# Dimension Reduction Algorithms for Near-Optimal Low-Dimensional Embeddings and Compressive Sensing

by

Elyot Grant

B.Math, University of Waterloo (2010)
M.Math, University of Waterloo (2011)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2013



Author .....
Department of Electrical Engineering and Computer Science
July 19, 2013

Certified by .....
Piotr Indyk
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

# Dimension Reduction Algorithms for Near-Optimal Low-Dimensional Embeddings and Compressive Sensing

by

Elyot Grant

Submitted to the Department of Electrical Engineering and Computer Science
on July 19, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

## Abstract

In this thesis, we establish theoretical guarantees for several dimension reduction algorithms developed for applications in compressive sensing and signal processing. In each instance, the input is a point or set of points in $d$-dimensional Euclidean space, and the goal is to find a linear function from $\mathbb{R}^d$ into $\mathbb{R}^k$, where $k << d$, such that the resulting embedding of the input pointset into $k$-dimensional Euclidean space has various desirable properties. We focus on two classes of theoretical results:

- First, we examine linear embeddings of arbitrary pointsets with the aim of minimizing *distortion*. We present an exhaustive-search-based algorithm that yields a $k$-dimensional linear embedding with distortion at most $\epsilon_{opt}(k) + \delta$ for any $\delta > 0$, where $\epsilon_{opt}(k)$ is the smallest possible distortion over all orthonormal embeddings into $k$ dimensions. This PTAS-like result transcends lower bounds for well-known embedding techniques such as the Johnson-Lindenstrauss transform.

- Next, motivated by compressive sensing of images, we examine linear embeddings of datasets containing points that are sparse in the *pixel basis*, with the goal of recoving a nearly-optimal sparse approximation to the original data. We present several algorithms that achieve strong recovery guarantees using the near-optimal bound of $O(k \log n)$ measurements, while also being highly "local" so that they can be implemented more easily in physical devices. We also present some impossibility results concerning the existence of such embeddings with stronger locality properties.

Thesis Supervisor: Piotr Indyk
Title: Professor of Electrical Engineering and Computer Science

# Acknowledgments

Portions of the work leading to this thesis were supported by grants from the Natural Sciences and Engineering Research Council of Canada, a grant from Draper Lab, an NSF CCF-1012042 award, the MADALGO project, and the Packard Foundation.

Parts of Chapters 3 and 4 were presented at the ACM 29th Annual Symposium on Computational Geometry (SoCG 2013).

# Contents

# Chapter 1

# Introduction

The so-called *curse of dimensionality* poses a central challenge in various signal processing problems including sensing, storage, transmission, and inference, as datasets often become increasingly difficult to organize and compute with as the number of dimensions increases. In many applications dealing with high dimensional metric spaces, an extremely useful tool to mitigate such problems is the notion of an *embedding* into a space of low dimension [14]. Such embeddings constitute a concise, yet faithful representation of the original metric space, and consequently enable the use of very efficient algorithmic tools and techniques in the smaller space.

In most situations, it is not necessary that the embedding preserves every aspect of the original pointset—it instead suffices that embeddings preserve whatever critical features of the original space are necessary to perform the computation at hand. Some examples include the following:

- *Neighbourhood-preserving projections and hashes*: Suppose that given a high dimensional data set and a set of queries, one wishes to find the nearest neighbours of those queries in the data set. For a set $V$ of $n$ points in $\mathbb{R}^d$, a naïve point query would incur $\mathcal{O}(nd)$ computations. On the other hand, one can bring this computational cost down to $\mathcal{O}(nk)$, if one can reduce the dimensionality from $d$ to $k$. This approach forms the core of *locality sensitive hashing* (LSH), a popular technique for pattern recognition and information

retrieval [15].

