همسان سازی هیستوگرام

علی نصیری سروی

اطلاعات گزارش

تاریخ: 99/01/15

واژگان کلیدي:
هیستوگرام
همسان سازی محلی
همسان سازی هیستوگرام

چکیده

در این تمرین به روش های همسان سازی هیستوگرام میپردازیم.همسان سازی هیستوگرام باعث افزایش کنتراست میشود.

1-مقدمه

هیستوگرام یک تصویر، تابعی گسسته است که اطلاعات تصویر دو بعدی را به برداری یک بعدی تبدیل میکند.در صورت تبدیل تصویر به کمک هیستوگرام دیگر از بردار بدست آمده نمیتوان تصویر را تولید کرد اما این بردار اطلاعات مفیدی مانند توزیع سطوح خاکستری را به ما میدهد.

2-شرح تکنیکال

2.1.1. برای محاسبه هیستوگرام کاری که میکنیم به این صورت میباشد که برای 256 سطح خاکستری مختلف تعداد پیکسل هایی که دارای آن سطح خاکستری میباشند را پیدا میکنیم.در واقع با یک پیمایش روی تمام پیکسل ها و شمارش تعداد پیکسل در هر سطح خاکستری،تعداد پیکسل های 256 سطح خاکستری بدست می آید.

هیستوگرام نرمالیزه شده همان بردار 256 تایی هست با این تفاوت که بردار را بر تعداد پیکسل های تصویر تقسیم میکنیم.

2. 2.1. همسان سازی هیستوگرام

همسان سازی هیستوگرام روشی است که در آن واریانس هیستوگرام یک تصویر افزایش میابد. این روش باعث میشود هیستوگرام یکنواخت تر بشود.

اگر تابع T تبدیل را انجام دهد، خواهیم داشت:

$$s = T(r), \ 0 \le r \le 1$$

که در آن r سطح خاکستری نرمالیزه شده قبل از تبدیل و s سطح خاکستری نرمالیزه شده بعد از تبدیل پیکسل ها میباشند.

تابع T باید دو شرط را ارضا کند:

- در بازه ی $0 \le r \le 1$ صعودی باشد.
- باید برای $0 \le r \le 1$، $0 \le T(r) \le 1$ باشد.

برای بدست آوردن T از cumulative distribution function استفاده میکنیم.اگر احتمال وقوع سطح خاکستری i ام در تصویر p_i باشد،CDF را به صورت زیر محاسبه میکنیم:

$$CDF(i) = \sum_{x=0}^{i} p_i$$

حال یک look up table برای نگاشت از r به s میسازیم.هر مقدار سطح خاکستری s_i برابر مقدار زیر میباشد:

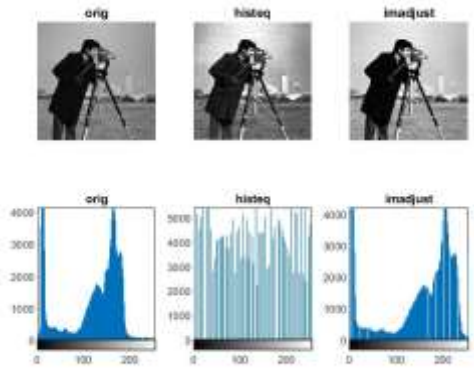
$$s_i = \lfloor (L-1) \times CDF(i) \rfloor$$

که در آن L تعداد کل سطوح خاکستری میباشد.

حال تابع T همان جدول میباشد و میتوان هر پیکسل در تصویر اصلی را به کمک آن تبدیل نمود.

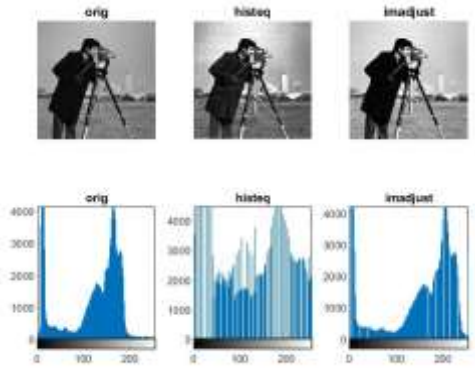
2.1.3. تابع imadjust تقریبا شبیه روشی که در بخش قبل گفتیم عمل کرده و هیستوگرام را به صورت خطی scale میکند.این تابع یک درصد بالا و پایین سطح خاکستری را اشباع میکند.

