Non-Blind Deconvolution of Blurred 1D Barcodes
Summer Research Program: iCAMP
Barcode Deblurring
Box Constraint

Brian Kwan and Kristopher Jetviroj

August 15th 2013

Abstract

Barcode Deblurring is a tool that will become more and more important as smaller and cheaper cameras become the norm amongst the average person. Developing an efficient non-blind deconvolution of a barcode using the box constraint and extensions of it is the primary focus in our research. We experiment with slightly different methods to numerically estimate a barcode with gradient projection from an initial blurry image. The methods involve adding additional constraints based on the structure of a barcode, thresholding, or an adaptive constraint that attempts to make educated guesses about what the barcode should be at certain sections during the solution process. We see improvements in our solution for varying amounts of noise and blur when implementing these additional constraints based on our knowledge of a barcode.

1 Introduction

Barcodes are on nearly every product sold in stores. Having an improved, more accurate way than current methods to deblur barcodes from images will make it easier to use images rather than typical laser scanners to read barcodes.

1.1 Barcodes

The standard UPC-A 1D barcode which we examine is made up of a few important aspects. First, it is a binary signal of black and white modules that encode 12 digits related to the product. Each digit is represented by 7 modules of the signal, making a total of 84 modules. However, a real barcode has 95 module in total. The extra 11 come from 3 modules in the beginning, 5 modules in the middle, and 3 modules in the end that is part of the structure of every UPC-A 1D barcode.

1.2 Images versus Laser Scanners

Because of the nature of images, there is a higher likelihood of blur and noise. While most commercial applications currently effectively implement laser scanners, the prevalence of cameras in the hands of consumers makes it reasonable to examine an effective way of deblurring and reconstructing barcodes from images. Images also have the advantage of being able to contain multiple barcodes in a single image. This could allow batch barcode reading for some commercial applications such as inventory management or distribution versus laser scanning products one by one.

1.3 Method and Experimentation

The primary method we implement to deblur barcodes is gradient projection. We used MATLAB as our primary program. Because of this iterative process, we have the benefit of including several constraints, namely the Box Constraint, and other factors to optimize the estimation of barcodes given various amounts of noise and blur. We attempt to find effective parameters that would best estimate the original barcode.

2 Model and Method

We examine the non-blind deconvolution of barcodes, which means the blurring kernel, explained below, is known.

2.1 Model

First, it is important to have a model of what might happen to an image that is blurred and contains noise. The model used is as follows:

$$f = G_k * u + n$$

In the model, f is the resulting signal after adding noise n to G_k convoluted with u, where G_k is the blurring kernel (typically Gaussian), k is the standard deviation of the kernel, and u is the original barcode signal [3].

2.2 Objective

The next step after modelling the barcode is to find the best solution given a known G_k and an unknown u. Given the nature of a convolution, a typical equation can be rewritten from $f * g$ to $F \times g$ where F is a matrix that defines either the linear or circular convolution (depending on implementation/method) of f and g. Having the relationship between the blur kernel and the original barcode be a linear transformation means it is possible to use linear methods to try and find the best solution. In this case, the method used is minimization of this equation:

$$\min_x \|Ax - b\|^2 + \lambda \|x\|^2$$

where A is the matrix representation of convoluting G_k with u, x is u, b is the signal f pulled from an image, and λ is a given constant that will help minimize the effect of noise on the blurred barcode signal.

Solving this problem however will not be guaranteed if x can only be 0 or 1 as in a real binary signal. Changing the conditions to be convex by allowing x to be between 0 and 1 guarantees that there is a minimum and that numerical methods, given the right conditions, will yield an answer. This condition is also known as a Box Constraint.

2.3 Method

Since the minimization of our objective equation (quadratic) and the set we are minimizing over are both convex, we can now use the Gradient Projection method to not just approach a minimum, but a global minimum. Finding the gradient of our main equation yields us:

$$A^T(Ax - b) + \lambda x$$

The method itself follows the recursive relation:

$$x_{n+1} = x_n - dt \times (\nabla F)$$

where ∇F is the gradient of the equation we are minimizing and dt is the time step determining how fast we approach the minimum. Substituting yields:

$$x_{n+1} = x_n - dt \times (A^T(Ax - b) + \lambda x)$$

2.3.1 The Fast Fourier Transformation

Because multiplying by large matrices is computationally expensive, we speed up our method using the Fast Fourier Transformation (FFT). With FFT, a convolution $f * g$ in the fourier domain is equivalent to $F(f * g) = F(f) \bullet F(g)$ where F represents the transformation into the fourier domain. With this, $f * g = F^{-1}(F(f * g)) = F^{-1}(F(f) \bullet F(g))$ where F^{-1} represents the inverse transformation out of the fourier domain.

3 Various Constraints

3.1 Thresholding

A very basic constraint that we can apply during the solution process is thresholding, where we push values that get close to the boundary all the way to the boundary. Since we know that we're working with a binary signal, it would make sense that values that get very close to being 0 or 1 will, with a high amount of certainty, actually be that value. With this,values will tend even more to the extremes, hopefully revealing a better barcode-like structure

3.2 Adaptive

Another constraint we can apply is an adaptive constraint. The idea behind this comes from wanting a solution that will be as close to the boundaries as possible, but having a greater amount of control that simple thresholding. What this constraint does is it guesses, based on a pre-set distance from the boundary, each module of the barcode for all 95 modules. This guess is then saved and the program moves onto the next iteration.

