Cvičení 5 - Jasové transformace III, Filtrování

Práce přímo s barevnými pixely

Transformace na barevných obrázcích z minulého cvičení byly definovány tak, že výsledná hodnota jedné barevné složky závisí pouze na této barevné složce (výsledná červená složka zívisí jen na červené složce vstupního obrazu). Některé jasové transformace ale takto definovat nemůžeme.

f_r, f_g, f_b ... barevné složky vstupního obrazu.

g_r, g_g, g_b ... barevné složky výstupního obrazu

Transformace:

g_r = a11 f_r + a12 f_g + a13 f_b

g_g = a21 f_r + a22 f_g + a23 f_b

g_b = a31 f_r + a32 f_g + a33 f_b

(koeficienty můžeme zapsat do matice)

Sepia

$$\begin{bmatrix} 0.393 & 0.349 & 0.272 \\ 0.769 & 0.686 & 0.534 \\ 0.189 & 0.268 & 0.131 \end{bmatrix}$$

```
f = im2double(imread("pastelky.png"));
g = f;

T = [0.393 0.349 0.272;
     0.769 0.686 0.534;
     0.189 0.268 0.131];

%cervena slozka
g(:,:,1) = T(1,1) * f(:,:,1) + T(2,1) * f(:,:,2) + T(3,1) * f(:,:,3);
g(:,:,2) = T(1,2) * f(:,:,1) + T(2,2) * f(:,:,2) + T(3,2) * f(:,:,3);
g(:,:,3) = T(1,3) * f(:,:,1) + T(2,3) * f(:,:,2) + T(3,3) * f(:,:,3);

figure, imshow(g);
```

Úkol 1

Jak by vypadala transformace. které přehazuje červenou a modrou složku (RGB -> BGR)? Naprogramujte jí.

```
```

Filtrování

Na rozdíl od operací jasových (z minulého cvičení), novou hodntu pixelu na souřadnicích [x,y] spočítáme z hodnot v jeho okolí. Postup aplikace filtru se nazývá korelace (případně konvoluce). Pro jednoduchost se obrázek zvětšuje doplněním 0 okolo (zero padding), aby se maska korelace dala aplikovat i na okrajové pixely.

Korelace / Konvoluce

Korelace

pomocí funkce `imfilter()`

Vstup: obrázek, metoda (konvoluce/korelace), informace o velikosti výsledného obrázku

Více informací:

`help imfilter`

```
f = [0 0 0 1 0 0 0];
```

```
w = [1 2 3 2 8];

g = imfilter(f,w,'corr','same')
```

```
g = 1×7
     0     8     2     3     2     1     0
```

Konvoluce

Srovnejte s výsledky korelace

```
f = [0 0 0 1 0 0 0];
w = [1 2 3 2 8];
g = imfilter(f,w,'conv','same')
```

```
g = 1×7
     0     1     2     3     2     8     0
```

Korelace dvourozměrných dat

```
f = [ 0 0 0 0 0;
       0 0 0 0 0;
       0 0 1 0 0;
       0 0 0 0 0;
       0 0 0 0 0];

w = [ 1 2 3;
      4 5 6;
      7 8 9];
g = imfilter(f,w,'corr','same')
```

```
g = 5×5
     0     0     0     0     0
     0     9     8     7     0
     0     6     5     4     0
     0     3     2     1     0
     0     0     0     0     0
```

Konvoluce ve 2D

Opět srovnejte s korelací

```
g = imfilter(f,w,'conv','same')
```

```
g = 5×5
     0     0     0     0     0
     0     1     2     3     0
     0     4     5     6     0
     0     7     8     9     0
     0     0     0     0     0
```

Vyhlazovací filtry

Filtrace průměrováním

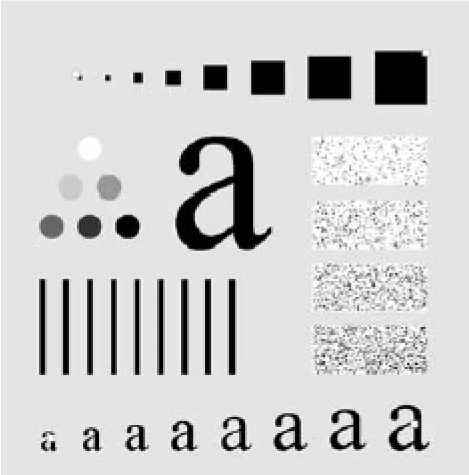
```
B = imread('lenagraysum.bmp'); % obrazek obsahuje Gaussovsky sum
```

```
% ones vytvori matici 3x3 samych 1.
w = (1/9) * ones(3);

% pripadne
% w = 1/9 * [1 1 1;
%            1 1 1;
%            1 1 1];

C = imfilter(B,w,'corr','same');
figure,
subplot(1, 2, 1), imshow(B);
title('Puvodni obrazek');
subplot(1, 2, 2), imshow(C);
title('Upraveny obrazek');
```

