On the Number of Rectangular Tilings

Dan Xu, *Student Member, IEEE,* and Minh N. Do, *Member, IEEE*

***Abstract*— Adaptive multiscale representations via quadtree splitting and 2D wavelet packets, which amount to space and frequency decompositions, respectively, are powerful concepts that have been widely used in applications. These schemes are direct extensions of their 1D counterparts, in particular, by coupling of the two dimensions and restricting to only one possible further partition of each block into four sub-blocks. In this paper, we consider more flexible schemes that exploit more variations of multidimensional data structure. In the meantime, we restrict to tree-based decompositions that are amenable to fast algorithms and have low indexing cost. Examples of these decomposition schemes are anisotropic wavelet packets, dyadic rectangular tilings, separate dimension decompositions, and general rectangular tilings. We compute the numbers of possible decompositions for each of these schemes. We also give bounds for some of these numbers. These results show that the new rectangular tiling schemes lead to much larger sets of 2D space and frequency decompositions than the commonly-used quadtree-based schemes, therefore bearing the potential to obtain better representation for a given image.**

***Index Terms*— Number of bases, wavelet packets, quadtree decompositions, rectangular tilings, multiscale representations, anisotropic bases, best basis.**

I. Introduction

Adaptive signal decompositions have been widely used in many applications. In these schemes, the "best" decomposition is selected adaptively among a class of decompositions for a given input signal. In practice, it is desirable that the class of decompositions has the following two properties: 1) all the candidate decompositions can be easily indexed so that an efficient best-decomposition search algorithm can be available, and 2) the size of the class needs to be large enough. The tree-structured decompositions usually have the first property because of the efficient indexing of the tree structure; the size of the classes, however, varies for different tree structures, which needs to be carefully calculated.

Two classes of tree-structured decompositions in two-dimensions have been extensively investigated. One class is 2D wavelet packets [1]-[3] and their dual in the spatial domain — *quadtree decompositions* (QD) [4]. The 2D wavelet packets couple the row and column decompositions, dividing the frequency domain into four equally-sized subbands, where each subband can be successively decomposed. Since the basis functions generated by 2D wavelet packets have the same scale along row and column dimensions, we refer to such wavelet packets as *isotropic wavelet packets* (IWP). QD is a very popular scheme used in image compression [4]-[7] and other applications [8], [9] because of its efficient structure. QD has the same tree structure as IWP. One example of QD is in Figure 1(a).

The other class is *anisotropic wavelet packets* (AWP) [10]-[13] and their dual in the spatial domain — *dyadic rectangular tilings* (DRT) [14]. AWP relaxes the coupling of the row and column decomposition steps in IWP by allowing an arbitrary order of row and column decompositions, therefore producing much more varieties of basis functions. DRT has the same tree structure as AWP. One example of DRT is shown in Figure 1(b), which shows a reduced cost over QD.

Dan Xu and Minh N. Do are with the Department of Electrical and Computer Engineering, the Beckman Institute, and the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801 (emails: danxu@uiuc.edu, minhdo@uiuc.edu).

This work was supported in part by National Science Foundation CAREER award CCR-0237633.

In natural images, separations of two homogeneous areas are typically at non-dyadic locations. In order to separate such areas, the DRT scheme has to divide the original image by several levels to approximately reach the location of the separator, where the first few decompositions are redundant. This prompts us to consider rectangular tilings where the location and size of each rectangle can be arbitrary. This feature helps capture the border of an object accurately and avoid unnecessary separators, as illustrated in Figure 1(c). We refer to this kind of rectangular tilings as *general rectangular tilings* (GRT).

In certain applications, it is desirable to achieve optimal decompositions for two dimensions that are separate in the sense that the decomposition in one dimension is done before the other. We refer to this class of decomposition schemes as *separate dimension decompositions* (SDD). A class of SDD, sorted dyadic SDD, was studied in [15]-[17], where time-varying wavelet packets were considered. The joint space-frequency segmentation problem was studied independently in [18]-[20], where either space or frequency domain can be decomposed first. This class of schemes is the unsorted dyadic SDD, as will be discussed In Section IV.

The main focus of this paper is to study the numbers of decompositions by all the schemes mentioned above. In particular, we were motivated to study how the various extensions of IWP and QD enlarge the set of possible decompositions or bases. It is important to notice that thanks to the tree-based structures, all of the schemes considered in this paper have fast algorithm to search for the best decomposition for given images. Examples are tree pruning-based algorithms for IWP/QD [1], [2], the fast algorithm in [12] for AWP/DRT, the double-tree algorithm [15], [16] for sorted dyadic SDD, and the fast algorithms in [18]-[20] for unsorted dyadic SDD. More notably, recent work in [21] developed a general framework and fast algorithms to search the best representations in all of these tree-structured decompositions.

