



Кодирование графической информации

Мы уже привыкли к оформлению интерфейса операционной системы и других программ графическими элементами, в целом создающими впечатление красиво оформленной страницы книги. Так было не всегда. Раньше перед пользователями был темный экран, на который выводились сообщения, а команды можно было ввести только в специальной строке. Что позволило коренным образом изменить представление информации на экране пользователю?

Для того чтобы ответить на этот вопрос, разберемся что происходит с графической информацией в памяти компьютера, иначе как кодируется графическая информация.

Изображение экрана можно представить состоящим из точек, которые принято называть пикселями – это наименьший элемент изображения. Все пиксели, заполняющие экран монитора образуют растр. Чем больше точек, тем четче изображение. Понятие разрешение экрана подразумевает количество пиксель на дюйм, чем больше эта цифра, тем качественнее изображение. Изображение экрана, состоящее из точек, называют растровым.



Пиксель - наименьший элемент экрана, точка.



Растр – пиксели, заполняющие экран монитора.



Разрешение монитора – количество пиксель на дюйм.

Представить в увеличенном виде растровое изображение, позволяют схемы для вышивания крестом, где один крестик можно соотнести с увеличенным пикселем экрана.

Рассмотрите схему для вышивания иллюстрации «Ежик в тумане».



На рисунке четко видно, из точек какого цвета, состоит ежик, видно квадратики, которые заполняют рисунок. Уменьшим масштаб схемы, тем самым уменьшая размер единичного квадрата. Изображение становится более сглаженным и целостным. Сейчас размер нашей картинке 72 квадрата (пикселя) по горизонтали и 54 по вертикали.

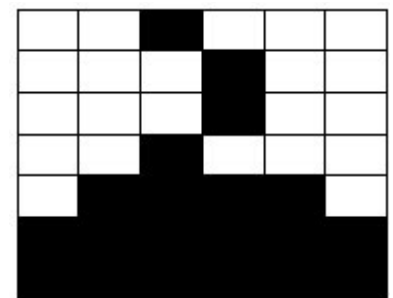
Если захотим вышить эту картинку, то придется для каждого квадрата вычислить его координаты по горизонтали и по вертикали и подобрать нужный по схеме цвет. Сейчас на этой картинке используется 10 основных цветов, представленных ниже на палитре.



Осталось понять, как закодировать цвет.

Сначала разберемся с простым вариантом черно белого изображения, например небольшим фрагментом нашей картины.

Чтобы обозначить два цвета черный и белый, достаточно 2-х знаков, например 0 - белый и 1 - черный. Эта информация может храниться 1 битом.



Для хранения информации обо всем изображении, считаем количество пиксель и умножаем на количество информации, необходимое, чтобы запомнить информацию о цвете каждого из пикселей.

Для нашего изображения, чтобы запомнить черно белое изображение картинке, потребуется: $72 \times 54 \times 1(\text{бит}) = 3888(\text{бит})$ памяти.

Вспомним соотношение единиц измерения памяти:

1 байт=8бит

1КБ=1024байт

1МБ=1024КБ

Теперь переведем биты в байты (разделить на 8), байты в КБ (разделить на 1024).

Таким образом, $3888\text{бит}=486\text{байт}=\underline{0,5\text{КБ}}$

Однако, признаемся себе, что от красивой картинке, при переводе ее в двухцветное изображение, фактически ничего не останется. Посмотрим, как будет выглядеть изображение при использовании серых полутонов. Этот режим нас устроит больше. Но тогда, нам надо сохранить не два цвета, а, например 8 оттенков - от белого до черного. Для хранения такого количества оттенков серого нам потребуется 3 бита, каждый из которых может принимать значение 0 или 1. Всего получится 8 различных комбинаций.

Количество памяти для хранения такого изображения потребуется больше, потому, что 1 цвет каждой точки требует 3 бита, а всего:

$$72 \times 54 \times 3 (\text{бита}) = 11664 (\text{бит}) = 1458 (\text{байт}) = \underline{1,5\text{КБ}}$$

Для сохранения цвета нашего изображения, потребуется еще больше памяти, так как мы посчитали нам необходимо от 10 основных цветов до 20 оттенков.

Современные графические редакторы могут закодировать цвет различными способами, затрачивая на хранение 1 цвета от 8 бит до 24 и более бит информации.



Цветовая палитра, основанная на использовании 8 битов для кодирования 1 цвета, дает 256 цветов. Для сохранения всех цветов нашего изображения этой палитры уже достаточно.

Такое изображение потребует:

$$72 \times 54 \times 8 (\text{бит}) = 31104 (\text{бит}) = 3888 (\text{байт}) = \underline{3,8\text{КБ}}$$

Представим, что наша картинка требует такого количества цветовых оттенков, что для кодирования 1 цвета нужно 24 бита. Тогда можно

подсчитать и объем памяти, необходимый для хранения такого изображения:
 $72 \times 54 \times 24 (\text{бит}) = 93312 (\text{бит}) = 11664 (\text{байт}) = \underline{11,4 \text{КБ}}$

Составим таблицу для сравнения полученных данных для одного изображения, размером 72x54 и различного количества цветов

Количество бит для 1 цвета	Количество цветов	Объем памяти для хранения изображения 72x54
1	2	0,5КБ
3	8	1,5КБ
8	256	3,8КБ
24	16 777 216	11,4КБ

Ясно, что при увеличении количества бит для кодирования 1 цвета

- количество различных цветов, используемых для передачи изображения – увеличивается;
- объем памяти, необходимой для хранения изображения – увеличивается;
- чем больше объем памяти компьютера, тем больше графических элементов может быть использовано для оформления операционной системы и других программ.



Количество бит для кодирования 1 цвета называют **глубиной цвета**.

Действительно, чем больше бит для кодирования 1 цвета, тем больше цветовых оттенков будет передано различными кодовыми комбинациями.

Обсуждая необходимый объем памяти для хранения изображения, мы говорим об оперативной памяти, именно там хранится интерфейс программ при работе пользователей с программами. Интересно, что объем памяти ЭВМ первого поколения (1945-1954гг.) не превышал **2КБ**, второго поколения (1950-1960гг.) - **32КБ**, третьего поколения (1960-середина 1970гг.) объём оперативной памяти достигал от **8 Мб** до **192 Мб**. Современный компьютер допускает объем памяти до 4ГБ. Теперь понятно, что с развитием компьютерных технологий, многоцветность и сложность графического изображения будет совершенствоваться.