

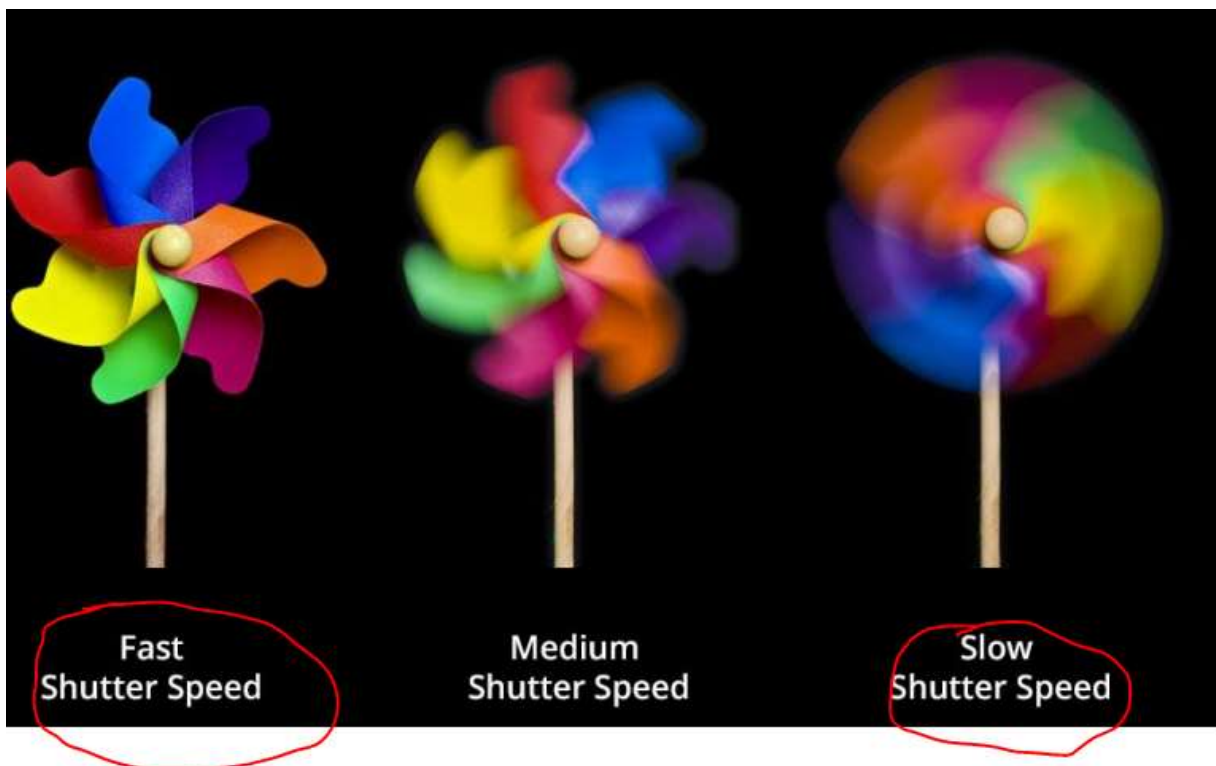
به نام خدا

امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

توضیحات تمرین سری اول - بینایی کامپیوتر

نکات :

(۱) برای سوالاتی که در رابطه با سرعت شاتر بودند؛ از این اسلاید استفاده کرده ایم :



یعنی هنگامی که تصویر واضح میخواستیم و مدت زمانی که دریچه باید باز باشد

کم در نظر بگیریم از **fast shutter speed** و در عکس موارد از **slow**

shutter speed استفاده کرده ام و داخل توضیحات بر این اساس نوشته ام.



تعریف جزوه هم این مورد میباشد. در هر سوال هر دو را نوشته ام.

(۱)

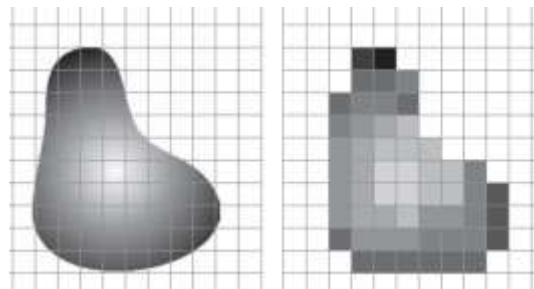
با توجه به اینکه در حالت اولیه ۲۵۶ سطح روشنایی داریم و آنرا به ۶۴ سطح کوانتیزه کردیم بازه جدید به صورت ۴ تایی میشود. (از حاصل تقسیم ۲۵۶ بر ۶۴ این مقدار را به دست آوردیم)

یعنی هر ۴ شدت روشنایی در حالت اولیه این بار یک حالت را تشکیل میدهند. به طور مثال مقدار ۰ تا ۳ یک سطح و مقدار ۴ تا ۷ یک سطح و ... به همین ترتیب.

