**CHAPTER 5**

**ALGORITHM**

**5.1 Approach / Method**

There are number of algorithms exists for barcode calculation, they all have there advantages and dis-advantages,

In some algorithms the barcode is firstly extracted/ separated from the image and then it’s computation is performed, But it is very time consuming, We used different approach in this algorithm first image is enhanced and we make it clear.

Then we compute the intensities horizontally by going down wards in image, this all is performed automatically by the system, for this purpose we make certain constraints and coded them to make it automatic. The left and right side obvious pixels are also separated by performing computation, the will not match out bit pattern of barcode and on the basis of properties of barcode they are ignored.

Firstly we convert the image in to pure black and white image and perform some transformation on it in a symmetric manner, like we perform imtophattransform(image,);,

In the above image we can see that the image is not stable, and on top portion the light is enough and in down portion the light is not enough,

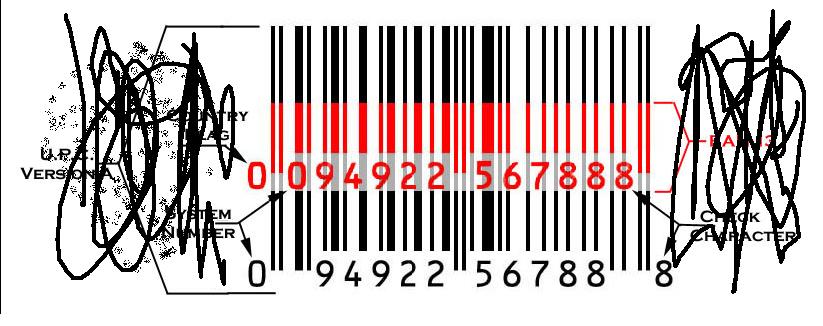
This transformation makes the bars in the image stable and straight .Then we perform im2bw(image); transformation on the image that converts image in to pure black and white image.

**5.2 Computation:**

Developed 5 functions for barcode decoding.

**5.2.1 Part-1**

* The first function computes the intensity on the columns of the image and returns them in an array , then it is assigned to “C”,



In the above image we can see the unwanted objects in the image, they mean nothing to us.

The length of the “c” is equal to the number of columns in the image, In barcode we assume the Black space as 1 and white space as 0, we put a constraint while recording the intensity information from the image, if the intensity in row at location (x,y) is equal to white mean 1, we store 0 in our array and if there is Black space the matlab consider it as 0 but we mention it in array as 1.

**Result:**

c = 823

>>C=823

**Result**

Because the image have 823 columns, as result we see 823 intensities.

**5.2.2 Part-2**

* The second function computes the number of black and white bar’s, then store the number of bars in to “td”

For this purpose we put a constraint that if the first and second intensity value stored in the array c is not same then increment the counter, by this way we got the total number of black and white bars.

**Result:**

td = 73

There are 73 bars in the image including the unwanted elements,These bar’s computed intensities will be aggregated and then used for computation.

**5.2.3 Part-3**

* The third function computes the number of bits and aggregates the bit information then sends them back.

We gave the horizontal bit Intensities/information and the number of bars in image. As an input

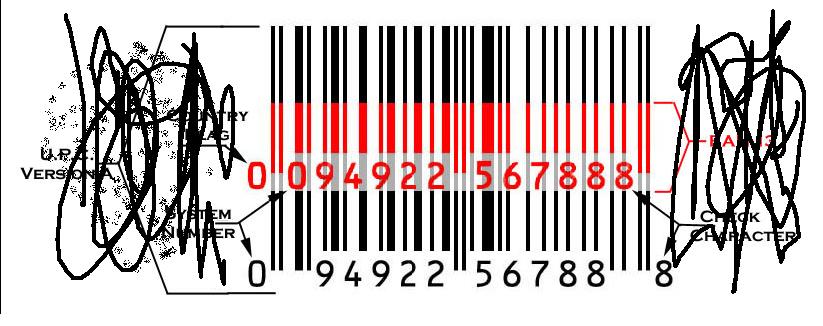
IF there are 4 zeros instead of keeping these 4 zeros we only maintain the 4 , and if there are 12 I’s we maintain the 12 in the record instead of 12 1’s.

