IMAGE INTERPOLATION USING WAVELET-BASED EDGE ENHANCEMENT AND TEXTURE ANALYSIS

by

Sai-Hsueh Grace Chang

Memorandum No. UCB/ERL M95/100

12 December 1995

COVER PAGE

IMAGE INTERPOLATION USING WAVELET-BASED EDGE ENHANCEMENT AND TEXTURE ANALYSIS

by

Sai-Hsueh Grace Chang

Memorandum No. UCB/ERL M95/100

12 December 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Image Interpolation Using Wavelet-Based Edge Enhancement and Texture Analysis

by

Sai-Hsueh Grace Chang

Abstract

In many image processing applications, it is often important to accurately expand images without loss of clarity. Traditional methods such as bilinear and bicubic spline interpolation tend to smooth out edge regions and result in blurry images. In this work, we propose two methods for image interpolation: a multiresolution approach for enhancing isolated edges, and a texture analysis approach for interpolating non-isolated edges such as those found in a texture image.

Edges and textures are among the most important features of an image. They have, however, very different characteristics, suggesting that they should be enhanced using different techniques. For edges, we propose a novel and theoretically elegant approach which extrapolates the high resolution information needed to sharpen the image. This information is obtained by estimating the wavelet transform of the higher resolution based on the evolution of wavelet transform extrema across the scales. The motivation for this algorithm comes from the fact that wavelet transform modulus maxima capture the sharp variations of a signal. and that their evolution across the scales characterizes the local regularity of the signal. By identifying three constraints that the higher resolution component needs to obey, we enhance the reconstructed image through alternating projections onto the sets defined by these constraints. Results show that the enhanced image is superior to those interpolated by traditional methods.

For interpolating textures, we have mainly only a theoretical approach, while the practical implementation is being investigated. We propose to decompose texture fields into deterministic and purely indeterministic components, and interpolate the components separately. The deterministic component is extracted by detecting periodic components using a Fourier-based method, and its interpolation is done by estimating the trend of the periodic component and extrapolating to the higher frequencies. The indeterministic component is modelled as a ARMA process, which is interpolated by finding a higher resolution process whose subsampled version has similar statistical properties. Preliminary 1-D experiments show that this method may be a promising approach to pursue.

Acknowledgments

I would like to thank my advisor, Professor Martin Vetterli, for his supervision of this project and for being such an awesome advisor. And I would also like to thank "cool preppie dude" Zoran Cvetković for his helpful guidance on the technical stuff and inspiring comments about life.

I would like to thank Mom and Dad and my brother Henry for being supportive and loving. Many thanks to Claire for proofreading this thesis and for being a wonderful friend.

Lastly, I would like to express my gratitude to the National Science Foundation for providing financial support, so that I could be a lazy bum for three years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Given a digital image, it is often necessary to magnify the details for many image processing applications. An example of such applications is expanding digital satellite or medical images, where resolution is limited and, for the magnified details to be useful, it is essential to achieve expansion without blurring. Furthermore, if it is possible to interpolate a low resolution image in such a way as to regain the high resolution details of the original image, then we can compress the image by retaining only its low resolution component rather than the original high resolution image in its entirety, implying using less memory or fewer bits.

During the magnification process, it is important to avoid blurring the images so as to make the enlarged image useful for recognition. Standard methods such as bilinear and bicubic spline interpolation tend to smooth out the edges since they do not utilize any information relevant to preserving the edge sharpness. To deblur these processed images, one could use the commonly used unsharp masking [4] which boosts the high frequency component. Other methods include modelling the edges, filtering with nonlinear filters to boost the high frequency components [3], or using a MAP estimator [7].

Edges and textures are among the most important features of an image. They have, however, very different characteristics. In the intensity domain, edges are characterized by drastically changing intensity values. On the other hand, textures usually have a periodic constituent along with some random components. Their different behaviors suggest that interpolation of edges and textures need to be treated differently.

For edges or isolated singularities, we propose a wavelet-based method which estimates the higher resolution information needed to preserve the sharpness of the edges during the interpolation process. This information is obtained by extrapolating the wavelet transform of the higher resolution based on the evolution of wavelet transform extrema across the scales. The motivation for this algorithm comes from the fact that wavelet transform modulus maxima capture the sharp variations of a signal, and that their evolution across the scales characterizes the local regularity of the signal [5]. This allows a locally adaptive scheme which learns about the different singularities and enhances them accordingly. For implementation reasons, wavelet transform extrema rather than modulus maxima are used, but this change has mild effects. By identifying three constraints that the higher resolution component needs to obey, we improve the estimates through alternating projections onto the sets defined by these constraints. Results show that the enhanced image is superior to those interpolated by traditional methods.

The wavelet-based approach is appropriate for isolated singularities. Textures, on the other hand, have a very different flavor and need a different interpolation technique. In particular, a texture patch is typically very homogeneous over the entire patch such that a viewer does not see a particular feature which stands out. It can usually be characterized by ordering many similar looking primitive cells according to some placement rule which may or may not be deterministic. This feature strongly suggests that texture images be decomposed into a deterministic and a stochastic (or indeterministic) component; that is, analyze the texture field with a 2-D analogy of the Wold decomposition [2].

The deterministic component captures the periodicity that is typically encountered in a texture field, while the indeterministic component can be modelled by a 2-D ARMA random field. The extraction of these components is similar to the problem of estimating from a mixed spectra, which is a very difficult problem. For simplicity, a method similar to that in [2] is adopted for our purpose.

Once the components are estimated, they are each treated separately for the interpolation problem, and the results are combined to obtain the final interpolated signal. The interpolation problem can be viewed as estimating the missing samples, or equivalently, as extrapolating in the frequency domain beyond the available bandlimited signal. The higher frequency details of the deterministic component are estimated by assuming that its frequency response follows some trend which can be identified and extrapolated. The indeterministic component is extrapolated by finding a higher resolution random process which yields the same statistical property as the available low resolution random process when it is subsampled.

The wavelet-based enhancement algorithm for isolated singularities is explained in Chapter 2, along with some basics of the wavelet transform, and the result is shown in Section 4.1. The results show that this method does make improvements over conventional methods.

Chapter 3 presents the texture interpolation scheme. This part is on-going work and thus is mainly a theoretical approach to the problem. Section 4.2 shows some preliminary 1-D experiments which suggest that it may be a worthwhile approach to pursue.

Chapter 2

Enhancement Algorithm for Edges

Points of sharp variations, or singularities, are among the most meaningful features of a signal. For images, these points typically correspond to edges, or boundaries between regions, and it is important to detect these contours for many image enhancement applications. In the case of image interpolation, it is often desirable to magnify an image while preserving the sharpness of the edges so that the resulting image is pleasing to the eye and that recognition is not hindered. Linear methods typically generate blurred images because they do not use information pertaining to retaining the sharpness of the edges, but rather smooth them out. To obtain a good quality image expansion, we must utilize a method which adapts to preserve the sharpness of local singularities.

In this work, we propose to use the extrema points of the wavelet transform to detect and classify sharp variation points. The motivation for using a wavelet-based method is that singularities can be captured by the modulus maxima or zero-crossing points in the wavelet transform domain, and it provides a multiscale decomposition of the signal which allows us to characterize the types of singular points [5]. The basic idea behind our approach is to use this multiscale information to classify the type of singularity using the local Lipschitz regularity, and based on this classification estimate the higher resolution information necessary for preserving the edges. The estimated information is then refined by identifying convex constraints and using the POCS (projection onto convex sets) method.

2.1 Background and Motivation

We first give the motivation for using the wavelet transform extrema in continuous time. However, any practical processing has to be done in the discrete time domain, and the basic theory of the discrete wavelet transform is developed in 2.1.3. For a more in-depth discussion about wavelets, the reader is referred to [8]. The model for the evolution of wavelet transforms across the scales is introduced, and the reader is referred to [5] for a more detailed treatment.

2.1.1 Edge Detector and its Relation to Wavelet Transform

Most traditional edge detectors determine sharp variation points by examining the first or second derivatives of the signal. This is because inflection points indicate neighborhoods of signal variation, and inflection points in the signal domain correspond to the extrema points of the first derivative and to the zero-crossings of the second derivative of the signal. Furthermore, extrema points of large magnitude in the first derivative (that is, the maxima points of the absolute value of the first derivative, which is called *modulus maxima*) represent points of sharp variation, while those of small magnitude imply points of slow transition. In [5], reconstruction based on wavelet modulus maxima and wavelet zero-crossings yield very good results. However, the wavelet modulus maxima represention is not a convex set and the zero-crossings representation is more sensitive to noise. In this work, we use the wavelet transform extrema representation to extract the singularities because, due to its convexity, there exists a simple reconstruction algorithm [1].

Define a *smoothing function* $\theta(x)$ which satisfies

$$\lim_{x \to \pm\infty} \theta(x) = 0 \quad ,$$

and

$$\int_{-\infty}^{\infty} \theta(x) dx = 1 \quad .$$

We assume that $\theta(x)$ is differentiable and we introduce a function $\psi(x)$ as the first derivative of $\theta(x)$:

$$\psi(x) = \frac{d\theta(x)}{dx} \quad .$$

We call a *wavelet* any function whose average is 0. Hence, $\theta(x)$ can be considered as a wavelet since

$$\int_{-\infty}^{\infty} \psi(x) dx = 0 \quad .$$

Now we define $\psi_s(x)$ the dilated version of the wavelet function

$$\psi_s(x) = \frac{1}{s}\psi(\frac{x}{s}) ,$$

and we refer to s as the *scale*. To take a wavelet transform of a signal, we convolve it with the wavelet function. The wavelet transform of $f(x)$ at scale s and position x is denoted by $W_sf(x)$, where

$$W_sf(x) = f(x) * \psi_s(x) ,$$

and $*$ is the convolution operator. From the linearity of convolution and differentiation, it is easy to verify that

$$W_sf(x) = f(x) * (s\frac{d\theta_s(x)}{dx}) = s\frac{d}{dx}(f * \theta_s)(x) , \tag{2.1}$$

where $\theta_s(x)$ is defined similarly as $\psi_s(x)$. In words, equation (2.1) says that taking the wavelet transform of the signal at scale s and at position x is equivalent to taking the first derivative of the smoothed signal. Hence, if we detect an extremum in the wavelet transform, we have found an inflection point in $f * \theta_s$. For an extremum of large magnitude, or a maximum of $|W_sf(x)|$, it has the physical meaning of being in a region of sharp transition in the signal domain, while for an extremum of small magnitude, it indicates a region of slow transition. When the scale s is small, the smoothing function is very much localized in time, and hence has almost no smoothing, and the detection provides the points of very sharp variations in $f(x)$. On the other hand, when s is large, the smoothing function has a large spatial support, and the edge detector yields information of variations on a more global scale.

From the above discussion we see that the wavelet transform modulus maxima of a signal allows us to detect the occurrence of singular points. Furthermore, Section 2.1.2 discusses how the evolution of the modulus maxima values corresponding to a singularity across the scales lets us classify the type of singularity [5]. We propose an enhancement algorithm in Section 2.2 for image expansion that preserves this property and thus preserves the image clarity.

The wavelet transform modulus maxima is not a convex representation and presents difficulty in reconstruction. In this work, we adopt the use of the wavelet transform extrema, which is a convex representation and allows simpler reconstruction [1]. Since most wavelet transform extrema points are also wavelet transform modulus maxima, this variation introduces minimal effect on the results. Henceforth we will use the terms wavelet transform extrema and wavelet transform modulus maxima interchangeably.

2.1.2 Motivation for Extrapolating Information from Low Resolution Component

From the previous discussion, we see that a sharp variation point induces modulus maxima in the wavelet transform. Figure 2.1 shows the wavelet transform of a waveform consisting of a step, an impulse, and their smoothed versions. Each isolated singularity produces extrema points which propagate across the scales. The way they propagate across the scales can be characterized by the local Lipschitz regularity, which will be explained in this section. Hence if we can determine the Lipschitz regularity, this allows us to estimate the extrema at any scale. The basis for our enhancement method is to use this characterization to estimate the information at fine scales to preserve the sharpness of edges in the interpolation process.

Before we state how the evolution is characterized across the scales, let us first define what a Lipschitz regularity is. Let $0 \leq \alpha \leq 1$. A function $f(x)$ is uniformly Lipschitz α over an interval (a, b) if and only if there exists a constant K such that for any $x_0, x_1 \in (a, b)$

$$|f(x_0) - f(x_1)| \leq K|x_0 - x_1|^\alpha \ .$$

The uniform Lipschitz regularity (or exponent) of $f(x)$ refers to the supremum α_0 over all α for which $f(x)$ is uniform Lipschitz α.

The value of the uniform Lipschitz regularity measures how differentiable and how smooth the function is in a local neighborhood. For example, if $f(x)$ is differentiable at x_0, then it is of Lipschitz regularity 1. The larger the α, the more regular the function is. If $f(x)$ is discontinuous but bounded in the neighborhood of x_0, then $\alpha_0 = 0$. The following result states that the Lipschitz regularity can be measured from the evolution of the absolute values of the wavelet transform across the scales [5]. For convenience, the regularity is denoted by α instead of α_0.

Fact 1 *A function $f(x)$ is uniformly Lipschitz α over (a, b) if and only if there exists a constant $K > 0$ such that for all $x \in (a, b)$, the wavelet transform (on the dyadic scales) satisfies*

$$|W_{2^j} f(x)| \leq K(2^j)^\alpha \ . \tag{2.2}$$

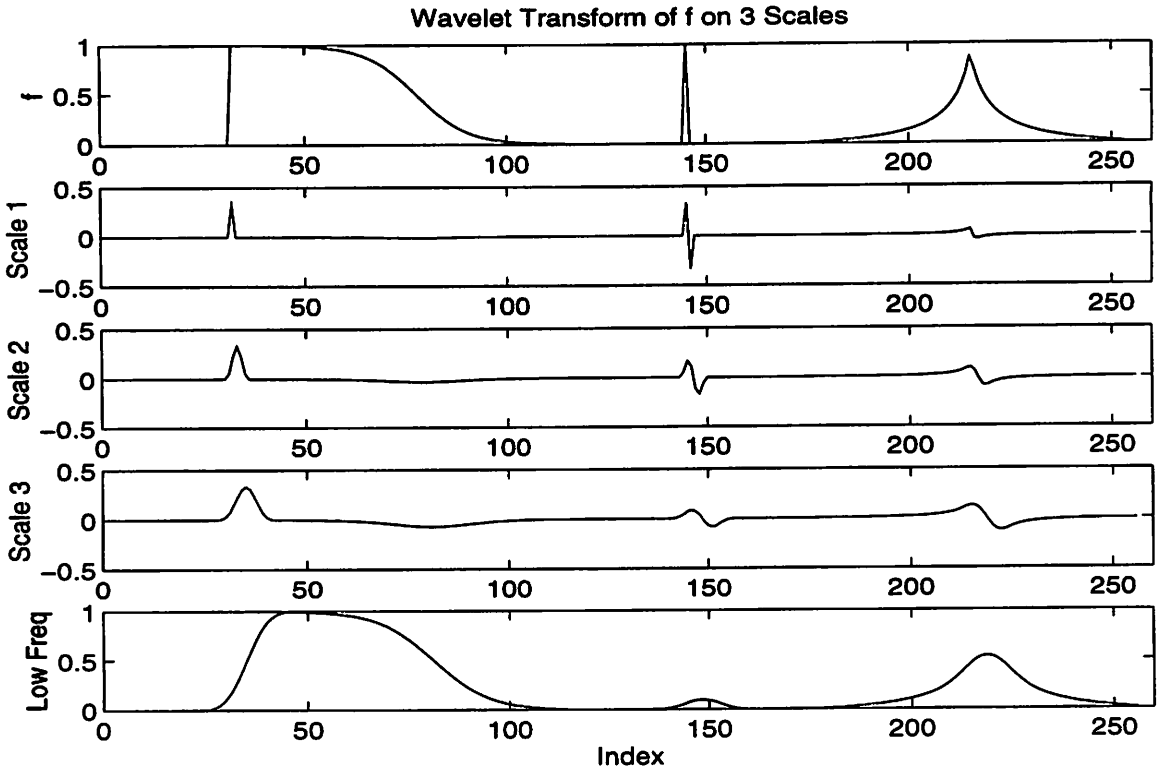


Figure 2.1: A synthetic waveform and its wavelet transform, showing the propagation of extrema points across the scales.

