

ИЗОБРАЖЕНИЯ В КОМПЬЮТЕРЕ

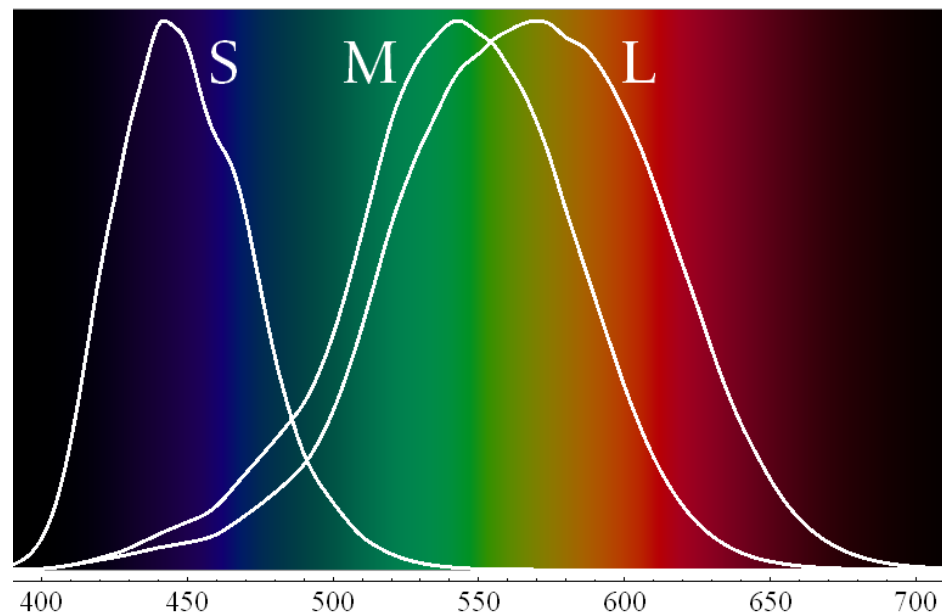
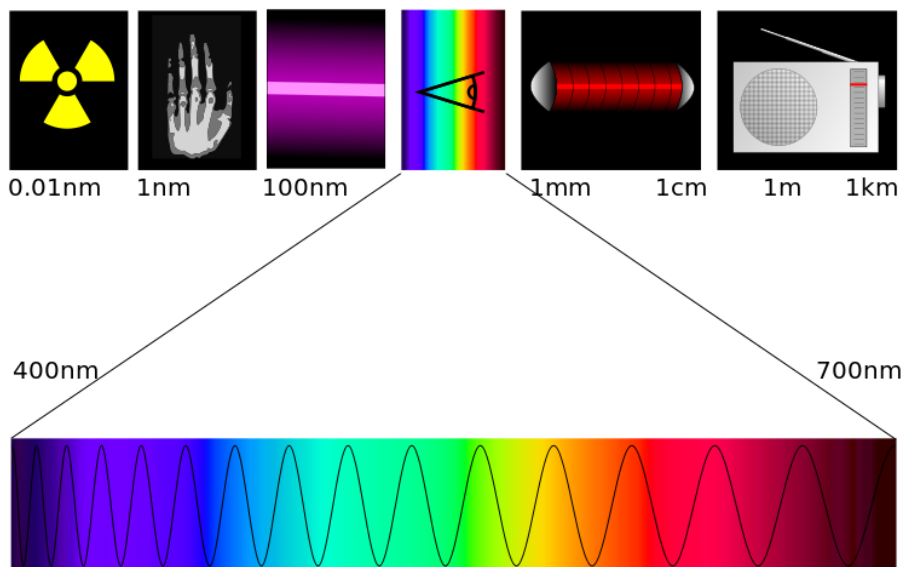
Представление цвета

- Чёрно-белое изображение
- Полутоновое изображение
(изображение в оттенках серого)
- Цветное изображение



Цвет и его восприятие человеком

- Видимый свет – электромагнитное излучение с длиной волны от 380 нм (790 ТГц) до 760—780 нм (385 ТГц).
- Человек воспринимает частоту монохроматического света как его цвет.
- Белый цвет – цвет излучения равной мощности во всём видимом диапазоне.
- Чёрный цвет – отсутствие излучения.



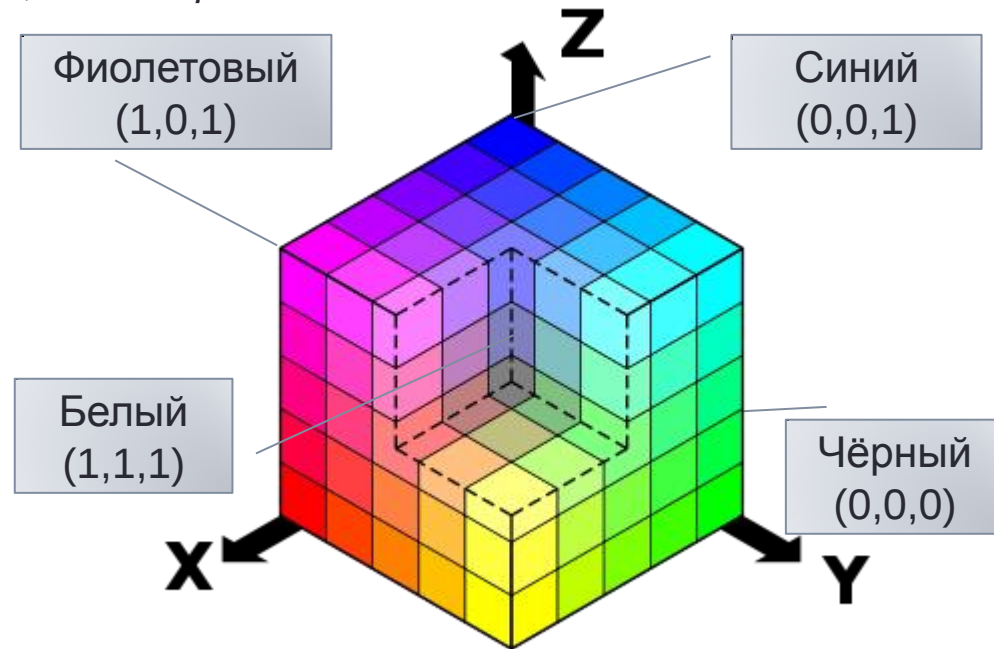
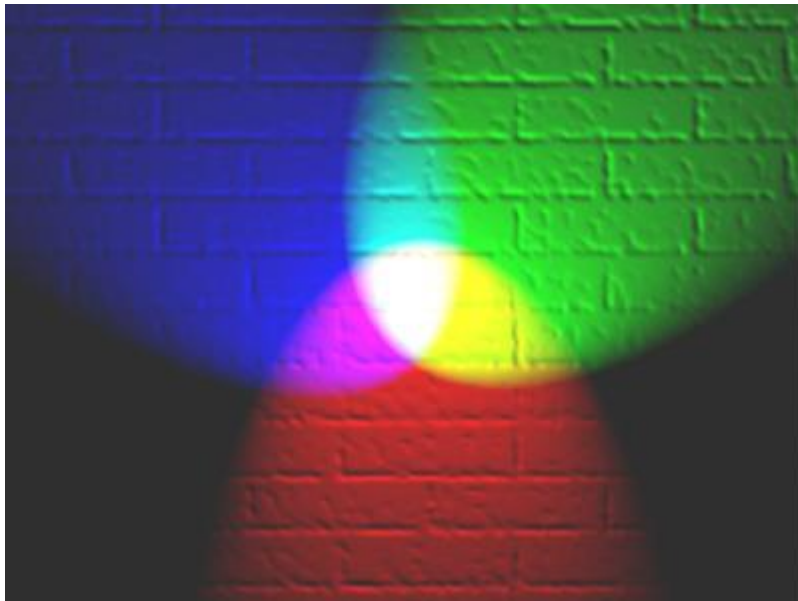
Аддитивное цветовое пространство

- В аддитивной цветовой модели цвета получаются добавлением к чёрному цвету.
- Основными цветами являются красный (Red), зелёный (Green) и синий (Blue) = RGB.

Выбор цветов обусловлен особенностями человеческого зрения.

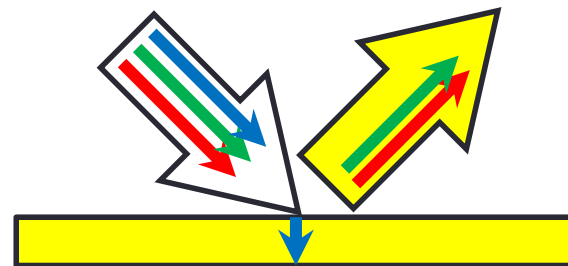
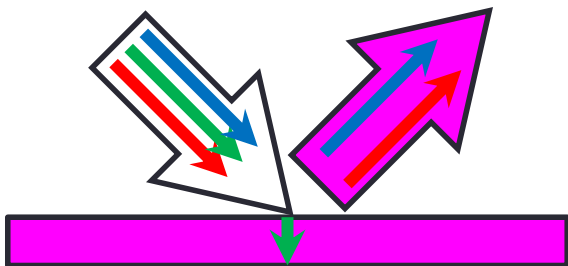
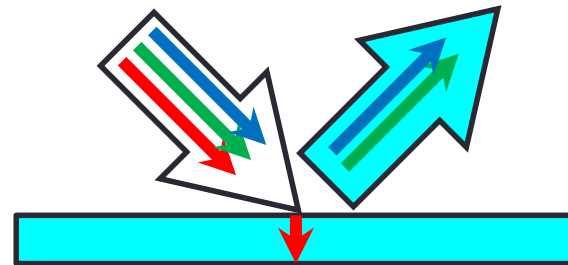
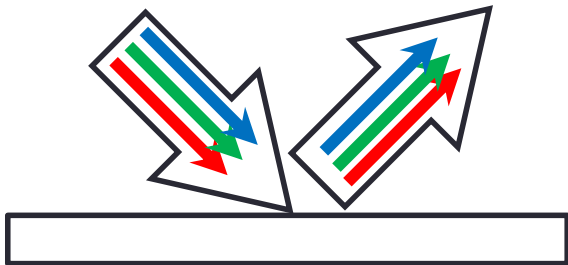
- Аддитивная модель «работает», когда мы смешиваем цвета от источников света.

