

Frequency Domain

نسیم فانی

اطلاعات گزارش	چکیده
تاریخ: 99/10/4	در این تمرین، به بررسی انواع فضاهاى رنگی، پردازش تصاویر در برخی از این فضاها و Quantization می پردازیم.
واژگان کلیدی: فضاهای رنگی مدل RGB Quantization	

1-مقدمه

استفاده از رنگ در پردازش تصویر، ناشی از دو عامل است:

1. رنگ توصیفگر قدرتمندی است که غالباً شناسایی و استخراج اشیا را از صحنه آسان می سازد.

2. انسان میتواند در مقایسه با فقط 24 سایه خاکستری، هزاران سایه رنگ و شدت را تشخیص دهد. این عامل دوم، مخصوصاً در تحلیل تصویر دستی (یعنی وقتی که توسط انسان انجام می گیرد) مهم است.

مدل ها و سیستم های مختلفی جهت تعیین و مشخص کردن رنگ های واقعی وجود دارد، که هر کدام از مدل ها و سیستم های یک سیستم مختصاتی دارد که در داخل آن ها هر رنگ توسط یک نقطه نشان داده می شود. مدل RGB، قرمز، سبز، آبی، به طور گسترده در تصویربرداری، پردازش و نمایش تصویر استفاده می شود. به عنوان مثال در دروین

های فیلمبرداری رنگی و مانیتورهای رنگی از مدل RGB استفاده می شود. بسیاری از رنگ ها را می توان با ترکیب سه رنگ اولیه، قرمز، سبز و آبی، همانند پاسخ سلولهای مخروطی در بخش شبکیه چشم انسان، ایجاد کرد.

فضاهای رنگی در پردازش تصویر به منظور تسهیل مشخصات رنگ ها به روشی استاندارد به کار می-روند.

انواع مختلفی از فضاهاى رنگی در زمینه های مختلف مانند سخت افزار، ایجاد انیمیشن و غیره استفاده می شوند.

مسیر بینایی انسان از سه نوع سلولهای مخروطی جهت درک و تفکیک هزاران رنگ مختلف استفاده می کند، این در حالی است که مسیر بینایی انسان حدوداً 30 سطح شدت روشنایی را میتواند از هم تفکیک کند. این قابلیت توانایی ما را در تشخیص و شناسایی اشیاء در یک صحنه را گسترش می دهد. در حالی که تصاویر رنگی، تصاویری که توسط یک

دوربین گرفته می شوند، رنگ های دیده شده توسط چشم را روی یک عکس تکرار می کنند، تصاویر پزشکی رنگ ها را ثبت نمی کنند، و تنها شدت روشنایی را ثبت می کنند.

برای مثال، شدت روشنایی در تصاویر پزشکی نشان دهنده ی میزان جذب اشعه X توسط بدن، یا میزان امواج ساطع شده توسط بدن در اولتراسوند می باشد.

در برخی موارد اضافه کردن رنگ به تصویر، شاید درک بهتر ویژگی های تصویر مفید باشد. رنگ اضافه شده به تصویر، رنگ کاذب یا شبه رنگ است، و رنگ های واقعی دیده شده توسط چشم نخواهد بود.

2-توضیحات تکنیکال

5.1

فضای رنگی یک سازمان خاص از رنگ ها است. در ترکیب با نمایه رنگ که توسط دستگاه های مختلف فیزیکی پشتیبانی می شود، و از نمایش های قابل تکرار رنگ پشتیبانی می کند - خواه این نمایش شامل نمایش آنالوگ باشد یا نمایشی دیجیتال.

یک فضای رنگی ممکن است دلخواه باشد، یعنی با رنگ های تحقق یافته فیزیکی که به مجموعه رنگ های فیزیکی اختصاص داده شده با نام رنگ اختصاص داده شده مربوطه (از جمله اعداد گسسته) به عنوان مثال در مجموعه (Pantone) اختصاص داده شده باشد، یا با دقت ریاضی ساختار یافته باشد (مانند سیستم NCS، Adobe RGB و sRGB).

"مدل رنگ" یک مدل ریاضی انتزاعی است که نحوه نمایش رنگ ها را به صورت مجموعه اعداد توصیف می کند (به عنوان مثال سه بردار در RGB یا چهار بردار در CMYK). با این حال، یک مدل رنگی بدون عملکرد نقشه برداری مرتبط با یک فضای

رنگی مطلق، یک سیستم رنگی کم و بیش دلخواه است و هیچ ارتباطی با هیچ سیستم تفسیر رنگی در سطح جهانی ندارد.

افزودن یک تابع نگاشت خاص بین یک مدل رنگی و یک فضای رنگی مرجع، در فضای رنگی مرجع یک "رد پا" مشخص ایجاد می کند، که به عنوان گستره شناخته می شود و برای یک مدل رنگی مشخص شده، این یک فضای رنگی را تعریف می کند. به عنوان مثال، Adobe RGB و sRGB دو فضای رنگی مطلق متفاوت هستند که هر دو براساس مدل رنگی RGB ساخته شده اند.

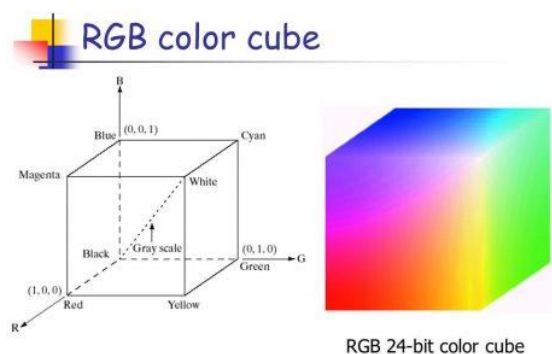
هنگام تعریف یک فضای رنگی، استاندارد مرجع معمول فضای رنگی CIELAB یا CIEXYZ است که به طور خاص برای در بر گرفتن همه رنگ هایی که یک انسان متوسط می تواند ببیند، طراحی شده اند.

از آنجا که "فضای رنگ" ترکیبی خاص از مدل رنگ و عملکرد نگاشت را مشخص می کند، این کلمه غالباً غیر رسمی برای شناسایی یک مدل رنگ استفاده می شود. با این حال، حتی اگر شناسایی یک فضای رنگی به طور خودکار مدل رنگی مرتبط را شناسایی می کند، این کاربرد به معنای دقیق نادرست است. به عنوان مثال، اگرچه چندین فضای رنگی خاص براساس مدل رنگی RGB بنا شده است، اما چیزی به عنوان فضای رنگی RGB منفرد وجود ندارد.