This constraint should cause good guesses to stay around the boundary or not change very much, whereas bad guesses would move away from the boundary it was pushed near and eventually would be removed as a guess from the list of guesses.

3.3 Barcode Structure

As mentioned at the beginning of the paper, each barcode has certain features that are the same for every barcode. If the beginning and end of a barcode are properly detected, then it is easily possible to add the addition constraints every barcode has. *Note*: **W** stands for **W** and **B** stands for black. First, the beginning and end both have a 3 module wide combination of **BWB**. Second, the middle has a 5 module wide combination of **WBWBW**. Finally, there is also two patterns also associated with barcodes: one for the first 6 digits and one for the last 6 digits. The first 6 digits all start with a **W** module and end with a **B** module, while the last 6 digits have the opposite pattern where they start with **B** and end with **W**.

4 Parameter Experimentation

To see which parameters would best estimate a barcode, we fix the standard deviation of the noise at 0.1 and the blur at 1. The noise is uniformly distributed with a mean of 0. The blur with standard deviation of 1 means approximately the size of two modules is how wide the blurring kernel weights the values.

Now, varying the three parameters λ, dt, and $Precision$, where $Precision$ is the value determining when to stop iterating gradient projection, we can see what value gives the lowest error. The equation that defines $Precision$ is $\|x_{n+1} - x_n\|/\|x_n\|$.

At the end of the paper will be 8 figures with example values we experimented with. For the black plot, that shows us what our "solution" looks like after our gradient projection method and Box Constraint. For the blue/red plot, the more blue and the less red means a higher error. On the other hand, the more red and the less blue, the lower the error. Now, we will explain here what each figure means.

In figure 1, we set λ to 1. Because this is our regularization term to reduce the effect of noise, setting it too high in this case gives us a shorter, smooth-looking graph and a very high error.

In figure 2, we set λ to 0.1. Because this is still fairly high, the noise is mostly eliminated from the plot but still gives a decent amount of error.

In figure 3, we set λ to 0.0001. Because this value is so much lower, it only somewhat reduces the noise but gives a much better barcode-looking solution and a low amount of error.

In figure 4, we set dt to 3. Because this step size is too high, the solution blows up and gives us something that looks nothing like a barcode.

In figure 5, we set dt to 1. Because this is much more reasonable, we get a decent solution that we can reconstruct a barcode from. Figure 6 gives only a slightly different solution.

In figure 7 and figure 8, we set $Precision$ to 0.00001 and 0.0000001. Because both of these are quite small, there is little difference between them. Both values, though, give good approximations of our final barcode. From further experiments, this value should be set depending on the standard deviation of the blur and the step size.

From the analysis of the tested values, we decided that setting λ to 0.001, dt to 1, and $Precision$ to 0.00001 were optimal parameters in most situations.

5 Error Analysis

With optimal parameters selected, we moved on to error analysis to test the different constraints examined earlier. In the colored plots at the end of the paper, we plotted the error between the real barcode and the "solved" barcode where the axes are the standard deviation of the noise versus the standard deviation of the blur. The error is calculated by: $\|RealBarcode - SolvedBarcode\|^2$.

In figure 9 and 10, we see only minor differences in error. There was not much improvement using thresholding over the gradient projection method itself.

In figure 11, there was a significant reduction in error, especially for low amounts of blur. This indicates that it might be possible, with more complex constraints, to counter higher amounts of noise and blur.

In figure 12, we see the best overall reduction in error compared to the other three plots. It nearly halved the maximum error compared to the other methods. This is a powerful constraint because it reduces from 95 unknowns to 64, an over 30% decrease.

6 Acknowledgements

We would like to thank the University of California - Irvine, its iCAMP program, and Dr. Sarah Eichhorn for introducing us to our first research experience. This research paper was created with the help and guidance of our advisor Ernie Esser and our mentor Yifei Lou.

6.1 Related Work

The 2012 iCAMP Barcode group's paper was helpful in getting up to speed on the gradient projection method we implemented [1]. Our advisor and mentor also co-authored an interesting paper using polynomial approximation and other techniques [2].

7 Conclusion

Using gradient projection is a strong first step at deblurring a barcode using our model, but it is possible to strengthen the conditions and constraints to better adhere to the binary behavior of a barcode for a better reconstruction of a blurry and noisy barcode. We are optimistic of the potential of our constraints and seek to lay out the foundation of future work to be done below.

7.1 Future Work

A finer grained analysis of parameters and errors would be ideal to improve the method that much more.

A strong contender for future examination is our adaptive constraint. Unfortunately, guessing wrong appears to hurt our final solution and we would like to deal with wrong guesses more effectively.

Next, we would look to combine the constraints into one. Namely, combining the barcode structure constraint with the adaptive constraint should vastly improve the estimation of the barcode.

