

## Chapter\_1\_:Classification

### 1.1 Gaussian-Mixture-Models

#### 1.add\_sample\_class\_gmm

功能： 把一个训练样本添加到一个高斯混合模型的训练数据上。

#### 2.classify\_class\_gmm

功能： 通过一个高斯混合模型来计算一个特征向量的类。

#### 3. clear\_all\_class\_gmm

功能： 清除所有高斯混合模型。

#### 4. clear\_class\_gmm

功能： 清除一个高斯混合模型。

#### 5. clear\_samples\_class\_gmm

功能：清除一个高斯混合模型的训练数据。

## 6. create\_class\_gmm

功能：为分类创建一个高斯混合模型。

## 7.evaluate\_class\_gmm

功能：通过一个高斯混合模型评价一个特征向量。

## 8. get\_params\_class\_gmm

功能：返回一个高斯混合模型的参数。

## 9. get\_prep\_info\_class\_gmm

功能：计算一个高斯混合模型的预处理特征向量的信息内容。

## 10. get\_sample\_class\_gmm

功能：从一个高斯混合模型的训练数据返回训练样本。

#### 11. `get_sample_num_class_gmm`

功能：返回存储在一个高斯混合模型的训练数据中的训练样本的数量。

#### 12. `read_class_gmm`

功能：从一个文件中读取一个高斯混合模型。

#### 13. `read_samples_class_gmm`

功能：从一个文件中读取一个高斯混合模型的训练数据。

#### 14. `train_class_gmm`

功能：训练一个高斯混合模型。

#### 15. `write_class_gmm`

功能：向文件中写入一个高斯混合模型。

#### 16. `write_samples_class_gmm`

功能：向文件中写入一个高斯混合模型的训练数据。

## 1.2 Hyperboxes

### 1. clear\_sampset

功能：释放一个数据集的内存。

### 2. close\_all\_class\_box

功能：清除所有分类器。

### 3. close\_class\_box

功能：清除分类器。

### 4. create\_class\_box

功能：创建一个新的分类器。

#### 5. `descript_class_box`

功能：分类器的描述。

#### 6. `enquire_class_box`

功能：为一组属性分类。

#### 7. `enquire_reject_class_box`

功能：为一组带抑制类的属性分类。

#### 8. `get_class_box_param`

功能：获取关于现在参数的信息。

#### 9. `learn_class_box`

功能：训练分类器。

#### 10. `learn_sampset_box`

功能：用数据组训练分类器。

#### 11. read\_class\_box

功能：从一个文件中读取分类器。

#### 12.read\_sampset

功能：从一个文件中读取一个训练数据组。

#### 13. set\_class\_box\_param

功能：为分类器设计系统参数。

#### 14. test\_sampset\_box

功能：为一组数组分类。

#### 15. write\_class\_box

功能：在一个文件中保存分类器。

## 1.3 Neural-Nets

### 1. add\_sample\_class\_mlp

功能：把一个训练样本添加到一个多层感知器的训练数据中。

### 2. classify\_class\_mlp

功能：通过一个多层感知器计算一个特征向量的类。

### 3. clear\_all\_class\_mlp

功能：清除所有多层感知器。

### 4. clear\_class\_mlp

功能：清除一个多层感知器。

### 5. clear\_samples\_class\_mlp

功能：清除一个多层感知器的训练数据。

## 6. create\_class\_mlp

功能：为分类或者回归创建一个多层感知器。

## 7. evaluate\_class\_mlp

功能：通过一个多层感知器计算一个特征向量的评估。

## 8. get\_params\_class\_mlp

功能：返回一个多层感知器的参数。

## 9. get\_prep\_info\_class\_mlp

功能：计算一个多层感知器的预处理特征向量的信息内容。

## 10. get\_sample\_class\_mlp

功能：从一个多层感知器的训练数据返回一个训练样本。



#### 11. get\_sample\_num\_class\_mlp

功能：返回存储在一个多层感知器的训练数据中的训练样本的数量。

#### 12. read\_class\_mlp

功能：从一个文件中读取一个多层感知器。

#### 13. read\_samples\_class\_mlp

功能：从一个文件中读取一个多层感知器的训练数据。

#### 14. train\_class\_mlp、

功能：训练一个多层感知器。

#### 15. write\_class\_mlp

功能：向一个文件中写入一个多层感知器。

## 16. write\_samples\_class\_mlp

功能： 向一个文件中写入一个多层感知器的训练数据。

## 1.4 Support-Vector-Machines

### 1. add\_sample\_class\_svm

功能： 把一个训练样本添加到一个支持向量机的训练数据上。

### 2. classify\_class\_svm

功能： 通过一个支持向量机为一个特征向量分类。

### 3. clear\_all\_class\_svm

功能： 清除所有支持向量机。

### 4. clear\_class\_svm

功能：清除一个支持向量机。

#### 5. `clear_samples_class_svm`

功能：清除一个支持向量机的训练数据。

#### 6. `create_class_svm`

功能：为模式分类创建一个支持向量机。

#### 7. `get_params_class_svm`

功能：返回一个支持向量机的参数。

#### 8. `get_prep_info_class_svm`

功能：计算一个支持向量机的预处理特征向量的信息内容。

#### 9. `get_sample_class_svm`

功能：从一个支持向量机的训练数据返回一个训练样本。

#### 10. `get_sample_num_class_svm`

功能：返回存储在一个支持向量机训练数据中的训练样本的数量。

#### 11. `get_support_vector_class_svm`

功能：从一个训练过的支持向量机返回一个支持向量的索引。

#### 12. `get_support_vector_num_class_svm`

功能：返回一个支持向量机的支持向量的数量。

#### 13. `read_class_svm`

功能：从一个文件中读取一个支持向量机。

#### 14. `read_samples_class_svm`

功能：从一个文件中读取一个支持向量机的训练数据。

#### 15. `reduce_class_svm`

功能：为了更快分类，用一个降低的支持向量机近似一个训练过的支持向量机。

#### 16. train\_class\_svm

功能：训练一个支持向量机。

#### 17. write\_class\_svm

功能：向一个文件中写入一个支持向量机。

#### 18.write\_samples\_class\_svm

功能：向一个文件中写入一个支持向量机的训练数据。

### Chapter\_2 Control

#### 1.assign

功能：为一个控制变量分配一个新值。

#### 2.break

功能：终止循环执行。

### 3. comment

功能：向程序添加一行注释。

### 4. continue

功能：跳过现在的循环执行。

### 5. else

功能：条件语句的替换。

### 6. elseif

功能：可选择的条件语句。

### 7. endfor

功能：for 循环的终止。

## 8. endif

功能：if 命令的终止。

## 9. endwhile

功能：while 循环的终止。

## 10. exit

功能：终止 HDevelop。

## 11. for

功能：执行一定数量的主体。

## 12. if

功能：条件语句。

## 13. ifelse

功能：有选择的条件语句。

#### 14. insert

功能：向一个元组分配一个量。

#### 15.repeat

功能：repeat..until 循环的开始。

#### 16. return

功能：终止程序调用。

#### 17. stop

功能：停止程序执行。

#### 18.until

功能：继续执行主体，只要条件是不真实的。



## 19. while

功能：继续执行主体，只要条件是真实的。

## Chapter\_3 :Develop

### 1. dev\_clear\_obj

功能：从 HALCON 数据库中删除一个图标。

### 2. dev\_clear\_window

功能：清除活动图形窗口。

### 3. dev\_close\_inspect\_ctrl

功能：关闭一个控制变量的监视窗口。

### 4. dev\_close\_window

功能：关闭活动图形窗口。

### 5. dev\_display

功能：在现有图形窗口中显示图像目标。

#### 6. dev\_error\_var

功能：定义或者不定义一个错误变量。

#### 7. dev\_get\_preferences

功能：通过设计查询 HDevelop 的参数选择。

#### 8. dev\_inspect\_ctrl

功能：打开一个窗口来检查一个控制变量。

#### 9. dev\_map\_par

功能：打开一个对话框来指定显示参数。

#### 10. dev\_map\_prog

功能：使 HDevelop\_的主窗口可视化。

#### 11. dev\_map\_var

功能：在屏幕上绘制可视化窗口。

#### 12. dev\_open\_window

功能：打开一个图形窗口。

#### 13. dev\_set\_check

功能：指定错误处理。

#### 14. dev\_set\_color

功能：设置一个或更多输出颜色。

#### 15. dev\_set\_colored

功能：设置混合输出颜色。

#### 16. dev\_set\_draw

功能：定义区域填充模式。

#### 17. dev\_set\_line\_width

功能：定义区域轮廓输出的线宽。

#### 18. dev\_set\_lut

功能：设置查询表\_(lut).