The rest of the paper is organized as follows. We first review some key results in IWP/QD and AWP/DRT in Section II. In Section III, we extend to GRT where a splitting separator is not subject to the dyadic-location constraint. In Section IV, we introduce a special family of tree-based decompositions, SDD, in both ordered and unordered, dyadic and general versions. We obtain formulae to calculate numbers of possible decompositions for all the above schemes and also provide bounds of some of these numbers. In Section V, we summarize numbers of decompositions for various schemes and study the cost to index the decomposition structure. Conclusion remarks are given in Section VI.

II. Isotropic and Anisotropic Wavelet Packets

A. Isotropic Wavelet Packets and Quadtree Decompositions

IWP [1]-[3] and its spatial-domain counterpart QD [4] share the same tree structure: an admissible quadtree, which is a quadtree with all its nodes having zero or four children. Each node of the quadtree represents a square in the 2D frequency or space plane, and each tree-splitting corresponds to a decomposition of a parent square into four equally-sized child squares (e.g., Figure 1(a)).

We denote Q_J as the number of decompositions for IWP/QD of up to level J, which equals to the number of admissible quadtrees of

(a) (b) (c)

Fig. 1. Spatial-domain decomposition of the "Cameraman" image (256×256) with maximum level being five and cost function being the entropy of the energy distribution [3]. (a) Quadtree decompositions. Cost= 1.59×10^3. (b) Dyadic rectangular tilings. Cost= 1.55×10^3. (c) General rectangular tilings. Cost= 1.31×10^3.

up to depth J. It can be seen by the quadtree structure that [3], [17]

$$Q_J = Q_{J-1}^4 + 1 \tag{1}$$

for $J \geq 1$ and $Q_0 = 1$. The bounds of Q_J were studied in [3]; but the upper bound for Q_J given in Prop. 8.5 in [3], $Q_J \leq 2^{\frac{49}{48}4^{J-1}}$, is not correct. For example, using (1), $Q_4 = 4.87 \times 10^{19} > 2^{\frac{49}{48}4^{4-1}} = 4.65 \times 10^{19}$. The bounds of 1D wavelet packets were studied in [17], based on a method to convert the recursive form to nonrecursive form for doubly exponential sequences [22]; but the bounds for 2D were not explicitly given. In the following proposition, we provide the corrected bounds of Q_J.

Proposition 1: For $J \geq 6$, Q_J satisfies

$$Q_J = C^{4^J}, \tag{2}$$

where

$$1.19372183770247 = C_{\text{lower}} \leq C \leq C_{\text{upper}} = 1.19372183770248.$$

Proof: The lower bound is readily obtained by $Q_J \geq Q_{J-1}^4 \geq \cdots \geq Q_5^{4^{J-5}} \geq (C_{\text{lower}})^{4^J}$ for $J \geq 5$, where Q_5 is calculated using (1). For the upper bound, because Q_J increases as J increases, we have $Q_J = (1 + Q_{J-1}^{-4})Q_{J-1}^4 \leq (1 + Q_5^{-4})Q_{J-1}^4$ for $J \geq 6$, which gives $Q_J \leq (1 + Q_5^{-4})^{-\frac{1}{3}}[Q_5(1 + Q_5^{-4})^{\frac{1}{3}}]^{4^{J-5}} \leq (C_{\text{upper}})^{4^J}$. ■

B. Anisotropic Wavelet Packets and Dyadic Rectangular Tilings

AWP [10]-[13] and its spatial-domain counterpart DRT [14] relax the coupling of row and column decompositions, therefore generating a larger family of bases than IWP/QD. The whole decompositions are characterized by a binary tree, where a "row" or "column" value is assigned to each bifurcation. Each node of the binary tree represents a rectangle in the 2D frequency or space plane, and each bifurcation represents a row-wise or column-wise decomposition of a parent rectangle into two equally-sized child rectangles, depending on whether a "row" or "column" value is assigned (e.g., Figure 1(b)).

