

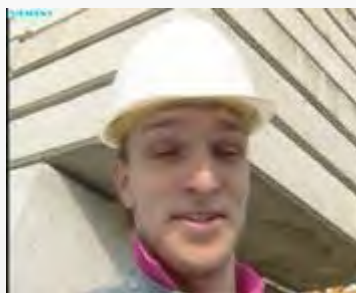


中国科学院大学

University of Chinese Academy of Sciences

# 数字视频处理

---



高绍帅

电子电气与通信工程学院

# 视频(Video)

---

- ◆ **Video**（源自于拉丁语的“我看见”）又称影片、视讯、视像、录像、动态影像，泛指将一系列的静态图像以电信号方式加以捕捉、纪录、处理、存储、传送与重现的各种技术。
- ◆ 包含**电视图像信号频谱分量**的频带内的频率。

# 视频(Video)

◆ 图像的集合

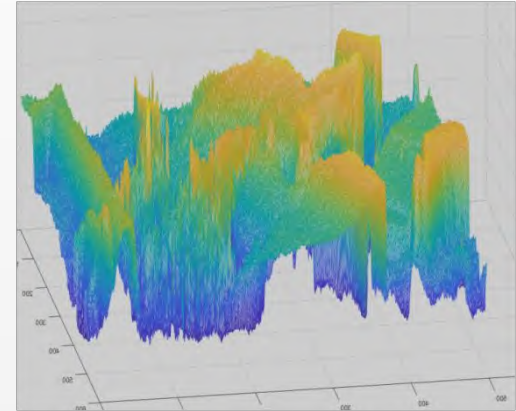
◆ 函数表示

◆  $s(x_1, x_2, x_3, t, \lambda)$

◆ 单色二维视频:  $s(x_1, x_2, t)$

◆ 矩阵表示

$$\begin{bmatrix} 46 & \dots & 180 \\ \dots & \dots & \dots \\ 76 & \dots & 186 \end{bmatrix} \begin{bmatrix} 49 & \dots & 185 \\ \dots & \dots & \dots \\ 83 & \dots & 189 \end{bmatrix} \dots \begin{bmatrix} 80 & \dots & 210 \\ \dots & \dots & \dots \\ 128 & \dots & 236 \end{bmatrix}$$



# 提纲

---

- 课程介绍
- 关于视频
- 为什么“数字”化？
- 什么是“数字视频处理（DVP）”？
- 为什么需要DVP？
- DVP应用
- 课程目标与特色
- 数字图像/视频处理Matlab初步

# 课程概要

---

◆ **课程属性：** 专业普及课      **学时/学分：** 50/3

◆ **预修课程：**

◆ 微积分、线性代数、概率论、信号与系统。

◆ **教学目的：**

◆ 掌握数字图像与视频滤波、分割、运动估计与跟踪、压缩等原理。

◆ 了解目前数字图像与视频处理领域的最新进展，为今后从事相关专业研究打下基础。

◆ **课程内容：**

◆ 图像滤波、运动估计与跟踪、图像/视频分割、视频滤波、图像/视频压缩等。

# 课程概要

## ◆ 答疑 (Office Hour)

- ◆ 时间：周四上午，3-4节
- ◆ 地点：学园2-369室
- ◆ Email: ssgao@ucas.ac.cn

## ◆ 教材

- ◆ A. Murat Tekalp (曹铁勇等译)，《数字视频处理》第2版，机械工业出版社, 2017。



## ◆ 参考书目

- ◆ 1、Alan Bovik, The Essential Guide to Video Processing. Academic Press, 2009.
- ◆ 2、Yao Wang, Jorn Ostermann, Ya-Qin Zhang (侯正信，杨喜，王文全译)，《视频处理与通信》，电子工业出版社，北京，2003。

