THE MATHEMATICAL FOUNDATION OF IMAGE PROCESSING USES THE QUANTUM QUATERNION FOURIER INVERSION TRANSFORM

Sandeep Bagul [1] Dr. Aarti Sharma [2]

[1] Sandeep Bagul, Assistant Professor, Late G. N. Sapkal College of Engineering, Nashik.

[2] Dr. Aarti Sharma, Associate Professor, Sage University, Indore.

sandeepbagul94@gmail.com

Abstract

The Fourier transform is a popular tool in many areas of practical mathematics, including data compression and signal processing. The underlying principle of the Fourier transform is that functions with the right characteristics may be expressed as a linear combination of trigonometric functions. Decomposing a signal using trigonometric functions is like taking a time-domain signal and separating its frequency-domain components; this opens up new and more efficient avenues for signal analysis and manipulation. An increasing number of people are looking at quantum image processing (QIP) as a way to enhance the efficiency of conventional methods and their applications by taking use of quantum computing's characteristics. Filtering and rectifying the frequency domain to get important picture information is the goal of both image design and interactive processing in this procedure. We demonstrate the operation of quantum operators and states in the context of quaternions and demonstrate how it may be used to expand the quantum complex Fourier transform (QCFT) to the quantum quaternion Fourier transform (QQFT). We demonstrate why convolution cannot be used to quantum image processing, and we create space-domain and quaternion-Fourier spectrum filters, including an edge detector and a quantum median filter. An enhanced method is used to provide the groundwork for the realization of picture design and interactive technology, drawing on the knowledge of FT and inverse transform.

Keywords: Fourier Transform, Function, Image Processing, Quantum Computing, Quaternion.

INTRODUCTION

When it comes to processing images, the Fourier transform is crucial. The classical Fourier transform is the second language for describing images; it is the basis of image processing and a way to translate images from the space domain to the frequency domain. It adds a new angle to the study of pictures and their relationship to the features of frequency distributions. The Fourier transform is a powerful tool for non-linear spatial filtering, among other things. The Fourier transform outperforms the spatial filter in terms of efficiency. We can also process certain picture frequencies from low-pass and high-pass with a high degree of accuracy using the Fourier transform for a big filter. When confronted with complicated and difficult-to-handle challenges, image processing frequently performs equivalent picture transformations by changing domains. The superposition of sins and cosines, which Fourier depicts as a function, is used extensively in the numerical and analytical solutions of differential equations, as well as in the analysis and processing of communication signals. The Fourier transform's usefulness lies in the fact that it can examine a signal's frequency content in the time domain. Because the Fourier coefficients of the converted function indicate the contribution of each sine and cosine function at each frequency, the signal may then be evaluated for its frequency content. As its name implies, an inverse Fourier transform shifts data from the frequency domain to the time domain [1].

An orthogonal trigonometric basis function is created by decomposing a signal using the Fourier transform. A continuous signal a(x) may be characterized by taking its Fourier transform and using the formula (1). For any temporal signal a(x), the global frequency distribution is given by the Fourier converted signal Aft(y). The inverse Fourier transform may be used to recreate the original signal, as seen in equation (2).

$$\mathrm{A_{yx}(y)} = \int_{-\infty}^{\infty} \mathrm{a\,(x)e^{-2\pi yx}\ dx} \tag{1}$$

$$\mathrm{a\,(x)} = \int_{-\infty}^{\infty} \mathrm{A_{yx}(y)e^{-2\pi yx}\ dy} \tag{2}$$

These equations allow one to convert between the time domain and the frequency domain for a signal a(x) [2].

