

Компьютерная обработка изображений

Лекция **1**: Введение. Основные понятия.
Наложение изображений.

Сафонов И.В., Крыжановский К.А., Егорова М.А.

2011

1

Цель курса

Цель курса - изучение базовых методов и алгоритмов цифровой обработки изображений.

В результате освоения дисциплины студент должен:

Знать: Основные понятия, математический аппарат и алгоритмы цифровой обработки изображений.

Уметь: Применять изученные методы для решения задач улучшения качества изображений, измерения параметров изображений, распознавания образов. Выполнять синтез комплексных алгоритмов обработки изображений.

Продemonстрировать способность реализовывать освоенные алгоритмы обработки изображений на одном из языков программирования высокого уровня (C/C++, Delphi, Java, C# и т.п.)

2

Структура и содержание курса

- Лекции

- Основные понятия
- Анализ изображений
- Преобразования между типами изображений
- Геометрические преобразования
- Свертка в пространственной и частотной областях, типы фильтров
- Восстановление и улучшение изображений, фильтрация шумов
- Повышение резкости, изменение контраста
- Сегментация изображений
- Морфологические операции
- Вычисление признаков
- Основы распознавания образов

- Лабораторные работы

- Индивидуальное домашнее задание

- Экзамен

3

Рекомендуемая литература

1. Р. Гонсалес, Р. Вудс "Цифровая обработка изображений", М.: Техносфера, 2005.
2. Л. Шапиро, Дж. Стокман "Компьютерное зрение", М.: БИНОМ, 2006.
3. Б.Яне "Цифровая обработка изображений", М.: Техносфера, 2007.
4. И.С.Грузман, В.С.Киричук, В.П.Косых, Г.И.Перетягин, А.А.Спектор "Цифровая обработка изображений в информационных системах", Новосибирск:НГТУ, 2002.
5. Р. Гонсалес, Р. Вудс, С. Эддинс "Цифровая обработка изображений в среде MATLAB" М.: Техносфера, 2006.
6. У.Прэтт "Цифровая обработка изображений" в 2-х книгах, М.: Мир, 1982.
7. Т.Павлидис "Алгоритмы машинной графики и обработки изображений", М.: Радио и связь, 1986.
8. П.И.Рудаков, И.В.Сафонов "Обработка сигналов и изображений: Matlab 5.X", М.: Диалог-МИФИ, 2000.

4

Замечание о терминах

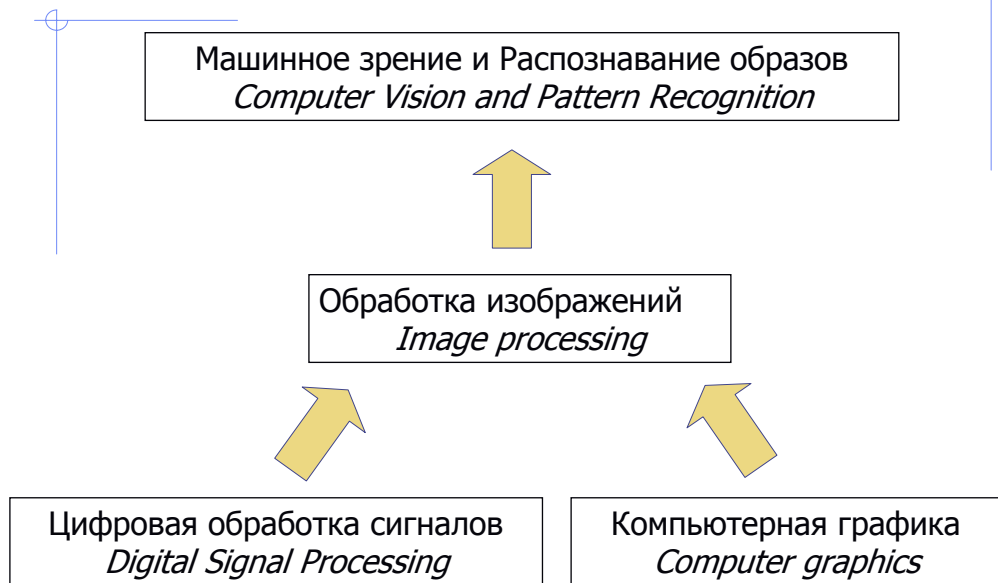
Обработка изображений бурно развивается, и в настоящее время нет устоявшихся русскоязычных названий для целого ряда терминов.

