

## **10. Базы данных для изображений**

10.1. Представление изображений

10.2. Сжатие изображений

10.3. Поиск по базе данных изображений

10.4. Структуры баз данных для изображений

# Представление изображений

- Оцифрованное (дискретное) представление полевых пространственных данных
- Цифровые (или растровые) изображения
- $m \times n$  матрица пикселей
- По  $k$  бит на каждый пиксел -  $2^k$  возможных значений

## Типы изображений:

- Черно-белые изображения (черный = 0; белый = 1; например, телефакс)
- Полутоновые изображения (обычно  $k = 8$ , что дает 256 уровней серого)
- Цветные изображения (различные представления)

# Представление изображений

Электромагнитные источники изображений:

- Видимый свет (фотографии)
- Инфракрасный диапазон (например, изображения со спутников)
- Рентгеновское излучение (рентгенологические (радиологические) и прочие медицинские изображения)

Устройства, создающие цифровые изображения:

- Сканер
- Цифровая камера
- Электронный микроскоп

Основные параметры:

- Разрешение (частота дискретизации, количество пикселей на дюйм)
- Точность (число бит на пиксел)

# Представление изображений

## Полутоновые изображения

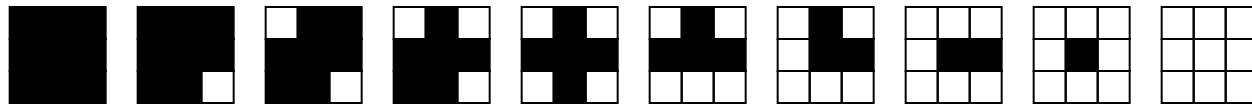
- Как правило 8 бит на пиксел, что дает 256 оттенков серого
- Передача полутонов (halftoning) – имитация уровней серого черным и белым

Стандартная технология печати: круглые точки разных размеров с равноотстоящими центрами



Цифровая передача полутонов:

- С помощью шаблонов: каждый пиксел заменяется черно-белым шаблоном («шрифтом») с определенной пропорцией черного



- Размывание (dithering): создание дополнительных цветов в изображениях с ограниченной цветовой палитрой; настройка значений соседних пикселей для создания нужных оттенков серого

# Представление изображений

## Цветные изображения

Модели естественного цвета (true color):

- Три компоненты (канала) на пиксел (плюс возможно 4-ый канал – альфа-канал ( $\alpha$ -channel) = информация о прозрачности)
- *RGB*: Red (красный), Green (зеленый), Blue (голубой); обычно  $3 \times 8 = 24$  бита на пиксел
- *CMY*: Cyan (голубой), Magenta (пурпурный), Yellow (желтый); *CMYK* используется в цветной печати, где *K* (key) - черный
- *HSI*: Hue (цвет), Saturation (насыщенность), Intensity (интенсивность); используется при обработке изображений
- *YUV*: цвет задается одной яркостной компонентой *Y* и двумя цветовыми компонентами *U* и *V*; используется при сжатии изображений; используется в формате JPEG, телевизионном стандарте PAL
  - Информация о яркости отделена от информации о цвете
  - Большая часть информации заключена в *Y*-компоненте
  - Эффективна для сжатия изображений (JPEG)

Изображения с цветовой таблицей:

- Информация о цвете хранится в отдельной таблице
- Палитра из, например, 256 цветов
- Отображение таблицы в RGB-значения
- Уменьшение размера; достаточно для многих приложений

# Представление изображений

## Форматы изображений

Существуют десятки форматов [2] для различных приложений и сред, среди них:

- BMP: Windows Bitmap - растровое изображение Windows
- GIF: Graphics Interchange Format; включает сжатие (без потерь), для обмена графическими изображениями
- JBIG: Joint Bi-level Image Experts Group; формат обмена графической информацией
- JPEG: Joint Photographic Experts Group; сжатие (с потерями) фотографических изображений
- PBM: Portable Bitmap Format; формат для черно-белых изображений; 1 бит на пиксел
- PGM: Portable Greymap Format; для полутоновой графики; 8 бит на пиксел
- PNG: Portable Network Graphics; замена GIF; сжатие без потерь
- PPM: Portable Pixmap Format; для цветных изображений; 24 бита на пиксел
- TIFF: Tagged Image File Format; для хранения изображений; гибкий и настраиваемый формат; множество опций

