Custom Folded Clausen based Low Pass Filters

Diganta Misra
School of Electronics Engineering
Kalinga Institute of Industrial Technology, Deemed to be University
Bhubaneswar, India
mishradiganta91@gmail.com

***Abstract*— Filter Design is an important procedure of Image Processing to present new digital filters which can challenge the current state of the art filters with improved performance and results. This paper is a comprehensive, intuitive and novel approach of designing Custom Folded Clausen based low pass filters inspired from the architecture of a standard Gaussian Gabor Filter which can be applied on images for noise removal, smoothening or blurring the image, low-level abstract detection of spatial orientated edges and low-level segmentation; and can be modified to improve performance or modulate its use.**

Keywords*— *Clausen Function, Digital Filters, Image Processing, Folded Normal Distribution, Gabor Kernel and Low Pass Filter.

I. INTRODUCTION

Digital Filters [1][2] are an integral part of the Signal Processing pipeline and Image Processing [3][4][5]. Digital filters in image processing are usually used to suppress either the high or the low-frequency components to perform various tasks like smoothening an image, blurring an image, removal of noise, contour modelling by edge detection, segmentation, et cetera. Over the years of digital processing of signals and images, various sub-types of filters primarily of 4 main types – Low-pass, High-pass, Band-stop, and Band-pass have been prevalent. In the context of Digital Signal Processing, there are two main categories which are FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filters. Some of the most common filters used in Digital Signal Processing include Butterworth Filter [6][8], Kalman Filter [7], Chebyshev Filter [8], et cetera while in the domain of Image Processing include Gabor Filter [9][10][11], Bilateral Mean Filter [12], Laplacian of Gaussian Filter [13], et cetera.

In this paper, a new intuitive low-pass filter design which is termed as "Custom Folded Clausen Filter" is introduced. These low-pass filters resemble Finite Impulse Response (FIR) low-pass filter characteristics and are inspired by the design architecture of Gaussian Gabor Filter. In this research, the term "Custom Folded Clausen Filters" represent an amalgam of 6 variants of Folded Clausen Filters having variable parameters set to predefined values which demonstrate unique image processing functionalities like texture analysis, low-level abstract edge detection, image smoothening and blurring. The paper also establishes the computational expenses of these custom filters being significantly less than that of Gabor Filter and also explores these filters in the Frequency Domain which provides insight into the similarity of Custom Clausen Filters to that of the conventional low pass FIR filters.

These custom Clausen filters provide a clear understanding of new methods to design filters with just using fundamental mathematical functions rather than defining exclusive filter parameters used in standard procedures of Filter designing. The paper aims to provide a fundamental analytical approach towards filter designing based on a particular image processing technique which enables each custom filters to have its own unique functionality.

II. MOTIVATION

Gaussian Gabor Filters [9][10][11] have been used extensively in Image Filtering techniques like Texture Analysis and detecting local feature representations. It is designed using two components- Gaussian Kernel and Sinusoidal Kernel. Gaussian Gabor Filters have been debated expansively to be comparable with the mammalian visual cortex system, thus providing insight into mammalian visual perception.

Inspired from a Gaussian Gabor filter, this paper involves in defining, and construction of novel custom low pass digital filters using two primary components – The Clausen Function [14][15][16] and The Folded Normal distribution [17]. The motivation behind designing these filters was to improvise on the robustness of Gabor Filter and provide a highly customizable and efficient filter which provides extensive functionalities like Image Blurring, Noise Removal, and Abstract Edge Detection while at the same time challenge the performance level of Gaussian Gabor Filter.

III. RELATED WORK AND CRITICAL ANALYSIS

Table I provides a comprehensive literature review along with their critical comments.

TABLE I. RELATED WORK AND LITERATURE ANALYSIS

Reference Number	Proposed Work/ Architecture	Critical Comments
[18]	The authors have developed the digital versions of Prolate Spheroidal function filters of Slepian, Landau, and Pollak and investigated the incompletely specified least mean squared error filter. The authors also have compared the digital filters on the basis of Frequency Response Analysis.	The authors haven't provided any definitive comparative analysis on the computational expenses of the low pass filters mentioned.
[19]	The author has presented a set of approximation relationships between linear-phase, FIR and low-pass filter parameters and also proposed a design framework for filter design capable of finding appropriate value of unspecified parameter	The author failed to provide any optimization techniques for parameter value estimation and also provide no data-based analytics or performance behavior comparison of the filters designed using the unspecified parameter. The paper also provides no time/ memory complexity analysis of the Filter designing process.

[20]	The authors have presented a general-purpose program that is capable of designing a large class of optimum FIR linear phase digital filters. The program is written Fortran and was documented by both comments and flowcharts. The algorithm uses a Remez exchange method to design filters with minimum weighted chebyshev error in approximating a desired ideal frequency response. It minimizes the difference between the ideal filter and the filter that can be realized in practice. This algorithm is capable of designing a variety of standard filters for any desired magnitude response that is specified by the user. It focuses on the description of how it works with the help of several examples that are illustrated with specific applications. The speed of this algorithm makes it useful for a wide range of design applications.	In spite of the practical approach, the author has failed to solve the problem of approximation theory. The approximation of the equiripple Low-Pass FIR filter remains unsolved. The limitation of the procedure is that the relative values of the amplitude error in the frequency bands are specified in means of weighting function and not by deviations.
[21]	The authors have demonstrated a novel improved way of designing FIR low-pass and band-pass filters using evolutionary algorithms- Particle Swarm Optimization (PSO) and using Minimax and Least Mean Squared (LMS) error based strategies to improve filter performances and convergence rates.	The author, however, provided no definitive strategy of finding optimal population size or iteration control thus failing to demonstrate parameter optimization for the PSO algorithm. The authors also fail to provide any details regarding the computational power used for optimization.
[22]	The authors have provided a new IIR filter design method using Artificial Bee Colony (ABC) algorithm and compared it to PSO and LSQ non-linear optimization functions.	The design architecture is not implemented for FIR filter design and the paper is comprehensively defined in the Z-domain, the filter properties aren't explored in the spatial domain or in any real-world data. The author also hasn't specified the computational complexity of the ABC optimization process and has not demonstrated whether ABC optimization has any memory or time complexity tradeoffs to that of PSO or LSQ Non-Linear optimization algorithms.

