

«به نام خدا»

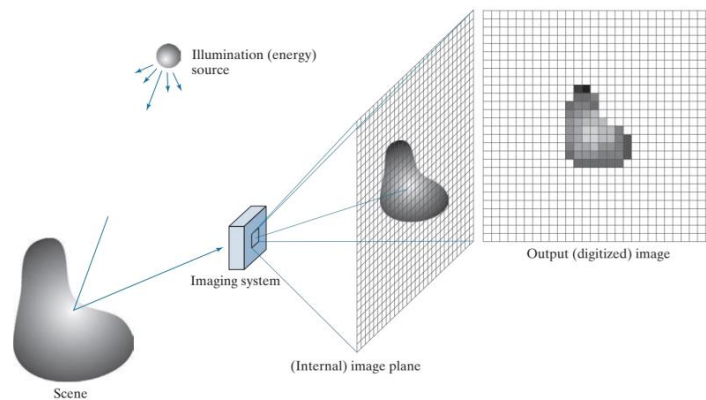
استاد: دکتر محمدرضا محمدی
درس مبانی بینایی کامپیوتر

نام: فاطمه زهرا بخشنده
شماره دانشجویی: 98522157

گزارش تمرین 1:

سوال اول:

برای ثبت یک تصویر دیجیتال، نور به محیط می تابد، محیط نور را بازتاب می کند و ما به صورت گسسته خروجی را ثبت می کنیم. مانند تصویر زیر:



شکل 1 ثبت تصویر دیجیتال

تصویر دیجیتال بدست آمده را با $f(x, y)$ نمایش می دهیم. $f(x, y)$ با دو جزء مشخص میشود:

- میزان روشنایی منبع نوری که به صحنه تابیده شده است.
- میزان روشنایی که توسط اشیاء موجود در صحنه منعکس میشود.

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

$$0 \leq i(x, y) < \infty, \quad 0 \leq r(x, y) \leq 1, \quad 0 \leq f(x, y) < \infty$$

برای مثال این معیار ها برای برخی محیط ها و اجسام مختلف به صورت زیر است:

شیء	r	محیط	$i(lm/m^2)$
مخمل سیاه	0.01	روز آفتابی	$> 90,000$
فولاد	0.65	روز ابری	$< 10,000$
دیوار سفید	0.80	شب مهتابی	≈ 0.1
نقره	0.90	دفتر اداری	$\approx 1,000$
برف	0.93		

$$L_{min} \leq f(x, y) \leq L_{max}$$

$$10 \leq f(x, y) \leq 1000$$

در هنگام کوانتیزه کردن، باید حواسمان به min و max باشد. اگر max خیلی عدد کوچکی باشد، پیکسل ها اشباع می شوند و به 255 میرسند. اگر آن را خیلی بزرگ بگیریم، تصویر تیره ای ثبت می شود. پس باید متناسب با محیط تنظیم شود.

برای مثال تصویر زیر با پارامترهای مناسب کوانتیزه شده است:



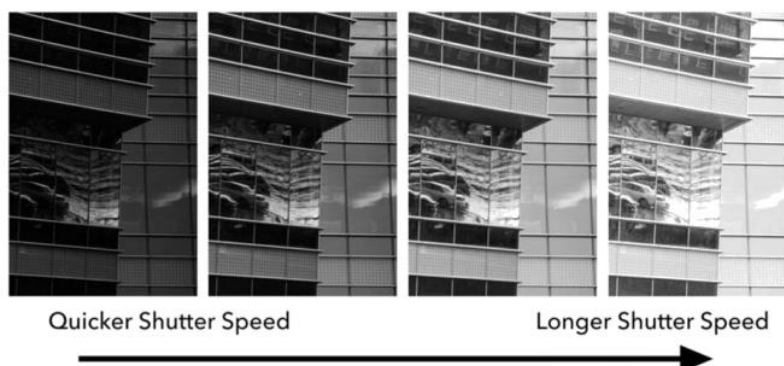
شکل 2

اما اگر L_{max} را خیلی کم بگیریم تصویر به صورت زیر در میاید. پیکسل های آن اشباع می شوند یعنی روشنایی پیکسل های آن خیلی زیاد می شود.



