Chapter 4

Still Image Zooming

In the last chapter we discussed methods to restore color images. We proposed a new approaches to consider interchannel blurring. Once the images are restored, we move on to enhance the spatial resolution of these images by applying zooming algorithms. In this chapter, we propose a new technique to still image zooming. In the following chapters, we extend this technique to zooming of compressed video.

4.1 Introduction

Image interpolation or zooming or generation of higher resolution images is an important branch of image processing which is attracting a good number of researchers. The recent IEEE conference on Image Processing (ICIP-2000) had a full section on interpolation. Classical methods include linear interpolation and pixel replication. Linear interpolation tries to fit a straight line between two points. This technique leads to blurred images. Pixel replication copies a neighboring pixel to the empty location. This technique tends to produce *blocky* images. Approaches like spline and sinc interpolation have been proposed to reduce these two extremities. Spline interpolation is inherently a smoothing operation, while sinc produces ripples (the Gibbs phenomenon) in the output image. Researchers have proposed different solutions to the interpolation problem. Schultz and Stevenson [121] propose a Bayesian approach for zooming. In the super-resolution domain, Deepu and Chaudhuri [112] propose a physics based approach. Carey *et al* [145] propose wavelet a based approach. Jensen and Anastassiou [62] propose a non-linear method for image zooming. Some of the recent approaches in the super-resolution domain are compiled by Chaudhuri [18]

In this chapter, we propose two new approaches for image interpolation and extend

these two basic techniques. The first approach models the image as a Markov Random Field (MRF) and the interpolation problem is cast as a Maximum *a posteriori* (MAP) problem. This gives a slightly blurred interpolated image. In the second approach, we exploit the properties of Multiresolution (wavelet) Analysis (MRA) of images. In particular, we use zerotree based image compression proposed by Shapiro [127]. In this approach, we propose a simple method to estimate high frequency wavelet coefficients to avoid smoothing of the edges. We use ideas of zero-tree coding [127] and the multiscale edge characterization of Mallat [84] to estimate the wavelet coefficients. Simulation results show that the proposed method gives better performance, particularly when we compare SNRs; but slightly blocky images. To strike a balance, we combine these two approaches and interpolation is done adaptively - either MRA or MRF. In the final extension of the MRA approach, we use the Hotelling or Karhunen Loéve (K-L) transform on multi spectral images and interpolate the principal component to get a high resolution monochrome image.

This chapter is organized as follows: Section 2 gives background on wavelets, multiresolution analysis (MRA) and K-L transform. In Section 3 we overview some of the existing methods. Section 4 gives the formulation for MRF based image interpolation. Section 5 discusses the proposed method using MRA, Karhunen Loéve (K-L) transform and scaling-function based interpolations. Section 6 extends Section 5 to color images. Section 7 presents a discussion on simulation results to illustrate superiority of the proposed method. Section 8 provides concluding remarks.

4.2 Background

We present background material on multiresolution (wavelet) analysis of signals and the K-L transform.

4.2.1 Wavelets

Let $L^2(I\!R)$ be the space of all square integrable functions. Then, it has been shown [16, 20] that there exists a multiresolution analysis of the form: $L^2(I\!R) = \bigcup_{j\in\mathbf{Z}} V_j$, ($\mathbf{Z}$ is set of integers) where, the sub-spaces $\{V_j\}$ have the following properties:[16]

1. $V_j \subset V_{j+1}, \qquad V_{-\infty} = \{0\}, \qquad V_{\infty} = L^2(I\!R)$
2. Let $\phi(t) \in L^2(I\!R)$ be a scaling function; then, $V_j = \overset{Span}{k}\{\phi_{j,k}(t), \quad k \in \mathbf{Z}\}$ and $\phi_{j,k} = 2^{j/2}\phi(2^j t - k)$ for all $j \in \mathbf{Z}$. As a consequence, $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$

3. According to property-2, we see that $\phi(t) \in V_0$, then $\phi(2t) \in V_1$. Moreover, $V_0 \subset V_1$, therefore, $\phi(t) \in V_1$. Consequently, $\phi(t)$ can be written as

$$\phi(t) = \sum_k h(k)\sqrt{2}\phi(2t-k) \tag{4.1}$$

where, $h(k)$ is the scaling function coefficient

4. Let the orthogonal direct sum decomposition of V_j be $V_j = V_{j-1} \oplus W_{j-1}$. Then we can write

$$L^2(I\!R) = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \tag{4.2}$$

Moreover, there exists a function $\psi(t)$ (referred to as the wavelet) such that

$$W_j = span\{\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k), \quad k \in \mathbf{Z}\}$$

5. Since $W_0 \subset V_1$ we can express $\psi(t)$ as

$$\psi(t) = \sum_k h_1(k)\sqrt{2}\phi(2t-k) \tag{4.3}$$

6. Finally, for any $g(t) \in L^2(I\!R)$, we have the decomposition

$$g(t) = \sum_k c_0(k)\phi_{0,k}(t) + \sum_{j=0}^{\infty}\sum_{k=-\infty}^{\infty} d_j(k)\psi_{j,k}(t) \tag{4.4}$$

Coefficients $c_0(k)$ and $d_j(k)$ are calculated (inner product) as:

$$\begin{array}{rcccl} c_0(k) & = & < g(t), \phi_k(t) > & = & \int g(t)\phi_k(t)dt \\ d_j(k) & = & < g(t), \psi_{j,k}(t) > & = & \int g(t)\psi_{j,k}(t)dt \end{array}$$

The subspace properties are illustrated in Fig. 4.1

Apart from these, we make use of the following two properties [85]: Let A_2^j be the operator which approximates a signal at a resolution 2^j. Then:

- The approximation signal at a resolution 2^{j+1} contains all the necessary information to compute the same signal at a lower resolution 2^j. This is the causality property.
- The approximation operation is similar at all resolutions. The spaces of approximated functions should thus be derived from one another by scaling each approximated function by the ratio of their resolution values. That is (as mentioned earlier),

$$\forall j \in \mathbf{Z}, \quad f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}.$$

