

APPLICATION OF A QUANTUM AMPLITUDE REDISTRIBUTION ALGORITHM TO THE DATA FILTERING PROBLEM

K. R. Zakharova

St. Petersburg State University
Russia
reshetova.carina@yandex.ru

A. A. Chernikov

St. Petersburg State University
Russia
artem.chernikov00@list.ru

S. S. Sysoev

St. Petersburg State University
Russia
s.s.sysoev@spbu.ru

Article history:

Received 23.03.2026, Accepted 19.06.2026

Abstract

This paper presents an analysis of the applicability of a quantum amplitude redistribution algorithm to the data filtering problem and the results of modeling the algorithm's operation in comparison with a median filter.

Key words

Quantum algorithms, data filtering, median filter.

1 Introduction

Quantum information studies the computational capabilities of physical systems described by the mathematical apparatus of quantum mechanics. The fields of algorithms, noise and error problems, and the application of quantum effects have intersections in a wide variety of computing fields ([13], [14]). One of the important and rapidly growing areas in this field is the development of new quantum algorithms in the circuit model [2], [3]. The advantage of such algorithms is based on the use of quantum parallelism — the ability of a quantum system to follow several computational trajectories simultaneously. The number of such trajectories grows exponentially with the size of the computational system, but when obtaining classical information from a quantum state (upon measurement) it collapses to a single trajectory [10]. The main approach to overcoming this drawback is to redistribute the probability amplitudes of quantum states before their collapse (for example, using interference). This makes it possible to increase the probability of measuring certain states (more informative ones) relative to others (less informative ones). In some cases, the exponential advantage can be preserved, as for example in Shor's algorithm [8].

The Grover algorithm [4] and its more general modifications [1] allow searching for numbers corresponding to an arbitrary algorithmic condition of the form $A(x) = 1$,

where A is some algorithm and x is the numerical parameter of A . The choice of the algorithm A determines the nature of the problem being solved. If, for example, x is a numerical representation of a path in a complete graph and A is a check of the condition $len(x) < b$, then the Grover algorithm can be used to solve the traveling salesman problem. In [7], the algorithm [1] is used to amplify the amplitude of the correct result of binary classification in quantum machine learning.

If the condition $A(x) = 1$ corresponds to the situation of selecting the correct signal value from an array containing a digitized signal with outliers (anomalies), then the Grover algorithm can be used as a filtering method. The drawback of such a solution is that redistribution of probability amplitudes in the Grover algorithm requires a fixed number of steps (iterations) $T = O(\sqrt{N})$, where N is the size of the input array.

In [11], [12] a filtering algorithm is proposed that operates in time $O(1)$. Unlike [4], [1], in this algorithm the amplitudes of the vectors encoding the array elements are decreased proportionally to the distance of the elements from some reference value. As a result of measuring the state, one obtains the array index after probability-amplitude redistribution. Thus, at the output of the method, one obtains an element that was originally present in the data, which makes it similar to a median filter [5], [6], [9].

In this work, the properties of the algorithm (QASNE in [11]) are investigated for the task of signal filtering; the dependence of filtering quality on certain characteristics of the signal and outliers is analyzed; and simulation results of the algorithm operation are presented in comparison with a median filter. The paper is organized as follows: in the second section, a brief description of the algorithm from [11] is given. In the third section, the transformation scheme used for amplitude redistribution is presented. The fourth section is devoted

3 Implementation of the Rotation

This section contains the definition of the transformation used in the algorithm (1) for redistributing amplitudes between array indices, as well as a proof of its unitarity and a description of its decomposition into a sequence of elementary gates with a circuit depth that grows quadratically with the number of qubits.

3.1 Definition

The transformation is defined by the matrix (1), $2^n \times 2^n$, where n is the number of qubits. An arbitrary placement of signs outside the diagonal is allowed, subject only to the requirement that the transformation is unitary. Among admissible sign arrangements, it was decided to use the one that proved simplest to implement.

We define the sign arrangement for one qubit as follows:

$$R_1(\varphi) = \begin{pmatrix} \cos \frac{\varphi}{2} & -\sin \frac{\varphi}{2} \\ \sin \frac{\varphi}{2} & \cos \frac{\varphi}{2} \end{pmatrix}$$

For $(n+1)$ qubits we define the sign arrangement recursively:

$$R_{n+1}(\varphi) = \begin{pmatrix} A_n & -B_n \\ B_n & A_n \end{pmatrix}$$

The structures of matrices A_n and B_n maintain the definition of $R_{n+1}(\varphi)$ (1). In the submatrix A_n the signs are arranged in the same way as in R_n , while in the submatrix B_n they are arranged as in the n -qubit Walsh–Hadamard transformation. Note that transposition of the matrix is equivalent to changing the signs of the elements outside the diagonal.

3.2 Proof of Unitarity

Lemma 3.1. *Let H_n be the Walsh–Hadamard operator for n qubits. Then the following equality holds:*

$$H_n R_n(\varphi) H_n = R_n^\dagger(\varphi) \quad (2)$$

Proof. We use mathematical induction over the number of qubits. For one qubit we have:

$$\begin{aligned} H_1 R_1(\varphi) H_1 &= \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \cos \frac{\varphi}{2} & -\sin \frac{\varphi}{2} \\ \sin \frac{\varphi}{2} & \cos \frac{\varphi}{2} \end{pmatrix} \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \\ &= \begin{pmatrix} \cos \frac{\varphi}{2} & \sin \frac{\varphi}{2} \\ -\sin \frac{\varphi}{2} & \cos \frac{\varphi}{2} \end{pmatrix} = R_1^\dagger(\varphi) \end{aligned}$$

Introduce the notation:

$$a = \sqrt{\frac{2^n}{2^{n+1} - 1}} \sin \frac{\varphi}{2}$$

Then $R_{n+1}(\varphi)$ can be represented in the following form:

$$R_{n+1}(\varphi) = \begin{pmatrix} \sqrt{1-a^2} R_n & -a H_n \\ a H_n & \sqrt{1-a^2} R_n \end{pmatrix}$$

The operator R_n is parameterized by some other angle different from φ ; its value is omitted for brevity. Since the sign arrangement in R_{n+1} is defined recursively, the matrix R_n satisfies formula (1).

