HW #1 Solutions

ECE 178 WINTER 2004

B.S. Manjunath

TAs: Srivatsan Pallavaram, Evan Ruzanski and Christopher Utley

Problem 2.2

Brightness adaptation.

Problem 2.5

From the geometry of Fig. 2.3,

Height of image = height of the lens = 7 mm (given)
Focal length (distance of image from lens) = 35 mm (given)
Height of object = z mm (say)
Distance of object = 500 mm (given)

We know by theory of similar triangles applied to Fig. 2.3 that,
Height of image / Focal length = Height of object / Distance of object from lens

⇨ 7 mm / 35 mm = z = 500 mm, or z = 100 mm

So the target (object) height is 100 mm on the side. Now, for 1 line on the object we have 1024 elements on the CCD. So the resolution of 1 line is 1024 / 100 = 10 elements/mm.

⇨ For 1 linepair (lp) the resolution is 5 lp/mm.

Problem 2.7

The image in question is given by

$f(x, y) = i(x, y)r(x, y)$
$= 255\,(\exp(-[(x-x_0)^2+(y-y_0)^2]\,))$ (1.0)
$= 255\,(\exp(-[(x-x_0)^2+(y-y_0)^2]\,))$

A cross section of the image is shown in Fig. P2.7(a). If the intensity is quantized using m bits, then we have the situation shown in Fig. P2.7(b), where $\Delta G = (255 + 1)=2^m$. Since an abrupt change of 8 gray levels is assumed to be detectable by the eye, it follows that $\Delta G = 8 = 256=2^m$, or m = 5. In other words, 32, or fewer, gray levels will produce visible false contouring.

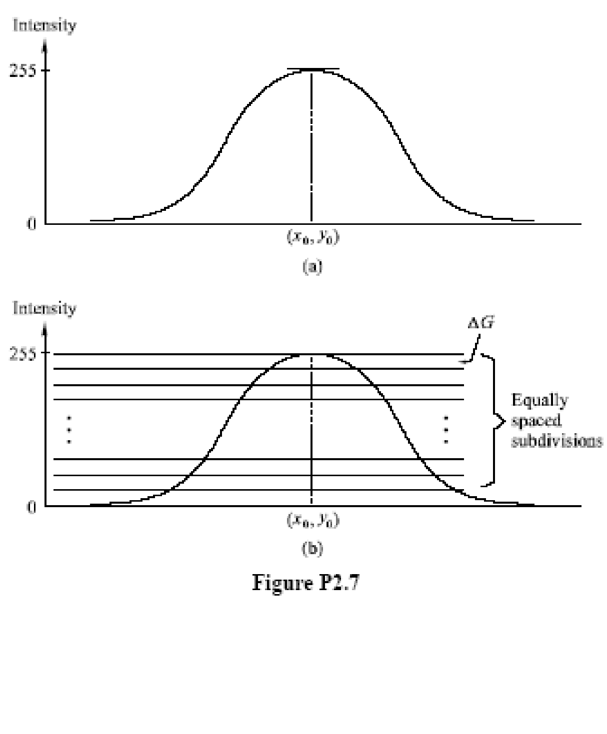

Figure P2.7

MATLAB Code: Brightness Adaptation Experiment

```
function hw1_04( )
%Description: Brightness Adaptation Experiment Program
%Author: Srivatsan PALLAVARAM
%Date: 01/29/2004 Context: ECE 178 HW#1, winter 2004, UCSB
close all

x=zeros(257);
y=zeros(257);
a=-129;
%Creating a matrix x with its values representing the x-coordinate of the
%corresponding pixels w.r.t the center of the matrix
for i=1:257,
    a=a+1;
    x(:,i)=a;
end

b=129;
%Creating a matrix y with its values representing the y-coordinate of the
%corresponding pixels w.r.t the center of the matrix
for i=1:257,
    b=b-1;
    y(i,:)=b;
end

z=zeros(257); %actual image
radius_dec=10; %the number of pixels by which we plan to reduce the radius of the circle
intensity_inc=1; %the number of graylevels by which we plan to increase the intensity
radius=128; %initial radius for a 256X256 image
intensity=1; %initial intensity of the outermost circle

while (intensity<256 && radius>0)
    for i=1:257,
        for j=1:257,
            %Identifying the pixels that lie within the radius of the
            %current circle and setting their intensities using the user's
            %perception-based feedback
            if (sqrt(x(i,j)*x(i,j)+y(i,j)*y(i,j)) <= radius)
                z(i,j)=intensity;
            else
            end %end of if
        end %end of for
    end %end of for
    imshow(uint8(z));
    %Getting user feedback
    ButtonName=questdlg('Can you see the new circle?', 'Checking your perception', 'No', 'Yes', 'No');
    %Checking feedback
    if strcmp(ButtonName, 'Yes')
        radius=radius-radius_dec; %Decrement the radius by whatever step you choose
    else
    end %end of if
      intensity=intensity+intensity_inc; %Increment intensity by whatever step you choose
end %end of while
close all
z=z(1:256,1:256);
imshow(uint8(z));
title('Final result');
```