Robust positioning patterns

Ross Berkowitz* Swastik Kopparty†

October 14, 2015

Abstract

In this paper, we construct large sequences and matrices with the property that the contents of any small window determine the location of the window, *robustly*. Such objects have found many applications in practical settings, from positioning of wireless devices to smart pens, and have recently gained some theoretical interest.

In this context, we give the first explicit constructions of sequences and matrices with high rate and constant relative distance. Accompanying these efficient constructions, we also give efficient decoding algorithms, which can determine the position of the window given its contents, even if a constant fraction of the contents have been corrupted.

1 Introduction

A 1-dimensional "positioning pattern" is a sequence of N symbols from some alphabet, with the property that any window of n consecutive elements from the sequence uniquely determines the position of the window. Similarly, a 2-dimensional "positioning pattern" is an $N \times N$ matrix of symbols from some alphabet, with the property that any $n \times n$ (contiguous) window of elements uniquely determines the position of the window. Positioning patterns have been classically studied in combinatorics under various names: de Bruijn sequences, perfect maps, pseudorandom sequences and arrays, etc. In recent years, these objects have found a number of useful real-world applications, such as robot localization [Sch01], camera localization [SZH+12], the Echo Smartpen, and smart stylus' [sty].

To see the utility of positioning patterns, let us briefly describe the application from [sty]. We are given a display device (such as a monitor or a laptop screen) whose sole capability is display (in particular, it cannot detect touch or the presence of a stylus/pen). The smart stylus from [sty] is based on a combination of software and hardware, and converts any such display into one which can take input from the stylus. The hardware component is a pen with a small camera at its nib, which when brought near the screen of the display device can view a small $n \times n$ window of the screen. The software component sets the lower order bits of the color attribute for each pixel on the screen according to a positioning pattern. This ensures that the lower order bits for any $n \times n$ window of the screen uniquely determines the position of the window: thus one can use the image from the pen camera to determine the location of the pen, and this is just as good as having a display that can detect the location of an associated stylus.

A "robust positioning pattern" is a sequence/matrix of symbols, which allows such position determination by reading a small window from the pattern *even if some errors occur while reading the small window*. Concretely, the sequence/matrix has the property that the contents of the different windows should be far apart from each other in Hamming

*Department of Mathematics, Rutgers University. berkowitz@math.rutgers.edu

†Department of Mathematics & Department of Computer Science, Rutgers University. Supported in part by a Sloan Fellowship and NSF grant CCF-1253886. swastik@math.rutgers.edu

distance. Algorithmically, we would like to be able to *efficiently decode* the position of the window, given the corrupted contents of a window.

We are interested in constructing such robust positioning patterns and designing associated decoding algorithms for them. Our motivation comes from both practice and theory. Firstly, these problems are naturally motivated by the applications of positioning patterns given above, which rely on physical devices and are thus prone to error. Secondly, this topic presents interesting combinatorial and algorithmic challenges at the confluence of error-correcting codes and combinatorial sequence design, both of which are extensively studied and have highly developed theories.

Our main results give explicit constructions of robust positioning patterns, along with associated decoding algorithms. These constructions are the first to achieve constant rate while being robust to a constant fraction of errors, and are also the first to achieve robustness to a constant number of errors with redundancy within a constant factor of optimal.

1.1 Results

We begin with the 1 dimensional setting.

Let $\sigma \in \Sigma^N$ be a string. We let $\sigma[i,j)$ denote the substring $\sigma_i\sigma_{i+1}\ldots\sigma_{j-1}$. We will be interested in substrings of the form $\sigma[i,i+n)$, which we will also call the "windows of length n". We define the window-n distance of σ to equal the minimum, over distinct $i,j \in [N-n+1]$ of

$$\Delta(\sigma[i,i+n),\sigma[j,j+n)),$$

where Δ denotes the Hamming distance.

The basic combinatorial problem here is to determine the length of the longest string with window-n distance at least d. The basic algorithmic problems here are: (1) **Encoding:** to explicitly construct a long string with window-n distance at least d, and (2) **Decoding:** for this sequence, given a "received string" $r \in \Sigma^n$ which is within distance e of some window $\sigma[i,i+n-1]$, to find i.

It is sometimes convenient to use the following terminology. Define the window-n rate of σ to equal $\frac{\log N}{n \log |\Sigma|}$. Define the window-$n$ relative distance to be the window-n distance divided by n.

It is clear that the length N of any sequence with window-n distance d cannot be more than the size of the largest error-correcting code $C \subseteq \Sigma^n$ with minimum distance d (since the n-windows of the sequence form such an error-correcting code). Thus we have the following rough upper bounds on the length of such a sequence:

1. for $d = \delta n$ (with $\delta > 0$ a constant), we have $N \leq |\Sigma|^{n(1-f(\delta))}$, for some function $f(\delta)$ that goes to 0 as δ goes to 0,
2. for $d = O(1)$, $|\Sigma|$ large, we have $N \leq \frac{|\Sigma|^n}{|\Sigma|^{\Omega(d)}}$,
3. for $d = O(1)$, $|\Sigma| = 2$, we have $N \leq \frac{2^n}{n^{\Omega(d)}}$.

A simple application of the Lovasz Local Lemma (suggested to us by Nathaniel Shar) shows that the above upper bounds on N are essentially tight (non-constructively); there exist strings in Σ^N matching the above bounds. A very nice result of Kumar and Wei [KW92] shows that a random irreducible Linear Feedback Shift Register Sequence matches the third of the above upper bounds with high probability (this result holds for all $d \leq \sqrt{n}$). It is natural to ask if we can match these bounds with explicit constructions and efficient decoding algorithms.

Our main results for 1-dimensional sequences give explicit constructions and efficient decoding algorithms for sequences, essentially matching the above parameters[1].

THEOREM 1. (1-DIMENSION, LARGE Σ, CONSTANT δ) *There exists an infinite sequence of n and alphabets Σ_n (with $|\Sigma_n| \leq O(n)$), such that for every $R \in (0,1)$, there is a sequence $\sigma \in \Sigma^{N_n}$ with:*

1. *the rate of σ is at least R,*
2. *the window-n relative distance of σ is at least $\max(1-3R,(1-R)/3) - o(1)$,*
3. *the i'th coordinate of σ can be computed in time $\mathrm{poly}(n)$,*
4. *n-windows of σ can be decoded from a constant fraction of errors in $\mathrm{poly}(n)$ time.*

This theorem follows from Theorem 6.

THEOREM 2. (1-DIMENSION, $|\Sigma| = 2$, CONSTANT δ) *There exists an infinite sequence of n such that for every $R \in (0,1)$, there is a sequence $\sigma \in \{0,1\}^{N_n}$ with:*

1. *the rate of σ is at least R,*
2. *the window-n relative distance of σ is at least $h(R) - o(1)$, (where $h(R) > 0$),*
3. *the i'th coordinate of σ can be computed in time $\mathrm{poly}(n)$,*

[1] We require a widely believed number theoretic conjecture to attain the third set of parameters.

4. *n-windows of σ can be decoded from a constant fraction of errors in* $\mathrm{poly}(n)$ *time.*

This theorem follows from Corollary 4.1.

THEOREM 3. (1-D, LARGE Σ, CONSTANT DISTANCE) *There exists an infinite sequence of n and alphabets Σ_n (with $|\Sigma_n| = O(n)$), such that for every constant d, there is a sequence $\sigma \in \Sigma_n^{N_n}$ with:*

1. $N_n \geq \frac{|\Sigma_n|^n}{|\Sigma_n|^{O(d)}}$,
2. *the window-n distance of σ is at least d,*
3. *the i'th coordinate of σ can be computed in time* $\mathrm{poly}(n)$,
4. *n-windows of σ can be decoded from $\Omega(d)$ errors in* $\mathrm{poly}(n)$ *time.*

This theorem follows from Theorem 6.

Our result for constant distance binary codes depends on the existence of suitable Mersenne-like primes. Such primes are widely believed to exist based on standard number theoretic heuristics.

Conjecture C: There exists a constant c and infinitely many n such that there exists a prime between $2^n - c \cdot n$ and $2^n - 1$.

Note that this conjecture would be implied by the existence of infinitely many Mersenne primes.

THEOREM 4. (1-D, $|\Sigma| = 2$, CONSTANT DISTANCE) *Assume conjecture C. There exists an infinite sequence of n such that for every constant d, there is a sequence $\sigma \in \{0,1\}^{N_n}$ with:*

1. $N_n \geq \frac{2^n}{n^{O(d)}}$,
2. *the window-n distance of σ is at least d,*
3. *the i'th coordinate of σ can be computed in time* $\mathrm{poly}(n)$,
4. *n-windows of σ can be decoded from $\Omega(d)$ errors in* $\mathrm{poly}(n)$ *time.*

The proof of this theorem is omitted from this version of the paper.

In two dimensions, we consider matrices $\Sigma^{N \times N}$, and consider $n \times n$ windows within it. All the definitions are similar, and we omit them from this introduction. Our main result gives efficient constructions of constant rate, constant relative distance matrices over large alphabets.