Например, *co-occurrence matrix* в [7] называется матрицей совместной встречаемости, в [6] - гистограммой второго порядка, в [2] - матрицей вхождений.

Поэтому для большинства терминов в лекциях также приводятся англоязычные термины.

5

Связь с другими дисциплинами



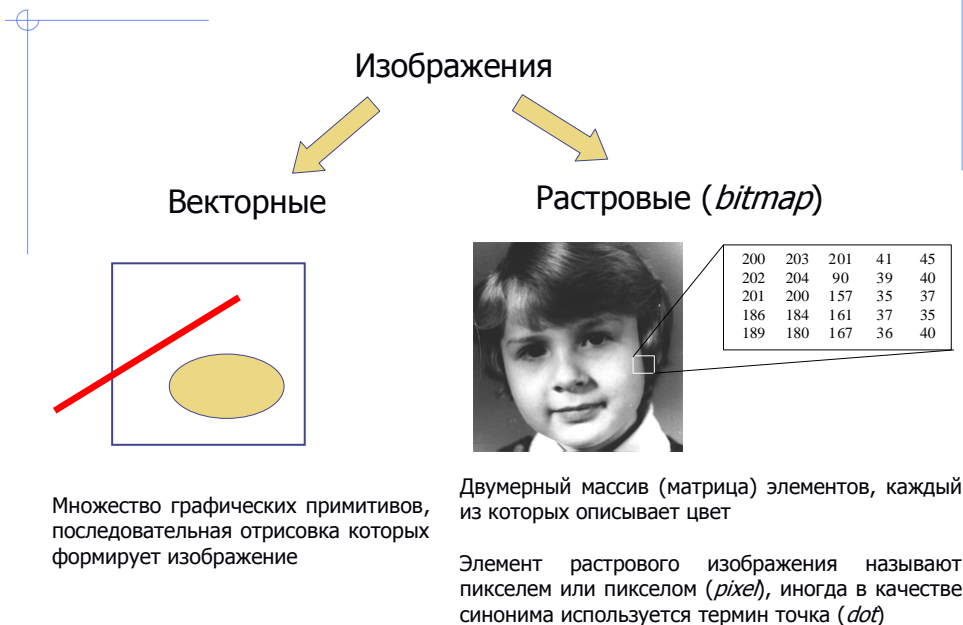
6

Области применения

- Бытовая электроника
 - фотокамеры, видеокамеры, камеры мобильных телефонов и т.п.
 - телевизоры
 - принтеры и МФП
 - оборудование для видеоконференций
- Медицинская техника
 - обработка рентгеновских снимков
 - томография и УЗИ
 - лабораторные микроскопические исследования
- Системы анализа качества в промышленности
- Полиграфическое оборудование и системы фотопечати
- Обработка данных дистанционного зондирования Земли
- Системы безопасности
 - системы ограничения доступа с распознаванием лиц
 - системы видеонаблюдения с распознаванием номеров
 - аппаратура анализа дорожной ситуации
- и многое другое ...

