Quantum Optimization for Phase Unwrapping in SAR Interferometry

Kay Glatting, Jan Meyer, Sigurd Huber, and Gerhard Krieger, *Fellow, IEEE*

***Abstract*—Phase unwrapping is the reconstruction of a phase given its values mod 2π. Phase unwrapping is an important image processing technique used in synthetic aperture radar interferometry, in the context of height estimation and ground deformation. By formulating the phase unwrapping problem as a global minimization problem, L^p norm techniques form the backbone of many established phase unwrapping algorithms. The L^0 norm is commonly agreed to produce the best unwrapping solutions, but the necessary brute-force calculations are infeasible for the phase unwrapping of SAR interferograms. Developments in variational quantum algorithms suggest computational advantages, when compared to classical approaches, for a multitude of applications. We therefore investigate the application of quantum algorithms to the phase unwrapping problem and introduce and derive an approximate L^0 norm optimization solver utilizing the quantum approximate optimisation algorithm. This hybrid algorithm uses classical nonconvex parameter optimization to produce optimal parameters for a quantum circuit encoding the L^0 phase unwrapping. Subsequent execution of this quantum circuit allows for an estimate of the solution of the phase unwrapping problem with advantages compared to brute force approaches. We validate our approach for small topographies, by demonstrating improved results when compared to established L^1 optimization.**

***Index Terms*—Interferometric synthetic aperture radar (InSAR), phase unwrapping (PU), quantum approximate optimization algorithm (QAOA), quantum computing.**

I. Introduction

INTERFEROMETRIC synthetic aperture radar (InSAR) is a well-known and powerful remote sensing technique used to measure geophysical features, such as ground deformation and topography. Interferograms are generated by combining two radar images of the same area but acquired with a spatial or temporal baseline, which are utilized for height estimation or ground deformation modeling, respectively.

A. Phase Unwrapping (PU)

InSAR systems encounter physical constraints that limit the measurement of the absolute phase ϕ modulo 2π. This limitation is not unique to SAR but also arises in interferometric synthetic aperture sonar, magnetic resonance imaging, and optical interference. Formally, the relationship is expressed as follows:

$$\phi = \psi + 2\pi k$$

where $\psi \in [-\pi, \pi)^{n\times m}$ represents the measured "wrapped" phase, and $k \in \mathbb{N}_0^{n\times m}$ is the ambiguity correction. PU is the process of extracting the absolute phase $\phi \in [-\pi, \infty)^{n\times m}$, up to a common global constant, from the wrapped phase ψ. Here, $n \times m$ denotes the image dimensions in azimuth and range. This procedure is exemplified in Figs. 1 and 2.

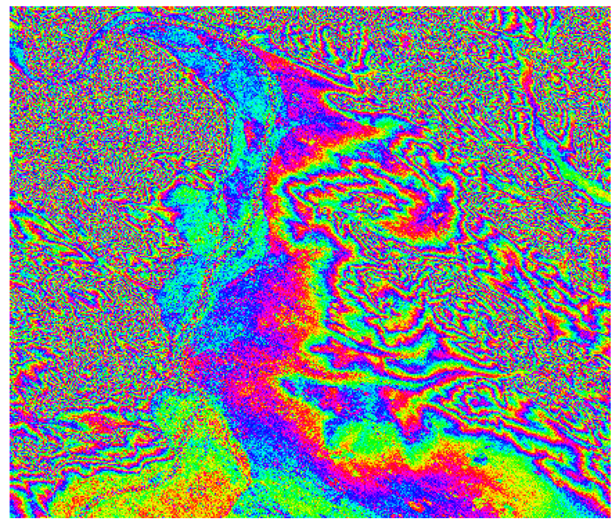


Fig. 1. Simulated wrapped phase, Budapest, Hungary, derived from TanDEM-X digital elevation model using TanDEM-X mission parameters (30 m resolution).

Mathematically, this is a highly ambiguous problem. As a result, virtually all PU algorithms are based on the assumption that the phase surface is smooth enough to produce phase differences less than π for neighboring pixels, as this allows for a straightforward determination of the absolute phase. However, if the true phase surface is highly noisy or the ground resolution is too low, the above-mentioned hypothesis is violated, giving rise to the need for sophisticated algorithms to compensate for the resulting errors.

While various approaches exist for tackling the PU problem, in this article, we will focus on the L^p-norm optimization class of methods and potential advantages gained by incorporating modern quantum computing algorithms and subroutines.

Received 13 August 2024; accepted 3 December 2024. Date of publication 17 December 2024; date of current version 17 January 2025. This work was supported in part by the DLR Quantum Computing Initiative and in part by the German Federal Ministry for Economic Affairs and Climate Action. *(Corresponding author: Kay Glatting.)*

The authors are with the German Aerospace Center (DLR), 82234 Weßling, Germany (e-mail: Kay.Glatting@dlr.de).

https://qci.dlr.de/qua-sar/.

Digital Object Identifier 10.1109/JSTARS.2024.3519701

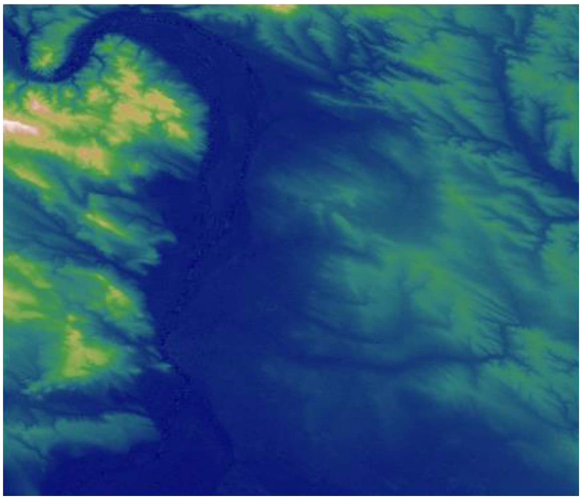


Fig. 2. Unwrapped phase of Fig. 1, graph cut approach. Darker blue areas represent low phase values, with brighter green and brown areas representing high phase values. These areas correspond to valleys and mountains, respectively.

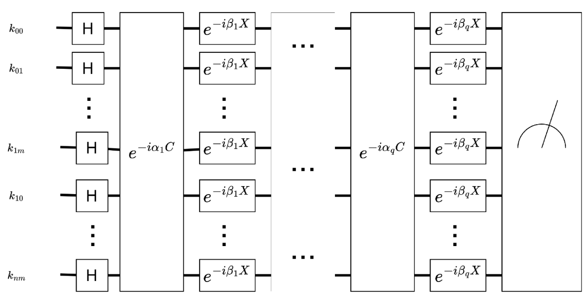


Fig. 3. QAOA L^0 PU quantum circuit.

B. L^p-Norm PU Algorithms

L^p-norm PU algorithms, compare Ghiglia et al.'s [1] work, commonly used in InSAR applications, aim to minimize differences of neighboring pixels, ij and $\tilde{i}\tilde{j}$ in the absolute interferometric phase ϕ by applying a correction term k: Maximize

$$\mathbf{C}(k) = \sum_{(ij,\tilde{i}\tilde{j})\in E} 1 - \left\| k_{ij} - k_{\tilde{i}\tilde{j}} - \left\lceil \frac{\psi_{ij} - \psi_{\tilde{i}\tilde{j}}}{2\pi} - 0.5 \right\rceil \right\|_p \quad (1)$$

where E is the set of edges (the list of adjacent pixel pairs), $\|\cdot\|_p$ denotes the L^p norm, and $\lceil\,\rceil$ denotes rounding up. Here, the edge weights are given by the unwrapped interferometric phase estimation. A more extensive mathematical description of the $p = 0$ case is included in Appendix A.

Employing least-squares methods ($p = 2$) proves efficient for handling large images, yet it has a tendency to smooth out discontinuities, leading to the propagation of errors and the generation of low-accuracy interferograms. The commonly used graph-cut/simplex methods ($p = 1$), discussed in Constantini's [2] work, significantly enhance the unwrapping quality but suffer from scalability issues and may misrepresent certain topographies. Reducing p below 1 further enhances the unwrapping quality at the expense of increased computational complexity. Notably, the L^0-norm, acknowledged for producing the highest quality unwrapped interferograms as indicated by Bioucas-Dias and Valadao [3], faces a challenge as its calculation constitutes an NP (nondeterministic polynomial time) problem, rendering it currently impractical for grid sizes encountered in real-world applications.

Fundamentally, quantum computers have demonstrated the capability to accelerate specific algorithms, scale well with large datasets and efficiently solve some NP problems [4], [5]. In this work, we therefore investigate the possibilities and challenges of utilizing gate-based quantum computers and algorithms to potentially improve the L^0-minimum norm methods for PU.

II. Quantum Approximate Optimization Algorithm (QAOA)

The potential application of quantum algorithms in the realm of L^p-norm optimization PU was recognized previously: Kelany et al. [6], [7] and Otgonbaatar et al. [8] delved into the possibilities of implementing least-squares and network approaches on quantum annealers, respectively.

In the context of our study, we analyze the obstacles, potential benefits, and the practical implementation of L^0-minimum norm methods through the use of the hybrid QAOA, as initially introduced by Farhi et al. [9], [10].