برخی از فضا های رنگی رایج عبارتند از:

- RGB
- CMY'K
- Y'UV
- YIQ
- Y'CbCr
- HSV

❖ مدل RGB:



❖ مدل HSV:

گرچه، مدل RGB به نظر می آید که با مدل فیزیولوژیکی چشم انسان که سلولهای مخروطی آن هر کدام به یک رنگ خاصی حساس است، مطابقت بیشتری داشته باشد، اما در برخی موارد استفاده از روشهای پردازش تصاویر رنگی مدل RGB باعث ایجاد نتایج ناخواسته می شود.

برخی از روشهای پردازش تصاویر سطح خاکستری، می توانند به صورت مستقیم روی تصاویر رنگی مدل RGB اعمال شوند، یعنی الگوریتم روی هر صفحه رنگی قرمز، سبز و آبی به صورت جدا اعمال شوند، اما در برخی موارد، نمی توان از این روش ها به صورت مستقیم روی تصاویر رنگی RGB استفاده کرد. و در برخی موارد انجام این عمل باعث افزایش پیچیدگی محاسباتی می شود، و شاید نتوان در کارهای زمان حقیقی از این روش ها استفاده کرد. برای مثال روش های بهبود لبه های تصویر، در فضاهای رنگی ای که شدت روشنایی (مقدار شدت روشنایی پیکسل) را از اطلاعات رنگی (Chrominance) جدا می کند، کارایی بیشتری دارند.

Chrominance توسط دو پارامتر فام (hue) و اشباع (saturation) تعیین می شود؛ فام فرکانس رنگ غالب می باشد و اشباع میزان خالص بودن

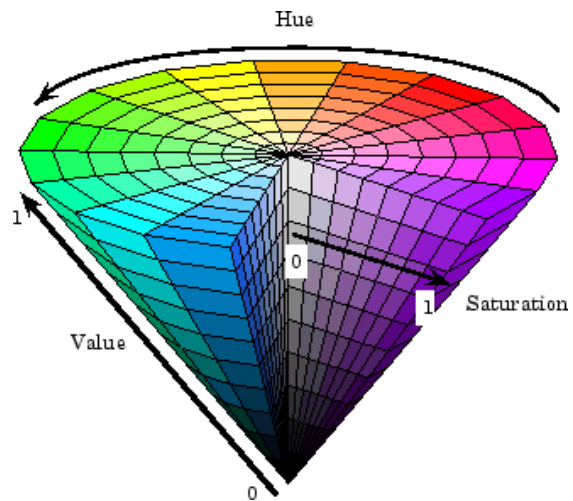
همانطور که در شکل زیر مشاهده می کنید، فضای رنگی RGB، یک مکعب واحد است و هر محور یکی از رنگهای اولیه را نشان می دهد. مبدا مختصات، جایی است که مکعب فاقد سه رنگ اصلی است و این نقطه نشان دهنده رنگ سیاه است، در حالی که رأس مخالف، ترکیب سه رنگ اولیه است، و این نقطه در مکعب نشان دهنده رنگ سفید می باشد. رأس های دیگر نشان دهنده ی رنگهای ثانویه فیروزه ای، بنفش و زرد می باشند که هر کدام از این رنگ های ثانویه از ترکیب دو رنگ اصلی بدست می آیند.

در این مدل، تمام رنگهای دیگر در داخل مکعب توسط سه مولفه مشخص می شوند، هر نقطه میزان مقادیر رنگهای اصلی مورد نیاز جهت ایجاد یک رنگ خاص را مشخص می کند. هر کدام از این مولفه ها، معمولاً با یک بایت (محدوده ای از 0-255)، تعیین می شود. برای مثال، قرمز روشن مقدار (255,0,0) و زرد روشن مقدار (255,255,0) دارد. از آنجا که رنگهای قرمز، آبی و سبز هر کدام میتوانند به صورت مستقل تعیین شوند، می توان 2563 رنگ با سه رنگ اصلی تولید کرد.

در فایل تصویر، سه صفحه، یا بهتر است بگوییم سه ماتریس، که هر صفحه برای یک رنگ اصلی می باشد، برای ذخیره هر تصویر استفاده می شود. از آنجا که هر پیکسل، توسط سه بایت مشخص می شود، عمق تصویر 24 بیت می باشد و حجم هر تصویر رنگی سه برابر بیشتر از حجم یک تصویر سطح خاکستری است. تصاویر رنگی RGB را با اسم تصاویر true color و یا 24 بیتی نیز می شناسند.

رنگ را مشخص می‌کند، برای مثال میزان نور سفید ترکیب شده با طیف رنگی را میزان خالص بودن رنگ مشخص می‌کند.

برای مثال، از ترکیب نور سفید با رنگ قرمز (اشباع شده)، رنگ صورتی (کمتر اشباع شده) تولید می‌شود. در چنین فضای رنگی، HSV (hue, saturation, value) یا HSB (hue, saturation, brightness) جهت بهبود لبه (تیز کردن لبه‌ها) یا صاف کردن لبه‌ها تنها مقادیر مولفه value/brightness نیاز به پردازش دارند.



اعمال الگوریتم‌های تیزسازی (sharpening)، یا صاف کردن (smoothing) لبه‌ها بر روی تصاویر با فضای رنگی RGB باعث شیف‌ت پیدا کردن سطوح رنگی (تغییر رنگها) می‌شوند.

چشم انسان به تغییرات جزئی در فضای رنگی RGB که توسط الگوریتم‌های sharpening و smoothing ایجاد می‌شوند، بسیار حساس است و تغییرات جزئی در فضای رنگی RGB را به صورت قابل توجهی تشخیص می‌دهد و نمی‌تواند این تغییرات را نادیده بگیرد، این در حالی است که تغییرات جزئی در اشباع و یا شدت روشنایی را متوجه نمی‌شود.

تبدیلات خطی را می‌توانیم در فضای رنگی RGB بر روی هر صفحه R, G و B به صورت جداگانه

اعمال کنیم، اما نمیتوانیم تبدیلات غیرخطی مثل، متعادل کردن هیستوگرام تصویر، فیلتر میانه، را در فضای رنگی RGB انجام دهیم.

اگر قرار است عملیات غیرخطی روی تصاویر انجام دهیم، ابتدا باید تصاویر رنگی از فضای رنگی RGB به فضای رنگی HSV/HSB تبدیل شوند، و سپس در این فضا عملیات بر روی صفحه شدت روشنایی (brightness / value) اعمال شوند، و البته میتوان روی صفحه saturation هم عملیات را انجام دهیم اما نباید تبدیلاتی روی صفحه hue انجام دهیم.

