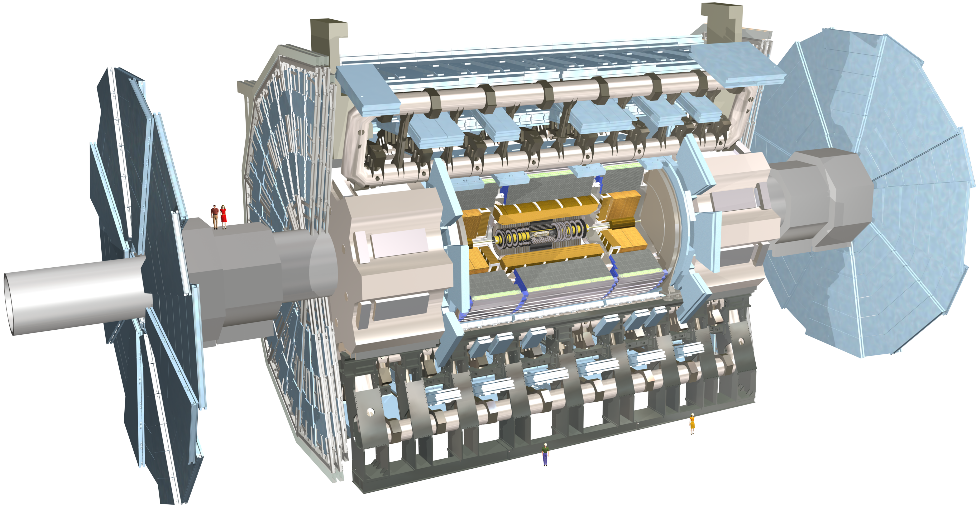
Quantum Hough Transform

Algorithm for Charged-Particle Track Reconstruction

08-25-2020

Amitabh Yadav
Heather Gray

HEP Quantum Pattern Recognition - https://hep-qpr.lbl.gov/



BERKELEY LAB
Lawrence Berkeley National Laboratory

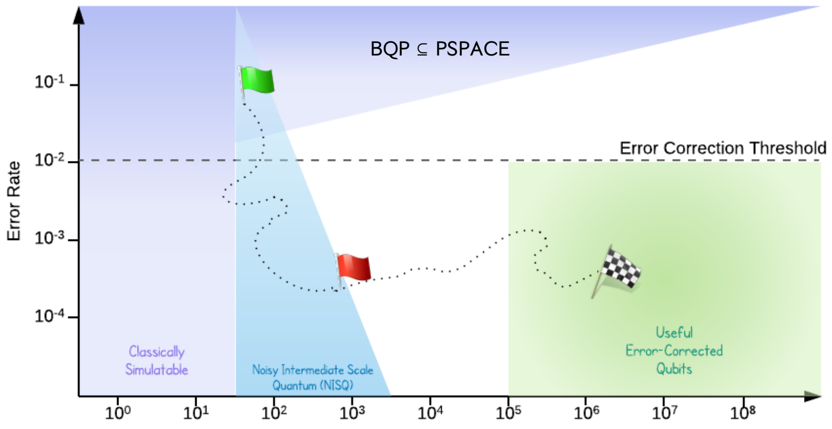
ATLAS EXPERIMENT

Introduction

- **Circuit Model QC**
  - Universal Quantum Computing
  - Solovay-Kitaev Theorem[KY97]

- **Quantum Computing Applications Approach**
  - Perfect Qubits
  - Erroneous Qubits in NISQ-era
    - Test algorithmic bounds of error-tolerance,
    - Pauli Frames,
    - Temporary Quantum Registers and SWAP gates,
    - Small (distance) EC Codes etc.

- **QPU vs Simulators platform**
  - Quantum accelerator model.
  - Good QC Packages out there for modern QC methods:
    (QISKit, Forest, Xanadu PennyLane, Entropica, QuEST, quantumSim, Project-Q, QX… to name a few)



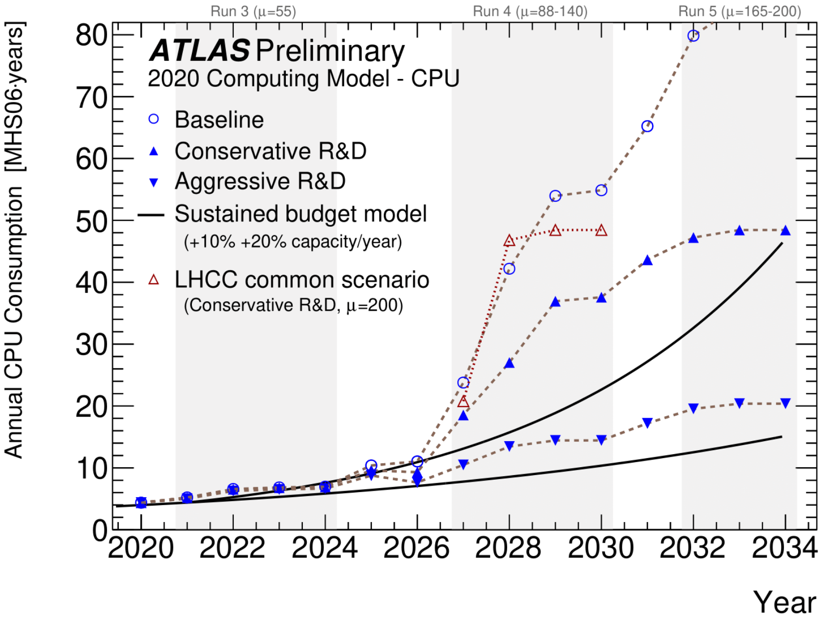
Ref: *https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html*

Ref.: Kitaev, Yu, *Quantum computations: algorithms and error correction,* Russian Mathematical Surveys(1997)

Hough Transform for Tracking

Why Hough Transform?