THEOREM 5. (2-D, LARGE Σ, CONSTANT δ) *There exists an infinite sequence of n and alphabets Σ_n (with $|\Sigma_n| \leq O(n)$, such that for every $R \in (0,1)$, there is a matrix $\sigma \in \Sigma^{N_n \times N_n}$ with:*

1. *the rate of σ is at least R,*
2. *the window-n relative distance of σ is at least $h(R) - o(1)$, where $h(R) > 0$,*
3. *the (i,j)'th coordinate of σ can be computed in time* $\mathrm{poly}(n)$,
4. *$(n \times n)$-windows of σ can be decoded from a constant fraction of errors in* $\mathrm{poly}(n)$ *time.*

The proof of this theorem is also omitted from this version of the paper. These can then be used to construct efficienctly encodable/decodable binary 2-dimensional robust positioning patterns of high rate and constant relative distance (we omit the formal theorem statement). We believe that our techniques generalize to higher dimensional positioning patterns too.

Our large alphabet constructions all use properties of polynomial-based error-correcting codes (especially using their cyclicity when the evaluation set is special), in conjunction with Gray codes.

Our binary constructions are based on a new "augmented" code concatenation scheme. This new scheme is based on two ideas: (1) using a low-autocorrelation sequence as a "marker", and (2) designing an inner code for the concatenation all of whose codewords are far away from all substrings of the marker.

1.2 Related work

The classical notions of de Bruijn sequences and M sequences are the basic examples of positioning patterns. The two-dimensional "de Bruijn torus" is the natural generalization to two dimensions, and were first constructed by [MS76]. These found applications in various practical settings for localization / positioning [SZH+12, Sch01, sty].

Efficiently decodable de Bruijn sequences and tori, which are extremely natural for the positioning applications, were given by [MEP96, MP94, BM93, DMRW93].

The requirement for robustness in positioning patterns is very natural for real-world applications where the positioning pattern is "measured" by a physical device. Indeed, several applied works encountered these problems (in applications such as wireless device localization, and markers for "augmented reality") [KY07, JMDB14, HHSZ13], and proposed ad hoc solutions.

On the theoretical side, there were some important papers on robust positioning such as [KW92, BEG+12, HMNO08]. [KW92] showed that a random linear feedback shift register sequence provides a nearly optimal tradeoff between the window-n distance and the length of the sequence (in the regime where the number of errors is less than $\sqrt{n}$). [BEG+12] gave constructions of

$N \times N$ 2-dimensional robust positioning patterns (for $n \times n$ windows) with $N = 2^{O(n)}$ (while there exist such patterns with $N = 2^{O(n^2)}$).

1.3 Overview of our constructions

We begin by describing a simple construction of a constant rate 1 dimensional sequence over a large alphabet with constant relative window-n distance. This simple construction only leads to codes with rate $R \leq 1/3$, and having rate R approaching 1 seems to require some significantly new ideas.

Let us also remark that there are several easy constructions of *low rate* sequences (with rate $< 1/2$) with constant relative window-n distance using "markers", but going to high rate introduces significant conceptual obstacles (in particular, one really needs to handle the overlap of the windows in the sequence).

Let C be the Reed-Solomon code of degree $\leq k$ polynomials over $\mathbb{F}_q$. We will take $n = q - 1$. Let us choose the sequence evaluation points for these polynomials to be $g, g^2, \ldots, g^{q-1}$, where g is a generator of $\mathbb{F}_q^*$. Thus the codeword corresponding to the polynomial $f(X)$ is a rotation of the codeword corresponding to the polynomial $f(g^i X)$ (for every i).

Partition C into the equivalence classes, where two codewords are equivalent if they are rotations of one another. Let $c_1, \ldots, c_M$ be a collection of codewords, one from each equivalence class. Let $\Sigma = \mathbb{F}_q$, let $N = M \cdot (q-1)$ and let $\sigma \in \Sigma^N$ be the sequence obtained by concatenating $c_1, c_2, \ldots, c_M$. Note that $N \approx q^{k+1}$.

We claim that σ has window-n distance at least $n - 3k$. Indeed, if we look at any length n window of σ, it looks like the concatenation of a suffix of c_i and a prefix of c_{i+1}. A moment's inspection, using the fact that every rotation of c_i is also a codeword of C shows that every n-window of this sequence looks like the splicing together of two codewords of C. Using this fact, it follows that the number of agreements between two distinct n windows is at most 3 times the maximum number of agreements between two codewords. Thus the distance between any two n-windows is at least $n - 3k$.

The above construction fails to do anything interesting if $k > n/3$. To go beyond, we will exploit our ability to carefully choose the ordering of $c_1, \ldots, c_M$. Our construction ensures that many of the windows that straddle c_i and c_{i+1} are (essentially) rotations of c_i (and in particular, they are essentially codewords of C). We do this using a Gray code. The analysis of the distance is somewhat mysterious, and takes advantage of the fact that windows now look like the gluing together of *overlapping* codewords. This leads to a bound of $(n - k)/3$ for the window-n distance.

Our generalization to 2 dimensions uses 2 variable polynomial codes (with bounded individual degree). We design the appropriate Gray code like property, and lay out the codewords of the 2 variable polynomial code on a large 2 dimensional grid. The construction is quite simple given the 1-dimensional construction, but the analysis gets significantly complex.

Our binary construction in the one dimensional case is based on a new "augmented" code concatenation scheme. This new scheme is based on two ideas: (1) using a low-autocorrelation sequence as a "marker", and (2) designing an inner code for the concatenation all of whose codewords are far away from all substrings of the marker.

2 Preliminaries and Notation for 1 Dimensional Robust Positioning Sequences

Some basic preliminaries. Throughout this paper we will use $[n]$ to refer to the first n natural numbers with 0 *included*. That is

$$[n] := \{0, 1, 2, \ldots, n-1\}$$

We need some notation for expressing and accessing values of sequences.

DEFINITION 2.1. *Given a sequence $S := (s_1, \ldots, s_N)$ we define $S[i]$ to be the i^{th} entry of S, i.e. $S[i] = s_i$. Further if $I := (i_0, i_2, \ldots, i_n)$ then we define*

$$S[I] := (S[i_0],\ S[i_1], \ldots,\ S[i_n])$$

Furthermore for our robust positioning patterns we will denote them as a sequence of length N, however we will frequently wish to consider the coordinates cyclically. To that extent for a sequence S of length N we will say that for any integer m even if m is negative or $> N$ we have that $S(m) := S(m \mod N)$.

Also, we will frequently need to refer to intervals of integers, and so we use the notation $[m_1, m_2]$.

DEFINITION 2.2. *Given $m_1 < m_2$ define $[m_1, m_2] := (m_1, m_1 + 1 \ldots, m_2)$. We will use square brackets for inclusive and open brackets for open boundaries much like intervals in $\mathbb{R}$. For example $(m_1, m_2] := (m_1 + 1, \ldots, m_2)$. Sometimes when more compact notation is needed, we will use $\langle m \rangle_n$ to denote $[m, m + n)$.*

For sequences $S_1, S_2 \in \Sigma^n$, we denote their Hamming distance by $\Delta(S_1, S_2)$, and denote their *agreement* by $\text{agree}(S_1, S_2)$. Thus $\Delta(S_1, S_2) + \text{agree}(S_1, S_2) = n$.

For sequences $S_1, S_2, \ldots S_n$, we will denote their concatenation by $(S_1, S_2, \ldots, S_n)$.

We will frequently want to rotate sequences, cyclically permuting their entries. We give special notation

to this operation. We define $\rho : \Sigma^n \mapsto \Sigma^n$ be the coordinate rotation map $\rho((x_1, \ldots, x_n)) = (x_2, x_3, \ldots, x_n, x_1)$.

The following definitions caption the relationship of two sequences being almost the same (i.e. differ in only one position) after a rotation.

DEFINITION 2.3. *Given two sequences $S_1, S_2 \in \Sigma^n$ we write $S_1 \sim S_2$ if there exists some rotation ρ^j such that $\Delta(\rho^j S_1, S_2) \leq 1$. Similarly if $T_1 \in \Sigma^m$ and $T_2 \in \Sigma^n$ where $m \leq n$ we write $T_1 \lesssim T_2$ if there is some i such that $T_1 \sim T_2[i, i+m-1]$.*

Finally we will need a way to quantify the error correcting properties of the sequences we create. We borrow the terms rate, distance and relative distance from Coding Theory as follows:

DEFINITION 2.4. *Given a q-ary sequence S of length N and an integer n (the window length), we say that the rate of S is*

$$R(S) := R := \frac{\log_q(N)}{n}.$$

We define the distance of S to be $\min_{0 \leq i \neq j \leq N} \Delta(S[i, i+n), S[j, j+n))$. Finally we define the relative distance of S to be

$$\delta_S := \min_{0 \leq i \neq j \leq N} \frac{\Delta(S[i, i+n), S[j, j+n))}{n}$$

3 Robust Positioning Sequences Over Large Alphabets

3.1 Overview

In this section we'll show an explicit construction of a robust positioning pattern over large alphabets which is "good" in the sense that it will acheive constant fraction distance and constant rate. Further the construction is capable of acheiving any rate between 0 and 1, and any relative distance between 0 and 1 as well.

The positioning pattern itself is acheived by listing consecutively the entries of a Reed-Solomon code, which has been suitably pruned so that no two codewords are rotations of one another. Further we need to list the remaining codewords in a specific order, namely in such a way that their prefixes form a q-ary gray code of length $\deg(p)$ (See Figure 1). This ordering will ensure that windows which are slightly misaligned and which see the end of some codeword C_i and the gray code prefix of its successor codeword C_{i+1} are tricked into believing they instead see a rotation of the first codeword $\rho^j(C_i)$ (see Remark 3.1 and the accompanying diagram Figure 2). But since we ensured that our codewords were not rotations of eachother, any such rotated codewords will be unique and distant and will allow us to recover the codeword C_i, while the rotation will tell us exactly the location of the current window.

