

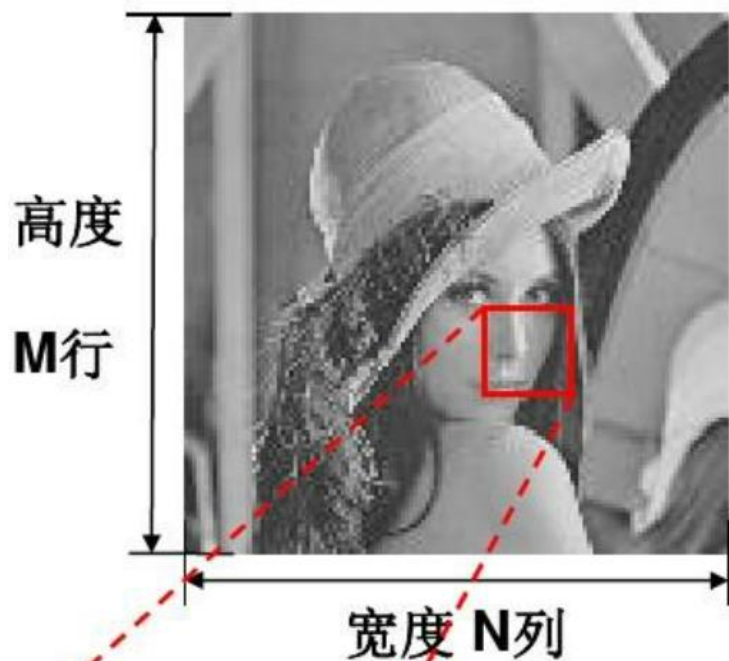
程序设计与实验II

电子与信息工程学院（微电子学院）

教师：庞志勇 高级实验师 助教：焦涵、桂海田博士研究生



数字图像表示



- 灰度图像用 $M \times N$ 的矩阵表示 (M 、 N 为正整数)
- 灰度等级数: $L=2^k$
- 存储灰度图像所需比特数
 $b=M \times N \times k$
- 彩色图像用 $M \times N \times 3$ 的矩阵表示