معمولاً در ابتدا تصویر RGB را به فضاهای رنگی دیگری منتقل می‌کنند، سپس الگوریتم‌های پردازشی را روی تصاویر در فضای رنگی جدید مثل HSI اعمال می‌کنند، و در نهایت برای نمایش تصویر، تصویر به فضای رنگی RGB برگردانند. روشهای زیادی برای تبدیل فضای رنگی RGB به سایر فضاهای رنگی وجود دارد.

❖ مدل HSI:

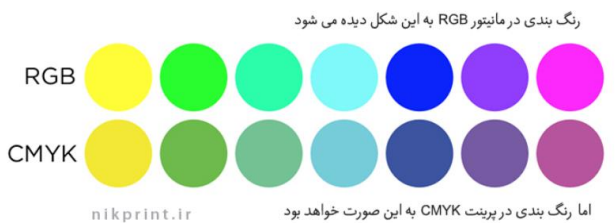
I) (و مؤلفه‌ی مقدار (V) است، 1 مدل HSI در حقیقت به لحاظ محاسبه‌ی روشنایی، متفاوت هستند. توزیع و محدوده‌ی فعال I یا V و اشباع را روشنایی تعیین می‌کند. در برخی منابع از مدل HLS نیز نام برده شده که دقیقاً مشابهی HSI است و در (L) برای اشاره به مفهوم روشنایی استفاده شده است. مناسب‌ترین مدل آن از خود واژه‌ی "روشنایی برای فرآیندهای سنتی پردازش تصویر مانند پیچش، تعدیل، عملیاتهای مبتنی بر هیستوگرام و ... که در آنها عملیات با دستکاری میزان درخشندگی انجام میشود، مدل HSI است. زیرا مؤلفه‌ی I آن به شدت وابسته به R, G و B

است. اما در عملیات‌های مبتنی بر تغییر اشباع (برای جابه‌جایی رنگ‌ها یا تغییر میزان رنگ)، مدل HSV ارجح است. زیرا محدوده‌ی اشباع وسیع‌تری را پوشش می‌دهد.

رنگ‌هایی که می‌توان از ترکیب چهار رنگ اصلی ساخت محدودیت‌هایی وجود دارد. در مواردی مانند چاپ بیلبرد، مجله، کاتالوگ و بروشور از مد رنگی CMYK استفاده می‌شود. از جمله نقاط ضعف این مد رنگی هزینه بالای رنگ‌ها می‌باشد.

نقاط قوت رنگ‌های CMYK

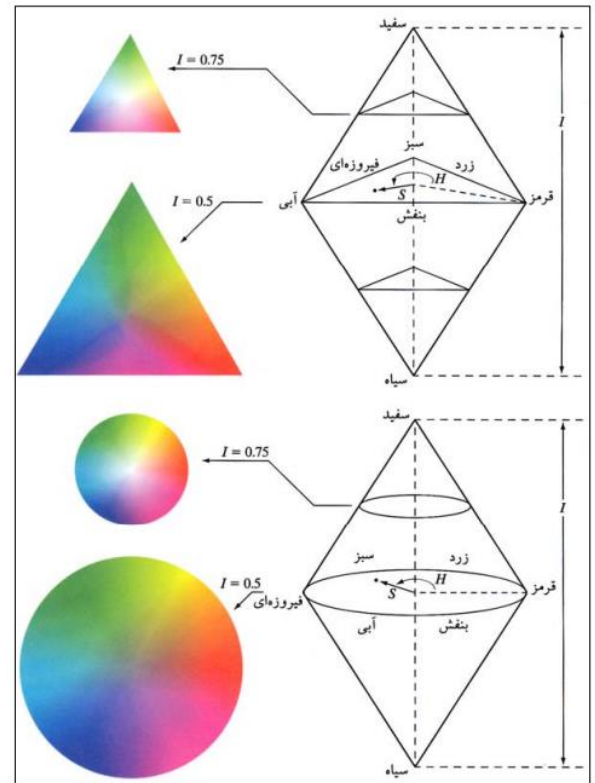
- ۱- جبران پردازنده چاپ
- ۲- رنگ شفاف بر روی کاغذ چاپ آبی-سفید
- ۳- رنگ‌های گرم‌تر بر روی کاغذ کرم-سفید



❖ مدل sRGB

sRGB یک فضای رنگی RGB (قرمز، سبز، آبی) است که HP و Microsoft به طور مشترک در سال 1996 ایجاد کردند تا از آن در مانیتورها، چاپگرها و وب استفاده کنند. متعاقباً توسط IEC به عنوان IEC 61966-2-1: 1999 استاندارد شد. این فضا غالباً فضای رنگی "پیش فرض" برای تصاویر است که فاقد اطلاعات مربوط به فضای رنگ هستند، خصوصاً اگر پیکسل تصاویر در عده‌های صحیح 8 بیتی در هر کانال رنگ ذخیره شده باشد.

sRGB از مقدمات اولیه ITU-R BT.709 استفاده می‌کند، همان مانیتورهای استودیویی و HDTV، یک تابع انتقال (گاما) معمولی برای CRT ها و یک محیط مشاهده برای مطابقت با شرایط معمول

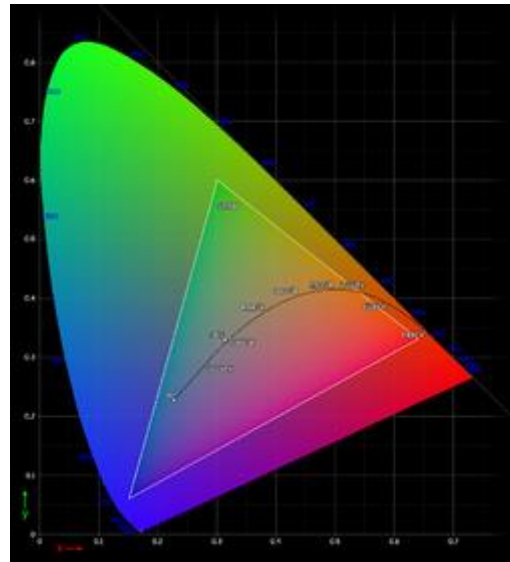


مدل رنگ HSI مبتنی بر صفحات (الف) مثلثی، (ب) دایره‌ای، مثلث‌ها و دایره‌ها بر محور عمودی شدت، عمود هستند.

