

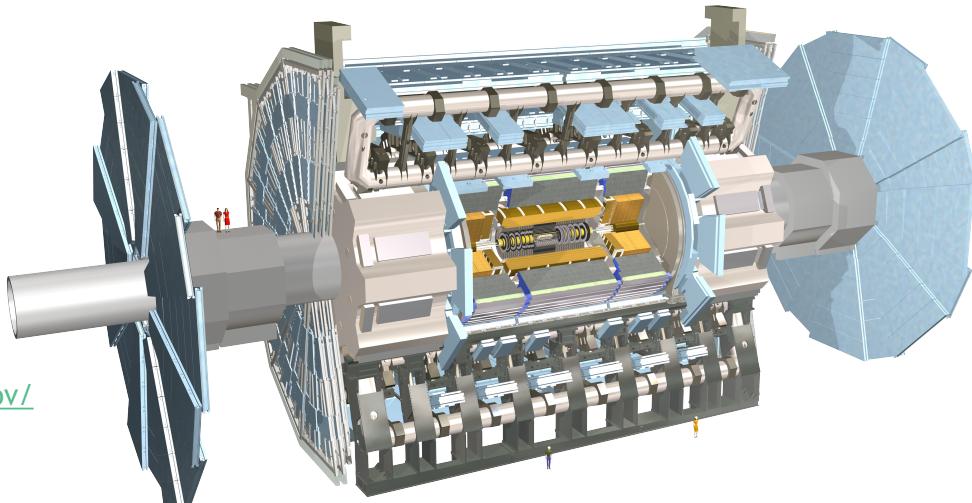
Quantum Hough Transform

Algorithm for Charged-Particle Track Reconstruction

08-05-2020

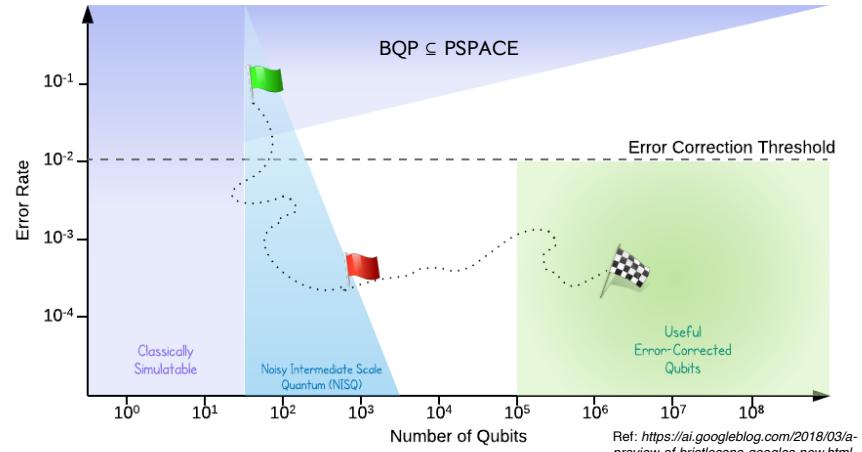
Amitabh Yadav
Heather Gray

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



Introduction

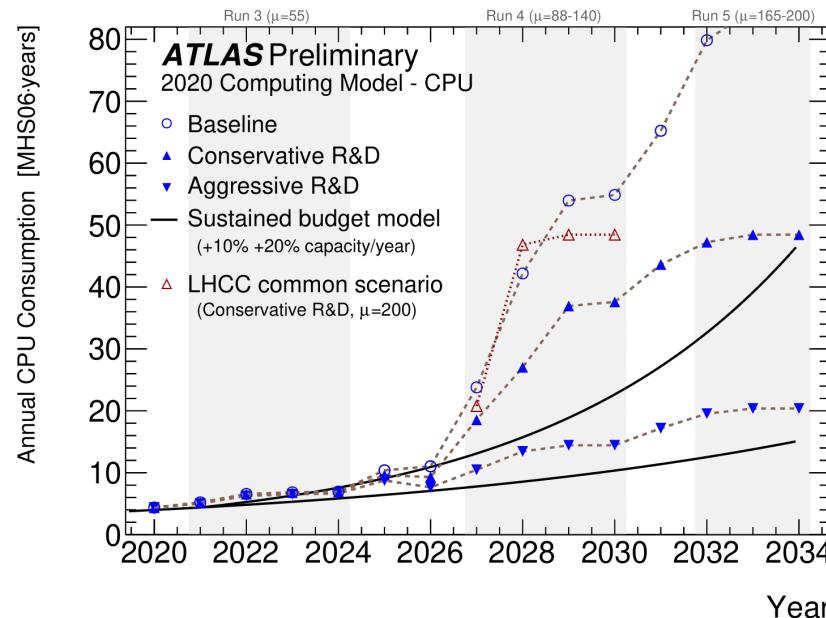
- **Circuit Model QC**
 - Universal Quantum Computing
 - Solovay-Kitaev Theorem^[KY97]
- **Quantum Computing Applications Approach**
 - Perfect Qubits
 - Erroneous Qubits in NISQ-era
 - Test algorithmic bounds of error-tolerance,
 - Pauli Frames,
 - Temporary Quantum Registers and SWAP gates,
 - Small (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)



