

Инструментарий

MatLab

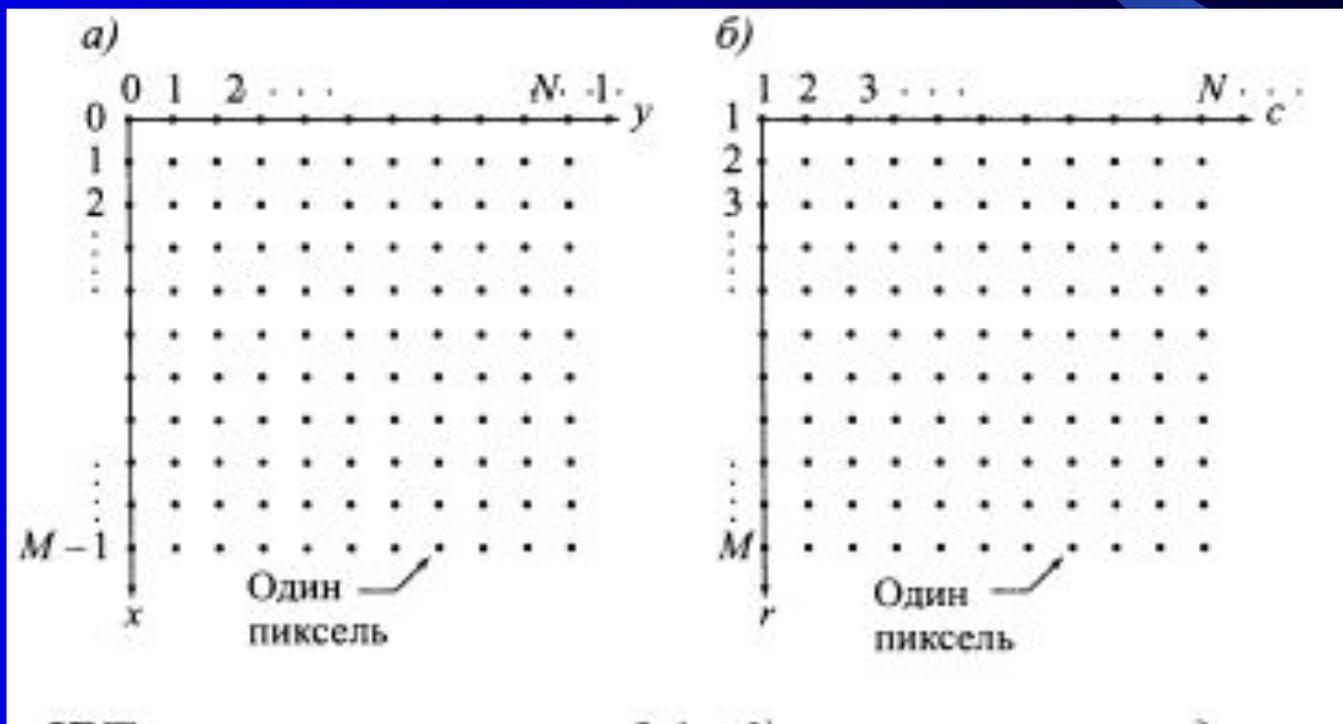
Image Processing ToolBox

Signal Processing ToolBox

Возможность MatLab

- Математические вычисления
- Разработка алгоритмов
- Моделирование и создание прототипных систем
- Анализ данных, их исследование и визуализации
- Разработка приложений

Координатное соглашение



Изображение как матрица

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N-1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1, 0) & f(M-1, 1) & \cdots & f(M-1, N-1) \end{bmatrix}.$$

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}(1, 1) & \mathbf{f}(1, 2) & \cdots & \mathbf{f}(1, N) \\ \mathbf{f}(2, 1) & \mathbf{f}(2, 2) & \cdots & \mathbf{f}(2, N) \\ \vdots & \vdots & & \vdots \\ \mathbf{f}(M, 1) & \mathbf{f}(M, 2) & \cdots & \mathbf{f}(M, N) \end{bmatrix},$$

