

2
3 Homework#1 Solution Set
45 1. BoundingBoxes.m
6

```
7 function BB = BoundingBoxes(InIm)
8 [nr nc] = size(InIm);
9 img = double (InIm);
10 blobno = 1;
11 for r=1:nr
12     for c=1:nc
13         if(img(r,c)==0)
14             blobno = blobno + 1;
15             img = CreateBlob(img,blobno,r,c);
16         end
17     end
18 end
19 if(blobno==1)
20     disp('No blobs present');
21     return;
22 end
23
24 % Finds the coordinates of each blob
25 for i=2:blobno
26     [r c] = find(img==i);
27     blobinfo(i-1,1) = min(r);
28     blobinfo(i-1,2) = min(c);
29     blobinfo(i-1,3) = max(r);
30     blobinfo(i-1,4) = max(c);
31     blobinfo(i-1,5) = (max(r)-min(r))*(max(c)-min(c));
32 end
33 sortedblobs = sortrows(blobinfo,-5);
34 for i=1:blobno-1
35     BB(1,1,i) = sortedblobs(i,1);
36     BB(2,1,i) = sortedblobs(i,2);
37     BB(1,2,i) = sortedblobs(i,3);
38     BB(2,2,i) = sortedblobs(i,4);
39 end
40 end
41
42 % Label the pixel at (x,y) and all connected pixels as blobno
43
44 function blobmatrix = CreateBlob(image,blobno,x,y)
45 mover = [ 0 -1 -1 -1 0 +1 +1 +1];
46 movec = [ -1 -1 0 +1 +1 +1 0 -1];
47 [nr nc] = size(image);
48 visitedcells(1,1) = x;
49 visitedcells(1,2) = y;
50 count = 1;
51 visited = 1;
52 image(x,y) = blobno;
53 while((visited<=count) || (count==visited==1))
54     r = visitedcells(visited,1);
55     c = visitedcells(visited,2);
56     image(r,c) = blobno;
57     for i=1:8
```

```

58         if(r+mover(i)>=1 && r+mover(i)<=nr && c+mover(i)>=1 &&
59 c+mover(i)<=nc )
60             if(image(r+mover(i),c+mover(i))==0)
61                 count = count + 1;
62                 visitedcells(count,1) = r+mover(i);
63                 visitedcells(count,2) = c+mover(i);
64                 image(r+mover(i),c+mover(i)) = blobno;
65             end
66         end
67     end
68     visited = visited + 1;
69 end
70 blobmatrix = image;
71 end
72
73 2. Equidistant.m
74
75 function [OutImage] = Equidistant(InImage,DistMeas)
76 %function OutImage = Equidistant(InImage,DistMeas)
77 %Finds all the pixels that are equidistant from three given blobs.
78 %
79 %InImage:  uint8 input image containing exactly 3 connected blobs.
80 %The pixels of one blob all have value 1, the second all have value 2,
81 %the third 3, and the background pixels all have value 0.
82 %
83 %DistMeas: a string, either 'cityblock' or 'chessboard'.
84 %
85 %OutImage: double normalized image in which the original 3 blobs all
86 %have greylevel 1.0, the background has greylevel 0.0, and the pixels that
87 %are equidistant from the three blobs all have greylevel 0.5.
88
89 % create 3 images (one for each blob)
90 blob1Img = zeros(size(InImage));
91 blob2Img = zeros(size(InImage));
92 blob3Img = zeros(size(InImage));
93
94 blob1Img(find(InImage == 1)) = 1;
95 blob2Img(find(InImage == 2)) = 1;
96 blob3Img(find(InImage == 3)) = 1;
97
98 % Call Chamfer algorithm for each of the image of blob image
99 chamferedBlob1Img = Chamfer(blob1Img,DistMeas);
100 chamferedBlob2Img = Chamfer(blob2Img,DistMeas);
101 chamferedBlob3Img = Chamfer(blob3Img,DistMeas);
102
103 % create output image
104 OutImage = zeros(size(InImage));
105 blobPixels = find(InImage ~= 0);
106 OutImage(blobPixels) = 1; % blobs pixels
107
108 % Find equidistant pixels
109 equidistantPixels = find((chamferedBlob1Img ~= 0) & (chamferedBlob2Img ~= 0)
110 & (chamferedBlob3Img ~= 0) ...
111 & (chamferedBlob1Img == chamferedBlob2Img) & (chamferedBlob2Img ==
112 chamferedBlob3Img));
113 OutImage(equidistantPixels) = 0.5;
114

```

```

115
116 % chamfer algorithm
117 function chamferdImage = Chamfer(InIm,DistMeas)
118
119 % Padded the input image with zeros
120 paddedImg = zeros(size(InIm)+2);
121 [nr nc] = size(InIm);
122 paddedImg(2:nr+1,2:nc+1) = InIm;
123 % create image F
124 F = zeros(size(paddedImg));
125 idx =find(paddedImg == 0);
126 F(idx) = inf;
127
128 %create masks for AL and BL
129 AL(:,1) = [-1 ; 0 ; +1 ; -1];
130 AL(:,2) = [-1 ; -1 ; -1 ; 0];
131
132 BL(:,1) = [-1 ; 0 ; +1 ; +1];
133 BL(:,2) = [+1 ; +1 ; +1 ; 0];
134
135 if(strcmp(DistMeas,'cityblock'))
136     Dist_AL = [2 ; 1 ; 2 ; 1];
137     Dist_BL = [2 ; 1 ; 2 ; 1];
138 elseif(strcmp(DistMeas,'chessboard'))
139     Dist_AL(1:4) = 1;
140     Dist_BL(1:4) = 1;
141 end
142
143 % Scan image using AL mask
144 for c = 2 : nc+1
145     for r = 2 : nr+1
146         for i = 1 : 4
147             rQ = r + AL(i,1);
148             cQ = c + AL(i,2);
149             dist_values(i) = Dist_AL(i) + F(rQ,cQ);
150         end
151         F(r,c) = min(F(r,c),min(dist_values));
152     end
153 end
154 % scan image using BL mask
155 for c = nc+1 : -1 : 2
156     for r = nr+1 : -1 : 2
157         for i = 1 : 4
158             rQ = r + BL(i,1);
159             cQ = c + BL(i,2);
160             dist_values(i) = Dist_BL(i) + F(rQ,cQ);
161         end
162         F(r,c) = min(F(r,c),min(dist_values));
163     end
164 end
165
166 %return chamfered output
167 chamferdImage = F(2:nr+1,2:nc+1);
168
169
170 3. BinRLC.m
171

```

```

172 function BRLC=BinRLC(InIm)
173
174 [nr nc] = size(InIm);
175
176 BRLC = '(';
177
178 startc = 0;
179 endc = 0;
180 %start = 0;
181
182 for r=1:nr % traverse row
183     rowopenbracket = 0;
184     start = 0;
185     for c=1:nc % traverse column
186         if(InIm(r,c)==0) % pixel is black
187             if(rowopenbracket==0) % start of row bracket
188                 BRLC = [BRLC,'(',num2str(r)];
189                 rowopenbracket = 1;
190             end
191
192             if(start==0) % start of a sequence
193                 start = 1;
194                 startc = c;
195             end
196         else
197             if(start==1) % end of sequence
198                 endc = c-1;
199                 start = 0;
200                 BRLC = [BRLC,' ',num2str(startc),' ',num2str(endc)];
201             end
202         end
203     end
204     if(rowopenbracket == 0)
205         BRLC = [BRLC,'(',num2str(r)];
206         rowopenbracket = 1;
207     end
208     if(start==1 && c==nc) %if the last column pixel is black
209         BRLC = [BRLC,' ',num2str(startc),' ',num2str(nc)];
210     end
211     if(rowopenbracket==1) % end of row bracket
212         BRLC = [BRLC,')'];
213     end
214 end
215
216 BRLC = [BRLC,')'];
217 end

```

219 4. ApproxRLC.m

```

220
221 function ARLC = ApproxRLC(InIm)
222
223 [nr nc] = size(InIm);
224 img1 = InIm >= 0 & InIm <= 63;
225 img2 = InIm >= 64 & InIm <= 127;
226 img3 = InIm >= 128 & InIm <= 191;
227 img4 = InIm >= 192 & InIm <= 255;
228

```

```
229  BRLC1 = BinRLC(img1);
230  BRLC2 = BinRLC(img2);
231  BRLC3 = BinRLC(img3);
232  BRLC4 = BinRLC(img4);
233
234  ARLC = strvcat(BRLC1,BRLC2,BRLC3,BRLC4);
235  end
```