شکل 3

علاوه بر این ها، **سرعت shutter** نیز می تواند روی روشنایی پیکسل های تصویر ثبت شده اثر بگذارد. سرعت دریاچه مدت زمانی است که دریاچه دوربین باز است و نور را بر روی حسگر دوربین قرار میدهد.



شکل 4 تصویر ثبت شده با shutter speed های مختلف

برای مثال فرض کنیم در ثبت تصویر بالا پارامتر های قبلی را ثابت می گیریم. در تصویر سمت چپ دریاچه زودتر بسته شده و انرژی کمتری را جذب کرده است. برعکس در تصویر سمت راست دریاچه کند بسته شده و کلی نور را جذب کرده است، بدون آنکه پارامتر های L_{min} و L_{max} را به صورت adaptive تنظیم کند.

حساسیت سنسور نسبت به نور نیز مهم است. برخی از سنسورهای قدیمی تغییرات کم نور را حس نمی کنند. طبیعتاً هرچه سنسور قوی تر و دقیق تر باشد، می تواند نور های خیلی دقیقی را ثبت کند.

در دوربین ها چیزی به نام نوردهی داریم. مقدار نوری که توسط دوربین دریافت می شود و بر سنسور یا نگاتیو دوربین اثر می گذارد، توسط دو عامل مقدار گشودگی دیافراگم و سرعت شاتر تعیین می گردد. یکی دیگر از عوامل تاثیر گذار بر میزان نوردهی، میزان حساسیت سنسور یا نگاتیو دوربین است. بنابراین، معقوله نوردهی ارتباط مستقیمی با میزان حساسیت سنسور، اندازه دیافراگم و سرعت شاتر دارد که به آن "EV" یا همان میزان نوردهی گفته می شود. هر چه EV بالاتر باشد، نشان دهنده موقعیتی بسیار روشن است که در این حالت می بایست سرعت شاتر را بالا برد و عدد دیافراگم را افزایش داد یعنی گشودگی دریاچه دیافراگم را کاهش داد و یا میزان حساسیت (ISO) را کم کرد تا از (overexposure) نوردهی بیش از حد جلوگیری کرد. همچنین برای جلوگیری از underexposure باید مخالف آنچه که بالا گفته شد عمل کرد.

زاویه لنز دوربین نسبت به منبع نور و فاصله آن از جسم نیز میتواند روی جذب نور و در نتیجه میزان روشنایی پیکسل های تصویر اثر بگذارد.

منابع: [لینک](#) و اسلاید ها

سوال دوم:

یک ایده کاربردی که بنظرم در کشور خودمان می توانیم آن را پیاده سازی کنیم: من به شخصه بسیاری از خریدهای خودم را از دیجی کالا انجام می دهم. در این میان، هر زمان که می خواهم کالای مورد نظرم را انتخاب کنم و نظرات کالاهای مختلف را بررسی می کنم، افراد زیادی از معیوب و ناقص بودن کالایی که بدستشان رسیده یا حتی اشتباهی بودن کالا گله می کنند. در اینجا می توان از بینایی ماشین استفاده کرد تا درصد اینگونه خطاها و کالاهای مرجوعی به حداقل برسد.

به این صورت که با استفاده از بینایی ماشین در هنگام آماده کردن کالاها و حتی در آخر با اسکن کردن محتوای درون جعبه از درست بودن آن ها اطمینان پیدا کنیم. برای این کار باید در هنگامی که کالاها آماده می شوند از آن ها تصویر برداری شود، تصویر به رایانه اختصاصی منتقل شود و فرایند های نرم افزاری بینایی و پردازش تصویر روی آن ها صورت بگیرد و تجزیه و تحلیل هایی مانند نام و رنگ و اندازه و شمارش تعداد صحیح کالا طبق سفارش مورد نظر روی آن ها انجام شود. سپس در صورت وجود مشکل سیگنالی به سامانه کنترلی فرستاده شود تا سفارش های اشتباه از خط خارج شده و رفع عیب شوند.

