

Research Article

A New Texture Synthesis Algorithm Based on Wavelet Packet Tree

**Hsi Chin Hsin,¹ Tze-Yun Sung,² Yaw-Shih Shieh,²
and Carlo Cattani³**

¹ *Department of Computer Science and Information Engineering, National United University,
Miaoli 36003, Taiwan*

² *Department of Electronics Engineering, Chung Hua University, Hsinchu City 30012, Taiwan*

³ *Department of Mathematics, University of Salerno, Via Ponte Don Melillo,
84084 Fisciano, Italy*

Correspondence should be addressed to Tze-Yun Sung, bobsung@chu.edu.tw

Received 14 February 2012; Accepted 15 March 2012

Academic Editor: Ming Li

Copyright © 2012 Hsi Chin Hsin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents an efficient texture synthesis based on wavelet packet tree (TSWPT). It has the advantage of using a multiresolution representation with a greater diversity of bases functions for the nonlinear time series applications such as fractal images. The input image is decomposed into wavelet packet coefficients, which are rearranged and organized to form hierarchical trees called wavelet packet trees. A 2-step matching, that is, coarse matching based on low-frequency wavelet packet coefficients followed by fine matching based on middle-high-frequency wavelet packet coefficients, is proposed for texture synthesis. Experimental results show that the TSWPT algorithm is preferable, especially in terms of computation time.

1. Introduction

Texture modeling can be effectively applied to a wide variety of natural surfaces such as plants, furs, skins, minerals, terrains, and fractal materials [1, 2] and is an important issue in cyber-physical systems [3–7]. Numerous techniques have been proposed for texture processing; one may refer to [8] for a complete survey.

Given an example texture, the goal of texture synthesis is to produce a visually similar image of any size. One may easily tile small textures to synthesize a larger image; however, there are some blocking effects near the tile edges [9]. Although some smoothing methods were proposed to reduce the blocking effects at the cost of computation time, it seems to be

limited in improvements, especially for structured textures [10]. Efros and Leung proposed a neighborhood matching method [11], in which each pixel of the synthesis image was obtained by searching the most similar one in the source image based on the user-defined neighboring pixels. Wei and Levoy took account of the order in which pixels were synthesized and proposed an order-independent search-based texture synthesis algorithm [12]. Instead of synthesizing one pixel at a time, a patch of pixels can be taken as a whole and synthesized by matching the overlap regions between neighboring patches [13–15]. To improve the synthesis speed, the multi-resolution approach has been widely used for texture analysis and synthesis [16, 17]; Fang and Lien developed a rapid image synthesis system by adopting a multi-resolution approach, which consisted of an analysis process and a synthesis process. It consumed most of the computation time in the analysis process; yet the speed of synthesis was very fast [16]. In [17], De Bonet proposed a scheme of generating the synthesis image by sampling the filtered outputs of a texture in the framework of Laplacian pyramid [18]. Burt adopted the Gaussian pyramid [19] to represent both the input texture and the synthesis image at multiple resolutions and synthesized the texture images from lower to higher resolutions [20].

Wavelet transform provides an efficient multi-resolution representation [21], in which the higher frequency components of an image are projected onto the shorter basis functions with higher spatial resolutions and the lower frequency components are projected onto the larger basis functions with higher spectral resolutions. Such a compact representation matches the characteristics of human visual system [22]. Wavelet theory has been successfully applied to many applications such as parameter estimation in fractal signals and images [23–29]. Yu et al. randomly sampled blocks of wavelet coefficients from the input texture to substitute that of the synthesis image [30]. Cui et al. adopted a 2-level wavelet transform and generated the synthesis image by minimizing the sum of squared distances between neighboring blocks of wavelet coefficients [31].

For images with textures, lots of wavelet coefficients are likely to be significant in the middle-high frequency subbands, which surely demand further decompositions for a more compact representation [32]. Note that wavelet transform only decomposes the low-frequency component of an image at each resolution. However, both the low-frequency and high-frequency components can be decomposed using wavelet packet transform, which provides more bases functions than wavelet transform [33]. In [34], we proposed an efficient scheme to organize the wavelet packet coefficients of an image into hierarchical trees called wavelet packet (WP) trees for image compression. In this paper, an efficient WP tree-based algorithm is proposed for the texture synthesis applications.

