

SUPER-RESOLUTION BY LOCAL FUNCTION APPROXIMATION

by

STEVEN LAWLESS, B.S.

A MASTER THESIS

IN

MATHEMATICS

Submitted to the Graduate Faculty

of Texas Tech University in

Partial Fulfillment of

the Requirements for

the Degree of

MASTER OF SCIENCE

Approved

Dr. Christopher Monico

Committee Chairman

Dr. Clyde Martin

Dr. Ram Iyer

Dr. Fred Hartmeister

Dean of the Graduate School

December, 2007

ACKNOWLEDGMENTS

I would like to thank those who have helped me from the moment of making my decision to start graduate school to the time of presenting this thesis work. It has been an interesting and wonderful journey.

First, I would like to express my sincere thanks to my Thesis Advisor, Dr. Chris Monico. This thesis was only possible because of his extraordinary support, guidance, advice, and patience.

“Teachers open the door. You enter by yourself” - Chinese Proverb.

Dr. Monico not only opened the door for me, he made sure that I did not stay by the door by helping me with his time and advice. Thank you for all the time and patience you have dedicated to me over the past year.

I would like to thank my thesis committee, Dr. Clyde Martin and Dr. Ram Iyer, for their time and valuable advice during the process of this work. You have been more than my committee, you have been my professors and mentors. I am grateful for all the opportunities you have provided me to learn and grow in mathematics.

To those that served with me in the U.S. Army, you have my eternal gratitude for the courage that you showed me. You taught me to never accept failure and always strive for the best. These are traits that I took from you and made them into personal strengths of mine. Without serving with you, I would never have been able to accomplish everything that I have.

Life is always a balance between personal responsibilities and desires. My special thank you to my friends James McCullough who always reminded me who I was; James Valles for keeping me sane through the years; Daniel Holder, who always listened to me bitch and moan; and Brock Erwin who always made me laugh. You guys have provided me strong support which made possible both my professional and academic development.

A special thank you to my brother Brian, who encouraged me to think and who

kept telling me I could do it. As you and Sherry start your family, know that you will always have my love and support for your encouragement that you have given me over the years.

When I reflect back on why I even started my graduate studies in the first place, I think about my parents. My father always encouraged me to continue furthering my education and pushed me to excel at whatever I did. Through his love and support, he showed me how to be a man. My mother is always there for me, always shows me her love, and she first made me believe that I could do anything. Mom, I can only begin to understand the depths of your love you have shown me over the years. Mom and Dad, you will always have my warm thanks for all the love, for your ability to teach me how to work, learn, be confident, and of course enjoy life.

CONTENTS

ACKNOWLEDGMENTS	ii
ABSTRACT	vi
LIST OF FIGURES	vii
I INTRODUCTION	1
1.1 What is Image Processing?	1
1.2 The Basics of an Image	1
1.3 Main Image Feature Classes	2
1.3.1 Smooth Region	2
1.3.2 Edge	2
1.3.3 Texture	3
1.4 RGB Channel	3
1.5 Super-Resolution	3
II FIND THE RELATIVE OFFSET OF THE SAMPLES	5
2.1 Fourier Transform	5
2.1.1 Continuous Fourier Transform	5
2.1.2 Discrete Fourier Transform	6
2.2 Magnitude and Phase	7
2.3 Coarse Angle Alignment	9
2.4 Alignment	10
2.4.1 Finding the Area	11
2.4.2 Relative Offset	14
III RECOMBINING THE IMAGES	16
3.1 Basis Functions	16
3.2 Solving the System of Equations	18
3.3 Point-Spread Function	20
3.3.1 Estimating the True Image	22
IV EXPERIMENT RESULTS	25

4.1	CMOS Chip Set	25
4.2	MPEG-4 Set	27
V	CONCLUSIONS, FUTURE WORK	29
5.1	Conclusions	29
5.2	Future Work	30
	BIBLIOGRAPHY	31
	APPENDIX	33

ABSTRACT

Super-resolution estimates a higher resolution image given a set of lower resolution images with negligible scene differences between them. There are two key techniques developed for performing the super-resolution that is discussed in this paper. First, we develop an accurate alignment algorithm for the low-resolution images that takes into account any horizontal, vertical, and rotational shifts between the set of sample images. Second, a technique for approximating a higher resolution image by using sub-pixel level basis functions is developed.

LIST OF FIGURES

2.1	Magnitude and Phase	8
2.2	Rotation	9
2.3	Phase Spectrum Comparisons	9
2.4	Log Transformation Comparisons	10
2.5	Image Alignment	11
2.6	Types of Pixel Intersections	12
2.7	Point in Polygon	13
2.8	Rotation Alignment	15
3.1	Super-Resolution Mesh	16
3.2	Total/Orthogonal Least Squares	19
3.3	Maximum-Likelihood Blur Estimation	24
4.1	CMOS Example	26
4.2	MPEG-4 Video Example	28

CHAPTER I

INTRODUCTION

1.1 What is Image Processing?

Image processing is the processing of information for which the input signal is an image; however, the output is not necessarily an image. Standard tools of image processing treat the image as a 2-dimensional signal and apply standard signal processing techniques. Signal processing is the analysis, interpretation and manipulation of any time-varying function such as radar, sound, images, and many more. Signal processing for digital signals may involve error detection, error correction, compression, and information extraction.

In image processing, it is usually assumed that there is an underlying image model for which a mathematical model is used to create an image [14, 18, 23]. By implementing multiple image models, a more accurate representation of the individual regions of an image can be determined. With different image models, the complete image can be obtained by a finite number of local image models [14].

1.2 The Basics of an Image

An image is a function $f(x, y)$ whose domain is a subset of \mathbb{R}^2 , where f represents the brightness or intensity of the image at (x, y) [6]. If the image is multicolor, then f is vector-valued where each component of the vector indicates the brightness in the corresponding color band [6]. A true image $\hat{f}(x, y)$ has an intensity level at all points, while a digital image is a discrete version of the true image and has intensity at pixel $[i, j]$ equal to:

$$f[i, j] = \iint_{[i, i+1] \times [j, j+1]} f(x, y) dA. \quad (1.1)$$

Therefore, the representation of a 2-dimensional array of a digital image would look like this:

$$\begin{bmatrix} f[0,0] & \dots & f[0,N-1] \\ \vdots & \ddots & \vdots \\ f[M-1,0] & \dots & f[M-1,N-1] \end{bmatrix} \quad (1.2)$$

where M and N are integers. By abuse of terminology, the reference to an image in this paper will refer to a digital image when no confusion arises.

A physical object is represented as an element in the array known as a pixel. In an image, a pixel measures the intensity of three colors where the most common application is red, blue, green.

1.3 Main Image Feature Classes

In image processing, there exists a mathematical model through which the image can be classified [14]. An image is a finite number of local image features that model a complete image and are usually summarized under three regions that are usually associated with three main image features: Smooth Regions, Edges, and Textures [14].

1.3.1 Smooth Region

Smooth regions comprise the largest portion of most images. The simplest model for this region is to assign a random variable with low variances to model the intensity level locally [13].

1.3.2 Edge

Edges represent the abrupt transitions between smooth regions and constitute the smallest region of most images. While they consist of smallest area in an image, they have the most information in an image [14]. A simple model for edges is to assign a random variable with high variance to the gray level value [14], however, this simple model may lead to uncertainty with textured regions [16].

1.3.3 Texture

Textures have a noise-like appearance even though they are distinct from noise since there is a pattern within them and their self-similarities [14]. The simplest way to model texture is to use independent and identically distributed random variables [14] but this does not take any relationships among the pixels into account [24]. A generalization of this method is the implementation of Gauss Markov Random Field [24] and Gibbs random field which model these local relationship properties [14].

1.4 RGB Channel

RGB is a three-channel representation of color images in terms of Red, Green, and Blue channels. The industry standard for an RGB image is 24-bits. This allows 8-bits per channel with a gray level value between 0 and 255. RGB Channels are the most common way of storing color information [13]. By combining the three colors in various ways, it is possible to reproduce the other colors.

