An Adaptive Zooming Algorithm for Images

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering

in

Computer Science and Engineering

By:

Mayank Kumar

(Regn. No. 80732013)

Under the supervision of

Mr. Singara Singh
Lecturer
SMCA

Mr. Ajay Kumar
Lecturer
CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

JULY 2009

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"An Adaptive Zooming Algorithm for Images"**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science & Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Mr. Singara Singh** and **Mr. Ajay Kumar** and refers other researcher's works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

(Mayank Kumar)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Mr Singara Singh)
Lecturer
School of Mathematics & Computer Applications
Thapar University
Patiala

(Mr Ajay Kumar)
Lecturer
Computer Science & Engineering Department
Thapar University
Patiala

Countersigned by

Head
Computer Science & Engineering Department
Thapar University,
Patiala

(Dr. R. K. Sharma)
Dean (Academic Affairs)
Thapar University,
Patiala.

Acknowledgment

I wish to express my deep gratitude to Mr Singara Singh, Lecturer, School of Mathematics And Computer Applications, TU, Patiala and Mr Ajay Kumar, Lecturer, Computer Science & Engineering Department, TU, Patiala for providing their uncanny guidance and support throughout the thesis.

I am thankful to Head, Computer Science & Engineering Department, TU, Patiala, for the motivation and inspiration that triggered me for the thesis work. I would also like to thank all the staff members who were always there at the need of the hour and provided with all the help and facilities, which I required for the completion of the thesis.

Last but not the least, I express my heartfelt thanks to all my friends Sushil Kumar, Jasmeet Singh, Sandeep Khode, Vaibhav Bhadade, Amit Gupta, Mukesh Kumar, Subhash Junas, Vineet Khera for encouraging me and providing me useful information during my work.

Most importantly, I would like to give God the glory for all of the efforts I have put into this report.

Finally, my special thanks go to authors whose works I have consulted and quoted in this work.

Mayank kumar
80732013
M.E.(Computer Science & Engineering)-2nd year
Computer Science & Engineering Department
Thapar University
Patiala -147004

Abstract

Image zooming is the process of enlarging the image to as desired magnification factor. But while enlarging an image there are few parameters that we have to keep in mind. When the image is zoomed, artifacts like blurring, jagging and ghosting may arise. So the main focus is on the reduction of these artifacts.

Our algorithm deals with the edges. It is basically designed to preserve the edges. It’s as adaptive zooming algorithm which focuses on preserving edges. Our algorithm reduces the jagging. Blurring is reduced a lot in our algorithm.

To compare our algorithm with existing algorithms, we have taken few real world images and results are visually compared. And we have come to the decision that our algorithm is better than the traditional methods. We have compared the images by four ways – MAE, MSE, CCC, and PSNR.

Abbreviations

PSNR	Peak Signal to Noise Ratio
FIR	Finite Impulse Response
MSE	Mean Squared Error
MAE	Mean Absolute Error
CCC	Cross Correlation Coefficient

Table of Contents

List of Figures

List of Tables

Chapter 1 – Introduction

1.1 Image zooming

Zooming is the process of enlarging something only in appearance, not in physical size. This enlargement is quantified by a calculated number, called *magnification*. When this number is less than one it refers to a reduction in size, called *minification*.

Image zooming [1] is among the fundamental image processing operations. Typically zooming is related to scaling up visuals or images to be able to see more detail, increasing resolution, using optics, printing techniques, or digital processing. In all cases, the zooming of the image does not change the perspective of the image. Applications are varied in different fields. In medical imaging, zooming can serve to improve the chances of diagnosing problems by highlighting any possible aberrations. Enhancing image details can also be useful for the purposes of identification, whether for improving the quality of an image interpreted by a biometric recognition system or trying to get a clearer view of the perpetrator of some crime. In entertainment, zooming can be used to resize a video frame to fit the field of view of a projection device, which may help to reduce blurring. Finally, the most obvious application of image zooming is to simply allow one to enjoy a larger version of a favorite image obtained from any commercially available digital imaging device such as a camera, camcorder or scanner.

Traditional image zooming techniques use up-sampling by zero-insertion followed by linear filtering to interpolate the high-resolution samples. The main drawback of this approach is that the frequency content of the high-resolution image is the same as the low-resolution image. This is due to the fact that linear techniques are incapable of introducing new information into the image. The lack of new high frequency content results in a variety of undesirable image artifacts such as blocking, staircase edges and blurring.

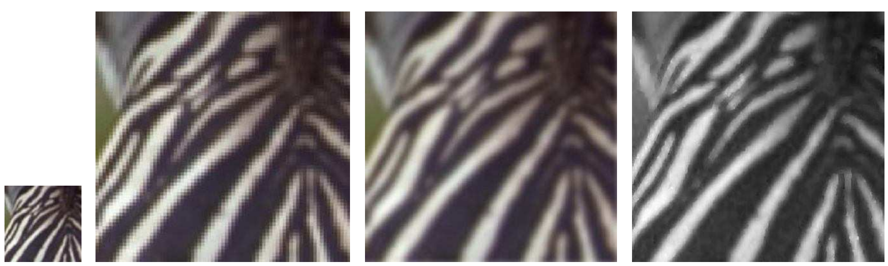
1.1.1 Zooming through Interpolation

Interpolation is the process of determining the values of a function at positions lying between its samples. It achieves this process by fitting a continuous function through the discrete input samples. This permits input values to be evaluated at arbitrary positions in the input, not just those defined at the sample points. While sampling generates an infinite bandwidth signal from one that is band limited, interpolation plays an opposite role: it reduces the bandwidth of a signal by applying a low-pass filter to the discrete signal. That is interpolation reconstructs the signal lost in the sampling process by smoothing the data samples with an interpolation function.

The process of interpolation is one of the fundamental operations in image processing. The image quality highly depends on the used interpolation technique.

Image resolution, or the appearance of resolution, is important in many aspects of today's digital world. For instance, high-resolution cameras are able to digitize scenes at a much finer scale and thus capture much more detail than lower-resolution cameras. Unfortunately, not all pictures and images can be stored at high resolution due to equipment, memory, and in the case of the Internet, bandwidth limitations.

Consumers still need low-resolution images to be enlarged to higher resolution for viewing, printing, and editing, creating a need for interpolation algorithms that give end-users these magnified images.



(a) (b) Jaggies (c) Blurring (d) Ghosting

Figure 1.1: Zebra image

The above image is magnified and three types of major artefacts are compared. There are many artifacts which are produced (as shown in figure 1.1) during zooming like *jagging*- blocks are formed due to replication of pixels, *burring*- It is unclearity of the image, *ghosting*- it is the distortion of the image. Current interpolation algorithms attempt to solve these artifacts in a number of ways. There are other factors have to be kept in mind while making an algorithm like *speed* (zooming should not take much time for enlargement), *memory requirement* (the algorithm should not take much memory space) [2].

1.1.2 Zooming as Number

A magnification is the ratio between the apparent size of an object and its true size, and thus it is a dimensionless number. In general, magnification is divided as:

(i) Linear or transverse magnification — For real images, such as images projected on a screen, size means a linear dimension (measured, for example, in millimeters or inches).

(ii) Angular magnification — For optical instruments with an eyepiece, the linear dimension of the image seen in the eyepiece (virtual image in infinite distance) cannot be given, thus size means the angle subtended by the object at the focal point (angular size). Strictly speaking, one should take the tangent of that angle (in practice, this makes a difference only if the angle is larger than a few degrees).

By convention, for magnifying glasses and optical microscopes, where the size of the object is a linear dimension and the apparent size is an angle, the magnification is the ratio between the apparent (angular) size as seen in the eyepiece and the angular size of the object when placed at the conventional closest distance of distinct vision of 25 cm from the eye.

1.2 Motivation

Study on image zooming has been done in various literature in variety of ways. Instead of so much work in this field there are very few image zooming software's which provide adaptive zooming methods. Even the latest software uses the traditional linear magnification algorithm. However such software's are recently brought into market.

Many interpolation algorithms currently used in consumer products produce magnifications that include undesirable artifacts (flaws) like blurring, "jaggies", and "ghosting" (Figure 1.1). The consumer, presented with the magnification, will be unsatisfied with the unrealistic image upon seeing the artifacts. As is known, sharpness of edges and freedom from artifacts are two critical factors in the perceived quality of the interpolated image [3]. More advanced magnification methods such as learning-based algorithms require specific training data, large running times, or extensive user input. Still there is a need for a more general interpolation technique that can minimize these flaws.

The motivation of this research is to create a magnification algorithm that creates realistic real-world image magnifications that do not require a large amount of user input. A realistic real world image would be an image that is free from artifacts such as blurring and jaggies. The image must include smooth contours and also rapid edge transitions in areas of the image where sharp edges are found in the low resolution image. Our adaptive zooming algorithm can produce magnifications with these properties.

1.2 Thesis Description

This thesis basically consists of how to do zooming of images while preserving the edges (adaptive zooming). As we all know that this is the field where already a lot of work has already been done. But still our approach is to study all the work earlier done and find a better zooming algorithm. We have started from the literature survey of the earlier techniques. We realized that most of the algorithms focus on the traditional methods, but we have changed our path to adaptive methods. The traditional methods have many artifacts, but we have tried to resolve then.

Firstly, we started from the literature survey. Most of the methods use the interpolation based techniques which is quite useful but it is non adaptive in most of the cases. We have described each and every interpolation based techniques with visual examples.

Secondly, the problem statement has been defined. Here, in depth description of the problem is given.

In next section, the proposed algorithm is explained. Our algorithm is an adaptive based algorithm which basically focuses on the edges.

Later, the result has been evaluated with the help of proper mathematical methods like cross correlation, PSNR, etc.

Finally, the conclusion has been given with the benefits as well as disadvantages of the proposed algorithm.

Chapter 2 – Literature Survey

2.1 Image Zooming Techniques

There are various zooming techniques. Some of them are given as follows:

2.1.1 Linear Techniques:

Linear techniques in the literature, [4]–[7], use linear space-invariant filters to interpolate the high-resolution samples. Common choices of interpolation filter are nearest neighbor, bilinear, bicubic, quadratic, Gaussian and various types of spline functions [4]. Since the theory behind linear interpolation is well established, most of the research on this approach is focused on finding new filters which reduce artifacts introduced by the traditional filters, as well as more efficient implementations. In [5], a modified version of the B-spline is used to obtain interpolation filters with better frequency responses, [6] proposes an FIR filter design method that attempts to account for the properties of human vision, and [7] develops non-separable cubic convolution kernels to replace the traditional separable cubic filter. Due to the relative simplicity and efficiency of linear interpolation techniques, they are the most common approach provided by commercial software packages such as Adobe PhotoShop and Matlab.

2.1.2 Non - Linear Techniques:

Non-linear techniques in the literature, [8]–[10], use non-linear optimization processes constrained by certain image features. In [8], a method which optimizes a convex cost function based on an approximation of the gradient of the high-resolution image from the low-resolution image is presented. This method attempts to preserve edges by adding constraints on their orientation. A different approach is taken by [9], in which the problem is viewed from a geometric perspective. In this method, an image is first linearly interpolated. Then spatial regions of constant intensity are warped such that level curves are smoothed, thereby sharpening boundaries between regions. In [10], a regularized image interpolation method is proposed which focuses on the correct modeling of the image acquisition and display processes.

2.1.3 Transform Methods

Transform techniques in the literature, [11, 12], are primarily focused on the use of multi-resolution decomposition, followed by interpolation applied to each level of the decomposition and/or extrapolation of higher resolution levels. These approaches aim at synthesizing the high frequency components of the magnified image by adapting the interpolation to suit the frequency content contained at each level of decomposition. In [11], higher resolution levels of Laplacian pyramid decomposition are extrapolated from lower ones. Another approach, taken by [12], makes use of a filter bank which extracts edge directional components from the low resolution image and interpolates each sub-band in a directional specific way as to enhance the edges it contains.

