UNIVERSITY OF OSLO

Faculty of Mathematics and Natural Sciences

Exam:	**INF 4300 / INF 9305 – Digital image analysis**
Date:	**Thursday December 4, 2014**
Exam hours:	**14.30-18.30 (4 hours)**
Number of pages:	**8 pages of sketches to a solution**
Enclosures:	**None**
Allowed aid:	**Calculator**

- Read the entire exercise text before you start solving the exercises. Please check that the exam paper is complete. If you lack information in the exam text or think that some information is missing, you may make your own assumptions, as long as they are not contradictory to the “spirit” of the exercise. In such a case, you should make it clear what assumptions you have made.

- You should spend your time in such a manner that you get to answer all exercises shortly. If you get stuck on one question, move on to the next question.

- Your answers should be **short**, typically a few sentences and / or a sketch should be sufficient.

Good luck!!

Exercise 1: Texture

You are given a gray level image of size MxN pixels with b bits per pixel.

a) Describe how a Gray Level Coccurrence Matrix is computed, and which parameters it involves.

Answer: This should be well known stuff!

b) Many GLCM features may be seen as a weighted sum of the cooccurrence matrix element values, where the weighting applied to each element is based on a given *weighting function* W(i, j). Such weighting functions fall into two categories:

1. Weighting based on the *value* of the GLCM element
2. Weighting based on the *position* in the GLCM.

Explain what we mean by this, using an example from each of the two categories.

Answer:An obvious example of the first is the Entropy feature,

$$E = -\sum_{i=0}^{G-1}\sum_{j=0}^{G-1} P(i,j) \times log(P(i,j))$$

Where –logP(i,j)is applied as a weight function when summing the P(i,j)elements of the matrix. Any of the four features given below may serve as an example of the second category.

c) Given four GLCM features; Inertia, Inverse Difference Moment, Cluster Shade, and Cluster Prominence, see equations below:

$$INR = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{i-j\}^2 \times P(i,j)$$

$$IDM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\frac{1}{1+(i-j)^2}P(i,j)$$

$$SHD = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{i+j-\mu_x-\mu_y\}^3 \times P(i,j)$$

$$PRM = \sum_{i=0}^{G-1}\sum_{j=0}^{G-1}\{i+j-\mu_x-\mu_y\}^4 \times P(i,j)$$

Discuss which of these features that can be used as an image edge detector, in the sense that a high feature value within a local window indicates a high gradient magnitude detected for the direction that P(i,j) has been accumulated.

Answer:

- The Inertia weighting function is zero along the diagonal (i = j), and increases towards $(G-1)^2$ away from the diagonal. Thus, it will favor contributions from P(i, j) away from the diagonal (i≠j), i.e., give higher values for images with high local contrast for the given (d,θ)-values). Useful as a local edge detector.
- The IDM weighting function has its maximum (W = 1) along the diagonal (i = j) and falls off toward (G2 − 2G + 2)−1 away from the diagonal. The result is a low IDM value for inhomogeneous images (i≠j), and a relatively higher value for homogeneous images (i≈j). Useless as a local edge detector.
- The Cluster Shade and Cluster Prominence weighting functions are zero along the bi-diagonal of the GLCM. They work on the sum, not the difference of i and j, and are therefore useless as local edge detectors.

Exercise 2: Representation of binary object boundaries

One of the methods of representing the contour of binary objects traverses the N boundary pixels clockwise and generates a sequence of N codes given by this algorithm:

$$c_i = Code\,(\Delta x, \Delta y)$$

$$where\ (\Delta x, \Delta y) = \begin{cases} (x_{i+1} - x_i, y_{i+1} - y_i) & for\ 0 \le i < N-1 \\ (x_0 - x_i, y_0 - y_i) & for\ i = N-1 \end{cases}$$

and $Code\,(\Delta x, \Delta y)$ is defined by the following table:

Δx	1	1	0	-1	-1	-1	0	1
Δy	0	1	1	1	0	-1	-1	-1
$Code\,(\Delta x, \Delta y)$	0	1	2	3	4	5	6	7

a) Which representation method is this?

Answer: It is an absolute chain code, using 8 neighbors coded like the familiar graph used in the lectures.

b) Given the representation above. With a sequence of N codes, we may use the following transformation:

$$d_i = \begin{cases} (c_{i+1} - c_i) \bmod 8 & for\ 0 \le i < N-1 \\ (c_0 - c_i) & for\ i = N-1 \end{cases}$$

where (<expression>)mod 8 finds the remainder of an integer division of <expression> by 8.

What is this transformation from a sequence of N *c-codes* to *d-codes* called.
What is it used for?

Answer: This is the first difference, converting the absolute code to a differential code. This is useful if we want a coding that is invariant to object rotation, when the start point is kept constant relative to the shape of the object.

c) We often perform circular shifts of a sequence of codes.
What is the purpose of that operation?

Answer: We are looking for the minimum value, as this will give us a start point invariant representation of the contour of the object.

PHD students only:

d) When we perform circular shifts of a sequence of codes to find an extremum value, the magnitudes may easily become too large to actually be computed.
How can we very easily determine which shift that gives the desired extremum?

