTL05] that contain global, regular structures as well as stochastic deviations from regularity are still very difficult if not impossible to synthesize faithfully. Examples of this type of texture are frequently found in the real world: most textiles (e.g. used for clothing, furniture, or car interiors) and construction elements (walls, floors, grid structures, corrugated sheet roofs) fall into this category, and even natural objects like water waves, feathers and fur.

The specific difficulty of these textures originates from the strong mixture of regular patterns which govern global structure, and the subtle, yet characteristic deviation from this regularity. Since the regular patterns may be of arbitrary scale, existing sample-based synthesis algorithms fail to faithfully reproduce these textures due to restrictions of neighborhood sizes and lack of adequate distance functions.

In this paper we introduce the concept of fractional Fourier texture masks (procedural textures for the regular parts), which are derived by extracting the regular structure from a texture using fractional Fourier analysis and "enlarging" it to a desired size. These masks are used to guide sample-based synthesis algorithms in faithful selection and placement of copied pixels or patches, effectively enforcing global, regular structure and therefore leading to drastically improved synthesis quality for near-regular textures.

Our new, robust method for synthesis of near-regular textures is composed of three major steps:

1. separation of the dominant regular structure from irregular texture detail using the fractional Fourier transform and an intensity filter,
2. synthesis of the regular structure based on the inverse fractional Fourier transform,
3. addition of irregular texture detail by extended sample-based texture synthesis algorithms.

In the following, we first review some related work in

© The Eurographics Association and Blackwell Publishing 2005. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

Section 2. Afterwards, Section 3 provides background and implementation details of our algorithms for separation and synthesis of the regular texture structure. Section 4 then shows how the resulting fractional Fourier texture masks can be used to guide existing sample-based texture algorithms to faithfully reproduce both local and global structure. After presenting and discussing our results in Section 5 we conclude and describe future directions of research in Section 6.

2. Related work

The objective of texture synthesis is to generate images that reproduce a distribution of textural features which humans perceive as a specific type of texture.

For a limited class of textures this distribution can be modeled using for example Perlin Noise [Per85] or reaction-diffusion systems [Tur91]. These types of procedural texture synthesis offer the advantage of user control and extremely compact representation.

To capture more general types of texture, statistical modeling has been adopted. Motivated by research on human texture perception, statistics of filter response vectors have been used most frequently. The actual synthesis is performed by iteratively matching statistics of a sample texture and the synthesized result. E.g., Heeger and Bergen [HB95] matched marginal histograms of filter response vectors at different spatial scales. Follow up publications [Bon97, PS00, BJEYLW01] improved upon this scheme by enforcing more complex joint statistics of filter coefficients but still fail on highly structured textures. Few publications (e.g. [ZWM98]) propose parametric texture models based on the Markov Random Field model of texture. Texture synthesis involves fitting the model to a sample texture and sampling from the resulting distribution which can be computationally very expensive and still reproduces mainly stochastic textures only.

These elaborate models are outperformed in speed, quality and applicability by simple non-parametric sampling as proposed in the seminal paper by Efros and Leung [EL99]. The method synthesizes a texture pixel by pixel by selectively copying pixels from a sample. Appropriate pixels are identified by comparing their neighborhoods whose size is the only tuning parameter of the algorithm. It can be shown that this scheme produces consistent estimates of the distribution of pixels given their neighborhood [Lev02]. This simple but powerful idea has inspired many researchers and numerous improvements of the original algorithm have been published over the years. Many of them focused on speeding up the search for similar neighborhoods using for example tree-structured vector quantization [WL00], k-coherance search [TZL*02], or jump maps [ZG02]. For textures with large structures compared to affordable neighborhood sizes, copying of whole patches instead of single pixels was advocated: Efros and Freeman [EF01] select fixed size blocks and paste them in a regular manner such that the color differences in the overlap region are minimized. A similar approach is followed by Liang et al. [LLX*01] but they apply feathering to merge overlapping patches. Kwatra et al. [KSE*03] generalize and improve these approaches by reducing the problem of combining the pixels to a minimal cut problem and allowing for blocks of arbitrary sizes. Regions of varying size are as well supported by the algorithm of Nealen and Alexa [NA03]. Despite the fact that copying large texture regions allows for preservation of large texture structures, the exact alignment of structures in the target texture often fails due to bad placement of patches.

To overcome these problems, the concept of *texture masks* was developed. These masks are either used to guide consistent [LYS01] or user-controlled [Ash01] texture placement, or to enforce that the shape of prominent features e.g. from animal fur is not broken apart during texture synthesis [ZZV*03]. We use a similar approach in the final synthesis step of our algorithm but we propose a unique and automatic analysis for generating such texture masks from near-regular textures.

These near-regular textures have been devoted attention from the community because they contain regular (tileable) structures that are not captured by local distributions, and irregular elements that are not reproduced correctly by tiling. Therefore, textures dominated by regular structures still pose problems for sample-based texture synthesis and need special attention. A first step in this direction was made by the work of Dischler et al. [DMLC02] who tried to identify elementary patches in the texture and analyzed their spatial arrangement which is then reproduced in the synthesized texture. An approach bridging the gap between regular tiling and stochastic placement of texture elements was presented by Cohen et al. [CSHD03]. They enforce boundary conditions along tiles but the placement does not account for global regular structures ranging across tiles.

A more general approach for handling regular and potentially large structures during texture synthesis was developed by Liu et al. [LCT04] and Liu and Tsin [LTL05]. They used the observation that an infinite variety of periodic patterns can be characterized by a finite number of symmetry groups to extract tileable parts of a texture that correspond to the regular structure of the texture. Tiling these textures with overlap and merging the tiles leads to very good results for most regular textures. The approach was recently extended to handle near-regular textures [LLH04] by treating them as deformations of regular textures. One drawback of this method is that extraction of tiles requires significant user intervention whereas our process is fully automatic. Another problem is that it requires at least two complete tiles to be included in the sample texture. Although this usually poses no problem, such data may be unavailable if regular structures of different scales are included in the texture (a complete tile would need to be as large as the least common multiple of the

© The Eurographics Association and Blackwell Publishing 2005.