2.1.4 Statistical Methods

Statistical techniques in the literature, [13, 14], attempt to estimate the high-resolution image based on the properties of the given low-resolution image. In [13], the high-resolution image is modeled by a Gibbs-Markov random field with specially selected clique potentials to classify the properties of each neighborhood. The chosen potentials allow the classification of pixels by degrees of smoothness or discontinuity, thereby being able to properly handle edges. Another approach creates a set of pixel classifications gathered from the statistics of pixels in typical training images [14]. Once trained, the algorithm interpolates an image by estimating the best filter coefficients (in the mean-square sense) for each neighborhood.

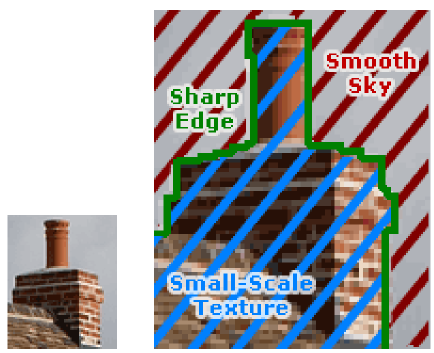
From these four categories, only the first is not an adaptive technique. Non-linear and statistical techniques seem to have found greater use in highly specialized applications such as super-resolution, where a sequence of video frames are combined to form a single, high-resolution image, and medical imaging, where the algorithms are tied to the underlying physics of the image-acquisition process or are highly constrained by prior knowledge of image features.

2.2 Algorithm Subdivision

Common interpolation algorithms can be grouped into two categories: adaptive and non-adaptive. Adaptive methods change depending on what they are interpolating (sharp edges vs. smooth texture), whereas non-adaptive methods treat all pixels equally.

2.2.1 Non-Adaptive Algorithms

It include: nearest neighbor, bilinear, bicubic and others. Depending on their complexity, these use adjacent pixels when interpolating. The more adjacent pixels they include, the more accurate they can become, but this comes at the expense of much longer processing time. These algorithms can be used to both distort and resize a photo.



(a) (b)

Figure 2.1: Enlarged Image using Adaptive Algorithm

2.2.2 Adaptive Algorithms:

It includes many proprietary algorithms in licensed software such as: Qimage, Photo Zoom Pro, Genuine Fractals and others. Many of these apply a different version of their algorithm (on a pixel-by-pixel basis) when they detect the presence of an edge aiming to minimize unsightly interpolation artifacts in regions where they are most apparent. These algorithms are primarily designed to maximize artifact-free detail in enlarged photos, so some cannot be used to distort or rotate an image.

The zoomed image of Figure 2.1(a) using adaptive algorithm is shown in Figure 2.1(b).

2.3 Function Fitting Methods

Interpolation is the estimation of values in a function between known points. When an image is magnified, the high-resolution grid contains pixels that need to be interpolated, or in other words, a value needs to be estimated for them. For instance, in 3x magnification, only 1/9 of the pixels in the high resolution grid are known from the low resolution grid. The remaining 8/9 of the pixels need to be estimated. There are several basic function-fitting methods, including Pixel Replication, Bilinear Interpolation, and Bicubic Interpolation.

2.3.1 Pixel Replication

Pixel replication is by far the simplest and fastest function fitting method. In order to estimate the unknown pixels in the high resolution grid, it simply uses the value of the nearest neighbour, or in other words, the closest original pixel value.

(a) Down Sampled Image (b) 4× Pixel Replication (c) Original image

Figure 2.2: Pixel Replication Example.

In Figure 2.2, original image (2.2c) down sampled by a scale of four (2.2a) then magnified 4× by using Pixel Replication. As shown in Figure 2.2, images made with Pixel Replication are extremely jagged. Hence the higher resolution image is blocky and jagged, since the original pixels have “grown” by exactly the magnification scale.

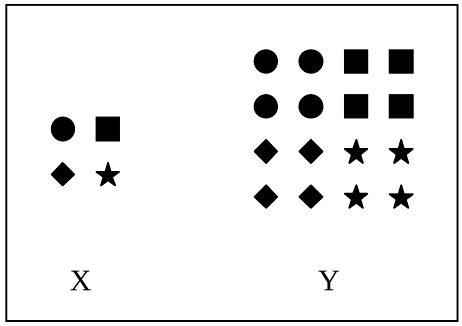


Figure 2.3: Pixel Replication Working

The above figure 2.3 shows how pixel replication actually works. The X image is 2X magnified and in the particular pattern and hence we got the image Y.

2.3.2 Bilinear Interpolation

Bilinear interpolation considers the closest 2x2 neighborhood of known pixel values surrounding the unknown pixel. It then takes a weighted average of these 4 pixels to arrive at its final interpolated value. This results in much smoother looking images than nearest neighbor.

The Figure 2.4 is for a case when all known pixel distances are equal, so the interpolated value is simply their sum divided by four.

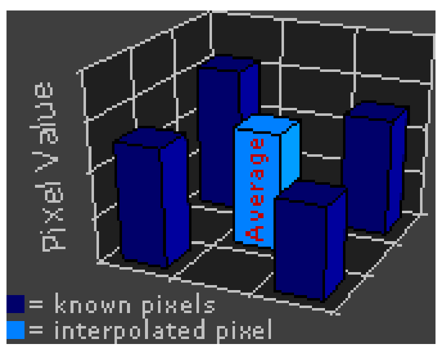


Figure 2.4: Bilinear Interpolation

As can be inferred from the name, Bilinear Interpolation fits a piecewise linear function between known pixel values. While somewhat more complicated than Pixel Replication, the visual results are greatly improved. Figure 2.5 gives an example of linear interpolation in one dimension. As can be seen from the figure, unknown pixel values in the high-resolution grid are estimated to exactly lie on the line that fits between two original pixel values.

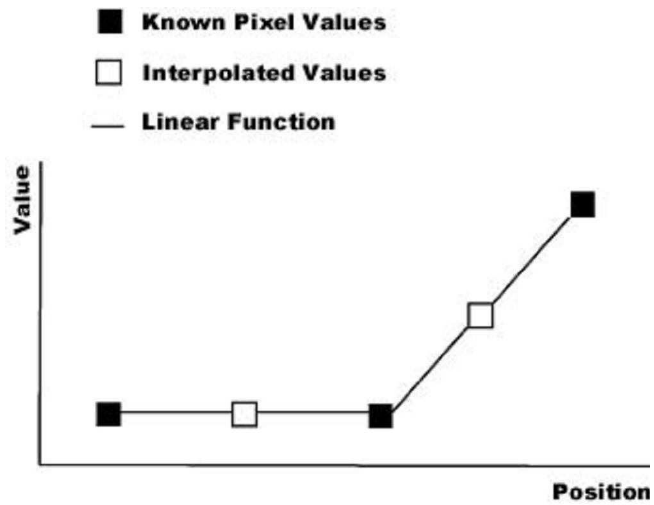


Figure 2.5: Linear Interpolation.

A linear function is fit between known values and all interpolated values fall on the fitted line. For example, in 2x magnification in one dimension, there would be only one unknown pixel between known pixel values. The pixel value would be calculated by taking half of the first known pixel plus half of the second known pixel. In 10x magnification, the first unknown pixel (from left to right) would be estimated by taking nine tenths of the first known pixel value plus one tenth of the second pixel value. This is how interpolation is performed in one dimension. Linear interpolation extends easily into two dimensions.

Bilinear Interpolation can be described as performing linear interpolation in one dimension followed by linear interpolation in the other. For instance, if we are estimating a pixel between a block of four original values, then two temporary values are first created: one, a linear interpolation between the top pair of pixels; and the second, a linear interpolation between the bottom pair of pixels. Lastly, a linear interpolation is performed between the two temporary values (Figure 2.6).

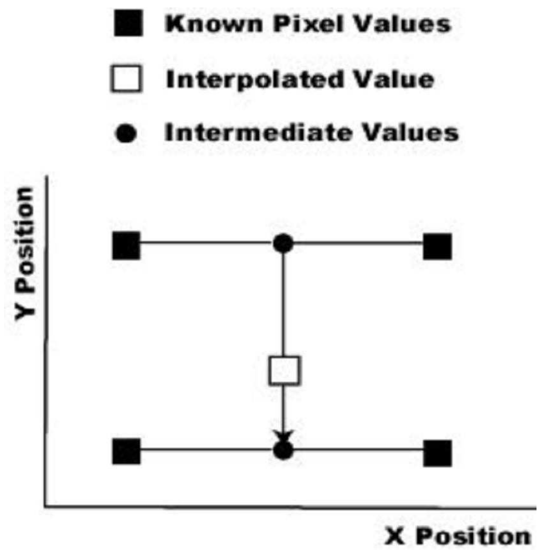
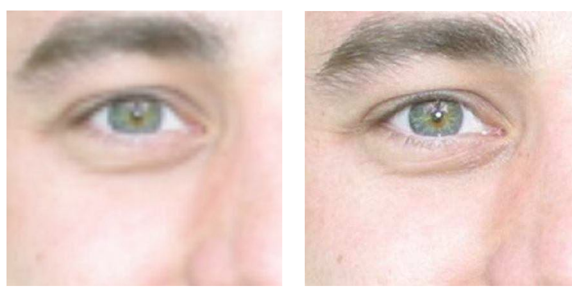


Figure 2.6: Bilinear Interpolation.

(a) Down Sampled Image (b) 4× Pixel Replication



(c) 4× Bilinear Interpolation (d) Original image

Figure 2.7: Pixel Replication vs Bilinear Interpolation.

In Figure 2.7, original image (2.7d) down sampled by a scale of four (2.7a) then magnified 4× by using Bilinear Interpolation. Bilinear Interpolation performs significantly better than Pixel Replication due to the increased accuracy and smoother contours. A bilinear interpolation typically consists of three linear interpolations. Bilinear interpolation produces magnifications that are more visually appealing than Pixel Replication. This can be seen in Figure 2.7. However, due to the interpolation process, the image becomes slightly blurred since pixel values are spread throughout the high-resolution grid.

2.3.4 Bicubic Interpolation

Bicubic goes one step beyond bilinear by considering the closest 4x4 neighborhood of known pixels-- for a total of 16 pixels. Since these are at various distances from the unknown pixel, closer pixels are given a higher weighting in the calculation as seen in figure 2.8. Bicubic produces noticeably sharper images than the previous two methods, and is perhaps the ideal combination of processing time and output quality. For this reason it is a standard in many image editing programs (including Adobe Photoshop), printer drivers and in-camera interpolation.

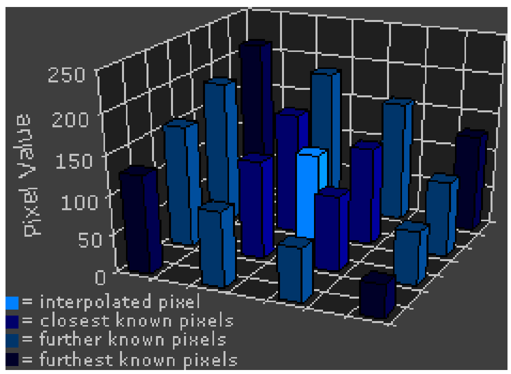


Figure 2.8: Bicubic Interpolation

Bicubic Interpolation uses the same principle as Bilinear Interpolation, except using a cubic function instead of a linear function to estimate pixels between known values (Figure 2.9). This form of interpolation has advantages and drawbacks over Bilinear Interpolation. First, calculating the cubic polynomial in a specific area of the image is more computationally expensive than simple linear fits and also requires a larger neighbourhood to calculate the curve. However, since Bicubic Interpolation utilizes a cubic curve, blurring is not as pronounced as in Bilinear Interpolation. That is because pixel value transitions can be more rapid on the curve. A linear function fits straight lines between known points, and a cubic function fits cubic splines. On the other hand, jaggies are more distinguished since the image isn't as blurred.

