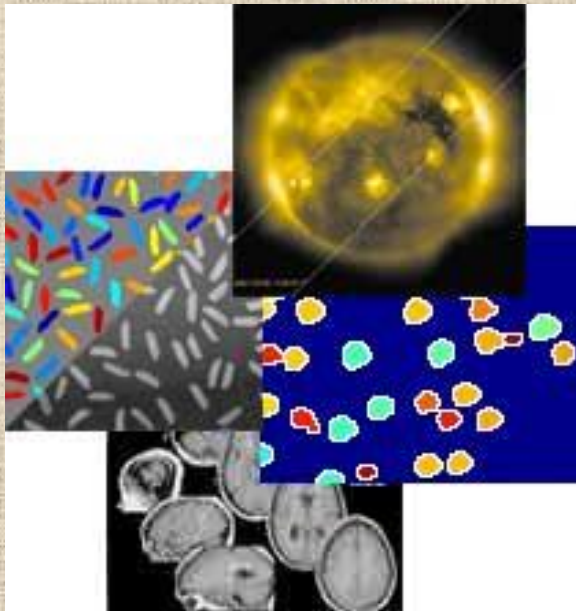

Преобразование растра



Будак Владимир Павлович,
Национальный исследовательский
университет «МЭИ»
кафедра светотехники

☐: +7 (095) 362-7067

BudakVP@mpei.ru

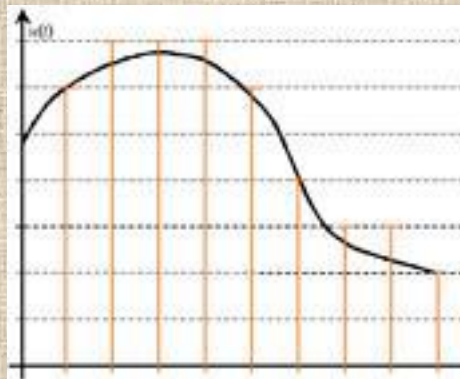
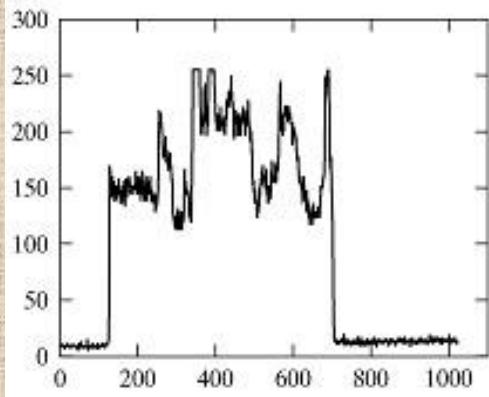
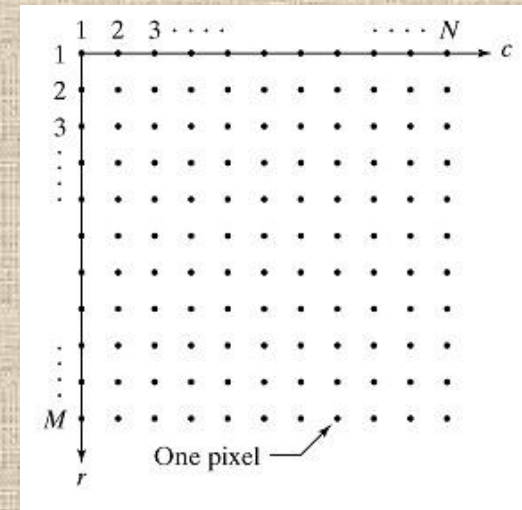


Цифровое изображение



Любое изображение представимо двумерной функцией $L(x,y) \equiv L(\mathbf{r})$ – яркость от координат на плоскости $\mathbf{r} = \{x,y\}$

Однако непрерывное изображение не может быть размещено в памяти и обрабатываться процессором – дискретизация – теорема Whittaker-Shannon-Котельников



