Spring 2023

Adaptive Windowed Sinc Filter for Image Interpolation

Dang Nguyen

San Jose State University

ADAPTIVE WINDOWED SINC FILTER FOR IMAGE INTERPOLATION

A Thesis

Presented to

The Faculty of the Department of Electrical Engineering

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Dang Nguyen

May 2023

The Designated Thesis Committee Approves the Thesis Titled

ADAPTIVE WINDOWED SINC FILTER FOR IMAGE INTERPOLATION

by

Dang Nguyen

APPROVED FOR THE DEPARTMENT OF ELECTRICAL ENGINEERING

SAN JOSÉ STATE UNIVERSITY

May 2023

Chang Choo, Ph.D.	Department of Electrical Engineering
Birsen Sirkeci, Ph.D.	Department of Electrical Engineering
Jeonghee Kim, Ph.D.	Department of Electrical Engineering

ABSTRACT

ADAPTIVE WINDOWED SINC FILTER FOR IMAGE INTERPOLATION

by Dang Nguyen

In this thesis, an image interpolation scheme using an adaptive windowed sinc, which varies in length and frequency response guided by the Laplacian operator, is proposed to improve quality and accuracy. First, an optimization is performed to minimize the squared error for each sliding window by sweeping a range of windowed sinc filters with varying filter lengths. The relationship between the windows, filter length, and edge intensity is analyzed to determine the optimal beta value and filter length. This analysis inspires an image interpolation approach that uses the Laplacian of each sliding window to choose the optimal beta and filter length. The performance of the proposed approach is compared with the optimized result, as well as traditional interpolation methods such as bilinear and bicubic, in terms of PSNR, SSIM, and a subjective visual test. The findings demonstrate the effectiveness of the proposed method in enhancing the resolution of images, contributing to a deeper understanding of adaptive windowed sinc filters and their potential applications in image interpolation.

ACKNOWLEDGEMENTS

I am deeply grateful for the guidance, patience, and support of Professor Choo throughout my thesis. His unwavering trust in my work helped me stay motivated and on track, even when my confidence wavered.

Secondly, I would like to express my sincere thanks to my family, especially my mother, for their unwavering support throughout my academic journey. My mother's sacrifices and encouragement played a crucial role in my accomplishments, and I will always be grateful for her selflessness.

And lastly, I would like to extend my heartfelt gratitude to my partner, Junelle, who has been my unwavering source of support since the beginning of this journey. Her patience, encouragement, and trust in me have been instrumental in helping me achieve this fulfilling conclusion to my thesis. While the completion of this thesis is a personal accomplishment, it was truly a team effort throughout, and I couldn't have done it without her by my side.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

LSE – Least Squared Error
PSNR – Peak Signal-to-Noise Ratio
SSIM – Structural Similarity Index Measure

1 INTRODUCTION

1.1 Background and Motivation

The increasing demand for high-resolution images is prevalent in today's usage of digital photography in social media, computer graphic in videos games, video processing in cinematography, and medical imaging for diagnosis. Image interpolation plays a crucial role in fulfilling these demands by resizing, resampling, or manipulation of low-resolution images to higher resolution formats while maintaining visual clarity. Traditionally these operations are performed by generalized approaches such as Bilinear, Bicubic, or edge guided algorithms. However recent works in neural network has outclassed traditional approaches in reconstructing super resolution images while preserving fine details and textures. Neural networks can discern the intricate relationships between low-resolution and high-resolution using multiple layers of interconnected neurons. Whereas generalized and deterministic algorithms are computationally simple compared to substantial number of multiply and add operations in neural networks. The tradeoff in visual quality and computation complexity between neural network and generalized algorithms does not make them exclusive to each other. Image interpolation approaches comprising of traditional approach to produce the initial high resolution and neural network to refine the finer details has been shown [1]. And while recent advancements in hardware are addressing the limitations of neural networks and subsequently decreased tradeoff of computation complexities and cost to image quality, traditional image interpolation approaches will continue to be applicable in energy and hardware resource limited use cases. In this thesis, an image interpolation scheme using an adaptive windowed sinc, which varies in length and frequency response guided by the

Laplacian operator, is proposed to improve quality and accuracy. First, an optimization is performed to minimize the squared error for each sliding window by sweeping a range of windowed sinc filters with varying filter lengths. The relationship between the windows, filter length, and edge intensity is analyzed to determine the optimal beta value and filter length. This analysis inspires an image interpolation approach that uses the Laplacian of each sliding window to choose the optimal beta and filter length. The performance of the proposed approach is compared with the optimized result, as well as traditional interpolation methods such as bilinear and bicubic, in terms of PSNR, SSIM, and a subjective visual test.

1.2 Traditional Interpolation Methods

Image interpolation traditionally is viewed as the reconstruction of a continuous-time signal from its sampled values. Following the Nyquist-Shannon sampling theorem, the continuous signal, *S(t)*, can be perfectly recovered if the sampling frequency is twice the highest frequency component of the continuous signal. By multiplying the frequency response of the sampled signal, *S[k]*, with an ideal rectangular low-pass filter or equivalently convoluting the, *S[n]*, with the sinc function in the time domain, *S(t)* can be perfectly synthesized. The time plot and magnitude of the frequency response of the sinc in time and its ideal rectangular low pass frequency response is shown in Fig. 1.

In the case where the sampling rate is insufficient and the continuous signal is not properly band-limited, the reconstructed signal will have aliasing or distortions due to overlap of signal frequencies. In addition, the sinc function is not practical as an interpolator due to its unbounded nature. Attempting to limit sinc function in the time domain will

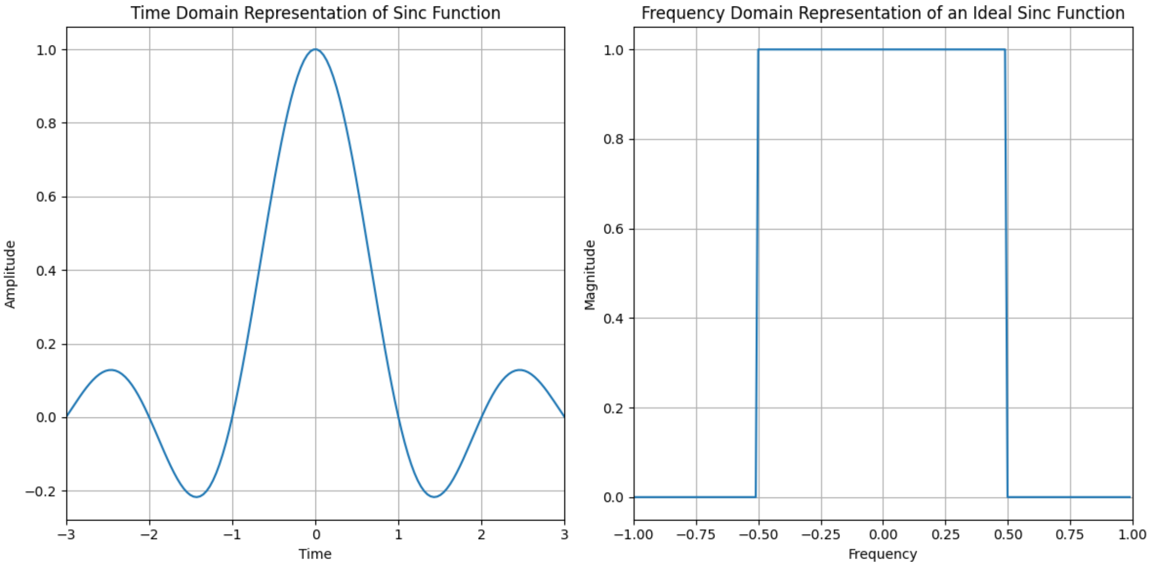


Fig. 1. Sinc function in time domain from [-3,3] and its frequency representation.

compromise the ideal cut off and stop band attenuation in the frequency domain as seen in Fig. 2. Traditional image interpolation techniques have been developed to approximate the sinc function while leveraging tradeoffs between the pass band roll off and high frequency attenuation.

