



دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی برق

بینایی ماشین (Machine Vision)

Assignment 4

نگارش

محمد مهدی نجفی زاده

سارا سالمی

علی دانشپور

استاد درس

دکتر زهرا سادات شریعتمداری مرتضوی

آذر ۱۴۰۳

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تمرین‌های برنامه‌نویسی

۱ - ساخت تصاویر پانوراما یکی از کاربردهای انطباق تصویر است. برای این کار از محیط موردنظر چندین تصویر گرفته می‌شود و سپس با توجه به روش‌های انطباق تصویر، این تصاویر به یکدیگر متصل می‌شوند. با استفاده از توابع `opencv` برنامه‌ای بنویسید که دو تصویر `image1` و `image2` در پیوست را به یکدیگر متصل نماید.

در این سوال به دنبال اتصال دو تصویر `image 1` و `image 2` و در نهایت ساخت یک تصویر پانوراما می‌باشیم. برای این منظور از کتابخانه `OpenCV` و تابع `cv2.Stitcher_create` استفاده کردیم. خروجی حاصل از اتصال دو تصویر `image 1` و `image 2` در شکل ۱ قابل مشاهده است.

Panorama Result



شکل ۱: خروجی حاصل از اتصال دو تصویر

۲ - در دو تصویر image3 و image4 ابتدا نقاط کلیدی را به کمک detector Harris استخراج نمایید. سپس با استفاده از همسایگی نقاط کلیدی، یک توصیفگر برای این نقاط استخراج کنید. در مرحله بعد، برای انطباق توصیفگرها از معیار correlation-cross normalized استفاده کنید. در نهایت، نقاط کلیدی متناظر را در دو تصویر با یک خط به هم متصل نمایید.

در این سوال به دنبال اتصال نقاط کلیدی دو تصویر image 3 و image 4 به یکدیگر می‌باشیم. ابتدا تصاویر image 3 و image 4 را به تصاویر سیاه و سفید تبدیل کرده و سپس با کمک ماژول cv2.cornerHarris اقدام به شناسایی نقاط گوشه می‌نمائیم. سپس با استفاده از ماژول cv2.dilate با عملیات گسترش^۱ اقدام به بهبود و مشخص‌تر کردن گوشه‌های شناسایی شده نمودیم. در مرحله بعد با در نظر گرفتن یک مقدار آستانه، تنها گوشه‌هایی که مقادیر آن‌ها بالاتر از این حد آستانه است را نگه می‌داریم.

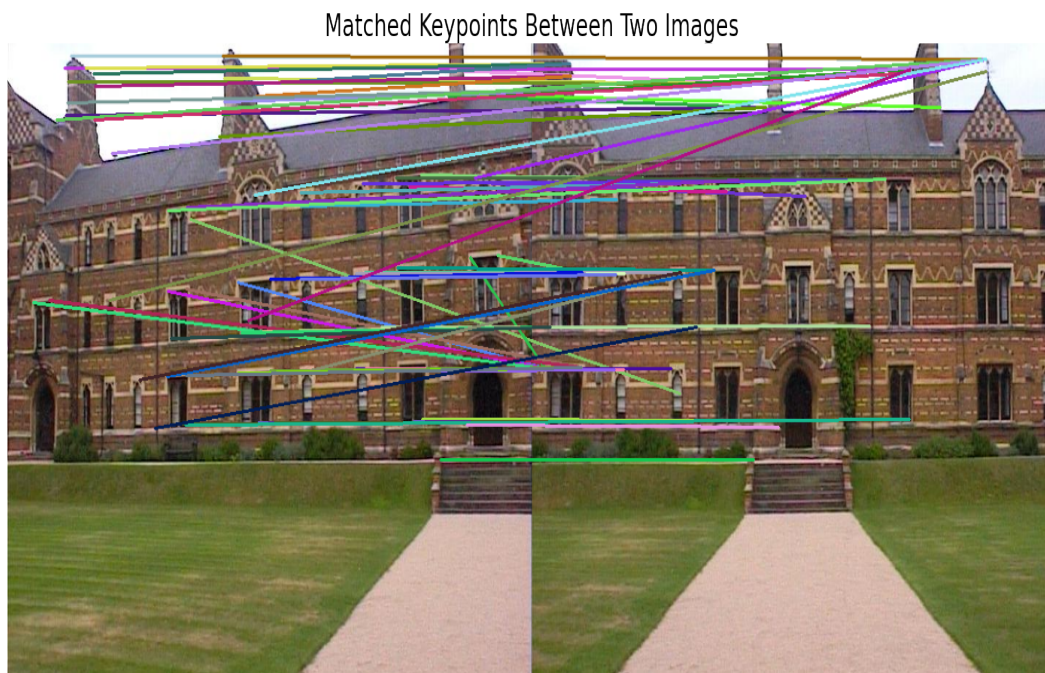
برای اطمینان از اعتبار نقاط کلیدی تشخیص داده شده، از تابع validate_keypoints استفاده می‌کنیم. این تابع بررسی می‌کند که آیا نقطه کلیدی تشخیص داده شده در محدوده معتبر تصویر قرار دارد یا خیر. علاوه بر این، با حذف نقاط کلیدی که در شعاع آستانه مشخصی قرار دارند، از تجمع بیش از حد نقاط کلیدی جلوگیری می‌کند. تابع apply_double_filtering این فرآیند اعتبارسنجی را روی نقاط کلیدی تشخیص داده شده در هر دو تصویر اعمال می‌کند تا تنها نقاط کلیدی با کیفیت بالا و فاصله مناسب برای تطبیق باقی بمانند.

