

**УДК 526.31**

## **АЛГОРИТМЫ ЦВЕТОДЕЛЕНИЯ**

***В.Е. Езовских***

В настоящее время для хранения изображения используются различные форматы представления цвета. Рассматриваются некоторые подходы к вопросу о преобразовании форматов.

Довольно широко распространены изображения в формате TrueColor. В нем цвет каждого пикселя задается 24 битами, причем интенсивность каждого основного цвета ( $R$  – красный,  $G$  – зеленый,  $B$  – синий) меняется в пределах от 0 до 255. Задача цветоделения возникает в тех случаях, когда используется меньшая глубина цвета или меньшее количество цветов (формат GIF, печать ограниченным набором красок), а также при печати изображений в формате BW или градациях серого.

### **1. Цветоделение для представления изображения в градациях серого**

Существует стандарт CCIR601 [1], задающий соотношения для переходов между цветовыми системами RGB и YCbCr:

*RGB to YCbCr*

$$\begin{aligned} Y &= 0.2990R + 0.5870G + 0.1140B \\ Cb &= -0.1687R - 0.3313G + 0.5000B \\ Cr &= 0.5000R - 0.4187G - 0.0813B \end{aligned}$$

*YCbCr to RGB*

$$\begin{aligned} R &= Y + 1.4020Cr \\ G &= Y - 0.3441Cb - 0.7141Cr \\ B &= Y + 1.7720Cb \end{aligned}$$

Найдем  $N_R, N_G, N_B$ , такие, что для некоторой степени  $p$  числа два (степень двойки выбираем для того, чтобы избежать выполнения операций деления и свести вычисления к сдвигам) величины  $y_r = N_R/2^p, y_g = N_G/2^p, y_b = N_B/2^p$  приближают коэффициенты  $Y_R = 0.2990, Y_G = 0.5870$  и  $Y_B = 0.1140$ , минимизируя при этом невязку

$$d = \sqrt{(y_r - Y_R)^2 + (y_g - Y_G)^2 + (y_b - Y_B)^2}.$$

Результаты вычислений приведены в таблице

$p$	$N_R$	$y_r$	$N_G$	$y_g$	$N_B$	$y_b$	$d$
4	5	0.3125	9	0.5625	2	0.1250	$1.5 * 10^{-2}$
5	10	0.3125	19	0.5938	4	0.1250	$9.3 * 10^{-3}$
6	19	0.2969	38	0.5938	7	0.1094	$4.2 * 10^{-3}$
7	38	0.2969	75	0.5859	15	0.1172	$1.5 * 10^{-3}$
8	77	0.3008	150	0.5859	29	0.1133	$1.1 * 10^{-3}$
9	153	0.2988	301	0.5879	58	0.1133	$5.7 * 10^{-4}$
10	306	0.2988	601	0.5869	117	0.1143	$1.6 * 10^{-4}$
11	612	0.2988	1202	0.5869	233	0.1138	$1.5 * 10^{-4}$
12	1225	0.2991	2404	0.5870	467	0.1140	$5.6 * 10^{-5}$
13	2449	0.2990	4809	0.5870	934	0.1140	$3.2 * 10^{-5}$
14	4899	0.2990	9617	0.5870	1868	0.1140	$1.5 * 10^{-5}$
15	9798	0.2990	19235	0.5870	3736	0.1140	$9.2 * 10^{-6}$
16	19595	0.2990	38470	0.5870	7471	0.1140	$3.4 * 10^{-6}$

Исходя из данных таблицы и соображений минимизации числа и сложности операций при преобразовании TrueColor изображений в изображения в градациях серого, при вычислениях можно рекомендовать одну из двух формул

$$Y = (77R + 150G + 29B) \pmod{256},$$

$$Y = (19595R + 38470G + 7471B) \pmod{65536}.$$

Погрешность первой из них —  $1.1 * 10^{-3}$ , второй —  $3.4 * 10^{-6}$ . При вычислениях по первой формуле используются двухбайтовые переменные типа *word* с выделением старшего байта, для второй — четырехбайтовые переменные типа *cardinal* с выделением третьего байта. Для точного округления достаточен анализ старшего бита (знак) одного из байтов результата.