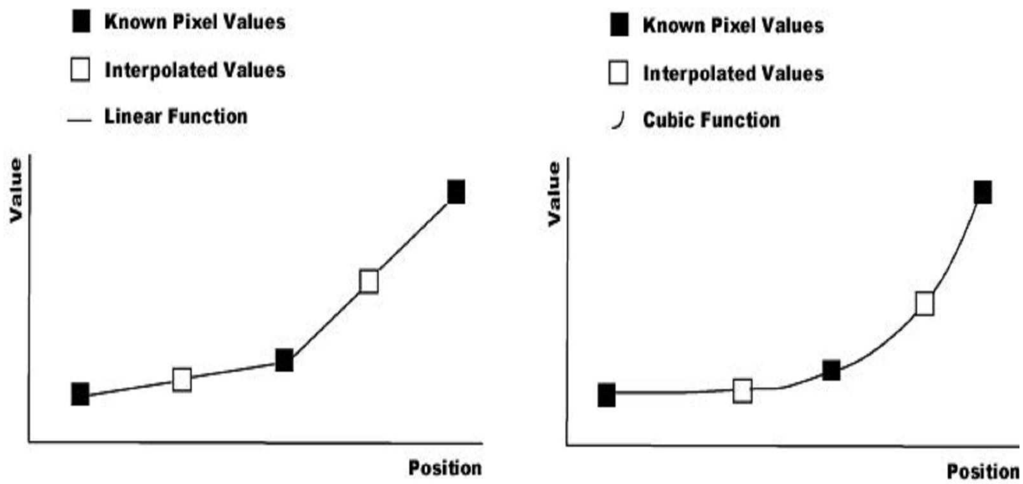


Figure 2.9: Linear vs BiCubic Interpolation

(a) Down Sampled Image (b) 4× Bilinear Interpolation

(c) 4× Bicubic Interpolation (d) Original image

Figure 2.10: Bilinear vs Bicubic Interpolation.

A comparison of Bilinear and Bicubic Interpolation is shown in Figure 2.10. In Figure 2.10, original image (2.10d) down sampled by a scale of four (2.10a) then magnified 4× using Bicubic Interpolation. Bicubic Interpolation produces magnifications, which are sharper and more jagged (the eyes for example) than Bilinear Interpolation.

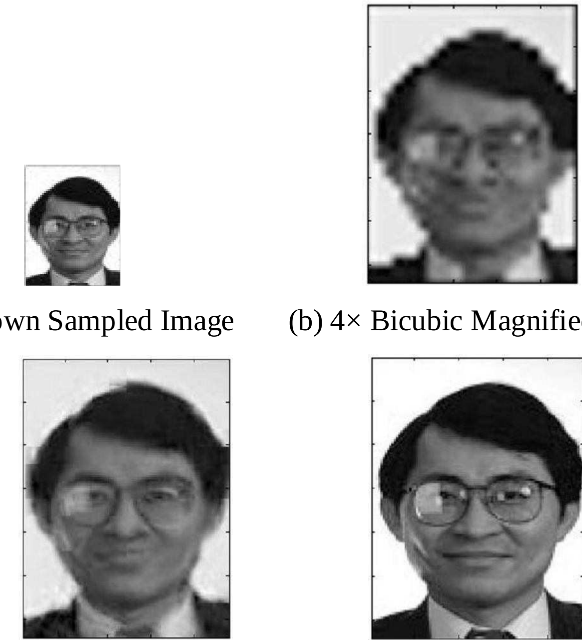
2.4 Filtering Methods

Another branch of magnification techniques uses filtering approaches to magnify low-resolution images. Filter-based methods use sampling theory to attempt to create perfect interpolations of images. A perfect reconstruction of sampled points is accomplished by convolving these points with a sinc function in the spatial domain. However, this is impossible due to the infinite extent of the sinc function. Filter-based methods overcome this by either truncating the sinc function, or even approximating a truncated sinc function with a cubic spline.

Due to these approximations, errors are introduced into the interpolation of the data, causing both blurring, jaggies, and ringing. Another limiting factor with filter based methods is the increased computational cost. Even with a truncated sinc function, the kernel can be quite large. For instance, the Lanczos [15] filter's kernel is 16×16, making the algorithm much more computationally expensive than Bicubic Interpolation.

2.5 Learning-Based Algorithms

Another general area of interpolation methods is learning-based algorithms. These algorithms pull information from training data in order to aid in the interpolation and create high resolution data [16, 17, 18, 19, 20]. For example, "Optimal face reconstruction using training" by Muresan and Parks [17] uses pairs of low-resolution and high-resolution images of faces. This training data is then used to aid in the interpolation of low resolution face images (Figure 2.11). Data in the high resolution image is not simply interpolated but can also be "created" by analyzing similar areas of face pairs. This facet of data creation is unique to learning-based algorithms. Unfortunately, these algorithms also have several drawbacks. First, analyzing training data can be computationally expensive. Second, the training data can require large amounts of storage memory. Also, specific training data is required for images. For instance, to make the most accurate face magnification, you need a large collection of face images to train with. Also, certain learning-based algorithms, such as Image Analogies [20], require extensive user interaction to identify regions in an image. These drawbacks hinder extensive use of learning-based algorithms for image magnification.



(a) Down Sampled Image (b) 4× Bicubic Magnified Image

(c) 4× Magnified Image (d) Original Image

Figure 2.11: Optimal Face Reconstruction Example.

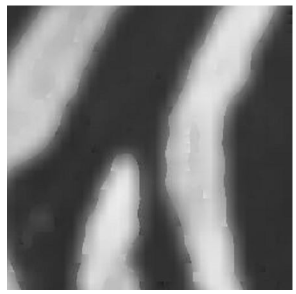
In Figure 2.11, original image (2.11d) is down sampled by a scale of four (2.11a) then magnified by 4× using Optimal Face Reconstruction [17]. Learning-Based algorithms can reconstruct areas such as the eyes far better than algorithms such as Bicubic Interpolation, but Learning-Based algorithms require specific training data in order to produce quality results.

2.6 Edge Directed Interpolation

The basic idea of Edge-Directed Interpolation techniques is to analyze edge information in the low-resolution image in order to aid in the interpolation step. Edge information can be used in a variety of ways, whether to interpolate in the edge direction or not to allow interpolations to cross edges [21, 22, 23, 24, 3].

A good example of an algorithm that uses edge information to enhance the magnification is “Edge-Directed Interpolation” by Allebach and Wong [3]. This algorithm uses a Laplacian-of-Gaussian filter to create a low-resolution edge map. This low-resolution map is then interpolated to a high-resolution edge map. In the rendering (interpolation) step, they modify Bilinear Interpolation in order to force interpolations to not cross edges in the edge map. The results of edge-directed interpolation show that edges remain sharp, as seen in Figure 2.12.


(a) Down Sampled Image (b) 8× Bilinear Interpolation



(c) 8× Edge-Directed Interpolation

Figure 2.12: Edge-Directed vs Bilinear Interpolation.

Original image down sampled by a scale of eight (a) then magnified by 8× using Edge-Directed Interpolation. Edge- Directed Interpolation produces strong edges but with ghosting. Edge-Directed interpolations unfortunately inherit all the downfalls of their constituent parts. For instance, Edge-Directed Interpolation by Allebach and Wong is limited to edges found by the Laplacian-of-Gaussian filter, which can detect edges in the middle of non-rapid edge transitions. This leads to “ghosting” in the final image. Also, enhancements to Bilinear Interpolation can only be seen where edges are detected, thus showing that a large portion of the image has the same problems as Bilinear Interpolation.

Our algorithm basically focuses on preserving the edges. The algorithms [21, 25] focus on edge preservation. Other edge-adaptive methods have been proposed by Jensen and Anastassiou [26], Li and Orchard [22], and Muresan and Parks [27, 28, 16, 18]. Currently, the methods presented in [22, 18] are the most widely known edge-adaptive methods. They can well enough avoid jagged edges, but a limitation is that they sometimes introduce highly visible artifacts into the magnified images, especially in areas with small size repetitive patterns. X. Yu, *et al.* [29] presented a method that computes a triangulation of the input image where the pixels in the input image are the vertices of the triangles in the triangulation. The input image at any arbitrary scale is reconstructed by rendering its triangulation. However, since the edges in the input image are approximated using piecewise linear segments, curved edges cannot be properly reconstructed especially when the scaling factor is a large number.

2.7 Properties of Interpolated Images

The first 8 are visual properties of the interpolated image; the last is a computational property of the interpolation method:

(1) Geometric Invariance: The interpolation method should preserve the geometry and relative sizes of objects in an image. That is, the subject matter should not change under interpolation.

(2) Contrast Invariance: The method should preserve the luminance values of objects in an image and the overall contrast of the image.

(3) Noise: The method should not add noise or other artifacts to the image, such as ringing artifacts.

(4) Edge Preservation: The method should preserve edges and boundaries; and sharpening them where possible.

(5) Aliasing: The method should not produce jagged or "staircase" edges.

(6) Texture Preservation: The method should not blur or smooth textured regions.

(7) Over-smoothing: The method should not produce undesirable piecewise constant or blocky regions.

(8) Application Awareness: The method should produce results appropriate to the type of image and order of resolution. For example, the interpolated results should appear realistic for photographic images, but for medical images, the results should have crisp edges and high contrast. If the interpolation is for general images, the method should be independent of the type of image.

(9) Sensitivity to Parameters: The method should not be too sensitive to internal parameters that may vary from image to image.

Chapter 3 – Problem Statement

An image zooming is of interest in many applications like scientific visualization, multimedia applications and image analysis tasks. A generic image zooming algorithm takes as an input a digital image and provides an output a picture of required size preserving as much as possible the information context of original image. Several good zooming techniques are now days available. For a large class of zooming techniques this is achieved b mean of some kind of interpolation: replication, bilinear and bicubic are the most popular choices and the routinely implemented in commercial digital image processing software.

Unfortunately, these techniques, while preserving while preserving the low frequencies content of the source image, are not equally able to enhance high frequencies in order to provide a picture whose visual sharpness matches the quality of the original image

In most of the cases zooming leads to distortion of image. This distortion can be of many ways such like jagging- blocks are formed due to replication of pixels, burring- It is unclearity of the image, ghosting- it is the distortion of the image. Current interpolation algorithms attempt to solve these artifacts in a number of ways. There are other factors have to be kept in mind while making an algorithm like speed (zooming should not take much time for enlargement), memory requirement (the algorithm should not take much memory space).

In this thesis, we have taken into account information about discontinuities or sharp intensity variations while doubling the input image.

The proposed technique is explained in the next section.

Chapter 4 – Proposed Algorithm

4.1 Algorithm Basic Stages

In this section we describes the basics of image zooming.

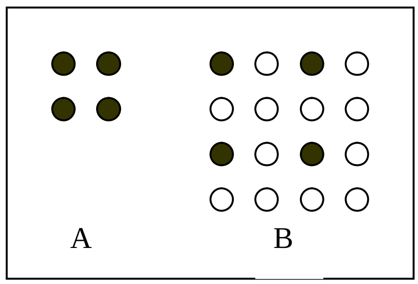


Figure 4.1: Basic Magnified Image

In Figure: 4.1 the image A is magnified 2 times to image B. The blank circles are the unknown pixel values and we have to find them. The above image A (2x2) is magnified to B (4x4). But in reality the original image (nxn) is magnified to (2n-1)x(2n-1).

Our algorithm includes 3 stages. All three stages have been discussed below.