A. QAOA for L^0-Norm PU

The QAOA algorithm consists of two principal components: Determining optimal classical parameters $\alpha \in [-\pi, \pi)^q$ and $\beta \in [0, \pi]^q$ using a quantum computer to create the state

$$\frac{1}{\sqrt{2^N}} \prod_{l=0}^{q-1} \mathbf{U}(\beta_{q-l}, \mathbf{X})\mathbf{U}(\alpha_{q-l}, \mathbf{C}) \sum_{s=0}^{2^N-1} |s\rangle$$

based on these parameters. Here, $|s\rangle$ denotes the quantum state encoding of s, in Dirac-notation. Fig. 3 describes the corresponding quantum circuit we use to create this state. The parameter q dictates the depth of the quantum circuit. The operator $\mathbf{C}$ represents the amount of fulfilled clauses, specifically the zero-contributions in (1), where $p = 0$. An in-detail description of the circuit depicted in Fig. 3, as well as the oracles to implement the gates $\mathbf{U}(\alpha, \mathbf{C})$ and $\mathbf{U}(\beta, \mathbf{X})$ are provided in Appendix B.

To derive the optimal parameters α, β we follow the approach in [4]. We treat phase differences between adjacent pixels as regular four-graphs, where the circuit depth q determines the distance up to which qubits representing adjacent pixels interact. Each pair of adjacent pixels gives rise to a corresponding subgraph of interaction, which can be categorized into distinct subgraph isomorphism classes g due to symmetry. The subgraph isomorphism classes are further described in Appendix C.

B. Subgraph Reduction

Subgraphs belonging to the same isomorphism class identically influence the subsequent calculation. Consequently, we only need to count the corresponding quantity, denoted as ω_g,

Algorithm 1: Phase Unwrapping via L^0 QAOA.

Data: Wrapped phase $\psi \in [-\pi, \pi)^{n\times m}$
Result: Optimal correction term k that maximises $\mathbf{C}(k)$
1. g, ω_g: Count the occurring subgraphs per isomorphism class in ψ;
2. Λ_g: Determine the weights per isomorphism class;
3. Λ: Calculate the overall weights for the reduced representation (Appendix C);
4. α, β: Use a non-convex optimisation algorithm for the reduced optimisation problem to determine the optimal set of control parameters;
5. k: Run circuit in Fig. 3 using the set of control parameters α, β;
6. k: Repeat step 5 to achieve statistical significance, keeping the result with the highest weighting function value $\mathbf{C}(k)$;

and then multiply it by the contribution associated with that particular isomorphism type. Through classical computation of the maximum expected value given by

$$\langle \alpha, \beta|\mathbf{C}|\alpha, \beta\rangle = \sum_g \omega_g \sum_{\substack{s\in\{0,..,2^{n_{q,b}}-1\}\\ \mathbf{C}_{g_0}(s)=1}} \left\|\eta_{q,b}^{\alpha,\beta}(s)\right\|_2^2 \quad (2)$$

to measure the operator C of the quantum state

$$|\alpha, \beta\rangle = \frac{1}{\sqrt{2^N}} \prod_{l=0}^{q-1} \mathbf{U}(\beta_{q-l}, \mathbf{X})\mathbf{U}(\alpha_{q-l}, \mathbf{C}) \sum_{s=0}^{2^N-1} |s\rangle$$

we determine the optimal values for α and β.

The complete derivation of the classical optimization reduction and of (2) is included in Appendix C and E. We now outline the overall procedure for using the L^0 QAOA PU algorithm.

C. PU Via L^0 QAOA

The weight parameters Λ_G and Λ are determined per wrapped phase and uniquely describe the optimization problem (2) in a reduced way, with significantly less computational complexity. An example derivation is described in Appendix E.

It is important to note that the QAOA is an approximate algorithm. In other words, the accuracy of results improves with the circuit depth q, as long as the parameters can be optimized. However, for certain problems, smaller values of q can already theoretically yield the optimal unwrapped phase. These smaller values depend on the specific scenery, making a general derivation of necessary QAOA circuit depths highly difficult. Instead, we aim to find appropriate performance metrics for specific PU scenarios by using experimental/simulation data to guide the derivation.

For this work, we therefore conducted experiments into the performance of our PU QAOA to investigate the potential advantages and shortcomings. Our simulations focus on establishing the influences and effects of grid sizes, necessary bit rates $b \leq 3$, with 2^b times the height of ambiguity equivalent to the maximum height difference we can unwrap, and the circuit depths $q \leq 2$ in scenarios where classical L^1 graph cut methods fail to determine the unwrapped phase.

III. Sample Experiments and Discussion

For both real and simulated quantum computers, the number of accessible qubits is limited. Most quantum circuit simulation software, at the time of writing, supports the simulation of up to 32 logical qubits.

Therefore, we designed our validation experiments with these restrictions in mind, limiting the grid sizes and circuit depth to $q = 1$ (the subgraphs would be bigger than the scenes themselves otherwise), while fixing the bit rate $b_k = 2$ of the ambiguity correction k. This allows us to focus on the possible advantages through the use of the L^0 QAOA PU while maintaining computability for the simulations. The resulting weight functions are of the form

$$\frac{1}{2^{16}}\Bigg(\lambda + 2\sum_{t=0}^{3}\cos(\beta)^{7-2t}\sin(\beta)^{2t+1}(\Lambda_{7-2t,2t+1}\sin(\alpha l)) + 2\sum_{t=1}^{3}\cos(\beta)^{8-2t}\sin(\beta)^{2t}(\Lambda_{8-2t,2t}\cos(\alpha l))\Bigg) \quad (3)$$

where the weight functions Λ depend only on the shape of the Graph G (see Appendix D).

In the following, we provide an in-depth description of four of our experiments.

A. Experiment 1: Small Complicated Terrain 1

For Experiment 1, we consider the terrain model in Fig. 4(a). The associated wrapped phase grid in Fig. 4(b) grid with dimensions 5×3 is

$$\psi_1 = \pi \cdot \begin{bmatrix} -0.45 & -0.9 & 0.65 \\ -0.2 & -0.9 & 0.2 \\ 0.6 & -0.2 & 0 \\ 0 & 0.9 & -0.2 \\ 0.45 & 0.9 & -0.65 \end{bmatrix}.$$

L^p methods with $p \geq 1$ exhibit PU errors. Following the outline Algorithm 1, we count the occurring subgraphs and the overall weights, λ and Λ, for the reduced representation, provided in Appendix D1). Fig. 5 shows the associated optimization landscape according to (3). Its maximum, corresponding to the optimal parameters for the QAOA circuit, is located at $\alpha \approx -0.79242$ and $\beta \approx 1.16553$. We use the Python code provided in Appendix F, utilizing the Qiskit package, to simulate the QAOA circuit presented in Fig. 3. 30 qubits are needed for the $5 \cdot 3 \cdot 2$ grid. The measurements of a quantum mechanical state involve a statistical process, equivalent to sampling from a probability distribution. Hence, repeated sampling is necessary for achieving statistical significance. In all sample experiments, we generated 100 000 samples. Fig. 7 illustrates the corresponding statistics, where the results of equally satisfied clauses are aggregated for both the QAOA (red bars) and sampling the equal superposition state as a reference.

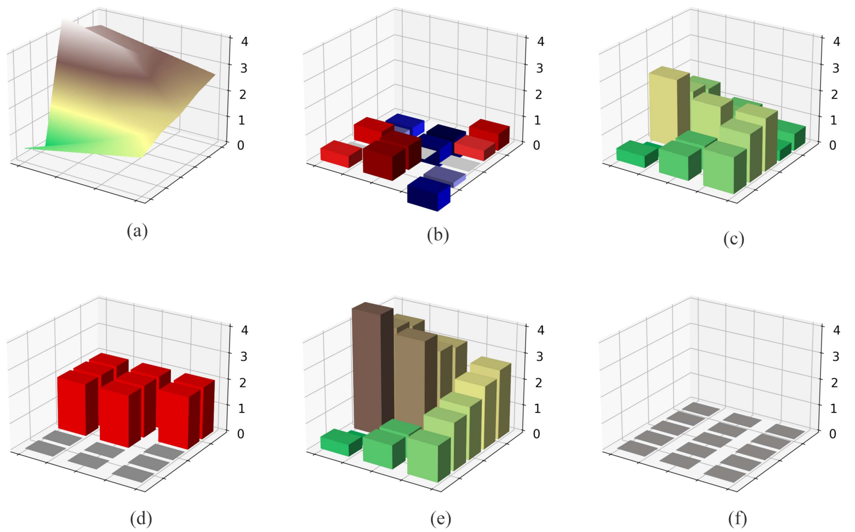


Fig. 4. PU process for Experiment 1. (a) Terrain model. (b) Associated wrapped phase. (c) L^1 graph cut approach unwrap. (d) L^1 unwrapping errors. (e) L^0 QAOA approach unwrap. (f) L^0 unwrapping errors.

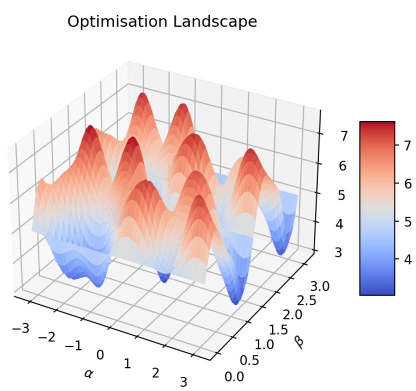


Fig. 5. Optimization landscape of (2), maximum located at $\alpha \approx -0.792421$ and $\beta \approx 1.165526$.

Fig. 7 exhibits an upward shift in the average of the probability distribution from 5.00518 to 8.92482. The maximum returned number of fulfilled clauses increases from 15 to 21. The associated ambiguity correction k enables a flawless reconstruction of the original topography, compare Fig. 4(e) and (f).

