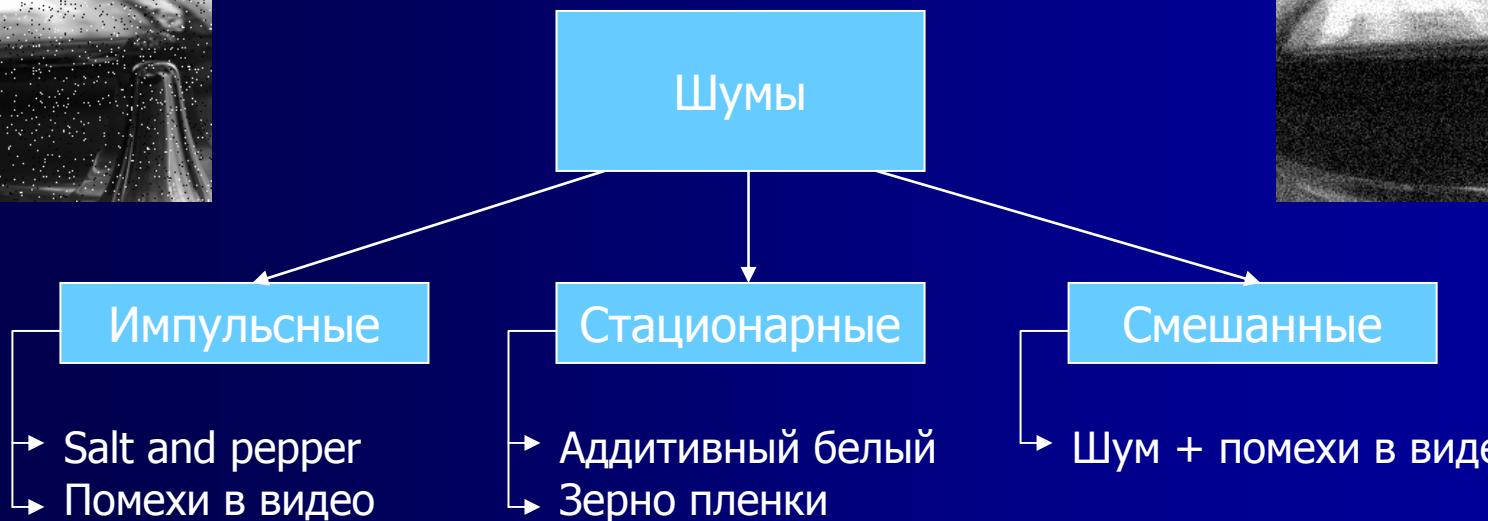
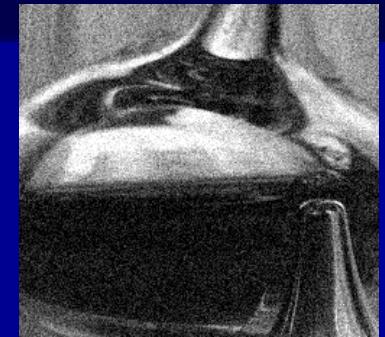


Лектор: Лукин Алексей Сергеевич

Шумоподавление для изображений

Виды и примеры шумов



Белый шум – пиксели случайны, не коррелированы друг с другом.
Гауссов/равномерный/треугольный шум – закон распределения амплитуд пикселей.

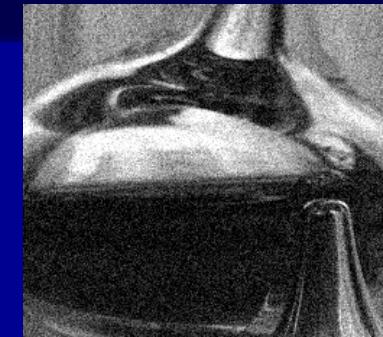
Аддитивный шум – прибавляется к «чистому» изображению и не зависит от него.