4.1.1 Stage I

First step is to expand the original (n x n) image to (2n-1) x (2n-1). We can make it to (2n x 2n) by extending the initial image borders. The figure 4.2 given below describes the above theory. In the figure the X (5 x 5) is the original image and Y (9 x 9) is the magnified image. X(i,j) is the pixel in the original image where i is the ith row and j is the jth column. In the same way in Y(m,n), m is the mth row and n is the nth.

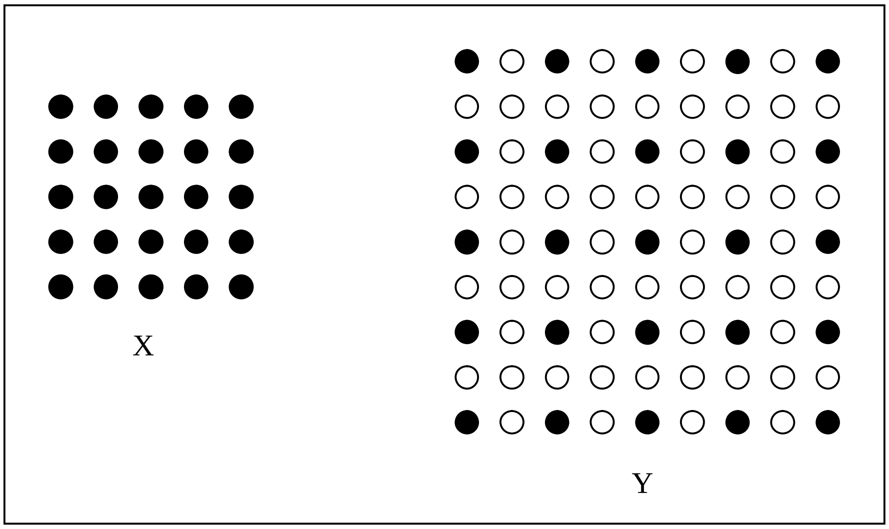


Figure 4.2: Zooming Phase

The black dots represent the pixels in the original image (X). And those pixels are as it is copied in the image Y. These pixels can be mapped as:

$$X(i , j) = Y(2i-1 , 2j-1)$$

The white dots represent the unknown pixels whose values we have to find. When an image is magnified 2 times, total number of pixels will be four times. Hence, we have now 1 known and 3 unknown pixel and we have to find 3 other with our algorithm.

4.1.2 Stage II

In this stage we have the centre pixel X. As we can see in the Figure 4.3, the centre pixel X has to be finding out.

The centre pixel X is deduced with the help of the algorithm which is described in the section 4.2.

The H_1 and H_2 are considered in case of the horizontal boundaries. V_1 and V_2 are considered in case of vertical boundaries. In all other case these four pixels are left vacant. The will be filled in the later stages.

H_1 is considered in case of upper boundary of image then,

$$H_1= (A+B)/2$$

H_2 is considered in case of lower boundary of image then,

$$H_2= (C+D)/2$$

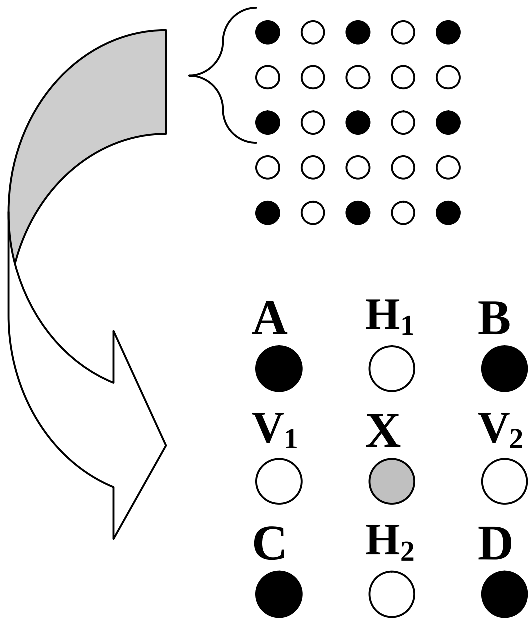


Figure 4.3: Stage II Zooming

V_1 is considered in case of left boundary of image then,

$$V_1 = (A+C)/2$$

V_2 is considered in case of right boundary of image then,

$$V_2 = (B+D)/2$$

Now, we have to repeat the above procedure for the complete image until all the unknown pixels of the boundaries and the centre pixels are found. After this we have to move to stage III.

Eventually we also have to plot the values of the boundaries of the image.

4.1.3 Stage III

In this stage, we have to start from the beginning of the image and find the left over pixels as shown in figure 4.4.

In the figure 4.4 (I) and (II), A & B are the pixels from original image. And X_1 and X_2 are the pixels derived from stage I.

All the pixels {A, B, X_1, X_2} are considered as in Stage I and computed in the same way.

Finally the value of M is computed and put in the place of centre pixel. The (I) and (II) are computed simultaneously for the complete image. The M is computed using algorithm of section 4.2.

After all these steps we finally get the zoomed image.

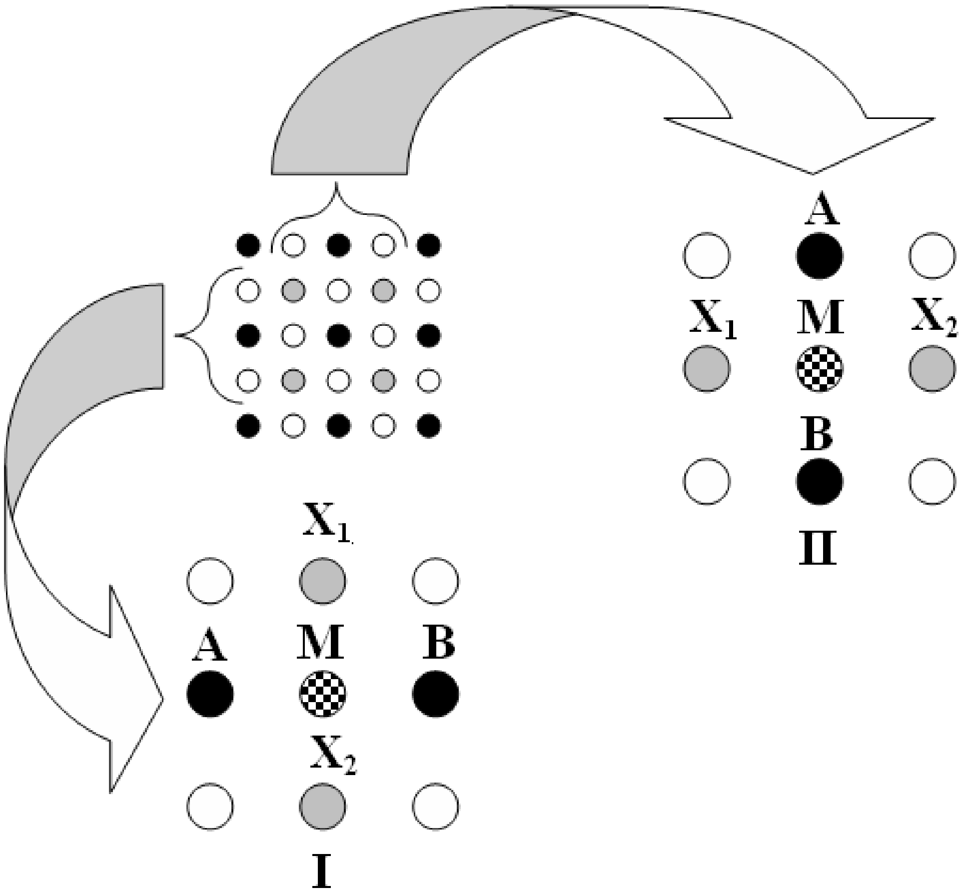


Figure 4.4: Stage III Zooming

4.2 Algorithm Description

In this section, main algorithm to find the centre pixel is explained. As we can see in figure 4.5, the main centre point M has to be find out.

This is an adaptive algorithm where M is dependent on P, Q, R and S pixels. We can call it in another way that every pixel bit depth is 8 and we can use it as weight.

Bit_Depth = 8

$W_t = (2^{Bit\_Depth} - 1) / 2$

Pixel_Weight = | Pixel_Value - W_t |

Our algorithm is following the basic weighted interpolation. Each value is not averaged but weights have been applied to each pixel. As we know WHITE and BLACK have equal priority, that's why $|W_t - 0|$ & $|W_t - 255|$ has equal weight. Hence we subtract each value from W_t to get the weight of the pixel. We have taken W_t because it is the centre point of 0 & 2^{Bit_Depth} and each pixel can only have value in between it.

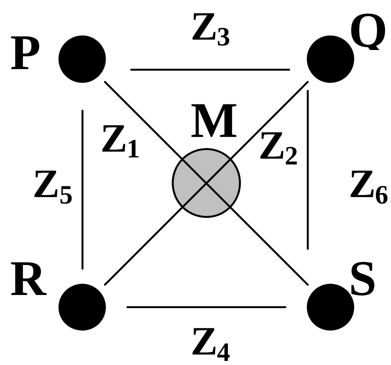


Figure 4.5: Finding Centre Point

The final value will be based on what are the intensities of other pixels.

First, we have to find Z_1, Z_2, Z_3, Z_4, Z_5, and Z_6 which is as follows:

Z_1=| P – S |……………………………(4.1)

Z_2=| Q – R |……………………………(4.2)

Z_3=| P – Q |……………………………(4.3)

Z_4=| R – S |……………………………(4.4)

Z_5=| P – R |……………………………(4.5)

Z_6=| Q – S |……………………………(4.6)

Now, we have to find the edge (if any) using these four known pixels (Figure 4.5). For that we have find the combination of two pixels where the intensity difference is the minimum. Then only we can detect the edge.

Now, we have many conditions from which we deduce the point M. If any one condition satisfies we need not to go further. But the conditions have to be read in order.

Condition I:

"All pixels are uniform".

If all pixels are having nearly same intensity, then condition I is applied. There is a factor K, which is the distance of each pixel from centre. After doing the series of tests, we have found that K (threshold) gives the best result at 10. As it can be seen in Figure 4.6

$$X = (P + Q + R + S)/4$$

$$ZZ = (|X - P| + |X - Q| + |X - R| + |X - S|)/4$$

If ZZ < K Then,

$$M = (P + Q + R + S)/4$$

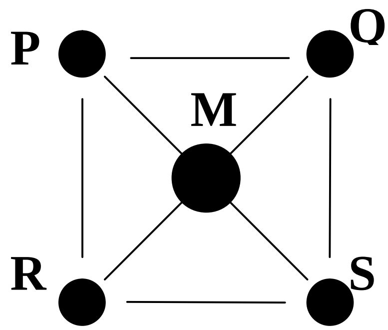


Figure 4.6: Uniform Condition

Here, ZZ is the average distance of each pixel from centre.

And if the average distance is less than K then all pixels are uniform.

<u>Condition II:</u>

"If the intensity of both the diagonal edges is same and lowest in all equations (4.1-4.6)"

If intensity difference at the diagonals is same then we have to take the average of both the diagonals.

