

NONPARAMETRIC WAVELET BASED TEXTURE SYNTHESIS

Claire Gallagher and Anil Kokaram

Department of Electrical and Electronic Engineering,
Trinity College, University of Dublin, Ireland.

ABSTRACT

This paper presents a new algorithm for synthesising image texture. Texture synthesis is an important process in image post-production. Previous approaches can be classified as either parametric or nonparametric. Of these nonparametric approaches have achieved the most impressive results. Unfortunately, these methods generally suffer from high computational cost and difficulty in handling scale in the synthesis process. This paper introduces a new idea of using wavelet decomposition as a basis for nonparametric texture synthesis. The results show an order of magnitude improvement in computational speed and a better approximation of the dominant scale in the synthesised texture. **Keywords:** *Texture Synthesis, Complex Wavelet Transform, Image Processing, Nonparametric Image Modeling.*

1. INTRODUCTION

The problem of texture synthesis has received much interest in recent years [1, 2, 3, 4]. The idea behind a successful texture synthesis algorithm is to use a small texture sample image to create a much larger texture image by synthesising *new* texture which will be perceived to be visually similar to the original texture sample. This kind of operation is often required in the post-production of digital images when a large area is to be covered with texture that *looks like* some small example. Picture editing often requires filling in missing information or removing certain objects [5] and texture synthesis processes are often used to fill in such holes with reasonable material [3].

The essential idea behind pixel-based synthesis is to estimate the probability density function (p.d.f.) of the image intensity $I(\underline{x})$, denoted by $P(I(\underline{x}))$ at a pixel site $\underline{x} = (i, j)$. The process of texture synthesis is then a matter of drawing a random sample from that distribution. What makes this difficult is estimating $P(I(\underline{x}))$. Two different approaches have emerged. Parametric techniques attempt to model $P(I(\underline{x}))$ with some definable process. Previous techniques include analysing texture using histograms of filter responses at multiple scales and orientations of a steerable pyramid [6]. Portilla and Simoncelli [7] improve on this idea by matching pairwise statistics across different scales and orientations. Campisi [4] uses a method which extracts the information contained in circular harmonic functions to represent the relevant features of interest in the texture. A simpler yet effective method for “filling” in

holes or missing pixels in images is the use of an Autoregressive model to describe the texture [3]. All of these methods work well on simple textures but fail for more structured textures [7]. Nonparametric approaches rather, attempt to *measure* the p.d.f. from the image itself. In both cases the visual quality of the generated textures will be influenced primarily by the accuracy of the model, while the efficiency of the sampling procedure will be directly related to the computational expense [2].

Because of the wide variability in image behavior, nonparametric methods have achieved by far the more visibly pleasing results [1, 2, 8]. The popularity of nonparametric approaches is mainly due to an algorithm introduced by Efros and Leung in 1999 [1]. Their approach extended an idea introduced by Popat [9] where the p.d.f. of each pixel was empirically measured using neighbourhood similarity. Both methods exploited a loose Markovian concept, i.e. the intensity value for a pixel given the intensities of its spatial neighbours is independent of the rest of the image. However, the Efros approach differs in that they used exhaustive nearest neighbour searching. Their algorithm generates impressive results and works well on a large range of textures. However, the algorithm is computationally expensive given that an entire search of the sample image is necessary for each pixel to be synthesised. In addition and as a consequence of the Markovian assumption, the success of the algorithm is very much dependent on the correct choice of neighbourhood size. This user defined parameter controls the randomness of the texture to be generated and should be as large as the largest feature present in the texture.

Ashikhmin [8] addresses the computational burden of the Efros algorithm by introducing coherent searching into the synthesis procedure. This speeds up the synthesis process by eliminating the need to search every possible neighbourhood in the sample image. Wei and Levoy [2] adapt the algorithm to include multi resolution synthesis. Using Gaussian pyramids they decompose the texture image into different resolutions and seek to transform a random noise sample to resemble the sample texture at each resolution using an Efros style neighbourhood searching approach. This method works well on stochastic (random) textures but is not suitable for deterministic (structured) textures.

In order to address the problems of scale and computational cost associated with nonparametric methods, we have introduced the novel idea of using the complex wavelet transform as a basis for nonparametric texture synthesis. The introduction of the wavelet decomposition into the synthesis procedure has two advantages. Firstly, it facilitates the measurement of texture statistics at particular scales. Unlike previous methods, who use scale information as a control [2], we directly synthesise texture at these

gallaghc@mee.tcd.ie www.mee.tcd.ie/~sigmedia This work is supported by The Irish Research Council for Science, Engineering and Technology (IRCSET) Research Scholarship Foundation.

different scales. This allows us to exploit the dominant frequencies present in the texture image. The second advantage of our method is the reduction in computational cost. By synthesising texture at coarser scales, the original information is represented by fewer pixels.