Denote $A_{I,J}$ as the number of decompositions for AWP/DRT of up to row level I and column level J. The number $A_{I,J}$ can be calculated recursively [12]:

$$A_{I,J} = 1 + A_{I-1,J}^2 + A_{I,J-1}^2 - A_{I-1,J-1}^4, \tag{3}$$

and $A_{I,0} = 1 + A_{I-1,0}^2$, $A_{0,J} = 1 + A_{0,J-1}^2$, for $I \geq 1$ and $J \geq 1$, and $A_{0,0} = 1$. The symmetry also holds: $A_{I,J} = A_{J,I}$. In the following proposition, we give bounds of $A_{J,J}$:

Proposition 2: For $J \geq 6$, $A_{J,J}$ satisfies

$$1.738^{4^J} \leq A_{J,J} \leq \frac{1.921^{4^J}}{2}. \tag{4}$$

Proof: One way of dividing a $2^J \times 2^J$ image using AWP/DRT is to divide the image into four equal squares, each of which has $A_{J-1,J-1}$ different decompositions. Therefore, the lower bound is obtained by the following:

$$A_{J,J} \geq A_{J-1,J-1}^4 \geq \cdots \geq A_{5,5}^{4^{J-5}} \geq 1.738^{4^J}. \tag{5}$$

To get the upper bound, we define another sequence $F_{I,J}$: $F_{I,J} = 1 + F_{I-1,J}^2 + F_{I,J-1}^2$, $F_{I,0} = 1 + F_{I-1,0}^2$, $F_{0,J} = 1 + F_{0,J-1}^2$, for $I \geq 1, J \geq 1$, and $F_{0,0} = 1$. It is obvious that $A_{I,J} \leq F_{I,J}$ for any I and J. As proved in Appendix A, $F_{J-2,J} \leq F_{J-1,J-1}$. Therefore, for $F_{J,J}$, we have

$$\begin{aligned} F_{J,J} &= 1 + 2F_{J-1,J}^2 = 1 + 2(1 + F_{J-2,J}^2 + F_{J-1,J-1}^2)^2 \\ &\leq 2(1 + 2F_{J-1,J-1}^2)^2 = 8(1 + \tfrac{1}{2F_{J-1,J-1}^2})^2 F_{J-1,J-1}^4 \\ &\leq 8(1 + \tfrac{1}{2F_{5,5}^2})^2 F_{J-1,J-1}^4 \end{aligned} \tag{6}$$

for $J \geq 6$, which yields $A_{J,J} \leq F_{J,J} \leq \frac{1.921^{4^J}}{2}$. ■

III. General Rectangular Tilings

Both IWP/QD and AWP/DRT decompose the 2D frequency or space plane at dyadic locations. In many cases, however, the borders of the homogeneous regions in an image are not at dyadic locations. It is therefore desirable to relax the dyadic restriction by allowing the separators of two regions to be at arbitrary location. We refer to this class of decompositions the *general rectangular tilings*. As shown in Figure 1(c), GRT have more flexibility to put the separators and therefore partition the original image more accurately and concisely.

Denote $R(m, n)$ as the number of different decompositions of an $m \times n$ image using GRT. The following proposition provides a recursive formula to calculate $R(m, n)$:

Proposition 3: The number $R(m, n)$ satisfies

$$\begin{aligned} R(m,n) = \; & 1 + \sum_{k=0}^{m-1} \sum_{\substack{l=0 \\ k+l \neq 0}}^{n-1} (-1)^{k+l+1} \\ & \sum_{\substack{0=i_0<i_1<\cdots<i_{k+1}=m \\ 0=j_0<j_1<\cdots<j_{l+1}=n}} \prod_{s=1}^{k+1} \prod_{t=1}^{l+1} R(i_s - i_{s-1}, j_t - j_{t-1}). \end{aligned} \tag{7}$$

Proof: Consider dividing an $m \times n$ image. Both row and column separators can happen at any integer locations. We define $A_i \triangleq \{\text{the } i\text{th row separator is on}\}$, and $B_j \triangleq$

$\{$the jth column separator is on$\}$, where $i = 1, \cdots, m-1$, $j = 1, \cdots, n-1$. We also denote $\|X\|$ as the number of different partitions from the pattern specified by the separator states X. Using $\|A \cup B\| = \|A\| + \|B\| - \|A \cap B\|$ repeatedly, we have

$$\begin{aligned} R(m,n) &= \|\text{no separator is on}\| + \|\text{at least one separator is on}\| \\ &= 1 + \|(\cup_{i=1}^{m-1} A_i) \cup (\cup_{j=1}^{n-1} B_j)\| \\ &= 1 + \sum_{k=0}^{m-1} \sum_{\substack{l=0 \\ k+l\neq 0}}^{n-1} (-1)^{k+l+1} \\ &\qquad \sum_{\substack{1\le i_1<\cdots<i_k\le m-1 \\ 1\le j_1<\cdots<j_l\le n-1}} \|A_{i_1} \cap \cdots \cap A_{i_k} \cap B_{j_1} \cap \cdots \cap B_{j_l}\|, \end{aligned} \tag{8}$$

where the term in the last "$\sum$" is the number of partitions when row separators $i_1, \cdots, i_k$, and column separators $j_1, \cdots, j_l$, are turned on, which is $\prod_{s=1}^{k+1} \prod_{t=1}^{l+1} R(i_s - i_{s-1}, j_t - j_{t-1})$ when defining $i_0 = j_0 = 0$, $i_{k+1} = m$, and $j_{l+1} = n$. ■

