

Introduction to Image Processing

HW4

Due: 12/15

HW4.1 Image interpolation through spatial filtering

Write a function to implement 2X image enlargement using zero-interleaving and spatial filtering. The function should have the syntax

`imenlarge(image, filt)`

where `image` is the input and `filt` is the filter to use.

For example, the command

`imenlarge(head, bicubic_filt)`

would enlarge an image “head” to 2 times its size,

using the 5×5 bicubic filtering.

$$\frac{1}{64} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

HW4.2 Image interpolation in frequency domain

By convolution property, the spatial filtering between an image M and a filter S can be realized as

$$M * S = \mathcal{F}^{-1}(\mathcal{F}(M) \cdot \mathcal{F}(S'))$$

where \mathcal{F} and \mathcal{F}^{-1} denote, respectively, the DFT and the inverse DFT. S' is the zero-padded version of S to make it have the same size as M .

Write a function to implement 2X image enlargement in frequency domain. The function should have the syntax

`imenlarge_fre(image, filt)`

where `image` is the input and `filt` is the filter to use.

A zero-interleaving is used to initialize the enlarged image.

For example, the command

`imenlarge_fre(head, bicubic_filt)`

would enlarge an image “head” to 2 times its size, using the 5×5 bicubic filtering.

Note:

1. You need to include a test.m to verify this function where a PSNR is computed between the original image and the rescaled back image.
2. Test on two filters.
3. Test on two images (one has complex textures)
4. Compare the results in HW4.1 and HW4.2

test.m

```
Img=imread('road.png');
imshow(Img);title('original image');

% define the used filter
filter_bic=1/64*[1 4 6 4 1; 4 16 24 16 4; 6 24 36 24 6; 4 16 24 16 4; 1 4 6 4 1];

% enlarge the image
Img =rgb2gray(Img);
Img_enl_S = imenlarge(Img, filter_bic); % image enlargement by spatial filtering
Img_enl_F = imenlarge_fre(Img, filter_bic); % image enlargement by filtering in frequency domain
figure
montage({Img_enl_S, Img_enl_F}), title('enlarged image by spatial filtering (left) vs enlarged image by
filtering in frequency domain');

% rescale the enlarged image
Img_re_S = imresize(Img_enl_S, 1/2, 'bicubic');
Img_re_F = imresize(Img_enl_F, 1/2, 'bicubic');
figure
montage({Img_re_S, Img_re_F}), title('rescaled image of spatial filtered image (left) vs rescaled image of
frequency filtered image');

% compute the PSNR between the original image and the image resclaed back
PSNR_1 = psnr(Img, Img_re_S);
PSNR_2 = psnr(Img, Img_re_F);
```