IV. Clausen Function

Clausen Function [14][15][16] was introduced by Thomas Clausen to be a single variable based non-finite transcendental special function. Clausen Function are of many classes and are an integral part of modern mathematics research used in solving polylogarithmic and logarithmic integrals. The Clausen function can be expressed in terms of trigonometric, definite integral and other special functions. It is extensively connected to other algorithms and functions including Barnes Function, Reimann-Zeta Function, Dirichlet Eta and Beta functions, et cetera. Contrary to being a multi-class function, Clausen Function of order 2 is often referred to as the Clausen Function which can be represented by the following integral:

$$Cl_2(\varphi) = \int_0^{\varphi} \log|2sin\frac{x}{2}|\,dx \tag{1}$$

In (1), the absolute sign bounding the sine function maybe omitted considering φ lies within the range $[0, 2\pi]$.

Upon replacing z with a non-negative integer in the standard clausen functions, it can be defined using Fourier Series Representations as given in (2), (3), (4) and (5):

$$Cl_{2m+2}(x) = \sum_{k=1}^{\infty} \frac{\sin(kx)}{k^{2m+2}} \tag{2}$$

$$Cl_{2m+1}(x) = \sum_{k=1}^{\infty} \frac{\cos(k\theta)}{k^{2m+1}} \tag{3}$$

$$Sl_{2m+2}(x) = \sum_{k=1}^{\infty} \frac{\cos(k\theta)}{k^{2m+2}} \tag{4}$$

$$Sl_{2m+1}(x) = \sum_{k=1}^{\infty} \frac{\sin(k\theta)}{k^{2m+1}} \tag{5}$$

In (2) and (3), $Cl_n(x)$ are termed to be as Standard Clausen functions while in (4) and (5), $Sl_n(x)$ are termed to be as SL type Clausen Functions or Glaisher-Clausen Functions [16] which is also represented as $Gl_n(x)$. The Glaisher-Clausen Functions have a close established relationship with Bernoulli Polynomials. Deriving from the Fourier series representations of Bernoulli Polynomials, we establish the SL-type Clausen functions of various orders, which are given in (6) and (7):

$$Sl_1(x) = \frac{\pi}{2} - \frac{x}{2} \tag{6}$$

$$Sl_2(x) = \frac{\pi^2}{6} - \frac{\pi x}{2} + \frac{x^2}{4} \tag{7}$$

The standard Clausen function of simply first order is represented by the equation given in (8):

$$Cl_1(x) = C_1(x) = -\ln\left|2\,sin\frac{x}{2}\right| \tag{8}$$

A. 1st Order Clausen Function

The research involved designing one of the two filters using a 1st order Clausen function [15][16]which is mentioned in (8) as one of the two building components of the filter. Analyzing this we observe its statistical distribution and their 2D representation as shown in figure (1).

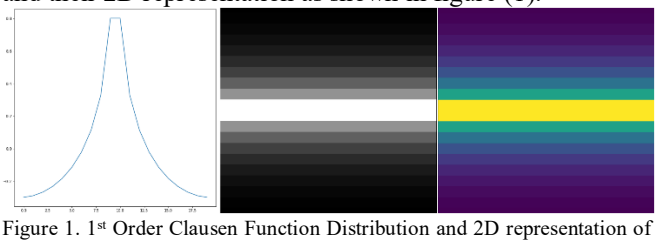

Figure 1. 1st Order Clausen Function Distribution and 2D representation of 1st Order Clausen Function Kernels in grayscale and RGB.

B. 2nd Order Glaisher Clausen Function

The second type of filter design was based on a 2nd Order Glaisher Clausen or SL-type Clausen Function as mentioned in (7) as one of the building components of the filter. In figure (2), the statistical distribution of the function and its 2D representation is shown.

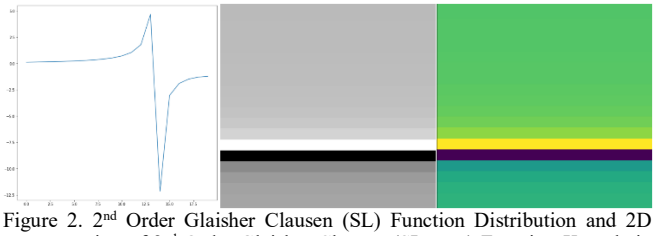

Figure 2. 2nd Order Glaisher Clausen (SL) Function Distribution and 2D representation of 2nd Order Glaisher Clausen (SL-type) Function Kernels in grayscale and RGB.

V. Folded Normal

The folded normal distribution [17] is a statistical distribution closely related to the Normal distribution and can be defined using mean (μ), variance (σ) and the distributed input x as given in (9) :

$$f(x) = \sqrt{\frac{2}{\pi\sigma^2}} e^{\frac{-(x^2+\mu^2)}{2\sigma^2}} \cosh\left(\frac{\mu x}{\sigma^2}\right) \quad (9)$$

The research involved using 6 pairs of parameters including various values for mean (μ) and variance (σ) as shown in Table II.

TABLE II. PARAMETER SETTINGS FOR FOLDED NORMAL

Sl. No.	Mean (μ)	Variance (σ)
1.	0	1
2.	1	1
3.	0.5	1.5
4.	0.5	2
5.	2	1
6.	0	0.5

The folded distribution's statistical graphical representation using mean set to be 0 and variance set to be 1 which makes the *cosh* term in (9) to be 1 is shown in figure (3).