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.

1
3,069,654
METHOD AND MEANS FOR RECOGNIZING
Paul V. C. Hough, Ann Arbor, Mich., assignee to the
United States of America as represented by the United
States Patent Office, Washington, D. C.
Filed Mar. 25, 1968, Ser. No. 17,715
U. S. Cl. 346—142
This invention relates to the detection 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 and smaller, the problems involved become
increasingly more complex. One of the more useful de-
vices in observing charged particles is the bubble chamber
which consists of a tank containing a liquid which is
composed of small bubbles approximately 0.01
inch apart, depending upon the specific ionization of the
incident particles. These tracks form straight lines in
and are readily photographed with the use of a dark back-
ground. With this device, multidimensional photographs
are produced with each photograph requiring several hours
of the point on the line segment from the horizontal mid-
line 109 of the frameline 108.
(3) Each line in the transformed plane is made to have
an intercept with the horizontal midline 101 of the pic-
ture point on the line segment in frameline 108.
Thus, for a given reference point 110 on line segment
102, the reference point 110 is at a distance from midline
101 of the picture point 110 is at a distance from midline
101 of the picture 110 is at a distance from the hori-
zontal midline 109 of the frameline 108.
It is an exact theorem that, if a series of points in a
framed picture 108 is such that all the corresponding lines in
the plane transform intersect in a point which will be
designated as a knot 112. It is therefore readily apparent
that the intersection of the lines 104 and 106 in frameline
108 have the following properties:
(1) The horizontal coordinates of the knot 112 equal
the horizontal coordinates of the frameline 108 at which
the straight line segments 102, 104 and 106 intercept
the horizontal midline 109 of the frameline 108.
(2) The knot 112, relative
picture 108, is propor-
tional to the straight line seg-
ments 102, 104 and 106 in
the plane transforms
dopes and intercepts of
34 and 106 in frameline
108.

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 the merits of this method 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, collinear points, point-line.

CR Category: Printed in Great Britain

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 co-linear or almost co-linear 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.

0031-3203/81/020111-12\$02.00/0
© Pattern Recognition Society
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 the curve. The method was shown how to detect both analytic curves^{1,2} and nonanalytic curves^{3,4} of these methods were generalized to the detection of some analytic curves in gray level images, specifically lines,^{5,6} circles^{5,6} and ellipses.⁶ The line detection case is the best known of these and has been ingeniously exploited in several applications.^{7,8}

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**

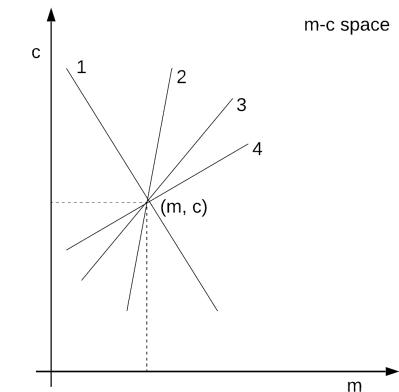
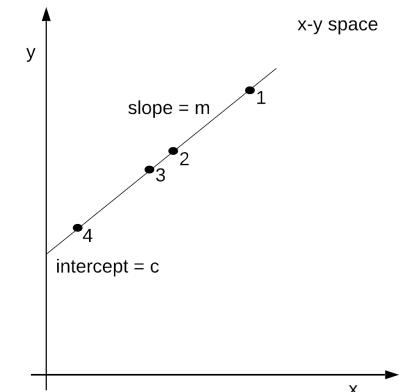
- A hit in xy -space is represented as a line in mc -space.
- Multiple hits renders multiple lines in parameter space.
- The intersection (m , c) represents the line connecting the hits.