We consider the computational expenditure of the L^0 (QAOA). The classical optimization used $5.832 \cdot 10^6$ queries to the cost function, compare Appendix E), and we sampled the QAOA simulation 10^5 times. Compare this to the best conventionally available L^0, i.e., trying all $2^{5\cdot3\cdot2} = 2^{30} = 1.073741824 \cdot 10^9$ configurations individually in a brute force approach. A conventional approach using a graph cut method results in a faulty reconstruction, compare Fig. 4(c) and (d).

B. Experiment 2: Small Complicated Terrain 2

For Experiment 2, we consider the terrain model, Fig. 6(a). The associated wrapped phase interferogram ψ_2, Fig. 6(b), with dimensions 4×4 is

$$\psi_2 = \pi \cdot \begin{bmatrix} 0.1 & 0.5 & 0.5 & 0.9 \\ 0.5 & -0.7 & -0.3 & 0.5 \\ -0.7 & 0.1 & 0.9 & -0.3 \\ 0.1 & 0.5 & 0.5 & 0.9 \end{bmatrix}.$$

Again, following the outline Algorithm 1, we count the occurring subgraphs and the overall weights, λ and Λ, for the reduced representation. The parameters are provided in Appendix D2). The parameter optimization resulted in $\alpha \approx -0.77699$ and $\beta \approx 1.15654$.

Similar to Experiment 1, Fig. 8 exhibits an upward shift in the average of the probability distribution, from 5.62771 to 9.55047. We simulated 32 qubits for the $4 \cdot 4 \cdot 2 = 32$ grid. The maximum returned number of fulfilled clauses increases from 17 to 22. The associated ambiguity correction k again enables a flawless reconstruction of the original topography, compare Fig. 6(e) and (f). As before we generate 10^5 samples. A conventional approach using a graph cut method results in a faulty reconstruction, compare Fig. 6(c) and (d).

C. Experiments 3 and 4: Medium-Sized Terrains

For Experiments 3 and 4 we consider larger scenes, 6×6 and 9×9, that incorporate the terrain of Experiment 1 to explore the behavior of our QAOA approach and the associated computational effort when applied to increasingly larger scenes. The wrapped phase interferograms are

$$\psi_3 = \pi \cdot \begin{bmatrix} -0.45 & -0.9 & 0.65 & 0.3 & 0.2 & -0.1 \\ -0.2 & -0.9 & 0.2 & 0.3 & 0.2 & -0.2 \\ 0.6 & -0.2 & 0 & 0.60.1 & -0.3 & \\ 0 & 0.9 & -0.2 & -0.6 & -0.6 & -0.7 \\ 0.45 & 0.9 & -0.65 & 0.5 & 0.9 & 0.9 \\ 0.2 & 0.3 & 0.65 & -0.7 & -0.2 & -0.2 \end{bmatrix}$$

and $\psi_4 =$

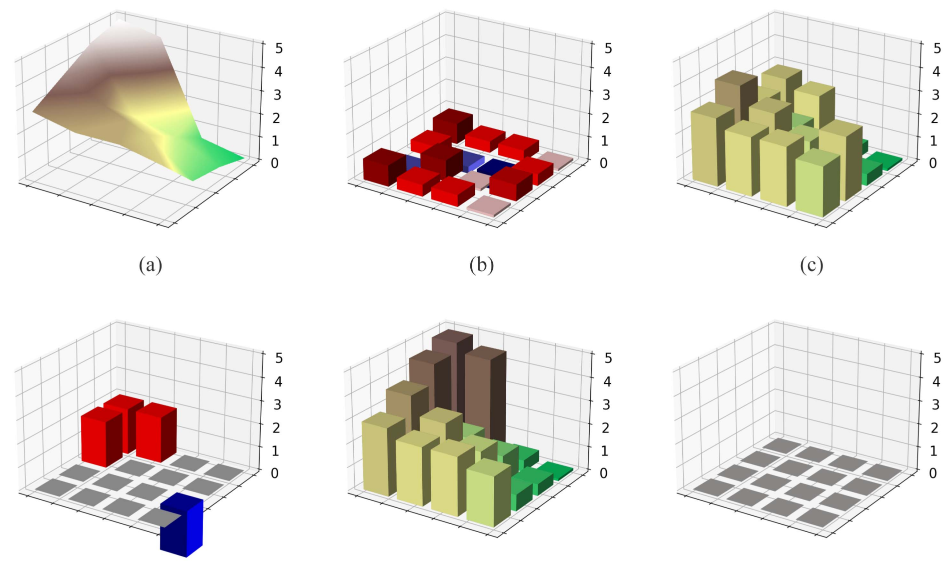


Fig. 6. PU process for Experiment 2. (a) Terrain model. (b) Associated wrapped phase. (c) L^1 graph cut approach unwrap. (d) L^1 unwrapping errors. (e) L^0 QAOA approach unwrap. (f) L^0 unwrapping errors.

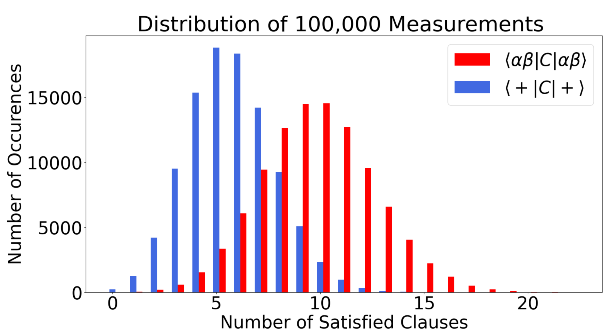


Fig. 7. Statistical evaluation of the QAOA approach for experiment 1, 5×3 scene, bitrate $b = 2$: Effect of the QAOA operation on the number of fulfilled clauses $\mathbf{C}(k)$ compared to the equal superposition state $|+\rangle$.

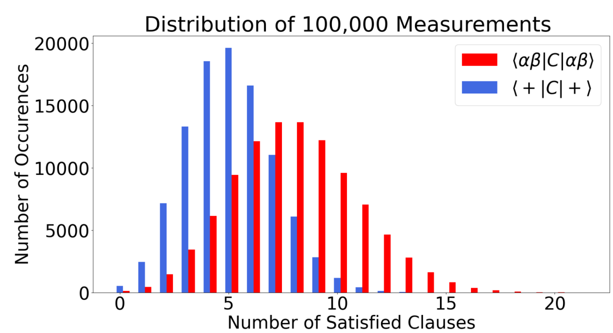


Fig. 8. Statistical evaluation of the QAOA approach for Experiment 2, 4×4 scene, bitrate $b = 2$: Effect of the QAOA operation on the number of fulfilled clauses $\mathbf{C}(k)$ compared to the equal superposition state $|+\rangle$.

$\pi\cdot$

$$\begin{bmatrix} -0.45 & -0.9 & 0.65 & 0.3 & 0.2 & -0.1 & -0.6 & 0.7 & 0.4 \\ -0.2 & -0.9 & 0.2 & 0.3 & 0.2 & -0.2 & -0.7 & 0.5 & 0.2 \\ 0.6 & -0.2 & 0 & 0.6 & 0.1 & -0.3 & 0.8 & 0.2 & -0.65 \\ 0 & 0.9 & -0.2 & -0.6 & -0.6 & -0.7 & 0.35 & -0.6 & 0.5 \\ 0.45 & 0.9 & -0.65 & 0.5 & 0.9 & 0.9 & 0 & -0.2 & -0.6 \\ 0.2 & 0.3 & 0.65 & -0.7 & -0.2 & -0.2 & -0.7 & 0.65 & 0.3 \\ 0.1 & -1. & -0.2 & 0. & 0.3 & 0.3 & 0. & -0.2 & 0. \\ -0.4 & -0.6 & -0.8 & 0.6 & 0. & 0.6 & 0.6 & -0.8 & -0.6 \\ -0.6 & -0.8 & 0.9 & 0.4 & 0. & 0. & 0.4 & 0.9 & -0.8 \end{bmatrix}.$$

Similar to Experiments 1 and 2, a bitrate of 2 is enough to represent the necessary ambiguity correction for the 6×6 scene. As such, $6 \cdot 6 \cdot 2 = 72$ qubits are required for the quantum circuit. The 9×9 scene requires a bitrate of 3 leading to an overall number of qubits $9 \cdot 9 \cdot 3 = 243$. Both qubit numbers are currently too high for us to simulate. To see this, consider the binary vector representation of the $6 \cdot 6$ state during the quantum circuit: Simulating these 72 qubits directly requires 2^{72} bits or $\approx 1.845 \times 10^7$ terabyte.

However, since the classical reduction and optimization scale well in the scene dimensions, we can calculate the optimal parameters α and β for these examples. As before, the overall weights λ and Λ for the reduced representation, are included in Appendix D. The parameter optimizations resulted in $\alpha \approx -2.36679842579177$, $\beta \approx 1.16282$ and $\alpha \approx -0.77322$, $\beta \approx 1.16596$, respectively.

IV. Conclusion and Outlook

Larger scale L^0 PU would significantly improve the quality of InSAR phase products in situations where the amount of phase data is limited or hard to interpret. Possible applications include the improvement of digital elevation models in remote and topographically complicated areas as well as precise deformation monitoring in disaster response scenarios, such as earthquakes.

In our research, we investigated the potential of an L^0-norm PU approach by employing the hybrid QAOA algorithm. Given a circuit depth q, by first classically deducing a set of optimal control parameters α, β and subsequently executing the QAOA quantum circuit repeatedly, optimal solutions can be found. We discussed a graph theoretical approach and the subgraph reduction that reduces the complexity of the classical optimization from $\mathcal{O}(2^N)$ to $\mathcal{O}(2^{2n_{q,b}})$, see Appendix C. This approach scales poorly in the circuit depth q, but q depends only on the neighborhood sizes we wish to consider, with $n_{q,b} << N$ including the necessary bitrate, which is mostly known *a priori* and depends at worst logarithmically on the size of the scenery. Deducing the explicit reduced form for $b = 2, q = 1$, we were successful in the reconstruction of validation samples for small scene sizes and low circuit depths when simulating an ideal quantum computer.