- *Compressive signal acquisition*: Instead of acquiring (or recording) a high dimensional signal (or image) $x \in \mathbb{R}^d$, the technique of compressive sensing (CS) prescribes recording only a few linear projections (or *measurements*) $y = \Phi x$. A rich, extensive theory specifying the types of allowable projection matrices $\Phi$, as well as efficient algorithms for recovering $x$ from $y$, has been developed; see, for example, the seminal papers of [3, 6]. The matrix $\Phi$ can be thought of as representing an embedding of $\mathbb{R}^d$ into $\mathbb{R}^k$, and, if chosen properly, such embeddings can facilitate various recovery guarantees, particularly in situations where $x$ is known to be sparse in a particular basis.

In this thesis, we will discuss two main algorithms for embeddings of high dimensional spaces. First, motivated by neighbourhood-preserving projections and hashes, we examine linear embeddings of arbitrary pointsets with the aim of minimizing the amount of *distortion* in the pairwise distances between points. Our main contribution in this area is a PTAS-like exhaustive-search-based algorithm that yields a $k$-dimensional linear embedding with almost-optimal distortion, with running time varying as the desired closeness to optimality increases. Chapter 2 discusses the exhaustive search algorithm for low distortion embeddings.

Secondly, motivated by compressive sensing of images, we examine linear embeddings of datasets containing points that are sparse in the *pixel basis*, with the goal of being able to recover a nearly-optimal sparse approximation to the original data from the embedded signal. We present several algorithms that achieve strong recovery guarantees using a nearly optimal number of measurements, while also being highly "local" so that they can be implemented more easily in physical devices. Chapter 3 contains these results.

Additionally, we also present an impossibility result concerning the existence of embeddings with stronger locality properties than those achieved in the results of Chapter 3. This result demonstrates that certain distributions of embeddings cannot both behave as universal hash functions while simultaneously representing continuous deformations of rectangular images into smaller rectangular regions. This impossibility theorem is presented in Chapter 4.

In the remainder of the introduction, we provide background information on low distortion embeddings, compressive sensing, and related concepts. Additionally, we introduce notation and motivate the specific problems studied in the remaining chapters.

## 1.1 Low Distortion Embeddings

Throughout this thesis, we focus on embeddings of high-dimensional point sets. Specifically, we consider inputs consisting of a set $V$ of $n$ vectors in $\mathbb{R}^d$, and seek embeddings of the form $f : \mathbb{R}^d \to \mathbb{R}^k$, where $k \ll d, n$. Here, we will consider the domain $\mathbb{R}^d$ and codomain $\mathbb{R}^k$ as being equipped with the Euclidean norm. Our goal is to minimize the *distortion* of the embedding $f$. Specifically, a function $f : \mathbb{R}^d \to \mathbb{R}^k$ is said to have distortion $\epsilon > 0$ if, for every $x \in V$,

$$1 - \epsilon \leq \frac{\|f(x)\|_2^2}{\|x\|_2^2} \leq 1 + \epsilon. \tag{1.1}$$

Our goal is to construct embeddings with as small a value of $\epsilon$ as possible. As intuition would suggest, there is a tradeoff between $k$, the number of dimensions of the embedding, and $\epsilon$, the distortion. A celebrated result by Johnson and Lindenstrauss [17] states that given any set $V$ of $n$ vectors in $\mathbb{R}^d$ and $\epsilon > 0$, if $k = O(\log n/\epsilon^2)$, then there exists an embedding $f : \mathbb{R}^d \to \mathbb{R}^k$ that satisfies (1.1). In fact, the embedding in this theorem is constructed by choosing a random orthonormal projection from $\mathbb{R}^d$ to $\mathbb{R}^k$, scaled by an appropriate factor; such functions $f$ may be called *J-L embeddings*. The random dimensionality reduction technique plays a foundational role in several areas, including high-dimensional similarity search and compressive sensing.

Unfortunately, the bound guaranteed by a J-L embedding cannot be improved (by much): by the result of [1], there exist sets of $n$ points that necessarily require $\Omega(\log n/\epsilon^2 \log(1/\epsilon))$ dimensions in order to be embedded with distortion at most $\epsilon$. However, real-world data often exhibit some low-dimensional structure, which can be potentially exploited to obtain an embedding of distortion $\epsilon$ using fewer dimensions. Thus, an intriguing question emerges: given a *particular* set $V$ possessing some hidden low-dimensional structure, is it possible

to examine $V$ and find an embedding into *fewer* than $O(\log n/\epsilon^2)$ dimensions, while still obtaining distortion $\epsilon$? As we discuss in Chapter 2, the answer is yes in a number of situations.