The remainder of this paper proceeds as follows. In Section 2, wavelet transform and wavelet packet transform are briefly reviewed. Section 3 describes the proposed scheme to synthesize a texture image based on the WP trees of an example texture. Experimental results are presented in Section 4. Conclusion can be found in Section 5.

2. Review of Wavelet Transform and Wavelet Packet Transform

Wavelet-transform- (WT-) based multiresolution analysis/synthesis has drawn a lot of attention to the signal/image/video applications. The extension of WT known as wavelet packet transform (WPT) provides a much larger family of bases functions with a more compact representation. In this section, a brief review of WT and WPT is given. For a more complete survey, we refer readers to [21].

2.1. Wavelet Transform

WT has a variety of desirable properties, for example, joint space-spatial frequency localization, self-similarity across subbands of the same orientation, and energy clustering within each subband. For a discrete signal at resolution ℓ , $S_\ell(n)$, the wavelet transform is defined as

$$\begin{aligned} S_{\ell+1}(n) &= \sum_k h(2n-k) \cdot S_\ell(k), \\ D_{\ell+1}(n) &= \sum_k g(2n-k) \cdot S_\ell(k), \end{aligned} \quad (2.1)$$

where

$$\begin{aligned} h(n) &= \langle \phi, \phi_{-1,-n} \rangle, \\ g(n) &= \langle \psi, \phi_{-1,-n} \rangle, \\ \phi_{-1,-n}(x) &= 2^{-1/2} \phi(2^{-1}x - n). \end{aligned} \quad (2.2)$$

ψ is a (mother) wavelet, ϕ is the corresponding scaling function, $S_{\ell+1}(n)$ is the approximation signal at the next coarser resolution $\ell+1$, $D_{\ell+1}(n)$ is the detail information between resolutions ℓ and $\ell+1$, $h(n)$ and $g(n)$ are low-pass filter and high-pass filter, respectively, and $\langle \cdot, \cdot \rangle$ is an inner product operator. $S_\ell(n)$ can be exactly reconstructed from $S_{\ell+1}(n)$ and $D_{\ell+1}(n)$ by using the following inverse wavelet transform (IWT):

$$S_\ell(n) = \sum_k \tilde{h}(n-2k) \cdot S_{\ell+1}(k) + \sum_k \tilde{g}(n-2k) \cdot D_{\ell+1}(k), \quad (2.3)$$

where $\tilde{h}(n) = h(-n)$ and $\tilde{g}(n) = g(-n)$.

For the image applications, 2D WT can be obtained by using a tensor product of two 1D WTs, horizontally followed by vertically or vice versa. Specifically, let $LL_0(m, n)$ be an image at the finest resolution 0, where m and n are indices for the vertical and horizontal orientations, respectively. The 2D WT of $LL_0(m, n)$ is as follows:

$$\begin{aligned} LL_1(m, n) &= \sum_i \sum_j h(i) \cdot h(j) \cdot LL_0(2m-i, 2n-j), \\ LH_1(m, n) &= \sum_i \sum_j h(i) \cdot g(j) \cdot LL_0(2m-i, 2n-j), \\ HL_1(m, n) &= \sum_i \sum_j g(i) \cdot h(j) \cdot LL_0(2m-i, 2n-j), \\ HH_1(m, n) &= \sum_i \sum_j g(i) \cdot g(j) \cdot LL_0(2m-i, 2n-j), \end{aligned} \quad (2.4)$$

where $LL_1(m, n)$ is the approximation image at the next coarser resolution 1, $LH_1(m, n)$, $HL_1(m, n)$, and $HH_1(m, n)$ are the detail images in the vertical, horizontal, and diagonal orientations, respectively. Figure 1 shows a 2-level 2D WT, where subbands LH_ℓ , HL_ℓ , HH_ℓ ; $\ell = 1, 2$, and LL_2 are delimited by solid lines. Similarly, 2D IWT can be obtained by using the tensor product of two 1D IWTs.

