Quantum Computing for Data Calibration in Parallel Magnetic Resonance Imaging Reconstruction

Girish Babu Reddy
Computer and Information Science Department
University of Massachusetts Dartmouth
North Dartmouth, MA, USA
greddy@umassd.edu

Gulfam Ahmed Saju
Computer and Information Science Department
University of Massachusetts Dartmouth
North Dartmouth, MA, USA
gsaju@umassd.edu

Yi Liu
Computer and Information Science Department
University of Massachusetts Dartmouth
North Dartmouth, MA, USA
yliu11@umassd.edu

Yuchou Chang
Computer and Information Science Department
University of Massachusetts Dartmouth
North Dartmouth, MA, USA
ychang1@umassd.edu

***Abstract*—Parallel imaging techniques, such as GeneRalized Autocalibrating Partially Parallel Acquisitions (GRAPPA) play an important role in Magnetic Resonance Imaging (MRI) by significantly reducing scan times and enhancing patient comfort without compromising image quality. GRAPPA's algorithmic framework involving calibration and synthesis stages is critical in reconstructing high-quality images. However, the computational load of the calibration stage, especially with large convolutional kernel sizes or an increased number of receiver coils, poses a significant bottleneck and limits its efficiency and applicability in clinical settings. In this paper, we introduce an approach by proposing a hybrid software architecture that integrates quantum computing into the GRAPPA reconstruction process. Our method exploits the computational capabilities of quantum computing to accelerate the calibration phase, thereby enabling real-time processing speeds. Through experimentation, we demonstrate that the quantum-enhanced approach can expedite the calibration process to around 10-20 milliseconds of Quantum Processing Unit (QPU) programming time for each Linear Time-Invariant (LTI) system solved. The method maintains the integrity of calibration outcomes, achieving results on par with conventional central processing unit (CPU)-based processes. The result represents the progress towards real-time MRI reconstruction by reducing clinical MRI workflows times, improving patient throughput, and potentially enabling new diagnostic capabilities. Solving LTI system with more attributes on D-Wave quantum computer will be studied to show advantages of QPU to CPU in the future work.**

Keywords— Magnetic Resonance Imaging Reconstruction, D-Wave Quantum Computing, Data Calibration, Quantum Processing Unit, GitHub Codespaces, MATLAB.

I. INTRODUCTION

Magnetic Resonance Imaging (MRI) has revolutionized the field of radiology since its inception in the 1970s. It provides insights into the human body without the need for invasive procedures [16-22]. As a parallel MRI technique, GeneRalized Autocalibrating Partially Parallel Acquisitions (GRAPPA) [1] is widely used in clinical applications. Modeled as a Linear Time-Invariant (LTI) systems, GRAPPA operates through a two-stage process: calibration and synthesis. During calibration, specific patterns within the MRI data are identified to estimate the interpolation coefficients, while the synthesis stage uses these patterns and estimated coefficients to predict missing k-space data and reconstruct the whole k-space from undersampled scan. However, the calibration stage is inherently slow due to many equations solved. When the number of phased-array coils or the size of interpolation kernel expands, the number of equations is significantly increased. This bottleneck significantly impacts the overall speed of the reconstruction process and poses challenges in clinical settings where time is critical.

To accelerate the slow calibration speed that hinders the efficiency of GRAPPA reconstruction in MR imaging, two primary strategies have been proposed to accelerate the process. The first approach involves optimizing the use of phased-array coils by employing dimension reduction techniques like Principal Component Analysis (PCA) [6-9] or by directly minimizing the number of hardware coils during data acquisition. This method essentially reduces the computational load by decreasing the dataset's complexity in the calibration stage. But information may be lost and reconstructed image quality will be degraded. The second strategy focuses on hardware-based enhancements using Field-Programmable Gate Arrays (FPGAs) which are renowned for their ability to perform high-speed and parallel computations, thus providing a pathway to real-time image reconstruction [10]. Extra hardware of FPGA is needed and existing hardware with MRI reconstruction may be modified. Despite the potential of these approaches to improve the speed and efficiency of GRAPPA reconstruction, they fundamentally rely on classical computing paradigms. Classical computers execute algorithms through binary logic gates such as AND, OR, and NOT, which still limits the speed at which the calibration stage of GRAPPA reconstruction can be performed. This inherent limitation emphasizes the need for exploring alternative computing paradigms to transcend these barriers and achieve unprecedented speeds in MR image reconstruction.