We also want to look at a pre-pass filter to reduce the amount of noise while preserving our signal as much as possible. This will increase the signal-to-noise ratio of our blurry data and hopefully give us a better estimate of the barcode. Specifically, we were looking at the Savitzky-Golay smoothing filter (conveniently built-in to MATLAB).

Another idea we were thinking about is a final-pass dictionary minimization. This minimization would involve comparing our solved barcode with the real digits that are encoded in a barcode. The digit that gave the least error would be the one selected for each section.

Finally, an interesting idea is a multi-pass solution that changes the values of the box constraint itself. The idea is that instead of allowing x to be from 0 to 1, change it to be from 0 to 0.2, 0.8 to 1, 0 to 0.3, 0.7 to 1, etc. with the idea that it might be possible to iteratively solve for the barcode.

The first pass with the new constraint from 0 to 0.2 would be able to give all the places in the barcode that are definitely 0 and caused by large sections with 0 values. A second pass with a different constraint from 0.8 to 1 would then give all the places are the definitely 1 and caused by large section with 1 values. Then the third pass would have a different constraint from 0 to 0.3 and would give more sections of the barcode that are 0 with a high probability. Each pass would provide information to the next and would help better define a real barcode.

This would continue and would hopefully give all the sections of the barcode that are either definitely or highly likely to be 0 or 1. Once all the passes are done, the final pass would run with the normal box constraint along with any other constraints and attempt to solve the intermediate values that were difficult to discern.

References

[1] Christine Lew and Dheyani Malde. Non-blind barcode deconvolution by gradient projection. iCAMP, 2012.

[2] Yifei Lou, Ernie Esser, Hongkai Zhao, and Jack Xin. Partially blind deblurring of barcode from out-of-focus blur.

[3] Jack Xin and Ernie Esser. Filtering and convolutions. `https://eee.uci.edu/13w/44960/home/filtering.pdf`.

