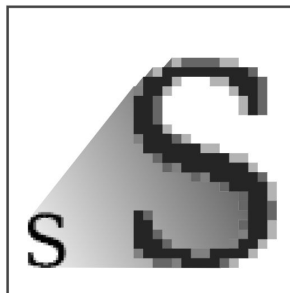


Цифровые изображения и их обработка

Что такое цифровое изображение?

- Изображение определяется как двумерная функция $F(x,y)$, где x и y - это пространственные координаты, а значение F является **яркостью** в данной паре координат. Когда x , y , и значения F являются дискретными и конечными, мы зовем такое изображение **цифровым изображением (digital image)**.
- Цифровое изображение бывает **растровым** и **векторным**.



РАСТР
.jpeg .gif .png



ВЕКТОР
.svg

Цифровое растровое изображение

- **Цифровое растровое изображение** представляется как прямоугольный двумерный массив чисел, при этом каждое числовое значение соответствует наименьшему логическому элементу изображения или **пикселю**.
- Значение конкретного пикселя представляет закодированный цвет. Количество памяти (битов), необходимое для его кодирования называют **глубиной цвета**.

174	190	60	31	41	17	26	18	20	28	15	21	17	16	24	38	49	22	14	13	20	17	76
189	59	41	26	35	18	24	17	16	17	19	15	18	19	42	119	132	76	28	12	15	14	36
77	48	52	34	17	11	23	34	34	16	29	11	14	61	60	158	194	162	141	38	14	45	29
61	53	32	27	23	15	27	63	58	31	67	17	16	90	27	137	191	183	180	121	35	56	74
81	72	49	48	29	9	31	62	94	38	58	56	67	40	39	191	196	202	196	167	75	35	91
93	92	83	78	66	15	16	56	85	86	35	27	28	50	146	198	198	198	185	156	98	55	90
89	98	96	104	85	45	36	64	84	93	94	70	91	151	184	190	195	194	186	121	112	117	70
93	102	116	108	110	93	71	67	37	86	107	123	129	146	160	174	182	167	141	126	109	84	79

Глубина цвета изображений

- Глубина часто выражается в количестве бит на пиксель: *bpp*(bits per pixel) и варьируется от 1 до более 48.
- Количество оттенков, которые можно получить из пикселя заданной глубины d равно 2^d .

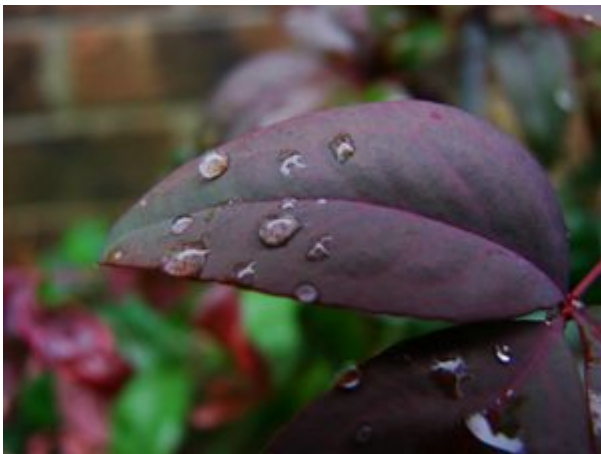


4-битное
изображение



8-битное
изображение

Глубина цвета изображений



24-битное изображение
16,777,216
цветов



8-битное изображение 256 цветов



4-битное изображение
16 цветов



2-битное изображение, 4 цвета



1-битное изображение, 2 цвета

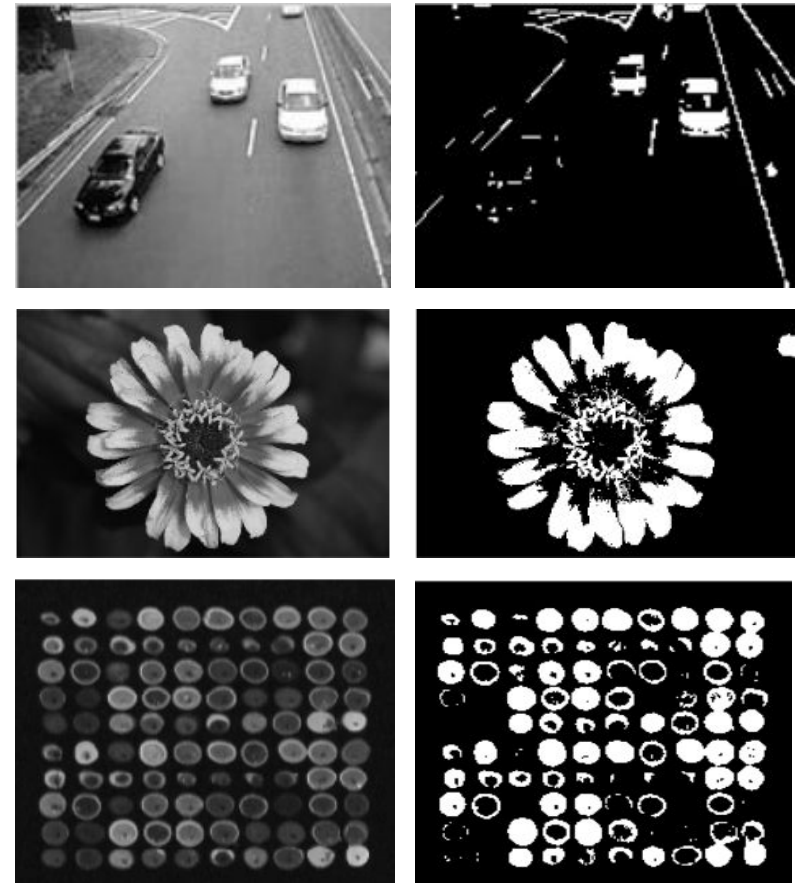
Способы кодирования цвета

Глубина цвета (d) так же определяет способ кодирования цветовой информации.

Например, для $d = 1$ (**бинарные** изображения, 2 оттенка) и для $d = 8$ (**полутонные** изображения, 256 оттенков) пиксели кодируются с помощью одномерной шкалы яркости.

Обычно это набор из чёрного и белого цветов и промежуточных оттенков серого.

Такие изображения называются монохромными.



$d = 8$

$d = 1$

Способ кодирования «реальных» ЦВЕТОВ

«Реальный» цвет представляется смешением **красного**, **зелёного** и **синего** цветов. Он, соответственно, кодируется 3 числами, отвечающими за их яркость. Такое кодирование называют RGB-моделью.

Для $d = 16$ (High Color, 65 536 различных цветов) 5 бит используется для представления красной составляющей, 5 для синей, 6 для зелёной.

Для $d = 24$ (True Color, 16,7 млн цветов) используется 8 бит для представления красной, синей и зелёной составляющих.







Способ кодирования с помощью палитры

- При относительно небольшой глубине цвета (1 – 12) изображения кодируются с помощью дискретного набора цветов, каждый из которых описан в палитре (массиве цветов).


























Пиксель представляет собой индекс из этого массива.

Примеры:

0 =  Палитра для
1 =  глубины пикселя
2 = 
3 =  $d = 2$

0	0	1	2	3
0	1	2	3	2
1	2	3	2	1
2	3	2	1	0
3	2	1	0	0

Закодированное
индексированное
изображение

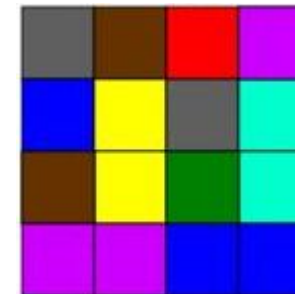
Представление
индексированного
изображения

0	
1	
2	
3	
4	
5	
6	
7	

Палитра(неполная)
для глубины
пикселя $d = 3$

4	3	0	2
1	7	4	5
3	7	6	5
2	2	1	1

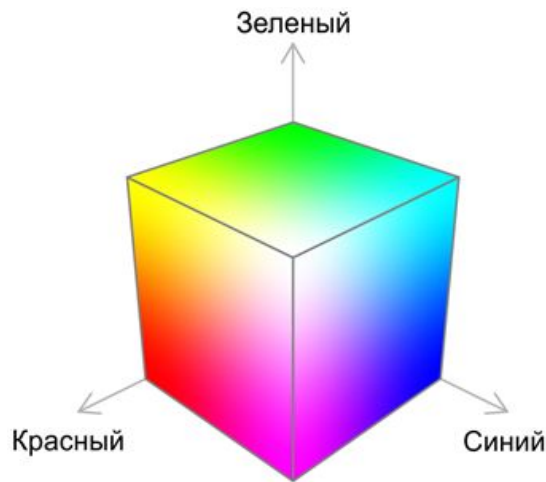
Закодированное
индексированное
изображение



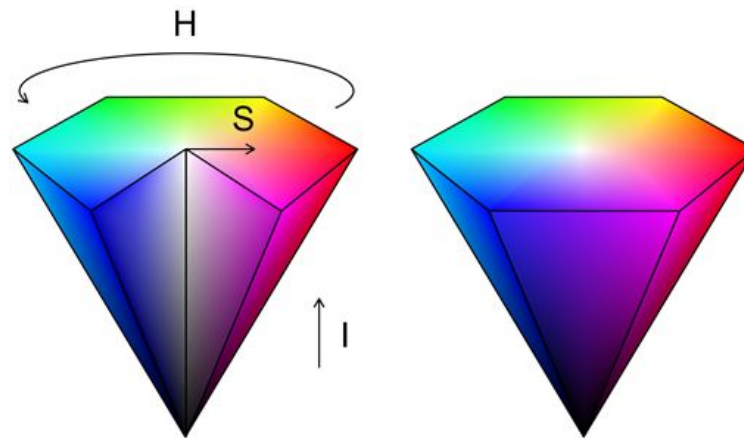
Представление
индексированного
изображения

