Article

Quantum Representations and Scaling Up Algorithms of Adaptive Sampled-Data in Log-Polar Coordinates

Chan Li, Dayong Lu * and Hao Dong

School of Mathematics and Statistics, Henan University, Kaifeng 475001, China; clchanli@163.com (C.L.); haodong96@126.com (H.D.)

* Correspondence: 10100088@vip.henu.edu.cn

Abstract: In log-polar coordinates, the conventional data sampling method is to sample uniformly in the log-polar radius and polar angle directions, which makes the sample at the fovea of the data denser than that of the peripheral. The central oversampling phenomenon of the conventional sampling method gives no more efficient information and results in computational waste. Fortunately, the adaptive sampling method is a powerful tool to solve this problem in practice, so the paper introduces it to quantum data processing. In the paper, the quantum representation model of adaptive sampled data is proposed first, in which the upper limit of the sampling number of the polar angles is related to the log-polar radius. Owing to this characteristic, its preparation process has become relatively complicated. Then, in order to demonstrate the practicality of the model given in the paper, the scaling up algorithm with an integer scaling ratio based on biarcuate interpolation and its circuit implementation of quantum adaptive sampled data is given. However, due to the special properties of the adaptive sampling method in log-polar coordinates, the interpolation process of adaptive sampled data becomes quite complicated as well. At the end of this paper, the feasibility of the algorithm is verified by a numerical example.

Keywords: quantum computation; quantum data representations; data scaling up; biarcuate interpolation; log-polar coordinates

Citation: Li, C.; Lu, D.; Dong, H. Quantum Representations and Scaling Up Algorithms of Adaptive Sampled-Data in Log-Polar Coordinates. *Entropy* **2021**, *23*, 1462. <https://doi.org/10.3390/e23111462>

Academic Editor: Donald J. Jacobs

Received: 16 September 2021

Accepted: 27 October 2021

Published: 4 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Quantum computation [1], as the interdisciplinary subject of many disciplines, such as computer science, mathematics, physics and engineering technology, has generated much interest in the world. Quantum computation can solve many problems that classical computers cannot solve or are inefficient at, and has the ability for data processing beyond classical computers by utilizing the coherent superposition characteristics of quantum states. Quantum algorithms [2,3] use quantum interference, quantum entanglement and other unique quantum mechanical principles to calculate, which can achieve exponential acceleration or quadratic acceleration for some classical algorithms.

Quantum data processing is a pioneer in the implementation of information processing on quantum computers and has currently become a hot research field in quantum computation. Research results include: quantum data representation [4–10], geometric transformation [11,12], color transformation [13], feature extraction [14], data scaling [15–17], data watermarking [18,19], data registration [20] and so on. These research achievements have greatly promoted the development of quantum data processing.

Although quantum data processing has made great progress, there are still many problems. Most of the existing quantum data representation models are carried out in Cartesian coordinates; only few studies on quantum data representation models are discussed in log-polar coordinates. Compared with Cartesian coordinates, log-polar coordinates have the advantages of rotation and scale invariance. Because any scale and rotation in Cartesian coordinates can be represented as shifting in the log-polar radius and the polar angular

directions in log-polar coordinates, respectively, it is easier to rotate data in log-polar coordinates than that in Cartesian coordinates. The existing quantum data representation model in log-polar coordinates is based on sample uniformly in both the log-polar radius and polar angle directions. The main problem is the high computational cost in data processing, which comes from central oversampling. In other words, the sample density in the data center is higher than that in the periphery, and no more useful information is gained from oversampling. An effective way to solve this problem is to employ the adaptive sampling method [21], and the detailed information is given in the next section.

According to the characteristics of log-polar coordinates, the adaptive sampling method can effectively solve the problem of oversampling in the center because the upper limit of the sampling number of the polar angles is related to the log-polar radius. This greatly avoids the computational waste caused by central oversampling. Hence, the new quantum data representation model based on the adaptive sampling method in log-polar coordinates is proposed in the paper. In quantum representations of adaptive sampled data in log-polar coordinates (QRASD), three entangled qubit sequences are applied to deposit the data information, the log-polar radius information and the polar angle information. In order to demonstrate the practicality of the representation model, the quantum data scaling up algorithm with integer scaling ratio based on biarcuate interpolation is given. The feasibility of the algorithm is verified by an example at the end of the paper.

The rest of this paper is arranged as follows. In Section 2, the conventional sampling method and adaptive sampling method in log-polar coordinates are introduced. The novel representation model and its preparation procedure are described in Section 3. In Section 4, the quantum data scaling up algorithm with integer scaling ratio using biarcuate interpolation based on the QRASD model is given. Finally, a conclusion is presented in Section 5.

2. Preliminaries

In this section, the conventional sampling method and adaptive sampling method in log-polar coordinates are introduced in detail first, and then the QUALPI model based on the conventional sampling method is also discussed.

The conventional mathematical expression for a mapping procedure in which log-polar coordinates are interchangeable with Cartesian coordinates is shown below:

$$\rho = \log_{base} \sqrt{(x - x_c)^2 + (y - y_c)^2}, \quad (1)$$

$$\theta = \tan^{-1} \frac{y - y_c}{x - x_c}, \quad (2)$$

where (x_c, y_c) represents the center datum of the transformation in Cartesian coordinates; (x, y) denotes the sampling datum in Cartesian coordinates; (ρ, θ) denotes the log-polar radius and the polar angular position in log-polar coordinates. For the sake of simplicity, the base is the natural logarithm. Figure 1 shows a conventional comparison between the two sampling methods. Figure 1a represents the conventional sampling method in log-polar coordinates, and Figure 1b represents the conventional sampling method in Cartesian coordinates.

Figure 1. (a) Sampling in log-polar coordinates. (b) Sampling in Cartesian coordinates.

Figure 1a indicates that the sample at the fovea of the data is denser than that of the peripheral using the conventional sampling method in log-polar coordinates, so the data processing focuses more on the fovea or central area while the peripheral area is given less consideration. Fortunately, the adaptive sampling method in log-polar coordinates [21] avoids this problem, which is introduced below.

Let n_ρ and n_θ be the numbers of samples in the log-polar radius and the polar angular directions, respectively; $R_i, i = 0, 1, \dots, n_\rho$, be the radius size which is sampled equally to cover 360° in the angular direction for $\theta = 0, 1, \dots, n_\theta$. Furthermore, to prevent undersampling in the log-polar radius direction, $R_{n_\rho} - R_{n_\rho-1} \leq 1$ must be satisfied. It is easy to see that the following equation should be true:

$$R_i = e^{\frac{i \times \log R_{\max}}{n_\rho}}, \quad R_{\max} = R_{n_\rho}, \quad (3)$$

$$n_\rho \geq \frac{\log R_{\max}}{\log R_{\max} - \log(R_{\max} - 1)}. \quad (4)$$

In the polar angular direction, the maximum sampling circumference, $R_{\max}$, has approximately $2\pi R_{\max}$ data. Hence, the number of samples in the polar angular direction is equal or greater than that amount, which is necessary to prevent data being missed in the sampling procedure, that is:

$$n_\theta \geq 2\pi R_{\max}. \quad (5)$$

To make the computation simpler, n_ρ and n_θ are selected as a multiple power of two. Then, Equations (4) and (5) can be expressed as:

$$n_\rho = 2^{\lceil \frac{1}{\log 2} \times \log \frac{\log R_{\max}}{\log R_{\max} - \log(R_{\max} - 1)} \rceil}, \quad n_\theta = 8R_{\max}, \quad (6)$$

where $\lceil A \rceil$ is the nearest integer equal to or greater than A . It is found that the most suitable sampling method is to set the number of polar angles to eight times the number of log-polar radii. Therefore, the adaptive sampling method uses an eight-fold sampling method for each circumference. The adaptive sampling method is shown in Figure 2a. Figure 2b describes the resulting sample distribution in the polar angular and log-polar radius directions, a series of sample points are arranged in the step-like manner, that is, as ρ increases, the length of the sample points (number of samples represented by θ) increases. The total number of samples is less than what is needed for the conventional sampling method to have the same sampling resolution.