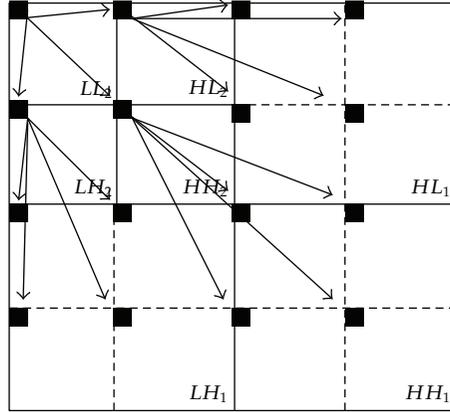


Figure 1: 2-level wavelet transform (solid boundaries), wavelet packet transform (dashed boundaries), and a wavelet packet tree with wavelet packet coefficients connected by arrows.

2.2. Wavelet Packet Transform

2D WT is only to decompose the lowest-frequency subband of an image in an iterative manner. More specifically, only the scaling coefficients are decomposed from higher to lower resolutions. However, for the texture applications, wavelet coefficients in the middle- and high-frequency subbands are likely to be significant, which needs to be taken into account to improve the multiresolution representation.

As one can see, both the low-frequency scaling coefficients and high-frequency wavelet coefficients of a signal, at any resolution, can be decomposed, which leads to wavelet packet transform (WPT), and a much larger family of bases functions can be produced [33]. Moreover, 2D WPT can be obtained by using a tensor product of two 1D WPTs. Figure 1 shows a 2-level 2D WPT, where all the wavelet subbands LH_1 , HL_1 , and HH_1 , at resolution 1, are further decomposed into wavelet packet subbands delimited by dashed lines.

3. Wavelet-Packet-Tree-Based Texture Synthesis

Wavelet packet transform provides more bases functions, which leads to a more compact representation in comparison with wavelet transform. For the image-coding applications, we had proposed an efficient scheme to organize the wavelet packet (WP) coefficients of an image into hierarchical trees called WP trees [32]. In this section, we explore the key features of WP trees and propose a WP-tree-based algorithm for texture synthesis.

3.1. Wavelet Packet Tree

The WP coefficients of a sequence of wavelet coefficients $D_\ell(n)$, at resolution ℓ , are computed by

$$\begin{aligned}\tilde{D}_{\ell,1}(n) &= \sum_k h(2n-k) \cdot D_\ell(k), \\ \tilde{D}_{\ell,2}(n) &= \sum_k g(2n-k) \cdot D_\ell(k),\end{aligned}\tag{3.1}$$

which can be efficiently rearranged and concatenated by

$$\tilde{D}_\ell(n) = \begin{cases} \tilde{D}_{\ell,1}\left(\frac{n}{2}\right); & \text{even } n, \\ \tilde{D}_{\ell,2}\left(\frac{n-1}{2}\right); & \text{odd } n. \end{cases} \quad (3.2)$$

After the rearrangement and concatenation above, the dyadic relationship of wavelet coefficients across subbands is still valid and can be used to construct the dyadic WP trees of a signal. Similarly, the (2D) WP trees of an image can be obtained by rearranging the WP coefficients horizontally followed by vertically or vice versa. Figure 1 shows a 2-level WP tree with arrows connecting the related WP coefficients.

The key idea behind the construction of WP trees is based on the spatial relationships of WP coefficients. It has the same structure, that is, quad-tree structure, as the conventional wavelet trees. Furthermore, the number of high-energy wavelet coefficients can be significantly reduced through the use of wavelet packet transform. Take the rice image shown in Figure 5 as an example. The cumulative energy distribution (CED) of wavelet coefficients or WP coefficients is given by

$$\text{CED}(n) = \frac{\sum_{i=1}^n |C(i)|}{\sum_{i=1}^N |C(i)|} \times 100\%, \quad (3.3)$$

where $|C(i)|, i = 1, 2, \dots, N$, is the sorted magnitudes of wavelet coefficients or WP coefficients in descending order, n is the number of coefficients, and N is the total number of coefficients. Figure 2 shows the CED curves of wavelet coefficients and WP coefficients, where the horizontal and vertical axes are the number of coefficients and percentage of energy, respectively. It is noted that the energy clustering of WP coefficients is more compact than wavelet coefficients. As a result, the WP-based representation is preferable to the wavelet-based representation for texture images.

