Stereoscopic families of permutations, and their applications

(Extended Abstract)

Uriel Feige[*] Robert Krauthgamer[†]

Abstract

A stereoscopic family of permutations maps an m-dimensional mesh into several 1-dimensional lines, in a way that jointly preserves distance information. Specifically, consider any two points and denote their distance on the m-dimensional mesh by d. Then the distance between their images, on the line on which these images are closest together, is $O(d^m)$.

We initiate a systematic study of stereoscopic families of permutations. We show a construction of these families that involves the use of $m+1$ images. We also show that under some additional restrictions (namely, adjacent points on the image lines originate at points which are not too far away on the mesh), three images are necessary in order to construct such a family for the 2-dimensional mesh.

We present two applications for stereoscopic families of permutations. One application is an algorithm for routing on the mesh that guarantees delivery of each packet within a number of steps that depends upon the distance between this packet's source and destination, but is independent of the size of the mesh. Our algorithm is exceptionally simple, involves no queues, and can be used in dynamic settings in which packets are continuously generated. Another application is an extension of the construction of nonexpansive hash functions of Linial and Sasson (STOC 96) from the case of one dimensional metrics to arbitrary dimensions.

1 Introduction

We study the following problem. Let U be an $n \times n$ mesh, and let V be an integer line with $N = n^2$ points, $V = [1 \dots N]$. We look for a family $\mathcal{F} = \{\pi_1, \dots, \pi_{|\mathcal{F}|}\}$ of one-to-one mappings from U to V, such that for any two points $x, y \in U$, there is some $\pi_i \in \mathcal{F}$ with $|\pi_i(x) - \pi_i(y)| = O((d_U(x, y))^2)$, where $d_U(x, y)$ denotes the Manhattan distance (L_1 norm) on U. That is, on at least one of the lines, the distance between the images of the two points is in the quadratic order of the original distance on the mesh. It is straightforward to see that the cardinality of such a family must be at least two (see Lemma 1), and if the cardinality of $\mathcal{F}$ is a constant independent of n, then the desired quadratic upper bound on distance expansion is the best one can achieve (Observation 2). Therefore, our objective is to minimize $|\mathcal{F}|$. We call this model a *stereoscopic family of permutations*, as each member of the family is a permutation of the N elements, and similarly to stereoscopic vision, a higher dimensional object is mapped into several lower dimensional objects in a way that preserves depth (distance) information.

Our study was motivated by the question of packet routing on the $n \times n$ mesh, where each edge of the mesh represents two antiparallel communication links, and each link can carry a constant number of packets per unit time. We want a simple routing algorithm that delivers each packet in time that depends on the distance between its source and destination, but not on the size of the network. Our suggestion is to construct few Hamiltonian paths on the mesh, each of a different "color" (which may correspond to wavelength, in case of optical routing). At the time of injection a packet chooses the Hamiltonian path on which its distance to destination is the smallest, and follows the color that corresponds to this path until reaching its destination. Quantitatively, for any packet with distance d_p between its source and destination, we require that the distance along at least one of the Hamiltonian paths is $O(d_p^2)$. This is exactly the question of constructing stereoscopic families of permutations, with the additional restriction that each individual permutation is a Hamiltonian path on the mesh. Our main result is that *three Hamiltonian paths suffice, but two do not*. A more detailed discussion of our results on routing and related work is presented in Section 3.

In order to deal with dimensions higher than two, we generalize our construction, showing that a stereoscopic family of $m+1$ permutations can map an m dimensional mesh into $m+1$ lines, such that distances d are expanded to $O(d^m)$.

[*]Department of Applied Math and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. `feige@wisdom.weizmann.ac.il`. Incumbent of the Joseph and Celia Reskin Career Development Chair. Yigal Alon Fellow.

[†]Department of Applied Math and Computer Science, The Weizmann Institute, Rehovot 76100, Israel. `robi@wisdom.weizmann.ac.il`.

Higher dimension constructions are applicable in the area of locality preserving hashing. Linial and Sasson [12] suggested such a hashing scheme (in fact, a nonexpansive scheme), for one dimensional metrics, and left the construction of higher dimensional hashing schemes as an open question. We propose a solution to the multi-dimensional case, based on stereoscopic families of permutations. Independently, Indyk et al. [9] proposed a different construction. This application is presented in Section 4, where the relevant definitions are given.

Related Work. Our construction of stereoscopic families of permutations uses a variant of Hilbert's space filling curve (see, for example, [16]). Researchers in geometric data structures and geographic information system (GIS) often use such curves and study their properties (see, for example, [17]). Asano et al. [1] study a related problem. Given a two dimensional square grid, they construct a one-to-one mapping to the one dimensional line that has the following property: any two dimensional interval (square) on the grid is mapped into at most three one dimensional intervals on the line. They also show that under certain restrictions on the mapping, three one dimensional intervals are necessary. We shall discuss these results again in Section 4.

In the context of routing, questions regarding covering a graph by a family of trees satisfying certain relations between distances on the graph and distances on the trees have been studied in [2]. In the context of hashing, mapping an m-dimensional mesh into buckets, such that each bucket contains few elements, and local neighborhoods (of one particular size) on the mesh are mapped into a small number of buckets, has been studied in [6]. The two studies mentioned above (and other similar studies) capture some aspects of stereoscopic families of permutations, but do not capture the main characteristics of our model – *a constant size family that simultaneously handles all distances*.

Open questions. There are gaps between our upper bounds and lower bounds for the cardinality of the families of stereoscopic permutations that map from m dimensions to q dimensions with distance expansion $O(d^{m/q})$. Our upper bounds use $m+1$ permutations, whereas our lower bounds show that at least two permutations are necessary (when $m > q$). We have made a first step towards showing a lower bound of three permutations for the case that $m = 2$ and $q = 1$. Closing (or narrowing) the gaps between the lower bounds and the upper bounds is the main open question.

1.1 Definitions

Consider a universe U (either a mesh or torus of arbitrary dimension) with N elements, with the corresponding distance d_U (L_1 norm).

Definition 1 *A* **permutation** π *is a one-to-one function* $\pi_i : U \mapsto V$*, where* U, V *are meshes (or tori) of arbitrary (possibly different) dimensions, each with* N *elements* ($|U| = |V| = N$)*.*

Assume V *is one dimensional (i.e.* $\pi : U \mapsto [1 \dots N]$*). Then* π *is called a* **Hamiltonian path** *or shortly a* **path** *if it is continuous, i.e.*

$$\textit{for all } 1 \leq x < N, \qquad d_U(\pi^{-1}(x), \pi^{-1}(x+1)) = 1$$

Definition 2 *Let* $\pi : U \mapsto [1 \dots N]$ *be a one-to-one function, where* U *is a mesh (or torus). Then* π *is called an* $(\boldsymbol{\alpha}, \boldsymbol{\beta})$**-shrinkable numbering** *of vertices if*

$$\forall x, y \in U, \qquad d_U(x, y) \leq \beta(|\pi(x) - \pi(y)|)^{\alpha}$$

Remark: The term shrinkable numbering was introduced in [7], based on earlier work [18, 11, 16].