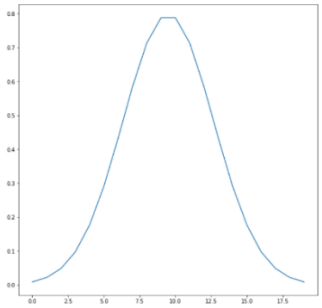

Figure.3. Folded Normal Distribution with μ=0 and σ=1

The 2D representation of all variants of Folded Normal used in this research is demonstrated in figure (4).

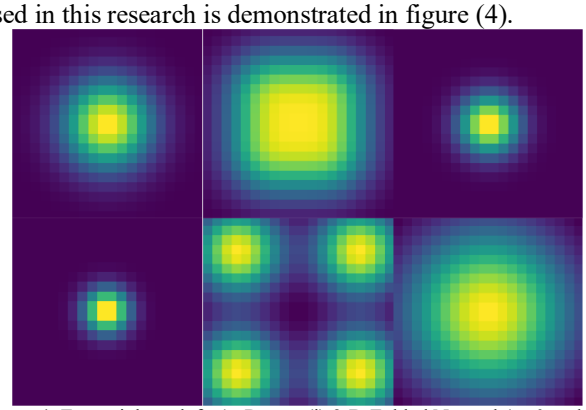

Figure 4. From right to left: 1st Row – (i) 2-D Folded Normal (μ=0 and σ=1) (ii) 2-D Folded Normal (μ=1 and σ=1) (iii) 2-D Folded Normal (μ=0.5 and σ=1.5). 2nd Row- (i) 2-D Folded Normal (μ=0.5 and σ=2) (ii) 2-D Folded Normal (μ=2 and σ=1) (iii) 2-D Folded Normal (μ=0 and σ=0.5).

For comparison with a Gaussian filter, the Folded Blur kernel with μ=0 and σ=1 was applied to a test image which shown in figure (5) demonstrates significant blurring and smoothening also resulting in reduced contrast levels:

Figure 5. From right to left :(i) Original Image(ii) Folded Normal applied Image in grayscale(iii) Folded Normal applied Image. (Original Image Credits: Scikit-Image)

VI. Filter Design and Analysis

Filter Design [18][19][20][21][22] is a crucial procedure in understanding and evaluating filters. The framework of a filter also provides insights into the computational complexity of the filter and it's limitations as well as its scope of functionality. The Custom Folded Clausen Filter variants are Folded Normal modulated clause kernels of variable parameters which are inspired from the framework of a Gaussian Gabor filter which is defined to be Normal modulated sinusoidal kernels.

Due to faster performance and more insight into filter properties, the frequency domain analysis of these filters were also performed where the frequency and impulse responses of the filters were observed.

The parameters, design procedure and architecture; and the frequency domain analysis of these custom filters are provided in the subsequent sub-sections.

A. Filter Representation, Configurations, and Analysis

Filters were defined based on 7 configurations involving various parameters which are defined in Table III. These parameters were predefined to observe the functionalities of the filters based on these parameters as demonstrated in the "Experimental Observations" section. However, these parameters are not limited to have these predefined values and can be modified to improve robustness, vary performance levels and explore different characteristics of the filters.

TABLE III. FILTER CONFIGURATIONS AND PARAMETERS

Filter Psuedo Name	Type of Clausen Function	Parameters for Folded Normal	
		Mean (μ)	*Variance (σ)*
Filter Y	1st Order Standard Clausen Function	0	1
Filter X	2nd Order Sl-Type Clausen Function	0	1

FILTER XI	2ND ORDER SL-TYPE CLAUSEN FUNCTION	1	1
FILTER XII	2ND ORDER SL-TYPE CLAUSEN FUNCTION	0.5	1.5
FILTER XIII	2ND ORDER SL-TYPE CLAUSEN FUNCTION	0.5	2
FILTER XIV	2ND ORDER SL-TYPE CLAUSEN FUNCTION	2	1
FILTER XV	2ND ORDER SL-TYPE CLAUSEN FUNCTION	0	0.5

Filter X and Y are represented in 2-D kernel format in figure (6):

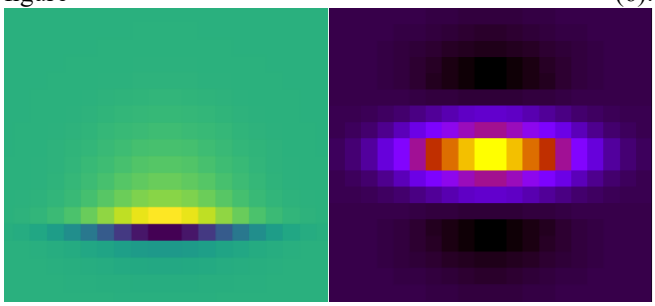

Figure 6. From right to left: (i) 2-D Filter X kernel (ii) 2-D Filter Y kernel.

The 3-D wireframe and surface representations of Filter X and Filter Y in comparison to a Gabor Filter in the spatial domain is shown in figure (7). Fig. 8 demonstrates the Group Delays of all custom Clausen Filters compared to that of the Gabor Filter's Group Delay. The Group delay is equal to the average time delay of the composite signal suffered at each component of frequency which is defined as $D(\omega)$ in (10):

$$D(\omega) \triangleq -\frac{d}{d\omega}\Theta(\omega) \tag{10}$$

In (10), $\Theta(\omega)$ represents the phase response of the filter which is covered in the subsequent section. This observation is crucial in understanding the delay/ time-shifting property observed in Filter X, which is also notable in figure (9) where the peaks of the filtered signal is right shifted in correspondence to the actual signal's peaks.