3.2. Proposed Algorithm

As noted, the low-frequency WP coefficients retain the global information of an image, and the high-frequency WP coefficients contain the local detail. It is desirable to coarsely synthesize an image based on the low-frequency WP coefficients and then tune the intermediate synthesis result based on the high-frequency WP coefficients. Motivated by the fact above, we propose an efficient WP-tree-based texture synthesis algorithm using a two-step process: a coarse searching followed by a fine tuning. Figure 2 depicts a flowchart of the proposed algorithm. It is presented in steps as follows.

Step 1 (initialization). Decompose the source image by wavelet packet transform, rearrange the high-frequency WP coefficients, and construct the WP trees. Randomly select a WP tree from the source image, which is replicated in the upper left corner of the synthesis image.

Step 2 (coarse matching). For every WP tree to be synthesized, starting from the upper left corner to the lower right corner of the synthesis image, search the candidate WP trees from the source image by using a coarse matching with a tolerance as follows:

$$\begin{aligned}
CWP_j &= \{WP_{source,i} \mid err_{LFN}(i,j) \leq tol_j, \forall i\}, \\
tol_j &= \min_i err_{LFN}(i,j) \cdot (1 + T_r), \\
err_{LFN}(i,j) &= \sum_{p \in N_i, q \in N_j} dist_{LFN}(p,q), \\
dist_{LFN}(p,q) &= \sum_{m,n \in LFN} (WP_{source,p}(m,n) - WP_{synthesis,q}(m,n))^2,
\end{aligned} \tag{3.4}$$

where CWP_j is the set of candidate WP trees obtained from the source image for matching the synthesis WP tree $WP_{synthesis,j}$, with the tolerance tol_j , T_r is a given threshold, $dist_{LFN}(p,q)$ is the distance between $WP_{source,p}$ and $WP_{synthesis,q}$ based on the low-frequency nodes (LFNs) of WP trees, and N_i and N_j are neighbors of $WP_{source,i}$ and $WP_{synthesis,j}$, respectively. Causal neighborhoods were used in our experiments.

Step 3 (fine matching). After the coarse matching in Step 2, the following fine matching is used to find the best WP tree based on the high-frequency nodes (HFNs) of the candidate WP trees.

$$\begin{aligned}
WP_{synthesis,j} &= WP_{source,i}; \quad i = \arg\left(\min_i err_{HFN}(i,j)\right), \\
err_{HFN}(i,j) &= \sum_{p \in N_i, q \in N_j} dist_{HFN}(p,q), \\
dist_{HFN}(p,q) &= \sum_{m,n \in HFN} (WP_{source,p}(m,n) - WP_{synthesis,q}(m,n))^2,
\end{aligned} \tag{3.5}$$

where $WP_{source,i} \in CWP_j$ in (3.5).

Step 4. Repeat Step 2 followed by Step 3 until all the WP trees to be synthesized are obtained from the WP trees of the source image.

Step 5. Finally, the synthesis image is obtained by taking the inverse wavelet packet transform of the synthesis WP trees.

To reduce the synthesis time, one can easily modify Steps 2 and 3 by using patches of WP trees instead of single WP trees. Moreover, it is noted that lots of high-frequency WP coefficients are not significant, and only a small portion of middle-frequency WP coefficients are sufficient for the fine matching in Step 3. Thus, the proposed texture synthesis based on wavelet packet tree (TSWPT) algorithm is simple and computationally efficient. Flowchart of the TSWPT algorithm is shown in Figure 3.

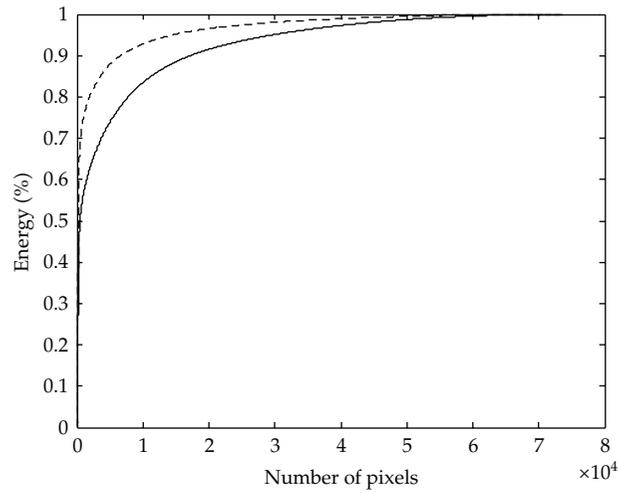


Figure 2: Cumulative energy distributions (CEDs) of the wavelet coefficients (solid line) and wavelet packet coefficients (dashed line) of the rice image shown in Figure 5.