- **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.



Hough Transform for Linear Track

- Vertical lines: $m, c = \text{unbounded}$.
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:

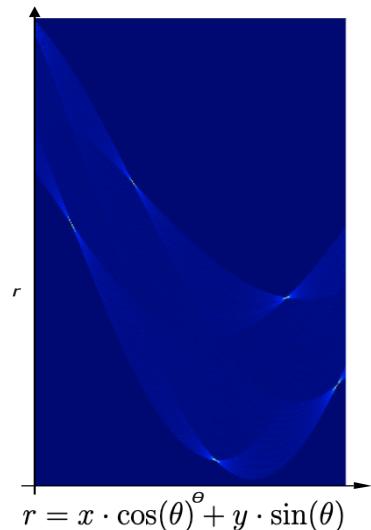
$$r = x \cos(\theta) + y \sin(\theta)$$

where,

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

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

Accumulator space
for 5 straight lines



Hough Transform for Circular Tracks

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

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

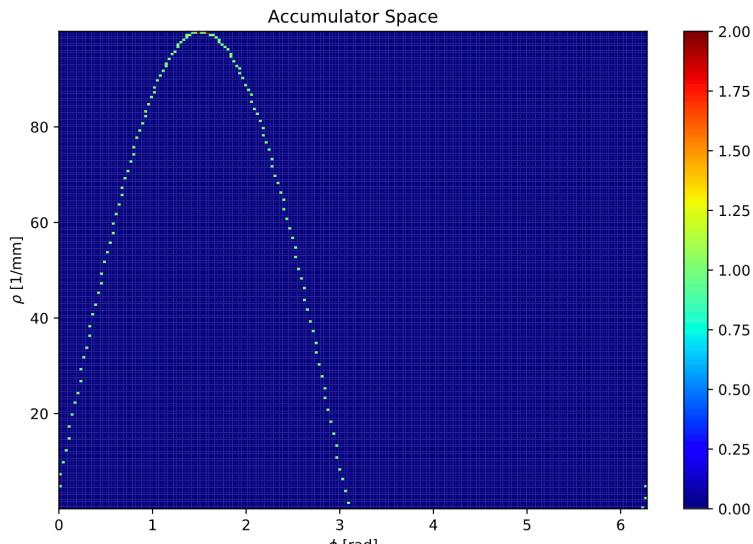
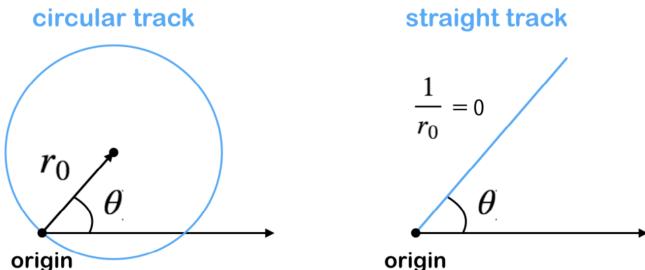
Voting & Accumulator Space

- 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:

$$\text{Hit } (r, \phi) \xrightarrow{\text{Transform}} \frac{1}{r_0} = \frac{2\cos(\phi - \theta)}{r}$$

One hit with coordinates (r, ϕ)

