



# 数字媒体技术基础

Meng Yang

[www.smartllv.com](http://www.smartllv.com)

SUN YAT-SEN University

机器智能与先进计算教育部重点实验室





## 5 视觉媒体信息表示



## 5 视觉媒体信息表示

### 5. 3 视觉颜色的表示

#### 5. 3. 1 人的视觉颜色感知

#### 5. 3. 2 颜色色度空间

#### 5. 3. 3 常用颜色空间

#### 5. 3. 4 数字视频

### 5. 4 视觉特征的表示

#### 5. 4. 1 基于像素值的主成分表示

#### 5. 4. 2 图像频域特征表示

#### 5. 4. 3 图像特征词袋表示



## 5.3 视觉颜色的表示

## 5.3.1 人的视觉颜色感知

问题：搬家用深色箱子还是浅色？



## 5.3.1 人的视觉颜色感知

### 日光的光谱分布

- The symbol for wavelength is  $\lambda$ . This curve is called  $E(\lambda)$ .

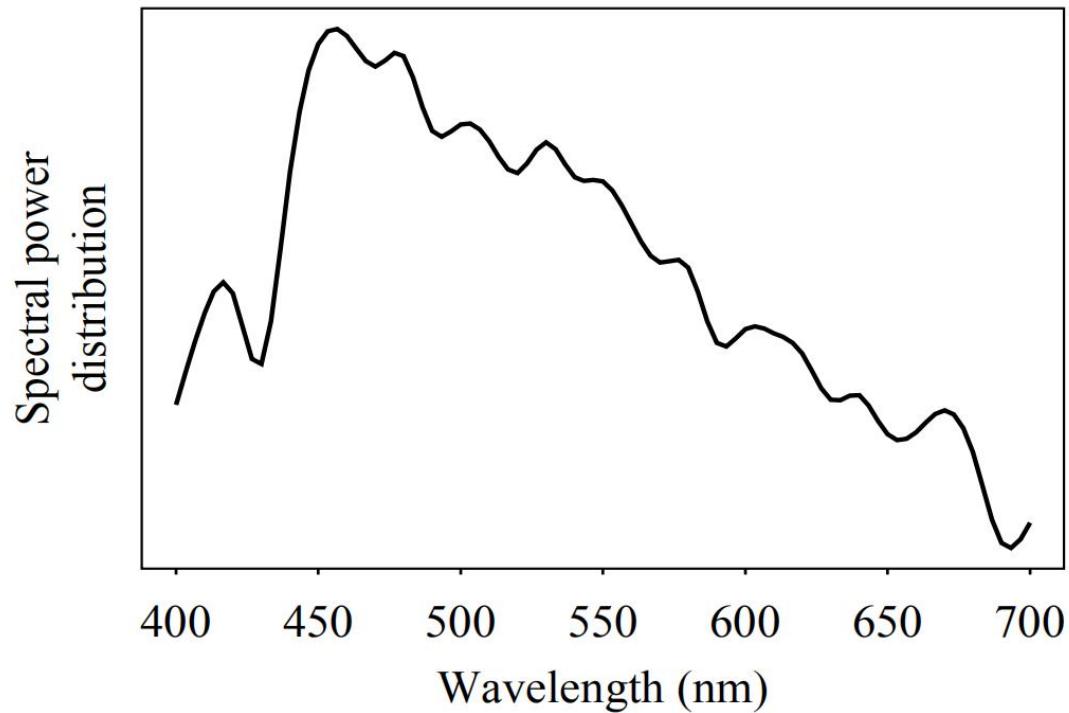


Fig. 4.2: Spectral power distribution of daylight.

## 5.3.1 人的视觉颜色感知

### 人眼视网膜细胞

- The achromatic channel produced by the cones is approximately proportional to  $2R + G + B/20$ .

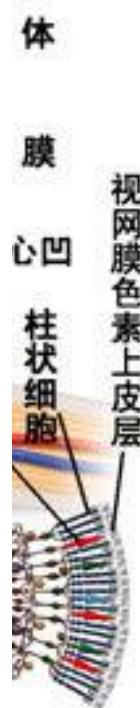
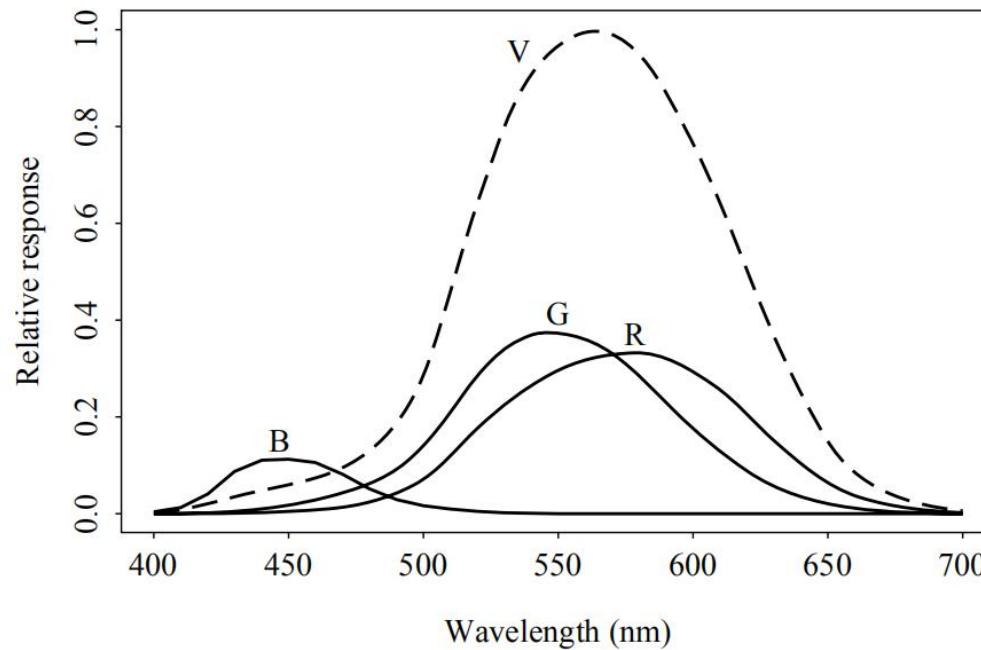
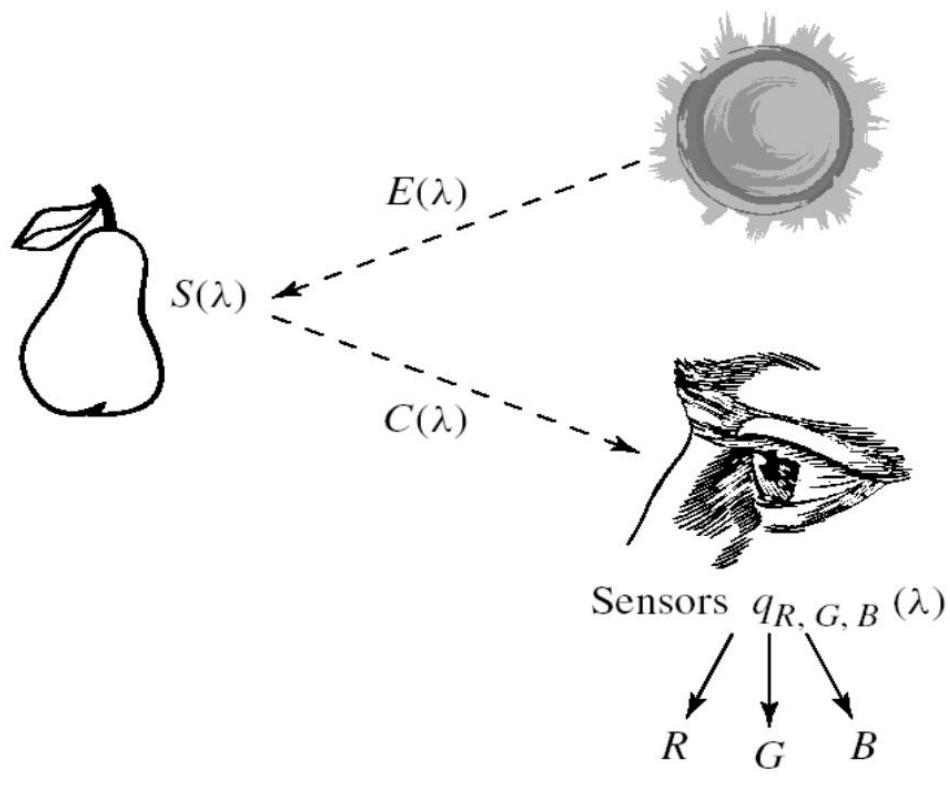


Fig. 4.3: R, G, and B cones, and Luminous Efficiency curve  $V(\lambda)$ .

