

# Homework 3: Introduction to Image and Video Processing

Huang Yiping

## 1 Compression

- 1.1 Make your own Huffman code for your last name in Matlab or Python. Demonstrate the results and discuss them.**
- 

**Answer:**

The string to be encoded is "Huang", and the encoded string for the name is: 110101110100. The mapping in between the input and output looks like below:

g	00
n	01
u	10
H	110
a	111

The huffman coding is a "prefix-free" encoding, which basically means no code bit is a prefix of some other code bit. And this ensures that the decoding is possible. Also huffman coding will try to use the minimum bit length for the symbols that occur most often and will use more bits for the one which occur least often. In the case of the sample string, both occur with the same possibility, but due to the order the heap tree is compressed, they get different length depending on how they are folded into the compressed tree. The specific order implemented in the code is described below.

Many possible encodings can be produced by the huffman algorithm so long they have the same average length. The different ordering is due to two factors. First when implementing the heap, the placement of the vertex when their frequencies are tight can be decided arbitrarily. For this specific implementation, the ordering of the symbol in the unicode encoding is also taken into account, which means when two symbols have the same frequencies, the one ordered in the front in the unicode encoding is taken as the smaller one for all the operations involving comparison.

Secondly in the compression step, the algorithm just removes two smallest children from the heap, and combine their values to form a new child and add it to the heap until there is only one child left. Then when we go down the tree, we can use different binary digit to specify the path. One can designate digit to different branch at different level arbitrarily so long that the branch at the same level assigned to different digit, but in the scheme of the submitted code, the left branch is designated with 0 and the right branch with 1, thus produces the result as it is shown above (together with the specific order in the heap when frequencies are tight as explained above).

The entropy of the original string is also calculated, using the formula  $\sum -p * \log_2(p)$ , and the result is 2.32. The

average length of the path is calculated using the formular  $\sum p_i * |C_i|$ , where  $|C_i|$  is the length of the ith symbol encoding. The result is 2.4. As we can see, the average length of the path is very close to the entropy, so the encoding is efficient in terms of compression.

---

## 2 Morphological image processing

Choose an image of your liking, that would be interesting in its black and white version

**2.1 Binarize it using a threshold that is based on the image statistics. Hint: e.g. by choosing a certain amount of standard deviations away from the intensity mean. This is just an option, you are encouraged to look up optimal thresholding methods.**

---

**Answer:**

The image chosen is shown below. This is a recent assignment I submitted for the course mathematical simulation. I used the camScanner, an app popular for quick scanning images with features including auto-alignment and sharpening. Somehow this scanning result is not satisfactory, probably due to the sensor of the camera on my mobile phone. There are a lot of noises in the image which are not present in the paper I wrote.

Question 1

$$\text{For } -\frac{1}{2}\pi \leq x < 0 \quad F(x) = \int_{-\frac{1}{2}\pi}^x \sin x dx \\ = -\frac{1}{2}\cos x + C_1 \\ F(-\frac{1}{2}\pi) = 0 \Rightarrow C_1 = 0 \\ \text{so } F(x) = -\frac{1}{2}\cos x \quad \int_{-\frac{1}{2}\pi}^0 f(x) = \frac{1}{2}$$

$$\text{For } 0 \leq x < \frac{\pi}{2} \quad F(x) = \int_0^x \frac{1}{2}x^3 dx \\ = \frac{x^4}{24} + C_2 \\ F(0) = \int_{-\frac{1}{2}\pi}^0 f(x) = \frac{1}{2} \\ \Rightarrow C_2 = \frac{1}{2} \\ \text{so } F(x) = \frac{x^4}{24} + \frac{1}{2} \quad \int_{-\frac{1}{2}\pi}^{\frac{\pi}{2}} f(x) = \frac{2^4}{24} + \frac{1}{2} = \frac{11}{12}$$

$$\text{For } \frac{\pi}{2} \leq x \leq 2e^{\frac{1}{2}} \quad F(x) = \int_{\frac{\pi}{2}}^x \frac{1}{2}x dx \\ = (\ln x + C_3) \quad (\text{cancel } \infty) \\ = \ln x + C_3 \quad (\text{since } \infty) \\ F(\frac{\pi}{2}) = (\ln 2 + C_3) = \frac{11}{12} \\ C_3 = \frac{11}{12} - \ln 2 \\ \text{so } F(x) = \ln x + \frac{11}{12} - \ln 2 \\ F(2e^{\frac{1}{2}}) = 1 \quad \text{so } F(x) \text{ is valid.}$$

