Quantum Clustering for Image Segmentation

A Comprehensive Review

Group 10

Idjourdikene Lounas • Boukhatem Mohamed Rafik • Karballa Chouaib

Miloudi Abdallah Redouane • Khemissa Ahmed

Specialty: Master MIV – 2025/2026

Supervised by: Dr. Naoual MEBTOUCHE

Faculty of Computer Science

University of Science and Technology Houari Boumediene (USTHB)

Algiers, Algeria

December 21, 2025

Abstract

Abstract

This report provides a comprehensive review of quantum clustering techniques applied to image segmentation. We explore the fundamental principles of quantum computing, examine how quantum algorithms can enhance traditional clustering methods, and discuss their application to the challenging task of image segmentation. The review covers both theoretical foundations and practical implementations, highlighting the potential advantages of quantum approaches over classical methods.

Keywords: Quantum Computing, Clustering, Image Segmentation, Quantum Machine Learning, NISQ Algorithms

Conventions Used in This Report

This report uses the following typographical conventions and special callout boxes to highlight important information:

Note

A **Note** box (blue) contains useful background information, additional context, or interesting observations that supplement the main text. Notes provide helpful details that are relevant but not critical to understanding the core material.

Tip

A **Tip** box (teal) offers practical advice, best practices, or suggestions that can help you better understand or apply the concepts being discussed. Tips often highlight shortcuts or insights gained from experience.

Warning

A **Warning** box (orange) indicates important caveats, potential pitfalls, or common mistakes to avoid. Pay attention to warnings to prevent misunderstandings or errors in applying the discussed methods.

Caution

A **Caution** box (red) signals critical information that could lead to serious problems if ignored. Cautions highlight significant limitations, dangerous assumptions, or fundamental constraints.

Definition

A **Definition** box (gray) provides formal definitions of key terms, mathematical concepts, or technical vocabulary. These boxes establish precise meanings for important terminology used throughout the report.

Key Concept

A **Key Concept** box (orange accent) highlights fundamental ideas, core principles, or essential takeaways that are central to understanding the material. These boxes summarize the most important points.

Example

An **Example** box (light gray) presents concrete illustrations, worked problems, or practical demonstrations of the concepts being explained. Examples help bridge theory and application.

Additional conventions:

- *Italics* are used for emphasis and for introducing new terms.
- **Bold** is used for important concepts and terminology.
- `Monospace font` is used for code, algorithms, and technical notation.
- Mathematical expressions use standard notation, with quantum states written in Dirac notation (e.g., $|\psi\rangle$, $\langle\phi|$, $\langle\phi|\psi|\phi|\psi\rangle$).

Contents

Introduction

Image segmentation stands as one of the most fundamental and challenging problems in computer vision and image processing. At its core, segmentation involves partitioning a digital image into multiple segments or regions, where each segment corresponds to meaningful objects or parts of objects. This process serves as a critical preprocessing step for numerous applications, including medical image analysis, autonomous driving, satellite imagery interpretation, and object recognition systems.

Traditionally, clustering algorithms have been the workhorses of image segmentation. Methods such as K-means, fuzzy C-means, and spectral clustering have demonstrated remarkable success in grouping pixels based on color, texture, or spatial proximity. However, as image resolutions increase and the demand for real-time processing grows, these classical approaches encounter significant computational bottlenecks. The curse of dimensionality, coupled with the need to process millions of pixels simultaneously, pushes classical algorithms to their limits.

> **Key Concept**
>
> Quantum computing offers the potential to process information in fundamentally different ways than classical computers, potentially providing speedups for certain computational tasks including clustering. By leveraging quantum phenomena such as superposition and entanglement, quantum algorithms can explore exponentially large solution spaces more efficiently than their classical counterparts.

This report provides a comprehensive review of quantum clustering techniques and their application to image segmentation. We explore how quantum mechanical principles can be harnessed to overcome the computational limitations of classical methods, examining both theoretical foundations and practical implementations.

Objectives of This Review

This report aims to:

- Establish a solid foundation in classical clustering methods and their application to image segmentation
- Identify the computational bottlenecks that motivate the exploration of quantum approaches
- Review the state-of-the-art in quantum image representation and processing
- Provide a comparative analysis of quantum clustering algorithms for image segmentation
- Discuss current limitations and future research directions

Report Structure

The remainder of this report is organized as follows:

Chapter 1 presents the classical paradigm of image segmentation via clustering, covering fundamental definitions, classical algorithms, and computational challenges.

Chapter 2 explores quantum clustering as a new frontier, discussing quantum image representations and reviewing quantum segmentation algorithms.

Conclusion summarizes key findings and outlines future research directions.

Appendix A provides a primer on quantum information processing for readers less familiar with quantum computing concepts.

Appendix B covers the fundamentals of fuzzy logic as applied to segmentation.

1 Quantum Clustering for Image Segmentation

1.1 Introduction

Image segmentation represents a fundamental task in computer vision that involves partitioning digital images into meaningful regions or objects. Traditional algorithmic approaches, while effective for many applications, face inherent limitations when dealing with complex, high-dimensional data characterized by non-linear relationships and noisy environments. Quantum clustering emerges as a novel computational paradigm that leverages principles from quantum mechanics to potentially overcome these challenges.

The conceptual foundation of quantum clustering rests upon treating data points as quantum particles within a potential landscape. This approach constructs a Hamiltonian operator whose ground state—corresponding to the minimum energy configuration—encodes the optimal clustering solution. Rather than employing iterative assignment algorithms common to classical methods, quantum clustering seeks global solutions through the natural evolution of quantum systems toward their minimal energy states.

Key quantum mechanical principles employed in this paradigm include:

- Wave function superposition, allowing data points to exist in multiple cluster states simultaneously
- Quantum entanglement, enabling non-local correlations between pixel assignments
- Quantum tunneling, providing mechanisms to escape local minima in optimization landscapes
- Quantum interference, amplifying probabilities of optimal configurations while suppressing suboptimal ones

The mathematical formulation typically begins with the time-independent Schrödinger equation:

$$\hat{H}\psi(\mathbf{x}) = E\psi(\mathbf{x}) \tag{1.1}$$

where $\hat{H}$ represents the Hamiltonian operator, $\psi(\mathbf{x})$ denotes the wave function, and E corresponds to energy eigenvalues. In quantum clustering applications, the Hamiltonian is specifically designed such that its ground state wave function exhibits high amplitude in regions of high data density, thereby naturally identifying clusters.

1.2 Classical Image Segmentation Techniques

1.2.1 K-Means Clustering

K-means clustering represents a centroid-based partitioning algorithm that divides a set of n observations into k predefined clusters. Each observation is assigned to the cluster whose mean (centroid) yields the smallest squared Euclidean distance. The algorithm minimizes the within-cluster sum of squares:

$$J = \sum_{i=1}^{k} \sum_{\mathbf{x} \in C_i} |\mathbf{x} - \boldsymbol{\mu}_i|^2 \tag{1.2}$$

where $\boldsymbol{\mu}_i$ denotes the centroid of cluster C_i. The algorithm proceeds through iterative refinement steps of assignment and centroid update until convergence criteria are satisfied.

1.2.2 Fuzzy C-Means Clustering

Fuzzy C-means clustering extends the K-means paradigm by permitting partial membership, where each data point may belong to multiple clusters with varying degrees of association. The objective function incorporates membership weights:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m |\mathbf{x}_i - \mathbf{c}j|^2 \tag{1.3}$$

where uij represents the membership degree of point $\mathbf{x}_i$ to cluster j, $\mathbf{c}_j$ denotes the cluster center, and $m > 1$ serves as a fuzziness parameter controlling the overlap between clusters.

1.2.3 Graph-Based Segmentation

Graph-based segmentation methods represent images as weighted graphs $G = (V, E)$, where vertices V correspond to pixels or superpixels, and edges E connect neighboring elements with weights reflecting similarity measures. Segmentation involves partitioning the graph by optimizing cut-based criteria.

Two fundamental approaches for graph partitioning are the minimum cut and maximum cut problems:

Minimum Cut seeks the partition that minimizes the sum of weights of edges crossing the partition boundary:

$$\min_{A \subset V} \; cut(A, V \setminus A) = \min_{A \subset V} \sum_{u \in A, v \notin A} w(u, v) \tag{1.4}$$

where $w(u, v)$ represents the weight (similarity) between vertices u and v. However, the minimum cut criterion alone often produces trivial partitions isolating single vertices.