Figure 2. (a) Adaptive sampling in log-polar coordinates. (b) The resulting sample distribution in the polar angular and log-polar radius directions.

The QUALPI model [9] based on conventional sampling method stores and processes data in log-polar coordinates. It utilizes three entangled qubit sequences to store the gray scale information, the log-polar radius position and the polar angular position information for each pixel in a log-polar image. The model is shown below:

$$|I\rangle = \frac{1}{\sqrt{2^{m+n}}} \sum_{\rho=0}^{2^m-1} \sum_{\theta=0}^{2^n-1} |g(\rho, \theta)\rangle \otimes |\rho\rangle \otimes |\theta\rangle, \quad (7)$$

$$g(\rho, \theta) = C_{\rho\theta}^0 C_{\rho\theta}^1 \dots C_{\rho\theta}^{q-1}, \quad C_{YX}^t \in \{0, 1\}, \quad g(\rho, \theta) \in [0, 2^q - 1]. \quad (8)$$

The QUALPI model realizes the quantum representation of log-polar images for the first time, but the problem of central oversampling in log-polar coordinates still exists. Therefore, it is of great significance for the development of quantum data processing in the future to improve the existing quantum data representation models and find a better method to represent quantum data in log-polar coordinates.

3. Quantum Data Representations and Preparation Based on Adaptive Sampling Method

The premise of the application of data in log-polar coordinates is the existence of its representation and preparation. In this section, details of representations and preparation procedures are given in Sections 3.1 and 3.2, respectively. The time complexity of the preparation procedure is also discussed in Section 3.3.

3.1. Quantum Data Representations Based on Adaptive Sampling Method

Motivated by the QUALPI model, a novel quantum representation model of adaptive sampled-data in log-polar coordinates (QRASD) to store and process data is proposed as follows:

$$|D\rangle = \frac{1}{\sqrt{2^{2m+3}}} \sum_{\rho=0}^{M-1} \sum_{\theta=0}^{8\rho+7} |f(\rho, \theta)\rangle |\rho\rangle |\theta\rangle, \quad (9)$$

$$m = \begin{cases} \lceil \log_2 M \rceil, & M > 1; \\ 1, & M = 1, \end{cases} \quad (10)$$

where ρ encodes the information of log-polar radius and θ encodes the information of the polar angle. For each fixed ρ , i.e., in the circumference with a log-polar radius ρ , the circumference is sampled equidistantly with sampling number of 8ρ and sampling interval of $2\pi/8(\rho+1)$. It is worth noting that both ρ and θ are coded from 0, which have a certain impact on subsequent calculations.

In order to improve the flexibility and breadth of the model, three different data information representation methods are presented.

1. Representing the data information as in the FRQI model

When we replace $|f(\rho, \theta)\rangle$ in Equation (9) with the method given in the FRQI model [7], i.e., $|f(\rho, \theta)\rangle = \cos \alpha_{\rho\theta} |0\rangle + \sin \alpha_{\rho\theta} |1\rangle$, then we have:

$$|D\rangle = \frac{1}{\sqrt{2^{2m+3}}} \sum_{\rho=0}^{M-1} \sum_{\theta=0}^{8\rho+7} (\cos \alpha_{\rho\theta} |0\rangle + \sin \alpha_{\rho\theta} |1\rangle) |\rho\rangle |\theta\rangle, \quad (11)$$

$$\alpha_{\rho\theta} \in \left[0, \frac{\pi}{2}\right], \quad \rho = 0, 1, \dots, M-1, \quad \theta = 0, 1, \dots, 8\rho+7. \quad (12)$$

Equation (11) is referred to as the FRQI model based on the adaptive sampling method, shorthand for the AS-FRQI sub-model.

2. Representing the data information as in the NEQR model

If $|f(\rho, \theta)\rangle$ is taken the form of the data information in the NEQR model [8], where $|f(\rho, \theta)\rangle = |C_{\rho\theta}^0 C_{\rho\theta}^1 \dots C_{\rho\theta}^{q-1}\rangle$, $C_{\rho\theta}^t \in \{0, 1\}$, $t = 0, 1, \dots, q-1$, then Equation (9) can be rewritten as:

$$|D\rangle = \frac{1}{\sqrt{2^{2m+3}}} \sum_{\rho=0}^{M-1} \sum_{\theta=0}^{8\rho+7} \bigotimes_{t=0}^{q-1} |C_{\rho\theta}^t\rangle |\rho\rangle |\theta\rangle. \quad (13)$$

The new representation method is called the NEQR model based on an adaptive sampling method, or the AS-NEQR sub-model for short.

3. Representing the data information as in the QR2-DD model

The IEEE (Institute of Electrical and Electronics Engineers) is an international organization that has designed specific binary formats for storing floating-point numbers. IEEE-754 is the most widely used standard for floating point arithmetic operations. Figure 3 shows the format of floating-point numbers in IEEE-754.

Figure 3. Format of floating-point numbers in IEEE-754.

The QR2-DD model [10] uses the IEEE-754 standard to represent quantum data information, which lays a foundation for quantum data processing. Let $|f(\rho, \theta)\rangle = |s_{\rho\theta}^0 s_{\rho\theta}^1 \dots s_{\rho\theta}^p s_{\rho\theta}^{p+1} \dots s_{\rho\theta}^{p+q-1}\rangle$, $s_{\rho\theta}^t \in \{0, 1\}$, then Equation (9) goes to the following form:

$$|D\rangle = \frac{1}{\sqrt{2^{2m+3}}} \sum_{\rho=0}^{M-1} \sum_{\theta=0}^{8\rho+7} \bigotimes_{t=0}^{p+q-1} |s_{\rho\theta}^t\rangle |\rho\rangle |\theta\rangle, \quad (14)$$

where $S_{YX}^s = s_{YX}^0$ is the sign qubit, $S_{YX}^e = s_{YX}^1 s_{YX}^2 \dots s_{YX}^p$ encodes the exponent and $S_{YX}^f = s_{YX}^{p+1} s_{YX}^{p+2} \dots s_{YX}^{p+q-1}$ encodes the mantissa. The QR2-DD model based on adaptive sampling method is abbreviated as the AS-QR2-DD sub-model.

Figure 4a takes a $\rho \times 8\rho$ ($\rho = 0, 1$) quantum data $|D\rangle$ as an example, that is, when $\rho = 0$, there is $\theta = 0, 1, \dots, 7$, and when $\rho = 1$, there is $\theta = 0, 1, \dots, 15$. Its quantum representation expression based on the AS-NEQR sub-model is given in Figure 4b. The data information of $|D\rangle$ from big to small are 192, 128, 64, 32 and 0. The quantum representation expression omits the part where the data is 0.

Figure 4. (a) A 2×16 adaptive sampled-data in log-polar coordinates. (b) Its quantum representation expression according to the AS-NEQR sub-model.

