Subject: Add new color Standard convertions RGB <-> Cie XYZ <-> Cie L*a*b*
Posted by tojocky on Tue, 03 May 2011 13:53:34 GMT
View Forum Message <> Reply to Message

Hello All (especialy Mirek),

According by addresses:
- http://en.wikipedia.org/wiki/Lab_color_space#cite_note-12
- http://www.easyrgb.com/index.php?X=MATH&H=01#text1
- and others source-code projects

I propose to add the following functions in Core/Color.h/cpp:

```
void RGBtoXYZ(double r, double g, double b, double& x, double& y, double& z){
 r /= 255;
 g /= 255;
 b /= 255;
 double delta = 0.04045;//?
 if (r > delta)
  r = pow(((r + 0.055) / 1.055), 2.4);
 else r /= 12.92;

 if (g > delta)
  g = pow(((g + 0.055) / 1.055), 2.4);
 else g /= 12.92;

 if (b > delta)
  b = pow(((b + 0.055) / 1.055), 2.4);
 else b /= 12.92;


 x = r * 41.24 + g * 35.76 + b * 18.05;
 y = r * 21.26 + g * 71.52 + b * 7.22;
 z = r * 1.93 + g * 11.92 + b * 95.05;
}

void XYZtoRGB(double x, double y, double z, int &r, int &g, int &b){
 double dr,dg,db;
 dr =  0.032406  * x - 0.015372 * y - 0.0049865 * z;
 dg = -0.0096891 * x + 0.018758 * y + 0.00041514* z;
 db =  0.00055708* x - 0.0020401* y + 0.01057   * z;

 if(dr > 0.0031308)
  dr = exp(log(dr) / 2.4) * 1.055-0.055;
 else dr = dr*12.92;

 if(dg > 0.0031308)
  dg = exp(log(dg) / 2.4) * 1.055 - 0.055;
 else dg = dg*12.92;
```

```
 if(db > 0.0031308)
  db = exp(log(db) / 2.4) * 1.055 - 0.055;
 else db = db*12.92;

 dr *= 255;
 dg *= 255;
 db *= 255;

 dr = minmax<double>(0.0,dr, 255.0);
 dg = minmax<double>(0.0,dg, 255.0);
 db = minmax<double>(0.0,db, 255.0);

 r = int(dr + 0.5); //?0.5
 g = int(dg + 0.5); //?0.5
 b = int(db + 0.5); //?0.5
}

void XYZtoCEILab(double x, double y, double z, double& l, double& a, double& b){

 y /= 100.000;          //ref_Y = 100.000
 z /= 108.883;          //ref_Z = 108.883
 double delta = 0.008856; //pow(6/29,3);
 double delta2 = 7.787; //pow(29/6, 2)/3;

 if (x > delta)
  x = pow(x, 1/3);
 else x = (delta2 * x) + (4 / 29);

 if (y > delta)
  y = pow(y, 1/3);
 else y = (delta2 * y) + (4 / 29);

 if (z > delta)
  z = pow(z, 1/3);
 else z = (delta2 * z) + (4 / 29);

 l = (116 * y) - 16;
 a = 500 * (x - y);
 b = 200 * (y - z);
}

void CEILabtoXYZ(double l, double a, double b, double& x, double& y, double& z){
 y = (l + 16) / 116;
 x = y + a / 500;
 z = y - b / 200;

 double delta = 0.20689655; //6/29;
```

```
 double delta2 = 7.787; //pow(29/6, 2)/3;

 if (x > delta)
  x = pow(x, 3);
 else x = ((delta2 * x) - (4 / 29)) / delta2;

 if (y > delta)
  y = pow(y, 3);
 else y = ((delta2 * y) - (4 / 29)) / delta2;

 if (z > delta)
  z = pow(z, 3);
 else z = ((delta2 * z) - (4 / 29)) / delta2;
}

void RGBtoCEILab(double r, double g, double b, double& l, double& a, double& lb){
 double x,y,z;
 RGBtoXYZ(r,g,b,x,y,z);
 XYZtoCEILab(x,y,z,l,a,lb);
}

void CEILabtoRGB(double l, double a, double lb, double& r, double& g, double& b){
 double x,y,z;
 CEILabtoXYZ(l,a,lb,x,y,z);
 RGBtoXYZ(x,y,z,r,g,b);
}
```

This functions I will use in histogram equalization of image according by:
http://en.wikipedia.org/wiki/Histogram_equalisation

HSV/HSL is not so exact as CIE L*a*b*

Any comment are welcome!

