Trimming Decoding of Color Codes over the Quantum Erasure Channel

Sangjun Lee, Mehdi Mhalla, Valentin Savin

Trimming Decoding of Color Codes over the Quantum Erasure Channel

Sangjun Lee[1] Mehdi Mhalla[2] Valentin Savin[1]

[1] CEA-LETI, MINATEC, F-38054 Grenoble, France. Email: {sangjun.lee, valentin.savin}@cea.fr

[2] Univ. Grenoble Alpes, CNRS, Grenoble INP, LIG, F-38000 Grenoble, France. Email: mhallam@univ-grenoble-alpes.fr

***Abstract*—We propose a decoding algorithm for color codes over the quantum erasure channel, which is linear-time maximum likelihood (ML) when the set of erased qubits satisfies a certain condition called trimmability. Two methods are proposed for general erasure sets, either by extending the erasure set to make it trimmable, or by inactivating some vertices. The former is linear time but not ML, while the latter is ML but not linear time. Numerical results are provided to assess the error correction performance and the complexity of both methods.**

I. Introduction

Error correction is a crucial step for the construction of a quantum computer. Quantum systems suffer from errors due to decoherence and noise. By using quantum error correction, one can prevent quantum information in a quantum computing device from being destroyed. Many efforts and improvements have been made to develop and study quantum error correction codes. Among them, topological codes like surface codes [1], [2] are expected to be deployed to build practical quantum computers due to their high thresholds and locality [3].

Color codes [4] are other promising topological quantum error correction codes for fault-tolerant quantum computing. They provide relatively good thresholds that are slightly below the ones of surface codes [5], [6], [7]. However, unlike surface codes, transversal Clifford operations can act as logical Clifford operations [8].

The quantum erasure channel [9], [10] is the simple noise model, in which some qubits are erased and we are given which qubits are erased. When a qubit is erased, that qubit is thought to be affected by a randomly chosen Pauli error. Having information which qubits are erased might make developing decoding algorithms less complicated. Recently, maximum likelihood (ML) decoding of surface codes in linear time over the quantum erasure channel has been proposed [11], and it is used as a subroutine for almost-linear time decoding algorithm of surface codes and color codes [6], by projecting them onto surface codes [12], [7] to correct both Pauli errors and erasures.

In this paper, we show that linear-time ML decoding of color codes over the quantum erasure channel is possible, when a set of erased qubits satisfies a certain *trimmability* condition, and propose a decoding algorithm, which we call trimming decoding. We also provide the ways how to use the trimming decoding when the trimmability constraint is not obeyed.

The paper is structured as follows. In Section II, we provide background knowledge such as the definitions of color codes, the shrunk lattices and the string operators, and the quantum erasure channel. In Section III-A, we first define *trimmable* erasure sets, and propose a linear time ML decoding for such erasure sets. Two extensions are then proposed for general erasure sets, the first by extending the erasure set to make it trimmable (linear time but not ML), and the second by inactivating some vertices (ML but not linear time). Finally in Section IV, we provide the simulation results to assess the error correction performance and the complexity of both methods.

II. Preliminaries

A. Color codes

Given a graph $(\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is the set of vertices and $\mathbf{E}$ the set of edges, a tiling of a surface $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{F})$ is the embedding of the graph on a surface, where $\mathbf{F}$ is the set of faces. Color codes [4] are defined by 3-colorable tilings of closed surfaces (all surfaces are assumed to be orientable, of arbitrary genus). A tiling is said to be 3-colorable if its faces can be colored with 3 colors (say red, green, blue), such that each vertex is incident to exactly one face of each color. Let $\mathcal{P}$ be the Pauli group on $|\mathbf{V}|$ qubits, and $\mathcal{S} \leqslant \mathcal{P}$ be the subgroup generated by the Pauli operators $S_X(f) = \otimes_{v \in \mathbf{V}(f)} X_v$, and $S_Z(f) = \otimes_{v \in \mathbf{V}(f)} Z_v$, times the identity on the remaining vertices, for each face $f \in \mathbf{F}$, where $\mathbf{V}(f)$ is the set of vertices incident to f. The color code associated with the tiling $\mathbf{G}$ is the stabilizer code defined by $\mathcal{S}$. It encodes a number of logical qubits, k, given by [4]:

$$k = 4 - 2\chi = 4g \tag{1}$$

where $\chi := |\mathbf{V}| - |\mathbf{E}| + |\mathbf{F}|$ denotes the Euler characteristic of the surface, and g its genus. Since g is a topological invariant of the surface, k does actually only depend on the surface, and not on the specific tiling, the latter determining however the minimum distance of the code (or more generally, the spectrum of weights of undetectable errors, see below).

A Pauli error is simply a Pauli operator E acting on the $|\mathbf{V}|$ qubits. Since operators in $\mathcal{S}$ act trivially on the code space, we shall assume that $E \in \mathcal{P} \setminus \mathcal{S}$. Omitting a global phase term, we may write $E = \otimes_{v \in \mathbf{V}} E_v \in \{I, X, Y, Z\}^{\otimes |\mathbf{V}|}$, and define the *support* and the *weight* of E, as $\text{supp}(E) = \{v \in \mathbf{V} | E_v \neq I\}$ and $w(E) = |\text{supp}(E)|$. A Pauli error E is said to be *undetectable* if it belongs to $\mathcal{Z} \setminus \mathcal{S}$, where $\mathcal{Z}$ denotes the centralizer of $\mathcal{S}$ in $\mathcal{P}$. The minimum distance of the code is defined as the smallest weight of an undetectable error.

As explained in [4], there is a close connection between the centralizer group $\mathcal{Z}$ and the one-dimensional homology group H_1 of the surface, which can be best understood by considering cycles in the *shrunk lattices* associated with $\mathbf{G}$. This connection is briefly explained below, as it is relevant to the purposes of this paper.

First, we note that any edge $e \in \mathbf{E}$ borders two faces $f_1, f_2 \in \mathbf{F}$ that must have different colors $c_1 \neq c_2 \in \{r, g, b\}$. We color e with the unique color c that is different from both c_1 and c_2. Moreover, let v', v'' denote the two endpoint vertices of e, and $f', f'' \in \mathbf{F}$ be the two faces such that v' is incident to f_1, f_2, f', and v'' is incident to f_1, f_2, f''. Hence, one may think at the edge e as *connecting* the faces f' and f'', and these two faces are necessarily of the same color c. The *c-color shrunk lattice* is obtained by shrinking each face of color c to a vertex, which will be referred to in the sequel as a *site* (to avoid confusion with the vertices in $\mathbf{V}$), and connecting them through edges of the same color. Let Γ be a cycle in the c-color shrunk lattice: it is a closed path, consisting of a sequence of edges that join a sequence of sites. Since the edges of the shrunk lattice are actually the edges in $\mathbf{E}$, of color c, we may define $\mathbf{V}(\Gamma)$ as the set of vertices in $\mathbf{V}$ incident to the edges that Γ contains. We define the following two operators, referred to as *string operators* [4]:

$$S_{\Gamma}^{c,\sigma} = \bigotimes_{v \in \mathbf{V}(\Gamma)} \sigma_v, \quad \text{for } \sigma \in \{X, Z\} \tag{2}$$

where σ_v denotes the σ operator on qubit v. It is easily seen that string operators commute with the generators of $\mathcal{S}$, and thus they belong to $\mathcal{Z}$. Moreover, $S_{\Gamma'}^{c,\sigma}$ and $S_{\Gamma''}^{c,\sigma}$ belong to the same equivalence class in the quotient group $\mathcal{Z}/\mathcal{S}$ (hence, they have the same effect on the code space) if and only if Γ' and Γ'' belong to the same equivalence class in the homology group H_1. Hence, we may further define $S_{\gamma}^{c,\sigma} \in \mathcal{Z}/\mathcal{S}$, for any homology class $\gamma \in H_1$, and the following properties hold [4].

