Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3075253 C 2023/08/29

(11)(21) 3 075 253

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

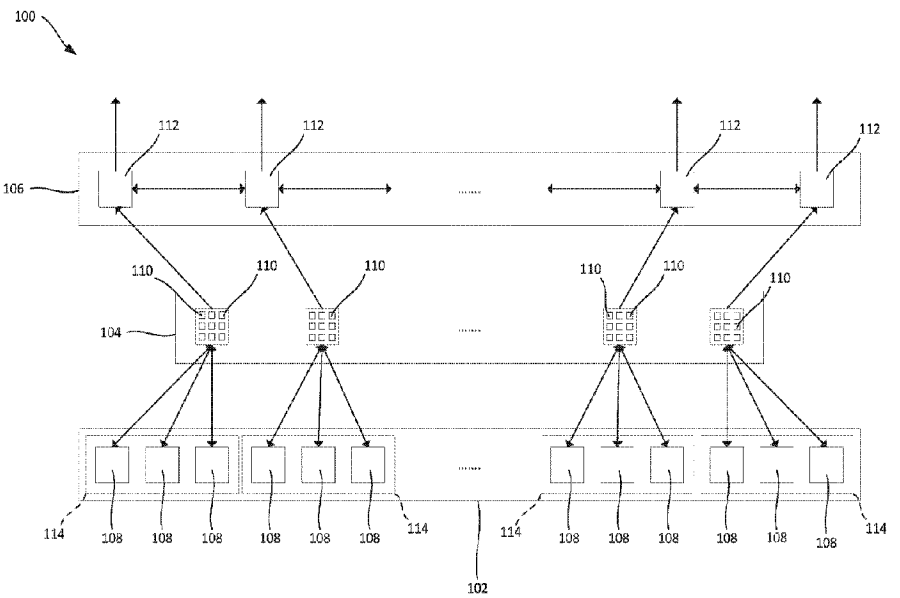
(86) Date de dépôt PCT/PCT Filing Date: 2017/09/12
(87) Date publication PCT/PCT Publication Date: 2019/03/21
(45) Date de délivrance/Issue Date: 2023/08/29
(85) Entrée phase nationale/National Entry: 2020/03/06
(86) N° demande PCT/PCT Application No.: US 2017/051193
(87) N° publication PCT/PCT Publication No.: 2019/054990

(51) Cl.Int./Int.Cl. *G06N 99/00* (2019.01)
(72) Inventeur/Inventor:
FOWLER, AUSTIN GREIG, US

(73) Propriétaire/Owner:
GOOGLE LLC, US

(74) Agent: SMART & BIGGAR LP

(54) Titre : CORRECTION D'ERREUR QUANTIQUE
(54) Title: QUANTUM ERROR CORRECTION



(57) Abrégé/Abstract:
A quantum error correction method including correcting a stream of syndrome measurements produced by a quantum computer comprises receiving a layered representation of error propagation through quantum error detection circuits, the layered representation comprises a plurality of line circuit layers that each represent a probability of local detection events in a quantum computer associated with one or more potential error processes in the execution of a quantum algorithm; during execution of the quantum algorithm: receiving one or more syndrome measurements from quantum error detection circuits; converting the syndrome measurements into detection events written to an array that represents a patch of quantum error correction circuits at a sequence of steps in the quantum algorithm; determining one or more errors in the execution of the quantum algorithm from the detection events in dependence upon the stored line circuit layers; and causing correction of the syndrome measurements based on the determined errors.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 March 2019 (21.03.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/054990 A1

(51) International Patent Classification:
G06N 99/00 (2010.01)

(21) International Application Number:
PCT/US2017/051193

(22) International Filing Date:
12 September 2017 (12.09.2017)

(25) Filing Language: English

(26) Publication Language: English

(72) Inventor; and
(71) Applicant: **FOWLER, Austin Greig** [AU/US]; 19010 Archwood St Unit 5, Reseda, California 91335 (US).

(74) Agent: **FRANZ, Paul E.**; Fish & Richardson P.C., P.O. Box 1022, Minneapolis, Minnesota 55440-1022 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,

(54) Title: QUANTUM ERROR CORRECTION

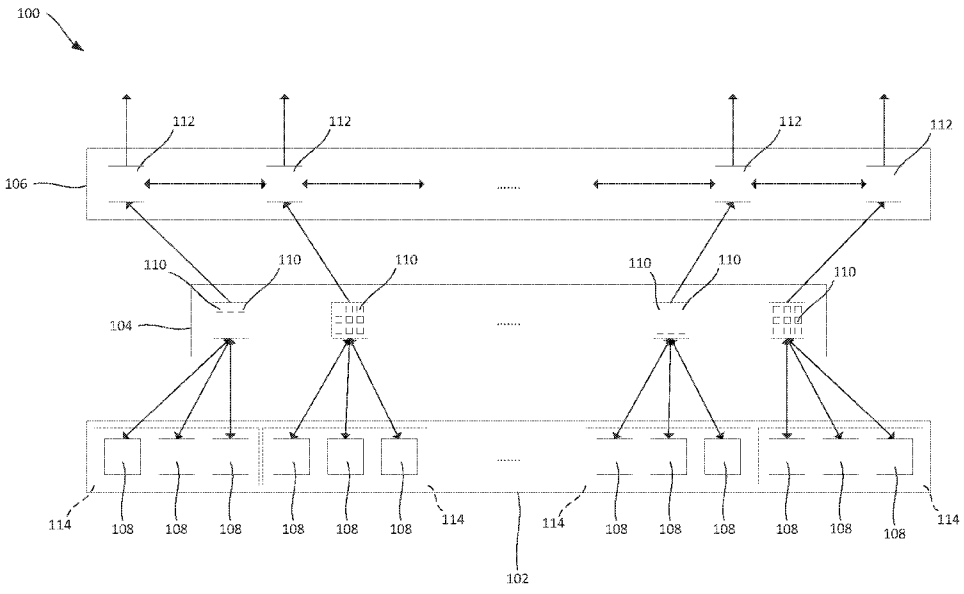


Fig. 1

(57) Abstract: A quantum error correction method including correcting a stream of syndrome measurements produced by a quantum computer comprises receiving a layered representation of error propagation through quantum error detection circuits, the layered representation comprises a plurality of line circuit layers that each represent a probability of local detection events in a quantum computer associated with one or more potential error processes in the execution of a quantum algorithm; during execution of the quantum algorithm: receiving one or more syndrome measurements from quantum error detection circuits; converting the syndrome measurements into detection events written to an array that represents a patch of quantum error correction circuits at a sequence of steps in the quantum algorithm; determining one or more errors in the execution of the quantum algorithm from the detection events in dependence upon the stored line circuit layers; and causing correction of the syndrome measurements based on the determined errors.

WO 2019/054990 A1

[Continued on next page]

WO 2019/054990 A1

MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

Quantum Error Correction

Field

The present disclosure relates to a method of quantum error correction. In particular, the present disclosure relates to methods of determining and correcting errors in an array of qubits executing a quantum algorithm in a quantum computer.

Background

Quantum computers are computing devices that exploit quantum superposition and entanglement to solve certain types of problem faster than a classical computer. The building blocks of a quantum computer are qubits. Qubits are two-level systems whose state can be in a superposition of its two states, rather than just in either of the two states as is the case for a classical bit.

Quantum algorithms are algorithms executed on quantum computers. During the execution of these algorithms on a quantum computer, errors can be introduced from a number of sources including decoherence and noise. Due to the so-called "no-cloning theorem", classical error detection and correction techniques involving creating multiple copies of a state are unsuitable. Instead, quantum error detection and correction techniques involve entangling qubits with a number of other qubits, and performing measurements on a subset of the entangled qubits in order to identify when an error has occurred.

Summary

According to a first aspect, there is described a method of correcting a stream of syndrome measurements produced by a quantum computer, the method comprising: receiving a layered representation of error propagation through quantum error detection circuits in the quantum computer, wherein the layered representation comprises a plurality of line circuit layers, each line circuit layer representing a probability of local detection events in the quantum computer associated with one or more potential error processes in the execution of the quantum algorithm; and during execution of the quantum algorithm: receiving one or more syndrome measurements from quantum error detection circuits in the quantum computer; converting the syndrome measurements into detection events written to an array, the array representing a patch of quantum error correction circuits at a sequence of steps in the

quantum algorithm; determining one or more errors in the execution of the quantum algorithm from the detection events in dependence upon the stored line circuit layers; and causing correction of the syndrome measurements based on the determined errors.

The method may further comprise constructing the layered representation of error propagation through quantum error correction circuits prior to the execution of the quantum algorithm.

Constructing a layered representation of error propagation through the quantum error detection circuits may comprise: for one or more quantum gates in the quantum circuit, determining one or more potential detection events associated with each potential error process occurring at that quantum gate; associating one or more lines with each potential error process, the lines each connecting a potential detection event associated with the potential error process to another potential detection event associated with the same potential error process or a boundary of the quantum circuit; and merging similar lines to form one or more merged lines, wherein the plurality of unique line circuit layers are constructed from a plurality of the merged lines.

The lines may each be associated with a weight indicative of the probability of the associated potential error process. The weights may be discretised.

Merging similar lines may comprise combining lines having at least one of: a same start point and a same end point; a same direction and length; and/or a common potential error associated with each of the lines.

Potential error processes not common to similar lines forming a merged line may be removed from association with said merged line.

The detection events may be cyclically written to the array.