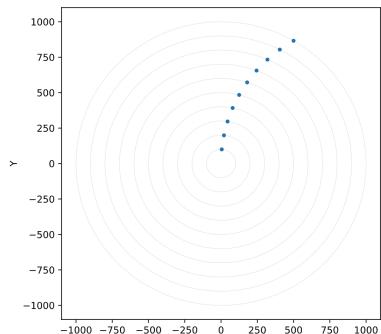


Hough Transform for Circular Tracks

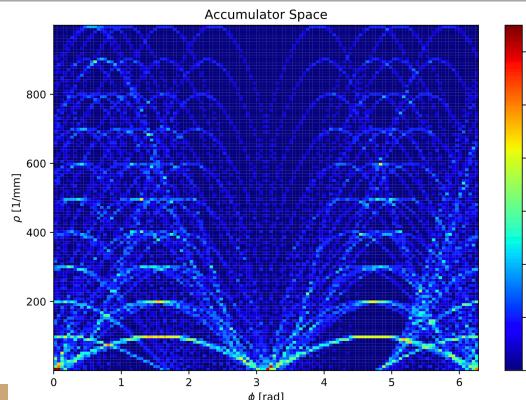
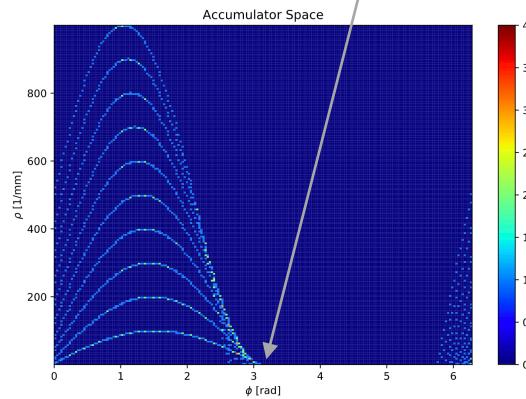
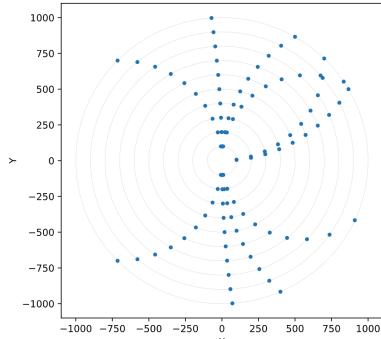
On a Toy Model Detector

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

One Track



Event



Quantum Hough Transform

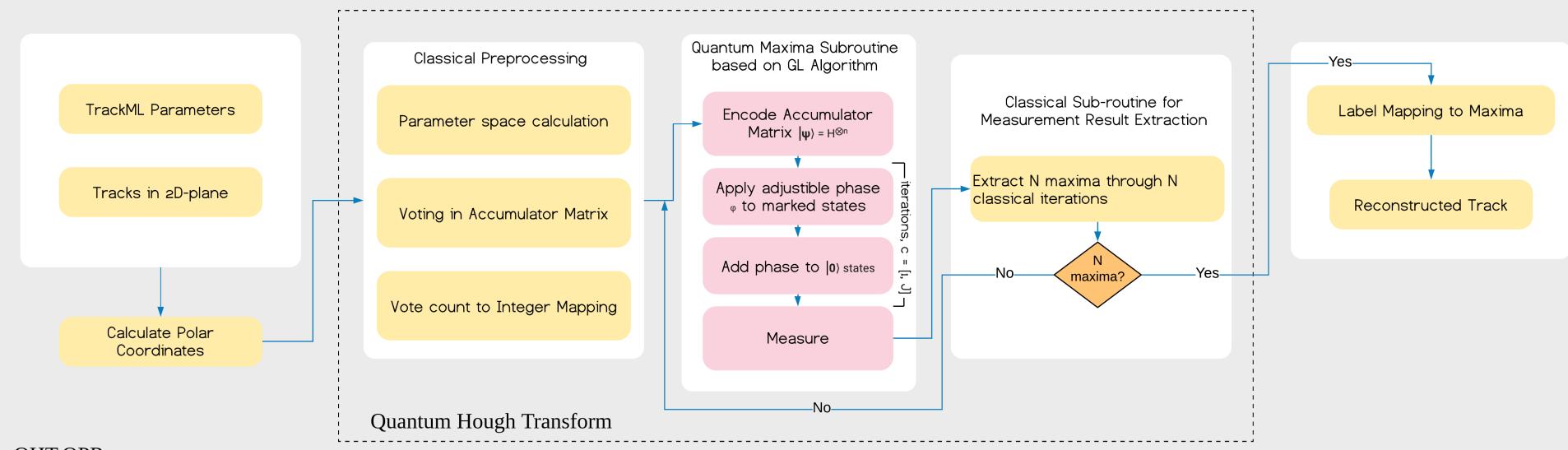
Grover-Long Algorithm for Search

- Quantum Advantage in Quantum Hough Transform:
 - Runs on Circuit-Model NISQ^[1]-devices (IBM-Q).
 - Speed: Speed Improvement $O(N)$ to $O(\sqrt{N})$.
 - 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

QHT.QPR

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, $|\Psi\rangle_{\text{init.}} = H^{\otimes n}$
 - One-to-one mapping between a data value and it's index.
- **GL = -WI₀W⁻¹O**
 - 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.