Загрузка изображения

```
>> f=imread('dogt.png');  
??? Error using ==> imread at 315  
File "dogt.png" does not exist.  
  
>> f=imread('G:\dogt.png');  
>> |
```

Загрузка изображения

```
f=imread('G:\dogt.png')
```

```
30  0  0  0  0  0  0  0  0  0  0  23  38  44  46  38
67  41  0  0  0  0  0  0  0  0  0  0  0  27  44  48
64  61  18  0  0  0  0  0  0  0  0  0  2  44  43
64  62  50  26  0  0  0  0  0  0  0  0  0  47  49
64  64  64  52  21  0  0  0  0  0  0  0  0  0  44
61  59  61  67  59  12  0  0  0  0  0  0  0  0  8
68  64  58  61  53  53  14  0  0  0  0  0  0  0  0
76  71  59  53  58  55  50  2  0  0  0  0  0  0  0
70  80  77  65  59  61  47  35  0  0  0  0  0  0  0
61  73  80  65  55  52  38  41  50  0  0  0  0  0  0
50  61  65  58  53  47  44  58  58  41  0  0  0  0  0
52  64  67  68  70  73  73  65  44  50  59  0  0  0  0
50  64  64  73  77  77  76  67  47  39  52  35  0  0  0
44  49  58  61  65  67  68  70  68  65  58  59  26  0  0
49  50  50  55  59  62  53  65  71  73  65  62  62  26  0
43  46  47  52  53  55  50  59  62  58  49  53  52  50  15
44  47  49  53  50  52  47  46  59  55  49  50  46  49  53
47  49  47  50  53  53  47  36  38  59  59  52  44  47  46
41  38  47  47  50  52  50  53  53  59  62  61  59  58  49  39
32  35  49  49  50  52  53  59  59  53  58  61  64  59  52  41
32  38  47  50  50  50  61  62  52  49  52  52  65  64  64
49  46  44  49  49  53  53  58  50  47  55  55  59  65  80
49  47  44  46  50  55  50  50  49  43  47  50  53  58  67
```

```
>>
```

Размер изображения

size(f)

[M,N]=size(f)

```
>> size(f)
ans =
      877      1200
>> [M,N]=size(f)
M =
      877
N =
      1200
>> [M,N]=size(f);
>> |
```

