



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف دوم درس مبانی بینایی کامپیوتر

نام و نام خانوادگی: علیرضا ابره فروش

شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: بهار ۱۴۰۱/۱۴۰۰

مدرس: دکتر نادر کریمی

دستیاران آموزشی: بهنام ساعدی - محمدرضا مزروعی

۱

ابتدا متغیرهای زیر را تعریف می‌کنیم.

H و W : ابعاد تصویر

I : تصویر اصلی

J : تصویر مورد بررسی

MAX_I : بزرگ‌ترین سطح روشنایی محتمل در تصویر (مقدار پیش‌فرض ۲۵۵)

MIN_I : کوچک‌ترین سطح روشنایی محتمل در تصویر (مقدار پیش‌فرض ۰)

۱.۱ MAE

۱.۱.۱ حداکثر

حالتی را تصور می‌کنیم که اختلاف سطح روشنایی هر دو پیکسل متناظر در تصویر حداکثر باشد (تصویر تمام سیاه و تمام سفید). بنابراین به ازای هر i و j دامنه، $I(i, j)$ برابر MAX_I و $J(i, j)$ برابر MIN_I یا بالعکس. پس داریم:

$$\max(\frac{1}{H.W} \sum_{i=1}^H \sum_{j=1}^W |I(i, j) - J(i, j)|) = \frac{1}{H.W} \times H.W \times |MAX_I - MIN_I| = |MAX_I - MIN_I|$$

که در حالت پیش‌فرض این مقدار برابر ۲۵۵ است.

۲.۱.۱ حداقل

حالتی را تصور می‌کنیم که اختلاف سطح روشنایی هر دو پیکسل متناظر در تصویر حداقل باشد (تصاویر برابر باشند). بنابراین به ازای هر i و j دامنه، $I(i, j)$ برابر $J(i, j)$ است. پس داریم:

$$\min(\frac{1}{H.W} \sum_{i=1}^H \sum_{j=1}^W |I(i, j) - J(i, j)|) = \frac{1}{H.W} \times H.W \times |I(i, j) - I(i, j)| = 0$$

۲.۱ MSE

۱.۲.۱ حداکثر

مشابه قسمت قبل، حالتی را تصور می‌کنیم که اختلاف سطح روشنایی هر دو پیکسل متناظر در تصویر حداکثر باشد (تصویر تمام سیاه و تمام سفید). بنابراین به ازای هر i و j دامنه، $I(i, j)$ برابر MAX_I و $J(i, j)$ برابر MIN_I یا بالعکس. پس داریم:

$$\max(\frac{1}{H.W} \sum_{i=1}^H \sum_{j=1}^W (I(i, j) - J(i, j))^2) = \frac{1}{H.W} \times H.W \times (MAX_I - MIN_I)^2 = (MAX_I - MIN_I)^2$$

که در حالت پیش‌فرض این مقدار برابر ۶۵۰۲۵ است.

۲.۲.۱ حداقل

مشابه قسمت قبل، حالتی را تصور می‌کنیم که اختلاف سطح روشنایی هر دو پیکسل متناظر در تصویر حداقل باشد (تصاویر برابر باشند). بنابراین به ازای هر i و j دامنه، $I(i, j)$ برابر $J(i, j)$ است. پس داریم:

$$\min(\frac{1}{H.W} \sum_{i=1}^H \sum_{j=1}^W (I(i, j) - J(i, j))^2) = \frac{1}{H.W} \times H.W \times (I(i, j) - I(i, j))^2 = 0$$

۳.۱ PSNR

برای بیشینه (کمینه) کردن تابع $10 \log_{10}(\frac{MAX_I^2}{MSE})$ کافیت تابع $\frac{MAX_I^2}{MSE}$ بیشینه (کمینه) کنیم.

۱.۳.۱ حداکثر

با فرض ناصفر بودن مقدار MAX_I داریم:

$$\max(10 \log_{10}(\frac{MAX_I^2}{MSE})) = \lim_{MSE \rightarrow 0}(10 \log_{10}(\frac{MAX_I^2}{MSE})) = \infty$$

پس به ازای دو تصویر با MSE نزدیک به صفر (دو تصویر تقریباً برابر)، مقدار PSNR به بی‌نهایت میل می‌کند.