Цветовые модели (пространства)

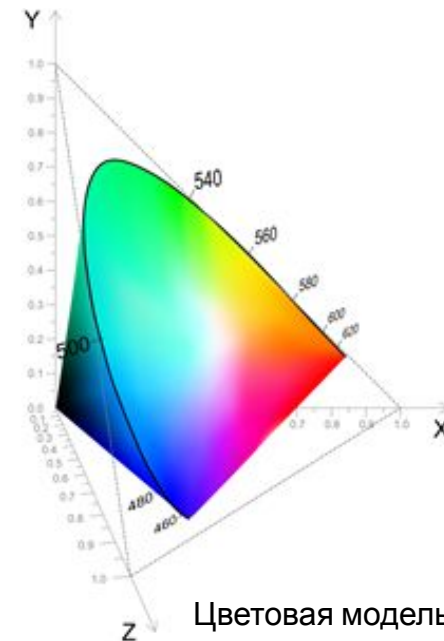
- Цвет пикселя можно кодировать целым вектором значений, где каждая компонента вектора будет отвечать за интенсивность определенной цветовой составляющей: $pixel = (x_1, x_2, \dots, x_n)$.
- Способ разделения цветового оттенка на составляющие его компоненты называется цветовой моделью.



Цветовая модель RGB



Цветовая модель HSI



Цветовая модель CIE XYZ

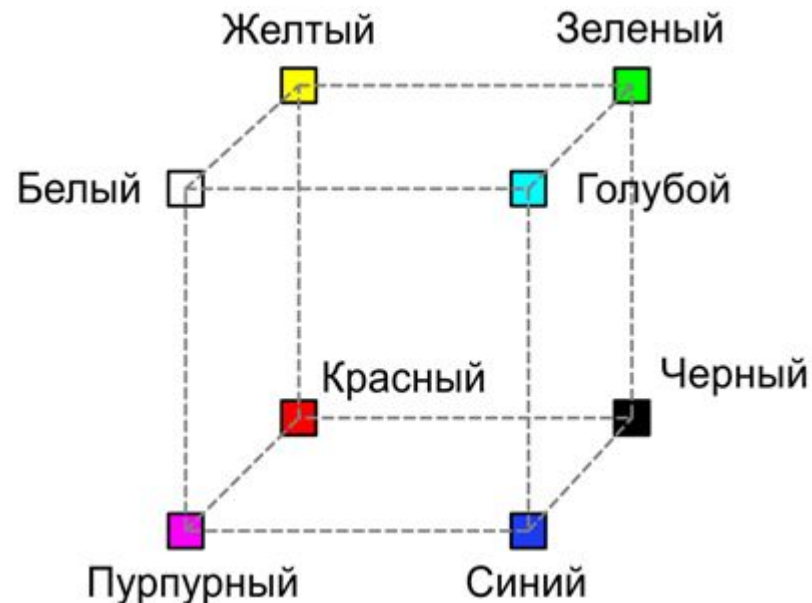
Цветовая модель RGB

- В модели RGB (**Red** – **красный**, **Green** – **зелёный**, **Blue** – **голубой**) все цвета получаются путём смешения трёх основных цветов – красного, зелёного и синего – в различных пропорциях.
- Пример использования - мониторы.



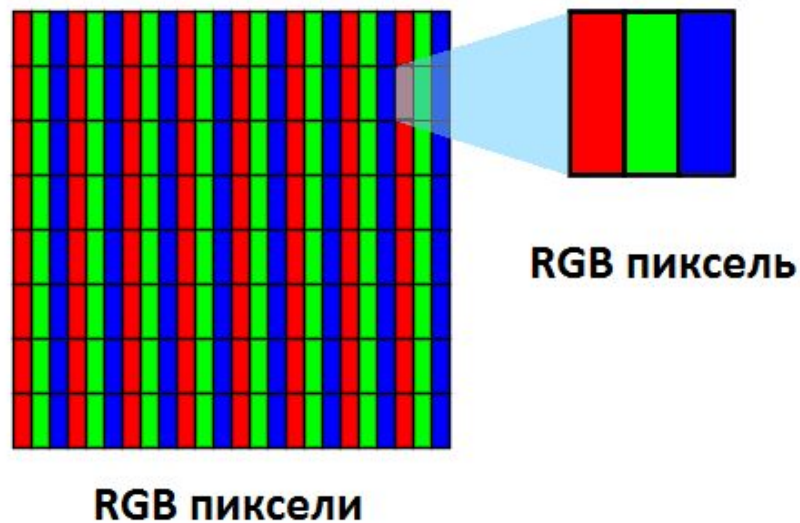
Цветовая модель RGB

Доля каждого базового цвета в имеющемся цветовом оттенке может восприниматься, как координата в соответствующем трёхмерном пространстве – **цветовом кубе**.

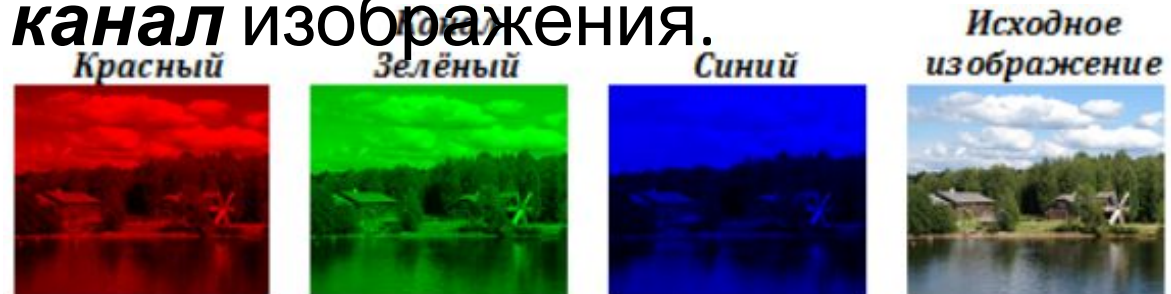


Цветовая модель RGB

Пиксель в такой модели представляется как вектор коэффициентов интенсивности красной, зеленой и синей составляющих: $pixel = (r, g, b)$.

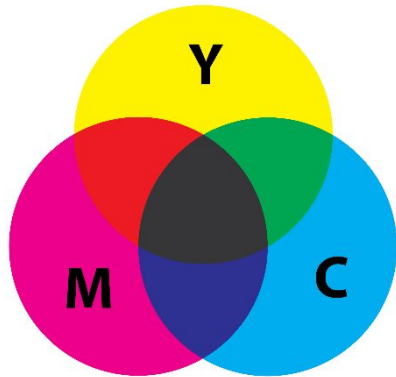


Если для всех пикселей изображения взять без изменения только одну составляющую (одинаковую для всех), а остальные занулить, мы получим **цветовой канал** изображения.



Цветовая модель СМУК

Субтрактивная модель СМУ (от англ. cyan — голубой, magenta — пурпурный, yellow — жёлтый) получает цвета путём вычитания из белого цвета первичных RGB цветов.



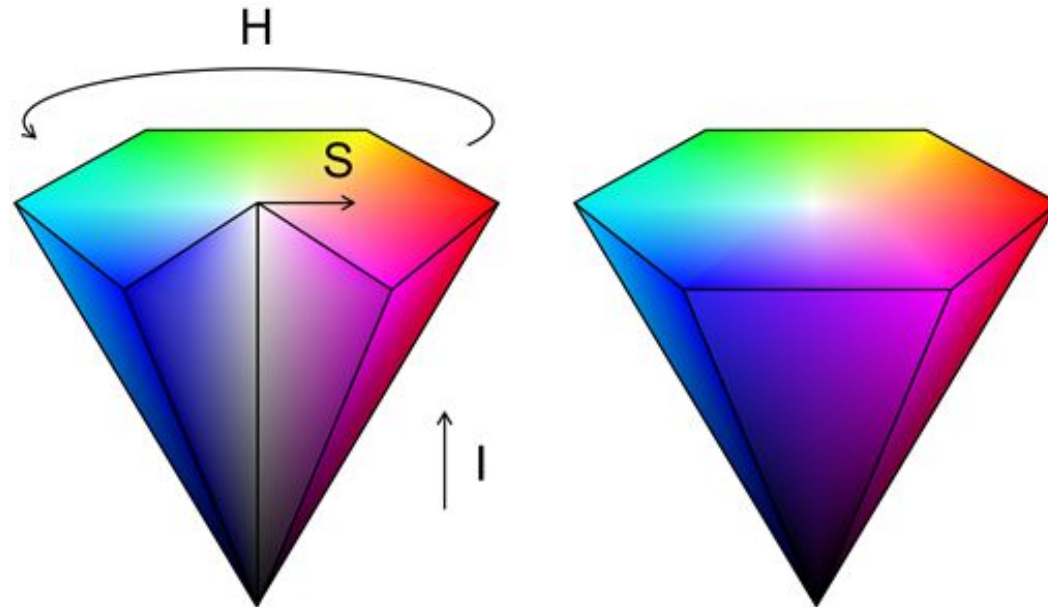
$$\begin{cases} C = 1 - R \\ M = 1 - G \\ Y = 1 - B \end{cases}$$

На практике модель СМУ расширяют до модели СМУК, добавляя к трём цветам ещё и чёрный $K^{[?]}$ (от англ. black).

