

Custom Wavelet Packet Image Compression Design*

Mladen Victor Wickerhauser †

August 28, 1996

Abstract

This tutorial paper presents a meta-algorithm for designing a transform coding image compression algorithm specific to a given application. The goal is to select a decorrelating transform which performs best on a given collection of data. It consists of conducting experimental trials with adapted wavelet transforms and the best basis algorithm, evaluating the basis choices made for a training set of images, then selecting a transform that, on average, delivers the best compression for the data set. A crude version of the method was used to design the WSQ fingerprint image compression algorithm.

1 Introduction

No single image compression algorithm can be expected to work well for all classes of digital images. The sampling rates, frequency content, and pixel quantization all influence the compressibility of the original data. Subsequent machine or human analyses of the compressed data, or its presentation at various magnifications, all influence the nature and visibility of distortion and artifacts. Thus compression standards like those of the JPEG committee [1], established for a “natural” images intended to be viewed by humans, do not satisfy the requirements for compressing fingerprint images intended to be scanned by machines. In that particular example, it was necessary to develop a new algorithm *WSQ* [2].

Both JPEG and WSQ are examples of *transform coding* image compression algorithms. That class provides a rich selection from which custom compression algorithms may be chosen. This paper presents a meta-algorithm for rationally and automatically choosing one of them to suit a particular application. It focuses on the transform portion of the compression algorithm: the *best basis method* is used to optimize it to provide the best average compression of a representative set of images, subject to speed constraints. A crude version of the method was used to design the WSQ fingerprint image compression algorithm.

2 Transform coding image compression

The generic transform coding compression scheme is depicted in Figure 1. It consists of three pieces:

- *Transform:* Apply a function, which is invertible or *lossless* in exact arithmetic, which should decorrelate the pixels in the image. It does this by decomposing the image into a superposition of independent patterns; it produces a sequence of floating-point amplitudes which are the intensities of the new components.
- *Quantize:* Replace the transform amplitudes with (small) integer approximations. This is the *lossy* or non invertible part of the algorithm, where all the distortion is introduced.
- *Code:* Rewrite the integer stream of quantized transform coefficients into a more efficient alphabet, so as to approach the information-theoretic minimum bit rate. This operation is akin to a table lookup, and is invertible.

These three steps are depicted in Figure 1.

*Research partially supported by NSF, AFOSR, and Southwestern Bell Corporation

†Department of Mathematics, Washington University, St. Louis, Missouri, 63130 USA

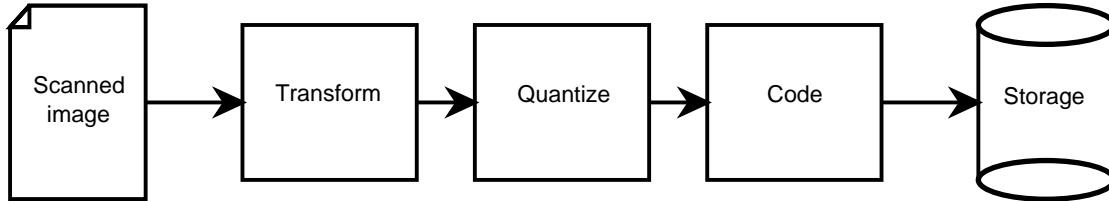


Figure 1: Generic transform coding image compression device.

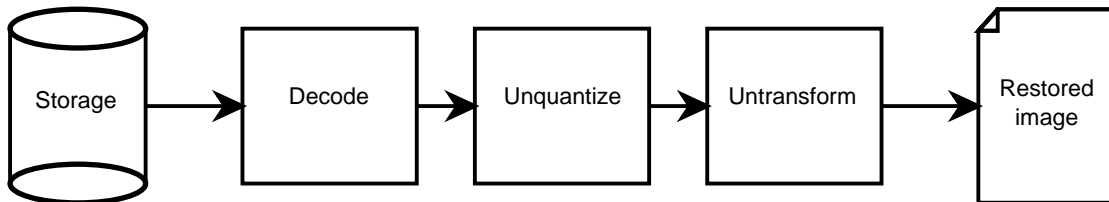


Figure 2: Inverse of the generic transform coder: the decoder.

To recover an image from the coded, stored data, the steps in Figure 1 are inverted as shown in Figure 2. The first and third blocks of the compression algorithm are exactly invertible in exact arithmetic, but the *Unquantize* block does not in general produce the same amplitudes that were given to the *Quantize* block during compression. The errors thus introduced can be controlled both by the fineness of the quantization (which limits the maximum size of the error) and by favoritism (which tries to reduce the errors for certain amplitudes at the expense of greater errors for others).

The *compression ratio* produced by such an algorithm is computed by dividing the size of the input file by the size of the output file. It thus takes into account all of the side information stored with the output file that is needed for reconstruction. Roughly speaking, if the coding step is perfectly efficient, the compression ratio is maximized for a given distortion when the transform and quantize steps produce a sequence with minimal entropy. However, since minimal entropy is hard to characterize and harder to achieve, it is better to aim at a broader target: a sequence with almost all of the values being zero. Such a sequence will have a low, if not minimal, entropy, since its value distribution will be highly peaked at zero.

This paper concentrates on the *Transform* operation. The goal is to choose, from a large family of wavelet, wavelet packet, and local trigonometric transforms, the one which can be expected to yield the largest fraction of negligible amplitudes on data represented by a training set. Those will be quantized to zero in exchange for a given degree of distortion, yielding the biggest peak at zero in the value distribution and resulting in the best compression. It will be assumed that the transforms are orthogonal or nearly orthogonal, so that their condition number is close to 1 and they introduce no significant redundancy.

3 Custom transforms

There are two fast ways to decompose images at the transform step: splitting into small blocks of pixels and then applying some fast transform to the blocks, or splitting the whole image into frequency subbands by convolving with short filters. Both methods cost $O(P \log P)$ operations for an P -pixel image. Detailed formulas and a proof of the complexity statement can be found in Reference [5], so only a brief summary will be presented here.

In the pixel splitting scheme, the image is cut into blocks, either of fixed or variable size, but small enough so that the intensities of all pixels contained within a block are correlated. This cutting is depicted in Figure 3. Then decorrelation is performed by applying the two-dimensional discrete cosine transform (DCT) to the blocks. This method is used in the JPEG still picture image compression standard [3]. The resulting amplitudes represent spatial frequency components in the blocks. Because digitized images are often limited in their spectral content, most of the amplitudes in each block will be negligible. To maximize the proportion

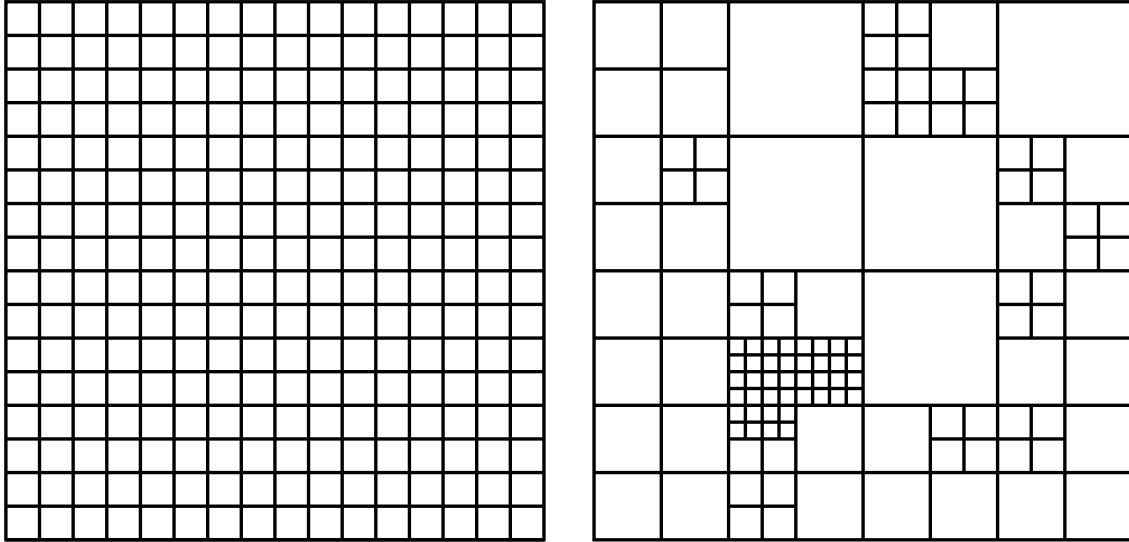


Figure 3: Division of a 128×128 pixel image into 8×8 blocks, as in JPEG, or into blocks varying from 4×4 to 32×32 .

of negligible amplitudes, the blocks should be chosen as large as possible subject to the constraints that (1) only a few spatial frequencies are present in each block, and (2) describing the block boundaries does not create too much side information.

