

Compressive sensing using locality-preserving matrices

Elyot Grant*

Piotr Indyk*

December 4, 2012

Abstract

Compressive sensing is a method for acquiring high-dimensional signals (e.g., images) using a small number of linear measurements. Consider an n -pixel image $x \in \mathbb{R}^n$, where each pixel p has value x_p . The image is acquired by computing the measurement vector Ax , where A is an $m \times n$ measurement matrix, for some $m \ll n$. The goal is to design the matrix A and the recovery algorithm which, given Ax , returns an approximation to x . It is known that $m = O(k \log(n/k))$ measurements suffices to recover the k -sparse approximation of x . Unfortunately, this result uses matrices A that are random. Such matrices are difficult to implement in physical devices.

In this paper we propose compressive sensing schemes that use matrices A that achieve the near-optimal bound of $m = O(k \log n)$, while being highly “local”. We also show impossibility results for stronger notions of locality.

*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, {elyot,indyk}@mit.edu

1 Introduction

In recent years, a new “linear” approach for acquiring digital images has been discovered [CRT06, Don06]. Traditional approaches to image acquisition first capture an entire n -pixel image and then process it for compression, transmission, or storage. In contrast, the new approach obtains a compressed representation directly, by acquiring a small number of nonadaptive linear measurements of the signal in hardware. Formally, for an image represented by a vector x , the representation is equal to Ax , where A is an $M \times n$ matrix. The advantage of this architecture is that it can use fewer sensors, and therefore can be cheaper and use less energy than a conventional camera [DDT⁺08, FTF06, Rom09].

In order to reconstruct the image x from a lower-dimensional measurement vector (or sketch) Ax , one needs to assume that the image x is k -sparse for some k (i.e., it has at most k non-zero coordinates) or at least be “well-approximated” by a k -sparse vector¹. Then, given Ax , one finds (an approximation to) x by performing *sparse recovery*. The latter problem is typically defined as follows: construct a matrix A such that, for any signal x , we can recover a vector x^* from Ax that is “close” to the best possible k -sparse approximation of x . The notion of closeness is typically parametrized by $1 \leq q \leq p$, and we require that

$$\|x - x^*\|_p \leq C \cdot \text{Err}_k^q(x) / k^{1/q-1/p} \tag{1}$$

where $\text{Err}_k^q(x) = \min_{k\text{-sparse } x'} \|x - x'\|_q$ and C is the *approximation factor*. This is often referred to as the ℓ_p/ℓ_q guarantee. Note that if x is k -sparse, then for any q we have $\text{Err}_k^q(x) = 0$, and therefore $x^* = x$. Although the main focus of this paper is signal acquisition, sparse recovery has applications to other areas such as data stream computing [Mut05, Ind07].

In this paper, we focus on the ℓ_∞/ℓ_1 guarantee.² For this case, it is known [CM04] (cf. [GI10]) that there exist random binary matrices A with $M = O(k \log n)$ rows, and associated recovery algorithms that, with constant probability, produce approximations x^* satisfying Equation (1) with constant approximation factor C . The matrices are induced via a collection of random hash functions $h_1 \dots h_T$ where $h_i : [n] \rightarrow [m]$. Each hash function h defines an $m \times n$ binary matrix that contains a one in entry (i, j) if and only if the pixel corresponding to column j is mapped by h onto the sensor corresponding to the row i . The final matrix is obtained via vertical concatenation of the resulting matrices. As long as the hash functions h_i are chosen independently from a *universal* family, (where the probability of a collision between any pair of elements is $O(1)/m$), $T = O(\log n)$ hash functions are sufficient to achieve the desired guarantee. See Section 2.1 for further details.

Unfortunately, random matrices are not easily implementable in optical or digital hardware, requiring either a complex optical system or a complex network of wires. To circumvent this issue, various structured matrix constructions were proposed. In particular, the papers [UGN⁺09, TAN10, GIPR11] proposed a “geometric” construction of measurement matrices, in which the image is partitioned into $\sqrt{m} \times \sqrt{m}$ squares, which are then superimposed onto a $\sqrt{m} \times \sqrt{m}$ sensor array. This technique corresponds to a linear mapping from n dimensions to m dimensions, where the identified pixels are added together. The process is repeated several times with different values of m , and the resulting mappings are concatenated together.

¹Often, to achieve sufficient sparsity, the signal needs to be first transformed by representing it in an appropriate bases (e.g., wavelet or Fourier). We ignore this issue in this paper, since for the applications we focus on (star tracking or muzzle flash detection), the signals are sparse in the standard (pixel) basis.

²The ℓ_1/ℓ_∞ guarantee discussed here is stronger than the more popular ℓ_1/ℓ_1 guarantee, see [GI10] for an overview.

The geometric approach has been shown to be useful for sparse recovery and processing of point sources, such as stars [GIPR11], muzzle flashes [HPYI12] or tracked objects [TAN10]. However, the theoretical guarantees for this method are not fully satisfactory. In particular, it is not known whether the construction satisfies the ℓ_p/ℓ_q approximation guarantee of Equation 1. Instead, the paper [GIPR11] showed a recovery guarantee for a class of images that possess additional geometric structure, namely that contain a small number of distinguishable objects (e.g., stars) plus some noise. Moreover, the proof applied only to a variation of the geometric construction where the image was partitioned into pieces of constant size which were then pseudorandomly permuted. To the best of our knowledge, no recovery guarantees are known for general images.

1.1 Our Contribution

In this paper, we present two variants of the geometric construction, called *wrapping* and *folding*, that both support the ℓ_∞/ℓ_1 guarantee. Our constructions are randomized, and the guarantee holds with a constant probability. The key feature of our constructions is that they use only $O(k \log n)$ measurements, matching the bounds previously known only for unstructured matrices.

In *wrapping*, the $\sqrt{m} \times \sqrt{m}$ squares are superimposed directly. That is, all pixels with the same coordinates modulo \sqrt{m} are added together. This is the construction used in [GIPR11] and [HPYI12]. Note that the resulting mapping from the pixels onto the sensor array is discontinuous, as e.g., the neighboring pixels $(0, \sqrt{m} - 1)$ and $(0, \sqrt{m})$ are mapped to distant sensors. This issue does not occur in *folding*, where we flip alternate squares before superimposing them, as one does when folding a paper map. In order to achieve provable guarantees for these constructions, we randomize them using discrete affine distortions, described formally in Section 2.2. The distortions are very “local” (in particular, they are *Lipschitz*), which ensures that they are easily implementable in optical or digital hardware. Our constructions yield the following guarantees:

- For a randomized distortion followed by *wrapping*, we show that the resulting family of mappings from $[\sqrt{n}]^2$ into $[\sqrt{m}]^2$ is universal. This implies that $O(\log n)$ such mappings suffice to achieve the ℓ_∞/ℓ_1 guarantee with constant probability, yielding the $O(k \log n)$ measurement bound. Unfortunately, the wrapping operation is highly discontinuous.
- For a randomized distortion followed by *folding*, we show that $O(\log n)$ such mappings also suffice to achieve the ℓ_∞/ℓ_1 guarantee with constant probability, despite the fact that the resulting family of mappings is *not* universal. However, the mappings are Lipschitz.