# 考核标准

---

- ◆ 课后作业：20%
- ◆ 大作业（编程，报告）：30%
- ◆ 开卷笔试：50%

# 课程内容

---

## ◆ 第一章 多维信号与系统

- ◆ 多维信号，多维变换，多维系统，多维采样理论，采样格式转换。

## ◆ 第二章 数字图像和视频

- ◆ 人类视觉系统和色彩，数字视频，3D视频，数字视频应用，图像和视频质量评估。

## ◆ 第三章 图像滤波

- ◆ 图像平滑，重采样和多分辨率表示，梯度估计、边缘和特征检测，图像增强，图像去噪，图像复原。

## ◆ 第四章 运动估计

- ◆ 图像的形成，运动模型，2D表观运动估计，差分法，匹配法，非线性优化法，变换域方法，3D运动估计和结构估计。



# 课程内容

---

## ◆ 第五章 视频分割与跟踪

- ◆ 图像分割，变化检测，运动分割，运动跟踪，抠图和视频抠像，性能评估。

## ◆ 第六章 视频滤波

- ◆ 空-时滤波理论，视频格式转换，多帧联合去噪，多帧联合复原，超分辨率重建。

## ◆ 第七章 图像压缩

- ◆ 图像压缩基础，离散余弦变换，JPEG，小波变换，JPEG2000。

## ◆ 第八章 视频压缩

- ◆ 视频压缩方法，早期视频压缩标准，H.264/AVC，HEVC。

# 关于“视频”

---

## ◆ 视频：运动图片（Motion Pictures）

- ◆ 电影、电视

- ◆ 网络广播：新闻、运动比赛转播、娱乐和教育内容等。

- ◆ 个人视频信息：使用摄像机、网络摄像头、移动相机等等。

- ◆ 闭路电视监控视频

## ◆ 包含大量而丰富的视觉数据信息。

- ◆ 一“图”胜千言  一“视频”胜千句？