Definition 3 *Let* $\mathcal{F}^{(N)} = \{\pi_1, \pi_2, \dots, \pi_l\}$ *be a family of permutations* $\pi_i : U \mapsto V$*, where* U, V *are meshes (or tori) of arbitrary (possibly different) dimensions, each with* N *elements* ($|U| = |V| = N$)*.* $\mathcal{F}^{(N)}$ *is called a* **stereoscopic family of permutations** *with* **expansion** $g_N : \mathbb{N} \mapsto \mathbb{N}$*, if*

$$\forall x, y \in U, \qquad \min_i \{\, d_V(\pi_i(x), \pi_i(y)) \,\} \ \leq\ g_N(d_U(x, y))$$

Notation: We will denote by d_i the L_1 distance on the i-dimensional universe (either mesh or torus). Unless stated explicitly otherwise, we shall always assume that V (the target mesh) is the one dimensional integer line.

We defined a stereoscopic family of permutations to contain any permutation (one-to-one) functions. Routing according to a stereoscopic family $\mathcal{F}$ requires particular permutations which are Hamiltonian paths, since at each time step, packets can only be directed to a neighboring node. The more general definition (arbitrary permutations) allows greater flexibility for other applications.

The definition of a stereoscopic family of permutations is based solely on the metric defined on the mesh (or torus). In principle, stereoscopic families of permutations can be defined on any graph (network), but this is beyond the scope of the current work.

Summing up, the model of stereoscopic family of permutations specifies m and q (the dimensions of the source and destination meshes/tori), N (size of domain, namely the number of elements to map), $|\mathcal{F}|$ (number of permutations used, which we would like to be a constant independent of N), and g_N (the function bounding distance expansion, which we would like to be independent of N, usually $g_N(d) = O(d^{m/q})$).

2 Construction of stereoscopic families of permutations

2.1 Preliminary Observations

One permutation does not suffice for the construction of stereoscopic families of permutations.

Lemma 1 *Let π be a one-to-one function from the $n \times n$ mesh to the line. Then*

$$\exists x, y \quad d_2(x, y) = 1 \ \wedge \ d_1(\pi(x), \pi(y)) > \frac{n}{2}$$

where d_1 denotes distance on the (one dimensional) line, and d_2 denotes distance on the (two dimensional) mesh.

Proof. Take the nodes x and y furthest apart on the line ($d_1(x, y) = n^2 - 1$). Let x', y' be the corresponding nodes in the mesh. Then $d_2(x', y') \leq 2n - 1$. Consider a shortest path from x' to y' on the two dimensional mesh, $x' = x_1, x_2, \ldots, x_r = y'$ for some $r < 2n$. Using the triangle inequality, there are two adjacent points x_i, x_{i+1} on this path with $d_1(\pi(x_i), \pi(x_{i+1})) \geq \frac{n^2-1}{2n-1} > n/2$. □

The following is a general lower bound on the expansion of stereoscopic families of permutations whose cardinality is constant.

Observation 2 *Let $\{\mathcal{F}^{(N)}\}_{N=1}^{\infty}$ be stereoscopic families of permutations on the m-dimensional mesh (or torus), and let $|\mathcal{F}^{(N)}| = O(1)$. Then $g_N(d) = \Omega(d^m)$.*

Proof. Consider the following counting argument. Let x be a fixed node on the m-dimensional mesh (or torus). Then the number of nodes on the mesh (or torus) in radius R from x is $\Theta(R^m)$. Since the number of permutations in $\mathcal{F}$ is constant, at least one of these nodes must be $\Omega(R^m)$ steps far from x on all permutations (lines). □

2.2 Constructive Upper Bound with Three Paths

We construct a stereoscopic family of permutations $\mathcal{F} = \{\pi_1, \pi_2, \pi_3\}$ for the two dimensional case with expansion $g(d) = O(d^2)$, which is the best possible expansion up to constant factors, by Observation 2. For simplicity, we shall illustrate this construction on an $n \times n$ torus, where $n = 3 \cdot 2^l$.

Lemma 3 *Let T_n be an $n \times n$ torus, where $n = 3 \cdot 2^l$, for arbitrary $l > 0$. Then there exists a family $\mathcal{F} = \{\pi_1, \pi_2, \pi_3\}$ of 3 Hamiltonian paths on T_n, such that*

$$\forall x, y \quad \min_{i=1,2,3} \{d_1(\pi_i(x), \pi_i(y))\} \leq 36(d_2(x, y))^2$$

Proof. Denote each of the torus nodes as a pair (i, j) where $1 \leq i, j \leq n$. We construct a Hamiltonian path π in a way similar to Hilbert's space filling curve [16]. Consider the torus T_n as an $n \times n$ mesh M, and we shall describe a Hamiltonian path on M from $(1, 1)$ to $(n, 1)$. This Hamiltonian path on M is defined recursively, by partitioning M to 2 segments in each axis, to get 4 equal size sub-meshes M_1, M_2, M_3, M_4, as in figure 1. The path starts at $(1, 1)$ which is in M_1, and ends up at $(n, 1)$ which is in M_4. Therefore, the path shall be $M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4$, as in figure 1.

Constructing the path inside M_1 is done by breaking M_1 into 4 sub-meshes M_{11}, M_{12}, M_{13} and M_{14}. The path has to travel from M_{11}, recursively cover all M_1 sub-meshes, and finally move to M_2. The only way to do it (see figure 1), is $M_{11} \rightarrow M_{14} \rightarrow M_{13} \rightarrow M_{12}$.

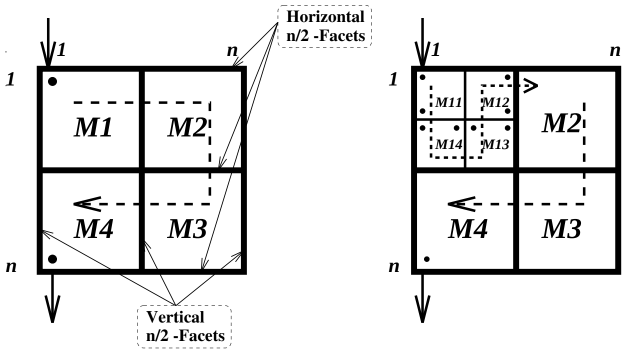

Figure 1. The Hamiltonian path and Refining it inside M_1

Formally, at each phase we break our mesh M into 2 segments at each axis, get 4 equal size sub-meshes M_1, M_2, M_3, M_4, and consider this high-level description as a 2×2 mesh. We define a ***d*-facet** to be a virtual border line between adjacent $d \times d$ sub-meshes (i.e. face/edge of the square, see figure 1).

By definition, transitions between successive sub-meshes are made only through the sub-meshes corner nodes (i.e. one of $(1, 1); (1, n); (n, 1); (n, n)$). In addition, in each sub-mesh, the entrance (corner) node and the exit (corner) node are necessarily adjacent corners. W.l.o.g. we assume the entrance node is $(1, 1)$, and thus the exit node is either $(1, n)$ or $(n, 1)$. A corresponding Hamiltonian path is always feasible by recursion, as shown in figure 2.

