



گزارش تمرین اول بینایی ماشین

نام تهیه کننده: ملیکا نوبختیان

شماره دانشجویی: ۹۷۵۲۲۰۹۴

نسخه: ۱

۱- سوال اول

برای هر پیکسل یک تصویر مقدار $f(x, y)$ با میزان روشنایی منبع نوری که به صحنه تابیده می شود $i(x, y)$ و میزان روشنایی که توسط اشیا موجود در صحنه بازتاب می شود $r(x, y)$ مشخص می شود. به طور دقیق $f(x, y)$ به شکل زیر به دست می آید:

$$f(x, y) = i(x, y) r(x, y)$$

هر چه مقدار به دست آمده بیشتر باشد، میزان روشنایی یک پیکسل بیشتر خواهد بود و هر چه این مقدار کمتر باشد پیکسل روشنایی کمتری خواهد داشت. اشیایی مانند برف بازتاب زیادی دارند در حالی که یک مخمل سیاه بیشتر نور را جذب می کند. هم چنین محیط های آفتابی روشنایی بیشتر و محیط های تاریک روشنایی کمتری خواهند داشت.

هم چنین اگر در هنگام عکاسی shutter ای داشته باشیم که سرعت زیادی داشته باشد، در هنگام عکاسی چون دریچه دوربین برای مدت کمتری باز است، میزان نور و انرژی کمتری وارد می شود و تصویر تاریک تر خواهد بود و بالعکس.

۲- سوال دوم

زبان اشاره راه ارتباطی با ناشنویان است و ناآشنا بودن با این زبان توسط بسیاری از مردم، می تواند در موقعیت های مختلف برای این قشر از جامعه مشکل ارتباطی ایجاد کند. یک کاربرد بینایی ماشین می تواند در این زمینه باشد که حرکات اشاره دست ناشنویان را تبدیل به معادل گفتاری یا نوشتاری آن کند و به این شکل ارتباط متقابل بین این افراد و دیگر افراد جامعه صورت گیرد.

۳- سوال سوم

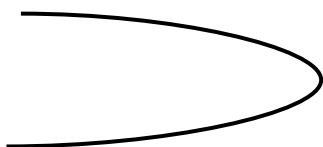
۳-۱- الف

در ابتدا فرض می کنیم ابعاد حسگر آرایه ای ما همان $1000 * 1000$ است و ابعاد حسگر خطی ما $1000 * 1$ است. اگر ابعاد تصویر ما هم همان $1000 * 1000$ باشد، حسگر آرایه ای تنها با یک بار و بدون هیچ حرکتی تصویر کل صحنه را خواهد گرفت. پس در این صورت ما تمام صحنه رو یک جا خواهیم گرفت بدون اینکه قسمت های مختلف آن را در بازه های زمانی مختلف گرفته باشیم.

اما در مورد حسگر خطی این موضوع متفاوت است. اگر فرض کنیم حسگر خطی از بالا به پایین حرکت خواهد کرد تا کل تصویر را بگیرد، برای این کار باید 1000 حرکت انجام دهد. در نتیجه هر کدام از این 1000 قسمت در زمان های مختلفی گرفته شده اند که در این حالت ممکن است چیزی که با حسگر خطی گرفته ایم دقیقاً همان تصویر اولیه کلی نباشد.

۲-۳- ب

حسگر آرایه‌ای چون تنها با یک بار از کل تصویر عکس می‌گیرد، تصویری که از آن خواهیم داشت تنها تصویر همان لحظه و بدون هیچ تغییری خواهد بود. اما در مورد حسگر خطی این موضوع متفاوت است. فرض می‌کنیم که حسگر خطی از بالا به پایین برای گرفتن عکس حرکت کند. برای مثال پره بالایی آسیاب در تصویر را در نظر می‌گیریم. چون حسگر خطی از بالا شروع به گرفتن تصویر می‌کند آن قسمت از تصویر به حالت عادی ثبت می‌گردد. اما هنگامی که حسگر خطی برای گرفتن قسمت پایین تر آن پره به سمت پایین حرکت می‌کند، آن پره هم به سمت راست حرکت کرده‌است و قسمت پایینی مورد نظر ما هم حالا هم به سمت راست رفته‌است و هم کمی بالاتر رفته‌است. تا زمانی که کلا پره صاف شود این بالاتر رفتن ادامه دارد و سپس این تغییر حرکت به سمت راست و پایین خواهد بود. به طور کلی می‌شود گفت تصویری که از این پره آسیاب ثبت می‌شود یک قوس دارد که قوس آن به سمت راست است:



۴- سوال چهارم

برای خواندن تصویر با cv2.imread به شکل زیر عمل می‌کنیم:

```
imm = cv2.imread('background.png', cv2.IMREAD_COLOR)
```

پارامتر اولی که این تابع برای خواندن تصویر می‌گیرد، filename یا همان اسم فایل و مسیر آن است. پارامتر دوم در این تابع flag نام دارد که چگونگی خواندن تصویر را مشخص می‌کند. در اینجا چون ما می‌خواستیم تصویر را به صورت رنگی و در سه کانال بخوانیم از cv2.IMREAD_COLOR استفاده کردیم. این مقدار default است که این تابع همیشه برای خواندن تصویر دارد. از دیگر حالت‌های این flag می‌توان به cv2.IMREAD_GRAYSCALE اشاره کرد که تصویر را به شکل سیاه و سفید می‌خواند.

برای تغییر ترتیب کانال‌ها به شکل زیر عمل می‌کنیم:

```
imm = cv2.cvtColor(imm, cv2.COLOR_BGR2RGB)
```

پارامتر دوم این تابع conversion code ای است که مشخص می‌کند چطور تصویر تغییر کند. در اینجا ما از cv2.COLOR_BGR2RGB استفاده کردیم که ترتیب کانال‌ها از BGR به RGB تغییر می‌دهد.

Resize تصویر را به شکل زیر انجام می‌دهیم:

```
imm = cv2.resize(imm, (570, 290))
```

برای کشیدن مستطیل باید از خط برای کشیدن آن استفاده کنیم و برای کشیدن خط باید نقطه آغازین و پایانی آن را مشخص کنیم. من به صورت زیر این نقاط را مشخص کردم و رنگ قرمز و ضخامت ۳ را برای کشیدن آن انتخاب کردم:

```
start_points = [(0, 0), (0, 0), (560, 280), (560, 280)]
end_points = [(0, 280), (560, 0), (560, 0), (0, 280)]
```

```
color = (0, 0, 255)
thickness = 3
```

سپس به صورت زیر مستطیل را رسم کردم:

```
for i in range(4):
    cv2.line(img, start points[i], end points[i], color, thickness)
```

پارامتر اول cv2.line تصویر موردنظر، پارامتر دوم و سوم نقاط آغازین و پایانی خط، پارامتر چهارم رنگ و پارامتر آخر میزان ضخامت خط را نشان می‌دهد.

برای رسم راس‌های مستطیل باید مراکز آن‌ها را مشخص کنیم که مراکز آن‌ها همان نقاط آغازین و پایانی در خطوط قسمت قبل هستند. برای هر کدام از راس‌ها رنگ مختلفی در نظر گرفتیم و در یک آرایه قرار دادیم. به شکل زیر این راس‌ها رسم کردم:

```
centers = [(0, 0), (0, 280), (560, 0), (560, 280)]
colors = [(0, 0, 0), (255, 0, 0), (0, 255, 0), (128, 0, 128)]
for center, colour in zip(centers, colors):
    cv2.circle(img, center, 3, colour, 5)
```

پارامتر اول در اینجا تصویر، پارامتر بعد مرکز دایره، پارامتر بعد طول شعاع دایره، پارامتر بعد رنگ دایره و پارامتر آخر ضخامت خطوط آن را نشان می‌دهد. در نهایت تصویر زیر به دست می‌آید:



برای کشیدن تصویر end.png لازم است ابعاد ما به گونه‌ای باشد که طول تصویر به ۷ قسمت تقسیم شود و عرض آن به دو قسمت تقسیم شود. برای همین از ابعاد 290×560 برای تصویر استفاده کردیم. در قطعه کد زیر points نشان‌دهنده ۸ نقطه موجود در طول است و heads راس‌های این مستطیل را نشان می‌دهند که از آن‌ها به هر ۸ نقطه تصویر رسم خواهد شد:

```
points = [(0, 145), (80, 145), (160, 145), (240, 145), (320, 145), (400, 145),
(480, 145), (560, 145)]
heads = [(0, 0), (0, 290), (560, 0), (560, 290)]
color = (0, 0, 0)
for head in heads:
```

```
for point in points:  
    cv2.line(new_img, head, point, color, 1)
```

در نهایت تصویر نهایی نیز به شکل زیر است:

