COMP2032

Introduction to Image Processing

Coursework -Item 2

Name: Kok Yong En

Word count: 2609words

Video link: https://youtu.be/rdDh-jz4V9k

Contents

Workflow

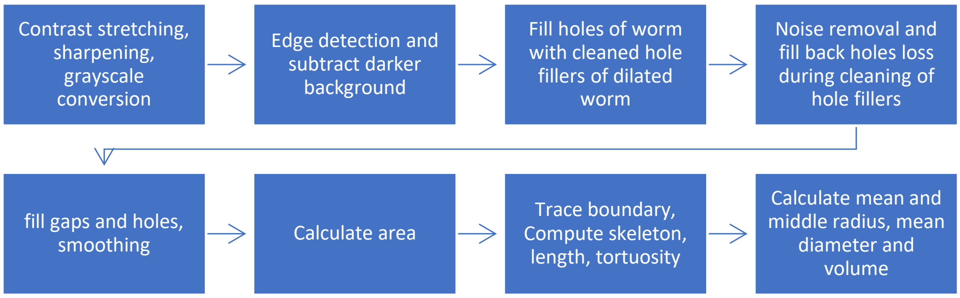


Figure 1: standard workflow of code

1 Contrast stretching, Sharpening, Grayscale Conversion

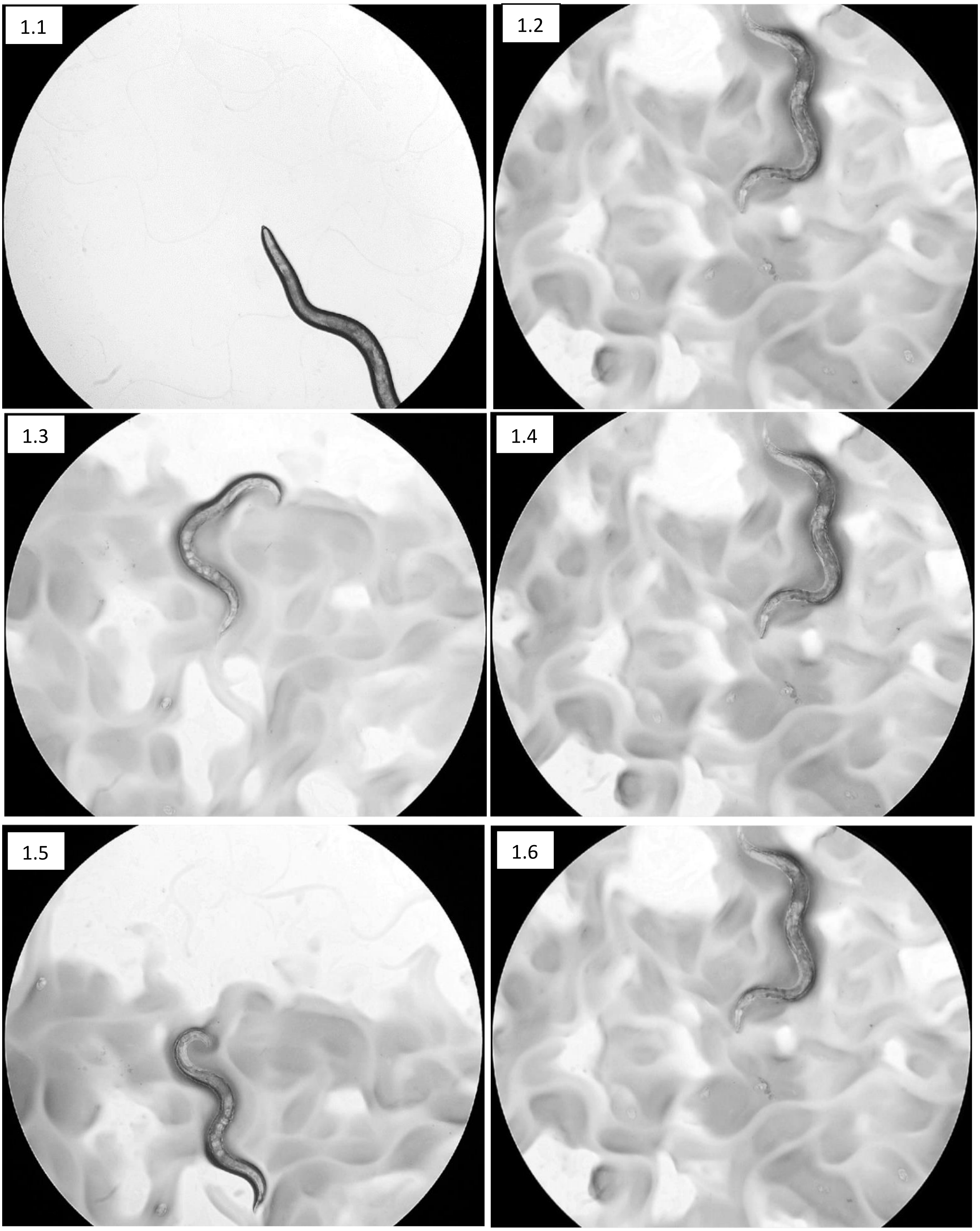
1.1 Explanation and Justification

Intensity values in the original image are clustered and tight. To obtain a more sophisticated worm image, the image is enhanced by contrast stretching to preserve the colour ratio while spreading out intensity values. This is followed by the sharpening of the image with radius of 1.5 and amount of 2 to increase the contrast of the edges pixels that are sharpened. The image is then converted to grayscale for edge detection. Instead of focusing on a specific colour channel, it is sufficient to use the standard rgb2gray for accurate perceived brightness because colours of the foreground and background are analogous.

1.2 Critical evaluation

By using contrast stretching, the intensity level of the background became higher and the middle ground(darker parts surrounding worm) intensity level became lower, allowing us to better differentiate the worm. However, this method also causes part of the worm body to be darken and can blend with the middle ground, specifically Fig 1.1. Another alternative to this method is histogram equalization in which the contrast of image will be increased more dramatically, but the drawback is that it can worsen the blending of darker parts of worm body with the middle ground due to its high contrast.

The standard rgb2gray conversion is effective in perceiving brightness, but some images still do not have a very well-defined foreground and background as the colour of fluid at background is similar to the worm body. To further improve the conversion of colour to grayscale images, Otsu thresholding based on a browness calculation can be considered to differentiate the worm body.


1.1
1.2
1.3
1.4
1.5
1.6

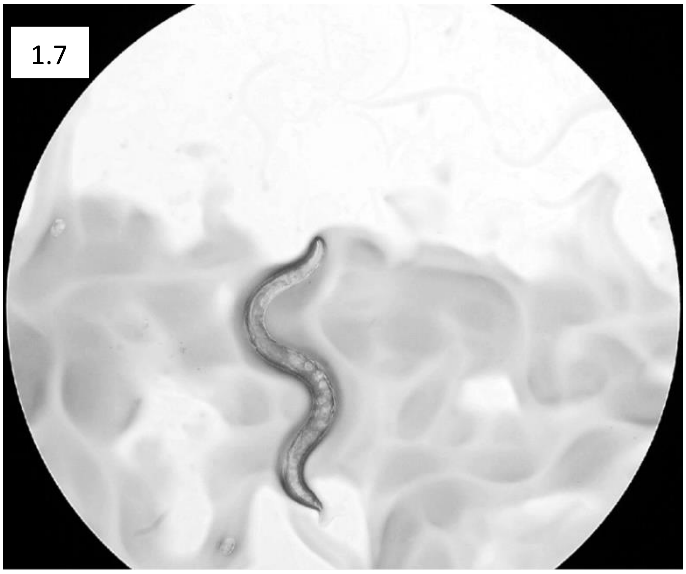


Figure 1: List of grayscale images after contrast stretching and sharpening

2 Edge detection and subtract darker background

2.1 Explanation and Justification

2.1.1 Edge detection

Different edge detection methods will detect edges based on either Laplacian or Gradient based operator. To obtain a more comprehensive binary gradient mask that covers different aspects of the worm body, both Laplacian of Gaussian (LoG) and Prewitt edge detection method are used to produce 2 different binary gradient masks that are later cleaned and combined into 1 comprehensive binary mask. Both operators complement each other as LoG operator is good in detecting sharp edges (Kumar, Saxena, & others, 2013) but can get diffracted by some of existing edges (Shrivakshan & Chandrasekar, 2012) while Prewitt operator can detect simple edges and orientation but cannot produce accurate edge detection with thin, smooth and sharp edges (Priyam & Shreya).

Both LoG and Prewitt operators are used to first calculate the threshold value on the resultant grayscale image. The thresholds are then tuned with fudge factor of 2.2 and 0.9 respectively. Subsequently, LoG and Prewitt edge detection operators are used again onto the grayscale image to produce 2 different binary gradient masks using the tuned thresholds.

2.1.2 Subtract darker background

Unnecessary lines such as the darker background surrounding the worm and the circle of the petri dish can be seen in both binary gradient masks. Thus, the solution is to detect these unnecessary lines and subtract them from the binary masks. Canny edge detection with a threshold of 0.55 is chosen to only detect the very high contrast unnecessary lines, particularly the circle of petri dish (Priyam &

Shreya). The binary gradient mask produced using the LoG and Prewitt operators are then subtracted by the dilated unnecessary lines using disk-shaped structuring element of radius 4 and 1 respectively.

However, noises are still noticeable in the cleaned binary gradient mask using Prewitt operator. To remove the middle ground (darker part surrounding worms), the dark channel is calculated:

$$Darkness = (GreenChannel/2 + BlueChannel/2)$$

The calculated dark channel lowers the intensity level for the middle ground, making most of the images to have a more well-defined middle ground and worm body, Otsu thresholding becomes a suitable method to calculate the threshold based on the dark channel that is then used to binarize the original grayscale image to produce a binary mask of middle ground(BMG). The BMG is then filtered to remove objects smaller than 10 pixels. Subsequently, it is morphologically closed with disk-shaped structuring element of radius 4 and then objects smaller than 130 pixels is removed. These series of steps to clean the BMG is performed to ensure only clearly visible middle ground is subtracted from the cleaned binary gradient mask produced using Prewitt operator.