One more small remark is that ocasionally our window will not only see the gray code bits of the subsequent Reed-Solomon codeword, but in such cases we may break the window down into two subwindows which are small windows of rotated codewords, and at a cost of losing some distance from the original Reed-Solomon code, we will be able to decode one of these shortened subwindows.

3.2 Definitions and Construction

First in order to explicitly write down a Reed-Solomon codeword, we need to fix an ordering of our underlying base field $\mathbb{F}$. So to that aim fix g a generator of $\mathbb{F}_q^\times$, and we will order the elments of $\mathbb{F}^\times$ as subsequent powers of g.

DEFINITION 3.1. *Fix $n := |\mathbb{F}^\times| = q - 1$. Given the function $f : [\mathbb{F}_q] \to [\mathbb{F}_q]$ we define the word $C^f := C(f) \in \mathbb{F}_q^n$ by setting*

$$C(f) := (f(g^0),\ f(g^1),\ \ldots,\ f(g^{n-1}))$$

Let $\mathcal{C} := \{C(f) \text{ s.t. } f \in \mathcal{F}\}$.

Then let Σ be a q-ary gray code of length k. Our robust positioning pattern will be built out of blocks consisting of encodings of a certain family of polynomials $\mathcal{F}$ given by interpolating a polynomial with of degree k with prefix given by some $\sigma \in \Sigma$, and with constant term 0, and coefficient of X fixed to be 1 (these last two properties will ensure that the family of polynomials define words which are not rotations of one another).

DEFINITION 3.2. *Given $\sigma \in \mathbb{F}^k$ let $f^\sigma(X) \in \mathbb{F}[X]$ be the unique interpolating polynomial of degree $k+1$ so that:*

- $\text{coeff}_X(f^\sigma) = 1$
- $\text{coeff}_1(f^\sigma) = 0$
- *for each $i \in [0, k)$, $f^\sigma(g^i) = \sigma_i$*

Further, define $\mathcal{F} := \{f^\sigma \text{ s.t. } \sigma \in \mathbb{F}^k\}$.

The first two conditions above are equivalent to saying $f^\sigma(X) := Xh^\sigma(X)$ where

- $h^\sigma(0) = 1$
- for each $i \in [0, k-1]$, $h^\sigma(g^i) = \sigma_i g^{-i}$

Given a polynomial we will encode it in the following manner.

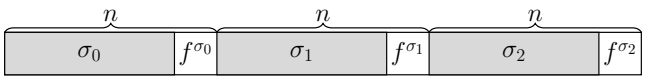

Figure 1: A view of the beginning of the robust positioning pattern S_Σ constructed in Definition 3.4. σ_i represents the i^{th} word in the Gray code Σ, and f^{σ_i} is the rest of the appropriate interpolated polynomial word so that the codeword $C(f^{\sigma_i})$ has σ_i as a prefix.

DEFINITION 3.3. *Given the function $f : [\mathbb{F}_q] \to [\mathbb{F}_q]$ we define the word $C^f := C(f) \in \mathbb{F}_q^n$ by setting*

$$C(f) := (f(g^0),\ f(g^1),\ \ldots,\ f(g^{q-2}))$$

Let $\mathcal{C} := \{C(f) \text{ s.t. } f \in \mathcal{F}\}$.

We now are ready to present the definition of the robust positioning pattern we will study:

DEFINITION 3.4. *Let $\Sigma = \sigma^0, \sigma^1, \ldots, \sigma^{q^k-1}$ be a q-ary gray code of window length k. For convenience of notation we will often write f^a for f^{σ_a} and C^a for $C^{f^{\sigma_a}}$. Then define the sequence S to be*

$$\begin{aligned} S := S_\Sigma &:= \left(C^{f^{\sigma_1}}, C^{f^{\sigma_2}}, \ldots, C^{f^{\sigma_{q^k}}}\right) \\ &:= \left(C^0, C^1, \ldots, C^{q^k-1}\right) \end{aligned}$$

See Figure 1 for a depiction of part of this construction.

3.3 Proof of Distance of S_Σ

The goal of this section is to prove the following distance result for S.

THEOREM 6. *The sequence $S := S_\Sigma = [C^{f^1}, C^{f^2}, \ldots, C^{f^{q^k}}]$ defined in Definition 3.4 is a q-ary sequence of rate $\frac{k+1}{q}$ and distance $\max\left(\frac{q-k}{3} - 3, q - 3k - 9\right)$ with window size $n := q - 1$.*

When considering a window $w = S[m] := S[m, m+n)$ often the most important identifying feature is $\bar{m} = m \mod n$ where $0 \le \bar{m} < n$. This tells us which symbols correspond to Gray code entries, and which are values of the interpolated polynomial f^σ. Larger values of $\bar{m}$ indicate that the Gray code has been pushed leftward (wrapping around) in our window.

Our first observation is that when $\bar{m}$ is small, then we see almost exactly a rotation of a copy of some codeword C^a.

OBSERVATION 3.1. *Let w be a length n window of S_Σ, $w = S_\Sigma\langle m\rangle_n := (S(m),\ S(m+1), \ldots,\ S(m+n-1))$ where $m = an + \bar{m}$ and $0 \le \bar{m} < k$. Then $\rho^{\bar{m}} C^a \sim w$, (i.e. $\Delta(w, \rho^{\bar{m}} C^a) \le 1$).*

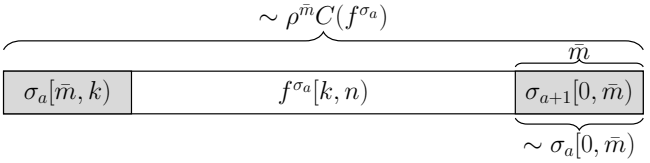

Figure 2: Accompanying diagram for Observation 3.1. Note how the size of $\bar{m}$ affects the position of the Gray code bits, and the fact that $\bar{m} < k$ is important to ensuring that the Gray code bits are a suffix.

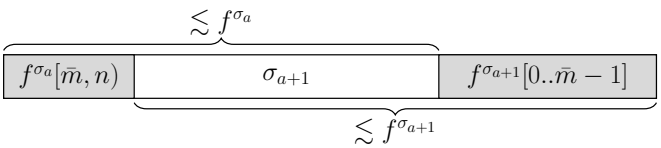

Figure 3: Accompanying figure for Observation 3.2

Proof. For a depiction of the argument see Figure 2. If $m = an + \bar{m}$ then we see that

$$\begin{aligned} w = S_\Sigma[aq + \bar{m}] &= \left(C^a[\bar{m}, n),\ C^{a+1}[0, \bar{m})\right) \\ &= \left(\sigma_a[\bar{m}, k),\ C^a[k, n),\ \sigma_{a+1}[0, \bar{m})\right) \end{aligned}$$

Therefore

$$w = \rho^{\bar{m}}\left(\sigma_{a+1}[0, \bar{m}), \sigma_a[\bar{m}, k),\ C^a[k, n)\right)$$

and from the definition

$$\rho^{\bar{m}} C^a = \rho^{\bar{m}}\left(\sigma_a[0, \bar{m}), \sigma_a[\bar{m}, k),\ C^a[k, n)\right)$$

As Σ is a Gray code we have that $\Delta(\sigma_a, \sigma_{a+1}) = 1$ so comparing the above two expressions it follows immediately that $\Delta(w, \rho^{\bar{m}} C^a) \le \Delta(\sigma_a, \sigma_{a+1}) = 1$ □

Second, we observe that when $\bar{m}$ is larger, the situation isn't as nice, but we can split the window up into two overlapping parts which do look like subwindows of codewords.

OBSERVATION 3.2. *Let $w = S\langle m\rangle_n$ where $m = an + \bar{m}$ and $k < \bar{m} \le n$. If we let $x_1 = n - \bar{m}$ then $w[0, x_1 + k) \lesssim C(f^a)$ and $w[x_1, n) \lesssim C(f^{a+1})$.*

Proof. $w[0, x_1 + k - 1] \subset S\langle m - \bar{m} + k\rangle_n \sim C(f^a)$ by Observation 3.1. We also have that $w\langle x_1, q-1\rangle_n \subset S\langle m + (q - \bar{m})\rangle_n = C(f^{a+1})$. □

We combine these two observations into a single corollary.

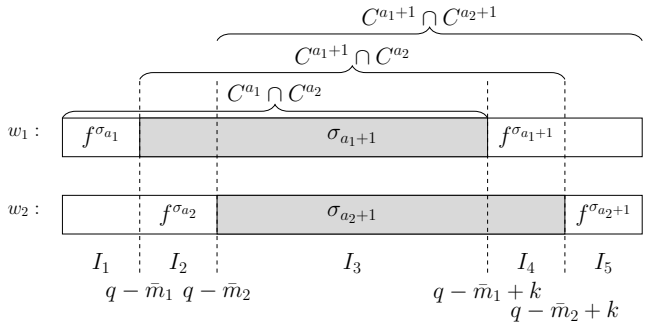

Figure 4: The partition in Case 1. The decomposition of w_1, w_2 into the intersections of rotations of codewords from $\mathcal{C}$ (shown by the curly brackets) is used to provide our distance bounds.