$$\text{so } F(x) = \begin{cases} 0 & x < -\frac{1}{2}\pi \\ \frac{1}{2}\cos x & -\frac{1}{2}\pi \leq x < 0 \\ \frac{x^4}{24} + \frac{1}{2} & 0 \leq x < \frac{\pi}{2} \\ (\ln x) + \frac{11}{12} - \ln 2 & \frac{\pi}{2} \leq x \leq 2e^{\frac{1}{2}} \\ 1 & x > 2e^{\frac{1}{2}} \end{cases}$$

$$\begin{aligned} \frac{1}{2}\cos x &= u & (0 \leq u < \frac{1}{2}) \\ x &= \arccos 4u \\ \frac{x^4}{24} + \frac{1}{2} &= u & (\frac{1}{2} \leq u < \frac{11}{12}) \\ x &= (24u - 6)^{\frac{1}{4}} \\ (\ln x) + \frac{11}{12} - \ln 2 &= u & (\frac{11}{12} \leq u < 1) \\ x &= e^{u - \frac{11}{12}} \end{aligned}$$

$$\begin{aligned} \text{so we generate } U(0,1) \\ \text{if } u < \frac{1}{2} & \quad x = \arccos 4u \\ \text{else if } u < \frac{11}{12} & \quad x = (24u - 6)^{\frac{1}{4}} \\ \text{else } & \quad x = e^{u - \frac{11}{12}} \end{aligned}$$

In order to extract the text from the image, the first step is to binarize the image. One way to select the global cutoff point is through statistical method such as Otsu's method. The Otsu's method tries to maximize the between-class variance, which is defined as  $P_1(m_1 - m_G)^2 + P_2(m_2 - m_G)^2$ , where  $P_1$  is the probability of class 1,  $P_2$  the probability of class 2, which is  $1 - P_1$ . The  $m_1$  is the mean of class 1,  $m_2$  is the mean of class 2, and  $m_G$  is the mean of the global. The intuition is that the two classes should be as different as possible in terms of intensity. Using this method, the threshold is calculated to be 159. The binarized result is shown below. However, we can see that there are still many noises after the binarization, particularly in the corners of the image. If we then apply the morphological methods, then the chances are that the script would also be removed together with the noise since they are both small in size compared to the background.

binary image without filtering

function 1

$$\text{For } -\frac{1}{2}\pi \leq x < 0 \quad F(x) = \int_{-\frac{1}{2}\pi}^x -\frac{1}{4} \sin x dx \\ = -\frac{1}{4} \cos x + C_1$$

$$F(-\frac{1}{2}\pi) = 0 \Rightarrow C_1 = 0$$

$$\text{so } F(x) = -\frac{1}{4} \cos x \quad \int_{-\frac{1}{2}\pi}^0 -\frac{1}{4} \cos x dx = \frac{1}{4}$$

$$\text{For } 0 \leq x < \frac{\pi}{2} \quad F(x) = \int_0^x \frac{1}{8} x^3 dx$$

$$= \frac{x^4}{24} + C_2$$

$$F(0) = \int_{-\frac{1}{2}\pi}^0 -\frac{1}{4} \cos x dx = \frac{1}{4}$$

$$\Rightarrow C_2 = \frac{1}{4}$$

$$\text{so } F(x) = \frac{x^4}{24} + \frac{1}{4} \quad \int_{-\frac{1}{2}\pi}^{\frac{\pi}{2}} f(x) dx = \frac{2^4}{24} + \frac{1}{4} = \frac{11}{12}$$

$$\text{For } \frac{\pi}{2} \leq x \leq 2e^{\frac{\pi}{12}}$$

$$F(x) = \int_{\frac{\pi}{2}}^x \frac{1}{8} x dx$$

$$= h|x| + C_3 \quad (\text{circled})$$

$$= h|x| + C_3 \quad (\text{since } \infty)$$

$$F(\pi) = h\pi + C_3 = \frac{11}{12}$$

$$C_3 = \frac{11}{12} - h\pi$$

$$\text{so } F(x) = h|x| + \frac{11}{12} - h\pi$$

$$F(2e^{\frac{\pi}{12}}) = 1 \quad \text{so } F(x) \text{ is valid.}$$

$$\text{so } F(x) = \begin{cases} 0 & x < -\frac{1}{2}\pi \\ -\frac{1}{4} \cos x & -\frac{1}{2}\pi \leq x < 0 \\ \frac{x^4}{24} + \frac{1}{4} & 0 \leq x < \frac{\pi}{2} \\ h|x| + \frac{11}{12} - h\pi & \frac{\pi}{2} \leq x < 2e^{\frac{\pi}{12}} \\ 1 & x > 2e^{\frac{\pi}{12}} \end{cases}$$

$$\frac{1}{4} \cos x = u \\ x = \arccos 4u \quad (0 \leq u < \frac{1}{4})$$

$$\frac{x^4}{24} + \frac{1}{4} = u \\ x = (24u - b)^{\frac{1}{4}} \quad (\frac{1}{4} \leq u < \frac{11}{12})$$

$$h|x| + \frac{11}{12} - h\pi = u \\ x = 2e^{u - \frac{\pi}{12}} \quad (\frac{\pi}{12} \leq u < 1)$$

so we generate  $U(0, 1)$

if  $U < \frac{1}{4}$   $x = \arccos 4u$

else if  $U < \frac{11}{12}$   $x = (24u - b)^{\frac{1}{4}}$

else  $x = 2e^{u - \frac{\pi}{12}}$

In order to minimize the noise, a gaussian filter with standard deviation of 1 is applied on the image first. Then we calculate the threshold again using the Otsu's method. This time the threshold is calculated to be 183. And the resulting binarized image is shown below. As we can see, this time the image has a much cleaner background than before. This is as expected, since the smoothing changes the shape of the histogram of the image, making it much easier to separate the foreground (in this case are the low pixels of the script) from the background (the bright pixel of the paper).

binary image with gaussian filtering

function 1

$$\text{For } -\frac{1}{2}\pi \leq x < 0 \quad F(x) = \int_{-\frac{1}{2}\pi}^x \sin x dx \\ = -\frac{1}{2}\cos x + C_1$$

$$F(-\frac{1}{2}\pi) = 0 \Rightarrow C_1 = 0$$

$$\text{so } F(x) = -\frac{1}{2}\cos x \quad \int_{-\frac{1}{2}\pi}^0 \sin x dx = -\frac{1}{2}$$

$$\text{For } 0 \leq x < \frac{1}{2}$$

$$F(x) = \int_0^x \sin x dx$$

$$= \frac{x^2}{2} + C_2$$

$$F(0) = \int_{-\frac{1}{2}\pi}^0 \sin x dx = -\frac{1}{2}$$

$$\Rightarrow C_2 = -\frac{1}{2}$$

$$\text{so } F(x) = \frac{x^2}{2} + -\frac{1}{2} \quad \int_{-\frac{1}{2}\pi}^{\frac{1}{2}} f(x) dx = \frac{2^2}{24} + -\frac{1}{2} = -\frac{1}{2}$$

$$\text{For } \frac{1}{2} \leq x \leq 2e^{\frac{1}{2}}$$

$$F(x) = \int_{\frac{1}{2}}^x \sin x dx$$

$$= C_3(x) + C_4$$

$$= C_3(x) \quad (\text{since } x \rightarrow \infty)$$

$$F(2) = C_3 + C_4 = \frac{11}{12}$$

$$C_4 = \frac{11}{12} - C_3$$

$$\text{so } F(x) = C_3(x) + \frac{11}{12} - C_3$$

$$F(2e^{\frac{1}{2}}) = 1 \quad \text{so } F(x) \text{ is valid.}$$

$$\text{so } F(x) = \begin{cases} 0 & x < -\frac{1}{2}\pi \\ \frac{1}{2}\cos x & -\frac{1}{2}\pi \leq x < 0 \\ \frac{x^2}{24} + \frac{1}{2} & 0 \leq x < \frac{1}{2} \\ (1-x) + \frac{11}{12} - C_3 & \frac{1}{2} \leq x < 2e^{\frac{1}{2}} \\ 1 & x > 2e^{\frac{1}{2}} \end{cases}$$

$$\frac{1}{2}\cos x = u \\ x = \arccos(2u) \quad (0 \leq u < \frac{1}{2})$$

$$\frac{x^2}{24} + \frac{1}{2} = u \\ x = (2u - b)^{\frac{1}{2}} \quad (\frac{1}{2} \leq u < \frac{11}{12})$$

$$(1-x) + \frac{11}{12} - C_3 = u \\ x = 2e^{u - \frac{11}{12}} \quad (\frac{11}{12} \leq u < 1)$$

so we generate  $U(0,1)$

$$\text{if } u < \frac{1}{2}, \quad x = \arccos(2u)$$

$$\text{else if } u < \frac{11}{12}, \quad x = (2u - b)^{\frac{1}{2}}$$

$$\text{else } x = 2e^{u - \frac{11}{12}}$$

After this step, the close operation is applied to the binarized image. The result is shown below. We see that the few noises that are still left after the last step are removed from this step. This is as expected, since the close operation is a dilation operation followed by a erosion step. In this case, the noises in the background are like holes. And the first dilation step would fill in these holes, and the erosion step then will thicken the scripts again after they got thinned (again since the foreground and background are reverse in intensity). The net effect is a cleaner image than the one in the previous step. After these steps, the result is satisfactory except some remaining noises at the bottom of the paper. But these can be easily got rid of by some manual filtering in the region.

function 1

$$\text{For } -\frac{1}{2}\pi \leq x < 0 \quad F(x) = \int_{-\frac{1}{2}\pi}^x -\frac{1}{4} \ln x dx \\ = -\frac{1}{4} \ln x + C_1 \\ F(-\frac{1}{2}\pi) \rightarrow \Rightarrow C_1 = 0 \\ \text{so } F(x) = -\frac{1}{4} \ln x \quad \int_{-\frac{1}{2}\pi}^0 f(x) dx = \frac{1}{4}$$

$$\text{For } 0 \leq x < 2 \quad F(x) = \int_0^x \frac{1}{4} x^3 dx \\ = \frac{x^4}{24} + C_2$$

$$F(0) = \int_{-\frac{1}{2}\pi}^0 f(x) dx = \frac{1}{4}$$

$$\Rightarrow C_2 = \frac{1}{4}$$

$$\text{so } F(x) = \frac{x^4}{24} + \frac{1}{4} \quad \int_{-\frac{1}{2}\pi}^2 f(x) dx = \frac{2^4}{24} + \frac{1}{4} = \frac{17}{2}$$

$$\text{For } 2 \leq x \leq 2e^{\frac{11}{12}} \quad F(x) = \int_2^x \frac{1}{x} dx \\ = \ln x + C_3 \quad (\text{cancel}) \\ = \ln x + C_3 \quad (\text{since } x > 0)$$

$$F(2) = C_3 = \frac{11}{12}$$

$$C_3 = \frac{11}{12} - \ln 2$$

$$\text{so } F(x) = \ln x + \frac{11}{12} - \ln 2$$

$$F(2e^{\frac{11}{12}}) = 1 \quad \text{so } F(x) \text{ is valid.}$$

after close operation

$$\text{so } F(x) = \begin{cases} 0 & x < -\frac{1}{2}\pi \\ -\frac{1}{4} \ln x & -\frac{1}{2}\pi \leq x < 0 \\ \frac{x^4}{24} + \frac{1}{4} & 0 \leq x < 2 \\ \ln x + \frac{11}{12} - \ln 2 & 2 \leq x \leq 2e^{\frac{11}{12}} \\ 1 & x > 2e^{\frac{11}{12}} \end{cases}$$

$$\frac{1}{4} \ln x = u \\ x = e^{4u} \quad (0 \leq u < \frac{1}{4})$$

$$\frac{x^4}{24} + \frac{1}{4} = u \\ x = (24u - 6)^{\frac{1}{4}} \quad (\frac{1}{4} \leq u < \frac{11}{12})$$

$$\ln x + \frac{11}{12} - \ln 2 = u \\ x = 2e^{u - \frac{11}{12}} \quad (\frac{11}{12} \leq u < 1)$$

so we generate  $U(0,1)$

if  $U < \frac{1}{4}$   $x = e^{4U}$

else if  $U < \frac{11}{12}$   $x = (24U - 6)^{\frac{1}{4}}$

else  $x = 2e^{U - \frac{11}{12}}$

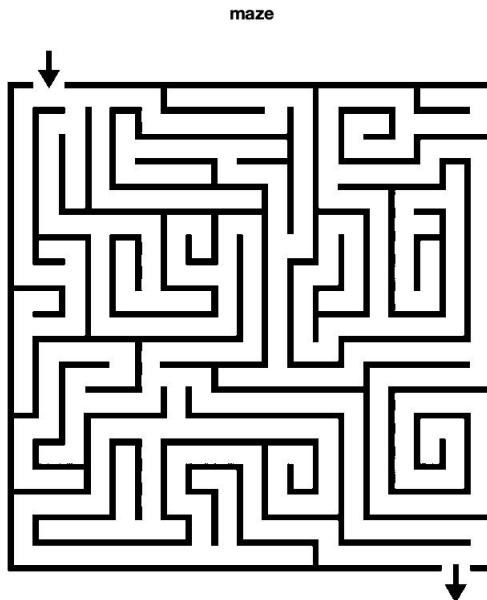
### 3 Alternative Bonus

#### 3.1 Make your own hit-and-miss transform example, implement it, and explain your results.

**Answer:**

The image chosen is shown below. The image is a maze, and my goal is to identify the location where the joint

sections are, i.e., where two roads cross each other.

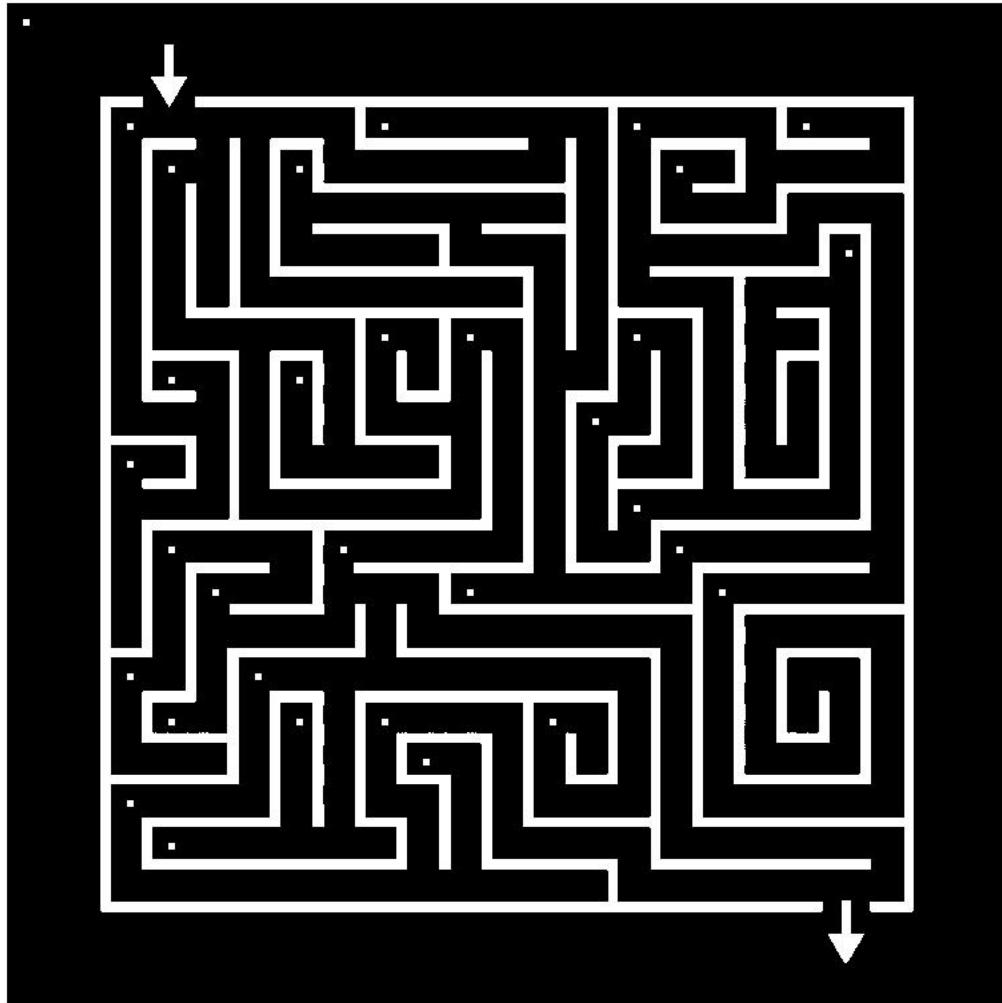


One way to think about finding these cross points is to identify separately the pattern for each cross point. This way of finding cross points links naturally to the hit miss method. For example, if we want to identify the cross point where one left end of a horizontal road meets the upper end of a vertical road, i.e., those road intersections with a top left corner, then we could specify two matrix  $B1$ ,  $B2$  as follows:

$$B1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

$$B2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

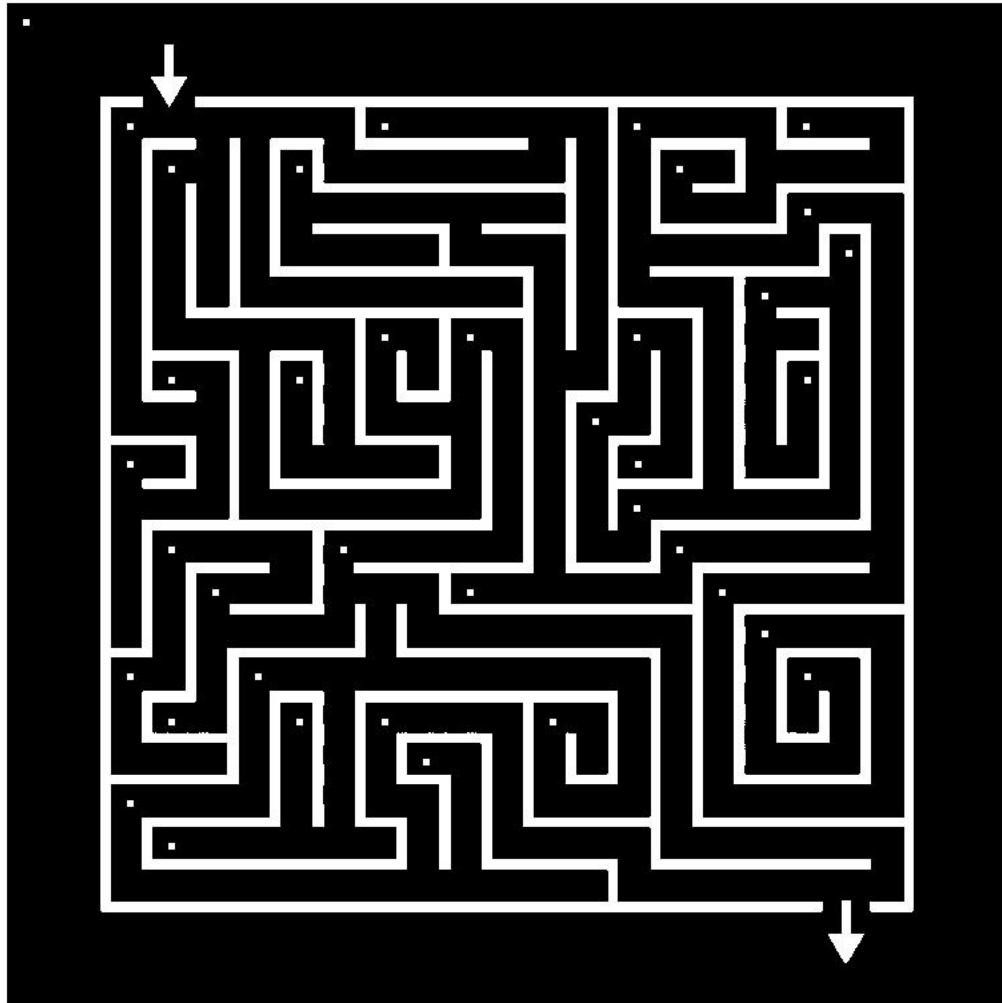
Basically, the  $B1$  matrix indicates the location we are interested, namely, where one end meets the other end. But this cannot be unconditioned, because we are interested currently only the top left corner, so we must also specify its background, which is why  $B2$  matrix comes in. What  $B2$  matrix says basically is that we need the pattern to be indeed have a background at its top and left side. The way hit miss find the location is through the formula  $(A - B1) \cap (A^c - B2)$ , which is basically the intersection of two erosions. The first erosion finds the point of interest and second erosion of the compliment of the image with the  $B2$  limits the point of interest. The result after these operations is shown below:



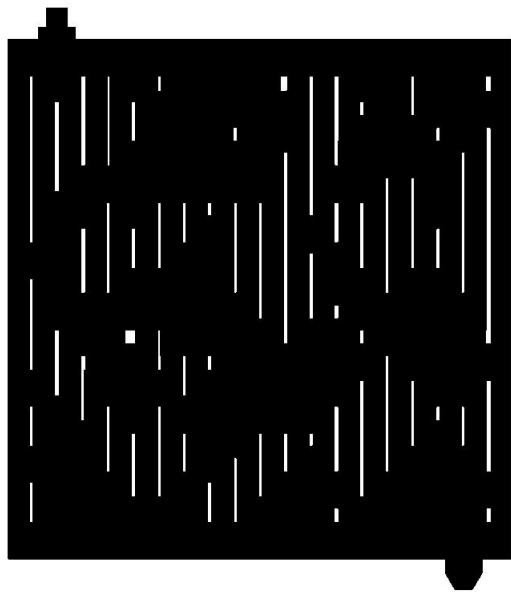
For a clear display, The locations have been shifted by 14 pixels to the right and below, and have been resized and overlayed on top of the original image. We found that it found almost all the top left corner points except at few places. For example, the one at the bottom right corner is not found. Careful inspection of these locations found that the edges of these roads are not straight. This can be viewed as another application of the hit miss operation, which is to detect product quality. A slight modification of  $B2$  matrix is made in the following form:

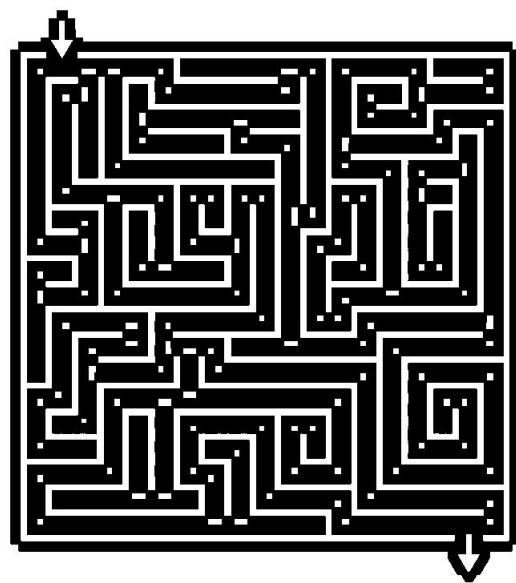
$$B2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

The bottom left corner of the matrix is filled with 0 instead of 1, which means a straight background along vertical direction is not a requirement anymore. Applying the previous operations again, the new result is shown below:



And we see that all such points are found. Then we can perform this hit miss operation multiple times with different  $B_1$  and  $B_2$  matrices to find all patterns of the locations of joint of roads. However, another easier method is simply to find all locations of horizontal and vertical roads and then find the intersection of the two, which is the joint location of interest. Doing this, we can see that the hit miss is just an erosion operator since we are not interested in how the background looks like anymore, but only the foreground patterns, i.e., the horizontal and vertical directions. For this, two matrices are specified for the horizontal and vertical directions respectively. The size of both matrices are  $40 \times 40$ . For the horizontal matrix, only 20 of the middle rows are filled with 1, and for the vertical matrix, only 20 of the middle columns are filled with 1. The found horizontal and vertical patterns and the final result of intersection overlayed on top of the original image are shown below.





We can see that it finds all the joint locations in the maze, although there are some differences in sizes from different joint points, which are due to the way the two matrices are set, and the difference of width of roads inherent in the original image. The overall effect is satisfactory.

---