سپس نقاط کلیدی بین دو تصویر با استفاده از تابع match_keypoints تطبیق داده می‌شوند. این تابع قطعات کوچکی از پیکسل‌ها که به آنها توصیفگر (Descriptor) گفته می‌شود، در اطراف هر نقطه کلیدی در تصویر اول استخراج کرده و آنها را با توصیفگرهای اطراف نقاط کلیدی در تصویر دوم مقایسه می‌کند. این مقایسه با استفاده از تطبیق الگو و با کمک گرفتن از ماژول cv2.matchTemplate انجام می‌شود و بهترین تطابق برای هر نقطه کلیدی شناسایی می‌شود. این روش شباهت بین توصیفگرها را محاسبه کرده و محتمل‌ترین نقاط متناظر را انتخاب می‌کند.

برای نمایش نقاط کلیدی تطبیق داده شده، از تابع connect_keypoints استفاده می‌کنیم. این تابع خطوطی بین نقاط متناظر در دو تصویر رسم می‌کند. دو تصویر ورودی در کنار یکدیگر قرار گرفته و نقاط کلیدی تطبیق داده شده با استفاده از خطوط با رنگ‌های تصادفی به یکدیگر متصل می‌شوند. این مرحله نمایش واضحی از فرآیند تطبیق ارائه می‌دهد و به ارزیابی دقت آن کمک می‌کند.

در نهایت، تابع find_matching_keypoints کل فرآیندهای توصیف شده در بالا را با یکدیگر ادغام کرده و به ترتیب اجرا می‌کند و در نهایت نتایج را نمایش می‌دهد. این تابع دو تصویر ورودی را دریافت کرده و یک تصویر ترکیبی با نقاط کلیدی تطبیق داده شده به عنوان خروجی تولید می‌کند. خروجی به دست آمده در شکل ۲ قابل مشاهده می‌باشد.

¹Dilate



شکل ۲: تصویر حاصل از اتصال نقاط کلیدی دو تصویر

۳ - در تصاویر logo_1 تا logo_4 موجود در پیوست، با استفاده از عملگرهای مورفولوژی، لوگوهای موجود در تصاویر را دیتکت نمایید. برنامه نوشته شده باید قابلیت تشخیص لوگوی موجود در هر چهار تصویر را داشته باشد.

در این تمرین به دنبال تشخیص لوگوهای موجود در ۴ تصویر می‌باشیم. به طور خلاصه کد پیاده‌سازی شده شامل مراحل پیش‌پردازش، عملیات مورفولوژیکی، تحلیل ویژگی‌ها و ... می‌باشد. کد پیاده‌سازی شده با تابع `preprocess_image` آغاز می‌شود، که در آن تصویر ورودی به فرمت خاکستری تبدیل می‌شود تا برای مراحل بعدی آماده شود. آستانه‌گذاری تطبیقی روی تصویر خاکستری اعمال می‌شود تا یک تصویر باینری ایجاد شود که نواحی مورد نظر را با تنظیم آستانه محلی بر اساس شدت پیکسل مشخص کند. علاوه بر این، از تشخیص لبه با استفاده از الگوریتم Canny برای برجسته‌سازی مرزها استفاده می‌شود که به شناسایی نواحی احتمالی لوگو کمک می‌کند.

پس از پیش‌پردازش، از تابع `apply_morphological_operations` به منظور پاکسازی تصویر باینری شده استفاده می‌کنیم. در این مرحله عناصر ساختاری مانند کرنل‌های مستطیلی و بیضی ایجاد می‌شوند که در عملیات مورفولوژیکی مانند بسته شدن، باز شدن و گسترش^۱ استفاده می‌شوند. عملیات بسته شدن برای پر کردن حفره‌های کوچک در داخل نواحی و عملیات باز شدن برای حذف نویز انجام می‌شود.

^۱Dilation

همچنین عملیات گسترش نیز برای تقویت و گسترش نواحی شناسایی شده اعمال می‌شود تا کیفیت نواحی کاندید برای تحلیل بیشتر بهبود یابد.