When the RGB channels are separated, which is achieved by setting two of the three channels to 0 (i.e. $(r,0,0)$ is red separation and r is equal to the intensity value of red at that pixel), it can be easily seen how the different intensity level values of red, green, and blue make up the different colors that are seen. For example, a strong intensity level value of all three colors makes up white, while a brownish color is composed of strong red and green intensity level values but very little blue intensity level value.

1.5 Super-Resolution

A limit in image resolution due to physical restrictions occurs in many applications. The images produced by some devices such as charge-coupled devices (CCD) cameras are under-sampled if their detector array is not sufficiently dense [11]. When taking a picture of the night sky with a CCD camera, the image of the star should form on the focal plane, hopefully as a point. However, it is known that the intensity is proportional to the square of the Fourier transform of the exit pupil of the camera

[10]. Therefore, a second and weaker-intensity nearby star might be missed because the higher-intensity star masks it. The purpose of super-resolution is to create one or more high-resolution images from a set of low-resolution images. Super-resolution has the goal of minimizing the effects of finite aperture size and discreteness and is used in the recovery of missing information [10]. The goal is to combine the information in a set of low-resolution images in order to obtain a high-resolution image that contains more information than any of the low-resolution images. To do this, certain assumptions must be made:

1. There are negligible changes between the set of low-resolution images in the region of interest.
2. The images are offset by either position and/or rotation.
3. The image sizes are sufficiently large for the extrapolation.
4. The signal-to-noise (SNR) ratio is sufficiently high to extrapolate meaningful results at higher frequencies.

To do this super-resolution, the relative offset of the low-resolution images must be determined. By using a Discrete Fourier Transform (DFT), the sample images will be decomposed into magnitude and phase components [10]. This will allow two images to be roughly aligned angularly. Then by minimizing the distance between two sample images with an appropriate metric, a finer alignment between the horizontal, vertical, and angle is determined.

After alignment, we locally approximate the true image by using sub-pixel level basis functions. With enough images, an over determined system of equations will occur. Since there will be error due to noise in the sample image and error in alignment, a total least squares approach is used to determine the approximation by basis functions. Once the true image is approximated, a point-spread function is estimated and used to remove the blurring that will be present. These are well-known techniques [5, 19] and will be applied directly to the super-resolution image.

CHAPTER II

FIND THE RELATIVE OFFSET OF THE SAMPLES

The goal of this chapter is fixing a reference image and aligning other images against it. The images are assumed to be offset by

$$f_1(x, y) = f_2(x \cos(\theta_0) + y \sin(\theta_0) + \Delta x, -x \sin(\theta_0) + y \cos(\theta_0) + \Delta y), \quad (2.1)$$

where $\Delta x, \Delta y$ are the horizontal and vertical offsets between the two images, and θ_0 is the rotational offset of the two images. By using the frequency domain, the images are roughly angularly aligned. After the coarse angle alignment is done, the images will be aligned more accurately by a method similar to steepest descent.

2.1 Fourier Transform

Every L^2 function on a compact subset of \mathbb{R} has a Fourier series expansion [13]. A Fourier Transform allows the decomposition of signals into basis signals of sinusoidal. Since an image is finite, the Fourier Domain makes an ideal place to work in to find the relative offset of the low-resolution images.

2.1.1 Continuous Fourier Transform

It is assumed that the true image f is L^2 square integrable therefore a two-dimensional Fourier Transform could be used to find the relative offset of sample images. The Fourier Transform is given by

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi iux} dx, \quad (2.2)$$

and its inverse transform by

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{2\pi iux} du. \quad (2.3)$$

(2.2) and (2.3) make up the *Fourier Transform Pair* of f . These pairings allows a function to be recovered from its transform. The *Fourier Transform Pair* for a two

variable, L^2 function are given by

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-2\pi i(ux+vy)} dx dy, \quad (2.4)$$

and its inverse transform by

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{2\pi i(ux+vy)} du dv. \quad (2.5)$$

2.1.2 Discrete Fourier Transform

Since a digital image is not given as a function on a compact subset of \mathbb{R}^2 , but as a finite collection of pixels, Discrete Fourier Transform (DFT) is more applicable than the Fourier Transform given above. A one-dimensional DFT pair is given by

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{-2\pi i u x / N}, \quad (2.6)$$

and its inverse transform by

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{2\pi i u x / N}. \quad (2.7)$$

Extending the one-dimensional DFT and its inverse to a two-dimensional DFT is given by

$$F(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x, y) e^{-2\pi i(ux/N+vy/M)}, \quad (2.8)$$

and the inverse DFT by,

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u, v) e^{2\pi i(ux/N+vy/M)}. \quad (2.9)$$

A DFT is computationally expensive procedure to implement, so multiple one-dimensional Fast Fourier Transforms (FFT) are used to compute the DFTs. Well established techniques have been developed for computing one-dimensional FFT such as the successive doubling method [13, 20].

If x is held constant, then the sum over y is a one-dimensional DFT. If y is held constant, then the sum over x is a one-dimensional DFT. By splitting (2.8) up as

$$F(u, v) = \frac{1}{NM} \sum_{x=0}^{N-1} e^{-2\pi i u x / N} \sum_{y=0}^{M-1} f(x, y) e^{-2\pi i v y / M} \quad (2.10)$$

it will be possible to implement multiple one-dimensional FFTs on a two-dimensional DFT. So, a two-dimensional DFT is computed by performing a one-dimensional FFT over each column, then computing a one-dimensional FFT over each row on the resulting values.

An important property of the *Discrete Transform Pair* is that the DFT and its inverse always exists [13]. The magnitude of a DFT is given by

$$|F(u, v)| = |Re^2F(u, v) + Im^2F(u, v)|^{1/2} = \sqrt{F(u, v)\overline{F(u, v)}}, \quad (2.11)$$

and the phase spectrum is given by

$$\phi(u, v) = \tan^{-1} \left[\frac{ImF(u, v)}{ReF(u, v)} \right]. \quad (2.12)$$

Since the components of a DFT are complex quantities, the magnitude (2.11) is used since it contains the most information about the spatial domain image.

When computing the DFT, by making use of the shifting theorem [13, 20], each component of the low-resolution images is multiplied by $(-1)^{x+y}$. This multiplication, (2.13), centers the zero-frequency of the low-resolution image in the frequency domain at the point $\left(\frac{M}{2}, \frac{N}{2}\right)$ [13] and will allow for the multiple low-resolution image magnitudes to be easily aligned in the frequency domain. That is, if $\mathcal{F}[\cdot]$ denotes the DFT then

$$\mathcal{F} [f(x, y) (-1)^{x+y}] = F(u - M/2, v - N/2). \quad (2.13)$$

2.2 Magnitude and Phase

Using the low-resolution image as seen in Figure 2.1(a), a DFT will be computed and the magnitude and phase components shown separately. The MATLAB code used to calculate both the magnitude and phase can be found in the appendix. As seen in Figure 2.1(b), the magnitude appears black except at the center where the zero-frequency is. A logarithmic transformation, (2.14), is applied in order to see the magnitude [13]. The constant, c , is chosen by (2.15) to ensure that the maximum value that each pixel intensity will take is 255. The variable r is the maximum

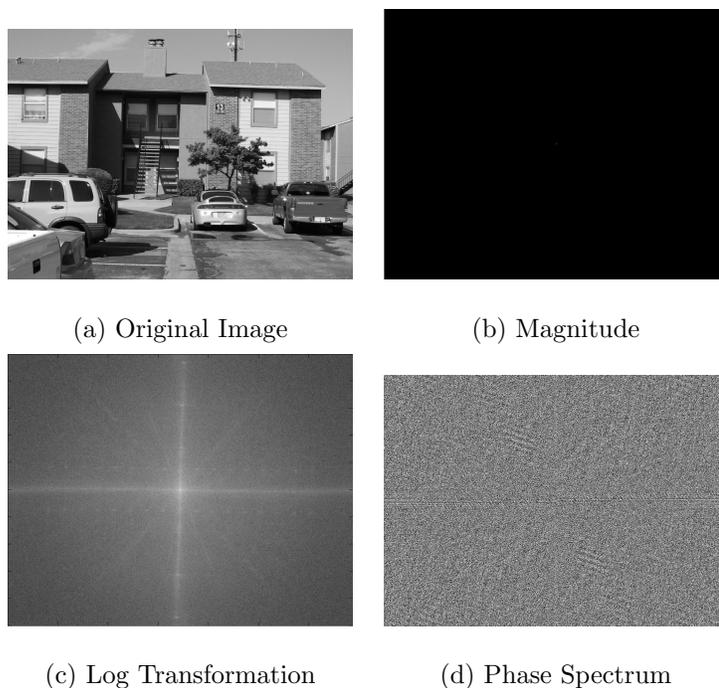


Figure 2.1: Magnitude and Phase

magnitude value of the DFT.

