# COLOR

[NAME](#gjdgxs)

[SYNOPSIS](#30j0zll)

[DESCRIPTION](#1fob9te)

[SEE ALSO](#3znysh7)

## NAME

|  |  |
| --- | --- |
|  | TIFFYCbCrToRGBInit, TIFFYCbCrtoRGB, TIFFCIELabToRGBInit, TIFFCIELabToXYZ, TIFFXYZToRGB − color conversion routines. |

## SYNOPSIS

|  |  |
| --- | --- |
|  | **#include <tiffio.h>**  **int TIFFYCbCrToRGBInit(TIFFYCbCrToRGB \****ycbcr***, float \****luma***, float \****refBlackWhite"***);"**  **void TIFFYCbCrtoRGB(TIFFYCbCrToRGB \****ycbcr***, uint32** *Y***, int32** *Cb***, int32** *Cr***, uint32 \****R***, uint32 \****G***, uint32 \****B* **);**  **int TIFFCIELabToRGBInit(TIFFCIELabToRGB \****cielab***, TIFFDisplay \****display***, float \****refWhite***);**  **void TIFFCIELabToXYZ(TIFFCIELabToRGB \****cielab***, uint32** *L***, int32** *a***, int32** *b***, float \****X***, float \****Y***, float \****Z***);**  **void TIFFXYZToRGB(TIFFCIELabToRGB \****cielab***, float** *X***, float** *Y***, float** *Z"***,***uint32***\*"***R***, uint32 \****G***, uint32 \****B***);** |

## DESCRIPTION

|  |  |
| --- | --- |
|  | TIFF supports several color spaces for images stored in that format. There is usually a problem of application to handle the data properly and convert between different colorspaces for displaying and printing purposes. To simplify this task libtiff implements several color conversion routines itself. In particular, these routines used in **TIFFRGBAImage(3TIFF)** interface.  **TIFFYCbCrToRGBInit()** used to initialize *YCbCr* to *RGB* conversion state. Allocating and freeing of the *ycbcr* structure belongs to programmer. *TIFFYCbCrToRGB* defined in **tiffio.h** as |

|  |  |
| --- | --- |
|  | typedef struct { /\* YCbCr->RGB support \*/  TIFFRGBValue\* clamptab; /\* range clamping table \*/ |

int\*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | Cr\_r\_tab;  int\* |  |
|  |  |  | Cb\_b\_tab;  int32\* |  |
|  |  |  | Cr\_g\_tab;  int32\* |  |
|  |  |  | Cb\_g\_tab; |  |

|  |  |
| --- | --- |
|  | int32\* Y\_tab;  } TIFFYCbCrToRGB; |

|  |  |
| --- | --- |
|  | *luma* is a float array of three values representing proportions of the red, green and blue in luminance, Y (see section 21 of the TIFF 6.0 specification, where the YCbCr images discussed). *TIFFTAG\_YCBCRCOEFFICIENTS* holds that values in TIFF file. *refBlackWhite* is a float array of 6 values which specifies a pair of headroom and footroom image data values (codes) for each image component (see section 20 of the TIFF 6.0 specification where the colorinmetry fields discussed). *TIFFTAG\_REFERENCEBLACKWHITE* is responsible for storing these values in TIFF file. Following code snippet should helps to understand the the technique: |

|  |  |
| --- | --- |
|  | float \*luma, \*refBlackWhite; uint16 hs, vs;  /\* Initialize structures \*/ ycbcr = (TIFFYCbCrToRGB\*) |

|  |  |
| --- | --- |
|  | \_TIFFmalloc(TIFFroundup(sizeof(TIFFYCbCrToRGB), sizeof(long)) |
|  | + 4\*256\*sizeof(TIFFRGBValue) |
|  | + 2\*256\*sizeof(int) |
|  | + 3\*256\*sizeof(int32)); |

|  |  |
| --- | --- |
|  | if (ycbcr == NULL) {  TIFFError("YCbCr->RGB", |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | "No space for YCbCr->RGB conversion state"); |  |

|  |  |
| --- | --- |
|  | exit(0);  }  TIFFGetFieldDefaulted(tif, TIFFTAG\_YCBCRCOEFFICIENTS, &luma);  TIFFGetFieldDefaulted(tif, TIFFTAG\_REFERENCEBLACKWHITE, &refBlackWhite);  if (TIFFYCbCrToRGBInit(ycbcr, luma, refBlackWhite) < 0) |

|  |  |
| --- | --- |
|  | exit(0); |

|  |  |
| --- | --- |
|  | /\* Start conversion \*/  uint32 r, g, b;  uint32 Y;  int32 Cb, Cr;  for each pixel in image |

|  |  |
| --- | --- |
|  | TIFFYCbCrtoRGB(img->ycbcr, Y, Cb, Cr, &r, &g, &b); |

|  |  |
| --- | --- |
|  | /\* Free state structure \*/  \_TIFFfree(ycbcr); |

|  |  |
| --- | --- |
|  | **TIFFCIELabToRGBInit()** initializes the *CIE L\*a\*b\* 1976* to *RGB* conversion state. **TIFFCIELabToRGB** defined as |

|  |  |
| --- | --- |
|  | #define CIELABTORGB\_TABLE\_RANGE 1500 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | typedef struct { |  |  |  |  | /\* CIE Lab 1976->RGB support \*/ |  |  |
|  |  | int | range; |  |  | /\* Size of conversion table \*/ |  |  |
|  |  | float |  | rstep, gstep, bstep; |  |  |  |  |
|  |  | float |  | X0, Y0, Z0; |  |  | /\* Reference white point \*/ |  |
|  |  | TIFFDisplay display; |  |  |  |  |  |  |
|  |  | float |  | Yr2r[CIELABTORGB\_TABLE\_RANGE + 1]; /\* Conversion of Yr to r \*/ |  |  |  |  |
|  |  | float |  | Yg2g[CIELABTORGB\_TABLE\_RANGE + 1]; /\* Conversion of Yg to g \*/ |  |  |  |  |
|  |  | float |  | Yb2b[CIELABTORGB\_TABLE\_RANGE + 1]; /\* Conversion of Yb to b \*/ |  |  |  |  |