#### 19. dev\_set\_paint

功能：定义灰度值输出模式。

#### 20. dev\_set\_part

功能：修改显示图像部分。

#### 21. dev\_set\_preferences

功能：通过设计设置 HDevelop 的参数选择。

## 22. dev\_set\_shape

功能：定义区域输出形状。

## 23. dev\_set\_window

功能：激活一个图形窗口。

## 24. dev\_set\_window\_extents

功能：改变一个图形窗口的位置和大小。

## 25. dev\_unmap\_par

功能：为图形参数隐藏窗口。

## 26. dev\_unmap\_prog

功能：隐藏主窗口。

## 27. dev\_unmap\_var

功能：隐藏变量窗口。

## 28. dev\_update\_pc

功能：在程序执行中指定 PC 的行为。

## 29. dev\_update\_time

功能：为操作符打开或关闭切换时间测量。

## 30. dev\_update\_var

功能：在程序执行中指定活动窗口的行为。

## 31. dev\_update\_window

功能：在程序执行中指定输出行为。

## Chapter\_4\_:File

### 4.1 Images

### 1. read\_image

功能：读取有不同文件格式的图像。

### 2. read\_sequence

功能：读取图像。

### 3. write\_image

功能：用图形格式写图像。

## 4.2 Misc

### 1. delete\_file

功能：删除一个文件。

### 2. file\_exists

功能：检查文件是否存在。

### 3. list\_files

功能：列出目录中的所有文件。

### 4. read\_world\_file

功能：从一个 ARC/INFO 世界文件中读取地理编码。

## 4.3 Region

### 1. read\_region

功能：读取二值图像或者 HALCON 区域。

### 2. write\_region

功能：在文件中写入地域。



## 4.4 Text

### 1. close\_all\_files

功能：关闭所有打开的文件。

### 2.close\_file

功能：关闭一个文本文件。

### 3. fnew\_line

功能：创建一个换行符。

### 4. fread\_char

功能：从一个文本文件中读取一个字符。

### 5. fread\_line

功能：从一个文本文件中读取一行。

## 6. fread\_string

功能：从一个文本文件中读取字符串。

## 7. fwrite\_string

功能：向一个文本文件中写入值。

## 8. open\_file

功能：打开文本文件。

## 4.5 Tuple

### 1. read\_tuple

功能：从一个文件中读取一个数组。

## 2. write\_tuple

功能：向一个文件中写入一个数组。

## 4.6\_XLD

### 1. read\_contour\_xld\_arc\_info

功能：从用 ARC/INFO 生成格式表示的文件读取 XLD 轮廓。

### 2. read\_contour\_xld\_dxf

功能：从一个 DXF 文件中读取\_XLD 轮廓。

### 3. read\_polygon\_xld\_arc\_info

功能：从用 ARC/INFO 生成格式表示的文件读取 XLD 多边形。

### 4. read\_polygon\_xld\_dxf

功能：从一个 DXF 文件中读取\_XLD 多边形。

#### 5. write\_contour\_xld\_arc\_info

功能：向用 ARC/INFO 生成格式表示的文件写入 XLD 轮廓。

#### 6. write\_contour\_xld\_dxf

功能：向一个 DXF 格式的文件中写入\_XLD 轮廓。

#### 7. write\_polygon\_xld\_arc\_info

功能：向用 ARC/INFO 生成格式表示的文件写入 XLD 多边形。

#### 8. write\_polygon\_xld\_dxf

功能：向一个 DXF 格式的文件中写入\_XLD 多边形。

### Chapter\_5:Filter

#### 5.1 Arithmetic

### 1. abs\_image

功能：计算一个图像的绝对值（模数）。

### 2. add\_image

功能：使两个图像相加。

### 3. div\_image

功能：使两个图像相除。

### 4. invert\_image

功能：使一个图像反像。

### 5. max\_image

功能：按像素计算两个图像的最大值。

### 6. min\_image

功能：按像素计算两个图像的最大小值。

## 7. mult\_image

功能：使两个图像相乘。

## 8. scale\_image

功能：为一个图像的灰度值分级。

## 9. sqrt\_image

功能：计算一个图像的平方根。

## 10. sub\_image

功能：使两个图像相减。

## 5.2 Bit

### 1. bit\_and

功能：输入图像的所有像素的逐位与。

### 2. bit\_lshift

功能：图像的所有像素的左移。

### 3. bit\_mask

功能：使用位掩码的每个像素的逻辑与。

### 4. bit\_not

功能：对像素的所有位求补。

### 5. bit\_or

功能：输入图像的所有像素的逐位或。

### 6. bit\_rshift

功能：图像的所有像素的右移。

## 7. bit\_slice

功能：从像素中提取一位。

## 8. bit\_xor

功能：输入图像的所有像素的逐位异或。

## 5.3 Color

### 1. cfa\_to\_rgb

功能：把一个单通道颜色滤波阵列图像变成 RGB 图像。

### 2. gen\_principal\_comp\_trans

功能：计算多通道图像的主要部分分析的转换矩阵。



### 3. linear\_trans\_color

功能：计算多通道图像的颜色值的一个仿射转换。

### 4. principal\_comp

功能：计算多通道图像的主要部分。

### 5. rgb1\_to\_gray

功能：把一个 RGB 图像转变成一个灰度图像。

### 6. rgb3\_to\_gray

功能：把一个 RGB 图像转变成一个灰度图像。

### 7. trans\_from\_rgb

功能：把一个图像从 RGB 颜色空间转变成任意颜色空间。

### 8. trans\_to\_rgb

功能：把一个图像从任意颜色空间转变成 RGB 颜色空间。

## 5.4 Edges

### 1.close\_edges

功能：使用边缘幅值图像消除边缘缺陷。

### 2. close\_edges\_length

功能：使用边缘幅值图像消除边缘缺陷。

### 3. derivate\_gauss

功能：用高斯派生物对一个图像卷积。

### 4. diff\_of\_gauss

功能：近似高斯的拉普拉斯算子。

## 5. edges\_color

功能：使用 Canny、Deriche 或者 \_Shen\_ 滤波器提取颜色边缘。

## 6. edges\_color\_sub\_pix

功能：使用 Canny、Deriche 或者 \_Shen\_ 滤波器提取子像素精确颜色边缘。

## 7. edges\_image

功能：使用 Deriche、\_Lanser、Shen 或者 \_Canny 滤波器提取边缘。

## 8. edges\_sub\_pix

功能：使用 Deriche、\_Lanser、Shen 或者 \_Canny 滤波器提取子像素精确边缘。

## 9. frei\_amp

功能：使用 Frei-Chen 算子检测边缘（幅值）。

#### 10. frei\_dir

功能：使用 Frei-Chen 算子检测边缘（幅值和相位）。

#### 11. highpass\_image

功能：从一个图像提取高频成分。

#### 12. info\_edges

功能：在 edges\_image 估计滤波器的宽度。

#### 13. kirsch\_amp

功能：使用 Kirsch 算子检测边缘（幅值）。

#### 14. kirsch\_dir

功能：使用 Kirsch 算子检测边缘（幅值和相位）。

#### 15. laplace

功能：使用有限差计算拉普拉斯算子。

#### 16. laplace\_of\_gauss

功能：高斯的拉普拉斯算子。

#### 17. prewitt\_amp

功能：使用 Prewitt 算子检测边缘（幅值）。

#### 18. prewitt\_dir

功能：使用 Prewitt 算子检测边缘（幅值和相位）。

#### 19. roberts

功能：使用 Roberts 滤波器检测边缘。

#### 20. robinson\_amp

功能：使用 Robinson 算子检测边缘（幅值）。

## 21. robinson\_dir

功能：使用 Robinson 算子检测边缘（幅值和相位）。

## 22. sobel\_amp

功能：使用 Sobel 算子检测边缘（幅值）。

## 23. sobel\_dir

功能：使用 Sobel 算子检测边缘（幅值和相位）。

## 5.5 Enhancement

### 1. adjust\_mosaic\_images

功能：全景图像的自动颜色更改。

### 2. coherence\_enhancing\_diff

功能：执行一个图像的一个一致性增强扩散。

### 3. emphasize

功能：增强图像对比度。

### 4. equ\_histo\_image

功能：图像的柱状图线性化。

### 5. illuminate

功能：增强图像对比度。

### 6. mean\_curvature\_flow

功能：把平均曲率应用在一个图像中。

### 7. scale\_image\_max\_

功能：最大灰度值在 0 到 255 范围内。

## 8. shock\_filter

功能：把一个冲击滤波器应用到一个图像中。

## 5.6 FFT

### 1. convol\_fft

功能：用在频域内的滤波器使一个图像卷积。

### 2. convol\_gabor

功能：用在频域内的一个 **Gabor** 滤波器使一个图像卷积。

### 3. correlation\_fft

功能：计算在频域内的两个图像的相互关系。

### 4. energy\_gabor



功能：计算一个两通道图像的能量。

## 5. `fft_generic`

功能：计算一个图像的快速傅里叶变换。

## 6. `fft_image`

功能：计算一个图像的快速傅里叶变换。

## 7. `fft_image_inv`

功能：计算一个图像的快速傅里叶逆变换。

## 8. `gen_bandfilter`

功能：生成一个理想带通滤波器。

## 9. `gen_bandpass`

功能：生成一个理想带通滤波器。

#### 10. `gen_derivative_filter`

功能：在频域内生成一个倒数滤波器。

#### 11. `gen_filter_mask`

功能：在空域内存储一个滤波器掩码作为实时图像。

#### 12. `gen_gabor`

功能：生成一个 Gabor 滤波器。

#### 13. `gen_gauss_filter`

功能：在频域内生成一个高斯滤波器。

#### 14. `gen_highpass`

功能：生成一个理想高通滤波器。

#### 15. `gen_lowpass`

功能：生成一个理想低通滤波器。

#### 16. `gen_sin_bandpass`

功能：用正弦形状生成一个带通滤波器。

#### 17. `gen_std_bandpass`

功能：用高斯或者正弦形状生成一个带通滤波器。

#### 18. `optimize_fft_speed`

功能：使 FFT 的运行时间最优化。

#### 19. `optimize_rft_speed`

功能：使实值的 FFT 的运行时间最优化。

#### 20. `phase_deg`

功能：返回用角度表示的一个复杂图像的相位。

## 21. phase\_rad

功能：返回用弧度表示的一个复杂图像的相位。

## 22. power\_byte

功能：返回一个复杂图像的功率谱。

## 23. power\_ln\_

功能：返回一个复杂图像的功率谱。

## 24. power\_real

功能：返回一个复杂图像的功率谱。

## 25. read\_fft\_optimization\_data

功能：从一个文件中下载 FFT 速度最优数据。

## 26. rft\_generic

功能：计算一个图像的实值快速傅里叶变换。

## 27. write\_fft\_optimization\_data

功能：把 FFT 速度最优数据存储在一个文件中。

## 5.7 Geometric-Transformations

### 1. affine\_trans\_image

功能：把任意仿射 2D 变换应用在图像中。

### 2. affine\_trans\_image\_size

功能：把任意仿射 2D 变换应用在图像中并且指定输出图像大小。

### 3. gen\_bundle\_adjusted\_mosaic

功能：把多重图像合成一个马赛克图像。

#### 4. gen\_cube\_map\_mosaic

功能：创建球形马赛克的 6 方位图像。

#### 5. gen\_projective\_mosaic

功能：把多重图像合成一个马赛克图像。

#### 6. gen\_spherical\_mosaic

功能：创建一个球形马赛克图像。

#### 7. map\_image

功能：把一个一般变换应用于一个图像中。

#### 8. mirror\_image

功能：镜像一个图像。

#### 9. polar\_trans\_image

功能：把一个图像转换成极坐标。

#### 10. polar\_trans\_image\_ext

功能：把一个图像中的环形弧转变成极坐标。

#### 11. polar\_trans\_image\_inv

功能：把极坐标中的图像转变成直角坐标。

#### 12. projective\_trans\_image

功能：把投影变换应用于一个图像中。

#### 13. projective\_trans\_image\_size

功能：把投影变换应用于一个图像中并且指定输出图像的大小。

#### 14. rotate\_image

功能：以一个图像的中心为圆心旋转。

## 15. zoom\_image\_factor

功能：把一个图像缩放规定因子倍。

## 16. zoom\_image\_size

功能：把一个图像缩放到规定大小。

## 5.8 Inpainting

### 1. harmonic\_interpolation

功能：对一个图像区域执行谐波插值。

### 2. inpainting\_aniso

功能：通过各向异性扩散执行图像修复。

### 3. inpainting\_ced



功能：通过一致性增强扩散执行图像修复。

#### 4. inpainting\_ct

功能：通过连贯传送执行图像修复。

#### 5. inpainting\_mcf

功能：通过水平线平滑执行图像修复。

#### 6. inpainting\_texture

功能：通过结构传导执行图像修复。

### 5.9 Lines

#### 1. bandpass\_image

功能：使用带通滤波器提取边缘。

## 2. lines\_color

功能：检测色线和它们的宽度。

## 3. lines\_facet

功能：使用面模型检测线。

## 4. lines\_gauss

功能：检测线和它们的宽度。

## 5.10 Match

### 1. exhaustive\_match

功能：模板和图像的匹配。

### 2. exhaustive\_match\_mg

功能：在一个分辨率塔式结构中匹配模板和图像。

### 3. gen\_gauss\_pyramid

功能：计算一个高斯金字塔。

### 4. monotony

功能：计算单一操作。

## 5.11 Misc

### 1. convol\_image

功能：用一个任意滤波掩码对一个图像卷积。

### 2. expand\_domain\_gray

功能：扩大图像区域并且在扩大的区域中设置灰度值。

### 3. gray\_inside

功能：对图像中的每一点在图像边界的任意路径计算尽可能低的灰度值。

### 4. gray\_skeleton

功能：灰度值图像的细化。

### 5. lut\_trans

功能：使用灰度值查询表转换一个图像。

### 6. symmetry

功能：沿一行的灰度值的对称性。

### 7. topographic\_sketch

功能：计算一个图像的地理原始草图。

## 5.12 Noise

### 1. add\_noise\_distribution

功能：向一个图像添加噪声。

### 2. add\_noise\_white

功能：向一个图像添加噪声。

### 3. gauss\_distribution

功能：产生一个高斯噪声分布。

### 4. noise\_distribution\_mean

功能：测定一个图像的噪声分布。

### 5. sp\_distribution

功能：产生一个椒盐噪声分布。

## 5.13 Optical-Flow

### 1. optical\_flow\_mg

功能：计算两个图像之间的光流。

### 2. unwarp\_image\_vector\_field

功能：使用一个矢量场来展开一个图像。

### 3. vector\_field\_length

功能：计算一个矢量场的矢量长度。

## 5.14 Points

### 1. corner\_response

功能：在图像中寻找角点。

## 2. dots\_image

功能：在一个图像中增强圆形点。

## 3. points\_foerstner

功能：使用 Förstner 算子检测关注点。

## 4. points\_harris

功能：使用 Harris 算子检测关注点。

## 5. points\_sojka

功能：使用 Sojka 算子找出角点。

## 5.15 Smoothing

### 1. anisotrope\_diff

功能：通过保边各向异性扩散平滑一个图像。

### 2. anisotropic\_diffusion

功能：对一个图像执行各向异性扩散。

### 3. binomial\_filter

功能：使用 binomial 滤波器平滑一个图像。

### 4. eliminate\_min\_max

功能：在空域内平滑一个图像来抑制噪声。

### 5. eliminate\_sp

功能：用中值替代阈值外的值。

### 6. fill\_interlace



功能：插补两个半个视频图像。

## 9. gauss\_image

功能：使用离散高斯函数平滑图像。

## 10. info\_smooth

功能：平滑滤波器 `smooth_image` 的信息。

## 11. isotropic\_diffusion

功能：对一个图像执行各向同性扩散。

## 12. mean\_image

功能：通过平均平滑一个图像。

## 13. mean\_n

功能：几个通道的平均灰度值。

#### 14. mean\_sp

功能：抑制椒盐噪声。

#### 15. median\_image

功能：使用不同级别掩码的中值滤波。

#### 16. median\_separate\_

功能：使用矩形掩码的离散中值滤波。

#### 17. median\_weighted

功能：使用不同级别掩码的加权中值滤波。

#### 18. midrange\_image

功能：计算掩码内最大和最小值的平均。

#### 19. rank\_image

功能：通过一个任意等级掩码平滑一个图像。

## 20. sigma\_image

功能：使用 **sigma** 滤波器的非线性平滑。

## 21. smooth\_image

功能：使用递归滤波器平滑一个图像。

## 22. trimmed\_mean

功能：使用任意等级掩码平滑一个图像。

## 5.16 Texture

### 1. deviation\_image

功能：计算矩形窗口内的灰度值的标准偏差。

## 2. entropy\_image

功能：计算矩形窗口内的灰度值的熵。

## 3. texture\_laws

功能：使用一个 Laws 文本滤波器过滤一个图像。

## 5.17 Wiener-Filter

### 1. gen\_psf\_defocus

功能：产生一个均匀散焦模糊的脉冲相应。

### 2. gen\_psf\_motion

功能：产生一个（线性）运动模糊的脉冲相应。

### 3. simulate\_defocus

功能： 对一个图像的均匀散焦模糊进行仿真。

#### 4. simulate\_motion

功能：（线性）运动模糊的仿真。

#### 5. wiener\_filter

功能： 通过 Wiener 滤波进行图像恢复。

#### 6. wiener\_filter\_ni

功能： 通过 Wiener 滤波进行图像恢复。

### Chapter\_6 :Graphics

#### 6.1 Drawing

##### 1. drag\_region1

功能： 一个区域的交互运动。

## 2. drag\_region2

功能：一个带有定点规格区域的交互运动。

## 3. drag\_region3

功能：一个带有限制位置区域的交互运动。

## 4. draw\_circle

功能：一个圆的交互绘图。

