Lossy Compression of Partially Masked Still Images

Léon Bottou
AT&T Labs Research
100 Schultz Dr.
Red Bank, NJ07701-7033
(732) 345-3336
leonb@research.att.com

Steven Pigeon
Université de Montréal
Pavillon Math-Info, #3339
2900 Edouard-Montpetit
Montreal, Quebec, Canada H3T 1J4
pigeon@iro.umontreal.ca

April 3, 1998

Abstract

Books and magazines often contain pages containing audacious mixtures of color images and text. Our problem consists in coding the background colors of a such documents without wasting bits on background pixels that will be masked by foreground text.

This paper presents a numerical method for achieving such a result. This wavelet technique described in this paper significantly reduces the compressed file sizes. It can handle arbitrarily complex masks with reasonable computational requirements. This direct method does not require the prior computation of a mask-dependent wavelet basis.

1 Introduction

Books and magazines often contain pages containing audacious mixtures of color images and text. Our reference application consists in coding the background colors of a such documents without wasting bits on background pixels that will be masked by foreground text.

Some wavelet image compression schemes control bit allocation using an a priori classification of image blocks (Jafarkhani and Farvardin, 1997). This procedure however is hardly applicable to our problem, since blocks are usually larger than the detailed shape of typical foreground characters. Other approaches consist of generating a suitable wavelet basis (Apostolopoulos, 1997) before performing the wavelet decomposition.

We present in this paper a simple and direct numerical method for setting a large number of wavelet coefficients to zero, while transforming the remaining wavelet coefficients in order to preserve the pixels located outside the mask. Suitable coding schemes then take advantage of these canceled coefficients. The next section exposes the principle of our iterative method. The following sections discusses its convergence and describes an improved method. Empirical results are presented in the last section.

2 Principle

2.1 Wavelet Image Compression

Multi-resolution wavelet decomposition is one of the most efficient schemes for coding color images. These schemes involve several operations: color space transform, image decomposition, coefficient quantization and coefficient coding.

The image is first represented as a linear combination of locally supported wavelets. This linear combination can be written as:

$$Aw = x \tag{1}$$

where w is a vector of wavelet coefficients, x is a vector of pixel values, and A is a square matrix whose columns represent the wavelet basis. Matrix A usually describes an orthogonal or nearly orthogonal transformation. Efficient multi-scale algorithms (Mallat, 1989; Sweldens, 1996) perform image decomposition (i.e. computing $A^{-1}x$) and image reconstruction (i.e. computing Aw) in time proportional to the number of pixels in the image.

The image local smoothness ensures that the distribution of the wavelet coefficients is sharply concentrated around zero. High compression efficiency is achieved using quantization and coding schemes that take advantage of this peaked distribution (Adelson, Simoncelli and Hingorani, 1987; Shapiro, 1993).

2.2 Partial Wavelet Masking

The visible pixels (i.e. pixels that are not masked) are never affected by the coefficients of wavelets whose support is *entirely located below the mask*. Therefore, a simple idea for solving our problem consists in either (a) skipping these coefficient while coding, or (b) setting them to zero, which is the most code-efficient value. The first solution saves a few bits, but requires to know the mask when decoding the compressed image file. The second solution does not suffer from this constraint: the compressed image file can be decoded with the usual algorithms, regardless of the mask.

This approach is unpractical when the mask (i.e. the set of masked pixels) is composed of small connected components. Efficient wavelet compression requires smooth wavelets whose smallest support is typically a dozen pixels wide. The backgrounds of our reference documents are masked by printed text. Very few wavelets will be entirely masked by the ink of typical printed characters.

Most of the information about masked background pixels is carried by wavelets whose support is *partially masked only*. Canceling the coefficient of a partially masked wavelet obviously changes pixels located outside the mask. The coefficient of other wavelets must be adjusted to compensate for this effect. The adjusted coefficients represent an image whose visible pixels exactly match the corresponding pixels of the target image. The masked pixels however can be different. Their value is simply a code-efficient interpolation of the visible pixels.