The array may comprise a plurality of layers, the number of layers being a power of two.

The array may be stored in a processor cache during execution of the quantum algorithm.

The array may be representative of a local patch of quantum error correction circuits in the quantum computer.

Determining the errors in the execution of the quantum computing algorithm may comprise using a perfect matching process.

Determining the errors in the execution of the quantum computing algorithm comprises using a minimum weight perfect matching process to pair detection events in the array that are connected by one or more lines in one or more of the line circuit layers.

Determining the errors in the execution of the quantum computing algorithm may comprise using minimum weight perfect matching to match one or more detection events with a boundary of array.

Determining the errors in the execution of the quantum computing algorithm may comprise at least one of: exploring the array around a selected detection event to find an untouched detection event and pairing the selected detection event with the untouched detection event; and/or exploring the array around a selected detection event to find a previously explored region associated with a different detection event and pairing the selected detection event with the different detection event.

The exploration of the array around a selected detection event may be guided by the line circuit layers.

The layered representation may constructed by a separate computer.

According to a further aspect, there is also described a system comprising: a quantum computer for executing a quantum algorithm comprising a plurality of quantum error detection circuits; and one or more classical processing cores, wherein a set of one or more classical processing cores are configured to any of the methods described herein.

The system may further comprise a further one or more classical processing cores configured to construct the layered representation of error propagation through quantum error detection circuits in the quantum computer.

According to a further aspect, there is also described a computer program which, when executed by a processing core, causes the processing core to perform any of the methods described herein.

According to another aspect, there is provided a method of correcting a stream of syndrome measurements produced by a quantum computer using a surface code, the method comprising: prior to execution of a quantum algorithm, constructing a layered representation of error propagation through quantum error detection circuits in the quantum computer, wherein the layered representation comprises a plurality of line circuit layers, each line circuit layer representing a probability of local detection events in the quantum computer associated with one or more potential error processes in the execution of the quantum algorithm, the constructing comprising: for each of one or more quantum gates in a quantum error correction circuit, determining one or more potential detection events associated with each potential error process occurring at that quantum gate; associating one or more lines with each potential error process, the one or more lines each connecting a potential detection event associated with the potential error process to another potential detection event associated with the same potential error process or a boundary of the quantum error correction circuit; merging similar lines to form one or more merged lines; and constructing the layered representation from a plurality of the merged lines, and during execution of the quantum algorithm: receiving one or more syndrome measurements from the quantum error detection circuits in the quantum computer; converting the one or more syndrome measurements into detection events written to an array, the array representing quantum error correction circuits that are grouped together at a sequence of steps in the quantum algorithm; determining one or more errors in the execution of the quantum algorithm from the detection events in dependence upon the plurality of line circuit layers; and causing correction of the one or more syndrome measurements based on the determined one or more errors.

According to another aspect, there is provided a system comprising: a quantum computer for executing a quantum algorithm comprising a plurality of quantum error detection circuits; and one or more classical processing cores, wherein a set of one or more classical processing cores are configured to perform the method disclosed above.

According to another aspect, there is provided a computer program product comprising a non-transitory computer readable medium containing program instructions for causing a processing core to perform the method disclosed above.

Date Reçue/Date Received 2022-06-08

According to another aspect, there is provided a method of correcting a stream of syndrome measurements produced by a quantum computer, the method comprising: receiving a layered representation of error propagation through quantum error detection circuits in the quantum computer, wherein i) the layered representation comprises a plurality of line circuit layers, each line circuit layer representing a probability of local detection events in the quantum computer associated with one or more potential error processes in an execution of a quantum algorithm, and ii) the layered representation is constructed prior to execution of the quantum algorithm, wherein constructing the layered representation comprises: for each of one or more quantum gates in quantum error correction circuits, determining one or more potential detection events associated with each potential error process occurring at that quantum gate, associating one or more lines with each potential error process, the lines each connecting a potential detection event associated with the potential error process to another potential detection event associated with the same potential error process or a boundary of the quantum error correction circuits, merging similar lines to form one or more merged lines, and constructing the layered representation from a plurality of the merged lines; and during execution of the quantum algorithm: receiving one or more syndrome measurements from the quantum error detection circuits in the quantum computer; converting the one or more syndrome measurements into detection events written to an array, the array representing quantum error correction circuits that are grouped together at a sequence of steps in the quantum algorithm; determining one or more errors in the execution of the quantum algorithm from the detection events in dependence upon the plurality of line circuit layers; and causing correction of the one or more syndrome measurements based on the determined one or more errors.

According to another aspect, there is provided a system comprising: a quantum computer for executing a quantum algorithm comprising a plurality of quantum error detection circuits; and one or more classical processing cores, wherein a set of one or more classical processing cores are configured to perform the method disclosed above.

According to another aspect, there is provided a computer program product comprising a non-transitory computer readable medium containing program instructions for causing a processing core to perform the method disclosed above.

Brief description of the drawings

Figure 1 shows a schematic example of a quantum computing system;

Date Reçue/Date Received 2022-06-08

Figure 2 shows a flow diagram of an overview of a method for quantum error correction;
Figure 3 shows an example of a three dimensional data structure representing a plurality of rounds of syndrome measurements;
Figure 4 shows a flow diagram of an example of a method for generating a layered representation of potential errors in a quantum algorithm;
Figure 5 shows an example of a list of detection events associated with potential errors occurring at a quantum gate; and
Figure 6 shows a flow chart of an example of a method of determining errors in the execution of the quantum algorithm from the detection events.

Detailed description

Figure 1 illustrates a schematic example of an embodiment of a quantum computer 100.

The quantum computer 100 comprises a quantum computing layer 102, a control layer 104 and a classical processing layer 106.

The quantum computing layer 102 comprising comprises an array of quantum devices or qubits 108 configured to perform a quantum computing algorithm. The quantum devices or qubits 108 comprise a mixture of data qubits and syndrome qubits (also referred to herein as measurement qubits). The quantum computing layer 102 further comprises a plurality of quantum gates (not shown) for performing operations on the qubits 108. In some embodiments, the quantum computing layer is in the form of a 2D two-dimensional array of quantum devices 108.

The quantum devices or qubits 108 can, for example, be superconducting qubits. The quantum computing layer 102 is kept at a sufficiently low temperature to maintain coherence between qubits throughout the execution of the quantum algorithm (for

Date Reçue/Date Received 2022-06-08

example, below 20mK. In order to facilitate high coherence, the qubits may be cleanly manufactured. Qubit geometry may also be chosen to enhance coherence. In embodiments where superconducting qubits are used, the temperature is kept below the superconducting critical temperature. To improve coherence, the temperature can be kept well below the superconducting critical temperature. Herein, for the term "qubit" and "quantum device" will be used interchangeably.

A control layer 104 comprising a plurality of control devices 110 is interposed between the quantum computing layer 102 and the classical processing layer 106. The control devices 110 receive raw measurement data from qubits 108 in the quantum computing layer 102 and convert them into binary measurement data for use in the classical processing layer 106. The control devices 110 can, in some embodiments, also issue instructions to the quantum devices 108, for example to instruct a quantum gate (not shown) to perform a qubit rotation. In some embodiments, each control device 110 is connected to around six quantum devices 108. The control devices 110 are, in some embodiments, Field Programmable Gated Arrays (FPGAs).

The classical processing layer 106 comprises an array of processing cores 112. The classical processing layer 106 is coupled to the quantum computing layer 102 via the control layer 104. Processing cores 112 in the classical computing layer 106 are associated with local patches 114 of qubits 108. A local patch 114 comprises a plurality of qubits 108 is grouped together. In some embodiments, the patch size is around one - hundred qubits. Each processor core comprises a central processing unit ("CPU" or "processor") and on-chip cache memory in the form of L1 and L2 caches (herein referred to as a "processor cache"). Processing cores 112 in the classical processing layer 106 are coupled to neighbouring processing cores 112 in the classical processing layer 106. This can allow for the processing cores 112 to exchange data with their neighbouring cores. In some embodiments, the processing cores 112 form a two-dimensional array of processing cores 112 within the classical processing layer 106.

The classical processing cores 112 are provided with dedicated assembly instructions, that, when executed by a processor in the processor core, cause the processor to access the processor cache of a neighbouring core. This can result in low latency communication between neighbouring cores, as a complex memory controller is not needed.

In use, the quantum computing layer 102 executes a quantum computing algorithm, such as Shor's algorithm or the Deutsch-Jozsa algorithm for example. Syndrome qubits in the quantum computing layer 102 interact with their neighbouring data qubits to produce raw measurement data. The raw measurement data is fed into the control devices 110 of the control layer 104. The control devices 110 convert the raw measurement data into binary measurement data and stream the binary measurement data into the processing cores 112 of the classical processing layer 106.

The classical processing layer 106 converts the binary measurement data from the control layer 104 into parities of the quantum states of qubits 108 in the quantum computing layer 102 that were measured. These parities are then processed by the classical processing layer 106 to determine any quantum errors and/or the required corrected parities for the quantum devices 108 in the quantum computing layer 102. The determined corrections can then be processed further to determine and cause the required corrective action. In some embodiments, this further processing can be performed in additional classical processing layers (not shown).

Figure 2 shows a flow diagram of an example of a method for correcting errors in a quantum computer. In the following, the method will be described in relation to the quantum computer in Figure 1. However, in general it can be performed by any quantum computing system that provides syndrome measurements from qubits to one or more classical processing cores.