## 5. draw\_circle\_mod

功能：一个圆的交互绘图。

## 6. draw\_ellipse

功能：一个椭圆的交互绘图。

## 7. draw\_ellipse\_mod\_

功能：一个椭圆的交互绘图。

## 8. draw\_line

功能：画一根线。

## 9. draw\_line\_mod

功能：画一根线。

## 10. draw\_nurbs

功能：一个 NURBS 曲线的交互绘图。

## 11. draw\_nurbs\_interp

功能：使用插值的一个 NURBS 曲线的交互绘图。

## 12. draw\_nurbs\_interp\_mod

功能：使用插值的一个 NURBS 曲线的交互修正。

### 13. draw\_nurbs\_mod

功能：一个 NURBS 曲线的交互修正。

### 14. draw\_point

功能：画一个点。

### 15. draw\_point\_mod

功能：画一个点。

### 16. draw\_polygon

功能：一个多边形的交互绘图。

### 17. draw\_rectangle1

功能：画一个与坐标轴平行的矩形。

### 18. draw\_rectangle1\_mod



功能：画一个与坐标轴平行的矩形。

#### 19. draw\_rectangle2

功能：任意定向矩形的交互绘图。

#### 20. draw\_rectangle2\_mod

功能：任意定向矩形的交互绘图。

#### 21. draw\_region

功能：一个闭区域的交互绘图。

#### 22. draw\_xld

功能：一个轮廓的交互绘图。

#### 23. draw\_xld\_mod

功能：一个轮廓的交互修正。

## 6.2 Gnuplot

### 1. gnuplot\_close

功能：关闭所有打开的 **gnuplot** 文件或者终止一个活动的 **gnuplot** 子流程。

### 2. gnuplot\_open\_file

功能：为图像和控制量的可视化打开一个 **gnuplot** 文件。

### 3. gnuplot\_open\_pipe

功能：为图像和控制量的可视化打开一个通道的 **gnuplot** 流程。

### 4. gnuplot\_plot\_ctrl

功能：使用 **gnuplot** 显示控制量。

## 5. `gnuplot_plot_funct_1d`

功能：使用 `gnuplot` 显示控制量的功能。

## 6. `gnuplot_plot_image`

功能：使用 `gnuplot` 使一个图像可视化。

## 6.3 LUT

### 1. `disp_lut`

功能：查询表的图解。

### 2. `draw_lut`

功能：交互利用查询表。

### 3. `get_fixed_lut`

功能：为实际彩色图像获取固定查询表。

#### 4. get\_lut

功能：获取现在的查询表。

#### 5. get\_lut\_style

功能：获取查询表的修正参数。

#### 6. query\_lut

功能：查询所有可得到的查询表。

#### 7. set\_fixed\_lut

功能：为实际彩色图像固定查询表。

#### 8. set\_lut

功能：设置查询表。

## 9. set\_lut\_style

功能：改变查询表。

## 10. write\_lut

功能：把查询表作为文件写入。

## 6.4 Mouse

### 1. get\_mbutton

功能：等待直到一个鼠标键被按下。

### 2. get\_mposition

功能：查询鼠标位置。

### 3. get\_mshape

功能：查询现在鼠标指针形状。

#### 4. query\_mshape

功能：查询所有可得到的鼠标指针形状。

#### 5. set\_mshape

功能：设置现在鼠标指针形状。

### 6.5 Output

#### 1. disp\_arc

功能：在一个窗口中显示圆形弧。

#### 2. disp\_arrow

功能：在一个窗口中显示箭头。

### 3. disp\_channel

功能：用几个通道显示图像。

### 4. disp\_circle

功能：在一个窗口中显示圆。

### 5. disp\_color

功能：显示一个彩色（RGB）图像。

### 6. disp\_cross

功能：在一个窗口中显示交叉。

### 7. disp\_distribution

功能：显示一个噪声分布。

### 8. disp\_ellipse

功能：显示椭圆。

## 9. disp\_image

功能：显示灰度值图像。

## 10. disp\_line

功能：在窗口中画一条线。

## 11. disp\_obj

功能：显示图像目标（图像，区域，XLD）。

## 12. disp\_polygon

功能：显示一个多叉线。

## 13. disp\_rectangle1

功能：显示和坐标轴对齐的矩形。



#### 14. disp\_rectangle2

功能：显示任意方向的矩形。

#### 15. disp\_region

功能：在一个窗口中显示区域。

#### 16. disp\_xld

功能：显示一个 XLD 物体。

### 6.6 Parameters

#### 1. get\_comprise

功能：获取一个图像矩阵的输出处理。

#### 2. get\_draw

功能：获取现在区域填充模式。

### 3. get\_fix

功能：获取现在查询表的固定模式。

### 4. get\_hsi

功能：获取现在颜色的 HSI 编码。

### 5. get\_icon

功能：查询区域输出的图标。

### 6. get\_insert

功能：获取现在显示模式。

### 7. get\_line\_approx

功能：获取轮廓显示的现在近似误差。

## 8. get\_line\_style

功能：获取轮廓的现在图解模式。

## 9. get\_line\_width

功能：获取轮廓显示的现在线宽。

## 10. get\_paint

功能：获取灰度值的现在显示模式。

## 11. get\_part

功能：获取图像部分。

## 12. get\_part\_style

功能：获取灰度值显示的现在插值模式。

## 13. get\_pixel

功能：获取查询表索引的现在颜色。

#### 14. get\_rgb

功能：获取 RGB 编码中的现在颜色。

#### 15. get\_shape

功能：获取现在区域输出形状。

#### 16. query\_all\_colors

功能：查询所有颜色名称。

#### 17. query\_color

功能：查询窗口中显示的所有颜色名称。

#### 18. query\_colored

功能：查询颜色输出的颜色数目。

#### 19. query\_gray

功能：查询显示的灰度值。

#### 20. query\_insert

功能：查询可能的图解模式。

#### 21. query\_line\_width

功能：查询可能的线宽。

#### 22. query\_paint

功能：查询灰度值显示模式。

#### 23. query\_shape

功能：查询区域显示模式。

#### 24. set\_color

功能：设置输出颜色。

## 25. set\_colored

功能：设置多输出颜色。

## 26. set\_comprise

功能：定义图像矩阵输出剪辑。

## 27. set\_draw

功能：定义区域填充模式。

## 28. set\_fix

功能：设置固定的查询表。

## 29. set\_gray

功能：定义区域输出的灰度值。

### 30. set\_hsi

功能：定义输出颜色（HSI 编码）。

### 31. set\_icon

功能：区域输出的图标定义。

### 32. set\_insert

功能：定义图像输出功能。

### 33. set\_line\_approx

功能：定义输出显示的近似误差。

### 34. set\_line\_style

功能：定义一个轮廓输出模式。

### 35. set\_line\_width

功能：定义区域轮廓输出的线宽。

### 36. set\_paint

功能：定义灰度值输出模式。

### 37. set\_part

功能：修正显示图像部分。

### 38. set\_part\_style

功能：为灰度值输出定义一个插值方法。

### 39. set\_pixel

功能：定义一个颜色查询表索引。

### 40. set\_rgb

功能：通过 RGB 值设置颜色定义。



## 41. set\_shape

功能：定义区域输出轮廓。

## 6.7 Text

### 1. get\_font

功能：获取现在字体。

### 2. get\_string\_extents

功能：获取一个字符串的空间大小。

### 3. get\_tposition

功能：获取光标位置。

### 4. get\_tshape

功能：获取文本光标的形状。

## 5. new\_line

功能：设置下一行的开始文本光标的位置。

## 6. query\_font

功能：查询可得到的字体。

## 7. query\_tshape

功能：查询文本光标的所有可得到的形状。

## 8. read\_char

功能：从一个文本窗口读取一个字符。

## 9. read\_string

功能：从一个文本窗口读取一个字符串。

## 10. set\_font

功能：设置文本输出的字体。

## 11. set\_tposition

功能：设置文本光标的位置。

## 12. set\_tshape

功能：设置文本光标的形状。

## 13. write\_string

功能：在一个窗口中打印文本。

# 6.8 Window

## 1. clear\_rectangle

功能：在输出窗口中删除一个矩形。

## 2. `clear_window`

功能：删除一个输出窗口。

## 3. `close_window`

功能：关闭一个输出窗口。

## 4. `copy_rectangle`

功能：在输出窗口间复制矩形内所有像素。

## 5. `dump_window`

功能：把窗口内容写入一个文件。

## 6. `dump_window_image`

功能：在一个图像目标中写窗口内容。

## 7. get\_os\_window\_handle

功能：获取操作系统图像处理。

## 8. get\_window\_attr

功能：获取窗口特征。

## 9. get\_window\_extents

功能：一个窗口大小和位置的信息。

## 10. get\_window\_pointer3

功能：一个窗口像素数据的通道。

## 11. get\_window\_type

功能：获取窗口类型。

## 12. move\_rectangle

功能：在一个输出窗口内部复制。

### 13. new\_extern\_window

功能：在 Windows\_NT 下创建一个虚拟图形窗口。

### 14. open\_textwindow

功能：打开一个文本窗口。

### 15. open\_window

功能：打开一个图形窗口。

### 16. query\_window\_type

功能：查询所有可得到的窗口类型。

### 17. set\_window\_attr

功能：设置窗口特征。

## 18. set\_window\_dc

功能： 设置一个虚拟图形窗口（Windows\_NT）的设计背景。

## 19. set\_window\_extents

功能： 修正一个窗口的位置和大小。

## 20. set\_window\_type

功能： 指定一个窗口类型。

## 21. slide\_image

功能： 两个窗口缓冲区的交互输出。

## Chapter\_7\_:Image

### 7.1 Access

#### 1. get\_grayval

功能：获取一个图像目标的灰度值。

## 2. get\_image\_pointer1

功能：获取一个通道的指针。

## 3. get\_image\_pointer1\_rect

功能：获取图像数据指针和输入图像区域内最小矩形内部的图像数据。

## 4. get\_image\_pointer3

功能：获取一个彩色图像的指针。

## 5. get\_image\_time

功能：查找图像被创建的时间。



## 7.2 Acquisition

### 1. close\_all\_framegrabbers

功能：关闭所有图像获取设备。

### 2. close\_framegrabber

功能：关闭指定的图像获取设备。

### 3. get\_framegrabber\_lut

功能：查找图像获取设备的查询表。

### 4. get\_framegrabber\_param

功能：查找一个图像获取设备的指定参数。

### 5. grab\_data

功能：从指定的图像获取设备获取图像和预处理图像数据。

## 6. grab\_data\_async

功能：从指定的图像获取设备获取图像和预处理图像数据并且开始下一个异步获取。

## 7. grab\_image

功能：从指定的图像获取设备获取一个图像。

## 8. grab\_image\_async

功能：从指定的图像获取设备获取一个图像并且开始下一个异步获取。

## 9. grab\_image\_start

功能：从指定的图像获取设备开始下一个异步获取。

## 10. info\_framegrabber

功能：从指定的图像获取设备查找信息。

### 11. open\_framegrabber

功能：打开并配置一个图像获取设备。

### 12. set\_framegrabber\_lut

功能：设置图像获取设备查询表。

### 13. set\_framegrabber\_param

功能：设置一个图像获取设备的指定参数。

## 7.3 Channel

### 1. access\_channel

功能：获取一个多通道图像的一个通道。

### 2. append\_channel

功能：把附加模型（通道）添加到图像上。

### 3. channels\_to\_image

功能：把单通道图像转变为一个多通道图像。

### 4. compose2

功能：把两个图像转变为一个两通道图像。

### 5. compose3

功能：把三个图像转变为一个三通道图像。

### 6. compose4

功能：把四个图像转变为一个四通道图像。

### 7. compose5

功能：把五个图像转变为一个五通道图像。

## 8. compose6

功能：把六个图像转变为一个六通道图像。

## 9. compose7

功能：把七个图像转变为一个七通道图像。

## 10. count\_channels

功能：计算图像的通道。

## 11. decompose2

功能：把一个两通道图像转变为两个图像。

## 12. decompose3

功能：把一个三通道图像转变为三个图像。

## 13. decompose4

功能：把一个四通道图像转变为四个图像。

#### 14. decompose5

功能：把一个五通道图像转变为五个图像。

#### 15. decompose6

功能：把一个六通道图像转变为六个图像。

#### 16. decompose7

功能：把一个七通道图像转变为七个图像。

#### 17. image\_to\_channels

功能：把一个多通道图像转变为一个通道图像。

### 7.4 Creation

### 1. copy\_image

功能：复制一个图像并为它分配新内存。

### 2. gen\_image1

功能：从像素的一个指针创建一个图像。

### 3. gen\_image1\_extern

功能：从带存储管理的像素的一个指针创建一个图像。

### 4. gen\_image1\_rect

功能：从像素（带存储管理）的指针创建一个矩形区域的图像。

### 5. gen\_image3

功能：从像素（红、绿、蓝）的三个指针创建一个图像。

### 6. gen\_image\_const

功能：创建一个固定灰度值的图像。

## 7. `gen_image_gray_ramp`

功能：创建一个灰度值阶梯。

## 8. `gen_image_interleaved`

功能：从交叉像素的一个指针创建一个三通道图像。

## 9. `gen_image_proto`

功能：创建一个指定的固定灰度值的图像。

## 10. `gen_image_surface_first_order`

功能：创建一阶多项式的一个弯曲灰度表面。

## 11. `gen_image_surface_second_order`

功能：创建二阶多项式的一个弯曲灰度表面。



## 12. region\_to\_bin

功能：把一个区域转变为一个二进制字节图像。

## 13. region\_to\_label

功能：把区域转变为一个标签图像。

## 14. region\_to\_mean

功能：用它们的平均灰度值绘制区域。

## 7.5 Domain

### 1. add\_channels

功能：把两个灰度值添加到区域中。

### 2. change\_domain

功能：改变一个图像的定义区间。

### 3. full\_domain

功能：把一个图像的区域扩大到最大值。

### 4. get\_domain

功能：获取一个图像的区域。

### 5. rectangle1\_domain

功能：把一个图像的区域缩小到一个矩形。

### 6. reduce\_domain

功能：缩小一个图像的区域。

## 7.6 Features

### 1. area\_center\_gray

功能：计算一个灰度值图像的区域面积和重心。

### 2. cooc\_feature\_image

功能：计算一个同时出现的矩阵并得出相关灰度值特征。

### 3. cooc\_feature\_matrix

功能：从一个同时出现的矩阵计算灰度值特征。

### 4. elliptic\_axis\_gray

功能：在一个灰度值图像中计算一个区域的方位和主轴。

### 5. entropy\_gray

功能：确定一个图像的熵和各向异性。

### 6. estimate\_noise

功能：从一个单一图像估计图像噪声。

#### 7. fit\_surface\_first\_order

功能：通过一个一阶表面（平面）计算灰度值力矩和近似值。

#### 8. fit\_surface\_second\_order

功能：通过一个二阶表面（平面）计算灰度值力矩和近似值。

#### 9. fuzzy\_entropy

功能：确定区域的模糊熵。

#### 10. fuzzy\_perimeter

功能：计算一个区域的模糊周长。

#### 11. gen\_cooc\_matrix

功能：在一个图像中计算一个区域中同时出现的矩阵。

## 12. gray\_histo

功能：计算灰度值分布。

## 13. gray\_histo\_abs

功能：计算灰度值分布。

## 14. gray\_projections

功能：计算水平和垂直灰度值预测。

## 15. histo\_2dim

功能：计算两通道灰度值图像的直方图。

## 16. intensity

功能：计算灰度值的平均值和偏差。

## 17. min\_max\_gray

功能：计算区域内的最大和最小灰度值。

#### 18. moments\_gray\_plane

功能：通过一个平面计算灰度值力矩和近似值。

#### 19. plane\_deviation

功能：从近似像平面计算灰度值的偏差。

#### 20. select\_gray

功能：选择基于灰度值特征的区域。

#### 21. shape\_histo\_all

功能：用极限值确定特征的一个直方图。

#### 22. shape\_histo\_point

功能：用极限值确定特征的一个直方图。

## 7.7 Format

### 1. change\_format

功能：改变图像大小。

### 2. crop\_domain

功能：去掉确定的灰度值。

### 3. crop\_domain\_rel

功能：去掉和定义域有关的图像区域。

### 4. crop\_part

功能：去掉一个矩形图像区域。

### 5. crop\_rectangle1

功能：去掉一个矩形图像区域。

## 6. tile\_channels

功能：把多重图像拼成一个大图像。

## 7. tile\_images

功能：把多重图像目标拼成一个大图像。

## 8. tile\_images\_offset

功能：把多重图像目标拼成一个有确定的位置信息的大图像。

## 7.8 Manipulation

### 1. overpaint\_gray

功能：重新绘制一个图像的灰度值。



## 2. overpaint\_region

功能：重新绘制一个图像的区域。

## 3. paint\_gray

功能：把一个图像的灰度值画在另一个图像上。

## 4. paint\_region

功能：把区域画在一个图像中。

## 5. paint\_xld

功能：把 XLD 目标画在一个图像中。

## 6.set\_grayval

功能：在一个图像中设置单灰度值。

## 7.9 Type-Conversion

### 1. complex\_to\_real

功能：把一个复杂图像转变为两个实际图像。

### 2. convert\_image\_type

功能：转变一个图像的类型。

### 3. real\_to\_complex

功能：把两个实际图像转变为一个复杂图像。

### 4. real\_to\_vector\_field

功能：把两个实值图像转变为一个矢量域图像。

### 5. vector\_field\_to\_real

功能：把一个矢量域图像转变为两个实值图像。

