

Fall 2021

# California State University, Northridge

Department of Electrical and Computer Engineering

Computer Assignment 4

November 5, 2021

ECE 551

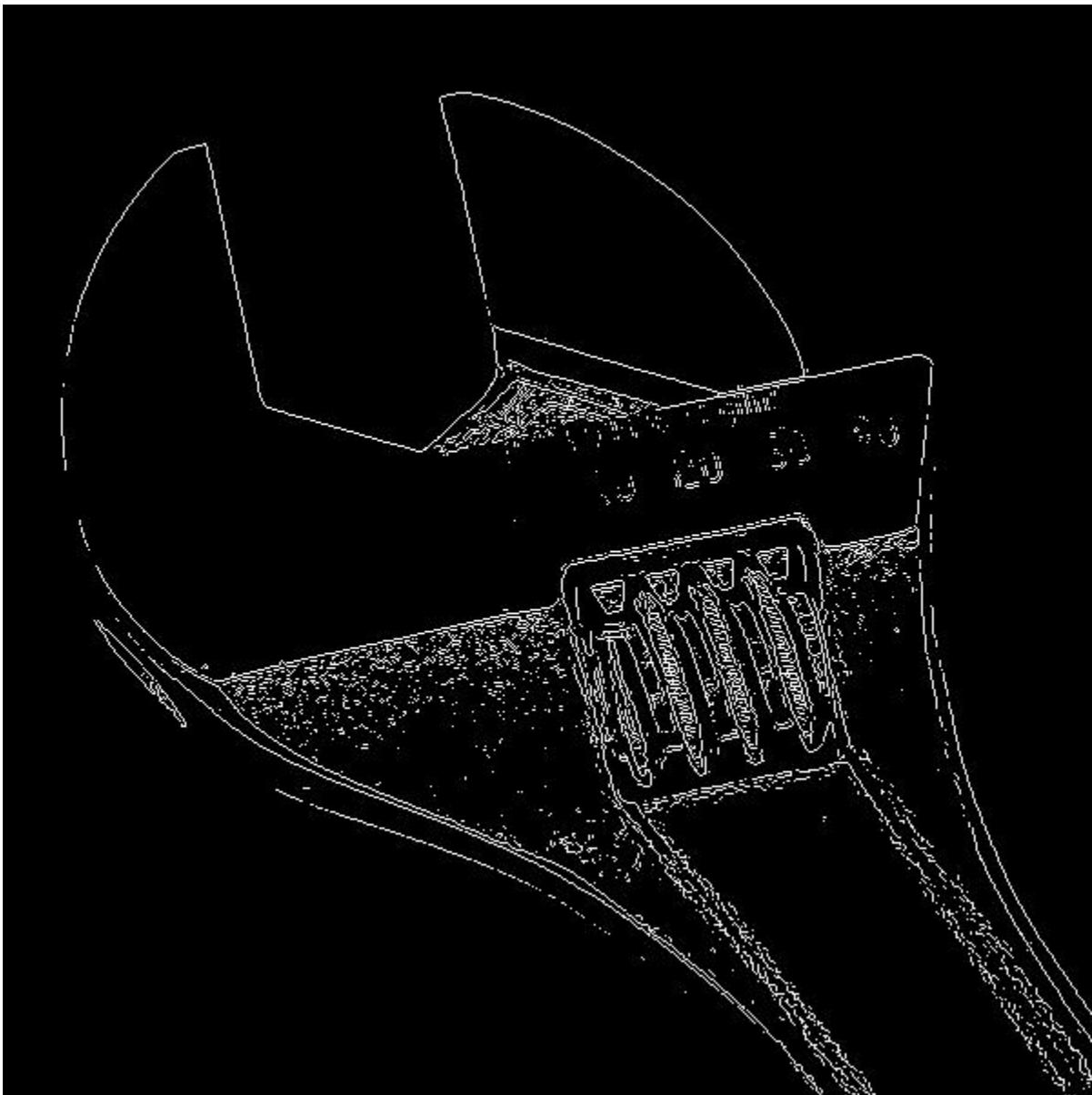
Professor: Shahnam Mirzaei

Written By: Morris Blaustein

## Experiment 1

### 1.

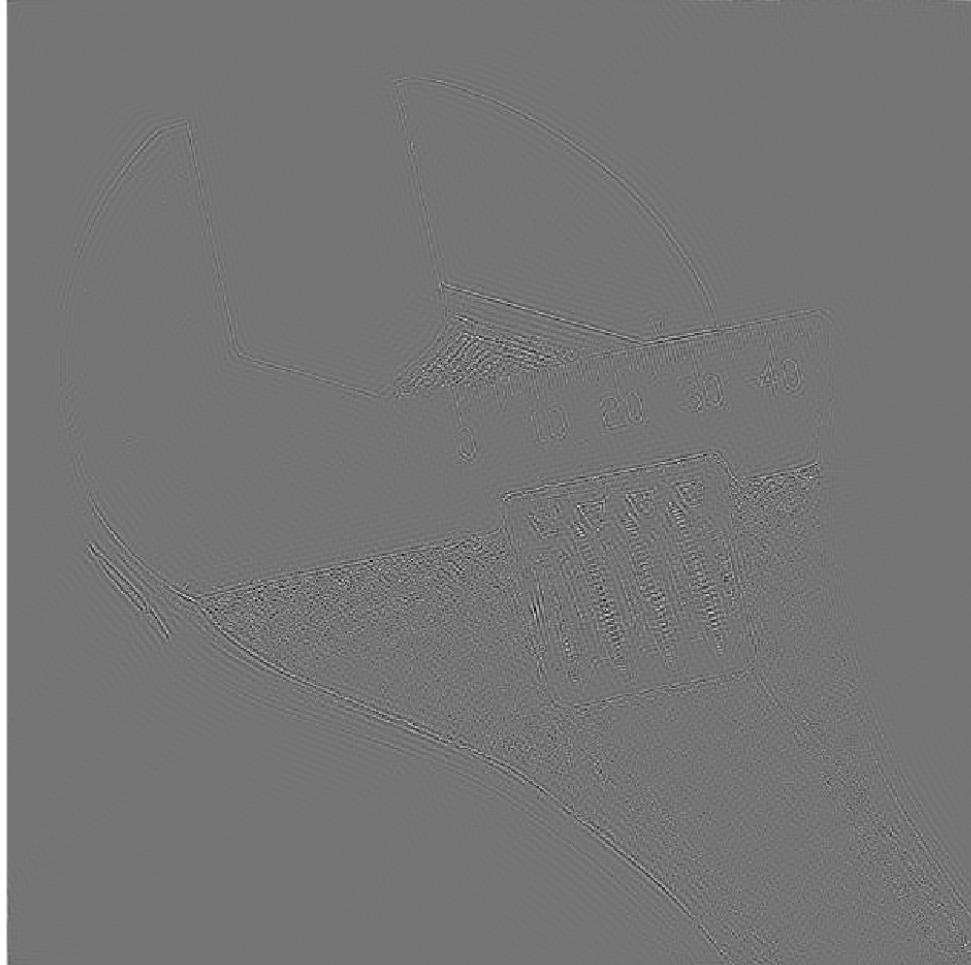
```
wrench = mat2gray(rgb2gray(imread('wrench.jpg')));  
wrench_edge = edge(wrench);  
imwrite(wrench_edge,'wrench_edge.jpeg')  
h = fspecial('sobel');
```