The path is constructed recursively, until we get down to a 3×3 mesh, in which a similar construction is also feasible (see figure 3). The path has the following properties:

1. After i recursion phases, the mesh M is partitioned into 2^i equal segments in each axis, total of $(2^i)^2 = 2^{2i}$

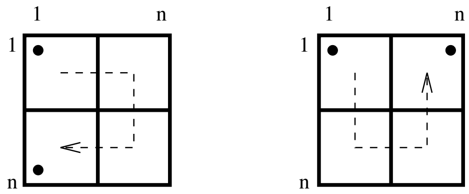

Figure 2. Demonstrating Hamiltonian paths on a 2×2 **Mesh**

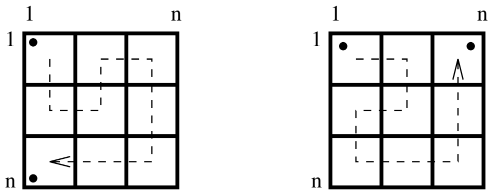

Figure 3. Feasibility of Hamiltonian paths on a 3×3 **mesh**

sub-meshes, each of size $\frac{n}{2^i} \times \frac{n}{2^i}$.

2. Each of the construction sub-meshes is traversed as a whole (i.e. it is covered sequentially without any jumps outside).

3. The distance on π of any two nodes in the same sub-mesh, say M' from the i-th phase, is less then the size of the sub-mesh, i.e. $d_1(\pi(x), \pi(y)) < (\frac{n}{2^i})^2$.

We define π_1 to be exactly π, as described above. Since the torus is cyclic, any path on it can be moved (shifted) using an additive transformation modulo n. So let π_2 be the same path as π shifted $\frac{n}{3}$ units in each axis (say right and down), and π_3 the same with $\frac{2n}{3}$ units shift. This can be alternatively viewed as fixing the mesh (on the torus) with an $\frac{n}{3}$ (or $\frac{2n}{3}$) shift in each axis, and then constructing the path recursively.

Let x, y be arbitrary nodes on the torus, and we will bound their distance on the π_i's. Denote $d' = d_2(x, y)$ their distance on the torus, and let $d = 3 \cdot 2^j$ be the smallest such that $d > 3d'$, so clearly, $d \leq 6d'$.

We now partition the torus into sub-meshes of size $d \times d$ (i.e. by $\log_2 \frac{n}{d} = l - j$ phases of the construction), separately for each path π_i (totally 3 sets of sub-meshes).

Let f_1 be a horizontal facet in the $d \times d$ partitioning of π_1, and f_2 a horizontal facet in the $d \times d$ partitioning of π_2. We claim that the distance between these facets is at least $\frac{d}{3}$ mesh nodes. Facets f_1 appear every d rows, and facets f_2 appear at $\frac{n}{3}$ plus multiples of d. Consider the location of f_2 relatively to f_1, i.e. its location modulo d (since f_1 appears at multiples of d). It comes down to computing $\frac{n}{3}$ modulo d, and $\frac{n}{3} = 2^l = d \cdot \frac{2^{l-j}}{3}$. Since 3 does not divide 2^{l-j}, we conclude that the difference between adjacent facets f_1 and f_2 is $\frac{d}{3}$.

By symmetry considerations, we conclude an important property that any two d-facets of different paths ($\pi_i \neq \pi_{i'}$) but with the same orientation (either horizontal or vertical) are separated by at least $\frac{d}{3}$ torus nodes.

We now claim that x, y belong to the same $d \times d$ sub-mesh in some Hamiltonian path π_i, and hence their distance on this π_i is less than the sub-mesh size, and we're done:

$$d_1(\pi_i(x), \pi_i(y)) < d^2 \leq (6d')^2 = 36(d_2(x, y))^2$$

Assume to the contrary, that x and y are in different $d \times d$ sub-meshes in all 3 Hamiltonian paths. Thus, x, y are separated from each other by at least one facet in any π_i. These 3 facets are either horizontal or vertical, so at least two of them must have the same orientation (by the pigeon-hole principle), w.l.o.g. say horizontal. Therefore, moving from x to y, one must cross these two horizontal facets, which are separated from each other by at least $\frac{d}{3}$ rows, so $d_2(x, y) \geq \frac{d}{3} > d' = d_2(x, y)$, which is contradictory. □

The specific paths used in this construction have yet another property which is complementary to our requirement of bounding the neighborhoods expansion.

Lemma 4 *Let π be the path constructed recursively in the proof of Lemma 3. Then π is a $(\frac{1}{2}, c)$-shrinkable numbering of vertices, for some constant c. That is*

$$\forall x, y \qquad d_2(x, y) \leq c\sqrt{d_1(\pi(x), \pi(y))}$$

Proof. A value of $c < 4$ was claimed without proof in [18]. For completeness, we present a simple proof for the case $c = 6$. Denote $d = d_1(\pi(x), \pi(y))$. Then there is some $d' \times d'$ square in the construction such that $\sqrt{d} \leq d' < 2\sqrt{d}$. Assume w.l.o.g. that x appears before y on the path. Then traveling on the path from x to y (total of $d - 1$ steps), one can either stay in the same $d' \times d'$ square or advance to a neighboring one, but not further (because the square's size is $(d')^2 \geq d$). Hence x and y are in the same $2d' \times d'$ rectangle on the mesh, and thus $d_2(x, y) \leq d' + 2d' \leq 6\sqrt{d}$. □

In other words, $\mathcal{F}$ always expands distances at least to their quadratics. Therefore, our stereoscopic families of permutations satisfy $\min_{i=1,2,3}\{d_1(\pi_i(x), \pi_i(y))\} = \Theta((d_2(x, y))^2)$.

2.3 A Lower Bound of three Hamiltonian paths

In this section we show that it is impossible to preserve quadratic distances with two Hamiltonian paths. Our proof is based on the following argument. Consider a segment of length k of the path π_1. It covers a region on the mesh. By the geometry of the mesh, this region has a boundary of length $\Omega(\sqrt{k})$. The other path π_2 must be responsible for connecting between points on the boundary and points just outside of it. In order to do this, π_2 must follow the boundary closely, because π_2 is continuous. Hence neither π_1 nor π_2 are able to connect points in the boundary to points of intermediate distance ($\sqrt[5]{k}$) from the boundary. Below we turn this intuitive argument to a full proof.

Definition 4 *Let π be a path. A* **segment s of π** *is a restriction of π resulting with the image set $[a \dots b]$, i.e. π restricted to the domain $\pi^{-1}([a \dots b])$.*

Let s be a segment of path π on the $n \times n$ mesh M_n (or torus T_n where $|s| < n$). Then it can be easily seen that the projection of the path on every single axis is an interval, and that at least one of these intervals is of size at least $\sqrt{|s|}$.

