

PATCH BASED FILTERING

1. CODE

```
function[new_image,blurred_image,changed_orig_image,rmsd]=myPatchBasedFiltering(orig_image,h)
window_size=25;
w=(window_size-1)/2;
patch_size=9;
p=(patch_size-1)/2;

%imwrite(orig_image,'../images/original_image.png');
H = fspecial('gaussian',3,0.66);
image = imfilter(orig_image,H,'replicate');
%imwrite(image,'../images/blurred_image.png');
image=myShrinkImageByFactorD(image,2);
changed_orig_image=myShrinkImageByFactorD(orig_image,2);

blurred_image=image;
[row,col]=size(image);

SIGMA=0.1*ones(2,2);
new_image=image;
num_patches=10;

for i=(1+p):(row-p)
    i
    xmin=max(i-w,1);
    xmax=min(i+w,row);
    for j=(1+p):(col-p)
        ymin=max(j-w,1);
        ymax=min(j+w,col);
        window=image(xmin:xmax,ymin:ymax);
        MU=[i j];
        patch1=image(i-p:i+p,j-p:j+p);
        total=0;
        weight=zeros(num_patches,1);
        for iter=1:num_patches
            q=mvnrnd(MU,SIGMA);
            i1=round(q(1));
            j1=round(q(2));
            i1=max(i1,xmin+p);
            i1=min(i1,xmax-p);
            j1=max(j1,ymin+p);
            j1=min(j1,ymax-p);
            patch2=image(i1-p:i1+p,j1-p:j1+p);
            weight(iter,1)=exp(-(norm(patch1-patch2)^2)/(h^2));
            total=total+weight(iter,1)*image(i1,j1);
        end
        new_image(i,j)=total/sum(sum((weight)));
    end
end
```

```

end
% subplot(1,3,1), imshow(old_image)
% subplot(1,3,2), imshow(image)
% subplot(1,3,3), imshow(new_image)
rmsd=sqrt(norm(final_image-changed_orig_image)^2/(row*col));
%imwrite(new_image,'../images/patch_based');

```

2. Final result:



3. Filter mask:

4. Optimal value of sigma= 0.1

RMSD optimum value= 0.0037

5. RMSD at sigma=0.09: 0.0038

RMSD at sigma=0.11: 0.0038