Compressive Point Cloud Super Resolution

Cody S. Smith
Utah State University

COMPRESSIVE POINT CLOUD SUPER RESOLUTION

by

Cody S. Smith

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical Engineering

Approved:

Dr. Scott Budge Major Professor	Dr. Don Cripps Committee Member
Dr. Jacob Gunther Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2012

Abstract

Compressive Point Cloud Super Resolution

by

Cody S. Smith, Master of Science

Utah State University, 2012

Major Professor: Dr. Scott Budge
Department: Electrical and Computer Engineering

Automatic target recognition (ATR) is the ability for a computer to discriminate between different objects in a scene. ATR is often performed on point cloud data from a sensor known as a Ladar. Increasing the resolution of this point cloud in order to get a more clear view of the object in a scene would be of significant interest in an ATR application.

A technique to increase the resolution of a scene is known as super resolution. This technique requires many low resolution images that can be combined together. In recent years, however, it has become possible to perform super resolution on a single image. This thesis sought to apply Gabor Wavelets and Compressive Sensing to single image super resolution of digital images of natural scenes. The technique applied to images was then extended to allow the super resolution of a point cloud.

(66 pages)

Public Abstract

Compressive Point Cloud Super Resolution

by

Cody S. Smith, Master of Science

Utah State University, 2012

Major Professor: Dr. Scott Budge
Department: Electrical and Computer Engineering

A Ladar is a laser-based radar. With a Ladar, a laser is directed toward a scene. The laser light is reflected back. With this information, the distance to a given point on an object is obtained. By getting many such distance measurements using the laser, a scene can be mapped out and displayed in three dimensions as a point cloud or in two dimensions as a range image.

This thesis sought to apply the mathematical framework known as Compressive Sensing to increase the number of range values, or the resolution, that is present in a range image. This range image was then used to find the point cloud. By increasing the number of points in a point cloud, the objects in a scene became more clear.

Acknowledgments

I would like to thank Dr. Scott Budge for the trust he placed in me and the freedom to follow my ideas. His ability to point me in the right direction at key times kept this research alive. I would also like to thank Dr. Jacob Gunther for the many explanations and ideas that were given.

Cody S. Smith

Contents

List of Tables

List of Figures

Chapter 1
Introduction

As human beings, we are able to get information about the scene in which we live because we have two eyes. This gives us two different perspectives so that we can estimate how far a particular object is from us. This allows us to do many different things, including move about our environment in an effective way. Since we can tell how far objects are from us, we can discern what a safe path is for us to get from one point to another. We can also tell where an object is for us to pick up, or even throw a ball in a certain direction. Our ability to decipher three-dimensional information also helps us to distinguish one object from another.

If a computer were able to get three-dimensional information about a scene, then the computer would be enabled to perform similar functions as us. For example, if a robot could have knowledge of a scene in three dimensions, then it would be able to traverse a path across a room even if there were objects in the way. This is becoming an important feat for a computer with even more robots becoming available for consumers to purchase that perform menial tasks, such as vacuuming the carpet. Another good example would be if a car were able to get information about the environment in which it is travelling in three dimensions. If this were possible, the car could warn the driver if there were obstacles in the car's current path that the car might hit. This functionality is currently available in some vehicles. The technology could even progress to the point where a car could drive itself.

Another application of three-dimensional information about a scene is Automatic Target Recognition (ATR). The goal of ATR is to decide if an object in a scene is of interest for a particular purpose. For example, an unmanned aerial vehicle (UAV) may obtain information about a scene with a tank in the field-of-view. If the UAV could determine if this is a hostile tank, then the UAV may be able to act on the target with little human interaction. However, for the technology to progress to this point, the information about the scene must be sufficiently detailed and have a high level of accuracy.

1.1 Ladar Systems

One way that this information can be obtained about a scene is through the use of a Ladar system. Classically, a Ladar system shines a laser at different points in a scene. The length of time that it takes to receive the reflected light is then sensed by the system. With this information, the distance to that point can be detected. By combining many different points in a scene in this manner, a three-dimensional representation can be achieved. This gathering of points can be represented in a three-dimensional space as a point cloud. Recently, a form of Ladar, known as a flash Ladar, has become popular. A flash Ladar lights up the entire field-of-view with a single laser pulse. By doing this, all of the light returns from an entire scene are captured at one time, which allows the point cloud to be collected much more efficiently.

Another possible representation for the scene captured by the Ladar is known as a range image. A range image is a two-dimensional representation of the scene. In a range image, the value of the pixel is representative of the distance to a particular point in the scene. If the scene is represented in two dimensions, then image processing techniques can be implemented on the Ladar data.

1.2 Super Resolution

In image processing, it is common to increase the size of an image in order to see more detail in the image itself or in an area of the image. The simplest way to increase the size of an image is to use interpolation to estimate the missing pixel values. This method, while easy, does not effectively preserve the edges of the image. Because of this loss of edge information, the details in an image get lost as the image size is increased.

Super resolution is a technique to estimate the missing pixels in order to get improved resolution in an image. A good tutorial on super resolution methods is given by Park et al. [1]. The result is greater detail in the image. In order to perform super resolution, one generally needs to have multiple low resolution images taken of the same scene. Each low resolution image is offset from one another with sub-pixel accuracy. All of these low resolution images are then registered onto the same coordinate system. Once registered together, these low resolution images can be used to estimate the high frequency information necessary to complete a high resolution image.

1.3 Previous Work

There is also a variant of super resolution which acts on a single image. Single image super resolution finds a way of up-sampling the image so as to not introduce blur. There are many different ways to do this. Two possible formulations for single image super resolution are now described.

One way of setting up the single image super resolution problem is discussed by Yang et al. [2]. The image is broken up into patches. Each patch is then super-resolved and the resulting high resolution patches are put together into a single image. The super resolution is performed over a set of basis functions that have been trained in such a way as to ensure that the high and low resolution matrices of basis functions, known as dictionaries, have corresponding codes.

Once these dictionaries are found, then the optimal coefficients are found that correspond to the low resolution image patch from the image to be super-resolved. A high resolution patch is then found using these coefficients over the high resolution dictionary. These patches are then put together to form the total high resolution image.

Sen and Darabi performed single image super resolution on an entire image at one time [3]. A basis of Daubechies wavelets is used. A wavelet is a local harmonic function. In other words, the function starts at zero, has some shape repeating with some frequency, then goes back to zero [4]. The low resolution basis is created by blurring and downsampling the high resolution basis [3].

Multi-frame super resolution has been performed on a point cloud by Schuon et al. [5]. Many point clouds were taken from the same scene, and then these point clouds were registered onto the same coordinate system. From there, the high resolution point cloud was estimated from the many different low resolution point clouds of the scene.

Another variant of super resolution that was performed by Yang et al. is based on bilateral filtering [6]. This method registered one or two color images to be used as reference for an iterative method of refining the low resolution range image. The low resolution range image is used to construct cost volume to represent the ranges in the scene probabilistically. Then using the color image as reference, the low resolution range image is iteratively filtered to obtain the high resolution range image.

1.4 Problem Statement

Flash Ladar imaging systems obtain three-dimensional information about a scene. For applications using three-dimensional data, it is advantageous to have more detailed information about the scene. Since it is possible to represent flash Ladar data as a two-dimensional image, image processing techniques can be applied to the problem with little adaptation.

This thesis is exploring the super resolution via image processing techniques for Ladar data that requires only a single point cloud of the scene. This will result in a super-resolved point cloud that can be viewed in a three-dimensional viewer. In order to accomplish this goal, a new image processing technique for two-dimensional data will also be formulated.

This research is applicable to any system that uses a Ladar system to gather three-dimensional data about a scene in order to plan a path through a scene or make a decision regarding a particular object in the scene, such as an ATR application.

1.5 Contributions

The contributions of this thesis are listed below.

1. Application of Gabor wavelets to two dimension single image super resolution.

2. Weighted averaging to combine image patches into a single image.

3. Application of the two-dimensional processing technique known as super resolution to increase the resolution of Ladar data in three dimensions.

4. Application of Compressive Sensing to Ladar data.

1.6 Outline

The rest of this thesis proceeds as follows. Chapter 2 gives the necessary background to understand the rest of this research. Chapter 3 describes the methods used to perform single image super resolution on a 512x512 digital image (two-dimensional data). Chapter 4 then describes the necessary modifications to the methods in Chapter 3 to super resolve a Ladar point cloud (three-dimensional data). Chapter 5 then gives a brief conclusion.

Chapter 2
Background

This chapter will give necessary background on the software used to simulate data for this research. Background will also be given on the theory that makes the solution methods used in this thesis possible.

2.1 Ladarsim

Ladarsim is a Matlab software package that allows for the simulation of different Ladar systems. Its main use is for system engineers who are designing new Ladar systems to simulate the performance of a particular hardware configuration before the investment is made to build the actual system. The parameters can easily be changed in order to quickly get an idea of how the system will perform once built.

In order to simulate a particular Ladar, the system parameters need to be entered. It is possible to simulate custom scan dynamics, scan elements, a custom focal plain array, optics, and transceiver performance. Once these parameters have been set, the Ladar system can be placed anywhere in a three-dimensional environment. Then the system can be set to move in a certain path and scan at a certain rate.

The interface provides for many different scenes to be created from separate object files. In order to incorporate new objects into the system, the Wavefront .obj file format is recognized. A scene file is generated and placed in the three-dimensional environment with the Ladar system. The scene can contain many different objects. The objects in the scene can even be configured to move within the scene if it is advantageous to do so. It is this scene that the Ladar system will scan in order to produce a point cloud. Ladarsim will accurately simulate the interaction of the laser ray with objects that may be in the scene. These objects may include trees, tanks, or other vehicles.

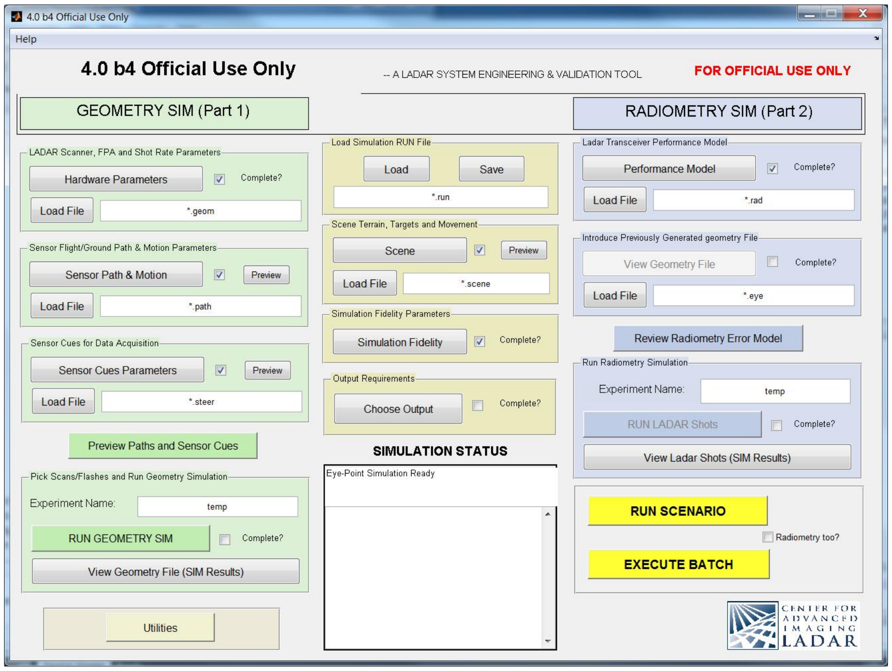
The user interface to Ladarsim is shown in Fig. 2.1. The left side gives the necessary parameters to change the geometry of the sensor and path of the sensor in the scene. The right side allows the parameters of the Ladar transceiver to be simulated. The middle section allows for the scene to be changed as well as to specify the type of output that is given by Ladarsim.

2.2 Gabor Wavelets

There are many ways of forming a matrix of basis functions over which to find a representation for an input signal. It is popular to use wavelets today of varying types. Today there is much research happening in the application of Gabor wavelets. These wavelets are based upon Gabor filters [7]. The Gabor filter was introduced by Dennis Gabor in 1946 [8]. The definition of the Gabor filter extended from the observation that many real life signals seem to be best defined by a combination of the time and frequency domains. An example of this is when one hears a siren, it is observed that the frequency of the sound is changing with time. Applying the theory of Quantum Mechanics to Communication Theory, Gabor found that there is an uncertainty in knowing the frequency of a signal that is inversely proportional to the duration of the signal. This led to the definition of a signal that would minimize the uncertainty between the duration of the signal in time and the frequency of the signal. Gabor found that the shape of the signal that minimizes this uncertainty is a Gaussian envelope overlaying a sine wave.

Gabor wavelets are self-similar objects, meaning that they can be generated through dilation and rotation of one wavelet known as the Mother wavelet [8]. In one dimension, an example Gabor wavelet in the time domain would appear as in Fig. 2.2.

