

پردازش تصویر رقمی - جناب آقای دکتر پور رضا

گزارش مینی پروژه شماره ۱

تبدیل هندسی ۱

تهیه شده توسط : یاسر آبروشن - شماره دانشجویی ۳۹۲۰۹۸۷۵

دانشگاه آزاد اسلامی واحد مشهد - ترم اول ۹۴-۹۵

موعد تحویل : ۱۳ آبان ۹۴ - تاریخ تحویل: ۱۳ آبان ۹۴

خلاصه:

۱- تبدیل هندسی

در این پروژه، کار بر روی تبدیل تصاویر ۳۶۰ درجه به مسطح و بر عکس آن مد نظر بوده است. کلیه تبدیلات به کار رفته در تمرین های زیر بایستی از نوع هندسی باشد. از این رو مساله ابتدا به تعریف تصاویر ۳۶۰ درجه و نمایش نمونه ای از آن می پردازد و سپس خواسته های خود را بیان می دارد

مقدمه و کلیات مساله:

استفاده از دوربین های همه جهته ابزاری مناسب برای ناوبری در کاربردهای مختلف است. تصویر زیر سمت راست، نمونه ای از تصاویر گرفته شده توسط یک دوربین با لنز همه جهته است. همانطور که مشاهده می شود دوربین قادر به گرفتن تصویر بصورت ۳۶۰ درجه است. با استفاده از تبدیل هندسی مناسب می توان این تصویر را همانند تصویر پایین سمت چپ، بصورت مسطح تبدیل کرد



۱- حل مساله ۱: تبدیل تصاویر ۳۶۰ درجه به مسطح

۱- برنامه‌ای بنویسید که تصویر lm112 و lm113 را دریافت و یک نسخه‌ی مسطح شده از آن را تولید کنید.

الف- شرح تکنیکال

این نوع تبدیل که به نام تکنیک تصویر سازی استریو گرافیک نیز شناخته می شود یکی از تبدیلات هندسی مشهور است که برای مشاهده بهتر و تولید تصاویر مسطح بزرگ (Landscape) استفاده می شود.

در این تبدیل، تصاویری که بوسیله لنزهای ۳۶۰ درجه گرفته شده است باید بوسیله یک تابع تبدیل هندسی به تصویر مسطح تبدیل شود. بهترین الگو برای این کار، استفاده از تبدیل دایره به خط است. در این تبدیل، تصویر ۳۶۰ درجه مد نظر، به صورت دایره های بهم پیوسته با شعاع های متغیر (بیرونی ترین دایره تا شعاع صفر - مرکز تصویر) در نظر گرفته می شود و هر دایره بر روی یک خط در تصویر مقصد نگاشت می شود.

در این تبدیل، مرکز تصویر ۳۶۰ درجه به عنوان مرکز تمام دایره ها، بزرگترین شعاع برابر نصف کوچکترین بعد تصویر در نظر گرفته می شود.

بر این اساس، بزرگترین محیط دایره در تصویر ۳۶۰ درجه برابر طول تصویر مسطح در مقصد خواهد بود و بنابراین تعداد دایره های موجود در تصویر مبدا (که برابر با طول شعاع بر اساس نقطه است) بیانگر تعداد خطوط قابل تولید در مقصد و یا عبارت بهتر ارتفاع تصویر مقصد خواهد بود.

بنابراین، نحوه پر شدن ماتریس مقصد بدین گونه است که ابتدا از پایین ترین خط در تصویر مقصد آغاز می شود. در ازای هر نقطه روی خط مقصد که از صفر تا مقدار طول تصویر ادامه می یابد، نقاط متناظر روی دایره تصویر مبدا با توجه به مقدار شعاع (برابر اندیس ارتفاع خط) با در نظر گرفتن زاویه (در جهت حرکت عقربه های ساعت) بر حسب رادیان از صفر تا 2π انتخاب خواهد شد.

ب- شرح نتایج

برای حل این مساله برنامه YA_DIP_MiniProject_11.m با متد شرح داده شده پیاده سازی شده است. ابتدا تصویر با ابعاد مربع lm112.png به الگوریتم برنامه داده شد و نتیجه lm112_out.jpg تولید شده است.