Induction step:

$$\begin{aligned} H_{n+1} R_{n+1}(\varphi) H_{n+1} &= \\ &= \frac{1}{\sqrt{2}} \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix} \begin{pmatrix} \sqrt{1-a^2} R_n & -a H_n \\ a H_n & \sqrt{1-a^2} R_n \end{pmatrix} \times \\ &\quad \times \frac{1}{\sqrt{2}} \begin{pmatrix} H_n & H_n \\ H_n & -H_n \end{pmatrix} = \\ &= \begin{pmatrix} \sqrt{1-a^2} R_n^\dagger & a H_n \\ -a H_n & \sqrt{1-a^2} R_n^\dagger \end{pmatrix} = R_{n+1}^\dagger(\varphi) \end{aligned}$$

Theorem 3.2. *The transformation $R_n(\varphi)$ is unitary.*

Proof. Again, we apply mathematical induction over the number of qubits. $R_1(\varphi)$ is nothing other than a rotation on the Bloch sphere around the Y axis, whose unitarity is evident. Let $b = 1 - a^2$.

Induction step:

$$\begin{aligned} R_{n+1} R_{n+1}^\dagger &= \\ &= \begin{pmatrix} \sqrt{b} R_n & -a H_n \\ a H_n & \sqrt{b} R_n \end{pmatrix} \begin{pmatrix} \sqrt{b} R_n^\dagger & a H_n^\dagger \\ -a H_n^\dagger & \sqrt{b} R_n^\dagger \end{pmatrix} = \\ &= \begin{pmatrix} b R_n R_n^\dagger + a^2 H_n H_n^\dagger & a \sqrt{b} R_n H_n^\dagger - a \sqrt{b} H_n R_n^\dagger \\ a \sqrt{b} H_n R_n^\dagger - a \sqrt{b} R_n H_n^\dagger & b R_n R_n^\dagger + a^2 H_n H_n^\dagger \end{pmatrix} \end{aligned}$$

By the induction hypothesis, $R_n R_n^\dagger = I$, hence

$$b R_n R_n^\dagger + a^2 H_n H_n^\dagger = b I + a^2 I = I$$

From Lemma 3.1 it follows that $H_n R_n^\dagger = R_n H_n^\dagger$, therefore

$$a \sqrt{b} H_n R_n^\dagger - a \sqrt{b} R_n H_n^\dagger = 0$$

Hence, $R_{n+1} R_{n+1}^\dagger = I$. Similarly one can show that $R_{n+1}^\dagger R_{n+1} = I$.

3.3 Decomposition into a Sequence of Gates

To use the transformation $R_n(\varphi)$ in practice, for example on a quantum computer simulator, one needs to decompose it into a sequence of elementary operators (gates). For convenience, we will refer to such a decomposition as a quantum algorithm that implements the operator $R_n(\varphi)$.

Find the basis in which the operator $R_{n+1}(\varphi)$ takes a simple form. First, we move to the basis with the transition matrix CH_n — a controlled Walsh–Hadamard operator controlled by the most significant qubit:

$$\begin{aligned} CH_n R_{n+1}(\varphi) CH_n &= \\ &= \begin{pmatrix} I & 0 \\ 0 & H_n \end{pmatrix} \begin{pmatrix} \sqrt{1-a^2}R_n & -aH_n \\ aH_n & \sqrt{1-a^2}R_n \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & H_n \end{pmatrix} = \\ &= \begin{pmatrix} \sqrt{1-a^2}R_n & -aI \\ aI & \sqrt{1-a^2}R_n^\dagger \end{pmatrix} \end{aligned}$$

Next, we move to another basis with a transition matrix equivalent to a rotation around the Y axis on the Bloch sphere by an angle $\frac{\pi}{2}$:

$$\begin{aligned} \frac{1}{2} \begin{pmatrix} I & -I \\ I & I \end{pmatrix} \begin{pmatrix} \sqrt{b}R_n & -aI \\ aI & \sqrt{b}R_n^\dagger \end{pmatrix} \begin{pmatrix} I & I \\ -I & I \end{pmatrix} &= \\ &= \begin{pmatrix} \frac{\sqrt{b}}{2}(R_n + R_n^\dagger) & \frac{\sqrt{b}}{2}(R_n - R_n^\dagger) - aI \\ \frac{\sqrt{b}}{2}(R_n - R_n^\dagger) + aI & \frac{\sqrt{b}}{2}(R_n + R_n^\dagger) \end{pmatrix} \end{aligned}$$

We have previously established that transposition of R_n is equivalent to changing the signs of the off-diagonal elements; therefore $(R_n + R_n^\dagger)$ is a diagonal matrix, and $(R_n - R_n^\dagger)$ is a matrix with zeros on the diagonal. After several simplifications, we obtain:

$$\begin{pmatrix} I \cos \frac{\varphi}{2} & -R_n^\dagger(\psi) \sin \frac{\varphi}{2} \\ R_n(\psi) \sin \frac{\varphi}{2} & I \cos \frac{\varphi}{2} \end{pmatrix}$$