3.2. Quantum Data Preparation Based on Adaptive Sampling Method

Since the preparation process for the three quantum data representation sub-models given above is similar, only one detailed preparation process is given in this subsection. Compared with the AS-FRQI sub-model and the AS-NEQR sub-model, floating-point numbers can represent decimal numbers, which is more suitable for data processing in actuality. Therefore, the AS-QR2-DD sub-model is taken as an example for quantum preparation.

Due to the inherent characteristics of the binary operation process of quantum computer, redundancy is an inevitable result in the process of binary computation in quantum computers. In Figure 5, the gray part contains data information, while the white part is redundant without data information.

Figure 5. The data information and the redundancy.

The empty quantum state with $(p + q + 2m + 3)$ qubits is constructed as follows:

$$|\psi_0\rangle = |0\rangle^{\otimes(p+q+2m+3)}. \quad (15)$$

The preparation process of adaptive sampled-data is shown in Figure 6. With the empty quantum state, the position information is first prepared in Figure 6a, and then the data information, which is the shaded part in Figure 6b, is prepared by using the comparator.

Figure 6. (a) Prepare position information. (b) Prepare data information.

Step 1: In this step, the single-qubit identity gate I and the Hadamard gate H are utilized to transform $|\psi_0\rangle$ to the intermediate state $|\psi_1\rangle$. The whole quantum operation U_1 in Step 1 is defined as follows:

$$U_1 = I^{\otimes(p+q)} \otimes H^{\otimes m} \otimes H^{\otimes(m+3)}, \quad (16)$$

which transforms the initial state $|\psi_0\rangle$ to the intermediate state $|\psi_1\rangle$:

$$|\psi_1\rangle = U_1(|\psi_0\rangle) = \frac{1}{\sqrt{2^{2m+3}}} \sum_{\rho=0}^{2^m-1} \sum_{\theta=0}^{2^{m+3}-1} |0\rangle^{\otimes(p+q)} |\rho\rangle |\theta\rangle. \quad (17)$$

Step 2: Compared with the preparation process of the existing quantum data representation models, the upper limit of the sampling number of θ in the QRASD model is related to ρ , that is $0 \leq \theta \leq (8\rho + 7)$, so a comparison between $(8\rho + 7)$ and θ is necessary. In order to achieve this goal, some quantum circuit modules are introduced first.

1. Plus one and minus one modules

This part is designed to perform plus one and minus one [17] operations. The quantum circuit of plus one and minus one modules are shown in Figure 7a,b, respectively.

Figure 7. (a) Quantum circuit of plus one module. (b) Quantum circuit of minus one module.

2. Multiply by eight module

This module implements the multiplication by eight, i.e., $|X\rangle \rightarrow |8X\rangle$, and the quantum circuit is shown in Figure 8. The essence of multiplying by eight is that the qubit shifts three places to the left. It can be seen that, the effect of the module is $|000x_{n-1}x_{n-2}\dots x_0\rangle \rightarrow |x_{n-1}x_{n-2}\dots x_0000\rangle$.

Figure 8. Quantum circuit for multiplying by eight module.

3. Comparator module

Comparator [22] plays an important role in quantum data processing. The function of this module is to compare $|x\rangle$ and $|y\rangle$, where $|x\rangle = |x_{n-1}x_{n-2}\dots x_0\rangle$, $|y\rangle = |y_{n-1}y_{n-2}\dots y_0\rangle$, $x_t, y_t \in \{0, 1\}$, $t = 0, 1, \dots, n-1$. The qubit $|z\rangle$ is the output. If $|z\rangle = |1\rangle$, then $x \geq y$; if $|z\rangle = |0\rangle$, then $x < y$. The quantum circuit of the comparator module is shown in Figure 9.

Figure 9. Quantum circuit of comparator module.

These are all the modules required for the process of comparison. As shown in Figure 10, the specific operation of this step is as follows: The qubit representing ρ is executed first by the

operation of adding one to obtain $(\rho + 1)$. Then, $(\rho + 1)$ is multiplied by eight and subtracted by one to obtain $(8\rho + 7)$. Finally, $(8\rho + 7)$ is compared with θ . When the output $|z\rangle$ is equal to $|1\rangle$, the next step is executed.

Figure 10. Quantum circuit of Step 2.

Step 3: Since there are $8(\rho + 1)$ polar angles on each circumference of log-polar radius ρ , this step is divided into $8(1 + 2 + \dots + M) = 4M(M + 1)$ sub-operations to set the data information at the corresponding position. The sub-operation $U_{\rho\theta}$ is expressed as below:

$$U_{\rho\theta} = I^{\otimes(p+q)} \otimes \sum_{\substack{l=0 \\ (l,k) \neq (\rho,\theta)}}^{2^m-1} \sum_{k=0}^{2^{m+3}-1} |lk\rangle\langle lk| + \Omega_{\rho\theta} \otimes |\rho\theta\rangle\langle\rho\theta|, \quad (18)$$

$$\rho = 0, 1, \dots, M-1, \quad \theta = 0, 1, \dots, 8\rho + 7, \quad (19)$$

where the value-setting operation $\Omega_{\rho\theta}$ can be shown as:

$$\Omega_{\rho\theta} = \bigotimes_{t=0}^{p+q-1} \Omega_{\rho\theta}^t, \quad \Omega_{\rho\theta}^t : |0\rangle \rightarrow |0 \oplus s_{\rho\theta}^t\rangle. \quad (20)$$

It is obviously that $\Omega_{\rho\theta}$ only affect data at the specified position. The process for $\Omega_{\rho\theta}$ to obtain data is as follows:

$$\Omega_{\rho\theta}|0\rangle^{\otimes(p+q)} = \bigotimes_{t=0}^{p+q-1} (\Omega_{\rho\theta}^t|0\rangle) = \bigotimes_{t=0}^{p+q-1} |0 \oplus s_{\rho\theta}^t\rangle = \bigotimes_{t=0}^{p+q-1} |s_{\rho\theta}^t\rangle = |f(\rho, \theta)\rangle. \quad (21)$$

The following formula shows the result of the intermediate state $|\psi_1\rangle$ under the influence of $U_{\rho\theta}$.

$$\begin{aligned} U_{\rho\theta}(|\psi_1\rangle) &= \frac{1}{\sqrt{2^{2m+3}}} \left(\sum_{\substack{l=0 \\ (l,k) \neq (\rho,\theta)}}^{2^m-1} \sum_{k=0}^{2^{m+3}-1} |0\rangle^{\otimes(p+q)} |l\rangle |k\rangle + \Omega_{\rho\theta} |0\rangle^{\otimes(p+q)} |\rho\rangle |\theta\rangle \right) \\ &= \frac{1}{\sqrt{2^{2m+3}}} \left(\sum_{\substack{l=0 \\ (l,k) \neq (\rho,\theta)}}^{2^m-1} \sum_{k=0}^{2^{m+3}-1} |0\rangle^{\otimes(p+q)} |l\rangle |k\rangle + |f(\rho, \theta)\rangle |\rho\rangle |\theta\rangle \right). \end{aligned} \quad (22)$$

To set the $4M(M + 1)$ log-polar data, the whole operation U_2 which transforms the intermediate state $|\psi_1\rangle$ to the final state $|\psi_2\rangle$ is defined below:

$$U_2 = \prod_{\rho=0}^{M-1} \prod_{\theta=0}^{8\rho+7} U_{\rho\theta}, \quad (23)$$

$$\begin{aligned} |\psi_2\rangle &= U_2(|\psi_1\rangle) \\ &= \frac{1}{\sqrt{2^{2m+3}}} \sum_{\rho=0}^{M-1} \sum_{\theta=0}^{8\rho+7} |f(\rho, \theta)\rangle |\rho\rangle |\theta\rangle. \end{aligned} \quad (24)$$

Hitherto, the initialized state $|0\rangle^{\otimes(p+q+2m+3)}$ is transformed to the desired state.