Úkol 2

Vytvořte průměrovací filtry s velikostmi 3x3, 4x4, 5x5, 7x7 a 10x10 a porovnejte mezi sebou výsledky pro následující obrázek

```
I = imread('a.png');
figure, imshow(I);
```



Vážené průměrování

Prvky, které jsou blíže, mají větší váhu na výsledek. Součet hodnot v matici musí být roven 1.

```
B = imread('lenagraysum.bmp');
w = 1/16 * [1 2 1;
            2 4 2;
            1 2 1];
C = imfilter(B,w,'corr','same');

figure,
subplot(1, 2, 1), imshow(B);
title('Puvodni obrazek');
subplot(1, 2, 2), imshow(C);
title('Upraveny obrazek');
```

Úkol 3

Vytvořte fitry pro vážené průměrování velikosti 3x3, 5x5, 7x7 představující vážené průměrování. Váhy jsou závislé na vzdálenosti od středu. Porovnejte výsledky mezi sebou a také s výsledky z Úkolu 1.

Mediánová filtrace

Mediánová filtrace je nelineární operace, není možné najít jednu matici, která by se aplikovala jako filtr.

V matlabu

`medfilt()`

Vstup: obrázek a velikost okolí.

```
B = imread('lenagraysum2.bmp'); % obrazek obsahuje sum typu Sul a pepr
C = medfilt2( B,[3 3] );

figure,
subplot(1,2,1), imshow(B);
subplot(1,2,2), imshow(C);
```

Percentilová filtrace (statistické filtry)

V matlabu

`ordfilt2()`

Vstup: obrázek, pořadí (kolikátý nejmenší prvek) a velikost okolí.

```
B = imread('lenagraysum2.bmp'); % obrazek obsahuje sum typu Sul a pepr
C1 = ordfilt2(B, 1, ones(3)); % min filtr
C2 = ordfilt2(B, 9, ones(3)); % maxfiltr

figure,
subplot(1,3,1), imshow(B);
subplot(1,3,2), imshow(C1);
title('min filtr');
subplot(1,3,3), imshow(C2);
title('max filtr');
```

Srovnání průměrovací a mediánové filtrace

```
w = 1/9 * [1 1 1;
           1 1 1;
           1 1 1];
C = imfilter(B,w,'corr','same');

C2 = medfilt2( B,[3 3] );

figure
subplot(1,2,1)
imshow(C,[])
title('prumer')
subplot(1,2,2)
imshow(C2,[])
title('median')
```

prumer

median

Filtry

Filtry je možné vytvořit pomocí funkce `fspecial()`. V předchozích příkladech tuto funkci ale nepoužívejte.

```
help fspecial
```

```
fspecial Create predefined 2-D filters.
   H = fspecial(TYPE) creates a two-dimensional filter H of the
   specified type. Possible values for TYPE are:

     'average'   averaging filter
     'disk'      circular averaging filter
     'gaussian'  Gaussian lowpass filter
     'laplacian' filter approximating the 2-D Laplacian operator
     'log'       Laplacian of Gaussian filter
     'motion'    motion filter
     'prewitt'   Prewitt horizontal edge-emphasizing filter
     'sobel'     Sobel horizontal edge-emphasizing filter

   Depending on TYPE which can be a string or a char vector, fspecial may
   take additional parameters which you can supply.  These parameters all
   have default values.

   H = fspecial('average',HSIZE) returns an averaging filter H of size
   HSIZE. HSIZE can be a vector specifying the number of rows and columns in
   H or a scalar, in which case H is a square matrix.
   The default HSIZE is [3 3].

   H = fspecial('disk',RADIUS) returns a circular averaging filter
   (pillbox) within the square matrix of side 2*RADIUS+1.
   The default RADIUS is 5.
```

H = **fspecial**('gaussian',HSIZE,SIGMA) returns a rotationally symmetric Gaussian lowpass filter of size HSIZE with standard deviation SIGMA (positive). HSIZE can be a vector specifying the number of rows and columns in H or a scalar, in which case H is a square matrix. Not recommended. Use imgaussfilt or imgaussfilt3 instead.
The default HSIZE is [3 3], the default SIGMA is 0.5.

