



دانشکده مهندسی کامپیوتر

تمرین سری ۷ بینایی کامپیوتر

نام درس

مبانی بینایی کامپیوتر

نام دانشجو

زهرا انوریان

نام استاد درس

دکتر محمدی

پاییز ۱۳۹۹

سوالات

۱. هر کدام از رنگ‌های زیر را از فضای رنگی مبدا به فضای رنگی مقصد ببرید. (محدوده رنگی فضاهای مبدا و مقصد ۰ تا ۲۵۵ فرض شود) (حل با راه حل و به صورت کامل باشد و جواب آخر به تنهایی نمره‌ای ندارد) (۲۰ نمره)

a. از RGB به CMYK $k = 255 - \max(R, G, B) = 255 - \max(0, 255, 100) = 0$

$$\begin{bmatrix} C \\ M \\ Y \\ K \end{bmatrix} = \begin{bmatrix} 255 - k \\ 255 - k \\ 255 - k \\ 255 - k \end{bmatrix} - \begin{bmatrix} 0 \\ 255 \\ 100 \\ 0 \end{bmatrix} = \begin{bmatrix} 255 \\ 0 \\ 155 \\ 0 \end{bmatrix}$$

b. از CMY به RGB $\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix} \rightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} 80 \\ 43 \\ 100 \end{bmatrix} = \begin{bmatrix} 175 \\ 212 \\ 155 \end{bmatrix}$

c. از CMYK به RGB $\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 115 \\ 87 \\ 0 \\ 155 \end{bmatrix}$

$$R = 255 * \left(1 - \frac{C}{255}\right) * \left(1 - \frac{k}{255}\right) = 255 * \left(1 - \frac{115}{255}\right) * \left(1 - \frac{155}{255}\right) = 255 * 0.54 * 0.39 \cong 54$$

$$G = 255 * \left(1 - \frac{M}{255}\right) * \left(1 - \frac{k}{255}\right) = 255 * \left(1 - \frac{87}{255}\right) * \left(1 - \frac{155}{255}\right) = 255 * 0.65 * 0.39 \cong 65$$

$$B = 255 * \left(1 - \frac{Y}{255}\right) * \left(1 - \frac{k}{255}\right) = 255 * (1 - 0) * \left(1 - \frac{155}{255}\right) = 255 * 1 * 0.39 \cong 99$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 54 \\ 65 \\ 99 \end{bmatrix}$$

۲. همبستگی یکی از راه‌های اندازه‌گیری شباهت و جابجایی میان دو تصویر است. یکی از روش‌های محاسبه همبستگی normalized cross-correlation است. در این باره تحقیق کرده و نحوه کارکرد این روش را توضیح دهید (۱۵ نمره).

پاسخ: از Normalized Cross-Correlation (NCC) معمولاً به عنوان معیاری برای ارزیابی میزان شباهت دو تصویر مقایسه شده، استفاده می‌شود. مزیت اصلی‌ای که NCC نسبت به Cross Correlation معمولی دارد، این است که نسبت به تغییرات خطی دامنه‌ی روشنایی دو تصویر حساسیت کمتری دارد و همچنین تنظیم مقدار آستانه‌ی تشخیص آن بسیار ساده‌تر از CC (ϕ_{xy}) است. علاوه بر این مقدار NCC ($\phi'_{xy}(t)$) در محدوده‌ی ۱ تا -۱ قرار دارد یعنی اگر $\phi'_{xy}(t)$ برابر ۱ شود، به این معنی است که در t این دو سری زمانی کاملاً یک شکل هستند (اما ممکن است دامنه‌ها متفاوت باشد) و اگر $\phi'_{xy}(t)$ برابر -۱ شود، به این معنی است که یک شکل‌اند

اما در جهت عکس یکدیگرند و اگر $\varphi'_{xy}(t)$ برابر صفر شود، نشان دهنده‌ی کاملاً متفاوت بودن است و وقتی روی سیگنال‌های گسسته‌ی واقعی NCC اعمال می‌شود، دیده شده که مقدار بیشتر از ۰.۷ و ۰.۸ نشان‌دهنده‌ی بهترین مطابقت است.

$$\varphi'_{xy}(t) = \frac{\varphi_{xy}}{\sqrt{\varphi_{xx}(0)\varphi_{yy}(0)}}$$

لینک:

[https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.675.1379&rep=rep1&type=pdf#:~:text=Normalized%20cross%20correlation%20\(NCC\)%20has,dissimilarity\)%20between%20two%20compared%20images.&text=The%20setting%20of%20detection%20threshold,simpler%20than%20the%20cross%20correlation.](https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.675.1379&rep=rep1&type=pdf#:~:text=Normalized%20cross%20correlation%20(NCC)%20has,dissimilarity)%20between%20two%20compared%20images.&text=The%20setting%20of%20detection%20threshold,simpler%20than%20the%20cross%20correlation.)