۲.۳.۱ حداقل

اگر I تصویر تمام سیاه (MAX_I ناچیز باشد)، و J تصویر غیر تمام سیاه باشد، آنگاه چون MAX_I تقریباً صفر است داریم:

$$\min(10 \log_{10}(\frac{MAX_I^2}{MSE})) = \lim_{MAX_I \rightarrow 0}(10 \log_{10}(\frac{MAX_I^2}{MSE})) = -\infty$$

۲

سطح روشنایی هر پیکسل در تصویر خاکستری‌گونه از MIN_I (پیش‌فرض ۰) تا MAX_I (پیش‌فرض ۲۵۵) متغیر است. بدترین MSE که در واقع بزرگ‌ترین MSE است زمانی رخ می‌دهد که اختلاف سطح روشنایی هر دو پیکسل متناظر در تصاویر ماکسیمم باشد. در نتیجه برای ساخت چنین تصویری فاصله‌ی سطح روشنایی هر پیکسل در تصویر اولیه را از مقادیر MIN_I و MAX_I به دست می‌آوریم (قدر مطلق تفاضل). در صورتی که سطح روشنایی به MIN_I (سیاه) نزدیک‌تر بود، مقدار MAX_I (سفید) و در صورتی که به MAX_I (سفید) نزدیک‌تر بود، مقدار MIN_I (سیاه) را در پیکسل متناظر قرار می‌دهیم. در این حالت چون قدر مطلق تفاضل‌ها به ازای همه پیکسل‌های تصویر ماکسیمم هستند پس تضمین می‌شود که ماکسیمم MSE (بدترین MSE) را نسبت به تصویر ورودی داریم.

۳

۱.۳ الف

Function ۱.۱.۳

```

1 function J = toBlackWhite(I)
2 %TOBLACKWHITE Summary of this function goes here
3 % RGB to black and white with maximum psnr
4 J = uint8(zeros(size(I)));
5 for i = 1: size(I, 1)
6     for j = 1: size(I, 2)

```

```

7         avg_of_rgb = round(double(I(i, j, 1)) / 3 + double(I(i, j, 2) / 3) + double(I
            (i, j, 3) / 3));
8         if(255 - avg_of_rgb <= avg_of_rgb)
9             J(i, j, 1) = 255;
10            J(i, j, 2) = 255;
11            J(i, j, 3) = 255;
12        else
13            J(i, j, 1) = 0;
14            J(i, j, 2) = 0;
15            J(i, j, 3) = 0;
16        end
17    end
18 end
19 end

```

Driver code ۲.۱.۳

```

1  clc
2  clear
3  close all
4  imtool close all
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6  I = imread("images\i3.jpg");
7  J = uint8(zeros(size(I)));
8
9  J = toBlackWhite(I);
10  psnr(J, I)
11
12  figure, imshow(I, []);
13  figure, imshow(J, []);

```

ب ۲.۳

Function ۱.۲.۳

```

1  function J = floydSteinberg(I)
2  %FLOYDSTEINBERG Summary of this function goes here
3      I = int16(I);
4      J = zeros(size(I));

```

```

5     for i = 1: size(I, 1)
6         for j = 1: size(I, 2)
7             old_value = I(i, j, 1);
8             remaining = int16(-1);
9             if(255 - I(i, j, 1) < I(i, j, 1))
10                I(i, j, :) = 255;
11                remaining = -int16(int16(255 - old_value));
12            else
13                I(i, j, :) = 0;
14                remaining = int16(old_value);
15            end
16            if(j < size(J, 2))
17                I(i, j + 1, :) = I(i, j + 1, 1) + floor((7 / 16) * remaining);
18                if(i < size(J, 1))
19                    I(i + 1, j + 1, :) = I(i + 1, j + 1, 1) + floor((1 / 16) * remaining)
20                ;
21            end
22        end
23        if(i < size(I, 1))
24            I(i + 1, j, :) = I(i + 1, j, 1) + floor((5 / 16) * remaining);
25            if(j > 1)
26                I(i + 1, j - 1, :) = I(i + 1, j - 1, 1) + floor((3 / 16) * remaining)
27            ;
28        end
29    end
30    J = uint8(I);
31 end

```

Driver code ۲.۲.۳

```

1 clc
2 clear
3 close all
4 imtool close all
5 %%%%%%%%%%%

