

Компьютерное зрение. Яркие преобразования

Лекция 2

Лектор: Демяненко Я.М.,
ФИИТ, 2026



От лекции 1 к лекции 2

Вспомним:

- CV — это извлечение СМЫСЛА из пикселей
- Первый уровень — LOW-LEVEL VISION
- Работаем с изображением → получаем изображение

Сегодня:

- Как улучшить изображение?
- Как исправить цвет и яркость?
- Как подготовить данные для анализа?



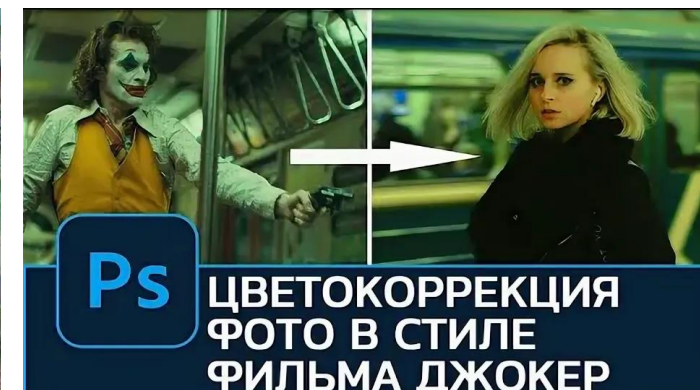
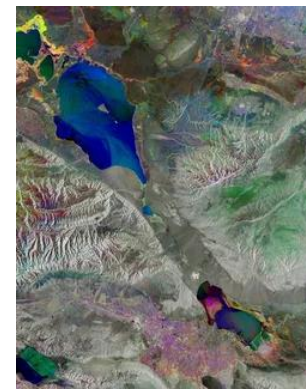
Почему это важно?

Без яркостных преобразований:

- ✗ Тёмное фото — не видно деталей
- ✗ Медицинский снимок — низкий контраст
- ✗ Спутниковый снимок — искажённые цвета

Основная задача:

- ✓ Улучшить визуальное восприятие
- ✓ Привести изображение к виду, удобному для машинного анализа



Цветовая коррекция изображений

Проблема: цветное освещение (фиолетовый свет)

Решение: компенсация цветового баланса

Причины:

- Неверное цветовосприятие камеры
- Цветное освещение (лампы, окна)



Пример — яркостные преобразования в киноиндустрии



Один и тот же кадр → разное настроение

Что такое яркостное преобразование?

Яркостное преобразование — это операция, которая ставит в соответствие каждому пикселю исходного изображения с интенсивностью f новый пиксель с интенсивностью g по определенному правилу:

$$g(x, y) = T[f(x, y)],$$

где

T — функция преобразования

$f(x, y)$ — входное изображение

$g(x, y)$ — выходное изображение

Ключевые понятия:

- Гистограмма изображения — распределение яркостей
- LUT — таблица поиска для быстрой реализации

Типы изображений



МНОГОКАНАЛЬНЫЕ (RGB)	ПОЛУТОНОВЫЕ (Gray)	БИНАРНЫЕ (Binary)
[цветное фото]	[ч/б фото]	[ч/б пиксели]
3 канала: R, G, B	1 канал: яркость	1 канал: 0 или 1
24 бита на пиксель	8 бит на пиксель	1 бит на пиксель
16.7 млн цветов	256 градаций серого	2 значения

Обработка цветных изображений

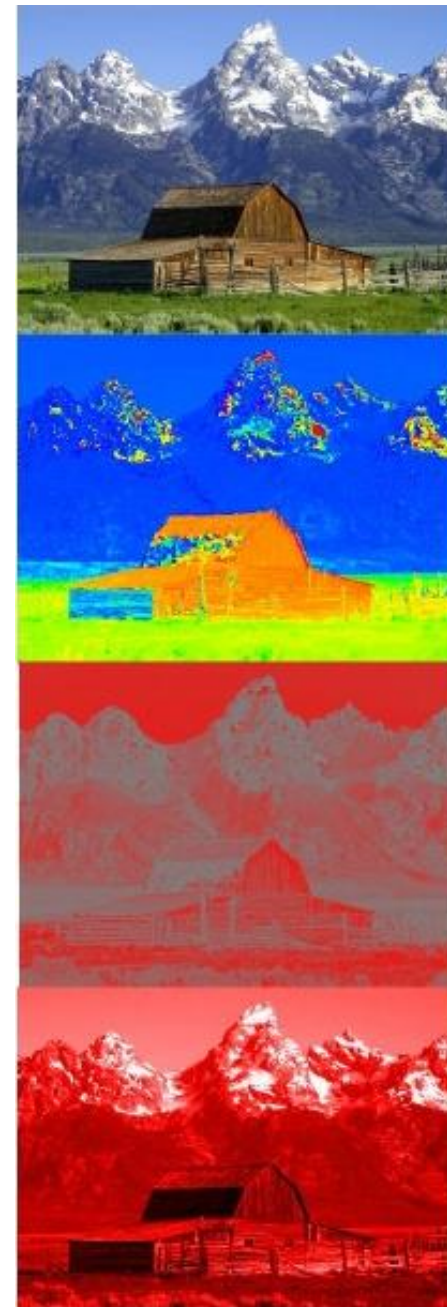
Два подхода:

1. Покомпонентная — обрабатываем каждый канал отдельно
2. Векторная — обрабатываем все каналы вместе

$$c = [c_R, c_G, c_B] = [R, G, B]$$

Пример: повышение контраста

1. Покомпонентно: $R \times 1.2$, $G \times 1.2$, $B \times 1.2$
2. Векторно: учитываем взаимосвязь каналов



Преобразование цветного к полутоновому



1. Простое усреднение:

$$\text{Gray} = (R + G + B) / 3$$

2. NTSC (стандарт ТВ):

$$\text{Gray} = 0.3 * R + 0.59 * G + 0.11 * B$$

3. sRGB (современный):

$$\text{Gray} = 0.21 * R + 0.72 * G + 0.078 * B$$

Почему коэффициенты разные?