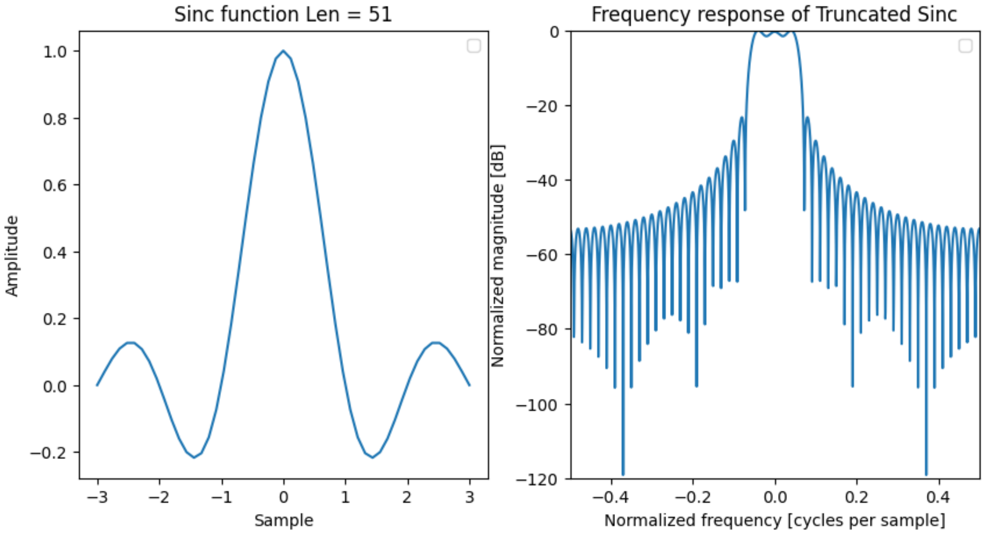


Fig. 2. Truncated sinc function and its compromised Fourier transform. The passband is no longer flat, transition band is not infinitely steep, and the stopband attenuation is not ideal.

1.3 Nearest Neighbor

Nearest neighbor image interpolation is a simple yet widely used method for increasing the size of an image. In this method, each pixel in the original image is simply replicated to create a larger image. As a result, nearest neighbor interpolation is fast and requires minimal computational resources. However, nearest neighbor interpolation produces images with low visual qualities such as blocky or pixelated and blurriness. The pixelated artifacts are evidence of poor high frequency attenuation and blurriness is a result of gradual frequency cut off. The resulting images can appear blocky or pixelated, with visible artifacts around sharp edges and transitions. Fig. 3 displays the time domain and frequency response comparison of the nearest neighbor to the truncated sinc. The time domain plot of the nearest neighbor is a poor approximation of the truncated sinc function. Thus the frequency response of the nearest neighbor has a undesirable stopband attenuation and wide transition band. Despite its limitations, nearest neighbor interpolation remains popular due to its simplicity and computational efficiency. It is often used in real-time applications where speed is critical, such as in video games or live video streaming.

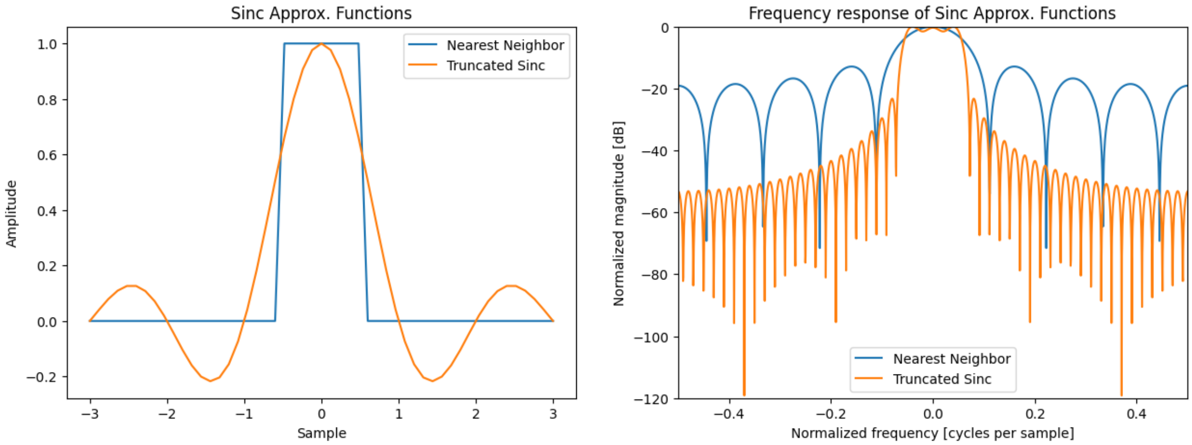


Fig. 1. Nearest neighbor kernel in time domain and magnitude of Fourier transform [2].

1.4 Bilinear Image Interpolation

Bilinear image interpolation is a more sophisticated method of increasing the size of an image compared to nearest neighbor interpolation. In this method, a new pixel is calculated as a weighted average of the nearest four pixels in the original image. The weight of each pixel is determined by its distance from the location of the new pixel. Bilinear interpolation can produce higher-quality images compared to nearest neighbor interpolation, but it still has limitations. The resulting images can still appear blurry or have jagged edges around sharp transitions. Fig. 4 displays the bilinear in comparison to the nearest neighbor and truncated sinc function in the time domain and frequency domain. With an extra degree in the polynomial, it is closer to a sinc approximation than the nearest neighbor. The frequency plot shows greater stopband attenuation than the bilinear yet still worse than the truncated sinc. Equation 1 describes the time domain function to calculate the weighted coefficients for each 2x2 neighboring pixels given their distance in X and Y.

$$f(x,y) = f(x_0,y_0)(1-w_x)(1-w_y) + f(x_1,y_0)w_x(1-w_y) + \\ f(x_0,y_1)(1-w_x)w_y + f(x_1,y_1)w_x w_y \tag{1}$$

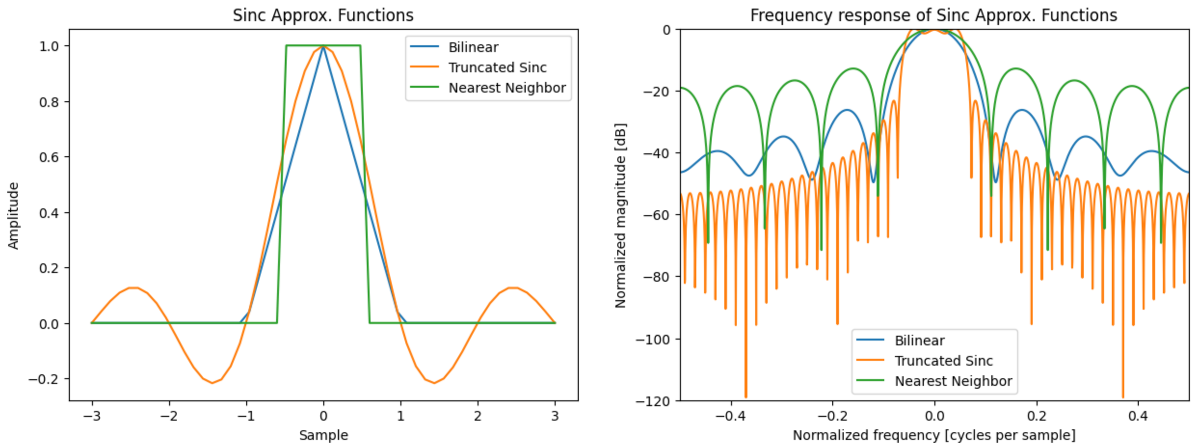


Fig. 4. Bilinear kernel and magnitude of fourier transform with comparison to the truncatded Sinc and nearest neighbor.

