Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 416186, 8 pages
http://dx.doi.org/10.1155/2013/416186

Research Article

Fast Texture Synthesis in Adaptive Wavelet Packet Trees

Ying-Shen Juang,[1] Hsi-Chin Hsin,[2] Tze-Yun Sung,[3] and Carlo Cattani[4]

[1] *Department of Business Administration, Chung Hua University, Hsinchu City 30012, Taiwan*
[2] *Department of Computer Science and Information Engineering, National United University, Miaoli 36003, Taiwan*
[3] *Department of Electronics Engineering, Chung Hua University, Hsinchu City 30012, Taiwan*
[4] *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 21 January 2013; Accepted 18 February 2013

Academic Editor: Shengyong Chen

Wavelet packet transform known as a substantial extension of wavelet transform has drawn a lot of attention to visual applications. In this paper, we advocate using adaptive wavelet packet transform for texture synthesis. The adaptive wavelet packet coefficients of an image are organized into hierarchical trees called adaptive wavelet packet trees, based on which an efficient algorithm has been proposed to speed up the synthesis process, from the low-frequency tree nodes representing the global characteristics of textures to the high-frequency tree nodes representing the local details. Experimental results show that the texture synthesis in the adaptive wavelet packet trees (TSIAWPT) algorithm is suitable for a variety of textures and is preferable in terms of computation time.

1. Introduction

The goal of texture synthesis is to generate an arbitrarily large image that resembles a given sample texture in appearance. With the emerging market in various applications ranging from computer graphics, machine vision to advanced imaging products [1–4], texture synthesis has been one of the increasingly active areas of research in recent years [5–7]. Though it is difficult to precisely define textures, two main categories have been commonly used to describe textures as either stochastic or structural patterns. For more complete surveys of texture analysis and synthesis, the reader is referred to [1, 8, 9].

One of the straightforward ways to synthesize a large textured image is to duplicate a given sample texture and stitch them together. In order to ameliorate the blocking seams [10], Cheng proposed a seamless montage method with suitable tiles designed by using the quilting algorithm [11] and Wang Tiles [12] in [13]. With the assumption of Markov random field (MRF) [14, 15], Efros and Leung estimated the conditional distributions of output pixels from the input pixels that are similar to the neighbors for nonparametric texture synthesis [16]. Wei and Levoy proposed a search-based order-independent algorithm to synthesize textures in an arbitrary order [17]. As MRF modeling is local and stationary, textures can be efficiently synthesized patch by patch, instead of pixel by pixel [18]. To speed up the process of synthesizing textures, many multiresolution based-texture synthesis algorithms were proposed [19–26]. In [19], an MRF-based similarity metric was used to minimize the difference between the synthesis image and the sample texture from lower to higher resolutions. The algorithm presented in [20] is a discrete version of [19] with an improvement in speed. In [21], the lowest-resolution texture was first synthesized, based on which a set of four candidate pixels were examined to find the best synthesis pixel at the next higher-resolution level; this process repeatedly proceeded until the highest-resolution texture was obtained. Fang presented a fast multiresolution image completion algorithm with improved convergence of the synthesis process [22]. De Bonet used the Laplacian pyramid [23] together with a filter bank to capture the characteristics of the input texture at multiple resolutions and then synthesized the output image with an efficient resolution recursive sampling procedure [24]. Wei proposed a scheme to

synthesize textures in the Gaussian pyramid [25] through the use of tree-structured vector quantization [26].

Wavelet transform provides an efficient multiresolution analysis [27–30]. It decomposes an image into subbands with orientation selectivity, in which the higher-frequency components are represented by shorter basis functions with higher spatial resolutions, and the lower frequency components are represented by larger basis functions with higher spectral resolutions; this property matches the human visual system [31]. Yu et al. proposed a simple scheme to synthesize wavelet coefficients by sampling from the wavelet coefficients of the input texture [32]. Cui proposed texture synthesis based on the lowest-frequency scaling coefficients only; the corresponding higher-frequency wavelet coefficients were obtained accordingly [33]. For images with textures, there are significant coefficients throughout wavelet subbands [34]; this needs to be taken into account for a more compact representation. Wavelet packet transform extends wavelet transform by including more basis functions [27]. In [35], we adopted an efficient scheme to organize wavelet packet coefficients into hierarchical trees, based on which an efficient algorithm had been proposed for texture synthesis. Though wavelet packet transform is preferable to wavelet transform in terms of the representational diversity, the dominant components of an image may only be distributed in parts of the wavelet subbands. Hence, it is not necessary to decompose all of the wavelet subbands of an image into wavelet packets [36–38]. In this paper, an adaptive wavelet packet transform has been proposed to represent textures in the adaptive wavelet packet trees, based on which a fast texture synthesis algorithm has also been proposed.

