A SYMBOL-BASED ALGORITHM FOR DECODING BAR CODES*

Mark A. Iwen†, Fadil Santosa§, and Rachel Ward¶

Abstract. We investigate the problem of decoding a bar code from a signal measured with a hand-held laser-based scanner. Rather than formulating the inverse problem as one of binary image reconstruction, we instead incorporate the symbology of the bar code into the reconstruction algorithm directly, and search for a sparse representation of the UPC bar code with respect to this known dictionary. Our approach significantly reduces the degrees of freedom in the problem, allowing for accurate reconstruction that is robust to noise and unknown parameters in the scanning device. We propose a greedy reconstruction algorithm and provide robust reconstruction guarantees. Numerical examples illustrate the insensitivity of our symbology-based reconstruction to both imprecise model parameters and noise on the scanned measurements.

Key words. Bar code decoding, symbol-based method, deconvolution, parameter estimation, inverse problem.

AMS subject classifications. AAA-xxx-00

1. Introduction. This work concerns an approach for decoding bar code signals. While it is true that bar code scanning is essentially a solved problem in many domains, as evidenced by its prevalent use, there is still a need for more reliable decoding algorithms in situations where the signals are highly corrupted and the scanning takes place in less than ideal situations. It is under these conditions that traditional bar code scanning algorithms often fail.

The problem of bar code decoding may be viewed as the deconvolution of a binary one-dimensional image involving unknown parameters in the blurring kernel that must be estimated from the signal [6]. Esedoglu [6] was the first to provide a mathematical analysis of the bar code decoding problem in this context, and he established the first uniqueness result of its kind for this problem. He further showed that the blind deconvolution problem can be formulated as a well-posed variational problem. An approximation, based on the Modica-Mortola energy [11], is the basis for the computational approach. The approach has recently been given further analytical treatment in [7].

A recent work [2] addresses the case where the blurring is not very severe. Indeed the authors were able to treat the signal as if it has not been blurred. They showed rigorously the variational framework can recover the true bar code image even if this parameter is not known. A later paper [3] consider the case where blurring is large and its parameter value known. However, none of these papers deal rigorously with noise although their numerical

simulations included noise. For an analysis of the deblurring problem where the blur is large and noise is present, the reader is referred to [7].

The approach presented in this work departs from the above image-based approaches. We treat the unknown as a finite-dimensional code and develop a model that relates the code to the measured signal. We show that by exploiting the symbology – the language of the bar code – a bar code can be identified with a sparse representation in the symbology dictionary. We develop a recovery algorithm that fits the observed signal to a code from the symbology in a greedy fashion, iterating in one pass from left to right. We prove that the algorithm can tolerate a significant level of blur and noise. We also verify insensitivity of the reconstruction to imprecise parameter estimation of the blurring function.

We were unable to find any previous symbol-based methods for bar code decoding in the open literature. In a related approach [4], a genetic algorithm is utilized to represent populations of candidate barcodes together with likely blurring and illumination parameters from the observed image data. Successive generations of candidate solutions are then spawned from those best matching the input data until a stopping criterion is met. That work differs from the current article in that it uses a different decoding method and does not utilize the relationship between the structure of the barcode symbology and the blurring kernel.

We note that there is a symbol-based approach for super-resolving scanned images [1]. However, that work is statistical in nature whereas the method we present is deterministic. Both this work and the super-resolution work are similar in spirit to lossless data compression algorithms known as 'dictionary coding' (see, e.g., [12]) which involve matching strings of text to strings contained in an encoding dictionary.

The outline of the paper is as follows. We start by developing a model for the scanning process. In Section 3, we study the properties of the UPC (Universal Product Code) bar code and provide a mathematical representation for the code. Section 4 develops the relation between the code and the measured signal. An algorithm for decoding bar code signals is presented in Section 5. Section 6 is devoted to the analysis of the algorithm proposed. Results from numerical experiments are presented in Section 7, and a final section concludes the work with a discussion.

2. A scanning model and associated inverse problem. A bar code is scanned by shining a narrow laser beam across the black-and-white bars at constant speed. The amount of light reflected as the beam moves is recorded and can be viewed as a signal in time. Since the bar code consists of black and white segments, the reflected energy is large when the beam is on the white part, and small when the beam is on the black part. The reflected light energy at a given position is proportional to the integral of the product of the beam intensity, which can be modeled as a Gaussian[1], and the bar code image intensity (white is high intensity, black is low). The recorded data are samples of the resulting continuous time signal.

Let us write the Gaussian beam intensity as a function of time:

$$g(t) = \alpha \frac{1}{\sqrt{2\pi}\sigma} e^{-(t^2/2\sigma^2)}. \tag{2.1}$$

---

[1]This Gaussian model has also been utilized in many previous treatments of the bar code decoding problem. See, e.g., [8] and references therein.

Figure 2.1. *Samples of the binary bar code function $z(t)$ and the UPC bar code. Note that in UPC bar codes, each bar - black or white - is a multiple of the minimum bar width.*

There are two parameters: (i) the variance σ^2 and (ii) the constant multiplier α. We will overlook the issue of relating time to the actual position of the laser beam on the bar code, which is measured in distance. We can do this because only relative widths of the bars are important in their encoding.

Because the bar code – denoted by $z(t)$ – represents a black and white image, we will normalize it to be a binary function. Then the sampled data are

$$d_i = \int g(t_i - \tau)z(\tau)d\tau + h_i, \quad i \in [m], \tag{2.2}$$

where the $t_i \in [0, n]$ are equally spaced discretization points, and the h_i represent the noise associated with scanning. We have used the notation $[m] = \{1, 2, ..., m\}$. We need to consider the relative size of the laser beam spot to the width of the narrowest bar in the bar code. We set the minimum bar width to be 1 in the artificial time measure.

It remains to explain the roles of the parameters in the Gaussian beam intensity. The variance σ^2 models the distance from the scanner to the bar code, with larger variance signifying longer distance. The *width* of a Gaussian represents the length of the interval, centered around the Gaussian mean, over which the Gaussian is greater than half its maximum amplitude; it is given by $2\sqrt{2\ln 2}\sigma$. Informally, the Gaussian blur width should be of the same order of magnitude as the size as the minimum bar width in the bar code for possible reconstruction. The multiplier α lumps the conversion from light energy interacting with a binary

bar code image to the measurement. Since the distance to the bar code is unknown and the intensity-to-voltage conversion depends on ambient light and properties of the laser/detector, these parameters are assumed to be unknown.

To develop the model further, consider the characteristic function

$$\chi(t) = \begin{cases} 1 & \text{for } 0 \leq t \leq 1, \\ 0 & \text{else.} \end{cases}$$

Then the bar code function can be written as

$$z(t) = \sum_{j=1}^{n} c_j \chi(t - (j-1)), \tag{2.3}$$

where the coefficients c_j are either 0 or 1 (see, e.g., Figure 2.1). The sequence

$$c_1, c_2, \cdots, c_n,$$

represents the information stored in the bar code, with a '0' corresponding to a white bar of unit width and a '1' corresponding to a black bar of unit width. For UPC bar codes, the total number of unit widths, n, is fixed to be 95 for a 12-digit code (further explanations in the subsequent).