Reordering the pixel vector x and the wavelet coefficient vector w allows a block-decomposition of equation (1) :

$$Aw = \begin{pmatrix} B & C \\ D & E \end{pmatrix} \cdot \begin{pmatrix} w' \\ w'' \end{pmatrix} = \begin{pmatrix} x' \\ x'' \end{pmatrix} = x \tag{2}$$

where x'' represents the masked pixels, x' represents the visible pixels, w'' represents the wavelet coefficients to cancel, and w' represents the remaining wavelet coefficients. We are seeking adjusted wavelet coefficients which are the solutions of equation :

$$\begin{aligned} Bw' &= x' \\ w'' &= 0 \end{aligned} \tag{3}$$

This equation has solutions if the rank of the rectangular matrix B is equal to the number of remaining (i.e. non canceled) wavelet coefficients. This problem will be addressed in section 3.1. The rank condition however implies that the number of canceled wavelet coefficients must be smaller than the number of masked pixels.

Given a mask and a wavelet decomposition, we must therefore choose which wavelet coefficients (called the *masked coefficients*) will be canceled. The choice of the masked coefficients impacts the resulting file size. Canceling a wavelet whose energy is significantly located outside the mask requires a lot of adjustments on the remaining coefficients. These adjustments are likely to use coefficients that would be null otherwise. We have empirically found that good results are achieved by canceling wavelet coefficients as soon as half the energy of the wavelet is located below the mask.

2.3 Successive Projections

Once we have selected the set of masked coefficients, we can proceed and solve equation (3). There are many techniques for solving sparse linear systems. There is however a method which relies only on the efficient wavelet decomposition and reconstruction algorithms.

Every image can be represented in pixel coordinates (i.e. a collection of pixel values) or in wavelet coordinates (i.e. a collection of wavelet coefficients). The coordinate transformation is described by matrix A. The solutions of our problem belong to the intersection of the following sets of images:

- The set $\mathcal{P}$ of all images whose pixels located outside the mask are equal to the corresponding pixels in the image being compressed. This set is a closed convex affine subspace of the image space.
- The set $\mathcal{Q}$ of all images whose wavelet representation contains zeroes for all masked coefficients. This set also is a closed convex affine subspace of the image space.

Let P (respectively Q) be the projection operator on set $\mathcal{P}$ (respectively $\mathcal{Q}$). The initial image x_0 is already an element of set $\mathcal{P}$. As shown in figure 1, we successively project this image on sets $\mathcal{Q}$ and $\mathcal{P}$:

$$\begin{aligned} x'_i &= Qx_i & \in \mathcal{Q} \\ x_{i+1} &= Px'_i = PQx_i & \in \mathcal{P} \end{aligned} \tag{4}$$

This sequence is known (Youla and Webb, 1982) to converge toward a point in the intersection of convex sets $\mathcal{P}$ and $\mathcal{Q}$, provided that this intersection is non empty. The simplest version of the *Successive Projections Algorithm* (Chen, Civanlar and Haskell, 1994) therefore consists of the following steps:

i) Initialize a buffer with the pixel values of the initial image.
ii) Perform the wavelet decomposition.
iii) Set all masked wavelet coefficients to zero (projection Q).
iv) Perform the image reconstruction.

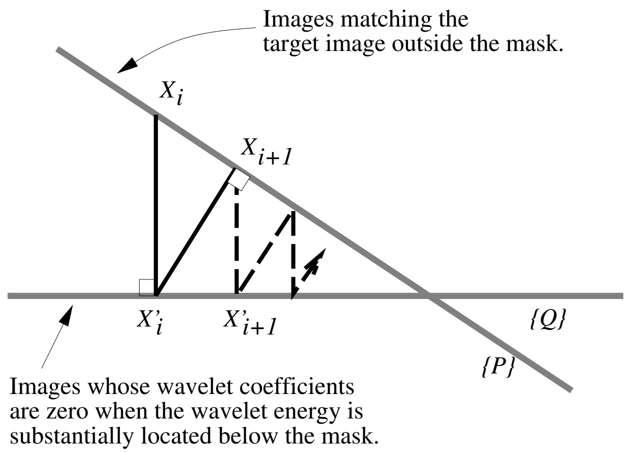


Figure 1: *Successive projections converge towards a point in the intersection of two convex subspaces representing (a) images matching the initial image outside the mask, and (b) images whose masked wavelet coefficients are zero.*

v) Reset all visible pixels to their value in the initial image (projection P).

vi) Loop to step *(ii)* until convergence is reached.

Convergence is easily monitored by measuring the distance between the visible pixels of the initial image and the corresponding pixels of the image reconstructed in step *(iv)*.

3 Convergence Issues

3.1 Convergence Speed

This section presents a bound on the convergence speed and a criterion on the existence of a solution. The bound depends only on the set of masked pixels and the set of masked coefficients. This bound therefore is a useful element for selecting the masked coefficients.