## 1.2 Compressive Sensing of Images

In recent years, a new "linear" approach for acquiring digital images has been discovered [4, 7]. 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 [8, 9, 22].

In order to reconstruct the image $x$ from a lower-dimension 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 vector[1]. 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

$$\|x - x^*\|_p \leq C \cdot \mathrm{Err}_k^q(x)/k^{1/q-1/p} \tag{1.2}$$

where $\mathrm{Err}_k^q(x) = \min_{k\text{-sparse } x'} \|x - x'\|_q$ and $C$ is the *approximation factor*. This is often referred to as the $\ell_p/\ell_q$ guarantee. Note that if $x$ is $k$-sparse, then for any $q$ we have $\mathrm{Err}_k^q(x) = 0$, and therefore $x^* = x$. Although the main focus of this paper is signal acquisition, sparse

[1]Often, 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.

recovery has applications to other areas such as data stream computing [13, 21].

In this paper, we focus on the $\ell_\infty/\ell_1$ guarantee. The $\ell_\infty/\ell_1$ guarantee discussed here is stronger than the more popular $\ell_1/\ell_1$ guarantee; see [10] for an overview. For this case, it is known [5] (cf. [10]) 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 (1.2) with constant approximation factor $C$. The matrices are induced via a collection of random hash functions $h_1 \ldots h_T$ where $h_i : [n] \to [m]$. 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 3.2 for further details.

Unfortunately, random matrices are not easy to implement 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 [11, 24, 25] 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.

The geometric approach has been shown to be useful for sparse recovery and processing of point sources, such as stars in astronomical images [11], muzzle flashes [12] or tracked objects [24]. However, the theoretical guarantees for this method are not fully satisfactory. In particular, it is not known whether the construction satisfies the $\ell_p/\ell_q$ approximation guarantee of Equation 1.2. Instead, the paper [11] 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.

## 1.3 Related work

The classical method to construct lower dimensional data representations is principal components analysis (PCA) [20], which involves orthogonally projecting a dataset into the subspace spanned by the top few eigenvectors of its covariance matrix. However, a global spectral technique such as PCA can potentially contract specific local distances, and hence cannot offer near-optimal distortion guarantees in general.

The optimal trade-offs between distortion and dimensions have been studied for many metrics [14]. Although most of those results were focused on the worst case distortion (along the line of the J-L theorem), there have been several works focused on designing algorithms that approximate the best distortion (see [23] and references therein). However, this research was focused on minimizing the distortion of *non-linear* embeddings, which is a much harder task. In particular, the minimum distortion of a non-linear embedding into a fixed-dimensional space is NP-hard to approximate, even up to a polynomial factor [19]. In contrast, our focus on linear and orthonormal embeddings enables us to obtain strong algorithmic results.

In addition to the aforementioned work on compressive sensing and sparse recovery, our work in Chapter 3 is related to the line of research on non-expansive and locality-preserving hashing [16, 18]. The two aforementioned papers present constructions of hash functions that are both Lipschitz and "induce few collisions". Specifically, the construction of paper [18] is 1-Lipschitz and universal, albeit it only works in one dimension. The construction of [16] 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 [16] uses an appropriately discretize random rotation

before applying the non-uniform overlapping.

In connection to our work, we note that our proof in Section 3.5, which shows that randomized distortions followed by folding leads to sparse approximation guarantees, could be plausibly applied to the construction of [16] as well. However, the non-uniform folding employed in this construction increases its complexity, making it less appealing in applications.