Although the above statement is for Lipschitz regularity α being $0 \leq \alpha \leq 1$, it can be extended to tempered distributions such as Dirac functions, which have negative Lipschitz exponents (in the case of a Dirac, $\alpha = -1$). Hence in this work, we do not constrain α to be in $[0, 1]$, but instead allow the range to be $[-2, 2]$ (the rationale being that the wavelet transform values should not differ by too much between adjacent scales). Given the constants K and α, equation (2.2) allows us to estimate the wavelet transform magnitude at any scale, and we show in Section 2.2 how this information is used for resolution enhancement.

2.1.3 Basics of Wavelet Transform

The framework of the wavelet transform modulus maxima and the zero-crossings representations stems from continuous-time theory. However, any practical implementation must be in discrete-time and there is a limit on the obtainable resolution. In this section, we introduce the 1-D and 2-D wavelet transform in discrete-time, which are among the fundamental tools used in this thesis. The wavelet transform has a filter bank interpretation, and,

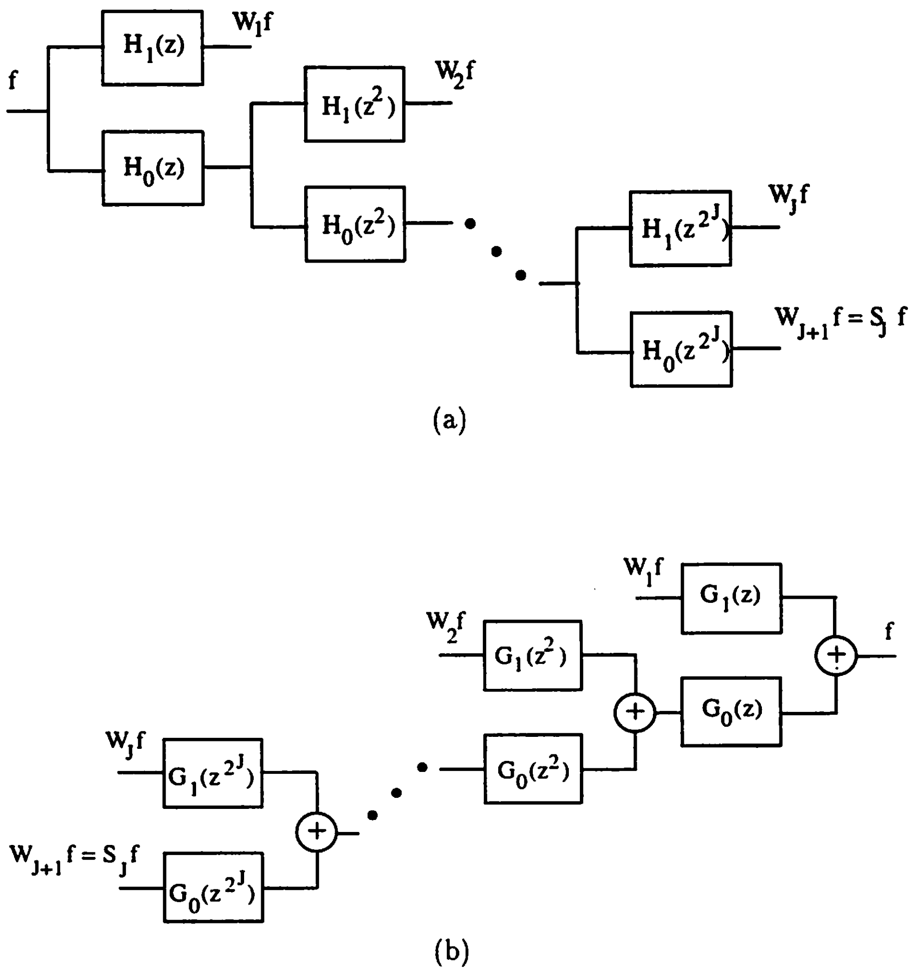


Figure 2.2: The 1-D wavelet transform implemented as a nonsubsampled filter bank. The wavelet transform is shown in (a) and the inverse wavelet transform in shown in (b).

as will be shown later, this interpretation is essential to the modelling and understanding of the interpolation problem. The 1-D and 2-D wavelet transforms are introduced, along with some of their properties. Note that since we will be using the dyadic scales $s = 2^j$, $j \in \mathbb{Z}$, the notation for the wavelet transform $W_{2^j} f$ at scale $s = 2^j$ will be changed to $W_j f$ for convenience.

One-Dimensional Wavelet Transform

The one-dimensional discrete-time wavelet transform is characterized by two analysis filters: a lowpass filter H_0 and a highpass filter H_1. Let $H_0^{(j)}$ and $H_1^{(j)}$ be the filters obtained by inserting $2^j - 1$ zeros between the coefficients of H_0 and H_1, respectively. The wavelet transform of a signal $f \in l^2(\mathbb{Z})$ can be computed through convolution with $H_0^{(j)}$ and $H_1^{(j)}$ in the following recursive manner [5]:

$$\begin{aligned} W_j f &= \frac{1}{\lambda_{j-1}} S_{j-1} f * H_1^{(j-1)} \\ S_j f &= S_{j-1} f * H_0^{(j-1)} \ , \quad j = 1, 2, \ldots, J, \end{aligned} \tag{2.3}$$

where $S_0 f = f$, $H_0^{(0)} = H_0$, $H_1^{(0)} = H_1$, and the smallest scale has been normalized to 1. (Note that the scaling constant λ_j is not usually present. It appears here because discretization introduces deviation to the estimation of the Lipschitz regularity, and scaling factors are needed to make the correction. See Section 2.2.2 for more details.) The wavelet transform consists of the set of sequences $\{S_J f, W_j f\}_{1 \le j \le J}$. For a fixed J, we use $W_{J+1} f = S_J f$ to simplify the notation. Let the wavelet transform operator $\boldsymbol{W}$ denote the linear operator mapping f to $\{W_j f\}_{j=1,\ldots,J+1}$. The operator $\boldsymbol{W}$ can be implemented by the octave band nonsubsampled filter bank shown in Figure 2.2(a).

In practice, there are only a finite number N of non-zero samples in a signal f. To mitigate this border problem, the signal is extended symmetrically as in the case of cosine transform; that is, we extend the signal $f[n], 0 \le n \le N-1$, to length $2N$ by "flipping" it: $f[n] = f[2N-1-n]$, for $N \le n < 2N$. This periodization avoids creating a discontinuity at the boundaries and the wavelet transform coefficients are also periodic $2N$.

For perfect reconstruction to be possible, it is necessary and sufficient that there exists a synthesis pair $G_0(z)$ and $G_1(z)$ which satisfy the perfect reconstruction condition

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 1 \ . \tag{2.4}$$

The inverse wavelet transform reconstructs the original signal by progressively adding finer and finer details onto the coarse residual signal $S_J f$. It can be calculated recursively:

$$S_{j-1} f = \lambda_j W_j f * G_1^{(j-1)} + S_j f * G_0^{(j-1)} \quad , j = J, J-1, \ldots, 1 \ , \tag{2.5}$$

where $G_0^{(0)} = G_0$, $G_1^{(0)} = G_1$. The inverse wavelet transform operator $\boldsymbol{W}^{-1}$ can be implemented as in Figure 2.2(b), which is called a nonsubsampled synthesis octave band filter bank. (Note again that the λ_j constants are needed to offset the scaling in the wavelet transform equation (2.4).)

The wavelet transform is an overcomplete, or redundant, representation of a function. For any set of sequences $\{g_j\}_{j=1,\ldots,J+1}$, it is not necessarily the wavelet transform of some function f in $l^2(\mathbb{Z})$. It is the wavelet transform of some function $f \in l^2(\mathbb{Z})$ if and only if

$$W(W^{-1}(\{g_j\}_{j=1,\dots,J+1})) = \{g_j\}_{j=1,\dots,J+1} \ . \tag{2.6}$$

If the set of sequences $\{g_j\}_{j=1,\dots,J+1}$ satisfies (2.6), then we say that $\{g_j\}_{j=1,\dots,J+1}$ belong to the range of the wavelet transform operator W. The operator WW^{-1} is the projection operator onto the range of the wavelet transform.

Two-Dimensional Wavelet Transform

A particular class of 2-D wavelets is used here. In particular, we choose separable filters for the 2-D wavelets, where the 1-D filters H_0, H_1, G_0, and G_1 are the same as in the case of the 1-D wavelet transform. We need an additional filter L whose Fourier transform satisfies

$$L(\omega) = \frac{1 + H_0(\omega)H_1(\omega)}{2} \ .$$

The 2-D wavelet transform and inverse wavelet transform can be computed in a recursive manner similar to the 1-D case [5], and they can be implemented with the filter banks shown in Figure 2.3.

2.2 Enhancement Algorithm

2.2.1 Overview

We first develop the problem in one dimension, and then extend it to the two-dimensional case. The discussion concentrates on magnification by a factor of 2 for simplicity, although larger magnification can be achieved through iteratively performing the algorithm.

The interpolation problem is modelled as in Figure 2.4. We model the available signal f as being obtained from the high resolution signal f_0 which we wish to recover, by lowpass filtering followed by downsampling by a factor of 2. Denote by $H_0(z)$ a lowpass filter, and by $H_1(z)$ a highpass filter such that the two filters, together with a synthesis pair $G_0(z)$ and $G_1(z)$, constitute a perfect reconstruction nonsubsampled filter bank. That is, the filters satisfy the perfect reconstruction condition (2.4). Note that the filter bank in the model is arbitrary, but we conjecture that as long as it is reasonable (i.e. a good lowpass/highpass pair of filters), the result of our algorithm will not depend strongly on the choice of filters.

In order to perfectly reconstruct the high resolution signal, we need to know both its highpass component g_s and its lowpass component f_s. However, only f, the downsampled

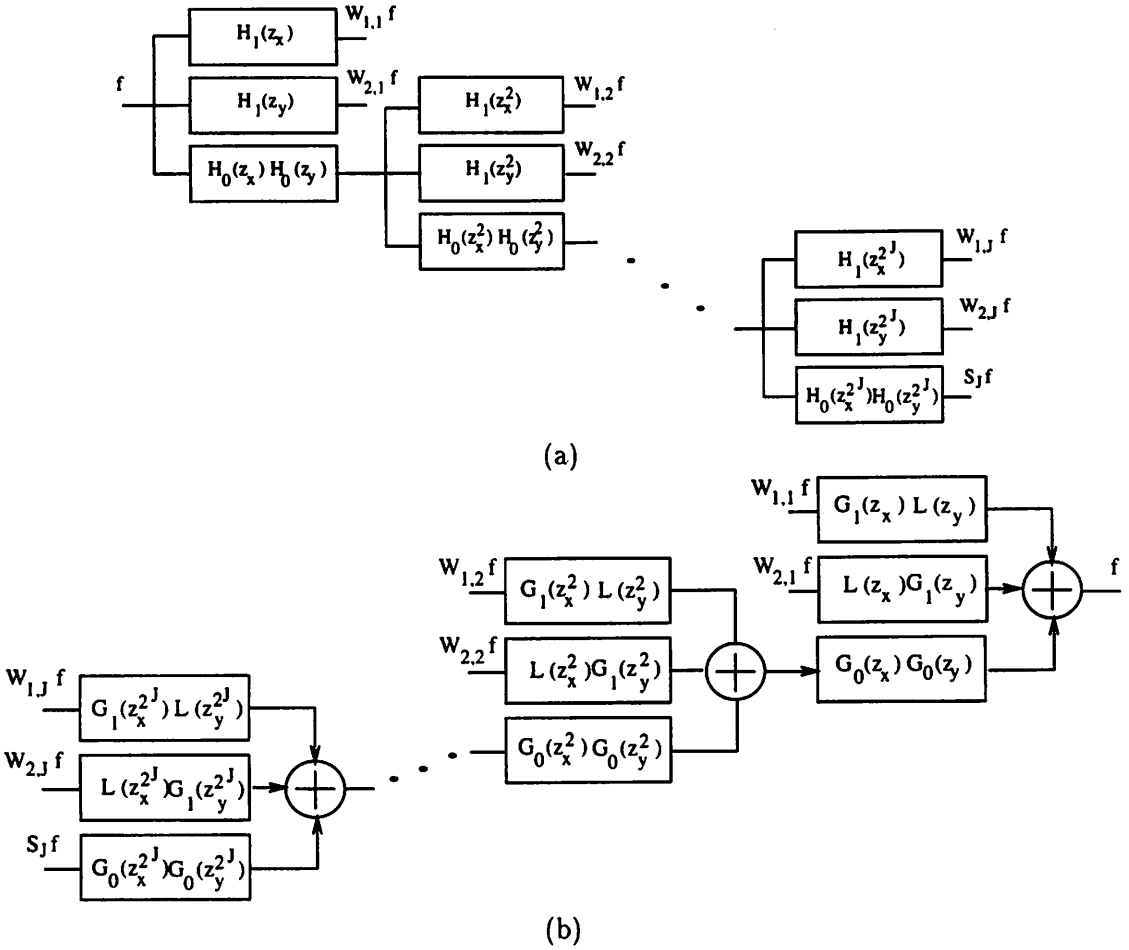


Figure 2.3: The 2-D discrete wavelet transform. (a) The wavelet transform. (b) The inverse wavelet transform.

version of f_s, is available. A standard approach to obtaining the high resolution signal f_0 would be to interpolate f using, for example, linear or spline interpolation, and possibly followed by some enhancement algorithm such as highpass filtering to deblur the result. The enhancement algorithm presented here is based on estimating the high frequency component g_s which is then combined with an estimate of f_s, through the synthesis filter bank, to give a reconstructed version of the high resolution signal.