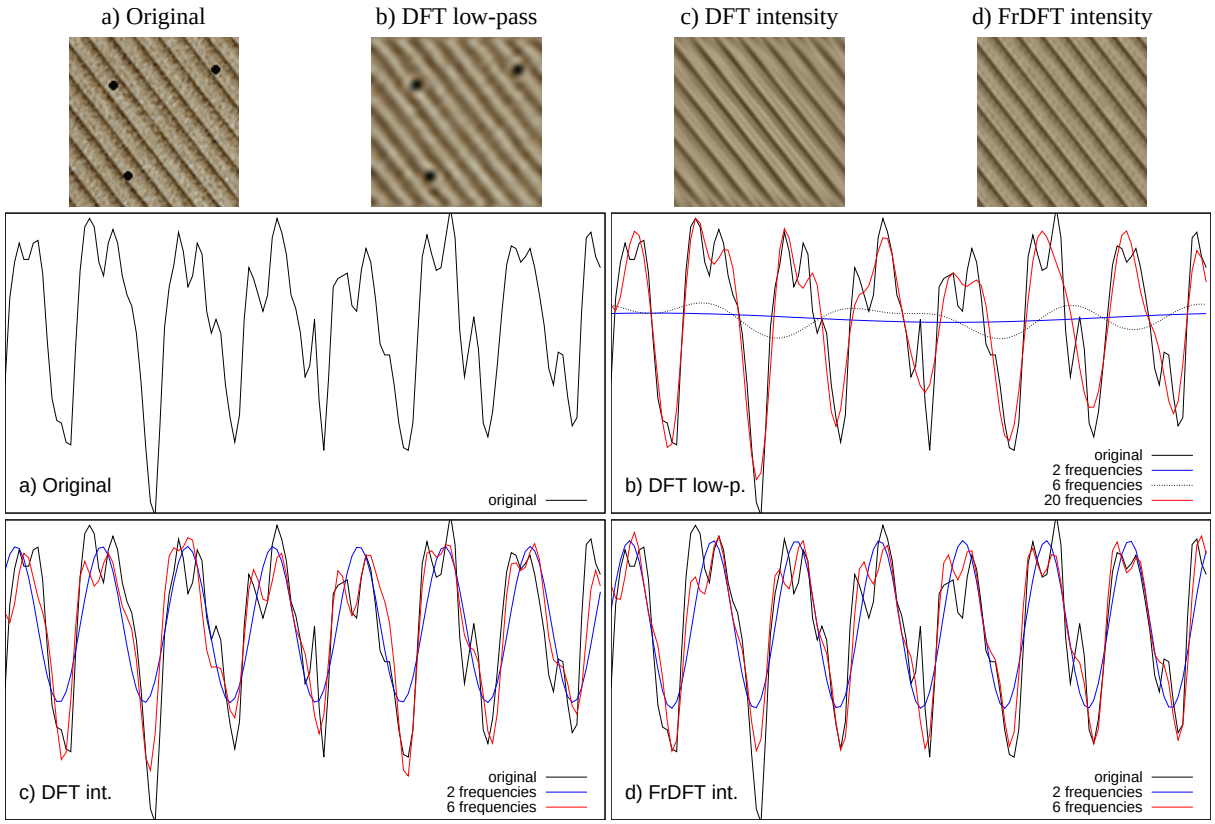

Figure 1: *Results of applying filters to a Corduroy texture samplewith black dots added at random locations. Whereas the images show the texture sample, the curves visualize the luminance values of the pixels of the first scanline only. While the lowpass filter result (b) still contains irregular structures and about 20 basis functions are required for a reasonable approximation of the 1D signal, the intensity filter (c) performs much better using just 6 frequencies. Nevertheless, the leakage error can only be eliminated applying the FrDFT intensity filter (d).*

different scales). Our approach is able to treat the different scales individually and therefore requires no such large textures. Finally, unlike our approach they neither separate regular from stochastic texture elements, which improves the quality of synthesized results, nor do they compute an explicit, compact model of the regular texture parts.

3. FrDFT-Analysis

In this section, we describe our approach for separating the regular structure, which dominates the appearance of many near-regular textures like knitware or woven textiles, from the irregular one. Our intuitive distinction between these two kinds of data is that structures are perceived as regular if they occur (at least nearly) periodically while irregular structures are distributed in a different, more complex way. Therefore, we represent a texture by periodic basis functions to determine and extract regular structures.

Such a representation can be derived by applying a discrete Fourier transformation (DFT) which represents the signal by a sum of sine and cosine functions. Given the resulting frequency spectrum, the question of which and how many frequencies are relevant for the regular part of the image signal arises. If too few frequencies are taken into account, most of the structure is smoothed out. If too many frequencies are selected, the signal itself is better reproduced but at the cost of introducing irregular structure.

One way to select the relevant frequencies is to apply a low pass filter to the image signal. In previous approaches for texture synthesis this idea is incorporated by building image pyramids and synthesizing the low-resolution levels first. Unfortunately, in general these low frequencies do not correspond to the regular structures directly, which is shown in Figure 1b.

A more appropriate selection criterion is based on two observations. First, since the regular structures in the image signal dominate the overall appearance of most near-regular textures (see e.g. the Corduroy sample in Figure 1),

© The Eurographics Association and Blackwell Publishing 2005.

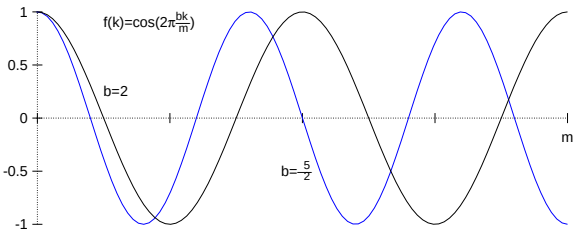

Figure 2: *Fractional frequency. Representing the perfectly regular fractional frequency $b = \frac{5}{2}$ by integer frequencies requires a large amount of basis functions, since the corresponding signal is not periodic w.r.t. the range $[0,m]$.*

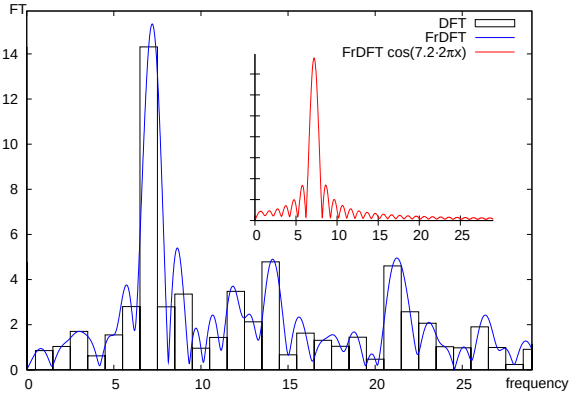

Figure 3: *Comparison of absolute DFT and FrDFT coefficients for the curve in Figure 1 after subtraction of mean value. The dominant regular structure is caused by frequency 7.2. The small plot shows the FrDFT of the function $cos(7.2 \cdot 2\pi x)$.*

they carry a much larger amount of energy than the irregular patterns. Second, while the energy contained in strongly visible irregular structures is distributed among a large number of frequencies, the main part of the energy contained in the periodic structures is carried by a few frequencies already (compare Figure 1c). We therefore select the frequencies carrying the most energy and call the selection based on this criterion an *intensity* filter.