Методы шумоподавления

Salt and pepper



AWGN



Шумы

Импульсные

Стационарные

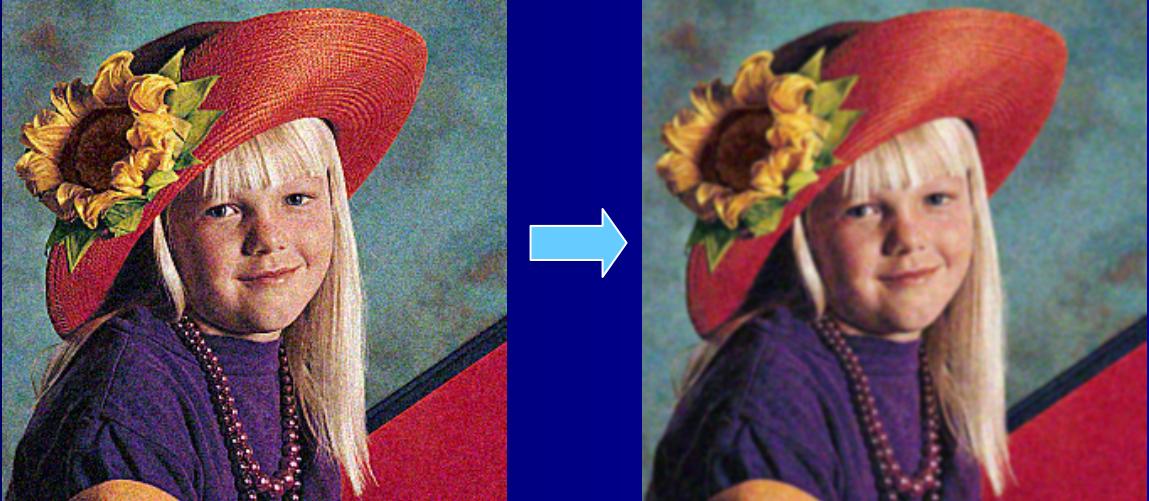
Смешанные

Медианный фильтр
Взвешенная медиана
Ранговые фильтры

Bilateral filter
Non-Local Means
Wavelet thresholding
DCT, PCA, ICA
Анизотропная диффузия
Алгоритм BM3D

Ранговые фильтры
Комбинированные методы

Простейшие методы

- Простейшие методы
 - ▶ Размытие изображения – вместе с шумом размывает детали
 - ▶ Размытие в гладких областях – остается шум вблизи границ
 - ▶ Медианная фильтрация – хорошо подавляет импульсный шум, но удаляет мелкие детали

Bilateral filter

- Адаптивные алгоритмы
 - ▶ Bilateral filter
усреднение окружающих
пикселей
с весами

$$y_{i,j} = \sum_{k,m \in Q} x_{i+k, j+m} \cdot W(i, j, k, m)$$

$$W(i, j, k, m) \approx \exp\left(-\frac{(x_{i,j} - x_{i+k, j+m})^2}{h^2}\right) \cdot \exp\left(-\frac{k^2 + m^2}{\rho^2}\right)$$

фотометрическая близость

пространственная близость



Bilateral filter

- Bilateral filter: художественное применение



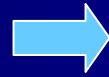
(при слишком сильном действии)

Non-Local Means

- Адаптивные алгоритмы
 - ▶ Non-local means (NL-means) – веса зависят от близости целых блоков, а не отдельных пикселей

$$W(i, j, k, m) \approx \exp\left(-\frac{\|v(x_{i,j}) - v(x_{i+k,j+m})\|^2}{h^2}\right)$$

$v(x_{i,j})$ – блок вокруг
пикселя $x_{i,j}$



Non-Local Means

■ Вычисление весов

Веса высоки для q_1 и q_2 ,
но не для q_3

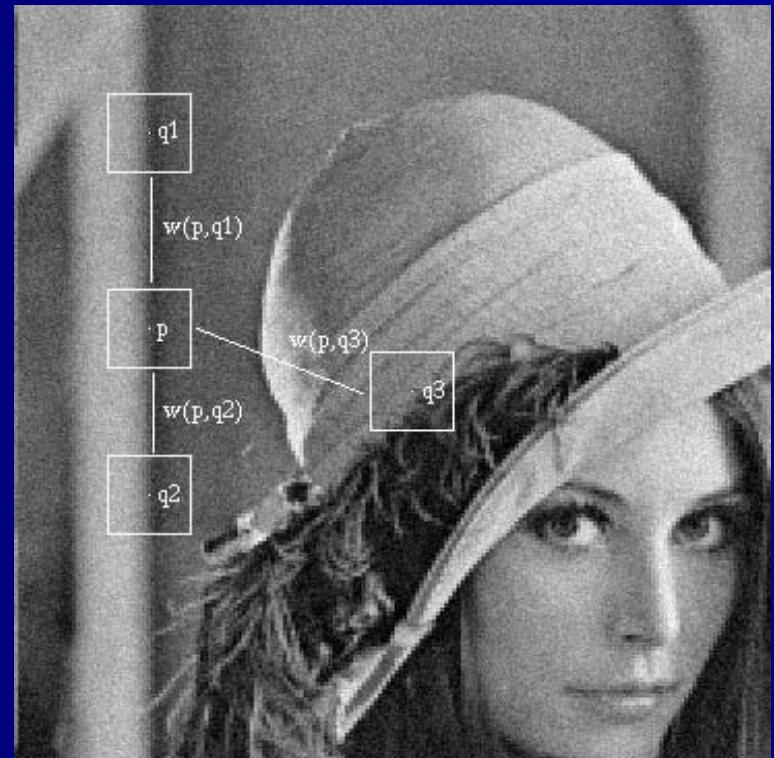
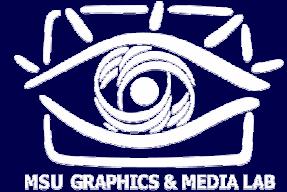