شکل ۱- تصویر ورودی ۳۶۰ درجه

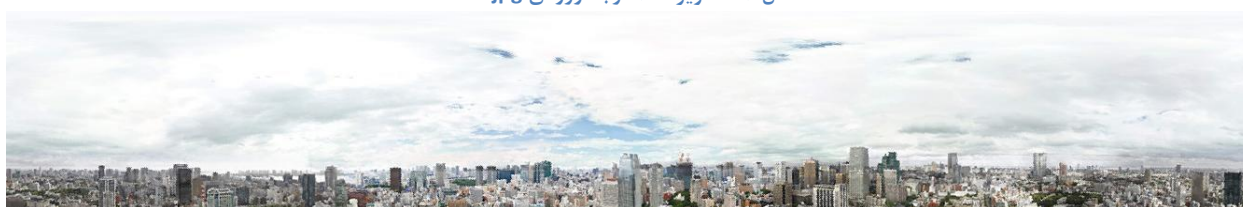


شکل ۲- تصویر خروجی مسطح شده lm112_out.jpg

در مورد تصویر دوم یعنی lm113.jpg که یک نمای زیبا از یک شهر پیشرفته است، یک نکته وجود دارد و آن عدم تساوی ابعاد تصویر است. برای تبدیل این تصاویر باید دقت نمود که شعاع تصویر برابر نصف کمترین بعد تصویر انتخاب شود تا تعادل در تصویر برقرار شود و از طرفی نسبت نقاط به مرکز صفحه به طور دقیق تعیین شوند. بدیهی است در این تصاویر داده های بسیاری که در گوشه های تصویر قرار دارند از دست خواهند رفت. شکل lm113_out.jpg خروجی برنامه را برای این تصویر نمایش می دهد.



شکل ۳- تصویر ۳۶۰ درجه ورودی lm113.jpg



شکل ۴- تصویر مسطح خروجی lm113_out.jpg

نتایج بخوبی تاثیر این الگوریتم را در ایجاد یک تصویر مناسب مسطح نشان می دهند. (کد برنامه در انتهای گزارش با کامنت های مرتبط آورده شده است)

۲- حل مساله ۲: تولید تصاویر Black/White Bar

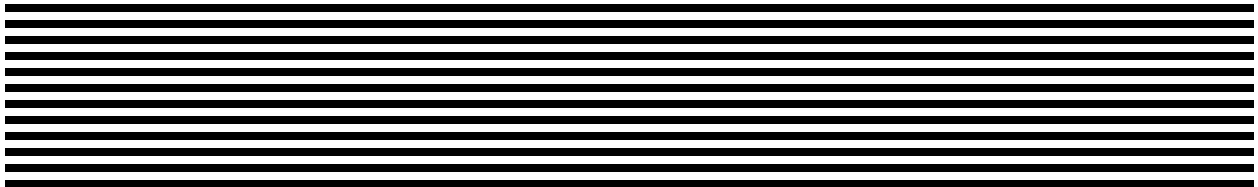
۲- تصویری به اندازه 250×1570 پیکسل را متشکل از نوارهای افقی و عمودی با فواصل مساوی به گونه‌ای تولید کنید که عرض نوارهای سیاه و سفید هر یک ۱۰ پیکسل باشد.

الف- شرح تکنیکال

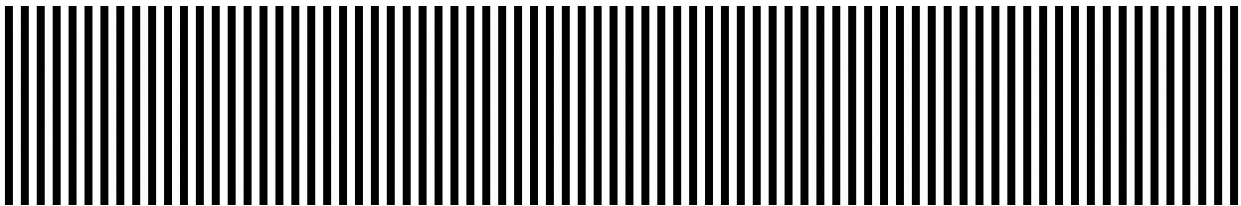
تولید این تصاویر به راحتی و با دو حلقه تو در تو، بگونه‌ای که به ازای هر ۱۰ پیکسل فرآیند تغییر سطح رنگ اتفاق بیفتد انجام شده است. برای تولید تصویر نوارهای عمودی، تغییر رنگ در حلقه تولید نقاط طولی، و برای تولید تصویر نوارهای افقی، تغییر رنگ در حلقه نقاط ارتفاع تصویر صورت گرفته است. تصاویر ۵ و ۶ نمونه‌های تولید شده برنامه نوشته شده برای همین منظور را نشان می‌دهند.