The method (herein also referred to as a quantum error correction method) corrects a stream of measurement data obtained from syndrome qubits in a quantum computer. In some embodiments, the method is used to correct errors in a two-dimensional array of qubits, such as the quantum computing layer described above in relation to Figure 1. In other embodiments, the method can be applied to a one-dimensional array of qubits.

At operation 115, the classical processing cores 112 in the classical processing layer 106 receive a layered representation of error propagation through quantum error detection circuits in the quantum computer 100, wherein the layered representation comprises a plurality of line circuit layers, each line circuit layer representing a probability of local detection events in the quantum computer associated with one or more potential error processes in the execution of the quantum algorithm.

In some embodiments, the layered representation comprises one or more layers (also referred to herein as line circuit layers). Each line circuit layer represents one or more potential detection events that can occur during a round of error detection at stages of the quantum algorithm. For example, a layer can comprise a representation of all the detection events that can occur at syndrome qubits during a round of error detection during execution of the quantum algorithm.

Emanating from each of the one or more potential detection events in a layer is a set of weighted lines connecting that detection event to one or more other potential detection events. The other potential detection events can occur in the same round of error detection as the detection event from which the line emanates, and hence lie within the same layer. The other detection events can also be in other rounds of error detection.

The weight of each line is representative of the probability of detection events at each end of the line occurring together as the result of an error. In some embodiments, the lines are weighted such that higher probability lines have a lower weight than lower probability lines. For example, the lines can be weighted according to:

$$w_i = -C \ln p_i,$$

where w_i is the weight of the i^{th} line, p_i is the probability associated with the i^{th} line, and C is a constant.

The layered representation is pre-determined based on knowledge of the quantum algorithm and the possible errors that can occur during its execution. By predetermined, it is meant that the layered representation is determined before the execution of the quantum algorithm begins. In some embodiments, the layered representation is constructed by a separate computing system to the quantum computing system. For example, the layered representation may be constructed by a remote computing system based on knowledge of the quantum algorithm that will be run on the quantum computing system. The layered representation can be transmitted from this remote computing system to the quantum computing system prior to the execution of the quantum algorithm.

In some embodiments, the layered representation is chosen such that the representation for each local patch of quantum devices in the quantum computing layer fits into the processor cache of the corresponding processor core in the first classical computing layer. This can reduce or even eliminate the number of computationally

slow RAM retrievals performed by the processor cores in the classical computing layer, which can speed up the error correction. Construction of the layered representation will be described in more detail below in relation to Figure 4.

At operation 116, the quantum computer begins execution of a quantum algorithm. A quantum algorithm is executed in a quantum computer. In some embodiments, the quantum computer executed the quantum algorithm in a quantum computing layer, as described in relation to Figure 1. The quantum computer executes a quantum algorithm with a surface code being used for error correction. The surface code can be visualized as a large checker-board. Each square represents a qubit. White squares correspond to qubits being used to store data, and are referred to as data qubits. Black squares correspond to qubits being used to check for errors in their neighbouring four data qubits, and are referred to as measurement or syndrome qubits. Computation is achieved by turning on and off regions of the surface code. Physically, this involves causing the measurement qubits to cease interacting with data qubits within specific regions of the quantum computing layer.

During execution of the quantum algorithm, syndrome qubits perform measurements on neighbouring data qubits to produce syndrome measurements. These measurements can be transmitted to the classical processing layer for processing to determine if any errors have occurred in the execution of the quantum computing algorithm. In some embodiments, the syndrome measurements are transmitted to the classical processing layer via the control layer.

Rounds of measurements are performed sequentially in time. The results of the method can be visualized as operating in a two dimensional (for a one-dimensional array of qubits) or three dimensional (for a two-dimensional array of qubits) data structure, sometimes called the space-time or volume.

At operation 118, the classical processing cores 112 receive one or more syndrome measurements from quantum error detection circuits in the quantum computer.

In some embodiments, the syndrome measurements pass via the control layer in the quantum computing system. The control layer converts the raw syndrome measurement data into binary measurement data that can be processed by the classical

processing cores. This binary syndrome measurement data is transmitted from the control to the classical processing cores for further processing.

In some embodiments, the classical processing cores receive the syndrome measurement data in its raw form.

At operation 120, the classical processing cores 112 convert the syndrome measurements into detection events written to an array, the array in each processing core representing a patch of quantum error correction circuits at a sequence of steps in the quantum algorithm.

The array in each processor core is constructed from the layered representation by building the array sequentially from layers representing the steps of the quantum algorithm. As a new round of error detection is preformed, each processor core adds a layer to the array representing that error detection round. A particular layer may be repeated multiple times within the array, either in succession or not, in dependence on the quantum algorithm being performed.

In some embodiments, the array is written cyclically to the processor cache of a processing core 112. Each processor cache stores an array representative of the local patch of qubits associated with its processing core 112.

In some embodiments, a detection event is determined to have occurred when a change in the measurement compared to previous syndrome measurements is detected. The change in value of the syndrome measurements is indicative of an end point of an error chain. The determined detection event is written to the array.

At operation 122 the classical processing cores 112 determine one or more errors in the execution of the quantum algorithm from the detection events using perfect matching and the stored line circuit layers.

Errors in the execution of the quantum computing algorithm are determined by matching pairs of detection events in the array, or by matching a detection event to an abstract boundary of the array. Pairs are matched such that the most likely pairings are found.

The pairing of detection events is performed by processing cores in the classical processing layer. Each processor core attempts to match the detection events in its array with detection events in the same array, detection events in the arrays of neighbouring processing cores, and/or an abstract boundary of the quantum circuit.

At operation 124 correction of the syndrome measurements based on the determined errors is performed.

Figure 3 shows an example of a three dimensional data structure representing a plurality of rounds of syndrome measurements. The three dimensional data structure is in the form of an array 126. The array 126 comprises a plurality of line circuit layers 128 (not all layers are labelled in Figure 3). Each layer 128 is representative of a round of syndrome measurements at a local patch of qubits performed during the execution of the quantum algorithm.

The array 126 is constructed from line circuit layers 128 in the layered representation. Each layer 126 comprises a plurality of points 130 (not all points are labelled in Figure 3) and a plurality of lines 132 (not all lines are labelled in Figure 3). Each point in the array corresponds to a potential error detection event at a syndrome qubit during the execution of the quantum algorithm. One or more lines 132 emanate from each point 130 in the array. Lines 132 connect pairs of points in the array that correspond to detection events resulting from known potential errors in the execution of the quantum algorithm.

During execution of the quantum algorithm, an array 126 representative of a local patch of quantum devices is constructed in a classical processing core. The array 126 is constructed from the layered representation. In some embodiments, such as the embodiment shown, the array 126 is written cyclically to the processor cache of the processor core. The break in the array indicates the layer representing the current round of error detection measurements.

As measurement data is streamed to the processor core, detection events 134 (only some of which are labelled in the figure) are identified and written to the array. The processor core attempts to pair up these detection events 134 using the lines 132 that emanate from them. In embodiments where lines 132 are weighted to have a lower

weight for a higher probability of the line 132 occurring, detection events 134 are paired using minimum weight perfect matching.

Figure 4 shows a flow diagram of an example of a method for generating a layered representation of potential errors in a quantum algorithm. Prior to the execution of the quantum algorithm, the data structures required for the algorithm are constructed.

At operation 136 the quantum circuit used for the execution of the quantum algorithm is input. An input file describing the quantum circuit used to implement quantum error detection is either read, or first generated and then read, into a classical computer. The input file describes the gate operations performed by qubits in the quantum computer to generate each round of syndrome measurement data and/or to execute the quantum algorithm.

At operation 138 potential error detection events in the quantum circuit are determined. For every gate in the circuit, one or more possible error processes is simulated. In some embodiments, every potential error at the gates is simulated. Single-qubit errors are simulated for single-qubit gates, and two-qubit errors are simulated for two-qubit gates.

The resulting detection events at one or more of the syndrome qubits are then determined. A detection event is a point in space and time where a stream of measurement results from a syndrome qubit changes value. The result is a list of all gate error processes.

Each potential error is associated with a list of detection events that result from it. In some embodiments, each error will have between 0 and 4 detection events associated with it. Figure 5 shows an example of a list of detection events associated with potential errors occurring at a quantum gate 140.

A number of errors 142 can occur at a quantum gate 140 in the quantum circuit during execution of the quantum algorithm. These errors are simulated by the classical computer, and the resulting detection events 144 at syndrome qubits in the quantum circuit are determined for each error.

Referring again to Figure 4, at operation 146, the error lists are broken down into shorter error lists. For each error with more than two associated detection events, this comprises constructing the detection event list from other lists of detection events associated with other errors with no more than two detection events associate with them.

At operation 148, "pre-lines" are associated with each list of errors. For each error associated with two detection events, a "pre-line" is associated which connects the two detection events. For each error associated with one detection event, a pre-line leading to an abstract boundary of the space-time is associated.

Each of these types of pre-lines is assigned the full probability of its generating error occurring. Errors associated with more than two detection events get associated with lists of pre-lines that collectively cover these detection events. In some embodiments, multiple distinct lists of pre-lines can exactly cover the list of detection events, and all lists are found. If there are N lists, each pre-line in each list gets assigned 1/Nth of the probability of its generating error.

At operation 150, pre-lines are coalesced. For example, if two pre-lines start and end at the same locations in space-time, the pre-lines are replaced with a single pre-line having the total probability of the original two. This process is continued until no pre-lines can be coalesced. In addition to the total probability, the coalesced pre-lines retain a list of all the errors generating detection events at their endpoints.