COROLLARY 3.1. *Let $w_1 \neq w_2$ be windows of S_Σ of length $\ell \leq n$. Assume that $w_1 = S\langle m_1 \rangle_n$ and $w_2 = S\langle m_2 \rangle_n$ where for all $i = 1, 2$ we have either $(k \leq \bar{m}_i$ and $\bar{m}_i + \ell < n + k)$ or $(0 \leq \bar{m}_i < k)$. Then agree$(w_1, w_2) \leq \min(\ell, k + 3)$.*

Proof. Note that the congruence conditions are exactly the conditions we need to apply the above observations. If $k \leq \bar{m}_i$ and $\ell < n - \bar{m}_i + k$ then by Observation 3.2 $w_i \lesssim C^{a_i}$ for some a_i. In the second case if $0 \leq \bar{m}_i \leq k$ then $w_i \subset S[m_i, m_i + n) \lesssim \rho^{\bar{m}_i} C^{a_i}$ for some a_i by Observation 3.1. So by using triangle inequality, the fact that $\rho^{\bar{m}_1} C^{a_1} \neq \rho^{\bar{m}_2} C^{a_2}$, and Lemma 3.1 we obtain that

$$\text{agree}(w_1, w_2) \leq \min(\ell, \text{agree}(C^1, C^2)+3) \leq \min(\ell, k+3)$$

□

Now we are ready to begin the proof of our main theorem. The basic strategy will be as follows: Corollary 3.1 will allow us to break each window into two pieces, each of which is a rotation of a subwindow of a codeword from $\mathcal{C}$. Then we will break our windows up into pieces based on these subwindows (a process which will require several cases), analyze what distance and agreement bounds we can get on each piece, and then recombine our answers for the final estimate.

THEOREM 6 *Let $w_1 \neq w_2$ be windows of size q of S_Σ. Then $\Delta(w_1, w_2) \geq \max(q - 3k - 9,\ \frac{q-k}{3} - 3)$.*

Proof. Assume $w_1 = S\langle m_1 \rangle_n$ and $w_2 = S\langle m_2 \rangle_n$. Let $m_1 \equiv \bar{m}_1,\ m_2 \equiv \bar{m}_2$ where $0 \leq \bar{m}_1, \bar{m}_2 < n$. Assume without loss of generality that $\bar{m}_2 \leq \bar{m}_1$. We proceed by cases.

Case 1 First assume that $\bar{m}_1 - \bar{m}_2 < k$ and $k < \bar{m}_2$. As a result we will have that

$$0 < n - \bar{m}_1 \leq n - \bar{m}_2 < n - (\bar{m}_1 - k) \leq n - \bar{m}_2 + k < n$$

Therefore we can partition the interval window $[0, n)$ into 5 pieces by letting (see Figure 4)

$$\begin{aligned} I_1 &:= [0, n - \bar{m}_1) \\ I_2 &:= [n - \bar{m}_1, n - \bar{m}_2) \\ I_3 &:= [n - \bar{m}_2, n - \bar{m}_1 + k) \\ I_4 &:= [n - (\bar{m}_1 - k), \bar{m}_2 + k) \\ I_5 &:= [n - (\bar{m}_2 - k), n) \end{aligned}$$

Note that it is possible that some of these intervals are empty (i.e. if $\bar{m}_1 = \bar{m}_2$) but this will not affect our argument.

For each j let $\text{agree}_j := \text{agree}(w_1[I_j], w_2[I_j])$. By Observation 3.2 for some a_1, a_2 we have that $w_1[I_1, I_2, I_3] \lesssim C^{a_1}$ and $w_1[I_3, I_4, I_5] \lesssim C^{a_1+1}$. Similarly we also have that $w_2[I_1, I_2, I_3, I_4] \lesssim C^{a_2}$ and $w_2[I_3, I_4, I_5] \sim C^{a_2+1}$. Therefore by Corollary 3.1

$$\begin{aligned} &\text{agree}_1 + \text{agree}_2 + \text{agree}_3 \\ &\qquad = \text{agree}(w_1[I_1, I_2, I_3], w_2[I_1, I_2, I_3]) \leq k + 3 \\ &\text{agree}_2 + \text{agree}_3 + \text{agree}_4 \\ &\qquad = \text{agree}(w_1[I_2, I_3, I_4], w_2[I_2, I_3, I_4]) \leq k + 3 \\ &\text{agree}_3 + \text{agree}_4 + \text{agree}_5 \\ &\qquad = \text{agree}(w_1[I_3, I_4, I_5], w_2[I_3, I_4, I_5]) \leq k + 3 \end{aligned}$$

Simply by noting that $|I_2| + |I_3| = |I_3| + |I_4| = k$ we find that

$$\begin{aligned} &\text{agree}_1 + \text{agree}_2 + \text{agree}_5 \leq n - |I_2| - |I_3| = n - k \\ &\text{agree}_1 + \text{agree}_4 + \text{agree}_5 \leq n - |I_3| - |I_4| = n - k \end{aligned}$$

Summing these five inequalities yields $\text{agree}(w_1, w_2) = \sum \text{agree}_m \leq \frac{2n+k}{3} + 3$. Summing only the first and third inequalities we find that

$$\begin{aligned} \text{agree}(w_1, w_2) &= \sum \text{agree}_j \\ &\leq \text{agree}_1 + 2\text{agree}_2 + 2\text{agree}_3 + 2\text{agree}_4 + \text{agree}_5 \\ &\leq 2k + 6 \end{aligned}$$

Therefore in this case we find that $\text{agree}(w_1, w_2) \leq \min(2k + 6, \frac{2n+k}{3} + 3)$.

Case 2 Here assume again that $\bar{m}_2 > k$ but now $\bar{m}_1 - \bar{m}_2 \geq k$. Here we have to partition slightly differently, as the Gray code bits will not overlap. Note that we have

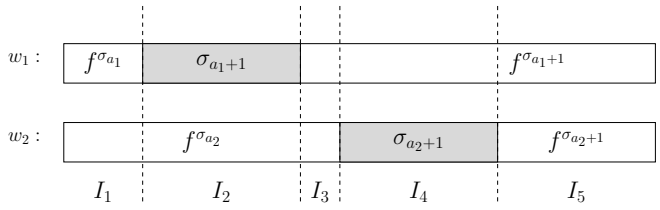

Figure 5: The partition in Case 2

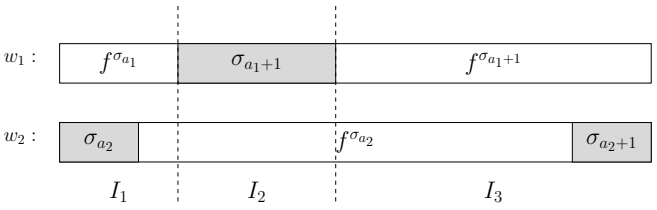

Figure 6: The partition in Case 4

$$0 < n - \bar{m}_1 \leq n - \bar{m}_1 + k \leq n - \bar{m}_2 \leq n - \bar{m}_2 + k \leq n$$

So therefore we can partition $[0, n)$ as follows:

$$\begin{aligned} I_1 &:= [0, n - \bar{m}_1) \\ I_2 &:= [n - \bar{m}_1, n - \bar{m}_1 + k) \\ I_3 &:= [n - \bar{m}_1 + k, n - \bar{m}_2) \\ I_4 &:= [n - \bar{m}_2, n - \bar{m}_2 + k) \\ I_5 &:= [n - \bar{m}_2 + k, n) \end{aligned}$$

Here we will have by Observations 3.1 and 3.2 that for some a_1, a_2 that $w_1[I_1, I_2] \lesssim C^{a_1}$ and $w_1[I_2, I_3, I_4, I_5] \lesssim C^{a_1+1}$, while $w_2[I_1, I_2, I_3, I_4] \lesssim C^{a_2}$ and $w_2[I_4, I_5] \lesssim C^{a_2+1}$. So again defining $\text{agree}_j := \text{agree}(w_1[I_j], w_2[I_j])$ we can use Corollary 3.1 to compute that

$$\text{agree}_1 + \text{agree}_2 = \text{agree}(w_1[I_1, I_2], w_2[I_1, I_2]) \leq k + 3$$

And again by similar reasoning applied on each pair of overlapping subwords

$$\begin{aligned} \text{agree}_1 + \text{agree}_2 &\leq k + 3 \\ \text{agree}_2 + \text{agree}_3 + \text{agree}_4 &\leq k + 3 \\ \text{agree}_4 + \text{agree}_5 &\leq k + 3 \end{aligned}$$

Also, simply by noting that $|I_2| = |I_4| = k$ we find that

$$\begin{aligned} \text{agree}_1 + \text{agree}_2 + \text{agree}_3 + \text{agree}_5 &\leq q - |I_2| = n - k \\ \text{agree}_1 + \text{agree}_3 + \text{agree}_4 + \text{agree}_5 &\leq q - |I_4| = n - k \end{aligned}$$

So summing all five inequalities we find that $3\sum \text{agree}_m \leq 2q + k + 9$. And so $\text{agree}(w_1, w_2) \leq \frac{2q+k}{3} + 3$. Also summing over only the first 3 inequalities we find that

$$\begin{aligned} \text{agree}(w_1, w_2) &= \sum \text{agree}_j \\ &\leq \text{agree}_1 + 2\text{agree}_2 + \text{agree}_3 + 2\text{agree}_4 + \text{agree}_5 \\ &\leq 3k + 9 \end{aligned}$$

Therefore in this case we find that $\text{agree}(w_1, w_2) \leq \min(3k + 9, \frac{2q+k}{3} + 3)$.

Case 3 In this case we assume that $0 \leq \bar{m}_2 \leq \bar{m}_1 \leq k$. By Lemma 3.1 we find that for some a_i $w_i \sim C^{a_i}$. So in this case we have by Corollary 3.1 that $\text{agree}(w_1, w_2) \leq k + 3$.

