

Компьютерное зрение

Лекция 1. Введение

Задачи компьютерное зрения, человеческое зрение,
формирование изображение, базовые операции с изображениями

28.05.2020

Руслан Алиев

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group
Vision Memo. No. 100.

July 7, 1966

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system. The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

PROJECT MAC

Artificial Intelligence Group
Vision Memo. No. 100.

July 7, 1966

THE SUMMER VISION PROJECT

Seymour Papert

The summer vision project is an attempt to use our summer workers effectively in the construction of a significant part of a visual system.

The particular task was chosen partly because it can be segmented into sub-problems which will allow individuals to work independently and yet participate in the construction of a system complex enough to be a real landmark in the development of "pattern recognition".

WHEN A USER TAKES A PHOTO,
THE APP SHOULD CHECK WHETHER
THEY'RE IN A NATIONAL PARK...

SURE, EASY GIS LOOKUP.
GIMME A FEW HOURS.

... AND CHECK WHETHER
THE PHOTO IS OF A BIRD.

I'LL NEED A RESEARCH
TEAM AND FIVE YEARS.



IN CS, IT CAN BE HARD TO EXPLAIN
THE DIFFERENCE BETWEEN THE EASY
AND THE VIRTUALLY IMPOSSIBLE.

Компьютерное зрение



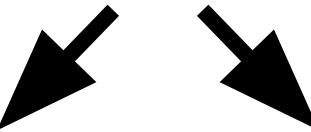
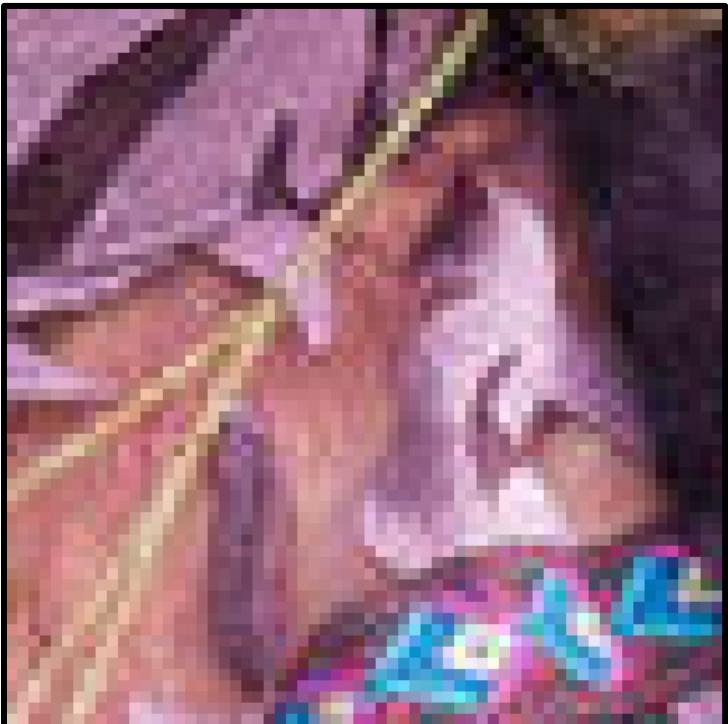
Что видит ВАЛЛИ?



Низкий уровень: изменение размера



Низкий уровень: изменение размера



Низкий уровень: цветовые изменения



Низкий уровень: преобразование в ЧБ



Низкий уровень: яркость



Низкий уровень: изменение насыщенности



Низкий уровень: оттенки



Низкий уровень: выделение краев



Низкий уровень: ориентированные границы



Низкий уровень: ориентированные границы



Низкий уровень: цветовая сегментация



Низкоуровневое компьютерное зрение

Простейшие манипуляции с изображениями

- Размер
- Цвет
- Яркость

Извлечение признаков

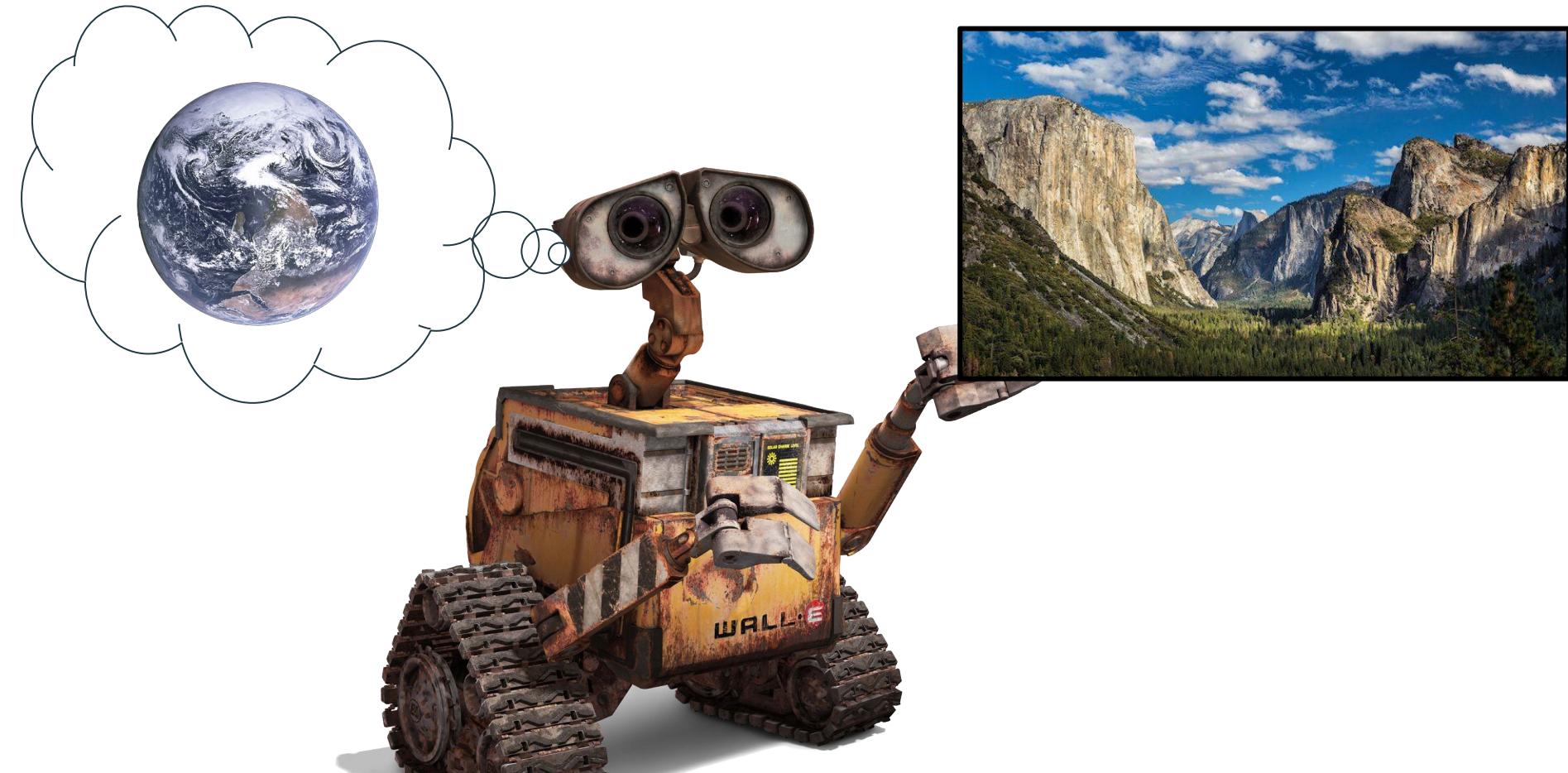
- Границы
- Ориентированные
границы
- Сегментации



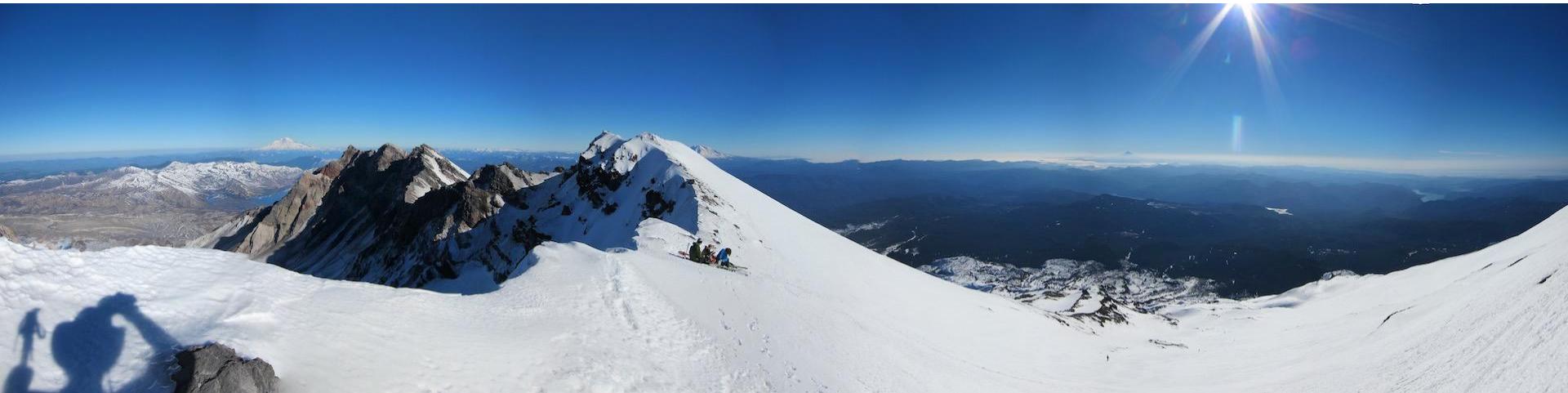
Низкоуровневое компьютерное зрение

Где это может быть полезно?

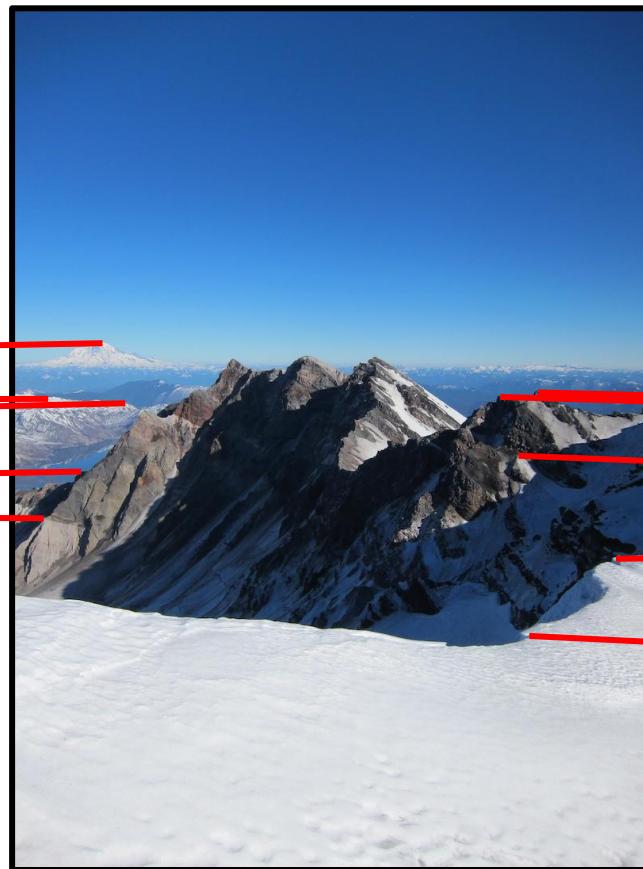
Средний уровень



Средний уровень: панорамы



Средний уровень: панорамы



Средний уровень: панорамы



Средний уровень: стерео изображения



Средний уровень: стерео изображения



Средний уровень: стерео изображения



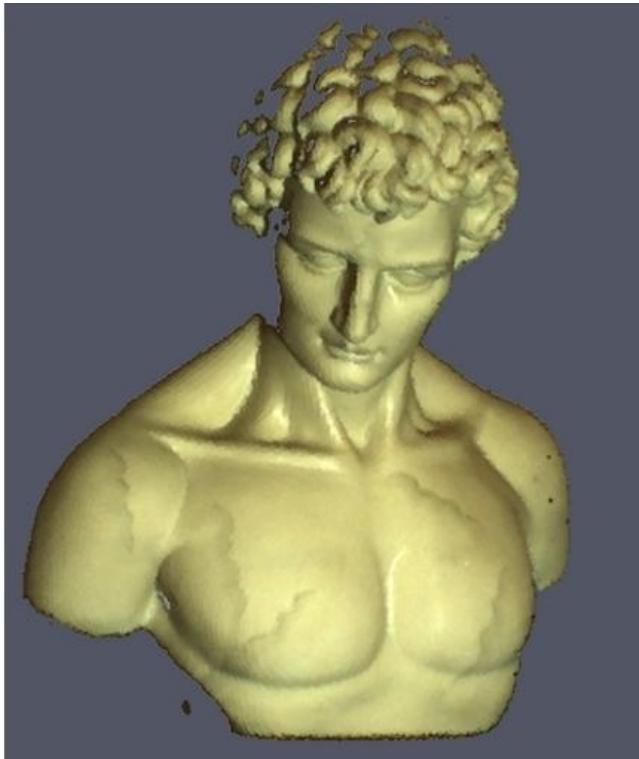
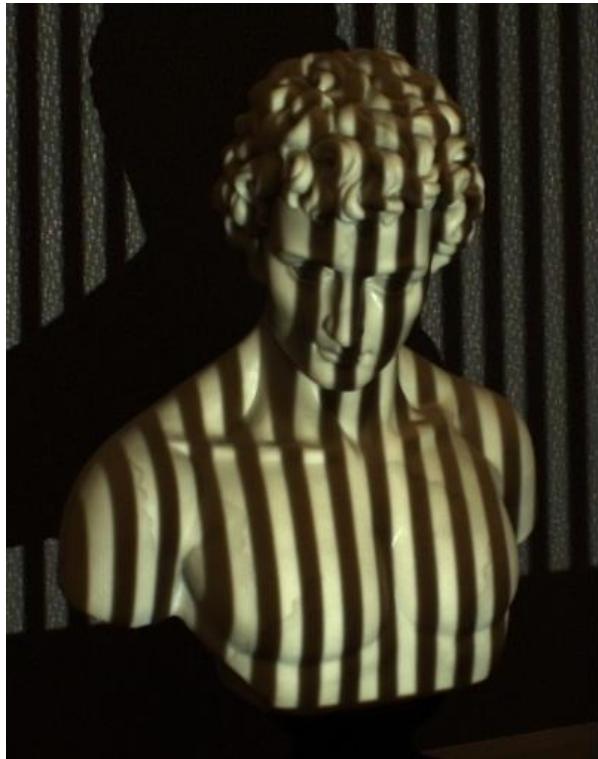
Средний уровень: стерео изображения



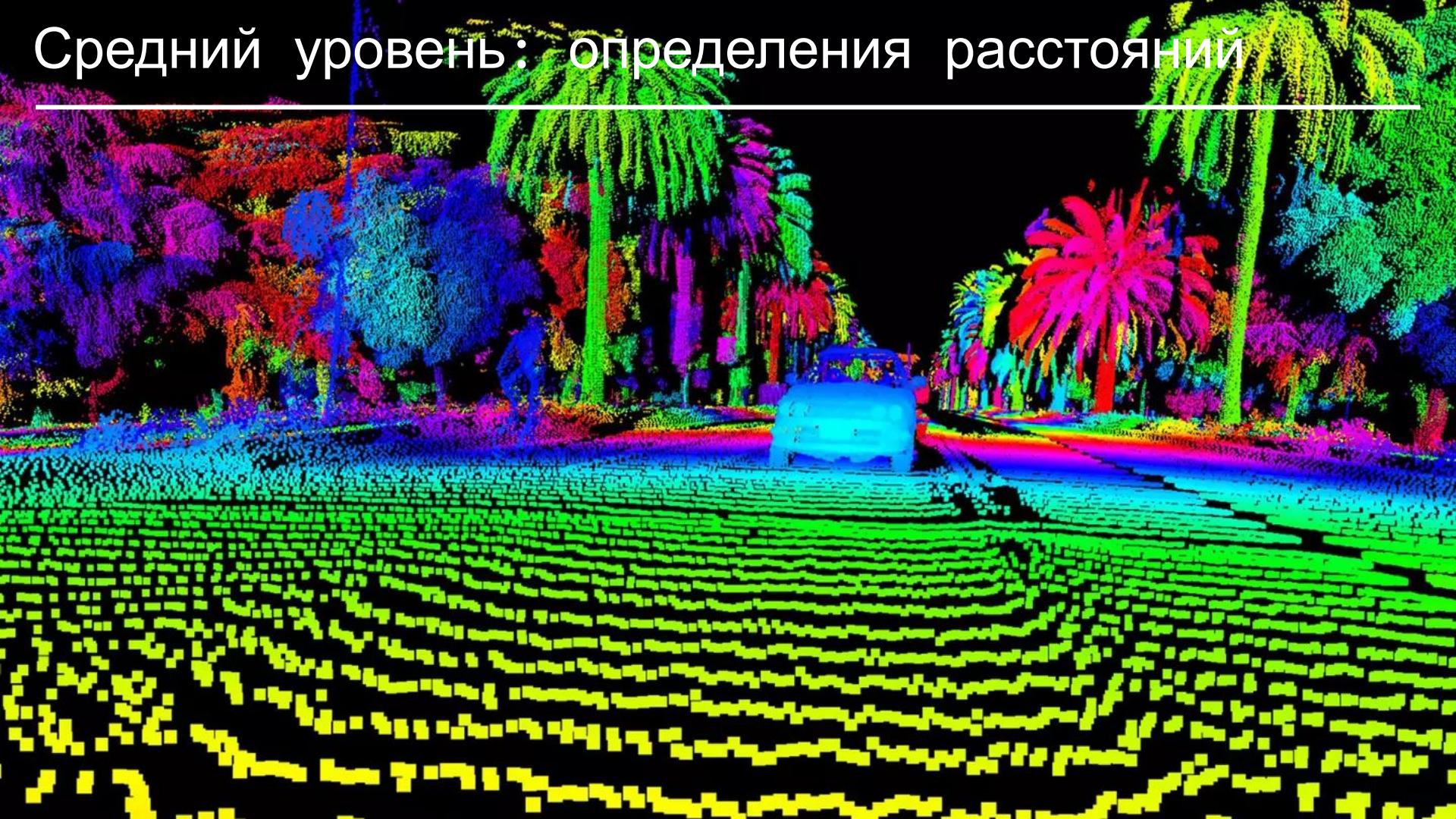
Средний уровень: Structured Light Scan



Средний уровень: Structured Light Scan



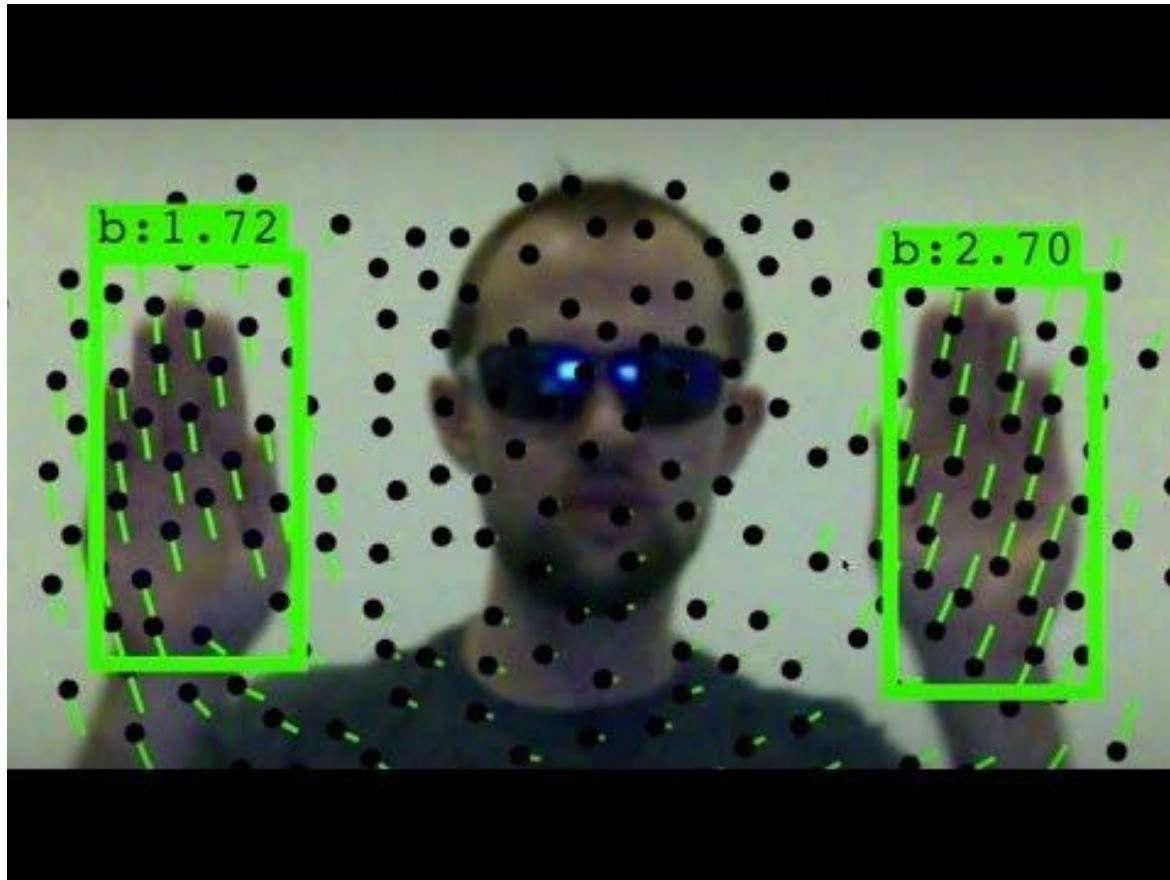
Средний уровень: определения расстояний



Средний уровень: optical flow



Средний уровень: optical flow



Среднеуровневое компьютерное зрение

Картина <-> Картина

- Panoramas

Картина <-> Мир

- Стерео изображения
- Structured light
- Определение
расстояния
(фотограмметрия)

Картина <-> время

- Optical flow



Средний уровень:

Приложения?



Высокий уровень



Высокий уровень: классификация

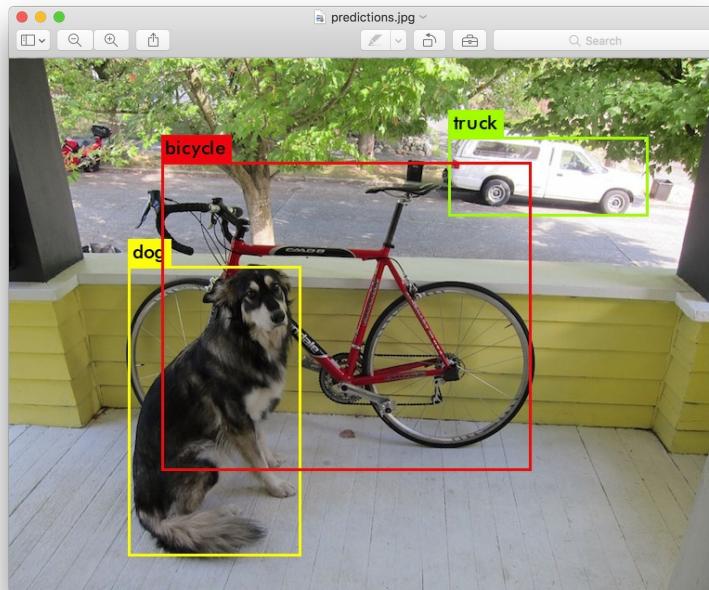
- Ответь, что изображено на картинке

Высокий уровень: тэггинг

- Перечисли ВСЕ предметы на изображении

Высокий уровень: детекция

- Перечисли ВСЕ предметы на изображении
- Покажи, где они находятся



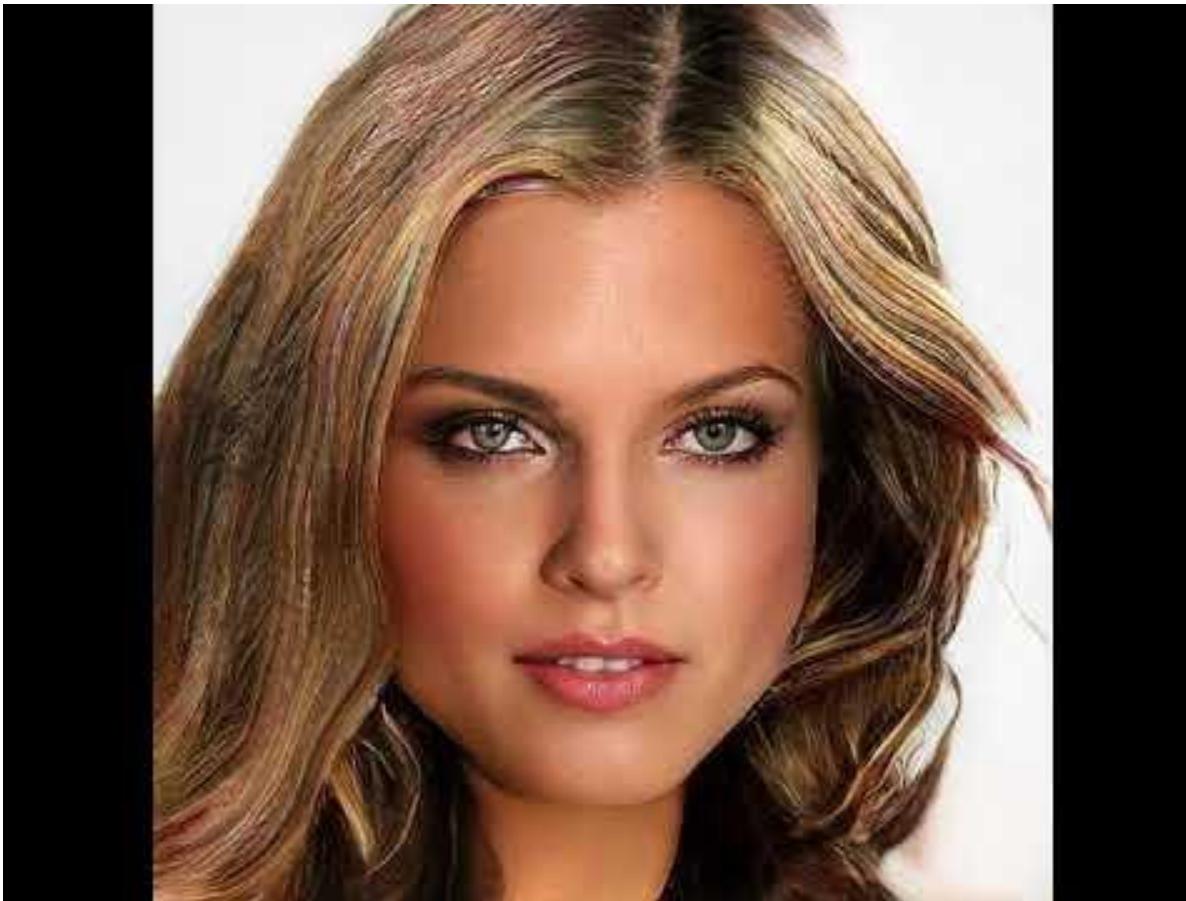
Высокий уровень: семантическая сегментация



Высокий уровень: инстанс сегментация



Высокий уровень: генерация



Высокий уровень

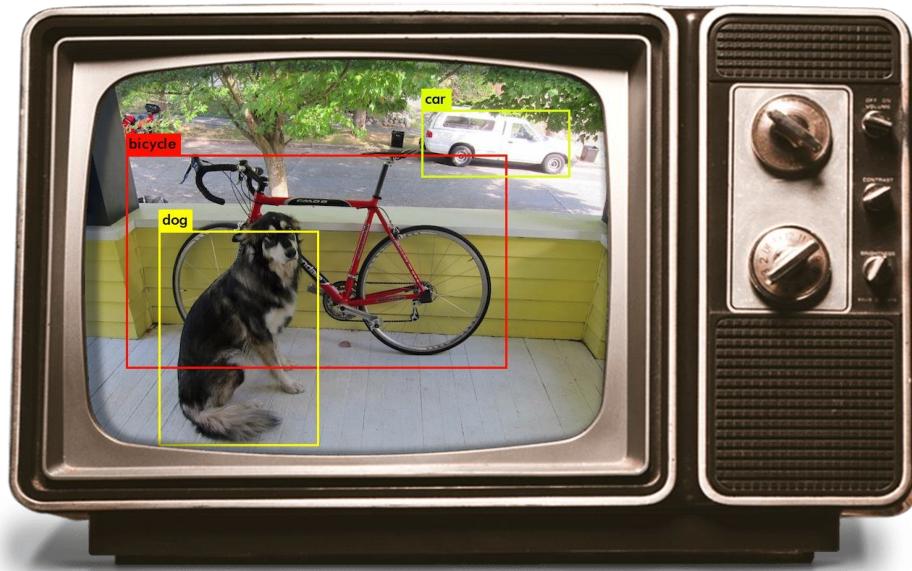
- Single image 3D
- Super-resolution
- Image restoration
- Image captioning
- Image generation
- И многое другое

Высокоуровневое компьютерное зрение

Появляется семантика!

Приложения

- Поиск изображений
- Беспилотные автомобили
- Наблюдение
- Диагностика заболеваний
- ...?



Организационные моменты

Организация курса

Курс разделен на 2 большие части



Классическое компьютерное зрение

- Смена цветовых пространств
- Ресайзинг
- Детекторы границ
- Детекторы ключевых точек
- SIFT
- Создание панорам
- Optical flow
- ...

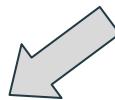


Нейросетевое компьютерное зрение

- CNN
- Segmentation (U-NET)
- Transfer learning
- Autoencoders/variational autoencoders
- GANs
- RNNs for image captioning
- ...

Организация курса

Курс разделен на 2 большие части



Классическое компьютерное зрение

- Смена цветовых пространств
- Ресайзинг
- Детекторы границ
- Детекторы ключевых точек
- SIFT, HoG
- Создание панорам
- Optical flow
- ...



Нейросетевое компьютерное зрение

- CNN
- Segmentation (U-NET)
- Transfer learning
- Autoencoders/variational autoencoders
- GANs
- RNNs for image captioning
- ...



PyTorch

Материалы курса

Лекции, семинары, домашки будут тут ->

<https://github.com/alievrusik/cv-course-hse-summer-2020>

Общаемся в телеграме:

- либо чатик курса
- либо мне в личку (но в пределах разумного)
tg: @ruslan_alive

Пререквизиты

Numpy:

<https://www.tutorialspoint.com/numpy/index.htm>

Линейная алгебра:

<https://www.youtube.com/playlist?list=PL49CF3715CB9EF31D>

Домашки

3 домашки

1 домашка - классическое CV

2,3 домашка - Neural network CV

(нужны будут GPU, работаем в Google Colab)

Домашкадается в виде jupyter notebook

Решенный и заполненный ноутбук нужно отправить мне в TG

Срок - 1-2 недели (в зависимости от сложности домашки)

Дедлайны строгие!

Дополнительные материалы

- Лекции Антона Конушина (ВМК МГУ) по компьютерному зрению:
https://www.youtube.com/playlist?list=PLbwKcm5vdiSYL_yEwQ6JIICBA4dMtHNxo
- CS231n: Convolutional Neural Networks for Visual Recognition
<https://www.youtube.com/watch?v=vT1JzLTH4G4>
- Вдохновение для дипломных проектов
<http://cs231n.stanford.edu/2017/reports.html>
- ods.ai
#cv #deep_learning #article_essence



Как устроено человеческое зрение



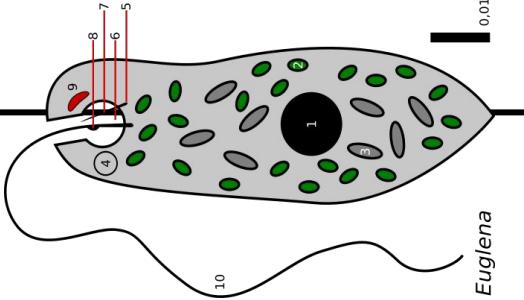
Зачем нужны глаза?

- Чтобы видеть вещи!
- Визуальный стимул - очень важный сигнал
- Появился как фоторецепторный протеин (eyespots)



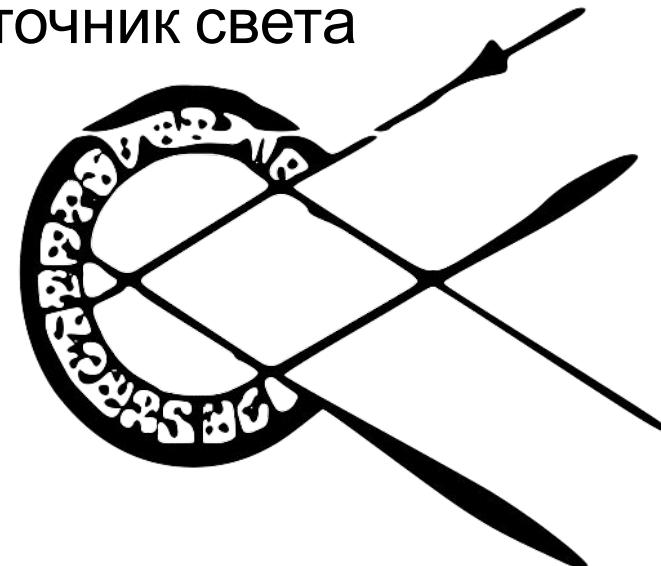
Начало зрения

- Реагировали на окружающий свет
- Нету информации о направлении:
 - Эвглены плывут к свету для лучшего фотосинтеза
 - Улитки отдаляются от света
- Никакой нервной системы, мозга, обработки сигналов
- Очень низкая восприимчивость (свет со всех направлений попадает на одну и ту же область)
- Началась эволюция!

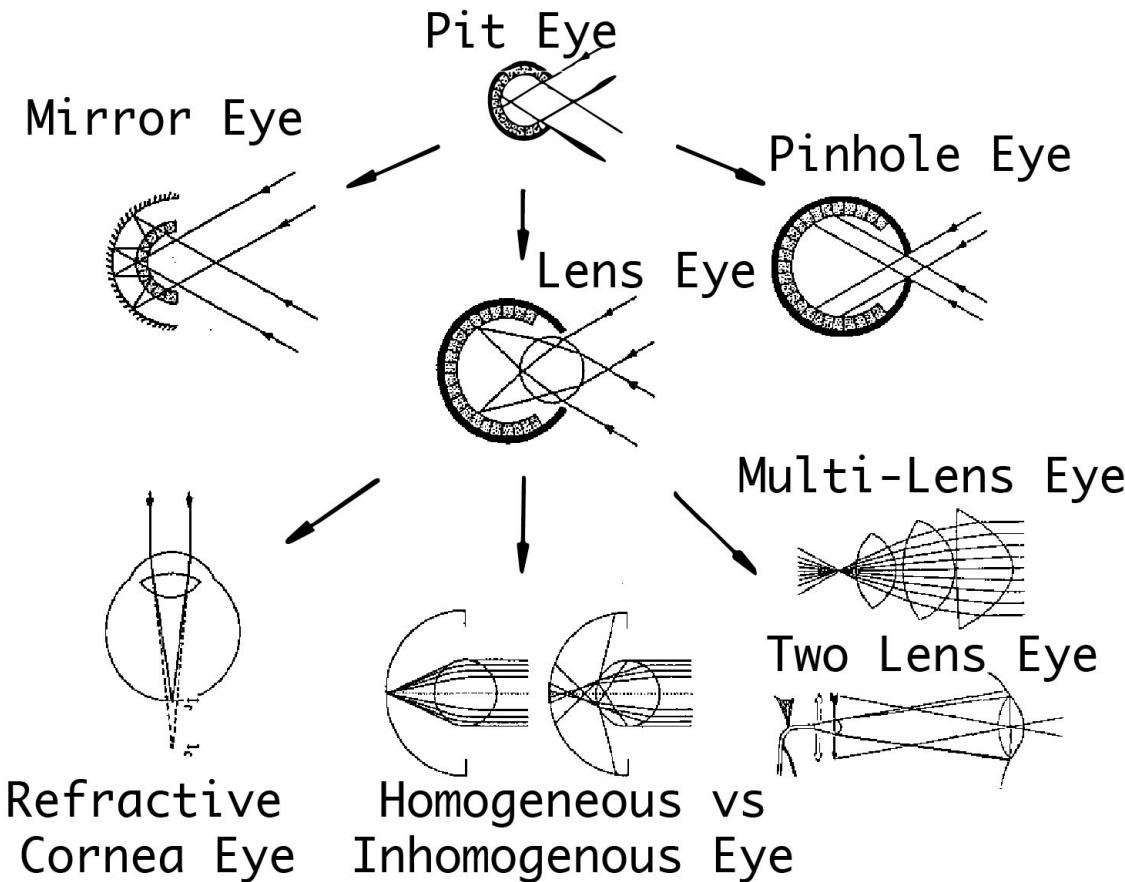


“Ямочки” - первые глаза?