- цифровое изображение представимо двумерным массивом

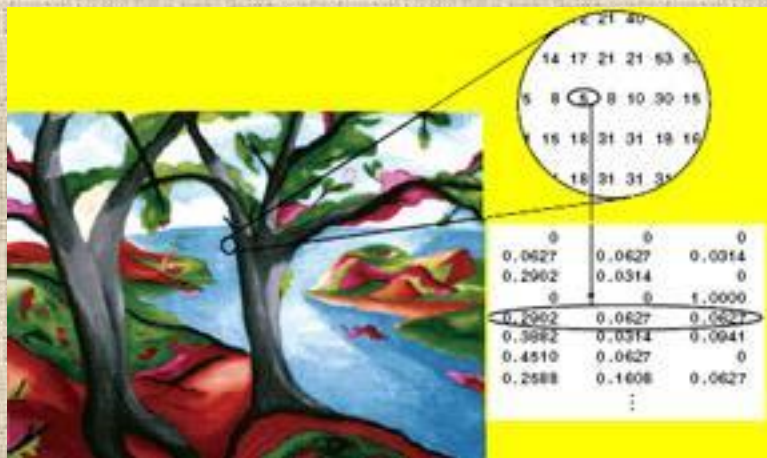
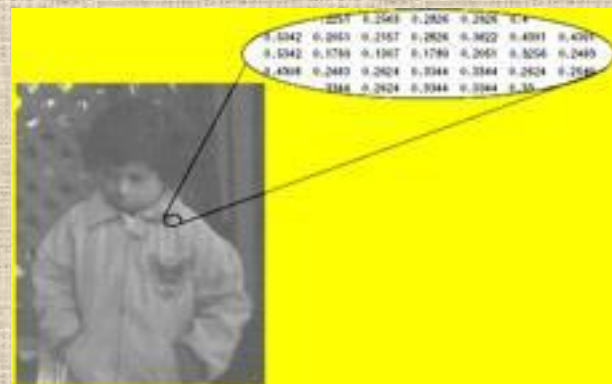
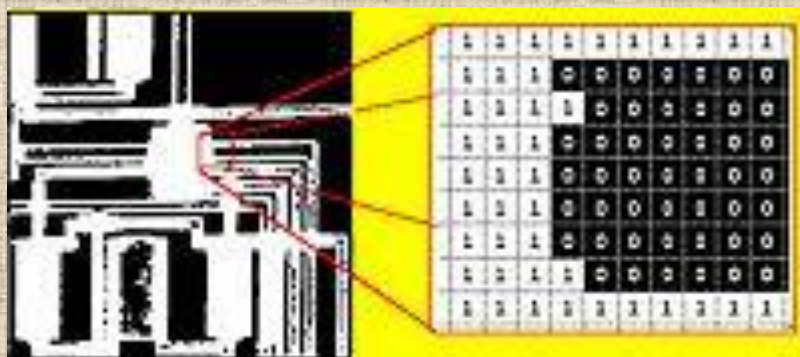
Дискретизованное и квантованное изображение – цифровое изображение

Типы цифровых изображений

Binary Двоичное	Логический массив содержащий только 0 и 1, представляющие белый или черный цвет - двухуровневое (<i>bilevel</i>) или черно-белое изображение (<i>black and white</i>)
Indexed Индексное	Цвет каждого пикселя определяется номером из цветовой таблицы – палитры (<i>colormap</i>): массив из $m \times 3$ - псевдоцветное изображение (<i>pseudocolorimage</i>)
Intensity Яркостное	Массив значений яркости пикселей – градации серого (<i>grayscale</i>)
Truicolor Полноцветное	Массив $m \times n \times 3$ яркостей трех основных цветов RGB – RGB-изображение (<i>RGB image</i>)

Любому двумерному массиву можно сопоставить изображение, а любому изображению - массив

Структура изображений



С изображением возможны любые операции допустимые для массивов

Типы данных

Name	Description
double	Double-precision, floating-point numbers in the approximate range -10^{308} to 10^{308} (8 bytes per element).
uint8	Unsigned 8-bit integers in the range [0, 255] (1 byte per element).
uint16	Unsigned 16-bit integers in the range [0, 65535] (2 bytes per element).
uint32	Unsigned 32-bit integers in the range [0, 4294967295] (4 bytes per element).
int8	Signed 8-bit integers in the range [-128, 127] (1 byte per element).
int16	Signed 16-bit integers in the range [-32768, 32767] (2 bytes per element).
int32	Signed 32-bit integers in the range [-2147483648, 2147483647] (4 bytes per element).
single	Single-precision floating-point numbers with values in the approximate range -10^{38} to 10^{38} (4 bytes per element).
char	Characters (2 bytes per element).
logical	Values are 0 or 1 (1 byte per element).