## 5.3.1 人的视觉颜色感知

### 图像成像模型



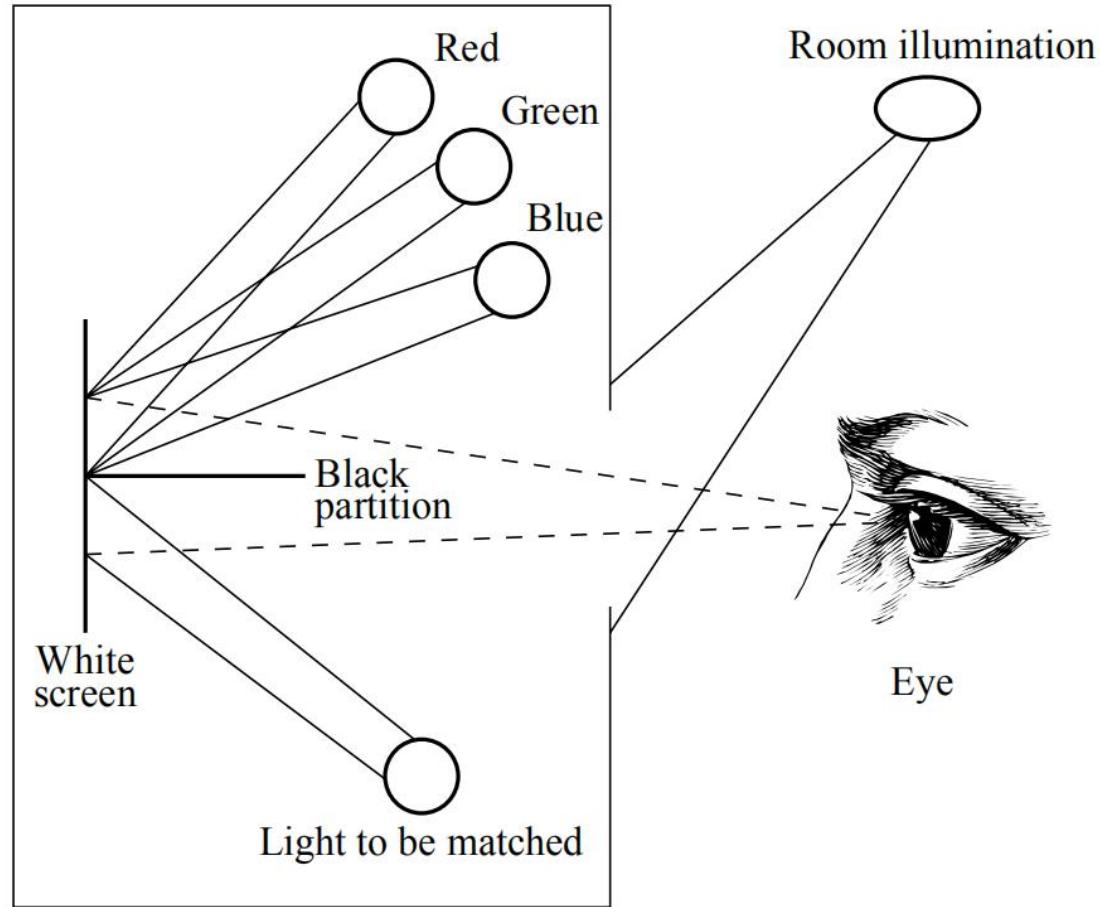
$$C(\lambda) = E(\lambda) S(\lambda).$$

Fig. 4.5: Image formation model.

## 5.3.2 颜色色度空间

### 色度计

用红色、绿色和蓝色合成任意颜色的光



## 5.3.2 颜色色度空间

### 用RGB合成任意颜色光

用红色、绿色和蓝色合成任意颜色的光，需要部分光为负值，因此红绿蓝并非最合适的三原色

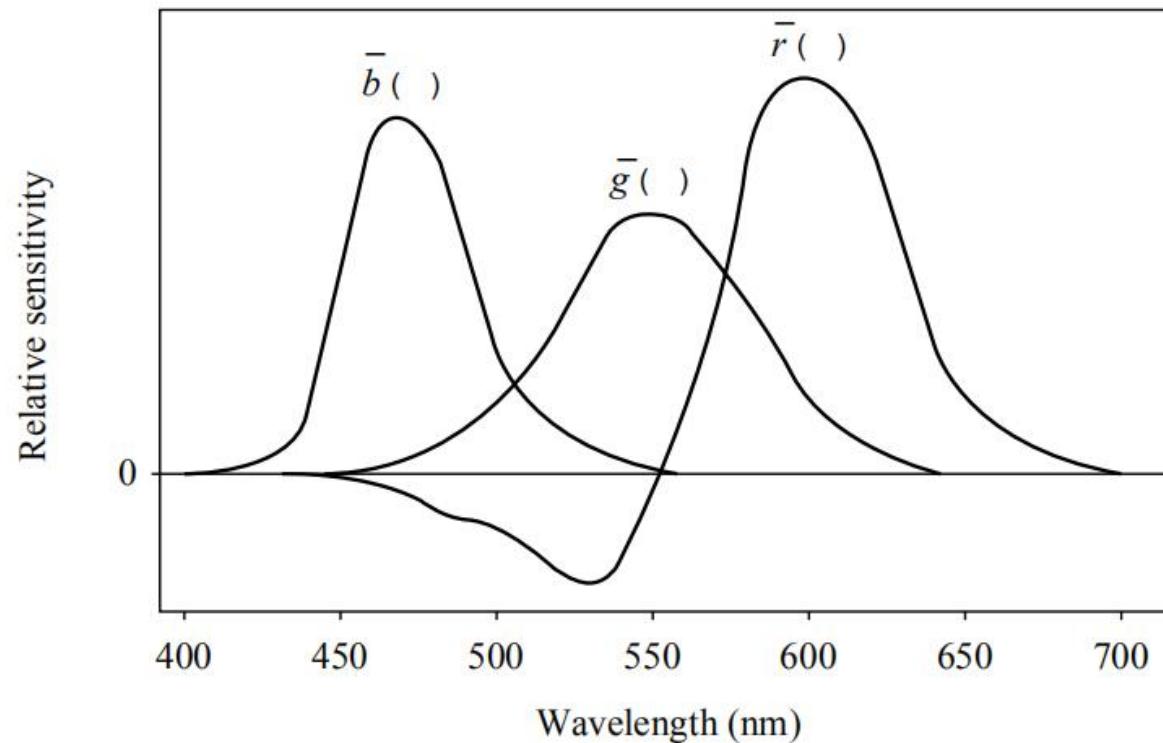


Fig. 4.9: CIE RGB color-matching functions  $\bar{r}(\lambda)$ ,  $\bar{g}(\lambda)$ ,  $\bar{b}(\lambda)$ .