$$S = c \ln(1 + |r|) \quad (2.14)$$

$$c = \frac{255}{\ln(1 + |r|)}$$

Examining Figure 2.1(c), distinct vertical and horizontal lines appear corresponding to some dominant vertical and horizontal features in the original image. If the image is rotated 45° as in Figure 2.2(a), the vertical and horizontal lines of the logarithmic transformation as seen in Figure 2.1(c) also rotate 45° also as seen in Figure 2.2(b). This distinctive relationship between the rotation of the image and the rotation in the magnitude will allow multiple low-resolution images to be aligned.

When examining the phase spectrum image of 2.1(d) with that of Figure 2.3(b) which is a 45° of the image in Figure 2.1(a), the same distinct vertical and horizontal lines appear in both phase spectrums. Since the phase does not yield a large amount of new information about the image [10], the alignment of the multiple low-resolution images will occur with the magnitude of the DFT.

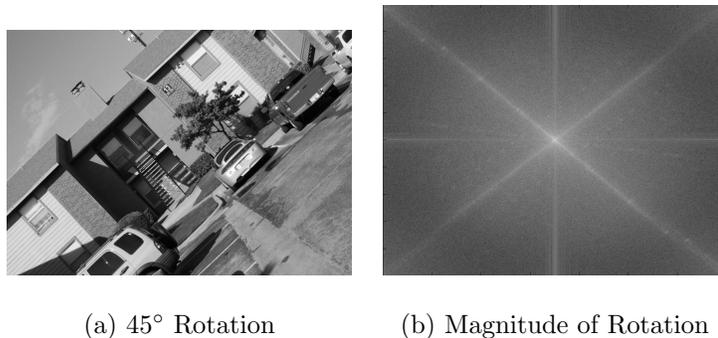


Figure 2.2: Rotation

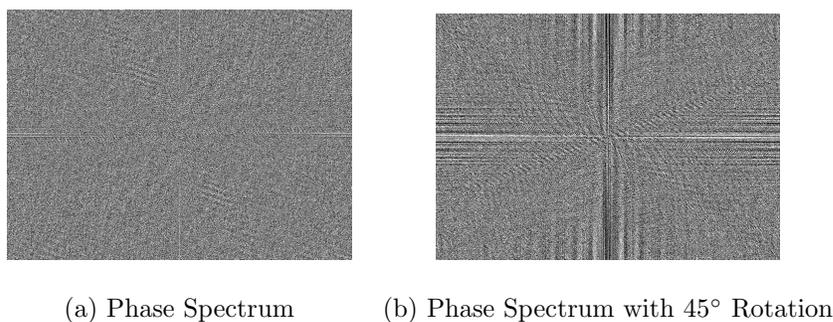


Figure 2.3: Phase Spectrum Comparisons

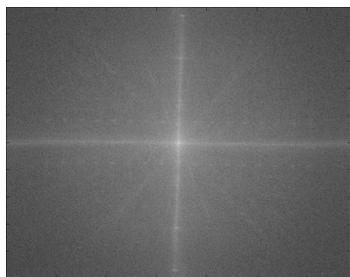
2.3 Coarse Angle Alignment

Using the DFT magnitude components of two images, $|F_1(u, v)|$ and $|F_2(u, v)|$, coarse angular alignment between the two images can be found. If the images have rotational angle difference similar to Figure 2.4(a) and Figure 2.4(b) or even a small rotational difference similar to Figure 2.4(a) and Figure 2.4(c), then a coarse angle approximation will be required. The coarse angle approximation will align the magnitudes of the two images in such a way so that the rotational difference between them is similar to that of Figure 2.4(a) and Figure 2.4(d).

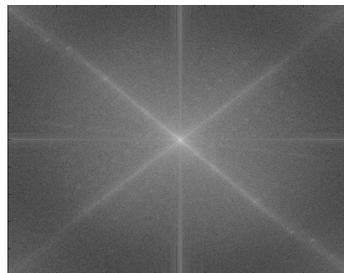
By using (2.1) and letting $G_1(r, \theta)$ represent the polar DFT magnitude of $f_1(x, y)$ and $G_2(r, \theta)$ represent the polar DFT magnitude of $f_2(x, y)$, then the two magnitudes have the following relationship [2]:

$$G_1(r, \theta) \approx G_2(r, \theta + \theta_0) \quad (2.15)$$

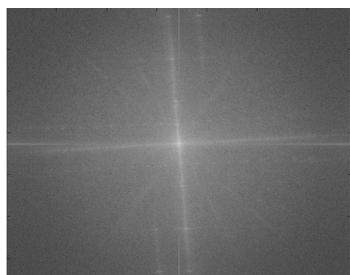
Therefore, by (2.15) the rotational shift of the two images can be determined by phase-



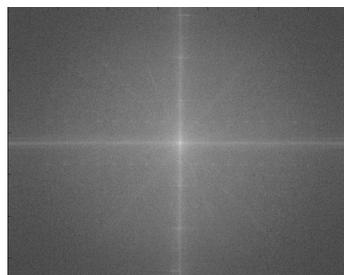
(a) Log Transformation



(b) Log Transformation with 45° Rotation



(c) Log Transformation with 5° Rotation



(d) Log Transformation with 1° Rotation

Figure 2.4: Log Transformation Comparisons

correlation techniques [2]. However, an ambiguity of π results when using (2.15) [9]. This ambiguity can be solved by rotating $f_2(x, y)$ by θ_0 and $\theta_0 + \pi$ and choosing the angle which minimizes the difference between the original images.

2.4 Alignment

After coarse angle alignment is done, the DFTs are discarded and the sample image is rotated by the coarse image alignment value. Using the two images, vertical, horizontal, and small rotational alignment is carried out. Figure 2.5 is a representation of sample image pixels alignment with a reference image pixel. The shaded box is a reference image pixel and the white boxes are four pixels from the sample image that is being aligned. The reference image pixel will have intensity $I_{(i,j)}^{(Ref)}$ while the sample image pixels will have intensities of $f[l, m]$. Assuming that $f[l, m]$ is evenly distributed over the entire pixel and letting $A\left(P_{(i,j)}^{ref} \cap P_{(l,m)}^{sample}\right)$, represent the area of the reference image in each pixel of the sample image, then the following relation

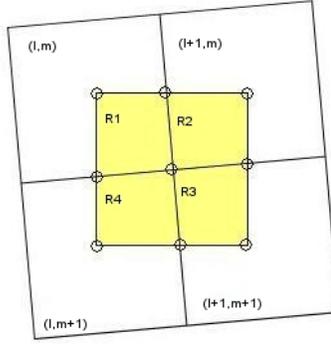


Figure 2.5: Image Alignment

occurs when the alignment is approximately correct:

$$I_{(i,j)}^{(Ref)} \approx \sum_{l,m} A \left(P_{(i,j)}^{ref} \cap P_{(l,m)}^{sample} \right) \cdot f[l, m]. \quad (2.16)$$

By minimizing the difference over all pixels between the sample and reference images by

$$\varepsilon = \left| I_{(i,j)}^{(Ref)} - \sum_{l,m} A \left(P_{(i,j)}^{ref} \cap P_{(l,m)}^{sample} \right) \cdot f[l, m] \right|, \quad (2.17)$$

where ε is the tolerance, then the sample and reference images are considered aligned.

