ERG4920CT

Thesis II

IMAGE DEBLURRING

Image Upsampling via Tight Frames

By

LEE Sing Chun (06611574)

Supervised

By

Prof. Raynomd Hou-fu CHAN

A FINAL YEAR PROJECT REPORT
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF BACHELOR OF INFORMATION ENGINEERING

Faculty of Engineering
Department of Information Engineering
The Chinese University of Hong Kong

May 2010

Declaration of Authorship

I, LEE Sing Chun (06611574), declare that this thesis titled 'IMAGE DEBLURRING: Image Upsampling via Tight Frames' submitted is original except for source material explicitly acknowledged. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the website http://www.cuhk.edu.hk/policy/academichonesty/

Signed: _____

Date: _____

Student Name: LEE Sing Chun

Student ID: 06611574

Course Code: ERG4920CT

Course Title: Thesis II

THE CHINESE UNIVERSITY OF HONG KONG

Abstract

Faculty of Engineering
Department of Information Engineering

Bachelor of Information Engineering

IMAGE DEBLURRING
Image Upsampling via Tight Frames

by LEE Sing Chun

In this thesis a new parameter-free algorithm for automatically upsampling images is proposed which is efficient to preserve natural detail and reduce Moire artifacts of input images. Multi-resolution analysis via Tight Frames is utilized in the algorithm to spontaneously correct the high frequency band data that correspond to the natural detail. The algorithm extends the image inpainting model of [1] and interprets the input image as the known data in the transformed domain. In addition, the thresholding in [1] is abandoned, yet results of the algorithm are not dominated by noise because of a good initial guess. On the contrary, they demonstrate the production of better visual perception images of this algorithm. On top of this, the algorithm becomes parameter-free because thresholding is not involved. It makes the algorithm more user-friendly that users need not to adjust any parameters. Image upsampling is done completely automatically. Finally, video upsampling is also performed by applying this algorithm frames by frames and compare with those from [2].

Contents

Chapter 1

Introduction

Reconstructing a high resolution image from a low resolution image, namely image upsampling, is a fundamental image operation that is nearly found in all image editing software. Recently, new high resolution display devices also introduce upsampling technique to adapt input images or footage for their high resolution output. To upsample an image, more unknown pixels have to be determined under additional assumptions on the desired high resolution image. It is a remarkably challenging problem and has been being studied for many years.

Varied upsampling algorithm assume different image characteristic. For instance, polynomial interpolation upsampling like bicubic upsampling presumes the image is smooth enough so that the image can be well-approximated by polynomials as in [3] and [4]. Example-based upsampling suggests the texture or detail content of the image is closely matched with the example images in the database such as [5], [6], [7] and [8]. These assumptions are considerably inappropriate in some cases. The upsampled image either is too blurry which is even worse than results of polynomial interpolation, or lose its original nature which results an artificially composite image. Moreover, an automatic segmentation is definitely required to clip the texture or detail content out of the image. A good segmentation algorithm that is suitable for all images is hardly found. Tight Frame is thereby introduced to perform upsampling in this thesis so as to preserve natural details of the upsampled image such as [9].

In this thesis, multi-resolution analysis via Tight Frames and iterative method are utilized. Through this analysis, the image domain is decomposed by a series of filter bands to form one set of nested disjointed intervals at which each interval represent an image domain with certain frequencies as well as a band-limited image domain. The input image is considered to be known datum of a low resolution image. This low resolution image and the desired high resolution image are in two different set of intervals from the analysis; therefore, the input and output images can be related by this analysis. On top of this, based on the perfect reconstruction identity of this analysis and its corresponding

synthesis, an iterative algorithm for image upsampling is developed. The algorithm is briefly described in the following (figure 1.1). More details will be discussed later.

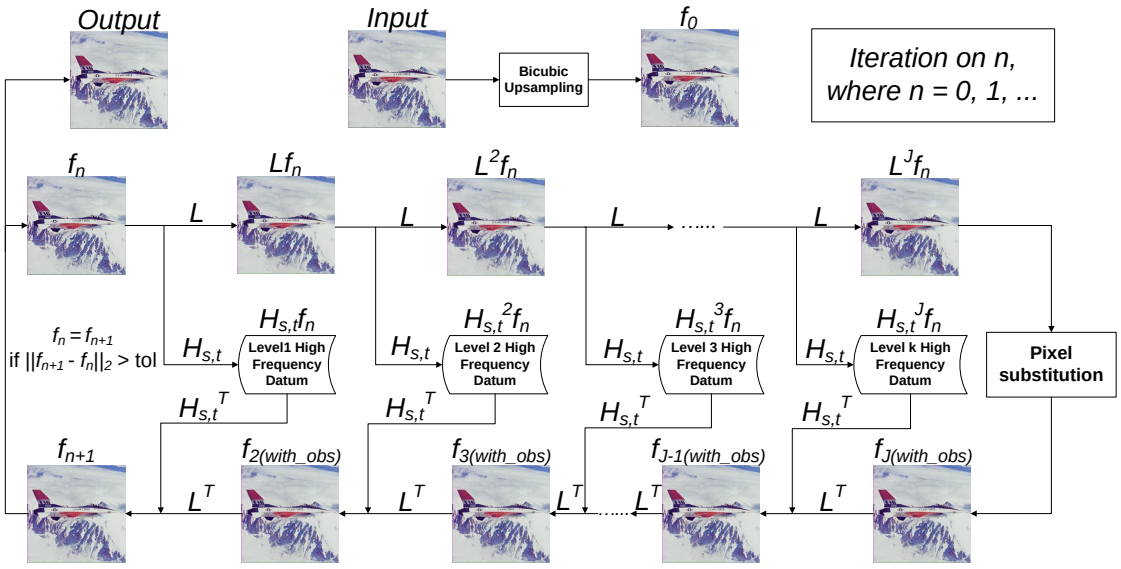

FIGURE 1.1: Schematic diagram of the proposed algorithm:

In addition, the input image of the algorithm is assumed to be obtained by any photographic devices and without any digital adaption. A gray-scale image is always taken as the input of the algorithm. However, for real-world images, mostly they are of RGB scale. To adapt to the algorithm, the RGB scale is converted to YUV colour space and then the Y channel is upsampled by the algorithm while the U and V channels are upsampled by bicubic interpolation. Similar to [2], the algorithm for colour images upsampling concentrates on the human perception rather than image chrominance. Besides, all the images mentioned in this thesis are assumed to be stacked column by column as a column vector.