Subject: Re: Add new color Standard convertions RGB <-> Cie XYZ <-> Cie L*a*b*
Posted by mirek on Tue, 07 Jun 2011 18:57:23 GMT
View Forum Message <> Reply to Message

tojocky wrote on Tue, 03 May 2011 09:53Hello All (especialy Mirek),

According by addresses:
- http://en.wikipedia.org/wiki/Lab_color_space#cite_note-12
- http://www.easyrgb.com/index.php?X=MATH&H=01#text1
- and others source-code projects

I propose to add the following functions in Core/Color.h/cpp:

```
void RGBtoXYZ(double r, double g, double b, double& x, double& y, double& z){
 r /= 255;
 g /= 255;
 b /= 255;
 double delta = 0.04045;//?
 if (r > delta)
  r = pow(((r + 0.055) / 1.055), 2.4);
 else r /= 12.92;

 if (g > delta)
  g = pow(((g + 0.055) / 1.055), 2.4);
 else g /= 12.92;

 if (b > delta)
  b = pow(((b + 0.055) / 1.055), 2.4);
 else b /= 12.92;


 x = r * 41.24 + g * 35.76 + b * 18.05;
 y = r * 21.26 + g * 71.52 + b * 7.22;
 z = r * 1.93 + g * 11.92 + b * 95.05;
}

void XYZtoRGB(double x, double y, double z, int &r, int &g, int &b){
 double dr,dg,db;
 dr =  0.032406  * x - 0.015372 * y - 0.0049865 * z;
 dg = -0.0096891 * x + 0.018758 * y + 0.00041514* z;
 db =  0.00055708* x - 0.0020401* y + 0.01057   * z;

 if(dr > 0.0031308)
  dr = exp(log(dr) / 2.4) * 1.055-0.055;
 else dr = dr*12.92;

 if(dg > 0.0031308)
  dg = exp(log(dg) / 2.4) * 1.055 - 0.055;
 else dg = dg*12.92;

 if(db > 0.0031308)
  db = exp(log(db) / 2.4) * 1.055 - 0.055;
 else db = db*12.92;

 dr *= 255;
 dg *= 255;
 db *= 255;

 dr = minmax<double>(0.0,dr, 255.0);
 dg = minmax<double>(0.0,dg, 255.0);
 db = minmax<double>(0.0,db, 255.0);
```

```
 r = int(dr + 0.5); //?0.5
 g = int(dg + 0.5); //?0.5
 b = int(db + 0.5); //?0.5
}

void XYZtoCEILab(double x, double y, double z, double& l, double& a, double& b){

 y /= 100.000;          //ref_Y = 100.000
 z /= 108.883;          //ref_Z = 108.883
 double delta = 0.008856; //pow(6/29,3);
 double delta2 = 7.787; //pow(29/6, 2)/3;

 if (x > delta)
  x = pow(x, 1/3);
 else x = (delta2 * x) + (4 / 29);

 if (y > delta)
  y = pow(y, 1/3);
 else y = (delta2 * y) + (4 / 29);

 if (z > delta)
  z = pow(z, 1/3);
 else z = (delta2 * z) + (4 / 29);

 l = (116 * y) - 16;
 a = 500 * (x - y);
 b = 200 * (y - z);
}

void CEILabtoXYZ(double l, double a, double b, double& x, double& y, double& z){
 y = (l + 16) / 116;
 x = y + a / 500;
 z = y - b / 200;

 double delta = 0.20689655; //6/29;
 double delta2 = 7.787; //pow(29/6, 2)/3;

 if (x > delta)
  x = pow(x, 3);
 else x = ((delta2 * x) - (4 / 29)) / delta2;

 if (y > delta)
  y = pow(y, 3);
 else y = ((delta2 * y) - (4 / 29)) / delta2;

 if (z > delta)
  z = pow(z, 3);
```

```
 else z = ((delta2 * z) - (4 / 29)) / delta2;
}

void RGBtoCEILab(double r, double g, double b, double& l, double& a, double& lb){
 double x,y,z;
 RGBtoXYZ(r,g,b,x,y,z);
 XYZtoCEILab(x,y,z,l,a,lb);
}

void CEILabtoRGB(double l, double a, double lb, double& r, double& g, double& b){
 double x,y,z;
 CEILabtoXYZ(l,a,lb,x,y,z);
 RGBtoXYZ(x,y,z,r,g,b);
}
```

This functions I will use in histogram equalization of image according by:
http://en.wikipedia.org/wiki/Histogram_equalisation

HSV/HSL is not so exact as CIE L*a*b*

Any comment are welcome!

This is fine code, but I am afraid too specialised for Core (I guess Core is sort of overloaded now).

What about bazaar or plugin or maybe Geom? Or perhaps Painter?

Maybe we should start a new "high level" raster graphics package? I think the equalization code would fit nicely there.

Subject: Re: Add new color Standard convertions RGB <-> Cie XYZ <-> Cie L*a*b*
Posted by tojocky on Tue, 07 Jun 2011 21:03:59 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 07 June 2011 21:57

This is fine code, but I am afraid too specialised for Core (I guess Core is sort of overloaded now).

What about bazaar or plugin or maybe Geom? Or perhaps Painter?

Maybe we should start a new "high level" raster graphics package? I think the equalization code would fit nicely there.

This code is not quite correct. I have the latest tested version. I used this code to autolevel color images (converted into CIE L*a*b*). HSV autolevel of color images losts some color (this problem persists in Photoshop too).

Ok, I can prepare a package and uploat into bazaar. But I think it is a better idea to include it in Painter.