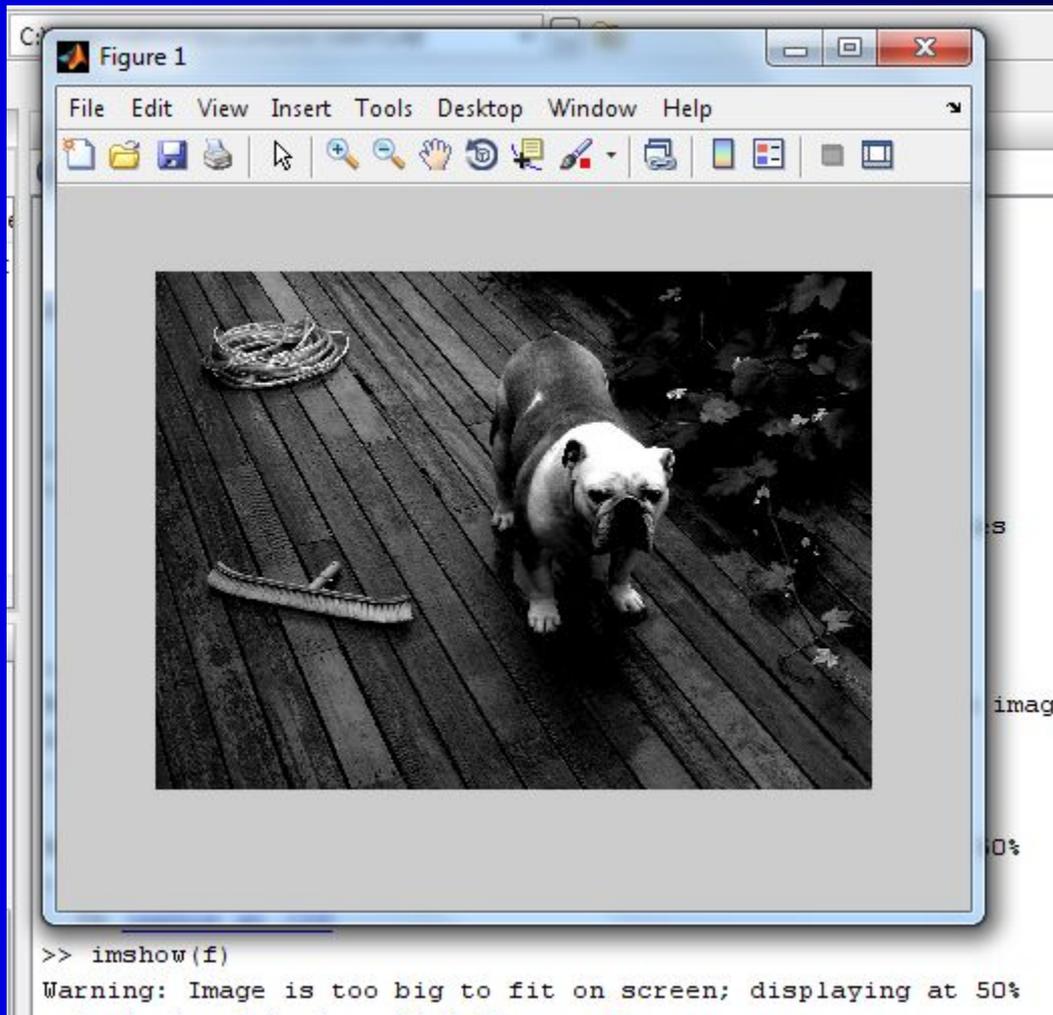
Дополнительная информация о массиве

```
>> [M,N]=size(f);
```