$$S_{\gamma}^{r,\sigma} S_{\gamma}^{g,\sigma} S_{\gamma}^{b,\sigma} = 1 \tag{3}$$

$$\left[S_{\gamma}^{c,\sigma}, S_{\tau}^{c',\sigma}\right] = \left[S_{\gamma}^{c,\sigma}, S_{\gamma}^{c',\sigma'}\right] = \left[S_{\gamma}^{c,\sigma}, S_{\tau}^{c,\sigma'}\right] = 0 \tag{4}$$

Using (3), and since H_1 is generated by $2g$ homology classes, say $\gamma_1, \ldots, \gamma_{2g}$, it follows that $\mathcal{Z}/\mathcal{S}$ is generated by $\left\{S_{\gamma_i}^{c,\sigma} \mid i = 1, \ldots, 2g, c \in \{r, g\}, \sigma \in \{X, Z\}\right\}$. This gives a set of $8g = 2k$ generators, which may be identified to k logical-X and k logical-Z operators.

B. Channel model and decoding

The channel we consider is the quantum erasure channel [11]: each qubit independently has a probability p_e of being erased. The set of erased qubits $\mathcal{E}$ is known and the erased qubits are replaced by a totally mixed state $\mathcal{I}/2 = \frac{1}{4}(\rho + X\rho X + Y\rho Y + Z\rho Z)$, which can be viewed as a uniform random Pauli error. Thus, we denote by $E = \otimes_{v \in \mathbf{V}} E_v \in \mathcal{P}$, such that $\text{supp}(E) \subseteq \mathcal{E}$, the error that actually occurred. For a Pauli error E, each face $f \in \mathbf{F}$ has syndrome bits $\mathbb{S}_\sigma(f) \in \{\pm 1\}$, with $\sigma \in \{X, Z\}$, such that $\mathbb{S}_\sigma(f) = 1$ if E and $S_\sigma(f)$ commute, and $\mathbb{S}_\sigma(f) = -1$, if they anticommute.

Lemma 1. *Let E' and E'' be two Pauli errors, with syndrome bits denoted by $\mathbb{S}'_\sigma(f)$ and $\mathbb{S}''_\sigma(f)$, respectively. Then,*

(i) *$E'E'' \in \mathcal{Z}$ if and only if E' and E'' have the same syndrome, that is $\mathbb{S}'_\sigma(f) = \mathbb{S}''_\sigma(f), \forall f \in \mathbf{F}, \forall \sigma \in \{X, Z\}$.*

(ii) *$E'E'' \in \mathcal{S}$ if and only if E' and E'' have the same syndrome and $\text{supp}(E'E'')$ does not contain any homologically non-trivial cycle (i.e., a cycle in some shrunk lattice, with non-trivial homology class).*

Proof. (i) follows from the fact that $E'E'' \in \mathcal{Z}$ if and only if $E'E''$ commutes with all $S_\sigma(f)$, and therefore $\mathbb{S}'_\sigma(f) = \mathbb{S}''_\sigma(f), \forall f \in \mathbf{F}, \forall \sigma \in \{X, Z\}$. (ii) follows from the fact that any operator in $\mathcal{Z}$ whose support does not contain a homologically non-trivial cycle, must be in $\mathcal{S}$ [4]. □

ML decoding amounts to determining all the errors E' having the same syndrome as E, and with support $\text{supp}(E') \subseteq \mathcal{E}$. Errors E' satisfying the above properties are equally likely to occur, so that ML decoding may output one randomly chosen error from the above set of solutions. ML decoding is successful if all such errors are equivalent to E, up to stabilizers, which happens if and only if $\mathcal{E}$ does not contain homologically non-trivial cycles (Lemma 1). In case $\mathcal{E}$ contains homologically non-trivial cycles, ML decoding fails with probability $1 - 2^a/2^b \geq 1/2$, where b the dimension of the space of solutions, and a is the dimension of the linear subspace of solutions equivalent to E. Hence, we shall say that an erasure set $\mathcal{E}$ containing homologically non-trivial cycles is *undecodable*.

