Evaluation of Different Image Interpolation Algorithms

Ankit Prajapati  
Computer Engg.  
Parul Institute of Technology, Vadodara, Gujarat.

Sapan Naik  
Asst. Prof. in Computer Science  
Uka Tarsadiya University  
Bardoli, Gujarat.

Sheetal Mehta  
Asst. Prof. in Computer Engg.  
Parul Institute of Technology, Vadodara, Gujarat.

ABSTRACT
Interpolation is the process of transferring image from one resolution to another without losing image quality. In Image processing field, image interpolation is very important function for doing zooming, enhancement of image, resizing any many more. Here in this paper, we have reviewed different interpolation algorithms. We have implemented all reviewed interpolation algorithms and done comparison of all.

Keywords
Interpolation, DWT, Filtering

1. INTRODUCTION
Approximating continues function’s value using discrete samples is called interpolation [16]. Image interpolation is nowadays available in many image processing tools like Photoshop [1] and other so we thought the review of interpolation method will be useful for researchers. Most common interpolation techniques are nearest neighbor, bilinear and cubic convolution. Applications of image interpolation methods are image enlargement, image reduction, subpixel image registration, image decomposition and to correct spatial distortions and many more [13, 14]. In figure 1 we have shown the basic concept of how we can enlarge image using interpolation.

Digital image is a signal, spatially varying in two dimensions. This signal is sampled and quantized to get values. All these values called pixels of image. When we increase the resolution of image from low to high, it is called up-sampling or up-scaling while reverse is called down sampling or down scaling. In this paper we have presented algorithms for up-sampling only. Two main categories are there for image interpolation algorithms called adaptive and non-adaptive. In non-adaptive method same procedure is applied on all pixels without considering image features while in adaptive methods, image quality and its features are considered before applying algorithm [2, 3].

In this paper we have studied many interpolation algorithms and implement all. We have done the comparisons of all these algorithms.

Paper organization is as follow. Section 2 contains non adaptive interpolation algorithms, section 3 contains adaptive algorithms, and in section 4 we have shown the comparisons of all algorithms and finally section 5 contain conclusion and future work.

Figure 1. Basic Interpolation Concept

2. NON ADAPTIVE INTERPOLATION ALGORITHMS
Without thinking or considering the content of image, we simply apply some computational in this methods. Normally commercial product like Adobe Photoshop cs5 uses this kind of interpolation methods.

2.1 Nearest Neighbour Interpolation
This method is very simple and requires less computation as it use nearest neighbor’s pixel to fill interpolated point. This method is just copies available values, not interpolate values as it doesn’t change values.

The interpolation kernel for each direction for this method is [16]:
\[
    u(x) = \begin{cases} 
        0 & |x| > 0.5 \\
        1 & |x| < 0.5 
    \end{cases}
\]

Where \( x \) = distance between interpolated point and grid point.

2.2 Bilinear Interpolation
In this method, interpolated point is filled with four closest pixel’s weighted average. In this method we performed two linear interpolations, in horizontal direction and then linear...
interpolation in vertical direction. We need to calculate four interpolation function for grid point in Bilinear Interpolation.

The interpolation kernel for linear interpolation is[16]:

\[ u(x) = \begin{cases} 
0 & |x| > 1 \\
1 - |x| & 0 < |x| < 1 \\
1 & |x| < 0
\end{cases} \]

Where \( x \) = distance between interpolated point and grid point.

### 2.3 Bicubic Interpolation

In this method, interpolated point is filled with sixteen closest pixel’s weighted average. We can get sharper image than bilinear method.

The interpolation kernel for cubic interpolation is[16]:

\[ u(x) = \begin{cases} 
\frac{3}{2} |x|^3 - \frac{5}{2} |x|^2 + 2 & 0 < |x| < 1 \\
-\frac{1}{2} |x|^3 + \frac{5}{2} |x|^2 - 4|x| + 2 & 1 < |x| < 2 \\
0 & 2 < |x|
\end{cases} \]

Where \( x \) = distance between interpolated point and grid point.

