

Politechnika Częstochowska
Instytut Informatyki Teoretycznej i Stosowanej

Sawicki, Adam

Konwersja przestrzeni kolorystycznych obrazów

Praca egzaminacyjna z przedmiotu
“Podstawy cyfrowego przetwarzania sygnałów”

Prowadzący
prof. Leonid Kompanets

Częstochowa, 2005/06

1. Sformułowanie zagadnienia

W wielu dziedzinach informatyki zachodzi potrzeba reprezentowania barw jako sygnałów cyfrowych. W tym celu stosuje się jeden z modeli barw (przestrzeni kolorystycznych) i stąd potrzebne są algorytmy do konwersji pomiędzy nimi.

2. Informacja wejściowa

Kolor wyrażony w jednej z przestrzeni kolorystycznych:

2.1. RGB (od Red, Green, Blue – czerwona, zielona, niebieska)

2.2. CMY (Cyan, Magenta, Yellow)

2.3. CMYK (Cyan, Magenta, Yellow, Black)

2.4. YUV, YIQ

2.5. HSV/HSB (Hue, Saturation, Value/Brightness)

3. Informacja wyjściowa

Kolor wyrażony w innej spośród tych przestrzeni kolorystycznych.

4. Algorytmy konwersji

4.1. Konwersja z RGB do odcieni szarości

$$I = 0.3R + 0.59G + 0.11B$$

4.2. Konwersja z RGB do CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

4.3. Konwersja z CMY do RGB

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} C \\ M \\ Y \end{bmatrix}$$

4.4. Konwersja z CMY do CMYK

$$\begin{aligned} K &= \min(C, M, Y) \\ C &= C - K \\ M &= M - K \\ Y &= Y - K \end{aligned}$$

4.5. Konwersja z RGB do YIQ

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.528 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

4.6. Konwersja z RGB do YUV

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

4.7. Konwersja z HSB do RGB

Funkcja *hsv2rgb* w języku C++.

Informacje wejściowe: *hue* – barwa, *sat* – nasycenie, *val* – wartość

Informacje wyjściowe: *red* – składowa czerwona, *green* – składowa zielona, *blue* – składowa niebieska

```
void hsv2rgb(float hue, float sat, float val, float *red, float *green,
float *blue)
{
    if (val == 0.0f)
    {
        *red = *green = *blue = 0.0f;
        break;
    }

    int i;
    float f, p, q, t;

    hue = fmodf(hue, 360.0f);
    hue /= 60.0f;
    i = (int)hue;
    f = hue - i;
    p = val * (1.0f - sat);
    q = val * (1.0f - (sat * f));
    t = val * (1.0f - (sat * (1.0f - f)));

    if (i == 0.0f)
    {
        *red = val;
        *green = t;
        *blue = p;
    }
    else if (i == 1)
    {
        *red = q;
        *green = val;
        *blue = p;
    }
    else if (i == 2)
    {
        *red = p;
        *green = val;
        *blue = t;
    }
    else if (i == 3)
    {
        *red = p;
        *green = q;
        *blue = val;
    }
    else if (i == 4)
    {
        *red = t;
        *green = p;
        *blue = val;
    }
    else if (i == 5)
    {
        *red = val;
        *green = p;
        *blue = q;
    }
}
```

4.8. Konwersja z RGB do HSV

Funkcja *rgb2hsv* w języku C++.

Informacje wejściowe: *red* – składowa czerwona, *green* – składowa zielona, *blue* – składowa niebieska

Informacje wyjściowe: *hue* – barwa, *sat* – nasycenie, *val* – wartość

```
void rgb2hsv(float red, float green, float blue, float *hue, float *sat,
float *val)
{
    int i;
    float x, val, f, hue, sat, val;

    x = std::min(std::min(red, green), blue);
    *val = std::max(std::max(red, green), blue);
    if (x == *val)
    {
        *hue = 0.0f; // dowolna
        *sat = 0.0f;
        *val = 1.0f;
    }

    f = (red == x) ? green-blue : ((green == x) ? blue-red : red-green);
    i = (red == x) ? 3 : ((green == x) ? 5 : 1);
    *hue = fmodf(floorf((i - f / (val - x)) * 60.0f), 360.0f);
    *sat = floorf(((val - x) / val));
    *val = floorf(val);
}
```

5. Wnioski

5.1. Istnieje kilka głównych przestrzeni barwnych stosowanych do reprezentacji barw w cyfrowym przetwarzaniu sygnałów. Każdy z nich opisuje barwę za pomocą jednej (jak odcienie szarości – patrz 4.1) lub wielu (najczęściej trzech, a w CMYK czterech – patrz 4.4) wartości liczbowych ze znormalizowanego zakresu od 0 do 1.

5.2. Aby dokonać konwersji między tymi przestrzeniami barwnymi, należy użyć algorytmu przekształcającego w odpowiedni sposób poszczególne składowe reprezentacji barwy (4.1 – średnia ważona, 4.2-4.6 – proste przekształcenia liniowe, 4.7-4.8 – bardziej złożone algorytmy z instrukcjami warunkowymi).