The remainder of this paper proceeds as follows. In Section 2, the construction of hierarchical wavelet packet trees is briefly reviewed. Section 3 presents the proposed adaptive-wavelet-packet-tree-based texture synthesis algorithm. Experimental results are given in Section 4. A conclusion can be found in Section 5.

2. Hierarchical Wavelet Packet Trees

One of the advantages of using wavelet transform is to represent signals at multiple resolutions. For various decompositions with more basis functions, the high-frequency wavelet subbands of a signal can be further decomposed using wavelet packet transform. More specifically, a sequence of wavelet coefficients (WC) $D_\ell(k)$ at resolution ℓ can be decomposed by

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

where $h(n)$ and $g(n)$ are the low-pass and high-pass wavelet filters, respectively, and $\widetilde{D}_{\ell,1}(n)$ and $\widetilde{D}_{\ell,2}(n)$ are wavelet packet coefficients (WPC), which can be efficiently combined into a single sequence [35]:

$$\widetilde{D}_\ell(n) \begin{cases} \widetilde{D}_{\ell,1}\left(\dfrac{n}{2}\right), & \text{even } n, \\ \widetilde{D}_{\ell,2}\left(\dfrac{n-1}{2}\right), & \text{odd } n. \end{cases} \tag{2}$$

The combination of the above is based on the spatial relationships.

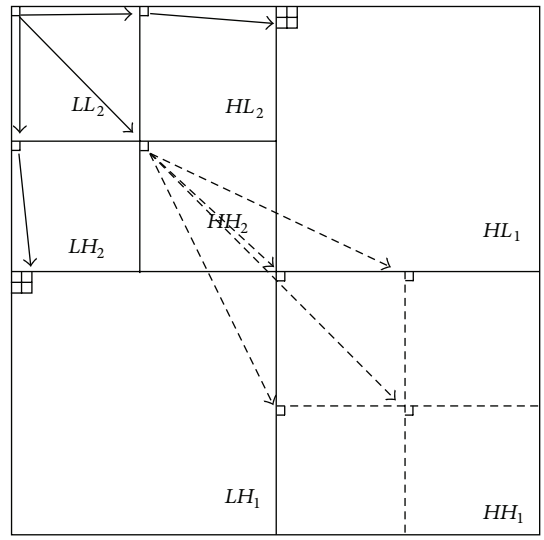

Figure 1: Example of an adaptive wavelet packet tree consisting of two wavelet subtrees in the horizontal and vertical directions (solid lines) and one wavelet packet subtree in the diagonal direction (dashed lines).

For image applications, 2D wavelet transform/wavelet packet transform can be obtained by using the tensor product of tow 1D wavelet transform/wavelet packet transform. Figure 1 shows a 2-level 2D wavelet transform, in which LL_2 is the lowest frequency subband representing the approximation of an image at the coarsest resolution 2; HL_ℓ, LH_ℓ, and HH_ℓ are the high frequency wavelet subbands representing the detail information in the horizontal, vertical, and diagonal directions, respectively, at resolution $\ell = 1, 2$; the wavelet subbands are delimited by solid lines, and two wavelet subtrees delineated by solid lines are also given to show the spatial relationships between subbands HL_1 and HL_2 (LH_1 and LH_2) at two successive resolution levels in the horizontal (vertical) direction. In addition, the wavelet subband HH_1 has been decomposed into 4 wavelet packet subbands (delimited by dashed lines), together with the subband HH_2; a 2D wavelet packet subtree delineated by dashed lines has been constructed by combining 2D WPC horizontally followed by vertically, or vice versa.

3. Adaptive Wavelet Packet Trees and Thier Application to Texture Synthesis

Wavelet packet transform has the advantage of representing images with more choices of basis functions. This section

Table 1: Numbers of significant wavelet coefficients (WC), wavelet packet coefficients (WPC), and adaptive wavelet packet coefficients (AWPC) from the most significant bit-plane 1 to the least significant bit-plane 8 and the reduced numbers of significant coefficients comparing WC to WPC and WPC to AWPC for the test image shown in Figure 2(a).

Bit plane	WC	WPC	AWPC	WC-WPC	WPC-AWPC
1	213	222	222	−9	0
2	1026	1084	1087	−58	−3
3	2558	2211	2250	347	−39
4	4647	3996	3880	651	116
5	6954	6271	5990	683	281
6	9663	9239	8858	424	381
7	12870	12616	12299	254	317
8	16683	16489	16264	194	225

presents an efficient algorithm to synthesize textures in the adaptive wavelet packet domain and a simple scheme to accelerate the synthesis process.