Bilinear interpolation is commonly used in applications such as image resizing, computer graphics, and image processing. Its relatively low computational complexity and improved image quality make it a popular choice for many practical applications.

1.5 Bicubic Image Interpolation

Bicubic image interpolation is a more complex method of increasing the size of an image compared to bilinear or nearest neighbor interpolation. In this method, a new pixel is calculated as a weighted average of the nearest 16 pixels in the original image, using a bicubic interpolation function. The weighted average for the 1-D interpolation is described in Equation 2. The time and frequency domain are plotted in Fig. 5 in comparison with the truncated sinc and bilinear functions. It is evident that the bicubic approximates the sinc function much better than the bilinear in the time domain. And this can be seen in the frequency plot that the stopband attenuation is better than the truncated sinc function. This leads to less distortions around the edges and overall smoothening of the entire image.

$$W(x) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}x^2 + 1, & 0 \leq |x| < 1 \\ -\frac{1}{2}|x|^3 + \frac{5}{2}x^2 - 4|x| + 2, & 1 \leq |x| < 2 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The additional complexity allows it to approximate the sinc function more closely and minimizes the abrupt discontinuity. The computational complexity of bicubic interpolation is higher than that of bilinear or nearest neighbor interpolation because it requires more calculations to determine the weighted average of the nearest 16 pixels. However, modern computing systems can process the additional computations required for bicubic interpolation quickly.

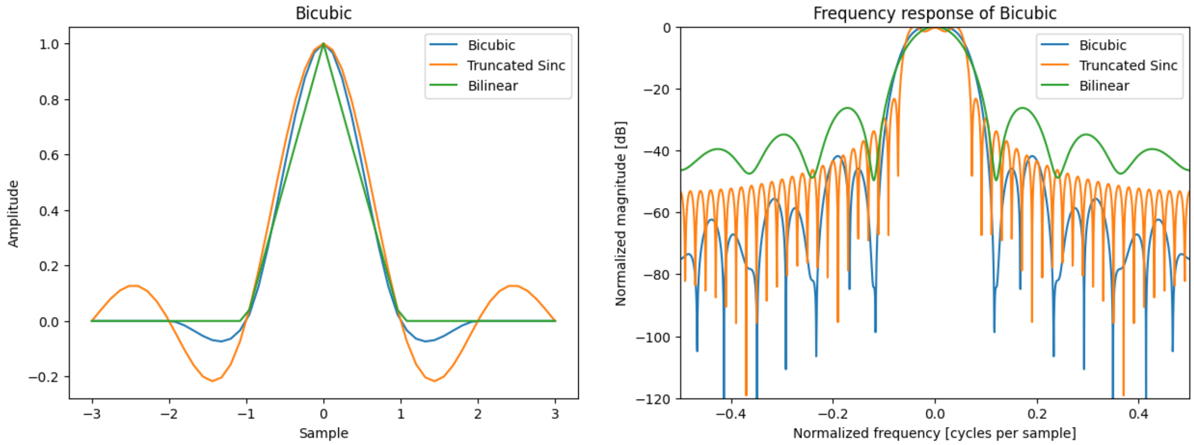


Fig. 5. Bicubic time domain approximation of the sinc function and magnitude response.

Bicubic interpolation can produce higher-quality images compared to bilinear or nearest neighbor interpolation, with smoother transitions and reduced artifacts. Bicubic interpolation is commonly used in applications such as image resizing, digital photography, and video processing. Its improved image quality makes it a popular choice for many practical applications, even though it is slower than other interpolation methods. Overall, bicubic interpolation strikes a balance between image quality and computation time, making it a popular choice for many image processing tasks.

1.6 Windowed Sinc

Similar to the previous approaches in obtaining a spatially finite sinc approximation, window functions are used to bound the sinc function in a gradual manner relative to the rectangular truncation. The lack of abrupt truncation reduces the high frequency content which contributes to the overshoot from Gibbs phenomenon in approximating signals with discontinuity or sharp edges. Many well-known window functions have been presented to taper the discontinuity, each with its own frequency responses tradeoff in term of passband ripple, and stopband attenuation. Shown in Fig. 6 and Fig. 7 are well known windows are the

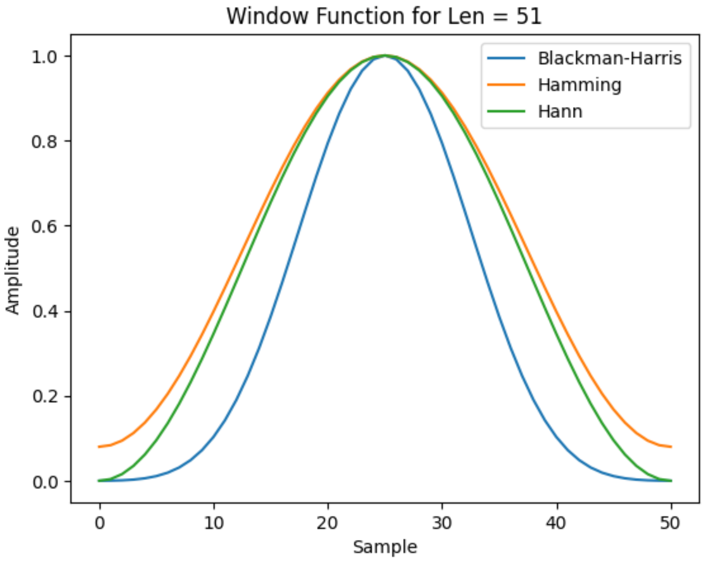


Fig. 6. Popular windowing function such as Blackman-Harris, Hamming, and Hanning.

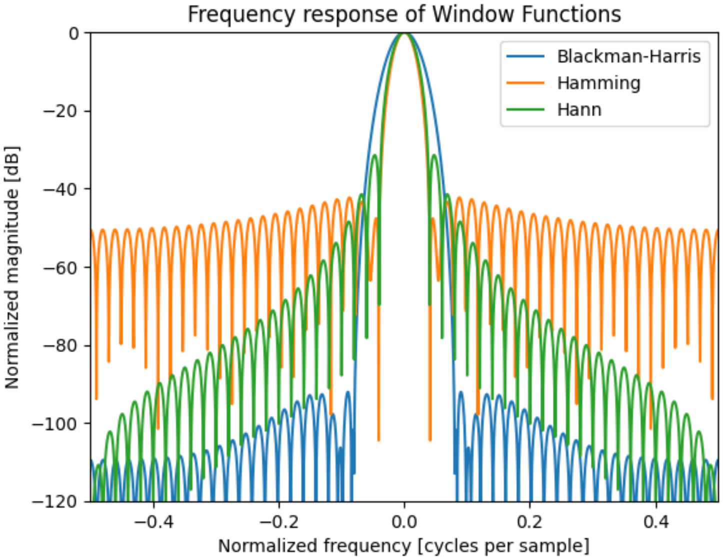


Fig. 7. Frequency response of Blackman-Harris, Hamming, and Hann windowing function.

