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 image segmentation, examine how quantum algorithms can enhance traditional clustering methods, and discuss their application to the challenging task of image segmentation. The review covers only theoretical foundations and not the practical implementations.

Keywords: Quantum Computing, Clustering, Image Segmentation, Quantum Image Processing, Image Processing.

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
- Present comprehensively the different approaches for quantum clustering applied to image segmentation
- Provide a theoretical comparative analysis of quantum clustering algorithms, examining their formulations, strengths, and limitations

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 state-of-the-art quantum segmentation algorithms including Q-Seg, QuantumLOGISMOS, FFQOAK, and QCFFCM.

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 mathematical foundations required to understand quantum image segmentation, including QUBO formulations, minimum graph cuts, Markov Random Fields, convex optimization, and fuzzy clustering.

1 The Classical Paradigm: Image Segmentation via Clustering

1.1 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.

1.1.1 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:

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{1.1}$$

where μ_i is the centroid of cluster C_i.

1.1.2 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

1.1.3 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

1.2 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.

1.2.1 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 1 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
repeat
    Assignment step: Assign each x_i to nearest centroid:
        $c_i = \arg\min_j \|x_i - \mu_j\|^2$
    Update step: Recompute centroids:
        $\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$
until 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 (convex), equally-sized clusters
> - Sensitive to initialization and outliers
> - Requires specifying k in advance
> - Cannot capture complex cluster shapes

1.2.2 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.

1.2.3 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:

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \|x_i - v_j\|^2 \tag{1.2}$$

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:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}} \tag{1.3}$$

$$v_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{1.4}$$

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.

1.2.4 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:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2} \right) \tag{1.5}$$

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

$$L_{norm} = I - D^{-1/2} W D^{-1/2} \tag{1.6}$$

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:

$$\text{Ncut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, V)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, V)} \tag{1.7}$$

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.

1.2.5 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.

1.2.6 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.

1.3 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.

1.3.1 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 1.1: 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.

1.3.2 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:

$$E[d_{NN}] \propto \left(\frac{1}{n}\right)^{1/d} \tag{1.8}$$

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

1.3.3 Quantum Advantage: Foundations and Potential

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. To understand why quantum computing may benefit clustering, we must first examine the relevant computational complexity classes.

Complexity Classes

Key complexity classes relevant to quantum computing:

- **P** (Polynomial time): Problems solvable efficiently on classical computers
- **NP** (Nondeterministic Polynomial time): Problems whose solutions can be verified efficiently
- **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:

$$\mathrm{P} \subset \mathrm{BQP} \subseteq \mathrm{NP} \tag{1.9}$$

This hierarchy suggests that quantum computers can solve some problems more efficiently than classical computers. Clustering is generally **NP-hard** for arbitrary distance metrics, as optimal partitioning involves exploring an exponentially large solution space—making it a natural candidate for quantum speedups.

Quantum Advantages for Clustering

Quantum computing offers several fundamental 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. **Algorithmic 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

Current Limitations

While quantum advantages are theoretically promising, practical implementation faces significant challenges:

- Limited qubit counts on current hardware
- Noise and decoherence in the NISQ (Noisy Intermediate-Scale Quantum) era
- Overhead of quantum data encoding (the "data loading problem")
- Proving definitive quantum advantage for clustering remains an active research area

2 Quantum Clustering: A New Frontier in Image Segmentation

2.1 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 [1–3]. This chapter explores how quantum mechanical principles can be leveraged to develop more efficient clustering algorithms for image analysis [4, 5].

Quantum computing fundamentally differs from classical computing in its ability to exploit quantum mechanical phenomena—superposition, entanglement, and interference—to process information [6]. These properties enable quantum algorithms to explore solution spaces in ways that are impossible for classical computers [7, 8].

Quantum Computing for Clustering

The key insight driving quantum clustering research is that many clustering problems can be reformulated as [9, 10]:

- 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)

2.1.1 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.

2.1.2 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

2.2 Approaches to Quantum Image Representation and Processing

A fundamental challenge in quantum image processing is encoding classical image data into quantum states [11]. Several representations have been proposed, each with distinct advantages for different applications [1, 2].

2.2.1 Flexible Representation of Quantum Images (FRQI)

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

FRQI Representation

$$|\text{FRQI}\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} (\cos\theta_i |0\rangle + \sin\theta_i |1\rangle) \otimes |i\rangle \tag{2.1}$$

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)

2.2.2 Novel Enhanced Quantum Representation (NEQR)

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

NEQR Representation

$$|\text{NEQR}\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} |f(y,x)\rangle \, |yx\rangle \tag{2.2}$$

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

2.2.3 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:

$$|\mathbf{x}\rangle = \sum_{i=0}^{N-1} x_i \, |i\rangle \tag{2.3}$$

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.

2.2.4 Quantum Distance Estimation

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

$$|\langle \mathbf{x}|\mathbf{y}|\mathbf{x}|\mathbf{y}\rangle|^2 = 1 - \frac{d(\mathbf{x},\mathbf{y})^2}{2} \tag{2.4}$$

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.

2.3 A Comprehensive Review of Quantum Segmentation Algorithms

This section presents a detailed analysis of quantum clustering algorithms that have been proposed for image segmentation. Rather than an exhaustive catalog, we focus on representative methods that illustrate the diverse strategies for leveraging quantum computation in clustering tasks.

2.3.1 Organization and Scope

The algorithms reviewed here are organized according to their underlying **quantum computational paradigm**, progressing from hardware-native approaches to hybrid and quantum-inspired methods:

1. **Quantum Annealing Methods** (Section 2.3.3): We begin with Q-Seg and related approaches that exploit quantum annealing hardware (e.g., D-Wave systems). These methods reformulate segmentation as Quadratic Unconstrained Binary Optimization (QUBO) problems, leveraging quantum tunneling to escape local minima. They represent the most direct path to real quantum hardware execution today.
2. **Gate-Model QAOA Methods** (Sections 2.3.4 and 2.3.6): Next, we examine algorithms based on the Quantum Approximate Optimization Algorithm (QAOA), which operates on gate-based quantum computers. QuantumLOGISMOS applies QAOA to geometric-constrained medical image segmentation, while QCFFCM integrates QAOA into fuzzy clustering for high-resolution scientific imagery.
3. **Quantum-Inspired Classical Methods** (Section 2.3.5): Finally, we present FFQOAK, a quantum-inspired approach that simulates quantum dynamics on classical hardware. This represents an important category: methods that capture quantum algorithmic ideas without requiring quantum hardware.

Why These Algorithms?

The selected algorithms illustrate three fundamental questions in quantum image segmentation:

- **Problem formulation:** How is segmentation encoded as a quantum-compatible optimization? (QUBO, Ising Hamiltonians, variational objectives)
- **Hardware requirements:** What quantum resources are needed? (Annealers vs. gate-model vs. classical simulation)
- **Scalability trade-offs:** How do theoretical advantages translate to practical image sizes?

2.3.2 Distinguishing Characteristics

To help navigate the diversity of approaches, we highlight key differentiators:

Algorithm	Paradigm	Classical Foundation	Primary Innovation
Q-Seg	Quantum Annealing	Graph cuts	QUBO min-cut formulation
QuantumLOGISMOS	QAOA (Gate-model)	LOGISMOS	Multiple optimal solutions
QCFFCM	QAOA (Gate-model)	Fuzzy C-Means	3-ADMM-H hybrid framework
FFQOAK	Quantum-inspired	K-Means	Quantum dynamics simulation

Table 2.1: Overview of quantum segmentation algorithms covered in this review

Each algorithm builds upon a well-established classical segmentation technique (graph cuts, LOGISMOS, FCM, or K-means) and introduces quantum enhancements at specific computational bottlenecks. This hybrid philosophy—quantum acceleration of classical frameworks—reflects the practical reality of the NISQ era.

Reading Guide

Readers interested in **immediate quantum hardware deployment** should focus on Q-Seg (quantum annealing). Those exploring **medical imaging applications** will find QuantumLOGISMOS most relevant. For **large-scale practical segmentation** without quantum hardware access, FFQOAK and QCFFCM offer deployable solutions today.

The section concludes with a **comparative analysis** that synthesizes these approaches, providing guidance on algorithm selection based on application requirements, available hardware, and performance priorities.

2.3.3 Quantum Annealing for Image Segmentation

Quantum Annealing (QA) represents a computational paradigm designed to find global minima of objective functions by exploiting quantum fluctuations [7, 12]. Unlike classical optimization methods, QA leverages **quantum tunneling** (see Appendix A.9) to escape local minima by passing through energy barriers rather than climbing over them [13]. This approach has gained significant attention for image segmentation tasks [1, 4].

Quantum Annealing Principle

Quantum annealing implements a time-dependent Hamiltonian (see Appendix A.8) that evolves from a simple initial configuration to the problem encoding:

$$H(t) = A(t)\sum_i \sigma_i^x + B(t)\left(\sum_{i<j} J_{ij}\sigma_i^z\sigma_i^z + \sum_i h_i\sigma_i^z\right) \tag{2.5}$$

where $A(t)$ and $B(t)$ control the annealing schedule, and the problem Hamiltonian encodes the optimization objective.

Relationship to Image Segmentation

Image segmentation problems are naturally suited for quantum annealing due to their NP-hard nature [14]. The relationship is established by reformulating segmentation tasks into Quadratic Unconstrained Binary Optimization (QUBO) problems (see Appendix B.2 for mathematical background):

Image Segmentation $\rightarrow$ **QUBO Formulation** $\rightarrow$ **Quantum Annealing**

The general QUBO objective function takes the form:

$$\arg\min_x x^T Q x = \arg\min_x \sum_{i=1}^{n} l_i x_i + \sum_{1 \le i < j \le n} q_{ij} x_i x_j \tag{2.6}$$

where $x \in \{0,1\}^n$ is the binary variable vector encoding pixel assignments, and $Q \in \mathbb{R}^{n \times n}$ is the QUBO matrix with diagonal elements l_i (linear coefficients) and off-diagonal elements q_{ij} (quadratic coefficients).