Quantum computing [11, 23-29] as an innovative computing paradigm has demonstrated significant advantages over classical

computing, particularly in tasks that involve complex problem-solving and data processing. Unlike classical computers that rely on binary logic gates to manipulate bits, quantum computers utilize quantum annealing or quantum gates to operate on qubits. These qubits [12] have the unique capability to exist in multiple states simultaneously, due to the principles of superposition and entanglement. This enables quantum computers to process vast amounts of data at faster speeds. One of the strengths of quantum computing is its efficiency in solving LTI systems, which are prevalent in various engineering and scientific applications. This capability is especially relevant in the context of MRI, where the calibration stage of GRAPPA reconstruction can be particularly time-consuming.

Motivated by the potential of quantum computing, this paper proposes leveraging quantum computing to accelerate the calibration stage of GRAPPA. By harnessing the power of quantum annealing or quantum gates and the inherent parallelism of quantum computing, we aim to significantly reduce the time required for the calibration of GRAPPA image reconstruction. A hybrid software using GitHub Codespaces and MATLAB is developed to accelerate the calibration of GRAPPA reconstruction. Introduction is presented in the first section of the paper. Method and implementation is given in the second part. Experimental results and conclusion are presented in the third and the fourth sections of the paper.

II. Method and Implementation

A. GRAPPA Reconstruction

The GRAPPA method contains two stages: calibration and synthesis. During the calibration phase, the method focuses on estimating the coefficients of a convolutional kernel, as a process achieved by solving a linear system of equations to obtain a least-squared solution for these coefficients. Subsequently, in the synthesis phase, the missing k-space data are predicted by employing the previously estimated coefficients of the convolution kernel, facilitating the reconstruction of the image. Note that the calibration stage demands a significantly higher time investment compared to the synthesis stage, primarily due to the intricate computations involved in estimating the convolutional kernel's coefficients.

In GRAPPA reconstruction [1], the process is essentially a linear interpolation process. This is demonstrated in equation (1), which mathematically represents the estimation of missing k-space data. The equation is defined as

$$S_j\left(k_y + r\cdot\Delta k_y, k_x\right) = \sum_{l=1}^{L}\sum_{b=-N_b}^{N_a}\sum_{h=-H_l}^{H_r} w_{j,r}(l,b,h)\times S_l\left(k_y + b\cdot R\cdot\Delta k_y, k_x + h\cdot\Delta k_x\right), \quad (1)$$

where S denotes the signals in k-space, w represents the weight coefficients, which are calculated from auto-calibration signal (ACS) data, R stands for the reduction factor, j refers to the target coil, which is interpolated through all other coils counted by l, and b as blocks along phase-encoding direction and h as columns along frequency-encoding direction construct the interpolation kernel. Additionally, k_x and k_y are indices that point to the data positions along the frequency encoding and phase encoding directions, respectively.

For the k-space-based GRAPPA without the coil suppression in a generalized form, the missing k-space data x can be recovered based on the following equation set

$$m = \mathbf{A}x, \quad (2)$$

where both m and $\mathbf{A}$ are ACS for estimating the interpolation coefficients x in the calibration process, and $\mathbf{A}$ represents the acquired k-space data and m is the reconstructed data in the synthesis process. Equation set (2) represents an LTI system.