Лампы со светофильтрами, светодиоды, мониторы.



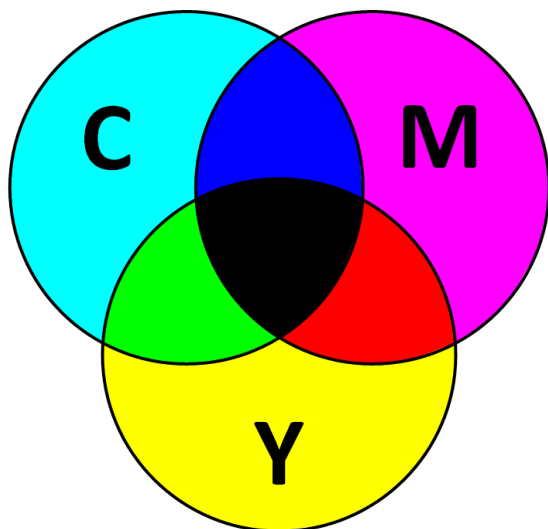
Субтрактивное цветовое пространство

- В субтрактивном цветовом пространстве цвета получаются вычитанием из белого цвета.
- Субтрактивная модель «работает» в условиях отражения.
Цвет поверхности, освещенной белым светом, определяется тем, какие длины волн она поглотила.



Цветовое пространство CMY

- В субтрактивном цветовом пространстве цвета получаются вычитанием из белого цвета.
- Субтрактивная модель «работает» в условиях отражения.
Цвет поверхности, освещенной белым светом, определяется тем, какие длины волн она поглотила.
- Основными цветами являются голубой (Cyan), фиолетовый (Magenta) и желтый (Yellow) = CMY.



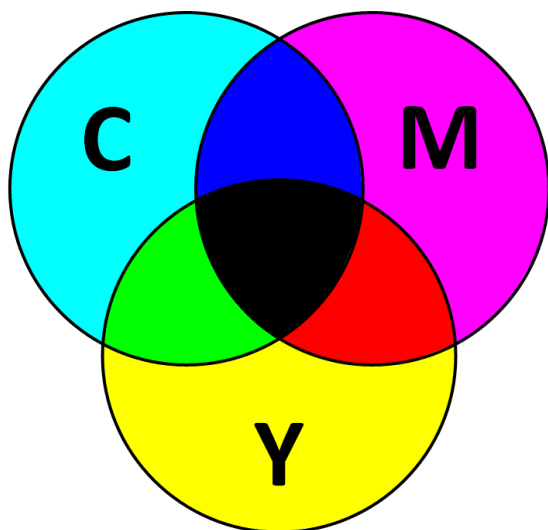
В цветовой модели CMY белому цвету соответствуют координаты $(0,0,0)$ – полное отсутствие краски.

*Другие цвета:
красный – $(0,1,1)$, синий – $(1,1,0)$, чёрный – $(1,1,1)$.*

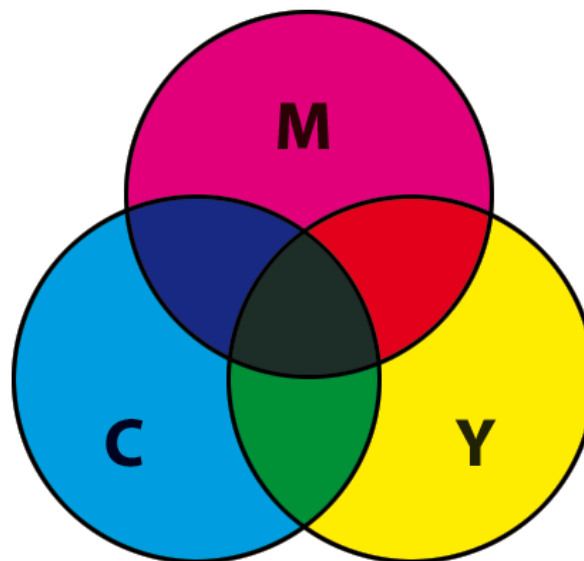
Цветовое пространство СМУК

- При реальной печати смешение трёх основных цветов даст вместо чёрного грязно-коричневый.
- Для улучшения качества печати используют четвертый цвет – чёрный (К) = СМУК.

Теория



Практика

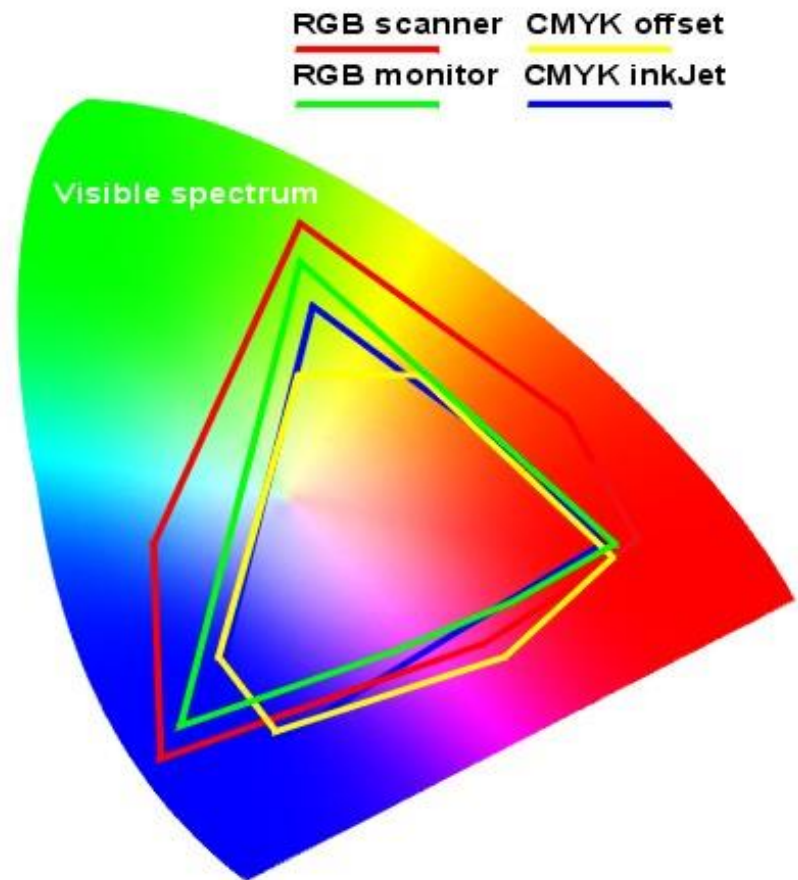
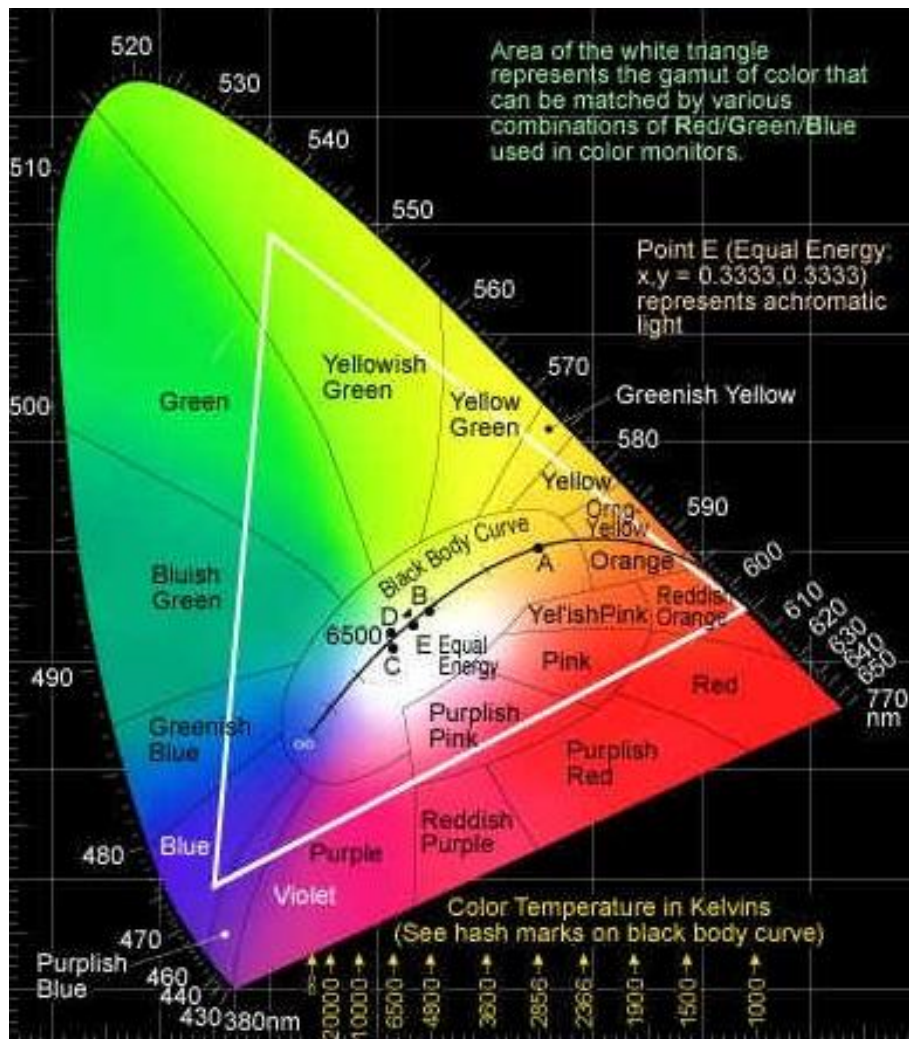


Цветоделение

- Перевод изображения из модели RGB в модель CMYK называется цветоделением. Это сложный процесс, который можно реализовать разными способами.