This thesis is organized as follows. A brief review of related literature is provided in chapter 2. In chapter 3, there are three sections. First section is about multi-resolution analysis via Tight Frames. Detail of proposed algorithm is described in the following section. The last section gives the mathematical analysis of the algorithm. In chapter 4, the results of the algorithm is presented and compared with those from [2]. Finally, discussion, summary and future research direction are described in chapter 5.

Chapter 2

Related Work

In the past few decades, scientists who has been researching in computer graphics or image processing has been paying more and more attention on image upsampling. Accordingly, many upsampling algorithms have been suggested with their own distinctive ideas, prior assumptions and additional information of images. The image formation model given by [10] provides a typical approach to model image upsampling. In this model several low resolution images are obtained to reproduce a high resolution image. Similar to others, our algorithm is also based on this model; however, it is modified to accept only one low resolution image as the input and then reproduce the high resolution image. That is also called single-image upsampling.

The traditional single-image upsampling is by polynomial interpolations such as bilinear, bicubic and [3]. The interpolation upsampling is simple to implement and always efficient in the sense of the running speed; therefore, it is widely used by commercial image/video processing software. However, this upsampling technique always gives inappropriate resulting images because of its assumption of images that presumes images are spatially smooth or band-limited. For real world images, they always contain sharp edges as well as high frequency textures. As a result, this smoothness assumption certainly makes the upsampled version of real world images become blurring, blocking or aliasing. This artifacts and details of interpolation upsampling can be referred to [4].

Another main approach of single-image upsampling is by example-based or patch-based non-parametric image models. This model was first proposed by [11] [12], also known as the "Image Analogies" framework, and then was further developed by [5]. The model assumes the high frequency bands (that is sharp edges and textures) of the desired high resolution image can be predicted by some example patches in a learning-database. These example patches are split into low frequency bands and high frequency bands, and only the low frequency bands of them are stored in the database. For any input image, one can match it with the low frequency bands in the database so as to find out the corresponding high frequency bands. The missing high frequency bands of

the input image are thereby predicted like this, and these predicted bands are used to upsample the image. This approach is really able to produce an upsampled image with reasonable sharp edges and textures; however, if there are no relevant patch information in the database, it seems impossible to upsample the image correctly. In this case, the upsampled image is fairly unpredictable and may even worse than the input. Besides this, another drawback of this approach is the relatively high computation time. Since the most optimal example patch among all in the database has to be found for every segment of textures, it is remarkably time expensive. Similar approaches with different refinements can be found in [6], [7] and [8].

Recently, natural image statistics stimulate researchers' minds to design image upsampling algorithm. Different statistic distribution of natural images are considered to be utilized in image upsampling as prior information of the high resolution image. For example, [13] and [14] use analytical Markov field models and employ the statistical edge information in constraining the image upsampling problem. This approach is success to reproduce the sharp edges of the image but it still fails to preserve small-scaled texture of the image. However, it is still a newly attempted approach and certainly has lots of room to improve and more interesting is this approach is somehow in between the typical approach and the database approach. It has the prior information like what the database approach has and also has the advantage of fast implementation as the typical approach does. Besides, [2] also utilizes the natural image gradient distribution in its total variational approach algorithm to solve the minimization problem. Our proposed algorithm in this thesis has not used the natural image statistics so far; nevertheless, it is reasonably an important future research direction that how to constrain the high frequency bands in our algorithm by the natural image statistics.

Chapter 3

Proposed Algorithm

In this chapter, some preliminaries of tight frames and multi-resolution analysis are presented. For simplicity, we only consider the one dimensional cases. The two dimensional cases can be constructed by tensor product of those in one dimensional cases. One can see more details in [15], [16] and [17]. Furthermore, the thigh frame image upsampling algorithm is also proposed in this chapter with mathematical analysis.

3.1 Tight Frames

A system $X \subseteq L_2(\mathbb{R})$ is called a *tight frames* of $L_2(\mathbb{R})$ if

$$f = \sum_{g \in X} \langle f, g \rangle \, g, \qquad \forall f \in L^2(\mathbb{R}).$$

Define X_Ψ as the collection of all dilations and translations of a finite set $\Psi \in L_2(\mathbb{R})$ by

$$X_\Psi := \left\{ 2^{k/2} \psi(2^k x - j) \mid \psi \in \Psi; k, j \in \mathbb{Z} \right\}.$$

If X_Ψ is a tight frames with compact support, then one can construct it by starting with a compactly supported refinable function $\phi \in \text{Ł}_2(\mathbb{R})$, also known as scaling function, by satisfying the dilation equation:

$$\phi(x) = \sum_{n=-l_1}^{l_2} \sqrt{2} h_\phi(n) \phi(2x - n) \qquad \forall x \in \mathbb{R}, \tag{3.1}$$

where $h_\phi(n)$ is the coefficients of a low-pass filter with length $(l_2 - l_1)$. It is equivalent to satisfy the refinement equation:

$$\hat{\phi}(2\omega) = H_\phi(\omega) \hat{\phi}(\omega) \qquad \forall \omega \in \mathbb{R},$$

where $\hat{\phi}$ is the Fourier transform of ϕ and $H_\phi(\omega)$ is the discrete Fourier transform of $\sqrt{2}h_\phi(n)$ with $H_\phi(0) = 1$.

After finding the refinable function ϕ, one can construct the tight frames system by satisfying the similar dilation equation:

$$\psi(x) = \sum_{n=-s_1}^{s_2} \sqrt{2}h_\psi(n)\phi(2x-n) \qquad \forall x \in \mathbb{R}, \tag{3.2}$$

where $h_\psi(n)$ is the coefficients of a high-pass filter with length $(s_2 - s_1)$. It is also equivalent to satisfy the following equation in Fourier domain:

$$\hat{\psi}(2\omega) = H_\psi(\omega)\hat{\phi}(\omega) \qquad \forall \omega \in \mathbb{R},$$

where $\hat{\psi}$ is the Fourier transform of ψ and $H_\psi(\omega)$ is the discrete Fourier transform of $\sqrt{2}h_\psi(n)$.

From [18], the unitary extension principle (UEP) implies that X_Ψ which is generated by a finite set Ψ forms a tight frame of $L_2(\mathbb{R})$ if $H_\phi(\omega)$ and $\{H_\psi(\omega)\}_{\psi\in\Psi}$ satisfy:

$$H_\phi(\omega)\overline{H_\phi(\omega+k\pi)} + \sum_{\psi\in\Psi} H_\psi(\omega)\overline{H_\psi(\omega+k\pi)} = \delta_{k,0}, \qquad k = 0, 1 \tag{3.3}$$

for almost all $\omega \in \mathbb{R}$.