If (Z_1 == 0 AND Z_2 == 0) OR <<Z_3, Z_4, Z_5, Z_6

$$M = \left(P \times |W_t - P| + Q \times |W_t - Q|\right) / \left(|W_t - P| + |W_t - Q|\right)$$

<u>Condition III:</u>

"If edge is at diagonal P-S"

If the intensity at the diagonals P-S in minimum then all other (4.1 - 4.6), then the edge is at P-S. Then also we have two conditions. The edge could be on either Q or R side

If Z_1 == 0 OR <<Z_2, Z_3, Z_4, Z_5, Z_6

To check this:

$$Q_1 = |Q - P| + |Q - S|$$

$$R_1 = |R - P| + |R - S|$$

III(A): If $Q_1<R_1$ “Edge on side Q”

Shown in Figure 4.7

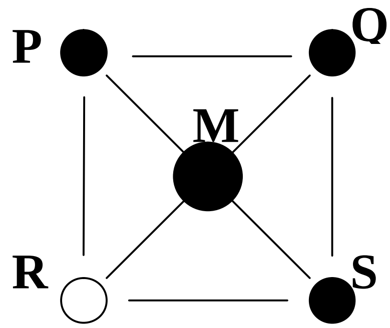


Figure 4.7: Edge on PS-Q

$$M = \left(P \times |W_t - P| + S \times |W_t - S| + Q \times |W_t - Q|\right) / \left(|W_t - P| + |W_t - S| + |W_t - Q|\right)$$

III(B): If $R_1<Q_1$ “Edge on side R”

Shown in Figure 4.8

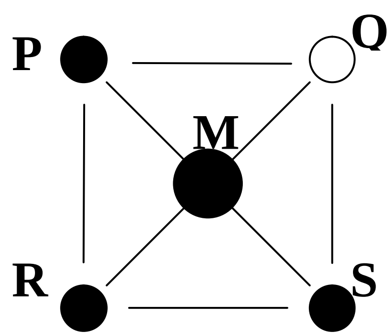


Figure 4.8: Edge on PS-R

$$M = \left(P \times |W_t - P| + S \times |W_t - S| + R \times |W_t - R|\right) / \left(|W_t - P| + |W_t - S| + |W_t - R|\right)$$

<u>*Condition IV:*</u>

“If edge is at diagonal Q-R”

If the intensity at the diagonal Q-R is minimum then all other (4.1 - 4.6), then the edge is at Q-R. Then also we have two conditions. The edge could be on either P or S side

If $Z_2 == 0$ OR $<<Z_1, Z_3, Z_4, Z_5, Z_6$

To check this:

$$P_1 = |P - Q| + |P - R|$$

$$S_1 = |S - Q| + |S - R|$$

IV(A): If $P_1 < S_1$ "Edge on side P"

Shown in Figure 4.9

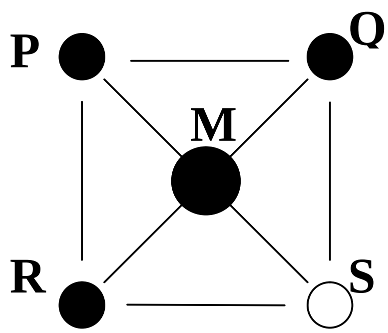


Figure 4.9: Edge on QR-P

$$M = (Q \times |W_t - Q| + R \times |W_t - R| + P \times |W_t - P|)/(|W_t - Q| + |W_t - R| + |W_t - P|)$$

IV(B): If $S_1 < P_1$ "Edge on side S"

Shown in Figure 4.10

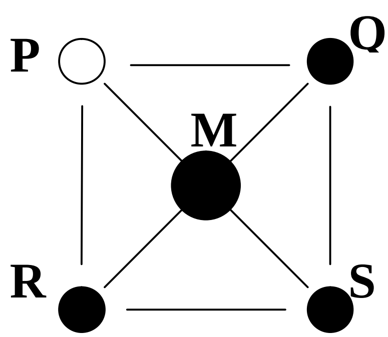


Figure 4.10: Edge on QR-S

$$M = (Q \times |W_t - Q| + R \times |W_t - R| + S \times |W_t - S|)/(|W_t - Q| + |W_t - R| + |W_t - S|)$$

<u>Condition V:</u>

"If the intensity of both the horizontal edges are same and lowest in all (4.1 – 4.6)"

If intensity difference at the vertical is same then we have to take the average of both the horizontal edges.

If (Z_3 == 0 AND Z_4 == 0) OR <<Z_1, Z_2, Z_5, Z_6

$$M = \left(P \times |W_t - P| + R \times |W_t - R|\right) / \left(|W_t - P| + |W_t - R|\right)$$

<u>*Condition VI:*</u>

"If edge is on side P-Q"

If the intensity of P-Q edge is minimum than all other (4.1 – 4.6), then edge is at P-Q.

If Z_3 == 0 OR <<Z_1, Z_2, Z_4, Z_5, Z_6

But still we have to consider some part of R-S. Hence,

$$D_1 = (R + S)/2$$

Here, D_1 is the average of the opposite side pixels.

Shown in Figure 4.11

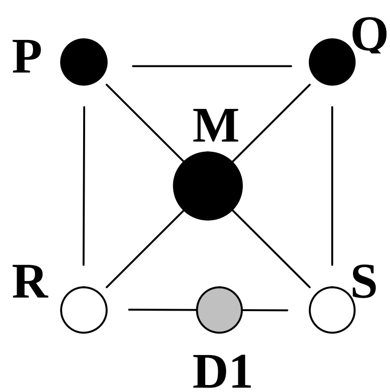


Figure 4.11: Edge on PQ

$$M = \left(P \times |W_t - P| + Q \times |W_t - Q| + D1 \times |W_t - D1|\right) / \left(|W_t - P| + |W_t - Q| + |W_t - D1|\right)$$

<u>*Condition VII:*</u>

"If edge is on side R-S"

If the intensity of R-S edge is minimum than all other (4.1 – 4.6), then edge is at R-S.

If Z_4 == 0 OR <<Z_1, Z_2, Z_3, Z_5, Z_6

But still we have to consider some part of P-Q. Hence,

$$D_1 = (P + Q)/2$$

Here, D_1 is the average of the opposite side pixels.

Shown in Figure 4.12

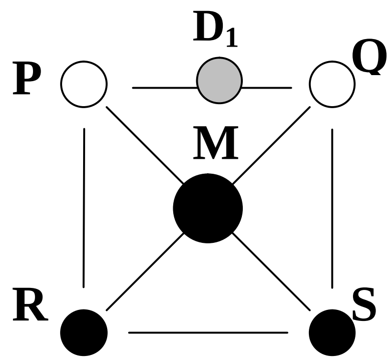


Figure 4.12: Edge on RS

$$M = \left(R \times |W_t - R| + S \times |W_t - S| + D1 \times |W_t - D1|\right)/\left(|W_t - R| + |W_t - S| + |W_t - D1|\right)$$

Condition VIII:

"If the intensity of both the vertical edges are same and lowest in all (4.1 – 4.6)"

If intensity difference at the vertical is same then we have to take the average of both the horizontal edges.

If (Z_5 == 0 AND Z_6 == 0) OR <<Z_1, Z_2, Z_3, Z_4

$$M = \left(P \times |W_t - P| + Q \times |W_t - Q|\right)/\left(|W_t - P| + |W_t - Q|\right)$$

Condition IX:

"If edge is on side P-R"

If the intensity of P-R edge is minimum than all other (4.1 – 4.6), then edge is at P-R.

If Z_5 == 0 OR <<Z_1, Z_2, Z_3, Z_4, Z_6

But still we have to consider some part of Q-S. Hence,

$$D_1 = (Q + S)/2$$

Here, D_1 is the average of the opposite side pixels.

Shown in Figure 4.13

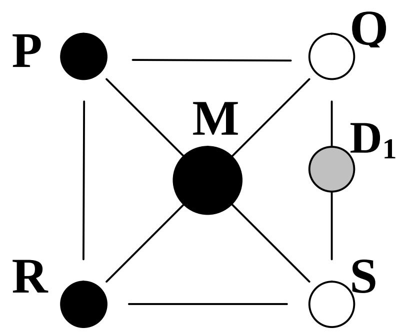


Figure 4.13: Edge on PQ

$$M = \left(P \times |W_t - P| + R \times |W_t - R| + D1 \times |W_t - D1|\right) / \left(|W_t - P| + |W_t - R| + |W_t - D1|\right)$$

Condition X:

"If edge is on side Q-S"

If the intensity of Q-S edge is minimum than all other (4.1 – 4.6), then edge is at Q-S.

If $Z_6 == 0$ OR $<< Z_1, Z_2, Z_3, Z_4, Z_5$

But still we have to consider some part of P-R. Hence,

$$D_1 = (P + R)/2$$

Here, D_1 is the average of the opposite side pixels.

Shown in Figure 4.14

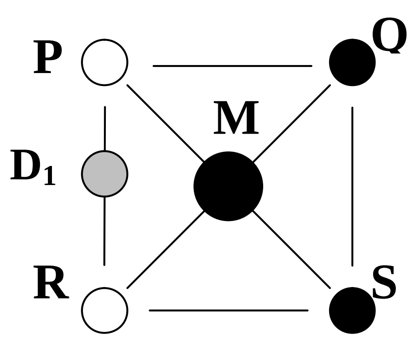


Figure 4.14: Edge on QS

$$M = \left(Q \times |W_t - Q| + S \times |W_t - S| + D1 \times |W_t - D1|\right) / \left(|W_t - Q| + |W_t - S| + |W_t - D1|\right)$$

By applying, all these conditions accordingly we can find the centre pixel M.

4.3 For Gray Scale Images

As we know that grey scale has only one layer shown in figure 4.15. So to implement the algorithm we just have to apply the algorithm as it is.

Every pixel on the grey scale image has the variable intensities from 0-255. We just have to take 4 pixels at a time and apply the algorithm as given above.

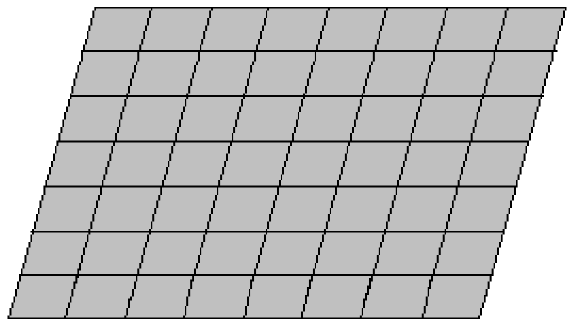


Figure 4.15: (m x n) Grey Scale Image

4.4 Color Images

Our algorithm can be easily implemented on the color images. A color image has 3 layers of Red, Green, and Blue shown in figure 4.16. The algorithm has to be implemented separately on different layers then combine the layers together later.

However, the computation increases as we move to the color images.

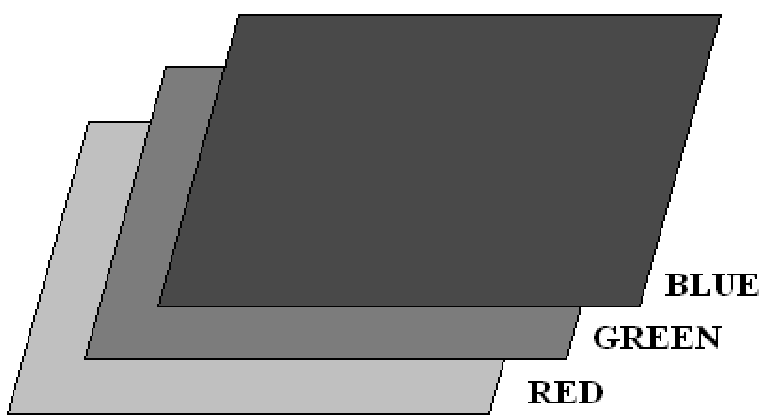


Figure 4.16: Layers of Color Image

4.4 Drawbacks

There are few drawbacks of this algorithm. As we know that it is an adaptive algorithm which mainly focuses on edges.

Firstly, it is not so good in preserving the small information because it mainly focuses on the edges. Small curves cannot be detected properly.

Secondly, the computation time is a bit larger on the color images also we have ten conditions that have to be compared.

Chapter 5 – Results

5.1 Visual Comparison

The first qualitative analysis of adaptive method is a series of image comparisons that are presented to the reader. It is clear that visually the output of our algorithm is better than the other existing techniques if the input image has sharp edges. The following pages include these image comparisons, showing various types of images and labeling their corresponding interpolation algorithm. All the original images are given in Appendix A.