Модель используется в полиграфии для печати изображений.

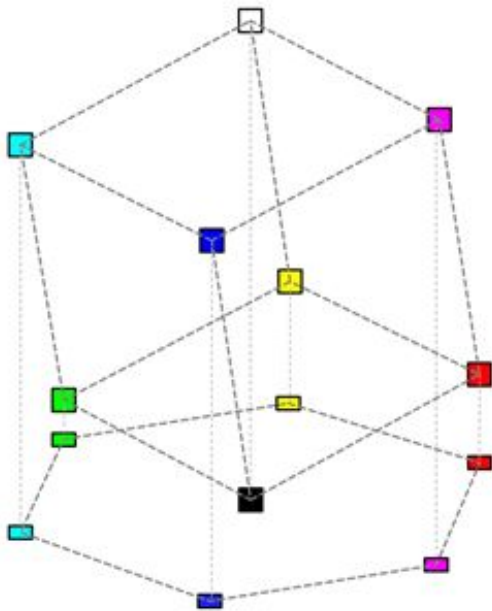
Цветовая модель HSI (HSL, HLS)

HSI (от англ. hue - тон, saturation - насыщенность, intensity - интенсивность) — цветовая модель, в которой цветовыми координатами являются тон, насыщенность и интенсивность.

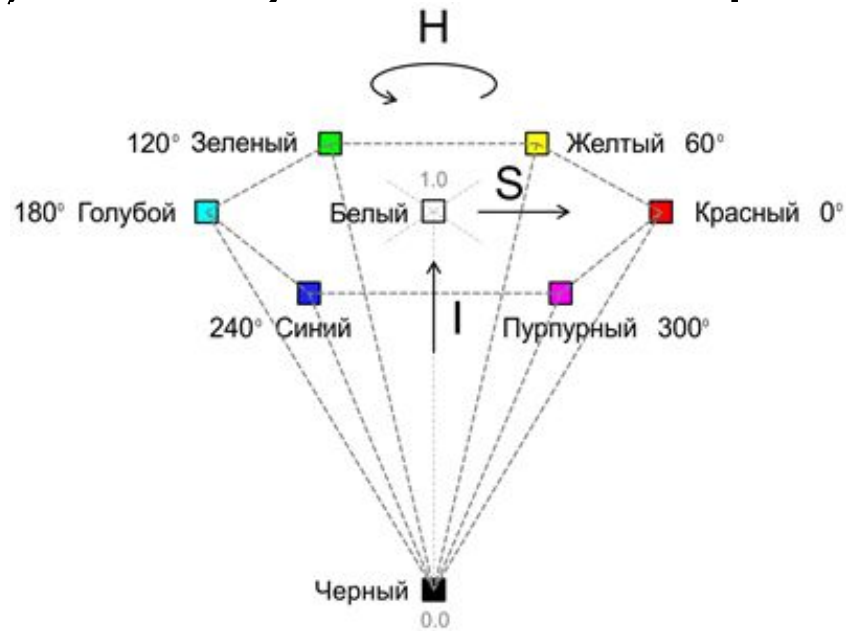


Цветовая модель HSI (HSL, HLS)

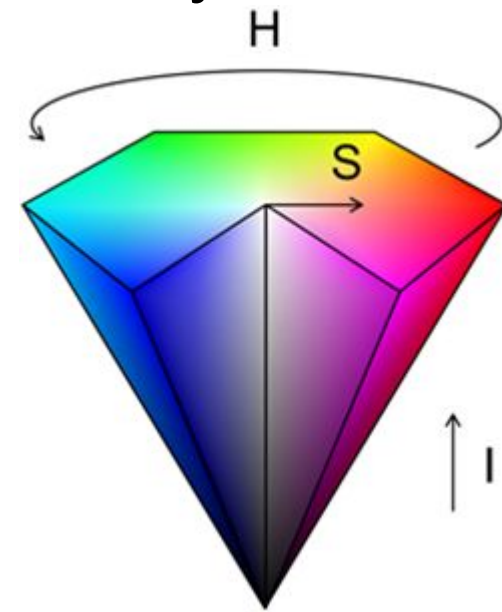
Если спроектировать RGB куб в направлении диагонали белый-чёрный и добавить вертикальную ось светлоты (или интенсивности), то получаем шестигранный конус HSI:



Проекция RGB куба в направлении диагонали белый-чёрный



Шестигранный конус HSI



Шестигранный конус HSI

Цветовая модель HSI

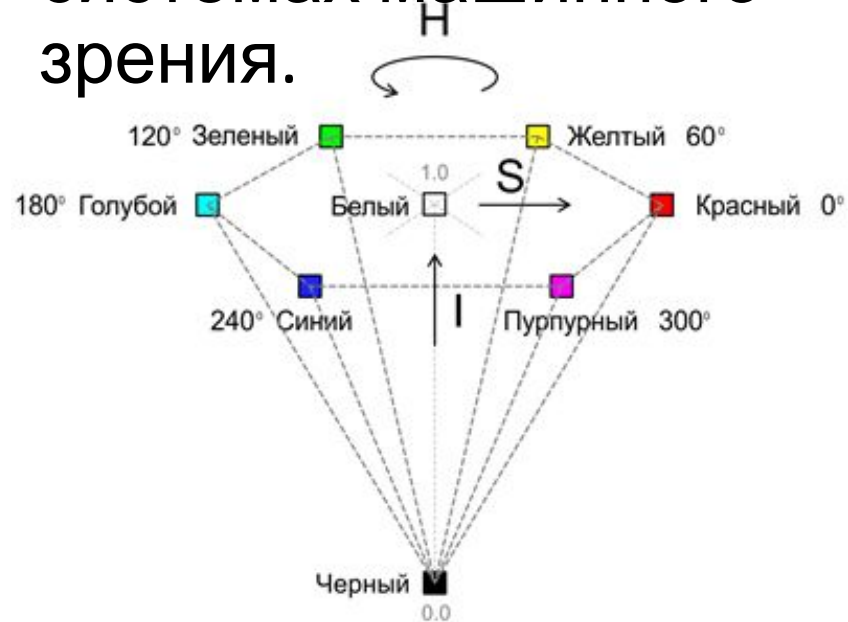
Алгоритм перевода из RGB в HSI можно выполнить, воспользовавшись следующими формулами:

$$\left\{ \begin{array}{l} H = \begin{cases} \theta; B \leq G \\ 360 - \theta; B > G \end{cases}, \text{ где } \theta = \arccos \left(\frac{\frac{1}{2} * ((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \\ S = 1 - \frac{3}{(R + G + B)} \min(R, G, B) \\ I = \frac{1}{3}(R + G + B) \end{array} \right.$$

СЛОЖНО

Цветовая модель HSI

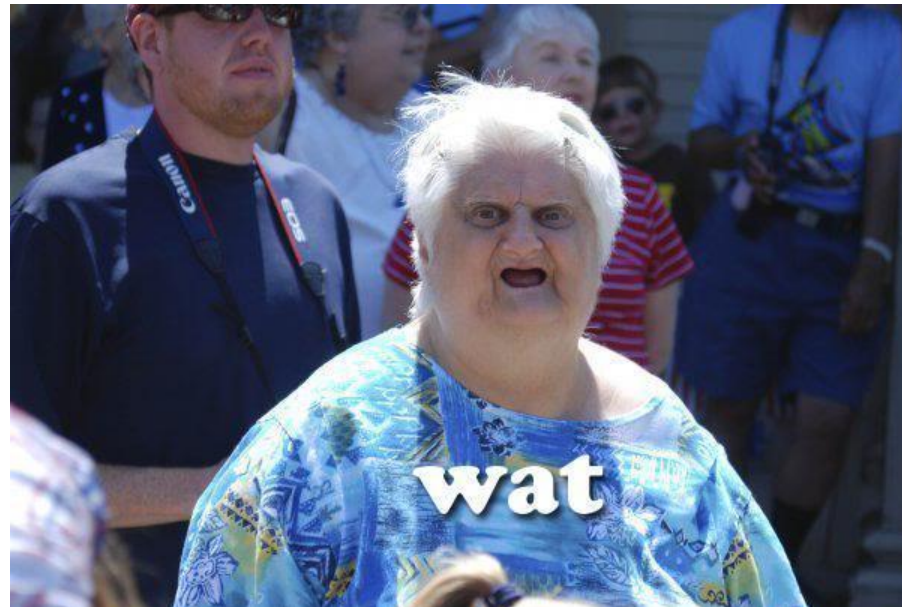
Модель очень популярна среди дизайнеров и художников, а также в системах машинного зрения.



	Интенсивность	Тон	Насыщенность
Уменьшенное значение			
Исходное значение			
Увеличение значения			

Цветовая модель CIE XYZ

Цветовая модель CIE XYZ была разработана с целью получить значения цвета и возможности отличать один спектр от другого, отталкиваясь от фотометрической яркости излучения (Y).