3.3. Time Complexity Analysis of the Preparation Procedure

Time complexity refers to the time needed to complete an algorithm in the field of computer science and engineering, and is an important parameter to evaluate the merits of an algorithm. Therefore, it is necessary to discuss the time complexity of the preparation procedure. To calculate the time complexity, the number of operation units of the preparation procedure must be estimated, and each operation unit runs for the same amount of time.

Theorem 1. In order to store $\rho \times 8\rho$ ($\rho = 0, 1, \dots, M-1$) data with a data range from 0 to $2^{p+q}-1$, the whole quantum preparation of the AS-QR2-DD sub-model costs no more than $O(m^2 + (p+q)mM^2)$.

Proof. Firstly, in Step 1, U_1 is consisted of $(2m+3)$ Hadamard gates, and:

$$m = \begin{cases} \lceil \log_2 M \rceil, & M > 1; \\ 1, & M = 1, \end{cases} \quad (25)$$

where $\lceil A \rceil$ is the nearest integers equal to or greater than A . Then it's obvious that the time complexity of this step is $O(m)$.

Secondly, U_2 can be decomposed into four operations: plus one, multiply by eight, minus one and comparator operations.

The plus one module consists of one NOT gate, one $C-NOT$ gate, one C^2-NOT gate, ..., and one $C^{m-1}-NOT$ gate. While a Toffoli gate (C^2-NOT gate) can be decomposed into six $C-NOT$ gates and some 1-bit gates, such as two Hadamard gates H , one phase gate S , three $\frac{\pi}{8}$ gates T and four conjugate transpose of the $\frac{\pi}{8}$ gate T^+ [23]. Summed up, a Toffoli gate has a time complexity of 16. Figure 11 depicts the process of implementing the Toffoli gate with elementary quantum gates. When $t = 3$ and 4, one C^t-NOT gate can be decomposed into $(3 \cdot 2^t - 4)$ $C-NOT$ gates and $2 \cdot 2^t$ 1-bit gates. When $t = 5, 6, \dots, m-1$, each controlled quantum gate C^t-NOT can be simulated by $8(t-3)$ Toffoli gates [24]. Therefore, the time complexity of the plus one module is $O(m^2)$.

Figure 11. Toffoli gate is decomposed into basic quantum gates.

Multiplied by the eight module contains m swap gates, which can be decomposed into three $C-NOT$ gates. So the time complexity of multiplying by the eight module is $O(m)$.

Similar to the plus one module, the minus one module requires one NOT gate, one $C-NOT$ gate, one C^2-NOT gate, ..., and one $C^{m+2}-NOT$ gate to complete. Then, the time complexity of the minus one module is also $O(m^2)$.

The comparator operation consists of one $C^{2(m+3)}-NOT$ gate, one $C^{2m+5}-NOT$ gate, three C^2-NOT gates, three C^4-NOT gates, ... and three $C^{2(m+2)}-NOT$ gates. The further decomposition has the time complexity of the comparator module $O(m^2)$.

To sum up, the time complexity of Step 2 is $O(m^2)$.

Last but not least, the operation U_2 in Step 3 can be divided into $4M(M+1)$ sub-operations $U_{\rho\theta}$ to set all the data at the corresponding position. Each $U_{\rho\theta}$ is composed of $(p+q)$ $C^{2m+3}-NOT$ gates. It is known that sub-operation $U_{\rho\theta}$ costs no more than $O((p+q)m)$. Hence, the time complexity of Step 3 will cost no more than $O((p+q)mM^2)$.

As the discussion above, the time complexity of the whole preparation procedure is about $O(m^2 + (p+q)mM^2)$. $\square$

The quantum circuit of preparation for the example data shown in Figure 4 is described in Figure 12.

Figure 12. The quantum circuit of preparation for the example data shown in Figure 4.

4. Quantum Data Scaling Up Algorithm with Integer Scaling Ratio Based on Biarcuate Interpolation

Data scaling up as a kind of data geometric transformation has been widely studied and applied in the classical data processing; however, the quantum version of which, in log-polar coordinates does not exist. Here, by employing biarcuate interpolation, a quantum algorithm to scale up adaptive sampled-data in log-polar coordinates is given. To make it easier to understand, the AS-NEQR sub-model is taken as an example.

4.1. Quantum Circuit Modules for Arithmetic Operations

Designing a quantum scaling up circuit of biarcuate interpolation is a complex task. A series of quantum circuit modules to realize special functions are introduced first, including multiplying the *C-NOT* operation module, adder module [25], subtractor module [25], multiplier module [26] and divider module [27].

4.1.1. Multiply *C-NOT* Operation Module

The multiply *C-NOT* operation is to copy the determined quantum states. The quantum circuit of the multiply *C-NOT* operation is shown in Figure 13. It utilizes n *C-NOT* gates to copy the n -qubit information of $|X\rangle = |x_{n-1}x_{n-2}\dots x_0\rangle$ into the n ancillary qubits $|0\rangle^{\otimes n}$, where $|x_{n-1}\rangle, |x_{n-2}\rangle, \dots, |x_1\rangle$ and $|x_0\rangle$ are the control qubits and the n ancillary qubits $|0\rangle^{\otimes n}$ are the target qubits. The time complexity of the multiply *C-NOT* operation module is $O(n)$.

Figure 13. Quantum circuit of multiply *C-NOT* operation.

4.1.2. Adder Module

The adder module [25] is not only the core of the multiplier, but also one of the core modules of constructing the biarcuate interpolation method. The adder (A) adding a n -qubit unsigned integer $|Y\rangle$ to a n -qubit unsigned integer $|X\rangle$ is designed by one half adder (HA) and $(n - 1)$ full adders (FA) as shown in Figure 14. Here, the input $|X\rangle$ comes

out as the sequence $|s_n s_{n-1} \dots s_0\rangle$, which represents the sum of $|X + Y\rangle$, and the input $|Y\rangle$ stays the same. Other unremarked qubits are the garbage outputs and the input qubit $|0\rangle$ is the ancillary constant input. For convenience, the block diagram of A omits the ancillary inputs and the garbage outputs. The time complexity of the adder module is $O(n)$.

Figure 14. (a) Quantum circuit of adder. (b) Quantum circuit of half adder. (c) Quantum circuit of full adder. (d) Quantum simple circuit of adder.

4.1.3. Subtractor Module

The function of the subtractor (S) is to implement the subtraction of binary unsigned integers $|X\rangle$ and $|Y\rangle$ [25]. The subtractor is designed by one half subtractor (HS) and $(n - 1)$ full subtractors (FS). The input $|X\rangle$ come out as the sequence $|d_n d_{n-1} \dots d_0\rangle$, which represents the difference of $|X - Y\rangle$, and the input $|Y\rangle$ stays the same. If $X \geq Y$, then $|d_n\rangle = |0\rangle$, if $X < Y$, then $|d_n\rangle = |1\rangle$. The quantum circuit of the subtractor module is shown in Figure 15. The time complexity of the subtractor module is $O(n)$.

Figure 15. (a) Quantum circuit of subtractor. (b) Quantum circuit of half subtractor. (c) Quantum circuit of full subtractor. (d) Quantum simple circuit of subtractor.

4.1.4. Multiplier Module

The purpose of the multiplier module [26] is to take the product of two numbers, which can be done by addition. Let $|X\rangle = |x_{m-1} x_{m-2} \dots x_0\rangle$ and $|Y\rangle = |y_{n-1} y_{n-2} \dots y_0\rangle$, then:

$$\begin{aligned}
|X \times Y\rangle &= |(2^{m-1}x_{m-1} + 2^{m-2}x_{m-2} + \dots + 2^0x_0) \times Y\rangle \\
&= |(2^{m-1}x_{m-1}) \times Y + (2^{m-2}x_{m-2}) \times Y + \dots + (2^0x_0) \times Y\rangle \\
&= |x_{m-1} \times (2 \times 2^{m-2}Y) + x_{m-2} \times (2 \times 2^{m-3}Y) + \dots + x_0 \times (2^0 \times Y)\rangle. \quad (26)
\end{aligned}$$

First, the auxiliary qubit $|0\rangle^{\otimes n}$ is given as the initial value of the partial product. When $|x_0\rangle = |1\rangle$, copy $|Y\rangle$ to the partial product, add '0' to the left of the partial product, and add '0' to the right of $|Y\rangle$. Then, if $|x_t\rangle = |1\rangle$, $t = 1, 2, \dots, m-1$, add the padded $|Y\rangle$ to the partial product. If $|x_t\rangle = |0\rangle$, the partial product remains unchanged. Since the adder is a carry addition, only after the process do we need to add '0' to the right of $|Y\rangle$. The block diagram of multiplier (M) is shown in Figure 16, where it consists of multiple adders. The time complexity of the multiplier module is $O(mn)$.

Figure 16. Quantum circuit of multiplier module.

4.1.5. Divider Module

Based on restoring the division algorithm, Khosropour et al. [27] proposed the quantum division circuit. In order to make full use of the operation modules given in this paper, appropriate modifications have been made in this subsection. In Figure 17, a pseudo code for this algorithm is given.

```

A = 0...0X
B = Y0...0
For t = n-1 to 0
    A = 2A
    if A ≥ B,
        qt = 1, A = A - B
    else
        qt = 0
    end
End
*Quotient = qn-1qn-2...q0
*Remainder = upper half of 2A

```

Figure 17. Pseudo code for the division process.

According to the algorithm, to get the quotient ($|Q\rangle$) and the remainder ($|R\rangle$), first of all, the lowest n -qubit of $|A\rangle$ are equal to the dividend $|X\rangle$, and the highest n -qubit are equal to zero. Furthermore, register $|B\rangle$ is initiated such that its highest n -qubit contains the divisor $|Y\rangle$ and its lowest n -qubit are set to zero. Then, $|A\rangle$ is multiplied by two and compared to $|B\rangle$. If the result is nonnegative, then the qubit $|q_t\rangle$ in the quotient is set to one and $|A\rangle$ should be subtracted $|B\rangle$; otherwise, $|q_t\rangle = |0\rangle$. This process is repeated n times, each time one qubit of the quotient is determined. A concrete calculation example is given to clarify this procedure. Let $|X\rangle = |1001\rangle$ and $|Y\rangle = |0010\rangle$, then the initial state is:

$$|A\rangle = |00001001\rangle, \quad |B\rangle = |00100000\rangle.$$

The first iteration:

$$|A\rangle = |2A\rangle = |00010010\rangle, \quad |A\rangle < |B\rangle, \quad |q_3\rangle = |0\rangle.$$

The second iteration:

$$|A\rangle = |2A\rangle = |00100100\rangle, \quad |A\rangle \geq |B\rangle.$$

$$|q_2\rangle = |1\rangle, \quad |A\rangle = |A - B\rangle = |00000100\rangle.$$

The third iteration:

$$|A\rangle = |2A\rangle = |00001000\rangle, \quad |A\rangle < |B\rangle, \quad |q_1\rangle = |0\rangle.$$

The fourth iteration:

$$|A\rangle = |2A\rangle = |00010000\rangle, \quad |A\rangle < |B\rangle, \quad |q_0\rangle = |0\rangle.$$

Therefore, it turns out that:

$$|R\rangle = |0001\rangle, \quad |Q\rangle = |0100\rangle.$$

Obviously, the calculation process of the divider can be achieved through the comparators, subtractors and multiply by two operations. The corresponding quantum circuit is shown in Figure 18. In order to facilitate the subsequent operations, the upper output is marked as Y, the middle output is marked as R and the lower output is marked as Q in the quantum simple circuit of the divider. The time complexity of the divider module is $O(n^3)$.

Figure 18. (a) Quantum circuit of divider. (b) Quantum circuit of multiply by two. (c) Quantum simple circuit of divider.

4.2. Quantum Biarcuate Interpolation Method in Log-Polar Coordinates

The principle of the biarcuate interpolation method in log-polar coordinates is similar to that of the bilinear interpolation method in Cartesian coordinates. The difference is that bilinear interpolation is based on line segments, but biarcuate interpolation is based on angular segments.

4.2.1. Arcuate Interpolation Method

The datum at position (ρ, θ) of the interpolated data can be reconstructed from the datum at positions (ρ, θ_1) and (ρ, θ'_1) of the original data. As shown in Figure 19, the value of the desired datum $f(\rho, \theta)$ can be obtained by the following equation:

$$\frac{f(\rho, \theta) - f(\rho, \theta_1)}{\theta - \theta_1} = \frac{f(\rho, \theta'_1) - f(\rho, \theta)}{\theta'_1 - \theta}, \quad (27)$$

where $f(\rho, \theta_1)$ and $f(\rho, \theta'_1)$ are two known data. That is to say,

$$f(\rho, \theta) = (1 - h)f(\rho, \theta_1) + hf(\rho, \theta'_1), \quad h = \frac{\theta - \theta_1}{\theta'_1 - \theta_1}. \quad (28)$$

Figure 19. Arcuate interpolation method.

4.2.2. Biarcuate Interpolation Method

The biarcuate interpolation method is more complicated than the arcuate interpolation method. The value $f(\rho, \theta)$ at position (ρ, θ) of the interpolated data can be reconstructed from the value at positions (ρ_1, θ_1) , (ρ_1, θ'_1) , (ρ_2, θ_2) and (ρ_2, θ'_2) of the original data. The corresponding relationship is shown in Figure 20.

Figure 20. Biarcuate interpolation method.

To get $f(\rho, \theta)$, auxiliary points (ρ_1, θ) and (ρ_2, θ) are added to facilitate the calculation. The datum at position (ρ_1, θ) can be calculated as follows:

$$\frac{f(\rho_1, \theta) - f(\rho_1, \theta_1)}{\theta - \theta_1} = \frac{f(\rho_1, \theta'_1) - f(\rho_1, \theta)}{\theta'_1 - \theta}, \quad (29)$$

where $f(\rho_1, \theta_1)$ and $f(\rho_1, \theta'_1)$ are given. That is to say,

$$f(\rho_1, \theta) = (1 - h)f(\rho_1, \theta_1) + hf(\rho_1, \theta'_1), \quad h = \frac{\theta - \theta_1}{\theta'_1 - \theta_1}. \quad (30)$$

In the same manner,

$$f(\rho_2, \theta) = (1 - l)f(\rho_2, \theta_2) + lf(\rho_2, \theta'_2), \quad l = \frac{\theta - \theta_2}{\theta'_2 - \theta_2}, \quad (31)$$

$$f(\rho, \theta) = (1 - w)f(\rho_1, \theta) + wf(\rho_2, \theta), \quad w = \frac{\rho - \rho_1}{\rho_2 - \rho_1}. \quad (32)$$