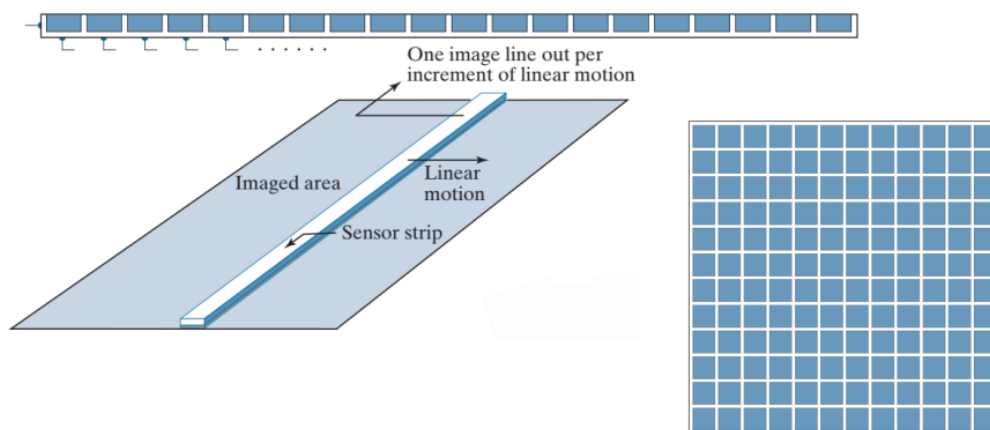
یک سری ایده دیگر که به ذهنم می رسد که می توانیم آن ها را پیاده سازی کنیم، استفاده از بینایی ماشین در تشخیص اصل بودن کالاها از روی شکل یا بارکد آن ها، یا تشخیص دزدی ها و تقلب هاست.

برای مثال در اسلایدها هم به کاربرد بینایی در تشخیص زاویه دید افراد اشاره شد. بنظرم از این کاربرد می توانیم در تشخیص تقلب دانش آموزان در امتحانات استفاده کنیم. (یکم خبیثانه شد 😊)

سوال سوم:

الف) برای ثبت تصویر دیجیتال، استفاده از تنها یک حسگر نوری دارای مزیت ارزان بودن است زیرا حرکت مکانیکی با دقت بالا قابل کنترل است. اما این روش کند است و به راحتی قابل حمل نیست.

به همین دلیل از چندین حسگر نوری استفاده می کنیم. حسگر آرایه ای (global shutter) و حسگر خطی (rolling shutter) هر دو به این شکل طراحی شده اند.



شکل 5 حسگر خطی و آرایه ای

حسگر خطی در یک زمان یک سطر را اسکن می کند. سپس جابجا می شود و به ترتیب سطرهای بعدی را ساکن می کند. اگر طبق سوال یک تصویر 1000×1000 داشته باشیم، و حسگر خطی ما دارای 1000 سنسور باشد، آنگاه برای ثبت تصویر، این 1000 سنسور باید 1000 بار جابجا شوند.

پس از نظر سرعت تا حدی نسبت به حسگر یک سنسوره پیشرفت کرده است اما گران تر شده است. ولی هنوز هم کمی کند است چون باید در یک جهت جابجا شود.

نکته: دوربین ها با شاتر چرخشی کیفیت تصویر خوبی دارند. به همین دلیل برای عکس برداری با رزولوشن بالا مناسب هستند. اما سرعت تصویر برداری در آنها پایین تر است. از این رو برای عکس برداری از اجسام متحرک چندان مناسب نیستند.

حسگر آرایه ای در یک زمان یک ناحیه دوبعدی را اسکن می کند. اگر طبق سوال یک تصویر 1000×1000 داشته باشیم، و حسگر آرایه ای ما دارای 1000,000 سنسور باشد (1000×1000) آنگاه حسگر نیازی به جابجایی ندارد.

پس از نظر سرعت نسبت به حسگرهای قبلی خیلی بهتر است و همینطور خیلی گرانتر است.

نکته: تصویر برداری یکباره در سنسورهای Global باعث می شود که نویز تصویر در این دوربین ها بیشتر باشد. این دوربین ها نسبت به دوربین های Rolling Shutter دارای رزولوشن کمتری هستند. به دلیل نوع تصویر برداری این دوربین ها، ساخت آنها در رزولوشن های بالا خیلی گران تمام می شود.