Remark 2.1. One can think of the sequence $\{c_1, c_2, \cdots, c_n\}$ as an instruction for printing a bar code. Every c_i is a command to lay out a white bar if $c_i = 0$, or a black bar if otherwise. Substituting the bar code representation (2.3) back in (2.2), the sampled data can be represented as follows:

$$d_i = \int g(t_i - t) \left[\sum_{j=1}^{n} c_j \chi(t - (j-1)) \right] dt + h_i$$

$$= \sum_{j=1}^{n} \left[\int_{(j-1)}^{j} g(t_i - t) dt \right] c_j + h_i.$$

In terms of the matrix $\mathcal{G} = \mathcal{G}(\sigma)$ with entries

$$\mathcal{G}_{kj} = \frac{1}{\sqrt{2\pi}\sigma} \int_{(j-1)}^{j} e^{-\frac{(t_k - t)^2}{2\sigma^2}} dt, \quad k \in [m], \; j \in [n], \tag{2.4}$$

the bar code determination problem reads

$$d = \alpha \mathcal{G}(\sigma) c + h. \tag{2.5}$$

The matrix entries $\mathcal{G}_{kj}$ are illustrated in Figure 2.2. In the sequel, we will assume this discrete version of the bar code problem. While it is tempting to solve (2.5) directly for c, σ and α, the best approach for doing so is not obvious. The main difficulty stems from the fact that c is a binary vector, while the Gaussian parameters are continuous variables.
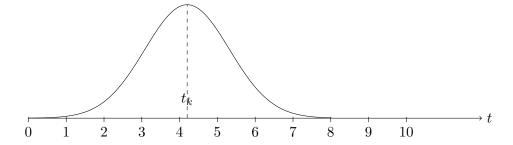
Figure 2.2. *The matrix element $\mathcal{G}_{kj}$ is calculated by placing a scaled Gaussian over the bar code grid and integrating over each of the bar code intervals.*

3. Incorporating the UPC bar code symbology. We now tailor the bar code reading problem to UPC bar codes, although we remark that our general framework should apply generally to any bar code of fixed length. In the UPC-A symbology, a bar code represents a 12-digit number. If we ignore the check-sum requirement, then any 12-digit number is permitted, and the number of unit widths, n, is fixed to 95. Going from left to right, the UPC bar code has 5 parts – the start sequence, the codes for the first 6 digits, the middle sequence, the codes for the next 6 digits, and the end sequence. Thus the bar code has the following structure:

$$SL_1L_2L_3L_4L_5L_6MR_1R_2R_3R_4R_5R_6E, \tag{3.1}$$

where S, M, and E are the start, middle, and end patterns respectively, and L_i and R_i are patterns corresponding to the digits.

In the sequel, we represent a white bar of unit width by 0 and a black bar by 1 in the bar code representation $\{c_i\}$.[2] The start, middle, and end patterns are fixed and given by

$$S = E = [101], \quad M = [01010].$$

The patterns for L_i and R_i are taken from the following table:

digit	L-pattern	R-pattern
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

$$\tag{3.2}$$

---

[2]Note that identifying white bars with 0 and black bars with 1 runs counter to the natural light intensity of the reflected laser beam. However, it is the black bars that carry information.

Note that the right patterns are just the left patterns with the 0's and 1's flipped. It follows that the bar code can be represented as a binary vector $c \in \{0,1\}^{95}$. However, not every binary vector constitutes a bar code – only 10^{12} of the possible 2^{95} binary sequences of length 95 – fewer than $10^{-16}\,\%$ – are bar codes. Specifically, the bar code structure (3.1) indicates that bar codes have specific sparse representations in the bar code dictionary constructed as follows: write the left-integer and right-integer codes as columns of a 7-by-10 matrix,

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix},$$

$$R = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The start and end patterns, S and E, are 3-dimensional vectors, while the middle pattern M is a 5-dimensional vector

$$S = E = [010]^T, \quad M = [01010]^T.$$

The bar code dictionary is the 95-by-123 block diagonal matrix

$$\mathcal{D} = \begin{bmatrix} S & 0 & \dots & & & & & & & & & \dots & 0 \\ 0 & L & & & & & & & & & & & \vdots \\ \vdots & & L & & & & & & & & & & \\ & & & L & & & & & & & & & \\ & & & & L & & & & & & & & \\ & & & & & L & & & & & & & \\ & & & & & & L & & & & & & \\ & & & & & & & M & & & & & \\ & & & & & & & & R & & & & \\ & & & & & & & & & R & & & \\ & & & & & & & & & & R & & \\ & & & & & & & & & & & R & \\ & & & & & & & & & & & R & \vdots \\ \vdots & & & & & & & & & & & R & 0 \\ 0 & \dots & & & & & & & & & \dots & 0 & E \end{bmatrix}.$$

The bar code (3.1), expanded in the bar code dictionary, has the form

$$c = \mathcal{D}x, \qquad x \in \{0,1\}^{123}, \tag{3.3}$$

where

1. The 1st, 62nd and the 123rd entries of x, corresponding to the S, M, and E patterns, are 1.
2. Among the 2nd through 11th entries of x, exactly one entry – the entry corresponding to the first digit in $c = \mathcal{D}x$ – is nonzero. The same is true for 12th through 22nd entries, etc, until the 61st entry. This pattern starts again from the 63rd entry through the 122th entry. In all, x has exactly 15 nonzero entries.

That is, x must take the form

$$x^T = [1, v_1^T, \cdots, v_6^T, 1, v_7^T, \cdots, v_{12}^T, 1], \tag{3.4}$$

where v_j, for $j = 1, \cdots, 12$, are vectors in $\{0,1\}^{10}$ having only one nonzero element. In this new representation, the bar code reconstruction problem (2.5) reads

$$d = \alpha \mathcal{G}(\sigma)\mathcal{D}x + h, \tag{3.5}$$

where $d \in \mathbb{R}^m$ is the measurement vector, the matrices $\mathcal{G}(\sigma) \in \mathbb{R}^{m \times 95}$ and $\mathcal{D} \in \{0,1\}^{95 \times 123}$ are as defined in (2.4) and (3.3) respectively, and $h \in \mathbb{R}^m$ is additive noise. Note that $\mathcal{D}$ has fewer rows than columns, while $\mathcal{G}$ will generally have more rows than columns; we will refer to the ratio of rows to columns as the oversampling ratio and denote it by $r = m/n$. Given the data $d \in \mathbb{R}^m$, our objective is to return a valid bar code $x \in \{0,1\}^{123}$ as reliably and quickly as possible.

4. Properties of the forward map. Incorporating the bar code dictionary into the inverse problem (3.5), we see that the map between the bar code and observed data is represented by the matrix $\mathcal{P} = \alpha \mathcal{G}(\sigma)\mathcal{D} \in \mathbb{R}^{m \times 123}$. We will refer to $\mathcal{P}$, which is a function of the model parameters α and σ, as the *forward map*.

4.1. Near block-diagonality. For reasonable levels of blur in the Gaussian kernel, the forward map $\mathcal{P}$ inherits an almost block-diagonal structure from the bar code matrix $\mathcal{D}$ as illustrated in Figure 4.1. In the limit as the amount of blur $\sigma \to 0$, the forward map $\mathcal{P}$ becomes exactly the block-diagonal bar code matrix. More precisely, we partition the forward map $\mathcal{P}$ according to the block-diagonal structure of the bar code dictionary $\mathcal{D}$,

$$\mathcal{P} = \begin{bmatrix} P^{(1)} \ P^{(2)} \ \cdots \ P^{(15)} \end{bmatrix}. \tag{4.1}$$

The 1st, 8th, and 15th sub-matrices are special as they correspond to the known start, middle, and end patterns of the bar code. In accordance with the structure of x where $c = \mathcal{D}x$, these sub-matrices are column vectors of length m,

$$P^{(1)} = p_1^{(1)}, \quad P^{(8)} = p_1^{(8)}, \quad \text{and} \quad P^{(15)} = p_1^{(15)}.$$

The remaining sub-matrices are blurred versions of the left-integer and right-integer codes L and R, represented as m-by-10 nonnegative real matrices. We write each of them as

$$P^{(j)} = \left[p_1^{(j)} \; p_2^{(j)} \; \cdots \; p_{10}^{(j)} \right], \quad j \neq 1, 8, 15, \tag{4.2}$$

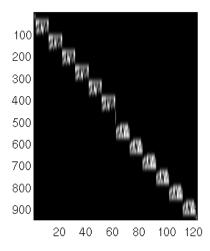where each $p_k^{(j)}$, $k = 1, 2, ..., 10$, is a column vector of length m.