The bounds of $R(m,n)$ are given in the following proposition, where the most commonly-used case $m = n = 2^J$ is considered and the notation $R_{J,J} \triangleq R(2^J, 2^J)$ is adopted.

Proposition 4: For $J \geq 3$, $R_{J,J}$ satisfies

$$2.014^{4^J} \leq R_{J,J} \leq 4^{4^J - 2^J}. \tag{9}$$

Proof: The first few values of $R_{m,n}$ are: $R(2,2) = 8$, $R(2,6) = R(6,2) = 2864$ and $R(6,6) = 446048380296$. For an 8×8 image, one class of partitions is through dividing it into four rectangles, with their numbers of partitions being $R(6,6)$, $R(2,2)$, $R(2,6)$, and $R(6,2)$, respectively. Therefore,

$$R_{3,3} = R(8,8) \geq 1 + R(6,6)R(2,2)R(2,6)R(6,2) \geq 2.926 \times 10^{19}. \tag{10}$$

For $J \geq 3$, the lower bound is then $R_{J,J} \geq R_{J-1,J-1}^4 \geq \cdots \geq R_{3,3}^{4^{J-3}} \geq 2.014^{4^J}$.

On the other hand, an $n \times n$ image has up to $2n(n-1)$ unit separators at integer locations. Each unit separator can be "on" or "off", which gives the upper bound $R_{J,J} \leq 2^{2 \cdot 2^J (2^J - 1)} = 4^{4^J - 2^J}$. ■

IV. Separate Dimension Decompositions

In some applications, e.g., tilings of the time-frequency plane [15], we deal with a 2D decomposition problem where the decompositions of one dimension are done before the other dimension. We refer to this class of decompositions as *separate dimension decompositions*. For SDD, when decompositions are allowed only at the dyadic locations, we call them dyadic SDD. When decompositions are allowed at an arbitrary location, we call them general SDD.

A. Dyadic Separate Dimension Decompositions

Depending on applications, dyadic SDD have ordered and unordered versions, which are defined below.

Definition 1: The *ordered dyadic SDD* (OD-SDD) are dyadic SDD where one of the dimensions must be decomposed before the other dimension, while the *unordered dyadic SDD* (UD-SDD) are dyadic SDD where either dimension can be decomposed first.

One example of OD-SDD is the time-varing wavelet packets [15]-[17], and one example of UD-SDD is the joint space-frequency segmentation [18]-[20], where either space or frequency domain can be decomposed first. Figure 2 shows some examples of OD-SDD and UD-SDD.

We denote $T_{I,J}$ as the number of decompositions for OD-SDD on a $2^I \times 2^J$ image, where the dimension corresponding to I is assumed to be decomposed first, and $S_{I,J}$ as the number for UD-SDD. The recursive formulae to calculate $T_{I,J}$ and $S_{I,J}$ are given by the following two propositions:

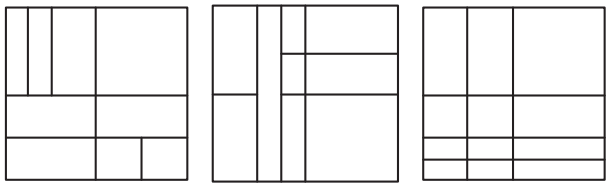

Fig. 2. Examples of dyadic SDD. (a) and (b) are instances where the vertical and horizontal dimensions, respectively, are decomposed first. (c) is a special instance that can result either from decomposition along vertical dimension followed by decomposition along horizontal dimension, or vise versa.

