# THE TECHNOLOGY BEHIND BAR CODES Part A

Theo Pavlidis



**E-mail:** t.pavlidis@ieee.org

**Web site: w**ww.theopavlidis.com


The Modern Deity of Commerce!

## Pros and Cons of Bar Codes

- Bar codes make store check out easier.
- Bar codes hide the price.
  - When they were first introduced some consumer advocates were asking for markings that could be read by both machines and people.



## Examples of Bar Codes



Bar code labels (symbols) contain both computer readable and human readable information. But the information displayed is only a key to a database. Price is included only rarely (second example).

## Pros and Cons for a Key

- Prices of items can be updated easily (every few hours in places with rampant inflation).
- Price displayed with the item need not correspond to the price in the database. (This is often the case with "sale" prices.)
  - However there is a paper trail!



## **How Bar Codes Work**
(and why were designed that way)

- Information is encoded in the relative widths of the dark and light stripes.
- Computers are good at precise measurements and numerical calculations. They are not good at figuring out shapes.
- People are the opposite: Good at shapes, bad in measurements.

## Web Resources

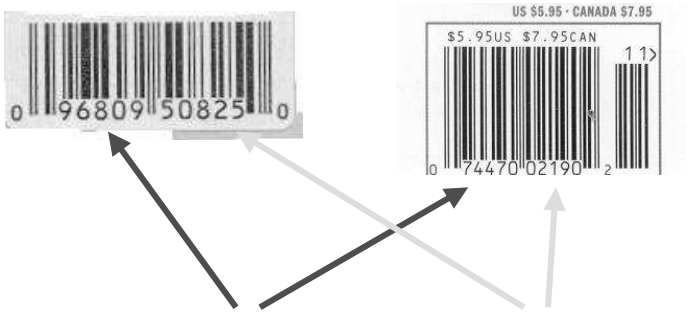
- **http://www.lintech.org/comp-per/06BARCD.pdf**



## Some Specifics

- UPC (**U**niversal **P**roduct **C**ode): It was introduced around 1970 and it is used mainly in supermarkets.
- It encodes the ten digits, each one in two **bars** and two **spaces**.
- If we use as unit the narrowest element (bar or space), the sum of the widths is equal to 7.

# Examples of UPC



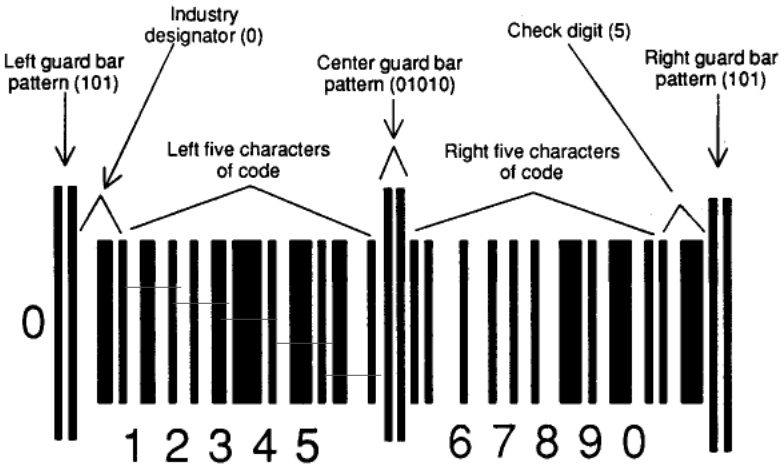

Manufacturer ---- Product



# Examples of Encoding

- Kleenex → 36000
- 85 3-ply 8.2”by 8.4” → 26085
- 12 pack of above → 22333
- 110 2-ply 8.2” by 8.4” → 28110
- Codes on product:
  - 3600,26085
    3600,22333
    3600,28110