At operation 152, the pre-lines are converted to lines. Every pre-line not ending on the abstract boundary has potentially two lines associated with it. Pre-lines connect two specific locations in space-time. By contrast, lines are vectors. For example a single time-axis aligned pre-line can become two lines, one pointing into the future, and one pointing into the past. However, a pre-line ending at the abstract boundary will potentially only become a single line. Each line keeps a list of all errors leading to the pre-line, but with error positions made relative rather than absolute. Each error is propagated back to the beginning of the circuit it is associated with, forming a circuit-independent list of errors for each original error.

In some embodiments, pre-lines are converted into lines if: the new line would point in a different direction or have a different length to any previous line; the new line would

have a different total probability despite pointing in the same direction and having the same length as a previous line; or the new line points in the same direction, has the same length, and the same total probability as one or more existing lines, but has no error lists in common with any of these lines. Each point in space-time is therefore associated with a list of lines containing 0 or more entries. The total number of times each line appears in some list (the "count" of the line) is recorded. Some lines have much higher counts than others.

At operation 154, similar lines are merged. Lines are determined to be similar if they point in the same direction, have the same length, and/or share one or more errors. In some embodiments, all three conditions are required to determine that lines are similar. Errors not common to both lines are discarded. The probability of the merged line, p_l, is set to be:

$$p_l = p_1^{\left(\frac{N_1}{N_1+N_2}\right)} p_2^{\left(\frac{N_2}{N_1+N_2}\right)},$$

where p_1 is the probability associated with the first line, p_2 is the probability associated with the second line, N_1 is the count of the first line and N_2 is the count of the second line.

In some embodiments, where there are more than two lines that are determined to be similar, the pair with the minimum total count $(N_1 + N_2)$ is chosen for merging. This can minimize the impact of information-lossy merging.

In some embodiments, lines are combined until there are fewer than 256 unique lines. By combining lines, the total number of lines can be reduced, which can simplify the identification of errors in the execution of the quantum algorithm. Reducing the number of lines can also reduce the size of the data-structures used in quantum error correction. In some embodiments, the size of the data-structures can be reduced to fit into the processor cache of a processor core. This can speed up execution of the error correction, as less or even no external memory access will be used. Accordingly, this results in a technological improvement in the field of quantum error correction in that fewer processing resources – memory and processing cycles - are require than in systems that do not implement the systems and methods described in this specification.

At operation 156 "jigsaw pieces" are constructed from the unique lines. The circuit description may contain instructions such as "repeat this sequence of circuit layers n times" and therefore the true data structures in every step can be made out of the smallest conveniently constructible number of jigsaw pieces that can be repeated to form the data structure corresponding to the complete circuit. The number of pieces can change from step to step. This decomposition into repeating pieces is necessary as the total number of repetitions of different parts of the circuit can be so large that it would not be feasible to construct and manipulate a non-decomposed representation of each step.

At operation 158 the combined lines are used to construct line circuit layers. Once the final compressed list of lines is obtained, the minimum number of unique layers of lines required to describe the complete data structure can then be constructed. Only the shortest error list for each line is kept.

The result is a highly compressed data structure representing the error detection part of the quantum circuit that can fit into the processor cache of a processor core. In some embodiments, the data structure is small enough to fit into the L1 cache of a processor core. Each line in each layer represents an approximation of the total probability of observing detection events at the endpoints of the line. Detection events in the quantum circuit occur at the endpoints of lines.

A complete run of the algorithm considers of order 100 qubits per local patch executing potentially millions of rounds of error detection during the operation of the quantum computer. The algorithm converts the stream of syndrome measurements from the quantum computer into a smaller number of detection events. The detection events are randomly distributed in the space-time. Rather than storing these detection events in a huge space-time, in some embodiments just thirty-two rounds, each with enough space for a sixteen-by-sixteen array of qubits, are kept. In some embodiments, it can be advantageous that the array dimensions are all powers of two.

To be explicit, in embodiments where thirty-two layers are stored, detection events associated with detection circuit zero will be stored in space-time layer zero, as will detection events associated with detection circuit thirty-two, sixty-four, and so on. The cyclic use of a small volume of space-time with power-of-two size allows the direction and length of each line to be represented using a single 2 byte integer. Each space-time

location can be considered a voxel, and given the index of any voxel, the index of any other voxel can be calculated by adding this 2 byte integer and discarding any unnecessary high bits. This can greatly speed up processing. Furthermore, as the data structures fit into the L1 cache, less or even no computationally expensive main RAM accesses are used, which can also speed up the method.

During execution of the algorithm, measurement data from the syndrome qubits in the quantum computer are streamed into processor cores in a classical computing layer (as described in relation to Figure 1, for example). In some embodiments, each processor core handles a local patch of around 100 qubits.

As the algorithm is executed, detection events start to populate the cyclic array representing the space-time volume currently being considered by the processor core. These detection events are the endpoints of unknown error chains, and must be matched up with other detection events or a boundary in order to determine a reasonable set of errors to use as the basis for corrections to the stream of syndrome measurements. Detection events are matched along paths of lines connecting them. As described above, each line has a list of errors that can be used in the correction of an error if the line is part of a matched path. The corrections propagate through the error detection circuits by flipping measurement values. This propagation is itself potentially computationally expensive and must be carefully compiled into a collection of cheap bit register manipulations before the quantum computer begins execution.

Figure 6 shows a flow chart of an example of a method of determining errors in the execution of the quantum algorithm from the detection events. In embodiments where the line weights are lower for lines with a higher probability of their associated errors occurring, minimum weight perfect matching is used to match the detection events.

At operation 160 an unmatched detection event in the array is chosen. The event can, in some embodiments, be chosen at random from the array. Alternatively, the detection event may be chosen using a rule, such as choosing the oldest unmatched detection event for example.

At operation 162, the array around the chosen detection event is explored. The lines emanating from the chosen detection event are treated like "water pipes", with the "water" traveling more quickly along the lines with higher probability. The region of

space-time with water associated with a given detection event is called that detection event's exploratory region. For example, lower weighted lines can be thought of as narrower pipes. During the exploration, the same volume of water can be added to each of the pipes. The water will extend further along the length of the narrow pipes than the wider ones (which are associated with lower probability event pairings).

At operation 164, it is determined whether additional measurement data is required. If the exploratory region starts flowing towards a space-time layer for which syndrome qubit measurement data is not yet available, then it is determined that more data is required.

If this determination is positive, additional syndrome measurement data is obtained at operation 166. The quantum computer provides this data by performing another round of error detection.

If this determination is negative, then at operation 168 the array continues to be explored around the detection event.

At operation 170 detection events found in the exploratory region are matched. Exploratory regions of different detection events are not permitted to overlap. If two detection events can be found such that just enough water can be added to each such that their exploratory regions touch without overlapping any other exploratory regions, they can be matched.

In more complicated situations, an alternating tree will be created that may in turn lead to blossoms being created or destroyed, and these details are described in the "Timing analysis" paper.

At operation 172, the processor core determines whether any unmatched detection events remain in the array. If unmatched detection events remain, another detection event is chosen, and the process is repeated.

If no further detection events remain, at operation 174 the processor core waits for further syndrome measurement data from the quantum computer.

In some embodiments, the exploration of the neighbourhood around a detection event is handled in a hierarchical manner. By checking one or more simpler cases before calling the general matching algorithm, the efficiency of the matching can be improved.

The initial step of the hierarchy is to explore the array around a selected detection event to find an untouched detection event and pair the selected detection event with the untouched detection event. Firstly, when a detection event is first created it is impossible for any neighbouring space-time voxel to contain any water. Such detection events are called untouched. If a pair of neighbouring untouched detection events can be found, it is possible to add just enough water to each to enable them to be matched without first checking for water in all of the voxels neighbouring both of these detection events. If this is the case, then these events are matched, and then another detection event is chosen. If no pair of neighbouring untouched detection events cannot be found, then the next special case is considered.

The second step of the hierarchy is to explore the array around a selected detection event to find a previously explored region associated with a different detection event and pair the selected detection event with the different detection event. Given neighbouring touched detection events (that is to say, detection events with water already added to their lines), the maximum amount of water possible to add to each must be calculated by checking how much water is in each of the neighbouring voxels and subtracting this from the weight of the line to that voxel. If enough water can be added to each to permit a matching (i.e. no other detection event exploratory region will be overlapped with), this is done, and another detection event is chosen.

If neither of the above cases applies, the general minimum weight perfect matching algorithm is called.

Implementations of the quantum subject matter and quantum operations described in this specification may be implemented in suitable quantum circuitry or, more generally, quantum computational systems, including the structures disclosed in this specification and their structural equivalents, or in combinations of one or more of them. The term "quantum computational systems" may include, but is not limited to, quantum computers, quantum information processing systems, quantum cryptography systems, or quantum simulators.

The terms quantum information and quantum data refer to information or data that is carried by, held or stored in quantum systems, where the smallest non-trivial system is a qubit, e.g., a system that defines the unit of quantum information. It is understood that the term "qubit" encompasses all quantum systems that may be suitably approximated as a two-level system in the corresponding context. Such quantum systems may include multi-level systems, e.g., with two or more levels. By way of example, such systems can include atoms, electrons, photons, ions or superconducting qubits. In many implementations the computational basis states are identified with the ground and first excited states, however it is understood that other setups where the computational states are identified with higher level excited states are possible. It is understood that quantum memories are devices that can store quantum data for a long time with high fidelity and efficiency, e.g., light-matter interfaces where light is used for transmission and matter for storing and preserving the quantum features of quantum data such as superposition or quantum coherence.