FOURIER TRANSFORMS

The coefficients of a linear combination of trigonometric functions are decomposed using Fourier transformations, which break down a function. It is possible to express the Fourier transform of a function f(t) as [8]:

$$\hat{f}(\omega) := \int_{-\infty}^{\infty} f(t)e^{-i\omega t}dt, \tag{3}$$

The function f may be any function that depends on time t, and its Fourier transform, $\hat{f}$, is defined as f depending on frequency ω. The inverse Fourier transform that corresponds to this is

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \hat{f}(\omega)e^{i\omega t}d\omega \tag{4}$$

Discrete measurements can only be taken in limited sequences in practice. When dealing with finite sequences that are equally spaced, we use the discrete Fourier transform (DFT) as

$$Y_k = \sum_{n=0}^{N-1} X_n \cdot e^{\frac{-2\pi i}{N} kn}; \tag{5}$$

where Xn represents the n-th element in the original time sequence and Y_k represents the k-th element in the discrete Fourier transform vector. The time series has N components. Here we have the definition of the inverse discrete Fourier transform (IDFT), which is similar to Eq. (4):

$$X_n = \frac{1}{N} \sum_{k=1}^{N} Y_k \cdot e^{\frac{2\pi i}{N} kn}. \tag{6}$$

By combining even and odd functions when computing the DFT and its inverse transform, we may greatly improve computational efficiency. The computational complexity of the DFT is reduced from $O(N^2)$ to O(N logN) thanks to the fast Fourier transform (FFT) and its inverse fast Fourier transform (IFFT), which couple even and odd functions.

INTERACTIVE IMAGE DESIGN UTILIZING FOURIER INVERSE TRANSFORM

Definition 1. Image design and interactive technology rely on the Fourier transform (FT), a complex technique of analysis. The image's representation as a two-dimensional discrete data matrix based on the RGB color space makes its use of FT a part of the discrete Fourier transform (DFT). Start by defining the picture data as x = 0, 1,...,M −1 and y = 0,1,...,N −1. Then, determine its DFT [9].

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp\left[-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right]$$

Proof. Among them, a certain formula u = 0,1,...,M −1; v = 0,1,...,N −1

The inverse transformation that corresponds to this is:

$$f(x,y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v) \exp\left[-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)\right]$$

Among them, a certain formula x = 0,1,...,M −1, y = 0,1,...,N −1

ISSN 0315-3681 Volume 122, 2025

Fast algorithm of image Fourier transform

Compressing digital photographs scientifically within a certain time constraint is a massive undertaking because to the enormous amounts of data they contain. Direct application of the DFT technique to the implementation of picture signal processing is not feasible at the moment. The advent of the fast DFT algorithm in the early 1960s was the catalyst for the shift.

With the goal of promoting the Winograd Fourier Transform technique (WFTA) to partially replace the underlying 2-FFT technique and integrating the two together. The following are the exact procedures: As a first stage, the log 2n-4 steps of the base 2-FFT algorithm's decomposition process results in N/16 DFTs of 16 lengths when partitioned according to the base 2-FFT algorithm's decomposition process. Secondly, the WFTA technique may be used to analyze DFTs with a length of 16 because, in the base 2-FFT approach, the DFTs acquired by all iterations are independent between each group.

Given that the calculation process and algorithm concept are inversely related, one may combine the WFTA algorithm to finish the calculation, then combine the calculations according to the base 2-FFT method, and finally achieve the DFT of length N.

This operation process exemplifies the benefits of the WFTA algorithm in multiplication calculations and is both simple and effective. Figure 1 shows the particular flowchart, while Figure 2 shows the butterfly diagram at N = 32.