- Tracking: a key computational challenge.
- Simple formulation.
- P.V.C. Hough ('62): "…adaptable to the study of subatomic particle tracks…"



Hough Transform for Tracking

Why Hough Transform?

- Tracking: a key computational challenge.
- Simple formulation.
- P.V.C. Hough ('62): "…adaptable to the study of subatomic particle tracks…"

P.V.C. Hough (1962)

- Hough Transform is a method of finding groups of collinear points by a mathematically equivalent problem of finding concurrent lines. This method involves transforming each of the figure points into a curve in a parameter space.

R.O. Duda, P.E. Hart (1972)

- Uses angle-radius rather than slope-intercept parameters to simplify computation.

D.H. Ballard (1980) - Generalized Hough Transform for Arbitrary Shapes

- Uses n-Dimensional parameter space for arbitrary shapes.
- Detects simple and complex shapes.

United States Patent Office

3,069,654
Patented Dec. 18, 1962

1

3,069,654
METHOD AND MEANS FOR RECOGNIZING COMPLEX PATTERNS
Paul V. C. Hough, Ann Arbor, Mich., assignor to the United States of America as represented by the United States Atomic Energy Commission
Filed Mar. 25, 1960, Ser. No. 17,715
6 Claims. (Cl. 340—146.3)

This invention relates to the recognition of complex patterns and more specifically to a method and means for machine recognition of complex lines in photographs or other pictorial representations.

This invention is particularly adaptable to the study of subatomic particle tracks passing through a viewing field. As the objects to be studied in modern physics become smaller, the problem of observing these objects becomes increasingly more complex. One of the more useful devices in observing charged particles is the bubble chamber wherein the charged particles create tracks along their path of travel composed of small bubbles approximately 0.01 inch apart, depending upon the specific ionization of the initiating particle. These tracks form complex patterns and are readily photographed with the use of a dark background. With this device, multitudinous photographs are produced with each photograph requiring several hours

2

of the point on the line segment from the horizontal midline 109 of the framelet 108.

(3) Each line in the transformed plane is made to have an intercept with the horizontal midline 101 of the picture 100 equal to the horizontal coordinate of its respective point on the line segment in framelet 108.

Thus, for a given reference point 110 on line segment 102 a line 110A is drawn in the plane transform 102A. The reference point 110 is approximately midway between the top and the horizontal midline 109 of framelet 108 and hence the line 110A is inclined to the right at an angle to the vertical whose tangent is approximately ½. The intersection of the line 110A with the horizontal midline 101 of picture 100 is at a distance from the left edge of the picture 100 equal to the horizontal coordinate of the point 110 on line segment 102.

It is an exact theorem that, if a series of points in a framelet lie on a straight line, the corresponding lines in the plane transform intersect in a point which we shall designate as a knot 112. It is therefore readily apparent that the rectangular coordinates of the knots 112 in picture 100 have the following properties:

(1) The horizontal coordinates of the knots 112 equal the horizontal coordinates in the framelet 108 at which the straight line segments 102, 104 and 106 intercept the horizontal midline 109 of the framelet 108.

the knots 112, relative picture 100, is propor- of the straight line seg- to the vertical. Thus, in the plane transforms slopes and intercepts of 04 and 106 in framelet

Graphics and Image Processing — W. Newman, Editor

Use of the Hough Transformation To Detect Lines and Curves in Pictures

Richard O. Duda and Peter E. Hart
Stanford Research Institute,
Menlo Park, California

Hough has proposed an interesting and computationally efficient procedure for detecting lines in pictures. This paper points out that the use of angle-radius rather than slope-intercept parameters simplifies the computation further. It also shows how the method can be used for more general curve fitting, and gives alternative interpretations that explain the source of its efficiency.

Key Words and Phrases: picture processing, pattern recognition, line detection, curve detection, colinear points, point-line

CR Categorie

1. Introduction

A recurring problem in computer picture processing is the detection of straight lines in digitized images. In the simplest case, the picture contains a number of discrete, black figure points lying on a white background. The problem is to detect the presence of groups of colinear or almost colinear figure points. It is clear that the problem can be solved to any desired degree of accuracy by testing the lines formed by all pairs of points.

Pattern Recognition Vol. 13, No. 2, pp. 111–122, 1981.
Printed in Great Britain.

0031–3203/81/020111–12 $02.00/0
Pergamon Press Ltd.

GENERALIZING THE HOUGH TRANSFORM TO DETECT ARBITRARY SHAPES*

D. H. BALLARD
Computer Science Department, University of Rochester, Rochester, NY 14627, U.S.A.

(*Received* 10 *October* 1979; *in revised form* 9 *September* 1980; *received for publication* 23 *September* 1980)

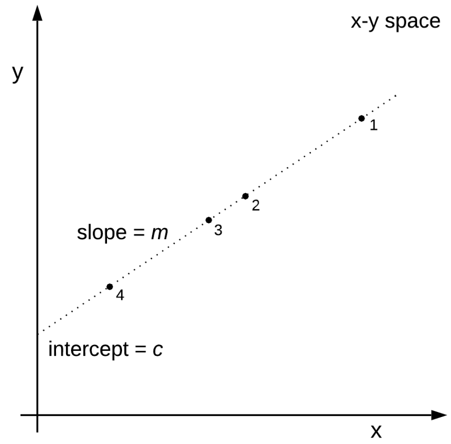
Abstract—The Hough transform is a method for detecting curves by exploiting the duality between points on a curve and parameters of that curve. The initial work showed how to detect both analytic curves[1,2] and non-analytic curves,[3] but these methods were restricted to binary edge images. This work was generalized to the detection of some analytic curves in grey level images, specifically lines,[4] circles[5] and parabolas.[6] The line detection case is the best known of these and has been ingeniously exploited in several applications.[7,8,9]

We show how the boundaries of an *arbitrary* non-analytic shape can be used to construct a mapping between image space and Hough transform space. Such a mapping can be exploited to detect instances of that particular shape in an image. Furthermore, variations in the shape such as rotations, scale changes or figure–ground reversals correspond to straightforward transformations of this mapping. However, the most remarkable property is that such mappings can be composed to build mappings for complex shapes from the mappings of simpler component shapes. This makes the generalized Hough transform a kind of universal transform which can be used to find arbitrarily complex shapes.