# Chapter 2

# Nearly Optimal Linear Embeddings into Very Low Dimensions

A key property of the embeddings produced by the Johnson-Lindenstrauss transform is that they correspond to orthonormal projections of $\mathbb{R}^d$ on a $k$-dimensional subspace. We too will focus on such so-called *orthonormal embeddings*, as they are convenient for a number of reasons. Note that for orthonormal embeddings, the right-hand-side inequality of Eq. 1.1 trivially holds, as projection does not expand distances by the definition. It therefore suffices to focus on the left-hand-side inequality (i.e., the lower bound) in Eq. 1.1. Also, we can assume without loss of generality assume that all vectors in $V$ have unit norm.

For a fixed set set $V \in \mathbb{R}^d$, we define $\epsilon_{opt}(k)$ to be the smallest achievable distortion over all possible orthonormal embeddings of $V$ into $\mathbb{R}^k$. Our specific contribution is an exhaustive-search-based algorithm that finds a linear embedding with distortion at most $\epsilon_{opt}(k) + \delta$, for any $\delta > 0$. This algorithm is space-efficient and can be achieved by a single pass over the data $V$. However, the runtime of this algorithm is exponential in $k$ and $1/\delta$. Unfortunately, due to the running time being exponential in $k$, its applications may be limited to cases where an embedding into a very small number of dimensions is desired. However, the algorithm exhibits many characteristics of a polynomial time approximation scheme (PTAS), so it is of theoretical interest.

Formally, our goal shall be to construct a linear embedding $f$ into $\mathbb{R}^k$ having distortion at most $\epsilon_{opt}(k) + \delta$ for an arbitrary $\delta > 0$. Our embedding will not necessarily be orthonormal; it will instead be the the composition of a random J-L embedding and another linear embedding. We establish the following:

**Theorem 2.0.1.** *Given a set $V$ consisting of $n$ points in $\mathbb{R}^d$, a positive integer $k < d$, and a parameter $\delta > 0$, there exists an algorithm $\mathcal{A}$ that returns an embedding $f$ of $V$ into $\mathbb{R}^k$ having distortion at most $\epsilon_{opt}(k) + \delta$, in time $O(n^2)(k/\delta)^{O(k^2 \log(n)/\delta^2)}$.*

*Proof.* Our algorithm is similar to that used by Badoiu et al., who solve a variety of geometric optimization problems by first reducing the dimension of the input, and then performing a brute force search on the lower dimensional space [2]. Define $U$ to be a $k$-dimensional subspace of $\mathbb{R}^d$ such that an orthonormal projection into $U$ yields an embedding with the optimal distortion $\epsilon_{opt}(k)$. We let $\{u_1, \ldots, u_k\}$ be an orthonormal basis for $U$. The first step of our algorithm is to perform a regular J-L embedding $g : \mathbb{R}^d \to \mathbb{R}^q$ on the input. We need to ensure that $g$ does not distort the angles between vectors in $U$ and $V$ too much; specifically, it suffices to obtain the following for every unit basis vector $u_i$ and each unit vector $v \in V$:

$$\langle g(u_i), g(v) \rangle^2 = \langle u_i, v \rangle^2 \pm \frac{\delta}{2k}. \tag{2.1}$$

Here, the '$\pm$' symbol is used to denote worst case deviations. Such a mapping $g$ can be performed on $V$ with high probability of success, using a codomain having $q = \Theta(\log(n)k/\delta^2)$ dimensions. Note that the bound still holds for squared inner products, because $U$ and $V$ consist entirely of unit vectors. Note also that the high probability of success holds even though we don't know what $U$ is.