In the subband splitting scheme, a low-pass and a high-pass filter are used along rows and columns to split the image into four subimages characterized by restricted frequency content. This process is repeated on the subimages, down to some maximum depth of decomposition, resulting in a segmentation of frequency space into subbands. Two such segmentations are depicted in Figure 4; the one on the right is used in the WSQ fingerprint image compression algorithm [2]. The resulting amplitudes again represent spatial frequency components, computed over portions of the picture determined by the depth of the subband and the location of the amplitude in its subband. Again, for images of limited spectral content, most of these amplitudes will be negligible. The two example subband decompositions are approximately radial with respect to the “origin” in the upper left hand corner; this works well for isotropic images, *i.e.*, where no direction is favored over any others.

4 The joint best basis

Both splitting schemes can be organized as quadtrees to a specified depth, with the selected transform determined by the leaves of a subtree like the one depicted in Figure 5. To choose the subtree and thus the transform, each member of a representative training set of images is decomposed into the complete quadtree of amplitudes. Then the squares of these amplitudes are summed into a sum-of-squares quadtree. Using an information cost function such as “number of nonnegligible amplitudes”, the sum-of-squares quadtree is searched for its *best basis*, which is the one that minimizes this cost ([5], p. 282). Figure 6 depicts this algorithm. The best basis for the Σ quadtree is the *joint best basis* for the training set of images $1, 2, \dots, N$. That is the transform which produces, on average, the largest number of negligible output coefficients.

To find the best basis requires examining each coefficient in the quadtree and examining each subband or pixel block at most twice, which means that the complexity is $O(P \log P)$ for P -pixel images. To find the joint best basis requires building the sum-of-squares tree first, which dominates the total complexity with its $O(NP \log P)$ cost for a training set of N P -pixel images.

Of course, the joint best basis transform is only optimal within its own class, and the class is determined by the technical details and mathematical properties of the splitting algorithm. If these constraints were removed and the search performed over all orthonormal transforms, then the joint best basis will be the

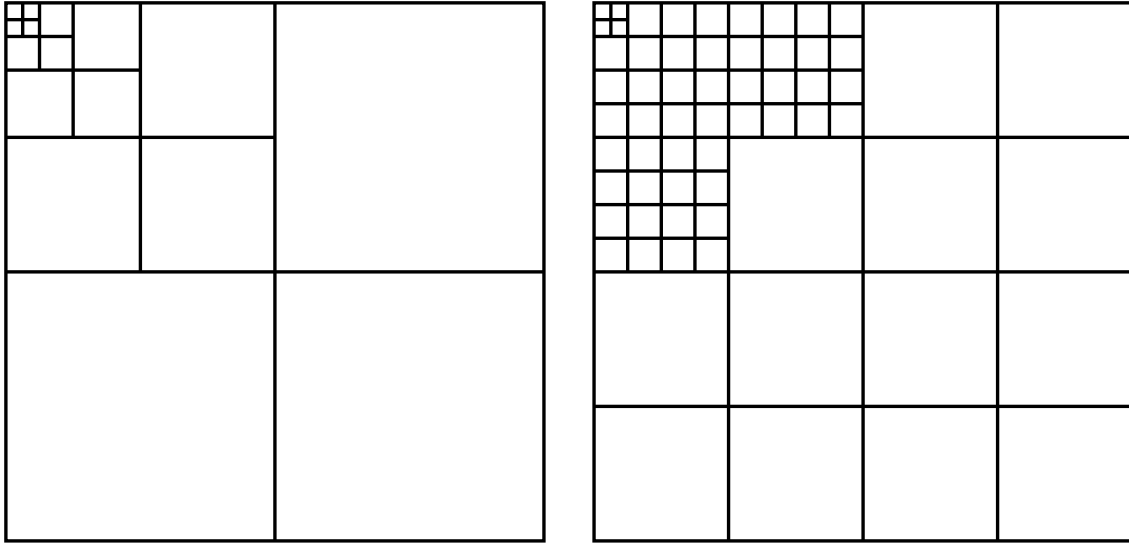


Figure 4: Division of an image into orthogonal wavelet subbands to level 5, or into the WSQ subbands. Frequencies increase down and to the right.