Figure 4.1. *A representative bar code forward map $\mathcal{P} = \alpha \mathcal{G}(\sigma) \mathcal{D}$ corresponding to oversampling parameter $r = 10$, amplitude $\alpha = 1$, and Gaussian standard deviation $\sigma = 1.5$. The lone column vectors at the start, middle, and end account for the known start, middle, and end patterns in the bar code.*

Recall that the over-sampling rate $r = m/n$ indicates the number of time samples per minimal bar code width. Given r, we can partition the rows of $\mathcal{P}$ into 15 blocks, each block with index set I_j of size $|I_j|$, so that each sub-matrix is well-localized within a single block. We know that if $P^{(1)}$ and $P^{(15)}$ correspond to samples of the 3-bar sequence "101" or "black-white-black", so $|I_1| = |I_{15}| = 3r$. The sub-matrix $P^{(8)}$ corresponds to samples from the middle 5 bar-sequence so $|I_8| = 5r$. Each remaining sub-matrix corresponds to samples from a digit of length 7 bars, therefore $|I_j| = 7r$ for $j \neq 1, 8, 15$.

We can now give a quantitative measure describing how 'block-diagonal' the forward map is. To this end, let ε be the infimum of all $\epsilon > 0$ satisfying both
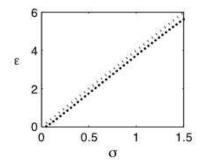
$$\left\| \left. p_k^{(j)} \right|_{[m] \setminus I_j} \right\|_1 < \epsilon, \quad \text{for all} \;\; j \in [15], \; k \in [10], \tag{4.3}$$

and

$$\left\| \left(\sum_{j'=j+1}^{15} p_{k_{j'}}^{(j')} \right) \right|_{I_j} \right\|_1 < \epsilon, \quad \text{for all} \;\; j \in [15], \text{ and all choices of } k_{j+1}, \ldots, k_{15} \in [10]. \tag{4.4}$$

The magnitude of ε indicates to what extent the energy of each column of $\mathcal{P}$ is localized within its proper block. If there were no blur, there would be no overlap between blocks and $\epsilon = 0$.

Simulation results such as those in Figure 4.2 suggest that for $\alpha = 1$, the value of ε in the forward map $\mathcal{P}$ can be expressed in terms of the oversampling ratio r and Gaussian standard deviation σ according to the formula $\varepsilon = (2/5)\sigma r$, at least over the relevant range of blur $0 \leq \sigma \leq 1.5$. By linearity of the forward map with respect to the amplitude α, this implies that more generally $\varepsilon = (2/5)\alpha\sigma r$.
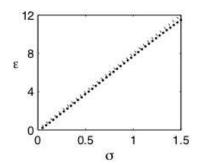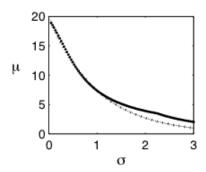

Figure 4.2. *For oversampling ratios $r = 10$ (left) and $r = 20$ (right) and $\alpha = 1$, the thick line represents the minimal value of ε satisfying (4.3) and (4.4) in terms of σ. The thin line in each plot represents the function $(2/5)\sigma r$.*

4.2. Column incoherence. We now highlight another property of the forward map $\mathcal{P}$ that allows for robust bar code reconstruction. The left-integer and right-integer codes for the UPC bar code, as enumerated in Table (3.2), are well-separated by design: the ℓ_1-distance between any two distinct codes is greater than or equal to 2. Consequently, if D_k are the columns of the bar code dictionary $\mathcal{D}$, then $\min_{k_1 \neq k_2} \|D_{k_1} - D_{k_2}\|_1 = 2$. This implies for the forward map $\mathcal{P} = \alpha\mathcal{G}(\sigma)\mathcal{D}$ that when there is no blur, i.e. $\sigma = 0$,

$$\mu := \min_{j,k_1 \neq k_2} \left\| p_{k_1}^{(j)} - p_{k_2}^{(j)} \right\|_1 = \min_{j,k_1 \neq k_2} \left\| p_{k_1}^{(j)} \Big|_{I_j} - p_{k_2}^{(j)} \Big|_{I_j} \right\|_1 = 2\alpha r, \qquad (4.5)$$

where r is the over-sampling ratio. As the blur increases from zero, the column separation factor $\mu = \mu(\sigma, \alpha, r)$ decreases smoothly. In Figure 4.3 we plot μ versus σ for different oversampling ratios, as obtained from numerical simulation. Simulations such as these suggest that μ closely follows the curve $\mu \approx 2\alpha r e^{-\sigma}$, at least in the range $\sigma \leq 1$.

5. A simple decoding procedure for UPC bar codes. We know from the bar code determination problem (3.5) that without additive noise, the observed data d is the sum of 15 columns from $\mathcal{P}$, one column from each block $P^{(j)}$. Based on this observation, we will employ a reconstruction algorithm which, once initialized, selects the column from the successive block to minimize the norm of the data remaining after the column is subtracted. This greedy algorithm is described in pseudo-code as follows.
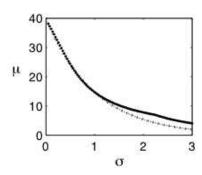
Figure 4.3. *For oversampling ratios $r = 10$ (left) and $r = 20$ (right), we plot the minimal column separation $\mu = \min_{k_1 \neq k_2} \left\| p_{k_1}^{(j)} - p_{k_2}^{(j)} \right\|_1$ for the forward map $\mathcal{P} = \mathcal{G}(\sigma)\mathcal{D}$, as a function of the standard deviation σ of the Gaussian kernel. The plots suggest that $\mu \approx 2\alpha r e^{-\sigma}$ for $\sigma \leq 1$.*

Algorithm 1: Recover UPC Bar Code

```
initialize:
     for  ℓ = 1, 62, 123,  xₗ = 1
     else xₗ = 0
     δ ← d
for  j = 2 : 7,  9 : 14
     kₘᵢₙ = arg minₖ ‖δ − pₖ⁽ʲ⁾‖₁
     if  j ≤ 7,  ℓ ← 1 + 10(j − 2) + kₘᵢₙ
     else ℓ ← 62 + 10(j − 9) + kₘᵢₙ
     xₗ ← 1
     r ← δ − p_{kₘᵢₙ}⁽ʲ⁾
end
```

---

6. Analysis of the algorithm. Algorithm 1 recovers the bar code one digit at a time by iteratively scanning through the observed data. The runtime complexity of the method is dominated by the 12 calculations of $k_{\min}$ performed by the algorithm's single loop over the course of its execution. Each one of these calculations of $k_{\min}$ consists of 10 computations of the ℓ_1-norm of a vector of length m. Thus, the runtime complexity of the algorithm is $O(m)$, and can be executed in less than a second for standard UPC bar code proportions.[3]

6.1. Recovery of the unknown bar code. Recall that the 12 unknown digits in the unknown bar code c are represented by the sparse vector x in $c = \mathcal{D}x$. We already know that $x_1 = x_{62} = x_{123} = 1$ as these elements corresponds to the mandatory start, middle, and end sequences. Assuming for the moment that the forward map $\mathcal{P}$ is *known*, i.e., that both σ and α are known, we now prove that the greedy algorithm will reconstruct the correct bar code

---

[3]In practice, when σ is not too large, a 'windowed' vector of length less than m can be used to approximate $\left\| \delta - p_k^{(j)} \right\|_1$ for each k, j. This can reduce the constant of proportionality associated with the runtime complexity.

from noisy data $d = \mathcal{P}x + h$ as long as $\mathcal{P}$ is sufficiently block-diagonal and if its columns are sufficiently incoherent. In the next section we will extend the analysis to the case where σ and α are unknown.