Человеческий глаз по-разному чувствителен к R, G, B

Цветное → полутоновое → бинарное



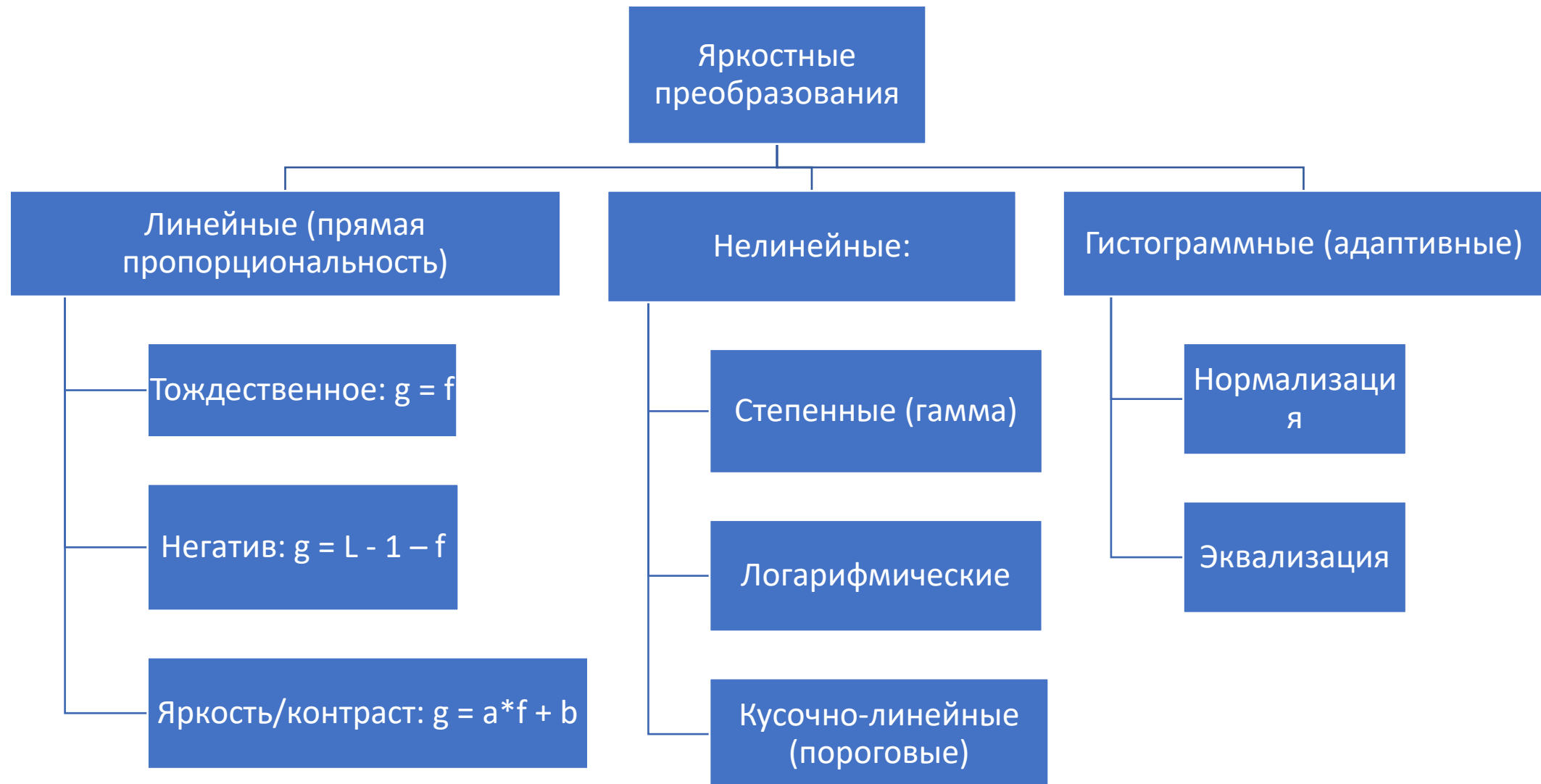
Зачем:

- Цветное → серое: убираем цвет, оставляем яркость
- Серое → бинарное: выделяем объект от фона

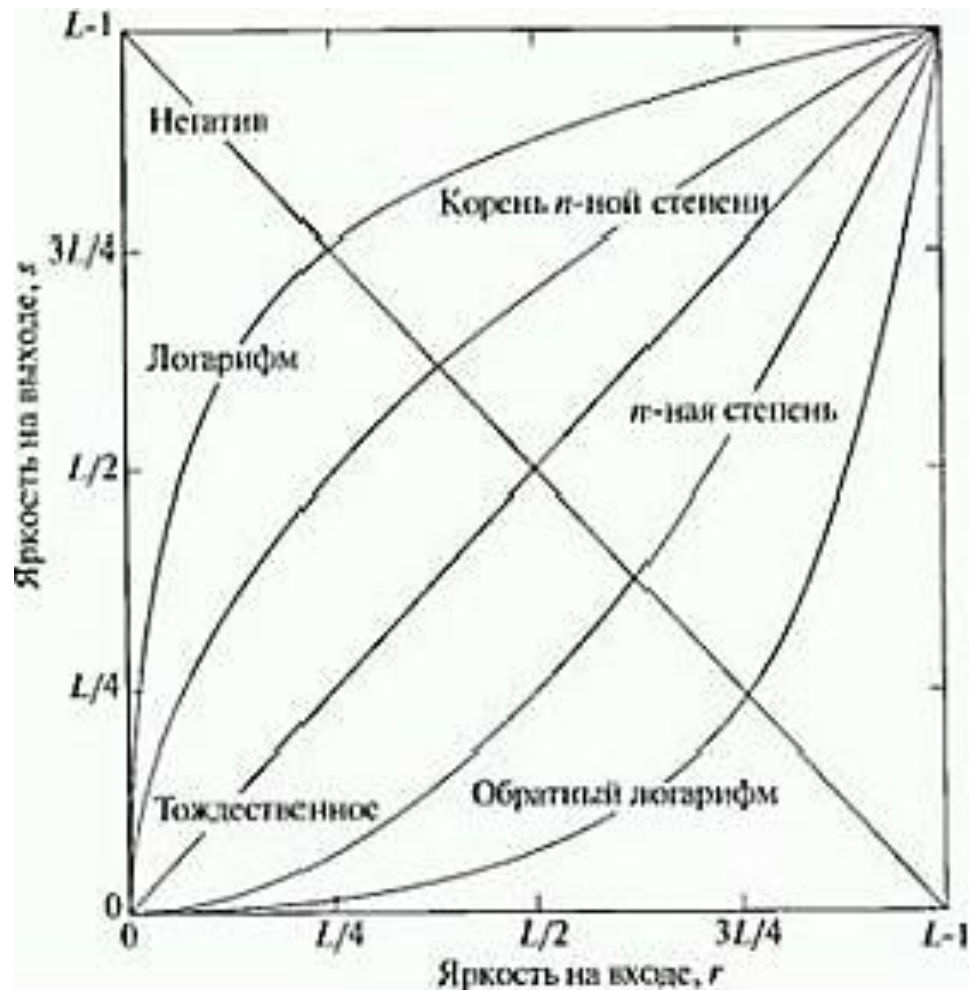
Применение:

OCR (распознавание текста), сегментация

Классификация яркостных преобразований



Графики преобразований



- Негатив (отрицательный наклон)
- Логарифм (быстрый рост в тенях)
- Корень n -й степени (медленный рост)
- n -я степень (быстрый рост в светах)
- Тождественное (прямая $y = x$)
- Обратный логарифм (быстрый рост в светах)

Линейные преобразования

- Тожественное преобразование:
 $g = f$
(Базовый случай)
- Негатив (инверсия):
 $g = L - 1 - f$,
где L – максимальное значение яркости (например, 255)
- Изменение яркости и контрастности:
 $g = a * f + b$
 - Коэффициент a (угол наклона) отвечает за контраст ($a > 1$ – увеличение, $a < 1$ – уменьшение).
 - Коэффициент b (сдвиг) отвечает за яркость ($b > 0$ – увеличение, $b < 0$ – уменьшение).