**Result:**

As a result we get the collective intensities information it also contains unwanted bars information.

**5.2.4 Part-4**

* The Fourth function computes the K&G the extra bits from left”K”, and extra bits from Right “G”.



We take the minimum and maximum value stored in array and compare them , we collect the maximum of the starting 11 locations of array and compare it with the minimum.

**Result:**

k = 9

g =7

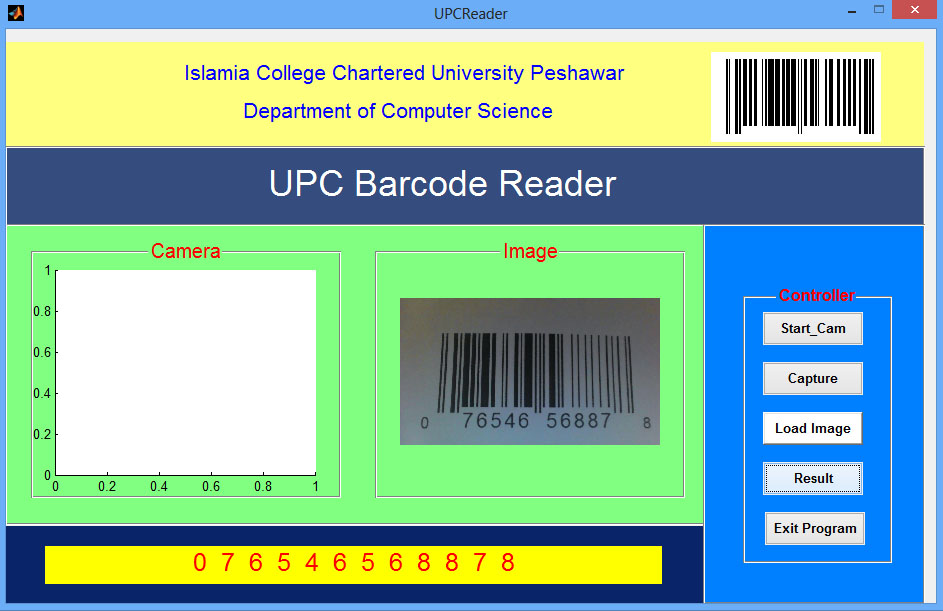
As a result we get two value, on bases of them we conclude that how many values stored in array are ambiguous for us, and separate them from the meaning full bar values.

The first ‘9’ bar’s information is ambigious, and the last ‘7’ bar’s information is ambigious, we will surpass it and will use the remaining bar information for calculation.

**5.2.5 Part-5**

* The fifth function computes the barcode and takes the stored information from array and match it with the barcode and right side pattern table. and flash result on screen.

**5.3 Interface Design**

 Interface is the basic part in the software development phase, The software is design by keeping in mind all of the usability heuristics.

**5.4 Design objectives**

Means designing the software that follows the design objectives.

**5.4.1 Easy to Understand**

Each and every component is labeled, and easy to learn at first look, no complication in the design are kept,

**5.4.2 Flexible**

The software is also flexible, and can easily can be adopt in any organization.

**5.4.3 Reliable**

The software is reliable, and can easly be operate.

**5.4.4 Efficient to operate**

Software is also easy to operate

**5.5 Components**

There are five buttons to operate the barcode reader,

' Start cam' it starts the web cam and show the frames of video on the camera tab, 'Capture' it captures the frame from the video and shows in the Image tab,

'Load Image' it loads the image from the computer directories, and loads the images which are stored in the Computer.

'Result' Whole algorithm is working under this button, by pressing it the image from the image tab is forwarded to the algorithm and computation is performed, The result is displayed down on the screen.

'Exit' By pressing it the software is closed.