# 关于“视频”

---

## ◆ 图像和视频的区别

- ◆ 图像：形状、颜色、纹理
- ◆ 视频：形状、颜色、纹理、**运动**（增加了**时间**维度）

## ◆ 二维与三维的区别

- ◆ 加入**深度**信息
- ◆ 一般为左右两幅视图（两个摄像机同时获取）

# 电影

老式的电影放映机



模拟(光学)到数字



数字电影放映机

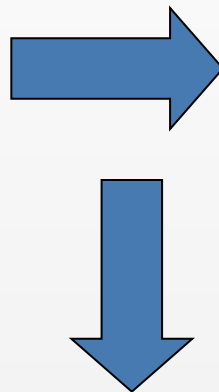


# 电视

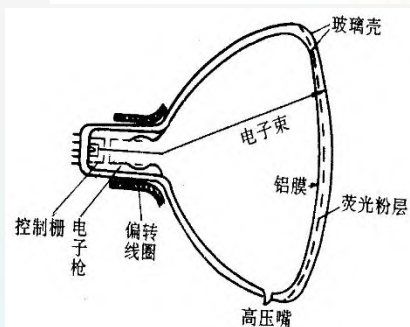


黑白到彩色

模拟到数字

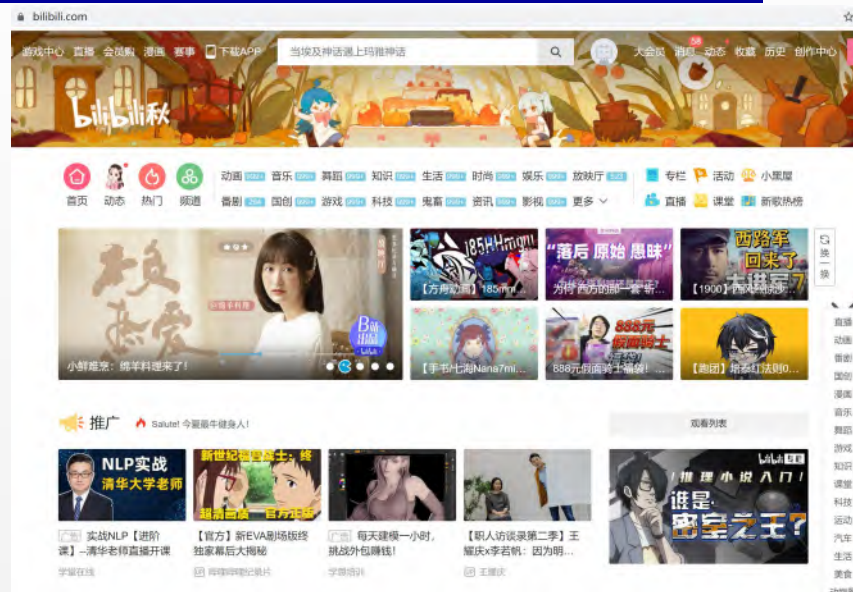
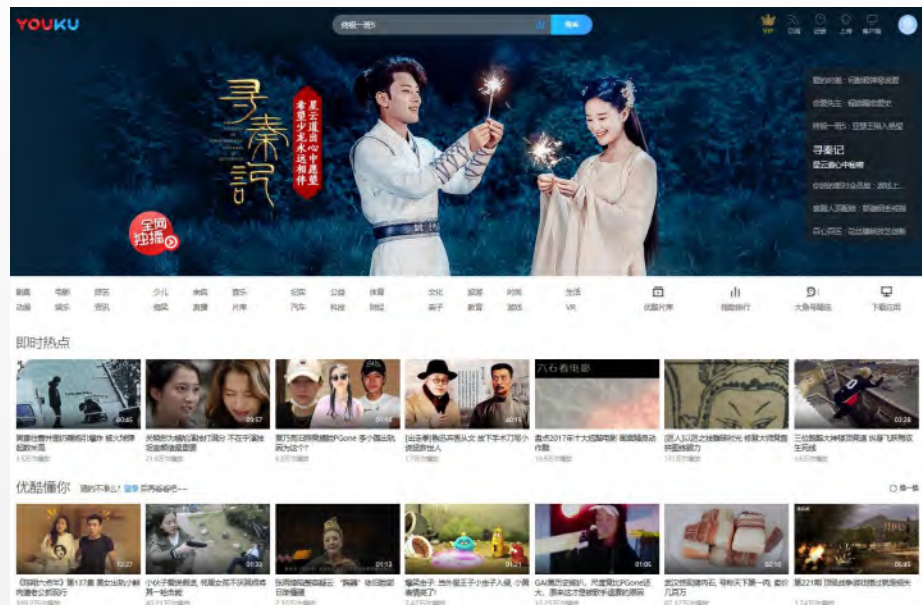


电视  
机显  
像管

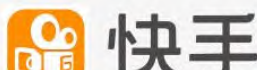




# 网络视频



超酷炫短视频社区



# 网络视频

---



# 网络视频





# 三维视频

---



# 为什么“数字化”？

---

## ◆ 精确性

- ◆ 完美复制没有质量退化
- ◆ 处理结果的完美复制

## ◆ 方便强大的计算机辅助处理

- ◆ 可以通过硬件或软件执行相当复杂的处理
- ◆ 甚至小朋友也可以做到!

## ◆ 容易存储和传输

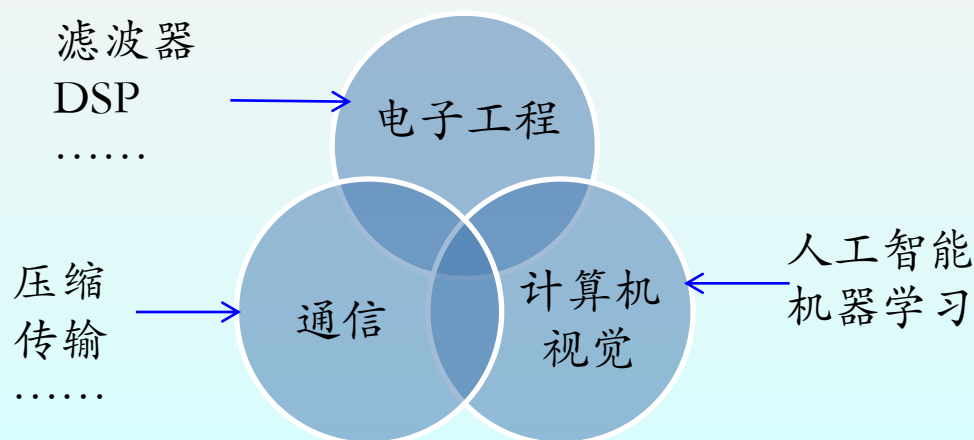
- ◆ 1 CD可以存储1视频或数百个家庭照片!
- ◆ 高质量照片可通过网络在几秒钟内进行无纸化传输

