Master’s Final Project Report

**Single Image Super Resolution in Medical Imaging**

****

A master project report submitted to the

Faculty of University of Bridgeport

in partial fulfillment of the

requirements for the degree of

**Master of Science**

in the Department of Computer Engineering

by

Muthuraman Palaniappan

(Student ID: 1005760)

Master’s Project Advisor: Dr. Miad Faezipour

University of Bridgeport, Bridgeport, CT 06604

**ACKNOWLEDGMENTS**

I would like to take this opportunity to thank the people who supported me throughout my Master’s program and towards the completion of my Master’s project report.

I would first like to thank my advisor Dr. Miad Faezipour. The door to Prof. Faezipour office was always open whenever I ran into a trouble spot or when I had a question about my project or report writing. She has helped me learn a lot in the field of Computer Engineering. With the knowledge she imparted and with her valuable guidance and motivation, which helped me complete this project on time.

I would also like to thank the rest of the faculty of the Computer Engineering department for guiding me through the various courses I took during my program.

Finally, I would like to express my very profound gratitude to my parents and my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of my master’s project. This accomplishment would not have been possible without them.

**ABSTRACT**

High-resolution images are required in medical diagnosis. In Bio-Medical images even small possible details can help doctors provide more accurate diagnosis. However, due to the expensive and incessantly upgrading medical imaging systems, imaging environment it is difficult to obtain an image at desired high resolution. Noise which is inherent in Bio-medical images might also reduce adversely the contrast and the visibility of detail that could contain the vital information. One of the solution for this problem is Super Resolution. There are various Super Resolution techniques that exist for this problem. In this project, focus is on enhancing the quality and Peak Signal to Noise Ratio(PSNR) of the enhanced images given a single low-resolution image, which is usually the case. For obtaining this we use an image interpolation method based on the amalgamation of bicubic interpolation and 2D interpolation filter. First bicubic interpolation algorithm is applied to the low-resolution image to obtain the high-resolution image. Then 2D interpolation filter algorithm is used to produce a Super Resolution image. The Super Resolution image obtained from this method is then compared with other interpolation method such as ICBI and INEDI.

**TABLE OF CONTENTS**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | | | | | **PAGE NO.** | |
|  | **ABSTRACT** | | | | | **iii** | |
|  | **LIST OF TABLES** | | | | | **vi** | |
|  | **LIST OF FIGURES** | | | | | **vii** | |
|  | **LIST OF ABBREVIATIONS** | | | | | **viii** | |
| **1** | **INTRODUCTION** | | | | | 1 | |
|  | | 1.1 SUPER RESOLUTION | | | | 1 | |
|  | | 1.2 NEED OF SUPER-RESOLUTION IN MEDICAL IMAGING | | | | 1 | |
|  | | 1.3 TYPES OF SUPER-RESOLUTION | | | | 2 | |
|  | | 1.4 IMAGE INTERPOLATION | | | | 2 | |
|  | | 1.5 TYPES OF INTERPOLATION | | | | 2 | |
|  | | 1.6 LINEAR INTERPOLATION | | | | 3 | |
|  | | 1.7 OBJECTIVE | | | | 4 | |
|  | | 1.8 ORGANIZATION OF THESIS | | | | 4 | |
| **2** | **LITERATURE SURVEY** | | | | | 5 | |
|  | | 2.1 EXSISTING METHODS REVIEW | | | | 5 | |
|  | | 2.2 INTERPOLATION APPROACH REVIEW | | | | 6 | |
|  | | 2.3 PROPOSED METHOD | | | | 7 | |
| **3** | **METHODOLOGY** | | | | | 9 | |
|  | | 3.1 INTERPOLATION FILTERS | | | | 9 | |
|  | | 3.2 SCHEME | | | | 10 | |
|  | | | | 3.2.1 Bicubic Interpolation | 10 | |
|  | | | | 3.2.2 Column Interpolation | 15 | |
|  | | | | 3.2.3 Row Interpolation | 17 | |
| **4** | **IMPLEMENTATION** | | | | 20 | |
|  | | | 4.1 RESIZING | | 20 | |
|  | | | 4.2 BICUBIC | | 21 | |
|  | | | 4.3 FILTER INTERPOLATION | | 22 | |
|  | | | 4.4 PSNR | | 23 | |
| **5** | **EXPERIMENTAL RESULTS** | | | | 25 | |
|  | | | 5.1 DISCUSSIONS | | 28 | |
| **6** | **CONCLUSION** | | | | 30 | |
|  | **APPENDIX- Matlab Code** | | | | 31 | |
|  | **REFERENCE** | | | | 39 | |

|  |  |  |
| --- | --- | --- |
| **LIST OF TABLES** | | |
| **TABLE NO.** | **TITLE** | **PAGE NO** |
| 5.1 | PSNR(dB) OBTAINED ON 4X ENLARGEMENT DIGITAL X-RAY IMAGES WITH THE PROPOSED, INEDI, ICBI METHOD | 28 |
| 5.2 | PSNR(dB) OBTAINED ON 8X ENLARGEMENT DIGITAL X-RAY IMAGES WITH THE PROPOSED, INEDI, ICBI METHOD | 29 |

|  |  |  |
| --- | --- | --- |
| **LIST OF FIGURES** | | |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| 3.1 | An illustration of Bicubic interpolated image obtained from LR image | 15 |
| 3.2 | An illustration of Column Interpolated image obtained from Bicubic Interpolated image | 17 |
| 3.3 | An illustration of row interpolated image obtained from the column interpolated image**.** | 18 |
| 3.4 | An illustration of Super Resolution image obtained from LR image by proposed method. | 19 |
| 4.1 | Image of skull after passing HR image to Resizing function. | 20 |
| 4.2 | Image of skull after bicubic interpolation of LR image. | 21 |
| 4.3 | Super resolution image of skull obtained from bicubic interpolated image. | 23 |
| 5.1 | Digital X-ray images from GE health care (a) skull, (b) hand, (c) foot, (d) chest, (e) aircon, (f) t-Spline, (g) pelvis. | 25 |
| 5.2 | An illustration of 4X enlargement of digital X-ray RGB skull image of size 200x200x3. (a) iNEDI, (b) ICBI, (c) Proposed Scheme | 26 |
| 5.3 | An illustration of 4X enlargement of digital X-ray RGB hand image of size 200x200x3. (a) iNEDI, (b) ICBI, (c) Proposed Scheme. | 27 |
| 5.4 | Plot of PSNR vs 4X enlarged Images for seven images taken from GE healthcare using ICBI, iNEDI, Proposed Scheme. | 27 |

**LIST OF ABBREVIATIONS**

SR - Super Resolution

HR - High Resolution

LR - Low Resolution

MATLAB - Matrix Laboratory

CT - Computerized Tomography

MRI - Magnetic Resonance Imaging

PSNR - Peak Signal-to-Noise Ratio

**CHAPTER 1**

**INTRODUCTION**

* 1. **Super Resolution**

High Resolution (HR) images are often necessary for variety of applications, such as remote sensing, object detection, feature extraction, security, biomedical imaging, video surveillance, astronomy, etc... Acquiring an HR image from the sensing device is very difficult task owing to hardware limitation and cost. Apart from this there are various other factors that are preventing us from obtaining the HR image such as improper lighting, moving objects, camera focus and distance. In order to obtain the desired HR image there are various Super Resolution (SR) techniques. SR is the process of generating HR image from one or more Low Resolution (LR) images [1].

**1.2 Need of Super-Resolution in Medical Imaging**

In medical imaging, images are obtained for medical purpose, for providing information about the physiologic, the anatomy and metabolic activities of the volume below the skin. The medical images are also used by doctors in diagnosis and to detect the shape, location, type of fracture. The arrival of medical imaging techniques such as Computerized Tomography (CT), Magnetic Resonance Imaging (MRI), Digital X-rays has modernized medical imaging. Despite of the advancement in the medical imaging technique, it is always not possible to obtain the high-quality images due to the factors such as imaging environment, the limitation of physical imaging system, etc.… For obtaining the high-quality images doctors continually seek ways to obtain it for diagnosis, by relying on expensive and incessantly upgrading machines. Thus, it is important to discover cheap, accurate, effective and efficient way to obtain HR images from the existing medical imaging machines.

* 1. **Types of Super-Resolution**

Super-Resolution was first introduced in 1984 by T. Huang and R.Tasi [2]. Then the term Super-Resolution was first used in the year 1990 [3]. After that there are many SR methods have been proposed until now. They can be broadly categorized into two main groups they are multi-image SR and single-image SR. The multi-image super resolution received a lot of attention because the certain prior knowledge about the images can help to obtain the enhanced image. But this method doesn’t work properly when we have only one LR image. The single-image SR have considerable attention in recent years, since it has emerged as a solution when only one LR image is available. In this project we deal with single-image SR since it is the case in most of the image SR. In this project we use the interpolation technique for obtaining the SR image from the single LR image.

**1.4 Image Interpolation**

Image Interpolation is the process by which the SR image is obtained from the LR image by finding the unknown pixel values by using the known pixel values [1]. Many research have been done on interpolation and they have achieved improved image quality. There is various different type of interpolation technique.

**1.5 Types of Interpolation**

Generally, Interpolation technique can be classified into two types they are Linear interpolation and Nonlinear interpolation. Linear interpolation technique is preferred in various applications because of easy implementation and fast computation. Where us Nonlinear interpolation are more complicated and it takes time for computation. Since in the medical application we are in need of computationally fast method we are going to use Linear interpolation method in this project.

**1.6 Linear interpolation**

Linear interpolation includes interpolation such as nearest neighbor, bilinear, bicubic, spline, sinc, lanczos, etc.… Depending on their complexity, these use anywhere from 0 to 256(or more) adjacent pixels when interpolating. The more adjacent the pixels they include, more accurate the image can become. The problem with linear interpolation is they distort and blur the edges.

Nearest neighbor technique is the simplest method for enhancing the image and requires least processing time of all interpolation algorithm. In this type of interpolation, the unknown pixel takes the same intensity value of its neighbor pixel. Bilinear interpolation the value of the unknown pixel is based on the weighted average of the 4 pixels in the nearest 2x2 neighborhood of the pixel in the original image. Bilinear interpolation has an antialiasing effect and therefore produces relatively smooth edges with hardly any jaggies. Bicubic interpolation goes a step beyond bilinear interpolation by considering the closet 4x4 neighborhood of known pixel i.e. total of 16 pixels. Since there are varying distances from the unknown pixel, closer pixels are given a higher weighting in the calculation. Bicubic interpolation produces noticeably sharper images than the previous methods and it is the ideal combination of processing time and output quality.

The higher order interpolation such as spline, sinc, lanczos, etc.… takes more surrounding pixels into consideration so that they are much more computationally intensive. These are extremely useful when the image requires multiple rotations or distortions in separate steps. However, for single-step enlargements or rotations, these higher-order algorithms provide diminishing visual improvement as processing time is increased. Since we are in need of single-step enlargement in medical imaging we will be using bicubic interpolation in our project.

**1.7 Objective**

The aim of the project is to design an interpolation method which has low computation time and to overcome the drawbacks of linear interpolation such as reduce the blur in the edges and as well as efficient than the existing interpolation methods for medical images. Then compare the obtained results with the other efficient algorithm to show the efficiency of our algorithm over another existing algorithm.

**1.8 Organization of Thesis**

The thesis is organized into five chapters including the introductory chapter.

The description about each chapter is as follows.

Chapter 1 deals with introduction, objective, Organization of chapters.

Chapter 2 deals with existing method review, interpolation approach review and proposed method.

Chapter 3 deals with methodology, interpolation filter, bicubic interpolation, column interpolation and row interpolation

Chapter 4 deals with implementation, resizing, bicubic, filter interpolation and PSNR.

Chapter 5 deals about experimental results and discussions.

Chapter 6 presents the conclusion of project.

**Chapter 2**

**LITERATURE SURVEY**

**2.1 Existing methods review**

Kian Kee Teoh and Haidi Ibrahim [4] had investigated nearest neighbor interpolation. Bilinear interpolation, bicubic interpolation, smoothing filter and nonlinear unsharp masking schemes to enhance the satellite images. However, all the schemes do not preserve the edge information significantly.