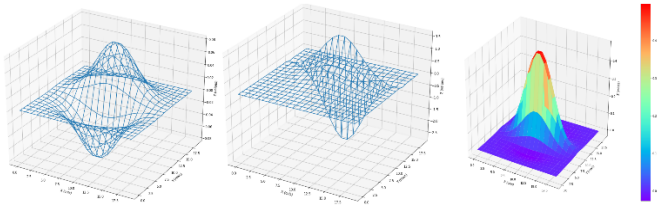

Figure 7. From right to left: (i) Wireframe 3-D plot of Gabor Kernel (ii) Wireframe 3-D plot of Filter X (iii) Surface 3-D plot of Filter Y.

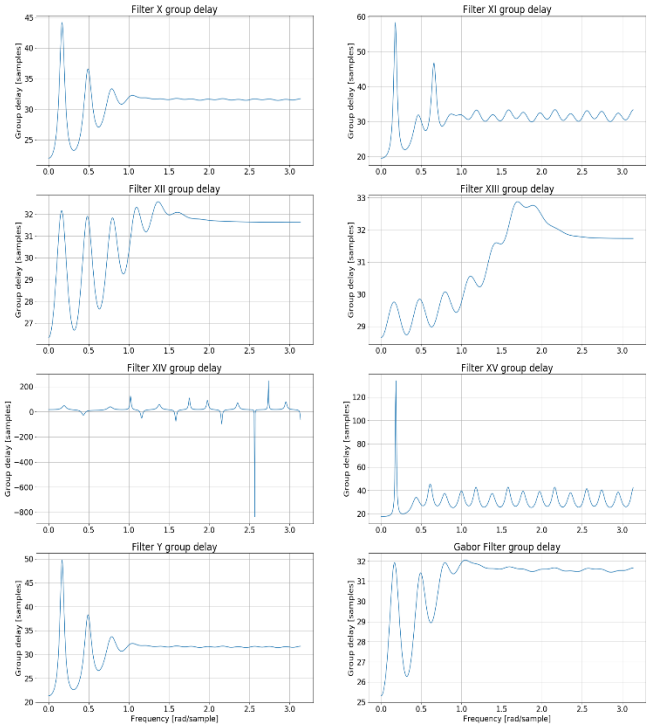

Figure 8. Group Delay Comparison of all Clausen Filters with Gabor Filter.

B. *Single Dimension Analysis*

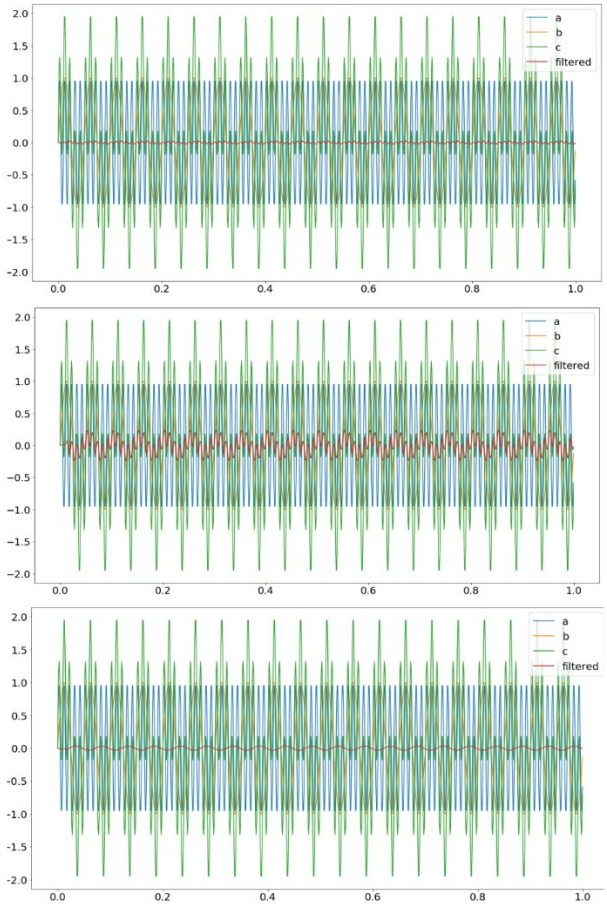

Figure 9. From top to bottom: (Summation of 2 Sinusoidal waves) – (i) Filter Y Result (ii) Filter X Result (iii) Gabor Filter Result.

Filter Designs are mostly first evaluated in a single dimension by applying the filter on random single dimensional signals like sinusoidal, square or sampled noisy

samples. This step also is conclusive to confirm the type of filter from being either a low-pass, high-pass or band-pass and in context of Image processing involves in establishing the inferences about the filter having various functionalities like Image Smoothening, Blurring, enhancing contrast levels or de-noising. Fig. 9 shows the filtering property of Filter X, Filter Y and Gabor Filter when applied on an input signal which is the summation of 2 sinusoidal signals.

Single Dimensional Analysis concluded that Filter Y and Gabor Filters have higher smoothening and truncating co-efficient rather than Filter X, where Filter X can be said having steeper transition factor.

C. *Fast Fourier Transform and Frequency Response Analysis*

Fast Fourier Transform (FFT) [23][24][25][26] is an essential algorithm in digital signal and image processing which computes the Discrete Fourier Transform (DFT) [27] or the Inverse Discrete Fourier Transform (IDFT) where the signal spatial representation is converted into one in the frequency domain or vice versa. Usually computational complexity is lower while analyzing signals in Frequency domain and FFT [24][27] is extremely fast at converting signal representation to frequency domain by factorizing the DFT matrix into a product of sparse factors which can result in enormous differences in run-time by reducing the complexity from $O(n^2)$ in terms of DFT computation to $O(nlogn)$ using FFT where n represents the size of the data. Thus, with higher data size, the runtime difference is extremely large. Fig.10. shows the 3-D surface plots of Gabor kernel, Filter Y and Filter X in the frequency domain.