از اثرات این کار این است که کنتراست کاهش پیدا میکند؛ یعنی اگر شدت روشنایی یک پیکسل برابر ۳ و پیکسل کناری آن برابر ۲ باشد این دو پیکسل بعد از کوانتیزه کردن یک شدت روشنایی دارند و به همین منوال هر ۴ تایی که در یک سطح قرار میگیرند دارای یک شدت روشنایی میشوند.

همچنین مطابق با شکل زیر که در جزوه داشتیم؛ یک سری جزئیات تصویر نیز دیگر دقیق ثبت نمیشوند و رزولوشن محدودی داریم؛ که به دلیل تصویر برداری دیجیتال است.

و به طور کلی کیفیت کاهش میابد که در شکل زیر نیز مشخص است.



(۲)

برای گرفتن عکس واضح از پرنده در حال پرواز باید **fast shutter speed** استفاده کنیم یعنی مدت زمانی که دریچه دوربین باز است کم باشد و عکس پرنده را در یک لحظه ثبت کند که در این صورت جزئیات قابل مشاهده است. (با توجه به بیان داخل اسلاید میتوان گفت سرعت شاتر باید پایین باشد یعنی مدت زمان کمی دریچه باز باشد).

اگر به صورت **slow shutter speed** یعنی مدت زمان زیادی دریچه باز باشد حرکت پرنده را در مدت زمانی که دریچه باز است ثبت میکند و عکس پرنده حالت کشیده پیدا میکند.

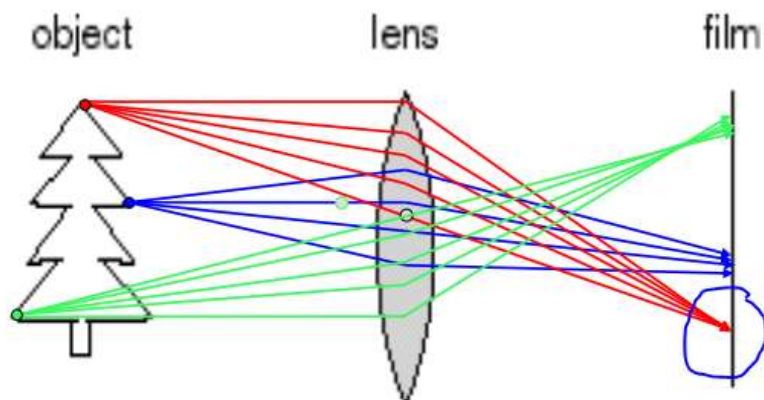
(۳)

برای اینکه یک تصویر واضح از یوزپلنگ در حال حرکت بگیریم باید مدت زمان باز بودن دریچه پایین باشد تا در یک لحظه عکس یوزپلنگ را ثبت کند و باید از **fast shutter speed** استفاده کنیم. به طور کلی و با توجه به معنی گفته شده میتوان گفت سرعت شاتر باید پایین باشد. (یعنی مدت زمان پایین باشد)

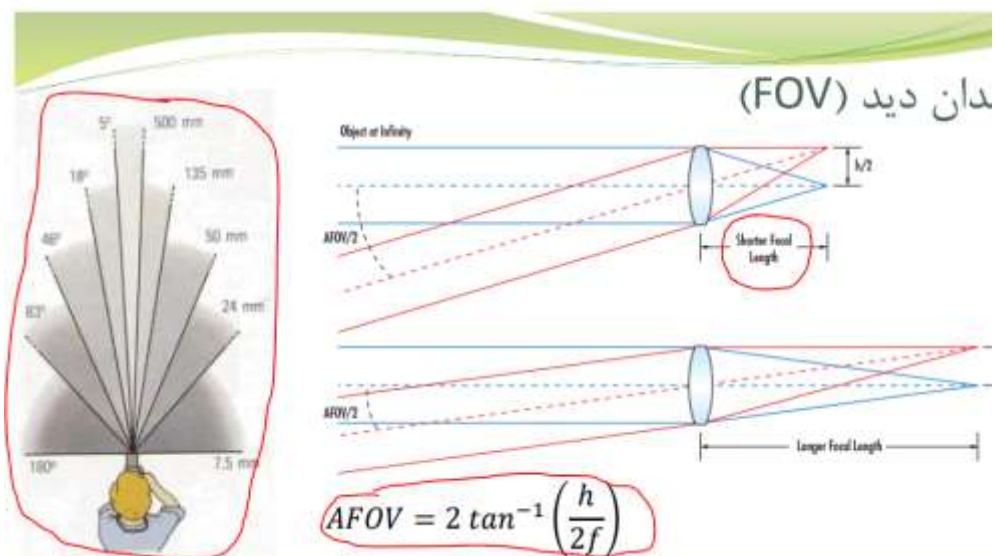
برای تار افتادن پس زمینه باید عمق میدان را مورد بررسی قرار دهیم. طبق فرمول $1/f = 1/v + 1/u$ باید مقادیر v, u را به نحوی پیدا کنیم که تصویر یوزپلنگ واضح شود یعنی نقطه همگرایی اشعه های ساطع شده از آن دقیقاً در دوربین ما همگرا شوند. و بدین ترتیب بقیه نقاط با کم و زیاد شدن v, u نقطه همگرایی قبل یا بعد از دوربین ما دارند که باعث میشود تصویر تار شود.