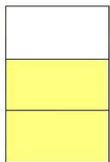
2.4.1 Finding the Area

We will explicitly find the intersection between a reference image pixel and a sample image pixel to find the area that they share in common. The reference image pixel and sample image pixel can share some common area as seen in Figure 2.6(a)-(b). When the two pixels share common area, it is possible for the polygon to have up to eight sides.

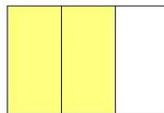
The first step to solving the area of intersection is to determine the points of intersection between the reference image pixel and the sample image pixel, it becomes a matter of finding the intersection of two line segments multiple times. The idea is to work with parametric equations for the line segments. The parametric form for the equation of a line passing through the points (x_i, y_j) and (x_{i+1}, y_{j+1}) is (2.18). If t is on the interval $[0, 1]$ after solving the parametric (2.18), then the point of intersection



(a) 3-Sides w/ Common Area (b) 8-Sides w/ Common Area



(c) Vertical Offset



(d) Horizontal Offset

Figure 2.6: Types of Pixel Intersections

is on the line segment verses being a point somewhere on the infinitely extended line.

$$\begin{aligned}x(t) &= x_i + (x_{i+1} - x_i)t \\y(t) &= y_j + (y_{j+1} - y_j)t\end{aligned}\tag{2.18}$$

Finding all points of intersection between the boundaries of two pixels will determine if the reference image pixel and the sample image pixel share common area. If the boundaries share a common segment as seen in Figure 2.6(c)-(d), then we will only include points from the segment which are also the vertices. Next, if they two pixels do not share a common segment, then we will need to find all vertices that are interior to the other pixel. This will be accomplished by a point in polygon method. The point in polygon method works by taking a ray and checking how many times it crosses the edge of a polygon. As seen in Figure 2.7, if there is an even number of crosses, then the point is outside the polygon while if there an odd number of crossings then the point is inside the polygon [15]. If the corner has an even number of crossings, then the corner is not a reference point to determine the area of intersection. However, if the corner has an odd number of crossings, then the corner is a reference point to determine the area of intersection.

Now that all reference points that form the polygon between the reference and

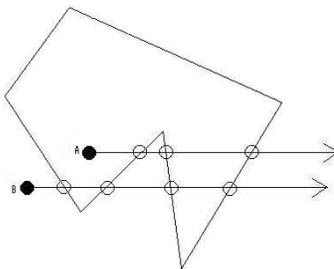


Figure 2.7: Point in Polygon

sample image pixels have been determined, it becomes a matter of finding the area of a polygon. Using Green's Theorem in a positive (counter-clockwise) rotation will find the area of the polygon formed between the sample image pixel and the reference image pixel. So it becomes necessary to label the reference points in some sort of order. Using a ray from the centroid of the polygon to each point and sweeping it around from 0 to 2π will locate and label the points of intersection between the pixels in a positive orientation, since the region is necessarily convex. The area of a region, R , in the xy -plane is given by $A = \iint_R dx dy$. Lemma 2.1 is a direct result from Green's Theorem.

Lemma 2.1. *If V_0, V_1, \dots, V_n is the sequence of adjacent vertices of a simple polygon ordered in the positive direction with $V_j = (x_j, y_j)$ and $V_0 = V_n$, then area of the polygon is given by*

$$A = \frac{1}{2} \sum_{j=0}^{n-1} (x_{j+1} + x_j)(y_{j+1} - y_j) \quad (2.19)$$

Proof. Green's Theorem states [20]:

Let R be a closed bounded region whose boundary C consists of finitely many smooth curves. Let M and N be functions that are continuous and have continuous partial derivatives $\frac{\partial M}{\partial y}$ and $\frac{\partial N}{\partial x}$ everywhere in some domain containing R . Then

$$\iint_R \left(\frac{\partial N}{\partial x} - \frac{\partial M}{\partial y} \right) dA = \oint_C M dx + N dy.$$

Since the area of R is $A = \iint_R dA$, letting $M = 0$ and $N = x$ we have

$$A = \iint_R dA = \iint_R \left(\frac{\partial N}{\partial x} - \frac{\partial M}{\partial y} \right) dA = \oint_C x dy = \sum_{j=0}^{n-1} \oint_{C_j} x dy,$$

where C is the boundary of the polygon formed by the two pixels and $\{C_j\}$ is the line segment formed between two consecutive points. Parameterize the line by

$$\begin{aligned} x(t) &= x_j + (x_{j+1} - x_j)t &\Rightarrow & dx = (x_{j+1} - x_j) dt, \\ y(t) &= y_j + (y_{j+1} - y_j)t &\Rightarrow & dy = (y_{j+1} - y_j) dt, \end{aligned}$$

to find the line integral over $\{C_j\}$. Therefore,

$$\begin{aligned} A &= \sum_{j=0}^{n-1} \oint_{C_j} x dy \\ &= \sum_{j=0}^{n-1} \int_0^1 [x_j (y_{j+1} - y_j) + (x_{j+1} - x_j)t (y_{j+1} - y_j)] dt \\ &= \sum_{j=0}^{n-1} \left[x_j (y_{j+1} - y_j) + \frac{1}{2} (x_{j+1} - x_j) (y_{j+1} - y_j) \right] \\ &= \sum_{j=0}^{n-1} \left[x_j (y_{j+1} - y_j) + \frac{1}{2} x_{j+1} (y_{j+1} - y_j) - \frac{1}{2} x_j (y_{j+1} - y_j) \right] \\ &= \sum_{j=0}^{n-1} \left[\frac{1}{2} x_j (y_{j+1} - y_j) + \frac{1}{2} x_{j+1} (y_{j+1} - y_j) \right] \\ &= \frac{1}{2} \sum_{j=0}^{n-1} (x_{j+1} + x_j) (y_{j+1} - y_j). \end{aligned}$$

□

2.4.2 Relative Offset

Having found a coarse angular alignment between the two images, we will more precisely align them horizontally, vertically, and angularly by a method similar to steepest descent. Figure 2.8 shows that there are twenty-seven different directions in which the sample image can be moved using vertical, horizontal, and angular shifts.

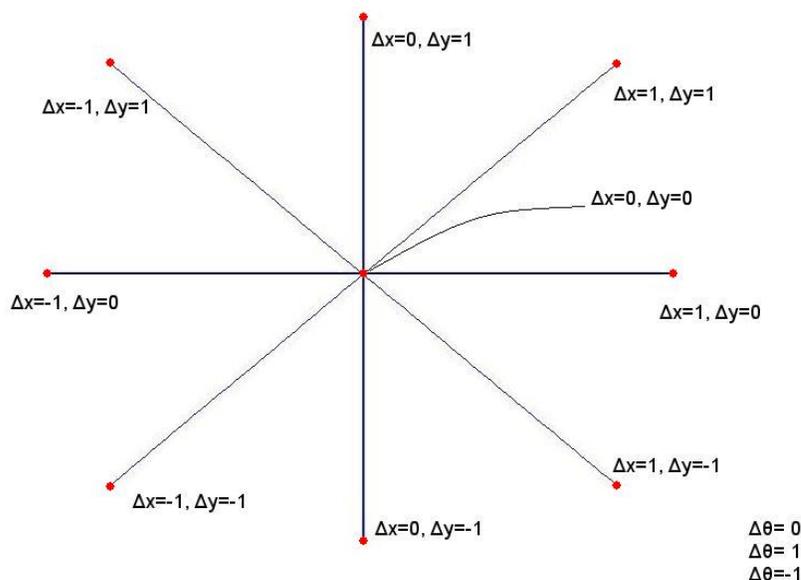


Figure 2.8: Rotation Alignment

The sample image can have a horizontal shift of 0, Δx , or $-\Delta x$ and/or it could also have a vertical shift of 0, Δy , or $-\Delta y$ and/or at the same time have an angular shift of 0, $\Delta\theta$, or $-\Delta\theta$ as seen in Figure 2.8. Compute the distance between the sample image and the reference image for each of the twenty-seven possible shifts and then choose the direction which minimizes this distance. Choosing the minimum over all twenty-seven directions will allow the images to be aligned because the distance between the sample image and reference image is expected to achieve its global minimum when the images are nearly perfectly aligned and after performing the coarse angle alignment. After the translation, multiply Δx , Δy , and $\Delta\theta$ by a constant value between $\frac{1}{2} \leq c < 1$ and then repeat. Since the distance goes to 0 as the images are aligned, a tolerance τ is used to determine when the two images are aligned. When four successive applications do not yield an improvement of at least τ , then the images are considered aligned.