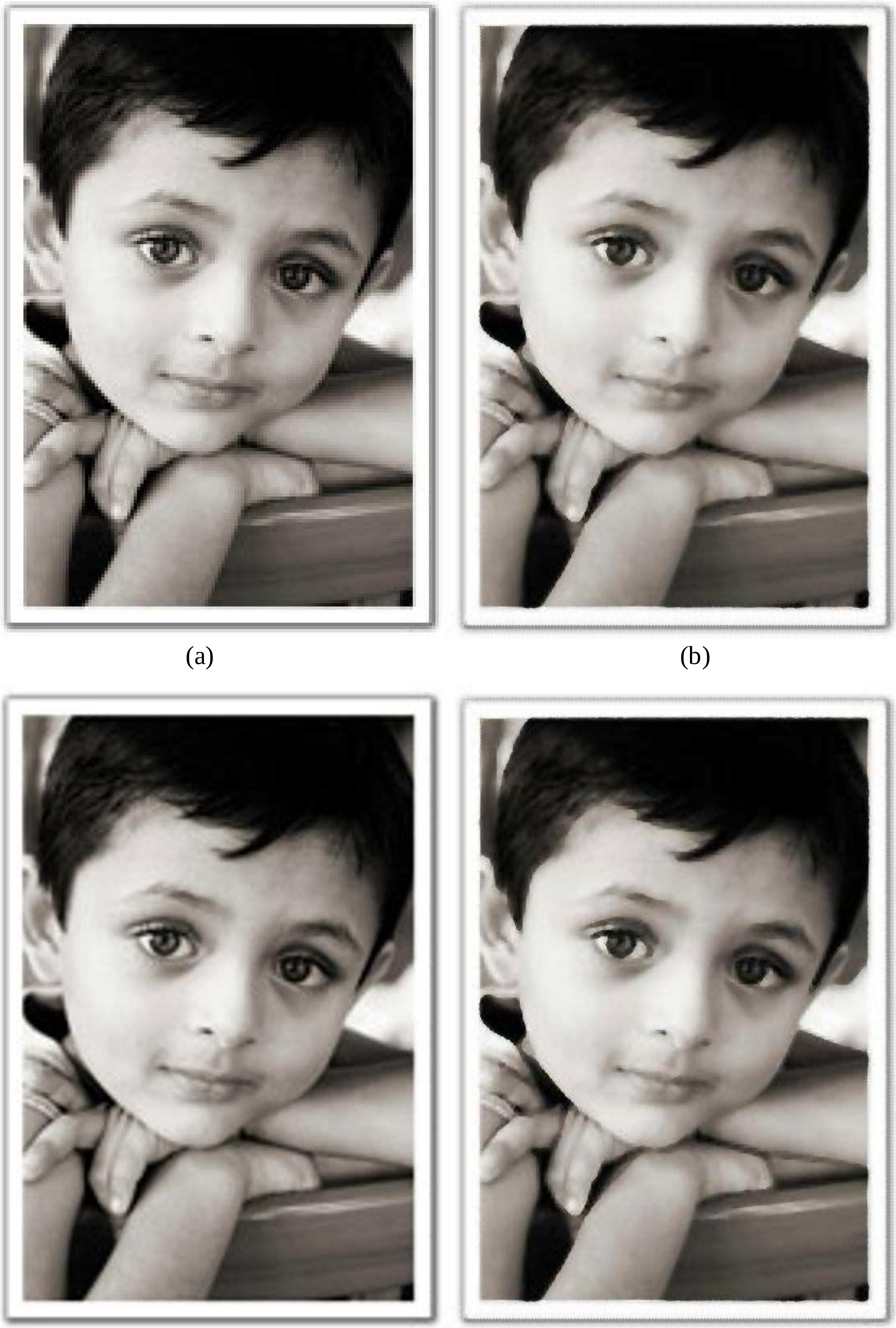
Figure 5.1: Boy Image Sub Sampled by 4

Figure 5.2: Boy Image magnified 4X with Pixel Replication

Figure 5.3: Boy Image magnified 4X with Bilinear Interpolation

Figure 5.4: Boy Image magnified 4X with Bicubic Interpolation

Figure 5.5: Boy Image magnified 4X with Proposed Algorithm



(a) (b)

(c) (d)

Figure 5.6: Boy Image comparison. (a) Pixel Replication (b) Bilinear Interpolation (c) Bicubic Interpolation (d) Proposed Algorithm

Figure 5.7: Cartoon Image Sub Sampled by 4

Figure 5.8: Cartoon Image magnified 4X with Pixel Replication

Figure 5.9: Cartoon Image magnified 4X with Bilinear Interpolation

Figure 5.10: Cartoon Image magnified 4X with Bicubic Interpolation

Figure 5.11: Cartoon Image magnified 4X with Proposed Algorithm

(a)

(b)

(c)

(d)

Figure 5.12: Cartoon Image comparison. (a) Pixel Replication (b) Bilinear Interpolation (c) Bicubic Interpolation (d) Proposed Algorithm

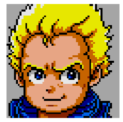


Figure 5.13: Face Image Sub Sampled by 4

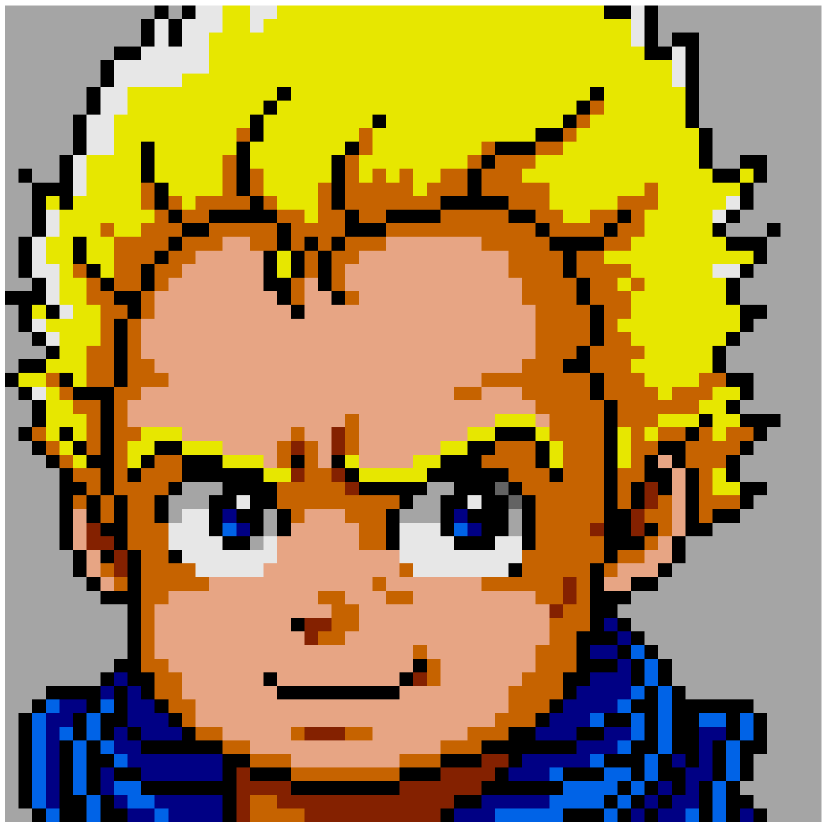
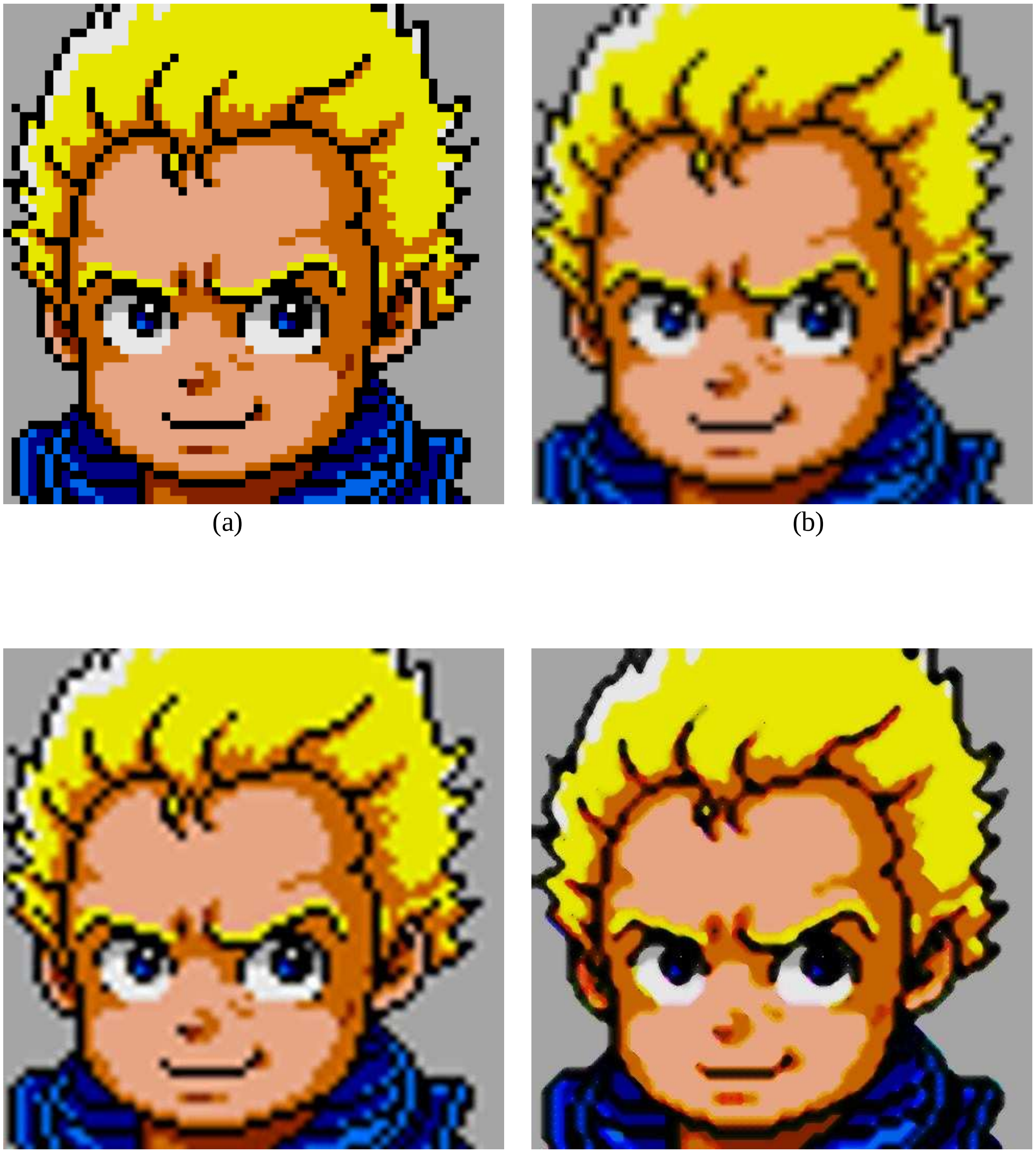


Figure 5.14: Face Image magnified 8X with Pixel Replication


Figure 5.15: Face Image magnified 8X with Bilinear Interpolation


Figure 5.16: Face Image magnified 8X with Bicubic Interpolation

Figure 5.17: Face Image magnified 8X with Proposed Algorithm



(a) (b) (c) (d)

Figure 5.18: Face Image comparison. (a) Pixel Replication (b) Bilinear Interpolation (c) Bicubic Interpolation (d) Proposed Algorithm

Figure 5.19: Girl Image Sub Sampled by 4

Figure 5.20: Girl Image magnified 4X with Pixel Replication

Figure 5.21: Girl Image magnified 4X with Bilinear Interpolation

Figure 5.22: Girl Image magnified 4X with Bicubic Interpolation

Figure 5.23: Girl Image magnified 4X with Proposed Algorithm

(a)