Eguchi et al. method [5] distribution of degradation information and original information as a distribution of probability density function and estimates the transfer function of the transfer system as the probability density function of the conditional probability. The iterative calculation is done based upon the Bayes theorem for the probability density function, estimation of the maximum likely distribution of the original information and estimation of the distribution of the transfer function are alternately obtained on the distribution of deterioration information, and finally the original information. When nearly change is same as that of the original information end of the computation is judged, but when the initial setting of the transfer function is inappropriate, divergence may occur in the course of iterative computation. If such case occurs a considerable time is required for the calculation, calculation also becomes large. Also, when the number of pixels of the degraded image is small, the restored result tends to be an image in which the edge is thickened, jaggies appear and the contrast is emphasized.

Maurya et al. proposed [6] a technique determines edges by comparing the intensity variations between center pixel and its neighboring pixel of window size 5x5. Then the blank spaces are filled with designed thresholds. They used a sliding window of 5x5 instead of 3x3. This method produced a promising result on medical images and satellite images. But this method sometimes over sharpens the edges.

Koji Kita and Michifumi Yoshioka [7] used a bilateral filter for obtaining the image super resolution. A bilateral filter which is a general noise removal method is an edge preserving type noise removal filter and is known to have an effect of emphasizing an edge together with noise removal effect. By applying this filter, they conducted experiments on images and calculates PSNR ratio for comparison. From the result it is found that edges become steeply larger than conventional methods. Since these method uses smoothing filter, the resulting images has smoothing effect.

**2.2 Interpolation approach review**

Many researches have been done on improving the image quality by performing edge directed interpolation by modifying the interpolation scheme. Edge-directed interpolation (EDI) methods [8] utilizes the local statistical and geometric properties to interpolate the unknown pixel value. In the New Edge-directed interpolation (NEDI) method [9], the unknown pixel values to be calculated are assuming the local image covariance in a large window at different scales. The interpolated pixel is calculated as the weighted average of the neighboring four pixels in vertical and horizontal directions. This method has higher computational complexity. The improved New Edge-directed interpolation (iNEDI) [10] method modifies the NEDI method by varying the dimension of the training window with respect to the edge size and this method uses a circular window instead of the rectangular window used in the NEDI method to obtain better PSNR performance.

Eri Hosogai and Yuichi Tanaka [11] proposed an edge-directed iterative image interpolation method based on the second-order derivatives along multiple directions. They assumed that second-order derivatives of image intensities are continuous along the interpolation directions. First the pixel values are initialized based on a grid filling and iteratively modify the interpolated pixel by minimizing an objective function depending on the second-order directional derivatives. This method produces better result at the edges and this method works faster than existing EDI method. But it still has more computational time compared to faster interpolation method.

The Iterative Curvature Based Interpolation (ICBI) method [12] considers the effects of the curvature continuity, curvature enhancement and isophote contour. This method produces the images which are visually improved by correctly balancing these three effects. In spite of the increase in the performance, the increased computational time prohibits the algorithm from being used in daily applications. While the conventional linear interpolation scheme such as bilinear and bicubic interpolation fails to capture the rapid evolving information around the edges and so they generate the interpolated images with artifacts and blurred edges. Linear interpolation scheme is generally preferred because of the computational simplicity. While the non-linear interpolation technique is preferred when there is a need for performance. In most of the application linear interpolation techniques are used because of the short computational time.

**2.3 Proposed Method**

In this project we propose a method which is a type of linear interpolation method. For this we use bicubic interpolation and two-dimensional (2D) filtering with a unique interpolation method to produce accurate and perceptually pleasant super resolution image. This method tends to be faster than the non-linear interpolation methods and also produce more accurate results than those methods.

The proposed scheme was tested on the digital X-Ray images that were obtained from the GE Healthcare. The results obtained from our proposed scheme shows that it has improved PSNR performance on digital X-ray images.

**Chapter 3**

**Methodology**

The drawbacks of the Linear interpolation technique such as bicubic, bilinear, etc.… has poor performance especially on the edges. This is the main reason for not using the linear interpolation method. Edges has the most important details in an image. Hence a loss or smoothness of edges may result in loss of sharpness, blur and artifacts is introduced in the image. The efficiency of linear interpolation technique over non-linear interpolation technique is well known. The advantage of the linear interpolation technique helps it to be used in many real-world applications even though it has its drawbacks. In order to overcome the poor performance of the linear interpolation technique we propose an interpolation technique that uses filter to preserve the edges.

**3.1 Interpolation filters**

The interpolation filters are used in the video coding. The interpolation operator used for sub-pixel accuracy in the video coding can be designed in various ways and it can impact the quality of the reconstructed frame. Among the various types of interpolation filters the most common well-known types of filter are Bilinear filter and 6-tap filter. These filters are used in H.264 coding standard. Dionisis Athanasopoulos and Thanos Stouraitis [13] compared the performance of Bilinear [1 1]/2 and the 6-tap filter [1 -5 20 20 -5 1]/32 and other filters and found out that the 6-tap filter works better than the other filters. We have used 6-tap filter in our project because of the better performance and the high pass characteristics to obtain sharp digital X-ray images. The equation of the 6-tap filter is as follows.

Y(z)=(1-5z-1+20z-2+20z-3-5z-4)/32 -----(3.1)

The choice of 6-tap filter with coefficients [1 -5 20 20 -5 1]/32 is made because it has more high pass and less low pass behavior which preserves edges in the image.

**3.2 Scheme**

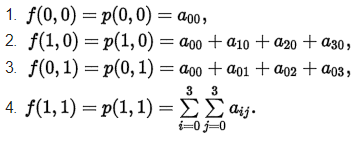
Our project works in three phases. In the first phase bicubic interpolation is performed on the LR image. In the second phase image obtained bicubic interpolated image is passed through the 6-tap filter to produce a column interpolated image. In third phase column interpolated image is passed through the 6-tap filter to produce the row interpolated image and the image obtained is the final super resolution image.

**3.2.1 Bicubic Interpolation**

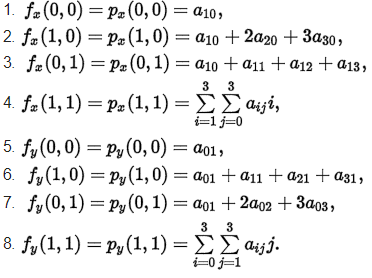
This is the first phase of the project. In this phase a LR image that is an image that is obtained from the HR image by modifying it to LR is taken. Then the bicubic interpolation on the image is done. Bicubic interpolation considers 16 pixels (4x4) for finding the unknown values. The final image that is obtained from the bicubic interpolation are smoother and have fewer interpolation artifacts. The bicubic interpolation on the image is computed as follows[14].