۳. درباره hessian detector تحقیق کرده و نحوه عملکرد آن را شرح دهید. سپس آن را با harris detector مقایسه کنید (تفاوت‌های پیاده‌سازی و خروجی را بیان کنید). (۱۵ نمره)

پاسخ: هر دو از الگوریتم‌های تشخیص ویژگی هستند که در حوزه‌ی بینایی ماشین و آنالیز تصویر کاربرد دارند. الگوریتم harris detector براساس نقاط کلیدی تشخیص داده شده‌ی تصویر در مقیاس‌های متفاوت با استفاده از اندازه‌گیری گوشه‌ی harris کار می‌کند. الگوریتم hessian detector نیز مانند harris با یک الگوریتم تکرار شونده^۱، برای نقاط کلیدی مقیاس‌های متفاوت را می‌سنجد و مقایسه می‌کند تا مقیاس مناسب برای نقاط را بدست آورد اما با استفاده از ماتریس hessian که در زیر آمده است که $L_{xx}(x)$ مشتق دوم در راستای x است و $L_{xy}(x)$ مشتق دوم در راستای x,y می‌باشد (که در الگوریتم harris مشتق اول محاسبه می‌شد). لازم به ذکر است که مشتقات در تکرار فعلی^۲ محاسبه می‌شوند و بنابراین آن‌ها مشتقات یک تصویر هموار شده توسط کرنل گوسی هستند (به این صورت که کرنل گوسی با سائزهای متفاوت در تصویر اصلی کانوالو شده است). فرمول:

$$L(x) = g(\sigma_l) * I(x)$$

مشتقات باید با یک فاکتوری مربوط به کرنل گوسی (σ_l^2) مقیاس^۳ شوند. در هر مقیاس، نقاط کلیدی نقاطی هستند که به طور همزمان دو مقدار دترمینال و تریس را ماکزیمم می‌کنند که این دو مقدار با استفاده از فرمول‌های زیر بدست می‌آید که تریس ماتریس hessian برابر با لاپلاسیان گوسی (LoG) می‌باشد.

$$H(x) = \begin{bmatrix} L_{xx}(x) & L_{xy}(x) \\ L_{yx}(x) & L_{yy}(x) \end{bmatrix}$$

¹ Iterative

² Current iteration

³ Scale

$$DET = \sigma^2_I (L_{xx}L_{yy}(x) - L_{xy}^2(x))$$

$$Trace = \sigma_I(L_{xx} + L_{yy})$$

به طور کلی، الگوریتم hessian مانند harris نقاط کلیدی زیاد و کوچکتري را نسبت به ديگر الگوريتم‌ها تشخيص مي‌دهد. براي يك تصوير، الگوريتم hessian معمولا از الگوريتم harris نواحی قابل اطمینان‌تری را شناسایی می‌کند اما عملکرد با توجه به نوع صحنه‌ی مورد نظر، متفاوت است. الگوریتم hessian در صحنه‌های texture که بیشتر دارای قسمت‌های شبیه گوشه است، عملکرد خوبی دارد. با این حال این الگوریتم برای برخی از صحنه‌های ساختارمند مانند ساختمان‌ها عملکرد بسیار خوبی دارد.

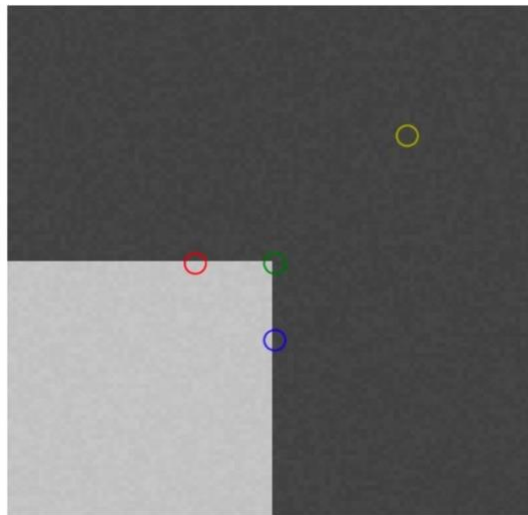
لینک:

https://en.wikipedia.org/wiki/Hessian_affine_region_detector

۴. مقادیر ویژه ماتریس‌های زیر را به دست آورید (در صورت نیاز می‌توانید از `scipy.linalg.eig` استفاده کنید).
با توجه به مقادیر ویژه به دست آمده، هر ماتریس متعلق به کدام قسمت مشخص شده در تصویر زیر است؟ (۳۰ نمره)

$$M_1 = \begin{bmatrix} 84.33 & -16.97 \\ -16.97 & 59.48 \end{bmatrix} \quad M_2 = \begin{bmatrix} 163.54 & -0.217 \\ -0.217 & 0.1053 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0.1714 & -0.496 \\ -0.496 & 164.4 \end{bmatrix} \quad M_4 = \begin{bmatrix} 0.1439 & -0.009 \\ -0.009 & 0.323 \end{bmatrix}$$