# 什么是“数字视频处理(DVP)” ?

## ◆ DVP用于处理下列计算任务

- ◆ 提取有用的视频信息
- ◆ 恢复原始视频信息
- ◆ 视频数据压缩
- ◆ 对非结构化视频进行表示(**结构化**)
- ◆ 评估视频质量(视频质量评估), 等等

**DVP:**  
可认为是一门**交叉**  
学科。



# 为什么需要DVP？



各种类型的信道



日益增长的个人视频信息



存储空间爆炸式增长

个人视频信息大爆炸



处理时间太长



难以获取目标视频



试图预览内容？

# 为什么需要DVP ?

---

## ◆ 便于视频存储和传输:

- ◆ 数码摄像机、VCD、DVD、蓝光盘等视频存储
- ◆ 发送视频给他人
- ◆ 视频压缩,运动估计

## ◆ 便于显示或打印:

- ◆ 调整图像大小
- ◆ 视频超分辨率

## ◆ 增强和恢复图片:

- ◆ 去除噪声和传输或压缩过程中造成的损伤
- ◆ 视频增强、视频去噪、减少压缩损伤

## ◆ 从视频中提取有用的信息:

- ◆ 提取或跟踪对象包括脸,文本,等等
- ◆ 对象分割和检测、识别、跟踪、行为检测和理解、视频分割和表示、索引、注释、检索

# DVP涉及的两大领域

---

## ◆ 计算机视觉 (CV: Computer Vision)

- ◆ 通过处理和理解数字图像与视频以模仿人类视觉
- ◆ 可分为
  - ◆ 高等级视觉 (High-level Vision) : 人工智能的一部分; 机器学习与模式识别; 目标识别
  - ◆ 低等级视觉 (Low-level Vision) : 与DVP中讨论的技术重合; 包括边缘检测、图像增强与恢复, 运动估计, 分割与跟踪等等。

## ◆ 视频压缩

- ◆ 去除视频信号中包含的冗余, 降低码率
- ◆ 视频存储
- ◆ 视频通信
- ◆ 国际编码标准: H.26x系列与MPEG系列, 以及JVT系列

# DVP中的具体研究领域

---

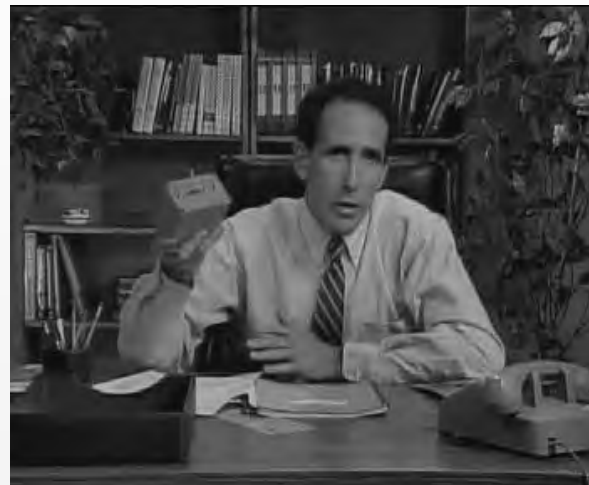
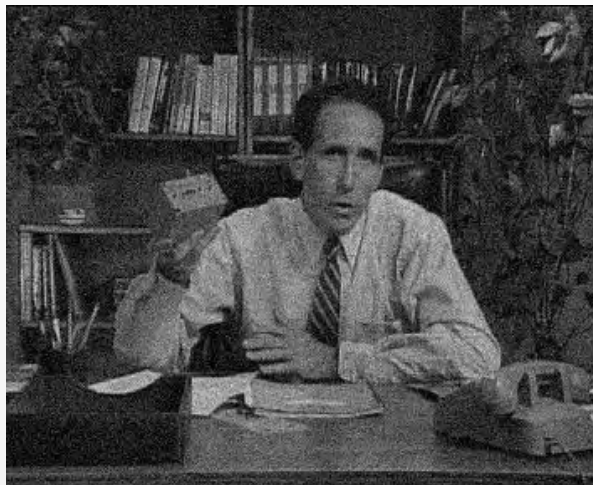
- ◆ 视频压缩
- ◆ 视频通信
- ◆ 视频增强与恢复
- ◆ 视频分割与跟踪
- ◆ 运动估计
- ◆ 立体视频处理
- ◆ 视频内容分析
- ◆ 视频索引和检索
- ◆ 对象检测与识别

.....