Two significant limitations of the QAOA approach have become apparent in our study. First, the current availability of qubits, both in simulation and on real quantum machines, is insufficient to unwrap large image sizes. We provide an estimate of the necessary qubits for sample scene sizes:

Scene size	est. bitrate b	No.qubits
5×3	2	30
4×4	2	32
6×6	2	72
9×9	3	243
15×15	4	900
200×100	4	80 000
$4\,000 \times 2\,000$	4	32 000 000

The estimated bitrates depend on the scenario and acquisition and scale logarithmically in the height of ambiguity; $b = 4$ allows for the representation of $2^4 = 16$ different absolute phase heights. The number of required qubits scales linearly in the scene dimensions and bitrate b. Current quantum computer developments, compare IBMQ roadmap [11], suggest that L^0-norm PU could be feasible for 8×8 scenes in 2029, with large scene sizes becoming available as the amount of error-corrected qubits increases.

The second limitation manifests in the form of the parameter q. The classical optimization exhibits poor scalability as q increases, conversely improving the accuracy of the QAOA circuit and reducing the necessary sampling. Our simulations suggest a complex balance between the necessary circuit depth q and this desired accuracy.

Nevertheless, our simulation experiments showed significant positive shifts in the probability distributions and allowed for the correct unwrapping of sample scenarios. This, combined with the local nature of the PU problem, suggests that results close enough to optimal solutions can be found even for small circuit depths q. The amount of samples necessary was reduced significantly when compared to the conventional brute force approach. Regular $L^p, p \geq 1$, approaches exhibit unwrapping errors.

We are currently investigating the effects, advantages and costs of using higher circuit depths. We aim to establish values where the tradeoff between accuracy and complexity is favorable. Furthermore, we are investigating the potential of enhancing PU of larger scenes through segmented approaches using the L^0 QAOA algorithm. Focusing the use of the L^0 QAOA on problematic areas and using L^1 methods to unwrap and join up other sectors of the scene should allow for an improvement of the accuracy of larger scale PU problems.

Appendix

We derive the mathematical formulations used in the article, provide coding outlines and supplementary data for the experiments.

A. PU Graph Theoretical Formulation

Consider a wrapped interferometric phase image as a matrix $\psi \in [-\pi, \pi)^{n\times m}$. We formulate the L^0 PU from a graph theoretical perspective.

Let $G = (V, E, w)$ be a graph, representing the adjacency of wrapped phases, V the set of vertices, with each vertex $ij \in V$ associated with the appropriate pixel entry, in row and column, in ψ; $E := \{(ij, \tilde{i}\tilde{j}) | ij, \tilde{i}\tilde{j} \in V; (i = \tilde{i} \wedge |j - \tilde{j}| = 1) \vee (|i - \tilde{i}| = 1 \wedge j = \tilde{j})\}$ be the weighted set of edges, that connect directly neighboring pixels/vertices, either vertically or horizontally; $w : E \to \{0, 1\}, w((ij, \tilde{i}\tilde{j})) = 1 - \|\lceil \frac{\psi_{ij} - \psi_{\tilde{i}\tilde{j}}}{2\pi} - 0.5\rceil\|_0$ the weights through the L^0 norm.

To perform the PU algorithm, we derive a per-pixel correction term, $k \in \mathbb{N}_0^{n\times m}$, that is applied to the appropriate vertices to estimate the unwrapped phase

$$\phi_{ij} = 2\pi k_{ij} + \psi_{ij}. \tag{4}$$

The optimal correction term maximizes the overall weight of the corrected edges/unwrapped phase estimation

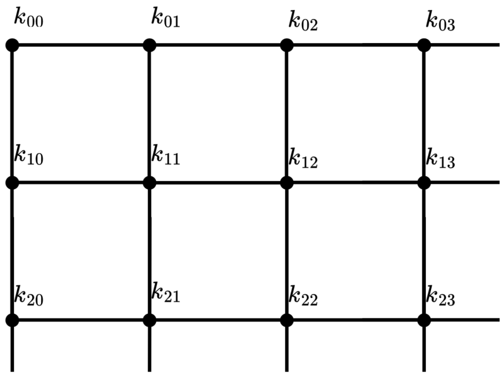
$$\begin{aligned}\sum_{(ij,\tilde{i}\tilde{j})\in E} w((ij, \tilde{i}\tilde{j})) &= \sum_{(ij,\tilde{i}\tilde{j})\in E} 1 - \left\| \left\lceil \frac{\phi_{ij} - \phi_{\tilde{i}\tilde{j}}}{2\pi} - 0.5 \right\rceil \right\|_0 \\ &= \sum_{(ij,\tilde{i}\tilde{j})\in E} 1 - \left\| k_{ij} - k_{\tilde{i}\tilde{j}} - \left\lceil \frac{\psi_{ij} - \psi_{\tilde{i}\tilde{j}}}{2\pi} - 0.5 \right\rceil \right\|_0 \\ &=: \sum_{(ij,\tilde{i}\tilde{j})\in E} \mathbf{C}_{ij,\tilde{i}\tilde{j}}(k) =: \mathbf{C}(k).\end{aligned} \tag{5}$$

The L^0-optimization graph problem is given by the following.

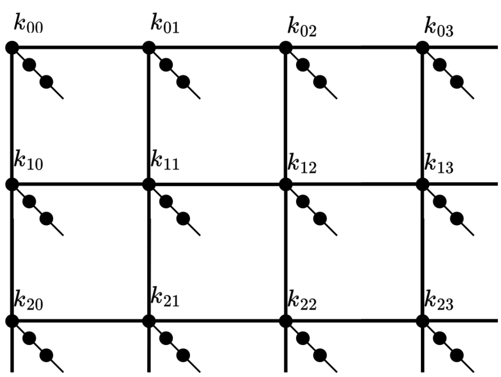
Find $\tilde{k} \in \mathbb{N}_0^{n\times m}$ s.t.

$$\mathbf{C}(\tilde{k}) = \max_{k\in\mathbb{N}_0^{n\times m}} \mathbf{C}(k). \tag{6}$$

The graph G has the overall shape (only denoting vertex labels and correction term k for clarity).



To be able to estimate the highest maximum expected topographic height h_{max} we use a b-bit binary discretization of the vertices, where $b = \lceil \log_2(h_{\text{max}}/h_{\text{amb}}) \rceil$, and h_{amb} is the height of ambiguity. With this b-bit binary discretization of the vertices, keeping the variable k for clarity, $k \in \{0,1\}^{n\times m\times b}$, we get the following.



With this, we derive the binary representation of the cost function (5), keeping the variable C for clarity

$$\sum_{(ij,\tilde{i}\tilde{j})\in E} 1 - \left\| \sum_{\tilde{b}=0}^{b-1} 2^{\tilde{b}} (k_{ij}(\tilde{b}) - k_{\tilde{i}\tilde{j}}(\tilde{b})) - \left\lceil \frac{\psi_{ij} - \psi_{\tilde{i}\tilde{j}}}{2\pi} - 0.5 \right\rceil \right\|_0$$
$$=: \sum_{(ij,\tilde{i}\tilde{j})\in E} \mathbf{C}_{ij,\tilde{i}\tilde{j}}(k) =: \mathbf{C}(k). \quad (7)$$

The binary L^0-optimization graph problem is then given by the following.

Find $\tilde{k} \in \{0,1\}^{n\times m\times b}$ s.t.

$$\mathbf{C}(\tilde{k}) = \max_{k\in\{0,1\}^{n\times m\times b}} \mathbf{C}(k). \quad (8)$$

Since $\{0,1\}^{n\times m\times b}$ is finite at least one solution of (8) exists. This solution does not need to be unique, as distinct correction terms can lead to optimal correction values. In addition, correction terms that only differ by an overall constant always exhibit the same correction value. As PU is a highly ambiguous process, finding one of these possible solutions is accepted to be adequate.

We encode each vertex as an individual qubit, requiring $N = n \cdot m \cdot b$ qubits in total.

Algorithm 2: L^0-PU QAOA Circuit.

Data: $q, \alpha \in [-\pi,\pi)^q, \beta \in [0,\pi]^q$
Result: Estimate $\tilde{k}$ of (8)
1. Initialise all qubits in the state $|0\rangle$;
2. Apply Hadamard gates to all qubits;
3. Set $i = 0$;
4. **while** $i < q$ **do**
   4.1. Apply the weighting operator $U(\alpha_l, \mathbf{C}) = e^{-i\alpha_l \mathbf{C}}$ by applying $e^{-i\alpha_l \mathbf{C}_{ij,\tilde{i}\tilde{j}}}$ to each edge $(ij, \tilde{i}\tilde{j})$ in V;
   4.2. Apply the mixing operator $U(\beta_l, \mathbf{X}) = e^{-i\beta_l \mathbf{X}}$ to each qubit;
   4.3. i++;
end
5. Measure the state;

B. QAOA Circuit

The following algorithm implements the QAOA-circuit 3.

C. QAOA-Circuit Optimization

We encode the graph in the overall qubit state, by associating its flattened binary representation, e.g., a 3-bit representation of $\begin{pmatrix} 0 & 2 \\ 5 & 7 \end{pmatrix} \widehat{=} 000010101111 \widehat{=} |000010101111\rangle \widehat{=} |175\rangle$.

The QAOA-optimization problem, for depth q.