## Chapter\_8\_:Lines

### 8.1 Access

#### 1. approx\_chain

功能：通过弧和线近似一个轮廓。

#### 2. approx\_chain\_simple

功能：通过弧和线近似一个轮廓。

### 8.2 Features

#### 1. line\_orientation

功能：计算线的方位。

#### 2. line\_position

功能：计算一条线的重心、长度和方位。

### 3. partition\_lines

功能：通过各种标准区分线。

### 4. select\_lines

功能：通过各种标准选择线。

### 5. select\_lines\_longest

功能：选择最长输入线。

## Chapter\_9\_:Matching

### 9.1 Component-Based

#### 1. clear\_all\_component\_models

功能：释放所有组件模型的内存。

## 2. clear\_all\_training\_components

功能：释放所有组件训练结果的内存。

## 3. clear\_component\_model

功能：释放一个组件模型的内存。

## 4. clear\_training\_components

功能：释放一个组件训练结果的内存。

## 5. cluster\_model\_components

功能：把用于创建模型组件的新参数用于训练结果。

## 6. create\_component\_model

功能：基于确定的指定组件和关系准备一个匹配的组件模型。

## 7. create\_trained\_component\_model

功能：基于训练过的组件准备一个匹配的组件模型。

#### 8. find\_component\_model

功能：在一个图像中找出一个组件模型的最佳匹配。

#### 9. gen\_initial\_components

功能：提取一个组件模型的最初组件。

#### 10. get\_component\_model\_params

功能：返回一个组件模型的参数。

#### 11. get\_component\_model\_tree

功能：返回一个组件模型的查找树。

#### 12. get\_component\_relations

功能：返回包含在训练结果内的模型组件间的关系。

### 13. get\_found\_component\_model

功能：返回一个组件模型的一个创建例子的组件。

### 14. get\_training\_components

功能：在一个特定的图像中返回初始值或者模型组件。

### 15. inspect\_clustered\_components

功能：检查从训练获取的刚性的模型组件。

### 16. modify\_component\_relations

功能：修改一个训练结果中的关系。

### 17. read\_component\_model

功能：从一个文件中读取组件模型。

### 18. read\_training\_components

功能： 从一个文件中读取组件训练结果。

#### 19. train\_model\_components

功能： 为基于组件的匹配训练组件和关系。

#### 20. write\_component\_model

功能： 把一个组件模型写入一个文件中。

#### 21. write\_training\_components

功能： 把一个组件训练结果写入一个文件中。

### 9.2 Correlation-Based

#### 1. clear\_all\_ncc\_models

功能： 释放 NCC 模型的内存。



## 2. clear\_ncc\_model

功能：释放 NCC 模型的内存。

## 3. create\_ncc\_model

功能：为匹配准备一个 NCC 模型。

## 4. find\_ncc\_model

功能：找出一个图像中的一个 NCC 模型的最佳匹配。

## 5. get\_ncc\_model\_origin

功能：返回一个 NCC 模型的原点（参考点）。

## 6. get\_ncc\_model\_params

功能：返回一个 NCC 模型的参数。

## 7. read\_ncc\_model

功能：从一个文件中读取一个 NCC 模型。

## 8. set\_ncc\_model\_origin

功能：设置一个 NCC 模型的原点（参考点）。

## 9. write\_ncc\_model

功能：向一个文件中写入 NCC 模型。

## 9.3 Gray-Value-Based

### 1. adapt\_template

功能：把一个模板用于一个图像的大小。

### 2. best\_match

功能：寻找一个模板和一个图像的最佳匹配。

### 3. best\_match\_mg

功能：在金字塔中寻找最佳灰度值匹配。

### 4. best\_match\_pre\_mg

功能：在预生成的金字塔中寻找最佳灰度值匹配。

### 5. best\_match\_rot

功能：寻找一个模板和一个旋转图像的最佳匹配。

### 6. best\_match\_rot\_mg

功能：寻找一个模板和一个旋转金字塔的最佳匹配。

### 7. clear\_all\_templates

功能：所有模板的内存分配。

### 9. clear\_template

功能：一个模板的内存分配。

#### 10. create\_template

功能：为模板匹配准备一个格式。

#### 11. create\_template\_rot

功能：为旋转模板匹配准备一个格式。

#### 12. fast\_match

功能：寻找一个模板和一个图像的所有好的匹配。

#### 13. fast\_match\_mg

功能：在金字塔中寻找所有好的灰度值匹配。

#### 14. read\_template

功能：从一个文件中读取一个模板。

### 15. set\_offset\_template

功能：模板的灰度值偏差。

### 16. set\_reference\_template

功能：为一个匹配模板定义参考位置。

### 17. write\_template

功能：向一个文件中写入模板。

## 9.4 Shape-Based

### 1. clear\_all\_shape\_models

功能：释放所有轮廓模型的内存。

### 2. clear\_shape\_model

功能：释放一个轮廓模型的内存。

### 3. create\_aniso\_shape\_model

功能：为各向异性尺度不变匹配准备一个轮廓模型。

### 4. create\_scaled\_shape\_model

功能：为尺度不变匹配准备一个轮廓模型。

### 5. create\_shape\_model

功能：为匹配准备一个轮廓模型。

### 6. determine\_shape\_model\_params

功能：确定一个轮廓模型的参数。

### 7. find\_aniso\_shape\_model

功能：在一个图像中找出一个各向异性尺度不变轮廓的最佳匹配。

#### 8. find\_aniso\_shape\_models

功能：找出多重各向异性尺度不变轮廓模型的最佳匹配。

#### 9. find\_scaled\_shape\_model

功能：在一个图像中找出一个尺度不变轮廓模型的最佳匹配。

#### 10. find\_scaled\_shape\_models

功能：找出多重尺度不变轮廓模型的最佳匹配。

#### 11. find\_shape\_model

功能：在一个图像中找出一个轮廓模型的最佳匹配。

#### 12. find\_shape\_models

功能：找出多重轮廓模型的最佳匹配。

#### 13. get\_shape\_model\_contours

功能：返回一个轮廓模型的轮廓表示。

#### 14. get\_shape\_model\_origin

功能：返回一个轮廓模型的原点（参考点）。

#### 15. get\_shape\_model\_params

功能：返回一个轮廓模型的参数。

#### 16. inspect\_shape\_model

功能：创建一个轮廓模型的表示。

#### 17. read\_shape\_model

功能：从一个文件中读取一个轮廓模型。

#### 18. set\_shape\_model\_origin

功能：设置一个轮廓模型的原点（参考点）。



## 19. write\_shape\_model

功能： 向一个文件中写入一个轮廓模型。

## Chapter\_10\_:Matching-3D

### 1. affine\_trans\_object\_model\_3d

功能： 把一个任意有限 3D 变换用于一个 3D 目标模型。

### 2. clear\_all\_object\_model\_3d

功能： 释放所有 3D 目标模型的内存。

### 3. clear\_all\_shape\_model\_3d

功能： 释放所有 3D 轮廓模型的内存。

### 4. clear\_object\_model\_3d

功能： 释放一个 3D 目标模型的内存。

#### 5. clear\_shape\_model\_3d\_

功能：释放一个 3D 轮廓模型的内存。

#### 6. convert\_point\_3d\_cart\_to\_spher

功能：把直角坐标系中的一个 3D 点转变为极坐标。

#### 7. convert\_point\_3d\_spher\_to\_cart

功能：把极坐标中的一个 3D 点转变为直角坐标。

#### 8. create\_cam\_pose\_look\_at\_point

功能：从摄像机中心和观察方向创建一个 3D 摄像机位置。

#### 9. create\_shape\_model\_3d

功能：为匹配准备一个 3D 目标模型。

#### 10. find\_shape\_model\_3d

功能：在一个图像中找出一个 3D 模型的最佳匹配。

#### 11. `get_object_model_3d_params`

功能：返回一个 3D 目标模型的参数。

#### 12. `get_shape_model_3d_contours`

功能：返回一个 3D 轮廓模型视图的轮廓表示。

#### 13. `get_shape_model_3d_params`

功能：返回一个 3D 轮廓模型的参数。

#### 14. `project_object_model_3d`

功能：把一个 3D 目标模型的边缘投影到图像坐标中。

#### 15. `project_shape_model_3d`

功能：把一个 3D 轮廓模型的边缘投影到图像坐标中。

#### 16. read\_object\_model\_3d\_dxf

功能：从一个 DXF 文件中读取一个 3D 目标模型。

#### 17. read\_shape\_model\_3d

功能：从一个文件中读取一个 3D 轮廓模型。

#### 18. trans\_pose\_shape\_model\_3d

功能：把一个 3D 目标模型的坐标系中的位置转变为一个 3D 轮廓模型的参考坐标系中的位置，反之亦然。

#### 19. write\_shape\_model\_3d

功能：向一个文件写入一个 3D 轮廓模型。

### Chapter\_11\_:Morphology

#### 11.1 Gray-Values

### 1. dual\_rank

功能：打开、取中值和关闭圆和矩形掩码。

### 2. gen\_disc\_se

功能：为灰度形态学生成椭圆结构基础。

### 3. gray\_bothat

功能：执行一个图像的一个灰度值 **bottom\_hat** 变换（原图像和它的闭之间的差）。

### 4. gray\_closing

功能：关闭一个图像的一个灰度值。

### 5. gray\_closing\_rect

功能：关闭带矩形掩码的灰度值。

### 6. gray\_closing\_shape

功能：关闭带选择掩码的灰度值。

## 7. gray\_dilation

功能：扩大一个图像上的灰度值。

## 8. gray\_dilation\_rect

功能：确定一个矩形的最小灰度值。

## 9. gray\_dilation\_shape

功能：确定一个选择的掩码的最大灰度值。

## 10. gray\_erosion

功能：腐蚀一个图像的灰度值。

## 11. gray\_erosion\_rect

功能：确定一个矩形的最小灰度值。

## 12. gray\_erosion\_shape

功能：确定一个选择的掩码的最小灰度值。

## 13. gray\_opening

功能：打开一个图像的灰度值。

## 14. gray\_opening\_rect

功能：打开一个矩形掩码的灰度值。

## 15. gray\_opening\_shape

功能：打开一个选择的掩码的灰度值。

## 16. gray\_range\_rect

功能：确定一个矩形的灰度值范围。

## 17. gray\_tophat

功能：执行一个图像的一个灰度值 `top_hat` 变换（原图像和它的开之间的差）。

## 18. `read_gray_se`

功能：为灰度形态学下载一个结构基础。

## 11.2 Region

### 1. `bottom_hat`

功能：计算区域的 `bottom_hat`（原图像和它的闭之间的差）。

### 2. `boundary`

功能：把一个区域减小到它的边界。

### 3. `closing`



功能：关闭一个区域。

#### 4. closing\_circle

功能：关闭一个圆形结构基础的一个区域。

#### 5. closing\_golay

功能：关闭格雷字母表中的元素的一个区域。

#### 6. closing\_rectangle1

功能：关闭一个矩形结构基础的一个区域。

#### 7. dilation1

功能：扩大一个区域。

#### 8. dilation2

功能：扩大一个区域（使用一个参考点）。

## 9. dilation\_circle

功能：扩大一个圆形结构基础的一个区域。

## 10. dilation\_golay

功能：扩大格雷字母表的元素的一个区域。

## 11. dilation\_rectangle1

功能：扩大一个矩形结构基础的一个区域。

## 12. dilation\_seq

功能：顺序地扩大一个区域。

## 13. erosion1

功能：腐蚀一个区域。

## 14. erosion2

功能：腐蚀一个区域（使用参考点）。

#### 15. erosion\_circle

功能：腐蚀一个圆形结构基础的一个区域。

#### 16. erosion\_golay

功能：腐蚀格雷字母表的一个元素的一个区域。

#### 17. erosion\_rectangle1

功能：腐蚀一个矩形结构基础的一个区域。

#### 18. erosion\_seq

功能：按顺序腐蚀一个区域。

#### 19. fitting

功能：执行多重结构基础的打开后关闭。

## 20. gen\_struct\_elements

功能：生成一个标准结构基础。

## 21. golay\_elements

功能：生成格雷字母表的结构基础。

## 22. hit\_or\_miss

功能：区域的 Hit-or-miss 运行。

## 23. hit\_or\_miss\_golay

功能：使用格雷字母表的区域的 Hit-or-miss 运行。

## 24. hit\_or\_miss\_seq

功能：使用格雷字母表的区域的 Hit-or-miss 运行（按顺序）。

## 25. minkowski\_add1

功能：执行一个区域的 Minkowski 添加。

## 26. minkowski\_add2

功能：扩大一个区域（使用参考点）。

## 27. minkowski\_sub1

功能：腐蚀一个区域。

## 28. minkowski\_sub2

功能：腐蚀一个区域（使用参考点）。

## 29. morph\_hat

功能：计算 bottom\_hat\_和 top\_hat 的联合。

## 30. morph\_skeleton

功能：计算一个区域的形态学框架。

### 31. morph\_skiz

功能：缩小一个区域。

### 32. opening

功能：打开一个区域。

### 33. opening\_circle

功能：打开一个圆形结构基础的一个区域。

### 34. opening\_golay

功能：打开格雷字母表的一个元素的一个区域。

### 35. opening\_rectangle1

功能：打开一个矩形结构基础的一个区域。

### 36. opening\_seg

功能：分离重叠区域。

### 37. pruning

功能：去掉一个区域的分支。

### 38. thickening

功能：把一个 Hit-or-miss 运行的结果添加到一个区域。

### 39. thickening\_golay

功能：把一个 Hit-or-miss 运行的结果添加到一个区域中（使用一个 Golay 结构基础）。

### 40. thickening\_seq

功能：把一个 Hit-or-miss 运行的结果添加到一个区域中（按顺序）。

### 41. thinning

功能：从一个区域移去一个 Hit-or-miss 运行的结果。

#### 42. thinning\_golay

功能：从一个区域移去一个 Hit-or-miss 运行的结果（使用一个 Golay 结构基础）。

#### 43. thinning\_seq

功能：从一个区域移去一个 Hit-or-miss 运行的结果（按顺序）。

#### 44. top\_hat

功能：计算区域的 top\_hat（原图像和它的开之间的差）。

### Chapter\_12:OCR（光字符识别）

#### 12.1 Hyperboxes

##### 1. close all ocrs

功能：删除所有光字符，释放存储空间，但会丢失所有的测试数据。



## 2. close ocr\_

功能：重新分配拥有 OcrHandle 数目的分级器的存储，但所有相应的数据会丢失，不过这些数据可由 write ocr 事先保存。

## 3. create ocr class box

功能：创建新的 OCR 分级器。

## 4. do ocr multi

功能：给每一个 Character（字符）分配一个类。

## 5. do ocr single

功能：给一些 Character（字符）分配一些类。

## 6. info ocr class box

功能：反馈 ocr 的有关信息。

## 7. ocr change char

功能：为字符建立新的查阅表。

#### 8. ocr get features

功能：计算给定 Character（字符）的特征参数。

#### 9. read ocr

功能：从文件的 FileName（文件名）读取 OCR 分级器。

#### 10. testd ocr class box

功能：测试给定类中字符的置信度。

#### 11. traind ocr class box

功能：通过一幅图像的特定区域直接测试分级器。

#### 12. trainf ocr class box

功能：根据指定测试文件测试分级器的 OCRHandle。

### 13. write ocr

功能：将 OCR 分级器的 OCRHandle 写入文件的 FileName(文件名)。

## 12.2 Lexica

### 1.clear\_all\_lexica

功能：清除所有的词汇（词典），释放它们的资源。

### 2. clear lexicon

功能：清除一个词汇（词典），释放相应的资源。

### 3. create lexicon

功能：根据一些 Words(单词)的元组创建一个新的词汇（词典）。

### 4. Import lexicon

功能：通过 **FileName**(文件名)选定的文件中的一系列单词创建一个新的词典。

#### 5. inspect lexicon

功能：返回 **Words** 参数的词典中所有单词的元组。

#### 6. lookup lexicon

功能：检查 **Word**（单词）是否在词典的 **LexiconHandle** 中，若在返回 1 否则返回 0。

#### 7. suggest lexicon

功能：将 **Word**（单词）与词典中所有词汇相比较，计算出将 **Word** 从词典中导入单词中所需的足校的编辑操作符 **NUMcorrections**。

### 12.3 Neural-Nets（神经网络）

### 1. clear all ocr class mlp

功能：清除所有的 create ocr class mlp 创建的 OCR 分级器，释放分级器占据的存储空间。

### 2. clear ocr class mlp

功能：清除所有的由 OCRHandle 给定的且由 create ocr class mlp 创建的 OCR 分级器，释放所有的分级器占据的存储空间。

### 3. create ocr class mlp

功能：利用 MLP（多层感知器）创建一个新的 OCR 分级器。

### 4. do ocr multi class mlp

功能：为根据给定区域字符和 OCR 分级器 OCRHandle 的灰度图像值而给定的每个字符计算出最好的类，将类返回到 Class 中，且将类的置信度返回到 Confidence 中。

### 5. do ocr single class mlp

功能：为根据给定区域字符和 OCR 分级器 OCRHandle 的灰度图像值而给定的字符计算出最好的 Num 类，将类返回到 Class 中，且将类的置信度返回到 Confidence 中。