Theorem 1. *Suppose $I_1, \ldots, I_{15} \subset [m]$ and $\varepsilon \in \mathbb{R}$ satisfy the conditions (4.3)-(4.4). Then, Algorithm 1 will correctly recover a bar code signal x from noisy data $d = \mathcal{P}x + h$ provided that*

$$\left\| p_{k_1}^{(j)} \Big|_{I_j} - p_{k_2}^{(j)} \Big|_{I_j} \right\|_1 > 2 \left(\left\| h|_{I_j} \right\|_1 + 2\varepsilon \right) \tag{6.1}$$

for all $j \in [15]$ and $k_1, k_2 \in [10]$ with $k_1 \neq k_2$.

Proof:

Suppose that

$$d = \mathcal{P}x + h = \sum_{j=1}^{15} p_{k_j}^{(j)} + h.$$

Furthermore, denoting $k_j = k_{\min}$ in the for-loop in Algorithm 1, suppose that $k_2, \ldots, k_{j'-1}$ have already been correctly recovered. Then the residual data, δ, at this stage of the algorithm will be

$$\delta = p_{k_{j'}}^{(j')} + \delta_{j'} + h,$$

where $\delta_{j'}$ is defined to be

$$\delta_{j'} = \sum_{j=j'+1}^{15} p_{k_j}^{(j)}.$$

We will now show that the j'th execution of the for-loop will correctly recover $p_{k_{j'}}^{(j')}$, thereby establishing the desired result by induction.

Suppose that the j'th execution of the for-loop incorrectly recovers $k_{\mathrm{err}} \neq k_{j'}$. This happens if

$$\left\| \delta - p_{k_{\mathrm{err}}}^{(j')} \right\|_1 \leq \left\| \delta - p_{k_{j'}}^{(j')} \right\|_1.$$

In other words, we have that

$$\begin{aligned}
\left\| \delta - p_{k_{\mathrm{err}}}^{(j')} \right\|_1 &= \left\| \delta|_{I_{j'}} - p_{k_{\mathrm{err}}}^{(j')} \Big|_{I_{j'}} \right\|_1 + \left\| \delta|_{I_{j'}^c} - p_{k_{\mathrm{err}}}^{(j')} \Big|_{I_{j'}^c} \right\|_1 \\
&\geq \left\| p_{k_{j'}}^{(j')} \Big|_{I_{j'}} - p_{k_{\mathrm{err}}}^{(j')} \Big|_{I_{j'}} \right\|_1 - \left\| \delta_{j'}|_{I_{j'}} \right\|_1 - \left\| h|_{I_{j'}} \right\|_1 \\
&\quad + \left\| \delta_{j'}|_{I_{j'}^c} + h|_{I_{j'}^c} \right\|_1 - \left\| p_{k_j}^{(j')} \Big|_{I_{j'}^c} \right\|_1 - \left\| p_{k_{\mathrm{err}}}^{(j')} \Big|_{I_{j'}^c} \right\|_1 \\
&\geq \left\| p_{k_{j'}}^{(j')} \Big|_{I_{j'}} - p_{k_{\mathrm{err}}}^{(j')} \Big|_{I_{j'}} \right\|_1 + \left\| \delta_{j'}|_{I_{j'}^c} + h|_{I_{j'}^c} \right\|_1 - \left\| h|_{I_{j'}} \right\|_1 - 3\varepsilon
\end{aligned}$$

from conditions (4.3) and (4.4). To finish, we simply simultaneously add and subtract $\|\delta_{j'}|_{I_{j'}} + h|_{I_{j'}}\|_1$ from the last expression to arrive at a contradiction to the supposition that $k_{\mathrm{err}} \neq k_{j'}$:

$$
\begin{aligned}
\left\|\delta - p_{k_{\mathrm{err}}}^{(j')}\right\|_1 &\geq \left(\left\|\left.p_{k_{j'}}^{(j')}\right|_{I_{j'}} - \left.p_{k_{\mathrm{err}}}^{(j')}\right|_{I_{j'}}\right\|_1 - 2\left\|h|_{I_{j'}}\right\|_1 - 4\varepsilon\right) + \left\|\delta_{j'} + h\right\|_1 \\
&= \left(\left\|\left.p_{k_{j'}}^{(j')}\right|_{I_{j'}} - \left.p_{k_{\mathrm{err}}}^{(j')}\right|_{I_{j'}}\right\|_1 - 2\left\|h|_{I_{j'}}\right\|_1 - 4\varepsilon\right) + \left\|\delta - p_{k_{j'}}^{(j')}\right\|_1 \\
&> \left\|\delta - p_{k_{j'}}^{(j')}\right\|_1.
\end{aligned}
\tag{6.2}
$$

$\square$

Remark 6.1. Equation (4.3) implies that

$$
\min_{j, k_1 \neq k_2} \left\|\left.p_{k_1}^{(j)}\right|_{I_j} - \left.p_{k_2}^{(j)}\right|_{I_j}\right\|_1 \geq \min_{j, k_1 \neq k_2} \left\|p_{k_1}^{(j)} - p_{k_2}^{(j)}\right\|_1 - 2\varepsilon = \mu - 2\varepsilon.[4]
$$

Thus, the recovery condition (6.1) in Theorem 1 will hold whenever

$$
\mu - 2\varepsilon > 2\left(\left\|h|_{I_j}\right\|_1 + 2\varepsilon\right).
$$

Using the empirical relationships $\varepsilon = (2/5)\alpha r\sigma$ and $\mu = 2\alpha r e^{-\sigma}$, we obtain the following upper bound on the level of sufficient noise for successful recovery:

$$
\max_{j \in [12]} \left\|h|_{I_j}\right\|_1 < \alpha r(e^{-\sigma} - (6/5)\sigma).
\tag{6.3}
$$

In practice the Gaussian blur width $2\sqrt{2\ln(2)}\sigma$ does not exceed the minimum width of the bar code, which we have normalized to be 1. This translates to a maximal standard deviation of $\sigma \approx .425$, and a noise ceiling in (6.3) of

$$
\max_{j \in [12]} \left\|h|_{I_j}\right\|_1 \leq .144\alpha r.
\tag{6.4}
$$

This should be compared to the ℓ_1-norm of the bar code signal over a block; the average ℓ_1 norm between the left-integer and right-integer codes is 3.5α.

Remark 6.2. In practice it may be beneficial to apply Algorithm 1 several times, each time changing the order in which the digits are decoded. For example, if the distribution of the noise is known in advance, it would be beneficial to to initialize the algorithm in regions of the bar code with less noise.

6.2. Stability of the greedy algorithm with respect to parameter estimation.

Insensitivity to unknown α. In the previous section we assumed a known Gaussian convolution matrix $\alpha\mathcal{G}(\sigma)$. In fact, this is generally not the case. In practice both σ and α must be estimated since these parameters depend on the distance from the scanner to the bar code, the reflectivity of the scanned surface, the ambient light, etc. This means that in practice,

---

[4] See equation (4.5) for the definition of μ.