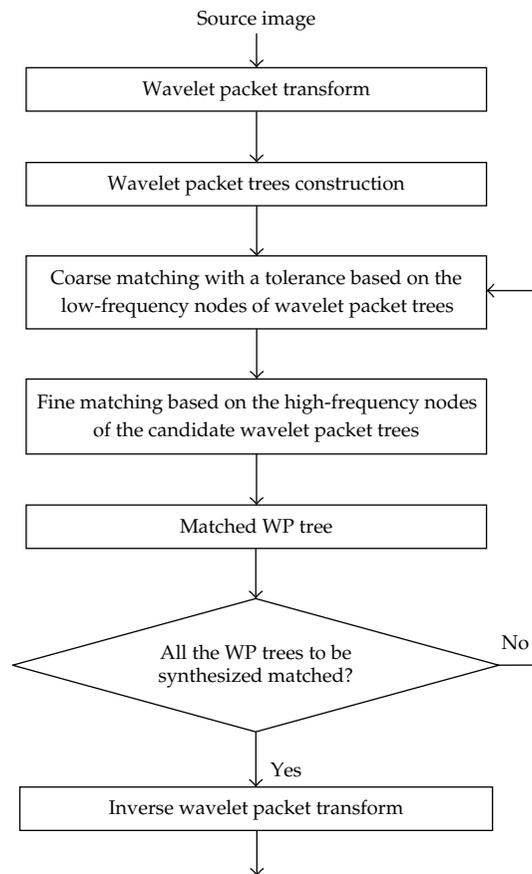


Figure 3: Flowchart of the TSWPT algorithm.