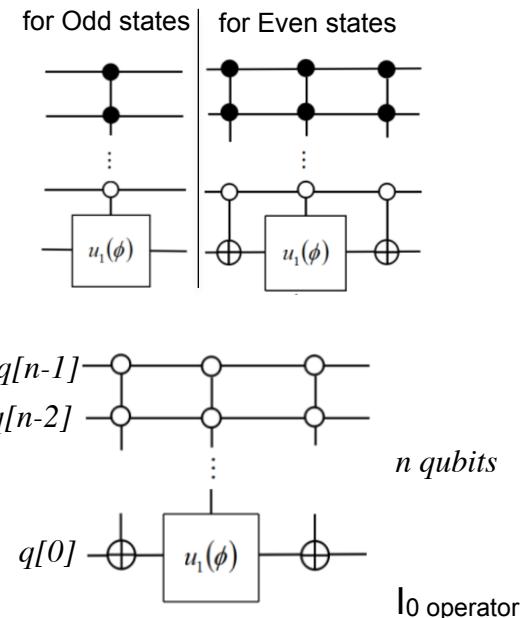
$$O = e^{i\phi}|v\rangle\langle v| + \sum_{\tau=0, \tau \neq v}^{2^n-1} |\tau\rangle\langle \tau|$$

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

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

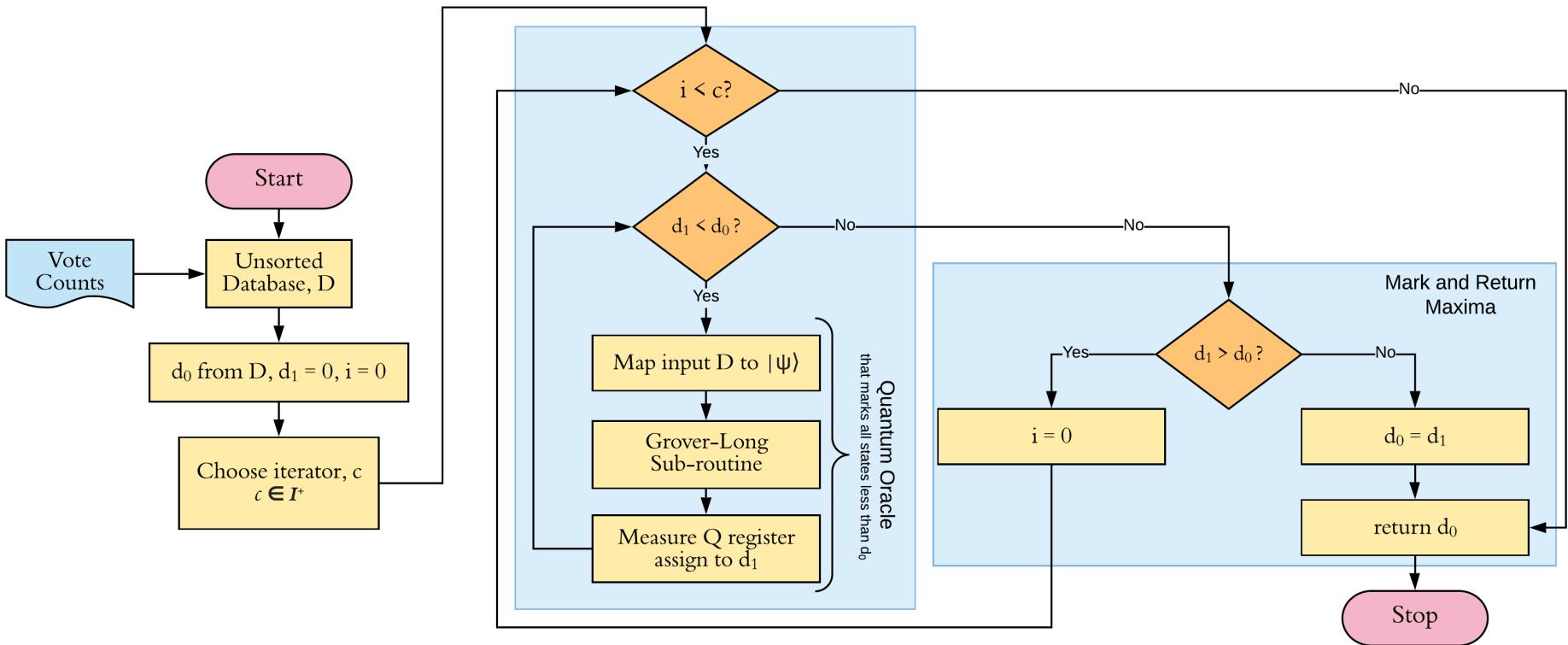
$$I_0 = e^{i\phi}|0\rangle\langle 0| + \sum_{\tau=1}^{2^n-1} |\tau\rangle\langle \tau| = \text{diag}[e^{i\phi}, 1, \dots, 1]_{2^n}$$