An initial estimate $\hat{f}_s$ of the low frequency component f_s can be obtained by simply interpolating f, for instance, using linear or spline interpolation. The first step in estimating g_s is to find its local extrema by analyzing the available data f. It is based on the fact

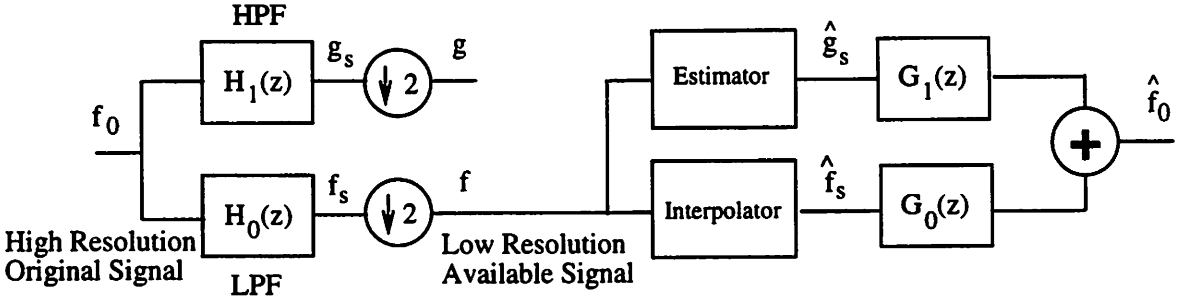


Figure 2.4: Problem Model

that local extrema of the wavelet transform propagate across the scales according to (2.2), which can be used to extrapolate the wavelet transform extrema of finer scales. That is, by modelling the evolution of the wavelet transform extrema of f across the scales using (2.2) and estimating the parameters, we can use this equation to estimate the corresponding extrema points of g. Note that the wavelet transform of f is the decimated version, by a factor of 2, of the wavelet transform of f_0 starting from the second scale (Figure 2.5). (Remark: This property is not exactly true in the 2-D wavelet transform because the separable filtering processes the signal in both the horizontal and vertical orientations. Therefore in the estimation process it is necessary to allow some error tolerances.) Hence, by examining the extrema of the wavelet transform of f, $\{W_j f\}_{j=1,\ldots,J+1}$, the extrema points of $g_s = W_1 f_0$ can be estimated. Since the wavelet transform of f, $\{W_j f\}_{j=1,\ldots,J+1}$, is obtained from subsampling the wavelet transform of f_0, $\{W_j f_0\}_{j=2,\ldots,J+2}$, there is some ambiguity in the estimation process introduced by the the downsampling operation. More specifically, the true extrema points of $\{W_j f_0\}_{j=2,\ldots,J+2}$ may not have been sampled to obtain $\{W_j f\}_{j=1,\ldots,J+1}$, hence there is ambiguity to the extrema values and locations (that is, the true extrema may be on the immediate left or right of the sampled extrema). In the implementation section, we will discuss some constraints which allow possible corrections to this ambiguity. Once the extrema points of g_s have been estimated, linear interpolation is used to obtain an initial estimate of the in-between points.

Signal enhancement is achieved by recognizing that the initial estimates of f_s and g_s can be further improved by identifying constraints that they should obey. Furthermore, these constraints define convex sets and we can utilize the POCS (projection onto convex sets) method to find a solution existing in the intersection of these sets, called the *reconstruction*

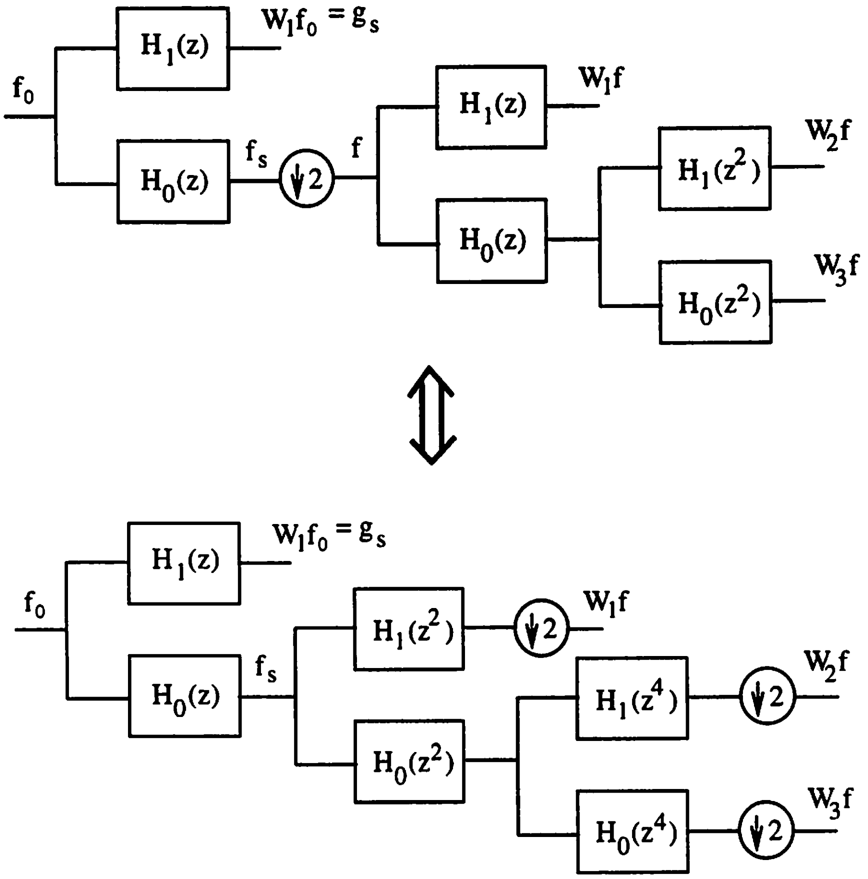


Figure 2.5: Showing the equivalence between the wavelet transform of f and the decimated version of the wavelet transform of f_0 starting from the second scale.

set. The POCS method alternatingly projects the signal onto the various convex sets until it converges to a solution in the reconstruction set (provided that it is nonempty). Figure 2.6 shows pictorially the idea of alternating projection. Any solution in the reconstruction set is called a *consistent reconstruction* and it satisfies all the imposed constraints.

The enhancement algorithm alternatingly projects the signal to satisfy three basic constraints:

1. The waveforms $\{\hat{f}_s, \hat{g}_s\}$ must be in the subspace V of $l^2(\mathbb{Z}^2)$, where V denotes the range of the wavelet transform.

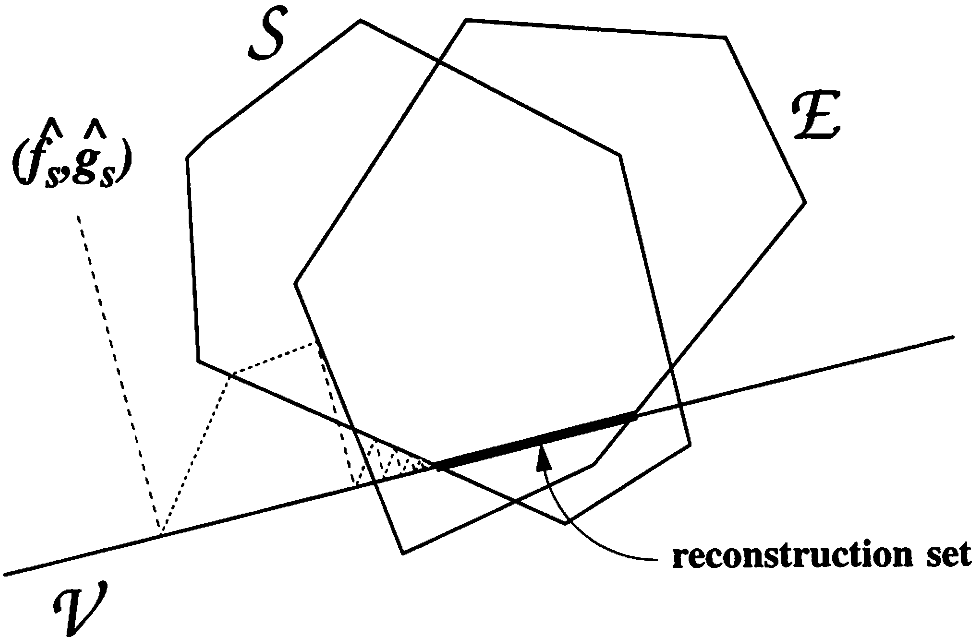


Figure 2.6: Projection onto convex sets. A signal is alternately projected onto the various sets until it converges to a point within the intersection.

2. The downsampled version of $\hat{f}_s$ must be equal to f, which is the original signal that is available.

3. The local extrema of $\hat{g}_s$ should reflect sharp variations in f_0, i.e. their values and locations are determined by singularities in f_0.

Let S be the set consisting of signals $g \in l^2(\mathbb{Z}^2)$ which satisfy the second constraint, i.e. $g[2n] = f[n]$, and let E be the set consisting of signals in $l^2(\mathbb{Z}^2)$ which satisfy the third constraint (To avoid obscuring the broad overview of the algorithm, the exact structure of a signal in E will be discussed in 2.2.2.) The first two items are hard constraints in that they follow from the way the problem is modelled in Figure 2.4, while the third constraint comes from our estimation of how the signal should be at fine scales. Furthermore, the set E should be defined in a way such that projecting $\hat{g}_s$ onto E improves the clarity of the reconstructed signal.

To speak of projections, it is more convenient to define the projection operator onto each convex set. The projection operator of the subspace V, denoted by P_V, filters the pair $(\hat{f}_s, \hat{g}_s)$ by the synthesis filter bank, followed by the analysis filter bank, as shown in Figure 2.7. For this work, we assume that the analysis and synthesis filter banks are perfectly

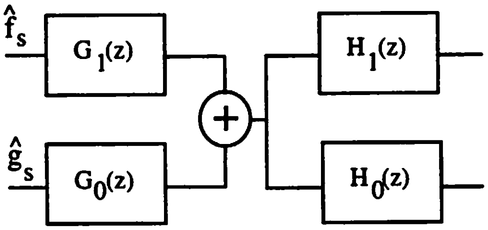


Figure 2.7: The projection operator onto the subspace V, the range of the wavelet transform.

reconstructing. The synthesis filters used in the implementation should be selected as

$$G_0(z) = \frac{H_0(z^{-1})}{P(z)} , \quad G_1(z) = \frac{H_1(z^{-1})}{P(z)} ,$$

where

$$P(z) = H_0(z)H_0(z^{-1}) + H_1(z)H_1(z^{-1}) .$$

In the case of power complementary analysis filters,

$$P(z) = H_0(z)H_0(z^{-1}) + H_1(z)H_1(z^{-1}) = 1 ,$$

which reduces to

$$G_0(z) = H_0(z^{-1}) , \quad G_1(z) = H_1(z^{-1}) .$$

For a detailed analysis of the projection operator onto V for both 1-D and 2-D signals, the reader is referred to [1].

The projection operator P_S onto convex set S simply assigns values of f to the even samples of $\hat{f}_s$.

The subspace V and the convex set S are well-defined, but the convex set E depends on our knowledge of the singularities of f_0. In this case, we only have estimated knowledge and thus we must allow some error tolerances. In Section 2.2.2, we discuss constraining the set E with varying degrees of leniency on the values and locations of the wavelet transform extrema, and finding a corresponding projection of $\hat{g}_s$ onto E.

Let $\{\hat{f}_s^{(0)}, \hat{g}_s^{(0)}\}$ be the initial estimates of $\hat{f}_s$ and $\hat{g}_s$. The alternating projection consists of iteratively operating on $\{\hat{f}_s^{(k)}, \hat{g}_s^{(k)}\}$ with $P_E \circ P_S \circ P_V$, so that at the end of the $(k+1)$-th iteration, the estimates of $\hat{f}_s$ and $\hat{g}_s$ are

$$\{\hat{f}_s^{(k+1)}, \hat{g}_s^{(k+1)}\} = P_E(P_S(P_V(\{\hat{f}_s^{(k)}, \hat{g}_s^{(k)}\}))) .$$

2.2.2 Implementation Details

Associating Extrema Across the Scales

For the extrapolation scheme, we need to first select important singularities and associate the corresponding extrema across the scales. Since the highest scale contains an abundance of extrema and is more sensitive to noise, selection of the extrema is done at the second scale. Due to discretization, the estimated Lipschitz regularity α disagrees from what it should be for a continuous time signal. Hence, scaling constants are multiplied to each scale of the wavelet transform, $W_j f$, and these constants are found empirically so as to make the discrete time step function have $\alpha = 0$.

Since we do not compute the wavelet transform on a dense scale but rather on dyadic scales, only some extrema propagate from scale j to $j+1$. To determine which extrema propagate to the next scale, some ad hoc criteria are used. First, the extrema have to be of the same sign and must all be maxima (or minima). Secondly, it is reasonable to assume that the extrema values should not differ by too much from scale to scale (i.e. they are approximately of the same order of magnitude), so we constrain the regularity α to have a maximum absolute value of 2. If α exceeds 2, then we have probably made a mistake in associating the extrema points, and thus, to avoid a really bad estimate, we set α to 0. Furthermore, since the wavelet transform filters are causal, linear phase and FIR, the extrema corresponding to a singularity are shifted from scale to scale, where the shift can be determined from the filters. Hence, we look within a reasonable neighborhood in the next scale to search for the extremum point. Each extremum point is only associated once.

Estimating Lipschitz Parameters

Let us first rewrite (2.2) in discrete-time and explicitly show the dependence of the local Lipschitz parameters on the different singularities. This results in

$$W_j f[x_n^{(j)}] = K_n (2^j)^{\alpha_n} \ , j = 1, \ldots, J \ , \tag{2.7}$$

where $W_j f$ is the wavelet transform of the input signal f at scale j, $x_n^{(j)}$ is the location of the local extremum at scale j corresponding to the nth singularity, α_n is the Lipschitz regularity of f at the singular point, and K_n is a nonzero constant.