As a result, one can construct the tight frame system by finding $L_2(\mathbb{R})$ if $H_\phi(\omega)$ and $\{H_\psi(\omega)\}_{\psi\in\Psi}$ such that they satisfy (3.3). In this thesis, the tight frame used is constructed by piecewise linear B-spline with low-pass filter coefficients $h_0 = \frac{1}{4}[1, 2, 1]$ and high pass filter coefficients $h_1 = \frac{\sqrt{2}}{4}[1, 0, -1]$ and $h_2 = \frac{1}{4}[-1, 2, -1]$. The corresponding scaling function ϕ and wavelet functions ψ_1 and ψ_2 are shown in Figure (3.1).

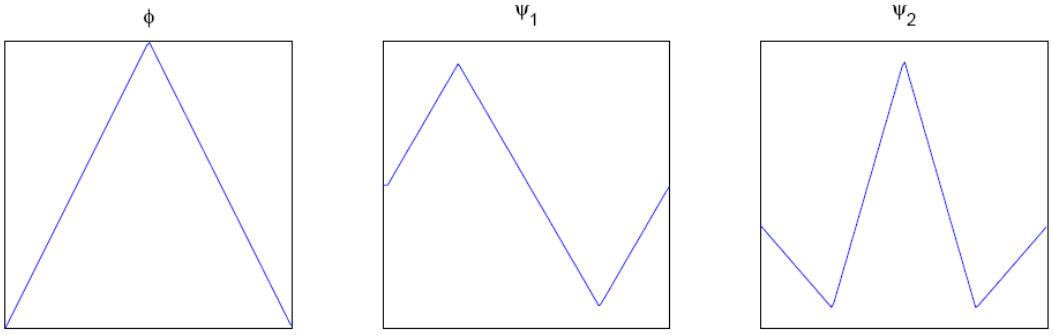

FIGURE 3.1: Scaling and wavelet functions

The two dimensional tight frame system used in the proposing algorithm is thus given by the tensor product of h_0, h_1 and h_2, i.e. $h_{s,t}$ = tensor product of (h_t, h_s) where $s = 0, 1, 2$ and $t = 0, 1, 2$.

In this thesis, the convolution of filters will be represented by matrix equations with their corresponding filter matrices. Furthermore, periodic boundary will be used here. For example, the filter matrix H_0 corresponds to the filter h_0 and $H_{0,0}$ associated to $h_{0,0}$ are:

$$H_0 = \frac{1}{4}\begin{bmatrix} 2 & 1 & 0 & \cdots & 0 & 1 \\ 1 & 2 & 1 & \ddots & 0 & 0 \\ 0 & 1 & 2 & \ddots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & 2 & 1 \\ 1 & 0 & 0 & \cdots & 1 & 2 \end{bmatrix} \quad \text{and} \quad H_{0,0} = H_0^T \otimes H_0.$$

and we have the following perfect reconstruction identity:

$$\sum_{s,t=0}^{2} H_{s,t}^T H_{s,t} = \textit{identity matrix} \tag{3.4}$$

3.2 Multi-Resolution Analysis

According to [16] and [17], the tight frame system mentioned above forms a multi-resolution analysis (MRA). First, consider the collection of all translations of the scaling function $\phi \in L_2(\mathbb{R})$ with dilation $k \in \mathbb{Z}$.

$$\Phi_k := \left\{ 2^{k/2}\phi(2^k x - j) \mid j \in \mathbb{Z} \right\}.$$

Let $V_k := \text{clos}_{L_2(\mathbb{R})}\Phi_k$. Since every $\phi_k \in V_k$ also satisfy (3.1), i.e.

$$\begin{aligned} \phi_k &= 2^{k/2}\phi(2^k x - j) \\ &= \sum_n \sqrt{2} h_\phi(n) 2^{k/2}\phi(2^{k+1}x - 2j - n) \\ &= \sum_n h_\phi(n) 2^{(k+1)/2}\phi(2^{k+1}x - 2j - n) \\ &= \sum_m h_\phi(m - 2j) 2^{(k+1)/2}\phi(2^{k+1}x - m) \end{aligned}$$

which implies ϕ_k is generated by $\left\{2^{(k+1)/2}\phi(2^{k+1}x - m)\right\}_{m\in\mathbb{Z}}$, hence $\phi_k \in V_{k+1}$ and $V_k \subset V_{k+1}$. As a result, we have a nested vector spaces spanned by scaling functions:

$$\{0\} \subset \cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset L_2(\mathbb{R}).$$

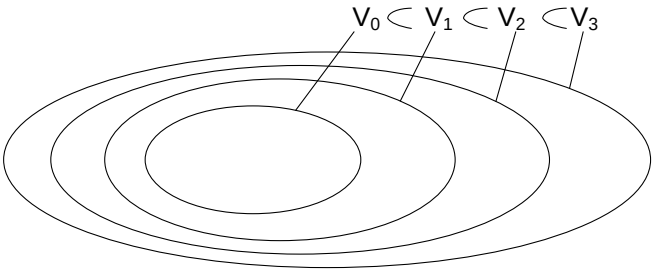

FIGURE 3.2: Nested vector spaces spanned by scaling functions

Next, consider the collection of all translations of the wavelet function $\psi \in L_2(\mathbb{R})$ with dilation $k \in \mathbb{Z}$.

$$\Psi_k := \left\{ 2^{k/2}\psi(2^k x - j) \mid j \in \mathbb{Z} \right\}.$$

Similarly, let $W_k := \text{clos}_{L_2(\mathbb{R})}\Psi_k$ and because of (3.2). We have, for any $\psi_k \in W_k$,

$$\begin{aligned} \psi_k &= 2^{k/2}\psi(2^k x - j) \\ &= \sum_n \sqrt{2} h_\psi(n) 2^{k/2} \phi(2^{k+1}x - 2j - n) \\ &= \sum_m h_\psi(m - 2j) 2^{(k+1)/2} \phi(2^{k+1}x - m) \end{aligned}$$

Hence, $\psi_k \in V_{k+1}$ and by the orthogonality of scaling and wavelet functions, $W_k \perp V_k$, which implies $V_{k+1} = W_k \oplus V_k$. As a result, we have

$$L_2(\mathbb{R}) = \cdots \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus W_2 \oplus \cdots .$$

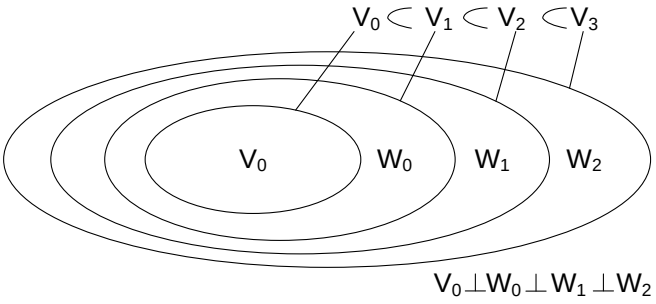

FIGURE 3.3: Nested vector spaces spanned by scaling functions and wavelet functions

3.3 Image Upsampling via Tight Frames

In this section, we will first device the upsampling algorithm via Tigh Frames and then state clearly the step to upsample image in our algorithm.