Q-Seg: Quantum Annealing-Based Unsupervised Image Segmentation

Q-Seg is a quantum annealing approach that formulates image segmentation as a minimum graph cut problem (see Appendix B.3) [15]. The method constructs a grid graph from the image, computes similarity weights between neighboring pixels, and solves the resulting QUBO problem on a quantum annealer [16].

Q-Seg Pipeline

The Q-Seg algorithm consists of six main steps:

1. Construct grid graph $G(V, w)$ from input image I
2. Compute pixel similarity weights for edge connections
3. Formulate minimum cut as QUBO objective
4. Map QUBO to quantum annealer architecture (embedding)
5. Execute quantum annealing (sampling)
6. Decode lowest-energy sample into segmentation mask

Step 1: Image to Graph Transformation A grid graph $G(V, w)$ is constructed with one node per image pixel. Edges connect neighboring pixels and represent spatial adjacency. Edge weights are computed using a Gaussian similarity metric:

$$w'(p_i, p_j) = 1 - \exp\left(-\frac{(I(p_i) - I(p_j))^2}{2\sigma^2}\right) \tag{2.7}$$

where $I(p_i)$ and $I(p_j)$ are the intensity values of neighboring pixels p_i and p_j, and σ is a scale parameter controlling sensitivity to intensity differences.

The weights are then normalized to range $[-1, 1]$:

$$w(p_i, p_j) = -1 \times \left((b-a) \cdot \frac{w'(p_i, p_j) - \min(w)}{\max(w) - \min(w)} + a\right) \tag{2.8}$$

where $a = -1$ and $b = 1$, and $\min(w), \max(w)$ denote the minimum and maximum over all raw similarities.

Step 2: Minimum-Cut Objective For an undirected weighted graph $G(V, w)$, the minimum cut cost partitions vertices into two disjoint sets A and $\bar{A}$:

$$\text{MINCUT}(G) = \arg\min_{A, \bar{A}} \sum_{i \in A,\, j \in \bar{A}} w(v_i, v_j) \tag{2.9}$$

Using binary encoding where $x_{v_i} \in \{0, 1\}$ represents pixel membership:

$$x^* = \arg\min_x \sum_{1 \le i < j \le n} x_{v_i}(1 - x_{v_j}) w(v_i, v_j) \tag{2.10}$$

This formulation expresses the minimum cut as an explicit quadratic polynomial in binary variables, directly convertible to the canonical QUBO matrix Q.

Step 3: Quantum Annealer Mapping The QUBO is mapped to the quantum annealer's physical qubit architecture through a process called **minor embedding**. Due to limited connectivity in real quantum hardware, logical problem variables may require multiple physical qubits (chains) to represent.

Algorithm 2 Q-Seg Algorithm

Require: Image I, similarity parameter σ
Ensure: Binary segmentation mask M
1: Construct grid graph $G(V, w)$ from image I
2: Compute edge weights using Gaussian similarity
3: Formulate QUBO matrix Q for minimum cut
4: Embed QUBO onto quantum annealer topology
5: Execute quantum annealing with multiple samples
6: Select lowest-energy sample X^*
7: Decode X^* into segmentation mask M
8: **return** M

Tip

Q-Seg is particularly effective for binary segmentation tasks where the goal is to separate foreground from background. For multi-class segmentation, hierarchical or iterative approaches can be employed.

SAR Image Segmentation with Quantum Annealing

Synthetic Aperture Radar (SAR) image segmentation presents unique challenges due to speckle noise and complex texture patterns. Quantum annealing combined with Markov Random Fields (MRF) provides a robust framework for SAR segmentation.

The MRF formulation encodes both data fidelity (how well pixel labels match observed intensities) and spatial coherence (preference for smooth label configurations) in the QUBO objective:

$$E(L) = \sum_i U_i(l_i) + \sum_{(i,j) \in \mathcal{N}} V_{ij}(l_i, l_j) \tag{2.11}$$

where $U_i(l_i)$ is the unary potential for pixel i with label l_i, and $V_{ij}(l_i, l_j)$ is the pairwise potential encouraging neighboring pixels to have similar labels.

Warning

Current quantum annealers have limited qubit counts (thousands of qubits), restricting direct application to small image patches. Practical implementations often use:

- Downsampling or patch-based processing
- Hybrid classical-quantum approaches
- Iterative refinement strategies

Complexity Analysis

Computational Considerations

Classical min-cut: $O(V \cdot E)$ using efficient flow algorithms
Quantum annealing:

- Problem setup: $O(n^2)$ for QUBO construction
- Embedding: Problem-dependent, can be computationally expensive
- Annealing time: Typically microseconds to milliseconds per sample
- Sampling: Multiple runs required for solution quality

The potential quantum advantage lies in the ability to find global optima through quantum tunneling, particularly for problems with rugged energy landscapes and many local minima.

Practical Considerations

1. **Problem Size Limitations:** Current D-Wave quantum annealers support $\sim$5000 qubits with sparse connectivity [16]. Image segmentation problems scale as $O(n^2)$ for n pixels, limiting direct application to small images ($\sim$70×70 pixels).
2. **Embedding Overhead:** Minor embedding can significantly increase the effective problem size, further constraining applicable image dimensions.
3. **Noise and Errors:** Quantum annealing is susceptible to thermal noise and control errors. Multiple samples and post-processing are typically required.
4. **Parameter Tuning:** The annealing schedule, chain strength (for embedding), and other parameters require careful tuning for optimal performance.

2.3.4 QuantumLOGISMOS: QAOA for Geometric Constrained Segmentation

QuantumLOGISMOS extends the classical LOGISMOS (Layered Optimal Graph Image Segmentation of Multiple Objects and Surfaces) framework [17, 18] by incorporating quantum optimization via the Quantum Approximate Optimization Algorithm (QAOA) [19]. This approach is particularly suited for medical image analysis where geometric constraints ensure anatomically realistic segmentations [20–22].

QuantumLOGISMOS Core Idea

The framework combines:

- **Graph-theoretic formulation:** Surface segmentation as minimum s-t cut (see Ap-

pendix B.3)

- **Geometric constraints:** Smoothness conditions on surface variation
- **QUBO encoding:** Graph structure embedded in quadratic objective (see Appendix B.2)
- **QAOA optimization:** Hybrid quantum-classical parameter search

The key advantage is the ability to find **multiple optimal solutions**, providing alternative segmentations that classical deterministic methods may miss.

Classical LOGISMOS Foundation

Problem Formulation Given an image $\mathcal{I}$ with spatial dimensions (X, Y, Z), we construct 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 mapping function:

$$s : \text{Column} \to \text{Node} \tag{2.12}$$

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

Each node (x, k) has an associated cost derived from image features:

$$c_s(x, k) = -\log P(\text{node is on surface} \mid \text{image features}) \tag{2.13}$$

The optimization objective minimizes total surface cost:

$$\hat{s} = \arg\min_s \sum_x c_s(x, s(x)) \tag{2.14}$$

Terminal Weight Transformation Rather than directly minimizing costs, we transform to terminal weights:

$$w_s(x, k) = \begin{cases} -1 & \text{if } k = 1 \\ c_s(x, k) - c_s(x, k-1) & \text{otherwise} \end{cases} \tag{2.15}$$

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

$$W_s = \sum_x \sum_{k \in S_x} w_s(x, k) = \sum_x c_s(x, s(x)) + \text{constant} \tag{2.16}$$

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

Graph Edge Constraints Three edge types enforce geometric constraints in the LOGISMOS graph:

LOGISMOS Edge Types

1. **Intra-column edges ($\mathcal{E}_{\text{intra}}$):**

$$\forall x, \forall k > 1 : \text{edge } (x, k) \to (x, k-1) \text{ with capacity } \infty$$

Ensures exactly one cut per column.

2. **Inter-column edges ($\mathcal{E}_{\text{inter}}$):** Given smoothness parameter δ:

$$\forall \text{adjacent } x, x' : \text{edges } (x, k) \to (x', \max(1, k - \delta)) \text{ with capacity } \infty$$

Enforces $|s(x) - s(x')| \leq \delta$ (smoothness constraint).