## 2.

```
wrench = rgb2gray(imread('wrench.jpg'));
h = fspecial('sobel');
h_f = fft2(h);
wrench_f = fft2(wrench);
wrench_filt = imfilter(double(wrench_f),h_f);
wrench_edge = ifft2(abs(wrench_filt).*exp(i*angle(wrench_filt)));
```

**ILPF, radius = 300**



```
saveas(gcf,'wrench_edge_f.png');
```

## Experiment 2

### 1 Code.

```
%ILPF
% wrench = rgb2gray(imread('wrench.jpg'));
% wrench_f = fft2(wrench);
% ILPF_m = zeros(600,600);
% for I = 1:600
%   for J = 1:600
%     if ((I-300)^2+(J-300)^2)^(1/2) < 400
%       ILPF_m(I,J) = 1;
%     else
%       ILPF_m(I,J) = 0;
%     end
%   end
% end
% wrench_f_filt = wrench_f.*ILPF_m;
% wrench_ILPF = ifft2(abs(wrench_f_filt).*exp(i*angle(wrench_f_filt)));
% imshow(wrench_ILPF,[]);
% title('ILPF, radius = 400');
% saveas(gcf,'wrench_ILPF_r400.png');

%BLPF
% wrench = rgb2gray(imread('wrench.jpg'));
% wrench_f = fft2(wrench);
% BLPF_m = zeros(600,600);
% d0=500;
% figure
% for n = 2:5
%   for I = 1:600
%     for J = 1:600
%       BLPF_m(I,J) = 1/(1+(((I-300)^2+(J-300)^2)^(1/2))/d0)^(2*n);
%     end
%   end
% wrench_f_filt = wrench_f.*BLPF_m;
% wrench_BLPF = ifft2(abs(wrench_f_filt).*exp(i*angle(wrench_f_filt)));
% imshow(wrench_BLPF,[]);
% title("Gaussian Low Pass Filter, d = 500");
% saveas(gcf,'wrench_ILPF_n2d300.png');
% plot(1:300,BLPF_m(300,301:600));
% legend('n = 2','n = 3','n = 4','n = 5');
% hold on
```

```

% end

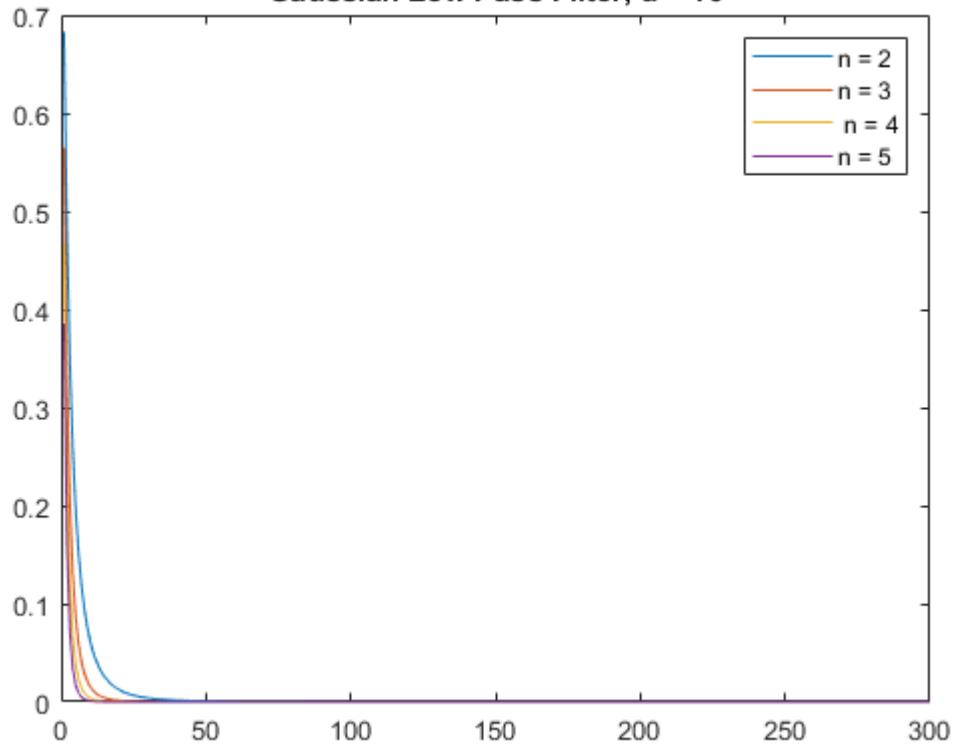
%GLPF
wrench = rgb2gray(imread('wrench.jpg'));
wrench_f = fft2(wrench);
GLPF_m = zeros(600,600);
d0=10;
figure
for n = [5 10 15 230]
    for I = 1:600
        for J = 1:600
            GLPF_m(I,J) = exp(-(((I-300)^2+(J-300)^2)^(1/2))/(2*n^2));
        end
    end
    wrench_f_filt = wrench_f.*GLPF_m;
    wrench_GLPF = ifft2(abs(wrench_f_filt).*exp(i*angle(wrench_f_filt)));
    imshow(wrench_GLPF,[]);
    title(["Gaussian Low Pass Filter, d=" + n]);
    saveas(gcf,"wrench_gLPF_d" + n + ".png");
%    plot(1:300,GLPF_m(300,301:600));
%    legend('n = 10','n = 20','n = 40','n = 100');
%    hold on
end

```

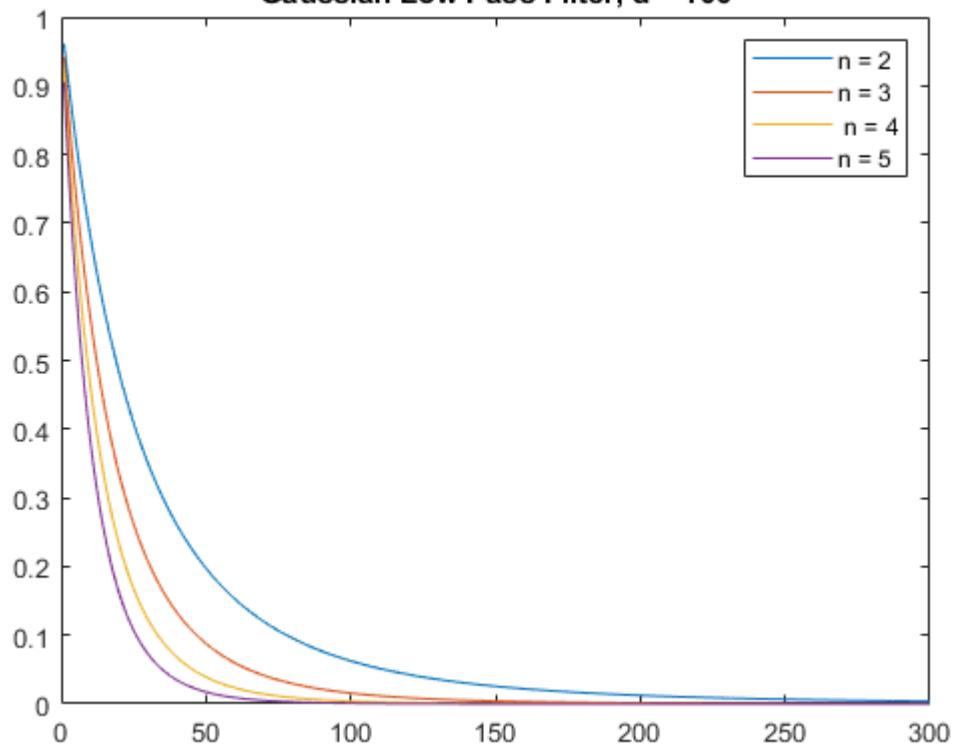
## 1 Results.