## 5.3.2 颜色色度空间

### 新的颜色坐标系CIE-XYZ

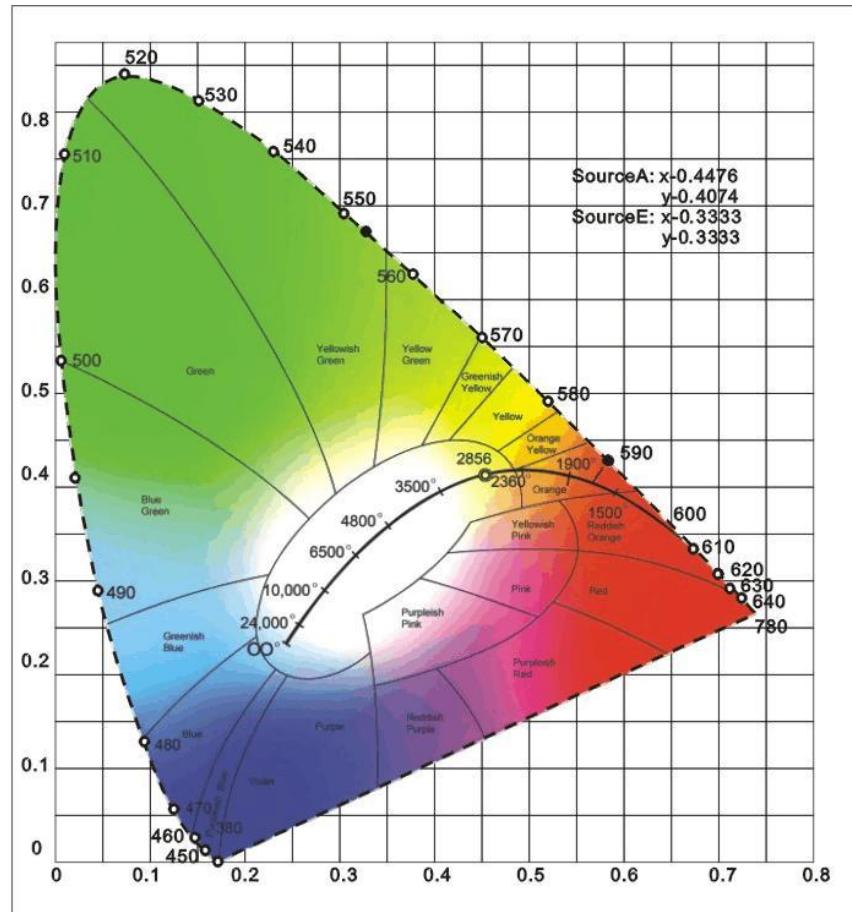
$$x = X/(X + Y + Z)$$

$$y = Y/(X + Y + Z)$$

$$z = Z/(X + Y + Z)$$

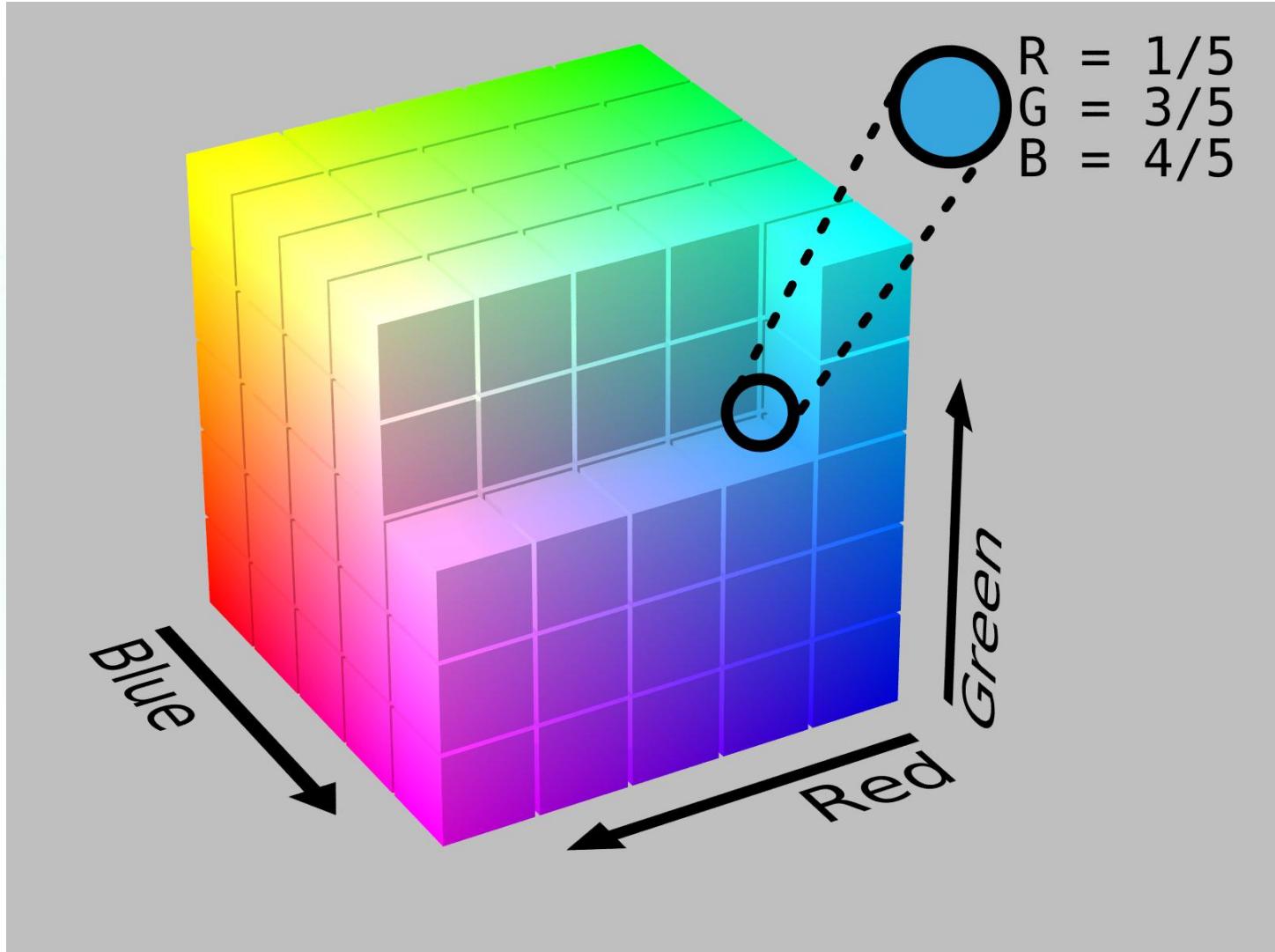
$$x + y + z = \frac{X + Y + Z}{X + Y + Z} \equiv 1$$

$$z = 1 - x - y$$



## 5.3.3 常用颜色空间

### RGB空间

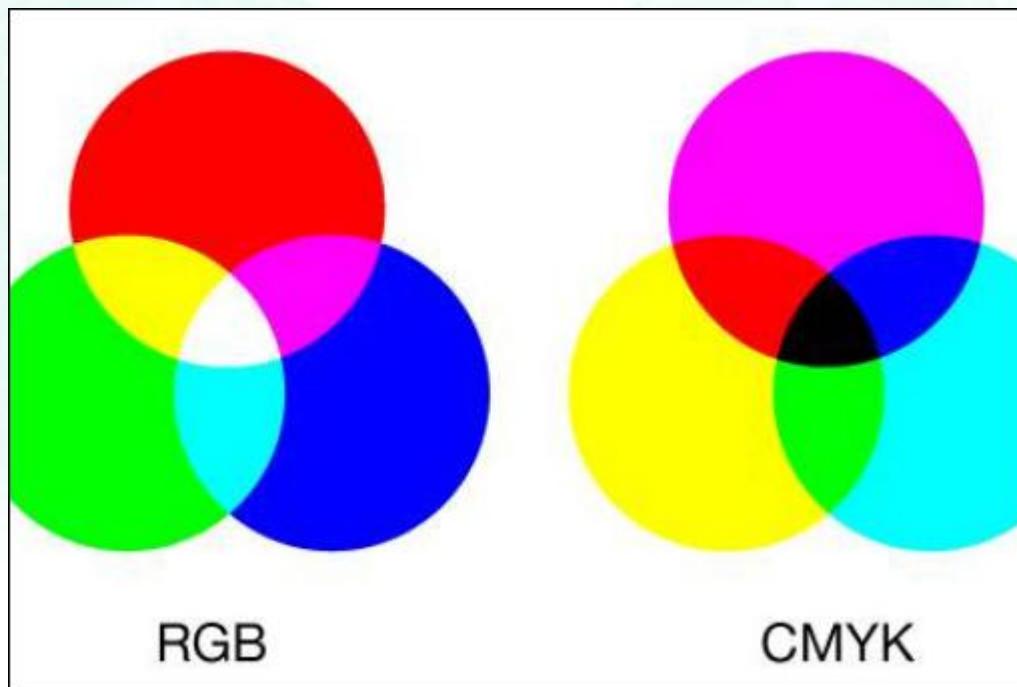


## 5.3.3 常用颜色空间

### 打印机颜色模型CMYK

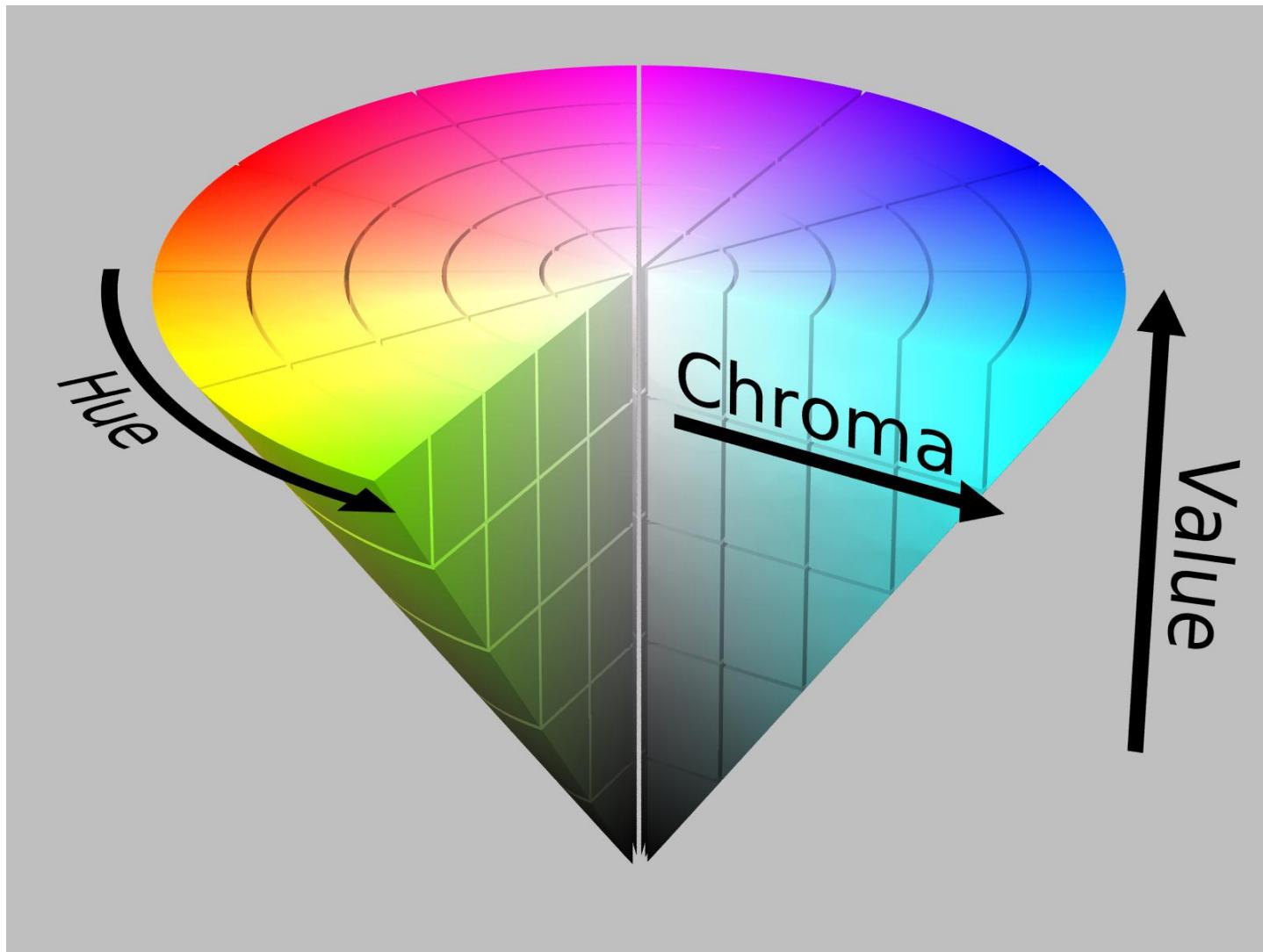


问题：RGB和CMYK哪些颜色模型是加性颜色模型？那个是减性颜色模型？



## 5.3.3 常用颜色空间

### HSV





## 5.3.3 常用颜色空间

问题：计算机常用的颜色空间？符合人眼视觉的颜色空间？

RGB	HSV	颜色名称
(255, 0, 255)	(300°, 100%, 100%)	<u>品红色</u>
(0, 0, 255)	(240°, 100%, 100%)	<u>蓝色</u>
(0, 255, 255)	(180°, 100%, 100%)	<u>青色</u>
(0, 255, 0)	(120°, 100%, 100%)	<u>绿色</u>
(255, 255, 0)	(60°, 100%, 100%)	<u>黄色</u>
(255, 0, 0)	(0°, 100%, 100%)	<u>红色</u>
(128, 0, 128)	(300°, 100%, 50%)	<u>紫色</u>
(0, 0, 128)	(240°, 100%, 50%)	<u>深蓝色</u>
(0, 128, 128)	(180°, 100%, 50%)	<u>鸭绿色</u>
(0, 128, 0)	(120°, 100%, 50%)	<u>深绿色</u>
(128, 128, 0)	(60°, 100%, 50%)	<u>橄榄色</u>
(128, 0, 0)	(0°, 100%, 50%)	<u>栗色</u>
(0, 0, 0)	(0°, 0%, 0%)	<u>黑色</u>
(128, 128, 128)	(0°, 0%, 50%)	<u>灰色</u>
(192, 192, 192)	(0°, 0%, 75%)	<u>银色</u>
(255, 255, 255)	(0°, 0%, 100%)	<u>白色</u>