Since $x'_{i+1} = Q(x_{i+1})$ is the orthogonal projection of x_{i+1} on $\mathcal{Q}$, we have (cf. figure 1) :

$$\|x_{i+1} - x'_i\|^2 = \left\|x_{i+1} - x'_{i+1}\right\|^2 + \left\|x'_{i+1} - x'_i\right\|^2 \geq \left\|x_{i+1} - x'_{i+1}\right\|^2 \tag{5}$$

The contraction ratio is therefore bounded by :

$$\frac{\left\|x_{i+1} - x'_{i+1}\right\|^2}{\|x_i - x'_i\|^2} \leq \frac{\|x_{i+1} - x'_i\|^2}{\|x_i - x'_i\|^2} \tag{6}$$

Vector $x_i - x'_i = x_i - Q(x_i)$ belongs the linear subspace orthogonal to $\mathcal{Q}$. It can be written as a linear combination of the wavelets e_j corresponding the the masked coefficients. We have then the following results :

$$x_i - x'_i \quad = \quad \sum_j \alpha_j e_j \tag{7}$$

$$x_{i+1} - x'_i = x_i - x'_i - P(x_i - x'_i) \quad = \quad \sum_j \alpha_j (e_j - P(e_j)) \tag{8}$$

Vector $e_j - P(e_j)$ represents the part of wavelet e_j which is not located below the mask. These *clipped wavelets* are completely defined by the mask and by the set of masked coefficients. Combining results (6), (7) and (8) provides a bound λ on the contraction ratio. This bound depends only on the set of masked pixels and the set of masked coefficients.

$$\frac{\left\|x_{i+1} - x'_{i+1}\right\|}{\left\|x_i - x'_i\right\|} \;\leq\; \sup_{\left\|\sum_j \alpha_j e_j\right\| = 1} \left\| \sum_j \alpha_j (e_j - P(e_j)) \right\| \;=\; \lambda \tag{9}$$

The right hand side of this inequality is easily interpretable. Adding a unit vector to the masked coefficients will cause a perturbation on the visible pixels. The norm of this perturbation will be less than *lambda*. Quantity λ naturally depends on the energy and shape of the part of the masked wavelets which overlaps visible pixels.

An argument similar to (5) ensures that $\|x_{i+1} - x_i\| \leq \|x_i - x'_i\|$ This result and inequality (9) provide bounds on the convergence speed :

$$\|x_{i+1} - x_i\| \;\leq\; \|x_i - x'_i\| \;\leq\; \lambda^i \;\; \|x_0 - x'_0\| \tag{10}$$

Condition $\lambda < 1$ therefore is a sufficient condition for ensuring that both sequences $(x_i) \in \mathcal{P}$ and $(x'_i) \in \mathcal{Q}$ converge geometrically to a same point x^*. This limit belongs to both $\mathcal{P}$ and $\mathcal{Q}$ because these sets are closed sets.

This result defines a remarkably fast convergence. The successive projection method reaches a solution with a predetermined accuracy after a number of iterations proportional to the *logarithm* of the number N_m of masked pixels only, as shown by equation (10) and the following bound :

$$\|x_0 - x'_0\| \leq \|x_0 - x^*\| + \|x'_0 - x^*\| \leq 2\,\|x_0 - x^*\| \leq 2\sqrt{N_m}$$

As a comparison, solving equation (3) with a typical sparse linear system technique, like the conjugate gradients method, would require a number of iterations proportional to the number N_v of visible pixels[1].

3.2 Multi-scale Successive Projections

This section shows how the multi-scale nature of the wavelet decomposition algorithm provides a way to improve the value of λ and therefore improve the convergence speed.

Developing the norm of the pixel perturbation term in equation (9) shows how quantity λ depends on the shapes and the scales of the set of masked wavelets :

$$\left\| \sum_j \alpha_j (e_j - P(e_j)) \right\|^2 \;=\; \sum_j \alpha_j^2 \,\|e_j - P(e_j)\|^2 \tag{11}$$

$$+ \;\sum_{j \neq k} \alpha_j \alpha_k \;(e_j - P(e_j)).(e_k - P(e_k)) \tag{12}$$

[1]The execution time of one iteration is proportional to the number of pixels in the case of successive projections, and to the number of visible pixels in the case of conjugate gradients.