Иллюстрация из
Buades et al 2005



- Способен сохранять текстуру изображения лучше,
чем bilateral filter



Non-Local Means

■ Достоинства и недостатки:

- + ► Высокое качество результирующего изображения
- ► В исходном варианте – очень высокая вычислительная сложность

■ Ускоряющие расчет оптимизации:

- Использование команд MMX/SSE для сравнения блоков
- Разбиение изображения на блоки и усреднение целых блоков, а не отдельных пикселей
- Промежуточный вариант между усреднением блоков и усреднением пикселей: усреднение маленьких блоков



Non-Local Means

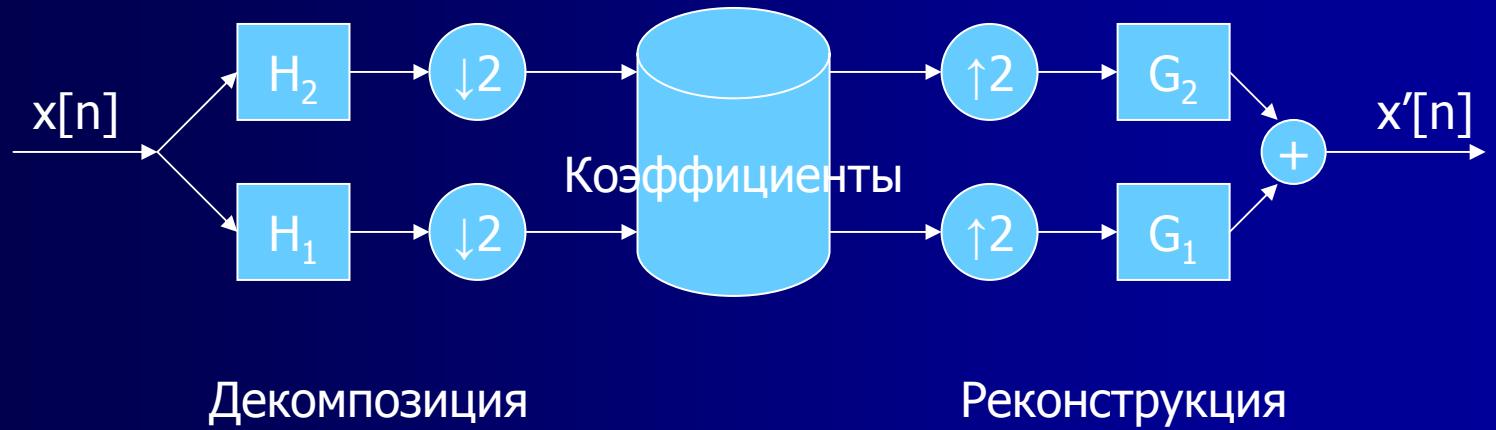
■ Применение к видео

- ▶ Область поиска блоков можно расширить на соседние кадры (сделать ее пространственно-временной)
- ▶ Для ускорения просчета можно применять сравнение блоков только по Y-каналу в цветовой модели YCrCb (YUV)