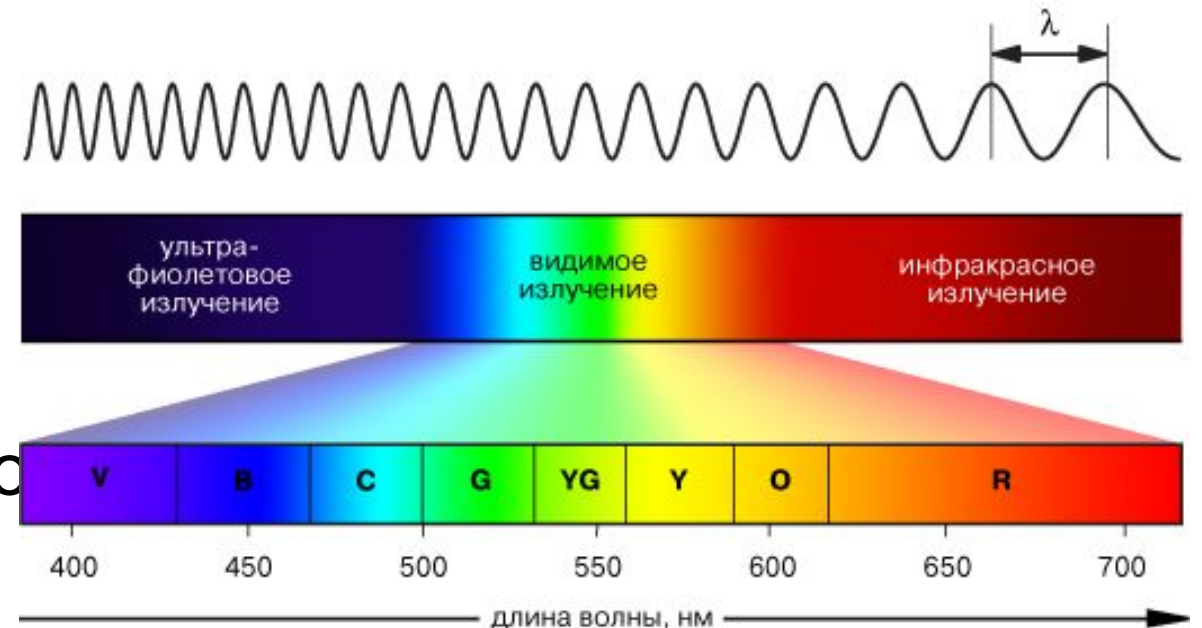


Предыстория...

- Что такое цвет?
- Визуальный колориметр.
- «Неидеальность» системы RGB.

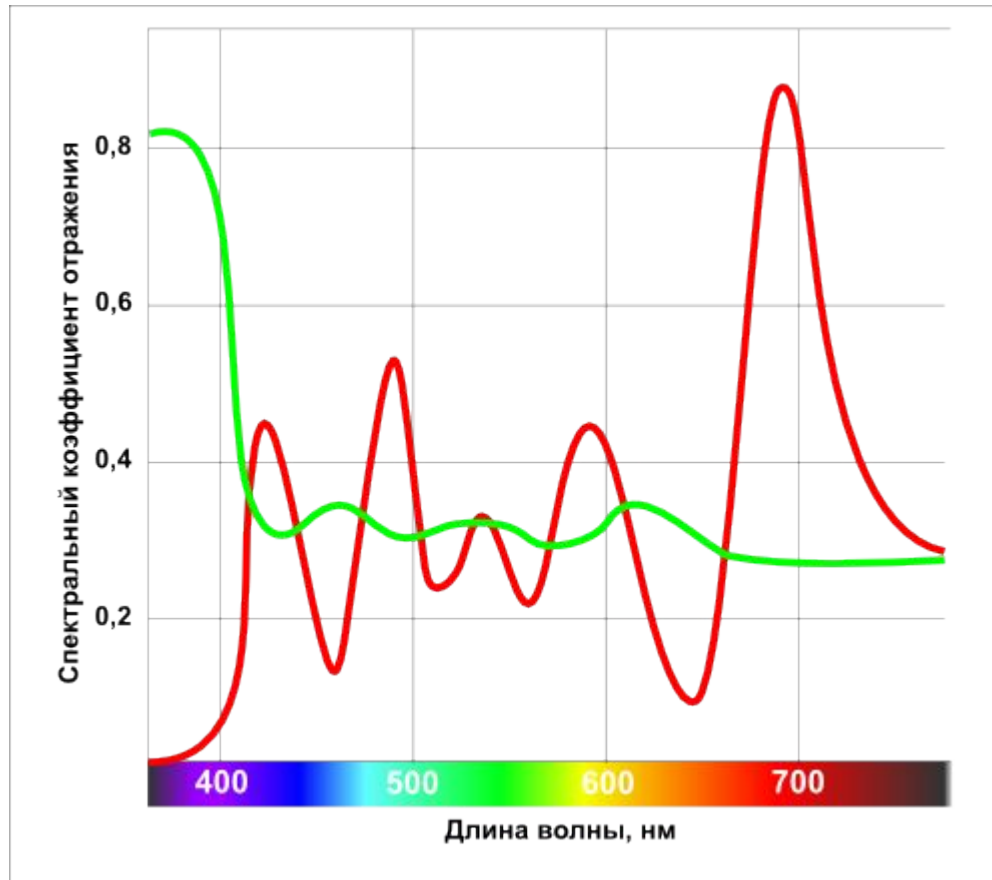
Что такое цвет?

- **Цвет** — это ощущение, которое испытывает человек при воздействии **света** на его глаза.
- **Свет** — это электромагнитное излучение.
Диапазон длин волн света, видимых для глаза: 390-740 нм.
- Физике известны и легко поддаются измерению параметры света:
мощность и **спектральный состав**
(распределение мощностей по длинам волн — спектр).



Спектр электромагнитных излучений и спектр видимого света

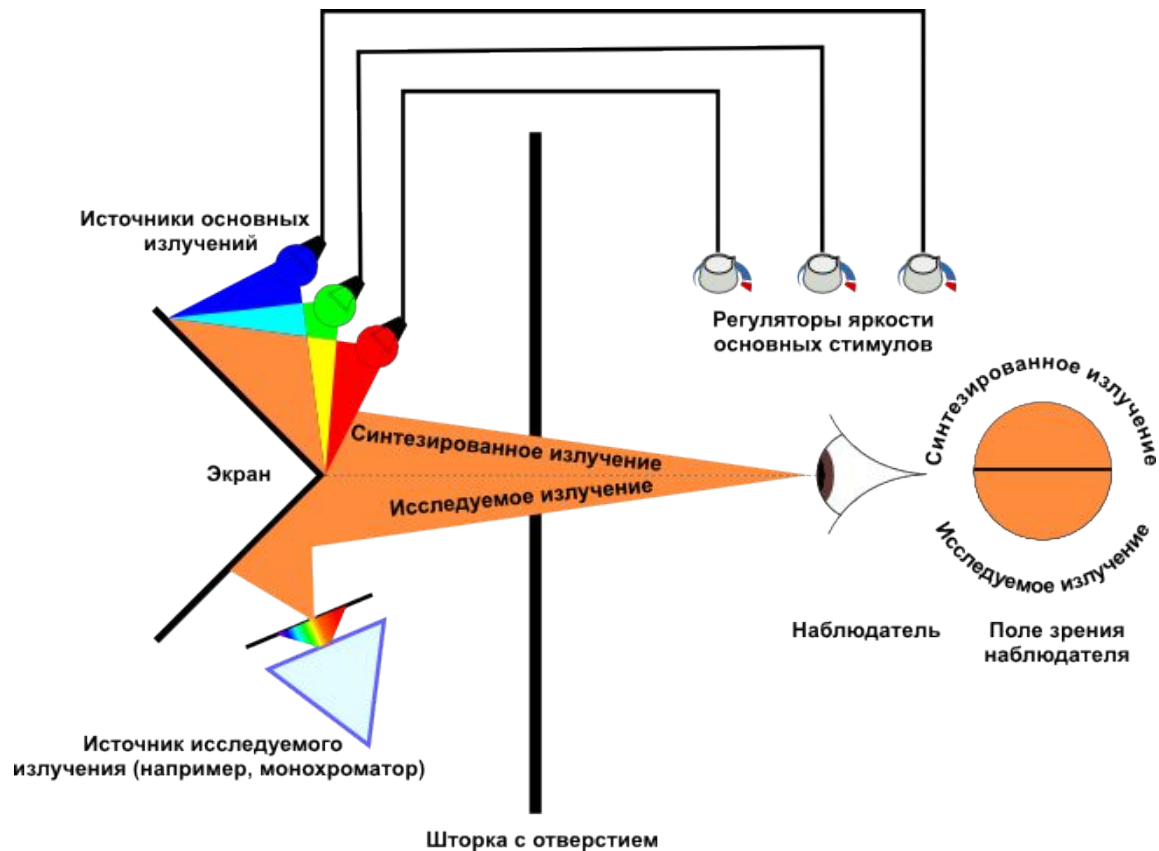
Что такое цвет?