The one-dimensional Gabor wavelet minimizes the uncertainty of representing a signal in the time and frequency domain. If these results are extended into the two-dimensional plane of spatial time in the x and y directions as well as frequency in the x and y directions, the result is the two-dimensional Gabor filter. It has been shown that a two-dimensional Gabor wavelet is of similar shape as the shape of the response of a simple cell to light energy in the mammalian visual cortex [9]. Lee showed that a compilation of of Gabor wavelets


4.0 b4 Official Use Only
Help
4.0 b4 Official Use Only
-- A LADAR SYSTEM ENGINEERING & VALIDATION TOOL
FOR OFFICIAL USE ONLY
GEOMETRY SIM (Part 1)
RADIOMETRY SIM (Part 2)
LADAR Scanner, FPA and Shot Rate Parameters
Hardware Parameters
Complete?
Load File
Sensor Flight/Ground Path & Motion Parameters
Sensor Path & Motion
Preview
Sensor Cues for Data Acquisition
Sensor Cues Parameters
Preview Paths and Sensor Cues
Pick Scans/Flashes and Run Geometry Simulation
Experiment Name:
RUN GEOMETRY SIM
View Geometry File (SIM Results)
Utilities
Load Simulation RUN File
Load
Save
Scene Terrain, Targets and Movement
Scene
Simulation Fidelity Parameters
Simulation Fidelity
Output Requirements
Choose Output
SIMULATION STATUS
Ladar Transceiver Performance Model
Performance Model
Introduce Previously Generated geometry File
View Geometry File
Review Radiometry Error Model
Run Radiometry Simulation
RUN LADAR Shots
View Ladar Shots (SIM Results)
RUN SCENARIO
Radiometry too?
EXECUTE BATCH
CENTER FOR ADVANCED IMAGING LADAR

*.geom
*.path
*.steer
Experiment Name: temp
*.run
*.scene
*.rad
*.eye
Experiment Name: temp
Eye-Point Simulation Ready

Fig. 2.1: Ladarsim interface.

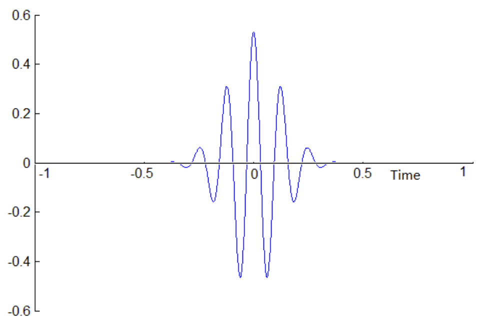


Fig. 2.2: One-dimensional Gabor filter.

can be used to successfully represent an image [7]. In a two-dimensional space, the equation to describe a Gabor wavelet is

$$G(x, y) = \frac{1}{2\pi\sigma\beta} e^{-\pi \frac{(x-x_0)^2}{\sigma^2} + \frac{(y-y_0)^2}{\beta^2}} e^{\imath[\xi_0 x + \nu_0 y]}. \tag{2.1}$$

In this equation, the ξ_0 and ν_0 terms control the spatial frequency in the x and y directions, respectively.

Equation (2.1) has a real and an imaginary part. The real part of this result is a cosine with frequency ξ_0 with a Gaussian envelope. The imaginary part is a sine wave with a frequency of ν_0 with a Gaussian envelope. The x-direction variance is controlled by σ and y-direction variance is controlled by β. By constraining σ and β to be equal, the wavelet will be circular. The image shown in Fig. 2.3 is an example of a two-dimensional Gabor wavelet.

2.3 Dictionary Training

A set of basis functions must be found in order to represent a signal. One way of forming a set of basis functions is to generate a general set of basis functions, such as a set of Gabor wavelets. This basis would not depend upon the type of signal that is being represented. Another way of generating the necessary basis functions is to exploit a field of research known as dictionary learning. The goal of dictionary learning is to find a set of

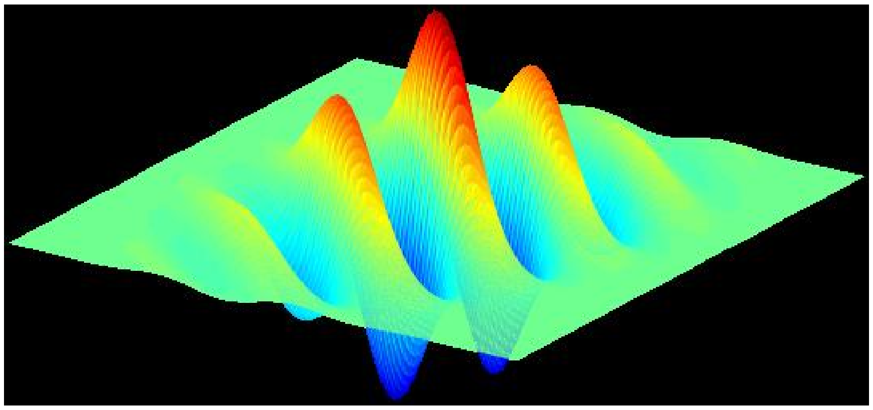


Fig. 2.3: Two-dimensional Gabor filter.

basis functions that allow the reconstruction of a signal sparsely with low error. To represent a signal sparsely in some basis is to represent the signal with as few coefficients as possible. Conversely, this means that most of the entries in the coefficients of the reconstruction are zero. The number of basis functions that is found is controlled by the user. Each column of the dictionary is a basis element with unit norm, called an atom. Each column of the dictionary must have a norm of less than or equal to one to prevent the norm of each atom of the dictionary growing towards infinity to drive the values of α (as shown in (2.2)) to zero.

Dictionary learning solves

$$\min_{D,\alpha} \quad \frac{1}{n}\sum_{i=0}^{n}(\frac{1}{2}\|x_i - \mathbf{D}\alpha_i\|_2^2 + \lambda\|\alpha_i\|_1) \tag{2.2}$$

for a set of input training example signals x_i. A set of basis functions found in this way changes with the type of signal to be reconstructed as represented by the example signals x_i.

This problem is not convex in both **D** and α at the same time. This means that there is not an efficient method to solve this problem. However, it is convex in each variable individually while the other is held constant. So in order for convergence of the algorithm, a separate problem minimizing over the vector α must first be solved while holding D constant. Then while holding α constant, another problem minimizing over the matrix D must then be solved. These two problems are then iterated until the error gets below a set threshold value or the dictionary matrix D changes by only a small amount [10]. An open source software package has been developed by Mairal et al. call Sparse Modeling Software (SPAMS). This software package is available under the GPLv3 license. SPAMS is written in C++ and is currently available with a Matlab, Python, and R interface [10].

2.4 Compressive Sensing

When sampling data, it is typically assumed that the Nyquist sampling rate must be met. That is to say that the signal must be sampled at a rate of twice the maximum frequency for faithful reconstruction. This is known as the Nyquist Sampling Theorem. However, it would be very beneficial to sample a signal at less than the Nyquist sampling rate. Sampling at less than the Nyquist sampling rate leads to a system of equations where there is not enough data to reach a unique solution. This is an under-determined system of equations. Using standard least squares methods, under-determined systems cannot be solved. However, Compressive Sensing (CS) is a framework that provides a way for a solution to be found to these ill-posed problems. There are two important assumptions in the CS framework, namely sparsity and incoherence.

Assuming that a given signal is sparse in some basis is useful in solving linear systems of equations. A signal is sparse if it can be represented in a certain basis with many coefficients being zero. If a signal can be represented with a certain number of coefficients k, it is said to be k-sparse. Sparsity provides a way for many compression algorithms.

Another large part of CS is incoherence in the sampling and representation basis. This is to say that there is a low amount of correlation in the sampling and the representation basis. As a measure of the coherence is given by "An Introduction To Compressive Sampling" [11] as

$$\mu(\theta, \psi) = \sqrt{n}\ \max |\langle \theta_i, \psi_j \rangle|. \tag{2.3}$$

Take the standard Fourier Transform as an illustration. The sampling basis are delta functions and the representation basis are complex sinusoids. By taking the sampling basis as θ and complex sinusoids as ψ we get a coherence of 1. This is the standard method of sampling. This is the maximum incoherence that is possible. The lower the coherence the better for the reconstruction of the signal [11].

2.4.1 Restricted Isometry Property

The Restricted Isometry Property (RIP) is another key to finding the solution of the under determined sets of matrices that have been discussed thus far in this section. The

RIP is said to be true for a matrix of basis functions A if

$$(1-\epsilon)\|v\|_2^2 \;\leq\; \|Av\|_2^2 \;\leq\; (1+\epsilon)\|v\|_2^2 \tag{2.4}$$

is true for all k-sparse (a vector with only k elements being nonzero) vectors v when $1 \leq \epsilon \leq \sqrt{n}$. This means that the k-sparse vectors are not in the null space of A. Since it is the k-sparse vectors that are required to be found for reconstruction, they cannot be in the null space of the solution basis.

If this is true, then any k columns of A form an approximately orthogonal basis. One can also look at this equation as stating that the length of vectors is preserved in the basis space. The RIP guarantees that the L_1 norm minimization problem has an efficient algorithm that converges to the solution.

Now the question becomes how many measurements are necessary to complete the reconstruction and how sparse vector v needs to be. The constraint on these parameters is given by

$$m = O(k\ logn). \tag{2.5}$$

In this equation, n is the size of the solution vector, m is the number of measurements that are available, and k is the sparsity level. The RIC needs to be true for all vectors that are $2{*}k$ sparse with $\epsilon = \sqrt{2} - 1$.

2.4.2 Solution Methods

In order to solve for the coefficients to reconstruct some input signal y over some set of basis functions A, the following problem must be solved.

$$\min_x \ \|x\|_1 \qquad subject\ to \qquad Ax = y \tag{2.6}$$

By minimizing the one norm, we are finding a sparse solution, as opposed to the two norm which tends to spread the power over many coefficients. The solution to this problem will reconstruct the input signal if the RIP is met. Equation (2.6) is a linear program, which can be solved using efficient algorithms.

The above formulation is for exact reconstruction of the input signal y. The problem can also be formulated for solution in the presence of noise by writing

$$\min_x \ \|x\|_1 \quad \textit{subject to} \quad \|Ax - y\|_2 \leq \epsilon. \tag{2.7}$$

Equation (2.7) can also be solved using efficient algorithms because it is a convex problem, specifically a second-order cone problem. This problem is commonly known as the LASSO which is short for least absolute shrinkage and selection operator. This problem can also be written as

$$\min_x \|Ax - y\|_2^2 + \lambda\|x\|_1, \tag{2.8}$$

where $\lambda \in [0, 1]$.

In this formulation λ is a parameter that allows for the trade-off of the standard least-squares solution and the sparse solution. If λ is zero, then the least squares solution is found. If λ is one, then the sparse solution is found. The closer the parameter value is to one, the closer the more sparse the solution becomes.

While this problem can be solved efficiently, the running time of the problem can still be extensive. This problem can also be solved using a class of solution known as greedy algorithms [11]. Greedy algorithms do not solve the actual minimization problem, but are a heuristic for finding the solution. In a greedy method, instead of finding the global optimal solution, at each iteration of the algorithm the locally optimal choice is performed. A popular example of a greedy algorithm is Orthogonal Matching Pursuit (OMP) [12]. OMP requires forward matrix, A in the above examples, and the backward matrix. The backward matrix in the above example would be A^*, or the adjoint of A. The backward matrix is used to find the approximation to the solution vector over the basis A. Once the solution approximation is found, the error from this to the input signal is found. Then the index of the column of A that reduces the error by the greatest amount is stored. This is repeated until a collection of k indices is found, corresponding to the k-sparse solution. The indices of A that reconstruct the input signal is returned. The columns of A corresponding to these indices are taken and the least squares solution, through utilization of the Moore-Penrose pseudoinverse, over these columns is found that approximate the solution.

Chapter 3
Two-Dimensional Super Resolution

Sen and Darabi performed super resolution on an entire image by forming a high resolution dictionary of wavelets. Then through blurring and downsampling, the low resolution dictionary was created [3]. Yang et al. performed super resolution on an image by splitting the input image into patches and utilizing a trained dictionary [2]. This chapter combines the idea of splitting up the image into patches to be super-resolved and the idea of blurring and downsampling the high resolution dictionary to create the low resolution dictionary. The application of Gabor wavelets to this process is a new technique to be explored. The images that are going to be used for testing are natural images and can be downloaded from http://redwood.berkeley.edu/bruno/sparsenet [13].

3.1 Problem Formulation

A common size for two-dimensional digital images is 512x512 pixels. This is the size of image on which the following super resolution algorithm will be performed. Processing of images involves vectorizing the input, which for a full image yields a vector y with 262144 elements. By examining

$$\min_x \|Ax - y\|_2^2 + \lambda \|x\|_1, \tag{3.1}$$

it is seen that the matrix of basis elements A would have at least 262144 rows. If matrix A were to be overcomplete, this matrix would be required to have even more columns than rows. This is a problem that has significant memory requirements as well as significant processing resources, as the solution methods of this problem requires the calculation of the gram matrix. In order to simplify the necessary memory and processing requirements of this problem, breaking up the problem into many smaller problems is necessary. The super resolution of a single large image can also be viewed as the super resolution of many smaller images. The size of each individual patch needs to be small enough to make processing efficient, since this problem will have to be solved many times. With this limiting factor in mind, the size of each patch should certainly be less than 64x64, which would result in a vector of size 4096.