Maximum Cut seeks the partition that maximizes the sum of weights of edges crossing between partitions:

$$\max_{A \subset V} \; cut(A, V \setminus A) = \max_{A \subset V} \sum_{u \in A, v \notin A} w(u, v) \tag{1.5}$$

This formulation maximizes dissimilarity between partitions but can produce unbalanced segments.

To address the limitations of basic cut criteria, the **normalized cut** objective seeks balanced partitions A and B by minimizing:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{1.6}$$

where $cut(A, B) = \sum_{u \in A, v \in B} w(u, v)$ represents the total weight of edges crossing the partition boundary, and $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ denotes the total connection from nodes in A to all nodes in the graph. This normalization prevents trivial solutions and encourages partitions with balanced internal coherence.

1.2.4 Spectral Clustering

Spectral clustering operates on the spectral decomposition of similarity matrices derived from data. The algorithm constructs a similarity matrix S, computes the corresponding graph Laplacian $L = D - S$ (where D is the degree matrix), and performs eigenvalue analysis to obtain a lower-dimensional embedding. Clustering then proceeds in this reduced eigen-space, typically via K-means, enabling separation of non-linearly separable clusters.

1.3 Quantum Computing Primitives

1.3.1 Variational Quantum Algorithms

Variational Quantum Algorithms constitute a class of hybrid quantum-classical computational frameworks designed for near-term quantum devices. These algorithms employ parameterized quantum circuits, known as ansätze, to prepare trial quantum states. A classical optimization loop adjusts circuit parameters to minimize a cost function that encodes the problem of interest. The general architecture comprises:

- A parameterized quantum circuit $U(\boldsymbol{\theta})$ generating trial states $|\psi(\boldsymbol{\theta})\rangle$
- A cost function $C(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$ representing the problem Hamiltonian H
- Classical optimization routines that iteratively update parameters $\boldsymbol{\theta}$ to minimize $C(\boldsymbol{\theta})$

1.3.2 Variational Quantum Eigensolver

The Variational Quantum Eigensolver represents a specific instantiation of variational quantum algorithms focused on determining the ground state energy of quantum Hamiltonians. Given a Hamiltonian H, VQE seeks to approximate:

$$E_0 = \min_{\boldsymbol{\theta}} \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle \tag{1.7}$$

The algorithm has found applications beyond quantum chemistry, serving as a template for optimization problems where the Hamiltonian encodes problem objectives.

1.4 Quantum Formulations and Encodings

1.4.1 Quadratic Unconstrained Binary Optimization

Quadratic Unconstrained Binary Optimization provides a mathematical framework for encoding combinatorial optimization problems, including image segmentation tasks. A QUBO problem is defined as:

$$\min_{\mathbf{x} \in 0,1^n} \left(\sum_{i \leq j} Q_{ij} x_i x_j \right) = \min_{\mathbf{x} \in 0,1^n} \mathbf{x}^T Q \mathbf{x} \tag{1.8}$$

where $\mathbf{x}$ represents a vector of n binary decision variables, and Q denotes an $n \times n$ upper-triangular matrix of real coefficients. The QUBO formulation naturally maps to Ising spin glass models through the transformation $s_i = 2x_i - 1$, where $s_i \in -1, +1$, yielding the equivalent Ising Hamiltonian:

$$H_{Ising} = \sum_i h_i s_i + \sum_{i<j} J_{ij} s_i s_j \tag{1.9}$$

This mathematical equivalence enables implementation on both quantum annealers and gate-based quantum processors.

1.4.2 Quantum Approximate Optimization Algorithm

The Quantum Approximate Optimization Algorithm represents a hybrid quantum-classical algorithm designed for combinatorial optimization problems. For a problem with n variables and cost function $C(\mathbf{z})$ where $\mathbf{z} \in 0, 1^n$, QAOA prepares parameterized quantum states through alternating application of problem and mixing Hamiltonians:

$$|\psi(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{k=1}^{p} e^{-i\beta_k H_B} e^{-i\gamma_k H_C} |+\rangle^{\otimes n} \tag{1.10}$$

where:

- H_C represents the problem Hamiltonian encoding $C(\mathbf{z})$
- $H_B = \sum_{i=1}^{n} \sigma_i^x$ denotes the transverse field mixing Hamiltonian
- p indicates the number of alternating layers (circuit depth)
- $\boldsymbol{\beta}, \boldsymbol{\gamma}$ represent tunable parameters optimized classically

The expected value $\langle H_C \rangle$ serves as the objective for classical optimization routines.

1.4.3 Projected Gradient Entanglement

Projected Gradient Entanglement constitutes a quantum-inspired classical algorithm that simulates quantum mechanical behavior through iterative projection and gradient operations. The algorithm maintains quantum-like representations of cluster centers and employs operations mimicking quantum superposition and measurement collapse. Key algorithmic steps include:

1. Initialization of quantum state representations for cluster centroids
2. Application of entanglement operations to establish quantum correlations
3. Projection onto measurement basis simulating quantum collapse
4. Gradient-based updates within the potential landscape
5. Iterative refinement until convergence criteria are satisfied

1.4.4 Adiabatic Binary Encoding

Adiabatic Binary Encoding formulates clustering problems as binary optimization tasks suitable for adiabatic quantum computation. For n data points and k clusters, binary variables $x_{i\alpha} \in 0, 1$ indicate assignment of point i to cluster α. The clustering Hamiltonian typically combines similarity terms with constraint enforcement:

$$H = \sum_{i<j} d_{ij} \sum_{\alpha} x_{i\alpha} x_{j\alpha} + \lambda \sum_{i} \left(\sum_{\alpha} x_{i\alpha} - 1 \right)^2 \tag{1.11}$$

where d_{ij} represents distance measures between points, and λ penalizes invalid assignment configurations.

1.4.5 Adiabatic Cluster Encoding

Adiabatic Cluster Encoding extends binary encoding approaches by directly representing cluster assignments within quantum state spaces. Rather than employing binary indicator variables, ACE utilizes quantum registers to encode cluster membership information. The Hamiltonian construction incorporates multiple components:

$$H_{ACE} = H_{intra} + H_{inter} + H_{constraints} \tag{1.12}$$

where H_{intra} promotes similarity within clusters, H_{inter} encourages separation between distinct clusters, and $H_{constraints}$ ensures valid assignment configurations.

1.5 Quantum Computational Techniques

1.5.1 Hybrid Quantum-Classical Algorithms

Hybrid quantum-classical algorithms represent computational frameworks that leverage both quantum and classical processing resources. Quantum processors execute computationally demanding subroutines such as state preparation and expectation estimation, while classical systems handle optimization, control logic, and error mitigation. This synergistic approach aims to overcome current quantum hardware limitations while exploiting potential quantum advantages for specific computational tasks.

1.5.2 Quantum Annealing

Quantum annealing constitutes a computational paradigm that exploits quantum tunneling and thermal effects to find global minima of optimization problems. The method implements time-dependent Hamiltonians that evolve from simple initial configurations to complex problem encodings:

$$H(t) = A(t) \sum_{i} \sigma_i^x + B(t) \left(\sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z + \sum_{i} h_i \sigma_i^z \right) \tag{1.13}$$

where $A(t)$ and $B(t)$ control the annealing schedule, σ_i^x represent transverse field terms, and the problem Hamiltonian encodes the optimization objective through longitudinal fields h_i and coupling terms J_{ij}.

1.5.3 Gate-Model Quantum Computation

Gate-model quantum computation employs sequences of discrete quantum logic gates to perform computational operations. Universal gate sets enable implementation of various quantum algorithms, including:

- Grover's algorithm for unstructured search problems
- Quantum Fourier Transform for period finding and phase estimation
- Variational algorithms such as VQE and QAOA
- Quantum walk algorithms for graph-based problems

Quantum circuits are constructed from fundamental gates including Hadamard, Pauli, controlled-NOT, and rotation gates.

1.5.4 Quantum-Inspired Classical Algorithms

Quantum-inspired classical algorithms incorporate mathematical concepts from quantum mechanics while executing entirely on classical computational hardware. These approaches leverage quantum-inspired data structures and operations, such as:

- Tensor network representations for high-dimensional data
- Simulated quantum annealing through classical Monte Carlo methods
- Quantum walk simulations on classical graphs
- Hamiltonian simulation techniques using linear algebraic methods

Such algorithms aim to capture certain quantum advantages without requiring quantum hardware infrastructure.

2

Quantum Optimization for Geometric Constrained Image Segmentation

This chapter explores quantum optimization methods for geometric constrained image segmentation, combining graph-theoretic approaches with hybrid quantum-classical optimization. We present the QuantumLOGISMOS framework which leverages the Quantum Approximate Optimization Algorithm (QAOA) to solve surface segmentation problems.