CHAPTER III

RECOMBINING THE IMAGES

3.1 Basis Functions

Our goal is to approximate the true image locally as a linear combination of some orthogonal basis functions $\beta_1 \cdots \beta_n$, defined on $[-\frac{1}{2}, \frac{1}{2}]^2$. Specifically, we wish to approximate the true image in the reference pixel (i, j) by a function $f^{(i,j)}$ of the form

$$f^{(i,j)}(x, y) = \sum_{k=1}^n c_k^{(i,j)} \beta_k(x, y), \quad (3.1)$$

where $c_k^{(i,j)} \in \mathbb{R}$ and $(x, y) \in [-\frac{1}{2}, \frac{1}{2}]^2$ are coordinates local to pixel (i, j) . For a fixed positive integer r , where r is the resolution enhancement factor of the super-resolution image, we define regions $R_1 \cdots R_{r^2}$ by

$$R_{a+br+1} = \left[-\frac{1}{2} + \frac{a}{r}, -\frac{1}{2} + \frac{a+1}{r} \right] \times \left[\frac{1}{2} - \frac{b+1}{r}, \frac{1}{2} - \frac{b}{r} \right], \quad (3.2)$$

for $0 \leq a < r, 0 \leq b < r$.

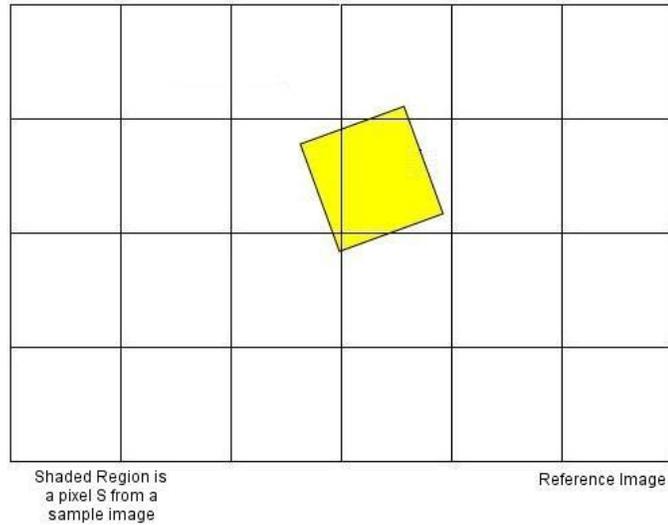


Figure 3.1: Super-Resolution Mesh

Figure 3.1 represents a single pixel S , the shaded region, from a sample image.

This single pixel gives the equation

$$\text{Intensity of } S = \sum_{\substack{0 \leq i < N \\ 0 \leq j < M}} \iint_{S \cap P_{(i,j)}} f^{(i,j)} dA, \quad (3.3)$$

where $P_{(i,j)}$ is the region associated with pixel (i, j) from the reference image. However, most of those integrals will be zero since S intersects only a few of the $P_{(i,j)}$. In turn, each $\iint_{S \cap P_{(i,j)}} f^{(i,j)} dA$ has an expansion into integrals of basis functions in each $P_{(i,j)}$ given by

$$\iint_{S \cap P_{(i,j)}} f(x, y) dA = \sum_{k=1}^n c_k^{(i,j)} \iint_{S \cap P_{(i,j)}} \beta_k(x, y) dA. \quad (3.4)$$

Since the basis functions are chosen in advance, the value of

$$\iint_{R_{k'}} \beta_k(x, y) dA = \oint_{\partial R_{k'}} M_k dx + N_k dy \quad (3.5)$$

is determined by using Green's Theorem and fixed anti-derivatives M_k and N_k are chosen in advance. For example, if some $\beta_k = x$, then using Green's Theorem and letting $N = \frac{1}{2}x^2$ and $M = 0$ the following relation occurs:

$$\iint_{R_{k'}} x dA = \oint_{\partial R_{k'}} \frac{1}{2}x^2 dy = \sum_{j=0}^{n-1} \oint_{\partial R_{k'_j}} \frac{1}{2}x^2 dy$$

Since the region is a polygon, we may parameterize the equations by

$$\begin{aligned} x(t) &= x_j + (x_{j+1} - x_j)t &\Rightarrow & dx = (x_{j+1} - x_j) dt, \\ y(t) &= y_j + (y_{j+1} - y_j)t &\Rightarrow & dy = (y_{j+1} - y_j) dt, \end{aligned}$$

on the interval $0 \leq t \leq 1$. Therefore,

$$\begin{aligned}
\iint_{R_{k'}} x dA &= \frac{1}{2} \sum_{j=0}^{n-1} \int_0^1 (x_j + (x_{j+1} - x_j)t)^2 (y_{j+1} - y_j) dt \\
&= \frac{1}{2} \sum_{j=0}^{n-1} \int_0^1 [x_j^2 + 2x_j(x_{j+1} - x_j)t + (x_{j+1} - x_j)^2 t^2] (y_{j+1} - y_j) dt \\
&= \frac{1}{2} \sum_{j=0}^{n-1} x_j^2 (y_{j+1} - y_j) + x_j (x_{j+1} - x_j) (y_{j+1} - y_j) + \frac{1}{3} (x_{j+1} - x_j)^2 (y_{j+1} - y_j) \\
&= \frac{1}{2} \sum_{j=0}^{n-1} x_j x_{j+1} (y_{j+1} - y_j) + \frac{1}{3} [x_j^2 - 2x_j x_{j+1} + x_{j+1}^2] (y_{j+1} - y_j) \\
&= \frac{1}{6} \sum_{j=0}^{n-1} [x_j^2 + x_j x_{j+1} + x_{j+1}^2] (y_{j+1} - y_j).
\end{aligned}$$

Now that the $\iint_{R_{k'}} \beta_k dA$ can be numerically computed for all β_k and is a constant value. (3.4) becomes a single equation with up to $6k$ unknowns. Therefore, if there are k basis functions then the process will require a minimum of k sample images.

The basis functions that will be used in the experiments of this project will be step functions:

$$\beta_k(x, y) = \begin{cases} 1 & (x, y) \in R_k, \\ 0 & \text{otherwise,} \end{cases} \quad (3.6)$$

for $0 \leq k < r^2$. In turn, the result of the integrals of the basis functions is the area of the polygon from Section 2.4.1. Therefore, (3.3) becomes the sum of all the overlapping areas between reference image and the sample image pixels and is expressed as

$$\text{Intensity of S} = \sum_{\substack{0 \leq i < N \\ 0 \leq j < M \\ 0 \leq k < r^2}} c_k^{(i,j)} \iint_{S \cap P_{(i,j)}} \beta_k dA. \quad (3.7)$$

3.2 Solving the System of Equations

Each of the sample images has some error in it. The noise associated with pixels, the finiteness of the set of basis functions, and error in image alignment will cause

error on both sides of the system of equations. Solving an over determined system of linear equations $Ax \approx b$, where both A and b have error in them, is done by the use of Orthogonal Least Squares (OLS) or Total Least Squares (TLS). Solving by TLS will reduce the error in the vertical and horizontal directions. The method of TLS will minimize the perpendicular distance verses favoring the vertical distances [3] as seen in Figure 3.2.