Find $\alpha \in [-\pi,\pi)^q, \beta \in [0,\pi]^q$ that maximize the expectation value of the QAOA circuit

$$\langle \alpha, \beta | \mathbf{C} | \alpha, \beta \rangle \quad (9)$$
$$= \langle 0 | \mathbf{H}^{\otimes N} \mathbf{Q}^\dagger(\alpha,\beta) \mathbf{C} \mathbf{Q}(\alpha,\beta) \mathbf{H}^{\otimes N} | 0 \rangle \quad (10)$$

where $\mathbf{Q}(\alpha,\beta) = \prod_{l=q}^{1} \mathbf{U}(\beta_l, \mathbf{X}) \mathbf{U}(\alpha_l, \mathbf{C})$.

Directly applying classical optimization methods to function (10) is impossible for large scene sizes, as the computation of the function itself scales at $\mathcal{O}(2^N)$. Following the outline in [9] we reduce this complexity significantly. Consider the expectation function (10)

$$\langle 0 | \mathbf{H}^{\otimes N} \mathbf{Q}^\dagger(\alpha,\beta) \mathbf{C} \mathbf{Q}(\alpha,\beta) \mathbf{H}^{\otimes N} | 0 \rangle$$
$$= \frac{1}{2^N} \sum_{z=0}^{2^N-1} \langle z | \mathbf{Q}^\dagger(\alpha,\beta) \mathbf{C} \mathbf{Q}(\alpha,\beta) \sum_{s=0}^{2^N-1} |s\rangle$$
$$= \frac{1}{2^N} \sum_{z=0}^{2^N-1} \sum_{(ij,\tilde{i}\tilde{j})\in E} \langle z | \mathbf{Q}^\dagger(\alpha,\beta) \mathbf{C}_{(ij,\tilde{i}\tilde{j})} \mathbf{Q}(\alpha,\beta) \sum_{s=0}^{2^N-1} |s\rangle. \quad (11)$$

The above-mentioned operator only involves the qubits $ij, \tilde{i}\tilde{j}$ and those qubits whose distance from $ij, \tilde{i}\tilde{j}$ on the graph is q or less, (including those used for binary representation). To see this, consider the innermost operator

$$\mathbf{U}(\alpha_q, \mathbf{C})^\dagger \mathbf{U}(\beta_q, \mathbf{X})^\dagger \mathbf{C}_{(ij,\tilde{i}\tilde{j})} \mathbf{U}(\beta_q, \mathbf{X}) \mathbf{U}(\alpha_q, \mathbf{C}).$$

The factors in the operator $\mathbf{U}(\beta_q, \mathbf{X}) \mathbf{U}(\alpha_q, \mathbf{C})$ that do not involve the qubits $ij, \tilde{i}\tilde{j}$ commute through $\mathbf{C}_{(ij,\tilde{i}\tilde{j})}$ and cancel

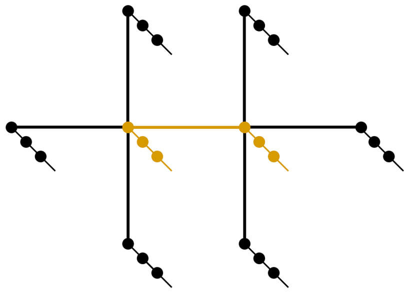


Fig. 9. Shape of the subsets, $q = 1$.

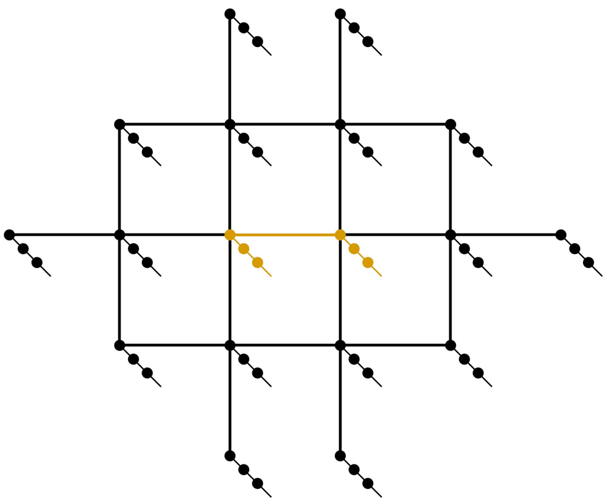


Fig. 10. Shape of the subsets, $q = 2$.

out leaving

$e^{i\alpha_q \mathbf{C}_{\tilde{N}((ij,\widetilde{\tilde{i}\tilde{j}}))}} e^{i\beta_q \mathbf{X}_{(ij,\widetilde{\tilde{i}\tilde{j}})}} \mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})} e^{-i\beta_q \mathbf{X}_{(ij,\widetilde{\tilde{i}\tilde{j}})}} e^{-i\alpha_1 \mathbf{C}_{\tilde{N}((ij,\widetilde{\tilde{i}\tilde{j}}))}}$.

This operator involves the qubits in the combined open neighborhood of $ij, \widetilde{\tilde{i}\tilde{j}}$. Each subsequent application increases the size of this neighborhood by a distance of 1.

With this we can reduce (11)

$$\frac{1}{2^N} \sum_{(ij,\widetilde{\tilde{i}\tilde{j}})\in E} \sum_{z=0}^{2^N-1} \langle z|\tilde{\mathbf{Q}}^\dagger(\alpha,\beta)\mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})}\tilde{\mathbf{Q}}(\alpha,\beta) \sum_{s=0}^{2^N-1} |s\rangle \tag{12}$$

where

$$\tilde{\mathbf{Q}}(\alpha,\beta) = \prod_{l=q}^{1} e^{-i\beta_l \mathbf{X}_{N^{q-l}[(ij,\widetilde{\tilde{i}\tilde{j}})]}} e^{-i\alpha_l \mathbf{C}_{\tilde{N}(N^{q-l}[(ij,\widetilde{\tilde{i}\tilde{j}})])}}.$$

Here $N^l[v] \subset V$ is the open neighborhood of distance l from the elements in $v \subset V$, i.e., $N^l[v] = N^{l-1}[N[v]], l \geq 1, N^0[v] = v$. $\tilde{N}(v) := \{(ij,\widetilde{\tilde{i}\tilde{j}}) \in E | ij \in v\} \subset E$ denotes the edges originating in a subset. These subsets have the same overall shape, apart from those on the edge of the graph for each original edge, depicted in Figs. 9 and 10.

Each subset contains $n_{q,b} = 2b \cdot (q+1)^2$ qubits, with the rest of the graph commuting through (12) and cancelling out, resulting in

$$\frac{1}{2^{n_{q,b}}} \sum_{(ij,\widetilde{\tilde{i}\tilde{j}})\in E} \sum_{z=0}^{2^{n_{q,b}}-1} \langle z|\tilde{\mathbf{Q}}^\dagger(\alpha,\beta)\mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})}\tilde{\mathbf{Q}}(\alpha,\beta) \sum_{s=0}^{2^{n_{q,b}}-1} |s\rangle. \tag{13}$$

Let $\theta_{q,l}(\alpha,\beta) := e^{-i\beta_l \mathbf{X}_{N^{q-l}[(ij,\widetilde{\tilde{i}\tilde{j}})]}} e^{-i\alpha_l \mathbf{C}_{\tilde{N}(N^{q-l}[(ij,\widetilde{\tilde{i}\tilde{j}})])}}$, s.t. $\tilde{\mathbf{Q}}(\alpha,\beta) = \prod_{l=q}^{1} \theta_{q,l}(\alpha,\beta)$.

Developing and summarizing the right-hand side of the expectation value, let $\mho := \{0,1\}, \mathbf{I}_l, \mathbf{X}_l$, the identity and not operator applied to qubit l

$$\prod_{l=q}^{1} \theta_{q,l}(\alpha,\beta) \sum_{s=0}^{2^{n_{q,b}}-1} |s\rangle = \prod_{l=q}^{2} \theta_{q,l}(\alpha,\beta) \prod_{v\in N^{q-1}[(ij,\widetilde{\tilde{i}\tilde{j}})]}$$

$$\prod_{b_i=1}^{b} (\cos\beta_1 \mathbf{I}_{v_{b_i}} - i\sin\beta_1 X_{v_{b_i}}) e^{-i\alpha_1 \mathbf{C}_{\tilde{N}(N^{q-1}[(ij,\widetilde{\tilde{i}\tilde{j}})])}} \sum_{s=0}^{2^{n_{q,b}}-1} |s\rangle$$

$$= \prod_{l=q}^{2} \theta_{q,l}(\alpha,\beta) \prod_{v\in N^{q-1}[(ij,\widetilde{\tilde{i}\tilde{j}})]} \prod_{b_i=1}^{b} (\cos\beta_1 \mathbf{I}_{v_{b_i}} - i\sin\beta_1 X_{v_{b_i}})$$

$$\times \sum_{s=0}^{2^{n_{q,b}}-1} |s\rangle^{-i\alpha_1 \mathbf{C}_{\tilde{N}(N^{q-1}[(ij,\widetilde{\tilde{i}\tilde{j}})])}(s)}. \tag{14}$$

Notation: $s \oplus_l \cdot$ denotes the bitwise moduloaddition in the qubits associated with $N^l[(ij,\widetilde{\tilde{i}\tilde{j}})]$, e.g., for 3-bit representation $s \oplus_0 001010$ flips the state s in the last qubit of ij and the second of $\widetilde{\tilde{i}\tilde{j}}$ equation (15) shown at the bottom of the next page

