Research Article

Quantum Meets SAR: A Novel Range-Doppler Algorithm for Next-Gen Earth Observation

Ali Ghandour[1]

1. CNRS-L, National Center for Remote Sensing, Lebanon

Synthetic aperture radar (SAR) data processing is crucial for high-resolution Earth observation and remote sensing applications, one of the most commonly used algorithms for this task is the Range Doppler Algorithm (RDA). Using the Fast Fourier Transform (FFT), the collected signal is transformed to the frequency domain and then goes through the processing steps of this algorithm. However, when it comes to large datasets, this process can be computationally expensive. This paper explores the implementation of a Quantum Range Doppler Algorithm (QRDA), relying on the Quantum Fourier Transform (QFT) as a speedup tool over the classical FFT. Additionally, it proposes a quantum version of the Range Cell Migration Correction (RCMC) in the Fourier domain, one of the key correctional steps of the RDA algorithm, and compares it with its classical counterpart.

Khalil Al Salahat and Mohamad El Moussawi equally contributed to this work.

Corresponding author: Ali J. Ghandour, aghandour@cnrs.edu.lb

I. Introduction

Synthetic Aperture Radar (SAR) is a powerful active imaging modality widely used in remote sensing due to its ability to operate independently of weather conditions and ambient lighting. In a typical SAR system, an airborne or spaceborne platform emits microwave pulses toward a target area and records the backscattered echoes. The raw data collected in this way encodes information about the scene's reflectivity properties, and transforming it into a usable image requires sophisticated processing to account for the radar's motion and the geometry of wave propagation. The **Range Doppler Algorithm (RDA)** is one of the most widely employed methods for SAR image formation. Its computational core

 doi.org/10.32388/8Q5OW6

relies on the Fast Fourier Transform (FFT) to convert the data into the frequency domain, where key operations such as range compression, **Range Cell Migration Correction (RCMC)**, and azimuth compression are performed. However, the FFT's $O(n \log n)$ complexity—while efficient for moderate datasets—becomes a bottleneck for large-scale or high-resolution SAR applications, where real-time processing is often desirable. In this work, we explore a **quantum computing** approach to accelerate the RDA by replacing classical FFT-based steps with their quantum counterparts. The **Quantum Fourier Transform (QFT)** offers a theoretical exponential speedup over the FFT, but this advantage is contingent on maintaining quantum coherence throughout the entire processing pipeline. Measuring intermediate results (e.g., after **QFT** but before **RCMC**) collapses the quantum state, negating the speedup. Thus, a practical quantum RDA must integrate all critical steps—including **RCMC**—within the quantum domain to avoid decoherence and maximize computational gains.[1] To address this challenge, we propose two key innovations (Fig. 1):

1. Efficient quantum encoding
2. A quantum-domain **RCMC** implementation

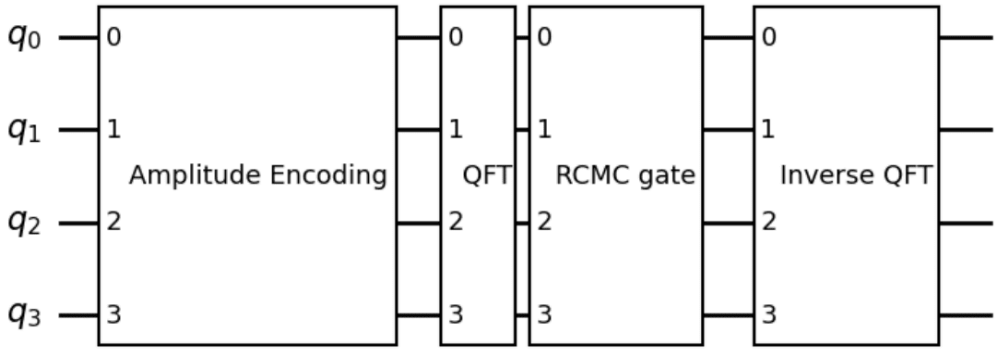

Figure 1. Quantum circuit approach for the Data Encoding and the RCMC gate implementation.