• تنها اشعه های نوری نقاطی که در فاصله u از لنز باشند در صفحه ای به فاصله v از لنز همگرا (متمرکز) می شوند

• نقاط با فاصله های دیگر دچار تاری خواهند شد



در این سوال اشعه های بازتاب شده از یوزپلنگ پس از برخورد به لنز همانند رنگ قرمز در این تصویر در یک نقطه و در دوربین ما همگرا شوند و مابقی نقاط در چند نقطه برسند و تار میشوند. در رابطه با وسعت باید بحث میدان دید را مورد بررسی قرار داد. خواسته ما این است که منطقه وسیعی را در کادر خود داشته باشیم؛ در نتیجه باید زاویه بیشتری را مد نظر قرار دهیم همچنین با توجه به فرمول و تصاویر زیر باید از فاصله کانونی کمتر استفاده کنیم.



با توجه به تصویر سمت چپ ، با افزایش شعاع دید ما زاویه نیز افزایش داشته همچنین با توجه به شکل بالا و فرمول آورده شده برای افزایش زاویه باید مقدار فاصله کانونی کاهش یابد.

(۴) این سوال نیاز به توضیح ندارد

(۵)

* برای این سوال در هر خط تقریبا کامنت گذاری صورت گرفته و توضیحات را نوشته ام.

تفاوت opencv , matplotlib :

تفاوت اصلی این دو کتابخانه و علت نمایش تصویر به صورتی که در صورت سوال (کد شما) بود به این دلیل است که ترتیب سه رنگ اصلی در کانال این دو کتابخانه متفاوت است.

در کتابخانه **matplotlib** به صورت **RGB** است و در کتابخانه **opencv** به صورت **BGR** میباشد.

ما تصویر را با استفاده از کتابخانه **opencv** خوانده ایم در نتیجه در متغیر **I** که تعریف کرده ایم کانال سه رنگ اصلی به صورت **BGR** است و هنگامی که با استفاده از کتابخانه **matplotlib** تصویر را نمایش میدهیم به صورتی است که در سوال نمایش داده شده است.

برای حل این موضوع با استفاده از کتابخانه **opencv** باید کانال های رنگی را مطابق با **RGB** کنیم و برای این کار از تابع **cvtColor** استفاده میکنیم.

توضیحات کد Mode ها :

ابتدا تصویر را دوباره خواندیم و در متغیر **img** قرار دادیم و سپس کانال های رنگی آنرا به **RGB** تغییر دادیم (برای سهولت در کدزنی)

در این ۴ قسمت باید انواع تغییرات در تصویر را اعمال کنیم.

برای این کار از تابع **flip** استفاده میکنیم. به این تابع عکس و نوع تغییری که میخواهیم را پاس میدهیم. اگر پارامتر دوم را 0 بگذاریم به صورت عمودی (**vertical**) تغییر میابد و اگر مقدار آنرا

برابر 1 بگذاریم به صورت افقی (horizontal) تغییر میابد و عکس جدید را در یک متغیر قرار میدهیم.

برای ترکیب دو عکس با هم از تابع **concatenate** استفاده میکنیم.

در حالت ترکیب عمودی تصویری که بالاتر باشد را به عنوان پارامتر اول و عکس پایین تر را پارامتر دوم پاس میدهیم. سپس مقدار $axis = 0$ قرار میدهیم.
برای ترکیب افقی دو عکس پارامتر اول سمت چپ قرار میگیرد و پارامتر دوم سمت راست قرار میگیرد و $axis = +1$ قرار میدهیم.

قسمت logo :

برای این قسمت به صورت دستی و امتحان کردن یک پنجره را پیدا کردیم و آنرا زرد رنگ کردیم.
برای زردرنگ کردن هم مقدار قرمز و سبز را که تشکیل دهنده زرد هستند ۲۵۵ دادیم و مقدار آبی را ۰ قرار دادیم.

(۶)

عکس لوگو را خواندیم. سپس آنرا برای سهولت در کدزنی و فهم به RGB تبدیل کردیم و نمایش دادیم.

برای اینکه عکس فقط در کانال آبی باشد مقدار کانال قرمز و سبز را برابر 0 قرار دادیم.

برای مابقی کانال ها و نمایش تنها رنگ هم از همین الگوریتم استفاده کردیم و مابقی کانال ها را صفر گذاشتیم.