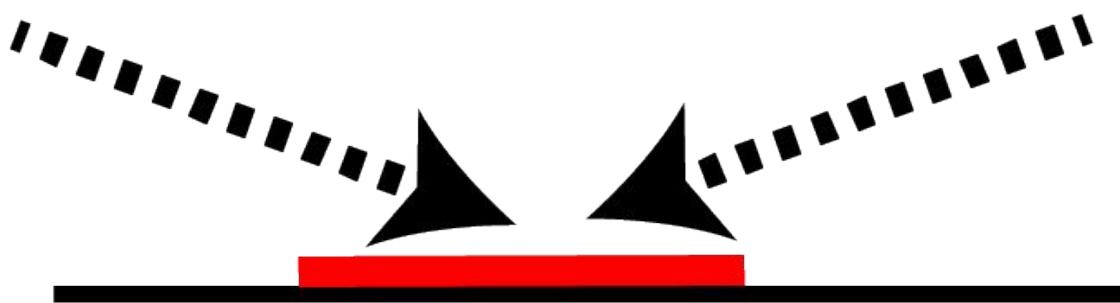
- Фотовосприимчивые клетки расположенные в выпуклости
- Блокируют некоторое количество света
- Больше информации о том, где источник света
- Очень распространено
 - Есть у 28 из 33 классов животных
- Довольно простые,
низкая восприимчивость



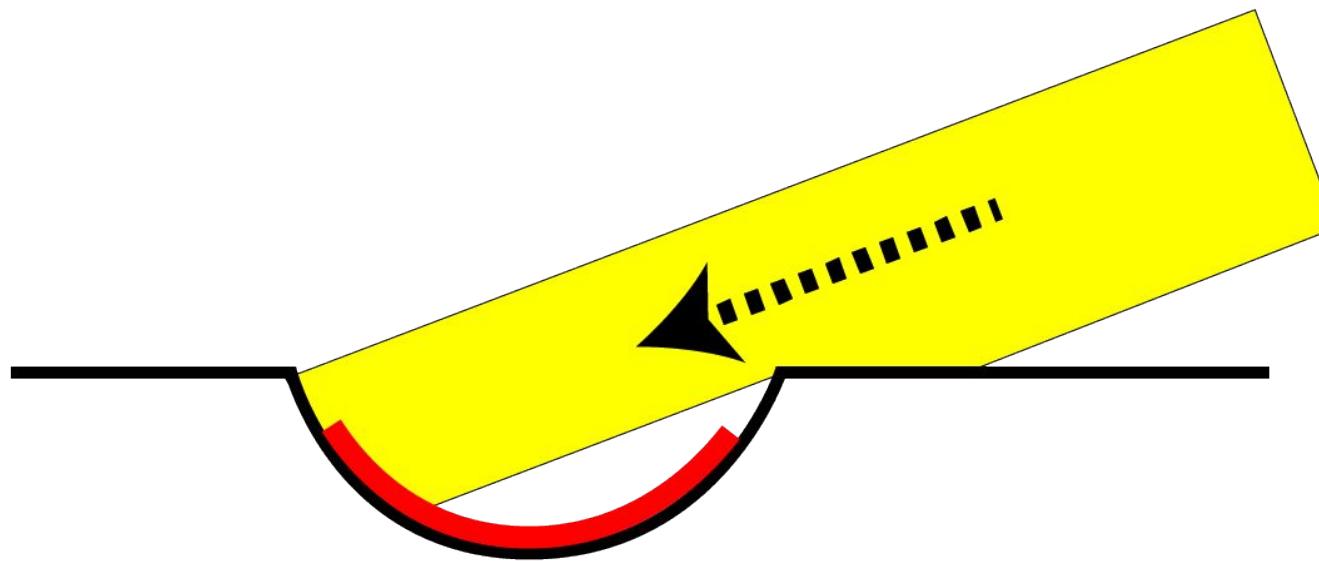
Простое зрение -> сложное зрение



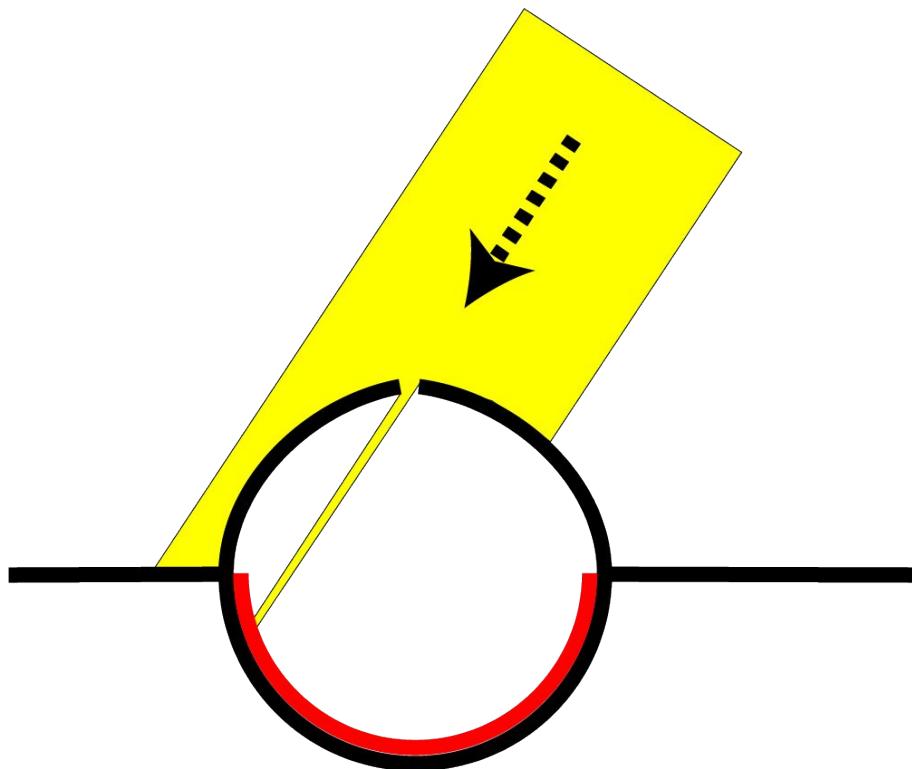
“Глазки” - нету восприимчивости



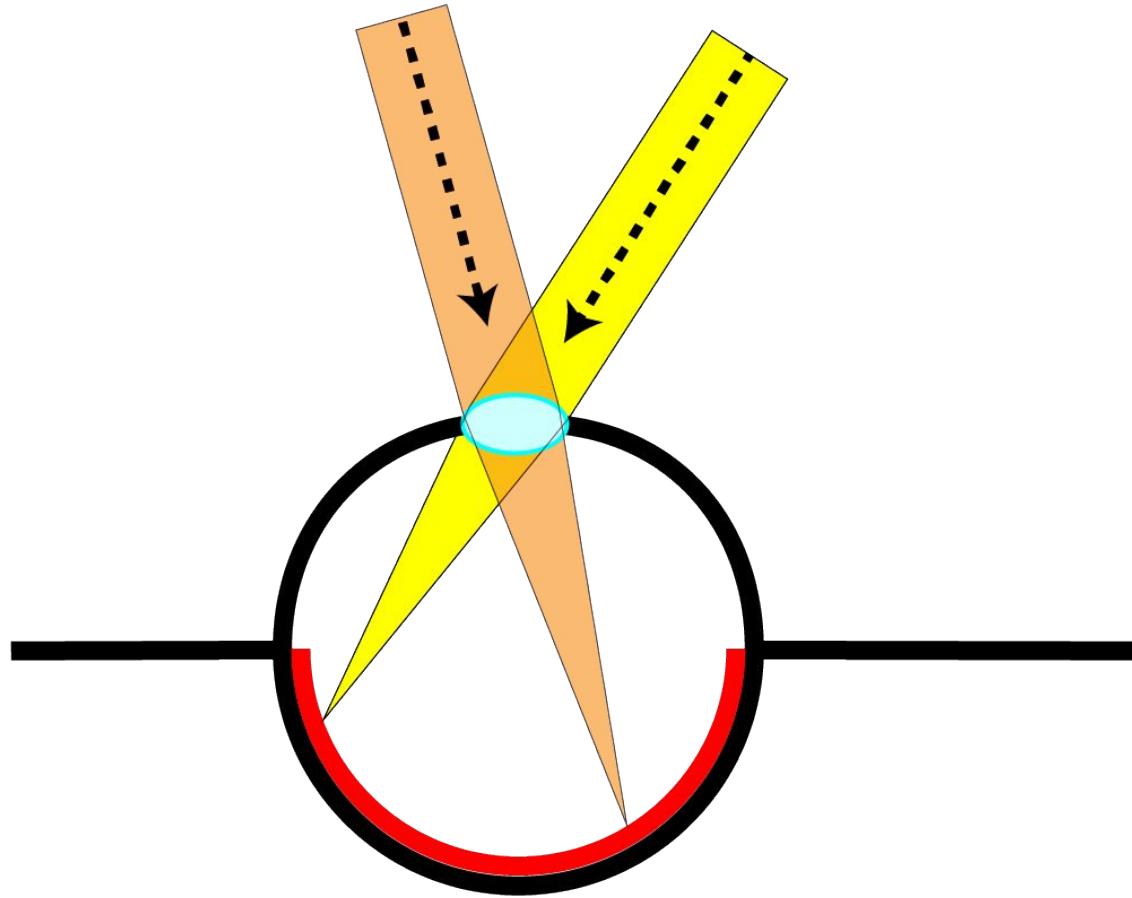
“Ямочки” - появляется восприимчивость



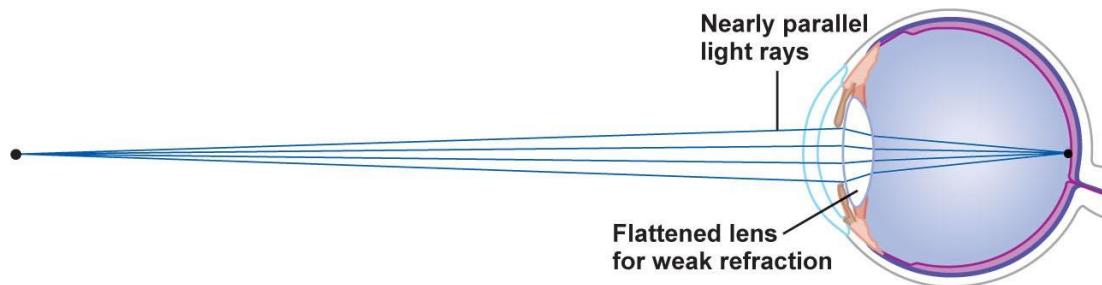
Сложное зрение - большая восприимчивость



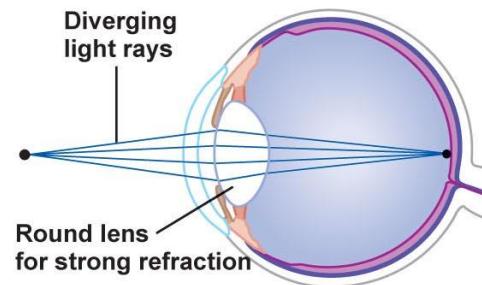
Рефракция: больше света + восприимчивость!



Фокусирование: изменение рефракции



(a) Viewing a distant object



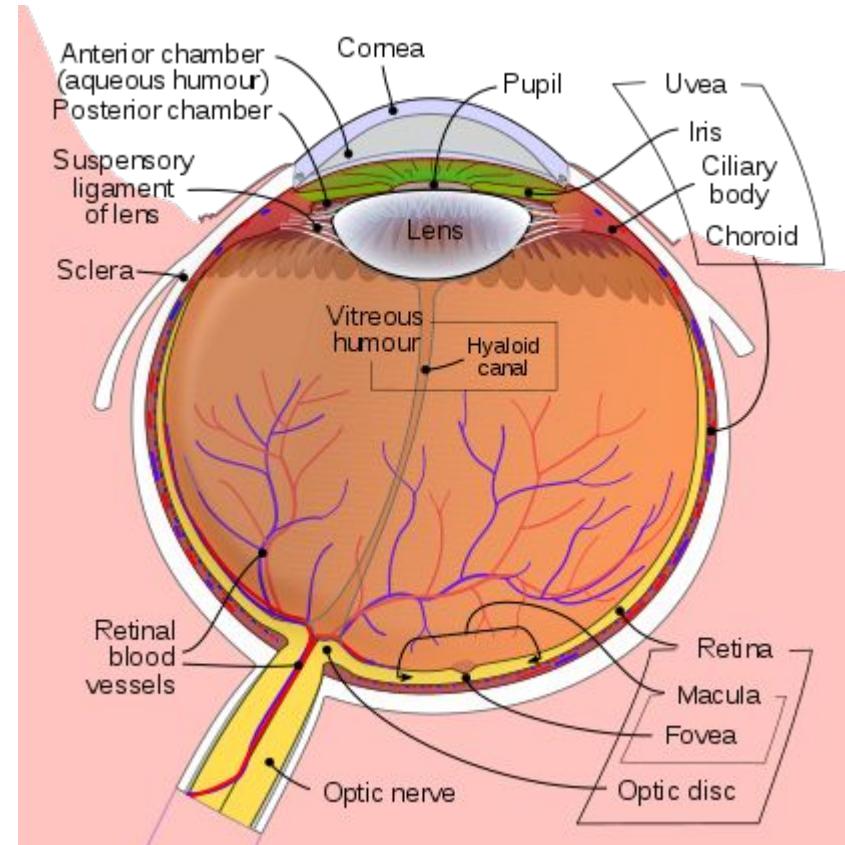
(b) Viewing a near object

Сложные глаза - большое преимущество

- Множество различных механизмов
- Та же цель: улучшить визуальную восприимчивость (разрешение)
- Редкое: у 6 из 33 классов животных
- Дает преимущество: есть у 96% из всех живущих видов
 - Все благодаря глазам??
- Формирование изображение: довольно высокая восприимчивость для обработки форм, объектов, границ и т.д.

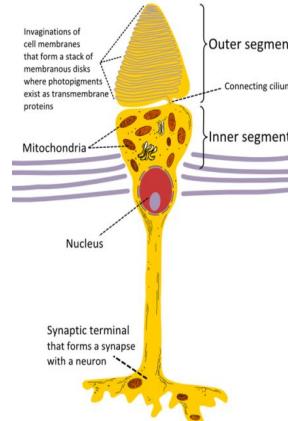
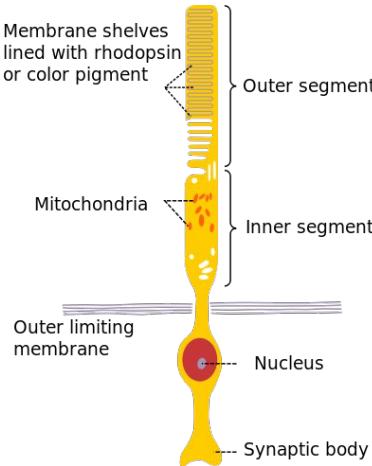
Как работают человеческие глаза?

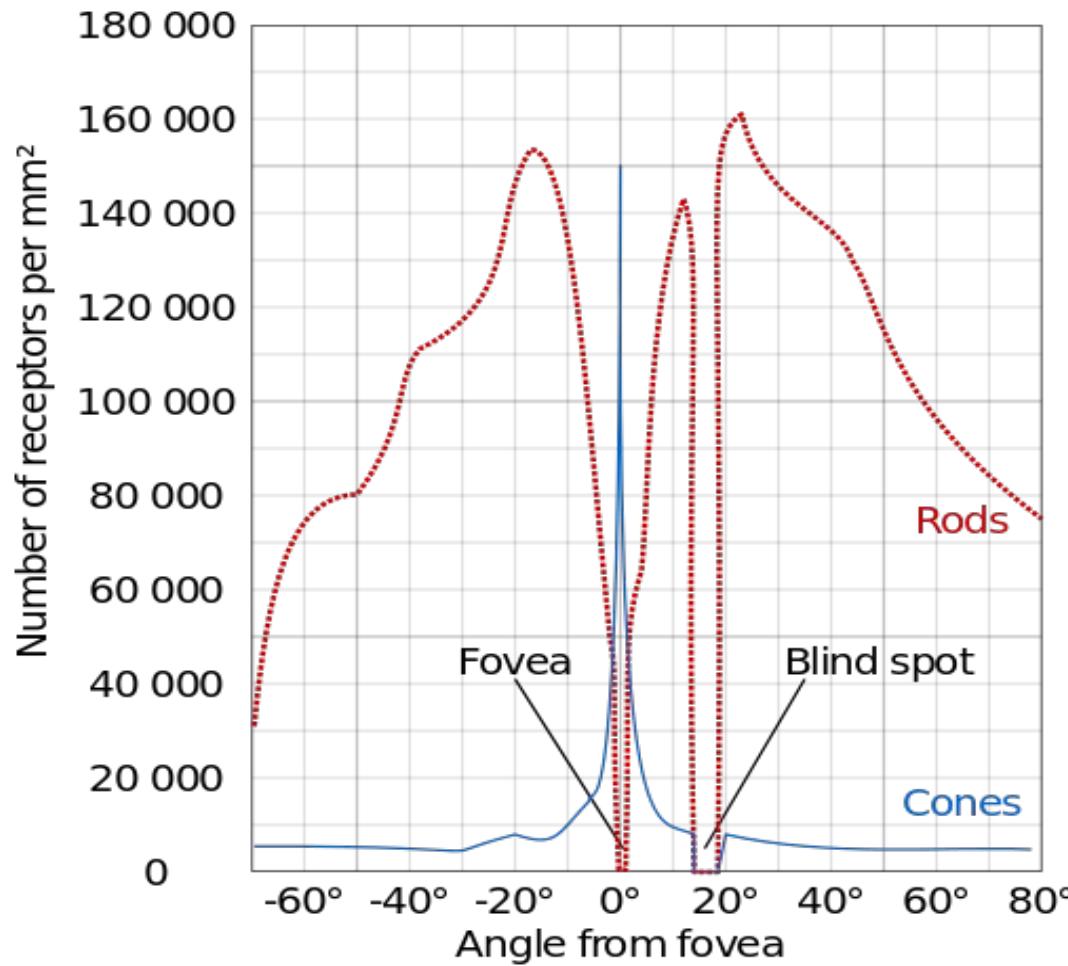
- Очень сложно!
- Свет проходит через
 - Роговицу, зрачок и хрусталик
- Свет падает на сетчатку
- Вызывает реакцию у фоточувствительных клеток
- Информация передается через оптический нерв в зрительную кору



Как работают человеческие глаза?

- Падает на светочувствительные клетки - палочки и колбочки (rods and cones)
- ~100 миллионов на сетчатке
- У всех разные функции, неравномерно распределены





Палочки - малая освещенность, монохромное зрение

- ~92 миллионов
- Чувствительны даже к одному фотону
- Долгое время реагирования
- Работают только в условиях низкой освещенности
- Быстро насыщаются при большом количестве света
- Необходимо около 7 минут для адаптации (ночное зрение)

Колбочки - детализированное, цветное зрение

- ~6 миллионов
- Нужно много фотонов для активации -> яркий свет
- Быстрый отклик
- Хорошая детализированность
- Способны к быстрой адаптации
- Отвественны за дневное зрение
- Большая часть расположена в одной области - центральной ямке (fovea)

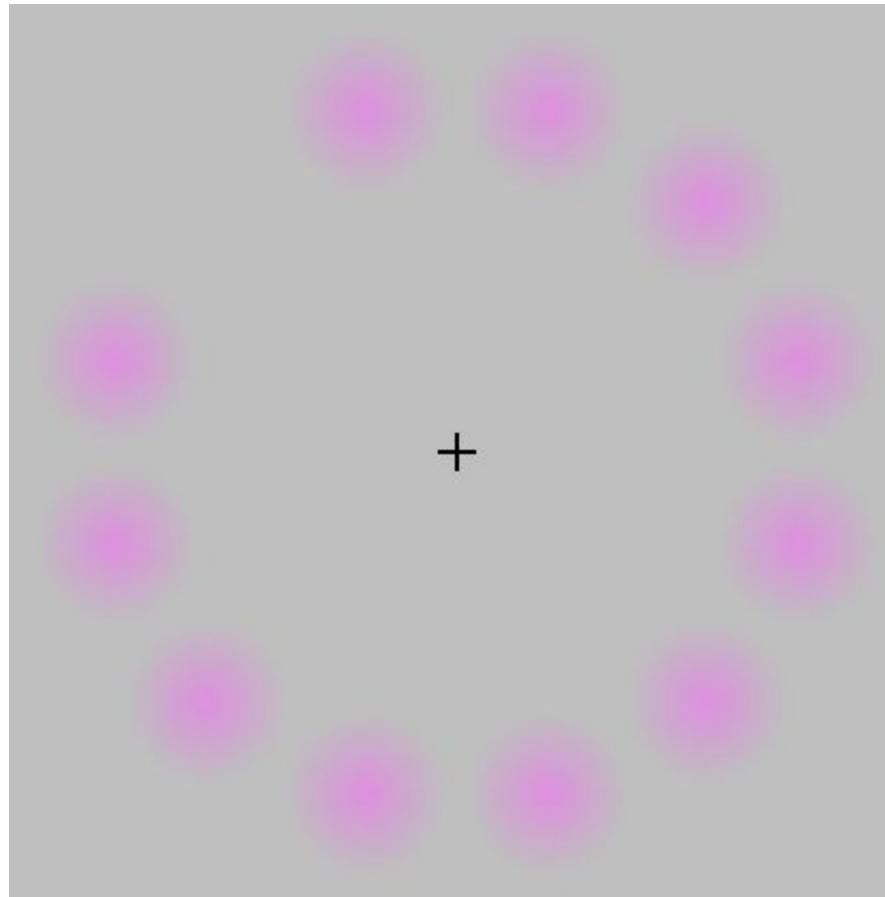
Центральная ямка (fovea)

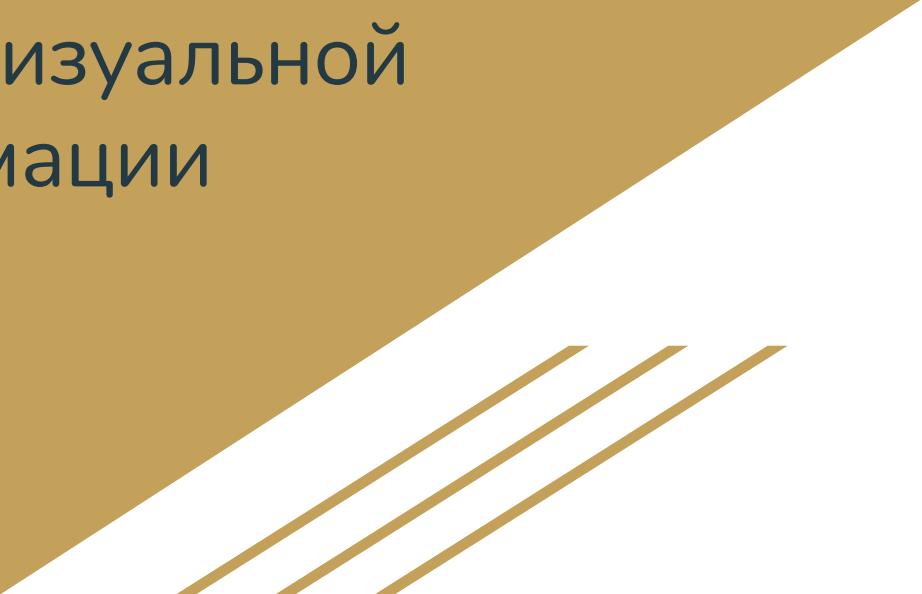
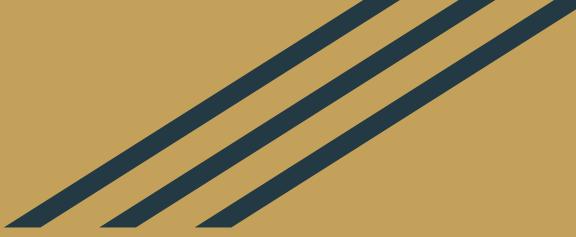
- Маленький кружочек на сетчатке, 1.5 мм
- Высокая плотность колбочек
 - 200,000 колбочек/мм²
- Наибольшая визуальная восприимчивость
- Чтение: мы двигаем глаза, чтоб текст попадал в центральную ямку

Периферийное зрение

- Мало колбочек
 - Низкая восприимчивость
 - Низкое восприятие света
- Много палочек (хорошо ночью)
- Ночью: смотрите не прямо на звезды, а немного вбок от них
- Ярче будет, когда вы не смотрите на звезды!

Фоторецепторы способны адаптироваться





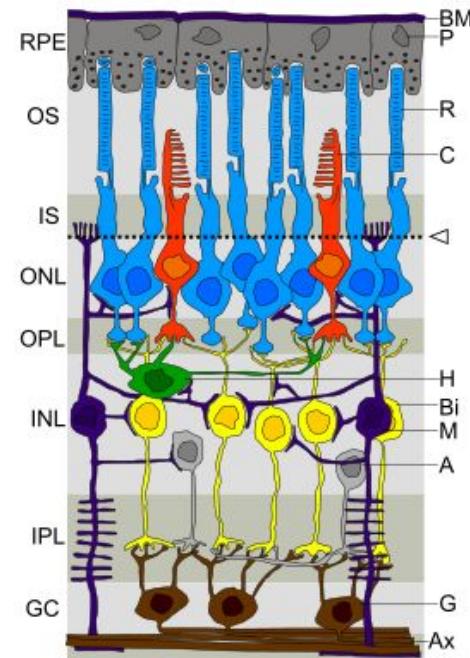
Мозг -
процессор визуальной
информации

Мозг появился благодаря глазам (скорей всего)!

- Некоторые животные напрямую реагируют на визуальный сигнал, не используя мозг
 - Глаза напрямую соединены с мускулами
 - Некоторые медузы
- Нет смысла иметь мозг без сложных входных сигналов
- Мозги и глаза эволюционировали вместе
 - Глаза становились сложнее, визуальная кора расширялась

Ганглионарные клетки передают сигнал в мозг

- ~ 1 миллион
- Различные ганглионарные клетки связаны с различными видами фоторецепторов, реагирующих на различные вещи
 - глубина, движение, ориентация/расположение объектов
 - цвет, форма, тонкие детали

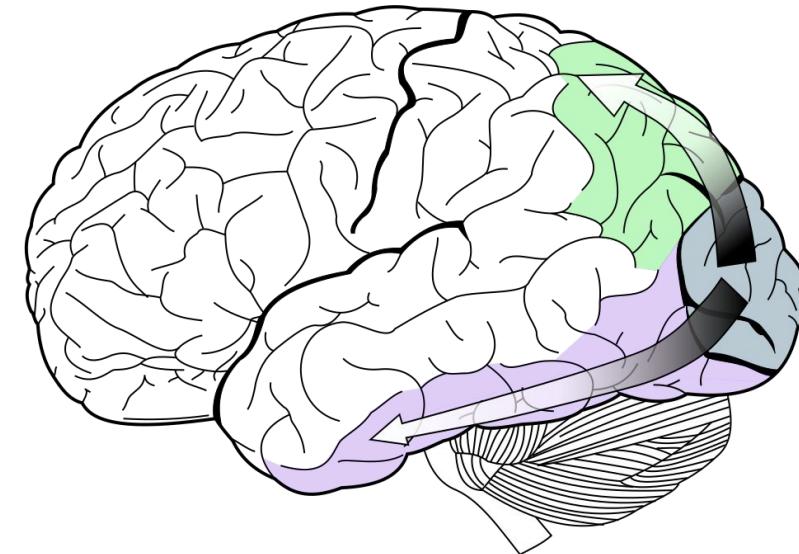


Визуальная кора анализирует сигналы

- Более 30 подотделов мозга для обработки визуальной информации
- V1: первичная визуальная кора
 - Детекция границ
 - Хорошо детектирует первичную пространственную информацию
- V2: вторичная визуальная кора
 - Размер, цвет, форма, возможно память
 - Передает сигналы дальше в V3, V4, V5, ...
 - Даёт фидбэк V1
- Все еще плохо изучено, что происходит

Гипотеза о двух потоках переработки зрительной информации

- Сигналы передаются через V1 и V2
- Сигналы разделяются на 2 потока -
вентральный/дорсальный
- У каждого - своя задача



Вентральный поток VS Дорсальный поток

Factor	Ventral system	Dorsal system
Function	Recognition/identification	Visually guided behaviour
Sensitivity	High spatial frequencies - details	High temporal frequencies - motion
Memory	Long term stored representations	Only very short-term storage
Speed	Relatively slow	Relatively fast
Consciousness	Typically high	Typically low
Frame of reference	Allocentric or object-centered	Egocentric or viewer-centered
Visual input	Mainly foveal or parafoveal	Across retina
Monocular vision	Generally reasonably small effects	Often large effects e.g. motion parallax

Вентральный поток VS Дорсальный поток

ЧТО?

ГДЕ?

Factor	Ventral system	Dorsal system
Function	Recognition/identification	Visually guided behaviour
Sensitivity	High spatial frequencies - details	High temporal frequencies - motion
Memory	Long term stored representations	Only very short-term storage
Speed	Relatively slow	Relatively fast
Consciousness	Typically high	Typically low
Frame of reference	Allocentric or object-centered	Egocentric or viewer-centered
Visual input	Mainly foveal or parafoveal	Across retina
Monocular vision	Generally reasonably small effects	Often large effects e.g. motion parallax

Что это означает?

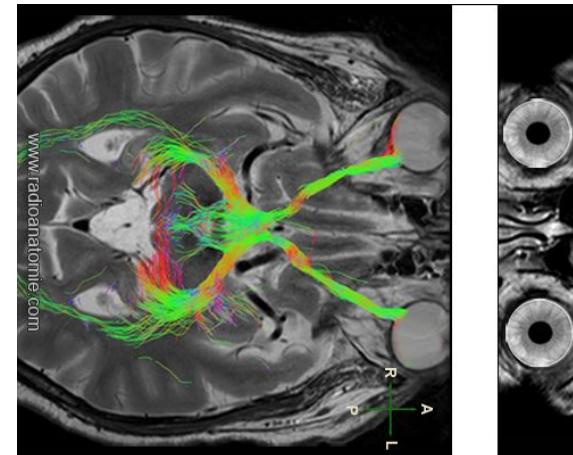
- Разпознавание и действие разделены!
- Повреждена дорсальная система:
 - Может распознавать объекты
 - Плохой визуальный контроль для заданий вроде “возьми яблоко”
- Повреждена вентральная система:
 - Не может распознавать объекты
 - Может с ними что-то делать (хватать, бросать, нести, огибать)
- Большая часть информации в дорсальной системе неосознаваема

Ориентация без зрения



Мозг и зрение

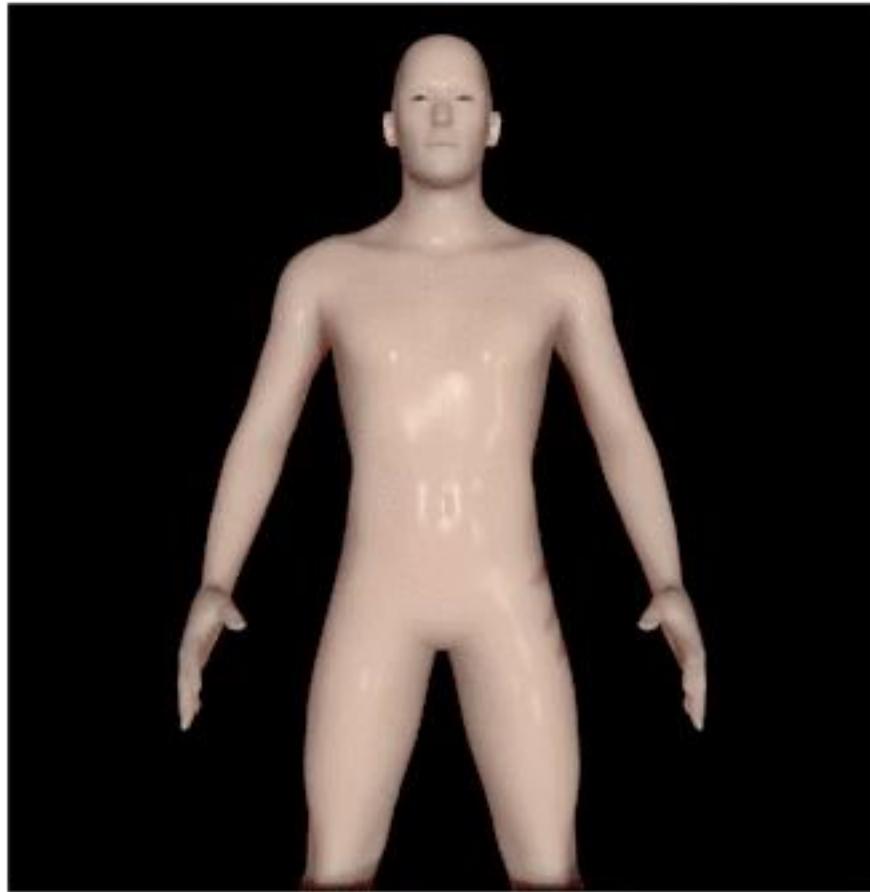
- Большая часть энергия тратиться на зрение:
- Визуальная кора - самая большая система в мозге:
 - 30% церебральной коры
 - $\frac{2}{3}$ электрической активности
- Большая часть обработки происходит бессознательно



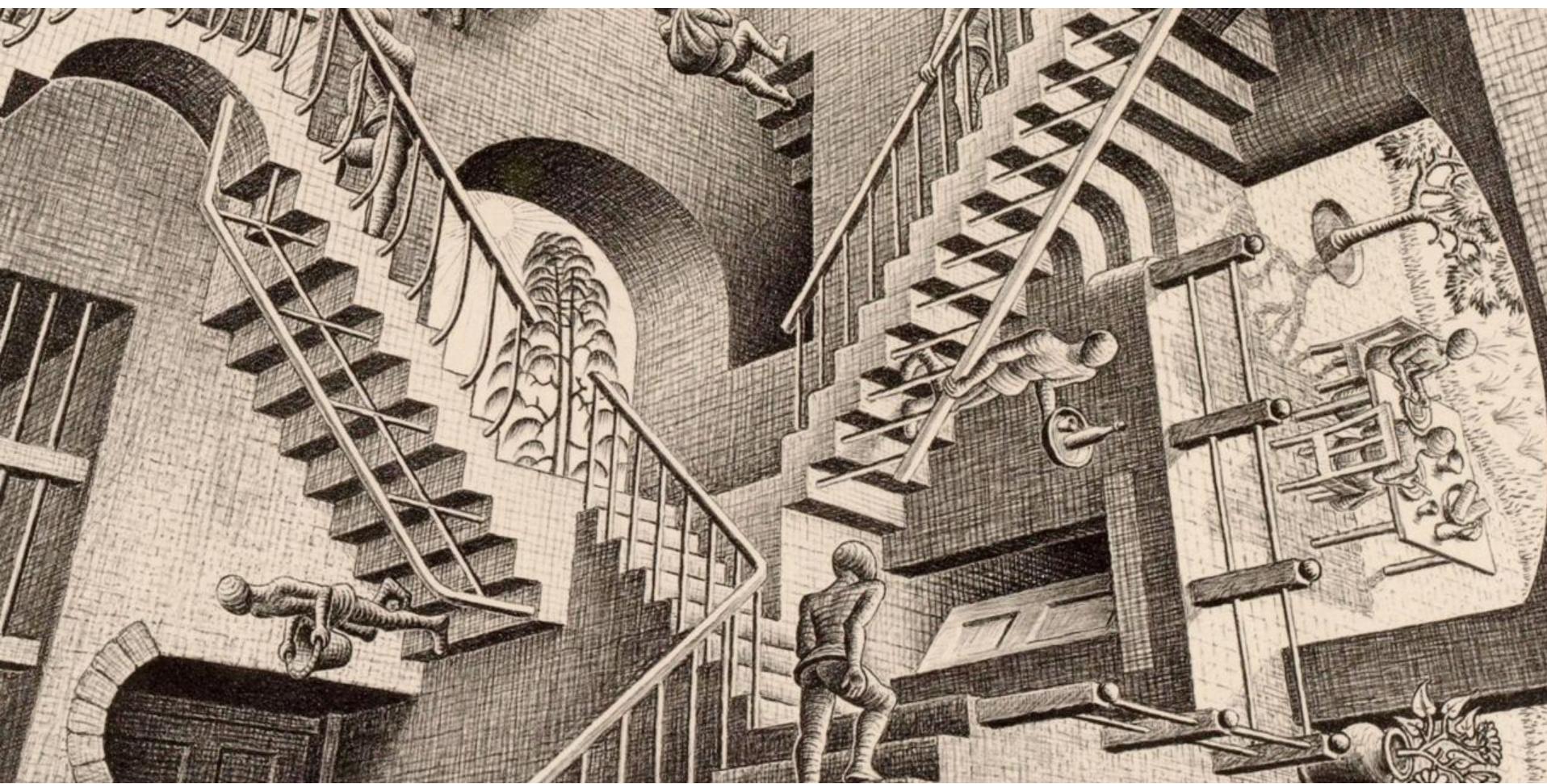
Как мозг видит 3d

- Один глаз:
 - Фокус - насколько хрусталик должен деформировать чтобы объект был виден четче
 - Размытие - размытые объекты находятся на различном расстоянии
 - Параллакс - наблюдатель или объект двигаются -> чем дальше, тем меньше двигается
- Два глаза:
 - Стерео - изображения для левого и правого глаза, немножко различны
 - Сходимость - направлению “линий исходящих из глаз”
- Мозг:
 - Кинетическая глубина: распознавание 3д формы движущегося объекта
 - Окклузия - объекты на переднем плане - ближе
 - Память - мы знаем, размер машины или кошки
 - Тени - светотень дает подсказки по поводу 3д размера