II. Fundamentals of Quantum Computing

A. Classical vs. Quantum Information

Classical computation relies on bits as its fundamental unit of information, where each bit represents a binary state (0 or 1). For a system with m distinct states, the minimum number of bits required is

doi.org/10.32388/8Q5OW6

$n = \lceil \log_2 m \rceil$. In contrast, quantum computation takes advantage of the qubit (quantum bit), which generalizes the classical bit by leveraging superposition and entanglement.[2]

B. The Qubit and Superposition

A qubit's state $|\psi\rangle$ is represented as a linear combination of basis states $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \text{ where } |\alpha|^2 + |\beta|^2 = 1$$

where $|\alpha|^2$ and $|\beta|^2$ tell us the probability of finding $|\psi\rangle$ in the states $|0\rangle$ and $|1\rangle$. When a qubit is measured, it will only be found to be in the state $|0\rangle$ or the state $|1\rangle$. This quality of superposition is at the core of **quantum computing**, allowing us to tap into more possibilities with a single qubit, along with giving us exponential scaling, since n qubits can represent 2^n states simultaneously, compared to classical bits, which can only store one state at a time.[2]

C. Quantum States and Basis Representations

A quantum state $|\psi\rangle$ can be written as a linear combination of a basis set $|v_i\rangle$ with complex coefficients of expansion c_i as:

$$|\psi\rangle = \sum_{i=1}^{n} c_i|v_i\rangle = c_1|v_1\rangle + c_2|v_2\rangle + \ldots + c_n|v_n\rangle$$

with $\sum_i |c_i|^2 = 1$. The modulus squared of a given coefficient c_i gives the probability that measurement finds the system in the state $|v_i\rangle$.[2]

D. Quantum Operators and Unitarity

Quantum operators are linear transformations that act on states. An operator $\hat{A}$ maps $|\psi\rangle$ to another state $\hat{A}|\psi\rangle = |\phi\rangle$. For quantum computation, operators must be unitary, satisfying $U^\dagger U = I$ ensuring reversibility and probability conservation.[2]

E. Key Quantum Gates

Quantum gates manipulate qubit states analogously to classical logic gates but with additional capabilities (e.g., phase shifts, superposition).[2] Two critical single-qubit gates are:

- Hadamard gate (H): Creates superposition from computational basis states:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \text{ where } H|0\rangle = |+\rangle, H|1\rangle = |-\rangle \quad (1)$$

doi.org/10.32388/8Q5OW6

with $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$

- Phase gate (P): Introduces a relative phase θ to $|1\rangle$:

$$P = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{pmatrix}, \text{ with } P|0\rangle = |0\rangle, P|1\rangle = e^{i\theta}|1\rangle \tag{2}$$

III. Range Doppler Algorithm

In SAR imaging systems, microwave pulses are transmitted from an airborne or spaceborne platform towards the target area. The backscattered echoes are collected and sampled, producing a two-dimensional raw signal $s(\tau, \eta)$ where τ represents the range (fast-time) dimension and η denotes the azimuth (slow-time) dimension. The **Range Doppler Algorithm (RDA)** processes these data into a focused image through sequential range and azimuth compression, leveraging Fourier-domain transformations.The algorithm achieves this through efficient utilization of Fast Fourier Transforms (FFTs).

A. Classical RDA

The classical RDA consists of four stages[3][4] (Fig. 2)

1. **Range Compression:** The raw signal undergoes range compression through frequency-domain matched filtering:

   $s_{rc}(\tau, \eta) = \text{IFFT}_\tau \left[\text{FFT}_\tau [s(\tau, \eta)] \cdot G(f_\tau) \right]$

   where $G(f_\tau)$ is the range reference function. This operation collapses all targets with identical slant ranges into single trajectories while preserving phase information.

2. **Azimuth FFT:** The range-compressed signal is transformed to the azimuth frequency domain:

   $s_1(\tau, f_\eta) = \text{FFT}_\eta [s_{rc}(\tau, \eta)]$

3. **Range Cell Migration Correction (RCMC):** A phase correction compensates for slant-range variations caused by platform motion:

   $$G_{\text{RCMC}}(f_r) = \exp\left[4i\pi \frac{f_r}{c} \left(R_0 \left(\frac{1}{D(f_\eta, V)} - 1 \right) \right) \right] \tag{3}$$

   where $D(f_\eta, V) = \sqrt{1 - (\lambda f_\eta / 2V)^2}$ is the Doppler compression factor, V is the platform velocity, and R_0 is the reference range. This step aligns target responses from curved trajectories to straight lines.

4. **Azimuth Compression:** Final focusing is achieved through azimuth matched filtering and transformation back to the time domain:

$s_{\text{ac}}(\tau, \eta) = \text{IFFT}_{\eta}\left[s_2(\tau, f_\eta) \cdot H(f_\eta)\right]$

The matched filter is given by: $H(f_a) = e^{-j\phi(f_a)}$. where $\phi(f_a)$ is the phase history of the target in the azimuth frequency domain.