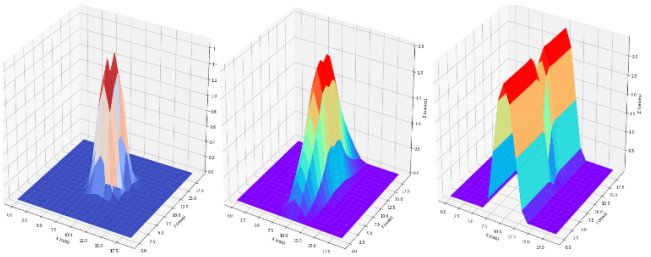

Figure 10. From right to left: (i) Surface 3-D plot of Gabor Kernel in Frequency Domain (ii) Surface 3-D plot of Filter Y in Frequency Domain (iii) Surface 3-D plot of Filter X in Frequency Domain.

Fig.11 demonstrates the 2-D FFT representations of all Clausen Based Filters compared to the FFT of the Gabor Kernel. Based on experimental results, the fastest way of convolution was also found out which was FFT based convolution rather than direct convolution. This also specifies the need for frequency domain analysis, since the process is much faster in the frequency domain.

For confirming the custom filters exhibiting the properties of standard low pass filters, the gain magnitude frequency response of the filters were observed and compared to that of a Gabor Filter. Generally, Low pass filters are used for Image blurring and smoothening. Frequency response is generally defined to be the relation between the filter output to the filter input in the Fourier domain. The Frequency response of a system is one of the conclusive observations to know the type of filter namely Low-pass, Band-pass or High-pass. Fig.12 shows the Frequency Response and Impulse Response graphs of all variants of SL-Clausen Filters (Filter X to Filter XV) and Filter Y compared to that of the responses observed in Gabor Filter Analysis. This demonstrates the clause filters being of non-linear phase type and also follows similar trends as observed in standard low-pass FIR filters.

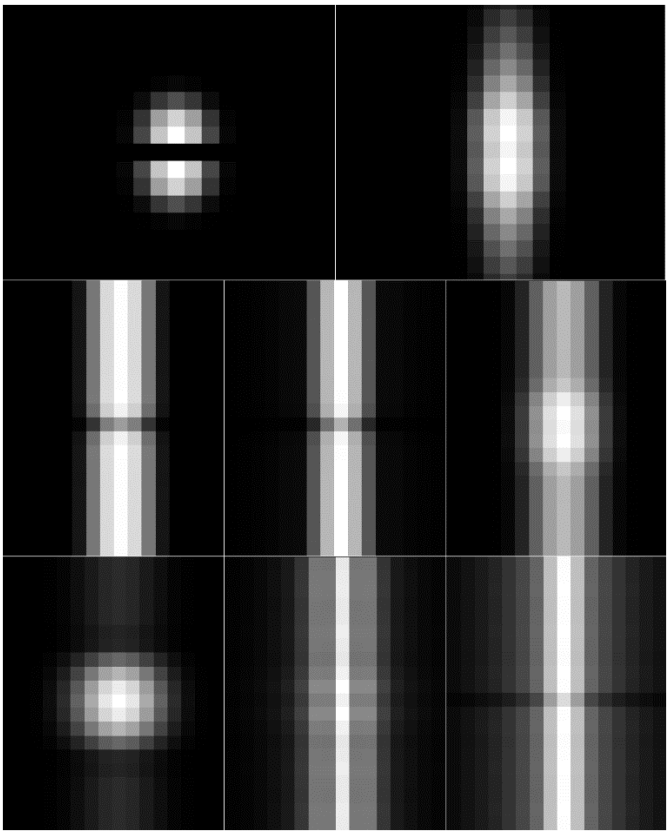

Figure 11. From right to left: 1st Row – (i) FFT of Gabor (ii) FFT of Filter Y. 2nd Row- (i) FFT of Filter X (ii) FFT of Filter XI (iii) FFT of Filter XII. 3rd Row- (i) FFT of Filter XIII (ii) FFT of Filter XIV (iii) FFT of Filter XV.

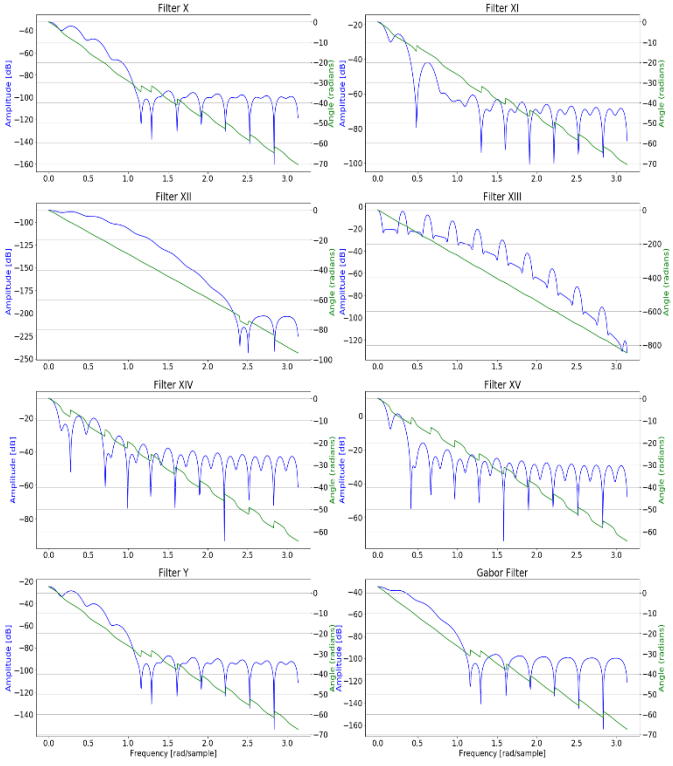

Figure 12. Frequency Response Comparison of Clausen and Gabor Filters.