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



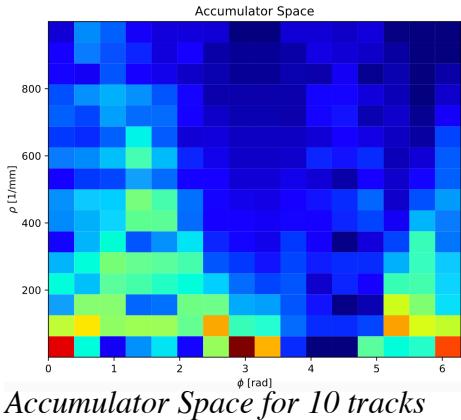
Quantum Hough Transform

Quantum Maxima Finding Algorithm



Quantum Maxima Finding

Results: Local Maxima Detection using Grover-Long Algorithm

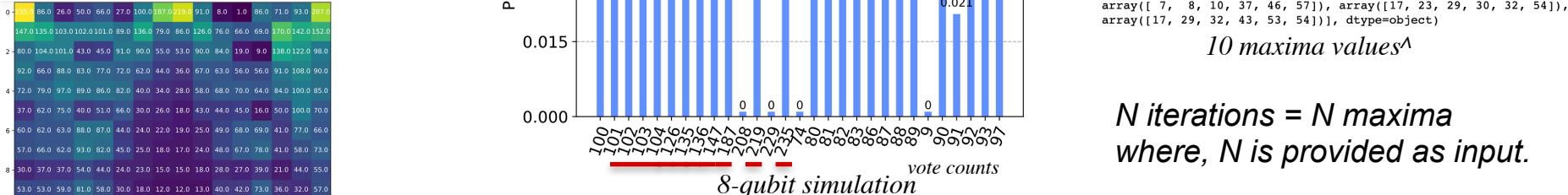


Accumulator Space for 10 tracks

$\xrightarrow{\text{Quantum Superposition}}$

0	235.0	86.0	26.0	50.0	66.0	27.0	100.0	87.0	219.0	91.0	8.0	1.0	86.0	71.0	93.0	20.0	
1	147.0	135.0	103.0	102.0	0.10.0	89.0	136.0	79.0	86.0	126.0	76.0	66.0	69.0	170.0	42.0	152.0	
2	80.0	80.0	104.0	101.0	43.0	45.0	91.0	90.0	55.0	53.0	90.0	84.0	19.0	1.0	38.0	122.0	98.0
3	92.0	66.0	88.0	83.0	77.0	72.0	62.0	44.0	36.0	67.0	63.0	55.0	56.0	91.0	108.0	90.0	
4	72.0	79.0	97.0	89.0	86.0	82.0	40.0	34.0	28.0	58.0	68.0	7.0	64.0	84.0	100.0	85.0	
5	37.0	62.0	75.0	40.0	51.0	66.0	30.0	26.0	18.0	43.0	44.0	16.0	50.0	100.0	70.0	70.0	
6	6.0	60.0	62.0	63.0	88.0	87.0	44.0	24.0	22.0	19.0	25.0	49.0	68.0	69.0	41.0	77.0	66.0
7	57.0	66.0	62.0	93.0	82.0	45.0	25.0	18.0	17.0	24.0	48.0	67.0	78.0	41.0	58.0	73.0	