#### 6.do ocr word mlp

功能：功能与 do ocr multi class mlp 相同，只是 do ocr word mlp 将字符组作为一个实体。

#### 7.get\_features\_ocr\_class\_mlp

功能：为根据 OCR 分级器 OCRHandle 确定的字符计算其特征参数，并将它们返回到 Features。

#### 8.get\_params ocr class mlp

功能：返回一个 OCR 分级器的参数只有当分级器由 do ocr multi class mlp 创建时。

#### 9.get\_prep\_info ocr class mlp

功能：计算 OCR 分级器预设定矢量特性的信息。

#### 10. read ocr class mlp

功能：从一个文件中读取 OCR 分级器。

#### 11. trainf ocr class mlp

功能：测试 OCR 分级器的 OCRHandle，根据存储在 OCR 文件中的测试特性。

#### 12. write ocr class mlp

功能：将 OCR 分级器的 OCRHandle 写入由文件名确定的文件中。

### 12.4 Support-Vector-Machines\_（支持矢量机）

#### 1. clear\_all ocr class svm

功能：清除所有的基于 OCR 分级器的 SVM，释放相应的存储空间。

## 2. `clear ocr class svm`

功能：清除基于 OCR 分级器的一个 SVM，释放相应的存储空间。

## 3. `create ocr class svm`

功能：利用支持向量机创建一个 OCR 分级器。

## 4. `do ocr multi class svm`

功能：根据基于 OCR 分级器的 SVM 将大量字符分类。

## 5. `do ocr single class svm`

功能：根据基于 OCR 分级器的 SVM 将单个字符分类。

## 6. `do ocr word svm`

功能：利用 OCR 分级器将一系列相关字符分类。

## 7. `get features ocr class svm`



功能：计算一个字符的特征。

#### 8. `get_params ocr class svm`

功能：返回一个 OCR 分级器的参数。

#### 9. `get_prep_info ocr class svm`

功能：计算基于 OCR 分级器的 SVM 的预定义特征矢量的信息内容。

#### 10. `get_support_vector num ocr class svm`

功能：返回 OCR 分级器支持的矢量的数目。

#### 11. `get_support_vector ocr class svm`

功能：返回基于支持向量机的已测试 OCR 分级器中支持向量的索引。

#### 12. `read ocr class svm`

功能：从文件中读取基于 OCR 分级器的 SVM。

### 13. reduce ocr class svm

功能：根据一个减小的 SVM 来接近一个基于 OCR 分级器的 SVM。

### 14. Trainf ocr class svm

功能：测试一个 OCR 分级器。

### 15. write ocr class svm

功能：将一个 OCR 分级器写入文件。

## 12.5 Tools

### 1. Segment characters

功能：将一副图像给定区域的字符分割。

### 2. select characters

功能：从一个给定区域中选择字符。

### 3. text\_line\_orientation

功能：决定一个文本行或段落的定向（定位）。

### 4. text\_line\_slant

功能：决定一个文本行或段落的字符的倾斜。

## 12.6 Training-Files

### 1. append ocr trainf

功能：将字符添加到一个测试文件中。

### 2. concat ocr trainf

功能：合并测试文件。

### 3. read\_ocr\_trainf

功能：从文件中读取字符，将其转换到图像中。

### 4. read\_ocr\_trainf\_names

功能：查询哪些字符存储在测试文件中。

### 5. read\_ocr\_trainf\_select

功能：从文件中读取测试特定字符，将其转换到图像中。

### 6. write\_ocr\_trainf

功能：将已测试的字符存储到文件中。

### 7. write\_ocr\_trainf\_image

功能：将字符写入正在测试的文件中。

## Chapter\_13:Object

### 13.1 Information

#### 1. count\_obj

功能：统计一个元组中的对象。

#### 2. get\_channel\_info

功能：一幅目标图像组成部分的信息。

#### 3. get\_obj\_class

功能：一副目标图像类的名称。

#### 4. test\_equal\_obj

功能：比较目标图像的平等性。

#### 5. test\_obj\_def

功能：测试目标是否被删除。

## 13.2 Manipulation

### 1. clear\_obj

功能：将一个对象的图标从 HALCON 数据库中删除。

### 2. concat\_obj

功能：连接两个目标元组的图标。

### 3. copy\_obj

功能：复制一个 HALCON 数据库中对象的图标\_。

### 4. gen\_empty\_obj

功能：创建一个空的目标元组。

## 5. integer\_to\_obj

功能：将一个整型数转换为一个图标。

## 6. obj\_to\_integer

功能：将一个图标转换为一个整型数。

## 7. select\_obj

功能：从一个目标元组中选择目标。

# Chapter\_14:Regions

## 14.1 Access

### 1. get\_region\_chain

功能：一个对象的轮廓(contour)作为链式码。

### 2. get\_region\_contour

功能：查询一个目标的轮廓(contour)。

### 3. get\_region\_convex

功能：查询突起的外表作为轮廓(contour)。

### 4. get\_region\_points

功能：查询一个区域的像素数。

### 5. get\_region\_polygon

功能：用一个多边形近似获取区域。

### 6. get\_region\_runs

功能：查询一个区域的扫描宽度编码。

## 14.2 Creation



### 1. gen\_checker\_region

功能：创建一个方格式区域。

### 2. gen\_circle

功能：创建一个圆周。

### 3. gen\_ellipse

功能：创建一个椭圆。

### 4. gen\_empty\_region

功能：创建一个空的区域。

### 5. gen\_grid\_region

功能：根据行或像素数创建一个区域\_。

### 6. gen\_random\_region

功能:创建一个随机区域。

## 7. gen\_random\_regions

功能：创建随机区域如圆周，矩形和椭圆。

## 8. gen\_rectangle1

功能：创建一个与坐标轴平行的长方形。

## 9. gen\_rectangle2

功能：创建任意方向的矩形。

## 10. gen\_region\_contour\_xld

功能：从 XLD 元组中创建一个区域。

## 11. gen\_region\_histo

功能：将一个直方图转换为一个区域。

## 12. gen\_region\_hline

功能：将 Hesse 正规形状中描述的输入线存储为区域。

## 13. gen\_region\_line

功能：将输入线以区域形式存储。

## 14. gen\_region\_points

功能：将个别的像素存储为图像区域。

## 15. gen\_region\_polygon

功能：将一个多边形存储为一个目标图像。

## 16. gen\_region\_polygon\_filled

功能：将一个多边形存储为一个已填充区域。

## 17. gen\_region\_polygon\_xld

功能：创建一个 XLD 多边形中的区域。

## 18. gen\_region\_runs

功能：创建一个扫描宽度编码中的图像区域。

## 19. label\_to\_region

功能：提取一幅图像中灰度值相同的区域。

## 14.3 Features

### 1. area center

功能：一个区域的面积（大小）和中心。

### 2. circularity

功能：影响一个区域与圆的相似度的形状系数。

### 3. compactness

功能：影响一个区域致密度的形状系数。

### 4. connect and\_holes

功能：连接部分和中断的数目。

### 5. contlength

功能：描述一个区域轮廓(contour)的长度。

### 6. convexity

功能：影响一个区域凸性的形状系数。

### 7. diameter region

功能：一个区域两个边界点的最大距离。

### 8. eccentricity

功能：来源于椭圆参数的形状系数。

## 9. elliptic\_axis

功能：相似椭圆的参数。

## 10. euler\_number

功能：计算 Euler 数目。

## 11. find\_neighbors

功能：搜寻直接邻域。

## 12. get\_region\_index

功能：包括给定像素在内的所有的区域的索引。

## 13. get\_region\_thickness

功能：查询主轴附近区域的宽度（厚度）。

#### 14. hamming distance

功能：两个区域间的汉明距离。

#### 15. hamming\_distance\_norm

功能：两个区域间的归一化汉明距离。

#### 16. inner\_circle

功能：一个区域内部最大的圆周。

#### 17. inner\_rectangle1

功能：一个区域内部最大的矩形。

#### 18. moments\_region\_2nd

功能：区域的某时刻几何特性，。

#### 19. moments\_region\_2nd\_invar

功能：区域的某时刻几何特性。

#### 20. moments\_region\_2nd\_rel\_invar

功能：计算相关时刻参数。

#### 21. moments\_region\_3rd

功能：区域的某时刻几何特性。

#### 22. moments\_region\_3rd\_invar

功能：区域的某时刻几何特性。

#### 23. moments\_region\_central

功能：区域的某时刻几何特性。

#### 24. moments\_region\_central\_invar

功能：区域的某时刻几何特性。



## 25. orientation\_region

功能：一个区域的定向。

## 26. rectangularity

功能：影响一个区域矩形相似度的形状系数。

## 27. roundness

功能：轮廓中获取的形状系数。

## 28. runlength\_distribution

功能：一个区域扫描宽度编码所需的顺串的分配。

## 29. runlength\_features

功能：区域扫描宽度编码的特征值。

## 30. select\_region\_point

功能：选择包括给定像素在内的所有区域。

### 31. select\_region\_spatial

功能：讨论区域的关联性。

### 32. select\_shape

功能：根据图形特征选择区域。

### 33. select\_shape\_proto

功能：选择彼此有某种关系的区域。

### 34. select\_shape\_std

功能：选择给定形状的区域。

### 35. smallest\_circle

功能：一个区域的最小周长。

### 36. smallest\_rectangle1

功能：平行于坐标轴的包围某区域的矩形。

### 37. smallest\_rectangle2

功能：任意方向包围某区域的最小矩形。

### 38. spatial\_relation

功能：根据坐标轴方向左、右、上、下排列相关区域。

## 14.4 Geometric-Transformations

### 1. affine\_trans\_region

功能：对区域进行任意的二维变换。

### 2. mirror\_region

功能：反馈一个平行于  $x$  或  $y$  坐标轴的区域。

### 3. move\_region

功能：对区域进行变换。

### 4. polar\_trans\_region

功能：将一个环状弧内的区域转换为极坐标。

### 5. polar\_trans\_region\_inv

功能：将极坐标中的区域转换为笛卡尔坐标中的区域。

### 6. projective\_trans\_region

功能：对一个区域进行射影变换。

### 7. transpose\_region

功能：翻译关于一个点的一个区域。

## 8. zoom\_region

功能：缩放一个区域。

## 14.5 Sets

### 1. complement

功能：返回一个区域的补码。

### 2. difference

功能：计算两个区域的差距（不同）。

### 3. intersection

功能：计算两个区域的交集。

### 4. symm\_difference

功能：计算两个区域对称差异。

## 5. union1

功能：返回所有输入区域的并集。

## 6. union2

功能：返回两个区域的并集。

## 14.6 Tests

### 1. test\_equal\_region

功能：检测两个目标区域是否相同。

### 2. test\_subset\_region

功能：检测一个区域是否包含在另一个区域中。

## 14.7 Transformation

### 1. background\_seg

功能：决定给定区域背景相连的部分。

### 2. clip\_region

功能：将一个区域修改为矩形。

### 3. clip\_region rel

功能：根据大小修改一个区域。

### 4. connection

功能：计算一个区域相连接的部分。

### 5. distance transform

功能：计算一个区域的距离变换。

## 6. eliminate\_runs

功能：消除一个给定宽度的顺串。

## 7. expand\_region

功能：填充区域间的间隙或分离互相重叠的区域。

## 8. fill\_up

功能：填充区域中的中断（裂缝等）。

## 9. fill\_up\_shape

功能：填充拥有给定图形特征区域的中断。

## 10. hamming\_change\_region

功能：创建一个有给定汉明距离的区域。



## 11. interjacent

功能：利用给定区域分割图像。

## 12. junctions\_skeleton

功能：找到框架中的结点和终点。

## 13. merge\_regions\_line\_scan

功能：从行扫描图像合并区域。

## 14. partition\_dynamic

功能：在较小垂直范围的位置水平分割一个区域。

## 15. partition\_dynamic

功能：将一个区域分割为等大的矩形。

## 16. rank\_region

功能：给对区域的操作归类。

#### 17. remove\_noise\_region

功能：去除一个区域内的噪声。

#### 18. shape\_trans

功能：改变一个区域的形状。

#### 19. skeleton

功能：计算一个区域的框架。

#### 20. sort\_region

功能：根据相邻位置归类区域。

#### 21. split\_skeleton\_lines

功能：用一个像素宽，没有分支的线来分离线。

## 22. split\_skeleton\_region

功能：用一个像素宽，没有分支的区域来分离线。

## Chapter\_15:Segmentation

### 15.1 Classification

#### 1. add\_samples\_image\_class\_gmm

功能：将从图像中获取的测试样本添加到高斯混合模型的测试数据库中。

#### 2. add\_samples\_image\_class\_mlp

功能：将从图像中获取的测试样本添加到多层视感控器的测试数据库中。

#### 3. add\_samples\_image\_class\_svm

功能：将从图像中获取的测试样本添加到一个支持向量机的测试数

据库中。

#### 4. class\_2dim\_sup

功能：采用二维空间像素分类分割图像。

#### 5. class 2dim unsup

功能：将两幅图像以聚类分割。

#### 6.class ndim\_box

功能：利用立方体将像素分类。

#### 7. class\_ndim\_norm

功能：利用球体或立方体将像素分类。

#### 8. classify\_image\_class\_gmm

功能：根据高斯混合模式分类图像。

## 9. `classify_image_class_mlp`

功能：根据多层视感控器分类图像\_。

## 10. `classify_image_class_svm`

功能：根据支持向量机分类图像。

## 11. `learn_ndim_box`

功能：利用多通道图像测试一个分级器。

## 12. `learn_ndim_norm`

功能：为 `class_ndim_norm` 构建类。

## 15.2 Edges

### 1. `detect_edge_segments`

功能：检测直线边缘分割。

## 2. hysteresis\_threshold

功能：对一副图像采取磁滞门限操作。

## 3. nonmax\_suppression\_amp

功能：抑制一幅图像上的非最大值点。

## 4. nonmax\_suppression\_dir

功能：利用指定图像抑制一幅图像上的非最大值点。

## 15.3 Regiongrowing

### 1. expand\_gray

功能：依据灰度值或颜色填充两个区域的间隙或分割重叠区域。

## 2. expand\_gray\_ref

功能：依据灰度值或颜色填充两个区域的间隙或分割重叠区域。

## 3. expand\_line

功能：从给定线开始扩充区域。

## 4. regiongrowing

功能：利用区域增长分割图像。

## 5. regiongrowing\_mean

功能：利用平均灰度值执行区域增长。

## 6. regiongrowing\_n

功能：利用区域增长为多通道图像分割图像。

## 15.4 Threshold

### 1. auto\_threshold

功能：根据直方图决定的阈值分割图像。

### 2. bin\_threshold

功能：根据自动产生的阈值分割图像。

### 3. char\_threshold

功能：为提取的字符产生一个分割阈值。

### 4. check\_difference

功能：一个像素一个像素的比较两幅图像。

### 5. dual\_threshold

功能：对标记的图像做门限操作。\_



## 6. dyn\_threshold

功能：利用局域阈值分割图像。

## 7. fast\_threshold

功能：利用全局阈值快速将图像二值化。

## 8. histo\_to\_thresh

功能：根据直方图决定灰度值门限。

## 9. threshold

功能：利用全局阈值分割图像。

## 10. threshold\_sub\_pix

功能：根据子像素的准确性从一副图像中提取水平（平坦）交叉口。

## 11. var\_threshold

功能：根据局域平均标准偏差分析将图像二值化。