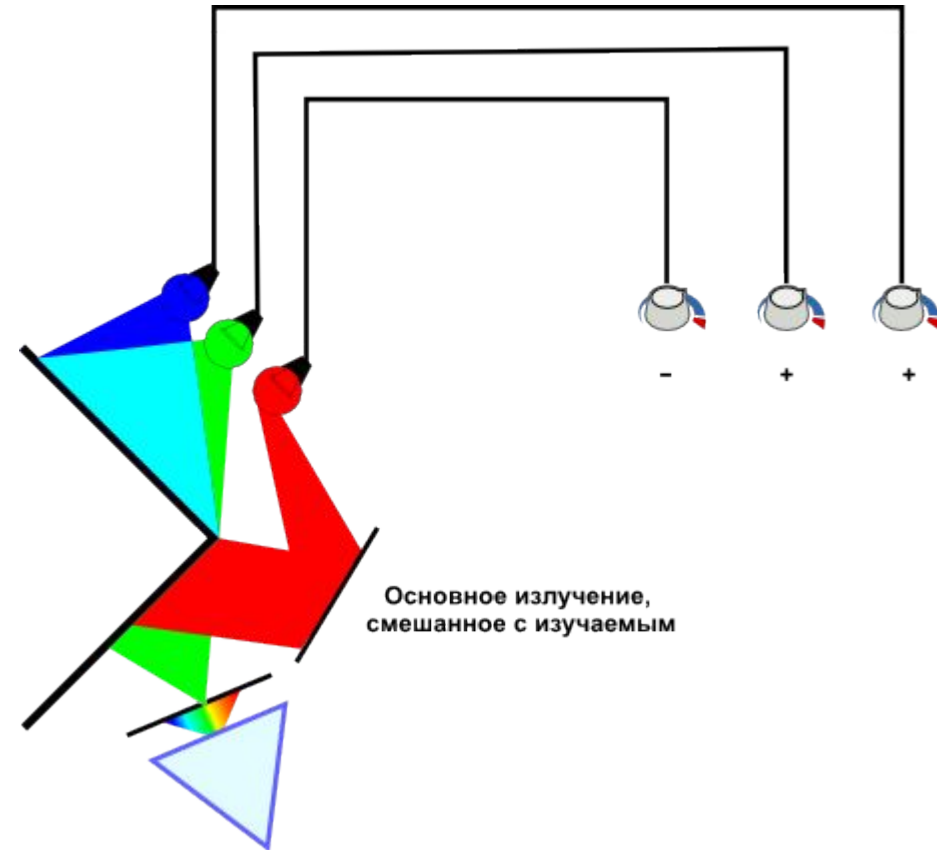
Цвет — это реакция зрительной системы человека на свет, а не свойство излучения.

Два существенно разных распределения спектра определяют один и тот же цвет – серый.

Визуальный колориметр



Процедура измерения
цвета исследуемого
излучения



Одно основное излучение прибора
смешивается с исследуемым
излучением

«Неидеальность» системы RGB

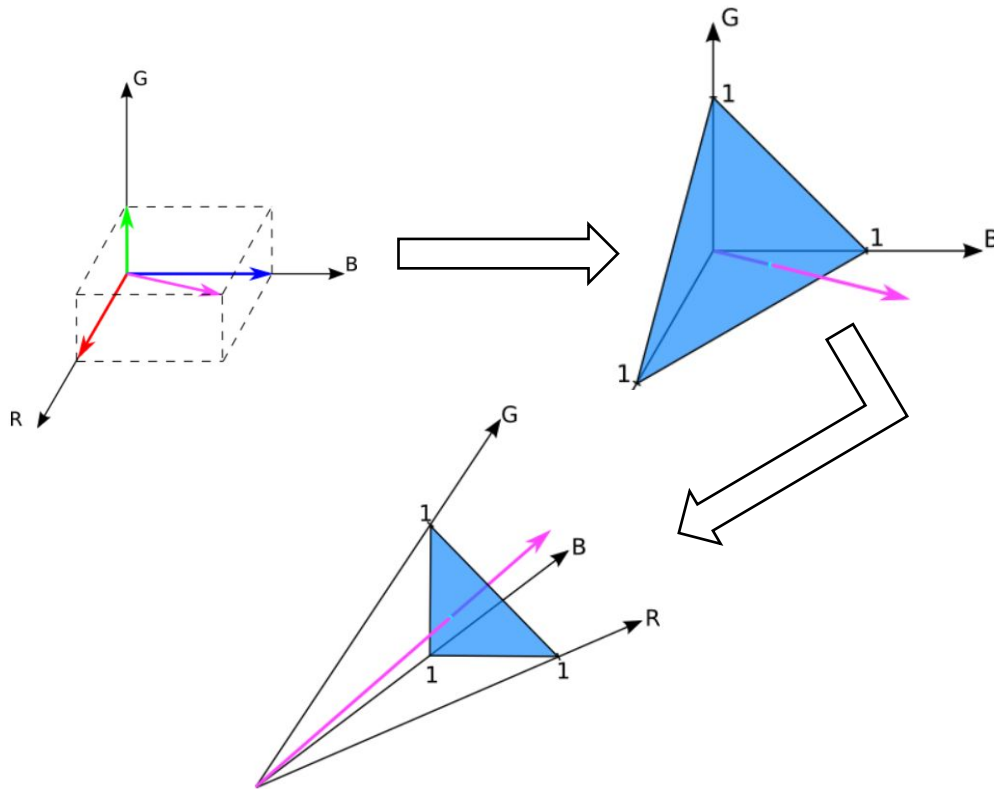
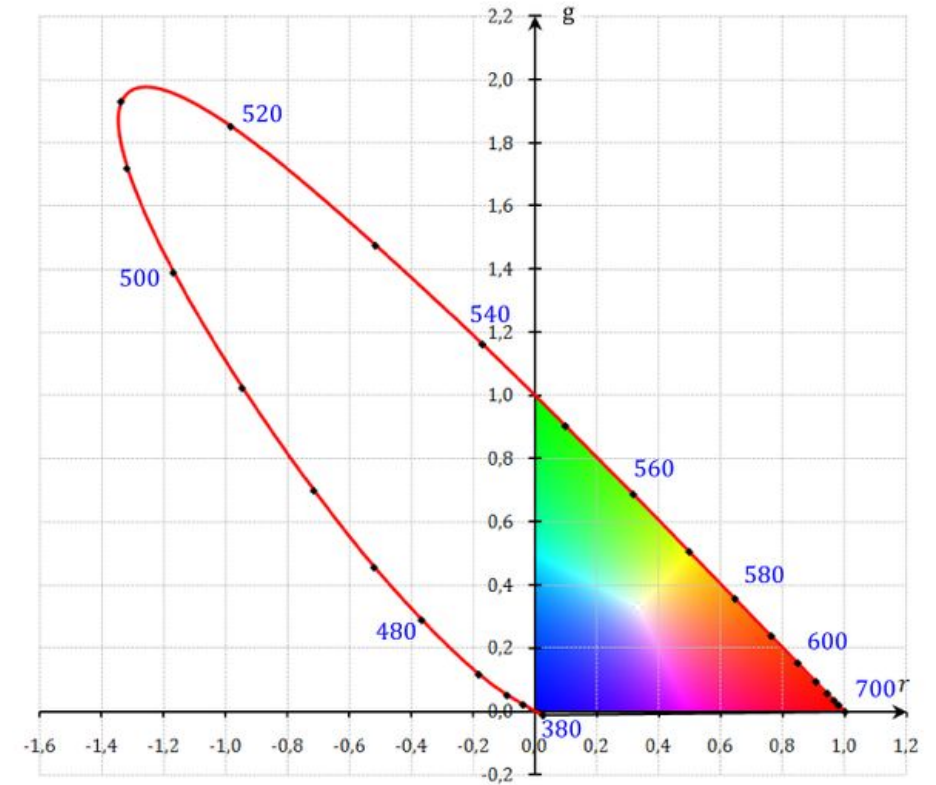


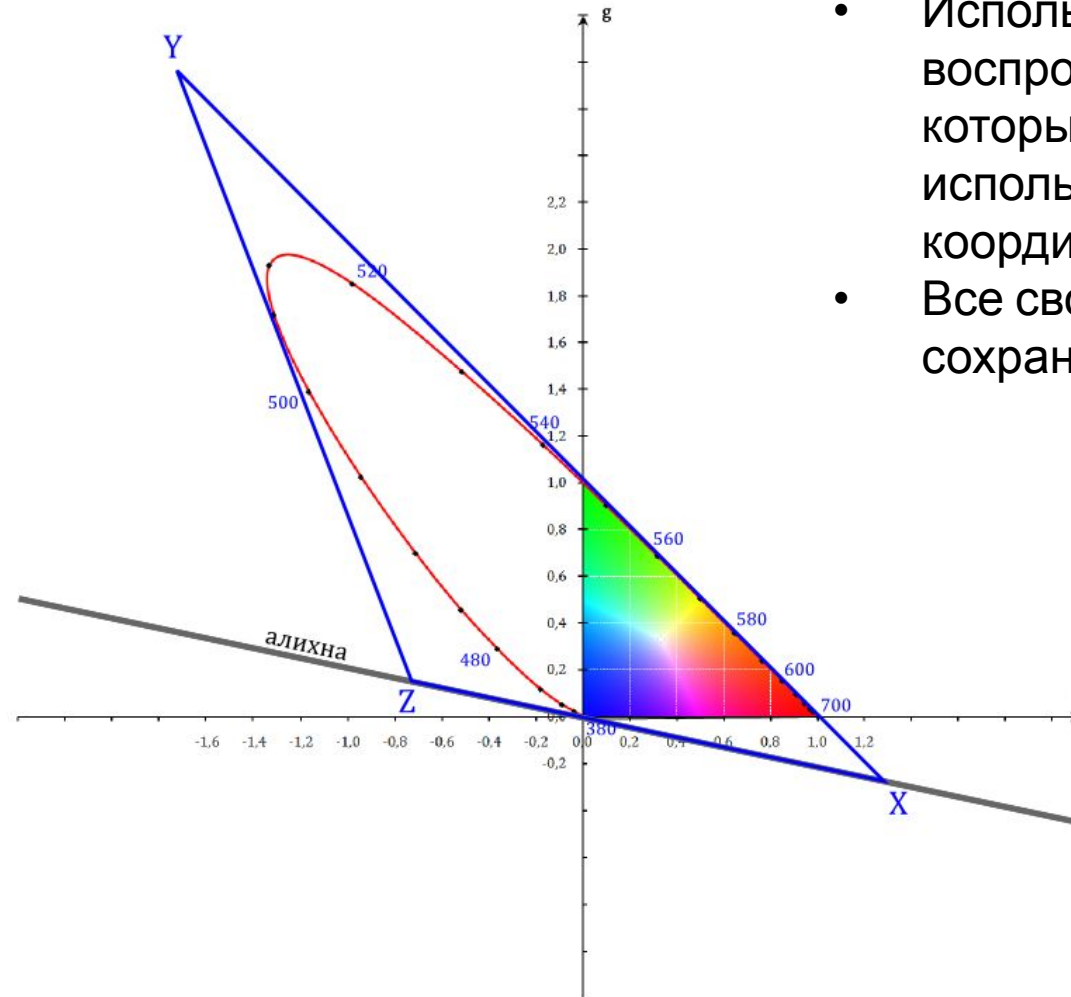
Диаграмма цветности показывает *соотношение* основных цветов независимо от светлоты.



