

1. کم و زیاد شدن دریچه در این دوربین چه اثری دارد؟

اگر دریچه بزرگ باشد :

- نور منعکس شده در بخش بیشتری از تصویر اثر میگذارد.
- -تصویر تار خواهد بود.

اگر دریچه کوچک باشد :

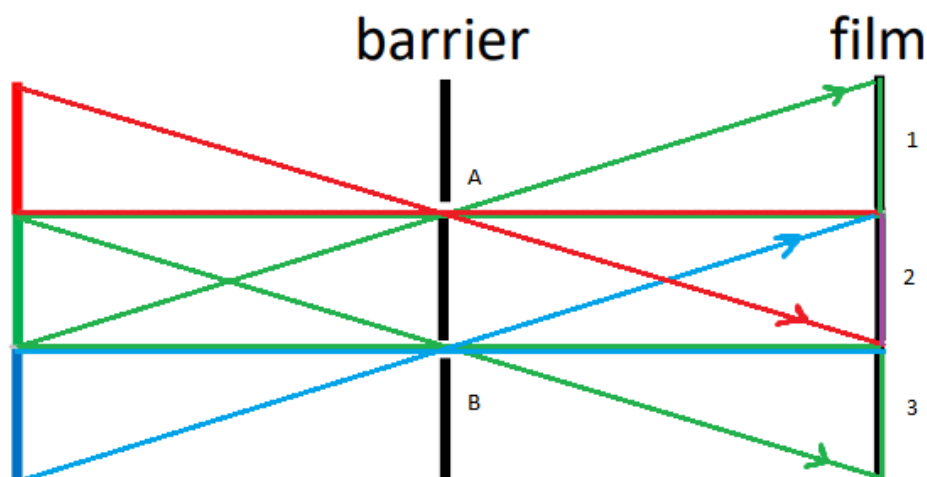
- تار شدن را کاهش میدهد اما مقدار نور وارد شده به دوربین را کم میکند.
- همچنین باعث پراکندگی نور میشود.(تار شدن تصویر)

همان طور که در شکل پایین نمایش داده شده است، نور قرمز رنگ تنها از دریچه A می تواند روی پرده ما تأثیر بگذارد و اگر از دریچه B خارج شود از محوطه پرده ما خارج می شود. پس طبق زیر نور ساطع شده از پارت قرمز رنگ، ناحیه ۲ را در فیلم ما قرمز می کند. پارت سبز رنگ از هر دوی دریچه های A, B می تواند روی فیلم تأثیر بگذارد و ناحیه های ۱ و ۳ را سبز رنگ می کند. ناحیه ابی رنگ نیز شرایطی مثل ناحیه قرمز دارد و فقط می تواند از ناحیه B عبور کند و پارت 2 را ابی می کند.

ناحیه 1 : سبز

ناحیه 2 : آبی + قرمز = بنفش

ناحیه 3 : سبز



2. اصولا موضوعات مربوط به DOF متعلق به تصاویری است که توسط دوربین های لنز دار گرفته شدند ولی در حال حاضر دوربین های ما هم دارای لنز هستند و هم دارای دریچه ولی مطمئنا عکس مبلوطه متعلق به دوربین pinhole نیست.

برای درک دلیل این عکس بهتر است نحوه کار کردن یک دوربین لنز دار را بررسی کنیم.

رابطه ی زیر برای لنز ها برقرار است که در آن :

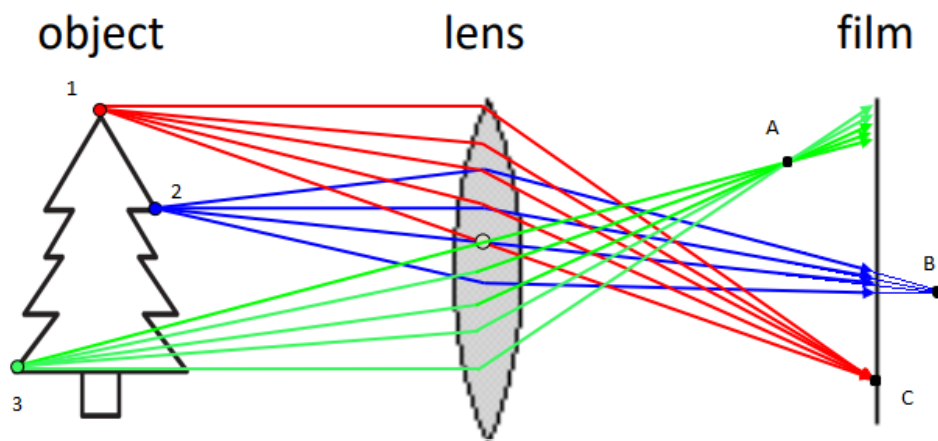
فاصله کانونی عدسی : f

فاصله جسم تا عدسی : u

فاصله تصویر تا عدسی : v

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$$

حال بیاید تصویر زیر برای جسم زیر روی پرده نشان دهیم:



همانطور که مشخص است، چون جسم ما هم دارای بعد z هست، همه اجزا آن روی پرده نمیفتند و مثلا نقطه 2 به روی نقطه B میفتد که پشت پرده قرار دارد و همین اتفاق برای نقطه 3 نیز میفتد و تصویر آن تار ذخیره میشود!