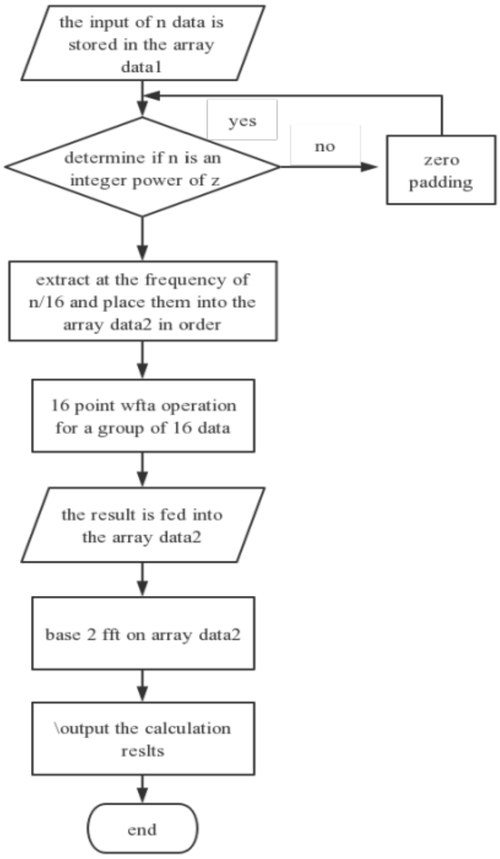

Figure 1: The enhanced algorithm's flowchart

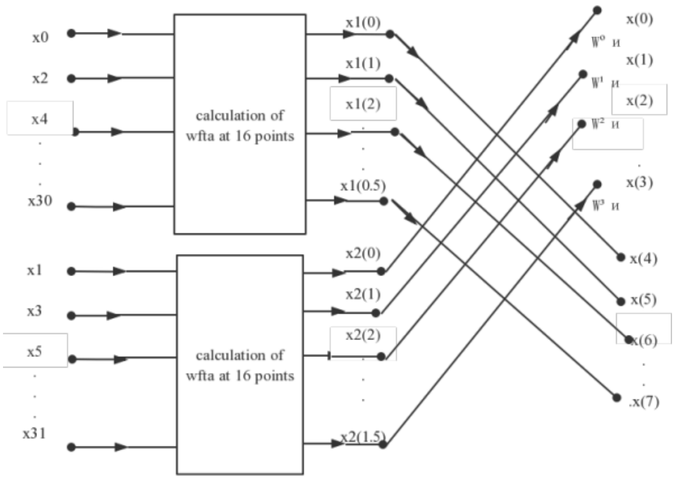

Figure 2: An enhanced 32-node butterfly graph

Figure 3 below shows the final result that is generated by combining the aforementioned steps with FT and inverse transform to finish picture design and interactive processing:

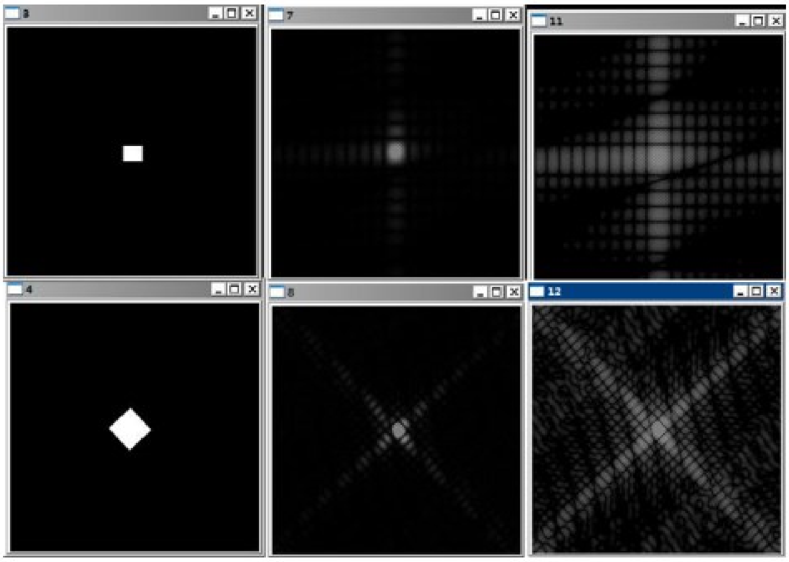

Figure 3: Design and interactive processing using FT and inverse transform

The inverse transform and fast Fourier transform are extensively used in image processing for tasks such as edge identification, picture enhancement, compression, and many more. With this technology, the spatial domain is transformed into the frequency domain using FT and inverse transform. Then, various operations are performed on the frequency domain, and the desired image processing results are obtained. This is the basic idea, at least from a practical standpoint. When it comes to improving and removing images, it's helpful to think of the image frequency spectrum as having two parts: the high frequency component, which represents the sharp edges of the image, and the low frequency component, which represents the smooth areas of the image, the contours. To create a new frequency function G (u, v), the augmentation and desiccating procedure uses various transfer functions H (u, v) to perform a convolution operation on the frequency function F (u, v). To eliminate picture noise, or surplus frequency signal, the new frequency function allows users to improve the stored frequency signal. Using inverse FT, the newly-created frequency function G (u, v) may get the revised frequency function G (u, v), which can both improve and remove the picture even more. Properly classifying the transfer function H (u, v) into high-pass, low-pass, and band-stop filters based on their respective functional differences is an essential part of this procedure. However, the noise signals often combine with the edge data. The high-frequency component of a spectral picture requires efficient processing, although edge detection works on the same concept as image enhancement. The conditions for edge detection can only be satisfied in this manner.

QUANTUM QUATERNION FOURIER TRANSFORM IMAGE PROCESSING

Not only has quantum computing been successful in various areas like phase estimation, quantum complex Fourier transform (QCFT), pattern recognition, classification, and quantum cryptography, but it can also speed up algorithms and shed light on how our minds work, making it an extremely important area of research. Notable scientist Richard Feynman admitted in 1982 that quantum computers may outperform traditional computers in terms of computational capability [10]. Quantum computing's remarkable features, such superposition, drastically cut down on space and temporal complexity. Thus, issues that classical computers aren't good at solving may be helped by quantum computers. The Fourier transform (FT) is among the most widely utilized algorithms in several ways within the fields of information theory and image processing. On the other hand, computing the discrete Fourier transform (DFT) takes a lot of time on computers. Fast Fourier transform (FFT), an algorithm introduced by Cooley and Tukey in 1965, allows a computer to execute the DFT effectively with a better computing cost of $O(N \log_2 N)$. Our work here involves expanding the QFT to include the QQFT, which stands for quantum quaternion Fourier transform. One new algorithm is the QQFT. An important distinction between classical and quantum fields theory (QFTs) is the quantum state into which the original data is transformed into probability amplitudes. The matching DFTs' amplitudes are changed. Traditional methods, on the other hand, take a whole vector with complex values and output its full DFT as a new vector of the same length.

Quaternion Algebra

H is a noncommutative and associative quaternion algebra. The components of a quaternion $q \in H$ are one real and three imaginary elements.

$$q = a + bi + cj + dk \quad a, b, c, d \in R. \tag{7}$$

The relationships are obeyed by the units i, j, and k.

$$i^2 = j^2 = -1, \; ij = k.$$

A polar plot of q looks like

$$q= |q|e^{i\phi}e^{k\psi}e^{j\theta}, \tag{8}$$

When the three quaternionic phases are represented by the angles (ϕ, θ, ψ) and the equation $|q| = \sqrt{q\bar{q}}$ holds. Here, $\bar{q}$ is the conjugate of $q = a - bi - cb - dk$. The equations $qa = as + a = a + axi + byb + azk$ and $qb = b + b = b + bxi + byb + bzk$ are given by the two quaternions. The quaternion product $\otimes$ is given by where the variables a, ax, ay, az, b, bx, by, bz $\in$ R are all real numbers.

$$qc = c + c = qa \otimes qb = (ab - a \cdot b)+(ab + ba + a \times b), \tag{9}$$

where the vectors' inner and cross products are represented by $\cdot$ and $\times$, respectively.

QQFT

Based on the above formula, the DFT transfers a vector {x1, x2, ..., xN−1} to a vector {y1, y2, ..., yN−1}.

$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j w_N^{ik}, \tag{10}$$

Where $w_N^{ik} = e^{-2\pi i \frac{jk}{N}}$.

