

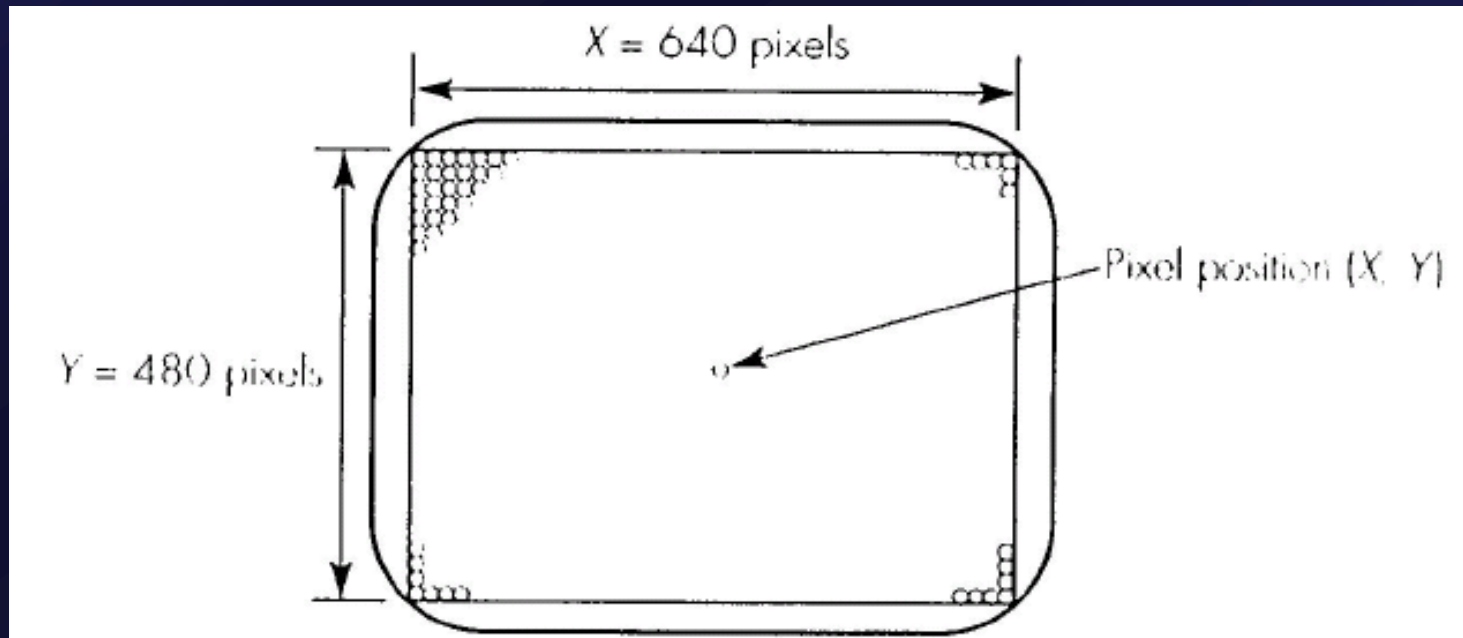
Цифровой анализ изображений

Лекция 4

Компьютерное представление изображений, Форматы файлов

Представление изображений - разрешение

- ❑ Изображения: оцифрованные изображения (например, фотографии, сканированные документы), компьютерные изображения.
- ❑ Изображение представляется двумерной матрицы элементов изображения (называемых пикселями).
- ❑ Пример:



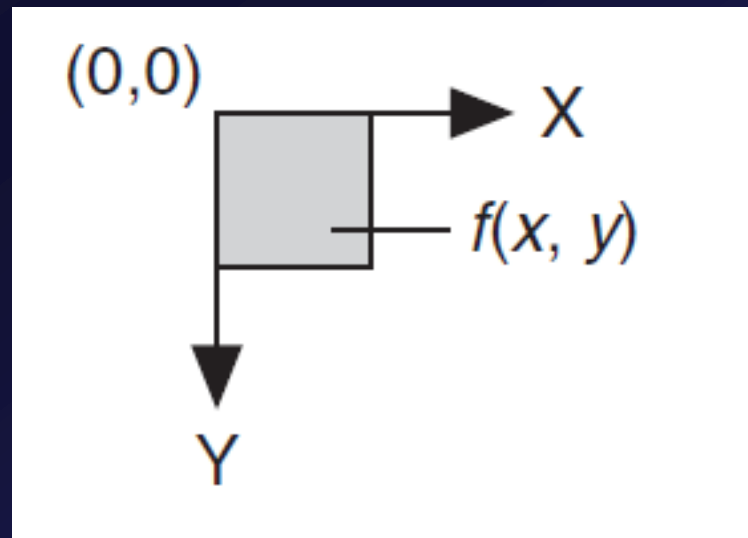
Изображение - 2D массив значений, представляющих интенсивность света. Для направления обработки изображений, термин изображение относится к цифровому изображению.

Изображение является функцией интенсивности света

$$f(x, y)$$

где F является яркость точки (x, y) , а x и y представляют собой пространственные координаты элемента изображения или пиксель.

По соглашению, пространственная привязка пикселя с координатами $(0, 0)$ находится в верхней, левой углу изображения. Обратите внимание на рисунок, что значение x увеличивается движущихся слева направо, а значение y увеличивается сверху вниз.



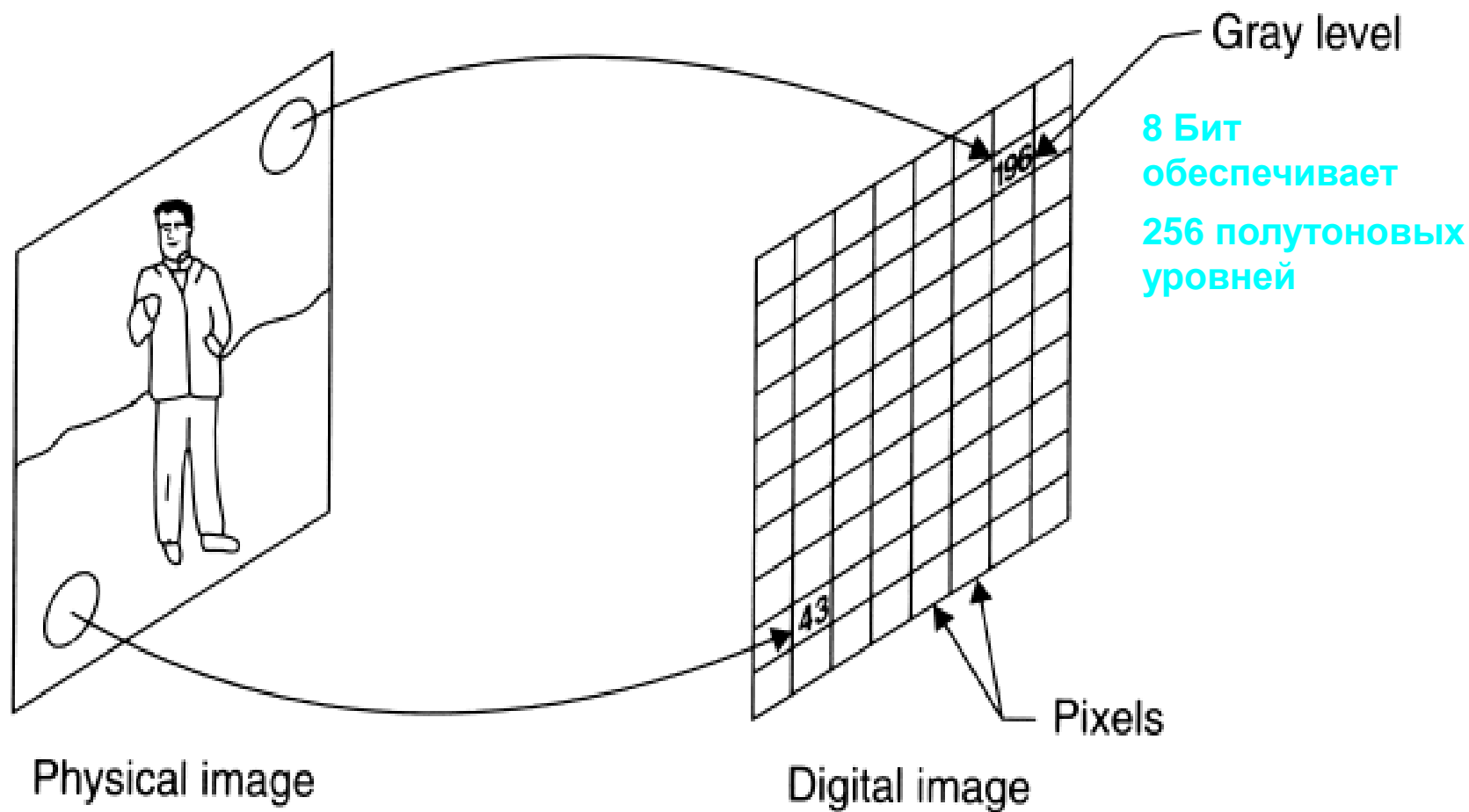


Figure 1-1 A physical image and a corresponding digital image

Представление изображений - разрешение

Пространственное разрешение изображения определяется его количеством строк и столбцов пикселей. Изображение состоит из t столбцов и p строк имеет разрешение $M \times N$. Это изображение имеет t пикселей по горизонтальной оси и p точек вдоль его вертикальной оси.

Это изображение имеет пространственное разрешение (разрешение) 640×480 и разрешением $640 \times 480 = 307200$ пикселей.

Если изображение имеет более высокое разрешение его более тонкие детали можно показать, его визуальное качество лучше.

Представление изображения







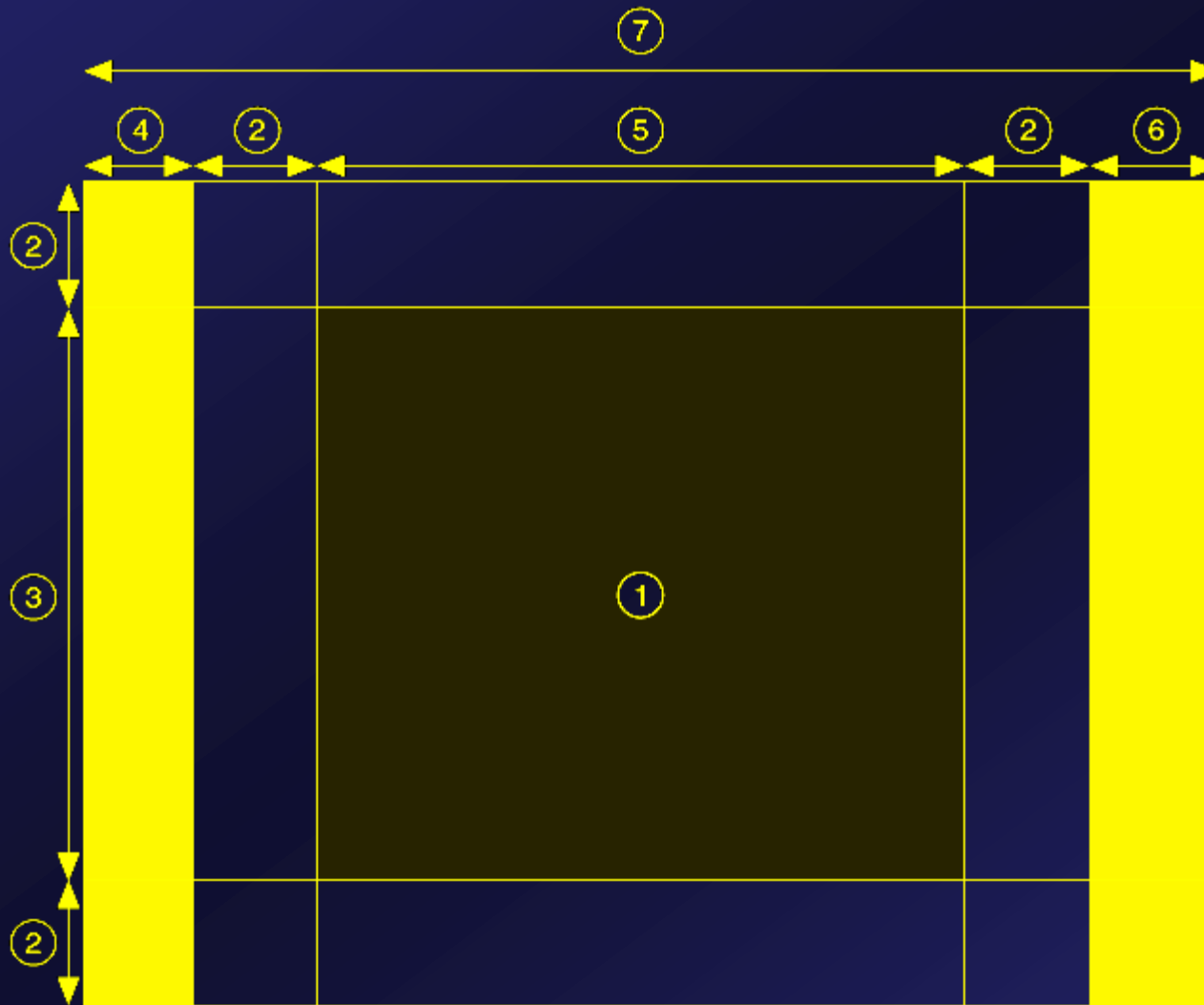
Image Type	Number of Bytes per Pixel Data
8-bit (Unsigned) Integer Grayscale (1 byte or 8-bit)	 <p>8-bit for the grayscale intensity</p>
16-bit (Signed) Integer Grayscale (2 bytes or 16-bit)	 <p>16-bit for the grayscale intensity</p>

Image Type	Number of Bytes per Pixel Data			
32-bit Floating-Point Grayscale (4 bytes or 32-bit)	 32-bit for the grayscale intensity			
RGB Color (4 bytes or 32-bit)	 8-bit for the alpha value (not used)	8-bit for the red intensity	8-bit for the green intensity	8-bit for the blue intensity
HSL Color (4 bytes or 32-bit)	 8-bit not used	8-bit for the hue	8-bit for the saturation	8-bit for the luminance
Complex (8 bytes or 64-bit)	 32-bit floating for the real part 32-bit for the imaginary part			

Представление изображения



1 Image	3 Vertical Resolution	5 Horizontal Resolution	7 Line Width
2 Image Border	4 Left Alignment	6 Right Alignment	

Границы

[illegible]

a.

[illegible]

b.

10	10	10	9	15	11	12	20	16	22	11	8	8	8
10	10	10	9	15	11	12	20	16	22	11	8	8	8
10	10	10	9	15	11	12	20	16	12	11	8	8	8
11	11	11	13	11	12	9	16	17	11	13	14	14	14
12	12	12	8	12	14	12	13	12	14	11	13	13	13
10	10	10	9	13	31	30	32	33	12	13	11	11	11
15	15	15	11	10	30	42	45	31	15	12	10	10	10
13	13	13	12	14	29	40	41	33	13	12	13	13	13
14	14	14	15	12	33	34	36	32	12	14	11	11	11
10	10	10	12	13	14	12	16	12	15	10	9	9	9
10	10	10	8	11	13	15	17	13	14	12	10	10	10
9	9	9	10	12	11	8	15	14	12	11	7	7	7
9	9	9	10	12	11	8	15	14	12	11	7	7	7
9	9	9	10	12	11	8	15	14	12	11	7	7	7

C.

13	11	11	13	11	12	9	16	17	11	13	14	14	13
9	10	10	9	15	11	12	20	16	12	11	8	8	11
9	10	10	9	15	11	12	20	16	12	11	8	8	11
13	11	11	13	11	12	9	16	17	11	13	14	14	13
8	12	12	8	12	14	12	13	12	14	11	13	13	11
9	10	10	9	13	31	30	32	33	12	13	11	11	13
11	15	15	11	10	30	42	45	31	15	12	10	10	12
12	13	13	12	14	29	40	41	33	13	12	13	13	12
15	14	14	15	12	33	34	36	32	12	14	11	11	14
12	10	10	12	13	14	12	16	12	15	10	9	9	10
8	10	10	8	11	13	15	17	13	14	12	10	10	12
10	9	9	10	12	11	8	15	14	12	11	7	7	11
10	9	9	10	12	11	8	15	14	12	11	7	7	11
8	10	10	8	11	13	15	17	13	14	12	10	10	12

d.

Размер границы соответствует самой большой окрестности обработки пикселя. Размер окрестности задается по размеру 2D-массива. Например, если функция использует восемь смежных соседей пикселя для обработки, размер части города 3×3 , что указывает на массив с трех столбцов и трех строк. Установить размер границы, чтобы быть больше или равен половине числа строк или столбцов матрицы 2D округляется до ближайшего целого числа. Например, если функция использует 3×3 окрестности, изображение должно иметь размер границы, по крайней мере, 1; если функция используется 5×5 окрестности, изображение должно иметь размер границы, по крайней мере, 2.