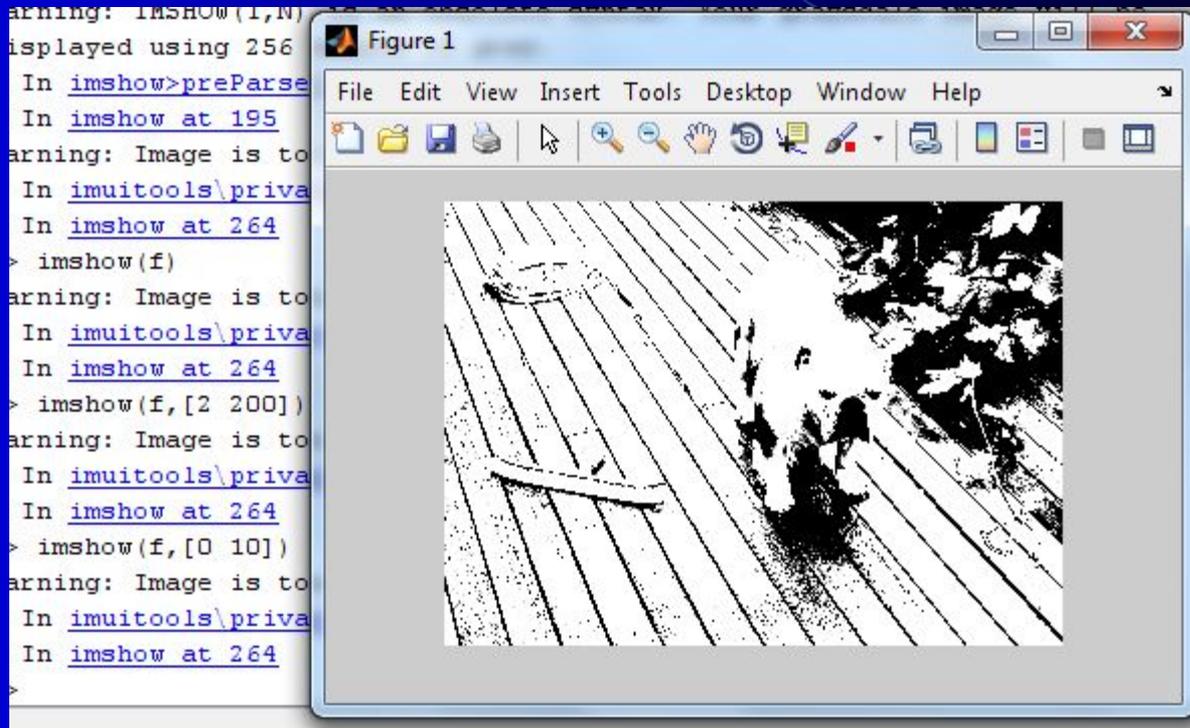
```
>> whos f
```

Name	Size	Bytes	Class	Attributes
f	877x1200	1052400	uint8	

Вывод изображения



Вывод изображения



Вывод изображения файл

```
>> imwrite(f,'g:\test1','jpg')  
>> imwrite(f,'g:\test1.tif')  
>> imwrite(f,'g:\test1.jpg','quality',q)
```

Информация о файле

```
>> imfinfo 'g:\test1.tif'

ans =

        Filename: 'g:\test1.tif'
      FileModDate: '25-окт-2011 11:25:37'
       FileSize: 877212
         Format: 'tif'
  FormatVersion: []
           Width: 1200
           Height: 877
        BitDepth: 8
        ColorType: 'grayscale'
FormatSignature: [73 73 42 0]
       ByteOrder: 'little-endian'
  NewSubFileType: 0
   BitsPerSample: 8
      Compression: 'PackBits'
PhotometricInterpretation: 'BlackIsZero'
      StripOffsets: [147x1 double]
```

Уровни сжатия

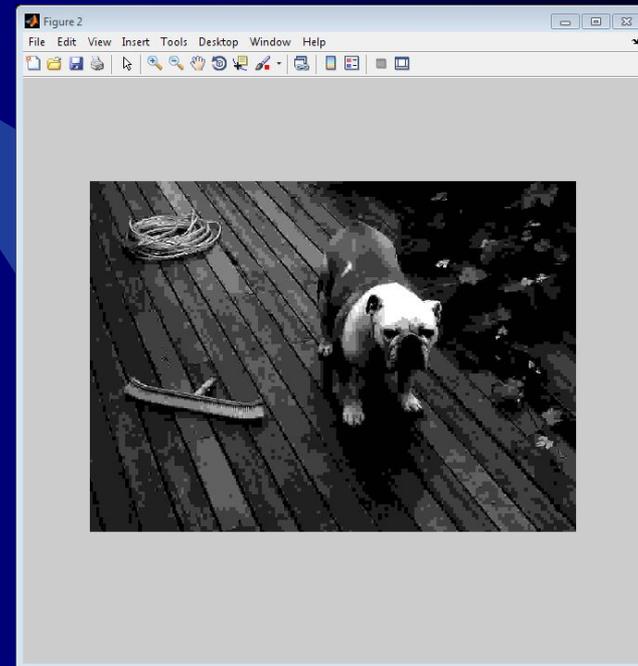
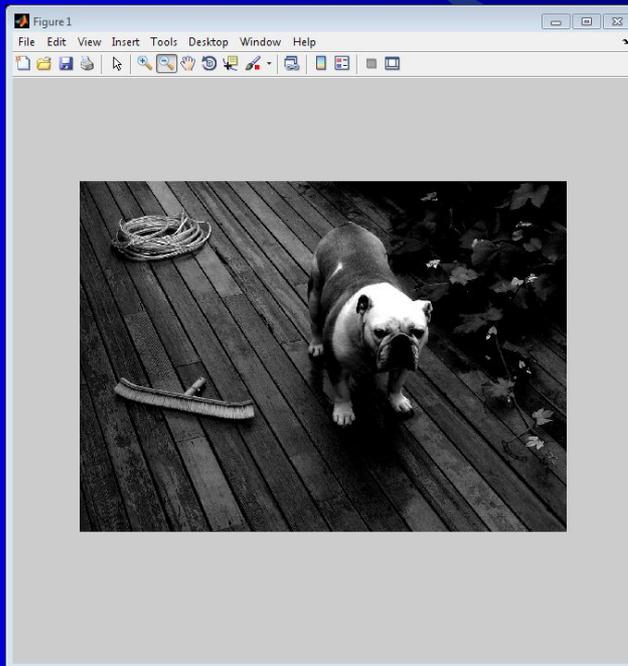
```
>> imwrite(f, 'g:\test50.jpg', 'quality', 50)
>> imwrite(f, 'g:\test25.jpg', 'quality', 25)
>> imwrite(f, 'g:\test15.jpg', 'quality', 15)
>> imwrite(f, 'g:\test10.jpg', 'quality', 10)
>> imwrite(f, 'g:\test5.jpg', 'quality', 5)
>> imwrite(f, 'g:\test0.jpg', 'quality', 0)
>> imfinfo 'g:\test0.jpg'

ans =

        Filename: 'g:\test0.jpg'
      FileModDate: '31-окт-2011 13:18:04'
       FileSize: 16472
         Format: 'jpg'
FormatVersion: ''
         Width: 1200
         Height: 877
       BitDepth: 8
       ColorType: 'grayscale'
FormatSignature: ''
NumberOfSamples: 1
   CodingMethod: 'Huffman'
CodingProcess: 'Sequential'
         Comment: {}
```