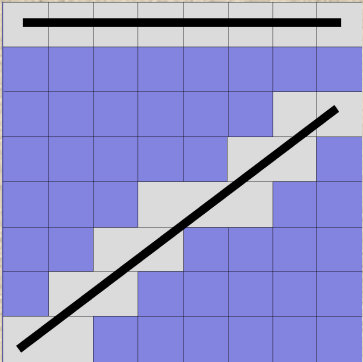
При single или double значения цвета изменяются в [0, 1]

Преобразование изображений

dither	Двоичное из яркостного или полно цветного полутоновым растриванием
gray2ind	Индексное из яркостного
grayslice	Индексное из яркостного по порогам
im2bw	Индексное из произвольного по порогам
ind2gray	Яркостное из индексного
ind2rgb	Полноцветное из индексного
mat2gray	Яркостное из матрицы масштабированием
rgb2gray	Яркостное из полноцветного
rgb2ind	Индексное из полноцветного

При преобразованиях почти всегда потеря качества за исключением преобразований в RGB

Вычерчивание отрезков



- Устройство раstra - матрица дискретных пикселей
- Каждый пиксель может быть подсвечен только целиком
- Нельзя провести отрезок из одной точки в другую
- Определения пикселей наилучшим образом аппроксимирующий отрезок - разложение в растр

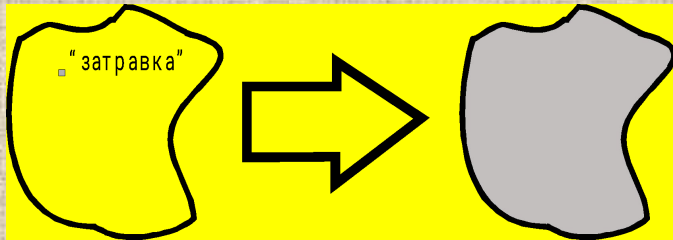
```
% Инициализация переменных
x = x1;
y = y1;
dx = x2 - x1;
dy = y2 - y1;
k = dy/dx;
e = k - 0.5;
```

```
% Вычерчивание отрезка
for i = 1: dx
    Pset(x, y);
    while e >= 0
        y = y + 1;
        e = e - 1;
    end
    x = x + 1
    e = e + k
```

end

Алгоритм Bresenham J.E. является алгоритмом оптимального выбора растровых координат для представления отрезка

Простой алгоритм заполнения с затравкой



- 4-связанные: от внутренней точки области заполнения можно достигнуть, двигаясь по 4 направлениям – вверх-вниз и влево-вправо;
- 8-связанные: из одной до другой точки, двигаясь по 8 направлениям - вверх-вниз, влево-вправо и 4 по двум диагоналям.

```
function FloodFill4(x, y, Oldval, Newval)
    if Pixel(x,y) = Oldval
        PSet(x, y, Newval);
        FloodFill4(x, y+1, Oldval, Newval);
        FloodFill4(x, y-1, Oldval, Newval);
        FloodFill4(x+1, y, Oldval, Newval);
        FloodFill4(x-1, y, Oldval, Newval);
    end
```

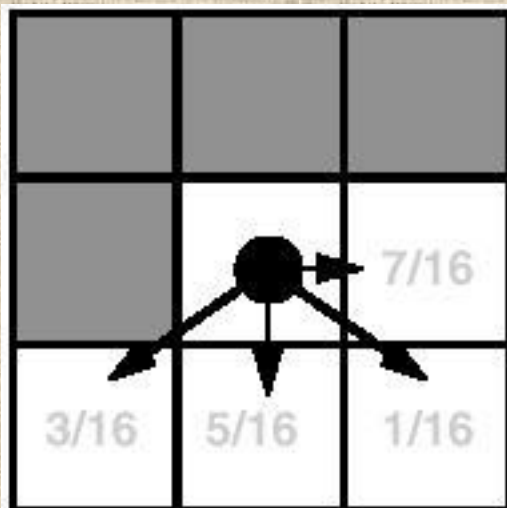
end

Главное достоинство растровых устройств является возможность представления сплошных конечных областей

Полутоновая аппроксимация изображений

Пороговый метод: если интенсивность пикселя $I(x,y)$ изображения превышает порог, то пиксель выводится белым, в противном случае черный:

if $I(x,y) > \text{порог}$, Белый, else Черный;

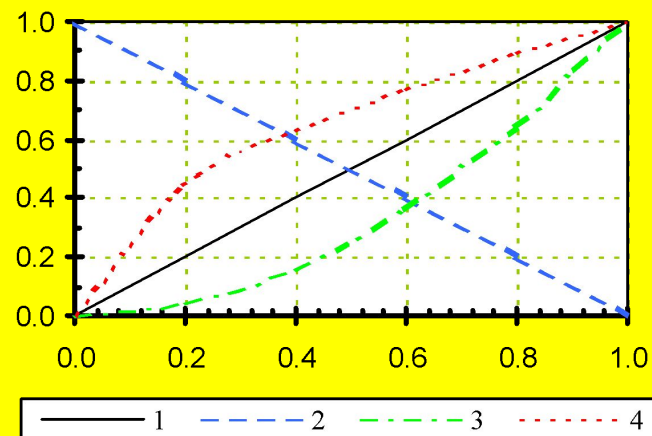


- В методе Флойда-Стейнберга эта ошибка распределяется на соседние пиксели.
- Такое распределение ошибки на соседние пиксели существенно улучшает изображение, т.к. информация, заключенная в изображении, не теряется полностью.