# Сжатие изображений

- Необходимо при работе с большими архивами изображений:
  - Уменьшение размера
  - Уменьшение времени передачи
- Возможно в силу избыточности информации в изображениях
- Несколько методов, приспособленных для различных типов изображений
- Основные этапы:
  1. Моделирование
    - Определение распределения символов или блоков
  2. Кодирование
    - Определение бинарных представлений для символов/блоков
    - На основе модели распределения
    - Алгоритмы: кодирование по алгоритму Хаффмана, арифметическое кодирование

# Сжатие изображений

## Характеристики методов

- Сжатие с потерями/без потерь: приблизительное или точное восстановление оригинального изображения
- Эффективность сжатия (скорость передачи данных), измеряется в битах на пиксел
- Скорость (отдельно для упаковки и распаковки)
- Искажение (для методов с потерями):
  - MAE (mean absolute error): средняя абсолютная ошибка
  - MSE (mean squared error): среднеквадратическая ошибка
  - RMSE (root mean squared error): квадратный корень из MSE
  - SNR (signal to noise ratio): отношение сигнал-шум
  - PSNR (peak signal to noise ratio): отношение максимальный сигнал-шум
- Ошибкоустойчивость (по отношению к потенциальным ошибкам во время передачи)
- Для методов с потерями: появление границ между блоками (блок – группа из, например, 8x8 пикселей), размывание изображения и т.д.



# Сжатие изображений

## Сжатие черно-белых (двухуровневых) изображений

- Как правило сжатие без потерь
- Вероятность(*белый*)  $\gg$  Вероятность(*черный*)
- Однородные области одного цвета
- Пример применения: телефакс

### Различные методы сжатия:

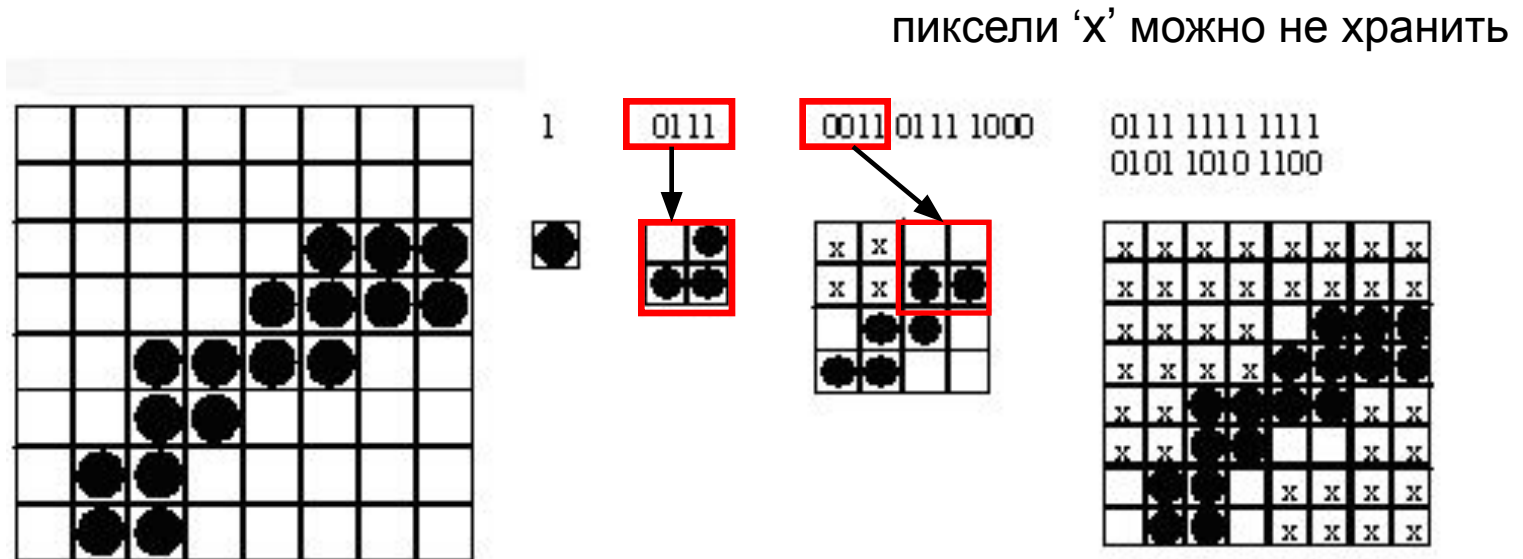
#### (1) Метод RLE (run-length coding):