This paper describes the wavelet based texture synthesis algorithm. Beginning with an outline of the single resolution nonparametric method developed by Efros [1], we describe how this was used as a basis for our wavelet based synthesis method. Results are presented and the computational efficiency of the algorithm is highlighted by comparing it to the original Efros algorithm. Synthesised images are shown and a comparison is done with similar nonparametric algorithms [2, 1].

2. SINGLE RESOLUTION TEXTURE SYNTHESIS

Let the image to be synthesised I_s be located on the grid \mathbf{X}_s of size $M \times N$ and I_e be the sample input texture on the grid \mathbf{X}_e of size $m \times n$. The algorithm assumes that I_e is large enough to capture the statistics of the underlying infinite texture. Let $\mathbf{p} \in \mathbf{X}_e$ be a pixel to be synthesised and $W(\mathbf{p})$ be the spatial neighbourhood of pixels surrounding \mathbf{p} with width w . To synthesise a value for \mathbf{p} an approximation to the conditional probability distribution $P(\mathbf{p}|W(\mathbf{p}))$ is constructed and sampled. The approximation is built by directly identifying all patches in I_e that are *perceptually similar* in some way to $W(\mathbf{p})$. The pixels at the centre of these similar patches then represent an empirical measurement of the p.d.f. required.

Let $d(W(\mathbf{p}_1), W(\mathbf{p}_2))$ denote the perceptual distance between two neighbourhoods or patches centred at locations \mathbf{p}_1 and \mathbf{p}_2 , where d is defined to be the sum of squared intensity differences. The most similar patch denoted W_{best} in I_e is found using the equation $W_{best} = \arg \min_{\mathbf{x} \in \mathbf{X}_e} d(W(\mathbf{p}), W(\mathbf{x}))$. All example patches W satisfying $d(W(\mathbf{p}), W) \leq (1 + \epsilon)d(W(\mathbf{p}), W_{best})$ are included in the set $\Omega(\mathbf{p})$. In this application $\epsilon = 0.1$. The centre pixel values of patches in $\Omega(\mathbf{p})$ provide a heuristic approximation to the p.d.f. of \mathbf{p} . This is sampled numerically to obtain a value for \mathbf{p} . To preserve the local structure of the texture, the error for pixels near the centre of the neighbourhood, i.e. close to \mathbf{p} , is larger than that for pixels close to the boundaries of the neighbourhood. This is achieved by weighting the distance measure $d(\cdot, \cdot)$ with a two-dimensional Gaussian Kernel of variance $w/6.4$.

This nonparametric single resolution algorithm generates impressive results on a wide variety of textures. However, searching the entire sample image for each pixel to be synthesised is computationally expensive and slows the algorithm considerably. In addition, the user defined neighbourhood width w is critical to successful texture synthesis. If the texture is presumed to be mainly stochastic at low spatial frequencies, the size of the window should be on the scale of the biggest regular feature. To avoid this dependence on neighbourhood size and decrease the computational cost involved in synthesising the texture, the complex wavelet transform has been incorporated into the synthesis process.

3. SYNTHESISING TEXTURE USING THE COMPLEX WAVELET TRANSFORM

The Dual Tree Complex Wavelet Transform (DT-CWT) developed by Kingsbury [10] has received much interest in image process-

ing applications. The exact workings for the DT-CWT is beyond the scope of this paper but the interested reader is directed toward [10] for a description of the transform. For this application all that is of interest is that the 2D DT-CWT produces six sub band images and one low pass image at each level of the transform. These sub band images are orientated at angles of $\pm 15^\circ, \pm 45^\circ, \pm 75^\circ$ and the resolution of the images decreases with level, level 1 is the original image. The DT-CWT gives a 4:1 redundancy for 2D images, this redundancy allows both shift invariance and good directional sensitivity.