3.3.1 Device the Algorithm

Assume the desired high resolution image, $f_{desired}$, is band limited in $L_2(\mathbb{R})$; therefore, $L_2(\mathbb{R})$ can be well-approximated by V_J for some $J \in \mathbb{Z}$ in the sense of representing the image. For simplicity, assume also the low resolution image, $\mathbf{f}_{low}$, is in V_0. Thought the multi-resolution analysis,

$$V_J = W_J \oplus W_{J-1} \oplus \cdots \oplus W_0 \oplus V_0.$$

Therefore, $\mathbf{f}_{desired}$ is split into high frequency parts in W_k, $0 \le k \le J$, and $\mathbf{f}_{low}$ in V_0, by the analysis of the filters $h_{s,t}$, $s,t = 0,1,2$. Analogue to [15], utilizing the perfect reconstruction identity (3.4), we have

$$\begin{aligned}
\mathbf{f}_{desired} &= \sum_{s,t=0}^{2} H_{s,t}^T H_{s,t} \mathbf{f}_{desired} \\
&= H_{0,0}^T H_{0,0} \mathbf{f}_{desired} + \sum_{\substack{s,t=0 \\ (s,t)\neq(0,0)}}^{2} H_{s,t}^T H_{s,t} \mathbf{f}_{desired} \\
&= (H_{0,0}^T)^2 (H_{0,0})^2 \mathbf{f}_{desired} + \sum_{j=0}^{1} \sum_{\substack{s,t=0 \\ (s,t)\neq(0,0)}}^{2} (H_{0,0}^T)^j H_{s,t}^T H_{s,t} (H_{0,0})^j \mathbf{f}_{desired} \\
&= \cdots \\
&= (H_{0,0}^T)^J (H_{0,0})^J \mathbf{f}_{desired} + \sum_{j=0}^{J-1} \sum_{\substack{s,t=0 \\ (s,t)\neq(0,0)}}^{2} (H_{0,0}^T)^j H_{s,t}^T H_{s,t} (H_{0,0})^j \mathbf{f}_{desired}.
\end{aligned}$$

Notice $(H_{0,0})^J \mathbf{f}_{desired}$ is the low resolution image in V_0, thus $(H_{0,0})^J \mathbf{f}_{desired} = \mathbf{f}_{low}$. So,

$$\begin{aligned}
\mathbf{f}_{desired} &= (H_{0,0}^T)^J (H_{0,0})^J \mathbf{f}_{desired} + \sum_{j=0}^{J-1} \sum_{\substack{s,t=0 \\ (s,t)\neq(0,0)}}^{2} (H_{0,0}^T)^j H_{s,t}^T H_{s,t} (H_{0,0})^j \mathbf{f}_{desired} \\
&= (H_{0,0}^T)^J \mathbf{f}_{low} + \sum_{j=0}^{J-1} \sum_{\substack{s,t=0 \\ (s,t)\neq(0,0)}}^{2} (H_{0,0}^T)^j H_{s,t}^T H_{s,t} (H_{0,0})^j \mathbf{f}_{desired} \\
&= (H_{0,0}^T)^J \mathbf{f}_{low} + T(\mathbf{f}_{desired}),
\end{aligned}$$

where

$$T(\mathbf{f}_{desired}) := \sum_{j=0}^{J-1} \sum_{\substack{s,t=0 \\ (s,t)\neq(0,0)}}^{2} (H_{0,0}^T)^j H_{s,t}^T H_{s,t} (H_{0,0})^j \mathbf{f}_{desired}$$

Let $\mathbf{f}_{input}$ be the input image with size half of $\mathbf{f}_{desired}$. We are going to upsamle $\mathbf{f}_{input}$ twice in size. That is to reconstruct $\mathbf{f}_{desired}$ from $\mathbf{f}_{input}$. Now, introduce the downsample

operators $D_{m,n}$, $m,n = 0,1$, as defined in [10]. Then, $\mathbf{f}_{input} = D_{0,0}\mathbf{f}_{low}$, and hence

$$\begin{aligned}
\mathbf{f}_{desired} &= (H_{0,0}^T)^J \sum_{m,n=0}^{1} D_{m,n}^T D_{m,n}\mathbf{f}_{low} + T(\mathbf{f}_{desired}) \\
&= (H_{0,0}^T)^J D_{0,0}^T D_{0,0}\mathbf{f}_{low} + (H_{0,0}^T)^J \sum_{\substack{m,n=0 \\ (m,n)\neq(0,0)}}^{1} D_{m,n}^T D_{m,n}\mathbf{f}_{low} + T(\mathbf{f}_{desired}) \\
&= (H_{0,0}^T)^J D_{0,0}^T \mathbf{f}_{input} + (H_{0,0}^T)^J \sum_{\substack{m,n=0 \\ (m,n)\neq(0,0)}}^{1} D_{m,n}^T D_{m,n}\mathbf{f}_{low} + T(\mathbf{f}_{desired}),
\end{aligned}$$

where the replacement of $D_{0,0}\mathbf{f}_{low}$ with $\mathbf{f}_{input}$ is called pixel substitution. Therefore,

$$\begin{aligned}
\mathbf{f}_{desired} &= (H_{0,0}^T)^J D_{0,0}^T \mathbf{f}_{input} + (H_{0,0}^T)^J \sum_{\substack{m,n=0 \\ (m,n)\neq(0,0)}}^{1} D_{m,n}^T D_{m,n}(H_{0,0})^J \mathbf{f}_{desired} + T(\mathbf{f}_{desired}) \\
&:= (H_{0,0}^T)^J D_{0,0}^T \mathbf{f}_{input} + P(\mathbf{f}_{desired}). \qquad (3.5)
\end{aligned}$$