- Построчная обработка, слева направо
- «Серия» - последовательность пикселей одного цвета
- Белые и черные серии чередуются
- Длины серий кодируются с помощью, например, кода Хаффмана или с помощью заранее известной таблицы кодов

# Сжатие изображений

## (2) Иерархическое блочное кодирование:

- Блоки, в которых все пиксели белые, кодируются 0, иначе 1
- Несколько уровней блоков: самый нижний уровень – непосредственно изображение
- Аналогично иерархическому сжатию битовых строк (см. тему 6, «Битово-матричное индексирование»)
- Но в данном случае, блоки бит – двумерные; деление на блоки как, например, в тетрарных деревьях
- Пример:



# Сжатие изображений

- (3) Кодирование с предсказанием (prediction coding):
- Для рассматриваемого пиксела, исходя из значений соседних пикселей (пиксели слева и сверху, т.е. уже закодированные), определяется вероятность иметь черный или белый цвет
  - Используется в JBIG (Joint Bi-level Image Experts Group)
    - Более новый стандарт известен как JBIG2, предыдущий – JBIG1
    - Стандарт от ISO/IEC и ITU-T
    - Для «предсказания» используются 7-10 пикселей
    - Окончательное кодирование с помощью QM-coder [3] (двоичное арифметическое кодирование)
    - Типичный коэффициент сжатия – 20:1

# Сжатие изображений

## Сжатие полутоновых изображений

- Сильная зависимость между значениями соседних пикселей
- Наличие текстур усложняет процесс сжатия
- Методы сжатия сильно отличаются от методов для черно-белых изображений

### (I) Сжатие без потерь:

- Относительно редкое использование
- Пример приложения: сжатие рентгенограмм
- Типичный коэффициент сжатия – 2:1
- Сжатие без потерь в JPEG (Joint Photographic Experts Group):
  - Основано на прогнозировании
  - Несколько вариантов возможных (предсказываемых) значений:  
 $a, b, c, (a+b)/2, a+b-c, b+(a-c)/2, a+(b-c)/2, \dots$
  - Кодировается (с помощью арифметического кодера) разница между реальным и «предсказанным» значениями пиксела

$c$	$a$
$b$	?

# Сжатие изображений

## (II) Сжатие с потерями:

- Удовлетворительно для большинства приложений
- Коэффициент сжатия 10:1 обычно дает гарантию “неотличимости” сжатого изображения от оригинала
- Группы применяемых методов:

### 1) Квантование (quantization):

- Как правило, не используется независимо - часть более сложных методов
- Диапазон значений задается одним значением
- Используются самые важные разряды в значениях пикселей

### 2) Прогнозирование:

- Строится «разностное» изображение (в котором значение каждого пиксела равно разнице между реальным и предсказанным значениями этого пиксела), затем полученное изображение квантуется и только потом кодируется
- DPCM - дифференциальная импульсно-кодовая модуляция (ДИКМ)

# Сжатие изображений

## 3) Субдискретизация (subsampling):

- Уменьшение разрешения; например, при замене блока в  $2 \times 2$  пикселя одним значением
- Обычно часть других методов

## 4) Пирамидальная техника:

- Итерационная субдискретизация (блок пикселей заменяется одним пикселем с усредненным значением, получившееся изображение опять разбивается на блоки, заменяемые одним пикселем, и т.д.); в итоге, получаем одно значение (= усредненный уровень серого)
- Получаемые (и хранимые) «разностные» изображения, в которых значение каждого пиксела равны разнице между реальным и усредненным по блоку значениями, позволяют восстановить оригинальное изображение
- Своего рода обобщение иерархического блочного кодирования
- Естественный подход для постепенного (progressive) отображения (передачи) графики: сначала отображение в плохом разрешении, затем последовательно все в более лучшем