Пример — цветовое дополнение

$$\text{neg} = \text{Max} - \text{pos}$$

Аналогично негативу для полутонного изображения

Применение:

 Медицина (рентген) — выявление деталей в тёмных областях

 Астрономия — слабые объекты

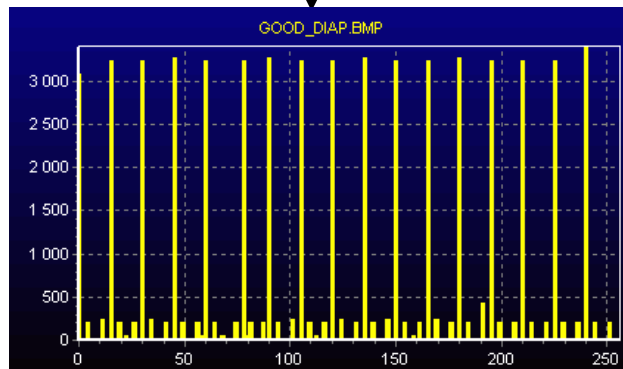
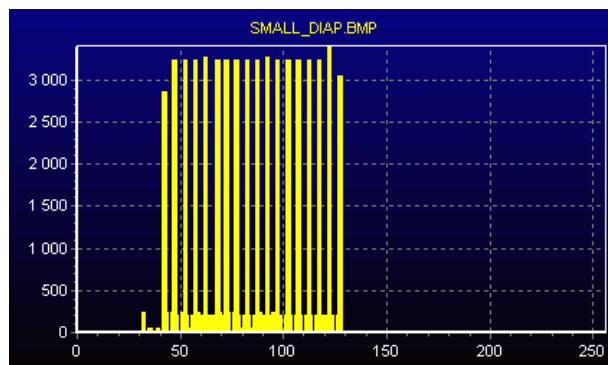
 Фотография — художественный эффект



Линейное растяжение

Проблема: узкий диапазон яркостей

Компенсация узкого диапазона яркостей – линейное растяжение:



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

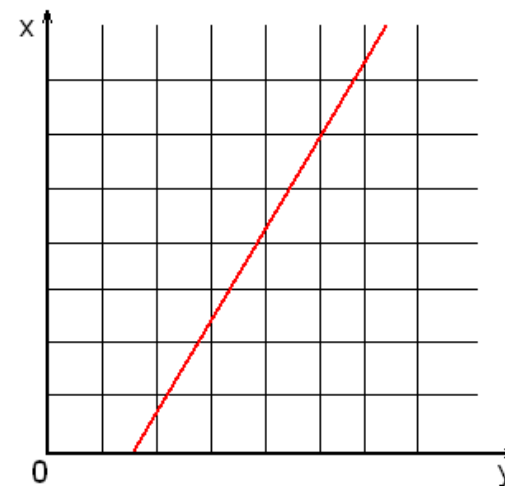


График функции $f^{-1}(y)$

Пример — Линейное растяжение (как AutoContrast в Photoshop)



Линейная коррекция помогает не всегда

Когда линейная коррекция не работает



Проблема:

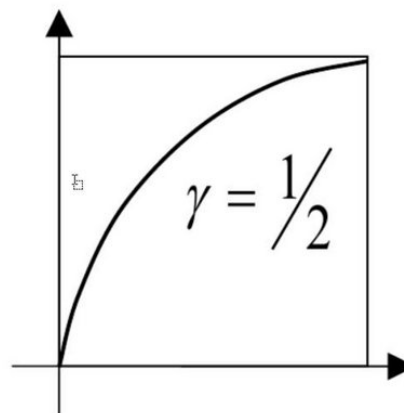
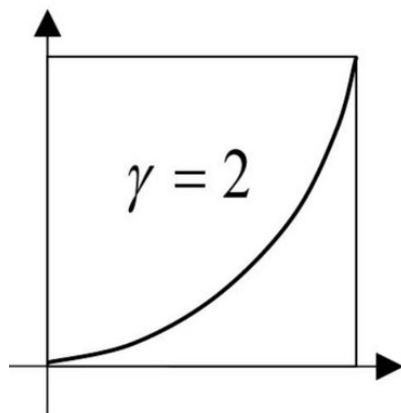
- Лицо слишком тёмное
- Лунный свет слишком светлый
- Линейное растяжение не помогает

Решение:

нелинейные преобразования (гамма)