2.1.3 Combination

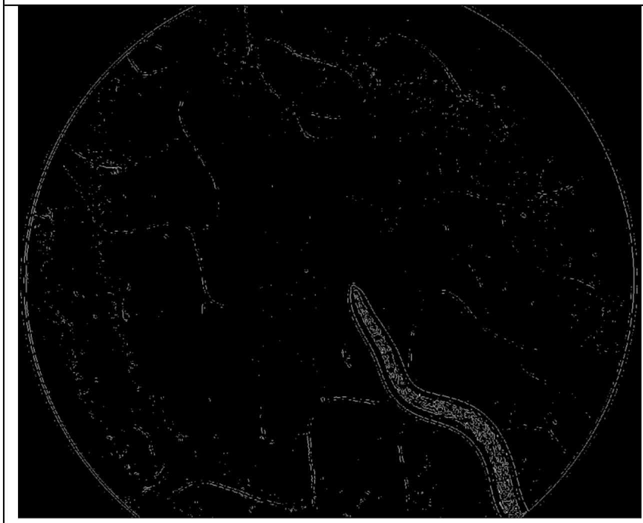
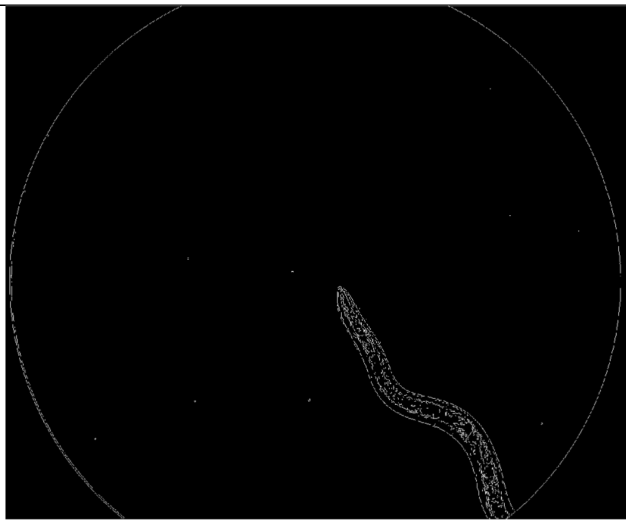
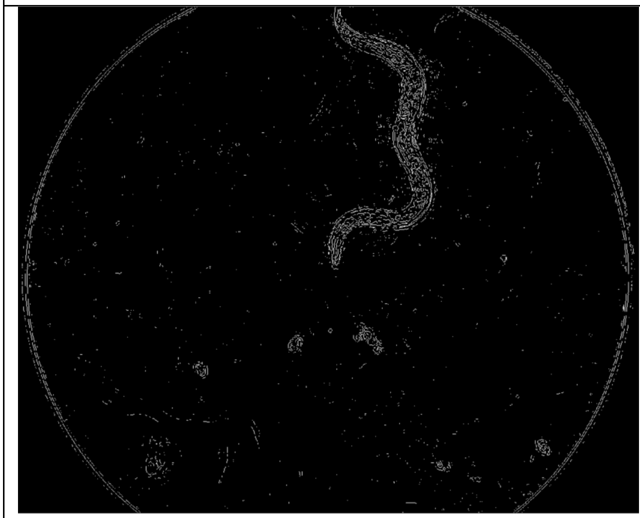
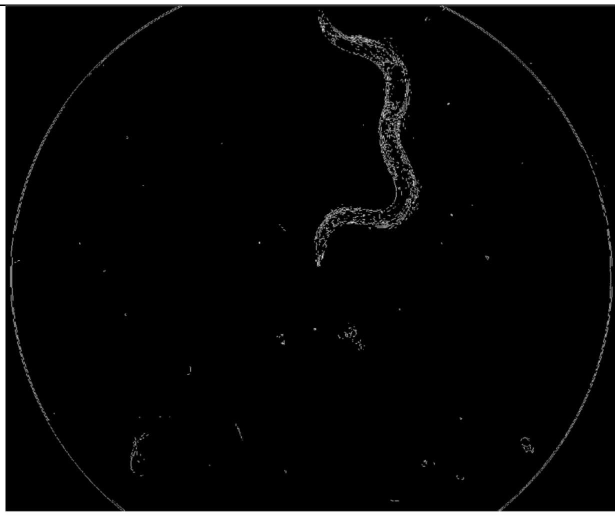
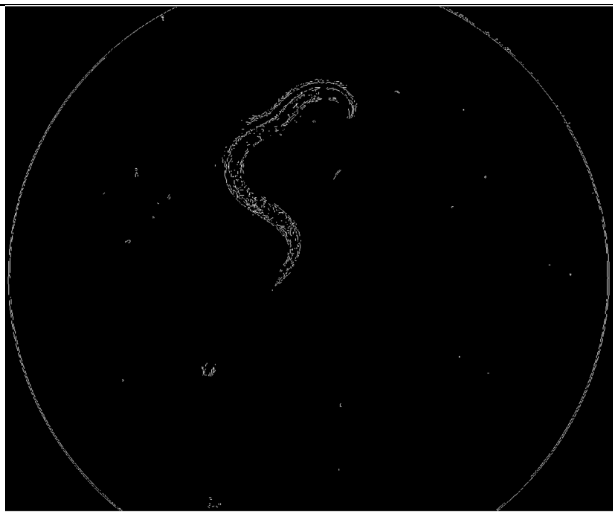
Both cleaned binary masks are then combined to form the final comprehensive binary mask (refer section 2.3).

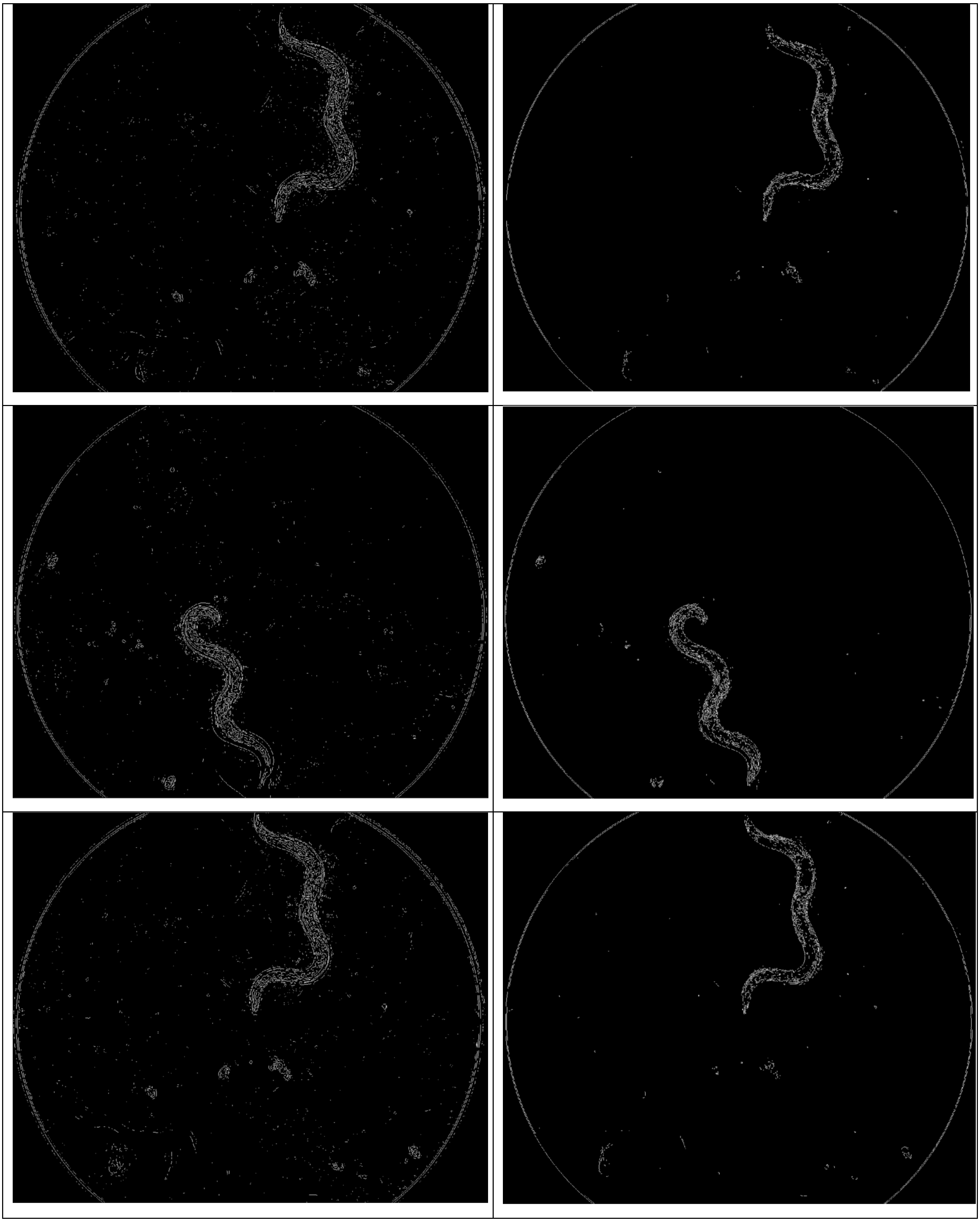
2.2 Critical Analysis

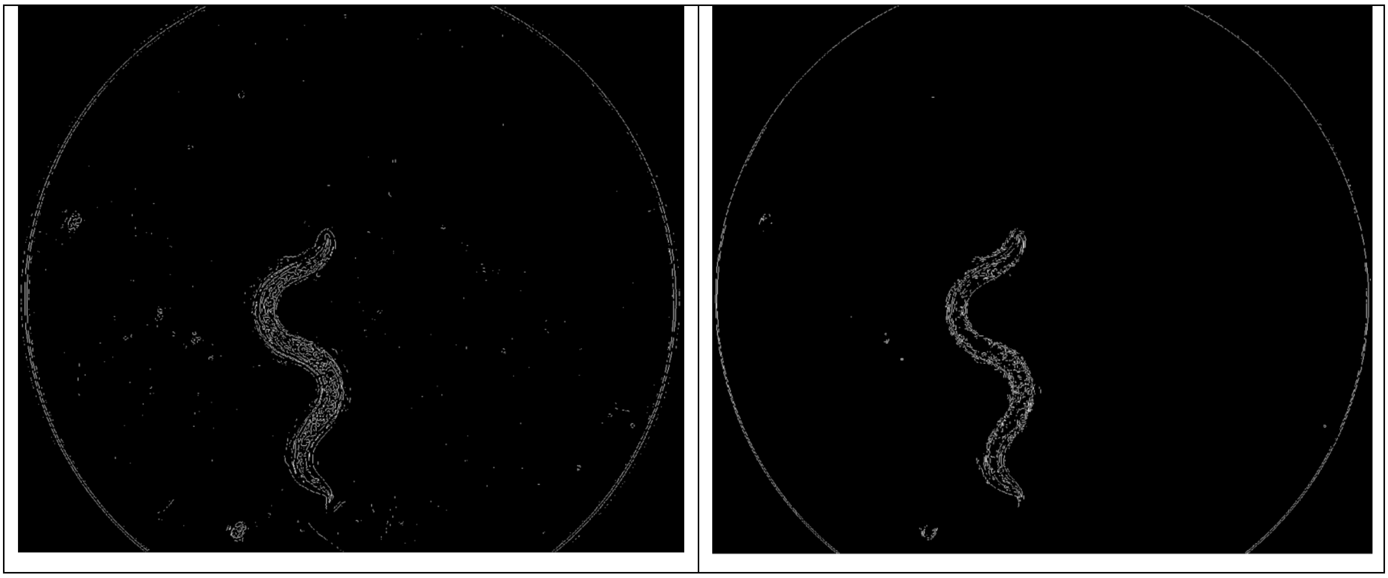
It can be noticed that the edge detection operators were successful in complementing their respective weakness, allowing the worm boundaries to be almost fully covered(refer section 2.3). Nevertheless, the combination of edge operators can introduce more noise to the image (Table 2). Therefore, the subsequent step of subtracting the darker background is employed before the combination to remove the noise attached to the worm. The first subtraction using Canny edge detection is effective in noise removal of the binary gradient mask produced using LoG method because a higher radius of 4 is used. A smaller radius of 1 is used for the binary gradient mask produced using Prewitt method(BPREW) because the edges detected by it is not smooth as compared to that using LoG operator and we might lose some information if a higher radius is used.