## 颜色空间转换



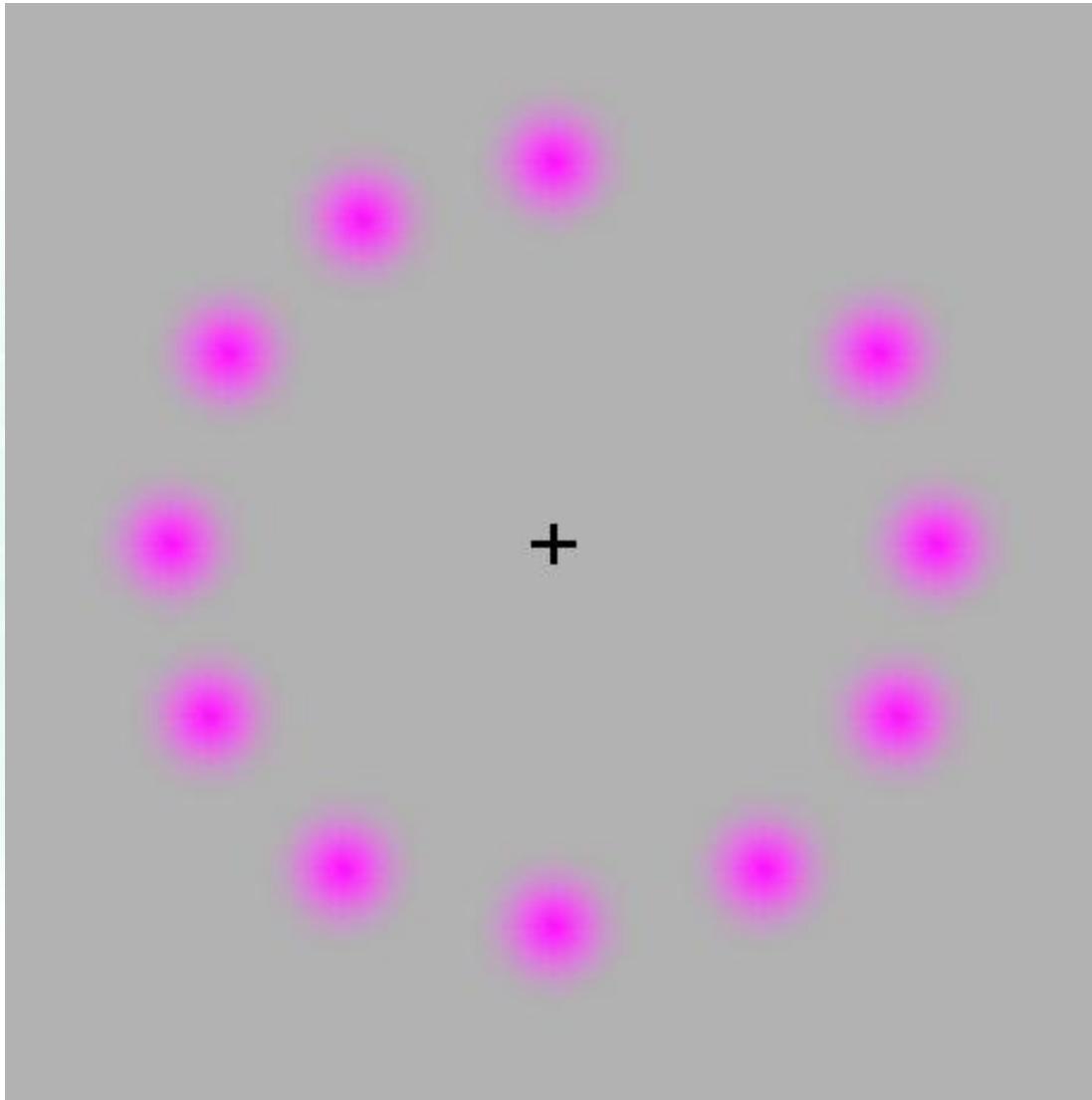
# 追逐丁香视错觉(Lilac Chaser Illusion)

- 又名吃豆子视错觉(Pac-Man Illusion) 是由 Jeremy Hinton于2005年创作的一幅经典视错觉图像。
- 它由12帧静止画面组成，每两帧之间的时间间隔约为0.1秒，每一帧画面中心黑色十字的周围都环绕着11个品红色原点。
- 周围原点总数应该为12个，但是每一帧都缺失1个原点，并且缺失的这1个原点的位置在每一帧中的位置是按顺时针排列的。





# 追逐丁香视错觉 (Lilac Chaser Illusion)





# 视觉暂留、视觉后遗像和特克斯勒消逝效应

- 一个与背景颜色相同颜色的空缺圆点在丁香圆环上顺时针依次出现；
- 这个空缺的圆点变成了一个绿色的圆点在丁香环上顺时针旋转；
- 品红色圆点慢慢消失，不久之后整个图像中只剩下中央的黑色十字、灰色的背景和一个独自旋转的绿色圆点。

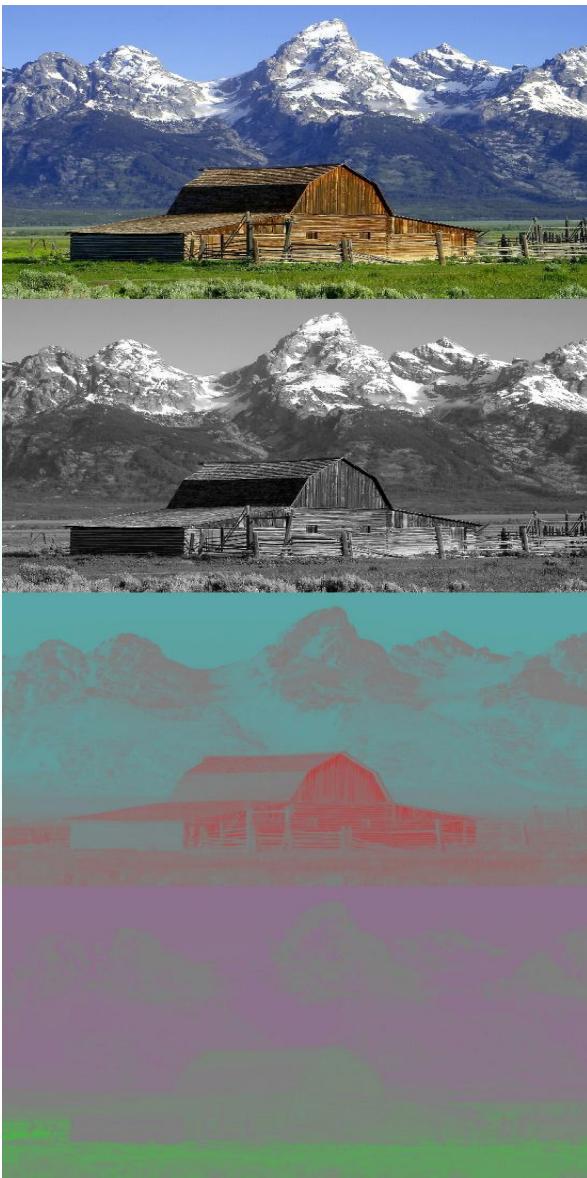
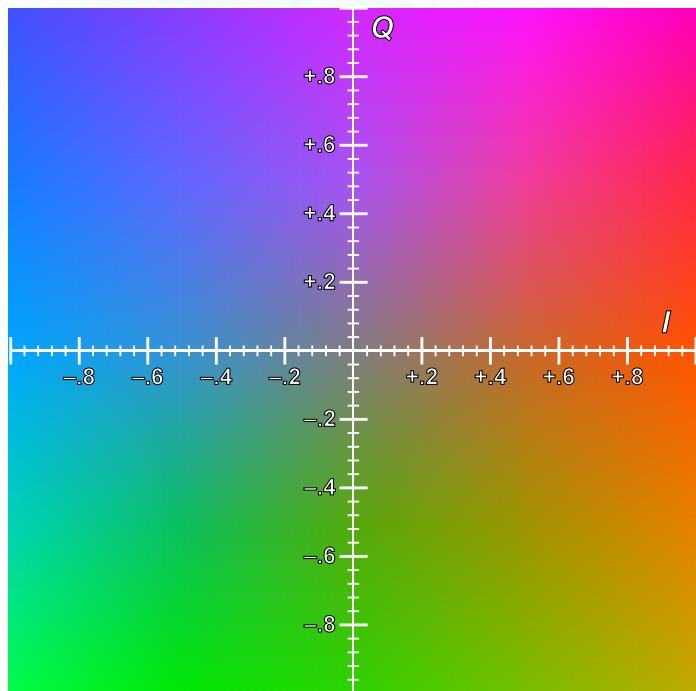


# 视觉暂留、视觉后遗像和特克斯勒消逝效应

- 但当物体移去时，视神经对物体的印象不会立即消失，而要延续0.1—0.4秒的时间。
- 由于部分视觉神经因长时间接收同样的刺激而变得“疲劳”，我们就会看到一个与原图“互补”的图像
- 即当一个人的目光聚焦在某个固定点上20秒或者更长时间之后，在该固定点周围，也就是在观察者余光中的其他视觉刺激源将会在观察者的视野中慢慢淡化直至最后消失。外