Нелинейные преобразования — гамма

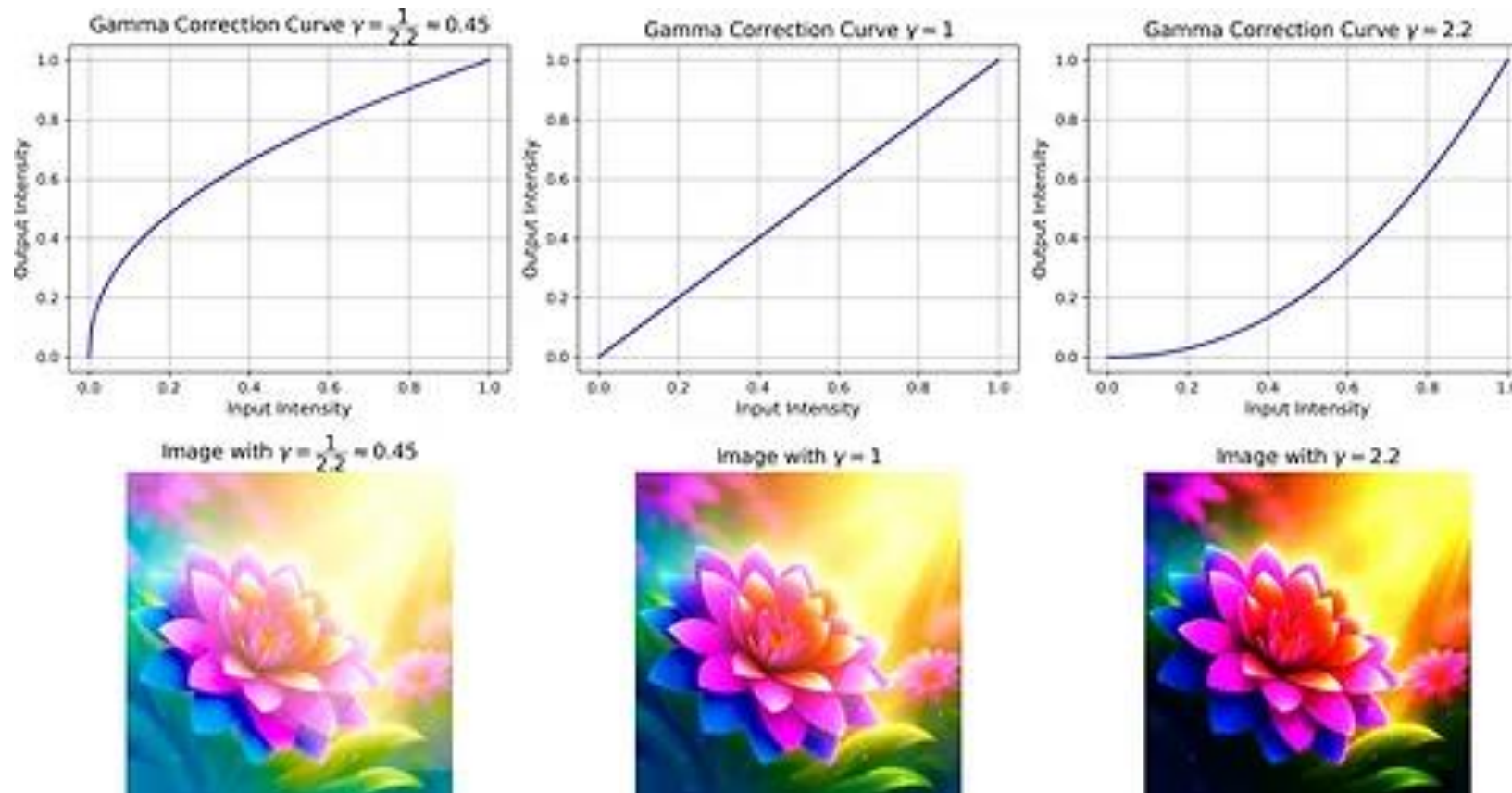
$$y = c \cdot x^\gamma$$



Параметр γ (гамма) определяет тип преобразования.

- $\gamma < 1$: «проявление теней» — растягивание темных участков, сжатие светлых
Пример: коррекция слишком темного изображения
- $\gamma > 1$: «проявление света» — растягивание светлых участков, сжатие темных
Пример: повышение резкости деталей в ярких областях
- $\gamma = 1$: линейное преобразование

Демонстрация гамма-коррекции изображения



Логарифмическое преобразование

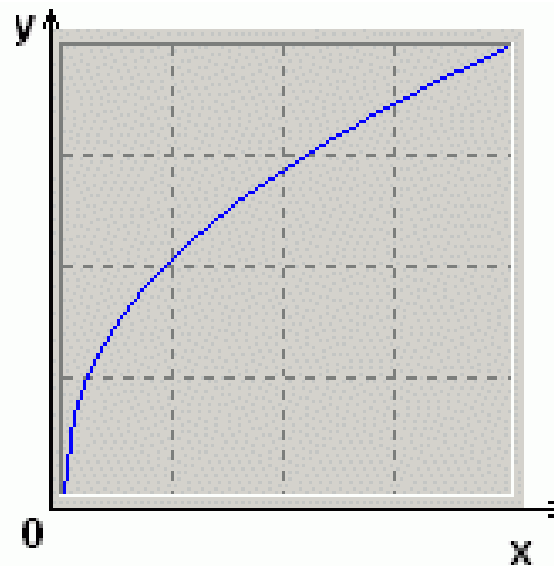
Эффект:

- Сильное растягивание тёмных пикселей
- Сжатие светлых пикселей

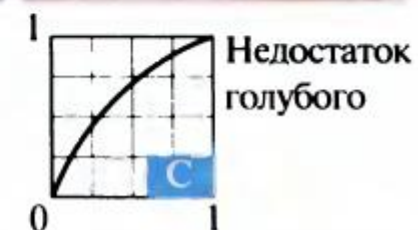
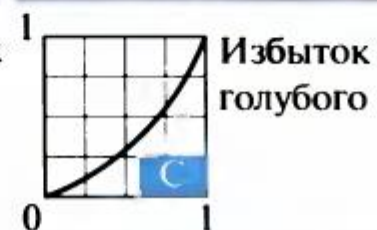
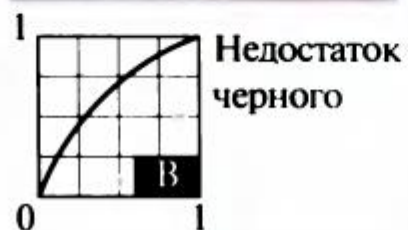
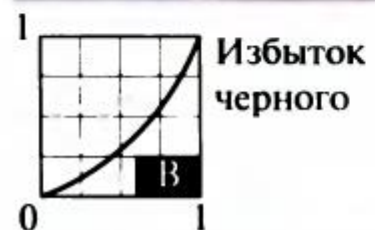
Применение:

- HDR (High Dynamic Range)
- Вывод деталей в окне на одном снимке
- Спутниковые снимки

$$g = c * \log(1 + f)$$



Цветовая коррекция по каналам



Частный случай — коррекция с опорным цветом



Предположение

Пользователь указывает цвет вручную;

Источник:

- Априорные знания — «облака — белые»
- Хорошая фотография этой же сцены

Метод

Преобразовать по каждому из каналов цвета по формулам

$$R * \frac{R_{dst}}{R_{src}}; \quad G * \frac{G_{dst}}{G_{src}}; \quad B * \frac{B_{dst}}{B_{src}};$$

Частный случай — «Серый мир»

Предположение: Сумма всех цветов на изображении естественной сцены дает серый цвет;

Метод: Посчитать средние яркости по всем каналам:

$$\bar{R} = \frac{1}{N} \sum R(x, y); \quad \bar{G} = \frac{1}{N} \sum G(x, y); \quad \bar{B} = \frac{1}{N} \sum B(x, y); \quad Avg = \frac{\bar{R} + \bar{G} + \bar{B}}{3};$$

Масштабировать яркости пикселей по следующим коэффициентам:

$$R' = R \cdot \frac{Avg}{\bar{R}}; \quad G' = G \cdot \frac{Avg}{\bar{G}}; \quad B' = B \cdot \frac{Avg}{\bar{B}};$$