2.1 Overview

Quantum image processing is an emerging field attracting attention from both quantum computing and image processing communities. We propose a novel method combining a graph-theoretic approach for optimal surface segmentation with hybrid quantum-classical optimization of the problem-directed graph. Surface segmentation is modeled classically as a graph partitioning problem with smoothness constraints to control surface variation for realistic segmentation. The problem-specific graph characteristics are embedded in a quadratic objective function whose minimum corresponds to the ground state energy of an equivalent Ising Hamiltonian.

This work explores the use of quantum processors in image segmentation problems, with important applications in medical image analysis. We present a theoretical basis for the quantum implementation of LOGISMOS and simulation results on simple images using the Quantum Approximate Optimization Algorithm (QAOA). The proposed approach can solve geometric-constrained surface segmentation problems optimally with the capability of locating multiple minimum points corresponding to globally minimal solutions.

Key Concept

Key Concepts: quantum computing, quantum algorithm, combinatorial optimization, image segmentation, graph theory, QAOA, LOGISMOS framework

2.2 Introduction

Image segmentation, particularly in medical imaging, involves partitioning an image into meaningful anatomical regions. The LOGISMOS framework (Layered Optimal Graph Image Segmentation of Multiple Objects and Surfaces) [**? ? ?**] reformulates surface segmentation as finding optimal boundaries subject to geometric constraints. Traditional algorithms solve this via maximum flow/minimum cut on directed graphs using classical optimization methods, but may miss alternative optimal solutions due to their deterministic nature.

Quantum computers offer potential advantages for combinatorial optimization problems through superposition (evaluating multiple solutions simultaneously), entanglement (correlated exploration of solution spaces), and quantum tunneling (escaping local minima more effectively). The Quantum Approximate Optimization Algorithm (QAOA) [**?**] provides a hybrid quantum-classical approach to approximate solutions of NP-hard problems by encoding them as ground state problems of Ising Hamiltonians.

This work introduces QuantumLOGISMOS, which: (1) maps LOGISMOS graph constraints to Quadratic Unconstrained Binary Optimization (QUBO) formulation, (2) implements quantum optimization via QAOA with classical parameter tuning, (3) demonstrates the method on synthetic 2D and 3D images, and (4) shows quantum advantage in finding multiple optimal segmentation solutions.

2.3 Classical LOGISMOS Framework

2.3.1 Mathematical Formulation

Given an image $\mathcal{I}$ with spatial dimensions (X, Y, Z), we represent it as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each pixel (2D) or voxel (3D) corresponds to a node $v \in \mathcal{V}$, organized into columns along a specific direction.

A surface $\mathcal{S}$ is defined by a function:

$$s : \text{Column} \rightarrow \text{Node}$$

where $s(x) = k$ indicates node k in column x belongs to the surface.

Each node (x, k) has an associated cost:

$$c_s(x, k) = -\log P(\text{node is on surface} \mid \text{image features})$$

Lower cost indicates higher likelihood of being on the desired surface.

The optimization objective is to find the surface minimizing total cost:

$$\hat{s} = \arg\min_s \sum_x c_s(x, s(x))$$

2.3.2 Graph Construction

Instead of directly minimizing costs, we transform to terminal weights:

$$w_s(x, k) = \begin{cases} -1 & \text{if } k = 1 \\ c_s(x, k) - c_s(x, k-1) & \text{otherwise} \end{cases}$$

For a closed set S (nodes below the surface):

$$W_s = \sum_x \sum_{k \in S_x} w_s(x, k) = \sum_x c_s(x, s(x)) + \text{constant}$$

Thus minimizing W_s is equivalent to minimizing the original cost function.

2.3.3 Edge Constraints

Three edge types enforce geometric constraints:

1. **Intra-column edges ($\mathcal{E}_{\text{intra}}$)**: $\forall x, \forall k > 1$: edge $(x, k) \to (x, k-1)$ with capacity ∞
   Ensures exactly one cut per column.
2. **Inter-column edges ($\mathcal{E}_{\text{inter}}$)**: Given smoothness parameter δ: $\forall$adjacent x, x' : edges $(x, k) \to (x', \max(1, k-\delta))$ with capacity ∞
   Enforces $|s(x) - s(x')| \leq \delta$.
3. **Terminal edges ($\mathcal{E}_W$)**: For nodes with $w_s(v) < 0$: edge $s \to v$ with capacity $|w_s(v)|$
   For nodes with $w_s(v) > 0$: edge $v \to t$ with capacity $|w_s(v)|$

2.3.4 Minimum Cut Reformulation

The optimal surface corresponds to the minimum s-t cut partitioning nodes into source set S (containing s) and sink set T (containing t). The cut capacity equals the total terminal weight of the corresponding closed set.

2.4 QuantumLOGISMOS Framework

The QuantumLOGISMOS framework, illustrated in Figure **??**, integrates classical graph-theoretic segmentation with quantum optimization through seven systematic steps:

1. **Cost Function Estimation and Terminal Weight Calculation**: Based on the input image, compute node costs reflecting surface likelihood and transform them to terminal weights using Equation **??**.
2. **Intra-column Edge Construction**: Create infinite-capacity directed edges within each column to ensure exactly one cut per column, enforcing the single-surface constraint.
3. **Inter-column Edge Addition**: Connect adjacent columns with infinite-capacity edges to impose smoothness constraints, restricting surface variation between neighboring columns.
4. **Source/Sink Node Addition**: Introduce source (s) and sink (t) nodes with capacity-weighted connections to all graph nodes based on terminal weight signs.
5. **Qubit Assignment**: Map each graph node to a qubit, where the qubit state ($|0\rangle$ or $|1\rangle$) represents node assignment to source or sink sets respectively.
6. **QAOA Optimization**: Execute the Quantum Approximate Optimization Algorithm to find the minimum cut by minimizing the problem Hamiltonian derived from the graph structure.
7. **Solution Extraction**: Decode the optimal bitstring obtained from QAOA measurements to identify the minimum closed set and corresponding optimal surface.

This hybrid framework leverages classical preprocessing (steps 1-4) to encode geometric constraints, while utilizing quantum optimization (steps 5-7) to explore the solution space more comprehensively than classical approaches.

2.5 Quantum Formulation

2.5.1 QUBO Conversion

For each node i, define binary variable $x_i \in \{0, 1\}$ where:

- $x_i = 0 \rightarrow$ node in source set S
- $x_i = 1 \rightarrow$ node in sink set T

For directed edge $i \rightarrow j$ with capacity w_{ij}:

$$F_{(i,j)}(x_i, x_j) = x_j - x_i x_j$$

This equals w_{ij} if the edge is cut ($x_i = 0, x_j = 1$), and o otherwise.

To enforce $x_s = 0, x_t = 1$:

$$F_{(s,t)}(x_s, x_t) = x_s x_t - x_s$$

Complete QUBO objective:

$$F_C(\mathbf{x}) = \sum_{(i,j)\in\mathcal{E}} w_{ij}(x_j - x_i x_j) + \varepsilon(x_s x_t - x_s)$$

where $\varepsilon = 1 + \sum_{(i,j)\in\mathcal{E}} w_{ij}$ ensures valid cuts have lower energy.

2.5.2 Matrix Formulation

$$F_C(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x}$$

where $\mathbf{Q}$ is symmetric with:

$$Q_{ii} = \sum_{j:i\rightarrow j} w_{ij}, \quad Q_{ij} = -\frac{w_{ij}}{2} \text{ for } i \neq j \text{ with edge } i \rightarrow j$$

2.5.3 Ising Hamiltonian Mapping

Binary variables map to qubit states via:

$$x_i = \frac{1 - Z_i}{2}$$

where Z_i is the Pauli-Z operator on qubit i.