We are working with full images, and so it is desirable for the size of the patch to go into the size of the full image evenly. As a starting point to finding the patch size, take the simplest possible method and find values that are less than 64 that go into 512 evenly. This will give the following values: 2, 4, 8, 16, 32, and 64. The point of this research is to do super resolution, so this patch will be down sampled. A patch size that leaves enough information about the scene to infer the higher resolution components is necessary, which allows 2, 4, and 8 to be eliminated.

However, when a single image is broken up into smaller patches for separate processing, it the "patchiness" of the resulting output image can be an issue. That is to say that the separate patches do not blend back together well. In order to mitigate this effect, the patches are chosen such that each patch overlaps the patches around it. In order to find a patch size that will still go into 512 evenly, the overlapping sections must be taken into account. This can be done by simply subtracting out the size of the overlap from the overall image size and patch size before dividing. This would give

$$B = \frac{imageSize - overlap}{patchSize - overlap}, \tag{3.2}$$

where B is the number of patches to be processed in order to go across the entire image. This is assuming that the patches are processed in a left to right raster scan fashion. By calculating B for an image size of 512 with the potential patch sizes that were determined above and sweeping the overlap size from 2 pixels to 14 pixels, it is found the for a patch size of 16 pixels an overlap of 8, 12, or 14 pixels would work. For a patch size of 32 pixels, an overlap of 8 or 12 pixels would work. Were a patch size of 64 pixels employed, an overlap of 8 pixels would work. A patch size of 16 pixels would mean that the minimum amount of overlap is half the patch size. Since the solver is able to solve faster for fewer variables, a smaller patch leads to a smaller dictionary (more on dictionary generation in the next section), and smaller patches seem to work better, a patch size of 16x16 will be used with an overlap of 8 pixels.

3.2 Dictionary Generation

Gabor wavelets in a two-dimensional space are to be used as the basis functions for the dictionary to be used for super resolution of digital images. Gabor wavelets were introduced in Section 2.2. Some modifications of this equation are necessary, however, in order to generate the dictionary. The sine wave needs to be allowed to rotate in the two-dimensional space, and the nonzero DC component needs to be out. These concerns are addressed by Lee [7]. The equation for two-dimensional Gabor wavelets is

$$\Psi(x, y, \omega_0, \theta) = \frac{\omega_0}{\sqrt{2\pi}\kappa} e^{-\frac{\omega_0^2}{8\kappa^2}(4(x\cos\theta + y\sin\theta)^2 + (x\cos\theta + y\sin\theta)^2)} [e^{\imath[\omega_0 x\cos\theta + \omega_0 y\sin\theta]} - e^{-\frac{\kappa^2}{2}}], \quad (3.3)$$

where

$$\kappa = \sqrt{2ln2}\frac{2^\phi + 1}{2^\phi - 1}.$$

Here ϕ is the bandwidth in octaves that the set of wavelets covers. Equation (3.4) can be thought of as a sine wave of frequency ω_0 that has been rotated in spatial orientation by θ with a two-dimensional Gaussian function overlaying it with a variance of $\frac{\kappa^2}{\omega_0^2}$. This formulation has twice the variance in the x-direction.

For a given θ and ω_0, the wavelet can be commuted over the spatial extent of the patch by using the equation from Lee [7], which is

$$\Psi_{m,n,k,l}(a_0^{-m}x - nb_0, a_0^{-m}y - kb_0). \quad (3.4)$$

Lee applies a scaling of a_0^{-m}, but for this thesis, each wavelet is vectorized and scaled such that each column of the dictionary has unit norm. For this reason, the scaling is left off.

Using (3.4), a series of Gabor wavelets can be be generated by setting $a_0^m = 2^m$,

$b_0 = 0.0625$, ranging n and k from 0 to 15, ranging θ over six evenly spaced values from 0 to π, and sampling at frequencies given by $f = 2^{-m/3}$, and ranging m from 0 to 5. Sampling at these frequencies leads to $\phi = 1.5$. This leads to $16 * 16 * 6 * 6 * 6 = 55296$ wavelets. By using a b_0 of 0.0625, it is assumed that the wavelet is ranging over 0 to 1 in the x and y directions.

Each wavelet will be 16x16 pixels in size. Each wavelet is vectorized into a vector of 256 pixels, each called an atom. By placing the atoms in columns, a high resolution matrix D_h of size 256x55296 will be generated. This matrix is the dictionary of basis functions from which the optimization problem can draw to represent the input image.

3.3 Application of Compressive Sensing

In Section 2.4, the constraints were discussed for the number of measurements that are necessary in order for the recovery to be made possible by the CS framework. These constraints can be met in order to provide a recovery method for single image super resolution for a dictionary of Gabor wavelets.

The dictionary described above is to represent the high resolution image patches. A low resolution dictionary needs to be generated that has atoms that correspond to the high resolution dictionary in order to perform the super resolution. This dictionary can be generated by downsampling each image patch in both the row and column space. The two-dimensional wavelet can be represented as

$$P = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,16} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,16} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{16,1} & a_{16,2} & a_{16,3} & \cdots & a_{16,16} \end{bmatrix}, \tag{3.5}$$

where each $a_{i,j}$ is a pixel making up the wavelet in two dimensions. In other words, this matrix is an image of the wavelet.

When matrix P is vectorized, it will become

$$P_v = \begin{bmatrix} a_{1,1} \\ a_{2,1} \\ a_{3,1} \\ \vdots \\ a_{16,1} \\ a_{1,2} \\ a_{2,2} \\ a_{3,2} \\ \vdots \\ a_{16,2} \\ \vdots \\ a_{1,16} \\ a_{2,16} \\ a_{3,16} \\ \vdots \\ a_{16,16} \end{bmatrix}. \tag{3.6}$$

The downsampling of the dictionary D_h from the previous section can be performed as a matrix multiply with a properly formed matrix. This matrix will have the number of columns as the high resolution input vector and have the number of rows as the low resolution vector. To illustrate the form of this matrix used for downsampling, for a matrix S as

$$S = \begin{bmatrix} \delta & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \delta & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \delta & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \delta & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \delta & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \delta & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \delta & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \delta & \mathbf{0} \end{bmatrix}, \tag{3.7}$$

where

$$\delta = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{3.8}$$

In matrix given in (3.7), 0 represents a matrix of size 16x16 that is full of zeros. These matrices are set up to downsample by a factor of two, but it illustrates the concept. It is worth noting that with this formulation for super resolution, it is simple to generate a low resolution dictionary for different super resolution factors.

With the above matrices having been defined, the expression for the downsampled dictionary D_{dn} would be given by (3.9).

$$D_{dn} = SD_h \tag{3.9}$$

With the generation of the low resolution dictionary, the coherence can be measured using the equation found in "An Introduction To Compressive Sampling" [11], which is

$$coherence = \sqrt{n} * \max_{1 \leq k,j \leq n} \langle S_k, D_{dn,j} \rangle. \tag{3.10}$$

For the Gabor dictionary as formed above, the coherence with the sampling matrix S is found by taking

$$coherence = \sqrt{n} * \max D_{dn}^T S \tag{3.11}$$

is calculated to be 0.7871.

However, the theory of CS says that a basis with more incoherence in the representation and the sampling matrices will work better for reconstruction of a signal. This implies that a matrix that has a smaller value for the coherence measurement will perform better in the CS framework. This can be done by low pass filtering the matrix D_h in the frequency domain before the downsampling is performed, which will blur the image in the spatial domain. In order to low pass filter the matrix in the frequency domain, a DFT matrix can be used to transform the matrix to the Fourier domain, then the conjugate of the DFT matrix can transform it back. If G is a matrix with a Gaussian on the main diagonal, this can be written as

$$\phi = \mathfrak{F}^* G \mathfrak{F}. \tag{3.12}$$

In this equation, $\mathfrak{F}$ is a DFT matrix. The variance of the Gaussian on the main diagonal is 1666.7. This is the variance of a Gaussian in the frequency domain that corresponds to a Gaussian with a variance of four in the spatial domain [3].

In order to blur out an individual image patch (a matrix) P in both row and column directions, the row space and the column space both need to have ϕ applied as

$$P_{blur} = \phi P \phi^T. \tag{3.13}$$

This is equivalent to taking the two-dimensional Fourier Transform of the patch, low pass filtering in the frequency domain, and then taking the inverse Fourier Transform.

The image patch matrices all get vectorized when the high resolution dictionary is formed. This means that in order to blur out an image patch that has been vectorized, the properties of the Kronecker Product can be useful. A matrix Φ can be formed so as to blur the row and column spaces with one operator applied to the vectorized image patch. This matrix is formed by

$$\Phi = \phi \otimes \phi, \tag{3.14}$$

where $\otimes$ denotes the Kronecker Product.

With the matrix Φ having been defined, the dictionary D_h can be blurred in the row and column space and downsampled as in

$$D_l = S\Phi D_h. \tag{3.15}$$

The matrix D_l is now a matrix that can be used to represent low resolution images. These same coefficients can then be used to generate a high resolution image of the same scene.

The point of all of this calculation was in to reduce the coherence measurement in order to make the dictionary of Gabor wavelets better suited for use in the CS framework. The coherence of the D_l with the sampling matrix S is calculated to be 0.7774. So the coherence has been decreased. The small decrease is because of the wide extent of the Gaussian used to blur the dictionary.

The constraints were given on the number of measurements m, the size of the output vector n, and the sparsity level k in (2.5).

As was mentioned, (2.4) (the RIP) needed to be true for $2{*}k$ sparse with an $\epsilon = \sqrt{2} - 1$. In this problem formulation, a patch size of 16x16 has been chosen for an overall vector size of 256 measurements of the scene represented in the selected patch. For the super resolution, let's examine a downsample factor of four. By downsampling the patch to a size of 4x4, an overall vector size of 16 measurements to represent the scene in the patch is all that is available for reconstruction of the full size image patch. If we solve (2.5) for k, k comes out to be on the order of seven.

The Restricted Isometry Property (RIP) is a major part of the CS framework. The equation given in (2.4) must be true for all vectors that are 2*k sparse when k is given by (2.5) and $\epsilon = \sqrt{2} - 1$. Equation (2.5) gives a k value on the order of 4.168. To test how the blurred and downsampled dictionary meet the RIP, a k-sparse vector (where the entries were randomly chosen) is used to test the blurred and downsampled dictionary against the RIP, as well as a downsampled dictionary against the RIP. Out of 10000 vectors with seven random entries being nonzero, the RIP was true for 989 vectors for the the downsampled dictionary. For the same 1000 vectors tested, the RIP was true for 992 vectors for the blurred and downsampled dictionary. This is a small increase, again, is because the blur function is really wide. This further confirms the use of the blurred and downsampled dictionary in CS.

3.4 Generating a Super-Resolved Image

The high resolution image is first blurred by convolving with a two-dimensional Gaussian kernel with variance of four and then downsampled by a factor of four to create the low resolution image. Then patches of size 4x4 pixels are taken out of the low resolution image for use in the super resolution algorithm. The mean value of each low resolution patch is then subtracted from the patch, so the patch values have a mean of zero. Then each patch is vectorized and normalized so as to have unit norm. These are the values used in the optimization problem.

The optimization problem gives the coefficients to reconstruct the individual low resolution image patch by solving (3.1), the corresponding high resolution patch can be found by using these same coefficients on the high resolution dictionary. The reconstructed high resolution patch needs to have the dynamic range extended back to that of the original low resolution patch before the mean was subtracted out and normalization. In order to ensure the dynamic range of the high resolution patch is correct, the max and min values are found from the low resolution input. The high resolution patch is estimated from the low resolution coefficients on the high resolution dictionary. Once the estimated high resolution patch is found (call this $\hat{X}$), the dynamic range of the patch $\hat{X}$ can be stretched by stepping through the following steps.

1. Subtract the smallest value from $\hat{X}$, so now the smallest value in $\hat{X}$ is zero. Call the smallest value ζ.

2. Divide by the largest value in $\hat{X}$, so that now the largest value in $\hat{X}$ is one. Call this value β.

3. Multiply $\hat{X}$ by the quantity $\beta - \zeta$.

4. Add ζ to $\hat{X}$.

The result of this process is an image patch that has the same maximum and minimum range values as the low resolution input image.

This is performed many times for each patch that makes up the entire image to be super-resolved. These different high resolution patches need to be blended together into a single image. Because the input image was separated in order to solve smaller problems, "patchiness" may be a problem. This could be described as being able to see where one patch ends and the next begins, and may be seen as lines running through the image in a checker board like manner. An overlap of eight pixels was chosen on each edge. These pixels must be blended back together in a way so as to minimize this effect. Two options to accomplish this are to average the overlapping pixel values, another is to average the overlapping pixels in a weighted manner.