### 2.4 Filtering-based (Re-sampling) Techniques

Transforming image pixel from one coordinate to another (different resolution) is called Re-sampling process. Digital image is discrete function in two dimension so we need 2D filtering or convolution. But it is computationally costly. We can separate 2D function and apply it first row wise and then column wise on 1D. We have studied different filters used for interpolation algorithm[1]. Filters are used for blur and noise removal as well as edge identification. We have listed some of the filters and website from which one can learn more about these filters.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Reference</th>
</tr>
</thead>
</table>
| Michell cubic polynomial filters | • www.mentalandscape.com/Papers_siggraph88.pdf  
• cognitrn.psych.indiana.edu/busey/erp/DigFilt.pdf |
| Bell quadratic polynomial filter | http://www.sweetwater.com/expert-center/glossary/--BellFilter |
| Hermite cubic polynomial filter | • http://en.wikipedia.org/wiki/Hermite_interpolation  
• http://www.bionixwallpaper.com/help/faq/compareion.html |
| Lanczos3 sinc-based filter | • en.wikipedia.org/wiki/Lanczos_resampling |
| Triangle Filter(linear based) | • http://www2.statistics.com/resources/glossary/t/trfiltp.jpg  
• http://www.bionixwallpaper.com/help/faq/compareion.html |
| Cubic B-spline filter | • http://www.dannyruijters.nl/cubicinterpolation/  
• http://ieeexplore.ieee.org/xpl/login.jsp?D1163154 |

More about different filters can be found from http://lodev.org/cgtutor/filtering.html. Many research papers are also available for studying different filters.

### 3. ADAPTIVE ALGORITHMS

Adaptive algorithm basically uses some of the image features and improves interpolation result. Adaptive algorithm works based on different intensity present and local structure of image. Many adaptive interpolation algorithms have been proposed which enhance edges in images[7,10]. Local gradient information can also be used to enhance non adaptive interpolation algorithm. Normal problem with interpolation technique is blurring and blocking artifacts. This can be solved using directional interpolation.
3.1 Recent interpolation Algorithm

We have studied some recent algorithms for interpolation. In [17], authors have presented edge-adaptive interpolation algorithm for Super-resolution reconstruction. Here they construct high resolution image using bilinear interpolation and detect edges. Then edges are refined using geometric duality between the low-resolution covariance and the high-resolution covariance and local structure feature.

In research paper [18], authors shows new interpolation algorithm based on Pulse-Coupled Neural Networks (PCNN). Here clusters and the propagation paths of pulse are obtained by using the synchronous pulse burst property of PCNN. Different interpolation methods are used in the inner of clusters and at the intervals of clusters to complete all interpolations of image.

We have studied one research paper of Image Enhancement by Intensity Based Interpolation and Selective Threshold[19]. Here they have proposed one technique which determines edges by comparing the intensity variations between center pixel and its neighboring pixels of window size 5x5. On the basis of designed thresholds, extrapolated blank pixels are filled.

We have studied one edge based interpolation algorithm presented in [20]. Here the gradient directions are explicitly estimated with a statistical-based approach. The local dominant gradient directions are obtained by using principal components analysis (PCA) on the four nearest gradients. The angles of the whole gradient plane are divided into four parts, and each gradient direction falls into one part. Then they implement the interpolation with one-dimension (1-D) cubic convolution interpolation perpendicular to the gradient direction.

We have studied one interpolation algorithm presented in [6]. Here they have applied threshold and Wavelet lifting transform based denoising method first on image and then interpolation algorithm. We have given steps of that algorithm here.

If (value of pixel < threshold) than

\[
A(L_i) = A(i-1,j-1) + A(i-1,j+1) + A(i+1,j-1) + A(i+1,j+1) / 4;
\]

\[
B(i,j) = B(i-1,j) + B(i+1,j) / 2;
\]

or

\[
B(L_i) = B(L_{i-1}) + B(L_{i+1}) / 2;
\]

Same way for C,D and E whatever original pixel available, use one of the B’s question.

We have refer on one algorithm presented in [11], which is the modification in above algorithm. We have given Modified algorithm below.

If (value of pixel < threshold) than

\[
I(m,n) = (I(m-1,n-1) + I(m-1,n+1) + I(m+1,n-1) + I(m+1,n+1))/4;
\]

\[
B = (I(m-1,n) + I(m,n+1) + I(m+1,n) + I(m,n))/4;
\]

\[
C = (I(m,n-1) + I(m+1,n) + I(m,n+1))/3;
\]

\[
D = (I(m-1,n) + I(m,n+1) + I(m+1,n))/3;
\]

\[
E = (I(m-1,n-1) + I(m-1,n) + I(m,n+1))/3;
\]

\[
F = (I(m-1,n-1) + I(m,n) + I(m+1,n))/3;
\]

So here more pixel values are considered while calculating interpolated point’s value as well as consider the noise present in image too. Here instead of finding mean value of pixels, we can use max or median function of matlab as per the type of image to implement algorithm.