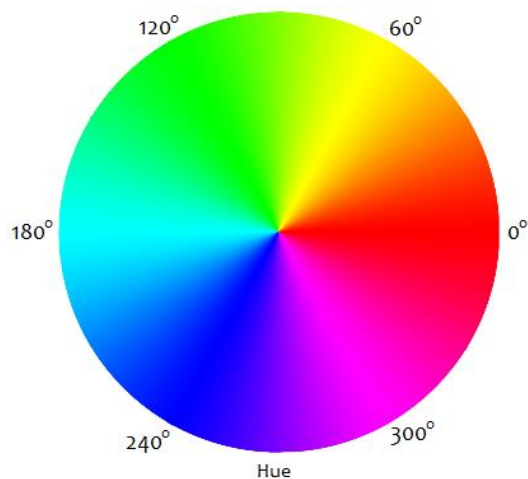
Цветовое пространство CIE



Другие цветовые пространства

HSL

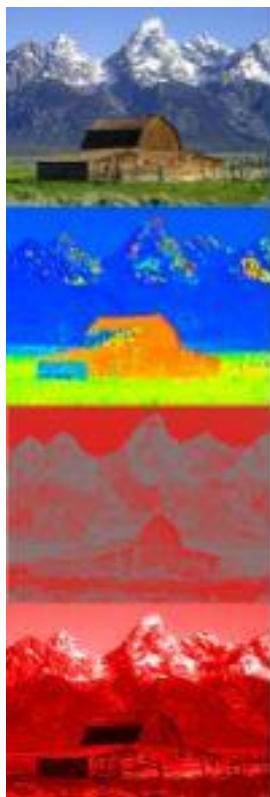
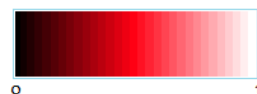
Hue, Saturation, Lightness
Тон, насыщенность, светлота



Color Saturation



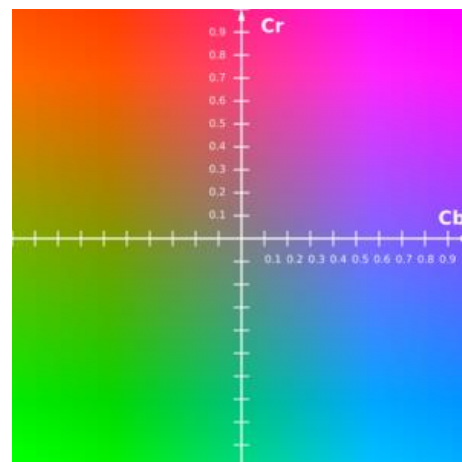
Lightness



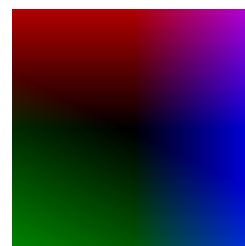
YCbCr

Яркость и цветоразностные
компоненты

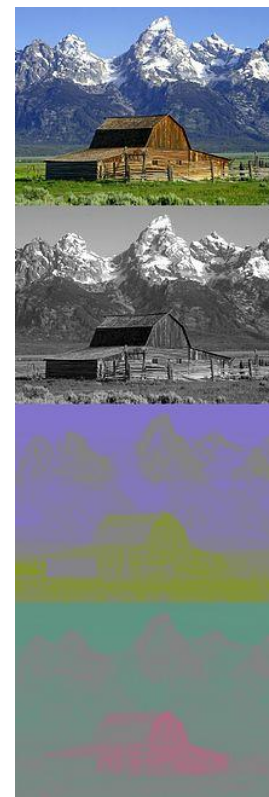
$Y=0.5$



$Y=0$



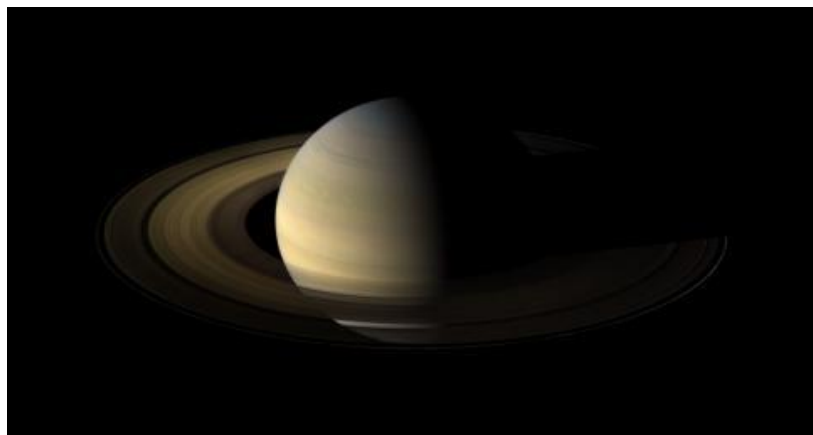
$Y=1$



Представление цветов в памяти компьютера

- В цветовых моделях, как правило, для описания компонент используются непрерывные значения из отрезка $[0,1]$.
- В памяти компьютера эти значения представляются целыми числами, размер которых зависит от числа выделенных бит.
- Сейчас для представления цвета в моделях RGB и CMY чаще всего используется вариант, когда каждая компонента использует 1 байт.
 - Каждая компонента изменяется от 0 (нет цвета) до 255 (максимальная интенсивность).
 - В RGB один пиксел занимает 3 байта = 24 бита.
 - Общее число доступных цветов RGB: $2^{(3*8)} \approx 16$ млн.
- Для достижения максимального качества при обработке изображений используют 2 или 4 байта на каждую компоненту, числа с плавающей точкой.

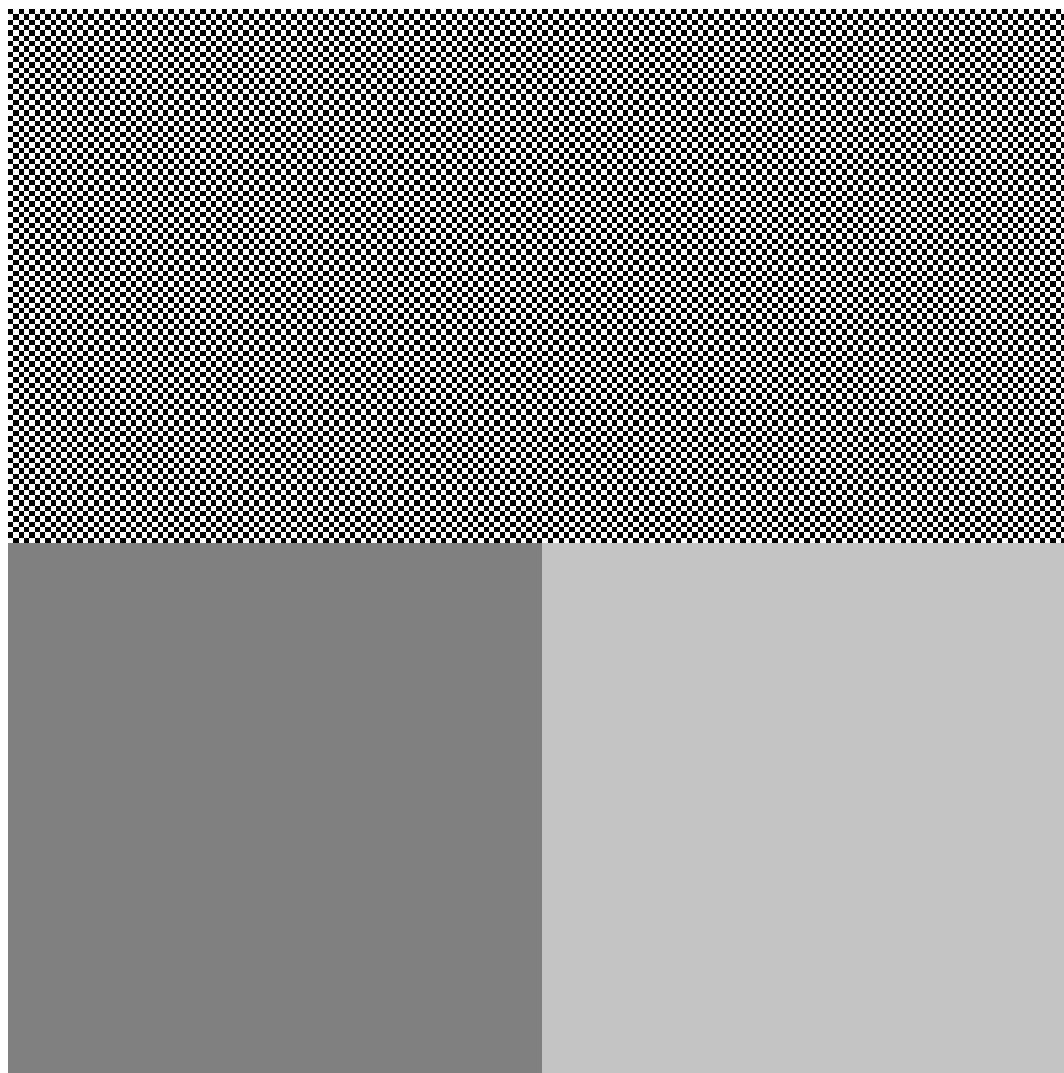
Искажение изображений при обработке



Уменьшение

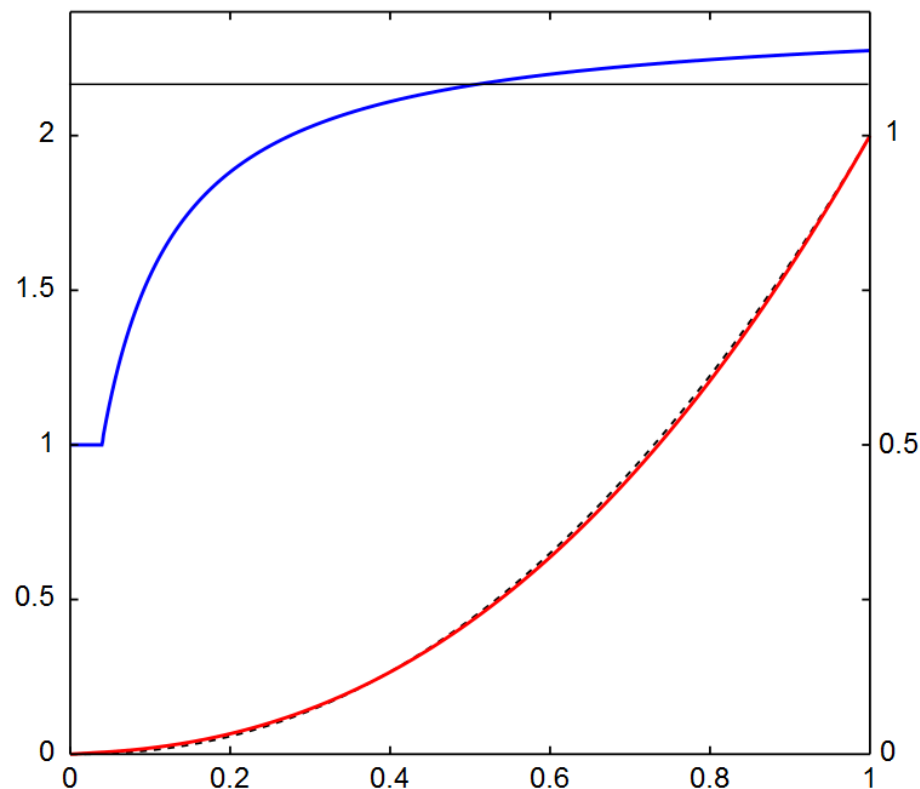


Искажение изображений при обработке



Степенная шкала яркостей

Код	Линейная яркость	Степенная яркость
0	0.0000	0.0000
32	0.1255	0.0144
64	0.2510	0.0513
96	0.3765	0.1170
128	0.5020	0.2159
160	0.6275	0.3515
186	0.7294	0.4969
192	0.7529	0.5271
224	0.8784	0.7454
225	1.0000	1.0000



Цветовое пространство sRGB

- Большинство форматов хранит RGB цвета в цветовом пространстве sRGB со степенной зависимостью яркости.
- Для качественной обработки цветов следует:
 1. Перевести цвет из степенного пространства sRGB в линейное цветовое пространство:

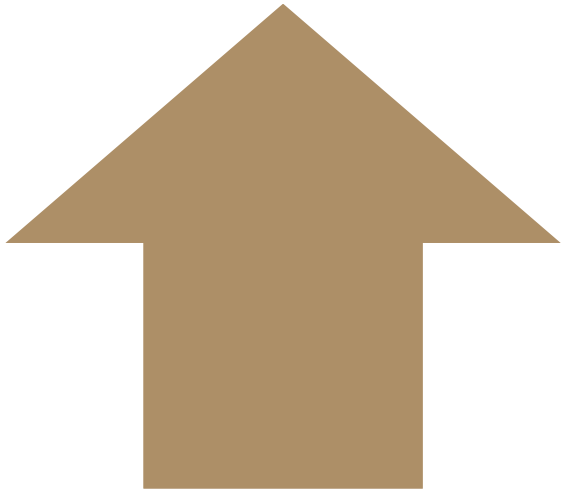
$$C_{linear} = \begin{cases} C_{sRGB}/12.92, & C_{sRGB} < 0.04045, \\ \left(\frac{C_{sRGB} + \alpha}{1 + \alpha} \right)^{2.4}, & C_{sRGB} \geq 0.04045, \end{cases}$$

где $\alpha = 0.055$, C_{sRGB} - компонента R, G или B в диапазоне [0,1].

2. Провести вычисления в линейном пространстве.
3. Перевести цвет обратно в пространство sRGB:

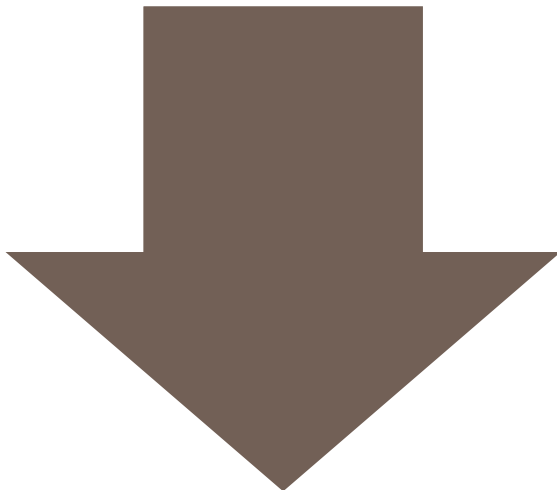
$$C_{sRGB} = \begin{cases} 12.92C_{linear}, & C_{linear} < 0.0031308, \\ (1 + \alpha)C_{linear}^{1/2.4} - \alpha, & C_{linear} \geq 0.0031308. \end{cases}$$

Способы представления изображений в компьютере



Векторные изображения

- Изображение представляется как набор геометрических примитивов (линии, эллипсы, прямоугольники, и т.п.).



Растровые изображения

- Изображение представляется как двумерный массив, в котором записаны цвета точек (пикселей).

Особенности векторных изображений

- Векторное изображение представляется как набор геометрических примитивов.
 - В роли примитивов могут выступать линии, прямоугольники, эллипсы, дуги, сплайновые кривые, текст и более сложные фигуры.
 - Расположение примитива задаётся координатами опорных точек.
 - Форма примитива может определяться дополнительными параметрами («степень скруглённости» скруглённого прямоугольника и т.п.).
 - Как правило, имеется возможность задать цвет внутренней области и границы фигуры; возможна поддержка градиентной заливки, текстуры и т.п.
- Степень детализации векторного изображения ограничивается числом использованных в нём графических примитивов.
- Векторное изображение может быть отмасштабировано при выводе в очень широких пределах без существенной потери качества.

Как можно представить себе растровое изображение

Создание растрового изображения:

- представьте, что Вы смотрите на пейзаж через окно;
- нарисуйте на стекле тонкими вертикальными и горизонтальными линиями сетку с квадратными ячейками;
- запишите цвет, который наиболее соответствует цвету каждой ячейки.

Каждая ячейка называется пикселом.

Вывод растрового изображения:

- возьмите лист бумаги;
- воспроизведите на нём сетку с тем же числом ячеек, которое использовалось при создании изображения;
- закрасьте каждую ячейку тем цветом, который был для неё записан.

Уровень детализации растрового изображения зависит от числа пикселей, и не может быть увеличен после создания.

- Это определяется теоремой Котельникова-Найквиста-Шеннона.

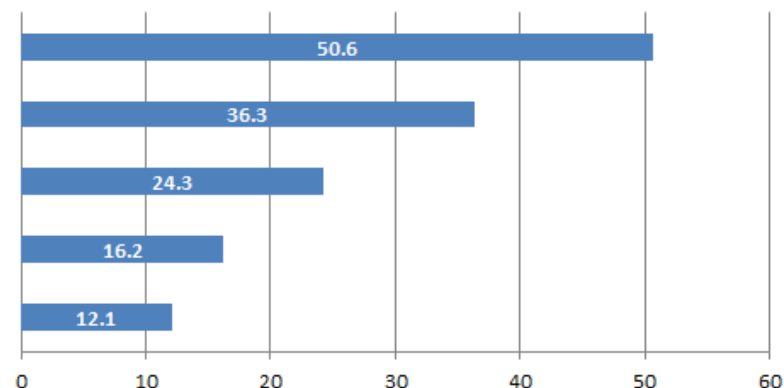
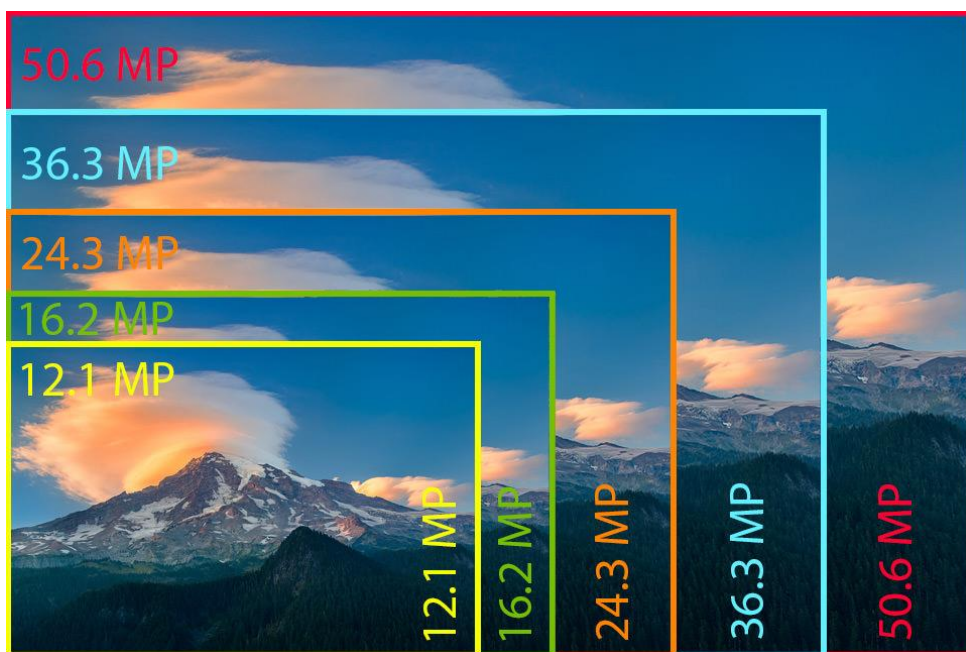
Размер растрового изображения

Когда растровое изображение хранится и обрабатывается в компьютере, оно не имеет размера в физических величинах длины (метрах).

- Есть только число ячеек сетки – по горизонтали и по вертикали.
- Размер ячейки в метрах (сантиметрах, миллиметрах) важен только в момент оцифровки или вывода растрового изображения.
- В памяти компьютера размер (ширина и высота) растрового изображения измеряется в пикселах.

«Мегапикселы»

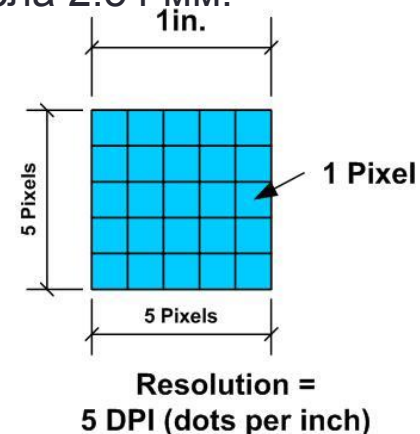
Производители цифровых фотоаппаратов часто измеряют размер кадра указывая общее число пикселей, не забывайте что этот параметр имеет смысл площади



Можно ли глядя на иллюстрацию сказать, что 24.3 MPix в два раза больше, чем 12.1?..

