

تمرین ۱

هادی تمیمی

اطلاعات گزارش	چکیده
تاریخ:	
واژگان کلیدی:	
کوانتیزیشن	
درون یابی	
تبدیل هندسی	

۱-مقدمه

۲-شرح تکنیکال

۱.۱.۱

اگر (x, y) مختصات یک feature point در تصویر A بوده و (u, v) متناظر آن در تصویر B باشد، آنگاه خواهیم داشت:

$$x = k_1 * u + k_2 * v + k_3 * u * v + k_4$$

$$y = k_5 * u + k_6 * v + k_7 * u * v + k_8$$

باید ضرایب k_1, \dots, k_8 را بدست آوریم. یک دستگاه با ۸ مجهول داریم و که با ۸ و ۴ feature point میتوان این ضرایب

را بدست آورد. حال برای تبدیل تصویر B به تصویر A میتوان از این تابع برای تمام نقاط B استفاده کرد. در صورتی که

نگاشت از A به B خواستیم باید دستگاه زیر را حل کنیم تا نگاشت بدست آید:

$$u = k_1 * x + k_2 * y + k_3 * x * y + k_4$$

$$v = k_5 * x + k_6 * y + k_7 * x * y + k_8$$

۱.۱.۲

نقاط feature point را متناسب با overlap انتخاب میکنیم. سپس در تصویر نهایی از ۰ تا شروع منطقه overlap را از تصویر اول برمیداریم. سپس باقی تصویر را از تصویر دوم به کمک نگاشت بدست آمده و درون یابی دوخطی محاسبه میکنیم.

۱.۱.۳

با ماتریکس rotaton دو نقطه (x,y) و (v,w) به صورت زیر بهم تبدیل خواهند شد.

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} v & w & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

۱.۲.۱

بررسی تاثیر histeq بر روی سطوح خاکستری.

level	4	8	16	32	64	128
Without histeq	۳۸۳.۵۵۱	۸۳.۳۱۹۸	۲۱.۰۸۹۷	۵.۴۷۴۴	۱.۵۰۵۱	۰.۵۰۰۰
With histeq	۳۳۳.۵۱۶	۸۳.۹۰۷۱	۲۰.۶۹۸۲	۵.۱۳۷۸	۱.۷۹۹۹	۰.۴۸۶۱

۱.۲.۲. در این مسئله ابتدا تصویر را به کمک دو روش averaging و remove row&col و downsampling میکنیم و سپس به کمک درون یابی دو خطی و تکرار پیکسل ها آنرا به سائز اصلی بر میگردانیم.

در بازگرداندن به کمک تکرار پیکسل ها هر پیکسل سه بار برای پیکسل های مجهول مجاورش تکرار میشود.

در بازگرداندن به کمک درون یابی دو خطی به ازای هر ۴ پیکسل ۳ تای آنها مجهول است که در نتیجه یک مجموعه ۱۶ تایی آنها را که شامل ۴ معلوم میباشد و برای ۱۲ تای مجهول میتوان از آنها استفاده کرد، استفاده کرده ام.

۲-شرح نتایج

- نتیجه تصویر panorama:



-تصویر اول



-تصویر دوم



-نتیجه نهایی.

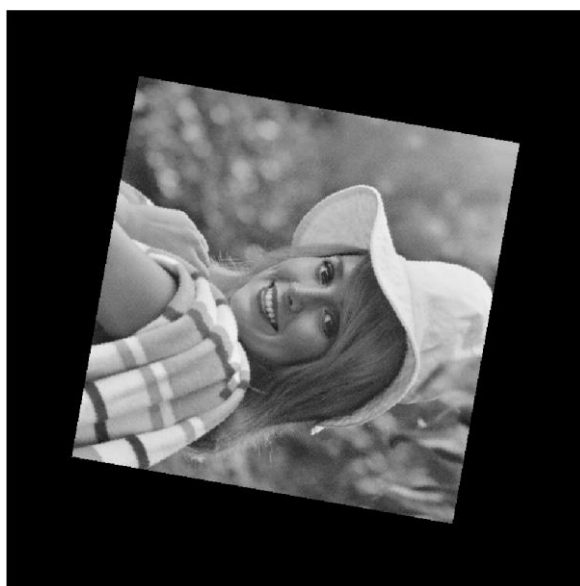
- نتیجه rotation :



-تصویر اصلی



- چرخش ۴۵ درجه



- چرخش ۱۰۰ درجه



- چرخش ۶۷۰ درجه

نتیجه histogram:



- تصویر اصلی



- تصویر اصلی خاکستری شده

0.500061



- تصوير با لول ۱۲۸ بدون histeq

1.5051



- تصوير با لول ۶۴ بدون histeq

5.47443



- تصوير با لول ۳۲ بدون histeq

21.0897



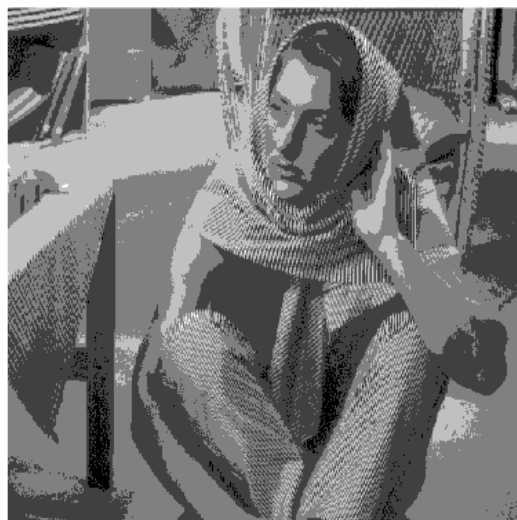
- تصوير با لول ۱۶ بدون histeq

83.3198



- تصویر با لول ۸ بدون histeq

383.551



- تصویر با لول ۴ بدون histeq

نتایج بعد از histeq:

0



- تصویر اصلی خاکستری بعد از histeq

0.486168



- تصویر با لول ۱۲۸ خاکستری بعد از histeq

1.7999



- تصویر با لول ۶۴ خاکستری بعد از histeq

5.13786



- تصویر با لول ۳۲ خاکستری بعد از histeq

20.6982



- تصویر با لول ۱۶ خاکستری بعد از histeq

83.9071



- تصویر با لول ۸ خاکستری بعد از histeq

333.516



- تصویر با لول ۴ خاکستری بعد از histeq

نتایج نهایی:

Report MSE

level	4	8	16	32	64	128
Without histeq	۳۸۳.۵۵۱	۸۳.۳۱۹۸	۲۱.۰۸۹۷	۵.۴۷۴۴	۱.۵۰۵۱	۰.۵۰۰۰
With histeq	۳۳۳.۵۱۶	۸۳.۹۰۷۱	۲۰.۶۹۸۲	۵.۱۳۷۸	۱.۷۹۹۹	۰.۴۸۶۱

کد ها:

۱.۱.۲:

لود کردن ورودی و ایجاد صفحه:

```
car1_image = imread("Images/1/car1.jpg");  
car2_image = imread("Images/1/car2.jpg");  
width = 1400;  
lenth = 750;  
output = zeros(lenth,width,3,'uint8');  
overlap_start_im1 = 436;
```

```

for i=1:lenth
    for j=1:overlap_start_im1
        output(i,j,:) = car1_image(i,j,:);
    end
end

```

تعیین نقاط و حل دستگاه:

```

[x1,y1] = deal(436,388);
[x2,y2] = deal(436,514);
[x3,y3] = deal(834,514);
[x4,y4] = deal(834,388);

```

```

[v1,w1] = deal(0,404);
[v2,w2] = deal(0,546);
[v3,w3] = deal(414,546);
[v4,w4] = deal(414,404);

```

```

syms c1 c2 c3 c4 c5 c6 c7 c8

```

```

eq1 = x1 == c1*v1 + c2*w1 + c3*v1*w1 + c4;
eq2 = x2 == c1*v2 + c2*w2 + c3*v2*w2 + c4;
eq3 = x3 == c1*v3 + c2*w3 + c3*v3*w3 + c4;
eq4 = x4 == c1*v4 + c2*w4 + c3*v4*w4 + c4;

```

```

eq5 = y1 == c5*v1 + c6*w1 + c7*v1*w1 + c8;
eq6 = y2 == c5*v2 + c6*w2 + c7*v2*w2 + c8;
eq7 = y3 == c5*v3 + c6*w3 + c7*v3*w3 + c8;
eq8 = y4 == c5*v4 + c6*w4 + c7*v4*w4 + c8;

```

```

sol = solve([eq1, eq2, eq3, eq4, eq5, eq6, eq7, eq8],
[c1,c2,c3,c4,c5,c6,c7,c8]);
c1 = double(sol.c1);
c2 = double(sol.c2);
c3 = double(sol.c3);
c4 = double(sol.c4);
c5 = double(sol.c5);
c6 = double(sol.c6);
c7 = double(sol.c7);
c8 = double(sol.c8);

```

انتقال پیکسل های بعد اورل به تصویر نهایی:

```

for i=1:lenth
    for j=overlap_start_im1:width
        v = i;
        w = j-overlap_start_im1+1;
        z = v*w;
        x = [c1 c2 c3 c4]*[v; w; z;0.1];
        y = [c5 c6 c7 c8]*[v; w; z;1];
    end
end