Let us consider that we have a function values f and the derivatives fx,fy,fxy are known at the four corners (0,0), (1,0), (0,1) and (1,0) of the unit square. The interpolated surface can be written as

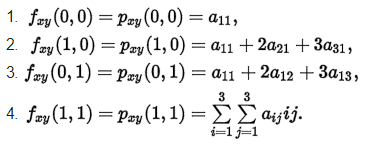
The interpolation problem consists of determining the 16 coefficients aij. Matching p(x,y) with the function values yields four equations:

 ---- (3.3)

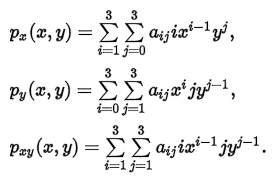
Likewise, eight equations for the derivatives in the x and y directions:

 ----(3.4)

And four equations for the xy mixed partial derivatives

 ----(3.5)

The expressions above have used the following identities

 ----(3.6)

This procedure yields a surface {\displaystyle p(x,y)}p(x,y) on the [unit square](https://en.wikipedia.org/wiki/Unit_square) {\displaystyle [0,1]\times [0,1]}[0,1] x [0,1]  that is continuous and has continuous derivatives. Bicubic interpolation on an arbitrarily sized [regular grid](https://en.wikipedia.org/wiki/Regular_grid) can then be accomplished by patching together such bicubic surfaces, ensuring that the derivatives match on the boundaries.

Grouping the unknown parameters {\displaystyle a\_{ij}}aij in a vector

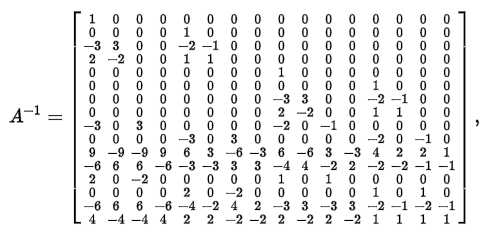
(3.7)

and letting

-(3.8)

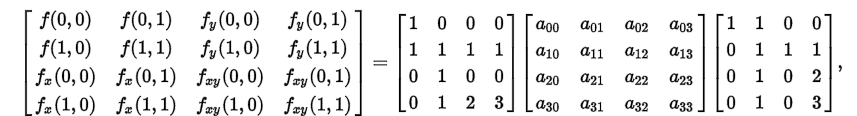
the above system of equations can be reformulated into a matrix for the linear equation Aα=x.

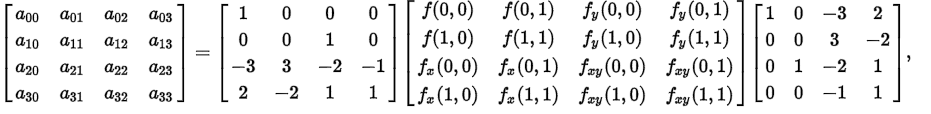
Inverting the matrix gives the more useful linear equation A-1x=α, where

(3.9)

Which allows α to be calculated quickly and easily.

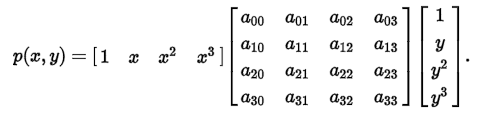
There can be another concise matrix form for 16 coefficients:





(3.10)

Where,

 (3.11)

For finding the single derivatives, fx or fy using the method, find the slope between the two surrounding points in the appropriate axis. For example, to calculate a x derivative at one points, find f(x,y) for the point to left and right of the target point and calculate their slope, similarly for y derivate.

For the cross derivative fxy, take the derivative in both axes one at a time. For example, first find a x derivative for a value above the target point. Then find the x derivate for point below the target point and find the y derivative for both points. This can also be done by first finding the y derivative and then x derivative.

At the edges of the dataset, when one or more pixel is missing they can be approximated by a number of methods. A simple and common method is to assume that the slope from the existing point to target point continues without further change, and using this to calculate a hypothetical value for the missing point.

Thus, by using this computation bicubic interpolation is done on the LR image and bicubic interpolated image is obtained. For example, if a LR image of size 4x4 is give the output of bicubic is 7x7 image.

LR pixel

LR image

HR pixel

LR pixel

LR pixel

Pixel to be interpolated

HR Grid of Image

Bicubic Interpolation

**Figure 3.1 An illustration of Bicubic interpolated image obtained from LR image**

**3.2.2 Column Interpolation**

This is the second phase of our project. In this phase the bicubic interpolated image that is obtained from the first phase is taken. Then it is given to the filter to produce the filtered image. The filter is a 6-tap filter with coefficients [1 -5 20 20 -5 1]/32. The filtered is applied on the image by using the following methods.

Linear filtering of an image is accomplished through an operation called convolution. Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. The matrix of weights is called the convolution kernel, also known as filter. A convolution kernel is a correlation kernel that has been rotated 180 degrees.

The following steps can be used to calculate the pixel output at position (x,y)

1. Rotate the correlation kernel 180 degrees about its center element to create a convolution kernel.
2. Slide the center element of the convolution kernel so that it lies on the top of the (x,y) element of A.
3. Multiply each weight in the rotated convolution kernel by the pixel of A underneath.
4. Sum the individual product from step 3.

Thus, sliding the filter along the image produce the filtered image. The produced image will be one-pixel size less than that of the original image. That is if the input image is of size 8x8 then the filtered image will be of size 7x7. To get the filtered image same as the size of the input image the other values when filter is moved is assumed to be zeros.

Thus, the obtained filtered image and the bicubic interpolated image are combined together i.e. a single column of bicubic interpolated image and a single column of filtered image and so on. They are arranged alternatively to produce a column interpolated image. The bicubic interpolated image that is of size 7x7 is given. Then a column interpolated image of size 7x13 is obtained after column interpolation.