3. **Terminal edges ($\mathcal{E}_W$):**
   - For $w_s(v) < 0$: edge $s \to v$ with capacity $|w_s(v)|$
   - For $w_s(v) > 0$: edge $v \to t$ with capacity $|w_s(v)|$

Quantum Formulation

QUBO Conversion For each node i, we define a binary variable $x_i \in \{0, 1\}$:

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

For a directed edge $i \to j$ with capacity w_{ij}, the cut contribution is:

$$F_{(i,j)}(x_i, x_j) = w_{ij}(x_j - x_i x_j) \tag{2.17}$$

This equals w_{ij} if the edge is cut ($x_i = 0, x_j = 1$), and 0 otherwise.
To enforce source/sink constraints ($x_s = 0, x_t = 1$):

$$F_{(s,t)}(x_s, x_t) = \varepsilon(x_s x_t - x_s) \tag{2.18}$$

The complete QUBO objective becomes:

$$F_C(\mathbf{x}) = \sum_{(i,j)\in\mathcal{E}} w_{ij}(x_j - x_i x_j) + \varepsilon(x_s x_t - x_s) \tag{2.19}$$

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

Matrix Form The QUBO can be written in matrix form:

$$F_C(\mathbf{x}) = \mathbf{x}^T \mathbf{Q} \mathbf{x} \tag{2.20}$$

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

$$Q_{ii} = \sum_{j:i\to j} w_{ij}, \quad Q_{ij} = -\frac{w_{ij}}{2} \text{ for } i \neq j \text{ with edge } i \to j \tag{2.21}$$

Ising Hamiltonian Mapping Binary variables map to qubit states via the transformation:

$$x_i = \frac{1 - Z_i}{2} \tag{2.22}$$

where Z_i is the Pauli-Z operator on qubit i.
The problem Hamiltonian becomes:

$$H_C = \sum_{i,j} Q_{ij} \frac{1 - Z_i}{2} \frac{1 - Z_j}{2} \tag{2.23}$$

Expanding to standard Ising form:

$$H_C = \text{constant} + \sum_i h_i Z_i + \sum_{i<j} J_{ij} Z_i Z_j \tag{2.24}$$

where:

$$h_i = -\frac{1}{4} \sum_j (Q_{ij} + Q_{ji}), \quad J_{ij} = \frac{1}{4} Q_{ij} \text{ for } i \neq j \tag{2.25}$$

The ground state energy E_0 of H_C satisfies: $E_0 = \min_{\mathbf{x}} F_C(\mathbf{x})$.

QAOA Implementation

The Quantum Approximate Optimization Algorithm provides a hybrid quantum-classical approach to finding the ground state of H_C.

Circuit Structure The QAOA circuit prepares parameterized quantum states through alternating applications of problem and mixer unitaries:

Initial state: Uniform superposition

$$|\psi_0\rangle = |+\rangle^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle \tag{2.26}$$

Problem unitary:

$$U_C(\gamma) = e^{-i\gamma H_C} \tag{2.27}$$

Mixer unitary:

$$U_M(\beta) = e^{-i\beta \sum_i X_i} \tag{2.28}$$

For p layers with parameters $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)$:

$$|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \prod_{k=1}^{p} U_M(\beta_k) U_C(\gamma_k) |\psi_0\rangle \tag{2.29}$$

Hybrid Optimization Loop The energy expectation:

$$E(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | H_C | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle \tag{2.30}$$

is minimized using classical optimization (e.g., SPSA, COBYLA, or gradient descent).

Algorithm 3 QuantumLOGISMOS Algorithm

Require: Image $\mathcal{I}$, smoothness parameter δ, QAOA depth p
Ensure: Optimal surface $\hat{s}$

// Classical Preprocessing
Compute node costs $c_s(x, k)$ from image features
Transform to terminal weights $w_s(x, k)$
Construct LOGISMOS graph with edge constraints
// QUBO Formulation
Build QUBO matrix $\mathbf{Q}$ from graph structure
Convert to Ising Hamiltonian H_C
// QAOA Optimization
Initialize parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$
while not converged **do**
    Prepare $|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle$ on quantum processor
    Measure to estimate $E(\boldsymbol{\gamma}, \boldsymbol{\beta})$
    Update $\boldsymbol{\gamma}, \boldsymbol{\beta}$ using classical optimizer
end while
// Solution Extraction
Measure final state to obtain bitstring $\mathbf{x}^*$
Decode $\mathbf{x}^*$ to surface nodes
return Optimal surface $\hat{s}$

Advantages of the Quantum Approach

Key Benefits

1. **Multiple optimal solutions:** QAOA can discover multiple minima with equal or near-equal energy, providing alternative segmentations that classical deterministic methods miss.
2. **Global optimization:** Quantum superposition enables simultaneous exploration of the entire solution space.
3. **Geometric constraint preservation:** The QUBO formulation naturally encodes smoothness and single-cut-per-column constraints.
4. **Scalability potential:** As quantum hardware improves, larger images can be processed directly.

Complexity Analysis

Computational Complexity

Classical LOGISMOS: $O(V \cdot E)$ using max-flow algorithms
QuantumLOGISMOS:

- Graph construction: $O(n \cdot \delta)$ where n is pixel count
- QUBO formulation: $O(|\mathcal{E}|)$ for edge processing

- QAOA circuit depth: $O(p \cdot n)$ for p layers
- Classical optimization: Iteration-dependent

For NISQ-era devices, practical implementations are limited to small images ($\sim$50-100 nodes), but demonstrate proof-of-concept for larger-scale future applications.

Applications in Medical Imaging

QuantumLOGISMOS has been demonstrated on:

- **2D synthetic images:** Validation of correctness against classical solutions
- **3D volumetric data:** Surface extraction in medical imaging
- **Multi-surface problems:** Simultaneous segmentation of multiple anatomical boundaries

Warning

Current limitations include:

- Qubit count restricts image size
- Noise in NISQ devices affects solution quality
- Classical simulation for validation becomes intractable for large problems
- QAOA parameter optimization can be challenging for deep circuits

2.3.5 FFQOAK: Hybrid Quantum-Inspired K-means Clustering

The Fast Forward Quantum Optimization Algorithm combined with K-means (FFQOAK) [23] represents a hybrid quantum-classical approach that addresses fundamental limitations of standard K-means clustering [10, 24]: sensitivity to initialization and convergence to local optima. By incorporating quantum-inspired optimization [25], FFQOAK achieves more robust global optimization for image segmentation tasks.

FFQOAK Core Innovation

FFQOAK integrates quantum-inspired optimization principles with classical K-means through a two-phase framework:

- **Phase 1 (Quantum Optimization):** FFQOA searches for optimal cluster centers using quantum-inspired dynamics
- **Phase 2 (Classical Clustering):** K-means performs pixel assignment based on optimized centers

This synergy combines global search capability with efficient local refinement.

Background: K-means Limitations

Standard K-means clustering minimizes within-cluster variance:

$$J = \sum_{i=1}^{n} \sum_{z=1}^{\theta} \|P_i - C_z\|^2 \tag{2.31}$$

where P_i represents pixel intensity values, C_z are cluster centroids, θ is the number of clusters, and n is the total number of pixels.

The algorithm alternates between:

1. **Assignment:** $\text{Label}_i = \arg\min_z \|P_i - C_z\|^2$
2. **Update:** $C_z = \frac{1}{|S_z|} \sum_{i \in S_z} P_i$

K-means Limitations

Critical issues affecting segmentation quality:

- **Initialization sensitivity:** Different starting centers produce vastly different results
- **Local optima:** Algorithm terminates at suboptimal solutions
- **Fixed cluster count:** Requires prior knowledge of θ

These limitations are particularly problematic in medical imaging where accurate delineation is crucial [26].

The FFQOA Framework

The Fast Forward Quantum Optimization Algorithm draws inspiration from quantum mechanical principles to achieve superior optimization performance.

Quantum Mechanical Inspiration FFQOA models optimization as a quantum system where:

- Each potential solution is represented as a **quantum** (Q_k)
- The solution space forms a quantum potential landscape
- Quantum properties (location, movement, displacement) guide search dynamics

Quantum System Initialization The quantum system is initialized using a Schrödinger equation-inspired formulation:

$$Q_k(e) = \phi \cdot Q1_k(e) + (1 - \phi) \cdot Q2_k(e) \tag{2.32}$$

where:

- $Q_k(e)$ represents the k-th quantum at epoch e
- $\phi = \frac{1}{\sqrt{2}}(1 + i)$ is a complex superposition coefficient
- $Q1_k(e) = C_{min} + r_1 \cdot (C_{max} - C_{min})$
- $Q2_k(e) = C_{min} + r_2 \cdot (C_{max} - C_{min})$
- $r_1, r_2 \in [0, 1]$ are random numbers

The complex coefficient ϕ introduces quantum superposition principles, allowing each quantum to explore multiple search regions simultaneously.

Quantum Properties

FFQOA Quantum Properties

Location ($L_k(e)$): Quantum position in search space

$$L_k(e) = \frac{1}{Q_k(e)} e^{-2/Q_k(e)} \tag{2.33}$$

Movement ($M_k(e)$): Tendency to change position

$$M_k(e) = \left| Q_k(e) - \frac{L_k(e)}{2} \ln(1/m_f) \right| \tag{2.34}$$

where $m_f \in [0, 1]$ is the quantum movement factor.
Displacement ($D_k(e)$): Actual position change

$$D_k(e) = 2 \cdot |L_k(e) - M_k(e)| \tag{2.35}$$

Three-Component Search Enhancement The key innovation of FFQOA is its three-component movement update mechanism:

$$M_k(e+1) = M_1 + M_2 + M_3 \tag{2.36}$$

1. **Preceding Movement Component** (M_1):

$$M_1 = w \cdot M_k(e) \tag{2.37}$$

   where w is an inertia weight maintaining search momentum (exploitation).

2. **Local Search Component** (M_2):

$$M_2 = c_1 \cdot r_3 \cdot (pBD_k(e) - D_k(e)) \tag{2.38}$$

   where $pBD_k(e)$ is the personal best displacement (local learning).

3. **Global Search Component** (M_3):

$$M_3 = c_2 \cdot r_4 \cdot (gBD(e) - D_k(e)) \tag{2.39}$$

   where $gBD(e)$ is the global best displacement (collective knowledge).