$$b = M \times N \times L$$

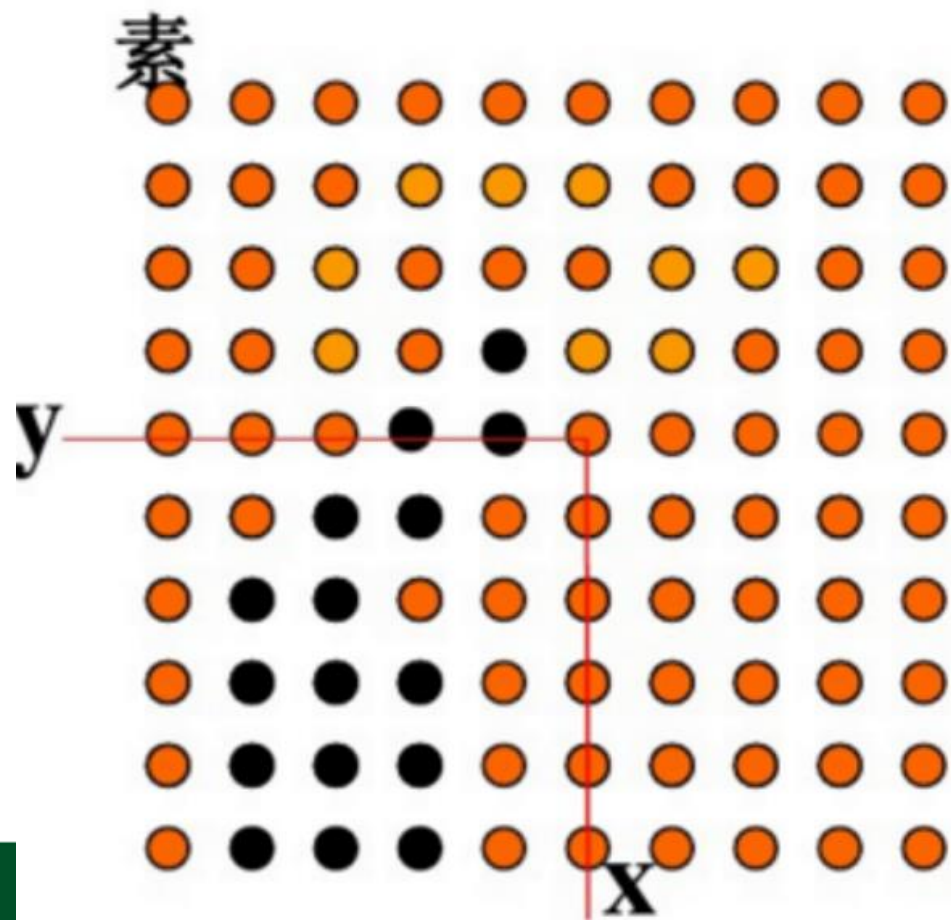
$$b = M \times N \times 2^k$$

	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
72	153	126	133	130	129	156	198	209	193	175	147	142	142	108	107	111	51	46	57
73	148	142	136	127	129	141	192	227	187	174	153	145	138	125	123	106	68	51	62
74	147	146	140	130	131	143	191	226	189	172	155	146	137	127	126	106	64	54	64
75	145	149	144	130	131	144	189	223	193	169	157	147	137	131	130	103	58	54	64
76	145	144	142	127	127	142	182	218	199	165	159	148	136	135	130	96	52	50	62
77	143	135	138	122	122	138	173	211	207	163	159	146	137	137	124	85	47	51	64
78	137	128	137	122	121	135	163	205	216	161	158	144	138	138	113	70	45	56	71
79	129	127	141	127	123	134	166	201	224	161	155	140	139	138	101	56	45	56	74
80	123	128	146	132	126	134	153	199	229	161	154	138	139	137	93	48	45	53	73
81	120	138	119	108	121	120	139	189	202	152	155	133	143	120	62	42	56	57	77
82	131	122	106	96	106	106	126	170	175	153	144	141	136	93	49	46	50	63	79
83	149	134	126	120	131	132	151	183	172	151	136	149	121	61	46	51	47	70	80
84	153	147	148	153	169	171	184	200	173	146	142	149	94	45	48	46	55	74	80
85	146	142	150	164	187	189	193	196	159	138	153	134	65	48	53	38	68	76	77
86	149	155	161	177	204	203	198	196	161	136	152	107	46	54	64	39	75	76	74
87	142	154	152	161	186	180	171	174	148	137	134	76	42	52	48	51	69	77	72
88	119	119	109	111	132	123	111	121	104	141	116	56	45	47	42	63	60	78	71
89	122	110	125	124	152	154	122	129	137	143	71	50	54	48	49	56	73	77	85
90	127	158	168	163	177	171	149	154	149	109	54	44	49	45	47	49	64	78	85



什么是像素

数字图像由二维的元素组成，每一个元素具有一个特定的位置 (x, y) 和幅值 $f(x, y)$ ，这些元素就称为像素





如何看待图像和矩阵的关系？

- 1) 矩阵是图像的一种表示。准确地说, 矩阵是二维单通道离散图像的一种表示. 图像还有其他的表示方法: 如张量, 多元函数, 随机场等. 在计算机中, 矩阵也不是图像的唯一表示方法, 还有游程, 四叉树, 多维数组等.
- 2) 数字图像处理中常见的卷积运算 (如均值滤波, 高斯滤波, Sobel 算子等) 和卷积神经网络中的卷积运算都可以转换为矩阵乘积运算。
- 3) 在图像领域, 经常把图像看成一个二维的矩阵。
- 4) 图像只是二维的数组, 而非矩阵, 或者说已经失去了矩阵表示线性变化的意义。