(b)

(c)

(d)

Figure 5.24: Girl Image comparison. (a) Pixel Replication (b) Bilinear Interpolation (c) Bicubic Interpolation (d) Proposed Algorithm

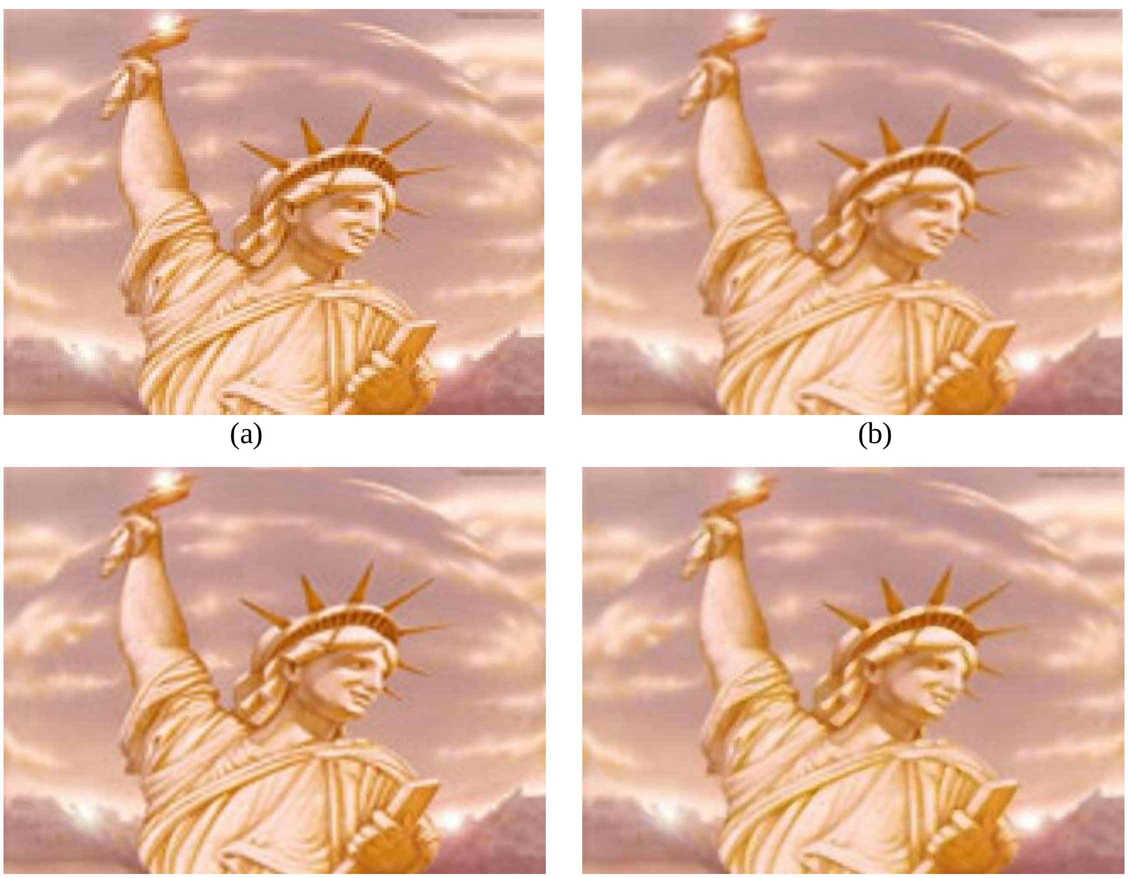
Figure 5.25: Statue Image Sub Sampled by 4

Figure 5.26: Statue Image magnified 4X with Pixel Replication

Figure 5.27: Statue Image magnified 4X with Bilinear Interpolation

Figure 5.28: Statue Image magnified 4X with Bicubic Interpolation

Figure 5.29: Statue Image magnified 4X with Proposed Algorithm



(a) (b)

(c) (d)

Figure 5.30: Stauue Image comparison. (a) Pixel Replication (b) Bilinear Interpolation (c) Bicubic Interpolation (d) Proposed Algorithm

Figure 5.31: Glass Image Sub Sampled by 4

Figure 5.32: Glass Image magnified 4X with Pixel Replication

Figure 5.33: Glass Image magnified 4X with Bilinear Interpolation

Figure 5.34: Glass Image magnified 4X with Bicubic Interpolation

Figure 5.35: Glass Image magnified 4X with Proposed Algorithm

(a)

(b)

(c)

(d)

Figure 5.36: Glass Image comparison. (a) Pixel Replication (b) Bilinear Interpolation (c) Bicubic Interpolation (d) Proposed Algorithm

5.2 Quantitative Analysis

Five different quantitative analysis measurements were used on five real-world images. In order to obtain these measures, an image was first down sampled by a factor of four. This lower-resolution image was then magnified by a factor of four using a variety of magnification techniques, and then compared with the original image. All the measurements reflect the accuracy of the magnification. These measurements are Mean Squared Error (MSE), Mean Absolute Error (MAE), and Cross-correlation Coefficient (CCC), Peak Signal to Noise Ratio (PSNR).Results of the quantitative analysis is shown in Table 5.1.

The manner in which the images are initially down sampled is also worth noting since we are down sampling the image, magnifying this lower resolution image and comparing it to the original image. Depending on the way the original image is mapped to the lower resolution image, the comparison of the magnified and original image can be inaccurate, since the image can experience sub-pixel shifts. We here have used Matlab to down sample the image.

Here all the magnified images are compared with the original images given in the Appendix A.

Image	Type	Pixel Rep.	Bilinear	Bicubic	Proposed
BOY	MSE	155.8897	296.7410	295.3281	310.1163
	MAE	6.3133	7.5121	7.3220	7.5554
	CCC	0.9893	0.9798	0.9798	0.9680
	PSNR	417.1376	219.1464	220.1953	210.8054
CARTOON	MSE	317.5174	317.7774	311.6193	321.2892
	MAE	8.5258	8.4788	8.5362	8.6030
	CCC	0.9554	0.9553	0.9562	0.9485
	PSNR	204.8464	204.7609	208.7646	202.4104
GLASS	MSE	259.5827	240.6331	230.2378	255.8283
	MAE	7.5494	7.3792	7.2965	7.5056
	CCC	0.9661	0.9687	0.9700	0.9668
	PSNR	250.6031	269.7026	282.5597	259.9710
GIRL	MSE	229.9045	444.4763	441.8590	452.8693
	MAE	8.7244	10.7595	10.4931	10.8190
	CCC	0.9868	0.9747	0.9748	0.9701
	PSNR	282.9294	146.4307	147.3097	135.1001
STATUE	MSE	149.2593	145.6853	138.6730	140.2800
	MAE	7.2680	7.1438	7.0160	7.1941
	CCC	0.9291	0.9356	0.9968	0.9351
	PSNR	509.7864	589.4892	589.4892	568.6020

Table 5.1: Results of performing the quantitative Analysis

5.3 Accuracy Measurement

The quantitative measurements were gathered by comparing the magnified image with the original image using several metrics which are Mean Squared Error, Mean Absolute Error, Cross-correlation Coefficient and Peak Signal to Noise Ratio. These metrics are defined as

Mean Squared Error = $\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left(\hat{I}(x,y)-I(x,y)\right)^2}{MN}$ ……........................(5.1)

Mean Squared Error = $\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left|\hat{I}(x,y)-I(x,y)\right|}{MN}$ …....................……(5.2)

Cross-Correlation Coefficient =

$$\left|\frac{\left(\sum_{x=1}^{M}\sum_{y=1}^{N}\hat{I}(x,y)I(x,y)-nab\right)}{\left(\left(\sum_{x=1}^{M}\sum_{y=1}^{N}\hat{I}^2(x,y)-na^2\right)\left(\sum_{x=1}^{M}\sum_{y=1}^{N}I^2(x,y)-nab^2\right)\right)^{\frac{1}{2}}}\right| \text{...................(5.3)}$$

Peak Signal to Noise Ratio = $\frac{MAX\left(\hat{I}.\hat{}2\right)}{SUM\left(\left(\hat{I}-I\right).\hat{}2\right)}$ ………………….............(5.4)

Where $\hat{I}$ is the magnified image, I is the original image, n is the total number of pixels and a, b are the corresponding average pixel value in each image.

In equation 5.4, MAX is a function that finds the maximum value of array, SUM is the function that finds the sum of array and "**.**^2" – it multiplies the array element by element by itself.