Next, we do a brute force search over the unit sphere of $\mathbb{R}^q$ to approximately guess the transformed basis $\{g(u_1), \ldots, g(u_k)\}$. This may seem formidable, but fortunately for our purposes, it suffices to consider only $k$-tuples of candidates in a $\frac{\delta}{4k}$-net $N$ over unit vectors in $\mathbb{R}^q$. A standard volume-packing argument states that it is possible to construct $N$ with cardinality at most $\left(\frac{4k}{\delta}\right)^{Cq}$ for some absolute constant $C$. We simply iterate over all possible $k$-tuples of vectors in $N$. Suppose $\mathcal{W} = (w_1, \ldots, w_k)$ are the vectors considered in a particular

iteration of the search, and define $M_{\mathcal{W}}$ to be the $k \times q$ matrix whose rows are the vectors $(w_1, \ldots, w_k)$. Among all such $k$-tuples $\mathcal{W} \in N^k$, we identify the $k$-tuple that minimizes the maximum of the *right-side distortion*

$$\text{RightDistortion}(\mathcal{W}) = \max_{v \in V} \|M_{\mathcal{W}} \cdot g(v)\|_2^2 - 1$$

and the *left-side distortion*

$$\text{LeftDistortion}(\mathcal{W}) = \max_{v \in V} \left(1 - \|M_{\mathcal{W}} \cdot g(v)\|_2^2\right).$$

We let $\mathcal{W}^* = (w_1^*, \ldots, w_k^*)$ be the minimizing set of vectors in $N^k$, and let $M^*$ be the corresponding matrix. Our algorithm shall output the final linear transformation $f(v) = M^* \cdot g(v)$, the composition of the linear transformation implied by $M^*$ with the J-L mapping $g$.

We now show that $f$ has distortion at most $\epsilon_{opt}(k) + \delta$. For all $i$, define $w_i'$ to be the element of $N$ that is closest in direction to $g(u_i)$. Vector $w_i'$ is then a unit vector whose angle from $g(u_i)$ is at most $\frac{\delta}{4k}$, since $N$ is a $\frac{\delta}{4k}$-net. It follows that

$$\langle w_i', g(v) \rangle = \langle g(u_i), g(v) \rangle \pm \frac{\delta}{4k},$$

and hence, for all $v \in V$,

$$\langle w_i', g(v) \rangle^2 = \langle g(u_i), g(v) \rangle^2 \pm \frac{\delta}{2k} = \langle u_i, v \rangle^2 \pm \frac{\delta}{k},$$

where the latter equality uses the bound in (2.1). Summing over all values of $i$, we see that

$$\|M^* \cdot g(v)\|_2^2 = \sum_{i=1}^{k} \langle w_i', g(v) \rangle^2 = \sum_{i=1}^{k} \langle u_i, v \rangle^2 \pm \delta.$$

By our choice of $U$ and the fact that orthonormal projections are contractive, the value of

$\sum_{i=1}^{k} \langle u_i, v \rangle^2$ must lie in the range $[1 - \epsilon_{opt}(k), 1]$, and hence:

$$1 - \epsilon_{opt}(k) - \delta \leq \|f(v)\|_2^2 \leq 1 + \delta.$$

From this, it follows that $f$ has distortion at most $\epsilon_{opt}(k) + \delta$.

The time complexity is dominated by the time required to compute the worst case stretch and shinkage for each $k$-tuple of vectors $(w_1, \ldots, w_k)$ in our $\delta/4k$-net $N$. Naïvely, there are $O(n)$ vectors in $V$, and $O((k/\delta)^{qkC})$ $k$-tuples for some constant $C \in O(1)$, giving a total running time of $O(n)(k/\delta)^{O(k^2 \log(n)/\delta^2)}$. The running time could be potentially reduced by pruning the brute-force search (for example, by only considering $k$-tuples of vectors in $N$ that are approximately mutually orthogonal), but we do not pursue that direction here. $\square$

# Chapter 3

# Compressive Sensing of Arrays via Local Embeddings

We recall the *geometric* approach to constructing measurement matrices for compressive sensing of images [11, 24, 25]—the image is partitioned into $\sqrt{m} \times \sqrt{m}$ squares, which are then superimposed onto a $\sqrt{m} \times \sqrt{m}$ sensor array. It is easy to see then when applied naively to a pathological example, recovery of the original image is impossible. However, by randomly distorting the image prior to partitioning and superimposing it, we can obtain algorithms for compressive sensing of images that have provable recovery guarantees for all sparse images.

