

---

```

%NAME: Krishna Kodali
%INST: IIT Bhubaneswar
%DATE: 29/11/2020
%CATEGORY: BTech
%BRANCH: Computer Science
%Roll Number: 17CS01008
%Assignment-05
%Motion blurring and restoration using different techniques
%Removing previous Buffer
clc;clear;close all;

%%Question-1: To motion blur an image and to apply Inverse
    filtering,Radially limited Inverse filtering and Wiener filtering.
% on the edegraded image.

I = imread('cameraman.tif');
Image = double(I);
[M, N] = size(Image); %Size of image
P = 2*M;
Q = 2*N;
pad_image = zeros(P,Q);
pad_image(1:M,1:N)=Image;

F = (fft2(pad_image));
H = zeros(P,Q);
T = 2; a = 0.01; b = 0.01;

for u=1:P
    for v = 1:Q
        V = pi*(u*a+v*b);
        H(u,v) = T/V * sin(V) * exp(-V*1j);
    end
end

G = zeros(P,Q);
for u=1:P
    for v = 1:Q
        G(u,v) = H(u,v)*F(u,v);
    end
end

g = ifft2((G));
real_g = real(g);

output = real_g(1:M,1:N);

figure;
subplot(2,2,1);imshow(uint8(output));title('Blurred Image');

Fcap = zeros(P,Q);
for u=1:P
    for v = 1:Q

```

---

---

```

        Fcap(u,v) = G(u,v)/H(u,v);
    end
end

Gcap = ifft2((Fcap));
real_gcap = real(Gcap);
output_cap = real_gcap(1:M,1:N);
subplot(2,2,2);imshow(uint8(output_cap));title('Inverse Filtered
Image');
D = zeros(P,Q);

for u=1:P
    for v = 1:Q
        D(u,v) = sqrt((u-(P/2))^2+(v-(Q/2))^2);
    end
end

GPF = zeros(P,Q);
D0 = 500;
for u=1:P
    for v = 1:Q
        GPF(u,v) = exp(-1*(D(u,v)^2)/(2*D0^2));
    end
end

Fcap_radial = zeros(P,Q);
for u=1:P
    for v = 1:Q
        Fcap_radial(u,v) = GPF(u,v)*Fcap(u,v);
    end
end

Gcap_Radical = ifft2((Fcap_radial));
R_Gcap_radial = real(Gcap_Radical);
Output_radial = R_Gcap_radial(1:M,1:N);
subplot(2,2,3);imshow(uint8(Output_radial));title('Radial Inverse
Filtered Image');

Hconj = conj(H);
MagH = zeros(P,Q);
for u=1:P
    for v = 1:Q
        MagH(u,v) = Hconj(u,v)*H(u,v);
    end
end

Weiner = zeros(P,Q);
K = 0; %No noise in the org image
for u=1:P
    for v = 1:Q
        Weiner(u,v) = 1/H(u,v) * MagH(u,v)/(MagH(u,v)+K) * G(u,v);
    end
end

```

---

---

```

W = ifft2((Weiner));
real_w = real(W);
output_weiner = real_w(1:M,1:N);
subplot(2,2,4);imshow(uint8(output_weiner));title('Weiner corrected
Image');

```

**Blurred Image**



**Inverse Filtered Image**



**Radial Inverse Filtered Image**



**Weiner corrected Image**



*%Question-2: Take an originally blurred image and apply inverse filtering, radial inverse filtering and % weiner filtering on the original image.*

```

I = rgb2gray(imread('blurred_cameraman.tif'));
Image = double(I);
[M, N] = size(Image); %Size of image
P = 2*M;
Q = 2*N;
pad_image = zeros(P,Q);
pad_image(1:M,1:N)=Image;
F = (fft2(pad_image));
H = zeros(P,Q);
T = 2; a = 0.01; b = 0.01;

for u=1:P
    for v = 1:Q
        V = pi*(u*a+v*b);
        H(u,v) = T/V * sin(V) * exp(-V*1j);
    end
end

```

---

```

end

figure;
subplot(2,2,1);imshow(I);title('Original Image');

Fcap = zeros(P,Q);
for u=1:P
    for v = 1:Q
        Fcap(u,v) = F(u,v)/H(u,v);
    end
end

gcap = ifft2((Fcap));
real_gcap = real(gcap);
output_cap = real_gcap(1:M,1:N);
subplot(2,2,2);imshow(uint8(output_cap));title('Inverse Filtered
Image');

D = zeros(P,Q);
for u=1:P
    for v = 1:Q
        D(u,v) = sqrt((u-(P/2))^2+(v-(Q/2))^2);
    end
end

GPF = zeros(P,Q);
D0 = 400;
for u=1:P
    for v = 1:Q
        GPF(u,v) = exp(-1*(D(u,v)^2)/(2*D0^2));
    end
end

Fcap_radial = zeros(P,Q);
for u=1:P
    for v = 1:Q
        Fcap_radial(u,v) = GPF(u,v)*Fcap(u,v);
    end
end

Gcap_Radical = ifft2((Fcap_radial));
R_Gcap_radial = real(Gcap_Radical);
Output_radial = R_Gcap_radial(1:M,1:N);
subplot(2,2,3);imshow(uint8(Output_radial));title('Radial Inverse
Filtered Image');

Hconj = conj(H);
MagH = zeros(P,Q);
for u=1:P
    for v = 1:Q
        MagH(u,v) = Hconj(u,v)*H(u,v);
    end
end
end

```

---

---

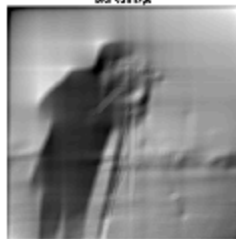
```

Weiner = zeros(P,Q);
K =0.01;
for u=1:P
    for v = 1:Q
        Weiner(u,v) = 1/H(u,v) * MagH(u,v)/(MagH(u,v)+K) * F(u,v);
    end
end

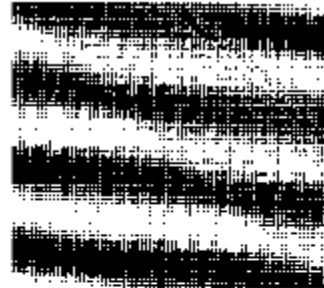
W = ifft2((Weiner));
real_w = real(W);
output_weiner = real_w(1:M,1:N);
subplot(2,2,4);imshow(uint8(output_weiner));title('Weiner corrected
Image');

```

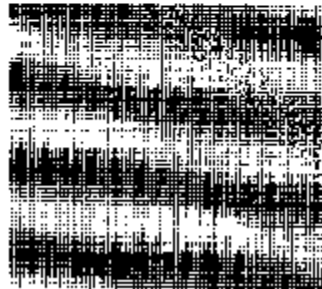
**Original Image**



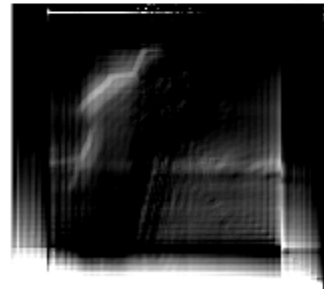
**Inverse Filtered Image**



**Radial Inverse Filtered Image**



**Weiner corrected Image**



*Published with MATLAB® R2020a*