Hence, the second subtraction of middle ground is performed again on BPREW before the final combination. These series of subtraction have a very significant impact in removing the noise attachments on worm, especially in Fig 2.1 and 2.5. However, the subtraction method still has some limitations in removing the middle ground for cases where the contrast is not very high at the endpoints of worm(eg. Fig 2.3 and 2.7). To improve the subtraction method, a better darkness equation needs to be formulated so that all the darker part of the image can be taken into account to remove the noises surrounding worm.

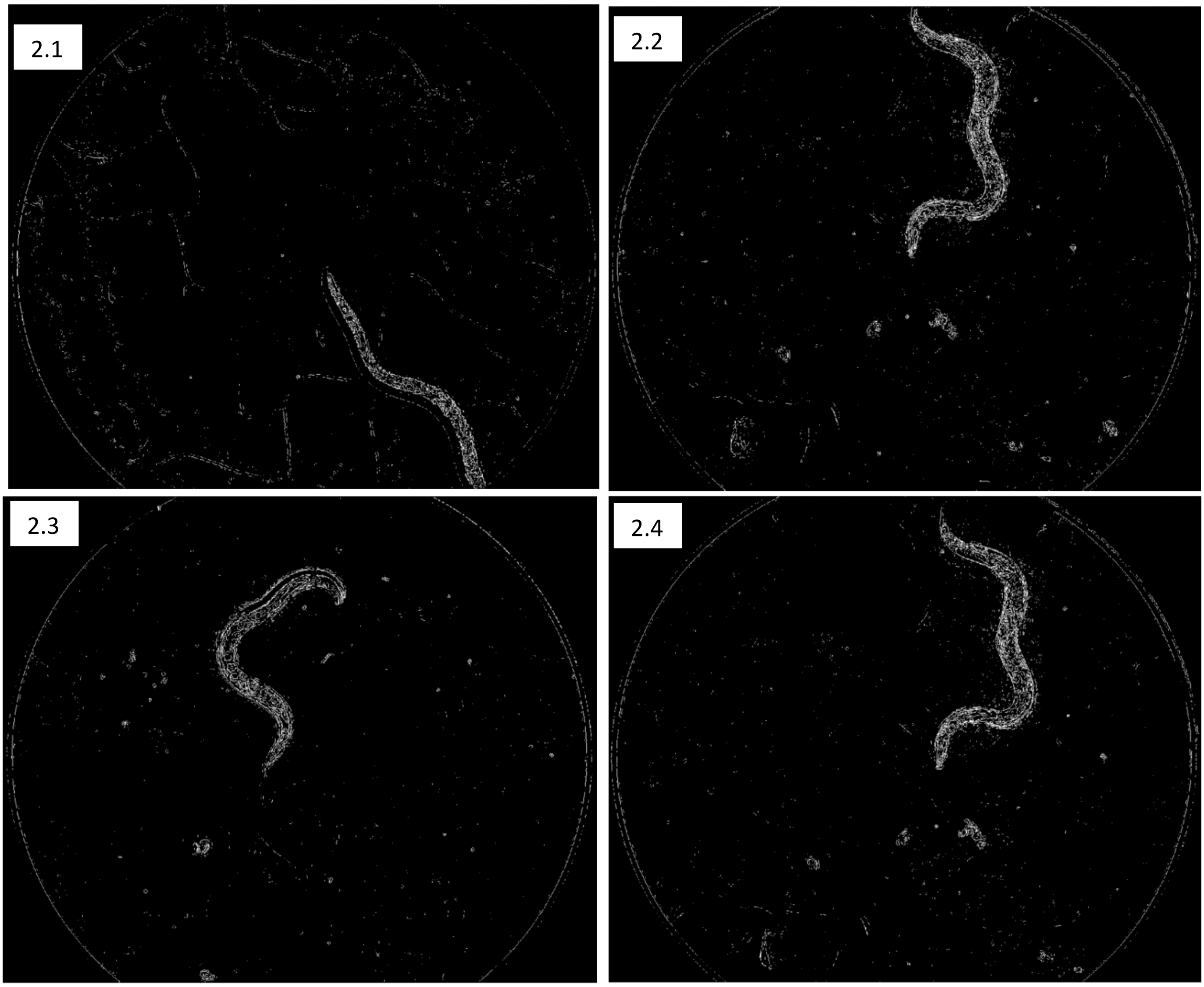
Table 2: Uncleaned edge detection using LoG and Prewitt

LoG detection(Uncleaned)	Prewitt detection(Uncleaned)
	
	
	





2.3 Final comprehensive binary mask



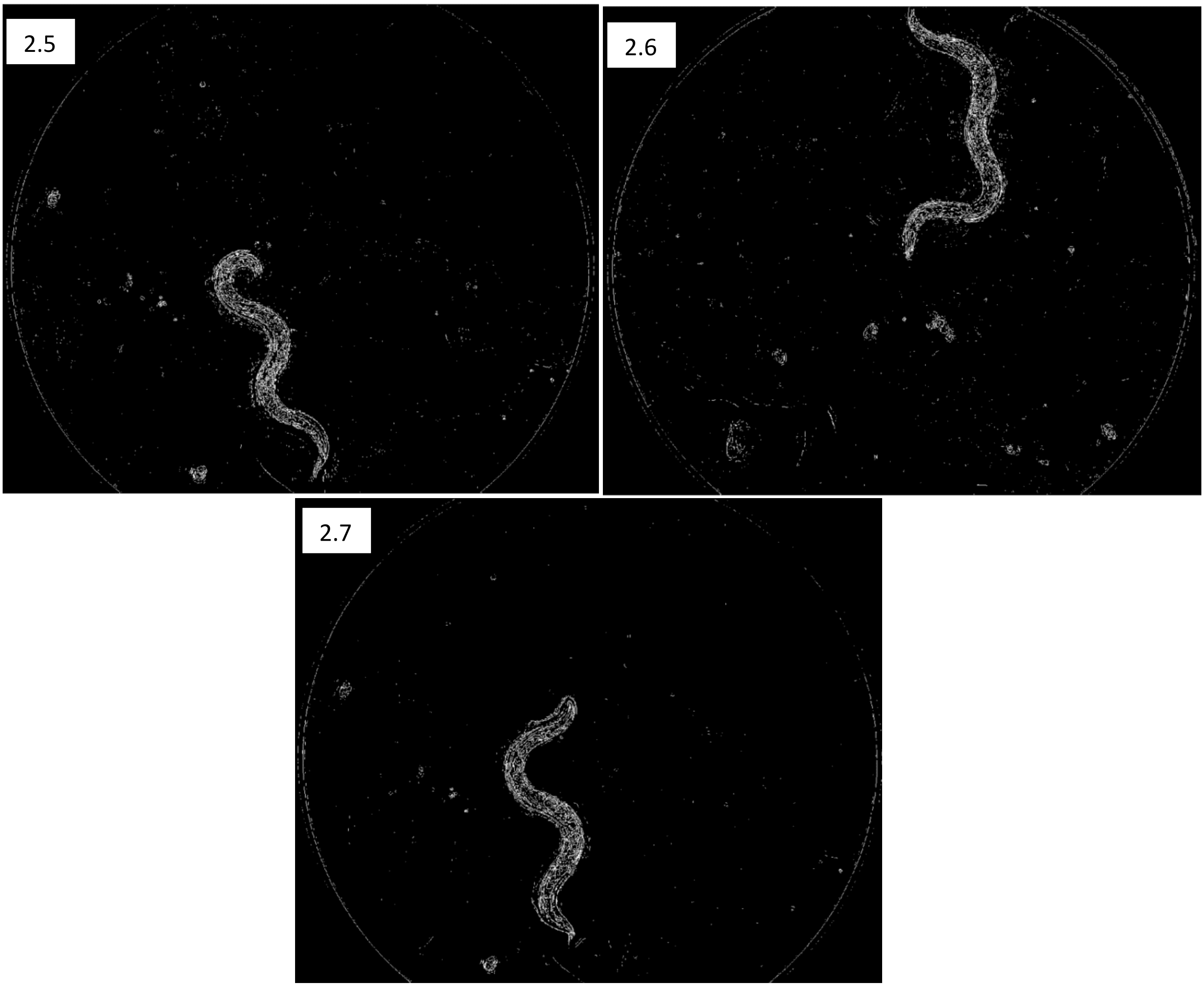


Figure 2: List of images after edge detection and subtraction of darker background (Final comprehensive binary mask)

3 Fill holes of worm with cleaned hole fillers of dilated worm

3.1 Explanation and Justification

In item/version 1, the gradient mask(BW) is dilated using 2 perpendicular linear structuring elements of length 2 which are the vertical structuring element (angle 90) followed by the horizontal one (angle 0) ->DilatedWorm. Then we use imfilll to fill the holes ->FilledDilatedWorm. Although dilating the worm will allow us to 'fully' filled the worm, but it will also increase the strength of the noise, causing noise attachments to the worm as depicted in the figures of version 1 in Table 3.

To fill holes without increasing the strength of noise, we get the hole fillers by finding the difference between FilledDilatedWorm and DilatedWorm. Then, we clean the hole fillers by removing objects lesser than 2 pixels to reduce the noise that could increase attachments on worm. Finally, we dilate the cleaned hole fillers with disk-shaped structuring element of radius 1 and combine it with BW resulting from the previous stage.

3.2 Critical Analysis

This method is effective in filling the holes of worm without introducing much noise to the image. Nonetheless, there are still gaps between edges as they are not dilated as in version 1. Aside from that, Version 1 and 2 started to become significantly different from each other at this point as the cleaned hole fillers are successful in breaking down the connection of the noise surrounding the worm (Table 3), allowing us to obtain a smoother worm body at stage 6 of this workflow(Table 6). For instance, this significant change can be observed from Fig 3 in which the connection with the middle ground noise is completely break off in version 2. But the cleaning can cause some information loss because some of the holes might not be filled for cases like Table 3.3.2. This was taken and accepted as a drawback for improving the quality of the majority cases.