Quantum Quaternion Fast Fourier Transform (QQFFT) Method Implementation

The quaternion state of the RGB picture at the point (i, b) with i = 1, ..., M and a = 1, ..., L may be represented as the quaternion r(i, b)i + g(i, b)j + b(i, b)k, given a 2D RGB image F = (Fi, b)M×L, Fi, b. This study has chosen a quantum representation model that transforms the M × L matrix F into a column vector of state quaternions with a dimension of ML × 1. The process is carried out by making the first M elements of f the first column of F, the following M elements the second column, and so on.

$$U_{QQFT} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & w_n & w_n^2 & w_n^3 & \dots & w_n^{N-1} \\ 1 & w_n^2 & w_n^4 & w_n^6 & \dots & w_n^{2(N-1)} \\ 1 & w_n^3 & w_n^6 & w_n^9 & \dots & w_n^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & w_n^{N-1} & w_n^{2(N-1)} & w_n^{3(N-1)} & \dots & w_n^{(N-1)(N-1)} \end{bmatrix}, \tag{11}$$

Where

$w_n = e^{2\pi\mu/2n}$ For the UIQQFT, we use $w^{ik}{}_N = e^{-2\pi\mu\, jk/N}$ instead.

$$f = [F1{,}1F2{,}1 \dots FM{,}1F1{,}2 \dots Fi{,}j \dots FM{,}L]^T \tag{12}$$

One may convert the quaternion vector into a quantum state of n qubits that is pure quaternions, that is,

$$|f> = \sum_{k=0}^{2^n-1} c_k |k>, \tag{13}$$

where n = [log2(ML)] and |k > represents the (i, j) position for every pixel. where ck is the value of the quaternion pixel and

$$c_k = \begin{cases} \frac{F_{i,j}}{\sqrt{\sum_{k=0}^{2^n-1} F_{i,j}^2}}, & k < ML \\ 0 & k \geq ML \end{cases} \tag{14}$$

That the final pixel value quaternion quantum state is normalized is why this step is so crucial. This method obtains normalized quaternion states from the final quaternion quantum pictures, which store information about colors and their locations in the images. The unique situation when ML = N = 2n is taken into account in our study. The image's Fourier transform follows a straight line. Here is how this linear translation looks in a quantum setting: In Equation (11), we get the unitary operator UQQFT, which is defined as the product of the input image quaternion state $|f >$ and the output image state $|g >= \text{UQQFT}|f>$.

Quaternion Quantum Image

The following is the representation of each pixel as a pure quaternion given an RGB f (x, y) image:

$$q(x, y)=(0, \vec{q}) = R(x, y)i + G(x, y)j + B(x, y)k. \tag{15}$$

Nevertheless, the following exponential function is what we recommend instead:

$$q(x, y) = e\theta n = \cos(\theta) + n \sin(\theta), \tag{16}$$

where

$$\boldsymbol{n} = \frac{R(x, y)\boldsymbol{i} + G(x, y)\boldsymbol{j} + B(x, y)\boldsymbol{k}}{\sqrt{R(x, y)^2 + G(x, y)^2 + B(x, y)^2}}$$

$$\theta = \sqrt{R(x, y)^2 + G(x, y)^2 + B(x, y)^2}.$$

Two qubits, which superimpose four quantum states in the following way, represent an RGB picture pixel x in a quantum quaternion:

$$x = \alpha 1|00 > +\alpha 2|01 > +\alpha 3|10 > +\alpha 4|11 >, \tag{17}$$

Where for

$$|\boldsymbol{n}| = \sqrt{R(x, y)^2 + G(x, y)^2 + B(x, y)^2},$$

$$\alpha_1 = \cos(\theta), \quad \alpha_2 = \frac{\sin(\theta)}{|\boldsymbol{n}|}R(x, y)\boldsymbol{i}, \quad \alpha_3 = \frac{\sin(\theta)}{|\boldsymbol{n}|}G(x, y)\boldsymbol{j}, \quad \alpha_4 = \frac{\sin(\theta)}{|\boldsymbol{n}|}B(x, y)\boldsymbol{k}, \tag{18}$$

Where

$$\sum_{n=1}^{4} \alpha_n^2 = 1.$$

Two quaternion quantum pictures of RGB colors are shown in Figure 4.