Quantum circuit elements may be used to perform quantum processing operations. That is, the quantum circuit elements may be configured to make use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data in a non-deterministic manner. Certain quantum circuit elements, such as qubits, may be configured to represent and operate on information in more than one state simultaneously. Examples of superconducting quantum circuit elements that may be formed with the processes disclosed herein include circuit elements such as co-planar waveguides, quantum LC oscillators, qubits (e.g., flux qubits or charge qubits), superconducting quantum interference devices (SQUIDs) (e.g., RF-SQUID or DCSQUID), inductors, capacitors, transmission lines, ground planes, among others.

In contrast, classical circuit elements generally process data in a deterministic manner. Classical circuit elements may be configured to collectively carry out instructions of a computer program by performing basic arithmetical, logical, and/or input/output operations on data, in which the data is represented in analogue or digital form. In some implementations, classical circuit elements may be used to transmit data to and/or receive data from the quantum circuit elements through electrical or electromagnetic connections. Examples of classical circuit elements that may be formed with the processes disclosed herein include rapid single flux quantum (RSFQ) devices,

30 reciprocal quantum logic (RQL) devices and ERSFQ devices, which are an energy-efficient version of RSFQ that does not use bias resistors. Other classical circuit elements may be formed with the processes disclosed herein as well.

During operation of a quantum computational system that uses superconducting quantum circuit elements and/or superconducting classical circuit elements, such as the circuit elements described herein, the superconducting circuit elements are cooled down within a cryostat to temperatures that allow a superconductor material to exhibit superconducting properties.

While this specification contains many specific implementation details, these should not be construed as limitations, but rather as descriptions of features that may be specific to particular implementations. Certain features that are described in this specification in the context of separate implementations can also be implemented in combination in a single implementation. Conversely, various features that are described in the context of a single implementation can also be implemented in multiple implementations separately or in any suitable sub-combination. Moreover, although features may be described above as acting in certain combinations, one or more features from a combination can in some cases be excised from the combination, and the combination may be directed to a sub-combination or variation of a sub-combination.

Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. For example, the actions can be performed in a different order and still achieve desirable results. In certain circumstances, multitasking and parallel processing may be advantageous. Moreover, the separation of various components in the implementations described above should not be understood as requiring such separation in all implementations.

A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. The invention, rather, is defined by the claims.

Date Reçue/Date Received 2022-06-08

Claims

1. A method of correcting a stream of syndrome measurements produced by a quantum computer using a surface code, the method comprising:

prior to execution of a quantum algorithm, constructing a layered representation of error propagation through quantum error detection circuits in the quantum computer, wherein the layered representation comprises a plurality of line circuit layers, each line circuit layer representing a probability of local detection events in the quantum computer associated with one or more potential error processes in the execution of the quantum algorithm, the constructing comprising:

for each of one or more quantum gates in a quantum error correction circuit, determining one or more potential detection events associated with each potential error process occurring at that quantum gate;

associating one or more lines with each potential error process, the one or more lines each connecting a potential detection event associated with the potential error process to another potential detection event associated with the same potential error process or a boundary of the quantum error correction circuit;

merging similar lines to form one or more merged lines; and

constructing the layered representation from a plurality of the merged lines, and during execution of the quantum algorithm:

receiving one or more syndrome measurements from the quantum error detection circuits in the quantum computer;

converting the one or more syndrome measurements into detection events written to an array, the array representing quantum error correction circuits that are grouped together at a sequence of steps in the quantum algorithm;

determining one or more errors in the execution of the quantum algorithm from the detection events in dependence upon the plurality of line circuit layers; and

causing correction of the one or more syndrome measurements based on the determined one or more errors.

2. The method of claim 1, wherein the one or more lines are each associated with a weight indicative of the probability of the associated potential error process.

3. The method of claim 2, wherein the weights are discretised.

Date Reçue/Date Received 2022-06-08

4. The method of any one of claims 1 to 3, wherein merging similar lines comprises combining lines of the one or more lines associated with each potential error process having at least one of:

a same start point and a same end point;

a same direction and length; and/or

a common potential error.

5. The method of any one of claims 1 to 4, wherein potential error processes not common to similar lines forming a merged line are removed from association with said merged line.

6. The method of any one of claims 1 to 5, wherein the detection events are cyclically written to the array.

7. The method of claim 6, wherein the array comprises a plurality of layers, the number of layers in the plurality of layers being a power of two.

8. The method of any one of claims 1 to 7, wherein the array is stored in a processor cache during execution of the quantum algorithm.

9. The method of any one of claims 1 to 8, wherein the array is representative of a local patch of the quantum error correction circuits in the quantum computer.

10. The method of any one of claims 1 to 9, wherein determining the errors in the execution of the quantum computing algorithm comprises using a perfect matching process.

11. The method of any one of claims 1 to 10, wherein determining the errors in the execution of the quantum computing algorithm comprises using a minimum weight perfect matching process to pair the detection events in the array that are connected by one or more lines in one or more of the line circuit layers.

12. The method of any one of claims 1 to 11, wherein determining the errors in the execution of the quantum computing algorithm comprises using minimum weight perfect matching to match one or more of the detection events with a boundary of the array.

Date Reçue/Date Received 2022-06-08

13. The method of any one of claims 1 to 12, wherein determining the errors in the execution of the quantum computing algorithm comprises at least one of:

exploring the array around a selected detection event to find an untouched detection event and pairing the selected detection event with the untouched detection event; and/or

exploring the array around a selected detection event to find a previously explored region associated with a different detection event and pairing the selected detection event with the different detection event.

14. The method of claim 13, wherein the exploring of the array around a selected detection event is guided by the line circuit layers.

15. The method of any one of claims 1 to 14, wherein the layered representation is constructed by a separate computer.

16. A system comprising:

a quantum computer for executing a quantum algorithm comprising a plurality of quantum error detection circuits; and

one or more classical processing cores,

wherein a set of one or more classical processing cores are configured to perform the method of any one of claims 1 to 15.

17. The system of claim 16, further comprising a further one or more classical processing cores configured to construct the layered representation of error propagation through quantum error detection circuits in the quantum computer.

18. A computer program product comprising a non-transitory computer readable medium containing program instructions for causing a processing core to perform the method of any one of claims 1 to 15.

19. A method of correcting a stream of syndrome measurements produced by a quantum computer, the method comprising:

receiving a layered representation of error propagation through quantum error detection circuits in the quantum computer, wherein

i) the layered representation comprises a plurality of line circuit layers, each line circuit layer representing a probability of local detection events in the quantum computer

Date Reçue/Date Received 2022-06-08

associated with one or more potential error processes in an execution of a quantum algorithm, and

ii) the layered representation is constructed prior to execution of the quantum algorithm, wherein constructing the layered representation comprises:

for each of one or more quantum gates in quantum error correction circuits, determining one or more potential detection events associated with each potential error process occurring at that quantum gate,

associating one or more lines with each potential error process, the lines each connecting a potential detection event associated with the potential error process to another potential detection event associated with the same potential error process or a boundary of the quantum error correction circuits,

merging similar lines to form one or more merged lines, and

constructing the layered representation from a plurality of the merged lines; and

during execution of the quantum algorithm:

receiving one or more syndrome measurements from the quantum error detection circuits in the quantum computer;

converting the one or more syndrome measurements into detection events written to an array, the array representing quantum error correction circuits that are grouped together at a sequence of steps in the quantum algorithm;

determining one or more errors in the execution of the quantum algorithm from the detection events in dependence upon the plurality of line circuit layers; and

causing correction of the one or more syndrome measurements based on the determined one or more errors.

20. The method of claim 19, wherein the one or more lines are each associated with a weight indicative of the probability of the associated potential error process.

21. The method of claim 20, wherein the weights are discretised.

22. The method of any one of claims 19 to 21, wherein merging similar lines comprises combining lines of the one or more lines associated with each potential error process having at least one of:

a same start point and a same end point;

Date Reçue/Date Received 2022-06-08

a same direction and length; and/or

a common potential error.

23. The method of any one of claims 19 to 22, wherein potential error processes not common to similar lines forming a merged line are removed from association with said merged line.

24. The method of any one of claims 19 to 23, wherein the detection events are cyclically written to the array.

25. The method of claim 24, wherein the array comprises a plurality of layers, the number of layers in the plurality of layers being a power of two.

26. The method of any one of claims 19 to 25, wherein the array is stored in a processor cache during execution of the quantum algorithm.

27. The method of any one of claims 19 to 26, wherein the array is representative of quantum error correction circuits that are grouped together in the quantum computer.

28. The method of any one of claims 19 to 27, wherein determining the errors in the execution of the quantum computing algorithm comprises using a perfect matching process.

29. The method of any one of claims 19 to 27, wherein determining the errors in the execution of the quantum computing algorithm comprises using a minimum weight perfect matching process to pair the detection events in the array that are connected by one or more lines in one or more of the line circuit layers.

30. The method of any one of claims 19 to 27, wherein determining the errors in the execution of the quantum computing algorithm comprises using a minimum weight perfect matching process to match one or more of the detection events with a boundary of the array.