Once we have the sequence of extrema $W_j f(x_n^{(j)}), j = 1, \ldots, J$ (corresponding to the nth local singularity) across the scales, we can estimate the parameters α_n and K_n from

(2.7). An alternative method is chosen over the minimization of squared error because the LSE method is too costly computationally. A simple way to find these parameters is to utilize extrema values from the three finest available scales, $j = 1, 2, 3$. Experiments show that it is valid to use only these scales, since the waveforms of larger scales are too smooth and do not retain many extrema points, and are thus useless for our purpose. The Lipschitz regularity α_n can be estimated by rearranging (2.7) as

$$\alpha_n = \log_2 \left(\frac{W_{j+1} f[x_n^{(j+1)}]}{W_j f[x_n^{(j)}]} \right) .$$

We first obtain two initial estimates of α_n by

$$\alpha_n^{(1)} = \log_2 \left(\frac{W_2 f[x_n^{(2)}]}{W_1 f[x_n^{(1)}]} \right) , \quad \alpha_n^{(2)} = \log_2 \left(\frac{W_3 f[x_n^{(3)}]}{W_2 f[x_n^{(2)}]} \right) ,$$

and then obtain the final estimate of α_n by

$$\alpha_n = \frac{\alpha_n^{(1)}}{\alpha_n^{(2)}} \alpha_n^{(1)} .$$

If the extrema only propagates across scales $j = 1, 2$, then our estimated regularity is $\alpha_n = \alpha_n^{(1)}$. The constant K_n is computed from

$$K_n = \frac{W_1 f[x_n^{(1)}]}{2^{\alpha_n}} .$$

After obtaining the estimates α_n and K_n, we can compute an initial estimate of the extrema points of the wavelet transform at scale 0 (that is, the extrema points of the finer scale g_s), which is given by

$$W_0 f[x_n^{(0)}] = K_n .$$

For the filters used in this work, the shift from scale 1 to 2 is about 1-2 positions, we thus assume that the shift from scale 0 to 1 is negligible. Hence, the estimated extrema values at scale 0 are assumed to be at the same location as the extrema values at scale 1; i.e. $x_n^{(0)} = x_n^{(1)}$.

Those extrema of the first scale that do not propagate to the coarser scales are simply assigned to the extrapolated scale using the same values and locations for the initial estimate of $\hat{g}_s$.

The parameters are allowed to take values within a confidence interval because the estimates may not be exact due to several factors. In particular, errors may be introduced if the algorithm identifies incorrectly the propagation of extrema. Also, discretization introduces errors: recall that the discrete dyadic wavelet transforms are only samples of the continuous wavelet transform, and thus the true extrema may not have been sampled. Another ambiguity introduced by subsampling comes from the fact that we are using the subsampled version of the wavelet transform to estimate the original wavelet transform by examining the extrema values (see Figure 2.5 and its associated discussion). Again, if the extrema are not sampled then the estimated finer scale extrema is not correct. Furthermore, there are more errors introduced in the 2-D application when it is treated as separable 1-D problems. This last point will be discussed later.

A simple ad hoc method for determining confidence intervals is used. For each extremum that is obtained through estimating (2.7), the upper bound is found by locating the maximum ratio between the extrema values of adjacent scales, and then extrapolating the maximum value to the 0th scale using this ratio for all scales. More formally, let

$$\gamma_n = \max_{1 \leq j \leq J-1} \frac{W_j f[x_n^{(j)}]}{W_{j+1} f[x_n^{(j+1)}]},$$

$$j_{max} = \arg \max_{1 \leq j \leq J-1} \frac{W_j f[x_n^{(j)}]}{W_{j+1} f[x_n^{(j+1)}]},$$

then the upper bound is obtained as $W_{j_{max}} f[x_n^{(j_{max})}] \cdot \gamma_n^{j_{max}}$. The lower bound is obtained in a similar way by replacing max by min.

Projection Operator onto Convex Set E

One of the constraints on the estimated waveform $\hat{g}_s$ is that its local extrema should reflect sharp variations in f_0. From f, we have some knowledge of what the extrema values and positions of $\hat{g}_s$ should be. Hence, the set E can be thought of as the set of waveforms minimizing a specified cost function which penalizes when the extrema values do not conform to this knowledge. In terms of the extrema values, there are various cost functions that could be used. We can either (a) constrain $\hat{g}_s$ to retain the initial extrema estimates throughout the reconstruction, (b) allow the values to be within an interval (the rationale being that the estimates of the extrema values may not be very reliable), or (c) have no constraints at all on the values. Approaches (a) and (c) are extreme cases, assigning either infinite cost for wrong values or no cost at all. The allowed interval of approach (b) serves as a moderation, and yields better results.

Since $\hat{g}_s$ is interpolated from the estimate of the subsampled waveform g, the sampling may be such that we miss the true extrema and obtain instead the adjacent points. Thus for each extremum of $\hat{g}_s$, the points immediately next to it are also allowed to be extrema to account for this ambiguity. More specifically, if we initially determine $x_n^{(0)}$ to be an extremum location of $\hat{g}_s$ induced by the nth singularity, then after the projection of $P_S \circ P_V$, $x_n^{(0)}$ may not be an extremum point anymore. However, we allow one of $\{x_n^{(0)} - 1, x_n^{(0)}, x_n^{(0)} + 1\}$ to be an extremum point throughout the iterations. If there are one or more maximum (minimum) points among these three points, then the one with the greatest (least) value is taken to be the estimated maximum (minimum) point for the nth singularity. If there are no maximum (minimum) points among them, then we retain the initially estimated position to be the maximum (minimum) point. The extrema values are then taken to be the values at these points which we determined to be the extrema.

Points between two adjacent extrema points should have values in the interval bounded by the extrema values, and, furthermore, there should not be any more extrema among these in-between points. Hence, once the extrema locations and values of $\hat{g}_s$ are determined, the points in between neighboring extrema points are modified to obey monotonicity. The algorithm used in constraining the points to obey monotonicity is a slight modification of the algorithm described in [1], and it consists of three steps:

1. The value at each extremum location must be in its range of confidence interval. This simply amounts to clipping each extremum value to be within its corresponding allowable range.

2. Between adjacent extrema points at x_n and x_{n+1}, clip the in-between points to be between the interval $[\hat{g}_s[x_n], \hat{g}_s[x_{n+1}]]$.

3. The in-between points must obey monotonicity; that is, there should not be any extrema points between two neighboring extrema points x_n and x_{n+1}. We proceed from the earlier extremum point at x_n and increase the index until we encounter a point at index k which does not obey monotonicity. Then we assign both $\hat{g}_s[k]$ and $\hat{g}_s[k-1]$ to the value $(\hat{g}_s[k]+\hat{g}_s[k])/2$. However, this might destroy the monotonicity of earlier points in the interval $[x_n, x_{n+1}]$, thus we must look back to ensure monotonicity by doing the averaging iteratively (This part is the modification from [1]).

Additional 2-D Considerations

In general, analyzing a 2-D problem by treating the two coordinates independently is not an optimal approach. However, for computational feasibility, we propose here to treat the two coordinates separately. For the wavelet transform, the data is filtered by the separable 2-D filter bank using algorithms proposed by [5] as shown earlier in this chapter. The wavelet transform generates the row-processed components $\{W_{1,j}f\}_{j=1,\ldots,J}$, the column-processed components $\{W_{2,j}f\}_{j=1,\ldots,J}$, and the low frequency residual $S_J f$. Each row of the row component (and similarly for the columns of the column component) of the image is processed as in the 1-D case to estimate the scale 0 information. That is, for an $N \times N$ image, the n-th row (column) of $\{W_{1,j}f\}_{j=1,\ldots,J}$ ($\{W_{2,j}f\}_{j=1,\ldots,J}$) is used to estimate the n-th row (column) of the scale 0 row (column) component as in the 1-D case. The in-between lines are then filled in by linear interpolation (as an initial estimate). All the estimated rows and columns are then combined to form the initial estimates of the row and column components of the 0-th scale.

For the alternating projections of the reconstruction scheme, the projection operator P_V is simply a 2-D inverse wavelet transform followed by a 2-D wavelet transform. The operator P_S amounts to making the assignment $\hat{g}_s[2n_1, 2n_2] = f[n_1, n_2]$. The projection operator P_E acts on the N available rows of the row component and on the N available columns of the column component. However, the higher resolution row (column) component has $2N$ rows (columns), and we only have constraints for N even rows, not the odd rows. There is the choice of whether to impose any constraints for the odd rows. We will return to this issue later in the section and show that better results are generated if we impose some reasonable constraints on the odd rows.

For two-dimensional applications, we can obtain less noisy estimates by considering, say, several neighboring rows of the row component and taking a weighted average. In this work, the two immediately adjacent rows are used. For the i-th row in scale 1 of the row component, $W_{1,1}f$, of the wavelet transform, if for each extremum point, there is an extremum in close vicinity on the adjacent $i-1$ and $i+1$ -th rows which has the value of the same order of magnitude, then the estimated parameters of the corresponding point on the $2i$-th row at scale 0 (i.e. on the $2i$-th row of $W_{1,1}f_0$) is smoothed by taking the weighted average of the parameters from the rows $2(i-1), 2i, 2(i+1)$, using the weights $[0.25, 0.5, 0.25]$. If only one adjacent row has similar value extremum, then the weighting is

[0.75, 0.25]. Similar processing is done for the column components.

Now we return to the issue of imposing constraints on the odd-numbered lines. By exploiting the continuity of contours, we can reduce the jaggedness of the edges. Since an edge or contour of an image is usually continuous and traverses through several rows and columns, it induces extrema which also traverse several rows of the row components (or columns of the column components). Hence, for each odd-numbered lines, we give it a constraint that is similar to its two neighboring lines whose constraints follow from estimating the available low resolution image. In particular, if two adjacent even-numbered lines have extrema that are in close proximity (allowed to be within 2 positions) and similar in value, then they are probably generated by the same curve and thus we impose an extremum in the in-between line whose location and value are taken as the average of those of the two neighboring lines. Since within such a small neighborhood, the difference in pixels is very small, for the sake of simplicity, averaging is used rather than fitting a smoothed curve across several lines. Results show that imposing constraints on these lines result in images that are far less jagged than without constraints.

Chapter 3

Enhancement Algorithm for Textures

While the algorithm described in Chapter 2 is an elegant approach to extrapolating the high resolution information of isolated edges, the theory does not address the treatment of non-isolated edges. In terms of images, these non-isolated edges may occur in texture regions. Natural images usually contain various texture patches which, along with their boundaries, are essential features of the images. Therefore, it is important to understand and parameterize texture patches so as to achieve good quality image interpolation based on reconstruction from these parameters.

Let us examine another interpretation to the interpolation problem. Suppose we model the available low resolution signal as the signal obtained from ideal lowpass filtering a higher resolution signal followed by downsampling. Then, assuming that no information about the higher frequency component is available, the interpolation scheme that yields the least mean squared error is upsampling, followed by passing the resulting signal through a lowpass filter (see Figure 3.1). Essentially, the higher frequency component is assumed to be zero. However, it is more reasonable to believe that the original high resolution signal is non-zero in the upper half of the spectrum, and we could improve the resolution of the interpolated signal if we could estimate this higher frequency component.

Because of the homogeneous quality of a typical texture image, we conjecture that its Fourier transform or power spectrum obey some trend which can be identified. In this chapter, we propose a method which decomposes the texture field into deterministic and

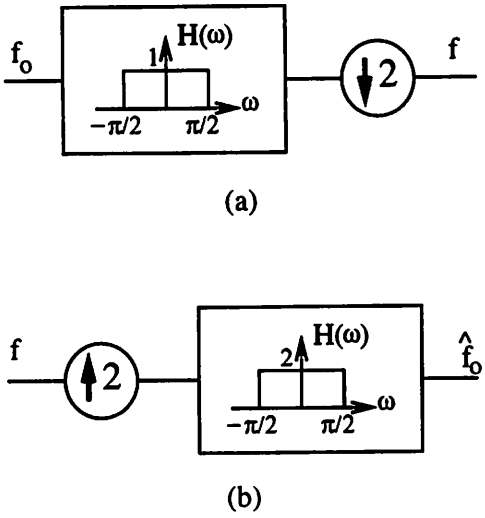


Figure 3.1: Problem Model. (a) Model the available low resolution signal f as the waveform obtained from lowpass filtering a higher resolution signal f_0 followed by downsampling. (b) Assuming that we have no information about the higher frequency component, the optimal interpolation scheme is upsampling followed by lowpass filtering.

indeterministic components, and extrapolates the high frequency content for each component. The interpolated image is then reconstructed from combining the available signal and the extrapolated information.

The material presented in this chapter is on-going work and is mainly a theoretical approach to the problem. Some preliminary results are shown, while more experimental results are presently being investigated.

3.1 Texture Model

A texture patch is usually composed of primitive cells which resemble each other, and which have some semi-regular ordering to their relative positions such that the overall structure looks uniform to a viewer. One extreme of a texture patch would be a purely deterministic pattern, such as a chess board, where all the primitive cells are exactly the same and the placement of the cells is done in a predictable way. At the other extreme is a purely stochastic pattern such as a scene of grass lawn, where the entire patch is organized by a structure usually modelled as a 2-D Gauss Markov random field. In the intermediate, the primitive cells are similar but not identical, and the placement of the cells is not defined by a completely predictable rule, but the overall patch still seems to follow some kind of ordering on a global scale.

This feature suggests that textures should be decomposed into a deterministic and an indeterministic component. The decomposition method and terminology presented here follow closely the work in [2]. The texture field is assumed to be a realization of a 2-D homogeneous (or stationary) random field. With this assumption, the texture field can be analyzed using a 2-D extension of the 1-D Wold decomposition for wide-sense stationary random processes, called a "2-D Wold-like decomposition" in [2]. The texture field is decomposed into two mutually orthogonal components, namely, the deterministic component, which makes up the underlying global superstructure, and the purely indeterministic component, which accounts for the unpredictable variations between the cells. The purely indeterministic component has a 2-D moving average representation, driven by a white innovation field. The deterministic component random field can be perfectly predicted from past samples. Note that the deterministic field is a random field and is deterministic only in the mean square sense. The deterministic part is further decomposed into two orthogonal components: the harmonic component and the generalized-evanescent component. Basically, the harmonic component consists of 2-D periodic elements of the global structure, while the generalized-evanescent component can be thought of as a set of plane waves, each traveling in a different direction.

3.2 Decomposition of the Homogeneous Random Field

In this section, we introduce the Wold-like decomposition along with several definitions. Here we merely state the relevant results and refer the reader to a more rigorous treatment in [2]. Without loss of generality, assume that the 2-D random field $\{f[n,m]\}_{n,m\in\mathbb{Z}}$ is real and zero-mean.

The random field $\{f[n,m]\}_{n,m\in\mathbb{Z}}$ is called a *homogeneous* random field if $E\{f^2[n,m]\} < \infty$ and $r_f[k,l] = E\{f[n+k,m+l]f[n,m]\}$, for all $(k,l) \in \mathbb{Z}^2$, is independent of n and m. Let $P(\omega,\nu)$ be the spectral distribution function of $\{f[n,m]\}$, defined on the square region $[-\frac{\pi}{2},\frac{\pi}{2}] \times [-\frac{\pi}{2},\frac{\pi}{2}]$. The autocovariance function is then found by

$$r_f[k,l] = \frac{1}{(2\pi)^2}\int_{-\pi/2}^{\pi/2}\int_{-\pi/2}^{\pi/2} e^{j(k\omega+l\nu)}dP(\omega,\nu) \ .$$

The corresponding spectral density function (or power spectrum), denoted by $p(\omega,\nu)$, is found by taking the Lebesgue derivative of $P(\omega,\nu)$.

