CTH, Image Analysis, SSY095 2009-04-15 Solutions

1a) $x = h_3 * I \quad y = h_1 * I$, ($*$ == correlation filter), Magnitude image: $\sqrt{x^2 + y^2}$, $10\sqrt{2} \approx 14$

$$x = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & -10 & 0 \\ 0 & 20 & 0 & -20 & 0 \\ 0 & 10 & 0 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix} \qquad y = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -10 & -20 & -10 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 20 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

$$\text{Magnitude image} = \begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 14 & 20 & 14 & 0 \\ 0 & 20 & 0 & 20 & 0 \\ 0 & 14 & 20 & 14 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

1b) After *hysteresis* thresholding:

$$\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}$$

Compare with Exercise BD-3

1c) Optimal for step edges corrupted by white noise.
1d) Median filter, non-linear diffusion filter …

2a) A
2b) Impossible, gray scale information is lost.
2c)

$$p_r(r) = A \cdot exp(-a \cdot r)$$

$$s = T(r) = \int_0^r p_w(w)dw = \int_0^r A \cdot exp(-aw)dw = \frac{-A}{a} exp(-ar) + \frac{A}{a} = \frac{A}{a}(1 - \exp(-ar))$$

2d) $\left(A = \frac{e}{e-1}, a = 1\right) \qquad s = A - A \cdot \exp(-r) \rightarrow r = -\ln(1 - s/A)$

$$\frac{dr}{ds} = \frac{1}{A} \cdot \frac{1}{1 - s/A}$$

$$p_s(s) = \left[p_r(r) \cdot \frac{dr}{ds}\right]_{r=T^{-1}(s)} = \mathrm{A} \cdot \exp(-r) \cdot \frac{1}{A} \cdot \frac{1}{(1 - s/A)} = \frac{\exp(\ln(1 - s/A))}{1 - s/A} = \frac{1 - s/A}{1 - s/A} = 1$$

3a) See Lecture 7, page 16.
3b) We might use Hu moment invariant Φ_7. It will change the sign for mirrored images. (see Lecture 7. Page 17)

3c) The artificial objects, like knitted fabric and checkered textile, should have the lowest textural entropy (most order in them).

3d) E.g. motion analysis, depth determination …
3e) Using Euler-Lagrange equation, dynamic programming, snake growing ..

4a) When ω=0 we have: $x(t) = x_c + \mathrm{a} \cdot cos(\mathrm{t})$, $y(t) = y_c + b \cdot sin(\mathrm{t})$, $t = 0..2\pi$

Then:

$$a = (x(t) - x_c)/\, cos(\mathrm{t})$$
$$b = (y(t) - y_c)/sin(\mathrm{t})$$

```
Algorithm 4a:
      xc= known value; yc =known value; w= known value = 0;
      A(1..M,1..N)= 0         % initialize the accumulator matrix
                              % use, for example, 1st index for a, 2nd for b
      for every pixel in the edge image I(x,y)
            if I(x,y)==1 then                        % edge pixel
                  for t=1..360 degrees
                        a = (x-xc)/cos(t)            % but avoid cos(t)=0
                        b = (y-yc)/sin(t)            % but avoid sin(t)=0
                        A(a,b) = A(a,b)+1            % give a vote to A(a,b)
                  end
            end
      end
      let maxV = maximum value of A
      [I,J] = find(A==maxV)   % find bin indices for the maxV
      Solution:
      a= value corresponding bin I, b= value corresponding bin J
```

4b) For an ellipse in general position we have:

$$x(t) = x_c + \mathrm{a} \cdot cos(t)\, cos(\omega) - b \cdot sin(t)\, sin\,(\omega)$$
$$y(t) = y_c + b \cdot sin(t)\, cos(\omega) + a \cdot cos(t)\, sin\,(\omega)$$
$$t = 0..2\pi$$

We may choose to loop over ω, a and b and calculate the corresponding (x_c, y_c) for voting:

$$x_c = x(t) - \mathrm{a} \cdot cos(t)\, cos(\omega) + b \cdot sin(t)\, sin\,(\omega)$$
$$y_c = y(t) - b \cdot sin(t)\, cos(\omega) - a \cdot cos(t)\, sin\,(\omega)$$

```
Algorithm 4b:
xc, yc, w, a, b - unknown
A= 0                    % initialize the accumulator matrix
                        % use A(a,b,w,xc,yc), for example
for every pixel in the edge image I(x,y)
   if I(x,y)==1 then                      % edge pixel
      for t=1..360 degrees
         for w=1..180 degrees       % some range of w
            for a=5..30             % some range of a
               for b=5..25          % some range of b
                  xc = x-a*cos(t)*cos(w) + b*sin(t)*sin(w)
                  yc = y-b*sin(t)*cos(w) - a*cos(t)*sin(w)
                  A(a,b,w,xc,yc) = A(a,b,w,xc,yc)+1   % give a vote to this A
               end
            end
         end
      end
   end
end
let maxV = maximum value of A
[I,J,K,L,M] = find(A==maxV)   % find bin/indices of A, with the maxV
Solution:
a=I, b=J, w=K, xc=L, yc=M
%(or more properly a = value corresponding bin I, and so on ..)
```

5a) See exercise BD-6.

5b) $w_1=2$, $w_2=3$
As the smoothness cost doesn't depend on the direction and all of the brightness costs are equal then all of the paths inside the additional layers are equivalent.

Additional cost = brightness cost + smoothness cost =
$w_1*(1+2+3+4+5) + 5*w_2*1 = 2*15+5*3 = 45$

Total cost = C + 45, where C is the minimum cost from part **(5a)**

5c) Every layer contributes to 4 paths, 5 layers → 4^5 =1024 paths

Total number minimum cost paths = 1024*N, where N is the number of minimum cost paths from **(5a).**

6a)
$P_1 + P_2 = 1$ * = sum *of a priori* probabilities=1
$P_i = k A_i \rightarrow P_1/P_2=A_1/A_2$ * P_i proportional to A_i

→ $P_i = A_i/(A_1 + A_2)$
$A_1=4200$ $A_2= 1080$
$P_1= 35/44$ $P_2=9/44$

6b) Optimal threshold: when the weighted pdfs cross each other :

y= -0.75 x +105 , Y=18→ T_{opt} = 116

Error = E1 + E2 = (140-116)*18/2 + (116-110)*18 = 324
Total area = A=60*18+ 60*60/2 + 60*80/2 = 5280
Minimal error = (E1+E2)/A= 0.0614 = ca 6%

6c)
$T > 160$ → sensitivity =0
T in [100 .. 160] → sensitivity = (160-T)/60
$T \leq 100$ → sensitivity =1

6d)
$T \geq 140$ → specificity = 1
T in [60 ..140] → specificity = 1- (140-T) /2)* (-0.75*T+105)/4200
T in [0 ..60] → specificity = T*T/(2*4200)