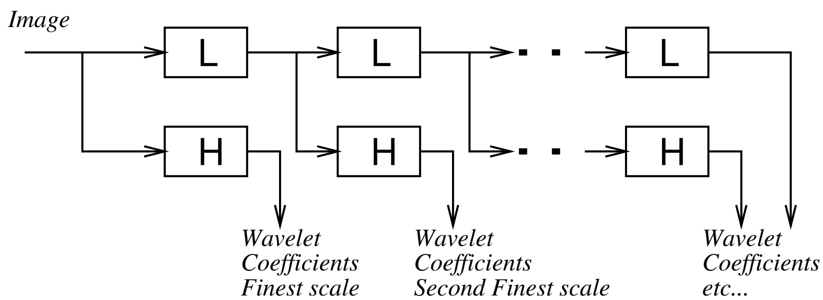


Figure 2: *Multi-scale wavelet decomposition works by repetitively applying a low-pass filter(L) which returns an image with half resolution, and a complementary high pass filter, which returns all wavelet coefficients at the current scale.*

The first terms of the sum (11) depends on the norm of the clipped wavelets $e_j - P(e_j)$. Since the wavelets e_j are normalized, and since we cancel only the wavelets whose support is substantially masked, the norm of the clipped wavelets is a quite small number (typically smaller than 1/2). The second terms (12) depend on the overlaps between clipped wavelets. Clipped wavelets of similar scale (i.e. wavelets whose support has identical size) are not likely generate much overlap, because they are designed to cover efficiently the pixel space. Large scale wavelets however overlap many small scale wavelets. These overlaps drive up the value of λ.

Multi-scale wavelet decomposition algorithms factor the decomposition (i.e. multiplying the image pixel by matrix A^{-1}) into a sequence of identical stages (cf. figure 2). Each stage consists in applying a low pass linear filter and a high pass linear filter to the input image. The low pass filter returns a half resolution image which is provided as input to the next stage. The high pass filter returns all the coefficients of wavelets a particular scale. The input image of each stage can be reconstructed by combining the output of both filters.

Since all the wavelets coefficients for the finest scale are produced by the first stage, we can cancel all masked coefficients for this scale using the Successive Projections Algorithm (cf. section 2.3) with a one stage decomposition only. This operation outputs a half-resolution image and a first set of coefficients fulfilling the masking conditions. The visible pixels of the initial image can be reconstructed by combining these outputs with the usual algorithms. We can then process the wavelet coefficients for the coarser scales by repeating this operation for each successive stage in the wavelet transform. In other words, the *Multi-scale Successive Projections Algorithm* consists of the following operations:

i) Initialize the current image with the pixels of the image being compressed. Initialize the current mask with the set of pixels that will be masked by foreground objects.

ii) Apply the Successive Projections Algorithm (cf. section 2.3) on the current image, using a *one-stage* wavelet decomposition only.

iii) Set the current image to the half resolution image returned by the low-pass wavelet filter. Set the current mask to a half resolution mask in which a pixel is masked if the corresponding pixels in the previous mask were masked.

iv) Go to step *(ii)* until all stages of the multi-scale wavelet decomposition have been processed.

This algorithm has been found to run one order of magnitude faster on realistic images than the

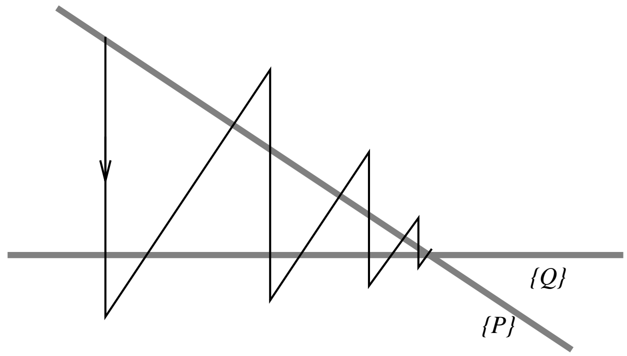


Figure 3: *In a high dimensional spaces (like image spaces), overshooting the projections usually leads to faster convergence.*

simple Successive Projections Algorithms (cf. section 2.3). This improvement is explained by the smaller values of λ and by the lower complexity of the projection operations (each stage of the algorithm processes an image whose size is half the size of the previous image).