Now consider the problem of finding the best "causal" linear estimator $\hat{f}[n,m]$ of $f[n,m]$ based on the "past" samples of $f[n,m]$. In 1-D, the notion of causality is obvious. In 2-D however, we need to define some kind of ordering. In this work, we choose the totally-ordered, nonsymmetrical-half-plane (NSHP) support, which is visualized as a scan from top-to-bottom, column-after-column from left to right. In mathematical terms, the ordering can be written as

$$(i,j) \prec (s,t) \Longleftrightarrow (i,j) \in \{(k,l)|k=s, l<t\} \cup \{(k,l)|k<s, -\infty<l<\infty\} . \qquad (3.1)$$

Let $\hat{f}[n,m]$ be the minimum-norm linear causal (with respect to the NSHP ordering) estimator of $f[n,m]$. Then the prediction error, $u[n,m] = f[n,m]-\hat{f}[n,m]$, is a wide-sense homogeneous random field, and we call $\{u[n,m]\}$ the *innovation* field. If $E\{u^2[n,m]\} = \sigma^2 > 0$ (that is, the innovation field does not vanish), then the field $\{f[n,m]\}$ is said to be *regular*. It is called *deterministic* if $\sigma^2 = 0$. Note that deterministic here only refers to being deterministic in the mean squared sense.

The 2-D Wold-like decomposition states that if $\{f[n,m]\}$ is a 2-D regular homogeneous random field, then it can be uniquely represented by the orthogonal decomposition

$$f[n,m] = f_d[n.m] + f_i[n,m] .$$

The field $f_d[n,m]$ is a deterministic random field. The most general form of $f_i[n,m]$ is the 2-D MA representation. Moreover, if the spectral density is strictly positive on the unit bicircle and analytic in some neighborhood (which is a mild assumption and will be used in this work), then it can also be represented as a 2-D AR process driven by the white noise innovation field $\{u[n,m]\}$

$$f_i[n,m] = \sum_{(0,0)\prec(k,l)} b[k,l]f_i[n-k,m-l] + u[n,m] \qquad (3.2)$$

where

$$\sum_{(0,0)\prec(k,l)} b^2[k,l] < \infty .$$

Since the random field $\{u[n,m]\}$ is a white noise field, the fields $\{f_d[n,m]\}$ and $\{u[k,l]\}$ are mutually orthogonal for all (n,m) and (k,l). A regular field is called *purely indeterministic* if the deterministic component $\{f_d[n,m]\}$ vanishes. Furthermore, the 2-D field $\{f_i[n,m]\}$ is regular and purely indeterministic.

The spectral distribution function $P(\omega, \nu)$ can then be written uniquely as

$$P(\omega, \nu) = P_d(\omega, \nu) + P_i(\omega, \nu) \quad ,$$

where $P_d(\omega, \nu)$ and $P_i(\omega, \nu)$ are respectively the spectral distribution function of $\{f_d[n, m]\}$ and $\{f_i[n, m]\}$.

Purely Indeterministic Component

From the 2-D AR representation of the purely indeterministic random field $\{f_i[n, m]\}$, the spectral distribution function $P_i(\omega, \nu)$ is absolutely continuous. From (3.2), $p_i(\omega, \nu)$ has the form

$$p_i(\omega, \nu) = \sigma^2 \left| 1 - \sum_{(0,0)\prec(k,l)} b[k, l]\, e^{j(k\omega + l\nu)} \right|^{-2} \quad .$$

Deterministic Component

The spectral distribution function $P_d(\omega, \nu)$ of the deterministic random field $\{f_d[n, m]\}$ is singular; that is, it consists of discontinuous jumps and its derivative, the spectral density $p_d(\omega, \nu)$, is nonzero only on a set of Lebesgue measure zero. The deterministic component $\{f_d[n, m]\}$ can be represented uniquely by the following orthogonal decomposition

$$f_d[n, m] = h[n, m] + g[n, m] \; .$$

The spectral density function of $\{f_d[n, m]\}$ consists of discrete points and continuous curves. For practical purposes, we can assume that there are only countably many points and curves.

The harmonic component $\{h[n, m]\}$ can be represented as a sum of sinusoids with random phase and random amplitude:

$$h[n, m] = \sum_k \{A_k \cos 2\pi(n\omega_k + m\nu_k) + B_k \sin 2\pi(n\omega_k + m\nu_k)\} \tag{3.3}$$

where A_k's and B_k's are mutually orthogonal random variables and $E\{A_k^2\} = E\{B_k^2\} = \sigma_k^2$. The autocorrelation function of $\{h[n, m]\}$ is

$$r_h[n, m] = \sum_k \sigma_k^2 \cos 2\pi(n\omega_k + m\nu_k) \; . \tag{3.4}$$

The harmonic component thus contributes discrete points in the power spectrum.

Before describing the generalized-evanescent component, we need to introduce alternative definitions to the NSHP order in (3.1). In particular, we introduce a family of total-order and NSHP-support definitions in which the boundary lines of the NSHP definitions are of rational slope. This is done by rotating the NSHP support in (3.1) and normalizing it so that the new samples fall on an integer grid. For more details, see [2]. Let O be the set of all possible total-order and NSHP with rational slope boundary lines, any support $o \in O$ is called rational nonsymmetrical half-plane (RNSHP). All the aforementioned results hold for any RNSHP support and the decomposition is unique and invariant to the support definition. Furthermore, although different RNSHP definitions result in different innovation fields, the variance of the innovation fields remains the same.

The generalized evanescent component $\{g[n,m]\}$ can be thought of as a sum of plane waves traveling in different directions. In particular, because $\{f[n,m]\}$ is sampled on a rectangular grid, we assume that the plane waves travel in directions orthogonal to lines of rational slopes. The generalized evanescent field $\{g[n,m]\}$ is a linear combination of a countable number of mutually orthogonal evanescent fields. A 2-D deterministic random field $\{e[n,m]\}$ is called *evanescent* with respect to a specific total-order and RNSHP if it spans a Hilbert space identical to the one spanned by the column-to-column innovations of the deterministic field $f_d[n]$ at each coordinate (n,m). The generalized evanescent field consists of a linear combination of evanescent field each defined with respect to a different total-order and RNSHP support $o \in O$, i.e.

$$g[n,m] = \sum_{o \in O} e_o[n,m] \; .$$

The spectral distribution of each evanescent field is separable in the ω and ν variables (which are also defined in their respective total-order and RNSHP support), and it is absolutely continuous in one dimension and singular in the orthogonal dimension. The spectral density function of each evanescent component is a linear combination of 1-D delta functions in the frequency domain:

$$p_e(\omega,\nu) = g(\omega) \sum_k \gamma_k^2 \{\delta(\nu - \nu_k) + \delta(\nu + \nu_k)\} \; .$$

In the space domain, the evanescent field is of the form

$$e[n,m] = s[n] \sum_k \{C_k \cos 2\pi m \nu_k + D_k \sin 2\pi m \nu_k\} \; , \tag{3.5}$$

where $\{s[n]\}$ is a purely indeterministic 1-D process with spectral density $2g(\omega)$, the coefficients C_k's and D_k's are mutually orthogonal random variables, and $E\{C_k^2\} = E\{D_k^2\} = \gamma_k^2$. This representation says that the generalized evanescent component $g[n,m]$ is a sum of waves traveling in directions orthogonal to lines of rational slopes. In the frequency domain, the spectral density function consists of a countable sum of 1-D delta functions which are supported on lines of rational slopes.

3.3 Extracting Parameters

Before extrapolating the high frequency component, we need to first estimate the parameters of the different components. The estimation problem is related to the 2-D mixed spectra estimation, which is a very difficult problem. In this work, we follow closely the simple estimation method proposed in [2], which claims that synthesized texture patches based on their method of parameter estimation are virtually indistinguishable from the original.

As suggested in the decomposition, the estimation is done in three stages, in the order of extracting the harmonic field, the evanescent field, and then the purely indeterministic random field.

3.3.1 Extracting the Harmonic Component

In practice, we only need to concern ourselves with a finite sum in (3.3) and in (3.4), thus we let the summation range from $k = 1$ to $k = M$, where M denotes the number of elements in the harmonic component. We assume that we only have one realization of the random field in the given texture patch. The estimation process needs to solve for three sets of unknowns: the number M of harmonic components, the values of the harmonic frequencies ω_k's and ν_k's, and a particular value for each realization of the A_k's and B_k's.

To simplify the problem, two assumptions are made. The first is that the unknown frequencies ω_k's and ν_k's are more than $1/N$ apart in each dimension (for a sample patch of $N \times N$). Secondly, the spectral density function of the purely indeterministic random field is sufficiently smooth, so that the harmonic component appears as isolated large sharp peaks in the spectral domain. With these assumptions, a DFT-based spectral estimator can be used, and, specifically, the periodogram is used. The periodogram for a $N \times N$ realization of a random field $\{y[n,m]\}$ is defined as

$$\begin{aligned}\hat{p}(\omega,\nu) &= \frac{1}{N^2}\left|\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} y[n,m]e^{-j(n\omega+m\nu)}\right|^2 \\ &= \frac{1}{N^2}\left|\mathrm{DTFT}\{y[n,m]\}\right|^2 .\end{aligned}$$

The sinusoids of $\{h[n,m]\}$ are detected by choosing the frequencies of the largest and sharpest isolated peaks of the periodogram.

The detection method initially sets the amplitude threshold value to the maximum value of the periodogram. Then the threshold is gradually lowered, while marking as a valid sinusoid the sharp peaks which are also maxima points. This procedure is done until the remaining peaks are too wide to be considered a sinusoid. The number of detected peaks is the estimate of the number of harmonic components. The coefficient values of the sinusoids are found by evaluating the DFT of the texture patch at the detected frequency location.

The spatial domain harmonic component is found by performing inverse DFT on the detected frequencies and complex amplitudes. The harmonic component is then subtracted from the texture patch, which should contain only the evanescent and purely indeterministic components.

3.3.2 Extracting the Evanescent Component

In practice, the spectral density function of the generalized evanescent field is composed of a finite sum of singular functions supported on straight lines of rational slopes in the 2-D frequency plane, implying that they are continuous on the line and singular in the orthogonal direction.

To detect the evanescent component, we detect the maxima of the periodogram which form approximately continuous lines in the frequency plane. In particular, we search for large peaks of approximately the same magnitude such that the peaks are along one dimension, while exhibiting fast decays in the orthogonal dimension. The evanescent component is then extracted by processing the signal through a filter which is unity at these frequencies, and zero otherwise. The magnitude and phase are obtained by evaluating the 2-D DFT at these frequencies. The evanescent component is then removed from the periodogram, which subsequently contains only the purely indeterministic component.

3.3.3 Extracting the Stochastic Component

After removing the deterministic component, the periodogram now contains only the purely indeterministic component. The parameters of the AR representation in (3.2) are estimated using a 2-D Levinson-type algorithm by Marzetta [6]. For 1-D signals, the parameters are estimated using the Burg method. To use these methods, one needs to select the model order. One way to select this is to start from low order and increase the order until the first few AR coefficients do not change much and the latter coefficients are relatively small in magnitude.

3.4 Interpolating the Various Components

3.4.1 Interpolating from the Harmonic Component

Once the sinusoids are detected, we can interpolate the harmonic component. We first proceed with 1-D interpolation. The interpolation problem is basically to find the missing samples between the available samples. In the frequency domain, this is equivalent to extending the frequency component of the upsampled signal beyond $\pi/2$. For example, a 1-D length-N periodic square wave function $f[n]$ with period N_0 has fundamental frequency $\omega_0 = 2\pi N_0/N$ and generates spikes at the harmonic frequencies $\omega_0, 3\omega_0, \ldots, (2m_1 + 1)\omega_0$, where $(2m_1+1)\omega_0$ is the largest frequency smaller than π. When we upsample $f[n]$ to length $2N$, the harmonic frequencies are shifted to $\omega_1, 3\omega_1, \ldots, (2m_1+1)\omega_1$ where $\omega_1 = 2\pi N_0/(2N)$. Furthermore, we expect to have spikes at $(2(m_1+1)+1)\omega_1, (2(m_1+2)+1)\omega_1, \ldots, (2m_2+1)\omega_1$, where $(2m_2 + 1)\omega_1$ is the largest frequency smaller than π (similarly with the negative frequencies).

In 1-D, the harmonic frequencies are spaced at periodic spacings, and thus one can predict where the high frequency harmonics should be. Without loss of generality, let's order the harmonic frequencies ω_k in increasing order. Let's denote the DFT value of the k-th harmonic as p_k, for $k = 1, \ldots, M$, and these frequencies correspond to the range in $[-\pi, \pi]$. To interpolate $f[n]$, we need to first upsample it, which renormalizes the harmonic frequencies $p_k, k = 1, \ldots, M$, to the range $[-\pi/2, \pi/2]$. The problem here is to extrapolate the values at the high frequency harmonics given the values at the low frequency harmonics $p_k, k = 1, \ldots, M$. We propose to model the magnitude of these values with exponential decay

$$p_k = ab^k, \; k = 1, \ldots, M, \tag{3.6}$$

and estimate the constants a and b to extrapolate p_k for $k = M+1, M+2, \ldots$. Note that other fitting methods could be used, but the exponential one is chosen for its simplicity.

This extrapolation scheme poses a problem, however, since the phase is not modelled. To avoid phase extrapolation, we extrapolate the DCT (discrete cosine transform) coefficients instead of the DFT coefficients. Pure sinusoids do not correspond to isolated peaks in the DCT domain. Empirically, we observe that for a periodic-N_0 signal, the higher frequency part of the DCT coefficients seems to be periodic-$4N/N_0$, with decreasing magnitudes. Hence the higher frequency part of the DCT coefficients are divided into blocks of $4N_0$ and their maximum values are modelled as in (3.6), and entire blocks are used to extrapolate to fill up the upper spectrum. The fitting of (3.6) could be done using the least mean square error criterion. For simplicity, a method similar to that in estimating the Lipschitz regularity in Section 2.2.2 is used. Preliminary results of this extrapolation method are shown in Chapter 4.

In 2-D, it is more complicated to determine the trend of the harmonic component. It is conjectured that the harmonic frequencies also have periodic spacings, and thus make predicting the high frequency component possible. The exact method is presently being considered.

3.4.2 Interpolating from the Evanescent Component

The evanescent component only applies to the 2-D case. The method of extrapolation is similar to that of the harmonic component, except with continuous lines rather than with discrete points. For example, for a grid-like texture (parallel lines to the horizontal and vertical axes), the periodogram has horizontal and vertical singular functions at periodic spacings, and this trend can be estimated and extrapolated.