We can now simplify the expectation value (utilizing the fact that both $\eta_{q,b}^{\alpha,\beta}(s)$ and $\mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})}$ are diagonal operators)

$$= \frac{1}{2^{n_{q,b}}} \sum_{(ij,\widetilde{\tilde{i}\tilde{j}})\in E} \sum_{z=0}^{2^{n_{q,b}}-1} (\eta_{q,b}^{\alpha,\beta}(z)^\dagger \langle z|)\mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})} \sum_{s=0}^{2^{n_{q,b}}-1} (|s\rangle \eta_{q,b}^{\alpha,\beta}(s))$$

$$= \frac{1}{2^{n_{q,b}}} \sum_{(ij,\widetilde{\tilde{i}\tilde{j}})\in E} \sum_{z=0}^{2^{n_{q,b}}-1} \sum_{s=0}^{2^{n_{q,b}}-1} \eta_{q,b}^{\alpha,\beta}(z)^\dagger \underbrace{\langle z||s\rangle}_{\delta_{z,s}} \mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})}(s)\eta_{q,b}^{\alpha,\beta}(s)$$

$$= \frac{1}{2^{n_{q,b}}} \sum_{(ij,\widetilde{\tilde{i}\tilde{j}})\in E} \sum_{s=0}^{2^{n_{q,b}}-1} \eta_{q,b}^{\alpha,\beta}(s)^\dagger \eta_{q,b}^{\alpha,\beta}(s)\mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})}(s)$$

$$= \frac{1}{2^{n_{q,b}}} \sum_{(ij,\widetilde{\tilde{i}\tilde{j}})\in E} \sum_{s=0}^{2^{n_{q,b}}-1} \left\|\eta_{q,b}^{\alpha,\beta}(s)\right\|_2^2 \mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})}(s)$$

$$= \frac{1}{2^{n_{q,b}}} \sum_{(ij,\widetilde{\tilde{i}\tilde{j}})\in E} \sum_{s\in\{0,..,2^{n_{q,b}}-1\}\mathbf{C}_{(ij,\widetilde{\tilde{i}\tilde{j}})}(s)=1} \left\|\eta_{q,b}^{\alpha,\beta}(s)\right\|_2^2. \tag{16}$$

This sum is over all edges, however, if multiple edges give rise to isomorphic subgraphs, their respective values are equivalent, denoting the original edge as g_0. Therefore, we can reduce the expectation value (9) to

$$\langle\alpha,\beta|\mathbf{C}|\alpha,\beta\rangle = \frac{1}{2^{n_{q,b}}} \sum_g \omega_g \sum_{\substack{s\in\{0,..,2^{n_{q,b}}-1\}\\ \mathbf{C}_{g_0}(s)=1}} \left\|\eta_{q,b}^{\alpha,\beta}(s)\right\|_2^2$$

where w_g is the number of occurrences of subgraph g in the original graph G. The amount of qubits necessary is reduced from 2^N to $2^{n_{q,b}}$ and the computational complexity from $\mathcal{O}(2^N)$ to $\mathcal{O}(2^{2n_{q,b}})$

D. Reduction Parameters

1) Experiment 1: $\lambda = 327680$ and $\Lambda =$

$$\begin{bmatrix} 0 & 185088 & 185472 & 166656 & 69376 & 4608 & 896 & 0 \\ 206336 & 171776 & 7360 & -139072 & -160000 & -64448 & -18880 & -3072 \\ 0 & -73600 & -190720 & -386048 & -303104 & -164608 & -65024 & -13440 \\ -103936 & -379520 & 47616 & 52672 & 240384 & 99904 & 37120 & 5760 \\ 0 & 61312 & 83648 & 251840 & 236736 & 124352 & 47168 & 10624 \\ 279296 & 163072 & -47552 & -147392 & -158208 & -65344 & -20288 & -358 \\ 0 & -144384 & -195456 & -168576 & -70784 & -5248 & -896 & 0 \end{bmatrix}.$$

2) Experiment 2: $\lambda = 368640.0$ and $\Lambda =$

$$\begin{bmatrix} 0 & 181760 & 203264 & 196480 & 80768 & 4480 & 768 & 0 \\ 262720 & 189696 & 2112 & -169472 & -185408 & -73856 & -22080 & -3712 \\ 0 & -15744 & -195776 & -444032 & -352960 & -192128 & -78144 & -16768 \\ -108800 & -494720 & 95936 & 54464 & 287488 & 1136006 & 448646 & 7168 \\ 0 & 46720 & 111168 & 316352 & 2849926 & 147008 & 57792 & 13568 \\ 320576 & 193792 & -567686 & -172480 & -183360 & -74560 & -23104 & -4096 \\ 0 & -162560 & -201984 & -200832 & -82944 & -4480 & -768 & 0 \end{bmatrix}.$$

$$\begin{aligned}
&= \prod_{l=q}^{2} \theta_{q,l}(\alpha,\beta) \sum_{s=0}^{2^{n_{q,b}}-1} \left(\sum_{s_{q-1}\in\mho^{n_{q-1}}} |s \oplus_{q-1} s_{q-1}\rangle (\cos\beta_1)^{n_{q-1}-|s_{q-1}|}(-i\sin\beta_1)^{|s_{q-1}|} e^{-i\alpha_1 \mathbf{C}_{\tilde{N}(N^{q-1}[(ij,\tilde{i}\tilde{j})])}(s)} \right) \\
&\overset{\text{Reor-dering}}{=} \prod_{l=q}^{2} \theta_{q,l}(\alpha,\beta) \sum_{s=0}^{2^{n_{q,b}}-1} |s\rangle \underbrace{\left(\sum_{s_{q-1}\in\mho^{n_{q-1}}} (\cos\beta_1)^{n_{q-1}-|s_{q-1}|}(-i\sin\beta_1)^{|s_{q-1}|} e^{-i\alpha_1 \mathbf{C}_{\tilde{N}(N^{q-1}[(ij,\tilde{i}\tilde{j})])}(s\oplus_{q-1}s_{q-1})} \right)}_{\eta^{\alpha,\beta}_{q,1,b}(s):=} \\
&= \prod_{l=q}^{3} \theta_{q,l}(\alpha,\beta) \prod_{v\in N^{q-2}[(ij,\tilde{i}\tilde{j})]} \prod_{b_i=1}^{b} (\cos\beta_2 \mathbf{I}_{v_{b_i}} - i\sin\beta_2 X_{v_{b_i}}) e^{-i\alpha_2 \mathbf{C}_{\tilde{N}(N^{q-2}[(ij,\tilde{i}\tilde{j})])}} \sum_{s=0}^{2^{n_{q,b}}-2} |s\rangle \eta^{\alpha,\beta}_{q,1,b}(s) \\
&= \prod_{l=q}^{3} \theta_{q,l}(\alpha,\beta) \sum_{s=0}^{2^{n_{q,b}}-1} \left(\sum_{s_{q-1}\in\mho^{n_{q-2}}} |s \oplus_{q-2} s_{q-2}\rangle (\cos\beta_2)^{n_{q-2}-|s_{q-2}|}(-i\sin\beta_2)^{|s_{q-2}|} e^{-i\alpha_2 \mathbf{C}_{\tilde{N}(N^{q-2}[(ij,\tilde{i}\tilde{j})])}(s)} \right) \eta^{\alpha,\beta}_{q,1,b}(s) \\
&\overset{\text{Reor-dering}}{=} \prod_{l=q}^{3} \theta_{q,l}(\alpha,\beta) \sum_{s=0}^{2^{n_{q,b}}-1} |s\rangle \underbrace{\left(\sum_{s_{q-2}\in\mho^{n_{q-2}}} (\cos\beta_2)^{n_{q-2}-|s_{q-2}|}(-i\sin\beta_2)^{|s_{q-2}|} e^{-i\alpha_2 \mathbf{C}_{\tilde{N}(N^{q-2}[(ij,\tilde{i}\tilde{j})])}(s\oplus_{q-2}s_{q-2})} \right) \eta^{\alpha,\beta}_{q,1,b}(s \oplus_{q-2} s_{q-2})}_{\eta^{\alpha,\beta}_{q,2,b}(s):=} \\
&\quad \vdots \\
&= \sum_{s=0}^{2^{n_{q,b}}-1} |s\rangle \eta^{\alpha,\beta}_{q,q,b}(s) := \sum_{s=0}^{2^{n_{q,b}}-1} |s\rangle \eta^{\alpha,\beta}_{q,b}(s). \qquad (15)
\end{aligned}$$

3) Experiment 3: $\lambda = 929792.0$ and $\Lambda =$

$$\begin{bmatrix} 0 & 434304 & 508544 & 502144 & 203264 & 11008 & 1664 & 0 \\ 697856 & 541056 & -17856 & -451008 & -501504 & -198720 & -59712 & -10112 \\ 0 & 56256 & -473472 & -1123840 & -887680 & -476544 & -194944 & -43456 \\ -256896 & -1306624 & 177600 & 189760 & 767360 & 294720 & 114752 & 19328 \\ 0 & -62208 & 292224 & 876928 & 756224 & 386048 & 156544 & 37248 \\ 811264 & 529600 & -122176 & -456640 & -489216 & -199232 & -62656 & -10944 \\ 0 & -390784 & -509824 & -508160 & -205312 & -11904 & -1664 & 0 \end{bmatrix}.$$