## 12. zero\_crossing

功能：从一幅图像中提取零相交。

## 13. zero\_crossing\_sub\_pix

功能：根据子像素准确性从一幅图像中提取零相交。

## 15.5 Topography

### 1. critical\_points\_sub\_pix

功能：一幅图像中主要点的子像素精确度检测。

### 2. local\_max

功能：检测一幅图像中所有的最大数。

### 3. local\_max\_sub\_pix

功能：一幅图像中局域最大数的子像素精确度检测。

### 4. local\_min

功能：检测一幅图像中所有的最小数。

### 5. local\_min\_sub\_pix

功能：一幅图像中局域最小数的子像素精确度检测。

### 6. lowlands

功能：检测凹地所有灰度值。

### 7. lowlands\_center

功能：检测凹地所有灰度值的中心。

### 8. plateaus

功能：检测所有平稳状态灰度值。

#### 9. plateaus\_center

功能：检测所有平稳状态灰度值的中心。

#### 10. pouring

功能：根据大于“pouring\_water”分割图像。

#### 11. saddle\_points\_sub\_pix

功能：一幅图像中底部点的子像素精确度检测。

#### 12. watersheds

功能：从一副图像中提取分界线和“盆地”。

#### 13. watersheds\_threshold

功能：利用阈值从一幅图像中提取“分水岭盆地”。

## Chapter\_16: System

### 16.1 Database

#### 1. count\_relation

功能：在 HALCON 数据库中实体的数目。

#### 2. get\_modules

功能：查询已使用模块和模块关键码。

#### 3. reset\_obj\_db

功能：HALCON 系统的初始化。

### 16.2 Error-Handling

#### 1. get\_check

功能：HALCON 控制模式的说明。

## 2. get\_error\_text

功能：查询 HALCON 错误测试后错误数目。

## 3. get\_spy

功能：HALCON 调试工具当前配置。

## 4. query\_spy

功能：查询 HALCON 调试工具可能的设置。

## 5. set\_check

功能：激活和钝化 HALCON 控制模式。

## 6. set\_spy

功能：HALCON 调试工具的控制。

## 16.3 Information

### 1. get\_chapter\_info

功能：获取程序有关章节的信息。

### 2. get\_keywords

功能：获取指定给程序的关键字。

### 3. get\_operator\_info

功能：获取关于 HALCON 程序的信息。

### 4. get\_operator\_name

功能：获取由给定字符串作为它们的名字的程序。

### 5. get\_param\_info

功能：获取关于程序参数的信息。

## 6. get\_param\_names

功能：获取一个 HALCON 程序参数的名字。

## 7. get\_param\_num

功能：获取一个 HALCON 程序不同参数类的数目。

## 8. get\_param\_types

功能：获取一个 HALCON 程序控制参数的缺省数据类型。

## 9. query\_operator\_info

功能：联合操作 `get_operator_info` 查询空档相关信息。

## 10. query\_param\_info

功能：查询关于操作 `get_param_info` 的空档的在线信息。



## 11. search\_operator

功能：寻找一个关键字所有进程的名字。

## 16.4 Operating-System

### 1. count\_seconds

功能：衡量时间。

### 2. system\_call

功能：执行系统请求。

### 3. wait\_seconds

功能：延迟操作的执行。

## 16.5 Parallelization

### 1. check\_par\_hw\_potential

功能：检测硬件进行并行处理的潜力。

### 2. load\_par\_knowledge

功能：从文件中导入自动平行化信息。

### 3. store\_par\_knowledge

功能：在文件中存储关于自动平行化的信息。

## 16.6 Parameters

### 1. get\_system\_

功能：根据 HALCON 系统参数获取关于当前的信息。

## 2. set\_system

功能：HALCON 系统参数的设置。

## 16.7 Serial

### 1. clear\_serial

功能：清除一个串行连接的缓冲。

### 2. close\_all\_serials

功能：关闭所有的串行设备。

### 3. close\_serial

功能：关闭一个串行设备。

### 4. get\_serial\_param

功能：获取一个串行设备的参数。

## 5. open\_serial

功能：打开一个串行设备。

## 6. read\_serial

功能：读取一个串行设备。

## 7. set\_serial\_param

功能：设置一个串行设备的参数。

## 8. write\_serial

功能：写入一个串行设备。

## 16.8 Sockets

### 1. close\_socket

功能：关闭一个插口（接口）。

### 2. get\_next\_socket\_data\_type

功能：决定下一个插口（接口）数据的 HALCON 数据类型。

### 3. get\_socket\_timeout

功能：获取一个插口（接口）的超时。

### 4. open\_socket\_accept

功能：打开一个接受连接请求的插口（接口）。

### 5. open\_socket\_connect

功能：打开一个插口到一个已存在的插口。

### 6. receive\_image

功能：通过插口连接接收一副图像。

## 7. receive\_region

功能：通过插口连接接收区域。

## 8. receive\_tuple

功能：通过插口连接接收一个元组。

## 9. receive\_xld

功能：通过插口连接接收一个 XLD 对象。

## 10. send\_image

功能：通过插口连接发送一副图像。

## 11. send\_region

功能：通过插口连接发送区域。

## 12. send\_tuple

功能：通过插口连接发送一个元组。

## 13. send\_xld

功能：通过插口连接发送一个 XLD 对象。

## 14. set\_socket\_timeout

功能：设置一个插口的超时。

## 15. socket\_accept\_connect

功能：接受一个监听插口的连接请求。

## Chapter\_17:Tools

### 17.1 2D-Transformations

#### 1. affine\_trans\_pixel

功能：对像素坐标轴进行任意的仿射二维变换。

## 2. `affine_trans_point_2d`

功能：对点进行任意的最简二维变换

## 3. `bundle_adjust_mosaic`

功能：对一幅图像的嵌合体采取一系列调整。

## 4. `hom_mat2d_compose`

功能：将两种相同类型二维变换矩阵相乘。

## 5. `hom_mat2d_determinant`

功能：计算一个同质的二维变换矩阵的行列式。

## 6. `hom_mat2d_identity`

功能：构建二维变换同样的同质变换矩阵。



## 7. hom\_mat2d\_invert

功能：插入一个同质二维变换矩阵。

## 8. hom\_mat2d\_rotate

功能：为一个同质二维变换矩阵添加一个循环。

## 9. hom\_mat2d\_rotate\_local

功能：为一个同质二维变换矩阵添加一个循环。

## 10. hom\_mat2d\_scale

功能：为一个同质二维变换矩阵添加一个缩放。

## 11. hom\_mat2d\_scale\_local

功能：为一个同质二维变换矩阵添加一个缩放。

## 12. hom\_mat2d\_slant

功能： 为一个同质二维变换矩阵添加一个斜面。

### 13. hom\_mat2d\_slant\_local

功能： 为一个同质二维变换矩阵添加一个斜面。

### 14. hom\_mat2d\_to\_affine\_par

功能： 计算一个来自一个同质二维变换矩阵的仿射变换参数。

### 15. hom\_mat2d\_translate

功能： 为一个同质二维变换矩阵添加一个旋转。

### 16. hom\_mat2d\_translate\_local

功能： 为一个同质二维变换矩阵添加一个旋转。

### 17. hom\_mat2d\_transpose

功能： 将一个同质二维变换矩阵转置。

#### 18. `hom_mat3d_project`

功能：给一个二维投影变换矩阵投影一个仿射三维变换矩阵。

#### 19. `hom_vector_to_proj_hom_mat2d`

功能：根据给定点的映射计算一个同质变换矩阵。

#### 20. `proj_match_points_ransack`

功能：通过找到两副图像中点与点之间的映射计算一个投影变换矩阵。

#### 21. `projective_trans_pixel`

功能：利用一个同质投影变换矩阵表示像素坐标轴。

#### 22. `projective_trans_point_2d`

功能：利用一个投影变换矩阵表示一个同质二维点。

### 23. vector\_angle\_to\_rigid

功能：从点和角度方面计算一个严格的仿射变换。

### 24. vector\_field\_to\_hom\_mat2d

功能：根据位移矢量字段获取一个最接近的近似图。

### 25. vector\_to\_hom\_mat2d

功能：根据点与点间的映射获取一个最接近的近似图

### 26. vector\_to\_proj\_hom\_mat2d

功能：利用给定点的映射计算一个映射变换矩阵。

### 27. vector\_to\_rigid

功能：根据点的映射获取一个近似严格的仿射变换。

### 28. vector\_to\_similarity

功能：根据点的映射获取一个近似的相似变换。

## 17.2 3D-Transformations

### 1. `affine_trans_point_3d`

功能：对点运用一个随即仿射三维变换。

### 2. `convert_pose_type`

功能：改变一个三维模式的表示类型。

### 3. `create_pose`

功能：创建一个三维模式。

### 4. `get_pose_type`

功能：获取一个三维模式的表示类型。

## 5. hom\_mat3d\_compose

功能：将两个同质三维变换矩阵相乘。

## 6. hom\_mat3d\_identity

功能：构建三维变换同样的同质变换矩阵。

## 7. hom\_mat3d\_invert

功能：插入一个同质三维变换矩阵。

## 8. hom\_mat3d\_rotate

功能：为一个同质三维变换矩阵添加一个循环。

## 9. hom\_mat3d\_rotate\_local

功能：为一个同质三维变换矩阵添加一个循环。

## 10. hom\_mat3d\_scale

功能：为一个同质三维变换矩阵添加一个缩放。

#### 11. `hom_mat3d_scale_local`

功能：为一个同质三维变换矩阵添加一个缩放。

#### 12. `hom_mat3d_to_pose`

功能：将一个同质变换矩阵转换为一个三维模式。

#### 13. `hom_mat3d_translate`

功能：为一个同质三维变换矩阵添加一个旋转。

#### 14. `hom_mat3d_translate_local`

功能：为一个同质三维变换矩阵添加一个旋转。

#### 15. `_pose_to_hom_mat3d`

功能：将一个三位模式转换为一个同质变换矩阵。

## 16. read\_pose

功能：从一个文本文件中读取一个三维模式。

## 17. set\_origin\_pose

功能：转换一个三位模式的原点。

## 18. write\_pose

功能：将一个三维模式写入一个文本文件。

## 17.3 Background-Estimator

### 1. close\_all\_bg\_esti

功能：清除所有的背景评估数据集。

### 2. close\_bg\_esti



功能：清除背景估测数据集。

### 3. create\_bg\_esti

功能：为背景评估创建和初始化一个数据集。

### 4. get\_bg\_esti\_params

功能：返回数据集的参数。

### 5. give\_bg\_esti

功能：返回估测背景图像。

### 6. run\_bg\_esti

功能：评估背景并返回前景区域。

### 7. set\_bg\_esti\_params

功能：改变数据集的参数。

## 8. update\_bg\_esti

功能：改变估测背景图像。

## 17.4 Barcode

### 1. clear\_all\_bar\_code\_models

功能：清除所有条形码模型，释放其分配的存储空间。

### 2. clear\_bar\_code\_model

功能：清除一个条形码模型，释放相应的存储空间。

### 3. create\_bar\_code\_model

功能：创建一个条形码阅读器模型。

### 4. find\_bar\_code

功能：检测和读取一幅图像中条形码符号。

#### 5. `get_bar_code_object`

功能：访问创建在搜寻或条形码符号解码过程中的对象图标。

#### 6. `get_bar_code_param`

功能：获取一个或多个描述条形码模式的参数。

#### 7. `get_bar_code_result`

功能：获取字母数字混合编码的结果，其是在条形码符号解码过程中累计的。

#### 8. `set_bar_code_param`

功能：设置条形码模型的选定参数。

### 17.5 Calibration

### 1. caltab\_points

功能：从校准板说明文件中读取标志中心点。

### 2. cam\_mat\_to\_cam\_par

功能：计算从一个相机矩阵获取的内部相机参数。

### 3. cam\_par\_to\_cam\_mat

功能：从相机内部参数计算一个相机矩阵。

### 4. camera\_calibration

功能：决定同时发生的最小化程序的所有相机参数。

### 5. change\_radial\_distortion\_cam\_par

功能：根据与特殊放射失真相一致决定新的相机参数。

### 6. change\_radial\_distortion\_contours\_xld

功能：改变了轮廓(contour)的放射失真。

## 7. change\_radial\_distortion\_image

功能：改变一幅图像的放射失真。

## 8. contour\_to\_world\_plane\_xld

功能：将一个 XLD 轮廓(contour)转换为一个坐标系统中平面 Z 为零。

## 9. create\_caltab

功能：创建一个描述文件和附文件的校准板。

## 10. disp\_caltab

功能：投射和视觉化图像中校准板的三维模型。

## 11. find\_caltab

功能：分割和标准化图像中的校准板区域。

## 12. find\_marks\_and\_pose

功能：从图像中提取二维校准标志和为外部计算机参数计算内部数值。

## 13. gen\_caltab

功能：创建一个校准板说明文件和相应的附文件。

## 14. gen\_image\_to\_world\_plane\_map

功能：创建一个投射图，其描述图像平面与坐标轴系统中平面  $Z$  为零之间的映射。

## 15. gen\_radial\_distortion\_map

功能：创建一个投射图，其描述图像与其相应正在改变的放射失真间的映射。

## 16. get\_circle\_pose

功能： 从一个圆周相应的二维投射中决定它的三维模式。

#### 17. get\_line\_of\_sight

功能： 计算相应于图像中一个点的视线。

#### 18. get\_rectangle\_pose

功能： 从一个矩形相应的二维投射中决定它的三维模式。

#### 19. hand\_eye\_calibration

功能： 执行一个手---眼校准。

#### 20. image\_points\_to\_world\_plane

功能： 将图像中的点转换到坐标轴平面  $z$  为零上。

#### 21. image\_to\_world\_plane

功能： 通过将一副图像转换为坐标轴系统中平面  $z$  为零而矫正图像。

## 22. project\_3d\_point

功能：将三维点投射到子像素图像坐标。

## 23. radiometric\_self\_calibration

功能：执行一个相机的辐射测量的自校准。

## 24. read\_cam\_par

功能：从文本文件中读取内部相机参数。

## 25. sim\_caltab

功能：根据校准板模拟一幅图像。

## 26. stationary\_camera\_self\_calibration

功能：投射一个静止投射相机的自校准。

## 27. write\_cam\_par



功能：将内部相机参数写入文本文件中。

## 17.6 Datacode

### 1. clear\_all\_data\_code\_2d\_models

功能：清除所有的二维数据模型并释放它们分配的存储空间。

### 2. clear\_data\_code\_2d\_model

功能：清除一个二维数据模型并释放它分配的存储空间。

### 3. create\_data\_code\_2d\_model

功能：创建一个二维数据编码类的模式。

### 4. find\_data\_code\_2d

功能：检测和读取一副图像或测试的二维数据编码模式中的二维数

据编码符号。

#### 5. `get_data_code_2d_objects`

功能：查询搜索二维数据编码符号过程中创建的对象图标。

#### 6. `get_data_code_2d_param`

功能：获取一个或多个描述二维数据编码模型的参数。

#### 7. `get_data_code_2d_results`

功能：获取字母数字混合编码的结果，其是在搜索二维数据编码符号过程中累计的。

#### 8. `query_data_code_2d_params`

功能：为一个给定二维数据编码模型获取通用参数或对象的名字，其也可用于其他的二维数据编码模型中。

#### 9. `read_data_code_2d_model`

功能：从一个文件中读取一个二维数据编码模型并新建一个模型。

#### 10. set\_data\_code\_2d\_param

功能：设置二维数据编码模型的选定参数。

#### 11. write\_data\_code\_2d\_model

功能：将一个二维数据编码模型写入一个文件。

### 17.7 Fourier-Descriptor

#### 1. abs\_invar\_fourier\_coeff

功能：根据起始点的位移标准化傅里叶系数。

#### 2. fourier\_1dim

功能：计算一个参数化的元组的傅里叶系数。

### 3. `fourier_1dim_inv`

功能：空间傅里叶变换（傅里叶逆变换）。

### 4. `invar_fourier_coeff`

功能：傅里叶系数标准化。