سپس تصاویر دودویی پاکسازی شده به تابع `detect_potential_logos` منتقل می‌شوند، که نواحی کاندید لوگو را با شناسایی کانتورهای موجود در تصویر شناسایی می‌کند. هر کانتور شناسایی شده بر اساس معیارهایی مانند مساحت، نسبت ابعاد و ... ارزیابی می‌شود. این فیلترها تضمین می‌کنند که فقط نواحی‌ای که شباهت به لوگو دارند حفظ شوند.

در مرحله بعد به تحلیل ویژگی‌های هر ناحیه کاندید با استفاده از تابع `analyze_region_features` می‌پردازیم. این تابع معیارهای کلیدی مانند تراکم لبه، واریانس رنگ و کنتراست را برای هر ناحیه محاسبه می‌کند. تراکم لبه با استفاده از عملگر Sobel برای اندازه‌گیری شدت میانگین لبه‌ها در ناحیه تعیین می‌شود. واریانس رنگ، تغییرات پیکسل‌های RGB را کمی‌سازی می‌کند و غنای رنگی را نشان می‌دهد. کنتراست، تفاوت بین حداکثر و حداقل مقدار در هر ناحیه را اندازه‌گیری می‌کند و تفاوت‌های واضح را برجسته می‌کند.

برای شناسایی بهترین کاندید لوگو، از تابع `select_best_logo` استفاده می‌کنیم. این تابع هر ناحیه کاندید را بر اساس یک سیستم امتیازدهی وزنی ارزیابی می‌کند. این امتیاز ترکیبی از تراکم لبه، واریانس رنگ و کنتراست است که به هر ویژگی وزن خاصی اختصاص می‌دهد. ناحیه‌ای که بالاترین امتیاز را کسب کند، به‌عنوان محتمل‌ترین لوگو انتخاب می‌شود. اگر هیچ ناحیه‌ای معیارها را برآورده نکند، تابع مقدار `None` باز می‌گرداند که نشان‌دهنده عدم شناسایی لوگوی مناسب است.

در نهایت، تابع `extract_logo` تمام مراحل فوق را برای پردازش ۴ تصویر ورودی ادغام می‌کند. این تابع تصاویر را می‌خواند، پیش‌پردازش می‌کند، عملیات پاکسازی مورفولوژیکی را اعمال می‌کند، نواحی کاندید لوگو را شناسایی کرده و بهترین لوگو را انتخاب می‌کند. همچنین تصویر اصلی و لوگوی استخراج شده را برای هر تصویر در کنار هم نشان می‌دهد.

لگوهای شناسایی شده برای هر ۴ تصویر (در کنار تصویر اصلی) به ترتیب در اشکال ۳، ۴، ۵ و ۶ قایل مشاهده می‌باشد.



شکل ۳: لگوی استخراج شده از تصویر `logo_1`



شکل ۴: لگوی استخراج شده از تصویر logo_2



شکل ۵: لگوی استخراج شده از تصویر logo_3



شکل ۶: لگوی استخراج شده از تصویر logo_4

تمرین‌های تشریحی

۱ - یکی از آشکارسازهای ویژگی پرکاربرد در فیلد بینایی ماشین، **hessian detector** می‌باشد. نحوه عملکرد این آشکارساز را تشریح نمایید.

hessian detector یک آشکارساز ویژگی پرکاربرد در بینایی ماشین برای شناسایی نقاط یا مناطق مورد نظر در یک تصویر است. **hessian detector** مبتنی بر تجزیه و تحلیل مشتقات مرتبه دوم شدت روشنایی پیکسل‌های تصویر است. در ادامه جزئیات این روش و نحوه عملکرد آن را به تفصیل شرح می‌دهیم: ابتدا به معرفی ماتریس **hessian** می‌پردازیم. مشتقات مرتبه دوم هر پیکسل از تصویر در ماتریسی تحت عنوان ماتریس **hessian** ذخیره می‌شود. این ماتریس تغییرات شدت روشنایی اطراف هر پیکسل از تصویر را مشخص می‌کند. مقادیر این ماتریس با توجه به رابطه ۱ محاسبه می‌شود.

$$H(x, y) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{bmatrix} \quad (1)$$

• I_{xx} : تغییرات شدت روشنایی در جهت افقی

• I_{yy} : تغییرات شدت روشنایی در جهت عمودی

• I_{xx} : تغییرات شدت روشنایی به صورت مورب

hessian detector نقاط مورد نظر را با توجه به دترمینان و اثر^۱ ماتریس **Hessian** شناسایی می‌کند. **دترمینان ماتریس**: مقدار بزرگ برای دترمینان به وجود یک گوشه یا ویژگی‌های مشابه دلالت دارد، زیرا بر شیب‌های شدید در جهات مختلف دلالت دارد. مقدار این دترمینان با توجه رابطه ۲ محاسبه می‌شود.