While application of the intensity filter to the DFT of the signal yields good results already, the leakage effect of the DFT introduces artifacts along the borders of the image as illustrated in Figure 1c. This prohibits high quality synthesis of the extracted regular structure since the erroneous regions would be contained repeatedly in the synthesized image. The reason for the existance of leakage problems is shown in Figure 2, where both functions represent perfectly periodic signals restricted to the interval $[0,m]$. With respect to the given interval, only the function with $b = 2$ is periodic, the one with $b = \frac{5}{2}$ is not. Applying the intensity filter to the DFT always yields a signal periodic w.r.t. the given interval, which prohibits a faithful representation of the perfectly regular frequency $b = \frac{5}{2}$. For a texture the restriction of the signal to such a window is usually defined implicitely when capturing the texture by a photograph. The window is typically neither perfectly aligned with the regular structure nor does it contain an *integer* amount of oscillations of the regular structure. A solution for this problem is the use of a fractional discrete Fourier transform.

3.1. Fractional DFT

Definition 3.1 (Fractional Frequency) For a function $f(k) = \cos(2\pi\frac{b}{m}k)$ defined on $[0,m]$, $b \in [0,\frac{m}{2}]$, we call b the frequency with respect to m. b is called a *fractional frequency* if $b \notin \mathbb{Z}$. For a two-dimensional function

$$f(k_x,k_y) = \cos\left(2\pi i\left(\frac{b_x k_x}{m_x} + \frac{b_y k_y}{m_y}\right)\right)$$

the analogon holds for a frequency pair (b_x,b_y) with respect to m_x and m_y.

Synthesizing the regular structure by continued evaluation of periodic functions requires exact knowledge of the fractional frequencies corresponding to the regular structures. A method to analyse these frequencies is the fractional Fourier transform (FrDFT, also called linear FrDFT [Wei]) [BS90]. While the DFT of a sequence $a_j, j = 0,\ldots,m-1$ is a function $FT : \mathbb{Z} \to \mathbb{C}$ with

$$FT(z) = \frac{1}{m}\sum_{j=0}^{m-1} a_j e^{-2\pi i\frac{zj}{m}}, \tag{1}$$

the fractional Fourier transform of a sequence $a_j, j = 0,\ldots,m-1$ is a function $FT : \mathbb{R} \to \mathbb{C}$ with

$$FT(k) = \frac{1}{m}\sum_{j=0}^{m-1} a_j e^{-2\pi i\frac{kj}{m}} \tag{2}$$

or for a two-dimensional texture

$$FT(k_x,k_y) = \frac{1}{m_x m_y}\sum_{j_y=0}^{m_y-1}\sum_{j_x=0}^{m_x-1} a_{j_x j_y} e^{-2\pi i\left(\frac{k_x j_x}{m_x}+\frac{k_y j_y}{m_y}\right)}. \tag{3}$$

The following property provides the key idea to extract the exact fractional frequency of a signal: The absolute value $|FT(k)|$ of the FrDFT of a *complex* periodic sequence of the form $a_j = F_b e^{2\pi i\frac{bj}{m}}, j = 0,\ldots,m-1, F_b \in \mathbb{C}$ with frequency $b \in \mathbb{R}$ reaches its maximum for $k = b$ [BS90] and $FT(b) = F_b$ is the Fourier coefficient that determines intensity and phase (see Figure 3). To exploit this property (which analogously holds for frequency pairs) in the context of texture synthesis, two problems have to be solved. First, textures are real-valued. Second, the regular structure of most relevant textures is contained in more than one frequency and therefore not only a single frequency has to be determined.

© The Eurographics Association and Blackwell Publishing 2005.

A widely accepted approximation to the first problem is the computation of the *analytic signal* from the real-valued input by computing the Hilbert transform [Bra99].

Solving the second problem is more difficult. If the regular structure is contained in several frequency pairs (as usual), the FrDFT is the sum of the FrDFT of each of the pairs. As the small plot in Figure 3 shows, $|FT(k)|$ contains many local maxima even for a function determined by a single dominant fractional frequency. The FrDFT of a real sequence as in the large plot of Figure 3 therefore contains various local maxima of which some correspond to fractional frequencies contained in the regular structure and some do not. In the following we will therefore describe a method to identify the correct maxima in $FT(k_x,k_y)$ corresponding to the fractional frequencies that carry the regular structure.

The method we developed is based on the fact that the magnitude m_{l_i} of a local maximum at (k_{x_i},k_{y_i}) is much smaller than the magnitude m_g of the global maximum at (k_{x_g},k_{y_g}) ($m_{l_i} \approx m_g/|\pi^2(k_{x_g}-k_{x_i})(k_{y_g}-k_{y_i})|$ for $k_{x_i} \neq k_{x_g}$ and $k_{y_i} \neq k_{y_g}$). This is due to the fact that the FrDFT of a 2D function with frequencies b_x and b_y can be approximated as follows (for a proof see Appendix A):

$$|FT_{b_x,b_y}(k_x,k_y)|^2 \approx |F_{b_x,b_y}|^2 \cdot \mathrm{sinc}^2(k_x-b_x) \cdot \mathrm{sinc}^2(k_y-b_y) \quad (4)$$

where $sinc(x)$ is defined as

$$\mathrm{sinc}(x) = \begin{cases} \frac{\sin(\pi x)}{\pi x} & x \neq 0 \\ 1 & x = 0 \end{cases}. \quad (5)$$

Due to this strong difference between the magnitude of local and global maxima, the global maxima should be well preserved even when multiple fractional frequencies are contained in the sequence. In special, the most intense maximum should still remain the global maximum, which turned out to be a valid assumption during our tests.

The basic method for extraction of relevant frequencies is therefore the following: we determine the frequency with largest absolute value of the corresponding fractional Fourier coefficient, remove the contributions from this frequency from the analyzed image, do a FrDFT transformation on the remaining image, and keep continuing extracting frequencies until the absolute value of the largest remaining Fourier coefficient is below some threshold. Determining largest values in each step is done by first sampling the coefficients with a fixed step size and then applying a standard maximization algorithm, starting at the sample with largest value.

Unfortunately, extracting frequencies in this way is very time consuming (the run-time of the algorithm is roughly $O(m_x m_y \frac{m_x}{d_x}\frac{m_y}{d_y} n)$, where d_x and d_y denote the distance in x and y direction between values for which the FrDFT coefficient is computed and n denotes the number of extracted frequencies). We therefore suggest to improve the run-time complexity of the algorithm in two ways. First, the maximum search in the 2D-FrDFT can be reduced to the one-dimensional case, because for fixed k_x or fixed k_y Equation (4) is a 1D-sinc^2 function. To find a 2D-maximum, we therefore first search a 1d-maximum with fixed k_y and then use its position as fixed k_x.