Opencv简介

- OpenCV 计算机视觉开源库，在OpenCV2.x版本以后，越来越多的函数实现了MatLab所具有的功能，函数名都一样（如 `imread`, `imshow`, `imwriter`等）。
- 在计算机内存中，数字图像以矩阵的形式存储和运算。在早期的OpenCV1.x版本中，图像的处理是通过`IplImage`结构来实现的。早期的OpenCV是用C语言编写，因此提供的借口也是C语言接口，其源代码完全是C的编程风格。`IplImage`结构是OpenCV矩阵运算的基本数据结构。
- 到OpenCV2.x版本，OpenCV开源库引入了面向对象编程思想，大量源代码用C++重写，`Mat`类（`Matrix`的缩写）是OpenCV用于处理图像而引入的一个封装类。从功能上讲，`Mat`类在`IplImage`结构的基础上进一步增强，并且，由于引入C++高级编程特性，`Mat`类的扩展性大大提高，查看`Mat`类的定义（`OpenCV3.1\sources\modules\core\include\opencv2\core\mat.hpp`），会发现其设计实现十分全面而具体，基本覆盖计算机视觉对于图像处理的基本要求。

Opencv Mat类

<https://blog.csdn.net/giantchen547792075/article/details/7061391>

Mat类: [OpenCV](#) C++ N 维稠密数组类

Mat类表示一个 n 维的稠密数值型的单通道或多通道数组。它可以用于存储实数或复数值的向量和矩阵、灰度或彩色图像、体素、向量场、点云、张量、直方图

```
// 创建一种新的 320 x 240 图像  
Mat img(Size(320,240),CV_8UC3);  
  
// 选择ROI(region of interest)  
Mat roi(img,Rect(10,10,100,100));
```



Opencv Mat类

<https://blog.csdn.net/giantchen547792075/article/details/7061391>

Mat:: Mat

各种Mat构造函数。

C++: Mat::Mat()

C++: Mat::Mat(int rows, int cols, int type)

C++: Mat::Mat(Size size, int type)

C++: Mat::Mat(int rows, int cols, int type, const Scalar& s)

C++: Mat::Mat(Size size, int type, const Scalar& s)

C++: Mat::Mat(const Mat& m)

C++: Mat::Mat(int rows, int cols, int type, void* data,
size_t step=AUTO_STEP)

C++: Mat::Mat(Size size, int type, void* data, size_t
step=AUTO_STEP)

C++: Mat::Mat(const Mat& m, const Range& rowRange,
const Range& colRange)

C++: Mat::Mat(const Mat& m, const Rect& roi)

.....

参数

ndims– 数组的维数.

rows – 2维数组中行行数

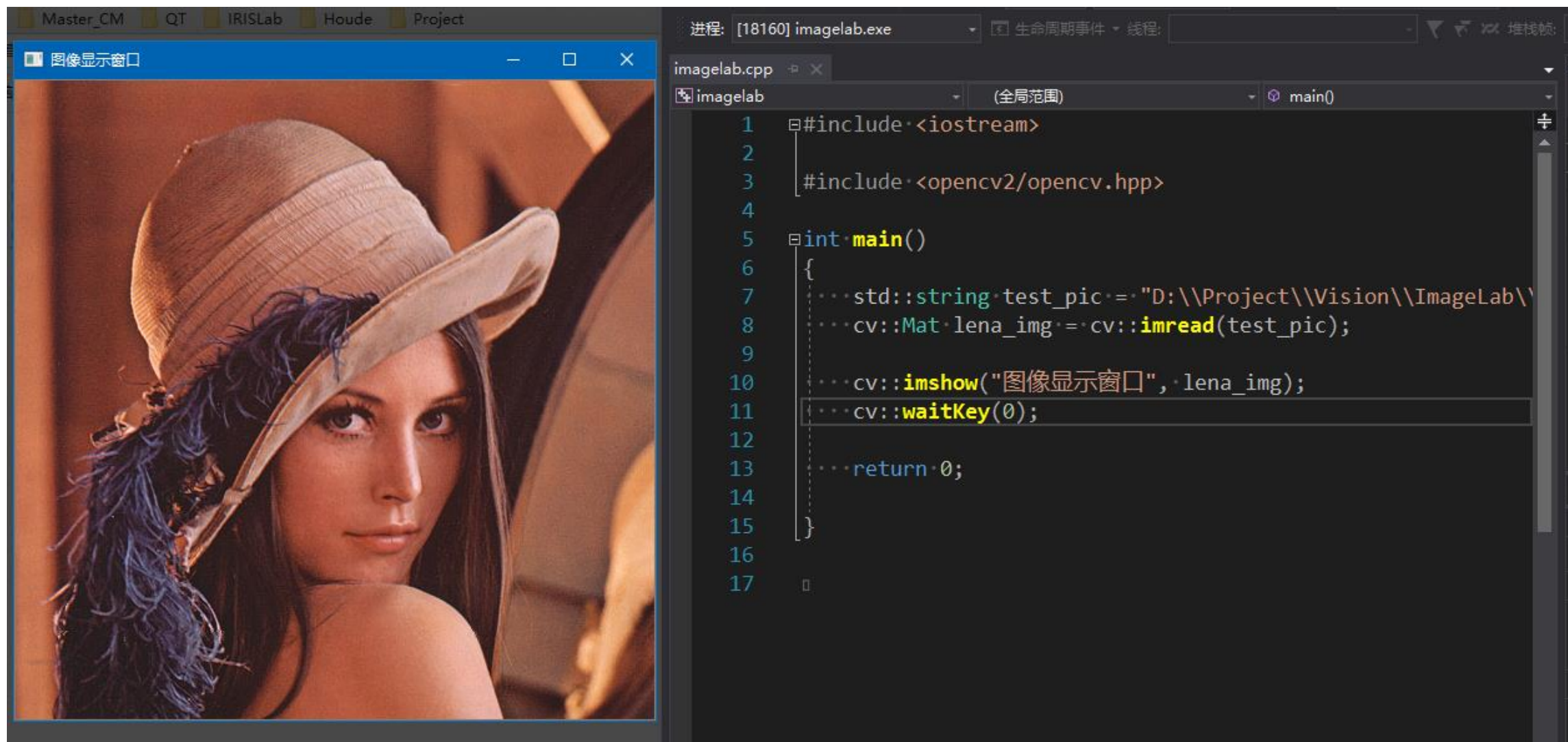
cols – Number of columns in a 2D array.