3.3 Overshooting

Another speedup can be obtained by applying an *overshooting* technique widely applied for successive projections onto convex sets (Youla and Webb, 1982; Sezan and H., 1982). Instead of successive projections, the overshooting technique uses the following sequences (cf. figure 3):

$$\begin{aligned} x'_i &= x_i + \gamma(Q(x_i) - x_i) \\ x_{i+1} &= x'_i + \gamma(P(x'_i) - x'_i) \end{aligned} \tag{13}$$

where $0 < \gamma < 2$. Choosing $\gamma = 1$ gives the Successive Projections Algorithms as introduced in section 2.3. There are strong evidences (Simard and Mailloux, 1988) that, in high dimension spaces, choosing a higher value of γ leads to faster convergence. In our implementation, choosing $\gamma = 3/2$ in the Multi-scale Successive Projections Approximation has divided the convergence time by three.

4 Empirical Results

Experiments have been carried out using several document images (cf. figure 5) representing pages of magazines and ancient books. Foreground objects are located using a thresholding technique.

Background images are generated by taking the page image and replacing the masked pixels by the average color of the neighboring visible pixels. This averaging is achieved by dividing the page into small blocks of four by four pixels. If a block contains both masked and visible pixels, the masked pixels are set to the average value of the visible pixels in the block. The process is then repeated with bigger blocks until all masked pixels have been set with a suitable interpolation of the visible pixel colors.

These interpolated background images are then processed using the Multi-scale Successive Projections Algorithm (cf. section 3.2). The wavelet decomposition is a five stage lifting decomposition

image			compressed file size		
name	size	% masked	regular	interpolated	masked
hobby001	825x1074	19%	131359	50770	39657
hobby002	825x1074	19%	140100	64905	52425
metric	744x1074	26%	169758	34756	25899
missel	610x429	40%	64359	29550	18777
plugin	757x1035	32%	189075	40642	31065

Figure 4: *Compressed file sizes on five test images.*

using Deslauriers-Dubuc interpolating wavelets with four analyzing moments and four vanishing moments (Sweldens, 1996).

The resulting wavelet coefficients (including the zeroes generated by the masking process) are then coded using our local embedded coding scheme. The design of this scheme minimizes the decoding memory requirements. Its coding properties however are comparable to the zero-trees (Shapiro, 1993) or the set-partitioning (Said and Pearlman, 1996) schemes.

Figure 4 reports results on five images. The third column indicates the proportion of masked pixels. The fourth and fifth indicate the file sizes obtained by compressing the raw image and the interpolated background respectively. The last column contains the file size obtained using the wavelet masking technique described in this paper. All three figures were obtained using the same conservative coefficient quantification parameters.

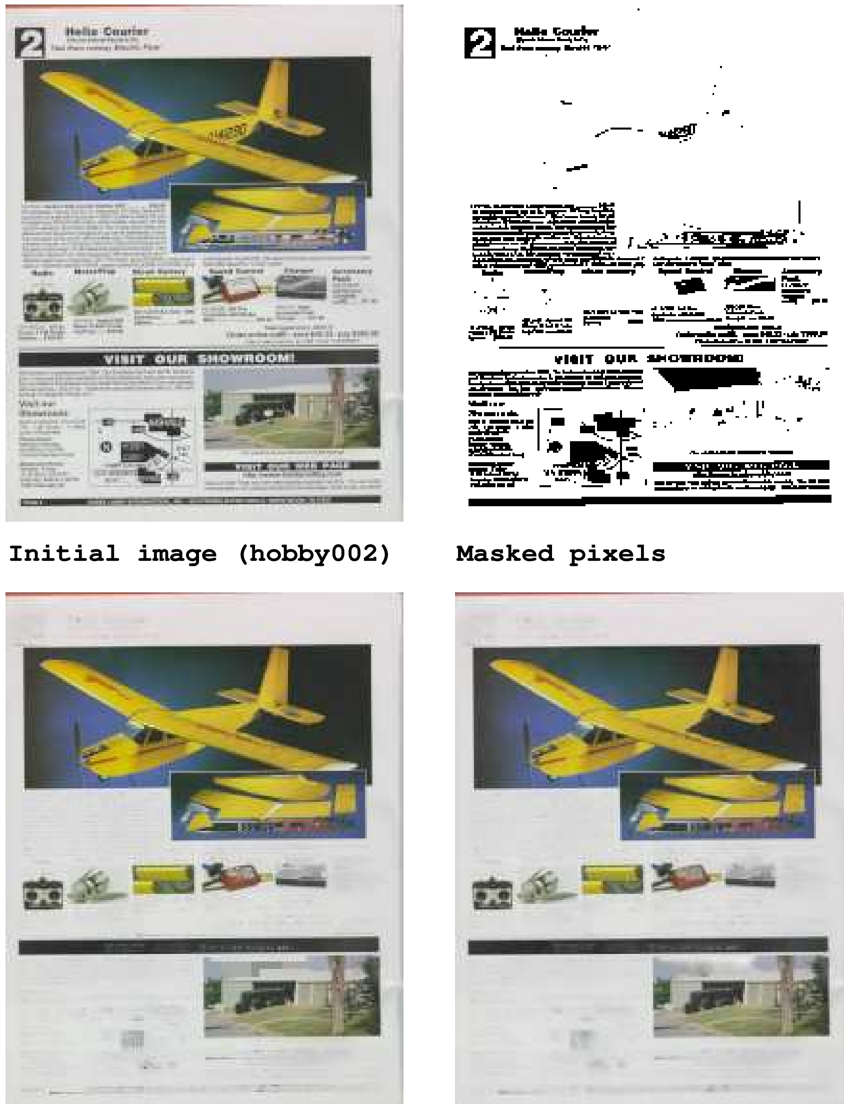
As expected, the compressed file sizes for the interpolated images are much smaller than the compressed file sizes for the raw images. The character shapes indeed represent most of the information on a document page. The wavelet masking technique further reduces the compressed file sizes. Figure 5 shows that this technique results in a smoother interpolation of the masked pixels. This smoothing can be interpreted as a code-efficient interpolation.