Thus, the datum in position (ρ, θ) of the interpolated data can be calculated as shown below:

$$f(\rho, \theta) = (1-w)(1-h)f(\rho_1, \theta_1) + (1-w)hf(\rho_1, \theta'_1) + w(1-l)f(\rho_2, \theta_2) + wl f(\rho_2, \theta'_2). \quad (33)$$

4.2.3. Quantum Realization of Biarcuate Interpolation Method

Because the number of samples of polar angles on each circumference is different, the interpolation process of adaptive sampled-data becomes quite complicated. Assume that a $\rho \times 8\rho$ ($\rho = 0, 1, \dots, M-1$) quantum data $|D\rangle$ is scaled up to a $\rho' \times 8\rho'$ ($\rho' = 0, 1, \dots, M'-1$) quantum data $|D'\rangle$ based on biarcuate interpolation. The scale ratio is r , i.e., $M' = rM$, where $r \in N$ and represented by k qubits. For the sake of simplicity, the interpolation process of adaptive sampled-data is divided into two steps according to the log-polar radius. The first step is interpolation based on the arcuate interpolation method. The second step is interpolation based on the biarcuate interpolation method using the data interpolated in the previous step. The specific process is discussed below.

Step 1: Circuit realization of arcuate interpolation method.

For each circumference with log-polar radius ρ in the scaled up data, if $(\rho + 1)$ is an integer multiple of r , i.e., $\rho + 1 = r(\rho_0 + 1)$, then the interpolated data is the corresponding circumference. As shown in Figure 21, ρ_0 and θ_0 in (a) are the log-polar radius and polar angle corresponding to ρ and θ' in (b), respectively. That is,

$$\rho_0 \xrightarrow{\text{enlarge}} \rho, \quad \theta_0 \xrightarrow{\text{enlarge}} \theta', \quad \theta_0 + 1 \xrightarrow{\text{enlarge}} \theta' + r. \quad (34)$$

The data after arcuate interpolation is shown in Figure 21c as the solid line. In this step, the data on the dotted line has not been interpolated.

Figure 21. (a) The original data. (b) Schematic of arcuate interpolation. (c) The data after arcuate interpolation.

It can be seen from Figure 21b, there are $(r-1)$ points between θ' and $(\theta' + r)$ that need to be interpolated. In order to get the data of these points, the value of ρ_0 and θ_0 in the original data must be found first. Obviously,

$$r(\rho_0 + 1) = \rho + 1, \quad r\theta_0 = \theta', \quad r(\theta_0 + 1) = \theta' + r. \quad (35)$$

Thus,

$$\rho_0 = \frac{\rho + 1}{r} - 1 = Q[\rho + 1, r] - 1, \quad \theta_0 = \frac{\theta'}{r} = Q[\theta', r] = Q[\theta, r], \quad (36)$$

where $\theta = \theta', \theta', \dots, \theta' + r - 1$ and $Q[A, B]$ is the quotient of A divided by B in division with remainder. It is easy to get $h = \frac{R[\theta, r]}{r}$, where $R[A, B]$ is the remainder of A divided by B in division with remainder. Therefore, combining Equations (28) and (36), the value

$f(\rho, \theta)$ in position (ρ, θ) of the interpolated data in Figure 21b can be calculated to arcuate interpolation according to the following equation:

$$\begin{aligned} f(\rho, \theta) &= (1-h)f(\rho_0, \theta_0) + hf(\rho_0, \theta_0 + 1) \\ &= \frac{r-R[\theta, r]}{r}f(Q[\rho+1, r]-1, Q[\theta, r]) + \frac{R[\theta, r]}{r}f(Q[\rho+1, r]-1, Q[\theta, r]+1) \\ &= \frac{(r-R[\theta, r])f(Q[\rho+1, r]-1, Q[\theta, r]) + R[\theta, r]f(Q[\rho+1, r]-1, Q[\theta, r]+1)}{r}, \end{aligned} \quad (37)$$

In order to prepare the data information $f(\rho, \theta)$ in position (ρ, θ) of the resulting data, the data information $f(\rho_0, \theta_0)$ and $f(\rho_0, \theta_0 + 1)$ in positions (ρ_0, θ_0) and $(\rho_0, \theta_0 + 1)$ of the original data need to be prepared first. To achieve the data information for the corresponding position, two quantum operators $\Omega_{\rho_0, \theta_0}$ and $\Omega_{\rho_0, \theta_0+1}$ are used to achieve the original data information, respectively. The quantum operator $\Omega_{\rho, \theta}$ can realize the aim of assigning the data information as follows:

$$\Omega_{\rho, \theta}|0\rangle^{\otimes q} = \bigotimes_{t=0}^{q-1} (\Omega_{\rho\theta}^t|0\rangle) = \bigotimes_{t=0}^{q-1} |0 \oplus s_{\rho\theta}^t\rangle = \bigotimes_{t=0}^{q-1} |s_{\rho\theta}^t\rangle = |f(\rho, \theta)\rangle. \quad (38)$$

Due to the quantum superposition, all data are processed simultaneously, so all original data must be assigned. Let the assignment operation $\Omega = \sum_{\rho=0}^{M-1} \sum_{\theta=0}^{8\rho+7} \Omega_{\rho, \theta}$, then the time complexity of the assignment operation is $O(qmM^2)$.

The quantum circuit of arcuate interpolation is shown in Figure 22, where $m = \begin{cases} \lceil \log_2 M \rceil, & M > 1; \\ 1, & M = 1, \end{cases}$ and $m' = \begin{cases} \lceil \log_2 M' \rceil, & M' > 1; \\ 1, & M' = 1, \end{cases}$. If the auxiliary bits are less than the input bits in multiple C-NOT operations, only the lowest bits of the input bits are copied. The dotted line serves no purpose but as an indicator. When $R = R[\rho+1, r] = 0$, that is, when $(\rho+1)$ is an integer multiple of r , the output $f(\rho, \theta)$ is the required data. Since the final output is the quotient without considering the remainder, $f(\rho, \theta)$ has an error of less than one. The time complexity of arcuate interpolation is $O(m'^3 + qmM^2)$.

Figure 22. The quantum circuit of arcuate interpolation.

Step 2: Circuit realization of biarcuate interpolation method.

As shown in Figure 23, this step interpolates the part which is not interpolated in Step 1, that is, the data in the dotted part of Figure 21c or Figure 23a. Figure 23b shows the specific schematics of biarcuate interpolation for adaptive sampled-data, where points (ρ_1, θ_1) , $(\rho_1, \theta_1 + 1)$, (ρ_2, θ_2) and $(\rho_2, \theta_2 + 1)$ in Figure 23a are the four closest points to (ρ, θ) . Figure 23c describes the data after biarcuate interpolation, which is the scaled up data.

Figure 23. (a) The data after arcuate interpolation. (b) Schematic of biarcuate interpolation. (c) The scaled up data.