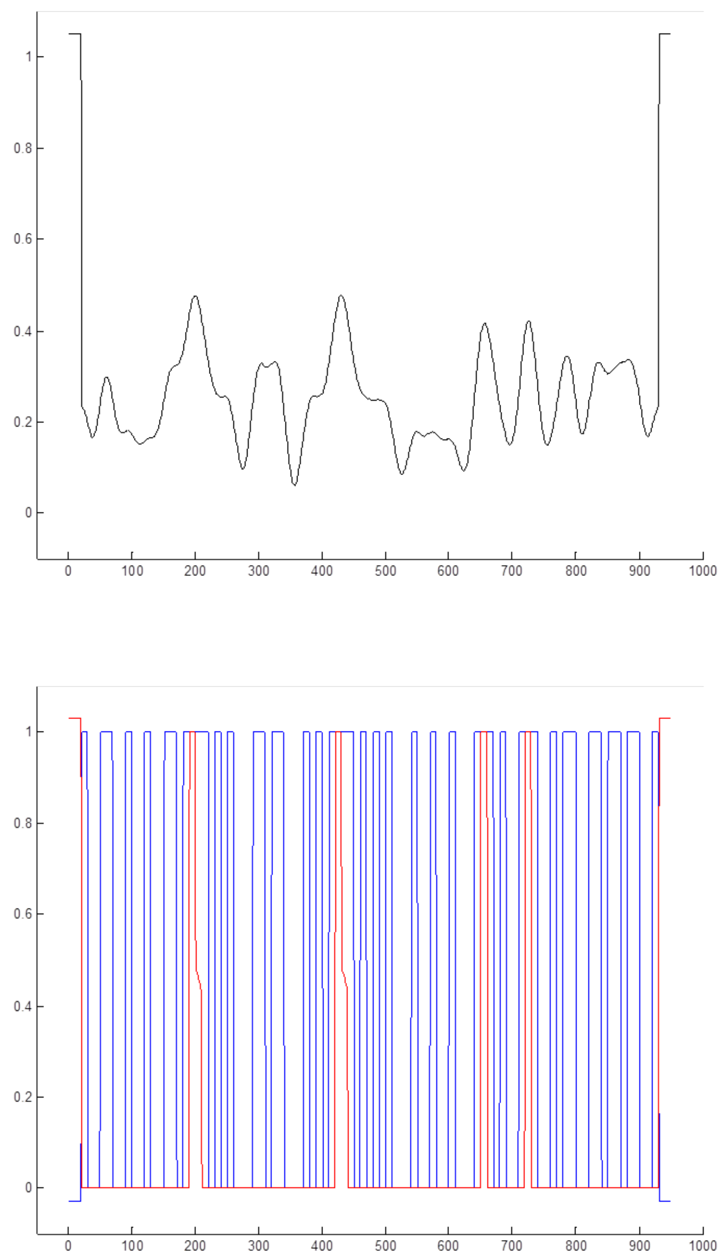


Figure 1: $\lambda = 1$

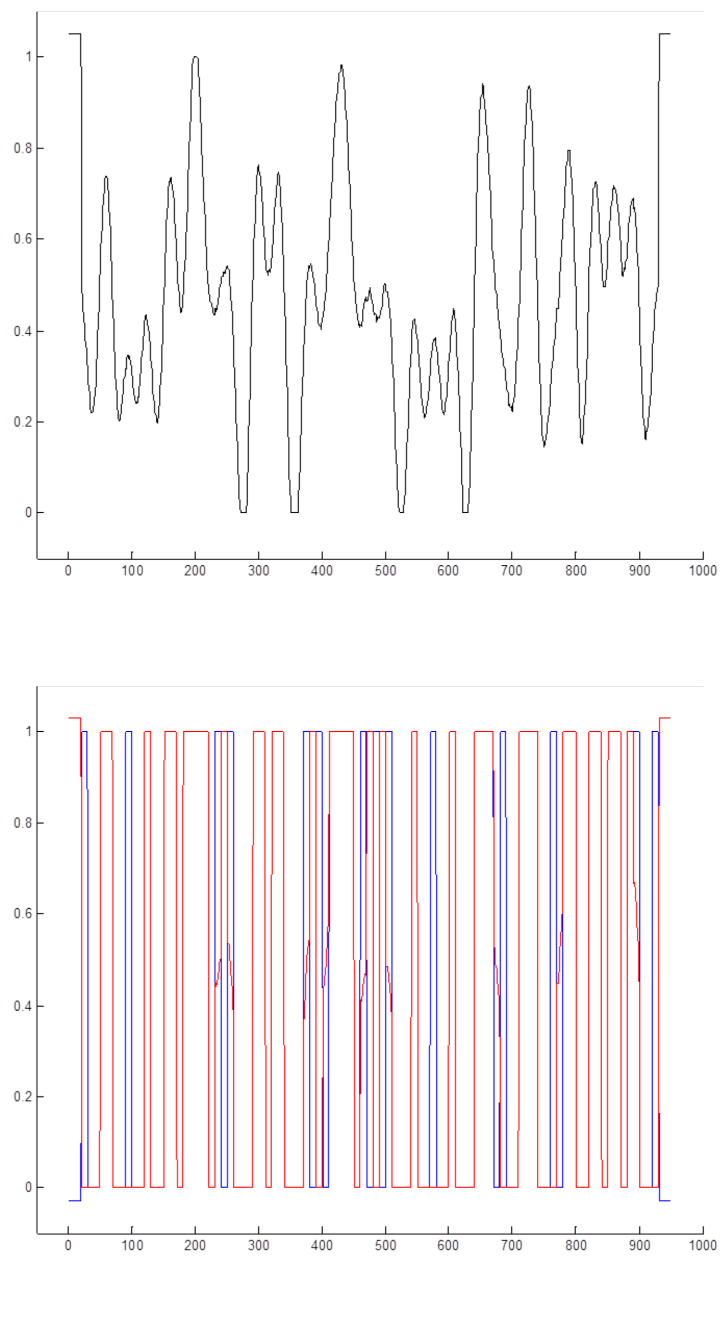
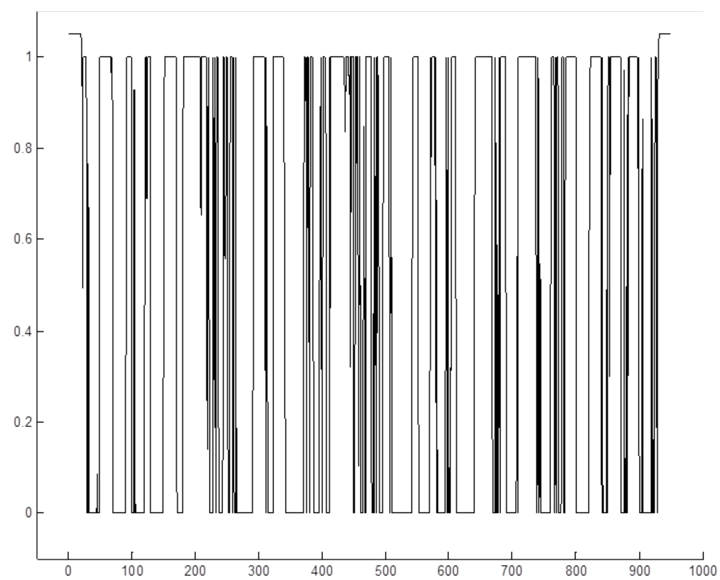


