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

Khalil Al Salahat† Mohamad El Moussawi† Veera Ganesh Yalla Ali J. Ghandour*
National Center for Remote Sensing *Earth Observation* *IHub Data* *National Center for Remote Sensing*
CNRS-L *RASID SARL* *IIIT Hyderabad* *CNRS-L*
Beirut, Lebanon Beirut, Lebanon Hyderabad, India Beirut, Lebanon

***Abstract*—Synthetic Aperture Radar (SAR) plays a vital role in remote sensing due to its ability to capture high-resolution images regardless of weather conditions or daylight. However, to transform the raw SAR signals into interpretable imagery, advanced data processing techniques are essential. A widely used technique for this purpose is the Range Doppler Algorithm (RDA), which takes advantage of Fast Fourier Transform (FFT) to convert signals into the frequency domain for further processing. However, the computational cost of this approach becomes significant when dealing with large datasets. This paper presents a Quantum Range Doppler Algorithm (QRDA) that utilizes the Quantum Fourier Transform (QFT) to accelerate processing compared to the classical FFT. Furthermore, it introduces a quantum implementation of the Range Cell Migration Correction (RCMC) in the Fourier domain, a critical step in the RDA pipeline, and evaluates its performance relative to its classical counterpart.**

***Index Terms*—Quantum computing, Synthetic aperture radar (SAR), Range Doppler Algorithm (RDA), Quantum Fourier Transform (QFT), Range Cell Migration Correction (RCMC), Remote Sensing**

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 encode information about the reflectivity properties of the scene, and transforming it into a usable image requires sophisticated processing to account for the motion of the radar and the geometry of wave propagation. The **Range Doppler Algorithm (RDA)** is one of the most widely used methods for the formation of SAR images. Its computational core 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 complexity of FFT $O(n \log n)$, although efficient for moderate datasets, becomes a bottleneck for large-scale or high-resolution SAR applications, where near-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 quantum states halfway through (e.g., after **QFT** but before **RCMC**) collapses the quantum states, negating the speedup by losing all superposition information from before the measurement and limiting it to a singular result. 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 these challenges, the contribution of this paper is two-fold:

1) Efficient quantum encoding, using as few qubits as possible, which allows us to encode N features with only $\log_2(N)$ qubits.
2) A quantum-domain **Range Cell Migration Correction (RCMC)** gate, which is a crucial part in the **Range Doppler algorithm**

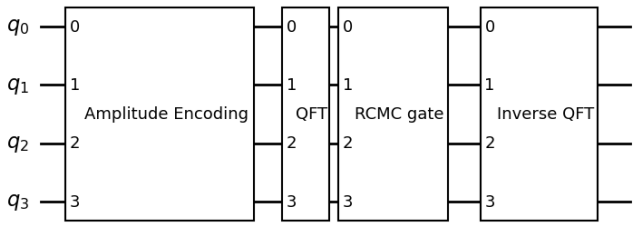

Fig. 1. Quantum circuit approach for the Data Encoding and the RCMC gate implementation.

II. Fundamentals of Quantum Computing

A. Classical vs. Quantum Information

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

*Corresponding Author: aghandour@cnrs.edu.lb.
†These authors contributed equally to this work

B. The Qubit and Superposition

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

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

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 shown in Equation 2:

$$|\psi\rangle = \sum_{i=1}^{n} c_i\,|v_i\rangle = c_1\,|v_1\rangle + c_2\,|v_2\rangle + ... + c_n\,|v_n\rangle \tag{2}$$

with $\sum_i |c_i|^2 = 1$. The squared modulus 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, the 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) which creates superposition from computational basis states as shown in Equation 3:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, \text{ where } H\,|0\rangle = |+\rangle\,, H\,|1\rangle = |-\rangle \tag{3}$$
with $|\pm\rangle = \frac{|0\rangle \pm |1\rangle}{\sqrt{2}}$

- Phase gate (P) which introduces a relative phase θ to $|1\rangle$ as shown in Equation 4:

$$P = \begin{pmatrix}1 & 0\\ 0 & e^{i\theta}\end{pmatrix}, \text{ with } P\,|0\rangle = |0\rangle\,, P\,|1\rangle = e^{i\theta}\,|1\rangle \tag{4}$$

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 the efficient utilization of Fast Fourier Transforms (FFTs).

