

# بینایی کامپیوتر

فصل دوم: تصاویر و ویژگیهای آنها

محمد داورپناه جزی

ترم اول ۱۴۰۵-۱۴۰۴

گروه مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی اصفهان

# روش مدیریت درس

## ضوابط ارزیابی درس

- تکالیف درسی ۱۰٪
- مقاله + ارائه در کلاس ۳۰٪
- امتحان میان ترم ۳۰٪ + جایزه
- امتحان پایان ترم ۳۰٪ + جایزه
- روز حساب دارای نمره اضافی به عنوان جایزه، ارائه اختیاری در کلاس دارای نمره اضافی به عنوان جایزه
- امکان تغییر درصدها تا ۵ درصد با توجه به روند برگزاری جلسات درس

## مطالب درسی

کلیه اطلاعات درس شامل اسلایدهای جلسات درس، فیلم جلسات، مراجع، تکالیف و آرایه‌ها همه روی سامانه دانشگاه در دسترس شماست، تحویل کلیه تکالیف و پروژه‌ها از طریق سامانه دانشگاه انجام می‌شود

## ارتباط با استاد

لازم است کلیه دانشجویان در گروه واتساپ تحت عنوان گروه درسی بینایی کامپیوتر عضو شوند  
کلیه ارتباطها با استاد از طریق این گروه خواهد بود

وب سایت استاد: [davarpanahjazi.iut.ac.ir](mailto:davarpanahjazi.iut.ac.ir) ایمیل استاد: [mdjazi7192@gmail.com](mailto:mdjazi7192@gmail.com)

# روش مدیریت درس

## مراجع فارسی درس

- ۱- مبانی بینایی ماشین و پردازش تصویر، نوشته محمد علی بالافر، نشر آیلار تهران
- ۲- پردازش دیجیتالی تصاویر (ویراست سوم)، نوشته گونزالس آر سی و وودز آر ای، ترجمه مجتبی لطفی، سعید میرقاسمی و امیرمسعود عمومی، نشر نیاز دانش تهران

## مراجع لاتین درس

- 1- A practical Introduction to Computer vision with OPENCV. Dawson-Howe K.  
John Wiley & Sons Ltd.
- 2- Computer vision a modern approach (2<sup>nd</sup> Edition). Forsyth D A and Ponce J.  
Prentice Hall.
- 3- Computer Vision: Algorithms and Applications (2<sup>nd</sup> Edition). Richard Szeliski  
September 2021.

## و دریای بیکران اینترنت

از جمله سایت آرمایشگاه بینایی کامپیوتر آقای دکتر صفابخش در دانشگاه امیرکبیر  
فایل مراجع لاتین در سامانه قرار داده شده و قابل دانلود است

# سرفصل مطالب درس

- فصل اول: تاریخچه و مقدمه
- فصل دوم: تصاویر و ویژگیهای آنها
- فصل سوم: هیستوگرامها و خوشه بندی
- فصل چهارم: دید باینری
- فصل پنجم: تبدیلات هندسی
- فصل ششم: لبه ها
- فصل هفتم: فیچرها
- فصل هشتم: تشخیص
- فصل نهم: ویدئو
- فصل دهم: یادگیری عمیق و بینایی کامپیوتر
- فصل یازدهم: اصول تشکیل تصویر
- فصل دوازدهم: پرسپکتیو دینامیک
- فصل سیزدهم: پروژه های بینایی کامپیوتر

# فصل دوم: تصویرها و ویژگیهای آنها

۲-۱ دوربینها

۲-۲ تصویر

۲-۳ تصویرهای رنگی

۲-۴ نویزها

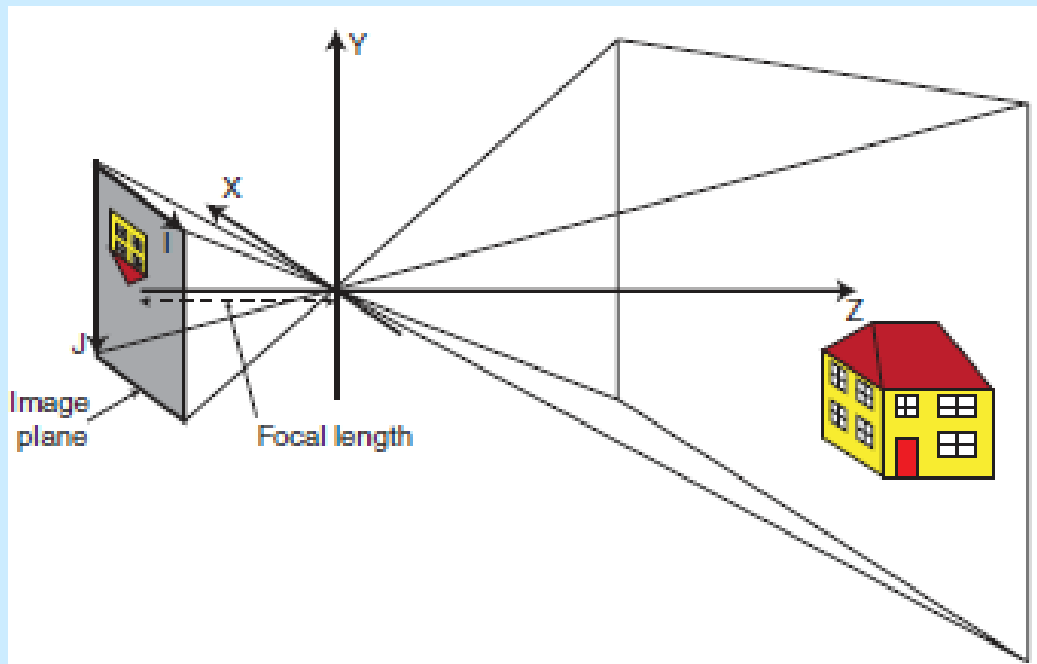
۲-۵ هموارسازی

# فصل دوم: تصویرها و ویژ گیهای آنها

## ۱-۲ دوربینها

تصویرها: نموده‌های چشمی یک صحنه توسط دستگاه تصویربرداری به عنوان قلب تپنده بینایی کامپیوتر پردازش تصویر: روشهای تجزیه و تحلیل تصویر برای استخراج ویژ گیهای مرتبط با آن و همچنین حذف نویزهای موجود در آن

دوربین: متمرکز کننده پرتوهای نور به شیوه کنترل شده از طریق لنزها بر روی صفحه حساس (فیلم) و همچنین جلوگیری از تابیدن نورهای هرز بر روی صفحه مزبور



دوربین ساده پین هول: تابیده شدن پرتوهای نور از طریق یک سوراخ ریز (به جای لنز) به صفحه حساس

ساده ترین سیستم تصویربرداری تبدیل واقعیت سه بعدی با پین هول به عنوان مبدا مختصات آن بر روی صفحه حساس دوبعدی

پیشگیری از ورود نورهای هرز با قرار دادن صفحه حساس در یک بدنه ایزوله

شکل ۱-۲: مدل ساده دوربین پین هول

# فصل دوم: تصویرها و ویژ گیهای آنها

## دوربینها (ادامه)

تبدیل مختصات سه بعدی  $(x, y, z)$  به معادل دوبعدی  $(i, j)$  در دوربین ساده پین هول مطابق رابطه زیر

$$\begin{bmatrix} i.w \\ j.w \\ w \end{bmatrix} = \begin{bmatrix} f_i & 0 & c_i \\ 0 & f_j & c_j \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$f_i$  نشاندهنده فاصله کانونی

$f_j$  اندازه پیکسلها در جهت  $i$  و  $j$

$(c_i, c_j)$  مختصات محل تقاطع محور اپتیکی با صفحه حساس یا مرکز اپتیکی

محور اپتیکی خط عمود بر صفحه حساس از نقطه پین هول

$w$  ضریب مقیاس در مختصات همگن

# فصل دوم: تصویرها و ویژگیهای آنها

## ۲-۲ تصویر

تصویر از دید فنی: یک پروجکشن یا بازتاب دوبعدی از یک صحنه سه بعدی  
نتیجه یک عکس ثبت شده توسط یک سنسور (دوربین)

یک تابع پیوسته شامل دو مختصات معمولاً در قالب  $(i, j)$  یا (سطر و ستون) بر روی صفحه حساس

تصویر دیجیتال: نمونه برداری و نمایش نقاط در قالب یک ماتریس شامل  $M$  سطر و  $N$  ستون  
ثبت هر درایه ماتریس به صورت عدد صحیح (کوانتیزه شده)  
تقسیم محدوده پیوسته به  $k$  فاصله مساوی (معمولاً ۲۵۶)

انواع تصاویر دیجیتال

تصویر باینری (سیاه و سفید): استفاده از تنها دو رنگ و تشکیل پیکسل‌های سیاه و سفید

تصویر در مقیاس خاکستری: استفاده از سایه روشن خاکستری برای نمایش پیکسلها در طیف خاکستری

تصویر رنگی: به کارگیری اطلاعات رنگ برای نمایش هر پیکسل از تصویر

تصویر چند طیفی: استفاده از داده‌های تصویر در محدوده الکترومغناطیسی با طول موج خاص و ثبت تصویر

محاسبه در OpenCV: تعریف و استفاده از ساختمان داده‌ای از نوع Mat برای ثبت تصویر

استفاده از تابع `at` برای تعریف مقادیر متمایز پیکسل

ذخیره مقدار هر پیکسل در نوع تعریف شده `CV_8UC1` (تصویر یک کاناله هشت بیت بدون علامت)

قرار دادن مقدار صفر در بیت‌های اضافی هر پیکسل برای تصویر در مقیاس خاکستری

توجه به متن برنامه در اسلاید بعدی



# فصل دوم: تصویرها و ویژگیهای آنها

تصویر (ادامه)

در OpenCV: متن برنامه نمایش تصویر به صورت زیر

```
void ChangeQuantisationGrey( Mat &image, int num_bits )
{
    CV_Assert( (image.type() == CV_8UC1) &&
        (num_bits >= 1) && (num_bits <= 8) );
    uchar mask = 0xFF << (8-num_bits);
    for (int row=0; row < image.rows; row++)
        for (int col=0; col < image.cols; col++)
            image.at<uchar>(row,col) =
                image.at<uchar>(row,col) & mask;
}
```

# فصل دوم: تصویرها و ویژگیهای آنها

## نمونه برداری (Sampling)

مفهوم: تبدیل یک تصویر مستمر به معادل آن در قالب عناصر گسسته

تصویر دیجیتال: یک آرایه دوبعدی شامل تعدادی عنصر یا نقطه تصویر (پیکسل یا نقطه نمونه برداری شده)

هر نقطه نمایشگر یک ناحیه تثبیت شده بر روی صفحه حساس

وجود مرز کوچک بین این نقاط بدون حساسیت به نور

احتمال خیلی کم نادیده گرفتن برخی نقاط موجود در این نواحی

پیکسل: نماینده مقدار میانگین روشنایی و رنگ یک ناحیه از تصویر بازتاب شده از شیئی در دنیای واقعی

در حالت خاص وجود چند شیئی مرز، جمع بندی نور منعکس شده از چندین شیئی

تعداد نمونه: امکان محدود شدن شیئیهای قابل نمایش در صورت کافی نبودن تعداد نمونهها

لZoom کافی بودن تعداد نمونهها برای امکان تفکیک پذیری اجزای تصویر

نیاز به زمان بیشتر برای پردازش و تاخیر در رسیدن به نتیجه در صورت بیش از حد بودن تعداد نمونه

لZoom سازش بین میزان تفکیک پذیری و تعداد نمونهها

مثال: تشخیص حضور مردم فقط در تصویر بالای شکل ۲-۲ اسلاید بعدی به دلیل بالاتر بودن تعداد نمونهها