31. The method of any one of claims 19 to 30, wherein determining the errors in the execution of the quantum computing algorithm comprises at least one of:

exploring the array around a selected detection event to find an untouched detection event and pairing the selected detection event with the untouched detection event; and/or

Date Reçue/Date Received 2022-06-08

exploring the array around a selected detection event to find a previously explored region associated with a different detection event and pairing the selected detection event with the different detection event.

32. The method of claim 31, wherein the exploring of the array around a selected detection event is guided by the line circuit layers.

33. The method of any one of claims 19 to 32, wherein the layered representation is constructed by a separate computer.

34. A system comprising:

a quantum computer for executing a quantum algorithm comprising a plurality of quantum error detection circuits; and

one or more classical processing cores,

wherein a set of one or more classical processing cores are configured to perform the method of any one of claims 19 to 33.

35. The system of claim 34, further comprising a further one or more classical processing cores configured to construct the layered representation of error propagation through quantum error detection circuits in the quantum computer.

36. A computer program product comprising a non-transitory computer readable medium containing program instructions for causing a processing core to perform the method of any one of claims 19 to 33.

Date Reçue/Date Received 2022-06-08

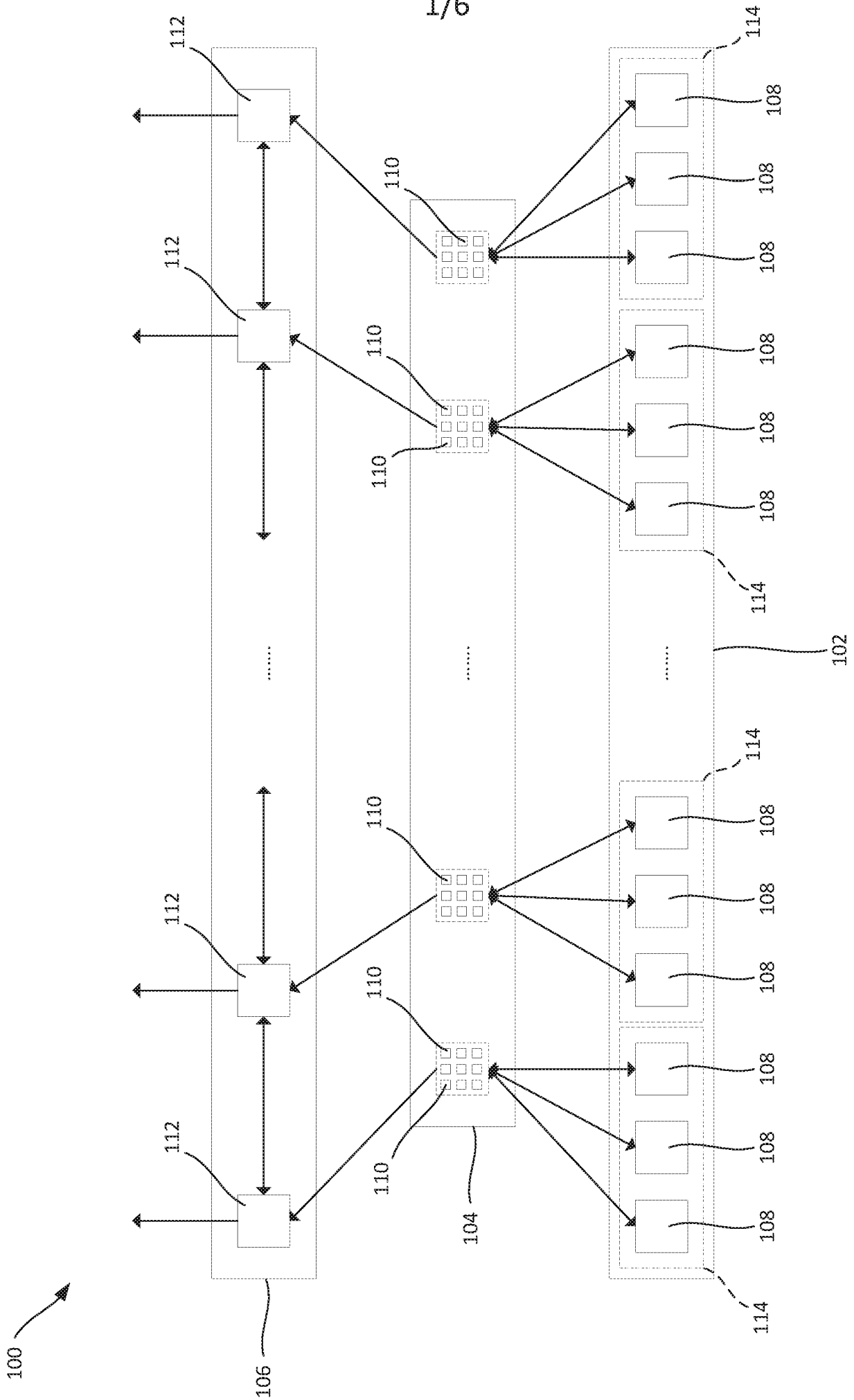


Fig. 1

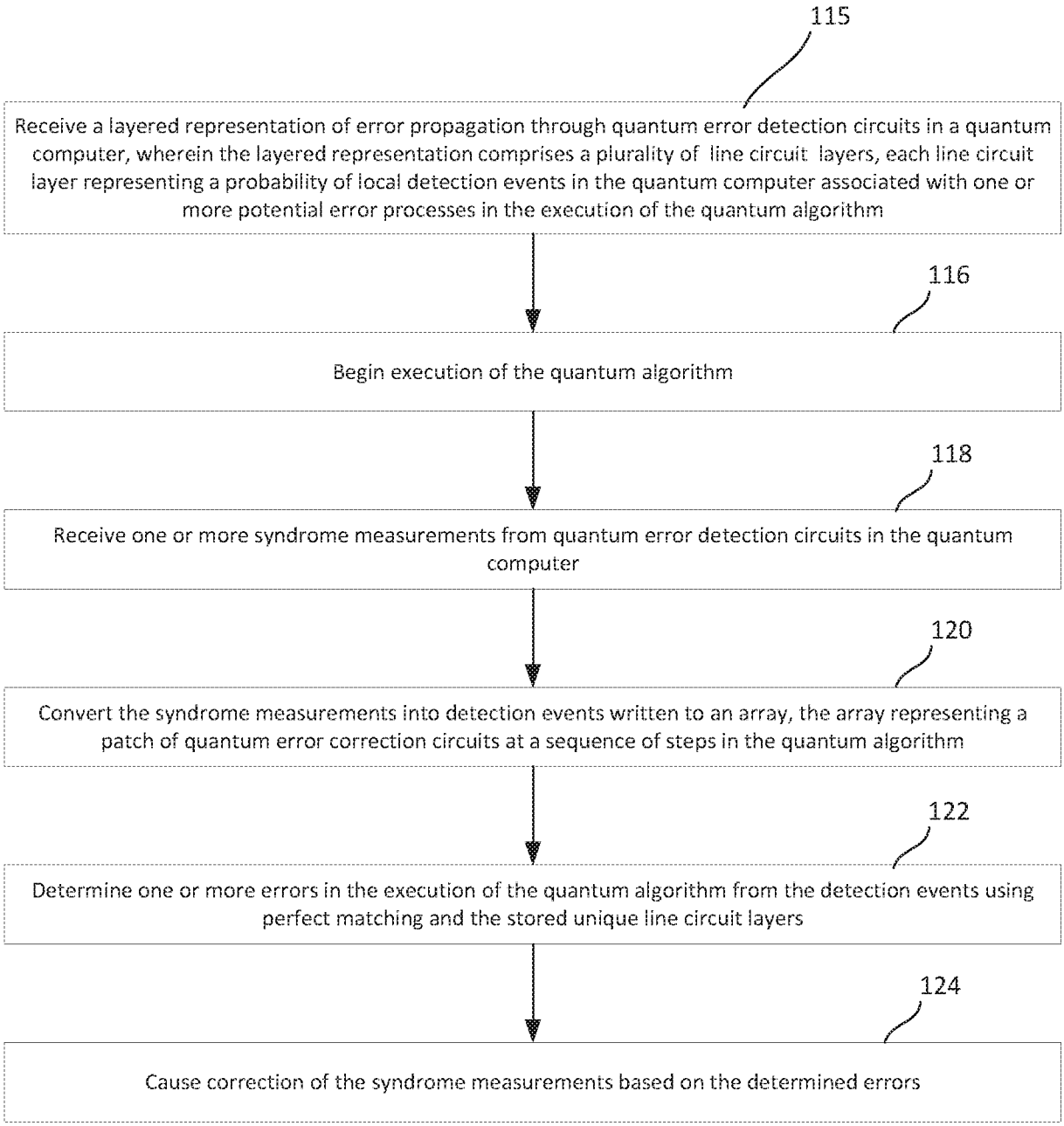

115
Receive a layered representation of error propagation through quantum error detection circuits in a quantum computer, wherein the layered representation comprises a plurality of line circuit layers, each line circuit layer representing a probability of local detection events in the quantum computer associated with one or more potential error processes in the execution of the quantum algorithm
116
Begin execution of the quantum algorithm
118
Receive one or more syndrome measurements from quantum error detection circuits in the quantum computer
120
Convert the syndrome measurements into detection events written to an array, the array representing a patch of quantum error correction circuits at a sequence of steps in the quantum algorithm
122
Determine one or more errors in the execution of the quantum algorithm from the detection events using perfect matching and the stored unique line circuit layers
124
Cause correction of the syndrome measurements based on the determined errors