Threshold	0.5	1	2	5
Frequency pairs found	175	76	35	14
Runtime	97 s	48 s	27 s	11 s

Table 1: *Runtime and result size of the FrDFT analysis for various thresholds and the texture from Figure 1 measured on a 2.4 GHz PC.*

Applying this first modification, the run-time complexity is reduced to $O(m_x m_y(\frac{m_x}{d_x}+\frac{m_y}{d_y})n)$ but still remains high. Therefore, our second optimization is to restrict the regions in which we search for the maxima. As visible in Figure 3, the position of a FrDFT global maximum can roughly be determined using the DFT. We are only looking for the most intense frequencies and thus only the highest maxima of FT. The DFT gives us the values of FT for $k_x,k_y \in \mathbb{Z}$ and therefore we only need to search the $[-1,1]^2$ regions around DFT coefficients that are larger than a threshold. By choosing a high threshold, a large amount of time can be saved (Table 1). Applying this second modification reduces the runtime to $O(m_x m_y(\frac{2}{d_x}+\frac{2}{d_y})c)$, where c denotes the number of DFT values above the threshold.

The extracted frequency pairs are near-optimal only since interference effects caused by the summation of multiple frequencies shift the location of maxima slightly. To compensate for this effect, we optimize the frequency pairs such that the least-squares color difference between the original image and the one reconstructed from the frequency pairs is minimized.

3.2. Fractional Fourier Texture Masks

The FrDFT analysis determines frequency pairs $b_1,\ldots,b_n$ with respective Fourier coefficients $F_1,\ldots,F_n$. These frequencies can be synthesized using the inverse DFT formula:

$$a_{j_x j_y} = \sum_{h=1}^{n} F_h e^{2\pi i\left(\frac{b_{h_x} j_x}{m_x}+\frac{b_{h_y} j_y}{m_y}\right)}. \quad (6)$$

If we choose $j_x, j_y \in \mathbb{Z}$, the size of the synthesized texture is unlimited and the frequencies of the regular structures represent the parameters of a procedural texture containing these structures. This process is known as Fourier synthesis [WW92] and we call the output *fractional Fourier texture masks* (FFTMs). Examples of FFTMs are shown in the third row of Figure 7.

We can use the frequency information to calculate the size

© The Eurographics Association and Blackwell Publishing 2005.

of a texture with *tilable* regular structures. A regular structure is tilable, if the frequencies describing the structure are not fractional w.r.t. the texture size. In simple cases, this is the most intense frequency pair. In the general case, the lowest frequency of the regular structures has to be determined. Let b_l be this frequency pair. Then

$$m'_x = m_x \cdot \frac{n_x}{b_{l_x}} \text{ and } m'_y = m_y \cdot \frac{n_y}{b_{l_y}} \tag{7}$$

produces tilable sizes for all $(n_x, n_y) \in \mathbb{N} \times \mathbb{N}$ and $n_x = n_y = 1$ is a single tile of the pattern as described by Liu et al. [LLH04].

The FrDFT analysis and synthesis are performed for each color channel separately. Our tests determined that transforming the RGB color data to YC_rC_b colorspace gives best results since it reduced deviation of the per-channel frequencies. In addition, for many textures the analysis turned out to become faster, because usually the Y-channel contains the most information.

4. Synthesis

In this section we describe how to incorporate FFTMs into existing pixel- or patch-based synthesis algorithms. The output texture will be denoted by T_{out}, by M_{out} we will refer to a FFTM with size equal to the desired output size, and by M_{in} to the part corresponding to input texture T_{in}. In this way for every pixel in T_{in} there is exactly one corresponding pixel in the FFTM M_{in}.

4.1. Pixel-Based Synthesis

In the pixel-based synthesis approach an output texture is generated from an input texture by selectively copying pixels. A pixel is chosen by comparing the neighborhood of the current pixel in the output to a candidate list of neighborhoods of pixels in the input texture. The comparison of neighborhoods is typically carried out by computing the L_2 norm of the difference between neighborhood vectors $\vec{N}(T_{in}, x_{in})$ and $\vec{N}(T_{out}, x_{out})$. A typical neighborhood is shown in Figure 4.

Incorporating FFTMs into pixel-based synthesis algorithms can be achieved in different ways, from which we implemented the two following ones:

Full neighborhood comparison. This approach is inspired by the texture mask approach as proposed by Zhang et al. [ZZV*03]. The neighborhood used to determine best matching pixels is extended from N_{upper} to $N_{full} = N_{upper} \cup N_{lower}$. Whereas the similarity of pixels in N_{upper} is measured by the standard L_2 norm

$$d_{TT} = \left\| \vec{N}(T_{in}, x_{in}) - \vec{N}(T_{out}, x_{out}) \right\|_2 . \tag{8}$$

For pixels in N_{lower} we evaluate the L_2 norm

$$d_{MM} = \left\| \vec{N}(M_{in}, x_{in}) - \vec{N}(M_{out}, x_{out}) \right\|_2 \tag{9}$$

of the respective FFTM values. The contributions from both parts are summed into a single value after scaling the FFTM-similarities by a mask weight c. Please note that incorporating d_{MM} can be interpreted as the previously missing way to improve the standard similarity measure to consider structural information as well.

Cluster ID match. The pixels of M_{in} are clustered based on similar neighborhoods N_{full} using k-means clustering. Whenever a pixel x_{out} is synthesized, the cluster ID of the corresponding pixel is determined. This cluster ID defines a candidate set of pixels in M_{in} and thus corresponding pixels T_{in} from which the pixel to be copied is selected using the measure d_{TT} defined above.

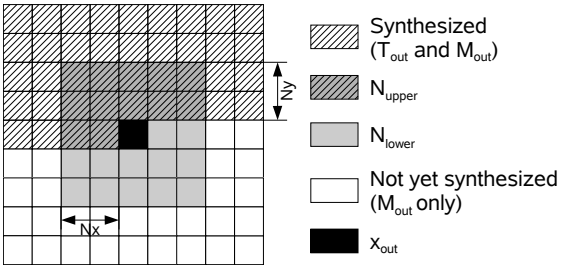

Figure 4: *The neighborhoods N_{upper} and N_{lower} with size $N_x \times N_y$. N_{upper} is defined for all textures and masks, but N_{lower} is undefined for T_{out}.*

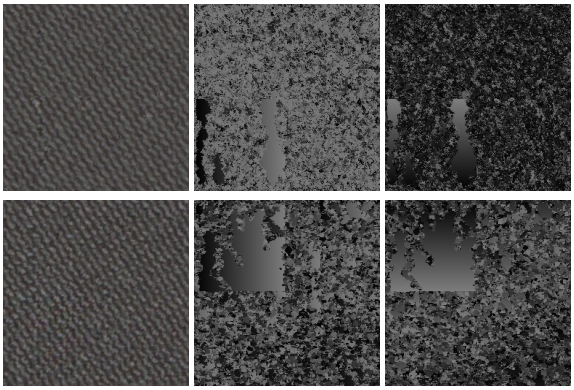

Figure 5: *Upper row: synthesis results with k-coherence and cluster ID match. The results with full search and full neighborhood comparison are similar. Lower row: ANN search and full neighborhood comparison. Both examples used a 3×3 neighborhood. The images on the right visualize x and y components of source coordinates respectively. The original texture is shown in Figure 7.*

To accelerate the actual neighborhood selection standard strategies like the k-coherence search of Tong et al. [TZL*02] and approximate nearest neighbor (ANN) search [LLX*01, ZG02] can be applied in both approaches.