To average the overlapping pixels is simple, but it may be possible to weight the pixels as they are averaged together in order to improve performance by decreasing the "patchiness" described above. The pixels from the patch that is closer to the center of the patch will be weighted higher than those pixels on the edge of the patch. The sum of the weighting coefficients should be one. For an overlap of eight pixels, the weighting average in the horizontal direction on the right side becomes

$$center \begin{bmatrix} 0.8 & 0.7 & 0.6 & 0.5 & 0.5 & 0.4 & 0.3 & 0.2 \end{bmatrix} edge. \tag{3.16}$$

In order for this weighting scheme to be valid (for each respective element to sum to one) the left weighting values become

$$edge \begin{bmatrix} 0.2 & 0.3 & 0.4 & 0.5 & 0.5 & 0.6 & 0.7 & 0.8 \end{bmatrix} center. \tag{3.17}$$

These values would be extended vertically along the entire left or right side of the respective image patch to be tied into the entire image. The weighting in the vertical direction on the bottom becomes

$$\begin{array}{c} center \\ \begin{bmatrix} 0.8 \\ 0.7 \\ 0.6 \\ 0.5 \\ 0.5 \\ 0.4 \\ 0.3 \\ 0.2 \end{bmatrix} \\ edge \end{array}. \tag{3.18}$$

As in the previous case, the top weighting scheme becomes

$$\begin{array}{c} edge \\ \begin{bmatrix} 0.2 \\ 0.3 \\ 0.4 \\ 0.5 \\ 0.5 \\ 0.6 \\ 0.7 \\ 0.8 \end{bmatrix} \\ center \end{array}. \tag{3.19}$$

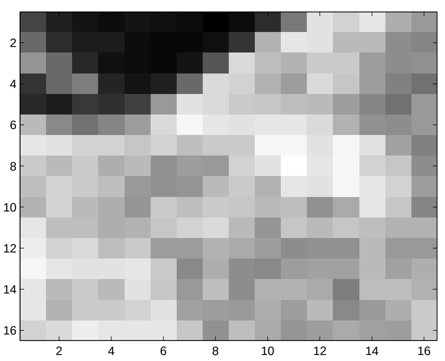
As before, these values are extended horizontally along the entire top or bottom of the image.

In performing these operations on every side of every patch leads to sections of each patch that need to be averaged in the horizontal and in the vertical directions. This happens in each 8x8 sized corner section. In order to properly handle these areas, first the two corners that overlap from patches that are on the same row are averaged according to the weighting scheme just described in the horizontal direction. Once this is complete, the two resulting corner patches that overlap that are on the same column are then averaged according to the weighting scheme in the vertical direction.

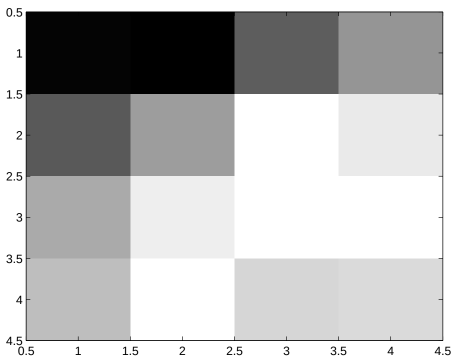
The low resolution image is generated from the high resolution image by convolving with a Gaussian with a variance of four and then taking every fourth sample (a downsample of four).

The first step to super resolving an image is to super resolve an individual patch. The results of super resolving an individual patch is given in Fig. 3.1.

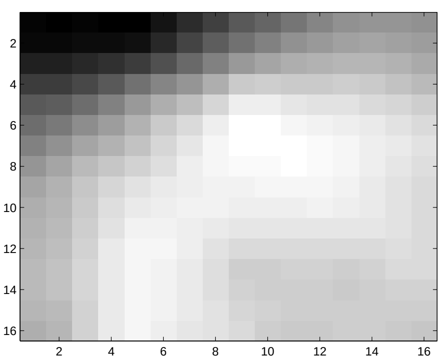
The results of super resolving an entire image are given in Table 3.1. These results are generated by using the Matlab software package called SGPL1. This software was



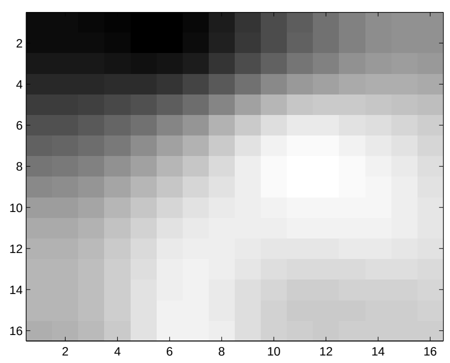
(a) Original High Resolution Patch



(b) Original Low Resolution Patch



(c) Reconstructed High Resolution Patch



(d) Bicubicly Interpolated Patch

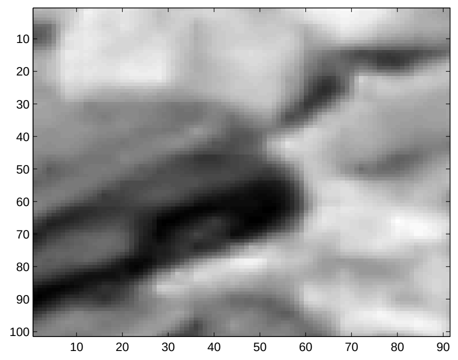
Fig. 3.1: Patch comparison.

developed by E. van den Berg and M. P. Friedlander [14,15]. As can be seen by these results, the difference in the averaged and the weighted averaging error is negligible. However, by examining each image close up, it can be seen that the images produced by weighted averaging has a significant increase in how smooth the final image is. In other words, the patches blend together much better if weighted averaging is applied. This can be seen by comparing Fig. 3.2(a) and Fig. 3.2(b). The corresponding area from the bicubicly interpolated image is shown in Fig. 3.2(c). The corresponding area from the original image is shown in Fig. 3.2(d).

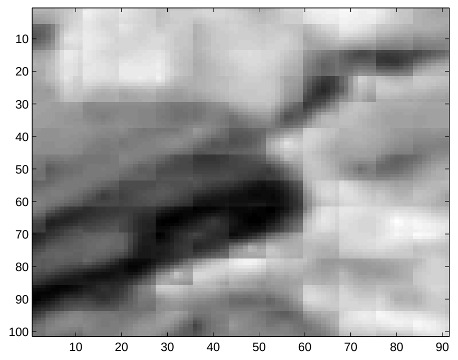
An full size image was upsampled with bicubic interpolation is seen in Fig. 3.3. The same image was super-resolved image using weighted overlapping can be seen in Fig. 3.4. The super-resolved image using averaged overlapping can be seen in Fig. 3.5. In comparing these images to the original image seen in Fig. 3.6. The MSE improved little, so an obvious improvement is difficult to see with the eye.

Table 3.1: SGPL1 image SR results.

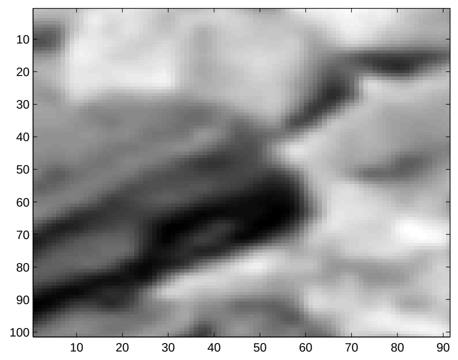
Image	SGPL1 MSE Weighted	SGPL1 MSE Averaged	Bicubic MSE
Image 1	0.2124	0.2210	0.2343
Image 2	0.0769	0.0821	0.0841
Image 3	0.3662	0.3829	0.3872
Image 4	0.1064	0.1090	0.1097
Image 5	0.5734	0.5928	0.5880
Image 6	0.1994	0.2035	0.2040
Image 7	0.2301	0.2456	0.2388
Image 8	0.0578	0.0591	0.0596
Image 9	0.0545	0.0555	0.0562
Image 10	0.0157	0.0164	0.0167



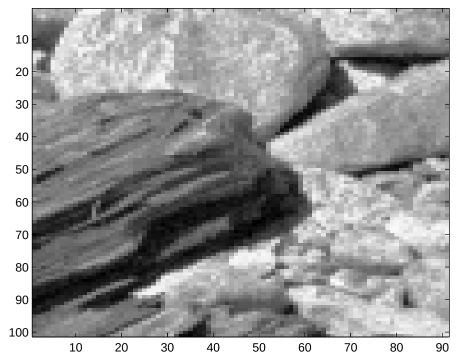
(a) Weighted Overlap



(b) Averaged Overlap



(c) Bicubicly Interpolated



(d) Original

Fig. 3.2: Patch overlap comparison.

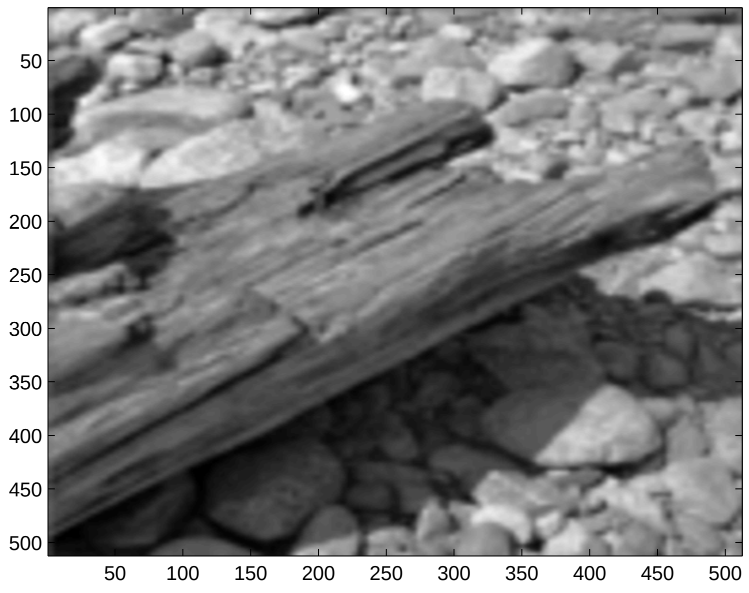


Fig. 3.3: Bicubicly interpolated image.

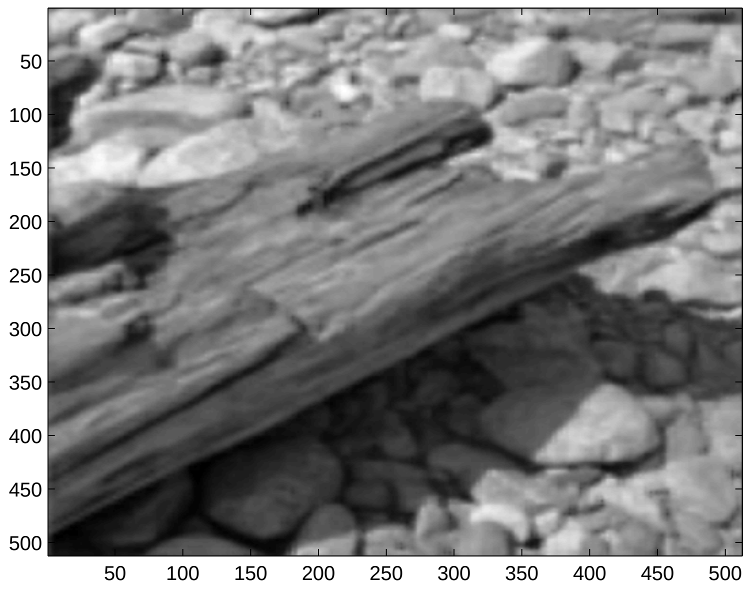


Fig. 3.4: Super-resolved weighted overlap image.

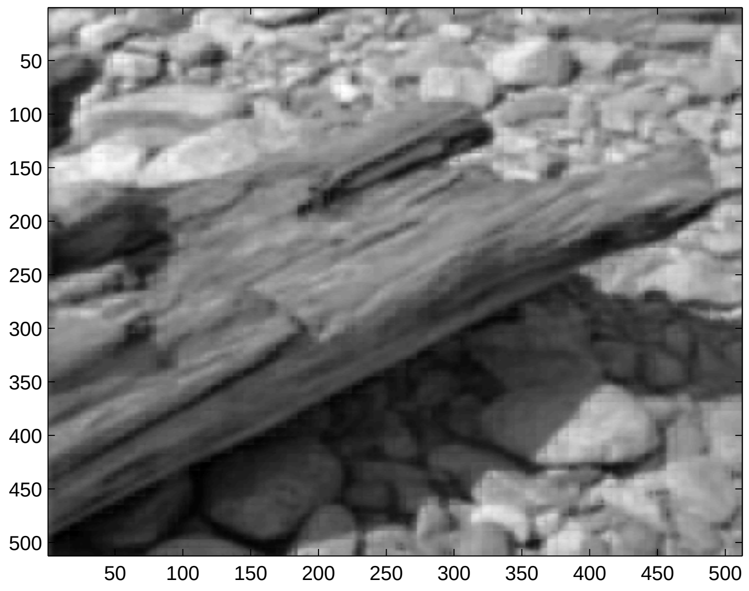


Fig. 3.5: Super-resolved averaged overlap image.