|  |  |
| --- | --- |
|  | } TIFFCIELabToRGB; |

|  |  |
| --- | --- |
|  | *display* is a display device description, declared as |

|  |  |
| --- | --- |
|  | typedef struct { |

|  |  |
| --- | --- |
|  | float d\_mat[3][3]; /\* XYZ -> luminance matrix \*/ |
|  | float d\_YCR; /\* Light o/p for reference white \*/ |
|  | float d\_YCG; |
|  | float d\_YCB; |
|  | uint32 d\_Vrwr; /\* Pixel values for ref. white \*/ |
|  | uint32 d\_Vrwg; |
|  | uint32 d\_Vrwb; |
|  | float d\_Y0R; /\* Residual light for black pixel \*/ |
|  | float d\_Y0G; |
|  | float d\_Y0B; |
|  | float d\_gammaR; /\* Gamma values for the three guns \*/ |
|  | float d\_gammaG; |
|  | float d\_gammaB; |

|  |  |
| --- | --- |
|  | } TIFFDisplay; |

|  |  |
| --- | --- |
|  | For example, the one can use sRGB device, which has the following parameters: |

|  |  |
| --- | --- |
|  | TIFFDisplay display\_sRGB = { |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | { /\* XYZ -> luminance matrix \*/ |  |  |
|  |  |  | { 3.2410F, -1.5374F, -0.4986F }, |  |
|  |  |  | { -0.9692F, 1.8760F, 0.0416F }, |  |
|  |  |  | { 0.0556F, -0.2040F, 1.0570F } |  |
|  |  | }, |  |  |
|  |  | 100.0F, 100.0F, 100.0F, /\* Light o/p for reference white \*/ |  |  |
|  |  | 255, 255, 255, /\* Pixel values for ref. white \*/ |  |  |
|  |  | 1.0F, 1.0F, 1.0F, /\* Residual light o/p for black pixel \*/ |  |  |
|  |  | 2.4F, 2.4F, 2.4F, /\* Gamma values for the three guns \*/ |  |  |

|  |  |
| --- | --- |
|  | }; |

|  |  |
| --- | --- |
|  | *refWhite* is a color temperature of the reference white. The *TIFFTAG\_WHITEPOINT* contains the chromaticity of the white point of the image from where the reference white can be calculated using following formulae: |

|  |  |
| --- | --- |
|  | refWhite\_Y = 100.0  refWhite\_X = whitePoint\_x / whitePoint\_y \* refWhite\_Y  refWhite\_Z = (1.0 - whitePoint\_x - whitePoint\_y) / whitePoint\_y \* refWhite\_X |

|  |  |
| --- | --- |
|  | The conversion itself performed in two steps: at the first one we will convert *CIE L\*a\*b\* 1976* to *CIE XYZ* using **TIFFCIELabToXYZ()** routine, and at the second step we will convert *CIE XYZ* to *RGB* using **TIFFXYZToRGB().** Look at the code sample below: |

|  |  |
| --- | --- |
|  | float \*whitePoint; float refWhite[3];  /\* Initialize structures \*/ img->cielab = (TIFFCIELabToRGB \*) |

|  |  |
| --- | --- |
|  | \_TIFFmalloc(sizeof(TIFFCIELabToRGB)); |

|  |  |
| --- | --- |
|  | if (!cielab) { |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | TIFFError("CIE L\*a\*b\*->RGB", |  |  |
|  |  |  | "No space for CIE L\*a\*b\*->RGB conversion state."); |  |
|  |  | exit(0); |  |  |

|  |  |
| --- | --- |
|  | }  TIFFGetFieldDefaulted(tif, TIFFTAG\_WHITEPOINT, &whitePoint);  refWhite[1] = 100.0F;  refWhite[0] = whitePoint[0] / whitePoint[1] \* refWhite[1];  refWhite[2] = (1.0F - whitePoint[0] - whitePoint[1]) |

|  |  |
| --- | --- |
|  | / whitePoint[1] \* refWhite[1]; |

|  |  |
| --- | --- |
|  | if (TIFFCIELabToRGBInit(cielab, &display\_sRGB, refWhite) < 0) { |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | TIFFError("CIE L\*a\*b\*->RGB", |  |  |
|  |  |  | "Failed to initialize CIE L\*a\*b\*->RGB conversion state."); |  |
|  |  | \_TIFFfree(cielab); |  |  |
|  |  | exit(0); |  |  |

|  |  |
| --- | --- |
|  | }  /\* Now we can start to convert \*/  uint32 r, g, b;  uint32 L;  int32 a, b;  float X, Y, Z;  for each pixel in image |

|  |  |
| --- | --- |
|  | TIFFCIELabToXYZ(cielab, L, a, b, &X, &Y, &Z); |
|  | TIFFXYZToRGB(cielab, X, Y, Z, &r, &g, &b); |

|  |  |
| --- | --- |
|  | /\* Don’t forget to free the state structure \*/  \_TIFFfree(cielab); |

## SEE ALSO

|  |  |
| --- | --- |
|  | **TIFFRGBAImage**(3TIFF) **libtiff**(3TIFF),  Libtiff library home page: **http://www.simplesystems.org/libtiff/** |