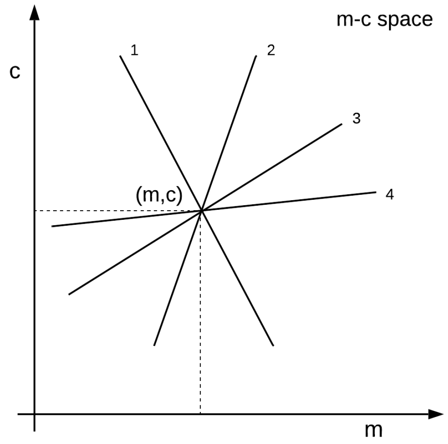
Hough Transform for Linear Track

$$y = mx + c$$

- Parameter Space
  - A hit in *xy*-space is represented as a line in *mc*-space.
  - Multiple hits renders multiple lines in parameter space.
  - The intersection (m, c) represents the line connecting the hits.
- Accumulator Space
  - Lets each hit point vote.
  - Voting allows us to give more weightage to hit points that fall along a line.
- Track Detection
  - The intersections are detected by means of voting in an accumulator matrix and detecting the local maximum.



(a) Image Space



(b) Parameter Space

Hough Transform for Linear Track

- Vertical lines: $m, c = unbounded$.
  This implies, accumulator needs infinite space.

- Mitigation: a different representation of the transformation.
  Point in (x, y) space is represented by a sinusoid in (r, θ).

- Line in polar coordinates:

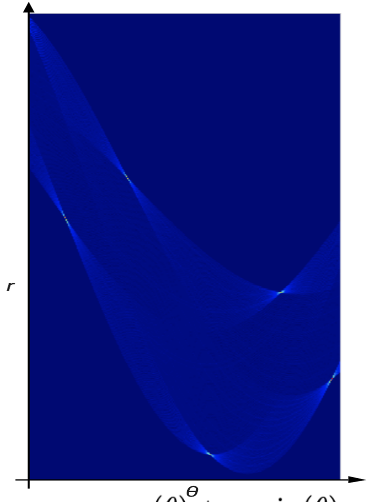
$$r = x \cos(\theta) + y \sin(\theta)$$

where,

$$\theta \in [-90^0 , +90^0]$$

$$r = [0, len(xy\text{-}space_{diagonal})]$$

Accumulator space for 5 straight lines

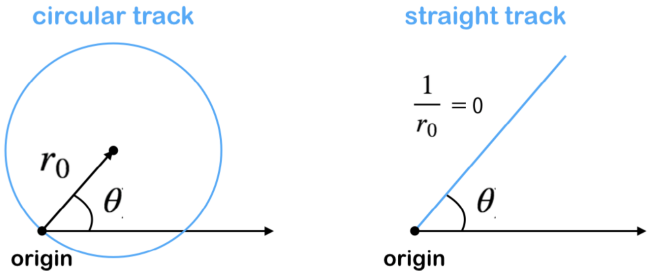


$r = x \cdot \cos(\theta) + y \cdot \sin(\theta)$

Hough Transform for Circular Tracks

Circle in polar (r, ϕ) coordinates with center (r_0, θ):

$$r = 2r_0 \cos(\phi - \theta)$$



Voting & Accumulator Space

- Construct Accumulator:
  Let each hit point in polar representation vote.
- Loop through each hit point.
  Voting: For each $\theta = [-90, 90]$ find p for each pixel.
- Each point generates a sinusoid in accumulator space while incremental voting at each bin in a matrix (accumulator matrix).

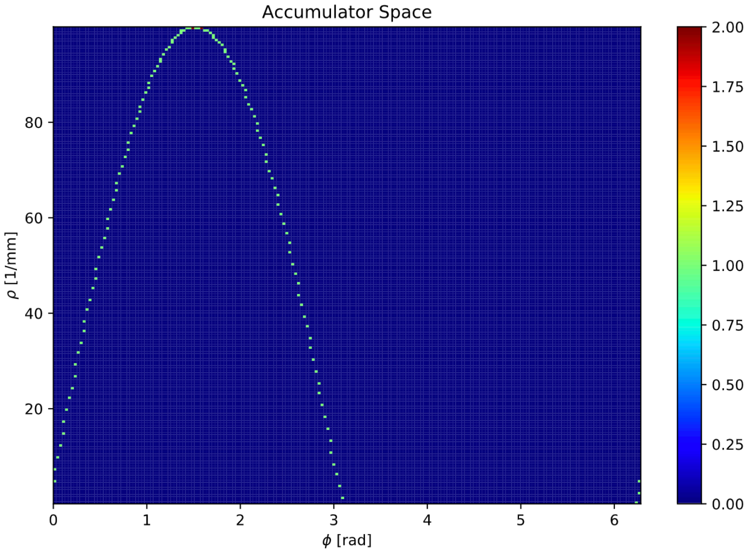
Hough Transform for a Hit:

Hit (r, ϕ)

Transform

$$\frac{1}{r_0} = \frac{2Cos(\phi - \theta)}{r}$$

One hit with coordinates (r, ϕ)