```

```

        [t,s] = nearest_neighbor(x,y,car2_image);
        output(i,j,:) = car2_image(t,s,:);

    end
end

```

تابع درون یابی دو خطی:

```

function [x,y] = nearest_neighbor(s,t,image)
    x = s;
    y =t;
    shape = size(image);
    floor_x = floor(x);
    ceil_x = ceil(x);
    floor_y = floor(y);
    ceil_y = ceil(y);

    if(abs(floor_x-x) < abs(ceil_x-x))
        x = floor(x);
    else
        x = ceil(x);
    end
    if(abs(floor_y-y) < abs(ceil_y-y))
        y = floor(y);
    else
        y = ceil(y);
    end
    if(x>shape(1))
        x= shape(1);
    end
    if(x<=0)
        x=1;
    end
    if(y>shape(2))
        y= shape(2);
    end
    if(y<=0)
        y=1;
    end
end
end

```

۱.۱.۳:

لود کردن و ایجاد تصویر:

```

image = imread("Images/1/Elaine.bmp");

```

```

%uncomment for different degree *****
D=670*pi/180;
%D=45*pi/180;
%D=100*pi/180;
%*****

rotate = [cos(D) sin(D) 0; -sin(D) cos(D) 0; 0 0 1];
Size = size(image);

base = zeros(Size(1)*1.5,Size(2)*1.5,'uint8');
for i=1:Size(1)
    for j=1:Size(2)
        base(Size(1)/4+i,Size(2)/4+j)=image(i,j);
    end
end
Size = size(base);
image_temp = base;

```

چرخش تصویر با درون یابی ها :

```

for i=1:Size(1)
    for j=1:Size(2)
        base(Size(1)/4+i,Size(2)/4+j)=image(i,j);
    end
end
Size = size(base);
image_temp = base;

for i = 1:Size(1)
    for j = 1:Size(1)
        [v,w] = nearest_neighbor_inter(i,j,base,rotate);
        image_temp(i,j)= base(v,w);
    end
end

```

درون یابی نزدیک ترین همسایه:

```

function [x,y] =
nearest_neighbor_inter(i,j,image,transform_matrix)

    shape = size(image);
    X = transform_matrix\[i-shape(1)/2;j-shape(2)/2;1];

```



```

x = X(1)+shape(1)/2;
y = X(2)+shape(2)/2;
floor_x = floor(x);
floor_y = floor(y);
ceil_x = ceil(x);
ceil_y = ceil(y);

if(abs(floor_x-x) < abs(ceil_x-x))
    x = floor(x);
else
    x = ceil(x);
end
if(abs(floor_y-y) < abs(ceil_y-y))
    y = floor(y);
else
    y = ceil(y);
end

if(x<=0)
    x=1;
end
if(x>shape(1))
    x= shape(1);
end

if(y>shape(2))
    y= shape(2);
end
if(y<=0)
    y=1;
end
end
end

```

۱.۲.۱

لود کردن داده:

```
image = imread("Images/1/Barbara.bmp");
```

تولید تصویر خاکستری:

```
x = rgb2gray(image);
```

کاهش سطوح خاکستری با لول های مختلف بدون histeq و نمایش آن:

```

level4= 64*round(x/64);
level8 =32*round(x/32);
level16 =16*round(x/16);
level32 =8*round(x/8);

```

```

level64 =4*round(x/4);
level128 =2*round(x/2);

figure
subplot(2,4,1)
imshow(x);
title(immse(x,x));
subplot(2,4,2)
imshow(level128);
title(immse(level128,x));
subplot(2,4,3)
imshow(level64);
title(immse(level64,x));
subplot(2,4,4)
imshow(level32);
title(immse(level32,x));
subplot(2,4,5)
imshow(level16);
title(immse(level16,x));
subplot(2,4,6)
imshow(level8);
title(immse(level8,x));
subplot(2,4,7)
imshow(level4);
title(immse(level4,x));

```

کاهش سطوح خاکستری بعد از histeq و نمایش آن:

```

x = histeq(x);

level4= 64*round(x/64);
level8 =32*round(x/32);
level16 =16*round(x/16);
level32 =8*round(x/8);
level64 =4*round(x/4);
level128 =2*round(x/2);

figure
subplot(2,4,1)
imshow(x);
title(immse(x,x));
subplot(2,4,2)
imshow(level128);
title(immse(level128,x));
subplot(2,4,3)
imshow(level64);
title(immse(level64,x));

```

```
subplot(2,4,4)
imshow(level32);
title(immse(level32,x));
subplot(2,4,5)
imshow(level16);
title(immse(level16,x));
subplot(2,4,6)
imshow(level8);
title(immse(level8,x));
subplot(2,4,7)
imshow(level4);
title(immse(level4,x));
```