در OpenCV: تغییر میزان نمونه برداری با تابع `resize`

```
resize( image, smaller_image, Size( image.cols/2, image.rows/2 ));
```

# فصل دوم: تصویرها و ویژ گیهای آنها

نمونه برداری (ادامه)

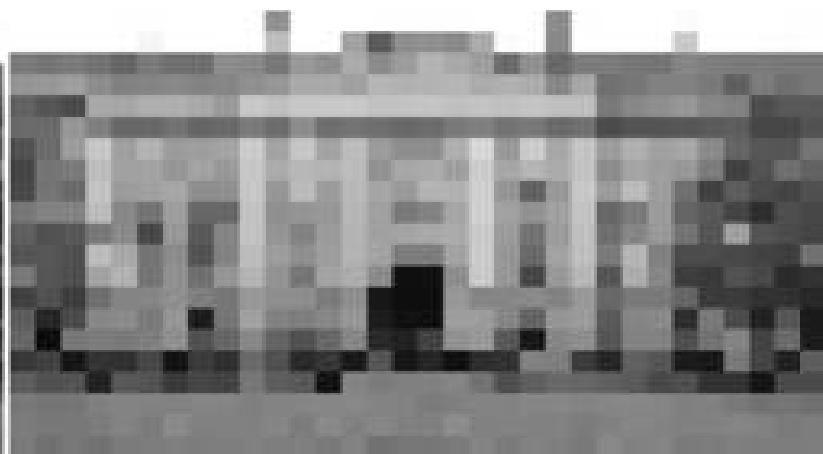
۱۲۸\*۹۶



۲۵۶\*۱۹۲



۳۲\*۲۴



۶۴\*۴۸



شکل ۲-۲: یک تصویر با چهار نمونه برداری مختلف

# فصل دوم: تصویرها و ویژگیهای آنها

## کمی شدن یا کوانتیزه شدن (Quantisation)

مفهوم: تبدیل مقادیر پیوسته ویژگیهای تصویر مستمر به معادل آن در فرم گسسته

تعریف ریاضی پیکسل:  $f(i, j)$  تابعی از مقادیر روشنایی صحنه

پیوسته بودن مقادیر روشنایی صحنه و لزوم تبدیل آن به فرم گسسته برای تصویر دیجیتال

تعداد سطوح روشنایی هر کانال برابر  $k = 2^b$  که  $b$  تعداد بیتها (معمولا ۸) را نشان می دهد

مقدار مناسب  $b$ : سوال مهم، تصمیم گیری برای تعداد بیت کافی برای نمایش پیکسلها

تعداد زیاد بیت منجر به نیاز به حجم زیاد حافظه

تعداد کم بیت منجر به محو شدن اطلاعات دقیق موجود در تصویر

لزوم کافی بودن تعداد بیتها جهت نمایش هر پیکسل برای ذخیره تصویر مناسب هر کاربرد

لزوم سازش بین واضح بودن تصویر و تعداد بیتها

مثال: تفاوت اندک بین تصاویر ۸ بیتی و ۶ بیتی، در بالای شکل ۲-۳ اسلاید بعدی با وجود ۲۵٪ بیت کمتر

به همین ترتیب شباهت تصاویر چهار بیتی و دو بیتی در پایین تصویر

نتیجه: لزوم تعیین تعداد بیت مورد نیاز با در نظر گرفتن هدف تصویربرداری

نیاز به کوانتیزاسیون بالاتر در صورت تفسیر خودکار صحنه توسط ماشین، نسبت به آنچه انتظار میرود

بروز خطوط ناخواسته مشابه پایین شکل ۲-۳ در تصاویر با تعداد بیت دو یا چهار

در OpenCV: تعیین تعداد بیت با ماسک کردن تعداد بیت اضافی، مشابه نمونه برنامه در اسلاید ۹

تغییر تعداد بیت در یک تصویر مثلا از ۳۲ بیت به ۱۶ بیت یا ۸ بیت

# فصل دوم: تصویرها و ویژ گیهای آنها

کمی شدن (ادامه) پیکسلهای ۶ بیتی



پیکسلهای ۸ بیتی



پیکسلهای ۲ بیتی



پیکسلهای ۴ بیتی



شکل ۲-۳: چهار حالت مختلف کوانتیزه شدن یک تصویر مقیاس خاکستری

# فصل دوم: تصویرها و ویژگیهای آنها

## ۲-۳ تصویرهای رنگی

تصویر مقیاس خاکستری (Monochromic/Greyscale): دارای یک کانال داده (اشتباه بنام سیاه و سفید) بیان میزان روشنایی (y) در هر نقطه (پیکسل) از یک صحنه تصویر

تصویر رنگی یا چندطیفی (Multispectral): دارای چند کانال داده برای نمایش میزان روشنایی و اطلاعات رنگ (رنگ تابی یا Chrominance) از درون صحنه

بزرگتر و بسیار پیچیده تر بودن تصویرهای رنگی با نمونه برداری و کوانتیزه یکسان ارائه اکثر شیوه های بینای کامپیوتر برای تصویرهای مقیاس خاکستری، مناسب نبودن برای رنگی

دلایل تمرکز الگوریتمهای بینایی کامپیوتر بر تصویرهای مقیاس خاکستری

۱- سهولت درک تصویرهای مقیاس خاکستری توسط انسان

۲- عدم نیاز به تحمل دردسر پیچیدگی تصویرهای رنگی توسط انسان

۳- محدودتر و کم حجم تر بودن تصویرهای مقیاس خاکستری

دلایل توجه به تصویرهای رنگی ترجیح دادن آنها به تصویرهای مقیاس خاکستری

آسان تر شدن انجام خیلی از پردازشها به دلیل ارائه اطلاعات بیشتری توسط فاکتور رنگ در تصویر

امکان قسمت بندی کردن تصویر به موجودیتهای مجزای فیزیکی مثل اشیاء و سطوح مختلف

مثال: ساده تر بودن تشخیص درختهای مجزا و مختلف در تصویر رنگی شکل ۲-۴ اسلاید بعدی

# فصل دوم: تصویرها و ویژگیهای آنها

## تصویرهای رنگی (ادامه)

امکان تشخیص درختهای مجزا و مختلف در تصویر رنگی در مقایسه با تصویر مقیاس خاکستری



شکل ۲-۴: تصویر رنگی RGB سمت چپ در مقایسه با تصویر مقیاس خاکستری سمت راست حساس بودن چشم انسان به طول موجهای بین ۴۰۰ نانومتر تا ۷۰۰ نانومتر ساخت دوربینهای تصویربرداری بر مبنای همین حساسیتهای مربوط به انسان

# فصل دوم: تصویرها و ویژگیهای آنها

## تصویرهای Red-Green-Blue یا RGB

ویژگی: پیچیده تر بودن قابل توجه داده های تصاویر رنگی در مقایسه با تصاویر مقیاس خاکستری

رایج ترین روش: به کارگیری سه کانال داده برای سه رنگ اصلی قرمز ، سبز و آبی

قرمز با طول موج ۷۰۰ نانومتر، سبز با طول موج ۵۴۶،۱ نانومتر و آبی با طول موج ۴۳۵،۸ نانومتر

حساس بودن عناصر تاثیرپذیر از نور در دوربینها از نظر طیفی به طول موجهای متمرکز روی این رنگها

نمایش تصویر RGB و کانالهای سه گانه آن به صورت مجزا در شکل ۲-۵ اسلاید بعدی

نمایش منحنی های حساسیت طیفی برای عناصر حساس به سه نور اصلی در شکل ۲-۶ اسلاید بعدی

در OpenCV: نمایش تصویرهای رنگی با استفاده از ساختمان داده Mat (اسلاید ۹)

تعریف سه کانال برای هر پیکسل به جای یک کانال ( آرایه سه تایی با ۸ بیت بدون علامت در هر درایه)

استفاده از تابع at برای دسترسی به مقادیر مجزای هر کانال

مثال: نمایش معکوس شده یک تصویر در تابع Invertcolour در اسلاید ۱۸

توجه به سه حلقه تودرتو به شرح زیر

حلقه بیرونی با اندیس row برای حرکت روی سطرها

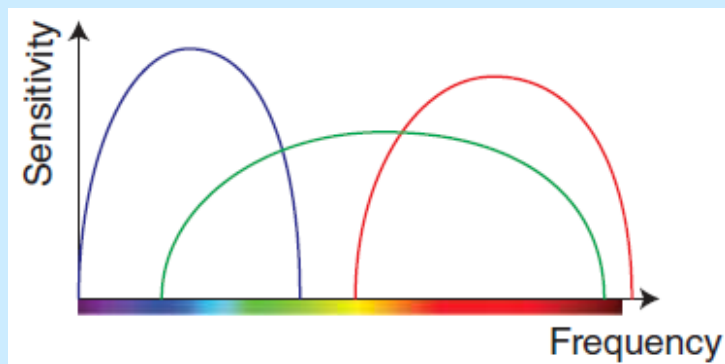
حلقه وسطی با اندیس col برای حرکت روی ستونها

و نهایتا حلقه داخلی با اندیس channel برای حرکت روی کانالهای رنگ



# فصل دوم: تصویرها و ویژ گیهای آنها

## تصویرهای RGB (ادامه)



شکل ۲-۶: منحنی‌های حساسیت طیفی  
برای عناصر حساس به سه نور اصلی



شکل ۲-۵: تصویر رنگی RGB بالا سمت چپ، کانال قرمز  
بالا سمت راست، کانال سبز پایین سمت چپ و کانال آبی  
پایین سمت راست

# فصل دوم: تصویرها و ویژگیهای آنها

## تصویرهای RGB (ادامه)

تابع نمایش معکوس یک تصویر رنگی

```
void InvertColour( Mat& input_image, Mat& output_image )
{
    CV_Assert( input_image.type() == CV_8UC3 );
    output_image = input_image.clone();
    for (int row=0; row < input_image.rows; row++)
        for (int col=0; col < input_image.cols; col++)
            for (int channel=0; channel < input_image.channels(); channel++)
                output_image.at<Vec3b>(row,col)[channel] = 255 -
                    input_image.at<Vec3b>(row,col)[channel];
}
```

نحوه نمایش: ارائه ترکیبی از داده‌های این سه کانال بر روی صفحه نمایش برای دیدن کاربر  
وجود تعداد  $16777216 = 256 * 256 * 256$  رنگ مختلف ولی عدم امکان ارائه تمام این رنگها  
نوع CV\_8UC3 به معنی تصویر سه کاناله هر کانال مقدار ۸ بیتی بدون علامت