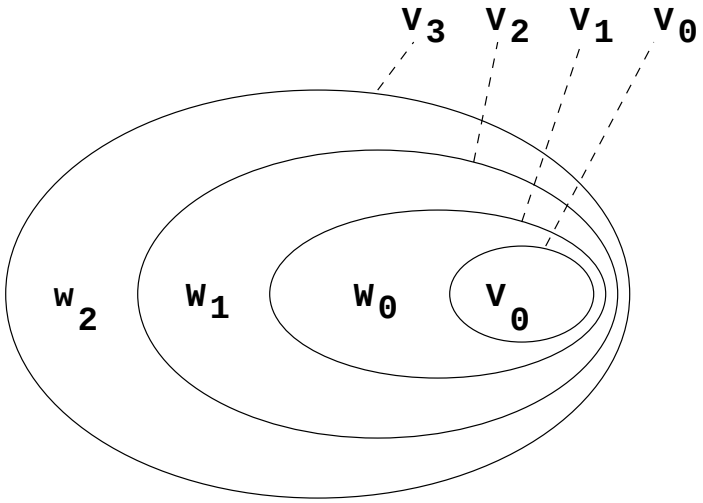


Figure 4.1: Scaling function and Wavelet Vector Spaces

4.2.2 The KL Transform

Let $\{x(n), 1 \leq n \leq M\}$ be a complex random sequence whose autocorrelation matrix is $\mathbf{R}$. Moreover, let

$$\mathbf{R}\Theta_k = \lambda_k \Theta_k \qquad 1 \leq k \leq M$$

Where λ_k and Θ_k are eigenvalues and eigenvectors, respectively, of $\mathbf{R}$,. Then the Karhunen-Loéve (KL) transform of $x = [x(1), \cdots, x(N)]^T$ is defined as [59]

$$y = \mathbf{\Phi}^{*T} x \quad \mathbf{\Phi} = [\Theta_1, \ldots, \Theta_M] \tag{4.5}$$

Moreover the inverse transform is

$$x = \mathbf{\Phi} y \; = \; \sum_{k=1}^{M} y(k)\Theta_k \tag{4.6}$$

where $y(k)$ is the k^{th} element of the vector y.

The KL transform orthogonalizes the data, namely,

$$\begin{aligned} &E[yy^{*T}] = \mathbf{\Phi}^{*T}\{E[xx^{*T}]\}\mathbf{\Phi} = \mathbf{\Phi}^{*\mathbf{T}}\mathbf{R}\mathbf{\Phi} = \mathbf{\Lambda} \\ &E[y(k)y^*(l)] = \lambda_k \delta(k-l) \end{aligned} \tag{4.7}$$

If $\mathbf{R}$ represents the covariance matrix rather than the auto-correlation matrix of x, then the sequence $y(k)$ is uncorrelated. The unitary matrix $\mathbf{\Phi}^{*T}$ is called the KL transform matrix and $\mathbf{\Phi}$ is an $M \times M$ unitary matrix, that reduces $\mathbf{R}$ to its diagonal form. One dimensional KL transform can easily be extended to two dimensions.

4.3 Some Existing Methods for Image Zooming

Interpolation involves filling intermediate values. Most commonly used methods involve placing zeros in the intermediate samples, and then passing this through a filter. Different methods of interpolation are attributed to different types of filters. Here we mention a few of the popular interpolation methods, paying special attention to wavelet-based interpolation techniques. We compare our method with some of these methods.

4.3.1 Spatial domain Methods

- *Pixel replication:* It is a zero-order hold method, where, each pixel along a scan line is repeated once and then each scan line is repeated. Or equivalently, pad rows and columns with zeros and then convolve with the mask

$$\mathbf{H} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

  Due to replication of pixels, it gives a blocky image.

- *Linear interpolation:* This is basically a first order hold method. Here, rows and columns of the low resolution images are first interleaved with zeros. Then, a straight line is fitted along rows, followed by straight line fit along columns. The straight line fits are equivalent to convolving the image with the mask,

$$\mathbf{H} = 1/4 \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 1 & 1/2 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}$$

  origin being the center of the mask. Since it is an averaging filter, linear interpolation tends to produce a blurred image.

- *Spline interpolation:* Spline (cubic spline) interpolation [88, 143] is one of the most popular techniques. The goal is to get an interpolation that is smooth in the first derivative, and continuous in the second derivative. We have used the spline routine of [148].

Other approaches include MRF based super-resolution proposed by Deepu and Chaudhuri [110, 111]. Here, authors consider availability of decimated, blurred and noisy versions of a high resolution image, which are used to generate a super-resolved image. A known blur acts as a cue in generating super-resolution images. They model the image as MRF; use

gradient descent and SA techniques. For super-resolution applications, they also propose a generalized interpolation method [109, 111]. Here, a space containing the original function values is decomposed into appropriate subspaces. These subspaces are chosen so that the re-scaling operation preserves those properties of the original function. On combining these re-scaled sub-functions, they get back the original space containing the scaled or zoomed function.

4.3.2 Transform Domain Methods

- *Sinc interpolation:* Sinc interpolation is basically a Fourier transform (FT) based interpolation method. It assumes the signal under consideration to be band-limited and thus can be carried out by zero extension of the FT. That is, we take the N point Discrete Fourier Transform (DFT) of the original sequence, pad it with zeros for $N+1$ to $2N$, and take $2N$ - point inverse DFT. This results in a sequence of $2\times$ length.

- Instead of choosing the DFT, we can choose Discrete Sine Transform (DST) or Discrete Cosine Transform (DCT) to perform interpolation. Method and performance will be similar to that using DFT. Martucci [87] proposes a new set of basis for DST and DCT approaches and use convolution-multiplication property of DSTs and DCTs which improves the performance.

4.3.3 Wavelet Techniques

Since our focus is on wavelet-based interpolation, we will give more emphasis to wavelet based interpolation techniques

In the recent past, wavelets have been used for modeling images - particularly the smooth regions [82, 84]. Extension of these works, particularly for image zooming can be found in [34, 89, 96, 98, 145].

Crouse *et al* [82] propose the use of Hidden Markov Model (HMM) for predicting wavelet coefficients over scales. In the training phase, HMM is trained using an image database. They predict *exact* coefficient from the *observed* coefficient of a noisy image - for de-noising application. The principle used here is that the coarser scale coefficients are less affected by noise, while the detail coefficients contain most of the noise. A similar idea can be extended to image zooming.