B. Quantum Computing for Linear Calibration

During the calibration stage, the target vector m is extracted from ACS data. The coefficients vector x undergoes estimation by solving the equation set (2). Constructing the matrix $\mathbf{A}$ with the size U by V involves utilizing ACS data acquired within the vicinity of the target data point (green color) as shown in Fig.1, with the inclusion of more neighborhood points exponentially increasing the matrix's attributes but calibration time will be significantly increased. Consequently, this expanded matrix facilitates the estimation of additional unknown coefficients within the LTI system. In the demonstration depicted in Figure 1, a fitting process utilizes six neighboring k-space points (annotated in blue color) to match the target data point (as green color). Although only one coil is illustrated, neighboring points from other phased-array coils can be integrated, introducing additional unknowns (denoted as x) into the equation set, as depicted on the right side of Figure 1. Consequently, the attribute count V increases correspondingly. The target data points, denoted as m, are fitted within the LTI system. The total number of equations, denoted as U and illustrated in Figure 1, is determined by the number of ACS target data points utilized in the calibration process.

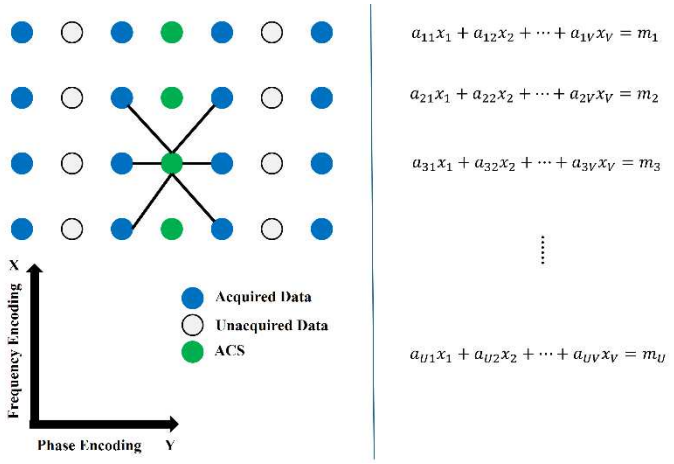

Fig. 1. During the calibration stage, the target ACS data point depicted in green is aligned with its neighboring acquired data points represented in blue. These neighboring data points can span both the phase-encoding and frequency-encoding directions within k-space. As the number of target data points increases, more equations are incorporated into the system, as illustrated on the right side of the figure.

Quantum annealing [13], particularly in systems like D-Wave, is fundamentally described by various equations including the system's Hamiltonian and the dynamic evolution of its quantum state. The time-dependent Schrödinger equation is a foundational principle that describes how the quantum state

The research is supported by UMass Dartmouth Seed Award.

evolves over time within the framework of the system's Hamiltonian. In quantum annealing, this Hamiltonian usually comprises two distinct terms: the problem Hamiltonian and the annealing Hamiltonian. This framework provides a robust mathematical foundation for understanding the intricate dynamics and optimization capabilities of quantum annealing systems like those developed by D-Wave [14] quantum computer system. In quantum annealing, the adiabatic theorem asserts that the system's evolution rate holds profound significance. At its core, this theorem assumes that as long as the system's evolution proceeds at a sufficiently gradual pace, it will steadfastly maintain its ground state throughout the entirety of the process. This principle indicates the delicate balance between the rate of evolution and the system's ability to retain its lowest energy state, thereby illuminating a fundamental aspect of quantum annealing's efficacy and reliability in solving optimization problems.

The application of the D-Wave quantum annealer is successful in solving a LTI system for model predictive control [2]. A quadratic unconstrained binary optimization problem is constructed by deriving an LTI filter [3], which provides innovative approaches in dynamic control problems. Motivated by the promising outcomes in LTI system solutions, we extend the application of quantum annealers to solve the LTI system challenges encountered in the GRAPPA calibration process. A hybrid software via Codespaces and MATLAB is constructed to solve the linear calibration process in GRAPPA reconstruction.

C. *Hybrid Software Implementation on Codespaces and MATLAB*

We use GitHub Codespaces [4] and MATLAB [5] to build a hybrid software to implement quantum computing based GRAPPA calibration and reconstruction. GitHub Codespaces is a cloud-based development environment, which enables to write, run, and debug code directly in their browsers without the need to configure and maintain a development setup on local machines. This platform provides a fully configured development space that is accessible and manageable via GitHub repositories. Note that Codespaces supports the Python programming language and its frameworks, providing flexible configuration options. This flexibility allows for the customization of environments based on specific requirements for Quantum Computing and the processing of MRI k-space data utilizing MATLAB. The architecture of GRAPPA reconstruction process on the hybrid software is demonstrated in Fig.2.