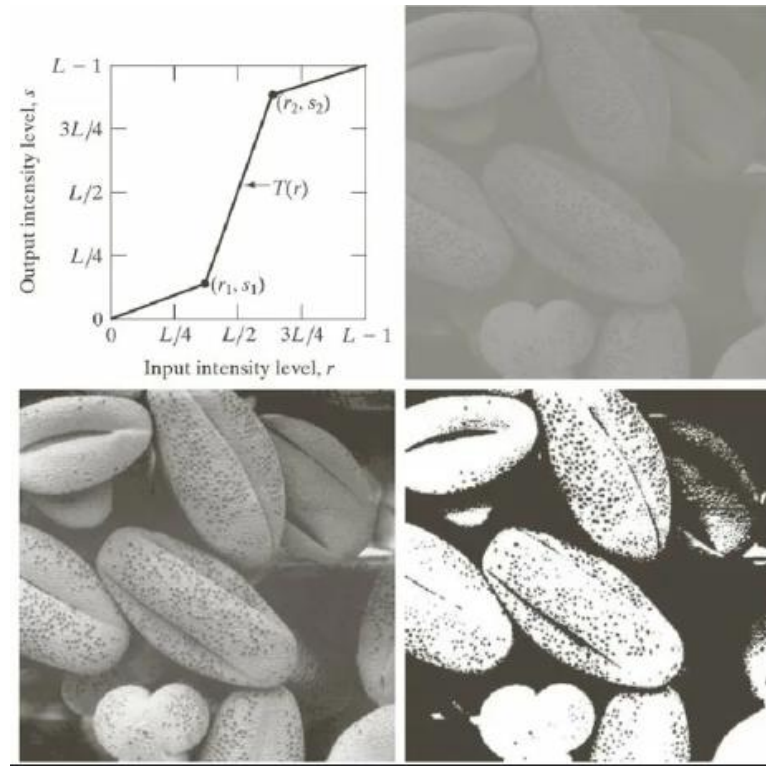
Кусочно-линейные преобразования

Общая идея:

Разные правила преобразования для разных диапазонов яркости.

Виды:

- пороговые преобразования
- бинаризация



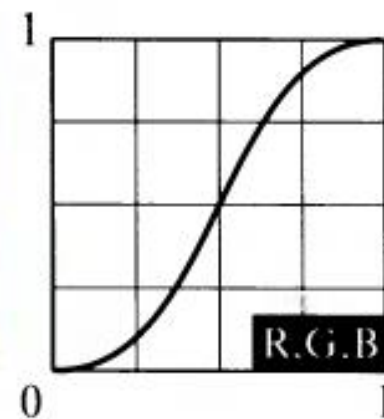
Пример ручной настройки яркостной коррекции



Малоконтрастное изображение



Результат коррекции



LUT — концепция

LUT (Look-Up Table, или таблица поиска)

Проблема:

Гамма-коррекция требует 10 миллионов операций возведения в степень для 10-мегапиксельного фото. Это медленно!

Решение:

LUT — заранее рассчитанная таблица

Для 8-битного изображения: всего 256 элементов!

Формула:

$$Im[i,j] = LUT[Im[i,j]]$$

LUT — это не отдельный вид преобразования, а **механизм (способ) эффективной реализации** практически любого яркостного преобразования

LUT — преимущества

1. Высокая скорость

Операция подстановки — одна из самых быстрых
Работает в реальном времени на слабых устройствах

2. Универсальность

Можно записать любую функцию, даже без формулы (кривая, нарисованная пользователем)

3. Стабильность

Одно и то же входное значение → одинаковый результат

Применение:



Кино (цветокоррекция)



Смартфоны (фильтры)



Игры (постобработка)

LUT — примеры

LUT в кино

Пример: Osiris и Impulz от VisionColor

Что можно делать с LUT:

- Тёплый/холодный тон
- Винтажный вид
- Кинематографичный контраст
- Стилизация под плёнку



Резюме

1. Яркостные преобразования — первый шаг обработки

$$g(x,y) = T[f(x,y)]$$

2. Типы изображений:

- Многоканальные (RGB)
- Полутоновые (Gray)
- Бинарные (Binary)

3. Линейные преобразования:

- Негатив
- Яркость/контраст
- Линейное растяжение

4. Нелинейные:

- Гамма-коррекция
- Логарифм

5. LUT — быстрая реализация

Домашнее задание

ДОМАШНЕЕ ЗАДАНИЕ

1. Реализовать линейное растяжение (Python + OpenCV)
2. Реализовать гамма-коррекцию
3. Сравнить с OpenCV (cv2.normalize, cv2.LUT)

Полезные ссылки:

- OpenCV: <https://docs.opencv.org/>
- Szeliski, Chapter 3

Что дальше?

Разобрали:

- Яркостные преобразования: $g = T[f]$
- LUT — быстрая реализация
- Гамма, логарифм — нелинейные функции

Рассмотрим:

- Как анализировать яркости?
- Как автоматически улучшить контраст?
- Что такое гистограмма?

Что такое гистограмма изображения?

Определение:

Гистограмма — это распределение яркостей пикселей

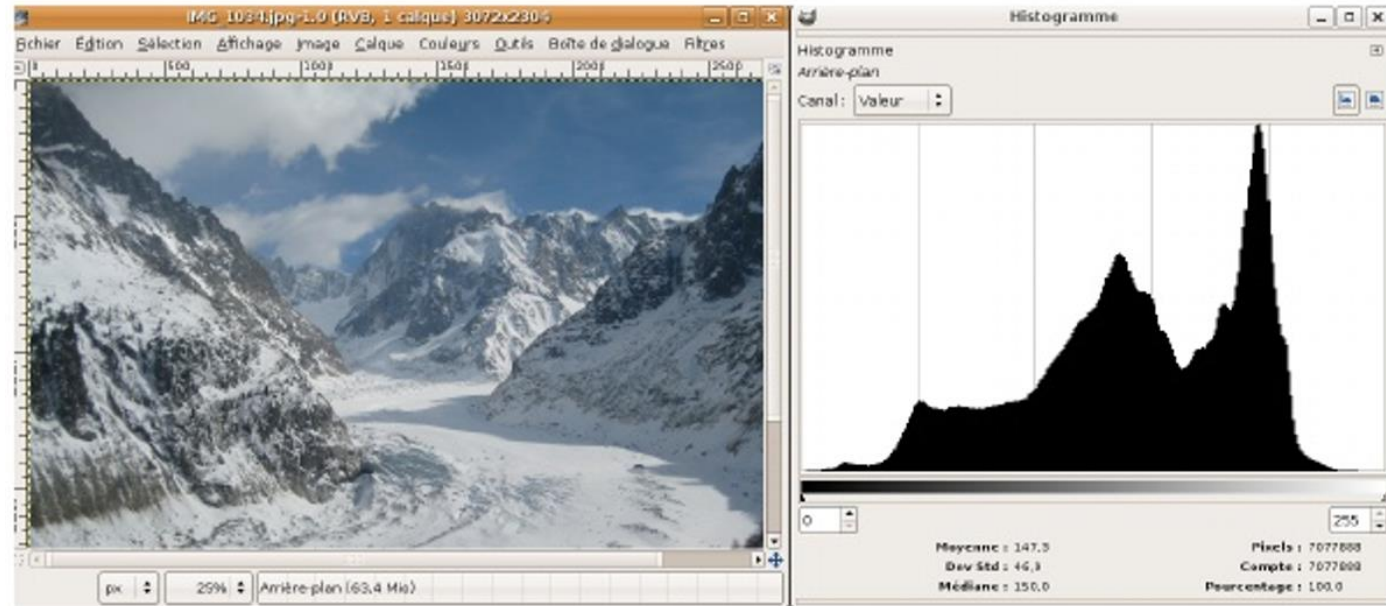
$$h(m) = |\{(r, c) \mid I(r, c) = m\}|$$

