CEG4311 Image Processing
Sept. 12, 2007

Getting started with creating and visualizing two-dimensional functions

Creating two-dimensional functions defined by an analytical formula can be done quite compactly using MATLAB matrix expressions. The steps are:

1. Define the set of points (x,y) at which the function will be evaluated, typically an N by N array of points. The value of N will depend on the type of plot and may be lower for perspective plots, and higher for contour and intensity plots.
2. Create two N by N arrays, the first with the x coordinate at each point and the second with the y coordinate, using the MATLAB function `meshgrid`.
3. Write a MATLAB expression to evaluate the given function at all the N^2 points in one shot.
4. Display a perspective, contour, or intensity plot as desired.
5. Set parameters of the plot, axes, etc.

Example 1
Generate a two-dimensional Gaussian function, centered at (0.5,0.5) and with $r = 0.2$ for $0 \le x, y \le 1$

$$f(x, y) = \exp[-((x-0.5)^2 + (y-0.5)^2)/2(0.2)^2]$$

a) suitable for perspective plot:

```
N=51;                   %generate a 51 by 51 array of points
x=0:(1/(N-1)):1; y=x; % x and y values between 0 and +1 spaced by %
                        0.02 (51 values)

r = 0.2;                %set the r parameter

[X,Y]=meshgrid(x,y);  % generate 2D grids of xy values (51 by 51
                        %arrays) X(i,j) = x(j) and Y(i,j) = y(i)

Z=exp(-((X-0.5).^2+(Y-0.5).^2)/(2*r^2));   % generate the
                        %Gaussian function on the grid

mesh(X,Y,Z)             % generate the perspective plot

colormap([0 0 0]);    % use black only
xlabel('x (ph)'), ylabel('y (ph)');
set(gca,'ydir','reverse');
```

Exercise: Generate a Gaussian function with parameter 0.12 and centered at (0.55, 0.45) and plot a perpective view. What is the effect of the parameter r?

b) suitable for contour plot:

```
N = 201;                %generate a 201 by 201 array of points
x=0:(1/(N-1)):1; y=x; % x and y values between 0 and +1 spaced by %
                        0.005 (201 values)

r = 0.2;                %set the r parameter

[X,Y]=meshgrid(x,y);  % generate 2D grids of xy values (201 by 201
                        %arrays) X(i,j) = x(j) and Y(i,j) = y(i)

Z=exp(-((X-0.5).^2+(Y-0.5).^2)/(2*r^2));   % generate the
                        %Gaussian function on the grid

v=0:.1:1.;              % contours will be from 0 to 1 in steps of 0.1
[C,h]=contour(X,Y,Z,v); % generate the contour plot, including values
                        %to label contours
axis square             %make the plot square
clabel(C,h)             %label the contours
xlabel('x (ph)'), ylabel('y (ph)');
set(gca,'ydir','reverse');
set(gca,'XAxisLocation','top');      %Xaxis labels on top
colormap([0 0 0]);     % use black only
```

Exercises: Generate a Gaussian function with parameter 0.12 and centered at (0.55, 0.45) and plot a contour view. How does this last plot look if you use N = 21? View this latter function (with N = 201) as an intensity plot using `imshow(Z)`.

Example 2

Generate a two-dimensional rectangular zero-one function $f(x, y)$, centered at (0.5,0.5) and of width 0.3 and height 0.4 . Note that there is no MATLAB `rect` function! Mathematically, this function is given by $f(x, y) = rect\left(\frac{x-0.5}{0.3}, \frac{y-0.5}{0.4}\right)$

This can be generated in MATLAB as follows, using a procedure that can be used for many zero-one functions.

```
%plot a rect function as a perspective plot
N=51;   %generate an N by N array of points
x=0:(1/(N-1)):1; y=x;   % x and y values between 0 and +1 spaced by
1/(N-1) (N values)
[X,Y]=meshgrid(x,y);    % generate 2D grids of xy values (N by N
arrays) X(i,j) = x(j) and Y(i,j) = y(i)

Z = zeros(N,N);
Z(abs(X-0.5) < 0.15  & abs(Y-0.5) < 0.2) = 1;
mesh(X,Y,Z) % generate the perspective plot

colormap([0 0 0]);  % use black only
xlabel('x (ph)'), ylabel('y (ph)');
set(gca,'ydir','reverse');
```

Exercises: Plot a rect function centered at (0.35, 0.55) of width 0.15 and height 0.25. Plot a circ function centered at (0.5, 0.55) and of radius 0..15. View these as contour plots and intensity plots as well.