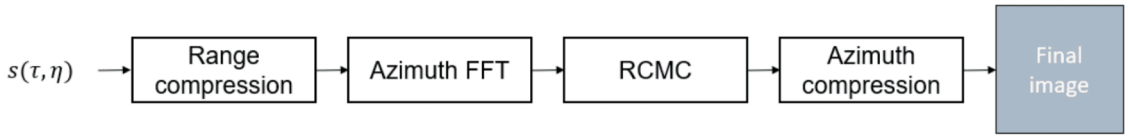

Figure 2. Range Doppler Algorithm: block diagram.

B. Quantum RDA

Building upon the classical RDA framework, the proposed quantum implementation replaces FFT operations with the **Quantum Fourier Transform (QFT)**, offering an exponential speedup in theory. This also means that the whole algorithm from the encoding to the measurement must be done in the quantum domain to actually achieve this speedup[1]. (Fig. 3)

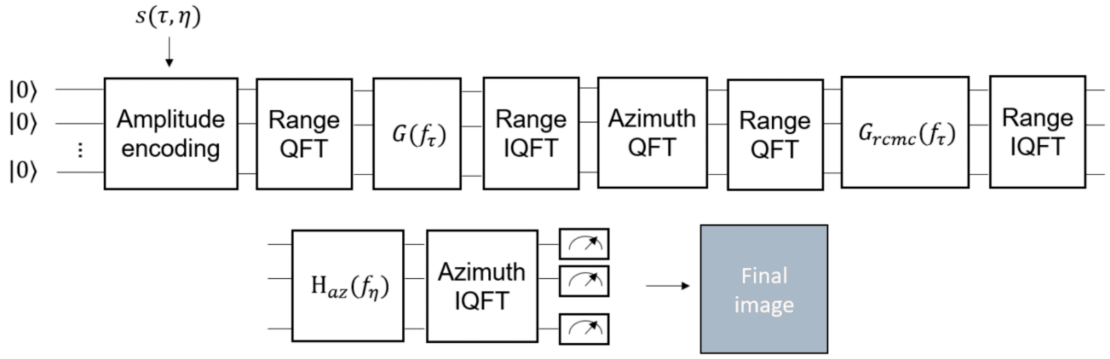

Figure 3. Quantum Range Doppler Algorithm: proposed quantum circuit approach.

IV. Amplitude Encoding

The first step in our algorithm is the encoding step, for this we have employed a method known as amplitude encoding. This technique embeds the information into the probability amplitudes of quantum states, this is done by first normalizing our dataset then initializing these values into our amplitudes. In our case, the encoded information consists of the radar data $s(\tau, f_\eta)$ samples, complex numbers which

doi.org/10.32388/8Q5OW6

contain both norm and phase data for each range and azimuth sample. Amplitude encoding is especially powerful because it allows us to encode N features with only $\log_2(N)$ qubits, for example, a 16x16 image requires only 16 qubits, and so on. Given a classical vector $\mathbf{x} = (x_0, x_1, \ldots, x_{N-1})$ representing our data, we encode it into a normalized quantum state:

$$|\psi\rangle = \sum_{i=0}^{N-1} x_i |i\rangle$$

where: $|i\rangle$ are computational basis states of an n-qubit system (for $N = 2^n$). The coefficients x_i are the amplitudes that encode the classical data. The vector must be normalized, meaning $\sum_i |x_i|^2 = 1$.[5][6]

V. Quantum Fourier Transform

The **Quantum Fourier Transform (QFT)** is the quantum counterpart of the classical Discrete Fourier Transform (DFT). For $N = 2^n$ samples, the **QFT** achieves an exponential speedup over the Fast Fourier Transform (FFT) —reducing complexity from $O(n \log n)$ to $O(\log^2 n)$ by exploiting quantum superposition and entanglement. The **QFT** transforms a computational basis state $|x\rangle$ (where $x \in \{0, 1, \ldots, N-1\}$) into a superposition of Fourier basis states

$$\mathrm{QFT}|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k / N} |k\rangle \tag{4}$$

where k represents frequency components for an n-qubit system.[7]

A. Circuit Implementation

The **QFT circuit** is constructed recursively using:

1. Hadamard gates (H) (eq. 1)
2. Controlled phase rotations to encode frequency-dependent phases: $R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix}$