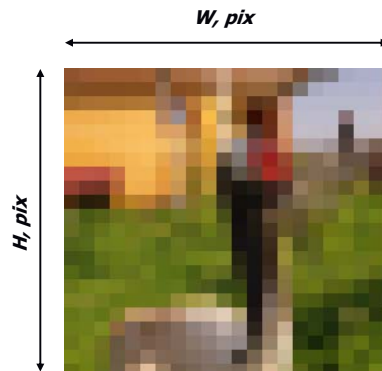
7

Растровые изображения



8

Размер и глубина цвета



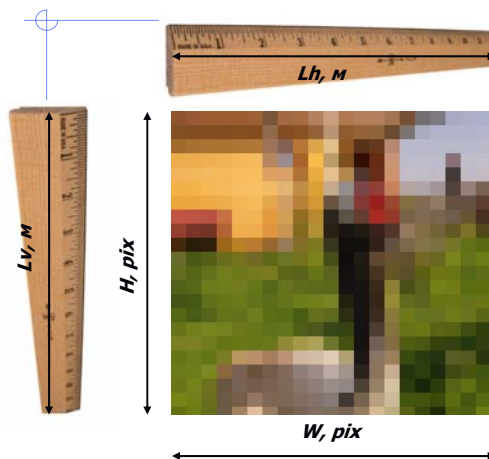
Размер изображения (*image size*) в пикселах определяется количеством строк или высотой (*height*) H и количеством столбцов или шириной (*width*) W .

Глубина цвета (*color depth*) D обозначает количество битов необходимых для хранения информации о цвете пиксела; измеряется глубина цвета в битах на пиксел (*bits per pixel, bpp*);

Если для описания цвета используется несколько цветовых компонент (каналов), например *RGB*, то для каждого канала существует количество битов на канал (*bits per channel*): D_r, D_g, D_b

9

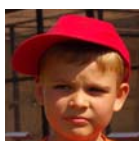
Разрешение



Разрешение (*resolution*) связывает размер изображения в пикселах с его реальными физическими размерами. Разрешение указывается в количестве пикселей, приходящихся на единицу длины, например, количество точек на метр или на дюйм.

Горизонтальное разрешение:
 $R_h = W/L_h$

Вертикальное разрешение:
 $R_v = H/L_v$



$R_h == R_v$, разрешение изотропно и у изображения "квадратный пиксел"



$R_h != R_v$, разрешение анизотропно и у изображения "неквдратный пиксел"

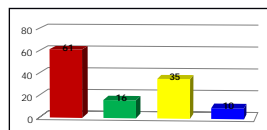
10

Типы изображений

В зависимости от способа кодирования информации о цвете пиксела и глубины цвета выделяют следующие типы изображений:

- Бинарные (*Black and White*)
- Полутоновые (*Grayscale, Intensity*)
- Палитровые (Indexed)
- Полноцветные или цветные (*True color or color*)

general problem of bo
weighted majority vot
on function and a sm
m for selecting a s
nificant variety. Drawin
ive procedure for sear
nificant variety.



11

Бинарные изображения

general problem of bo
weighted majority vot
on function and a sm
m for selecting a s
nificant variety. Drawin
ive procedure for sear
nificant variety.

Пиксели бинарного изображения могут принимать только два значения: 0 и 1, которые обозначают черный и белый цвет (или наоборот). Иногда значения бинарных изображений рассматривают как логические величины, обозначающие состояние "выключено" / "включено" или "ложь" / "истина". Глубина цвета D составляет 1 bpp.

Бинарные изображения используются в случаях, когда важна форма объектов или их расположение на изображении, а цвет объектов не представляет интереса. Бинарные изображения требуют минимального объема памяти. Например, в бинарных изображениях имеет смысл хранить отсканированные изображения текста и чертежей. Также бинарные изображения широко используются в качестве промежуточного способа хранения информации в сложных комплексных алгоритмах обработки изображений.

12

Полутоновые изображения



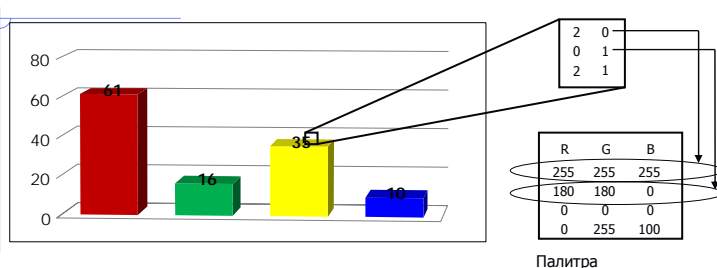
Полутоновым называют изображение, пиксеты которого могут принимать одно из значений интенсивности в диапазоне от 0 до 2^D-1 , где D – глубина цвета. Обычно считают, что в полутоновом изображении хранятся градации серого цвета в диапазоне от черного до белого. Поэтому иногда полутоновые изображения называют серыми или изображениями в градациях (в шкале) серого, а в качестве синонима интенсивности используется термин яркость пиксела.