Blackman-Harris, Hamming, and Hanning windows in time and frequency domain. The Blackman-Harris has a narrower main lobe in the time domain thus it has a larger cut off frequency and slower passband to stopband transition than the Hamming and Hanning.

1.7 Windowed Sinc Filter Length

The filter length is an important parameter in designing the windowed sinc filters as it affects the passband and stopband transition. Fig. 8 shows the frequency magnitude in logarithmic scale of the Blackman-Harris windowed sinc filter at lengths 51, 81, 111, and 141. It is evident that longer filter lengths lead to sharper transition of the passband to stopband and higher stopband attenuation. And subsequently lower filter lengths may not adequately suppress higher frequencies, leading to aliasing in the interpolated image. Overall, the filter length is another parameter where the tradeoff of image quality and computational complexities occurs.

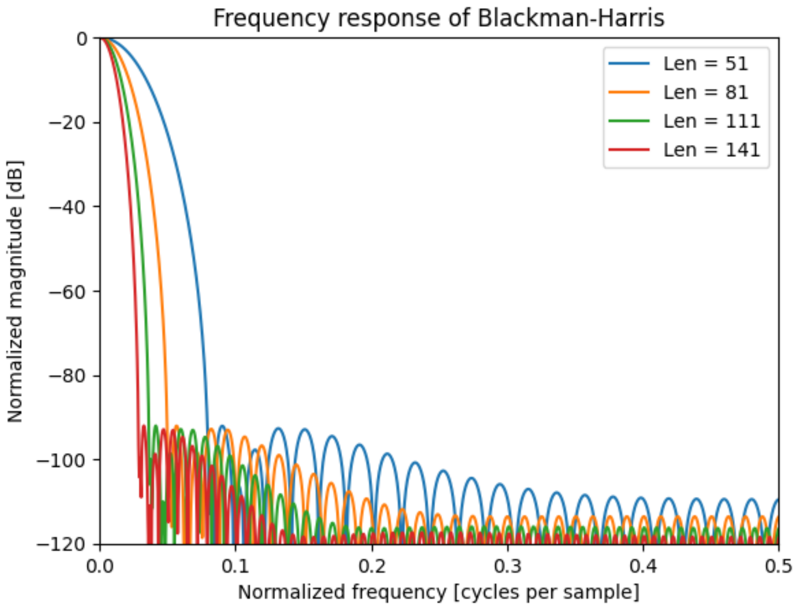


Fig. 8. Frequency of Blackman-Harris window for different length.

1.8 Kaiser Window

The Kaiser Window is a popular choice due to its adjustable properties, which allow for the fine-tuning of the trade-off between frequency resolution and side lobe attenuation. The

Kaiser window is a parameterized window function that depends on a single parameter, known as the beta parameter (β). This parameter controls the trade-off between the main lobe width and the side lobe level of the window function. A larger value of β results in a narrower main lobe but higher side lobe levels, while a smaller value of β results in a wider main lobe and lower side lobe levels as seen in Fig. 9. Fig. 10 shows the Kaiser window at beta 0 to 15 at increments of 5. At β equal to zero, the windowing function is a rectangular window or truncation operation. As β increases, the main lobe in the time domain become narrower and the tapering has greater effect. This in turns leads to a wider transition band in the frequency domain and higher stopband attenuation. The flexibility of the Kaiser window allows the optimization of least squared error detailed later in the thesis.

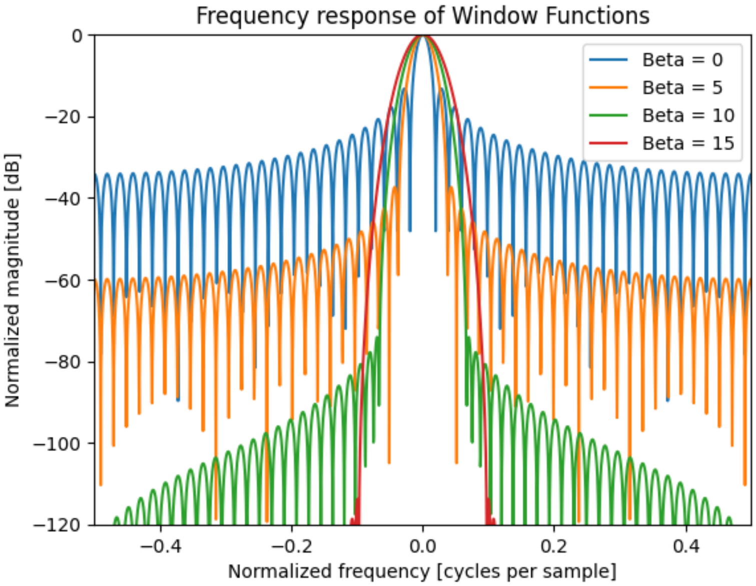


Fig. 9. The frequency response for the Kaiser window for beta = 0 to 15 at increments of 5. As beta increases, the passband and stopband transition less steep and the cut off frequency increases.

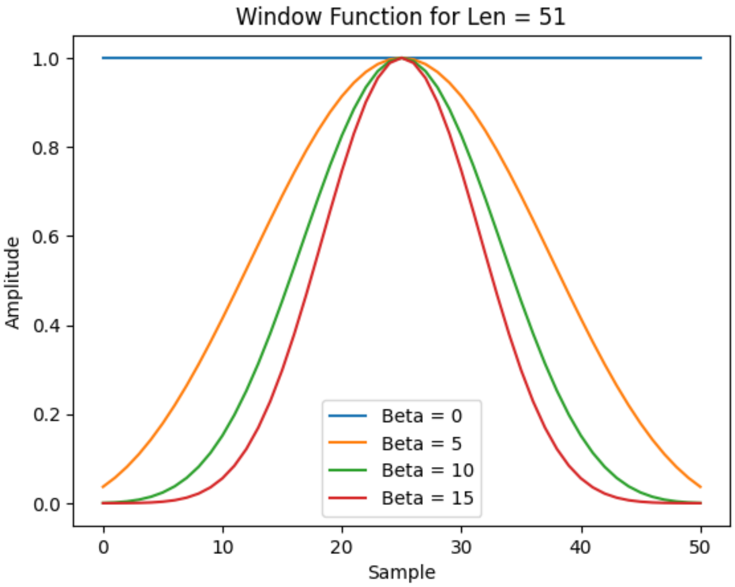


Fig. 10. The Kaiser window for beta zero to 15 in increment of 5. At beta = 0, the Kaiser window is a rectangular window. The main lobe increases as the beta increases.

2 LITERATURE REVIEW

In recent years, various image interpolation techniques have been proposed to address the challenge of preserving image structures and minimizing artifacts. This literature review focuses on three significant works in this field, presenting different approaches to adaptive image interpolation.

Zhang and Wu [3] introduce a novel edge-guided image interpolation algorithm that significantly improves image quality by preserving edges and texture details. The algorithm utilizes directional filtering and data fusion techniques, starting with a directional decision process to identify the dominant orientation of local image structures. It then interpolates the image based on this orientation and employs data fusion to merge the results from different directions. Experimental results demonstrate that this edge-guided interpolation method outperforms other popular techniques in both objective and subjective image quality assessments.