```

```

6 I = imread("images\i3b.png");
7 J = floydSteinberg(I);
8 K = imread("images\i3b(ground truth).png");
9 figure, imshow(I, []);
10 figure, imshow(J, []);
11 figure, imshow(K, []);
12 %%
13
14 x = I(:, :, 1);
15 y = J(:, :, 1);
16 z = K(:, :, 1);

```

۳.۳ ج

```

1 clc
2 clear
3 close all
4 imtool close all
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 I = imread("images\i3b.png");
7 J = floydSteinberg(I);
8 K = toBlackWhite(I);
9
10 fs_psnr = psnr(J, I);
11 bw_psnr = psnr(K, I);
12
13 figure, imshow(I, []);
14 figure, imshow(J, []);
15 figure, imshow(K, []);

```

با مقایسه‌ی PSNR در دو تصویر تولید شده در قسمت‌های قبل درمی‌یابیم که بهتر بودن PSNR یک تصویر نسبت به تصویر دیگر الزاما به معنی بهتر بودن کیفیت بصری آن تصویر نیست. همانطور که می‌بینیم، تصویر تولید شده توسط الگوریتم Floyd-Steinberg با وجود داشتن PSNR پایین‌تر نسبت به تصویر تولید شده توسط الگوریتم حریصانه (تابع toBlackWhite) جزئیات تصویر را بهتر مشخص می‌کند و کیفیت بصری بهتری دارد. پس نتیجه می‌گیریم که PSNR در همه جا معیار دقیقی برای مقایسه‌ی کیفیت تصاویر نیست.



شکل ۱: خروجی الگوریتم Floyd-Steinberg



شکل ۲: خروجی الگوریتم حریصانه

۴

Function ۱.۴

```

1 function J = My_Imresize_BL(Input_Image, Resizing_Factor)
2 %MY_IMRESIZE_BL Summary of this function goes here
3     old_row = size(Input_Image, 1);
4     old_column = size(Input_Image, 2);
5     new_row = ceil(old_row * Resizing_Factor);
6     new_column = ceil(old_column * Resizing_Factor);
7     J = uint8(zeros(new_row, new_column, 3));
8     r_ratio = double(old_row - 1) / double(new_row - 1);
9     c_ratio = double(old_column - 1) / double(new_column - 1);
10    for i = 1: new_row
11        i_floor = uint32(floor(r_ratio * i)) + 1;

```

```

12     i_ceil = uint32(ceil(r_ratio * i));
13     if (i_floor > old_row)
14         i_floor = old_row;
15     end
16     if (i_ceil > old_row)
17         i_ceil = old_row;
18     end
19     y = uint32(r_ratio * i - i_floor);
20
21     for j = 1: new_column
22         j_floor = uint32(floor(c_ratio * j)) + 1;
23         j_ceil = uint32(ceil(c_ratio * j));
24         if (j_floor > old_column)
25             j_floor = old_column;
26         end
27         if (j_ceil > old_column)
28             j_ceil = old_column;
29         end
30         x = uint32(c_ratio * j - j_floor);
31
32         a = uint32(zeros(1, 3));
33         b = uint32(zeros(1, 3));
34         c = uint32(zeros(1, 3));
35         d = uint32(zeros(1, 3));
36         e = uint32(zeros(1, 3));
37         %if (i_floor <= old_row && i_ceil <= old_row && j_floor <= old_column &&
j_ceil <= old_column)
38             a = uint32(Input_Image(i_floor, j_floor, :));
39             b = uint32(Input_Image(i_floor, j_ceil, :));
40             c = uint32(Input_Image(i_ceil, j_floor, :));
41             d = uint32(Input_Image(i_ceil, j_ceil, :));
42             e = uint32(round(a + (b - a) * x + (c - a) * y + (a - b - c + d) * x * y)
);
43             J(i, j, :) = e;
44         %end
45     end

```



```

46     end
47 end

```

Driver code ۲.۴

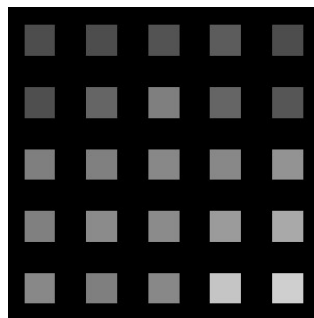
```

1  clc
2  clear
3  close all
4  imtool close all
5  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6  I = imread("images\i3b.png");
7  rf = 0.3;
8  J = My_Imresize_BL(I, rf);
9  K = uint8(imresize(double(I), rf, 'bilinear'));
10 figure, imshow(I, []);
11 figure, imshow(J, []);
12 figure, imshow(K, []);
13 immse(J, K)