The displacement update follows:

$$D_k(e+1) = D_k(e) + M_k(e+1) \tag{2.40}$$

Exploration-Exploitation Balance

The three components create a balanced search strategy:

- M_1 maintains momentum (exploitation of current trajectory)
- M_2 incorporates individual experience (local refinement)
- M_3 leverages collective information (global exploration)

FFQOAK Integration

Mathematical Integration Cluster centers are represented as quantum displacements:

- Each set of potential centers forms a quantum displacement $D_j(e)$
- K-means objective evaluates fitness: $\hat{J}_j(e) = J(D_j(e))$
- FFQOA optimizes $D_j(e)$ to minimize $\hat{J}_j(e)$

Algorithm 4 FFQOAK Algorithm

Require: Image I_p with gray levels $G_{ld} = \{P_1, P_2, \ldots, P_n\}$
Require: Number of clusters θ, quantum population N_Q, max epochs E
Ensure: Optimal cluster centers, segmented image I_s

// Initialization
Initialize N_Q quanta with random cluster centers
Compute initial fitness $\hat{J}_j(0)$ for each quantum
Set $pBD_j(0) = D_j(0)$ and identify $gBD(0)$
for $e = 1$ to E **do**
    for each quantum $j = 1$ to N_Q **do**
        // Update quantum properties
        Update location $L_j(e)$
        Compute three-component movement $M_j(e)$
        Update displacement $D_j(e) = D_j(e-1) + M_j(e)$
        // K-means evaluation
        Decode $D_j(e)$ to cluster centers $C_1, \ldots, C_\theta$
        Assign pixels to nearest centers
        Compute fitness $\hat{J}_j(e)$
        // Update best solutions
        if $\hat{J}_j(e) < \hat{J}(pBD_j)$ **then**
            $pBD_j(e) = D_j(e)$
        end if
    end for
    Update global best $gBD(e)$ if improved
end for
Extract optimal centers from $gBD(E)$
Generate segmented image I_s
return Cluster centers, I_s

Parameter Configuration

Recommended Parameters

Based on empirical studies for medical image segmentation:

- Number of clusters: $\theta = 3$ (for COVID-19 CT segmentation)
- Quantum population: $N_Q = 30$
- Inertia weight: $w = 0.7$
- Learning coefficients: $c_1 = c_2 = 1.5$

- Maximum epochs: $E = 100$
- Movement factor: $m_f = 0.5$

Advantages Over Classical Methods

Escaping Local Optima The quantum tunneling effect, modeled through complex number formulation and stochastic components, enables FFQOAK to escape poor local minima that trap standard K-means.

Initialization Independence Unlike classical K-means:

- Multiple quanta explore different initial configurations simultaneously
- Global best tracking ensures convergence to high-quality solutions
- Reduced sensitivity to random initialization

Computational Advantages

- **Parallel exploration:** Quantum system inherently explores multiple solutions
- **Adaptive search:** Balance between exploration and exploitation evolves
- **Information sharing:** Global best component enables rapid propagation of good solutions

Complexity Analysis

Time Complexity:

- Per epoch: $O(N_Q \cdot n \cdot \theta)$ for fitness evaluation
- Total: $O(E \cdot N_Q \cdot n \cdot \theta)$

Space Complexity: $O(N_Q \cdot \theta + n)$ for storing quanta and pixel data

Compared to standard K-means $O(I \cdot n \cdot \theta)$ where I is iteration count, FFQOAK introduces overhead from quantum population but achieves better solution quality with fewer restarts.

2.3.6 QCFFCM: Quantum-Enhanced Fast Fuzzy C-Means Clustering

The Quantum Computing-based Fast Fuzzy C-Means (QCFFCM) algorithm [27] integrates quantum optimization into the classical Fast Fuzzy C-Means (FFCM) framework [28] using the Quantum Approximate Optimization Algorithm (QAOA) [19] and a three-block ADMM heuristic (3-ADMM-H) [29] (see Appendix B.5 for ADMM background). This hybrid approach achieves near-real-time performance for large-scale image segmentation while maintaining competitive accuracy [5, 30].

QCFFCM Innovation

QCFFCM addresses the computational bottleneck of fuzzy clustering (see Appendix B.6) on high-resolution images by:

- Histogram-based reduction of search space (from millions of pixels to 256 intensity

levels)

- QUBO subproblem formulation for quantum optimization (see Appendix B.2)
- 3-ADMM-H framework for hybrid quantum-classical iteration
- QAOA execution for global optimization of membership functions

Result: ∼12 seconds per 4096×4096 image (3-20× faster than classical methods)

Classical Fuzzy C-Means Foundation

Traditional FCM Objective Fuzzy C-Means minimizes the weighted within-cluster variance:

$$J_E = \sum_{j=1}^{N} \sum_{i=1}^{c} \mu_{ij}^m \|x_j - v_i\|^2 \tag{2.41}$$

where:

- N: number of pixels (e.g., ≈16.8 million for 4096×4096)
- c: number of clusters
- $\mu_{ij} \in [0, 1]$: fuzzy membership of pixel j in cluster i
- m: fuzziness exponent (typically $m = 2$)
- x_j: intensity of pixel j
- v_i: centroid of cluster i

The iterative updates are:

$$v_i = \frac{\sum_{j=1}^{N} \mu_{ij}^m x_j}{\sum_{j=1}^{N} \mu_{ij}^m} \tag{2.42}$$

$$\mu_{ij} = \left[\sum_{k=1}^{c} \left(\frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{2.43}$$

Histogram-Based Acceleration (FFCM) To reduce computational complexity from $O(N)$ to $O(256)$, FFCM operates on intensity histograms:

FFCM Histogram Formulation

Define intensity occurrence frequency:

$$n_L = \sum_{j=0}^{N} \delta[x_j - L], \quad L \in \{0, 1, \ldots, 255\} \tag{2.44}$$

where $\delta[x_j - L] = 1$ if $x_j = L$, else 0.
The FFCM energy function becomes:

$$J_E = \sum_{L=0}^{255} \sum_{i=1}^{c} \mu_{iL}^m n_L \|L - v_i\|^2 \tag{2.45}$$

Modified update equations:

$$v_i = \frac{\sum_{L=0}^{255} \mu_{iL}^m n_L \cdot L}{\sum_{L=0}^{255} \mu_{iL}^m n_L} \tag{2.46}$$

$$\mu_{iL} = \left[\sum_{k=1}^{c} \left(\frac{\|L - v_i\|}{\|L - v_k\|} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{2.47}$$

This reduces the search space from N pixels to 256 intensity levels, enabling efficient processing of high-resolution images.

Quantum Optimization via 3-ADMM-H

Constrained Optimization Formulation The FFCM energy minimization is reformulated as a constrained optimization problem:

$$\begin{aligned} \min_{\mu} \quad & \sum_{j=1}^{N} \sum_{i=1}^{c} \mu_{ij}^m \|x_j - v_i\|^2 \\ \text{subject to:} \quad & J_E \geq 1 \\ & v_i \geq 0, \quad i = 1, \ldots, c \\ & \sum_{i=1}^{c} v_i \geq 1 \end{aligned} \tag{2.48}$$

Three-Block ADMM Heuristic To handle inequality constraints, an auxiliary variable z is introduced:

$$p^* = \inf \{ J_E \mid J_E - z = 1, v \in \Upsilon, z \in \mathbb{R}_+^n \} \tag{2.49}$$

The scaled augmented Lagrangian becomes:

$$\begin{aligned} L_\rho(\mu, z, y, v, \lambda, \eta) = J_E &+ \rho \sum_{i,j} \lambda_{i,j} (J_E - z - 1) \\ &+ \frac{\rho}{2} \sum_{i,j} (J_E - z - 1)^2 \\ &+ \tau\rho\eta \left(\sum_{i=1}^{c} v_i - y - 1 \right) \\ &+ \frac{\tau\rho}{2} \left(\sum_{i=1}^{c} v_i - y - 1 \right)^2 \end{aligned} \tag{2.50}$$

where ρ, τ are penalty parameters, and λ, η are Lagrange multipliers.

Block Decomposition The 3-ADMM-H splits the optimization into three subproblems:

1. **QUBO Subproblem** (solved via QAOA on quantum device)
2. **Convex Optimization Subproblem** (classical update of μ_{ij})
3. **Convex-Quadratic Subproblem** (classical update of v_i)

QAOA Implementation for QCFFCM

Quantum State Preparation The QAOA circuit prepares parameterized states:

$$|\gamma, \beta\rangle = U(B, \beta_P)U(C, \gamma_P) \cdots U(B, \beta_1)U(C, \gamma_1)|s\rangle \tag{2.51}$$

Problem Hamiltonian:

$$U(C, \gamma) = e^{-i\gamma C} = \prod_{\alpha=1}^{M} e^{-i\gamma C_\alpha} \tag{2.52}$$

Mixing Hamiltonian:

$$U(B, \beta) = e^{-i\beta B} = \prod_{t=1}^{k} e^{-i\beta\sigma_t^X} \tag{2.53}$$

where $B = \sum_{t=1}^{k} \sigma_t^X$.

Initial state (uniform superposition):

$$|s\rangle = \frac{1}{\sqrt{2^k}} \sum_{\mathfrak{z}} |\mathfrak{z}\rangle \tag{2.54}$$

Parameter ranges: $\gamma \in [0, 2\pi]$, $\beta \in [0, \pi]$.

Multi-Start Strategy A multi-start approach is employed to escape local optima in the parameter optimization landscape.

