

# Кодирование графической информации

# Растровое изображение

- Графическая информация, как и информация любого другого типа, хранятся в памяти компьютера в виде двоичных кодов.
- Изображение, состоящее из отдельных точек, каждая из которых имеет свой цвет, называется *растровым изображением*.
- Минимальный элемент такого изображения в полиграфии называется *растр*, а при отображении графики на мониторе минимальный элемент изображения называют *пиксель (pix)*.

# Минимальная единица изображения: пиксель и растр



*Пиксель*

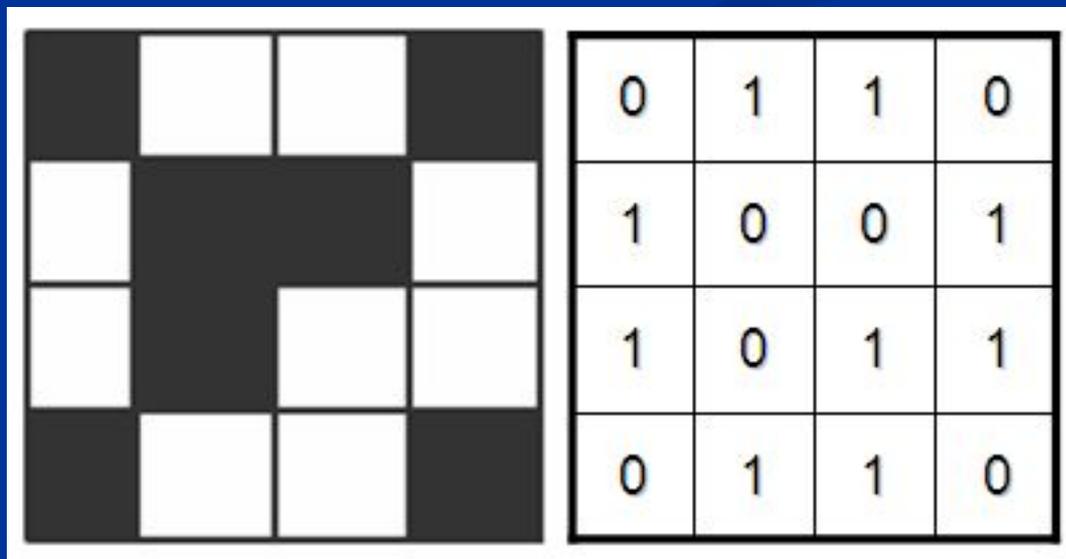
*Растр*

# Глубина цвета

- *Количество бит* , используемых для представления цвета при кодировании *одного пикселя* графического растрового изображения.
- Вычисляется по формуле:  
$$\log_2(\text{количество цветов}) = \text{количество бит}$$

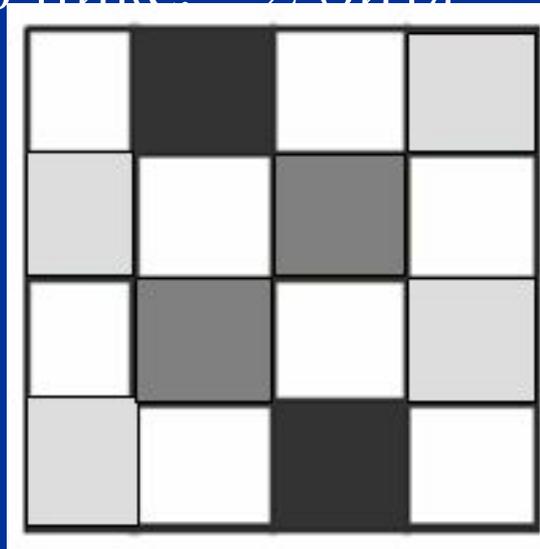
# 1 бит на пиксель – 2 цвета

- Если пиксель изображения может быть раскрашен только в один из **двух** цветов (например, в черный или в белый), то для хранения информации о цвете пикселя достаточно 1 бита памяти ( $\log_2(2(\text{цвета}))=1$  бит).
- Объем всего изображения будет равен числу пикселей в этом изображении.  $4 \times 4 = 16$  бит (2 байта)



# 2 бита на пиксель – 4 цвета

- Если под хранение информации о цвете пикселя выделить 2 бита, то число цветов для раскраски каждого пикселя увеличится до **четырёх** ( $N=2^2=4$ )  
 $\text{Log}_2(4(\text{цвета}))=2$  бита.
- Объем файла изображения в битах будет **вдвое** больше, чем количество составляющих его пикселей.  
 $4 \times 4 = 16$  пикс. \* 2 бита = 32 бита = 4 байта



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

# Вычисление информационного объема изображения

- Глубину цвета умножить на количество пикселей в изображении
  - Например, изображение черно-белое (2 цвета).  
Объем 1 точки равен:  $\log_2(2) = 1$  бит.  
Для картинки 800x600 объем равен:  
 $800 * 600 = 480\ 000$  бит  
 $480\ 000 / 8 = 60\ 000$  байт  
 $60\ 000 / 1024 = 58,6$  Кбайт.

