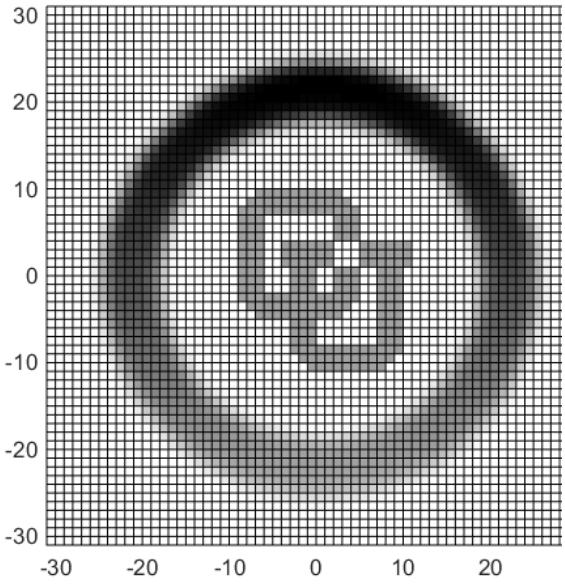
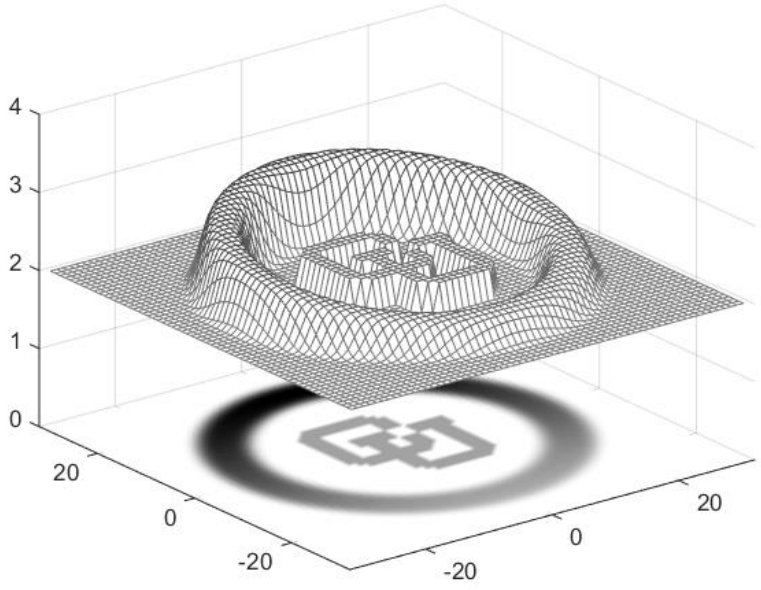
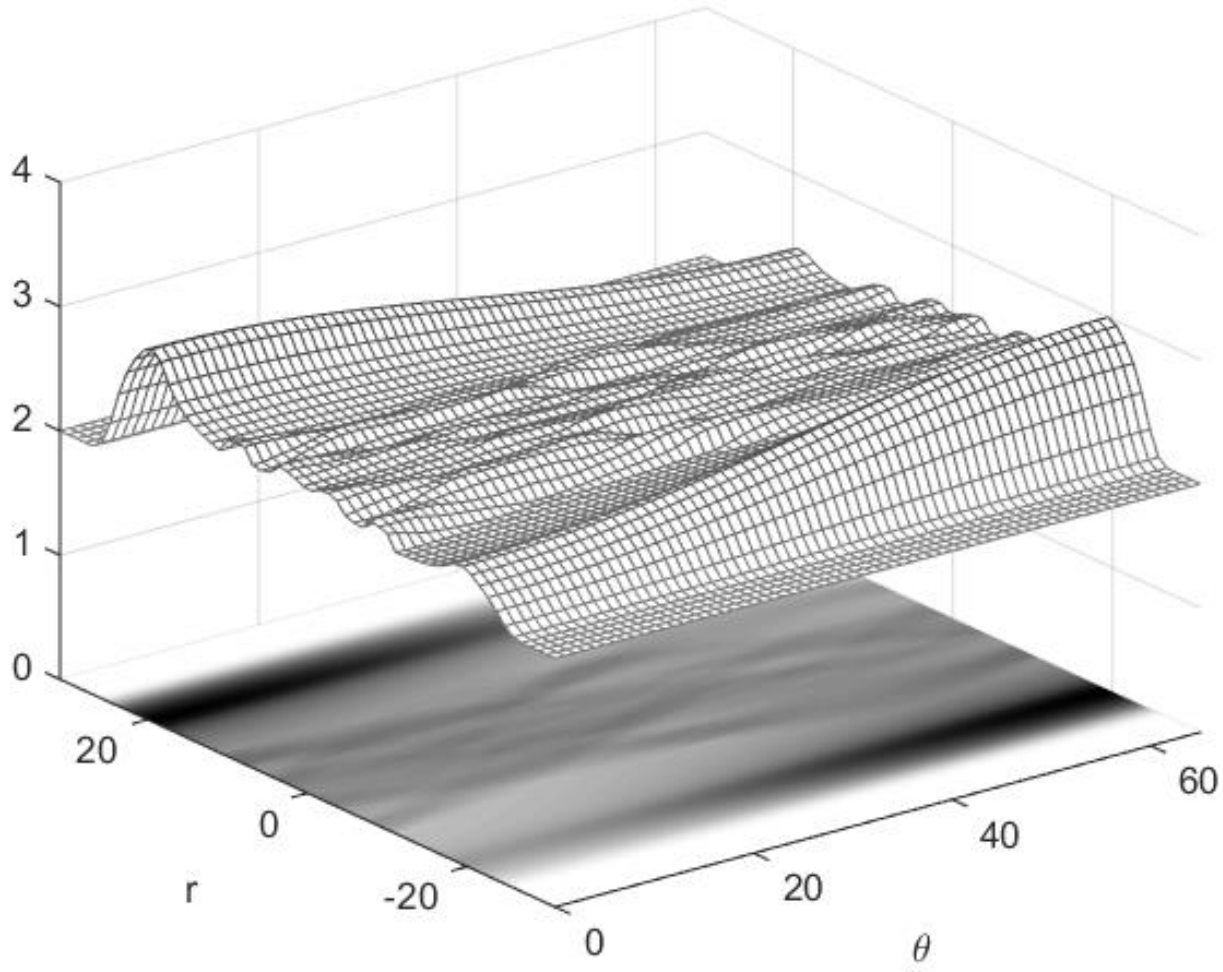
```
% LOGO_SCAN  Routine lays out CU logo inside a ring, and then simulates a scan.
close all;
% Creation of logo that is 'hardwired' to the size [-31,31]x[-31,31]

cu=['   XXXXXXXXXX        ';...  % The array  a(i,j) is structured as
    '  XXXXXXXXXXXX       ';...  % follows:
    ' XXXXXXXXXXXXXX      ';...  %
    ' XXX        XXX      ';...  %                 ^ y
    ' XXX        XXX      ';...  %                 |
    ' XXX        XXX      ';...  %   63  + - - - - + - - - - +
    ' XXX  XXXXXX   XXXXXX';...  %                  31
    ' XXX  XXXXXX   XXXXXX';...  %       |         |         |
    ' XXX  XXXXXX   XXXXXX';...  %    .
    ' XXX   XXX  XXX  XXX ';...  %    .  |         |         |
    ' XXX   XXX  XXX  XXX ';...  %       -31                31
    ' XXX   XXX  XXX  XXX ';...  %    i  + - - - - + - - - - + - > x
    ' XXXXXXXXXXXXXX  XXX ';...  %
    '  XXXXXXXXXXXX   XXX ';...  %    .  |         |         |
    '   XXXXXXXXXX    XXX ';...  %    .
    '       XXX       XXX ';...  %       |         |         |
    '       XXX       XXX ';...  %                 -31
    '       XXX       XXX ';...  %    1  + - - - - + - - - - +
    '       XXXXXXXXXXXXX ';...  %
    '        XXXXXXXXXX   ';...  %       1      .. j ..      63
    '         XXXXXXXX    '];    %

% Create ring
x = -31:31;
[xx,yy] = meshgrid(x,x);        % 2-D arrays with x and y-coord. respectively
r = sqrt(xx.^2+yy.^2)/31;       % Distance to center of grid
a = 2*exp(max(-40.0,-6000.0*(r-0.7).^4)).*(1.1-0.75*yy/31.0);

% Place CU logo at the middle of the ring; replace value by 1.2
% wherever there is an 'X' (or other non-blank symbol) in logo
ac = a(22:42,22:42); ac(cu~=' ') = 1.2; a(22:42,22:42) = ac;