Here

$$\sin \frac{\psi}{2} = \frac{1}{\sqrt{2^{n+1} - 1}},$$

$$R_n(\psi) = \frac{1}{\sqrt{2^{n+1} - 1}} \begin{pmatrix} \sqrt{2^n} \pm 1 & \cdots & \pm 1 \\ \pm 1 & \sqrt{2^n} & \cdots & \pm 1 \\ \vdots & \vdots & \ddots & \vdots \\ \pm 1 & \pm 1 & \cdots & \sqrt{2^n} \end{pmatrix}$$

Finally, we move to the basis with the transition matrix of the operator $R_n^\dagger(\psi)$ controlled by the most significant qubit:

$$\begin{pmatrix} I & 0 \\ 0 & R_n^\dagger(\psi) \end{pmatrix} \begin{pmatrix} I \cos \frac{\varphi}{2} & -R_n^\dagger(\psi) \sin \frac{\varphi}{2} \\ R_n(\psi) \sin \frac{\varphi}{2} & I \cos \frac{\varphi}{2} \end{pmatrix} \times$$

$$\times \begin{pmatrix} I & 0 \\ 0 & R_n(\psi) \end{pmatrix} = \begin{pmatrix} I \cos \frac{\varphi}{2} & -I \sin \frac{\varphi}{2} \\ I \sin \frac{\varphi}{2} & I \cos \frac{\varphi}{2} \end{pmatrix}$$

Thus, we find the basis in which the original operator $R_{n+1}(\varphi)$ acts as a rotation around the Y axis on the Bloch sphere for the most significant qubit by the angle φ , implemented by just a single gate. To carry out the transformation, it is sufficient to switch to this basis, perform the rotation, and return to the initial basis, as illustrated in Fig. 3.

By recursively expanding $R_n(\psi)$ using the same scheme, we obtain an exponential growth of circuit depth with increasing number of qubits. However, this algorithm can be substantially simplified after decomposing the two internal operators $R_n(\psi)$. Between the transition back to the initial basis in the first transformation and the transition to the required basis in the second, no other gates occur. Therefore, these two transitions can be omitted. Carrying out such simplifications recursively and expanding all subsequent R_k , we obtain the optimized circuit shown in Fig. 4.

As a result, the number of elementary gates implementing the required algorithm becomes asymptotically $O(n^2)$, where n is the number of qubits on which the transformation is defined. Thus, it depends quadratically on the number of qubits, which is a quite acceptable result. If in one time step we allow the simultaneous execution of several single-qubit Walsh–Hadamard operators controlled by the same qubit, then the circuit depth grows linearly with the number of qubits.

4 Analysis of Applicability of the Algorithm to the Data Filtering Problem

4.1 Probability Distribution After Applying the Algorithm

Before applying the algorithm (1), the elements of the input array in register D are pairwise entangled with their indices in register C , and the probability distribution among them is uniform. After applying the algorithm, the amplitudes in register C are redistributed and the pairwise entanglement with the array elements disappears. To analyze the possibility of using the algorithm as a filter, we consider how the probability of an arbitrary index C_k changes:

Theorem 4.1. *If all elements in the array D are distinct, then the probability of the element with index k when measuring register C after a single application of the algorithm (1) is given by:*

$$\begin{aligned} P(C_k) &= \frac{1}{M} \left[\cos^2 \left(\sum_{i=0}^{n-1} \frac{\pi(r_i - d_i^k)}{2^{n-i+1}} \right) + \right. \\ &\quad \left. + \frac{1}{M-1} \sum_{j \in [0, M-1], j \neq k} \sin^2 \left(\sum_{i=0}^{n-1} \frac{\pi(r_i - d_i^j)}{2^{n-i+1}} \right) \right], \end{aligned} \quad (3)$$

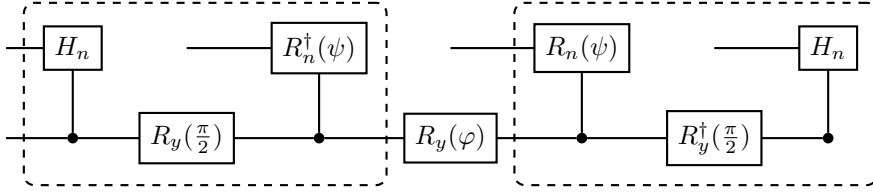


Figure 3. The first step in finding the decomposition of $R_{n+1}(\varphi)$. The dashed lines highlight transition blocks to the appropriate basis. Here $\sin \frac{\psi}{2} = \frac{1}{\sqrt{2^{n+1}-1}}$.

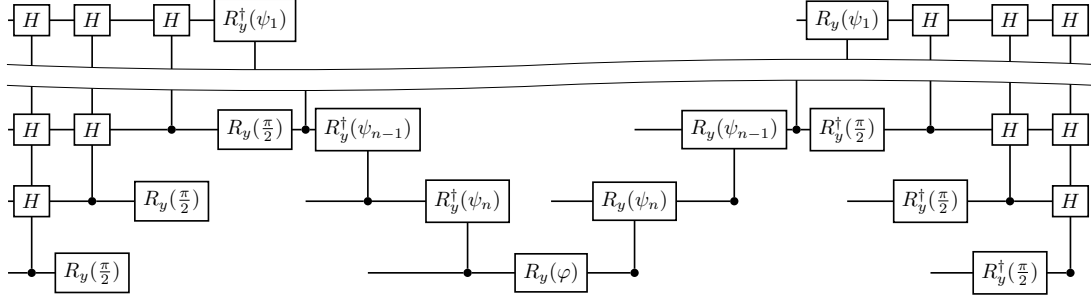


Figure 4. Decomposition of $R_{n+1}(\varphi)$. Here $\sin \frac{\psi_k}{2} = \frac{1}{\sqrt{2^{k+1}-1}}$.