3.1. Adaptive Wavelet Packet Trees. For images with textures, it is likely that lots of wavelet coefficients are significant throughout the wavelet subbands, depending on the characteristics of the constituent textures. Though all of the wavelet subbands can be fully decomposed into wavelet packets, there may still be textures that are dominated by some of the wavelet subbands. Hence, we advocate the use of adaptive wavelet packet transform for texture synthesis. For the sake of simplicity, the following criterion has been used to determine the significance of a coefficient:

$$\mathrm{Sig}\left(P_i\left(x,y\right)\right) = \begin{cases} 1, & \dfrac{\left|P_i\left(x,y\right)\right|}{\max\left|P_i\left(x,y\right)\right|} > T_r, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where the magnitude of a coefficient $P_i(x, y)$ at position (x, y) in subband B_i is first normalized and then compared to a threshold T_r. For each non-LL subband, it is further decomposed until the number of significant coefficients has no longer decreased; this leads to the desired adaptive wavelet packet transform.

Take the image shown in Figure 2(a) as an example, Table 1 shows the numbers of significant wavelet coefficients (WC), wavelet packet coefficients (WPC), and adaptive wavelet packet coefficients (AWPC) with respect to threshold 2^{-j} at bit-plane j, $j = 1, \ldots, 8$, together with the reduced numbers of significant coefficients comparing WC to WPC and WPC to AWPC. As noted, the numbers of significant coefficients have been reduced starting form bit-plane 3 by using wavelet packet transform and reduced more starting from bit-plane 4 by using the proposed adaptive wavelet packet transform. It implies that the dominant components of high-detailed textures are likely to be more concentrated in the adaptive wavelet packet domain.

As only the significant wavelet subbands of an image are decomposed into wavelet packet subbands, the resulting adaptive wavelet packet trees may consist of both wavelet subtrees and wavelet packet subtrees. Figure 1 shows an adaptive wavelet packet tree consisting of two wavelet subtrees in the horizontal and vertical directions, and one wavelet packet subtree in the diagonal direction. It still retains the same hierarchical structure as the conventional wavelet trees.

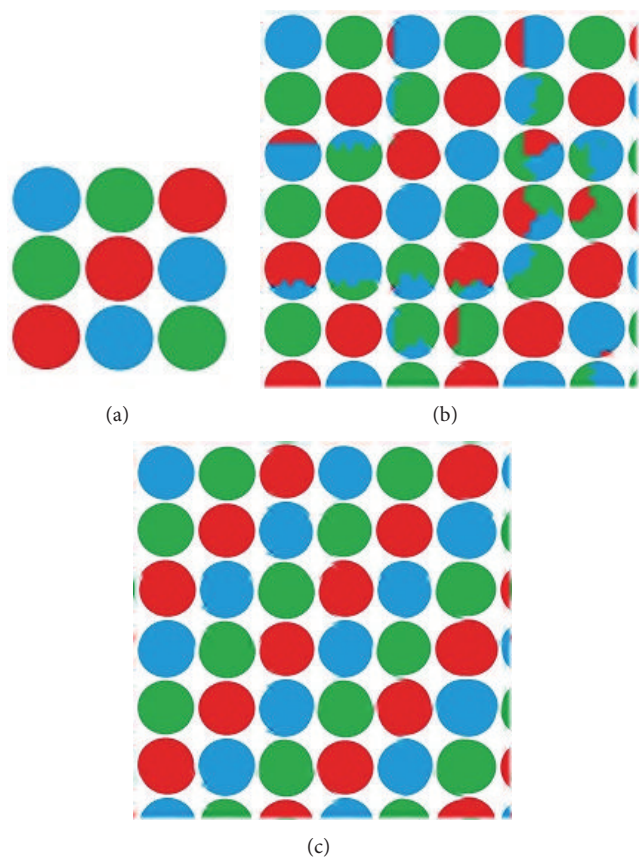

Figure 2: (a) Test image; (b) and (c) synthesis results using TSIWPT [35] and TSIAWPT.

3.2. Acceleration. In [35] we proposed the TSIWPT algorithm to synthesize a large textured image in wavelet packet trees. To reduce the computation time, TSIWPT minimizes the difference between the output wavelet packet trees and the input

wavelet packet trees based on their respective luminance components only. However, it may cause color distortions, especially for quasi-regular textures with different colors. For example; Figure 2(a) shows a regular texture consisting of blue, green, and red circular objects. The synthesis result using TSIWPT has some color mixes as shown in Figure 2(b). Though one can easily solve such color distortion problem by taking the color components into account directly while searching for the best output wavelet packet trees, it often leads to a great increase in computation time.