Algorithm 1 will be decoding bar codes using only an approximation to $\alpha\mathcal{G}(\sigma)$. Suppose that the true standard deviation generating a sampled sequence d is σ, but that Algorithm 1 uses a different value $\widehat{\sigma}$ for reconstruction. We can regard the error incurred by $\widehat{\sigma}$ as additional additive noise in our sensitivity analysis, setting $h' = h + \alpha\big(\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma})\big)\mathcal{D}x$ and rewriting the inverse problem as

$$
\begin{aligned}
d &= \alpha\mathcal{G}(\sigma)\mathcal{D}x + h \\
&= \alpha\mathcal{G}(\widehat{\sigma})\mathcal{D}x + \Big(h + \alpha\big(\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma})\big)\mathcal{D}x\Big) \\
&= \alpha\mathcal{G}(\widehat{\sigma})\mathcal{D}x + h'.
\end{aligned}
\tag{6.5}
$$

We now describe a procedure for estimating α. Note that the middle portion of the observed data of length $5r$, $d_{mid} = d|_{I_8}$, represents a blurry image of the known middle pattern $M = [01010]$. Let $\mathcal{P} = \mathcal{G}(\widehat{\sigma})\mathcal{D}$ be the forward map generated by the estimate $\widehat{\sigma}$ when $\alpha = 1$, and consider the sub-matrix

$$
p_{mid} = P^{(8)}\Big|_{I_8}
$$

which is also a vector of length $5r$. If $\widehat{\sigma} = \sigma$ or $\widehat{\sigma} \approx \sigma$,[5] we expect a good estimate for α to be the least squares solution

$$
\widehat{\alpha} = \arg\min_a \|ap_{mid} - d_{mid}\|_2 = p_{mid}^T d_{mid}/\| p_{mid}\|_2^2.
\tag{6.6}
$$

Dividing both sides of the equation (6.5) by $\widehat{\alpha}$, the inverse problem becomes

$$
\frac{d}{\widehat{\alpha}} = \frac{\alpha}{\widehat{\alpha}}\mathcal{G}(\widehat{\sigma})\mathcal{D}x + \frac{1}{\widehat{\alpha}}h'.
\tag{6.7}
$$

Suppose that $1 - \gamma \le \alpha/\widehat{\alpha} \le 1 + \gamma$ for some $0 < \gamma < 1$. Then fixing the data to be $\widehat{d} = d/\widehat{\alpha}$ and fixing forward map to be $\mathcal{P} = \mathcal{G}(\widehat{\sigma})\mathcal{D}$, the recovery conditions (4.3), (4.4), and (6.1) become respectively

1. $\left\| p_k^{(j)}\Big|_{[m]\setminus I_j} \right\|_1 < \frac{\varepsilon}{1+\gamma}$ for all $j \in [15]$ and $k \in [10]$.

2. $\left\| \left(\sum_{j'=j+1}^{15} p_{k_{j'}}^{(j')}\right)\Big|_{I_j} \right\|_1 < \frac{\varepsilon}{1+\gamma}$ for all $j \in [14]$ and valid $k_{j'} \in [10]$.

3. $\left\| p_{k_1}^{(j)}\Big|_{I_j} - p_{k_2}^{(j)}\Big|_{I_j} \right\|_1 > 2\Big(\frac{1}{\alpha}\left\| h|_{I_j} \right\|_1 + \left\| \big(\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma})\big)\mathcal{D}x|_{I_j} \right\|_1 + \frac{2\varepsilon}{1-\gamma}\Big)$

Consequently, if $\sigma \approx \widehat{\sigma}$ and $1 \lessgtr \alpha \approx \widehat{\alpha}$, the conditions for correct bar code reconstruction do not change much.

Insensitivity to unknown σ. We have seen that one way to estimate the scaling α is to guess a value for σ and perform a least-squares fit of the observed data. In doing so, we found that the sensitivity of the recovery process with respect to σ is proportional to the quantity

$$
\left\| \big(\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma})\big)\mathcal{D}x|_{I_j} \right\|_1
\tag{6.8}
$$

---

[5]Here we have assumed that the noise level is low. In noisier settings it should be possible to develop more effective methods for estimating α by incorporating the characteristics of the scanning noise.

in the third condition immediately above. Note that all the entries of the matrix $\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma})$ will be small whenever $\widehat{\sigma} \approx \sigma$. Thus, Algorithm 1 should be able to tolerate small parameter estimation errors as long as the "almost" block diagonal matrix formed using $\widehat{\sigma}$ exhibits a sizable difference between any two of its digit columns which might (approximately) appear in any position of a given UPC bar code.

To get a sense of the size of the term (6.8), let us further investigate the expressions involved. Recall that using the dictionary matrix $\mathcal{D}$, a bar code sequence of 0's and 1's is given by $c = \mathcal{D}x$. When put together with the bar code function representation (2.3), we see that

$$[\mathcal{G}(\sigma)\mathcal{D}x]_i = \int g_\sigma(t_i - t)z(t)dt,$$

where

$$g_\sigma(t) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\left(\frac{t^2}{2\sigma^2}\right)}.$$

Therefore, we have

$$[\mathcal{G}(\sigma)\mathcal{D}x]_i = \sum_{j=1}^{n} c_j \int_{j-1}^{j} g_\sigma(t_i - t)dt. \tag{6.9}$$

Now, using the definition for the cumulative distribution function for normal distributions

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}dt,$$

we see that

$$\int_{j-1}^{j} g_\sigma(t_i - t)dt = \Phi\left(\frac{t_i - j + 1}{\sigma}\right) - \Phi\left(\frac{t_i - j}{\sigma}\right).$$

and we can now rewrite (6.9) as

$$[\mathcal{G}(\sigma)\mathcal{D}x]_i = \sum_{j=1}^{n} c_j \left[\Phi\left(\frac{t_i - j + 1}{\sigma}\right) - \Phi\left(\frac{t_i - j}{\sigma}\right)\right].$$

We now isolate the term we wish to analyze:

$$[(\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma}))\mathcal{D}x]_i$$
$$= \sum_{j=1}^{n} c_j \left[\Phi\left(\frac{t_i - j + 1}{\sigma}\right) - \Phi\left(\frac{t_i - j + 1}{\widehat{\sigma}}\right) - \Phi\left(\frac{t_i - j}{\sigma}\right) + \Phi\left(\frac{t_i - j}{\widehat{\sigma}}\right)\right].$$

We are interested in the error

$$|[(\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma}))\mathcal{D}x]_i|$$
$$\leq \sum_{j=1}^{n} c_j \left|\Phi\left(\frac{t_i - j + 1}{\sigma}\right) - \Phi\left(\frac{t_i - j + 1}{\widehat{\sigma}}\right) - \Phi\left(\frac{t_i - j}{\sigma}\right) + \Phi\left(\frac{t_i - j}{\widehat{\sigma}}\right)\right|$$
$$\leq \sum_{j=1}^{n} \left|\Phi\left(\frac{t_i - j + 1}{\sigma}\right) - \Phi\left(\frac{t_i - j + 1}{\widehat{\sigma}}\right)\right| + \left|\Phi\left(\frac{t_i - j}{\sigma}\right) - \Phi\left(\frac{t_i - j}{\widehat{\sigma}}\right)\right|,$$
$$\leq 2\sum_{j=0}^{n} \left|\Phi\left(\frac{t_i - j}{\sigma}\right) - \Phi\left(\frac{t_i - j}{\widehat{\sigma}}\right)\right|.$$

Suppose that $\xi = (\xi_k)$ is the vector of values $|t_i - j|$ for fixed i, running j, sorted in order of increasing magnitude. Note that ξ_1 and ξ_2 are less than or equal to 1, and $\xi_3 \leq \xi_1 + 1$, $\xi_4 \leq \xi_2 + 1$, and so on. We can center the previous bound around ξ_1 and ξ_2, giving

$$|[(\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma}))\,\mathcal{D}x]_i| \leq \sum_{j=0}^{n} \left| \Phi\left(\frac{\xi_1 + j}{\sigma}\right) - \Phi\left(\frac{\xi_1 + j}{\widehat{\sigma}}\right) \right| + \left| \Phi\left(\frac{\xi_2 + j}{\sigma}\right) - \Phi\left(\frac{\xi_2 + j}{\widehat{\sigma}}\right) \right| \quad (6.10)$$