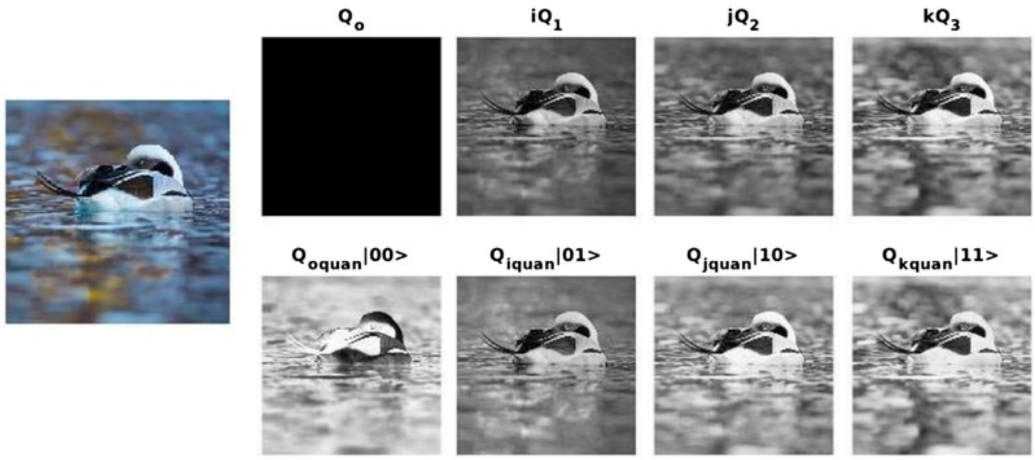

Figure 4: Images in the RGB color space shown as quantum quaternions

Quantum Image Processing

No quantum mechanical apparatus has yet been able to calculate the correlation and convolution between quantum states. Therefore, neither the spatial nor the frequency domains are suitable for applying the traditional convolution to the task of picture filtering. A theory put out by Chris Lomont [11] establishes that quantum convolution is impossible. In order to extract an image's low and high frequencies, we provide filters that pick out a particular region of the QQFFT (Figure 5).

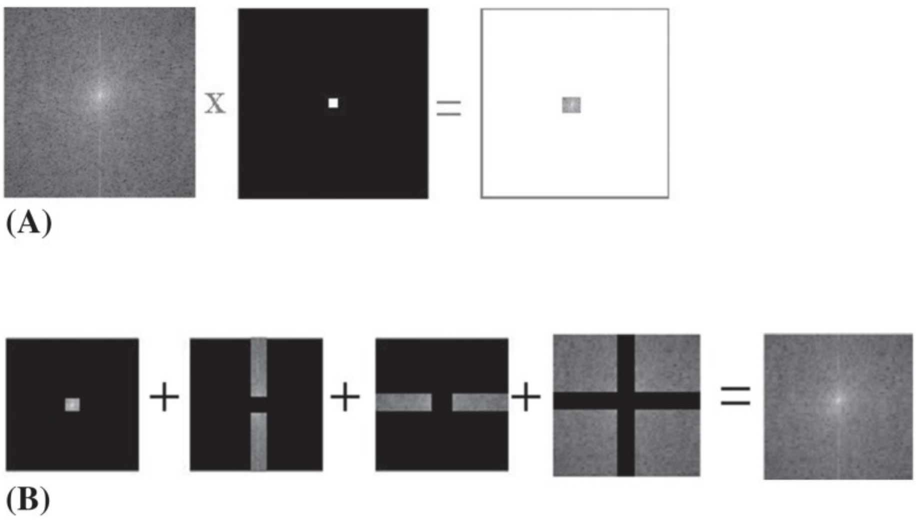

Figure 5: (A) frequency-domain box filter, and (B) spectrum decomposition using a series of filters in the same domain.