The EpicFlow method [4] consists of two main steps. First, it establishes sparse correspondences between images using a feature-matching algorithm, such as SIFT or DeepMatching. This sparse correspondence is then used as input for an edge-preserving interpolation step, which computes a dense flow field while considering the underlying image structures. By leveraging edge information from the input images, EpicFlow ensures that interpolated flow values align with the edges and structures of the scene, resulting in a more accurate and coherent flow field.

Tai et al. [5] propose a super-resolution algorithm that enhances the quality of low-resolution images by leveraging edge information and single-image detail synthesis. The algorithm consists of two main steps. First, an edge-prior-based interpolation is applied to estimate high-resolution edge structures from the low-resolution input image. This step focuses on preserving crucial edge information in the image. Second, single-image detail synthesis is employed to recover high-frequency texture details, using both the low-resolution input image and the interpolated high-resolution edge structures as inputs.

In conclusion, these three works focus on improving the quality of images by preserving crucial edge information and recovering high-frequency details. Each method presents a unique approach to improve interpolation performance by adapting to local image features, preserving edge structures, and minimizing artifacts.

3 METHODOLOGY

3.1 Overview

This section of the thesis describes the methodology of obtaining the optimal Kaiser windowed sinc filter and filter length for each sliding window in an image interpolation. First a filter bank is created based upon the beta parameters and maximum filter length. Based upon the Fourier analysis of windowed sinc, the filter bank contains filters which ranges in cut off frequency, stopband transition, and steepness of the roll off**.** To obtain the optimal windowed Kaiser window and filter length for each sliding window, each filter is used with the sliding window of appropriate size centered at $I_{LR}(x, y)$. The optimal beta value and filter length of the filter, which corresponds in the least squared error to the reference image pixel, $I_{HR}(x,y)$, is returned. Each beta value and filter length are placed within its respective image sized array. A subjective visual analysis based upon the 2D plot is conducted followed by a statistical analysis to confirm the observed visual trend. Based on the confirmed trends of beta values and filter lengths, a generalized image interpolation technique guided by the Laplacian operator can be applied to a wider range of images.

3.2 Image Interpolation Operation

For each interpolating pixel, a different kernel is applied based on its distance from the center pixel of the sliding window. Shown in Fig. 11 is an illustration of operation broken down into two 1-D convolution steps. First an intermediate pixel along the x-axis is interpolated using a 1xN kernel dependent upon the lateral distance or the result of pixel window * 1xN0 in the figure for cell A. Then the intermediate results are convoluted with a 1xN1 dependent upon the vertical distance for the final pixel value.

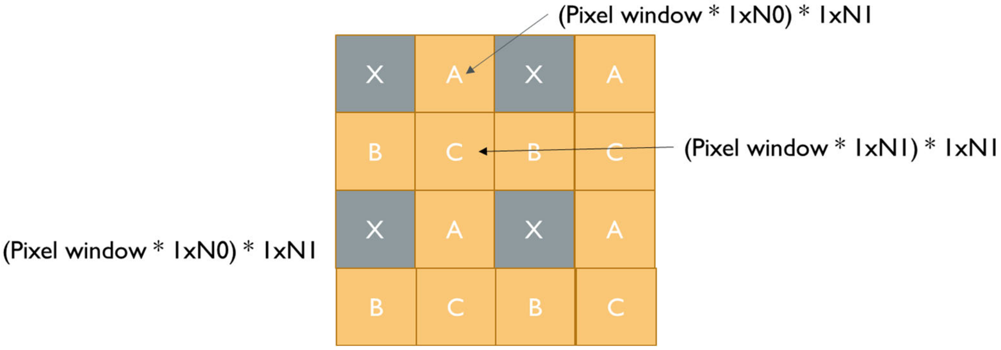

(Pixel window * 1xN0) * 1xN1
X
A
X
A
B
C
B
C
(Pixel window * 1xN1) * 1xN1
X
A
X
A
(Pixel window * 1xN0) * 1xN1
B
C
B
C

Fig. 11. Image convolution operation for a scale of 2. Cells A, B, and C are the interpolated pixel values while gray X is the replicated sampled pixel. N0 is a windowed sinc 1xN kernel with displacement in X axis and zero displacement in Y.

These steps can be combined by matrix multiplication of the two 1xN0 and $(1xN1)^T$ kernels to obtain an NxN kernel. An element-wise dot product the NxN sliding window with the NxN kernel to obtain the resulting interpolated pixel.

3.3 Filter Bank

The filter bank is generated using three parameters, a maximum beta β_{max}, the incremental step of the beta values β_{incr}, and a filter length maximum value. The minimum beta value, β_{min}**,** is set to zero and the number of beta values is shown in Equation 3.

$$Number\ of\ Beta\ Values = \frac{\beta_{max}}{\beta_{incr}} \tag{3}$$

The minimum filter value is set to 3, the incremental value is set to 2, and the number of filter values is shown in Equation 4**.** The numbers of filters within a filter bank are the products of unique beta and filter values.

$$Number\ of\ Filter\ Values = \frac{F_{max}-1}{2} \tag{4}$$

The python implementation of the filter bank is constructed as a 2-dimensional array of class objects. The first dimension of the array holds the filters of a unique beta value. The second dimension holds all the filters of the same filter size. Within each element of the array are a list of shifted windowed sinc of a beta value, β, and filter size for a specified scaling factor. An illustratio nof the filter bank with the maximum length parameter of 11 is shown in Fig. 12.

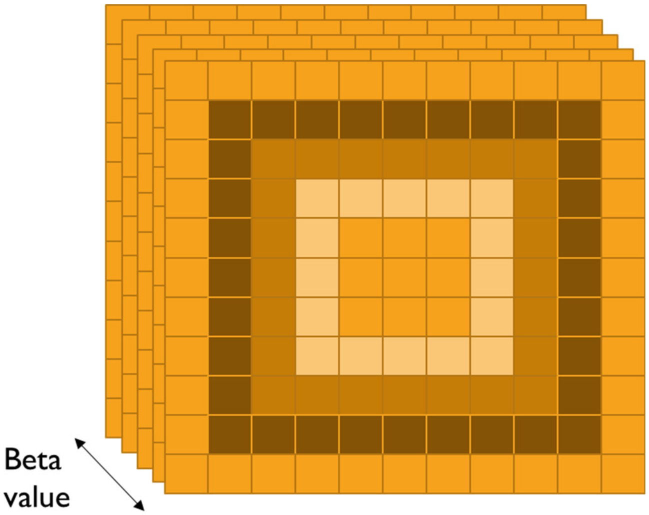
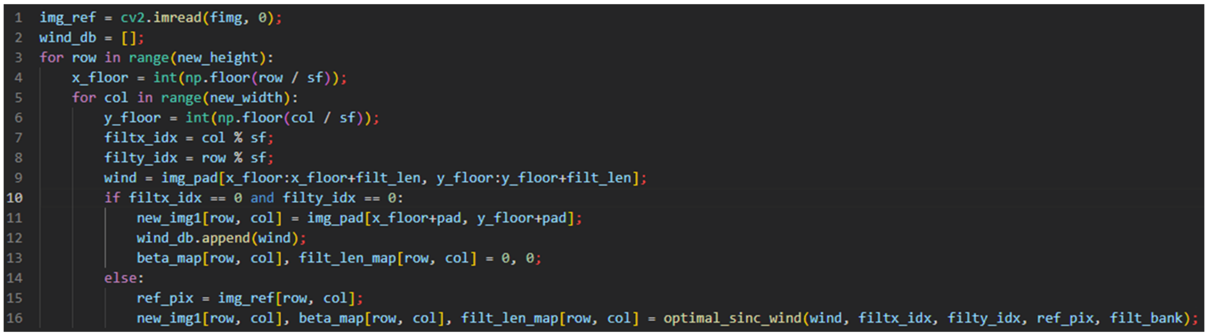


Fig. 12. Filter bank illustration for filter max parameter of 11, the max value not being inclusive.