پاسخ: با توجه به مقادیر ویژه بدست آمده که در تصویر زیر مشاهده می‌کنید، ماتریس $M1$ متعلق به دایره‌ی سبز است زیرا دارای بیشترین مقادیر ویژه است پس نشان دهنده‌ی گوشه در تصویر می‌باشد. ماتریس $M2$ متعلق به دایره‌ی آبی است زیرا دارای یک مقدار ویژه بزرگ است پس نشان‌دهنده‌ی وجود لبه در تصویر است و همچنین باتوجه به ماتریس مقدار مشتق در راستای X بزرگتر است پس لبه عمودی است. ماتریس $M3$ متعلق به دایره‌ی قرمز است زیرا دارای یک مقدار ویژه بزرگ است پس نشان‌دهنده‌ی وجود لبه در تصویر است و همچنین باتوجه به ماتریس مقدار مشتق در راستای Y بزرگتر است پس لبه افقی است. ماتریس $M4$ نیز متعلق به دایره‌ی زرد است زیرا مقادیر ویژه آن کوچک است و نشان دهنده‌ی تصویری صاف است.

```
import numpy as np
import scipy.linalg as la

M1 = np.array([[84.33,-16.97],
               [-16.97,59.48]])

result1 = la.eig(M1)
print("result1: ",result1[0])

M2 = np.array([[163.54,-0.217],
               [-0.217,0.1053]])

result2 = la.eig(M2)
print("result2: ",result2[0])

M3 = np.array([[0.1714,-0.496],
               [-0.496,164.4]])

result3 = la.eig(M3)
print("result3: ",result3[0])

M4 = np.array([[0.1439,-0.009],
               [-0.009,0.323]])

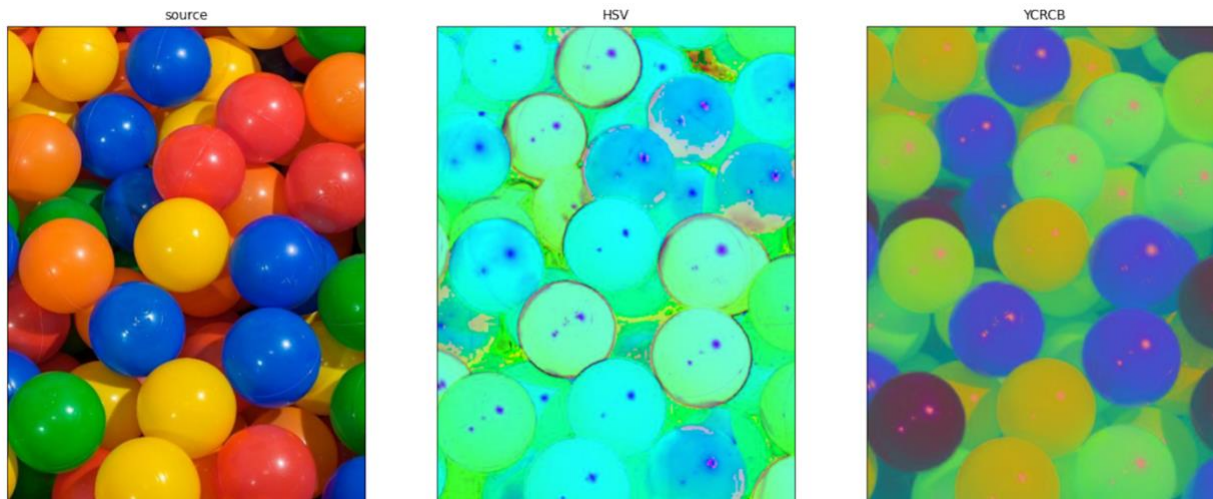
result4 = la.eig(M4)
print("result4: ",result4[0])

result1: [92.93739228+0.j 50.87260772+0.j]
result2: [1.63540288e+02+0.j 1.05011879e-01+0.j]
result3: [ 0.169902+0.j 164.401498+0.j]
result4: [0.14344888+0.j 0.32345112+0.j]
```

۵. تبدیل فضای رنگی

الف) با استفاده از توابع OpenCV عکس 5.jpg را به ۲ فضای رنگی YCrCb و HSV ببرید (۱۰ نمره).

پاسخ: با استفاده از تابع cv2.cvtColor() و دو متغیر COLOR_BRG2YCr_Cb و COLOR_BRG2HSV تصویر زیر را به فضاهای HSV و YCrCb برده‌م.