OpenCV уже есть функция реализующая копирование изображения с последующим окружением границей с заданным значением — `cvCopyMakeBorder()`

```
CVAPI(void) cvCopyMakeBorder( const CvArr* src, CvArr* dst, CvPoint offset,  
                             int bordertype, CvScalar value CV_DEFAULT(cvScalarAll(0)));
```

— копирует исходное изображение, окружая его границей

`src` — исходное изображение

`dst` — изображение для сохранения результата

`offset` — смещение (координаты левого верхнего угла целевого изображения)

`bordertype` — тип границы:

```
#define IPL_BORDER_CONSTANT  0 // все пиксели границы заливаются value
```

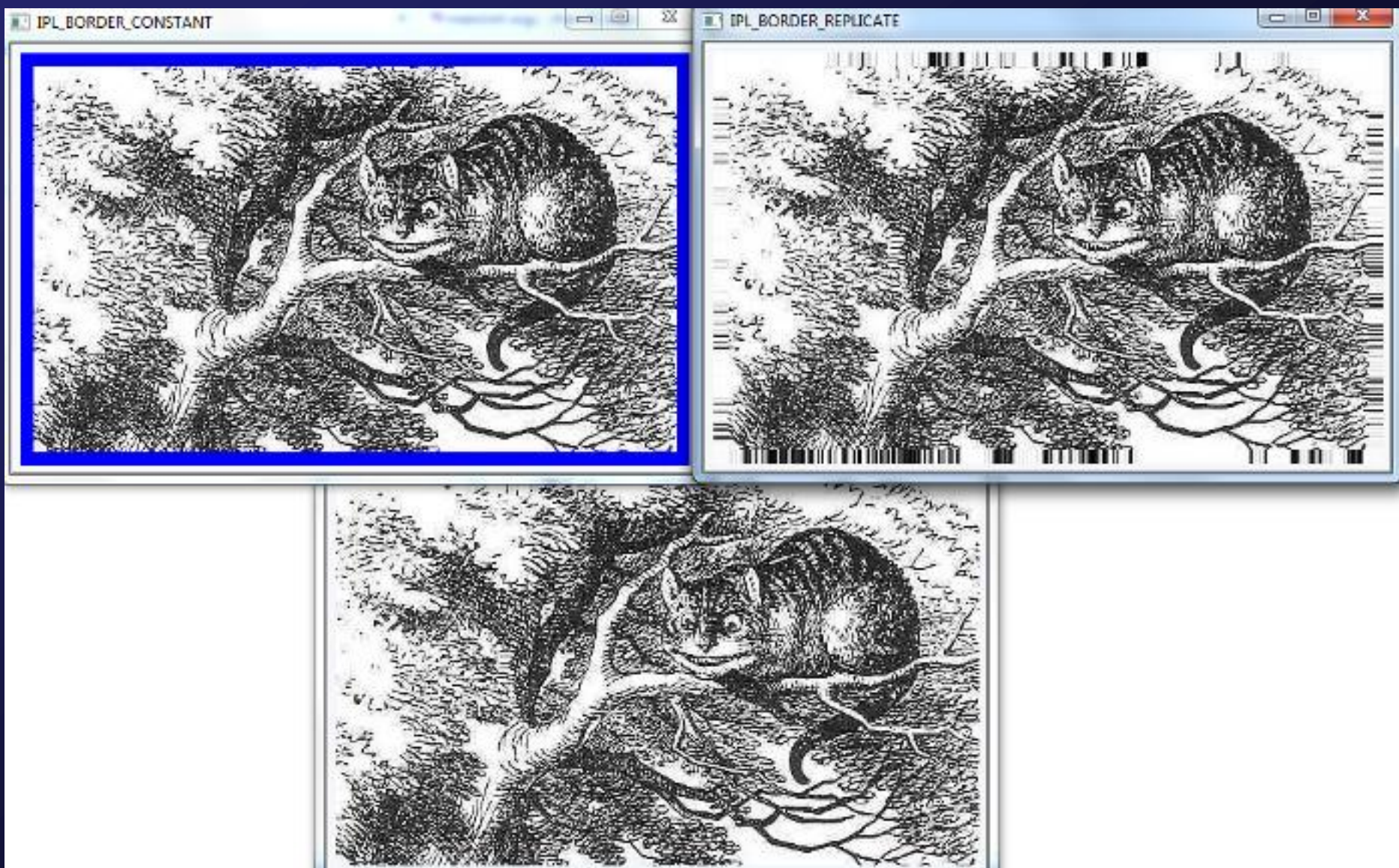
```
#define IPL_BORDER_REPLICATE 1 // крайние пиксели изображения используются для  
заливки
```

```
// другие типы границы пока не поддерживаются:
```

```
#define IPL_BORDER_REFLECT   2 // пока не поддерживается
```

```
#define IPL_BORDER_WRAP      3 // пока не поддерживается
```

`value` — цвет заливки границы



Функция свёртки `cvFilter2D()` внутри себя уже вызывает функцию `cvCopyMakeBorder()` с параметром `IPL_BORDER_REPLICATE`.

Представление изображений - цвет

Глубина Pixel это количество битов, используемых для представления пиксель.

❑ Черно-белое изображение

- ❑ • Если каждый пиксель представлен на 1 бит, то это либо черный и белый.
- ❑ • Если каждый пиксель представлен n битов
 - ❑ → Есть $2^n - 2$ уровней серого между черным и белым
 - ❑ → более высокое качество изображения.

Представление изображений - цвет

- ❑ Каналы изображения: красный, зеленый, синий
 - ❑ • Три основных цвета: красный (R), зеленый (G), синий (B).
 - ❑ • Все цвета могут быть получены путем смешивания соответствующих пропорциях из R, G и B.
 - ❑ • Если мы используем 8 бит на основного цвета, глубина пикселя составляет 24 бита.
- ❑ Есть 2^{24} (более 16 миллионов) цвета.

Представление изображений - цвет

Цвет изображения: яркости и цветности

- Объединить R, G, B сигналов для получения яркости (яркость) сигнала Y и два цветности (цвет) сигналы I и Q:

$$Y = 0.30R + 0.59G + 0.11B$$

$$I = 0.60R - 0.28G - 0.32B$$

$$Q = 0.21R - 0.52G + 0.31B$$

(Различные системы могут использоваться различные способы комбинирования сигналов R, G, B).

Представление изображений - цвет

- Каждый цвет пикселя представлена в эквивалентной области Y, I, Q (вместо R, G, B).

Замечания

1. Когда были черно-белый телевизоры, они получили сигнал яркости.
2. Глаза человека более чувствительны к сигнала яркости, чем сигналов цветности.

Каналы цвета

Цветное изображение кодируется в памяти либо как красного, зеленого и синего (RGB) изображения или оттенка, насыщенности и яркости (HSL) изображения. Цветное изображение пикселей являются составной из четырех значений. RGB изображения магазин информация о цвете с использованием 8 бит каждый для красного, зеленого и синего самолетов. HSL изображения магазин информация о цвете с использованием 8 бит каждый для оттенка, насыщенности и яркости. RGB U64 изображения магазин информация о цвете с использованием 16 бит каждый для красного, зеленого и синего самолетов. В моделях RGB и HSL цвета, дополнительное значение 8-бит не используется в. Это представление называется 4 × 8-битных или 32-битной кодировке.

В цветовой модели RGB U64, дополнительное значение 16-бит не используется в. Это представление называется 4 × 16 бит или 64-битной кодировке.

Alpha plane (not used)



Red or hue plane



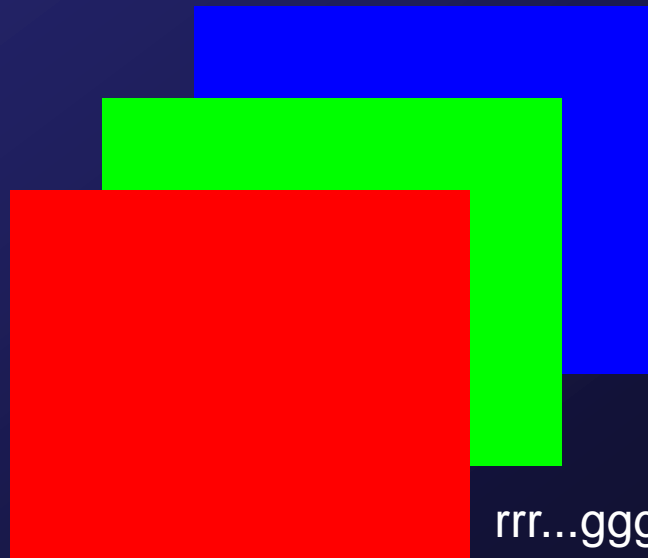
Green or saturation plane



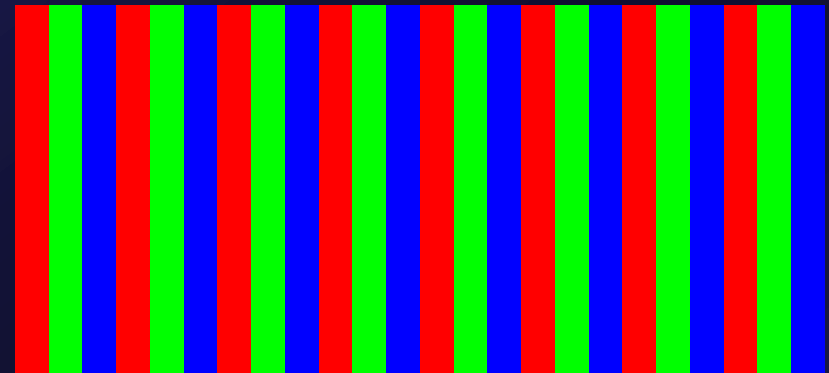
Blue or luminance plane



Представление изображения



rrr...ggg...bbb...).