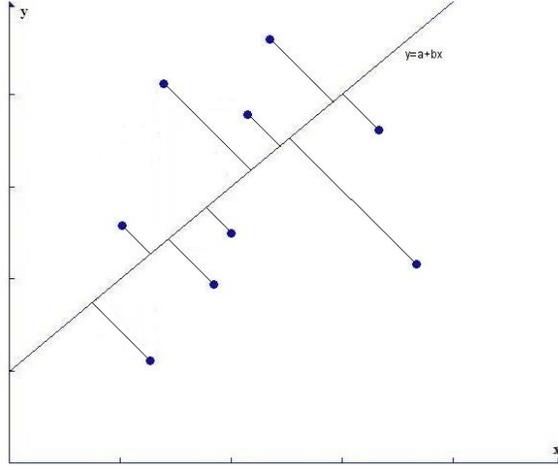


Figure 3.2: Total/Orthogonal Least Squares

The $[\cdot|\cdot]$ will denote an augmented matrix. The Frobenius norm $\|A\|_F$ of an $m \times n$ matrix A is defined by

$$\|A\|_F \equiv \sqrt{\sum_{i=1}^m \sum_{j=1}^n |a_{i,j}|^2}. \quad (3.8)$$

The following definition of TLS is from [21].

Definition 3.2.1. (Total Least Squares) *Given an over determined set of m linear equations $Ax \approx b$ in n unknowns x . The TLS problem seeks to solve*

$$\min_{[\hat{A}|\hat{b}] \in \mathbb{R}^{m \times (n+1)}} \left\| [A|b] - [\hat{A}|\hat{b}] \right\|_F, \quad (3.9)$$

subject to $\hat{b} \in \text{colsp}(\hat{A})$. Once a minimizing $[\hat{A}|\hat{b}]$ is found, then any x^ satisfying*

$$\hat{A}x^* = \hat{b} \quad (3.10)$$

is called a *TLS solution*.

The TLS models are the observed variables that satisfy one or more unknown but exact linear relations of the form: [21]

$$\alpha_1 x_1 + \cdots + \alpha_n x_n = \beta_n \quad (3.11)$$

The m equations in A , b are related to the n unknown parameters of x [21] by:

$$A_0 x = b_0, \quad A = A_0 + \Delta A \text{ and } b = b_0 + \Delta b, \quad (3.12)$$

where ΔA , Δb are the error in the measurements [21]. There is no assumed distribution of the errors on TLS. If the error in TLS is independent and identically distributed (i.i.d.) with mean zero and covariance $\sigma_v^2 I$, then TLS method converges to the true solution, x_0 , as m (the number of equations) goes to infinity [21]. Just as LS has an analytical expression [8] of

$$x^* = (A^T A)^{-1} A^T b, \quad (3.13)$$

so does TLS have an analytical expression [21] of

$$x^* = (A^T A - \sigma_{n+1}^2 I)^{-1} A^T b, \quad (3.14)$$

where σ_{n+1}^2 is the smallest singular value of $[A|b]$. While the TLS is more ill-conditioned than the LS because small changes in the constant coefficients will result in a large changes in the solutions, it does however asymptotically remove the bias from the $A^T A$ matrix [21].

3.3 Point-Spread Function

After the super-resolution image is obtained, there will be blurring in the image due to error in alignment and some due to miss-focus. The original images were images were only in focus to a level detectable at the lower-resolution. To account for this blurring, a Point-Spread Function (PSF) is used. The PSF, $\tilde{d}(x, \alpha, y, \beta)$, conveys how much the output value at (α, β) is influenced by the input value at (x, y)

[6]. Letting $g(x, y)$ represent the blurred image, $f(x, y)$ represent the true image, and $\eta(x, y)$ is noise in the system that is independent of position, then the following Fredholm integral of the first kind is obtained [13]:

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) \tilde{d}(x, \alpha, y, \beta) d\alpha d\beta + \eta(x, y). \quad (3.15)$$

It is expected that $\eta(x, y)$ is negligible since noise is averaged out by the super-resolution method provided that we have enough sample images available.

The importance of (3.15) is that if the response of $\tilde{d}(x, \alpha, y, \beta)$ is known and $\eta(x, y)$ is sufficiently small, then $f(\alpha, \beta)$ can be calculated for all α and β using (3.15) [13]. It is usually assumed that the blurring function of the camera lens is position invariant [22], that is $\tilde{d}(x, \alpha, y, \beta) = d(x - \alpha, y - \beta)$. In this case (3.15) becomes

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta) d(x - \alpha, y - \beta) d\alpha d\beta, \quad (3.16)$$

a convolution integral. Therefore, (3.16) becomes:

$$g(x, y) = d(x, y) * f(x, y), \quad (3.17)$$

and taking the 2-Dimensional Fourier Transform of (3.17) the following expression is obtained in the frequency domain [13]:

$$G(u, v) = D(u, v)F(u, v). \quad (3.18)$$

Thus, if $d(x, y)$ were known, we could easily recover $f(x, y)$ by

$$f(x, y) = \begin{cases} 0 & \mathcal{F}^{-1} \left[\frac{G(u, v)}{D(u, v)} \right] < 0, \\ 255 & \mathcal{F}^{-1} \left[\frac{G(u, v)}{D(u, v)} \right] > 255, \\ \mathcal{F}^{-1} \left[\frac{G(u, v)}{D(u, v)} \right] & \text{otherwise.} \end{cases} \quad (3.19)$$

3.3.1 Estimating the True Image

A significant problem with recovering $f(x, y)$ is the lack of information about the blurring function $d(x, y)$ [19]. A 3×3 matrix

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}, \quad (3.20)$$

with an initial guess can model the PSF of blurring function [7]. This matrix A determines the PSF $d(x, y)$ by

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} d[-1, 1] & d[0, 1] & d[1, 1] \\ d[-1, 0] & d[0, 0] & d[1, 0] \\ d[-1, -1] & d[0, -1] & d[1, -1] \end{bmatrix}. \quad (3.21)$$

After an initial guess, generally better coefficients of the matrix can be determined using a well-known Maximum-Likelihood Blur Estimations (ML) technique developed in [5, 19]. Using (3.22) and (3.23) where $A(u, v)$ is the DFT of $a_{i,j}$, σ_v^2 is the variance of the observation noise, and σ_w^2 is the variance of image noise. [7].

$$L(\theta) = - \sum_u \sum_v \left(\log(P(u, v)) + \frac{|G(u, v)|^2}{P(u, v)} \right), \quad (3.22)$$

where

$$P(u, v) = \sigma_v^2 \frac{|D(u, v)|^2}{|1 - A(u, v)|^2} + \sigma_w^2. \quad (3.23)$$

Both σ_v^2 and σ_w^2 are assumed to be Gaussian distributed for the ML.

Maximizing the log-likelihood function, $L(\theta)$, over the parameters

$$\theta = \{a_{i,j}, \sigma_v^2, d(x, y), \sigma_w^2\} \quad (3.24)$$

is the goal [7]. In order to obtain a solution to this ML, some constraints must be placed on the PSF. First, the Energy Conservation

$$\sum_{x=0}^{N-1} \sum_{y=0}^{M-1} d[x, y] = 1, \quad d[x, y] \geq 0, \quad (3.25)$$

must be met [19]. Second, the symmetry

$$d[-x, -y] = d[x, y]. \quad (3.26)$$

of the PSF of (3.26) must be maintained [7]. Thus the relationship between (3.20) and (3.25) - (3.26) is

$$\sum_{i=1}^3 \sum_{j=1}^3 a_{i,j} = 1, \quad a_{i,j} \geq 0, \quad \text{and} \quad \begin{cases} a_{1,1} = a_{3,3} \\ a_{1,2} = a_{3,2} \\ a_{1,3} = a_{3,1} \\ a_{2,1} = a_{2,3} \end{cases}. \quad (3.27)$$

So for example, letting

$$A = \begin{bmatrix} 0.056 & 0.159 & 0.042 \\ 0.136 & 0.214 & 0.136 \\ 0.042 & 0.159 & 0.056 \end{bmatrix}$$

would satisfy the properties (3.25) and (3.26) of the PSF.

With a good initial guess of θ , an expectation-minimization (EM) algorithm is a general procedure for finding the ML [5, 7]. Figure 3.3 is a black-box diagram of the Maximum-likelihood blur estimation by EM procedure [7].