Figure 2: $\lambda = 0.1$



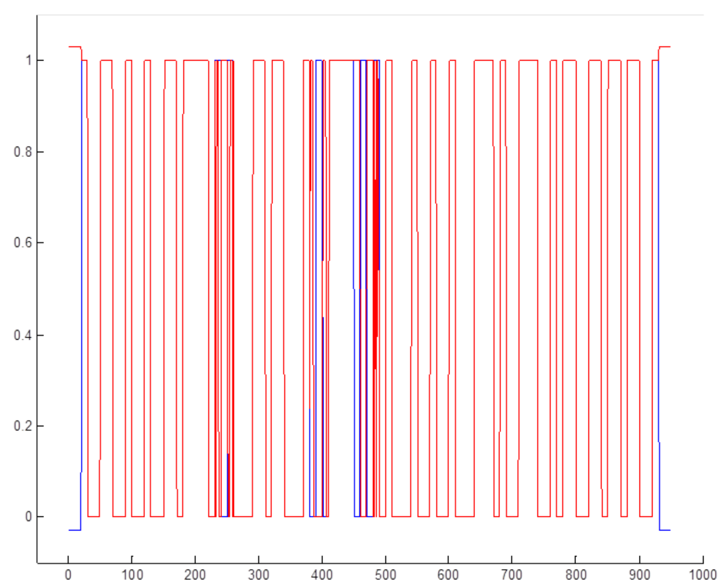
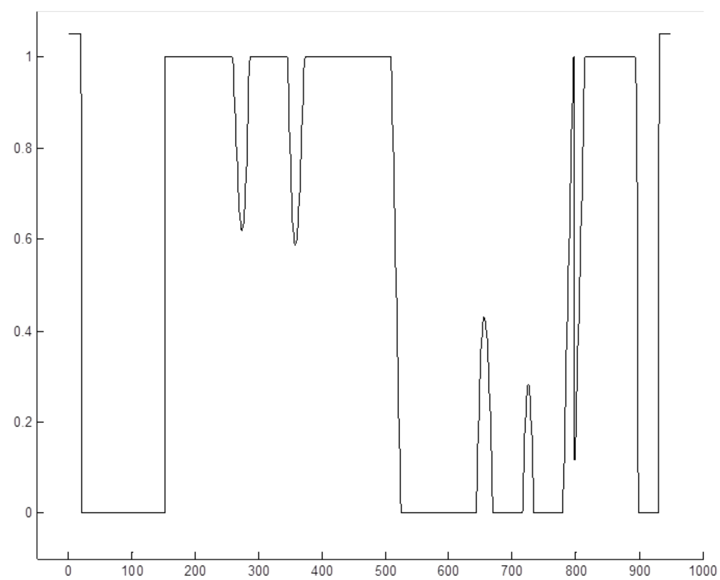
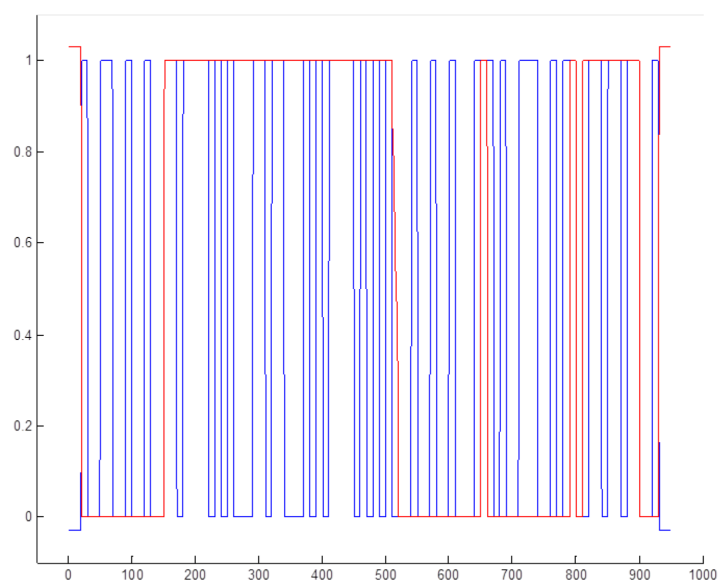