# DVP应用举例

去噪



去雾





# DVP应用举例

---



压缩视频去块效应

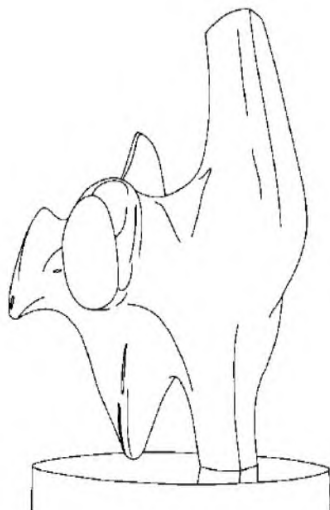
# DVP应用举例

---



图像修复  
(Image In-Painting)

# DVP应用举例

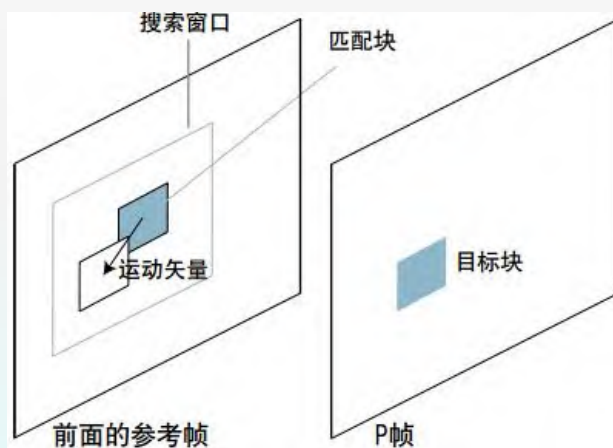


边缘检测

角点检测

# DVP应用举例

## 二维运动估计





# DVP应用举例

超分辨率



(a) LR 图像



(b) MAP 算法



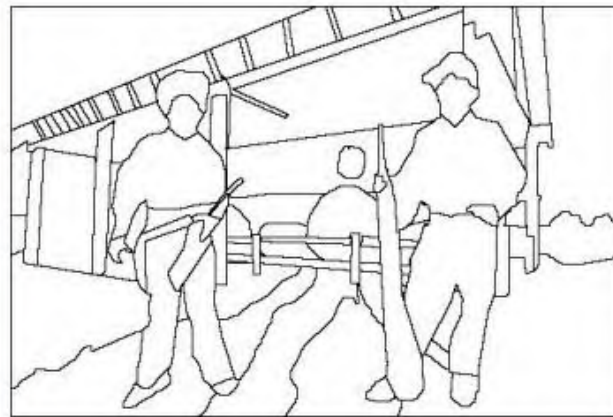
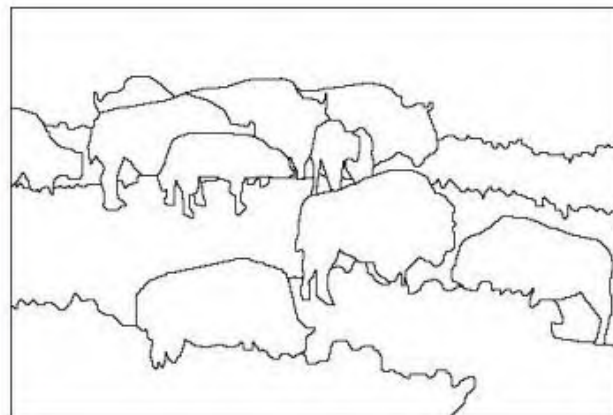
(c) POCS 算法