Our first construction uses a family of mappings that is universal but not local (in particular, not Lipschitz), while our second construction uses a family of mappings that is local but not universal. Naturally, one might ask if there exists mappings that are both universal and local. In Section 3, we show that, for natural definitions of ‘local’ and ‘universal’, such mappings do not exist.

1.2 Related work

In addition to the aforementioned work on compressive sensing and sparse recovery, our work is related to the line of research on non-expansive and locality-preserving hashing [LS96, IMRV97]. The two aforementioned papers present constructions of hash functions that are both Lipschitz and “induce few collisions”. Specifically, the construction of paper [LS96] is 1-Lipschitz and universal, albeit it only works in one dimension. The construction of [IMRV97] is $O(1)$ -Lipschitz, but not

universal: for some pairs of points the probability of collision is $\omega(1/m)$. Both constructions are based on “non-uniform” overlapping, where the spacing between consecutive blocks is random (i.e., the superimposed parts of the grid $[\sqrt{n}]^2$ have different sizes). The construction of [IMRV97] uses an appropriately discretize random rotation before applying the non-uniform overlapping.

In connection to our work, we note that our proof in Section 2.4, which shows that randomized distortions followed by folding leads to sparse approximation guarantees, could be plausibly applied to the construction of [IMRV97] as well. However, the non-uniform folding employed in this construction increases its complexity, making it less appealing in applications.

1.3 Notation

We use $[n]$ to denote the set $\{0 \dots n-1\}$, and use $a \bmod b$ to denote the integer remainder obtained when dividing a by b . We define $d(x, y)$ to be the Euclidean distance between two points x, y . We say that a function f is Lipschitz with constant c if $d(f(x), f(y)) \leq c \cdot d(x, y)$.

2 Compressive sensing of arrays via local transformations

In this section, we describe our compressive sensing constructions. We shall consider an n -dimensional positive-valued signal $x \in \mathbb{R}^n$, and regard it as containing the intensity values of an image consisting of n square pixels. For simplicity in our exposition, we shall restrict ourselves to the case where the image itself is square, with dimensions \sqrt{n} by \sqrt{n} (where \sqrt{n} is an integer).

2.1 Sparse recovery and hashing

Our signal acquisition algorithms all employ hash functions $h : [\sqrt{n}]^2 \rightarrow [\sqrt{m}]^2$ that map keys, representing locations of pixels in the input image, to values, representing the locations of sensors in a rectangular array. These hash functions each define an $m \times n$ binary matrix A_h that contains a one in entry (i, j) if and only if the pixel corresponding to column j is mapped by h onto the sensor corresponding to row i . The matrix A_h contains a single one in every column and provides a complete representation of h . By randomly choosing $T = O(\log n)$ hash functions $h_1 \dots h_T$ from a carefully chosen distribution H , and then vertically concatenating the resulting A_{h_t} matrices, we may obtain a matrix A such that, with high probability, we can reconstruct an approximation x^* to x when given only Ax . The recovery is very simple: each coordinate x_j is estimated as

$$x_j^* = \text{median}_{t=1 \dots T}(A_{h_t}x)_{h_t(j)} \tag{2}$$

For the purposes of accurate recovery of a sparse approximation to x , a sufficient condition for the correctness of the above estimator is if the hash function distribution H is *universal*.

Definition 2.1. *Let $C \geq 1$ be a constant, and let H be a distribution over a family of hash functions, each from some finite domain \mathcal{D} of size n to any finite codomain \mathcal{R} of size m . Then H is called C -universal if, for all $a, b \in \mathcal{D}$ with $a \neq b$, we have $\Pr[h(a) = h(b)] \leq \frac{C}{m}$, where h is a hash function randomly chosen according to the distribution H .*

In this paper, we shall say that a hash function is *universal* whenever it is C -universal for some fixed constant C . The constant C shall be called the *universality constant*.

Let $x^{(k)}$ be a closest k -sparse approximation to x , i.e., $x^{(k)}$ contains the k largest entries of x , and is equal to 0 elsewhere. One can show the following [CM04] (cf. [GI10]):

Fact 2.2. Assume that H is a C -universal distribution of hash functions $h : [n] \rightarrow [m]$. Let $T \geq c \log(n)$ and let $m > c'k$, where c, c' are large enough constants depending on the universality constant C . Then, for each $j \in \{1 \dots n\}$, the estimator in Equation 2 satisfies

$$\Pr[|x_j - x_j^*| > \|x - x^{(k)}\|_1/k] < 1/n \quad (3)$$

Note that the number of rows of the sketch matrix A is $mT = O(k \log n)$.

Proof. We will briefly outline the argument of [CM04] (cf. [GI10]), as we will re-use it later. Let S be the support of $x^{(k)}$, $|S| = k$. Then, for $c' > 10$, one can observe that, for any j

$$\Pr[h(j) \in h(S - \{j\})] \leq 1/10 \quad (4)$$

and

$$E \left[\sum_{j' \notin S - \{j\}: h(j') = h(j)} |x_j| \right] \leq \frac{\|x - x^{(k)}\|_1}{10k} \quad (5)$$

Applying Markov's inequality to Equation 5 then yields

$$\Pr \left[\sum_{j' \notin S - \{j\}: h(j') = h(j)} |x_j| > \|x - x^{(k)}\|_1/k \right] \leq 1/10.$$

The guarantee then follows from the standard properties of the median estimator, and the fact that $1/10 + 1/10 < 1/2$. \square

A universal distribution H can easily be constructed by simply choosing a completely random hash function each time. As we shall see, by employing a geometric approach based on randomized affine distortions and wrapping, we can obtain the same result using far less randomness.