3.4.3 Interpolating from the Indeterministic Component

The indeterministic component is modelled with AR parameters. Because our interpolation scheme for an AR process depends on factoring polynomials, we have only considered the 1-D case so far, since 2-D factorization is a difficult problem. The 2-D AR process interpolation remains an open question.

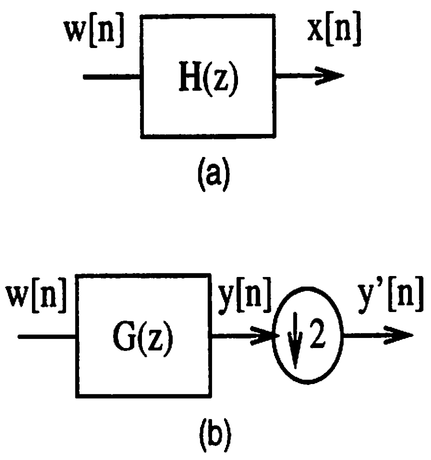


Figure 3.2: (a) An ARMA process $x[n]$ generated from filtering a white Gaussian noise process, $w[n]$, with canonical synthesis filter $H(z)$. (b) An ARMA process $y[n]$ with synthesis filter $G(z)$, which is designed such that $y'[n]$ has the same second-order statistical properties as $x[n]$.

Interpolating a 1-D ARMA Process

A purely stochastic process is often modelled as an ARMA process, and this is the framework that we use. Assume without loss of generality that all random processes mentioned here are zero-mean. The idea in interpolating an ARMA process is as follows. Suppose $x[n]$ is a real ARMA process. We wish to interpolate the "low resolution" $x[n]$ to obtain a "higher resolution" ARMA process $y[n]$ such that the subsampled version of $y[n]$, denoted as $y'[n]$, has the same first- and second-order statistical property as $x[n]$. More specifically, Figure 3.2(a) shows $x[n]$ with its canonical synthesis filter $H(z)$ and its generating white noise process $w[n]$ with power σ^2, and Figure 3.2(b) shows $y[n]$ with its canonical synthesis filter $G(z)$. Let $y'[n]$ be the subsampled version of $y[n]$, i.e. $y'[n] = y[2n]$. Given that we know $H(z)$, we wish to design $G(z)$ such that

$$r_{y'}[m] = E\{y'[n]y'[n-m]\} = E\{x[n]x[n-m]\} = r_x[m] \ ,$$

where $r_{y'}[m]$ and $r_x[m]$ denote respectively the autocorrelation function of $y'[n]$ and $x[n]$ (the stationarity of $y'[n]$ will be shown later). There are two questions to be answered. First, is such a process possible? And if yes, how do we find $G(z)$?

Before answering these questions, let's derive the relationship between the various signals.

$$\begin{aligned}
E\{y'[n]y'[n-m]\} &= E\{y[2n]y[2n-2m]\} \\
&= E\left\{\sum_k g[k]w[2n-k]\sum_l g[l]w[2n-2m-l]\right\} \\
&= \sum_k\sum_l g[k]g[l]\underbrace{E\{w[2n-k]w[2n-2m-l]\}}_{\delta[2m+l-k]} \qquad (3.7) \\
&= \sum_l g[2m+l]g[l] \\
&= r_y[2m] \qquad (3.8)
\end{aligned}$$

Note that the expression in (3.7) is simplified by the fact that $w[n]$ is a white noise process and hence $r_w[m] = E\{w[n]w[n-m]\} = \delta[m]$. Equation (3.8) shows that $y'[n]$ is a wide-sense stationary process and that $r_{y'}[m] = r_y[2m]$.

To design $G(z)$ such that $r_{y'}[m] = r_x[m]$, it is sometimes easier to equate them in the z-transform domain:

$$r_{y'}[m] = r_y[2m] = r_x[m] \overset{\mathcal{ZT}}{\longleftrightarrow} S'_y(z) = \frac{1}{2}(S_y(z^{\frac{1}{2}}) + S_y(-z^{\frac{1}{2}})) = S_x(z) \ . \qquad (3.9)$$

Now we are ready to answer the two questions. The answer to the first question is affirmative, since there is always the trivial answer. First note that the power spectrum of $x[n]$ and $y[n]$ are respectively

$$\begin{aligned}
S_x(z) &= \sigma^2 H(z)H(z^{-1}) \ , \\
S_y(z) &= \sigma^2 G(z)G(z^{-1}) \ .
\end{aligned}$$

Let $y[n]$ be generated from the canonical synthesis filter $G(z) = \sqrt{2}H(z^2)$, then it can be easily verified from (3.9) that the power spectrum of $y'[n]$ is

$$\begin{aligned}
S_{y'}(z) &= \frac{\sigma^2}{2}(G(z^{\frac{1}{2}})G(z^{-\frac{1}{2}}) + G(-z^{\frac{1}{2}})G(-z^{-\frac{1}{2}})) \\
&= \frac{\sigma^2}{2}(H(z)H(z^{-1}) + H(z)H(z^{-1})) \\
&= \sigma^2 H(z)H(z^{-1}) \\
&= S_x(z) \ .
\end{aligned}$$

However, this answer is not very interesting because this model gives no correlation between the odd and even samples. This counteracts the intuition of interpolation, which usually utilizes the correlation of the closest neighbors to estimate the missing samples. Hence, we wish to find a process $y[n]$ which has correlation between the odd and even samples.

First, consider interpolating the AR(1) process, where $H(z) = \frac{1}{1-az^{-1}}$. For this simple case, it is actually easier to set $r_y[2m] = r_x[m]$ to find the parameters. It is easy to verify that for the AR(1) process, the autocorrelation function is

$$r_x[m] = \sigma^2 \frac{a^{|m|}}{1-a^2} . \tag{3.10}$$

Intuitively, since the AR(1) equation for $x[n]$ is

$$x[n] = ax[n-1] + w[n] ,$$

one expects that $y[n]$ is also related by an AR(1) equation with parameter $\sqrt{a}$ (provided that $a > 0$). Indeed, if

$$y[n] = \sqrt{a}y[n-1] + w[n] ,$$

then

$$r_y[m] = \sigma^2 \frac{a^{\frac{|m|}{2}}}{1-a} . \tag{3.11}$$

Note that subsampling $r_y[m]$ in (3.11) yields a $r_y[2m]$ which is similar to $r_x[m]$ in (3.10), except a difference in the scaling constant. Hence, $\frac{1}{\sqrt{1+a}}y[2n]$ (or equivalently we can scale the innovation power) has the same second-order statistics as $x[n]$.

Now consider higher order AR processes. Since the autocorrelation of an AR process is a sum of exponentials of its poles (i.e. the roots of the denominator of the synthesis filter), the constraint $r_y[2m] = r_x[m]$ suggests that $y[n]$ is characterized by a (possibly) mixed AR and MA process, where the AR process is parameterized by the square roots of the poles of $H(z)$.

For the case of $x[n]$ being an AR(2) process, a pure AR(2) for $y[n]$ does not yield the desired $r_y[2m]$ because of the scaling coefficients to each of the exponentials. This is where the study of AR(1) gives insight to the treatment of higher order AR processes. Since we can scale the AR(1) process to give the desired $r_y[2m]$, we can match each of the AR(2) exponentials separately, as shown in Figure 3.3. That is, let r_1 and r_2 be the poles of $H(z)$, then find β_1 and β_2 such that the synthesis filter $G(z)$ for $y[n]$ is of the form

$$G(z) = \beta_1 \frac{1}{1-\sqrt{r_1}z^{-1}} + \beta_2 \frac{1}{1-\sqrt{r_2}z^{-1}} ,$$

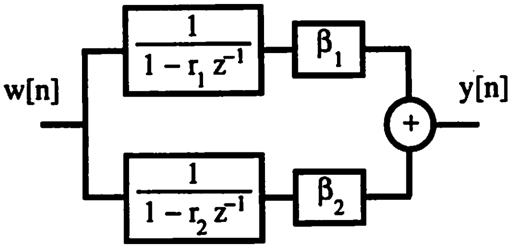


Figure 3.3: Decomposing an ARMA(2,1) process as a linear combination of two AR(1) processes.

where β_1 and β_2 are chosen such that $r_y[2m] = r_x[m]$. With this formulation, it suggests that $y[n]$ is an ARMA(2,1) process. Hence we conclude that to match an AR(2) process $x[n]$, we need $y[n]$ to be an ARMA(2,1) process.

So how do we calculate the parameters of the ARMA(2,1) process $y[n]$? The easiest way is to use the z-transform in (3.9). Let $H(z)$ and $G(z)$ be, respectively,

$$\begin{aligned} H(z) &= \frac{1}{1-a_1z^{-1}-a_2z^{-2}} = \frac{1}{(1-r_1z^{-1})(1-r_2z^{-1})}, \\ G(z) &= \frac{c_1(1+c_2z^{-1})}{1-b_1z^{-1}-b_2z^{-2}}, \end{aligned}$$

where r_1 and r_2 are the roots of the denominator of $H(z)$. Using (3.9) and after some algebra, matching the terms yields several equations of constraints which are used to solve for b_2, b_1, c_2, and c_1 in that order:

$$\begin{aligned} b_2^2 &= -a_2 \\ b_1^2+b_2 &= a_1 \\ -2b_2+2b_1c_2-2b_1b_2c_2-2b_2c_2^2 &= 0 \\ c_1^2(1+b_1^2+b_2^2+2b_1c_2-2b_1b_2c_2+c_2^2+b_1^2c_2^2+b_2^2c_2^2) &= 1 \end{aligned}$$

From the above equations, we see that for $y[n]$ to be a real process, there are only certain allowable range of values for a_1 and a_2 (specifically, $a_2 < 0$). Note that this does not imply that it is not possible to find $y[n]$ such that $r_y[2m] = r_x[m]$, but just that it is not possible if we constrain $y[n]$ to be an ARMA(2,1) process. Also, for valid a_1 and a_2, notice that there might be several solutions to the coefficients $b_i's$ and $c_i's$ because of the freedom to choose the positive and negative roots.

For higher order AR(p) processes, the procedure is similar. To find the denominator of $G(z)$ (which is possibly a ARMA(p,$p-1$) process), one can factor the denominator of

$H(z)$ as

$$(1-a^{(0)}z^{-1})\prod_{k=1}^{\lfloor \frac{p}{2} \rfloor}(1-a_1^{(k)}z^{-1}-a_2^{(k)}z^{-2}) \; ,$$

and match the denominator for each first- and second-order polynomial. The numerator terms are matched by expanding out (3.9). Again, there are constraints on the $a_i^{(k)}$'s and the resulting $G(z)$ is not unique.

Reconstruction from Estimated Parameters in 1-D

Once we obtain the parameters of $G(z)$, we can interpolate $x[n]$. Since we wish to retain as much as possible the information contained in $x[n]$, we propose to use the scheme shown in Figure 3.4(a) to interpolate $x[n]$. Essentially, this scheme adheres to the goal of the interpolation problem, which may be viewed as estimating the missing odd samples, but assuming the available samples are correct. In details, this scheme first processes $x[n]$ with the inverse filter $H^{-1}(z)$ to obtain the innovation $w[n]$. Then $w[n]$ is interleaved with an independently generated white Gaussian noise $v[n]$ with the same variance σ^2, and the result is filtered with $G(z)$. The even samples of the output of $G(z)$ are then replaced with the original $x[n]$ to obtain $y[n]$, the final interpolated signal. A more detailed block diagram of this method is shown in Figure 3.4(b).

From Figure 3.4(b), we can find the equivalent filters relating the inputs $x[n]$ and $v[n]$ to the output $y[n]$. To express this relationship in the z-transform domain, we can take the z-transform of a realization of the random processes $x[n]$ and $y[n]$ to obtain

$$\begin{aligned} Y(z) &= X(z^2)+\frac{1}{2}[z^{-1}V(z^2)(G(z)+G(-z))+X(z^2)H^{-1}(z^2)(G(z)-G(-z))] \\ &= X(z^2)+z^{-1}[V(z^2)G_0(z^2)+X(z^2)H^{-1}(z^2)G_1(z^2)] \\ &= X(z^2)[1+z^{-1}H^{-1}(z^2)G_1(z^2)]+z^{-1}V(z^2)G_0(z^2) \; , \end{aligned} \tag{3.12}$$

where $G_0(z)$ and $G_1(z)$ are the polyphase components of $G(z) = G_0(z^2)+z^{-1}G_1(z^2)$. From (3.12), we can find an equivalent interpolation filter to the one in Figure 3.4(b), as shown in Figure 3.4(c).