Вывод двух изображений

```
>> f=imread('G:\test50.jpg');  
>> g=imread('G:\test0.jpg');  
>> imshow(f),figure,imshow(g);
```



Вывод информации в структуру

```
>> K=imfinfo('g:\test50.jpg');  
>> imbyte=K.Width*K.Height*K.BitDepth/8  
  
imbyte =  
  
    1052400  
  
>> com=imbyte/K.FileSize  
  
com =  
  
    8.2980  
  
>> |
```

Сохранение изображения в файл из окна вывода

```
> print -fl -dtiff -r150 testp  
> |
```

Классы данных

Имя	Описание
double	Вещественные числа с плавающей запятой двойной точности в диапазоне, примерно, от -10^{308} до 10^{308} (8 байт на число).
uint8	Целые без знака в интервале $[0, 255]$ (1 байт на число).
uint16	Целые без знака в интервале $[0, 65535]$ (2 байта на число).
uint32	Целые без знака в интервале $[0, 4294967295]$ (4 байта на число).
int8	Целые со знаком в интервале $[-128, 127]$ (1 байт на число).
int16	Целые со знаком в интервале $[-32768, 32767]$ (2 байта на число).
int32	Целые со знаком в интервале $[-2147483648, 2147483647]$ (4 байта).
single	Вещественные числа с плавающей запятой обычной точности в диапазоне, примерно, от -10^{38} до 10^{38} (4 байта на число).
char	Символы (буквы и знаки) (2 байта на символ).
logical	Значения 0 или 1 (1 байт на элемент).

Типы изображений

- полутоновые изображения (изображения с градацией серого цвета);
- двоичные изображения;
- индексированные изображения;
- цветные изображения RGB.

