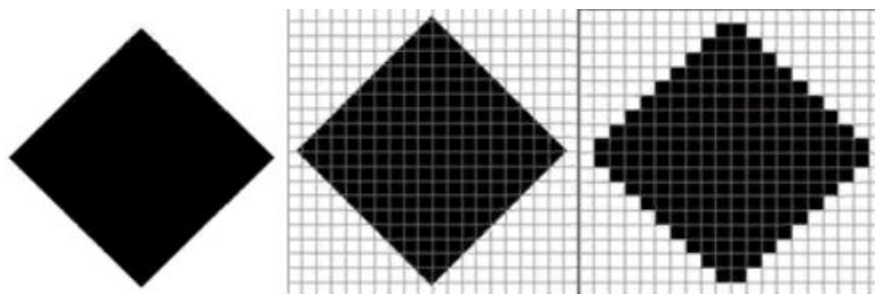


## КОДИРОВАНИЕ РАСТРОВОГО ИЗОБРАЖЕНИЯ

*Бурцев Т.Р., студент 3 курса факультета физики и математики*

*г. Бирск, ФГБОУ ВО Бирский филиал УУНиТ*

Изображение, как и все виды информации, храниться на компьютере в виде определенной комбинации нулей и единиц – двоичного кода. Так как любой рисунок или изображение состоит из бесконечного множества бесконечно малых точек, то и для хранения информации об одном изображении понадобится бесконечно большое количество памяти. Поэтому пытаться закодировать изображение, представляя его в виде бесконечного количества точек, не получится. Однако эту идею можно преобразовать. Начнем с рисунка черного ромба на белом фоне (рис.1). Наложим на это изображение сетку, которая разобьет его на квадраты (рис.2). Такая сетка называется растром. После закрасим каждый квадрат в черный или белый. Для более точного изображения, присвоим белый цвет квадратам, имеющие большую часть белого цвета, а черным закрасим квадраты, имеющие большую часть черного цвета (рис.3).



(рис.1)

(рис.2)

(рис.3)

Таким образом мы получили растровое изображение, которое состоит из множества квадратов, называемыми пикселями.

Пиксель (англ. pixel = pictureelement, элемент рисунка) – это наименьший элемент рисунка, которому присвоен определенный цвет. Разбив изображение на отдельные элементы – пиксели, мы выполнили дискретизацию изображения.

Таким образом, из одного единого объекта (изображение), мы получили множество элементов (пикселей).

Теперь попробуем закодировать подобное изображение, полученное в результате дискретизации (рис.4). Чтобы составить двоичный код для данного изображения, заменим черные квадраты единицами, а белые нулями.

			■	■		■		0	0	0	1	1	0	1	0
		■				■	■	0	0	1	0	0	1	1	0
	■						■	0	1	0	0	0	0	1	0
■	■	■	■	■	■	■	■	1	1	1	1	1	1	1	1
	■					■		0	1	0	0	0	0	1	0
	■		■	■				0	1	0	1	1	0	1	0
	■		■	■				0	1	0	1	1	0	1	0
	■	■	■	■	■			0	1	1	1	1	1	1	0

(рис.4)

(табл.1)

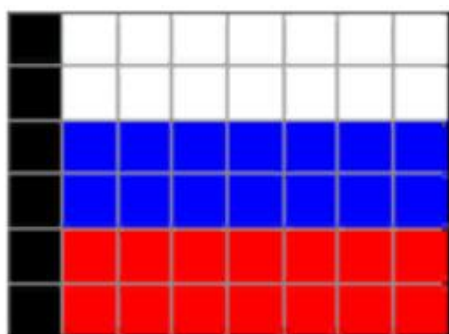
Ширина данного изображения – восемь пикселей, значит для кодирования каждой строки изображения понадобится восемь битов. Построим таблицу, состоящую из нулей и единиц(табл.1). Получим код для первой строки изображения – 00011010. Такую же процедуру проводим с остальными точками. Код изображения будет состоять из длинной цепочки единиц и нулей, поэтому, чтобы уменьшить длину кода, лучше всего использовать шестнадцатеричный код. Тогда для первой строки получаем код – 1A16, а для всего рисунка – 1A2642FF425A5A7E16.

В результате дискретизации получилось закодировать рисунок в двоичном коде, но при этом изображение исказилось. Вместо изображения черного ромба на белом фоне (рис.1), получили набор из черных и белых квадратов (рис.3). Причиной искажения в том, что в некоторых квадратах части исходного рисунка были окрашены разными цветами. Из-за этого часть исходной информации после кодирования была потеряна. Избежать такие потери невозможно, но можно попробовать их уменьшить. Чтобы это сделать,

нужно уменьшить размер пикселя, то есть увеличивать разрешение закодированного изображения.

Разрешение – это величина, определяющая количество пикселей на единицу длины изображения. Разрешение измеряется в пикселях на дюйм –ppi (pixelsperinch). Например, если разрешение изображения 300 ppi– это означает, что на один дюйм (2,54 см) приходится 300 пикселей. Получается, что чем больше разрешение изображение, тем более точнее кодируется изображение и теряется меньше информации, но из-за этого растет объем файла.