ML decoding can be implemented in $\mathcal{O}(|\mathbf{V}|^3)$ by using Gaussian elimination (this complexity can be decreased $\mathcal{O}(|\mathbf{V}|^{2.367})$ by using more sophisticated algorithms). This cost may become prohibitive for large codes, in which case decoding methods with linear computational complexity are desirable. One such approach is the *peeling decoding*, which tries to solve the decoding problem by recursively searching for faces with only one incident erased vertex (see Section III-A). The trimming decoding algorithm we propose in this paper extends the peeling decoding, so as to cope with cases where there is no longer any face with only one incident erased vertex. The main idea is to "force" a guess on some vertices, and then reactivate the peeling decoding again.

The following lemma will be needed to prove the validity of the trimming algorithm.

Lemma 2. *Let Γ be a cycle in the shrunk lattice of color c, such that $\mathbf{V}(\Gamma) \subset \mathcal{E}$, and let $v \in \mathbf{V}(\Gamma)$. Then, there exists a Pauli error E' with the same syndrome as E, such that $\text{supp}(E') \subset \mathcal{E}$ and $E'_v = I$.*

Proof. If $E_v = I$, then $E' = E$ holds. Suppose that $E_v \neq I$. Let $S_{\Gamma}^{c,\sigma}$, be the string operator associated with Γ. Then $E' = S_{\Gamma}^{c,\sigma} E$, with $\sigma = E_v$, satisfies $E'_v = I$. From Lemma 1, E' and E have the same syndrome, since $E'E = S_{\Gamma}^{c,\sigma} \in \mathcal{Z}$. □

III. Trimming Decoding

A. Decoding for trimmable erasure sets

Consider a color code state transmitted over the quantum erasure channel. Let $\mathcal{E}$ denote the erasure set and E the actual error that occurred, with $\mathrm{supp}(E) \subseteq \mathcal{E}$, and $\mathbb{S}(E) = (\mathbb{S}_\sigma(f) \in \{\pm 1\} \mid f \in \mathbf{F}, \sigma \in \{X, Z\})$ be the syndrome of E.

By a slight abuse of notation, in the following we shall denote by $\mathcal{E}$ both the erasure set and the induced subgraph (the edges in the induced subgraph are the edges $(u, v) \in \mathbf{E}$ with both incident verticies $u, v \in \mathcal{E}$). For any face $f \in \mathbf{F}$, let $\mathcal{E}(f)$ denote the set of vertices incident to f that belong to $\mathcal{E}$.

Definition 3. *An erasure set $\mathcal{E}$ is said to be* trimmable *if for any $f \in \mathbf{F}$, the vertices in $\mathcal{E}(f)$ belong to the same connected component of $\mathcal{E}$.*

Put differently, the above definition means that any two vertices in $\mathcal{E}(f)$ can be connected by a path on $\mathcal{E}$. However, it is worth noticing that such a path needs not be in $\mathcal{E}(f)$, that is, $\mathcal{E}(f)$ needs not be connected.

Let $\mathcal{T}$ be a *spanning tree* of a connected component of $\mathcal{E}$. Hence, $\mathcal{T}$ is a subgraph that is a tree and includes all the vertices of the considered connected component. If $\mathcal{E}$ is not connected, a *spanning forest* $\mathcal{F}$ is a graph consisting of a spanning tree for each of its connected components.

Our decoding algorithm is described in Algorithm 1. It starts by constructing a spanning forest $\mathcal{F}$ of $\mathcal{E}$. For instance, such a spanning forest can be effectively determined by running the Depth-First Search (DFS) algorithm on $\mathcal{E}$. Then, the decoding procedure is very simple: it suffices to run through the leaves (vertices of degree 0 or 1 in $\mathcal{F}$) of the spanning forest, and to either peel the current leaf vertex, whenever this is possible, or force a guess on it, otherwise.