❖ CMYK

مد یا رنگ‌بندی CMYK که به آن مد چهاررنگ هم گفته می‌شود از حروف چهار رنگ اصلی Cyan (آبی)، Magenta (قرمز)، Yellow (زرد) و Black (مشکی) برگرفته شده است. مد CMYK در صنعت چاپ و تقریباً در هر جای دیگری استفاده می‌شود که قرار است رنگ‌ها به صورت فیزیکی بر روی کاغذ یا هر چیز دیگری تظاهر یابند. در صنعت چاپ، رنگ‌های مختلف با مد CMYK و از مخلوط کردن چهار رنگ آبی، قرمز، زرد و مشکی ساخته می‌شوند. چون در این مد، رنگ‌ها به صورت فیزیکی مخلوط و ساخته می‌شوند، بنابراین در تنوع و تعداد

مشاهده در منزل و دفترکار. این مشخصات باعث می شد sRGB مستقیماً در مانیتورهای معمولی CRT آن زمان نمایش داده شود، که به پذیرش آن بسیار کمک کرد.



❖ مدل xvYCC یا extended-gamut YCbCr

xvYCC یا گستره‌ی گسترده YCbCr یک فضای رنگی است که می‌تواند در وسایل الکترونیکی ویدئویی دستگاه‌های تلویزیونی مورد استفاده قرار گیرد تا از گستره 1.8 برابر فضای رنگ sRGB پشتیبانی کند. xvYCC توسط سونی پیشنهاد شد، توسط IEC در اکتبر 2005 تعیین شد و در ژانویه 2006 با عنوان IEC 61966-2-4 منتشر شد.

ویدیوی رمزگذاری شده xvYCC همان رنگ اولیه و نقطه سفید BT.709 را حفظ می‌کند و از ماتریس تبدیل و کدگذاری BT.601 یا BT.709 RGB به YCC استفاده می‌کند که این امکان را برای آن فراهم می‌کند تا از طریق مسیرهای

موجود دیجیتال YCC حرکت کند و هر رنگی در محدوده طبیعی سازگار خواهد بود. این کار با اجازه دادن به ورودی های RGB منفی و گسترش رنگ خروجی کار می‌کند. اینها برای رمزگذاری رنگهای اشباع تر با استفاده از بخش بیشتری از مقادیر RGB قابل رمزگذاری در سیگنال YCbCr در مقایسه با آنچه در Broadcast Safe Level استفاده می‌شود، استفاده می‌شوند. سپس رنگ‌های اضافی را می‌توان توسط دستگاهی نمایش داد که فناوری اساسی آن با مقدماتی استاندارد محدود نمی‌شود.

در مقاله ای که توسط انجمن نمایش اطلاعات در سال 2006 منتشر شد، نویسندگان 769 رنگ را در آبشار مونسل به فضای BT.709 و به فضای xvYCC ترسیم کردند. حدود 55٪ از رنگهای Munsell را می‌توان در محدوده sRGB ترسیم کرد، اما 100٪ آن رنگها را در محدوده xvYCC می‌توان ترسیم کرد و همچنین می‌توان رنگ های عمیق تری ایجاد کرد - به عنوان مثال یک فیروزه ای عمیق تر با دادن منفی به ماده اولیه اصلی (قرمز).

انگیزه ایجاد xvYCC این واقعیت بود که فناوری های مدرن نمایشگر و ضبط اغلب دارای مقدماتی اولیه RGB با اشباع قابل توجهی بالاتر از نمایشگرهای CRT سنتی هستند (که به عنوان پایه sRGB و Rec. 709 عمل می‌کنند)، به آنها امکان می‌دهد طیف وسیع تری از رنگ را کنترل کنند. اما این دستگاه ها بدون برهم زدن کالیبراسیون اساسی قادر به انجام این کار نبوده اند، زیرا تمام سیستم های ذخیره سازی و انتقال فیلم موجود بر اساس مقدماتی CRT هستند و بنابراین به محدوده CRT محدود می‌شوند.

دارای اطلاعاتی در محدوده نارنجی-آبی هستند ، برای دستیابی به وفاداری رنگی مشابه ، باید به هر دو ملفه همان پهنای باند من را بدهید.

تعداد کمی از تلویزیون ها رمزگشایی واقعی I و Q را انجام می دهند ، به دلیل هزینه های زیاد چنین عملیاتی. در مقایسه با رمزگشایی ارزانتر RY و BY که فقط به یک فیلتر احتیاج دارد ، I و Q هر کدام به فیلتر متفاوتی احتیاج دارند تا تفاوت پهنای باند بین I و Q را برآورده سازند.

این اختلافات پهنای باند همچنین نیاز دارد که فیلتر "I" دارای تأخیر زمانی باشد تا با تأخیر بیشتر در فیلتر 'Q'. رادیو دیجیتال راکول مدولار (MDR) یکی از مجموعه های رمزگشایی I و Q بود که در سال 1997 می توانست در حالت فریم در یک زمان با کامپیوتر یا در زمان واقعی با پردازشگر سریع IQ (FIQP) کار کند.

برخی از گیرنده های تلویزیون خانگی RCA Colortrak " ساخته شده در حدود سال 1985 نه تنها از رمزگشایی I / Q استفاده می کردند ، بلکه از مزایای آن به همراه مزایای فیلتر شانه به عنوان "100 درصد پردازش" کامل برای ارائه محتوای اصلی تصویر اصلی رنگ تبلیغ می کردند. پیش از این ، بیش از یک مارک تلویزیونی رنگی (RCA ، آروین) از رمزگشایی I / Q در سال مدل 1954 یا 1955 در مدل های با صفحه نمایش حدود 13 اینچ (اندازه گیری مورب) استفاده کرده است. تلویزیون اصلی طرح ریزی ظهور از رمزگشایی I / Q استفاده می کرد.

❖ مدل YIQ

YIQ فضای رنگی است که توسط سیستم تلویزیون رنگی NTSC استفاده می شود و عمدتاً در آمریکای شمالی و مرکزی و ژاپن استفاده می شود. I مخفف in-phase است ، در حالی که Q مخفف quadrature است و به اجزای استفاده شده در مدولاسیون دامنه quadrature اشاره دارد. برخی از اشکال NTSC اکنون از فضای رنگی YUV استفاده می کنند که توسط سیستم های دیگری مانند PAL نیز استفاده می شود.

مولفه Y نشان دهنده اطلاعات luma است و تنها مولفه ای است که توسط گیرنده های تلویزیون سیاه و سفید استفاده می شود. I و Q نشان دهنده اطلاعات رنگزا هستند. در YUV می توان اجزای U و V را مختصات X و Y در فضای رنگ دانست. من و Q را می توان به عنوان یک جفت دوم محور در همان نمودار تصور کرد که با چرخش 33 درجه است. بنابراین ضریب هوشی و UV نشان دهنده سیستم های مختصات مختلف در یک صفحه هستند.

