

Лекция 2

Цветовые модели

Эволюция компьютерных видеосистем

Методы улучшения растровых
изображений

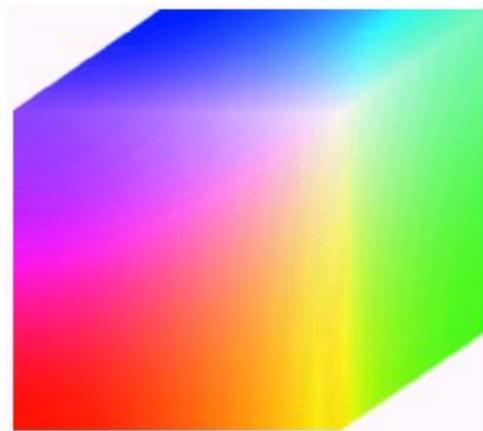
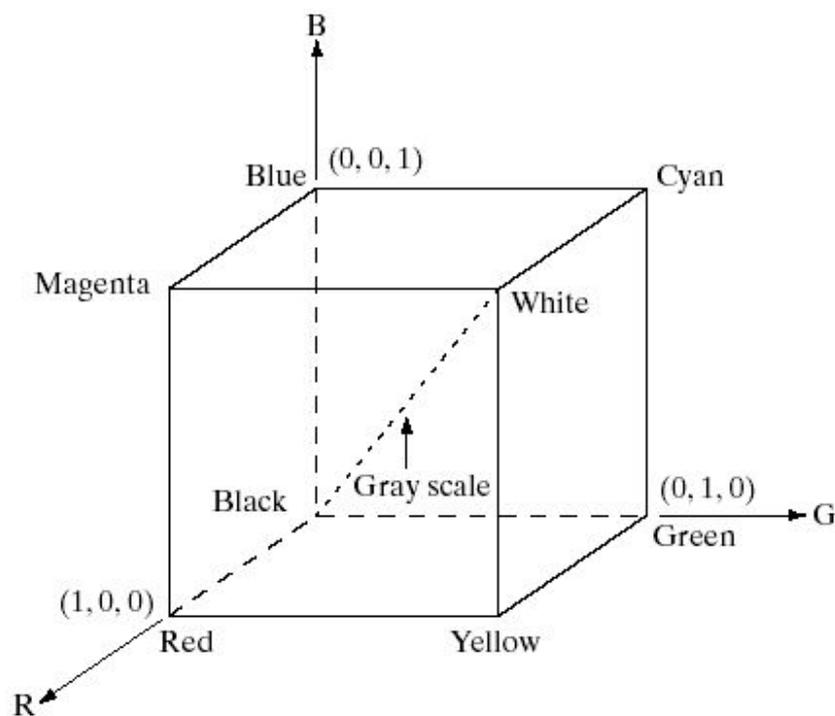
Нижельский С.С.,
ст. преп. каф. СИУ

Новокузнецк, 2008

Цветовая модель RGB

Модель RedGreenBlue основана на первичных основных цветах источников света

Модель используется для описания цветов, полученных на устройствах излучения – мониторах, проекторах

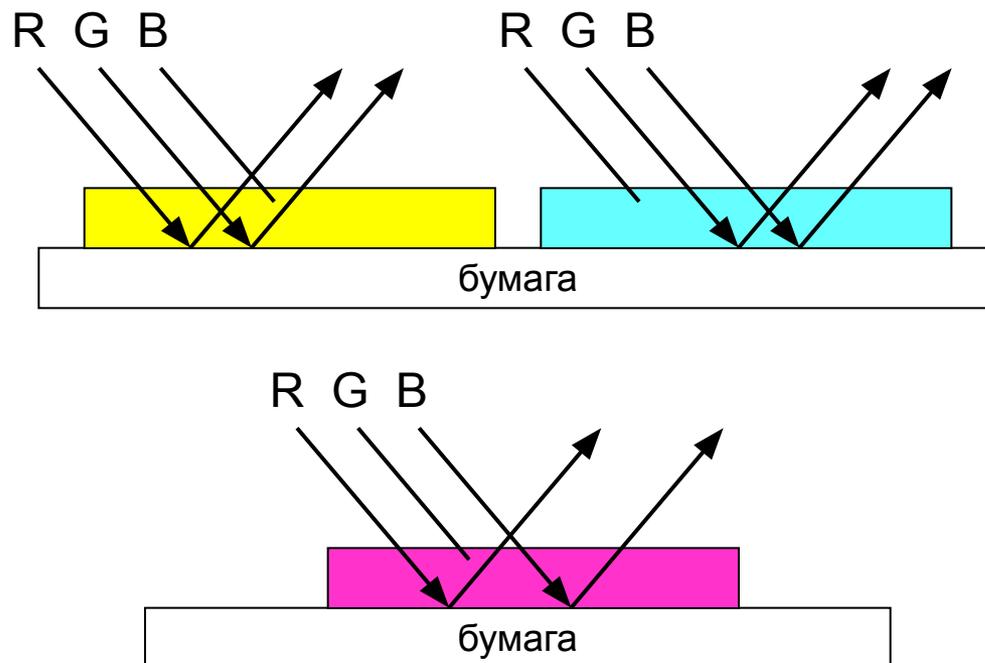
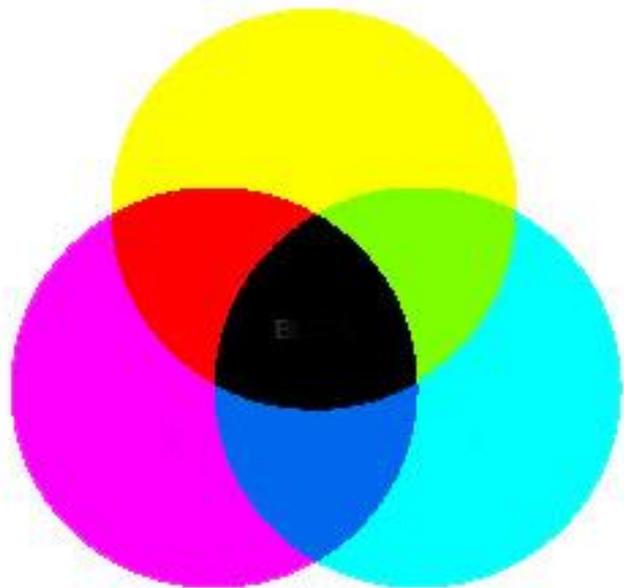