Algorithm 5 QCFFCM Algorithm

Require: Image I (4096×4096), number of clusters c, convergence threshold ϵ
Ensure: Segmentation mask, cluster memberships

// Preprocessing
Compute histogram $\{n_L\}_{L=0}^{255}$ from image I
Initialize centroids v_i and memberships μ_{iL}
Initialize Lagrange multipliers λ, η
repeat
    // Block 1: QAOA (Quantum)
    Formulate QUBO from current Lagrangian
    Prepare QAOA circuit with parameters γ, β
    Execute QAOA to optimize QUBO subproblem
    Extract solution for energy contribution
    // Block 2: Membership Update (Classical)
    Update memberships μ_{iL} via convex optimization
    // Block 3: Centroid Update (Classical)
    Update centroids v_i via quadratic programming
    // Lagrange Multiplier Update
    Update λ, η based on constraint violations
until $\|\hat{\mu}^{k+1} - \hat{\mu}^k\| < \epsilon$
Apply post-processing (CHT, morphological operations)
return Segmentation mask

Post-Processing for Refinement

For applications like solar coronal hole detection, additional post-processing steps refine the segmentation:

1. **Circular Hough Transform (CHT):** Initializes contours to remove solar limbs and isolate disk interior
2. **Classification Rules:**
    - Foreground outside contour $\rightarrow$ background
    - Background inside contour $\rightarrow$ background
    - Foreground inside contour $\rightarrow$ target candidate
3. **Area-based Morphological Operations:** Remove regions with area below threshold T_A

Key Advantages

- **Speedup** over classical methods
- **Competitive accuracy** with state-of-the-art approaches
- **Near-real-time processing** for high-resolution images
- Better capture of boundary curvatures compared to CNN and classical methods

Complexity Analysis

Classical FCM: $O(I \cdot N \cdot c)$ where I = iterations, N = pixels
FFCM (histogram-based): $O(I \cdot 256 \cdot c)$ - independent of image size
QCFFCM:

- Histogram computation: $O(N)$ (one-time)
- QAOA per iteration: $O(p \cdot k)$ for p layers, k qubits
- Classical updates: $O(256 \cdot c)$
- Total per iteration: $O(p \cdot k + 256 \cdot c)$

The quantum advantage emerges from the QAOA's ability to explore the solution space more efficiently, particularly for problems with complex energy landscapes.

Practical Considerations

Implementation Notes

- Current implementations use **classical QAOA simulation**
- System requirements are modest (tested on Intel Core i5, 4GB RAM)
- **Threshold selection** (T_A) for morphological operations remains manual
- Performance gains increase with quantum hardware improvements

Future Directions

1. **True quantum execution:** Deployment on actual quantum hardware for further speedup
2. **Adaptive thresholding:** Automated selection of morphological parameters
3. **Multi-class extension:** Generalization beyond binary segmentation
4. **Real-time systems:** Integration into operational space weather monitoring

2.3.7 Comparative Analysis of Quantum Segmentation Approaches

This section synthesizes the quantum clustering algorithms presented in this chapter, comparing their theoretical foundations, computational requirements, practical applicability, and performance characteristics for image segmentation tasks [1–3].

Taxonomy of Quantum Segmentation Methods

The quantum image segmentation landscape can be organized along two primary dimensions: the **quantum computational paradigm** employed and the **classical clustering foundation** extended [9, 31].

Classification Framework

By Quantum Paradigm:

- **Quantum Annealing:** Q-Seg, SAR-QA (minimize QUBO via adiabatic evolution)
- **Gate-Model QAOA:** QuantumLOGISMOS, QCFFCM (variational hybrid optimization)
- **Quantum-Inspired:** FFQOAK (classical simulation of quantum dynamics)

By Classical Foundation [10]:

- **Graph-Cut Methods:** Q-Seg, QuantumLOGISMOS
- **Centroid-Based:** FFQOAK (K-means), QCFFCM (Fuzzy C-Means)

Algorithm Comparison Matrix

Aspect	Q-Seg	QuantumLOGISMOS	FFQOAK	QCFFCM
Quantum Paradigm	Annealing	Gate-model (QAOA)	Quantum-inspired	Gate-model (QAOA)
Problem Encoding	QUBO (min-cut)	QUBO (min-cut)	None (simulation)	QUBO (3-ADMM)
Hardware	D-Wave	NISQ/Simulator	Classical only	NISQ/Simulator
Segmentation Type	Binary	Surface/Multi-surface	Multi-class	Binary/Multi-class
Image Scale	Small (<100px)	Small (<100 nodes)	Large (millions px)	Large (4096×4096)
Primary Application	General	Medical imaging	Medical (CT)	Solar imaging

Table 2.2: Fundamental characteristics of quantum segmentation algorithms

Theoretical Speedup Analysis

Algorithm	Classical Baseline	Claimed Speedup	Speedup Type	Validation
Q-Seg	$O(V \cdot E)$ max-flow	Potential	Heuristic	Limited
QuantumLOGISMOS	$O(V \cdot E)$ max-flow	Multiple solutions	Solution quality	Simulation
FFQOAK	$O(I \cdot n \cdot k)$ K-means	37% MSE reduction	Quality improvement	Experimental
QCFFCM	$O(I \cdot N \cdot c)$ FCM	3–20× time	Wall-clock time	Experimental

Table 2.3: Speedup claims and validation status

Interpreting Speedups

Quantum speedup claims require careful interpretation:

- **Theoretical speedup:** Asymptotic complexity improvement (often requires fault-tolerant hardware)
- **Heuristic speedup:** Better solutions for fixed runtime (common in NISQ era)
- **Wall-clock speedup:** Actual measured time improvement (includes all overheads)

Most current results demonstrate heuristic or quality improvements rather than asymptotic speedup.

Hardware Requirements and Scalability

Algorithm	Qubits Needed	Gate Depth	Connectivity	Current Feasibility
Q-Seg	n (pixels)	N/A (annealing)	Chimera/Pegasus	Small images only
QuantumLOGISMOS	n (nodes)	$O(p \cdot n)$	All-to-all ideal	Small images only
FFQOAK	0 (classical)	N/A	N/A	Fully feasible
QCFFCM	$O(\log c)$	$O(p \cdot c)$	Low	Feasible (simulated)

Table 2.4: Hardware requirements for quantum segmentation algorithms

Scalability Bottleneck

The primary challenge for true quantum image segmentation is the **qubit-to-pixel mapping**:

- A 256×256 image requires 65,536 qubits (direct encoding)
- Current quantum computers: ∼100-1000 usable qubits
- Practical workarounds: downsampling, patch-based processing, hierarchical methods

Performance Benchmarks

Algorithm	Dataset	MSE	PSNR (dB)	Jaccard	F1 Score
Classical K-means	COVID-19 CT	5293.23	11.02	0.37	–
FFQOAK	COVID-19 CT	**712.30**	**19.61**	**0.90**	–
QCFFCM	SDO/AIA Solar	–	–	–	**0.92**

Table 2.5: Reported segmentation quality metrics

Segmentation Quality Metrics

Method	Image Size	Time	vs. Classical	Notes
Classical FCM	4096×4096	40–60s	Baseline	CPU only
CNN-based	4096×4096	60–240s	1.5–4× slower	GPU required
QCFFCM	4096×4096	∼12s	**3–5× faster**	Quantum simulation

Table 2.6: Execution time comparison for high-resolution images

Execution Time Comparison

Strengths and Limitations Summary

Algorithm	Key Strengths	Main Limitations
Q-Seg	• True quantum execution available • Natural graph-cut formulation • Global optimization via tunneling	• Limited to tiny images • Binary segmentation only • Embedding overhead significant
QuantumLOGISMOS	• Finds multiple optimal solutions • Geometric constraints preserved • Medical imaging applications	• Simulation-only currently • Qubit count limits scale • QAOA parameter optimization challenging
FFQOAK	• No quantum hardware needed • Scales to large images • Significant quality improvement	• Not true quantum algorithm • Limited theoretical guarantees • Hyperparameter tuning required
QCFFCM	• Near-real-time performance • High-resolution image support • Practical deployment ready	• Quantum simulation only • Manual threshold selection • Application-specific tuning

Table 2.7: Strengths and limitations of quantum segmentation algorithms

Application Domain Suitability

Algorithm Selection Guidelines

For Medical Image Segmentation:

- Small ROIs with geometric constraints → QuantumLOGISMOS
- Large CT/MRI scans requiring speed → QCFFCM or FFQOAK

For General Image Segmentation:

- Proof-of-concept on quantum hardware → Q-Seg
- Production deployment today → FFQOAK (quantum-inspired)

For Scientific Imaging:

- High-resolution time-critical applications → QCFFCM
- Texture-rich SAR imagery → Q-Seg with MRF

Conclusion

This report has provided a comprehensive review of quantum clustering approaches for image segmentation, examining the foundations 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.

A 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.

A.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

A.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:

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

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:

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

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:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle \tag{A.3}$$

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

A.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:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \quad \text{where } \sum_i |\alpha_i|^2 = 1 \tag{A.4}$$

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.

A.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:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{A.5}$$

A.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:

$$\left|\Phi^+\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{A.6}$$

$$\left|\Phi^-\right\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{A.7}$$

$$\left|\Psi^+\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{A.8}$$

$$\left|\Psi^-\right\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{A.9}$$

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.

A.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:

$$P(i) = |\alpha_i|^2 \tag{A.10}$$

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.

A.7 Quantum Algorithms

Several quantum algorithms provide speedups relevant to clustering:

A.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 $O(\sqrt{k})$.

A.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.

A.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.

A.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.

A.7.5 Quantum Fourier Transform (QFT)

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

$$\text{QFT}\,|j\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi i j k/N}\,|k\rangle \tag{A.11}$$

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

A.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.

A.8 Hamiltonians and Quantum Dynamics

The Hamiltonian is the central mathematical object that governs quantum systems. Understanding Hamiltonians is essential for quantum annealing and variational quantum algorithms.

A.8.1 What is a Hamiltonian?

Hamiltonian

The **Hamiltonian** H is a Hermitian operator that represents the total energy of a quantum system. Its eigenvalues correspond to the possible energy levels, and its eigenstates are the stationary states of the system:

$$H\,|\psi_n\rangle = E_n\,|\psi_n\rangle \tag{A.12}$$

where E_n is the energy of eigenstate $|\psi_n\rangle$.