# فصل دوم: تصویرها و ویژ گیهای آنها

## تصویرهای RGB (ادامه)

تبدیل تصویر رنگی به مقیاس خاکستری: استفاده از رابطه زیر برای تبدیل داده های تصویر رنگی RGB به معادل آن در مقیاس خاکستری

$$Y = 0.299R + 0.587G + 0.114B$$

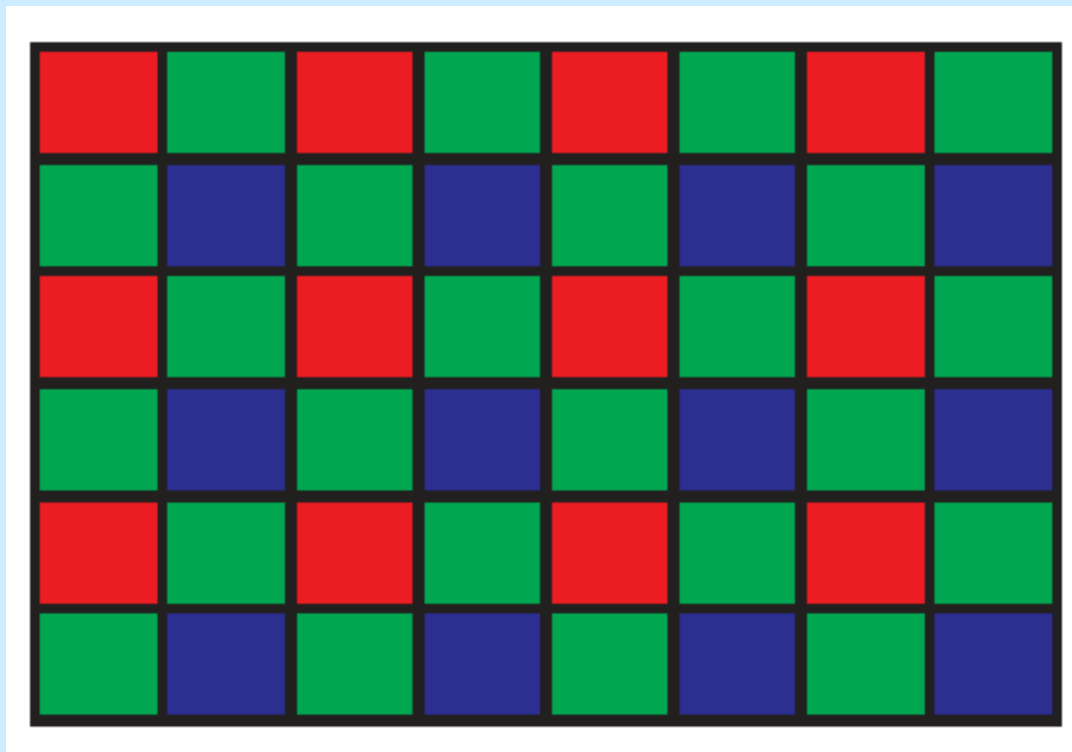
ساختار صفحه حساس دوربین: قرار نداشتن اجزای حساس به سه طول موج نور متفاوت روی همدیگر تشکیل یک الگوی منظم مانند شکل ۲-۷ اسلاید بعدی معروف به الگوی بایر (Bayer) خانه های قرمز، سبز و آبی نشان دهنده سلولهای حساس به نورهای به ترتیب قرمز، سبز و آبی درون یابی مقادیر حس شده توسط سنسور برای محاسبه میزان RGB در هر پیکسل نتیجه مطابق نبودن مقادیر حاصله با نمونه برداری دقیق از تصویر پیوسته به کار گیری این الگو در دوربینهای مدرن CCD و دوربینهای قدیمی CMOS

در OpenCV: توجه به نمونه برنامه های ارائه شده در اسلاید ۲۱ تبدیل نمایش رنگی به مقیاس خاکستری با استفاده از تابع `cvtColor` تبدیل تصویر رنگی به مولفه های کانال با استفاده از تابع `split` استفاده از اشاره گر ها به جای استفاده مستقیم از تابع `at` به دلیل بالا بودن حجم محاسبات و کندی برنامه بالا بردن سرعت با به کار گیری اشاره گر ها علیرغم پیچیدگی کد نوشته شده (توصیه به مطالعه کتاب C)

# فصل دوم: تصویرها و ویژ گیهای آنها

## تصویرهای RGB (ادامه)

ثبت تصویر رنگی با تشکیل یک الگوی منظم مانند شکل زیر معروف به الگوی بایر (Bayer) خانه های قرمز، سبز و آبی نشان دهنده سلولهای حساس به نورهای به ترتیب قرمز، سبز و آبی



شکل ۲-۷: الگوی منظم تشکیل سلولهای حساس به نورهای مختلف معروف به الگوی بایر (Bayer)

# فصل دوم: تصویرها و ویژ گیهای آنها

## تصویرهای RGB (ادامه)

نمونه برنامه اول در محیط OpevCV: انجام تبدیل بین نمایشهای مختلف تصویر  
تبدیل نمایش رنگی به نمایش مقیاس خاکستری در جمله دوم  
تبدیل تصویر رنگی به مولفه های سه گانه کانالهای قرمز، سبز و آبی در جمله چهارم  
جداسازی کانال آبی از کل داده ها در جمله آخر  
توجه به استفاده از حروف BGR به جای RGB در تعریف نوعها در محیط OpenCV

```
Mat bgr_image, grey_image;  
cvtColor(bgr_image, grey_image, CV_BGR2GRAY);  
vector<Mat> bgr_images(3);  
split(bgr_image, bgr_images);  
Mat& blue_image = bgr_images[0];
```

# فصل دوم: تصویرها و ویژگیهای آنها

## تصویرهای RGB (ادامه)

نمونه برنامه دوم در محیط OpevCV: به کارگیری اشاره گرها به جای پردازش مستقیم داده ها  
معکوس کردن تصویر سه کاناله با استفاده از اشاره گرها به جای استفاده مستقیم از تابع `at`  
نتیجه، پایین آوردن حجم محاسبات و افزایش سرعت برنامه (اشاره به پروژه رادار)

```
int image_rows = image.rows;
int image_columns = image.cols;
for (int row=0; row < image_rows; row++) {
    uchar* value = image.ptr<uchar>(row);
    uchar* result_value = result_image.ptr<uchar>(row);
    for (int column=0; column < image_columns; column++)
    {
        *result_value++ = *value++ ^ 0xFF;
        *result_value++ = *value++ ^ 0xFF;
        *result_value++ = *value++ ^ 0xFF;
    }
}
```

# فصل دوم: تصویرها و ویژگیهای آنها

## تصویرهای Cyan–Magenta–Yellow یا CMY

روش RGB: به کارگیری سه رنگ اصلی قرمز، سبز و آبی برای ثبت سه کانال داده تصویر رنگی  
مدل رنگ CYM: ثبت تصویر با به کارگیری سه رنگ فرعی فیروزه‌ای (C)، ارغوانی (M) و زرد (Y)  
در قالب سه کانال داده

مقادیر کانالهای این سه رنگ برابر حاصل تفریق سه رنگ اصلی از سفید کامل به صورت زیر

$$C = 255 - R, M = 255 - G, Y = 255 - B$$

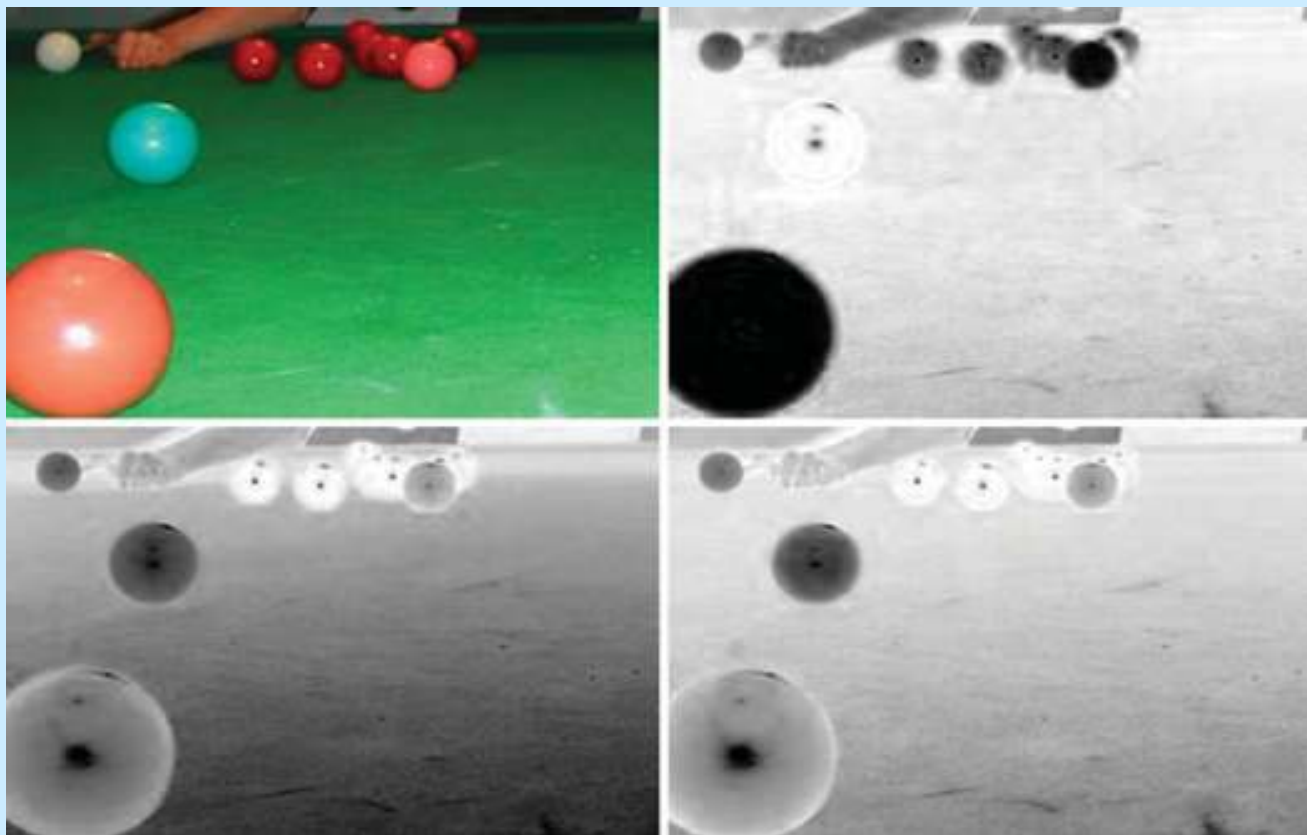
استفاده از این روش در برخی مدل‌های پرینترهای رنگی

در OpenCV: روش تبدیل RGB به CYM با استفاده از تابع `Invertcolour` در اسلاید ۱۸ و معادل آن با به کارگیری اشاره‌گرها در اسلاید ۲۲

تصویرها: نمایش تصویر CYM و کانالهای سه‌گانه آن به صورت مجزا در شکل ۲-۸ اسلاید بعدی

# فصل دوم: تصویرها و ویژ گیهای آنها

## تصویرهای CYM (ادامه)



شکل ۲-۸: تصویر رنگی CYM بالا سمت چپ، کانال زرد بالا سمت راست،  
کانال ارغوانی پایین سمت چپ و کانال فیروزه‌ای پایین سمت راست



# فصل دوم: تصویرها و ویژ گیهای آنها

## تصویرهای YUV

مدل رنگ YUV: ثبت تصویر با به کارگیری فاکتور روشنایی یا درخشندگی (Y) و دو رنگ تکمیلی، رنگ U برابر آبی منهای روشنایی (Y) و رنگ V برابر قرمز منهای روشنایی (Y) محاسبه از RGB: نحوه محاسبه مقادیر روشنایی (Y)، رنگ (U) و رنگ (V) به صورت زیر

$$\begin{aligned}Y &= 0.299R + 0.587G + 0.114B \\U &= 0.492*(B - Y) \\V &= 0.877*(R - Y)\end{aligned}$$

کاربرد: استفاده از این روش در سیگنالهای تلویزیونیهای آنالوگ مثل PAL و NTSC دلیل، حساسیت بیشتر سیستم بینایی انسان به روشنایی (Luminance) در مقایسه با رنگ تابی (Chrominance)