Цветовое пространство модели представляет собой куб

Цветовая модель CMY

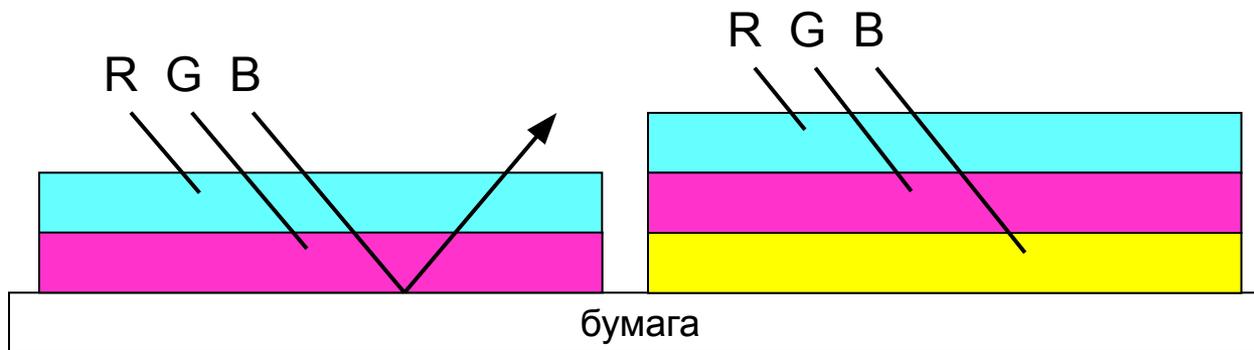
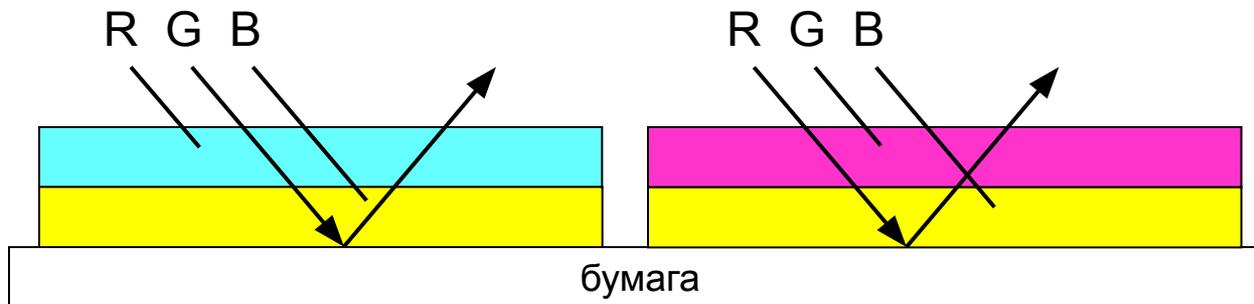
Модель формирует цвета из основных субтрактивных цветов: Cyan (голубой), Magenta (пурпурный), Yellow (желтый)

Используется для устройств основанных на поглощении (вычитании) цветов – цветных принтеров



Цветовая модель CMY

Субтрактивность для двух и трех красок



Цветовая модель CMY и CMYK

Цветные печатные устройства требуют представления входных данных в модели CMY, либо осуществляют преобразование данных из RGB в CMY:

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

при условии, что компоненты цвета кодируются числами в диапазоне [0, 1]

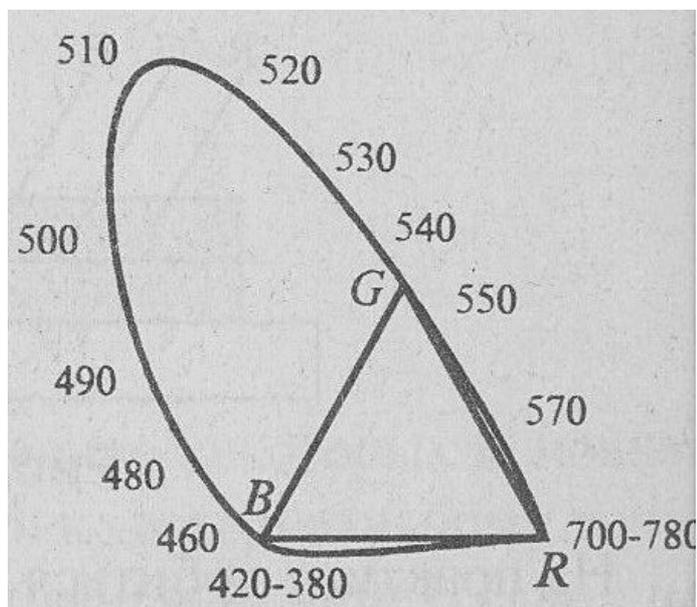
На практике получить черный цвет смешением трех цветов CMY сложно, поэтому в принтерах используют четвертую краску – черную (black)