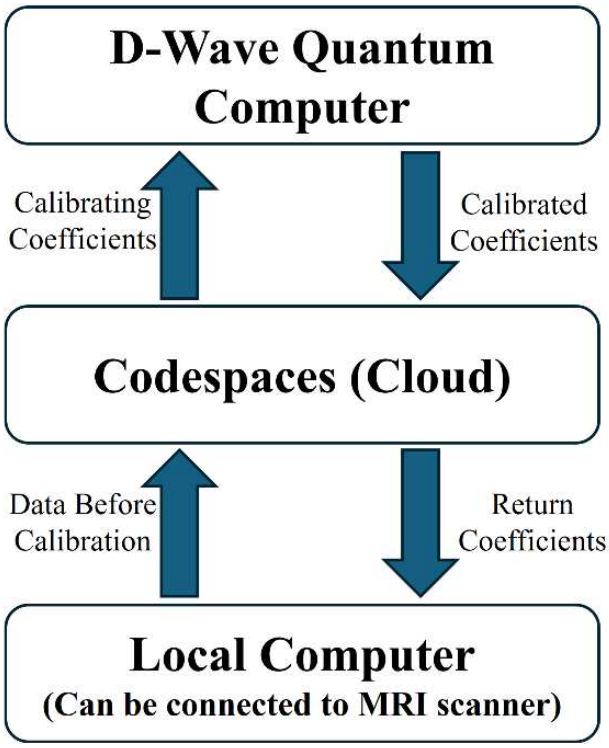

Fig. 2. Workflow diagram illustrating the interaction between a local computer, cloud-based GitHub Codespaces, and a D-Wave Quantum Computer. The process begins with data before calibration, which is sent from the local computer to Codespaces. Codespaces then communicates with the D-Wave Quantum Computer, which performs the calibration and computes the coefficients. These calibration coefficients are returned to Codespaces and subsequently sent back to the local computer which can be connected to a MRI scanner in a hospital. This represents the loop process of calibration and data synthesis in a quantum computing environment. It contains the roles of cloud computing and local processing.

Fig. 2 presents a three-tiered data calibration and processing software system that integrates a local computer, cloud computing via GitHub Codespaces, and a D-Wave Quantum Computer. Initially, uncalibrated data, potentially from a MRI scanner, is preprocessed on a local computer which can interface directly with MRI hardware. This preprocessed data is then uploaded to GitHub Codespaces, as a cloud development environment that facilitates the processing and analysis to calibration data. GitHub Codespaces send data to the D-Wave quantum computer for fast calibration. Leveraging the computational power of quantum computing, the D-Wave system calculates the necessary coefficients to calibrate the data. Once this quantum-calibrated process is complete, the refined coefficients are sent back to Codespaces. From there, the calibrated coefficients are transferred to the local computer where they can be further used for GRAPPA synthesis or to inform subsequent MRI scans, thus creating a feedback loop that enhances the precision and accuracy of the MRI data collected and reconstructed. This system shows the integration of classical and quantum computing resources. It utilizes cloud and local computational strengths to accelerate data calibration processes.

We use GitHub Codespaces for the D-Wave system based quantum computing and exemplify its adaptability. MATLAB on a local computer plays a crucial role in generating data for linear calibration. Following the completion of calibration, we

apply the calibrated coefficients in k-space synthesis on a local computer. This hybrid software framework provides numerous potential benefits which can swiftly initiate and conduct experiments with quantum algorithms. By customizing the Codespaces environment, we can select appropriate computational resources such as CPU, graphics processing unit (GPU), quantum processing unit (QPU) configurations, and install specific versions of software and libraries including D-Wave Ocean SDK [15]. This adaptability is particularly beneficial for managing complex quantum computing models and datasets. It facilitates a more efficient and streamlined research and development process in the quantum computing based MRI reconstruction.