Problem Hamiltonian:

$$H_C = \sum_{i,j} Q_{ij} \frac{1 - Z_i}{2} \frac{1 - Z_j}{2}$$

Expanding to standard Ising form:

$$H_C = \text{constant} + \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j$$

where:

$$h_i = -\frac{1}{4} \sum_j (Q_{ij} + Q_{ji}), \quad J_{ij} = \frac{1}{4} Q_{ij} \ (i \neq j)$$

Ground state energy E_0 of H_C satisfies:

$$E_0 = \min_{\mathbf{x}} F_C(\mathbf{x})$$

2.6 Quantum Optimization via QAOA

2.6.1 QAOA Circuit Structure

Initial state: uniform superposition

$$|\psi_0\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z\in\{0,1\}^n} |z\rangle$$

Problem unitary: $U_C(\gamma) = e^{-i\gamma H_C}$
Mixer unitary: $U_M(\beta) = e^{-i\beta \sum_i X_i}$
For p layers with parameters $\gamma = (\gamma_1, ..., \gamma_p)$, $\beta = (\beta_1, ..., \beta_p)$:

$$|\psi(\gamma, \beta)\rangle = \prod_{k=1}^{p} U_M(\beta_k) U_C(\gamma_k) |\psi_0\rangle$$

2.6.2 Hybrid Optimization

Energy estimation:

$$E(\gamma, \beta) = \langle\psi(\gamma, \beta)|H_C|\psi(\gamma, \beta)\rangle$$

Classical optimization uses Simultaneous Perturbation Stochastic Approximation (SPSA) [**?**]:

Algorithm 1 Hybrid Quantum-Classical Optimization

1: Initialize parameters γ, β
2: **while** not converged **do**
3: Prepare $|\psi(\gamma, \beta)\rangle$ on quantum processor/simulator
4: Measure to estimate $E(\gamma, \beta)$
5: Compute gradients via finite differences
6: Update γ, β using SPSA to minimize E
7: **end while**
8: Measure final state to obtain optimal bitstring $\mathbf{x}^*$
9: Decode $\mathbf{x}^*$ to surface nodes

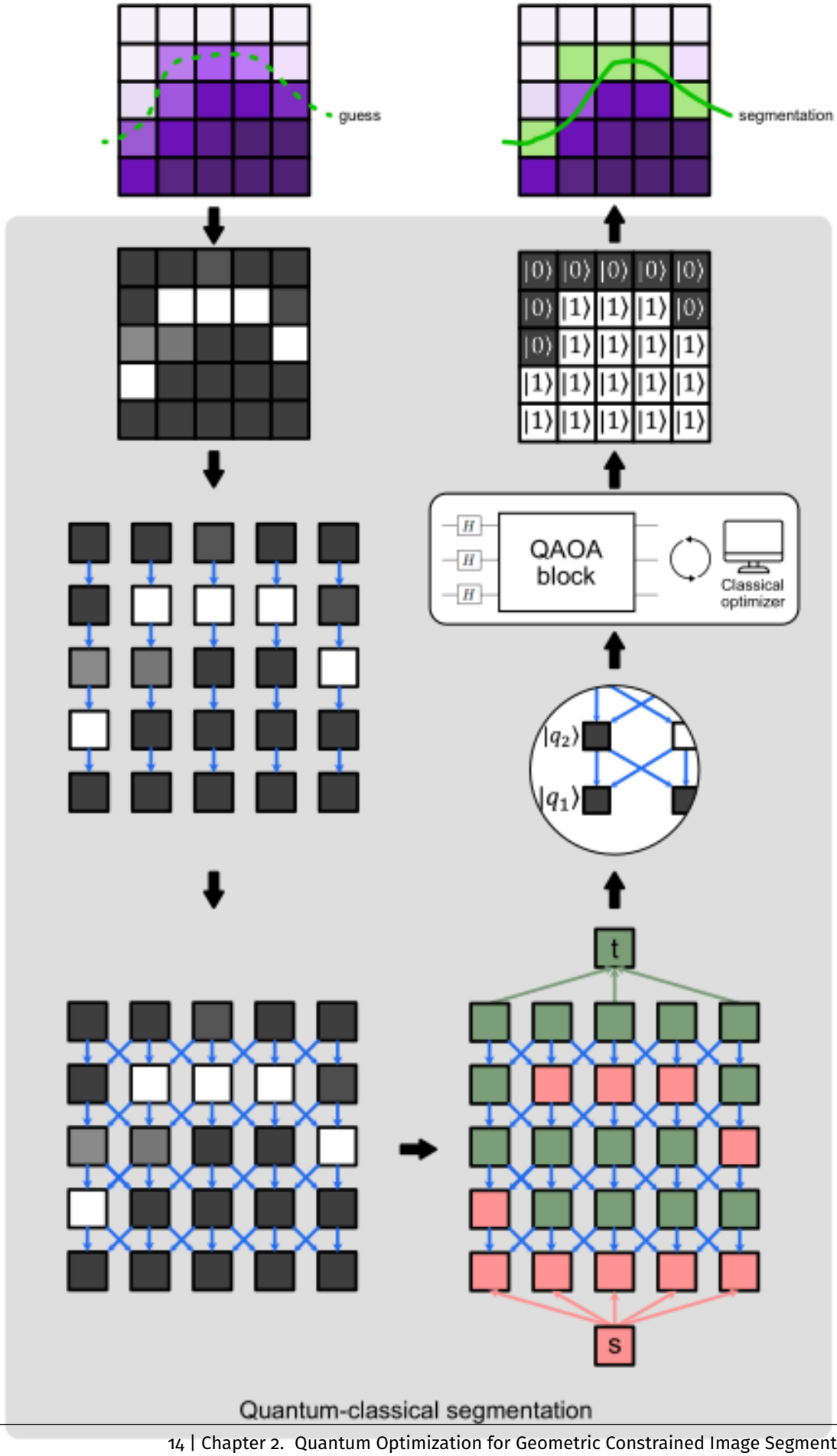

Figure 2.1: The proposed Quantum LOGISMOS framework: (1) Estimate cost functions and calculate terminal weights, (2) Introduce internal connections within columns to ensure that the optimal surface cut passes through each column only once, (3) Add inter-edges to impose

3 The Classical Paradigm: Image Segmentation via Clustering

3.1 Fundamentals and Core Definitions: Defining Clustering and the Segmentation Problem

Before delving into quantum approaches, it is essential to establish a rigorous understanding of the classical framework for image segmentation through clustering. This section defines the fundamental concepts and mathematical formulations that underpin both classical and quantum methods.

3.1.1 What is Clustering?

Clustering is an unsupervised machine learning technique that aims to partition a dataset into groups (clusters) such that objects within the same cluster are more similar to each other than to objects in different clusters.

Clustering

Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ where $x_i \in \mathbb{R}^d$, clustering seeks to find a partition $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ such that:

1. $\bigcup_{i=1}^{k} C_i = X$ (completeness)
2. $C_i \cap C_j = \emptyset$ for $i \neq j$ (mutual exclusivity in hard clustering)
3. $C_i \neq \emptyset$ for all i (non-emptiness)

The quality of clustering is typically measured by optimizing an objective function. For example, the within-cluster sum of squares (WCSS) for K-means clustering:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{3.1}$$

where μ_i is the centroid of cluster C_i.

3.1.2 The Image Segmentation Problem

Image segmentation is the process of partitioning a digital image into multiple segments, where each segment consists of pixels that share certain characteristics.

Image Segmentation

For an image I defined over a domain Ω, segmentation produces a partition $\{R_1, R_2, \ldots, R_n\}$ such that:

1. $\bigcup_{i=1}^{n} R_i = \Omega$
2. R_i is connected for all i
3. Pixels within each R_i satisfy a homogeneity predicate $P(R_i) = \text{TRUE}$
4. $P(R_i \cup R_j) = \text{FALSE}$ for adjacent regions R_i and R_j

In practice, pixels are represented as feature vectors that may include:

- **Color features**: RGB, HSV, or Lab color space values
- **Spatial features**: Pixel coordinates (x, y)
- **Texture features**: Local binary patterns, Gabor filter responses
- **Gradient features**: Edge magnitude and orientation

3.1.3 Clustering as a Segmentation Approach

When clustering is applied to image segmentation, each pixel (or region) is treated as a data point in a feature space. The clustering algorithm groups similar pixels together, effectively segmenting the image.

Note

The connection between clustering and segmentation is natural: both seek to identify groups of similar elements. However, image segmentation introduces additional constraints, such as spatial coherence and the preservation of object boundaries.

The general pipeline for clustering-based segmentation involves:

1. **Feature extraction**: Transform each pixel into a feature vector
2. **Clustering**: Apply a clustering algorithm to group feature vectors
3. **Label assignment**: Assign each pixel the label of its cluster
4. **Post-processing**: Refine segment boundaries and remove noise

3.2 Classical Clustering Architectures: From K-Means to Spectral Graph Theory

This section reviews the major classical clustering algorithms used for image segmentation, analyzing their mathematical foundations, strengths, and limitations.

3.2.1 K-Means Clustering

K-means is one of the most widely used clustering algorithms due to its simplicity and efficiency. It partitions data into k clusters by iteratively refining cluster centroids.