Quantum edge detection

By calculating the gradients of the detected edge (Eg1), the real edge (Eg2), and the potential location (Eg3), one may tell that the detected edge is quite near to the actual edge. The following is the procedure for choosing the edge's orientation angle:

$$
\begin{aligned}
0^\circ &\rightarrow Eg_2 > Eg_3 \& Eg_3(i,j) > Eg_3(i-1,j) \& Eg_3(i,j) > Eg_3(i+1,j) \\
45^\circ &\rightarrow Eg_2 > Eg_3 \& Eg_3(i,j) > Eg_3(i+1,j-1) \& Eg_3(i,j) > Eg_3(i-1,j+1) \\
90^\circ &\rightarrow Eg_2 > Eg_3 \& Eg_3(i,j) > Eg_3(i,j-1) \& Eg_3(i,j) > Eg_3(i,j+1) \\
135^\circ &\rightarrow Eg_2 > Eg_3 \& Eg_3(i,j) > Eg_3(i-1,j-1) \& Eg_3(i,j) > Eg_3(i+1,j+1).
\end{aligned} \quad (19)
$$

It is our attempt to suggest a new algorithm for the QQFT. We describe why convolution cannot be used for quantum image processing and demonstrate how to construct suitable filters for such processing in the quaternion Fourier spectrum.

CONCLUSION

We can sum up, the most basic definition of an image is a representation of something, whether it be a color, an item, or a scene captured by photography or painting. To someone like you and me, it's a piece of cake to spot comparable or dissimilar items in a picture; for example, a flower or an animal might be easily identified. Even though it's easy for a human to see what's in a picture, it can be more challenging for a machine to do it without human intervention. This is especially true when there aren't obvious labels or headings in the image, but we still need to be able to detect and recognize colors, objects, or similar things. The FT and inverse transform provide a fresh perspective on the state of the art in computer image processing, allowing for the rapid realization of visual designs and the full realization of interactive technologies. You may think of the spatial domain picture as a superimposition of sinusoidal curves that are uncorrelated in most frequency domains; by continuously varying the actual amplitude value, you can easily display the appropriate image information. For classification, segmentation, and time-variant signals like video and tracking, we suggest methods that use the QQFT and quaternion quantum filters to extract states in the quantum space. These quaternion qubits are then fed into diverse quantum quaternion convolutional neural networks (CNNs).

REFERENCES

1. Bracewell, R.N., (2001), The Fourier Transform and Its Applications. third edition, McGraw Hill Book Co., New York, 1999.

2. Praveen Shrivastava, Arvind Gupta and S. K. (2013) Vijay, Fourier Transform and Application in Image Processing. J. Comp. & math. Sci. Vol.4 (1), 1-3.

3. Bayro-Corrachono, Eduardo & Vazquez-Flores, Zuleima. (2023). Image processing using the quantum quaternion Fourier transform. Mathematical Methods in the Applied Sciences. 47. 10.1002/mma.9124.

4. Haque, Mohammad & Uddin, Mohammad Shorif. (2011). Accelerating Fast Fourier Transformation for Image Processing using Graphics Processing

Unit. Journal of Emerging Trends in Computing and Information Sciences. 2. 367-375.

5. Al-Ta'ani, Ola & Alqudah, Ali & Al-bzoor, Manal. (2019). Implementation and Analysis of Quantum Fourier Transform in Image Processing. 5. 11-26.

6. Sahnoun, Khaled & Noureddine, Benabadji. (2014). The Fourier Transform for Satellite Image Compression. Computer Science & Information Technology. 4. 119-125. 10.5121/csit.2014.4112.

7. Lu, Shaojun & Abozinadah, Ehab & Erkec, Elif. (2021). Image design and interaction technology based on Fourier inverse transform. Applied Mathematics and Nonlinear Sciences. 7. 10.2478/amns.2021.1.00061.

8. Michael T Heideman, Don H Johnson, and C Sidney Burrus. (1985). Gauss and the history of the fast Fourier transform. In: Archive for history of exact sciences 34.3, pp. 265–277.

9. Yongfeng Ren, Hao Man Wu, Storage, etc. (2019). 2 D FFT image edge enhancement based on FPGA [J]. Journal of Graphics, 40 (01): 139-144

10. Feynman, Richard (1982). Simulating Physics with Computers. *International Journal of Theoretical Physics*. 21 (6–7): 467–488.

11. C. Lomont, (2003). arXiv:quant-ph/0309070v23.