Конвертирование класса

```
>> m=logical(f);  
>> imshow(m)
```



Функции для конвертации

Имя	Преобразует в	Допустимый класс
<code>im2uint8</code>	<code>uint8</code>	<code>logical, uint8, uint16, double</code>
<code>im2uint16</code>	<code>uint16</code>	<code>logical, uint8, uint16, double</code>
<code>mat2gray</code>	<code>double</code> (в область [0,1])	<code>double</code>
<code>im2double</code>	<code>double</code>	<code>logical, uint8, uint16, double</code>
<code>im2bw</code>	<code>logical</code>	<code>uint8, uint16, double</code>

Пример

```
>> h=uint8([255,0;128,25])
```

```
h =
```

```
255    0
```

```
128    25
```

```
>> g=im2double(h)
```

```
g =
```

```
1.0000    0
```

```
0.5020    0.0980
```

Пример

```
>> h=uint8([255,0;128,25])

h =

    255     0
    128    25

>> g=im2double(h)

g =

    1.0000         0
    0.5020    0.0980

>> gb=im2bw(g,0.6)

gb =

     1     0
     0     0
```

Индексирование массива

```
>> gb(1,1)

ans =

     1

>> gb(1,2)

ans =

     0

>> gb(1,2)=1

gb =

     1     1
     0     0

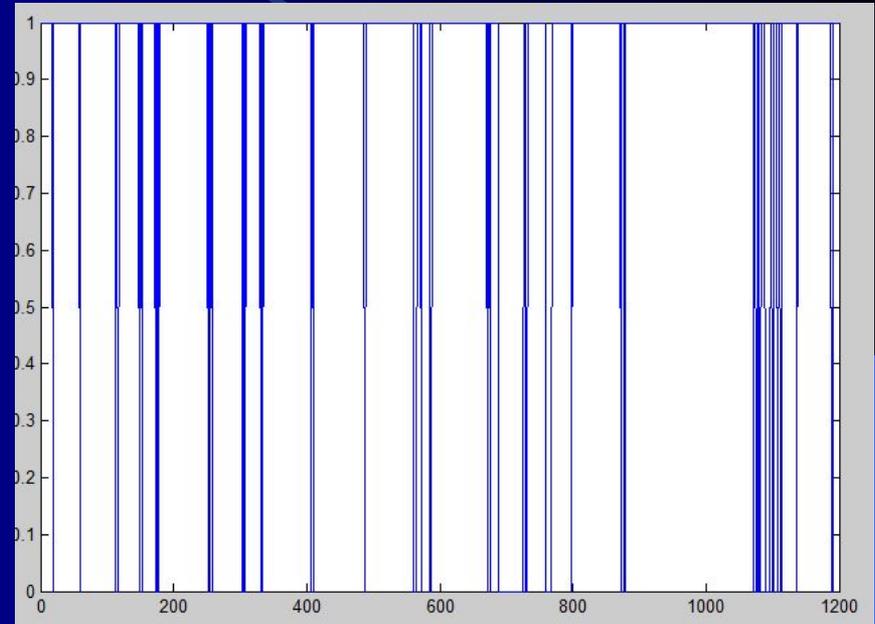
>> bg=gb.'

bg =

     1     0
     1     0
```

Построение графика

```
plot (m (512, :))
```



Размерность массива

```
>> k=size(m,1)
```

```
k =
```

```
877
```

```
>> k=size(m,2)
```

```
k =
```

```
1200
```

Стандартные массивы

- Функция `zeros(M,N)` генерирует матрицу $M \times N$ из одних нулей класса `double`.
- Функция `ones(M,N)` генерирует матрицу $M \times N$ из одних единиц класса `double`.
- Функция `true(M,N)` строит логическую матрицу $M \times N$ из единиц (истина).
- Функция `false(M,N)` строит логическую матрицу $M \times N$ из нулей (ложь).
- Функция `magic(M)` порождает «магический квадрат» $M \times M$. В этой квадратной матрице суммы чисел по любой строке, по любому столбцу и по главным диагоналям равны между собой. Магические квадраты бывают полезными при тестировании, т.к. их легко строить и их элементами служат целые числа.
- Функция `rand(M,N)` генерирует матрицу $M \times N$, элементами которой являются нормально распределенные (гауссовы) случайные величины со средним 0 и с дисперсией 1.