

# Brief Article

The Author

January 13, 2004

## 0.1 Wavelet Transform: Quad Tree

The primary two addition to the wavelet transform in case of the Quad Tree variety is the tree and spare matrix. The spare matrix is used to keep the transform operation separate during computation. This may not be necessary in distributed versions.

The quad tree node has 9 elements side from links to

- Energy (the section energy)
- coordinates for the upper left corner
- coordinates for the lower left corner
- coordinates for the upper right corner
- coordinates for the lower right corner

The procedure for the quad tree wavelet transform involves loading the root node with its coordinates and the energy of the matrix. This energy is used as an early stopping condition. In the non-distributed form, it useful to load a temporary matrix global to the class/ object for the purpose of memory conservation, and system call conservation.

Given quad tree node, The steps are to compute the column wavelet transform followed by the row transform on matrix one and matrix two. Following the computation of the wavelet transform, the energy for each segment must be computed. If the energy limit (epsilon) or size limit has been reached, then stop procedure must be initiated. Otherwise, the computation is continued for the 4 sub-components produced by the current wavelet transform.

The quad tree requires another feature in its structure, breadth first and depth first searching. The purpose behind these searches is to extract the structure of the wavelet quad-tree transform and store it in an array. This array can then be stored in a database, file, or passed to another procedure or returned to a calling procedure. In the case of the leaves, this array can be made to store the actual matrix data, and its dimensions in another array.

The breadth first search requires a queue structure to hold pointers to the tree nodes. First the root is put queue. While the queue is not empty, the bread first search works on elements in the queue in a loop.

First step in the loop is to pop the element in the queue. Then stores that information. Next the four children are inserted into the queue. Then loop is repeated. The only time that elements are not added to the queue is if they themselves are empty. Eventually, the leaves are all reached, which have empty children. This cause the queue to exhaust itself, and terminate the loop.

The depth first search essentially uses a stack, which implies that a recursive method can also be used. The visit procedure is as follows:

In particular there are two places where these traversal methods are necessary for the search, store, retrieval, transform, and inverse transform. The first three are somewhat obvious. The transform and inverse transform are not trivial.

In the depth first method, the transform is performed first for the average branches, starting at the root. Once the average branch is transformed, the vertical branch is transformed by treating the first vertical child as its own root, and recursively repeating the transform. This is again done for the horizontal and diagonal components.

The breadth first method first transforms the root, the same as the other methods. Next, the branches formed by the transform are placed into a queue, and those elements are then transformed. The step is repeated for subsequent transforms of each branch.

A hybrid approach takes the four branches after the transform uses size to evaluate whether to breadth first or depth first. In the case of depth first, all four branches performed by separate threads of execution since the operations are independent of one another.

### 0.1.1 Tree - Queue representation.

For efficiency in speed, this implementation uses an array based tree and queue. The reason is for ease of traversal, copying, storing, and retrieving. The relation on the the array is as follows:

- $i_r = \frac{i_c-1}{4}$
- $i_a = 4i_r + 1$
- $i_a = 4i_r + 2$
- $i_a = 4i_r + 3$
- $i_d = 4i_r + 4$

Second is the construction of the tree data being stored. For these experiments, wave-node keeps the energy of the matrix, the maximum value, the minimum value, the four corners, width, and width.

# 1 File Acquisition

This is an orthogonal but important issue for a practical implementation of the wavelet transform. It has very little to do with the mathematics except for setting up the input to the wavelet transform. There are four categories to be considered by this project: the array loader, array saver, WX group, and the WIX group.

1. Tree - Array Loader
  - (a) Data File
  - (b) Database
2. Tree - Array Saver
  - (a) Data File
  - (b) Database
3. WX Group
  - (a) Default Root
  - (b) Designated Root
4. WIX group
  - (a) work up to the default root
  - (b) work up to the designated root

Why is this important? The simplest case is the WX default root. In this case the goal is to load the root a copy of the matrix to be worked on from a file. However, it is important to note that a designated root can be worked on either WX or WIX, too.

In the WX designated root, consideration of the rest of tree must be accounted for. There are two ways to handle this case. One is to treat the designated root as its own tree with building the rest of the tree, use the default root scheme, and merge the designated root and its tree with the remainder of the tree after the fact. The other is have the rest of build and work on the designated root independently of the rest of the tree. These subtle differences are applied to distributed forms where memory and message passing produce a design constraint. If the application is running exclusively on the same machine and in the same memory space, then it is appropriate to make a master tree which is shared by all threads and procedures that are working on the transform. In case that the same memory space may not be used, then it is appropriate to split up the tree into independent trees and merge them once computation is complete and they are brought together. In the case of WX, this calculation starts from a root and spans to the children. The case of splitting up the tree is in case of computing the children once they are independent of their siblings.

The case of WIX is almost opposite of the WX case. The computation starts from the leaves and works up to the root, designated or default. It is important for the file acquisition to map files to their appropriate leaves, and load them. There are multiple file formats which may be practical.

- Single File
- Single Master File and Multiple Children Files
- Database of files and Multiple Children Files
- Database with master children as records