3.1. Neighbourhood Searching

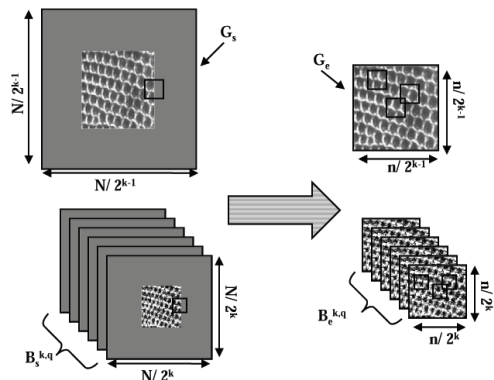


Fig. 1. $[B_s^l, L_s]$ (left) and $[B_e^l, L_e]$ (right) represent the DT-CWT decomposition for I_s and I_e . Each wavelet is synthesised by constructing its 6-D neighbourhood with centre pixels $\underline{\mathbf{x}}$ in each of the six sub band images. This neighbourhood is compared to all possible neighbourhoods in B_e^k . The most similar neighbourhoods form the conditional p.d.f. used to synthesise the new wavelet coefficient in B_e^k .

Given the example texture image I_e of size $n \times m$ and the output size $N \times M$ of the image to be synthesised I_s , the set of band pass images B_e^k where $k = 1 \dots l$ (level index) and the low pass image L_e is created for the l th level DT-CWT of I_e . At each of the l levels, B_e^k contains six band pass images orientated at $\pm 15^\circ, \pm 45^\circ, \pm 75^\circ$. Due to sub sampling with level, the resolution at level k is half that at level $k - 1$.

Texture synthesis is then the problem of generating l levels of band pass images B_s^k that are then combined to create I_s . To do this, each of B_s^k is seeded with part of the corresponding sub band images in B_e^k . The size, source and location of this sample should be consistent with that placed at the level below. That is, at level k the seed should be half that placed at level $k - 1$.

At level $k = l$ which is the coarsest level in terms of detail, an adaptation of the neighbourhood searching algorithm described in section 2 is used to synthesise the unknown wavelet coefficients at the coarsest level l . Because of correlation among the sub band images, each of the six sub band images is considered to be one dimension of an 6-D vector, see Figure 1. Let $\underline{\mathbf{p}}$ denote the six

unknown wavelet coefficients located at $\underline{x} = (i, j)$ in each of the six sub band images. The 6-D neighbourhood $W(\underline{\mathbf{p}})$ consisting of individual neighbourhoods around \underline{x} in each of the six sub band images is constructed. $W(\underline{\mathbf{p}})$ is compared to each of the possible neighbourhoods in B_e^k . The similarity between two neighbourhoods is given by the sum of the differences in magnitude between neighbourhoods in each of the individual sub band images. The set of neighbourhoods which are most similar to $W(\underline{\mathbf{p}})$ are found. The values at the centre of these neighbourhoods form an approximation to the p.d.f. $\Omega(\underline{\mathbf{p}})$ of $\underline{\mathbf{p}}$. $\Omega(\underline{\mathbf{p}})$ is sampled and the new value assigned to $\underline{\mathbf{p}}$. To synthesise the rest of the wavelet coefficients in B_s^k for $k < l$, one of three updating techniques can be applied as follows.

Copy Wavelet coefficients are fetched using the location of synthesised wavelet coefficients at the previous coarse resolution as a guide. For example, if the wavelet coefficient at location $\mathbf{x}_s = (i, j)$ in $B_s^{k,q}$ was chosen from location $\mathbf{x}_e = (i', j')$ in $B_e^{k,q}$, then the simplest choice for coefficients $(2i, 2j)$, $(2i - 1, 2j)$, $(2i, 2j)$ and $(2i - 1, 2j - 1)$ in $B_s^{k-1,q}$ is to copy the wavelet coefficients from $(2i', 2j')$, $(2i' - 1, 2j')$, $(2i', 2j')$ and $(2i' - 1, 2j' - 1)$ in $B_e^{k-1,q}$. This process is repeated for G_s and for all k levels of bandpass images where $k < L$.

Refined Searching Rather than directly copying, it is possible to refine the estimate with level. Thus the locations of good matches at a level l are used as seed locations for seeking refined neighbourhood matches in level $l - 1$. For practical purposes the neighbourhood size is increased as the resolution increases. By searching at higher resolutions, finer details in the texture are accounted for and so the result is a sharper synthesised texture. This sharpness is however at the expense of computational load.

Single Resolution Synthesis using Multi resolution Searching This method uses the DT-CWT as a guide rather than a basis for synthesising pixel coordinates. Neighbourhood similarity is measured as before at the coarse level of the DT-CWT. However, rather than synthesising wavelet coefficients, the location list $\Omega(\underline{\mathbf{x}})$ obtained at the coarse level is used as a guide for synthesising pixel values in the single resolution image. This implies an Efros style approach with a much reduced neighbourhood search.