Filtered Image

Bicubic Interpolation

2D-Filtering using Correlation

Column Interpolation

**Figure 3.2 An illustration of Column Interpolated image obtained from Bicubic Interpolated image**

**3.2.3 Row Interpolation**

This is the third and final phase in our project. The column interpolated image that is obtained in the previous phase is passed again through the 6-tap filter. The final filtered image is obtained. The filtered image output is obtained in the same method the filter image obtained in column interpolation. The column interpolated image and the final filtered image are combined together i.e. a single row of column interpolated image and a single row of final filtered image and so on. They are arranged alternatively to obtain the final super resolution image. The column interpolated image of size 7x13 is given to row interpolation the final super resolution image of size 13x13 is obtained.

2D-Filtering using Correlation

Row Interpolation

Column Interpolation

Final Filtered Image

**Figure3.3 An illustration of row interpolated image obtained from the column interpolated image.**

Super Resolution image

**Figure 3.4 An illustration of Super Resolution image obtained from LR image by proposed method.**

Super Resolution image

Column Interpolation

Final Filtered Image

2D-Filtering using Correlation

2D-Filtering using Correlation

Bicubic Interpolation

HR Grid of Image

LR pixel

LR pixel

LR pixel

Pixel to be interpolated

HR pixel

Filtered Image

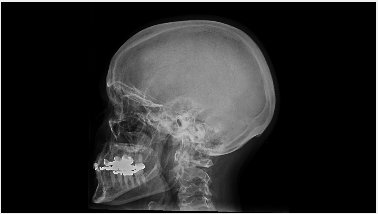
**Chapter 4**

**Implementation**

We have implemented our project using Matlab. The images for implementation was taken from the GE healthcare website. The images are digital X-ray RGB images. We have implemented our project using four functions they are resizing, bicubic, filter interpolation, PSNR.

**4.1 Resizing**

The resizing function is used to resize the image. The resizing of the image is needed because we consider the picture that is obtained from the GE healthcare [15] website as the HR image. In order to compare our output obtained from proposed method we use the image as HR image to compare it. Thus, we use the resizing function to produce a LR image. The resizing image takes HR image and resizing factor as the input arguments and produce LR image as output. In our sample images they are of size 1500x844 when resizing factor is 4 the output image of size 375x211 is produced as the output. The function name in our code is bicubic resize and the sample of LR image that is obtained from passing through this function is shown in figure 4.1.



**Figure 4.1 Image of skull after passing HR image to Resizing function.**

**4.2 Bicubic**

This function is used to do the bicubic interpolation on the LR image. For doing this we create two pairs of arrays one same size of the image and the other one is of size how many times the image has to be increased. This is done by using the meshgrid command. For example [X,Y] = meshgrid(x,y) [16] transforms the domain specified by vectors x and y into arrays X and Y, which can be used to evaluate functions of two variables and three-dimensional mesh/surface plots. The rows of the output array X are copies of the vector x; columns of the output array Y are copies of the vector y. The LR image values are converted to double and the values of meshgrids and image double values are given to the interp2 command to calculate the bicubic interpolated image. Then the bicubic interpolated image is obtained as the output. The input arguments for this function are LR image and how many times the image has to be increased. The LR image of size 375x211 is given to bicubic function along with increasing factor of 2 the output image of size 750x422 is produced as output. The bicubic interpolated image is shown in figure 4.2.

**Figure 4.2 Image of skull after bicubic interpolation of LR image.**

**4.3 Filter Interpolation**

The filter interpolation function is used to apply the filter to the bicubic image and then do the column interpolation and row interpolation to the image and finally produce the output super resolution image. For applying the filter to the function, we have used filter2 command of Matlab. It performs the 2D correlation and yields the central component of the correlation as the outcome that is of equal dimension of the input image. The 2D correlation is performed by implementing 2D convolution with the filter coefficient rotated 180 degrees. Thus, the filter2 rotates the 6-tap filter with coefficients [1 -5 20 20 -5 1]/32 180 degrees to create a convolution kernel[17], it then calls con2, the 2D convolution function of Matlab, to implement the filtering operation. By default, filter2 then selects the central component of the convolution that is the same size as the input image, and returns this as the result. Then the filtered image is arranged alternatively to form column interpolated image and again the filter is applied using the filter2 function to produce the final filtered image. Then the image is arranged alternatively in row wise to form the row interpolated image. The final output is obtained from this function. The input arguments of this function are bicubic interpolated image and the filter coefficient. The output image obtained from this function is of size 1499x843.The final super resolution image obtained from this function is shown in figure 4.3

**Figure 4.3 Super resolution image of skull obtained from bicubic interpolated image.**

**4.4 PSNR**

PSNR is the Peak signal-to-noise ratio. It is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is an approximation to human perception of reconstruction quality. Highest PSNR indicates higher quality but in some case it may not. PSNR is defined via mean squared error (MSE).

MSE = --(4.1)

Where Iup(i,j) is the output image of proposed system Iorig(i,j) is the original image, W and H are the image dimensions.

PSNR = 10 log10 (MAXPIX2/MSE) --(4.2)

Where MAXPIX is the maximum possible value of the image. When the pixel is represented using 8 bit per sample, this is 255. If the pixel is represented in decimal the value is1.

The PSNR function is used to calculate the PSNR values between the original image and the processed image and gives the output of PSNR value. The implementation in Matlab is done in such a way that it works for both RGB and gray scale images.

**Chapter 5**

**Experimental Results**

A set of seven digital X-ray images were taken from the GE healthcare website to compare the performance of our scheme with iNEDI and ICBI interpolation methods. All the seven images are resized to 50x50x3 in order to reduce the process time of ICBI and iNEDI methods. The figure 5.1 shows all seven digital X-ray images resized. Then the resized images are enlarged 4X and 8X using iNEDI, ICBI and proposed method. Then the PSNR ratio of all seven digital X-rays are calculated using the equation (5.2).

MSE = --(5.1)

PSNR = 10 log10 (MAXPIX2/MSE) --(5.2)

Where Iup(i,j) is the output image of proposed system Iorig(i,j) is the original image, W and H are the image dimensions, MAXPIX is the maximum possible value of the image.