H = **fspecial**('laplacian',ALPHA) returns a 3-by-3 filter approximating the shape of the two-dimensional Laplacian operator. The parameter ALPHA controls the shape of the Laplacian and must be in the range 0.0 to 1.0.
The default ALPHA is 0.2.

H = **fspecial**('log',HSIZE,SIGMA) returns a rotationally symmetric Laplacian of Gaussian filter of size HSIZE with standard deviation SIGMA (positive). HSIZE can be a vector specifying the number of rows and columns in H or a scalar, in which case H is a square matrix.
The default HSIZE is [5 5], the default SIGMA is 0.5.

H = **fspecial**('motion',LEN,THETA) returns a filter to approximate, once convolved with an image, the linear motion of a camera by LEN pixels, with an angle of THETA degrees in a counter-clockwise direction. The filter becomes a vector for horizontal and vertical motions. The default LEN is 9, the default THETA is 0, which corresponds to a horizontal motion of 9 pixels.

H = **fspecial**('prewitt') returns 3-by-3 filter that emphasizes horizontal edges by approximating a vertical gradient. If you need to emphasize vertical edges, transpose the filter H: H'.

```
    [1 1 1;0 0 0;-1 -1 -1].
```

H = **fspecial**('sobel') returns 3-by-3 filter that emphasizes horizontal edges utilizing the smoothing effect by approximating a vertical gradient. If you need to emphasize vertical edges, transpose the filter H: H'.

```
    [1 2 1;0 0 0;-1 -2 -1].
```

Class Support

H is of class double.

Notes

For 'gaussian' and 'log' kernels, When a value is provided for sigma and HSIZE is specified as [], a default HSIZE of dimension 2*ceil(2*sigma)+1 is chosen.

Example

```
   I = imread('cameraman.tif');
   subplot(2,2,1);imshow(I);title('Original Image');
   H = fspecial('motion',20,45);
   MotionBlur = imfilter(I,H,'replicate');
   subplot(2,2,2);imshow(MotionBlur);title('Motion Blurred Image');
   H = fspecial('disk',10);
   blurred = imfilter(I,H,'replicate');
   subplot(2,2,3);imshow(blurred);title('Blurred Image');
```

See also conv2, edge, filter2, fsamp2, fwind1, fwind2, imfilter, imsharpen

```
Documentation for fspecial
```

Filtrování barevných obrázků

Obrázek = trojrozměrná matice.

```
I = imread("lena_rgb.png");
size(I)
```

```
ans = 1×3
   512   512     3
```

```
% jednotlive slozky
I_R = I(:, :, 1); % cervena
I_G = I(:, :, 2); % zelena
I_B = I(:, :, 3); % modra

figure,
subplot(1,3,1), imshow(I_R);
title('cervena slozka');
subplot(1,3,2), imshow(I_G);
title('zelena slozka');
subplot(1,3,3), imshow(I_B);
title('modra slozka');
```

Filtrování v RGB prostoru

Fitrujeme všechny složky zvlášť.

```
% prumerovaci filtr
w = fspecial('average',5);

J_R = imfilter(I_R,w,'corr','same');
J_G = imfilter(I_G,w,'corr','same');
J_B = imfilter(I_B,w,'corr','same');

J=uint8([]);
J(:,:,1) = J_R;
J(:,:,2) = J_G;
J(:,:,3) = J_B;

figure,
subplot(1,2,1), imshow(I);
title('Puvodni obrazek')
subplot(1,2,2), imshow(J);
title('Upraveny obrazek')
```

Filtrování v HSV prostoru

Stačí filtrovat jen jasovou složku. Pro převod mezi RGB a HSV modely použijeme funkce `rgb2hsv()` a `hsv2rgb()`.

```
% prumerovaci filtr
I_hsv = rgb2hsv(I);
w = fspecial('average',5);
I_hsv(:,:,3) = imfilter(I_hsv(:,:,3) ,w,'same');
I_rgb = im2uint8(hsv2rgb(I_hsv));

figure,
subplot(1,2,1), imshow(I);
title('Puvodni obrazek')
subplot(1,2,2), imshow(I_rgb,[]);
title('Upraveny obrazek')
```

Úkol 4

Porovnejte výsledky při filtrování v RGB prostoru a v HSV prostoru. Rozdíl mezi obrázky berte jako průměrný rozdíl korespondujících pixelů.

Úkol 5

Vyberte vhodný filtr a odstraňte (minimalizujte) nežádoucí informaci v obrázcích:

```
I1 = imread('cv5-img1.png');
I2 = imread('cv5-img2.png');

figure,
subplot(1,2,1), imshow(I1);
```

```
title('Obrazek 1')
subplot(1,2,2), imshow(I2);
title('Obrazek 2')
```