

阿洲的程式教學

關於Qt、OpenCV、影像處理演算法

色彩空間轉換(cvtColor)

OpenCV的cvtColor()讓影像在不同色彩空間之中轉換，由於OpenCV從外界讀入圖檔時，比如使用imread()讀取JPEG檔時，Mat內存為BGR而不是RGB格式，所以輸入參數通常使用CV_BGR2XXX、CV_XXX2BGR，代表從BGR色彩空間轉到其他色彩空間，或從其他色彩空間轉到BGR色彩空間，依此類推，CV_RGB2XXX代表從RGB色彩空間轉到其他色彩空間，通常較少使用。

內文索引 [\[隱藏\]](#)

[1 OpenCV 色彩空間轉換](#)

[2 BGR & Gray](#)

[3 BGR & YCrCb](#)

[4 BGR & HSV](#)

[5 BGR & HSL](#)

[6 BGR & CIE XYZ](#)

[7 BGR & CIE Lab*](#)

[8 BGR & CIE Luv*](#)

OpenCV 色彩空間轉換

cvtColor(const Mat& src, Mat& dst, int code)

- src：來源圖，支援 CV_8U、CV_16U、或 CV_32F位元深度，1、3、4通道的圖，有些色彩空間轉換無法在16位元上操作，使用前可參考文件。
- dst：目標圖，尺寸大小、深度會和來源圖相同。
- code：指定在何種色彩空間轉換，比如CV_BGR2GRAY、CV_GRAY2BGR、CV_BGR2HSV等。

BGR & Gray

參數：CV_BGR2GRAY、CV_RGB2GRAY、CV_GRAY2BGR、CV_GRAY2RGB

BGR to Gray： $Y=0.299R + 0.587G + 0.114*B$

Gray to BGR： $B=Y$ ， $G=Y$ ， $R=Y$

人眼對綠色的敏感感較大，而對藍色最小，因此綠色權重較大，藍色較小，上述公式為彩色轉灰階的標準。

BGR & YCrCb

參數：CV_BGR2YCrCb、CV_RGB2YCrCb、CV_YCrCb2BGR、CV_YCrCb2RGB

BGR to YCrCb： $Y=0.299R + 0.587G + 0.114B$ ， $Cr=(R-Y)0.713 + \text{delta}$ ， $Cb=(B-Y)*0.564 + \text{delta}$

YCrCb to BGR： $R=Y + 1.403(Cr-\text{delta})$ ， $G=Y - 0.714(Cr-\text{delta}) - 0.344(Cb-\text{delta})$ ， $B=Y + 1.773(Cb-\text{delta})$

delta：8位元->128，16位元->32768，浮點數->0.5

其中 Y 是亮度(Luminance)，Cb、Cr 是色差(chrominance)，Cb是藍色色差，Cr是紅色色差，雖然RGB與YCbCr都為三個通道無法節省儲存空間，但實際上可利用人類視覺對亮度比較敏感，而對彩度比較不敏感的特質來減少內存，也就是減少Cb、Cr的取樣個數。取樣格式有4：2：0、4：2：2、4：4：4三種，4：2：0格式代表每2×2的4個像素中，Y會對每個像素取樣，而色差CbCr僅會在第一行兩個像素的中間取樣，因此8位元情況下，原本每個像素需要3byte儲存空間，4個像素需要12個byte，現在只要6個byte，減少了一半的內存，現今像Jpeg、Mpeg4等影像格式，都是利用YCbCr去壓縮的。

BGR & HSV

HSV(hue、saturation、value)用來表示色相、飽和度和明度，這種系統比RGB更接近人對色彩的感知，色相決定顏色的本質，像我們說紅、澄、黃就是指一種色相，飽和度是指顏色的深淺比例，顏色越深飽和度越高，白色所占比例越高，飽和度越低，明度表示顏色的明暗程度，數值越大越亮。

參數：CV_BGR2HSV、CV_RGB2HSV、CV_HSV2BGR、CV_HSV2RGB

BGR to HSV：當我們計算8位元或16位元圖時，R、G、B都轉成0到1之間的浮點數，比如 $R=R/(R+G+B)$ ，接著用下列公式進行轉換，分別得到H、S、V的值。

$$V \leftarrow \max(R, G, B)$$

$$S \leftarrow \begin{cases} \frac{V - \min(R, G, B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

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

如果 $H < 0$ ，則 $H = H + 360$ ，讓H範圍0到360，S範圍0到1，V範圍0到1，接著根據圖的形態，轉換到合適的範圍。8位元： $H = H/2$ ， $S = S/255$ ， $V = V/255$ 。16位元： $H = H$ ， $S = S/65535$ ， $V = V/65535$ 。浮點數圖：維持不變。

BGR & HSL

HSL(hue、saturation、lightness)用來表示色相、飽和度和亮度，這種系統比RGB更接近人對色彩的感知，色相決定顏色的本質，像我們說紅、澄、黃就是指一種色相，飽和度是指顏色的深淺比例，顏色越深飽和度越高，白色所占比例越高，飽和度越低，亮度表示顏色的明暗程度，數值越大越亮。