Proposition 5: The number $T_{I,J}$ is

$$T_{I,J} = \sum_{k=0}^{B_I - 1} B_J^{M_I(k)}, \tag{11}$$

for $I, J \geq 0$, where B_J is the number of decompositions of 1D binary tree of up to depth J [1], [3]: $B_J = B_{J-1}^2 + 1$ for $J \geq 1$, $B_0 = 1$, and $M_I(k)$ is the number of segments generated in the kth 1D binary-tree decomposition of up to I levels ($k = 0, 1, \cdots, B_I - 1$) that is defined recursively by $M_I(k) = M_{I-1}(m) + M_{I-1}(n)$, $M_I(0) = 1$, with $m = \lfloor \frac{k-1}{B_{I-1}} \rfloor$ and $n = (k-1) \bmod B_{I-1}$ ($\lfloor x \rfloor$ is the largest integer smaller than or equal to x and "mod" is the modulus operation).

Proof: We first establish a recursive formula to calculate $M_I(k)$. A 1D binary tree of up to depth I has B_I different decompositions (tree structures). We index the decomposition in the following recursive way. For depth zero, $k = 0$ corresponds to the root node. Suppose we have finished indexing for decompositions with depth of up to $I-1$. For decompositions with depth of up to I, we use $k = 0$ as the case where no decomposition happens (root node); for other cases where at least one decomposition happens, as shown in Figure 3, we find the index m for the left subtree, index n for the right subtree (m and n can be determined since both subtrees have depth up to $I-1$), and define $k = mB_{I-1} + n + 1$ for $m, n = 0, 1, \cdots, B_{I-1} - 1$. The above indexing yields $M_I(k) = M_{I-1}(m) + M_{I-1}(n)$, $M_I(0) = 1$, where m and n are readily obtained from the definition of k: $m = \lfloor \frac{k-1}{B_{I-1}} \rfloor$ and $n = (k-1) \bmod B_{I-1}$.

Now we prove (11). The first dimension (corresponding to I) has B_I different decompositions. The kth decomposition has $M_I(k)$ segments, each of which can be further decomposed along the second dimension (B_J different varieties). Therefore, for the kth decomposition along the first dimension, we have $B_J^{M_I(k)}$ different decompositions in 2D. Summing over k yields (11). ■

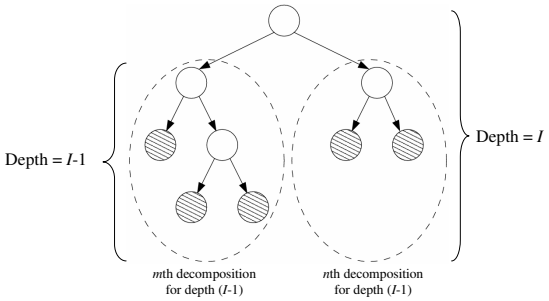

Fig. 3. The number of segments for a 1D dyadic decomposition. Each shaded circle represents a leaf node, which corresponds to a segment after a decomposition.

Proposition 6: For $I, J \geq 0$, $S_{I,J}$ is

$$S_{I,J} = T_{I,J} + T_{J,I} - B_I B_J. \tag{12}$$

Proof: For UD-SDD, decomposition either happens first along the vertical axis, (Figure 2(a)), which has $T_{I,J}$ decompositions, or first along the horizontal axis (Figure 2(b)), which has $T_{J,I}$ decompositions. These two types of decompositions share an overlapped subset as shown in Figure 2(c), where the dyadic separators on both dimensions hit both sides of the boundary. The number of decompositions corresponding to this overlapped subset is $B_I B_J$. Equation (12) follows from the above analysis. ■

B. General Separate Dimension Decompositions

For general SDD, decomposition along either dimension can happen at arbitrary integer locations. This brings about more flexibility and accuracy in many applications than the dyadic SDD. General SDD also have ordered and unordered versions, which are defined below.

Definition 2: The *ordered general SDD* (OG-SDD) are general SDD where one of the dimensions must be decomposed before the other dimension, while the *unordered general SDD* (UG-SDD) are general SDD where either dimension can be decomposed first.

We denote $G_{m,n}$ as the number of decompositions for OG-SDD on an $m \times n$ image, where the dimension corresponding to m is assumed to be decomposed first, and $U_{m,n}$ as the number for UG-SDD. The recursive formulae to calculate $G_{m,n}$ and $U_{m,n}$ are given by the following two propositions.

Proposition 7: For $m, n \geq 1$, $G(m,n)$ is

$$G(m,n) = 2^{n-1}(2^{n-1}+1)^{m-1}. \tag{13}$$

Proof: Let us first consider the 1D general decomposition of a length-n line. Since every one of the $n-1$ separators can be turned "on" or "off", the total number of decompositions is 2^{n-1}. The number of segments after the decomposition can range from 1 to n. Furthermore, there are $\binom{n-1}{k-1}$ decompositions that generate k segments.

