

Digital image processing and vision systems – lab #9

Date performed: 01.06.2021	Group 2
Author name: Krzysztof Klimczyk	

1. Source codes and screenshots:

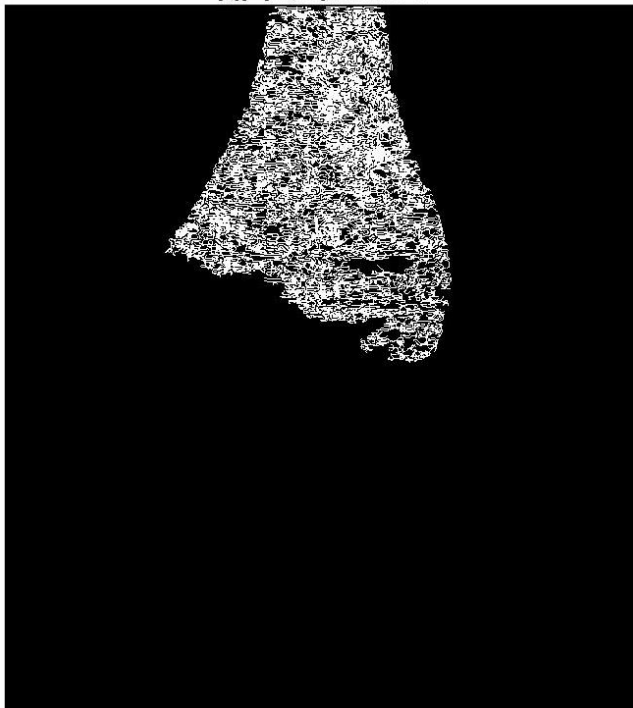
Task 9.4. – Region growing segmentation:

```
1. image = imread("knee.png");
2. subplot(1,2,1)
3. imshow(image);
4. title("Input image");
5. thresh = 4;
6. [s_x,s_y] = ginput(1);
7. s_x = round(s_x);
8. s_y = round(s_y);
9. hold on;
10. plot(s_x,s_y,'d','color',[1 1 0]);
11. image = double(image);
12. [n,m] = size(image);
13. visited = zeros(n,m);
14. segmented = zeros(n,m);
15. fQueue = 1;
16. lQueue = 2;
17. queue = zeros(10000,2);
18.
19. %add to queue
20. queue(fQueue,1)=s_x;
21. queue(fQueue,2)=s_y;
22.
23. %mark as visited and segmented
24. segmented(s_y,s_x)=1;
25. visited(s_y,s_x)=1;
26.
27. while(fQueue < lQueue)
28.     %queue pop
29.     x = queue(fQueue,1);
30.     y = queue(fQueue,2);
31.     fQueue = fQueue + 1;
32.
33.     if x > m-1 || y > m-1 || x < 2 || y < 2
34.         continue
35.     end
36.
37.     for i=-1:1:1
38.         for j=-1:1:1
39.             temp_x = x + i;
40.             temp_y = y + j;
41.             diff = abs(image(y,x) - image(temp_y,temp_x));
42.             if diff < thresh && visited(temp_y,temp_x)==0
43.                 queue(lQueue,1) = temp_x;
44.                 queue(lQueue,2) = temp_y;
45.                 lQueue = lQueue + 1;
46.                 segmented(temp_y,temp_x)=1;
47.             end
48.             %mark as visited
49.             visited(temp_y,temp_x)=1;
50.         end
51.     end
52. end
53.
54. subplot(1,2,2);
55. imshow(segmented);
56. title("[x,y] = [" + s_x + ", " + s_y + "], threshold = " + thresh);
```