III. Experimental Results

A. MRI Datasets

Two MRI datasets were used for MRI reconstruction. The firstly scanned dataset was acquired on a GE 3T scanner (GE Healthcare, Waukesha, WI) with an 8-channel head coil. The dataset was an axial brain image acquired using a 2D spin echo sequence (TE/TR = 11/700 ms, matrix size = 256 x 256, FOV = 220 mm^2). In the first dataset, a uniform water phantom was scanned using a gradient echo sequence (TE/TR = 10/100 ms, 31.25 kHz band-width, matrix size = 256 x 256, FOV = 250 mm^2). The code was implemented in MATLAB (Natick, MA) on a local computer with using only CPU. The local computer is connected to GitHub Codespaces which is connected to D-Wave quantum computing cloud.

We use the metric Normalized Mean Square Error (NMSE) [30] to evaluate reconstruction performance. NMSE is commonly used to measure the differences between values predicted by a model and the reference values of MR image without k-space data undersampling. It serves to gauge the accuracy of a reconstruction process. NMSE calculates based on reconstruction errors, i.e., the differences between reconstructed and actual values. It provides the square root of the average of these error squares as a measure of the magnitude of errors. It can be formulated as

$$NMSE = \frac{\frac{1}{F}\sum_{f=1}^{F}(p_f - \hat{p}_f)^2}{\frac{1}{F}\sum_{f=1}^{F} p_f^2}, \quad (3)$$

where F is the total number of pixels, p_f is the actual reference values for the f-th pixel, and $\hat{p}$ is the reconstructed value for the k-th pixel. The smaller the RMSE value, the higher the accuracy of the reconstructions.

B. Results

For the first brain dataset, k-space data is undersampled with 10 ACS lines and the outer reduction factor of 2. The NMSE value is 0.0172. The convolutional kernel size is 1 by 1 which is very small. The current implementation of LTI system solution on D-Wave quantum computing cannot accept over 30 attributes for each equation, so the kernel size selection is very small. This will degrade reconstruction image quality. Because k-space has complex data and D-Wave quantum computing cannot accept complex data, 8 coils of complex data are split into 16 coils of real data. Real and imaginary parts of complex data are assigned to different coils. D-Wave Solver Advantage_system4.1 is used.

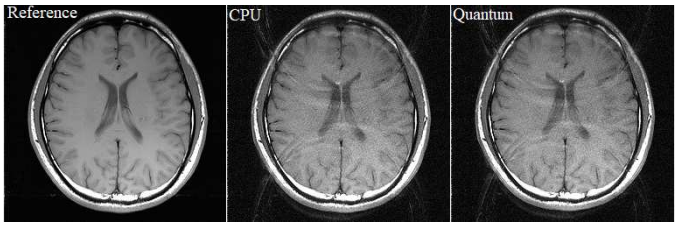

Fig. 3. Reconstructed images by using fully sampled k-space data as the reference image, GRAPPA reconstruction using regular CPU, and GRAPPA reconstruction in which D-Wave quantum computing is used for the calibration and regular CPU is used from the synthesis.

The parameters of D-Wave quantum computing are shown in TABLE I, which also shows the duration for each step of quantum computing for solving one equation set in data calibration. The qpu_sampling_time is the time spent by the QPU to sample solutions. The qpu_anneal_time_per_sample is the time taken for the QPU to perform one quantum annealing process for a sample. The qpu_readout_time_per_sample is the time required to read the qubits' states for one sample after annealing. The qpu_access_time is the total time the computation occupies the QPU including all processes and overheads. The qpu_access_overhead_time is extra time spent on tasks not directly related to computation, such as setup and initialization. The qpu_programming_time is time required to configure the QPU with the problem parameters before annealing. The qpu_delay_time_per_sample is additional delay for each sample not involved in annealing or readout. The total_post_processing_time is time spent on processing the QPU's output to obtain the final solution. The post_processing_overhead_time is overhead time associated with post-processing tasks, like data transfer and formatting. Furthermore, µs represents microseconds and ms denotes microseconds. On the other hand, calibration time on regular CPU of local computer costs 337 µs, which is shorter than the total time of all quantum computing steps shown in TABLE I.