سیستم YIQ در نظر گرفته شده است تا از ویژگی های واکنش رنگ انسان بهره برد. چشم نسبت به تغییر در محدوده نارنجی-آبی (I) نسبت به محدوده سبز-بنفش (Q) حساسیت بیشتری دارد - بنابراین پهنای باند کمتری برای Q مورد نیاز است. پخش NTSC محدودیت I تا 1.3 مگاهرتز و Q تا 0.4 مگاهرتز را نشان می دهد. I و Q فرکانس درون سیگنال 4 مگاهرتز Y قرار دارند ، که پهنای باند سیگنال کلی را به 4.2 مگاهرتز کاهش می دهد. در سیستم های YUV ، از آنجا که U و V هر دو

❖ Y اطلاعات luma (روشنایی یا

درخشندگی) و همگام سازی (همگام

سازی) را حمل می کند. $Y = 0.2126 R + 0.7152 G + 0.0722 B$

تولیز یون رنگی ، محور Y بر روی نمایشگر

اسیلوسکوپ شکل موج ویدئویی شدت خط

اسکن را نشان می دهد. با رنگ ، Y هنوز

شدت را نشان می دهد اما ترکیبی از

رنگهای جز component است.

❖ P_B تفاوت بین آبی و luma را به همراه

دارد $(B - Y)$.

❖ P_R تفاوت بین قرمز و لوما را دارد $(R - Y)$.

❖ P_B تفاوت بین آبی و luma را به همراه

دارد $(B - Y)$.

ارسال سیگنال سبز به عنوان مولفه چهارم زائد است

، زیرا می توان با استفاده از اطلاعات آبی ، قرمز و

luma آن را بدست آورد.

هنگامی که سیگنالهای رنگی برای اولین بار به

استاندارد فیلم سیاه و سفید کدگذاری شده توسط

NTSC اضافه شدند ، رنگ با تغییر فاز یک زیر

حامل مرجع رنگ نشان داده شد. P برای اطلاعات

فاز یا تغییر فاز برای نشان دادن اطلاعات رنگ حتی

در مواردی که دیگر از تغییر فاز برای نشان دادن

رنگ استفاده نمی شود ، انجام شده است. بنابراین ،

نامگذاری $Y P_B P_R$ از معیارهای مهندسی ایجاد

شده برای استاندارد رنگ NTSC گرفته شده است.

از همین کابل ها می توان برای YPbPr و فیلم

ترکیبی استفاده کرد. این بدان معناست که کابل -

های اتصال دهنده RCA زرد ، قرمز و سفید که

معمولاً با اکثر تجهیزات صوتی / تصویری بسته

بندی می شوند ، می توانند به جای اتصالات

YPbPr استفاده شوند ، به شرطی که کاربر نهایی

مراقب باشد هر کابل را به اجزای مربوطه در هر دو

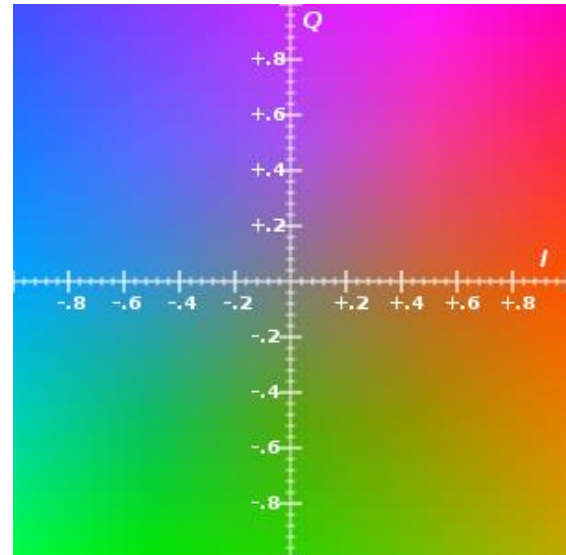


Figure- 1 IQ color space at Y=0.5

❖ فضای رنگی YPbPr

YPbPr یا Y'PbPr ، که به عنوان $Y P_B P_R$ نیز

نوشته می شود ، یک فضای رنگی است که در

الکترونیک ویدئو استفاده می شود ، به ویژه در مورد

کابل های ویدیویی. YPbPr نسخه آنالوگ فضای

رنگی YCbCr است. این دو عدد برابر هستند اما

YPbPr برای استفاده در سیستم های آنالوگ

طراحی شده در حالی که YCbCr برای فیلم

دیجیتال در نظر گرفته شده است.

از تولید کنندگان معمولاً از YPbPr به عنوان

ویدیوی جز component یاد می شود. با این حال ،

انواع مختلفی از ویدیوی component وجود دارد

که بیشتر آنها به نوعی RGB هستند. برخی از

کارت های ویدئویی دارای پورت های خروجی ویدئو

در ویدئو (VIVO) برای اتصال به دستگاه های

ویدیویی component هستند.

YPbPr از سیگنال ویدئویی RGB تبدیل می شود

، که به سه جز Y ، P_B و P_R تقسیم می شود.

روش‌های مختلفی برای این عمل وجود دارد برای مثال:

- uniform quantization
- minimum variance quantization

اصطلاح مهمی در بحث کوانتایز تصویر، مکعب رنگی RGB است.

مکعب رنگی RGB یک آرایه سه بعدی از تمام رنگهایی است که برای یک نوع داده خاص تعریف شده است.

از آنجا که تصاویر RGB در MATLAB می‌توانند از نوع uint8، uint16 یا دو برابر باشند، سه تعریف ممکن برای مکعب رنگی وجود دارد.

به عنوان مثال، اگر یک تصویر RGB از کلاس uint8 باشد، 256 مقدار برای هر صفحه رنگی (قرمز، آبی و سبز) تعریف می‌شود و در کل، 224 (یا 16,777,216) رنگ وجود دارد که توسط مکعب رنگ تعریف شده است. این مکعب رنگی برای همه تصاویر RGB uint8 یکسان است، صرف نظر از اینکه از چه رنگ‌هایی در واقع استفاده می‌کنند.

مکعب‌های uint8، uint16 و double رنگ همه دارای طیف رنگی یکسانی هستند. به عبارت دیگر، روشن‌ترین قرمز در یک تصویر RGB uint8 همانند قرمز روشن در یک تصویر double RGB به نظر می‌رسد. تفاوت در این است که مکعب رنگی double RGB سایه‌های قرمز بیشتری دارد (و سایه‌های بیشتری از همه رنگ‌ها). شکل زیر یک مکعب رنگی RGB را برای یک تصویر uint8 نشان می‌دهد.