2.2 Hashing via affine transformations, folding, and wrapping

In this section, we shall define two randomized geometric hash functions—named *Distort-and-Wrap* and *Distort-and-Fold*—that each facilitate sparse recovery. Both hash functions map integer lattice points in $[\sqrt{n}]^2$ to integer lattice points in $[\sqrt{m}]^2$, and both require only $\Theta(\log(n))$ random bits. The reason that we provide two distinct examples is that, as we shall show, there is a necessary trade-off between locality properties and universality properties among such hash functions.

Both of the hash functions we introduce can be described as the composition of two maps:

- First, a *distortion* map, which randomly deforms the input array via a discretized affine transformation. Using a relatively simple family of transformations, we can distribute hash collisions sufficiently uniformly as to facilitate sparse recovery.
- Secondly, a *dimension reduction* map, which takes the distorted \sqrt{n} by \sqrt{n} array and maps each location into some cell of the final \sqrt{m} by \sqrt{m} array.

We employ the same distortion map in defining both *Distort-and-Wrap* and *Distort-and-Fold*, but the dimension reduction maps differ. The *Distort-and-Wrap* hash function achieves universality,

immediately implying that sparse recovery is possible for $m = \Theta(k \log n)$ via Fact 2.2. The *Distort-and-Fold* hash function is not universal, but exhibits stronger locality properties than the *Distort-and-Wrap* hash function—it is Lipschitz and preserves distances and areas locally, up to a constant factor. Despite not being universal, *Distort-and-Fold* still supports sparse recovery when $m = \Theta(k \log n)$ (though establishing this requires some additional work beyond applying Fact 2.2).

The randomized distortion map we use is defined as follows:

Definition 2.3. Define a DISTORT step to be a randomized mapping from \mathbb{Z}^2 to \mathbb{Z}^2 , taking

$$(x, y) \mapsto \left(x + \left\lfloor \frac{\lambda_x x}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+y)}{\sqrt{n}} \right\rfloor, y + \left\lfloor \frac{\lambda_y y}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+y)}{\sqrt{n}} \right\rfloor \right),$$

where λ_x , λ_y , and λ_{xy} are three random integers, each uniformly and independently selected, with replacement, from the set $[\sqrt{n}]$.

The DISTORT mapping, roughly speaking, is a discretized version of the operation performed via left multiplication by the following matrix:

$$M = \frac{1}{\sqrt{n}} \begin{pmatrix} \sqrt{n} + \lambda_x + \lambda_{xy} & \lambda_{xy} \\ \lambda_{xy} & \sqrt{n} + \lambda_y + \lambda_{xy} \end{pmatrix}$$

In practice, a DISTORT step could be simulated by a device that implements (e.g. via optical methods) the continuous linear transformation represented by M . Since $\frac{\lambda}{\sqrt{n}} \in [0, 1)$ for each randomly chosen λ in the expression above, we have $1 \leq \det(M) < 8$. Consequently, multiplication by M always preserves areas, up to a constant factor, without ever shrinking them. Moreover, the DISTORT step satisfies the following properties, as we show in the appendix:

Lemma 2.4. *The mapping produced by any DISTORT step is Lipschitz. In particular, its Lipschitz constant is at most 4.*

Lemma 2.5. *The mapping produced by any DISTORT step is one-to-one.*

After randomly distorting the input array, we perform an operation to reduce the size of the input from n to m . Our two hash functions arise from two possible methods of doing this:

Definition 2.6. Define a WRAP step to be a mapping from \mathbb{Z}^2 to $[\sqrt{m}]^2$ that maps each point (x, y) to $(x \bmod \sqrt{m}, y \bmod \sqrt{m})$.

Definition 2.7. Define a FOLD step to be a mapping from \mathbb{Z}^2 to $[\sqrt{m}]^2$, taking (x, y) to $(\text{fold}(x + \rho_x, \sqrt{m}), \text{fold}(y + \rho_y, \sqrt{m}))$, where, for positive integers a and b , the expression $\text{fold}(a, b)$ is defined to equal $a \bmod b$ whenever $(a \bmod 2b) < b$, and $b - 1 - (a \bmod b)$ otherwise. Here, ρ_x and ρ_y are random integers, uniformly and independently selected, with replacement, from the set $[\sqrt{m}]$.

Observe that the WRAP step is a deterministic operation, but the FOLD step incorporates a randomized shift, which shall be useful later for obtaining sparse recovery guarantees.

We note that wrapping produces “discontinuities” near locations mapped near the boundary of $[\sqrt{m}]^2$; for example, $(\sqrt{m} - 1, \sqrt{m} - 1)$ and (\sqrt{m}, \sqrt{m}) get mapped to distant locations. However, folding is more “continuous” than wrapping in the sense that it is a discretized version of a continuous mapping from $[0, \sqrt{n}]^2$ to $[0, \sqrt{m}]^2$. In particular, we observe the following:

Proposition 2.8. *The mapping produced by any FOLD step is Lipschitz with constant 1.*

We now define the two randomized hash functions we study herein, which are obtained by combining our randomized DISTORT operation with wrapping and folding:

Definition 2.9. *The Distort-and-Fold hash function consists of performing a DISTORT step followed by a FOLD step. The Distort-and-Wrap hash function consists of performing a DISTORT step followed by a WRAP step.*

Since every possible DISTORT transformation is Lipschitz with constant at most 4, and the FOLD step is Lipschitz with constant 1, we can immediately deduce the following:

Proposition 2.10. *Any Distort-and-Fold transformation is Lipschitz with constant at most 4.*

2.3 Sparse recovery guarantees for wrapping

In this section we show that the family of mappings obtained by composing randomized distortion and wrapping is universal. This implies that $O(\log n)$ such mappings suffice to achieve the ℓ_∞/ℓ_1 guarantee with constant probability, yielding the $O(k \log n)$ measurement bound.

Theorem 2.11. *Let H be the uniform distribution of all Distort-and-Wrap hash functions over all choices of constants λ_x , λ_y , and λ_{xy} selected during the DISTORT step. Then H is universal. In particular, H is C -universal for some universality constant³ $C \leq 91$.*

Proof. Let $h \in H$ be randomly chosen, and let λ_x , λ_y , and λ_{xy} be the three independently chosen parameters associated to h , each uniformly selected from $[\sqrt{n}]$. Let f be the underlying DISTORT operation used by h . Consider two distinct integer lattice points $P = (x, y)$ and $Q = (x + a, y + b)$, with $0 \leq x, y, x + a, y + b < \sqrt{n}$, and $(a, b) \neq (0, 0)$. Our goal will be to show that $\Pr[h(P) = h(Q)]$ is at most $\frac{C}{m}$. This is equivalent to showing that, with probability at most $\frac{C}{m}$, we will have $f(P)$ and $f(Q)$ congruent modulo \sqrt{m} in both their horizontal and vertical coordinates.

We begin by noting that if $d(P, Q) < \frac{\sqrt{m}}{4}$, then we must have $d(f(P), f(Q)) < \sqrt{m}$ by Lemma 2.4. However, since Lemma 2.5 implies that we cannot have $f(P) = f(Q)$, we must then have $h(P) \neq h(Q)$ in such a case, because $f(P)$ and $f(Q)$ can only be congruent modulo \sqrt{m} in both their horizontal and vertical coordinates if $f(P) = f(Q)$, or $d(f(P), f(Q)) \geq \sqrt{m}$. Accordingly, we shall henceforth assume that $d(P, Q) \geq \frac{\sqrt{m}}{4}$.

To proceed, we investigate the underlying structure of the DISTORT operation. Observe that, using vector arithmetic, we can write

$$f((x, y)) = (x, y) + \left\lfloor \frac{\lambda_x x}{\sqrt{n}} \right\rfloor (1, 0) + \left\lfloor \frac{\lambda_y y}{\sqrt{n}} \right\rfloor (0, 1) + \left\lfloor \frac{\lambda_{xy}(x + y)}{\sqrt{n}} \right\rfloor (1, 1)$$

and thus

$$\begin{aligned} f(Q) - f(P) = & (a, b) + \left(\left\lfloor \frac{\lambda_x(x + a)}{\sqrt{n}} \right\rfloor - \left\lfloor \frac{\lambda_x x}{\sqrt{n}} \right\rfloor \right) (1, 0) + \left(\left\lfloor \frac{\lambda_y(y + b)}{\sqrt{n}} \right\rfloor - \left\lfloor \frac{\lambda_y y}{\sqrt{n}} \right\rfloor \right) (0, 1) \\ & + \left(\left\lfloor \frac{\lambda_{xy}(x + y + a + b)}{\sqrt{n}} \right\rfloor - \left\lfloor \frac{\lambda_{xy}(x + y)}{\sqrt{n}} \right\rfloor \right) (1, 1). \end{aligned}$$

³In the proof of Theorem 2.11, we make little effort to optimize the universality constant C , instead opting for the clearest possible exposition.

Let Z_x be the integer-valued random variable equal to $\left\lfloor \frac{\lambda_x(x+a)}{\sqrt{n}} \right\rfloor - \left\lfloor \frac{\lambda_x x}{\sqrt{n}} \right\rfloor$, and consider its distribution as λ_x varies. Let $S_x = \{0, \dots, a\}$ if $a \geq 0$, and let $S_x = \{-a, \dots, 0\}$ otherwise. We observe that the support of Z_x is contained in the set S_x , and for each $t \in S_x$, we have

$$\Pr[Z_x = t] \leq \frac{\left\lceil \frac{\sqrt{n}}{|a|} \right\rceil}{\sqrt{n}} \leq \frac{1}{|a|} + \frac{1}{\sqrt{n}}.$$

Analogously, we define $Z_y = \left\lfloor \frac{\lambda_y(y+b)}{\sqrt{n}} \right\rfloor - \left\lfloor \frac{\lambda_y y}{\sqrt{n}} \right\rfloor$ and $Z_{xy} = \left\lfloor \frac{\lambda_{xy}(x+y+a+b)}{\sqrt{n}} \right\rfloor - \left\lfloor \frac{\lambda_{xy}(x+y)}{\sqrt{n}} \right\rfloor$, and note that for any integer t , we have $\Pr[Z_y = t] \leq \frac{1}{|b|} + \frac{1}{\sqrt{n}}$, and $\Pr[Z_{xy} = t] \leq \frac{1}{|a+b|} + \frac{1}{\sqrt{n}}$. Observe that Z_x , Z_y , and Z_{xy} are all independent, as λ_x , λ_y , and λ_{xy} are.

To finish the proof, we must use the above bounds on the randomness of Z_x , Z_y , and Z_{xy} to prove that there is a very low probability that $f(Q) - f(P)$ is divisible by \sqrt{m} in both its horizontal and vertical coordinates. To accomplish this, we use the assumption that $d(P, Q)$ is large to show that at least *two* of the three lengths $\{|a|, |b|, |a+b|\}$ must be large, which will imply that both the horizontal and vertical coordinates of $f(Q) - f(P)$ are unlikely to be divisible by \sqrt{m} , given the randomness introduced by our choice of values for Z_x , Z_y , and Z_{xy} .

Since we assumed that $d(P, Q) \geq \frac{\sqrt{m}}{4}$, we have $\sqrt{a^2 + b^2} \geq \frac{\sqrt{m}}{4}$, so one of $|a|$ or $|b|$ must be at least $\frac{\sqrt{m}}{4\sqrt{2}}$. Without loss of generality, we shall assume that $|a| \geq \frac{\sqrt{m}}{4\sqrt{2}}$. We consider two cases:

Case 1: $|a+b| \geq \frac{\sqrt{m}}{8\sqrt{2}}$. Fix $Z_y = t$, and consider what happens to $f(Q) - f(P) = (a, b+t) + Z_x(1, 0) + Z_{xy}(1, 1)$ as Z_x and Z_{xy} range over their respective distributions. To have $h(P) = h(Q)$, we must have $b+t + Z_{xy}$ and $a + Z_x + Z_{xy}$ both divisible by \sqrt{m} , which occurs if and only if both $Z_{xy} \equiv -b-t \pmod{\sqrt{m}}$ and $Z_x \equiv b+t-a \pmod{\sqrt{m}}$. Since these events occur independently, it suffices to bound their respective probabilities.

If $|a| \leq \sqrt{m}$, then there is only one value that Z_x can take on so that $Z_x \equiv b+t-a \pmod{\sqrt{m}}$, and our previous analysis of the distribution of Z_x can then be used to deduce that

$$|a| \leq \sqrt{m} \Rightarrow \Pr[Z_x \equiv b+t-a \pmod{\sqrt{m}}] \leq \frac{1}{|a|} + \frac{1}{\sqrt{n}} \leq \frac{4\sqrt{2}}{\sqrt{m}} + \frac{1}{\sqrt{n}} \leq \frac{7}{\sqrt{m}}.$$

However, if $|a| \geq \sqrt{m}$, then there could be up to $\left\lceil \frac{|a|}{\sqrt{m}} \right\rceil$ distinct values of Z_x for which $Z_x \equiv b+t-a \pmod{\sqrt{m}}$. In this case, we instead obtain the bound

$$|a| > \sqrt{m} \Rightarrow \Pr[Z_x \equiv b+t-a \pmod{\sqrt{m}}] \leq \left\lceil \frac{|a|}{\sqrt{m}} \right\rceil \left(\frac{1}{|a|} + \frac{1}{\sqrt{n}} \right) \leq \frac{2}{\sqrt{m}} + \frac{2|a|}{\sqrt{n}\sqrt{m}} \leq \frac{4}{\sqrt{m}},$$

where in the final inequality, we used the fact that $|a| \leq \sqrt{n}$. Combining both subcases, we obtain $\Pr[Z_x \equiv b+t-a \pmod{\sqrt{m}}] \leq \frac{7}{\sqrt{m}}$.

For Z_{xy} , our analysis is similar, and we obtain

$$|a+b| \leq \sqrt{m} \Rightarrow \Pr[Z_{xy} \equiv -b-t \pmod{\sqrt{m}}] \leq \frac{1}{|a+b|} + \frac{1}{\sqrt{n}} \leq \frac{8\sqrt{2}}{\sqrt{m}} + \frac{1}{\sqrt{n}} \leq \frac{13}{\sqrt{m}}$$

and

$$|a+b| > \sqrt{m} \Rightarrow \Pr[Z_{xy} \equiv -b-t \pmod{\sqrt{m}}] \leq \left\lceil \frac{|a+b|}{\sqrt{m}} \right\rceil \left(\frac{1}{|a+b|} + \frac{1}{\sqrt{n}} \right) \leq \frac{2}{\sqrt{m}} + \frac{2|a+b|}{\sqrt{n}\sqrt{m}} \leq \frac{6}{\sqrt{m}},$$

using the fact that $|a + b| \leq 2\sqrt{n}$. Therefore, we conclude that $\Pr[Z_{xy} \equiv -b - t \pmod{\sqrt{m}}] \leq \frac{13}{\sqrt{m}}$, and hence $\Pr[h(P) = h(Q)] \leq \frac{91}{m}$.

Case 2: $|a + b| < \frac{\sqrt{m}}{8\sqrt{2}}$. Then since we assumed that $|a| \geq \frac{\sqrt{m}}{4\sqrt{2}}$, we must have $|b| \geq \frac{\sqrt{m}}{8\sqrt{2}}$ by the triangle inequality, and we can proceed similarly to how we did in Case 1. We fix $Z_{xy} = t$, and observe that $h(P) = h(Q)$ if and only if $Z_x \equiv -a - t \pmod{\sqrt{m}}$ and $Z_y \equiv -b - t \pmod{\sqrt{m}}$. By a similar argument to that used in Case 1, we have $\Pr[Z_x \equiv -a - t \pmod{\sqrt{m}}] \leq \frac{7}{\sqrt{m}}$, and $\Pr[Z_y \equiv -b - t \pmod{\sqrt{m}}] \leq \frac{13}{\sqrt{m}}$, from which it follows that $\Pr[h(P) = h(Q)] \leq \frac{91}{m}$. \square

2.4 Sparse recovery guarantees for folding

When wrapping is replaced by folding, our hash function no longer has the universality property. Specifically, for any two adjacent points $P = (x, y)$ and $Q = (x, y + 1)$, it is not difficult to see that the probability of their collision under the Distort-and-Fold hash function is $\Omega(1/\sqrt{m})$, not $O(1/m)$. This is because the first coordinates of P and Q remain equal with constant probability during the DISTORT operation, in which case they collide during folding with probability $1/\sqrt{m}$. However, the folding construction still satisfies the following weaker properties, which are sufficient to guarantee sparse recovery:

Lemma 2.12. *Fix n , let k be an integer with $0 < k \leq n$, and let m be a perfect square of size roughly $c'k$, where c' is a sufficiently large constant. Let h be a randomly chosen Distort-and-Fold transformation mapping into $[\sqrt{m}]^2$, and let h consist of a DISTORT function f followed by a FOLD function g , (i.e., $h(P) = g(f(P))$). Write $f = (f_x, f_y)$. Then there exist absolute positive constants C, C' such that:*

(1) *For any two distinct points $P, Q \in [\sqrt{n}]^2$, we have*

$$\Pr[f_x(P) = f_x(Q)] \leq \frac{C}{\|P - Q\|_\infty}, \quad (6)$$

and the same statement holds for f_y .

(2) *For any two points $P, Q \in [\sqrt{n}]^2$ with $\|P - Q\|_\infty > C'\sqrt{k}$,*

$$\Pr[h(P) = h(Q)] \leq \frac{1}{20k}. \quad (7)$$

Lemma 2.12 can easily be proven using the same techniques as those used in the proof of Theorem 2.11. See the appendix for a full proof.

Using Lemma 2.12, we can obtain the following sparse recovery guarantee for Distort-and-Fold:

Theorem 2.13. *Let $T \geq c \log(n)$ and let $m > c'k$, where c, c' are sufficiently large constants. Let h be a composition of a DISTORT function g and a FOLD function f , i.e., $h(P) = f(g(P))$. Then, for each $p \in \{1 \dots n\}$, the estimator in Equation 2 satisfies*

$$\Pr[|x_p - x_p^*| > \|x - x^{(k)}\|_1/k] < 1/n \quad (8)$$

Proof. Let P be the point in $[\sqrt{n}]^2$ corresponding to the index p . Let S' be the support of $x^{(k)}$ and S'' be the set of points $Q \in [\sqrt{n}]^2$ such that $\|P - Q\|_\infty \leq C'\sqrt{k}$; let $S = S' \cup S''$. By the arguments outlined in the proof of Fact 2.2, it suffices to show that Equations 4 and 5 hold.

We will first show that Equation 4 holds. Let $S''_x = \{Q : g_x(Q) = g_x(P)\}$ and $S''_y = \{Q : g_y(Q) = g_y(P)\}$. Note that both S''_x and S''_y are random variables defined by g .

Lemma 2.14. $E[|S''_x|] \leq c'''\sqrt{k}$ for some absolute constant c''' .