Carey *et al* [145] use a property, namely, near sharp edges, the wavelet coefficients decay exponentially over scale [84, 86]. At each index, an exponential fit over scale is

used for wavelet coefficients. If the fit was close enough to exponential, then it is used to predict the detail signal at the finer scale, else data is left unmodified. On a similar basis, Chang *et al* [34] extrapolate the features in textured regions as well. Their method extrapolates wavelet transform extrema across scales, and, important singularities are selected. Corresponding extrema across the scales are associated using least squares error criterion.

Nicolier *et al* [98] use zero-crossings to predict the high frequency coefficients in the wavelet domain. Using 2^{nd} order type wavelets, zero-crossings in the detail coefficients are produced at the location of a step signal. They have shown the result using 9th order B-spline wavelet for the purpose.

4.3.4 Scaling Function Based Zooming

This method was proposed by Savagoankar [120]. Consider a wavelet ψ; and let this wavelet generate a multiresolution analysis of $L^2(I\!R)$. Moreover, if $f \in V_j$ for some j, where, $V_{n-1} \subset V_n$ is the MRA corresponding to ψ, then one can write,

$$f(.) = \sum_{k=-\infty}^{\infty} c_{j,k}\phi_{j;k}(.) \tag{4.8}$$

where ϕ is the scaling function corresponding to the wavelet ψ. Thus, we can express f completely in terms of its scaling function coefficients $c_{j,k}$. Hence, from the given data samples, if we can somehow estimate the scaling function coefficients at resolution j then we have solved the problem. Given an MRA of $L^2[0,1]$ with a compactly supported, p times differentiable, scaling function ϕ, we want to estimate the scaling function coefficients of a smoothest function $\hat{f}$, at some resolution j. This scaling function passes through the samples of the given function $b_i = f(i/m)$, $i = 1, \ldots, m = 2^k$. We assume that f and $\hat{f} \in V_j$, for some known $j \geq k$.

Let us denote the j^{th} scale scaling coefficient as $c_{j,l}$. From the above assumption, we can write

$$\hat{f}(.) = \sum_{l=0}^{2^j-1} c_{j,l}\phi_{j;l}(.) \tag{4.9}$$

Then, we have the following conditions on $\hat{f}$.

1.

$$\hat{f}(i/2^k) = \sum_{l=0}^{2^j-1} c_{j,l}\phi_{j;l}(i) = f(i/2^k) = b_i \quad i = 1, \ldots, 2^k \tag{4.10}$$

2. $\hat{f}$ should be at least as smooth as f.

Once we have $c_{j,l}$, we can compute the value of $\hat{f}$ at any point using Eqn. (4.9). Thus, the problem of estimating f is the same as that of estimating $c_{j,l}$. Next, if b and c are vectors such that $b^k = [b_1, b_2, \ldots, b_{2^j}]^T$ and $c^j = [c_{j,0}, c_{j,1}, \ldots, c_{j,2^j-1}]^T$, then Eqn. (4.10) can be written as:

$$b^k = \Phi_k^j . c^j \tag{4.11}$$

where, Φ is a matrix, given by

$$\Phi_k^j = \begin{pmatrix} \Phi_{j;0}(1) & \Phi_{j;1}(1) & \ldots & \Phi_{j;2^j-1}(1) \\ \Phi_{j;0}(2) & \Phi_{j;1}(2) & \ldots & \Phi_{j;2^j-1}(2) \\ \vdots & \vdots & \ddots & \vdots \\ \Phi_{j;0}(2^k) & \Phi_{j;1}(2^k) & \ldots & \Phi_{j;2^j-1}(2^k) \end{pmatrix} \tag{4.12}$$

Consider the minimization problem ($\|c^j\|^2 = (c^j)^T c^j$)

$$\min_{\Phi_k^j c^j = b^k} \|c^j\| \tag{4.13}$$

The solution for Eqn. (4.13) is well known and is given by

$$c^j = \Phi_k^{j*} \psi (\Phi_k^j . \Phi_k^{j*})^{-1} \psi b^k \tag{4.14}$$

Thus we have estimated scaling coefficients of a smoothest possible linear interpolation. The smoothness condition assures visual quality of the signal while interpolating images.

4.4 Proposed Method - 1 : MRF Based Approach

Assume that the given low resolution image Y is modeled as

$$Y = \mathbf{D}X + N \tag{4.15}$$

where, Y is the image (vector), obtained from down sampling the original image (vector) X, $\mathbf{D}$ is the decimation matrix and N is a Gaussian noise vector with zero mean and covariance $E[NN^T] = \sigma^2 I$. Moreover N is statistically independent of X. Then, using Bayes rule, the *a posteriori* distribution can be expressed as

$$P[X = x|Y = y] = \frac{P[Y = y|X = x]P[X = x]}{P[Y = y]} \tag{4.16}$$

Next using (4.15) we have

$$P[X = x|Y = y] = \frac{P[Y = \mathbf{D}X + N|X = x]P[X = x]}{P[Y = y]} = \frac{P[N = Y - \mathbf{D}X|X = x]P[X = x]}{P[Y = y]} \tag{4.17}$$

Now exploiting the fact that N is statistically independent of X we obtain

$$P[X = x|Y = y] = \frac{\exp\{-\|Y - \mathbf{D}X\|^2\}\exp\{-U(x)\}}{ZK} \tag{4.18}$$

where $K = P[Y = y]\ (2\pi\sigma^2)^{N^2}$, Z is the normalizing constant. In Eqn. (4.18), we see that maximizing $P[X = x|Y = y]$ is equivalent to minimizing $U_p(x) = \|Y - \mathbf{D}X\|^2 + U(x)$. Assume the observed and interpolated images Y and $\hat{X}$, respectively, are lexicographically ordered vectors. Then we assume the observed low resolution is generated from the high resolution image X, according to the relation of Eqn. (4.15), where, $\mathbf{D}$ is the decimation matrix. Its size will be $M^2 \times M^4$. Y has a dimension $M^2 \times 1$ and X has a dimension $M^4 \times 1$. Typical structure of $\mathbf{D}$ is:

$$\mathbf{D} = \begin{pmatrix} \mathbf{C} & \mathbf{C} & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \mathbf{C} & \mathbf{C} & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathbf{C} & \mathbf{C} \end{pmatrix} \tag{4.19}$$

and each $\mathbf{C}$ is an $M \times M^2$ matrix. For the Daubechies wavelet (DAUB4) [20], $\mathbf{C}$ has the structure:

$$\mathbf{C} = \begin{pmatrix} c_1 & c_2 & c_3 & c_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_1 & c_2 & c_3 & c_4 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ c_3 & c_4 & 0 & 0 & 0 & 0 & c_1 & c_2 \end{pmatrix} \tag{4.20}$$

c_i's being DAUB4 (low-pass) wavelet filter coefficients [20]. The observation model of Eqn. (4.15) implies that we assume the given low resolution image is generated from one level wavelet transformed high resolution image, by retaining low-low frequency components.