تابع histeq هیستوگرام عکس ورودی را سعی میکند به یک هیستوگرام هدف نزدیک کند.به صورت دیفالت اگر هیستوگرام هدفی به آن داده نشود، هیستوگرام با 64 تا bin و به صورت تقریبا صاف را هدف در نظر میگیرد.

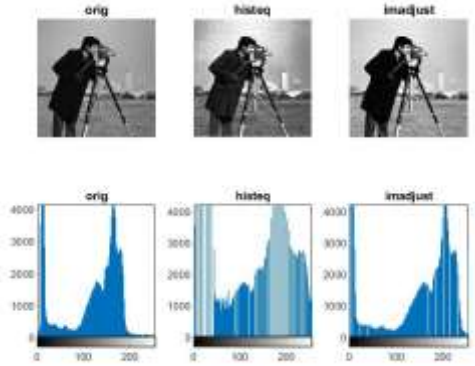


- نتایج با پارامترهای دیفالت

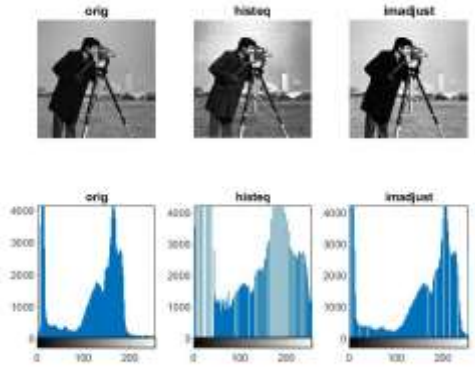
همانطور که مشاهده میشود imadjust شکل کلی هیستوگرام اصلی را حفظ میکند.



- افزایش تعداد bin های histeq به 128



- افزایش تعداد bin های histeq به 256

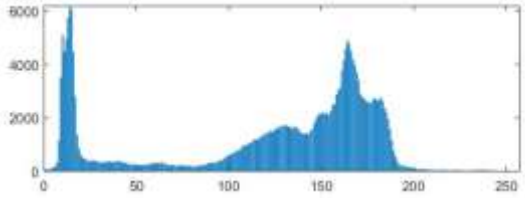


- افزایش تعداد bin های histeq به 512

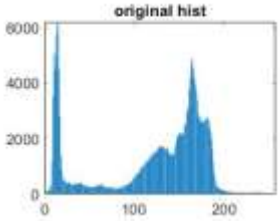
همانطور که مشاهده میشود در صورتی که تعداد bin های histeq را افزایش دهیم مشاهده میشود که شبیه به imadjust عمل میکند.

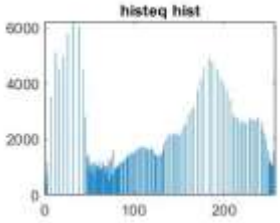
2-شرح نتایج

- نتیجه cameraman histogram:



- نتیجه cameraman with histeq:





2.2.1. همسان سازی محلی هیستوگرام

در روش قبلی اتفاقی که ممکن است رخ دهد این است که برخی نواحی ممکن است روشنایی بالایی داشته باشند که بر روی کل تصویر اثر بگذارند.ینی تصویری که به طور کلی تیره است ولی یک ناحیه خیلی روشن دارد ممکن است در کل روشن تر شود که نتیجه مطلوب نیست.

برای حل این مشکل از همسان سازی محلی هیستوگرام استفاده میشود. در این روش پنجره ای با ابعاد مشخص شده در نظر گرفته میشود و تصویر به قطعاتی با ابعاد این پنجره تقسیم میشود و همسان سازی هیستوگرام بر روی این قطعات صورت میگیرد.

این روش از نظر محاسباتی نسبت به روش قبل بدتر عمل میکند.همچنین نویز در این روش تشدید خواهد شد.مشکل دیگری که این روش دارد **blocking** که باعث شطرنجی شدن تصویر میشود.برای رفع این مشکل باید پنجره مان هم پوشانی داشته باشد.

در حالت با هم پوشانی پنجره میتواند یک فیلتر گاوسی باشد که وزن هر پیکسل و تاثیر آن در هیستوگرام بر اساس همین فیلتر باشد.بدین صورت در هر مرحله نقاط مرکزی پنجره تاثیر بیشتری بر روی تابع نگاشت خواهند داشت.نقاط کناری در صورتی که هم پوشانی داشته باشیم و سایز هر استپ را مناسب انتخاب کنیم در مراحل بعدی خود به مرکز آمده و تاثیرشان را میگذارند.برای این که پیکسل های کناری تصویر هم موثر واقع شوند میتوان از padding استفاده کرد.