Thus, in order to reconstruct $\mathbf{f}_{desired}$ from $\mathbf{f}_{input}$, we suggest to iterate on it as:

$$\mathbf{f}_{n+1} = (H_{0,0}^T)^J D_{0,0}^T \mathbf{f}_{input} + P(\mathbf{f}_n). \qquad (3.6)$$

3.3.2 Proposed Algorithm

Finally, the step of using our image upsampling algorithm (Figure 1.1) is:

1. Set an initial guess, $\mathbf{f}_0$, as the bicubic upsampling of $\mathbf{f}_{input}$ and *tol* as the tolerance allowed.
2. Set $\mathbf{f}_n = \mathbf{f}_0$.
3. Find $\mathbf{f}_{n+1}$ by the equation (3.6) by these processes.
   (a) Analysis $\mathbf{f}_n$ to level J by Thigh Frame $H_{s,t}$.
   (b) Perform pixel substitution with $\mathbf{f}_{input}$ to $(H_{0,0})^J\mathbf{f}_n$, i.e. $D_{0,0}(H_{0,0})^J\mathbf{f}_n = \mathbf{f}_{input}$.
   (c) Synthesis them by $H_{s,t}^T$ to find $\mathbf{f}_{n+1}$.
4. Set $\mathbf{f}^* = \mathbf{f}_{n+1}$ if $\|\mathbf{f}_{n+1} - \mathbf{f}_n\|_2 \leq tol$, where $\mathbf{f}^*$ is our upsampled image; else set $\mathbf{f}_n = \mathbf{f}_{n+1}$ and repeat step 3 and 4.

In conclusion, our algorithm is able to upsample a low resolution image in V_0 to a high resolution image in V_J by iteration on the perfect reconstruction identity of J level multi-resolution analysis via Tight Frames, where J is the lowest positive integer such that the desired high resolution image can be well-approximated in V_J.

3.4 Mathematical Analysis

In this section, the convergence of (3.6) is analyzed. First of all, let

$$A_J = \begin{bmatrix} (H_{0,0})^J \\ H_{0,1}(H_{0,0})^{J-1} \\ H_{1,0}(H_{0,0})^{J-1} \\ H_{1,1}(H_{0,0})^{J-1} \\ \vdots \\ \vdots \\ H_{0,1} \\ H_{1,0} \\ H_{1,1} \end{bmatrix} \quad \textbf{and} \quad P_J = \begin{bmatrix} & & & & \\ D_{0,0}^T D_{0,0} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \mathbf{O} & \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \underbrace{\mathbf{O} \quad \mathbf{O} \quad \mathbf{O} \quad \vdots \quad \mathbf{O}}_{3(J-1)+1 \textbf{ times}} \end{bmatrix},$$

where $D_{0,0}$ is the downsampling matrix as described above and $\mathbf{O}$ is the zero matrix with the same dimension as $H_{s,t}$. i.e. (3.5), without substituting $\mathbf{f}_{input}$ and $\mathbf{f}_{low}$, can be written as

$$\mathbf{f}_{desired} = A_J^T \left(P_J A_J \mathbf{f}_{desired} + (\mathbf{I} - P_J) A_J \mathbf{f}_{desired} \right). \tag{3.7}$$

Then, based on our definitions of A_J and P_J, for any given $\mathbf{f}_{input}$, we can always find $\mathbf{x}$ defined as

$$\mathbf{x} = \begin{bmatrix} D_{0,0}^T \mathbf{f}_{input} \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix},$$

such that

$$P_J \mathbf{x} = P_J A_J \mathbf{f}_{desired}.$$

Hence, (3.7) can be written as

$$\mathbf{f}_{desired} = A_J^T \left(P_J \mathbf{x} + (\mathbf{I} - P_J) A_J \mathbf{f}_{desired} \right),$$

and our algorithm (3.6) can be written as

$$\mathbf{f}_{n+1} = A_J^T \left(P_J \mathbf{x} + (\mathbf{I} - P_J) A_J \mathbf{f}_n \right), \tag{3.8}$$

where $\mathbf{x}$ is found by the given low resolution image $\mathbf{f}_{input}$.

Similar to [19], write (3.8) as

$$\begin{cases} \mathbf{y}_n &= P_J \mathbf{x} + (\mathbf{I} - P_J) A_J \mathbf{f}_n, \\ \mathbf{f}_{n+1} &= A_J^T \mathbf{y}_n, \end{cases}$$

and define the set S as

$$S := \{\mathbf{y} | P_J \mathbf{y} = P_J \mathbf{x}\},$$

which is closed, nonempty and convex. Therefore, $\mathbf{y}_n$ can be viewed as the minimizer of the constrained minimization problem

$$\mathbf{y}_n = \arg\min_{\mathbf{y} \in S} \left\{ \frac{1}{2} \|A_J \mathbf{f}_n - \mathbf{y}\|_2^2 \right\} \tag{3.9}$$

By adding the indicator function i_S for the convex set S, which is defined as

$$i_S(\mathbf{y}) = \begin{cases} 0, & \text{if } \mathbf{y} \in S, \\ +\infty, & \text{otherwise,} \end{cases}$$

the constrained minimization (3.9) can be written as an unconstrained minimization:

$$\mathbf{y}_n = \arg\min_{\mathbf{y} \in S} \left\{ \frac{1}{2} \|A_J \mathbf{f}_n - \mathbf{y}\|_2^2 + i_S(\mathbf{y}) \right\} \tag{3.10}$$

Besides, we can also write $\mathbf{f}_{n+1}$ as a minimization like:

$$\mathbf{f}_{n+1} = \arg\min_{\mathbf{f}} \frac{1}{2} \left\| A_J^T \mathbf{y}_n - \mathbf{f} \right\|_2^2. \tag{3.11}$$

Combining (3.10) and (3.11), our algorithm can be viewed as solving the following minimization:

$$\mathbf{f}^* = \arg\min_{\mathbf{f}} \left\{ \arg\min_{\mathbf{y} \in S} \left\{ \frac{1}{2} \|A_J \mathbf{f} - \mathbf{y}\|_2^2 \right\} \right\}. \tag{3.12}$$

Remarks

1. The convergence of our algorithm is not yet proved. Thought out our experiments on this algorithm, the proposed algorithm does not have the global convergence; nevertheless, the local convergence behaviour is observed in our experiment with a bicubic initial guess.