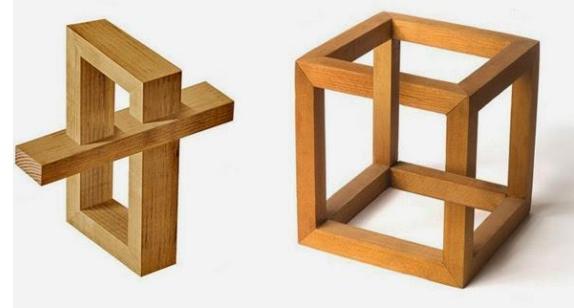






Вопросов больше, чем ответов

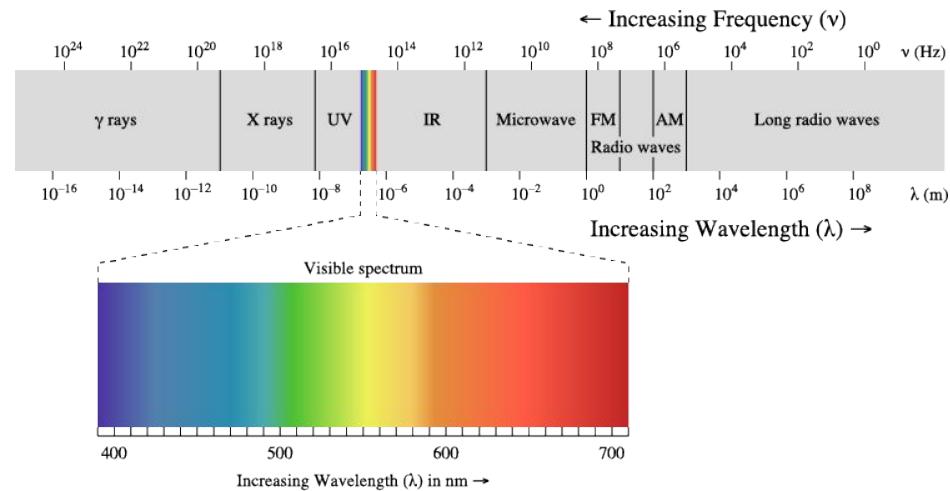
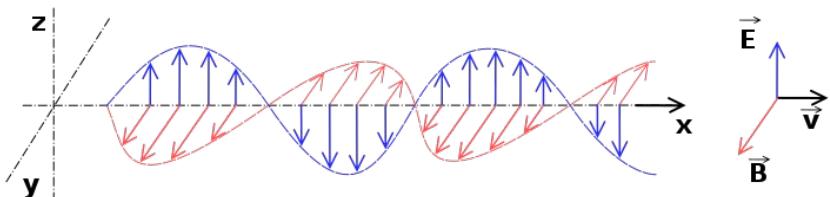
- Визуальная кора - хорошо изученная область мозга
- Только приблизительные идеи, за что отвечают различные компоненты
- До сих пор в зрении появляются новые открытия:
 - Мы моргаем, чтобы глаза могли “бросить” свою позицию в пространстве
 - Визуальная кора может принимать некоторые “высокоуровневые” решения



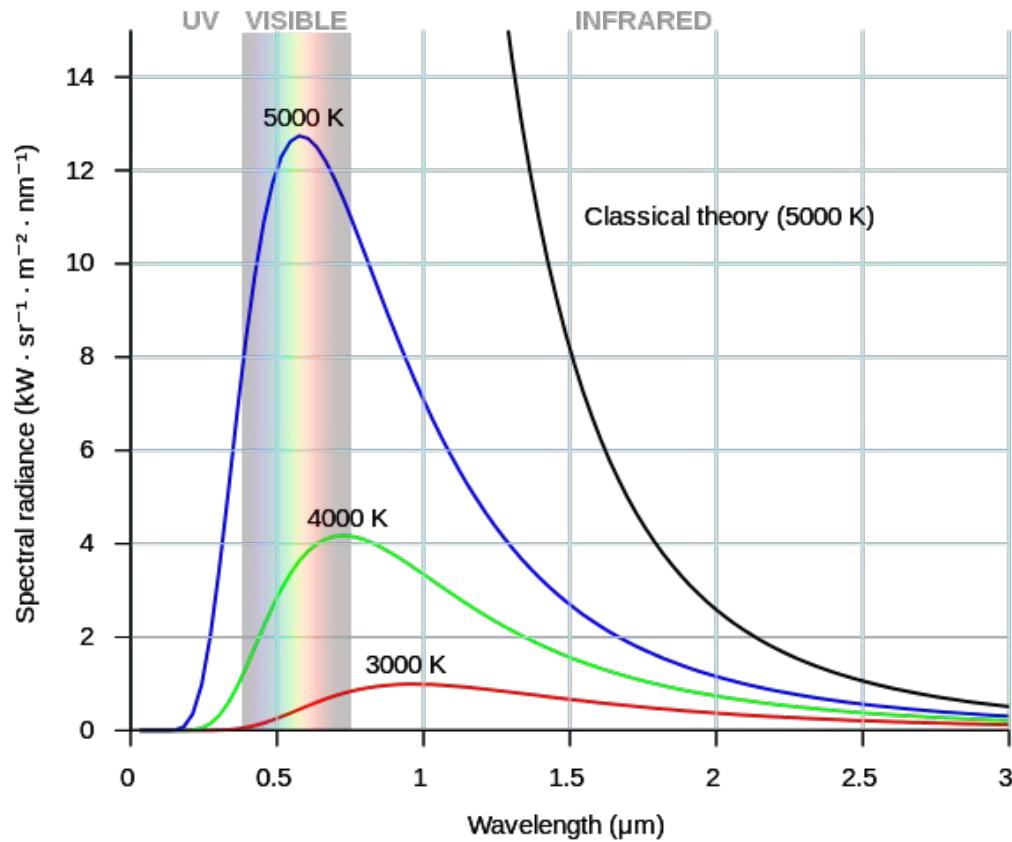
Физика света

Но что же мы видим?

- Электромагнитное излучение
 - Свет? Частица?
- Фотон - элементарная единица света
- Видимый свет: ~400-700 нанометров
- Почему именно этот диапазон?

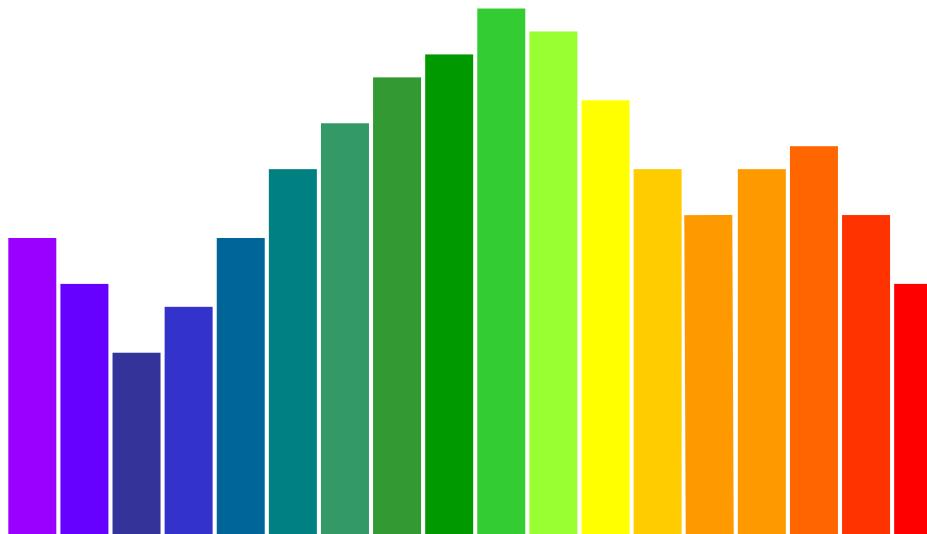


Видимый свет и Солнце

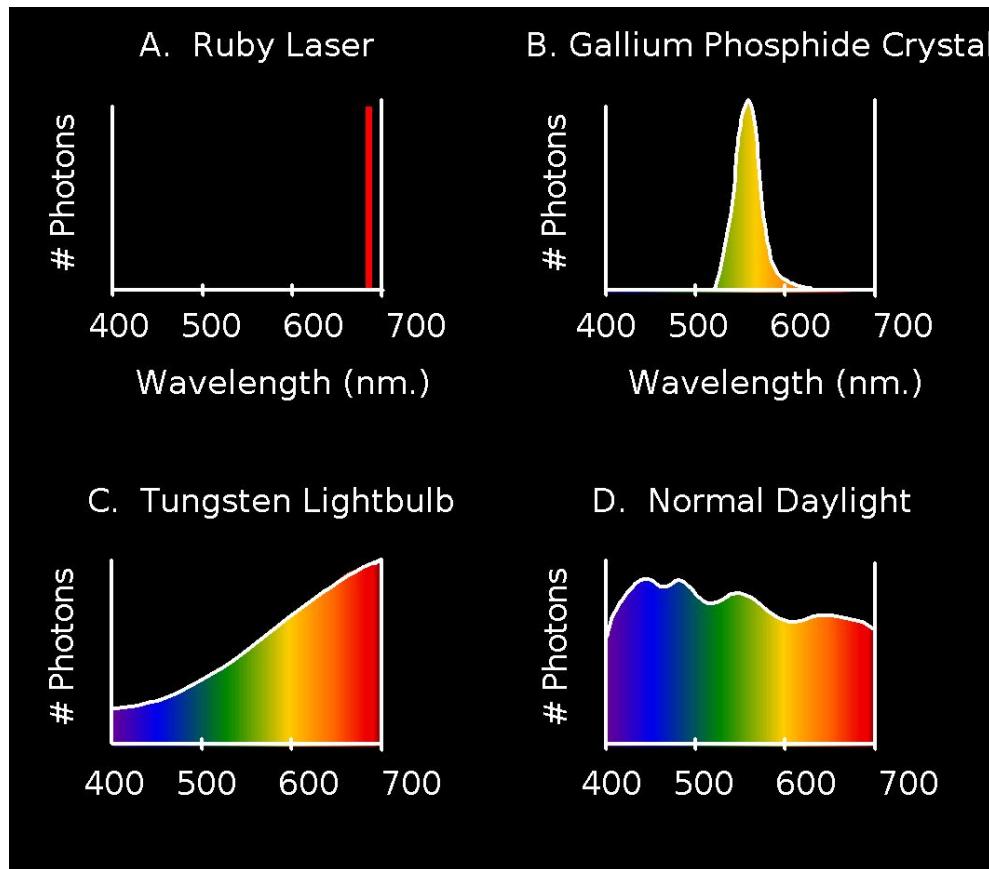


Свет - комбинация волн различной частоты

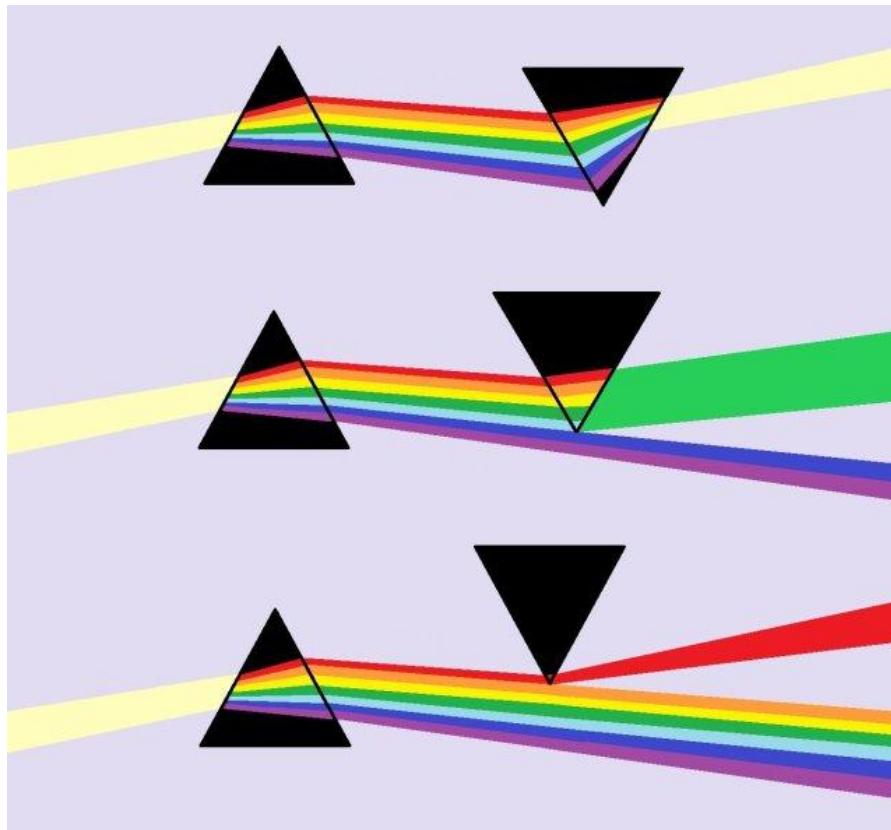
- Как аккорд в музыке
- Свет может быть описана как сумма его частей
- Надо просто разложить его на части



Различные источники света

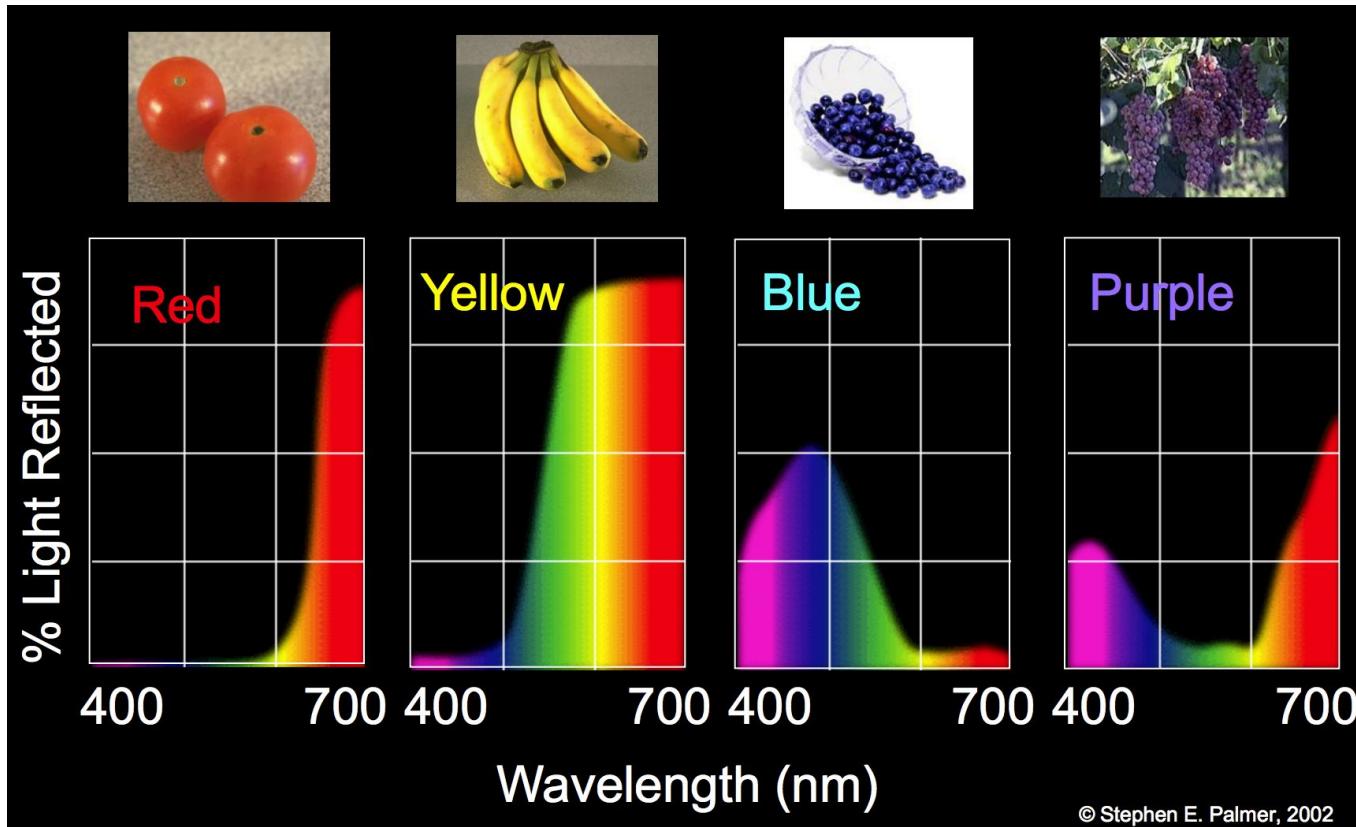


“Белый” свет содержит весь диапазон



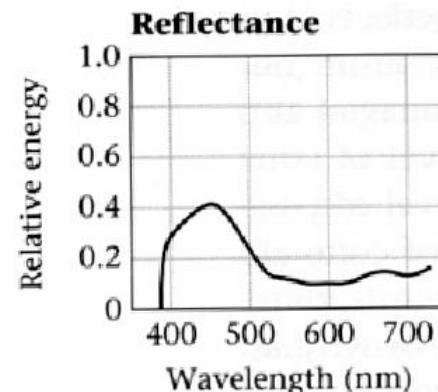
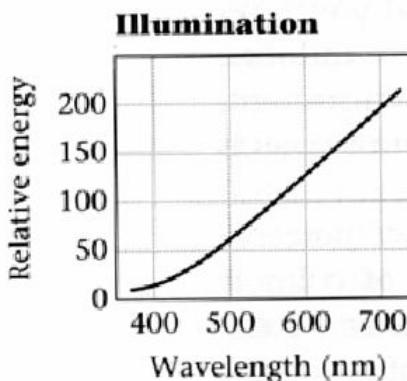
We know this thanks to Newton!

Объекты отражают лишь часть света

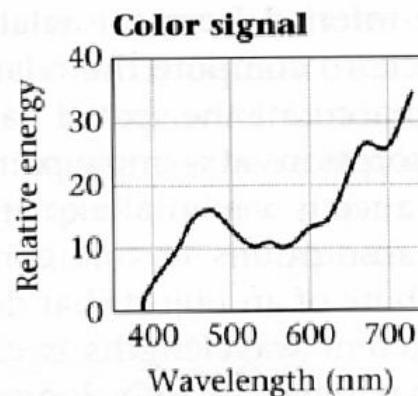


Какого цвета объекта?

- “Цвет” объекта зависит от падающего света и отражающих свойств объекта



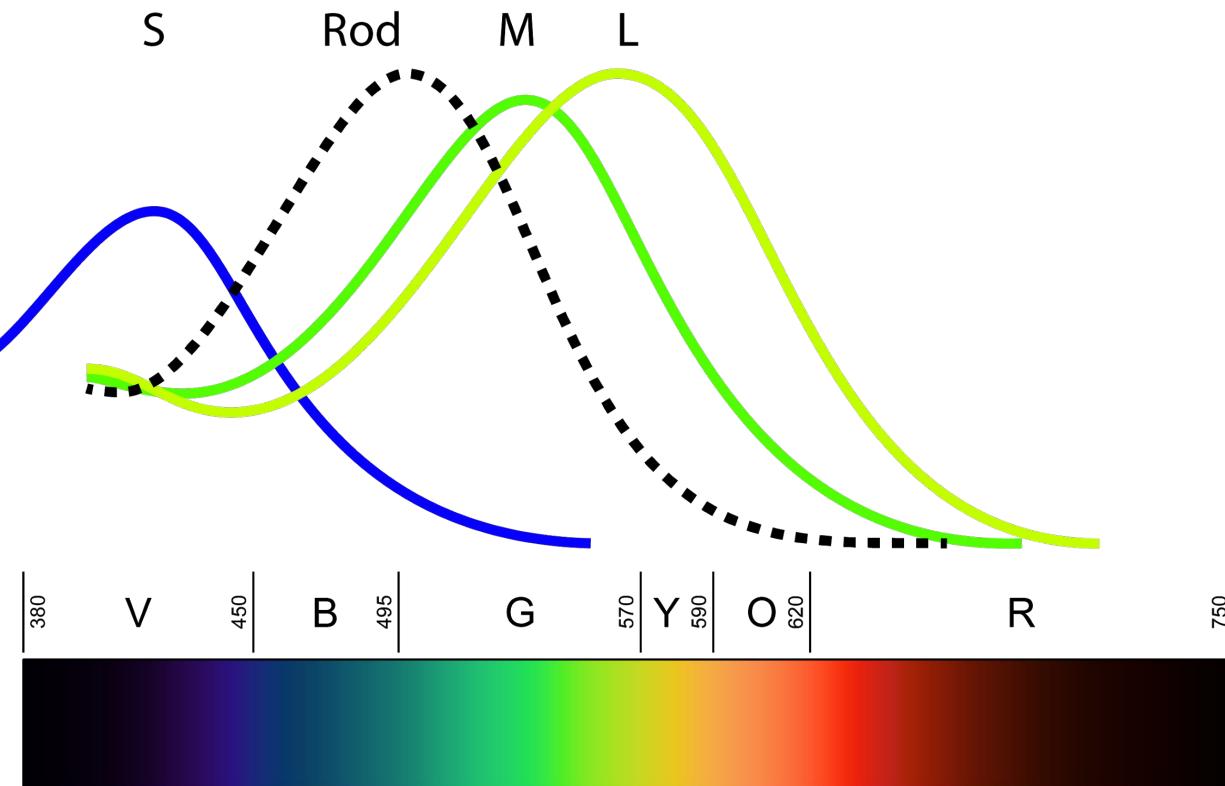
=



Фоторецепторы и свет

- Каждый рецептор по разному реагируют на свет (кривая отклика)
- Каждый “специализируется” на своем цвете
- Палочки - пик около 500 нм
- Колбочки: 3 типа
 - Короткие: пик около 420 нм
 - Средние: пик около 530 нм
 - Длинные: пик около 560 нм

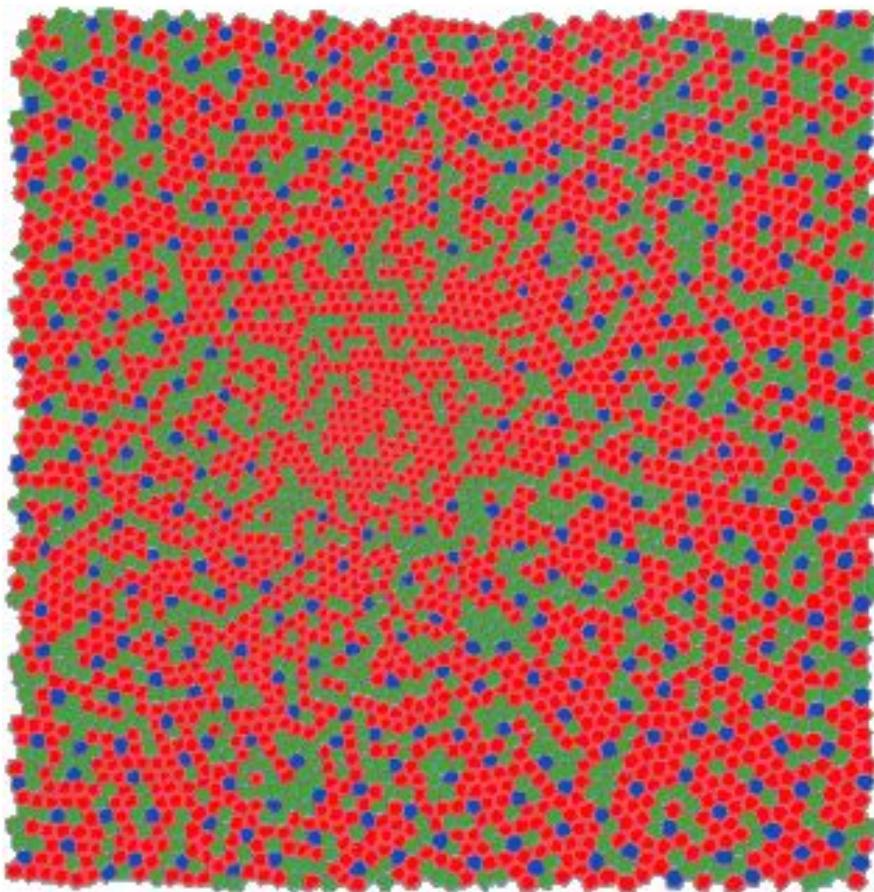
Фоторецепторы и свет



Колбочки и свет

- Мы воспринимаем свет благодаря колбочкам
- В зависимости от длины волны свет стимулирует различные сигналы
- Каждая колбочка выдает один выходной сигнал
- Чтобы посчитать:
 - Умножить входную кривую света на кривую отклика
 - Посчитать площадь под кривой
- Цвет который мы видим - активация 3 различных типов колбочек

Колбочки неодинаково распределены



Множество вариаций,

State	Types of cone cells	Approx. number of colors perceived	Carriers
Monochromacy	1	100	marine mammals, owl monkey, Australian sea lion, achromat primates
Dichromacy	2	10,000	most terrestrial non-primate mammals, color blind primates
Trichromacy	3	1 million	most primates, especially great apes (such as humans), marsupials, some insects (such as honeybees)
Tetrachromacy	4	100 million	most reptiles, amphibians, birds and insects, rarely humans
Pentachromacy	5	10 billion	some insects (specific species of butterflies), some birds (pigeons for instance)

Цвет - это наше восприятие световых волн

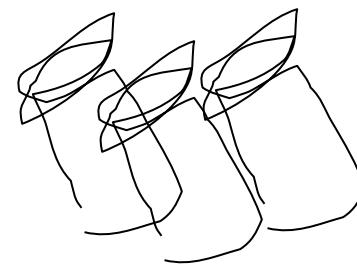
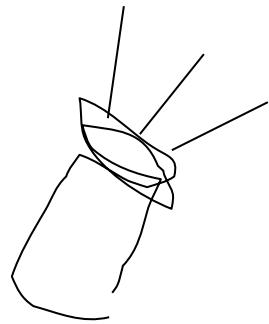
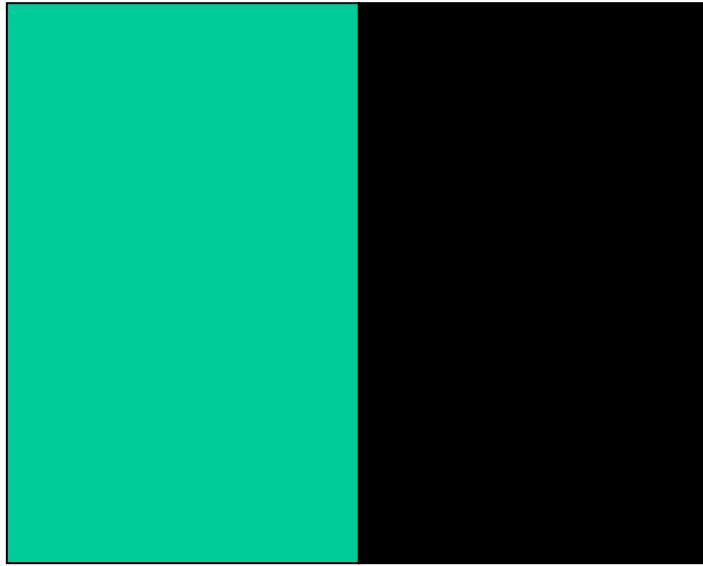
- Цвет - не сама волна
- Цвет - психологическое понятие
- У нас только три типа колбочек, свет характеризуются только 3 выходными сигналами
- Многие световые волны получаются одинаковыми: метамеры
- Является ли этой проблемой??

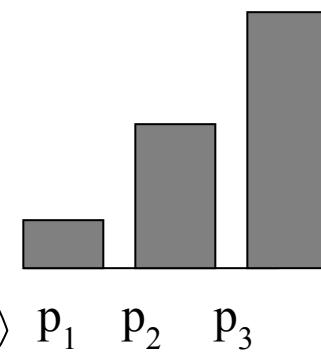
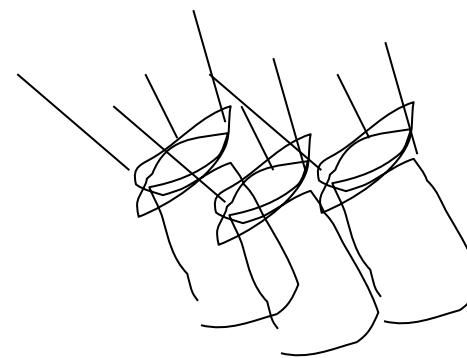
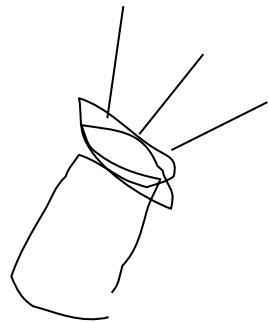
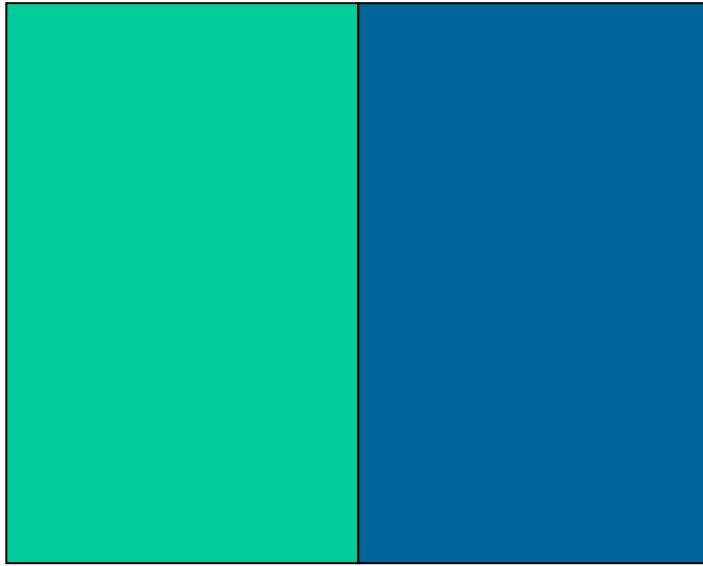
Метамеры это классно!

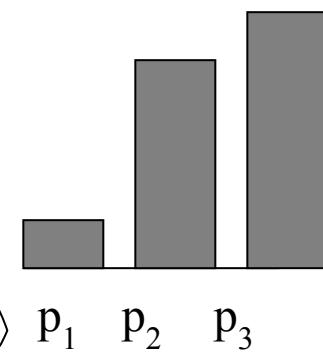
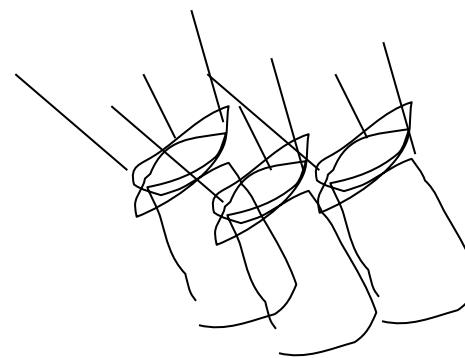
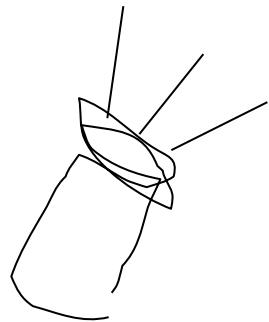
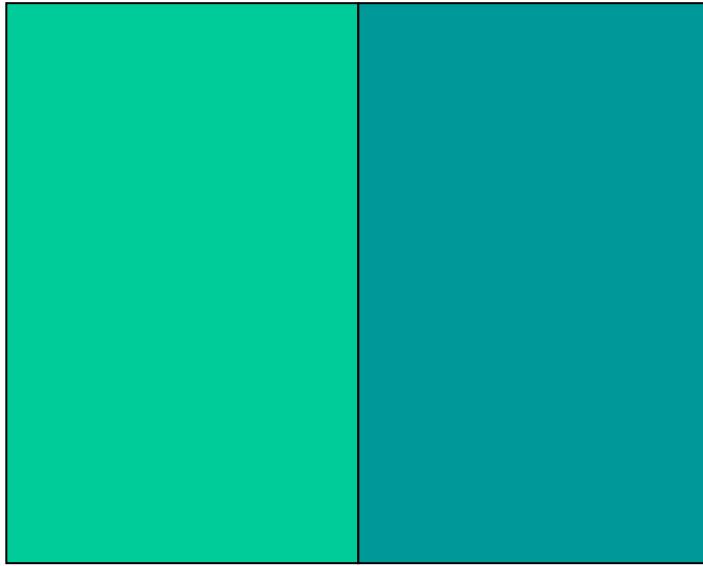
- Представьте мы бы идеально обрабатывали световые волны
 - Чтобы воспроизвести свет, нужно воспроизвести волну
 - Очень сложно было бы сделать телевизор
 - Цветные принтеры требовали тысячи видов чернил
- Но этого не нужно, ведь существуют метамеры!
 - Можно воспроизвести множество цветов имитируя необходимую волну

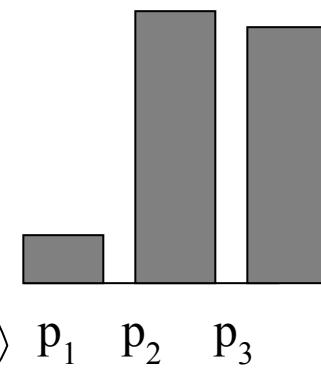
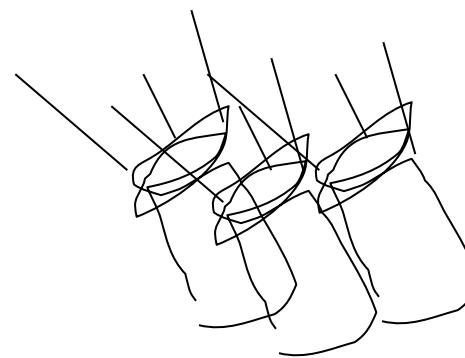
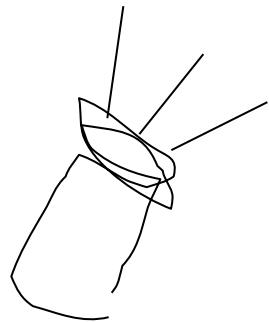
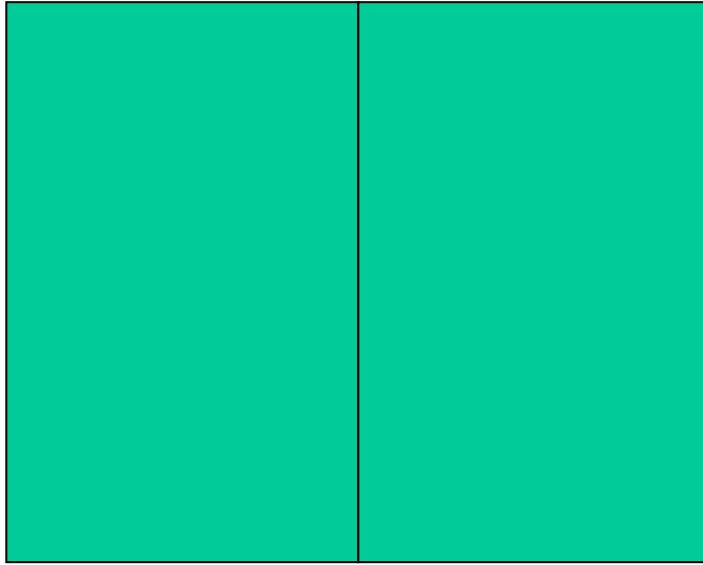
CIE 1931

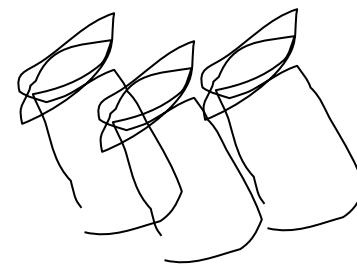
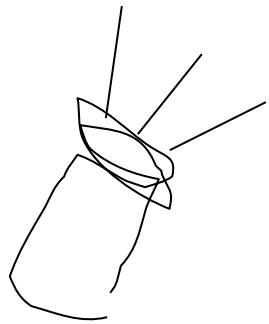
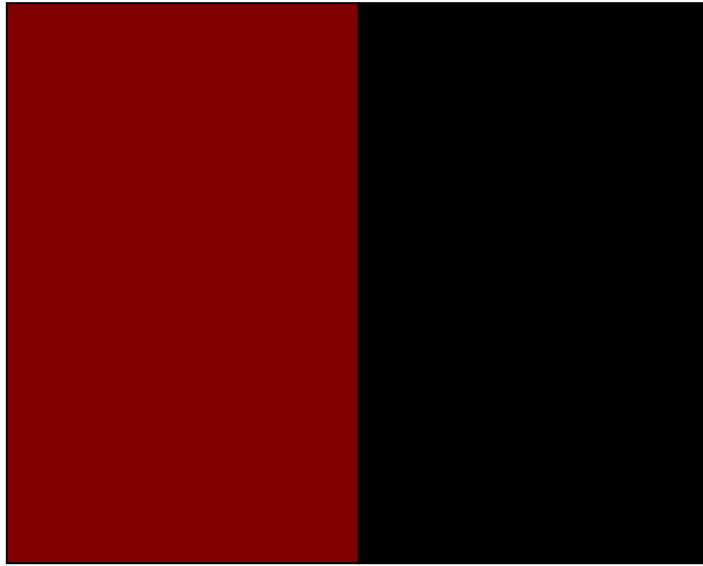
- В 1920х годах Уильям Райт и Джон Гилд экспериментировали с цветами! (и людьми)
- Участники эксперимента имели доступ к трем источникам базовых цветов
- Участникам показывали цвет
- Участники должны были так отрегулировать базовые цвета, чтобы оба цвета совпадали