In this paper, we adopt the clustering approach to address the above speed and quality issue. The idea is to group the input adaptive wavelet packet trees into similarity sets, from which the best output adaptive wavelet packet trees are to be constructed. The running time can be reduced as the search of the best output adaptive wavelet packet trees is restricted to the similarity sets. In addition, the similarity sets of adaptive wavelet packet trees are constructed from the input adaptive wavelet packet trees; thus, a look-up table (LUT) can be used to store their locations in the input adaptive wavelet packet domain, which is independent of the size of the output image. As a result, the larger the synthesis image is, the greater the reduced computation time will be.

3.3. Proposed Algorithm. Figure 3 depicts a flowchart of the texture synthesis in adaptive wavelet packet trees (TSIAWPT) algorithm. We summarize TSIAWPT as follows.

Step 1. Construct the input the adaptive wavelet packet trees (AWPT) via adaptive wavelet packet transform.

Step 2. Cluster the input AWPT into similarity sets and build an LUT to store their locations.

Step 3. Take a patch of input AWPT as the initial patch of output AWPT. For the next patch of output AWPT to be synthesized, take the union of the neighboring similarity sets as the candidate set; evaluate the low-frequency tree nodes to refine the candidate set. Empirically, the tree nodes at the top two AWPT levels are suitable for the refinement of candidate sets.

Step 4. Search the refined candidate set for the best patch of output AWPT based on the high-frequency tree nodes.

Step 5. Repeat Step 3 followed by Step 4 until all the patches of output AWPT are synthesized.

Step 6. Take the inverse of the output AWPT to produce the synthesis image.

For a raster scan with rectangular patches, two types of neighboring patches, that is, the upper and left neighboring patches, are involved in clustering the input AWPT into similarity sets. The synthesized patches of output AWPT are thus obtained from the union of their respective upper and left neighboring similarity sets.

To determine a suitable patch size for textures with quasi-periodic structures, we adopt the use of autocorrelation

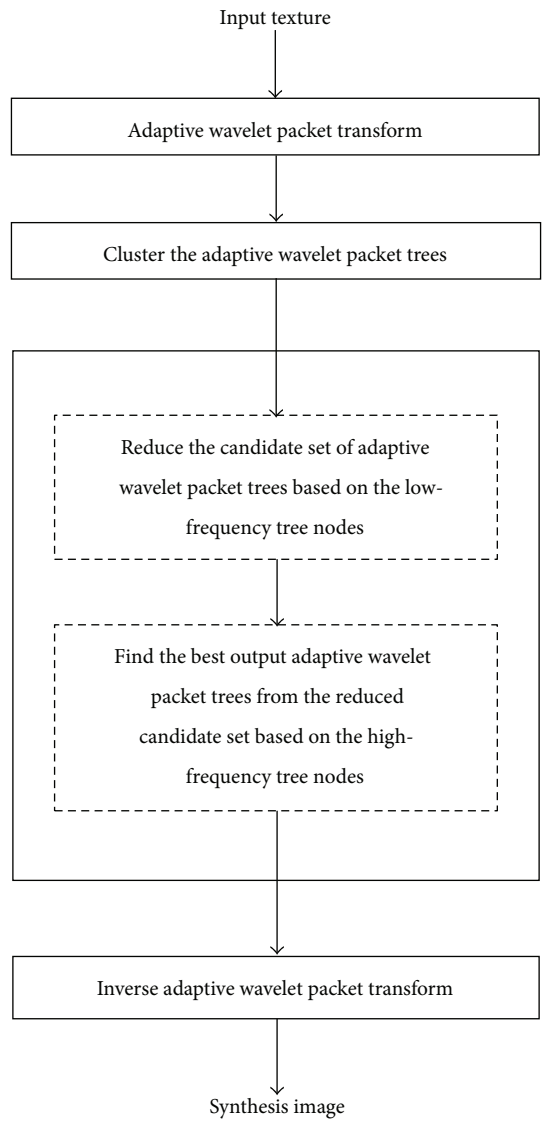

Figure 3: Flowchart of the TSIAWPT algorithm.

function (ACF) given by

$$R(m,n) = \frac{E\left[P_0(x,y)\,P_0(x+m,y+n)\right]}{E\left[P_0^2(x,y)\right]}, \tag{4}$$

where $P_0(x,y)$ is an adaptive wavelet packet coefficient at position (x,y) in the lowest-frequency subband 0 and m and n are the displacements in the horizontal and vertical directions, respectively. The ACF-based patch size $S_x \times S_y$ is determined by

$$\begin{gathered} S_x \times S_y = \arg\max_{m,n} R(m,n), \\ \left\lfloor \frac{M_0}{8} \right\rfloor \le m \le \left\lfloor \frac{M_0}{2} \right\rfloor, \qquad \left\lfloor \frac{N_0}{8} \right\rfloor \le n \le \left\lfloor \frac{N_0}{2} \right\rfloor, \end{gathered} \tag{5}$$

where $M_0 \times N_0$ is the size of the lowest-frequency subband of the input texture and $\lfloor z \rfloor$ denotes the largest integer less than z.