2. As it can be seen in the formula, the thresholding is not used; therefore, it is possible that the algorithm will amplify the noise. Nevertheless, it can be avoided by choosing a better initial guess, by our observation, the result of bicubic upsampling seems to be a good initial guess so far.

3. More investigation on the initial guess, proof of convergence and improvement of our algorithm will be studied in the future and is described in chapter 5.

Chapter 4

Results

We are going to compare the results with those of [2]. In this chapter, the image comparison can be found and the video comparison can be found on this webpage[1]. Three test images and their bicubic upsampling results are shown in Figure 4.1.

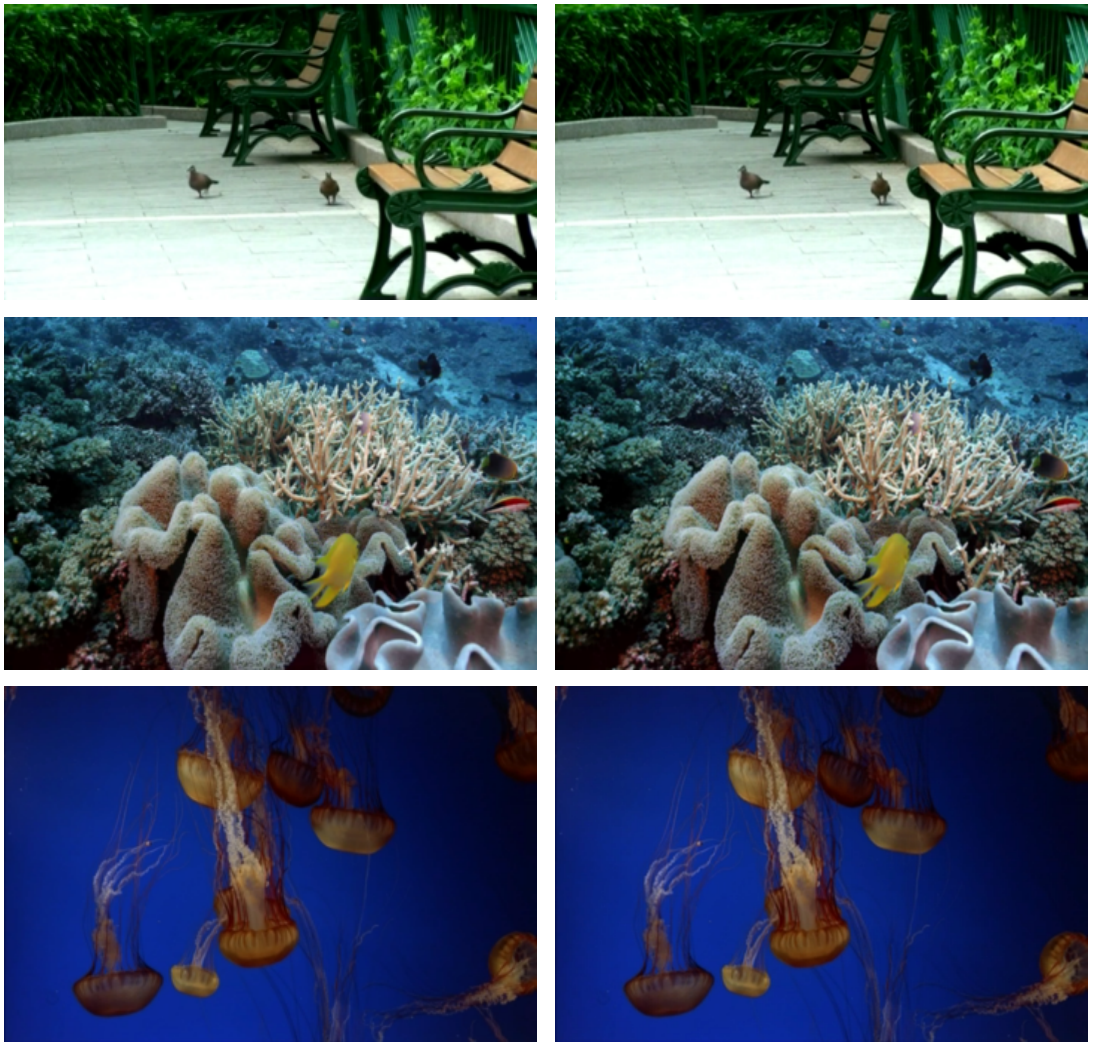

FIGURE 4.1: The three test images (left) and their bicubic results (right)

[1] http://personal.ie.cuhk.edu.hk/∼sclee6/ERG4920CR/result.html

Results comparison of the first test image:

FIGURE 4.2: Input, By [2], Level 2, Level 4 and Level 6 (Left: From up to down) Bicubic, Level 1, Level 3, Level 5 and Level 7 (Right: From up to down)

Comparison of results of Bicubic, [2] and our algorithm (Level 4):

FIGURE 4.3: Bicubic Result

FIGURE 4.4: Result of [2]

FIGURE 4.5: Result of our algorithm (Level 4)

And the comparison of the second test image can be found in Figure 4.6:

FIGURE 4.6: Input, By [2], Level 2, Level 4 and Level 6 (Left: From up to down) Bicubic, Level 1, Level 3, Level 5 and Level 7 (Right: From up to down)

Comparison of results of Bicubic, [2] and our algorithm (Level 4):

FIGURE 4.7: Bicubic Result

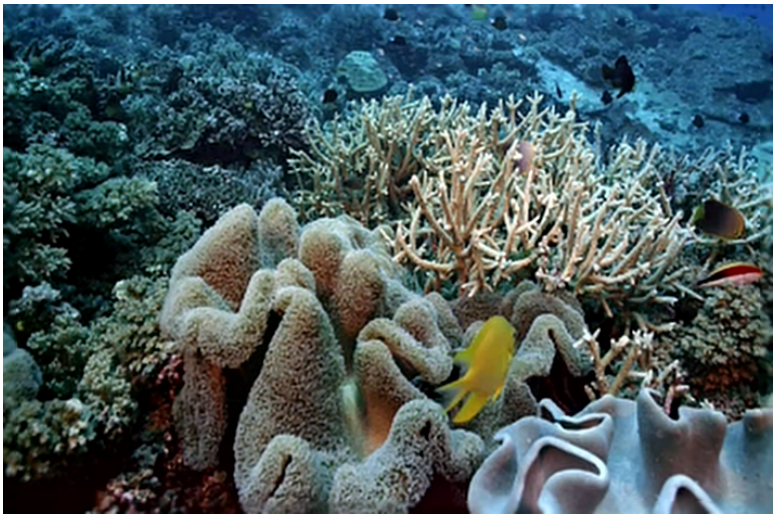

FIGURE 4.8: Result of [2]