where M is the number of elements of the input array, n is the number of qubits required to store one element, $r \in \{0, 1\}^n$ is the reference value, the upper indices k and j correspond to the element numbers d of the input array, the lower indices i correspond to the bit numbers starting from the least significant bit. Neglecting the schematic details of the bitwise execution of the algorithm, the formula 3 can be rewritten as:

$$P(C_k) = \frac{1}{M} [\cos^2(\frac{\pi}{2^{n+1}}(r - d^k)) + \frac{1}{M-1} \sum_{j \in [0, M-1], j \neq k} \sin^2(\frac{\pi}{2^{n+1}}(r - d^j))], \quad (4)$$

Proof. The algorithm (1) consists of sequentially applying the matrix $R_n(\varphi)$ (rotations) to register C , controlled by the values of qubits in register D . The angle φ for each rotation is determined by the index of the control qubit $0 \leq i < n$:

$$\varphi_i = \pi/2^{i+1}$$

In this case:

1. if $r_i > d_i^k$, then a rotation by φ_i is applied to the vector $|C_k\rangle$,
2. if $r_i < d_i^k$, then a rotation by $-\varphi_i$ is applied to the vector $|C_k\rangle$,
3. if $r_i = d_i^k$, then no rotation is applied to the vector $|C_k\rangle$.

This rule can be generalized by the following expression for the rotation angle of the vector $|C_k\rangle$:

$$\varphi_i^k = \varphi_i(r_i - d_i^k)$$

From the decomposition of the rotation matrix (1) into a sequence of elementary gates in Fig. 4, it follows that applying the rotations sequentially is equivalent to performing a single rotation by their total angle:

$$\prod_i R_n(\varphi_i) = R_n(\sum_i \varphi_i)$$

The most significant qubit $i = n - 1$ controls the rotation by the largest angle $\pi/4$, while the least significant qubit ($i = 0$) controls the rotation by the smallest angle $\pi/2^{n+1}$. For the vector $|C_k\rangle$, the total angle φ_k is:

$$\varphi_k = \sum_{i=0}^{n-1} \frac{\pi(r_i - d_i^k)}{2^{n-i+1}} \quad (5)$$

Thus, after applying all rotations, the amplitude of the vector $|C_k\rangle$ is distributed among all vectors in the space of register C :

$$\begin{aligned} |C_k\rangle &\rightarrow \cos(\varphi_k)|C_k\rangle \\ |C_j\rangle_{j \neq k} &\rightarrow \pm \frac{\sin(\varphi_j)}{\sqrt{M-1}}|C_j\rangle \end{aligned} \quad (6)$$

The rotations applied to vectors $C_{j, j \neq k}$ also redistribute the amplitudes among all vectors in the space of register C . At the same time, no interference occurs between vectors with different indices j , since they are entangled with different values in register D . Therefore, from each such vector, the value a_j is added to the amplitude of $|C_k\rangle$:

$$a_j = \pm \frac{\sin(\varphi_j)}{\sqrt{M-1}}$$

Taking into account the initial amplitudes of all vectors in the space, which are $1/\sqrt{M}$, for the probability $P(|C_k\rangle)$ we obtain:

$$P(|C_k\rangle) = \frac{1}{M}(\cos^2(\varphi_k) + \sum_{j \neq k} a_j^2) \quad (7)$$

4.2 Dependence of Filtering Quality on Input Data

Theorem 4.1 lets us formulate several useful corollaries regarding the applicability of the investigated algorithm.

Expression (3) determines the probability of obtaining the outlier index upon measurement only in the case when all values in the array D are distinct. Repeated values in the input array lead to interference in registers C and D , which can both decrease and increase this probability. For this reason, for practical filter implementations one can consider a modification of the algorithm in which the values of the array are loaded into register D together with their indices (in the least significant bits). This modification requires increasing register D by the size of register C . The added qubits do not participate in rotations and are needed only to prevent interference of indices in register $|C\rangle$ when the array values A are identical.

From expression (3) it follows that the reduction of the outlier probability is influenced by two factors:

- the number and significance of the bits in which the outlier differs from the reference value,
- the number of array elements that differ from the reference value (outliers).

The first factor determines how close the angle φ_k under the cosine in the first part of expression (3) is to $\pi/2$, while the second factor determines how many non-zero terms will appear in the second part of the expression.

Corollary 4.1.1 (Best Conditions for Filtering). *If the input array contains a single deviation (outlier) d^k :*

$$\forall i \quad d_i^k > r_i$$

and all other elements of the array match the reference value:

$$\forall j \in [0, M-1], j \neq k \quad d^j = r,$$

then the probability of obtaining the index k when measuring register C after applying the algorithm is the smallest among all possible input arrays and is equal to:

$$P(|C_k\rangle) = \frac{1}{M} \cos^2\left(\frac{\pi}{2} - \frac{\pi}{2^{n+1}}\right)$$

Proof. Since the outlier d^k is unique, its index in register C does not interfere with other indices, and the cosine value in expression (3) remains valid for it. For the remaining indices, rotations are not performed due to the condition $d^j = r \quad \forall j \neq k$. Therefore, nothing is added to the amplitude of $|C_k\rangle$ from the other superposition vectors.