A. Classical RDA

The classical RDA consists of four stages [3] [4] as depicted in Figure 2:

1) **Range Compression**: The raw signal undergoes range compression through frequency-domain matched filtering as expressed in Equation 5:

$$s_{rc}(\tau, \eta) = \text{IFFT}_\tau\left[\text{FFT}_\tau[s(\tau, \eta)] \cdot G(f_\tau)\right] \tag{5}$$

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 as expressed in Equation 6:

$$s_1(\tau, f_\eta) = \text{FFT}_\eta[s_{rc}(\tau, \eta)] \tag{6}$$

3) **Range Cell Migration Correction (RCMC)**: A phase correction compensates for slant-range variations caused by platform motion as shown in Equation 7

$$G_{\text{RCMC}}(f_r) = \exp\left[4i\pi\frac{f_r}{c}\left(R_0(\frac{1}{D(f_\eta, V)} - 1\right)\right] \tag{7}$$

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 the 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 as shown in Equation 8:

$$s_{ac}(\tau, \eta) = \text{IFFT}_\eta\left[s_2(\tau, f_\eta) \cdot H(f_\eta)\right] \tag{8}$$

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.

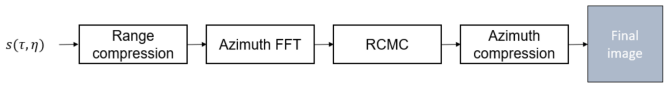

Fig. 2. Range Doppler Algorithm: block diagram.

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 encoding to measurement must be performed in the quantum domain to actually achieve this speedup [1]. The proposed approach is depicted in Figure 3 where the classical information is first encoded into the quantum domain using amplitude encoding. The next steps involve applying the range compression filter, the RCMC filter, and the azimuth compression filter after moving to the frequency domain and back using QFT (and inverse QFT) applied to the range and azimuth lines, respectively. More details are provided in the following.

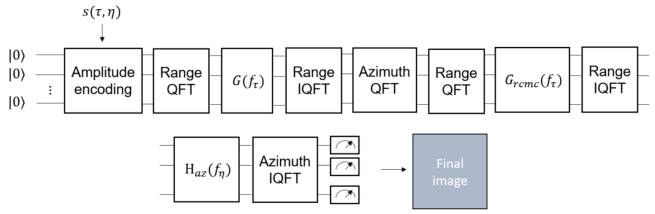

Fig. 3. Quantum Range Doppler Algorithm: proposed quantum circuit approach.

C. Amplitude Encoding

The first step in our algorithm is the encoding step, where we relied on amplitude encoding technique. Amplitude encoding embeds the information into the probability amplitudes of quantum states, which is done by first normalizing the dataset and then initializing the values into amplitudes. In our case, the encoded information consists of the SAR samples $s(\tau, f_\eta)$, made up of complex numbers that contain norm and phase information for each range and azimuth sample. Amplitude encoding is especially powerful because it allows one to encode N features with only $\log_2(N)$ qubits; for example, a 16x16 image would require only 16 qubits. Given a classical vector $\mathbf{x} = (x_0, x_1, ..., x_{N-1})$ representing our data, we encode it in a normalized quantum state as shown in Equation 9:

$$|\psi\rangle = \sum_{i=0}^{N-1} x_i \, |i\rangle \tag{9}$$

where $|i\rangle$ are the 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, that is, $\sum_i |x_i|^2 = 1$ [5] [6].

IV. QFT AND QRCMC

A. 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 as depicted in Equation 10:

$$\text{QFT}\,|x\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i x k/N} \, |k\rangle \tag{10}$$

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

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

1) Hadamard gates (H) previously discussed in Equation 3.
2) Controlled phase rotations to encode frequency-dependent phases defined in Equation 11:

$$R_k = \begin{pmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{pmatrix} \tag{11}$$

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, as illustrated in Figure 4 [8].

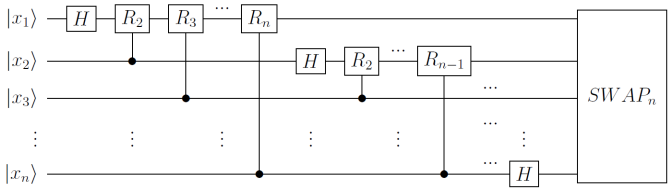

Fig. 4. Quantum Fourier Transform Gate.