size – 2维数组的尺寸Size(cols, rows) .在Size()构造函数中行数和列数在次序上刚好反转过来了。

sizes–指定 n 维数组形状的整数数组。



OpenCV C++处理图像





```
namespace pr {
```

```
typedef std::vector<cv::Mat> Character;
```

```
enum PlateColor { BLUE, YELLOW, WHITE, GREEN, BLACK, UNKNOWN};
```

```
enum CharType {CHINESE, LETTER, LETTER_NUMS, INVALID};
```

代码分析

```
class PlateInfo {    //车牌信息类  
public:
```

```
    std::vector<std::pair<CharType, cv::Mat>> plateChars;
```

```
    std::vector<std::pair<CharType, cv::Mat>> plateCoding;
```

```
    float confidence = 0;
```

```
    PlateInfo(const cv::Mat &plateData, std::string plateName, cv::Rect plateRect, PlateColor plateType) {
```

```
        licensePlate = plateData;
```

```
        name = plateName;
```

```
        ROI = plateRect;
```

```
        Type = plateType;
```

```
    }
```

```
    PlateInfo(const cv::Mat &plateData, cv::Rect plateRect, PlateColor plateType) {
```

```
        licensePlate = plateData;
```

```
        ROI = plateRect;
```

```
        Type = plateType;
```

```
    }
```

```
    PlateInfo(const cv::Mat &plateData, cv::Rect plateRect) {
```

```
        licensePlate = plateData;
```

```
        ROI = plateRect;
```

```
    }
```

```
    PlateInfo() {
```

```
    }
```

构造函数

关联容器-----std::pair的使用

`std::pair`是二元组类模板，就是表示两种相互关联的事物，第一个叫做键，第二个叫做值，一个pair对象称为一个键值对。pair对象有两个变量，分别是first和second，调用first可以获取pair的键，调用second可以获取pair键所对应的值。我们既可以直接定义一个pair对象，也可以使用make_pair函数创建一个pair：

```
std::pair<char, int> a( 'a', 97 );  
make_pair( 'a', 97 );
```

cv::dnn::Net

OpenCV中的深度学习模块（DNN）只提供了推理功能，不涉及模型的训练，即模型训练好了，进行预测，支持多种深度学习框架，比如TensorFlow, Caffe, Torch和Darknet。