ب- نتایج

برنامه YA_DIP_MiniProject_12.m برای حل این مساله پیاده سازی شده است. (کد برنامه در پیوست گزارش)



شکل ۵- تولید تصویر با نوارهای افقی به طول ۱۰ نقطه - BWBar_Horizontal.jpg



شکل ۶- تولید تصویر با نوارهای عمودی به طول ۱۰ نقطه - BWBar_Vertical.jpg

۳- حل مساله ۳: تبدیل تصاویر سطح به ۳۶۰ درجه

۳- برنامه‌ای بنویسید که با تبدیل هندسی مناسب، تصویر قسمت (۲) را به تصویری شبیه به یک تصویر گرفته شده با لنز همه‌جهته تبدیل کند.

الف- شرح تکنیکال

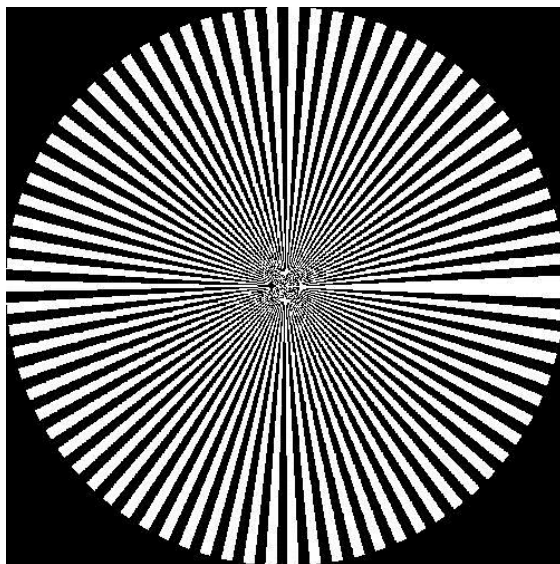
برای حل این مساله از روشی مشابه با جواب سوال یک استفاده شده است. با این تفاوت که تبدیل هندسی این بار برای تبدیل خط به یک دایره استفاده شده است.

برای این منظور، طول تصویر برابر بزرگترین محیط دایره در تصویر ۳۶۰ درجه مقصد در نظر گرفته می شود و بر همین اساس شعاع دایره مقصد نیز تعیین می شود. این شعاع برابر میزانی از ارتفاع در تصویر مقصد خواهد بود که برای عملیات تبدیل استفاده خواهد شود. بدیهی سایر نقاط خارج از محدوده ارتفاع مورد نظر در تبدیل قرار نمی گیرند. فرآیند پر شدن ماتریس مقصد بدین صورت خواهد بود:

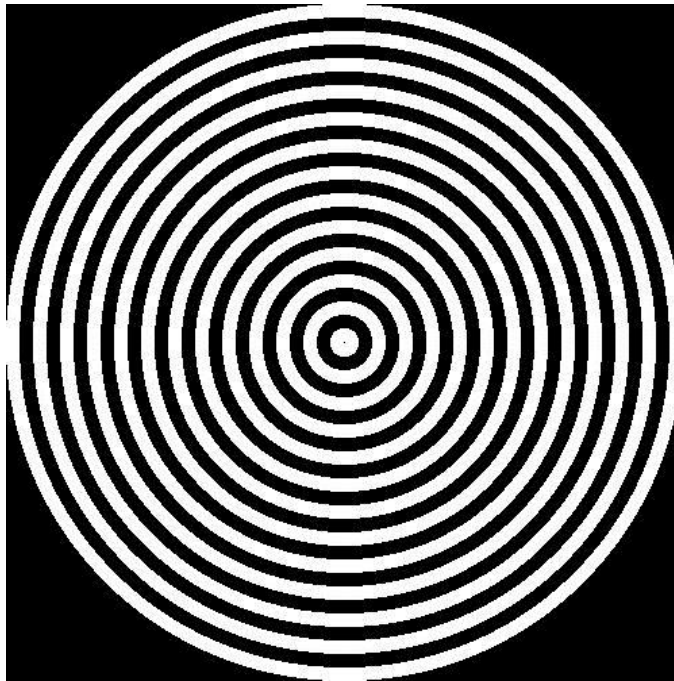
ابتدا دو حلقه تو در تو وظیفه تعریف شعاع دایره ها در مقصد و تعریف زاویه ها (از ۰ تا 2π) را بر عهده دارند. در مورد شعاع پله تغییرات برابر یک نقطه و در مورد زاویه، پله تغییرات برابر ۱ تقسیم بر محیط بزرگترین دایره خواهد بود که بیانگر زاویه بر حسب رادیان خواهند شد. در این حلقه، ابتدا نقاط در مقصد با استفاده از تعریف پارامتری دایره مشخص می گردند و سپس نقاط متناظر با آن در تصویر مبدا مشخص می گردند.

ب- شرح نتایج

برای حل این مساله برنامه YA_DIP_MiniProject_13.m نوشته و پیاده سازی شده است. هر دو تصویر تولید شده در بخش دوم توسط الگوریتم این برنامه با موفقیت به تصاویر شبیه ۳۶۰ درجه تبدیل شده است که در شکل‌های ۷ و ۸ مشخص شده اند.



شکل ۷- خروجی تبدیل ۳۶۰ درجه یک تصویر با میله های عمودی - BWBar_Vertical_out.jpg



شکل ۸- تبدیل ۳۶۰ درجه تصویر میله های افقی - BWBar_Horizontal_out.jpg

۴- حل مساله ۴: تغییر ابعاد و تبدیل تصویر مسطح به ۳۶۰ درجه

۴- تصویر lm111 را با تبدیل هندسی مناسب به ابعاد ۲۵۰×۱۵۷۰ تبدیل کرده و برنامه‌ی نوشته شده در قسمت (۳) را روی آن اعمال کنید.

الف- شرح تکنیکال

برای حل این مساله، ابتدا با استفاده از تکنیک ماتریس Affine ضرایب تغییر ابعاد در هر دو بعد طول و ارتفاع تصویر لحاظ گردیده است و سپس از الگوریتم مرحله قبل برای تبدیل تصویر مسطح به ۳۶۰ درجه استفاده شده است. لازم به توضیح است که ارتفاع قابل تاثیر در تبدیل، برابر با میزان شعاع دایره در مقصد و از بالای تصویر در نظر گرفته شده است. مقدار شعاع از رابطه محیط دایره که برابر با طول تصویر مبدا تقسیم بر 2π است بدست می آید. (بدیهی است سایر نقاط تصویر از دست می روند)

ب- شرح نتایج

برای حل این مساله در بخش تغییر ابعاد، برنامه YA_DIP_MiniProject_14.m نوشته شده است و برای تبدیل به حالت ۳۶۰ درجه از برنامه قسمت ۳ استفاده شده است. شکل شماره ۹، تصویر ورودی به نام lm111.tif را نمایش می دهد. شکل ۱۰، تصویر تغییر یافته به ابعاد ۲۵۰*۱۵۷۰ را نشان می دهد و در نهایت تصویر ۱۱، تبدیل ۳۶۰ درجه تصویر را نمایش می دهد:



شکل ۹- تصویر ورودی به الگوریتم تغییر ابعاد و تبدیل ۳۶۰ درجه - lm111.tif



شکل ۱۰- تصویر تغییر یافته از نظر ابعاد به ۲۵۰*۱۵۷۰ - lm111_1570x250.jpg



شکل ۱۱- تبدیل ۳۶۰ درجه تصویر ورودی بعد از تغییر ابعاد - lm111_1570x250_out.jpg

پوست – لیست کدهای نوشته شده

YA_DIP_MiniProject_11.m – ۱

```
%% Digital Image Processing Course
% Prof. H. Pourreza
% Mini Project No #1
% 360 Degree Pictures Convert To Landscape
FileName = 'D:\Yasser\University\Courses\Digital Image
Processing\MiniProjects\dipmwrk\11\Im113.jpg';
%FileName = 'D:\Yasser\University\Courses\Digital Image
Processing\MiniProjects\dipmwrk\11\Im112.png';
[Pathstr, FName, FExt]= fileparts(FileName);

SourcePic = imread(FileName);
imshow(SourcePic);

[SourcePic_Height, SourcePic_Width, Dimension] = size(SourcePic);

%Finding Center of Picture
CenterPoint =[fix(SourcePic_Width / 2) fix(SourcePic_Height / 2)];

%Finding Radius of 360 Input Image + respect to non square pictures
Radius = min(CenterPoint);

%Finding Output Image Size
LandscapeImageHeight = Radius;
LandscapeImageWidth = fix(2 * pi * Radius);
LandscapeImage=[];

%Algorithm : Filling all points of Landscape Image with suitable
points of
%360 degree Image by dividing 360 degree image into circles with
different
%radius and mapping each circle to a line in Landscape image.

%Scrolling Landscape Image (Output)
for YIndex = LandscapeImageHeight:-1:1
    for XIndex = 1:LandscapeImageWidth

        %Finding Radius of desired circle related to the destination Line
        %by suitable Height
        %First Circle is the biggest
        TempRadius = YIndex; %Radius - YIndex;

        %Finding Teta on destination width Index (radian Base),for
```



```

    %finding desired point in departue
    %Negative in formule because of scrolling clockwise
    Teta =2 * pi * (-XIndex / LandscapeImageWidth);

    SourceX = round(TempRadius * cos(Teta));
    SourceY = round(TempRadius * sin(Teta));

    %Adjusting to Source Image Cartesian to find Accurate
Indices
    SourceX = SourceX + CenterPoint(1); %Radius; %CenterPointX -
SourceX
    SourceY = CenterPoint(2) - SourceY; %Radius - SourceY;
%CenterPointY - SourceY

    %Checking for not exceed the sourceImage bound
    if (SourceX < 1)
        SourceX = 1;
    end
    if (SourceX < (CenterPoint(1)-Radius))
        SourceX = CenterPoint(1)-Radius;
    end
    if (SourceX > (CenterPoint(1)+ Radius) )
        SourceX = CenterPoint(1)+ Radius;
    end
    if (SourceY < 1);
        SourceY = 1;
    end
    if (SourceY < (CenterPoint(2)-Radius));
        SourceY = CenterPoint(1)-Radius;
    end

    if (SourceY > (CenterPoint(2)+Radius))
        SourceY = CenterPoint(2)+Radius;
    end

    %Final Assignment
    LandscapeImage(YIndex, XIndex, :) = SourcePic(SourceY,
SourceX, :);

    end %YIndex
end %XIndex

figure;
imshow(mat2gray(LandscapeImage));
imwrite(mat2gray(LandscapeImage),
strcat(Pathstr, '\', FName, '_out', '.jpg'), 'jpg');

```

٢

مینی پروژه ۱ - تبدیل هندسی ۱ - یاسر آبروشن - صفحه ۱۰ از ۱۳

```

        end

        for XIndex = 0:OutImageWidth-1
            OutImage(YIndex+1, XIndex+1) = CColor;
        end %XIndex
    end %YIndex

figure;
imshow(mat2gray(OutImage));
imwrite(mat2gray(OutImage),
strcat('D:\Yasser\University\Courses\Digital Image
Processing\MiniProjects\dipmwrk\11\BWBar', '_Horizontal', '.jpg'),
'jpg');