Разрешение растрового изображения

- Разрешение определяет размер пиксела в момент создания или вывода растрового изображения.
- Разрешение чаще всего измеряют в «точках на дюйм» = Dots per Inch = DPI.
 - Иногда используют обозначение Pixels per Inch = PPI.
- Информация о разрешении изображения чаще всего сохраняется в файле, но она может быть изменена в любой момент никак не влияя на изображение.
 - Некоторые программы игнорируют эту информацию.
- Пусть есть изображение размером 1000x1000 точек.
- При выводе с разрешением 300 dpi получим размер стороны картинки $1000/300 = 3$ дюйма ≈ 8.5 см. Сторона пиксела 0.85 мм.
- При выводе с разрешением 10 dpi получим размер стороны картинки $1000/10 = 100$ дюймов ≈ 2.54 м. Сторона пиксела 2.54 мм.

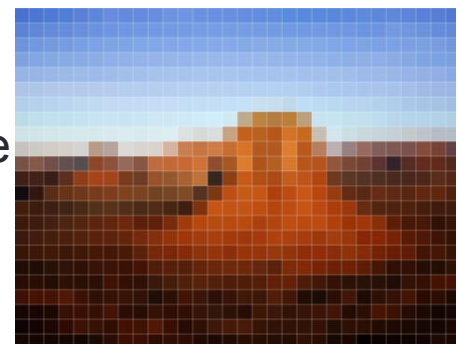
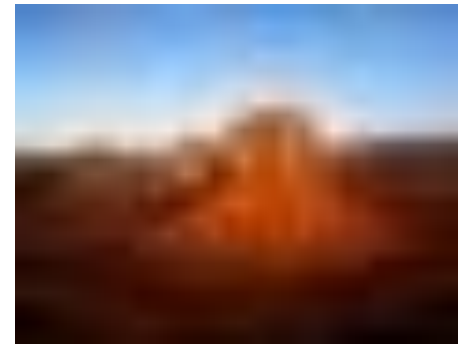


Изменение размера растрового изображения

- Изменение размера изображения может потребоваться при выводе изображения в условиях, когда размер картинки, разрешение устройства вывода и желаемый физический размер не соответствуют друг другу.

Размер в пикселах = Разрешение * Размер в "метрах"

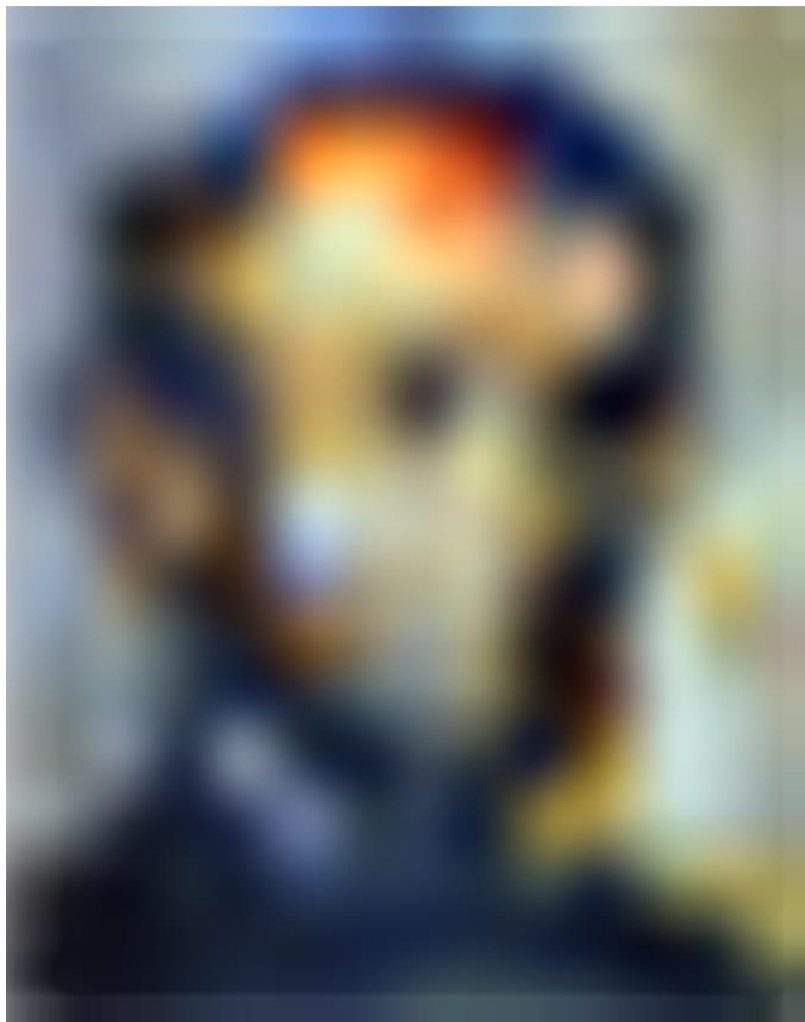
- Размер в пикселах меньше, чем нужно.
 - При выводе произойдёт ухудшение качества. Для того, чтобы «пикселы были не сильно квадратными» следует использовать интерполяцию, но чуда она сделать не может.
- Размер в пикселах больше, чем нужно.
 - При использовании интерполяции качество изображения улучшится – уменьшатся шумы. Но «глупые»/быстрые алгоритмы всё равно могут испортить качество.
- В любом случае, результат будет лучше, если изменение размера осуществляется в целое число раз.
 - Из 2x2 пикселей исходного изображения делаем 1 пиксел уменьшенного, или наоборот.



Повышение разрешения изображения — что нельзя сделать



Повышение разрешения изображения — что нельзя сделать



Повышение разрешения изображения — что нельзя сделать



Повышение разрешения изображения — что можно сделать

Повышение разрешения в 4 раза с использованием программы Neural Enhance на основе нейронных сетей.

До обработки



После обработки

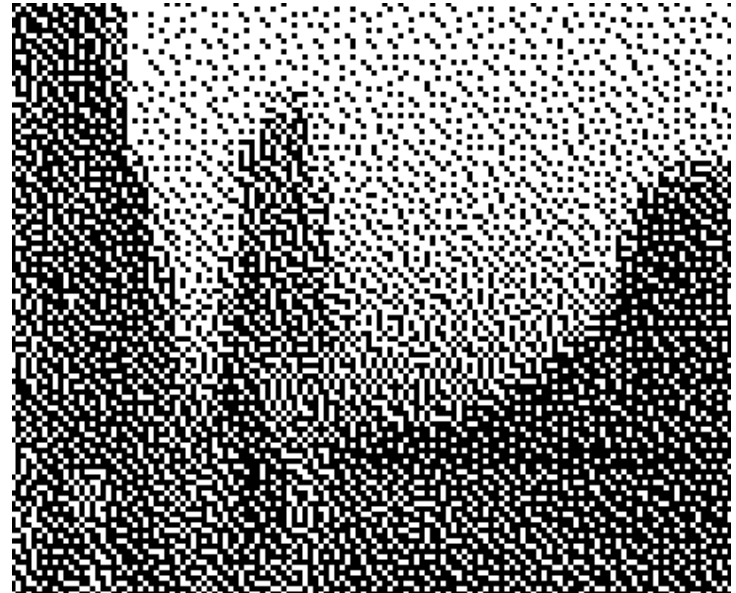
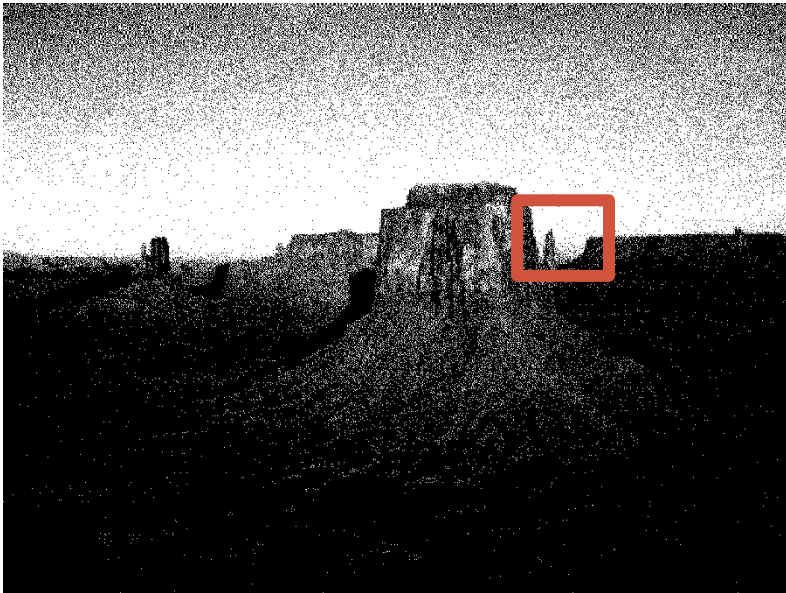


Связь между разрешением и числом ЦВЕТОВ

- Струйные и лазерные принтеры имеют в своём распоряжении очень ограниченное число цветов.
- Используя несколько соседних пикселов можно увеличить эффективное число цветов за счёт уменьшения разрешения.

Пусть у нас есть 2 цвета. Вместо 1 пиксела возьмём квадратик 2x2 пиксела. Тогда мы можем получить 4 градации серого.

- Принтеры также могут регулировать размер точки, в струйных принтерах добавляют дополнительные цвета (светлые).