3.4.4 Adding the Interpolated Components Together

To obtain the interpolated signal, we add together the individually interpolated component. The reconstruction of the high resolution signal $f_0[n,m]$ is done in the frequency

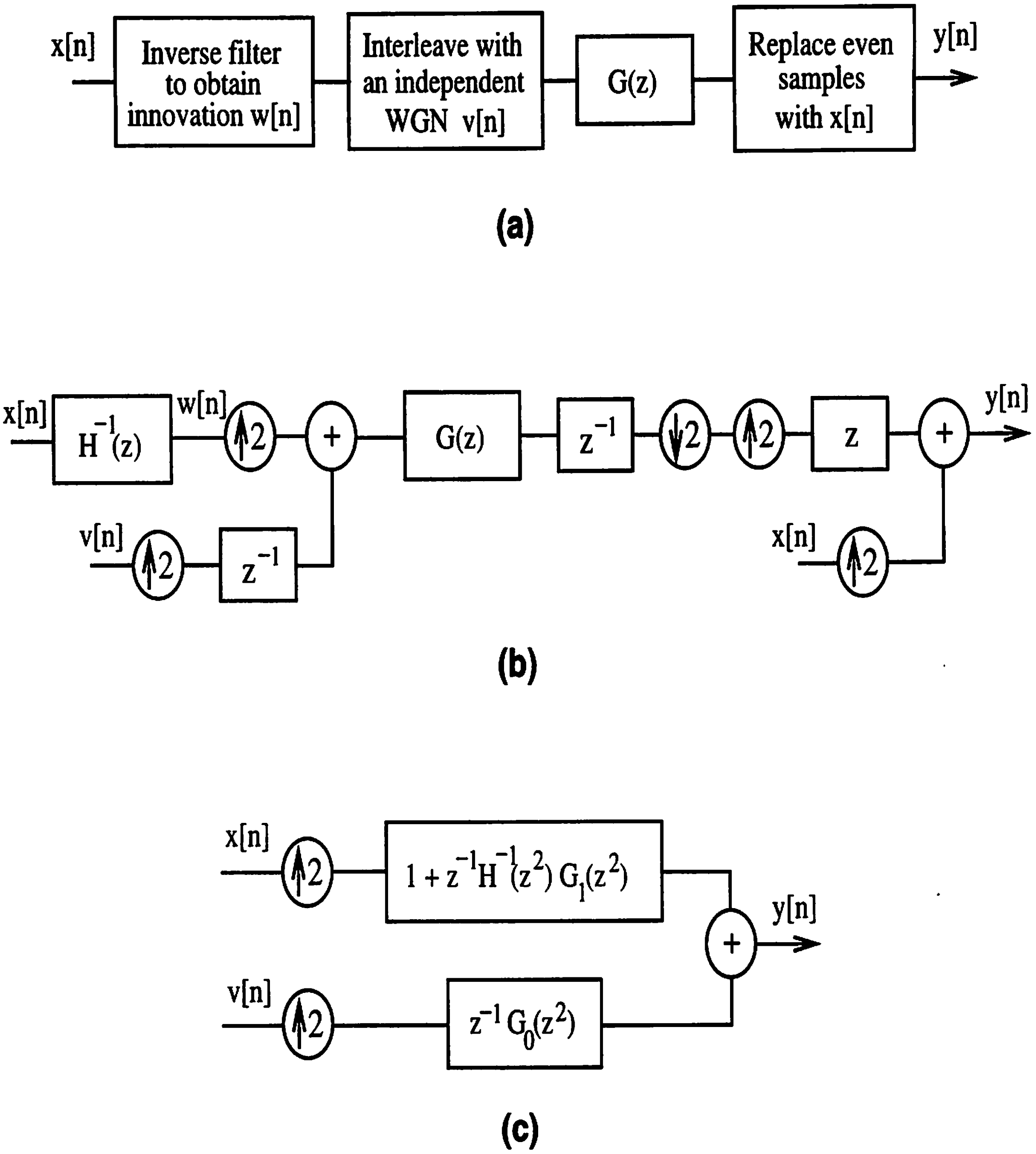

x[n]
Inverse filter to obtain innovation w[n]
Interleave with an independent WGN v[n]
G(z)
Replace even samples with x[n]
y[n]
(a)
x[n]
H^-1(z)
w[n]
↑2
+
G(z)
z^-1
↓2
↑2
z
+
y[n]
v[n]
↑2
z^-1
x[n]
↑2
(b)
x[n]
↑2
1 + z^-1 H^-1(z^2) G1(z^2)
y[n]
+
v[n]
↑2
z^-1 G0(z^2)
(c)

Figure 3.4: Interpolating an ARMA process. (a) shows the global view of the scheme, while (b) shows the specific details. The diagram in (c) is a simplified equivalent system as the one in (b).

domain. We assume that the lower half of the spectrum of $f_0[n,m]$ is from the available signal, but the upper half is to be estimated. The interpolated AR process is added to the interpolated (time-domain) harmonic component, which is then transformed into the frequency domain. The upper half of this spectrum is taken to be the estimate of the upper half of the spectrum of $f_0[n,m]$. The reconstructed signal $f_0[n,m]$ is obtained by taking the inverse Fourier transform, followed by assigning the even samples to the original available signal.

Chapter 4

Results

The results of the wavelet-based enhancement algorithm are shown in Section 4.1, and some preliminary results of the texture interpolation are shown in Section 4.2.

4.1 Experimental Results on Wavelet-Based Enhancement

To obtain a test image, the original 512×512 Lena is lowpass filtered, subsampled by 2, and the process is repeated to obtain a low resolution image of 128×128, from which a 64×64 subimage is extracted as the available data for all of the interpolation methods. This available image is shown in Figure 4.1(a). The lowpass filter used in obtaining the test image is a separable 2-D filter, $H(\omega_1, \omega_2) = H_1(\omega_1)H_1(\omega_2)$, where the impulse response of $H_1(\omega)$ is $[-1, 0, 9, 16, 9, 0, -1]$ normalized to $H_1(0) = 1$. Note that this filter is chosen independently of the filters used in the wavelet transform.

The filters used in the filter bank are similar to those used in [5], and they are derived from the 1-D quadratic spline wavelet. The difference here is that the filters are normalized such that $H(z = 1) = 1, G(z = 1) = 0, H(z = -1) = 0, G(z = -1) = 1$. The coefficients of the filters H_0, H_1, G_0, G_1, and L are given in Table 4.1.

Due to discretization, the estimated Lipschitz exponent is not the same as what the continuous-time model predicts. Therefore, scaling constants are multiplied to each scale of the wavelet transform, $W_j f$ (see equation (2.4)). In particular, these constants are found empirically such that the extrema values of a step edge are the same across the scales (corresponding to Lipschitz exponent $\alpha = 0$). Table 4.2 lists these scaling constants.

Figure 4.1(b) shows a 256×256 image of Lena obtained from performing the enhance-

n	H_0	H_1	G_0	G_1	L
-3				-0.03125	0.0078125
-2		0.125		-0.21874	0.0468750
-1	0.125	0.375		-0.68750	0.1171875
0	0.375	0.375	0.5	0.68750	0.6562500
1	0.375	0.125	-0.5	0.21874	0.1171875
2	0.125			0.03125	0.0468750
3					0.0078125

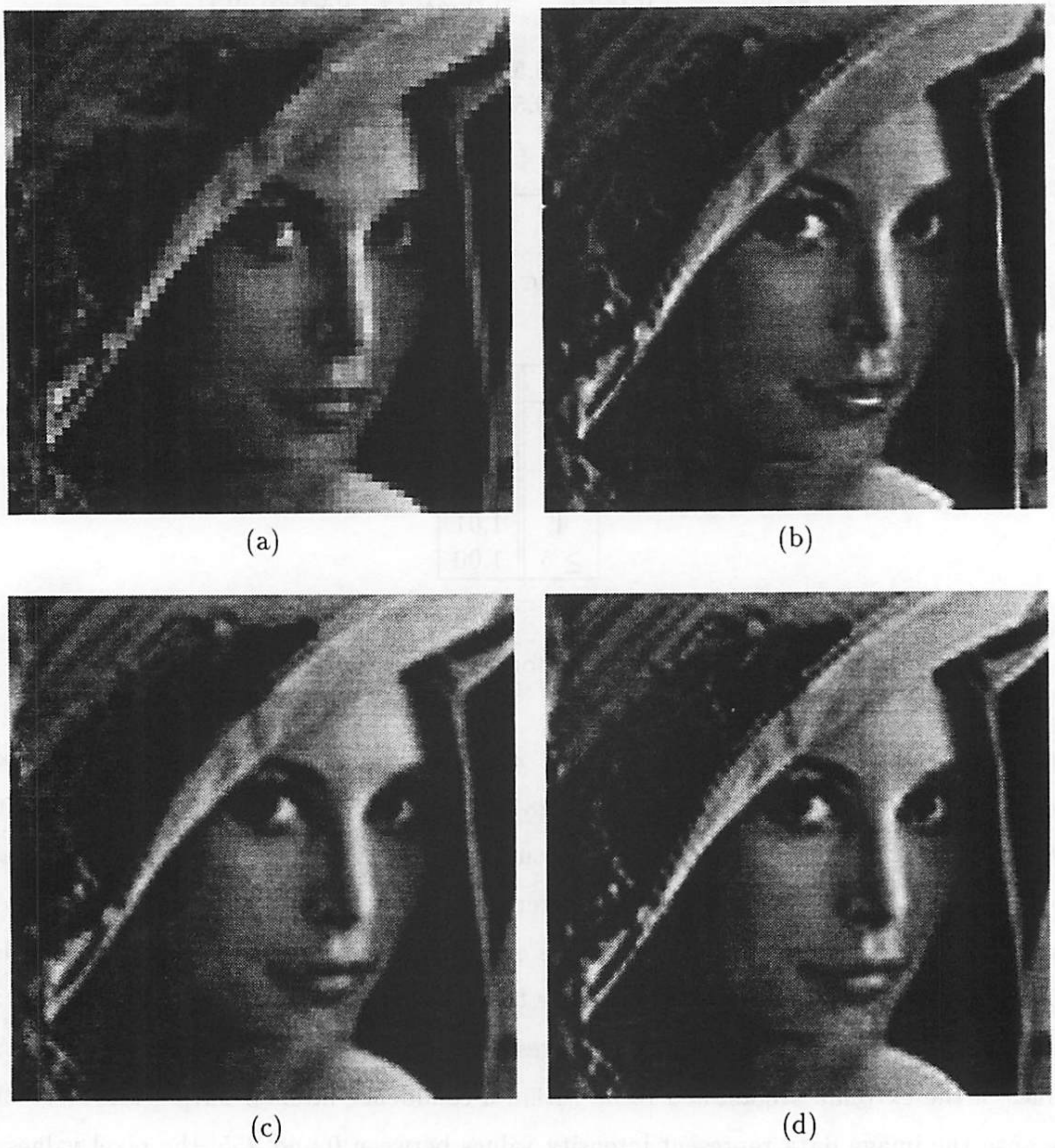
Table 4.1: Coefficients of the quadratic spline FIR filters H_0, H_1, G_0, G_1 and L.

j	λ_j
1	1.50
2	1.12
3	1.03
4	1.01
≥ 5	1.00

Table 4.2: Scaling constants for the quadratic spline wavelets.

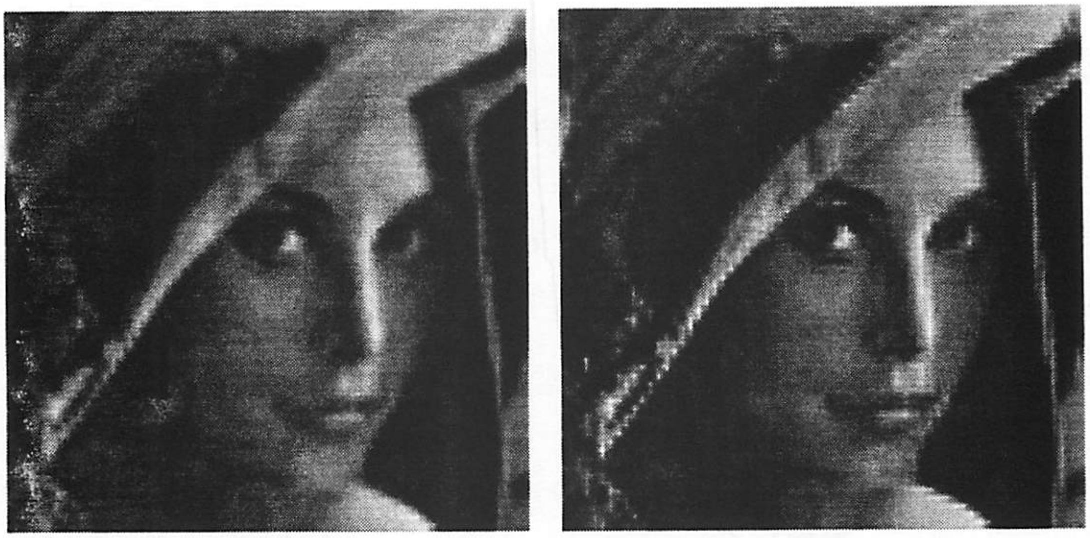
ment algorithm iteratively twice on the 64 × 64 test image (i.e. magnifying the 64 × 64 image, and then magnifying the resulting 128 × 128 image again using the algorithm, with a total of 4× magnification). Convergence occurs rather quickly and the reconstruction after obtaining the initial estimates or after 1-2 iterations is acceptable. After 5-10 iterations the image quality is quite good, and the images either do not change discernibly afterwards or they degrade due to overshoots. We observe that using regular filters such as the quadratic spline yields better results (less blocky images) than non-regular filters (such as Haar). The values of the extrema are allowed to be within a confidence interval during reconstruction. Because the image data represent intensity values between 0 and 255, the pixel values are clipped to be within this interval during reconstruction.

We choose several standard methods to compare our method against. Figure 4.1(c) shows a bicubic spline interpolated image, and Figure 4.1(d) shows the image resulting from performing twice iteratively the 2× magnification with bicubic spline followed by unsharp masking. Figure 4.1(e) shows a bilinearly interpolated image, and Figure 4.1(f) shows the result of performing twice iteratively the 2× magnification with bilinear interpolation



(a) (b) (c) (d)

Figure 4.1: Comparison of various interpolation methods on Lena (All are 4× magnification). (a) Original low resolution available image (64 × 64). (b) Magnifying Lena 4× using the 2× interpolation algorithm iteratively (256 × 256 image). (c) Bicubic spline interpolation. (d) Bicubic spline followed by unsharp masking.



(e) (f)

Figure 4.1: Comparison (continued). (e) Bilinear interpolation. (f) Bilinear interpolation followed by unsharp masking.

followed by unsharp masking. The unsharp masking used is the common discrete Laplacian gradient [4] with $\lambda = 1$. The images produced by the standard methods are blurrier or have blockier edges.

For the algorithm variant that does not constrain the signal extrema, ringing effects tend to occur around the edges. On the other hand, when constraining the values to be the initial estimates, the resulting image is not very good because the initial estimates are not very reliable. A direct 4× magnification was also briefly tried, but the result was not as good as performing two 2× magnifications. This is probably due to the fact that the more restrictive constraints of 2× magnification narrows the possible solutions, and performing it iteratively narrows the reconstruction set successively, whereas the direct 4× magnification has a larger reconstruction set.