Our SGI workstations need about 15 seconds to perform the compression with partial wavelet masking of the biggest images (`hobby001` and `hobby002`). This is about three times the time required for a regular compression. The decompression times are not affected by the masking procedure.

5 Conclusion

The wavelet masking technique described in this paper significantly reduces the compressed file sizes. It can handle arbitrarily complex masks with reasonable computational requirements. There is no need to generate a wavelet basis whose support is restricted to the visible pixels.

The wavelet masking technique is currently used as one of the building blocks of a more general scheme for compressing full resolution color document images. This scheme, named *DejaVu*, will be described in a forthcoming paper. Yet the utilization of the wavelet masking technique has resulted in significantly smaller files and greatly improved image quality.



Initial image (hobby002)

Masked pixels

Interpolated
then Compressed
65k

Interploated
then Mask–Compressed
52k

q

Figure 5: *The wavelet masking technique is used to compress the background colors of an image* `hobby002` *(top-left). The mask (top-right) identifies the pixels masked by foreground objects. The interpolated background (bottom-left) is computed by replacing the masked pixels by the average color of neighboring visible pixels. Compressing this image while taking the mask into account (as explained in this paper) produces a smaller file (bottom-right) and a smoother interpolation.*

Acknowledgments

The authors gratefully acknowledge fruitful discussions with Patrice Simard, Hamid Jafarkhani and Yoshua Bengio.

References

Adelson, E. H., Simoncelli, E., and Hingorani, R. (1987). Orthogonal pyramid transform for image coding. In *Proc. SPIE vol 845: Visual Communication and Image Processing II.*, pages 50–58, Cambridge, MA.

Apostolopoulos, J. (1997). Personal Communication. Paper in preparation.

Chen, H., Civanlar, M., and Haskell, B. (1994). A block transform coder for arbitrary shaped image segments. In *International Conference on Image Processing*, volume 1, pages 85–89.

Jafarkhani, H. and Farvardin, N. (1997). Adaptive Image Coding Using Spectral Classification. *IEEE Transactions on Image Processing*. To appear.

Mallat, S. (1989). A theory for multiresolution dignsl decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:674–693.

Said, A. and Pearlman, W. A. (1996). A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 6(3):243–250.

Sezan, M. I. and H., S. (1982). Image Restoration by the Method of Convex Projections: Part 2 – Applications and Numerical Results. *IEEE Transactions on Medical Imaging*, MI-1(2):95–101.

Shapiro, J. M. (1993). Embedded image coding using zerotrees of wavelets coefficients. *IEEE Transactions on Signal Processing*, 41:3445–3462.

Simard, P. and Mailloux, G. E. (1988). A Projection Operator for the Restoration of Divergence–Free Vector Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(2):248–256.

Sweldens, W. (1996). The lifting scheme: A custom-design construction of biorthogonal wavelets. *Journal of Applied Computing and Harmonic Analysis*, 3:186–200.

Youla, D. C. and Webb, H. (1982). Image Restoration by the Method of Convex Projections: Part 1 – Theory. *IEEE Transactions on Medical Imaging*, MI-1(2):81–84.