Next we simply majorize the expression

$$f(x) = \Phi\left(\frac{x}{\sigma}\right) - \Phi\left(\frac{x}{\widehat{\sigma}}\right).$$

To do so, we take the derivative and find the critical points, which turn out to be

$$x_* = \pm\sqrt{2}\sigma\widehat{\sigma}\sqrt{\frac{\log\sigma - \log\widehat{\sigma}}{\sigma^2 - \widehat{\sigma}^2}}.$$

Therefore, each term in the summand (6.10) can be bounded by

$$\left| \Phi\left(\frac{\xi + j}{\sigma}\right) - \Phi\left(\frac{\xi + j}{\widehat{\sigma}}\right) \right| \leq \left| \Phi\left(\sqrt{2}\widehat{\sigma}\sqrt{\frac{\log\sigma - \log\widehat{\sigma}}{\sigma^2 - \widehat{\sigma}^2}}\right) - \Phi\left(\sqrt{2}\sigma\sqrt{\frac{\log\sigma - \log\widehat{\sigma}}{\sigma^2 - \widehat{\sigma}^2}}\right) \right|$$
$$:= \triangle_1(\sigma, \widehat{\sigma}). \qquad (6.11)$$

On the other hand, the terms in the sum decrease exponentially as j increases. To see this, recall the simple bound

$$1 - \Phi(x) \;=\; \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-t^2/2}dt \;\leq\; \frac{1}{\sqrt{2\pi}}\int_x^\infty \frac{t}{x}e^{-t^2/2}dt \;=\; \frac{e^{-x^2/2}}{x\sqrt{2\pi}}.$$

Writing $\sigma_{max} = \max\{(\sigma, \widehat{\sigma})\}$, and noting that $\Phi(x)$ is a positive, increasing function, we have for $\xi \in [0, 1)$

$$\left| \Phi\left(\frac{\xi + j}{\sigma}\right) - \Phi\left(\frac{\xi + j}{\widehat{\sigma}}\right) \right| \leq 1 - \Phi\left(\frac{\xi + j}{\sigma_{max}}\right)$$
$$\leq \frac{\sigma_{max}}{(\xi + j)\sqrt{2\pi}}e^{-(\xi+j)^2/(2\sigma_{max}^2)}$$
$$\leq \frac{\sigma_{max}}{(\xi + j)\sqrt{2\pi}}e^{-(\xi+j)/(2\sigma_{max}^2)} \quad \text{if } j \geq 1$$
$$= \frac{\sigma_{max}}{(\xi + j)\sqrt{2\pi}}\left(e^{-(2\sigma_{max}^2)^{-1}}\right)^{\xi+j}$$
$$\leq \frac{\sigma_{max}}{j\sqrt{2\pi}}\left(e^{-(2\sigma_{max}^2)^{-1}}\right)^{j}$$
$$:= \Delta_2(\sigma_{max}, j). \qquad (6.12)$$

Combining the bounds (6.11) and (6.12),

$$\left| \Phi\left(\frac{\xi + j}{\sigma}\right) - \Phi\left(\frac{\xi + j}{\widehat{\sigma}}\right) \right| \leq \min\left((\triangle_1(\sigma, \widehat{\sigma}), \triangle_2(\sigma_{max}, j)\right).$$

Suppose that j_1 is the smallest integer in absolute value such that $\triangle_2(\sigma_{max}, j_1) \leq \triangle_1(\sigma, \widehat{\sigma})$. Then from this term on, the summands in (6.10) can be bounded by a geometric series:

$$\sum_{j \geq j_1}^{n} \left| \Phi\left(\frac{\xi + j}{\sigma}\right) - \Phi\left(\frac{\xi + j}{\widehat{\sigma}}\right) \right| \leq \frac{2\sigma_{max}}{j_1 \sqrt{2\pi}} \sum_{j \geq j_1} a^j, \quad a = e^{-(2\sigma_{max}^2)^{-1}}$$

$$\leq \frac{2\sigma_{max}}{j_1 \sqrt{2\pi}} \cdot a^{j_1} (1-a)^{-1}.$$

We then arrive at the bound

$$|\left[(\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma})) \mathcal{D}x\right]_i| \leq 2 \cdot j_1 \triangle_1(\sigma, \widehat{\sigma}) + \frac{4\sigma_{max} \cdot a^{j_1}(1-a)^{-1}}{j_1 \sqrt{2\pi}}$$

$$=: B(\sigma, \widehat{\sigma}). \tag{6.13}$$

The term (6.8) can then be bounded according to

$$\left\| (\mathcal{G}(\sigma) - \mathcal{G}(\widehat{\sigma})) \mathcal{D}x|_{I_j} \right\|_1 \leq |I_j| B(\sigma, \widehat{\sigma}) \leq 7r B(\sigma, \widehat{\sigma}), \tag{6.14}$$

where $r = m/n$ is the over-sampling rate.

Recall that in practice the width $2\sqrt{2\ln(2)}\sigma$ of the Gaussian kernel is on the order of the minimum bar width in the bar code, which we normalized to 1. When the blur exactly equals the minimum bar width, we arrive at $\sigma \approx .425$. Below, we compute the error bound $B(\sigma, \widehat{\sigma})$ for $\sigma = .425$ and several values of $\widehat{\sigma}$.
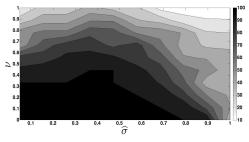
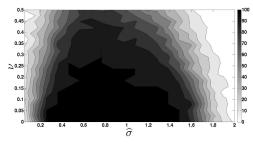| $\widehat{\sigma}$	0.2	0.4	.5	0.6	.8
$B(.425, \widehat{\sigma})$	.3453	.0651	.1608	.3071	.589

While the bound (6.14) is very rough, note that the tabulated error bounds incurred by inaccurate σ are at least roughly the same order of magnitude as the empirical noise level tolerance for the greedy algorithm, as discussed in Remark 6.1.

7. Numerical Evaluation. In this section we illustrate with numerical examples the robustness of the greedy algorithm to signal noise and imprecision in the α and σ parameter estimates. We assume that neither α nor σ is known a priori, but that we have an estimate $\widehat{\sigma}$ for σ. We then compute an estimate $\widehat{\alpha}$ from $\widehat{\sigma}$ by solving the least-squares problem (6.6).

The phase diagrams in Figure 7.1 demonstrate the insensitivity of the greedy algorithm to relatively large amounts of noise. These diagrams were constructed by executing Algorithm 1 on many trial input signals of the form $d = \alpha \mathcal{G}(\sigma)\mathcal{D}x + h$, where h is mean zero Gaussian noise. More specifically, each trial signal, d, was formed as follows: a 12 digit number was first generated uniformly at random, and its associated blurred bar code, $\alpha \mathcal{G}(\sigma)\mathcal{D}x$, was formed using the oversampling ratio $r = m/n = 10$. Second, a noise vector n with independent and identically distributed entries $n_j \sim \mathcal{N}(0, 1)$ was generated and then rescaled to form the additive noise vector $h = \nu \|\alpha \mathcal{G}(\sigma)\mathcal{D}x\|_2 \, (n/\|n\|_2)$. Hence, the parameter $\nu = \frac{\|h\|_2}{\|\alpha \mathcal{G}(\sigma)\mathcal{D}x\|_2}$ represents the noise-to-signal ratio of each trial input signal d.

We note that in laser-based scanners, there are two major sources of noise: electronic noise [9], which is often modeled as $1/f$ noise [5], and speckle noise [10], caused by the roughness

(a) True parameter values: $\sigma = .45$, $\alpha = 1$.

