

بسمه تعالی



دانشکده مهندسی کامپیوتر

بینایی کامپیوتر

نام استاد: دکتر محمدی

تمرین دوم

نام دانشجو: آرمان حیدری

شماره دانشجویی: ۹۷۵۲۱۲۵۲

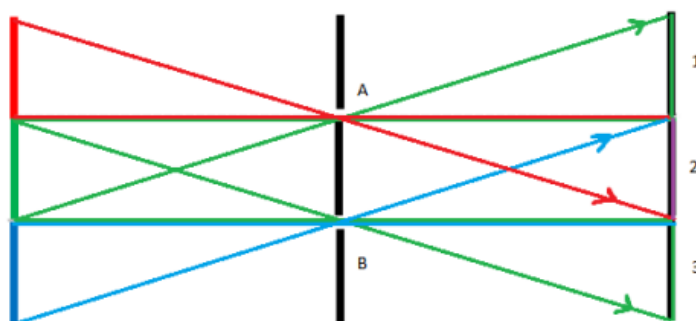
مهر ۱۴۰۱

فهرست

۳	پاسخ سوال اول
۳	منابع
۴	پاسخ سوال دوم
۴	منابع
۵	پاسخ سوال سوم
۵	منابع
۶	پاسخ سوال چهارم
۶	(۱)
۷	(۵)
۷	(۶)
۸	(۷)
۹	منابع

پاسخ سوال اول

کم شدن اندازه دریاچه‌ها منجر به کم شدن روشنایی می‌شود چون نور کمتری از دریاچه عبور می‌کند. همچنین اگر خیلی کوچک شوند، باعث پراکندگی نور پس از عبور شده و تصویر تار می‌شود. بزرگ شدن دریاچه‌ها باعث می‌شود روشنایی با عبور نور بیشتر، بیشتر شود. اما چون نور منعکس شده در بخش بیشتری تاثیر می‌گذارد، تصویر تار می‌شود.



شکل ۱: پراکندگی رنگ‌ها با دو دریاچه

با توجه به شکل ۱، ناحیه اول و سوم فقط حاوی رنگ سبز هستند و ناحیه ۲، حاوی رنگ‌های آبی و قرمز است.

منابع

- ویدئو جلسه دوم کلاس

پاسخ سوال دوم

مربوط به دوربین لنزدار است و مشکل ناشی از سه بعدی بودن جسم و دو بعدی بودن تصویر است که به آن مشکل عمق میدان DOF گفته می شود. بخش های بالایی و پایینی، تصویرشان پشت یا جلوی فیلم تشکیل شده است. و فقط بخش وسط تصویر با فاصله کانونی لنز منطبق است.

تصویر به صورت perspective بوده و v برای بالای تصویر زیاد و برای پایین آن کم است، پس طبق فرمول منطبق با f لنز نیستند و فقط وسط منطبق است.

بهترین تغییر در نوع تصویربرداری، استفاده از دریچه برای افزایش یا کاهش عمق میدان تصویر ماست. با استفاده از یک دریچه خیلی بزرگ، می توانیم به نور های بسیار زیادی اجازه ورود 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، 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 میبینیم. با یک trade-off مناسب، و انتخاب اندازه مناسب دریچه این مشکل حل می شود.

منابع

- ویدئو جلسه دوم کلاس

پاسخ سوال سوم

رابطه ای که میدانیم:

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

فرضیات مسئله:

$$f = 10 \text{ cm}, \quad u = 30 \text{ cm}, \quad v = 10 \text{ cm}$$

با جایگذاری u و f در این رابطه، میبینیم که $v=15 \text{ cm}$ به دست می آید. یعنی فاصله بهینه برای فیلم ۱۵ باید باشد که در این سوال ۱۰ است. پس تصویر روی فیلم تار خواهد بود و تصویر مناسب پشت فیلم تشکیل می شود.

حالا با فرض ثابت بودن فاصله فیلم و شی و ثابت بودن نوع دوربین، باید فاصله بهینه را پیدا کنیم. یعنی دو معادله دو مجهول $u+v=40 \text{ cm}$ و $0.1 = 1/u + 1/v$ را باید حل کنیم. که به دست می آید: $u=20 \text{ cm}$ و $v=20 \text{ cm}$. که یعنی باید لنز را ۱۰ سانتی متر به شی نزدیک تر کنیم.

منابع

- اسلاید جلسه دوم درس

پاسخ سوال چهارم

کد مربوط به این سوال در فایل Q4.py پیوست شده است.