As seen in the earlier section (sec. 3.4), we have to minimize the energy function $U_p(x)$. We have selected *a posteriori* energy function that is similar to the one used by Bhat *et al* [12], namely,

$$\begin{aligned} U_p(x) = \ & \|Y - \mathbf{D}X\|^2 + \\ & \textstyle\sum_{i,j}[\mu\{(x(i,j) - x(i,j-1))^2(1 - v1) + ((x(i,j) - x(i,j+1))^2(1 - v2) \\ & +((x(i,j) - x(i-1,j))^2(1 - h1) + ((x(i,j) - x(i+1,j))^2(1 - h2)\} \\ & +\gamma(v1 + v2 + h1 + h2)] \end{aligned} \tag{4.21}$$

where, parameter μ is the smoothness parameter and γ provides the penalty for discontinuities. $v1$, $v2$, $h1$ and $h2$ represent *vertical and horizontal line fields* as introduced by Geman and Geman [36]. They are given by $v1 = 1(|x_{i,j} - x_{i,j-1}| - \theta)$, such that $1(z) = 1$ if $z > 0$, θ being a constant threshold. $v2, h1$ and $h2$ are defined similarly. Simulation is carried out by minimizing Eqn. (4.21). We opted for simulated annealing [73] as it is guaranteed to converge in probability [36]. SA algorithm is initialized as follows: The given low resolution image is first interpolated using a simple linear interpolation method, namely, the gray value at pixel $(i, j+1)$ will be

$$x_{lin\_interpol}(i, j+1) = (x(i, j) + x(i, j+2))/2$$

But, the actual value of this pixel $x_{lin\_interpol}(i, j+1)$ may be anywhere between, and including, $x(i, j)$ and $x(i, j+2)$. To account for this, a new sample of x_{new} is generated from a Gaussian sampler, with variance $|(x(i, j) - x(i, j+1))/2|$. Then,

$$\hat{x}_{initial}(i, j+1) = x_{lin\_interpol}(i, j+1) + x_{new}. \tag{4.22}$$

This $\hat{x}_{initial}$ is used as the estimate in the SA algorithm for minimizing Eqn. (4.21). The above illustration is for interpolation along rows. A similar procedure is adopted for columns.For simulation purposes, in Eqn. (4.21) the following parameter values were used: $\lambda = 0.05$, $\mu = 950$ and θ =20. Stopping criterion was 500 iterations. The pseudo code (for row-wise) is given below.

```
BEGIN
    interpolate the given image x by linear interpolation method ;
    FOR all rows DO
        generate line fields v1,v2,h1 and h2 ;
        generate likelihood candidate for xnew(i,j) from a Gaussian sampler
          with variance |x(i,j-1) - x(i,j+1)|/2
        Using Eq.(4.21) calculate new energy function Up(new) with xnew(i,j);
        IF  Up(new) < Up(old)
            Accept new configuration  ;
        ELSE
          IF exp(Up(new) - Up(old)) > Uniform[0,1]
            accept new configuration ;
        REPEAT till convergence ;
    ENDFOR
END.
```

In the above algorithm, $U_p(old)$ is the the energy function calculated from Eqn. (4.21), with the old value of $x(i,j)$. $U_p(new)$ is the same energy function calculated with new value of x estimated as $\hat{x}_{initial}$ as per Eqn (4.22).

4.5 Proposed Method - 2 : MRA Based Approach

In this section we present a Multiresolution analysis (MRA) based approach for estimating the wavelet coefficients at higher (finer) scales. Some of these results have been reported earlier in[69, 72].

4.5.1 MRA method

The basic strategy for zooming is depicted in Fig. 4.2, where X_{avl} is the available low resolution image, while X_{unk} is the (unknown) high resolution image. H and L are appropriate high pass and low pass filters in the wavelet analysis. Using X_{avl}, we estimate the coefficients required for synthesizing the high resolution signal. Having estimated the coefficients, rest is a standard wavelet synthesis filter.

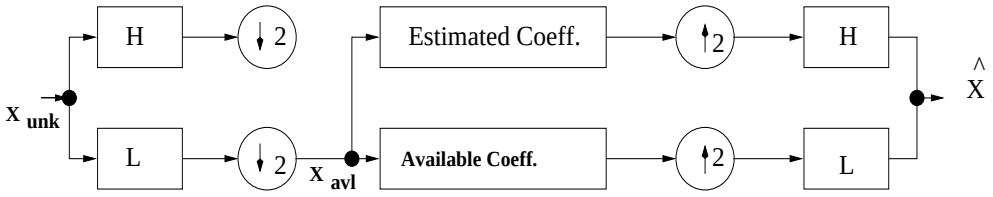


Figure 4.2: Basic Zooming Strategy

In order to illustrate the estimation of the coefficients in Fig. 4.2, consider Figure 4.3(a). We assume that a wavelet transform of an $M \times M$ image composed of boxes $0, I, II, IV, V, VII$ and $VIII$ is available and we want to zoom it to size $2M \times 2M$. This would be possible if we can estimate the wavelet coefficients in boxes III, VI and IX. Having estimated these wavelet coefficients, we simply feed these along with the $M \times M$ image to the wavelet based image synthesis filter bank (Fig. 4.3(b)) and obtain the interpolated (zoomed) image of size $2M \times 2M$. We exploit ideas from the zerotree concept [127] to estimate the wavelet coefficient in boxes III, VI and IX. The zero tree concept has the following properties:

- If a wavelet coefficient at a coarser scale is insignificant with respect to a given threshold θ, then all wavelet coefficients of the same orientation in same spatial location at finer scales are likely to be insignificant with respect to that θ.

- In a multiresolution system, every coefficient at a given scale can be related to a set of coefficients at the next coarser scale of similar orientation.