(b) True parameter values: $\sigma = .75$, $\alpha = 1$

Figure 7.1. *Recovery Probabilities when $\alpha = 1$ for two true σ settings. The shade in each phase diagram corresponds to the probability that the greedy algorithm will correctly recover a randomly selected bar code, as a function of the relative noise-to-signal level, $\nu = \frac{\|h\|_2}{\|\alpha\mathcal{G}(\sigma)\mathcal{D}x\|_2}$, and the σ estimate, $\widehat{\sigma}$. Black represents correct bar code recovery with probability 1, while pure white represents recovery with probability 0. Each data point's shade (i.e., probability estimate) is based on 100 random trials.*

of the paper. However, the Gaussian noise used in our numerical experiments is sufficient for the purpose of this work.
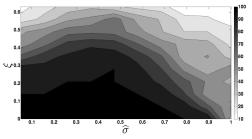
To create the phase diagrams in Figure 7.1, the greedy recovery algorithm was run on 100 independently generated trial input signals for each of at least 100 equally spaced $(\widehat{\sigma}, \nu)$ grid points (a 10×10 mesh was used for Figure 7.1(a), and a 20×20 mesh for Figure 7.1(b)). The number of times the greedy algorithm successfully recovered the original UPC bar code determined the color of each region in the $(\widehat{\sigma}, \nu)$-plane. The black regions in the phase diagrams indicate regions of parameter values where all 100 of the 100 randomly generated bar codes were correctly reconstructed. The pure white parameter regions indicate where the greedy recovery algorithm failed to correctly reconstruct any of the 100 randomly generated bar codes.
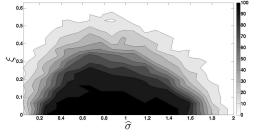
Looking at Figure 7.1 we can see that the greedy algorithm appears to be highly robust to additive noise. For example, when the σ estimate is accurate (i.e., when $\widehat{\sigma} \approx \sigma$) we can see that the algorithm can tolerate additive noise with Euclidean norm as high as $0.25\|\alpha\mathcal{G}(\sigma)\mathcal{D}x\|_2$. Furthermore, as $\widehat{\sigma}$ becomes less accurate the greedy algorithm's accuracy appears to degrade smoothly.

The phase diagrams in Figures 7.2 and 7.3 more clearly illustrate how the reconstruction capabilities of the greedy algorithm depend on σ, α, the estimate of σ, and on the noise level. We again consider Gaussian additive noise on the signal, i.e. we consider the inverse problem $d = \alpha\mathcal{G}(\sigma)\mathcal{D}x + h$, with independent and identically distributed $h_j \sim \mathcal{N}(0, \xi^2)$, for several noise standard deviation levels $\xi \in [0, .63]$. Note that $\mathbb{E}\left(\left\|h|_{I_j}\right\|_1\right) = 7r\xi\sqrt{2/\pi}$.[6] Thus, the numerical results are consistent with the bounds in Remark 6.1. Each phase diagram corresponds to different underlying parameter values (σ, α), but in all diagrams we fix the oversampling ratio at $r = m/n = 10$. As before, the black regions in the phase diagrams indicate parameter values $(\widehat{\sigma}, \xi)$ for which 100 out of 100 randomly generated bar codes were reconstructed, and white regions indicate parameter values for which 0 out of 100 randomly generated bar codes were reconstructed.

Comparing Figures 7.2(a) and 7.3(a) with Figures 7.2(b) and 7.3(b), respectively, we can
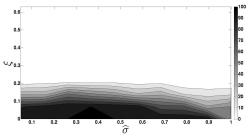
---

[6]This follows from the fact that the first raw absolute moment of each h_j, $\mathbb{E}(|h_j|)$, is $\xi\sqrt{2/\pi}$.
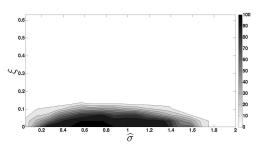
(a) True parameter values: $\sigma = .45$, $\alpha = 1$.

(b) True parameter values: $\sigma = .75$, $\alpha = 1$

Figure 7.2. *Recovery probabilities when $\alpha = 1$ for two true σ settings. The shade in each phase diagram corresponds to the probability that the greedy algorithm correctly recovers a randomly selected bar code, as a function of the additive noise standard deviation, ξ, and the σ estimate, $\widehat{\sigma}$. Black represents correct bar code recovery with probability 1, while pure white represents recovery with probability 0. Each data point's shade (i.e., probability estimate) is based on 100 random trials.*
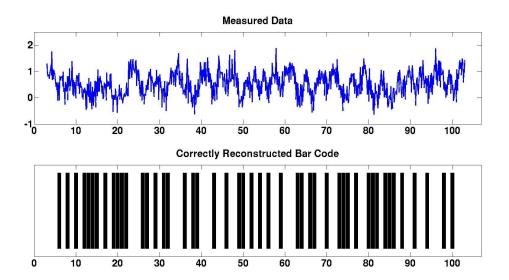

(a) True parameter values: $\sigma = .45$, $\alpha = .25$.

(b) True parameter values: $\sigma = .75$, $\alpha = .25$

Figure 7.3. *Recovery Probabilities when $\alpha = .25$ for two true σ settings. The shade in each phase diagram corresponds to the probability that the greedy algorithm will correctly recover a randomly selected bar code, as a function of the additive noise standard deviation, ξ, and the σ estimate, $\widehat{\sigma}$. Black represents correct bar code recovery with probability 1, while pure white represents recovery with probability 0. Each data point's shade (i.e., probability estimate) is based on 100 random trials.*
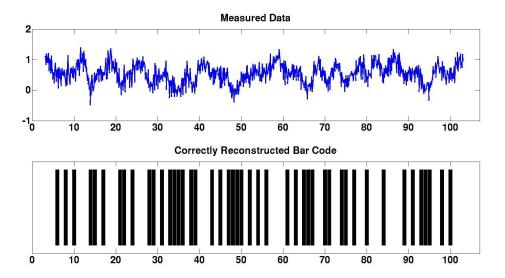
see that the greedy algorithm's performance appears to degrade with increasing σ. Note that this is consistent with our analysis of the algorithm in Section 6. Increasing σ makes the forward map $P = \alpha \mathcal{G}(\sigma)\mathcal{D}$ less block diagonal, thereby increasing the effective value of ε in conditions (4.3) and (4.4). Hence, condition (6.3) will be less likely satisfied as σ increases.

Comparing Figures 7.2 and 7.3 reveals the effect of α on the likelihood that the greedy algorithm correctly decodes a bar code. As α decreases from 1 to .25 we see a corresponding deterioration of the greedy algorithm's ability to handle additive noise of a given fixed standard deviation. This is entirely expected since α controls the magnitude of the blurred signal $\alpha \mathcal{G}(\sigma)\mathcal{D}x$. Hence, decreasing α effectively decreases the signal-to-noise ratio of the measured input data d.