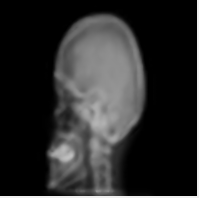
1. (b) (c) (d) (e)

(f) (g)

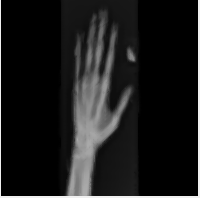
**Figure 5.1 Digital X-ray images from GE health care (a) skull, (b) hand, (c) foot, (d) chest, (e) aircon, (f) t-Spline, (g) pelvis.**

For the purpose of testing the images we resized each image into two different sizes one image is of size 50x50x3 this image is considered as LR image and the other image is of size 200x200x3 this image is considered as HR image for 4X enlargement and image is resized to 400x400x3 for 8X enlargement is considered as HR image. This HR image is used to compare with the image that is obtained from interpolation for calculating the PSNR.

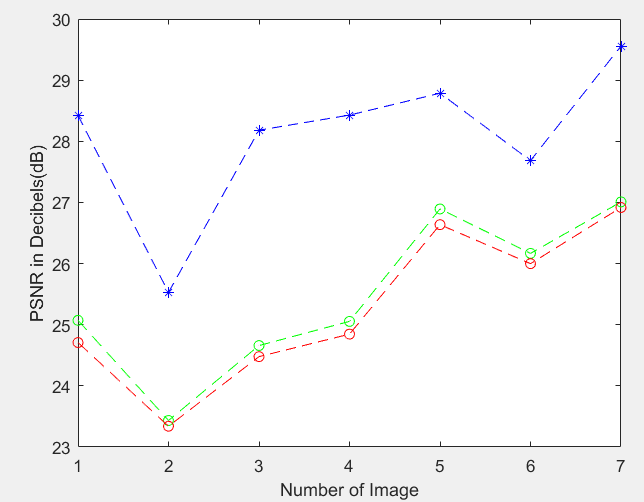
The results that are obtained from the proposed method shows sharpness in the perceived quality of images and it seems to be more effective than other methods. We can also observe that annoying ringing artifacts are dramatically suppressed in the interpolated images in our scheme due to 2D interpolation filtering. The figure 5.2 shows the 4X enlargement of skull using different scheme. It can be observed that our scheme generates the image with highest quality. The figure 5.3 shows the 4X enlargement of hand using different scheme.

1. (b) (c)

**Figure 5.2 An illustration of 4X enlargement of digital X-ray RGB skull image of size 200x200x3. (a) iNEDI, (b) ICBI, (c) Proposed Scheme**

 (a) (b) (c)

**Figure 5.3 An illustration of 4X enlargement of digital X-ray RGB hand image of size 200x200x3. (a) iNEDI, (b) ICBI, (c) Proposed Scheme.**

The figure 5.4 shows the performance of our scheme versus iNEDI and ICBI scheme in terms of PSNR value of 4X enlargement of image. It can be seen that our scheme clearly outperforms the other schemes. The plot is done using Matlab.

--\* Proposed Method

--o ICBI

--0 iNEDI

**Figure 5.4 Plot of PSNR vs 4X enlarged Images for seven images taken from GE healthcare using ICBI, iNEDI, Proposed Scheme.**

**5.1 Discussions**

PSNR ratio obtained by comparing the image that is up-scaled by approximately 4X and 8X factors with the reference image is summarized in table 5.1 and table 5.2 respectively. The new proposed method has an average PSNR of 28.0840 dB and 27.0799 in 4X enlargement and 8X enlargement respectively that is clearly higher than that of the other methods.

The proposed interpolation scheme obtained an average increment of 2.61 dB when compared to ICBI technique and 2.81 dB when compared to iNEDI technique when the images are enlarged 4X time. The proposed interpolation scheme obtains an average increment of 2.81 dB when compared to ICBI technique and 3.04 when compared to iNEDI technique when the images are enlarged 8X time.

The results of both ICBI and iNEDI technique are obtained with a Matlab implementation. Thus it can be seen that our interpolation scheme is clearly superior to iNEDI and ICBI techniques and it is also computationally faster than both methods.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Image** | **T-spline** | **Aircon** | **Chest** | **Hand** | **Pelvis** | **Foot** | **Skull** | **Avg** |
| **ICBI** | 25.0712 | 23.4335 | 24.6593 | 25.0548 | 26.8935 | 26.1670 | 27.0084 | 25.47 |
| **iNEDI** | 24.7085 | 23.3393 | 24.4780 | 24.8457 | 26.6362 | 25.9985 | 26.9173 | 25.27 |
| **Our method** | 28.4225 | 25.5284 | 28.1812 | 28.4284 | 28.7861 | 27.6841 | 25.5284 | 28.08 |

**Table 5.1 PSNR(dB) obtained on 4X enlargement digital X-ray images with the proposed, iNEDI, ICBI method.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Image** | **T-spline** | **Aircon** | **Chest** | **Hand** | **Pelvis** | **Foot** | **Skull** | **Avg** |
| **ICBI** | 23.8609 | 22.2408 | 23.4554 | 23.7243 | 25.7784 | 24.9607 | 25.8801 | 24.27 |
| **iNEDI** | 23.4761 | 22.0176 | 23.3098 | 23.4743 | 25.5201 | 24.7071 | 25.7742 | 24.04 |
| **Our method** | 27.6687 | 24.5837 | 27.2150 | 27.5999 | 27.8588 | 26.2744 | 28.3597 | 27.08 |

**Table 5.2 PSNR(dB) obtained on 8X enlargement digital X-ray images with the proposed, iNEDI, ICBI method.**

**Chapter 6**

**Conclusion**

In this project we have presented an interpolation method that is more accurate when compared to the other single image interpolation technique for digital X-rays images. By using our method, we can notice that the edge artifacts and over smoothing problems are decreased and the blurring effect on the image is eliminated. In our method we overcome the existing problems of linear interpolation by amalgamation of bicubic interpolation and 2D interpolation by using a 6-tap filter. The performance of our method is compared with the existing method such as ICBI and iNEDI using Matlab. The result obtained showed that proposed method has good performance with consistent PSNR performance.