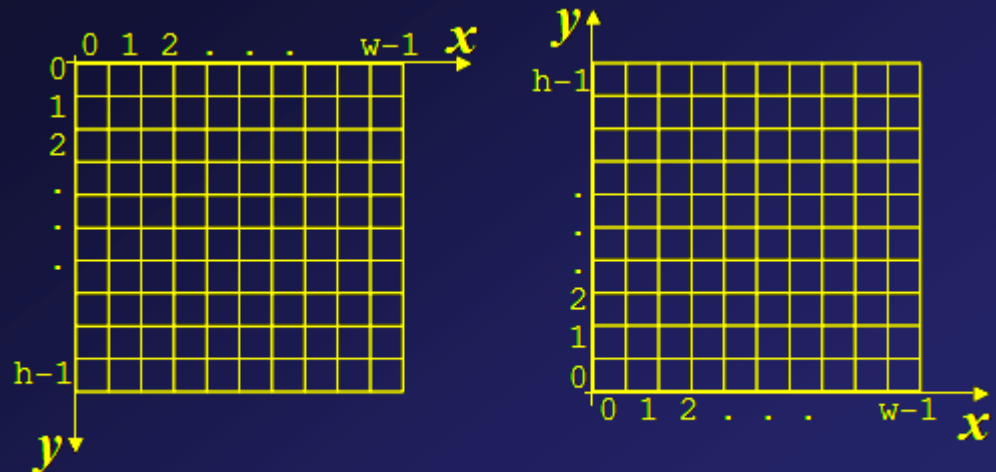


rgbrgbrgb...)

Разделенные и упакованные компоненты пикселей

IM_RGB
IM_MAP
IM_GRAY
IM_BINARY
IM_CMYK
IM_YCBCR
IM_LAB
IM_LUV
IM_XYZ.

IM_BYTE
IM_USHORT
IM_INT
IM_FLOAT
IM_CFLOAT.



Представление буфера данных изображения библиотеки IM (<http://www.tecgraf.puc-rio.br/im/>)

Чтобы создать сырой буфер изображения вы можете просто использовать функцию :

```
int width, height, color_mode, data_type;
```

```
int size = imImageDataSize(width, height, color_mode, data_type);
```

```
void* buffer = malloc(size);
```

ак что, если тип данных IM_FLOAT, мы могли бы написать:

```
float* idata = (float*)buffer;
```

Тогда, чтобы найти пиксель на линии y, колонка x, Компонент D просто написать:

```
float value;
```

```
if (is_packed)
```

```
value = idata[y*width*depth + x*depth + d]
```

```
else
```

```
value = idata[d*width*height + y*width + x]
```

Но обратите внимание, что этот код будет возвращать значения в разных местах пиксельных для сверху вниз и снизу вверх ориентаций.

Форматы изображений

Файл изображения состоит из заголовка с последующим значений пикселей. В зависимости от формата файла, заголовок содержит информацию об изображении о горизонтального и вертикального разрешения на, определения пикселей, и оригинальный палитры. Файлы изображений может также хранить информацию о калибровочных, сопоставление с образцом шаблонов и накладок. Ниже приведены распространенные форматы файлов изображений:

- Bitmap (BMP)
- Tagged image file format (TIFF)
- Portable network graphics (PNG)—

имеет возможность хранения информации об изображении о пространственной калибровки, сопоставление с образцом шаблонов и накладок

- Joint Photographic Experts Group format (JPEG)
- National Instruments internal image file format (AIPD)—пользовательское сохранение вещественных комплексных и других типов изображения

Форматы файлов

Стандартные форматы для 8-битные оттенки серого и цветных изображений RGB являются BMP, TIFF, PNG, JPEG, и AIPD. Стандартные форматы для 16-битные оттенки серого, 64-битной RGB, и сложные изображения, TIF, PNG и AIPD.

Но TIFF не может быть идеальным форматом для многих ситуаций . В W3C стандарты включают только JPEG, GIF и PNG для веб-браузеров . JPEG заставляет изображение , чтобы быть гамма или серый с потерями сжатого . GIF заставляет изображение , которое будет MAP со сжатием LZW . PNG заставляет образ , чтобы быть RGB, MAP , серый или двоичной , со сжатием Deflate . Таким образом, эти качества необходимы , чтобы заставить малые значения для более быстрой загрузки .

JPEG будет использоваться для фотографической содержания , PNG должны использоваться для остальных случаях , но GIF -прежнему является лучшим , чтобы делать простые анимированные изображения .

Для некоторых конкретных случаях, когда формат , необходимых для совместимости исключением , другие форматы имеют меньшее значение . TGA , PCX, RAS , SGI и BMP имеют практически одинаковую полезность .

ЯПО2 должны использоваться для сжатия JPEG - 2000 , будет хорошо, если новая спецификация TIFF включает этот стандарт.

С PNM имеет текстовую заголовок это очень просто , чтобы научить студентов , чтобы они могли на самом деле "видеть" заголовок . Это также формат легко обмениваться изображениями , но это не сделать гораздо больше , чем это.

TIFF и формат GIF также поддержку нескольких изображений. Это не обязательно определяет анимации, пирамиду , ни объем, но несколько раз они используются в этих отношениях.

GIF стал очень популярным , чтобы построить анимации для Web , а так как патент LZW истек Unisys понял, что зарядки использование не собирается работать , и поэтому они не продлевать его . LZW полностью поддерживается в IM.

И.М. также поддерживает видео форматы, такие как AVI и WMV в качестве внешних библиотек . В этих случаях кадры также загружены в виде последовательности отдельных изображений . Звук не поддерживается.

TIFF, JPEG и PNG есть обширный список атрибутов , большинство из них перечислены в документации , но некоторые пользовательские атрибуты могут придумать при чтении изображение из файла.

Сжатие изображения- Концепция

Мы можем сжимать изображения,
потому что

- изображения имеют избыточность
(например, соседние пиксели могут
быть очень похожи друг на друга),
- мы можем выбрать жертвовать
качеством изображения для
уменьшения его размера.

Сжатие изображения- Концепция

Два подхода к сжатию изображений:

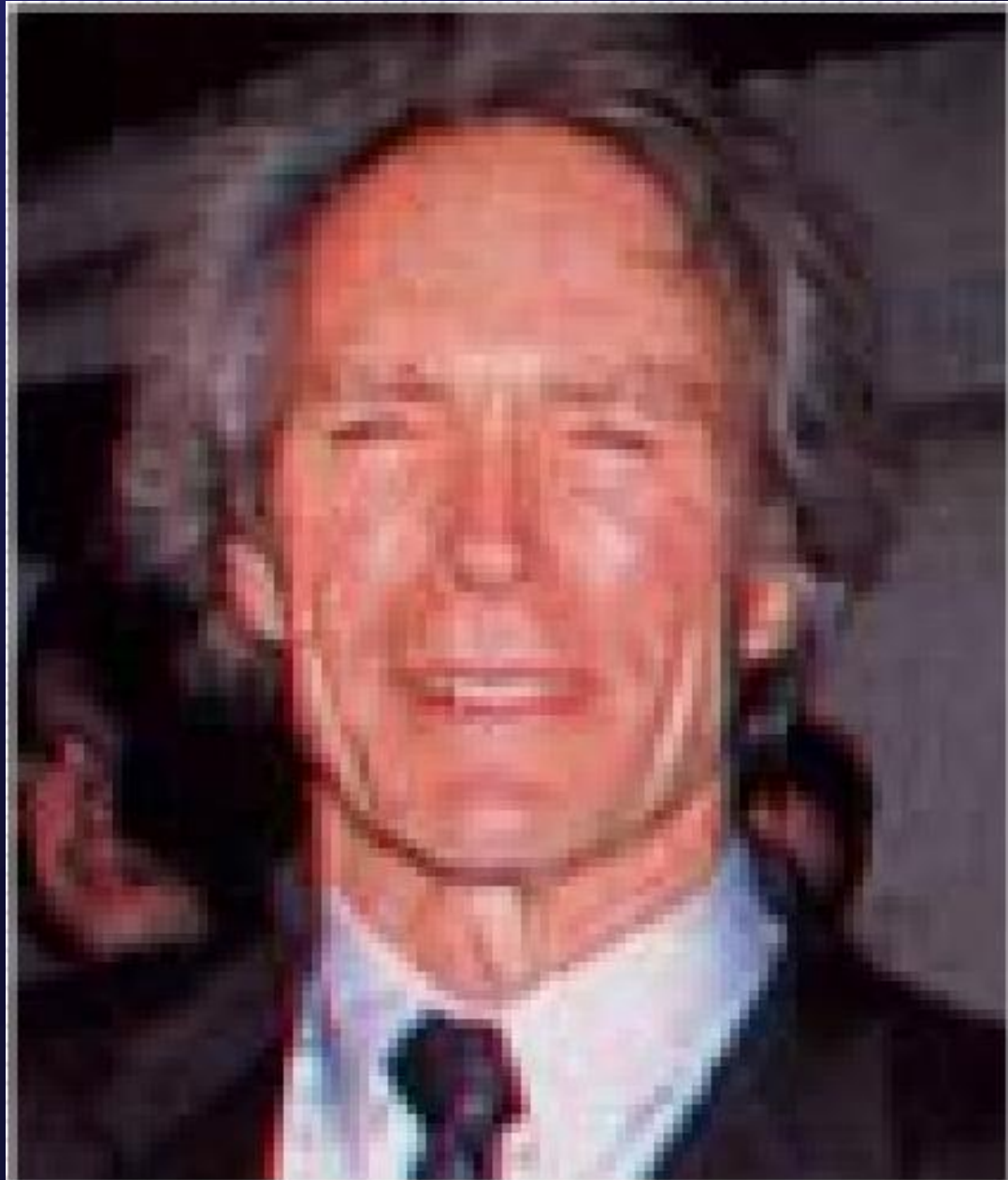
1. Сжатие без потерь: уменьшить размер изображения без потери качества изображения.
2. Сжатие с потерями: уменьшить размер изображения при условии приемлемого ухудшает качество изображения.
 - Это дает большие коэффициенты сжатия.
 - Он широко используется для сжатия изображений.