Definition 5 *Let U be an $n \times n$ mesh (or torus) universe. Then $\pi : U \mapsto \{1, 2, \dots, n^2\}$ is a* **diagonal neighbors path** *if for all i, $\pi^{-1}(i)$ and $\pi^{-1}(i+1)$ are either neighbors or diagonal neighbors (see figure 4 for illustration).*

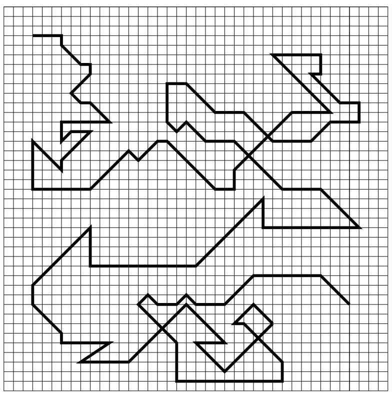

Figure 4. Diagonal neighbors path

Lemma 5 *Let T_n be the $n \times n$ torus (or mesh), and π_1, π_2 be Hamiltonian paths on T_n. Then for any constant $c > 0$ and large enough n,*

$$\exists x, y \in T_n \qquad \min_{i=1,2}\{d_1(\pi_i(x), \pi_i(y))\} > c(d_2(x, y))^2$$

Proof. Assume n is large enough, so for example $\sqrt[4]{n} >> c$. From now on, we deal only with respectively small objects (segments of paths, etc.), up to size $\frac{1}{2}n$. Therefore we can consider our torus to be a mesh, assuming w.l.o.g. that the following construction takes place in the center of the mesh.

Let s_1 be a segment of length $k = \sqrt{n}$ of π_1 (in the center of the mesh), whose endpoints are E, F. W.l.o.g. the projection of s_1 on the x-axis (horizontal) is larger than on the y-axis. Denote this projected interval by I, thus $|I| \geq \sqrt{k}$. Let I_E, I_F be $\frac{1}{10}\sqrt{k}$ intervals around the x-axis projection of E and F, respectively. Removing I_E and I_F from I, we get at most 3 sub-intervals of total size $|I \setminus (I_E \cup I_F)| > \frac{1}{2}|I|$, so at least one sub-interval I' is of size $|I'| > \frac{1}{3}(\frac{1}{2}|I|) \geq \frac{1}{6}\sqrt{k}$. Denote this interval as $I' = [a \dots b]$, and let E', F' be the highest nodes in the columns a and b, respectively, which are on s_1 (see figure 5). Let t_1 be the sub-segment of s_1, whose endpoints are E' and F'. Partition the interval $[a \dots b]$ into quarters and consider the 3 columns separating the quarters, i.e. every $\frac{b-a}{4}$ nodes.

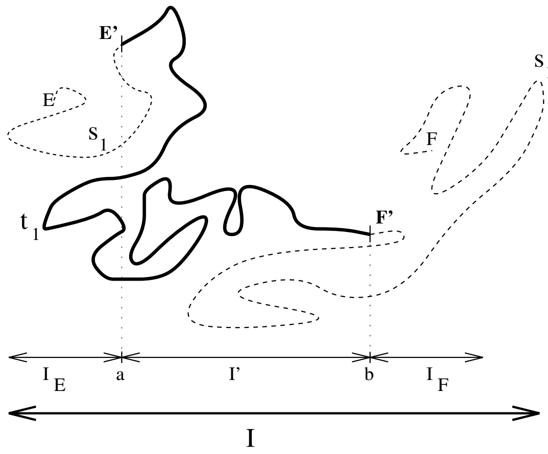

Figure 5. We get $I' = [a \dots b]$ by removing the endpoints intervals I_E, I_F from s_1

Let y_i $(i = 1, 2, 3)$ be a node in this i-th (quarter) column, which is above t_1 in this column and whose distance from t_1 is $l = \sqrt[5]{k}$. Trivially, such y_i must exist and let z_i be its closest node on t_1, so $d_2(z_i, y_i) = l$ (see figure 6).

We define a border line R starting at E' and going $2k$ nodes vertically upwards, then horizontally right until the column of F', and then vertically down to F' (see figure 6). Let $C = C(t_1)$ be all nodes reachable from y_1 without using (crossing) nodes from the border line R nor the path t_1. In fact, defining C with any of the y_i's results with exactly the same set C, since all y_i's are easily connected without crossing t_1.

Let $B = B(t_1)$ be all nodes in t_1 which have a neighbor in C. We claim that B is a diagonal neighbors path connecting E' and F'. Indeed, consider the nodes as squares in a continuous plane, then C is actually a connected set of these squares (add the squares one by one), whose boundary is thus a closed polygonal line. In particular, this line connects

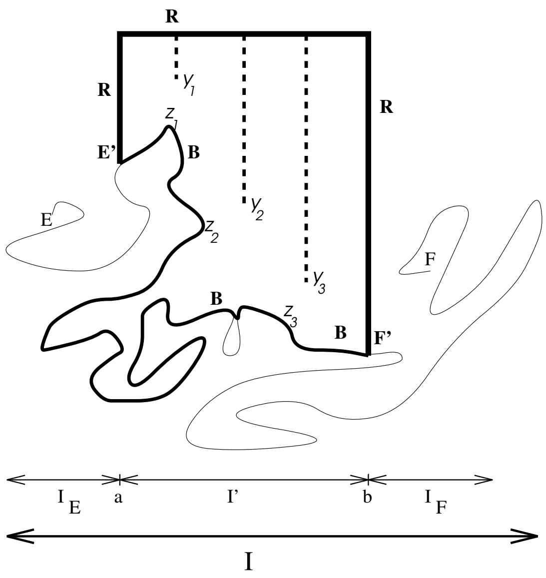

Figure 6. Border line R and boundary $B(t_1)$

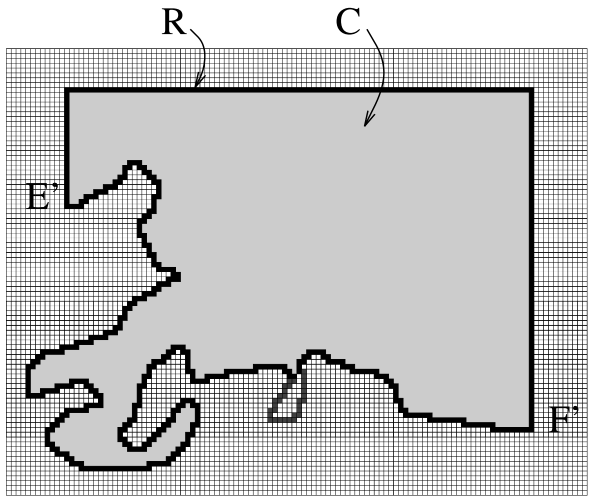

Figure 7. Diagonal neighbors path $B(t_1)$, induced by $C(t_1)$ (connecting E' and F')

E' and F' and corresponds to B above, proving B's nature (see figure 7).

We also claim that $z_i \in B$ as follows. By definition, $z_i \in t_1$. Consider a shortest path from z_i to y_i. No node in this path is in t_1, because z_i was defined as the closest to y_i from t_1. No node in this path is in R because the boundary R is much further than l from y_i (recall that $l = \sqrt[5]{k} << \frac{1}{4}|I'|$). Therefore, every single node in this path is in C, and thus $z_i \in t_1$ has a neighbor in C.