Case 4 The last case is when $\bar{m}_2 \leq k$ but $\bar{m}_1 > k$. In this case if we let

$$\begin{aligned} I_1 &:= [0, n - \bar{m}_1 - 1] \\ I_2 &= [n - \bar{m}_1, q - \bar{m}_1 + d - 1] \\ I_3 &:= [n - \bar{m}_1 + d, q - 1] \end{aligned}$$

Then we have again by Observation 3.1 that $w_2 \sim C^{a_2}$ for some a_2. By Observation 3.2 that for some a_1, $w_1[I_1, I_2] \lesssim C^{a_1}$ and $w_1[I_2, I_3] \lesssim C^{a_1+1}$. If we define $\text{agree}_j := \text{agree}(w_1[I_j],\ w_2(w_2[I_j])$ then we will have that

$$\text{agree}_1 + \text{agree}_2 = \text{agree}(w_1[I_1, I_2], w_2[I_1, I_2]) \leq k + 3$$

And due to similar reasoning to the above we will have that

$$\begin{aligned} \text{agree}_1 + \text{agree}_2 &\leq k + 3 \\ \text{agree}_2 + \text{agree}_3 &\leq k + 3 \end{aligned}$$

Also, simply by noting that $|I_2| = k$ we find that

$$\text{agree}_1 + \text{agree}_3 \leq n - k$$

So aggregating these inequalities we find that both $2\sum \text{agree}_m \leq n + k + 6$ and $\sum \text{agree}_m \leq 2k + 6$. As a result in this final case we get

$$\begin{aligned} \text{agree}(w_1, w_2) = \sum \text{agree}_j &\leq \min\left(\frac{n+k}{2} + 3, 2k + 6\right) \\ &\leq \min\left(\frac{2n+k}{3} + 3, 3k + 9\right) \end{aligned}$$

□

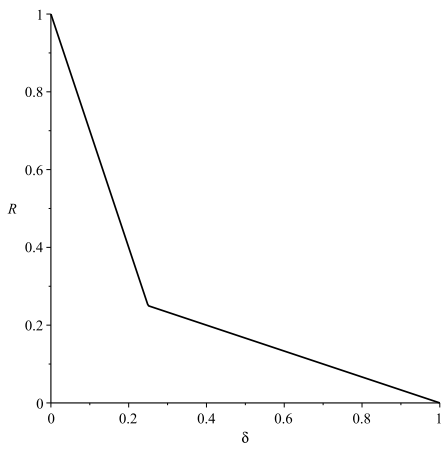

Figure 7: The Rate vs. Distance tradeoff of our construction as $q \to \infty$

COROLLARY 3.2. *For any $0 < R < 1$ and $\delta < \max(\frac{1-R}{3}, 1-3R)$, for large enough q there exists a q-ary sequence of window length q, rate R and relative distance δ.*

Proof. We can compute the rate of S_Σ is

$$R = \frac{\log_n(n \cdot q^k)}{n} \geq \frac{\log_q(nq^k)}{n} = \frac{k+1}{n} - o_n(1)$$

And by Theorem **??** we have that the relative distance is

$$\begin{aligned}\delta &= \frac{\max\left(n-3k,\ \frac{n-k}{3}\right)}{n} - o_n(1) \\ &\geq \max\left(\frac{1-R}{3}, 1-3R\right) - o_n(1)\end{aligned}$$

□

Here we prove the useful fact that not only are any two codewords in $\mathcal{C}$ far apart, but also any two rotations of codewords of $\mathcal{C}$ are also distant as well. This is crucial for our analysis which routinely uses the fact that misaligned windows of the robust positioning pattern S_Σ still look like rotated codewords from $\mathcal{C}$.

LEMMA 3.1. *For any $i_1, i_2 \in [n]$ and any $C^{a_1}, C^{a_2} \in \mathcal{C}$, so long as $(a_1, i_1) \not\equiv (a_2, i_2)$, then $\Delta(\rho^{i_1}(C^{a_1}), \rho^{i_2}(C^{a_2})) \geq q-k-1$ and therefore also $agree(\rho^{i_1}(C^{a_1}), \rho^{i_2}(C^{a_2})) \leq k+1$.*

Proof. First, we note that

$$\rho^{i_\ell}(C^{a_\ell}) = (f^{a_\ell}(g^{i_\ell}), f^{a_\ell}(g^{i_\ell+1}), \ldots, f^{a_\ell}(g^{i_\ell-1}))$$

But this is exactly the encoding $C(p_\ell)$ of the degree $k+1$ polynomial $p_\ell(X) = f^{i_m}(g^{i_\ell}X)$. Furthermore, we have that $p_1(0) = p_2(0) = 0$.

Therefore we will have that $\Delta(\rho^{j_2}(C^{i_2}), \rho^{j_2}(C^{i_2})) \geq q - deg(p_2 - p_1) \geq q-k-1$ if we can show that $p_2 - p_1 \neq 0$.

But note that

$$\begin{aligned}\text{coeff}_X(p_1) &= \text{coeff}_X(f^{a_1}(g^{i_1}X)) = g^{i_1} \\ \text{coeff}_X(p_2) &= \text{coeff}_X(f^{a_2}(g^{i_2}X)) = g^{i_2}\end{aligned}$$

Therefore $p_1 = p_2$ only if $g^{i_1} = g^{i_2}$, which occurs only if $i_1 \equiv i_2 \mod n$.

If that is the case then $p_1 = p_2$ directly implies that $f^{a_1} = f^{a_2}$, contradicting our assumption that $(a_1, i_1) \not\equiv (a_2, i_2)$. □

4 Binary Positioning Sequences

4.1 Preliminaries

To concatenate down to binary we first need a marker to let us know the boundaries between words. To this end we construct a suitable binary word Ψ so that any two rotations of Ψ agree and differ in almost exactly half the coordinates.

LEMMA 4.1. *For any $t \in \mathbb{N}$ there exists a binary word Ψ of length $2^t - 1$ so that for any two i, j, $0 \leq i \neq j \leq 2^t - 1$, the rotations $\rho^i(\Psi), \rho^j(\Psi)$ have the property $\left|agree(\rho^i(\Psi), \rho^j(\Psi)) - \frac{\ell}{2}\right| \leq 2^{\frac{t}{2}-1} \cdot O(t)$.*

We sketch the construction. Let g be a generator of $\mathbb{F}_{2^t}^\times$. Order the elements of $\mathbb{F}_{2^t}^\times$ by $x_i := g^i$, and take $\psi : \mathbb{F} \to \{\pm 1\}$ to be a nontrivial additive character of $\mathbb{F}_{2^t}$. Now we can define $\tilde{\Psi} := [\psi(g^0), \psi(g^1), \ldots, \psi(g^{2^t-2})]$. We will then let our codeword be the binary verison of this string by replacing -1 with 0.

Next, in order to use this marker appopriately in our concatenation, we must have a binary outer code which is also far from any window of Ψ.

LEMMA 4.2. *Given $\delta < \frac{1}{2}$ and $R < 1 - H_2(\delta)$ For sufficiently large n, and any word W of length n, there exists a binary code $\mathcal{C}$ of block length n, relative distance δ and rate R so that all codewords of $\mathcal{C}$ have distance at least $\frac{\delta n}{2}$ from any rotation of W.*

Proof. By the Gilbert-Varshamov bound, for sufficiently large n there exists a code $\mathcal{C}_0$ with block length n relative distance δ and rate $R > 1 - H_2(\delta)$. Now we can define $\mathcal{C}$ from $\mathcal{C}_0$ by simply removing any codeword which has distance less than $\frac{\delta n}{2}$ from any rotation of W. Since $\mathcal{C}$ has distance greater than δn there can be at most one codeword removed per rotation of W. Therefore $|\mathcal{C}| \geq |\mathcal{C}| - n$ and so the rate is asymptotically unchanged. As the distance of $\mathcal{C}$ as at least the distance of $\mathcal{C}_0$, $\mathcal{C}$ satisfies the conditions we need. □

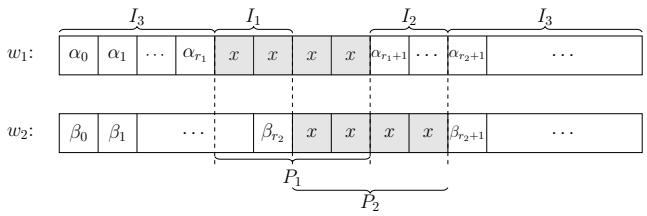

Figure 8: An illustration of the argument in Lemma 4.3. $w_i[I_3]$ is a subset of a window of S of length $n - |P_1 \cup P_2|$, and so contributes at least $d - |P_1 \cup P_2|$ in distance. Meanwhile there is additional contribution of $|P_1| + |P_2| - 2|P_1 \cap P_2|$ to the distance from I_1 and I_2 where copies of x are compared to elements of Σ.

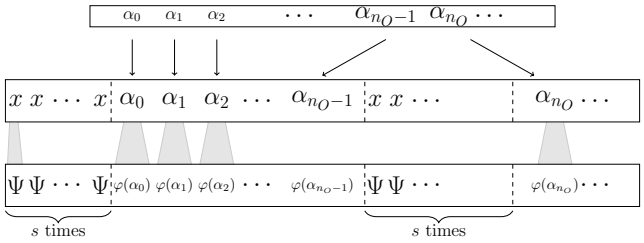

Figure 9: A view of the binary Robust Self Location Pattern T constructed in Section 4.3. Note that the sequence consists of a concatenation of the large alphabet sequence S to binary, intermixed with a locator word Ψ, which will aid in making detecting where a window lies in T modulo n_I.