3.4 Image Interpolation Operation

The image interpolation procedure consists of two loops which traverses through the upscaled image array size row and column as shown in Fig. 13. For each corresponding row and col value in the upscaled image, I_{HR}, the nearest pixel in I_{LR} is calculated on line 4 and 6. The x_floor and y_floor paired values indicate the center of the sliding window and line 9 samples an F_{max} x F_{max} sliding window from I_{LR}. The code then calls a function called optimal_sinc_wind() and passes the sliding window, the row and column pair within the I_{HR}, reference pixel value, and the filter bank. Within the optimal_sinc_wind function, two loops



```
img_ref = cv2.imread(fimg, 0);
wind_db = [];
for row in range(new_height):
    x_floor = int(np.floor(row / sf));
    for col in range(new_width):
        y_floor = int(np.floor(col / sf));
        filtx_idx = col % sf;
        filty_idx = row % sf;
        wind = img_pad[x_floor:x_floor+filt_len, y_floor:y_floor+filt_len];
        if filtx_idx == 0 and filty_idx == 0:
            new_img1[row, col] = img_pad[x_floor+pad, y_floor+pad];
            wind_db.append(wind);
            beta_map[row, col], filt_len_map[row, col] = 0, 0;
        else:
            ref_pix = img_ref[row, col];
            new_img1[row, col], beta_map[row, col], filt_len_map[row, col] = optimal_sinc_wind(wind, filtx_idx, filty_idx, ref_pix, filt_bank);
```

Fig. 13. Code excerpt for image interpolation. The center of each sliding window is calculated based upon the upscaled location.

traverse through the beta value and filter size, respectively. Upon achieving an error which is less than the current least error value, it will update the optimal beta, filter length, and pixel value and return at the end of both loops.

3.5 Performance Metrics

To evaluate the performance of the filter bank optimization approach against other image interpolations methods, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) will be used as the performance metrics. PSNR is widely used for measuring the quality of image construction. The arithmetic operation is described in Equation 5**.** The MSE is calculated by taking the average of the squared differences between each pixel in I_{HR} and in the reconstructed image. The PSNR is then calculated as the ratio of maximum possible pixel value, 255, to the square root of the MSE. Typically, PSNR is expressed in decibels (dB), thus a higher PSNR indicates a good measure of image similarity.

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right) \tag{5}$$

SSIM measures the similarity between two images by considering their luminance, contrast, and structural similarity. The luminance comparison is based on the mean brightness of the two images, while contrast is based on the standard deviation of the pixel intensities. The structural comparison is based on the correlation between local pixel neighborhoods in the two images. The index ranges from -1 to 1, with the value of 1 indicating perfect structural similarity between the reference image, I_{HR}, to the reconstructed image. Equation 6 describes the measurement for image ‘x’ and image ‘y’.

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2) + \delta} \tag{6}$$

Both metrics are widely used for image interpolation quantitative measurements, but they should be used in conjunction with subjective evaluations. PSNR does not consider perceptual quality of the image as it is the difference in terms of pixel values. Both metrics are global and provide a single score for the entire image. PSNR and SSIM do not provide qualitative conclusions to local variations in image quality.

4 RESULTS

4.1 Interpolated Image Result

The reconstructed image using the optimal windowed sinc filters showed comparable PSNR and comparable SSIM to both bicubic and bilinear interpolation methods, when evaluated against the original image. Shown in Fig. 14 is the result of the Lena image interpolated by scale of 2 using Bicubic, Bilinear, and the Optimization for least squared error from the filter bank. The Bicubic has the highest PSNR at 32.22 dB and 0.7614. Table 1 displays the PSNR and SSIM for the optimal windowed sinc sweep, bicubic, and bilinear approaches with scale factors of 2, 4, 5, and 10. The performance in terms of PSNR and SSIM is of the optimal window sinc sweep is comparable to the Bicubic and better than the Bilinear across scaling factors.

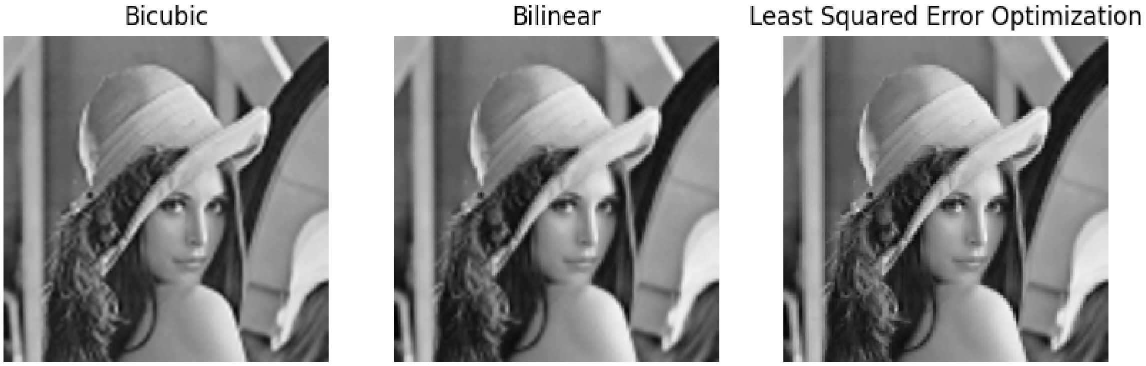


Fig. 14. Image interpolation of Lena image for scale of 2. Bicubic, Bilinear, and Optimization from filter bank sweep.

However, it is evident there are high pixelated artifacts in the optimal windowed sinc sweep result. This is an artifact from the highly pixelated downscaled image, I_{LR}, as shown in Fig. 15. It is absent in the Bicubic and Bilinear and it is suspected that post processing blurring occurred. The obstacle is not fully addressed in this thesis, it presents an opportunity for further investigations.

Table 1
Results Comparison between Bilinear, Bicubic, and Optimization for Least Squared Error in Terms of PSNR and SSIM

Scale	PSNR				SSIM		
	Bilinear	Bicubic	Optimization LSE		Bilinear	Bicubic	Optimization LSE
2	35.22	**36.19**	35.85		0.9126	**0.932**	0.926
4	33.06	**33.63**	33.32		0.809	0.835	**0.838**
5	32.02	**32.22**	32.10		0.745	**0.761**	0.729
10	30.81	**31.15**	31.08		0.64	0.66	**0.68**

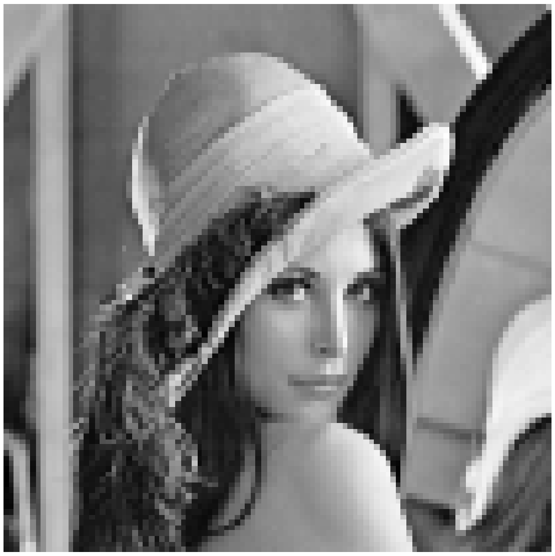


Fig. 15. Lena image downsampled by scale of 2. Noticable pixelation that has transpired to the upsampled image.