A remark should be made on comparing our enhancement method with bilinear interpolation followed by unsharp masking. The procedure of iteratively upsampling a signal, performing linear interpolation and unsharp masking does not converge to a regular filter

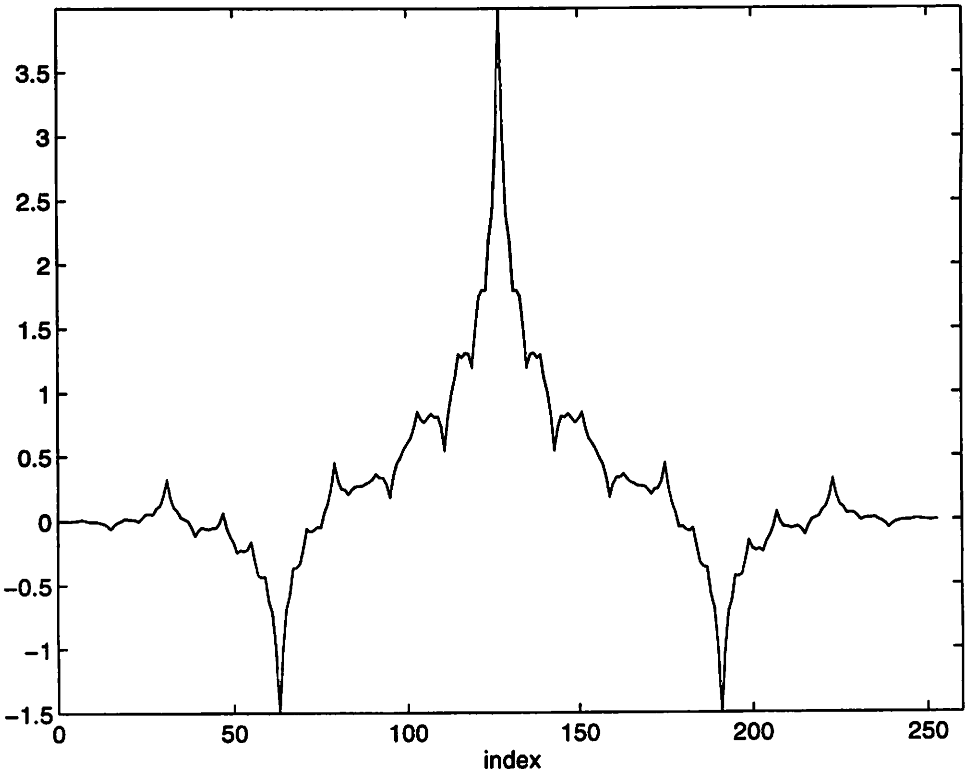


Figure 4.2: Iterating the process of upsampling by 2, followed by filtering by linear interpolator and unsharp masking using the discrete Laplacian gradient with $\lambda = 1$.

when the mask used is the Laplacian gradient with the free parameter $\lambda = 1$,

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \lambda \begin{bmatrix} 0 & -\frac{1}{4} & 0 \\ -\frac{1}{4} & 1 & -\frac{1}{4} \\ 0 & -\frac{1}{4} & 0 \end{bmatrix},$$

where the origin is taken to be the center. Figure 4.2 shows the 1-D case of the iterated filter where the linear interpolation and unsharp masking impulse responses are, respectively, $[.5,\ 1,\ .5]$ and $[-.5\lambda,\ 1+\lambda,\ -.5\lambda]$. Furthermore, because of the change in sampling rate, this operation not only accentuates the high frequency components but also other frequency components from replicates of the original spectrum. Hence, this method may not be adequate for image interpolation.

4.2 Preliminary Results on Texture Enhancement

As previously mentioned, the texture enhancement work is mostly theoretical for now, and only preliminary results have been obtained. In Section 4.2.1, we show a 1-D interpolated AR(2) process. Section 4.2.2 shows interpolating a synthetic 1-D "texture-like" signal. For 2-D interpolation, we have not been able to obtain a satisfactory decomposition of the various components as claimed in [2] and thus do not yet have results.

4.2.1 Interpolating a 1-D AR Process

An experiment was done on a AR(2) process to see how the interpolation described in Section 4.2.1 performs. An AR(2) process with the canonical synthesis filter

$$H(z) = \frac{1}{(1 - .5z^{-1})(1 - .25z^{-1})} = \frac{1}{1 - .75z^{-1} + .125z^{-2}} \tag{4.1}$$

is generate by a white Gaussian noise with variance $\sigma^2 = 1$. Then using the interpolation scheme in Figure 3.4, we interpolate the AR process. As a preliminary experiment, we assume that $H(z)$ is known, so only the parameters of $G(z)$ has to be calculated. The high resolution synthesis filter $G(z)$ is

$$G(z) = \frac{.7121 - .1621z^{-1}}{1 - 1.2071z^{-1} + .3536z^{-2}} \; .$$

Figure 4.3 shows the comparison between the proposed scheme and linear interpolation. The result of our scheme shows a lot more fluctuation.

4.2.2 Interpolating a Synthetic 1-D Texture-like Signal

To test out the entire interpolation scheme (i.e., extracting and interpolating the deterministic and indeterministic component) in 1-D, a synthetic signal is created. The synthetic signal consists of a square wave of magnitude ±20 and period 32, with additive AR(2) process having the same $H(z)$ as in (4.1) and driven by a white Gaussian process with power $\sigma^2 = 10$. First, the sharp peaks in the deterministic component are selected and extrapolated to the high frequencies. The deterministic component is subtracted to obtain a residual that should only contain the indeterministic component. We use Burg's method to estimate the AR(2) parameters of the residual process (for preliminary experiment, we assume the order is known to be 2, and thus there was no order selection), which is estimated

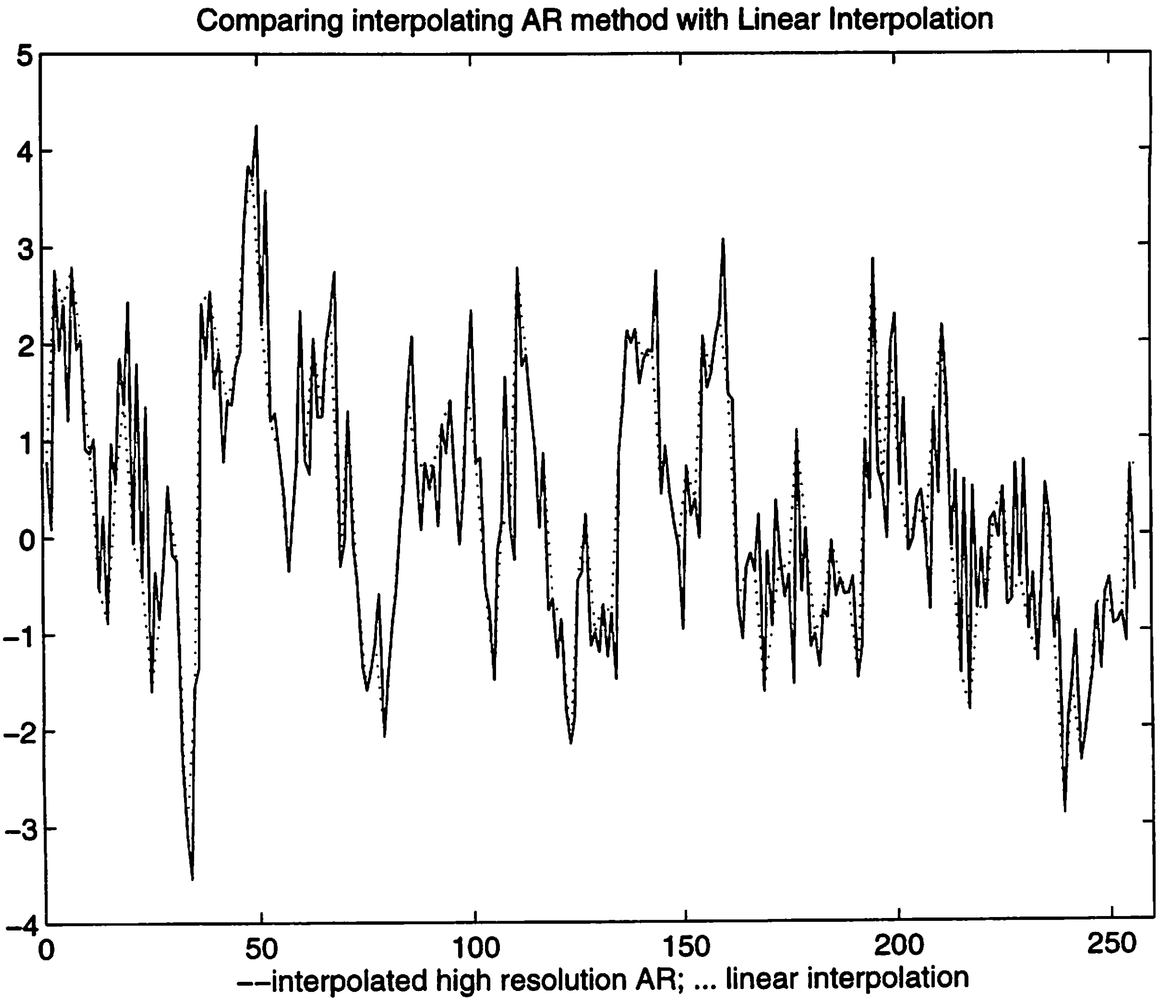


Figure 4.3: The result of interpolating an AR(2) process. The solid line is the result using the proposed scheme, and the dotted line is the linearly interpolated.

to be

$$H(z) = \frac{1}{1 - .6937z^{-1} + .1477z^{-2}} ,$$

and the innovation power is estimated to be 9.71. The high resolution synthesis filter $G(z)$ is then calculated to be

$$G(z) = \frac{.7161 - .1741z^{-1}}{1 - 1.2093z^{-1} + .3843z^{-2}} .$$

The interpolated signal is shown in Figure 4.4 in solid lines, in comparison with linear interpolation in dotted lines. Figure 4.4(a) shows the entire signal, and Figure 4.4(b) shows a small segment, where we can see that the proposed method yields sharper edges.

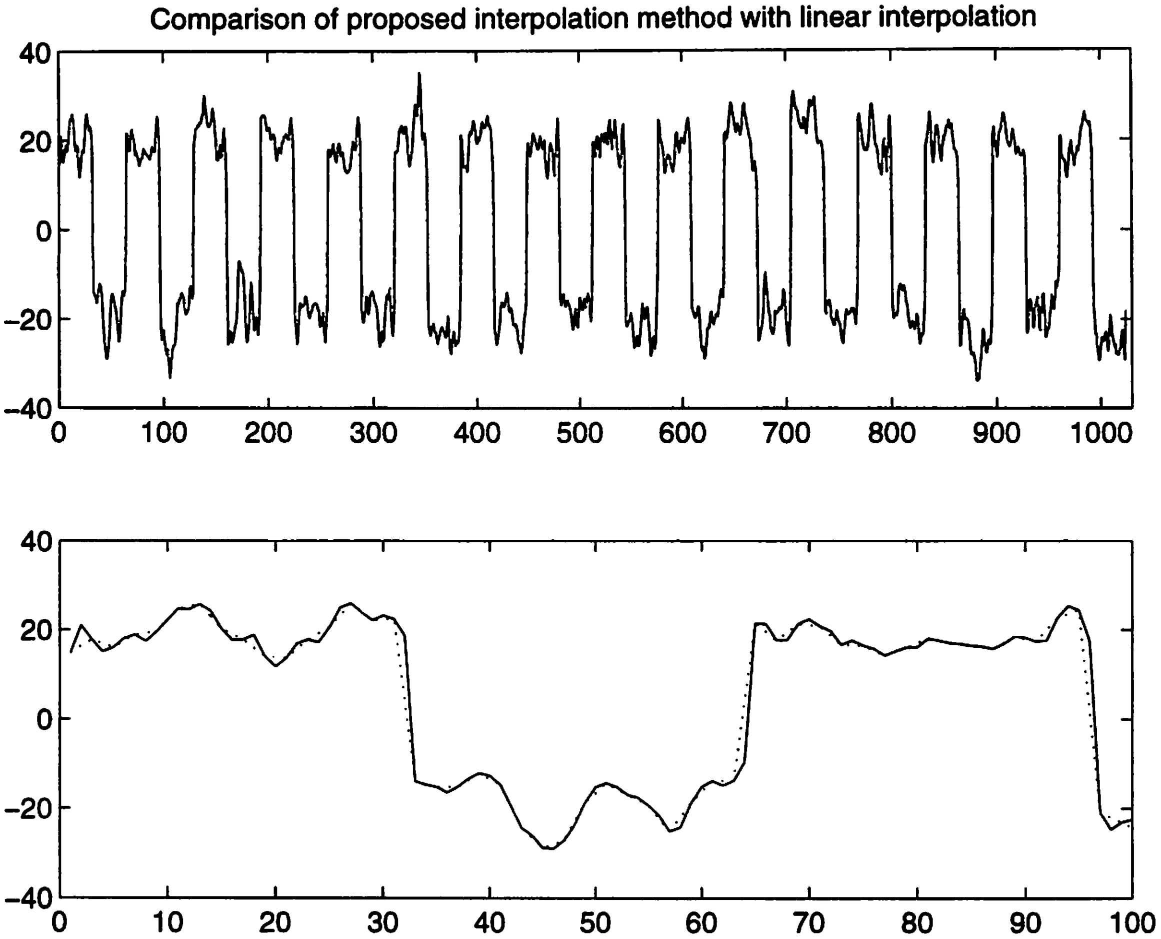


Figure 4.4: Comparison of interpolation methods on a synthetic square wave with additive AR(2) colored noise. The solid line is the proposed method, and the dotted line is linear interpolation. (a) shows the entire sequence and (b) shows an enlargement of one section.

Chapter 5

Conclusion and Future Work

This thesis proposed two approaches to the image interpolation problem, aimed toward two very different features of an image: isolated edges and texture regions. The wavelet-based enhancement algorithm is an elegant and novel approach which adaptively enhances edges based on what is available in the low resolution signal. Experiments show that the enhancement algorithm produces a sharper image than traditional methods. There are, however, drawbacks. The proposed enhancement method is a much more complex method than the traditional linear methods. Reconstruction using iterative projections is also a costly process. Future work would include cutting down on the amount of necessary computation and finding some simplifications.

The second method is a theoretical proposal on interpolating texture regions. The basic idea is to decompose a texture field into deterministic and indeterministic components and interpolate each separately. Preliminary experiments on 1-D synthetic signals show that the method generates sharper edges than linear interpolation, implying that it may be a method worthwhile for further experiments. More work is necessary to estimate the parameters accurately in the 2-D framework.

With these two tools, different regions of an image can be treated differently. The final comprehensive goal to the interpolation problem would be to segment an image into edge and texture regions and interpolate them separately before adding them back together, generating a image that is enhanced in both edge regions and texture areas.

Bibliography

[1] Z. Cvetković and M. Vetterli, "Discrete-time wavelet extrema representation: design and consistent reconstruction", *IEEE Trans. on Signal Processing*, vol. 43, no. 3, pp. 681-693, March 1995.

[2] J.M. Francos, A.Z. Meiri and B. Porat, "A unified texture model based on a 2-D Wold-like decomposition", *IEEE Trans. on Signal Processing*, vol. 41, no. 8, pp. 2665-2678, August 1993.

[3] H. Greenspan and C. Anderson, "Image enhancement by non-linear extrapolation in frequency space", pre-print.

[4] A. Jain. *Fundamentals of Digital Image Processing*, Prentice Hall, 1989.

[5] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges", *IEEE Trans. on PAMI*, vol. 14, no. 7, pp. 2207-2232, July 1992.

[6] T.L. Marzetta, "Two-dimensional linear prediction: Autocorrelation arrays, minimum-phase prediction error filters and reflection coefficient arrays", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-28, pp. 725-733, 1980.

[7] R.R. Schultz and R.L. Stevenson, "A Bayesian approach to image expansion for improved definition", *IEEE Trans. on Image Processing*, vol. 3, no. 3, pp. 233-242, May 1994.

[8] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Prentice Hall, 1995.

[9] S.G. Chang, Z. Cvetković and M. Vetterli, "Resolution enhancement of images using wavelet transform extrema extrapolation", *Proc. Int. Conf. Acoust. Speech, Signal Processing* (Detroit, MI), vol. 4, pp. 2379-2382, May 1995.