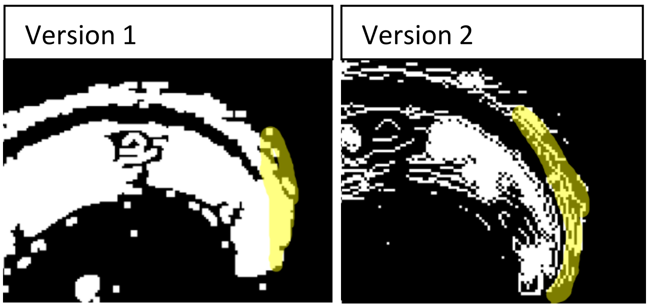
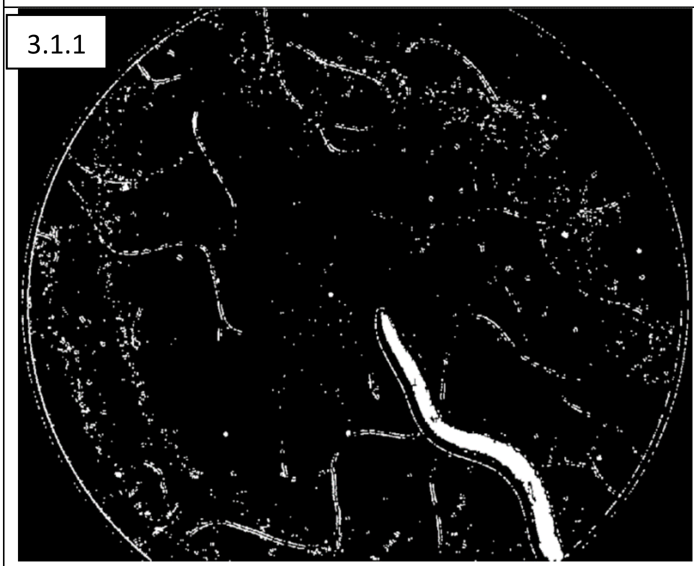
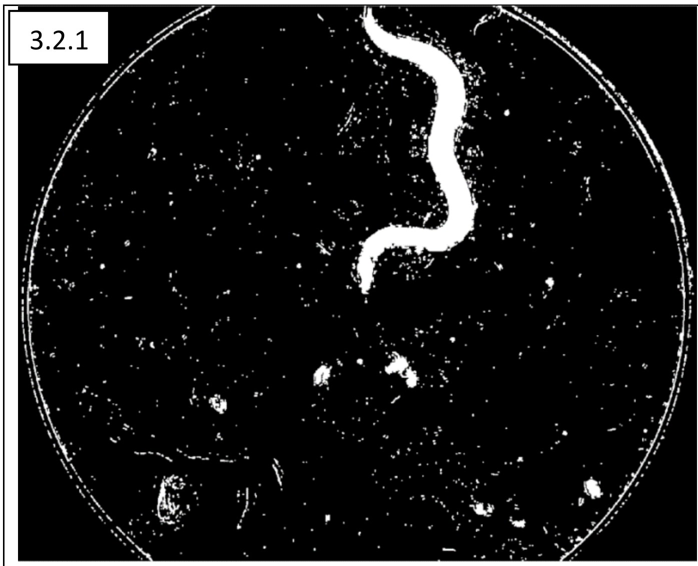


Figure 3: Zoom view of image numbered 3

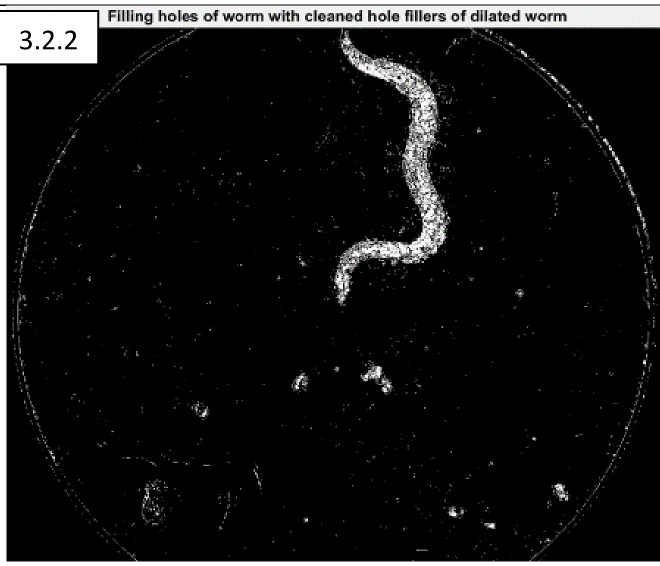
Table 3:Table of comparision between version 1 and 2 on filling the holes of worm