Once all of the coefficients have been generated, the sub band and low pass images are inverse transformed to give an image which should resemble that of the sample texture. The above algorithm is based on generating gray scale texture from an initial seed. To synthesise colour textures, the image is first transformed from rgb colour space to yuv colour space. Synthesis is performed on the y (luminance) component and the corresponding wavelet values from the sample texture u and v are assigned to the u and v images of the synthesised texture.

4. COMPUTATIONAL LOAD

The table below compares the computational load of the new CWT algorithm with the single resolution algorithm discussed in section 2.

Algorithm	One Pixel p	Whole Image I_s
Efros Method $nhood = w$	$4mnw^2$	$4MNmnw^2$
CWT Method $nhood = w_1$	$4\frac{mn}{2^l}w_1^2 \times 6$	$4\frac{mn}{2^l}w_1^2 \times 6$

The load for performing in the CWT (*approx* 80mn) is negligible in comparison to the overall load and so not included in the overall sum. The table shows that the CWT method is faster than the Efros method by a factor of *approx*. $2^{2l}w^2/w_1^2$. The size of the neighbourhood used depends on the texture. This size should be as large as the largest feature present. The new algorithm synthesises at a series of resolutions using a subset of possible locations, hence the neighbourhood width w_1 is much smaller than that necessary for the Efros method. For example, in Figure 4, the sample images I_e were of size 128×128 pixels and the synthesised images I_s were of 256×256 pixels. The number of levels used was $l = 3$ and the neighbourhood width was set at $w_1 = 5$. Using a basic C++ implementation of the algorithm, the total synthesis process took approximately 1.7 seconds for the grayscale images. In comparison, the Efros method required a block size of 17×17 took approximately 4442 seconds to synthesise the equivalent images.

5. RESULTS

Synthesised images generated using the wavelet algorithm are shown in Figures 2, 3 and 4. The black square in the middle indicates the sample texture or “seed”. The resolution of the “seed” measures 128×128 pixels for the single texture images and 256×256 pixels for the two texture image. All output textures (512×512 pixels) were synthesised at level 3 of the DT-CWT using a neighbourhood width of $w_1 = 7$. Lower level wavelet coefficients were synthesised using the basic copy method described earlier.

Figure 3 shows the effects of using the three different methods for updating wavelet coefficients during the synthesis process. All three methods give similar results with the copy method being favoured because of its low computational cost. However, for very deterministic textures (e.g. text), the refined method or the single resolution using CWT will generate sharper textures as the neighbourhood search at higher resolutions includes more fine details present in the texture.

Comparing the CWT synthesised results against other similar methods [1, 2], it is clear that all three methods generate good results. However, the CWT method performs the synthesis at the fraction of the computational cost. Since the Efros approach synthesises at the highest resolution it generates the sharpest texture, however the algorithm is sensitive to correct use of neighbourhood size and is computationally expensive. Incorrect specification of neighbourhood size can result in the algorithm growing garbage as seen in the righthand corner of the top Efros image. The Wei and Levoy method performs well but fails for more deterministic textures (Figure 4). The refined searching method described earlier for updating wavelet coefficients was used when comparing with the Efros and Wei and Levoy method.

6. FINAL COMMENTS

In this paper a new texture synthesis algorithm was introduced. Given an initial sample texture, the algorithm generates new texture using a nonparametric technique that incorporates the Dual Tree Complex Wavelet Transform (DT-CWT). Results show that the algorithm works well on a wide variety of textures and is computationally efficient. Three variations of the algorithm are presented as well as some of the results attained. The algorithm has

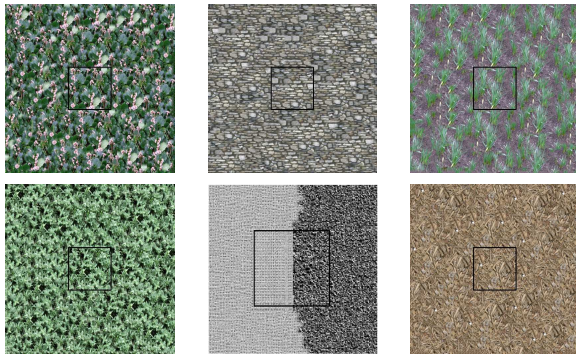


Fig. 2. Results from the CWT texture synthesis algorithm using the copy method. Example textures shown inside the black square have resolutions of 128×128 (single texture images) and 256×256 (two texture image). All output texture have a resolution of 512×512 .