Theorem 4.2. *If the input array contains a single outlier d^k , such that for some $l \in \mathbb{N}$:*

$$d^k \geq 2^l r,$$

and the deviation from r for all other elements does not exceed some σ :

$$\forall j \neq k \quad |d^j - r| \leq \sigma$$

then the probability of obtaining the index k when measuring register C after applying the algorithm can be upper-bounded:

$$P(|C_k\rangle) \leq \frac{1}{M} \left(\cos^2\left(\frac{\pi}{4} - \frac{\pi}{2^{l+2}}\right) + \sin^2\left(\frac{\pi\sigma}{2^{n+1}}\right) \right) \quad (8)$$

If σ is also estimated in terms of r :

$$\sigma \leq \frac{r}{2^p}$$

then the upper bound for $P(|C_k\rangle)$ becomes:

$$P(|C_k\rangle) \leq \frac{1}{M} \left(\cos^2\left(\frac{\pi}{4} - \frac{\pi}{2^{l+2}}\right) + \sin^2\left(\frac{\pi}{2^{l+p+1}}\right) \right) \quad (9)$$

Proof. The worst-case scenario that maximizes the probability $P(|C_k\rangle)$ under the conditions of the theorem is the following case:

r is a power of 2 ($r = 2^s$),

$$d^k = 2^l r,$$

$$\forall j \neq k \quad d^j - r = \sigma = \frac{r}{2^p} \text{ or } r - d^j = \sigma = \frac{r}{2^p}.$$

The first two conditions minimize the angle φ_k , which is the argument of the cosine in (3), since $\varphi_k = \frac{\pi}{2^{n+1}}(d^k - r)$. If s is the position of the most significant bit in r , then

$$d^k - r \geq 2^{l+1}r - r = r(2^{l+1} - 1) \geq 2^s(2^{l+1} - 1)$$

The third condition maximizes the values of the angles φ_j , which are the arguments of the sines in (3). In this case,

$$\frac{r}{2^p} \leq \frac{d_k}{2^{l+p}} \leq \frac{2^n}{2^{l+p}}$$

Substituting these worst-case values into (3) and taking the most significant bit of d^k as $n-1$ ($n-1 = s+l$), we obtain:

$$\begin{aligned} P(|C_k\rangle) &\leq \frac{1}{M} \left(\cos^2\left(\frac{\pi}{2^{n-(n-1)+1}} - \frac{\pi}{2^{n-(n-1-l)+1}}\right) + \right. \\ &\quad \left. + \frac{1}{M-1} \sum_{j \neq k} \sin^2\left(\frac{\pi\sigma}{2^{n+1}}\right) \right) \leq \\ &\leq \frac{1}{M} \left(\cos^2\left(\frac{\pi}{4} - \frac{\pi}{2^{l+2}}\right) + \sin^2\left(\frac{\pi}{2^{l+p+1}}\right) \right) \end{aligned}$$

An important corollary of Theorems 4.1 and 4.2 is the following set of recommendations for using the algorithm:

The algorithm is sensitive to how fully populated the values of the most significant qubits are. The absence of differences between outliers and useful data in the most significant qubits leads to small values of the cosine argument φ_k in (3), and consequently to an increase in the probability of obtaining the outlier index upon measurement.

The algorithm is also sensitive to the filling of the least significant qubits. Shifting all array values by bits to the right increases the ratio d^k/r , which can significantly decrease the probability of measuring the outlier index in (8).

5 Experimental Evaluation of the Algorithm

The motivation for using the algorithm (1) instead of a median filter is the gain in computational complexity. The median filter requires sorting the input array, which makes its complexity $O(n * M * \log_2 M)$, where M is the array size and n is the bit-width of the numbers compared during sorting. The complexity of the algorithm (1) does not depend on the array size M and is determined only by the bit-width: $O(n)$. Obtaining a reference value (for example, as the average value of the array elements) may require $O(n * M)$ bit-addition operations, leaving the gain of $\log_2 M$ times. If the reference value is taken as the value obtained after applying the algorithm to the previous window (the previous data segment), then computing the average is not required.

Adding a register with element indices that ensures the uniqueness of compared values does not increase computational complexity, since the added qubits do not participate in rotations.

The diagrams in Fig. (5, 6) demonstrate the results of applying the algorithm to arrays of 4 and 8 elements. It can be seen that the algorithm orders the measurement probabilities of element indices in accordance with their values.

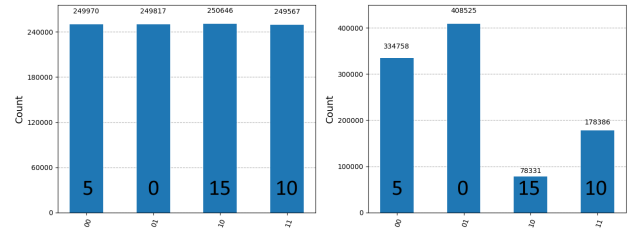


Figure 5. Probabilities of array indices [5 0 15 10] for the reference value $R = 0$ before processing (left) and after applying the rotations (right), 1,000,000 measurements

Below are processing plots of a noisy triangular signal using a classical median filter and a quantum feedback filter, where the reference value (i.e., an analogue of the median) is the measurement result obtained in the previous cycle. The first iteration searches for the median in the selected window. In the second iteration, the resulting median acts as a reference value. At the end of the second and all subsequent iterations, the quantum counter is measured and the window element, whose number was obtained during the measurement, becomes the reference value for the next iteration.