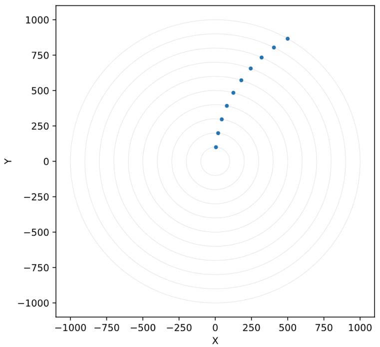
Ref.: *TrackML particle tracking challenge: https://www.kaggle.com/c/trackml-particle-identification/overview*

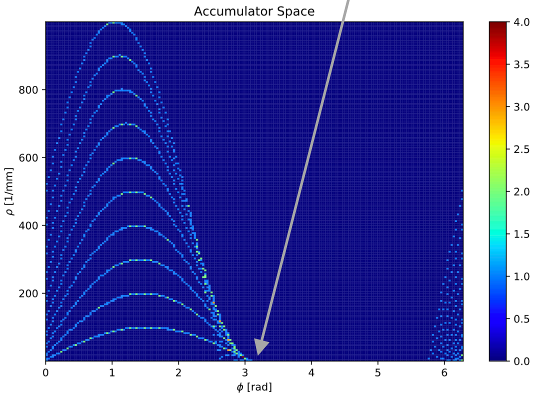
Hough Transform for Circular Tracks

On a Toy Model Detector

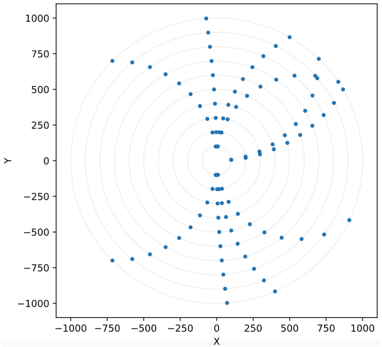
Tracks → $$\frac{1}{r_0} = \frac{2Cos(\phi - \theta)}{r}$$ → Recognized Track

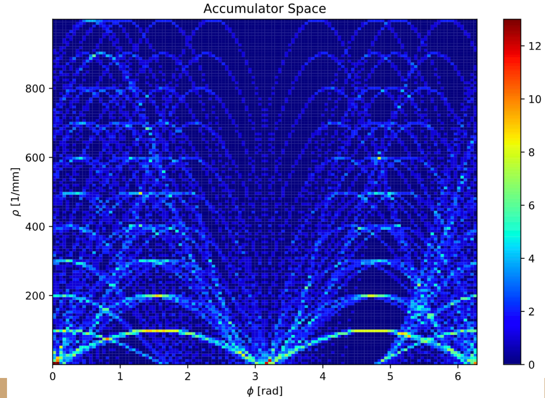
One Track





Event





Recognized Tracks
(local maxima / hot bins)

Quantum Hough Transform

Grover-Long Algorithm for Search

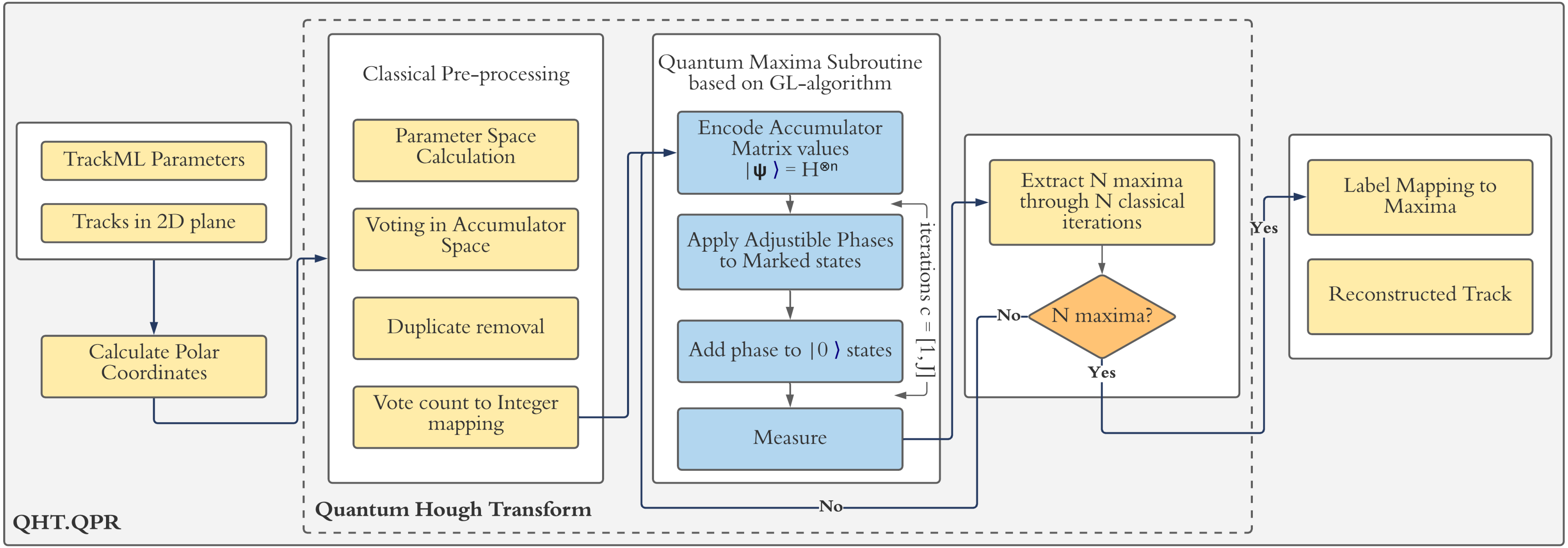
- Quantum Advantage in Quantum Hough Transform:
  - Runs on Circuit-Model NISQ[1]-devices (IBM-Q).
  - Speed: Speed Improvement O(N) to O(√N).
  - Memory: Accumulator Encoding can potentially provide exponential improvement.
  - Highly effective applications in n-dimensional Accumulator Spaces.