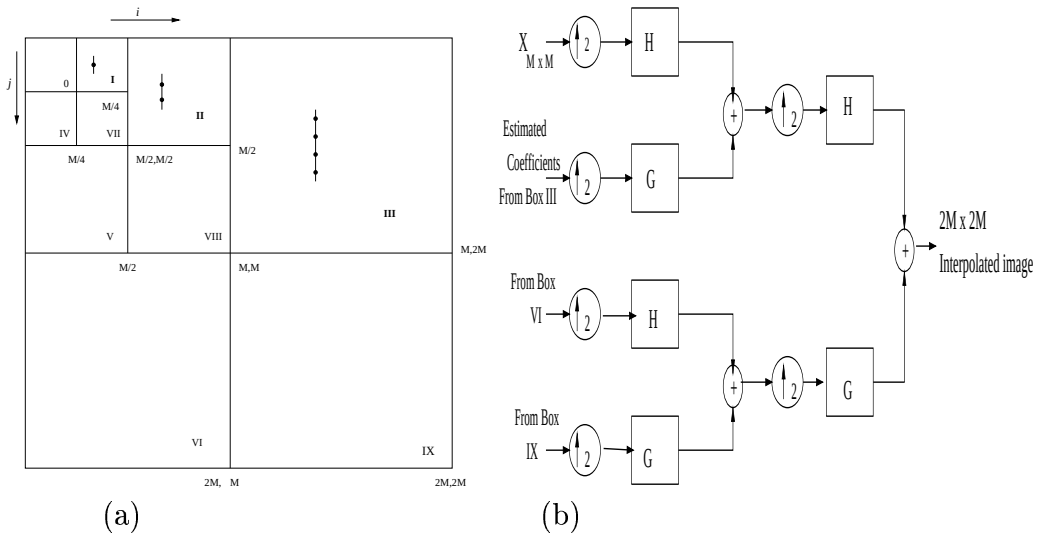


Figure 4.3: Zooming Process

To estimate the wavelet coefficients at the next finer level, namely, in boxes *III*, *VI* and *IX*, we find the significant wavelet coefficients at two resolutions (namely, in boxes $I - II$, $IV - V$, *VII* and *VIII*). For example, consider boxes I and II of Fig.4.3a, with significant coefficients shown as dots (on thin line). Denote the coefficients in respective boxes as $d_1(i_1, j_1) \in I$ and $d_2(i_2, j_2) \in II$. Note that, i_1, j_1 satisfy $M/4 \leq i_1 \leq (M/2) - 1$ and $0 \leq j_1 \leq (M/4) - 1$. Also, i_1 and i_2 are related by $i_1 = \lfloor i_2/2 \rfloor$ ($\lfloor . \rfloor$ represents the floor operator); j_1 and j_2 are similarly related. We have found through empirical studies that the ratio of the coefficients of finer scale (box II) the the next coarser scale (box I) remains almost invariant. We define $D_{(.)}(i, j)$ as (between boxes I and II):

$$D_1(i, j) = \frac{d_2(i, j)}{d_1(\lfloor i/2 \rfloor, \lfloor j/2 \rfloor)} \tag{4.23}$$

$$D_2(i, j) = \frac{d_2(i, j+1))}{d_1(\lfloor i/2 \rfloor, \lfloor (j+1)/2 \rfloor)} \tag{4.24}$$

These $D_{(.)}(i, j)$ values are used to estimate coefficients $\hat{d}$ at the finer scale (box *III*).

$$\begin{aligned} \hat{d}(2i, 2j) &= D_1(i, j) d_2(i, j)(1 - l_{d(i,j)}) \\ \hat{d}(2i, 2j+2) &= D_2(i, j) d_2(i, j+1)(1 - l_{d(i,j+1)}) \end{aligned} \tag{4.25}$$

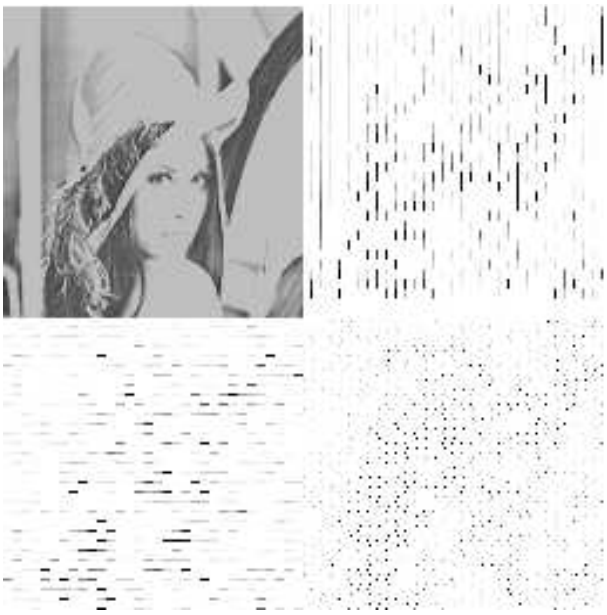


Figure 4.4: Estimated significant wavelet coefficients in box VI,IX and XI for the Lena image based on MRA zooming scheme

(a) (b)

Figure 4.5: (a) Original Boat Image, and (b) MRA Based Zoomed Image.

Figure 4.6: Sinc (top) and Spline Based (bottom) zoomed Boat Images

Figure 4.7: zoomed boat images. Left:Scaling function and Right: MRF based

$l_{d(i,j)}$ is an indicator function; $l_{d(i,j)}$ is set to zero, if $d(i,j)$ is significant, else to one. We set:

$$\begin{aligned} \hat{d}(2i, 2j+1) &= \hat{d}(2i, 2j) \\ \hat{d}(2i, 2j+3) &= \hat{d}(2i, 2j+2) \end{aligned} \tag{4.26}$$

We define $d(i,j)$ to be significant if $|d(i,j)| > \theta$. Note that Eqn. (4.25) implies an exponential decay and this is consistent with what is reported in [34, 84, 145]. Thus, we refer to $D_{(.)}(i,j)$ as the decay parameter.