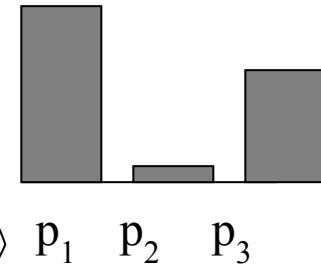
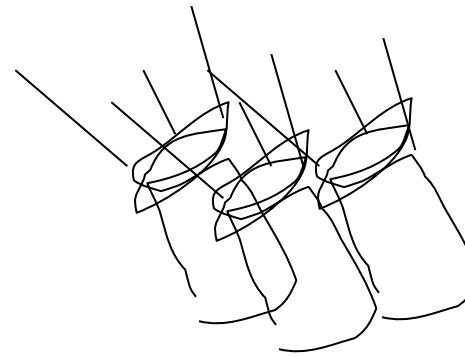
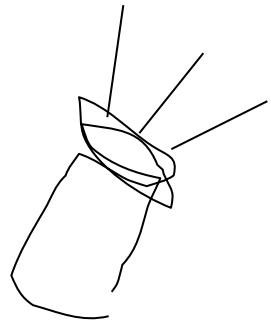
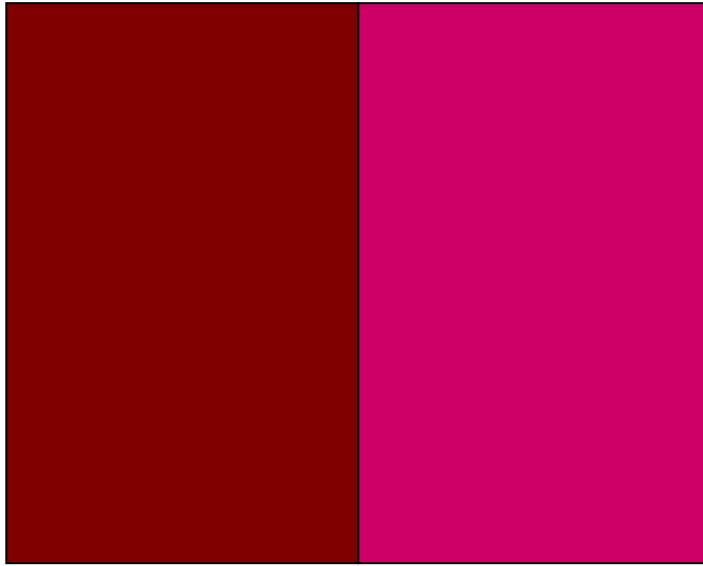


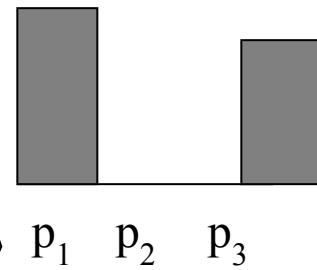
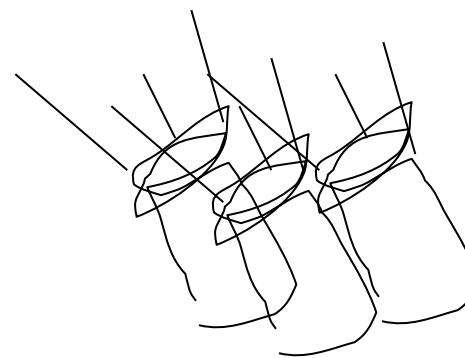
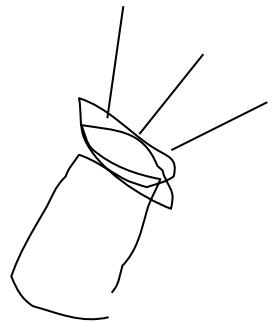
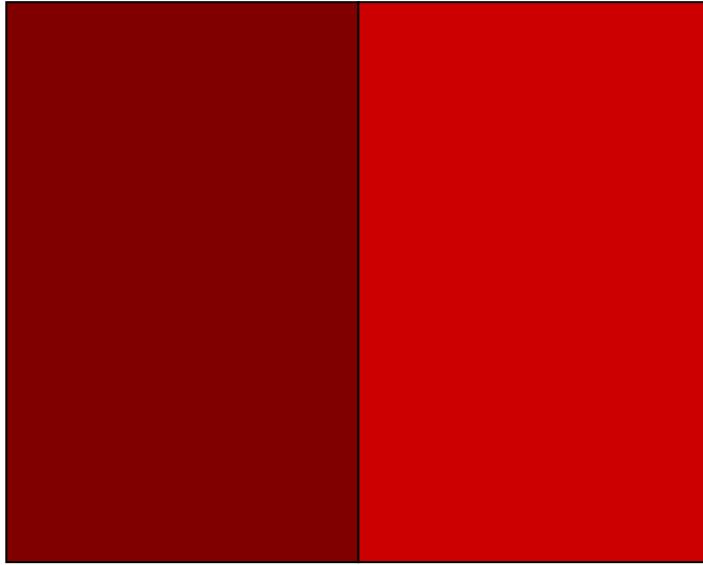


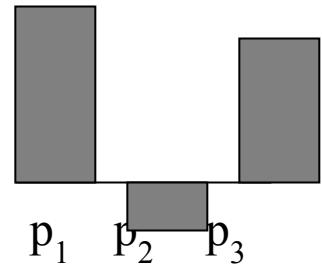
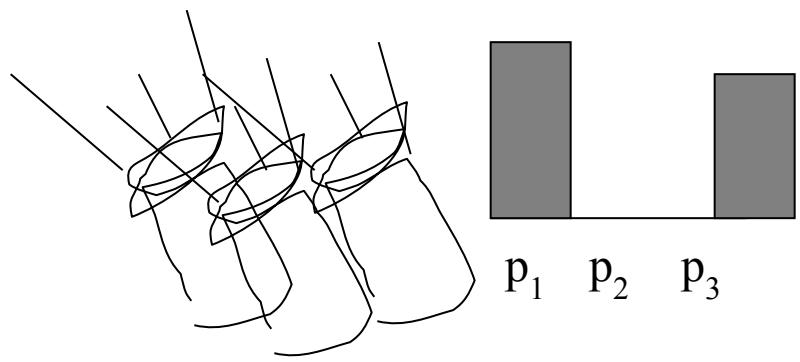
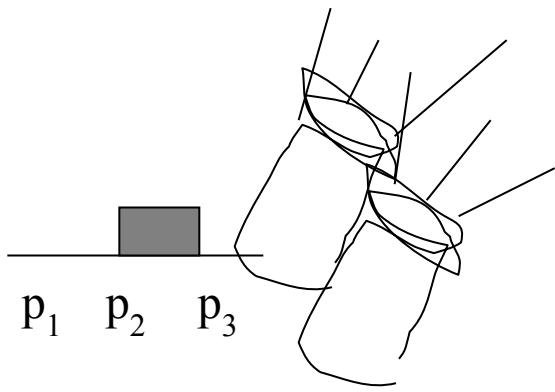
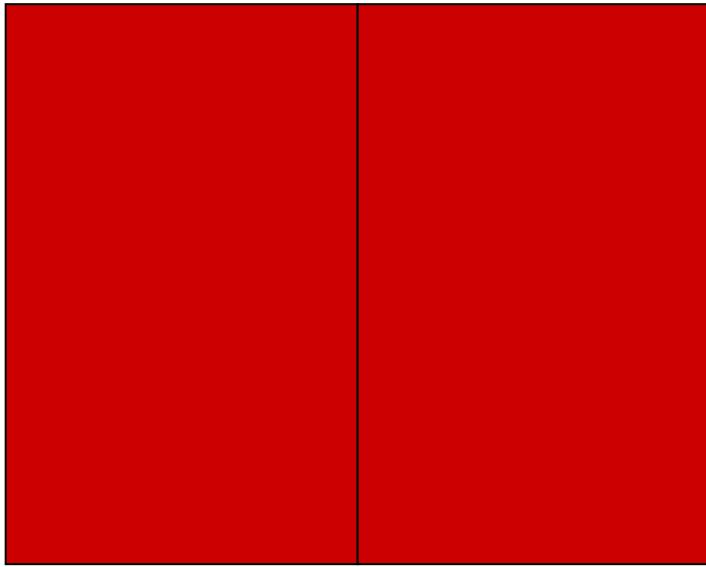


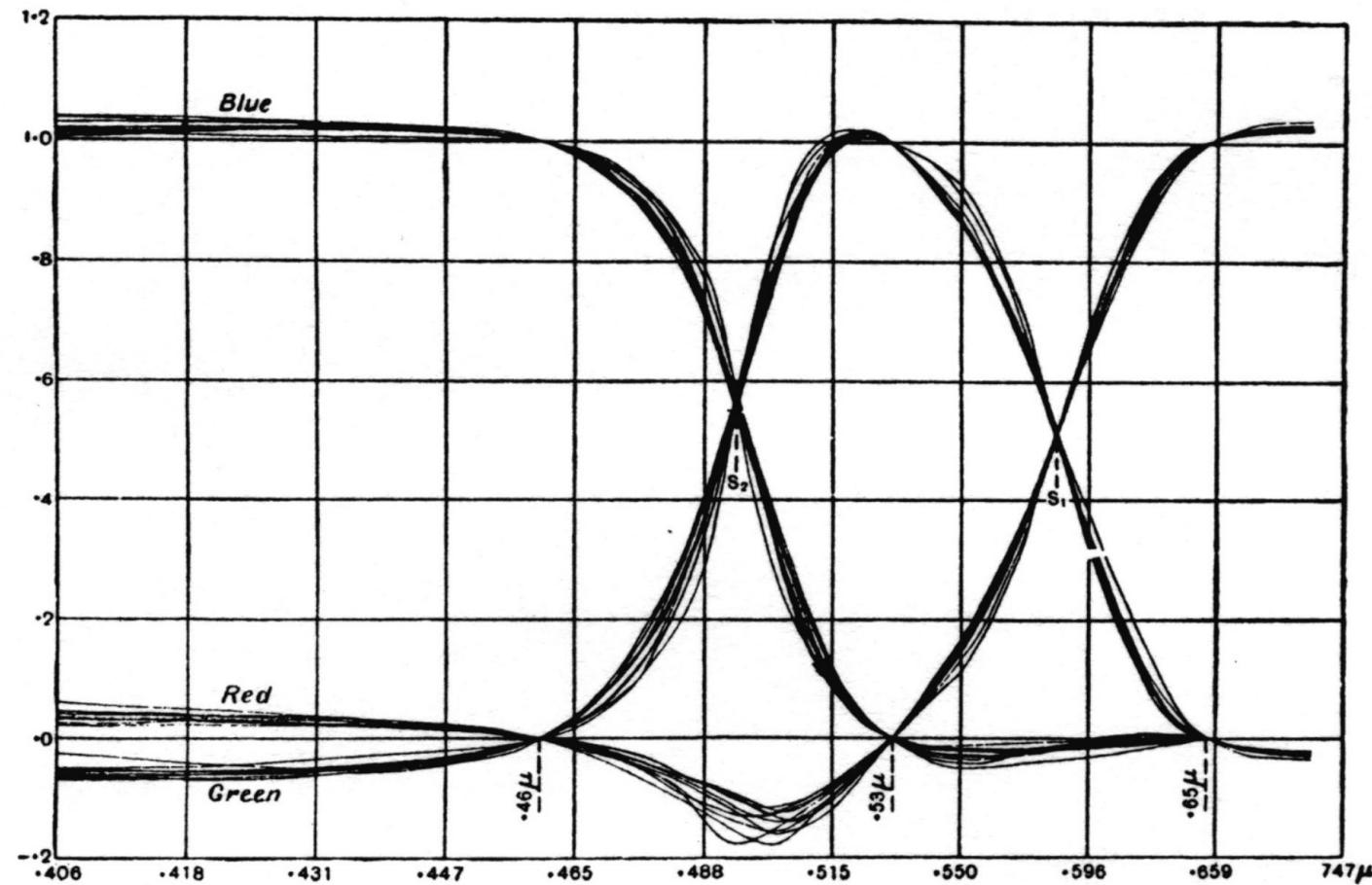


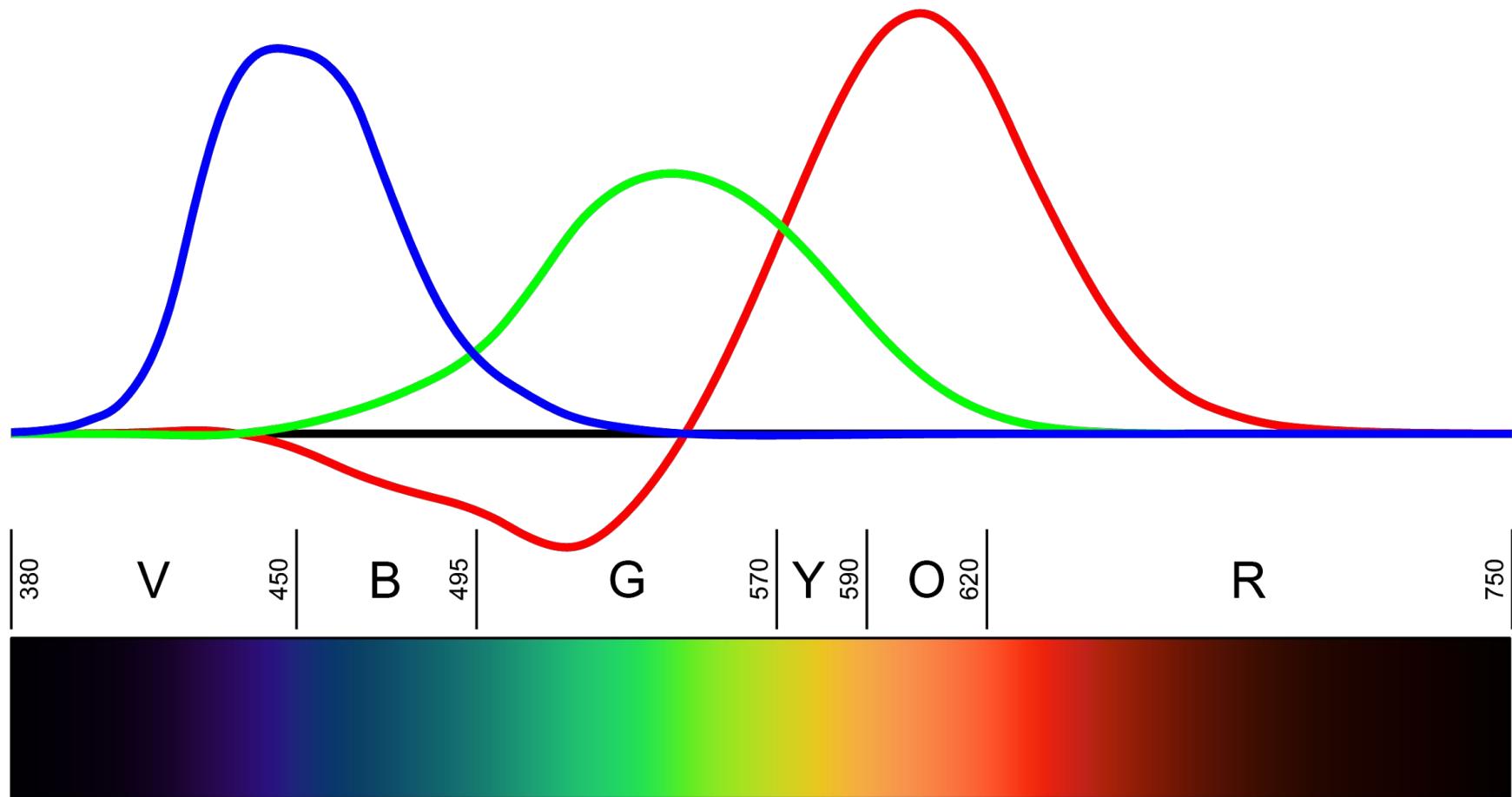


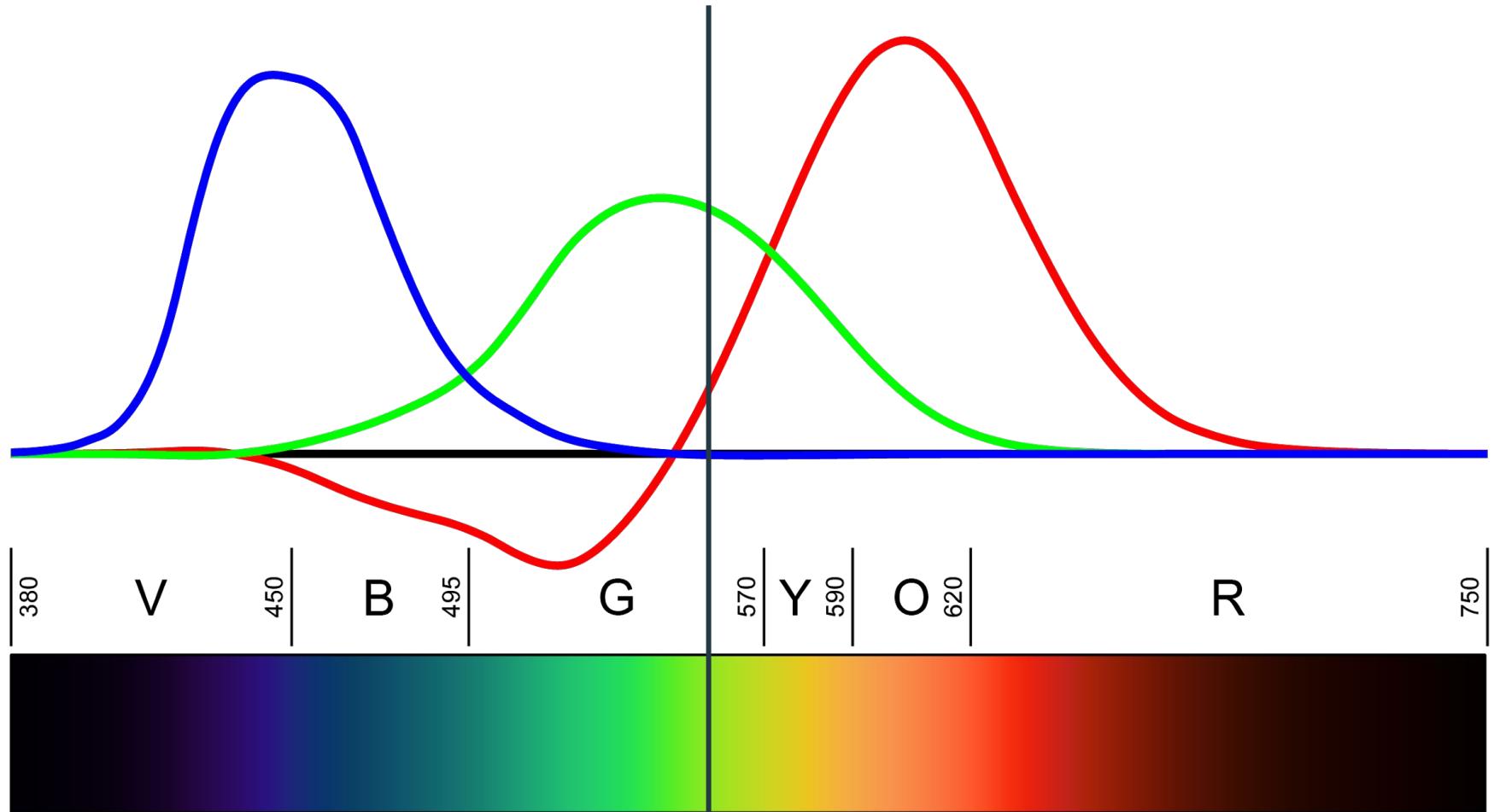


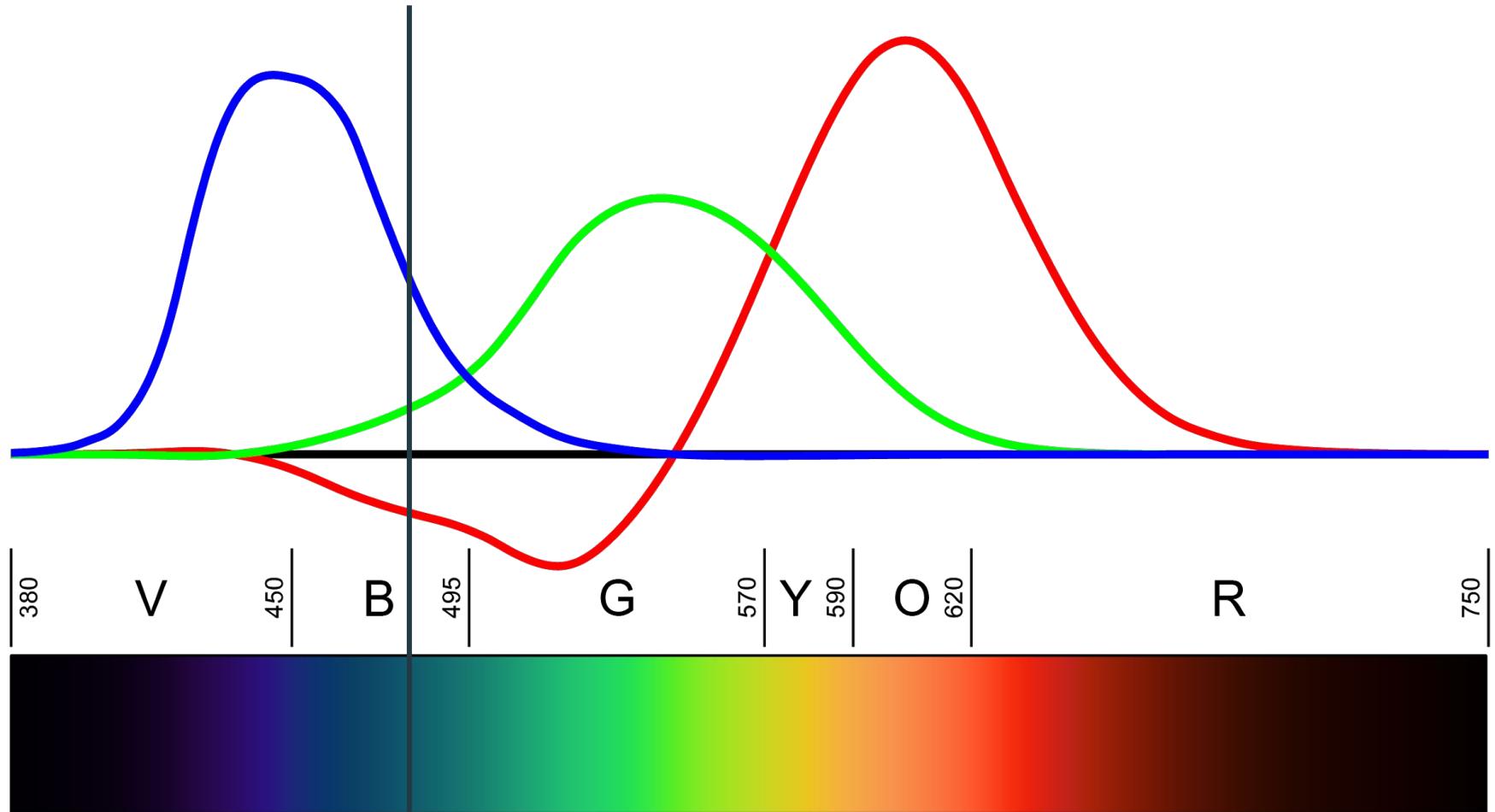








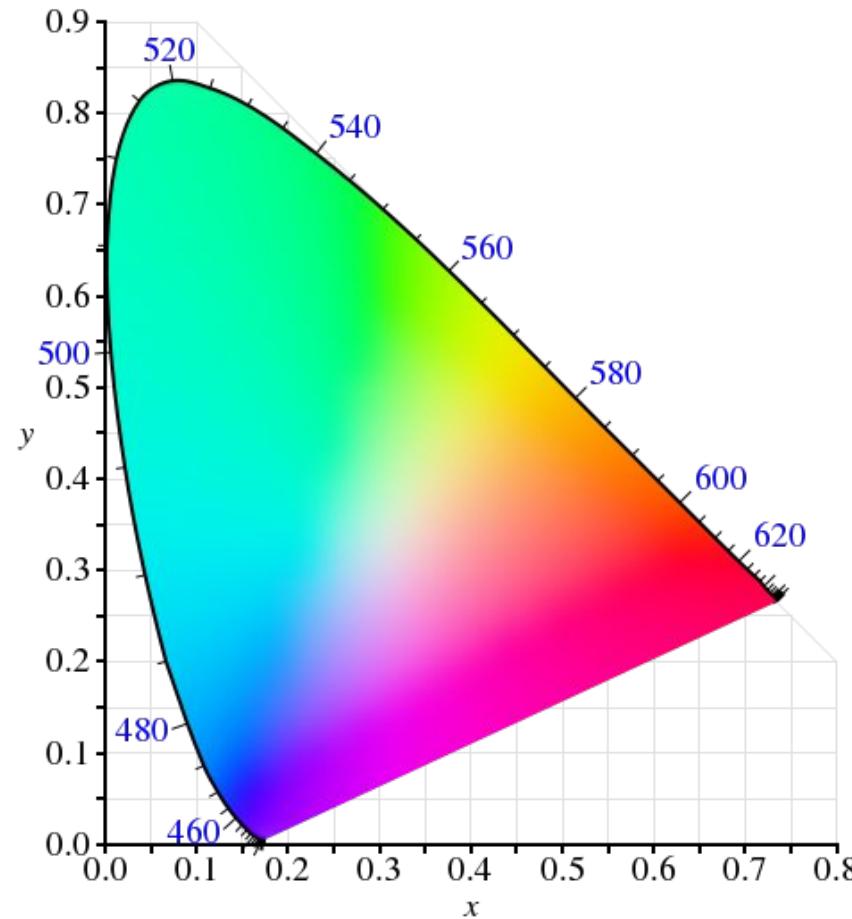




Итог

- Имея 3 базовых цвета, люди могут воспроизвести любой цвет
- Люди выбирают похожие распределение для любого цвета
- Цвет подчиняется классному линейному закону: законы Гассмана
 - $A=B+C \Rightarrow A+D=B+C+D$

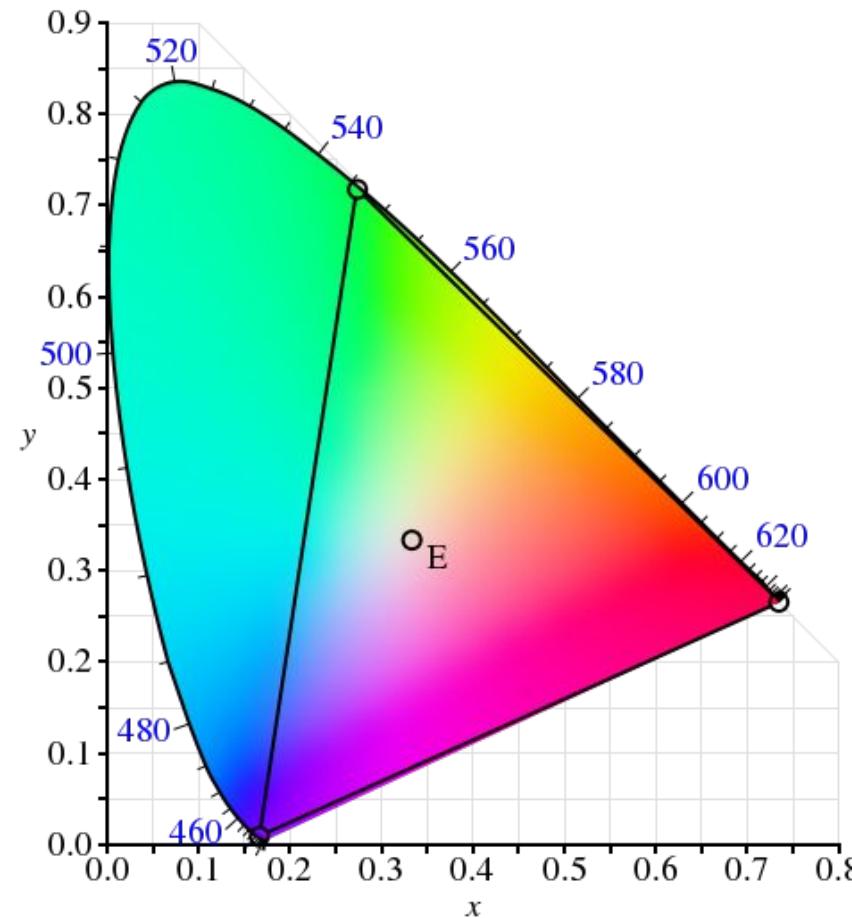
Теперь можно сделать карту цветов!



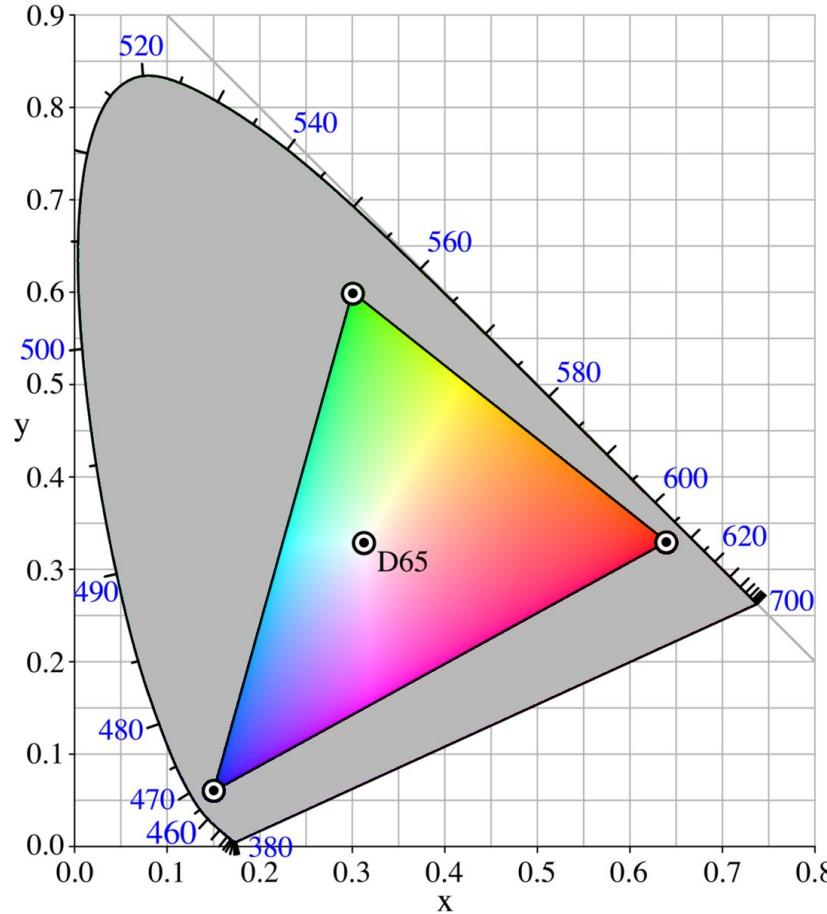
Линейное цветовое пространство

- Выбрать некоторые базовые цветы
- Можно смешивать базовые цветы, чтобы матчить с любым цветом вне заданного треугольника

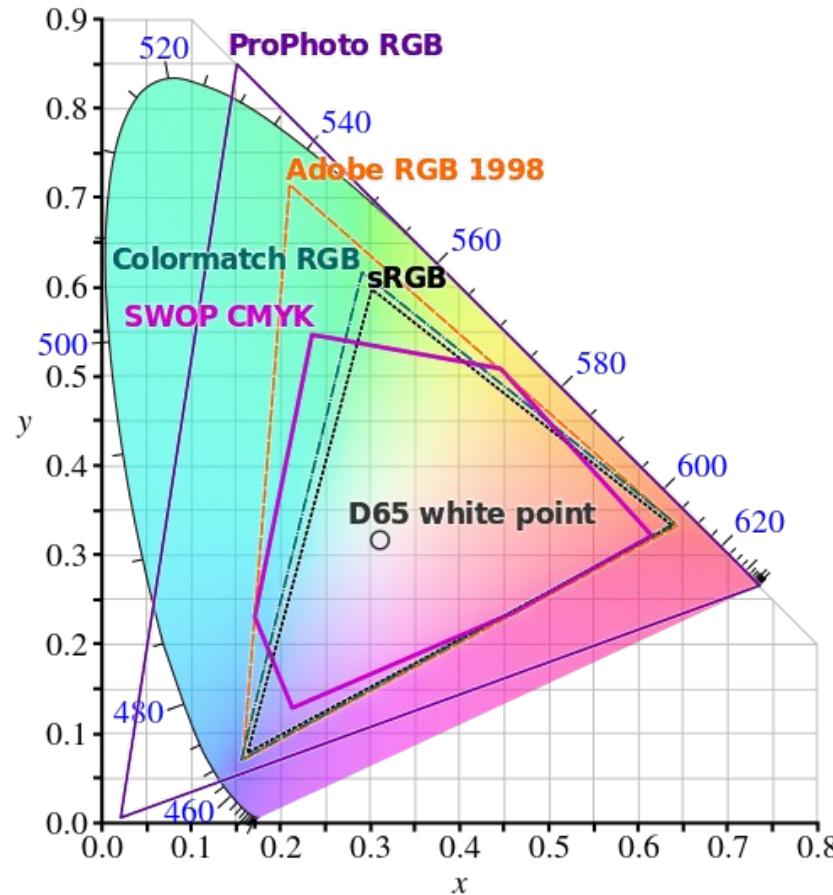
“Теоретический” CIE RGB треугольник



sRGB треугольник (HP + Microsoft)



Множество различных цветовых пространств

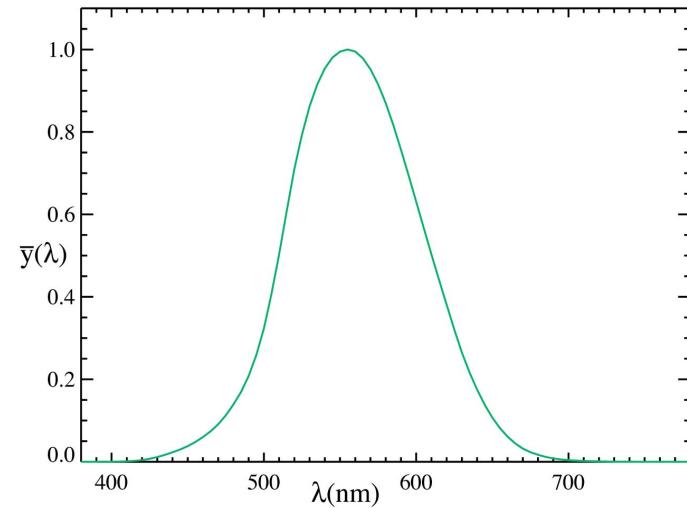


Что это значит при работе с цифровыми изображениями?

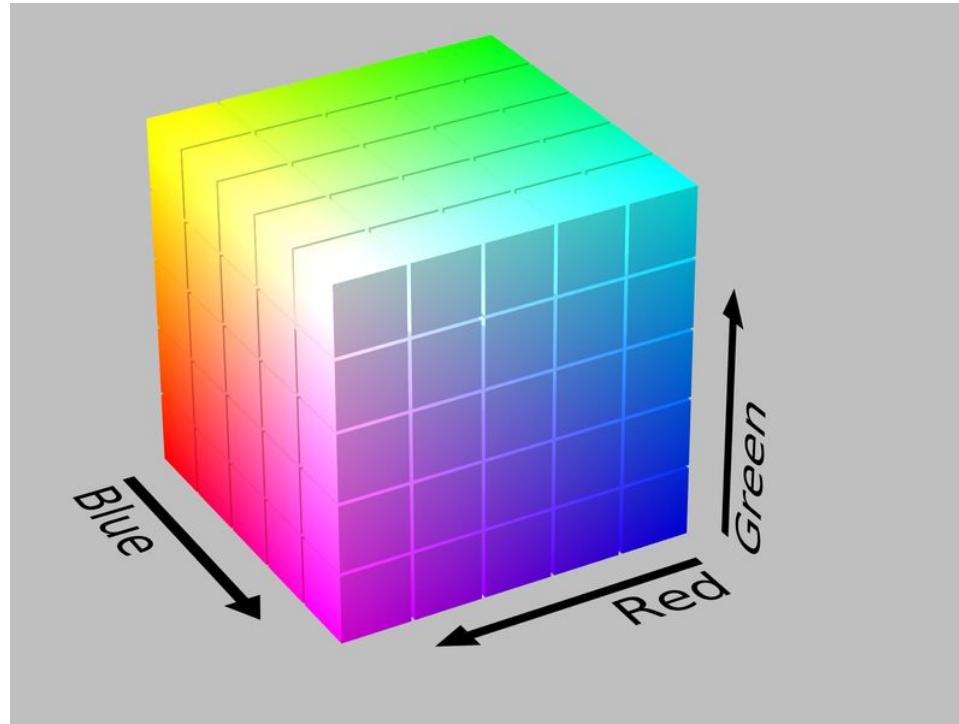
- Мы представляем изображение в виде сетки пикселей
- У каждого пикселя есть цвет, 3 компоненты: RGB
- Не каждый цвет может быть представлен в виде RGB!
- RGB был задизайнен, чтобы обхитрить зрение людей, а не чтобы быть точным
- sRGB на самом деле не линейный
 - Люди видят разницу между различными темными тонами больше, чем между светлыми
 - Сжать светлые тона, расширить темные тона - эффективней
- Можно представить цвет в виде 3 чисел:
 - #ff00ff; (1.0, 0.0, 1.0); 255,0,255; и т.д.