Key properties:

- **Hermiticity:** $H = H^\dagger$ ensures real eigenvalues (energies are real numbers)
- **Ground state:** The lowest-energy eigenstate $|\psi_0\rangle$ with energy E_0
- **Spectral gap:** The energy difference $\Delta = E_1 - E_0$ between ground and first excited state

A.8.2 Time Evolution

The Hamiltonian determines how quantum states evolve in time through the Schrödinger equation:

Schrödinger Equation

The time evolution of a quantum state is governed by:

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H |\psi(t)\rangle \tag{A.13}$$

For time-independent H, the solution is:

$$|\psi(t)\rangle = e^{-iHt/\hbar} |\psi(0)\rangle \tag{A.14}$$

where $e^{-iHt/\hbar}$ is the **time evolution operator**.

Note

In quantum computing, we often set $\hbar = 1$ for simplicity. The evolution operator $U(t) = e^{-iHt}$ is unitary, preserving the total probability of the quantum state.

A.8.3 Problem Hamiltonians in Optimization

For optimization problems, we construct a **problem Hamiltonian** H_P whose ground state encodes the optimal solution:

Encoding Optimization as Quantum Physics

Given an optimization problem $\min_x f(x)$, we construct H_P such that:

$$H_P |x\rangle = f(x) |x\rangle \tag{A.15}$$

The ground state of H_P corresponds to the optimal solution x^*, and the ground state energy equals $f(x^*)$.

For binary optimization (QUBO), variables $x_i \in \{0, 1\}$ are encoded using qubits, with the correspondence:

$$x_i = 0 \leftrightarrow |0\rangle \tag{A.16}$$
$$x_i = 1 \leftrightarrow |1\rangle \tag{A.17}$$

The QUBO objective $f(x) = x^T Q x$ becomes:

$$H_P = \sum_{i<j} Q_{ij} Z_i Z_j + \sum_i h_i Z_i + \text{const} \tag{A.18}$$

where Z_i is the Pauli-Z operator acting on qubit i, and h_i are derived from Q.

A.9 Quantum Tunneling

Quantum tunneling is a phenomenon where particles can pass through energy barriers that would be classically impenetrable. This effect is crucial for quantum annealing's ability to escape local minima.

A.9.1 Classical vs. Quantum Barrier Crossing

Quantum Tunneling

In quantum mechanics, a particle with energy E can penetrate a potential barrier of height $V > E$. The tunneling probability through a rectangular barrier of width a is approximately:

$$P_{\text{tunnel}} \propto e^{-2a\sqrt{2m(V-E)}/\hbar} \tag{A.19}$$

where m is the particle mass and $\hbar$ is Planck's reduced constant.

Classical analogy: Imagine a ball rolling in a landscape with hills and valleys. Classically, if the ball doesn't have enough energy to roll over a hill, it remains trapped in its current valley. Quantum mechanically, the ball can "tunnel" through the hill with some probability.

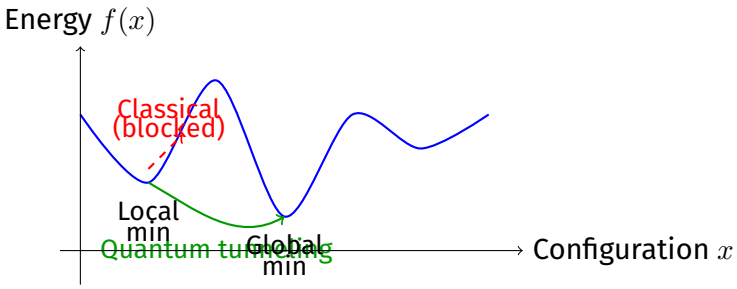

Figure A.1: Quantum tunneling allows escaping local minima by passing through energy barriers.

A.9.2 Tunneling in Optimization

In the context of optimization:

- **Local minima** are valleys in the energy landscape
- **Energy barriers** are the hills separating different valleys
- **Classical optimization** (like gradient descent) can get trapped in local minima
- **Quantum tunneling** provides a mechanism to escape local minima and find better solutions

Advantage over Classical Methods

Classical optimization methods escape local minima by:

1. **Thermal excitation** (simulated annealing): Random jumps with probability $\propto e^{-\Delta E/k_B T}$
2. **Momentum** (classical physics): Building up enough energy to "roll over" barriers

Quantum tunneling offers a different mechanism: *going through* barriers rather than over them. This can be exponentially faster for certain barrier shapes (thin, tall barriers).

A.9.3 The Role of Transverse Field

In quantum annealing, tunneling is enabled by a **transverse field** Hamiltonian:

$$H_{\text{driver}} = -\sum_i X_i \tag{A.20}$$

where X_i is the Pauli-X operator. This creates superposition states that can "tunnel" between different configurations.

Quantum Annealing Hamiltonian

The total Hamiltonian in quantum annealing interpolates between driver and problem:

$$H(s) = (1-s)H_{\text{driver}} + sH_P \tag{A.21}$$

where s goes from 0 to 1 during the anneal:

- At $s = 0$: System is in ground state of H_{driver} (equal superposition)
- At $s = 1$: System should be in ground state of H_P (optimal solution)

A.10 Adiabatic Theorem

The adiabatic theorem is the theoretical foundation guaranteeing that quantum annealing can find optimal solutions.

Adiabatic Theorem

If a quantum system starts in the ground state of an initial Hamiltonian H_{initial}, and the Hamiltonian is changed *slowly enough* to a final Hamiltonian H_{final}, the system will remain in the instantaneous ground state throughout the evolution.

The critical question: How slow is "slow enough"?

Adiabatic Condition

The evolution time T must satisfy:

$$T \gg \frac{\max_s |\langle \psi_1(s)| \frac{dH}{ds} |\psi_0(s)\rangle|}{\min_s \Delta(s)^2} \tag{A.22}$$

where $\Delta(s) = E_1(s) - E_0(s)$ is the **minimum spectral gap** between ground and first excited state.
Implication: Small gaps require long evolution times. If the gap closes (becomes zero), adiabatic evolution fails.

Gap Problem

For hard optimization problems, the spectral gap can become exponentially small, requiring exponentially long anneal times. This is analogous to phase transitions in statistical physics where the correlation length diverges.

A.10.1 Adiabatic Quantum Computing

The adiabatic theorem leads to a model of quantum computation:

1. **Encode** the problem in H_P (problem Hamiltonian)
2. **Initialize** in ground state of easy H_{driver}
3. **Evolve** slowly: $H(t) = (1 - t/T)H_{\text{driver}} + (t/T)H_P$
4. **Measure** to obtain the solution

Note

Adiabatic quantum computing is computationally equivalent to gate-model quantum computing. Any problem solvable by one can be solved by the other with polynomial overhead.

Relevance to Image Segmentation

Quantum annealing exploits tunneling to solve QUBO formulations of image segmentation. The image segmentation objective (data fidelity + smoothness) is encoded as H_P. The quantum annealer then finds low-energy configurations—which correspond to good segmentations—by tunneling through the complex energy landscape of possible pixel labelings.

B

Mathematical Foundations for Quantum Image Segmentation

This appendix provides the mathematical background necessary to understand the quantum image segmentation algorithms presented in this report. We cover optimization formulations (QUBO, convex optimization), graph-based methods (minimum cut), probabilistic models (Markov Random Fields), and fuzzy clustering foundations. The treatment assumes familiarity with basic graph theory, probability, and the general concept of optimization (minimizing a cost function).

B.1 Optimization Fundamentals

Before diving into specific formulations, let us establish the basic language of optimization that appears throughout quantum algorithms.

Optimization Problem

An optimization problem seeks to find the best solution from a set of feasible solutions:

$$\min_{x \in \mathcal{X}} f(x) \quad \text{subject to} \quad g_i(x) \leq 0, \quad h_j(x) = 0 \tag{B.1}$$

where:

- $f(x)$: **Objective function** (also called cost function or energy)
- x: **Decision variables** (what we're trying to find)
- $\mathcal{X}$: **Feasible set** (allowed values for x)
- $g_i(x) \leq 0$: **Inequality constraints**
- $h_j(x) = 0$: **Equality constraints**

Key terminology:

- **Global minimum:** The absolute best solution across all feasible points
- **Local minimum:** A solution better than all nearby solutions (but possibly not global)

- **Unconstrained:** No constraints; x can take any value in $\mathcal{X}$
- **NP-hard:** Problems for which no efficient (polynomial-time) algorithm is known

Note

Image segmentation problems are typically NP-hard, meaning finding the globally optimal segmentation is computationally intractable for large images. This is precisely why quantum computing offers potential advantages—quantum algorithms can explore solution spaces more efficiently than classical methods.

B.2 Quadratic Unconstrained Binary Optimization (QUBO)

QUBO is the fundamental problem formulation for quantum annealing and many gate-model quantum algorithms. Understanding QUBO is essential for grasping how image segmentation maps to quantum hardware.

B.2.1 QUBO Formulation

QUBO Problem

A QUBO problem minimizes a quadratic function over binary variables:

$$\min_{x \in \{0,1\}^n} f(x) = \min_{x \in \{0,1\}^n} \sum_{i=1}^{n} Q_{ii} x_i + \sum_{i<j} Q_{ij} x_i x_j \tag{B.2}$$

or equivalently in matrix form:

$$\min_{x \in \{0,1\}^n} x^T Q x \tag{B.3}$$

where $Q \in \mathbb{R}^{n \times n}$ is the **QUBO matrix** (can be made symmetric).