In principle, for each coefficient $d_2(i,j) \in II$ we should have four coefficients in box *III*. But, our experiments have shown that doing this leads to a "blocky" zoomed image. Hence, we generate only *two* coefficients in box *III* corresponding to $d_2(i,j) \in II$. Moreover, we know that the detail sub-images using the wavelet transform yields vertical lines in boxes *I*, *II* and *III*; horizontal lines in boxes *IV*, *V* and *VI*; and diagonal lines in boxes *VII*, *VIII* and *IX*. We use this intuition and compute wavelet coefficients along vertical direction in box *III*, along horizontal direction in box *VI*, and along diagonal direction in box *IX* and that too, only along alternate lines. For box *III* equations (4.25) and (4.26) hold good. Analogous expressions can easily be obtained for wavelet coefficients in box *IV* and *IX*. Now, the estimated $\hat{d}$'s and the original $M \times M$ image is fed to the wavelet based image synthesizer (Fig 4.3b) to obtain the zoomed image which is of twice the size of the given image. In all our simulations the threshold θ was selected as half the maximum coefficient in the respective boxes, namely, boxes *II*, *V* and *VII*. Fig.4.4

shows an example of the estimated wavelet coefficients that were used in zooming of the Lena image. DAUB4 wavelet was used for computing the discrete wavelet transform. A pseudo-code for the above scheme follows:

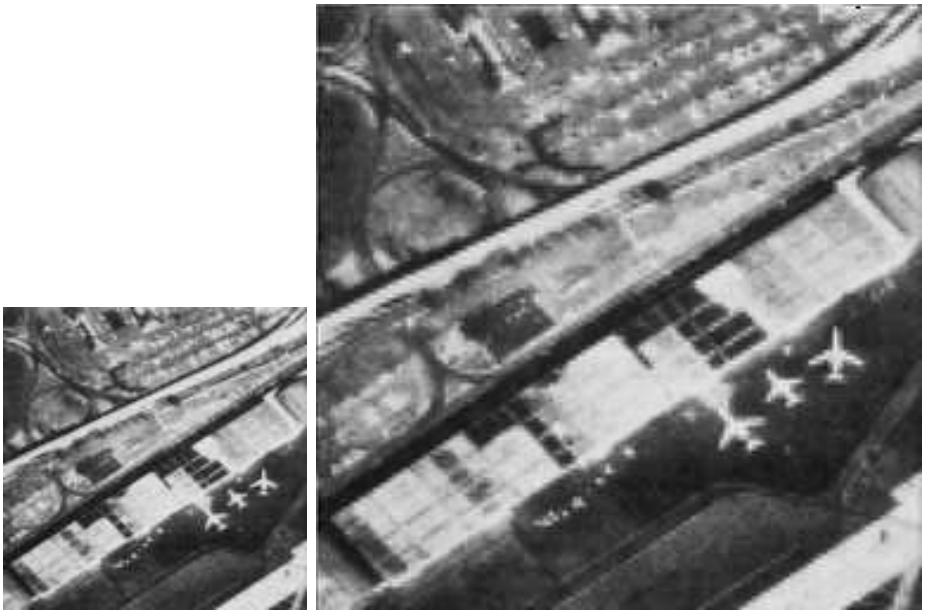
```
BEGIN
take two level wavelet transform of image(x) size M x M ;
  /* for box II of Fig. 4.3(a) */
      FOR i=M/2 TO M DO
      FOR j = 0 TO M/2 DO
          find the max coefficient in box II ;
      T= max/2 ;
      FOR i=M/2 TO M DO
      FOR j = 0 TO M/2 DO
      BEGIN
          IF (x[i][j] && x[i/2][j/2] > T )
              estimate wavelet coefficients for box II /* Eqn.  4.25 */
      END FOR
      /*  Repeat for boxes V and VII **/
      take 3-level inverse wavelet transform of x
                  to get 2M x 2M image ;
END.
```

Results from individual methods are shown in the Figures 4.5-4.10.

4.5.2 Joint MRA and MRF method

It is seen that the MRF approach gives comparatively smooth images and MRA gives sharper, but, slightly 'blocky' images. To achieve a good balance we propose a combined approach. Smoother part of the image is interpolated using MRF and more detailed part of the image is interpolated using MRA approaches.

The given low resolution image is partitioned into 32×32 blocks. For each of the blocks, the variance (V) is calculated, followed by the mean of these variances (MOV). Depending on the variance, each block is interpolated using either MRA or MRF. For a smoother part of the image, ($V < MOV$), we use MRF based interpolation according



(a) (b)

Figure 4.8: (a) Original Airport Image and (b) MRA based zoomed image

to Eqn.(4.21). Initial estimates of x are obtained using linear interpolation. We use the MRA interpolation method for detailed part of image, $(V > MOV)$. A pseudo code for joint MRF and MRA based method is:

```
BEGIN
    partition the image into 32x32 blocks ;
        FOR i= first_block TO last_block DO
        Evaluate the variance of this block v[i] ;
        mean of variance MOV=0;
for i = first_block TO Last_block DO
        MOV=MOV+v[i] ;
MOV = MOV / number_of_blocks ;
for i = first_block TO last_block DO
BEGIN
        IF (v[i] < MOV) THEN
   interpolate 32 x 32 block using MRF (section 4.4)
        ELSE interpolate the 32x32 block using MRA (section 4.5) method;
        END FOR
END.
```

