Blurring

g = fspecial('gaussian', 15, 3);
imtool(g,[])
surf(g);

I = imread('coins.png');
I2 = imfilter(I, g);
Reducing Noise

\[ I = \text{imread('coins.png')}; \]
\[ \text{IN} = \text{imnoise}(I, \text{'gaussian'}, 0, 0.01); \]
\[ g = \text{fspecial('gaussian', 5, 1)}; \]
\[ f = \text{imfilter}(\text{IN}, g); \]
\[ p = \text{improfile(IN, [100 150], [100 100]);} \]
\[ \text{figure, plot}(p) \]
\[ p = \text{improfile}(f, [100 150], [100 100]); \]
\[ \text{figure, plot}(p) \]
“Unsharp Masking”

- Get an image (say “moon.tif”)
- Blur using a 2D Gaussian
- Subtract blurred image from the original
- Add that difference to the original image (times some constant k)

```matlab
>> I = double(I);
>> b = imfilter(I, g);
>> diff = I - b;
>> figure, imshow(diff,[])
>> I2 = I + 2*diff;
>> figure, imshow(I2,[])
```
Edge Enhancement

• Compute the gradient of the “coins.png” image by filtering the image with the Sobel operators in the horizontal and vertical directions.

• Display the gradient magnitude and the gradient angles.

```matlab
clear all
close all

I = imread('coins.png');
I = double(I);

hx = [-1 0 1; -2 0 2; -1 0 1];
hy = hx';

Gx = imfilter(I, hx);
Gy = imfilter(I, hy);
GM = (Gx.^2 + Gy.^2) .^ 0.5;
imshow(GM, []);

GA = atan2(Gy, Gx);
figure, imshow(GA, []);
```
Edge Enhancement (continued)

- Threshold the gradient magnitude image, using “T = GM > v;”, where GM is the gradient magnitude image and v is some value that you select experimentally.
- Multiply the threshold image with the gradient angle image. This should zero out the noisy angle values, corresponding to the places where the gradient magnitude was too small.

```matlab
T = (GM>200);
GA = T .* GA;
figure, imshow(GA, []);

% colormap jet
% colorbar
```