$$\det(H) = I_{xx}I_{yy} - (I_{xy})^2 \quad (2)$$

اثر ماتریس: این پارامتر به بررسی تغییرات کلی شدت روشنایی هر پیکسل از تصویر می‌پردازد. البته به اندازه دترمینان قدرت تشخیص تفاوت بین گوشه‌ها و لبه‌ها را ندارد. مقدار اثر ماتریس با توجه رابطه ۳ محاسبه می‌شود.

$$\text{trace}(H) = I_{xx} + I_{yy} \quad (3)$$

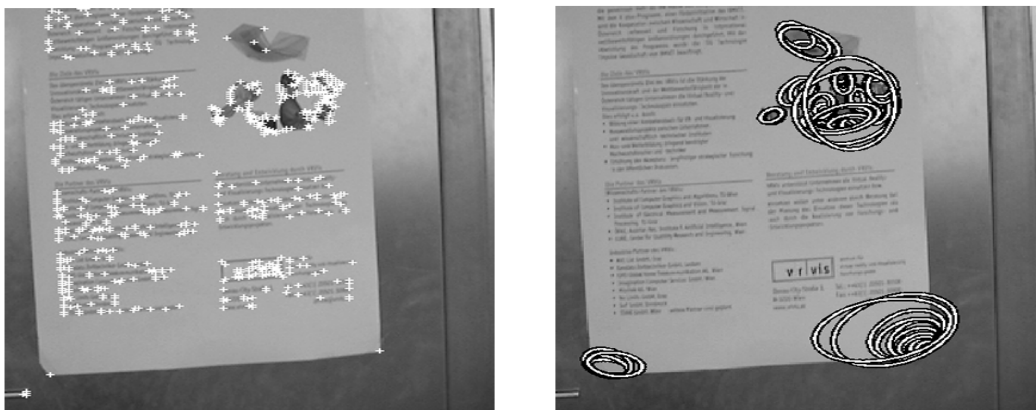
^۱Trace

به طور ساده و خلاصه آشکارساز Hessian به هر پیکسل از تصویر نگاه می کند و با استفاده از مشتقات مرتبه دوم بررسی می کند که شدت روشنایی اطراف هر پیکسل از تصویر به چه صورت تغییر می کند. سپس پیکسل هایی که تغییرات شدید در جهت های مختلف دارند را به عنوان «نقاط کلیدی» شناسایی می کند.

آشکارساز hessian اغلب در تشخیص گوشه و دایره های کوچک (مانند حباب) استفاده می شود و به عنوان پایه و اساس آشکارسازهای ویژگی های پیشرفته مانند آشکارساز SURF^۱ می باشد. اگر بخواهیم به مقایسه آشکارساز hessian و آشکارساز harris بپردازیم، ذکر این نکته ضروری است که آشکارساز harris از ماتریس مومنت دوم^۲ استفاده می کند در حالی که آشکارساز hessian از ماتریس مشتقات مرتبه دوم (رابطه ۱) که پیشتر اشاره کردیم، استفاده می کند. second-moment matrix با توجه به رابطه ۴ محاسبه می شود. harris detector بر اساس اندازه گیری تغییرات شدت تصویر در اطراف یک پیکسل در یک پنجره محلی عمل می کند و با محاسبه گرادیان تصویر، گوشه ها را با ارزیابی مقادیر ویژه با توجه به ماتریس زیر شناسایی می کند

$$H(x, y) = \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_{yy}^2 \end{bmatrix} \quad (۴)$$

به طور کلی، harris detector برای شناسایی گوشه ها مناسب تر بوده در حالی که hessian detector برای شناسایی ویژگی هایی به شکل حباب یا لبه هایی که انحنا دارند، مناسب تر است. یک نمونه از تفاوت عملکرد دو آشکارساز harris detector و hessian detector در شکل ۷ قابل مشاهده می باشد. شکل a مربوط به harris detector و شکل b مربوط به hessian detector می باشد



(a)

(b)

شکل ۷: تفاوت عملکرد دو آشکارساز hessian detector و harris detector [۱]
(a :harris detector b :hessian detector)