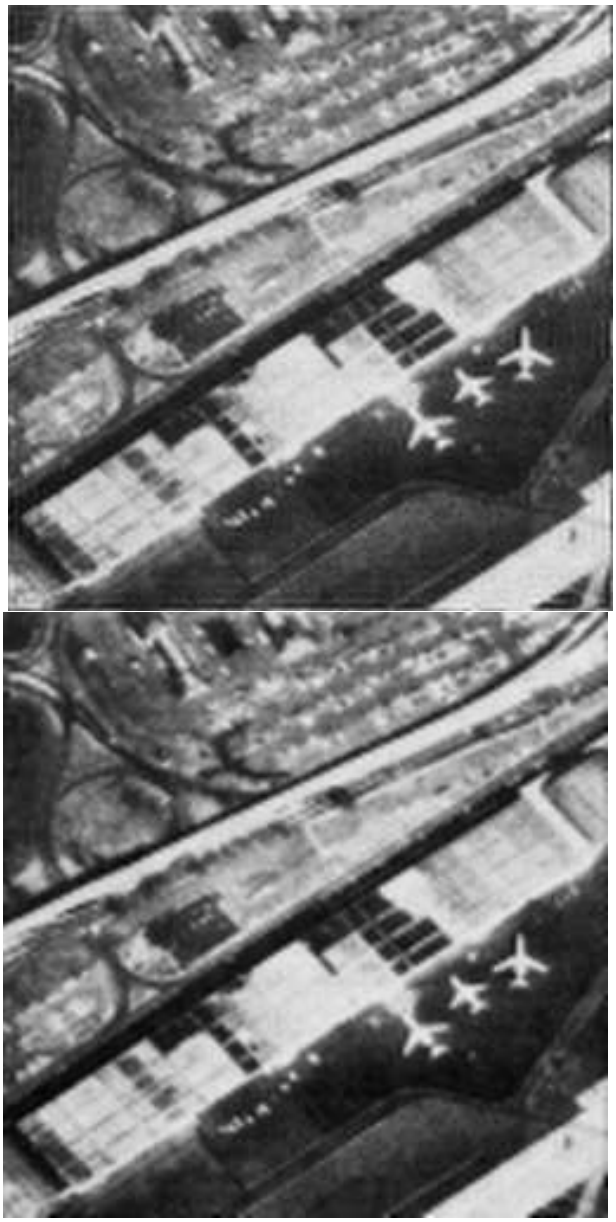


Figure 4.9: Sinc (top) and Spline based (bottom) zoomed Airport images

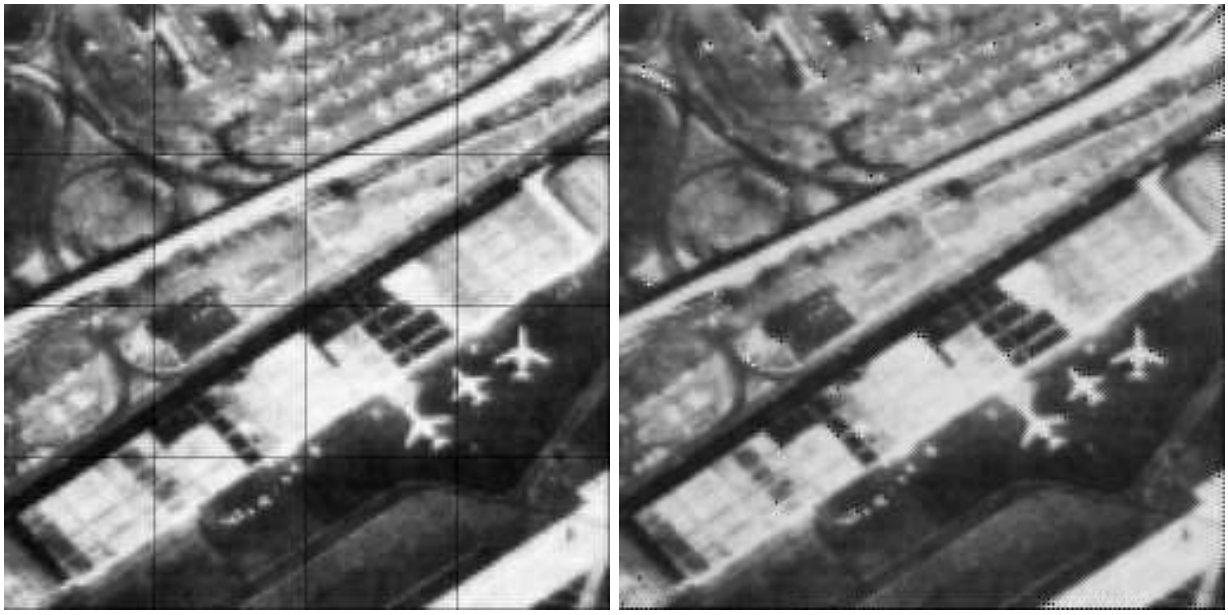


Figure 4.10: Zoomed airport images. Left:Scaling function based Right:MRF

4.6 Color Images

The proposed method is extended to color images. Many color coordinates are reported in literature and are in use [55, 100, 138, 154]. Most of the color coordinates are obtained as a transformation of other coordinates, usually the RGB. Here we discuss the results obtained from YIQ color coordinates. The Y component of the image is considered as the simple gray scale image and the proposed MRA algorithm was run on it. For I and Q components, pixel replication and linear interpolation give similar results as the other methods, due to the fact that these components are comparatively smooth. Keeping computational complexity in mind, we opted for linear interpolation for zooming I and Q components and MRA based method for zooming Y component. Results for the Suzie image are shown in Fig.s 4.13 - 4.18. We compare the proposed MRA method with Spline interpolation method. For both the methods, I and Q components are linearly interpolated and Y component alone is zoomed by the appropriate methods (MRA or spline). We can readily observe the proposed method performing better than Spline interpolation method. Performance can be improved by applying pre and/or post-processing.

Figure 4.11: Intermediate Results: MRF Model Based (top) and MRA Based (bottom)

Figure 4.12: Interpolated Using Joint MRA and MRF

Figure 4.13: Y component of Suzie image: Original and zoomed using MRA

Figure 4.14: I component of Suzie image: Original and zoomed using linear interpolation

Figure 4.15: Q component of Suzie image: Original and zoomed using linear interpolation

Figure 4.16: Color Image: Original and zoomed using MRA

Figure 4.17: Spline interpolated Suzie image Y component and color image

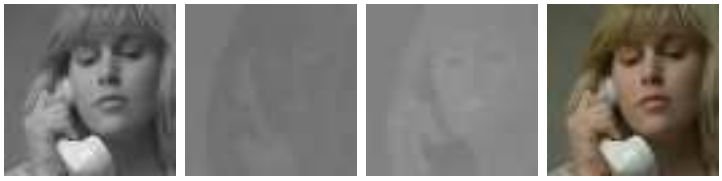


Figure 4.18: Low resolution Suzie image: Y,I,Q components and color image

Figure 4.19: Original Gujarat Coast Line

4.6.1 K-L Transform

The KL transform method is used for multi spectral images - color images, in our case. With a color image having RGB components, X of Eqn. (4.5) is three-dimensional. Then, the covariance and $\mathbf{\Phi}$ (Sections 2.2 and 4.2) matrices will be of size 3×3. The covariance matrix $\mathbf{C_x}$ is given by $\mathbf{C_x} = \frac{1}{N}\sum_{k=1}^{N} X_k X_k^T - m_x m_x^T$, where, m_x is the vector corresponding to the mean of the individual spectra. Now, we have a monochrome image Y generated from the color image X as per Eqn.(4.5). Once we have generated a monochrome image Y, as the principal component of color image X, interpolation of this Y is carried out using MRA as described in section (4.5). We can interpolate and retain only the principal component which results in an interpolated monochrome image from a multi-spectral image. An example of Gujarat coastline image[1] is shown in Fig.s 4.19 and 4.20. Alternatively, we can take the inverse KL transform and combine the three components to get a zoomed color image. In such a method, we see that there is a slight deterioration in the color contrast of the image. This can to be taken care of by proper post-processing. The pseudo code for this method is:

```
BEGIN
    read the RGB component of the image ;
    evaluate the mean for R, G and B component ;
    evaluate the covariance matrix, and -
       - eigenvectors of covariance matrix ;
    estimate the principle component as y
    interpolate the principle component y using MRA
END.
```

[1]Image ©SAC, Ahmedabad, India.