ЧБ изображения - шаг назад

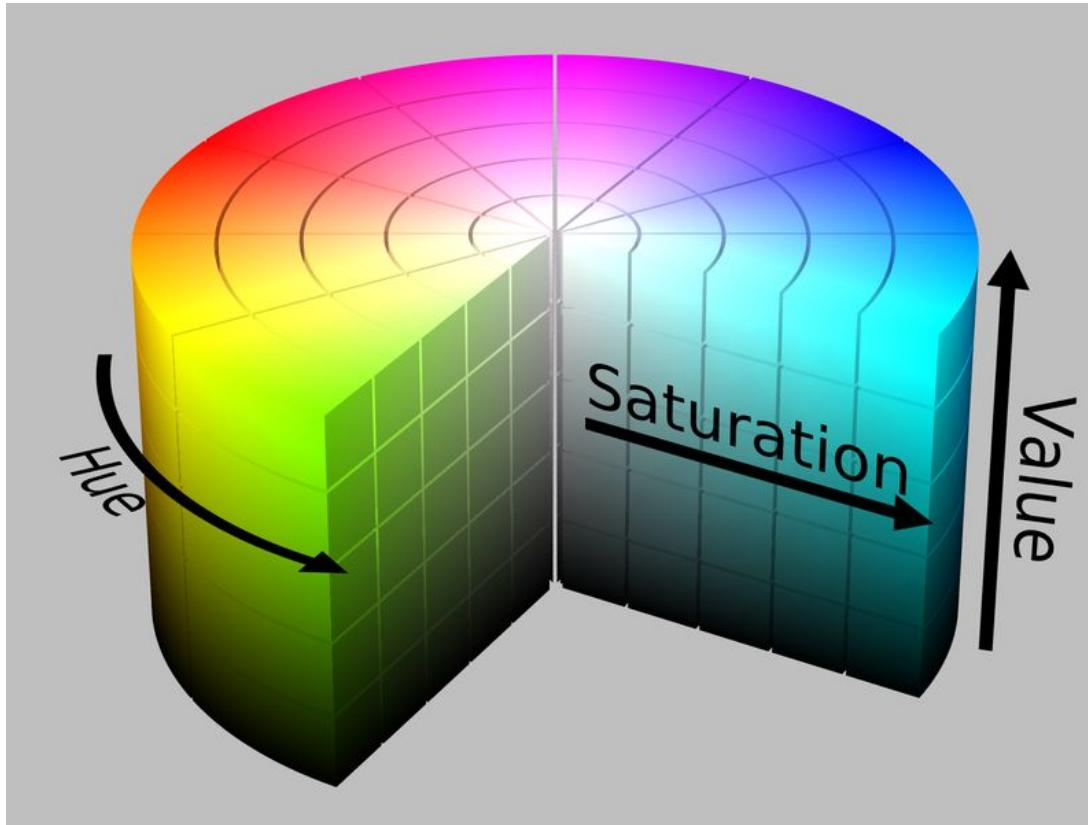
- Можно симулировать монохромные изображения с помощью RGB
- Хотим оставить лишь компоненту яркости изображения
- $(R+B+G/3)$ - выглядит так себе
- *По-хорошему надо сделать*
 - Гамма декомпрессинг
 - Высчитывания яркости
 - Гамма компрессинг
- Или так :)
 - $.30R + .59G + .11B$



RGB: кубик

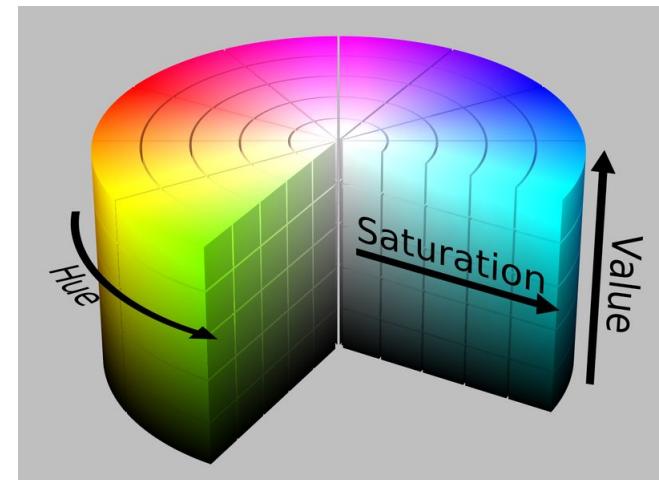


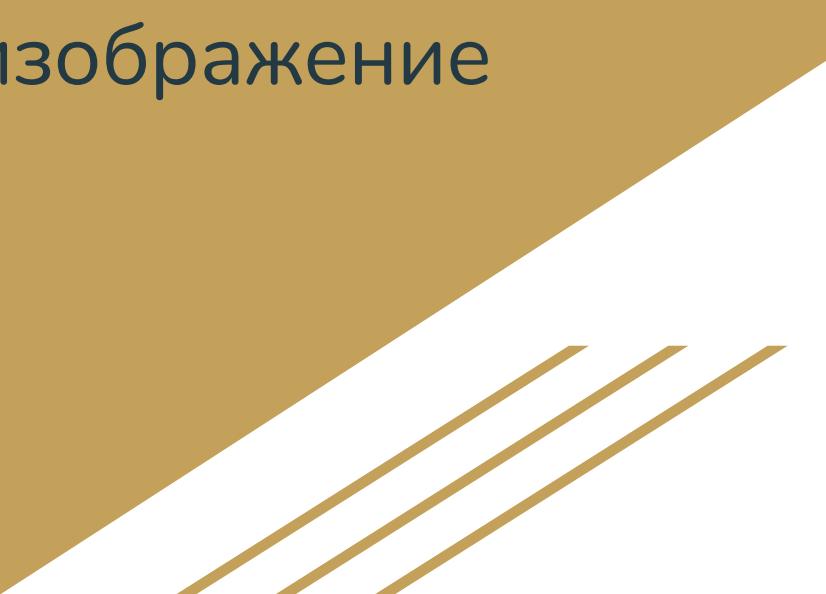
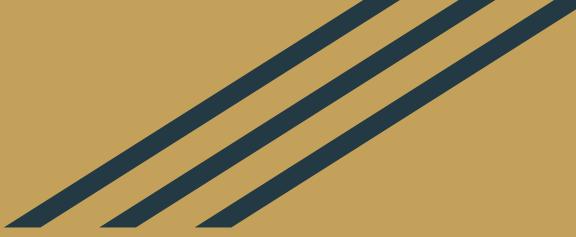
Hue, Saturation, Value (HSV): цилиндр!



HSV

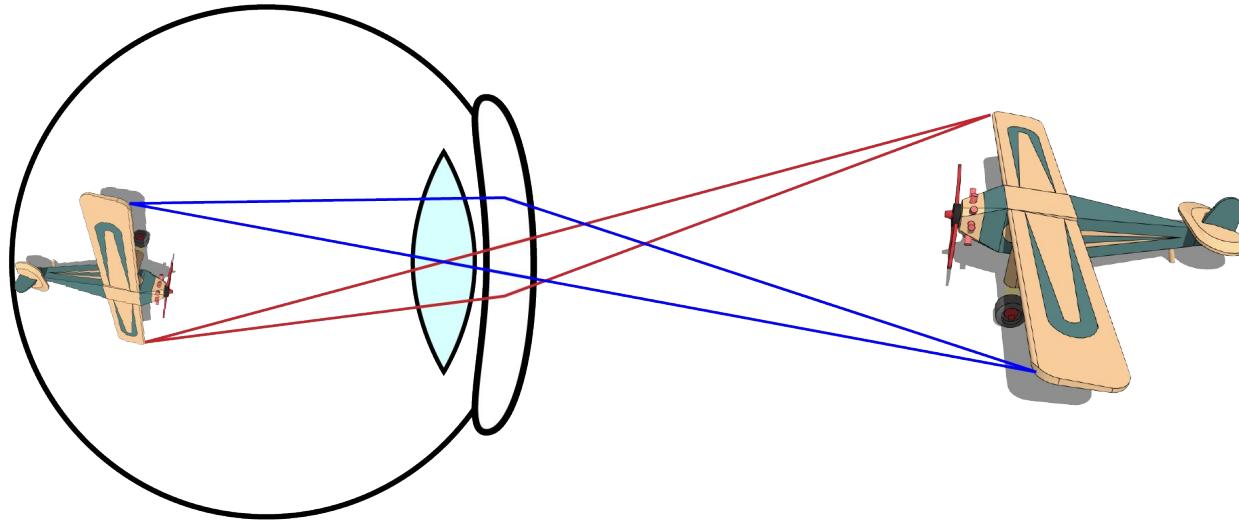
- Другое цветовое пространство
- Лучше аппроксимирует человеческое восприятие цвета
- Hue (оттенок): какой цвет
- Saturation (насыщенность): сколько цвета
- Value (яркость): насколько ярко
- Позволяет легче производить цветовые преобразования
 - Смещать оттенки
 - Повышать насыщенность



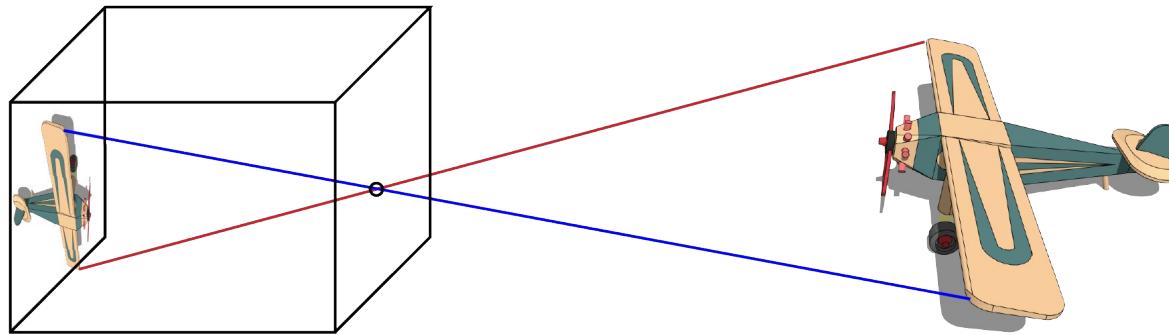


Что же такое изображение

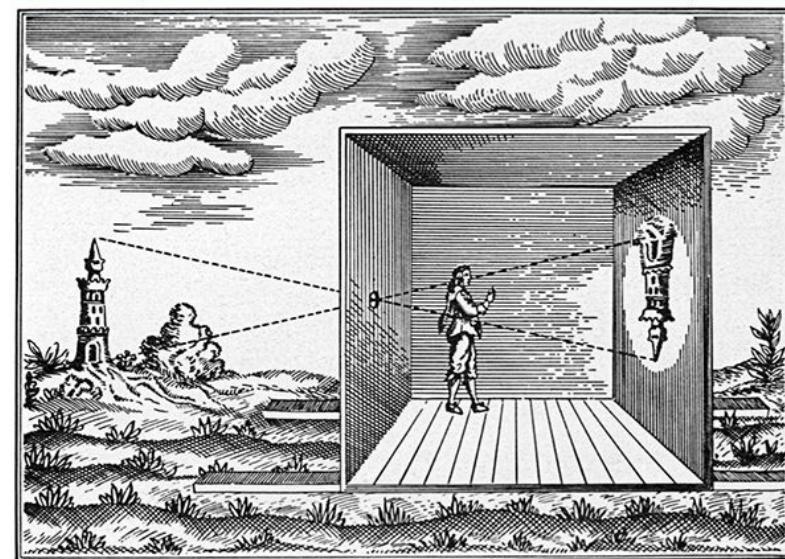
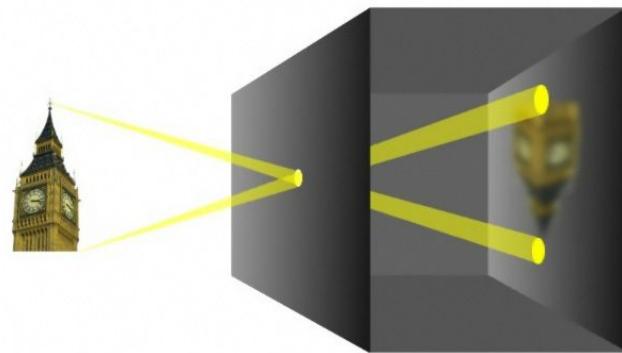
Глаза: проекция на сетчатку



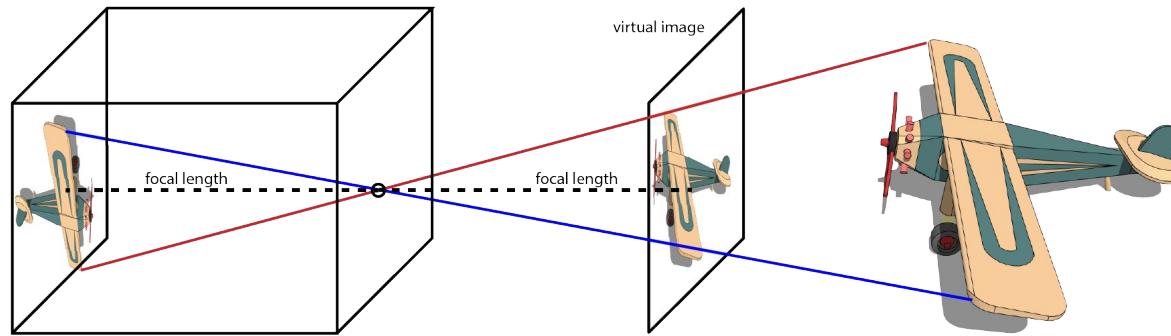
Модель: pinhole camera (камера обскура)



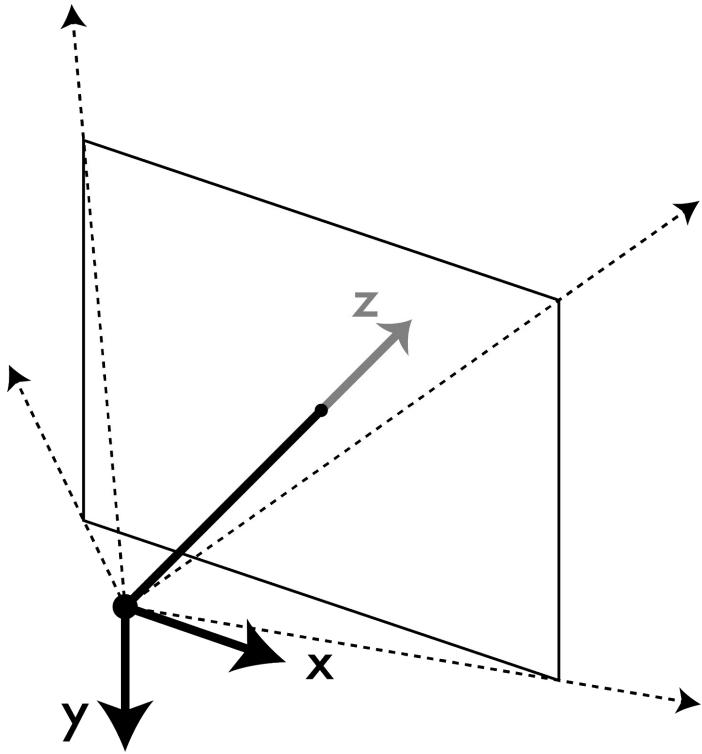
Камера обскура



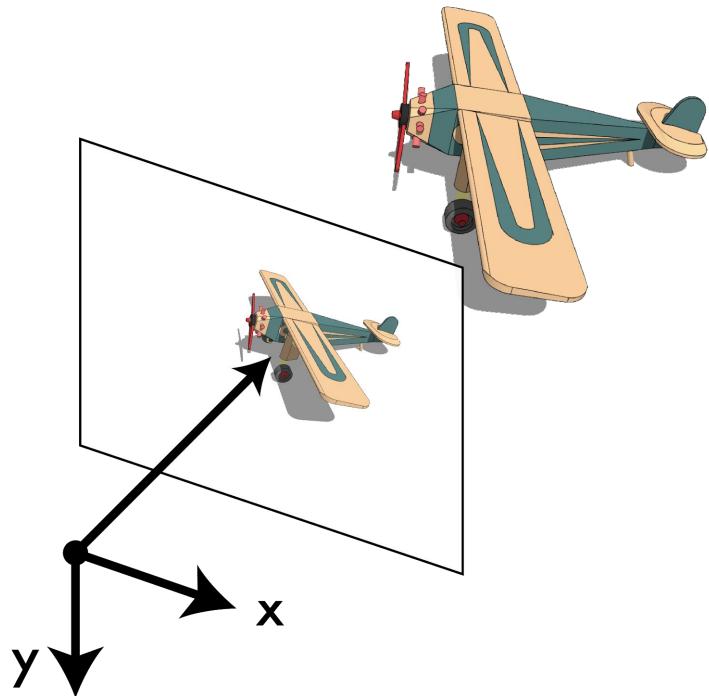
Модель: pinhole camera (камера обскура)



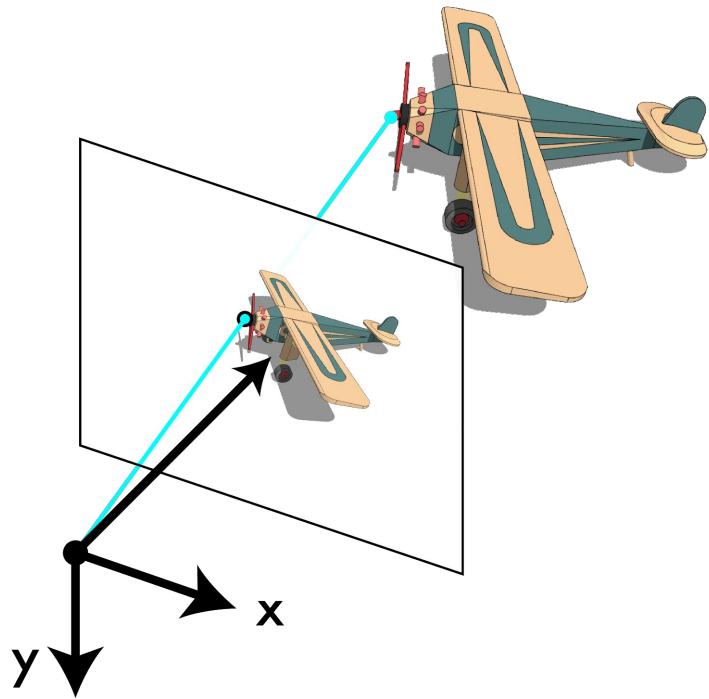
Изображение: 3d \rightarrow 2d проекция окружающего мира



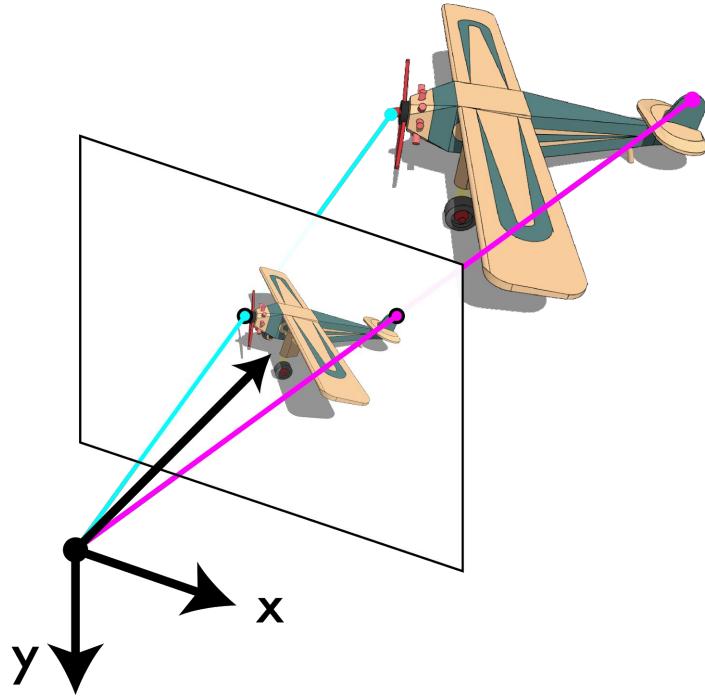
Изображение: 3d \rightarrow 2d проекция окружающего мира



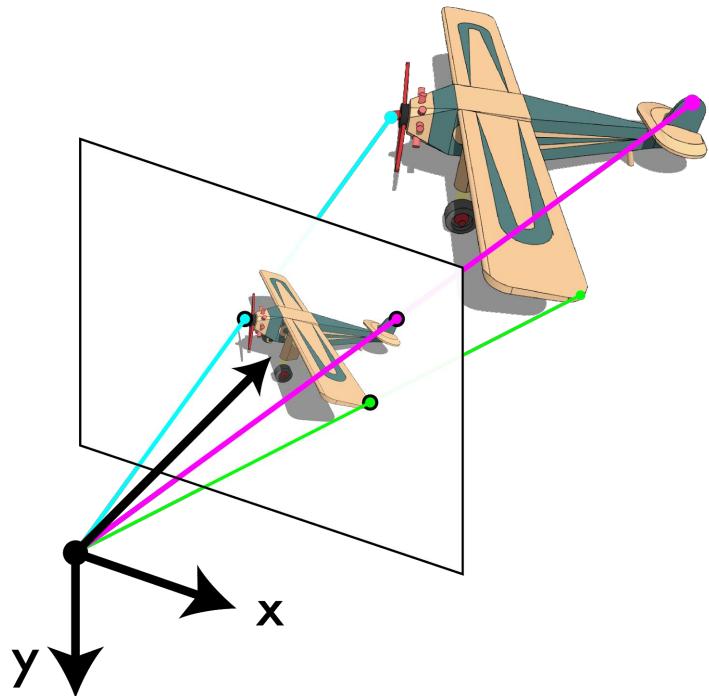
Изображение: 3d \rightarrow 2d проекция окружающего мира



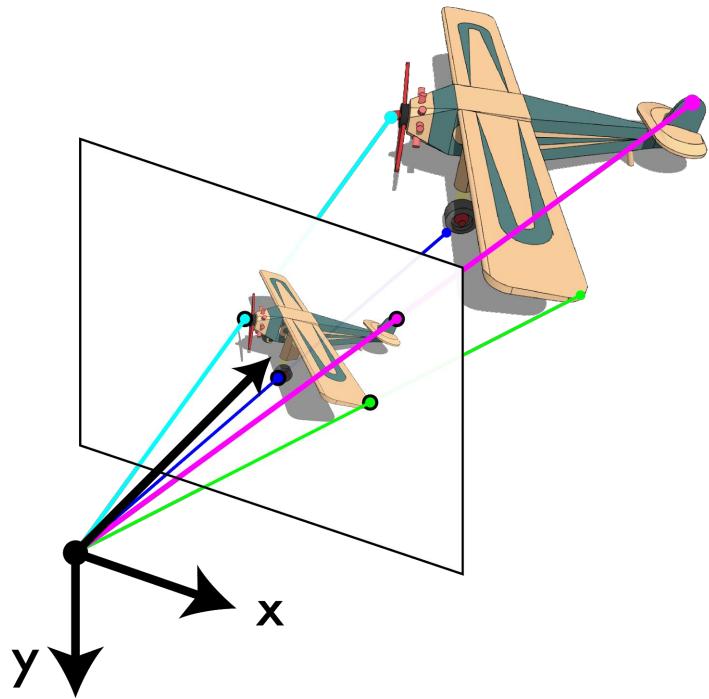
Изображение: 3d \rightarrow 2d проекция окружающего мира



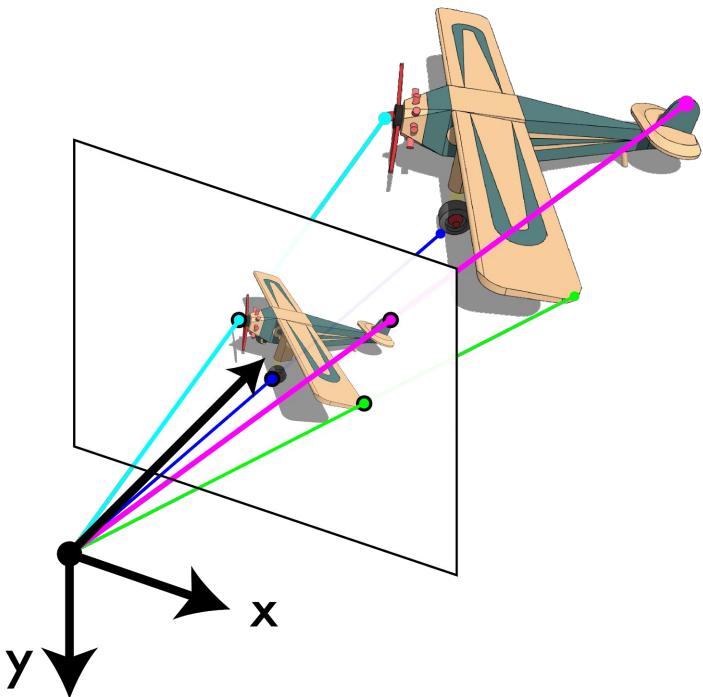
Изображение: 3d \rightarrow 2d проекция окружающего мира



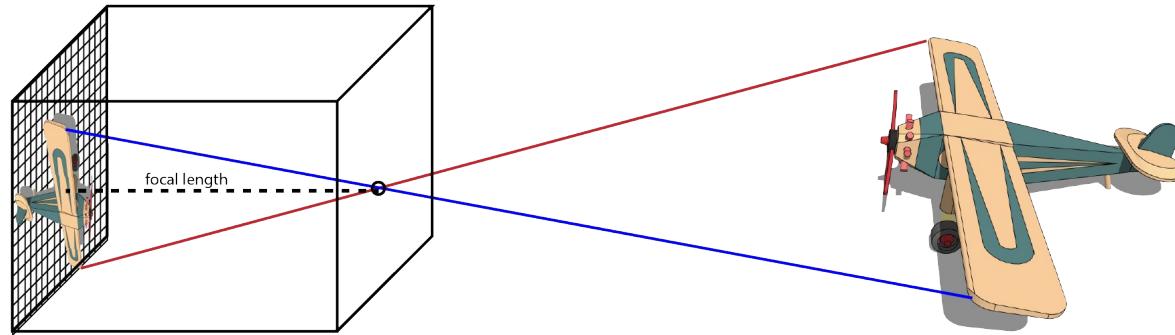
Изображение: 3d \rightarrow 2d проекция окружающего мира



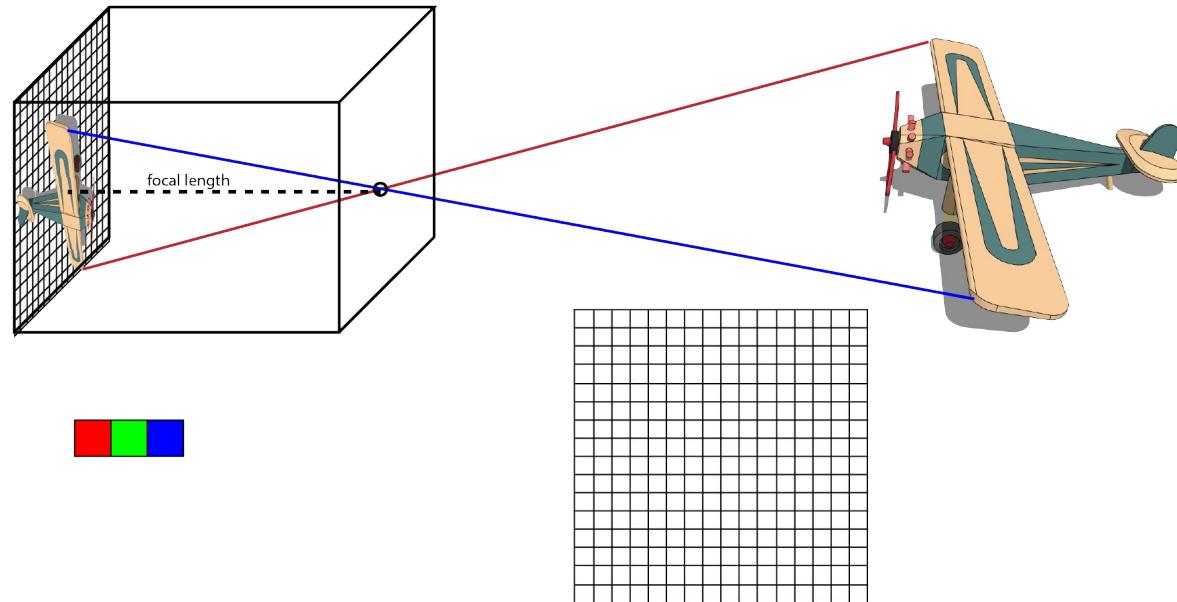
В каждой точке мы записываем входящий свет



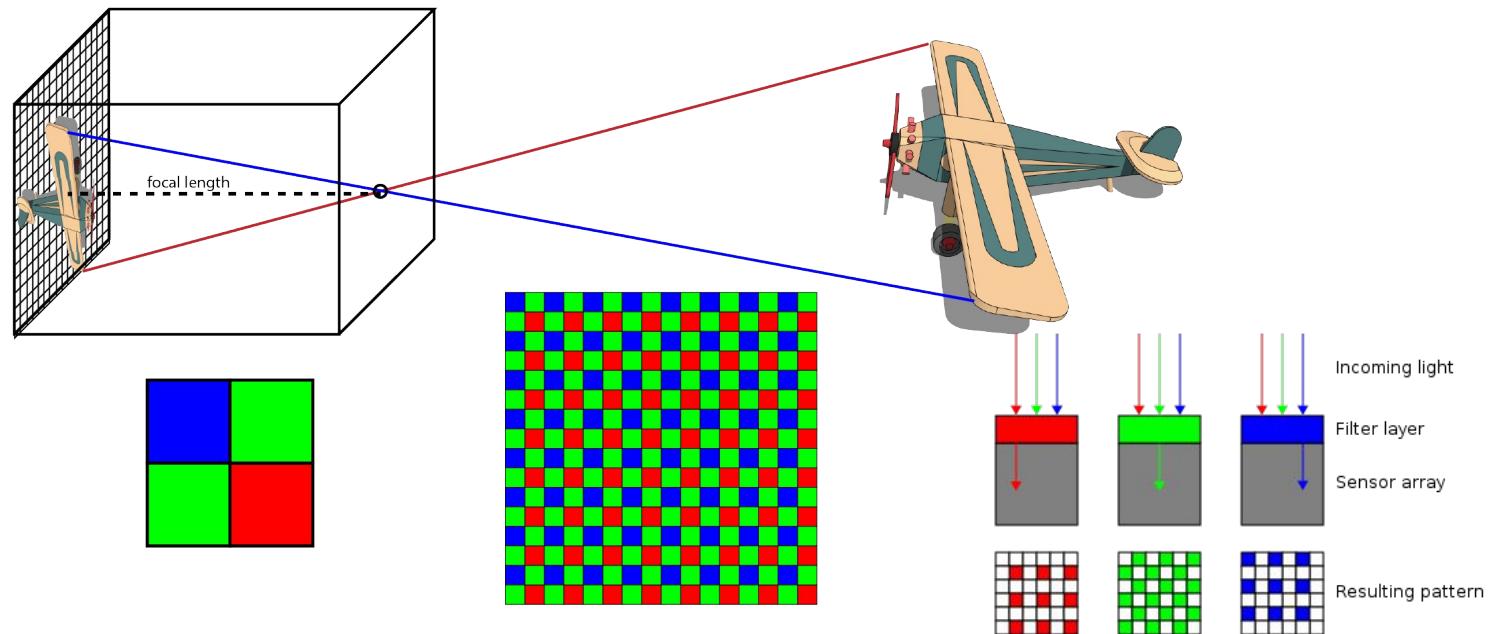
В каждой точке мы записываем входящий свет



Как записывать цвет?

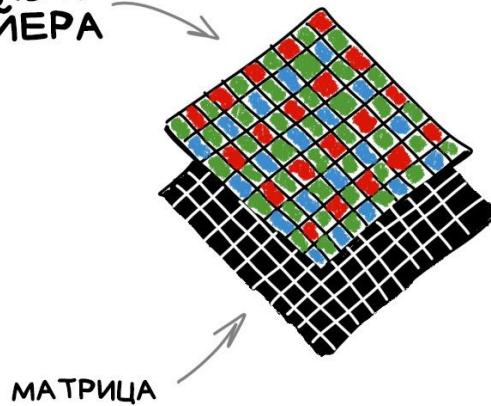


Фильтр Байера

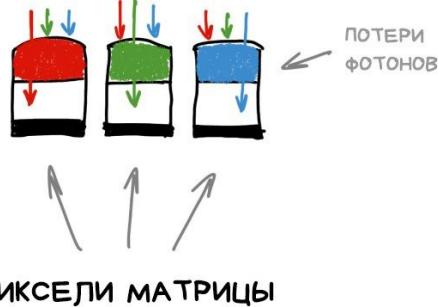


Фильтр Байера

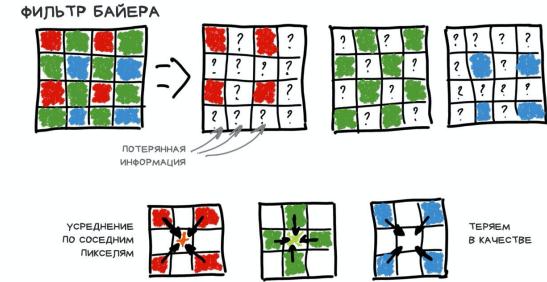
ФИЛЬТР
БАЙЕРА



СВЕЕЕЕЕЕЕЕЕЕТ

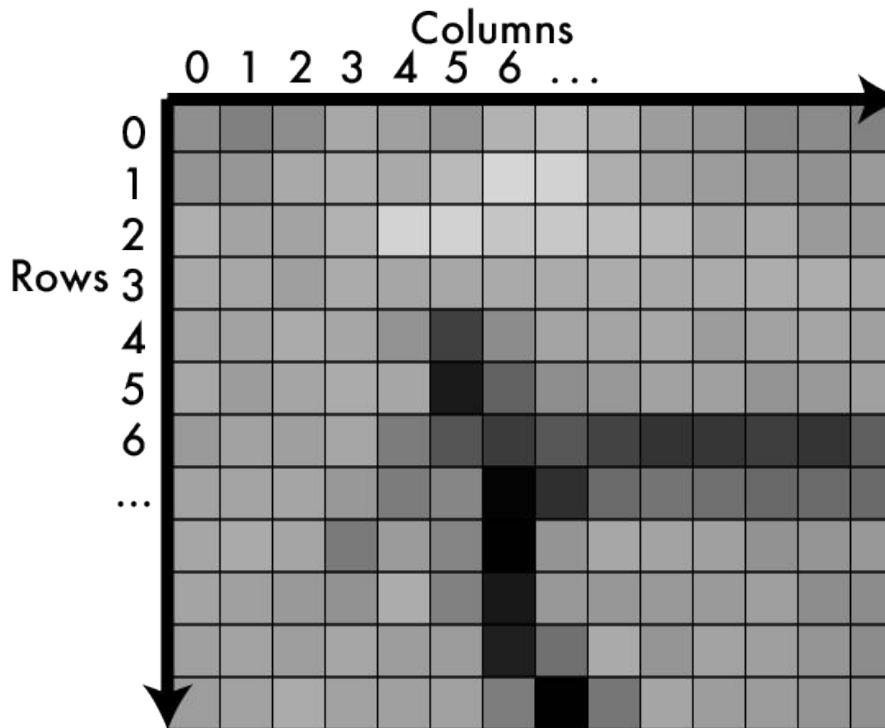


ДЕБАЙЕРИЗАЦИЯ (ДЕМОЗАИК)



Очень интересный блогпост про цифровую фотографию:
https://vas3k.ru/blog/computational_photography/

Изображение - матрица



Значения в матрице = сколько света попало

		Columns													
		0	1	2	3	4	5	6	...						
Rows	0	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	1	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	2	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	3	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	4	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	5	100	102	107	102	132	30	60	156	148	122	115	104	105	103
	6	100	102	107	102	132	40	20	50	32	20	20	24	30	62
	...	100	102	107	102	132	71		156	51	57	57	58	62	58
		100	102	107	102	132	69		156	148	122	115	104	105	103
		100	102	107	102	132	89	12	156	148	122	115	104	105	103
		100	102	107	102	132	146	13	45	148	122	115	104	105	103
		100	102	107	102	132	146	46		42	122	115	104	105	103

Значения в матрице = сколько света попало

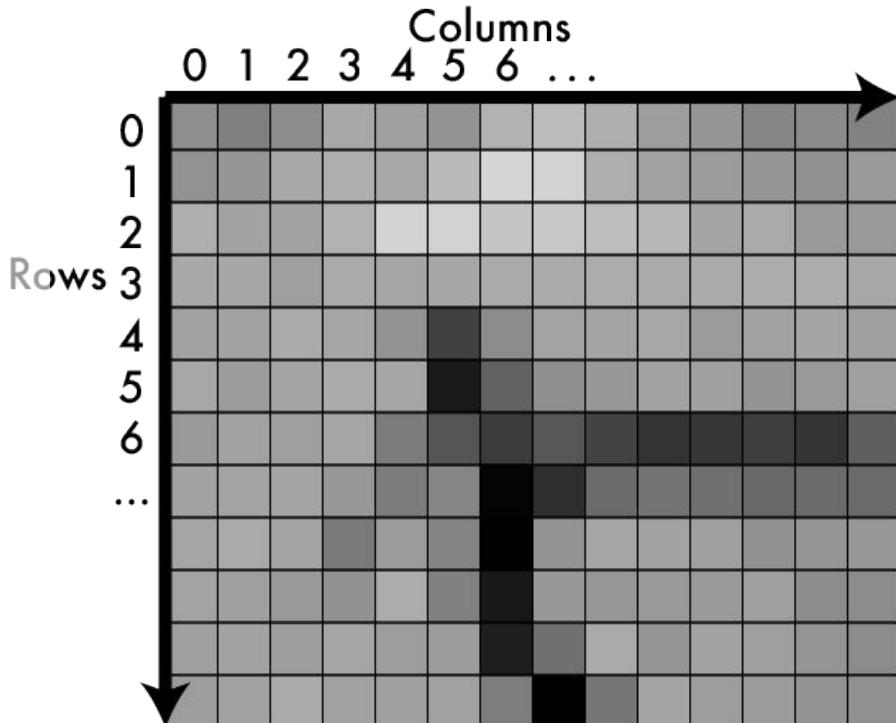
- Выше значения = больше света
- Меньше значения = меньше света
- Ограничено сверху и снизу
 - Нету света = 0
 - Лимит устройства = max
 - Типичные границы:
 - [0-255], 1 байт
 - [0-1], floating point
- Пиксели!