انتهای متصل کند. همچنین، بسیاری از تلویزیون‌ها از اتصال سبز یا فقط برای luma یا برای ورودی فیلم ترکیبی استفاده می‌کنند. از آنجایی که YPbPr حتی با رمزگشایی ویدیوی کامپوننت با بخش درخشندگی فیلم کامپوزیت سازگار است، می‌توان از طریق این ورودی از فیلم کامپوزیت استفاده کرد، اما فقط اطلاعات luma به همراه نقاط کروم نمایش داده می‌شود. برعکس تا زمانی که از i480 یا i576 استفاده شود برعکس است.

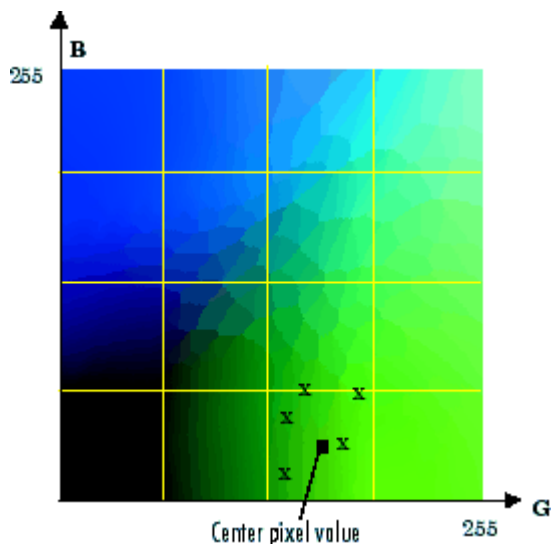
مزایا:

سیگنال‌ها با استفاده از YPbPr جدایی کافی را ایجاد می‌کنند که نیازی به مالتی پلکس شدن رنگ نیست، بنابراین کیفیت تصویر استخراج شده تقریباً یکسان با سیگنال از پیش رمزگذاری شده است. S-Video و فیلم ترکیبی با استفاده از مالتی پلکسینگ الکترونیکی سیگنال‌ها را با هم مخلوط می‌کنند. تخریب سیگنال برای فیلم ترکیبی معمول است، زیرا اکثر سیستم‌های نمایش قادر به جدا کردن کامل سیگنال‌ها نیستند، اگرچه تلویزیون‌های HDT تمایل دارند چنین تفکیکی را بهتر از اکثر واحدهای CRT انجام دهند (به خزیدن نقطه مراجعه کنید). S-Video می‌تواند برخی از این مشکلات بالقوه را کاهش دهد، زیرا luma آن جدا از کروم منتقل می‌شود.

در بین رابط‌های آنالوگ مصرف‌کننده، فقط فیلم‌های YPbPr و آنالوگ RGB می‌توانند ویدیو و وضوح بالاتر از i480 یا i576، تا p1080 برای YPbPr را با هم تداخل نداشته باشند.

5.2.1

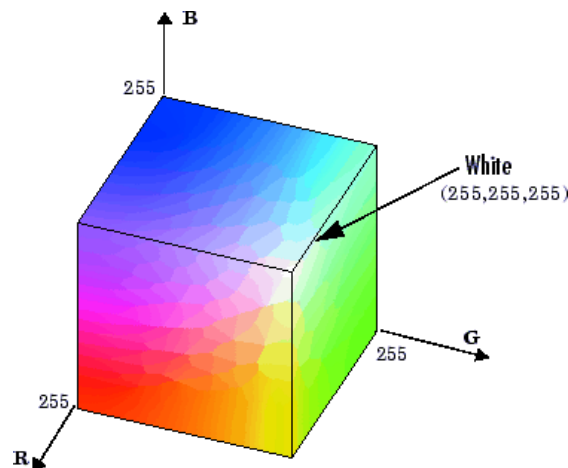
کاهش تعداد رنگها در یک تصویر، کوانتیزه شدن است.



بعد از اینکه مکعب رنگی تقسیم شد، همه جعبه های خالی به بیرون پرتاب می شوند. بنابراین، فقط یکی از جعبه ها برای تولید رنگ برای نقشه رنگ استفاده می شود.

5.2.3

به طور کلی از الگوریتم های خوشه بندی برای گروه بندی نقاط داده مشابه استفاده می شود. به نقاط داده ای که مشابه هستند مقداری اختصاص داده می شود که نشان دهنده مقدار متوسط همه نقاط موجود در آن خوشه است. اگر نقاط داده اضافی جمع آوری شود، می توان آن ها را با مقادیر متوسط خوشه های دیگر مقایسه کرد و به نزدیکترین آن ها اختصاص داد. K-mean یک الگوریتم تکرار شونده است که خوشه بندی را با شروع با مکان های سانرویید به صورت تصادفی اختصاص داده شده به عنوان مرکز هر خوشه، پیاده سازی می کند. در هر تکرار، نزدیکترین نقاط داده به هر مرکز تیروئید تعیین شده و خطای کل با جمع کردن فاصله کل از هر نقطه به مرکز متناظر آن محاسبه می شود. سپس، سانروییدها تنظیم می شوند تا زمانی که نماینده هر خوشه باشند و خطای کل به حداقل برسد. در زیر تصویری از این فرآیند برگرفته از یک



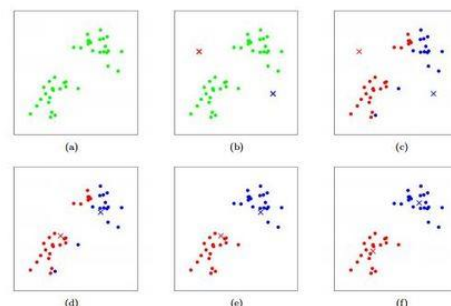
کوانتیزه سازی شامل تقسیم مکعب رنگی RGB به تعداد جعبه های کوچکتر و سپس نقشه برداری از تمام رنگ های موجود در هر جعبه به مقدار رنگ در مرکز آن جعبه است.

minimum و Uniform quantization

variance quantization در روشی که برای تقسیم مکعب رنگی RGB استفاده می شود متفاوت اند. با Uniform quantization، مکعب رنگی به جعبه هایی با اندازه یکسان (مکعب های کوچکتر) برش داده می شود. با کمترین مقدار سنجش واریانس، مکعب رنگی به جعبه هایی تقسیم می شود (نه لزوماً مکعب) در اندازه های مختلف. اندازه جعبه ها به نحوه توزیع رنگ ها در تصویر بستگی دارد.