Answer: By simple lexicographic ordering of the different shifted sequences, without actually computing any aritmetic values.

Exercise 3: Linear classification

In this exercise we work with a Gaussian classifier with equal diagonal covariance matrices, which results in a linear classifier. Consider two classes with equal prior probabilities $P(\omega_i)$. The discriminant functions for this classifier are given as:

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + w_{i0}$$

$$\mathbf{w}_i = \frac{1}{\sigma^2} \boldsymbol{\mu}_\mathbf{i}$$

$$w_{i0} = -\frac{1}{2\sigma^2} \boldsymbol{\mu}_i^T \boldsymbol{\mu}_\mathbf{i}$$

a) Consider 2D feature space defined by two features x_1 and x_2. Assume that we have two classes with class means μ_1 and μ_2

$$\mu_1 = \begin{bmatrix} -2 \\ 0 \end{bmatrix} \quad \mu_2 = \begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

Plot them and the associated decision boundary for the classifier given above.

Answer: The decisioun boundary will be perpendicular to the line connecting the means, and intersecting the point halfway between them.

b) We know that the equation $g_1(x)=g_2(x)$ can be written as:

$$\mathbf{w}^T(\mathbf{x} - \mathbf{x}_o) = 0$$

$$\mathbf{w} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$$

$$\mathbf{x}_0 = \frac{1}{2}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2)$$

Draw the point $\mathbf{x}_0$ in your plot.

c) Consider points **z** and **y**, where

$$z = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad y = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

1. Draw the vectors $\mathbf{z}$-$\mathbf{x}_0$ and $\mathbf{y}$-$\mathbf{x}_0$ on your plot.
2. Which class are these points classified to?

Answer: Z is classified to class 1 and y to class 2.

d) Study the angle θ between the vector $\mathbf{z}$-$\mathbf{x}_0$ and the vector $\mathbf{w} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$, and compare it to the angle between vector $\mathbf{y}$-$\mathbf{x}_0$ and the vector $\mathbf{w} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2$.

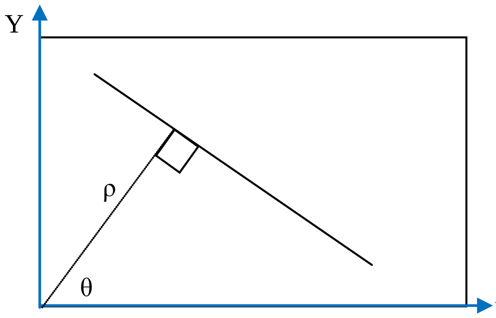
1. Discuss how this angle will change depending if the point is classified to class 1 or 2.
2. How can $cos(\theta)$ be used to determine the class label of a sample?

Answer: The angle will change from $<\pi/2$ to $> \pi/2$ and $cos(\theta)$ will change sign if we compare points from the different classes.

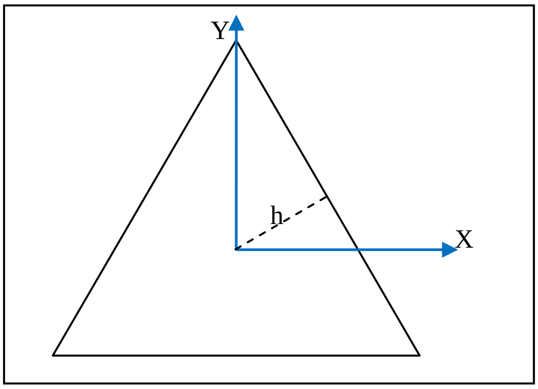
Exercise 4: Hough transform of lines, triangles, and ellipses

a) Copy the sketch below of a line segment in a 2D image.
Explain the "normal representation" of a straight line in the Hough transform, and indicate the parameters involved.

Answer: The "normal representation" is the (θ,ρ)-representation of lines, where a normal to the line through the origin is drawn, and θ is the orientation of this normal with respect to the x-axis, and ρ is its length, as illustrated by the dotted line in the figure below.
For all points (x',y') along a line segment, the distance to the origin is given by ρ'=x'cosθ'+y'sinθ'. This is also true for the point (x,y) where the normal through the origin intersects the line segment, giving the unique representation of the line segment.



b) An equilateral triangle of a given size is positioned so that the origin coincides with the centre of mass of the triangle, as shown in the sketch below.



Draw a sketch to indicate how this triangle will be represented in the Hough domain, using the normal representation of lines.

Answer: In an equilateral triangle, the normals onto the three sides from the centre are of the same length, so their length are all ρ=h. The direction of the three normals – or their extensions-, given that the θ-domain is usually limited to [-π/2,π/2], are -π/2 (or π/2) for the bottom line, π/6 for the right hand line, and –π/6 for the left hand line. So the three three peaks will be π/3 apart.

c) What will happen in the Hough domain, if the triangle is rotated anti-clockwise around its centre of mass, and the angular domain is limited to [-π/2,π/2]?