***The following 3 graphs are Butterworth LPF, not Gaussian:***

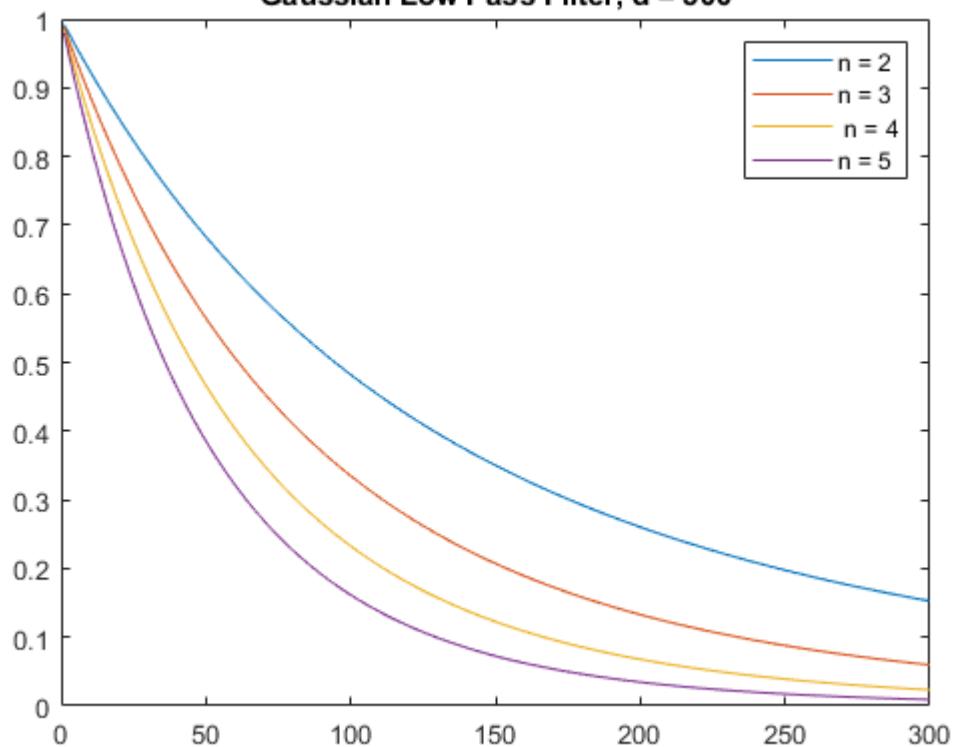
**Gaussian Low Pass Filter,  $d = 10$**



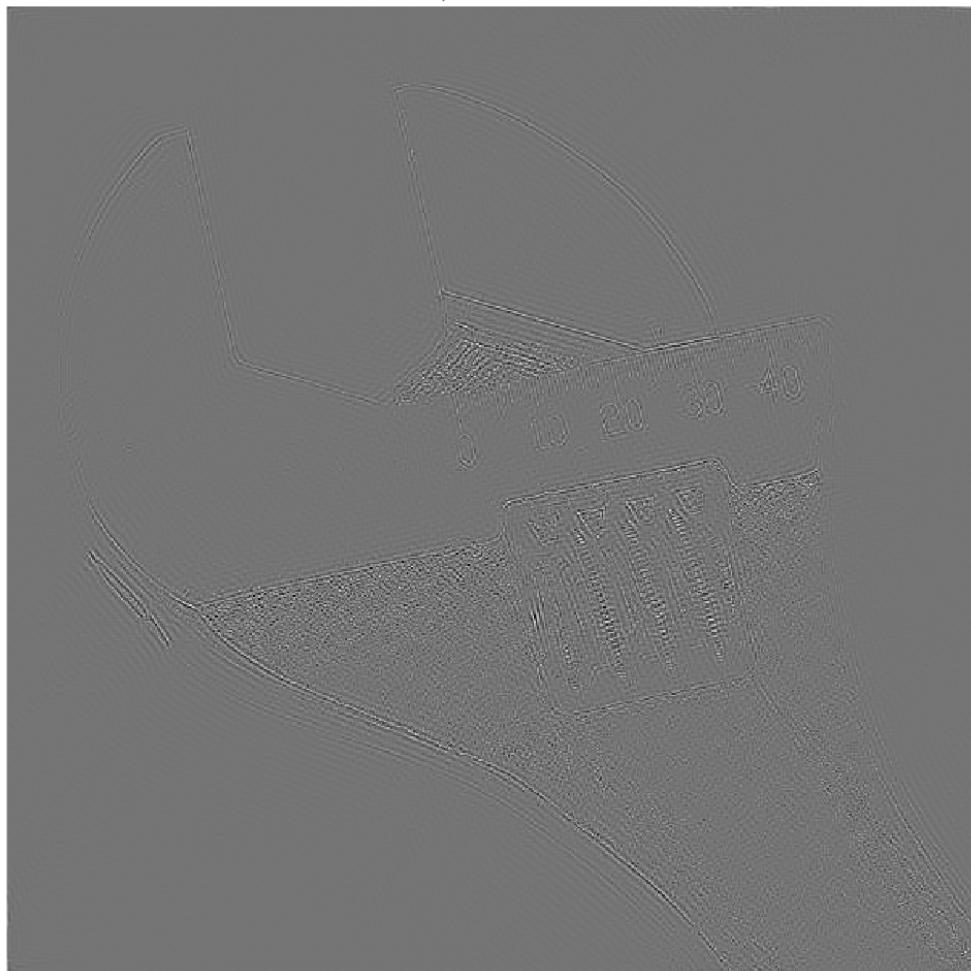
**Gaussian Low Pass Filter,  $d = 100$**



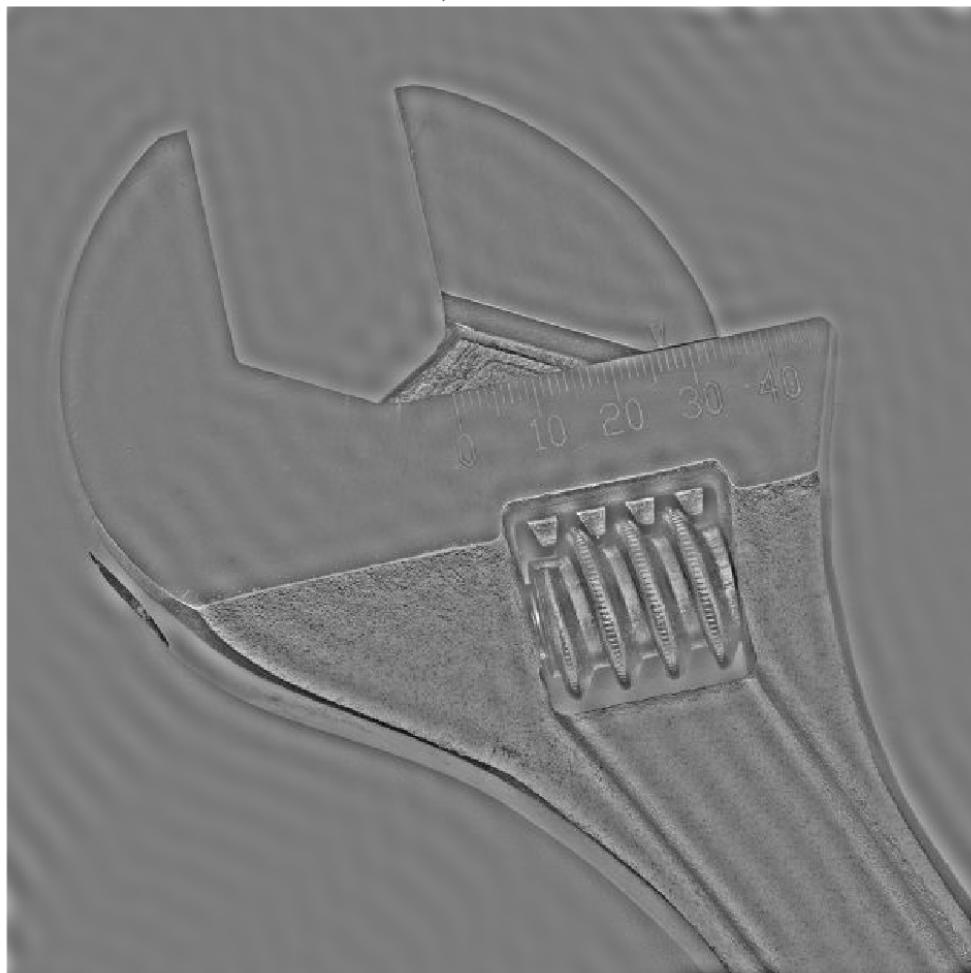
**Gaussian Low Pass Filter,  $d = 500$**