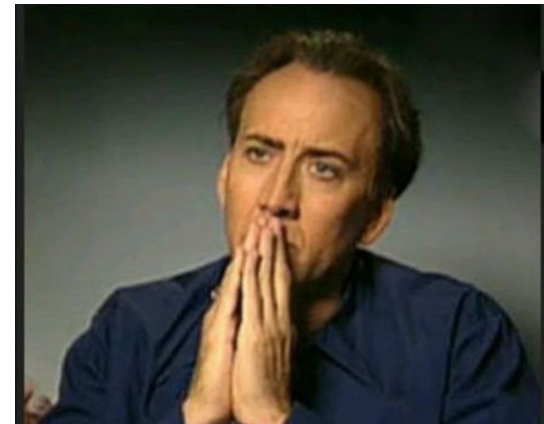
Треугольник цветностей основных излучений RGB и линия спектрально чистых излучений, ограничивающая область реальных цветов (красная линия).

Цветовое пространство CIE XYZ

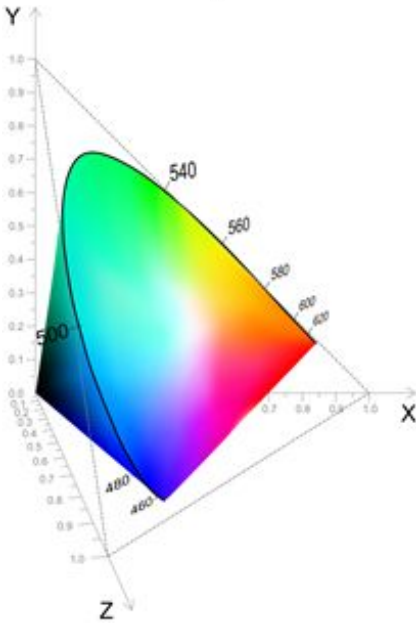
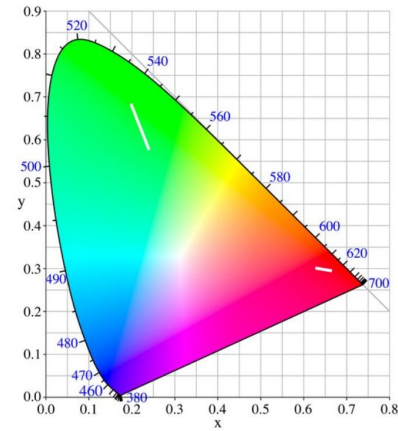
А что если...



- Используем цвета, которые невозможно воспроизвести и увидеть, но координаты которых можно с лёгкостью использовать в уравнениях наравне с координатами реальных цветов.
- Все свойства смешения цветов для них сохраняются.



Цветовое пространство CIE XYZ



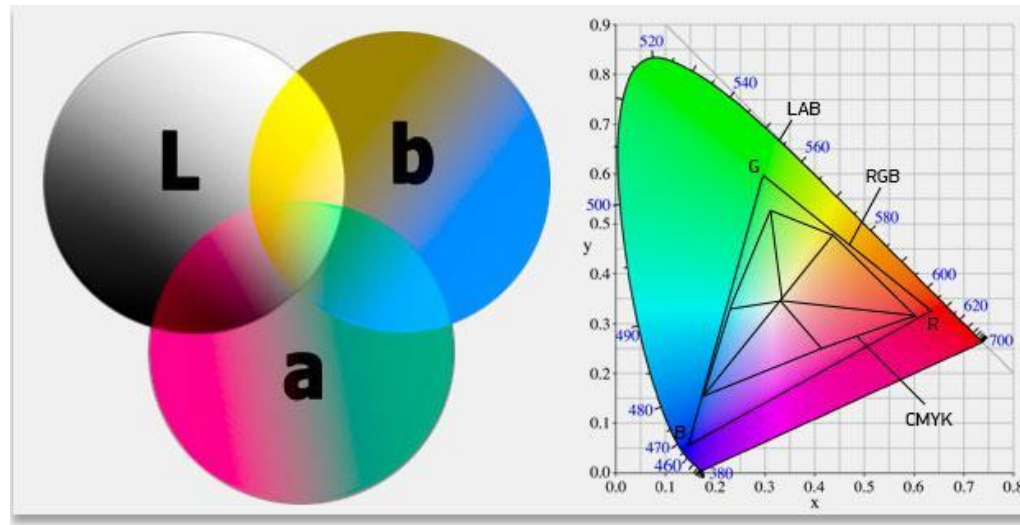
Основная причина создания системы XYZ — облегчения расчётов:

- + Координаты цвета и цветности видимых излучений будут всегда положительными.
- + Яркость полностью перешла к одному из трёх компонент системы — Y .

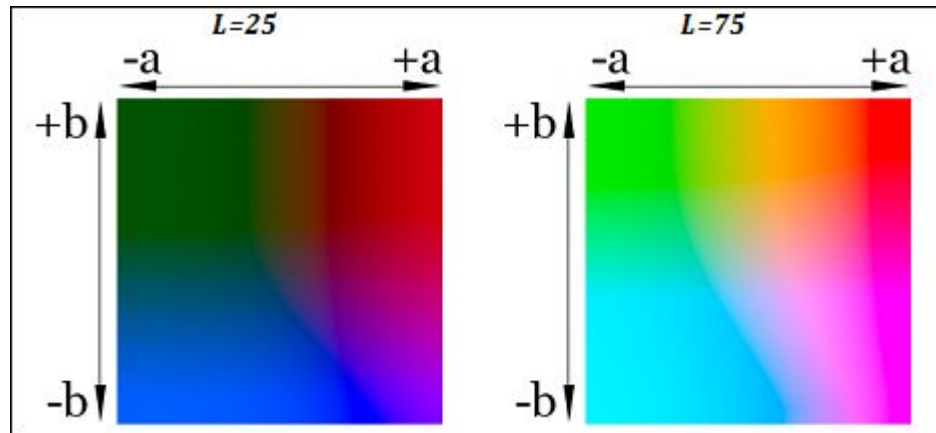
Единственный недостаток системы — это то, что равные отрезки на разных участках диаграммы не означают одинаковую воспринимаемую разницу в цвете.

Цветовое пространство CIE L*a*b*

Цветовое пространство CIE L*a*b* на данный момент является международным стандартом и разрабатывалось чтобы устранить нелинейность системы CIE XYZ с точки зрения человеческого восприятия.



Цветовое пространство CIE $L^*a^*b^*$



Срезы цветового тела CIE $L^*a^*b^*$ для двух значений светлоты.

В системе CIE $L^*a^*b^*$ координата L означает светлоту (в диапазоне от 0 до 100), а координаты a, b – означают позицию между зелёным-красным, и синим-жёлтым цветами.

Цветовое пространство CIE $L^*a^*b^*$ выводится из CIE XYZ.

Какую практическую пользу приносит CIE L*a*b*?

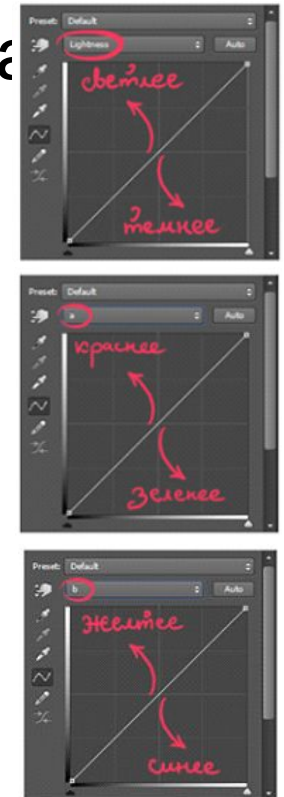
- CIE L*a*b* нашёл широкое применение в программном обеспечении для обработки изображений в качестве **промежуточного** цветового пространства.
- Он позволяет переводить картинки из одной цветовой модели в другую с выжимкой максимального качества из устройства, поскольку его цветовой диапазон шире, чем у электронных носителей.
- Adobe Photoshop по умолчанию использует его как посредник в конвертации изображений.

Какую практическую пользу приносит CIE L*a*b*?

- Редактирование изображений в таком пространстве становится мощным инструментом цветокоррекции.