Consider going from E' to F' along B and visiting z_i's on the way, denoted as$E' = x_0, x_1, \ldots, x_q = F'$. B is a diagonal neighbors path, so $d_2(x_j, x_{j+1}) \leq 2$. By definition of B, any x_j has a neighbor $x'_j \in C$, and thus x'_j is not in t_1, and in fact not in s_1 (notice that s_1 and C are disjoint).

Hence $d_2(x'_j, x_{j+1}) \leq 3$, and they must be connected by some π_i within $c \cdot 3^2 = 9c$ steps, or otherwise we're done. But x'_j is not on s_1, so their distance on π_1 is at least $\frac{1}{10}\sqrt{k} >> 9c$. Therefore their distance on π_2 must be at most $9c$ steps. The same argument applies also to x_j and x'_j, which are neighbors, so we conclude that they are connected by π_2 within $c \cdot 1^2 = c$ steps. Hence, $d_1(\pi_2(x_j), \pi_2(x_{j+1})) \leq c + 9c = 10c$.

Consider the occurrences of z_1, z_2, z_3 on π_2. Each pair (say z_1 and z_2) have a series of successive x_j's connecting them (e.g. $z_1 = x_{r_1}$ and $z_2 = x_{r_2}$, then the series is $x_{r_1}, x_{r_1+1}, \ldots, x_{r_2-1}, x_{r_2}$). The distance between two successive x_j's is at most $10c$. Hence $d_1(\pi_2(z_1), \pi_2(z_2)) \leq 10ck << n$, and we conclude that all z_1, z_2, z_3 must be close together on π_2. It is therefore possible to find the *middle* z_i, say z_3, as in figure 8.

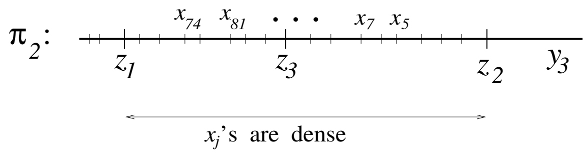

Figure 8. The segment $[z_1 \ldots z_2]$ is "dense" with x_j's

The distance of y_3 from any $x_j \in B$ is at least $l >> 10c$. Since the interval $[z_1 \ldots z_2]$ (on π_2) is "dense" with x_j's at every $10c$ steps, y_3 must be outside the interval. However, both z_1, z_2 are at least $\frac{1}{4}|I'| - l >> cl^2$ far from z_3 on π_2. So z_3 and y_3 are not connected by π_2 within cl^2. Their distance on π_1 is also more than cl^2 (because z_3 is at least $\frac{1}{4}|I'| - l$ far from t_1 endpoints). So we're done:

$$\min_{i=1,2}\{d_1(\pi_i(z_3), \pi_i(y_3))\} > cl^2$$

$\square$

Remark: The proof of Lemma 5 uses the continuity property of π_1, π_2, when these curves are at the center of the mesh. For the purpose of the proof, it does not matter whether at the boundaries of the mesh the curves are not continuous, and jump from one location on the boundary to another. We shall use this fact in Section 2.5.

2.4 Enhancing the Lower Bound

The lower bound of Lemma 5 is given for the case of Hamiltonian paths and quadratic expansion ($g(d) = O(d^2)$). It can be generalized to permutations π in which $d_U(\pi^{-1}(x), \pi^{-1}(x+1)) \leq \lambda$ (instead of Hamiltonian paths we allow jumps bounded by λ), and arbitrary expansion functions $g(d)$, whenever λ and $g(d)$ are independent of the mesh size n. This can be further extended to cases where λ and $g(d)$ depend on n in a weak sense. For example, if λ does not grow too quickly with n (i.e., $\lambda_n = o(n^{1/15})$), then two λ_n-bounded jumps permutations cannot achieve quadratic expansion (for sufficiently large n).

The above implies that either we use large jumps and then some distance shrinks considerably, or some distance expands by more than squaring. Hence for families $\mathcal{F}$ of two permutations, $\min_{i=1,2}\{d_1(\pi_i(x), \pi_i(y))\} = \Theta((d_2(x,y))^2)$ is impossible.

2.5 Extended Model - Path with Repetitions

We present here a variant of the model of stereoscopic families of permutations. Instead of a family of t permutations, we would like a single one-to-many mapping from the mesh U to the line $V = [1 \ldots tN]$, such that each mesh point is mapped to t distinct locations on the line. For two points $x, y \in U$, their distance on V is defined to be the distance between the two images that are closest together. What would be the minimal t needed to bound distance expansion to quadratic order? Does $t = 2$ suffice? We provide a negative answer for the case where continuity is required.

Definition 6 *Let U be a mesh (or torus) of N elements, and t some fixed integer. Then $W : \{1, 2, \ldots, tN\} \mapsto U$ is called a* **walk with** t **repetitions** *if W maps exactly t elements in the domain to every $x \in U$ and*

$$\text{for all } 1 \leq j < tN, \qquad d_U(W(j), W(j+1)) = 1$$

A **segment** *of the walk W is a restriction of W to a sub-interval of the domain $[1 \ldots tN]$.*
The walk W **connects** *two nodes $x, y \in U$ within δ steps if there exist $j \in W^{-1}(x)$ and $k \in W^{-1}(y)$ such that $|j - k| \leq \delta$.*
We say that W **visits** *at time j the node $W(j)$.*
The walk W **visits twice** *a node x within δ steps if there exist $j, k \in W^{-1}(x)$ such that $|j - k| \leq \delta$.*

Lemma 6 *Let T_n be the $n \times n$ torus (or mesh), and let W be a walk on T_n with 2 repetitions. If every pair of nodes x, y in a $k \times k$ square is connected by W within $c(d_2(x,y))^2$ steps, then there is a node in the square which is visited by W twice within k steps.*

Proof. Consider the middle point q of the square, and look at a segment s_1 of length $k/2$ on the walk such that q is in its middle. This segment cannot get out of the square. We assume that this segment does not contain any point twice, as otherwise we are done. We also assume w.l.o.g. that the projection of s_1 on the x-axis is larger than on the y-axis. Denote this projected interval by I, then $|I| \geq \sqrt{|s|} = \sqrt{k/2}$. As in the proof of Lemma 5, we remove from I intervals of $\frac{1}{10}\sqrt{k}$ around the endpoints E, F of s_1, and get at least one sub-interval $I' = [a \ldots b]$ of size $|I'| \geq \frac{1}{6}\sqrt{k}$. Let E', F' be the highest nodes of s_1 in the columns a, b, respectively, and restrict s_1 to the sub-segment t_1 between E', F'.

Let y be a node above t_1 in the central column of I' whose distance from t_1 is $l = \sqrt[5]{k}$. Trivially, such y must exist and let x be its closest node on t_1, so $d_2(x, y) = l$. Define S_y as the $l \times l$ square around y (y is in the center of S_y). Then the nodes of S_y are not visited by t_1.