# The Gory Details - 1





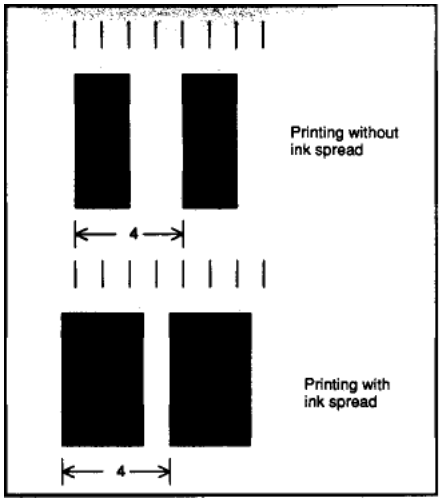

# The Gory Details - 2

| | Left (odd) | Right (even) | Width Pattern | *t*-Distances (odd) | (even) |
|---|---|---|---|---|---|
| 0 | 0001101 | 1110010 | 3,2,1,1 | 4,3 | 5,3 |
| 1 | 0011001 | 1100110 | 2,2,2,1 | 3,4 | 4,4 |
| 2 | 0010011 | 1101100 | 2,1,2,2 | 4,3 | 3,3 |
| 3 | 0111101 | 1000010 | 1,4,1,1 | 2,5 | 5,5 |
| 4 | 0100011 | 1011100 | 1,1,3,2 | 3,4 | 2,4 |
| 5 | 0110001 | 1001110 | 1,2,3,1 | 4,5 | 3,5 |
| 6 | 0101111 | 1010000 | 1,1,1,4 | 5,2 | 2,2 |
| 7 | 0111011 | 1000100 | 1,3,1,2 | 3,4 | 4,4 |
| 8 | 0110111 | 1001000 | 1,2,1,3 | 4,3 | 3,3 |
| 9 | 0001011 | 1110100 | 3,1,1,2 | 3,2 | 4,2 |

Code expressed through modules

Code expressed through widths

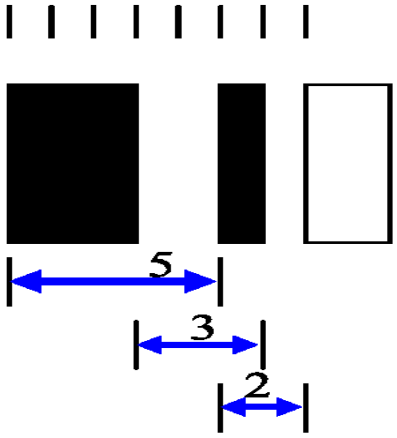
# Real World Problems –1a




Engineering must deal with real world imperfections. Because of ink spread, bar widths are greater than space widths of the same (theoretical) value. The distance between the start of two elements is called the *t* distance.



# Real World Problems –1b




Triplet of *t*-distances

0 = 5,3,2
1 = 4,4,3
2 = 3,3,4
3 = 5,5,2
…
7 = 4,4,3
8 = 3,3,4

## Real World Problems –1c

- The *t*-distances are not always unique, even if we add a third number.
- The (even) pairs 1/7 and 2/8 have the same triplets so there is a need to examine widths.

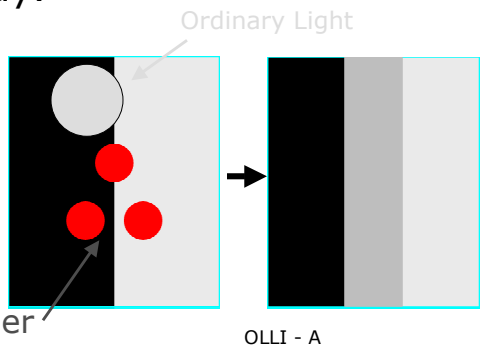


## Real World Problems -2

- Even if an image has only two colors (say black and white) a scanner element has finite dimensions, so it will average colors if its field covers an area with more than one color. We end up getting a big range of gray!

How do we get gray from black and white

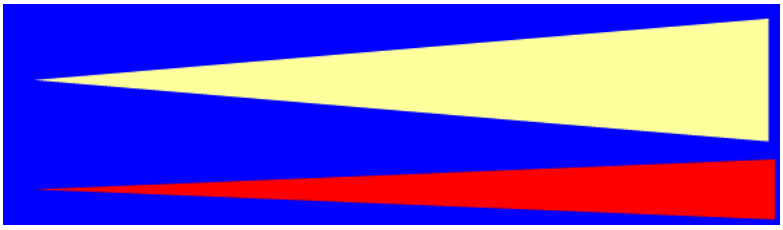
Red mark laser scanner spots and orange an ordinary light spot. If a spot saddles two colors we get gray.






# Laser Scanning

- Laser light beams stay more focused than ordinary light beams, that is why they are used for bar code scanning.



# Some Arithmetic

- Suppose the scanner spot covers five pixels. Then the scanner output will be a weighted average of the values of the five.