^۱Speeded-Up Robust Features

^۲second-moment matrix

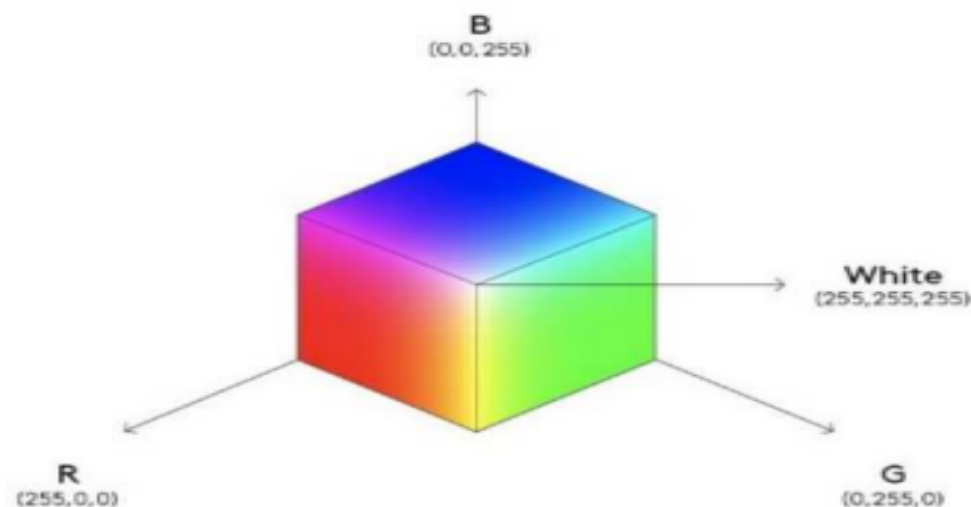
۲ - LAB color space یکی از فضا‌های رنگی پر کاربرد در تحلیل تصاویر رنگی می‌باشد. کانال‌های مختلف این فضا را معرفی نمایید. چگونه می‌توان یک تصویر را از فضای RGB به فضای LAB تبدیل نمود؟

اهداف تمرین:

هدف این تمرین آشنایی با فضای رنگ LAB و کاربردهای آن در تحلیل تصاویر رنگی است. همچنین نقش کانال‌های مختلف در پردازش تصویر و نحوه تبدیل فضای LAB به RGB را بررسی می‌کند. این تمرین به درک عمیق‌تر مفاهیم مرتبط با فضا‌های رنگی و کاربرد آن‌ها در نمایش و پردازش تصاویر کمک می‌کند. همچنین با استفاده‌های عملی این تبدیل‌ها در بهبود نمایش رنگ‌ها و کاهش وابستگی به دستگاه‌ها آشنا می‌شویم. در نهایت، این تمرین ارتباط تئوری و عملی در مسائل مرتبط با بینایی ماشین را تقویت می‌کند.

مقدمه:

فضای رنگی RGB پرکاربردترین فضای رنگی است. پردازش تصویر رنگی قرمز (R)، سبز (G) و آبی (B) مقادیر جزء پیکسل منطبق به عنوان نشانگر برای نقاط رنگی در این فضا می‌باشد.



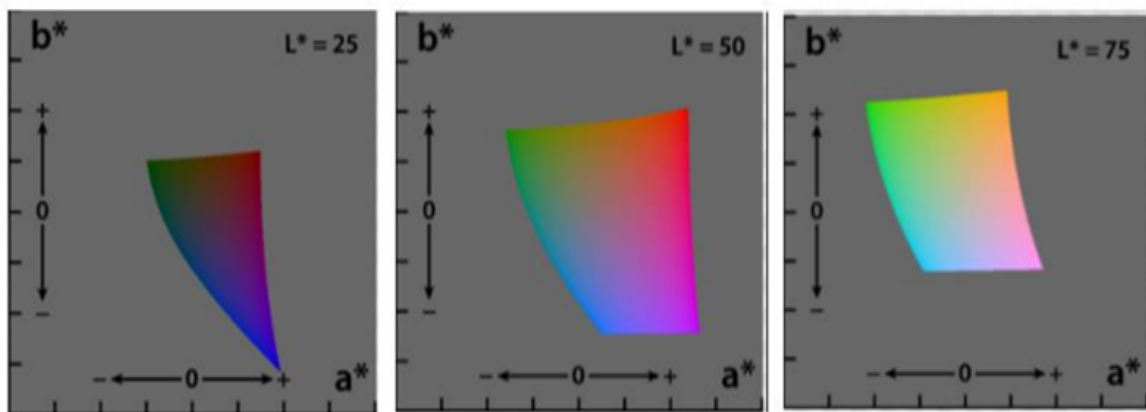
شکل ۸: فضای رنگ RGB

در دنیای رنگ دیجیتال، فضای رنگ LAB به عنوان ابزاری قدرتمند برای نمایش و دستکاری دقیق رنگ‌ها برجسته است. برخلاف مدل‌های رنگی آشنا مانند RGB و CMYK، LAB رویکرد منحصر به فردی برای توصیف رنگ‌ها ارائه می‌دهد که به طور نزدیک با درک انسانی همخوانی دارد و برای درک گرفتن تمام رنگ‌های قابل مشاهده برای چشم انسان طراحی شده است. این بر اساس نحوه درک انسان از رنگ است و آن را به ابزاری بسیار مفید برای متخصصان رنگ تبدیل می‌کند. فضای رنگی LAB از سه جزء تشکیل شده است:

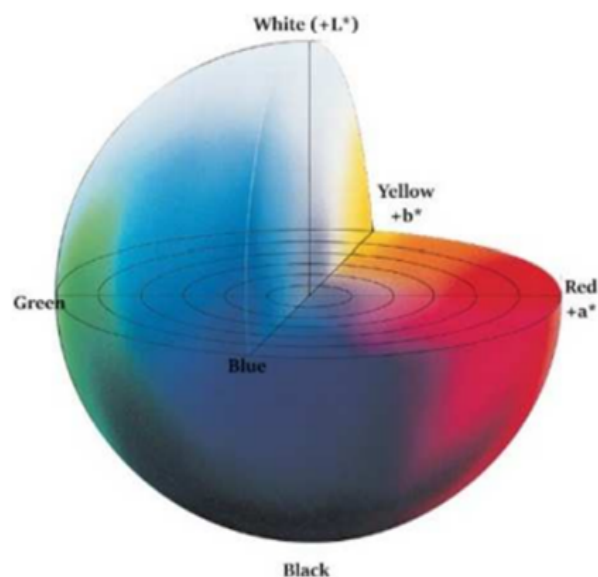
• **L^* (Lightness):** روشنایی یک رنگ را نشان می‌دهد که از ۰ (سیاه) تا ۱۰۰ (سفید) متغیر است.

• **a^* :** نشان‌دهنده محور سبز-قرمز، با مقادیر منفی نشان‌دهنده سبز و مقادیر مثبت نشان‌دهنده قرمز است.

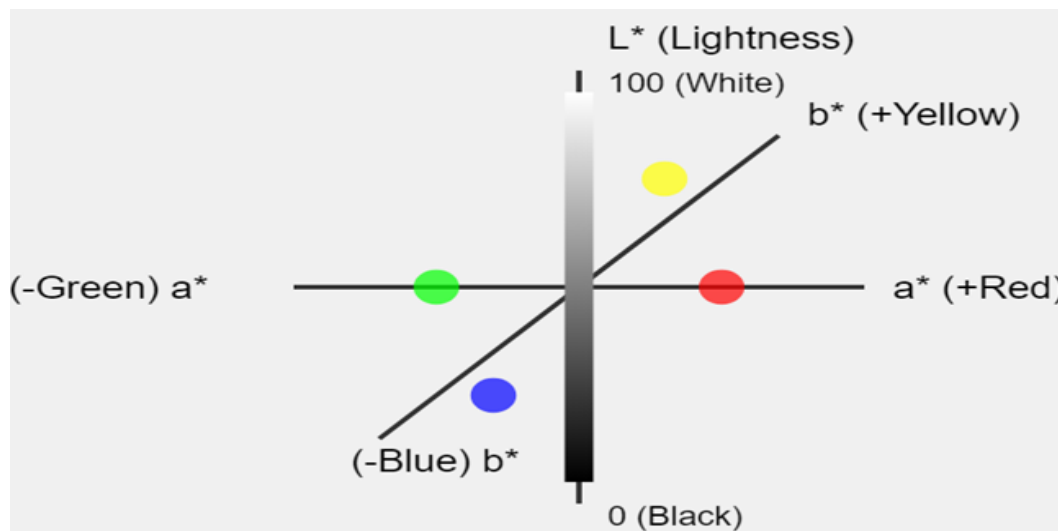
• **b^* :** نشان‌دهنده محور آبی-زرد، با مقادیر منفی نشان‌دهنده آبی و مقادیر مثبت نشان‌دهنده زرد است.



شکل ۹: فضای رنگی LAB



شکل ۱۰: فضای رنگ LAB: "روشنایی" از قسمت‌های دیگر جدا می‌شود زیرا وقتی تنظیم می‌شود، تغییر بیشتر شبیه دید انسان است و نتیجه "درست" بیشتری را برای چشم انسان فراهم می‌کند.



شکل ۱۱: فضای رنگ LAB

مزیت نسبت به سایر مدل‌های رنگی:

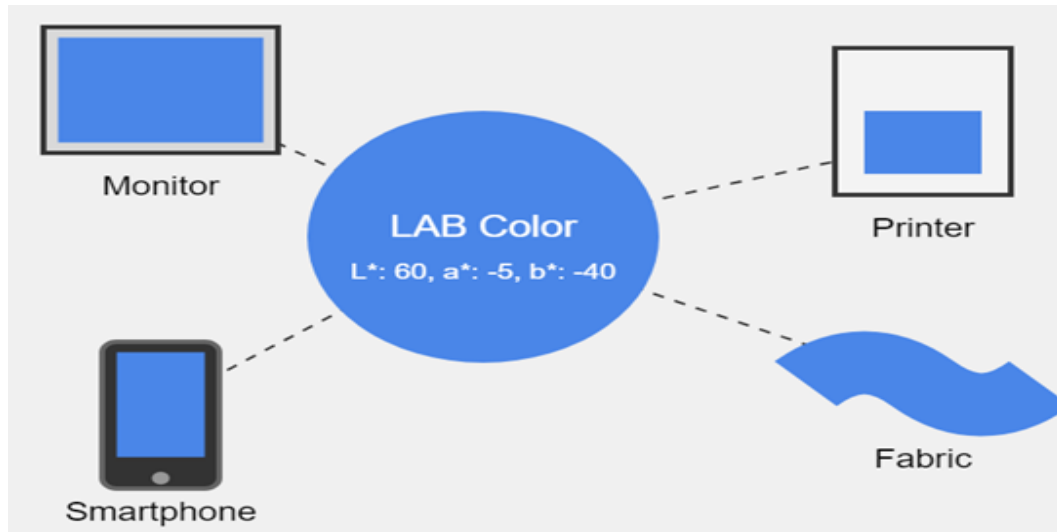
برخلاف RGB یا CMYK که به دستگاه‌ها یا روش‌های خروجی خاصی گره خورده‌اند، رنگ LAB مستقل از دستگاه است. این بدان معناست که رنگ‌های تعریف‌شده در فضای LAB در دستگاه‌ها و رسانه‌های مختلف یکدست ظاهر می‌شوند و آن را برای گردش‌های کاری حساس به رنگ ایده‌آل می‌کند. همچنین LAB طیف وسیع‌تری از رنگ‌ها را نسبت به RGB یا CMYK در بر می‌گیرد، از جمله رنگ‌هایی که نمی‌توانند در مانیتورهای معمولی یا به صورت چاپی بازتولید شوند. این باعث می‌شود آن را برای ذخیره و دستکاری داده‌های تصویری با کیفیت بالا ارزشمند کند.

فضای رنگی LAB طوری طراحی شده است که از نظر ادراکی یکنواخت باشد، به این معنی که همان مقدار تغییر عددی در مقادیر رنگ مربوط به همان مقدار تغییر درک‌شده بصری است. این کار تنظیمات دقیق رنگ را آسان‌تر می‌کند. با جداسازی روشنایی از اطلاعات رنگ، LAB امکان ویرایش دقیق و بصری رنگ را فراهم می‌کند. به عنوان مثال، شما می‌توانید روشنایی یک تصویر را بدون تأثیر بر تعادل رنگ آن تنظیم کنید.

کاربردهای فضای رنگی LAB:

عکاسان حرفه‌ای اغلب تصاویر را برای تصحیح و بهبود رنگ دقیق به LAB تبدیل می‌کنند. این به ویژه برای تنظیم کنتراست بدون تأثیر بر اشباع رنگ مفید است. همچنین LAB برای اطمینان از دقت رنگ در فرآیندهای مختلف چاپ و زیرلایه‌ها در گردش‌های کاری چاپ با کیفیت بالا استفاده می‌شود. این به پیش‌بینی چگونگی ظاهر شدن رنگ‌ها در خروجی چاپ نهایی کمک می‌کند.

ویراستاران ویدئو و رنگ‌سازان از LAB برای انجام تنظیمات دقیق رنگ در فیلم و فیلم پس از تولید استفاده می‌کنند. توانایی آن در تفکیک روشنایی از اطلاعات رنگ به ویژه در این زمینه ارزشمند است. همچنین LAB به عنوان یک زبان رنگ مشترک در سیستم‌های مدیریت رنگ عمل می‌کند و ترجمه دقیق رنگ را بین دستگاه‌های مختلف و فضاهای رنگی تسهیل می‌کند.



شکل ۱۲: کاربرد این فضا

تبدیل فضای رنگ LAB به RGB:

تبدیل فضای رنگ LAB به RGB یک فرآیند کلیدی در پردازش تصویر و نمایش داده‌های رنگی است، زیرا فضای رنگ LAB که به طور خاص برای تطابق با سیستم بینایی انسان طراحی شده است، اغلب در دستگاه‌های دیجیتال که از فضای رنگ RGB استفاده می‌کنند، قابل نمایش نیست و بسیاری از نرم‌افزارهای ویرایش تصویر از فضای LAB برای اصلاح رنگ استفاده می‌کنند و سپس تصاویر را به RGB تبدیل می‌کنند.