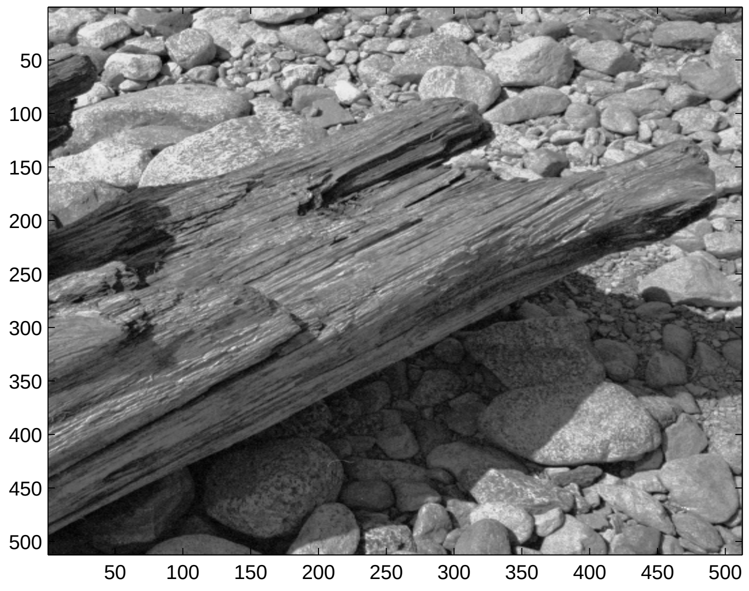


Fig. 3.6: Original image.

Chapter 4

Point Cloud Super Resolution

This chapter discusses the necessary changes to the method presented in Chapter 3 in order to super resolve a point cloud.

4.1 Range Image Super Resolution Testing

As with the two-dimensional images, the point cloud super resolution will be tested by downsampling the high resolution image. The low resolution image is derived from the high resolution image because to find the error between the super-resolved image and the original high resolution image, the two images must line up. The only way to ensure this is true is to derive the low resolution image from the high resolution image.

In Chapter 3, the low resolution image is derived from the high resolution image by blurring before the downsampling. The reason for the difference is in the sensor that is being simulated in each case. In this case, a Ladar is being simulated. A downsample is a much better approximation of a range image at a farther distance. This is because in a Ladar, the farther away the sensor is from the target, the more the laser diverges. This means that each beam front print will be larger on the target, while still only receiving one sample per beamlet. This results in fewer shots on the target.

4.2 Patch Size for Range Images

In this thesis, range images of a tank in a plain scene will be the focus. That is to say that the tank will be placed on a flat section of ground with no other objects in the scene. Because the tank is the only object of interest in the scene, the entire image does not need to be kept. This is reasonable since this thesis is focused on the idea of getting a better look at what kind of tank is in the scene. The size of the tank in the image is about 40x104 for the high resolution image. This is much smaller than the 512x512 that was worked with for the two-dimensional super resolution, but still not small enough to allow the super resolution to be performed over the entire image. An image of size 40x104 vectorized is 4160 elements. With these many elements, convex solvers would take a long time to solve. Another problem with super resolving the entire tank image is that depending on the orientation of the tank in the scene and the angle from which the image is taken, the size of the tank in the scene can vary a great deal. This size variance means that the length of the vectorized image will change. Since the dictionary must have the same number of rows as the length of the vectorized image, a dictionary would need to be generated for every Ladar shot. This is expensive computationally and wasteful of memory. These facts lead to a conclusion that the range images must be split up into patches as well.

With the size reduction mentioned above, a corresponding size reduction in the patch size is appropriate. In the two-dimensional super resolution case, the entire scene had to be super-resolved. This was a major consideration in choosing a patch size to work with. For point cloud super resolution, however, the only constraint is that the entire area of the tank and the immediately surrounding area must be super-resolved. Because of this, if a few of the bordering range values are missed by the super resolution algorithm, it does not matter, as long as these values are not a part of the tank.

The size of the high resolution patch needs to be large enough to allow the low resolution version to have a enough information about the scene for super resolution to take place. For example, if the high resolution patch was 4x4 elements, a corresponding low resolution image would contain only a single element. This would not provide for a very interesting super resolution problem, as the only thing to do would extend the single range value over the 4x4 high resolution patch. A patch size of 12x12 allows for the testing of two, three, and four time super resolution evenly while allowing some information in the low resolution image in order for the algorithm to operate.

With this change in patch size must also come a change in the number of elements that are overlapped between sections. In Chapter 3, a quarter of the total patch size was overlapped on each side. For a patch size of 12 pixels an overlap of four pixels will be used.

These overlapping values can either be averaged together or averaged with a weighting. For an overlap of five pixels, for the weighting to gradually move from one image patch to another, the values should be

$$center \begin{bmatrix} 1 & 0.6667 & 0.3333 & 0 \end{bmatrix} edge. \tag{4.1}$$

Again, in order for this weighting scheme to be valid (for each respective element to sum to one) the left side weighting values become

$$edge \begin{bmatrix} 0 & 0.3333 & 0.6667 & 1 \end{bmatrix} center. \tag{4.2}$$

These values would be extended vertically along the entire left or right side of the respective image patch to be tied into the entire image. The weighting in the vertical direction on the bottom becomes

$$\begin{array}{c} center \\ \begin{bmatrix} 1 \\ 0.6667 \\ 0.3333 \\ 0 \end{bmatrix} \\ edge \end{array}. \tag{4.3}$$

As in the previous case, the top weighting scheme becomes

$$\begin{array}{c} edge \\ \begin{bmatrix} 0 \\ 0.3333 \\ 0.6667 \\ 1 \end{bmatrix} \\ center \end{array} . \tag{4.4}$$

As before, these values are extended horizontally along the entire top or bottom of the image.

In performing these operations on every side of every patch leads to sections of each patch that need to be averaged in the horizontal and in the vertical directions. This happens in each 4x4 sized corner section. In each corner, there are values from four patches that must be averaged together. In order to properly handle these areas, first the two corners that overlap from patches that are on the same row are averaged according to the weighting scheme just described in the horizontal direction. This will result in two 4x4 sub patches, one from the top patches and one from the bottom patches. These two resulting corner patches are then averaged according to the weighting scheme in the vertical direction.

4.3 Range Image Super Resolution Using Gabor Wavelets

The process described in Chapter 3 to super resolve two-dimensional natural images can be applied to range images with little modification. The patch size was shrunk to 12x12 pixels with an overlap of four pixels on each side. This necessitates changing the b_0 parameter to 0.0833 in the generation of the Gabor wavelet dictionary. Each wavelet has a spatial extent of 12x12 pixels.

Another necessary modification is that each patch taken from the low resolution image does not have the mean subtracted out. Nor is each vectorized patch normalized to unit norm. Instead, the mean is subtracted from the entire low resolution range image. This mean is then added to the super-resolved high resolution image.

Taking into account the above modifications, the super resolution algorithm described in Chapter 3 is applied to the range images using weighted averaging. As in Chapter 3, the spgl1 solver is used.

4.4 Range Image Super Resolution Using Dictionary Training

Another technique to super resolve images would be to use a trained dictionary instead of a Gabor wavelet dictionary. This dictionary can be trained using the following technique.

Dictionary training is used to generate a set of basis functions that sparsely represent a particular kind of data. Because this thesis is focusing on super resolving range images of various tanks, an a priori assumption can be made, namely that the object in the image is a tank. This is because in many applications, it is known what the target image is in general, namely a tank of some kind and the goal is to get more information about the particular tank in the scene. Assuming that the target in the scene is a tank of some kind, it is possible to train a custom dictionary to sparsely represent a tank. As mentioned in Chapter 2 of this thesis, the Matlab software package known as Sparse Modeling Software (SPAMS) will be used for the dictionary training algorithm. This software package requires an input of example images. The algorithm generates a dictionary of atoms that is optimized to sparsely represent these example images.

In order to generate the example patches for the dictionary training, a number of range images must be generated. The Matlab software package Ladarsim will be used to generate these images. The parameters to simulate a flash Ladar of size 128x128 will be used. In order to get an image of each side of the tank, scans will be taken in 60 degree increments. In order to cover different distances from the tank, three different circular paths will be simulated, each with six scans per circle. Each circular path has a constant distance from the tank for each of the six scans. This means that there are 18 range images for a given tank. Scans of three different tanks will be used, which means that patches will be taken from total of 54 range images over three different tanks. By taking 2000 random patches from every range image for training, the total number of example patches becomes 108000. Each patch is size 12x12. By vectorizing each patch and placing each vector on a new column, a matrix of size 144x108000 is generated in order to train a dictionary. The columns of this matrix are the x_i values from (2.2). In this case, n (from the same equation) is 108000. After training different sized dictionaries, it was determined that a dictionary with 40 atoms (with 40 columns) works best. This trained dictionary is D_h, or the high resolution dictionary.

4.4.1 Application of Compressive Sensing

The purpose of the dictionary described in the previous section is to reconstruct the high resolution patches. In order to obtain the low resolution dictionary, the high resolution dictionary can be downsampled and blurred in the way described in Chapter 3 of this thesis. The coherence of the downsampled dictionary without the blur function is 0.2948. The coherence of the downsampled and blurred dictionary is 0.1600, and thus confirming the application of the blur in the downsampling process for use in CS. This blurred and downsampled dictionary is D_l, or the low resolution dictionary.

4.4.2 Testing the Restricted Isometry Property

As described in the Chapter 2 of this thesis, the Restricted Isometry Property (RIP) is a major part of the CS framework. The equation given in (2.4) must be true for all vectors that are 2*k sparse when k is given by (2.5) and $\epsilon = \sqrt{2} - 1$. Equation (2.5) gives a k value on the order of 4.168. To test how the blurred and downsampled dictionary meet the RIP, a k-sparse vector (where the entries were randomly chosen) is used to test the blurred and downsampled dictionary against the RIP, as well as a downsampled dictionary against the RIP. Out of 40 vectors, the RIP was true for only one vector for the the downsampled dictionary. For the 40 vectors tested, the RIP was true for 26 vectors for the blurred and downsampled dictionary. This further confirms the use of the blurred and downsampled dictionary in CS.

4.4.3 Total Variation Minimization

The type of image that we are trying to reconstruct has many flat sections with a few quick transitions. Think a tank sitting in the middle of a field for example. In order to eliminate much of the noise in the image while preserving these quick transitions, Total Variation (TV) minimization can be applied. TV is applied by adding a parameter to the minimization problem. The goal of TV minimization is to decrease the amount of change in a signal while still allowing the occasional change. This is a convex problem, so a convex solver can solve a TV problem. The Total Variation parameter can be written for some one-dimensional signal x by using

$$\sum_{i=1}^{n-1} |x_{i+1} - x_i|. \tag{4.5}$$

This parameter is the difference in the signal. It assigns large values to signals that are changing quickly, but less value on large changes. This tends to minimize the quick changes while allowing large changes [16].

This can be written as a matrix multiplied by a vector representing the one-dimensional signal. Do this by defining D_v as

$$D_v = \begin{bmatrix} A_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_v & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_v & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_v & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_v & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_v & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_v & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_v \end{bmatrix},$$

where

$$A_v = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

By taking off the last column of D_v, this matrix becomes the correct size to multiply by the vectorized high resolution image patch. To apply TV to an image, the variation parameter in the vertical direction can be written as

$$\|D_v D_h \gamma\|_1. \tag{4.6}$$

In this equation, the coefficients γ are being found by the solver. So the $D_h\gamma$ term is the high resolution patch. By multiplying D_v, the vertical variation is found in the high resolution patch.

In an image, TV needs to be applied in the horizontal direction as well. This can be done by defining the matrix D_{horz} as

$$D_{horz} = \begin{bmatrix} A_h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & A_h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_h & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_h & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_h & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_h & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_h & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_h & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_h & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & A_h \end{bmatrix},$$

where A_h is defined as

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

With these matrices defined, the horizontal TV parameter can be written as

$$\|D_{horz} D_h \gamma\|_1. \tag{4.7}$$

As before, the coefficients γ are being found by the solver. By multiplying D_{horz} by the high resolution patch $D_h\gamma$, the horizontal variation is found in the high resolution patch.

With these additions to the minimization, the minimization equation becomes

$$\min_{\gamma} \|D_l\gamma - y\|_2^2 + \lambda_1\|\gamma\|_1 + \lambda_2\|D_v D_h\gamma\|_1 + \lambda_3\|D_{horz} D_h\gamma\|_1. \tag{4.8}$$

By adding the vertical and horizontal variation parameters to the minimization problem, the variation will be minimized by the solver. The different λ values are to balance how important each term becomes in the minimization problem. The results for super resolving a range image using a trained dictinary are generated by setting $\lambda_1 = 0.5, \lambda_2 = 1$, and $\lambda_3 = 1$. This is solved via the convex solver CVX. CVX was developed by Boyd and Grant [17, 18]. A discussion on the solution methods used in the solver itself is beyond the scope of this thesis.

4.5 Results

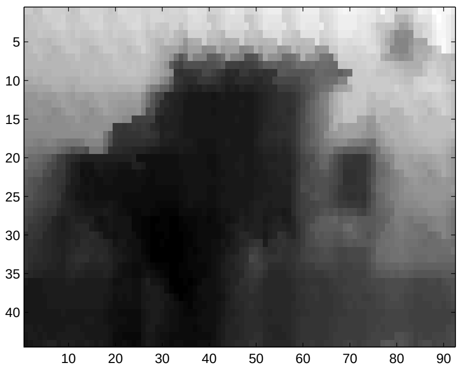
The results for super resolution of a range image using both Gabor wavelets and dictionary learning will now be presented.