參數：CV_BGR2HLS、CV_RGB2HLS, CV_HLS2BGR, CV_HLS2RGB

BGR to HSV：當我們計算8位元或16位元圖時，R、G、B都轉成0到1之間的浮點數，比如 $R=R/(R+G+B)$ ，接著用下列公式進行轉換，分別得到H、S、L的值。

$$V_{\max} \leftarrow \max(R, G, B)$$

$$V_{\min} \leftarrow \min(R, G, B)$$

$$L \leftarrow \frac{V_{\max} + V_{\min}}{2}$$

$$S \leftarrow \begin{cases} \frac{V_{\max} - V_{\min}}{V_{\max} + V_{\min}} & \text{if } L < 0.5 \\ \frac{V_{\max} - V_{\min}}{2 - (V_{\max} + V_{\min})} & \text{if } L \geq 0.5 \end{cases}$$

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

如果 $H < 0$ ，則 $H = H + 360$ ，讓 H 範圍0到360， S 範圍0到1， L 範圍0到1，接著根據圖的形態，轉換到合適的範圍。8位元： $H = H/2$ ， $S = S/255$ ， $L = L/255$ 。16位元： $H = H$ ， $S = S/65535$ ， $L = L/65535$ 。浮點數圖：維持不變

BGR & CIE XYZ

在CIE XYZ色彩空間中，有一組X、Y和Z的值，對應於紅色、綠色和藍色，並不是真的紅藍綠，而是使用匹配函數來計算出來。

參數：CV_BGR2XYZ、CV_RGB2XYZ, CV_XYZ2BGR, CV_XYZ2RGB

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \leftarrow \begin{bmatrix} 3.240479 & -1.53715 & -0.498535 \\ -0.969256 & 1.875991 & 0.041556 \\ 0.055648 & -0.204043 & 1.057311 \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

BGR & CIE Lab*

參數：CV_BGR2Lab、CV_RGB2Lab、CV_Lab2BGR、CV_Lab2RGB

Lab色彩空間是顏色-對立空間，帶有維度L表示亮度，a和b表示顏色對立維度，基於非線性壓縮的CIE XYZ色彩空間坐標，Lab顏色被設計來接近人類視覺，致力於感知均勻性，L分量密切匹配人類亮度感知，因此可以被用來通過修改a和b分量的輸出色階來做精確的顏色平衡，或使用L分量來調整亮度對比。當我們計算8位元或16位元圖時，R、G、B都轉成0到1之間的浮點數，比如 $R = R/(R+G+B)$ ，接著用下列公式進行轉換，分別得到L、a、b的值。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$X \leftarrow X/X_n, \text{ where } X_n = 0.950456$$

$$Z \leftarrow Z/Z_n, \text{ where } Z_n = 1.088754$$

$$L \leftarrow \begin{cases} 116 * Y^{1/3} - 16 & \text{for } Y > 0.008856 \\ 903.3 * Y & \text{for } Y \leq 0.008856 \end{cases}$$

$$a \leftarrow 500(f(X) - f(Y)) + \text{delta}$$

$$b \leftarrow 200(f(Y) - f(Z)) + \text{delta}$$

$$f(t) = \begin{cases} t^{1/3} & \text{for } t > 0.008856 \\ 7.787t + 16/116 & \text{for } t \leq 0.008856 \end{cases}$$

$$\text{delta} = \begin{cases} 128 & \text{for 8-bit images} \\ 0 & \text{for floating-point images} \end{cases}$$

最後L範圍0到100，a範圍-127到127，b範圍-127到127，接著根據圖的形態，轉換到合適的範圍。8位元：L=L*255/100，a=a+128，b=b+128。16位元：目前不支援。浮點數圖：維持不變。

BGR & CIE Luv*

參數：CV_BGR2Luv, CV_RGB2Luv, CV_Luv2BGR, CV_Luv2RGB

當我們計算8位元或16位元圖時，R、G、B都轉成0到1之間的浮點數，比如R=R/(R+G+B)，接著用下列公式進行轉換，分別得到L、a、b的值。

最後L範圍0到100，u範圍-134到220，v範圍-140到122，接著根據圖的形態，轉換到合適的範圍。8位元：L=L*255/100，u=255/354(u+134)，v=255/262*(v+140)。16位元：目前不支援。浮點數圖：維持不變。

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \leftarrow \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$L \leftarrow \begin{cases} 116Y^{1/3} & \text{for } Y > 0.008856 \\ 903.3Y & \text{for } Y \leq 0.008856 \end{cases}$$

$$u' \leftarrow 4 * X / (X + 15 * Y + 3Z)$$

$$v' \leftarrow 9 * Y / (X + 15 * Y + 3Z)$$

$$u \leftarrow 13 * L * (u' - u_n) \quad \text{where } u_n = 0.19793943$$

$$v \leftarrow 13 * L * (v' - v_n) \quad \text{where } v_n = 0.46831096$$

[回到首頁](#)

[回到OpenCV教學](#)

參考資料：

[OpenCV 教程](#)

[CIE 1931色彩空間](#)

[Lab色彩空間](#)

📅 2015-12-09 👤 阿宅 📁 OpenCV, 繪圖與影像空間 🔗 cvtColor, 色彩空間

0 Comments

猴子遇到0與1! 程式學習筆記

1 Login ▾ Recommend Share

Sort by Best ▾



Start the discussion...

Be the first to comment.

ALSO ON 猴子遇到0與1! 程式學習筆記**文件對話框(QFileDialog)**

1 comment • 6 months ago

楊政穎 — dialog.cpp 裡面的 QString s
=
QFileDialog::getOpenFileName(this, tr

Qt主窗口(Top Level Window)

1 comment • 6 months ago

mike — 喔喔

 Subscribe Add Disqus to your site Add Disqus Add Privacy

自豪的採用 WordPress