Fig. 2

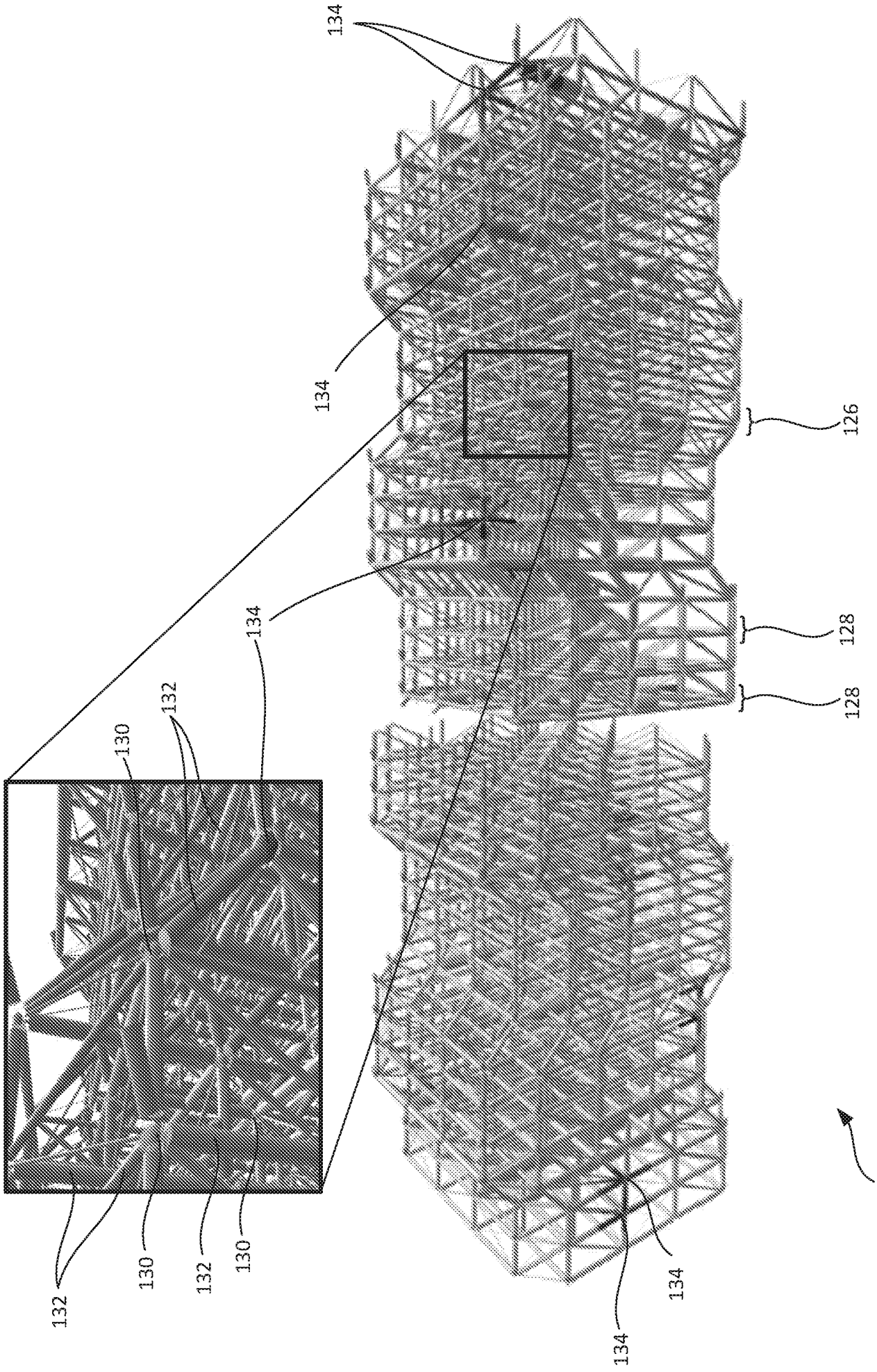


Fig. 3

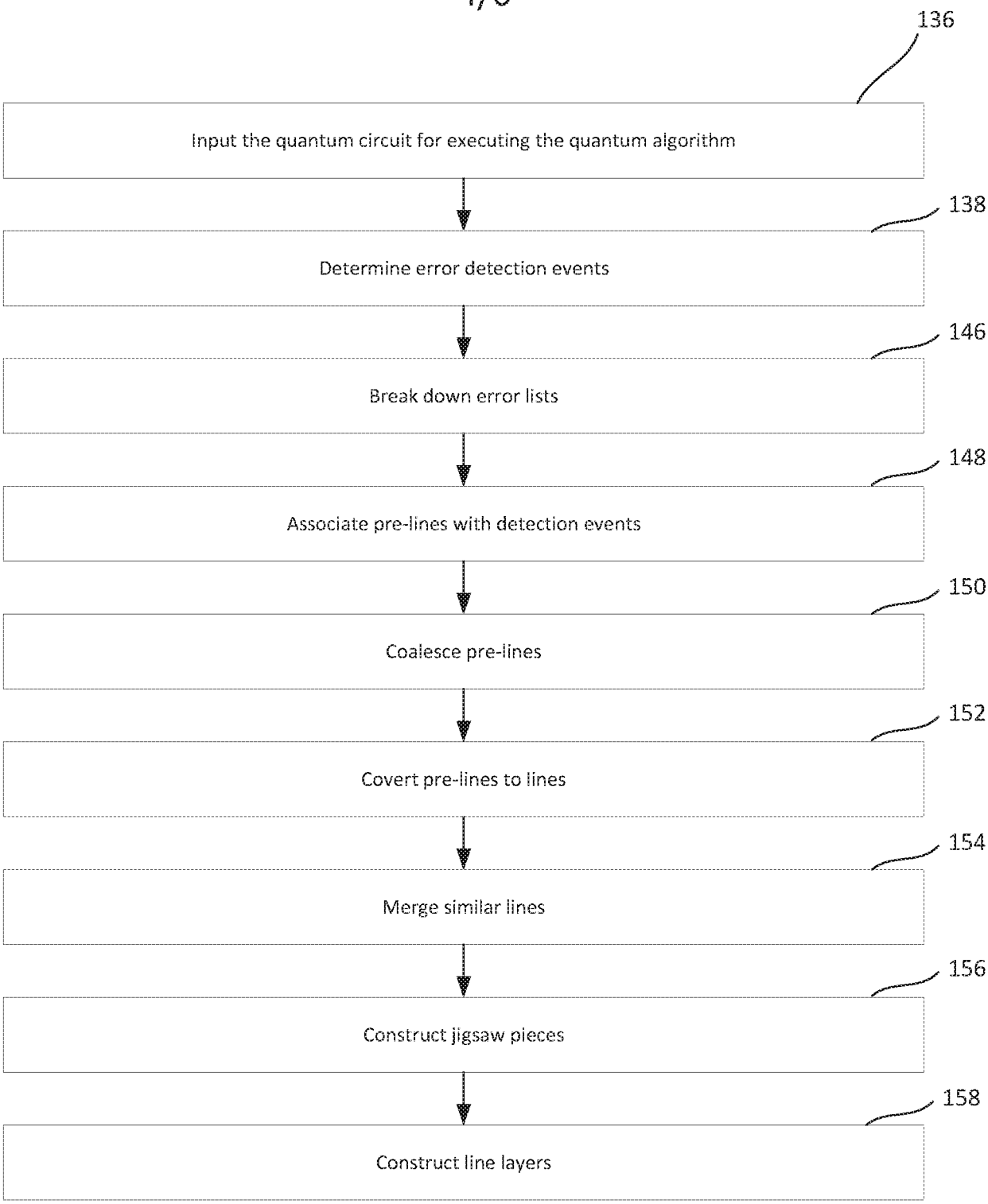

136
Input the quantum circuit for executing the quantum algorithm
138
Determine error detection events
146
Break down error lists
148
Associate pre-lines with detection events
150
Coalesce pre-lines
152
Covert pre-lines to lines
154
Merge similar lines
156
Construct jigsaw pieces
158
Construct line layers