پ) با استفاده از دانش خود تفاوت دو عکس زیر (b1.jpg5 و b2.jpg5) را بیابید (۱۵ نمره).

پاسخ: ابتدا دو تصویر را خاکستری می‌کنیم و سپس یکی از آن‌ها را به عنوان کانال قرمز (R) و دیگری را به عنوان کانال‌های سبز (G) و آبی (B) تصویر جدید در نظر می‌گیریم و سپس عکس تولید شده تفاوت‌های این دو عکس را به رنگ‌های فیروزه‌ای (که حاصل ترکیب سبز و آبی است) و قرمز نشان می‌دهد.



۶. تابعی پیاده‌سازی کنید که به کمک آن بتوانید دایره‌های قرمز موجود در یک تصویر را به رنگ سفید درآورده و به کمک آن دایره‌های قرمز موجود در ویدئو 4balls.mp را به رنگ سفید تبدیل کنید. (۳۰ نمره)

پاسخ: برای تشخیص توپ‌های قرمز لازم است که ابتدا دو آستانه بالا و پایین برای رنگ قرمز بدست آوریم و سپس هر دو رنج آستانه را با یکدیگر جمع کنیم. حال ابتدا توپ‌های قرمز را با کمک تابع [cv2.bitwise_not\(\)](#) و مقدار آستانه‌ی بدست آمده، از فریم‌ها حذف می‌کنیم. حال برای آنکه توپ‌های قرمز به رنگ سفید تغییر کنند لازم است که بگوییم هر جایی از فریم که دارای این آستانه‌ی بدست آمده برای رنگ قرمز است را ۲۵۵ (سفید) کند و حال با کمک تابع [cv2.bitwise_and\(\)](#) فریمی بدست می‌آید که فقط دارای توپ‌های قرمز است که به رنگ سفید تغییر کرده‌اند. حال برای آنکه دیگر توپ‌ها (یعنی توپ‌های سبز و آبی) را نیز داشته باشیم می‌توان فریم بدست آمده را با فریمی که تنها شامل توپ‌های سبز و آبی است و توپ‌های قرمز را از آن حذف کردیم، جمع کرد تا نتیجه‌ی دلخواه بدست آید.

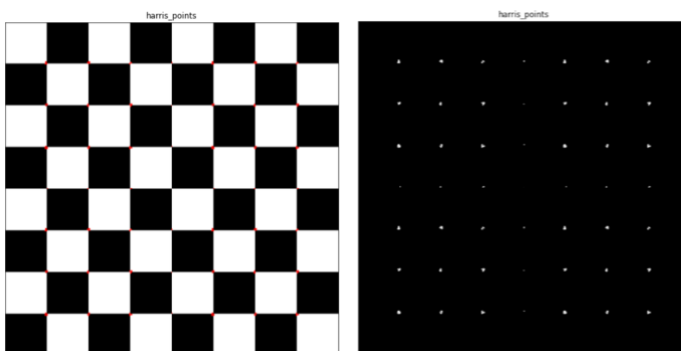
۷. تابع `harris_points` را کامل کنید. در این تابع شما الگوریتم `harris corner detector` را پیاده‌سازی می‌کنید که ورودی این تابع تصویر مدنظر است. نقاط کلیدی تصویر `7.jpg` را با این تابع به دست آورید. خروجی این تابع، تصویری هست که در آن نقاط کلیدی مشخص هستند. تصویر زیر یک نمونه خروجی است (با اعمال آستانه بهتر نتیجه بهتری را نیاز میتوان به دست آورد). از توابع آماده `harris detection` استفاده نکنید (۳۰ نمره).

پاسخ: برای پیاده‌سازی الگوریتم `harris corner detector` ابتدا باید تصویر را خاکستری کنیم و به کمک فیلتر Sobel مقادیر مشتق در هر دو راستا (I_x و I_y) را بدست آوریم و سپس با استفاده از فیلتر گوسی نیز مقادیر I_{xx} و I_{yy} و I_{xy} را بدست می‌آوریم و سپس دترمینال M (تفاضل ضرب دو قطر ماتریس در هم) را بدست آورده و M (مجموع قطر اصلی) را نیز بدست آورده و سپس با استفاده از فرمول زیر R را بدست می‌آوریم و در آخر مقادیر بیشینه را حذف می‌کنیم زیرا نقاطی که توسط R بدست آمده است مجموع چندین پیکسل است و ما باید در واقع نقاط ماکزیمم محلی را پیدا کنیم و آن‌ها را به رنگ قرمز درآوریم و روی عکس نمایش دهیم.

$$trace(M) = I_{xx}^2 + I_{yy}^2$$

$$R = det(M) - k * (trace(M))^2$$

نتیجه R:



نتیجه نهایی:

