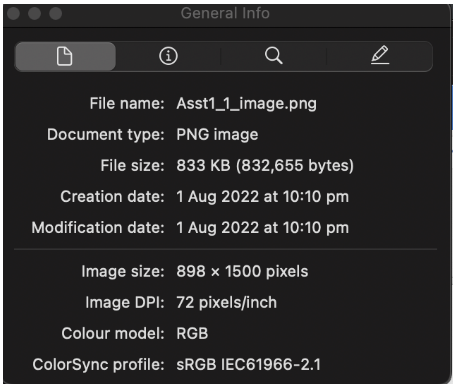
Q1



File name: Asst1_1_image.png

Document type: PNG image

File size: 833 KB (832,655 bytes)

Creation date: 1 Aug 2022 at 10:10 pm

Modification date: 1 Aug 2022 at 10:10 pm

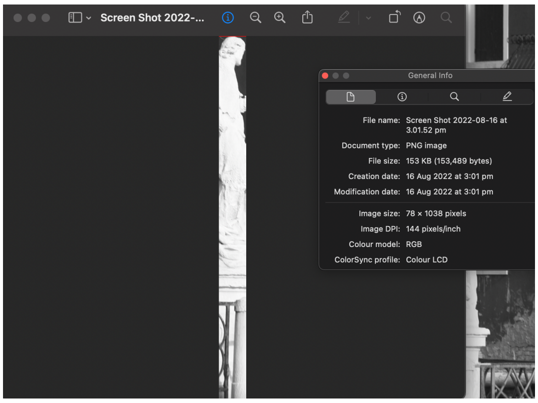
Image size: 898 × 1500 pixels

Image DPI: 72 pixels/inch

Colour model: RGB

ColorSync profile: sRGB IEC61966-2.1

This shows the information of the original image. The height is 1500 pixels.



File name: Screen Shot 2022-08-16 at 3.01.52 pm

Document type: PNG image

File size: 153 KB (153,489 bytes)

Creation date: 16 Aug 2022 at 3:01 pm

Modification date: 16 Aug 2022 at 3:01 pm

Image size: 78 × 1038 pixels

Image DPI: 144 pixels/inch

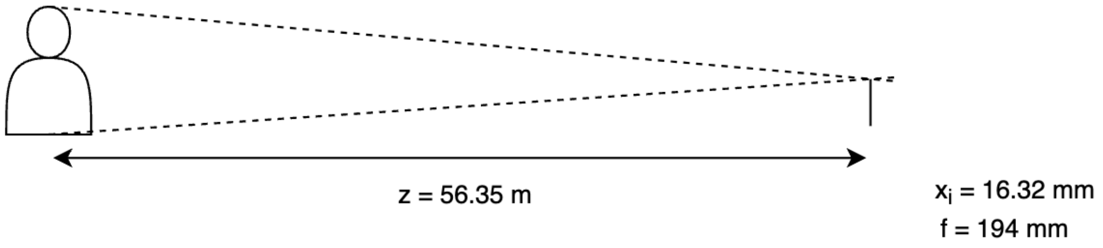
Colour model: RGB

ColorSync profile: Colour LCD

This shows the information of the screen shot from the bottom of the sculpture to the top. The height is 1038 pixels.

The ratio of the height of the sculpture to the height of the whole image is 1038/1500.

To calculate the true height of the sculpture, I calculated the true height of original image then times the ratio.



Let the true height of the whole image be x.
x = x_i/f * z = 16.32/194 * 56.35 = 4.74 m

the true height of the sculpture is x * ratio = 4.74 * 1038/1500 = 3.28 m

Q2

Maze 1:
Number of intersections = 87
Number of dead ends = 98

Maze 2:
Number of intersections = 71
Number of dead ends = 74

Method:
After I inspect the image, I found that the intersections have these patterns listed below.

$$\begin{matrix}255&0&0&255\\0&0&0&0\\0&0&0&0\\255&0&0&255\end{matrix}$$ for ┼

$$\begin{matrix}255&255&255&255\\0&0&0&0\\0&0&0&0\\255&0&0&255\end{matrix}$$ for ┬

$$\begin{matrix}255&0&0&255\\255&0&0&0\\255&0&0&0\\255&0&0&255\end{matrix}$$ for ├

$$\begin{matrix}255&0&0&255\\0&0&0&255\\0&0&0&255\\255&0&0&255\end{matrix}$$ for ┤

$$\begin{matrix}255&0&0&255\\0&0&0&0\\0&0&0&0\\255&255&255&255\end{matrix}$$ for ┴

Intuitively, I designed 5 different kernels to detect 5 types of intersections.

$$\begin{matrix}0&1&1&0\\1&1&1&1\\1&1&1&1\\0&1&1&0\end{matrix}$$ for ┼

$$\begin{matrix}0&0&0&0\\1&1&1&1\\1&1&1&1\\0&1&1&0\end{matrix}$$ for ┬

$$\begin{matrix}0&1&1&0\\0&1&1&1\\0&1&1&1\\0&1&1&0\end{matrix}$$ for ├

$$\begin{matrix}0&1&1&0\\1&1&1&0\\1&1&1&0\\0&1&1&0\end{matrix}$$ for ┤

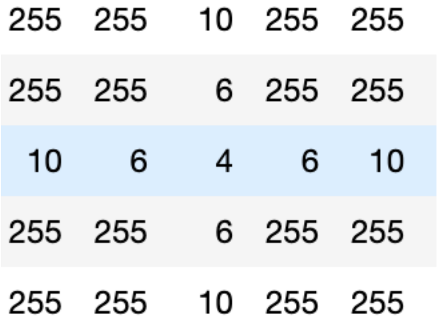
$$\begin{matrix}0&1&1&0\\1&1&1&1\\1&1&1&1\\0&0&0&0\end{matrix}$$ for ┴

However, this method requires 5 kernels, and it will over count when the intersection is a cross.

Then, I try to use only one kernel to detect all 5 types of intersections.

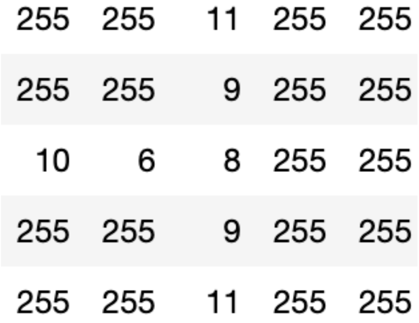
$$\begin{matrix}\frac{1}{255}&\frac{2}{255}&\frac{2}{255}&\frac{1}{255}\\\frac{2}{255}&1&1&\frac{2}{255}\\\frac{2}{255}&1&1&\frac{2}{255}\\\frac{1}{255}&\frac{2}{255}&\frac{2}{255}&\frac{1}{255}\end{matrix}$$

Using this kernel, cross intersections will give the result of 4.



255	255	10	255	255
255	255	6	255	255
10	6	4	6	10
255	255	6	255	255
255	255	10	255	255

And other intersections will give the result of 8.



255	255	11	255	255
255	255	9	255	255
10	6	8	255	255
255	255	9	255	255
255	255	11	255	255

Simply count the pixels with value 4 and 8. Add them together to get the number of intersections.

For the detection of dead ends, I identified 4 types of dead ends.

255	255	255	255
255	0	0	255
255	0	0	255
255	0	0	255

255	0	0	255
255	0	0	255
255	0	0	255
255	255	255	255

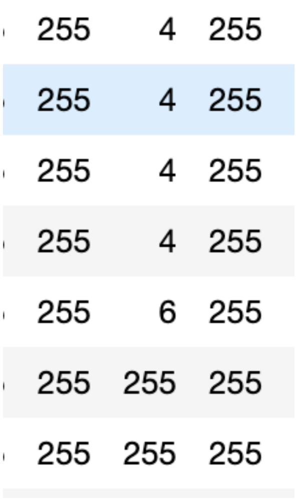
255	255	255	255
0	0	0	255
0	0	0	255
255	255	255	255

255	255	255	255
255	0	0	0
255	0	0	0
255	255	255	255

At first, I also designed 4 kernels for each type of dead ends.
Then, I came up with one kernel to detect all types of dead ends which is

$$\begin{matrix} 0 & \frac{1}{255} & \frac{1}{255} & 0 \\ \frac{1}{255} & 1 & 1 & \frac{1}{255} \\ \frac{1}{255} & 1 & 1 & \frac{1}{255} \\ 0 & \frac{1}{255} & \frac{1}{255} & 0 \end{matrix}$$

Using this kernel, all types of dead ends will give the result of 6.



255	4	255
255	4	255
255	4	255
255	4	255
255	6	255
255	255	255
255	255	255

Simply count the pixels with value 6 to get the number of dead ends.