4) Experiment 4: $\lambda = 2232320$ and $\Lambda =$

$$\begin{bmatrix} 0 & 1081472 & 1219712 & 1203072 & 485888 & 27904 & 3200 & 0 \\ 1683072 & 1231360 & -27136 & -1085888 & -1176192 & -464576 & -137728 & -22912 \\ 0 & 124800 & -1162240 & -2750400 & -2148736 & -1157568 & -467456 & -102016 \\ -713088 & -2976896 & 401344 & 475392 & 1794176 & 704384 & 270656 & 44032 \\ 0 & 1536 & 738304 & 2056448 & 1780928 & 9088000 & 3628800 & 8473600 \\ 1972224 & 1240256 & -326784 & -1099200 & -1151232 & -466368 & -144000 & -24896 \\ 0 & -964224 & -1213824 & -1226240 & -493568 & -29568 & -3200 & 0 \end{bmatrix}.$$

E. Classical QAOA L^0 Reduction for the Circuit Depth $q = 1$ and Bitrate $b = 2$ Case

For a given scenery, we can heavily reduce the complexity of the optimization function. In this work, we consider the case $q = 1, b = 2$. Let $\mho_k^n := \{x \in \{0,1\}^n | \sum_{i=1}^n x_i = k\}$

$$\frac{1}{2^{n_{q,b}}} \sum_g \omega_g \sum_{\substack{s \in \{0,..,2^{n_{q,b}}-1\} \\ \mathbf{C}_{g_0}(s)=1}} \left\| \eta_{1,1,2}^{\alpha,\beta}(s) \right\|_2^2 \overset{s_t := s \oplus_0 t}{=} \frac{1}{2^{n_{q,b}}} \sum_g \omega_g \sum_{\substack{s \in \{0,..,2^{n_{q,b}}-1\} \\ \mathbf{C}_{g_0}(s)=1}}$$

$$\Big\| \cos(\beta)^4 e^{-i\alpha \mathbf{C}_g(s)} - i\cos(\beta)^3 \sin(\beta) \left(e^{-i\alpha \mathbf{C}_g(s_{0001})} + e^{-i\alpha \mathbf{C}_g(s_{0010})} + e^{-i\alpha \mathbf{C}_g(s_{0100})} + e^{-i\alpha \mathbf{C}_g(s_{1000})} \right)$$

$$- \cos(\beta)^2 \sin(\beta)^2 \left(e^{-i\alpha \mathbf{C}_g(s_{0011})} + e^{-i\alpha \mathbf{C}_g(s_{0101})} + e^{-i\alpha \mathbf{C}_g(s_{0110})} + e^{-i\alpha \mathbf{C}_g(s_{1001})} + e^{-i\alpha \mathbf{C}_g(s_{1010})} + e^{-i\alpha \mathbf{C}_g(s_{1100})} \right)$$

$$+ i\cos(\beta) \sin(\beta)^3 \left(e^{-i\alpha \mathbf{C}_g(s_{0111})} + e^{-i\alpha \mathbf{C}_g(s_{1011})} + e^{-i\alpha \mathbf{C}_g(s_{1101})} + e^{-i\alpha \mathbf{C}_g(s_{1110})} \right) + \sin(\beta)^4 e^{-i\alpha \mathbf{C}_g(s_{1111})} \Big\|_2^2$$

$$= \frac{1}{2^{n_{q,b}}} \sum_g \omega_g \sum_{\substack{s \in \{0,..,2^{n_{q,b}}-1\} \\ \mathbf{C}_{g_0}(s)=1}} \Bigg\{ 1 + 2\cos(\beta)^7 \sin(\beta) \left[\sum_{v \in \mho_1^4} \sin(\alpha(\mathbf{C}_g(s) - \mathbf{C}_g(s_v))) \right]$$

$$+ 2\cos(\beta)^6 \sin(\beta)^2 \left[\sum_{v \in \mho_2^4} - \cos(\alpha(\mathbf{C}_g(s) - \mathbf{C}_g(s_v))) + \sum_{v \in \mho_1^4, w \in \mho_1^4} \cos(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_w))) \right]$$

$$+ 2\cos(\beta)^5 \sin(\beta)^3 \left[\sum_{v \in \mho_3^4} - \sin(\alpha(\mathbf{C}_g(s) - \mathbf{C}_g(s_v))) + \sum_{v \in \mho_2^4, w \in \mho_1^4} \sin(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_w))) \right]$$

$$+ 2\cos(\beta)^4 \sin(\beta)^4 \Bigg[\cos(\alpha(\mathbf{C}_g(s) - \mathbf{C}_g(s_4))) - \sum_{v \in \mho_3^4, w \in \mho_1^4} \cos(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_w)))$$

$$+ \sum_{v \in \mho_2^4, w \in \mho_2^4} \cos(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_w))) \Bigg]$$

$$+ 2\cos(\beta)^3 \sin(\beta)^5 \left[\sum_{v \in \mho_1^4} - \sin(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_4))) + \sum_{v \in \mho_2^4, w \in \mho_3^4} \sin(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_w))) \right]$$

$$+ 2\cos(\beta)^2 \sin(\beta)^6 \left[\sum_{v \in \mho_2^4} - \cos(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_4))) + \sum_{v \in \mho_3^4, w \in \mho_3^4} \cos(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_w))) \right]$$

$$+ 2\cos(\beta) \sin(\beta)^7 \left[\sum_{v \in \mho_3^4} \sin(\alpha(\mathbf{C}_g(s_v) - \mathbf{C}_g(s_4))) \right] \Bigg\}.$$

Consider the underlined term

$$\sum_{\substack{s \in \{0,..,2^{n_{q,b}}-1\} \\ \mathbf{C}_{g_0}(s)=1}} 2\cos(\beta)^7 \sin(\beta) \left[\sum_{v \in \mho_1^4} \sin(\alpha(\mathbf{C}_g(s) - \mathbf{C}_g(s_v))) \right]$$

$$= 2\cos(\beta)^7 \sin(\beta) \left[\sum_{\substack{s \in \{0,..,2^{n_{q,b}}-1\} \\ \mathbf{C}_{g_0}(s)=1}} \sum_{v \in \mho_1^4} \sin(\alpha(\mathbf{C}_g(s) - \mathbf{C}_g(s_v))) \right].$$

We observe, since $\mathbf{C}_g(s_v) - \mathbf{C}_g(s_w) = -7; -6; \ldots; 0; \ldots; 7$ and using $\sin(-x) = -\sin(x)$, that we can collect the terms

$$\sum_{l=0}^{7} \sum_{\substack{s \in \{0,..,2^{n_{q,b}}-1\} \\ \mathbf{C}_{g_0}(s)=1}} \left(\sum_{\substack{v \in \mho_1^4 \\ C_g(s) - \mathbf{C}_g(s_v) = l}} \sin(\alpha l) + \sum_{\substack{v \in \mho_1^4 \\ C_g(s) - \mathbf{C}_g(s_v) = -l}} -\sin(\alpha l) \right)$$

$$= \sum_{l=0}^{7} \sin(\alpha l) \underbrace{\sum_{\substack{s \in \{0,..,2^{n_{q,b}}-1\} \\ \mathbf{C}_{g_0}(s)=1}} \left(\sum_{\substack{v \in \mho_1^4 \\ C_g(s) - \mathbf{C}_g(s_v) = l}} 1 - \sum_{\substack{v \in \mho_1^4 \\ C_g(s) - \mathbf{C}_g(s_v) = -l}} 1 \right)}_{\Lambda_{7,l} :=}.$$

Expanding this collection to the other terms, using $\cos(-x) = \cos(x)$ where applicable, we can reduce the optimization function to $7 \cdot 8$ terms

$$= \frac{1}{2^{n_{q,b}}} \Bigg(\lambda + 2\cos(\beta)^7 \sin(\beta) \left[\sum_{l=0}^{7} \Lambda_{7,l} \sin(\alpha l) \right] + 2\cos(\beta)^6 \sin(\beta)^2 \left[\sum_{l=0}^{7} \Lambda_{6,l} \cos(\alpha l) \right]$$

$$+ 2\cos(\beta)^5 \sin(\beta)^3 \left[\sum_{l=0}^{7} \Lambda_{5,l} \sin(\alpha l) \right] + 2\cos(\beta)^4 \sin(\beta)^4 \left[\sum_{l=0}^{7} \Lambda_{4,l} \cos(\alpha l) \right] + 2\cos(\beta)^3 \sin(\beta)^5 \left[\sum_{l=0}^{7} \Lambda_{3,l} \sin(\alpha l) \right]$$

$$+ 2\cos(\beta)^2 \sin(\beta)^6 \left[\sum_{l=0}^{7} \Lambda_{2,l} \cos(\alpha l) \right] + 2\cos(\beta) \sin(\beta)^7 \left[\sum_{l=0}^{7} \Lambda_{1,l} \sin(\alpha l) \right] \Bigg)$$

$$= \frac{1}{2^{16}} \left(\lambda + 2 \sum_{t=0}^{3} \cos(\beta)^{7-2t} \sin(\beta)^{2t+1} (\Lambda_{7-2t,2t+1} \sin(\alpha l)) + 2 \sum_{t=1}^{3} \cos(\beta)^{8-2t} \sin(\beta)^{2t} (\Lambda_{8-2t,2t} \cos(\alpha l)) \right)$$

where the collected terms $\Lambda \in \mathbb{Z}^{7 \times 8}$ and $\lambda = \sum_g \omega_g \sum_{\substack{s \in \{0,..,2^{n_{q,b}}-1\} \\ \mathbf{C}_{g_0}(s)=1}} 1$ uniquely describe the reduced problem.