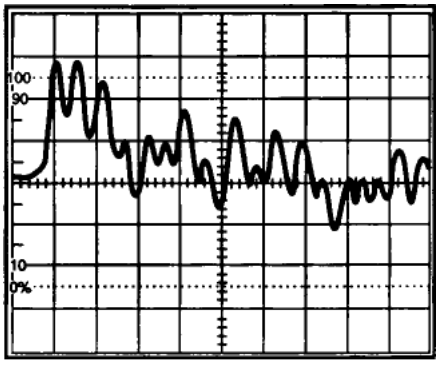
250 | 250 | **10** | 10 | 10 → 55

250/16+ 250/8+5*10/8+10/8+10/16 = 55

1/16 + 1/8 + 5/8 + 1/8 + 1/16 = 1

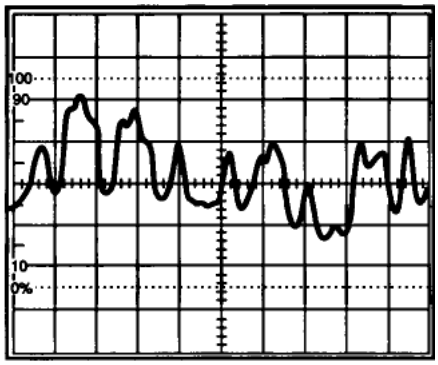


# Oscilloscope Tracing of a Bar Code with a Laser Scanner - 1




Bar Code was printed with a high quality printer so the distortion is due only to the scanner.

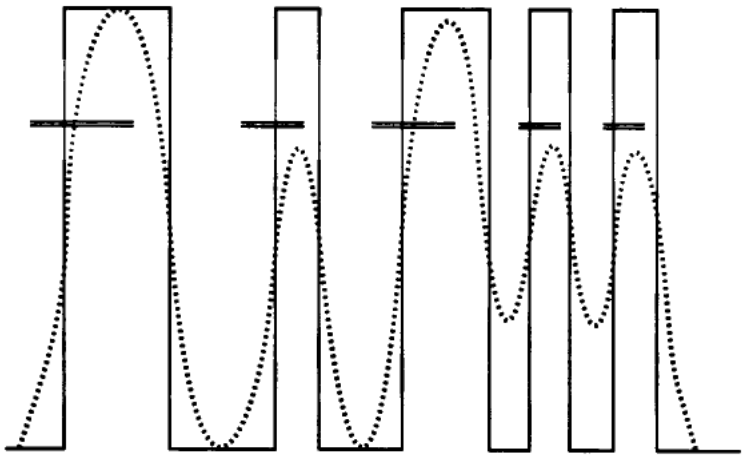
## Oscilloscope Tracing of a Bar Code with a Laser Scanner - 2




Bar code was printed with a dot matrix printer, so the distortion is due to both the printer and the scanner.



## Simulated Tracing



## Decoding Bar Codes is Harder than it Looks!

- Because of distortions due to the printer and the scanner, decoding bar codes is a challenging problem.
- There is an interesting trade-off: Use computing power (cheap these days) to make up for distortions caused by low quality (cheap) optics!



## De-blurring

- We can decodes bar code scans if we de-blur them. But de-blurring is a mathematically **ill-defined** problem. (A bit like dividing by a number close to zero.)
- We need clever mathematical "tricks" that can be implemented on cheap micro-processor and run in milliseconds.

## Help for Decoders

- The arrangement of bars and spaces is not arbitrary but subject to several constraints.
- Symbols contain "checksums" that make possible error detection. (Keep scanning until we get a valid checksum.)



## Bar Code Types

- **UPC** – encodes only digits (used in supermarkets)
- **Code 39** – it has 44 code words: 10 digits, 26 letters, and 8 special symbols ($, /, …)
- **Code 128** – it has 105 code words
- Etc, etc, etc.

# Code 39

- 3 of 9: three wide elements out of nine, five bars and four spaces
- Standard for the Department of Defense since 1980.
- In contrast to UPC, it has no specifications for the printed symbol.



# Code 39 - samples

| | Bars | Spaces | Pattern |
|---|---|---|---|
| 1 | 10001 | 0100 |  |
| 2 | 01001 | 0100 |  |
| A | 01001 | 0010 |  |
| K | 10001 | 0001 |  |