The following presents the results of processing various images with artificially introduced artifacts (highlighted in white) in comparison with a median filter. The settings of the quantum filter were optimized with respect to the maximum and minimum values within the window (array of pixels). In these experiments, the measurement result from the previous cycle also acts as a reference value.

Elimination of repeated elements is achieved via additional $m = \log M$ qubits for each pixel, containing the element index in the array.

Image sources:

<https://sipi.usc.edu/database/>,

<https://rentgenogram.com/>.

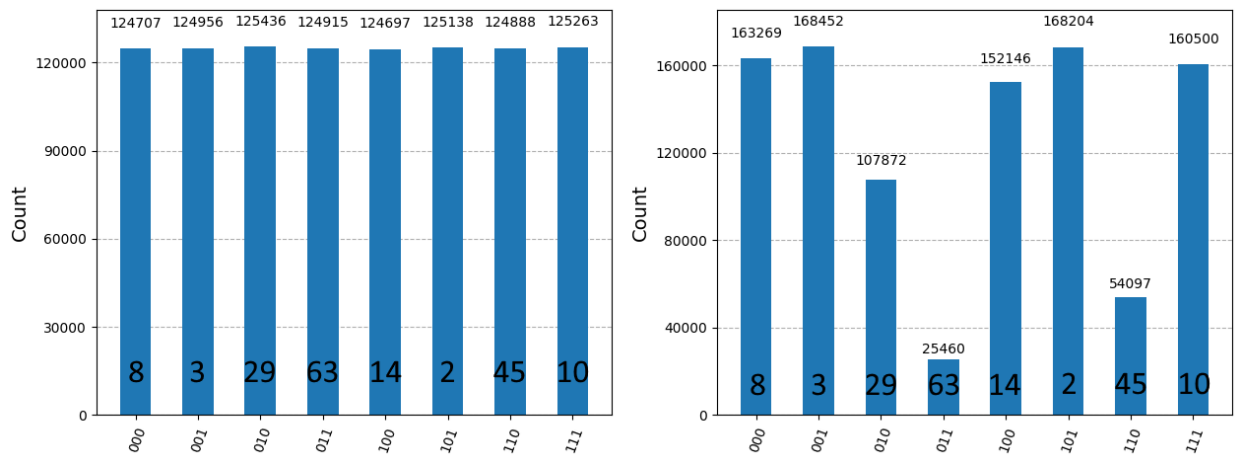


Figure 6. Probabilities of array indices [8 3 29 63 14 2 45 10] for the reference value $R = 0$ before processing (left) and after applying the rotations (right), 1,000,000 measurements

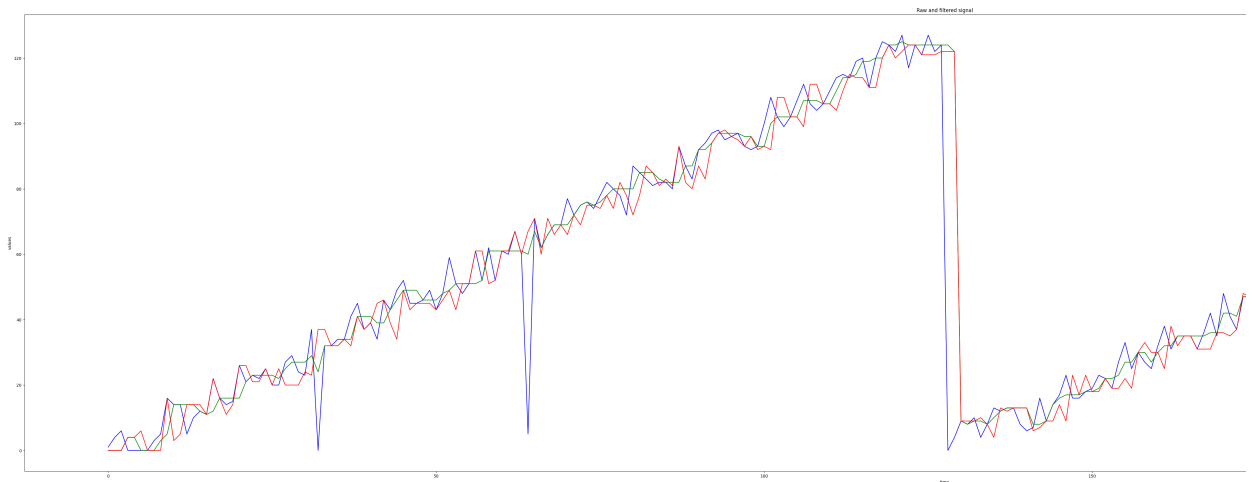


Figure 7. Processing a noisy signal with two outliers (blue) using a median filter (green) and a quantum filter (red)



Figure 8. Processing a black-and-white 8-bit image with an artifact in PNG format, 512*512 pixels. Original image (left), result of the median filter (middle), and result of the quantum feedback filter (right). Window width is 8 pixels.



Figure 9. Processing a black-and-white 8-bit image with an artifact in PNG format, 512*512 pixels. Original image (left), result of the median filter (middle), and result of the quantum feedback filter (right). Window width is 16 pixels.



Figure 10. Processing a black-and-white 8-bit image with an artifact in TIFF format, 512*512 pixels. Original image (left), result of the median filter (middle), and result of the quantum feedback filter (right). Window width is 8 pixels.



Figure 11. Processing a black-and-white 8-bit image with an artifact in TIFF format, 512*512 pixels. Original image (left), result of the median filter (middle), and result of the quantum feedback filter (right). Window width is 16 pixels.