- Quantum Maxima Algorithm
  - Grover-Long (GL) Algorithm: Finite Search.
  - Close estimates of the ratio of the number of solutions M and the searched space N, renders success probability close to 100%.

[1] NISQ = Noisy Intermediate Scale Quantum devices based on Circuit Model QC such as, IBM-Q Hardware/Simulator.

Quantum Hough Transform

Grover-Long Algorithm for Search on a Toy Model Detector



Algorithm Flowchart

Quantum Hough Transform

GL Subroutine and Accumulator Mapping

- Initial State Preparation
  - Each data value must be a: distinct, positive, integer.
  - Each data value is represented by a binary string. Orthonormal basis states of $|\Psi\rangle$ store data value, where $|\Psi\rangle$ is the initial state. State preparation: $\log_2(N)$ steps.
  - For N $\in 2^n$ where n is #qubits. then, $|\Psi\rangle_{init.} = H^{\otimes n}$
  - One-to-one mapping between a data value and it's index.

- $GL = -WI_0W^{-1}O$
  - Next slide

- Quantum Mapping to vote database:
  - Discretize vote counts.
  - Remove Duplicates.
  - One-to-One mapping of label array with accumulator vote array.
  - Labels: Hits corresponding to hot bins.

Quantum Hough Transform

GL Subroutine

- $GL = -WI_0W^{-1}O$ *where, W, I_0 and O are unitary operators.*

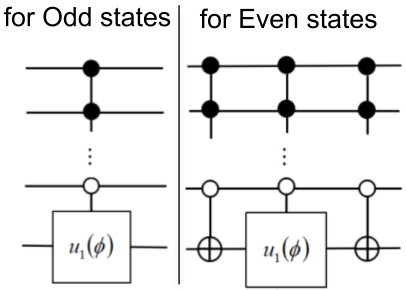
Create Superposition

$$|\Psi\rangle = W|0^{\otimes n}\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle$$

Add rotation phase to marked states $|v_\tau\rangle$ state.

$$O = e^{i\phi}|v\rangle\langle v| + \sum_{\tau=0,\tau\neq v}^{2^n-1}|\tau\rangle\langle\tau| \qquad O = e^{i\phi}\sum_{\tau=1}^{M}|v_\tau\rangle\langle v_\tau| + \sum_{\tau=0,\tau\notin V}^{2^n-1}|\tau\rangle\langle\tau|$$

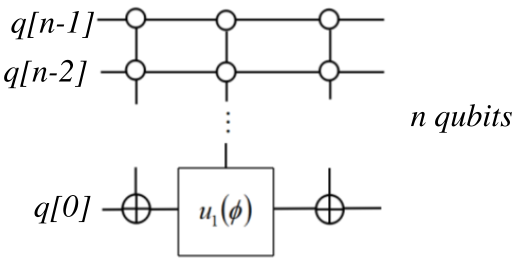
O operator



Add adjustable rotation phase to $|0\rangle$ state.

$$I_0 = e^{i\phi}|0\rangle\langle 0| + \sum_{\tau=1}^{2^{n-1}}|\tau\rangle\langle\tau| = \mathrm{diag}\left[e^{i\phi}, 1, \ldots, 1\right]_{2^n}$$