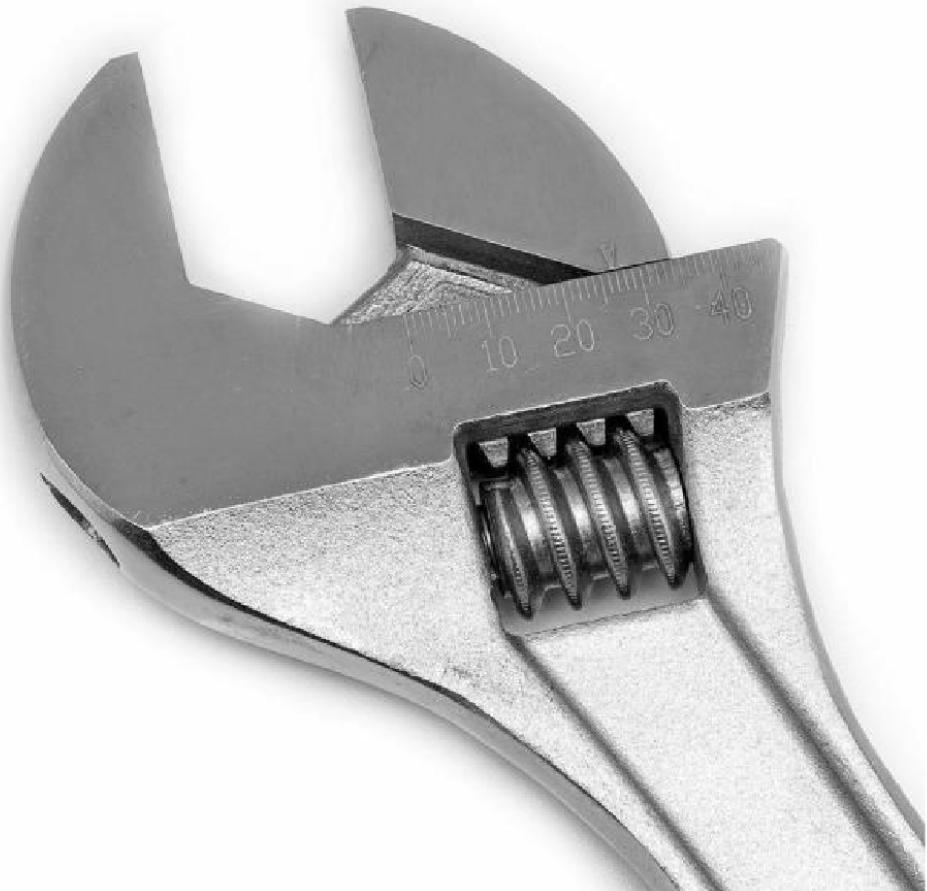
**ILPF, radius = 300**



**ILPF, radius = 400**



**ILPF, radius = 500**



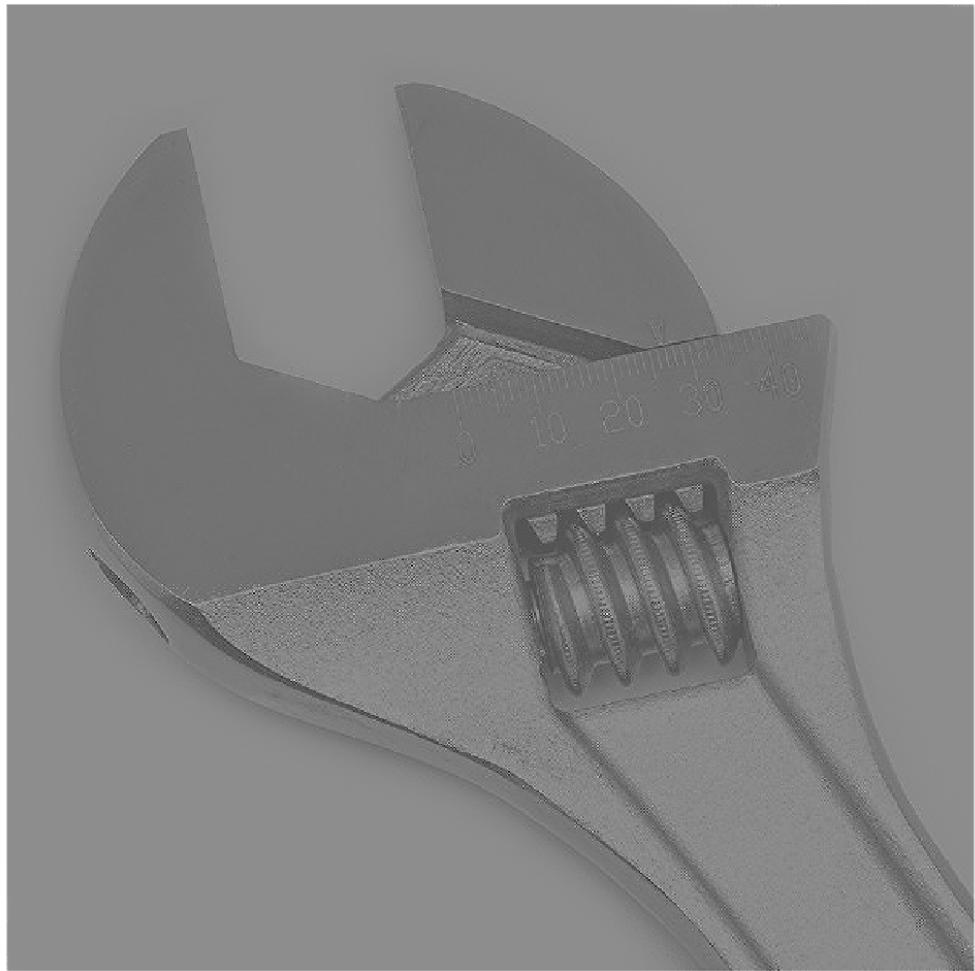
**BLPF, n = 2, d0 = 10**



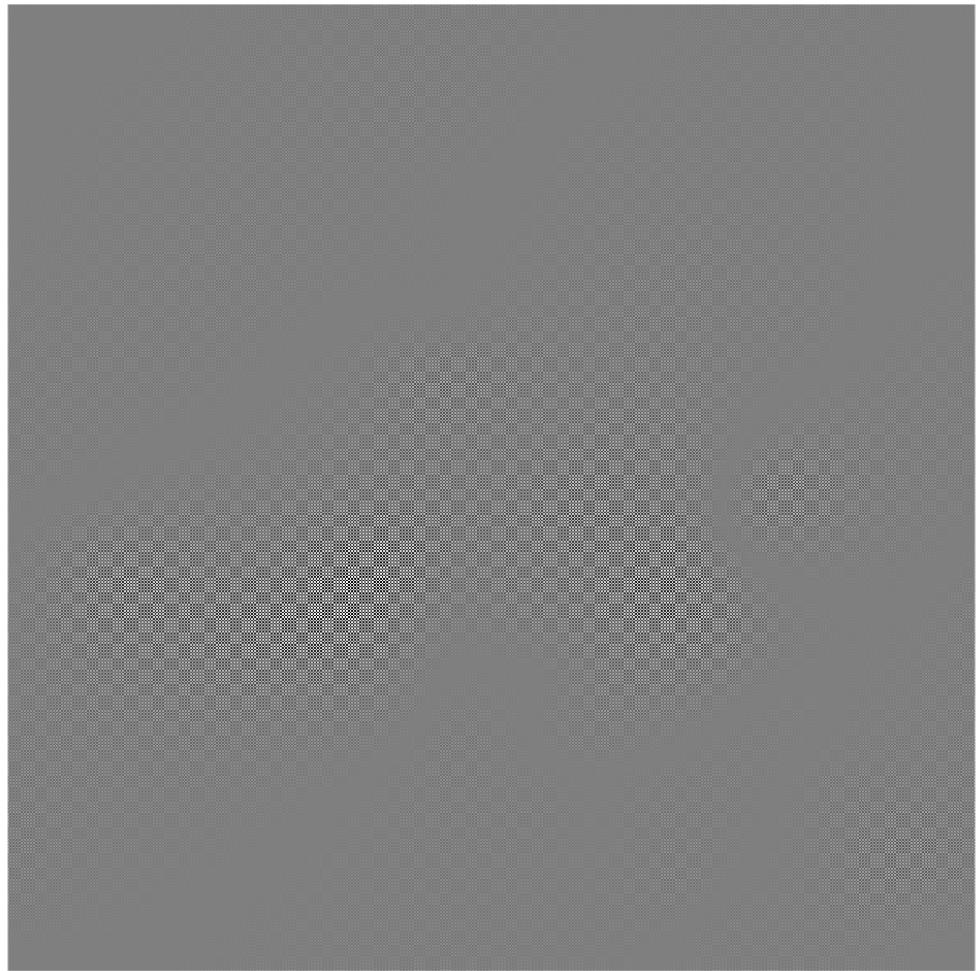
**BLPF, n = 2, d0 = 100**



**BLPF, n = 2, d0 = 300**



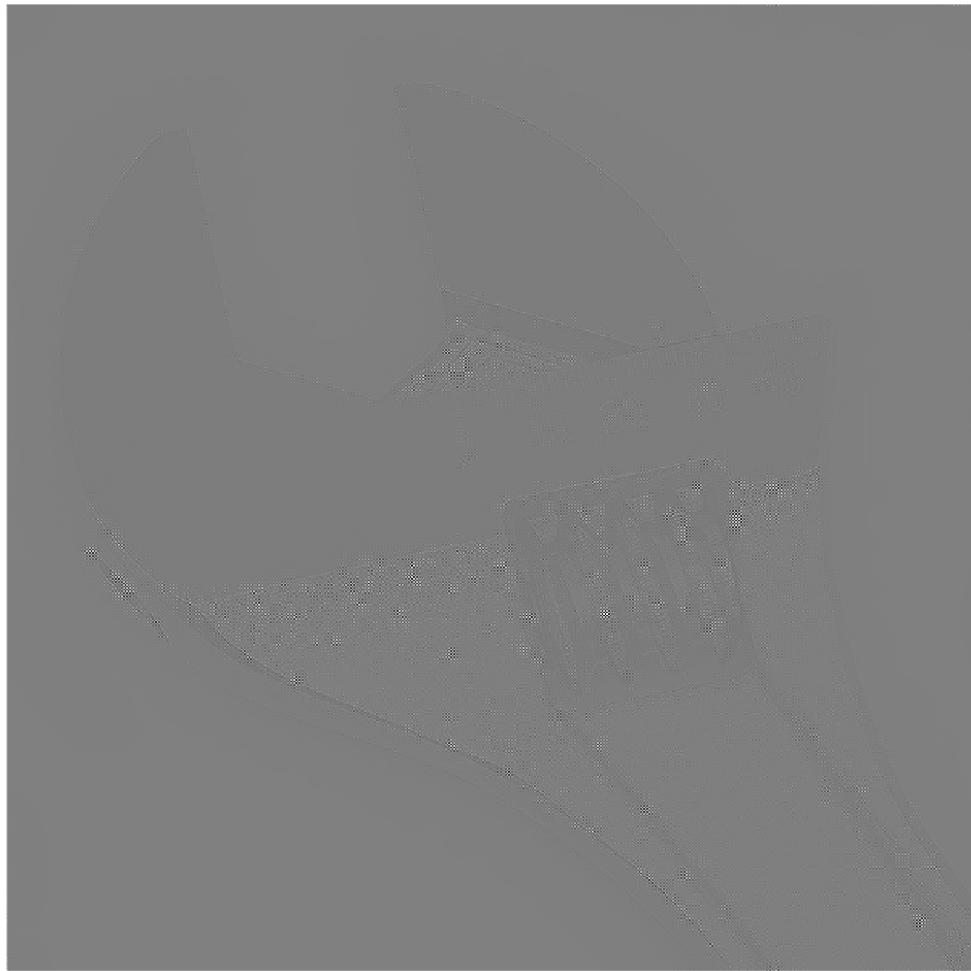
**BLPF, n = 4, d0 = 10**



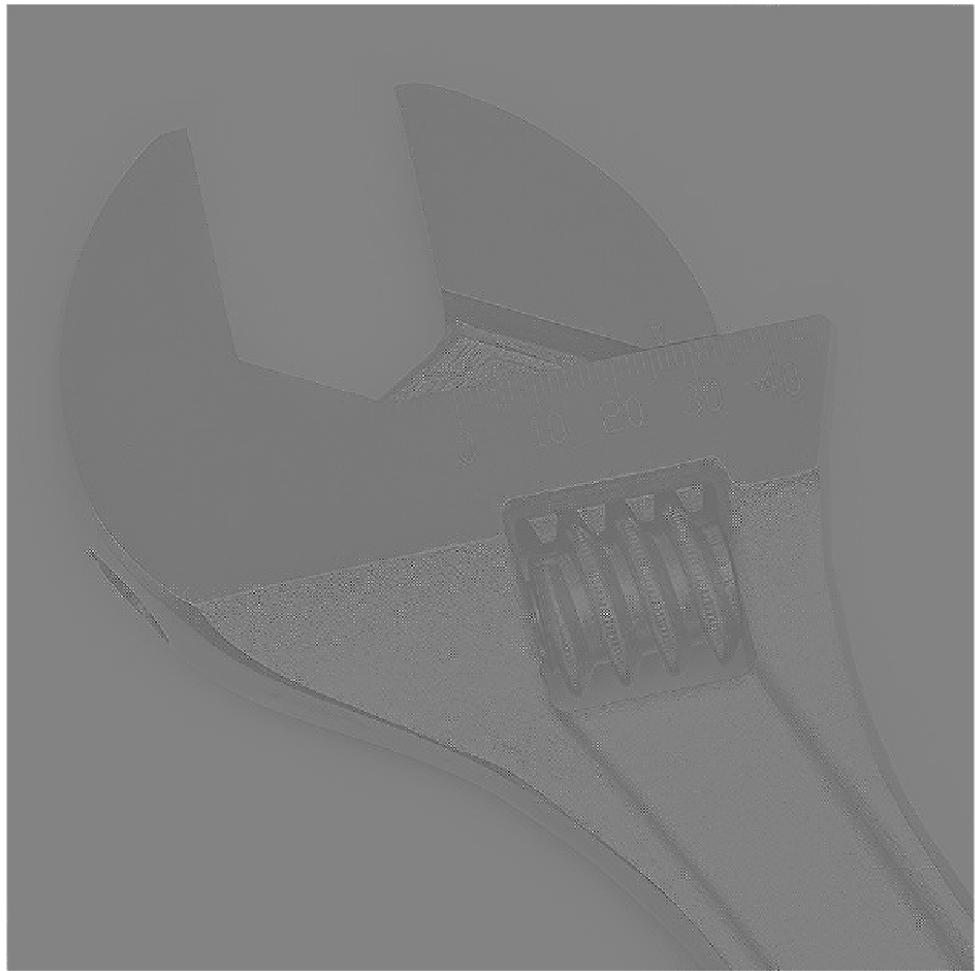
**BLPF, n = 4, d0 = 100**



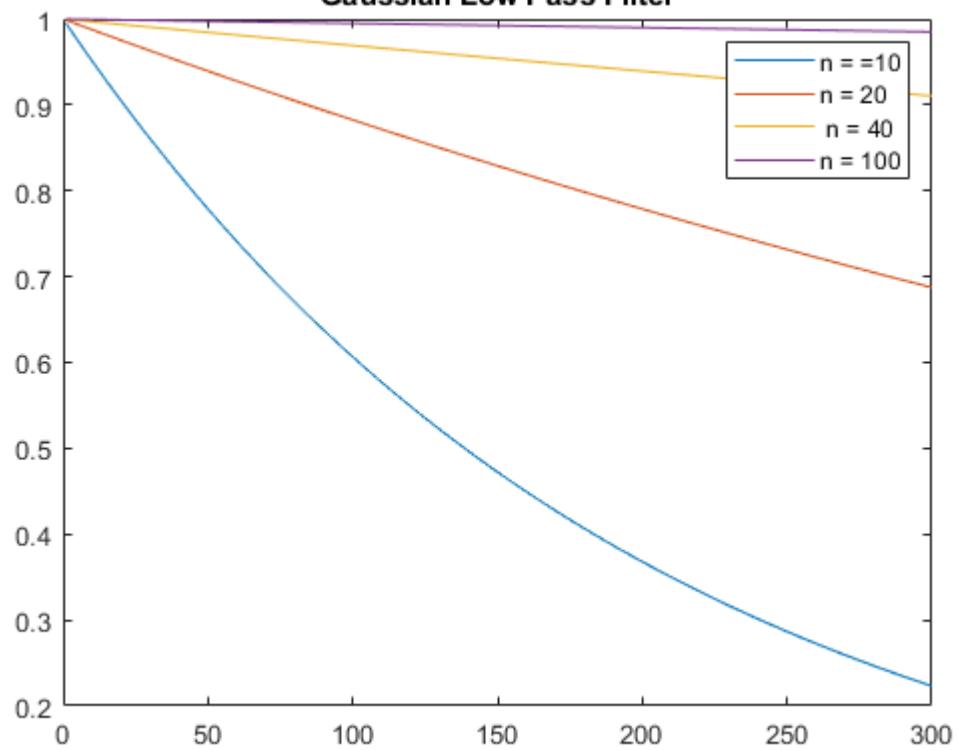
**BLPF, n = 4, d0 = 300**



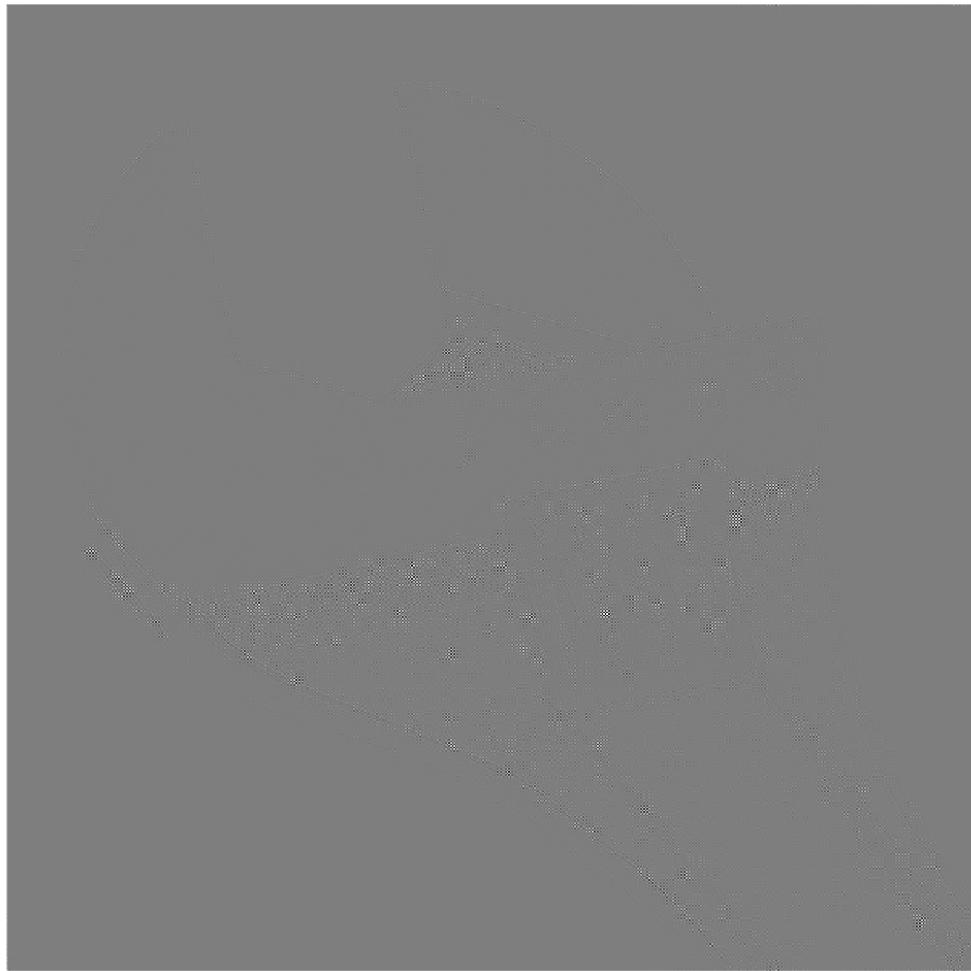
**BLPF, n = 4, d0 = 500**



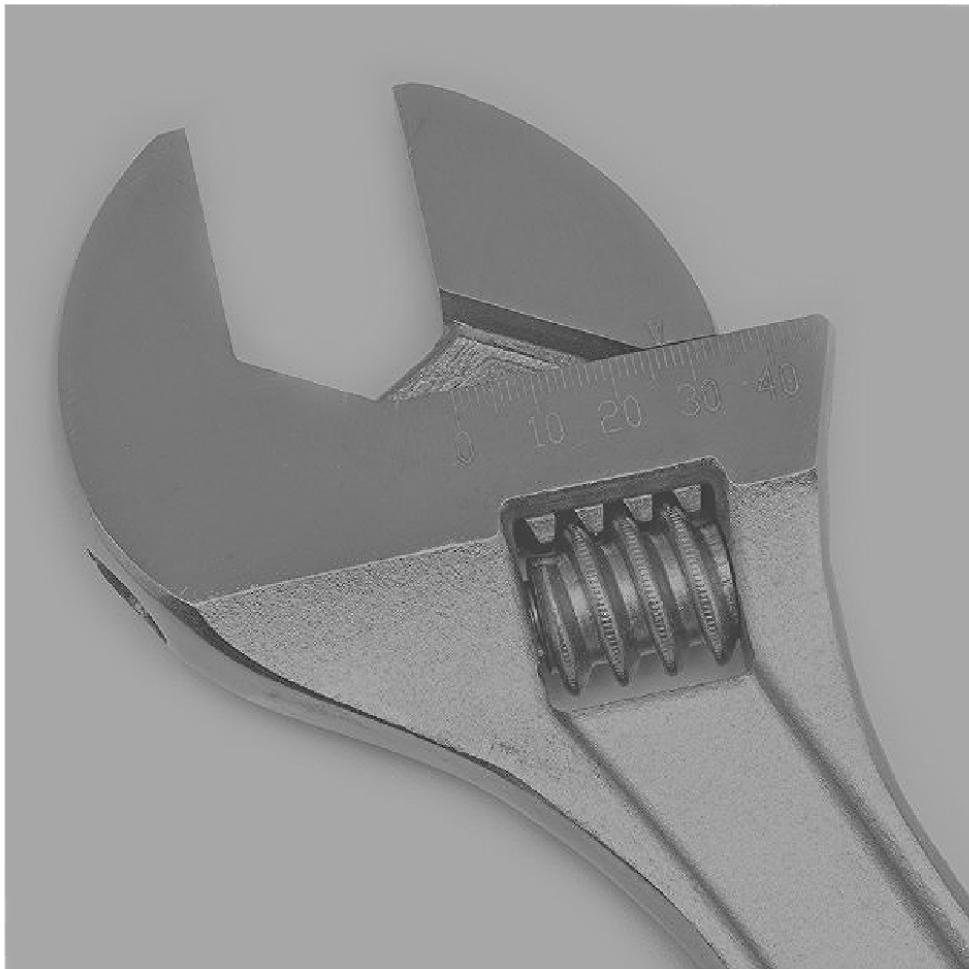
### Gaussian Low Pass Filter



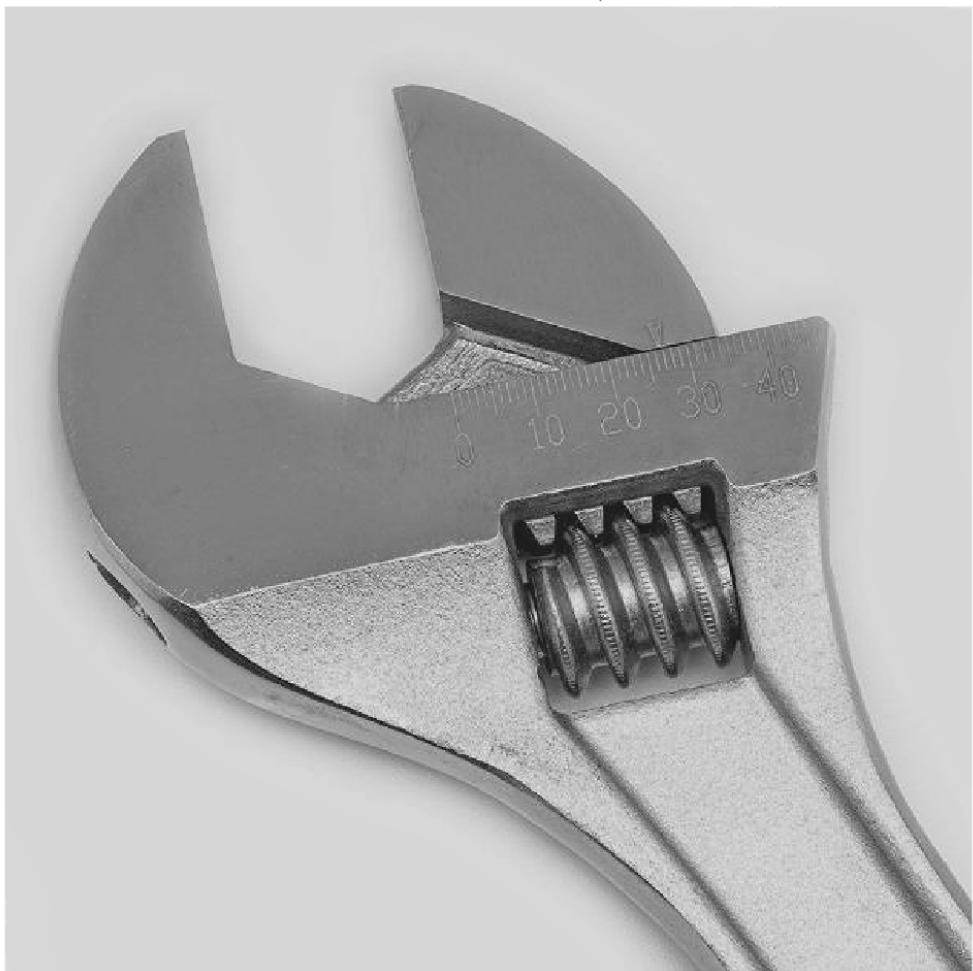
**Gaussian Low Pass Filter, d=5**



**Gaussian Low Pass Filter, d=10**



Gaussian Low Pass Filter, d=15