Version 1	Version 2
3.1.1 	3.1.2 

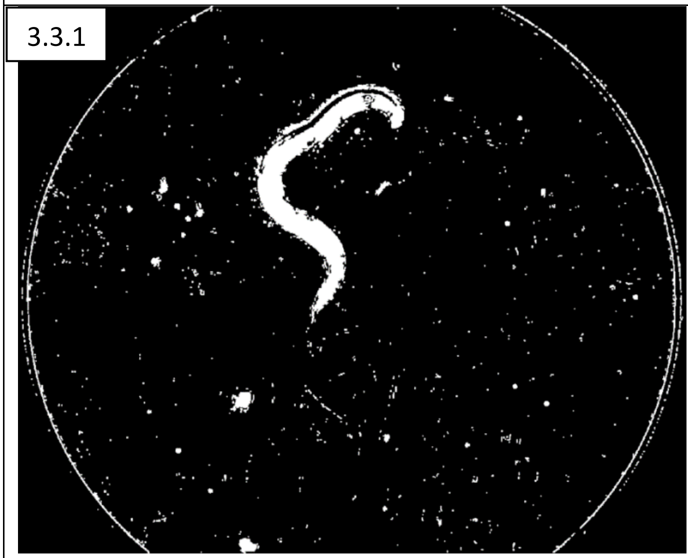
3.2.1



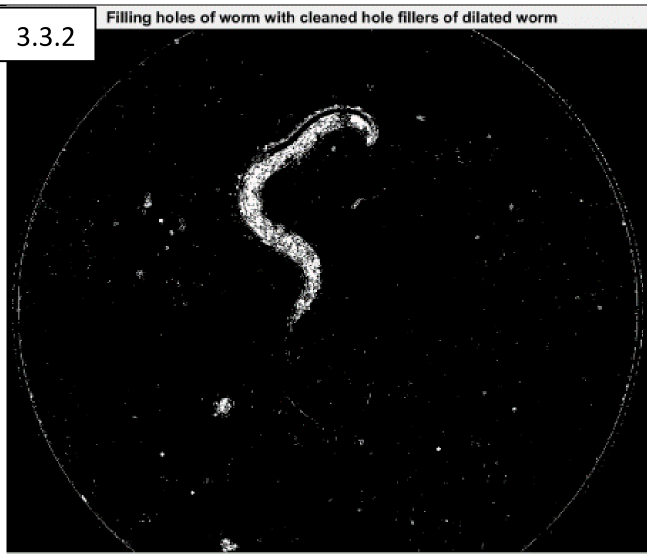
3.2.2



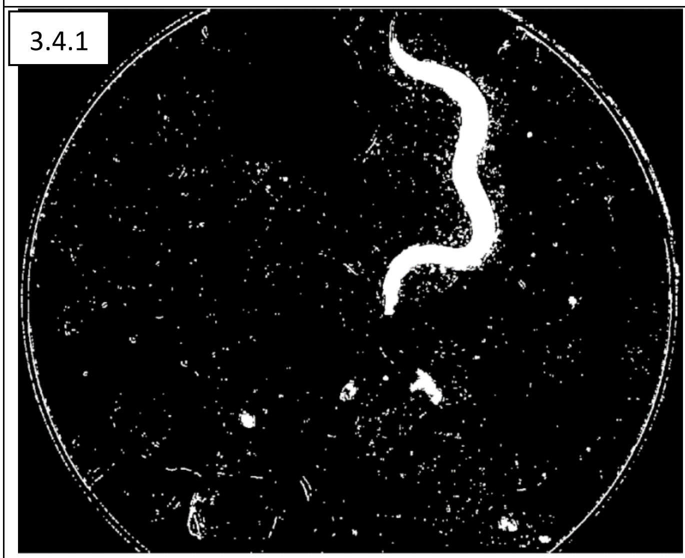
3.3.1



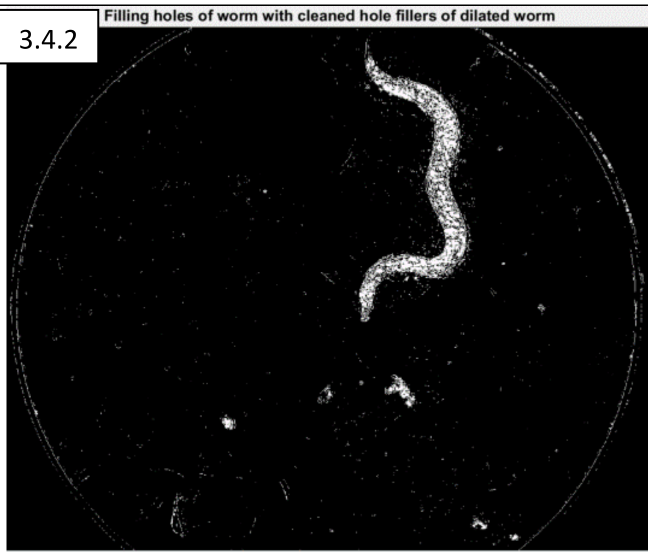
3.3.2

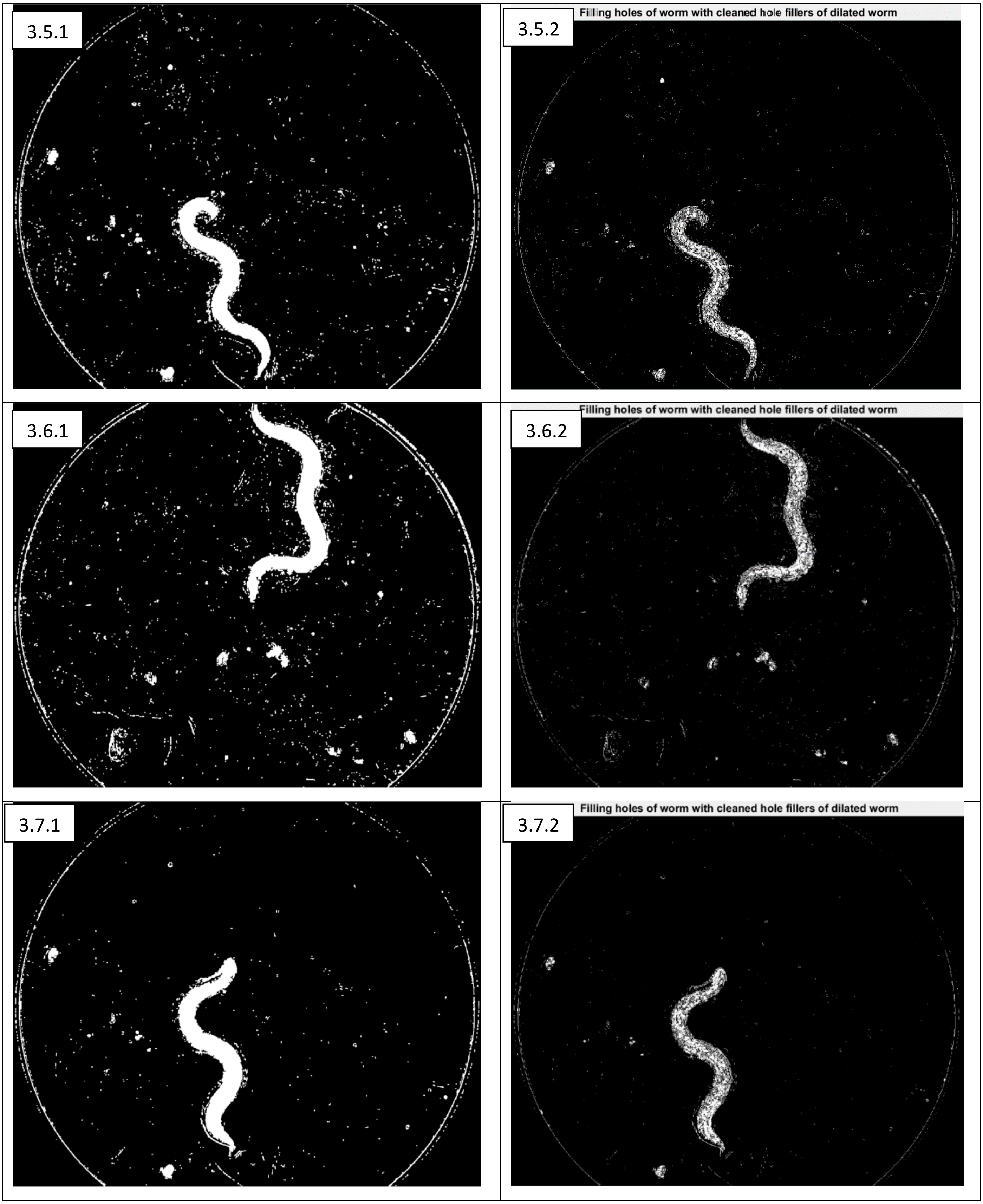


3.4.1



3.4.2




3.5.1
3.5.2
Filling holes of worm with cleaned hole fillers of dilated worm
3.6.1
3.6.2
Filling holes of worm with cleaned hole fillers of dilated worm
3.7.1
3.7.2
Filling holes of worm with cleaned hole fillers of dilated worm

4. Noise removal and fill back holes loss during cleaning of hole fillers

4.1 Explanation and Justification

To segment the worm from all noises, bwareafilt is used to retain only 1 largest blob in image. Now that it is cleaned, we combine the same dilated hole fillers from previous stage(but without going through cleaning of objects less than 2 pixels) with the segmented worm as we might have loss some information previously while clearing it. Again, bwareafilt is used to segment the largest blob in image to remove the unconnected hole fillers resulting from the combination step. Only 1 largest blob is retained because this program is limited to evaluation of 1 worm in image.

4.2 Critical Analysis

The retain method is capable in retaining the vital part of the worm and clearing all the noises but it might not succeed in other cases where the vital components fail to connect with the main body at stage 3. Another solution is to retain objects in a specified range to allow some vital but unconnected components to remain in image but this method will fail to remove the noises that fall in between the specified range. Moreover, it is hard to specify a range as the size of the vital but unconnected part can vary.

The combination with the dilated hole fillers is effective in bringing back the information loss during cleaning in stage 3 but it may introduce a little noise that can be later resolved by the retain method.

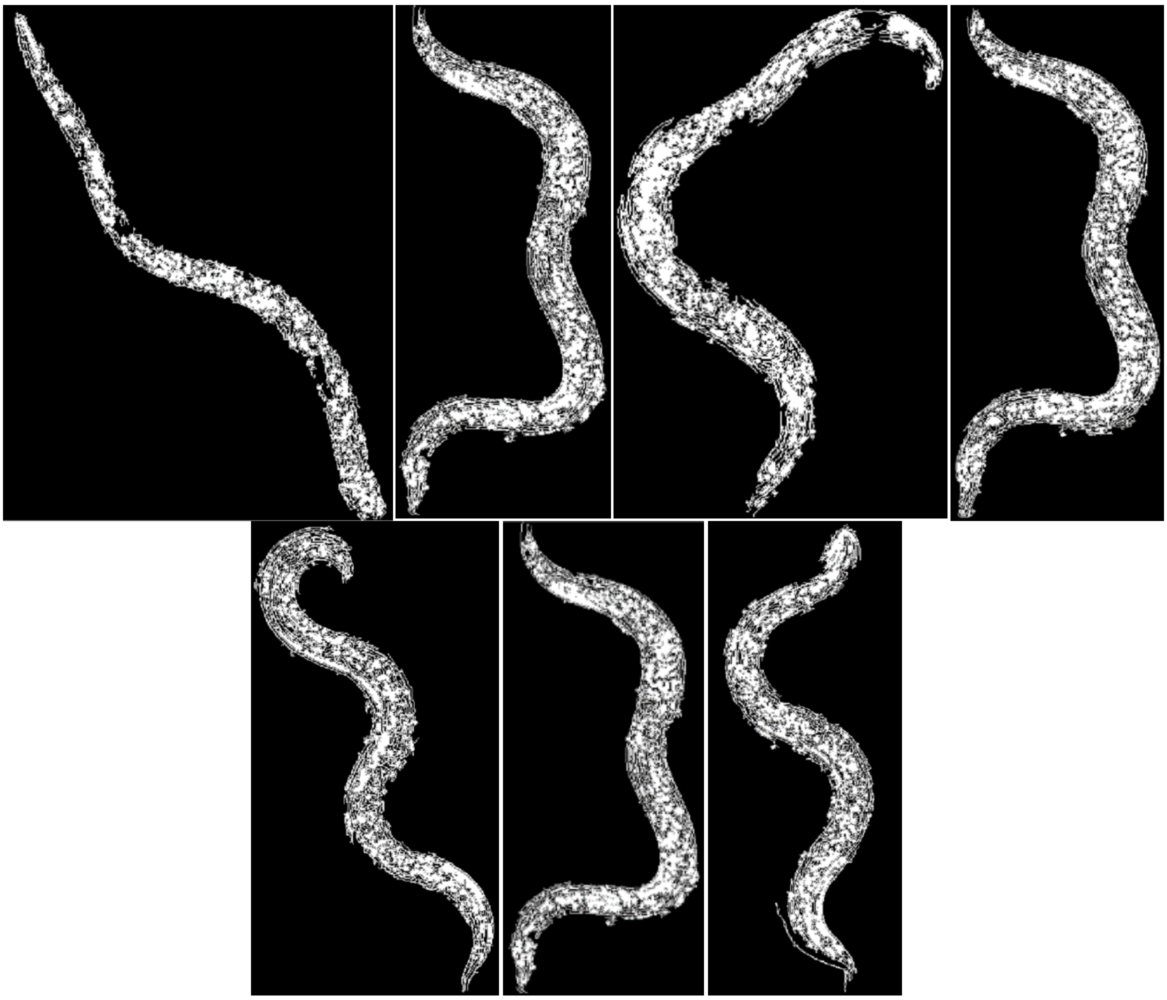


Figure 4: List of images after noise removal and fill back holes loss during cleaning of hole fillers

5 Fill gaps and holes, smoothing

5.1 Explanation and Justification

To reconnect the gaps in worm, the worm is dilated using the previous 2 perpendicular linear structuring elements. Then, the worm is morphologically closed with a disk-shaped structuring element of radius 6 to completely fill all the holes and gaps and then opened again with radius of 7 to smoothen the worm. Finally, to further smoothen the worm, the worm is eroded with disk-shaped structuring element of radius 2.

5.2 Critical Analysis

The close operation has successfully reconnected all gaps and fill all the holes, more information is preserved as compared to morphologically open it first. The opening and erosion operations are then performed and most of the small noise attachments are removed. But it is noticeable that there are still little spots attaching to the body. An alternative solution to this is to further increase the radius for the opening operation to completely remove the spots but the drawback is the thinner parts(head/tail) will be lost when the erosion during the opening process is too high.