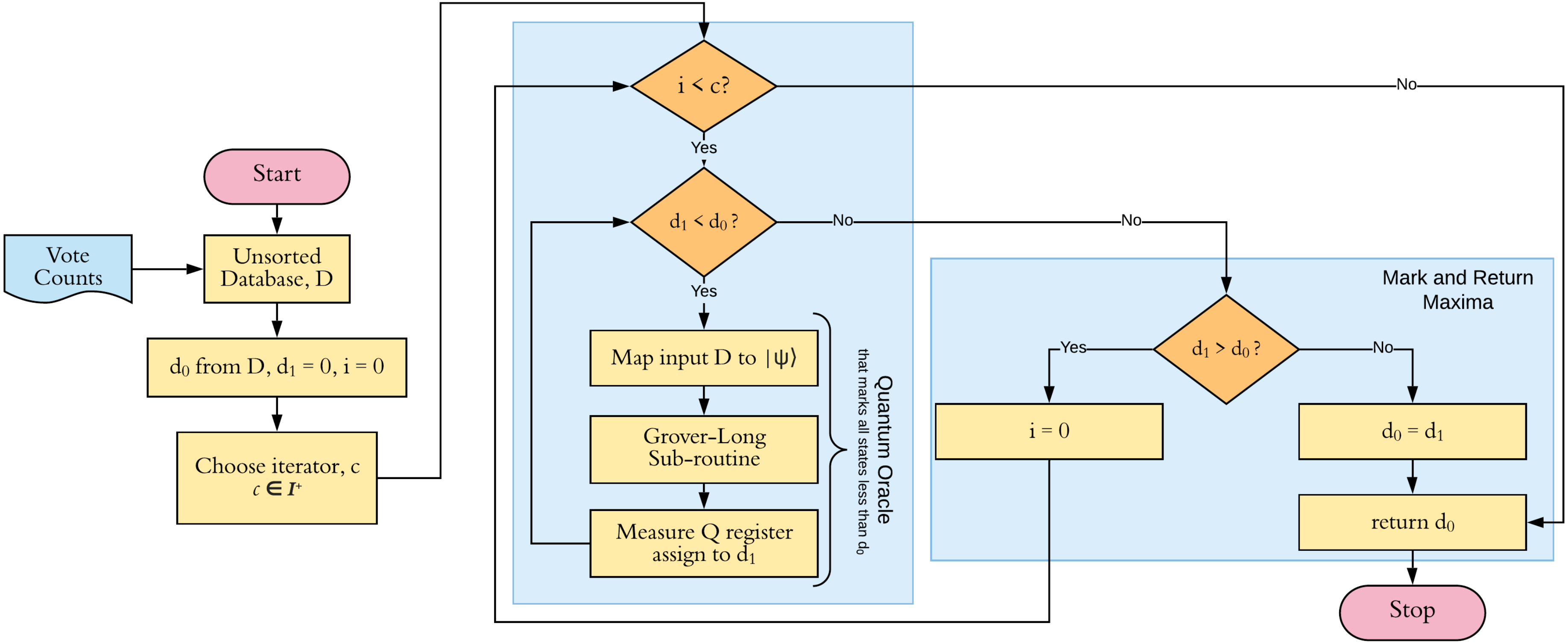
I_0 operator



$$\boxed{GL = G|\Psi\rangle\langle\Psi| = \left(-\left(e^{i\phi}-1\right)|\Psi\rangle\langle\Psi| - I\right)\left(e^{i\phi}|\Psi_{\text{good}}\rangle + |\Psi_{\text{bad}}\rangle\right)}$$

Quantum Hough Transform

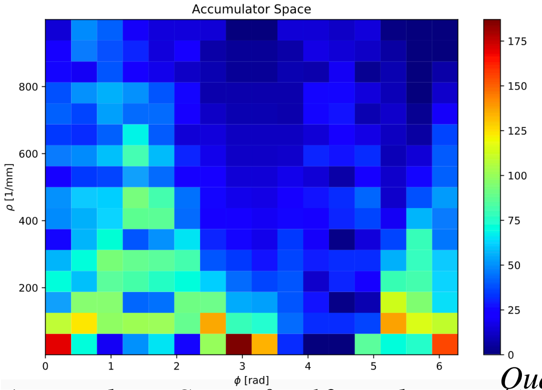
Quantum Maxima Finding Algorithm



[CQGW19] An Optimized Quantum Maximum or Minimum Searching Algorithm and its Circuits. https://arxiv.org/abs/1908.07943

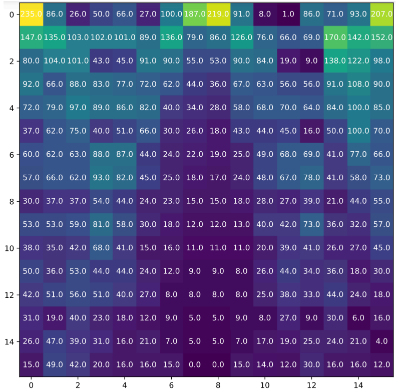
Quantum Maxima Finding

Results: Local Maxima Detection using Grover-Long Algorithm



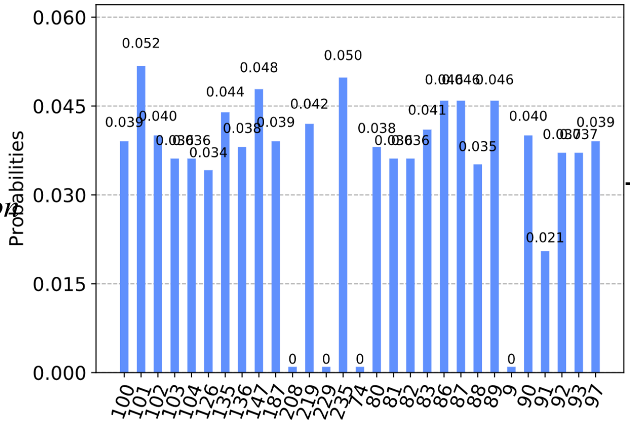
Accumulator Space for 10 tracks

Quantum Superposition



Accumulator Matrix for 10 tracks

*Probability of detecting all non-zero marked states**



8-qubit simulation

#tracks = 4, #bins up to 2^8 = 256, time = 58.6s

* Updates Iteratively upon each maxima extraction.

```
track_index

array([array([13, 19, 22, 28, 35, 48, 52]),
       array([ 2,  5, 16, 36, 40, 49]),
       array([23, 29, 30, 32, 43, 53, 54]),
       array([17, 18, 23, 30, 33, 39, 45, 50]),
       array([17, 18, 23, 29, 30, 33, 39, 50]),
       array([ 0,  9, 14, 34, 38, 42]), array([ 1, 11, 12, 20, 24, 27]),
       array([ 7,  8, 10, 37, 46, 57]), array([17, 23, 29, 30, 32, 54]),
       array([17, 29, 32, 43, 53, 54])], dtype=object)
```

*10 maxima values^*

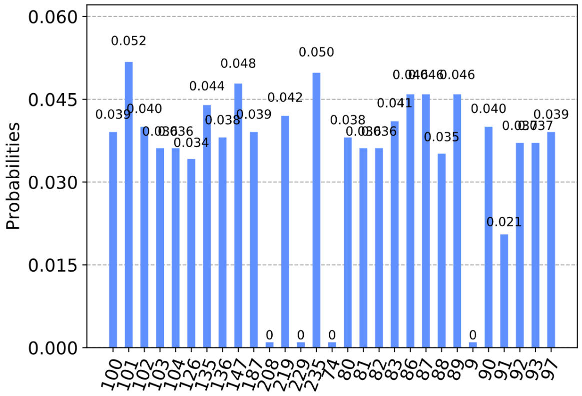
N iterations = N maxima where, N is provided as input.