Fig. 4

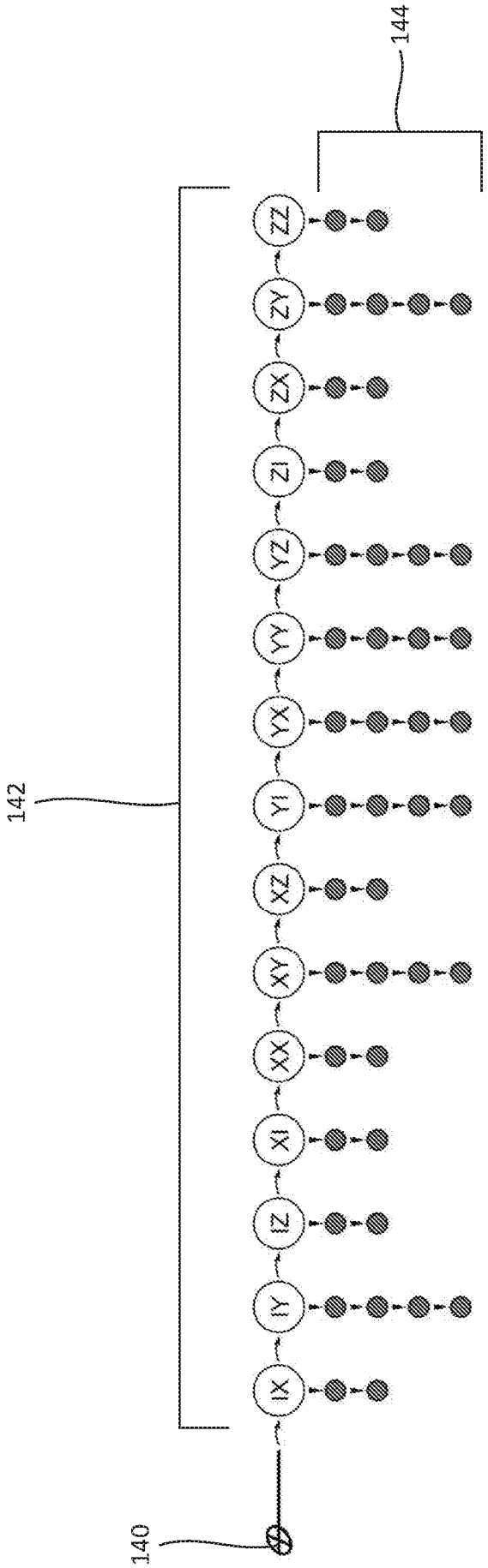


Fig. 5

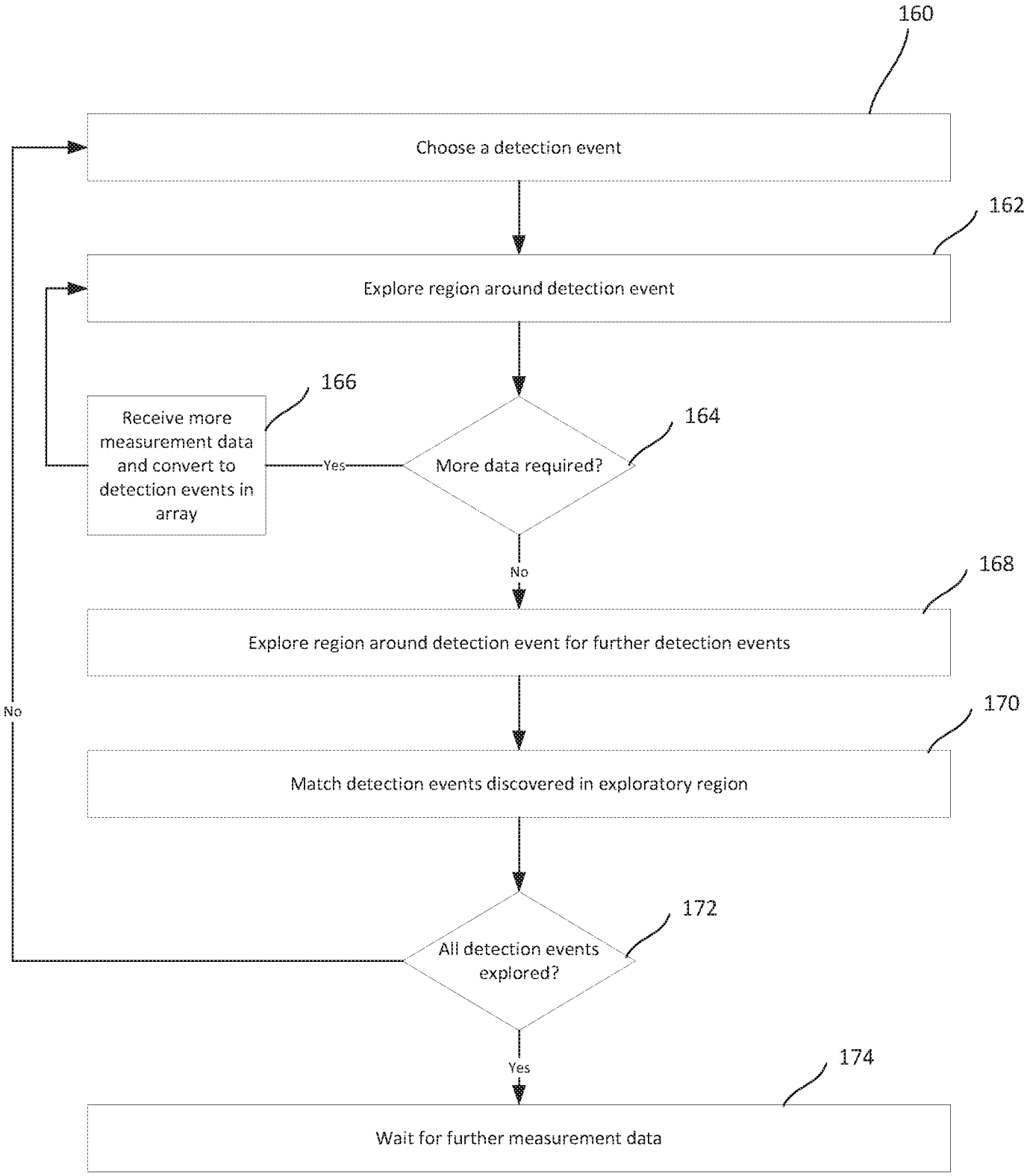
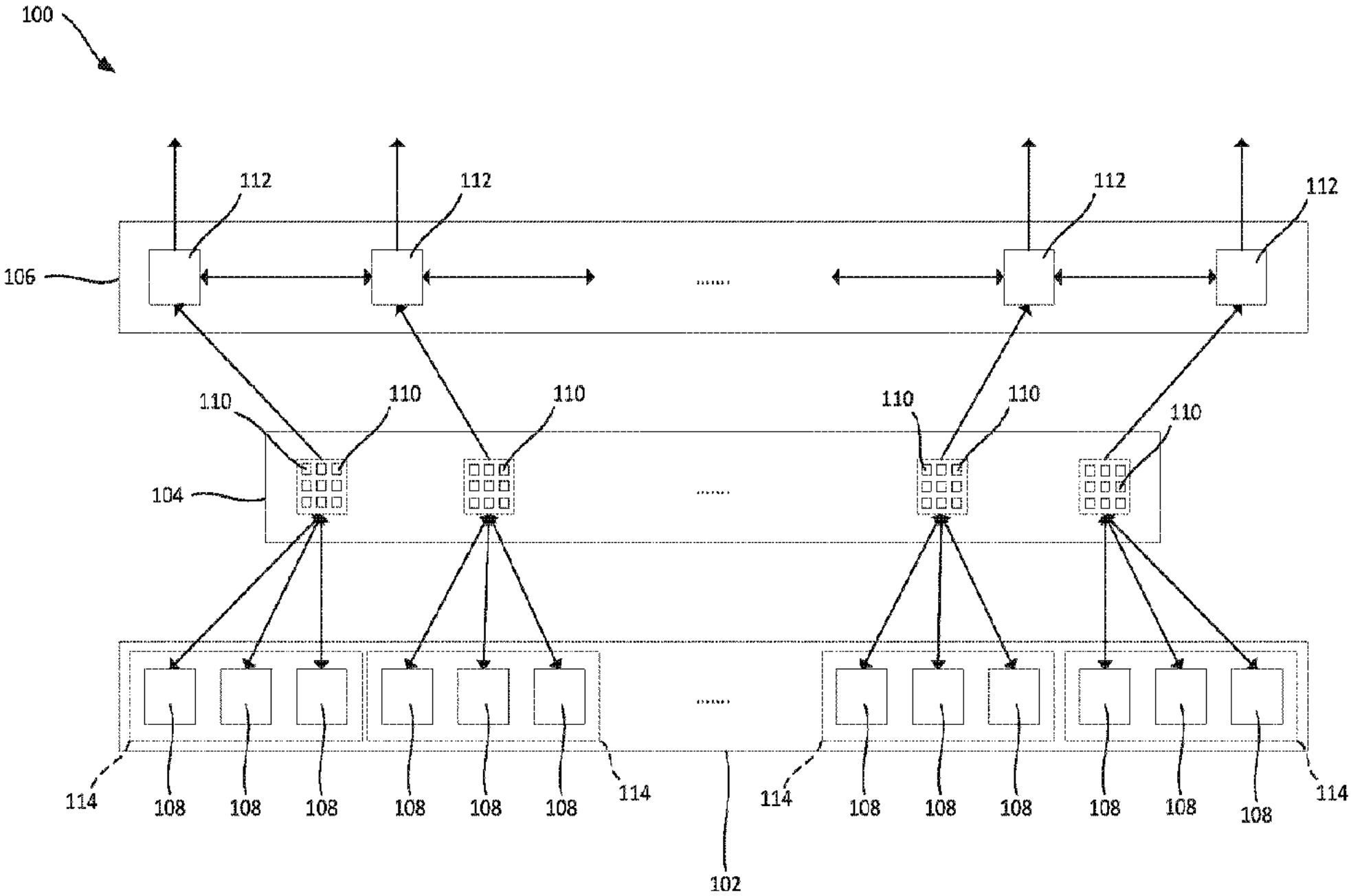

160
Choose a detection event
162
Explore region around detection event
166
Receive more measurement data and convert to detection events in array
Yes
164
More data required?
No
168
Explore region around detection event for further detection events
170
Match detection events discovered in exploratory region
172
All detection events explored?
No
Yes
174
Wait for further measurement data

Fig. 6


100
112
112
112
112
106
110
110
110
110
110
110
104
114
108
108
108
108
108
108
114
114
108
108
108
108
108
108
114
102