8	30.0	37.0	37.0	54.0	44.0	24.0	23.0	15.0	15.0	18.0	28.0	27.0	39.0	21.0	44.0	55.0	
9	53.0	53.0	59.0	81.0	58.0	30.0	18.0	12.0	12.0	13.0	40.0	42.0	73.0	36.0	32.0	57.0	
10	38.0	35.0	42.0	68.0	41.0	15.0	16.0	11.0	11.0	11.0	20.0	39.0	41.0	26.0	27.0	45.0	
11	50.0	36.0	53.0	44.0	44.0	24.0	12.0	9.0	9.0	8.0	26.0	44.0	34.0	36.0	18.0	30.0	
12	42.0	51.0	56.0	51.0	40.0	27.0	8.0	8.0	8.0	25.0	38.0	33.0	44.0	24.0	18.0	18.0	
13	31.0	19.0	40.0	23.0	18.0	12.0	9.0	5.0	5.0	9.0	8.0	27.0	9.0	30.0	6.0	16.0	
14	26.0	47.0	39.0	31.0	16.0	21.0	7.0	5.0	5.0	7.0	17.0	19.0	25.0	24.0	21.0	4.0	
15	15.0	49.0	42.0	20.0	16.0	16.0	15.0	0.0	0.0	15.0	14.0	12.0	30.0	16.0	16.0	12.0	

Accumulator Matrix for 10 tracks



10 maxima values^

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

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

* Updates Iteratively upon each maxima extraction.

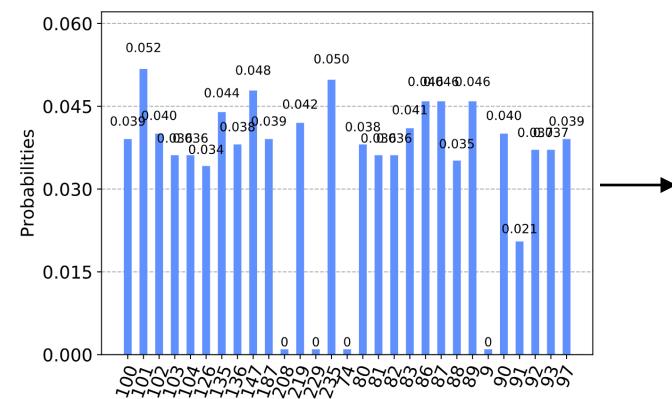
^ detected through classical HT

Quantum Maxima Finding

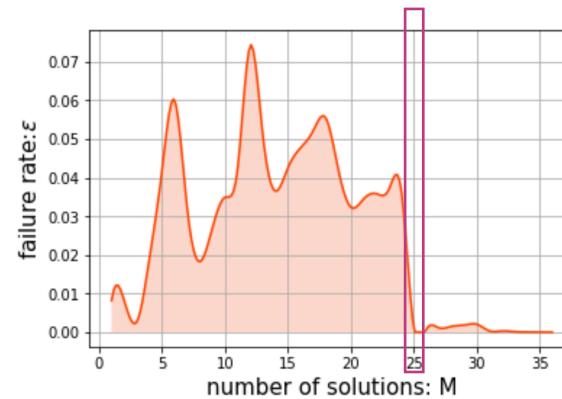
Results: Local Maxima Detection using Grover-Long Algorithm