В настоящее время на практике используют полутоновые изображения с D от 2 до 16 bpp. Наиболее широкое распространение имеют полутоновые изображения 8 bpp, каждый пиксел которых может отображать 256 градаций яркости (от 0 до 255).

С полутоновыми изображениями работают в том случае, когда необходима информация только о яркости, а сигналы цветности не представляют интереса, или информация о цветовых тонах изначально отсутствует, как, например, для рентгеновских снимков.

13

Палитровые изображения



В отличие от полутоновых изображений, в которых значение цвета непосредственно соответствует его яркости, в палитровых изображениях значения пикселей являются ссылками (индексами) в определенные строки карты цветов (*color map*) или палитры. Палитра является двумерным массивом, в столбцах которой расположены интенсивности цветовых составляющих, а количество строк, т.е. количество различных цветов в изображении, меньше либо равно 2^D .

Палитровые изображения выгодно использовать в том случае, когда необходимо сократить объем памяти для хранения цветных изображений, при этом частью оттенков можно пренебречь, или цветные изображения изначально используют относительно малое количество цветов. Такие изображения можно эффективно использовать в большинстве задач презентационной и оформительской графики.

14

Цветные изображения



Полноцветными называют изображения, пиксели которых непосредственно хранят информацию об интенсивности цветовых компонент. Каждый пиксел имеет не одно значение, а как минимум 3, в зависимости от используемой цветовой системы (*color space*). Эти изображения содержат полную информацию о цвете, но для хранения требуют значительных объемов памяти.

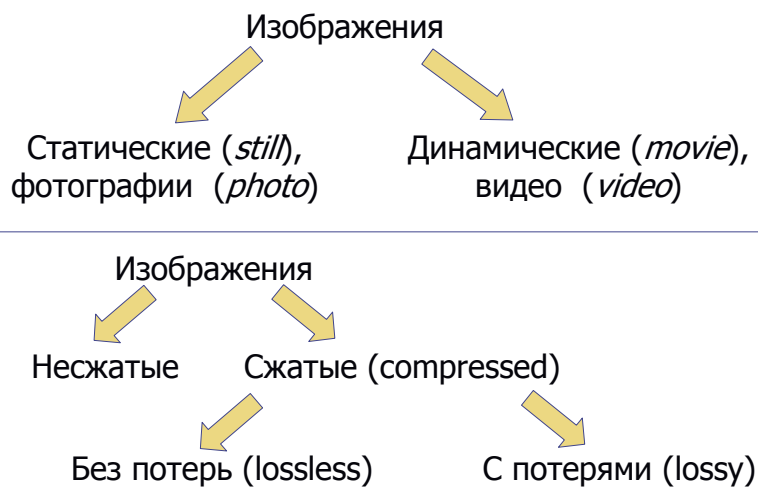
Для цветовой системы RGB наиболее широкое распространение получили цветные изображения с глубиной цвета 15 bpp (5:5:5 bits per channel), 16 bpp (5:6:5), 24 bpp (8:8:8), 30 bpp (10:10:10), 36 bpp (12:12:12).

Полноцветные изображения следует использовать в том случае, когда нельзя терять информацию об оттенках цвета, например, при обработке и печати фотографий и иллюстраций в полиграфии.

15

Иные способы разбиения изображений на типы

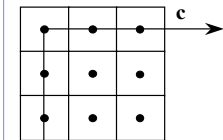
Существует еще множество способов классификации изображений по типам.