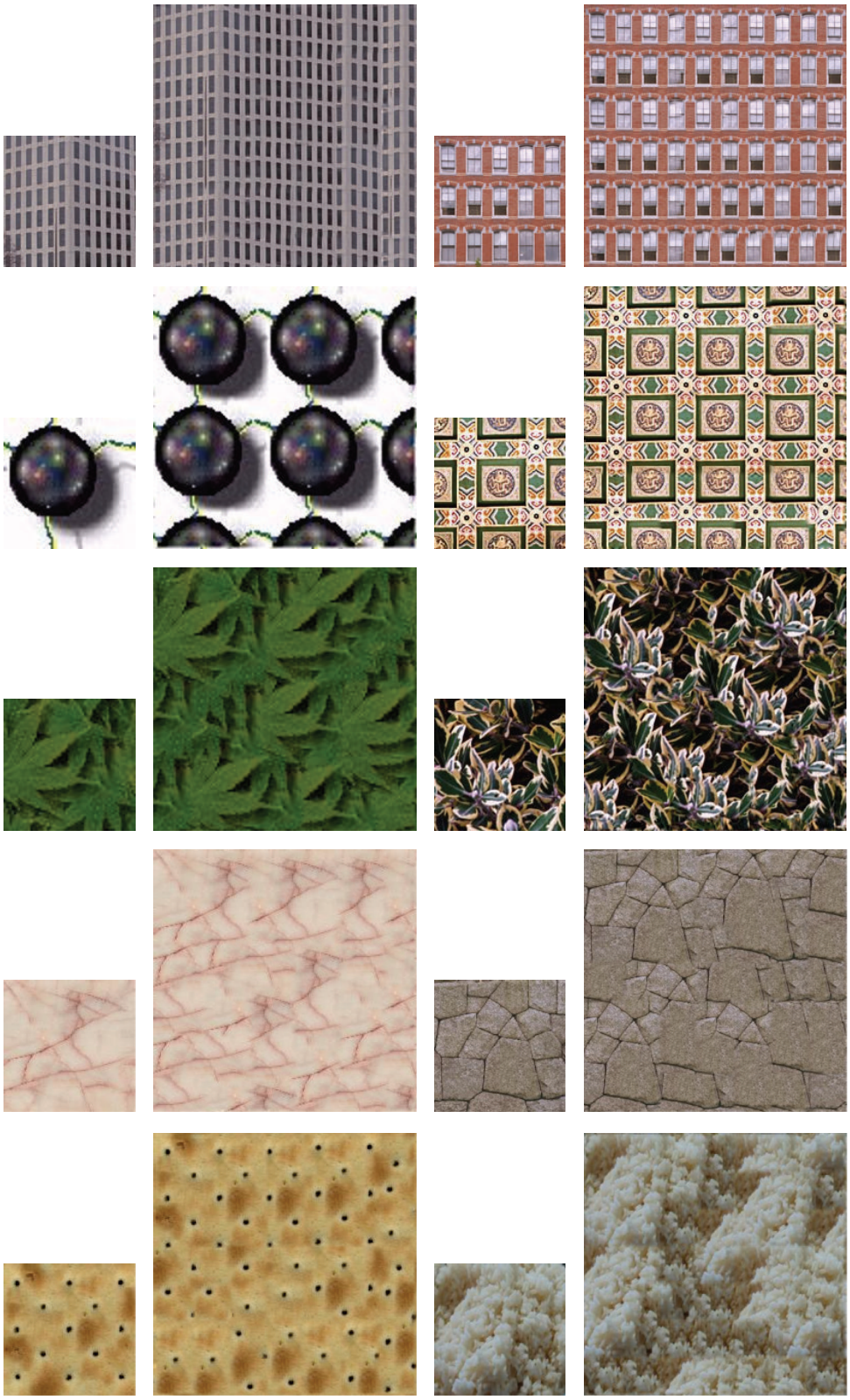

Figure 4: More synthesis results using TSIAWPT: architectures (1st row), regular structures (2nd row), leaves (3rd row), raw materials (4th row), and man-made materials (5th row).