# Вычисление информационного объема изображения (продолжение)

- В изображении используются 256 цветов.  
Объем 1 точки (пикселя) равен:  $\log_2(256) = 8$  бит.  
Для картинки 800x600 объем равен:  
 $800 * 600 * 8 = 3\,840\,000$  бит  
 $3\,840\,000 / 8 = 480\,000$  байт  
 $480\,000 / 1024 = 468,75$  Кбайт.

# Глубина цвета и количество отображаемых цветов

Глубина цвета (I)	Количество отображаемых цветов (N)
8	$2^8 = 256$
16 (High Color)	$2^{16} = 65\,536$
24 (True Color)	$2^{24} = 16\,777\,216$
32 (True Color)	$2^{32} = 4\,294\,967\,296$

# Цветовые модели

- Все объекты окружающего мира можно разделить на:
  - излучающие (светящиеся: солнце, лампа, монитор)
  - отражающие излучение (бумага)
  - пропускающие (стекло)



# RGB – излучающая модель

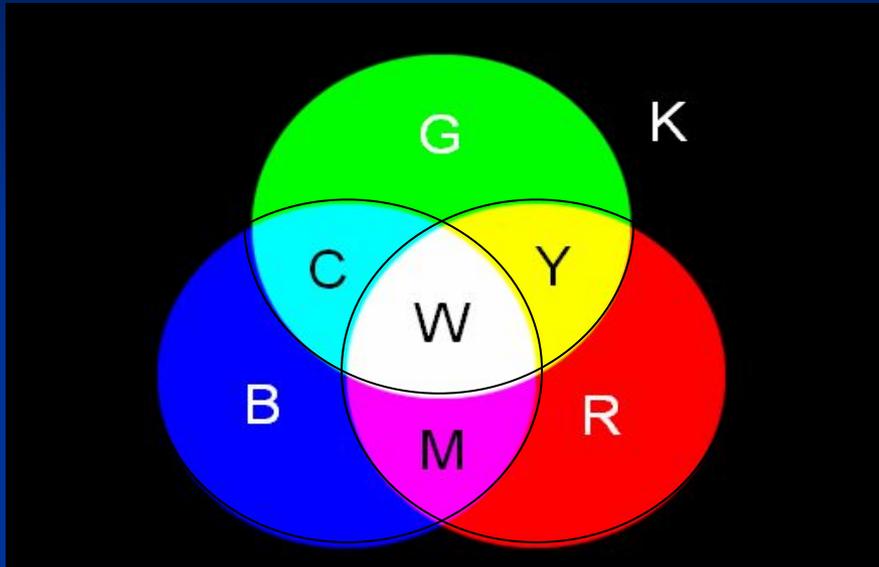
- Используется в телевизорах, мониторах, проекторах, сканерах, цифровых фотоаппаратах.
- Эта модель является **аддитивной** (суммарной): цвета **добавляются к черному (black)** цвету.
- **Основные** цвета:  
красный (Red), зеленый (Green), синий (Blue).
- Их парное сочетание в равных долях дает **дополнительные** цвета:  
 $R+G=Y$  **желтый (Yellow)**  
 $G+B=C$  **голубой (Cyan)**  
 $B+R=M$  **пурпурный (Magenta)**
- Сумма всех трех основных цветов в равных долях дает **белый (White)** цвет:  $R+G+B=W$ .