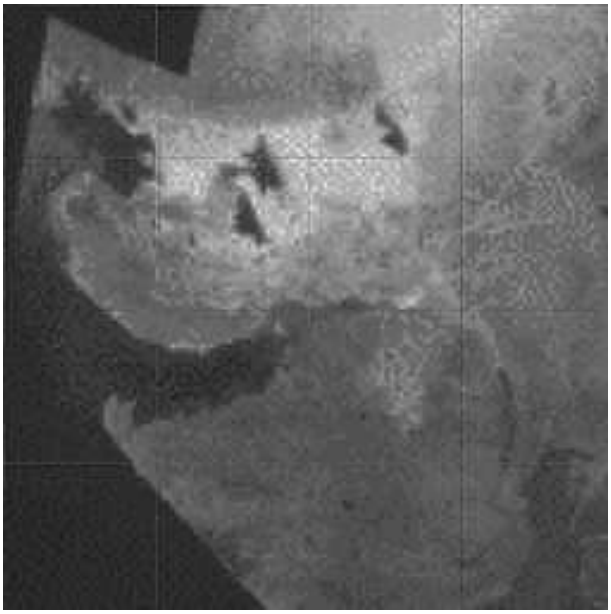


Figure 4.20: Zoomed Gujarat coast image, Using KL Transform (Principal Component)

4.7 Results and Discussion

The results of individual methods are illustrated for two image samples in Fig.s 4.5-4.10. First set of images Fig.4.5-4.7 shows the results for the boat image and the second set namely, Fig.s4.8-4.10, for the airport image. As expected, Sinc method produces ripples at the image boundaries, as is evident from Fig. 4.6(a) and 4.9(a). According to Fig. 4.6(b) and 4.9(b), visual qualities of Spline and Scaling function based methods are comparable, and seem to perform better. However, both of them smooth out the sharp edges of original low-resolution images. This can be observed at the rope which is tied to the tyre in boat image Fig. 4.5(b) and the road outside the airport 4.8(b). The MRF method generates a better set of images, but they are slightly smoothed, whereas MRA seems to do a better job of retaining sharp edges.

We evaluate the performance of these techniques by calculating Peak Signal to Noise Ratio (PSNR), which is defined as:

$$PSNR = \frac{255^2}{\sum_{i,j}(x(i,j) - \hat{x}(i,j))^2} \tag{4.27}$$

where, x is the original image and $\hat{x}$ is the zoomed image. The PSNR values are tabulated in Table(4.1).

Note: To calculate PSNR, a low resolution version of the high resolution image is zoomed. A low resolution image is generated according to the method proposed by Schultz

and Stevenson [121].

Table 4.1: PSNR Values

Image	*spline*	*Sinc*	*MRA*	*Scal.fn.*	*MRF*	*Joint*
Boat.	24.97	24.49	29.21	25.72	25.91	26.92
Airport	23.82	22.88	26.98	24.44	24.48	25.55
Lena	25.73	24.14	29.80	23.49	26.69	28.18
Bird	30.89	20.00	33.25	21.41	31.80	31.80
Einstein	28.28	19.18	30.17	24.51	30.17	28.87

We observe that the proposed method performs well for various classes of images. This is evident from the PSNR improvements, as per Table (4.1). Although a little amount of staircase effect is observed in the zoomed image from the proposed method, overall quality of zoomed image is good, as sharp edges of the original image are retained (ropes in boat image). To retain these sharp edges, we have used the DAUB4 wavelet. It is observed that higher order wavelets, like DAUB6, tends to smoothen the edges and the Haar wavelet produces more of the staircase effect.

For color images, it is seen that color contrast deteriorates slightly while operating on RGB color coordinates. This has to be taken care of by suitable post-processing. In this regard, YIQ color coordinate gives satisfactory results, without the need for pre- or post-processing.

We compare the wavelet coefficients obtained by the proposed MRA method with those obtained by the scaling function-based method. Results of obtaining coefficients from these two methods are shown in Fig. 4.21. It is observed that coefficients obtained from both the methods are almost the same, and hence, the proposed method is justified. The proposed method is computationally less taxing than the scaling function based method, and hence faster. Visually, the scaling function based method performs slightly better.

For zooming up to 8 times (8×), the output suffers from the staircase effect. For such cases spline does a better job. Another limitation with the proposed technique is the presence of spurious edges, when there is a rapid change in the gray levels. For checkerboard kind of images, these spurious edges are more dominant. These spurious edges can be attributed to the insertion of high frequency components in the image. This effect can be overcome by post-processing techniques. For a relatively smooth image, the proposed method performs very well.

Instead of DWT we can use DCT to zoom a given image. Method then will be similar

to sinc interpolation. That is we take the DCT of the given image, pad it with zeros and take IDCT of twice the size. Here again, we observe that zoomed images will have ripples and the performance is similar to the sinc interpolation. A sample zoomed image obtained by zero padding in DCT domain is shown in Fig. 4.22

4.8 Conclusion

We reviewed some of the techniques for image interpolation, paying special attention to wavelet based zooming methodologies. The basic idea of image zooming in wavelet domain is to estimate coefficients at the finer scale. We overviewd some of the techniques reported in literature, to estimate these coefficients.

We proposed a new computationally efficient scheme. As the proposed scheme is efficient, it can be used for real-time applications (application for video is discussed in Chapter 6). We mentioned the advantages and limitations of the proposed scheme.

Comparing the performance of the proposed technique with a few of the conventional approaches, we observe that output images are sharper and there is a good improvement of PSNR up-to about 3dB. Thus, the proposed method performs better than conventional approaches, both visually and numerically.

Figure 4.21: Estimated wavelet coefficients: (a) MRA method (b) Scaling function based method

Figure 4.22: Boat image zoomed by zero padding in DCT domain. Note the ripples near the ropes