**Appendix**

**Matlab Code**

**Resizing function**

function resized\_image=bicubicResize(image,factor)

% This function is used for obtaining resized image

% input: image: image to resize

% factor: factor to resize the image by

% output: resized\_image: image obtained after resizing

if (length(size(image)) == 3)

% if the input image is an RGB image

% resize the three components separetely

for i=1:3

resized\_image(:,:,i) = bicubicResize(image(:,:,i),factor);

end

else

[m n]=size(image);

reim=uint8(zeros(m/factor,n/factor));

for k=1:m/factor

for j=1:n/factor

reim(k,j)=image(factor\*k-1,factor\*j-1);

end

end

resized\_image=reim;

end

**Bicubic Function**

function processed\_image=bicubic(image,zoom)

% This function is used for obtaining bicubic of image

% input: image: image to do bicubic on

% zoom: 2x or 4x increase in size using bicubic

% output: processed\_image: image obtained after bicubic

if (length(size(image)) == 3)

% if the input image is an RGB image

% interpolate the three components separetely

%

for i=1:3

processed\_image(:,:,i) = bicubic(image(:,:,i),zoom);

end

else

[m n]=size(image);

[X Y]=meshgrid(1:n,1:m);

if zoom==2

[X1 Y1]=meshgrid(linspace(1,n,2\*n), linspace(1,m,2\*m));

elseif zoom==4

[X1 Y1]=meshgrid(linspace(1,n,4\*n), linspace(1,m,4\*m));

end

l=im2double(image);

bicubicimage=interp2(X,Y,l,X1,Y1,'cubic');

processed\_image=bicubicimage;

end

**Filter**

function filtered\_image = fil(im, filter\_coef)

% This function applies the specified filter

% Input: im: input image

% filter\_coef: filter coefficients

% example: filter\_coef = [1 1]/2 (Bilinear Filter)

% filter\_coef = [1 -5 20 20 -5 1]/32 (6-tap filter)

% Output: filtered\_image:Output after applying the filter to the image

if (length(size(im)) == 3)

% if the input image is an RGB image

% just apply the filter to three components separetely

for i=1:3

filtered\_image(:,:,i) = fil(im(:,:,i),filter\_coef);

end

else

filtered\_image= filter2(filter\_coef,im);

end

**Filter Interpolation**

function filtered\_image = filter\_interp(image, filter\_coef)

% Interpolation function

% This function performs the interpolation of the input image,

% using the specified filter

%

% Input: image: input image

% filter\_coef: filter coefficients

% example: filter\_coef = [1 1]/2 (Bilinear Filter)

% filter\_coef = [1 -5 20 20 -5 1]/32 (6-tap filter)

% Output: filtered\_image: interpolated image

if (length(size(image)) == 3)

% if the input image is an RGB image

% just interpolate the three components separetely

for i=1:3

filtered\_image(:,:,i) = filter\_interp(image(:,:,i),filter\_coef);

end

else

[m,n] = size(image);

image\_filter\_ = []; filtered\_image = [];

% columns interpolation

image\_filter\_col= filter2(filter\_coef,image);

for i=1:n

image\_filter\_ = [image\_filter\_ image(:,i) image\_filter\_col(:,i)];

end

image\_filter\_(:,end) = [];

% rows interpolation

image\_filter\_rows = filter2(filter\_coef,image\_filter\_')';

for i=1:m

filtered\_image = [filtered\_image; image\_filter\_(i,:); image\_filter\_rows(i,:)];

end

filtered\_image(end,:) = [];

end

**PSNR RGB**

function PSNRCol = PSNRRGB(Original,Filtered)

%This function Calculates the PSNR of RGB image

%Input: Original: Original image

% Filtered: Image obtained after applying filters

% and after interpolating

%Output: PSNR: PSNR value of the Filtered image

[m n]=size(Filtered(:,:,1));

Red=Original(:,:,1);

for k=1:m

for j=1:n

ModifiedR(k,j)=Red(k,j);

end

end

ModifiedR=im2double(ModifiedR);

FilterR=im2double(Filtered(:,:,1));

FiltcropR=imcrop(FilterR,[1 1 n-1 m-1]);

ModcropR=imcrop(ModifiedR,[1 1 n-1 m-1]);

[a b]=size(ModcropR);

eR=imsubtract(FiltcropR,ModcropR);

seR=(sum(eR(:).^2));

mseR=seR/(a\*b);

[m n]=size(Filtered(:,:,2));

Green=Original(:,:,2);

for k=1:m

for j=1:n

ModifiedG(k,j)=Green(k,j);

end

end

ModifiedG=im2double(ModifiedG);

FilterG=im2double(Filtered(:,:,2));

FiltcropG=imcrop(FilterG,[1 1 n-1 m-1]);

ModcropG=imcrop(ModifiedG,[1 1 n-1 m-1]);

[a b]=size(ModcropG);

eG=imsubtract(FiltcropG,ModcropG);

seG=(sum(eG(:).^2));

mseG=seG/(a\*b);

[m n]=size(Filtered(:,:,3));

Blue=Original(:,:,3);

for k=1:m

for j=1:n

ModifiedB(k,j)=Blue(k,j);

end

end

ModifiedB=im2double(ModifiedB);

FilterB=im2double(Filtered(:,:,3));

FiltcropB=imcrop(FilterB,[1 1 n-1 m-1]);

ModcropB=imcrop(ModifiedB,[1 1 n-1 m-1]);

[a b]=size(ModcropB);

eB=imsubtract(FiltcropB,ModcropB);

seB=(sum(eB(:).^2));

mseB=seB/(a\*b);

mse=(mseR+mseG+mseB)/3;

PSNRCol=10\*log10(1/mse);

**PSNR Grayscale**

function p = PSNR(ip,op)

op=rgb2gray(op);

ip=rgb2gray(ip);

[m n]=size(op);

% processed\_image=rgb2gray(processed\_image);

for k=1:m

for j=1:n

u(k,j)=ip(k,j);

end

end

u=im2double(u);

op=im2double(op);

% processed\_image=im2double(processed\_image);

y=imcrop(op,[1 1 n-1 m-1]);

p=imcrop(u,[1 1 n-1 m-1]);

[a b]=size(p);

e=imsubtract(y,p);

se=(sum(e(:).^2));

mse=se/(a\*b);

psnr=10\*log10(1/mse);

p = psnr;

end

**Proposed Method**

function []=proposed()

image = imread('D:\spring17\images\skull.jpg');

processimage=bicubicResize(image,4);

figure,imshow(processimage),title('Original Image');

bicubicimage=bicubic(processimage,2);

figure,imshow(bicubicimage),title('Bicubic Image');

filter\_coef = [1 -5 20 20 -5 1]/32 ;

filterinterpimage = filter\_interp(bicubicimage, filter\_coef);

figure,imshow(filterinterpimage),title('Filtered Interpolated Image');

if (length(size(image))==3)

psnr = PSNRRGB(image,filterinterpimage)

else

psnr = PSNRImages(image,filterinterpimage)

end

**Proposed Test**

function []=proposedtest()

image=imread('D:\spring17\images\hand.jpg');

zoom=8;

processimage=imresize(image,[50,50],'bicubic');