- نتایج local histeq:

- تصویر he1 با سایز پنجره 64

- تصویر he1 با سایز پنجره 128

- تصویر he1 با سایز پنجره 256

- تصویر he1 با سایز پنجره 512

- تصویر he2 با سایز پنجره 64

- تصویر he2 با سایز پنجره 128

- تصویر he3 با سایز پنجره 128

- تصویر he3 با سایز پنجره 256

- تصویر he3 با سایز پنجره 512

- تصویر he2 با سایز پنجره 256

- تصویر he2 با سایز پنجره 512

- تصویر he3 با سایز پنجره 64

- تصویر he4 با سایز پنجره 64

- تصویر he4 با سایز پنجره 512

- تصویر he4 با سایز پنجره 128

- تصویر he4 با سایز پنجره 256

کد ها:

2.1.1:

Histogram calculator:

```
image =
imread('Homeworks\Images\2
\Camera Man.bmp');


hist =Img_Hist(image);
subplot(2,1,1);
imshow(image);
subplot(2,1,2);
bar(hist);
function Hist =
Img_Hist(img)
    [R,C]=size(img);
    Hist=zeros(256,1);
    for r = 1:R
        for c=1:C

Hist(img(r,c)+1,1)=Hist(im
g(r,c)+1,1)+1;
        end
    end

end
```

2.1.2:

Histogram equalization:

```
image =
imread('Homeworks\Images\2
\Camera Man.bmp');


new_image =
hist_eq(image);
hist_orig = hist(image);
hist_eqimg =
hist(new_image);

subplot(2,2,1);
imshow(image);
title('original');
subplot(2,2,2);
bar(hist_orig);
title('original hist');
subplot(2,2,3);
imshow(new_image);
title('histeq');
subplot(2,2,4);
bar(hist_eqimg);
title('histeq hist');

function new_image =
hist_eq(img)
    [r,c] = size(img);
    n = r * c;

    new_image =
zeros(r,c,'uint8');

    cdf = zeros(256,1);
    count = zeros(256,1);
    out = zeros(256,1);
    for i=1:r
        for j=1:c
            value =
img(i,j);
            count(value+1)
= count(value+1)+1;
        end
    end
    pdf = count/n;
    sum = 0; L =255;
    for i=1:256
        sum = sum +pdf(i);
        cdf(i) = sum;

        out(i) =
round(cdf(i)*L);
    end
    for i=1:r
        for j=1:c
            new_image(i,j)
= out(img(i,j)+1);
        end
    end
end
```

```
            end
            if(c-j <=
step_size)
                y = c-j;
            end

            local =
img(i:i+x-1,j:j+y-1,:);

            new_local =
hist_eq(local);
            output(i:i+x-
1,j:j+y-1,:) = new_local;

        end
    end

end
function new_image =
hist_eq(img)
    [r,c,z] = size(img);
    n = r * c;

    new_image =
zeros(r,c,z,'uint8');

    cdf =
zeros(256,1,'double');
    pdf =
zeros(256,1,'double');
    count =
zeros(256,1,'double');
    out =
zeros(256,1,'double');
    for i=1:r
        for j=1:c
            value =
img(i,j,:);
            count(value+1)
= count(value+1)+1;
        end
    end
    pdf = count/n;
    sum = 0; L =255;
    for i=1:256
        sum = sum +pdf(i);
        cdf(i) = sum;
```

```
function Hist = hist(img)
    [R,C]=size(img);
    Hist=zeros(256,1);
    for r = 1:R
        for c=1:C

Hist(img(r,c)+1,1)=Hist(im
g(r,c)+1,1)+1;
        end
    end
end
```

:2.2.1

:Histogram equalization

```
he1 =
imread('Homeworks/Images/2
/HE1.jpg');
he2 =
imread('Homeworks/Images/2
/HE2.jpg');
he3 =
imread('Homeworks/Images/2
/HE3.jpg');
he4 =
imread('Homeworks/Images/2
/HE4.jpg');


im = rgb2gray(he4);
x = local_hist(im,512);
imshow(x);
function
output=local_hist(img,step
_size)
    [r,c] = size(img);
    output = img;

    for i=1:step_size:r
        for
j=1:step_size:c
            x = step_size;
            y = step_size;
            if(r-i
<=step_size)
                x = r-i;
```

```
        out(i) =
round(cdf(i)*L);
    end
    for i=1:r
        for j=1:c

new_image(i,j,:) =
out(img(i,j,:)+1);
        end
    end
end
```