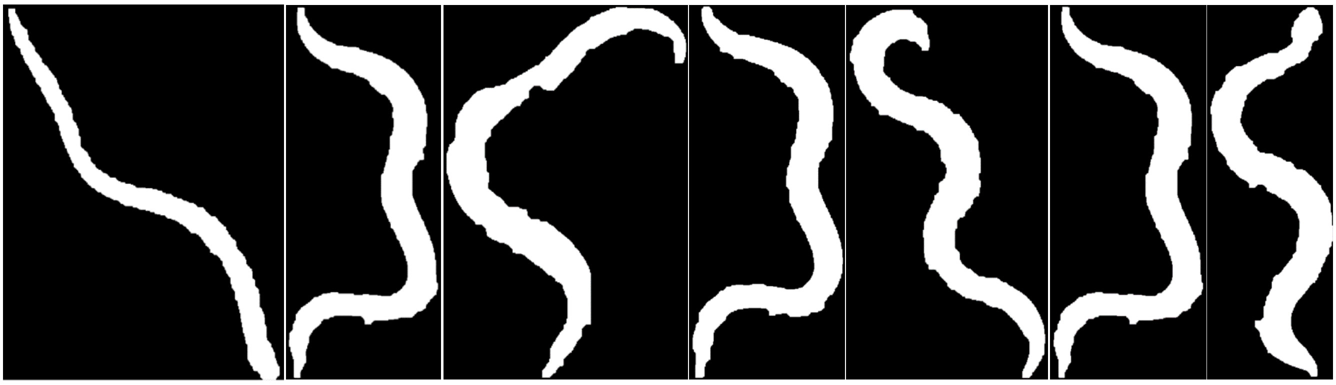


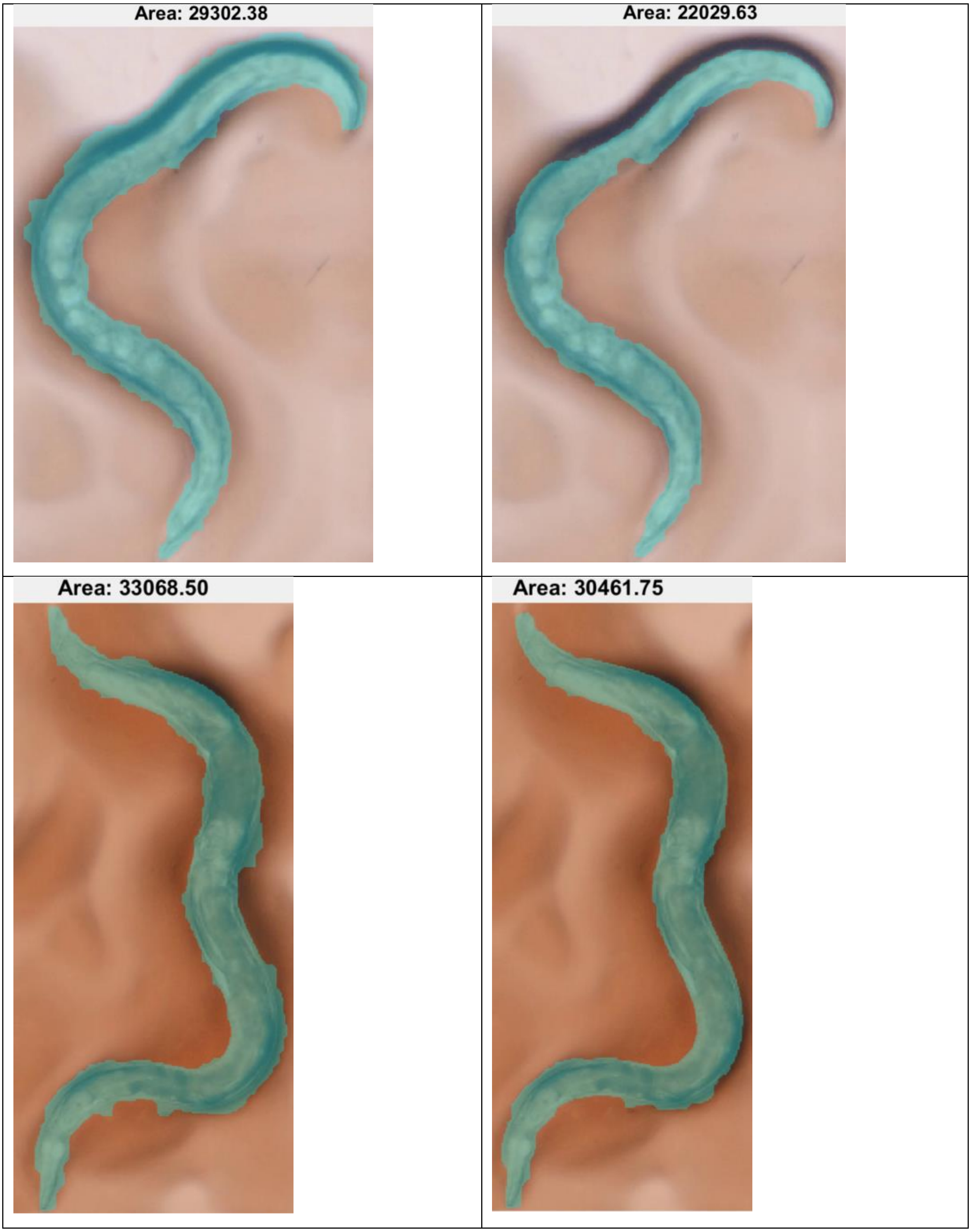
Figure 5: List of images after filling gaps and holes, smoothing

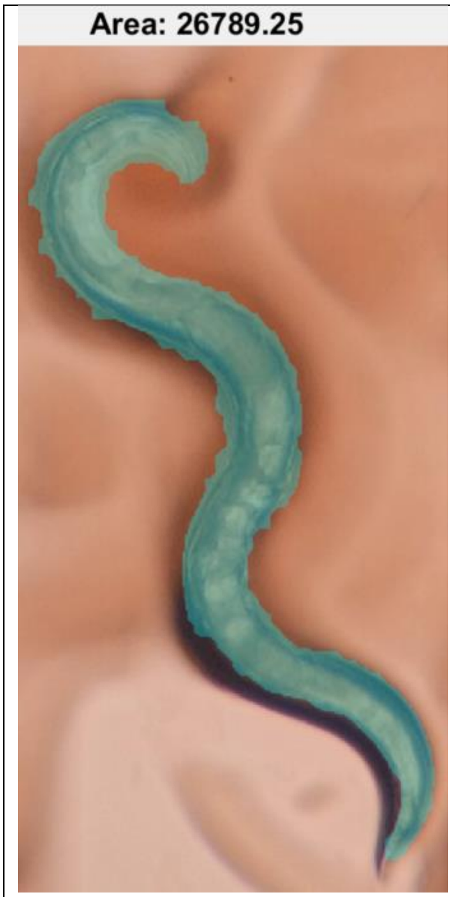
6 Calculate area

bwarea is used to estimate the area of worm in image. The unit corresponds roughly to the total number of pixels in the image as it estimates the area based on weighted pattern on pixels.

Table 6: Table of comparison between version 1 and 2 on the area coverage

Version 1	Version 2
	
	

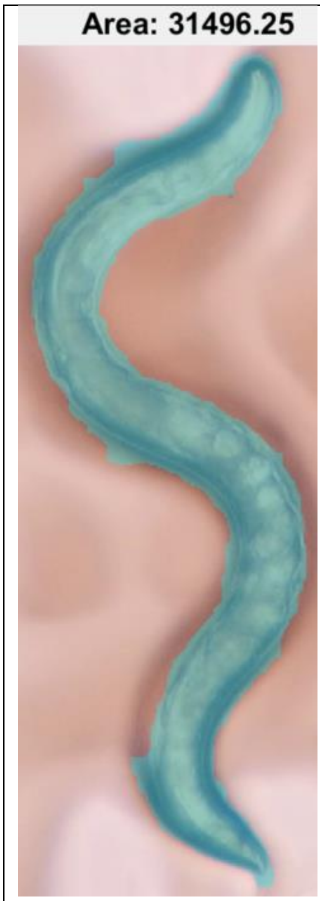

Area: 29302.38
Area: 22029.63
Area: 33068.50
Area: 30461.75


Area: 26789.25

Area: 25573.75

Area: 33281.38

Area: 30719.00



7 Trace boundary, Compute skeleton, length, tortuosity

7.1 Explanation and justification for tracing boundary, computing skeleton and length

To highlight the boundary of worm, Canny edge detection method is used to trace the edge of segmented worm. The highlighted outline is then placed as an overlay on the original image for comparison.

To skeletonise the worm, bwskel is used to extract the centerline (FirstSkelImage), but there might be short spurs attached to the centerline if the structure of worm is still not smooth enough. Hence, we estimated the minimum branch length as follows:

$$miniumum\ branch\ length\ \approx\ (\sum FirstSkelImage)/3$$

Then, bwskel is used again to prune all spurs shorter than the minimum branch length. The skeletonize worm is produced and spine length is calculated by summing all pixels in it.

7.2 Critical evaluation of computing skeleton

Since the worm body is not smooth enough, there will be cases like Table 7.7.2 in which the centerline cannot reach the true end of head/tail. An alternative solution is to find out the nearest boundary coordinates to the endpoints and then connect them with a line. But it would result in connecting the incorrect boundary coordinates if the centerline is far away from the true end. This solution will only be suitable for centerline that is already at the near end of boundary.

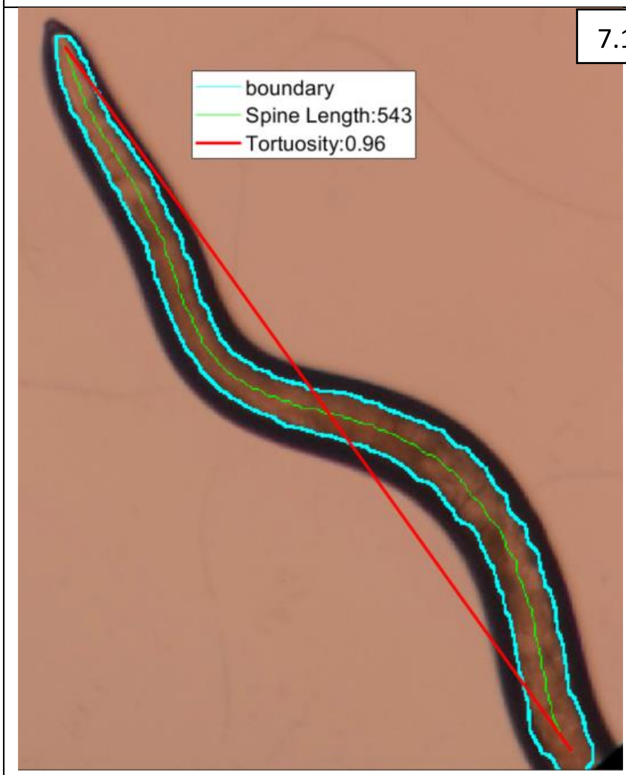
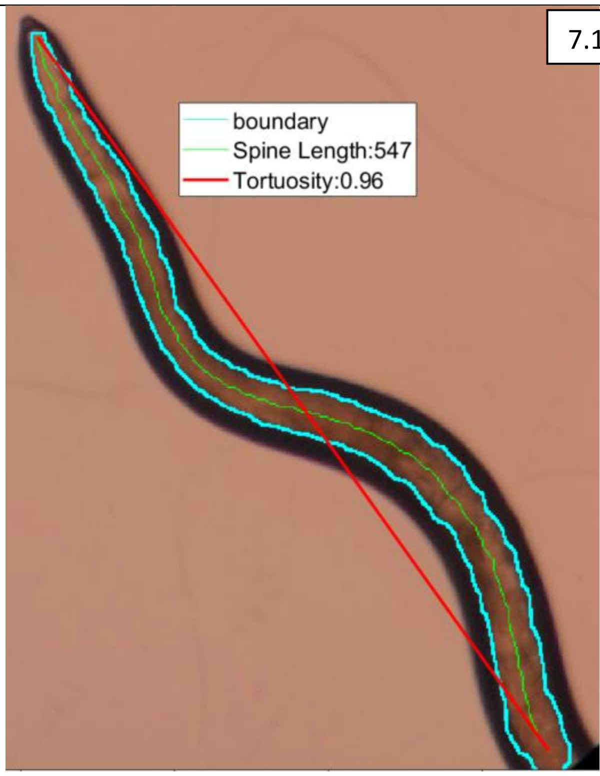
7.3 Explanation and justification for computing tortuosity

To compute tortuosity, we need to first calculate the straight-line distance of worm. On this account, we use bwmorph to find the endpoints(x_1, y_1) (x_2, y_2) of the skeleton image. The Euler distance between the endpoints is the straight-line distance of worm:

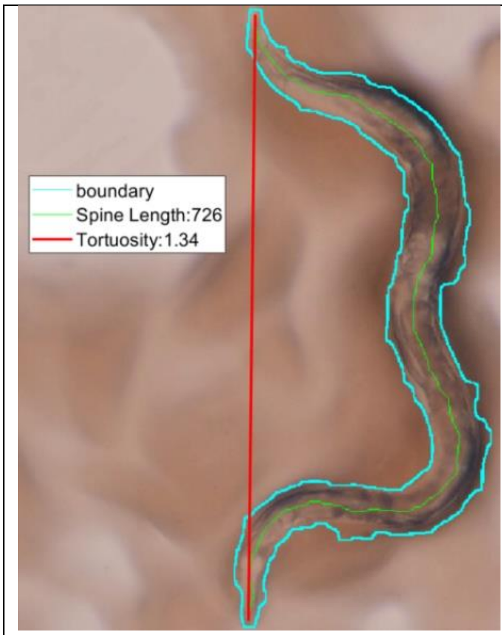
$$straightLineDistance = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$$

$$tortuosity = spineLength/straightLineDistance$$

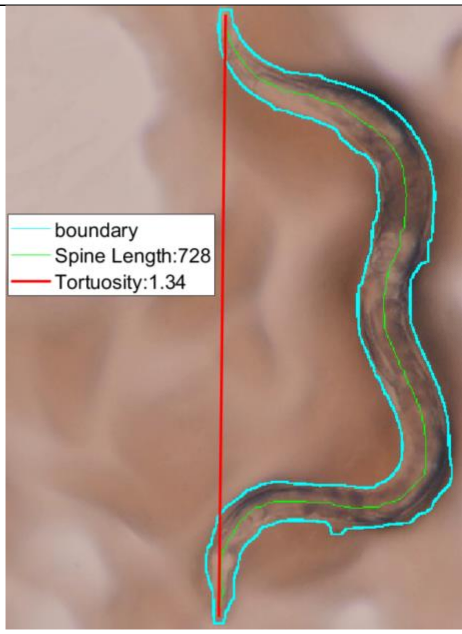
Table 7: Table of comparison between version 1 and 2 on tracing boundary, computing skeleton, length and tortuosity

Version 1	Version 2
7.1.1 	7.1.2 

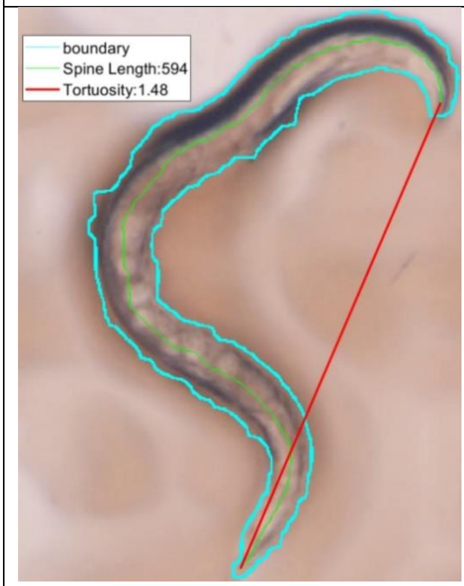
7.2.1



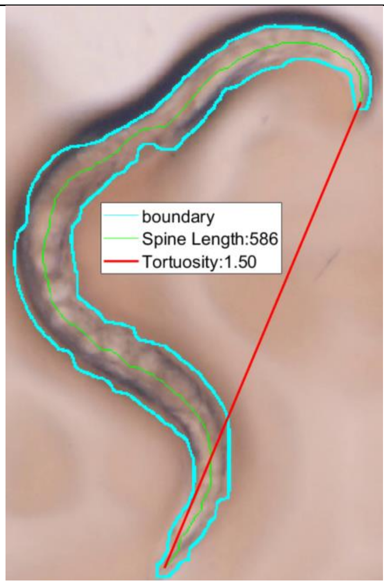
7.2.2



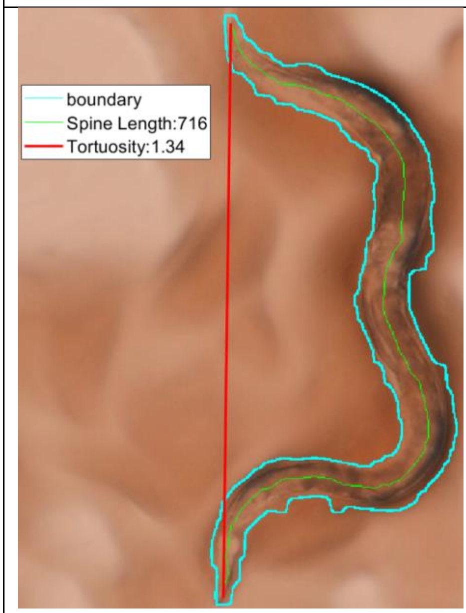
7.3.1



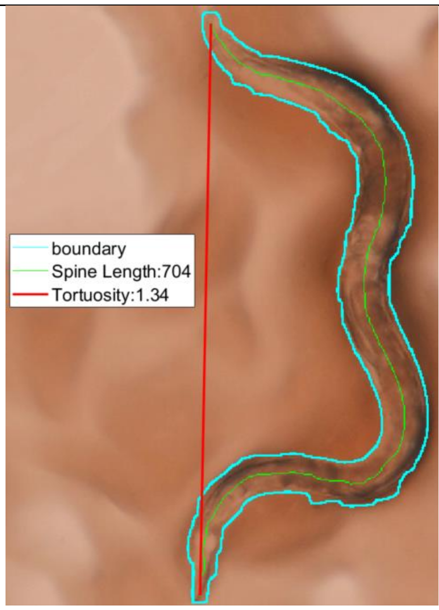
7.3.2



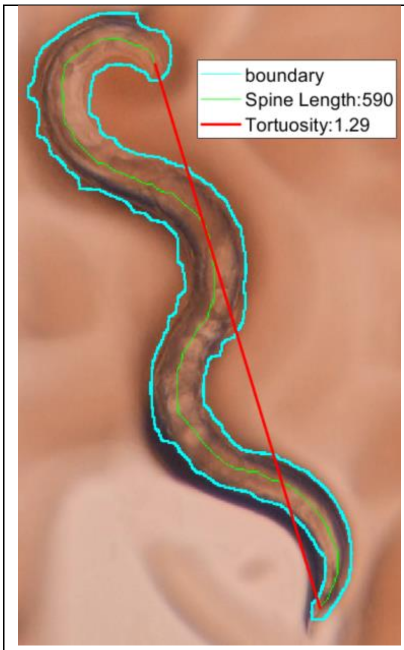
7.4.1



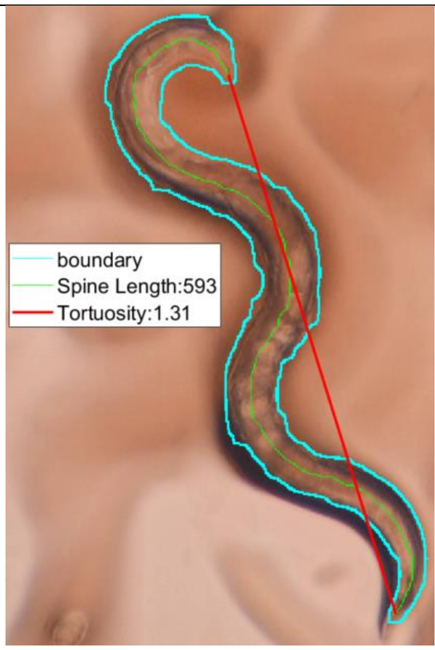
7.4.2



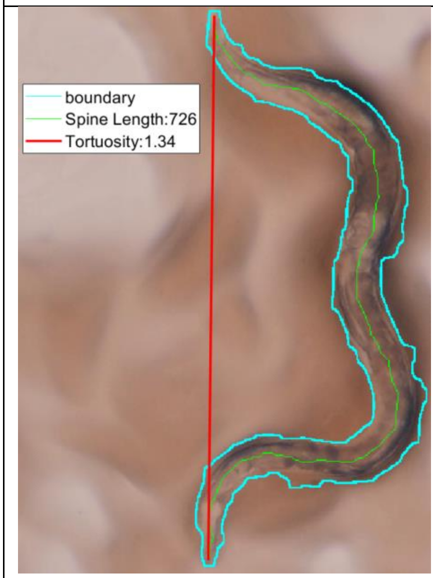
7.5.1



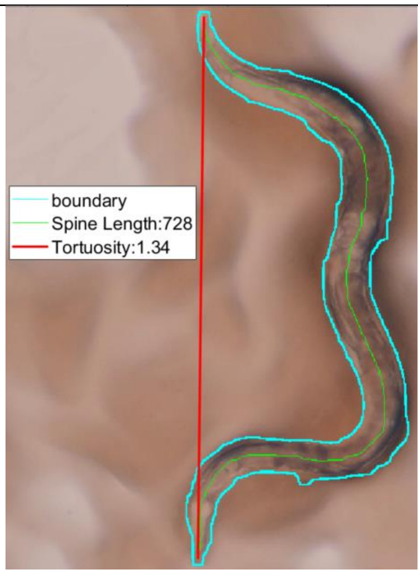
7.5.2



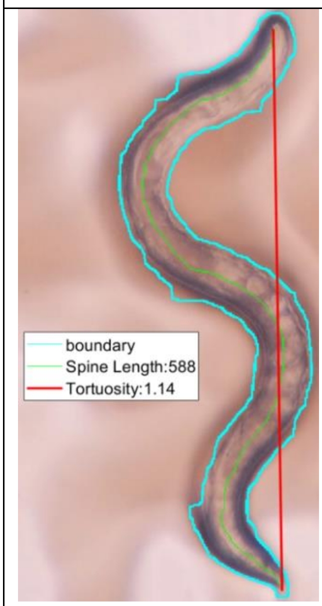
7.6.1



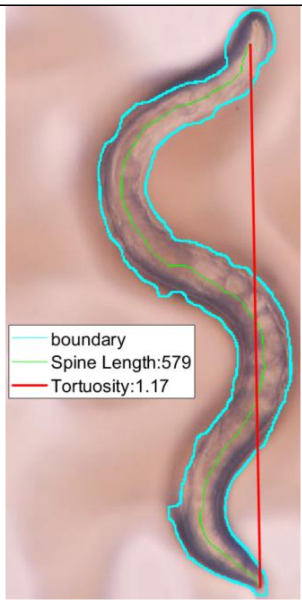
7.6.2



7.7.1



7.7.2



8 Calculate mean and middle radius, mean diameter and volume

8.1 Mean and middle radius, mean diameter

The mean radius of worm is calculated by first getting the Euclidean distance transform of the binary segmented worm. Then, mean radius is computed by looking along the skeleton of the distance transform. While the mean diameter is the double of mean radius. (refer Figure 8.1)

