Barcode Deblurring

Yifei Lou, Ernie Esser

July 3, 2012

A one-dimensional barcode is a finite series of alternating black and white stripes whose relative widths encode information. The most familiar is probably the UPC barcode used in supermarkets where laser scanners are used to decode barcode signals. Due to the distance or vibrations of the scanner when barcode is taken, the signal can be blurred and noisy. Therefore, it comes a general question: given a corrupted signal, how can we effectively reconstruct the associated barcode?

Standard commercial decoding techniques are based on classic edge detectors, such as those that find local extrema of the derivative of the signal, which hopefully corresponds to edges. However, this approach is highly unstable to small changes in the signal, for example, in the presence of noise. Furthermore, if the support size of the blurring kernel is relatively large, some edges in the original barcode may not have corresponding extrema of the corrupted signal.

Alternatively we consider to decode the barcode signal in an image deblurring framework. In particular, we assume that the ideal barcode signal $u$ is a one-dimensional $0-1$ step signal and the signal $f$ received by a scanner is a noisy, blurred version of $u$. We model the image formation process by

$$f = G_\sigma * u + n,$$

where $n$ is the noise, $G_\sigma$ is a Gaussian kernel with standard deviation $\sigma$ and $*$ denotes the convolution operator. The Gaussian type of kernels well approximates the actual blur. If $G_\sigma$ is known, recovering $u$ given $f$ is often called “non-blind deconvolution”; otherwise it is called “blind deconvolution”.

We will start the hands-on decoding experience by nonblind deconvolution methods, such as classical Tikhonov and Wiener filters. Then we will look at a blind deconvolution technique in [4] which only involves linear algebra. The main objective is to improve the performance of mobile imaging devices, like hand-held phones. We will test the accuracy of different types of methods and see how far we can push towards a better solution.

We may also try to explicitly take advantage of known features of the barcode to improve the deblurring. All UPC barcodes have the same patterns at the beginning, middle and end. Moreover, we know how each digit is encoded. In [2], these patterns are used to construct a block diagonal barcode dictionary, $D$, such that the barcode can be represented by $Dx$, where $x$ is a sparse binary
vector satisfying some constraints to ensure the barcode is valid. The blurry barcode signal, \( d \), is then modeled by

\[
d = \alpha G(\sigma) Dx + h,
\]

where \( h \) is noise and \( \alpha G(\sigma) \) is a blurring operator depending on parameters \( \alpha \) and \( \sigma \). In [2], they use a greedy algorithm to estimate \( x \), essentially recovering one digit of the barcode at a time. Thanks to the incoherence of the columns of \( D \), their greedy approach is somewhat insensitive to the noise and uncertainty in the blurring parameters.

An idea for improving upon this approach is to use partial reconstructions of the barcode to more accurately recover the blur parameters, which can then be used to improve the accuracy of the overall barcode reconstruction. The preexisting known features, such as the middle pattern, are probably too short to use to determine the blur parameters unless the width of the blur kernel is very small. However, knowing a longer section of the barcode, perhaps consisting of a few digits, may provide enough information to directly solve for the blurring parameters. This could be estimated using linear algebra or simple variational approaches. Alternatively, the known features could be used to construct linear filters that produce isolated delta functions when applied to barcode signals containing the known features. If these delta functions are sufficiently isolated compared to the width of the blur, we can apply these filters to the blurry data and attempt to read off an estimate of the blur kernel for each feature that we use. If the majority of the partial barcode reconstructions are accurate, then we should be able to get a good estimate of the blur parameters, which could be used to improve the performance of the method in [2] or in some altogether different method for non-blind deblurring.

If there is time, it may also be interesting to investigate some blind deblurring strategies for when we don’t have good a-priori assumptions about the blur and also don’t have sufficient knowledge of barcode features. For example, we could attempt to apply to barcodes the strategy in [3], which aims to recover a sharp signal and a blur kernel consistent with the data such that the ratio of the \( l_1 \) and \( l_2 \) norms of the high frequency component of the sharpened signal is small.
The tentative weekly agenda goes as follows,

**week 1** Introduction to barcode denoising/deblurring problems.

**week 2** Matlab generation of clean barcode signals and noisy/blurred barcode signals.

**week 3** Learn and experiment with some classical deblurring methods.

**week 4-6** Consider applications and extensions of some recent work: (see references).

**week 6-7** Computation and Matlab GUI.

**week 8** Summarize results, write up presentation.

**References**