Despite the pixelated artifacts, further subjective analysis at the optimal windowed sinc sweep shows highly refined details that is absent from Bilinear and Bicubic result. Within the enlarged area of Lena's eyes in Bicubic, Bilinear, and the optimization of least squared error approach for scale of 10. The bicubic and bilinear results are noticeably blurred and lacked details such as her pupils.

4.2 Optimal Beta Map

Based upon the 2-D plot of optimal beta image sized array, a visual trend of low beta values corresponding to edge features is observed. Fig. 16 is the 2D plot of the optimal beta value and its distribution from interpolating the Lena image at scale of 2, beta value ranges from 0 to 15 at steps of 1, and the filter max length parameter is 13. The colorbar depicts low beta values as dark blue color and large beta values as bright green. A visual trend of non-zero beta values as the optimal value near edge features can be seen. This indicates that using rectangular windows of the sinc function leads to accurate interpolation for smooth surfaces. For sliding window with high edge content, a wide bandwidth window of the sinc function leads to the least squared error. This conclusion leads to a hypothesis of edge detection kernels are able to guide the filter bank to an optimal filter for each sliding window.

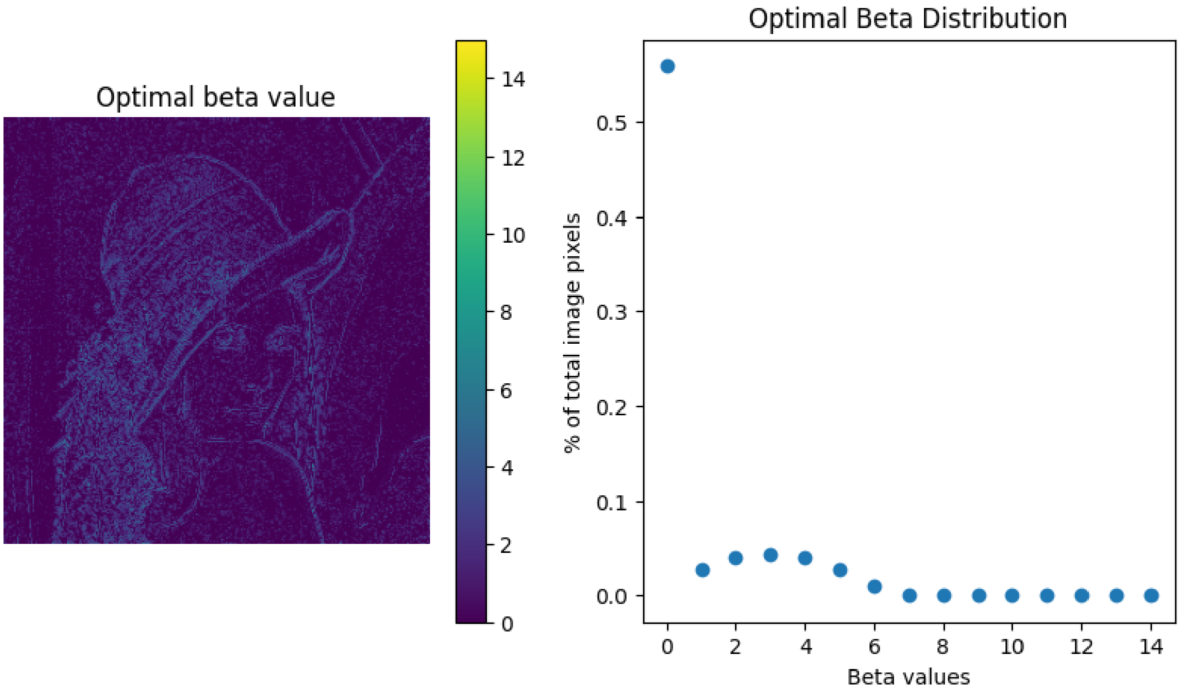


Fig. 16. Optimal beta map of Lena at scale of 2 and its distribution with respect to total image pixel. A visual trend of low beta near edge features is observed.

4.3 Optimal Filter Length Map

A subjective observation of the optimal filter 2D plot indicates a strong relationship of high filter length and sliding windows with large constrasts. Fig. 17 is the 2D plot of the optimal length of Lena at scale of 2 and filter max parameter at 13. For accurate interpolation, this plot indicates filter sizes larger than 3x3 on edge features and 3x3 filter size on smooth surfaces should be used. Low filter length leads to slow transition of the passband and high attenuation of the stopband. The distribution of filter sizes with respect to the total number of pixels within the image as shown in Fig. 17**.** Roughly half of the distribution uses 3x3 filter and 15% of the total image uses 5x5. And this distribution will vary from image but the trend of lower filter length at smooth surfaces and high filter length on high contrast area will lead to least squared error in the interpolated result.

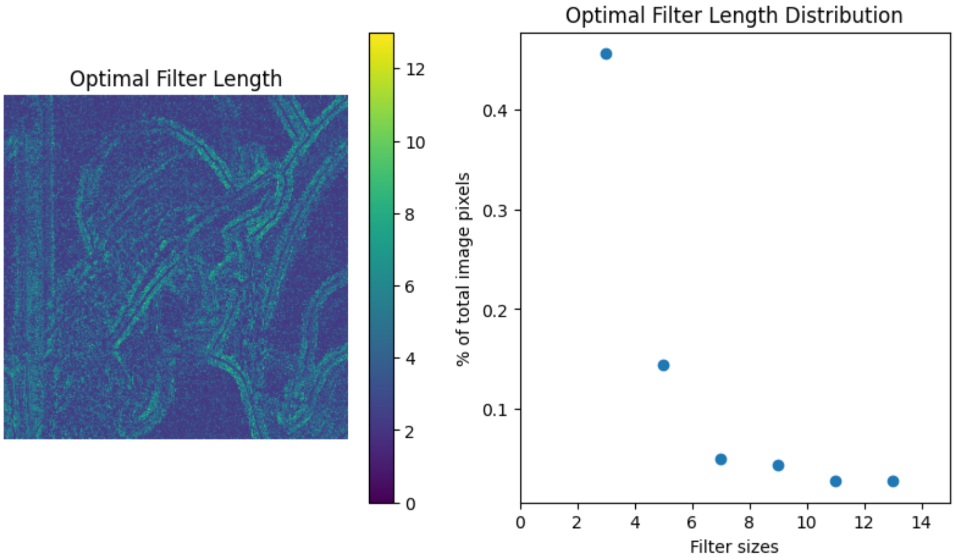


Fig. 17. Filter length value which leads to the least error for scale factor of 2. It is evident that lower filter sizes near edge features leads to lower error.

