

```

%%                                     Non Local Means - Edge Preserving Filter
%   V Khagesh Kumar
%   170102070
%   ECE
%   This program will take 2 minutes to compute.

clear;

%% Read Image
original_image = imread('lenna.noise.jpg');
image = double(original_image);

%% Set parameters
patch = 7; %For Gaussian weighted Sum of Square Differences (GSSD) between the patches
f = floor(patch/2);
search_patch = 5;
ff = floor(search_patch/2);
sigma_gssd = 10;
h = 30; %variance for weights

%% Weights for GSSD
W_d = zeros(patch,patch);
for i=1:patch
    for j=1:patch
        W_d(i,j) = exp( -(sum(abs([i,j]-[f+1,f+1]))^2)/(2*sigma_gssd*sigma_gssd) );
    end
end
W_d = W_d/(sum(sum(W_d)));

%% Non-Local Means
image = padarray(image,[f,f],'replicate');
[m,n] = size(image);

final_image = original_image;
for i=1+f+ff:m-f-ff
    for j=1+f+ff:n-f-ff
        W = weights(image,i,j,patch,search_patch,h,W_d);
        final_image(i-ff-1,j-ff-1) = sum(sum(W.*image(i-ff:i+ff,j-ff:j+ff)));
    end
end

final_image = uint8(final_image);
figure;
imshow(original_image);
figure;
imshow(final_image);

imwrite(final_image, 'output_lenna.noise.jpg', 'JPG');

%% Weights
function result = weights(image,x,y,patch,search_patch,h,W_d)
    f = floor(patch/2);
    ff = floor(search_patch/2);

```

```

result = zeros(search_patch,search_patch);
for i=x-ff:x+ff
    for j=y-ff:y+ff
        mod_d = GSSD(image(x-f:x+f,y-f:y+f), image(i-f:i+f,j-f:j+f),W_d);
        result(i-x+ff+1,j-y+ff+1) = exp(-mod_d/(h*h));
    end
end
result = result/sum(sum(result));
end

```

%% Gaussian weighted Sum of Square Differences (GSSD) between the patches

```

function result = GSSD(image, similar_image,W_d)
    image = (image-similar_image).^2;
    %result = sum(sum(exp(-image/(2*sigma_gssd*sigma_gssd))));
    result = sum(sum(W_d.*image));
end

```