растровые форматы

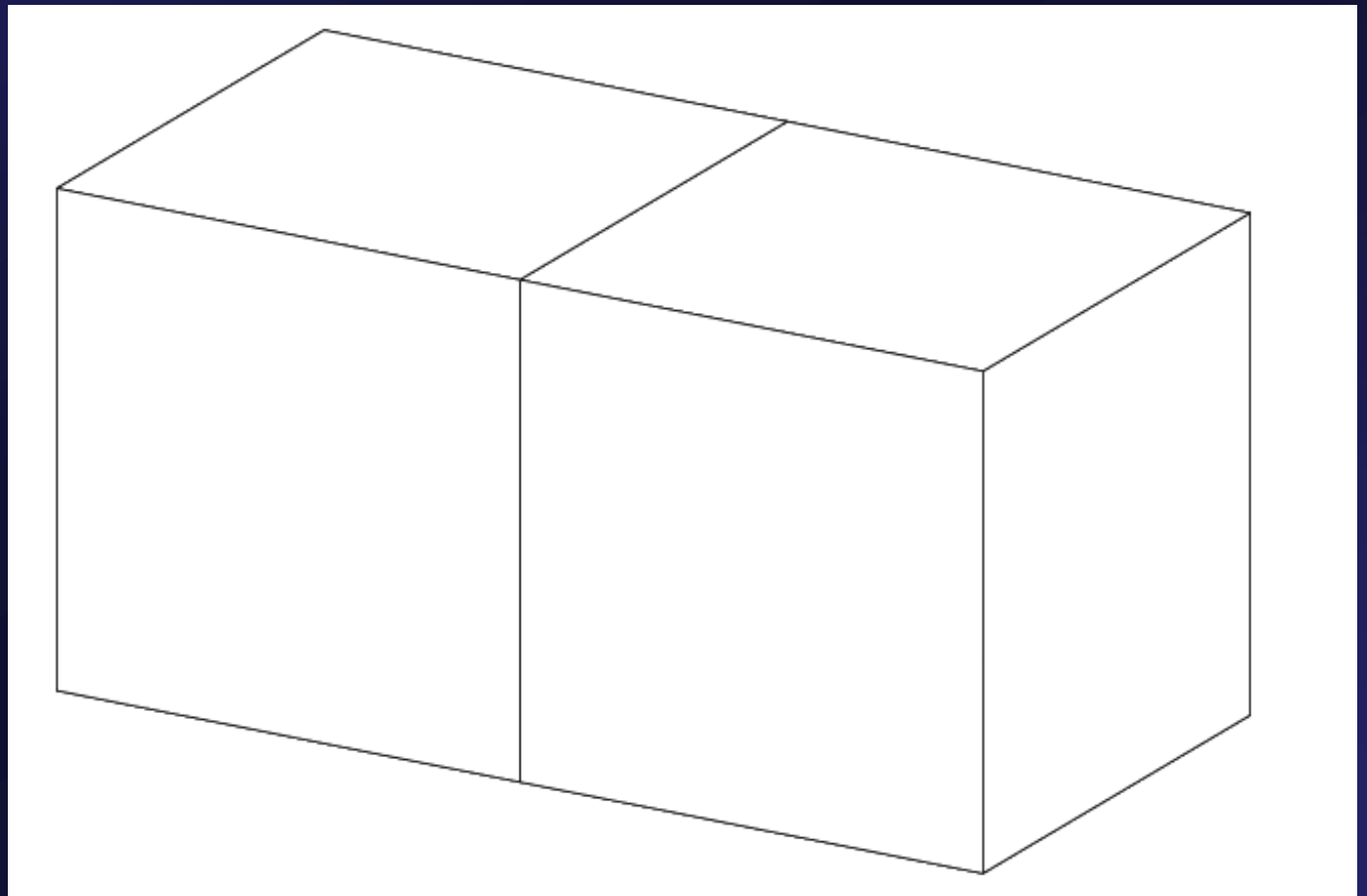
Разбивает изображение на серии цветных точек, называемых «пикселей»

В каждом пикселе число битов определяет максимальное число цветов

- 1 бит = 2 (2¹) цветов
- 2 бита = 4 (2²) цветов
- 4 бита = 16 (2⁴) цвета
- 8 бит = 256 (2⁸) цвета
- 16 бит = 65536 (2¹⁶) цветов
- 24 бита = 16777216 (2²⁴) цвета!



ВМЕСТО ...



Векторные форматы

- преобразование изображения в набор математических описаний форм: кривой, дуги, прямоугольника, сферы и т.д.
- Разрешение-независимыми: масштабируемое без проблемы «pixelating».
- Не все изображения легко описываются в математической форме. Как описать фотографию?

Сравнение

Растр

- Разрешение зависит от
- Подходит для фотографий
- Гладкие тона и тонкие детали
- Большой размер

Вектор

- Разрешение независимым
- Подходит для чертежей, CAD, логотипы
- Плавные изгибы
- Меньший размер

Каковы общие типы графических форматов?

- Расторые: GIF (Graphics Interchange Format), Bitmap, JPEG, TIFF, PBM (Portable Bit Map - binary), PGM (Portable Gray Map – grayscale), PPM (Portable Pixel Map - color), PNM (Portable Any Map – any three), PCD (Photo CD), PNG (Portable Network Graphics), etc.
- Векторные: PS (Postscript), EPS (Embedded Postscript), CDW (CorelDraw), WMF (Windows Metafile), SVG (Scalable Vector Graphics), etc.

CompuServ GIF — Graphics Interchange Format

- Впервые стандартизированы в 1987 году CompuServ (так называемый GIF87a)
- Обновлен в 1989 году включить transparency, переплетения, и анимацию (так называемый GIF89a)
- Используйте LZW (Lempel-Ziv Welch) алгоритм для сжатия
- Максимум 256 цветов, так не очень хорошо работает для фотографий
- Подходит для небольших изображений, таких как иконки
- Простые анимации
- Переплетение против не-переплетение

Bitmaps

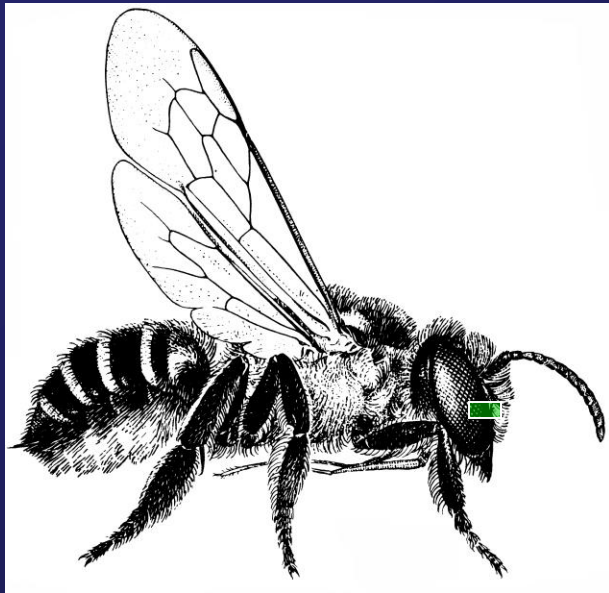
- Может создать большое изображение с 24 или даже 32 бит на пиксель
- Размер файла большой, например, растровое изображение размером 1024x768 с 24 бит на пиксел, по крайней мере? $1024 \times 768 \times 3 = 2 \text{ МБ}$
- Как уменьшить размер? Кодирование длин серий (УПИ) - без потерь
- А как насчет еще меньшего размера? Кодирование с потерями, такой как JPEG.