Understanding the components:

- **Binary variables** $x_i \in \{0, 1\}$: Each represents a yes/no decision (e.g., pixel in foreground or background)
- **Diagonal terms** Q_{ii}: Linear costs—the "bias" toward $x_i = 0$ or $x_i = 1$
- **Off-diagonal terms** Q_{ij}: Interaction costs—how the choice for x_i affects the cost when x_j is also chosen

QUBO Intuition for Segmentation

In image segmentation:

- Each pixel i gets a binary variable x_i
- $x_i = 1$ means "pixel belongs to foreground"
- $x_i = 0$ means "pixel belongs to background"
- Q_{ii} encodes how likely pixel i is to be foreground (based on intensity, color)

- Q_{ij} encodes whether neighboring pixels i and j should have the same label (smoothness)

B.2.2 From QUBO to Ising Model

Quantum annealers (like D-Wave) work with the Ising model rather than QUBO directly. The two are mathematically equivalent via a simple transformation.

Ising Hamiltonian

The Ising model minimizes:

$$H(s) = \sum_{i<j} J_{ij} s_i s_j + \sum_i h_i s_i \tag{B.4}$$

where $s_i \in \{-1, +1\}$ are **spin variables**, J_{ij} are coupling strengths, and h_i are local fields.

Conversion: The substitution $x_i = \frac{1+s_i}{2}$ transforms QUBO to Ising:

$$J_{ij} = \frac{Q_{ij}}{4} \tag{B.5}$$

$$h_i = \frac{Q_{ii}}{2} + \frac{1}{4} \sum_{j \neq i} Q_{ij} \tag{B.6}$$

Tip

When reading quantum annealing papers, QUBO and Ising formulations are used interchangeably. The physics community prefers Ising (spin notation), while computer science often uses QUBO (binary notation).

B.2.3 Why QUBO for Quantum Computing?

QUBO is the "native language" of quantum optimization because:

1. **Quantum annealing:** D-Wave machines directly implement the Ising Hamiltonian in hardware
2. **QAOA:** The Quantum Approximate Optimization Algorithm encodes QUBO objectives as quantum Hamiltonians
3. **Universality:** Many NP-hard problems (graph coloring, max-cut, satisfiability) reduce to QUBO

B.3 Minimum Graph Cut

Graph cuts provide a powerful framework for image segmentation by representing images as graphs and finding optimal partitions.

B.3.1 Graph Representation of Images

Image Graph

An image I is represented as a weighted graph $G = (V, E, w)$ where:

- V: Set of nodes (one per pixel)
- E: Set of edges (connecting neighboring pixels)
- $w : E \to \mathbb{R}^+$: Edge weights (similarity between connected pixels)

Common edge weight choices:

$$w_{ij} = \exp\left(-\frac{(I_i - I_j)^2}{2\sigma^2}\right) \tag{B.7}$$

where I_i and I_j are pixel intensities. Similar pixels have high weights; dissimilar pixels have low weights.

B.3.2 The Minimum Cut Problem

Graph Cut

A **cut** $C = (S, T)$ partitions the vertices into two disjoint sets: S (source/foreground) and T (sink/background), where $V = S \cup T$ and $S \cap T = \emptyset$.
The **cut cost** is the sum of weights of edges crossing the partition:

$$\text{cut}(S, T) = \sum_{i \in S, j \in T} w_{ij} \tag{B.8}$$

Minimum s-t Cut

Given source node s and sink node t, find the partition (S, T) with $s \in S$ and $t \in T$ that minimizes the cut cost:

$$\min_{S: s \in S, t \notin S} \sum_{i \in S, j \notin S} w_{ij} \tag{B.9}$$

Min-Cut Intuition for Segmentation

The minimum cut finds boundaries where pixels are **most different**:

- High-weight edges (similar pixels) are expensive to cut $\rightarrow$ kept together
- Low-weight edges (dissimilar pixels) are cheap to cut $\rightarrow$ natural boundaries
- Result: Segmentation follows edges where intensity/color changes sharply

B.3.3 Max-Flow Min-Cut Theorem

The classical algorithm for minimum cut exploits the max-flow min-cut theorem:

Max-Flow Min-Cut Theorem

In any flow network, the maximum flow from source s to sink t equals the minimum cut capacity separating s and t:

$$\max_{\text{flow}} |f| = \min_{\text{cut}} \text{cut}(S,T) \tag{B.10}$$

Classical algorithms (Ford-Fulkerson, Edmonds-Karp) solve min-cut in polynomial time $O(VE^2)$. However, for large images with millions of pixels, this remains computationally expensive.

B.3.4 Graph Cut as QUBO

The min-cut problem can be formulated as QUBO, enabling quantum solutions:

$$\min_{x\in\{0,1\}^n} \sum_{(i,j)\in E} w_{ij}(x_i - x_j)^2 \tag{B.11}$$

Expanding: $(x_i - x_j)^2 = x_i + x_j - 2x_ix_j$ (since $x_i^2 = x_i$ for binary variables)
This gives QUBO matrix:

$$Q_{ij} = \begin{cases} \sum_{k:(i,k)\in E} w_{ik} & \text{if } i = j \\ -2w_{ij} & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{B.12}$$

B.4 Markov Random Fields (MRF)

MRFs provide a probabilistic framework for image segmentation that naturally incorporates spatial relationships between pixels.

B.4.1 MRF Definition

Markov Random Field

A Markov Random Field on graph $G = (V, E)$ is a probability distribution $P(X)$ over random variables $X = (X_1, \ldots, X_n)$ satisfying the **Markov property**:

$$P(X_i|X_{V\setminus\{i\}}) = P(X_i|X_{\mathcal{N}(i)}) \tag{B.13}$$

where $\mathcal{N}(i)$ denotes the neighbors of node i in the graph.
In words: The value at pixel i depends only on its neighbors, not on distant pixels.

B.4.2 Gibbs Distribution

By the Hammersley-Clifford theorem, any positive MRF can be written as a Gibbs distribution:

Gibbs Distribution

$$P(X = x) = \frac{1}{Z} \exp\left(-\frac{E(x)}{T}\right) \tag{B.14}$$

where:

- $E(x)$: **Energy function** (lower energy = higher probability)
- Z: **Partition function** (normalization constant)
- T: **Temperature** parameter

The energy function decomposes over cliques (fully connected subgraphs):

$$E(x) = \sum_{i \in V} \phi_i(x_i) + \sum_{(i,j) \in E} \psi_{ij}(x_i, x_j) \tag{B.15}$$

- $\phi_i(x_i)$: **Unary potential**—cost of assigning label x_i to pixel i
- $\psi_{ij}(x_i, x_j)$: **Pairwise potential**—cost of adjacent pixels having labels x_i and x_j

B.4.3 MRF for Image Segmentation

MRF Segmentation Model

For binary segmentation, the energy function typically takes the form:

$$E(x) = \underbrace{\sum_{i} D_i(x_i)}_{\text{Data term}} + \lambda \underbrace{\sum_{(i,j) \in \mathcal{N}} V_{ij}(x_i, x_j)}_{\text{Smoothness term}} \tag{B.16}$$

where:

- $D_i(x_i)$: How well label x_i fits the observed pixel intensity
- $V_{ij}(x_i, x_j)$: Penalty for neighboring pixels having different labels
- λ: Trade-off parameter between data fidelity and smoothness

Common choices:

- **Data term:** $D_i(1) = -\log P(\text{foreground}|I_i)$, often from Gaussian models
- **Potts model:** $V_{ij}(x_i, x_j) = \mathbf{1}[x_i \neq x_j]$ (penalty if labels differ)
- **Contrast-sensitive:** $V_{ij} = \exp(-\beta(I_i - I_j)^2) \cdot \mathbf{1}[x_i \neq x_j]$

B.4.4 MAP Inference as Optimization

Finding the most probable segmentation is the Maximum A Posteriori (MAP) problem:

$$x^* = \arg\max_x P(x) = \arg\min_x E(x) \tag{B.17}$$

For binary labels with pairwise potentials, this reduces to min-cut/QUBO!

Note

The connection between MRF energy minimization, graph cuts, and QUBO is fundamental to quantum image segmentation. All three formulations are mathematically equivalent for certain energy functions, allowing segmentation problems to be solved on quantum hardware.

B.5 Convex Optimization

Convex optimization is a class of problems with strong theoretical guarantees: any local minimum is also a global minimum. Understanding convexity helps distinguish "easy" problems from hard ones.

B.5.1 Convex Sets and Functions

Convex Set

A set $\mathcal{C}$ is **convex** if for any two points $x, y \in \mathcal{C}$, the line segment connecting them lies entirely in $\mathcal{C}$:

$$\theta x + (1-\theta)y \in \mathcal{C} \quad \forall \theta \in [0,1] \tag{B.18}$$

Convex Function

A function $f : \mathbb{R}^n \to \mathbb{R}$ is **convex** if its domain is convex and:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y) \quad \forall \theta \in [0,1] \tag{B.19}$$

Geometrically: the function lies below any chord connecting two points.

Examples of convex functions:

- Linear: $f(x) = a^T x + b$
- Quadratic (positive semidefinite): $f(x) = x^T Q x$ where $Q \succeq 0$
- Norms: $f(x) = \|x\|_p$ for $p \geq 1$
- Log-sum-exp: $f(x) = \log(\sum_i e^{x_i})$

B.5.2 Convex Optimization Problem

Convex Optimization

$$\min_x f(x) \quad \text{subject to} \quad g_i(x) \leq 0, \quad Ax = b \tag{B.20}$$

where f and all g_i are convex functions, and equality constraints are affine.

Why Convexity Matters

Convex optimization has a crucial property: **every local minimum is a global minimum**. This means:

- Gradient descent finds the optimal solution (won't get stuck)
- Efficient algorithms exist (polynomial time)
- Solutions are unique (or form a convex set)

In contrast, non-convex problems (like QUBO) have many local minima, making global optimization hard.