a = a/max(max(a));              % Normalize a so in range [0,1]

% Display test object gridpoint-by-gridpoint using 2-D gray scale
gr = gray;                      % Pick up standard gray map, but swap
colormap(gr(64:-1:1,:))         % so 0 values (background) becomes light
pcolor(xx,-yy,a); axis equal ; axis tight

% Start new figure; Surface plot plus grey scale image below
figure
colormap(gr(64:-1:1,:))         % Again swap direction of grey scale
mesh(xx,-yy,a+2)
axis ([-32 32 -32 32 0 4])
hold on                         % Want to do two plots using the same axes
pcolor(xx,-yy,a*4); shading interp   % Remove grid lines in base plane

% Create the scan. This time specify size as desired
nr = 60;                        % Number of rays for each angle
m  = 66;                        % Number of angles to use
sc = zeros(nr,m);               % Array to hold the scan vectors

% Describe (still non-rotated) grid for scanning
n2 = (nr+1)/2-1;  xt = linspace(-31,31,nr);
[xx1,yy1] = meshgrid(xt,xt);
F = griddedInterpolant(xx',yy',a','cubic'); % Prepare image for interpolation

for j=0:m-1                     % Loop over m angles
  th = pi*j/m; st = sin(th); ct = cos(th);
  xr = xx1*ct-yy1*st;           % Generate the x1,x2-coordinates for
  yr = xx1*st+yy1*ct;           % the grid points in the xp1,xp2 grid.
  s  = F(xr',yr')';             % Interpolate
  sc(:,j+1) = sum(s,2);         % Sum along each ray
end
sc = sc/max(max(sc));           % Normalize so max value = 1.

% Display scan data as surface plot plus grey scale image below
figure
colormap(gr(64:-1:1,:))
mesh (0:m-1,-xt,sc+2)
axis ([0 m-1 -31 31 0 4])
hold on
pcolor (0:m-1,-xt,sc*4); shading interp
xlabel('\theta'); ylabel('r')
```


30
20
10
0
-10
-20
-30
-30
-20
-10
0
10
20


4
3
2
1
0
20
0
-20
20
0
-20


4
3
2
1
0
20
0
r
-20
0
20
40
60
θ