**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

% subplot(1,3,1), imshow(old\_image)

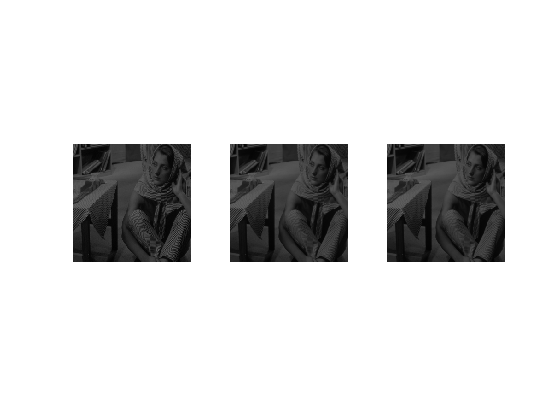
% subplot(1,3,2), imshow(image)

% subplot(1,3,3), imshow(new\_image)

rmsd=sqrt(norm(final\_image-changed\_orig\_image)^2/(row\*col));

%imwrite(new\_image,'../images/patch\_based');

1. Final result:



1. Filter mask:
2. Optimal value of sigma= 0.1

RMSD optimum value= 0.0037

1. RMSD at sigma=0.09: 0.0038

RMSD at sigma=0.11: 0.0038