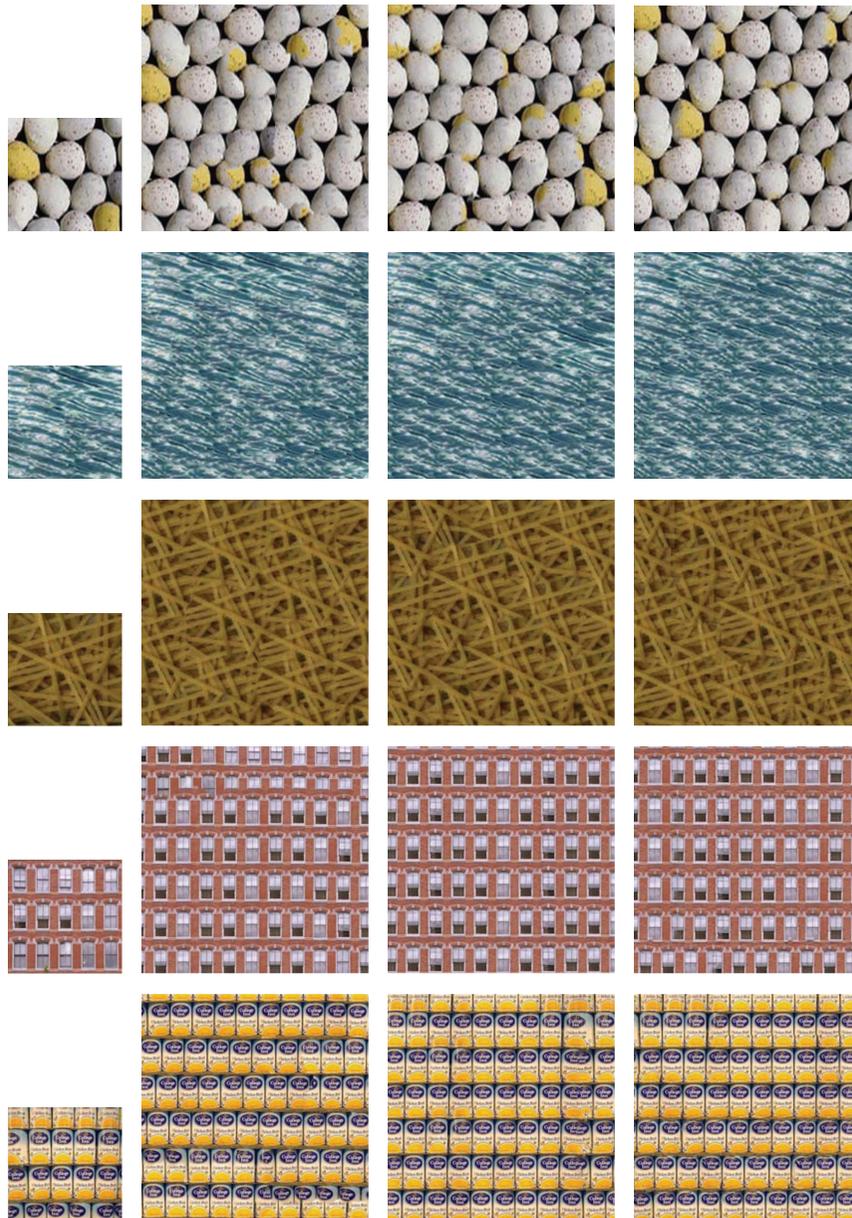


Figure 4: Visual comparison. Source images (1st column), synthesis images using Efron's algorithm [13] (2nd column), Cui's algorithm [31] (3rd column), and the TSWPT algorithm (4th column).

4. Experimental Results

In the first experiment, the size of source images is 128×128 , which are shown in the 1st column of Figure 4; the size of synthesis images is 256×256 . The TSWPT algorithm is applied to patches of WP trees in order to reduce computation time. The size of patches is 11×11 , and the width of overlapped regions between neighboring patches is 3. The root nodes of WP trees, that is, WP coefficients in the lowest frequency subband, are used in the coarse matching



Figure 5: Synthesis images at double the size of the source image (top) using Efron's algorithm [13] (middle) and the TSWPT algorithm (bottom).

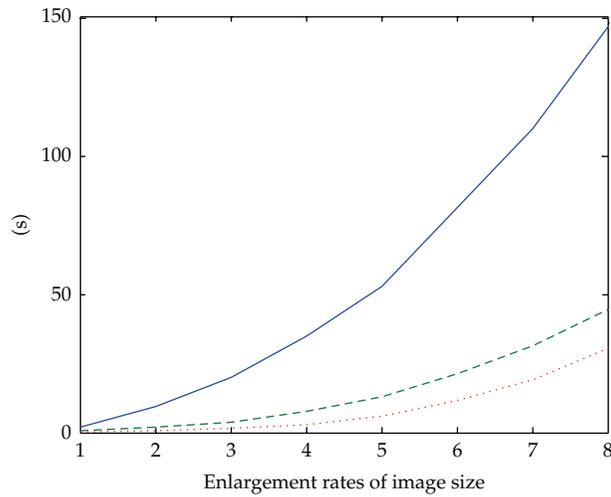
step to retain the global appearance of the input source image. Since the essential components of textures are mainly distributed in the middle-frequency subbands, the highest-frequency WP coefficients, that is, the leaf nodes of WP trees, can be ignored in the fine matching step. The biorthogonal 9/7 wavelet is used. The threshold value, T_r , is set to 0.1.

Two widely used algorithms, Efron's algorithm [13] and Cui's algorithm [31], were used for comparisons with the TSWPT algorithm. The synthesis results are shown in the 2nd, 3rd, and 4th columns of Figure 4, respectively. Visual inspection shows that TSWPT is comparable to Cui's algorithm and is preferable to Efron's algorithm. All the algorithms were implemented by Matlab without optimization in source codes. Table 1 shows the computation times running on PC with CPU of 1.7 GHz. It is noted that TSWPT is superior to both Efron's and Cui's algorithms.

In the second experiment, the sizes of source image and synthesis image are 192×128 and 384×256 , respectively. All the settings are the same as the first experiment. The source image and synthesis results using Efron's algorithm and TSWPT are shown in Figure 5. It is shown that there are some blocking effects in the synthesis image using Efron's algorithm. However, blocking effects are likely to be eliminated by using the TSWPT algorithm due largely to the filtering operations of inverse wavelet packet transform in Step 5. Figure 6

Table 1: Comparison of computation times.