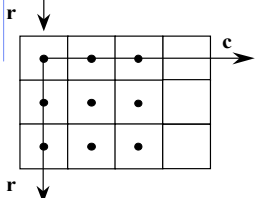
16

Хранение в памяти

Существует несколько вариантов хранения изображений в памяти.



Строки пикселей в памяти расположены друг за другом подряд без промежутков. Пиксели в строке идут слева направо, строки сверху вниз.



Для повышения эффективности обработки адрес каждой строки изображения может быть выровнен на 4, или 8, или 16, или 32, или 64 байта. Это достигается путем дополнения каждой строки (*padding*) необходимым количеством незначащих (часто нулевых) байтов.

При хранении цветных изображений составляющие цвета пиксела часто располагают в соседних ячейках памяти, изображение хранят в формате *RGBRGBRGB...* или *BGRBGRBGR...* (тройками или триплетами). Для повышения эффективности работы с памятью каждый пиксел может занимать 32 или 64 бита. Дополнительные биты используют для служебных целей или не используют и обозначают каналом *A*. Изображение тогда хранится в *RGBA* формате (четверками).

Также иногда хранят все значения одной составляющей, потом все значения другой и так далее, то есть изображение хранят в формате *RRR...GGG...BBB...* или похожем (по-канально).

17

Обращение к пикселям изображения

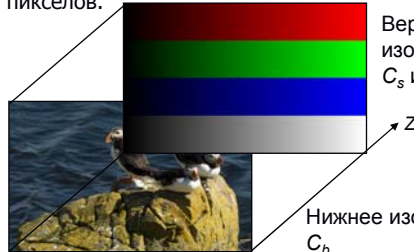
Несмотря на то, что изображение является по крайней мере двумерным массивом, с точки зрения программной реализации на C/C++ целесообразно работать с ним как с массивом одномерным. Для примера рассмотрим функцию вычисления средней яркости для полутонового изображения 8 bpp, строки выровнены, и каждая занимает *iWidthBytes* байт.

```
int AverageBrightness4Gray8bpp(const unsigned char* pbImg, int iHeight,
                               int iWidth, int iWidthBytes)
{
    unsigned long ulSum = 0;
    const unsigned char* pbCurRow = pbImg;
    for (int y = 0; y < iHeight; ++y)
    {
        for (int x = 0; x < iWidth; ++x)
        {
            // все 3 варианта ниже дают одинаковый результат
            ulSum += pbCurRow[x];
            // ulSum += *(pbCurRow + x);
            // ulSum += *(pbImg + y*iWidthBytes + x);
        }
        pbCurRow += iWidthBytes;
    }
    return (int)(ulSum/(iHeight*iWidth));
}
```

18

Наложение изображений

Алгоритмы наложения (*compositing*) изображений позволяют генерировать одно изображение из нескольких, расположенных одно под другим. Изображения имеют одинаковый размер и комбинируются попарно. Пиксел с координатами (x, y) нижнего изображения комбинируется с пикселом (x, y) верхнего независимо от значений соседних пикселов.



Верхнее изображение C_s и α

Верхнему изображению соответствует значение α (альфа) в диапазоне от 0 до 1, которое интерпретируется как прозрачность (*transparency*) или непрозрачность (*opacity*).

Результат наложения:
 $C_r = (1 - \alpha) \times C_b + \alpha \times C_s$



$\alpha = 1$, полная непрозрачность, видно только верхнее изображение



$\alpha = 0$, полная прозрачность, видно только нижнее изображение

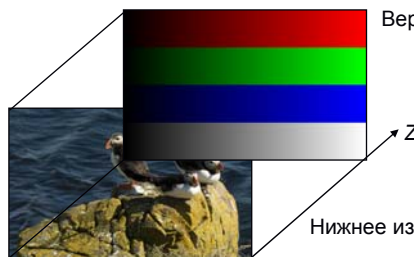


$\alpha = 0.5$, сквозь пиксели верхнего изображения "просвечивают" пиксели нижнего

19

Альфа канал

Каждому пикселу верхнего изображения может соответствовать собственное значение α . Все эти значения α образуют так называемый альфа-канал. Иногда альфа-канал называют маской. Такие маски широко используют для отображения изображений произвольной формы.