Input image



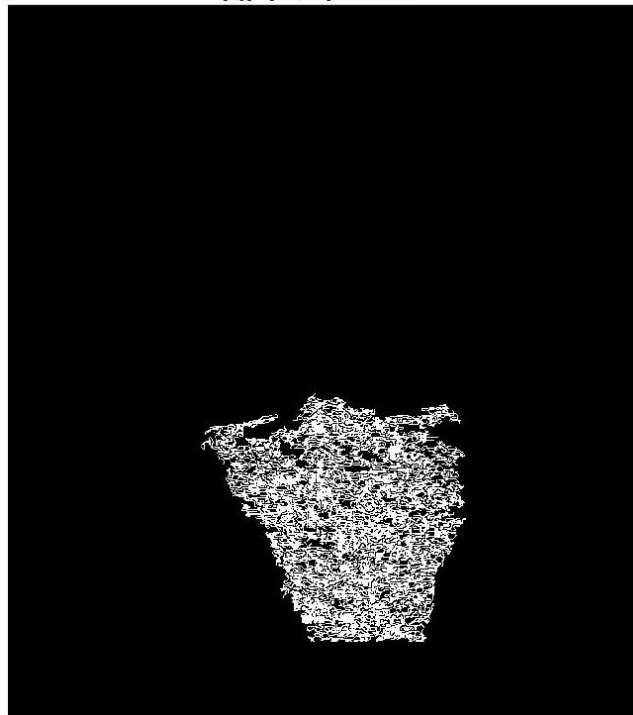
[x,y] = [339,198], threshold = 4



Input image



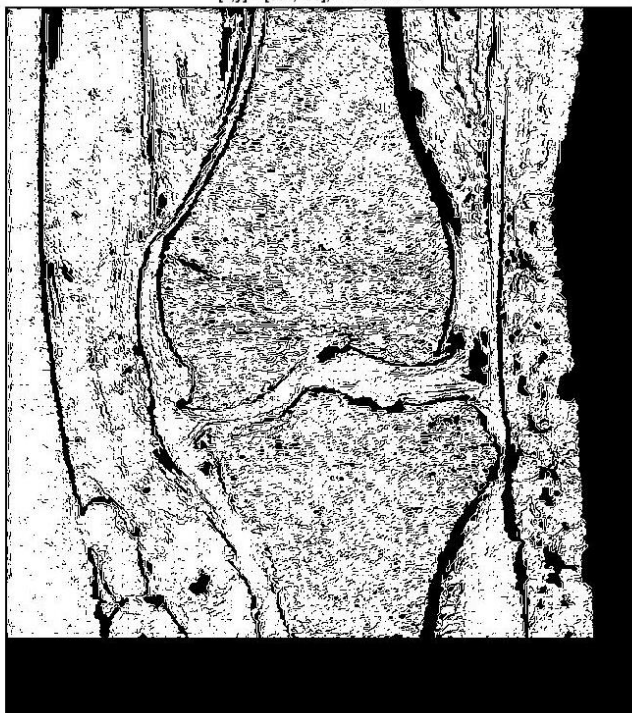
[x,y] = [402,569], threshold = 4



Input image



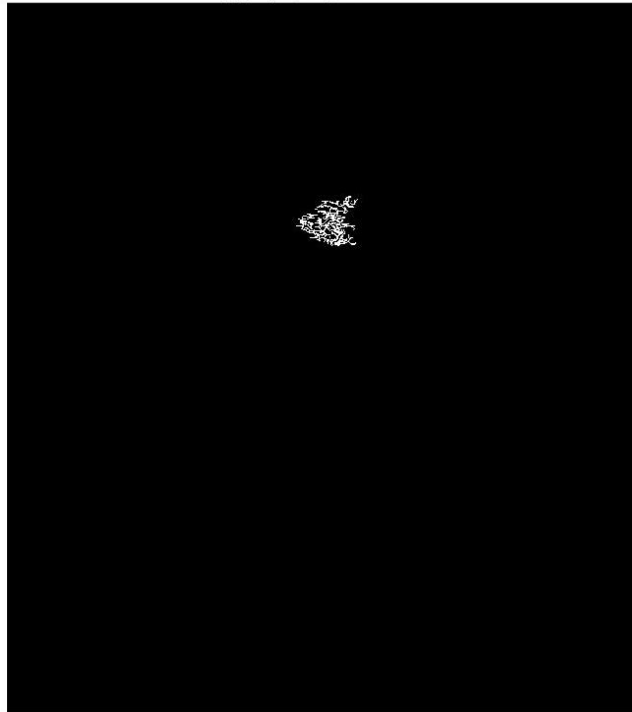
[x,y] = [332,203], threshold = 6

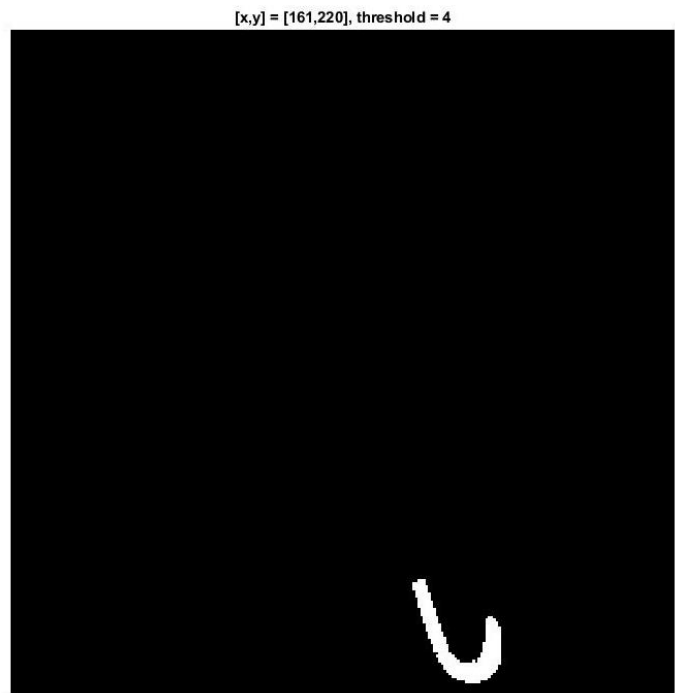


Input image



[x,y] = [358,225], threshold = 3





2. Conclusions:

This implementation can not analyze RGB images properly so the additional `rgb2grey` operation is required. Threshold with the value of 4 is the most optimal when the algorithm is working on `kneel.png` and `umbrella.png` images. Increasing the threshold value by 2 significantly lengthens the computation time and gives the most of the input image as the result area. Decreasing the threshold value gives the opposite result so the output contains a small area around the starting point.