The current quantum computing based calibration cannot show advantages of faster calibration speed to calibration on regular CPU. This may be caused that the convolutional kernel size is too small, so the number of attributes in the LTI system is also tiny. Regular CPU can compute a small number of a LTI system with a small number of attributes in a fast way.

TABLE I. Durations of All Steps in D-Wave Quantum Computing for Dataset 1

Timing Performance	***Duration***
qpu_sampling_time	213 µs
qpu_anneal_time_per_sample	20 µs
qpu_readout_time_per_sample	173 µs
qpu_access_time	15.996 ms
qpu_access_overhead_time	3.419 ms
qpu_programming_time	15.783 ms
qpu_delay_time_per_sample	21 µs

Timing Performance	*Duration*
total_post_processing_time	1 µs
post_processing_overhead_time	1 µs

The second dataset has a similar format as the first Dataset. Fig. 4 presents a comparison of reconstructed images, showcasing different k-space data processing methods in MRI. The reference image is reconstructed using fully sampled k-space data. This high-quality image serves as a benchmark. The second image is obtained through GRAPPA reconstruction using a regular CPU. This demonstrates the capabilities of traditional computational techniques. The third image uses a hybrid approach. D-Wave quantum computing is employed for the calibration phase of GRAPPA reconstruction. A regular CPU is used for the synthesis phase. This combination aims to leverage quantum computing for better calibration accuracy. It potentially leads to improved image reconstruction quality. The comparison illustrates the potential benefits of integrating quantum computing into the MRI reconstruction process.

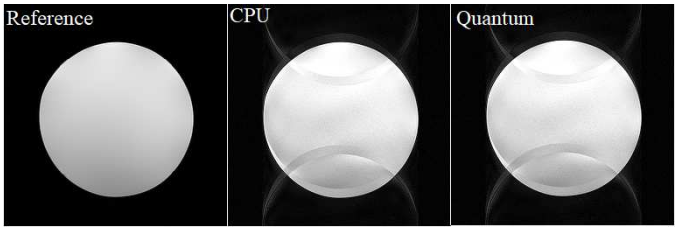

Fig. 4. Reconstructed phantom images by using fully sampled k-space data as the reference phantom image, GRAPPA reconstruction using regular CPU, and GRAPPA reconstruction in which D-Wave quantum computing is used for the calibration and regular CPU is used from the synthesis.

The Table II presents the durations of various steps in D-Wave quantum computing for Dataset 2. The qpu_sampling_time is 210.62 microseconds. Qpu_anneal_time_per_sample is 20 microseconds. Qpu_readout_time_per_sample is 170.04 microseconds. The qpu_access_time is 15.99739 milliseconds. Qpu_access_overhead_time is 547.61 microseconds. The qpu_programming_time is 15.78677 milliseconds. Qpu_delay_time_per_sample is 20.58 microseconds. Total_post_processing_time is 1 microsecond. Post_processing_overhead_time is also 1 microsecond. This table provides a detailed breakdown of the timing performance of each step.

TABLE II. DURATIONS OF ALL STEPS IN D-WAVE QUANTUM COMPUTING FOR DATASET 2

Timing Performance	*Duration*
qpu_sampling_time	210.62 µs
Qpu_anneal_time_per_sample	20 µs
qpu_readout_time_per_sample	170.04 µs
qpu_access_time	15.99739 ms
qpu_access_overhead_time	547.61 µs
qpu_programming_time	15.78677 ms
qpu_delay_time_per_sample	20.58 µs
total_post_processing_time	1 µs
post_processing_overhead_time	1 µs