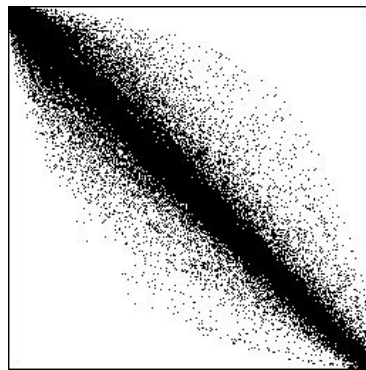
# Сжатие изображений

## 5) Векторное квантование:

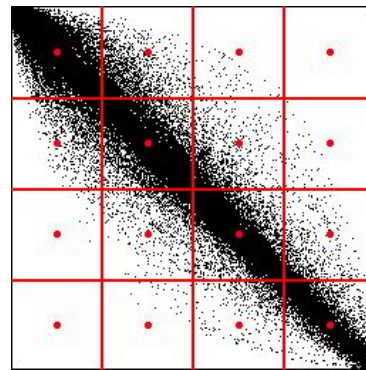
- Каждому блоку пикселей (например, двум соседним пикселям или блоку в 4x4 пиксела) ставится в соответствие вектор
- Выбираются наиболее типичные вектора, которые помещаются в «кодую книгу»
- Создается (и кодируется) индекс, который каждому блоку ставит в соответствие определенный вектор из кодовой книги
- Быстрая распаковка
- Различные эвристики для формирования кодовой книги
- Пример (блок – два соседние пиксела; двумерные вектора):



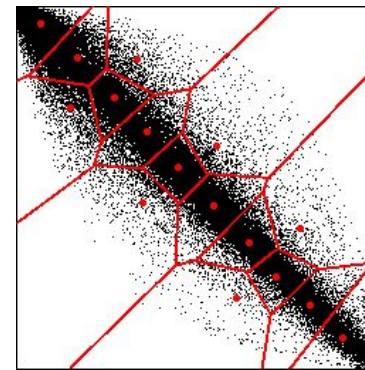
Полутоновая «Лена»



Представление пар пикселей  
в двумерном пр-ве [0,0]-[255,255]



Два бита на пиксел



Четыре бита на вектор  
(разбиение Вороного)

Взято - [http://www.gamasutra.com/features/20010416/ivanov\\_01.htm](http://www.gamasutra.com/features/20010416/ivanov_01.htm)

# Сжатие изображений

## 6) Преобразование:

- Рассмотрение частотных характеристик изображения
- Основной инструмент: дискретное Фурье-преобразование (DFT – discrete Fourier transformation)
- Для изображений более распространено: дискретное косинус-преобразование [1] (DCT)

Для блока  $N \times N$ :

$$C(u, v) = \alpha(u) \alpha(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \{ f(x, y) \cos[u\pi(2x+1)/2N] \cos [v\pi(2y+1)/2N] \}$$

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} \{ \alpha(u) \alpha(v) C(u, v) \cos[u\pi(2x+1)/2N] \cos [v\pi(2y+1)/2N] \}$$

$$\alpha(u) = \sqrt{1/N} \text{ for } N = 0, \alpha(u) = \sqrt{2/N} \text{ for } N = 1, 2, \dots$$

- Стандарт JPEG:
  - DCT применяется к блоку 8 на 8 пикселей, что дает 64 коэффициента
  - Низкие (самые важные) частоты находятся слева вверху
  - Коэффициенты квантуются (с помощью матрицы квантования) и кодируются
  - Высокая эффективность и качество; возможность постепенного отображения (передачи) графики



# Сжатие изображений

## 7) Вейвлеты:

- Учитываются не только частотные характеристики (как в DCT), но пространственное расположение
- Разные семейства базовых вейвлет-функций (например, Хаара, Добеши)
- Дискретное вейвлет-преобразование (DWT) используется в новом стандарте JPEG 2000
- JPEG 2000 превосходит JPEG

## 8) Фракталы:

- Фрактал - структура, состоящая из частей, которые в каком-то смысле подобны целому
- В некоторых изображениях (например, фотографиях природных объектов) части изображения могут быть похожими на другие части того же изображения
- Могут достигаться очень высокие коэффициенты сжатия
- Очень медленное сжатие, относительно быстрая распаковка
- Не является реальным конкурентом JPEG

# Сжатие изображений

## Сжатие цветных изображений

- Непосредственное расширение методов сжатия полутоновых изображений
- Простейшая идея: применить сжатие для полутонов к каждому из цветов (красный, синий, зеленый)
- Но тогда не будет учитываться высокая корреляция между цветами
- Лучшая идея: преобразовать в YUV-модель цвета (сделано в JPEG)
- Человеческий глаз наиболее чувствителен к яркости (компонента Y); компоненты цветности (U и V) менее важны и, значит, могут кодироваться менее точно (например, к ним можно применить субдискретизацию)
- Изображения с цветовой таблицей:
  - Методы сжатия полутоновых изображений не подходят (значения цветов не являются непрерывными)
  - Можно использовать методы сжатия для ASCII-файлов
  - Пример формата: GIF [1]

# Поиск по базе данных изображений

1. С помощью иерархической классификации изображений

При поиске пользователь использует иерархию, например:

*Художественные произведения*

*Живопись*

*Россия*

*19-ый век*

2. С помощью индекса признаков

Изображения рассматриваются как документы с индексом терминов

3. Поиск по содержимому

Поиск по шаблону, возвращающий изображения похожие на заданное изображение, фигуру и т.д.

# Поиск по базе данных изображений

## Извлечение признаков и индексация изображений

- Извлечение описательных атрибутов из изображений
- Вручную, автоматически или комбинированная схема (автоматическое сегментация и ручное добавление свойств)

Ручное (неавтоматическое) индексирование:

- Выполняется «экспертом», знакомым (обученным) с шаблонами (графическими образами) и словарем, которые используются в рассматриваемой базе данных изображений
- Множество экспертов: строгие правила непротиворечивости, общий глоссарий
- Распознавание образов может осуществляться автоматическими средствами
- Каждый заслуживающий внимания объект (пространственная структура) дополняется описательными атрибутами и вводится в систему вручную
- Помощь при выборе индексных терминов: иерархические словари, системы перекрестных ссылок, тезаурусы предметных областей
- Вреязатратно и дорого

# Поиск по базе данных изображений

## Автоматическое индексирование:

- Адаптировано к различным областям применения (распознавание документов, распознавание символов, технических чертежей, рентгенограмм и т.д.)
- В начале, система должна быть «обучена», а объекты рассматриваемой предметной области упорядочены по категориям
- Допускается определенная степень неопределенности (нечеткости)
- Важная область применения автоматического анализа изображений и распознавания объектов: преобразование бумажных документов в цифровую форму, и последующая индексация оцифрованных документов (→ **электронные библиотеки**)

## Сегментация:

- Выявление областей (представляющих интерес в каком-то отношении) в изображениях
- Сегмент – связная область, удовлетворяющая предикату однородности
- Основа для последующего поиска
- Одна из самых трудных задач обработки изображений
- Несколько возможных (эвристических) методов

# Поиск по базе данных изображений

Примеры предикатов однородности:

- Черно-белые изображения:  $p\%$  пикселей связной области имеют определенный цвет (черный или белый)
- Полутоновые изображения со сгруппированными по значениям пикселями, например – 0..9, 10-19, и т.д.: связная область однородна, если как минимум  $p\%$  ее пикселей относятся к одной группе значений
- Динамическая классификация полутонов: значения для каждого из классов не задаются предварительно, вместо задается размер интервала – значения серого у  $p\%$  пикселей должны быть в пределах  $\delta$  единиц
- Полутоновые изображения с заданной базовой функцией (для однородности): для как минимум  $p\%$  пикселей рассматриваемой области верно  $\text{grey\_level}(x, y) - f(x, y) < \delta$

Связная область:

- Для каждой пары пикселей -  $(x_1, y_1), (x_n, y_n)$  существует последовательность пикселей из данной области  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  таких, что пиксели  $\{(x_i, y_i), (x_{i+1}, y_{i+1})\}$  являются смежными (соседями) для любого  $i$

# Поиск по базе данных изображений

Разнообразные методы сегментации:

а) Сегментация на регулярные блоки:

- Пример: разбиение пространства с помощью тетрадного или двоичного дерева до тех пор пока не будут получены однородные области
- Как правило, не удовлетворяется условие максимальности для сегментации: соседние блоки могут образовывать одну однородную область
- Обобщение сегментации с помощью двоичного дерева: разбиение может производиться в любом направлении – сегментация на полигоны

Компромиссное решение: разбиение только в направлениях  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  и  $135^\circ$

б) Разбиение и слияние:

- Дополнение методов а) для выполнения условия максимальности
- Парная тестовая проверка полученных регионов на однородность
- Обычно, сегментация для произвольного предиката однородности не является уникальной

# Поиск по базе данных изображений

## в) Нарастивание областей:

- Начать с набора затравочных (начальных) точек (seed points)
- Нарастивать однородные области, рассматривая пиксели, примыкающие к (сначала) затравочным точкам и (далее) к растущим однородным областям
- Основное затруднение: как выбирать затравочные точки?

## г) Движение по контуру:

- Движение по контуру объекта (области), которую предполагается обнаружить; контур можно определить, следуя вдоль наибольшего градиента (изменений в значениях уровня серого)

## д) Сравнение с пороговым значением:

- Применимо, если представляющие интерес объекты и фон изображения имеют достаточно отличающиеся значения уровней серого
- Гистограмма значений серого для изображения имеет два или более пиков, между которыми можно выбрать пороговые значения для уровня серого
- Должна быть, как правило, дополнена более сложными методами



# Поиск по базе данных изображений

## Поиск по содержанию в базах данных изображений

Общее свойство:

- Не стопроцентно точный поиск

Виды запросов:

- Найти изображения с определенными признаками (цвет, текстура, фигура и т.д.)
- Найти изображения, содержащие определенный тип(-ы) объектов
- Найти объекты (в изображениях) с определенными атрибутами, например, определенной фигуры (круг, треугольник и т.д.), размера, цвета и т.д.
- Найти изображения, в которых объект типа А расположен слева от объекта типа Б (пространственное отношение)
- Поиск по сходству: найти изображения (сегменты), похожие на заданное изображение (сегмент)

Приложения: распознавание/идентификация людей по фотографиям или по отпечаткам пальцев; распознавание военных объектов (самолетов, кораблей и т.д.)

# Поиск по базе данных изображений

Подходы к организации поиска по сходству:

а) Метрический подход:

- Вводится функция расстояния (метрическая функция) для изображений (сегментов)
- Задача: найти ближайшего соседа ( $k$  ближайших соседей) к заданному в запросе изображению (сегменту)
- Простейшая метрическая функция для  $m \times n$  цветных изображений:
  - Сумма евклидовых расстояний между попарными значениями пикселей (в трехмерном пространстве)
  - Евклидово расстояние в  $(m \times n \times 3)$ -мерном пространстве
- Требуется большой объем вычислений
- Упрощение:

Для уменьшения размерности использовать извлечение признаков: например, применить DCT (см. «Сжатие изображений») и далее для вычисления расстояний использовать полученные коэффициенты; новая метрическая функция  $d'$  должна удовлетворять (приблизительно) условию:  $d'(a, b) < d'(a, c)$ , если  $d(a, b) < d(a, c)$

# Поиск по базе данных изображений

## б) Трансформационный подход:

- Один из видов метрического подхода
- Основная идея: степень несходства пропорциональна минимальной «цене» трансформации одного изображения (сегмента) в другое
- Выбрать изображение, которое наименее непохоже на заданное в запросе изображение
- Примеры трансформационных операторов: сдвиг (смещение), вращение, масштабирование (увеличение, уменьшение), растяжение, закрашивание и т.д.
- У каждого оператора своя функция стоимости
- Общая цена трансформации – сумма стоимостей каждой из операций