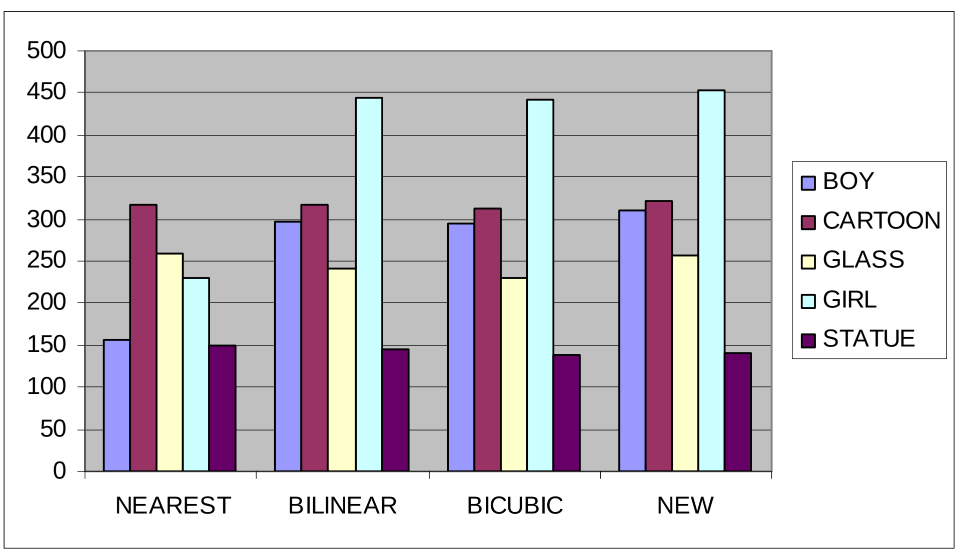


Figure 5.37: Mean Squared Error results

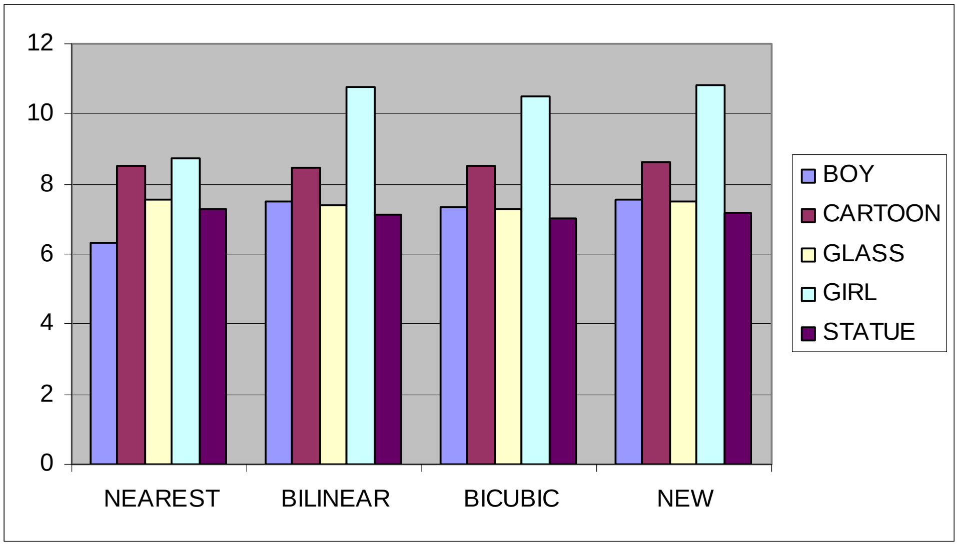


Figure 5.38: Mean Absolute Error results

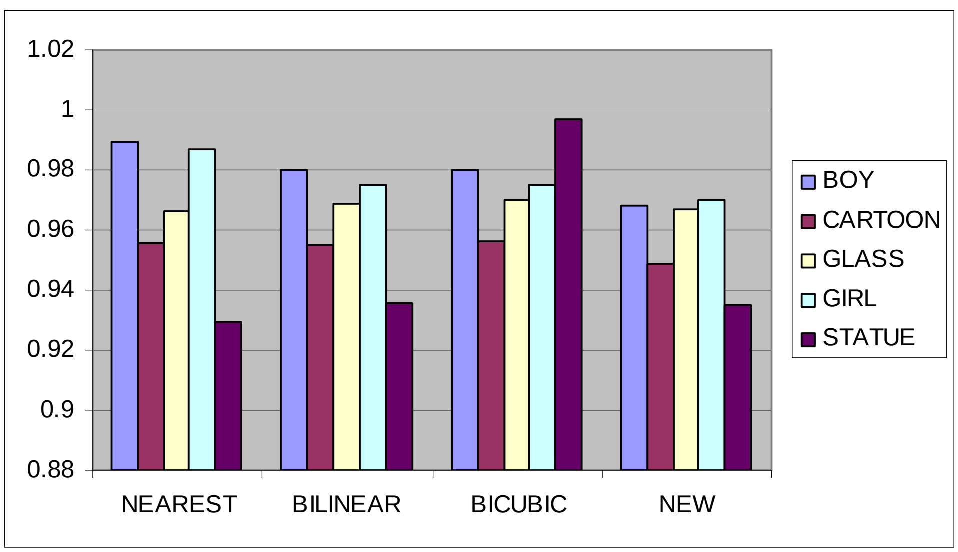


Figure 5.39: Cross-Correlation Coefficient results

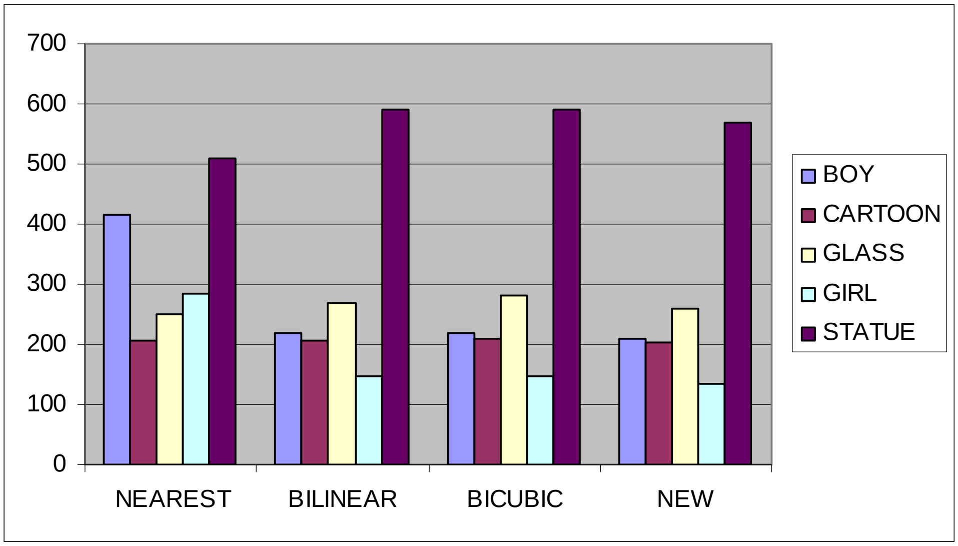


Figure 5.40: Peak Signal to Noise Ratio Results

It should also be noted that lower values of mean squared and absolute error indicate higher accuracy while values closer to one indicate higher accuracy in cross-correlation coefficient values. And higher the value less noise in peak signal to noise ratio.

When these results are analyzed visually, we can conclude that the results are better than others. And it becomes clear that the accuracy measurements alone are not sufficient to determine image quality.

Visual inspection also clearly shows obvious artifacts such as blurring and jaggies in images produced with Bilinear Interpolation but less in case of our method. Therefore, accuracy measurements are not enough to indicate image quality.

As we know that in accuracy measurement our method is not giving the same results as other methods but visually it is better.

Error can be introduced in our method if there is variable intensity change. But it gives good result at the sharp intensity change (edges).

This leads to the conclusion that our adaptive zooming method provides better visual results.

Chapter 6 – Conclusion

6.1 Summary

Digital images are becoming a necessary element of all types of media in today's digital age. Images are often magnified to a greater resolution for a number of reasons, including viewing, printing, or editing. This thesis has proposed a magnification technique that produces higher quality image magnifications than previous standard techniques. This algorithm is called *An Adaptive Zooming algorithm*.

The Adaptive Zooming algorithm attempts to remove many of the artifacts that are predominant in standard magnification techniques.

The proposed algorithm basically focuses on edges present in the image. It works on the sharp intensity changes and successfully removes artifacts at the edges.

The Adaptive Zooming algorithm is able to produce real-world magnifications that visual studies show that magnification is better than existing methods. It produces very good result on the animated images.

6.2 Thesis Contribution

This thesis has made the following original contributions:

1. A comprehensive survey of the current literature on the topic of adaptive image magnification was performed.
2. The various traditional algorithms were studied and implemented. These algorithms were implemented on Matlab 7.0.
3. The performance of each algorithm was analyzed by determining the various mathematical techniques. A summary of the analysis is presented.
4. We have developed an algorithm which minimizes the zooming artifacts.

5. Our algorithm is compared visually and quantitatively with the traditional methods. And the results in case of visual comparison are much better and quantitatively it is up to the mark.

6.3 Future Work of Research

The future work would involve the improvement of the proposed algorithm for the images containing continuous intensity variation.

The weight factors used in the proposed algorithm can be combined with other existing algorithms and the results can be improved visually as well as quantitatively.

REFERENCES

[1] R. C. Gonzalez, R. E. Woods, “Digital Image Processing”, Prentice Hall.

[2] Chandan Singh, “Interpolation Methods Image Zooming”, *Proceeding of the National Conference FACM Thapar University*, pp-29-34, March 2005.

[3] J. Allebach and P.W. Wong, “Edge-directed interpolation”, *IEEE International Conference on Image Processing*, vol 3:707–710, 1996.

[4] T. M. Lehmann, C. Gonner, and K. Spitzer, “Survey: Interpolation methods in medical image processing”, *IEEE Trans. Medical Imaging*, vol. 18, pp. 1049–1075, November 1999.

[5] A. Gotchev, J. Vesma, T. Saramaki, and K. Egiazarian, “Digital image resampling by modified B-spline functions”, *IEEE Nordic Signal Processing Symposium*, pp. 259 – 262, June 2000.

[6] H. Chen and G. E. Ford, “An FIR interpolation filter design method based on properties of human vision”, *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 581–585, November 1994.

[7] S. E. Reichenbach and F. Geng, “Two-dimensional cubic convolution”, *IEEE Trans. Image Processing*, vol. 12, pp. 857–865, August 2003.

[8] H. Jiang and C. Moloney, “A new direction adaptive scheme for image interpolation”, *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 369–372, 2002.

[9] B. S. Morse and D. Schwartzwald, “Image magnification using level-set reconstruction”, *Proc. IEEE Conf. Computer Vision Pattern Recognition*, vol. 3, pp. 333–340, 2001.

[10] H. Aly and E. Dubois, “Regularized image up-sampling using a new observation model and the level set method”, *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 665–668, September 2003.

[11] Y. Takahashi and A. Taguchi, “An enlargement method of digital images with the prediction of high-frequency components”, *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 4, pp. 3700–3703, 2002.

[12] X. Lu, P. S. Hong, and M. J. T. Smith, “An efficient directional image interpolation method”, *Proc. IEEE Int. Conf. Acoustics Speech Signal Processing*, vol. 3, pp. 97–100, 2003.

[13] R. R. Schultz and R. L. Stevenson, “A Bayesian approach to image expansion for improved definition”, *IEEE Trans. Image Processing*, vol. 3, pp. 233–242, May 1994.

[14] C. B. Atkins, C. A. Bouman, and J. P. Allebach, “Optimal image scaling using pixel classification”, *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 864–867, 2001.

[15] C. Duchon. Lanczos, “Filtering in one and two dimensions”, *Journal of Applied Meteorology*, vol. 18, pp. 1016–1022, 1979.

[16] D. D. Muresan and T. W. Parks, “Optimal recovery approach to image interpolation”, Proceedings of the *IEEE International Conference on Image Processing*, vol. 3, pp. 848–851, 2001.

[17] D. D. Muresan and T. W. Parks, “Optimal face reconstruction using training”, *IEEE International Conference on Image Processing*, vol. 3, pp. 373–376, 2002.

[18] D. D. Muresan and T.W. Parks, “Adaptively quadratic (AQua) image interpolation”, *IEEE Transactions on Image Processing*, vol. 13(5), pp. 690–698, 2004.

[19] J. Sun and N. Zheng and H. Tao and H. Shum, "Image hallucination with primal sketch priors", Proceedings of the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 729–736, 2003.

[20] A. Hertzmann and C.E. Jacobs and N. Oliver and B. Curless and D.H. Salesin, "Image Analogies", Proceedings of the *28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 327–340, 2001.

[21] S. Battiato and G. Gallo and F. Stanco, "A new edge-adaptive zooming algorithm for digital images", Proceedings of *Signal Processing and Communications SPC*, pp. 144–149, 2000.

[22] X. Li and M. T. Orchard, "New edge-directed interpolation", *IEEE Transactions on Image Processing*, vol. 10(10), pp. 1521–1527, 2001.

[23] Q. Wang and R. Ward, "A New edge-directed image expansion scheme", *IEEE International Conference on Image Processing*, vol. 3, pp. 899–902, 2001.

[24] K. Jensen and D. Anastassiou. "Subpixel edge localization and the interpolation of still images". *IEEE Transactions on Image Processing*, 4(3):285–295, 1995.

[25] Muhammad Sajjad, Naveed Khattak, and Noman Jafri, "Image Magnification Using Adaptive Interpolation by Pixel Level Data-Dependent Geometrical Shapes", *International Journal of Computer Science and Engineering*, vol. 1(2), pp. 118-127.

[26] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images", *IEEE Transactions on Image Processing*, vol. 4(3), pp.285–295, 1995.

[27] D. D. Muresan and T. W. Parks, "New image interpolation techniques", Proceedings of *IEEE 2000 Western New York Image Processing Workshop*, 2000.

[28] D. D. Muresan and T. W. Parks, “Adaptive, optimal-recovery image interpolation”, Proceedings of *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1949–1952, 2001.

[29] X. Yu, B. S. Morse, and T. W. Sederberg, “Image reconstruction using data-dependent triangulation”, *IEEE Computer Graphics and Applications*, vol. 21(3), pp. 62–68, 2001.

Communicated Paper

Mayank Kumar, Singara Singh and Ajay Kumar, "**Image Zooming Techniques Comparison",** Submitted in *National Conference on Emerging Principles and Practices of Computer Science and Information Technology (EPPCSIT 2009)*, 4-5 September, Ludhiana.

APPENDIX A

A.1 Boy

Figure A.1: Boy Original Image

A.2 Cartoon

Figure A.2: Cartoon Original Image

A.3 Girl

Figure A.3: Girl Original Image

A.4 Statue

Figure A.4: Statue Original Image

A.5 Glass

Figure A.5: Glass Original Image