در چاپ حرفه‌ای، ابتدا رنگ‌ها در فضای LAB تنظیم می‌شوند و سپس برای تطابق با چاپگر به RGB یا CMYK تبدیل می‌شوند. هیچ فرمول ساده‌ای برای تبدیل رنگ RGB به مدل رنگ آزمایشگاهی وجود ندارد، بنابراین ابتدا باید RGB را به یک مدل رنگ مشخص به عنوان CIEXYZ و سپس به فضای رنگ آزمایشگاهی تبدیل کنیم. فضای LAB ابتدا باید به فضای XYZ، که یک مدل مرجع برای توصیف رنگ‌ها است، تبدیل شود. این تبدیل غیرخطی است و شامل فرمول‌های زیر می‌باشد:

معادله شماره ۱)

$$G(t) = \begin{cases} t^3 & \text{if } t > \sigma \\ 3\sigma^2 \left(t - \frac{\sigma}{3}\right) & \text{otherwise} \end{cases}$$

where : $\sigma = 29$

معادله شماره ۲) کانال‌های LAB براساس تابع

$$\begin{aligned} L^* &= 116 \left(\frac{Y}{Y_n} \right) - 16 \\ a^* &= 500 \left(g\left(\frac{X}{X_n}\right) - g\left(\frac{Y}{Y_n}\right) \right) \\ b^* &= 200 \left(g\left(\frac{Y}{Y_n}\right) - g\left(\frac{Z}{Z_n}\right) \right) \end{aligned}$$

معادله شماره ۳) تبدیل پارامترهای کانال LAB به فضای XYZ

$$X' = X'_{116} g^{-1} \left(\frac{L' + 16}{116} + \frac{a'}{500} \right)$$

$$Y' = Y'_{116} g^{-1} \left(\frac{L' + 16}{116} \right)$$

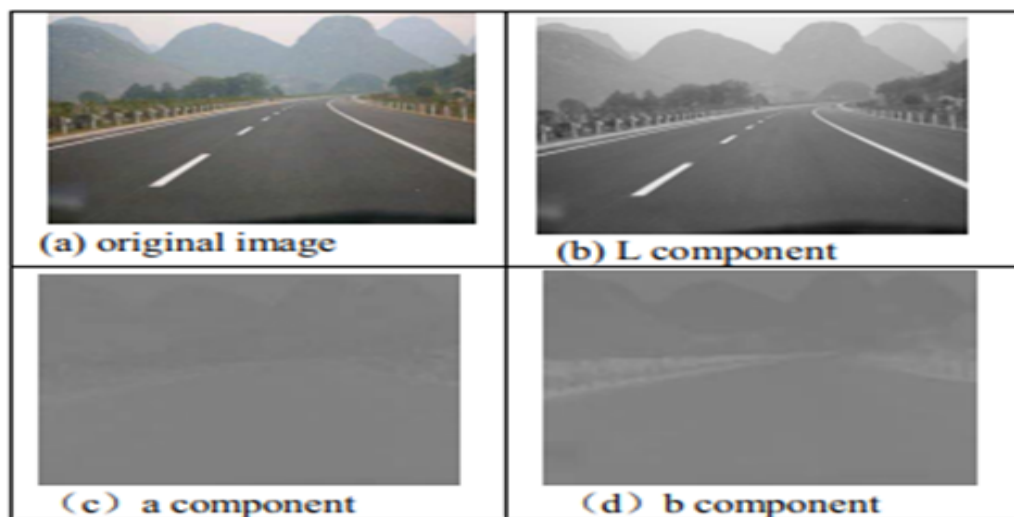
$$Z' = Z'_{116} g^{-1} \left(\frac{L' + 16}{116} + \frac{b'}{200} \right)$$

معادله شماره ۴) تبدیل پارامترهای فضای XYZ به RGB

$$0.4986Z - 0.5372Y - 0.2406X = R$$

$$0.415Z + 0.8758Y - 0.9689X = G$$

$$0.570Z + 0.2040Y - 0.557X = B$$



شکل ۱۳: تصویر رنگی RGB و تبدیل آن به تصاویر سه کاناله L, a, b

نتیجه گیری:

فضای رنگی LAB جایگزین قدرتمندی برای مدل های رنگی سنتی است که دقت، انعطاف پذیری و استقلال دستگاه را بیشتر می کند. همانطور که گردش های کاری حیاتی رنگ پیچیده تر می شوند، درک و استفاده از فضای رنگی LAB می تواند به طور قابل توجهی شیوه های مدیریت رنگ را در زمینه های مختلف خلاقانه و فنی افزایش دهد.

با در نظر گرفتن فضای رنگی LAB، متخصصان می توانند به کنترل رنگ دقیق تری دست یابند، از ثبات در رسانه های مختلف اطمینان حاصل کنند و در نهایت خروجی رنگ با کیفیت بالاتری را در پروژه های خود ارائه دهند. همانطور که تکنولوژی رنگ به تکامل خود ادامه می دهد، فضای رنگی LAB سنگ بنای تکنیک های پیشرفته مدیریت رنگ باقی می ماند.

منابع و مراجع

- [1] Winter, Martin, Bischof, Horst, and Fraundorfer, Friedrich. Maximally stable corner clusters: A novel distinguished region detector and descriptor. 01 2005.