Algorithm 2 K-Means Clustering

Input: Dataset $X = \{x_1, \ldots, x_n\}$, number of clusters k **Output:** Cluster assignments and centroids Initialize k cluster centroids $\{\mu_1, \ldots, \mu_k\}$ randomly **Assignment step:** Assign each x_i to nearest centroid: $c_i = \arg\min_j \|x_i - \mu_j\|^2$ **Update step:** Recompute centroids: $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$ convergence (centroids no longer change)

Complexity Analysis: Each iteration requires $O(nkd)$ operations, where n is the number of data points, k is the number of clusters, and d is the dimensionality. For images, n can be millions of pixels.

> **Warning**
>
> K-means has several limitations for image segmentation:
>
> - Assumes spherical, equally-sized clusters
> - Sensitive to initialization and outliers
> - Requires specifying k in advance
> - Cannot capture complex cluster shapes

3.2.2 Hierarchical Clustering

Hierarchical clustering builds a tree-like structure (dendrogram) of nested clusters without requiring the number of clusters to be specified beforehand.

Agglomerative (bottom-up) approach:

1. Start with each point as its own cluster
2. Iteratively merge the two closest clusters
3. Continue until all points belong to a single cluster

The distance between clusters can be computed using various linkage criteria:

- **Single linkage:** $d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$
- **Complete linkage:** $d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y)$
- **Average linkage:** $d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y)$
- **Ward's method:** Minimizes within-cluster variance

Complexity: Standard hierarchical clustering has $O(n^3)$ time complexity and $O(n^2)$ space complexity, making it impractical for large images.

3.2.3 Fuzzy C-Means

Unlike hard clustering methods, Fuzzy C-Means (FCM) allows each data point to belong to multiple clusters with varying degrees of membership.

Fuzzy C-Means Objective

FCM minimizes the following objective function:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \|x_i - v_j\|^2 \tag{3.2}$$

where u_{ij} is the membership degree of x_i in cluster j, v_j is the cluster center, and $m > 1$ is the fuzziness parameter.

The membership values and cluster centers are updated iteratively:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}} \tag{3.3}$$

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{3.4}$$

Tip

FCM is particularly useful for image segmentation when boundaries between regions are gradual or ambiguous, such as in medical imaging where tissue boundaries may be unclear.

3.2.4 Spectral Clustering and Graph Theory

Spectral clustering uses eigenvalues of similarity matrices to perform dimensionality reduction before clustering. It can identify clusters with complex, non-convex shapes.

Unlike traditional clustering methods that rely on direct distance measures, spectral clustering captures the global structure of the image by analyzing how pixels are connected in a graph representation. Each pixel is treated as a node, and edges reflect similarity in color, intensity, or texture. By performing clustering in the low-dimensional space defined by the leading eigenvectors, spectral clustering can effectively separate complex or non-convex regions, making it particularly useful for segmenting images with irregular shapes or subtle boundaries.

Graph-based formulation:

1. Construct an affinity matrix W where W_{ij} represents similarity between points i and j:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \tag{3.5}$$

2. Compute the graph Laplacian. The normalized Laplacian is:

$$L_{norm} = I - D^{-1/2} W D^{-1/2} \tag{3.6}$$

   where D is the diagonal degree matrix with $D_{ii} = \sum_j W_{ij}$

3. Compute the k smallest eigenvectors of L_{norm}
4. Form matrix $U \in \mathbb{R}^{n \times k}$ from these eigenvectors
5. Apply K-means to the rows of U

Tip

Spectral clustering is particularly effective for image segmentation because it can capture non-convex cluster shapes and naturally incorporates spatial relationships through the affinity matrix.

Normalized Cuts: A popular spectral method for image segmentation that minimizes:

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \tag{3.7}$$

where $\text{cut}(A, B)$ is the total weight of edges between segments A and B, and $\text{assoc}(A, V)$ is the total weight of edges from A to all nodes.

3.2.5 Density-Based Clustering (DBSCAN)

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) groups pixels into clusters based on local density, marking sparse regions as noise.

DBSCAN Core Concepts

DBSCAN defines clusters using two parameters:

- ε (eps): The neighborhood radius
- MinPts: Minimum number of points required to form a dense region

A point p is a **core point** if at least MinPts points are within distance ε of p.

Algorithm steps:

1. Find core points based on ε and MinPts
2. Connect core points that are within ε of each other
3. Assign non-core points to nearby clusters or mark as noise

Advantages for image segmentation:

- Can find arbitrarily shaped clusters (unlike K-means)
- Does not require specifying number of clusters k in advance
- Naturally handles outliers and noise
- Particularly useful for images with irregular region shapes

Warning

DBSCAN's results depend heavily on setting ε (neighborhood radius) and MinPts correctly. Poor parameter choices can merge distinct regions or split coherent ones.

3.2.6 Other Classical Methods

Several other clustering methods are used for image segmentation:

- **Mean-Shift Clustering:** A non-parametric technique that iteratively shifts points toward modes (local maxima) of the density function. Does not require specifying k.
- **Gaussian Mixture Models (GMM):** A probabilistic approach that models data as a mixture of Gaussian distributions, providing soft cluster assignments similar to FCM.
- **Graph Cuts:** Energy minimization methods that formulate segmentation as finding optimal cuts in a graph, balancing data fidelity and smoothness terms.
- **Watershed Algorithm:** A morphological approach treating the image as a topographic surface and finding basin boundaries.

Motivation for Quantum Approaches

Classical clustering methods typically minimize simple distance-based cost functions (like within-cluster variance) and lack mechanisms to easily exploit very high-dimensional structure. This motivates exploring quantum techniques, which can embed data in quantum states and optimize quantum-native objectives that may better capture complex relationships in image data.

3.3 Computational Bottlenecks: The Case for Quantum Advantage

Despite their effectiveness, classical clustering algorithms face significant computational challenges when applied to modern image segmentation tasks. This section analyzes these bottlenecks and motivates the exploration of quantum computing solutions.

3.3.1 Scalability Issues in Classical Methods

Modern imaging systems produce increasingly large datasets. A single 4K image contains over 8 million pixels, each potentially represented by multiple features. The computational complexity of clustering algorithms presents serious scalability challenges:

Table 3.1: Computational Complexity of Classical Clustering Algorithms

Algorithm	Time Complexity	Space Complexity
K-Means	$O(nkdI)$	$O(nd + kd)$
Hierarchical	$O(n^3)$	$O(n^2)$
Fuzzy C-Means	$O(nc^2dI)$	$O(nd + cd)$
Spectral	$O(n^3)$	$O(n^2)$

Here, n is the number of data points, k or c is the number of clusters, d is dimensionality, and I is the number of iterations.

Warning

For spectral clustering on a 1-megapixel image, computing the affinity matrix alone requires 10^{12} operations and 10^{12} bytes of storage—clearly impractical for real-time applications.

3.3.2 High-Dimensional Data Challenges

Image segmentation often requires high-dimensional feature spaces to capture color, texture, and spatial information. This leads to the *curse of dimensionality*:

- **Distance concentration:** In high dimensions, the ratio of nearest to farthest neighbor distances approaches 1, making distance-based methods less discriminative
- **Sparse data:** Data becomes increasingly sparse as dimensions increase, requiring exponentially more samples
- **Computational overhead:** Distance calculations scale linearly with dimensionality

Curse of Dimensionality

For uniformly distributed data in a d-dimensional hypercube, the expected distance to the nearest neighbor grows as:

$$E[d_{NN}] \propto \left(\frac{1}{n}\right)^{1/d} \tag{3.8}$$

As $d \to \infty$, all points become approximately equidistant.

3.3.3 Why Quantum Computing?

Quantum computing offers several potential advantages for clustering and image segmentation:

Quantum Advantages

1. **Exponential state space:** n qubits can represent 2^n states simultaneously through superposition
2. **Quantum parallelism:** Operations can be performed on all superposition states at once
3. **Amplitude encoding:** N classical data points can be encoded in $\log_2 N$ qubits
4. **Quantum speedups:** Grover's search provides quadratic speedup; HHL algorithm offers exponential speedup for linear systems

Theoretical speedups for clustering-related tasks:

- **Distance calculation:** Quantum algorithms can compute distances between vectors in $O(\log d)$ time using amplitude encoding, compared to $O(d)$ classically

- **Eigenvalue problems:** Quantum phase estimation can find eigenvalues in $O(\text{poly}(\log n))$ time, potentially improving spectral clustering
- **Optimization:** Quantum approximate optimization algorithms (QAOA) and variational quantum eigensolvers (VQE) offer new approaches to NP-hard clustering problems

Note

While quantum advantages are theoretically promising, practical implementation faces challenges including limited qubit counts, noise in current hardware (NISQ era), and the overhead of quantum data encoding.

3.3.4 Quantum Advantage and Complexity

Quantum advantage (or quantum supremacy) refers to the demonstration that a quantum computer can solve a problem that is practically impossible for classical computers within a reasonable timeframe. Understanding the theoretical foundations of quantum advantage requires examining computational complexity classes.

Complexity Classes

Key complexity classes relevant to quantum computing:

- **P** (Polynomial time): Problems solvable efficiently on classical computers in polynomial time
- **NP** (Nondeterministic Polynomial time): Problems whose solutions can be verified efficiently in polynomial time
- **BQP** (Bounded-error Quantum Polynomial time): Problems solvable efficiently on quantum computers with bounded error probability

The relationship between these classes is believed to be:

$$\text{P} \subset \text{BQP} \subseteq \text{NP} \tag{3.9}$$

This hierarchy suggests that quantum computers can solve some problems more efficiently than classical computers, while still respecting fundamental computational limits.

Quantum Relevance to Clustering

Clustering is generally **NP-hard** for arbitrary distance metrics. The optimal partitioning of data points into clusters involves exploring an exponentially large solution space. Quantum algorithms may offer:

- **Polynomial speedups** for distance calculations via amplitude encoding
- **Quadratic speedups** for search-based operations via Grover's algorithm
- **Potential exponential speedups** for specific formulations using quantum linear algebra

However, proving definitive quantum advantage for clustering remains an active research area, as the data loading problem and NISQ limitations complicate theoretical claims.

4 Quantum Clustering: A New Frontier in Image Segmentation

4.1 Introduction: Bridging Quantum Mechanics and Computer Vision

The intersection of quantum computing and computer vision represents an emerging field with significant potential for advancing image segmentation capabilities. This chapter explores how quantum mechanical principles can be leveraged to develop more efficient clustering algorithms for image analysis.

Quantum computing fundamentally differs from classical computing in its ability to exploit quantum mechanical phenomena—superposition, entanglement, and interference—to process information. These properties enable quantum algorithms to explore solution spaces in ways that are impossible for classical computers.

Quantum Computing for Clustering

The key insight driving quantum clustering research is that many clustering problems can be reformulated as:

- Distance estimation problems (quantum advantage via amplitude encoding)
- Eigenvalue problems (quantum advantage via quantum phase estimation)
- Optimization problems (quantum advantage via QAOA or quantum annealing)

4.1.1 The NISQ Era and Its Implications

Current quantum computers operate in the Noisy Intermediate-Scale Quantum (NISQ) era, characterized by:

- **Limited qubit counts:** Typically 50-1000 qubits
- **High error rates:** Gate errors of 10^{-3} to 10^{-2}
- **Limited coherence times:** Microseconds to milliseconds
- **Restricted connectivity:** Not all qubits can interact directly

Warning

These limitations mean that theoretically optimal quantum algorithms may not be practical on current hardware. Much research focuses on variational and hybrid classical-quantum approaches that are more noise-tolerant.

4.1.2 From Classical to Quantum: Key Transformations

Applying quantum computing to image segmentation requires several key transformations:

1. **Data encoding:** Converting classical image data into quantum states
2. **Algorithm design:** Developing quantum circuits that perform clustering operations
3. **Measurement and interpretation:** Extracting classical cluster assignments from quantum measurements

4.2 Approaches to Quantum Image Representation and Processing

A fundamental challenge in quantum image processing is encoding classical image data into quantum states. Several representations have been proposed, each with distinct advantages for different applications.

4.2.1 Flexible Representation of Quantum Images (FRQI)

FRQI encodes a $2^n \times 2^n$ grayscale image using $2n + 1$ qubits:

FRQI Representation

$$|\text{FRQI}\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} (\cos\theta_i |0\rangle + \sin\theta_i |1\rangle) \otimes |i\rangle \tag{4.1}$$

where $\theta_i \in [0, \pi/2]$ encodes the grayscale value of pixel i, and $|i\rangle$ encodes the pixel position.

Advantages:

- Compact representation: $2n + 1$ qubits for 2^{2n} pixels
- Natural encoding for grayscale images

Limitations:

- Complex state preparation requiring $O(2^{2n})$ gates
- Limited grayscale precision (single qubit for intensity)

4.2.2 Novel Enhanced Quantum Representation (NEQR)

NEQR improves upon FRQI by using a basis encoding for pixel intensities:

NEQR Representation

$$|\text{NEQR}\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} |f(y,x)\rangle \, |yx\rangle \tag{4.2}$$

where $|f(y,x)\rangle = \left|c_0^{yx} c_1^{yx} \dots c_{q-1}^{yx}\right\rangle$ is the q-bit binary representation of the grayscale value at position (y,x).

Advantages:

- Higher precision: q bits for intensity values (e.g., $q = 8$ for 256 levels)
- Simpler image operations through bit-level manipulations
- More efficient for certain quantum image processing operations

4.2.3 Amplitude Encoding for Feature Vectors

For clustering applications, amplitude encoding is particularly important as it allows efficient representation of high-dimensional feature vectors:

Amplitude Encoding

A normalized vector $\mathbf{x} = (x_0, x_1, \dots, x_{N-1})^T$ with $\|\mathbf{x}\| = 1$ can be encoded as:

$$|\mathbf{x}\rangle = \sum_{i=0}^{N-1} x_i \, |i\rangle \tag{4.3}$$

requiring only $\lceil \log_2 N \rceil$ qubits to represent N amplitudes.

Tip

Amplitude encoding enables exponential compression: a feature vector with $N = 2^n$ components requires only n qubits. This is crucial for encoding high-dimensional image features efficiently.

4.2.4 Quantum Distance Estimation

A key operation for clustering is computing distances between data points. Quantum computers can estimate distances using the swap test:

$$|\langle \mathbf{x}|\mathbf{y}|\mathbf{x}|\mathbf{y}\rangle|^2 = 1 - \frac{d(\mathbf{x},\mathbf{y})^2}{2} \tag{4.4}$$

where $d(\mathbf{x},\mathbf{y})$ is the Euclidean distance between normalized vectors.

The swap test circuit measures the overlap between two quantum states with $O(1)$ quantum operations (after state preparation), compared to $O(d)$ classical operations for d-dimensional vectors.

4.3 A Comprehensive Review of Quantum Segmentation Algorithms

This section provides a detailed analysis of quantum clustering algorithms that have been proposed for image segmentation, comparing their approaches, theoretical advantages, and practical limitations.

4.3.1 Quantum K-Means and Variants

Quantum versions of K-means leverage quantum speedups in distance calculations and centroid updates.

q-Means Algorithm

The q-means algorithm, proposed by Kerenidis et al. (2019), achieves exponential speedup over classical K-means under certain conditions:

Algorithm 3 q-Means Algorithm (Simplified)

Input: Quantum access to data matrix $V \in \mathbb{R}^{n \times d}$, number of clusters k Initialize centroids using quantum sampling Use quantum distance estimation to find nearest centroid for each point Update centroids using quantum linear algebra convergence
Output: Cluster assignments

Complexity: $O\left(k^2 d \frac{\eta^{2.5}}{\delta^2} \text{polylog}(nd)\right)$ per iteration, where η is a condition number and δ is the desired precision.

> **Warning**
>
> Current quantum hardware (NISQ devices) has limited qubits and high error rates, which constrains practical implementations. The q-means algorithm requires fault-tolerant quantum computers with quantum RAM (qRAM), which are not yet available.

Variational Quantum K-Means

For NISQ devices, variational approaches are more practical:

1. Encode data points using parameterized quantum circuits
2. Use a variational classifier to assign cluster labels
3. Optimize circuit parameters using classical optimization

4.3.2 Quantum Spectral Clustering

Quantum spectral clustering leverages quantum algorithms for eigenvalue problems, potentially offering exponential speedup for the most computationally intensive step.

Quantum Principal Component Analysis (qPCA)

qPCA, based on the HHL algorithm, can be used to find the principal eigenvectors of the Laplacian matrix:

1. Prepare the density matrix $\rho = \frac{L}{\text{tr}(L)}$
2. Apply quantum phase estimation to extract eigenvalues
3. Sample from the eigenvector subspace

Theoretical complexity: $O(\text{polylog}(n))$ compared to $O(n^3)$ classically.

Quantum Normalized Cuts

The normalized cut problem can be formulated as a quadratic unconstrained binary optimization (QUBO) problem, suitable for quantum annealers:

$$\min_{\mathbf{x}\in\{0,1\}^n} \mathbf{x}^T Q \mathbf{x} \tag{4.5}$$

where Q encodes the normalized cut objective.

> **Note**
>
> D-Wave quantum annealers have been used to solve small-scale normalized cut problems, demonstrating proof-of-concept for quantum image segmentation.

4.3.3 Variational Quantum Clustering

Variational approaches use parameterized quantum circuits (PQCs) optimized through hybrid classical-quantum loops.

Quantum Approximate Optimization Algorithm (QAOA)

QAOA can be applied to clustering formulated as combinatorial optimization:

$$|\psi(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle = \prod_{p=1}^{P} e^{-i\beta_p H_M} e^{-i\gamma_p H_C} |s\rangle \tag{4.6}$$

where H_C encodes the clustering objective and H_M is a mixing Hamiltonian.

Variational Quantum Eigensolver (VQE) for Clustering

VQE can find the ground state of a Hamiltonian encoding the clustering problem:

1. Define Hamiltonian H such that its ground state encodes optimal clustering
2. Prepare parameterized ansatz $|\psi(\boldsymbol{\theta})\rangle$
3. Measure $\langle H|H\rangle = \langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle$
4. Classically optimize $\boldsymbol{\theta}$ to minimize $\langle H|H\rangle$

> **Tip**
>
> Variational methods are the most promising for near-term quantum devices because they can tolerate noise and work with limited qubit counts. They have been demonstrated on actual quantum hardware for small-scale clustering problems.

4.3.4 Quantum Fuzzy Clustering

Quantum extensions of fuzzy C-means combine the benefits of soft clustering with quantum speedups.

Quantum Fuzzy C-Means (QFCM)

QFCM uses quantum parallelism to compute membership degrees and update cluster centers:

1. Encode membership matrix in quantum state
2. Use quantum arithmetic to compute weighted distances
3. Apply Grover's search to find optimal membership assignments

Potential speedup: Quadratic improvement in the number of data points due to Grover's algorithm.

Quantum-Inspired Fuzzy Clustering

Even without full quantum hardware, quantum-inspired algorithms on classical computers can improve fuzzy clustering:

- Tensor network representations for efficient computation
- Quantum sampling techniques for initialization
- Amplitude estimation-inspired distance calculations

4.3.5 Comparative Analysis

Table **??** summarizes the key characteristics of quantum clustering algorithms for image segmentation.

Table 4.1: Comparison of Quantum Clustering Algorithms

Algorithm	Speedup	Hardware	Maturity
q-Means	Exponential	Fault-tolerant + qRAM	Theoretical
Quantum Spectral	Exponential	Fault-tolerant	Theoretical
QAOA Clustering	Potential	NISQ	Experimental
VQE Clustering	Heuristic	NISQ	Experimental
Quantum Fuzzy	Quadratic	Fault-tolerant	Theoretical

Current State of the Art

While theoretical quantum algorithms promise significant speedups, practical quantum image segmentation remains limited by:

- Data loading bottleneck: Encoding classical images into quantum states
- Hardware limitations: Qubit counts, error rates, and coherence times
- Scalability: Current demonstrations limited to small images

Variational methods on NISQ devices represent the most promising near-term approach.

Conclusion

This report has provided a comprehensive review of quantum clustering approaches for image segmentation, examining both the theoretical foundations and practical considerations of this emerging field.

Summary of Key Findings

Classical Limitations: Traditional clustering algorithms face significant computational bottlenecks when applied to modern image segmentation tasks. K-means, while efficient, assumes spherical clusters and scales linearly with data size. Spectral clustering offers superior segmentation quality but suffers from $O(n^3)$ complexity. These limitations become critical as image resolutions continue to increase.

Quantum Promise: Quantum computing offers theoretical speedups through:

- Exponential state space compression via amplitude encoding
- Efficient distance calculations using quantum interference
- Speedups for eigenvalue problems central to spectral methods
- New optimization paradigms through QAOA and quantum annealing

Current Reality: Despite theoretical advantages, practical quantum image segmentation remains in its infancy:

- The data loading problem presents a significant bottleneck
- NISQ devices limit algorithm complexity and problem size
- Most demonstrations remain at proof-of-concept scale
- Hybrid classical-quantum approaches show the most near-term promise

Future Directions and Open Challenges

Key areas for future research include:

1. **Efficient quantum data encoding:** Developing practical methods for loading classical image data into quantum states without negating computational advantages
2. **Noise-resilient algorithms:** Designing quantum clustering algorithms that maintain accuracy despite hardware noise, possibly through error mitigation techniques
3. **Hybrid architectures:** Optimizing the division of labor between classical and quantum processors for maximum practical benefit

4. **Application-specific designs:** Tailoring quantum algorithms to specific image segmentation tasks (medical imaging, satellite imagery, etc.)
5. **Benchmarking frameworks:** Establishing standardized benchmarks to fairly compare quantum and classical approaches
6. **Hardware advances:** As quantum computers improve, revisiting theoretical algorithms that may become practical

Looking Ahead

Quantum clustering for image segmentation represents a promising but challenging frontier. While the path to practical quantum advantage remains uncertain, continued research in algorithm design, error mitigation, and hardware development may eventually unlock the potential of quantum computing for computer vision applications.

Bibliography

[] Kang Li, Xiaodong Wu, DZ Chen, and Milan Sonka. Globally optimal segmentation of interacting surfaces with geometric constraints. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–I, 2004. doi: 10.1109/CVPR.2004.1315059.

[] Kang Li, Xiaodong Wu, DZ Chen, and Milan Sonka. Optimal surface segmentation in volumetric images—a graph-theoretic approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):119–134, 2006. doi: 10.1109/TPAMI.2006.19.

[] Xiaodong Wu and Danny Z Chen. Optimal net surface problems with applications. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, pages 1029–1042. Springer, 2002.

[] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[] James C Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998.

[] Andrew V Goldberg and Robert E Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940, 1988. doi: 10.1145/48014.61051.

Primer on Quantum Information Processing

This appendix provides essential background on quantum computing concepts for readers less familiar with the field. Understanding these fundamentals is crucial for appreciating the quantum clustering algorithms discussed in this report.

.1 From Classical to Quantum

Classical computers process information using bits that exist in one of two states: 0 or 1. Quantum computers use quantum bits, or *qubits*, which can exist in superpositions of both states simultaneously. This fundamental difference enables quantum computers to process information in ways that are impossible for classical systems.

Classical vs. Quantum

The key distinctions between classical and quantum computing:

- **Classical bits:** Deterministic states (0 or 1)
- **Qubits:** Probabilistic superposition of states
- **Classical operations:** Boolean logic gates
- **Quantum operations:** Unitary transformations preserving probability

.2 Quantum Bits (Qubits)

The fundamental unit of quantum information is the quantum bit, or qubit. Unlike classical bits that exist definitively in state 0 or 1, qubits can exist in quantum superposition states.

Qubit State

A qubit state $|\psi\rangle$ can be written as:

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{7}$$

where $\alpha, \beta \in \mathbb{C}$ are complex amplitudes satisfying $|\alpha|^2 + |\beta|^2 = 1$.

The state can be visualized on the Bloch sphere using the parametrization:

$$|\psi\rangle = \cos\frac{\theta}{2}\,|0\rangle + e^{i\phi}\sin\frac{\theta}{2}\,|1\rangle \tag{8}$$

where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$.

Multiple qubits: A system of n qubits exists in a 2^n-dimensional Hilbert space:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \tag{9}$$

This exponential growth in state space is the source of quantum computational power.

.3 Superposition

Superposition is the principle that a quantum system can exist in multiple states simultaneously until measured. For n qubits, a quantum system can represent 2^n states simultaneously.

Quantum Superposition

A quantum state $|\psi\rangle$ is in superposition when it cannot be expressed as a single basis state, but rather as a linear combination:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \quad \text{where } \sum_i |\alpha_i|^2 = 1 \tag{10}$$

This exponential scaling is the foundation of **quantum parallelism**. A quantum algorithm can process all 2^n states in parallel, potentially achieving exponential speedups for certain problems.

Relevance to Clustering

In clustering, superposition could allow simultaneous evaluation of a pixel's distance to all cluster centroids, potentially reducing the number of operations required. Instead of computing k distances sequentially, a quantum state can encode all distance comparisons in superposition.