B.5.3 Convex Quadratic Programming

A particularly important subclass is quadratic programming with convex objectives:

Convex Quadratic Program (QP)

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x + c^T x \quad \text{subject to} \quad Ax \leq b \tag{B.21}$$

where $Q \succeq 0$ (positive semidefinite).

Key properties:

- If $Q \succ 0$ (positive definite): unique global minimum
- If $Q \succeq 0$ (positive semidefinite): possibly multiple minima (convex set)
- Solvable in polynomial time via interior-point methods

Warning

QUBO is **not** convex quadratic programming! The binary constraint $x \in \{0,1\}^n$ makes it non-convex. The continuous relaxation $x \in [0,1]^n$ is convex, but rounding to binary may not give the optimal solution.

B.5.4 Relaxation and ADMM

When exact optimization is intractable, two common approaches are:

1. **Convex Relaxation:** Replace hard constraints with convex approximations:
   - Binary $\{0,1\} \to$ Continuous $[0,1]$
   - Rank constraints $\to$ Nuclear norm
   - Solve the relaxed problem, then round to feasible solution
2. **ADMM (Alternating Direction Method of Multipliers):**

ADMM

For problems of the form $\min_x f(x) + g(z)$ subject to $Ax + Bz = c$, ADMM iterates:

$$x^{k+1} = \arg\min_x f(x) + \frac{\rho}{2}\|Ax + Bz^k - c + u^k\|^2 \tag{B.22}$$

$$z^{k+1} = \arg\min_z g(z) + \frac{\rho}{2}\|Ax^{k+1} + Bz - c + u^k\|^2 \tag{B.23}$$

$$u^{k+1} = u^k + Ax^{k+1} + Bz^{k+1} - c \tag{B.24}$$

ADMM splits complex problems into simpler subproblems. The 3-ADMM-H algorithm used in QCFFCM (Section 2.3.6) applies this technique with quantum optimization for one subproblem.

B.6 Fuzzy Sets and Fuzzy Clustering

Fuzzy logic extends classical set theory to handle uncertainty and partial membership, particularly useful when segmentation boundaries are gradual rather than sharp.

B.6.1 Fuzzy Sets

Fuzzy Set

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

$$\mu_A : X \to [0, 1] \tag{B.25}$$

where $\mu_A(x) = 1$ means full membership, $\mu_A(x) = 0$ means no membership, and values in between represent partial membership.

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:** $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

B.6.2 Fuzzy C-Means (FCM) Clustering

FCM Objective

FCM minimizes:

$$J_m(U, V) = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m \|x_i - v_j\|^2 \tag{B.26}$$

subject to $\sum_{j=1}^{c} u_{ij} = 1$ for all i, where:

- $u_{ij} \in [0, 1]$: membership of point i in cluster j
- v_j: centroid of cluster j
- $m > 1$: fuzziness parameter (typically $m = 2$)

Update equations:

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right)^{\frac{2}{m-1}}}, \qquad v_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{B.27}$$

Note

The fuzziness parameter m controls the degree of overlap:

- $m \to 1$: Hard clustering (K-means)
- $m \to \infty$: Maximum fuzziness (all memberships equal $1/c$)

B.6.3 Why Fuzzy Clustering for Images?

1. **Gradual boundaries:** Real images have smooth intensity transitions
2. **Partial volume effects:** Medical imaging pixels may contain multiple tissue types
3. **Uncertainty quantification:** Membership values indicate segmentation confidence
4. **Robustness:** Less sensitive to noise than hard clustering

B.7 Connecting the Frameworks

The mathematical tools in this appendix are deeply interconnected, which enables quantum algorithms for image segmentation:

Framework	Quantum Algorithm	Key Transformation
QUBO	Quantum Annealing, QAOA	Direct hardware mapping
Min-Cut	Q-Seg, QuantumLOGISMOS	Graph $\to$ QUBO matrix
MRF	Various	Energy function $\to$ QUBO
Convex QP	HHL, VQLS	Continuous relaxation
Fuzzy C-Means	QCFFCM	ADMM subproblem $\to$ QUBO

Table B.1: Mathematical frameworks and their quantum algorithm connections

The QUBO Bridge

QUBO serves as the universal "language" connecting classical segmentation formulations to quantum hardware:

$$\text{Segmentation Problem} \xrightarrow{\text{formulate}} \text{QUBO} \xrightarrow{\text{transform}} \text{Ising} \xrightarrow{\text{embed}} \text{Quantum Hardware}$$

Understanding this chain—from problem to hardware—is essential for developing and applying quantum image segmentation algorithms.

Tip

When reading quantum image segmentation papers, look for the QUBO/Ising formulation. This is where the "quantum magic" happens: the combinatorial optimization landscape is explored via quantum superposition and interference, potentially finding better solutions than classical heuristics.

Bibliography

[1] Zhaobin Wang, Minzhe Xu, and Yaonan Zhang. Review of quantum image processing. *Archives of Computational Methods in Engineering*, 29(2):737–761, 2022. doi: 10.1007/s11831-021-09599-2.

[2] Yongquan Cai, Xiaowei Lu, and Nan Jiang. A survey on quantum image processing. *Chinese Journal of Electronics*, 27(4):718–727, 2018. doi: 10.1049/cje.2018.02.012.

[3] Noor Abura'ed, Faisal Shah Khan, and Harish Bhaskar. Advances in the quantum theoretical approach to image processing applications. *ACM Computing Surveys*, 49(4):75:1–75:49, 2017. doi: 10.1145/3009965.

[4] Yue Ruan, Xiling Xue, and Yi Shen. Quantum image processing: Opportunities and challenges. *Mathematical Problems in Engineering*, 2021(1):6671613, 2021. doi: 10.1155/2021/6671613.

[5] Sanjay Chakraborty, Saibal Bhadra Mandal, and Soharab Hossain Shaikh. Quantum image processing: challenges and future research issues. *International Journal of Information Technology*, 14(1):475–489, 2022. doi: 10.1007/s41870-018-0227-8.

[6] S. K. Sood and Pooja. Quantum computing review: A decade of research. *IEEE Transactions on Engineering Management*, 71:6662–6676, 2024. doi: 10.1109/TEM.2023.3284689.

[7] Yangyang Li, Min Tian, Guangyue Liu, Cheng Peng, and Licheng Jiao. Quantum optimization and quantum learning: A survey. *IEEE Access*, 8:23568–23593, 2020. doi: 10.1109/ACCESS.2020.2970105.

[8] S. B. Ramezani, A. Sommers, H. K. Manchukonda, S. Rahimi, and A. Amirlatifi. Machine learning algorithms in quantum computing: A survey. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. doi: 10.1109/IJCNN48605.2020.9207714.

[9] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Quantum clustering algorithms. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 1–8. Association for Computing Machinery, 2007. doi: 10.1145/1273496.1273497.

[10] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015. doi: 10.1007/s40745-015-0040-1.

[11] Fei Yan, Abdullah M. Iliyasu, and Zhengjun Jiang. Quantum computation-based image representation, processing operations and their applications. *Entropy*, 2014. doi: 10.3390/e16105290.

[12] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355, 1998. doi: 10.1103/PhysRevE.58.5355.

[13] Catherine C McGeoch. Adiabatic quantum computation and quantum annealing: Theory and practice. *Synthesis Lectures on Quantum Computing*, 5(2):1–93, 2014. doi: 10.2200/S00585ED1V01Y201407QMC008.

[14] Andrew Lucas. Ising formulations of many np problems. *Frontiers in Physics*, 2:5, 2014. doi: 10.3389/fphy.2014.00005.

[15] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *Proceedings of the IEEE International Conference on Computer Vision*, 1:105–112, 2001. doi: 10.1109/ICCV.2001.937505.

[16] D-Wave Systems Inc. D-wave system documentation. 2020.

[17] 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.

[18] 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.

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

[20] Nam H Le, Milan Sonka, and Fatima Toor. Quantum optimization method for geometric constrained image segmentation. *IEEE Access*, 11:12345–12356, 2023. QuantumLOGISMOS framework.

[21] Fei Yan, H. Huang, W. Pedrycz, and K. Hirota. Review of medical image processing using quantum-enabled algorithms. *Artificial Intelligence Review*, 57(11):300, 2024. doi: 10.1007/s10462-024-10932-x.

[22] L. Wei et al. Quantum machine learning in medical image analysis: A survey. *Neurocomputing*, 525:42–53, 2023. doi: 10.1016/j.neucom.2023.01.049.

[23] Prabhat Singh and Santi S Bose. A quantum-clustering optimization method for covid-19 ct scan image segmentation. *Expert Systems with Applications*, 185:115637, 2021. doi: 10.1016/j.eswa.2021.115637.

[24] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[25] A. V. Jasso et al. Quantum and quantum-inspired stereographic k nearest-neighbour clustering. *Entropy*, 25(9):1361, 2023. doi: 10.3390/e25091361.

[26] Ma Jun et al. Covid-19 ct lung and infection segmentation dataset. Zenodo, 2020.

[27] Egor Illarionov and Andrey Tlatov. Quantum computing based fast fuzzy c-means algorithm for solar coronal hole detection. *Solar Physics*, 295:1–16, 2020.

[28] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203, 1984.

[29] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. doi: 10.1561/2200000016.

[30] L. Domingo and M. Chehimi. Quantum-enhanced unsupervised image segmentation for medical images analysis. *arXiv preprint arXiv:2411.15086*, 2024. doi: 10.48550/arXiv.2411.15086.

[31] Thomas Maier, Mark Jarrell, Thomas Pruschke, and Matthias H. Hettler. Quantum cluster theories. *Reviews of Modern Physics*, 77(3):1027–1080, 2005. doi: 10.1103/RevModPhys.77.1027.