Figure 3: $\lambda = 0.0001$



1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800
900
1000



1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800
900
1000

Figure 4: $dt = 3$

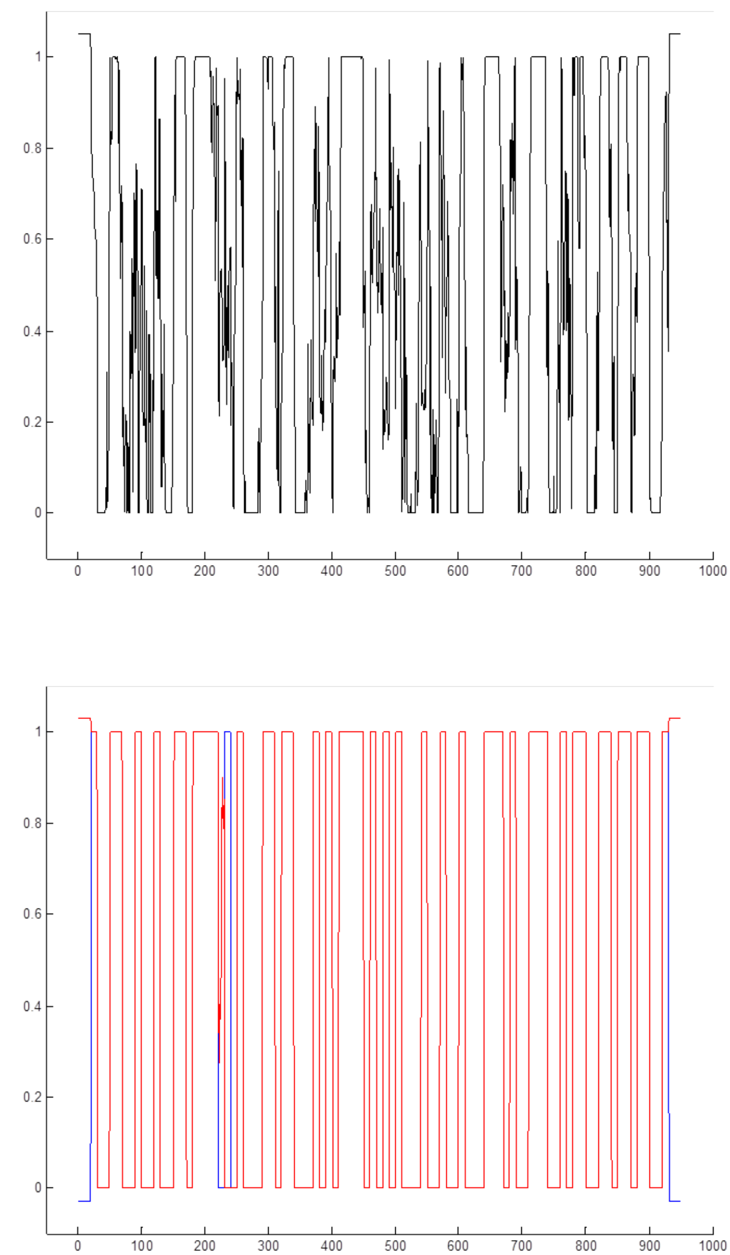

1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800
900
1000
1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800
900
1000