B. Quantum RCMC

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 that implements the phase shift coefficients that we have calculated classically. The **RCMC** filter, previously discussed in Equation 7, is an array of phase elements that will filter the main radar data once multiplied, where each **RCMC** element needs to be multiplied by every range line.

To implement this logic in our quantum circuit, we created a gate that applies the corresponding phase shift to each of the 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 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.

RCMC implementation is hence proposed here as a diagonal unitary operation as introduced in Equation 12:

$$U_{\text{RCMC}} = \bigoplus_{k=1}^{N_r} e^{i\Theta_k} \otimes I_{N_a} \tag{12}$$

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

C. Circuit Realization

For a minimal 2×2 example (2 range bins × 2 azimuth samples) illustrated in Figure 1, the **RCMC** operator takes the following form shown in Equation 13:

$$U_{\text{RCMC}} = \begin{pmatrix} e^{i\theta_1} & 0 & 0 & 0 \\ 0 & e^{i\theta_1} & 0 & 0 \\ 0 & 0 & e^{i\theta_2} & 0 \\ 0 & 0 & 0 & e^{i\theta_2} \end{pmatrix} \tag{13}$$

When applied to a state $|\psi\rangle = \begin{bmatrix}\alpha_1, \alpha_2, \alpha_3, \alpha_4\end{bmatrix}^T$, it yields to results as in Equation 14:

$$U_{\text{RCMC}} |\psi\rangle = \begin{bmatrix} e^{\theta_1}\alpha_1 \\ e^{\theta_1}\alpha_2 \\ e^{\theta_2}\alpha_3 \\ e^{\theta_2}\alpha_4 \end{bmatrix} \tag{14}$$

Where α_i is the probability amplitude of each state $|i\rangle$, which contains 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).

V. Results

We verified our quantum **RCMC** implementation using the Qiskit state vector simulator (AerSimulator), which emulates an ideal noise-free quantum computer [9]. We compared the results with classical processing on Sentinel-1 SAR data (64×64 subset) [10]. As shown in Figure 5, the phase difference plot demonstrates near-perfect alignment between the quantum and classical outputs, confirming that the quantum implementation reproduces the expected classical behavior. The minimal discrepancies observed are attributable to numerical precision rather than algorithmic error. This test confirms the correctness of the proposed quantum approach.

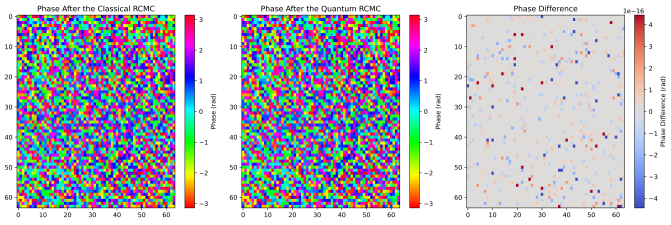

Fig. 5. The Phase Value Difference of the RDA Classical and Quantum approaches.

VI. Conclusion

This work demonstrates the feasibility of quantum acceleration for **Earth Observation** applications. Our implementation maintains mathematical equivalence with classical RDA while being fully executable on quantum hardware, as demonstrated through simulations of a 64×64 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. Furthermore, future advancements in quantum sensors for satellites could eliminate the need for data encoding, further enhancing speed and efficiency.

In 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] 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] D. McMahon, *Quantum Computing Explained*. Hoboken, NJ: Wiley-Interscience, 2007, ch. 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). https://doi.org/10.1140/epjqt/s40507-024-00285-3

[6] Ranga, D., Rana, A., Prajapat, S., Kumar, P., Kumar, K., & Vasilakos, A. V. (2024). "Quantum Machine Learning: Exploring the Role of Data Encoding Techniques, Challenges, and Future Directions.", Mathematics, 12(21), 3318. https://doi.org/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] Hamza Jaffali, "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] Javadi-Abhari, A., Treinish, M., Krsulich, K., Wood, C., Lishman, J., Gacon, J., Martiel, S., Nation, P., Bishop, L., Cross, A., Johnson, B. & Gambetta, J. Quantum computing with Qiskit. (2024)

[10] R. Hall, "Sentinel-1 Level-0 Decoding Demo," Jupyter Notebook, Available: https://nbviewer.org/github/Rich-Hall/sentinel1Level0DecodingDemo/blob/main/sentinel1Level0DecodingDemo.ipynb.