^ detected through classical HT

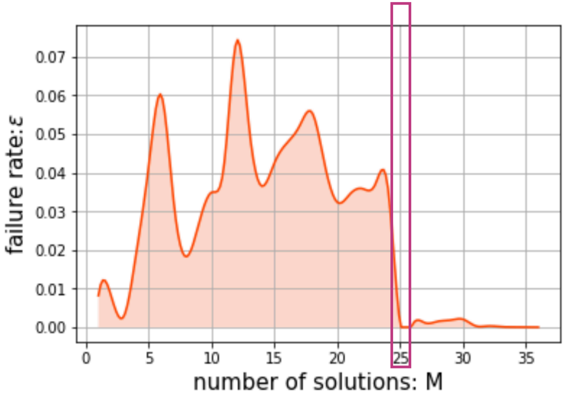
Quantum Maxima Finding

Results: Local Maxima Detection using Grover-Long Algorithm

- Failure Rate as number of Marked States:
  - Number of Grover-Long iterations can be fixed with iterator c.
  - Scalable: GL failure rate is less than 10%.
  - Can be improved with even higher number of marked states.



8-qubit simulation:
number of marked states = 25

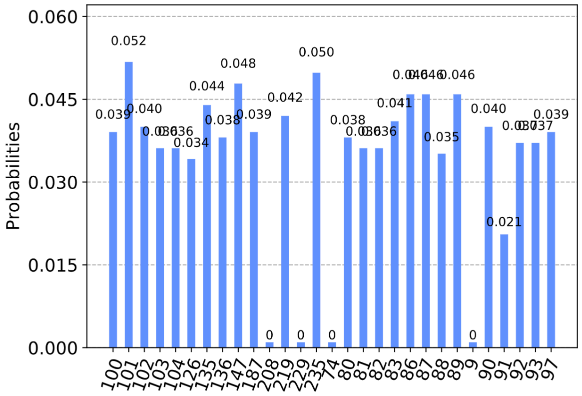


$$\textit{Theoretical Error Rate} = \varepsilon_{GL} = 1 - M \times \mathrm{abs}\left(\alpha_{\mathrm{good}}^{(\tilde{J})}\right)^2$$

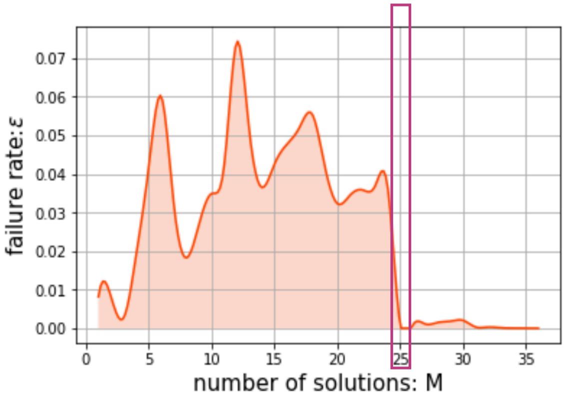
Quantum Maxima Finding

Results: Local Maxima Detection using Grover-Long Algorithm

- Failure Rate as number of Marked States:
  - Number of Grover-Long iterations can be fixed with iterator c.
  - Scalable: GL failure rate is less than 10%.
  - Can be improved with even higher number of marked states.



8-qubit simulation:
number of marked states = 25



$$\textit{Theoretical Error Rate} = \varepsilon_{GL} = 1 - M \times \text{abs}\left(\alpha_{\text{good}}^{(\tilde{J})}\right)^2$$

Confidence Level

E \ C	50%	75%	80%	85%	95%	99%	99.90%
0.01	1140	3307	4096	5184	9604	16590	19741
0.03	127	358	456	576	1068	1844	2194
0.05	46	133	164	208	385	664	790
0.1	12	34	41	52	97	166	198
0.15	6	15	19	24	43	74	88
0.2	3	9	11	13	25	42	50

Acceptable Error

Classical Recons. Eff. — 16-20 qubits

Current QHT — 8 qubits

Minimum Sample Size

[CQGW19] An Optimized Quantum Maximum or Minimum Searching Algorithm and its Circuits. https://arxiv.org/abs/1908.07943

Conclusion

DONE:

- Quantum Algorithm for Maxima/Minima finding {$O(\sqrt{N})$ + $O(\log_2(N))$ for state preparation} + {classical processing}.
- Only Maxima subroutine implemented. Uses one-to-one digitized mapping.
- Returns a 2D array each row containing corresponding hits of the identified tracks.
- Correctness - Initial analysis of failure rate.

TO DO:

- Get the label output data of identified tracks and plot.
-

Metrics	Typical Value
Reconstruction Efficiency ($N_{recog\_tracks}/N_{tracks}$)	~95 — 99%
Fake Rate ($N_{unwanted\_tracks}/N_{tracks}$)	1% — 10% (poor)
Duplicate Rate ($N_{duplicate\_tracks}/N_{tracks}$)	negligible

IMPROVEMENTS:

- Labeling is done classically when the accumulator space is constructed i.e. during voting.
- Classical Accumulator space construction does not render speed up in terms of memory complexity.
  - Start from scratch, create superposition hits and of $(1/r_0, \theta)$ space [V13 for n-dim].
  - Construct accumulator space superposition in parts
    - needs separation of both: hits and respective labels
    - can be based on detector geometry.

[CQGW19] An Optimized Quantum Maximum or Minimum Searching Algorithm and its Circuits. https://arxiv.org/abs/1908.07943

[V13] K. M. Varadarajan, Quantum Hough Transform, GraphiCon'2013 (2013)

fin.