JPEG (Joint Photographic Experts Group)

- С потерями кодирования
- Как переплетенных GIF-, есть прогрессивные файлы JPEG

TIFF (Tag Image File Format)

- На основе тегов формат изображения
- Возникла в 1986 году в Aldus корпорации (PageMaker), последней версии 6.0
- Разработано Aldus и Microsoft
- Не зависит от платформы
- В основном используется сканеров и настольных издательских систем
- <http://www.libtiff.org/> для TIFF библиотеки
- Поддержка сжатия из CCITT Fax 3 & 4, LZW, JPEG и т.д.
- Поддержка нескольких цветовых пространств: Оттенки серого, RGB, YCbCr, CMYK и т.д.



Без потери компресии:

- RLE (Run Length Encoding)
- LZW (Lempel-Ziv-Welch)
- Huffman
- Zip

Uncompressed

```
0000000000000000000001111
11111111100000000000001111
1111111111111111111
```

RLE

Row 1, 21:0;13:1;13:0;22:1

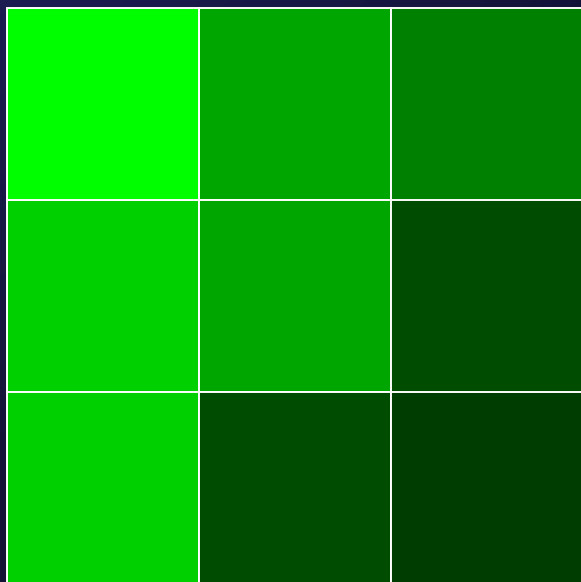
LZW, etc. use algorithms in
addition to RLE

```
01010101010101010101
```



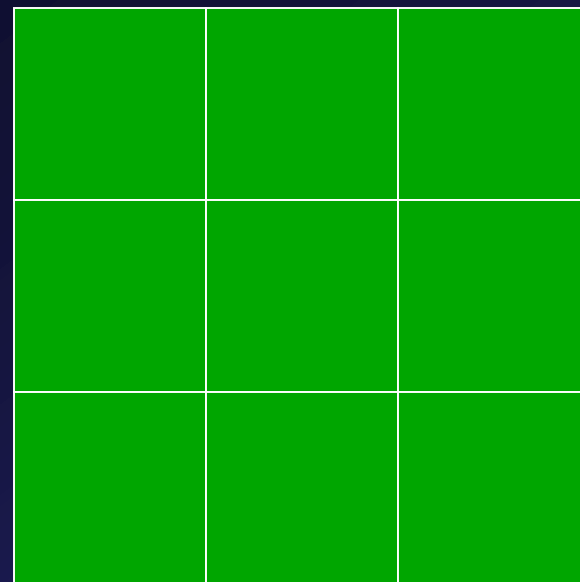
С "Потерями" типы сжатия :

- **JPEG**
- Genuine Fractals
- Mr. SID



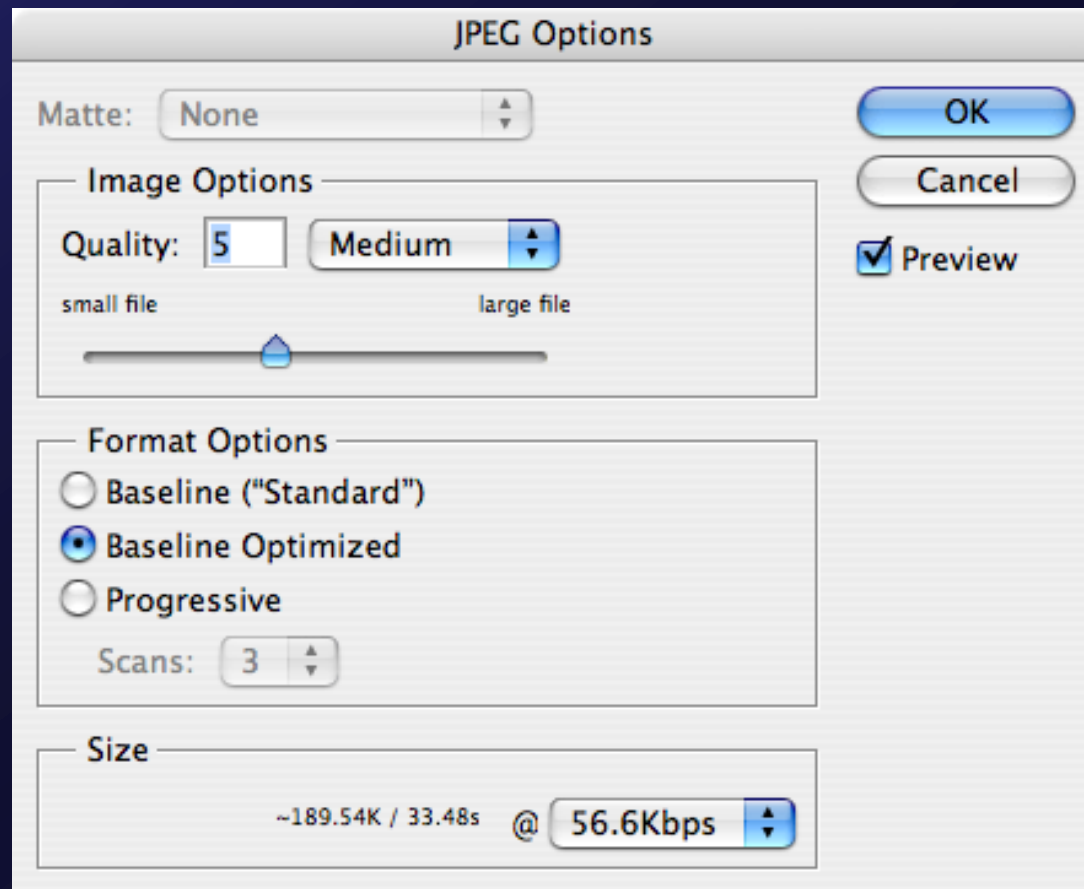
Исходный

9:1
“усреднение”



сжатый

В Photoshop, когда вы выбираете «Сохранить как ...» файл JPEG, вы можете выбрать уровень сжатия и, следовательно, размер и качество файла.



JPEG предупреждает:

Изображения с резкими краями, высокие контрасты, угловых областях и текста страдают от сжатия JPEG. Отсканированные "естественные" фотографии не теряют много, особенно в условиях высоких или максимальное качество.

Только сохранить готовые картинки, как в формате JPEG, каждый раз при открытии и сохранить снова, даже если вы не измените, вы теряете качества.

Всегда держите в оригинальную версию, не-JPEG (родной. PSD Format).

Так зачем использовать JPEG?