в этом случае модель называют **CMYK**

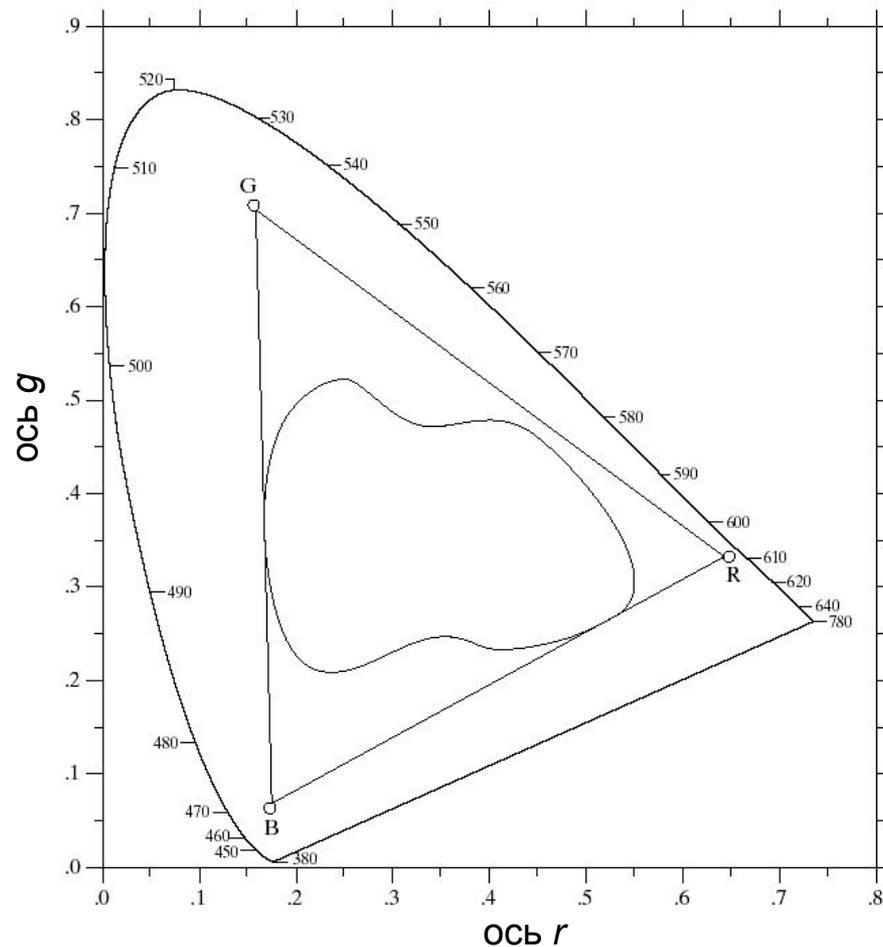
Цветовые модели

Ограниченность цветковых моделей

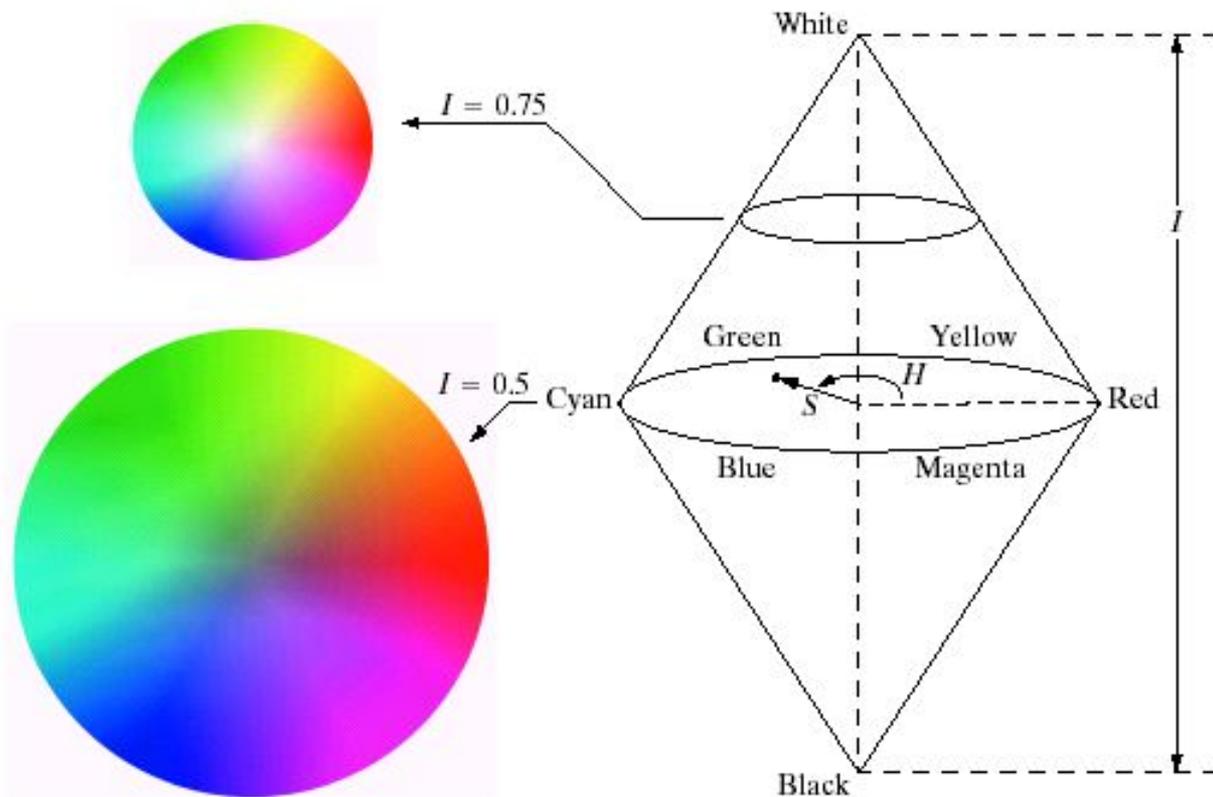
Блинова Т.А., Порев В.Н. Компьютерная графика, 2006 г.