شکل زیر کمی سازی یکنواخت تصویر uint8 را نشان می دهد. برای درک راحت تر، شکل یک برش دو بعدی (یا صفحه رنگی) از مکعب رنگ را نشان می دهد که در آن قرمز = 0 و سبز و آبی از 0 تا 255 است. مقادیر واقعی پیکسل با مراکز x نشان داده می شوند.

مقاله آنالیز برای دوره استنفورد در مورد هوش مصنوعی آورده شده است.



در زمینه پردازش تصویر، یک مشکل رایج در تعیین نحوه نمایش یک تصویر رنگی در دستگاهی است که فقط می تواند تعداد محدودی از رنگها را نمایش دهد بدون اینکه کیفیت تصویر زیادی را از بین ببرد.

تصویر رنگی به طور معمول به عنوان سه ماتریس موازی ذخیره می شود که هر ماتریس نشان دهنده اجزای قرمز، سبز و آبی تصویر است. هر مولفه می تواند از 0 تا 255 باشد، به این معنی که 3×256 رنگ را می توان نشان داد. از آنجا که چشم انسان نمی تواند تقریباً بسیاری از رنگهای منحصر به فرد را تشخیص دهد، بنابراین انتخاب مقدار محدودی از رنگ برای نشان دادن یک تصویر رنگی منطقی است.

با استفاده از k-means می توان ترکیب رنگها را به تعداد مشخصی از سطوح اندازه گیری کرد. این به خوبی کار می کند زیرا چشم انسان نمی تواند طیف رنگی کامل را درک کند. در زمینه k-means، این سطوح رنگی کوانتیزه شده، مرکز سانترال خواهد بود. برای هر پیکسل، نزدیکترین مرکز مرکز با در نظر گرفتن هر پیکسل به عنوان بردار $\langle r, g, b \rangle$ و استفاده از فرمول فاصله برای یافتن فاصله بین پیکسل و هر مرکز مرکز تعیین می شود. الگوریتم به هر مرکز مرکز یک مقدار رنگ اختصاص می دهد که

میانگین تمام پیکسل های نزدیک به آن مرکز را نشان می دهد.

تصاویر زیر نتیجه استفاده از k-means برای اندازه گیری یک تصویر رنگی را نشان می دهد. تصویری که با 256 سطح کمی شده است تقریباً از تصویر اصلی قابل تشخیص نیست.



تصویر اصلی



$K = 2$



$K = 20$

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 255 \\ 255 \\ 255 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

در فرمول های زیر فرض شده است که:

$$R, G, B, Y \in [0, 1], \quad I \in [-0.5957, 0.5957], \quad Q \in [-0.5226, 0.5226]$$

From RGB to YIQ [\[edit\]](#)

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \approx \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5959 & -0.2746 & -0.3213 \\ 0.2115 & -0.5227 & 0.3112 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

From YIQ to RGB [\[edit\]](#)

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0.956 & 0.619 \\ 1 & -0.272 & -0.647 \\ 1 & -1.106 & 1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

Note that the top row is identical to that of the YUV color space

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

5.1.1 تصویر ورودی:



K = 256

3-شکل ها، جدول ها و روابط (فرمول ها)

5.1:

تبدیل رنگها از RGB به HSI.

$$H = \begin{cases} \theta & \text{اگر } B \leq G \\ 360 - \theta & \text{اگر } B > G \end{cases}$$

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right\}$$

$$S = 1 - \frac{3}{(R + G + B)} [\min(R, G, B)]$$

$$I = \frac{1}{3}(R + G + B)$$

تبدیل رنگها از HSI به RGB:

$$B = I(1 - S)$$

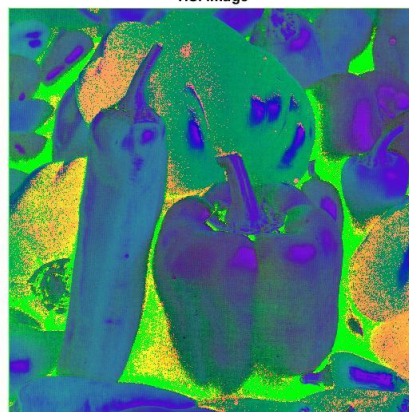
$$R = I \left[1 + \frac{S \cos H}{\cos(60^\circ - H)} \right]$$

$$G = 3I - (R + B)$$

تبدیل به CMY:

تبدیل به HSI:

HSI Image



S



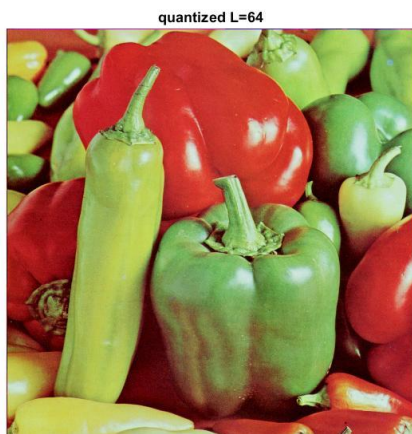
I



H



PSNR = 39.14



MSE = 7.32
PSNR = 39.49

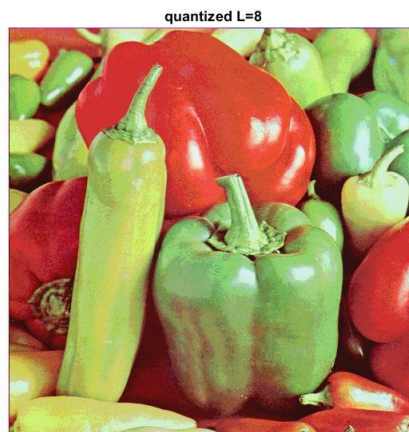
همان طور که انتظار داشتیم با بالاتر رفتن تعداد سطوح، کیفیت تصویر بهبود می یابد اما چشم انسان نسبت به حالتی که همین عمل را با تصویر grayscale انجام بدهیم، تغییر کمتری حس می کند.