In this thesis, we present two variants of the geometric construction, called *wrapping* and *folding*, that both support the $\ell_\infty/\ell_1$ 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 [11] and [12]. Note that the resulting mapping from the pixels onto the sensor array is discontinuous, as e.g., the neighbouring 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 3.3. The distortions are very "local" (in particularly, 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 $\ell_\infty/\ell_1$ 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 $\ell_\infty/\ell_1$ 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 Chapter 4, we show that, for natural definitions of 'local' and 'universal', such mappings do not exist.

## 3.1 Preliminaries and Notation

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). Generalization of our results to rectangular images is straightforward.

Notation-wise, we use $[n]$ to denote the set $\{0 \ldots n-1\}$, and use $a \mod b$ 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 $d(f(x), f(y)) \leq c \cdot d(x, y)$.

## 3.2 Sparse recovery and hashing

Our signal acquisition algorithms all employ hash functions $h : [\sqrt{n}]^2 \to [\sqrt{m}]^2$ 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 $h_1 \ldots h_T$ 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

$$x_j^* \quad = \quad \text{median}_{t=1\ldots T}(A_{h_t}x)_{h_t(j)} \tag{3.1}$$

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 3.2.1.** *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 [5] (cf. [10]):

**Fact 3.2.2.** *Assume that $H$ is a $C$-universal distribution of hash functions $h : [n] \to [m]$. 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 3.1 satisfies*

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

*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 [5] (cf. [10]), 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 $j$

$$\Pr[h(j) \in h(S - \{j\})] \leq 1/10 \tag{3.2}$$

and

$$E\left[\sum_{j' \notin S - \{j\} : h(j') = h(j)} |x_j|\right] \leq \frac{\|x - x^{(k)}\|_1}{10k} \tag{3.3}$$

Applying Markov's inequality to Equation 3.3 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 $1/10 + 1/10 < 1/2$. □

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.

## 3.3 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 3.2.2. 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 3.2.2).

The randomized distortion map we use is defined as follows:

**Definition 3.3.1.** *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 $\lambda_x$, $\lambda_y$, and $\lambda_{xy}$ 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 $\lambda$ 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.

We proceed by establishing that the DISTORT step does not increase the distances between points too much:

**Lemma 3.3.2.** *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 $\lambda_x$, $\lambda_y$, and $\lambda_{xy}$. 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 $\lambda_y$ and $\lambda_{xy}$. Consequently,

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

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

We also show that the DISTORT step can never map two distinct points to the same target:

**Lemma 3.3.3.** *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 $\lambda_x$, $\lambda_y$, and $\lambda_{xy}$. 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

$$\left\lfloor \frac{\lambda_x x}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+y)}{\sqrt{n}} \right\rfloor = a + \left\lfloor \frac{\lambda_x(x+a)}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+a+y+b)}{\sqrt{n}} \right\rfloor \tag{3.4}$$

and

$$\left\lfloor \frac{\lambda_y y}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+y)}{\sqrt{n}} \right\rfloor = b + \left\lfloor \frac{\lambda_y(y+b)}{\sqrt{n}} \right\rfloor + \left\lfloor \frac{\lambda_{xy}(x+a+y+b)}{\sqrt{n}} \right\rfloor. \tag{3.5}$$

We note that if $a+b = 0$, then equations (3.4) and (3.5) 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 (3.4) 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. □

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

**Definition 3.3.5.** *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* $a \mod b$ *whenever* $(a \mod 2b) < b$*, and* $b - 1 - (a \mod b)$

*otherwise. Here, $\rho_x$ and $\rho_y$ 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 3.3.6.** *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 3.3.7.** *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 3.3.8.** *Any Distort-and-Fold transformation is Lipschitz with constant at most 4.*

## 3.4 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 $\ell_\infty/\ell_1$ guarantee with constant probability, yielding the $O(k \log n)$ measurement bound.