VII. Experimental Observations

For analysis, two images of a varying resolution were primarily used to compute the comparative run-time study of the filters against a standard Gaussian Gabor filter. These images were chosen as such to test the robustness of the filters on varying image resolutions and pixel densities. All images used in figure (13) are used for official benchmarking purposes. All of these images used for the experiment were first converted to grayscale format on which the respective filters were applied.

Fig.13 shows Filter X being applied on 3 different images having resolutions (512×512), (512×512) and (400×600) all being in grayscale.

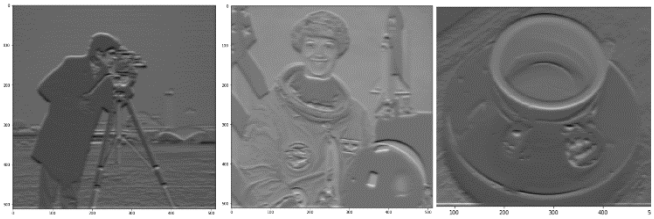

Figure 13. Filter X results on various images of varying resolutions. (Original Image Credits: Scikit-Image)

Fig. 14. shows the comparison between all filters (all variants of SL-type / X filters, Filter Y and Gabor Filter) applied on a standard grayscale 512×512 size image.

Fig.14 also shows the different properties exhibited by the various filters, with some having extensive blurring effect and smoothening effect [28] like Filter Y, Filter XII, Filter XIII; others like Filter XI, Filter XIV and Filter XV exhibiting low-level segmentation and abstract edge defining or texture analysis properties comparable to that of a standard Gaussian Gabor filter.

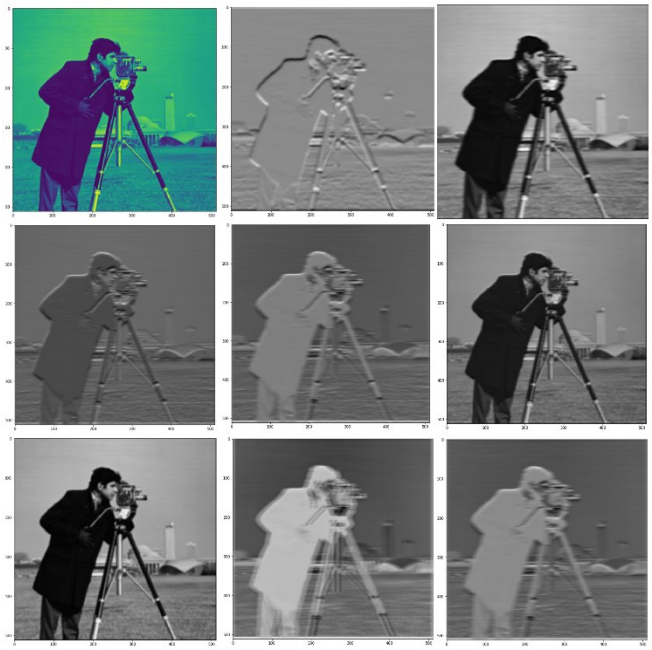

Figure 14. From right to left: 1st Row – (i) Original Image in a single channel (ii) Gabor Filter Result (iii) Filter Y Result. 2nd Row- (i) Filter X Result (ii) Filter XI Result (iii) Filter XII Result. 3rd Row- (i) Filter XIII Result (ii) Filter XIV Result (iii) Filter XV Result. (Original Image Credits: Scikit-Image)

Fig.15 shows the fluctuations in run-time in seconds across various iterations of all filters used in the experiment on the cloud environment compared to that of a standard Gaussian Gabor filter when applied on the same grayscale image having resolution 2021×2021 pixels. The specification of the cloud environment is provided in the following "Experimental Setup" section.

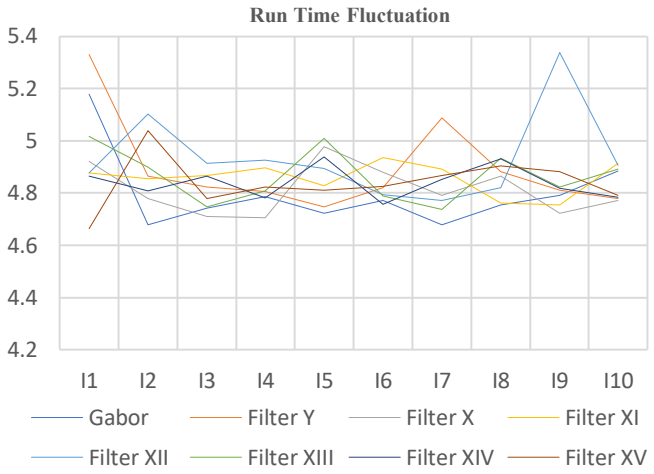

Figure 15. Run-Time fluctuation comparison of all filters on sample image across various iterations.

Fig. 15. shows Filter XV having the lowest run-time across all iteration equaling to 4.66482760000008 seconds comparable to Gabor's lowest run-time across all iterations equaling to 4.67 seconds approximately. Table IV is a run-time comparison across one iteration between all custom filters and a standard Gaussian Gabor filter when applied on two different grayscale test images having resolution 512×512 and 2021×2021 pixels while simulated on the cloud environment.