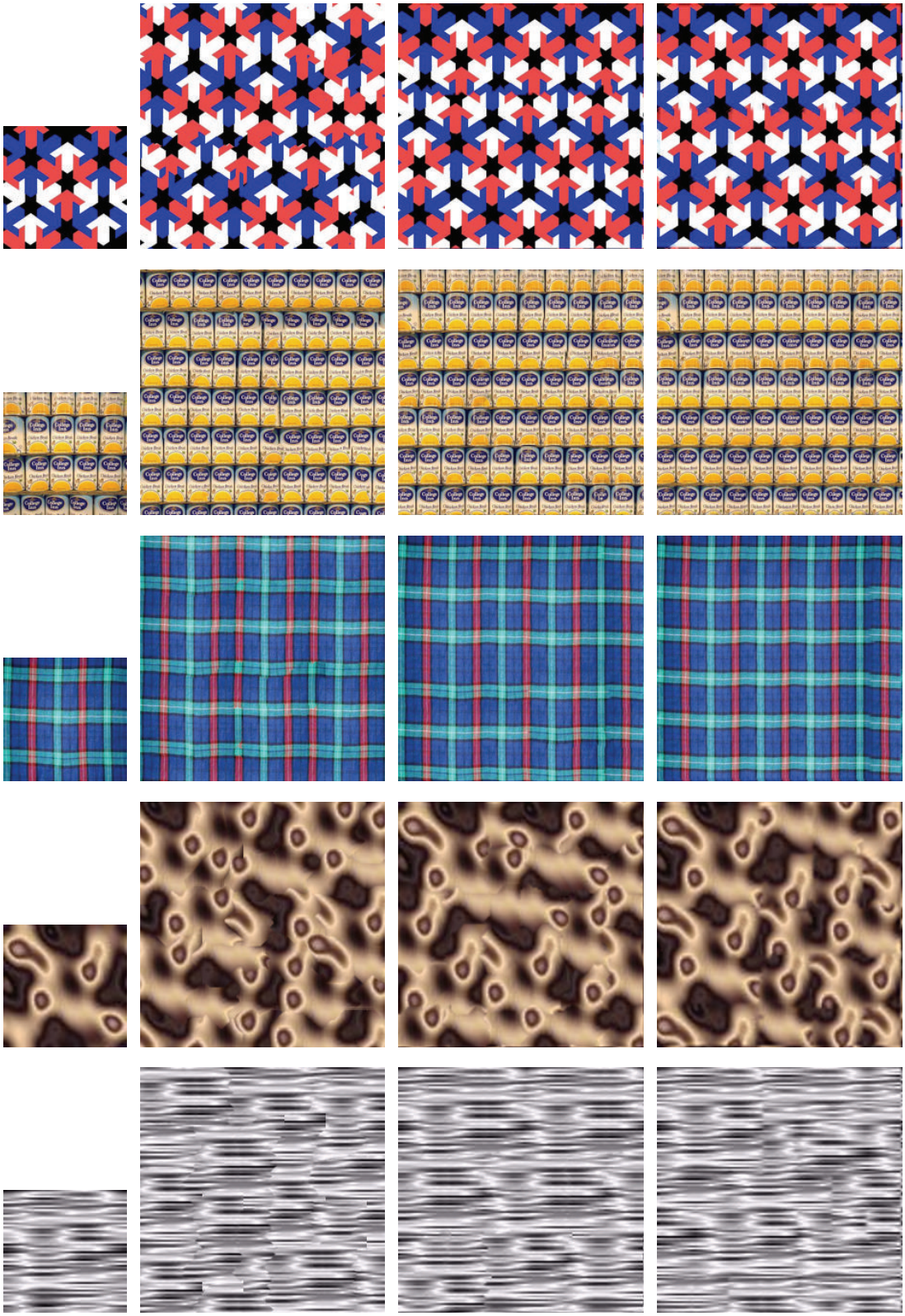

Figure 5: Synthesis results of the input images (1st column) using Efros's algorithm [11] (2nd column), Cui's algorithm [33] (3rd column), and the TSIAWPT algorithm (4th column).

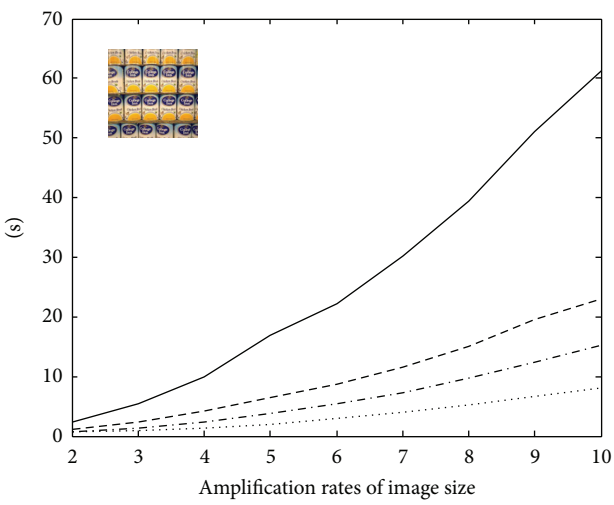

Figure 6: Computation times of Efros's algorithm [11] (solid line), Cui's algorithm [33] (dashed line), TSIWPT [35] (dashdot line), and TSIAWPT (dotted line).