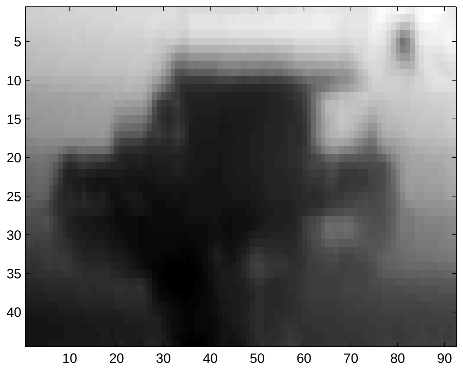
In order to form a complete output image, many patches are combined together. The results of super resolving an entire image are given in the Appendix.

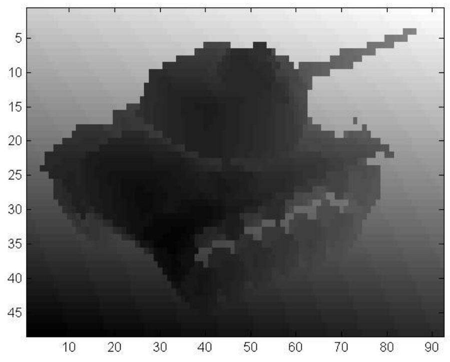
Some of the images where the error in the super-resolved image for the learned dictionary is greater than the error of the bicubic interpolation correspond to the images that were taken from the front of the tank. There are less images of the front of the tank and of the sides of the tank and so there are few patches in the dictionary to reconstruct these images. Note that only one image using the Gabor wavelet dictionary had greater error than the bicubicly interpolated range image. Examples of super-resolved images can be seen in Fig. 4.1.

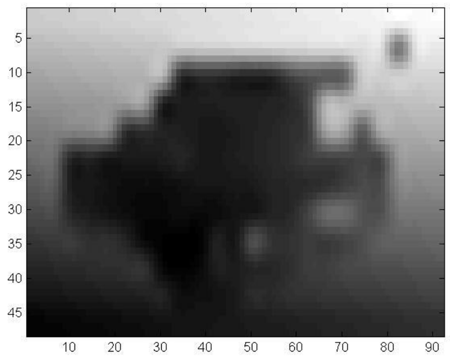
4.6 Azimuth and Elevation Upsampling

The assumption for this thesis is that the data is coming from a flash Ladar of size 128x128 instead of a flying spot Ladar. With a flying spot Ladar, a single detector senses the return from a single laser shot. The laser is then pointed in different directions by


(a) Super-resolved Image Using Learned Dictionary


(b) Super-resolved Image Using Gabor Wavelets


(c) Original Image


(d) Bicubicly Interpolated Range Image

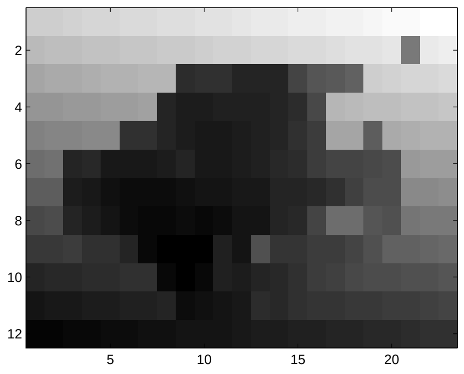

(e) Original Low Resolution Range Image

Fig. 4.1: Range image super resolution comparison.

different scanning mirrors. The flying spot Ladar potentially has jitter in the scanning mirrors can cause the range values that are found to not be on a standard sampling grid. This complicates the ability to represent the range data as a range image. With a flash Ladar, many returns are sensed at the same time. The laser shot is split into many smaller rays via an optics system. There are many different sensors that then sense the returns from these shots all at one time. One of the advantages of a flash Ladar is that many returns can be sensed at one time. Another advantage is that all of the return values are on a grid and can be easily represented as a matrix.

This grid like sampling of the data makes the process of going from the low resolution azimuth values to the high resolution azimuth values (and similarly for elevation) a much easier process. Because of the nature of the azimuth values, a linear interpolation of the azimuth and elevation values will produce results with very low error when compared to the actual high resolution azimuth and elevation values. The upsampled azimuth MSE is measured to be 4.4565e-13. The upsampled elevation error is measured to be 2.0833e-13.

4.7 Range Image to Point Cloud

Data from a Ladar system can be represented in two different ways. If the range data from the scene is on grid, the data can be easily represented as a two-dimensional image known as a range image. The other form of data that Ladar data can be represented as is in a point cloud. A point cloud is a collection of points that can be represented in a three-dimensional space. In order to convert from a range image to a point cloud, the azimuth and elevation values must be known. With a knowledge of the range, azimuth, and elevation values, the x, y, and z coordinates can be found via a simple transformation. This transformation is given in (4.9).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} cos(AZ)cos(EL) \\ sin(AZ)cos(EL) \\ sin(-EL) \end{bmatrix} \tag{4.9}$$

In this equation, R represents the range value at the particular azimuth and elevation.

This is the conversion for a single range, azimuth, and elevation point. By repeating this for every point in the range image, the x, y, and z coordinates that make up an entire point cloud can be found.

4.8 Point Cloud Results

It is difficult to calculate numerically the improvement of the point cloud itself. The reduced error in the range image shows that the range values are closer to the original high resolution image than with an image that has been upsampled via bicubic interpolation. Since the azimuth and elevation values used to convert either the interpolated image or the super-resolved image to a point cloud are the same, it can be assumed that since the range values are closer to the original the super-resolved point cloud will also be closer to the original point cloud. The super-resolved point cloud using the learned dictionary is shown by itself in Fig. 4.2. The super-resolved point cloud using Gabor wavelets is shown by itself in Fig. 4.3. By inspecting the point clouds, the point cloud derived from the interpolated image has a feature above the turret that is not in the original. This can be seen in Fig. 4.4. The original high resolution point cloud is seen in Fig. 4.5. Since the main application here is ATR, the goal would be to improve the identifiability of a point cloud of a specific type of tank. Since the range values of the super-resolved tanks are closer to the values of the original than the original, this seems likely. The point cloud super-resolved using Gabor wavelets has less distortion of the flat ground, but there is still some ringing.

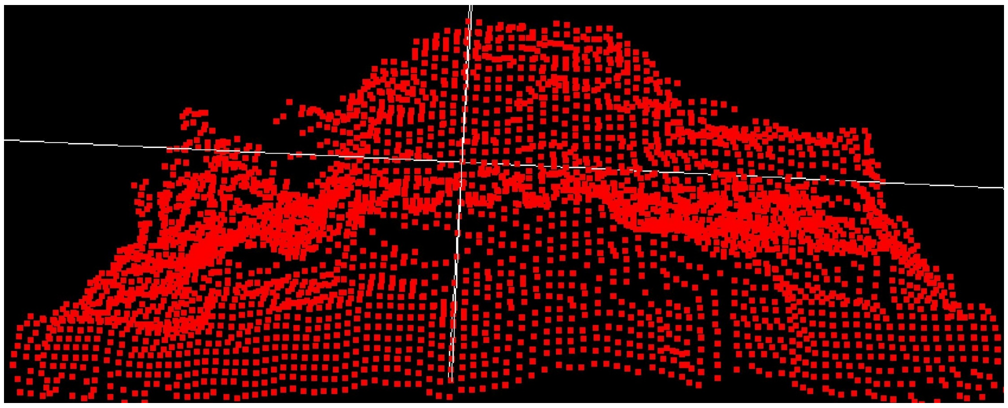


Fig. 4.2: Super-resolved point cloud using a learned dictionary.

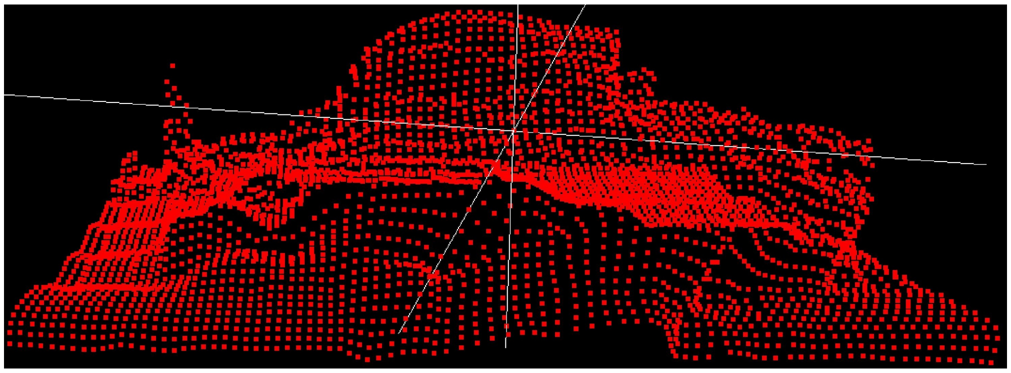


Fig. 4.3: Super-resolved point cloud using Gaborlets.

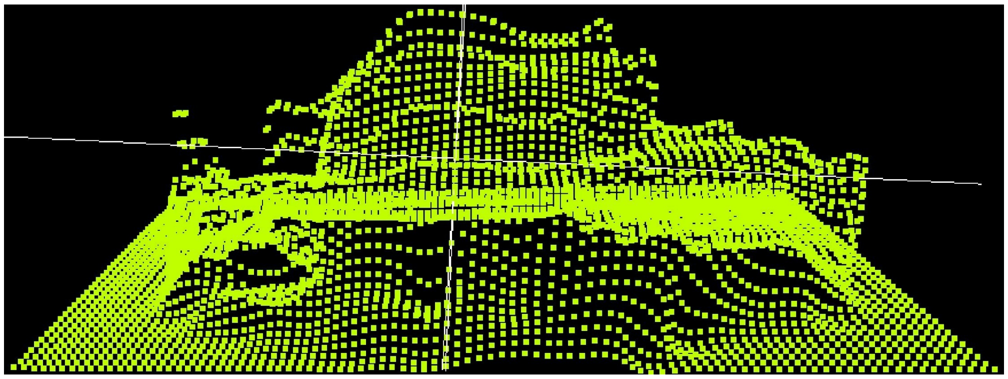


Fig. 4.4: Bicubicly interpolated point cloud.

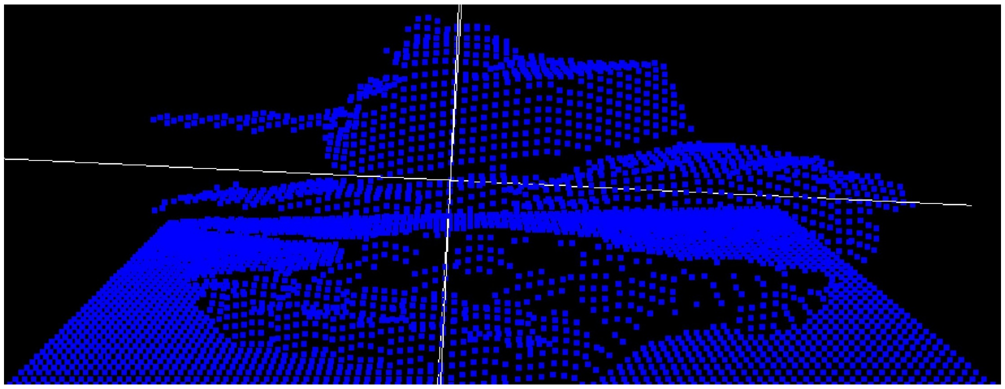


Fig. 4.5: Original high resolution point cloud.

A comparison between the bicubicly interpolated point cloud and the original point cloud is shown in Fig. 4.6. In this figure an artifact can be seen above the turrent. It looks a bit the brim of a hat sitting on top of the turret. In Fig. 4.7, the super-resolved point cloud using dictionary learning is shown from the same point of view. The artifact that is expressed in the bicubicly interpolated point cloud is gone. Figure 4.8 shows the a point cloud super-resolved using Gabor wavelets. This also eliminates the artifact previously spoken of. In Fig. 4.9 is another view of the bicubicly interpolated point cloud. The next figures, Fig. 4.10 and Fig. 4.11, give the same view for each respective point cloud. A third view of the interpolated point cloud is shown in Fig. 4.12. The same view for the super-resolved points clouds is given in Fig. 4.13 and Fig. 4.14, respectively.