Если рисунок имеет несколько цветов, то при кодировании цвета пикселя уже не обойтись одним битом. Например, в показанном ниже изображении (рис.5) всего 4 цвета: белый, синий, красный и черный, значит для кодирования каждого пикселя понадобится 2 бита. Обозначим черный цвет 00, 01 – красный, 10 – синий и 11 – белый. Тогда можно составить следующую таблицу (табл.2):



(рис.5)

00	11	11	11	11	11	11	11
00	11	11	11	11	11	11	11
00	10	10	10	10	10	10	10
00	10	10	10	10	10	10	10
00	01	01	01	01	01	01	01
00	01	01	01	01	01	01	01

(табл.2)

Значит, если закодировать каждый цвет определенным числом или набором чисел, то можно закодировать любой рисунок. Для этого была создана модель цвета RGB (redgreenblue). Любой цвет можно имитировать, используя только три световых луча – красный, зеленый и синий, лишь меняя их яркость. Цвет кодируется тремя цифрами, которые обозначают яркость определенного канала от 0 до 255 (R,G,B). Например, чтобы получить белый цвет, нужно увеличить яркость канала до максимума, значит код белого цвета – (255,255,255), а чтобы получить черный цвет, нужно наоборот, уменьшить

яркость всех трех каналов, значит код черного цвета – (0,0,0). Чтобы сделать светло-синий (голубой) цвет, нужно в синем цвете (0,0,255) одинаково увеличить яркость красного и зеленого каналов, например, (100,100,255) – это голубой цвет. А если равномерно уменьшить яркость всех каналов, то можно получить более темный цвет, например (100,100,100) – это серый цвет.

Всего есть по 256 вариантов яркости для каждого из трех цветов. Это позволяет закодировать 16777216 оттенков, что более чем достаточно для передачи всех цветов исходного изображения. В общей сложности, каждая составляющая занимает 8 бит ( $256=2^8$ ) или 1 байт, а вся информация о каком-либо цвете – 24 бита или 3 байта. Это и есть глубина цвета.

Глубина цвета – это величина, которая обозначает количество бит, используемое для кодирования одного пикселя. Глубина цвета используется тогда, когда нужно рассчитать размер изображения. Например, изображение с разрешением 30×30 пикселей и глубиной цвета 24 бита (3 байта) будет иметь размер 2700 байтов.

Чем больше цветов используется, тем более точным будет изображение, и тем меньше будет потери информации, но и размер файла увеличится. Но если количество цветов в изображении (например, в графике, чертеже) невелико (не более 256), то используется кодирование с цветовой палитрой. Цветовая палитра – это таблица, в которой каждому цвету, заданному в виде составляющей в модели RGB, сопоставляется числовой код.

Например, для кодирования цвета изображения, которое состоит из белого, синего, красного и черного цвета вместо 24 битов, можно использовать лишь 2 бита. Составляем палитру (табл.3), записывая сначала код цвета в RGB, а после кодируем каждый цвет не в виде значений RGB-составляющих, а как номер цвета в палитре.

0	0	0	255	0	0	0	0	255	255	255	255		
цвет 00 <sub>2</sub>			⋮	цвет 01 <sub>2</sub>			⋮	цвет 10 <sub>2</sub>			⋮	цвет 11 <sub>2</sub>	

(табл.3)

Палитра записывается в специальную служебную область в начале файла.

Таким образом, изучив растровое кодирование изображения, можно выделить достоинства и недостатки такого способа кодирования. К достоинствам растрового кодирования относится следующее: оно позволяет закодировать любое изображение; точно передает исходные цвета; изображение можно распечатать. К недостаткам растрового кодирования можно отнести: потерю информации при дискретизации; точность изображения зависит от разрешения; большой размер файла; размер файла не зависит от сложности исходного изображения, а определяется лишь разрешением и глубиной цвета.

### Литература

1. RDG – что это? Принцип работы RGB [Электронный ресурс]. URL: <https://photo7.ru/rgb-chto-jeto-princip-raboty-rgb/> (дата обращения: 26.02.2023).
2. Кодирование графической информации [Электронный ресурс]. URL: <https://studfile.net/preview/3208480/page:8/>(дата обращения: 27.02.2023).
3. Растровое кодирование [Электронный ресурс]. URL: [https://spravochnick.ru/informatika/kodirovanie\\_informacii/rastrovoe\\_kodirovanie/](https://spravochnick.ru/informatika/kodirovanie_informacii/rastrovoe_kodirovanie/)(дата обращения: 2.03.2023).
4. Что такое глубина цвета в фотографии? [Электронный ресурс]. URL:[https://dphotoworld.net/publ/spravochnaja\\_informacija/chto\\_takoe\\_glubina\\_cveta\\_v\\_fotografii/3-1-0-955](https://dphotoworld.net/publ/spravochnaja_informacija/chto_takoe_glubina_cveta_v_fotografii/3-1-0-955) (дата обращения: 2.03.2023).