%filter\_coef = [1 1 1; 1 1 1;1 1 1]/9 ;

figure,imshow(processimage)

filter\_coef = [1 -5 20 20 -5 1]/32 ;

if zoom==4

inputimage=imresize(image,[200,200],'bicubic');

bicubicimage=bicubic(processimage,2);

elseif zoom==8

inputimage=imresize(image,[400,400],'bicubic');

bicubicimage=bicubic(processimage,4);

end

filteredimage = filter\_interp(bicubicimage, filter\_coef);

figure,imshow(filteredimage)

if (length(size(image))==3)

psnr = PSNRRGB(inputimage,filteredimage)

else

psnr = PSNRImages(inputimage,filteredimage)

end

**ICBI test**

function []=icbitest()

image=imread('D:\spring17\images\hand.jpg');

inputimage=imresize(image,[200,200],'bicubic');

processimage=imresize(image,[50,50],'bicubic');

%inputimage=imresize(processimage,[200,200],'bicubic');

% input arguments for icbi is icbi(IM, ZK, SZ, PF, VR);

% IM : Source image.

% ZK : Power of the zoom factor (default = 1)

% SZ : Number of image bits per layer (default = 8).

% PF : Potential to be minimized (default = 1).

% VR : Verbose mode, if true prints some information during calculation

% (default = false).

processedimage = icbi(processimage, 2, 8, 1, true);

figure,imshow(processedimage)

if (length(size(inputimage))==3)

psnr = PSNRRGB(inputimage,processedimage)

else

psnr = PSNRImages(inputimage,processedimage)

end

**iNEDI test**

function []=ineditest()

image=imread('D:\spring17\images\hand.jpg');

inputimage=imresize(image,[200,200],'bicubic');

processimage=imresize(image,[50,50],'bicubic');

%input arguments for inedi is inedi(IM, ZK, MT, ML, BT, BS, SZ, VR);

%IM: Input image

%ZK:the image enlargement on vertical and horizontal direction is 2^ZK

%MT:maximum radius of the local circular window

%ML:minimum radius of the local circular window

%BT:brightness tollerance

%BS:maximum brigthess difference for smooth areas

%SZ:number of image bits per layer

%VR:verbose mode

processedimage=inedi(processimage, 2, 6, 3, 16, 8, 8, true);

figure,imshow(processedimage)

if (length(size(inputimage))==3)

psnr = PSNRRGB(inputimage,processedimage)

else

psnr = PSNRImages(inputimage,processedimage)

end

**References**

[1] W. C. Siu and K. W. Hung, "Review of image interpolation and super-resolution," Proceedings of The 2012 Asia Pacific Signal and Information Processing Association Annual Summit and Conference, Hollywood, CA, 2012, pp. 1-10.

[2] R. Y. Tsai and T. S. Huang, "Multiframe image restoration and registration," in Advances in Computer Vision and Image Processing, vol. 1, chapter 7, pp. 317-339, JAI Press, Greenwich, Conn, USA, 1984.

[3]  M. Irani and S. Peleg. 1991, "Super Resolution From Image Sequences" ICPR, 2:115--120, June 1990.

[4] Kian Kee Teoh, H. Ibrahim and S. K. Bejo, "Investigation on several basic interpolation methods for the use in remote sensing application," 2008 IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, Cyberjaya, 2008, pp. 60-65.

[5] [Mitsuo Eguchi](https://www.google.com/search?tbo=p&tbm=pts&hl=en&q=ininventor:%22Mitsuo+Eguchi%22), [Tetsuhiko Yoshida](https://www.google.com/search?tbo=p&tbm=pts&hl=en&q=ininventor:%22Tetsuhiko+Yoshida%22) ,“Degradation information restoring method and device “ WO 2006041127.

[6] S. K. Maurya, P. K. Mishra, R. K. Singh and A. K. Misra, "Image Enhancement by Intensity Based Interpolation and Selective Threshold," 2012 International Conference on Communication Systems and Network Technologies, Rajkot, 2012, pp. 174-178.

[7] K. Kita, M. Yoshioka and K. Inoue, "Single frame super resolution using bilateral filter," 2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS), Otsu, 2017, pp. 1-7.

[8] J. Allebach and Ping Wah Wong, "Edge-directed interpolation," Proceedings of 3rd IEEE International Conference on Image Processing, Lausanne, 1996, pp. 707-710 vol.3.

[9] X. Li and M. T. Orchard, “New edge-directed interpolation,” *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.

[10] N. Asuni and A. Giachetti, “Accuracy improvements and artifacts removal in edge based image interpolation,” *in Proc. 3rd Int. Conf. Compit. Vis.* *Theory Appl (VISAPP)*, pp. 58–65, 2008.

[11] E. Hosogai and Y. Tanaka, "Iterative edge-directed image upscaling with high directionality," Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific, Siem Reap, 2014, pp. 1-4.

[12] Andrea Giachetti, Nicola Asuni. "Fast Artifacts- Free Image Interpolation". Leeds, UK : *Proceedings of the British Machine Vision Conference (BMVC)*, September, 2008,pp. 10.

[13] Stouraitis, Athanasopoulos Dionysios and Thanos. "Content-Adaptive Wavelet-Based Scalable Video Coding". New Orleans, LA : *ICAS*, 2007. 1-4244-0920-9,pp. 981-984.

[14] [Online] <https://en.wikipedia.org/wiki/Bicubic_interpolation>

[15] [Online] GE HealthCare, 2015. http://www3.gehealthcare.in/en/products/categories/radiography-and-fluoroscopy/flashpad-wireless-digital detector#tabs/tab7543D897AFF144AB808437F6ECA81AF5

[16] MATLAB Function Reference. [Online] The MathWorks, Inc., 1994-2017. https://www.mathworks.com/help/matlab/ref/filter2.html

[17] Losada, Ricardo A. Practical FIR Filter Design in MATLAB. USA : The MathWorks, Inc., January 12, 2004, pp. 32.