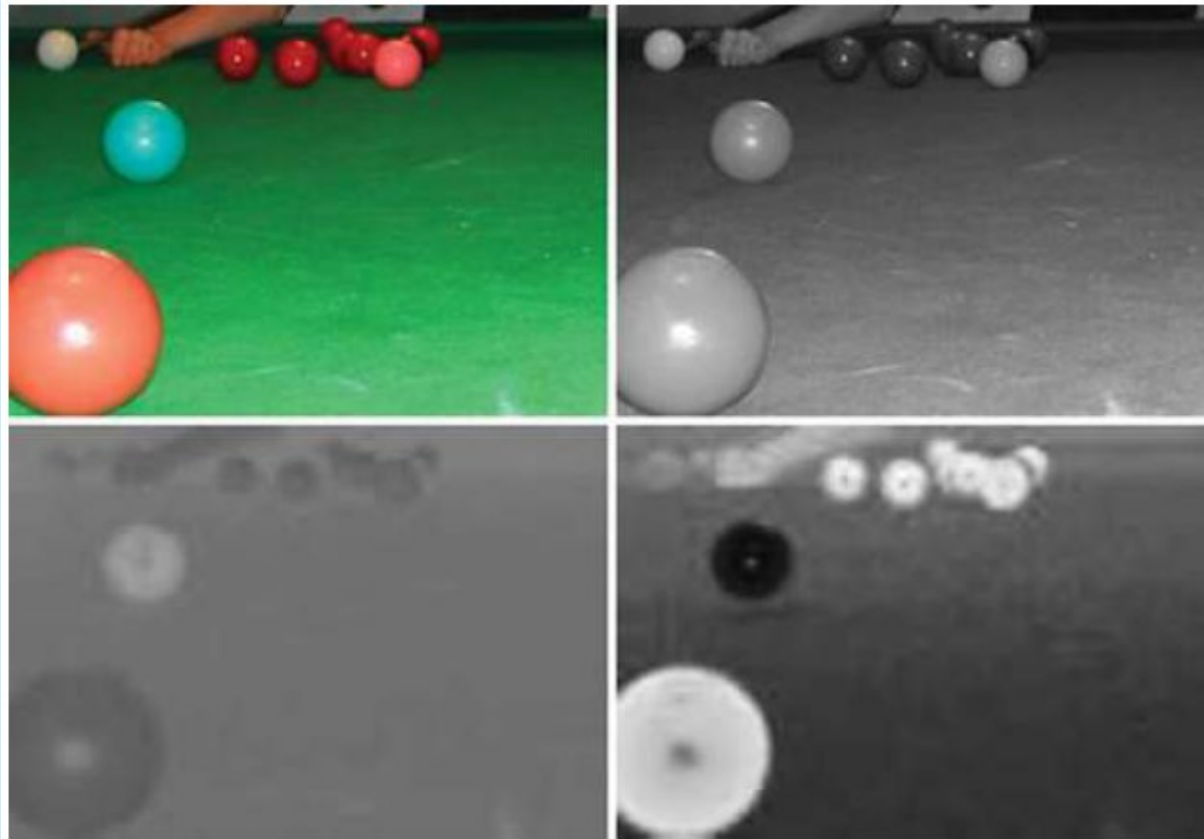
در OpenCV: تبدیل از RGB به YUV با استفاده از تابع `cvtColor` به صورت زیر

```
cvtColor(image, yuv_image, CV_BGR2YUV);
```

تصویرها: نمایش تصویر YUV و کانالهای سه گانه آن به صورت مجزا در شکل ۲-۹ اسلاید بعدی

# فصل دوم: تصویرها و ویژ گیهای آنها

## تصویرهای YUV (ادامه)



شکل ۲-۹: تصویر رنگی YUV بالا سمت چپ، کانال روشنایی (Y) بالا سمت راست،  
کانال رنگ U پایین سمت چپ و کانال رنگ V پایین سمت راست

# فصل دوم: تصویرها و ویژگیهای آنها

## تصویرهای HLS (Hue Luminance Saturation)

مدل رنگ HLS: دارای کاربرد فراوان در موضوعهای بینایی کامپیوتر

دلیل جداسازی دقیق دو فاکتور روشنایی و رنگ تابی

تجزیه رنگ تابی به دو فاکتور فام (Hue) و اشباع (Saturation)

ارائه کمیت‌های مناسب برای تعریف رنگهایی چون آبی تیره، قرمز روشن یا سبز متوسط

ویژگیهای مدل HLS: محدوده میزان روشنایی یا L از صفر تا یک بیان کننده میزان درخشندگی

فام یا H توصیف کننده رنگ و محدوده آن از صفر تا ۳۶۰

اشباع یا S در محدوده صفر تا یک، تعیین درجه استحکام یا خلوص رنگ

تبدیل این مقادیر در کاربردهای عملی به محدوده صفر تا ۲۵۵

در OpenCV محدوده فام بین صفر تا ۱۷۹

نمایش اجزای مدل HLS در شکل ۲-۱۰

بیان میزان روشنایی (Luminance) در محور عمودی

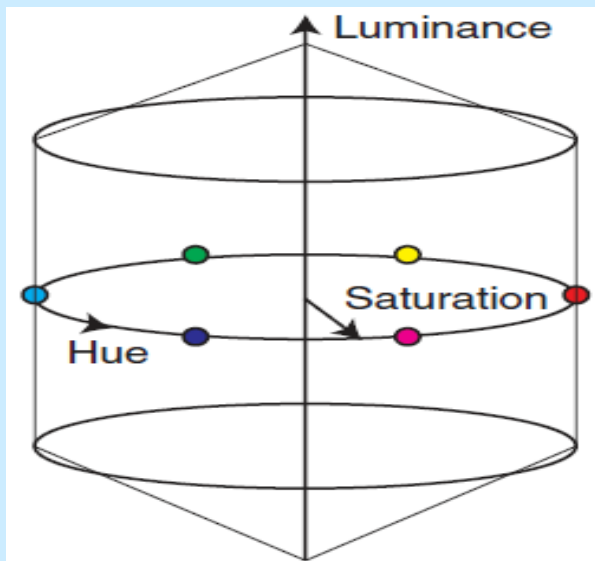
نمایش رنگهای مختلف حول محور فام در وسط استوانه

بیان عمق رنگ در قالب دوری و نزدیکی شعاع اشباع از مرکز

محدوده استوانه‌ای وسط عریض برای نمایش عادی

پایین و بالای استوانه با مقادیر خیلی بالا یا خیلی پایین درخشندگی

برای بیان شیئی بسیار براق یا بسیار تاریک فاقد داده رنگ موثر



شکل ۲-۱۰: فضای HLS شامل روشنایی، فام و اشباع

# فصل دوم: تصویرها و ویژگیهای آنها

## تصویرهای HLS (ادامه)

واقعیت‌های مدل HLS: توجه به اجزای مدل در شکل ۲-۱۰ اسلاید قبل  
تفاوت فقط یک واحد بین میزان کمینه (صفر) و بیشینه (۱۷۹) برای فام به دلیل ماهیت دایره‌ای آن  
متناظر بودن مقادیر بالا با پیکسل قرمز

بررسی نمایش تصویرهای مدل کامل و کانالهای مختلف مدل در شکل ۲-۱۱ اسلاید بعدی  
نمایش توپهای قرمز در این تصویر با هر دو پیکسل سفید و سیاه یعنی مقادیر فام نزدیک به صفر و ۱۷۹  
نتیجه: نیاز به دقت و انجام پردازشهای ویژه زمان انجام محاسبات روی مقادیر مختلف در کانال فام  
مثال: در محاسبه میانگین مقادیر ۱۷۷، ۱، ۱۷۸، ۰ و ۱۷۹ مقدار نتیجه باید ۱۷۹ باشد به جای ۱۰۷  
در OpenCV: تبدیل از RGB به HLS با استفاده از تابع `cvtColor` به صورت زیر

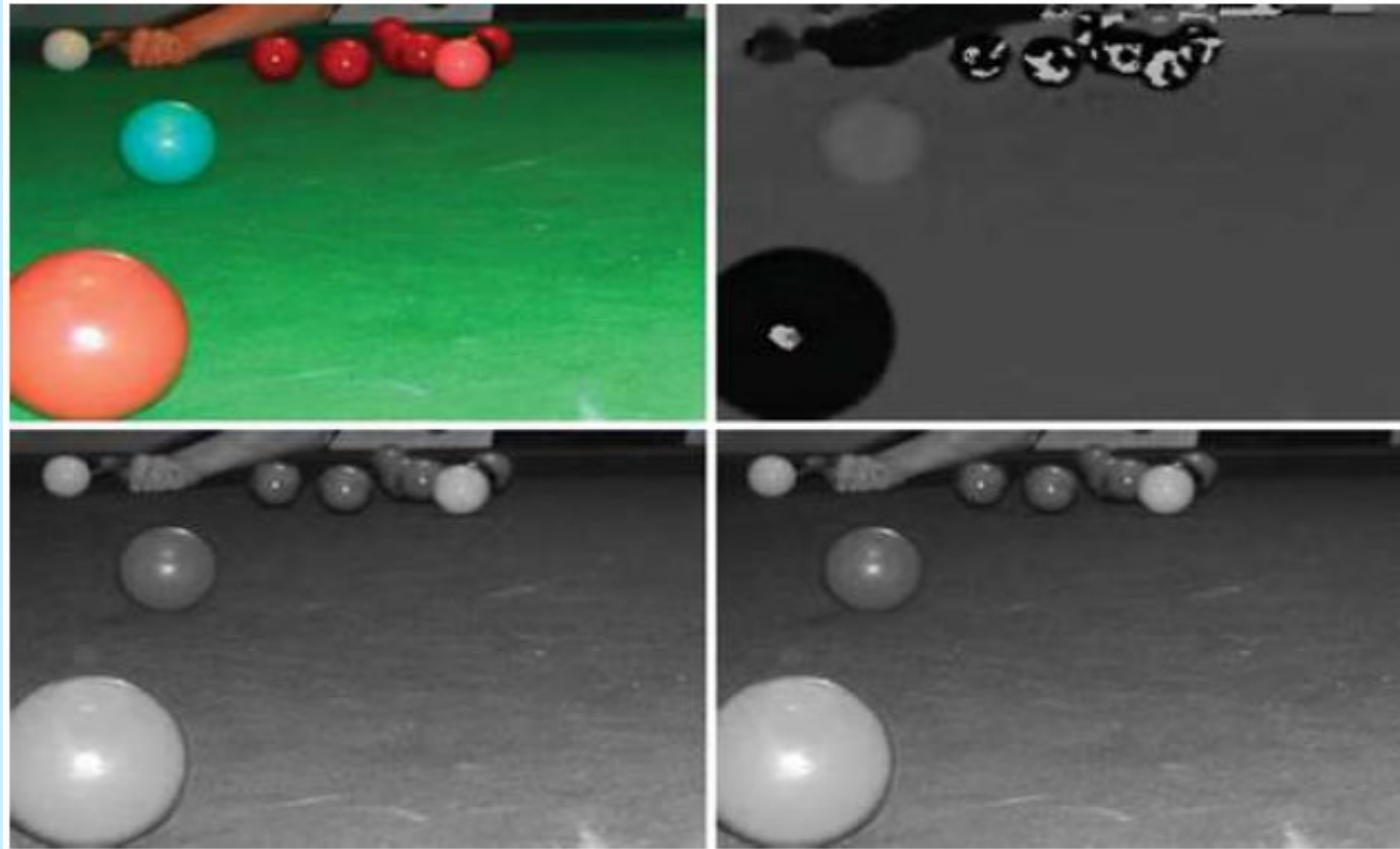
```
cvtColor(bgr_image, hls_image, CV_BGR2HLS);
```

توجه به محدوده مقادیر اشباع و روشنایی از صفر تا ۲۵۵ و مقدار فام از صفر تا ۱۷۹

محاسبه مقادیر: رابطه‌های ریاضی محاسبه مقادیر HLS از مقادیر RGB در اسلاید ۲۹

# فصل دوم: تصویرها و ویژ گیهای آنها

## تصویرهای HLS (ادامه)



شکل ۲-۱: تصویر رنگی HLS بالا سمت چپ، کانال روشنایی (L) بالا سمت راست،  
کانال فام (H) پایین سمت چپ و کانال اشباع (S) پایین سمت راست