Верхнее изображение C_s

Нижнее изображение C_b

Альфа-канал

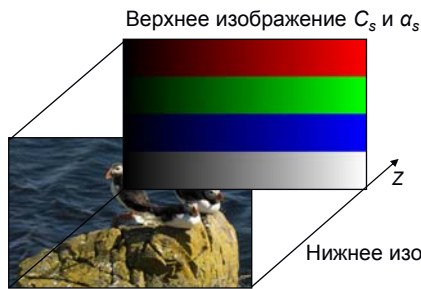


Результат наложения

20

Обобщенная формула наложения

В общем случае верхнему изображению соответствует α_s (одно значение или альфа-канал), нижнему изображению соответствует α_b (значение или альфа-канал), кроме того существует функция смешивания (*blending*) $B(C_b, C_s)$ пикселей верхнего и нижнего изображений:

$$C_r = [(1 - \alpha_s) \times \alpha_b \times C_b] + [(1 - \alpha_b) \times \alpha_s \times C_s] + [\alpha_b \times \alpha_s \times B(C_b, C_s)]$$


$B(C_b, C_s)$ - функция (или режим смешивания), таких функция может быть много, ниже приведено несколько примеров



Пример наложения:
 $B(C_b, C_s) = C_b \times C_s$,
 $\alpha_b = 1$,
 $\alpha_s = 0.5$

Наложение используется не только в задачах художественного дизайна, но и при улучшении качества изображений, когда, например, изображение смешивается со своей отфильтрованной копией.

21

Режимы смешивания (1)

Normal – выбирается только верхнее изображение

$$B(C_b, C_s) = C_s$$



Multiply – результат темнее смешиваемых цветов

$$B(C_b, C_s) = C_b \times C_s$$



Screen – результат светлее смешиваемых цветов

$$B(C_b, C_s) = 1 - (1 - C_b) \times (1 - C_s)$$



Во всех приведенных формулах значения пикселей нормированы в диапазон от 0 до 1. Цветовые каналы обрабатываются независимо друг от друга.

22

Режимы смешивания (2)

Darken – из интенсивностей пикселей верхнего и нижнего изображений выбирается меньшая интенсивность

$$B(C_b, C_s) = \min(C_b, C_s)$$



Lighten – из интенсивностей пикселей верхнего и нижнего изображений выбирается большая интенсивность

$$B(C_b, C_s) = \max(C_b, C_s)$$



Difference – разница интенсивностей

$$B(C_b, C_s) = |C_b - C_s|$$



23

Режимы смешивания (3)

ColorDodge – светлые участки верхнего изображения усиливают яркость нижнего, а темные участки не оказывают никакого воздействия.

$$B(c_b, c_s) = \begin{cases} \min(1, c_b / (1 - c_s)) & \text{if } c_s < 1 \\ 1 & \text{if } c_s = 1 \end{cases}$$



ColorBurn – темные участки верхнего изображения уменьшают яркость нижнего.

$$B(c_b, c_s) = \begin{cases} 1 - \min(1, (1 - c_b) / c_s) & \text{if } c_s > 0 \\ 0 & \text{if } c_s = 0 \end{cases}$$



SoftLight – эффект аналогичен освещению нижнего изображения диффузионным точечным источником

$$B(c_b, c_s) = \begin{cases} c_b - (1 - 2 \times c_s) \times c_b \times (1 - c_b) & \text{if } c_s \leq 0.5 \\ c_b + (2 \times c_s - 1) \times (D(c_b) - c_b) & \text{if } c_s > 0.5 \end{cases}$$

$$D(x) = \begin{cases} ((16 \times x - 12) \times x + 4) \times x & \text{if } x \leq 0.25 \\ \sqrt{x} & \text{if } x > 0.25 \end{cases}$$



24