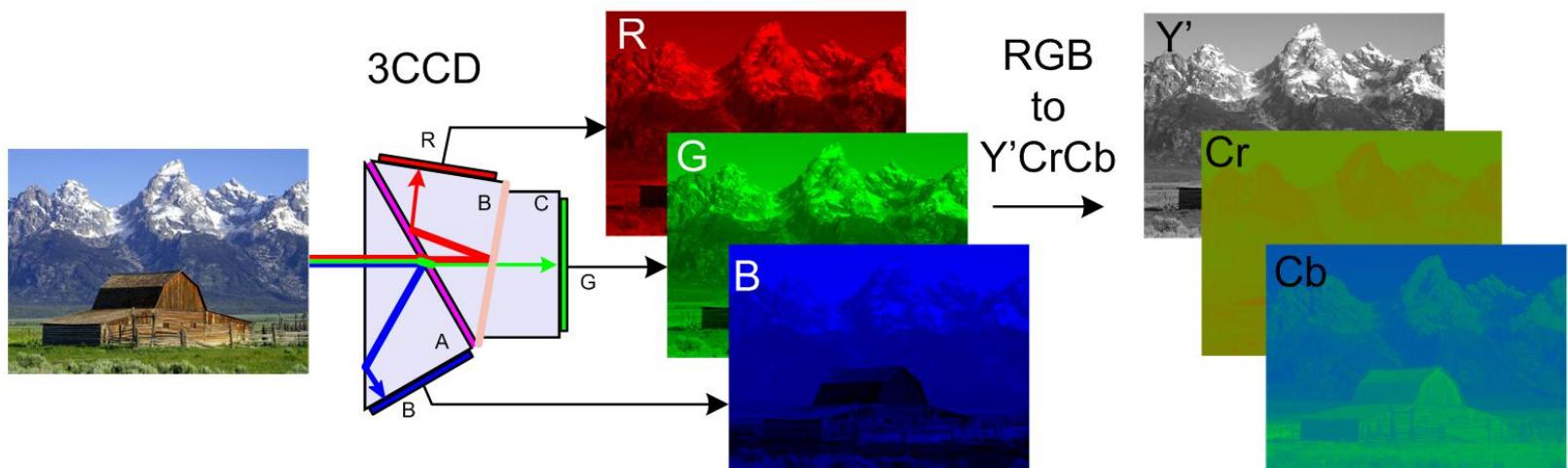
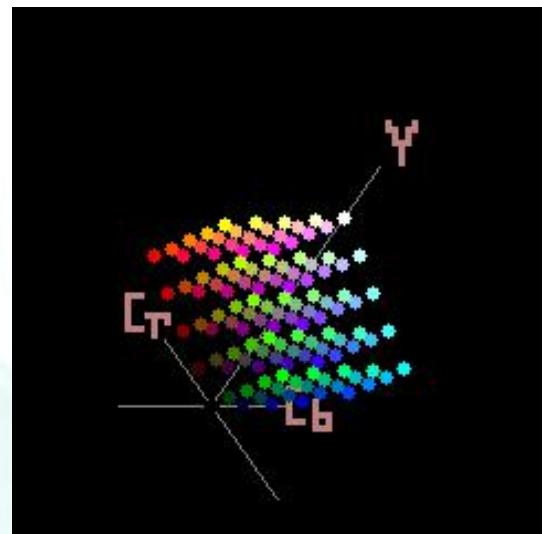
## 5.3.4 数字视频

### 视频颜色模型YIQ



## 5.3.4 数字视频

### 视频颜色YCbCr



## 5.3.4 数字视频

### 色度的二次采样

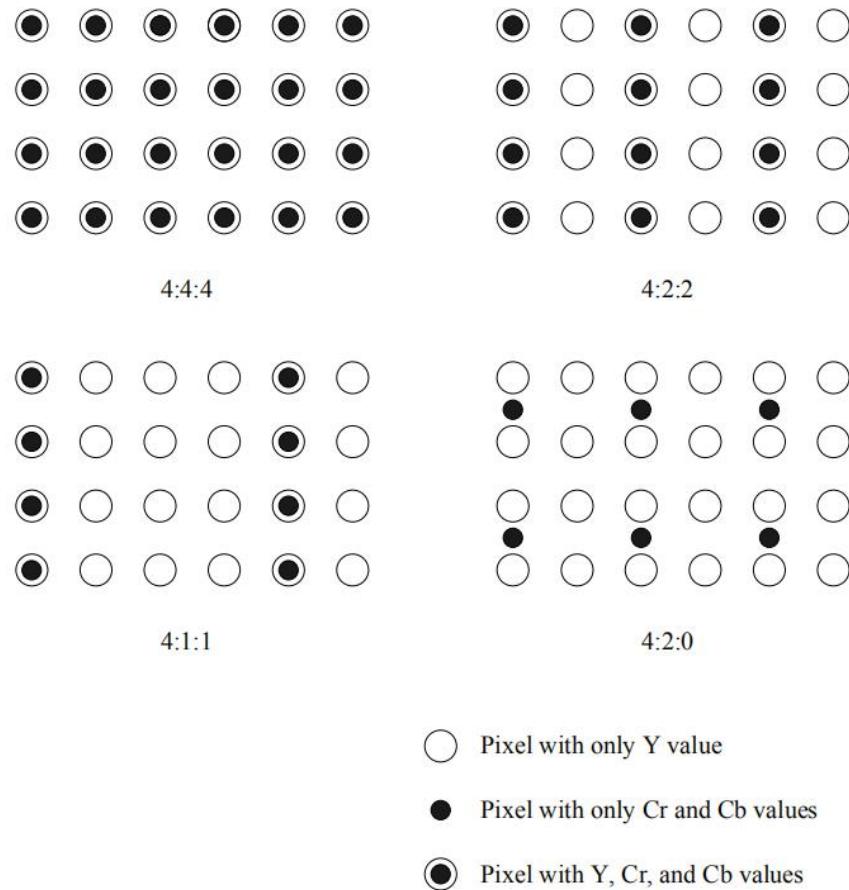


Fig. 5.6: Chroma subsampling.



## 5.3.4 数字视频

### 格式与数据量

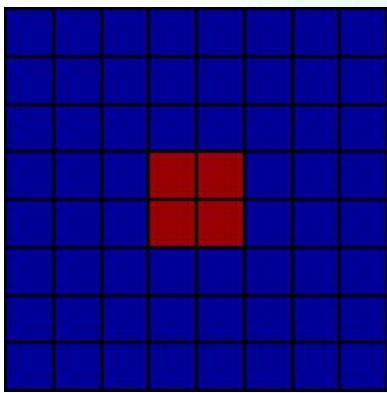
Table 5.3: Digital video specifications

	CCIR 601 525/60 NTSC	CCIR 601 625/50 PAL/SECAM	CIF	QCIF
Luminance resolution	720 × 480	720 × 576	352 × 288	176 × 144
Chrominance resolution	360 × 480	360 × 576	176 × 144	88 × 72
Color Subsampling	4:2:2	4:2:2	4:2:0	4:2:0
Aspect Ratio	4:3	4:3	4:3	4:3
Fields/sec	60	50	30	30
Interlaced	Yes	Yes	No	No