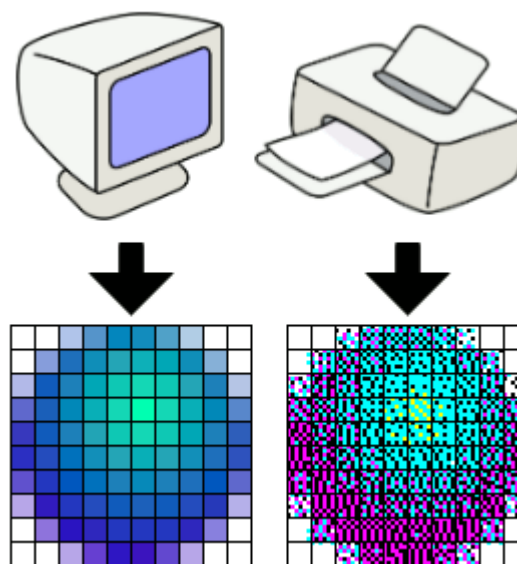
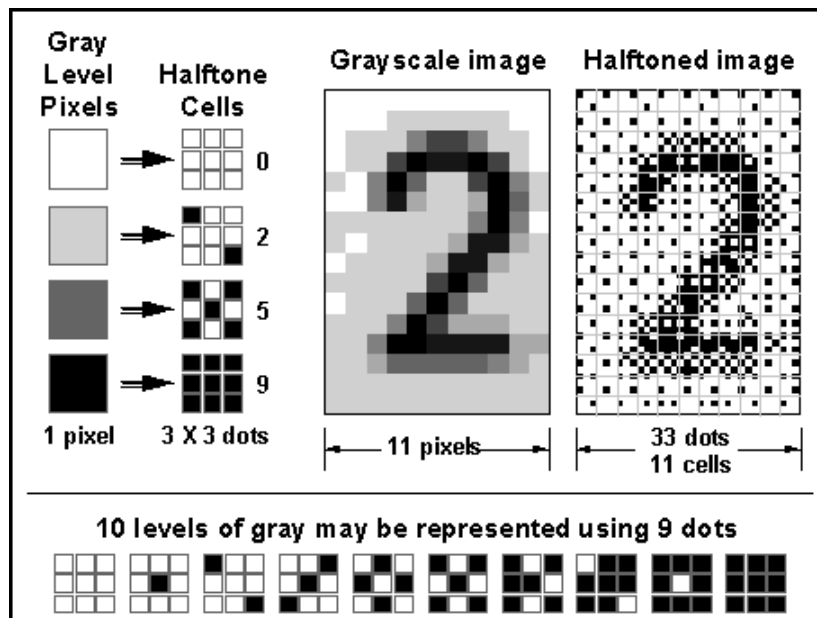


Связь между разрешением и числом ЦВЕТОВ

- Струйные и лазерные принтеры имеют в своём распоряжении очень ограниченное число цветов.
- Используя несколько соседних пикселов можно увеличить эффективное число цветов за счёт уменьшения разрешения.

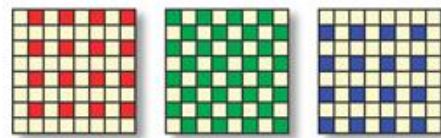
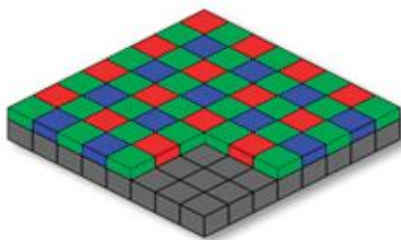
Пусть у нас есть 2 цвета. Вместо 1 пиксела возьмём квадратик 2x2 пиксела. Тогда мы можем получить 4 градации серого.

- Принтеры также могут регулировать размер точки, в струйных принтерах добавляют дополнительные цвета (светлые).

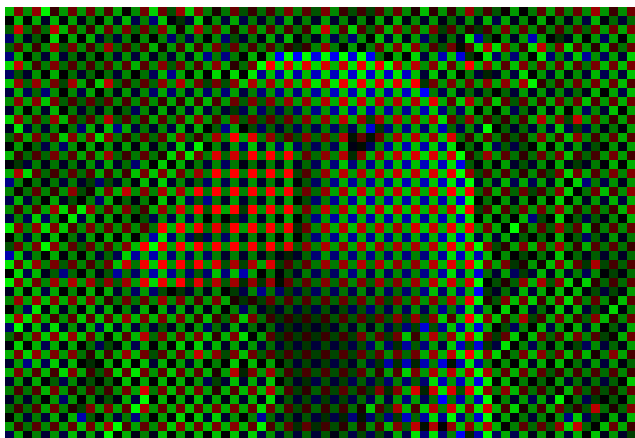


Сенсоры цифровых камер

- Сенсоры цифровых камер чувствительны ко всему диапазону видимого цвета (и даже больше).
- Для создания цветных сенсоров используют светофильтры.



Данные с фильтра Байера



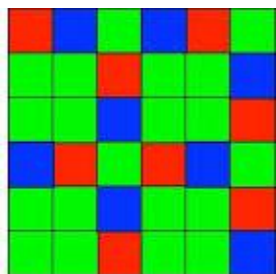
Интерполяция
в 3 раза.

Восстановленное изображение

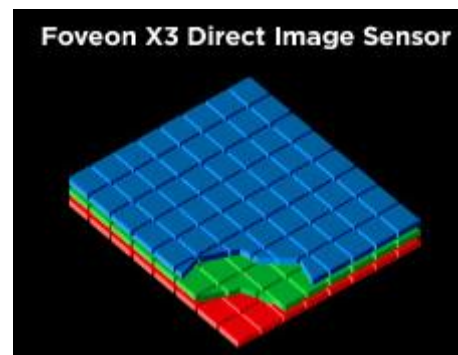
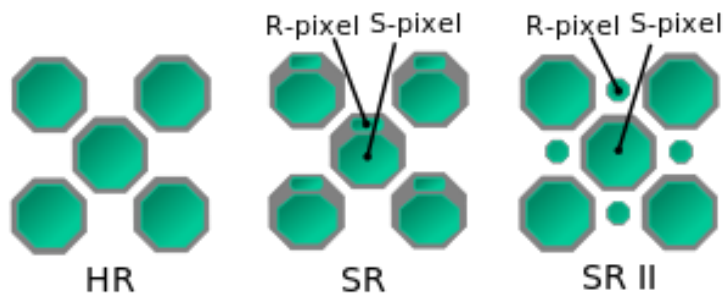
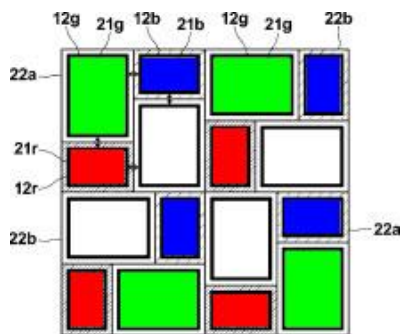


Сенсоры цифровых камер

- Чем больше площадь пиксела, тем больше света на него попадает за одно время – большие пикселы обеспечивают большую чувствительность.
- Если света слишком много – произойдет насыщение АЦП сенсора, данные в засвеченной области восстановить нельзя.
- Если света мало – уровень сигнала будет близок к уровню шума, но что-то разглядеть будет можно.

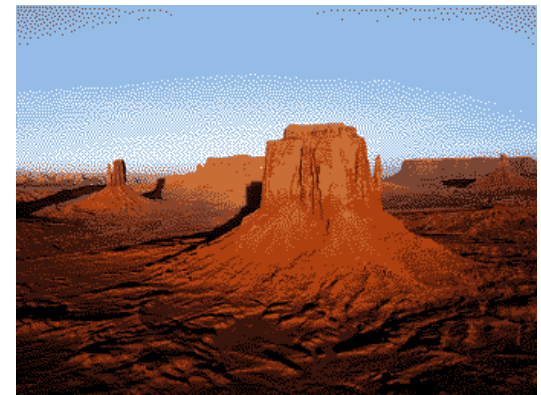
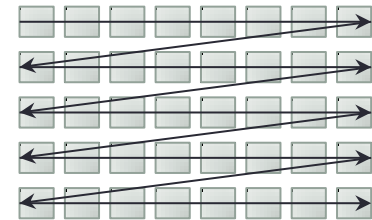


FUJI X-TRANS



Представление растровых изображений в памяти компьютера

- Чёрно-белое изображение
Один бит на пиксель: 0 – чёрный, 1 – белый.
- Полутонное изображение (изображение в оттенках серого)
Один байт на пиксель: 0 – черный,... , 255 белый.
- Цветное изображение
 - Палитра
Используется фиксированное число цветов, как правило не более 255.
Сначала записывается палитра: значения BGR для используемых цветов;
Затем, для каждого пиксела записывается индекс цвета в палитре (1 байт или меньше).
 - Полноцветное изображение
Записывается матрица из значений цвета для каждого пиксела BGR, BGR, BGR,



Форматы хранения растровых изображений

- Универсальные
 - TIFF
 - Чаще всего используется для хранения изображений без сжатия. Возможно очень много вариантов внутреннего устройства. При этом всегда работает.
- Без сжатия (простейшее сжатие без потерь)
 - BMP
 - Стандарт из операционной системы Windows. Почти всегда обрабатывается правильно.
 - GIF
 - Изображения с палитрой. Поддерживает анимацию.
 - PNG
 - Открытый стандарт для изображений без сжатия.
- Сжатие с потерями
 - JPG
- RAW
 - «Сырые» данные с сенсора камеры, без дополнительной обработки. Используется для получения максимального качества при последующей обработке.

JPEG

- Наиболее популярным форматом сжатия изображений с потерями является JPEG.
- Основные идеи:
 - Яркостная компонента изображения сохраняется целиком, размер цветовых компонент сокращается в 2 раза.
 - Изображение обрабатывается блокам 8x8 пикселей.
 - Удаляются мелкие детали.
 - Информация дополнительно сжимается сжатием без потерь.
- Особенности:
 - Многократное сжатие и декомпрессия приводят к потере качества – лучше не редактировать изображение в JPEG.
 - Степень сжатия определяется большим числом параметров – «качество 80%» это очень условно.