From the above analysis, there are $\binom{m-1}{k-1}$ ways to decompose the first dimension into k segments, which give rise to $(2^{n-1})^k$ decompositions in 2D plane, as each segment has 2^{n-1} different decompositions along the second dimension. Therefore, $G(m,n) = \sum_{k=1}^{m} \binom{m-1}{k-1}(2^{n-1})^k = 2^{n-1}(2^{n-1}+1)^{m-1}$. ■

Proposition 8: For $m, n \geq 1$, $U(m,n)$ is

$$U(m,n) = 2^{n-1}(2^{n-1}+1)^{m-1} + 2^{m-1}(2^{m-1}+1)^{n-1} - 2^{m+n-2}. \tag{14}$$

Proof: Similar to the proof of Proposition 6, for UG-SDD, decomposition either happens first along the vertical axis, which has $G_{m,n}$ decompositions, or first along the horizontal axis, which has $G_{n,m}$ decompositions. These two types of decompositions also share an overlapped subset, which has $2^{m-1} \cdot 2^{n-1} = 2^{m+n-2}$ decompositions. Equation (14) follows from the above analysis. ■

V. Summary of Numbers of Decompositions and Index Cost

A. Summary of Numbers of Decompositions

A comparison of numbers of decompositions by various schemes is in Table I,[1] showing that roughly

$$\begin{aligned} R(2^J, 2^J) &\gg U(2^J, 2^J) \approx 2G(2^J, 2^J) > A_{J,J} \\ &\gg S_{J,J} \approx 2T_{J,J} \gg Q_J. \end{aligned} \tag{15}$$

[1]To compute a larger J for $R(2^J, 2^J)$, a lot of iterations are needed and the process is time-consuming. Therefore we only calculate the actual numbers of $R(2^J, 2^J)$ for up to $J = 2$ and list the lower bounds according to (9) for $J = 3, 4, 5$.

B. Index Cost

For each of the aforementioned subspace decomposition frameworks, extra bits have to be spent to index the decomposition structure. It is important to know these costs before applying the decomposition frameworks, since a high cost of indexing can affect the applicability of the schemes.

For IWP/QD, the quadtree of up to J levels has a total of $1 + 4 + \cdots + 4^J = (4^{J+1}-1)/3$ nodes, each of which needs one bit to record the information of "decomposed" or "null", which results in $(4^{J+1}-1)/3$ extra bits for the cost of index. For AWP/DRT, the binary tree of up to J levels has $1 + 2 + \cdots + 2^{2J} = 2^{2J+1} - 1$ nodes, each of which needs $\log_2 3$ bits to record the information of "row", "column" or "null", which results in $(2^{2J+1}-1)\log_2 3$ extra bits for the cost of index.

To index OD-SDD, we use a complete binary tree of $I+J$ levels, with the upper I levels corresponding to the first dimension and the lower J levels to the second dimension. The whole tree has $2^{I+J+1}-1$ nodes where each node needs one bit to record the information of "decomposed" or "null", therefore giving rise to $2^{I+J+1}-1$ extra bis for the cost of index. Indexing UD-SDD is the same as indexing OD-SDD, except that one extra bit is used to specify whether horizontal or vertical direction is the first dimension, which causes 2^{I+J+1} bits for the cost of index.

To index OG-SDD, we record the "on" or "off" state of each separator. For the first dimension, we only need $m-1$ bits, since all the separators in the first dimension hit both sides of the boundary of the 2D plane; for the second dimension, we need $m(n-1)$ bits for all the unit separators. Therefore we need a total of $mn-1$ bits for the cost of index. For UG-SDD, we adopt the same indexing as OG-SDD does, except that we use one more bit to specify whether horizontal or vertical direction is the first dimension, which causes mn bits for the cost of index.

To index GRT, we record the states of all the unit separators in both dimensions, which yields $m(n-1)+n(m-1) = 2mn-m-n$ bits for the cost of index.

The cost of indexing different schemes is summarized in Table II. Note that the cost for dyadic schemes is relatively small for applications. For example, indexing AWP/DRT of up to 4 row and column levels needs about 810 bits, which is negligible in most practical applications.

VI. Conclusion

We studied several space and frequency-domain rectangular tiling schemes in this paper to generalize the commonly-used quadtree-based scheme. We computed the numbers of different decompositions for each of these schemes and gave bounds for some of these numbers. These results show that the new schemes lead to much larger sets of 2D space and frequency decompositions than the quadtree-based schemes, therefore bearing the potential to obtain more compact representation of given images. We also discussed the issue of indexing various decomposition structures and calculated the costs of index for all the schemes. These costs are small for most applications.