The structure of the circuit reflects a recursive decomposition of the Fourier transform, with each qubit undergoing a Hadamard gate followed by progressively finer phase rotations conditioned on higher-order qubits (Fig. 4)

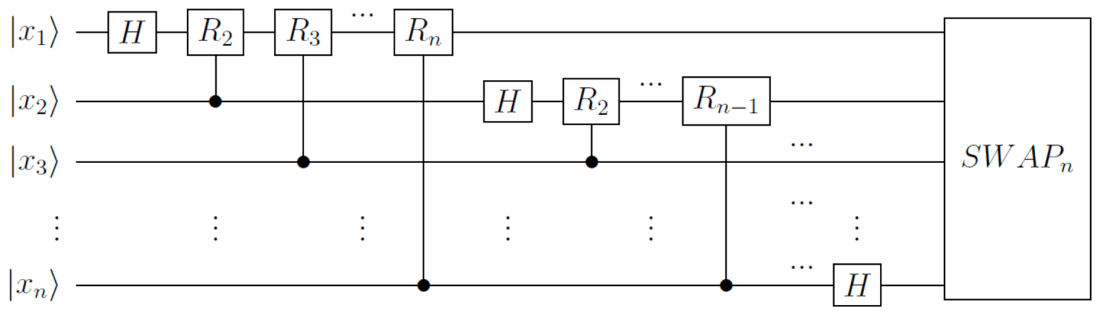

Figure 4. Quantum Fourier Transform Gate.[8]

VI. Quantum RCMC

A. Theoretical Foundation

Seeing that the **RCMC** filter is a correctional phase shift that realigns received signals—shifted due to a moving target or radar—our aim will be to create a quantum gate which implements the phase shift coefficients that we have calculated classically. The **RCMC** filter formula is (eq. 3) meaning it is an array of phase elements which will filter the main radar data once multiplied (each **RCMC** element needs to be multiplied by every range line in our data)

B. Quantum Gate Implementation

To implement this in our quantum circuit we will have to create a gate that applies the corresponding phase shift to each of our amplitude encoded data respectively. Being a phase-only array, implementing the filter into a gate that acts on the quantum states will not alter any of the probability amplitudes, in fact, making a diagonal matrix out of the **RCMC** elements (duplicated because each element corresponds to a range line and not a single sample) will give us a reversible unitary gate, which is exactly what we need, each element on the diagonal will be multiplied by the phase of the corresponding state, So we implemented the **RCMC** as a diagonal unitary operation:

$$U_{\mathrm{RCMC}} = \bigoplus_{k=1}^{N_r} e^{i\Theta_k} \otimes I_{N_a} \tag{5}$$

where Θ_k contains the phase corrections for the k -th range bin, and I_{N_a} is the identity on azimuth qubits.

doi.org/10.32388/8Q5OW6

C. Circuit Realization

For a minimal 2x2 example (2 range bins x 2 azimuth samples), (Fig. 1) the **RCMC** operator takes the form:

$$U_{\mathrm{RCMC}} = \begin{pmatrix} e^{i\theta_1} & 0 & 0 & 0 \\ 0 & e^{i\theta_1} & 0 & 0 \\ 0 & 0 & e^{i\theta_2} & 0 \\ 0 & 0 & 0 & e^{i\theta_2} \end{pmatrix} \tag{6}$$

When applied to a state $|\psi\rangle = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^T$ it yields:

$$U_{\mathrm{RCMC}}|\psi\rangle = \begin{bmatrix} e^{\theta_1}\alpha_1 \\ e^{\theta_1}\alpha_2 \\ e^{\theta_2}\alpha_3 \\ e^{\theta_2}\alpha_4 \end{bmatrix}$$

Where α_i is the probability amplitude of each state $|i\rangle$, containing the radar phase data, and e^{α_k} is the **RCMC** filter element acting on each range line k (containing 2 samples in this case)

D. Results

We verified our quantum **RCMC** implementation using Qiskit's statevector simulator (AerSimulator), which emulates an ideal, noise-free quantum computer. We compared the results with classical processing[9] on Sentinel-1 SAR data (64x64 subset) (Fig. 5). The phase differences between classical and quantum **RCMC** processing are on the order of 10^{-16} consistent with 64-bit floating-point precision limits. This test confirms the correctness of the quantum approach.