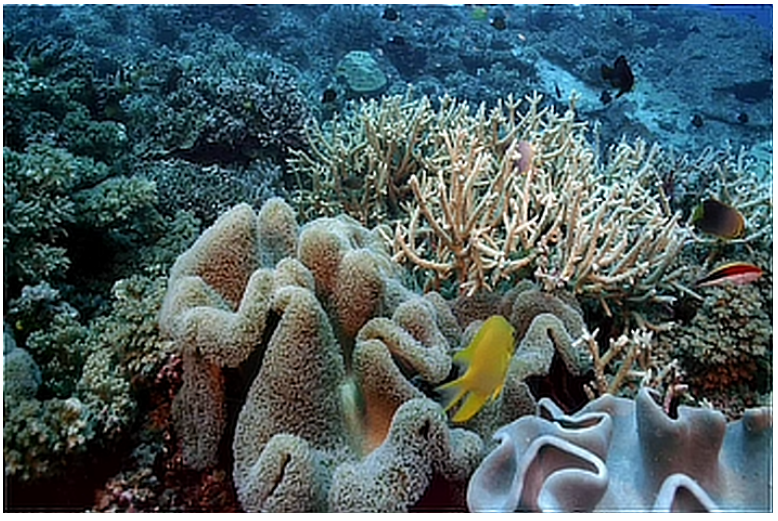

FIGURE 4.9: Result of our algorithm (Level 4)

And the comparison of the third test image can be found in Figure 4.10:

FIGURE 4.10: Input, By [2], Level 2, Level 4 and Level 6 (Left: From up to down) Bicubic, Level 1, Level 3, Level 5 and Level 7 (Right: From up to down)

Comparison of results of Bicubic, [2] and our algorithm (Level 4):

FIGURE 4.11: Bicubic Result

FIGURE 4.12: Result of [2]

FIGURE 4.13: Result of our algorithm (Level 4)

And the enlarged version of particular part of all levels and [2] can be found in Figure 4.14:

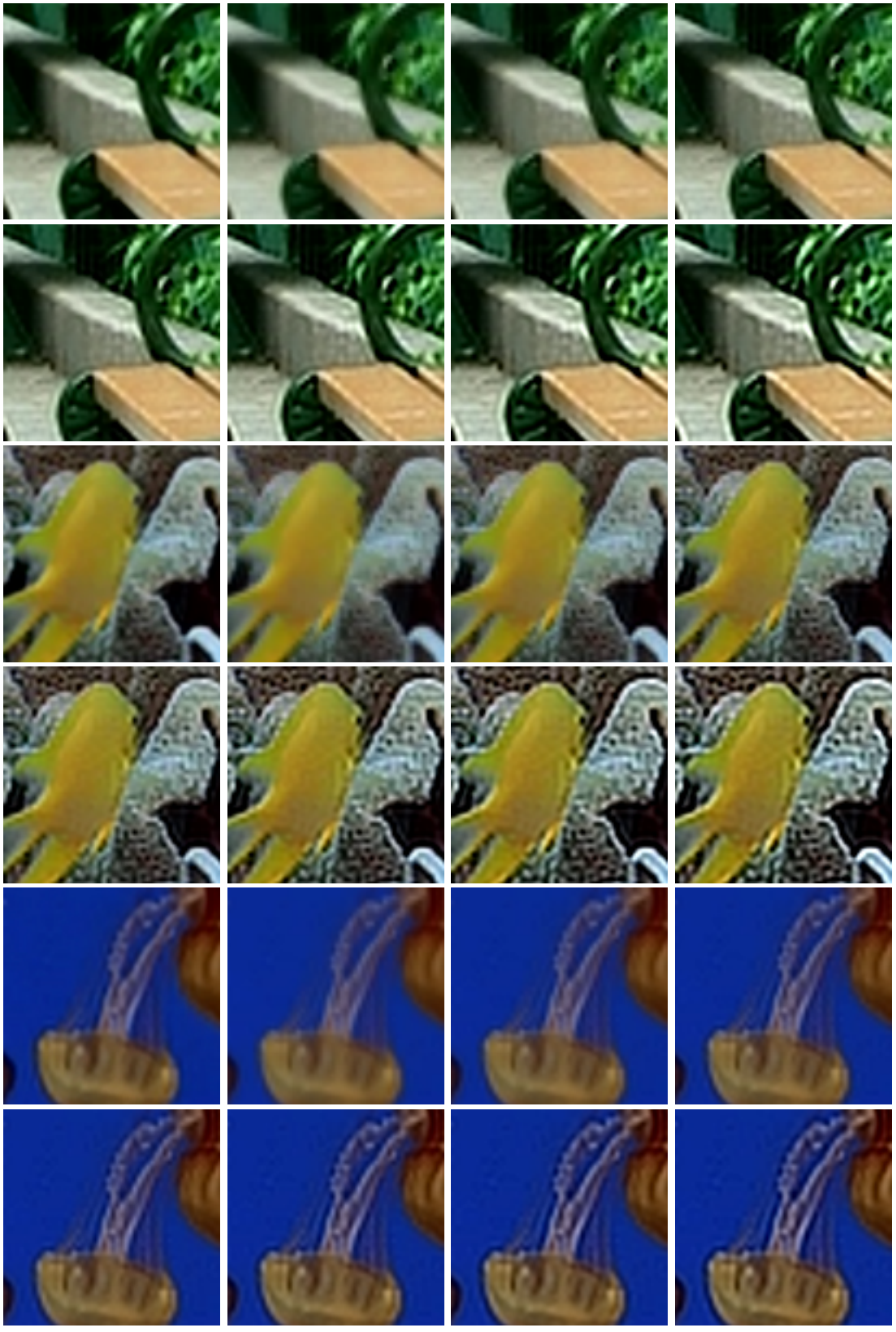

FIGURE 4.14:
By [2], Level 1, Level 2 and Level 3 (Odd number rows: From left to right)
Level 4, Level 5, Level 6 and Level 7 (Even number rows: From left to right)

Chapter 5

Conclusions and Future Directions

The algorithm is implemented by **Matlab R2009b** and is executed on a **1.66GHz CPU N280 AUSU EEEPC**. Different level tight frame algorithms take different time to perform the upsampling. The higher the level is, the longer the time is required. Of course, it also depends on the initial guess and stopping criteria. In our implementation, bicubic upsampling of the input image is used as the initial guess and $e := \frac{\|\mathbf{f}_{n+1}-\mathbf{f}_n\|_2}{\|\mathbf{f}_{input}\|_2}$ is used to measure the reconstruction error. Our stopping criteria is $e < 0.001$. On average, the computation time required for an $\mathbf{512^2}$ **pixels** input image upsampling to $\mathbf{1024^2}$ **pixels** of seven levels used in the algorithm are listed below:

It is relatively efficient and comparable to those using typical iterative approach. For

TABLE 5.1: Computation time

Level	1	2	3	4	5	6	7
Time taken	79s	292s	10 mins	19 mins	36 mins	67 mins	112 mins

example, the image upsampling provided by [2] executed on the same situation requires approximately 13 mins. We have also used the images from [2] as our input which can be found on

http://www.cse.cuhk.edu.hk/∼leojia/projects/upsampling/index.html.