.4 Quantum Gates

Quantum computation is performed through quantum gates, which are unitary transformations on qubit states. A transformation U is unitary if $U^\dagger U = UU^\dagger = I$.

Common Single-Qubit Gates

- **Pauli-X (NOT gate):** $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ — Bit flip
- **Pauli-Y:** $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ — Bit and phase flip
- **Pauli-Z:** $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ — Phase flip
- **Hadamard (H):** $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ — Creates superposition

- **Phase gates (S, T):** Apply phase shifts of $\pi/2$ and $\pi/4$ respectively
- **Rotation gates (R_x, R_y, R_z):** Parameterized rotations around Bloch sphere axes

Two-qubit gates: Multi-qubit gates enable entanglement and conditional operations:

- **CNOT (Controlled-NOT):** Fundamental entangling gate—flips target qubit if control is $|1\rangle$
- **SWAP:** Exchanges the states of two qubits
- **Controlled-phase gates:** Apply conditional phase shifts based on control qubit state

The CNOT gate matrix representation:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{11}$$

.5 Quantum Entanglement

Entanglement is a uniquely quantum phenomenon where the states of multiple qubits become correlated in ways that cannot be described classically.

Bell States

The four maximally entangled two-qubit states are:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{12}$$

$$\left|\Phi^-\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{13}$$

$$\left|\Psi^+\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{14}$$

$$\left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{15}$$

Entanglement is a resource for quantum algorithms, enabling correlations that speed up computation. Measuring one qubit immediately determines the state of the other, regardless of the distance between them—a property Einstein famously called "spooky action at a distance."

Relevance to Clustering

Entanglement could capture complex correlations between pixels, potentially enabling more sophisticated similarity measures and better representation of spatial relationships in images. Entangled qubits can encode non-local dependencies that are difficult to represent classically.

.6 Quantum Measurement

Measurement is the process of extracting classical information from a quantum state. Upon measurement, a quantum state *collapses* to one of the basis states with probability determined by the amplitudes.

Measurement Postulate

For a state $|\psi\rangle = \sum_i \alpha_i |i\rangle$, measurement in the computational basis yields outcome i with probability:

$$P(i) = |\alpha_i|^2 \tag{16}$$

After measurement, the state collapses to $|i\rangle$.

Measurement Problem

Measurement destroys superposition, collapsing the state to a single outcome. This means quantum algorithms must be carefully designed to extract useful information before decoherence occurs. The art of quantum algorithm design lies in arranging interference patterns so that correct answers have high measurement probability.

.7 Quantum Algorithms

Several quantum algorithms provide speedups relevant to clustering:

.7.1 Grover's Search Algorithm

Grover's algorithm searches an unstructured database of N items in $O(\sqrt{N})$ queries, compared to $O(N)$ classically.

Application to clustering: Finding the nearest centroid among k centroids can be accelerated from $O(k)$ to $O(\sqrt{k})$.

.7.2 Quantum Phase Estimation

Given a unitary U and its eigenstate $|u\rangle$ with $U|u\rangle = e^{2\pi i\phi}|u\rangle$, phase estimation determines ϕ to n bits of precision using $O(2^n)$ applications of controlled-U.

Application to clustering: Finding eigenvalues of the Laplacian matrix for spectral clustering.

.7.3 HHL Algorithm for Linear Systems

The Harrow-Hassidim-Lloyd (HHL) algorithm solves $A\mathbf{x} = \mathbf{b}$ in time $O(\text{poly}(\log N, \kappa))$ where κ is the condition number of A.

Application to clustering: Computing matrix inversions and solving linear systems that arise in various clustering formulations.

.7.4 Shor's Algorithm

Shor's algorithm factors large integers exponentially faster than known classical algorithms, reducing the complexity from sub-exponential to polynomial time.

Significance: Demonstrates that quantum computers can solve problems believed to be intractable classically, with major implications for cryptography.

.7.5 Quantum Fourier Transform (QFT)

The Quantum Fourier Transform is exponentially faster than the classical Fast Fourier Transform (FFT):

$$\text{QFT}\,|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N}\,|k\rangle \tag{17}$$

Application to clustering: QFT is a subroutine in many quantum algorithms including phase estimation, which underlies quantum spectral clustering methods.

.7.6 Variational Quantum Eigensolver (VQE)

VQE is a hybrid quantum-classical algorithm designed for NISQ devices. It finds ground states of quantum systems by:

1. Preparing a parameterized quantum state (ansatz)
2. Measuring the expected energy
3. Classically optimizing the parameters to minimize energy

Application to clustering: VQE can optimize clustering objectives encoded as Hamiltonians, making it suitable for near-term quantum hardware.

> **Tip**
>
> While these algorithms offer theoretical speedups, they often require fault-tolerant quantum computers. NISQ-era implementations typically use variational approximations (like VQE and QAOA) that trade optimality for noise resilience.

Fundamentals of Fuzzy Logic in Segmentation

This appendix covers the principles of fuzzy logic and fuzzy clustering, which form the basis for quantum fuzzy clustering algorithms discussed in the main text.

.8 Introduction to Fuzzy Sets

Classical set theory operates on crisp membership: an element either belongs to a set or it does not. Fuzzy set theory, introduced by Lotfi Zadeh in 1965, extends this to allow partial membership.

Fuzzy Set

A fuzzy set A in a universe of discourse X is characterized by a membership function:

$$\mu_A : X \to [0, 1] \tag{18}$$

where $\mu_A(x)$ represents the degree to which x belongs to A.

Key operations on fuzzy sets:

- **Union:** $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$
- **Intersection:** $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$
- **Complement:** $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

.9 Fuzzy C-Means Clustering

Fuzzy C-Means (FCM), developed by Dunn and later improved by Bezdek, is the most widely used fuzzy clustering algorithm.

FCM Objective Function

FCM minimizes:

$$J_m(U, V) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \|x_i - v_j\|^2 \tag{19}$$

subject to:

$$\sum_{j=1}^{c} u_{ij} = 1 \quad \forall i, \quad u_{ij} \geq 0 \tag{20}$$

where:

- u_{ij} is the membership degree of point x_i in cluster j

- v_j is the center of cluster j
- $m > 1$ is the fuzziness parameter (typically $m = 2$)

Update equations:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}} \tag{21}$$

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{22}$$

where $d_{ij} = \|x_i - v_j\|$.

Note

The parameter m controls the degree of fuzziness:

- As $m \to 1$: FCM approaches hard K-means
- As $m \to \infty$: All memberships approach $1/c$ (maximum fuzziness)
- Typical choice: $m = 2$

.10 Fuzzy Logic in Image Segmentation

Fuzzy clustering is particularly well-suited for image segmentation because:

1. **Gradual boundaries:** Real images often have smooth transitions between regions
2. **Uncertainty handling:** Noise and ambiguous pixels can be represented with partial membership
3. **Robustness:** Less sensitive to outliers than hard clustering

Applications in medical imaging:

- Brain MRI segmentation (gray matter, white matter, CSF)
- Tumor boundary detection
- Organ segmentation in CT scans

Extensions for image segmentation:

- **Spatial FCM:** Incorporates neighborhood information
- **Kernel FCM:** Uses kernel functions for non-linear boundaries
- **Type-2 Fuzzy FCM:** Handles uncertainty in membership functions

.11 Quantum Extensions of Fuzzy Clustering

Quantum computing can enhance fuzzy clustering in several ways:

.11.1 Quantum Speedup for Distance Calculations

The membership update requires computing distances from each point to all centroids. Quantum amplitude encoding and the swap test can accelerate this:

$$d_{ij}^2 = \|x_i\|^2 + \|v_j\|^2 - 2\langle x_i|v_j|x_i|v_j\rangle \tag{23}$$

The inner product $\langle x_i|v_j|x_i|v_j\rangle$ can be estimated quantumly in $O(\log d)$ time.

.11.2 Quantum Optimization for Cluster Centers

The centroid update can be formulated as a weighted least squares problem:

$$v_j = \arg\min_v \sum_{i=1}^{n} u_{ij}^m \|x_i - v\|^2 \tag{24}$$

Quantum algorithms for linear systems (HHL) can potentially solve this more efficiently.

.11.3 Grover Search for Membership Assignment

Finding the cluster with maximum membership for defuzzification can use Grover's search for quadratic speedup.

Tip

Quantum fuzzy clustering combines the interpretability of fuzzy methods with quantum computational advantages. This is particularly promising for medical image segmentation where both accuracy and interpretability are crucial.