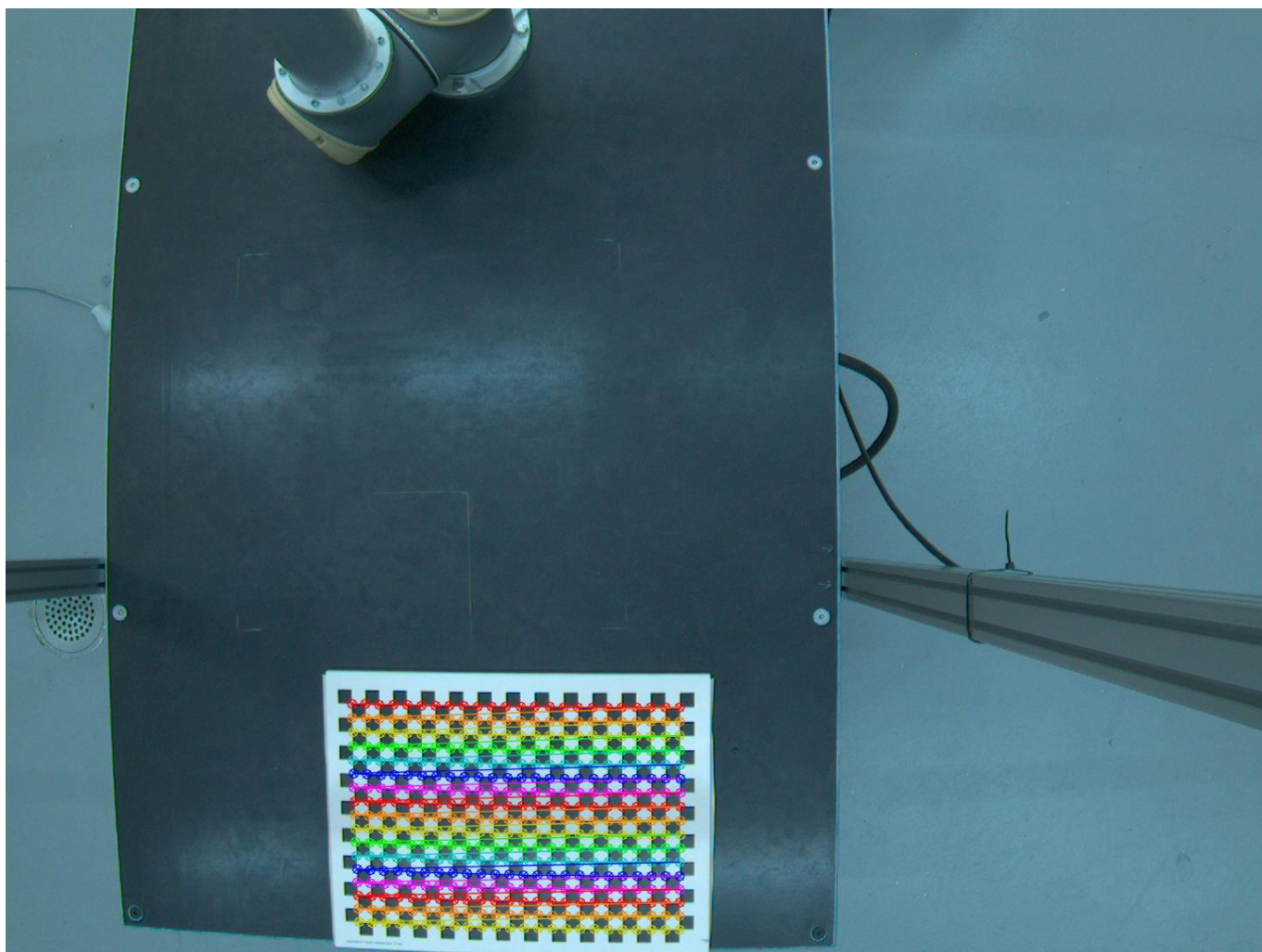
(۱)

دو نوع عمده تحریف عبارتند از: اعوجاج شعاعی و اعوجاج مماسی اعوجاج شعاعی باعث می شود خطوط مستقیم منحنی به نظر برسند. هر چه نقاط دورتر از مرکز تصویر باشند، اعوجاج شعاعی بزرگتر می شود به همین ترتیب، اعوجاج مماسی به این دلیل رخ می دهد که لنز تصویربرداری هم تراز نیست.

کاملاً موازی با صفحه تصویربرداری. بنابراین، برخی از مناطق در تصویر ممکن است نزدیکتر به نظر برسند. در عمل، هیچ لنز ساخته شده ای کامل نیست، تصاویر گرفته شده توسط دوربین ها سوژه هستند به تحریف این تحریفات بر اساس انتخاب به دو نوع اصلی تقسیم می شوند.