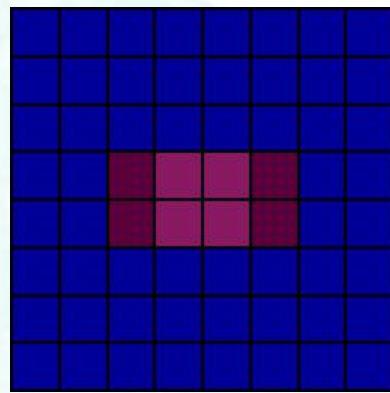
## 5.3.4 数字视频

### 压缩失真

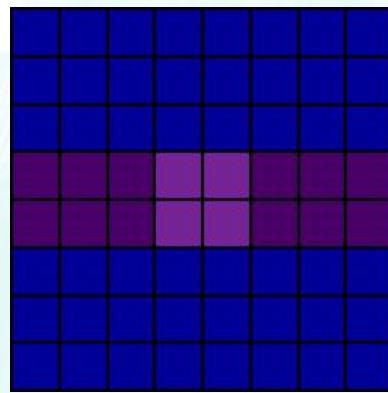
原始4:4:4



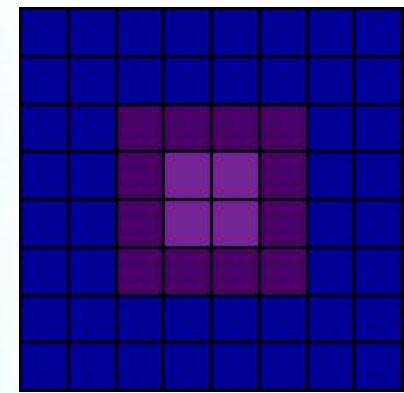
色度取样4:2:2



色度取样4:1:1



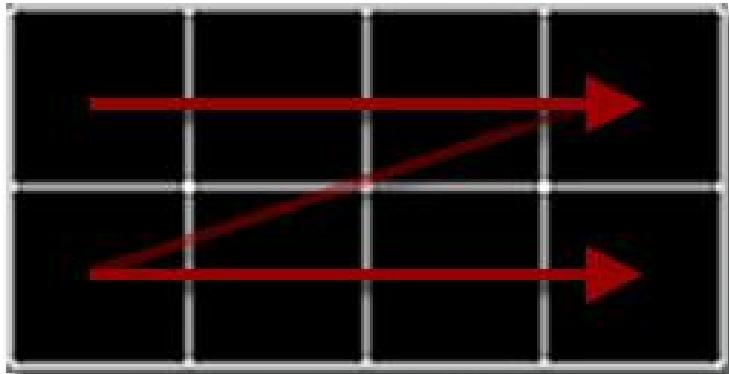
色度取样4:2:0



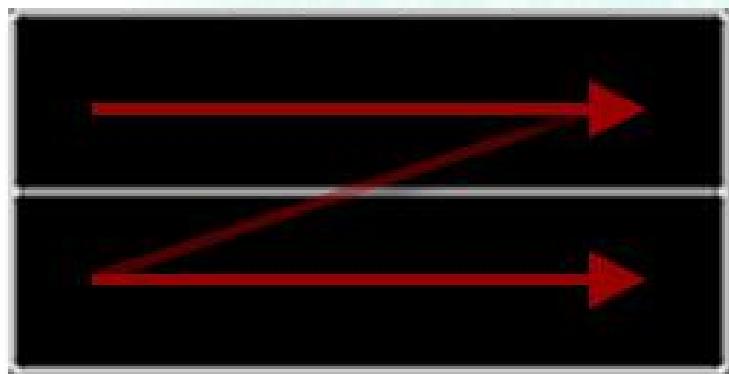
## 5.3.4 数字视频

### 压缩失真

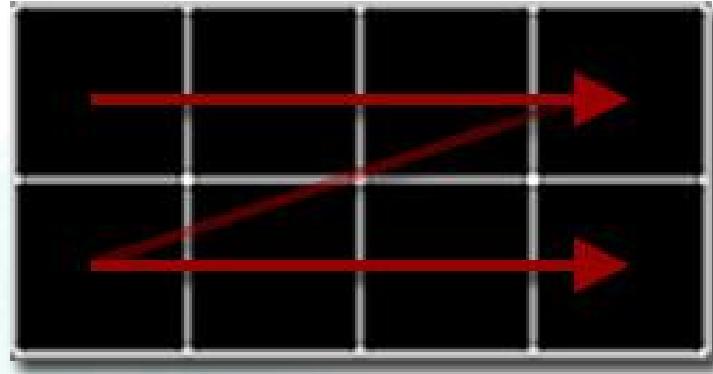
亮度分辨率



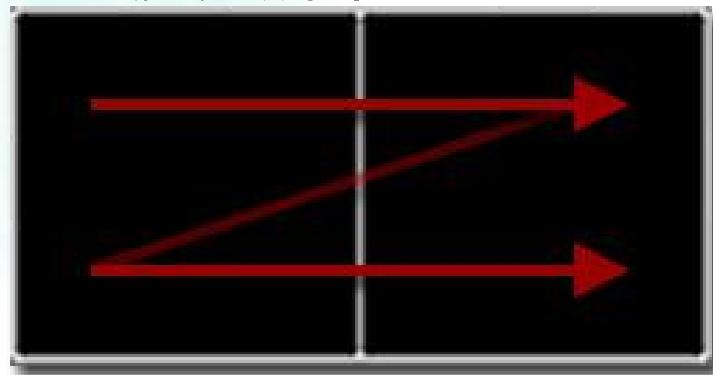
色度分辨率4:1:1



色度分辨率4:4:4



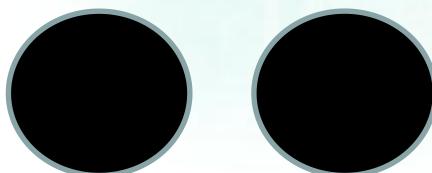
色度分辨率4:2:0



## 5.3.4 数字视频

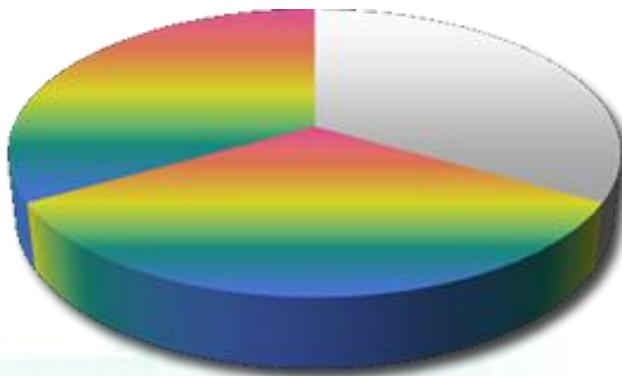
### 人眼分辨率

- ①当照度太强、太弱时或当背景亮度太强时，人眼分辨率降低。
- ②当视觉目标运动速度加快时，人眼分辨率降低。
- ③人眼对彩色细节的分辨率比对亮度细节的分辨率要差，如果黑白分辨率为1，则黑红为0.4，绿蓝为0.19

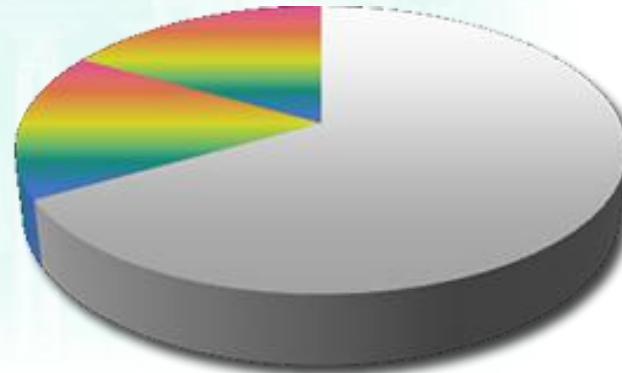


## 5.3.4 数字视频

原始图像



色度取样后



## 5.3.4 数字视频

### 模拟视频接口



**Fig. 5.7** Connectors for typical analog display interfaces. From left to right: Component video, Composite video, S-video, and VGA

## 5.3.4 数字视频

### 数字视频接口

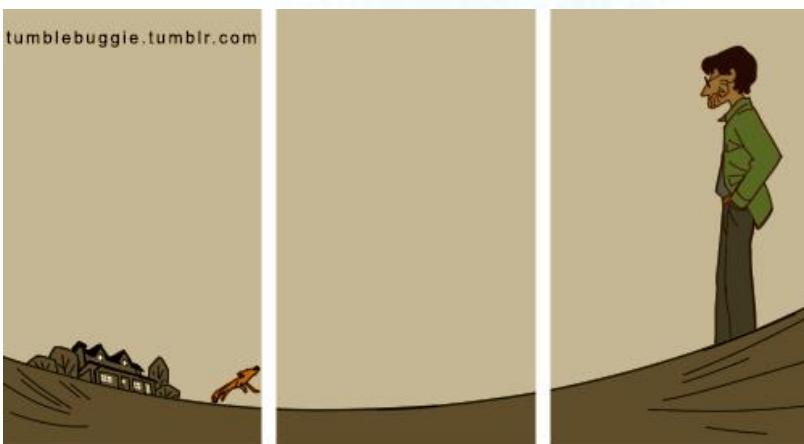


**Fig. 5.8** Connectors of different digital display interfaces. From left to right: DVI, HDMI, DisplayPort

## 5.3.4 数字视频

### 立体视频

tumblebuggie.tumblr.com



At night I leave the lights on in my little house and walks across the flat fields. When I look back from a distance the house is like a boat on the sea. It's really the only time I feel safe.