Figure 5: $dt = 1$

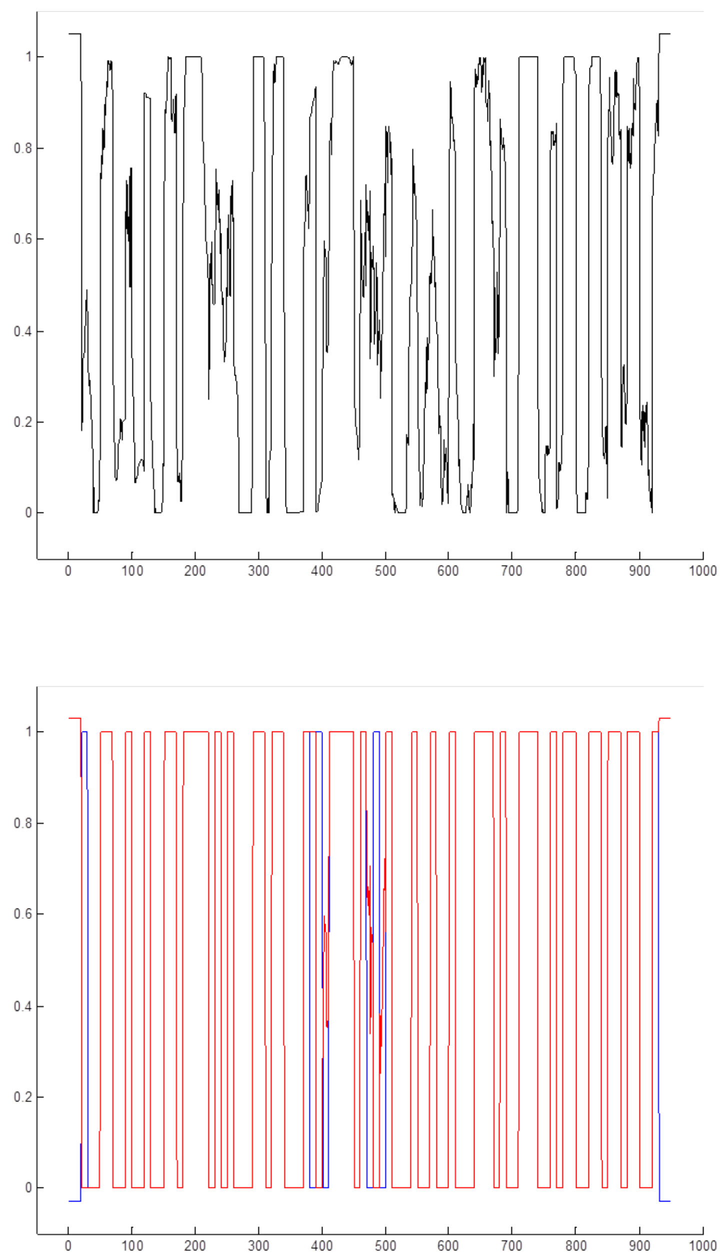


Figure 6: $dt = 0.1$

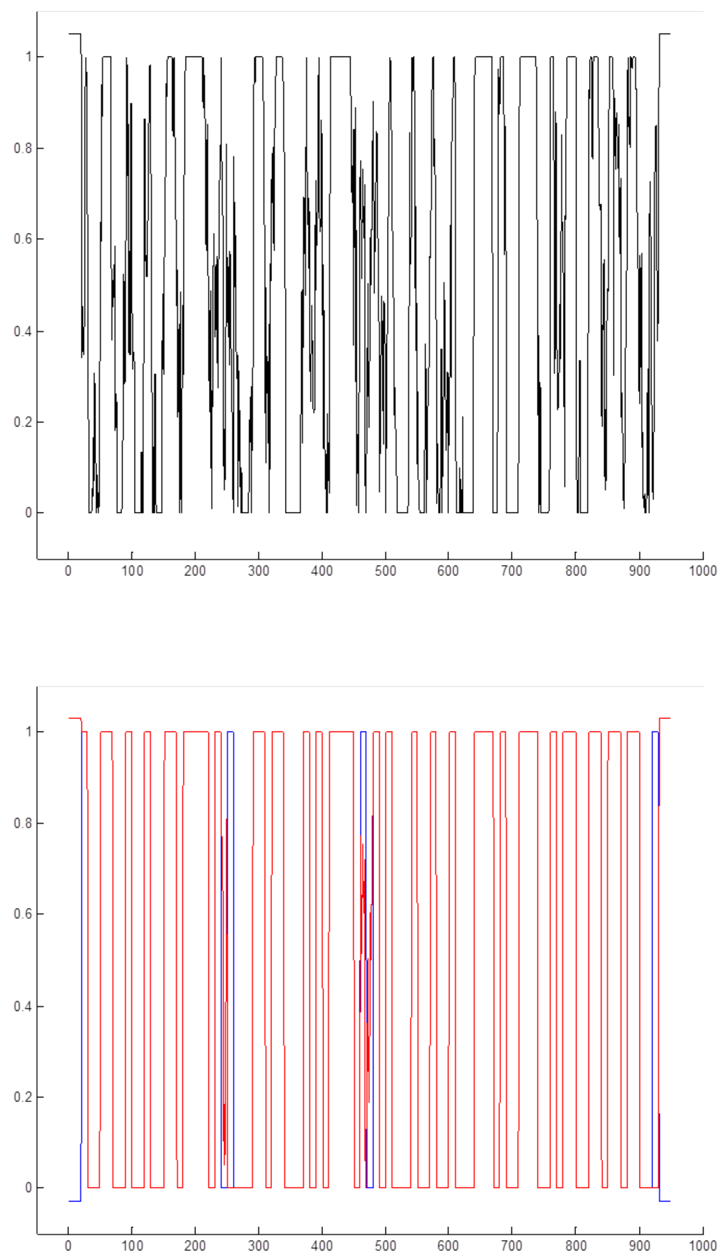


Figure 7: $Precision = 0.00001$

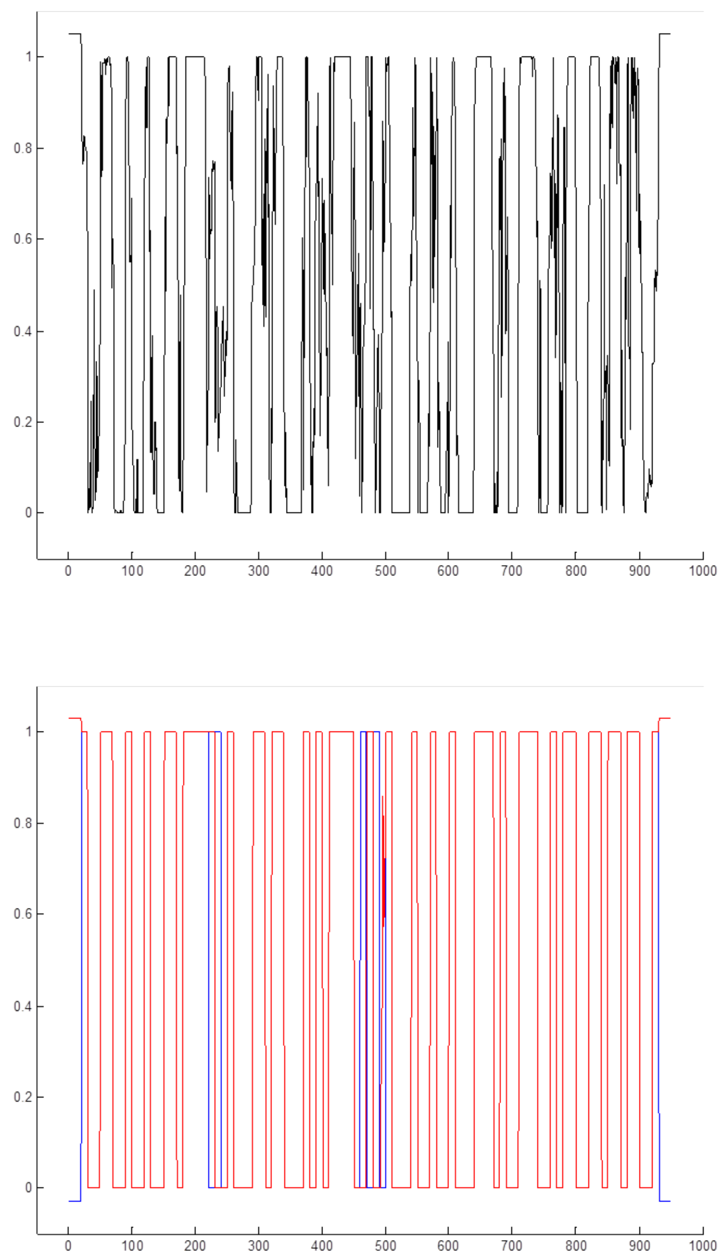


Figure 8: $Precision = 0.0000001$

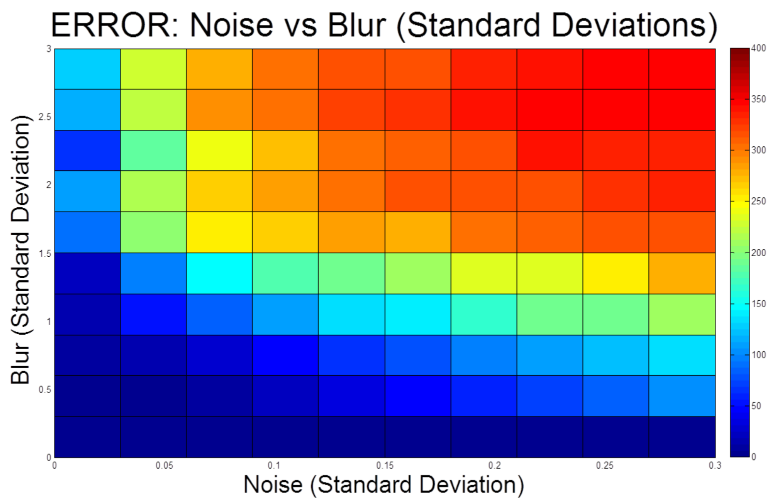