C. *Limitations of Solving Linear Calibration on D-Wave Quantum Computer*

The equation set depicted in Fig.1 has a crucial bottleneck in the calibration process. The number of attributes denoted by V increases, the local computer's calibration time is proportionally increased. This computational time becomes significantly elongated when dealing with thousands of attributes, potentially extending calibration time to several hours or days. However, the current calibration implementation on the D-Wave quantum computer are limited to handling no more than 30 attributes ($V \leqslant 30$). This constraint degrade the potential advantages of quantum computing, because the reduced attribute capacity results in marginal time savings compared to traditional local computer implementations. Therefore, a significant improvement for future work is clear to enhance the capability of D-Wave's quantum computing for accommodating a larger attribute set within LTI systems. Such an advancement would be necessary to fully leverage the theoretical computational speed of quantum algorithms for more complex and attribute-dense calibration tasks. Furthermore, since the data needs to be submitted to D-Wave Quantum Cloud Service. Data uploading and downloading also cost time usage. This is an disadvantage to the local CPU computation for data calibration in GRAPPA reconstruction.

IV. CONCLUSION

In conclusion, a hybrid software is proposed for data calibration using quantum computing. D-Wave Quantum Cloud Service, GitHub Codespaces and MATLAB are used for data calibration and synthesis in GRAPPA MRI reconstruction. Equation sets of a LTI system are solved by quantum computing. Experimental results show that the quantum computing based solution of a LTI system has the same solutions as solved by a CPU on local computer. Total time of quantum computing steps is slower than that on local CPU, because the number attributes of the LTI system cannot be over 30 using the current implementation of quantum computing. Future work will focus on enlarging the number of attributes and finding advantages of QPU over CPU.

ACKNOWLEDGMENT

The research is supported by UMass Dartmouth Seed Award R32024033000000.

REFERENCES

[1] Mark A Griswold, Peter M Jakob, Robin M Heidemann, Mathias Nittka, Vladimir Jellus, Jianmin Wang, Berthold Kiefer, and Axel Haase, “Generalized autocalibrating partially parallel acquisitions (GRAPPA),” Magnetic Resonance in Medicine, vol. 47, no. 6, pp. 1202-1210, June 2002.

[2] Daisuke Inoue and Hiroaki Yoshida, “Model predictive control for finite input systems using the D-Wave quantum annealer,” Scientific Reports, vol.10, Article number: 1591, 2020.

[3] Prasanna Date and Thomas Potok, “Adiabatic quantum linear regression,” Scientific Reports, vol. 11, Article number: 21905, 2021.

[4] Codespaces. URL: https://github.com/features/codespaces

[5] MATLAB. URL: https://www.mathworks.com/products/matlab.html

[6] Yuchou Chang, Jiming Zhang, Huy Anh Pham, Jingyuan Lyu, and Zhiqiang Li, “Interpretable dimension reduction for MRI channel suppression”, The 44th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Glasgow, July 11 – 15, 2022.

[7] Yuchou Chang and Haifeng Wang, “Kernel principal component analysis of coil compression in parallel imaging”, Computational and Mathematical Methods in Medicine, Volume 2018, Article ID 4254189, 9 pages, 2018.

[8] Feng Huang, Sathya Vijayakumar, Yu Li, Sarah Hertel, George R Duensing, “A software channel compression technique for faster reconstruction with many channels,” Magnetic Resonance Imaging, vol. 26, no. 1, pp. 133-141, 2008.

[9] Shuo Feng, Yudong Zhu, and Jim Ji, “Efficient large-array k-domain parallel MRI using channel-by-channel array reduction,” Magnetic Resonance Imaging, vol. 29, no. 2, pp. 209-215, 2011.

[10] Abdul Basit, Omair Inam, Hammad Omer, “Accelerating GRAPPA reconstruction using SoC design for real-time cardiac MRI”, Computers in Biology and Medicine, vol. 160, 107008, 2023.

[11] Mikio Nakahara and Tetsuo Ohmi, “Quantum computing: from linear algebra to physical realizations”, CRC Press, 2008.

[12] Brian Clegg, “Quantum computing: the transformative technology of the qubit revolution”, Icon Books, 2021.