It is obvious that the four interpolated data $f(\rho_1, \theta_1)$, $f(\rho_1, \theta_1 + 1)$, $f(\rho_2, \theta_2)$ and $f(\rho_2, \theta_2 + 1)$ from the previous step are needed to complete the interpolation for this step. The mathematical expression is as follows:

$$f(\rho, \theta) = (1-w)(1-h)f(\rho_1, \theta_1) + (1-w)hf(\rho_1, \theta_1 + 1) + w(1-l)f(\rho_2, \theta_2) + wl f(\rho_2, \theta_2 + 1). \quad (39)$$

From Step 1, $(\rho_1 + 1)$ and $(\rho_2 + 1)$ are both integer multiples of r , assuming that:

$$\rho_1 + 1 = kr, \quad \rho_2 + 1 = (k+1)r. \quad (40)$$

where $k = Q[\rho_1 + 1, r]$. That is:

$$\rho_1 = rk - 1 = rQ[\rho + 1, r] - 1, \quad (41)$$

$$\rho_2 = rk + r - 1 = rQ[\rho + 1, r] + r - 1, \quad (42)$$

$$\rho_2 - \rho_1 = r. \quad (43)$$

Oobviously,

$$w = \frac{R[\rho + 1, r]}{\rho_2 - \rho_1} = \frac{R[\rho + 1, r]}{r}. \quad (44)$$

Because there are $8(\rho + 1)$ polar angles on each circumference with log-polar radius ρ , and the common multiple of $8(\rho_1 + 1)$ and $8(\rho + 1)$ is $8(\rho + 1)(\rho_1 + 1)$. The principle of common multiples is used, there are:

$$\theta_1 = Q[\theta(\rho_1 + 1), (\rho + 1)] = Q[\theta r Q[\rho + 1, r], \rho + 1], \quad (45)$$

$$h = \frac{R[\theta(\rho_1 + 1), (\rho + 1)]}{\rho + 1} = \frac{R[\theta r Q[\rho + 1, r], \rho + 1]}{\rho + 1}. \quad (46)$$

In the same manner,

$$\theta_2 = Q[\theta(\rho_2 + 1), (\rho + 1)] = Q[\theta(rQ[\rho + 1, r] + r), \rho + 1], \quad (47)$$

$$l = \frac{R[\theta(\rho_2 + 1), (\rho + 1)]}{\rho + 1} = \frac{R[\theta(rQ[\rho + 1, r] + r), \rho + 1]}{\rho + 1}. \quad (48)$$

Thus, the Equation (39) can be rewritten as:

$$\begin{aligned}
 f(\rho, \theta) = & \left[(r - R[\rho + 1, r]) (\rho + 1 - R[\theta r Q[\rho + 1, r], \rho + 1]) \right. \\
 & \times f(rQ[\rho + 1, r] - 1, Q[\theta r Q[\rho + 1, r], \rho + 1]) \\
 & + (r - R[\rho + 1, r]) R[\theta r Q[\rho + 1, r], \rho + 1] \\
 & \times f(rQ[\rho + 1, r] - 1, Q[\theta r Q[\rho + 1, r], \rho + 1] + 1) \\
 & + R[\rho + 1, r] (\rho + 1 - R[\theta(rQ[\rho + 1, r] + r), \rho + 1]) \\
 & \times f(rQ[\rho + 1, r] + r - 1, Q[\theta(rQ[\rho + 1, r] + r), \rho + 1]) \\
 & + R[\rho + 1, r] R[\theta(rQ[\rho + 1, r] + r), \rho + 1] \\
 & \left. \times f(rQ[\rho + 1, r] + r - 1, Q[\theta(rQ[\rho + 1, r] + r), \rho + 1] + 1) \right] \div r \div (\rho + 1).
 \end{aligned} \quad (49)$$

As shown in Figure 24, the quantum circuit of biarcuate interpolation is given, where:

$$m' = \begin{cases} \lceil \log_2 M' \rceil, & M' > 1; \\ 1, & M' = 1, \end{cases} \text{ and the } +(r-1) \text{ operation can be realized by } (r-1) \text{ plus}$$

one operation. The dotted line serves no purpose but as an indicator. $R = R[\rho + 1, r]$ represents the remainder of $(\rho + 1)$ divided by r . When $R \neq 0$, the output $f(\rho, \theta)$ is the required data. Similarly, $f(\rho, \theta)$ has an error of less than one. However, due to the characteristics of the adaptive sampling method, there may be additional errors, which will be explained in detail in the next subsection. The time complexity of biarcuate interpolation is $O(m'^3 + r q m' M^2)$. Therefore, the time complexity of the whole interpolation process is $O(m'^3 + r q m' M^2)$.

In classical computers, the algorithm complexity is mainly evaluated by two parameters, one is the number of rows of pre-cached data required to complete interpolation, and the other is the number of operation steps such as addition, subtraction, multiplication, division and displacement required to complete interpolation. Here, the time complexity of the proposed algorithm is estimated by 1-bit addition and subtraction as the unit of operation on classical computers. The $m \times n$ multiplication consists of $O(mn)$ additions, and the $n \times n$ division can be decomposed into $O(n^2)$ subtractions, assuming that all of the original data is known. By Equations (37) and (49), the time complexity of arcuate interpolation and biarcuate interpolation are $O((m'^2 + k^2 q^2) r M^2)$ and $O((k^2 + q^2) r q^2 m'^2 M^2)$, respectively. Therefore, the time complexity of the whole interpolation process in classical computers is $O((k^2 + q^2) r q^2 m'^2 M^2)$. Obviously, the time complexity is much higher than that of quantum computers.

Figure 24. The quantum circuit of biarcuate interpolation.

4.3. Example Verification

In this section, the sample data in Figure 4 is used to demonstrate the specific scaling-up process. For the $\rho \times 8\rho$ ($\rho = 0, 1$) data, we scale it up three times to become a $\rho' \times 8\rho'$ ($\rho' = 0, 1, \dots, 5$) dataset. When $\rho' = 2$ or 5, $(\rho' + 1)$ is an integer multiple of three. Therefore, the data when $\rho' = 2$ and 5 are interpolated first. When $\rho' = 2$, the Equation (37) can be rewritten as:

$$f(2, \theta) = \frac{(3 - R[\theta, 3])f(0, Q[\theta, 3]) + R[\theta, 3]f(0, Q[\theta, 3] + 1)}{3}, \quad (50)$$

where $\theta = 0, 1, \dots, 23$. The specific operation is as follows:

$$f(2, 0) = f(0, 0) = 0, \quad (51)$$

$$f(2, 1) = \frac{2f(0, 0) + f(0, 1)}{3} = 10, \quad (52)$$

$\vdots$

$$f(2, 23) = \frac{f(0, 7) + 2f(0, 8)}{3} = \frac{f(0, 7) + 2f(0, 0)}{3} = 64. \quad (53)$$

Owing to the characteristics of the plus one module, $f(0, 8) = f(0, 0)$. In the same manner, the data when $\rho' = 5$ can be achieved. The data after arcuate interpolation are shown in Figure 25.

Figure 25. The data after arcuate interpolation.

Then, the rest of the data can be calculated from the data already obtained by Equation (50). When $\rho' > r = 3$, the data that need to be interpolated can be easily calculated. For example, when $\rho' = 4$, the Equation (49) can be rewritten as:

$$f(4, \theta) = \left[(5 - R[3\theta, 5])f(2, Q[3\theta, 5]) + R[3\theta, 5]f(2, Q[3\theta, 5] + 1) \right. \quad (54)$$

$$\left. + 2(5 - R[6\theta, 5])f(5, Q[6\theta, 5]) + 2R[6\theta, 5]f(5, Q[6\theta, 5] + 1) \right] \div 15, \quad (55)$$

where $\theta = 0, 1, \dots, 39$. And the specific operation is as follows:

$$f(4,0) = \frac{5f(2,0) + 10f(5,0)}{15} = 42, \quad (56)$$

$$f(4,1) = \frac{2f(2,0) + 3f(2,1) + 8f(5,1) + 2f(5,2)}{15} = 27, \quad (57)$$

$\vdots$

$$f(4,39) = \frac{3f(2,23) + 2f(2,24) + 2f(5,46) + 8f(5,47)}{15} = 72, \quad (58)$$

where $f(2,24) = f(2,0) = 0$. However, when $\rho' < r = 3$, the process becomes a little more complicated. When $\rho' = 0$, the Equation (49) turns into:

$$f(0,\theta) = \frac{2f(-1,0) + f(2,3\theta)}{3} = \frac{f(2,3\theta)}{3}, \quad \theta = 0, 1, \dots, 7, \quad (59)$$

where $f(-1,0)$ should be the data of the pole in the log-polar coordinates. However, the adaptive sampling method does not sample at the poles; thus, the characteristics of the minus one module are $f(-1,0) = f(7,0)$. In this example, $f(7,0)$ is a redundant datum that has not been prepared, so $f(7,0) = 0$. If $f(7,0) \neq 0$, there is a certain error. The effect of all the data after biarcuate interpolation is shown in Figure 26.