(۳)

نتیجه این بخش در فایل calib1_corners.png است.



(۵)

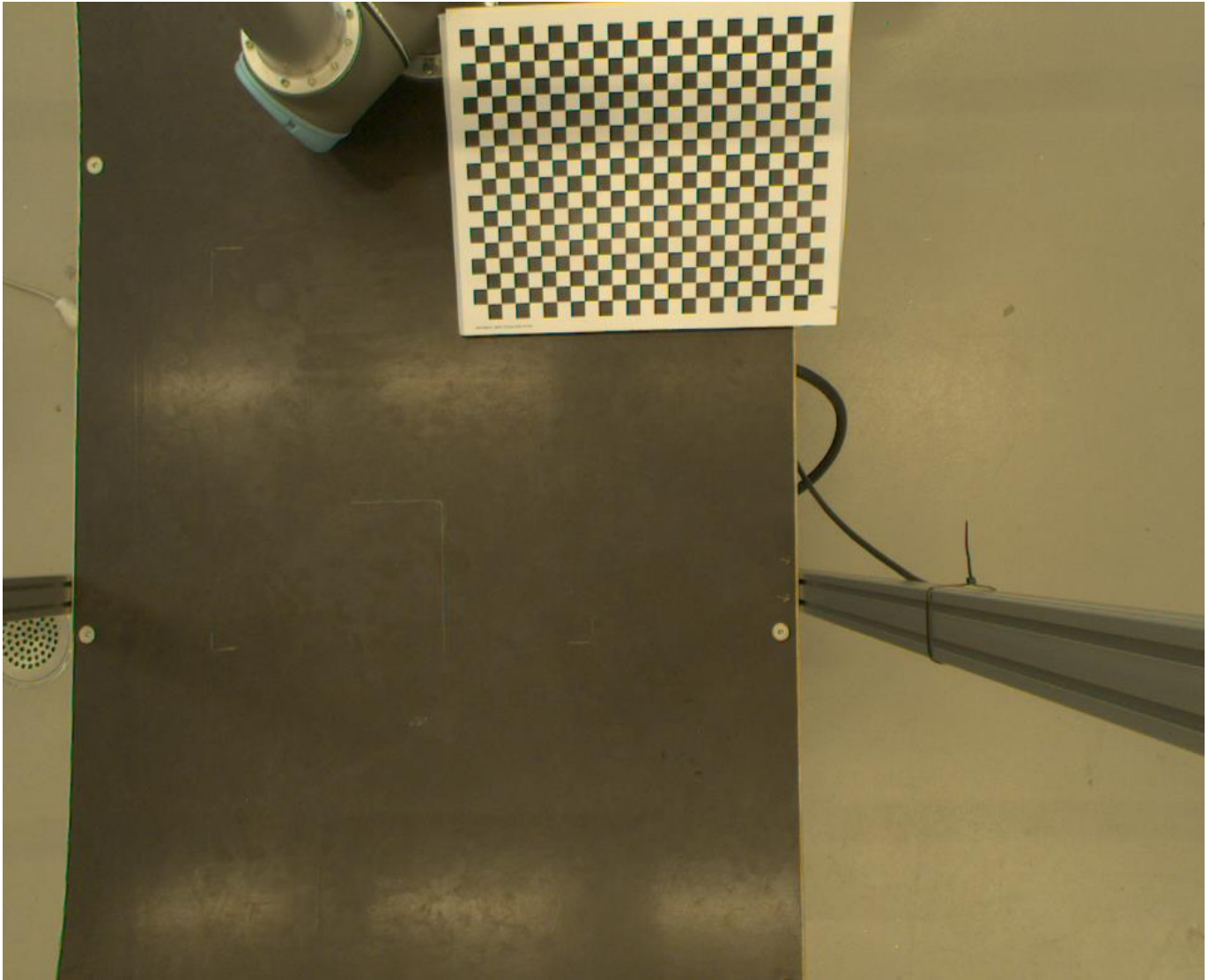
پارامترها به ترتیب از راست به چپ:

```
01 0 = {float64} -0.20625468220093468
01 1 = {float64} 0.19895320011233164
01 2 = {float64} -0.0008226067017545929
01 3 = {float64} -0.016491390046288943
01 4 = {float64} -0.18016533311994923
```