Gaussian Low Pass Filter, d=230



## 1 Conclusion.

The ideal lowpass filter with a radius equal to 40. The larger the bandwidth, or radius, the more the image comes through the filter. For very small radii, only the lowest frequencies get passed so the output image become blurry. The butterworth lowpass filter does have much ringing, though it has some ringing that is more shaped like a square than a round edge. It looks pixelated. The spread factor on the gaussian low pass filter outputs almost the same original image when it equals 230.

## 2. Code

```
% IHPF  
wrench = rgb2gray(imread('wrench.jpg'));  
wrench_f = fft2(wrench);  
ILPF_m = zeros(600,600);
```

```

for I = 1:600
    for J = 1:600
        if ((I-300)^2+(J-300)^2)^(1/2) < 420
            ILPF_m(I,J) = 1;
        else
            ILPF_m(I,J) = 0;
        end
    end
end
IHPF_m = 1 - ILPF_m;
wrench_h_filt = wrench_f.*IHPF_m;
wrench_HLPF = ifft2(abs(wrench_h_filt).*exp(i*angle(wrench_h_filt)));
imshow(wrench_HLPF,[]);
title('IHPF, radius = 20');
saveas(gcf,'wrench_IHPF_r20.png');
%
% %BHPF
% wrench = rgb2gray(imread('wrench.jpg'));
% wrench_f = fft2(wrench);
% BLPF_m = zeros(600,600);
% d0=580;
% figure
% for n = 2:4
%     for I = 1:600
%         for J = 1:600
%             BLPF_m(I,J) = 1/(1+(((I-300)^2+(J-300)^2)^(1/2))/d0)^(2*n);
%         end
%     end
%     BHPF_m = 1-BLPF_m;
%     wrench_f_filt = wrench_f.*BHPF_m;
%     wrench_BHPF = ifft2(abs(wrench_f_filt).*exp(i*angle(wrench_f_filt)));
%     figure