در یک مقایسه کلی می توان گفت حسگر خطی نسبت به حسگر آرایه ای جابجایی بیشتری دارد و کندتر است، ولی ارزان تر است. حسگر آرایه ای سریع تر و گران تر است. دوربینی که حسگر آرایه ای داشته باشد تصاویر و فیلم هارا دقیق تر ثبت میکند چون به صورت فریز در یک لحظه کل مقادیر را ثبت می کند. به همین جهت هم به آن global shutter می گوییم. (اگر در زمان استفاده از دوربین با حسگر خطی جسم با سرعت زیادی حرکت کند یا ما در حال حرکت باشیم، آنگاه تصویر انحنای می گیرد و دقیق ثبت نمی شود)

کاربردها: در بینایی ماشین در موارد زیادی نیاز داریم از اشیا در حال حرکت عکس برداری کنیم. عموماً در این کاربردها از دوربین ها با شاتر گلوبال استفاده می کنیم. به عنوان نمونه یکی از موارد کاربرد این دوربین ها ، خواندن حروف و اعداد از روی محصولات در حال حرکت است. دوربین های Rolling Shutter برای عکس برداری از موارد غیر متحرک یا کند مناسب هستند. مخصوصاً وقتی که نیاز به کیفیت تصویر بالایی داشته باشیم. به عنوان نمونه یکی از کاربردهای این دوربین ها در سیستم های اندازه گیری می باشد.

ب) همانطور که گفتیم دوربینی که حسگر آرایه ای داشته باشد تصاویر را دقیق تر ثبت میکند چون به صورت فریز در یک لحظه کل مقادیر را ثبت می کند. پس حسگر آرایه ای با سرعت زیاد تصویر را ثبت کرده و در این تصویر پره های آسیاب صاف ثبت می شوند. اگر با این دوربین از آسیاب فیلم بگیریم نیز حرکت آسیاب نشان داده شده و چرخش پره ها در آن همانطور که هست نشان داده می شود و پره ها صاف هستند.

اما همانطور که گفتیم اگر از حسگر خطی استفاده کنیم و جسم با سرعت زیادی حرکت کند، آنگاه تصویر انحنای می گیرد و دقیق ثبت نمی شود. پس اگر با این دوربین تصویر بگیریم پره ها به صورت منحنی و در برخی بخش های تصویر به صورت کشیده و مایل ثبت می شوند و تصویر تخریب می شود. از آن جایی که پره های آسیاب در جهت عقربه های ساعت در حرکت هستند، در تصویر به شکل منحنی، کشیده شده و مایل به سمت چپ نشان داده می شوند.

منابع: [لینک](#) و اسلاید ها

سوال چهارم:

کد کامل تمام قسمت ها در فایل HW1.ipynb در همین پوشه قرار دارد.

1.4- با استفاده از کامند های مناسب openCV را نصب می کنیم.

```
pip install opencv-python
```

```
pip install opencv-contrib-python
```

2.4- برای نمایش هر پیکسل در یک تصویر در کانال رنگی RGB ، نیاز به سه عدد صحیح در محدوده صفر تا 255 داریم که عدد اول بیانگر میزان قرمز بودن رنگ، عدد دوم بیانگر میزان سبز بودن رنگ و عدد سوم بیانگر میزان آبی بودن رنگ است.

در OpenCV میتوان به کمک تابع imread یک تصویر را از روی حافظه خواند. این تابع دو پارامتر میگیرد. پارامتر اول یک رشته است که بیانگر آدرس تصویر در حافظه و پارامتر دوم بیانگر این است که تصویر به چه صورت خوانده شود. اگر این پارامتر صفر مقداردهی شود یعنی تصویر به صورت یک کاناله خوانده شود. به صورت پیشفرض همان صورت که تصویر ذخیره شده است را در نظر می گیرد.

cv2.IMREAD_COLOR: بارگذاری یک تصویر رنگی را مشخص می کند. هر گونه شفافیت تصویر نادیده گرفته خواهد شد. این flag پیش فرض است. از طرف دیگر، می توانیم مقدار صحیح 1 را برای این flag ارسال کنیم.

cv2.IMREAD_GRAYSCALE: مشخص می کند که یک تصویر در حالت خاکستری بارگذاری شود. از طرف دیگر، می توانیم مقدار صحیح 0 را برای این flag ارسال کنیم.

cv2.IMREAD_UNCHANGED: مشخص می کند که یک تصویر از جمله کانال آلفا بارگیری شود. از طرف دیگر، می توانیم مقدار صحیح 1- را برای این flag ارسال کنیم.