4.2 Augmented Sequences

First to build our binary sequence we will need to define a method of augmenting large alphabet sequences to include marker symbols which will help us with alignment issues.

Definition 4.1. *Let S be a positioning sequence over Σ with window length n. Fix any $x \notin \Sigma$ and define the s-augmented sequence $A := A_s(S)$ over the alphabet $\Sigma \cup x$ of window length $n + s$ by*

$$A[a(n+s)+b] := \begin{cases} x & \text{if } 0 \le b < s \\ S[an+b-s] & \text{if } s \le b < n+s \end{cases}$$

Lemma 4.3. *Let S be a positioning sequence and A the s-augmented sequence. If S has distance d then A has distance at least* $\min(d, 2s)$.

Proof. Take $w_1 \neq w_2$ to be arbitrary distinct length $n + s$ windows of U where $w_i := U\langle m_i \rangle_{n+s}$. Now for each window we define P_i to be the places corresponding to the copies of x. That is

$$P_i := \{j \in [n+s] \text{ s.t. } m_i + j \mod (n+s) < s\}$$

Also define

$$I_1 := P_1 \cap P_2^c \quad I_2 := P_1^c \cap P_2 \quad I_3 := (P_1 \cup P_2)^c$$

Because $w_1[I_1]$ is a string of only the character x and $w_2[I_1]$ contains no copies of x at all, we have that $\Delta(w_1[I_1], w_2[I_1]) = |I_1|$. Similarly we find that $\Delta(w_1[I_2], w_2[I_2]) = |I_2|$

There are two cases to consider. First if P_1 and P_2 are disjoint then $\Delta(w_1, w_2) \ge 2s$ as $|I_1| = |I_2| = s$.

In the second case we have $P_1 \cap P_2$ is nonempty. Therefore as the union of two contiguous (on the circle) intervals of length s, $P_1 \cup P_2$ is also such an interval but of length $\ell := |P_1 \cup P_2|$. There are two more subcases to consider. First if $P_1 \cup P_2 = [a, a+\ell)$ for some $a \le n+s-\ell$ or second if $P_1 \cup P_2 = [a, n+s) \cup [0, \ell - n - s + a)$ for some $a \ge n + s - \ell$. Define $\tilde{w}_i := w_i[P_i^c]$, and note that by the construction of A, $\tilde{w}_i$ is a length n window of S and $\tilde{w}_1 \neq \tilde{w}_2$ so $\Delta(\tilde{w}_1, \tilde{w}_2) \ge d$. Here we will have that $I_3 := (P_1 \cup P_2)^c = [0, a) \cup [a + \ell, n + s)$ and so for $i = 1, 2$ it can be seen that $w_i[I_3] = \tilde{w}_i[I]$ where $I = [0, a) \cup [a + \ell - s, n)$. Therefore as $|I| = n + \ell - s$ it follows that

$$\begin{aligned} \Delta(w_1, w_2) &\ge \Delta(w_1[I_1], w_2[I_1]) + \Delta(w_1[I_2], w_2[I_2]) \\ &\quad + \Delta(w_1[I_3], w_2[I_3]) \\ &\ge |I_1| + |I_2| + \Delta(\tilde{w}_1, \tilde{w}_2) - (\ell - s) \\ &= 3s - 2|P_1 \cap P_2| + d - |P_1 \cup P_2| \\ &\ge d \end{aligned}$$

In the second subcase the argument is exactly the same, but here we will have $I_3 := (P_1 \cup P_2)^c = [\ell - n - s + a, a)$ a contiguous interval of length $n + s - \ell$, and so $w_1[I_3]$ and $w_2[I_3]$ are subwindows of S of length $n+s-\ell$ and so the proof follows through using the same estimates as above.

□

4.3 Construction of the Binary Robust Positioning Sequence

Now we are ready to proceed with the construction of our binary robust positioning pattern.

Fix some $t \in \mathbb{N}$ and let $n := 2^t - 1$. Then take S to be a q-ary robust positioning sequence with window length n_O, rate R_O and relative distance δ_O. Let Ψ be a word of the form promised by Lemma 4.1 with length $n = 2^t - 1$ and let $\mathcal{C}$ be a binary code with q messages

n_I	$:= \lvert\varphi(\alpha)\rvert + \lvert\Psi\rvert = \tilde{n}_I + u$	R_I	$:= \frac{\log_2 q}{n}$
δ_I	$:= H_2^{-1}(1-r) - \eta$	$\lvert\Psi\rvert$	$:= 2^t - 1 := n$
N	$:= n(n_O + s)$	$\bar{m}$	$:= m \mod \star$

Figure 10: Summary of Notation

and block length n (and therefore rate $R_I := \frac{\log q}{n}$), relative distance $\delta_I > H_2^{-1}(1-R_I)$ chosen as promised by Lemma 4.2 to have all windows distant from Ψ.

Take A to be an s-augmentation of the q-ary sequence S, then let $\varphi : \mathbb{F} \cup \{x\} \to \mathcal{C}$ to be an encoding of $\mathbb{F}_q$ to codewords of $\mathcal{C}$ with $\varphi(x) := \Psi$. Then we can define our binary positioning pattern to be given by

$$T := [\varphi(A[1]),\ \varphi(A[2]),\ \varphi(A[3]), \ldots]$$

In particular we can see that if $m = aN + bn + c$ with $0 \le b < N$ and $0 \le s < n$ then

$$T[m] = T[aN + bn + c] := \begin{cases} \Psi[c] & \text{if } b < s \\ \phi(T[an_O + b])[c] & \text{if } s \le b \end{cases}$$

4.4 Proof of Distance

We state our main result about the distance of T.

THEOREM 7. *T is a binary robust positoning pattern of block length $(n_O + s)n_I$ and distance at least*

$$\min\left(\frac{(\min(n_O\delta_O - 1, 2s)\delta_I n_I}{2},\ (s-2)\frac{\delta_I n}{2}\right)$$

and rate at least $R_O R_I \frac{n_O}{n_O+s}$

LEMMA 4.4. *Let $w_1 \neq w_2$ be any two windows of T of the form $w_i := T\langle m_i\rangle_N$. If $m_1 \neq m_2 \mod n$ then $\Delta(w_1, w_2) \ge (s-2)\frac{\delta n}{2}$.*

Proof. Define

$$P_1 := \{i \in [N] \text{ s.t. } m_1 + i = aN + bn + c \text{ where } 0 \le c < n \text{ and } 0 \le b < s\}$$

That is to say that P_1 is the set of indices corresponding to entries in w_1 coming from copies of Ψ. In particular if $i \in P_1$ and $i \equiv \bar{i} \mod n$ then $w_1[i] = \Psi[\bar{i}]$.

Now let

$$A_2 := \{i \in [N] \text{ s.t. } m_2 + i \equiv 0 \mod n\}$$

Assume that $m_2 \equiv \bar{m}_2 \mod n$. If we consider the sequence of subwindows $\langle in + n - m_2\rangle_n$.

So A_2 is the set of beginnings of length n windows corresponding to either a copy of Ψ or a codeword of $\mathcal{C}$. Because P_1 is a set of size sn consisting of at most 2 runs of consecutive integers, it must contain at least $s-2$ length n windows of the form $\langle a\rangle_n$ where $a \in A_2$. But for such a window we will have $w_2\langle a\rangle_n$ is either a codeword of $\mathcal{C}$ or a copy of Ψ, while if $x := m_2 - m_1 \not\equiv 0 \mod n$ we will have that $w_1\langle a\rangle_n = \rho^x\Psi$. So either by the construction of Ψ to have low autocorrelation (Lemma 4.1) or by the distance of all codewords of $\mathcal{C}$ from all rotations of Ψ (Lemma 4.2) we will have that

$$\Delta(w_1[\langle a\rangle_n], w_2[\langle a\rangle_n]) \ge \min\left(\frac{\delta_I n}{2}, (1-o(1))\frac{n}{2}\right) = \frac{\delta_I n}{2}$$

Because P_1 contains $s-2$ disjoint such windows, the result follows. □

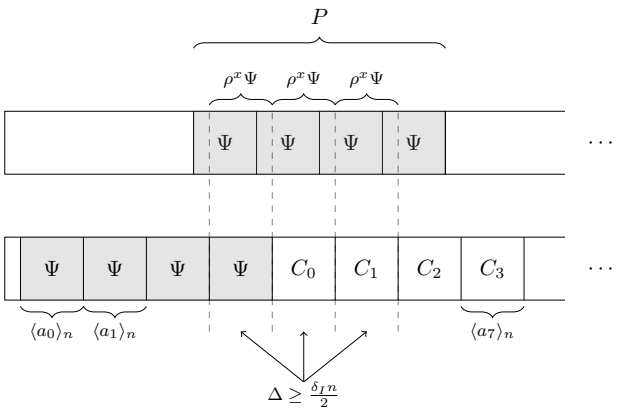

Figure 11: An illustration of the argument in Lemma 4.4. The distance in the nonaligned case comes from comparing rotations of the marker word Ψ to codewords of $\mathcal{C}$ and other copies of Ψ.

Next we cover the case when the codewords are aligned modulo n. Here we will have that binary codewords and copies of Ψ will be compared to eachother and will get our distance from the distance of the concatenation code combined with the distance of the large alphabet positioning sequence.