Answer: The three maxima in the (θ,ρ)-domain will slide in the positive θ-direction, keeping the 60 degree (π/3) distance between the maxima and the constant ρ-value, sliding out of the [-π/2,π/2]-domain at π/2 and re-appearing at –π/2.

For the PhD-students only (d) and e)):

d) The Hough transform is most often based on a thresholded gradient magnitude image. From a look at the normal representation of straight line segments in a gradient magnitude image, how can you determine whether the gradient direction information has been used in addition to the gradient magnitude?

Answer: If only the thresholded gradient magnitude is used, sets of sinusoids representing all the pixels of each straight line segment willform "butterfly" patterns, where sinusoids are interacting to overlap into local maxima in the Hough domain.

If gradient direction information is used, θ is known at least approximately, so each pixel on the line will only give rise to a single point in the Hough domain, and the points will accumulate and overlap to give a local maximum without "butterfly" wings.

e) If the triangle is substituted by an ellipse having unknown semi-axes (a, b) and orientation θ, and the ellipse is positioned so that the origin coincides with the centre of mass, we will need a 3D Hough space [a,b,θ]. Where in this Hough-space will we find all variants of an elliptical object, given that the area of the object is constant (A), and that the angular domain is [-π/2,π/2]? Hint: The area of an ellipse is A = πab.

Answer: The hint gives the area of an ellipse as A=πab. So if the ellipse degenerates into a circle, that circle will have a radius $r=(A/\pi)^{1/2}$, so while its orientation is undetermined, its parameters are on a line parallel to the θ-axis at $[(A/\pi)^{1/2}, (A/\pi)^{1/2},\theta]$.

The locus of all ellipses will be on a hyperbolic surface passing through this line, since b =A/(πa).

Exercise 5: Morphology on gray level images

In graylevel morphology, we may use non-flat structuring elements.
We then define the two basic operators like this

$$[f\,\theta\,h](x,y)=\min_{forall\,(i,j)\in h}\{f(x+i,y+j)-h(i,j)\} \qquad \text{(erosion)}$$

$$[f\oplus h](x,y)=\max_{forall\,(i,j)\in h}\{f(x-i,y-j)+h(i,j)\} \qquad \text{(dilation)}$$

Given the 3x3 non-flat structuring element where empty means that the operation does not depend on the value in the corresponding position. The origin is at the centre of the structuring element. Given the 6x6 pixel gray level image below:

x	1	x
1	2	1
x	1	x

6	7	3	4	8	7
5	6	6	8	3	6
6	4	5	2	5	5
6	4	2	3	7	4
5	6	4	5	4	3
7	6	5	4	3	2

a) Give the formula for morphological closing in terms of the basic operators above.

Answer: Closing is defined by the equation

$$f \bullet h=(f\oplus h)\,\theta\,h$$

b) Find the closing of the central part of the given gray level image, i.e., where the entire structuring element is inside the image.
Give the results of each step.

Answer: The first step in a closing operation is a dilation (max), which gives the result:

8	9	10	9
7	7	9	8
7	6	8	9
8	7	7	8

The second step is an erosion (min) of the result of the dilation, giving the closing:

5	6
4	5

c) What will change in the handling of the structuring element, and what is the result if the structuring element is changed to

0	1	0
1	2	1
0	1	0

Answer: The only change is that the "don't care" parts of the structuring element now have to be included, even if their values are "0". This only affects one pixel in the dilation result:

8	9	10	9
7	8	9	8
7	6	8	9
8	7	7	8

And it does not affect the result of the erosion of the previous result

5	6
4	5

d) As Masters or PhDs in Informatics, you should be able to read scientific papers and understand what methods that have been used.
In a 30 year old manuscript describing a method to estimate a feature called the fractal dimension of a gray level image f(x,y), the gray level image surface f(x,y) is successively covered by a "blanket" of an increasing thickness from above and below, so that the resolution is reduced. The covering blankets are defined by their upper surface u_ε and lower surface l_ε. An exerpt of the text goes like this:

"Initially, given the gray level function *f(x,y)*, $u_0(x,y) = l_0(x,y) = f(x,y)$.
Then for ε = 1, 2, 3, …, the successive blanket surfaces are defined as follows:

$$u_\varepsilon(x,y) = \max\left\{u_{\varepsilon-1}(x,y)+1, \max_{|(m,n)-(x,y)|\le 1}\left[u_{\varepsilon-1}(m,n)\right]\right\}$$

$$l_\varepsilon(x,y) = \min\left\{l_{\varepsilon-1}(x,y)-1, \min_{|(m,n)-(x,y)|\le 1}\left[l_{\varepsilon-1}(m,n)\right]\right\}$$

The image points *(m,n)* with distance less than one from *(i,j)* were taken to be the four immediate neighbors of *(i,j)*.

Which simple morphological operators do these two equations describe, and what is the size and shape of the structuring element?

Answer: The first is simply a gray level dilation and the second is a gray level erosion of the previous blanket surface, where the midpoint of the structuring element is raised / lowered by 1. So the structuring element is plus-shaped (4-neighbors) with values given by:

```
x 0 x
0 1 0
x 0 x
```

Thank You for Your Attention!