



## گزارش تمرین کامپیوتری سوم ریاضی مهندسی

دانشور امراللهی  
شماره دانشجویی: ۸۱۰۱۹۷۶۸۵

در این پروژه به طراحی اپلیکیشن گویی با استفاده از Flutter پرداخته‌ایم که کاربر با استفاده از دوربین گویی عکسی را می‌گیرد و برنامه روی عکس گرفته شده نگاشت Fisheye را اعمال می‌کند و نتیجه را به کاربر نشان می‌دهد. گزارش از دو بخش تئوری و عملی تشکیل شده است.

### بخش تئوری

در این بخش به طور مفصل به بخش ریاضی پشت نگاشت Fisheye که کد آن پیاده‌سازی شده می‌پردازیم. نگاشت Fisheye را می‌توان به ۳ مرحله تقسیم کرد:

- ۱) نگاشت تصویر مستطیلی دوربین به مربع به مرکز مبدا مختصات و ضلع ۲
- ۲) نگاشت مربع به مرکز مبدا مختصات و ضلع ۲ به دایره محاطی درونش
- ۳) انبساط دایره به ابعادی مشابه تصویر اولیه تا برای کاربر قابل مشاهده باشد مرحله اول:

برای طول و عرض نقاط به طور مشابه این نگاشت را اعمال می‌کنیم. فرض کنید طول یا عرض را در نظر گرفته‌ایم و طول آن  $L$  واحد باشد. درواقع می‌خواهیم پاره خطی به طول  $L$  را به بازه  $[-1, 1]$  نگاشت دهیم. نقطه‌ای به فاصله  $x$  از ابتدای پاره خط به طول  $L$  در نظر بگیریم. می‌خواهیم به همین نسبت روی پاره خط از  $-1$  تا  $1$  حرکت کرده باشیم. بنابراین ضابطه نگاشت برابر خواهد بود با:

$$\begin{aligned}x' &= -1 + \frac{2x}{width} \\ y' &= -1 + \frac{2y}{height}\end{aligned}$$

که  $L$  همان  $width$  یا  $height$  که ابعاد تصویر است می‌باشند. به این عملیات نرمالایز نیز می‌گویند. مرحله دوم:

برای نگاشت مربع به مرکز مبدا مختصات به دایره واحد به مرکز مبدا مختصات ابتدا به نگاشت خطی با  $x$  ثابت یا  $y$  ثابت به بیضی‌ای درون دایره فکر کنید. بیضی با  $x$  ثابت معادله‌ای به شکل زیر دارد:

$$1 = \frac{x'^2}{x^2} + \frac{y'^2}{b^2}$$

می‌خواهیم تمام نقاط بالایی دایره از زاویه 45 تا 135 درجه پوشش داده‌شوند. بنابراین برای نقاط  $-1 \leq x \leq 1$  می‌خواهیم بیضی از نقطه

$$(x', y') = \left( \frac{x}{\sqrt{2}}, \sqrt{1 - \frac{x^2}{2}} \right)$$

با جایگذاری این مقادیر در رابطه بالا داریم:

$$1 = \frac{\frac{x^2}{2}}{x^2} + \frac{1 - \frac{x^2}{2}}{b^2}$$

$$\frac{b^2}{2} = 1 - \frac{x^2}{2}$$

$$b = \sqrt{2 - x^2}$$

بنابراین معادله بیضی با  $x$  ثابت به شکل زیر است:

$$1 = \frac{x'^2}{x^2} + \frac{y'^2}{2 - x^2}$$

از روی این معادله  $x'$  به دست می‌آید:

$$x'^2 = x^2 \left(1 - \frac{y'^2}{2 - x^2}\right)$$

به طور مشابه برای بیضی با  $y$  ثابت خواهیم داشت:

$$1 = \frac{x'^2}{2 - y^2} + \frac{y'^2}{y^2}$$

با جایگذاری  $x'$  به دست آمده در معادله بیضی با  $y$  ثابت به دست می‌آید:

$$1 = \frac{x^2 \left(1 - \frac{y'^2}{2 - x^2}\right)}{2 - y^2} + \frac{y'^2}{y^2}$$

$$y^2(2 - y^2)(2 - x^2) = x^2 y^2 (2 - x^2 - y'^2) + y'^2 (2 - y^2)(2 - x^2)$$

$$y'^2 = y^2 \frac{(2 - x^2)(2 - y^2 - x^2)}{(2 - x^2)(2 - y^2) - x^2 y^2}$$

$$y'^2 = y^2 \frac{(2 - x^2)(2 - y^2 - x^2)}{4 - 2x^2 - 2y^2}$$

$$y'^2 = y^2 \left(1 - \frac{x^2}{2}\right)$$

$$y' = y \sqrt{1 - \frac{x^2}{2}}$$

بنابراین  $y'$  را به دست آوردیم. طبق تقارن  $x'$  نیز به دست می‌آید. در نهایت داریم:

$$(x', y') = \left(x \sqrt{1 - \frac{y^2}{2}}, y \sqrt{1 - \frac{x^2}{2}}\right)$$

مرحله سوم:

تا الان نقاط به دایره واحد به مرکز مبدا مختصات نگاشت شده‌اند. اما از آنجا که ابعاد تصویر ورودی  $720 \times 480$  پیکسل است نیاز داریم ابتدا تمام نقاط دایره را به ربع اول صفحه مختصات بیاوریم. یعنی انتقال با مقدار +1 واحد نیاز دارد. در نهایت از انبساط استفاده می‌کنیم. در این اپلیکیشن از ضریب زیر استفاده شده است:

$$\begin{aligned} x' &= 240(x + 1) \\ y' &= 240(y + 1) \end{aligned}$$

بدین ترتیب دایره مورد نظر بزرگ می‌شود و قابلیت نمایش به کاربر را پیدا می‌کند

## بخش عملی

در این بخش به بخش توضیح پیاده‌سازی بخش نگاشت Fisheye در برنامه می‌پردازیم. برنامه پس از دریافت تصویر از دوربین، تابع fisheye که در شکل (۱) پیاده‌سازی آن نشان داده شده‌است را به ازای آن تصویر صدا می‌زند و پس از ساخت تصویر نهایی، آن را خروجی می‌دهد.

نحوه کار این تابع به این صورت هست روی تمام پیکسل‌های تصویر اولیه حرکت می‌کند. به ازای هر نقطه تابع mapX و mapY که خود سه مرحله توضیح داده شده را درون خود دارند برای طول و عرض هر نقطه صدا می‌زند. رنگ نقطه به دست آماده را با رنگ نقطه اولیه قبل از نگاشت دادن جایگذاری می‌کند. بدین ترتیب به تصویر نهایی می‌رسیم.

برای جزئیات کامل کد به مخزن GitLab پروژه به آدرس زیر مراجعه فرمایید:

<https://gitlab.com/daneshvar.amrollahi/fisheye>

```
img.Image fisheye(img.Image input){
    var w = input.width;
    var h = input.height;
    var output = img.Image(w, h);
    for (int i = 0 ; i < w ; i++)
        for (int j = 0 ; j < h ; j++)
        {
            int nx = mapX(i, j), ny = mapY(i, j);
            //print("$i $j $nx $ny");
            output.setPixel(ny + 115, nx, input.getPixel(i, j));
        }

    print('width and height are $w $h');

    return output;
}
```

شکل ۱: پیاده‌سازی تابع نگاشت