LEMMA 4.5. *Let $w_1 \neq w_2$ be any two windows of T of the form $w_i := T\langle m_i\rangle_N$. If $m_1 \equiv m_2 \mod n$ then $\Delta(w_1, w_2) \ge \frac{(\min(n_O\delta_O - 1, 2s)\delta_I n_I}{2}$.*

Proof. Let $0 \le \bar{m} < n$ such that $m_1 \equiv m_2 \equiv \bar{m} \mod n$. Now for $j \in [n_O + s - 1]$ define $I_j := \langle nj + n - \bar{m}\rangle_n$. Then for any i, j $w_i[I_j]$ corresponds to a codeword in $\mathcal{C}$ or a copy of Ψ, and in fact it must be that $u_1 := \{\varphi^{-1}(w_1[I_j])\}_{j\in[n_O+s-1]}$ and $u_2 := \{\varphi^{-1}(w_2[I_j])\}_{j\in[n_O+s-1]}$ are distinct windows of A of length $n_O + s - 1$. By Lemma 4.3 we know that these

windows differ in at least $\min(n_O\delta_O, 2s) - 1$ positions. Since $\mathcal{C} \cup \{\Psi\}$ forms a code of distance $\frac{n\delta_I}{2}$ our result follows. □

We are now ready to restate and prove our main result of this section:

THEOREM 8. *T is a binary robust positoning pattern of block length $(n_O + s)n_I$ and distance at least*

$$\min\left(\frac{(\min(n_O\delta_O - 1, 2s)\delta_I n_I}{2},\ (s-2)\frac{\delta_I n}{2}\right)$$

and rate at least $R_O R_I \frac{n_O}{n_O+s}$

Proof. The statement of distance is a combination of Lemmas 4.5 and 4.4. Meanwhile the rate statement follows from the standard calculation

$$R = \frac{\log_2 |T|}{(n_O+s)n_I} \geq \frac{\log_2 |q| \log_2 |S|}{(n_O+s)n_I} = R_O R_I \frac{n_O}{n_O+s}$$

□

We note that choosing s, R_O and R_I properly we can obtain the following corollary.

COROLLARY 4.1. *For any $0 < R < 1$ there is some $\delta(R)$ such that the above construction yields binary positioning patterns of arbitrarily long block length, rate R and relative distance $\delta(R)$.*

5 Efficient encoding and decoding

6 Encoding/Decoding Over Large Alphabets

The following is an algorithm for encoding a window of $S := S_\Sigma$ as defined in Definition 3.4. Assume we wish to compute $S[m]$ where $0 \leq m \leq q^k - 1$. Assume we are using the Gray Code Σ given by the standard inductive construction, for which an efficient method of encoding and decoding exists.

ALGORITHM 6.1. Given m to compute $S\langle m\rangle_n$ do:

1. Find $0 \leq \bar{m} < q$ and a so that $m = aq + \bar{m}$
2. Find σ_a and σ_{a+1} (the a^{th} and $(a+1)^{st}$ entries in the q-ary gray code Σ.
3. Interpolate the polynomials f^a and f^{a+1} so that for $j = 0, 1$
   (a) $f^{a+j}(0) = 0$
   (b) $(f^{a+j})'(0) = 1$
   (c) $f^{a+j}(g^i) = \sigma_i^{a+j}$ for $0 \leq i \leq k-1$
4. Output $[(f^a(g^j))_{j=\bar{m}}^{q-1},\ (f^a(g^j))_{j=0}^{\bar{m}-1}]$

THEOREM 9. *The following algorithm for decoding received words of the window sequence S can correct $\min(\delta n, \frac{q+k}{2} - 1 - \sqrt{q(k+1)})$ errors (where $\delta n = \max(q - 3k - 9,\ \frac{q-k}{3} - 3)$). Furthermore the algorithm runs in time $O(poly(q))$.*

ALGORITHM 6.2. Assume we receive a window w of length q. To decode do:

1. Run the Guruswami-Sudan list decoding algorithm [GS99] for Reed Solomon codes on w, reuturning the list of degree $k+1$ polynomials $L := \{p \text{ s.t. } \Delta(C(p), w) \leq q - \sqrt{q(k+1)}\}$.
2. For each polynomial $p \in L$ do the following:
   (a) For each $i \in [q]$ find the index a_i such that

$$\begin{aligned}\rho^i(C^p)[0, k-1] &= (p(g^i), p(g^{i+1}), \ldots, p(g^{i+k-1})) \\ &= \Sigma[a_i, a_i + k - 1)\end{aligned}$$

   where Σ is the q-ary Gray code in use.
   (b) Make the guesses $\mu_p^1 = a_i q + (q - i)$ and $\mu_p^0 = (a_i - 1)q + (q - i)$
   (c) For each guess μ_p^i let $w_p^i = S\langle\mu_p^i\rangle_n$. If $\Delta(w, w_p^i) < \frac{\delta q}{2} = \frac{1}{2}\max(q - 3k - 6,\ \frac{q-k}{3} - 2)$ then return w_p^i and its index μ_p^i

Proof. Assume that the sent window was $u = S[m]$ with $m = aq + \bar{m}$, so that $\Delta(u, w) \leq \frac{q+k}{2} - 1 - \sqrt{q(k+1)}$. Let $x = q - \bar{m}$, and $I_0 = [0, x + k - 1]$, $I_1 = [x, q-1]$. If $\bar{m} < \frac{q-k}{2}$ then by either Observation 3.1 or 3.2 we will have that $u[I_0] \sim \rho^{\bar{m}} C^a[I_0]$, and therefore

$$\begin{aligned}\Delta(u, \rho^{\bar{m}} C^a) &\leq (q - |I_0|) + \Delta(u[I_0], \rho^{\bar{m}} C^a[I_0]) \\ &\leq (q - |I_0|) + 1\end{aligned}$$

Similarly if $\bar{m} \geq \frac{q-k}{2}$ then $u[I_1] \sim C^{a+1}[0, k + \bar{m} - 1] = \rho^{\bar{m}} C^{a+1}[x, q-1]$. So

$$\begin{aligned}\Delta(\rho^x u, C^{a+1}) &= \Delta(u, \rho^{\bar{m}} C^{a+1}) \\ &\leq (q - |I_1|) + \Delta(u[I_1], \rho^{\bar{m}} C^{a+1}[I_1]) \\ &\leq (q - |I_1|) + 1\end{aligned}$$

Because $|I_0| + |I_1| = q + k$ for some j we must have $|I_j| \geq \frac{q+k}{2}$.

Therefore we can compute that

$$\begin{aligned}\Delta(w, \rho^{\bar{m}} C^{a+j}) &\leq \Delta(w,u) + \Delta(u, \rho^{\bar{m}} C^{a+j}) \\ &\leq \frac{q+k}{2} - 1 - \sqrt{q(k+1)} + q - |I_j| + 1 \\ &\leq q - \sqrt{q(k+1)}\end{aligned}$$

Therefore we see that the Guruswami-Sudan list decoding algorithm will place either $f^{a+j}(x+\bar{m})$ in its list L.

Therefore when step 2a tries $p = f^{a+j}(x+\bar{m})$ and $i = q - \bar{m}$ we will have $\rho^i \rho^{\bar{m}} C^{a+j}[0, k-1] = C^{a+j}[0, k-1] = \sigma^{a+j}$, and conseuqently the guesses $\mu_p^1 = (a+j)q + \bar{m}$ and $\mu_p^0 = (a+j-1)q + \bar{m}$ will be made. If $j = 0$ then the former will be correct, and if $j = 1$ then the latter will be. Either way step c will check $\mu = aq + \bar{m} = m$, and because $\Delta(w, S\langle m\rangle_n) < \frac{\delta n}{2}$ the algorithm will return m and $S\langle m\rangle_n$.

The only thing left to check is that the algorithm returns no false positives. But, by Theorem **??** we know that all windows of S have distance at least $\delta n := \max(q - 3k - 9, \frac{q-k}{3} - 3)$ from eachother. Therefore it follows that there is always at most one window $S\langle\mu\rangle_n$ so that $\Delta(S\langle\mu\rangle_n, w) \leq \frac{\delta n}{2}$, for any window w, and therefore only the correct window could ever be returned.

To check the runtime claim we note that the Algorithm in step 1 runs in time $poly(q)$ and returns a list of size $|L| \leq q^2$. Furthermore each of the operation in steps 2a takes time $poly(q)$ by the decodability of Σ. The operations in step 2b take time $O(1)$, and each step in 2c runs in time $poly(q)$ by the encoding algorithm of S given □

6.1 Encoding/Decoding in Binary

Let T be a binary robust positioning sequence as constructed section 4.3. with window length $N = (n_O + s)n$ and distance d.

First we comment on construction of the sequence. The only point of interest here is to find the locator word Ψ its accompanying code $\mathcal{C}$. Ψ is just a character over $\mathbb{F}_{2^n}$ and so can be any multilinear polynomial. over $\mathbb{F}_2[X_1, \ldots, X_n]$. For $\mathcal{C}$ we may pick any efficiently encodable and decodable good distance binary code of rate R, of which numerous constructions exist. Then to pair it with Ψ we only have to remove the codeword in $\mathcal{C}$ of distance less than $\frac{d}{2}$ from each rotation of Ψ, a process which takes at most n calls to the decoding algorithm of $\mathcal{C}$.