Гонсалес Р., Вудс Р. Цифровая обработка изображений, 2005 г.



Другие цветовые модели



HSI

Hue (цветовой тон)

Saturation (насыщенность)

Intensity (интенсивность)

BHS

Brightness (яркость)

Hue (цветовой тон)

Saturation

(насыщенность)

HLS

Hue (цветовой тон)

Lighting (яркость)

Saturation

(насыщенность)

HSV

Hue (цветовой тон)

Saturation

(насыщенность)

Value (яркость)

Кодирование цвета, палитра

Код цвета	R	G	B	Название цвета
0	0	0	0	черный
1	128	0	0	темно-красный
2	0	128	0	зеленый
3	128	128	0	коричнево-зеленый
4	0	0	128	темно-синий
5	128	0	128	темно-пурпурный
6	0	128	128	сине-зеленый
7	128	128	128	серый 50%
8	192	192	192	серый 25%
9	255	0	0	красный
10	0	255	0	ярко-зеленый
11	255	255	0	желтый
12	0	0	255	синий
13	255	0	255	пурпурный
14	0	255	255	голубой
15	255	255	255	белый

Для работы с цветными изображениями на компьютере, необходимо представить цвет в виде чисел – закодировать его

При глубине цвета True Color каждый компонент кодируется байтом, т.е. каждый из RGB цветов имеет 256 градаций

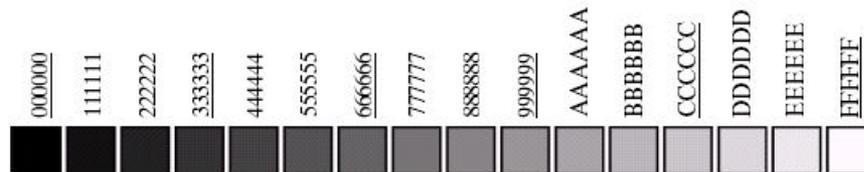
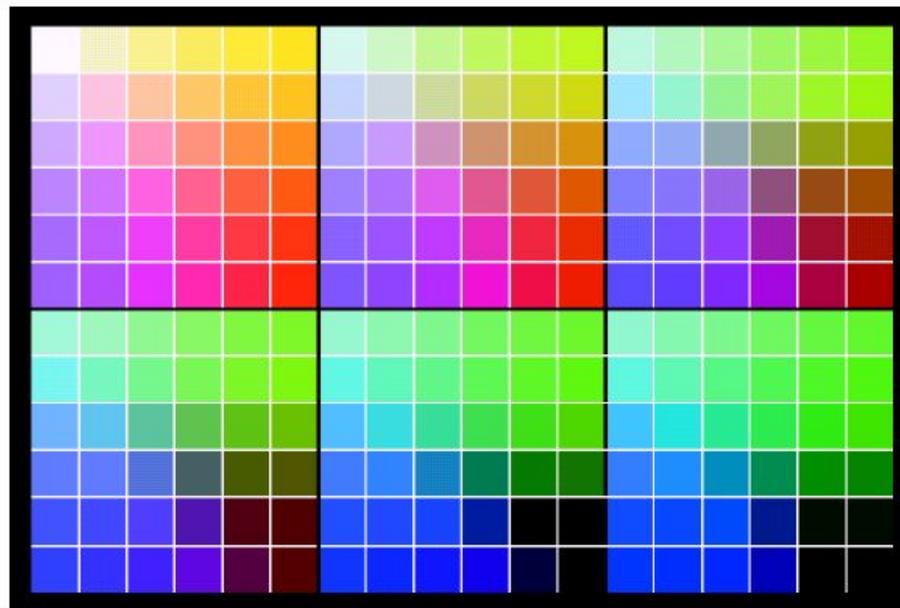
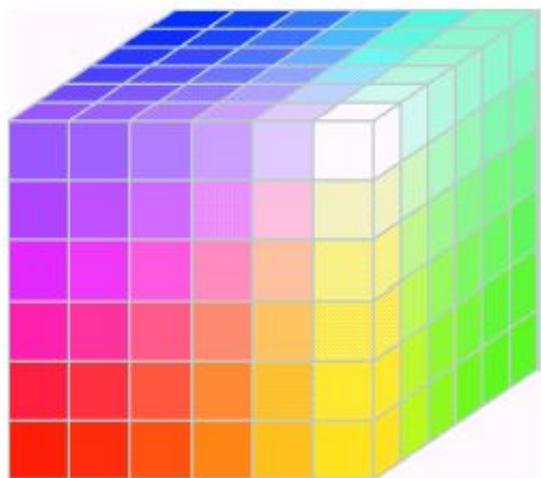
$$R = 0..255, G = 0..255, B = 0..255$$