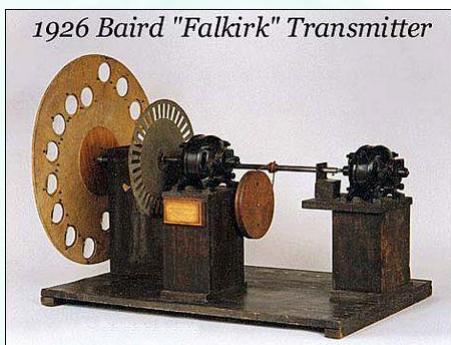


## 5.3.4 数字视频

### 电视



贝尔德 (John Logie Baird, 1888-1946) 英国发明家。被誉为“电视之父”。



1928年通用电气公司制造的这个可爱小木柜被公认为最早的声音交互设备。

1939 RCA TRK-9



1939年RCA产电视机9英寸的显示屏已属大屏幕。

## 5.3.4 数字视频

### 电视



1954年，美国得克萨斯仪器公司研制出第一台全晶体管电视接收机。

1958年，我国第一台黑白电视机北京牌14英寸黑白电视机在天津712厂诞生。

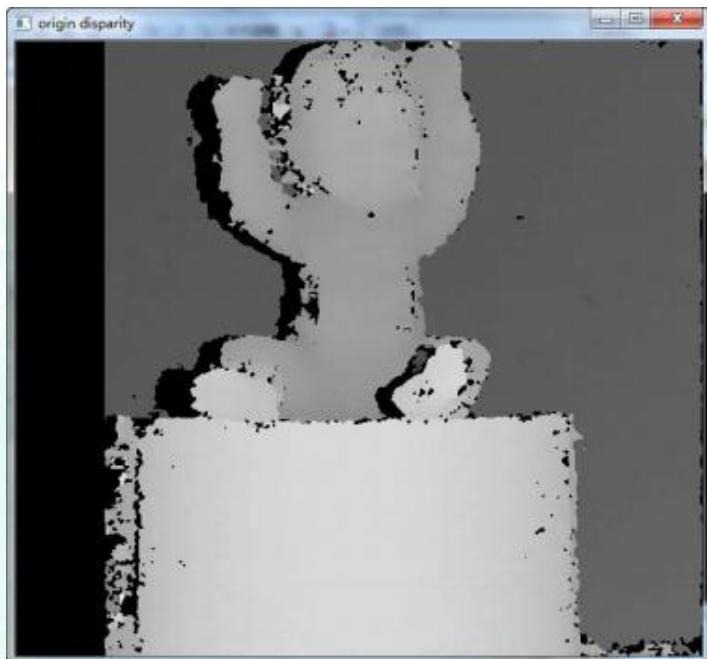
## 5.3.4 数字视频

### 电视



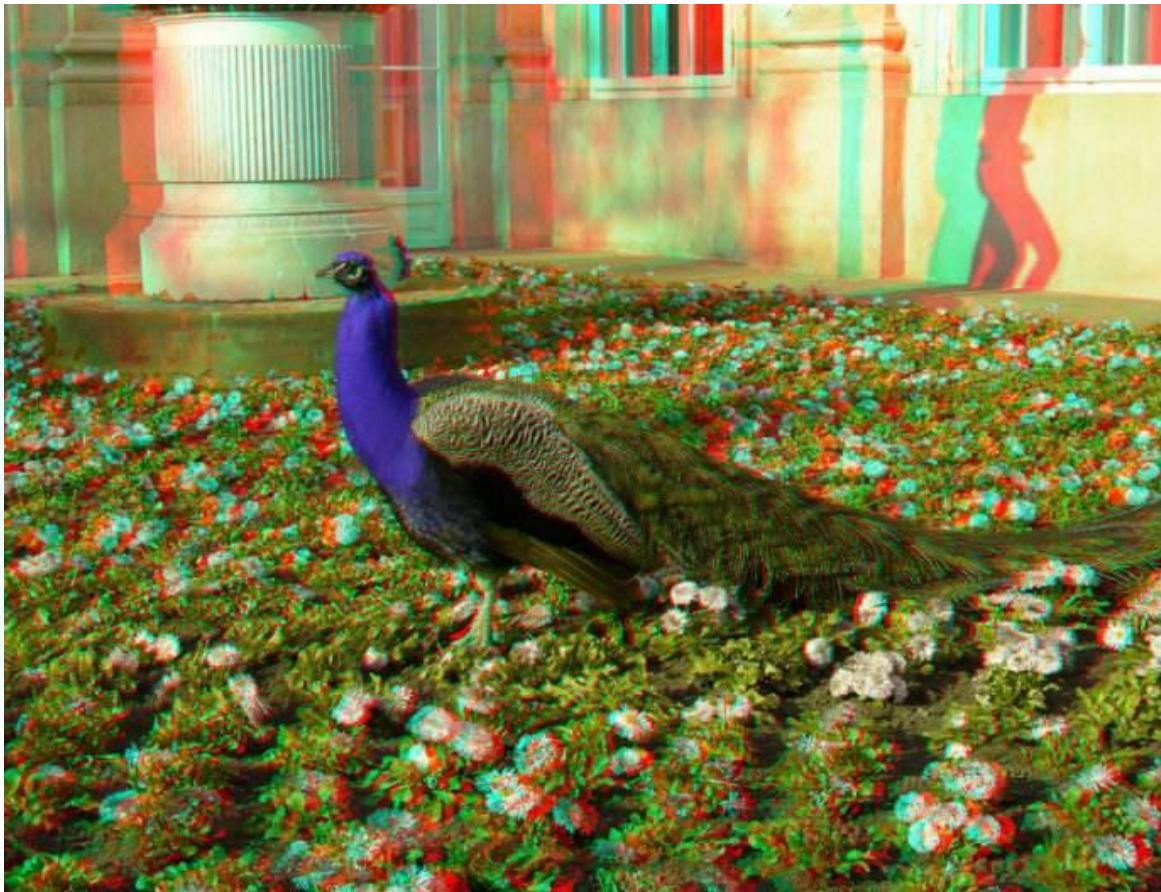
## 5.3.4 数字视频

### 双目立体视觉



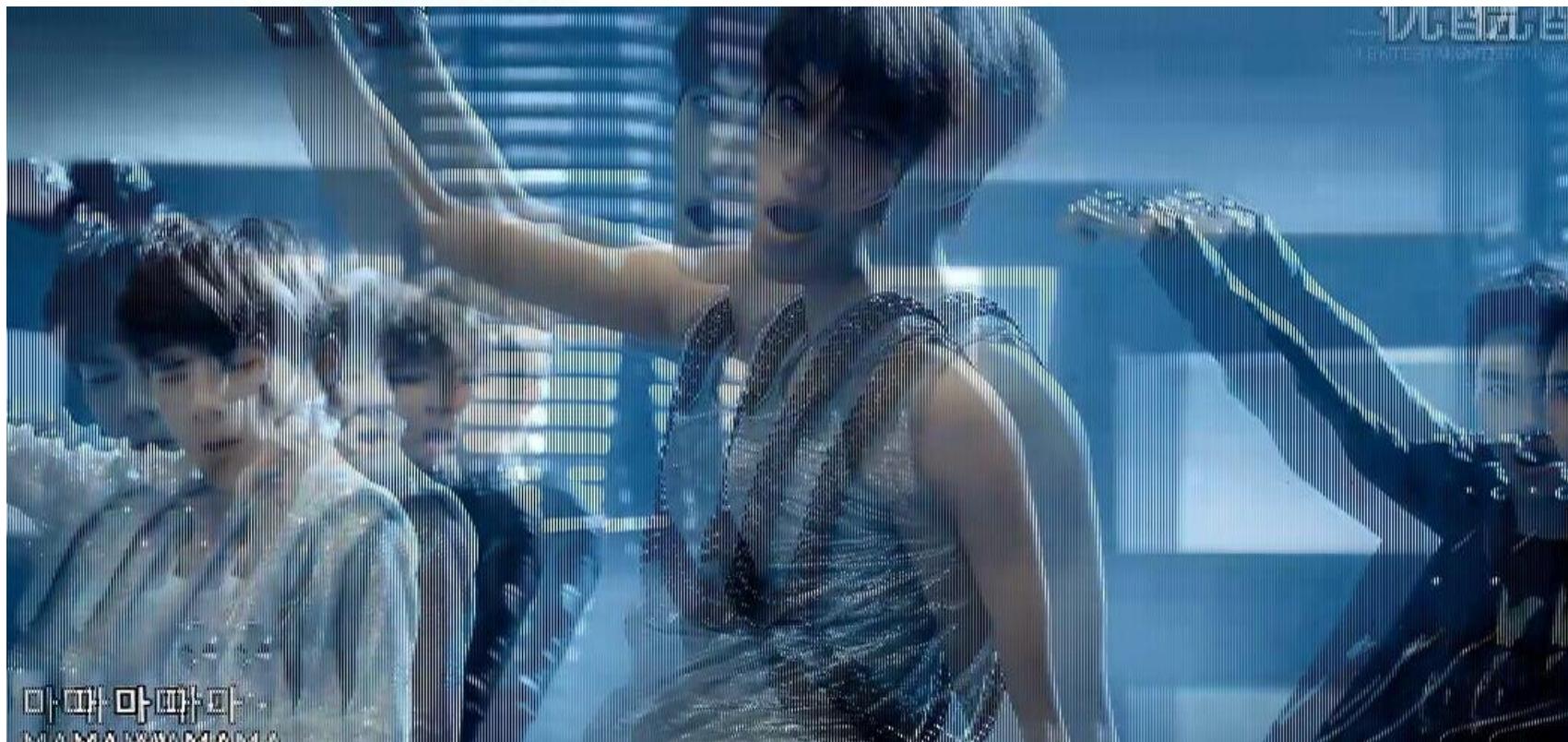
## 5. 3. 4 数字视频

### 红蓝3D



## 5. 3. 4 数字视频

### 偏振3D



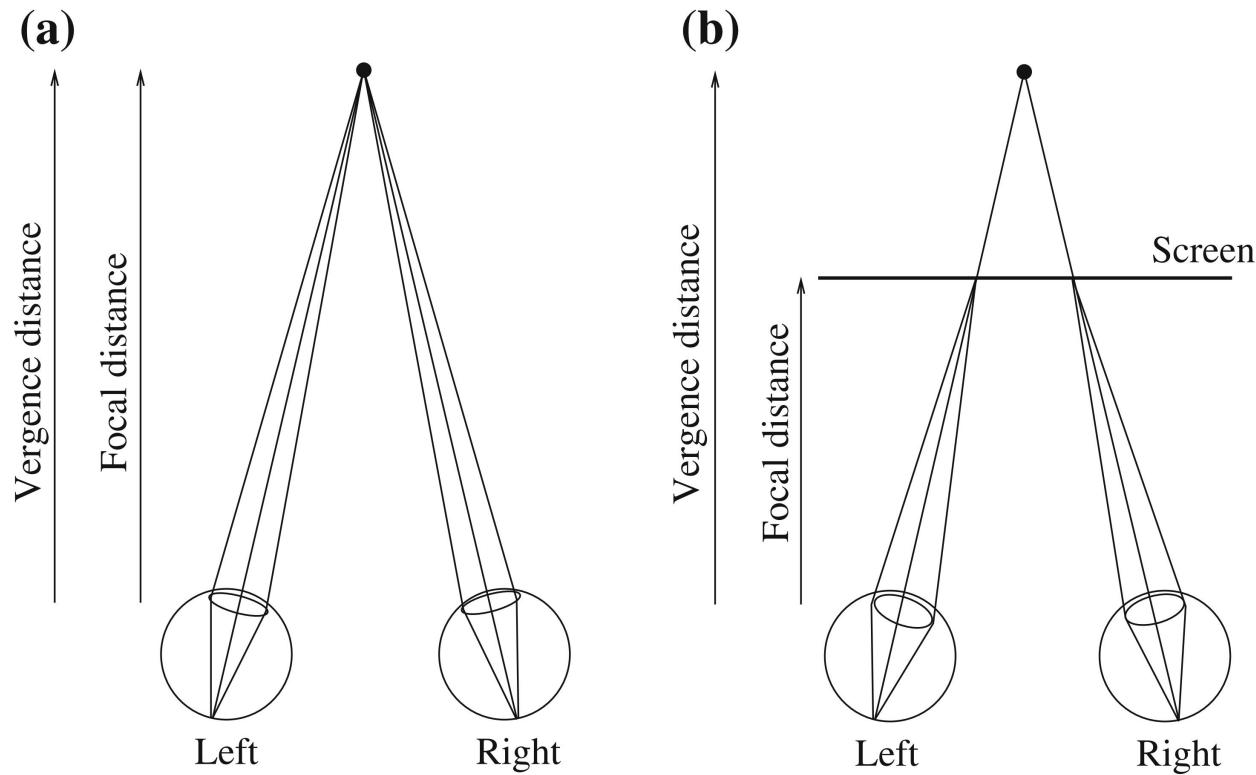
## 5. 3. 4 数字视频

### Shutter 眼镜3D



## 5.3.4 数字视频

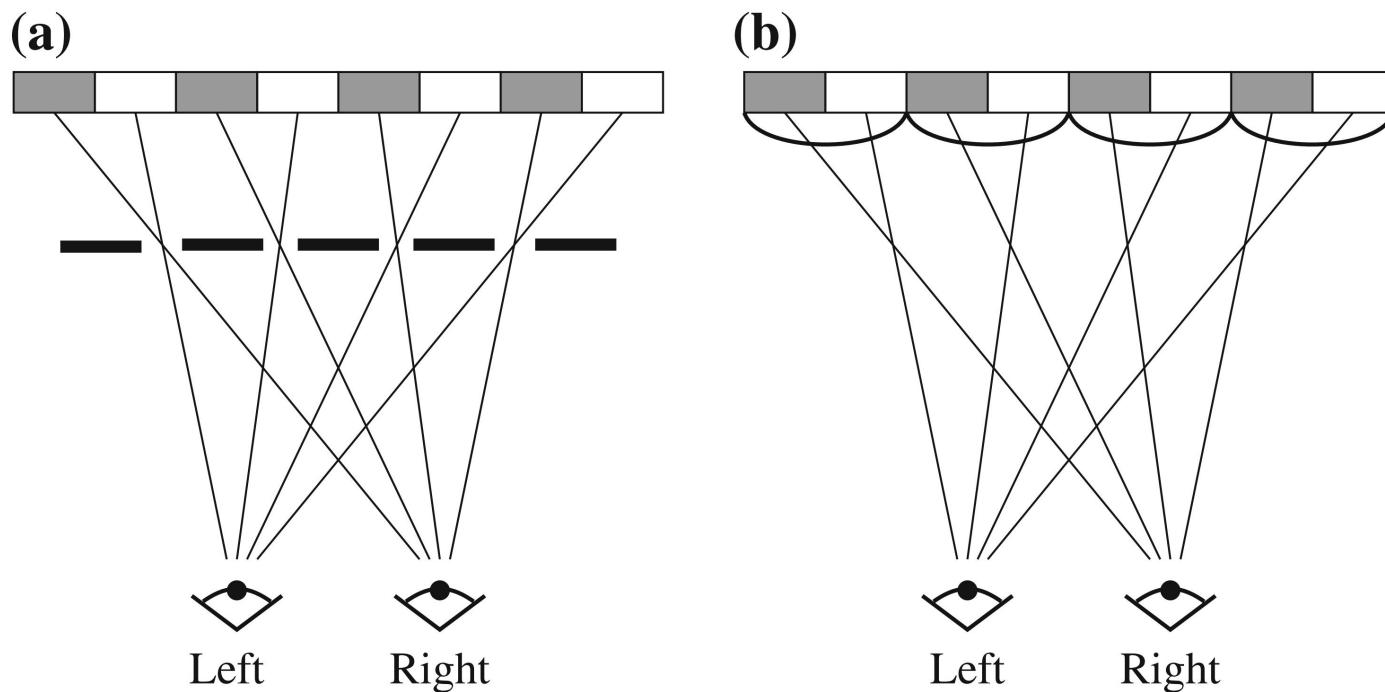
### 3D显示问题



**Fig. 5.9** The Vergence-Accommodation Conflict. **a** Real World and **b** 3D Display

## 5.3.4 数字视频

### 3D显示技术



**Fig. 5.10** Autostereoscopic display devices. **a** Parallax Barrier and **b** Lenticular Lens

## 5.3.4 数字视频

### 裸眼3D



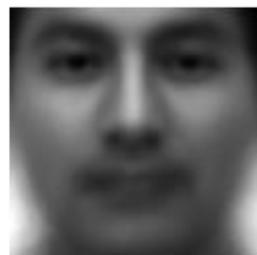


## 5.4 视觉特征的表示

## 5.4.1 基于图像像素值的主成分分析

PCA(主成分分析)算法用于人脸识别

得到的均值图像mean\_face



前8个最大元对应的“特征脸”





## 5.4.1 基于图像像素值的主成分分析

### 人脸图像PCA代码讲解

```
from sklearn.decomposition import PCA
```

导入模块

```
PCA(n_components=n_components, svd_solver='auto',  
whiten=True).fit(all_data_set)
```