(d) IBP 算法

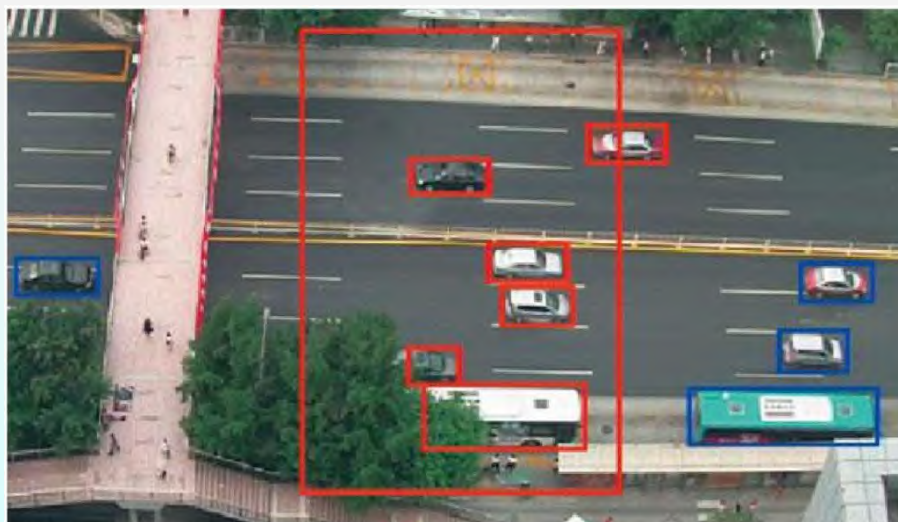
# DVP应用举例

视频分割



# DVP应用举例

视频跟踪



# 课程目标与特色

---

- ◆ 由问题导入，由浅入深，力求浅入深出
  - ◆ 微积分、线性代数、概率论、信号与系统、最优化方法等知识的具体演练。
- ◆ PowerPoint展示为主，板书为辅
- ◆ 经典算法的程序执行演示：获得直观感受
- ◆ 理论与实践结合
  - ◆ 大作业：亲自体验实现算法的乐趣



# 数字图像/视频处理Matlab初步

## ◆ 基本数据类型：

<code>A = [1 2; 3 4];</code>	<code>% Creates a 2x2 matrix</code>
<code>B = [1,2; 3,4];</code>	
<code>N = 5</code>	<code>% A scalar</code>
<code>v = [1 0 0]</code>	<code>% A row vector</code>
<code>v = [1; 2; 3]</code>	<code>% A column vector</code>
<code>v = v'</code>	<code>% Transpose a vector</code>
<code>v = 1:5:3</code>	<code>% A vector filled in a specified range:</code>
<code>v = pi*[-4:4]/4</code>	<code>% [start:stepsize:end]</code>
	<code>% (brackets are optional)</code>
<code>v = []</code>	<code>% Empty vector</code>

## ◆ 特殊矩阵：

<code>m = zeros(2, 3)</code>	<code>% Creates a 2x3 matrix of zeros</code>
<code>v = ones(1, 3)</code>	<code>% Creates a 1x3 matrix (row vector) of ones</code>
<code>m = eye(3)</code>	<code>% Identity matrix (3x3)</code>
<code>v = rand(3, 1)</code>	<code>% Randomly filled 3x1 matrix (column vector)</code>
<code>m = zeros(3)</code>	<code>% Creates a 3x3 matrix (!) of zeros</code>

# 数字图像/视频处理Matlab初步

## ◆ 矩阵运算：

<code>a = [1 2 3 4]';</code>	<code>% A column vector</code>
<code>2 * a</code>	<code>% Scalar multiplication</code>
<code>a / 4</code>	<code>% Scalar division</code>
<code>b = [5 6 7 8]';</code>	<code>% Another column vector</code>
<code>a + b</code>	<code>% Vector addition</code>
<code>a - b</code>	<code>% Vector subtraction</code>
<code>a .^ 2</code>	<code>% Element-wise squaring (note the ".")</code>
<code>a .* b</code>	<code>% Element-wise multiplication (note the ".")</code>
<code>a ./ b</code>	<code>% Element-wise division (note the ".")</code>
<code>log([1 2 3 4])</code>	<code>% Element-wise logarithm</code>
<code>round([1.5 2; 2.2 3.1])</code>	<code>% Element-wise rounding to nearest integer</code>
<code>a = rand(3,2)</code>	<code>% A 3x2 matrix</code>
<code>b = rand(2,4)</code>	<code>% A 2x4 matrix</code>
<code>c = a * b</code>	<code>% Matrix product results in a 3x4 matrix</code>
<code>a = [1 2; 3 4; 5 6];</code>	<code>% A 3x2 matrix</code>
<code>b = [5 6 7];</code>	<code>% A 1x3 row vector</code>
<code>b * a</code>	<code>% Vector-matrix</code>
<code>a = [1 3 2; 6 5 4; 7 8 9];</code>	<code>% A 3x3 matrix</code>
<code>inv(a)</code>	<code>% Matrix inverse of a</code>
<code>eig(a)</code>	<code>% Vector of eigenvalues of a</code>
<code>[V, D] = eig(a)</code>	<code>% D matrix with eigenvalues on diagonal;</code>
<code>[U, S, V] = svd(a)</code>	<code>% Singular value decomposition of a.</code>

# 数字图像/视频处理Matlab初步

## ◆ 画图：

```
x = [0 1 2 3 4];
plot(x);
plot(x, 2*x);
axis([0 8 0 8]);
figure;
x = pi*[-24:24]/24;
plot(x, sin(x));
xlabel('radians');
ylabel('sin value');
title('dummy');
figure;
subplot(1, 2, 1);
plot(x, sin(x));
axis square;
subplot(1, 2, 2);
plot(x, 2*cos(x));
axis square;
figure;
plot(x, sin(x));
hold on;
plot(x, 2*cos(x), '--');
legend('sin', 'cos');
hold off;
figure;
m = rand(64,64);
imagesc(m);
colormap gray;
axis image;
axis off;
```

% Basic plotting  
% Plot x versus its index values  
% Plot 2\*x versus x  
% Adjust visible rectangle  
% Open new figure  
  
% Assign label for x-axis  
% Assign label for y-axis  
% Assign plot title  
  
% Multiple functions in separate graphs  
  
% Make visible area square  
  
% Multiple functions in single graph  
% '--' chooses different line pattern  
% Assigns names to each plot  
% Stop putting multiple figures in current % graph  
% Matrices vs. images  
  
% Plot matrix as image  
% Choose gray level colormap  
% Show pixel coordinates as axes  
% Remove axes

# 数字图像/视频处理Matlab初步

## ◆ 图像读取、显示、写入等操作

```
I = imread('pollen.tif');           % Read a TIF image
figure
imagesc(I)                          % Display it as gray level image
colormap gray;                      % Turn on color bar on the side
colorbar                            % Display pixel values interactively
imshow(I, 'truecolor')              % Display at resolution of one screen pixel per image pixel

I2 = imresize(I, 0.5, 'bil');        % Resize to 50% using bilinear interpolation
I3 = imrotate(I2, 45, 'bil', 'crop'); % Rotate 45 degrees and crop to original size

I3 = double(I3);                    % Convert from uint8 to double, to allow math operations
imagesc(I3.^2)                      % Display squared image (pixel-wise)
imagesc(log(I3))                    % Display log of image (pixel-wise)
I3 = uint8(I3);                     % Convert back to uint8 for writing
imwrite(I3, 'test.png')              % Save image as PNG
```

# 数字图像/视频处理Matlab初步

## ◆ 视频读取、显示、写入等操作

```
readerobj = VideoReader('xylophone.mp4', 'tag', 'myreader1');

% Read in all video frames.
vidFrames = read(readerobj);

% Get the number of frames.
numFrames = get(readerobj, 'NumberOfFrames');

% Create a MATLAB movie struct from the video frames.
for k = 1 : numFrames
    mov(k).cdata = vidFrames(:,:,k);
    mov(k).colormap = [];
end

% Create a figure
hf = figure;

% Resize figure based on the video's width and height
set(hf, 'position', [150 150 readerobj.Width readerobj.Height])

% Playback movie once at the video's frame rate
movie(hf, mov, 1, readerobj.FrameRate);
```