	Efros and freeman [13]	Cui et al. [31]	TSWPT
Image 1	3.784 s	0.703 s	0.416 s
Image 2	3.893 s	0.854 s	0.425 s
Image 3	3.939 s	0.810 s	0.404 s
Image 4	3.914 s	0.737 s	0.372 s
Image 5	3.891 s	0.731 s	0.392 s

**Figure 6:** Synthesis times of the texture image (Figure 5) using Efros' algorithm [13] (solid line), Cui's algorithm [31] (dashed line), and the TSWPT algorithm (dotted line).

shows the computation times at various enlargement rates of image size. As one can see, the TSWPT algorithm outperforms both Efros' and Cui's algorithms.

5. Conclusion

The multi-resolution approach is suitable for texture synthesis in terms of computation time. Wavelet packet transform provides more bases functions than wavelet transform and therefore produces a more compact representation. We adopt wavelet packet transform to analyze the input texture and organize wavelet packet coefficients to form hierarchical trees called wavelet packet trees. The low-frequency nodes of wavelet packet trees contain the global characteristics of an image; the high-frequency nodes contain the local details. Thus, we propose texture synthesis based on wavelet packet tree (TSWPT). It has the advantage of saving computation time dramatically, and moreover, no training process is needed. Given a 128×128 texture, experimental results show that the computation time for synthesizing an 256×256 image is only a fraction of a second.

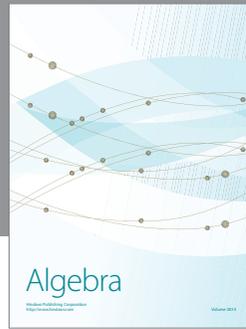
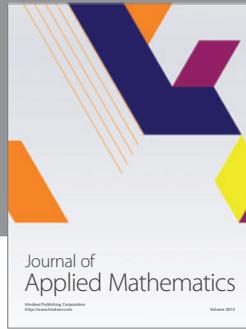
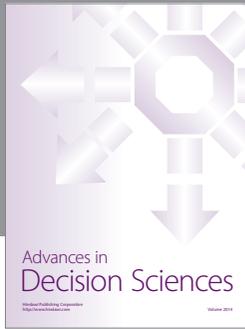
Acknowledgment

The National Science Council of Taiwan, under Grants NSC100-2628-E-239-002-MY2, supported this work.

References

- [1] M. Li, "Fractal time series—a tutorial review," *Mathematical Problems in Engineering*, vol. 2010, Article ID 157264, 26 pages, 2010.
- [2] C. Cattani, G. Pierro, and G. Altieri, "Entropy and multifractality for the myeloma multiple TET 2 gene," *Mathematical Problems in Engineering*, vol. 2012, Article ID 193761, 14 pages, 2012.
- [3] M. Li and W. Zhao, "Visiting power laws in cyber-physical networking systems," *Mathematical Problems in Engineering*, vol. 2012, Article ID 302786, 13 pages, 2012.
- [4] A. Hausner, "Simulating decorative mosaics," in *Computer Graphics Annual Conference (SIGGRAPH '01)*, pp. 573–580, August 2001.
- [5] F. Neyret, "Modeling, animating, and rendering complex scenes using volumetric textures," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 1, pp. 55–70, 1998.
- [6] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H. Y. Shum, "Synthesis of bidirectional texture functions on arbitrary surfaces," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 665–672, 2002.
- [7] E. Risser, C. Han, R. Dahyot, and E. Grinspun, "Synthesizing structured image hybrids," *ACM Transactions on Graphics*, vol. 29, no. 4, article no. 85, 2010.
- [8] N. Pietroni, P. Cignoni, M. Otaduy, and R. Scopigno, "Solid-texture synthesis: a survey," *IEEE Computer Graphics and Applications*, vol. 30, no. 4, pp. 74–89, 2010.
- [9] Y. Xu, B. Guo, and H. Y. Shum, "Chaos mosaic: fast and memory efficient texture synthesis," MSR-TR-2000-32, Microsoft Research, 2000.
- [10] J. Portilla and E. P. Simoncelli, "Parametric texture model based on joint statistics of complex wavelet coefficients," *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–71, 2000.
- [11] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *7th IEEE International Conference on Computer Vision (ICCV '99)*, pp. 1033–1038, September 1999.
- [12] L. Y. Wei and M. Levoy, "Order independent texture synthesis," Stanford Computer Science TR-2002-01, 2002.
- [13] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Computer Graphics Annual Conference (SIGGRAPH '01)*, pp. 341–346, August 2001.
- [14] L. Liang, C. Liu, Y. Xu, B. Guo, and H. Y. Shum, "Real-time texture synthesis using patch-based sampling," *ACM Transactions on Graphics*, vol. 20, no. 3, pp. 127–150, 2001.
- [15] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 277–286, 2003.
- [16] C.-W. Fang and J.-J. J. Lien, "Rapid image completion system using multiresolution patch-based directional and nondirectional approaches," *IEEE Transactions on Image Processing*, vol. 18, no. 12, pp. 2769–2779, 2009.
- [17] J. S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Conference on Computer Graphics (SIGGRAPH '97)*, pp. 361–368, August 1997.
- [18] P. J. Burt and E. H. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983.
- [19] P. J. Burt, "Fast algorithms for estimating local image properties.," *Computer Vision, Graphics, & Image Processing*, vol. 21, no. 3, pp. 368–382, 1983.
- [20] L. Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Computer Graphics Annual Conference (SIGGRAPH '00)*, pp. 479–488, July 2000.
- [21] G. Strang and T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, Mass, USA, 1996.
- [22] C. K. Su, H. C. Hsin, and S. F. Lin, "Wavelet tree classification and hybrid coding for image compression," *IEE Proceedings—Vision, Image, and Signal Processing*, vol. 152, no. 6, pp. 752–756, 2005.
- [23] C. Cattani and J. Rushchitsky, *Wavelet and Wave Analysis as Applied to Materials with Micro or Nanostructure*, vol. 74 of *Series on Advances in Mathematics for Applied Sciences*, World Scientific, Hackensack, NJ, USA, 2007.