# فصل دوم: تصویرها و ویژگیهای آنها

## تصویرهای HLS (ادامه)

تبدیل مقادیر: محاسبه مقادیر L, H و S با استفاده از روابط زیر از مقادیر R, G و B  
توجه به فرض نرمال سازی مقادیر R, G و B بین ۰,۰ و ۱,۰

$$L = \frac{Max(R, G, B) + Min(R, G, B)}{2}$$

$$S = \begin{cases} \frac{Max(R, G, B) - Min(R, G, B)}{Max(R, G, B) + Min(R, G, B)} & \text{if } L < 0.5 \\ \frac{Max(R, G, B) - Min(R, G, B)}{2 - (Max(R, G, B) + Min(R, G, B))} & \text{if } L \geq 0.5 \end{cases}$$

$$H = \begin{cases} 60 \cdot (G - B) / S & \text{if } R = Max(R, G, B) \\ 120 + 60 \cdot (B - R) / S & \text{if } G = Max(R, G, B) \\ 240 + 60 \cdot (R - G) / S & \text{if } B = Max(R, G, B) \end{cases}$$

نتیجه: حاصل مقادیر روشنایی L و اشباع S بین صفر و یک و مقدار فام H بین صفر و ۳۶۰  
افزودن عدد ۳۶۰ به هر مقدار منفی H و کسر عدد ۳۶۰ از هر مقدار H بزرگتر از ۳۶۰

# فصل دوم: تصویرها و ویژ گیهای آنها

## سایر فضاها ی رنگی

خلاصه: وجود فضاها ی رنگی متعدد به عنوان شیوه های جایگزین برای ارائه تصویر نداشتن اطلاعاتی بیشتر یا کمتر از مدل های دیگر تصویر مثل RGB یا CMY بر مبنای تئوری اطلاعات وجود اطلاعات بیشتر در آنها نسبت به تصویرهای مقیاس خاکستری امکانات OpenCV: ارائه و پشتیبانی شش فضای رنگی دیگر در این محیط به کمک توابع تبدیل فضای HSV یا Hue Saturation Value: مشابه HLS با کمی تفاوت در تعریف کانالها فضای CIE XYZ: استاندارد مرجع فضای رنگ با واکنشی شبیه واکنش سیستم بینایی انسان فضای YCrCb: نسخه ای از YUV برای استفاده در فشرده سازی تصویر و ویدئو فضای CIE  $L^*u^*v^*$ : برای دستیابی به یک درک متحد از فضای رنگ در زمان وجود اختلاف بین رنگها با آنچه انسان درک می کند،  $L^*$  میزان روشنایی و  $u^*$  و  $v^*$  مقادیر رنگ تابی فضای CIE  $L^*a^*b^*$ : یک فضای رنگ مستقل از سنسور برای نمایش کلیه رنگهای قابل درک توسط انسان الگوی Bayer: مورد استفاده در سنسورهای CCD برای به کارگیری در داده های خام (درون یابی نشده) توجه به توضیحات شکل ۲-۷ اسلاید ۲۰

# فصل دوم: تصویرها و ویژگیهای آنها

## کاربردهای خاص رنگ

موضوع: نیاز به مشخص کردن پیکسلها با یک رنگ خاص در برخی کاربردها  
مثال جستجو به دنبال رنگهای سفید، سیاه، آبی و قرمز برای یافتن علائم جاده  
روش کار: امکان تشخیص و شناسایی رنگهای مختلف با بیان فرمولهای ریاضی ساده  
نیاز به دقت فراوان برای استفاده از این شیوه تشخیص (توجه به مطالب بخش ۲-۵ در ادامه فصل)  
امکان به دست آوردن نتایج مثبت کاذب در شناسایی زیرفضاهای بزرگ در فضای رنگ  
دلیل، مبتنی نبودن دقیق معیارها با تصویرهای مورد پردازش، خام بودن معیارها و وجود اشیای مختلف و نورپردازیهای متفاوت در تصویرها  
توجه به کاربرد اول در ادامه و شناسایی اشتباه برخی بخشهای تصویر  
کاربرد اول-شناسایی پوست: امکان تشخیص پوست یا Skin detection با تحلیل ساده مقادیر پیکسلها  
رابطه حاصل از تجربیات ساده برای تشخیص پوست

$$(Saturation \geq 0.2) \text{ AND } (0.5 < Luminance/Saturation < 3.0) \text{ AND } (Hue \leq 28^\circ \text{ OR } Hue \geq 330^\circ)$$

نتیجه اجرا: توجه به نتیجه در شکل ۲-۱۲ اسلاید بعدی  
جداسازی بخشهای دیگر تصویر مثل پرچم و سرآستین علاوه بر پوست صورت فزانورد



# فصل دوم: تصویرها و ویژ گیهای آنها

## کاربردهای خاص رنگ (ادامه)

مثال: کاربرد رنگ برای شناسایی پوست با استفاده از رابطه اسلاید قبلی



شکل ۲-۱۲: نتیجه اجرای رابطه تشخیص پوست، توجه به خطای جداسازی بخشهای دیگر تصویر مثل پرچم و سرآستین علاوه بر پوست صورت فضانورد

# فصل دوم: تصویرها و ویژگیهای آنها

## کاربردهای خاص رنگ (ادامه)

در OpenCV: اجرای دستورات زیر برای شناسایی یک پیکسل نشاندهنده پوست با استفاده از رابطه ارائه شده در اسلاید ۳۲

```
uchar H = hls_image.at<Vec3b>(row,col) [0] ;  
uchar L = hls_image.at<Vec3b>(row,col) [1] ;  
uchar S = hls_image.at<Vec3b>(row,col) [2] ;  
double LS_ratio = ((double) L) / ((double) S) ;  
bool skin_pixel = (S >= 50) && (LS_ratio > 0.5) &&  
                  (LS_ratio < 3.0) && ((H <= 14) || (H >= 165)) ;
```

مقاله مرجع برای کسب اطلاعات بیشتر در مورد شناسایی پوست

(Kakumanu, Makrogiannis, & Bourbakis, 2007)

# فصل دوم: تصویرها و ویژگیهای آنها

## کاربردهای خاص رنگ (ادامه)

کاربرد دوم – چشم قرمز: امکان تشخیص چشم قرمز یا Red eye در تصویر با تحلیل ساده مشابه رابطه حاصل از تجربیات ساده برای تشخیص چشم قرمز

$(\text{Luminance} \geq 0.25) \text{ AND } (\text{Saturation} \geq 0.4) \text{ AND}$   
 $(0.5 < \text{Luminance}/\text{Saturation} < 1.5) \text{ AND } (\text{Hue} \leq 14^\circ \text{ OR Hue} \geq 324^\circ)$

نتیجه اجرا: نتیجه تشخیص چشم قرمز در شکل ۲-۱۳ اسلاید بعدی، یک نقطه شروع مناسب برای این منظور در OpenCV: اجرای دستورات زیر برای شناسایی یک پیکسل نشاندهنده چشم قرمز

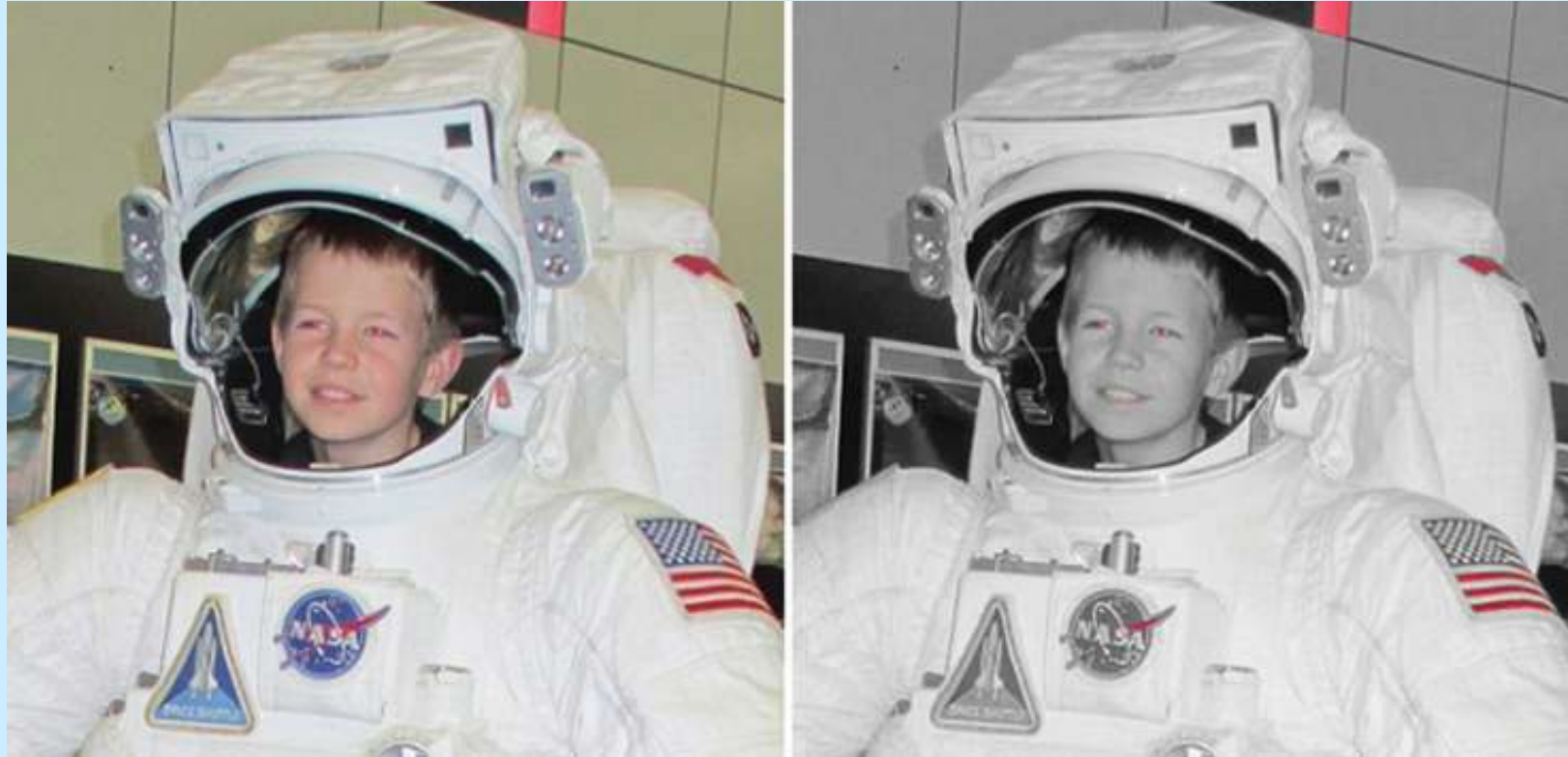
```
uchar H = hls_image.at<Vec3b>(row,col)[0];
uchar L = hls_image.at<Vec3b>(row,col)[1];
uchar S = hls_image.at<Vec3b>(row,col)[2];
double LS_ratio = ((double) L) / ((double) S);
bool red_eye_pixel = (L >= 64) && (S >= 100) &&
                    (LS_ratio > 0.5) && (LS_ratio < 1.5) &&
                    ((H <= 7) || (H >= 162));
```

مقاله مرجع برای کسب اطلاعات بیشتر در مورد شناسایی چشم قرمز (Gasparini & Schettini, 2009)