*При полутоновой аппроксимации цветных изображений –
тонирование для каждого из трех основных цветов*

Поточечная обработка

$f(v) = v$	Идентичное преобразование – 1
$f(v) = 1-v$	Переход к негативу: черное → белое, белое → черное – 2
$f(v) = v^p, p > 1$	Затенение изображения – 3
$f(v) = v^p, p < 1$	Увеличение яркости – 4



Функция $f(v) = kv$ приводит к изменению контраста, причем при $k > 1$ контраст увеличивается, а при $k < 1$ – уменьшается

Фильтрация

$$L_o(x, y) = L_o(\mathbf{r}) = \int_{\text{по входному изображению}} L_i(x', y') h(x - x', y - y') dx' dy' = \int_{\text{по входному изображению}} L_i(\mathbf{r}') h(\mathbf{r} - \mathbf{r}') d^2 r'$$

$$G_o(\mathbf{k}) = \int e^{i\mathbf{k}\mathbf{r}} \int L_i(\mathbf{r}') h(\mathbf{r} - \mathbf{r}') d^2 r' d^2 r = \int L_i(\mathbf{r}') e^{i\mathbf{k}\mathbf{r}} \int h(\mathbf{r} - \mathbf{r}') e^{i\mathbf{k}(\mathbf{r}-\mathbf{r}')} d^2 r d^2 r' = G_i(\mathbf{k}) H(\mathbf{k})$$

$$L_o(m, n) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} L_i(u, v) h(m-u, n-v) = \sum_{u=-\infty}^{+\infty} \sum_{v=-\infty}^{+\infty} L_i(m-u, n-v) h(u, v)$$

явная запись фильтра:

$$L_i(m, n) = (L_o(m-1, n-1) + L_o(m, n-1) + L_o(m+1, n-1) + L_o(m-1, n) + L_o(m, n) + L_o(m+1, n) + L_o(m-1, n+1) + L_o(m, n) + L_o(m+1, n+1))/9$$

матричная форма:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

В компьютерной графике часто саму операцию свертки называют фильтрацией, а функцию $h(m, n)$ фильтром

Фильтрация

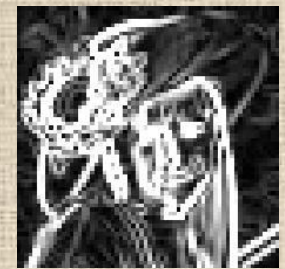


Исходный растр
 64×64

Смаз 3×3

Смаз 5×5

Горизонтальная
производная



Вертикальная
производная

Лапласиан

Четкость =
исходное + c * лапласиан

Фильтр резкости

*Свертку с матрицей коэффициентов фильтра
не путать с матричным умножением*

Геометрические преобразования растра

Геометрические преобразования растра изменяют позиции пикселей в изображении:

- Смещение - сместить каждый пиксель по горизонтали и вертикали
- Масштабирование – растяжение или сжатие изображения
- Поворот изображения на некоторый угол ϕ

Ошибка округления: в математике координаты вещественные, в графике - целые

$$x' = x \cos \phi + y \sin \phi,$$

$$y' = -x \sin \phi + y \cos \phi,$$

$$x = x' \cos \phi + y' \sin \phi,$$

$$y = -x' \sin \phi + y' \cos \phi.$$

Более точное осуществление: полученные значения индексов не округлять, а выполнить интерполяцию исходного изображения

Основные выводы:

1. Дискретизация изображения – разложение изображения по регулярной структуре
 2. Квантование уровня – определение величины в целом значении единицы измерения
 3. Цифровое изображение (растр) – дискретизованное и квантованное изображение
 4. Пиксель – наименьший элемент изображения: координаты (x, y) и цвет (яркость)
 5. Каждому изображению в соответствии двумерный массив и каждому двумерному массиву - изображение
 6. К изображению применимы все матричные операции
-

Операции с битовым массивом изображения – обработка изображения (Image processing)