TABLE IV. RUNTIME VARIATIONS OF FILTERS (IN SECONDS)

FILTER NAME	**RUN-TIME (RESOLUTION - 512×512)**	**RUN-TIME (RESOLUTION - 2021×2021)**
GABOR	0.7231871	4.742698799999971
FILTER Y	0.31141730000000223	4.7652437000001555
FILTER X	0.721678	4.695399900000211
FILTER XI	0.31112860000001774	4.724164799999926
FILTER XII	0.3133619999999837	4.69758690000026
FILTER XIII	0.2977893999999992	4.775738300000057
FILTER XIV	0.3143409999998994	4.790427799999861
FILTER XV	0.31068419999996677	4.756758600000012

Table IV shows significant improvement of run-time for all custom filters when applied on the lower resolution image while for the higher resolution image, Filter X, Filter XI, and

Filter XII have better run-time than that of a Gabor Filter. This analysis proves the custom filters being near or better than Gabor Filters in terms of run-time performances.

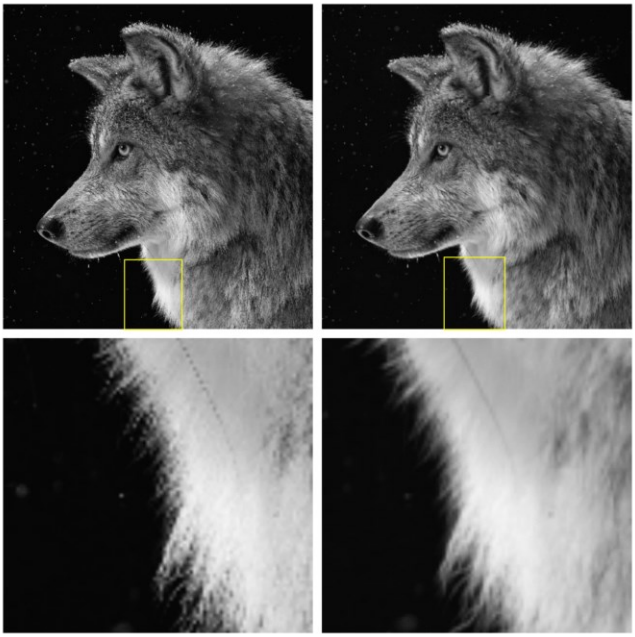

Figure 16. Original Sharp Image v/s Filter XIII applied Image. (Original Image Credits: National Geographic)

Fig.16 shows the blurring and smoothening property of Filter XIII while applied on a test input image. Fig.17 is the comparison between image quality [29][30] of the original image and Filter XIII applied image based on metrics like MSE (Mean Squared Error), PSNR (Peak Signal to Noise Ratio) [29][30] and SSIM (Structural Similarity Index) [20][30]. Due to smoothening or reduced sharpness in the image, the MSE value is quite high at 57632.79 while the PSNR value of around 20 dB does suggest lossy transformation however of acceptable quality. Thus, Filter XIII performs at par with other smoothening low pass filters used in image processing.

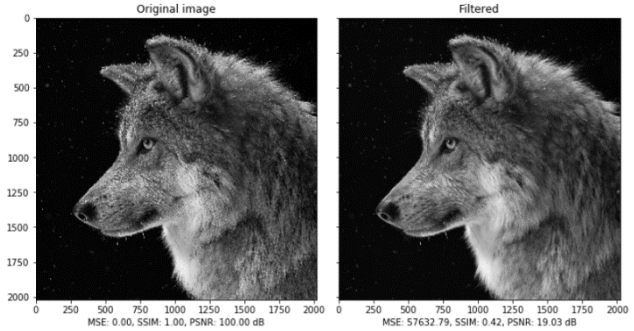

Figure 17. Image Quality Metrics of Original Image v/s Image Quality Metrics of Filter XIII applied Image. (Original Image Credits: National Geographic)

Table V provides the computational comparison analysis on a local GPU of the filters applied on a standard image of resolution (512×512). The details of the local GPU are provided in the "Experimental Setup" section. The comparison analysis provides constructive proof of the improved performance levels of the Custom Clausen Folded Filters as compared to that of Gaussian Gabor Filter. Filter X, XI, XIII, XIV and Y utilize half the amount of GPU and CPU RAM to that of Gaussian Gabor filter.

TABLE V. COMPUTATIONAL COST ANALYSIS

Filter Name	**GPU Temperature**	**GPU Utilization**	**CPU RAM Utilization**	**Average Run Time (Mean of 3 Runs)**
Gabor	63°c	14%	2.9GB	0.7978 sec
Filter X	66°c	7%	1.4GB	0.336 sec
Filter XI	61°c	7%	1.3GB	0.2691 sec
Filter XII	74°c	14%	3.0GB	0.2914 sec
Filter XIII	60°c	7%	1.3GB	0.5034 sec
Filter XIV	57°c	7%	1.4GB	0.7612 sec
Filter Y	57°c	7%	1.3GB	0.5858

VIII. EXPERIMENTAL SETUP

All of the experiments and simulations were conducted on a cloud environment – Google Colaboratory and a local system. The specification sheet of both environments is provided in Table VI. All of the software simulations were performed using Numpy, timeit, Matplotlib, OpenCV, Matplotlib, PSUtil, GPUtil, Humanize and Subprocess packages of Python 3 programming language.

TABLE VI. ENVIRONMENT SPECIFICATION SHEET

SPECIFICATION NAME	GOOGLE COLAB	LOCAL SYSTEM
CPU	INTEL XEON @ 2.30GHZ	INTEL CORE I7-8750H @ 2.20 GB
RAM	13GB	16GB
GPU	NVIDIA TESLA T4	NVIDIA GTX-1060
GPU MEMORY	16 GB	14GB

IX. CONCLUSION

The Custom Folded Clausen filters provided definitive insight into digital image filter designing and reducing computing redundancy while still maintaining and even outperforming State of the Art (SOTA) Gaussian Gabor Kernel. While Gaussian Gabor Kernel has limited functionalities of texture analysis and low-level abstract edge detection, the custom Clausen folded filters provide wide range functionalities based on modifying versions having variable parameters. The variants of Custom Folded Clausen filters have a wide range of functionalities including texture analysis, image blurring and smoothening, et cetera. These filters also require comparably less computational power than SOTA Gaussian Gabor and were also able to handle images of varying resolutions which is a statement of the robustness of their architectures.