%     imshow(wrench_BHPF,[]);
%     title(["Gaussian High Pass Filter, d = 100, n = " + n]);
%     saveas(gcf,"wrench_IHPF_n2d100n" + n + ".png");
% % plot(1:300,BLPF_m(300,301:600));
% % legend('n = 2','n = 3',' n = 4','n = 5');
% hold on
% end

```

## 2. Results

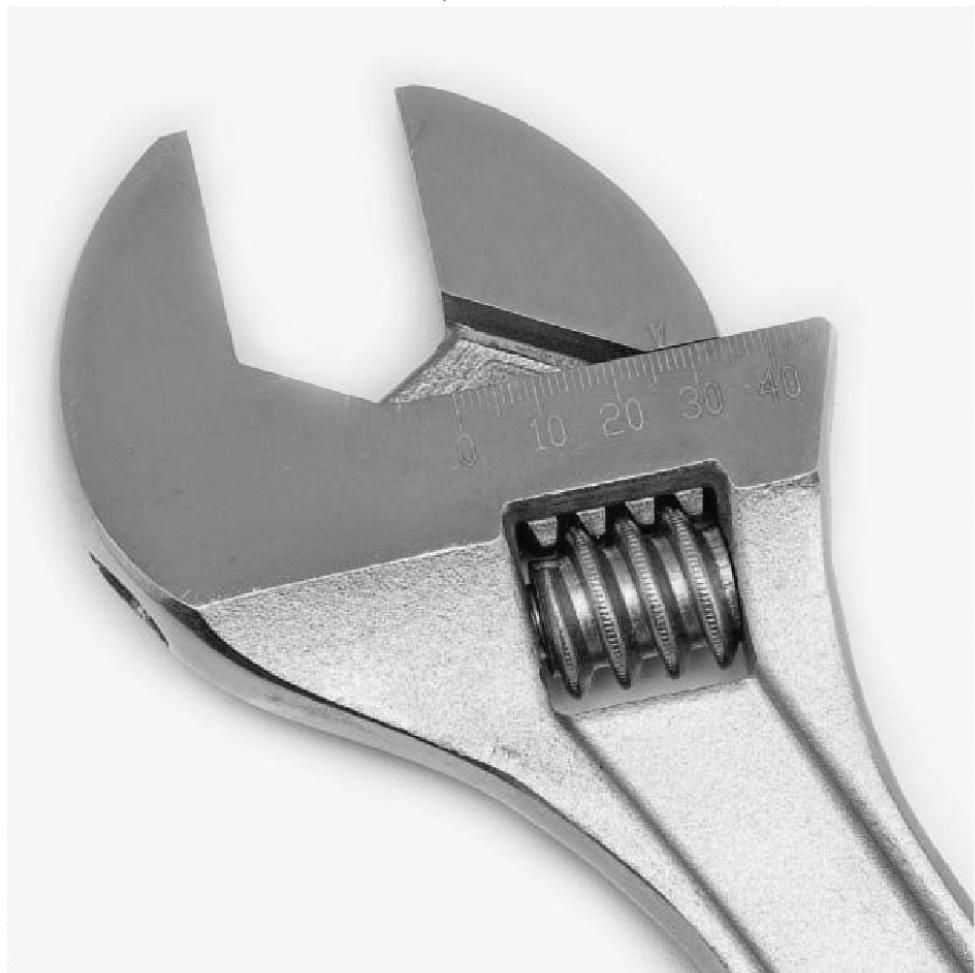
IHPF, radius = 50



IHPF, radius = 100



IHPF, radius = 200



IHPF, radius = 400



## 2. Conclusion

I was not getting good results on the high pass filter, which made me question whether I did the low pass filter correctly. I am not sure if I am suppose to apply the shift function to the FFT before applying the mask. It seemed to work well like that for the low pass filter, but then when I did the highpass filter it did not look correct to me.