© The Eurographics Association and Blackwell Publishing 2005.

We achieved the best results for the combination of full neighborhood comparison with ANN search. An examination of the output source coordinates as illustrated in Figure 5 shows the reason for the better quality of the ANN search results. The apparent noise in the output texture is caused by very little spatial coherence in the source coordinates. The noise is *not* caused by problems in the k-coherence or the ANN search algorithm, because it occurs even with an exhaustive search that always finds the best matching neighborhood. If the ANN search with the full neighborhood search is used, the source coordinates show that small spatially coherent regions are copied from the input texture. This is due to the implementation of the ANN algorithm we use [AM], which favors neighborhood vectors close to the query vector if the tree was built with the vectors in scanline order.

During our experiments, we found that for reproducing simple structures with the ANN approach, a small 2×2 neighborhood and a weight $c \approx 1$ produce good results. Reproduction of more complex structures like knitted wool requires a larger 6×6 neighborhood to obtain good results and possibly a higher value for c. Usually, if $c < 0.1$ the mask has no effect, and if $c > 5$ the mask weight is too large. In case large neighborhoods are required, we apply Principal Component Analysis to reduce the dimension of the neighborhood vectors to about 16, which significantly increases synthesis speed while just slightly decreasing synthesis quality.

4.2. Patch synthesis

Incorporation of FFTMs into patch-based synthesis algorithms basically follows the same idea as for pixel-based methods. Whereas the quality of pixel-based algorithms was improved by completing the full neighborhood with values from FFTMs, we applied the same principle to parts of patches that do not overlap with already synthesized pixels.

Additionally, to support iterative improvement of synthesized textures using the Graphcut algorithm [KSE*03] we extended the pixel representation from standard 3D RGB vectors to 6D vectors that additionally contain the respective mask values, scaled by the FFTM weight c. This way, we can simply reuse the standard L_2 norm to compute $d_{MM} + c \cdot d_{TT}$, which helps to find well fitting patches that additionally preserve the regular structure.

4.3. Extensions

Our algorithm can easily be extended to create tilable textures. With Equation (7) we calculate an appropriate output size for the regular structures. If we wrap around the neighborhoods at the border and add a second pass for the pixels where the opposite neighborhood was undefined during the first pass, the irregular structures produced by the pixel synthesis fit without visible edges.

5. Results

We tested our algorithms with various texture samples. The pixel-based algorithm is very easy to use since the only manual parameters that need to be selected are the intensity filter threshold, the neighborhood size, and the mask weight. For synthesizing textures with the help of FFTMs we used a 3×3 neighborhood which is (much) smaller than the largest structures in the textures. When omitting FFTMs much larger neighborhoods were used (for the images in Figure 7, from left ro right 6×6, 5×5, 8×8, 10×10, and 10×10) which increases synthesis times drastically.

As the results in Figure 7 shows, FFTMs clearly help to preserve regular structures. While standard ANN-accelerated per-pixel synthesis of the textile textures fails to preserve the appearance of the textures – especially reproduction of the rightmost texture fails completely – simply employing the FFTMs as textures performs much better already, since they preserve the textures' most significant property: their regularity. Addition of irregular detail further improves the quality of the texture whereas the degree of improvement depends on the visual dominance of the regular structure. Especially textures with visually rich appearance like the knitted wool or the rightmost texture profit extremely. To our knowledge no other automatic texture synthesis algorithm is capable of producing results that resemble the input textures that closely. Additional, very nice results that do not represent textiles are shown in Figure 8.

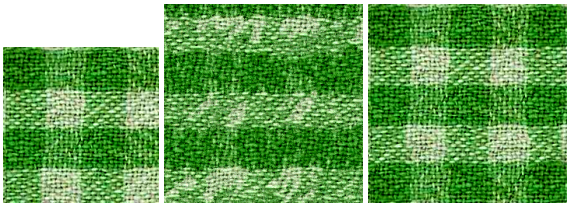

Figure 6: *Example for improved quality of patch-based synthesis. Although the mask has good quality, the ANN synthesis (middle) fails to reproduce the irregular structures. Applying extended patch-based synthesis incorporating FFTMs (right) the results become very pleasing.*

Synthesizing textures using our patch-based approach leads to very good results which are superior to the results of the ANN-accelerated pixel-based method in some cases. Figure 6 shows an example in which the pixel-based algorithm fails to add the irregular details but which is handled by the patch-based algorithm very well. Further results of this method are shown in Figure 9. Synthesis was done using the sub-patch matching strategy [KSE*03], which was previously not known to give good results for near-regular texture, which confirms the usefulness of FFTMs. The used patch size can be determined automatically by computing parts of the texture that are tileable (see Section 3.2), which

© The Eurographics Association and Blackwell Publishing 2005.

are usually much smaller than the full texture, leading to runtime advances compared to the entire patch matching strategy [KSE*03]. The determined patch size needs not be very accurate to achieve pleasing results (e.g. the bottom right texture in Figure 9 contains features of size 75×60 pixels and was synthesized using a patch size of 40×40). Please note that synthesis of the textures in Figure 6 and in the top left of Figure 9 was previously not possible using fully automatic algorithms (see [LHW*04]).

	128^2 pixels		239^2 pixels	
Neighborhoodsize	No PCA	PCA	No PCA	PCA
2×2	0.16	0.07	0.42	0.10
3×3	0.42	0.12	1.44	0.14
4×4	0.90	0.21	3.22	0.21
5×5	1.65	0.37	—	0.30

Table 2: *Runtime per output pixel for the ANN synthesis in milliseconds for the first (128^2 pixels) and fourth (239^2 pixels) sample in figure 7. PCA was applied to the neighborhood vectors from T_{in} and M_{in} to reduce their dimension to 16.*

Table 2 shows some runtime examples for the ANN synthesis, measured on a 2.4 GHz Pentium 4. The runtime increases with the neighborhood size, but can be reduced significantly using PCA. Other measurements show that the ANN search time depends on the sample appearance, because the time varies for samples of the same size. In general, we found an approximate runtime of $O(\frac{|N|}{\varepsilon} \log |T_{in}|)$ per output pixel.