[13] Arnab Das and Bikas Chakrabarti, “Quantum annealing and related optimization methods”, Lecture Notes in Physics, vol. 679, 2005.

[14] The D-Wave System. URL: https://www.dwavesys.com/

[15] Ocean Developer Tools. URL: https://www.dwavesys.com/solutions-and-products/ocean/

[16] Yuchou Chang, Dong Liang, and Leslie Ying, “Nonlinear GRAPPA: A kernel approach to parallel MRI reconstruction,” Magnetic Resonance in Medicine, vol. 68, no. 3, pp. 730-740, 2012.

[17] Dong Liang, Shanshan Zhang, Jiang Zhu, and Li Feng, “Sensitivity encoding reconstruction with nonlocal total variation regularization,” Magnetic Resonance in Medicine, vol. 65, no. 5, pp. 1384-1392, 2011.

[18] Haifeng Wang, Hanchuan Peng, Yuchou Chang, and Dong Liang, “A survey of GPU-based acceleration techniques in MRI reconstructions,” Quantitative Imaging in Medicine and Surgery, vol. 8, no. 2, pp. 196-208, 2018.

[19] Yuchou Chang, James G. Pipe, John P. Karis, Wende N. Gibbs, Nicholas R. Zwart, and Michael Schär, “The effects of SENSE on PROPELLER imaging,” Magnetic Resonance in Medicine, vol. 74, no. 6, pp. 1598-1608, 2015.

[20] Gulfam Saju, Alan Okinaka, and Yuchou Chang (2023). Exploiting Generative Adversarial Networks in Joint Sensitivity Encoding for Enhanced MRI Reconstruction. In International Symposium on Visual Computing (pp. 443-451). Cham: Springer Nature Switzerland.

[21] Gulfam Saju, Zhiqiang Li, Hui Mao, Tianming Liu, and Y Chang (2023). Suppressing image blurring of PROPELLER MRI via untrained method. Physics in Medicine & Biology, 68(17), 175002.

[22] Alan Okinaka, Gulfam Saju, and Yuchou Chang (2023). Automating kernel size selection in MRI reconstruction via a transparent and interpretable search approach. In International Symposium on Visual Computing (pp. 420-430). Cham: Springer Nature Switzerland.

[23] Zebo Yang, Maede Zolanvari, and Raj Jain (2023). A survey of important issues in quantum computing and communications. IEEE Communications Surveys & Tutorials, 25(2), 1059-1094.

[24] Ur Rasool, Raihan, Hafiz Farooq Ahmad, Wajid Rafique, Adnan Qayyum, Junaid Qadir, and Zahid Anwar. "Quantum computing for healthcare: A review." Future Internet 15, no. 3 (2023): 94.

[25] Sriramakrishnan Chandrasekaran, Rashmi Agrawal, Vishal Dutt, and Narayan Vyas. "Improved Healthcare System with Quantum Computing." In 2023 International Conference on Artificial Intelligence and Smart Communication (AISC), pp. 99-103. IEEE, 2023.

[26] Wei Lin, Haowen Liu, Jing Xu, Lei Shi, Zheng Shan, Bo Zhao, and Yufei Gao. "Quantum machine learning in medical image analysis: A survey." Neurocomputing 525 (2023): 42-53.

[27] Gupta, Shivam, Sachin Modgil, Priyanka C. Bhatt, Charbel Jose Chiappetta Jabbour, and Sachin Kamble. "Quantum computing led innovation for achieving a more sustainable Covid-19 healthcare industry." Technovation 120 (2023): 102544.

[28] Lakshita Aggarwal, Shelly Sachdeva, and Puneet Goswami (2023). Quantum healthcare computing using precision based granular approach. Applied Soft Computing, 144, 110458.

[29] Ubaid Ullah and Begonya Garcia-Zapirain (2024). Quantum machine learning revolution in healthcare: a systematic review of emerging perspectives and applications. IEEE Access.

[30] Rafael C. Gonzalez and Richard E. Woods, “Digital image processing”, Pearson, 2007.