4.4 Optimal Beta and Filter Length vs Laplacian Operator

The visual trend of 3x3 filters with low beta values near edge features is strongly correlated with the second derivative of the image. The Laplacian operator is a second-order differential operator that computes the rapid intensity changes of an image. Fig. 18 displays the Laplacian result after an absolute operation, the binarized image after thresholding the absolute result to 9 and greater, and the locations of filter size greater than 3x3, respectively. The strong correlation observed between the filter size greater than 3x3 and specific thresholding of the Laplacian image indicates that the Laplacian serves as an excellent guide for choosing the optimal filter size for each sliding window. By leveraging this relationship, a more efficient and adaptive image interpolation scheme can be developed, resulting in improved image quality and accuracy.

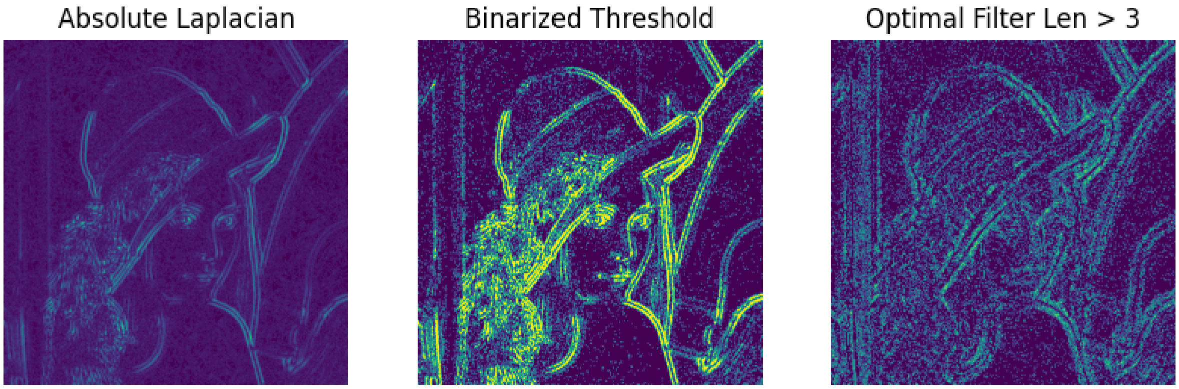


Fig. 18. Plot of absolute Laplacian output, binarized image when thresholding for values 9 and above, and the locations of 3x3 filters in the optimal window sinc sweep for scale factor of 2.

5 ADAPTIVE FILTER GUIDED BY LAPLACIAN OPERATOR

The strong correlation observed between the low filter size and specific thresholding of the Laplacian image indicates that the Laplacian serves as an excellent guide for choosing the optimal filter size for each sliding window. By leveraging this relationship, a more efficient and adaptive image interpolation scheme can be developed, resulting in a practical implementation of an adaptive window sinc filter which varies in frequency response and filter size according to the Laplacian value. Fig. 19 illustrates the implementation of the adaptive filter where a sliding window undergoes a Laplacian operation and subsequently a binarize thresholding. Based upon the threshold, the correlating filter with an optimal beta and filter size is selected for the element-wise dot product. The buffer allows the sliding window to wait for the optimal filter to be selected.

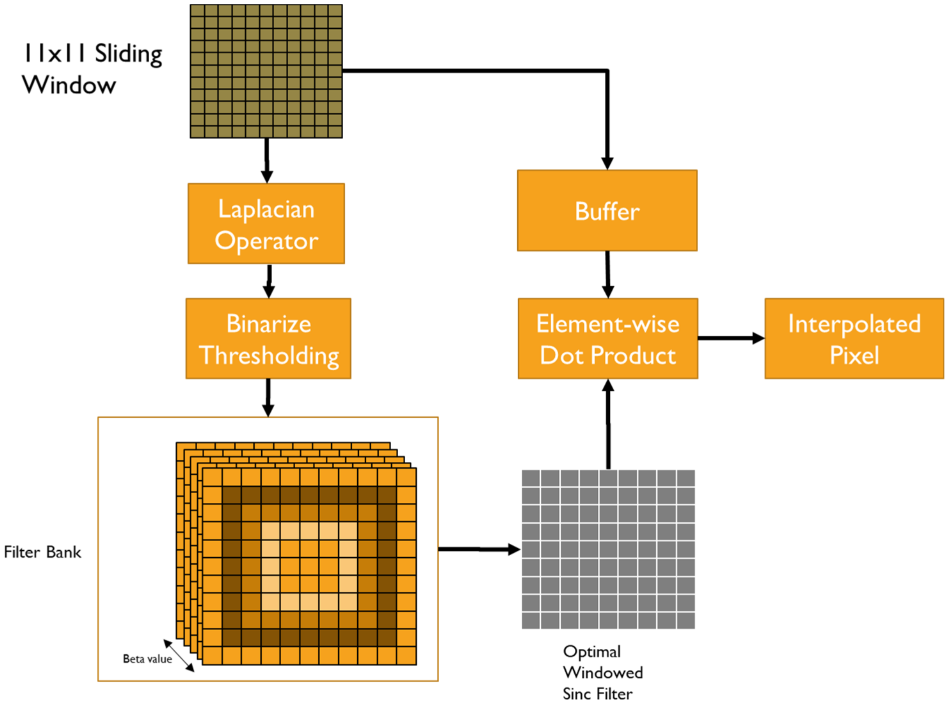


Fig. 19. Implementation flow of an adaptive window sinc filter which varies in frequency response and filter size guided by the Laplacian Operator.

6 CONCLUSION

By optimizing for the least squared error using a range of Windowed Sinc filters with varying filter sizes, a clear trend emerged that favored using Kaiser windows and filter lengths that were strongly correlated with the Laplacian output of the image. This correlation allowed for the generalization of choosing the window function and filter size that resulted in the least squared error to the original image. The interpolation, guided by the Laplacian filter, achieved results comparable to other traditional methods. While the resulting image did contain pixelation artifacts, it also exhibited fine details that were absent in Bicubic and Bilinear methods. The findings of this work are valuable for applications that prioritize intricate details of local areas within an image with limited computational hardware resources.

7 FURTHER WORK

Although this thesis has demonstrated the effectiveness of using varying Kaiser windows and filter lengths strongly correlated with the Laplacian output of the image, there are still opportunities for further work. One possible direction is to explore the implementation of this method on an FPGA platform. FPGA implementation has been shown to have advantages over traditional CPU or GPU implementations in terms of parallelism and power efficiency. Therefore, developing an FPGA-based implementation could significantly speed up the processing time and improve the energy efficiency of this method.

Another avenue for further research is the application of AI neural networks for the task of selecting the optimal windowed sinc filter and filter size for a sliding window input. The neural network could learn the more intricate relationships of a sliding window to the optimal parameters given the large amount of sliding window as training inputs per image. This solution of using both neural network and generalized approach could possibly lead to better trade-off of computational complexities and image quality.

Literature Cited

[1] C. Dong, C. C. Loy, K. He, and X. Tang, “Learning a deep convolutional network for image super-resolution,” in *European Conference on Computer Vision* (D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, eds.), pp. 184-199, Springer, 2014.

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

[3] L. Zhang and X. Wu, “An edge-guided image interpolation algorithm via directional filtering and data fusion,” *IEEE Transactions on Image Processing*, vol. 15, pp. 2226-2238, 2006.

[4] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, “EpicFlow: Edge-preserving interpolation of correspondences for optical flow,” in *IEEE Conference on Computer Vision and Pattern Recognition*, (Boston, MA, USA), June 2015.

[5] Y.-W. Tai, S. Liu, M. S. Brown, and S. Lin, “Super resolution using edge prior and single image detail synthesis,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (San Francisco, CA, USA), June 2010.