Synthesis times for the patch-based method are much longer, typically many minutes for the examples shown in this paper. The exact time naturally depends on the size of the synthesized texture, the number of improvement steps necessary or allowed, and the size of copied patches. Nevertheless, synthesis times are not noticeably longer than without incorporation of FFTMs.

Comparing the pixel- and patch-based synthesis approaches is difficult. For most textures, ANN-accelerated synthesis already yields very good results at a fast speed. Nevertheless, we found the patch-based algorithm to be more robust: it provides good results even in cases where the pixel-based approach fails.

The quality of the FFTM has great influence on the result. If the mask contains the regular structures without errors, the synthesized textures are of high quality. Otherwise, the synthesis algorithms produce errors as shown in Figure 10.

The errors are caused by slight deviations from the true frequency in the FrDFT analysis process and occur especially, if the texture has sharp structures like the brick wall. The Fourier transform of these structures consists of several frequencies at distinct, fixed ratios to each other. If the FrDFT analysis fails to determine these frequencies exactly due to numerical problems or due to local minima the Levenberg-Marquardt optimization method gets stuck in, undesired interference patterns as visible in the mask in Figure 10 occur. The errors are usually small enough to allow a reproduction of the sample, thus the FrDFT intensity filter can always be used to avoid the DFT leakage effect. However, for textures with difficult structures, the larger the FrDFT synthesized area becomes, the more disturbed the result gets (see Figure 10).

6. Conclusions

In this paper a new method for synthesis of near-regular textures was proposed. The key observation behind this method is that an DFT intensity filter can be applied to separate dominant regular structures from irregular texture detail. Since the DFT intensity filter works for already tileable textures only, we applied FrDFT instead of DFT. We introduced an algorithm for automatic extraction and synthesis of the regular patterns, resulting in fractional Fourier texture masks. Based on these masks we proposed sample-based texture synthesis algorithms that add the missing irregular texture details and showed the high quality of results by several examples.

Application of the FrDFT intensity filter should prove beneficial in combination with other methods as well. Liu et al.'s [LTL05] approach for computation of shape and size of tileable texture elements, which is based on autocorrelations, should be improved by removing irregular texture parts. Additionally, the extracted frequencies represent an explicit description of unique texture properties and might thus contribute to accurate texture recognition. Finally, the intensity filter might be employed as the basis of a specialized compression technique.

Although our method turned out to be very robust for textures with regular structures that carry a large amount of energy, handling of regular textures like pinstriped suits is difficult. Improving our technique for textures where the regular patterns are very small and appear at very distant locations – leading to a much smaller amount of energy – remains for future research.

In addition, handling the problem of inexact extraction of frequencies as described in Section 5 will also be investigated further.

Finally, in order to make our approach applicable to near-regular textures with substantial geometric variation, we plan to combine our method with regularization approaches like the one of Liu et al. [LLH04].

Acknowledgements

This work was partially funded by the European Union under project RealReflect (IST-2001-34744).

© The Eurographics Association and Blackwell Publishing 2005.

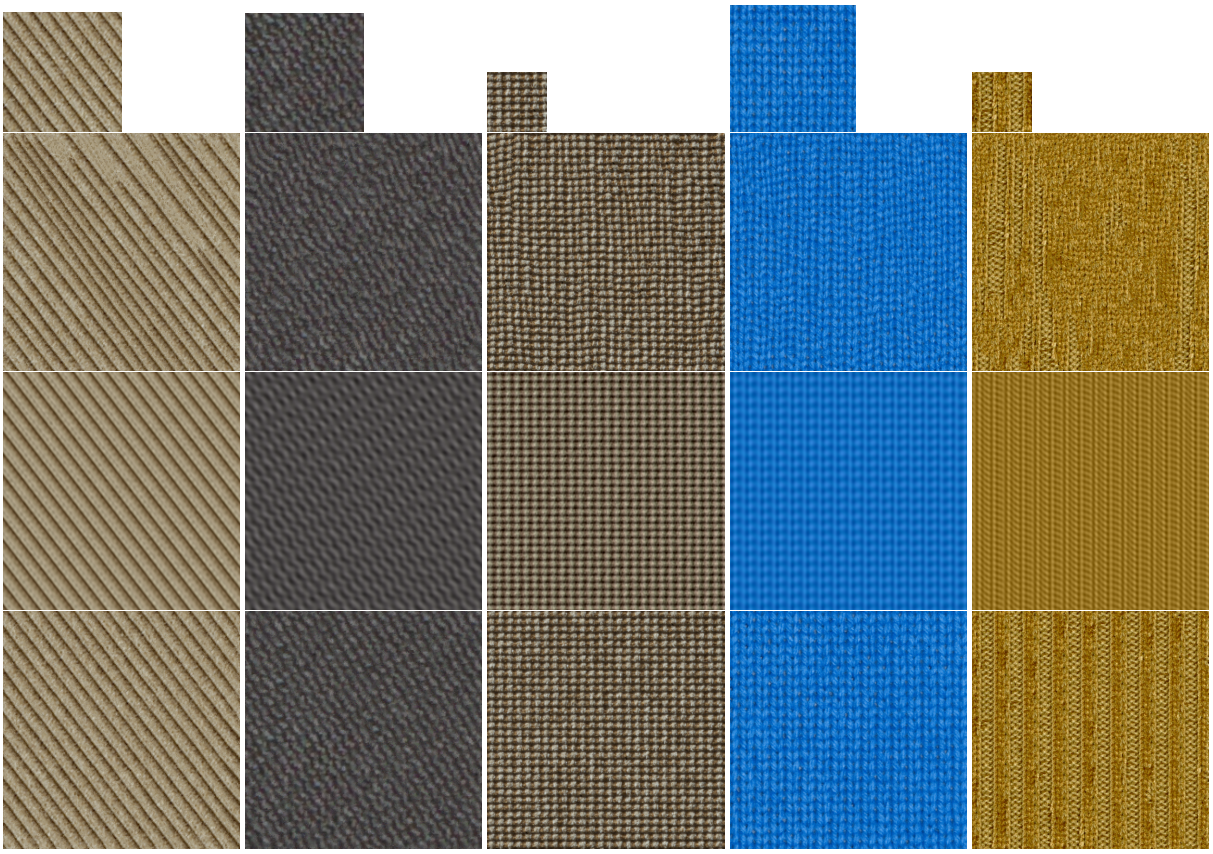

Figure 7: *Examples for successful synthesis. Each column shows from top to bottom the input sample, standard ANN synthesis without a mask, the mask created with the FrDFT synthesis, and the ANN synthesis result with mask. In many cases the FFTM provides a very accurate approximation of the synthesized texture already.*