$$256 \times 256 \times 256 =$$

$$= 16\,777\,216 \text{ цветов } (2^{24})$$

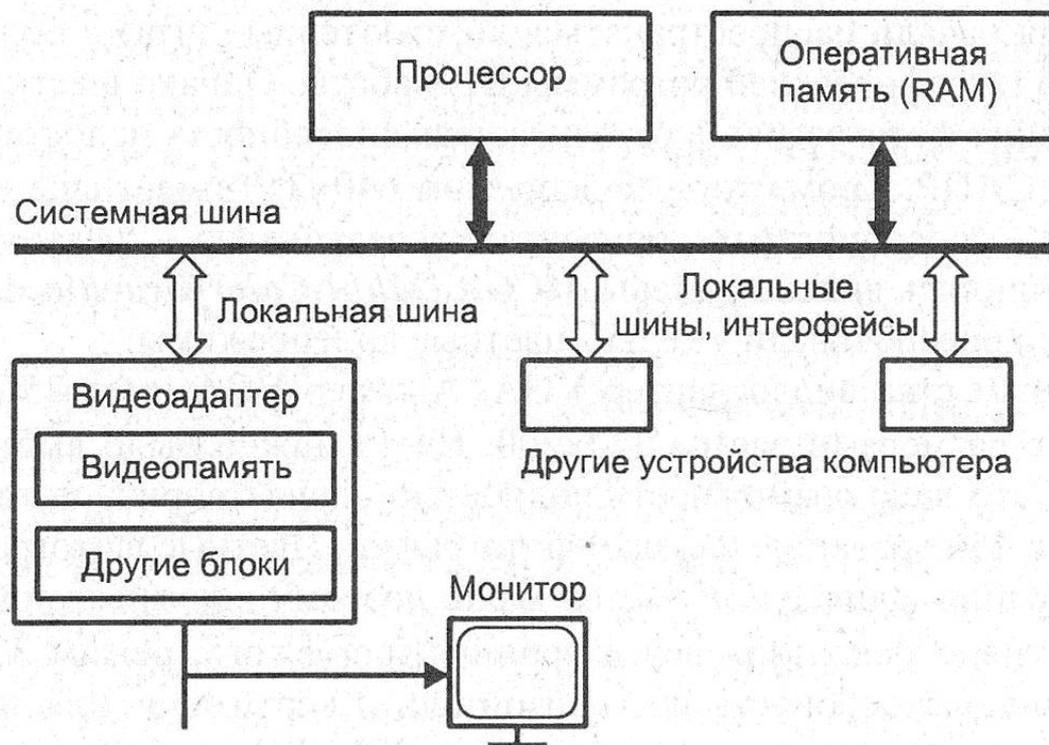
Кодирование цвета, палитра

Палитра фиксированных 216-ти RGB цветов, являющихся общими для большинства систем



Система счисления	Значения RGB компонент					
Шестнадцатеричная	0	33	66	99	CC	FF
Десятичная	0	51	102	153	204	255

Эволюция компьютерных видеосистем



PCI (Peripheral Component Interconnect)

32-х разрядная шина с базовой частотой 33 МГц и скоростью обмена до 132 Мбайт/с

AGP (Accelerated Graphics Port)

64-х разрядная шина с базовой частотой 66 МГц и скоростью обмена 528 Мбайт/с и более

Эволюция компьютерных видеосистем

- **MDA** (Monochrome Display Adapter) – текстовый режим (80×25 символов)
- **HGC** (Hercules Graphic Card) – графический черно-белый режим (720×350)
- **CGA** (Color Graphic Adapter) – графические режимы: черно-белый (640×200) и цветной (320×200, глубина цвета – 2 бита на пиксел)
- **EGA** (Enhanced Graphic Adapter) – цветной графический режим (640×350, 16 цветов из 64-цветной палитры)
- **MCGA** (Multi-Color Graphic Array) и
- **VGA** (Video Graphic Array) – 256-цветный графический режим (320×200, палитра 256 тысяч цветов); 16-цветный графический режим (640×480)
...800×600, 1024×768 для 16 и 256 цветов...
- **SVGA** (SuperVGA) – в 1989 г. ассоциация *VESA* разработала стандарт для видеоадаптеров SVGA

~~**True Color** – фирма *TrueVision* первой достигла глубины цвета в 24 бит на пиксел с видеоадаптером *Targa 24*~~

Эволюция компьютерных видеосистем

Видеопамять сохраняет растровое изображение, демонстрирующееся на экране монитора

Необходимый объем видеопамяти, например, для 24-битного видеорежима при разрешении экрана 1024×768, составляет:

$$1024 \times 768 \times 24 = 18\,874\,368 \text{ бит} = 2,25 \text{ Мбайт}$$

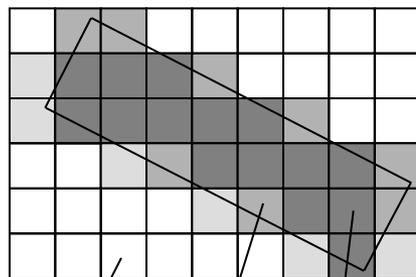
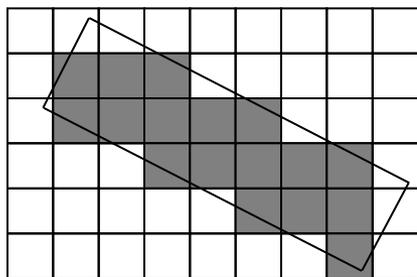
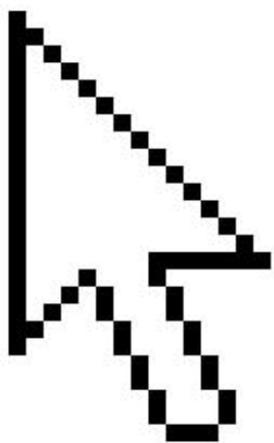
Графический процессор (GPU – Graphic Processor Unit) – визуализация кадрового буфера, растровые операции, построение графических примитивов, операции 3D-графики, наложение текстур, выполнение шейдеров