### 5. `match_fourier_coeff`

功能：两个元组的相似性。

### 6. `move_contour_orig`

功能：将原点变换到引力的中心。

### 7. `prep_contour_fourier`

功能：参数化传输的元组。

## 17.8 Function

### 1. abs\_func1d

功能：Y 值的绝对值。

### 2. compose\_func1d

功能：组合两个函数。

### 3. create\_func1d\_array

功能：从 Y 值的序列中创建一个函数。

### 4. create\_func1d\_pairs

功能：从 (X, Y) 集合中创建一个函数。

### 5. derivate\_func1d

功能：计算一个函数的派生物。

## 6. distance\_funct\_1d

功能：计算两个函数的间隔。

## 7. funct\_1d\_to\_pairs

功能：查询一个函数的（X，Y）值。

## 8. get\_pair\_funct\_1d

功能：根据控制点的索引查询一个函数值。

## 9. get\_y\_value\_funct\_1d

功能：返回任意位置函数的值。

## 10. integrate\_funct\_1d

功能：计算一个函数的正区域和负区域。

## 11. invert\_funct\_1d

功能： 计算一个函数的反转。

#### 12. local\_min\_max\_funct\_1d

功能： 计算一个函数的局域最小和最大值点。

#### 13. match\_funct\_1d\_trans

功能： 计算两个函数传递参数。

#### 14. negate\_funct\_1d

功能： 对 Y 值取非（反）。

#### 15. num\_points\_funct\_1d

功能： 函数控制点的数目。

#### 16. read\_funct\_1d

功能： 从文件中读取一个函数。

#### 17. sample\_funct\_1d

功能：再间隔区等距取样。

#### 18. scale\_y\_funct\_1d

功能：将 Y 值相乘和相加。

#### 19. smooth\_funct\_1d\_gauss

功能：采用高斯函数平滑一个等距一维函数。

#### 20. smooth\_funct\_1d\_mean

功能：采用平均值将一个等距一维函数平滑化。

#### 21. transform\_funct\_1d

功能：根据给定传递参数变换你一个函数。

#### 22. write\_funct\_1d



功能：将一个函数写入一个文件。

### 23. x\_range\_funct\_1d

功能：函数的最小和最大  $x$  值。

### 24. y\_range\_funct\_1d

功能：函数的最小和最大  $y$  值。

### 25. zero\_crossings\_funct\_1d

功能：计算一个函数的零点。

## 17.9 Geometry

### 1. angle\_ll

功能：计算两条线的夹角。

## 2. angle\_lx

功能：计算一条线与垂直轴之间的角度。

## 3. distance\_cc

功能：计算两个轮廓(contour)间的距离。

## 4. distance\_cc\_min

功能：计算两个轮廓(contour)间的最小距离。

## 5. distance\_lc

功能：计算一条线和一个轮廓(contour)间的距离。

## 6. distance\_lr

功能：计算一条线和一个区域间的距离。

## 7. distance\_pc

功能：计算一个点和一个轮廓(contour)间的距离。

#### 8. distance\_pl

功能：计算一个点和一条线间的距离。

#### 9. distance\_pp

功能：计算两个点之间的距离。

#### 10. distance\_pr

功能：计算一个点和一个区域间的距离。

#### 11. distance\_ps

功能：计算一个点和一条分割线间的距离。

#### 12. distance\_rr\_min

功能：两个相邻区域的相同像素间的最小距离。

### 13. distance\_rr\_min\_dil

功能：膨胀时两个区域间的最小距离。

### 14. distance\_sc

功能：计算一条分割线和一个轮廓(contour)间的距离。

### 15. distance\_sl

功能：计算一条分割线和一条线间的距离。

### 16. distance\_sr

功能：计算一条分割线和一个区域间的距离。

### 17. distance\_ss

功能：计算两条分割线间的距离。

### 18. get\_points\_ellipse

功能：计算椭圆上特定角度的一个点。

## 19. intersection\_ll

功能：计算两条线的交点（相交点）。

## 20. projection\_pl

功能：计算一条线上一个点的投影。

## 17.10 Grid-Rectification

### 1. connect\_grid\_points

功能：建立矫正网格的矫正点间的连接。

### 2. create\_rectification\_grid

功能：建立一个附文件，描述矫正网格。

### 3. find\_rectification\_grid

功能：分割图像中矫正网格区域。

### 4. gen\_arbitrary\_distortion\_map

功能：产生一个投射图，其描述随意扭曲图像与正确图像间的映射。

### 5. gen\_grid\_rectification\_map

功能：计算扭曲图像与基于规律的网格的正确的图像的映射。

## 17.11 Hough

### 1. hough\_circle\_trans

功能：返回指定半径的圆周的 Hough 变换。

### 2. hough\_circles

功能：特定半径的圆周的圆心。

### 3. `hough_line_trans`

功能：对区域中的线进行 Hough 变换。

### 4. `hough_line_trans_dir`

功能：利用局部方向梯度对线进行 Hough 变换。

### 5. `hough_lines`

功能：借助 Hough 变化查询图像中的线，并将其返回到 HNF 中。

### 6. `hough_lines_dir`

功能：借助采用局部方向梯度的 Hough 变换查询图像中的线，并将它们以正常形式返回。

### 7. `select_matching_lines`

功能：选取 HNF 中线的集合中匹配区域最好的线。

## 17.12 Image-Comparison

### 1. clear\_all\_variation\_models

功能：释放所有变化模型（`variation_model`）的存储空间。

### 2. clear\_train\_data\_variation\_model

功能：释放变化模型（`variation_model`）的测试数据的存储空间。

### 3. clear\_variation\_model

功能：释放一个变化模型（`variation_model`）的存储空间。

### 4. compare\_ext\_variation\_model

功能：将一副图像与一个变化模型（`variation_model`）相比较。

### 5. compare\_variation\_model



功能：将一副图像与一个变化模型（`variation_model`）相比较。

#### 6. `create_variation_model`

功能：为图像对比创建一个变化模型。

#### 7. `get_thresh_images_variation_model`

功能：返回阈值图像用于图像对比。

#### 8. `get_variation_model`

功能：返回图像用于图像对比。

#### 9. `prepare_direct_variation_model`

功能：为图像对比准备一个变化模型。

#### 10. `prepare_variation_model`

功能：为图像对比准备一个变化模型。

## 11. read\_variation\_model

功能：从一个文件中读取一个变化模型。

## 12. train\_variation\_model

功能：测试一个变化模型。

## 13. write\_variation\_model

功能：将一个变化模型写入文件。

## 17.13 Kalman-Filter

### 1. filter\_kalman

功能：借助 Kalman（卡尔曼）滤波器估测系统的当前状态。

### 2. read\_kalman

功能：读取一个卡尔曼滤波器的说明文件。

### 3. sensor\_kalman

功能：卡尔曼滤波器测量值的交互式输入。

### 4. update\_kalman

功能：读取一个卡尔曼滤波器的更新文件。

## 17.14 Measure

### 1. close\_all\_measures

功能：清除所有测试对象。

### 2. close\_measure

功能：清除一个测试对象。

### 3. fuzzy\_measure\_pairing

功能：提取与矩形或环状弧垂直的直线边缘。

### 4. fuzzy\_measure\_pairs

功能：提取与矩形或环状弧垂直的直线边缘。

### 5. fuzzy\_measure\_pos

功能：提取与矩形或环状弧垂直的直线边缘。

### 6. gen\_measure\_arc

功能：垂直与环状弧的直线边缘的提取。

### 7. gen\_measure\_rectangle2

功能：垂直与矩形的直线边缘的提取。

### 8. measure\_pairs

功能：提取与矩形或环状弧垂直的直线边缘。

## 9. measure\_pos

功能：提取与矩形或环状弧垂直的直线边缘。

## 10. measure\_projection

功能：提取垂直于一个矩形或环状弧的灰度值轮廓(contour)。

## 11. measure\_thresh

功能：提取沿着一个矩形或环状弧，特殊灰度值的点。

## 12. reset\_fuzzy\_measure

功能：重置一个模糊元函数。

## 13. set\_fuzzy\_measure

功能：指定一个模糊元函数。

#### 14. set\_fuzzy\_measure\_norm\_pair

功能：为边缘匹配指定一个规范化模糊元函数。

#### 15. translate\_measure

功能：转化（解释）一个测试对象。

### 17.15 OCV（Open\_Circuit\_Voltage\_|\_光学字符校验）

#### 1. close\_all\_ocvs

功能：关闭所有 OCV 工具。

#### 2. close\_ocv

功能：关闭一个 OCV 工具。

#### 3. create\_ocv\_proj

功能：创建一个基于灰度值突出的新的 OCV 工具。

#### 4. do\_ocv\_simple

功能：利用一个 OCV 工具查证一个模式。

#### 5. read\_ocv

功能：从文件中读取一个 OCV 工具。

#### 6. traind\_ocv\_proj

功能：测试一个 OCV 工具。

#### 7. write\_ocv

功能：将一个 OCV 工具保存到文件。

### 17.16 Shape-from

### 1. depth\_from\_focus

功能：利用多倍聚焦灰度级提取高度（厚度）。

### 2. estimate\_al\_am

功能：估测一个平面的反射率和反射光的数目。

### 3. estimate\_sl\_al\_lr

功能：估测一个光源的倾斜度和一个平面的反射率。

### 4. estimate\_sl\_al\_zc

功能：估测一个光源的倾斜度和一个平面的反射率。

### 5. estimate\_tilt\_lr

功能：估测一个光源的倾斜。

### 6. estimate\_tilt\_zc



功能：估测一个光源的倾斜。

## 7. phot\_stereo

功能：根据至少三个灰度值的图像来重建一个平面。

## 8. select\_grayvalues\_from\_channels

功能：利用索引图像选择一个多通道图像的灰度值。

## 9. sfs\_mod\_lr

功能：从一个灰度值图像重建一个平面。

## 10. sfs\_orig\_lr

功能：从一个灰度值图像重建一个平面。

## 11. sfs\_pentland

功能：从一个灰度值图像重建一个平面。

## 12. shade\_height\_field

功能：遮蔽一个突起的字段。

## 17.17 Stereo

### 1. binocular\_calibration

功能：决定一个双目视觉立体系统的所有相机参数。

### 2. binocular\_disparity

功能：计算一个矫正图像对的不均衡。

### 3. binocular\_distance

功能：计算一个矫正立体图像对的间隔值。

### 4. disparity\_to\_distance

功能：将不均衡值转换为矫正双目视觉立体系统中的间隔值。

#### 5. disparity\_to\_point\_3d

功能：将一个图像点和它的不均衡值转换为一个矫正立体系统中的三维点。

#### 6.distance\_to\_disparity

功能：将一个间隔值转换为一个矫正立体系统中的一个不均衡值。

#### 7. essential\_to\_fundamental\_matrix

功能：计算一个从原始矩阵衍生而来的基本矩阵。

#### 8. gen\_binocular\_proj\_rectification

功能：计算弱双目视觉立体系统图像的投射矫正值。

#### 9. gen\_binocular\_rectification\_map\_

功能：创建传输图，其描述从一个双目相机到一个普通的矫正图像

面的图像的映射。

#### 10. `gen_binocular_rectification_map`

功能：从一个双目相机系统视觉中两条线的交点中获取一个三维点。

#### 11. `match_essential_matrix_ransack`

功能：通过自动发掘图像点间对应关系来计算立体图像对的原始（本质）矩阵。

#### 12. `match_fundamental_matrix_ransack`

功能：通过自动发掘图像点间对应关系来计算立体图像对的基本矩阵。

#### 13. `match_rel_pose_ransack`

功能：通过自动发掘图像点间对应关系来计算两个相机间的相对方位。

#### 14. reconst3d\_from\_fundamental\_matrix

功能：计算基于基本矩阵的点的投影的三维重建。

#### 15. rel\_pose\_to\_fundamental\_matrix

功能：计算两个相机相关方向中获取的基本矩阵。

#### 16. vector\_to\_essential\_matrix

功能：计算给定图像点间映射和已知相机矩阵的原始矩阵，重建三维点。

#### 17. vector\_to\_fundamental\_matrix

功能：计算给定图像点间映射的集合的基本矩阵，重建三维点。

#### 18. vector\_to\_fundamental\_matrix

功能：计算给定图像点间对应关系和已知相机参数的两个相机的相对方位，重建三维点。

## 17.18 Tools-Legacy

### 1. decode\_1d\_bar\_code

功能：一个条形码的顺序解码。

### 2. decode\_2d\_bar\_code

功能：解码二维条形码数据。

### 3. discrete\_1d\_bar\_code

功能：从元素宽度创建一个离散条形码。

### 4. find\_1d\_bar\_code

功能：搜索一幅图像中的一个条形码。

### 5. find\_1d\_bar\_code\_region

功能：搜索一幅图像中的多种条形码。

#### 6. find\_1d\_bar\_code\_scanline

功能：搜索一幅图像中的一个条形码。

#### 7. find\_2d\_bar\_code

功能：搜索可能包括一个二维条形码的区域。

#### 8. gen\_1d\_bar\_code\_descry

功能：创建一个一维条形码的说明。

#### 9. gen\_1d\_bar\_code\_descr\_gen

功能：创建一个一维条形码的类属描述。

#### 10. gen\_2d\_bar\_code\_descry

功能：创建一个二维条形码的类属描述。

### 11. get\_1d\_bar\_code

功能：提取一个条形码中元素的宽度。

### 12. get\_1d\_bar\_code\_scanline

功能：提取一个条形码区域中元素的宽度。

### 13. get\_2d\_bar\_code\_

功能：提取一个条形码区域（“数据矩阵符号”）中数据元素（在 ECC200：“模块”中）的值。\_

### 14. get\_2d\_bar\_code\_pos

功能：提取一个条形码区域（“数据矩阵符号”）中数据元素（在 ECC200：“模块”中）的数值和它们在图像中的位置。

## Chapter\_18:Tuple

### 18.1 Arithmetic



### 1. tuple\_abs

功能：计算一个元组的绝对值。

### 2. tuple\_acos

功能：计算一个元组的反余弦。

### 3. tuple\_add

功能：两个元组相加。

### 4. tuple\_asin

功能：计算一个元组的反余弦。

### 5. tuple\_atan

功能：计算一个元组的反正切。

### 6. tuple\_atan2

功能：计算一个元组四个象限的反正切。

## 7. tuple\_ceil

功能：计算一个元组的上限函数。

## 8. tuple\_cos

功能：计算一个元组的余弦。

## 9. tuple\_cosh

功能：计算一个元组的双曲余弦。

## 10. tuple\_cumul

功能：计算一个元组的累计和。

## 11. tuple\_deg

功能：将一个元组从弧度转换为角度。

## 12. tuple\_div

功能：将两个元组相除。

## 13. tuple\_exp

功能：元组的指数运算。

## 14. tuple\_fabs\_

功能：计算一个元组（例如浮点数）的绝对值。

## 15. tuple\_floor

功能：计算一个元组的“地板函数”。

## 16. tuple\_fmod

功能：计算两个元组浮点数相除的余数。

## 17. tuple\_ldexp

功能：计算两个元组的返回长双精度指数函数。

#### 18. tuple\_log

功能：计算一个元组的自然对数。

#### 19. tuple\_log10

功能：计算一个元组底为 10 的对数。

#### 20. tuple\_max2

功能：计算两个元组的元素宽度的最大值。

#### 21. tuple\_min2

功能：计算两个元组的元素宽度的最小值。

#### 22. tuple\_mod

功能：计算两个元组整型数相除的余数。

### 23. tuple\_mult

功能：两个元组相乘。

### 24. tuple\_neg

功能：将一个元组取反。

### 25. tuple\_pow

功能：计算两个元组的幂函数。

### 26. tuple\_rad

功能：将一个元组从角度转换为弧度。

### 27. tuple\_sgn

功能：计算一个元组的正负。

### 28. tuple\_sin

功能：计算一个元组的正弦。

### 29. tuple\_sinh

功能：计算一个元组的双曲正弦。

### 30. tuple\_sqrt

功能：计算一个元组的平方根（二次方根）。

### 31. tuple\_sub

功能：两个元组相减。

### 32. tuple\_tan

功能：计算一个元组的正切。

### 33. tuple\_tanh

功能：计算一个元组的双曲正切。