By comparing with the results of [2], it can be seen that our results at higher level preserve more details and the visual perception seems better; however, the time spent is much longer. On the other hand, although our results at lower level are not as good as those of [2], our implementation speed at lower level is faster than [2] and the results is considerably acceptable. For more details with the video comparison and the alpha version of image upsampling software of our algorithm, they can be found on:

http://personal.ie.cuhk.edu.hk/∼sclee6/ERG4920CT/result.html;
http://personal.ie.cuhk.edu.hk/∼sclee6/ERG4920CT/Upsampling_pkg.exe;

Besides, more importantly, our algorithm does not require users adjust any parameters for image upsampling for each level that is different from [2] and many from the literature. In the future, the level can be automatically determined so that the algorithm can completely automatically upsample images.

In a word, we proposed an interesting and researchable algorithm for image upsampling based on the multiresolution analysis via Tight Frames. This algorithm yields a better visual quality comparing to those in literature. We will further study on it in the ways being described below.

As mentioned in the remark of chapter 4, the proof of the local convergence of our algorithm with bicubic initial guess should be dug out first. Like the convergence proof in [19], the idea of proximal forward-backward splitting in [20] may be useful in our convergence proof. Besides, different upsampling factors other than two (which is the only factor considered in this thesis) should be further considered. For other upsampling factors, other tight frame systems should be used. In this thesis, linear B-spline is used because its low pass filter is suitable to upsample an image twice its size due to its coefficient weighting. Similarly, in the future, cubic B-spline may be considerably suitable for upsampling with factor four.

Furthermore, for reconstructing a better image from the input (low resolution) image, some prior information about the desired high resolution image is always preferred. In order to accomplish this, we are going to make use of the natural science statistic which provide the prior information about the gradient density distribution of natural images as the one mentioned in the supplementary file of [2]. In our proposed algorithm, one of the high frequency band h_1 provide exactly the information of gradient of the image. In the future, we are going to use the distribution from natural science to constraint this high frequency band so that the results will more close to the desired high resolution image. Even more, we can try to use all these gradient distributions from natural statistics to constraint all the high frequency bands in our algorithm to see if a better output image can be achieved. Moreover, it can be seen in our results that too high level reconstruction may provide too much noise component and also suffer from expensive computation time while too low level cannot fully reconstruct the natural details. Therefore, it is also important for us to find an appropriate level for our algorithm to upsample images in the future.

Bibliography

[1] Jian-Feng Cai, Raymond Honfu Chan, and Zouwei Shen. A framelet-based image inpainting algorighm. *Applied and Computational Harmonic Analysis*, 24(2):131–149, 2008.

[2] Qi Shan, Zhaorong Li, Jiaya Jia, and Chi-Keung Tang. Fast image/video upsampling. *ACM Transactions on Graphics (SIGGRAPH ASIA)*, 27(5):1–7, 2008.

[3] R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 29(6):1153–1160, 1981.

[4] P. Thvenaz, T. Blu, and M. UUnser. Image interpolation and resampling. In I. Bankman, editor, *Handbook of Medical Imaging, Processing and Analysis*, pages 393–420. Academic Press, San Diego CA, USA, 2000.

[5] W. T. Freeman, T. T. Jones, and E. C. Pasztor. Example-based super-rsolution. *IEEE Computer Graphics and Applications*, 22(2):56–65, 2002.

[6] M. F. Tappen, B. C. Russell, and W. T. Freeman. Efficient graphical models for processing images. In *CVPR 2004*, pages 673–680. IEEE Computer Society, 2004.

[7] M. Ebrahimi and E. R. Vrscay. Solving the inverse problem of image zooming using self-exapmles. In *ICIAR 2007*, pages 117–130. Springer, 2007.

[8] D. Blasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV 2009*, pages 673–680, 2009.

[9] Raymond H. Chan, Sherman D. Riemenschneider, Lixin Shen, and Zuowei Shen. Tight frame: an efficient way for high-resolution image reconstruction. *Applied and Computational Harmonic Analysis*, 17(1):91–115, 2004.

[10] N. K. Bose and K. J. Boo. High-resolution image reconstruction with multisensors. *International Journal of Imaging Systems and Technology*, 9(4):131–149, 1998.

[11] W. T. Freeman and E. C. Pasztor. Learning to estimate scenes from images. In *Advances in neural information processing systems 11*, pages 775–781, Cambridge, MA, USA, 1999. MIT Press.

[12] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *SIGGRAPH 01*, pages 327–340, New York, USA, 2001. ACM.

[13] R. Fattal. Image upsampling via imposed edge statistics. *ACM Transactions on Graphics*, 26(3):95, 2007.

[14] J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *CVPR 2008*, pages 1–8, 2008.

[15] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: Mra-based constructions of wavelet frames. *Applied and Computational Harmonic Analysis*, 14(1):1–46, 2003.

[16] C. de Boor, R. DeVore, and A. Ron. On the construction of multivariate (pre)-wavelets. *Constructive Approximation*, 9(2):123–166, 1993.

[17] R. Jie and Z. Shen. Multiresolution and wavelets. In *Edinburgh Mathematics Society*, pages 271–300, 1994.

[18] A. Ron and Z. Shen. Affine system in $L_2(R^d)$: the analysis of the analysis operator. *Journal Functional Analysis*, 148:408–447, 1997.

[19] Jian-Feng Cai, Raymond H. Chan, Lixin Shen, and Zuowei Shen. Simultaneously inpainting in image and transformed domains. *Numerische Mathematik*, 112(4): 509–533, 2009.

[20] P. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting, multiscale modeling and simulation. *A SIAM Interdisciplinary Journal*, 4:1168–1200, 2004.