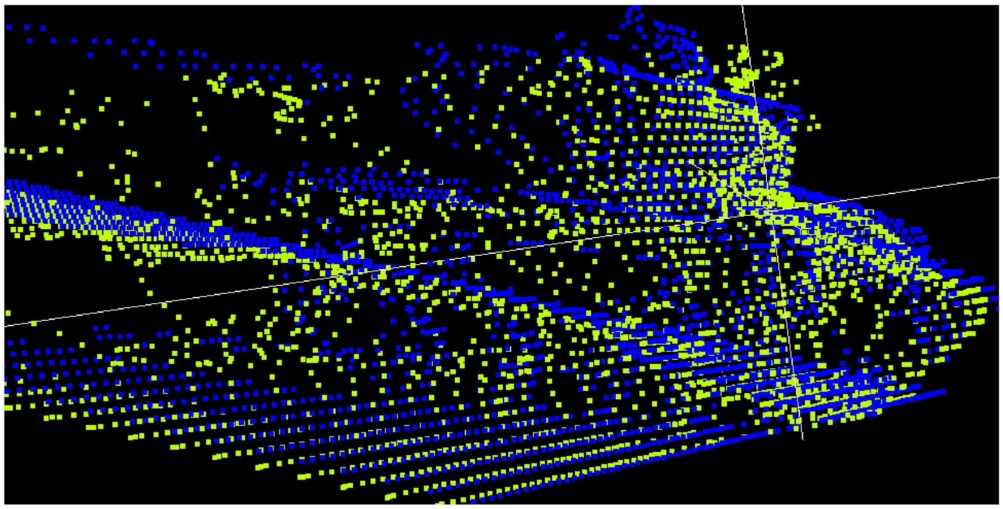


Fig. 4.6: Bicubicly interpolated point cloud comparison with the original point cloud. View 1 (Original: blue, Bicubic: green).

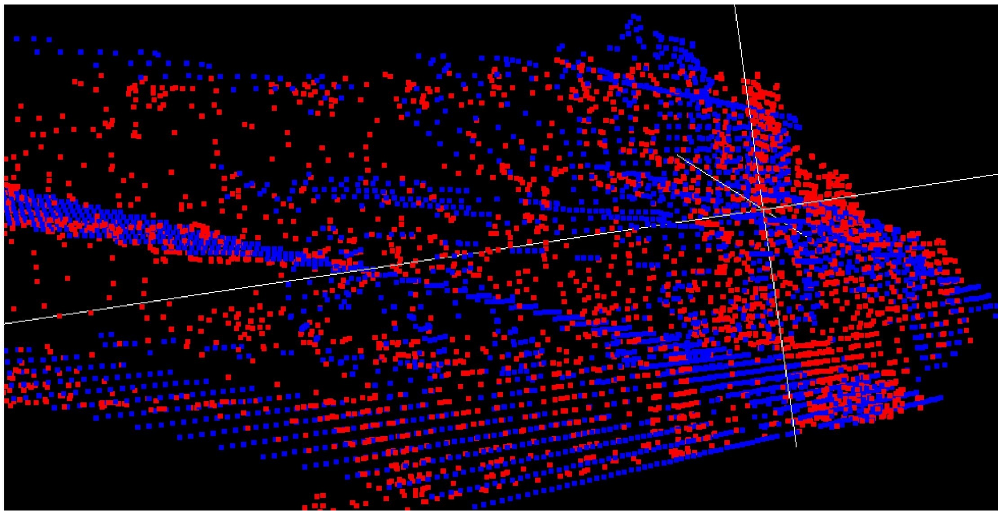


Fig. 4.7: Super-resolved point cloud using learned dictionary comparison with the original point cloud. View 1 (Original: blue, super-resolved: red).

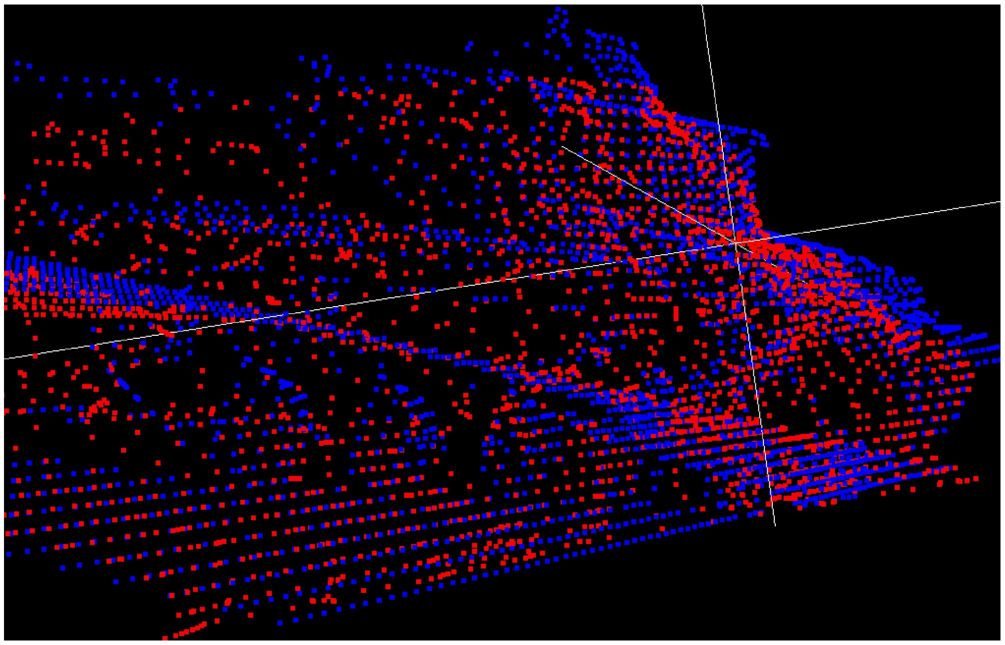


Fig. 4.8: Super-resolved point cloud using Gaborlets comparison with the original point cloud. View 1 (Original: blue, super-resolved: red).

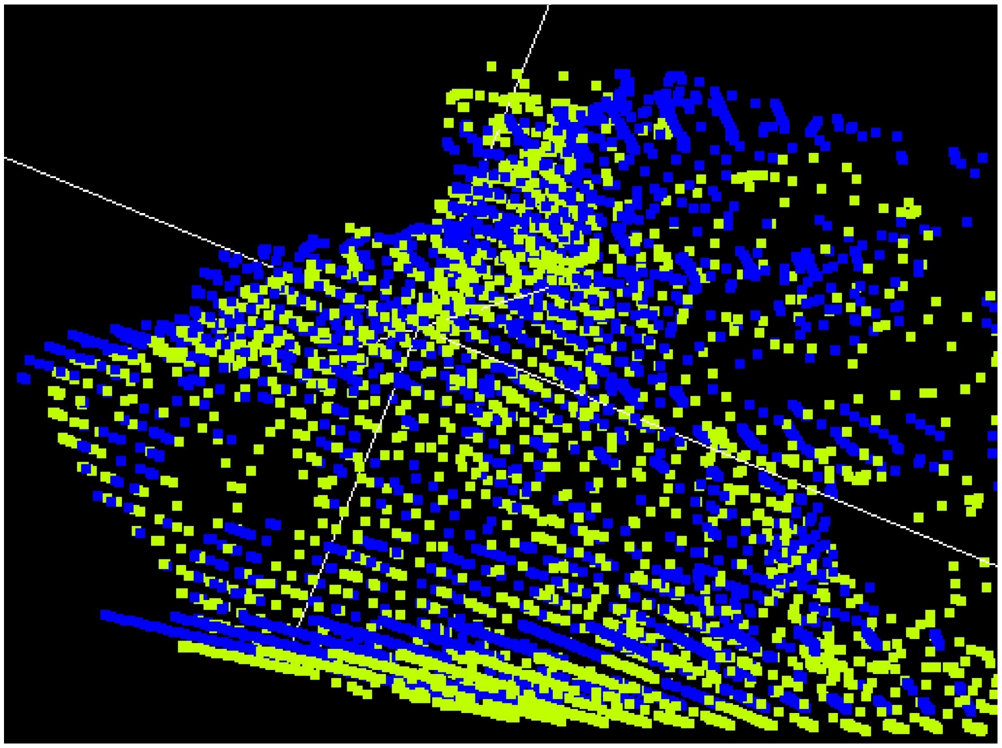


Fig. 4.9: Bicubicly interpolated point cloud comparison with the original point cloud. View 2 (Original: blue, Bicubic: green).

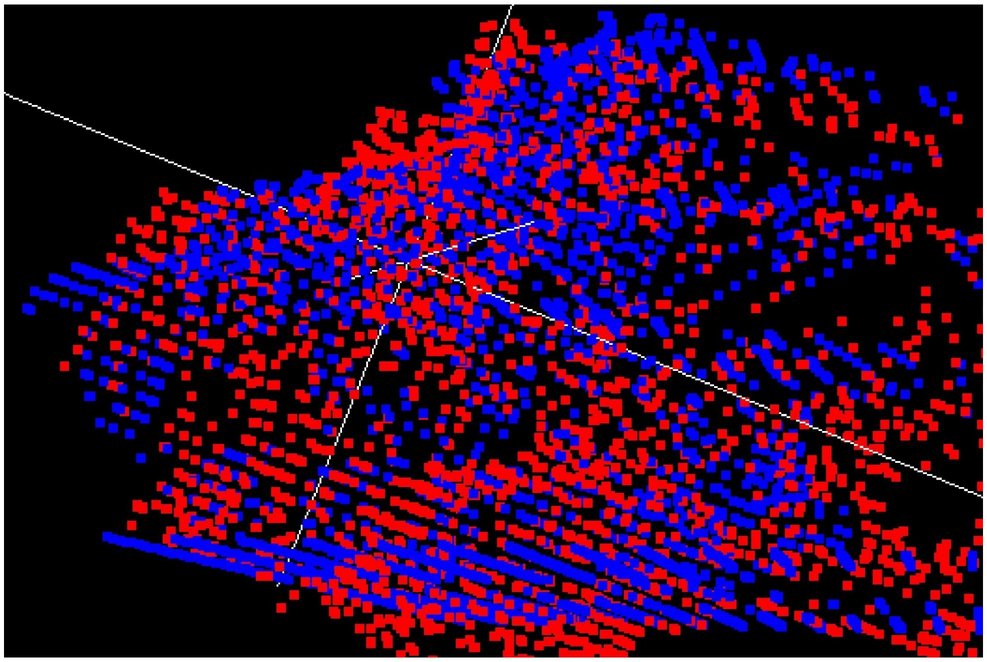


Fig. 4.10: Super-resolved point cloud using learned dictionary comparison with the original point cloud. View 2 (Original: blue, super-resolved: red).

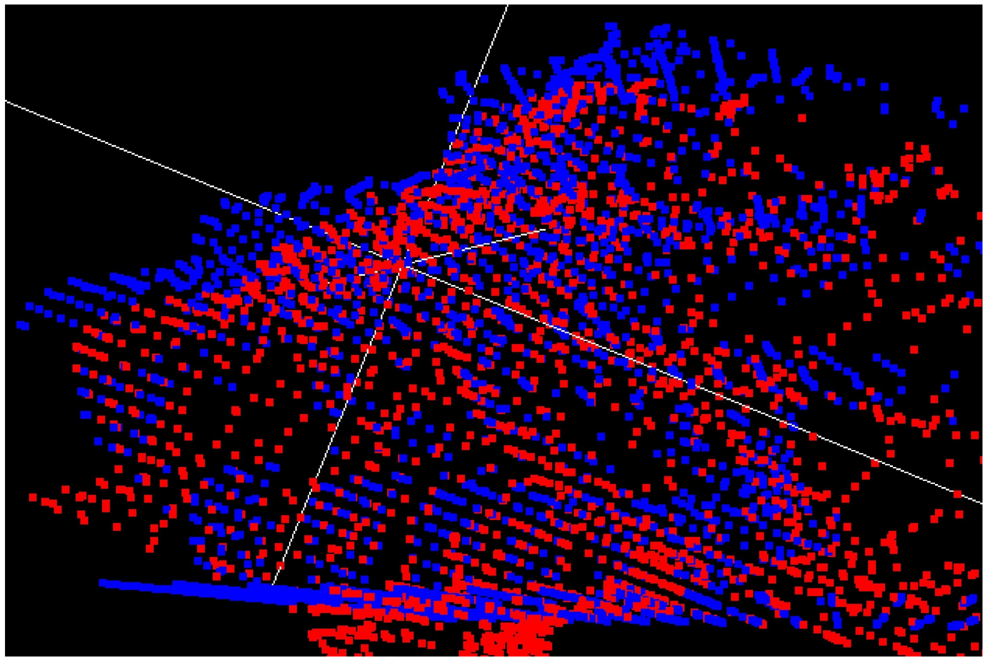


Fig. 4.11: Super-resolved point cloud using Gaborlets comparison with the original point cloud. View 2 (Original: blue, super-resolved: red).

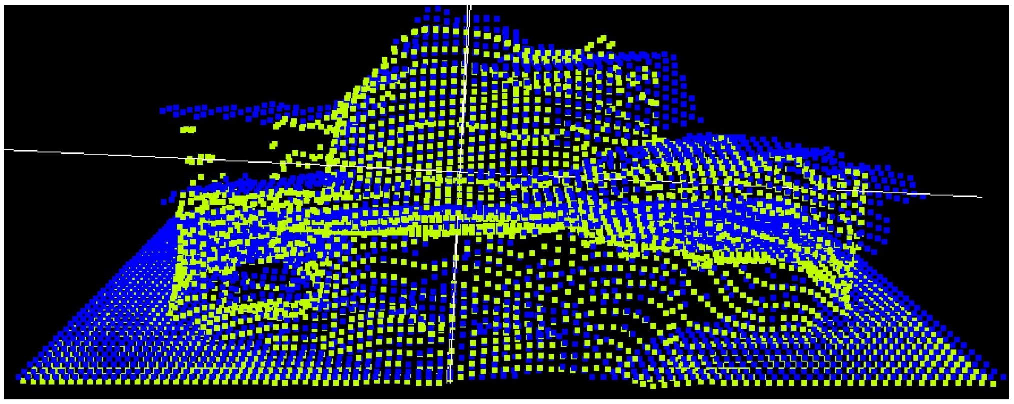


Fig. 4.12: Bicubicly interpolated point cloud comparison with the original point cloud. View 3 (Original: blue, Bicubic: green).