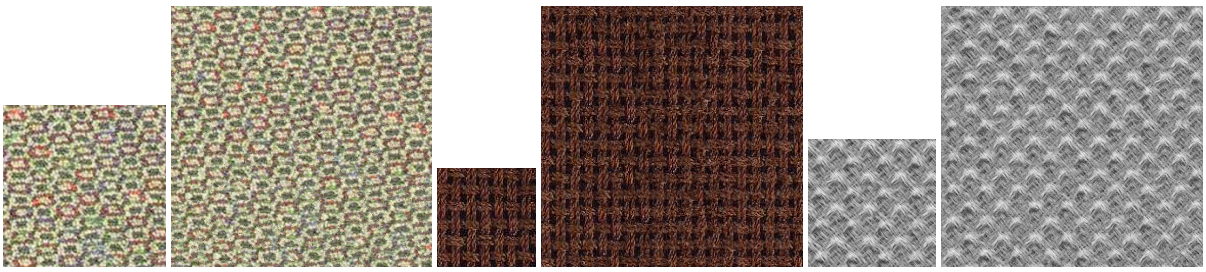

Figure 8: *Examples for successful synthesis of non-textile samples with the pixel-based approach.*

© The Eurographics Association and Blackwell Publishing 2005.

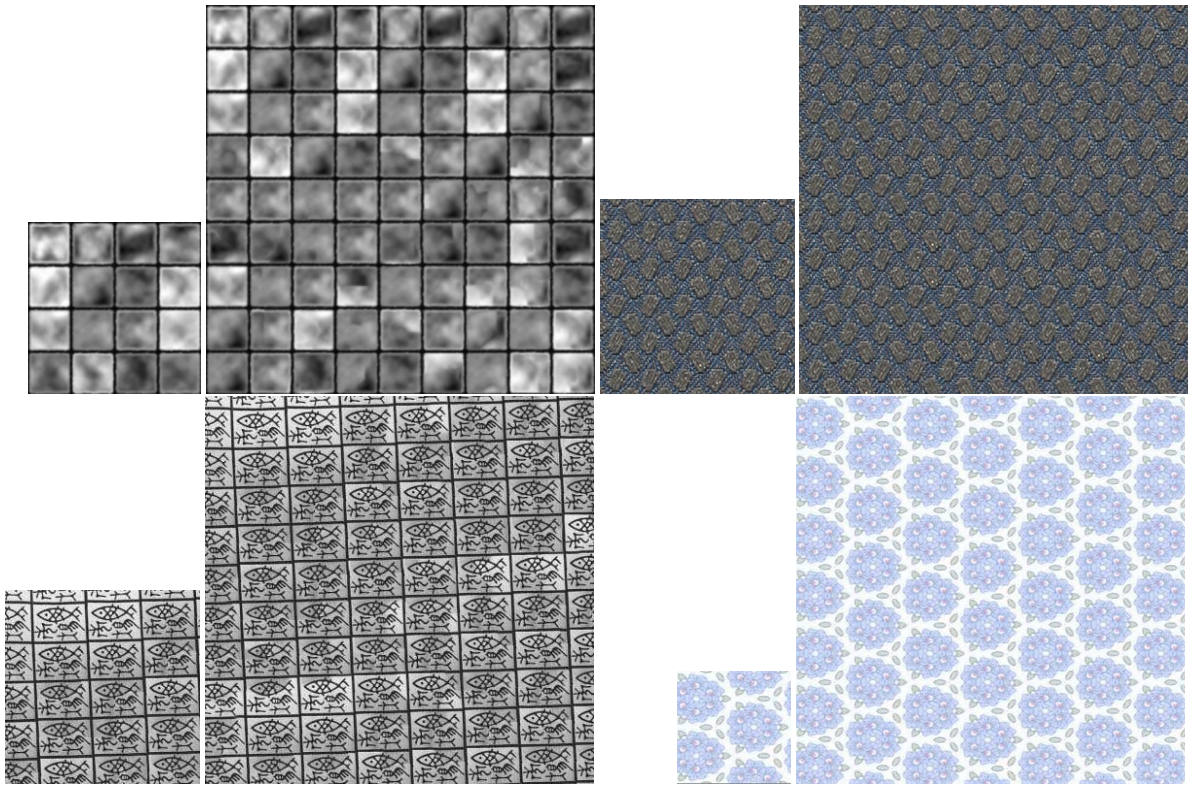

Figure 9: *Examples for successful synthesis with the patch-based approach.*

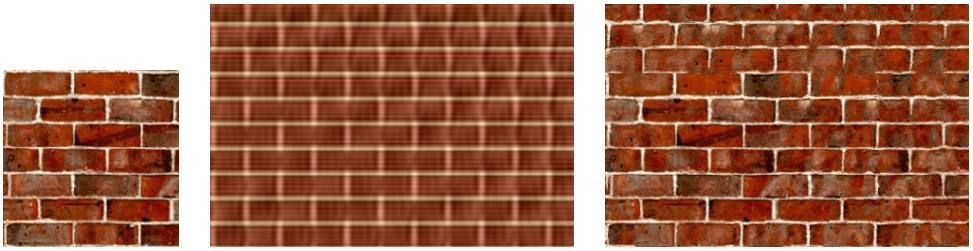

Figure 10: *Problematic synthesis. The FFTM (middle) contains errors if the frequencies are extracted inaccurately, which leads to undesired interference. In such a case, texture synthesis (right) fails to reproduce the original appearance (left).*

Appendix A: Proof of Approximation 4

Theorem A.1 Let $d := 2\pi(k-b)$. Then for $k, b \in [0, m[$

$$|FT_b(k)|^2 \approx \operatorname{sinc}^2(k-b). \tag{10}$$

Proof A.1

$$sinc^2(k-b) = \frac{2}{d^2}\left(1-\cos(d)\right).$$

$$|FT_b(k)|^2 = \frac{1}{m^2}\left|\frac{1-e^{id}}{1-e^{\frac{id}{m}}}\right|^2 = \frac{1-\cos(d)}{m^2(1-\cos(\frac{d}{m}))} \text{ [BS90]}$$

The Taylor series of $1-\cos\left(\frac{d}{m}\right)$ gives

$$1-\cos\left(\frac{d}{m}\right) \approx \frac{d^2}{2m^2}$$
$$\Rightarrow \frac{1-\cos(d)}{m^2(1-\cos(\frac{d}{m}))} \approx \frac{2}{d^2}\left(1-\cos(d)\right)$$

If frequency b is described by Fourier coefficient F_b, then $|FT_b(k)|^2 \approx (|F_b| \cdot \operatorname{sinc}(k-b))^2$. For a two-dimensional frequency (b_x, b_y) with coefficient F_{b_x,b_y}

$$|FT_{b_x,b_y}(k_x,k_y)|^2 \approx |F_{b_x,b_y}|^2 \cdot \operatorname{sinc}^2(k_x-b_x) \cdot \operatorname{sinc}^2(k_y-b_y) \tag{11}$$

for $b_x, k_x \in [0, m_x[$ and $b_y, k_y \in [0, m_y[$.