Figure 26. The sample data in Figure 4 is magnified three times.

5. Conclusions

In the existing quantum data representation models, the two variables representing positions are independent of each other. However, in the quantum data representation model proposed in this paper, the upper limit of the sampling number of the polar angles is related to the log-polar radius. This is an inspiration for the future research work about the representation of irregularly shaped data.

The quantum scaling up algorithm with integer scaling ratio based on biarcuate interpolation and its circuit of adaptive sampled-data in log-polar coordinates are also given in this paper. In the quantum scaling up circuit, the operation modules required are constructed on the basis of the existing methods in the references. Meanwhile, the feasibility of the quantum algorithm is also verified by an example. However, the quantum scaling up algorithm still has certain shortcomings, and the quantum scaling down circuit has not been given. That's one of the most important things we'll be looking at in the future.

In addition, how to change the scaling ratio from integer multiples to real multiples and how to optimize the circuit are also problems worth studying in future work.

Author Contributions: Writing—original draft, C.L.; Writing—review and editing, D.L. and H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Feynman, R.P. Simulating physics with computers. *Int. J. Theor. Phys.* **1982**, *21*, 467–488. [\[CrossRef\]](#)
2. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
3. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC), Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
4. Venegas-Andraca, S.E.; Bose, S. Storing, processing and retrieving an image using quantum mechanics. In Proceedings of the SPIE 5105, Quantum Information and Computation, Orlando, FL, USA, 21–25 April 2003; pp. 137–147.
5. Latorre, J.I. Image compression and entanglement. *arXiv* **2005**, arXiv:quant-ph/0510031.
6. Venegas-Andraca, S.E.; Ball, J.L. Processing images in entangled quantum systems. *Quantum Inf. Process.* **2010**, *9*, 1–11. [\[CrossRef\]](#)
7. Le, P.Q.; Dong, F.Y.; Hirota, K. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Inf. Process.* **2011**, *10*, 63–84. [\[CrossRef\]](#)
8. Zhang, Y.; Lu, K.; Gao, Y.H.; Wang, M. NEQR: A novel enhanced quantum representation of digital images. *Quantum Inf. Process.* **2013**, *12*, 2833–2860. [\[CrossRef\]](#)
9. Zhang, Y.; Lu, K.; Gao, Y.H.; Xu, K. A novel quantum representation for log-polar images. *Quantum Inf. Process.* **2013**, *12*, 3103–3126. [\[CrossRef\]](#)
10. Zhang, R.; Xu, M.Y.; Lu, D.Y. A generalized floating-point quantum representation of 2-D data and their applications. *Quantum Inf. Process.* **2020**, *19*, 20. [\[CrossRef\]](#)
11. Wang, B.; Hao, M.Q.; Li, P.C.; Liu, Z.B. Quantum representation of indexed images and its applications. *Int. J. Theor. Phys.* **2020**, *59*, 374–402. [\[CrossRef\]](#)
12. Le, P.Q.; Iliyasu, A.M.; Dong, F.; Hirota, K. Fast Geometric Transformations on Quantum Images. *IAENG Int. J. Appl. Math.* **2010**, *40*, 113–123.
13. Jiang, N.; Wu, W.; Wang, L.; Zhao, N. Quantum image pseudocolor coding based on the density-stratified method. *Quantum Inf. Process.* **2015**, *14*, 1735–1755. [\[CrossRef\]](#)
14. Jiang, N.; Ji, Z.; Li, H.; Wang, J. Quantum image interest point extraction. *Mod. Phys. Lett. A* **2021**, *36*, 16. [\[CrossRef\]](#)
15. Jiang, N.; Wang, J.; Mu, Y. Quantum image scaling up based on nearest-neighbor interpolation with integer scaling ratio. *Quantum Inf. Process.* **2015**, *14*, 4001–4026. [\[CrossRef\]](#)
16. Zhou, R.G.; Hu, W.W.; Fan, P.; Lan, H. Quantum realization of the bilinear interpolation method for NEQR. *Sci. Rep.-UK* **2017**, *7*, 17. [\[CrossRef\]](#) [\[PubMed\]](#)
17. Li, P.C.; Liu, X. Bilinear interpolation method for quantum images based on quantum Fourier transform. *Int. J. Quantum Inf.* **2018**, *16*, 26. [\[CrossRef\]](#)
18. Zhou, R.G.; Cheng, Y.D.; Lan, H.; Liu, X.A. Quantum watermarking algorithm based on chaotic affine scrambling. *Int. J. Quantum Inf.* **2019**, *17*, 23. [\[CrossRef\]](#)
19. Li, P.C.; Xiao, H.; Li, B.X. Quantum representation and watermark strategy for color images based on the controlled rotation of qubits. *Quantum Inf. Process.* **2016**, *15*, 4415–4440. [\[CrossRef\]](#)
20. Yan, F.; Abdullah, M.I.; Phuc, Q.L.; Sun, B.; Dong, F.Y.; Kaoru, H. A parallel comparison of multiple pairs of images on quantum computers. *Int. J. Innov. Comput. Appl. (IJICA)* **2013**, *5*, 199–212. [\[CrossRef\]](#)
21. Matungka, R. Studies on Log-Polar Transform for Image Registration and Improvements Using Adaptive Sampling and Logarithmic Spiral. Ph.D. Thesis, The Ohio State University, Columbus, OH, USA, 2009.
22. Wang, D.; Liu, Z.; Zhu, Y.; Li, S. Design of Quantum Comparator Based on Extended General Toffoli Gates with Multiple Targets. *Comput. Sci.* **2012**, *39*, 302–306.
23. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*; Cambridge University Press: New York, NY, USA, 2000.
24. Barenco, A.; Bennett, C.H.; Cleve, R.; Divincenzo, D.P.; Margolus, N.; Shor, P.; Sleator, T.; Smolin, J.; Weinfurter, H. Elementary gates for quantum computation. *Phys. Rev. A* **1995**, *52*, 3457–3467. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Vedral, V.; Barenco, A.; Ekert, A. Quantum networks for elementary arithmetic operations. *Phys. Rev. A* **1996**, *54*, 147–153. [\[CrossRef\]](#) [\[PubMed\]](#)

-
26. Li, P.C.; Shi, T.; Zhao, Y.; Lu, A. Design of threshold segmentation method for quantum image. *Int. J. Theor. Phys.* **2020**, *59*, 514–538. [[CrossRef](#)]
 27. Khosropour, A.; Aghababa, H.; Forouzandeh, B. Quantum Division Circuit Based on Restoring Division Algorithm. In Proceedings of the 2011 Eighth International Conference on Information Technology, Pafos, Cyprus, 31 August–2 September 2011; pp. 1037–1040.