We have studied An Edge-Guided Image Interpolation Algorithm via Directional Filtering and Data Fusion presented in [5]. For edge information they have partition pixels into two directional and orthogonal subsets. Directional interpolation is made for each Subset and two interpolated values are fused. Algorithm presented in [5] work for gray scale images only. Figures below shows how they fill interpolated point.

Modification for this algorithm presented in [12]. Here they made modification so that algorithm works for RGB image also. They have stored each R,G and B components of one image into three different images of two dimension (same as grayscale image) and give that as a input to original algorithm. Finally they have merged all three output arrays into single RGB image[4].

![Original Image (m x n)](image)

![SR Image (2m-1 x 2n-1)](image)

Figure 3: Threshold based interpolation[11]
for i=1:x
    for j=1:y
        R(i,j)=Input(i,j,1);
        G(i,j)=Input(i,j,2);
        B(i,j)=Input(i,j,3);
    end
end

for i=1: (2*x)
    for j=1: (2*y)
        RGB(i,j,1)=Output(i,j);
    end
end % same way RGB(i,j,2)and RGB(i,j,3)is achieved

We have studied “image up-sampling using DWT presented in [9]. Here they have used 9/7 bi-orthogonal spline based DWT. Presented algorithm preserve the edges and color of original image. Below we have shown the figure, how they do image up-sampling (Here S is scaling factor and I is original image).

4. EXPERIMENT RESULTS AND COMPARISON
We have used Photoshop [1] for basic methods and for other algorithms, we have implemented basic algorithms in matlab. All the experiments have been done on the laptop with configuration Intel i3 processor, 4GB RAM and 512 MB graphics card. Experiment we have done on LINA and Monalisa images of the size 256 x 256. We have used Pick Signal to Noise Ratio (PSNR) for comparison of all
algorithms. Visual quality is the most important parameter for the effectiveness of algorithm. Below we have shown all comparison. We have shown the snapshot of Photoshop also in which we can apply basic methods. The PSNR is defined as: \( \text{PSNR} = 20 \cdot \log_{10}(\text{MAX} / \sqrt{\text{MSE}}) \). We have compare algorithms in terms of their computation time, complexity of algorithm, PSNR and visual quality.

5. CONCLUSION AND FUTURE WORK

In this paper we have shown different interpolation algorithms. We have implemented all algorithms and done the comparison. For commercial tools and applications which are not very important (use for commercial purpose only), non-adaptive algorithms are better to use while in application like medical, bioinformatics, satellite images, adaptive methods are good to use.

In future we would like to modify above algorithms so it will work better as well as we would like to implement our own algorithm for interpolation based on wavelet.

6. REFERENCES


[16] www.wikipedia.com


Figure 6 : Original Input Images
Figure 7: LENA image, a. nearest neighbour, b. Bilinear, c. Bicubic, d. Edge Guided, e. DWT based, f. Threshold based adaptive[1,15]
<table>
<thead>
<tr>
<th>Interpolation Algorithms</th>
<th>PSNR(dB) LENA</th>
<th>PSNR(dB) MONALISA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest neighbour</td>
<td>26.05</td>
<td>28.71</td>
</tr>
<tr>
<td>Bilinear</td>
<td>27.12</td>
<td>29.13</td>
</tr>
<tr>
<td>Bicubic</td>
<td>27.18</td>
<td>29.36</td>
</tr>
<tr>
<td>Edge Guided</td>
<td>30.43</td>
<td>31.54</td>
</tr>
<tr>
<td>DWT based</td>
<td>28.81</td>
<td>29.67</td>
</tr>
<tr>
<td>Threshold based</td>
<td>29.99</td>
<td>30.82</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interpolation Algorithms</th>
<th>Computation Time</th>
<th>Complexity of Algorithm</th>
<th>Visual Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest neighbour</td>
<td>Low</td>
<td>Low</td>
<td>Poor</td>
</tr>
<tr>
<td>Bilinear</td>
<td>Low</td>
<td>Low</td>
<td>Average</td>
</tr>
<tr>
<td>Bicubic</td>
<td>Average</td>
<td>Average</td>
<td>Good</td>
</tr>
<tr>
<td>Edge Guided</td>
<td>Very High</td>
<td>High</td>
<td>Very Good</td>
</tr>
<tr>
<td>DWT based</td>
<td>Average</td>
<td>Low</td>
<td>Average</td>
</tr>
<tr>
<td>Threshold based</td>
<td>High</td>
<td>Average</td>
<td>Good</td>
</tr>
</tbody>
</table>

Figure 8: Snapshot of Photoshop