Now consider the other occurrence of x on W (i.e. outside t_1). This must be of distance more than k from the first occurrence of x (on t_1), as otherwise we are done. All points of S_y are at distance of at most $2l$ from x on the mesh. However, S_y points are outside t_1, and the first occurrence of x (on t_1) is roughly in the middle of I', so this occurrence of x is not connected within $4cl^2$ to any of S_y points (recall that $\sqrt{k} >> 4cl^2$). It follows that all S_y points must appear in a $4cl^2$ segment around the second occurrence of x (outside t_1).

Consider now a larger segment of length $9cl^2 << k$ around the second occurrence of x (outside t_1). No node is visited twice in this segment, as otherwise we are done. Hence all visits of W to S_y can be partitioned into two distinct regions on W - those at distance at most $4cl^2$ from x (exactly one occurrence for each node in S_y), and those of distance more than $9cl^2$ from x. Note that the two regions are separated by at least $5cl^2$ nodes, so no pair of S_y nodes is connected within $4cl^2$ using occurrences in different regions. In other words, the occurrences which do connect a pair of nodes $x, y \in S_y$ belong to the same region in W.

Hence, we can split W to the two regions, remove the nodes which are not from S_y, and get two paths which handle all pairs of nodes in S_y with a quadratic distance expansion. This contradicts Lemma 5, where the case of two paths is considered. (Removing nodes outside of S_y results in permutations which are continuous at internal nodes of S_y but not necessarily on its boundary. Lemma 5 holds also in this weaker case. See remark at the end of Section 2.3.) □

Corollary 7 *Let T_n be the $n \times n$ torus (or mesh), and let W be a walk on T_n with 2 repetitions. Then there exist $x, y \in T_n$, which are not connected by W within $c(d_2(x,y))^2$.*

Proof. Let $k = \sqrt[3]{n}$. Consider $(n/2k)^2$ squares on T_n, where the side-length of each square is k, and the distance between neighboring squares is k. (See figure 9).

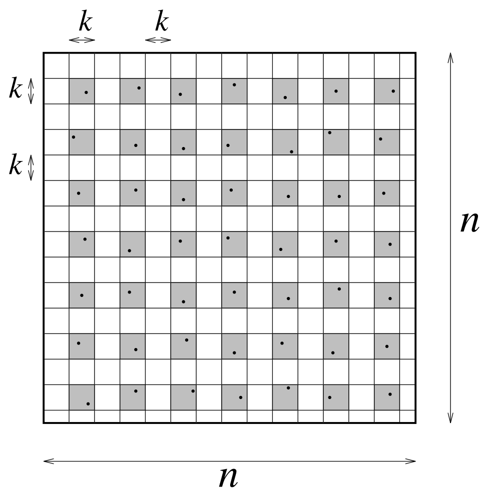

Figure 9. The squares and their representative form a "mesh"

In each square, Lemma 6 guarantees the existence of a node which is visited by W twice within k steps. Select one such node from each square, and call it the *square representative*. The distance between distinct squares is at least k, so visits of W to distinct square representatives do not interleave. Number the square representatives from 1 to $(n/2k)^2$, in the order that the walk W visits them. Call this indexing π'. We shall apply the argument of Lemma 1 on the square representatives and their indexing π'.

Consider a pair of square representatives whose distance on π' is $n^2/8k^2$. Since the squares (i.e. the nodes selected from them) form a "mesh", it is possible to connect any pair of squares by a sequence of $\frac{n}{k}$ neighboring squares (i.e. successive squares in the sequence are neighbors). In this sequence there must be at least one pair of neighboring squares which are separated by at least $\frac{n}{8k}$ steps on π'. The corresponding square representatives x, y are at least $n/8$ steps apart on W, since every transition in π' translates to at least k steps in W (recall that selected nodes from neighboring squares are at least k steps apart on W).

However, these square representatives are at distance at most $3k$ apart on T_n. Hence the expansion of W for this pair of representatives is more than quadratic. □

2.6 Upper Bound for Higher Dimensional Torus

The construction of stereoscopic families of Hamiltonian paths can be extended to higher dimensions.

Lemma 8 *Let T_n be an m-dimensional torus ($n \times n \times \cdots \times n$), for $n = s \cdot 2^l$, arbitrary $l > 0$ and odd $s \in \{m+1, m+2\}$. Then there exists a stereoscopic family of Hamiltonian paths on T_n, $\mathcal{F}^{(m)} = \{\pi_1, \pi_2, \ldots, \pi_{m+1}\}$, such that for all $x, y \in T_n$*

$$\min_{i=1,2,\ldots,m+1} \{d_1(\pi_i(x), \pi_i(y))\} \le (2(m+2) \cdot d_m(x,y))^m$$

where d_1 denotes distance on the (one dimensional) path, and d_m denotes L_1 distance on the (m-dimensional) torus.

Proof. Extends the proof of Lemma 3. Details omitted. □

The paths used in this construction also have a complementary property. When the nodes of the line are placed on the mesh along the path, their distances on the line shrink to their m-th root order.

Lemma 9 *Let π be the path constructed recursively in the proof of Lemma 8. Then π is a shrinkable numbering of vertices with $\alpha = \frac{1}{m}$, and $\beta = 2(m+1)$. That is*

$$\forall x, y \qquad d_m(x,y) \le 2(m+1) \sqrt[m]{d_1(\pi(x), \pi(y))}$$

Proof. Extends the proof of Lemma 4. Details omitted. □

3 Routing along Stereoscopic Families

We are interested in *simple* routing algorithms for the two dimensional torus (with bidirectional edges), for which the number of time steps that a packet p spends in the network depends only on d_p, the distance between the packet's source and destination, and is independent of the size of the network.

An elementary Store-and-Forward algorithm can route packets in the two dimensional mesh (or torus) in $O(d_p^2)$ steps. We assume that every node can be the origin of at most one packet. Packets are routed along an arbitrary shortest path. In case of contention, priority is given to packets whose distance from origin to destination (i.e. d_p) is minimal, and other packets wait. A packet p is delayed only by higher priority packets whose origin is at most d_p steps from p's route. The number of such origins is at most $O(d_p^2)$.

We would like an even simpler algorithm, in the sense that packets are never stored at intermediate locations (called "hot potato" or "deflection" routing [3, 4]). Even with this added requirement, there is a relatively simple solution for the case of batch routing, in which all nodes inject their packets to the network in the same time step.

The idea (on the torus) is to route each packet in a spiral path (a snail), starting at its origin, as in figure 10. In this case, every node can be the source of up to 4 packets, each injected on a spiral of a different orientation (initial direction). Simple analysis based on timing considerations (omitted in this version) shows that no two packets will want to cross the same edge at the same time. Every single spiral path covers the whole torus, so each packet will reach its destination within $O(d_p^2)$ steps.

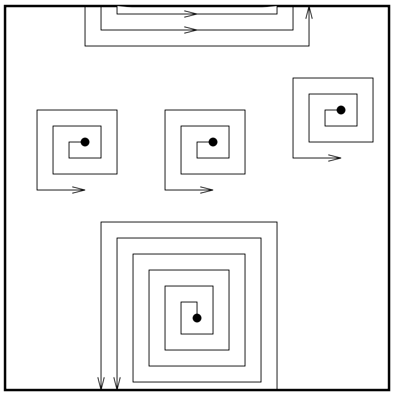

Figure 10. Batch routing along spirals

Edges in the torus belong to different spirals at different time steps. Therefore, the routing table in each node must depend either on the time t, or alternatively, on the source of each incoming packet. With this information available, the node can decide how to assign an incoming packet to an outgoing edge. Observe that the packet's destination is only used for checking whether the packet has reached its destination.