Дискретное вейвлет-преобразование



■ Схема одномерного ДВП



Свойство точного восстановления (PR): $x[n] \equiv x'[n]$

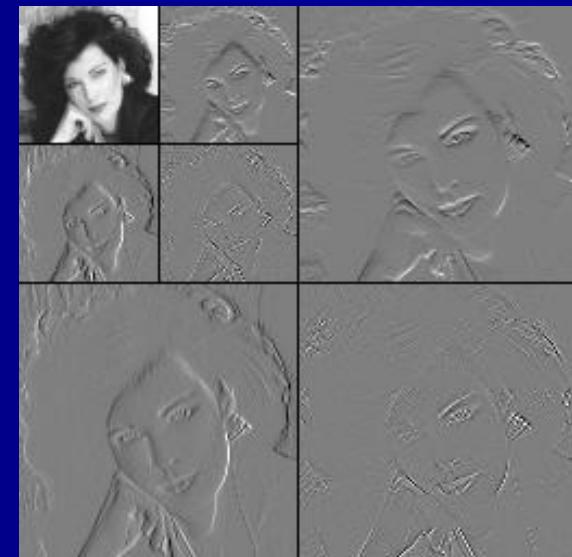
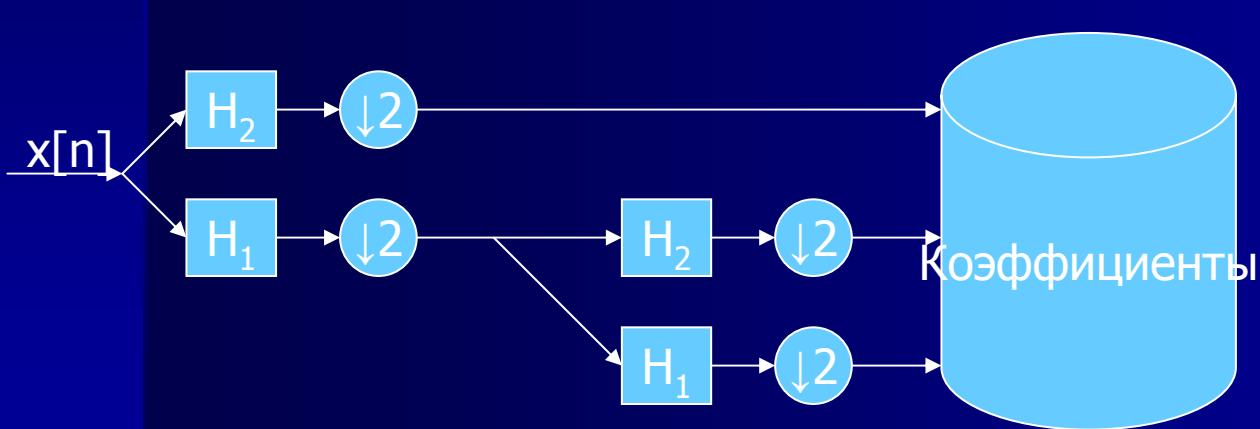
Количество информации не изменяется.

Существуют фильтры H и G , обеспечивающие точное восстановление.

Дискретное вейвлет-преобразование

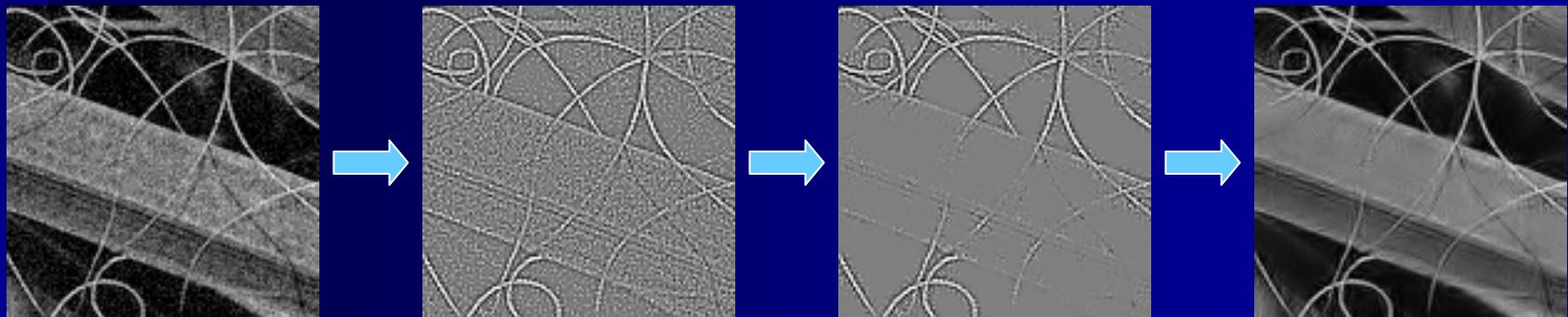


- ДВП можно применять несколько раз («несколько уровней разложения»)
- Двумерное ДВП – композиция одномерных ДВП