**Theorem 3.4.1.** *Let $H$ be the distribution of all Distort-and-Wrap hash functions, chosen uniformly over all choices of constants $\lambda_x$, $\lambda_y$, and $\lambda_{xy}$ selected during the DISTORT step. Then $H$ is universal. In particular, $H$ is $C$-universal for some universality constant*[1] *$C \leq 91$.*

*Proof.* Let $h \in H$ be randomly chosen, and let $\lambda_x$, $\lambda_y$, and $\lambda_{xy}$ 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 3.3.2. However, since Lemma 3.3.3 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 either $f(P) = f(Q)$ 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

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

[1] In the proof of Theorem 3.4.1, 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 $\lambda_x$ varies. Let $S_x = \{0, \ldots, a\}$ if $a \geq 0$, and let $S_x = \{-a, \ldots, 0\}$ 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 $\lambda_x$, $\lambda_y$, and $\lambda_{xy}$ 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 $Z_{xy} \equiv -b - t \mod \sqrt{m}$ and $Z_x \equiv b + t - a \mod \sqrt{m}$. 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 $Z_x \equiv b + t - a \mod \sqrt{m}$, and our previous analysis of the distribution of $Z_x$ can then be used to deduce

that

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

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 $Z_x \equiv b + t - a \mod \sqrt{m}$. In this case, we instead obtain the bound

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

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

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

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

and

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

using the fact that $|a+b| \leq 2\sqrt{n}$. Therefore, we conclude that $\Pr[Z_{xy} \equiv -b-t \mod \sqrt{m}] \leq \frac{13}{\sqrt{m}}$, 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 $Z_x \equiv -a - t \mod \sqrt{m}$ and

$Z_y \equiv -b - t \mod \sqrt{m}$. By a similar argument to that used in Case 1, we have $\Pr[Z_x \equiv -a - t \mod \sqrt{m}] \leq \frac{7}{\sqrt{m}}$, and $\Pr[Z_y \equiv -b - t \mod \sqrt{m}] \leq \frac{13}{\sqrt{m}}$, from which it follows that $\Pr[h(P) = h(Q)] \leq \frac{91}{m}$. □

## 3.5 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 3.5.1.** *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*

$$\Pr[f_x(P) = f_x(Q)] \leq \frac{C}{\|P - Q\|_\infty}, \tag{3.6}$$

*and the same statement holds for $f_y$.*

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

$$\Pr[h(P) = h(Q)] \leq \frac{1}{20k}. \tag{3.7}$$

*Proof.* Both parts of this proof employ techniques similar to those used in the proof of

Theorem 3.4.1. 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 $Z_x \equiv -a - t \mod 2\sqrt{m}$ or $Z_x + t_x \equiv a + t - t_x \mod 2\sqrt{m}$. 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. □

Using Lemma 3.5.1, we can obtain the following sparse recovery guarantee for Distort-

and-Fold:

**Theorem 3.5.2.** *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 3.1 satisfies*

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

*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 3.2.2, it suffices to show that Equations 3.2 and 3.3 hold.

We will first show that Equation 3.2 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 3.5.3.** *$E[|S"_x|] \leq c'''\sqrt{k}$ for some absolute constant $c'''$.*

*Proof.* Let $S" = \{Q_1 \dots Q_r\}$, and assume that the points $Q_1, Q_2 \dots Q_r$ are sorted in the order of increasing distance from $P$. By Lemma 3.5.1, 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 $8\ell$ distinct points $Q$ with $\|P - Q\|_\infty = \ell$. $\square$

The lemma for $S"_y$ is similar. By Markov's inequality, it follows that

$$\Pr[|S"_x| + |S"_y| > 160c'''\sqrt{k}] \leq 1/80.$$

Moreover, 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. Consequently, we have:

$$Pr\left[h(P) \in h(S"_x \cup S"_y - \{P\}) : |S"_x| + |S"_y| \leq 160c'''\sqrt{k}\right] \leq 1/80$$

The previous two equations thus imply

$$\Pr[h(P) \in h(S"_x \cup S"_y - \{P\})] \leq 1/80 + 1/80 = 1/40.$$

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

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

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 3.5.1, 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 3.2 holds.