The delicate timing considerations prevent extension of this algorithm to dynamic routing. In the dynamic case, nodes generate packets continuously, so packets are injected to the network at different times. Spiral paths generated at different time steps may interfere with each other, producing contention on links.

We now suggest a different hot potato routing algorithm, based on stereoscopic families of permutations. We assume that each link (edge) is composed of three channels so it can transfer 3 packets in a single time step. Such networks can be implemented by multiplexing 3 packets on each link. In this network model, we suggest that packets travel only along the 3 Hamiltonian paths of a corresponding stereoscopic family of permutations, $\mathcal{F} = \{\pi_1, \pi_2, \pi_3\}$. Since the capacity of each link is 3, packets traveling along different paths do not interfere with each other. (It was our original hope that fewer Hamiltonian paths would suffice for the results stated below, but the lower bounds of Lemma 1 and Lemma 5 exclude this possibility.)

Consider batch routing where each node is the source of at most one packet. Then each packet will be injected to the network on its preferred path $\pi_i \in \mathcal{F}$ (the one along which its distance to destination is smallest), and travel along this path until reaching its destination. The routing is collision-free since the 3 channels enable the coexistence of the 3 paths, without any contentions. According to Lemma 8, each packet will reach its destination in $O(d_p^2)$ steps, and with no deflections.

The batch routing can be easily extended to dynamic routing by allowing packets injection, on a vacancy basis. A node generating a new packet, finds the preferred (best) path according to its destination, and waits for an opportunity to inject the packet along this path. From the moment the packet is injected, it travels along its path with no further delays, reaching its destination within the next $\Theta(d_p^2)$ steps. However, the overall time required to deliver a packet to its destination might be much larger, as there is no a-priori bound on the number of time steps that a packet is forced to wait in its source node until it can be injected into the network.

It is straightforward to obtain rigorous results regarding the performance of the algorithm in the dynamic case, most notably, in the *adversarial queueing model* of [5]. Assume that at every time step, an adversary generates packets on the mesh with the following restriction: if a packet is generated at node s and its destination is d steps away, then no other packet is generated in the same time step at a node at distance less than $6d$ from s. Using the fact that routing along stereoscopic families of permutations is essentially routing along paths, and using the results of [5] for routing along paths, it can be shown that our algorithm is stable against such an adversary. Namely, the number of undelivered packets at any time step remains bounded by some constant that depends on the mesh size but not on the number of time steps that elapsed.

We remark that for the adversarial model, [5] observe that by the analysis of [10], the *one bend algorithm* on the mesh is stable, and this holds for loads that are higher than those that our algorithm can handle. The new aspect in our result is the fact that queues at intermediate vertices are bounded (either no queues at all, if each link can carry three packets, or queues of bounded size, in the case of time multiplexing of the packets).

The main advantage of routing along stereoscopic families is its simplicity. Initially, the injecting node assigns a path to the packet (possibly by a fixed table prepared in advance). At intermediate nodes, the assignment of packets to outgoing edges is trivial. If the packet is destined to the current node, no routing needs to be done. Otherwise, the incoming edge (or more precisely, the channel) alone defines the assignment to an outgoing edge, according to predefined decisions (which reflect the Hamiltonian paths). There is no need to consider the destination of the packet, nor make any calculations. The trivial routing table requires minimal computational resources (both time and memory).

4 Applications to Hashing

A **dictionary** is a data structure for storing elements from universe U in memory M. It has to be capable of storing any subset $D \subset U$ whose cardinality is not too big, and provide efficient implementation of the following operations:

- Queries: Membership (given $x \in U$ decide whether $x \in D$) and Find (determine where x is stored in M).
- Update operations: Add/Delete/Change elements in D.

We consider the "noisy" version of the problem, where Linial and Sasson [12] provide the following definitions. Let U be a metric space with distance function d_U, reflecting the structure of the inputs. Then the dictionary has to provide the following operations:

- Queries: Membership (given $x \in U$ and $\Delta > 0$ (noise/uncertainty), list all items $y \in D$ which are close to x, namely $d_U(x, y) \leq \Delta$); and Find (determine where are these items stored in M).
- Update operations: Add/Delete/Change elements in D.

Any dictionary can be adapted to deal with noisy data as well. Given $x \in U$, check for every y in the Δ-neighborhood of x (denoted by Δ_x), whether it is stored in the dictionary. However, each check of $y \in \Delta_x$ requires an application of the hashing function. Overall, the procedure may require $\Omega(|\Delta_x|)$ applications of the hashing function. Moreover, elements $y \in \Delta_x$ might be hashed to distant locations in the memory. Large memory is usually paged, in which case the procedure might require access to $\Omega(|\Delta_x|)$ different pages.

The problem of hashing noisy data appears in several variants in many applications (see [13, 8, 6, 15] and references therein). Linial and Sasson [12], devise a "noisy" hashing scheme for a one dimensional universe U and a one dimensional memory M, with any noise measure Δ. It is a non-expansive hashing scheme, i.e. one which translates every Δ-neighborhood in U to a constant number of Δ-neighborhoods in M. Linial and Sasson define a specific family $\mathcal{H}$ of functions with the property that for every $x \in U$, either $f(x+1) = f(x)+1$ or $f(x+1) = f(x)-1$. Such function is a long path with "turning points" on the interval U, and can be specified by its "starting point" and its "turning points". They restrict $\mathcal{H}$ to functions f whose "turning points" are selectively chosen. Hashing the dictionary D then uses several hashing tables, each corresponding to a different hash function $f \in \mathcal{H}$.
We introduce the following notation:

$H_{1,1}$ hashing scheme for the one dimensional universe and one dimensional memory.

$H_{m,q}$ hashing scheme for m-dimensional universe and q-dimensional memory, with L_1 distance.

$C^{Member}_{m,q}(N, \Delta)$ number of pages (blocks of size $|\Delta_x|$) accessed by a Membership query of the $H_{m,q}$ scheme.

$C^{Update}_{m,q}(N)$ number of pages accessed by an Update operation (such as Add, Delete and Change) of the $H_{m,q}$ scheme.

$|M_{m,q}|$ memory complexity of the $H_{m,q}$ hashing scheme.

The non-expansive hashing scheme [12] is a $H_{1,1}$ scheme. For arbitrary $\epsilon > 0$, it uses $O(\log \frac{1}{\epsilon})$ hashing tables, each of size $|D|^{\frac{1}{1-\epsilon}}$, with the following complexity:

$$C^{Member}_{1,1}(N, \Delta) = C^{Update}_{1,1}(N) = O(\log \frac{1}{\epsilon})$$

$$|M_{1,1}| = O(\log \frac{1}{\epsilon} \cdot |D|^{\frac{1}{1-\epsilon}})$$

We show how to transform any $H_{1,1}$ scheme (and in particular, that of Linial and Sasson) into an $H_{m,q}$ scheme, with arbitrary m and q. Our $H_{m,q}$ scheme is based on a stereoscopic family of permutations on the m-dimensional torus, and a shrinkable numbering of vertices in the q-dimensional mesh (or torus). Lemma 8 guarantees the existence of a family $\mathcal{F} = \{\pi_1, \ldots, \pi_{m+1}\}$ of paths, which are, in particular, one-to-one mappings from the m-dimensional universe to the integer line. Lemma 9 shows how to construct shrinkable numbering of vertices.