Используйте графические форматы правильно

		Сжатие без потерь PNG	Сжатие с потерями JPG
432x238	• Графика • Скриншоты с текстом • Мало цветов • Большие области одного цвета • Тонкие линии	Размер файла 11 кБ Качество Без искажений 	Размер файла 23 кБ Качество Ужасно
379x277	• Фотографии • Много цветов • Плавные переходы цвета	Размер файла 136 кБ Качество Без искажений ✓ Качественная обработка. ✓ Хранение ценных фото. ✗ Выложить в интернет ✗ Хранить много разного	Размер файла 19 кБ Качество Хорошее ✗ Качественная обработка. ✗ Хранение ценных фото. ✓ Выложить в интернет ✓ Хранить много разного

Преобразование между растровыми и векторными изображениями

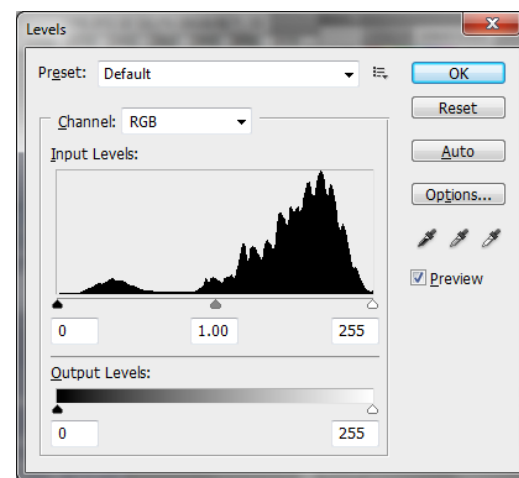
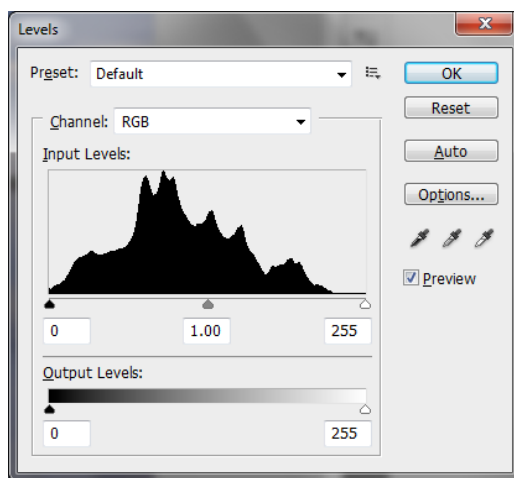
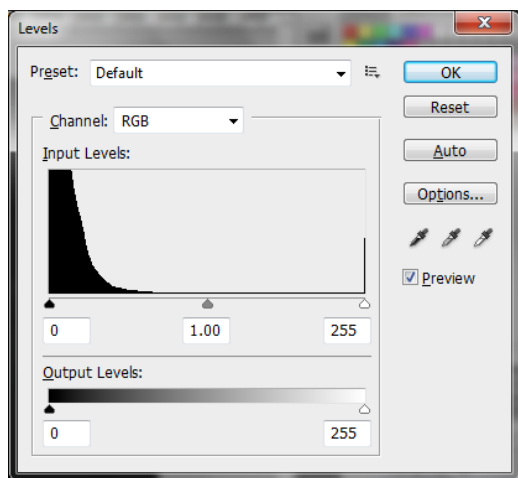
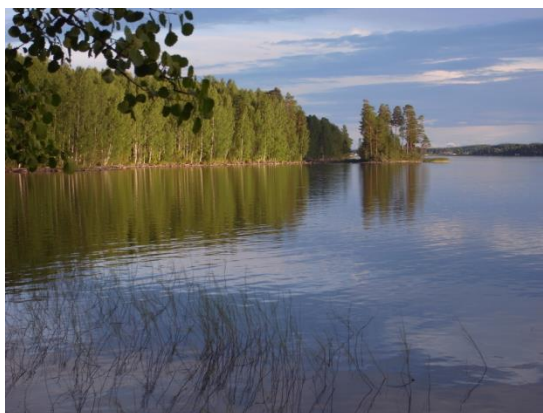


ОБРАБОТКА ИЗОБРАЖЕНИЙ

Анализ гистограммы

Гистограмма изображения

Гистограмма – график статистического распределения элементов изображения, в котором по оси абсцисс представлена яркость, а по оси ординат – число пикселов с таким значением яркости.



Как построить гистограмму

- Создать массив из 255 элементов, присвоить начальное значение 0.
- Для каждого пиксела изображения:
 - Вычислить яркость этого пиксела:

Общий уровень	Общий уровень с учётом sRGB	Для конкретной компоненты (R, G или B)	Для конкретной компоненты в sRGB
$Y = 0.3R + 0.59G + 0.11B$	Перевести все компоненты в линейное пространство. $Y_{linear} = 0.21R_{linear} + 0.72G_{linear} + 0.07B_{linear}$	Использовать значение компоненты.	Перевести значение компоненты в линейное пространство.

- Увеличить на 1 значение ячейки массива, соответствующее яркости рассматриваемого пиксела.
- Поделить каждый элемент массива на число пикселов в изображении.

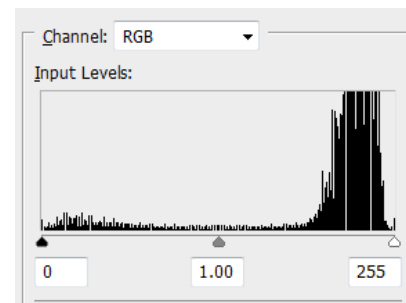
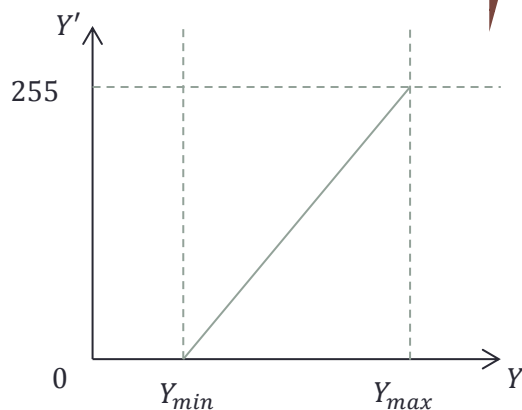
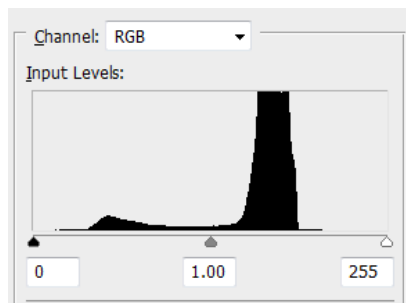
Линейная коррекция контраста

Коррекция – к изображению применяется преобразование яркостей, компенсирующее нежелательный эффект: $f^{-1}(Y) = Y'$, где Y – яркость пиксела на исходном изображении, Y' – яркость пиксела после коррекции.

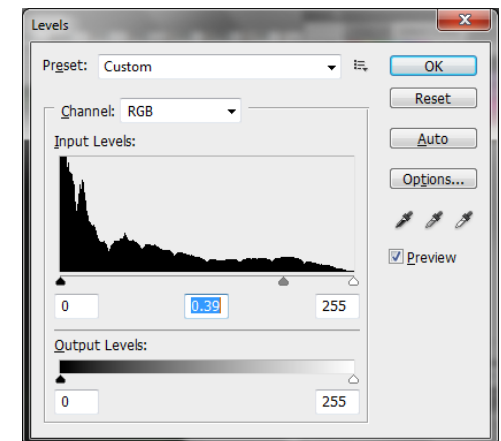
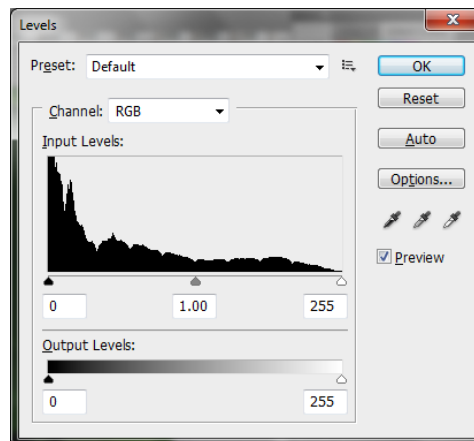
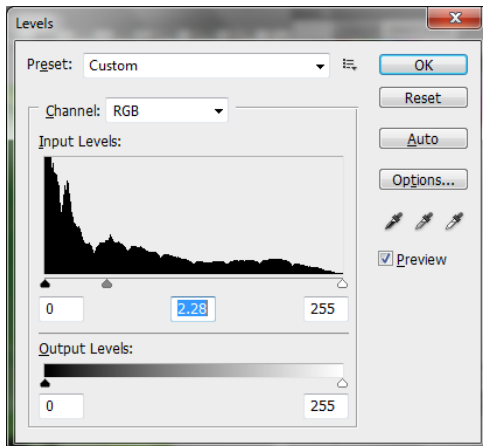
Узкий диапазон яркостей можно компенсировать линейным растяжением.



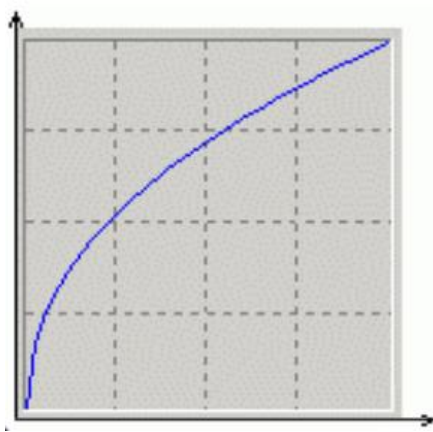
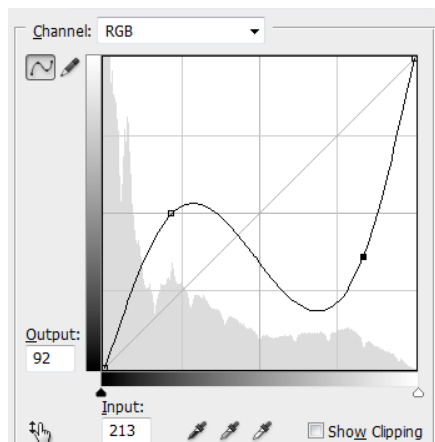
$$f^{-1}(Y) = (Y - Y_{min}) * \frac{255 - 0}{Y_{max} - Y_{min}}$$