Using the θ parameters, a Wiener restoration filter

$$H(u, v) = \frac{\overline{D(u, v)}}{D(u, v)D(u, v) + \frac{S_w(u, v)}{S_f(u, v)}} \quad (3.28)$$

estimates the new true image $f_E(x, y)$ [7], where $S_f(u, v)$ is the power spectrum of the ideal image and $S_w(u, v)$ is the power spectrum of the noise [7]. An approach that is used when the quantities of $S_f(u, v)$ and $S_w(u, v)$ are not known is to approximate (3.28) by

$$H(u, v) \approx \frac{\overline{D(u, v)}}{D(u, v)D(u, v) + K}, \quad (3.29)$$

where K is a constant [13]. The constant K is usually chosen from a list of values depending on the type of distribution of the noise [13]. Therefore, using a Wiener

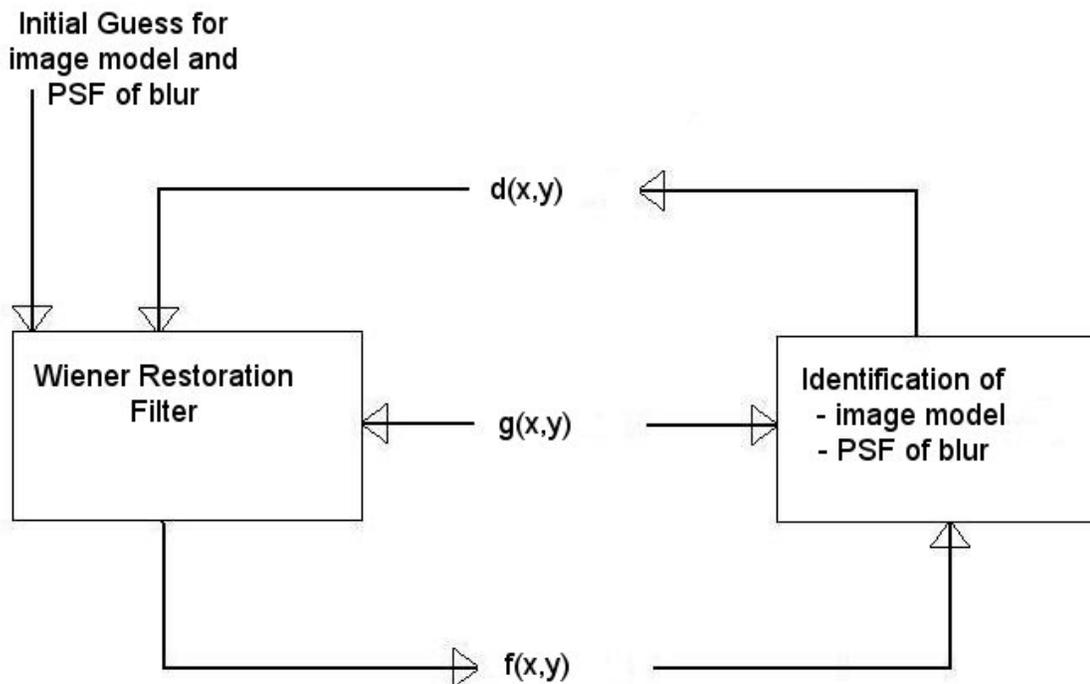


Figure 3.3: Maximum-Likelihood Blur Estimation

restoration filter estimates (3.19) by

$$f_E(x, y) = \begin{cases} 0 & \mathcal{F}^{-1} [G(u, v)H(u, v)] < 0, \\ 255 & \mathcal{F}^{-1} [G(u, v)H(u, v)] > 255, \\ \mathcal{F}^{-1} [G(u, v)H(u, v)] & \text{otherwise.} \end{cases} \quad (3.30)$$

This is known as the expectation step. The coefficients of the PSF given by (3.20) can be approximated from a discrete convolution

$$g(x, y) \approx A * f_E(x, y), \quad (3.31)$$

with the $f_E(x, y)$ estimating the new parameters of θ directly [4, 7]. This is known as the maximization step. By doing this EM algorithm, the nonlinear parameters of θ are approximated using the expectation step and the maximization step. By alternating between these two steps, a local optimum of the ML is obtained [7].

CHAPTER IV EXPERIMENT RESULTS

4.1 CMOS Chip Set

The photos in Figure 4.1(a)-(d) are four typical images from the sample set of forty-five that were taken of Dr. Monico's bookshelf in his office using a stv680 CMOS chip. When we look at the photos, we notice that none of the names of the books can be made out. All forty-five images were used in the super-resolution image process that were combined for the super-resolution image seen in Figure 4.1(f).

Figure 4.1(e) is the super-resolution process before any blurring is removed. While it is possible to make out more information than the low-resolution images, the image is not as clear as the super-resolution image seen in Figure 4.1(f).

Figure 4.1(f) is the super-resolution image after implementing an EM algorithm to remove any blurring in the image. Now it is possible to make out names on some of the books and even on the books you cannot make out names more detail about the books are noticeable.



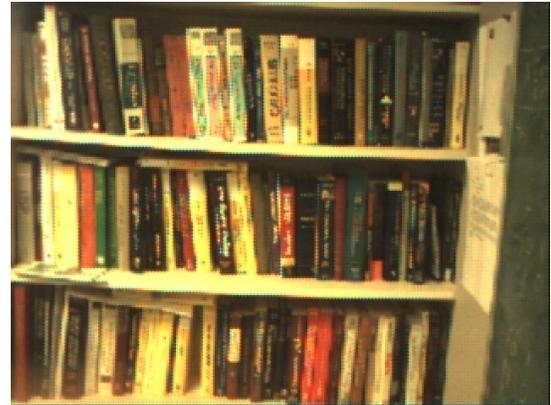
(a) Reference Image



(b) Sample Image



(c) Sample Image



(d) Sample Image



(e) Blurred SR Image



(f) SR Image

Figure 4.1: CMOS Example

4.2 MPEG-4 Set

The photos in Figure 4.2 are four typical images from the sample set of forty-seven that were taken of Dr. Monico's bookshelf in his office using a MPEG-4 compressed video stream captured from an Aiptek DZO-V5T. When we look at the photos, we notice that only one of the names of the books can be made out easily. All forty-seven images were used in the super-resolution image process that were combined for the super-resolution image seen in Figure 4.1(f).

Figure 4.2(e) is the super-resolution process before any blurring is removed. While it is possible to make out even more information than from the MPEG-4 compressed video stream low-resolution images, the image is not as clear as the super-resolution image seen in Figure 4.2(f).

Figure 4.2 is the super-resolution image after removing any blurring from the image by implementing an EM algorithm. Now it is possible to make out even more names on some of the books and even on the books you cannot make out names more detail about the books are noticeable.



(a) Reference Image



(b) Sample Image



(c) Sample Image



(f) Sample Image



(e) Blurred SR Image



(f) SR Image

Figure 4.2: MPEG-4 Video Example

CHAPTER V

CONCLUSIONS, FUTURE WORK

5.1 Conclusions

Super-resolution is the process of taking a set of low-resolution images and combining them into one or more higher-resolution that contains more information than any of the low-resolution images contain. To do this super-resolution experiment certain assumptions are made about the set of low-resolution images:

1. The set of low-resolution images are of the same scene with negligible differences in the scenes.
2. Each of the low-resolution images are offset in either position and/or rotation.
3. The sample size of the low-resolution images are sufficiently large for the extrapolation.
4. The SNR is sufficiently high, so that meaningful results at higher frequencies can be extrapolated.

Since each of the low-resolution images can be offset in rotation, it will be necessary to align the sample images to the reference image by a coarse angle alignment. To implement this coarse angle alignment, a DFT is used which will break down the sample image into magnitude and phase components. By using the magnitude of the polar DFT and making use of the relation from (2.15), the rotational angle can be determined by phase-correlation techniques [2].

After the sample image and the reference image are aligned using the coarse angle alignment, a finer relative offset is determined. First, a point-in-polygon algorithm is used to determine if the two image pixels have overlapping area. Second, by checking the distance between the reference image and sample image using (2.17), then the sample image is moved in the direction that minimizes the distance over the twenty-seven directions.