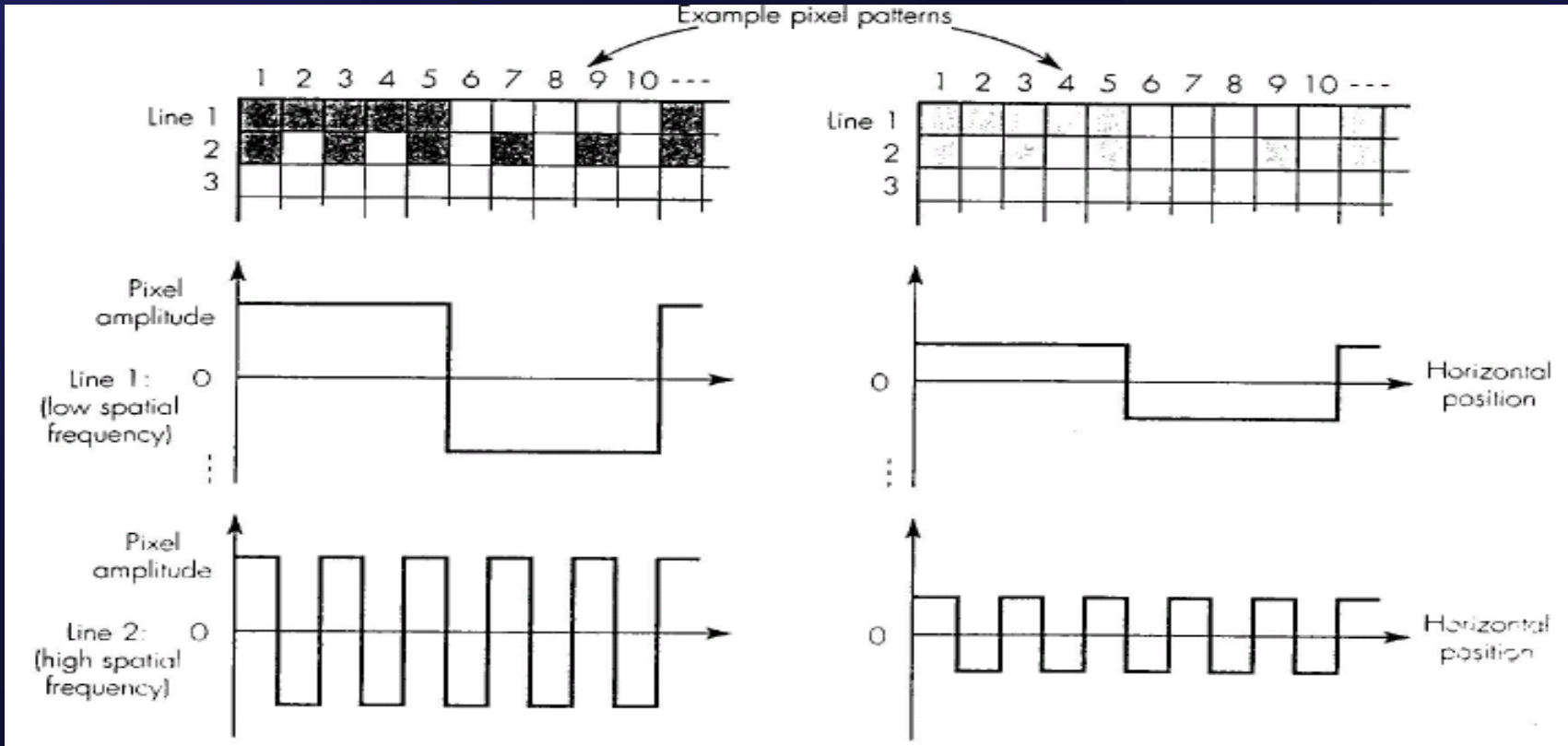
Это лучший формат для фотографических изображений в Интернете.

Это способность сжатия очень велика.

Сжатие изображений - DCT

Discrete Cosine Transform (DCT)

Концепция пространственной частоты :

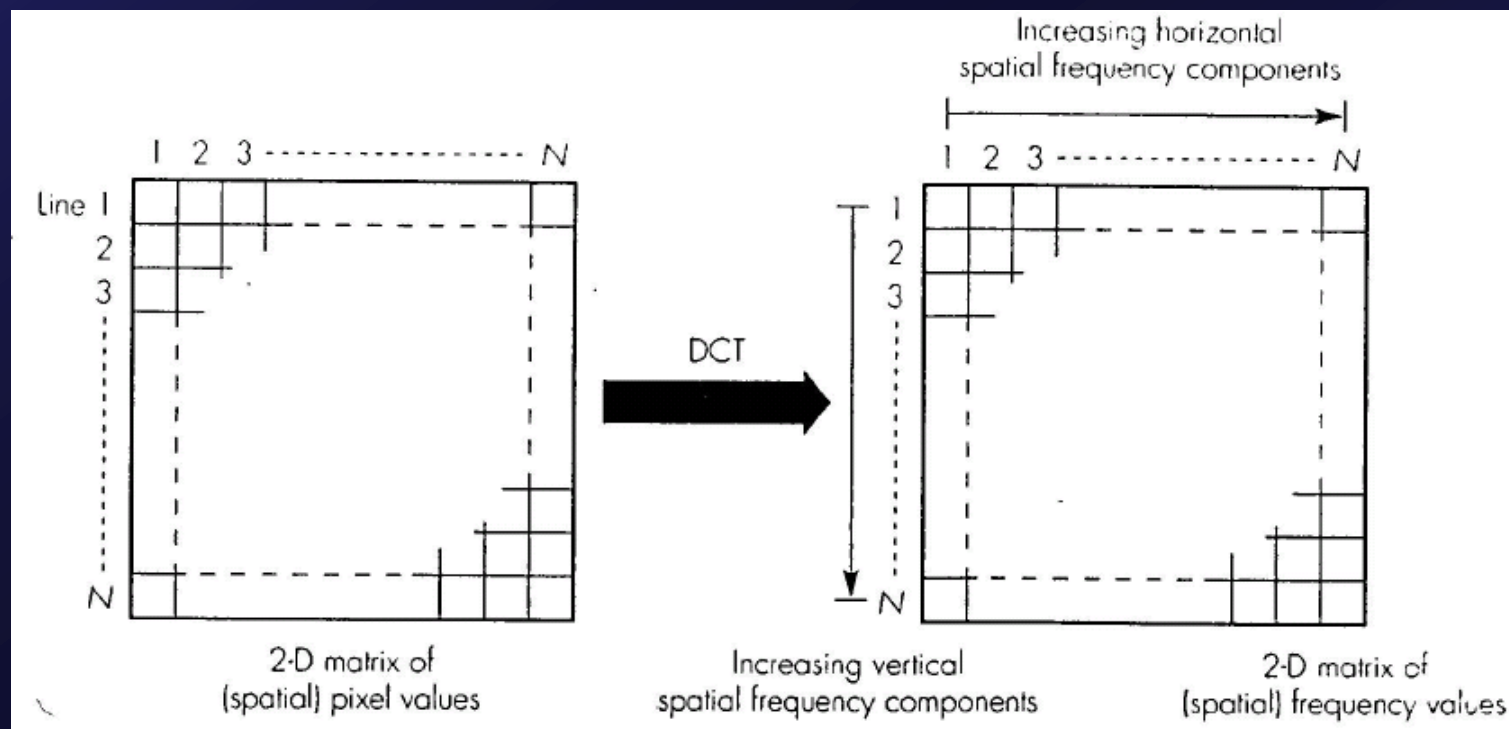


Сжатие изображений- DCT

- Как мы пересекаем вдоль строки (или столбца) пикселей, скорость изменения по величине называется **пространственная частота**.
- Пространственная частота является двумерным: в горизонтальном направлении и вертикальном направлении.

Концепция сжатия изображений- DCT

DCT преобразует 2-D пиксели в 2-D компоненты пространственной частотой :



Сжатие изображений - DCT

- Mathematically, for a 8×8 block of pixels, a pixel value $f(x, y)$ is transformed to a DCT coefficient $F(u, v)$ as follows*:

$$F(u, v) = \frac{1}{4} C_u C_v \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos\left(\frac{(2x+1)u\pi}{16}\right) \cos\left(\frac{(2y+1)v\pi}{16}\right)$$

where

$$C_i = \begin{cases} \frac{1}{\sqrt{2}} & i = 0 \\ 1 & \text{otherwise} \end{cases}$$

Сжатие изображений - DCT

Свойства пространственных частотных составляющих:

1. Соседние пиксели имеют высокую корреляцию (т.е., они имеют тенденцию быть похожими).

→ Как подать заявку это свойство для сжатия?

2. Глаза человека, менее чувствительны к более высокой частоты компонентов пространственных.

→ Для сжатия, мы можем упустить высокочастотные компоненты пространственного с малой амплитудой.

Сжатие изображений- JPEG

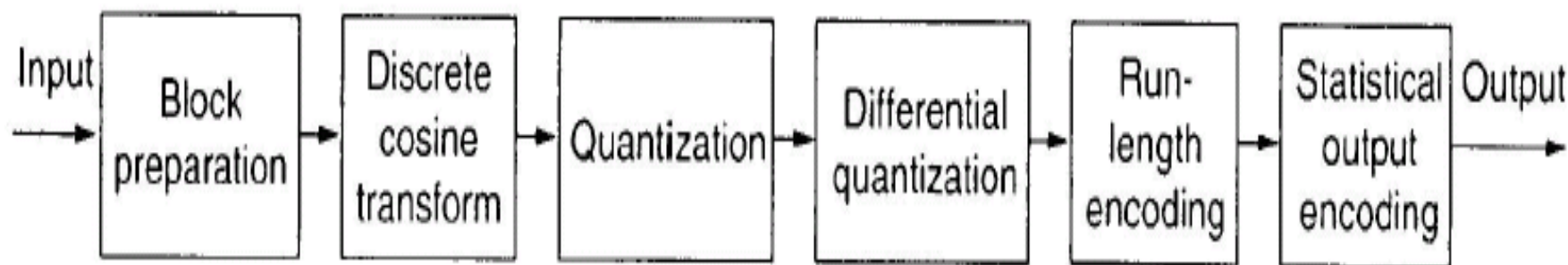
JPEG: Joint Photographic Experts Group.

JPEG является стандартом для сжатия изображений. Он широко используется, с типичными степенью сжатия 10 - 50.

JPEG имеет четыре режима и много вариантов. Рассмотрим потерями последовательный режим на 24-битных RGB изображений.