где:

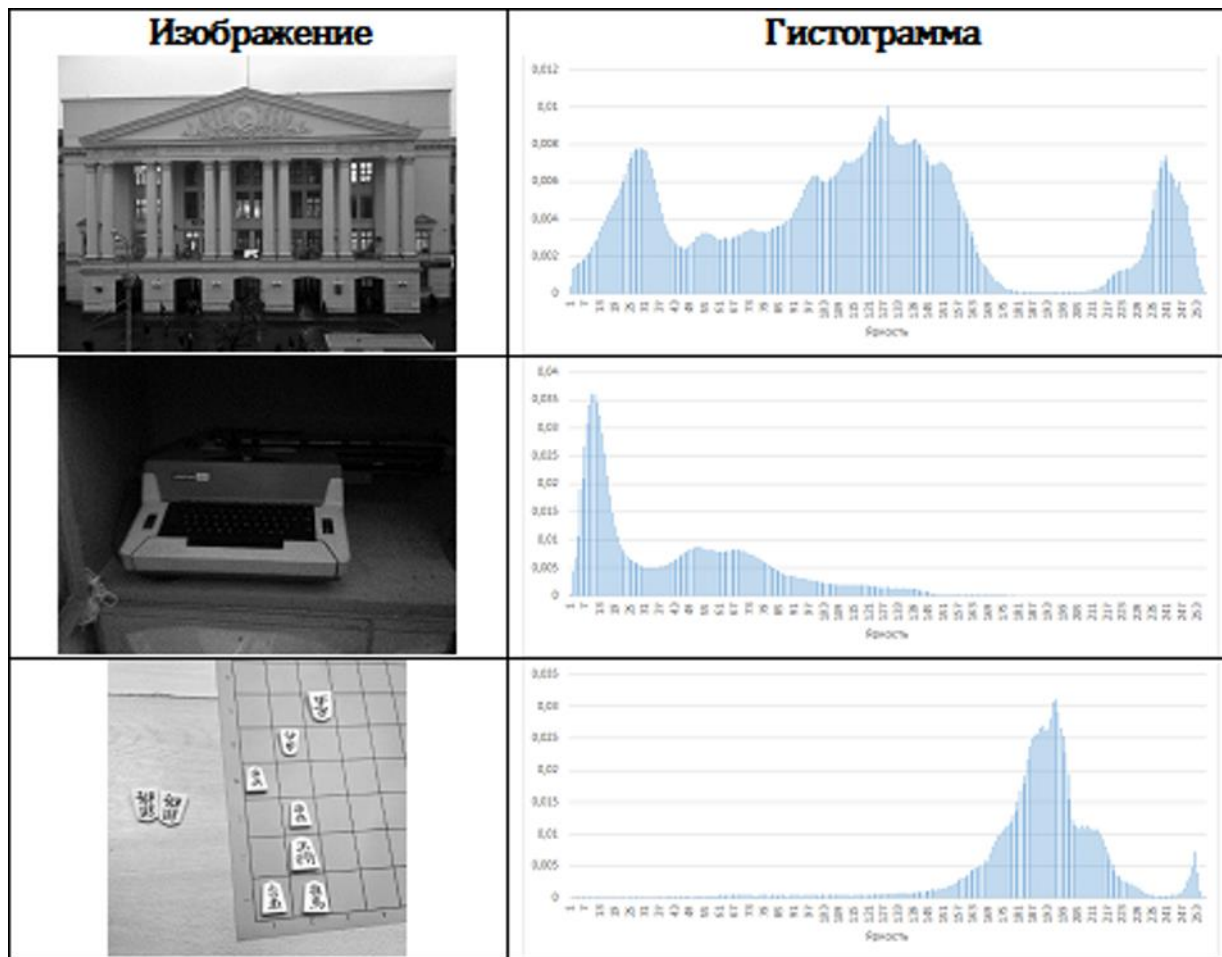
- m — значение яркости (0-255)
- $I(r, c)$ — яркость пикселя в позиции (r, c)
- $h(m)$ — количество пикселей с яркостью m

Что показывает:

- Тёмное изображение → пик слева
- Светлое → пик справа
- Контрастное → широкий диапазон
- Малоконтрастное → узкий диапазон



Примеры гистограмм



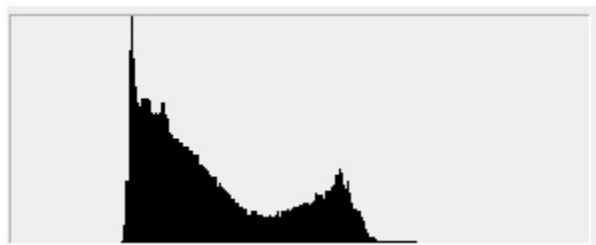
Выводы:

- По гистограмме можно понять, что не так с изображением
- Тёмное → нужно «проявить тени»
- Светлое → нужно «проявить свет»
- Малоконтрастное → нужно растянуть диапазон

Яркостная нормализация изображения



Исходное



Нормализованное



Идея: растянуть гистограмму на весь диапазон [0, 255]

Формула:

$$g = (f - f_{\min}) * 255 / (f_{\max} - f_{\min})$$

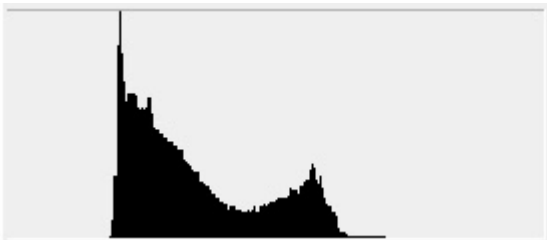
Когда применять:

- Малоконтрастные изображения
- Узкий диапазон яркостей

Эквализация гистограммы



Исходное



С эквализованной гистограммой



Цель: максимальное соответствие равномерному закону распределения

Идея:

- Растягиваем часто встречающиеся яркости
- Сжимаем редко встречающиеся

Отличие от нормализации:

- Нормализация — линейное растяжение
- Эквализация — нелинейное преобразование

Сравнение нормализации и эквализации



Нормализованное



Нормализация:

- Линейное растяжение
- Сохраняет форму распределения
- Может не убрать «провалы»

Эквализация:

- Нелинейное преобразование
- Выравнивает распределение
- Даёт более контрастное изображение
- Может усилить шум



С эквализованной гистограммой



Как работает эквализация

- Значение яркости 0:

$\text{sum}(\text{hist}[i], i=0..i(0)) \leq q$

- Значение яркости 1:

$\text{sum}(\text{hist}[i], i=i(0)+1..i(1)) \leq q$ или

$\text{sum}(\text{hist}[i], i=0..i(1)) \leq 2q$ и т.д.