Графические интерфейсы – программные средства, обеспечивающие взаимодействие с аппаратными ресурсами компьютера

- *OpenGL* – разработан Silicon Graphics, поддерживается многими ОС и производителями видеоадаптеров
- *DirectX* – разработан Microsoft, только для ОС Windows

Ступенчатый эффект

В растровых системах при невысокой разрешающей способности существует проблема ступенчатого эффекта (*aliasing*)

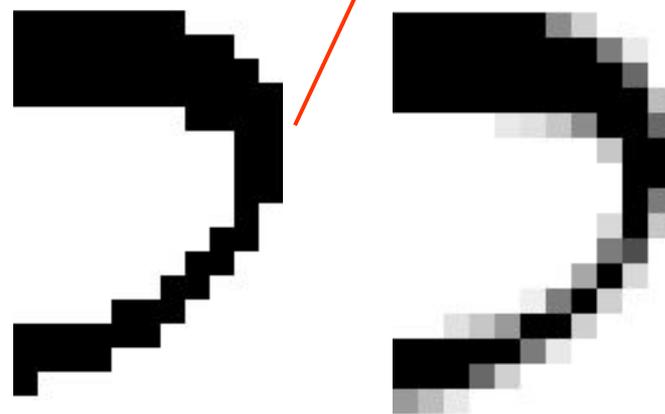


C_ϕ

C_x

C

Aliasing

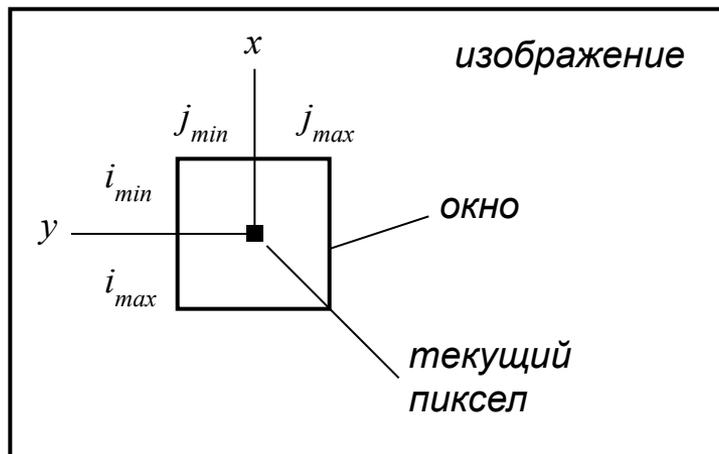


$$C_x = \frac{C \cdot S_x + C_\phi \cdot (S - S_x)}{S}$$

Ступенчатый эффект

Методы сглаживания растровых изображений можно разделить на две большие группы:

- сглаживание в процессе формирования (вывода) изображения
- сглаживание (обработка) существующих изображений – чаще всего используют алгоритмы цифровой фильтрации, и одним из них является локальная фильтрация



$$F_{x,y} = \frac{1}{K} \sum_{i=i_{\min}}^{i_{\max}} \sum_{j=j_{\min}}^{j_{\max}} P_{x+j,y+i} \cdot M_{i-i_{\min},j-j_{\min}}$$

P – цвет текущего пиксела

F – новое значение цвета пиксела

K – нормирующий коэффициент

M – маска фильтра

Ступенчатый эффект

Различают два вида фильтрации:

- рекурсивная – в ходе обработки новые значения цвета пикселов записываются в первоначальный растр и используются при вычислении цвета последующих пикселов
- нерекурсивная – при вычислениях используются только исходные значения цветов пикселов

исходное
изображение

$$\frac{1}{4} \begin{vmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix}$$

$$\frac{1}{16} \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$$



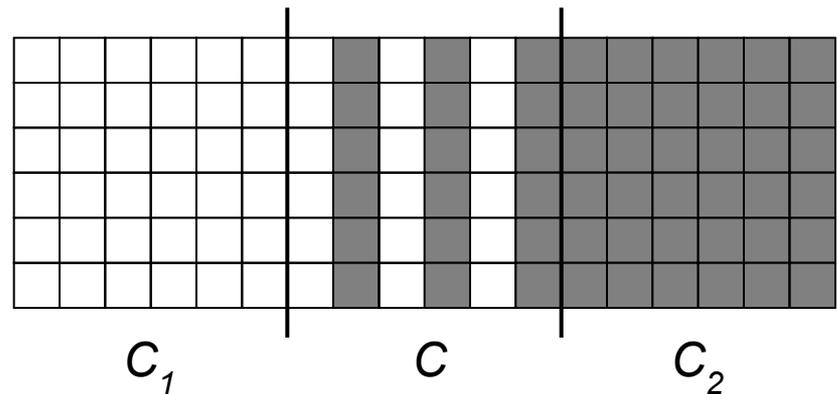
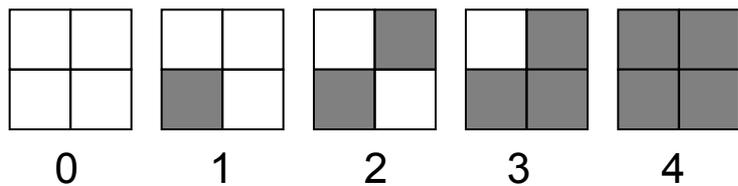
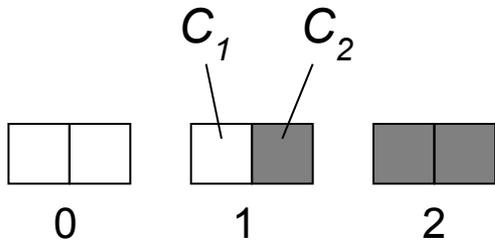
Дизеринг