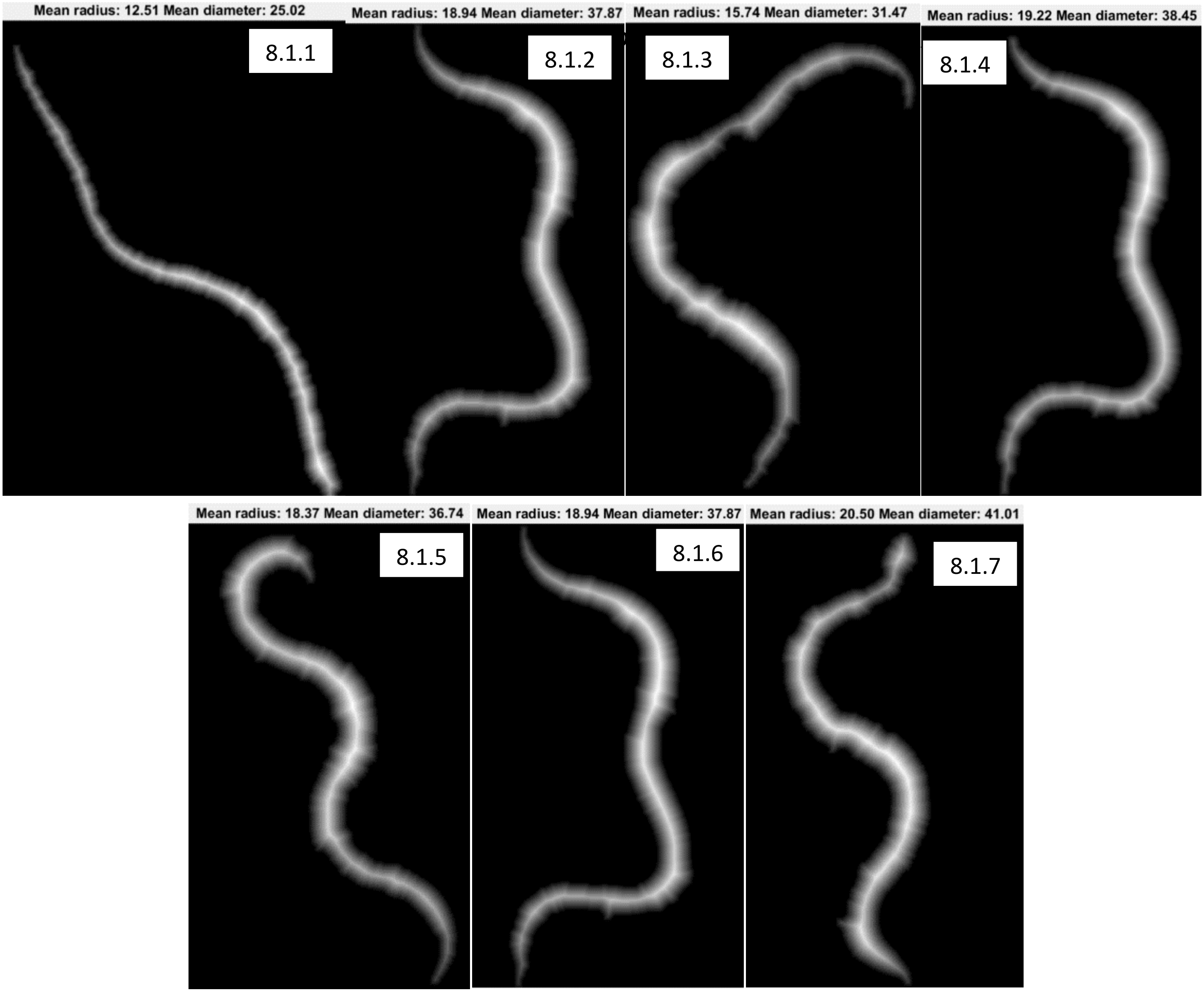


Figure 8.1: List of Euclidean Distance Transform images

To find the middle radius and volume, we must first calculate the distance between skeleton and worm boundary by determining the coordinates of skeletonize worm and of worm boundary. The coordinates of skeletonize worm are computed using bwtraceboundary and the start point is the endpoint (x_1, y_1) we calculated previously. The function then returns the coordinates by traversing the spine from the starting point. On the other side, the worm boundary coordinates are computed using bwboundaries because the order of the coordinates is not essential for the calculation of radius.

The distances from the coordinate (x_m, y_m) at the middle of skeletonize worm to all other points(bx,by) of boundaries are calculated and the closest distance will be the middle radius:

$$middle\ radius_{(in\ pixels)} = min(\sqrt{(by - y_m)^2 + (bx - x_m)^2})$$

The closest distance is taken as the radius to ensure better overall accuracy because error due to the uneven worm body can occur. Furthermore, it is more efficient to find the closest distance rather than trying to formulate a strategy to find an unknown point that cannot guarantee its accuracy.

8.2 Volume

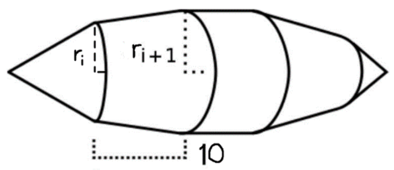
If we assume the worm is round, then its volume is equivalent to the 3D integral of its contour rotated around its skeleton. The volume is computed by sampling radius at intervals of a fixed size (10 pixels) along the skeleton, treating each segment as a frustum of a cone, and summing the volume of these frustums (Moore, Jordan, & Baugh, 2013).

The interval size is set at 10 because error due to the distinct nature of pixel positions can be introduced if the interval size is too small, and if it is too large, the cone segments will be too large to accurately capture the tapering of worm.

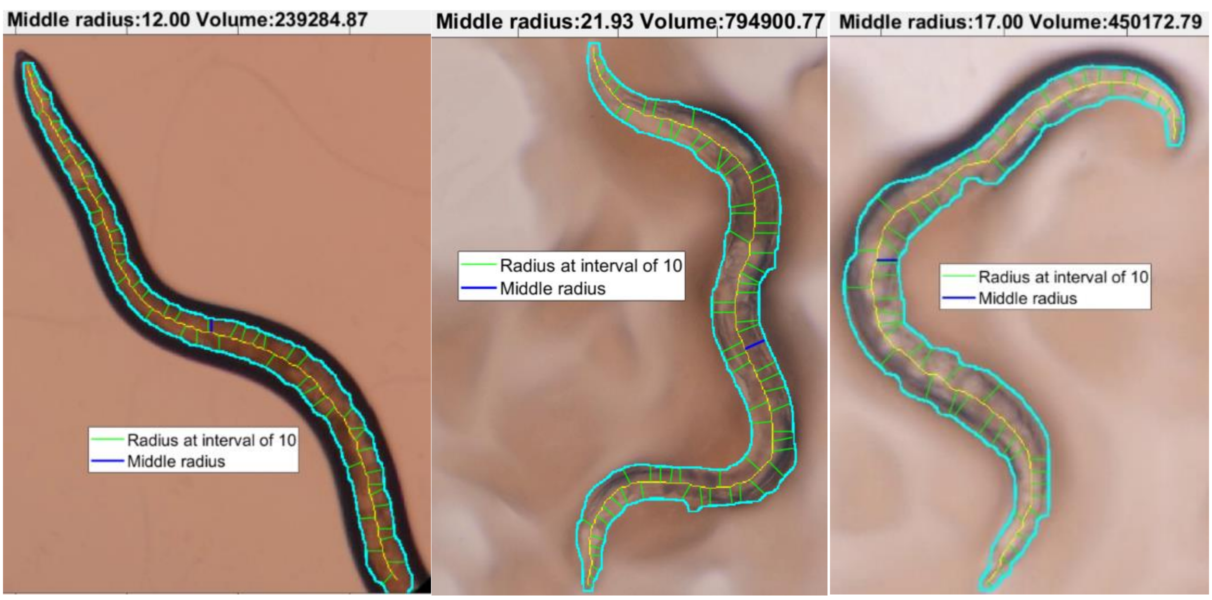
The radii of the worm at interval of 10 pixels:

$$r_{i=1:\lceil spineLength/10 \rceil} = min(\sqrt{(by - y_{10i-9})^2 + (bx - x_{10i-9})^2})$$

The volume of worm is calculated as follows:



$$Volume_{(in\ pixel^3)} = \sum_{i=1}^{len(r)-1} \frac{\pi * 10}{3}(r_i^2 + r_i * r_{i+1} + r_{i+1}^2)$$



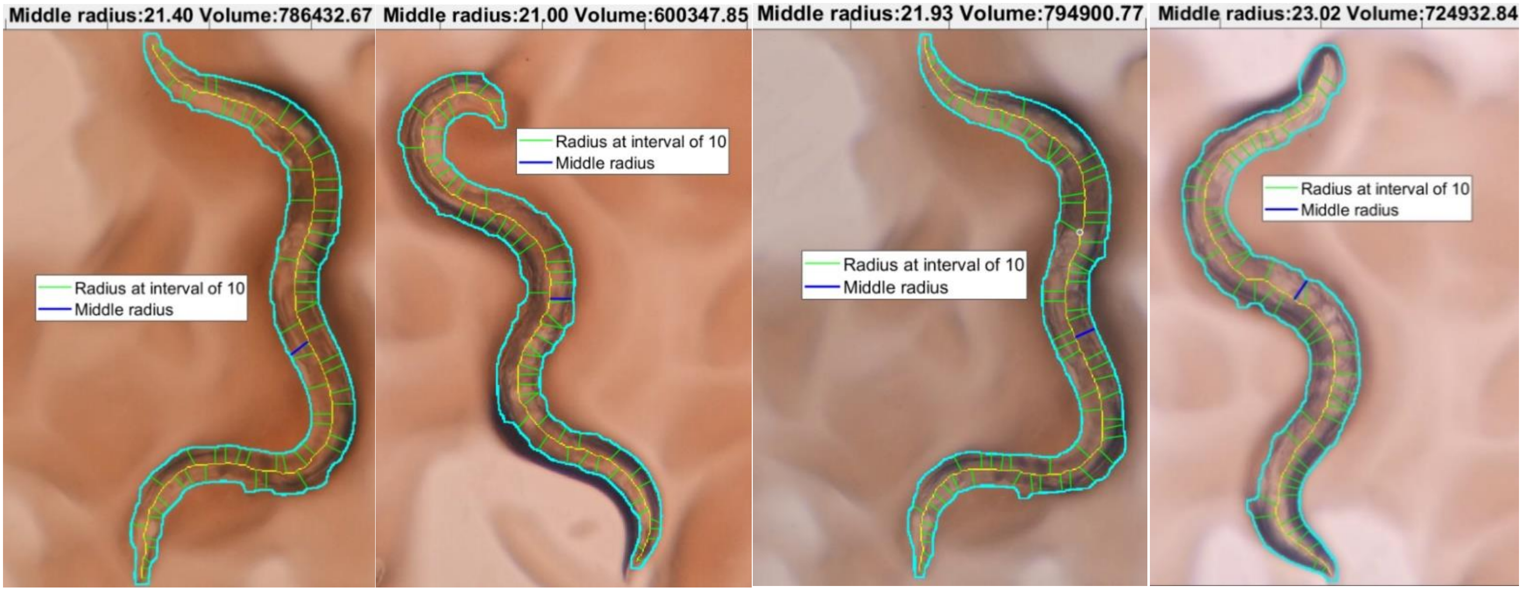


Figure 8.2: List of images showing the boundary, middle radius, spine and radii at interval of 10 pixels

References

Kumar, M., Saxena, R., & others. (2013). Algorithm and technique on various edge detection: A survey. *Signal & Image Processing, 4*, 65.

Moore, B. T., Jordan, J. M., & Baugh, L. R. (2013). WormSizer: high-throughput analysis of nematode size and shape. *PloS one, 8*.

Priyam, D. D., & Shreya, D. P. (n.d.). Edge Detection by Using Canny and Prewitt.

Shrivakshan, G. T., & Chandrasekar, C. (2012). A comparison of various edge detection techniques used in image processing. *International Journal of Computer Science Issues (IJCSI), 9*, 269.