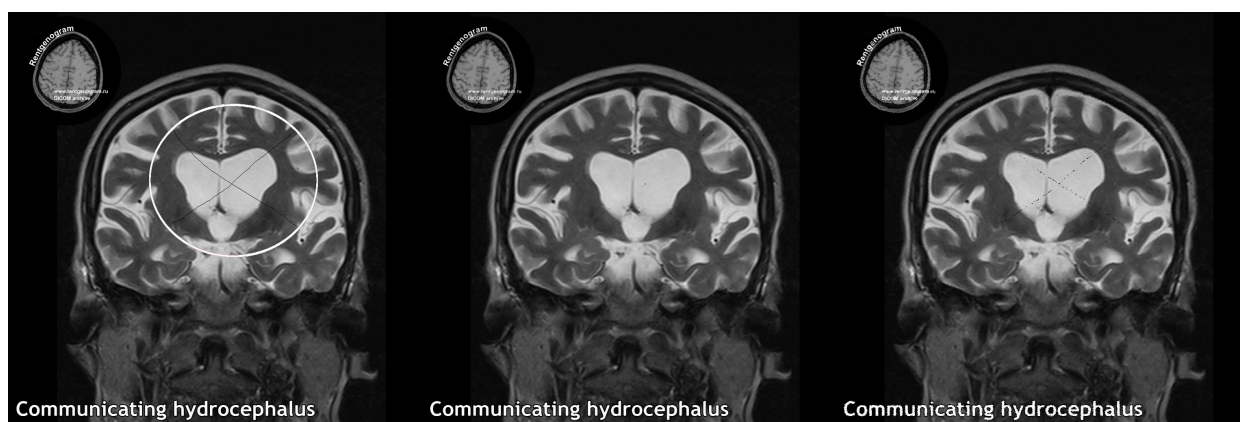


Figure 12. Processing a black-and-white 8-bit MRI image with an artifact in JPG format, 1018*1018 pixels. Original image (left), result of the median filter (middle), and result of the quantum feedback filter (right). Window width is 8 pixels.

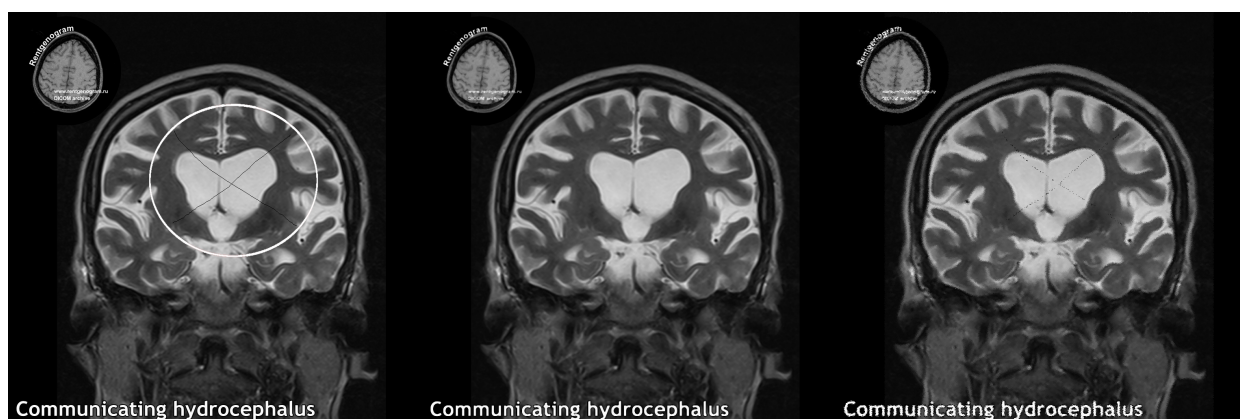


Figure 13. Processing a black-and-white 8-bit MRI image with an artifact in JPG format, 1018*1018 pixels. Original image (left), result of the median filter (middle), and result of the quantum feedback filter (right). Window width is 16 pixels.

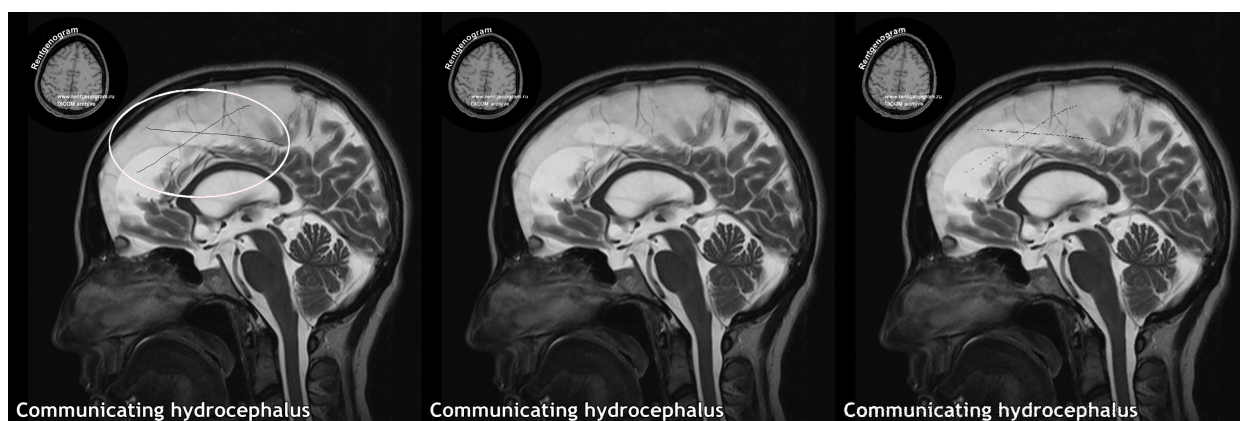


Figure 14. Processing a black-and-white 8-bit MRI image with an artifact in JPG format, 1036*1036 pixels. Original image (left), result of the median filter (middle), and result of the quantum feedback filter (right). Window width is 8 pixels.