Если для монитора основной проблемой сейчас является низкая разрешающая способность, то для печатающих устройств ограничением выступает количество воспроизводимых цветов

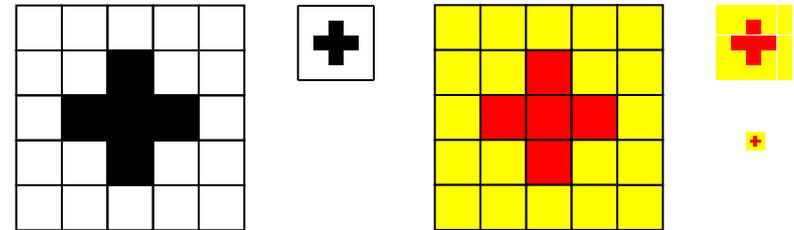
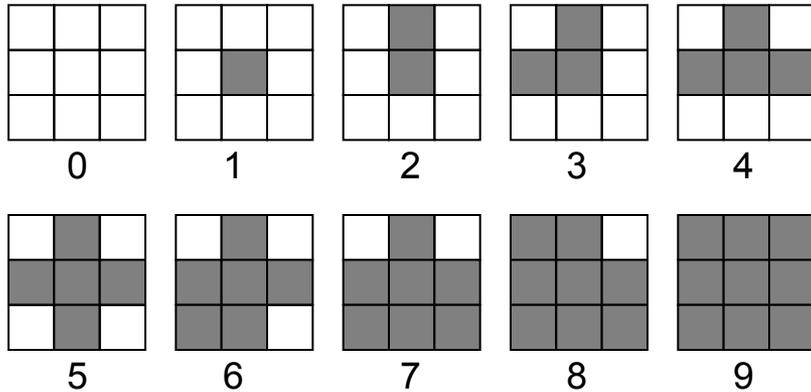
Если достаточно близко расположить маленькие точки разных цветов, то они будут восприниматься как одна точка с некоторым усредненным цветом

dithering – разрежение, дрожание

$$C = \frac{C_1 + C_2}{2}$$



Дизеринг



$C_1 (R_1, G_1, B_1) = (255, 255, 255)$ – белый

$C_2 (R_2, G_2, B_2) = (0, 0, 0)$ – черный

$$C = \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} (S_1 R_1 + S_2 R_2) / S \\ (S_1 G_1 + S_2 G_2) / S \\ (S_1 B_1 + S_2 B_2) / S \end{bmatrix} = \begin{bmatrix} 204 \\ 204 \\ 204 \end{bmatrix}$$

$C_1 (R_1, G_1, B_1) = (255, 255, 0)$ – желтый

$C_2 (R_2, G_2, B_2) = (255, 0, 0)$ – красный

$C (R, G, B) = (255, 204, 0)$ – оранжевый

Цвет некоторой ячейки C можно оценить соотношением:

$$C = \frac{S_1 C_1 + S_2 C_2}{S}$$

где S – общая площадь, S_1 и S_2 – площади, занятые пикселями цветов C_1 и C_2

Дизеринг

Ячейки размером $n \times n$ образуют растр с разрешающей способностью в n раз меньшей, чем у исходного растра, а глубина цвета возрастает пропорционально n^2

Для характеристики изображений, созданных методом дизеринга, используют термин – *линиатура растра*

Линиатура определяется количеством линий (ячеек) на единицу длины и обозначается как *lpi* (по аналогии с dpi)

Реализацию дизеринга для изображения размером $p \times q$, начинают с определения размера ячейки $m \times n$, обеспечивающей необходимое количество цветов

Дизеринг

Непосредственное преобразование можно осуществить двумя способами:

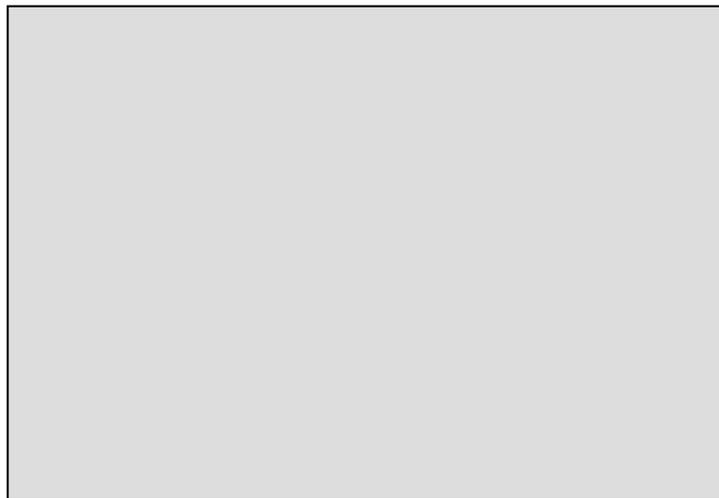
1) каждый пиксел исходного изображения заменяется ячейкой из $m \times n$ пикселов, при этом размер растра увеличится и составит $mp \times nq$ пикселов