# فصل دوم: تصویرها و ویژ گیهای آنها

## کاربردهای خاص رنگ (ادامه)

مثال: کاربرد رنگ برای شناسایی چشم قرمز با استفاده از رابطه اسلاید قبلی



شکل ۲-۱۳: نتیجه اجرای رابطه تشخیص چشم قرمز و جداسازی قرمزیهای دیگر

# فصل دوم: تصویرها و ویژگیهای آنها

## ۲-۴ نویزها

تعریف کلی: هر نوع داده اضافی همراه با داده‌های اصلی تصویر که از کیفیت و کمیت تصویر کم کند  
منبعهای نویز: دستگاه تصویربرداری، مزاحمت‌های الکتریکی و الکترونیکی، فرایند دیجیتال سازی، ...  
نتیجه فنی: ایجاد تاثیر روی پردازش تصویر و استخراج نتیجه در بینایی کامپیوتر  
رویکرد ما در پردازش تصویر: تلاش برای شناسایی، اندازه‌گیری و تصحیح داده‌ها (حذف نویز)  
در این بخش، بررسی انواع نویز، تاثیر نویز بر تصویرها و نحوه شبیه‌سازی نویز  
معیار سنجش: نسبت سیگنال به نویز به عنوان رایج‌ترین معیار سنجش با رابطه زیر  
 $f(i, j)$  بیان کننده تصویر و  $v(i, j)$  بیان کننده نویز  
استفاده از معیار زیر و موارد مرتبط برای اجرا و ارزیابی تکنیکهای حذف نویز

$$S/Nratio = \frac{\sum_{(i,j)} f^2(i,j)}{\sum_{(i,j)} v^2(i,j)}$$

انواع نویز: نویز گاوسی (Gaussian) و نویز نمک و فلفل (Salt and Pepper) دونمونه نویز مرسوم  
توضیح این دو نوع نویز در اسلایدهای بعدی

# فصل دوم: تصویرها و ویژگیهای آنها

## نویز گاوسی (Gaussian)

ویژگی کلی: نزدیکترین نوع نویز به نویزهای واقعی تحت عنوان  $v(i, j)$  مدلسازی این نویز با توزیع گاوسی حول میانگین  $\mu$  با مقدار معمولاً صفر و انحراف استاندارد  $\sigma$  مثال: توجه به چهار تصویر ارائه شده در شکل ۲-۱۴ اسلاید بعدی تصاویر اصلی رنگی بالا و مقیاس خاکستری پایین در سمت چپ شکل همان تصاویر با نویز گاوسی اعمال شده با میانگین صفر و انحراف استاندارد ۲۰ در سمت راست نسبت سیگنال به نویز تصویر رنگی ۴۳,۳ و در تصویر مقیاس خاکستری ۴۰,۳ در OpenCV: قطعه کد ارائه شده در اسلاید ۴۰ برای اعمال نویز گاوسی به یک تصویر لزوم استفاده از مقادیر علامتدار به دلیل وجود هر دو نوع نویز با مقادیر مثبت و یا منفی استفاده از خانه‌های ۱۶ بیتی علامتدار برای ذخیره تصویر به منظور وجود دقت کافی تعریف این نوع حافظه با ثابت نمادی CV\_16SC3 در متن قطعه کد توجه به مراحل تبدیل تصویر به ۱۶ بیتی علامتدار اعمال نویز و سپس برگرداندن تصویر به حالت عادی در سه سطر پایانی کد برنامه



# فصل دوم: تصویرها و ویژگیهای آنها

## نویز گاوسی (ادامه)

تصاویر رنگی و مقیاس خاکستری همراه با تصاویر با نویز گاوسی اعمال شده



شکل ۲-۱۴: تصاویر رنگی و مقیاس خاکستری سمت چپ و تصاویر با اعمال نویز گاوسی سمت راست

# فصل دوم: تصویرها و ویژگیهای آنها

## نویز گاوسی (ادامه)

قطعه کد در محیط OpevCV برای اعمال نویز گاوسی به یک تصویر

```
void addGaussianNoise(Mat &image, double average=0.0,  
                      double standard_deviation=10.0)  
{  
    // We need to work with signed images (as noise can be  
    // negative as well as positive). We use 16 bit signed  
    // images as otherwise we would lose precision.  
    Mat noise_image(image.size(), CV_16SC3);  
    randn(noise_image, Scalar::all(average),  
          Scalar::all(standard_deviation));  
  
    Mat temp_image;  
    image.convertTo(temp_image, CV_16SC3);  
    addWeighted(temp_image, 1.0, noise_image, 1.0, 0.0, temp_image);  
    temp_image.convertTo(image, image.type());  
}
```



# فصل دوم: تصویرها و ویژگیهای آنها

## نویز Salt and Pepper

ویژگی کلی: نوع خاصی از نویز Impulse اشباع شده

مجزا بودن خطای پیکسلهای نویزی در نویز Impulse

به معنی وجود اختلاف قابل توجه در روشنایی پیکسل نویزی در مقایسه با پیکسلهای همسایه  
ایجاد خرابی با پیکسلهای سفید خالص یا سیاه خاص در نویز Salt and Pepper (مفهوم اشباع)

مثال: توجه به چهار تصویر ارائه شده در شکل ۲-۱۵ اسلاید بعدی

تصاویر اصلی رنگی بالا و مقیاس خاکستری پایین در سمت چپ شکل

همان تصاویر با نویز Salt and Pepper اعمال شده به میزان ده درصد در سمت راست

نسبت سیگنال به نویز تصویر رنگی ۷,۵ و در تصویر مقیاس خاکستری ۶,۷

در OpenCV: قطعه کد ارائه شده در اسلاید ۴۳ برای اعمال نویز Salt and Pepper به تصویر

توجه به نحوه محاسبه تعداد پیکسل نویز قابل اعمال

توجه به محاسبه تصادفی محل اعمال نویز در سطر و ستون و کانال

توجه به ایجاد پیکسل سفید یا سیاه در سطر پایانی کد

# فصل دوم: تصاویرها و ویژگیهای آنها

## نویز Salt and Pepper (ادامه)

تصاویر رنگی و مقیاس خاکستری همراه با تصاویر با نویز Salt and Pepper اعمال شده



شکل ۲-۱۵: تصاویر رنگی و مقیاس خاکستری سمت چپ و تصاویر با اعمال نویز Salt and Pepper سمت راست

# فصل دوم: تصویرها و ویژگیهای آنها

## نویز Salt and Pepper (ادامه)

قطعه کد در محیط OpevCV برای اعمال نویز Salt and Pepper به یک تصویر

```
void addSaltAndPepperNoise(Mat &image, double noise_percentage)
{
    int image_rows = image.rows;
    int image_columns = image.cols;
    int image_channels = image.channels();
    int noise_points = (int) (((double) image_rows *
                               image_columns * image_channels) * noise_percentage / 100.0);
    for (int count = 0; count < noise_points; count++)
    {
        int row = rand() % image_rows;
        int column = rand() % image_columns;
        int channel = rand() % image_channels;
        uchar* pixel = image.ptr<uchar>(row) +
                        (column * image_channels) + channel;

        *pixel = (rand() % 2 == 1) ? 255 : 0;
    }
}
```

# فصل دوم: تصویرها و ویژگیهای آنها

## مدلهای نویز

ویژگی نویز: لزوم پیوند خوردن داده‌های نویز با داده‌های تصویر  
انتخاب مدل نویز بر مبنای مستقل از داده بودن نویز یا وابسته به داده بودن نویز  
توجه به انواع نویز توضیح داده شده در اسلایدهای قبلی

نویز افزودنی: مناسب برای موارد نویز مستقل از داده و بدون ارتباطی با خود داده‌های تصویر  
ایجاد نویز افزودنی با استفاده از رابطه زیر،  $g(i, j)$  تصویر ایده‌آل،  $v(i, j)$  نویز و  $f(i, j)$  تصویر واقعی  
مثالها: تصویرهای موجود در شکل‌های ۲-۱۴ اسلاید ۳۹ و شکل ۲-۱۶ اسلاید بعدی

$$f(i, j) = g(i, j) + v(i, j)$$

نویز ضربی: مناسب برای موارد نویز وابسته به داده و ارتباط مقداری از نویز با داده‌های خود تصویر  
ایجاد نویز ضربی با استفاده از رابطه زیر، توجه به تفاوت با رابطه بالا  
اینجا هم  $g(i, j)$  تصویر ایده‌آل،  $v(i, j)$  نویز و  $f(i, j)$  تصویر واقعی

$$f(i, j) = g(i, j) + g(i, j).v(i, j)$$



# فصل دوم: تصویرها و ویژگیهای آنها

## مدلهای نویز (ادامه)

اعمال مدل گاوسی با میانگین صفر و انحراف استاندارد ۳۰ روی تصویر سمت چپ

نتیجه در تصویر وسط با نسبت سیگنال به نویز ۱۷,۵

میانگین گیری از هشت تصویر نویزی (موضوع بخش ۲-۵) در سمت راست با نسبت سیگنال به نویز

۱۲۹,۷ (بسیار بالا و نتیجه خیلی نزدیک به تصویر اصلی)



شکل ۲-۱۶: تصویر اصلی سمت چپ، تصویر با نویز گاوسی وسط و میانگین هشت تصویر نویزی سمت راست

# فصل دوم: تصویرها و ویژگیهای آنها

## روند تولید نویز

دلیل نیاز به تولید نویز: نیاز به شبیه‌سازی نویز به منظور ایجاد امکان حذف یا کاهش آن برای سنجش میزان موفقیت الگوریتمهای کاربردی

هدف: ایجاد نویز با توزیع گاوسی با میانگین  $\mu$  برابر صفر و انحراف استاندارد خاص  $\sigma$

بررسی ریاضی: شروع کار با محاسبه توزیع احتمال  $p(k)$

سپس محاسبه توزیع احتمال تجمعی  $p_{\text{cum}}(k)$  برای تمام مقادیر ممکنه نویز  
بازه مقادیر ممکنه از کوچکترین تغییر مقدار منفی تا بزرگترین تغییر مقدار مثبت

بازه مزبور در عمل  $k = -255 -254 -253 \dots +253 +254 +255$

محاسبات به روش بازگشتی

$$p(k) = e^{-k^2/2\sigma^2} / \sigma \sqrt{2\pi} \quad k = -(G-1), \dots, -1, 0, 1, \dots, G-1$$

$$p_{\text{cum}}(k) = p_{\text{cum}}(k-1) + p(k)$$

$$p_{\text{cum}}(-(G-1)) = p(-(G-1))$$

ادامه محاسبات در اسلاید بعدی

# فصل دوم: تصویرها و ویژگیهای آنها

## روند تولید نویز (ادامه)

محاسبه نویز: محاسبه مقدار نویز برای هر پیکسل در تصویر پس از تعیین توزیع تجمعی به شرح زیر

$$f^*(i, j) = g(i, j) + \operatorname{argmin}_k (\operatorname{rand}() - p_{\text{cum}}[k])$$

Set

$$f'(x, y) = 0 \quad \text{if } f^*(x, y) < 0$$

$$f'(x, y) = G - 1 \quad \text{if } f^*(x, y) > G - 1$$

$$f'(x, y) = f^*(x, y) \quad \text{otherwise}$$

نکته اول: لزوم توجه به تابع  $\operatorname{argmin}_k$  برای محاسبه شاخص کوچکترین مقدار در اینجا برگرداندن مقداری از  $k$  که مقدار تابع تجمعی برای آن  $k$  نزدیکترین به عدد تصادفی است

نکته دوم: کوتاه کردن بازه مورد بحث با تغییر مقادیر برای قرار گرفتن مقادیر در بازه صفر تا ۲۵۵ توجه به تغییر ماهیت نویز گاوسی تا حدی