- Значение яркости n:

$\text{sum}(\text{hist}[i], i=i(n-1)+1..i(n)) \leq q$ или

$\text{sum}(\text{hist}[i], i=0..i(n)) \leq nq$

$$q = \frac{w * h}{N}, \quad N = 256$$

Алгоритм эквализации

1. Построить гистограмму $h[i]$
2. Нормировать: $h[i] = h[i] / (W * H)$
3. Построить накопленную гистограмму:
for $i = 1$ to 255:
 $h[i] = h[i-1] + h[i]$
4. Заменить яркости:
 $Pixels[i,j] = h[round(255 * Pixels[i,j])]$

```
const k = 255;  
h: array [0 .. k] of double;
```

```
//построение гистограммы  
for i := 0 to k do h[i] := 0;
```

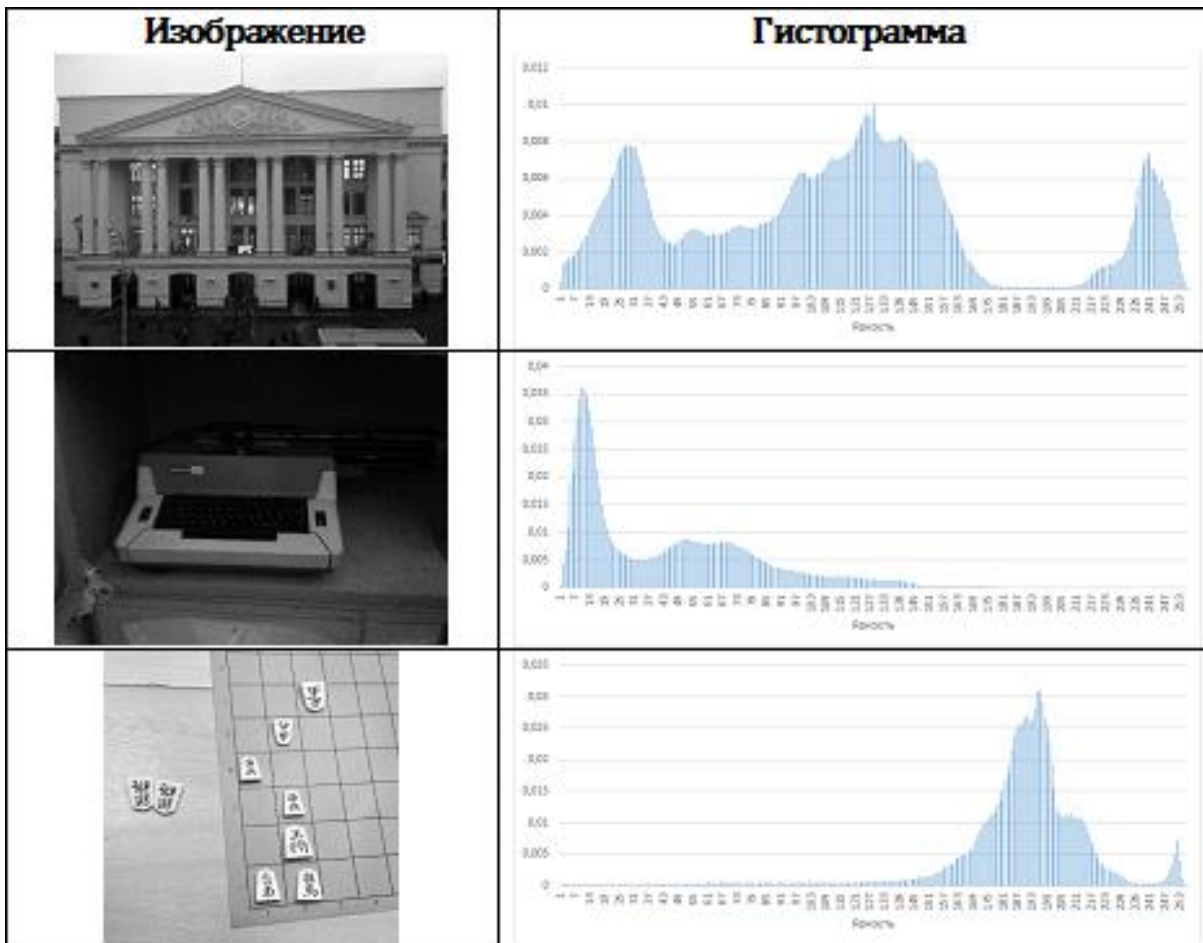
```
for i := 0 to Height - 1 do  
  for j := 0 to Width - 1 do  
     $h[round(k * Pixels[i, j])] := h[round(k * Pixels[i, j])] + 1;$ 
```

```
//нормирование гистограммы  
for i := 0 to k do  
   $h[i] := h[i] / (Height * Width);$ 
```