2) без изменения размера

- определяют координаты пиксела (x, y) исходного растра
- определяют цвет пиксела (x, y)
- находят номер ячейки (k) , наиболее точно представляющей требуемый цвет
- по координатам (x, y) вычисляют координаты пиксела внутри ячейки
$$\begin{aligned}x_k &= x \bmod m, \\y_k &= y \bmod n\end{aligned}$$
- находят цвет C пиксела ячейки с координатами (x_k, y_k)
- записывают в преобразованный растр пиксел (x, y) с цветом C

Дизеринг

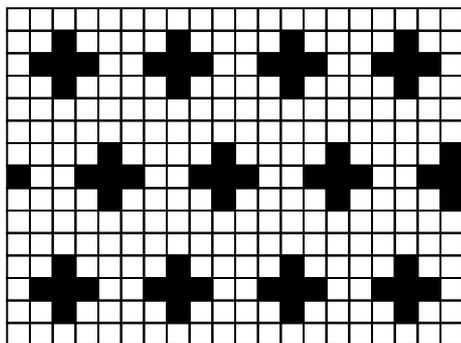
Квадратный растр



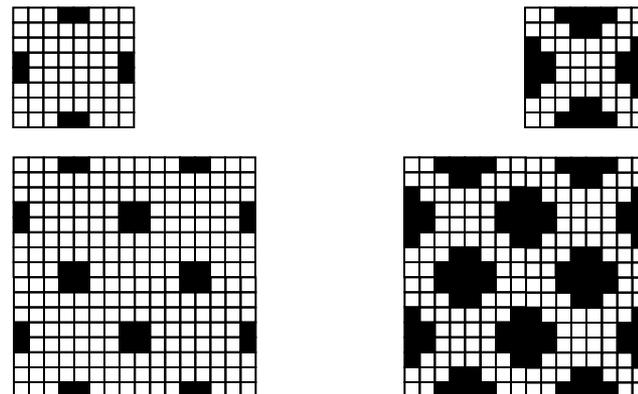
Диагональный растр



Сдвиг ячеек

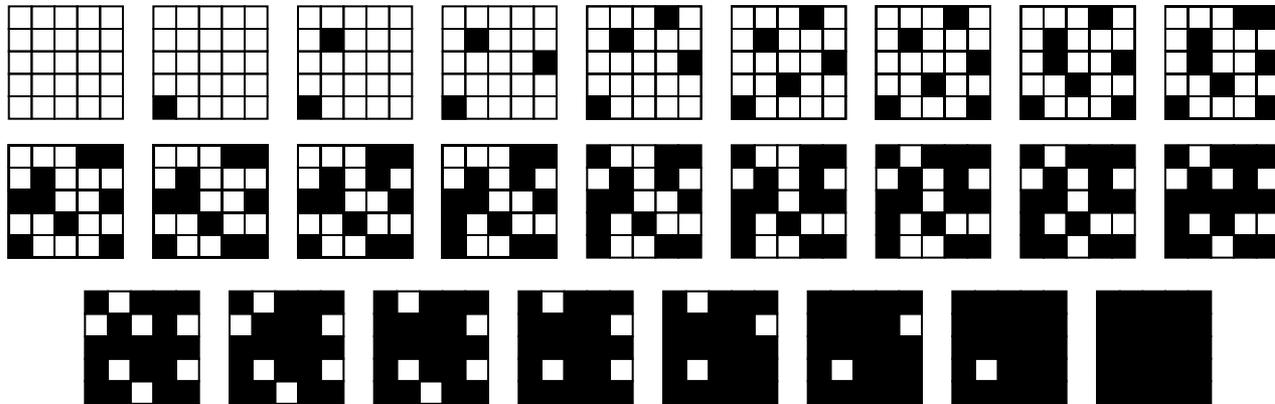


Ячейки другого типа



Дизеринг

Метод частотной модуляции – переменная плотность расположения точек постоянного размера



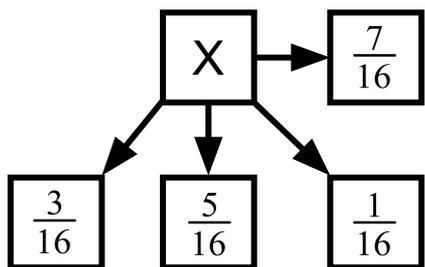
Общим недостатком методов, использующих регулярное расположение одинаковых ячеек, является то, что всегда образуется текстура, появляется муар, лишние контуры

Дизеринг

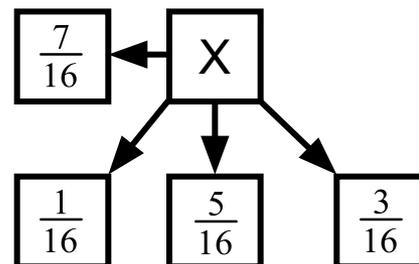
Существуют методы дизеринга, в которых не используются повторяемые ячейки, например, один из самых известных –

метод **“error diffusion” Флойда-Стейнберга**

Сканирование слева
направо 



Обратное направление
сканирования 



При обработке первой строки растра для первого пиксела цвет (С) заменяется на ближайший из возможных (X). Для этого пиксела вычисляется ошибка $E = C - X$. Эта ошибка в различных пропорциях распределяется по соседним пикселам