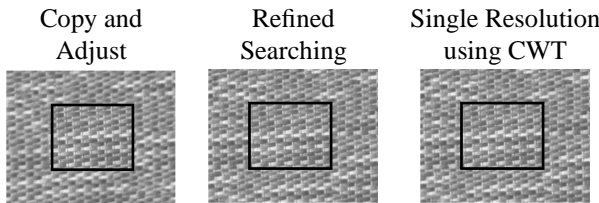


Fig. 3. Comparing the three methods for updating wavelet coefficients. The original texture “seed” is shown inside the black square.

the advantages of scale independence and low computational cost. It compares well with the best previous approaches at a fraction of the computation load. It is suitable for both growing texture from an initial “seed” and can be adjusted to work on filling-in missing, i.e. texture synthesis with boundary conditions.

7. REFERENCES

- [1] Alexei A. Efros and Thomas K. Leung, “Texture synthesis by non-parametric sampling,” in *IEEE International Conference on Computer Vision*, Corfu, Greece, September 1999, pp. 1033–1038.
- [2] Li-Yi Wei and Marc Levoy, “Fast texture synthesis using tree-structured vector quantization,” *Proceedings of ACM SIGGRAPH 2000*, pp. 479–488, 2000.
- [3] Anil Kokaram, “Parametric texture synthesis for filling holes in pictures,” in *IEEE International Conference on Image Processing*, Rochester, New York, USA, September 2002, pp. 325–328.

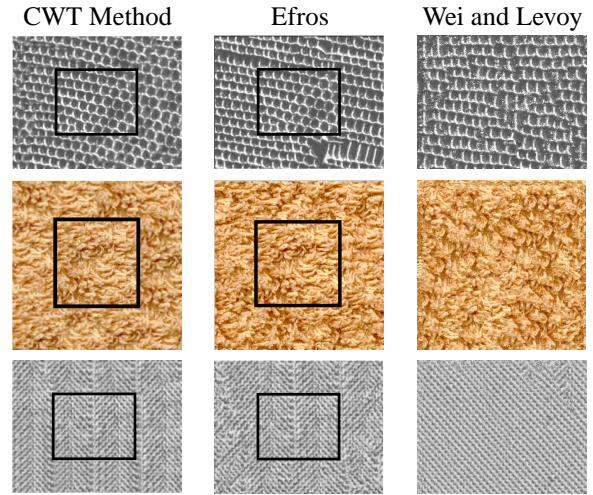


Fig. 4. Comparing the Texture Synthesis algorithms. The texture “seed” is shown inside the black square. Note the garbage generation problem associated with the Efros results (row 1, column 2). The Wei and Levoy method transforms a random noise sample to resemble the texture and is therefore not initialised with an initial texture “seed”.

- [4] Patrizio Campisi and Gaetano Scarano, “A multiresolution approach for texture synthesis using the circular harmonic functions,” *IEEE Transactions on Image Processing*, vol. 11, no. 1, January 2002.
- [5] Simon Robinson Bill Collis and Paul White, “Wire removal,” in *The IEE 1st European Conference on Visual Media Production (CVMP)*, March 15-16, London, UK, 2004, pp. 133–138.
- [6] D.J. Heeger and J.R. Bergen, “Pyramid based texture analysis or synthesis,” *SIGGRAPH 1995 Conference Proceedings, Annual Conference Series, ACM SIGGRAPH, Addison Wesley*, pp. 229–238, August 1995.
- [7] J. Portilla and E.P. Simoncelli, “A parametric texture model based on joint statistics of complex wavelet coefficients,” *Int’l Journal of Computer Vision*, vol.40(1), pp. 49–71, December 2000.
- [8] Michael Ashikhmin, “Synthesizing natural textures,” in *ACM Symposium on Interactive 3D Graphics*, Research Triangle Park, North Carolina, USA, March 2001, pp. 217–226.
- [9] Kris Popat and Rosalind W. Picard, “A novel cluster-based probability model for texture synthesis, classification, and compression,” in *Proc. SPIE Visual Communications ’93*, Cambridge, Mass., 1993, pp. 756–768.
- [10] Nick Kingsbury, “Shift invariant properties of the dual-tree complex wavelet transform,” in *Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing*, March 1999.