شکل ۲: پارامترهای کالیبریشن با *img1*

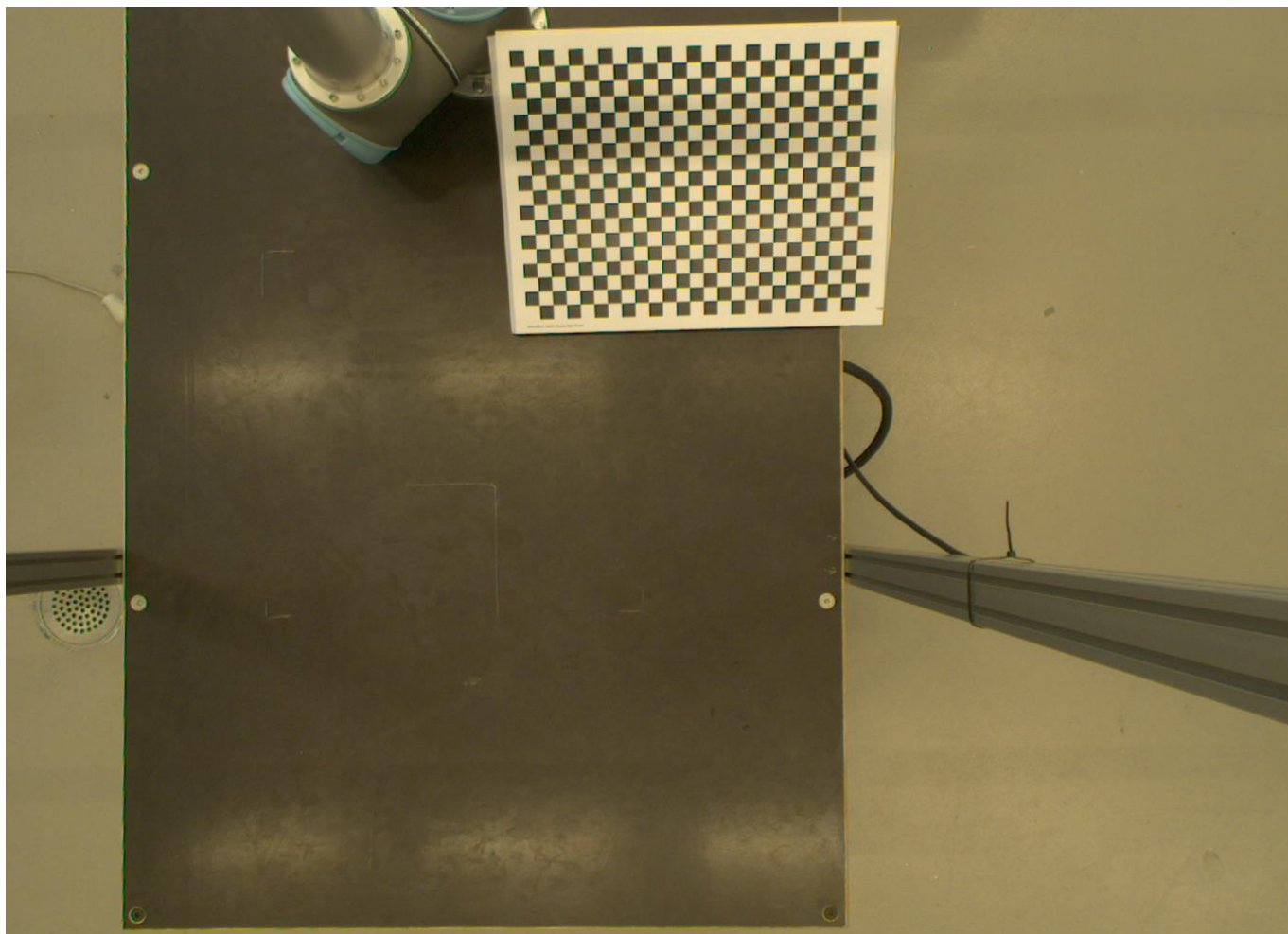
(۶)

خروجی این بخش در فایل *calib1.png* است.



(۷)

نتیجه این بخش در فایل `calib2.png` پیوست شده است.



میبینیم که خروجی این بخش کمی بهتر از قسمت قبل است چون پارامترها از چندین عکس این دوربین به دست آمده اند و مقادیر واقعی تری هستند. به دو دلیل این بهبود را شاهد هستیم.

۱. صفحات شطرنج این عکس ها در جاهای مختلفی هستند و ممکن است عمق دوربین در آن ها متفاوت باشد پس کمی پارامترهای کالیبره متفاوت به دست بیاید و در مجموع همه آن ها پارامترها واقعی تر خواهد بود و در قسمت های مختلف عکس `img5` بهتر کالیبراسیون انجام می شود.

۲. برای حل معادله چند مجهول پارامترها، با افزایش معادلات به پارامترهای دقیق تری می رسیم.

منابع

- https://opencv24-python-tutorials.readthedocs.io/en/stable/py_tutorials/py_calib3d/py_calibration/py_calibration.html