4. Experimental Results

In our experiments, the commonly used 9/7-wavelet filters adopted in the JPEG2000 standard are used to decompose the input image into adaptive wavelet packets; the size of the lowest-frequency subband is 32 × 32; in other words, the number of decomposition levels is 3 for 256 × 256 images; rectangular patches with the ACF-based patch sizes are used; the width of overlapping borders is set to one-sixth of the patch size; the k-means algorithm is used to cluster the input AWPT into similarity sets.

In the first experiment, we demonstrate that the color distortion caused by using TSIWPT can be avoided by using TSIAWPT, as shown in Figures 2(b) and 2(c), respectively. More synthesis results using TSIAWPT for various types of textures, namely architectures, regular structures, leaves, raw materials, and man-made materials, are given in Figure 4.

The second experiment is to compare the TSIAWPT algorithm with two other well-known algorithms, Efros's algorithm [11] and Cui's algorithm [33]. The size of synthesis images is twice the size of the input images. Figure 5 shows the synthesis results. It is noted that TSIAWPT outperforms Efros's algorithm and is marginally preferable to the Cui's algorithm. Moreover, there are some blocking defects in the fourth and fifth synthesis images using Efros's algorithm. As the last step of TSIAWPT is inverse adaptive wavelet packet transform, which essentially involves low pass filtering, there is almost no blocking defect using TSIAWPT.

The last experiment is to evaluate the running speed. Figure 6 shows the computation times required to synthesize images with different sizes using Efros's algorithm, the Cui's algorithm, TSIWPT, and TSIAWPT. All of the above were simulated on a PC equipped with a CPU of 1.73 GHz and 4 GB of RAM. It is noted that TSIAWPT is preferable to the others in terms of computation time, especially for synthesizing large textures.

5. Conclusion

Adaptive wavelet packet transform provides a more compact representation for textured images, and moreover they can be efficiently represented in adaptive wavelet packet trees with the same hierarchical structure as the conventional wavelet trees. An efficient algorithm, TSIAWPT, has been proposed to synthesize textures in adaptive wavelet packet trees. It has the advantage of reducing computation time substantially and there is no training process involved. Specifically, the average time required to synthesize an 256 × 256 image from an 128 × 128 input texture is in a fraction of a second.

Acknowledgment

The National Science Council of Taiwan, under Grants NSC100-2628-E-239-002-MY2, and NSC100-2410-H-216-003 supported this work.

References

[1] S.-P. Du, S.-M. Hu, and R. R. Martin, "Semi-regular solid texturing from 2D image exemplars," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 460–469, 2012.

[2] M. Petrou, M. H. Jaward, S. Chen, and M. Briers, "Super-resolution in practice: the complete pipeline from image capture to super-resolved subimage creation using a novel frame selection method," *Machine Vision and Applications*, vol. 23, no. 3, pp. 441–459, 2011.

[3] J. Zhang, S. Y. Chen, S. Liu, and Q. Guan, "Normalized weighted shape context and its application in feature-based matching," *Optical Engineering*, vol. 47, no. 9, Article ID 097201, 2008.

[4] M. A. Akhloufi, X. Maldague, and W. B. Larbi, "A new color-texture approach for industrial products inspection," *Journal of Multimedia*, vol. 3, no. 3, pp. 44–51, 2008.

[5] S. Lefebvre and H. Hoppe, "Appearance-space texture synthesis," in *Proceedings of the ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH '06)*, pp. 541–548, Boston, Mass, USA, August 2006.

[6] L. Ritter, W. Li, B. Curless, M. Agrawala, and D. Salesin, "Painting with texture," in *Proceedings of the 17th Eurographics Symposium on Rendering*, pp. 371–376, Nicosa, Cyprus, 2006.

[7] V. Kwatra, D. Adalsteinsson, T. Kim, N. Kwatra, M. Carlson, and M. C. Lin, "Texturing fluids," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 5, pp. 939–952, 2007.

[8] L. Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the art in example-based texture synthesis," in *Proceedings of the Eurographics State of the Art Reports (STARs)*, Eurographics, March 2009.

[9] 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.