# RGB

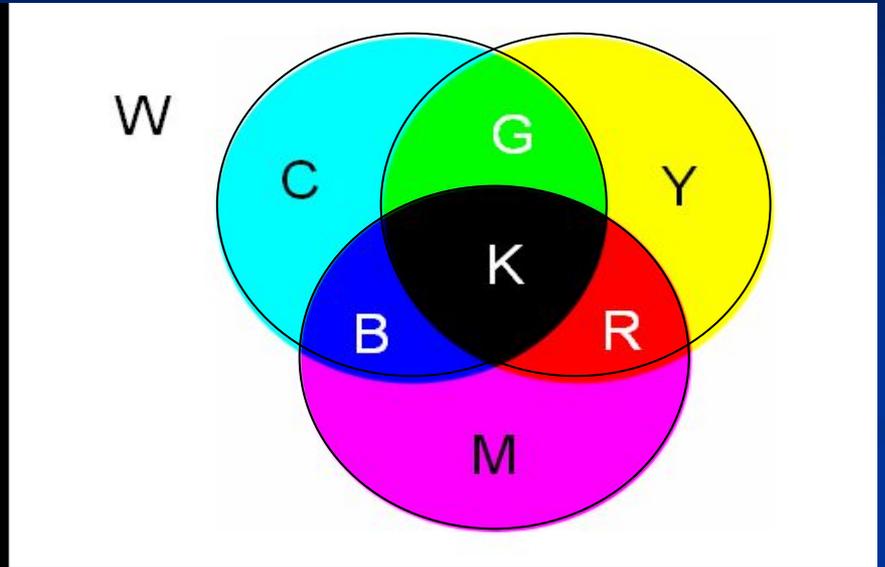
- Для сохранения информации о цвете каждой точки цветного изображения в модели RGB обычно отводится 3 байта (т.е. 24 бита) - по 1 байту (т.е. по 8 бит) под значение цвета каждой составляющей.
- Таким образом, каждая RGB-составляющая может принимать значение в диапазоне от 0 до 255 (всего  $2^8=256$  значений), а каждая точка изображения, при такой системе кодирования может быть окрашена в один из  $2^{3*8}=2^{24}=16\ 777\ 216$  цветов.
- Такой набор цветов принято называть True Color (правдивые цвета), потому что человеческий глаз все равно не в состоянии различить большего разнообразия.

# СМУК – отражающая модель

- Цветовая модель СМУК используется в полиграфии при формировании изображений, предназначенных для печати на бумаге.
- Основными цветами в ней являются те, которые являются дополнительными в модели RGB, т.к. они получаются вычитанием цветов RGB из белого цвета. Поэтому модель СМУК называется **субтрактивной**  
 $C=W-R$   
 $M=W-G$   
 $Y=W-B$
- В теории, сумма  $C+M+Y=K$ , т.е. дает **черный (black)** цвет, но поскольку реальные типографские краски имеют примеси, их цвет не совпадает в точности с теоретически рассчитанным голубым, желтым и пурпурным



Излучающий объект RGB



Отражающий объект CMYK

<i>Изображение</i>	<i>Основа кодирования</i>	<i>Памяти на пиксель</i>		<i>Кол-во цветов</i>
		<i>байт</i>	<i>бит</i>	
Черно-белое	Витмар		1	
Оттенки серого	256 градаций серого	1	8	$2^8=256$
Цветное излучающее	RGB	3	24	$2^{24}=16\,777\,216$
Цветное отражающее	CMYK	4	32	$2^{32}=429\,4967\,296$

# Векторная графика

- Основным логическим элементом является геометрический объект:
  - Простые геометрические фигуры (примитивы - прямоугольник, окружность, эллипс, линия)
  - Составные фигуры - фигуры, построенные из примитивов
  - Цветовые заливки, в том числе градиенты.



# Преимущество векторной графики

- заключается в том, что форму, цвет и пространственное положение составляющих ее объектов можно описывать с помощью математических формул.
- Важным объектом векторной графики является сплайн. **Сплайн** - это кривая, посредством которой описывается та или иная геометрическая фигура. На сплайнах построены современные шрифты **TrueType** и **PostScript**.

# Преимущество векторной графики

- Экономна в плане дискового пространства, необходимого для хранения изображений: так как сохраняется не само изображение, а только некоторые основные данные, используя которые, программа всякий раз воссоздает изображение заново. Кроме того, описание цветовых характеристик почти не увеличивает размер файла.
- Объекты векторной графики легко трансформируются и модифицируются, что не оказывает практически никакого влияния на качество изображения. Масштабирование, поворот, искривление могут быть сведены к паре-тройке элементарных преобразований над векторами.

# Недостатки векторной графики

- Ограничена в живописных средствах: в программах векторной графики практически невозможно создавать фотореалистические изображения.
- Не позволяет автоматизировать ввод графической информации, как это делает сканер для точечной графики.