The dictionary D is first mapped to $m+1$ separate images on the (one dimensional) interval, one for each path π_i ($i = 1, \ldots, m+1$). Each of these $m+1$ intervals is then hashed by the non-expansive hashing $H_{1,1}$ to another one dimensional representation. Finally, the one dimensional tables are, in turn, mapped by the shrinkable numbering of vertices to the q-dimensional mesh (or torus).

Membership query is straightforward. Given an input $x \in U$, find its $m+1$ images in the q-dimensional memory, and check their corresponding neighborhoods. The stereoscopic family of permutations $\mathcal{F}$ enlarges distances to $O(\Delta^m)$. The non-expansive hashing does not expand distances, and the shrinkable numbering shrinks distances to $O(\Delta^{m/q})$. So, overall

$$C^{Member}_{m,q}(N, \Delta) = (m+1) \cdot C^{Member}_{1,1}(N, O(\Delta^{m/q}))$$

$$C^{Update}_{m,q}(N) = (m+1) \cdot C^{Update}_{1,1}(N)$$

The stereoscopic family of permutations $\mathcal{F}$ copies $m+1$ times each dictionary element (or its index or a pointer to the element). The shrinkable numbering of vertices requires the same size of memory as its domain. Hence, the memory complexity of scheme $H_{m,q}$ is $|M_{m,q}| = (m+1)|M_{1,1}|$.

We now consider possible modifications of the approach described above. First note that unlike the routing application of Section 3, here we do not need each of the individual permutations to be continuous. Hence rather than base our stereoscopic family of permutations on the Hilbert space filling curve, we can use Morton order [14], which is much easier to manipulate. Similar to the Hilbert order, the Morton order recursively subdivides the plane into square tiles, and then covers each tile by recursively covering the tiles contained in it. However, the order in which tiles are covered is different, resulting in a simpler rule for the mapping from the plane to the line. This rule (bit interleaving) works in arbitrary dimensions. To get a stereoscopic family of permutations based on the Morton ordering, use $m + 1$ shifted versions of it, as we did for the Hilbert ordering.

The use of a stereoscopic family of permutations in the context of hashing has the disadvantage that each element is hashed $m+1$ different times, and memory requirements are increased by a factor of $m+1$. This can be avoided by taking just one member of the family (e.g., one Morton ordering), and then each element is hashed only once, and no overhead in memory is required. But then, in the Δ-neighborhood of an element in the m-dimensional mesh there may be up to 2^m different tiles that are m-cubes of side-length roughly Δ, resulting in access to potentially 2^m pages per membership query. Possibly, this number of pages can be reduced drastically by choosing a different indexing scheme. In particular, [1] design an indexing scheme that maps every two dimensional interval into only three one dimensional intervals. Perhaps such an indexing scheme can be generalized to an arbitrary number of dimensions. Alternatively, instead of recursively subdividing the m-dimensional mesh into m-dimensional cubes, one can use tiles of other shapes, with the property that any interval is completely covered by $m+1$ tiles of comparable size. Self similar tilings with this property are known for the two dimensional mesh (these tilings are derived from the hexagon), but we are not aware of work on this in higher dimensions.

It appears to us that when the dimension is low, it is preferable to use only one indexing scheme (and not pay the penalty of elements duplication), but that for high dimensions, there may be advantages to using stereoscopic families of permutations, so as to avoid the exponential blowup in number of pages accessed.

Independently of our work, Indyk et al. [9] proposed a different way of constructing hash functions that preserve locality in multidimensional spaces. They devise an $H_{m,q}$ scheme for $m = q$.

Acknowledgements

We thank Nati Linial and Ori Sasson for helpful discussions.

References

[1] T. Asano, D. Ranjan, T. Roos, E. Welzl, and P. Widmayer. Space filling curves and their use in the design of geometric data structures. *Proceedings of the Second Intern. Symp. of Latin American Theoretical Informatics Latin'95, LNCS 911*, pages 36–48, 1995.

[2] B. Awerbuch, S. Kutten, and D. Peleg. On buffer-economical store-and-forward deadlock prevention. *IEEE Transactions on Communications*, 42(11):2934–2937, November 1994.

[3] P. Baran. On distributed communication networks. *IEEE Transactions on Communications*, pages 1–9, 1964.

[4] A. Borodin and J. Hopcroft. Routing, merging and sorting on parallel models of computation. *JCSS*, 30:130–145, 1985.

[5] A. Borodin, J. Kleinberg, P. Raghavan, M. Sudan, and D. Williamson. Adversarial queueing theory. *Proceedings of the 37th Annual IEEE Symposium on Foundations of Computer Science*, pages 376–385, 1996.

[6] D. Dolev, Y. Harari, N. Linial, N. Nisan, and M. Parnas. Neighbohood preserving hashing and approximate queries. *5th ACM Symposium on Discrete Algorithms*, pages 251–259, 1994.

[7] U. Feige. Observations on hot potato routing. *Proc. of third Israel Symposium on the Theory of Computing and Systems*, pages 30–39, January 1995.

[8] D. Greene, M. Parnas, and F. Yao. Multi-index hashing for information retrieval. *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, 1994.

[9] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-preserving hashing in multidimensional spaces. *Proceedings of the 29th Annual Symposium on Theory of Computing*, To appear 1997.

[10] T. Leighton. Average case analysis of greedy routing algorithms on array. *Proceedings of the 2nd Annual ACM Symposium on Parallel Algorithms and Architectures*, pages 2–10, 1990.

[11] A. Lempel and J. Ziv. Compression of two-dimensional data. *IEEE Trans. Inform. Theory*, IT-32(1):2–8, 1986.

[12] N. Linial and O. Sasson. Non-expansive hashing. *Proceedings of the 28th Annual Symposium on Theory of Computing*, pages 509–518, 1996.

[13] M. Minsky and S. Papert. *Perceptrons*. MIT Press, Cambridge, Massachusetts, 1969.

[14] G. Morton. A computer oriented geodetic database and a new technique in file sequencing. *IBM, Ottawa, Canada*, 1966.

[15] M. Parnas. *Robust Algorithms and Data structures for information retrieval*. Ph.D. dissertation, Hebrew University, Jerusalem, Israel, 1994.

[16] H. Sagan. *Space Filling Curves*. Universitext, Springer-Verlag, 1994.

[17] H. Samet. *The design and analysis of spatial data structures*. Addison-Wesley Publishing Company, Inc., 1989.

[18] Q. Stout. Topological matching. *Proceedings of the 15th Annual Symposium on Theory of Computing*, pages 24–31, 1983.