## 18.2 Bit-Operations

### 1. tuple\_band

功能：计算两个元组的按位运算。

### 2. tuple\_bnot

功能：两个元组逐位取逻辑非。

### 3. tuple\_bor

功能：计算两个元组的按位运算。

### 4. tuple\_bxor

功能：两个元组逐位进行互斥逻辑或运算。

## 5. tuple\_lsh

功能：元组逐位左移。

## 6. tuple\_rsh

功能：元组逐位右移。

# 18.3 Comparison

## 1. tuple\_equal

功能：测试两个元组是否相同。

## 2. tuple\_greater

功能：测试一个元组是否大于另一个元组。

## 3. tuple\_greater\_equal



功能：测试一个元组是否大于等于另一个。

#### 4. tuple\_less

功能：测试一个元组是否小于另一个元组。

#### 5. tuple\_less\_equal

功能：测试一个元组是否小于等于另一个。

#### 6. tuple\_not\_equal

功能：测试两个元组是不是不等。

### 18.4 Conversion

#### 1. tuple\_chr

功能：根据 ASCII 码将整型元组转换为字符串。

## 2. tuple\_chrt

功能：根据 ASCII 码将整型元组转换为字符串。

## 3. tuple\_int

功能：讲一个元组转换为一个整型元组。

## 4. tuple\_is\_number

功能：检测一个字符串元组是否表示数字。

## 5. tuple\_number

功能：将一个字符串元组转换为一个数字元组。

## 6. tuple\_ord

功能：将长度为 1 的字符串的元组转换为它们相应的 ASCII 码元组。

## 7. tuple\_ord

功能：将一个字符串的元组转换为它们 ASCII 码的元组。

## 8. tuple\_real

功能：将一个元组转换为一个浮点数的元组。

## 9. tuple\_round

功能：将一个元组转换为一个整型数的元组。

## 10. tuple\_string

功能：将一个元组转换为一个字符串元组。

# 18.5 Creation

## 1. tuple\_concat

功能：合并两个元组为一个新的。

## 2. tuple\_gen\_const

功能：创建一个特殊长度的元组和初始化它的元素。

## 3. tuple\_rand

功能：返回任意值为 0 或 1 的元组。

## 18.6\_Element-Order

### 1. tuple\_inverse

功能：将一个元组反置（反转）。

### 2. tuple\_sort

功能：按照升序分类（排列）元组的元素。

### 3. tuple\_sort\_index

功能：将元组的元素分类并返回分类元组的目录。

## 18.7 Features

### 1. tuple\_deviation

功能：返回一个元组元素的标准差。

### 2. tuple\_length

功能：返回一个元组元素数目。

### 3. tuple\_max

功能：返回一个元组的最大元素。

### 4. tuple\_mean

功能：返回一定数量元组的平均值。

## 5. tuple\_median

功能：返回一个元组元素的中值。

## 6. tuple\_min

功能：返回一个元组的最小元素。

## 7. tuple\_sum

功能：返回一个元组所有元素的和。

# 18.8 Logical-Operations

## 1. tuple\_and

功能：两个元组的逻辑与。

## 2. tuple\_not

功能：两个元组的逻辑非。

### 3. tuple\_or

功能：两个元组的逻辑或。

### 4. tuple\_xor

功能：两个元组的逻辑互斥或。

## 18.9 Selection

### 1. tuple\_find

功能：返回一个元组所有出现的符号，同时位于另一个元组内。

### 2. tuple\_first\_n

功能：选取一个元组的第一个元素。

### 3. tuple\_last\_n

功能：选择从符号“n”开始到元组末尾的所有元素。

### 4. tuple\_remove

功能：从一个元组中移出元素。

### 5. tuple\_select

功能：选择一个元组中单一元素。

### 6. tuple\_select\_range

功能：选择一个元组中的一些元素。

### 7. tuple\_select\_rank

功能：选择一个元组中序号为 n 的元素。

### 8. tuple\_str\_bit\_select



功能：选择一个元组中单一符号或位。

## 9. tuple\_uniq

功能：丢弃元组中除成功归类的元素外的所有元素。

## 18.10 String-Operators

### 1. tuple\_environment

功能：读取一个或多个环境变量。

### 2. tuple\_regexp\_match

功能：利用公式提取子链。

### 3. tuple\_regexp\_replace

功能：用有规律的公式代替一个子链。

#### 4. tuple\_regexp\_select

功能：选择符合公式的元组元素。

#### 5. tuple\_regexp\_test

功能：测试一个字符串是否满足一个规则公式的要求。

#### 6. tuple\_split

功能：在预定义的独立字符间将字符串分离为子链。

#### 7. tuple\_str\_first\_n

功能：分割从第一个字符直到字符串元组外的位置“n”处。

#### 8. tuple\_str\_last\_n

功能：从字符串元组外位置“n”处开始分割所有的字符。

#### 9. tuple\_strchr

功能：前向搜索一个位于字符串元组内的字符。

#### 10. tuple\_strlen

功能：字符串元组中每个字符串的长度。

#### 11. tuple\_strchr

功能：后向搜索一个位于字符串元组内的字符。

#### 12. tuple\_strrstr

功能：后向搜索一个位于字符串元组内的字符串。

#### 13. tuple\_strstr

功能：前向搜索一个位于字符串元组内的字符串。

## 19.1 Access

### 1. get\_contour\_xld

功能：返回 XLD 轮廓(contour)的坐标。

### 2. get\_lines\_xld

功能：返回一个 XLD 多边形（polygon）数据。

### 3. get\_parallels\_xld

功能：返回一个 XLD 并行数据。

### 4. get\_polygon\_xld

功能：返回一个 XLD 多边形（polygon）数据。

## 19.2 Creation

### 1. gen\_contour\_nurbs\_xld

功能：将一个 NURBS 曲线转换为一个 XLD（密度？）轮廓(contour)。

### 2. gen\_contour\_polygon\_rounded\_xld

功能：根据一个多边形（polygon）（以元组形式给出）的圆形角点创建一个 XLD 轮廓(contour)。

### 3. gen\_contour\_polygon\_xld

功能：根据一个多边形（polygon）（以元组形式给出）创建一个 XLD 轮廓(contour)。

### 4. gen\_contour\_region\_xld

功能：根据区域创建 XLD 轮廓(contour)。

### 5. gen\_contours\_skeleton\_xld

功能：将框架转换为 XLD 轮廓(contour)。

## 6. gen\_cross\_contour\_xld

功能：根据每个输入点交叉的形状创键一个 XLD 轮廓(contour)。

## 7. gen\_ellipse\_contour\_xld

功能：根据相应的椭圆弧创建一个 XLD 轮廓(contour)。

## 8. gen\_parallel\_xld

功能：提取并行 XLD 多边形 (polygon)。

## 9. gen\_polygons\_xld

功能：根据多边形近似创建 XLD 轮廓(contour)。

## 10. gen\_rectangle2\_contour\_xld

功能：创建一个矩形 XLD 轮廓(contour)。

## 11. mod\_parallel\_xld

功能：提取一个包括同质区域的并行 XLD 多边形（polygon）。

### 19.3 Features

#### 1. area\_center\_points\_xld

功能：被看做点云的轮廓(contour)和多边形（polygon）的面积和重心。

#### 2. area\_center\_xld

功能：轮廓(contour)和多边形（polygon）的面积和重心。

#### 3. circularity\_xld

功能：影响轮廓(contour)或多边形（polygon）圆度（与圆相近的程度）的形状系数。

#### 4. compactness\_xld

功能：影响轮廓(contour)或多边形 (polygon) 致密性的形状系数。

#### 5. contour\_point\_num\_xld

功能：返回一个 XLD 轮廓(contour)中点的数目。

#### 6. convexity\_xld

功能：影响轮廓(contour)或多边形 (polygon) 凹凸性的形状系数。

#### 7. diameter\_xld

功能：两个轮廓(contour)或多边形 (polygon) 点间的最大距离。

#### 8. dist\_ellipse\_contour\_points\_xld

功能：计算所有轮廓(contour)内的点到一个椭圆的距离。

#### 9. dist\_ellipse\_contour\_xld

功能：轮廓到一个椭圆的距离。



#### 10. dist\_rectangle2\_contour\_points\_xld\_

功能：计算所有轮廓(contour)内的点到一个矩形的距离。

#### 11. eccentricity\_points\_xld\_

功能：被看做点云的轮廓(contour)或多变形（polygon）的 Anisometry。

#### 12. eccentricity\_xld

功能：源自轮廓(contour)或多边形（polygon）的椭圆参数的形状系数。

#### 13. elliptic\_axis\_points\_xld

功能：被看做点云的轮廓(contour)或多边形（polygon）的等价椭圆参数。

#### 14. elliptic\_axis\_xld

功能：轮廓(contour)或多变形（polygon）的等价椭圆参数。

#### 15. fit\_circle\_contour\_xld

功能：根据圆周近似获取 XLD 轮廓(contour)。

#### 16. fit\_ellipse\_contour\_xld

功能：根据椭圆或椭圆弧近似获取 XLD 轮廓(contour)。

#### 17. fit\_line\_contour\_xld

功能：根据分割线近似获取 XLD 轮廓(contour)。

#### 18. fit\_rectangle2\_contour\_xld

功能：用矩形来匹配 XLD 轮廓(contour)。

#### 19. get\_contour\_angle\_xld

功能：为每个轮廓(contour)点计算一个 XLD 轮廓(contour)方向。

#### 20. get\_contour\_attrib\_xld

功能：返回一个 XLD 轮廓(contour)的点的特征值。

#### 21. get\_contour\_global\_attrib\_xld

功能：返回一个 XLD 轮廓(contour)的全局特征值。

#### 22. get\_regress\_params\_xld

功能：返回 XLD 轮廓(contour)参数。

#### 23. info\_parallel\_xld

功能：返回被 XLD 多边形 (polygon) 包围的区域的灰度值的信息。

#### 24. length\_xld

功能：轮廓(contour)或多边形 (polygon) 的长度。

#### 25. local\_max\_contours\_xld

功能：选择局域最大灰度值的 XLD 轮廓(contour)。

## 26. max\_parallel\_xld

功能：合并具有相同多边形（**polygon**）的重建 XLD 并行。

## 27. moments\_any\_points\_xld

功能：被看做点云的轮廓(**contour**)或多变形（**polygon**）的任意几何时刻（**moments**）。

## 28. moments\_any\_xld

功能：轮廓(**contour**)或多变形（**polygon**）的任意集合时刻（**moments**）。

## 29. moments\_points\_xld

功能：被看做点云的轮廓(**contour**)或多变形（**polygon**）的几何时刻（**moments**） M20, \_M02, 和 \_M11。

## 30. moments\_xld

功能：轮廓(contour)或多变形的几何时刻（moments）  
M20,\_M02,\_and\_M11。

### 31. orientation\_points\_xld

功能：被看做点云的轮廓(contour)或多变形（polygon）的方向。

### 32. orientation\_xld

功能：轮廓(contour)或多变形（polygon）的方向。

### 33. query\_contour\_attribs\_xld

功能：返回一个 XLD 轮廓(contour)定义的属性的名字。

### 34. query\_contour\_global\_attribs\_xld

功能：返回一个 XLD 轮廓(contour)定义的全局属性的名字。

### 35. select\_contours\_xld

功能：根据一些特征选择 XLD 轮廓(contour)。

### 36. select\_shape\_xld

功能：根据形状特征选择轮廓(contour)或多边形 (polygon)。

### 37. select\_xld\_point

功能：选择包括给定点在内的所有的轮廓(contour)或多边形 (polygon)。

### 38. smallest\_circle\_xld

功能：轮廓(contour)或多边形 (polygon) 的最小封闭圆。

### 39. smallest\_rectangle1\_xld

功能：平行与轮廓(contour)或多边形 (polygon) 的坐标轴的封闭矩形。

### 40. smallest\_rectangle2\_xld

功能：轮廓(contour)或多边形 (polygon) 任意方向的最小封闭矩形。

#### 41. test\_self\_intersection\_xld

功能：测试轮廓(contour)或多边形 (polygon) 自身相交性。

#### 42. test\_xld\_point

功能：测试一个或多个包括给定点在内的轮廓(contour)或多边形 (polygon)。

### 19.4 Geometric-Transformations

#### 1. affine\_trans\_contour\_xld

功能：对 XLD 轮廓(contour)进行一个任意二维仿射变换。

#### 2. affine\_trans\_polygon\_xld

功能：对 XLD 多边形 (polygon) 进行一个任意仿射变换。

### 3. gen\_parallel\_contour\_xld

功能：计算一个 XLD 轮廓(contour)的平行轮廓(contour)。

### 4. polar\_trans\_contour\_xld

功能：将一个环状弧中的轮廓(contour)转换为极坐标形式。

### 5. polar\_trans\_contour\_xld\_inv

功能：将极坐标下的轮廓(contour)转换为笛卡尔坐标下的形式。

### 6. projective\_trans\_ontour\_xld

功能：对一个 XLD 轮廓(contour)进行射影变换。

## 19.5 Sets

### 1. difference\_closed\_contours\_xld



功能： 闭合轮廓(contour)的差异。

## 2. difference\_closed\_polygons\_xld\_

功能： 闭合多边形 (polygon) 的差异。

## 3. intersection\_closed\_contours\_xld

功能： 闭合轮廓(contour)的交集。

## 4. intersection\_closed\_polygons\_xld

功能： 闭合多边形 (polygon) 的交集。

## 5. symm\_difference\_closed\_contours\_xld

功能： 闭合轮廓(contour)的对称差异。

## 6. symm\_difference\_closed\_polygons\_xld

功能： 闭合多边形 (polygon) 的对称差异。

## 7. union2\_closed\_contours\_xld

功能： 闭合轮廓(contour)的并集。

## 8. union2\_closed\_polygons\_xld

功能： 闭合多边形（polygon）的并集。

# 19.6 Transformation

## 1. add\_noise\_white\_contour\_xld

功能： 向 XLD 轮廓(contour)中加入噪声。

## 2. clip\_contours\_xld

功能： 修剪一个 XLD 轮廓(contour)。

## 3. close\_contours\_xld

功能：关闭一个 XLD 轮廓(contour)。

#### 4. combine\_roads\_xld

功能：合并两个等级分辨率中的路（road）。

#### 5. crop\_contours\_xld

功能：切割一个 XLD 轮廓(contour)。

#### 6. merge\_cont\_line\_scan\_xld

功能：合并连续线扫描图像中的 XLD 轮廓(contour)。

#### 7. regress\_contours\_xld

功能：计算一个 XLD 轮廓(contour)回归线的参数。

#### 8. segment\_contours\_xld

功能：将 XLD 轮廓(contour)分割为分割线和圆周或椭圆弧。

## 9. shape\_trans\_xld

功能：改变轮廓(contour)或多边形 (polygon) 的形状。

## 10. smooth\_contours\_xld

功能：XLD 轮廓(contour)的平滑。

## 11. sort\_contours\_xld

功能：根据相关位置分类轮廓(contour)。

## 12. split\_contours\_xld

功能：在主要点分割 XLD 轮廓(contour)。

## 13. union\_adjacent\_contours\_xld

功能：合并终点连接在一起的轮廓(contour)。

## 14. union\_cocircular\_contours\_xld

功能：合并属于同一个圆周的轮廓(contour)。

#### 15. union\_collinear\_contours\_ext\_xld

功能：合并位于同一条直线上的轮廓(contour)（由附加函数操作）。

#### 16. union\_collinear\_contours\_xld

功能：合并位于同一条直线上的轮廓(contour)。

#### 17. union\_straight\_contours\_histo\_xld

功能：合并到给定线有相似距离的相邻直线轮廓(contour)。

#### 18. union\_straight\_contours\_xld

功能：合并具有相似方向的相邻直线轮廓(contour)。