Можно избежать умножений, если предварительно вычислить  $3*256=768$  произведений и заполнить ими массивы  $C_R, C_G$  и  $C_B$ . В этом случае вычисления проводятся по формуле

$$Y = (C_R[R] + C_G[G] + C_B[B]) \pmod{256(65536)}.$$

## 2. Равномерное цветоделение

Графический режим TrueColor использует 167777216 цветов при глубине цвета (количество бит для кодирования компоненты цвета) восемь. Некоторые графические режимы VGA и SVGA, а также графические форматы (например — GIF) используют иное количество цветов и другую глубину цвета. Так VGA режимы 13 и 13X используют 256 цветов при глубине цвета шесть.

Для цветоделения при преобразовании TrueColor to 13X можно использовать следующий алгоритм. Найдем  $N$ , такое, что  $N^3 \leq 256$ . Очевидно, что максимальное значение  $N = 6$  ( $6^3 = 216$ ), так что установим палитру с шестью градациями каждого основного цвета, учитывая его глубину. Можно использовать алгоритм Брезенхейма [1] для линейной интерполяции из точки плоскости  $(0,0)$  в точку  $(5,63)$  при заполнении массивов  $I_R[6], I_G[6], I_B[6]$ . Используем заполнение палитры со смещением  $Off = 16$ , не изменяя основные VGA цвета, а именно для атрибута  $A$  в диапазоне  $[16, 231]$  записываем следующие значения компонент цвета

$$\begin{aligned} R_A &= I_R[(A - Off) \text{ mod } 6] \\ G_A &= I_G[((A - Off) \text{ div } 6) \text{ mod } 6] \\ B_A &= I_B[((A - Off) \text{ div } 36) \text{ mod } 6]. \end{aligned}$$

Для уменьшения объема вычислений заполним вспомогательный массив  $W[64]$  номерами градаций цвета от 0 до 5, для чего можно воспользоваться алгоритмом Брезенхейма. Если в исходном изображении TrueColor имелась точка с цветом  $(R, G, B)$ , то ее атрибут в VGA13X вычисляется как

$$f = Off + W[R \text{ shr } 2] + 6W[G \text{ shr } 2] + 36W[B \text{ shr } 2].$$

Заметим, что меняя характер заполнения массивов  $I_R, I_G, I_B, W$ , можно осуществить цветовую и  $\gamma$ -коррекцию.

## 3. Адаптивное цветоделение

Равномерное цветоделение не учитывает специфических цветовых особенностей изображения. Ясно, что изображения зимней пармы и морского пейзажа требуют разных палитр.

Пусть имеется TrueColor изображение из  $N$  пикселей с цветом  $(R_i, G_i, B_i)$  ( $i = 1 \dots N$ ). Изобразим все  $N$  точек в цветовом пространстве RGB. Построим окаймляющий прямоугольный параллелепипед. Найдем самое длинное его ребро. Пусть это будет ребро  $R$ . По координате  $R$  находим медиану  $M_R$  и разбиваем все множество исходных точек

на два со значениями  $R_i \leq M_R$  и  $R_i \geq M_R$ , причем число точек в этих множествах отличается не более чем на единицу. Для этих множеств точек находим окаймляющие параллелепипеды, повторяем процесс до тех пор, пока число параллелепипедов не станет равным требуемому числу цветов, например – 256. Всем точкам, которые попадают внутрь определенного параллелепипеда, присваивается цвет, соответствующий геометрическому центру параллелепипеда.

Отметим, что этот алгоритм требует большого объема оперативной памяти. В то же время некоторые особенности исходных данных позволяют написать довольно эффективную его реализацию. Основой служит вспомогательный байтовый массив  $W[N]$ . В который записывается номер параллелепипеда, в который попадает точка. Имеется достаточное число массивов, в которые записывается число появления среди  $R_i, G - R_i, B_i$  определенного значения в диапазоне  $0 \dots 255$ .

При одном сканировании массива  $W$  параллельно решаются задачи о нахождении всех окаймляющих параллелепипедов определенного шага алгоритма и нахождение медиан. Для распределения точек между параллелепипедами требуется еще два прохода.

## Литература

1. Роджерс Дж. Алгоритмические основы машинной графики. М.: 1985.

### Summary

#### Ezovskih V.E. Color sampling algorithms

Three algorithms of color sampling suitable for color conversion are considered. Some practical features of realization are briefly discussed.

*Сыктывкарский университет*

*Поступила 11.10.2005*