Appendix

A. Proof of $F_{J-2,J} \leq F_{J-1,J-1}$

In the following, we prove a stronger statement:

$$F_{I-(n+1),J+(n+1)} \leq F_{I-n,J+n} \tag{16}$$

for any $1 \leq I \leq J$, and any $n = 0, 1, \cdots, I-1$.

We prove (16) by induction. For $I = J = 1$, $n = 0$, we have $F_{0,2} = 1 + F_{0,1}^2 = 5 < 9 = F_{1,1}$. Suppose (16) holds for $1 \leq I \leq$

Scheme	$J=1$	$J=2$	$J=3$	$J=4$	$J=5$
IWP/QD (Q_J)	2	17	8.35×10^{4}	4.87×10^{19}	5.61×10^{78}
AWP/DRT ($A_{J,J}$)	8	6857	2.32×10^{15}	2.90×10^{61}	7.00×10^{245}
OD-SDD ($T_{J,J}$)	6	905	4.86×10^{11}	3.94×10^{45}	2.88×10^{181}
UD-SDD ($S_{J,J}$)	8	1785	2.43×10^{11}	1.97×10^{45}	1.44×10^{181}
OG-SDD ($G(2^J,2^J)$)	6	5832	7.61×10^{16}	1.77×10^{72}	4.19×10^{298}
UG-SDD ($U(2^J,2^J)$)	8	11600	1.52×10^{17}	3.54×10^{72}	8.37×10^{298}
GRT ($R(2^J,2^J)$)	8	68480	$>2.88\times10^{19}$	$>6.91\times10^{77}$	$>2.27\times10^{311}$

TABLE I

COMPARISON OF NUMBERS OF DECOMPOSITIONS BY VARIOUS SCHEMES IN FREQUENCY AND SPATIAL DOMAIN.

Scheme	Number of bits	Scheme	Number of bits
IWP/QD (Q_J)	$(4^{J+1}-1)/3$	OG-SDD ($G(m,n)$)	$mn-1$
AWP/DRT ($A_{J,J}$)	$(2^{2J+1}-1)\log_2 3$	UG-SDD ($U(m,n)$)	mn
OD-SDD ($T_{I,J}$)	$2^{I+J+1}-1$	GRT ($R(m,n)$)	$2mn-m-n$
UD-SDD ($S_{I,J}$)	2^{I+J+1}		

TABLE II

COST OF INDEX FOR VARIOUS DECOMPOSITION SCHEMES.

$J=K-1$, $K\geq 2$, $n\in[0,I-1]$, we need to prove that (16) holds for $1\leq I\leq J=K$, $0\leq n\leq I-1$. Under the induction assumption, we get

$$F_{(K-1)-(n+1),(K-1)+(n+1)}\leq F_{(K-1)-(n-1),(K-1)+(n-1)}. \quad (17)$$

for $0\leq n\leq I-1$ (inequality (17) holds for $n=0$ because $F_{K-2,K}=F_{K,K-2}$). Therefore,

$$\begin{aligned} F_{K-(n+1),(K-1)+(n+1)} &= 1+F^2_{(K-1)-(n+1),(K-1)+(n+1)} \\ &\quad +F^2_{K-(n+1),(K-1)+n} \\ &\leq 1+F^2_{(K-1)-(n-1),(K-1)+(n-1)} \\ &\quad +F^2_{K-(n+1),(K-1)+n} \\ &= F_{K-n,(K-1)+n} \end{aligned} \quad (18)$$

for $0\leq n\leq I-1$. Inequality (18) yields

$$F_{K-(n+2),(K-1)+(n+2)}\leq F_{K-n,(K-1)+n} \quad (19)$$

for $0\leq n\leq I-2$. Using the induction assumption again, we have

$$F_{I-(n+2),(K-1)+(n+2)}\leq F_{I-n,(K-1)+n} \quad (20)$$

is valid for $1\leq I\leq K-1$, $0\leq n\leq I-2$. Combined with (19), inequality (20) holds for $1\leq I\leq K$, $0\leq n\leq I-2$. Therefore,

$$\begin{aligned} F_{I-(n+1),K+(n+1)} &= 1+F^2_{I-(n+2),(K-1)+(n+2)}+F^2_{I-(n+1),K+n} \\ &\leq 1+F^2_{I-n,(K-1)+n}+F^2_{I-(n+1),K+n} \\ &= F_{I-n,K+n} \end{aligned} \quad (21)$$

for $1\leq I\leq K$, $0\leq n\leq I-2$. For the case when $J=K$ and $n=I-1$, we only need to show $F_{0,I+K}\leq F_{1,I+K-1}$, which is obvious as $F_{1,I+K-1}=1+F^2_{0,I+K-1}+F^2_{0,I+K-2}\geq 1+F^2_{0,I+K-1}=F_{0,I+K}$. Therefore, (21) holds for $1\leq I\leq J=K$ and $0\leq n\leq I-1$, which completes the proof of (16). Replacing both I and J in (16) with $J-1$ and letting $n=0$ yield $F_{J-2,J}\leq F_{J-1,J-1}$.