[10] Y. Xu, B. Guo, and H. Y. Shum, "Chaos mosaic: fast and memory efficient texture synthesis," Tech. Rep. MSR-TR-2000-32, Microsoft Research, 2000.

[11] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 341–346, August 2001.

[12] M. F. Cohen, J. Shade, S. Hiller, and O. Deussen, "Wang tiles for image and texture generation," in *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '03)*, pp. 287–294, San Diego, Calif, USA, July 2003.

[13] G. Cheng and J. Dong, "Seamless montage of natural texture," in *3rd International Conference on Advanced Computer Control (ICACC '11)*, pp. 48–51, Harbin, China, 2011.

[14] S. Y. Chen, H. Tong, and C. Cattani, "Markov models for image labeling," *Mathematical Problems in Engineering*, vol. 2012, Article ID 814356, 18 pages, 2012.

[15] Q. Guan, B. Du, Z. Teng, J. Gillard, and S. Chen, "Bayes clustering and structural support vector machines for segmentation of carotid artery plaques in multi-contrast MRI," *Computational and Mathematical Methods in Medicine*, vol. 2012, Article ID 549102, 6 pages, 2012.

[16] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the 7th IEEE International Conference on Computer Vision, (ICCV '99)*, vol. 2, pp. 1033–1038, Kerkyra, Greece.

[17] L. Y. Wei and M. Levoy, "Order independent texture synthesis," Tech. Rep. TR-2002-01, Stanford Computer Science, 2002.

[18] 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.

[19] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," in *Proceedings of the ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH '05)*, pp. 795–802, Los Angeles, Calif, USA, August 2005.

[20] J. Han, K. Zhou, L. Y. Wei et al., "Fast example-based surface texture synthesis via discrete optimization," *Visual Computer*, vol. 22, no. 9–11, pp. 918–925, 2006.

[21] F. Dong and X. Ye, "Multiscaled texture synthesis using multisized pixel neighborhoods," *IEEE Computer Graphics and Applications*, vol. 27, no. 3, pp. 41–47, 2007.

[22] 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.

[23] 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.

[24] J. S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Proceedings of the ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH '06)*, pp. 361–368, Los Angeles, Calif, USA, August 1997.

[25] P. J. Burt, "Fast algorithms for estimating local image properties," *Computer Vision, Graphics, & Image Processing*, vol. 21, no. 3, pp. 368–382, 1983.

[26] L. Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proceedings of the ACM Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH '00)*, pp. 479–488, New Orleans, La, USA, July 2000.

[27] J. C. Goswami and A. K. Chan, *Fundamentals of Wavelets: Theory, Algorithms, and Applications*, Wiley Series in Microwave and Optical Engineering, John Wiley & Sons, Hoboken, NJ, USA, 2nd edition, 2011.

[28] S. C. Lim, C. H. Eab, K. H. Mak, M. Li, and S. Y. Chen, "Solving linear coupled fractional differential equations by direct operational method and some applications," *Mathematical Problems in Engineering*, vol. 2012, Article ID 653939, 28 pages, 2012.

[29] C. Cattani, "Harmonic wavelet approximation of random, fractal and high frequency signals," *Telecommunication Systems*, vol. 43, no. 3-4, pp. 207–217, 2010.

[30] C. Cattani, "Shannon wavelets for the solution of integrodifferential equations," *Mathematical Problems in Engineering*, vol. 2010, Article ID 408418, 22 pages, 2010.

[31] A. Abbate, C. M. DeCusatis, and P. K. Das, *Wavelets and Subbands: Fundamentals and Applications*, Applied and Numerical Harmonic Analysis, Springer, Boston, Mass, USA, 2002.

[32] Y. Yu, J. Luo, and C. W. Chen, "Multiresolution block sampling based method for texture synthesis," in *Proceedings of the 16th International Conference on Pattern Recognition*, pp. 239–242, 2002.

[33] H. F. Cui, X. Zheng, and T. Ruan, "An efficient texture synthesis algorithm based on WT," in *Proceedings of the 7th International Conference on Machine Learning and Cybernetics (ICMLC '08)*, pp. 3472–3477, Kunming, China, July 2008.

[34] 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.

[35] H. C. Hsin, T.-Y. Sung, Y.-S. Shieh, and C. Cattani, "A new texture synthesis algorithm based on wavelet packet tree," *Mathematical Problems in Engineering*, vol. 2012, Article ID 305384, 12 pages, 2012.

[36] H. C. Hsin, T. Y. Sung, and L. T. Ko, "A fast Wavelet packet based algorithm for texture synthesis," in *21st International Conference on Pattern Recognition (ICPR '12)*, pp. 3124–3127, Proceedings of the Tsukuba, Japan, 2012.

[37] 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.

[38] 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.