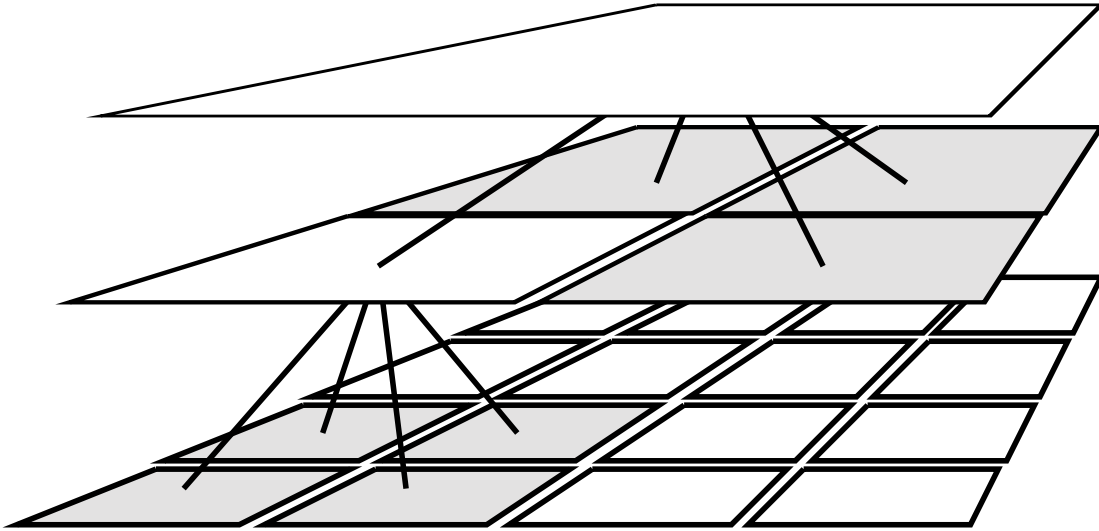


Figure 5: Splitting schemes produces quadtrees; custom bases are determined by the leaves of a subtree such as the one shown here, shaded for emphasis.

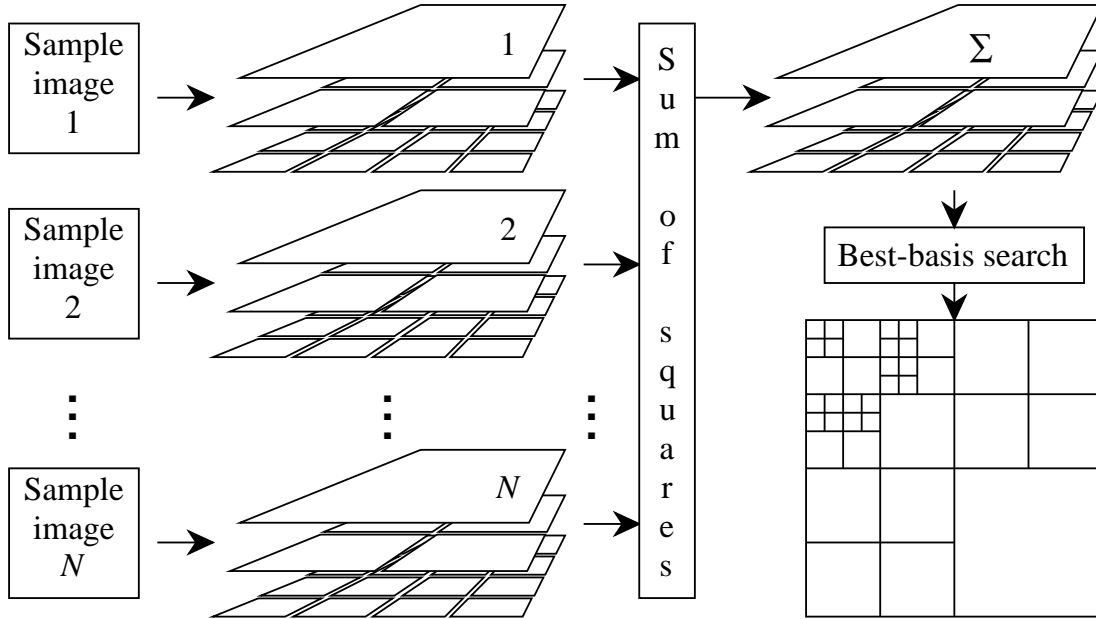


Figure 6: A joint best basis from a class of splitting algorithms is determined by a sample set of N images.