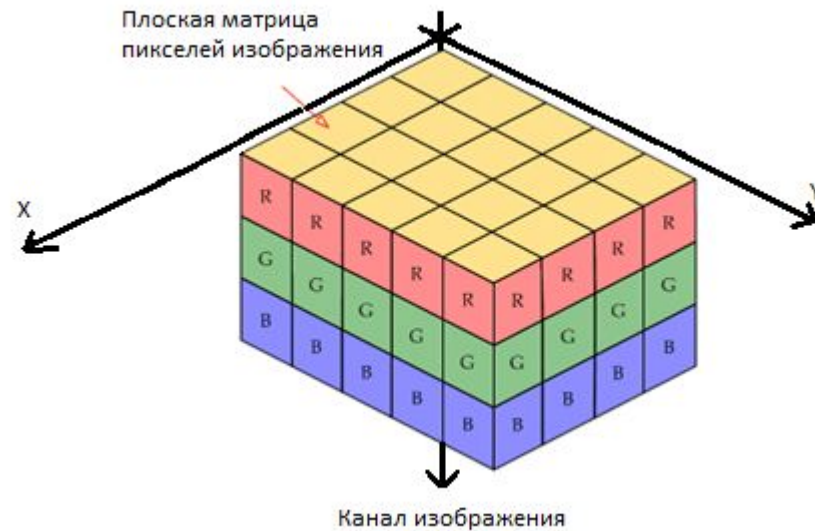
Пример коррекции изображения кривыми L и b в Adobe Photoshop.



Кривые L, a и b в Adobe Photoshop.

3-х мерное представление изображения

В соответствии с различными способами кодирования цвета, мы можем представлять цифровое изображение в виде не двумерного, а трехмерного массива, где за «дополнительное» измерение отвечает количество каналов изображения:



RGB 3D матрица
изображения

Цифровая обработка изображений

- Поэлементные преобразования изображений.
- Матричные преобразования изображений.
- Изменение размера изображения.

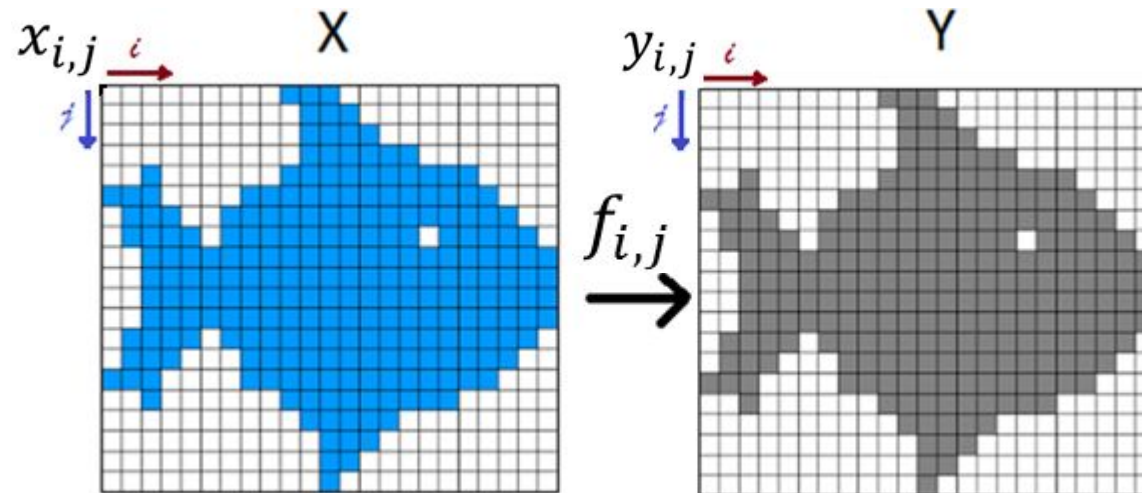


Поэлементные преобразования изображений

Пусть $x_{i,j}$, $y_{i,j}$ – значения яркости исходного и получаемого после обработки пикселя изображения в точке i, j .

Тогда, существует такая функциональная однозначная зависимость между яркостями $x_{i,j}$ и $y_{i,j}$ – $f_{i,j}$, что: $y_{i,j} = f_{i,j}(x_{i,j})$.

Если обработка однородная, то $y = f(x)$.



Поэлементные преобразования изображений

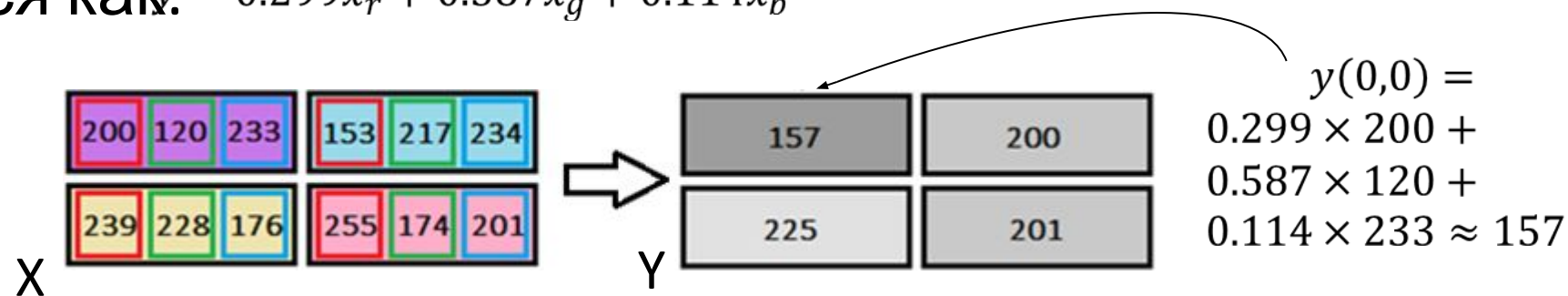
Примеры:

- Перевод изображения из одного цветового пространства в другое.
- Преобразование изображения в градации серого.
- Бинаризация.
- Контрастирование.



Преобразование изображения в градации серого и бинаризация

При конвертации в полутоновое изображение каждый пиксель вычисляется как:

$$y = 0.299x_r + 0.587x_g + 0.114x_b$$


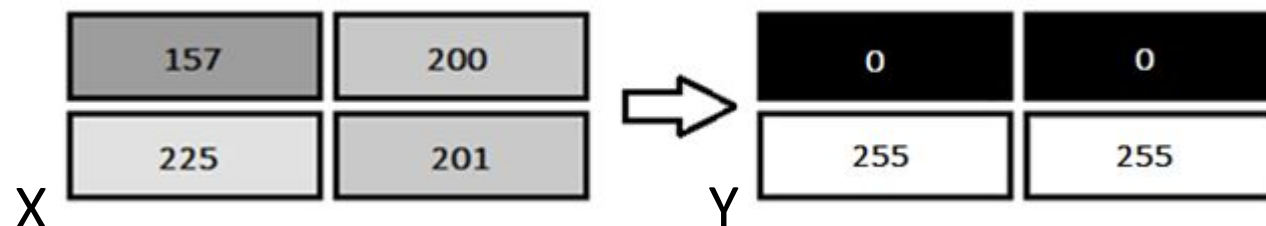
При бинаризации каждый пиксель вычисляется как:

$$y = \begin{cases} 255, & \text{если } x > threshold, \\ 0, & \text{иначе} \end{cases}$$

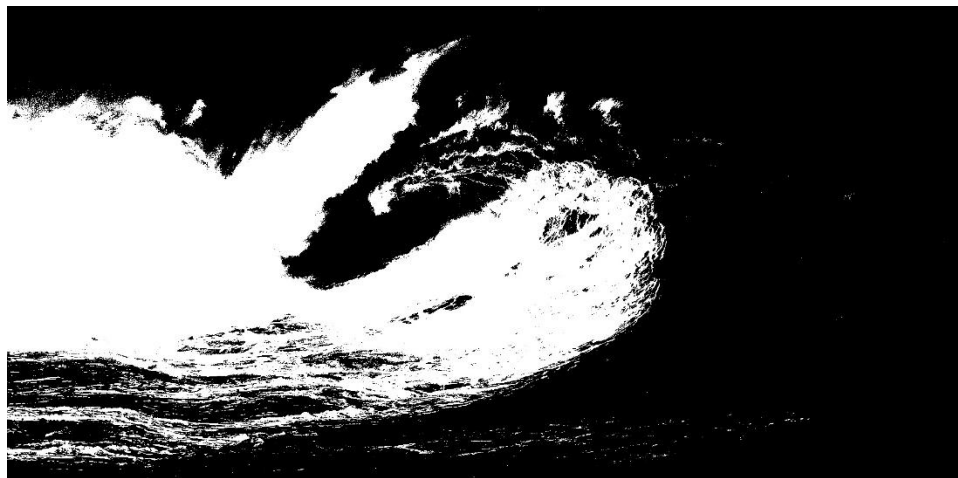
Существует множество алгоритмов вычисления параметра $threshold$.

Пуст

$threshold = 200$



Преобразование изображения в градации серого и бинаризация



threshold = 180

Контрастирование изображения

При контрастировании изображения каждый пиксель вычисляется по формуле: $y = \frac{x - x_{min}}{x_{max} - x_{min}} (y_{max} - y_{min}) + y_{min}$.

Пример:

Дано полутоновое изображение с $x_{min} = 180, x_{max} = 240$.

Цель – добиться минимальной яркости в изображении $y_{min} = 0$, максимальной – $y_{max} = 255$.