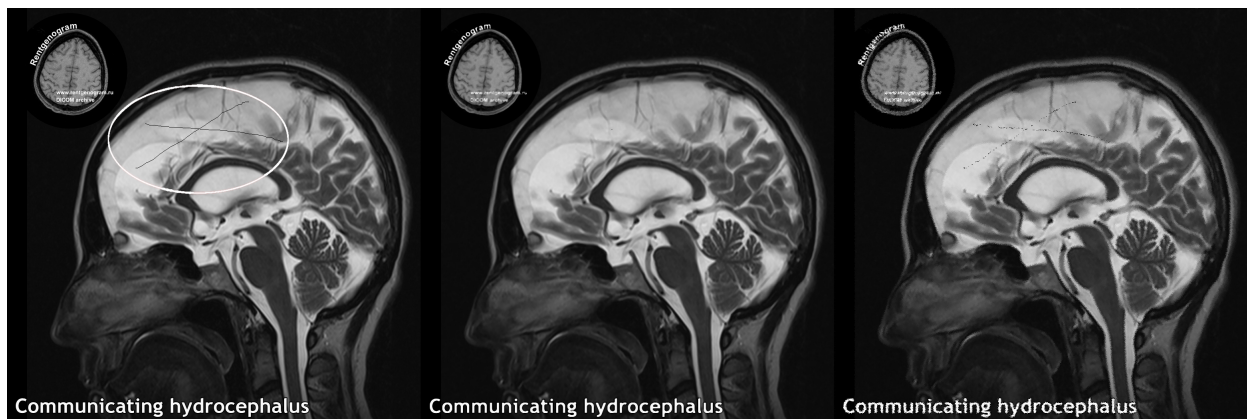


Figure 15. Processing a black-and-white 8-bit MRI image with an artifact in JPG format, 1036*1036 pixels. Original image (left), result of the median filter (middle), and result of the quantum feedback filter (right). Window width is 16 pixels.

6 Conclusion

As a result of the theoretical and experimental study of the algorithm presented in [11], [12], the following recommendations were formulated:

For the algorithm to operate as expected as a replacement for a median filter, it is necessary to ensure the uniqueness of elements in the array being processed.

In register D , which contains a superposition of array elements, one must ensure the absence of data-unfilled qubits on the top and bottom. Unfilled qubits are those that contain zeros for all possible array values.

Increasing the size of the sliding window in image processing leads to better filtering quality.

From the results presented in Section 5, it can be seen that the algorithm performs slightly worse than the median filter, while providing a gain in computational complexity from $O(n * M * \log_2 M)$ to $O(n)$.

Acknowledgements

The work of K.R.Zakharova was supported by the Ministry of Science and Higher Education of the Russian Federation (agreement 075-15-2025-344 dated 29/04/2025 for Saint Petersburg Leonhard Euler International Mathematical Institute at PDMI RAS).

References

- Brassard, G., Hoyer, P., Mosca, M., Tapp, A. Quantum amplitude amplification and estimation. // arXiv preprint quant-ph/0005055, 2000.
- Deutsch, D. Quantum theory, the Church–Turing principle and the universal quantum computer. // Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 400.1818 (1985): 97–117.
- DiVincenzo, D. P. Topics in quantum computers. // Mesoscopic electron transport. Dordrecht: Springer Netherlands, 1997. 657–677.
- Grover, L. K. A fast quantum mechanical algorithm for database search. // Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. 1996.
- Huang, T. S.; Yang, G. J.; Tang, G. Y. A fast two-dimensional median filtering algorithm. // IEEE Transactions on Acoustics, Speech, and Signal Processing. 27 (1): 13–18. 1979.
- Rezaee, Alireza. Partition Fuzzy Median Filter for Image Restoration. // Fuzzy Information and Engineering. 13 (2): 199–210. 2021.
- Safina, L. I. Quantum Amplitude Amplification Algorithm Simulation for Prediction of a Binary Classification Problem. // Lobachevskii Journal of Mathematics 44.2 (2023): 747–756.
- Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. // SIAM review 41.2 (1999): 303–332.
- Tay, David B. Median Autoregressive Graph Filters. // IEEE Signal Processing Letters. 30: 833–837. 2023.
- Von Neumann, J. Mathematical foundations of quantum mechanics: New edition. // Princeton university press, 2018.
- Zakharova, K. R. Quantum algorithm for search of the nearest element (QASNE). // Cybernetics and Physics. Founders: Institute of Problems in Machine Science, RAS 14.1 (2025): 83–88.
- Zakharova, K. R., Chernikov, A. A., Sysoev, S. S. Quantum algorithm for reducing amplitudes in order to search and filter data. // Russian Supercomputing Days. Cham: Springer Nature Switzerland, 2025. 319–332.
- Melnikov, B., Terentyeva, Yu., Chaikovskii, D. Pseudo-geometric version of the traveling salesman problem: application in quantum physics models and a heuristic variant of point placement. // Cybernetics and Physics. Founders: Institute of Problems in Machine Science, RAS 12.3 (2023): 194–200.
- Vakhitov, A. T., Granichin, O. N., Sysoev, S. S. A randomized stochastic optimization algorithm: Its estimation accuracy. // Automation and Remote Control, 67.4 (2006), 589–597.