		Columns													
		0	1	2	3	4	5	6	...						
Rows	0	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	1	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	2	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	3	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	4	100	102	107	102	132	146	136	156	148	122	115	104	105	103
	5	100	102	107	102	132	30	60	156	148	122	115	104	105	103
	6	100	102	107	102	132	40	20	50	32	20	20	24	30	62
	...	100	102	107	102	132	71		156	51	57	57	58	62	58
	8	100	102	107	102	132	69		156	148	122	115	104	105	103
	9	100	102	107	102	132	89	12	156	148	122	115	104	105	103
	10	100	102	107	102	132	146	13	45	148	122	115	104	105	103
	11	100	102	107	102	132	146	46		42	122	115	104	105	103

Нумерация пикселей

- Способы индексирования:

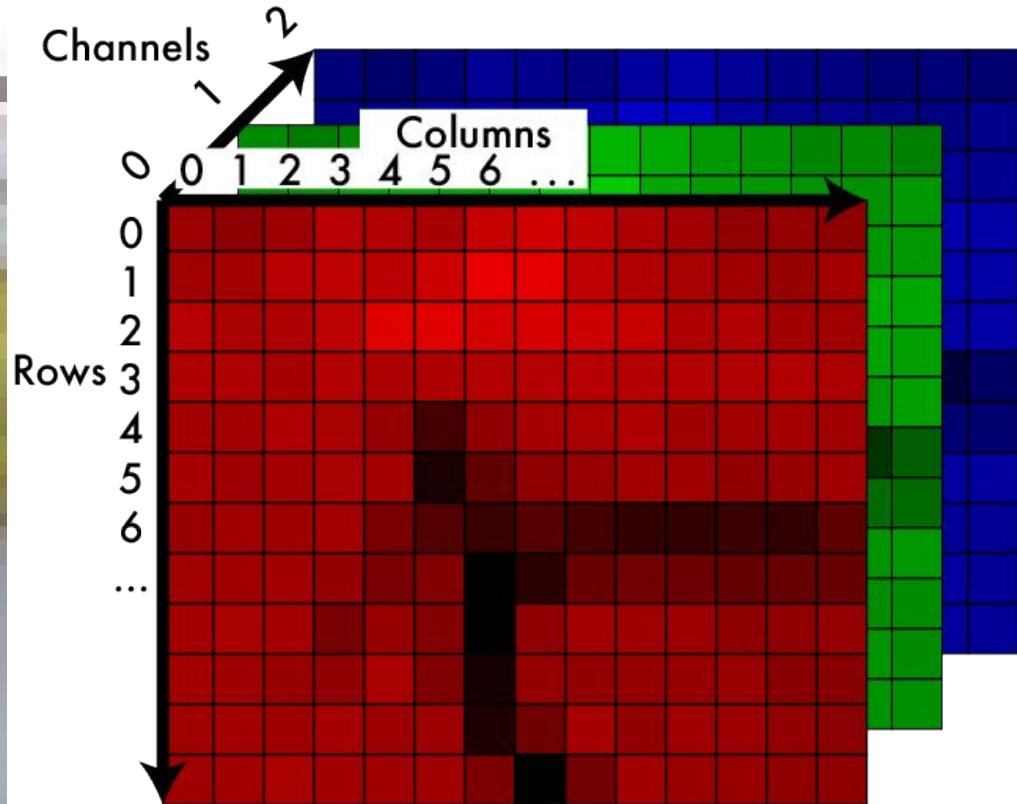
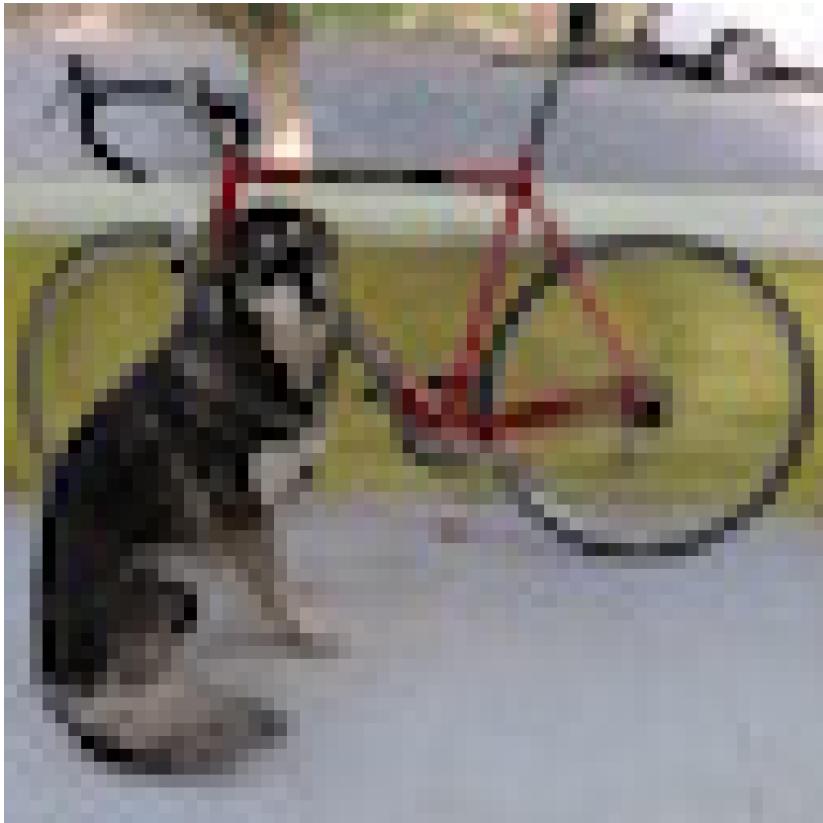
- (x,y)
 - Декартовы координаты
 - $(3,6)$ это столбец 3 строка 6
- (r,c)
 - Матричные обозначения
 - $(3,6)$ это строка 3 столбец 6



- Выбор опционален

- Главное не запутаться

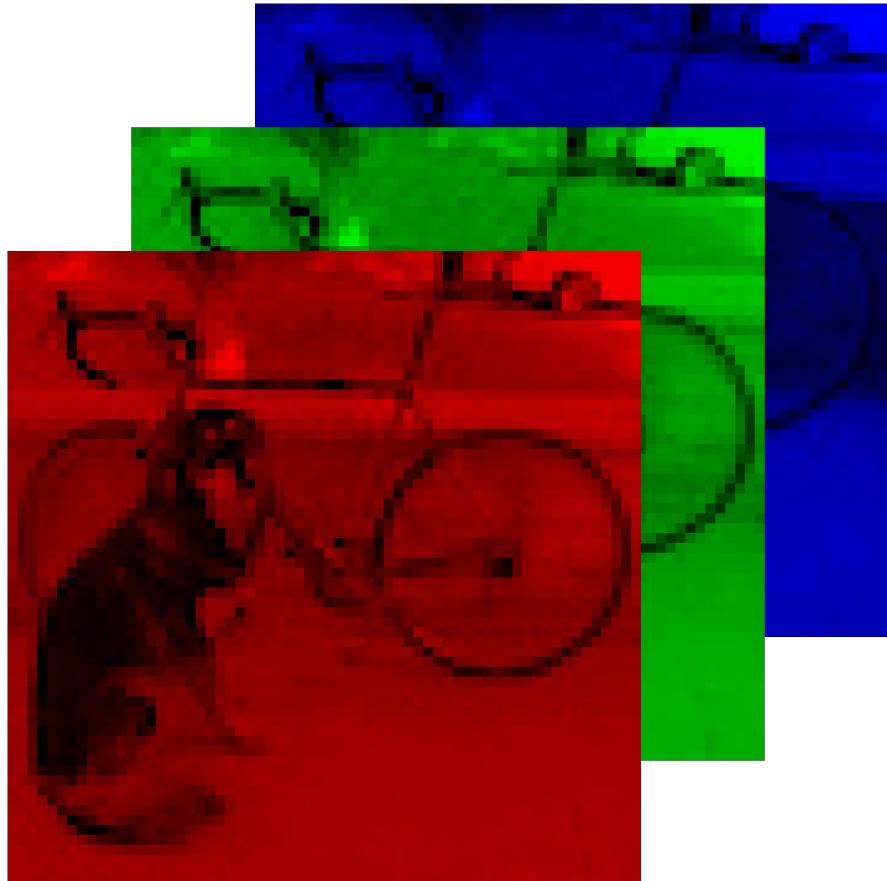
Цветное изображение: 3д тензор



Информация о каждом цвете в отдельном канале

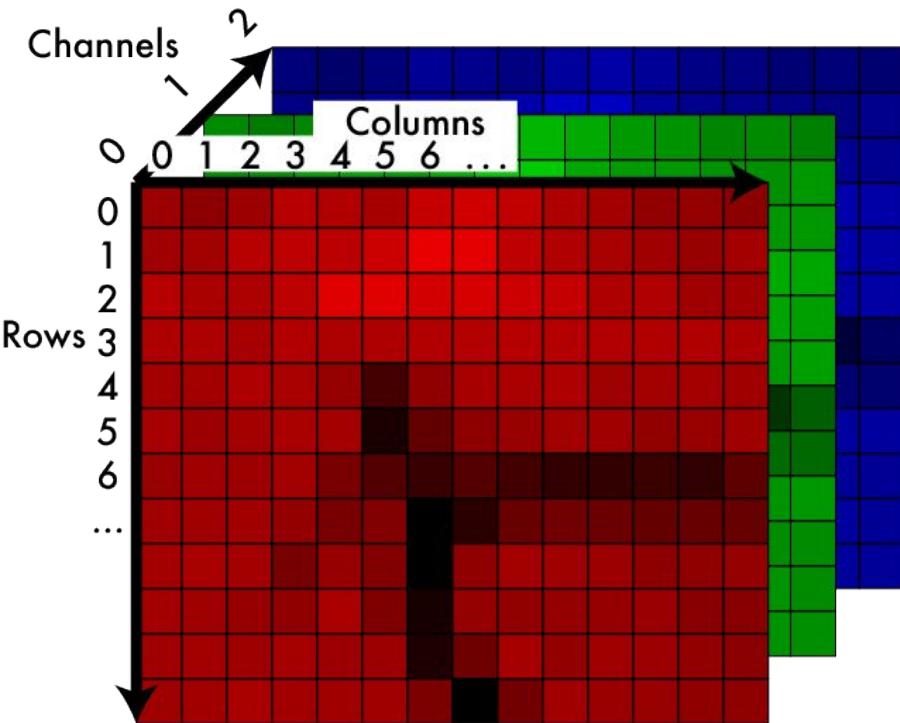
Помним: мы можем воспроизводить цвета, используя смесь базовых цветов.

Каждый канал представляет информацию об одном базовом цвете. Складывая значения каждого базового цвета, можно воспроизвести оригинальный цвет.

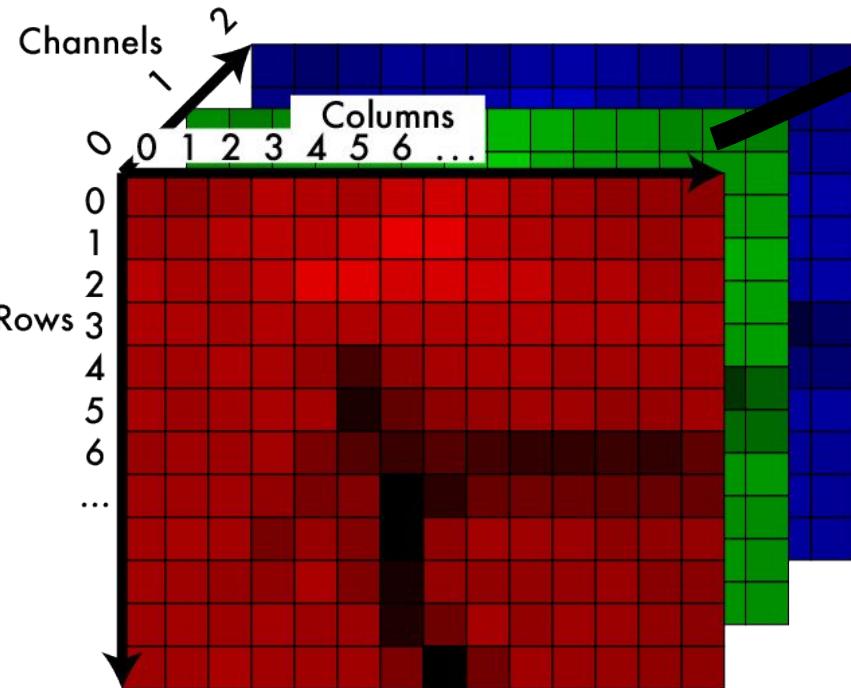


Нумерация пикселей

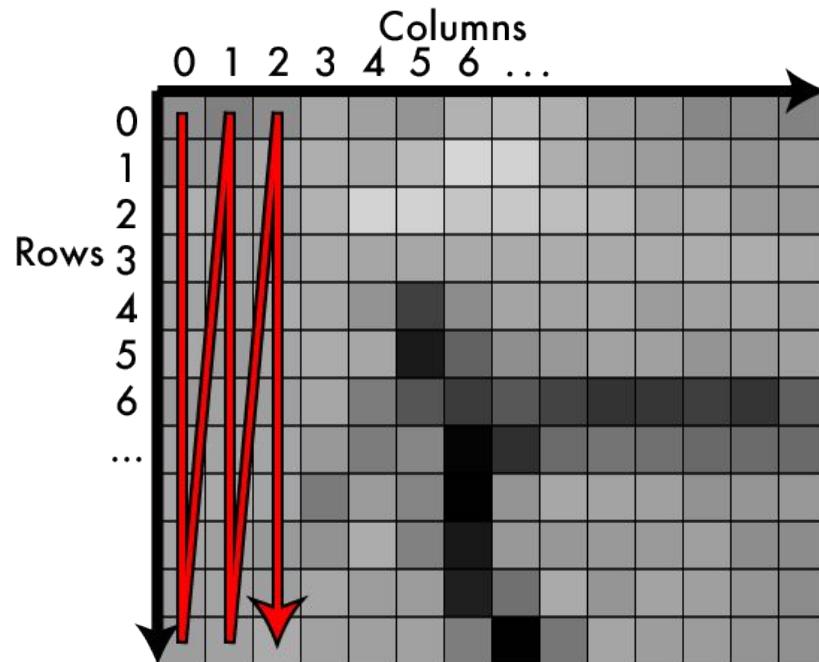
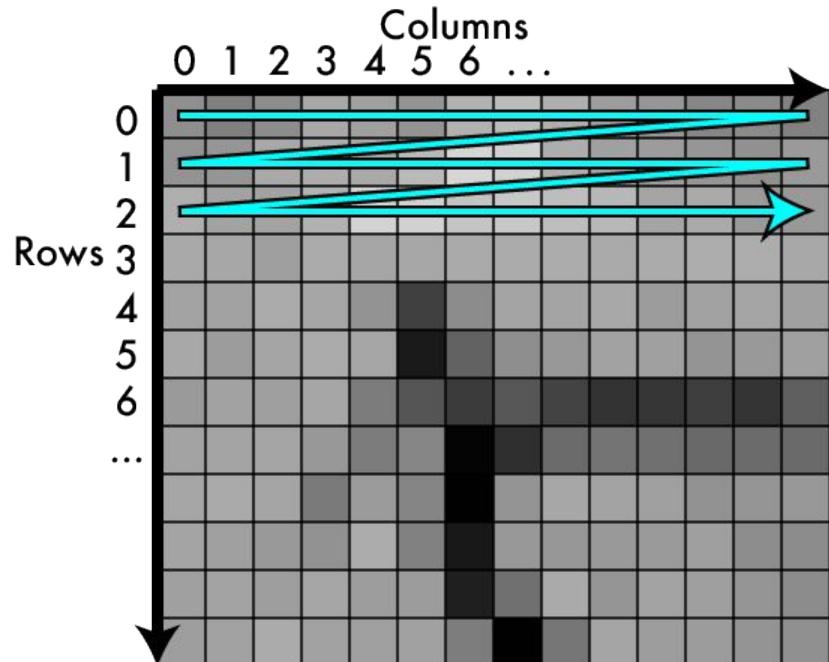
- (x, y, c)
 - $(1, 2, 0)$:
 - столбец 1, строка 2, канал 0
- Выбор все также опционален
- Главное не запутаться
- Размер изображений:
 - $1920 \times 1080 \times 3$:
 - 1920 пикселей в ширину
 - 1080 пикселей в высоту
 - 3 канала



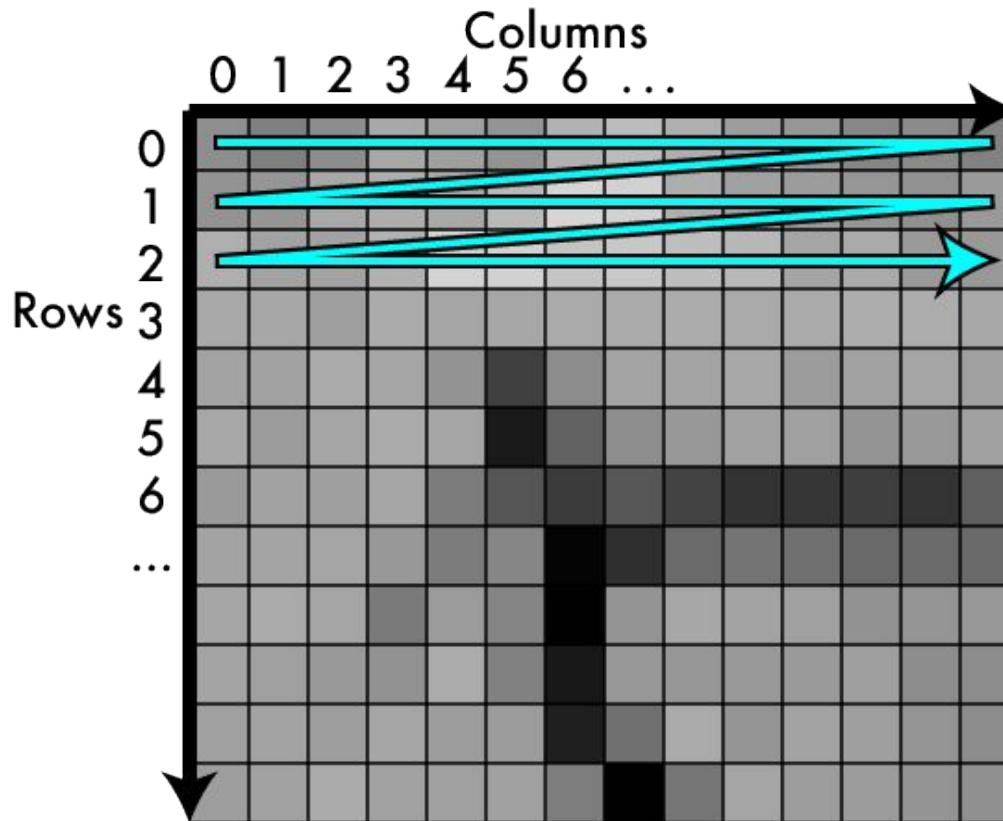
Как хранить изображения



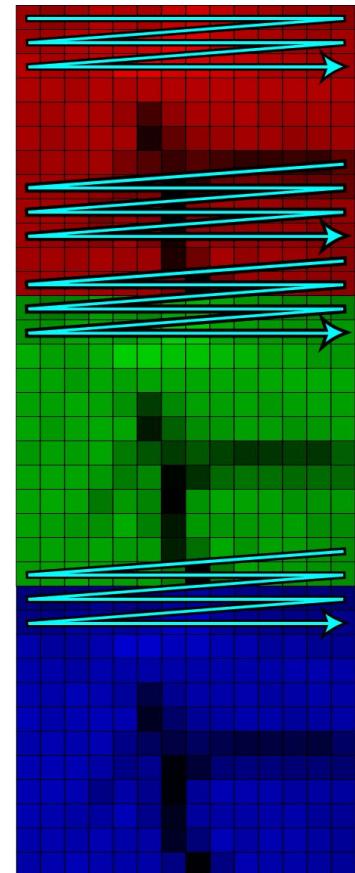
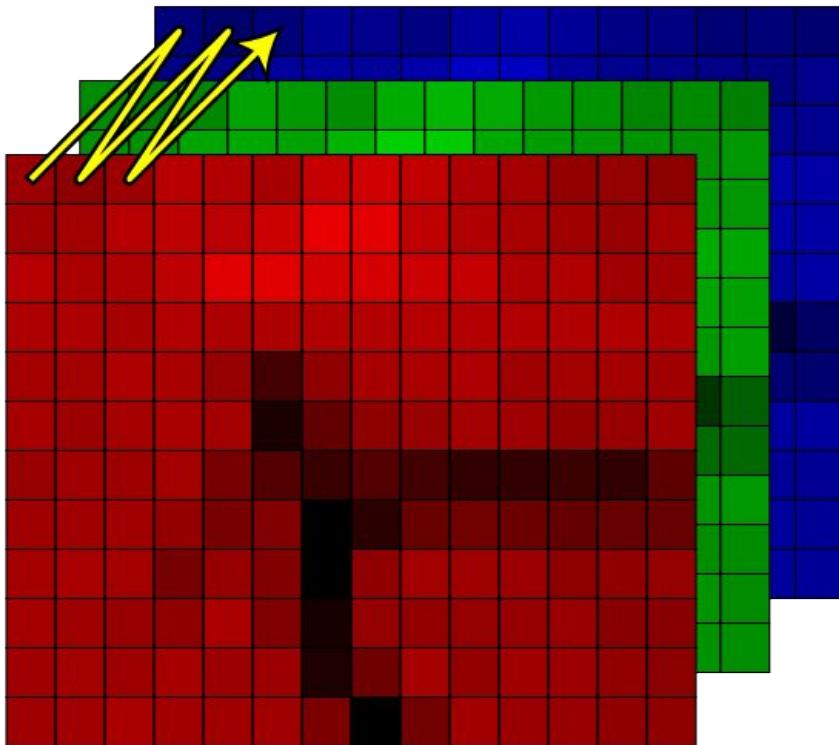
Хранение: row major vs column major



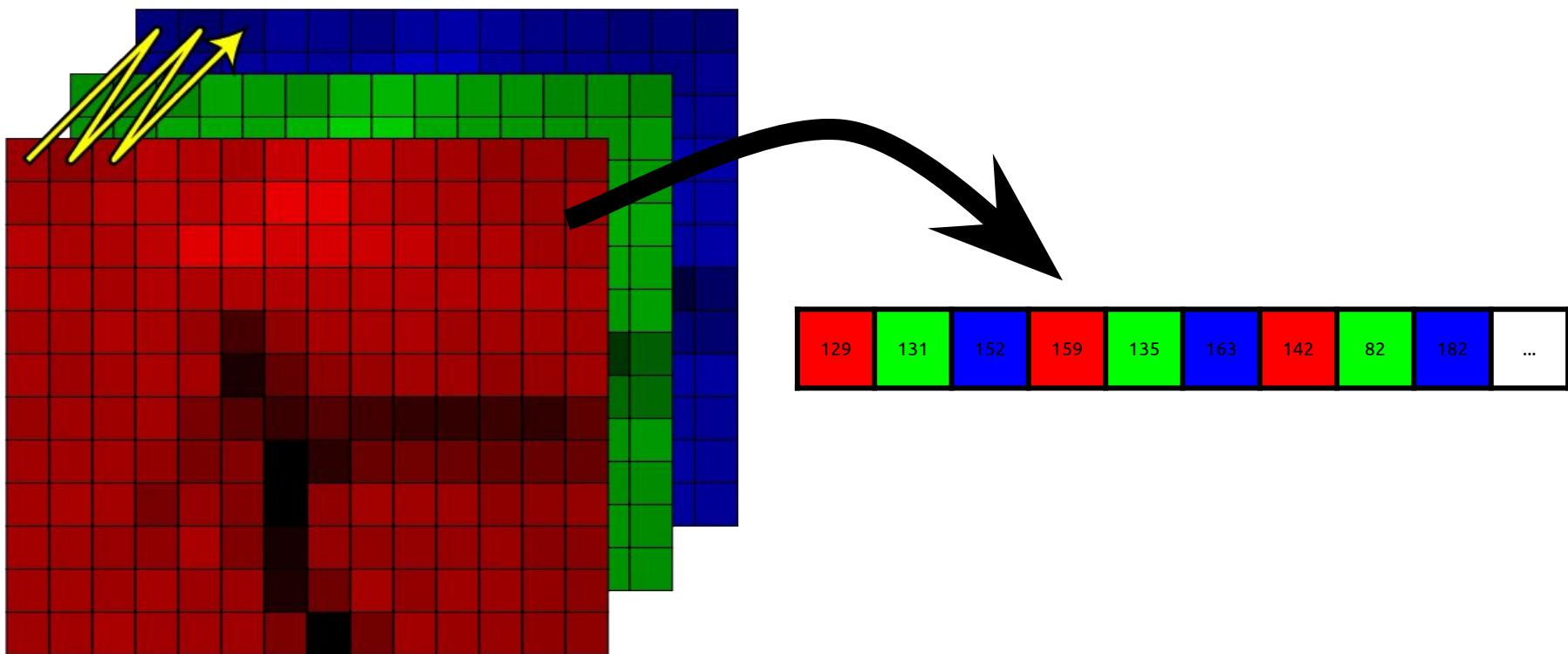
Обычно используется row major



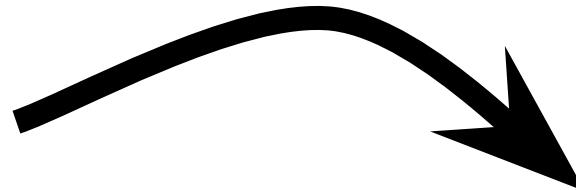
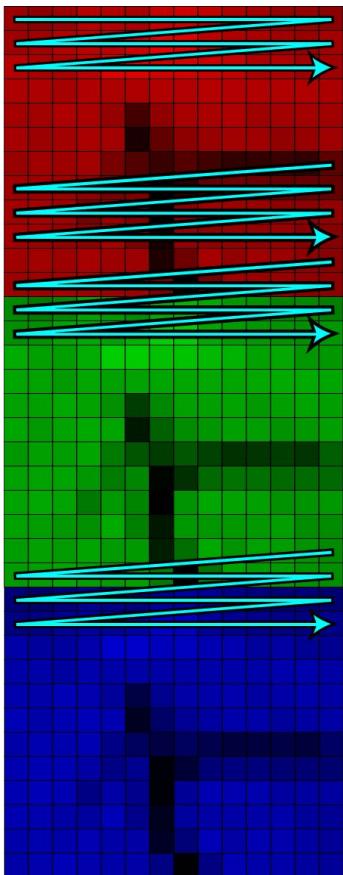
В 3d больше выбора!



HWС: каналы чередуются

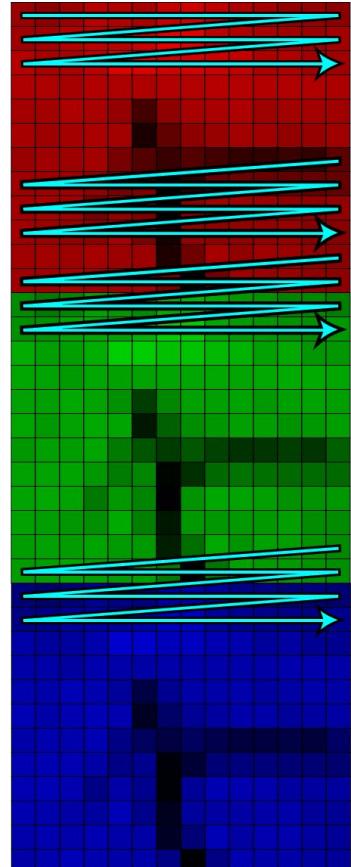


CHW: каналы разделяются



CHW: квиз

В массиве, соответствующем изображению размера, 1920 x 1080 x 3, какой будет индекс у пикселя (15,192,2)?



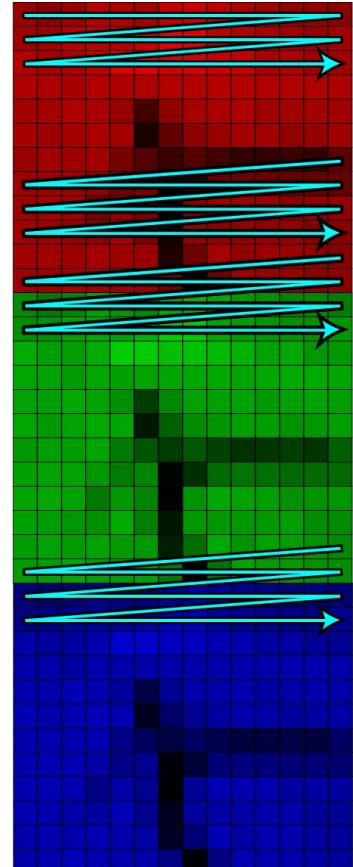
CHW: квиз

В массиве, соответствующем изображению размера, 1920 x 1080 x 3, какой будет индекс у пикселя (15,192,2)?

В общем случае: пиксель (x,y,z) изображения WxHxC имеет индекс

$$x + y * W + z * W * H$$

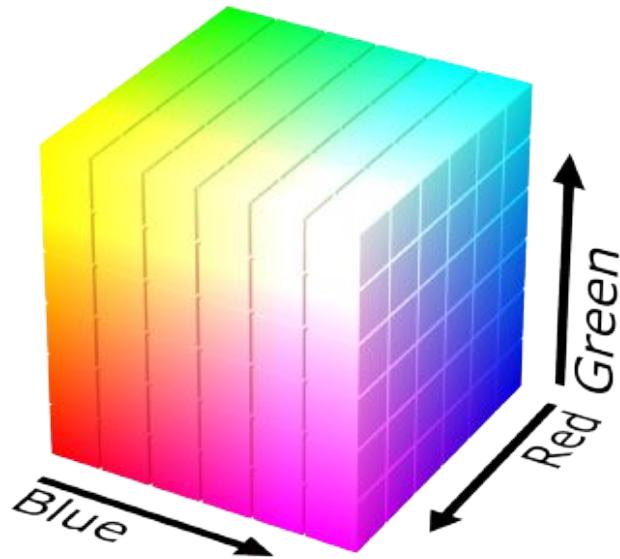
$$15 + 192 * 1920 + 2 * 1920 * 1080 = 4,515,855$$



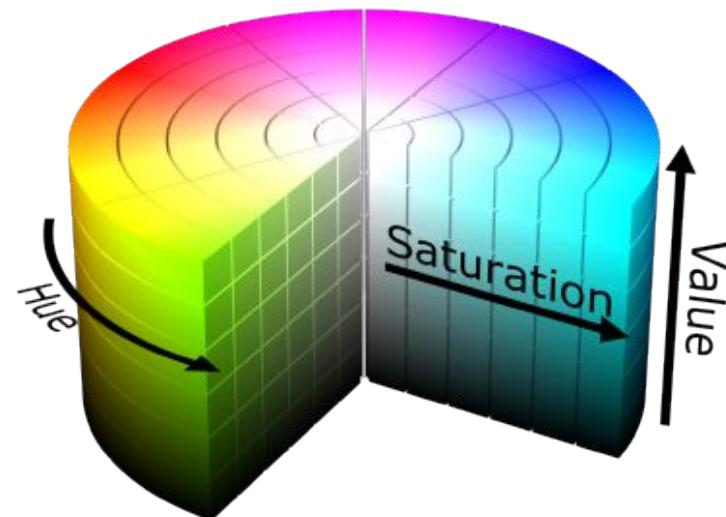


Еще немного про цветовые
пространства

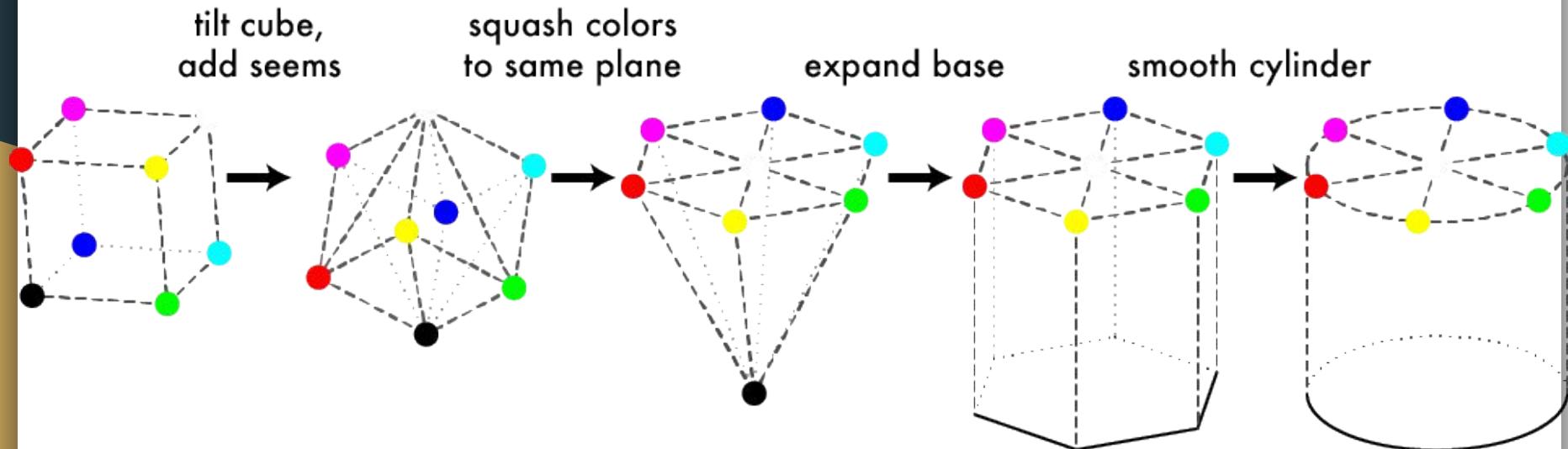
RGB



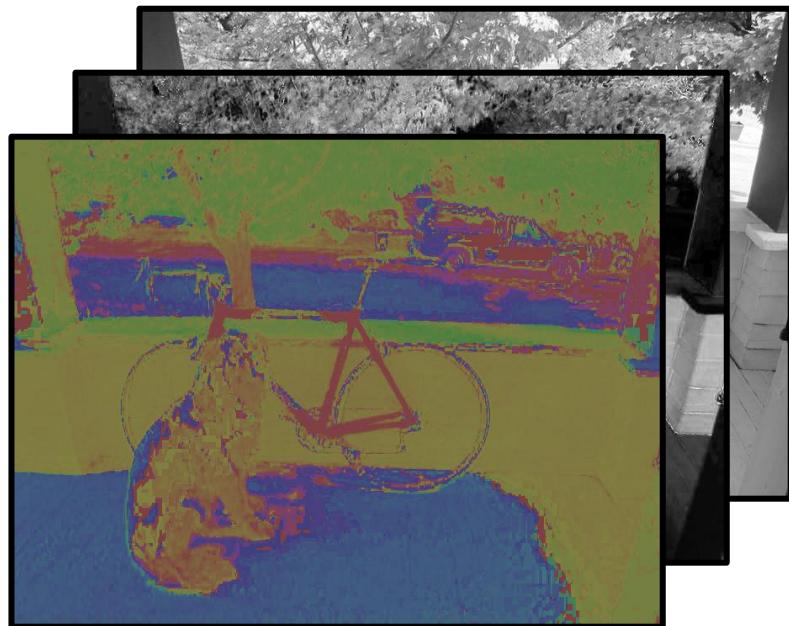
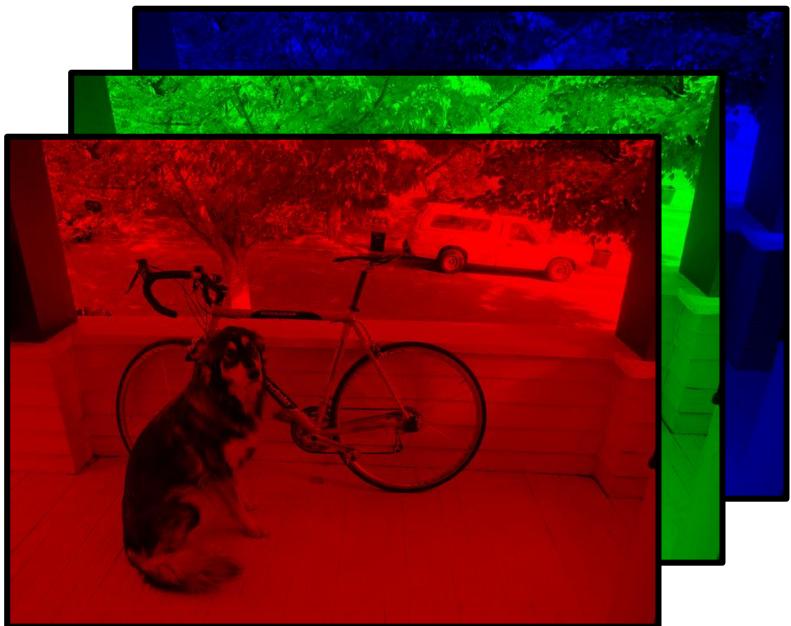
HSV