Proof. Let $S'' = \{Q_1 \dots Q_r\}$, and assume that $Q_1, Q_2 \dots Q_r$ are sorted in the order of increasing distance from P . By Lemma 2.12, we have

$$E[|S''_x|] \leq \sum_{i=1}^r \Pr[g_x(P) = g_x(Q_i)] \leq \sum_{i=1}^r \frac{C}{\|P - Q_i\|_\infty} = \sum_{\ell=1}^{C'\sqrt{k}} 8\ell C/\ell = 8CC'\sqrt{k},$$

where we used the fact that there are at most 8ℓ distinct points Q with $\|P - Q\|_\infty = \ell$. \square

The lemma for S''_y is similar. By Markov's inequality, it follows that $\Pr[|S''_x| + |S''_y| > 160c'''\sqrt{k}] \leq 1/80$. For sufficiently large m , each $Q \in S''_x \cup S''_y$ can collide with P under f with probability at most $1/\sqrt{m}$ (due to the random translation applied during the FOLD step). This collision probability is at most $1/80 \cdot \frac{1}{160c'''\sqrt{k}}$ for c''' large enough. Thus, conditioned on $|S''_x| + |S''_y| \leq 160c'''\sqrt{k}$, we have $\Pr[h(P) \in h(S''_x \cup S''_y - \{P\})] \leq 1/80$. It follows that

$$\begin{aligned} \Pr[h(P) \in h(S''_x \cup S''_y - \{P\})] &\leq \Pr\left[h(P) \in h(S''_x \cup S''_y - \{P\}) : |S''_x| + |S''_y| \leq 160c'''\sqrt{k}\right] \\ &\quad + \Pr\left[|S''_x| + |S''_y| > 160c'''\sqrt{k}\right] \\ &\leq 1/40. \end{aligned}$$

Moreover, all other points in S'' collide with P under f with probability at most $1/m$, so

$$\Pr[h(P) \in h(S'' - (S''_x \cup S''_y \cup \{P\}))] \leq \frac{|S''|}{m} \leq \frac{(C')^2 k}{m},$$

which is less than $\frac{1}{40}$ for c''' large enough. It follows that $\Pr[h(P) \in h(S'' - \{P\})] \leq \frac{1}{40} + \frac{1}{40} = \frac{1}{20}$.

Finally, by Lemma 2.12, for any $Q \notin S''$ we have $\Pr[h(P) = h(Q)] \leq \frac{1}{20k}$. Therefore, we have $\Pr[h(P) \in h(S' - S'' - \{P\})] \leq \frac{k}{20k} = \frac{1}{20}$ and thus $\Pr[h(P) \in h(S - \{P\})] \leq \frac{1}{20} + \frac{1}{20} = \frac{1}{10}$, so Equation 4 holds.

From this, Equation 5 by applying linearity of expectations, and the theorem follows. \square

3 Impossibility of Universality for Local Hash Functions

In this section, we prove an impossibility result that effectively rules out the construction of universal hash functions for images, if it is required that those functions are sufficiently “local”. Here, our notions of locality and universality are continuous: we consider functions that map the vertices of each pixel of a large image to locations in the continuous square region $[0, \sqrt{m}]^2$, so that each pixel is effectively mapped to a polygon in $[0, \sqrt{m}]^2$. We show that if such a mapping is appropriately “local”, then some pair of pixels must collide (i.e., overlap) with substantial probability.

In the following, we formalize these notions. Let H be a distribution over a family of functions from the domain $\mathcal{D} = [\sqrt{n}]^2$ to the continuous region $\mathcal{R} = [0, \sqrt{m}]^2$.

Definition 3.1. For $h \in H$ and a point $P = (x, y) \in [\sqrt{n}]^2$, define the pixel $R_h(P)$ to be the convex hull of the four points $\{h(x-1, y-1), h(x-1, y), h(x, y-1), h(x, y)\}$. We say that two pixels $R_h(P)$ and $R_h(Q)$ collide whenever their interiors intersect.

Since our notion of collision is continuous, we need to redefine universality for this setting:

Definition 3.2. For any $C \geq 1$, we say that H is continuously C -universal if, for all points $P, Q \in \mathcal{D}$ with $P \neq Q$, we have $\Pr[R_h(P) \text{ collides with } R_h(Q)] \leq \frac{C}{m}$, where h is a function randomly chosen according to the distribution H .

Our notion of “locality” of a mapping is formalized as follows:

Definition 3.3. Let h be a function from $[\sqrt{n}]^2$ to $[0, \sqrt{m}]^2$. For $C \geq 1$, we define h to be C -approximately locally isometric whenever the following hold:

- (1) The function h is Lipschitz with constant at most C .
- (2) Each pixel $R_h(P)$ has area at least $\frac{1}{C}$.

The first condition is a prerequisite of any local mapping (the distances cannot expand too much). The second condition essentially states that, locally, the distances cannot shrink too much either. In particular, this rules out the possibility of projecting a “large” image into a “small” image by simply scaling it down. Note that the continuous version of our Distort-and-Fold mapping from Section 2.4 satisfies both conditions for a small value of C .

With the notions of locality and universality formalized, we now state our impossibility result:

Theorem 3.4. Let $C_1 \geq 1$ and $C_2 \geq 1$ be any constants. Then there exist sufficiently large values of m and n , dependent only on C_1 and C_2 , such no distribution H over a family of C_1 -approximately locally isometric hash functions from $\mathcal{D} = [\sqrt{n}]^2$ to $\mathcal{R} = [0, \sqrt{m}]^2$ is continuously C_2 -universal.

Proof. We define two points P and Q to be *adjacent* whenever $d(P, Q) = 1$, and say that two pixels are adjacent whenever the corresponding points are. Intuitively, the general idea behind our proof is that any C -approximately locally isometric mapping from \mathcal{D} to \mathcal{R} must create a large number of “creases” (or fold lines) in order to continuously embed the large n -pixel input region \mathcal{D} into the small range \mathcal{R} , which has area only m . These creases create collisions among adjacent pixels, and, as it turns out, create sufficiently many collisions that H cannot be continuously universal.

In the appendix, we prove the following using an elementary geometric argument:

Lemma 3.5. Let h be a C_1 -approximately locally isometric hash function from $\mathcal{D} = [\sqrt{n}]^2$ to $\mathcal{R} = [0, \sqrt{m}]^2$. Then the number of adjacent pairs of pixels that collide under h is at least $c \frac{n}{\sqrt{m}}$ for some absolute constant $c > 0$ that depends only on C_1 .

Using Lemma 3.5, it is easy to show that H cannot be continuously universal. We define S to be the set of all unordered pairs $\{P, Q\}$ of adjacent points in \mathcal{D} , noting that $|S| \leq 2n$. Lemma 3.5 implies that, for each h in the support of H , there are at least $c \frac{n}{\sqrt{m}}$ pairs $\{P, Q\} \in S$ such that $R_h(P)$ and $R_h(Q)$ collide. Therefore, by the pigeonhole principle, there must exist some pair of adjacent points $\{P, Q\} \in S$ such that, if h is randomly selected according to the distribution H , the probability that $R_h(P)$ and $R_h(Q)$ collide is at least $\frac{c/2}{\sqrt{m}}$. By selecting m to be sufficiently large, it then follows that H cannot be continuously C_2 -universal. \square

References

- [CM04] G. Cormode and S. Muthukrishnan. Improved data stream summaries: The count-min sketch and its applications. *Latin*, 2004.
- [CRT06] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1208–1223, 2006.
- [DDT⁺08] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, and R. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine*, 2008.
- [Don06] D. L. Donoho. Compressed Sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, 2006.
- [FTF06] R. Fergus, A. Torralba, and W. T. Freeman. Random lens imaging. *MIT CSAIL-TR-2006-058*, 2006.
- [GI10] A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of IEEE*, 2010.
- [GIPR11] R. Gupta, P. Indyk, E. Price, and Y. Rachlin. Compressive sensing with local geometric features. *SOCG*, 2011.
- [HPYI12] L. Hamilton, D. Parker, C. Yu, and P. Indyk. Focal plane array folding for efficient information extraction and tracking. *AIPR*, 2012.
- [IMRV97] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-preserving hashing in multidimensional spaces. *STOC*, 1997.
- [Ind07] P. Indyk. Sketching, streaming and sublinear-space algorithms. *Graduate course notes, available at <http://stellar.mit.edu/S/course/6/fa07/6.895/>*, 2007.
- [LS96] N. Linial and O. Sasson. Non-expansive hashing. *STOC*, 1996.
- [Mut05] S. Muthukrishnan. Data streams: Algorithms and applications). *Foundations and Trends in Theoretical Computer Science*, 2005.
- [Rom09] J. Romberg. Compressive sampling by random convolution. *SIAM Journal on Imaging Science*, 2009.
- [TAN10] V. Treeaporn, A. Ashok, and M. A. Neifeld. Increased field of view through optical multiplexing. *Optics Express*, 18(21), 2010.
- [UGN⁺09] S. Uttam, A. Goodman, M. A. Neifeld, C. Kim, R. John, J. Kim, and D. Brady. Optically multiplexed imaging with superposition space tracking. *Optics Express*, 17(3), 2009.

A Appendix

A.1 Proof of Lemma 2.4

Lemma 2.4 The mapping produced by any DISTORT step is Lipschitz. In particular, its Lipschitz constant is at most 4.

Proof. Consider two integer lattice points $P = (x, y)$ and $Q = (x + a, y + b)$, and let f be any DISTORT step with parameters λ_x , λ_y , and λ_{xy} . Since $\frac{\lambda}{\sqrt{n}} \in [0, 1)$ for any $0 \leq \lambda < n$, we have

$$\left\lfloor \frac{\lambda_x(x+a)}{\sqrt{n}} \right\rfloor - \left\lfloor \frac{\lambda_x x}{\sqrt{n}} \right\rfloor \leq a$$

and similarly for expressions involving λ_y and λ_{xy} . Consequently,

$$d(f(P), f(Q)) \leq \sqrt{(3a+b)^2 + (3b+a)^2} \leq \sqrt{16a^2 + 16b^2} = 4d(P, Q),$$

where, for the second inequality, we used the fact that $2ab \leq a^2 + b^2$. \square

A.2 Proof of Lemma 2.5

Lemma 2.5 The mapping produced by any DISTORT step is one-to-one.

Proof. Again, we consider two integer lattice points $P = (x, y)$ and $Q = (x + a, y + b)$, and let f be any DISTORT step with parameters λ_x , λ_y , and λ_{xy} . We assume that $f(P) = f(Q)$, with the goal of showing that $a = 0$ and $b = 0$, hence proving that $P = Q$. The assumption $f(P) = f(Q)$ implies that

$$\left\lfloor \frac{\lambda_x x}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+y)}{\sqrt{n}} \right\rfloor = a + \left\lfloor \frac{\lambda_x(x+a)}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+a+y+b)}{\sqrt{n}} \right\rfloor \quad (9)$$

and

$$\left\lfloor \frac{\lambda_y y}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+y)}{\sqrt{n}} \right\rfloor = b + \left\lfloor \frac{\lambda_y(y+b)}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+a+y+b)}{\sqrt{n}} \right\rfloor. \quad (10)$$

We note that if $a + b = 0$, then equations (9) and (10) can only be satisfied if $a = b = 0$. Consider instead the case where $a + b > 0$. In this case, at least one of a or b must be positive, so we may assume, without loss of generality, that $a > 0$. However, if both $a + b$ and a are positive, then equation (9) cannot be satisfied, as its left side would be strictly less than its right side. A similar contradiction occurs in the case where $a + b < 0$, so we must have $a = b = 0$, completing the proof. \square

A.3 Proof of Lemma 2.12

Lemma 2.12 Fix n , let k be an integer with $0 < k \leq n$, and let m be a perfect square of size roughly $c'k$, where c' is a sufficiently large constant. Let h be a randomly chosen Distort-and-Fold transformation mapping into $[\sqrt{m}]^2$, and let h consist of a DISTORT function f followed by a FOLD function g , (i.e., $h(P) = g(f(P))$). Write $f = (f_x, f_y)$. Then there exist absolute positive constants C, C' such that:

(1) For any two distinct points $P, Q \in [\sqrt{n}]^2$, we have

$$\Pr[f_x(P) = f_x(Q)] \leq \frac{C}{\|P - Q\|_\infty}, \quad (11)$$

and the same statement holds for f_y .

(2) For any two points $P, Q \in [\sqrt{n}]^2$ with $\|P - Q\|_\infty > C'\sqrt{k}$,

$$\Pr[h(P) = h(Q)] \leq \frac{1}{20k}. \quad (12)$$

Proof. Both parts of this proof employ techniques similar to those used in the proof of Theorem 2.11. As we did there, we write $P = (x, y)$ and $Q = (x + a, y + b)$ with $(a, b) \neq (0, 0)$, and define the same independent random variables Z_x, Z_y , and Z_{xy} . We observe that at least two of the values in the set $\{|a|, |b|, |a + b|\}$ must be at least $\frac{\|P - Q\|_\infty}{2}$, and thus at least two of the three random variables in $\{Z_x, Z_y, Z_{xy}\}$ must have a large support. This forms the basis for the arguments we use to establish the lemma.

In proving (1), we must use the fact that at least one of $\{|a|, |a + b|\}$ is at least $\frac{\|P - Q\|_\infty}{2}$. We consider the case where $|a| \geq \frac{\|P - Q\|_\infty}{2}$; the case where $|a + b| \geq \frac{\|P - Q\|_\infty}{2}$ is similar. Fixing Z_y and Z_{xy} , we observe that there is at most one value that the random variable Z_x can attain that will cause $f_x(P)$ and $f_x(Q)$ to be equal. Since for any integer t , we have $\Pr[Z_x = t] \leq \frac{1}{|a|} + \frac{1}{\sqrt{n}}$, it follows that

$$\Pr[f_x(P) = f_x(Q)] \leq \frac{1}{|a|} + \frac{1}{\sqrt{n}} \leq \frac{3}{\|P - Q\|_\infty},$$

which establishes the result. For f_y , we proceed similarly.

We prove (2) for the case where $c' = 1280$ and $C' = 144$, making little effort to optimize the constants. If $\|P - Q\|_\infty > C'\sqrt{k} \approx \frac{144}{\sqrt{1280}}\sqrt{m} > 4\sqrt{m}$, then $\|P - Q\|_\infty > 4\sqrt{m}$ and thus at least two of $\{|a|, |b|, |a + b|\}$ are greater than $2\sqrt{m}$. We shall complete the proof assuming that both $|a|$ and $|b|$ are greater than $2\sqrt{m}$, but the other cases are similar. Fix $Z_{xy} = t$, and fix the two horizontal and vertical shift parameters so that $\rho_x = t_x$ and $\rho_y = t_x$. By the nature of the FOLD operation, as Z_x is allowed to vary, the horizontal coordinates $h_x(P)$ and $h_x(Q)$ of $h(P)$ and $h(Q)$ will be equal if and only if $Z_x \equiv -a - t \pmod{2\sqrt{m}}$ or $Z_x + t_x \equiv a + t - t_x \pmod{2\sqrt{m}}$. Therefore,

$$\Pr[h_x(P) = h_x(Q)] \leq 2 \left\lceil \frac{|a|}{\sqrt{m}} \right\rceil \left(\frac{1}{|a|} + \frac{1}{\sqrt{n}} \right) \leq \frac{8}{\sqrt{m}}.$$

The vertical coordinates of $h(P)$ and $h(Q)$ will be equal with the same probability. Since these events are independent, we have $\Pr[h(P) = h(Q)] \leq \frac{64}{m} \leq \frac{1}{20k}$, completing the proof. \square

A.4 Proof of Lemma 3.5

Lemma 3.5 Let h be a C_1 -approximately locally isometric hash function from $\mathcal{D} = [\sqrt{n}]^2$ to $\mathcal{R} = [0, \sqrt{m}]^2$. Then the number of adjacent pairs of pixels that collide under h is at least $c\frac{n}{\sqrt{m}}$ for some absolute constant $c > 0$ that depends only on C_1 .

Proof. Each of the $(\sqrt{n} - 1)^2$ pixels $R_h(P)$ is a convex polygon in $[0, \sqrt{m}]^2$ having three or four edges (fewer edges are not possible since each pixel has positive area). We give special names to some of these edges: define a *crease edge* to be an edge that is the boundary between two adjacent colliding pixels, and define a *boundary edge* to be any of the $4(\sqrt{n} - 1)$ edges that are *not* the border between two adjacent pixels. Let C be the set of all crease edges, and let B be the set of all boundary edges.

Next, we shall define a function α , taking point-pixel pairs of the form $(p, R_h(P))$, where p is a point in $R_h(P)$, to edges in $B \cup C$. We define $\alpha(p, R_h(P))$ algorithmically as follows: let ℓ_p be the horizontal line passing through p . Consider the process of moving rightward along ℓ_p until an edge e_0 of $R_h(P)$ is encountered. If e_0 is a crease or boundary edge, we set $\alpha(p, R_h(P)) = e_0$. If not, we let $R_h(P_1)$ be the pixel neighbouring $R_h(P)$ that also has e_0 as one of its edges, and continue moving rightward along ℓ_p , through the interior of $R_h(P_1)$, until a second edge e_1 of $R_h(P_1)$ is encountered. Again, if e_1 is a crease or boundary edge, we set $\alpha(p, R_h(P)) = e_1$, and if not, we continue through further pixels to the right of P_1 . This process must terminate, because some boundary or fold edge must be encountered before ℓ_p exits the square $[0, \sqrt{m}]^2$. We can ignore the points p for which this process is not well defined due to ℓ_p intersecting a vertex of one of the pixels, or colliding with a horizontal edge; such points comprise a set of measure zero, which will not be relevant during our analysis.

Given a boundary or crease edge $e \in B \cup C$ and a point $p \in [0, \sqrt{m}]^2$, we let $U(p, e)$ be the set of all pixels $R_h(P)$ with $p \in R_h(P)$ and $\alpha(p, R_h(P)) = e$. We claim that $|U(p, e)| \leq 2$. To see this, observe that the algorithm used to generate $\alpha(p, R_h(P))$ can be run in reverse, starting from $\ell_p \cap e$ and moving leftwards instead of rightwards. The only decision to be made is which pixel, of the two having e as an edge, to begin moving leftward in initially.

For $e \in B \cup C$, we define μ_e as a measure of the total area of all the point-pixel pairs $(p, R_h(P))$ with $\alpha(p, R_h(P)) = e$. Formally, we let

$$\mu_e = \sum_P \mu\{p \in R_h(P) : \alpha(p, R_h(P)) = e\},$$

where μ is the standard (e.g. Lebesgue) measure in \mathbb{R}^2 . Using the previous claim that $|U(p, e)| \leq 2$, we can see that $\mu_e \leq 2\sqrt{m} \|e\|_2$, since the area of all points $p \in [0, \sqrt{m}]^2$ with $\ell_p \cap e \neq \emptyset$ is at most $\sqrt{m} \|e\|_2$. It follows that $\mu_e \leq 2C_1\sqrt{m}$, since h is C_1 -approximately locally isometric.

We note that $\sum_{e \in B \cup C} \mu_e$ is simply the sum of the areas of all the pixels, which is at least $\frac{\Theta(n)}{C_1}$, since h is C_1 -approximately locally isometric. It follows that $|B \cup C| \geq \frac{\Theta(n)}{2C_1^2\sqrt{m}}$. Since $|B| = 4(\sqrt{n} - 1)$, it follows that $|C| \geq \frac{\Theta(n)}{\sqrt{m}}$, which yields the result. \square