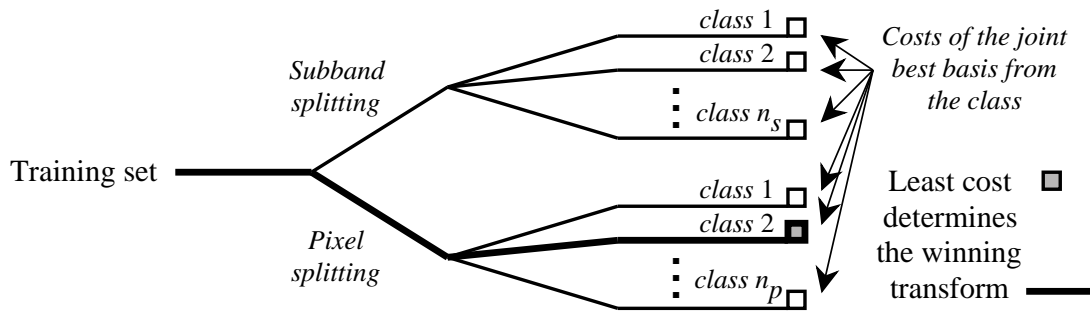


Figure 7: A meta-algorithm for deciding which splitting algorithm to use with a particular class of images.

Karhunen-Loève (KL) or *principal orthogonal* basis [4], which is known to be the minimizer of the number of nonnegligible amplitudes. With the constraints, whose purpose is to speed things up, the chosen transform is just an approximation to KL.

5 Choosing the best transform from multiple classes

There is a *meta-algorithm* for relaxing the constraints a bit while preserving the speed. Namely, a custom transform can be chosen by checking many classes of splitting algorithms in order to further increase the expected number of negligible coefficients. This scheme was first proposed by Yves Meyer, and is depicted in Figure 7. At the end of each path is a cost figure, the expected number of nonnegligible coefficients for the training set of images. The path that leads to the lowest cost determines which algorithm should be used to find the custom transform for compressing the images represented by the training set.

Examples of different classes are the different subband splitting schemes associated to different conjugate quadrature filters ([5], Chapter 5 and Appendix C), or the adapted local trigonometric bases determined by different windows ([5], Chapters 3 and 4).

6 Conclusion

Given a training set of images, a transform coding image compression algorithm may be rationally chosen from a class of fast splitting algorithms. The choice criterion is a cost function that, when low, yields high compression ratios for transform coding image compression. The method works for wavelet packet and local trigonometric transforms and thus produces well-conditioned compression and decompression methods of complexity $O(P \log P)$ for P -pixel images. Searching for the best choice itself costs $O(NP \log P)$, where N is the number of training images.

References

- [1] ISO/IEC JTC1 Draft International Standard 10918-1. Digital compression and coding of continuous-tone still images, part 1: Requirements and guidelines. Available from ANSI Sales, (212)642-4900, November 1991. ISO/IEC CD 10918-1 (alternate number SC2 N2215).
- [2] IAFIS-IC-0110v2. WSQ gray-scale fingerprint image compression specification. Version 2, US Department of Justice, Federal Bureau of Investigation, 16 February 1993.
- [3] Gregory K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34:30–44, April 1991.
- [4] Mladen Victor Wickerhauser. Fast approximate factor analysis. In Martine J. Silbermann and Hemant D. Tagare, editors, *Curves and Surfaces in Computer Vision and Graphics II*, volume 1610 of *SPIE Proceedings*, pages 23–32, Boston, October 1991. SPIE.
- [5] Mladen Victor Wickerhauser. *Adapted Wavelet Analysis from Theory to Software*. AK Peters, Ltd., Wellesley, Massachusetts, 9 May 1994. With optional diskette.