Advancement of this research includes building more robust filters with improved performance in comparison to state of the art filters and more effective optimization of filter performance including introducing evolutionary algorithms to optimize filter parameters.

REFERENCES

[1] Hamming, Richard Wesley. Digital filters. Courier Corporation, 1998.

[2] Parks, T.W. and Burrus, C.S., 1987. Digital Filter Design. Topics in Digital Signal Processing (p. 226). John Wiley & Sons, New York.

[3] Baxes, Gregory A. Digital image processing: principles and applications. New York: Wiley, 1994.

[4] Schalkoff, Robert J. Digital image processing and computer vision. Vol. 286. New York: Wiley, 1989.

[5] Pitas, Ioannis. Digital image processing algorithms and applications. John Wiley & Sons, 2000.

[6] Butterworth, Stephen. "On the theory of filter amplifiers." Wireless Engineer 7, no. 6 (1930): 536-541.

[7] Kalman, Rudolph Emil. "A new approach to linear filtering and prediction problems." Journal of basic Engineering 82, no. 1 (1960): 35-45.

[8] Weinberg, L.O.U.I.S. and Slepian, P., 1960. Takahasi's results on Tchebycheff and Butterworth ladder networks. IRE Transactions on Circuit Theory, 7(2), pp.88-101.

[9] Mehrotra, Rajiv, Kameswara Rao Namuduri, and Nagarajan Ranganathan. "Gabor filter-based edge detection." Pattern recognition 25, no. 12 (1992): 1479-1494.

[10] Weldon, Thomas P., William E. Higgins, and Dennis F. Dunn. "Efficient Gabor filter design for texture segmentation." Pattern recognition 29, no. 12 (1996): 2005-2015.

[11] Bianconi, Francesco, and Antonio Fernández. "Evaluation of the effects of Gabor filter parameters on texture classification." Pattern recognition 40, no. 12 (2007): 3325-3335.

[12] Paris, Sylvain, and Frédo Durand. "A fast approximation of the bilateral filter using a signal processing approach." International journal of computer vision 81, no. 1 (2009): 24-52.

[13] Kong, Hui, Hatice Cinar Akakin, and Sanjay E. Sarma. "A generalized Laplacian of Gaussian filter for blob detection and its applications." IEEE transactions on cybernetics 43, no. 6 (2013): 1719-1733.

[14] Borwein, Jonathan Michael, David J. Broadhurst, and Joel Kamnitzer. "Central binomial sums, multiple Clausen values, and zeta values." Experimental Mathematics 10, no. 1 (2001): 25-34.

[15] Choi, Junesang, and Hari M. Srivastava. "The Clausen function ${\sf Cl} _2 (x) $ and its Related Integrals." Thai Journal of Mathematics 12, no. 2 (2014): 251-264.

[16] Carlitz, L., 1953. A theorem of Glaisher. Canadian Journal of Mathematics, 5, pp.306-316.

[17] Leone, F. C., L. S. Nelson, and R. B. Nottingham. "The folded normal distribution." Technometrics 3, no. 4 (1961): 543-550.

[18] Tufts, D., and J. Francis. "Designing digital low-pass filters--Comparison of some methods and criteria." IEEE Transactions on Audio and Electroacoustics 18, no. 4 (1970): 487-494.

[19] Rabiner, L. "Approximate design relationships for low-pass FIR digital filters." IEEE Transactions on Audio and Electroacoustics 21, no. 5 (1973): 456-460.

[20] McClellan, J., T. W. Parks, and L. Rabiner. “A computer program for designing optimum FIR linear phase digital filters.” IEEE Transactions on Audio and Electroacoustics 21, no. 6 (1973): 506-526

[21] Najjarzadeh, Meisam, and Ahmad Ayatollahi. "FIR digital filters design: particle swarm optimization utilizing LMS and minimax strategies." In 2008 IEEE International Symposium on Signal Processing and Information Technology, pp. 129-132. IEEE, 2008.

[22] Karaboga, Nurhan. “A new design method based on artificial bee colony algorithm for digital IIR filters.” Journal of the Franklin Institute 346, no. 4 (2009): 328-348.

[23] Brigham, E. Oran, and E. Oran Brigham. The fast Fourier transform and its applications. Vol. 448. Englewood Cliffs, NJ: prentice Hall, 1988.

[24] Van Loan, Charles. Computational frameworks for the fast Fourier transform. Vol. 10. Siam, 1992.

[25] Nussbaumer, H.J., 2012. Fast Fourier transform and convolution algorithms (Vol. 2). Springer Science & Business Media.

[26] Cochran, William T., James W. Cooley, David L. Favin, Howard D. Helms, Reginald A. Kaenel, William W. Lang, George C. Maling, David E. Nelson, Charles M. Rader, and Peter D. Welch. "What is the fast Fourier transform?." Proceedings of the IEEE 55, no. 10 (1967): 1664-1674.

[27] Winograd, Shmuel. "On computing the discrete Fourier transform." Mathematics of computation 32, no. 141 (1978): 175-199.

[28] Kaiser, J. F., and W. A. Reed. "Data smoothing using low - pass digital filters." Review of Scientific Instruments 48, no. 11 (1977): 1447-1457.

[29] Huynh-Thu, Quan, and Mohammed Ghanbari. "Scope of validity of PSNR in image/video quality assessment." Electronics letters 44, no. 13 (2008): 800-801.

[30] Hore, Alain, and Djemel Ziou. "Image quality metrics: PSNR vs. SSIM." In 2010 20th International Conference on Pattern Recognition, pp. 2366-2369. IEEE, 2010.