Extras: Quantum Maxima Finding using GL Algorithm

- GL subroutine: $GL = -WI_0W^{-1}O$: *where, W, I_0 and O are unitary operators.*

Step 1: apply phase to marked states: O operator

$$O|\Psi\rangle = \begin{bmatrix} e^{i\phi} & \\ & 1 \end{bmatrix} \left(|\Psi_{\text{good}}\rangle + |\Psi_{\text{bad}}\rangle\right)$$

$$\sin\beta = \sqrt{\frac{M}{N}}.$$

$$J \geq \text{floor}\left(\frac{\frac{\pi}{2} - \beta}{\beta}\right) + 1$$

Step 2: $-WI_0W^{-1}$ operator

$$-W\left(\left(e^{i\phi} - 1\right)|0\rangle\langle 0| - I\right)W^{-1} = -\left(e^{i\phi} - 1\right)|\Psi\rangle\langle\Psi| - I$$

$$\phi = 2\arcsin\left(\frac{\sin\frac{\pi}{4J+2}}{\sin\beta}\right)$$

Step x: GL operator overall, can be expressed as follows:

$$G|\Psi\rangle\langle\Psi| = \left(-\left(e^{i\phi} - 1\right)|\Psi\rangle\langle\Psi| - I\right)\left(e^{i\phi}|\Psi_{\text{good}}\rangle + |\Psi_{\text{bad}}\rangle\right)$$

Good and Bad states i.e. Marked and Unmarked states are represented as:

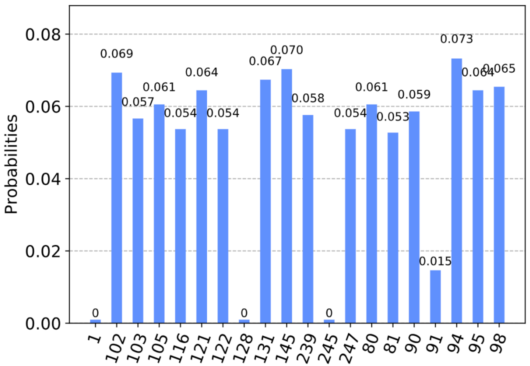
$$\alpha_{\text{good}}^{(j)} = -\alpha_{\text{good}}^{(j-1)} \times \left(1 + \frac{e^{i\tilde{\phi}} - 1}{N}\right) - \alpha_{\text{good}}^{(j-1)}(M-1) \times \frac{e^{i\tilde{\phi}} - 1}{N} - \alpha_{\text{bad}}^{(j-1)} \times (N-M) \times \frac{e^{i\tilde{\phi}} - 1}{N}$$

$$\alpha_{\text{bad}}^{(j)} = -\alpha_{\text{bad}}^{(j-1)} \times \left(1 + \frac{e^{i\tilde{\phi}} - 1}{N}\right) - \alpha_{\text{good}}^{(j-1)} \times M \times \frac{e^{i\tilde{\phi}} - 1}{N} - \alpha_{\text{bad}}^{(j-1)} \times (N-M-1) \times \frac{e^{i\tilde{\phi}} - 1}{N}$$

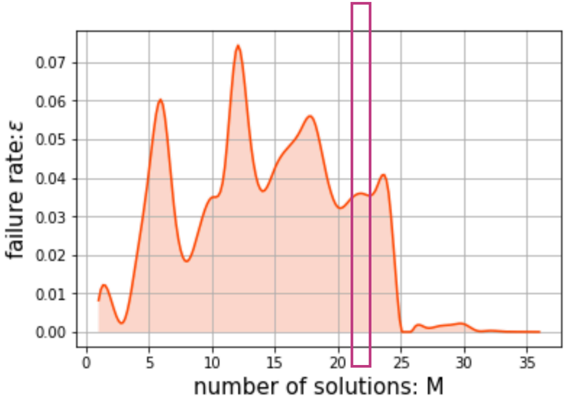
Quantum Maxima Finding

Results: Local Maxima Detection using Grover-Long Algorithm

- Failure Rate as number of Marked States = 17



8-qubit simulation:
number of marked states = 17



Theoretical Error Rate = $\varepsilon_{GL} = 1 - M \times \mathrm{abs}\left(\alpha_{\text{good}}^{(\tilde{J})}\right)^2$

Extras: Quantum Encoding for Accumulator Matrix

- Quantum embeddings for machine learning [2020]
https://arxiv.org/abs/2001.03622, https://arxiv.org/abs/1803.07128, https://arxiv.org/abs/1804.11326

-A divide-and-conquer algorithm for quantum state preparation [2020]
https://arxiv.org/abs/2008.01511

Extras: Data Fitting

Discrete Fourier transform for Pattern Classification [2014]:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1018.1299&rep=rep1&type=pdf
- uses classical Fourier Transform to implement nearest-neighbor based classifier.

Quantum Support Vector Machine for classification [2013]:
https://arxiv.org/abs/1307.0471

Quantum Data Fitting [2012]
https://arxiv.org/abs/1204.5242

Modern Quantum Algorithmic Landscape

for Gate Model QC (...lots to Explore!)

- **Simulation**
  - Variational Quantum Eigensolver (VQEs) and α-VQEs
  - QITE/QLanczos
  - Thermal Averages: METTS Algorithm, Quantum Importance Sampling
  - Unitary Quantum Dynamics,
  - Field Theory Simulation, HEP Simulations etc.

- **Scientific Computing/Mathematical Methods**
  Finite Element Methods, Partial Differential Equations, Poisson Solver
  - Quantum Linear System Algorithms/HHL
  - Quantum Support Vector Machine
  - Quantum Phase Estimation
  - Quantum Fourier Transform

- **Quantum Machine Learning**
  - Bayesian Deep Learning
  - Quantum Tensor Networks
  - Quantum Neural Networks
  - Quantum Principal Component Analysis

- **Quantum Graph Theory/Other Approaches**
  - Quantum Random Walks
  - Quantum Monte Carlo
  - Quantum Probably Approximately Correct
  - Grover's Search

- **Optimization**
  - Quantum Alternating Ansatz Algorithms (QAAA)
    - QAOA and Hamiltonian-based QAOA
  - Quantum Annealing (on D-Wave)

- **Quantum-Inspired Classical Algorithms**
  - Recommendation Systems