Определить для изображения последовательность трансформационных шагов с наименьшей стоимостью, затем найти минимум по всем рассматриваемым изображениям

- Более настраиваемый подход (по сравнению с метрическим); пользователи могут задавать свои собственные трансформационные операторы и функции стоимости
- Метрический подход обеспечивает лучшую индексацию

# Поиск по базе данных изображений

Пример: Система QBIC [4] (IBM's Query By Image Content)

- Поиск по содержанию изображений
- Извлечение признаков при наполнении базы данных:
  - Расположение цветов/текстур
  - Идентификация объектов: ручная, полуавтоматическая, автоматическая сегментация
- Графически задаваемые запросы, основанные на:
  - Изображениях-образцах
  - набросках, эскизах, введенных пользователем (параметры фигур)
  - Цветах (основной цвет или цветовая гистограмма)
  - Структуре текстур (крупнозернистость, контраст, направленность)
  - Движении камеры или объекта (для видео)
- Быстрый поиск:
  - Фильтрация и индексация
  - Уменьшение размерности с помощью преобразований (индексация на основе  $R^*$ -дерева)

# Структуры баз данных для изображений

- Последовательное хранение матрицы пикселей (возможно в сжатом виде), занимающей в большинстве случаев несколько дисковых страниц
  - Каждое изображение можно рассматривать как один самостоятельный файл
- а) Реляционное представление:
- Представление изображения: идентификатор изображения и его основные свойства (атрибуты)
  - Представление объекта: объекты (сегменты, прямоугольники) внутри изображений; извлекаются вручную или автоматически  
Атрибуты включают: id изображения, id объекта, координаты минимального ограничивающего прямоугольника, признаки
  - Обобщение: вероятностные отношения - объект  $x$  находится в изображении  $i$  с вероятностью  $p$
  - Запросы: применять стандартную технику запросов, используя значения признаков в условиях запроса

# Структуры баз данных для изображений

б) Пространственное представление:

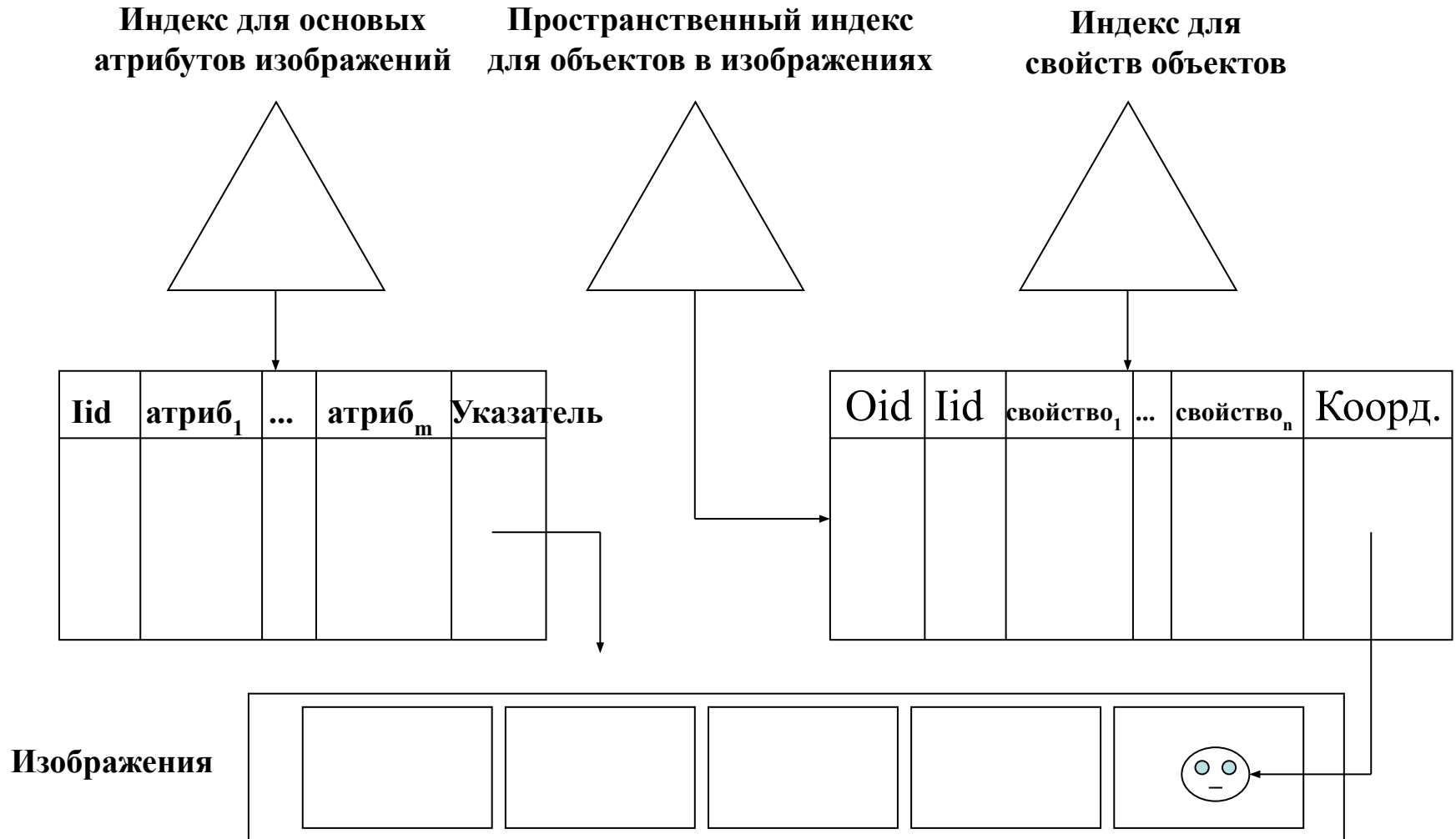
- Например, с помощью R- или R\*-деревьев
- Построить одно R-дерево для всех изображений в базе данных
- Страница, соответствующая листу, содержит близко-расположенные объекты (их MBR'ы) со списком указателей на исходные изображения
- Также сохранены дополнительные свойства (признаки) объекта
- Для не пространственных свойств объектов может быть построен отдельный индекс

Общие замечания:

- При отсутствии «пространственного» контекста, изображения могут рассматриваться как документы, и процесс выборки данных может осуществляться с помощью общих методов информационного поиска
- Совместное использование пространственных и не пространственных критериев при поиске достигается за счет совмещения (объединение, пересечение, ...) списков указателей из соответствующих индексов

# Структуры баз данных для изображений

## Примерная архитектура базы данных для изображений



# Упражнения

1. Применить пирамидальное кодирование (см «пирамидальная техника») к приведенному ниже полутоновому изображению размером в 4 на 4 пикселя. Затем восстановить (раскодировать) изображение. Диапазон значений серого – 0..255. Кодирование в двоичные значения опустить.

83	88	91	92
79	86	96	101
71	83	95	108
63	75	90	107

2. Для приведенного ниже полутонового изображения произвести сегментацию на регулярные блоки (разбиение с помощью, например, тетрадного дерева). Далее, проверить соседние области на однородность и слить, где возможно. Область однородна, если разница между наибольшим и наименьшим значением пикселя не более 10.

75	77	78	82	83	92	94	95
76	80	80	84	85	94	96	97
78	82	83	87	90	94	97	101
86	89	89	91	90	95	99	104
87	87	90	93	89	94	98	99
91	89	91	92	88	92	96	97
94	93	94	91	86	88	87	94
97	96	96	93	84	83	83	85



# Ссылки на литературу

- [1] J. Miano. Compressed Image File Formats: JPEG, PNG, GIF, XBM, BMP. ACM Press, Addison-Wesley Professional, 1999
- [2] [http://en.wikipedia.org/wiki/Category:Graphics\\_file\\_formats](http://en.wikipedia.org/wiki/Category:Graphics_file_formats)
- [3] ITU-T Recommendation T.82. Information Technology - Coded Representation of Picture and Audio Information - Progressive Bi-level Image Compression, 1993
- [4] <http://wwwqbic.almaden.ibm.com/>