F. Python Qiskit Circuit for Experiments 1 and 2

We provide the Qiskit code to create the QAOA-circuit for the above-mentioned case as follows:

```
import numpy as np
import qiskit as qi
import qiskit.circuit.library as qi_l

phi1 = np.array([[-0.45, -0.9, 0.65,], [-0.2, -0.7, 0.4], [0.6, 0, 0], #wrapped scene phases
                [0, 0.9, -0.2], [0.45, 0.9, -0.65]]) * np.pi
phi2 = np.array([[0.1, 0.5, 0.5, 0.9], [0.5, -0.7, -0.3, 0.5],
                 [0.7, 0.1, 0.9, -0.3], [0.1, 0.5, 0.5, 0.9]]) * np.pi

al1, be1 = -0.79242, 1.16553
al2, be2 = -0.77699, 1.15654

phi = phi1 #choose wrapped phase
al, be = al1, be1 # choose qaoa parameters

n, m = phi.shape
dphy = np.round((phi[1:, :] - phi[:-1, :]) / (2 * np.pi))
dphx = np.round((phi[:, 1:] - phi[:, :-1]) / (2 * np.pi))

N = 2*n*m
aC0 = np.identity(16, dtype=complex)
aC0[0, 0], aC0[5, 5], aC0[10, 10], aC0[15, 15] = np.ones(4)*e**(-1*j*al)
aC0 = qi_l.UnitaryGate(aC0)

aC1 = np.identity(16, dtype=complex)
aC1[1, 1], aC1[6, 6], aC1[11, 11] = np.ones(3)*e**(-1*j*al)
aC1 = qi_l.UnitaryGate(aC1)

aCm1 = np.identity(16, dtype=complex)
aCm1[4, 4], aCm1[9, 9], aCm1[14, 14] = np.ones(3)*e**(-1*j*al)
aCm1 = qi_l.UnitaryGate(aCm1)

eX = qi_l.UnitaryGate(sc.linalg.expm(-1*j*be*np.array([[0, 1], [1, 0]])))

cir = qi.QuantumCircuit(N, N)
cir.h(range(N))
for iy in np.arange(dphy.shape[0]):
    for ix in np.arange(dphy.shape[1]):
        if dphy[iy, ix] == 0:
            cir.append(aC0, [(iy*m+ix)*2, (iy*m+ix)*2+1,
                             ((iy+1)*m+ix)*2, ((iy+1)*m+ix)*2+1])
        elif dphy[iy, ix] == 1:
            cir.append(aCm1, [(iy*m+ix)*2, (iy*m+ix)*2+1,
                              ((iy+1)*m+ix)*2, ((iy+1)*m+ix)*2+1])
        elif dphy[iy, ix] == -1:
            cir.append(aC1, [(iy*m+ix)*2, (iy*m+ix)*2+1,
                             ((iy+1)*m+ix)*2, ((iy+1)*m+ix)*2+1])

for iy in np.arange(dphx.shape[0]):
    for ix in np.arange(dphx.shape[1]):
        if dphx[iy, ix] == 0:
            cir.append(aC0, [(iy*m+ix)*2, (iy*m+ix)*2+1,
                             (iy*m+ix+1)*2, (iy*m+ix+1)*2+1])
        elif dphx[iy, ix] == 1:
            cir.append(aCm1, [(iy*m+ix)*2, (iy*m+ix)*2+1,
                              (iy*m+ix)*2+2, (iy*m+ix)*2+3])
        elif dphx[iy, ix] == -1:
            cir.append(aC1, [(iy*m+ix)*2, (iy*m+ix)*2+1,
                             (iy*m+ix)*2+2, (iy*m+ix)*2+3])

for i in range(N):
    cir.append(eX, [i])
cir.measure(range(N), range(N))
cir = qi.compiler.transpile(cir)
job_sim = backend.run(cir, shots=100000)
result_sim = job_sim.result()
counts = result_sim.get_counts(cir)
print(counts)
```

References

[1] D. C. Ghiglia and L. A. Romero, “Minimum Lp-norm two-dimensional phase unwrapping,” *JOSA A*, vol. 13, no. 10, pp. 1999–2013, 1996.

[2] M. Costantini, “A novel phase unwrapping method based on network programming,” *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 3, pp. 813–821, May 1998.

[3] J. M. Bioucas-Dias and G. Valadao, “Phase unwrapping via graph cuts,” *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 698–709, Mar. 2007.

[4] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999.

[5] P. W. Shor, “Algorithms for quantum computation: Discrete logarithms and factoring,” in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, IEEE, 1994, pp. 124–134.

[6] K. A. H. Kelany, N. Dimopoulos, C. P. J. Adolphs, B. Barabadi, and A. Baniasadi, “Quantum annealing approaches to the phase-unwrapping problem in synthetic-aperture radar imaging,” in *Proc. 2020 IEEE Int. Conf. Quantum Comput. Eng. (QCE)*, IEEE, 2020, pp. 120–129.

[7] K. A. H. Kelany, N. Dimopoulos, C. P. J. Adolphs, and A. Baniasadi, “Quantum annealing methods and experimental evaluation to the phase-unwrapping problem in synthetic aperture radar imaging,” *IEEE Trans. Quantum Eng.*, vol. 3, 2022, Art. no. 3101420.

[8] S. Otgonbaatar and M. Datcu, “Quantum annealer for network flow minimization in InSAR images,” in *Proc. 13th Eur. Conf. Synthetic Aperture Radar*, 2021, pp. 1–4.

[9] S. Otgonbaatar and M. Datcu, “Quantum annealer for network flow minimization in InSAR images,” in *Proc. 13th Eur. Conf. Synthetic Aperture Radar*, 2021, pp. 1-4. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9472581

[10] E. Farhi and Aram W. Harrow, “Quantum supremacy through the quantum approximate optimization algorithm,” 2016, *arXiv:1602.07674*.

[11] *IBMQ Roadmap*. Accessed: Oct., 05, 2023. [Online]. Available: https://www.ibm.com/roadmaps/quantum/

Kay Glatting received the M.Sc. degree in mathematics (science and engineering) from the Technical University of Munich, Munich, Germany, in 2021, with a focus on dynamical systems, optimization, and mathematical modeling. He is currently working toward the Ph.D. degree in electrical engineering with Friedrich-Alexander-University Erlangen, Erlangen, Germany.

Since then, he has been with Microwaves and Radar Institute, German Aerospace Center, Oberpfaffenhofen, Germany. His research interests include the application of classical and quantum optimization, phase unwrapping, image processing, and machine learning.

Jan Meyer received the M.Sc. degree in physics of complex systems from the Georg-August-University of Göttingen, Göttingen, Germany, in 2022, focusing on quantum mechanics, scientific programming, and machine learning of dynamical systems. He is currently working toward the Ph.D. degree in electrical engineering with Microwaves and Radar Institute, German Aerospace Center, Oberpfaffenhofen, Germany.

In his research, he studies the application of quantum computing and machine learning algorithms for radar remote sensing, particularly for cognitive radar. His research interests include quantum field theory, complex systems, and geometric approaches to machine learning.

Sigurd Huber received the Dipl.-Ing. (M.S.) degree from the Technical University of Munich, Munich, Germany, in 2005, and the Dr.-Ing. (Ph.D.) (Hons.) degree from the Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2014, both in electrical and communication engineering.

Since 2005, he has been with the Microwaves and Radar Institute of the German Aerospace Center, Oberpfaffenhofen, Germany, where he is engaged in the conception of future synthetic aperture radar systems, utilizing advanced multichannel antenna architectures. He was involved in national and international projects in the field of Earth observation, encompassing cooperations with industry partners (Airbus, OHB), as well as research organizations (ESA, NASA/JPL, and JAXA). For the German Earth observation mission proposal Tandem-L and other SAR mission studies he is responsible for SAR system and performance engineering. He is currently leading a team exploring the potentials of quantum computation in the field of radar remote sensing. He has authored or coauthored more than 100 publications and patents. His research interests include digital signal processing, classical and quantum field theory, inverse problems, array processing, and antenna theory.

Gerhard Krieger (Fellow, IEEE) received the Dipl.-Ing. (M.S.) and Dr.-Ing. (Ph.D.) (Hons.) degrees in electrical and communication engineering from the Technical University of Munich, Munich, Germany, in 1992 and 1999, respectively.

From 1992 to 1999, he was with the Ludwig Maximilians University, Munich, Germany, where he conducted multidisciplinary research on neuronal modeling and nonlinear information processing in biological and technical vision systems. Since 1999, he has been with Microwaves and Radar Institute, German Aerospace Center, Oberpfaffenhofen, Germany, where he started working as a Research Associate developing signal processing algorithms for a novel forward-looking radar system employing digital beamforming on receive. From 2001 to 2007, he led the New SAR Missions Group which pioneered the development of advanced bistatic and multistatic radar systems, such as TanDEM-X, as well as innovative multichannel SAR techniques and algorithms for high-resolution wide-swath SAR imaging. Since 2008, he has been the Head of the Radar Concepts Department which currently hosts about 60 scientists focusing on new SAR techniques, missions, and applications. He has been serving as Mission Engineer for TanDEM-X and he made also major contributions to the development of the Tandem-L mission concept, where he led the Phase-0 and Phase-A studies. Since 2019, he holds also a professorship with Friedrich-Alexander-University Erlangen, Erlangen, Germany. He has authored or coauthored more than 100 peer-reviewed journal papers, nine invited book chapters, about 500 conference papers, and more than 30 patents.

Dr. Krieger was the recipient of several national and international awards, including two Best Paper Awards at the European Conference on Synthetic Aperture Radar, two Transactions Prize Paper Awards of the IEEE Geoscience and Remote Sensing Society, and the W.R.G. Baker Prize Paper Award from the IEEE Board of Directors. He has been an Associate Editor for IEEE Transactions on Geoscience and Remote Sensing since 2012. In 2014 and 2024, he was the Technical Program Chair for the European Conference on Synthetic Aperture Radar and in 2014 also as a Guest Editor for the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.