Figure 9: Only Box Constraint

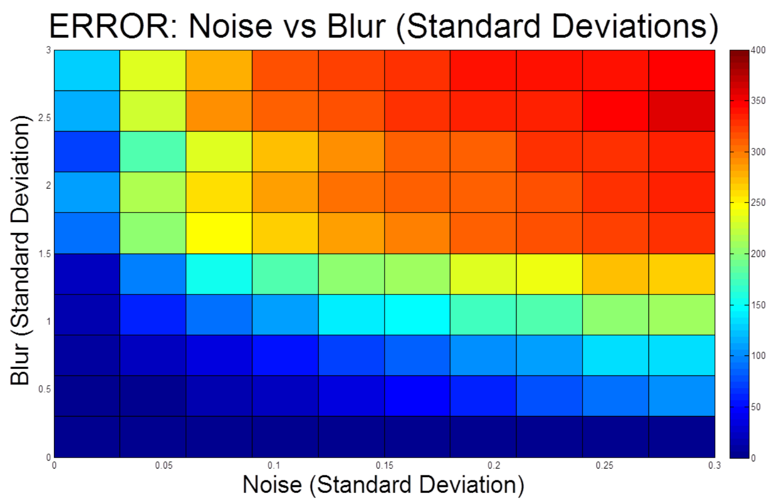


Figure 10: Thresholding

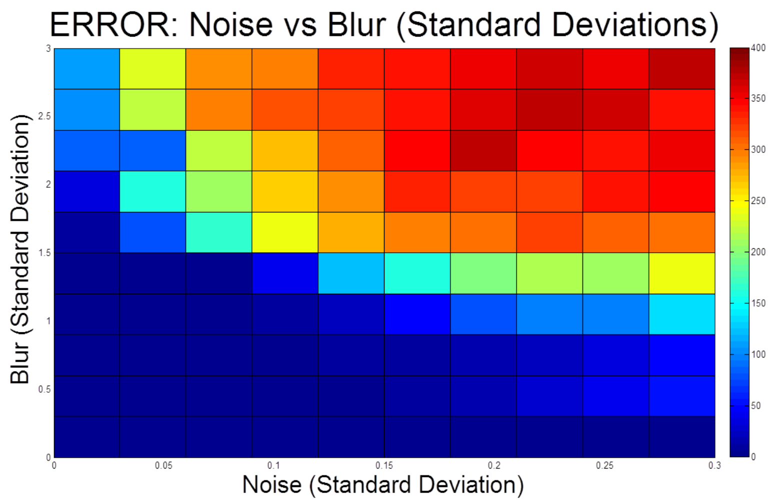


Figure 11: Adaptive Constraint and Thresholding

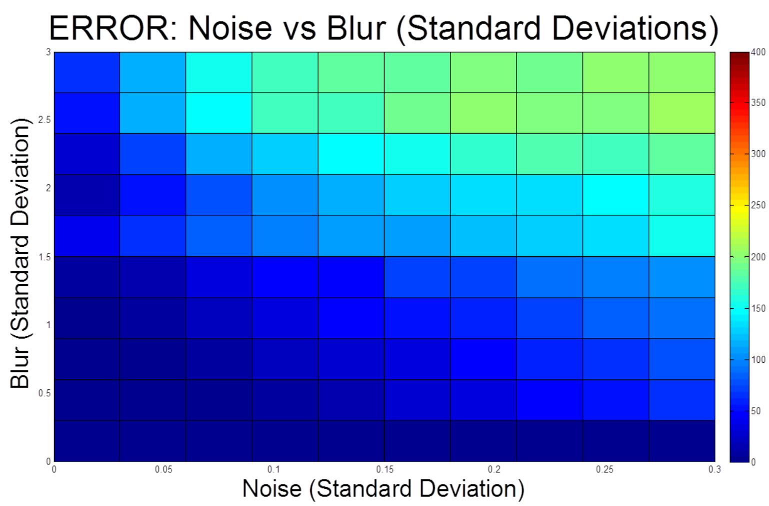


Figure 12: Barcode Structure