- 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



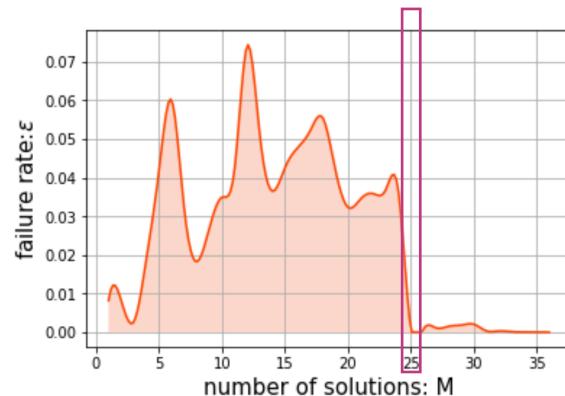
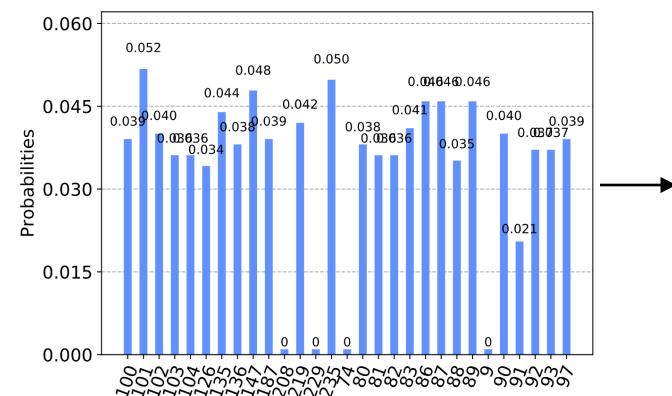
$$\text{Theoretical Error Rate} = \varepsilon_{GL} = 1 - M \times \left| \alpha_{\text{good}}^{(j)} \right|^2$$

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

$$\text{Theoretical Error Rate} = \epsilon_{GL} = 1 - M \times \text{abs} \left(\alpha_{\text{good}}^{(j)} \right)^2$$

C	E	Confidence Level						
		50%	75%	80%	85%	95%	99%	99.90%
0.01	1140	3307	4096	5184	9604	16590	19741	16-20 qubits
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	Current QHT 8 qubits
0.2	3	9	11	13	25	42	50	

Minimum Sample Size

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 ($N_{\text{recog_tracks}}/N_{\text{tracks}}$)	~95 – 99%
Fake Rate ($N_{\text{unwanted_tracks}}/N_{\text{tracks}}$)	1% – 10% (poor)
Duplicate Rate ($N_{\text{duplicate_tracks}}/N_{\text{tracks}}$)	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

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

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

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

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

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

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

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

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

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

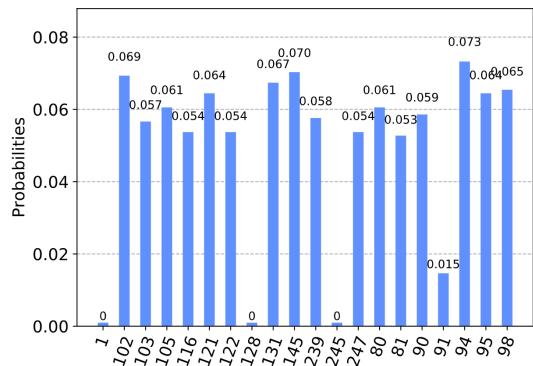
$$J \geq \text{floor}\left(\frac{\frac{\pi}{2} - \beta}{\beta}\right) + 1$$

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

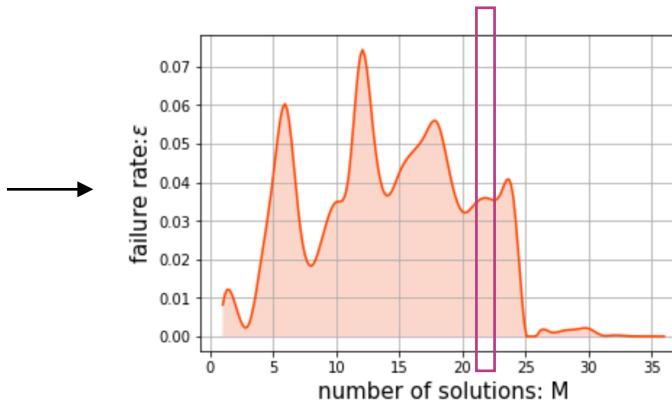
Quantum Maxima Finding

Results: Local Maxima Detection using Grover-Long Algorithm

- Failure Rate as number of Marked States = 17



8-qubit simulation:
number of marked states = 17



$$\text{Theoretical Error Rate} = \epsilon_{GL} = 1 - M \times \left| \alpha_{\text{good}}^{(j)} \right|^2$$

Extras: Data Fitting for Tracking

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>

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>

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