When the weighting functions and alignment have been determined, the super-resolution will still need to under-go a process to remove blurring. This process will be done by determining and applying a point-spread function to the blurred super-resolution image of well-known techniques. Several things must be known about the blurring function:

1. The energy conservation principle must be met by (3.25) and (3.26)
2. The symmetry of the point-spread function must be maintained

After all these things are done, a higher-resolution image is obtained from a set of lower-resolution images. This super-resolution has many applications, including astral-photography, facial recognition, and synthetic aperture radar.

5.2 Future Work

Interesting future work with alignment is to see if we can get a more accurate alignment using an iterative procedure. This could be done by aligning the sample image S_1 to the reference image R . Then set R_1 to be the stack of R and S_1 . Next, align S_2 to the stack R_1 . Then set R_2 to be the stack of R_1 and S_2 done from weighting R_1 . Continue this processes until all the sample images are stacked. Another alignment issue is to see if we can perform this super-resolution image with non-affine alignments, handling a moving camera relative to a fixed scene.

We would like to be able to use other basis functions such as trigonometric and polynomial basis functions. Also, we would like to see if this super-resolution can be performed directly in the frequency domain. We would like to develop better techniques for solving large systems such as using a maximum likelihood method approach for solving the coefficients. An analytic representation of the PSF dependent on choice of basis functions.

Finally, we would like to compare our approach with other methods such as iterated back-projection deblurring [17].

BIBLIOGRAPHY

- [1] Y. Altunbasak, A.U. Batur, B.K. Gunturk, M.H. Hayes III, and R.M. Mersereau. Eigenface-domain super-resolution for face recognition. *IEEE Transaction on Image Processing*, Vol. 12(Issue 5):pp. 597 – 606, May 2003.
- [2] A. Averbuch, Y. Keller, and Y. Shkolnisky. The angular difference function and its application to image registration. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 27(Issue 6):pp. 969 – 976, June 2005.
- [3] R. Berger and G. Casella. *Statistical Inference*. Duxbury, Pacific Grove, California, second edition, 2002.
- [4] J. Biemond and R.L. Lagendijk. *Iterative Identification and Restoration of Images*. Kluwer Academic Publishers, Boston, Massachusetts, first edition, 1991.
- [5] J. Biemond, R.L. Lagendijk, and A.M Tekalp. Maximum likelihood image and blur identification: a unifying approach. *Optical Engineering*, Vol. 29(Issue 9):pp. 422 – 435, 1990.
- [6] P. Bosdogianni and M. Petrou. *Image Processing: The Fundamentals*. Wiley, New York City, New York, first edition, 1999.
- [7] A. Bovik. *Handbook of Image & Video Processing*. Academic Press, San Diego, California, first edition, 2000.
- [8] O. Bretscher. *Linear Algebra with Applications*. Prentice Hall, Upper Saddle River, New Jersey, second edition, 2001.
- [9] B.N. Chatterji and S. Reddy. An FFT-based technique for translation, rotation, and scale invariant image registration. *IEEE Transaction on Image Processing*, Vol. 3(Issue 8):pp. 1266 – 1270, August 1996.
- [10] O. Ersoy. *Diffraction, Fourier Optics and Imaging*. Wiley, Hoboken, New Jersey, first edition, 2007.
- [11] G. Golub, P. Milanfar, and N. Nguyen. A computationally efficient superresolution image reconstruction algorithm. *IEEE Transactions on Image Processing*, Vol. 10(Issue 4):pp. 573 – 583, 2001.
- [12] G.H. Golub and C.F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, third edition, 1996.
- [13] R.C. Gonzalez and R. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, New Jersey, second edition, 2002.

- [14] L. Guan, S.W. Perry, and H. Wong. *Adaptive Image Processing: A Computational Intelligence Perspective*. CRC Press, New York City, New York, first edition, 2002.
- [15] E. Haines. *Graphics Gems IV*. Academic Press, San Diego, California, first edition, 1994.
- [16] R.M. Haralick. Digital step edges from zero crossings of second directional derivatives. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6(Issue 1):pp. 58 – 68, 1984.
- [17] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, Vol. 53(Issue 3):pp. 231 – 239, 1991.
- [18] A.K. Jain. Advances in mathematical models for image processing. *Proceedings of the IEEE*, vol. 69(Issue 5):pp. 502 – 528, 1981.
- [19] M. Kaveh and Y.L You. A regularization approach to joint blur identification and image restoration. *IEEE Transaction on Image Processing*, Vol. 5(Issue 3):pp. 416 – 428, 1996.
- [20] E. Kreyszig. *Advanced Engineering Mathematics*. Wiley, Hoboken, New Jersey, ninth edition, 2006.
- [21] P. Lemmerling and S. van Huffel. *Total Least Squares and Errors-in-Variables Modeling*. Kluwer Academic Publishers, Boston, Massachusetts, first edition, 2002.
- [22] J.B. Martens. *Image Technology Design: A Perceptual Approach*. Kluwer Academic Publishers, Boston, Massachusetts, first edition, 2003.
- [23] A. Rosenfeld. *Image Modeling*. Academic Press, New York City, New York, first edition, 1981.
- [24] J.W. Woods. Two-dimensional discrete markovian fields. *IEEE Transactions Information Theory*, vol. 18(Issue 2):pp. 232 – 240, 1972.
- [25] L. Yilong, C. Yuping, and Z. Lin. A super resolution SAR imaging method based on CSA. *Geoscience and Remote Sensing Symposium*, Vol. 6:pp. 3671 – 3673, June 2002.

APPENDIX

MATLAB CODE

The following MATLAB code was used to take a DFT of one of the low-resolution images and output the magnitude, log transformation of the magnitude, and the phase of the DFT. This code was used in Chapter II to produce the images of the DFT.

```
% This function finds the Fourier magnitude of an image and displays
% the Fourier magnitude. It makes use of mrgb2gray function that
% converts the image to grayscale. The mrgb2gray function is courtesy
% of Kristian Sveen and can be found at
% http://www.mathworks.com/matlabcentral/fileexchange

clc
close all
clear all

[filename, pathname] = uigetfile('*..*', 'Anyfile');
    % Gets the image filename
    % The image must be stored in the current directory
[img1, map1] = imread(filename);
    % Stores the image
gray_img = mrgb2gray(img1, 'mean');
    % Converts the image to grayscale
DFT_img = fft2(gray_img);
    % Takes the 2-D DFT of the grayscale image
center = fftshift(DFT_img);
    % Centers the 2-D DFT
center_mag = abs(center);
    % Finds the magnitude of the 2-D DFT

R = 0;
    % Sets the max magnitude equal to 0
[N M] = size(DFT_img);
    % Finds the size of the image
for i = 1:N
    for j = 1:M
        R1 = abs(center_mag(i,j));
        if R1 > R
            % Checks the max magnitude
            R = R1;
            % Sets the new max magnitude
        end
    end
end
```

```
        end
    end
end
    % The preceding loop is to find the max magnitude of the DFT

c = 255/(log(1+R));
    % Sets the constant for the log transformation

log_trans = c*log(center_mag+1);
    % Log Transformation of the magnitude
    % The addition of 1 insures that log(0) does not occur
imagesc(center_mag);
    % Outputs the magnitude
colormap(gray);
    % Makes the magnitude output grayscale
figure;
    % Creates another figure to display the next image
imagesc(log_trans);
    % Outputs the log transformation of the magnitude
imwrite(center,'Phase.jpg','jpeg');
    % Outputs the phase to a .jpg file
colormap(gray);
    % Makes the log transformation output grayscale
```

PERMISSION TO COPY

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Texas Tech University or Texas Tech University Health Sciences Center, I agree that the Library and my major department shall make it freely available for research purposes. Permission to copy this thesis for scholarly purposes may be granted by the Director of the Library or my major professor. It is understood that any copying or publication of this thesis for financial gain shall not be allowed without my further written permission and that any user may be liable for copyright infringement.

Agree (Permission is granted.)

Steven Lawless
Student Signature

28 November 2007
Date

Disagree (Permission is not granted.)

Student Signature

Date