5.2.2

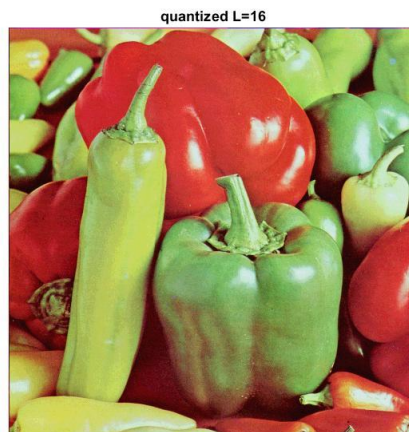


MSE = 95.73
PSNR = 18.56

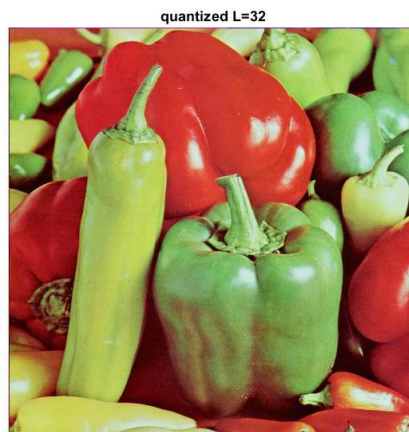
5.2.1



MSE = 347.29
PSNR = 22.72



MSE = 92.25
PSNR = 28.49



MSE = 25.05



L=16



L=32

4- کدها:

5.1.1

```
function rgbtohsi (F)

F=im2double (F);
r=F(:, :, 1);
g=F(:, :, 2);
b=F(:, :, 3);
th=acos ((0.5 * ((r-g) + (r-b))) ./ ((sqrt((r-g).^2 + (r-b).^2 * (g-b))) + eps));
H=th;
H (b>g)=2*pi-H (b>g);
H=H / (2*pi);
S=1-3.* (min (min (r,g), b)) ./ (r+g+b+eps);
I=(r+g+b) / 3;
hsi=cat (3, H, S, I);

figure, imshow (H), title ('H');
figure, imshow (S), title ('S');
figure, imshow (I), title ('I');
figure, imshow (hsi), title ('HSI Image');

end
```

همان‌طور که انتظار داشتیم، رنگ آبی کمتر شده است و دو رنگ دیگر به نسبت رنگ آبی بیشتر شده اند و به طور کلی رنگ سفید نیز کاهش یافته است. به طور کلی ترکیب رنگ تصویر دچار تغییر شده است.

5.2.3

تصویر ورودی:



L=8



5.2.3

```
function [y] = kmeans(x,K)
MAX_IT = 10; %max number of iterations
ep = 10; %max error allowed in solution
iteration = 0;
errorsum = 11;
[M,N,c] = size(x);
labels = zeros(M,N);
centroids = uint8(zeros(1,K,3));
centroids(1,1) = uint8(linspace(0,256,K));
centroids(1,2) = uint8(linspace(0,256,K));
centroids(1,3) = uint8(linspace(0,256,K));
y = zeros(M,N,c);
x = double(x);
sums = zeros(1,K,3);
counts = zeros(1,K);
while iteration <= MAX_IT && errorsum > ep
    errorsum = 0;
    for m = 1:M
        for n = 1:N
            smallest_dist = 500;
            closestk = 1;
            for k = 1:K
                dist = sqrt((x(m,n,1) - double(centroids(1,k,1)))^2 + (x(m,n,2) - double(centroids(1,k,2)))^2 + (x(m,n,3) - double(centroids(1,k,3)))^2);
                if dist < smallest_dist
                    smallest_dist = dist;
                    closestk = k;
                end
            end
            errorsum = errorsum + smallest_dist; %update total error
            labels(m,n) = closestk;
            counts(closestk) = counts(closestk) + 1; %update new label count
            sums(1,closestk,1) = sums(1,closestk,1) + x(m,n,1);
            sums(1,closestk,2) = sums(1,closestk,2) + x(m,n,2);
            sums(1,closestk,3) = sums(1,closestk,3) + x(m,n,3);

            y(m,n,1) = centroids(1,closestk,1);
            y(m,n,2) = centroids(1,closestk,2);
            y(m,n,3) = centroids(1,closestk,3);
        end %column
    end %row
    for k = 1:K
        if counts(k) > 0
            centroids(1,k,1) = sums(1,k,1) / counts(k);
            centroids(1,k,2) = sums(1,k,2) / counts(k);
            centroids(1,k,3) = sums(1,k,3) / counts(k); %update centroid values
        else
            centroids(1,k,1) = centroids(1,k,1) / 10;
            centroids(1,k,2) = centroids(1,k,2) / 2;
            centroids(1,k,3) = centroids(1,k,3) / 3; %randomly reassign
        end
    end %change location of centroid k
    iteration = iteration + 1;
    errorsum = errorsum / (M*N);
end %one iteration of the k-mean process
imshow(uint8(y))
end
```

5.2.1

```
function [out] = uniform_quantize(img,L)
Red = img(:,:,1);
Green = img(:,:,2);
blue = img(:,:,3);
q = 256/L;
qr = quantize(Red,q);
qg = quantize(Green,q);
qb = quantize(blue,q);
out = cat(3,qr,qg,qb);
figure;
imshow(out,[]),title("quantized L=64");
end
```

```
function [out1] = quantize(f,q)
[n,m] = size(f);
for i=1:n
    for j=1:m
        if(f(i,j)>0 && f(i,j)<255)
            out1(i,j) = floor(f(i,j)/q)*q+q/2;
        elseif(f(i,j)<=0)
            out1(i,j) = q/2;
        elseif(f(i,j)>=255)
            out1(i,j) = 255 - q/2;
        end
    end
end
end
```

5.2.2

```
function [out] = quantize2(img)
Red = img(:,:,1);
Green = img(:,:,2);
blue = img(:,:,3);
qr = 256/6;
qg = 256/6;
qb = 256/4;
qr1 = quantize(Red,qr);
qg1 = quantize(Green,qg);
qb1 = quantize(blue,qb);
out = cat(3,qr1,qg1,qb1);
figure;
imshow(out,[]),title("quantized");
end
```

```
function [out1] = quantize(f,q)
[n,m] = size(f);
for i=1:n
    for j=1:m
        if(f(i,j)>0 && f(i,j)<255)
            out1(i,j) = floor(f(i,j)/q)*q+q/2;
        elseif(f(i,j)<=0)
            out1(i,j) = q/2;
        elseif(f(i,j)>=255)
            out1(i,j) = 255 - q/2;
        end
    end
end
end
end
```

مراجع

- <https://onlinebme.com/unit/rgb/>

- <https://en.wikipedia.org/wiki/XvYCC>
- <https://en.wikipedia.org/wiki/YIQ>
- <https://en.wikipedia.org/wiki/SRGB>
- <https://en.wikipedia.org/wiki/YPbPr>
- <https://kgut.ac.ir/useruploads/1554342785847rqc.pdf>
- https://www.projectrhea.org/rhea/index.php/Quantization_and_Classification_using_K-Means_Clustering