From this, Equation 3.3 by applying linearity of expectations, and the theorem follows. □

# Chapter 4

# Impossibility of Universality for Local Hash Functions

In this chapter, 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. These results naturally complement the results of Chapter 3, where we showed that sparse recovery guarantees could be achieved using hash functions that were either universal (*Distort-and-Wrap*), or Lipschitz (*Distort-and-Fold*).

In this chapter, we employ the same notation as used in Chapter 3 (see Section 3.1).

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 4.0.4.** *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 4.0.5.** *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 $\Pr[R_h(P)$ 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 4.0.6.** *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 3.5 satisfies both conditions for a small value of $C$.

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

**Theorem 4.0.7.** *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.

The rest of the proof relies on the following structural lemma:

**Lemma 4.0.8.** *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 $\alpha$, 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 $\ell_p$ be the horizontal line passing through $p$. Consider the process of moving rightward along $\ell_p$ 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 $\ell_p$, 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 $\ell_p$ exits the square $[0, \sqrt{m}]^2$. We can ignore the points $p$ for which this process is not well defined due to $\ell_p$ 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 $\mu_e$ 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 $\mu$ 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. □

Using Lemma 4.0.8, 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 4.0.8 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. □

# Bibliography


[1] N. Alon. Problems and results in extremal combinatorics. *Discrete Math.*, 273(1):31–53, 2003.

[2] M. Badoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core sets. In *Symposium on the Theory of Computing*, pages 250–257, 2002.

[3] E. Candès. Compressive sampling. In *Proc. Int. Cong. Math.*, volume 3, pages 1433–1452, Madrid, Spain, 2006.

[4] 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.

[5] G. Cormode and S. Muthukrishnan. Improved data stream summaries: The count-min sketch and its applications. *Latin*, 2004.

[6] D. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.

[7] D. L. Donoho. Compressed Sensing. *IEEE Trans. Info. Theory*, 52(4):1289–1306, 2006.

[8] 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.

[9] R. Fergus, A. Torralba, and W. T. Freeman. Random lens imaging. *MIT CSAIL-TR-2006-058*, 2006.

[10] A. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of IEEE*, 2010.

[11] R. Gupta, P. Indyk, E. Price, and Y. Rachlin. Compressive sensing with local geometric features. *SOCG*, 2011.

[12] L. Hamilton, D. Parker, C. Yu, and P. Indyk. Focal plane array folding for efficient information extraction and tracking. *AIPR*, 2012.

[13] P. Indyk. Sketching, streaming and sublinear-space algorithms. *Graduate course notes, available at http://stellar.mit.edu/S/course/6/fa07/6.895/*, 2007.

[14] P. Indyk and J. Matousek. Low distortion embeddings of finite metric spaces. *Handbook of Discrete and Comp. Geom.*, 273:177–196, 2004.

[15] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Symposium on the Theory of Computing*, pages 604–613, New York, NY, 1998.

[16] P. Indyk, R. Motwani, P. Raghavan, and S. Vempala. Locality-preserving hashing in multidimensional spaces. *STOC*, 1997.

[17] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Proc. Conf. Modern Anal. and Prob.*, New Haven, CT, Jun. 1982.

[18] N. Linial and O. Sasson. Non-expansive hashing. *STOC*, 1996.

[19] J. Matousek and A. Sidiropoulos. Inapproximability of metric embeddings into $\mathbb{R}^d$. *Trans. Amer. Math. Soc*, 362(12):6341–6365, 2010.

[20] B. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, 26(1):17–32, 1981.

[21] S. Muthukrishnan. Data streams: Algorithms and applications). *Foundations and Trends in Theoretical Computer Science*, 2005.

[22] J. Romberg. Compressive sampling by random convolution. *SIAM Journal on Imaging Science*, 2009.

[23] A. Sidiropoulos. *Computational Metric Embeddings*. PhD thesis, Massachusetts Instt. Tech., May 2008.

[24] V. Treeaporn, A. Ashok, and M. A. Neifeld. Increased field of view through optical multiplexing. *Optics Express*, 18(21), 2010.

[25] 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.