REFERENCES

[1] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Info. Theory*, vol. 38, no. 2, pp. 713–718, March 1992.

[2] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Proc.*, vol. 2, no. 2, pp. 160–175, April 1993.

[3] S. Mallat, *A wavelet tour of signal processing*, 2nd ed. Academic Press, 1999.

[4] G. J. Sullivan and R. L. Baker, "Efficient quadtree coding of images and video," *IEEE Trans. Image Proc.*, vol. 3, no. 3, pp. 327–331, May 1994.

[5] R. Leonardi and M. Kunt, "Adaptive split and merge for image analysis and coding," in *Proc. SPIE*, vol. 594, 1985.

[6] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Proc.*, vol. 40, no. 8, pp. 2040–2060, August 1992.

[7] R. Shukla, P. L. Dragotti, M. N. Do, and M. Vetterli, "Rate-distortion optimized tree structured compression algorithms for piecewise smooth images," *IEEE Trans. Image Proc.*, vol. 14, no. 3, pp. 343–359, 2005.

[8] D. L. Donoho, "Wedgelets: Nearly-minimax estimation of edges," *Ann. Statist.*, vol. 27, pp. 859–897, 1999.

[9] M. H. Gross, O. G. Staadt, and R. Gatti, "Efficient triangular surface approximations using wavelets and quadtree data structures," *IEEE Trans. Visualization Computer Graphics*, vol. 2, no. 2, pp. 130–143, June 1996.

[10] M. V. Wickerhauser, *Adapted wavelet analysis from theory to software*. A K Peters, 1994.

[11] D. L. Donoho, "CART and best-ortho-basis: A connection," *Ann. Statist.*, vol. 25, pp. 1870–1911, 1997.

[12] N. N. Bennet, "Fast algorithm for best anisotropic Walsh bases and relatives," *Appl. and Comput. Harmonic Analysis*, vol. 8, pp. 86–103, 2000.

[13] D. Xu and M. N. Do, "Anisotropic 2-D wavelet packets and rectangular tiling: Theory and algorithms," in *Proc. SPIE Conf. on Wavelets X*, San Diego, August 2003, pp. 619–630.

[14] W. S. Lee, "Tiling and adaptive image compression," *IEEE Trans. Info. Theory*, vol. 46, no. 5, pp. 1789–1799, August 2000.

[15] C. Herley, J. Kovačević, K. Ramchandran, and M. Vetterli, "Tilings of the time-frequency plane: Construction of arbitrary orthogonal bases and fast tiling algorithms," *IEEE Trans. Signal Proc.*, vol. 41, no. 12, pp. 3341–3359, December 1993.

[16] C. Herley and M. Vetterli, "Orthogonal time-varying filter banks and wavelet packets," *IEEE Trans. Signal Proc.*, vol. 42, no. 10, pp. 2650–2663, October 1994.

[17] M. V. Wickerhauser, "Some problems related to wavelet packet bases and convergence," *Arab. J. Sci. and Eng.*, vol. 28, no. 1C, pp. 45–58, June 2003.

[18] C. Herley, Z. Xiong, K. Ramchandran, and M. T. Orchard, "Joint space-frequency segmentation using balanced wavelet packet trees for least-cost image representation," *IEEE Trans. Image Proc.*, vol. 6, no. 9, pp. 1213–1230, September 1997.

[19] C. M. Thiele, "A fast algorithm for adapted time-frequency tilings," *App. Comp. Harmon. Anal.*, vol. 3, pp. 91–99, 1996.

[20] J. R. Smith and S.-F. Chang, "Frequency and spatially adaptive wavelet packets," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Detroit, MI, 1995.

[21] Y. Huang, I. Pollak, M. N. Do, and C. A. Bouman, "Fast search for best representations in multitree dictionaries," *IEEE Trans. Image Proc.*, to appear.

[22] A. V. Aho and N. J. A. Sloane, "Some doubly exponential sequences," *Fibonacci Quart.*, vol. 11, pp. 429–437, 1970.