Тоновая коррекция изображения

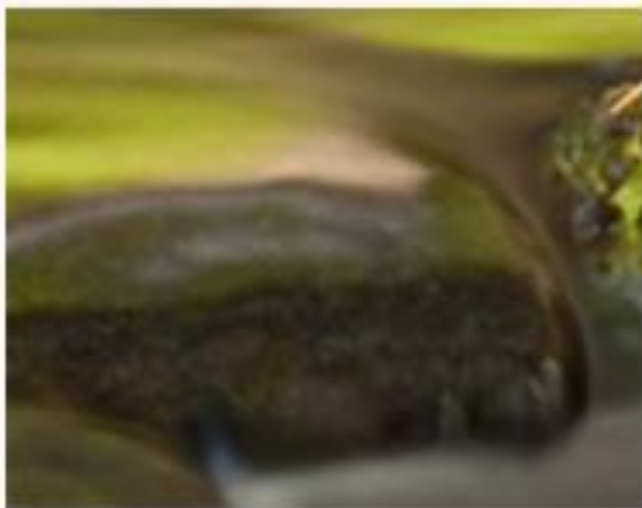


Нелинейная коррекция изображения



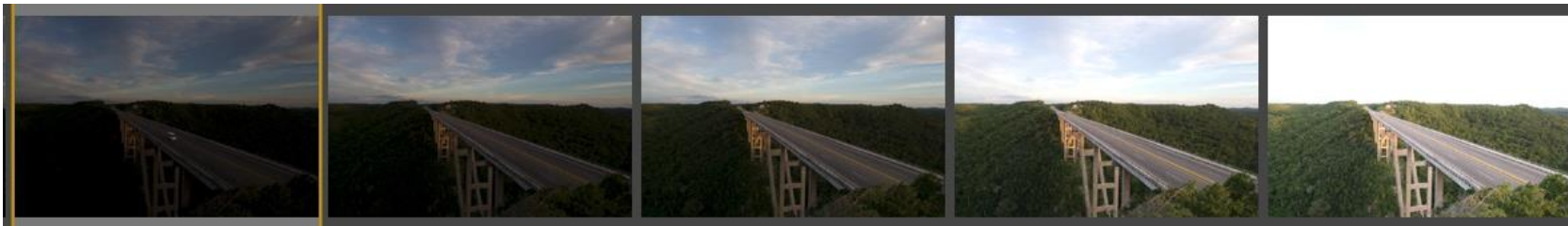
Объединение нескольких изображений

Уменьшение шумов



Фотография Clive Dexter

Увеличение динамического диапазона



Фотография
Harold Davis

Увеличение глубины резкости

Одно изображение



Объединение 8 изображений



Фотография Erez Marom

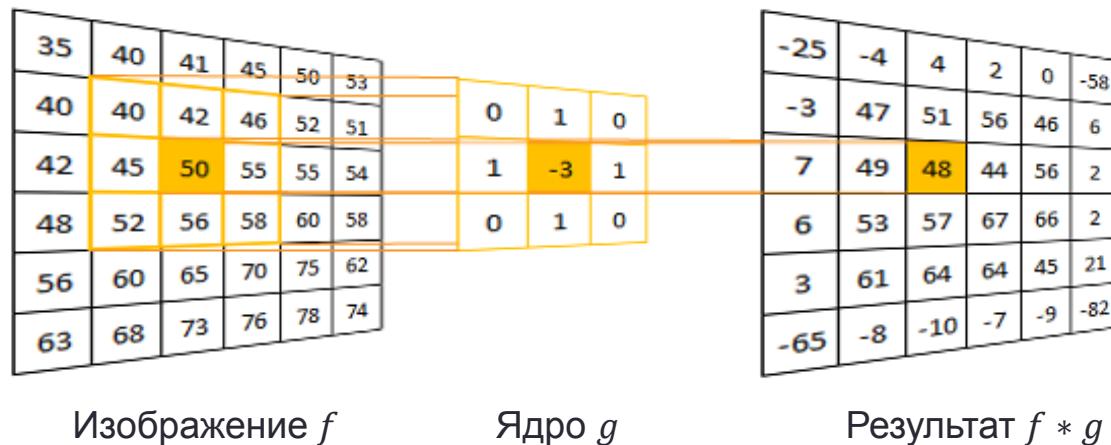
Фильтрация изображений с помощью свёртки.

Свёртка изображения с ядром

- Пусть ядро g – матрица размером $(2K + 1) \times (2L + 1)$.
- Пусть f – изображение.
- Свёртка, это изображение $f * g$, каждый пиксел которого вычисляется по следующему правилу:

$$(f * g)[col, row] = \sum_{k=-K}^K \sum_{l=-L}^L f[col - k, row - l] g[k + K, l + L].$$

- Или: центральный элемент ядра располагается над пикселем изображения. Новое значение пиксела вычисляется как сумма произведений соседних пикселей на элементы ядра.



Свёртка на краях изображения

- При вычислении свёртки для крайних пикселей изображения ядро выходит за границы изображения.
- Откуда взять недостающие пиксели изображения?
 1. Положить равными 0.
 2. Взять с другой стороны изображения.
 3. Повторить последний крайний пиксел изображения.
 4. Отразить изображение относительно границы.

	35	40	41	45	50	53
?	40	40	42	46	52	51
?	42	45	50	55	55	54
?	48	52	56	58	60	58
	56	60	65	70	75	62
	63	68	73	76	78	74

	35	40	4
0	40	40	4
0	42	45	5
0	48	52	5
	56	60	6

1

	35	40	41
51	40	40	42
54	42	45	50
58	48	52	56
	56	60	65

2

	35	40	4
40	40	40	4
42	42	45	5
48	48	52	5
	56	60	6

3

	35	40	4
40	40	40	4
45	42	45	5
52	48	52	5
	56	60	6

4

Простейшие фильтры

- Исходное изображение:



- Фильтр

0	0	0
0	1	0
0	0	0

0	0	0
0	0	1
0	0	0

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

- Результат



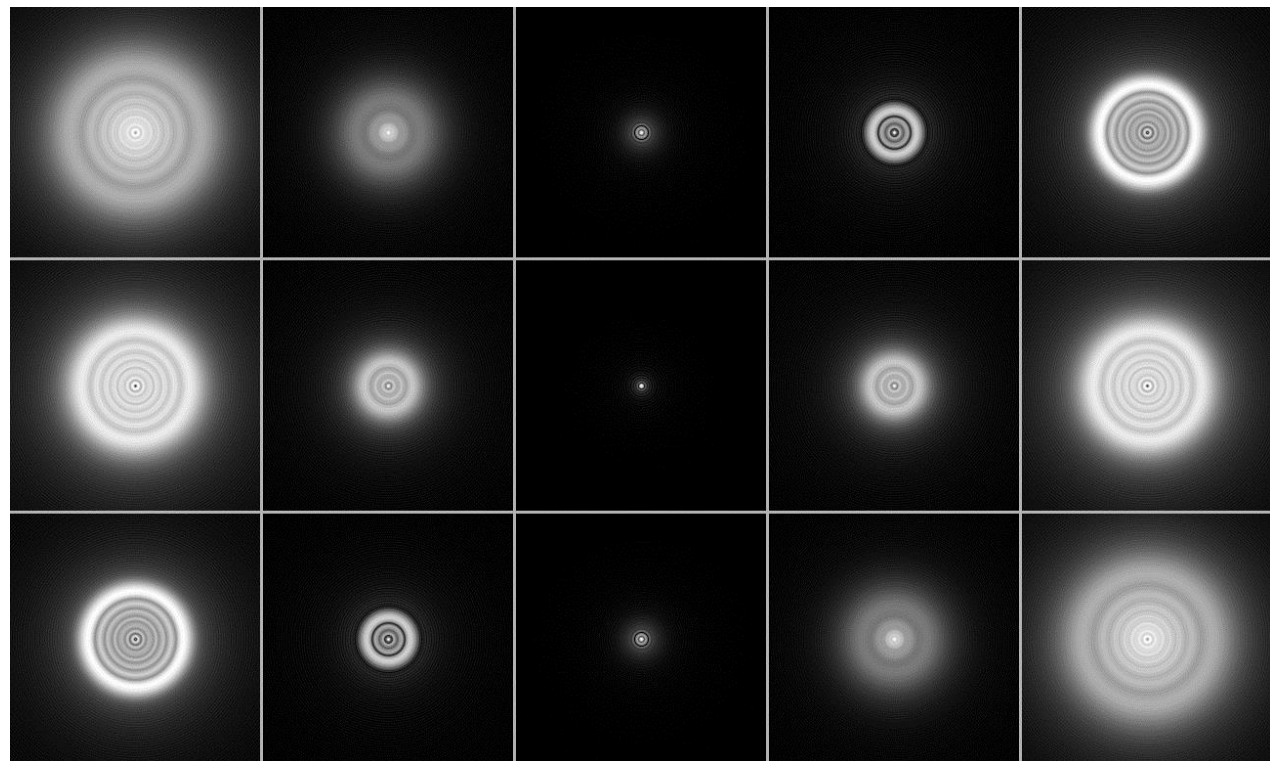
Размытие изображения в оптической системе

- Point Spread Function – оптический отклик системы на точечный источник света.

Отрицательная
абберация

Нулевая
абберация

Положительная
абберация



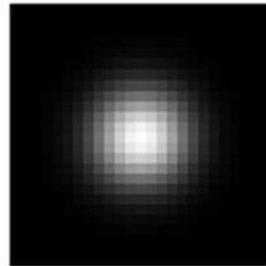
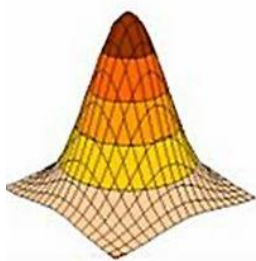
слишком близко ←

в фокусе

→ слишком далеко.

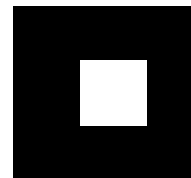
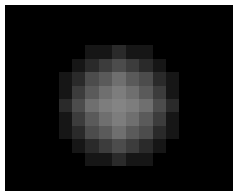
Размытие с помощью фильтра Гаусса

- Ядро фильтра Гаусса $G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{l^2+k^2}{2\sigma^2}}$



0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

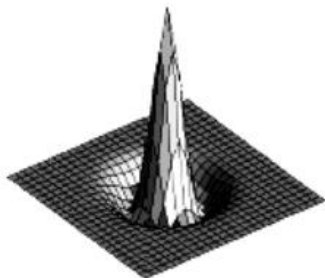
5 x 5, $\sigma = 1$



Другие применения свёртки

- Повышение резкости

$$\frac{1}{10} \cdot \begin{vmatrix} -1 & -2 & -1 \\ -2 & 22 & -2 \\ -1 & -2 & -1 \end{vmatrix}$$



- Выделение границ

0	1	0
1	-3	1
0	1	0



Другие применения свёртки

- Оператор Собеля

-1	-2	-1
0	0	0
1	2	1

+127

-1	0	1
-2	0	2
-1	0	1

+127



- Тиснение

-2	-1	0
-1	1	1
0	1	2