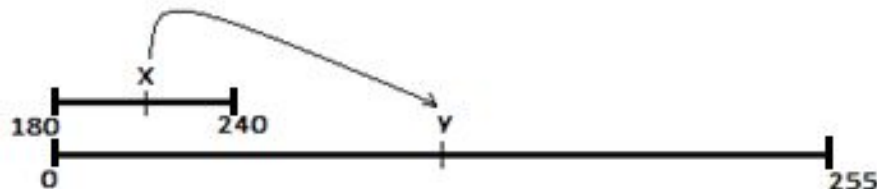
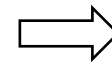


Иллюстрация
формулы



$$\begin{aligned}x_{min} &= 180 \\x_{max} &= 240\end{aligned}$$

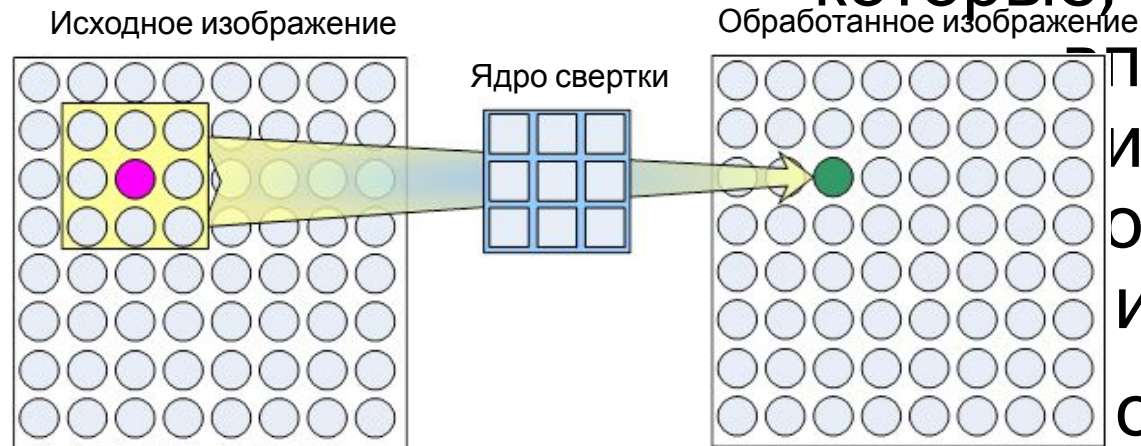
$$\begin{aligned}y_{min} &= 0 \\y_{max} &= 255\end{aligned}$$

Матричные преобразования изображений

Матричные фильтры вычисляют **новое значение пикселя** учитывая значения **окружающих его пикселей**.

Для этого используется небольшая **матрица свертки**, которая накладывается на изображение своим центральным элементом на вычисляемый пиксель. Матрица свертки хранит веса, на

которые,



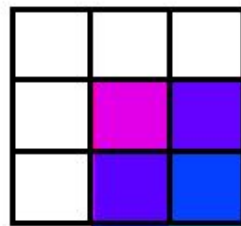
Последствия умножаются пиксели исходного изображения, попавшие в окрестность ядра матрицы.

Полученные значения складываются в результирующий пиксель.

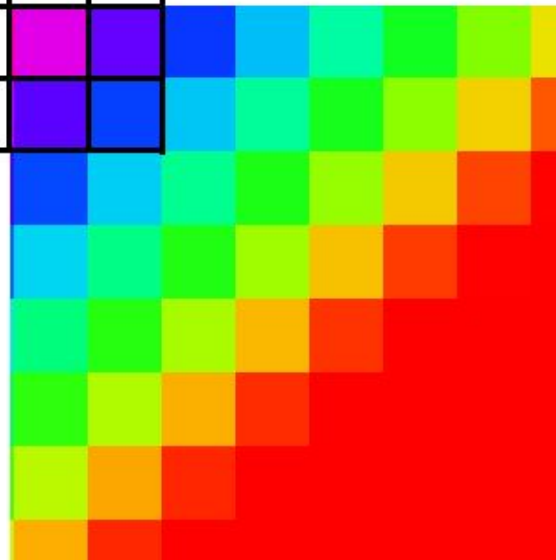
Матричные преобразования изображений

- При матричных преобразованиях координата y_{ij} будет рассчитываться следующим образом: $y_{ij} = \sum_k \sum_l f(k, l) \times x_{k+i, l+j}$, где $f(k, l)$ - матрица свертки (или ядро).

Матрица

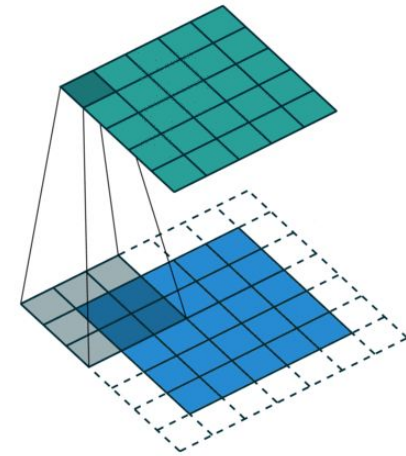


Изображение



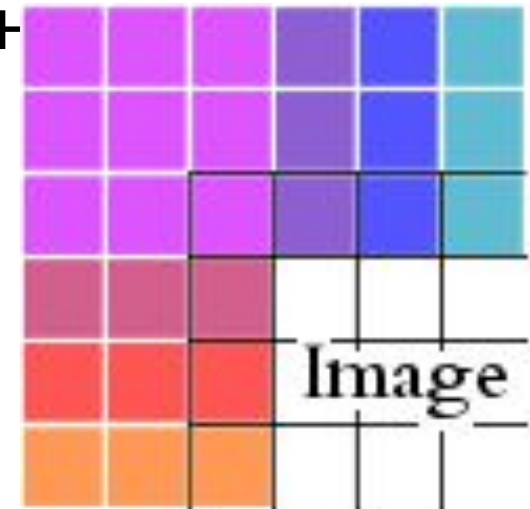
Матричные преобразования изображений

Ядро свертки обычно требует значений от пикселей за пределами изображения.



Решение: *расширение ребер изображения.*

Ближайшие граничные пиксели концептуально расширяются настолько, насколько это необходимо для обеспечения значений для свертки. Угловые пиксели растягиваются в квадратные фигуры. Остальные краевые пиксели растянуты линиями.



Фильтр размытия изображений

- Фильтр для размытия изображения Normalized Box Filter имеет ядро:

$$K = \frac{1}{\text{ksize.width} * \text{ksize.height}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 \\ & & \dots & & & \\ 1 & 1 & 1 & \dots & 1 & 1 \end{bmatrix}$$

Пример его работы для $\text{ksize.width} = 10$, $\text{ksize.height} = 10$:



Фильтры для улучшения резкости изображений

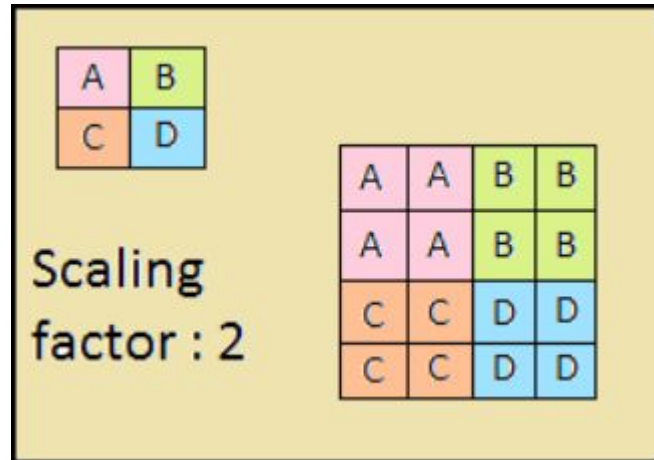
- Следующая матрица определяет коэффициенты цифрового фильтра размером 3 на 3 пикселя, используемого для повышения резкости и

$-k/8$	$-k/8$	$-k/8$
$-k/8$	$k+1$	$-k/8$
$-k/8$	$-k/8$	$-k/8$

- Параметр k определяет степень повышения контраста.
- Пример для $k=8$:



Изменение размеров изображения



Алгоритм resize Nearest Neighbor:

1) Рассчитать ширину и высоту измененного изображения:

- Ширина измененного изображения = ширина \times коэффициент масштабирования по ширине
- Высота измененного изображения = высота \times коэффициент масштабирования по высоте

2) Пройти через каждый пиксель измененного изображения и скопировать ближайшее значение пикселя из исходного изображения.

Ближайшее значение рассчитывается по формуле:

$i \div$ коэффициент масштабирования по высоте,
 $j \div$ коэффициент масштабирования по ширине.

Изображения были взяты из:

- [Растровое изображение, википедия](#)
- [Изображение как двумерный массив данных](#)
- [Глубина цвета](#)
- [Color depth, википедия](#)
- [Converting BMP image to set of instructions for a plotter?](#)
- [Object Detection in Video Frames Using](#)
- [Microarray Spot Finding Example](#)
- [Investigate Bit Depth](#)

Изображения были взяты из:

- [Indexed Color](#)
- [Last time](#)
- [О цветовых пространствах](#)
- [Mark Impex Red Pigment](#)
- [Excelente accidente](#)
- [Pixel](#)
- [Che cosa sono l'RGB e il CMYK?](#)
- [WAT](#)

Изображения были взяты из:

- [Освещение для растений](#)
- [Основы теории цвета. Система CIE XYZ](#)
- [Graphical presentation of RGB 3d matrix](#)
- [ДомиК](#)
- [Интерьер](#)
- [Pixeled fish](#)
- [Waterfall](#)
- [Brightness and contrast](#)

Изображения были взяты из:

- [Blue big wave HD wallpaper](#)
- [Линейное контрастирование изображения](#)
- [Фильтрация зашумленных изображений](#)
- [Матричные фильтры обработки изображений](#)
- [Конволюция](#)
- [Kernel \(image processing\)](#)
- [Макро капли](#)
- [Nearest-Neighbor Method](#)