© The Eurographics Association and Blackwell Publishing 2005.

References

[AM] ARYA S., MOUNT D.: ANN: Library for Approximate Nearest Neighbor Searching. www.cs.umd.edu/∼mount/ANN/.

[Ash01] ASHIKHMIN M.: Synthesizing natural textures. In *Proc. of Symposium on Interactive 3D Graphics* (2001), pp. 217–226.

[BJEYLW01] BAR-JOSEPH Z., EL-YANIV R., LISCHINSKI D., WERMAN M.: Texture Mixing and Texture Movie Synthesis Using Statistical Learning. *IEEE Trans. Vis. and Comp. Graphics 7*, 2 (2001), 120–135.

[Bon97] BONET J. S. D.: Multiresolution sampling procedure for analysis and synthesis of texture images. In *Computer Graphics* (1997), pp. 361–368.

[Bra99] BRACEWELL R.: *The Fourier Transform and Its Applications*, 3rd ed. McGraw-Hill, NY, 1999.

[BS90] BAILEY D. H., SWARZTRAUBER P. N.: *The Fractional Fourier Transform and Applications*. Tech. Rep. RNR-90-004, NASA Advanced Supercomputing, Mail Stop T27-A, Moffett Field,CA 94035, April 1990.

[CSHD03] COHEN M. F., SHADE J., HILLER S., DEUSSEN O.: Wang tiles for image and texture generation. *ACM Trans. Graphics 22*, 3 (2003), 287–294.

[DMLC02] DISCHLER J.-M., MARITAUD K., LÉVY B., CHAZANFARPOUR D.: Texture Particles. In *Proc. of Eurographics 2002* (2002).

[EF01] EFROS A. A., FREEMAN W. T.: Image quilting for texture synthesis and transfer. *Proc. of SIGGRAPH 2001* (August 2001), 341–346.

[EL99] EFROS A. A., LEUNG T. K.: Texture synthesis by non-parametric sampling. In *Proc. of IEEE Int. Conf. on Computer Vision – Volume 2* (1999), p. 1033.

[HB95] HEEGER D. J., BERGEN J. R.: Pyramid-based texture analysis/synthesis. In *Proc. of SIGGRAPH 1995* (1995), pp. 229–238.

[KSE*03] KWATRA V., SCHÖDL A., ESSA I., TURK G., BOBICK A.: Graphcut Textures: Image and Video Synthesis Using Graph Cuts. *ACM Trans. Graphics 22*, 3 (July 2003), 277–286.

[LCT04] LIU Y., COLLINS R. T., TSIN Y.: A Computational Model for Periodic Perception Based on Frieze and Wallpaper Goups. *IEEE Trans. Pattern Analysis and Machine Intelligence 26*, 3 (2004), 354–371.

[Lev02] LEVINA E.: *Statistical Issues in Texture Analysis*. PhD thesis, University of California, Berkeley, 2002.

[LHW*04] LIN W.-C., HAYS J. H., WU C., KWATRA V., LIU Y.: *A Comparison Study of Four Texture Synthesis Algorithms on Regular and Near-regular Textures*. Tech. Rep. CMU-RI-TR-04-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 2004.

[LLH04] LIU Y., LIN W.-C., HAYS J. H.: Near regular texture analysis and manipulation. In *Proc. of SIGGRAPH '04* (August 2004).

[LLX*01] LIANG L., LIU C., XU Y.-Q., GUO B., SHUM H.-Y.: Real-Time Texture Synthesis by Patch-Based Sampling. *ACM Trans. Graphics 20*, 3 (2001), 127–150.

[LTL05] LIU Y., TSIN Y., LIN W.-C.: The Promise and Perils of Near-Regular Texture. *International Journal of Computer Vision 62*, 1-2 (2005), 145–159.

[LYS01] LIU X., YU Y., SHUM H.-Y.: Synthesizing bidirectional texture functions for real-world surfaces. In *Proc. of SIGGRAPH 2001* (2001), pp. 97–106.

[NA03] NEALEN A., ALEXA M.: Hybrid texture synthesis. In *Proc. of 14th Eurographics Workshop on Rendering* (2003), pp. 97–105.

[Per85] PERLIN K.: An image synthesizer. In *Proc. of SIGGRAPH 1985* (1985), vol. 19(3), pp. 287–296.

[PS00] PORTILLA J., SIMONCELLI E. P.: A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients. *International Journal of Computer Vision 40*, 1 (2000), 49–70.

[Tur91] TURK G.: Generating textures on arbitrary surfaces using reaction-diffusion. In *Proc. of SIGGRAPH 1991* (1991), pp. 289–298.

[TZL*02] TONG X., ZHANG J., LIU L., WANG X., GUO B., SHUM H.-Y.: Synthesis of bidirectional texture functions on arbitrary surfaces. In *Proc. of the 29th annual conference on Computer graphics and interactive techniques* (2002), pp. 665–672.

[Wei] WEISSTEIN E. W. ET AL.: Fractional fourier transform. from mathworld–a wolfram web resource. mathworld.wolfram.com/FractionalFourierTransform.html.

[WL00] WEI L.-Y., LEVOY M.: Fast texture synthesis using tree-structured vector quantization. In *Proc. of SIGGRAPH 2000* (2000), ACM Press/Addison-Wesley Publishing Co., pp. 479–488.

[WW92] WATT A., WATT M.: *Advanced Animation and Rendering Techniques*. ACM Press New York / Addison-Wesley, 1992.

[ZG02] ZELINKA S., GARLAND M.: Towards real-time texture synthesis with the jump map. In *Proc. of 13th Eurographics Workshop on Rendering* (2002).

[ZWM98] ZHU S. C., WU Y., MUMFORD D.: Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision 27*, 2 (1998), 107–126.

[ZZV*03] ZHANG J., ZHOU K., VELHO L., GUO B., SHUM H.-Y.: Synthesis of progressively-variant textures on arbitrary surfaces. *ACM Trans. Graphics 22*, 3 (2003), 295–302.

© The Eurographics Association and Blackwell Publishing 2005.