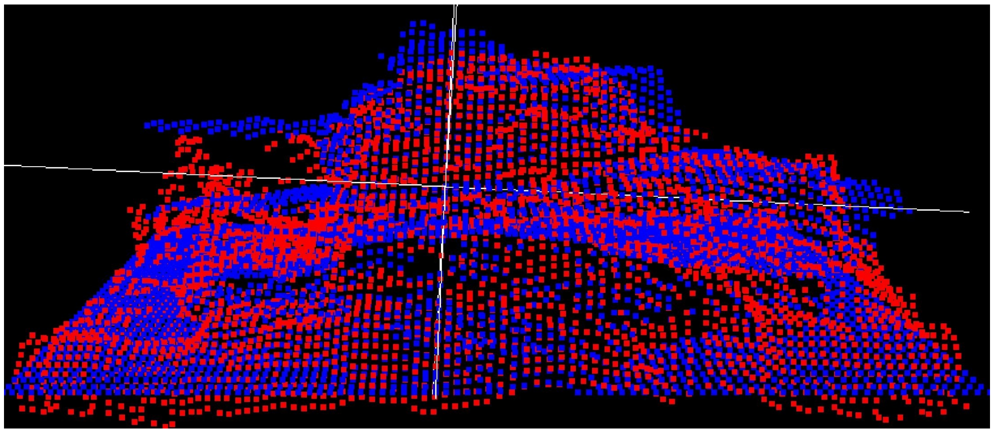


Fig. 4.13: Super-resolved point cloud using learned dictionary comparison with the original point cloud. View 3 (Original: blue, super-resolved: red).

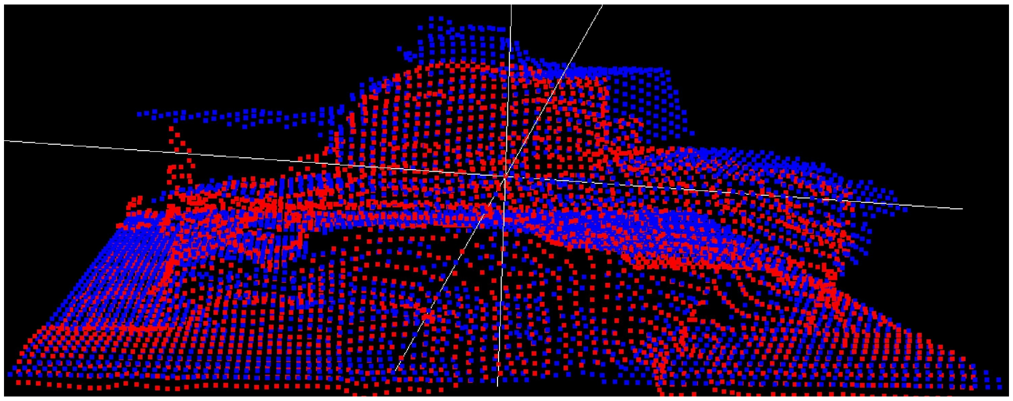


Fig. 4.14: Super-resolved point cloud using Gaborlets comparison with the original point cloud. View 3 (Original: blue, super-resolved: red).

Chapter 5
Conclusion

In this thesis it has been shown that it is possible use super resolution techniques on data in a three-dimensional space. This was applied to simulated flash Ladar data of tank images. Tank images were used in order to be applied to ATR. This technique, however, is not limited to ATR specifically. Any application where knowledge of a scene in three dimensions using a Ladar as the sensor may have a potential benefit of this technique.

5.1 Summary

In order to perform the super resolution on a point cloud, the Compressive Sensing framework was used. A flash Ladar system was simulated in Ladarsim. A background of these ideas, as well as an explanation of Gabor wavelets and dictionary learning were given in Chapter 2 of this thesis.

Chapter 3 gave an overview of an algorithm to super resolve two-dimensional images. This algorithm solved many smaller optimization problems on smaller patches out of the original image. Each patch is super resolved and then the resulting high resolution patches are stitched back together by averaging the overlapping pixels. There are two different methods presented and compared for stitching the patches back together. Gabor wavelets are used as the basis functions for the super resolution of the two-dimensional images. This method of super resolution makes it easy and computationally efficient to change the super resolution factor.

Chapter 4 extended the methodology presented in Chapter 3 to point cloud data. Two differing dictionaries are used in the super resolution of point clouds. The first step in super resolving a point cloud is to super resolve a range image. The azimuth and elevation values corresponding to the low resolution image are interpolated in order to get the high resolution values. With the range, azimuth, and elevation information, a simple transformation can be performed on each point to get the corresponding x, y, and z information. This is necessary so that the point cloud can be shown in a three-dimensional viewer. Both the Gabor wavelets and the learned dictionary decrease the MSE of the range image when compared to bicubic interpolation. The Gabor wavelets produced a smoother point cloud than the learned dictionary that would most likely perform better in ATR based on this smoothness. The Gabor wavelet point clouds, however, do exhibit some ringing in the flat sections of the scene.

5.2 Further Work

It may be possible to tune the parameters of the optimization problems to discriminate the object features best, instead of tuning the parameters to decrease the error. Also, custom tuning of the learned dictionary may yield improved results. If some atoms were added that better represented the flat areas of the scene, the image may be smoother. Also, customization by way of choosing the most appropriate example patches for dictionary training may also yield improved results. It would be interesting to explore the relationship in the error of the range images and how well the object is recognized by an ATR algorithm.

Further improvement in the super resolution of a point cloud may potentially come by performing an iterative process on the point cloud. The steps of bicubicly interpolating the range image, then splitting up the point cloud into the key features of the scene (an example would be for a tank would be to break up the pieces of the tank like the turrent, the tank body, and the barrel), and performing a smoothing and/or super resolution technique on each piece separately. Then piece the scene back together.

A potential way of performing super resolution on the point cloud would be in the full three-dimensional space. Put in a different way, this would be to explore the idea of performing super resolution in the x, y, and z space of the point cloud instead of the range, azimuth, and elevation space. That is to say, in the Cartesian coordinate space instead of the spherical coordinate space.

Another possibility for further work would be continued exploration in the area of applying non-convex super resolution techniques to range images in order to improve a point cloud. It is also possible that non-convex methods of framing the problem could improve the results of the super resolution. A non-convex problem would require a completely different set of solution methods than those explored in this thesis.

It also may be possible to extend the work in this thesis to discriminate between the different tanks themselves. This could potentially be accomplished through different training algorithms of the dictionary or changes to the optimization problem itself. It may be possible to train a separate dictionary for each type of vehicle to be discriminated. If tanks were to be discriminated against, a separate dictionary may be trained for each type of tank. If tanks and trucks are to be discriminated then a separate tank and truck dictionary may be generated. Once the super resolution is performed, the dictionary with the most basis functions used would be the type of vehicle estimated.

Another possible extension would be to train a dictionary from a vehicle in an open field. Then explore taking Ladar shots of the same vehicle under trees or other form of obstacle. The obstacle would reduce the amount of information known to the sensor about the vehicle itself. With the custom trained dictionary, it may be possible to reconstruct what the original vehicle is.

References

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: A technical overview," *Signal Processing Magazine, IEEE*, vol. 20, pp. 21–36, May 2003.

[2] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *Image Processing, IEEE Transactions*, vol. 19, pp. 2861–2873, Nov. 2010.

[3] P. Sen and S. Darabi, "Compressive image super-resolution," in *Signals, Systems and Computers, 2009 Conference Record of the Forty-Third Asilomar Conference*, pp. 1235–1242, Nov. 2009.

[4] A. Habibi, "Introduction to wavelets," in *Military Communications Conference Conference Record, IEEE*, vol. 2, pp. 879–885 vol. 2, Nov. 1995.

[5] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "Lidarboost: Depth superresolution for ToF 3D shape scanning," in *Computer Vision and Pattern Recognition, IEEE Conference*, pp. 343–350, June 2009.

[6] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," in *Computer Vision and Pattern Recognition, IEEE Conference*, pp. 1–8, June 2007.

[7] T. S. Lee, "Image representation using 2D Gabor wavelets," *Pattern Analysis and Machine Intelligence, IEEE Transactions*, vol. 18, pp. 959–971, Oct. 1996.

[8] D. Gabor, "Theory of communication. Part 1: The analysis of information," *Electrical Engineers - Part III: Radio and Communication Engineering, Journal of the Institution*, vol. 93, pp. 429–441, Nov. 1946.

[9] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. 2, pp. 1160–1169, 1985.

[10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *ArXiv e-prints*, Aug. 2009.

[11] E. Candes and M. Wakin, "An introduction to compressive sampling," *Signal Processing Magazine, IEEE*, vol. 25, pp. 21–30, Mar. 2008.

[12] M. Gharavi-Alkhansari and T. Huang, "A fast orthogonal matching pursuit algorithm," in *Acoustics, Speech and Signal Processing. Proceedings of the 1998 IEEE International Conference*, vol. 3, pp. 1389–1392, May 1998.

[13] B. Olshausen, "Sparse coding simulation software." `http://redwood.berkeley.edu/bruno/sparsenet/`.

[14] E. van den Berg and M. P. Friedlander, “Probing the pareto frontier for basis pursuit solutions,” *SIAM Journal on Scientific Computing*, vol. 31, no. 2, pp. 890–912, 2008.

[15] E. van den Berg and M. P. Friedlander, “SPGL1: A solver for large-scale sparse reconstruction,” June 2007. http://www.cs.ubc.ca/labs/scl/spgl1.

[16] S. Boyd and L. Vandenberghe, *Convex Optimization.* New York, Cambridge University Press, 2004.

[17] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 1.21.” `../../cvx`, Apr. 2011.

[18] M. Grant and S. Boyd, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control* (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, pp. 95–110, Heidelberg, Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

Appendix

Full Range Image Results

The results of super resolving all 54 images from which example images are taken are listed below for the learned dictionary, Gabor wavelets (Gaborlets), and bicubic interpolation..

Table .1: Range image SR results.

Image	MSE Learned Dictionary	MSE Gaborlets	Bicubic MSE
Image 1	0.4115	0.3014	0.4731
Image 2	0.5459	0.4381	0.6357
Image 3	0.4144	0.3355	0.4653
Image 4	0.6282	0.4681	0.4193
Image 5	0.4637	0.3824	0.4957
Image 6	0.4361	0.2793	0.4985
Image 7	0.6338	0.5408	0.8710
Image 8	0.8234	0.6948	0.9604
Image 9	0.6682	0.5474	0.7857
Image 10	1.1853	0.8004	0.8670
Image 11	0.6840	0.6431	0.7972
Image 12	0.7052	0.6771	0.9019
Image 13	1.5183	1.2643	1.8256
Image 14	1.8178	1.5072	1.6187
Image 15	1.4243	1.2549	1.7881
Image 16	2.2057	1.6730	2.0896
Image 17	1.1499	1.0435	1.3680
Image 18	1.5154	1.4650	2.2588
Image 19	0.4134	0.3423	0.5297
Image 20	0.3482	0.2522	0.3588
Image 21	0.3612	0.2361	0.3397
Image 22	0.2980	0.3381	0.4978
Image 23	0.3537	0.2854	0.3597
Image 24	0.3749	0.3124	0.3718
Image 25	0.7724	0.5771	0.8204
Image 26	0.5395	0.4667	0.6102
Image 27	0.6369	0.5114	0.5434
Image 28	0.6429	0.4443	0.6037
Image 29	0.4741	0.4747	0.6280
Image 30	0.6008	0.6451	0.9251
Image 31	1.4262	1.0518	1.3351
Image 32	1.0216	0.8380	1.1152
Image 33	1.3135	1.0628	1.6914
Image 34	1.3302	1.2369	1.8816
Image 35	1.2007	1.0511	1.4599
Image 36	0.9961	1.0142	1.4864
Image 37	0.5015	0.3381	0.5214
Image 38	0.4449	0.4196	0.7905
Image 39	0.3873	0.2958	0.3264
Image 40	0.5518	0.5324	0.6267
Image 41	0.3479	0.2319	0.3245
Image 42	0.4675	0.2966	0.4223
Image 43	0.9178	0.6639	1.0529
Image 44	0.7359	0.5117	0.7598
Image 45	0.7009	0.5610	0.6762
Image 46	0.7480	0.7908	0.9968
Image 47	0.5573	0.4168	0.6345
Image 48	0.6564	0.5728	0.8848
Image 49	1.2786	1.1086	1.7993
Image 50	1.5656	1.4944	2.1564
Image 51	1.3722	1.0670	1.4083
Image 52	1.2029	1.2757	1.8184
Image 53	0.9034	0.7513	0.9743
Image 54	1.4184	1.1096	1.5543