Now we discuss the decoding algorithm. Here the process proceeds in two steps. First we find where in the window are the s copies of Ψ. Once we know that, we know exactly which blocks of length n correspond to concatenated codewords, and can apply usual decoding methods for concatenated words.

Assume that S has a decoder algorithm D_O which given a window $w \in \Sigma_O^{n_O}$ will determine (if possible) the unique window $S\langle m\rangle_{n_O}$ so that $\Delta(S\langle m\rangle_{n_O}, w) < d$ in time $poly(q)$. Assume also that we also have for the inner alphabet a decoding algorithm D_I so that for any received word $w \in [2]^n$ D_I returns the unique letter $\alpha \in [q]$ so that $\Delta(C_I(\alpha), w) < \frac{\delta_I n}{2}$ in time $poly(q)$.

ALGORITHM 6.3. For each i from 0 to N do

1. For each $0 \leq j < n_O$ decode (if possible) the length n window $(\rho^i w)\langle jn\rangle_n$ to α_j using the decoder of $\mathcal{C}$.
2. Let $\tilde{w}_i$ be the q-ary string $(\alpha_0, \ldots, \alpha_{n_O-1})$.
3. Run the decoder D_O over large alphabets on $\tilde{w}_i$ (padded with an extra bit if necessary) and return its index $\tilde{\mu}_i$
4. Let $\mu = n_O\tilde{\mu} - i$.
5. If $\Delta(T[\mu], w) < \frac{d}{2}$ then return $T[\mu], \mu$.

THEOREM 10. *Algorithm 6.3 runs in time* poly(N) *and given a window $w \subset [2]^N$ returns (if it exists) the unique window $u := T\langle m\rangle_N$ such that $\Delta(u, w) < \delta_O\delta_I\frac{(n-1)n_O}{4}$ in time $poly(N)$.*

The argument that this decoding works is very similar to decoding an ordinary concatenated code. We can brute force through every rotation $\rho^i w$ for $i \in [N]$ of the received window w, and try decoding $\rho^i w$ as we would any concatenated code of length $n_O n$ (the algorithm will return when the unchecked sn entries correspond to the s copies of Ψ). Since one of the rotations will correspond to us having a word which is $n-1$ blocks of concatenated codewords (and possibly 1 junk block from the beginning and end of the window), we will be able to decode at least a $\delta_O N_O$ fraction of these blocks correctly, and the decoder for the large alphabet robust positioning sequence handles decoding the resulting large alphabet sequence. In step 5 we use the fact that T has good distance to eliminate any possible false positives.

7 Positioning sequences with constant distance

In this section, we give a brief description of our construction of positioning sequences with constant

distance d. The details appear in the full version of this paper.

Over large alphabets Σ, it follows by inspecting the parameters in Theorem 6 that the construction there leads to sequences of length $\frac{|\Sigma|^n}{|\Sigma|^{O(d)}}$, which is essentially optimal (upto the constant in the $O(d)$) by the Singleton bound.

Over the binary alphabet, we have to do something different. Here we are aiming to get a sequence of length $\frac{2^n}{n^{O}(d)}$. The concatenation scheme described for the case of constant relative distance codes is insufficient, since any nontrivial concatenation map leads to a drastic reduction in the length of the sequence. Instead, we will use a trivial concatenation map, along with a simpler marker, at the cost of having to rely on an unproven conjecture (Conjecture C from the introduction).

Assuming Conjecture C, for infinitely many r we can choose a prime q between $2^r - cr$ and $2^r - 1$. We start with a large alphabet positioning sequence over the alphabet $\Sigma = \mathbb{F}_q$ with distance d. Now choose a one-to-one map $\phi : \mathbb{F}_q \to \{0,1\}^r$ whose image avoids the string 0^r: this is possible since $q \leq 2^r - 1$. We will be using the map ϕ to encode large alphabet symbols into sequences of binary symbols. The final binary sequence is then obtained by taking the ϕ-encoding of each symbol of the large alphabet sequence, along with the marker sequence $(0^{2r}1^r)^{3d}$. The goal of this marker sequence, as in the case of the constant relative distance codes, is to ensure alignment. The fact that the image of ϕ avoids 0^r is what ensures that this marker sequence cannot have too much agreement with any symbols outside the marker sequence. Finally, the fact that $q > 2^r - cr$ ensures that the length of the sequence so constructed is as long as $\frac{2^n}{n^{O}(d)}$: encoding elements of Σ by ϕ did not make us lose too much in the rate.

8 Two dimensional positioning patterns

In this section, we give a brief description of our constructions of 2 dimensional positioning patterns of high rate and constant relative distance.

We begin with the case of large alphabets. Following the 1 dimensional case, our strategy is to consider 2-variable polynomials over a field $\mathbb{F}$ with bounded individual degree, and to evaluate them on $\mathbb{F}^* \times \mathbb{F}^*$. As before, we order the elements of $\mathbb{F}^*$ as a geometric progression, and consider two codewords equivalent if they are (two-dimensional) rotations of one another. We then choose one codeword from each equivalence class, and lay them out in a 2-dimensional grid in a certain carefully chosen order (the carefully chosen order is again based on Gray codes). This gives us our 2-dimensional positioning pattern for $n \times n$ windows (with $n = |\mathbb{F}^*|$).

The analysis of the distance of this construction is now significantly more involved. We need to take two $n \times n$ windows in this 2-dimensional pattern, and measure their distance. This could involve up to eight 2-variable polynomials (four per window) and depending on the locations of these windows a number of different cases arise. Ultimately the analysis relies on the distance properties of 2-variable polynomials, that their restrictions to rows/columns are 1-variable polynomials, and the Gray-code based ordering of the polynomials that leads to the four polynomials in any window being very closely related to each other. We omit the details in this version of the paper.

Over small alphabets, our constructions are based on concatenation using markers again. The ideas are very similar to the 1-dimensional case. The notable difference is that we need to construct a 2-dimensional marker that is far away from every rotation of itself, which we do again based on character sum bounds.

Unfortunately, our constructions do not seem delicate enough to handle the constant distance regime with optimal redundancy (as opposed to constant relative distance). This seems like an interesting question for future work.

References

[BEG+12] A.M. Bruckstein, T. Etzion, R. Giryes, N. Gordon, R.J. Holt, and D. Shuldiner. Simple and robust binary self-location patterns. *Information Theory, IEEE Transactions on*, 58(7):4884–4889, July 2012.

[BM93] John Burns and Chris J Mitchell. Coding schemes for two-dimensional position sensing. In *Institute of Mathematics and Its Applications Conference Series*, volume 45, pages 31–31. Oxford University Press, 1993.

[DMRW93] ZD Dai, KM Martin, MJB Robshaw, and PR Wild. Orientable sequences. In *INSTITUTE OF MATHEMATICS AND ITS APPLICATIONS CONFERENCE SERIES*, volume 45, pages 97–97. OXFORD UNIVERSITY PRESS, 1993.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[HHSZ13] Zsolt Horváth, Adam Herout, István Szentandrási, and Michal Zachariáš. Design and detection of local geometric features for deformable marker fields. In *Proceedings of*

the 29th Spring Conference on Computer Graphics, SCCG '13, pages 073:73–073:80, New York, NY, USA, 2013. ACM.

[HMNO08] Mariko Hagita, Makoto Matsumoto, Fumio Natsu, and Yuki Ohtsuka. Error correcting sequence and projective de bruijn graph. *Graphs and Combinatorics*, 24(3):185–194, 2008.

[JMDB14] Lode Jorissen, Steven Maesen, Ashish Doshi, and Philippe Bekaert. Robust global tracking using a seamless structured pattern of dots. In Lucio Tommaso De Paolis and Antonio Mongelli, editors, *Augmented and Virtual Reality*, volume 8853 of *Lecture Notes in Computer Science*, pages 210–231. Springer International Publishing, 2014.

[KW92] P.Y. Kumar and V.K. Wei. Minimum distance of logarithmic and fractional partial m-sequences. *Information Theory, IEEE Transactions on*, 38(5):1474–1482, Sep 1992.

[KY07] Bhaskar Krishnamachari and Kiran Yedavalli. Secure sequence-based localization for wireless networks. In Radha Poovendran, Sumit Roy, and Cliff Wang, editors, *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, volume 30 of *Advances in Information Security*, pages 237–247. Springer US, 2007.

[MEP96] C.J. Mitchell, T. Etzion, and K.G. Paterson. A method for constructing decodable de bruijn sequences. *Information Theory, IEEE Transactions on*, 42(5):1472–1478, Sep 1996.

[MP94] ChrisJ. Mitchell and KennethG. Paterson. Decoding perfect maps. *Designs, Codes and Cryptography*, 4(1):11–30, 1994.

[MS76] F.J. MacWilliams and N.J.A. Sloane. Pseudo-random sequences and arrays. *Proceedings of the IEEE*, 64(12):1715–1729, Dec 1976.

[Sch01] E.R. Scheinerman. Determining planar location via complement-free de brujin sequences using discrete optical sensors. *Robotics and Automation, IEEE Transactions on*, 17(6):883–889, Dec 2001.

[sty] Microsoft mulls a stylus for any screen. http://www.technologyreview.com/news/428521/microsc mulls-a-stylus-for-any-screen/ .

[SZH+12] I. Szentandrasi, M. Zacharias, J. Havel, A. Herout, M. Dubska, and R. Kajan. Uniform marker fields: Camera localization by orientable de bruijn tori. In *Mixed and Augmented Reality (ISMAR), 2012 IEEE International Symposium on*, pages 319–320, Nov 2012.