# فصل دوم: تصویرها و ویژگیهای آنها

## ارزیابی نویز

دو روش ارزیابی نویز: روش غیرعینی یا Subjective و عینی یا Objective  
ارزیابی غیرعینی: ارائه نمودن تصویر به ناظر مسئول برای ارزیابی آن بر مبنای معیارها و نشانه‌های خاص و تعیین نمره برای آن تصویر

ارزیابی عینی: مقایسه تصویر مورد ارزیابی  $f(i, j)$  با یک تصویر مرجع شناخته شده  $g(i, j)$  برای تعیین اختلاف آنها به یکی از روشهای زیر، مشکل ارزیاب عینی نیاز داشتن به یک تصویر ایده‌آل مرجع

الف - میانگین اختلاف درجه دوم  
$$\text{Mean Quadratic Difference} = \sum \sum (g(i, j) - f(i, j))^2$$

ب - میانگین اختلاف مطلق  
$$\text{Mean Absolute Difference} = \sum \sum |g(i, j) - f(i, j)|$$

پ - محاسبه سیگنال به نویز با فرض  $v(i, j) = f(i, j) - g(i, j)$

روش عملی: افزودن نویز به صورت مصنوعی و سپس ارزیابی تکنیک مورد استفاده برای حذف نویز  
استفاده از همین روش در نمونه‌های ارائه شده در این فصل



# فصل دوم: تصویرها و ویژگیهای آنها

## ۲-۵ هموارسازی (Smoothing)

مفهوم: حذف یا کاستن از میزان نویز در تصویر به روشهای مختلف  
توجه به مناسب بودن هر تکنیک برای استفاده در شرایط خاص  
انتخاب و به کارگیری تکنیک مناسب بر مبنای فرضیات و یا ماهیت داده‌های تصویر

رایج‌ترین شیوه: به کارگیری تبدیلات خطی هموارسازی (Linear smoothing transformation)  
سادگی به دلیل بیان محاسبات بر مبنای مجموع خطی

مشکل روش خطی: مبهم یا تار شدن لبه‌های تیز در تصویر هنگام به کارگیری این روش  
امکان به کارگیری شیوه تبدیل غیرخطی ولی دارای پیچیدگی در مقایسه با روش خطی  
لزوم انجام مقداری عملیات منطقی و تست در هر منطقه از تصویر در شیوه تبدیل غیرخطی

مشکل کلی: عدم امکان حذف لکه‌های بزرگ یا خشهای ضخیم از روی تصویر در هیچیک از این روشها  
لزوم به کارگیری تکنیکهای image restoration در حوضه فرکانس برای حل اینگونه مشکلات  
توضیح: شرح موضوعات پردازش حوضه فرکانس در مباحث فصلهای آخر درس

# فصل دوم: تصویرها و ویژگیهای آنها

## میانگین تصویر

مفهوم: روشی برای کاستن از میزان نویز در زمان وجود چندین تصویر دقیقاً از یک صحنه محاسبه میانگین  $n$  تصویر با استفاده از رابطه زیر با فرض استفاده از مدل نویز افزودنی

$$f'(i,j) = \frac{1}{n} \sum_{k=1..n} f_k(i,j) = \frac{1}{n} \sum_{k=1..n} g_k(i,j) + v_k(i,j)$$

فرضیات: یکسان بودن همه  $g_k(i, j)$  ها یعنی تصاویر از یک صحنه ثابت و با دوربین ثابت عدم وجود وابستگی آماری بین نویز  $v_k(i, j)$  در هر تصویر

استفاده از توزیع گاوس با میانگین صفر و انحراف استاندارد  $\sigma$  برای نویز  $v_k(i, j)$

نتیجه: تغییر در توزیع نویز پس از میانگین گرفتن با حفظ ماهیت گاوسی ولی کاستن از انحراف استاندارد با ضریب جذر  $n$

مثال: تصویر سمت راستی در شکل ۲-۱۶ اسلاید ۴۵

نکته ۱: مناسب نبودن این نوع هموارسازی در نویز Salt and Pepper به دلیل میانگین شدن نویز در تصویر

نکته ۲: بروز تاثیرات ناخواسته در صورت ثابت نبودن صحنه و وجود شیء های متحرک در آن

```
addWeighted(image1,0.5,image2,0.5,0.0,average_image);
```

در OpenCV: نحوه میانگین گیری  
بدین صورت

# فصل دوم: تصویرها و ویژگیهای آنها

## میانگین محلی و هموارسازی گاوس

مورد استفاده: روش میانگین گیری زمانی که فقط یک تصویر در اختیار باشد  
محاسبه میانگین در همسایگی محلی اطراف نقطه مورد نظر به جای تصویرهای متعدد از یک نقطه  
نحوه کار: به دست آوردن میانگین از یک بلوک  $3 \times 3$  به مرکزیت نقطه مورد نظر یعنی خود نقطه و هشت همسایه اطراف آن (ماسک  $3 \times 3$ )

میانگین محلی: به دست آوردن این میانگین در صورت برابر فرض کردن وزن همه نقاط  
میانگین وزن دار: در صورت تخصیص وزنهاى متفاوت به نقاط همسایه نسبت به نقطه اصلی  
هموارسازی گاوس: نمونه‌هایی از روشهای وزن‌دهی برای محاسبه میانگین در این نوع هموارسازی  
 $h_1$  یک فیلتر میانگین محلی

$h_2$  فیلتر هموارسازی گاوس با مقدار کوچک  $\sigma$  (انحراف استاندارد تابع گاوس)

$h_3$  فیلتر هموارسازی گاوس با مقدار بزرگتر  $\sigma$ ، توجه به نمونه‌های ارائه شده در شکل‌های ۲-۱۷ و ۲-۱۸

$$h_1 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_2 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad h_3 = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

# فصل دوم: تصویرها و ویژ گیهای آنها

## میانگین محلی و هموارسازی گاوس (ادامه)

نمونه‌هایی از هموارسازی: a- تصویر عادی، b- اعمال فیلتر میانگین محلی، c- اعمال هموارسازی گاوس  
d- اعمال ماسک چرخشی و e- اعمال فیلتر میانه (دوتای آخر مربوط به مباحث بعدی)



شکل ۲-۱۷: نتیجه اعمال هموار سازیهای مختلف روی یک تصویر

مشکلات: بروز محوشدگی (Blur) در اثر میانگین‌گیری با پیکسل‌های همسایه در این نوع هموارسازی و کاهش نویز گاوسی قابل مشاهده

بروز تاثیرات قابل توجه در اثر Blur در کرانها یا لبه‌های تصویر (مکانهای دارای تغییر ناگهانی در مقیاس خاکستری یا رنگ تصویر)

احتمال تیره یا مبهم شدن تصویر و پیچیده شدن کار پردازش آن

نکته: الزام به پذیرش این محوشدگی در برخی موارد برای کاستن از نویز تا یک سطح قابل قبول

# فصل دوم: تصویرها و ویژگیهای آنها

## میانگین محلی و هموارسازی گاوس (ادامه)

در OpenCV: نحوه میانگین گیری محلی با فیلتر  $3 \times 3$

```
blur(image, smoothed_image, Size(3,3));
```

در OpenCV: نحوه هموارسازی گاوس با فیلتر  $5 \times 5$  و  $\sigma=1.5$

```
GaussianBlur(image, smoothed_image, Size(5,5), 1.5);
```

نکته: مناسب نبودن این نوع فیلتر در نویز Salt and Pepper به دلیل هموارسازی شدن نویز در تصویر

ماسکهای بزرگتر: استفاده از ماسکهای فیلتر بزرگتر از  $3 \times 3$  با تاثیر بیشتر در کاهش نویز و افزایش Blur

ماسک پیچیدگی: انجام این تکنیکهای فیلتر کردن با استفاده از تکنیک Convolution یا پیچیدگی

تعریف یک ماسک پیچیدگی  $h(i, j)$  بیان کننده وزنهای فیلتر هموارسازی

پیچیده شدن ماسک  $h(i, j)$  در تصویر ورودی  $f(i, j)$  برای تولید خروجی  $f'(i, j)$  به صورت زیر

$h(i, j)$  مقدار غیر صفر فقط برای یک ناحیه کوچک

در ماسک  $3 \times 3$  مقادیر غیر صفر بین  $-1$  و  $+1$

مربعی بودن ماسک فیلتر با تعداد فرد پیکسل در دو طرف

با اطمینان از تقارن حول نقطه اصلی مثل  $3 \times 3$  یا  $5 \times 5$

$$f'(i, j) = \sum_m \sum_n f(i, j).h(i - m, j - n)$$

# فصل دوم: تصویرها و ویژگیهای آنها

## ماسک چرخشی (Rotating Mask)

مفهوم: تعریف یک عملگر غیرخطی بر مبنای یکی از نه فیلتر میانگین محلی ممکن در نظر گرفتن ناحیه‌ای از تصویر با بیشترین مشابهت، متغیر بودن شکل و اندازه ماسکهای مورد استفاده

توجه به شکل ۲-۱۸ اسلاید بعدی: نقطه اصلی مشخص شده با علامت  $X$  نمایش تمام نواحی  $3 \times 3$  ممکن حاوی نقطه اصلی و نواحی با شکل و هدف متفاوت حاوی نقطه اصلی هدف: کاستن از نویز در نقطه اصلی با انتخاب مناسب‌ترین ماسک برای نقطه مزبور محاسبه میانگین نقطه اصلی با سایر نقاط مشابه از همان محرک فیزیکی (واسط یا شیء) تخمین مقدار پیکسل با بسط منشا محلی به جای بیان دقیق منشا فیزیکی پیکسل