We say that a vertex v *can be peeled* if it has at least one incident face f that is not incident to any other vertices in $\mathcal{E}$. In such a case, *peeling* v means that the value of E'_v is set according to the syndrome bits of f. Precisely, we set $E'_v = X^{s_X(f)} Z^{s_Z(f)}$ where $s_\sigma(f) := \frac{1 - \mathbb{S}_\sigma(f)}{2} \in \{0, 1\}$, $\sigma \in \{X, Z\}$. Moreover, we shall assume that syndrome bits of the faces incident to v are updated.

Algorithm 1 Decoding for trimmable erasure sets

Input: A trimmable erasure $\mathcal{E} \subset \mathbf{V}$, and an error syndrome $\mathbb{S}$
Output: A Pauli error E', with $\mathrm{supp}(E') \subseteq \mathcal{E}$ and $\mathbb{S}(E') = \mathbb{S}$

```
1:  Determine a spanning forest F of E;
2:  while F non empty do
3:      v ← leaf node of F;
4:      if v can be peeled then
5:          Determine E'_v by peeling v;
6:      else
7:          Set E'_v = I;
8:      end if
9:      Remove v from F;
10: end while
```

Proposition 4. *Assume that $\mathcal{E}$ is a trimmable erasure set and $\mathcal{F}$ is a spanning forest of $\mathcal{E}$. Then Algorithm 1 determines a valid Pauli error E', that is, an error such that $\mathrm{supp}(E') \subseteq \mathcal{E}$ and $\mathbb{S}(E') = \mathbb{S}$.*

Proof. We have to prove that each time in Algorithm 1 we need to force the identity on some vertex v (line 7), then there exists indeed a valid error E', such that $E'_v = I$.

We distinguish several cases, according to the $\deg_{\mathcal{E}}(v)$ value, the degree of v in the subgraph induced by $\mathcal{E}$. We denote by $\mathcal{T}$ the spanning tree of $\mathcal{F}$ containing v.

Case 1) $\deg_{\mathcal{E}}(v) = 0$. This may only happen if there is only one vertex left in the tree, that is, $\mathcal{T} = \{v\}$. In this case, v can be peeled, by using any of its incident faces. Indeed, let f be any face incident to v. If there were other erased vertices incident to f, they would necessarily belong to another connected component (since $\mathcal{T}$ contains only v). However, this is impossible, because of the trimmable property of $\mathcal{E}$, which implies that the erased vertices incident to the same face f belong to the same connected component.

Case 2) $\deg_{\mathcal{E}}(v) = 1$. Let us assume that v cannot be peeled (since otherwise, there is nothing to prove). We know that v is connected to exactly one vertex in $\mathcal{E}$, which must be its parent node in $\mathcal{T}$. Let u be the parent node of v in $\mathcal{T}$, and $u_1, u_2 \in \mathbf{V} \setminus \mathcal{E}$ be the other two neighbor vertices of v (see figure 1). Denote by f the unique face incident to v, u_1, and u_2. Finally, let c be the color of f. Since v cannot be peeled, there must be another vertex incident to f that belongs to $\mathcal{E}$. Call this vertex $w_1 \in \mathcal{E}$. By the definition of a trimmable set, w_1 and v belong to the same connected component of $\mathcal{E}$. It follows that $w_1 \in \mathcal{T}$, and there must be a path $p = (w_1, w_2, \ldots, w_m)$ in $\mathcal{T}$, such that $w_m = v$. In particular, it follows that $w_{m-1} = u$, the parent of v in $\mathcal{T}$, and the edge (u, v) is of the same color as f, say c. The edges of p of color c determine a cycle Γ in the shrunk lattice such that $v \in \mathbf{V}(\Gamma)$. Therefore, we may apply Lemma 2, which states that there exists a valid error E' such that $E'_v = I$.