- [24] Z. W. Liao, S. X. Hu, M. Li, and W. Chen, "Noise estimation for single-slice sinogram of low-dose X-ray computed tomography using homogenous patch," *Mathematical Problems in Engineering*, vol. 2012, Article ID 696212, 16 pages, 2012.
- [25] M. Li, C. Cattani, and S. Y. Chen, "Viewing sea level by a one-dimensional random function with long memory," *Mathematical Problems in Engineering*, vol. 2011, Article ID 654284, 13 pages, 2011.
- [26] J. W. Yanga, M. Li, Z. Chen, and Y. Chen, "Cutting affine invariant moments," *Mathematical Problems in Engineering*. In press.
- [27] B. Chen and W.-S. Chen, "Noisy image segmentation based on wavelet transform and active contour model," *Applicable Analysis*, vol. 90, no. 8, pp. 1243–1255, 2011.
- [28] B. Chen, P. C. Yuen, J. H. Lai, and W. S. Chen, "Image segmentation and selective smoothing based on variational framework," *Journal of Signal Processing Systems*, vol. 54, no. 1–3, pp. 145–158, 2009.
- [29] B. Chen, Y. Li, and J. L. Cai, "Noisy image segmentation based on nonlinear diffusion equation model," *Applied Mathematical Modeling*, vol. 36, no. 3, pp. 1197–1208, 2012.
- [30] Y. Yu, J. Luo, and C. W. Chen, "Multiresolution block sampling-based method for texture synthesis," in *16th International Conference on Pattern Recognition (ICPR '02)*, pp. 239–242, 2002.
- [31] H. F. Cui, X. Zheng, and T. Ruan, "An efficient texture synthesis algorithm based on WT," in *7th International Conference on Machine Learning and Cybernetics (ICMLC '08)*, pp. 3472–3477, July 2008.
- [32] T. Y. Sung and H. C. Hsin, "An efficient rearrangement of wavelet packet coefficients for embedded image coding based on SPIHT algorithm," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E90-A, no. 9, pp. 2014–2020, 2007.
- [33] N. M. Rajpoot, R. G. Wilson, F. G. Meyer, and R. R. Coifman, "Adaptive wavelet packet basis selection for zerotree image coding," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1460–1472, 2003.
- [34] H. C. Hsin and T. Y. Sung, "Adaptive selection and rearrangement of wavelet packets for quad-tree image coding," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E91-A, no. 9, pp. 2655–2662, 2008.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