اتفاقی که در عکس صورت سوال نیز افتاده است دقیقا مانند بالا است. پارت هایی که تار افتادند نقاطی هستند که v آنها متفاوت بوده است و روی پرده نیفتاده اند و به همین علت تار شده اند، در حالی که عبارت وسط در تصویر سوال، دقیقا روی پرده افتاده است و واضح دیده میشود.

برای بهبود تصویر راهکار مورد استفاده است، استفاده است دریچه برای افزایش یا کاهش عمق میدان تصویر ماست. با استفاده از یک دریچه خیلی بزرگ مانند $f/1.4$ ما میتوانیم به نور های بسیار زیادی اجازه ورودی دهیم ولی میزان عمق میدان ما بسیار کم میشود و فقط در عمق های محدودی تصویر واضح داریم در حالی که با کاهش اندازه دریچه به اندازه ای مانند $f/22$ میزان نور عبوری را خیلی کم میکنیم ولی میزان عمق میدان ما بسیار زیاد میشود و تا بازه خیلی خوبی همه تصاویر به عمق های متفاوت را شفاف و غیر تار میکنیم.

۳. فرضیات صورت سؤال به شرح زیر است:

$$f = 10 \text{ cm}, u+v = 40 \text{ cm}, v = 10 \text{ cm}$$

طبق بالا می توان برداشت کرد که فاصله جسم تا عدسی برابر با ۳۰ سانتی متر است. حال با داشتن مقادیر f و u مقدار مناسب را برای پرده پیدا می کنیم:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$$

$$\frac{1}{10} = \frac{1}{30} + \frac{1}{v} \Leftrightarrow v = \frac{3-1}{30} = 15 \text{ cm}$$

پس فاصله مناسب برای پرده یا فیلم ما ۱۵ سانتی متر است درحالی که الان برابر با ۱۰ سانتی متر است. در این شرایط، تصویر ما پشت پرده تشکیل می شود و تصویر درست شده روی پرده تار خواهد بود.

حال باید با فرض ثابت بودن فاصله دوربین با شی، برای ثبت تصویر با کیفی تر لنز دوربین تنها کافی است که پرده را ۵ سانتی متر به عقب ببریم تا تصویر ما واضح شود.

۷. عکس شماره ۵ کالیبره شده‌ای که با به‌دست‌آوردن پارامترهای کالیبراسیون از عکس‌های ۱ تا ۴ درست شده بود، شکل طبیعی‌تری نسبت به حالتی دارد که ما این پارامترهای کالیبراسیون را فقط از عکس ۱ برداشت کرده بودیم.

می‌توان دلیل این موضوع را هم در ۲ عنوان بیان کرد:

- برای کالیبره کردن به‌عکس، ما در حال حل یک معادله ماتریسی هستیم که جواب‌هایی که محاسبه می‌کنیم دقیق نیستند. حال هر چه تعداد معادلات ما برای حل این معادله ماتریسی بیشتر باشد، به جواب دقیق‌تری می‌رسیم. برای افزایش تعداد این معادلات نیز، نیاز به داشتن عکس‌های متعددی داریم. پس منطقاً در حالتی که عکس‌های ورودی ما ۱ تا ۴ بودند به جواب دقیق‌تری برای کالیبره عکس ۵ می‌رسیم.
- دیدگاه بعدی این است که هر صفحه شطرنجی ما پارامترهای کالیبراسیون که تولید می‌کند مربوط به محیط لوکال خودش است و منطقاً تضمینی وجود ندارد در یک محیط دیگر همین شرایط برای کالیبره کردند موردنیاز باشند. به طور مثال اگر صفحه شطرنجی ما در سمت چپ پایین باشند، به احتمال زیاد داری عمق متفاوتی است که نقاط ما در سمت راست بالا دارند و با داشتن پارامترهای کالیبراسیون برای سمت چپ پایین، قادر به برطرف کردن اعوجاج همه جهات نیستیم.

منابع:

- [OpenCV: Camera Calibration](#)
- [Camera Calibration with Python - OpenCV - GeeksforGeeks](#)