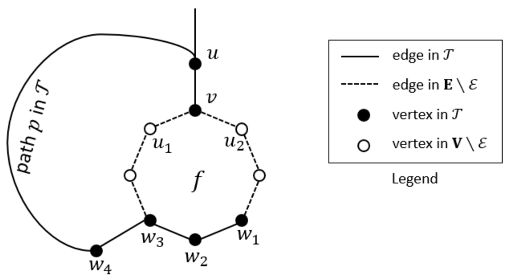

Fig. 1. Proof of the trimming procedure – case 2) $\deg_{\mathcal{E}}(v) = 1$

Case 3) $\deg_{\mathcal{E}}(v) \geq 2$. In this case, it is easily seen that any face incident to v must be incident to at least another vertex in $\mathcal{E}$. Hence, v cannot be peeled and we need to prove that there exists a valid error E' such that $E'_v = I$. Since $\mathcal{T}$ contains only one edge incident to v, say (u, v) where u is the parent node of v in $\mathcal{T}$, there must be an edge incident to v in $\mathcal{E}$, but which is not in $\mathcal{T}$. Since $\mathcal{T}$ is a spanning tree of a connected component of $\mathcal{E}$, such an edge determines a (fundamental) cycle C of $\mathcal{E}$, passing through v. Let Γ be the cycle in the shrunk lattice determined by the edges of C of the same color as (u, v). Then, from Lemma 2, we may find a valid error E' such that $E'_v = I$. $\square$

The trimming decoding algorithm is ML over the trimmable erasure set since all possible errors are equiprobable, and its complexity is linear in the number of vertices. Indeed, the spanning forest $\mathcal{F}$ can be determined by running a DFS on the subgraph induced by $\mathcal{E}$, which takes linear time. Trimming leaf nodes is a linear-time procedure since each leaf node is considered only once, and it is either peeled or the identity is forced on it.

Two ways to extend the previous algorithm to general erasure sets are proposed. The first method is to connect some spanning trees that violate the trimming property, so that the extended erasure set becomes trimmable. The second method is to get rid of vertices that violate the trimming property, and their errors are calculated later by solving a small linear system.

B. Trimming decoding with extended erasure set

Let $\mathcal{F}$ be a spanning forest of $\mathcal{E}$. Each tree in $\mathcal{F}$ corresponds to a connected component of the subgraph induced by $\mathcal{E}$. In order to have a trimmable erasure set, if different trees share a same face, we connect them by inserting *pseudo-erasures* between them on the border of the shared face. We recursively repeat the above procedure until we get an extended erasure set $\bar{\mathcal{E}} \supset \mathcal{E}$ that is trimmable. We then apply Algorithm 1 to $\bar{\mathcal{E}}$.

The decoding complexity remains linear, however inserting pseudo-erasures may create homologically non trivial cycles in $\bar{\mathcal{E}}$ (that are non in $\mathcal{E}$), thus degrading the error correcting performance with respect to ML decoding.

C. Trimming decoding with inactivation

The other way is to inactivate vertices, which means that the errors on some vertices are assumed to be known and the vertices are removed from the erasure set. The trimming decoding with inactivating is described in Algorithm 2. Because the erasure set may not be trimmable, there are three cases to consider. The first two cases have been already discussed for the trimming algorithm, when the leaf can either be peeled, or its pendant face (face f in Figure 1) is touched by only one spanning tree. The new case is when some vertices incident to the pendant face of the leaf do not belong to the spanning tree to which the leaf belongs. In this case v is inactivated (that is, a variable is assigned to E'_v, which is pretended to be known), and the trimming decoding procedure is continued. After all the vertices are trimmed or inactivated, a linear system is constructed with the variables assigned to the inactivated vertices. Any linear system solver such as Gaussian elimination returns a valid solution.

The trimming decoding with inactivation is ML, but its complexity is dominated by the resolution of the linear system. Therefore the algorithm is not linear time, however numerical results (see SectionIV) show that the number of inactivated vertices is generally small with respect to the total number of vertices.

Algorithm 2 Trimming decoding with inactivation

Input: A trimmable erasure $\mathcal{E} \subset \mathbf{V}$, and an error syndrome $\mathbb{S}$
Output: A Pauli error E', with $\mathrm{supp}(E') \subseteq \mathcal{E}$ and $\mathbb{S}(E') = \mathbb{S}$

```
Determine a spanning forest F of E;
while F non empty do
    v ← leaf node of F;
    if v can be peeled then
        Determine E'_v by peeling v;
    else if the pendant face of v is touched by only one
    spanning tree then
        Set E'_v = I;
    else
        Inactivate v;
    end if
    Remove v from F;
end while
Solve the linear system determined by inactivated vertices;
```

IV. Numerical Results

We first show the result of the trimming decoding algorithm with the erasure set extension in the hexagonal lattice on the torus. In Fig. 2, the simulation results are shown by plotting the logical X error rates p_{l_X} against the erasure rates p_e with several minimum distances d. The trimming decoding with the erasure set extension gives a threshold of 43% in the hexagonal lattice on the torus. The threshold obtained using the trimming algorithm on the extended erasure set is close to the theoretical limit of the quantum erasure channel since the number of the pseudo-erasures is not relatively substantial and the trimming algorithm is ML for the extended trimmable erasure set.

The peeling decoding solely provides the very poor error correction performance compared to the one of the trimming decoding. See the dashed line in Fig. 2 for the peeling

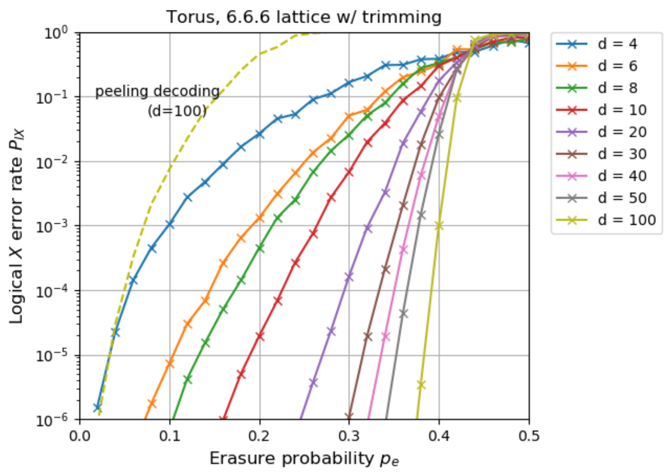

Fig. 2. Trimming decoding with extension in the hexagonal lattice on the torus

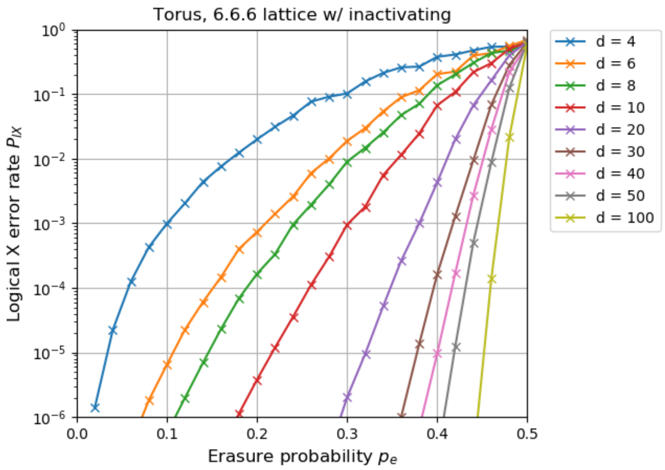

Fig. 3. Trimming decoding with inactivation in the hexagonal lattice on the torus

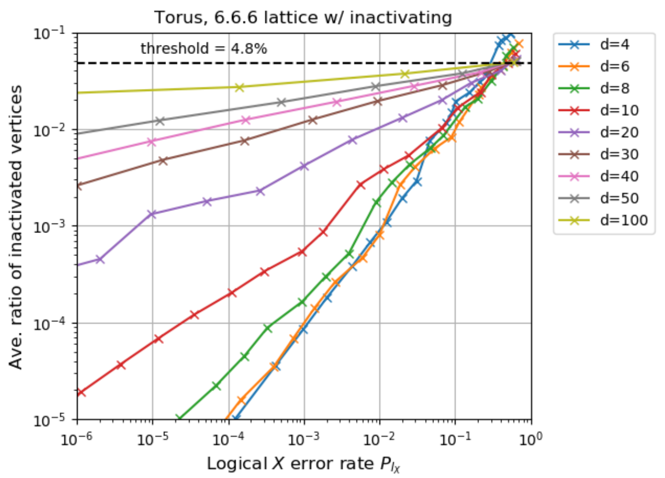

Fig. 4. The average numbers of the inactivated vertices divided by the total number of the vertices for trimming decoding with inactivation in the hexagonal lattice on the torus

