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*, **const 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/**