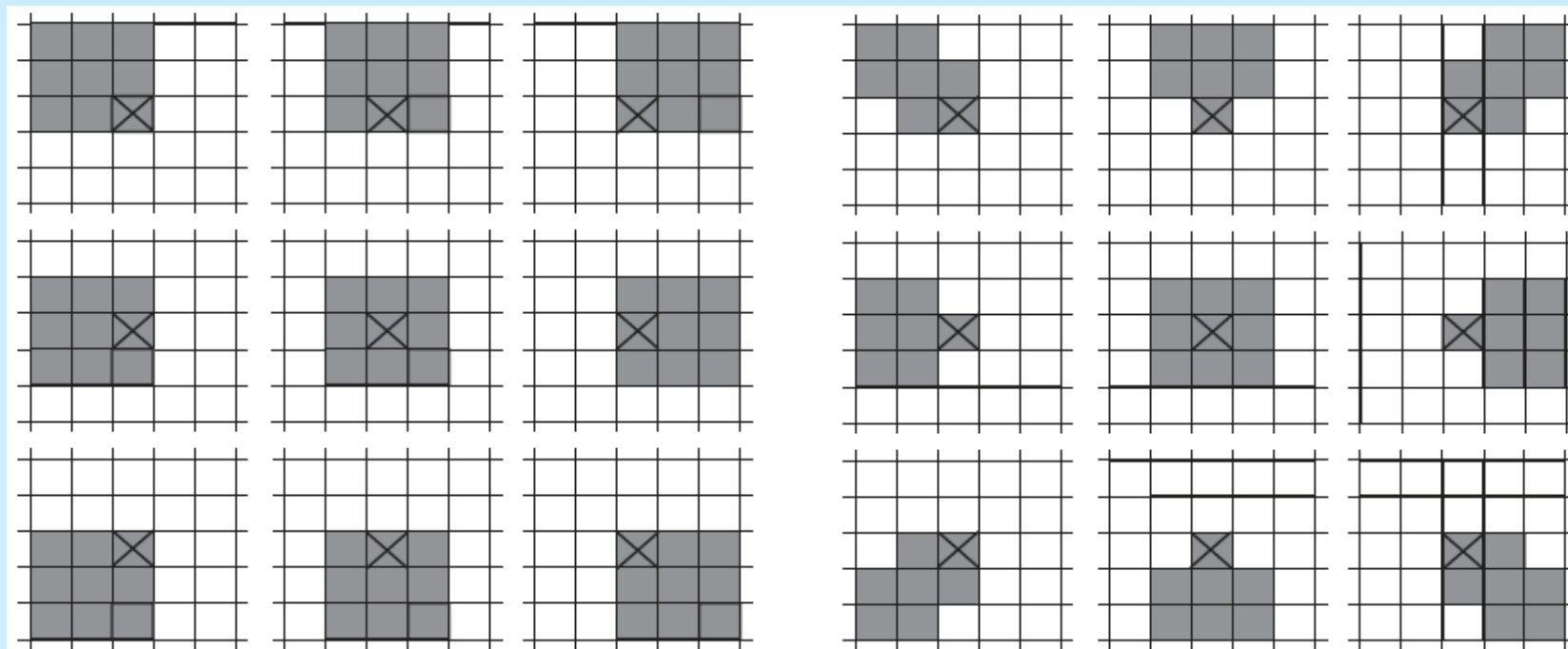
مراحل اجرا: برای هر نقطه  $f(i, j)$

- ۱- محاسبه خروجی برای هر یک از ماسکها در قالب میانگین محلی برای نقطه اصلی
  - ۲- محاسبه میانگین کل برای مقادیر مشخص شده حاصل از ماسکهای مورد استفاده
  - ۳- محاسبه میزان پراکندگی یعنی میزان اختلاف پیکسلها در هر میانگین محلی با مقدار میانگین کلی
  - ۴- یافتن ماسکی که میزان پراکندگی آن با میانگین از همه پایین‌تر است
  - ۵- تخصیص مقدار یافت شده در مرحله قبل به عنوان  $f'(i, j)$  به نقطه مورد بررسی
- تکرار مراحل بالا تا زمان رسیدن به هیچ تغییر یا کمترین تغییر در نتایج متوالی همگرایی سریعتر الگوریتم (تعداد تکرار کمتر) با افزایش سایز ماسک

# فصل دوم: تصویرها و ویژگیهای آنها

## ماسک چرخشی (ادامه)

مجموعه‌های فیلتر هموارسازی ماسک چرخشی شامل تمام  $3 \times 3$  ممکن و همان نواحی با شکل و هدف متفاوت نمایش نه عدد فیلتر در هر مجموعه با نقطه اصلی مشخص شده با X



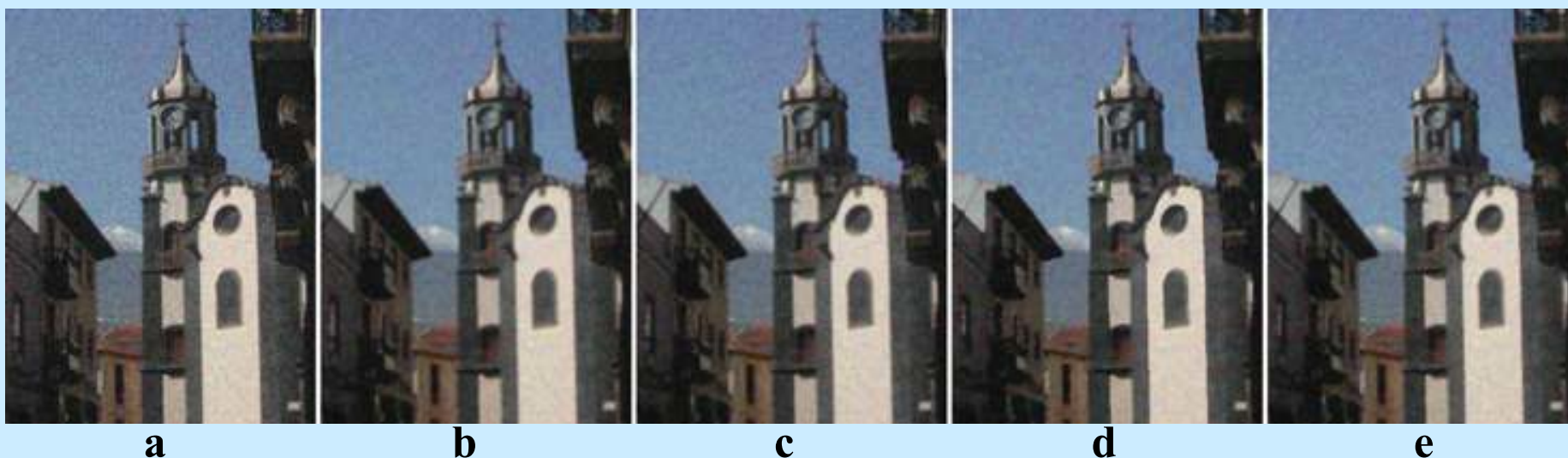
شکل ۲-۱: نمونه دو مجموعه از ماسکهای مناسب برای فیلتر هموارسازی ماسک چرخشی در OpenCV: نبود تابع خاص برای ماسک چرخشی ولی ساده بودن پیاده‌سازی آن طبق الگوریتم



# فصل دوم: تصویرها و ویژ گیهای آنها

## ماسک چرخشی (ادامه)

مزایای ماسک چرخشی: موثر بودن قابل توجه میانگین گیری در رفع نویز و تیز (Sharp) شدن لبه ها  
مشکل: کندی نسبت به محاسبه میانگین ساده محلی به دلیل حجم بیشتر محاسبات ریاضی (اسلاید ۵۴)  
توجه به شکل های ۲-۱۷ d و ۲-۱۹ d



شکل ۲-۱۹: a- تصویر با نویز گاوسی با میانگین صفر و انحراف استاندارد ۲۰، b- پس از میانگین گیری محلی ۳\*۳، c- هموارسازی گاوسی، d- تصویر با ماسک چرخشی و e- فیلتر میانه  
سیگنال به نویز به ترتیب: a- ۳۷,۳، b- ۱۱۵,۵، c- ۱۳۰,۶، d- ۱۰۷,۱ و e- ۱۱۲,۱



# فصل دوم: تصویرها و ویژگیهای آنها

## ماسک چرخشی (ادامه)

$$D = \frac{1}{n} \sum_{(i,j) \in \text{Mask}} \left( f(i,j) - \frac{1}{n} \sum_{(i',j') \in \text{Mask}} f(i',j') \right)^2$$

محاسبات ریاضی میانگین گیری با ماسک چرخشی  
محاسبه پراکندگی D برای هر ماسک (مرحله ۳)  
با استفاده از مربع اختلاف هر نقطه در ماسک

با میانگین کل ماسکها، استفاده از آن برای تعیین ماسک با کمترین D (مرحله ۴)  
ساماندهی مجدد با بسط مربع برای سهولت پیاده سازی با محاسبات کارآمد

$$D = \frac{1}{n} \sum_{(i,j) \in \text{Mask}} \left( f(i,j)^2 - \frac{2}{n} \cdot f(i,j) \cdot \sum_{(i',j') \in \text{Mask}} f(i',j') + \frac{1}{n^2} \left( \sum_{(i',j') \in \text{Mask}} f(i',j') \right)^2 \right)$$

وارد کردن سیگما به داخل و ساده سازی، رسیدن به رابطه ساده تر و کاهش قابل توجه در حجم محاسبات

$$D = \frac{1}{n} \left( \sum_{(i,j) \in \text{Mask}} f(i,j)^2 - \frac{2}{n} \left( \sum_{(i',j') \in \text{Mask}} f(i',j') \right)^2 + \frac{n}{n^2} \left( \sum_{(i',j') \in \text{Mask}} f(i',j') \right)^2 \right)$$

$$D = \frac{1}{n} \left( \sum_{(i,j) \in \text{Mask}} f(i,j)^2 - \frac{1}{n} \left( \sum_{(i',j') \in \text{Mask}} f(i',j') \right)^2 \right)$$

امکان اعمال ماسک چرخشی به نویز Salt and Pepper  
بروز نتیجه غیر قابل انتظار در صورت نزدیکی نویز به لبه ها

# فصل دوم: تصویرها و ویژگیهای آنها

## فیلتر میانه (Median Filter)

مفهوم: نوع دیگری از هموارسازی غیرخطی با جایگزینی هر پیکسل با میانه پیکسلها در ناحیه اطراف آن (مثلا  $3 \times 3$ )  
مفهوم میانه: کمیت آماری به معنی عدد وسطی از یک لیست مرتب شده

مثال: با فرض ناحیه  $3 \times 3$  مقیاس خاکستری با مقادیر (۲۵ ۲۱ ۲۳ ۲۵ ۱۸ ۲۵ ۳۰ ۱۳ ۲۲)  
مرتب سازی به صورت (۱۳ ۱۸ ۲۱ ۲۲ ۲۳ ۲۵ ۲۵ ۳۰ ۲۵۵) و استخراج ۲۳ به عنوان میانه  
توجه به تفاوت با میانگین یعنی ۴۸ به دلیل وجود نقطه تکی نویز ۲۵۵

مزایا: عملکرد خوب این فیلتر در رفتار با نویز و عدم ایجاد محوشدگی یا Blur زیاد در لبه ها  
امکان استفاده مکرر از آن و تاثیر آن تا حدی مشابه استفاده از ماسک چرخشی

توجه به تصویرهای ارائه شده در شکل های ۲-۱۷ و ۲-۱۹

مشکلات: امکان صدمه زدن به خطوط باریک و گوشه ها در تصویر

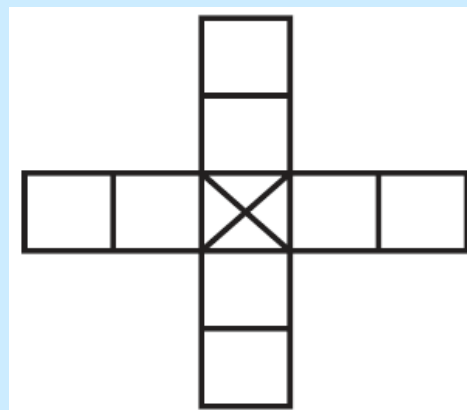
کاهش این تاثیر با به کارگیری ناحیه غیر مستطیلی به صورت شکل ۲-۲۰

بالا بودن هزینه محاسباتی الگوریتم از درجه  $O(k^2 \log k)$

نکته: ارائه الگوریتمی برای این فیلتر توسط (Perreault, 2007) با زمان اجرای

کمتر با استفاده از موضوع هیستوگرام (موضوع فصل بعدی)

در OpenCV: اعمال فیلتر میانه (با فیلتر  $5 \times 5$ )



شکل ۲-۲۰: ماسک غیر مربعی برای فیلتر میانه

```
medianBlur(image, smoothed_image, 5);
```

## فصل دوم: تصویرها و ویژگیهای آنها

تکلیف فصل دوم – دو نفری

مهلت دو هفته

شرح کلی: انتخاب یک نمونه تصویر واقعی و اجرای فیلترهای بررسی شده در این فصل (حداقل دو مورد) روی آن و ارائه نتایج (تعداد فیلتر بیشتر نمره بالاتری دارد)

الف: انتخاب تصویر دلخواه

ب: اعمال نویز مناسب روی آن

پ: اجرای الگوریتمهای حذف نویز انتخاب شده و استخراج نتیجه

ت: تهیه یک گزارش حاوی همه تصاویر و توضیحات کافی برای هر یک

ث: استفاده از محیط OpenCV برای اینکار

ج: انجام ارائه ده دقیقه‌ای در کلاس روی برنامه‌ها

### پایان فصل دوم