```

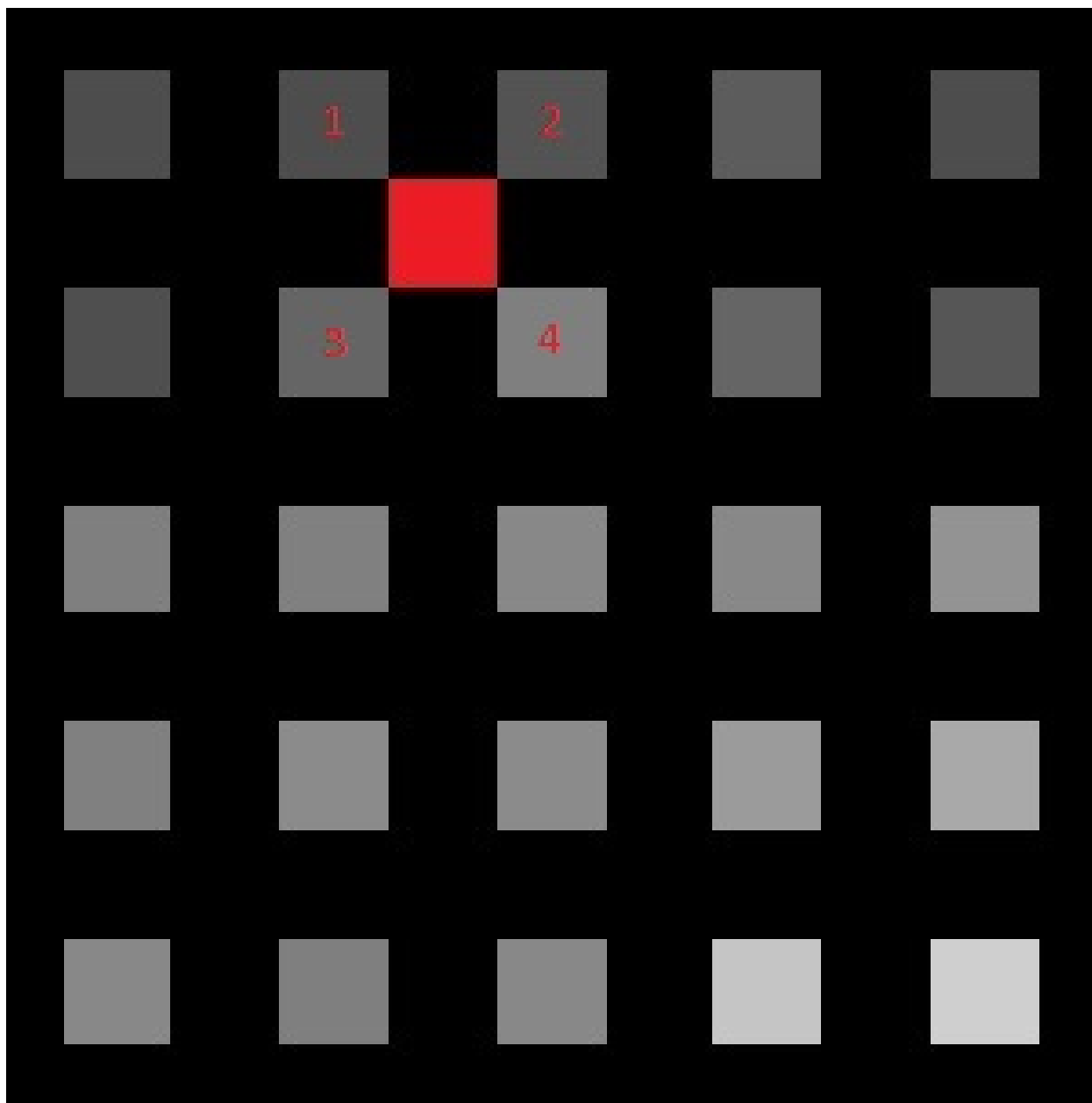
۵

پیکسل‌های تصویر را به طور یک در میان به شکل زیر می‌چینیم:



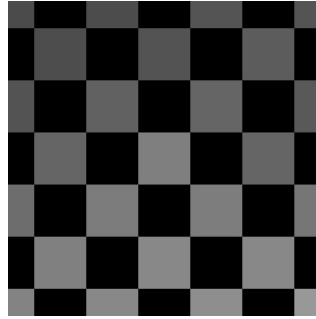
شکل ۳

سطح روشنایی پیکسل‌های قرار گرفته در سطر زوج و ستون زوج را برابر با میانگین پیکسل‌های مجاور (۴ پیکسل مشخص شده در شکل زیر در صورت وجود) قرار می‌دهیم. سطح روشنایی پیکسل‌های مرزی که دارای ۲ همسایه‌ی مورب و ۱ همسایه‌ی مورب هستند را هم به همین صورت مقداردهی می‌کنیم.



شکل ۴

پیکسل‌ها تصویر به شکل زیر در می‌آیند:



شکل ۵

در گام بعد، سطح روشنایی پیکسل‌های باقی‌مانده را برابر میانگین پیکسل‌های مجاور (بالا، پایین، چپ و راست) قرار می‌دهیم. سطح روشنایی پیکسل‌های مرزی هم به طریق مشابه تعیین می‌شوند.

۶

۱.۶ الف

Function ۱.۱.۶

```

1 function hist_vector = myHist(I)
2 %MYHIST Summary of this function goes here
3     x_axis = 0: 255;
4     y_axis = zeros(1, 256);
5     for i = 1: size(I, 1)
6         for j = 1: size(I, 2)
7             y_axis(I(i, j) + 1) = y_axis(I(i, j) + 1) + 1;
8         end
9     end
10    hist_vector = y_axis;
11    figure, bar(x_axis, y_axis);
12 end

```

Driver code ۲.۱.۶

```

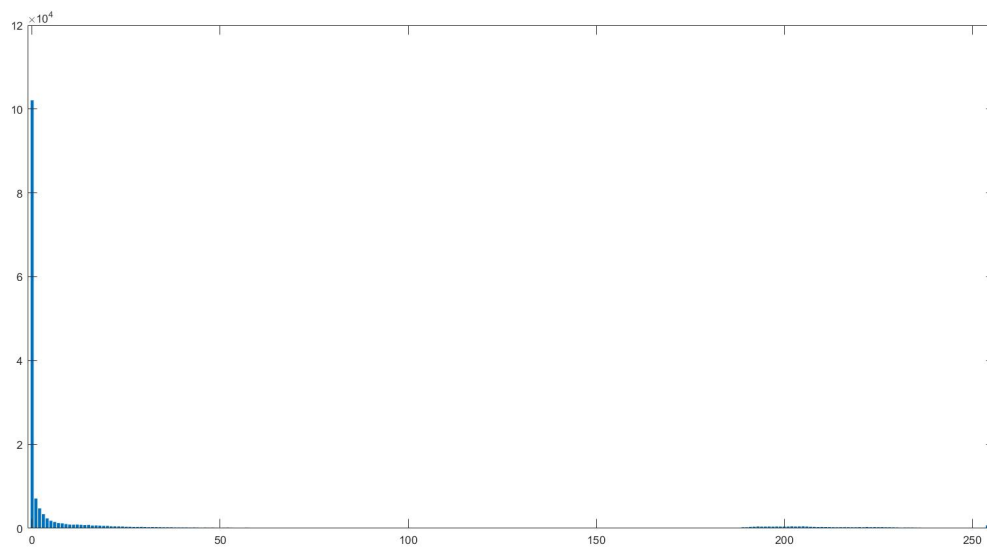
1 clc
2 clear
3 close all
4 imtool close all
5 %%%%%%%%%%

```

```

6 I = imread("images\Image.tif");
7 myHist(I);

```



شکل ۶: هیستوگرام تصویر *Image.tif*

۲.۶ ب

همانطور که در بخش قبل دیدیم، سطح روشنایی‌های صفر و نزدیک به صفر بیشتر پیکسل‌های این تصویر را تشکیل داده‌اند. اما سطح روشنایی‌های بزرگ‌تر و نزدیک به ۲۵۵ هم

منابع