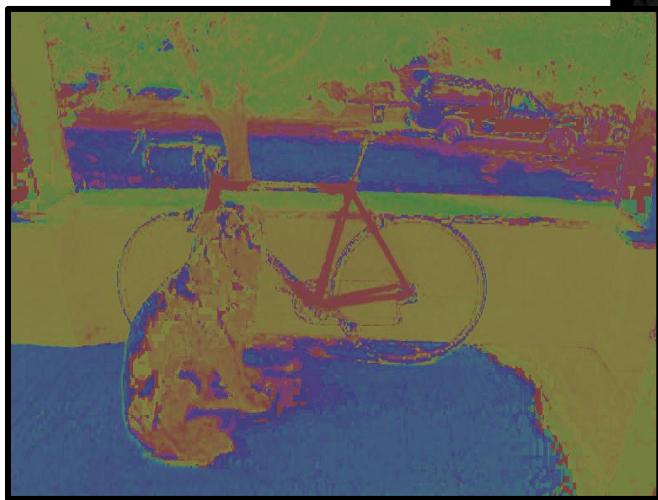
Геометрические преобразования: HSV2RGB



Все еще 3D тензор, другая информация



Hue



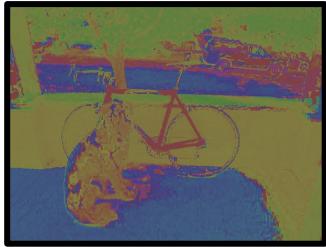
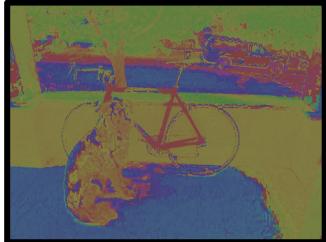
Saturation



Value



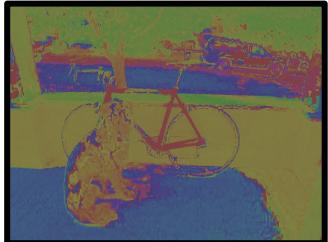
Больше Saturation = более насыщенные цвета



2x



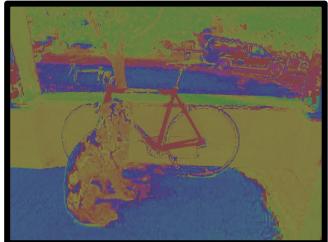
Больше Value = изображение ярче



2x



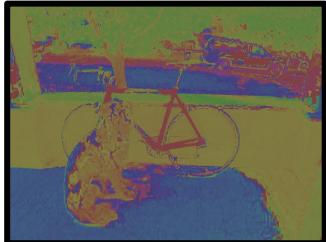
Смещение Ние = смещение цветов



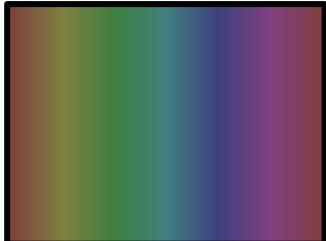
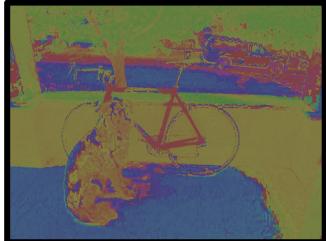
- .2



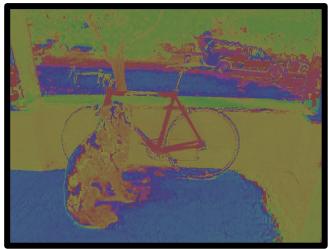
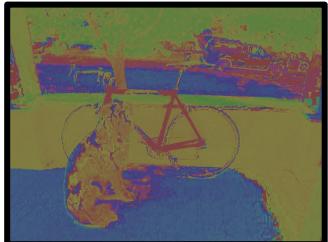
Монотонный цвет



Или узор...



Пороговый Saturation





Интерполяция изображений и ресайзинг

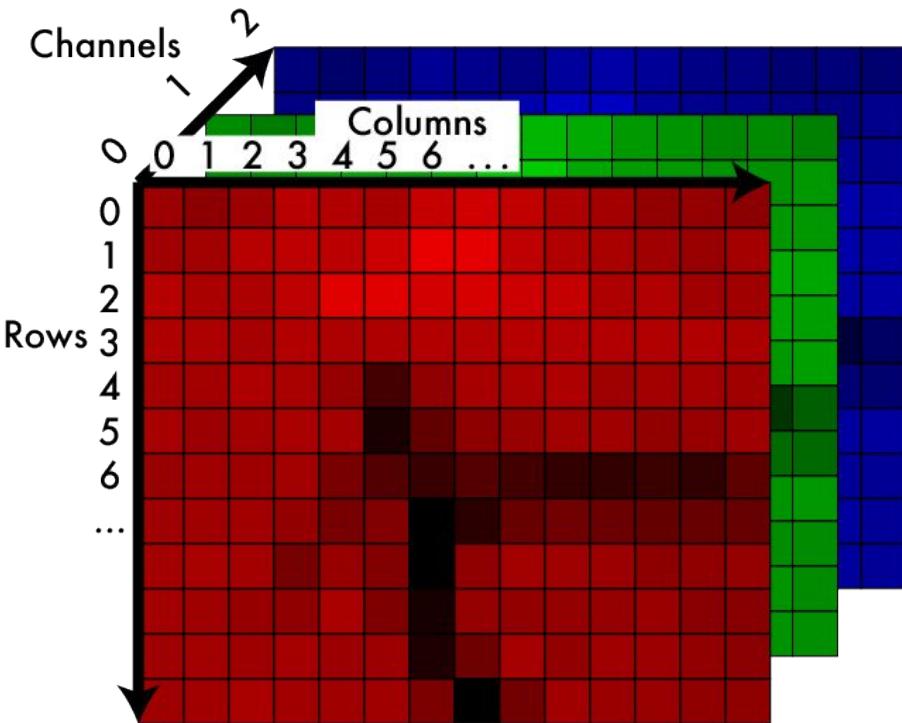
Изображение - функция!

Изображение - функция из
пространства индексов в
пространство значений
пикселей:

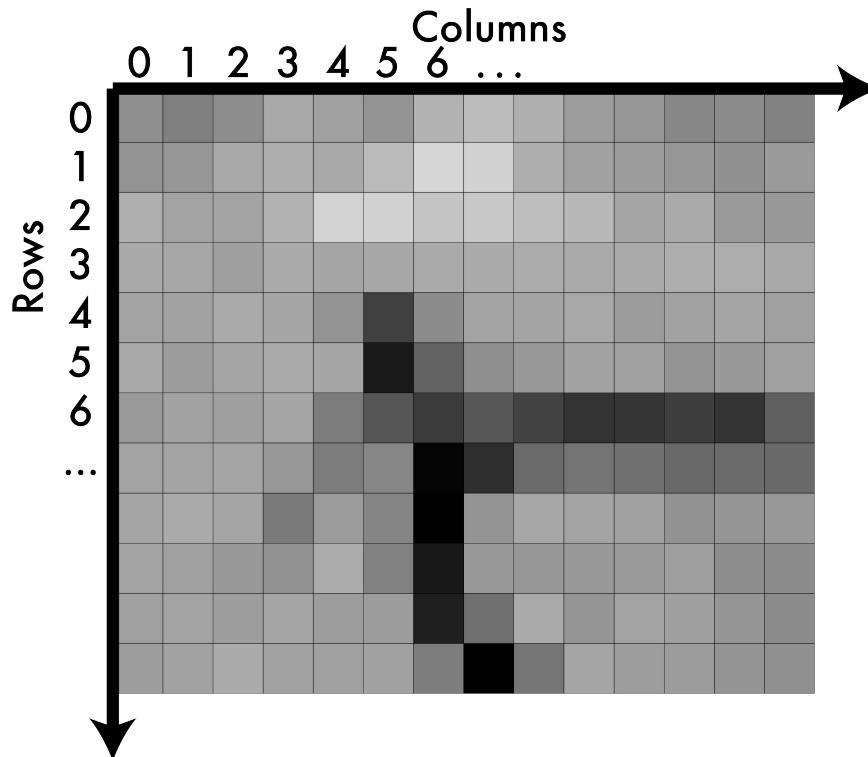
- $\text{Im}: I \times I \times I \rightarrow R$

Мы бы хотели получить
функцию, принимающую
вещественные значения

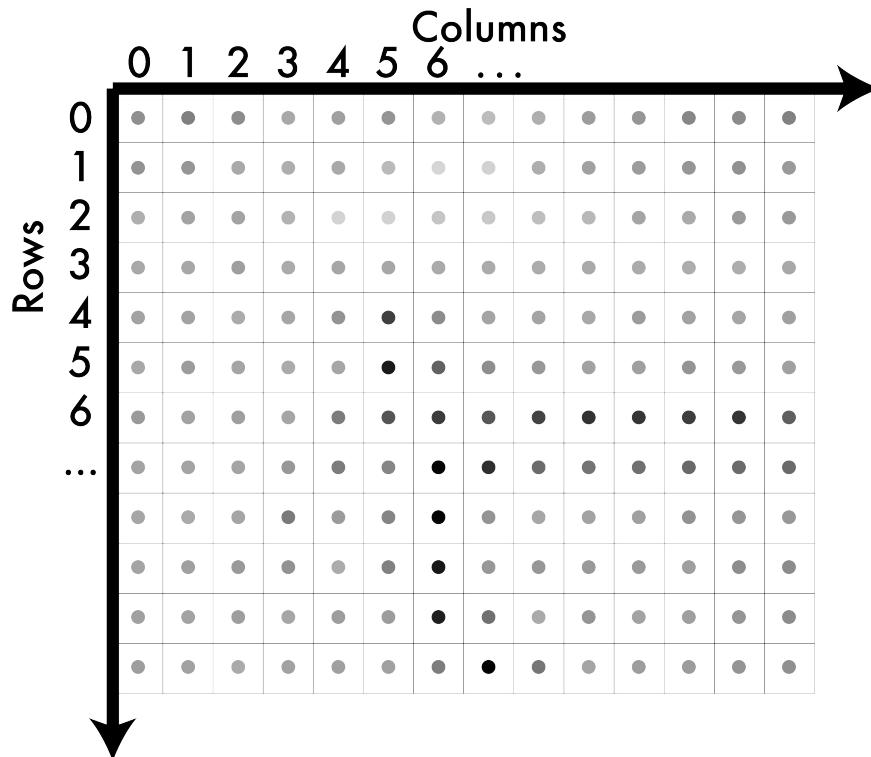
- $\text{Im}': R \times R \times I \rightarrow R$



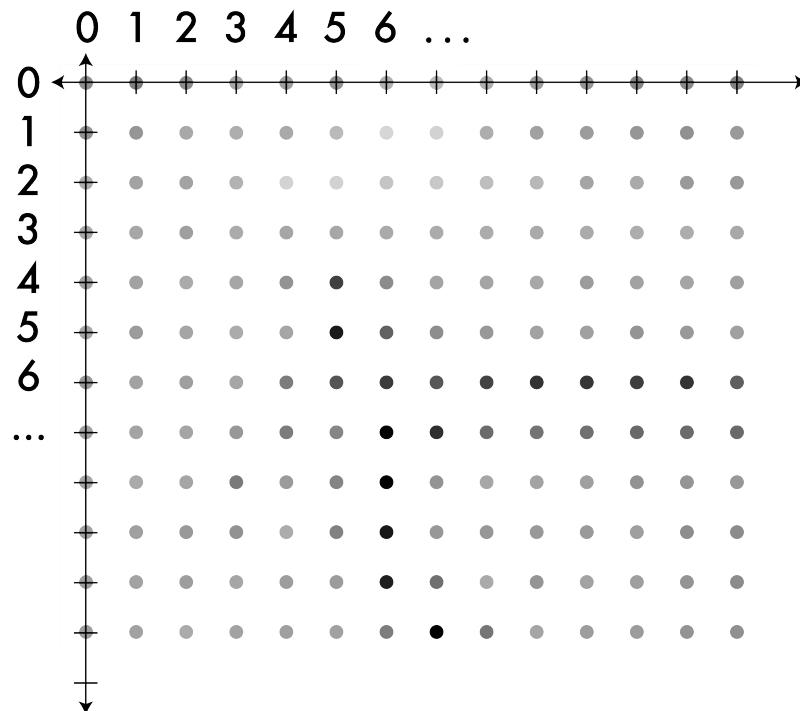
Координаты изображения



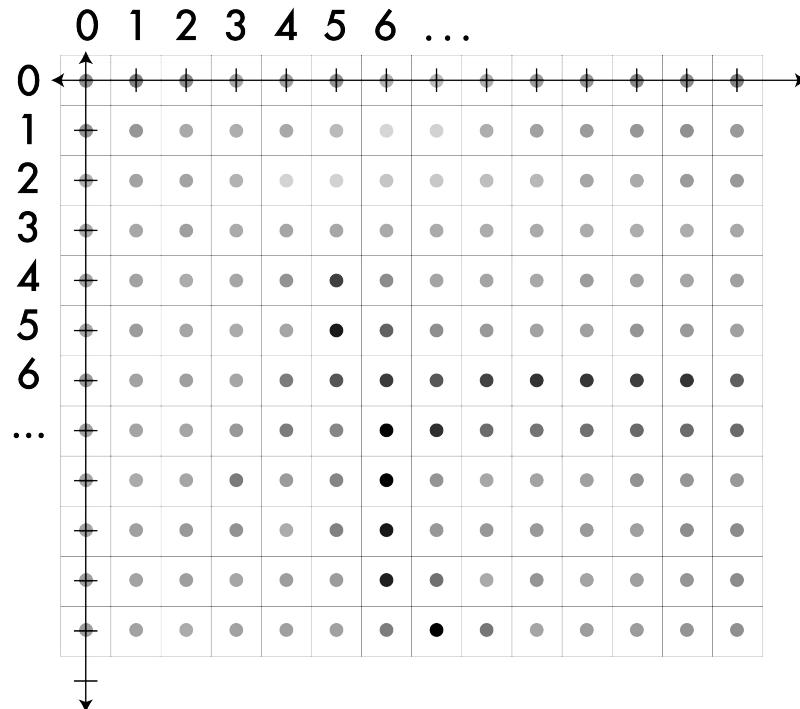
Координаты изображения



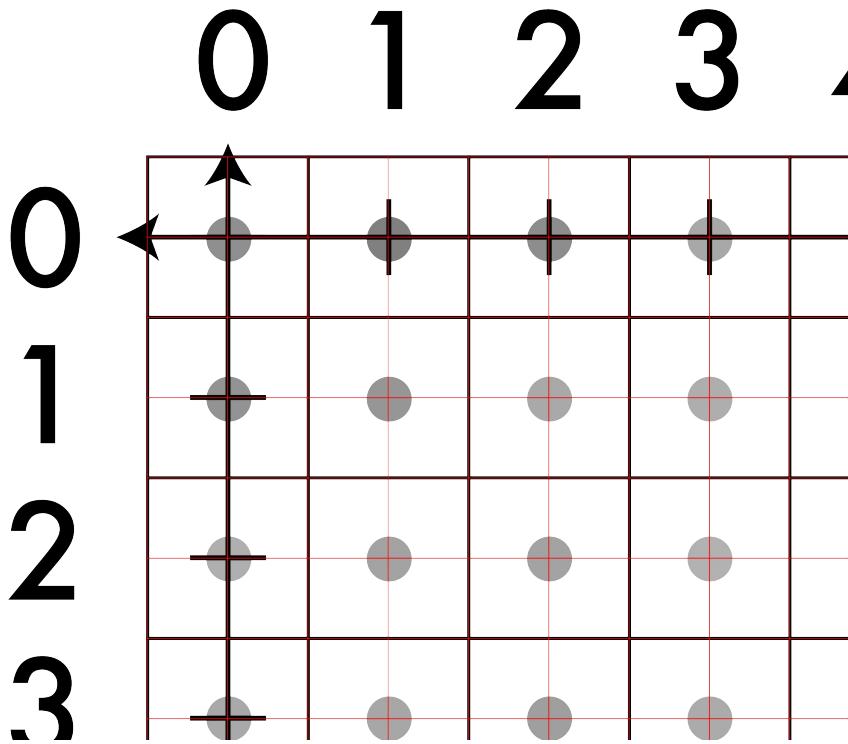
Координаты изображения



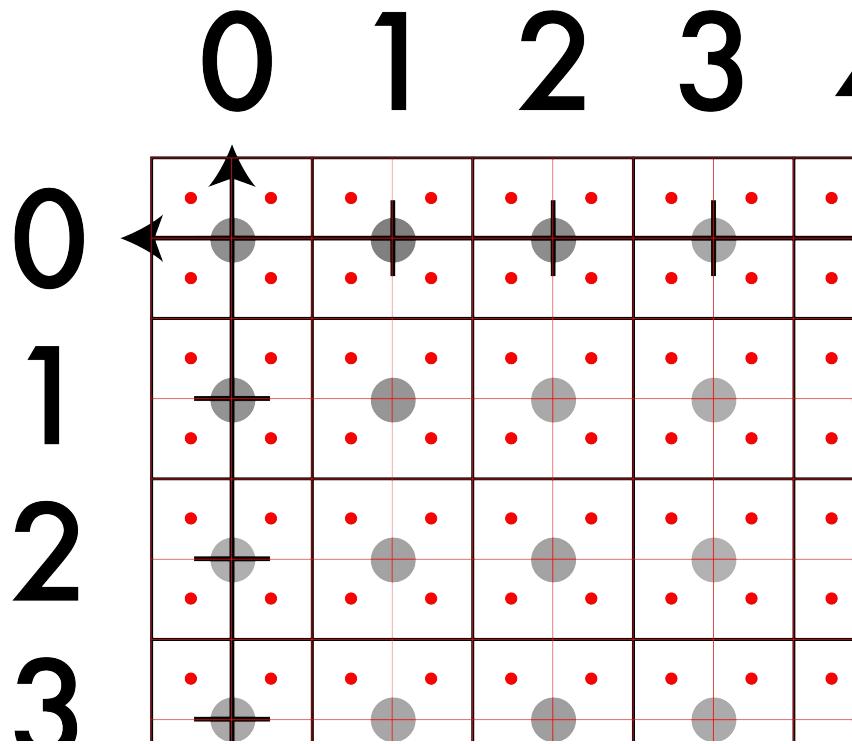
Координаты изображения



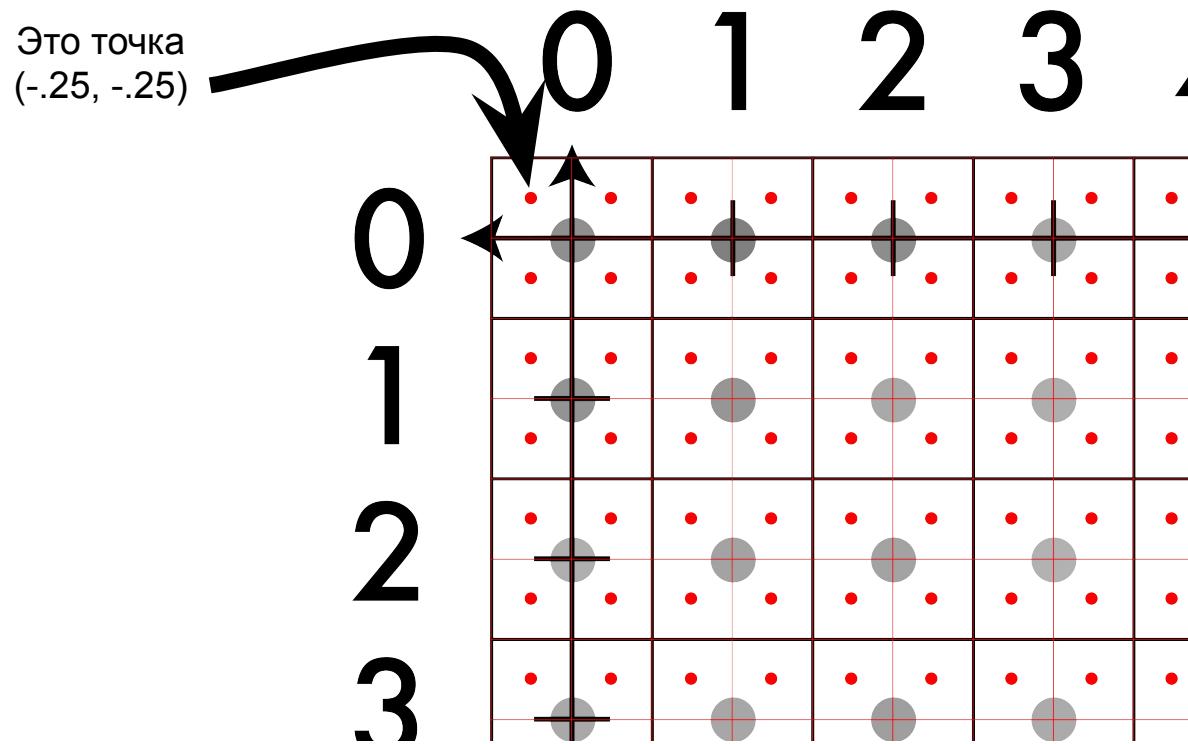
Координаты изображения



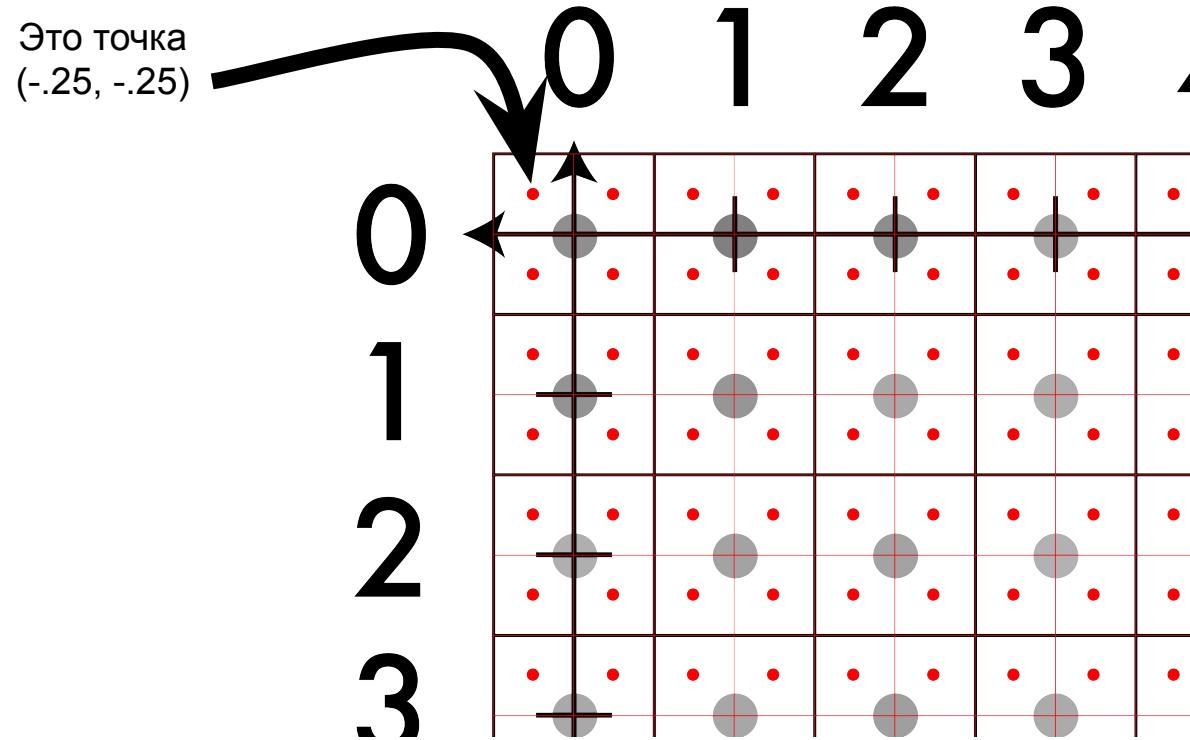
Координаты изображения



Координаты изображения



Нужно быть осторожным, легко запутаться!



Nearest neighbor интерполяция

$f(x,y,z) = \text{Im}(\text{round}(x), \text{round}(y), z)$

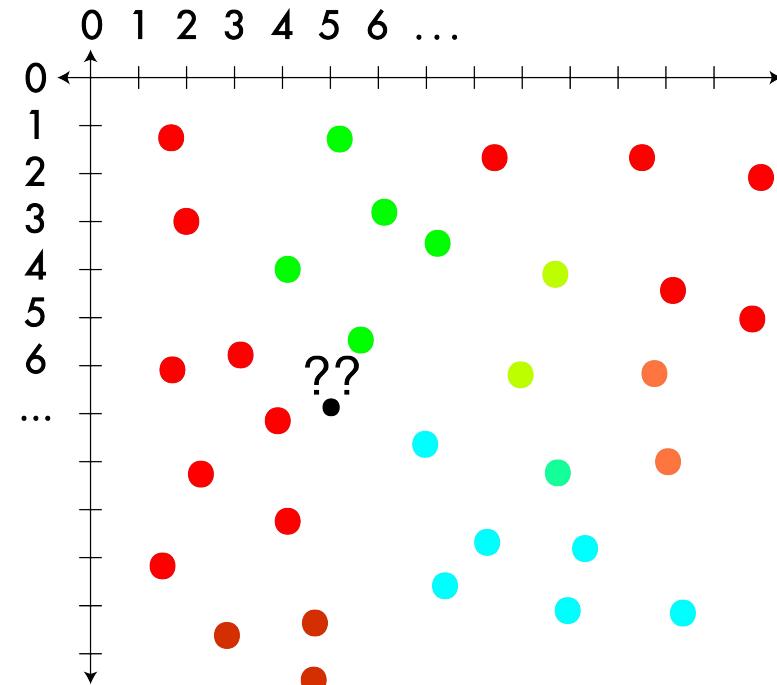
- “Лестничные” артефакты
- z - все еще Integer



Triangle interpolation: менее структурированные изображения

Иногда бывает не
прямоугольная,
повторяющаяся сетка, а
иррегулярный паттерн
пикселей

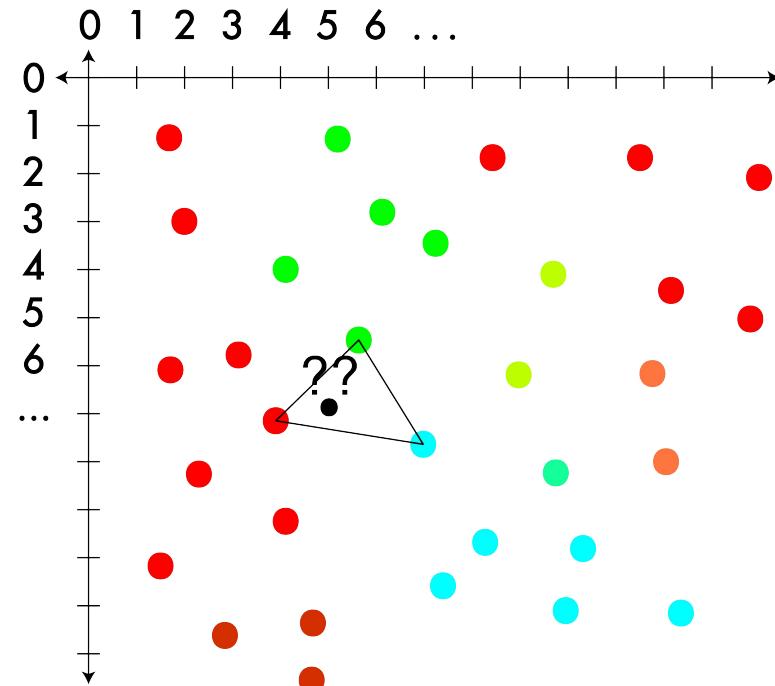
Будем использовать
треугольники!



Triangle interpolation: менее структурированные изображения

Иногда бывает не
прямоугольная,
повторяющаяся сетка, а
иррегулярный паттерн
пикселей

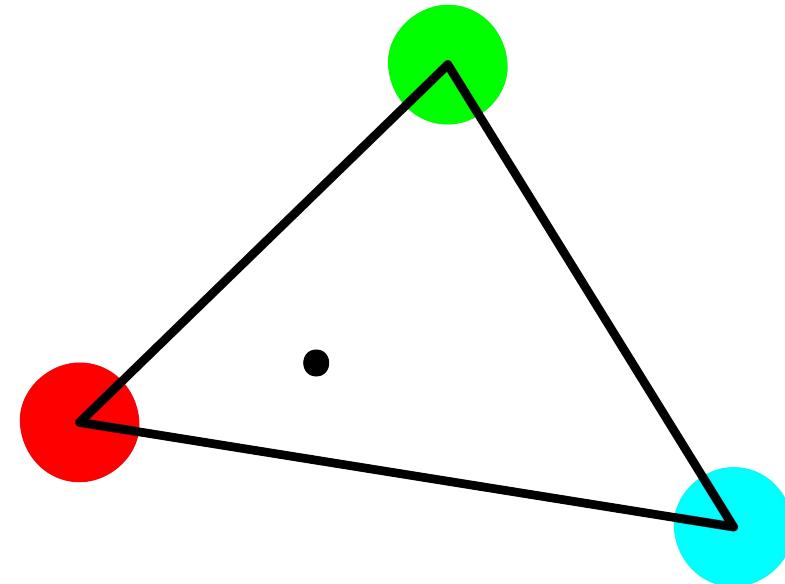
Будем использовать
треугольники!



Triangle interpolation: менее структурированные изображения

Иногда бывает не
прямоугольная,
повторяющаяся сетка, а
иррегулярный паттерн
пикселей

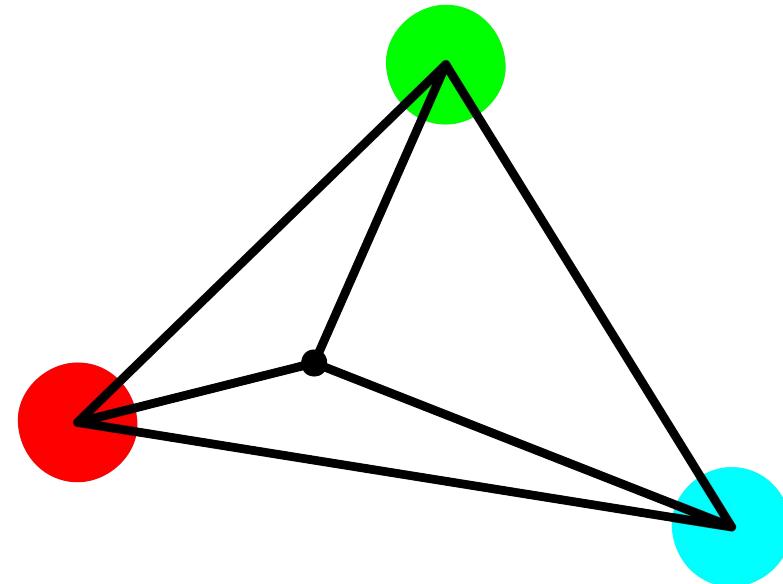
Будем использовать
треугольники!



Triangle interpolation: менее структурированные изображения

Иногда бывает не
прямоугольная,
повторяющаяся сетка, а
иррегулярный паттерн
пикселей

Будем использовать
треугольники!