**n\_components**: 这个参数可以帮我们指定希望PCA降维后的特征维度数目

**whiten** : 判断是否进行白化。所谓白化，就是对降维后的数据的每个特征进行归一化，让方差都为1

**svd\_solver**: 即指定奇异值分解SVD的方法，由于特征分解是奇异值分解SVD的一个特例，一般的PCA库都是基于SVD实现的。有4个可以选择的值：{'auto', 'full', 'arpack', 'randomized'}

**all\_data\_set**: 训练的数据集



## 5.4.1 基于图像像素值的主成分分析

### 人脸图像PCA代码讲解

```
eigenfaces =  
pca.components_.reshape((n_components, 100, 80))
```

获取特征脸，把特征脸转换成100\*80大小





## 5.4.2 图像频域分析

### 二维DFT傅里叶变换

- 一个图像尺寸为 $M \times N$ 的函数 $f(x,y)$ 的离散傅立叶变换 $F(u,v)$ :

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

- $F(u, v)$ 的反变换:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$$

## 5.4.2 图像频域分析

频谱图中暗的点数更多，那么实际图像是比较柔和的，反之，如果频谱图中亮的点数多，那么实际图像一定是尖锐的，边界分明且边界两边像素差异较大的





## 5.4.2 图像频域分析

### 图像频域变换代码讲解

```
f = np.fft.fft2(img)
```

用python的numpy库实现图像的快速傅里叶变换，输入是numpy类型的图片

```
fshift = np.fft.fftshift(f)
```

将低频移动到图像的中心

```
magnitude_spectrum = 20 * np.log(np.abs(fshift))
```

先取绝对值，表示取模。取对数，将数据范围变小

## 5.4.3 图像词袋模型

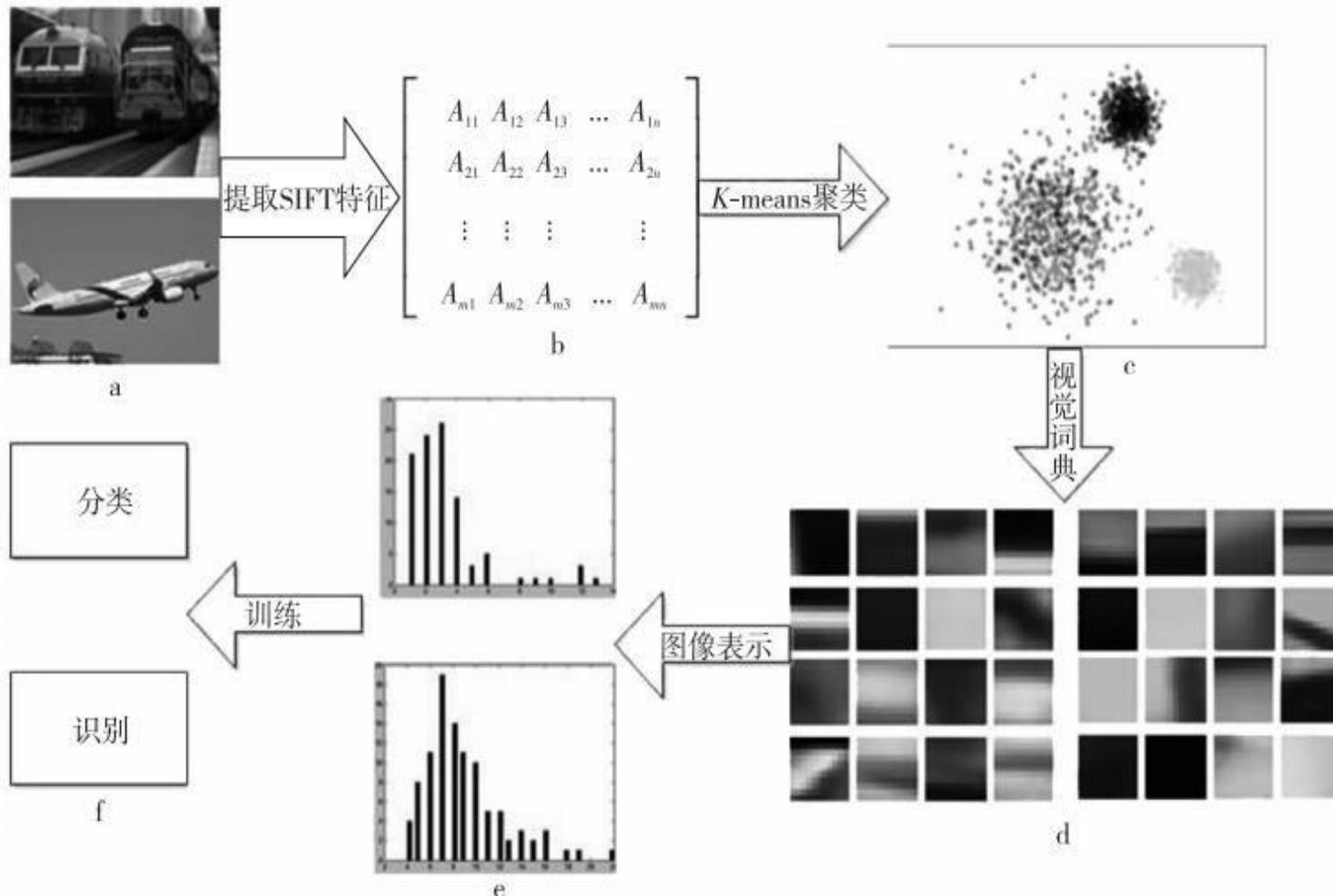


Figure 1 Framework of the BOW model

图 1 词袋模型框架



## 5. 4. 3 图像词袋模型

### 图像词袋模型代码讲解

```
def makecodebook():
```

该函数目的是：训练一个模型，建立词袋库，保存成./bow\_codebook.pkl文件，供测试使用

```
print("SIFT feature extraction")
x_train = [extract_sift_descriptors(img) for img
in x_train]
```

提取每个图片的SIFT特征





## 5.4.2 图像词袋模型

### 图像词袋模型代码讲解

```
print("Building the codebook, it will take some time")
codebook = build_codebook(x_train,
voc_size=VOC_SIZE)
```

利用k-mean算法建立图像词袋