Finally, all four of the phase diagrams in Figures 7.2 and 7.3 demonstrate how the greedy algorithm's probability of successfully recovering a randomly selected bar code varies as a function of the noise standard deviation, ξ, and σ estimation error, $|\widehat{\sigma} - \sigma|$. As both the noise level and σ estimation error increase, the performance of the greedy algorithm smoothly
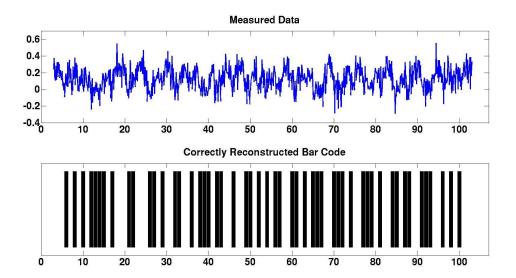
(a) True parameter values: $\sigma = .45$, $\alpha = 1$. Estimated $\widehat{\sigma} = .3$ and Noise Standard Deviation $\xi = .3$. Solving the least-squares problem (6.6) yields an α estimate of $\widehat{\alpha} = .9445$ from $\widehat{\sigma}$. The relative noise-to-signal level, $\nu = \frac{\|h\|_2}{\|\alpha \mathcal{G}(\sigma)\mathcal{D}x\|_2}$, is 0.4817.
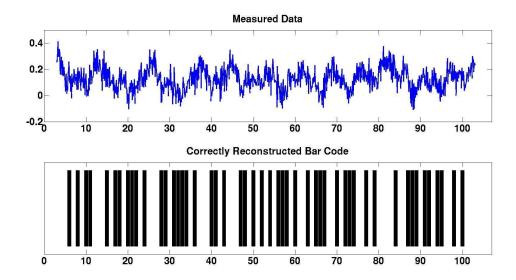

(b) True parameter values: $\sigma = .75$, $\alpha = 1$. Estimated $\widehat{\sigma} = 1$ and Noise Standard Deviation $\xi = .2$. Solving the least-squares problem (6.6) yields an α estimate of $\widehat{\alpha} = 1.1409$ from $\widehat{\sigma}$. The relative noise-to-signal level, $\nu = \frac{\|h\|_2}{\|\alpha \mathcal{G}(\sigma)\mathcal{D}x\|_2}$, is 0.3362.

Figure 7.4. *Two example recovery problems corresponding to dark regions in each of the phase diagrams of Figure 7.2. These recovery problems are examples of problems with $\alpha = 1$ for which the greedy algorithm correctly decodes a randomly selected UPC bar code approximately 80% of the time.*

(a) True parameter values: $\sigma = .45$, $\alpha = .25$. Estimated $\widehat{\sigma} = .5$ and Noise Standard Deviation $\xi = .1$. Solving the least-squares problem (6.6) yields an α estimate of $\widehat{\alpha} = 0.2050$ from $\widehat{\sigma}$. The relative noise-to-signal level, $\nu = \frac{\|h\|_2}{\|\alpha\mathcal{G}(\sigma)\mathcal{D}x\|_2}$, is 0.7001.

(b) True parameter values: $\sigma = .75$, $\alpha = .25$. Estimated $\widehat{\sigma} = .8$ and Noise Standard Deviation $\xi = .06$. Solving the least-squares problem (6.6) yields an α estimate of $\widehat{\alpha} = 0.3057$ from $\widehat{\sigma}$. The relative noise-to-signal level, $\nu = \frac{\|h\|_2}{\|\alpha\mathcal{G}(\sigma)\mathcal{D}x\|_2}$, is 0.4316.

Figure 7.5. *Two example recovery problems corresponding to dark regions in each of the phase diagrams of Figure 7.3. These recovery problems are examples of problems with $\alpha = .25$ for which the greedy algorithm correctly decodes a randomly-selected UPC bar code approximately 60% of the time.*

degrades. Most importantly, we can see that the greedy algorithm is relatively robust to inaccurate σ estimates at low noise levels. When $\xi \approx 0$ the greedy algorithm appears to suffer only a moderate decline in reconstruction rate even when $|\widehat{\sigma} - \sigma| \approx \sigma$.

Figure 7.4 gives examples of two bar codes which the greedy algorithm correctly recovers when $\alpha = 1$, one for each value of σ presented in Figure 7.2. In each of these examples the noise standard deviation, ξ, and estimated σ value, $\widehat{\sigma}$, were chosen so that they correspond to dark regions of the example's associated phase diagram in Figure 7.2. Hence, these two examples represent noisy recovery problems for which the greedy algorithm correctly decodes the underlying UPC bar code with relatively high probability.[7] Similarly, Figure 7.5 gives two examples of two bar codes which the greedy algorithm correctly recovered when $\alpha = 0.25$. Each of these examples has parameters that correspond to a dark region in one of the Figure 7.3 phase diagrams.

8. Discussion. In this work, we present a greedy algorithm for the recovery of bar codes from signals measured with a laser-based scanner. So far we have shown that the method is robust to both additive Gaussian noise and parameter estimation errors. There are several issues that we have not addressed that deserve further investigation.

First, we assumed that the start of the signal is well determined. By the start of the signal, we mean the time on the recorded signal that corresponds to when the laser first strikes a black bar. This assumption may be overly optimistic if there is a lot of noise in the signal. Preliminary numerical experiments suggest that the algorithm is not overly sensitive to uncertainties in the start time, and we are currently working on the development of a fast preprocessing algorithm for locating the start position from the samples.

Second, while our investigation shows that the algorithm is not sensitive to the parameter σ in the model, we did not address the best means for obtaining reasonable approximations to σ. In applications where the scanner distance from the bar code may vary (e.g., with handheld scanners) other techniques for determining $\widehat{\sigma}$ will be required. Given the robustness of the algorithm to parameter estimation errors it may be sufficient to simply fix $\widehat{\sigma}$ to be the expected optimal σ parameter value in such situations. In situations where more accuracy is required, the hardware might be called on to provide an estimate of the scanner distance from the bar code it is scanning, which could then be used to help produce a reasonable $\widehat{\sigma}$ value. In any case, we leave more careful consideration of methods for estimating σ to future work.

The final assumption we made was that the intensity distribution is well modeled by a Gaussian. This may not be sufficiently accurate for some distances between the scanner and the bar code. Since intensity profile as a function of distance can be measured, one can conceivably refine the Gaussian model to capture the true behavior of the intensities.

REFERENCES

[1] M. BERN AND D. GOLDBERG, *Scanner-model-based document image improvement*, Proceedings. 2000 International Conference on Image Processing, IEEE, 2000, 582–585.

---

[7]The ξ and $\widehat{\sigma}$ values were chosen to correspond to dark regions in a Figure 7.2 phase diagram, not necessarily to purely black regions.

[2] R. CHOKSI AND Y. VAN GENNIP, *Deblurring of one-dimensional bar codes via total variation energy minimisation*, SIAM J. Imaging Sciences, 3-4 (2010), pp. 735–764.

[3] R. CHOKSI, Y. VAN GENNIP, AND A. OBERMAN, *Anisotropic total variation regularized L^1-approximation and denoising/deblurring of 2D bar codes*, Inverse Problems and Imaging, 5 (2011), 591–617.

[4] L. DUMAS, M. EL RHABI AND G. ROCHEFORT, *An evolutionary approach for blind deconvolution of barcode images with nonuniform illumination*, IEEE Congress on Evolutionary Computation, 2011, 2423–2428.

[5] P. DUTTA AND P. M. HORN, *Low-frequency fluctuations in solids: $1/f$ noise*, Rev. Mod. Phys., 53-3 (1981), 497–516.

[6] S. ESEDOGLU, *Blind deconvolution of bar code signals*, Inverse Problems, 20 (2004), 121–135.

[7] S. ESEDOGLU AND F. SANTOSA, *Error estimates for a bar code reconstruction method*, to appear in Discrete and Continuous Dynamical Systems B.

[8] J. KIM AND H. LEE, *Joint nonuniform illumination estimation and deblurring for bar code signals*, Optic Express, 15-22 (2007), 14817–14837.

[9] S. KOGAN, *Electronic Noise and Fluctuations in Solids*, Cambridge University Press, 1996.

[10] E. MAROM, S. KRESIC-JURIC AND L. BERGSTEIN, *Analysis of speckle noise in bar-code scanning systems*, J. Opt. Soc. Am., 18 (2001), 888–901.

[11] L. MODICA AND S. MORTOLA, *Un esempio di Γ–convergenza*, Boll. Un. Mat. Ital., 5-14 (1977), 285–299.

[12] D. SALOMON, *Data Compression: The Complete Reference*, Springer-Verlag New York Inc, 2004.