Вейвлетный метод

- Вейвлетное шумоподавление для изображений
 1. DWT
 2. Оценка уровня и спектра шума
 3. Подавление вейвлет-коэффициентов (thresholding, shrinkage)
 4. Обратное DWT



Подавление шума различного масштаба



Отсутствие инвариантности к сдвигу
Плохая локализация энергии для наклонных границ

Метод главных компонент



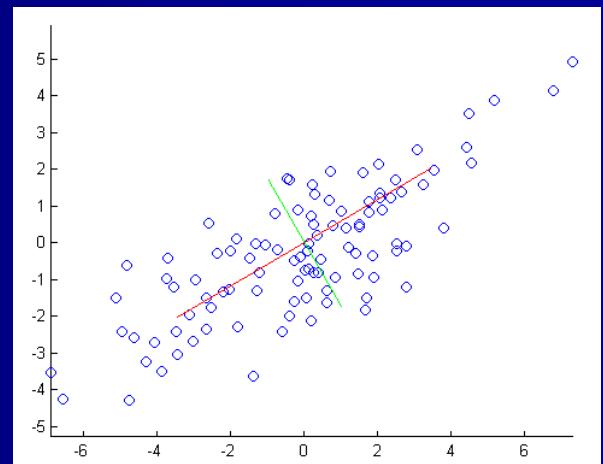
Principal Component Analysis (PCA, KLT)

- Идея: так выбрать базисные вектора, чтобы минимальным числом базисных векторов можно было наилучшим образом приблизить данные из некоторого набора.
- Решение: базисные вектора – собственные векторы e_i матрицы ковариации C_x исходных данных x :

$$x = (x_1, \dots, x_n)^T$$

$$\mu_x = E\{x\}$$

$$C_x = E\{(x - \mu_x)(x - \mu_x)^T\}$$

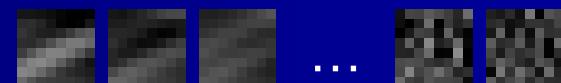
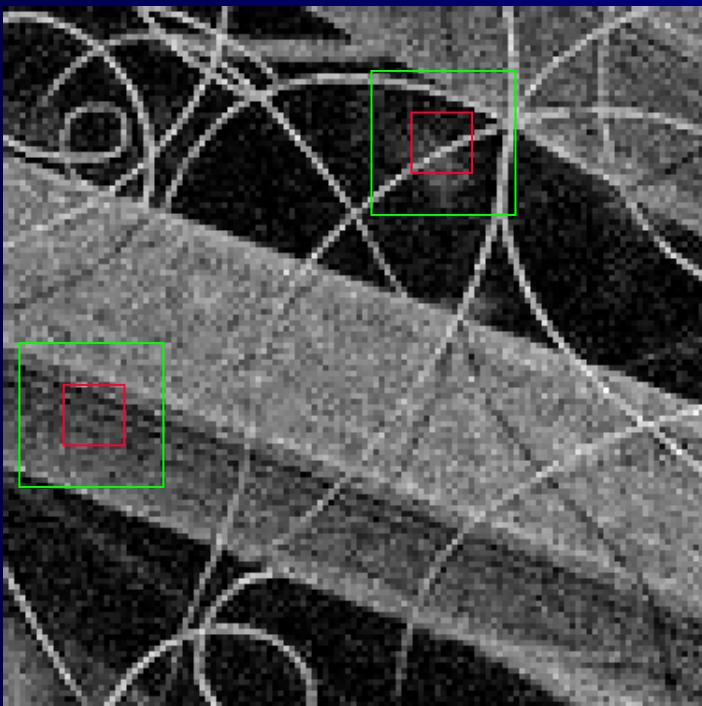


Шумоподавление

Метод главных компонент

- Применение к блокам изображения 8x8:

$$x = (x_1, \dots, x_{64})^T$$

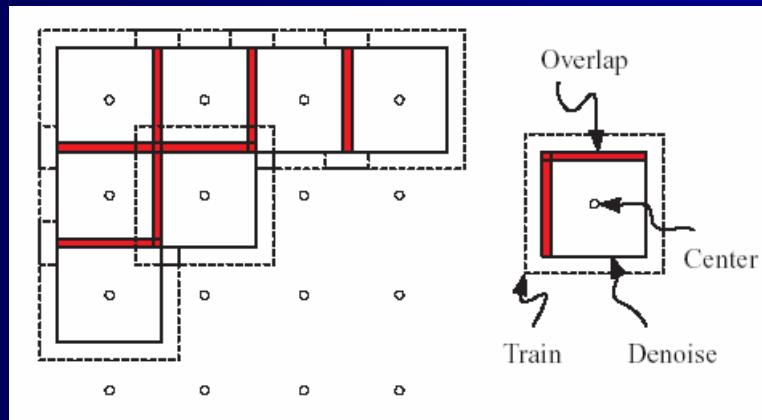


64 базисных вектора

Шумоподавление

Существующие подходы (*Muresan/Parks, 2003*)

- РСА-деноизинг изображений
 1. Блоchное преобразование РСА
 2. Подавление коэффициентов в новом базисе
 3. Обратное преобразование и наложение блоков с перекрытием



Адаптация к деталям изображения (в т.ч. – к линиям любого наклона)



При больших блоках – эффект Гиббса,
при маленьких – не подавляется крупный шум

Шумоподавление

Результаты



Зашумленное изображение



Вейвлетное шумоподавление

Шумоподавление

Результаты



Шумоподавление с помощью РСА



Разработанный нами метод

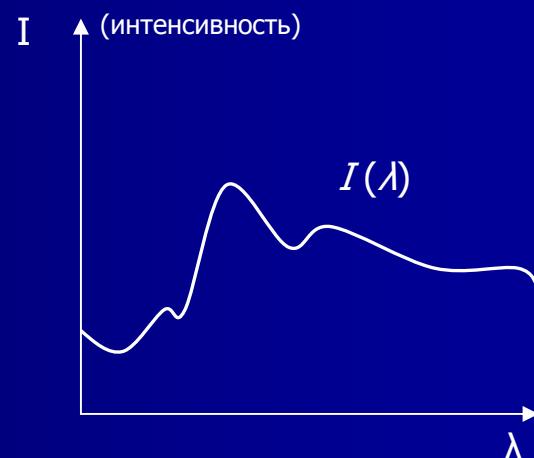
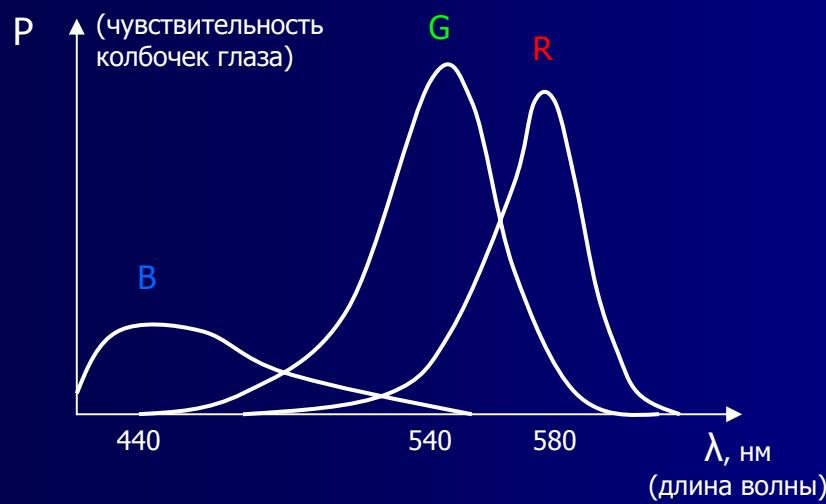
Свет и цвет

■ Восприятие цвета

$$R = \int I(\lambda)P_R(\lambda)d\lambda$$

$$G = \int I(\lambda)P_G(\lambda)d\lambda$$

$$B = \int I(\lambda)P_B(\lambda)d\lambda$$



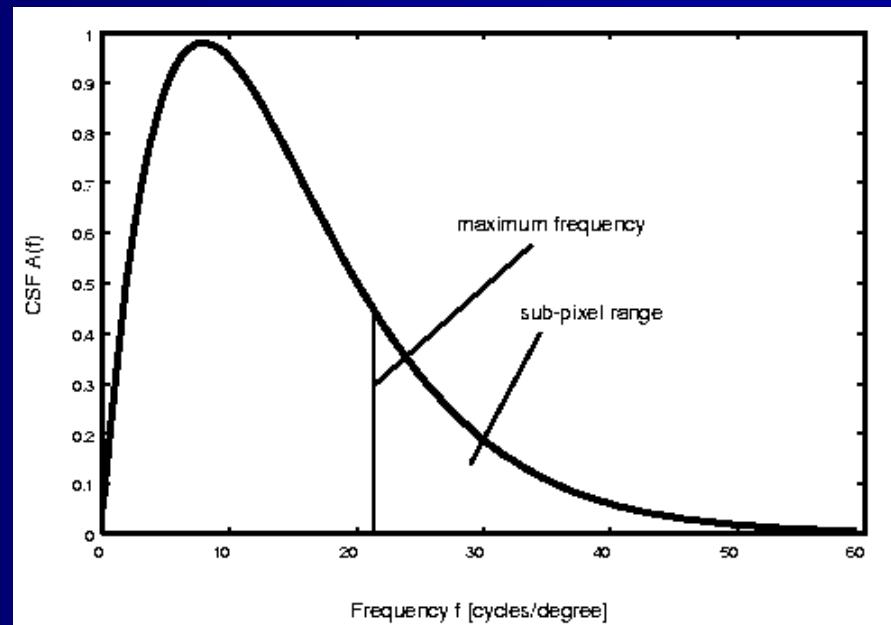
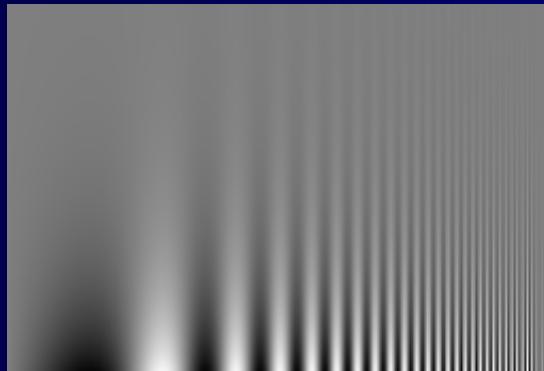
Какой это цвет?

Метрики близости

- PSNR не всегда отражает реальный видимый уровень искажений
 - Как улучшить?
 - ▶ Использовать функцию чувствительности глаза к различным частотам (CSF)
 - ▶ Использовать свойство маскировки
 - ▶ Использовать равномерные к восприятию цветовые пространства (CIE Lab, CIEDE2000)
- HVS models
(human visual system)

Метрики качества

- Contrast sensitivity function (CSF)
 - ▶ Показывает чувствительность глаза к различным частотам



Абсцисса – пространственная частота
(колебаний / градус угла обзора)

Алгоритм Retinex

- Задача: уменьшить вариации освещенности объектом на фотографиях, повысить контраст
- Предполагаем, что изображение представимо в виде произведения компонент освещенности (illumination) и собственно цветов предметов (reflectance)



$$S(x, y) = I(x, y)R(x, y)$$

Алгоритм Retinex

- Предположение: освещенность меняется медленно, это низкие частоты
- Как разделить $S(x,y)$ на компоненты?

$$S(x, y) = I(x, y)R(x, y)$$

- Идея гомоморфной обработки: применяем к исходным данным преобразование, чтобы свести задачу к линейной

$$\log S(x, y) = \log I(x, y) + \log R(x, y)$$

- Теперь компоненты разделимы с помощью фильтрации (свертки)

Алгоритм Retinex

■ Схема алгоритма

$$\log R(x, y) = \log S(x, y) - H(\log S(x, y))$$

Н – оператор НЧ-фильтрации,
напр. – гауссова размытия

■ Альтернативный вариант

$$R(x, y) = \exp(\log S(x, y) - H(\log S(x, y)))$$

возвращаем R(x,y) в исходный масштаб значений

■ Тонкие моменты

- ▶ $\log(0)$
- ▶ Диапазон результирующих значений
- ▶ Как быть с цветами?

Преобразование цвета



- Система YUV: разделение яркости и хроматических компонент

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = -0.147R - 0.289G + 0.436B$$

$$V = 0.615R - 0.515G - 0.100B$$

- После обработки компоненты Y, можно скорректировать значения U и V, чтобы поддержать насыщенность цветов

Алгоритм Retinex

■ Результаты