Сжатие изображений- JPEG

С Потерями последовательный режим
имеет шесть основных этапов:

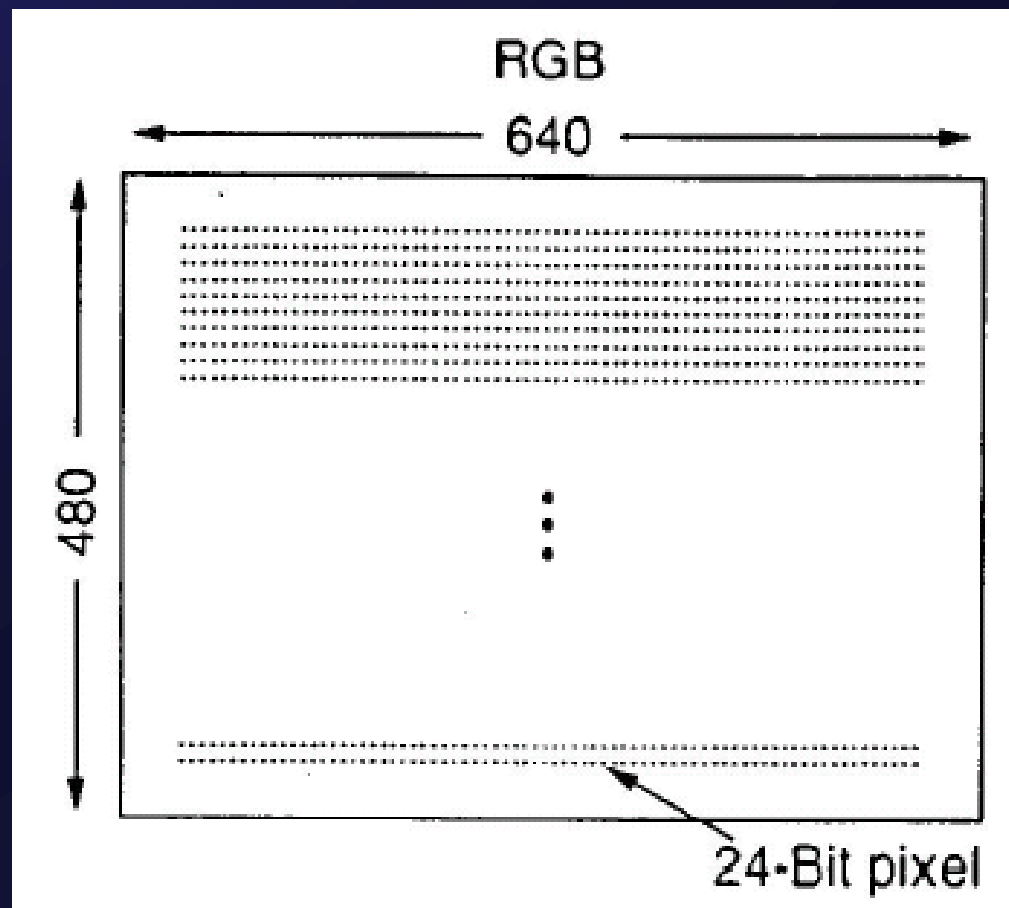


Декомпрессия: запустить шаги назад, требуя примерно в то же время выполнения как сжатия.

Сжатие изображения - JPEG

Шаг 1: Подготовка Блоков

Для удобства объяснения, рассмотрим
изображение RGB с 24 бит / пиксель:



Сжатие изображений - JPEG

0. Initially, we have R, G, B matrices of size 640×480 .
1. Transform RGB domain to YIQ domain to get Y , I , Q matrices.
2. For each square block of four pixels in the I and Q matrices, take the average of the four pixel values, so each matrix (I or Q matrix) is reduced to 320×240 . (Why?)
3. Subtract 128 from each element of all three matrices, so the middle of the range is 0.
4. Each of the three matrices is divided into 8×8 blocks.
→ There are 7200 blocks.

Сжатие изображений - JPEG

Шаг 2: дискретного косинусного преобразования

□ Применение дискретного косинусного преобразования (DCT) для каждого из 7200 блоков.

Шаг 3: Квантование

□ Разделите каждый коэффициент ДКП на весу, взятой из таблицы квантования, где веса резко возрастает от начала координат.

→ Протрите высокочастотные компоненты пространственного которые обычно менее важны.

Многие → квантованные коэффициенты в нуль, который может быть сжат эффективно.

Сжатие изображений- JPEG

- Example:

DCT Coefficients								Quantization table								Quantized coefficients							
150	80	40	14	4	2	1	0	1	1	2	4	8	16	32	64	150	80	20	4	1	0	0	0
92	75	36	10	6	1	0	0	1	1	2	4	8	16	32	64	92	75	18	3	1	0	0	0
52	38	26	8	7	4	0	0	2	2	2	4	8	16	32	64	26	19	13	2	1	0	0	0
12	8	6	4	2	1	0	0	4	4	4	4	8	16	32	64	3	2	2	1	0	0	0	0
4	3	2	0	0	0	0	0	8	8	8	8	8	16	32	64	1	0	0	0	0	0	0	0
2	2	1	1	0	0	0	0	16	16	16	16	16	16	32	64	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	32	32	32	32	32	32	32	64	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	64	64	64	64	64	64	64	64	0	0	0	0	0	0	0	0

19

- The quantization table is not part of the JPEG standard. Instead, each JPEG implementation must choose its own to control the quality-size tradeoff.

Сжатие изображения- JPEG

Шаг 4: Дифференциальная Квантование

- Наблюдение

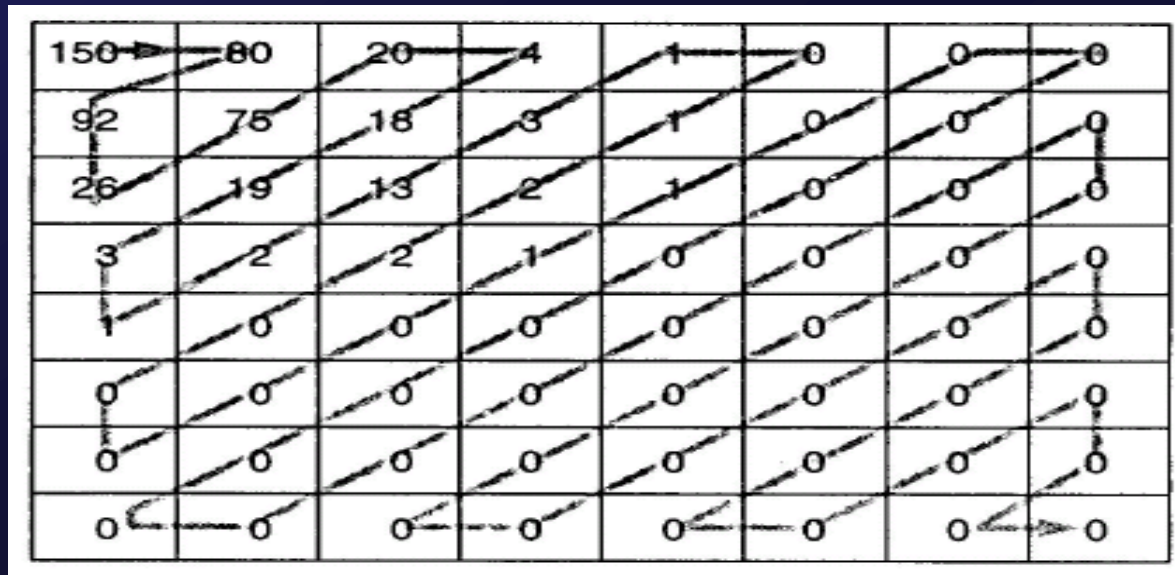
- Коэффициент $(0, 0)$ является средним блока, и это как правило, имеет наибольшее значение.
- Каждый блок охватывает лишь очень небольшую часть изображения, поэтому коэффициенты $(0, 0)$ в последовательных блоках есть небольшая разница.

- Дифференциальный квантования: заменить коэффициент $(0, 0)$ каждого блока на разность между его величиной и соответствующего значения в предыдущем кадре.

Сжатие изображений - JPEG

War 5: Run-Length Encoding

- Для каждого блока, следуйте зигзагом сканирования для извлечения последовательность 64 цифр, а затем применить кодирования длины сжимать эту последовательность.



Полученная последовательность имеет 38 последовательных нулей, которые могут быть эффективно сжаты продольным кодированием.

Сжатие изображений- JPEG

Шаг 6: Статистическая Кодировка вывода

**Применить кодирования
Хаффмана (Huffman) для сжатия
цифры, полученные в шаге 5.**

Сжатие изображений – JPEG2000

- ❑ JPEG2000 (или JPEG2k) является улучшенной версией JPEG, давая гораздо меньший размер изображения при том же качестве изображения.
- ❑ JPEG2000 использует вейвлет-преобразование вместо DCT.
- ❑ JPEG2000 имеет расширение файла JP2. В настоящее время поддерживается основными программного обеспечения, таких как Photoshop.