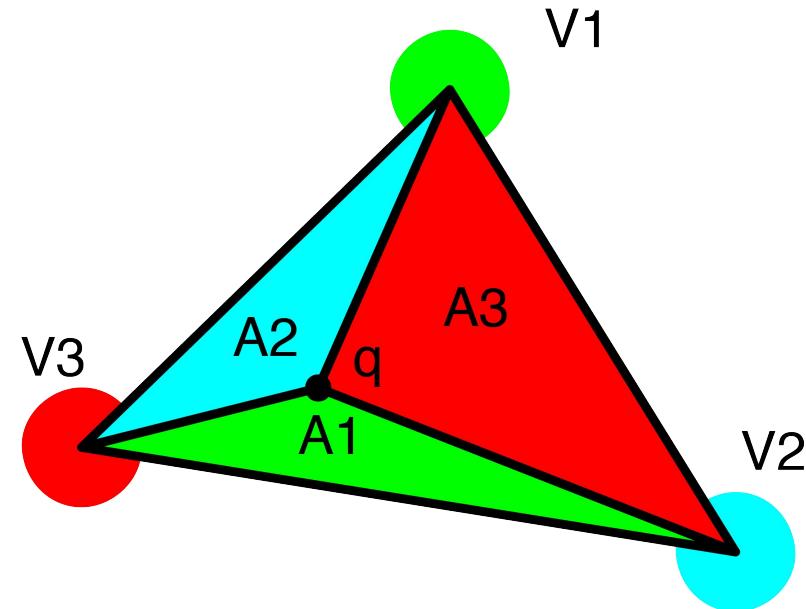


Triangle interpolation: менее структурированные изображения

Взвешенная сумма

$$Q = V1 * A1 + V2 * A2 + V3 * A3$$

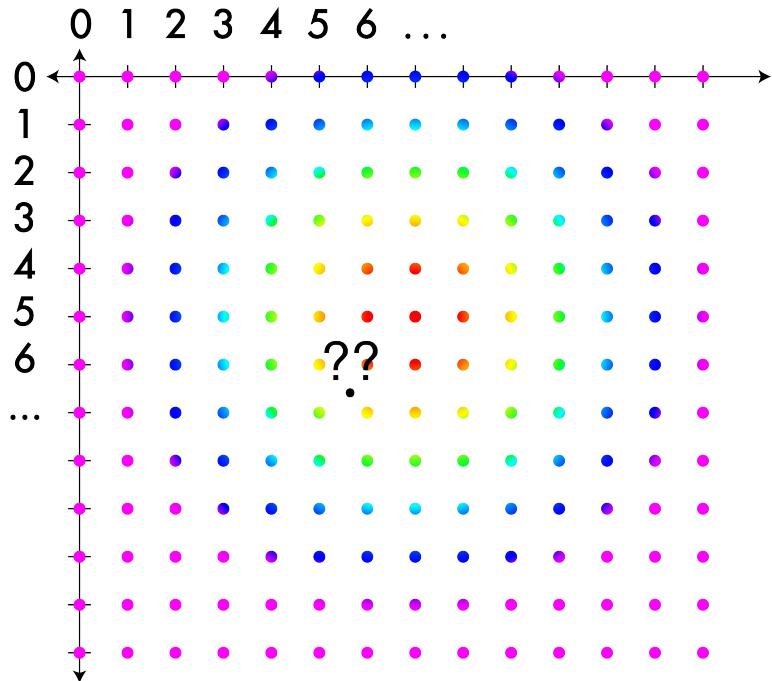
Нормализуем на общую
площадь



Билинейная интерполяция

Прямоугольная сетка

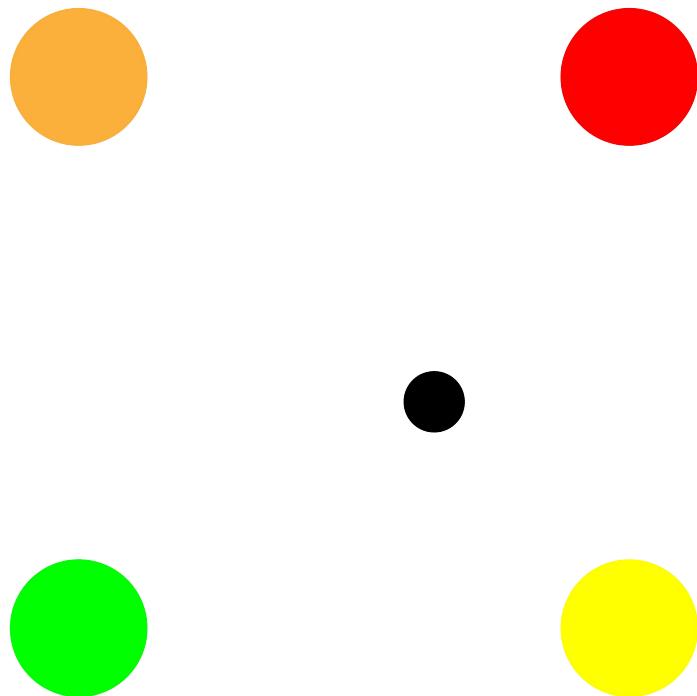
В этот раз будем искать ближайших соседей по прямоугольнику



Билинейная интерполяция

Прямоугольная сетка

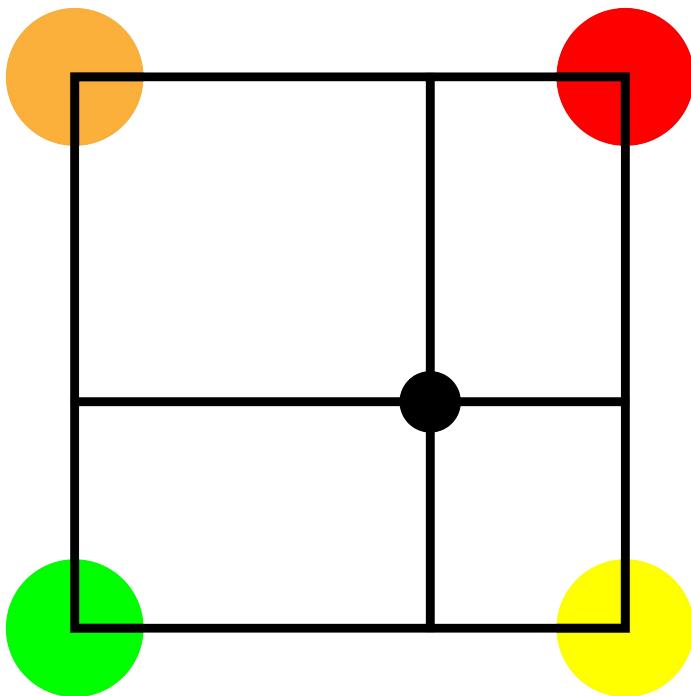
В этот раз будем искать
ближайших соседей по
прямоугольнику



Билинейная интерполяция

Прямоугольная сетка

В этот раз будем искать ближайших соседей по прямоугольнику



Билинейная интерполяция

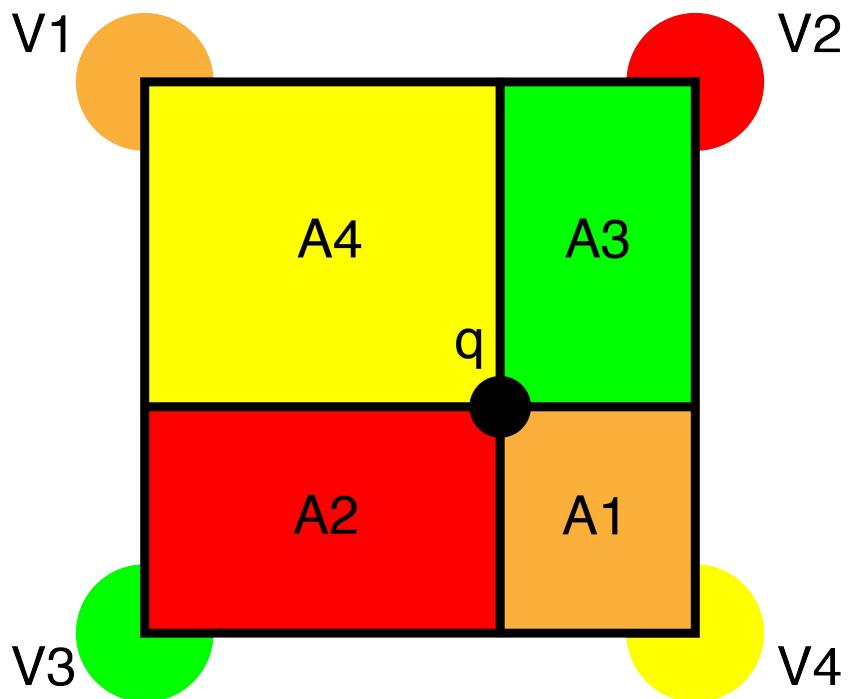
В этот раз будем искать ближайших соседей по прямоугольнику

Идея такая же: взвешенная сумма на основе противолежащих прямоугольников

$$Q = V1 * A1 + V2 * A2 + V3 * A3 + V4 * A4$$

Все еще нужно нормализовывать!

Или не нужно? 🤔



Билинейная интерполяция

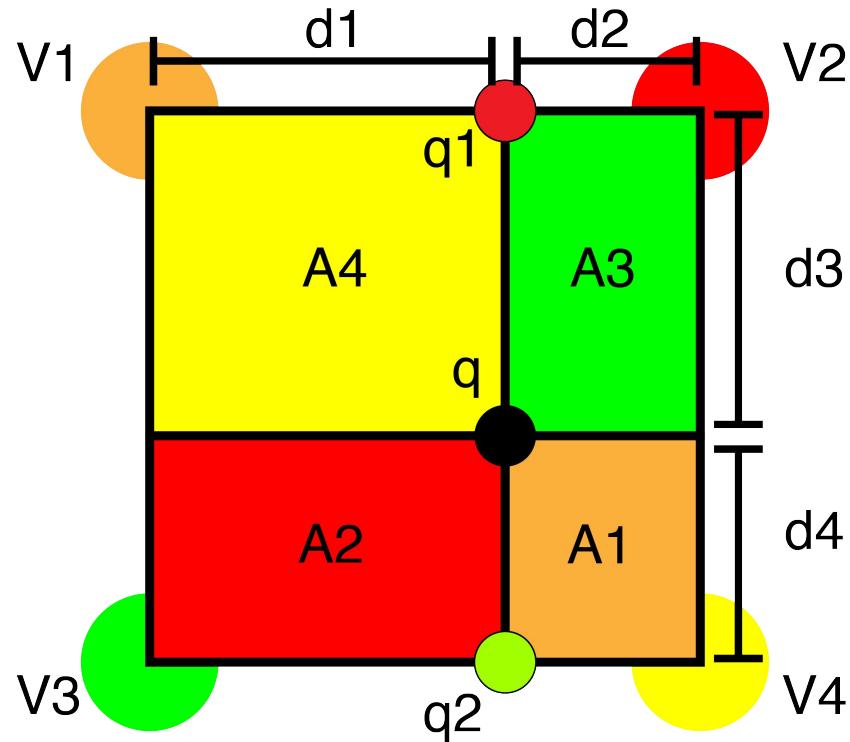
Эквивалентно:

Линейная интерполяция
линейных интерполяций

$$q_1 = V1 * d2 + V2 * d1$$

$$q_2 = V3 * d2 + V4 * d1$$

$$q = q1 * d4 + q2 * d3$$

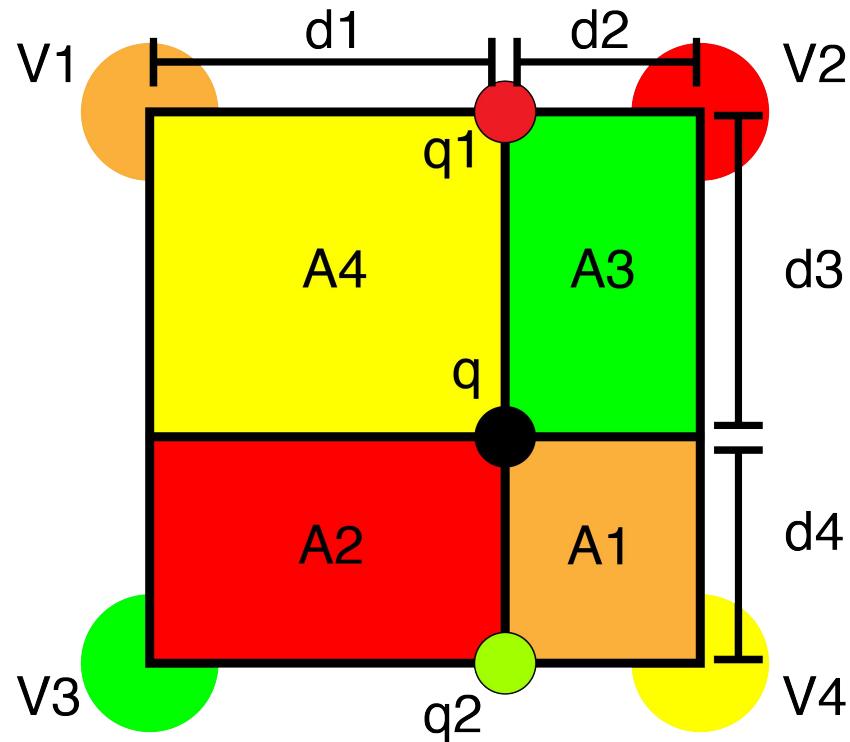


Билинейная интерполяция

$$q_1 = V_1 * d_2 + V_2 * d_1$$

$$q_2 = V_3 * d_2 + V_4 * d_1$$

$$q = q_1 * d_4 + q_2 * d_3$$



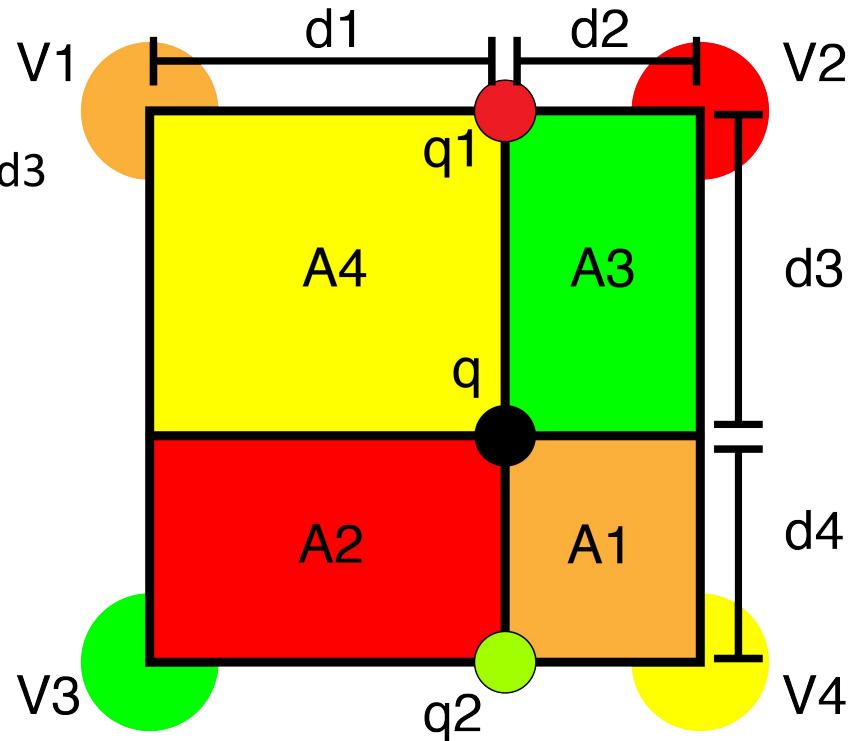
Билинейная интерполяция

$$q_1 = V_1 * d_2 + V_2 * d_1$$

$$q_2 = V_3 * d_2 + V_4 * d_1$$

$$q = q_1 * d_4 + q_2 * d_3$$

$$q = (V_1 * d_2 + V_2 * d_1) * d_4 + (V_3 * d_2 + V_4 * d_1) * d_3$$



Билинейная интерполяция

$$q_1 = V_1 * d_2 + V_2 * d_1$$

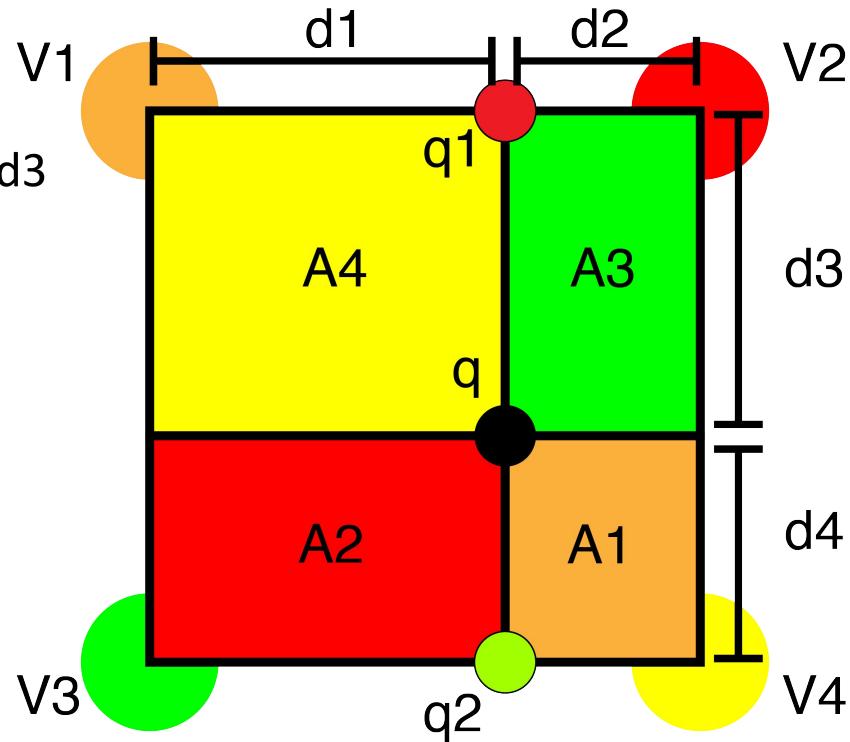
$$q_2 = V_3 * d_2 + V_4 * d_1$$

$$q = q_1 * d_4 + q_2 * d_3$$

$$q = (V_1 * d_2 + V_2 * d_1) * d_4 + (V_3 * d_2 + V_4 * d_1) * d_3$$

$$q = V_1 * d_2 * d_4 + V_2 * d_1 * d_4 + V_3 * d_2 * d_3 +$$

$$V_4 * d_1 * d_3$$



Билинейная интерполяция

$$q_1 = V1 * d2 + V2 * d1$$

$$q_2 = V3 * d2 + V4 * d1$$

$$q = q_1 * d4 + q_2 * d3$$

$$q = (V1 * d2 + V2 * d1) * d4 + (V3 * d2 + V4 * d1) * d3$$

$$q = V1 * d2 * d4 + V2 * d1 * d4 + V3 * d2 * d3 +$$

$$V4 * d1 * d3$$

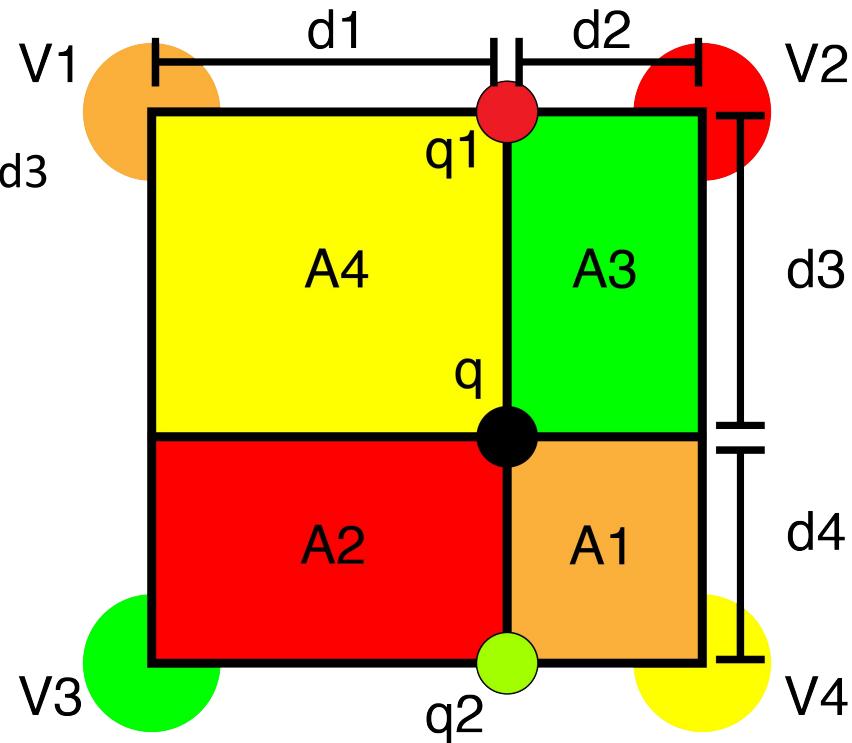
Помним:

$$A1 = d2 * d4$$

$$A2 = d1 * d4$$

$$A3 = d2 * d3$$

$$A4 = d1 * d3$$



Билинейная интерполяция

$$q_1 = V1 * d2 + V2 * d1$$

$$q_2 = V3 * d2 + V4 * d1$$

$$q = q_1 * d4 + q_2 * d3$$

$$q = (V1 * d2 + V2 * d1) * d4 + (V3 * d2 + V4 * d1) * d3$$

$$q = V1 * d2 * d4 + V2 * d1 * d4 + V3 * d2 * d3 + V4 * d1 * d3$$

Помним:

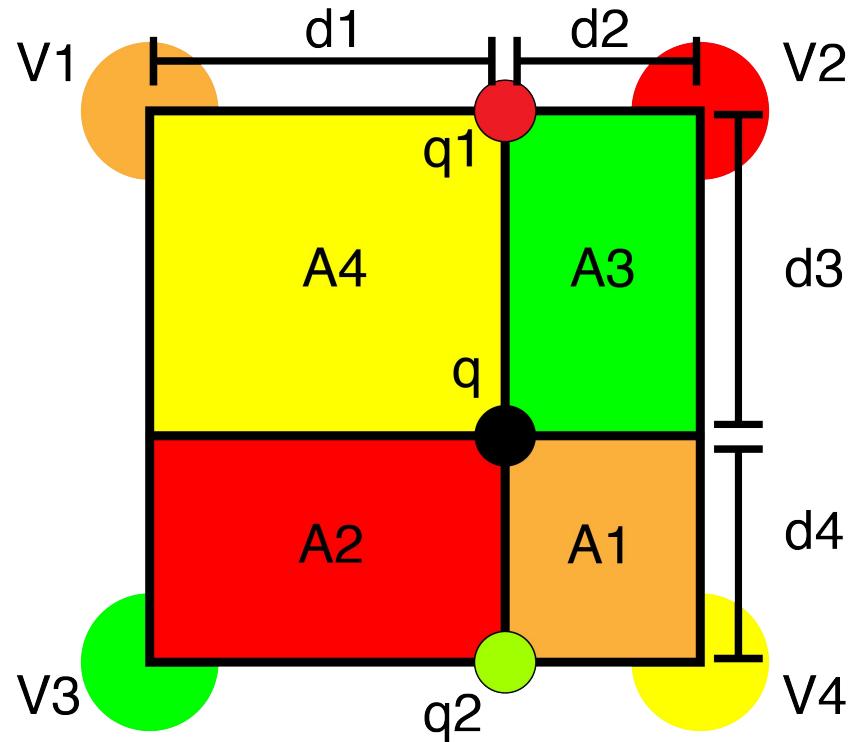
$$A1 = d2 * d4$$

$$A2 = d1 * d4$$

$$A3 = d2 * d3$$

$$A4 = d1 * d3$$

$$q = V1 * A1 + V2 * A2 + V3 * A3 + V4 * A4$$



Билинейная интерполяция

$$q_1 = V1 * d2 + V2 * d1$$

$$q_2 = V3 * d2 + V4 * d1$$

$$q = q_1 * d4 + q_2 * d3$$

$$q = (V1 * d2 + V2 * d1) * d4 + (V3 * d2 + V4 * d1) * d3$$

$$q = V1 * d2 * d4 + V2 * d1 * d4 + V3 * d2 * d3 + V4 * d1 * d3$$

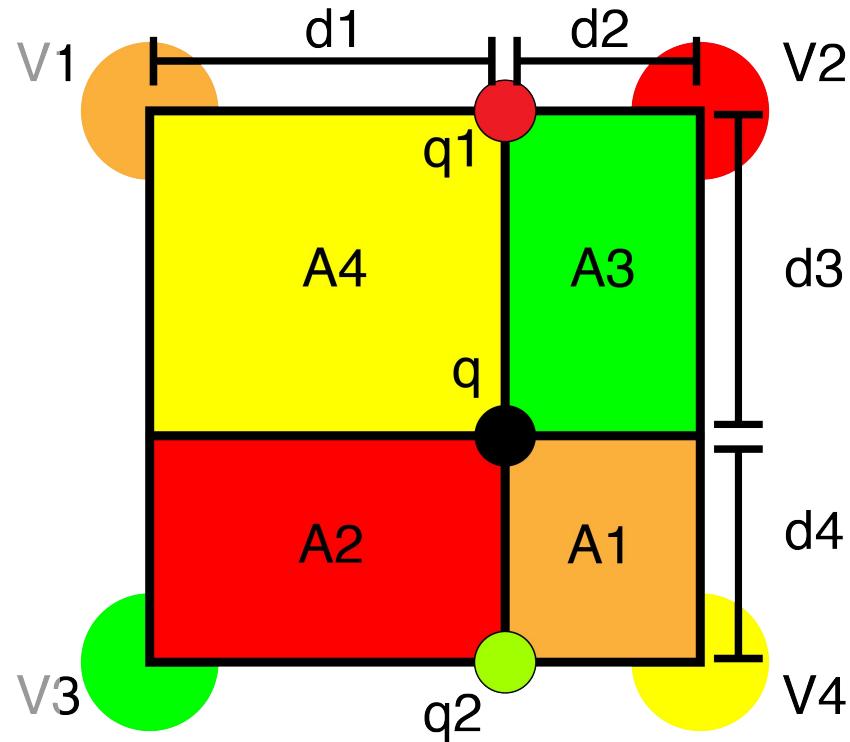
Помним:

$$A1 = d2 * d4$$

$$A2 = d1 * d4$$

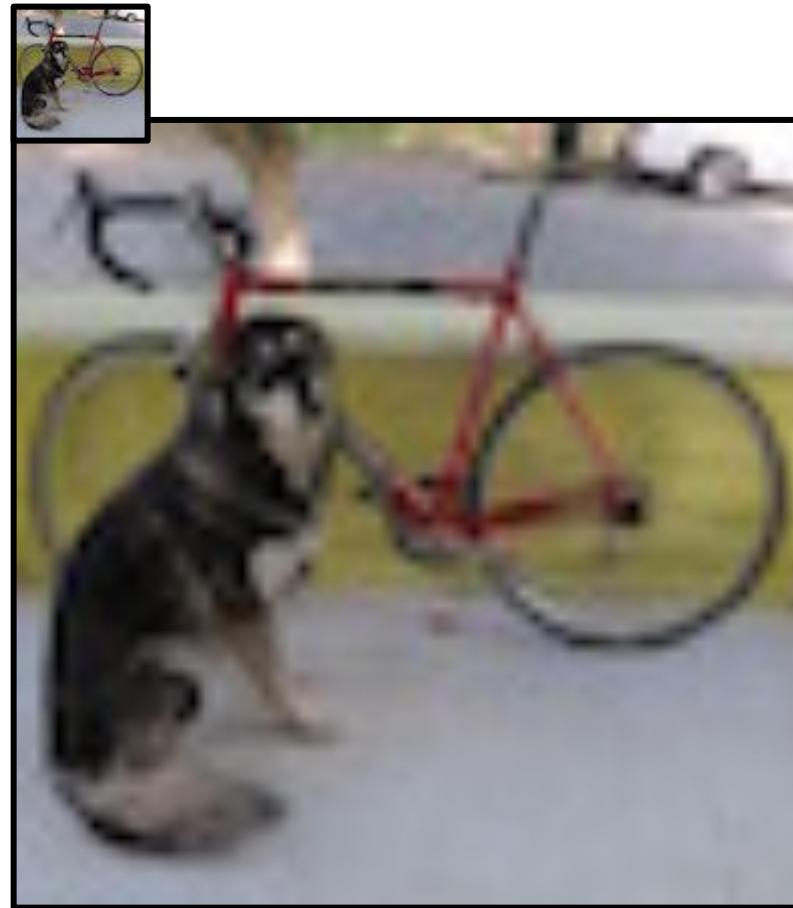
$$A3 = d2 * d3$$

$$A4 = d1 * d3$$

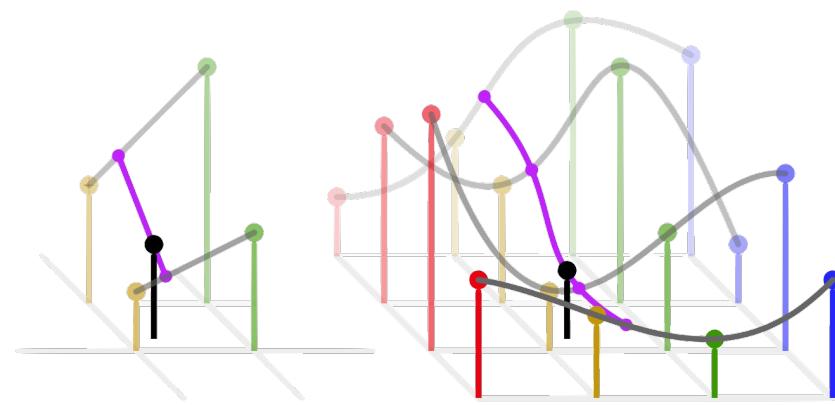


$$q = V1 * A1 + V2 * A2 + V3 * A3 + V4 * A4 \text{ (ура!)}$$

Билинейная интерполяция



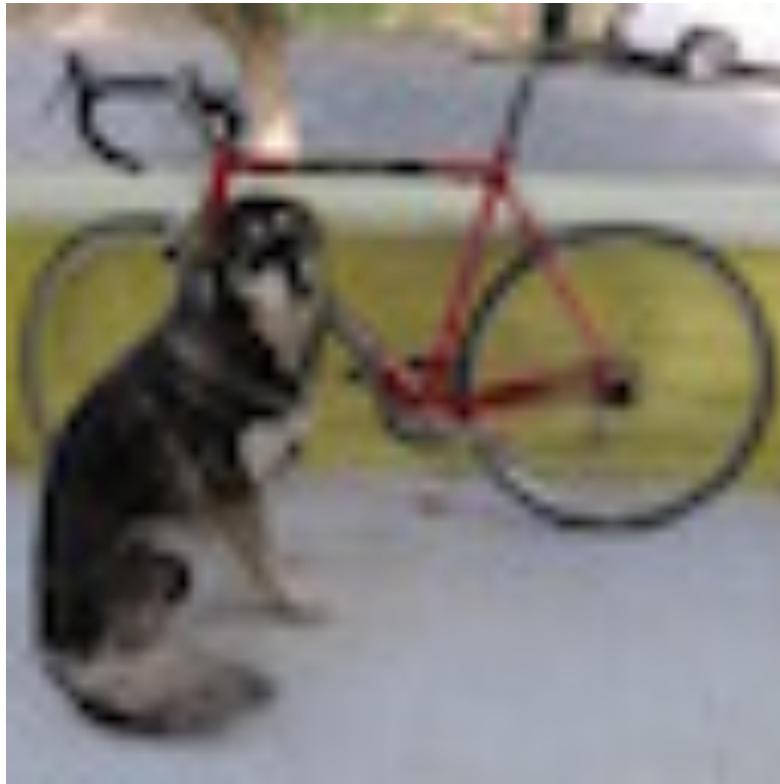
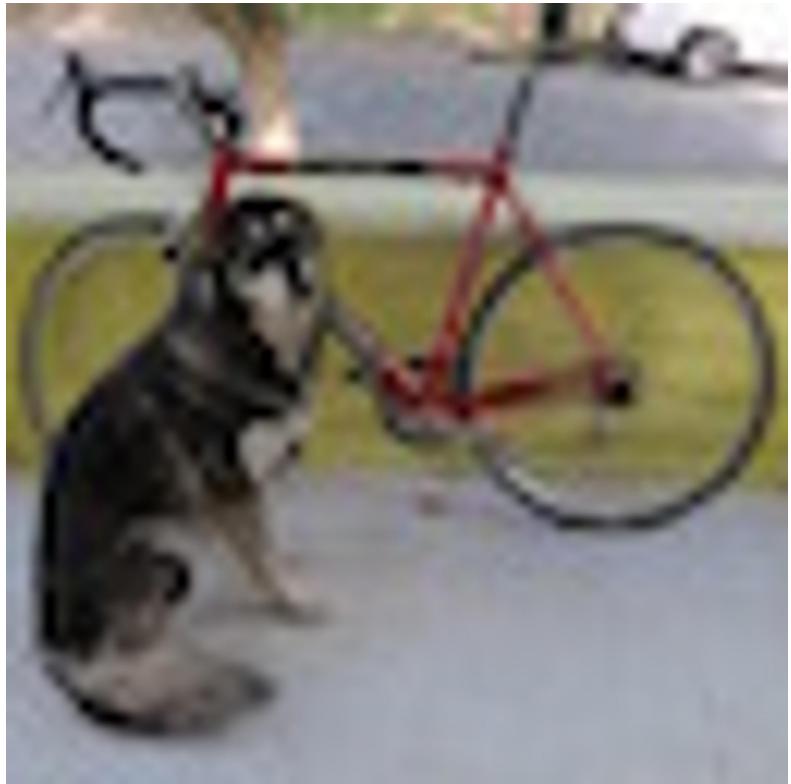
Бикубическая интерполяция



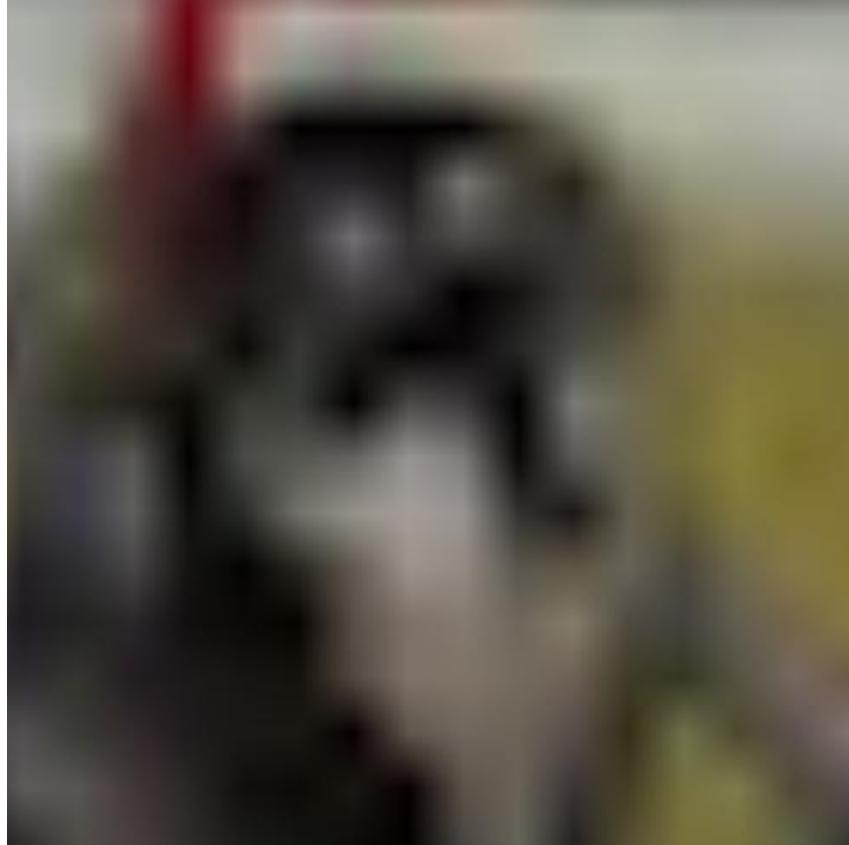
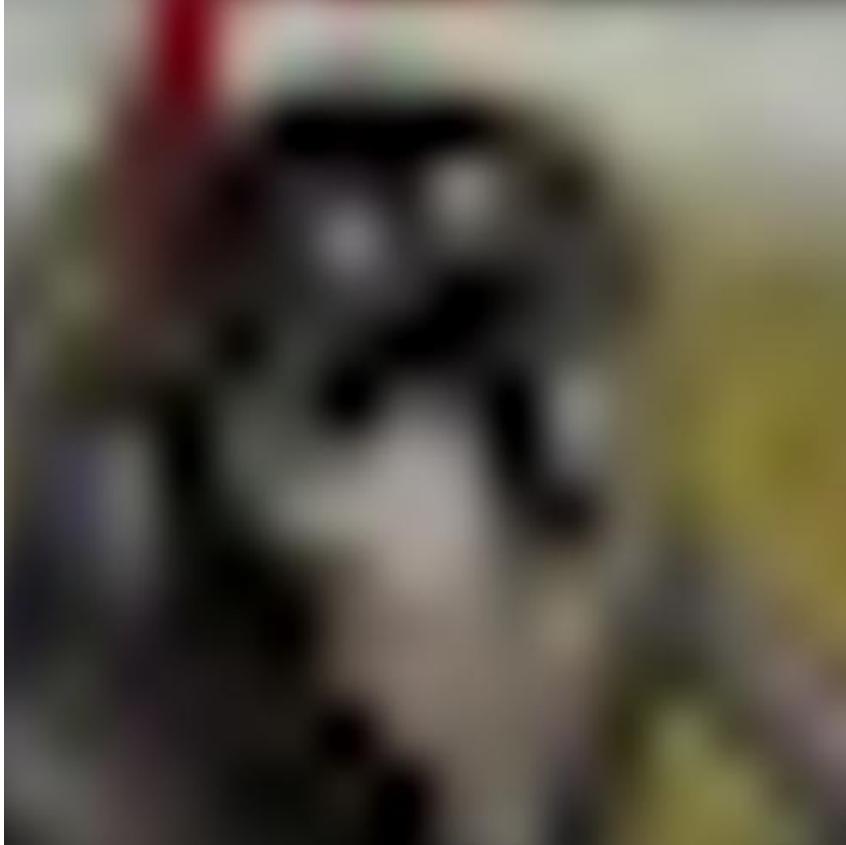
Bilinear

Bicubic

Бикубическая vs билинейная

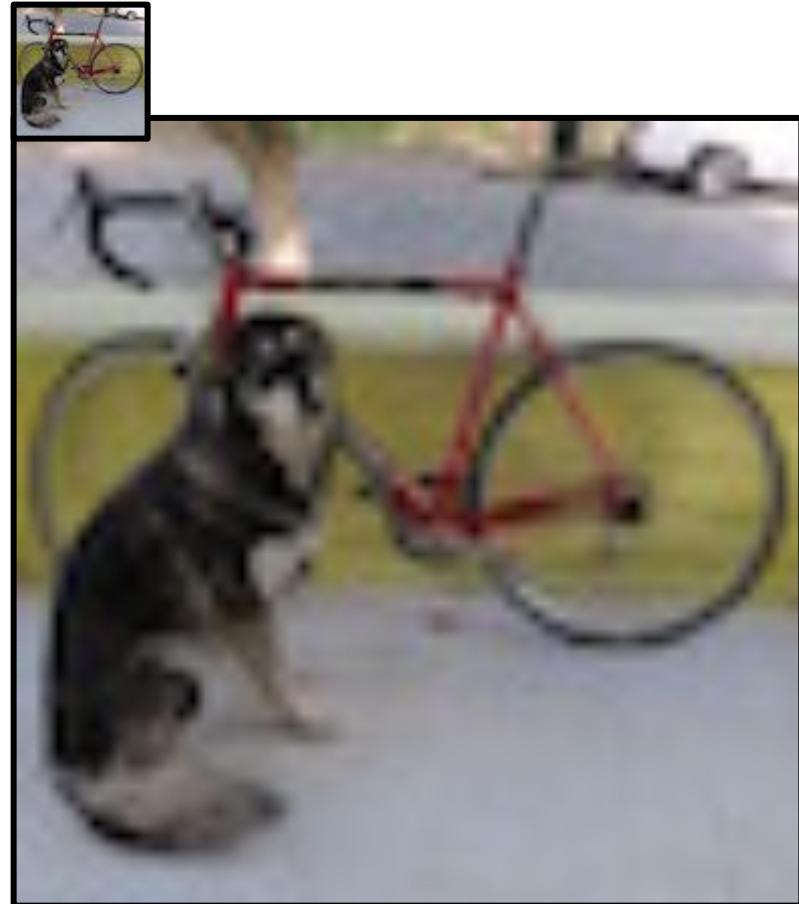


Бикубическая vs билинейная



Алгоритм ресайзинга:

- Для каждого пикселя в новом изображении
 - Отобразить в координаты старого изображения
 - Интерполировать значения
 - Присвоить значение в новом изображении



А как уменьшать размер?

- NN и Bilinear смотрят только на небольшую область
- Большое количество артефактов
- “Лестничные” узоры по диагонали
- Это можно пофиксить с помощью фильтров!





It's coding time!