decoding when the minimum distance d is 100. Note that the peeling decoding cannot peel any erasure set that covers a face border. As a consequence, the logical error rate of the peeling decoding is lower-bounded by p_e^6, and thus the error correction threshold is zero.

For the trimming decoding with the inactivation, the results in the hexagonal lattice on the torus are shown in Fig. 3. The threshold is obviously 50%, since the trimming decoding with the inactivation is ML.

In order to evaluate the time complexity of the trimming decoding with the inactivation in the practical regions, the ratios of the average numbers of the inactivated vertices to the total number of the vertices for the various minimum distance are calculated, which are shown in Fig. 4. As the minimum distance increases, the average ratios of the inactivated vertices when the logical X error rate P_{l_X} is 10^{-6} tend to converge. Thus, when the minimum distance is very large, it is expected that only around 4.8% of the vertices are inactivated, as shown in Fig. 4. Since the number of the variables of the linear system is relatively low, the decoding procedure using the trimming algorithm with the inactivation can be done fast compared to solving the linear system for all the vertices.

V. Conclusion

We proposed the trimming decoding for color codes over the quantum erasure channel. If the erasure set is trimmable, then the trimming algorithm is linear-time ML. For the erasure set that is not trimmable, extending the erasure set and inactivating some vertices were provided to exploit the trimming property. The trimming algorithm with extending the erasure set is of linear time, but ML for the extended erasure set, not the original erasure set. The trimming decoding with the inactivation is ML, but the time complexity is not linear since the linear system needs to be solved. The decoding algorithm discussed in this paper has the potential to be used for decoding of color codes over the depolarizing channel by extending the erasure set as discussed in [6].

Acknowledgment

This work is partly funded by the French Public Authorities through Carnot Quantique program.

References

[1] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," *Annals of Physics*, vol. 303, no. 1, pp. 2–30, 2003.

[2] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4452–4505, 2002.

[3] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Physical Review A*, vol. 86, no. 3, p. 032324, 2012.

[4] H. Bombin and M. A. Martin-Delgado, "Topological quantum distillation," *Phys. Rev. Lett.*, vol. 97, no. 18, p. 180501, Oct 2006.

[5] A. J. Landahl, J. T. Anderson, and P. R. Rice, "Fault-tolerant quantum computing with color codes," *arXiv preprint arXiv:1108.5738*, 2011.

[6] N. Delfosse and N. H. Nickerson, "Almost-linear time decoding algorithm for topological codes," *arXiv preprint arXiv:1709.06218*, 2017.

[7] A. Kubica and N. Delfosse, "Efficient color code decoders in $d \geq 2$ dimensions from toric code decoders," *arXiv preprint arXiv:1905.07393*, 2019.

[8] A. Kubica and M. E. Beverland, "Universal transversal gates with color codes: A simplified approach," *Physical Review A*, vol. 91, no. 3, p. 032330, 2015.

[9] M. Grassl, T. Beth, and T. Pellizzari, "Codes for the quantum erasure channel," *Physical Review A*, vol. 56, no. 1, p. 33, 1997.

[10] C. H. Bennett, D. P. DiVincenzo, and J. A. Smolin, "Capacities of quantum erasure channels," *Physical Review Letters*, vol. 78, no. 16, p. 3217, 1997.

[11] N. Delfosse and G. Zémor, "Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel," *arXiv preprint arXiv:1703.01517*, 2017.

[12] N. Delfosse, "Decoding color codes by projection onto surface codes," *Physical Review A*, vol. 89, no. 1, p. 012317, 2014.