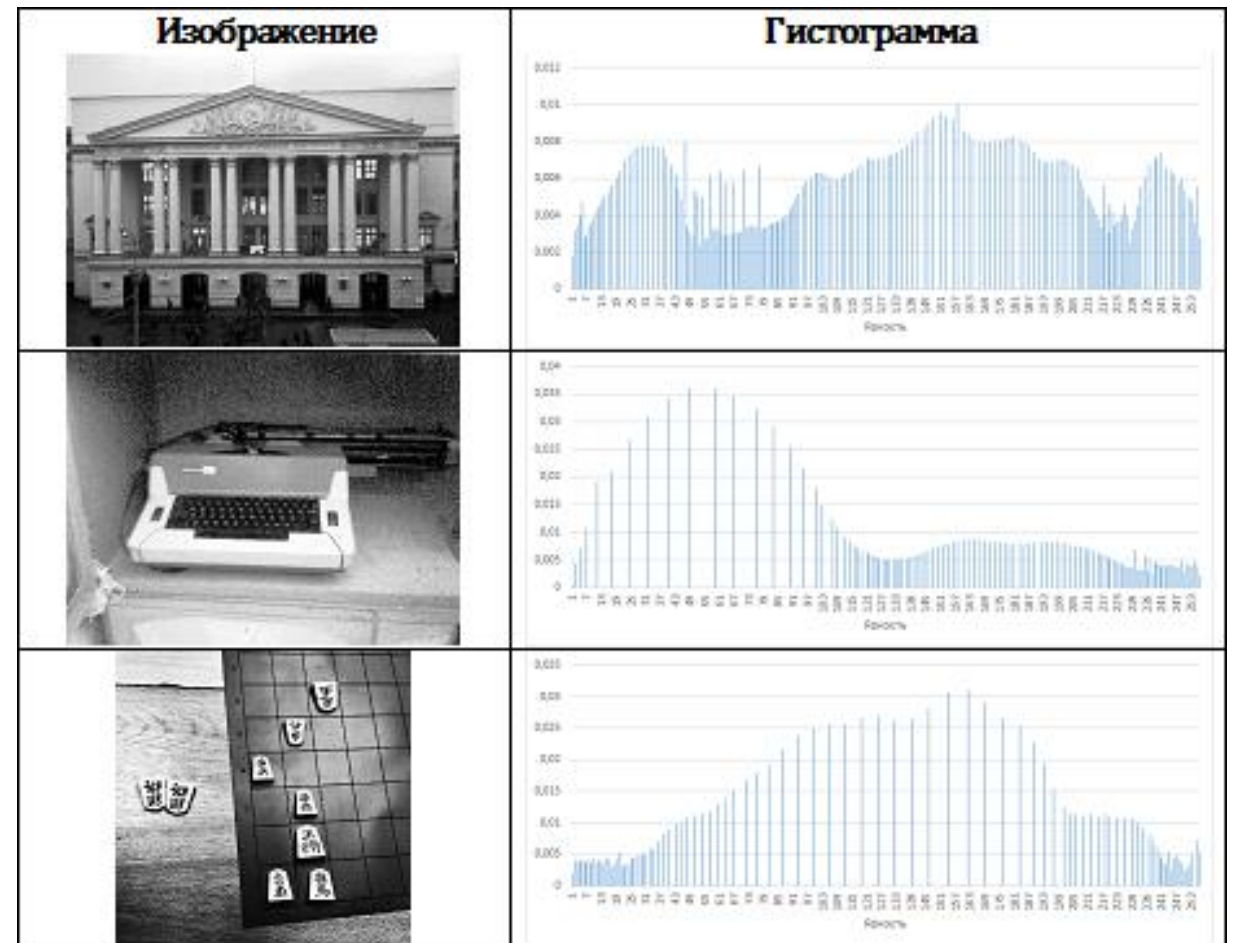
```
// построение гистограммы с накоплением  
for i := 1 to k do  
   $h[i] := h[i - 1] + h[i];$ 
```

```
// равномерное распределение значений  
for i := 0 to Height - 1 do  
  for j := 0 to Width - 1 do  
     $Pixels[i, j] := h[round(k * Pixels[i, j])];$ 
```

Примеры: изображения и их эквализации



1. Здание — улучшение контраста
2. Тёмное фото — проявление деталей
3. Шахматы — выравнивание



Вывод:

Эквализация работает автоматически, без настройки параметров!

Эквализация для цветных изображений

Проблема эквализации RGB:

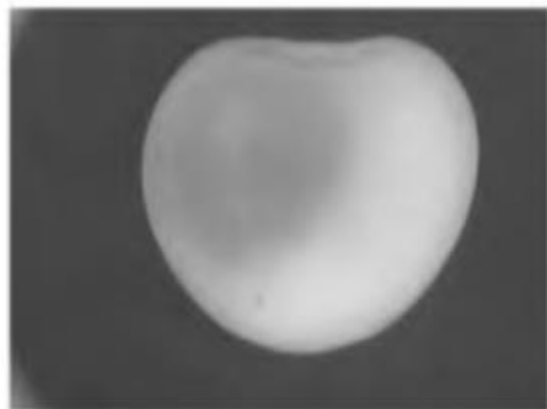
- Каналы обрабатываются отдельно
- Меняется цветовой баланс
- Появляются артефакты

Решение:

- Перейти в HSL
- Эквализовать только канал L (интенсивность)
- Вернуться в RGB



Бимодальная гистограмма



Что такое бимодальная гистограмма?

- Два пика: объект и фон
- Между ними — долина

Применение:

- Автоматический выбор порога для бинаризации
- Сегментация объекта от фона

Поиск порога на гистограмме

Для гистограммы с двумя пиками

Алгоритм:

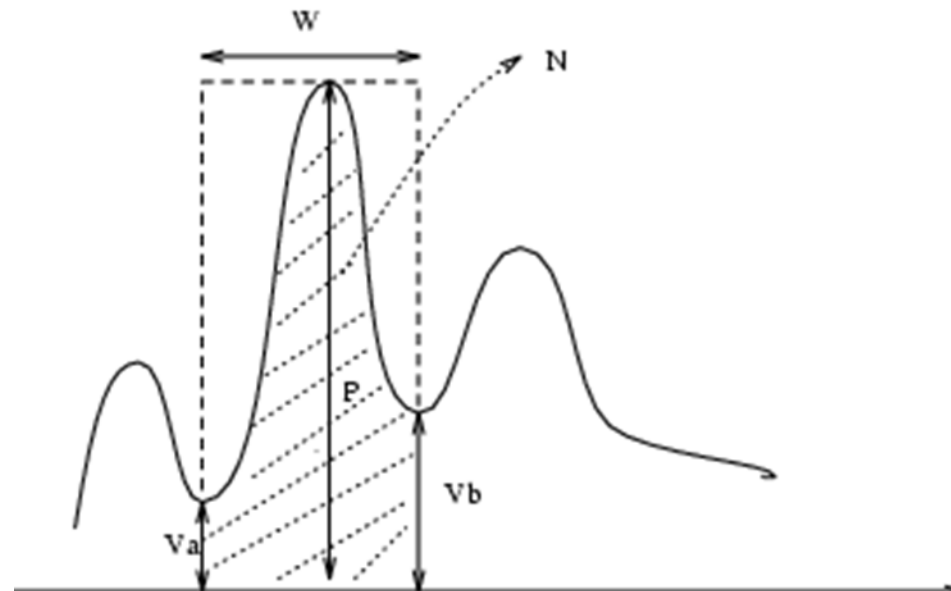
1. Найти локальные максимумы (пики)
2. Рассчитать меру пиковости
3. Отфильтровать слабые пики
4. Найти минимум между пиками — это порог

Формула для расчета пиковости

$$Peak = \left(1 - \frac{(V_a + V_b)}{2P}\right) \cdot \left(1 - \frac{N}{(W \cdot P)}\right)$$

где:

- P — высота пика
- V_a, V_b — высоты долин
- W — ширина пика
- N — количество пикселей



Резюме

1. Гистограмма — распределение яркостей

$$h(m) = |\{(r, c) \mid I(r, c) = m\}|$$

2. Нормализация — линейное растяжение

3. Эквиализация — выравнивание распределения

- Работает автоматически
- Улучшает контраст
- Для цветных — в пространстве HSL

4. Бимодальная гистограмма → поиск порога

Домашнее задание

ДОМАШНЕЕ ЗАДАНИЕ

1. Построить гистограмму для 3 разных фото
2. Реализовать нормализацию
3. Реализовать эквализацию
4. Сравнить результаты