```

YA_DIP_MiniProject_13.m — ۳

```

%% Digital Image Processing Course
% Prof. H. Pourreza
% Mini Project No 1-Section 3
% Landscape To 360 Degree Pictures Convert

FileName = 'D:\Yasser\University\Courses\Digital Image
Processing\MiniProjects\dipmwrk\11\BWBar_Vertical.jpg';
%FileName = 'D:\Yasser\University\Courses\Digital Image
Processing\MiniProjects\dipmwrk\11\BWBar_Horizontal.jpg';
FileName = 'D:\Yasser\University\Courses\Digital Image
Processing\MiniProjects\dipmwrk\11\Im111_1570x250.jpg';
[Pathstr, FName, FExt]= fileparts(FileName);

SourcePic = imread(FileName);
imshow(SourcePic);

[SourcePic_Height, SourcePic_Width, Dimension] = size(SourcePic);

%Algorithm: Converting each line from Source Picture to a Circle
in
%Destination. Source Picture Width would be Perimeter of circle of
Destination and
%Radius of Destination calculated from Perimeter.

%Mapping SourcePic.Width to 2*pi*R
DestPic_Radius = round (SourcePic_Width / (2 * pi));

DestinationPic = [];

```

```

%Correcting Height of Source Picture, according to R
SourcePic_Height = DestPic_Radius;
Teta_Step_Radian = 1 / SourcePic_Width;

DestinationPic_Center=[DestPic_Radius DestPic_Radius];

for RadiusIndex = DestPic_Radius:-1:1
    for TetaIndex=Teta_Step_Radian:Teta_Step_Radian:2*pi

        %Finding Target Point (Circle Formula)
        DestX = DestinationPic_Center(1) + round(RadiusIndex *
cos(TetaIndex));
        DestY = DestinationPic_Center(2) - round(RadiusIndex *
sin(TetaIndex));

        %Finding Source Point
        SourceX = round(TetaIndex * DestPic_Radius);
        SourceY = RadiusIndex;

        %Checking for not exceed the sourceImage bound
        if (SourceX < 1)
            SourceX = 1;
        end
        if (SourceX > SourcePic_Width)
            SourceX = SourcePic_Width;
        end
        if (SourceY < 1);
            SourceY = 1;
        end
        if (SourceY > RadiusIndex)
            SourceY = RadiusIndex;
        end

        DestinationPic(DestY+1 , DestX+1) = SourcePic(SourceY,
SourceX);%because it starts from 0
    end
end

figure;
imshow(mat2gray(DestinationPic));
imwrite(mat2gray(DestinationPic),
strcat(Pathstr, '\', FName, '_out', '.jpg'), 'jpg');

```

YA_DIP_MiniProject_14.m –۴

```
%% Digital Image Processing Course
% Prof. H. Pourreza
% Mini Project No 1-Section 4
% Scaling Picture to 250x1570
Desired_Width = 1570;
Desired_Height = 250;
FileName = 'D:\Yasser\University\Courses\Digital Image
Processing\MiniProjects\dipmwrk\11\Im111.tif';
[Pathstr, FName, FExt]= fileparts(FileName);

SourcePic = imread(FileName);
imshow(SourcePic);
[SourcePic_Height, SourcePic_Width, Dimension] = size(SourcePic);

%Finding Scale Factors
Cx = Desired_Width / SourcePic_Width;
Cy = Desired_Height / SourcePic_Height;
DestinationPic=[];
for YIndex = 1:Desired_Height
    for XIndex=1:Desired_Width

        %Finding Source Point
        SourceX = round(XIndex / Cx);
        SourceY = round(YIndex / Cy);

        %Checking for not exceed the sourceImage bound
        if (SourceX < 1)
            SourceX = 1;
        end
        if (SourceX > SourcePic_Width)
            SourceX = SourcePic_Width;
        end
        if (SourceY < 1);
            SourceY = 1;
        end
        if (SourceY > SourcePic_Height)
            SourceY = SourcePic_Height;
        end

        DestinationPic(YIndex , XIndex) = SourcePic(SourceY, SourceX);
    end
end
figure;
imshow(mat2gray(DestinationPic));
imwrite(mat2gray(DestinationPic),
strcat(Pathstr,'\ ',FName,'_',int2str(Desired_Width),'x',
int2str(Desired_Height),'.jpg'), 'jpg');
```