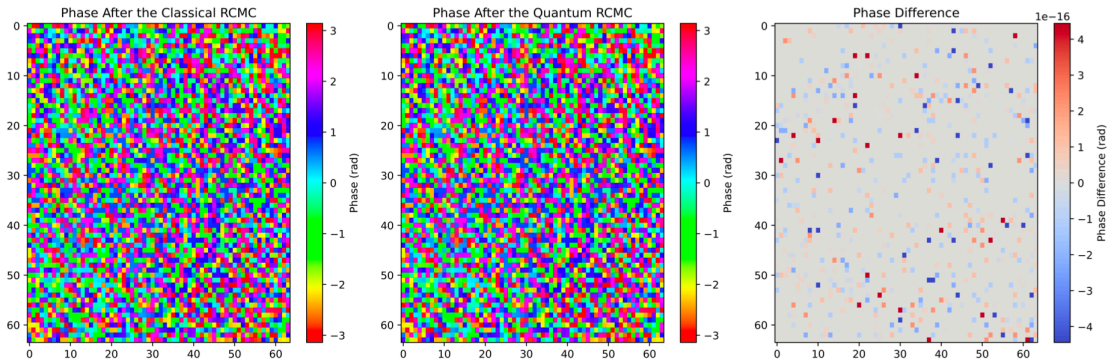

Figure 5. The Phase Value Difference of the RDA Classical and Quantum approaches.

doi.org/10.32388/8Q5OW6

VII. Conclusion

This work demonstrates the feasibility of quantum acceleration for **Earth observation** applications. Our implementation maintains mathematical equivalence to classical RDA while being fully executable on quantum hardware, as demonstrated through simulations of a 64x64 Sentinel-1 SAR subset.

Future work will focus on developing a complete, practical quantum RDA that integrates all processing steps within the quantum domain, as achieving a true speedup requires the successful implementation of the entire pipeline. Additionally, future advancements in quantum sensors for satellites could eliminate the need for data encoding, further enhancing speed and efficiency.

At the current stage of Noisy Intermediate-Scale Quantum (NISQ) devices, the process remains constrained by hardware limitations. Moreover, we have not yet accounted for errors and noise inherent in real quantum computers, which will be a critical consideration as the field progresses.

References

1. ^[a] ^[b] *European Space Agency. "Quantum Computing for Earth Observation (QC4EO): Use Case Definition and Design Report". REFERENCE: D1:QC4EO Study 1, 18/11/2023, ISSUE: 4, [Online]. Available: https://eo4society.esa.int/wp-content/uploads/2024/01/QC4EO-WP1-D1-Use-case-definition-and-design-report.pdf.*
2. ^[a] ^[b] ^[c] ^[d] ^[e] *McMahon D. Quantum Computing Explained. Hoboken, NJ: Wiley-Interscience; 2007. Chapter 1-3, 8.*
3. ^ *Naeim Dastgir. "Processing SAR data using Range Doppler and Chirp Scaling Algorithms." Royal Institute of Technology (KTH) 100 44 Stockholm, Sweden, April 2007 [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1065439/FULLTEXT01.pdf.*
4. ^ *European Space Agency. Sentinel-1 Level-1 Detailed Algorithm Definition. [Online]. Available: https://sentinel.esa.int/documents/247904/1877131/Sentinel-1-Level-1-Detailed-Algorithm-Definition.*
5. ^ *Rath M, Date H. "Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy." EPJ Quantum Technol. **11**, 72 (2024). doi:10.1140/epjqt/s40507-024-00285-3.*
6. ^ *Ranga D, Rana A, Prajapat S, Kumar P, Kumar K, Vasilakos AV (2024). "Quantum Machine Learning: Exploring the Role of Data Encoding Techniques, Challenges, and Future Directions." Mathematics. **12**(21): 3318. doi:10.3390/math12213318.*

7. ^Qiskit. "Lecture 2: Quantum circuits and algorithms." Jupyter Notebook. 2020. Available: https://github.com/Qiskit/platypus/blob/main/notebooks/summer-school/2020/lec-02.ipynb.
8. ^Jaffali H. "Getting to know Quantum Fourier Transform". ColibrITD Quantum. Apr 25, 2022. [Online]. Available: https://medium.com/colibritd-quantum/getting-to-know-quantum-fourier-transform-ae60b23e58f4.
9. ^Hall R. "Sentinel-1 Level-0 Decoding Demo." Jupyter Notebook. Available: https://nbviewer.org/github/Rich-Hall/sentinel1Level0DecodingDemo/blob/main/sentinel1Level0DecodingDemo.ipynb.

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.