اگر تصویر قابل خواندن نباشد (به دلیل از دست رفتن فایل یا فرمت نامعتبر یا ...)، این متد یک ماتریس خالی برمیگرداند.

```
path = r'images\background.png'
```

```
img = cv2.imread(path, 1)
```

3.4- متد `cv2.cvtColor()` برای تبدیل تصویر از یک فضای رنگی (color space) به فضای دیگر استفاده می شود. بیش از 150 متد تبدیل color space در OpenCV موجود است.

```
image_RGB = cv2.cvtColor(image, cv2.COLOR_BGR2RGB )
```

4.4- تاپلی که برای تعیین اندازه تصویر جدید به این متد پاس می دهیم از ترتیب (width, height) پیروی می کند.

```
image_RGB_resized = cv2.resize(image_RGB, (570, 290))
```

5.4- متد `cv2.line()` به صورت `cv2.line(image, start_point, end_point, color, thickness)` است. برای کشیدن مستطیل باید 4 تا خط روی تصویرمان بکشیم.

من میخوام مستطیل را در وسط تصویر بکشم. برای این کار ابتدا x و y شروع و پایان مستطیل را پیدا می کنم. به طوری که طول مستطیل 560 و عوض آن 270 باشد و وسط تصویر بیفتد.

```
start_x = int((570 - 560) / 2) #5
end_x = start_x + 560          #565
start_y = int((290 - 280) / 2) #5
end_y = start_y + 280          #285
```

سپس با مقداردهی نقاط و بقیه پارامترها مستطیل را میکشیم.

```
for i in range(4):
    rectangle = cv2.line(rectangle, points[i - 1], points[i], color, thickness)
```

6.4- متد `cv2.circle()` به صورت زیر است:

```
cv2.circle(image, center_coordinates, radius, color, thickness)
```

پس از مقداردهی پارامترها با رنگ های مختلف روی راس های مستطیل لوپ می زنیم و دایره می کشیم.


```
for i in range(4):
    vertices = cv2.circle(vertices, points[i], radius, colors[i], thickness)
```

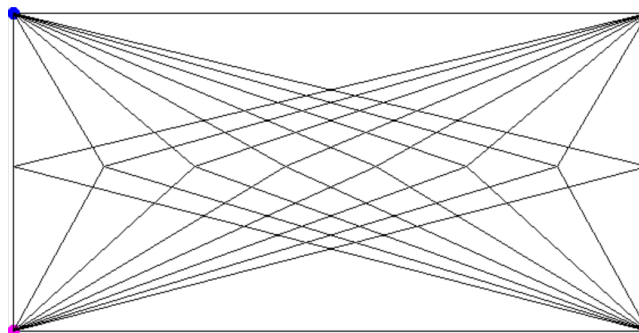
7.4- برای ساختن یک تصویر سفید جدید می توانیم از کد زیر استفاده کنیم. و سپس شکل را روی آن بکشیم.

```
blank_image = np.zeros((278, 557, 3), np.uint8) + 255
```

ولی از آن جایی که سایز تصویر تقریباً هم سایز مستطیل تصویر قبلی است، حدس می زنم منظورتان این بوده که این شکل را در درون مستطیل قبلی بکشیم. پس من با این فرض جلو می روم.

پس از مشخص کردن مقدار برخی پارامتر ها، با استفاده از loop، 4 نقطه گوشه را که در آرایه points ذخیره شده اند، به 8 نقطه وسط وصل می کنیم تا شکل مورد نظر ایجاد گردد.

```
for p in points:
    for x in range(start_x, end_x + 1, int((end_x - start_x) / 7)):
        y = int((end_y - start_y) / 2)
        cv2.line(end_image, p, (x, y), color, thickness)
```



8.4- این تابع به صورت cv2.imwrite(filename, image) است. تصویر را در پوشه images کنار بقیه عکس ها ذخیره می کنیم.

```
cv2.imwrite("images/mypic.jpg", end_image)
```

منابع: [geeksforgeeks](https://www.geeksforgeeks.org/)