第5章 XLD

XLD是eXtended Line Description的缩写，是HALCON中表示轮廓和多边形的数据结构。XLD是由直线连接起来的一系列二维控制点，与图像点和区域点用像素坐标表示的不一样，XLD中的二维点是以亚像素坐标表示的，亚像素坐标是一种将像素坐标细分从而得到更高分辨率的坐标，具有比像素坐标更高的精度，因此，XLD常被理解为亚像素轮廓，是HALCON中非常重要的数据结构之一。

本章主要介绍XLD的获取、XLD的特征和XLD的处理，涉及的知识点如下。

- XLD的获取：亚像素级边缘提取、亚像素阈值等。
- XLD的特征：基础特征、形状特征、点集特征、几何矩特征和特征筛选等。
- XLD的处理：创建、分割、合并、拟合等。

视频讲解

5.1 XLD的获取

因为XLD具有亚像素精度，因此使用XLD来测量是实现高精度测量的有效方法，用于测量的XLD通常由测量图像通过边缘提取获取，HALCON支持像素级精度的边缘提取和亚像素级精度的边缘提取，如果像素级精度已经可以满足测量需求，则无须使用亚像素级精度的边缘提取，因为后者的运算需要消耗更多的时间。像素级的边缘提取不直接获取XLD，而是获取边缘的轮廓区域，该轮廓区域可以转换为XLD。亚像素级的边缘提取则可以直接获取XLD。

▶ 5.1.1 亚像素级边缘提取

1. 边缘提取方法

亚像素级边缘提取最常用的算子是edges_sub_pix()。edges_sub_pix()是针对单通道灰度图像的算子，其内部实现了多种边缘提取算法，在边缘检测算法参数中选择需要的方法即可实现相应的边缘提取算法，实际使用时可以根据不同算法的测试效果选择合适的边缘提取算法，该算子的原型如下。

```
edges_sub_pix(Image : Edges : Filter, Alpha, Low, High :)
```

- Image：输入的单通道灰度图像。
- Edges：输出的XLD轮廓。
- Filter：边缘检测的算法。
- Alpha：边缘检测的平滑参数。
- Low、High：滞后阈值操作的低阈值和高阈值。

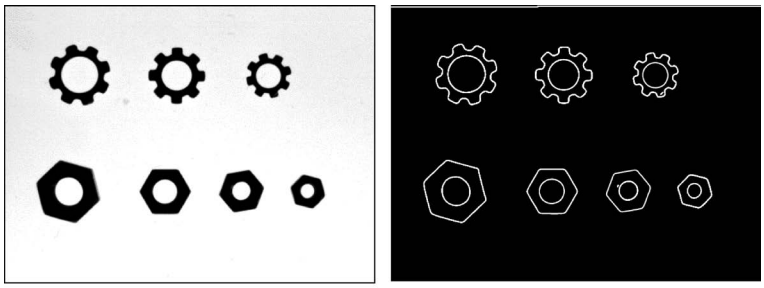
edges_sub_pix()实现的边缘检测方法都可以在参数Filter中选择，Filter包含使用递归实现的滤波器(根据Deriche、Lanser、Shen方法及其变体)和Canny提出的使用传统高斯导数的滤波器，默认值是canny。Alpha参数可以为边缘提取的滤波器设置平滑的程度(除了sobel

相关的方法),对于使用递归实现的滤波器,Alpha 越大平滑越小,而 Canny 滤波器方法则与之相反,Alpha 越大平滑越大。平滑越大,滤波器的抗噪性越好,但是也降低了检测小细节的能力。edges_sub_pix()算子通过使用类似于滞后阈值操作的算法将边缘点链接到边缘,该操作通过参数 Low 和 High 设置实现,值低于 Low 的像素点则认为不是边缘,值高于 High 的像素点则认为是边缘,而介于 Low 和 High 之间的像素点则根据相邻像素点是否是边缘进行判断。

【例 5-1】 edges_sub_pix()实现亚像素级边缘提取示例。

```
* 关闭窗口
dev_close_window()
* 读取图像
read_image(Image, 'rings_and_nuts')
* 打开适应图像大小的窗口
dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
* 对图像进行亚像素边缘提取
edges_sub_pix(Image, Edges, 'canny', 1, 20, 40)
* 显示
dev_display(Image)
dev_display(Edges)
```

实例中使用默认的滤波器检测方法 canny,其检测的边缘 XLD 效果如图 5.1 所示,图 5.1(a)为检测原图,图 5.1(b)为检测出来的边缘轮廓。



(a) 原图　　(b) 边缘轮廓XLD图

图 5.1　edges_sub_pix()边缘检测效果

亚像素边缘提取除了使用 edges_sub_pix(),还有专门针对多通道彩色图像的 edges_color_sub_pix()算子,该算子与 edges_sub_pix()十分相似,此处不再详细解释。

2. 线条提取方法

除了边缘提取方法,HALCON 还可以通过线条提取实现亚像素精度 XLD 获取,线条提取获取的是曲线结构的线条,其最常用的算子是 lines_gauss()。lines_gauss()实现的也是单通道灰度图像的线条提取,与之对应的还有实现多通道彩色图像线条提取的算子 lines_color()。lines_gauss()算子的原型如下。

```
* 单通道图像的亚像素线条提取
lines_gauss(Image : Lines : Sigma, Low, High, LightDark, ExtractWidth, LineModel, CompleteJunctions :)
```

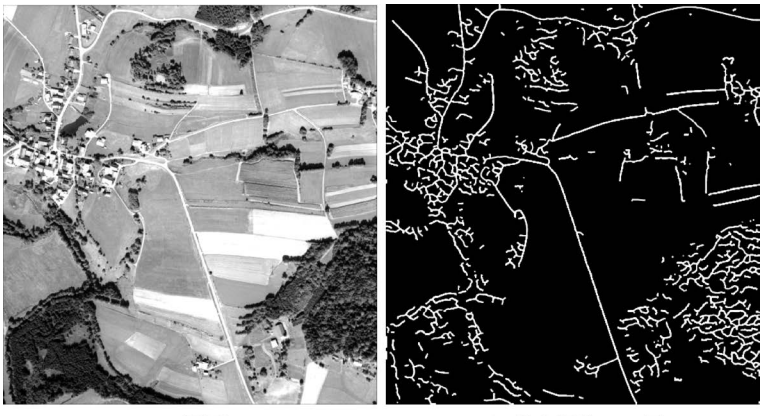
- Image:输入的单通道灰度图像。
- Lines:提取的线条 XLD。
- Sigma:要应用的高斯平滑量。
- Low、High:滞后阈值操作的低阈值和高阈值。
- LightDark:要提取暗的线条还是亮的线条。

- ExtractWidth：是否需要提取线宽。
- LineModel：用来调整线条位置的线条模型。
- CompleteJunctions：在线条不连续的地方是否添加连接。

lines_gauss()提取线条的方法是通过使用高斯平滑核的偏导数来确定图像中每个点在 x 和 y 方向上的二次多项式的参数来完成的，这是一个比较复杂的过程，相关参数选择可以做以下简单的参考：Sigma 参数决定了平滑量的大小，值越大平滑越大，但是较大的平滑可能会导致线条的定位产生偏差。Low 和 High 参数的原理类似于 edges_sub_pix()中相同的参数，只不过对比的对象不是像素点，而是二阶偏导数，选择的 Sigma 越大，二阶偏导数就越小，因此应该选择相应较小的 High 和 Low 值。LineModel 参数提供了三种不同的线条模型，用于调整线的位置和宽度，分别为条形线(LineModel＝'bar-shaped')、抛物线(LineModel＝'parabolic')和高斯线(LineModel＝'gaussian')，条形线可以适用大多数情况，背光场景可以考虑另外两种线型，抛物线适用于线条比较清晰的情况，高斯线适用于线条不是特别清晰的情况。这些线条模型的选择只有在 ExtractWidth 为'true'时才有效。

【例 5-2】 lines_gauss()实现亚像素级线条提取的示例，效果如图 5.2 所示。

```
* 关闭窗口
dev_close_window()
* 读取图像
read_image(Image, 'mreut4_3')
* 打开适应图像大小的窗口
dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
* 对图像进行亚像素线条提取
lines_gauss(Image, Lines, 1.5, 3, 8, 'light', 'true', 'bar-shaped', 'true')
* 显示
dev_display(Image)
dev_display(Lines)
```



(a) 原图　　(b) 线条提取XLD图

图 5.2　lines_gauss()线条提取效果

▶ 5.1.2　亚像素阈值

HALCON 提供了一个简单的通过阈值分割实现 XLD 提取的方法，该方法通过算子 threshold_sub_pix()实现，该算子通过单个阈值，将图像的灰度值分割为高于阈值和低于阈值两部分，边界线即为提取的 XLD，该算子适用于在照明条件稳定的情况下，作为快速边缘提取

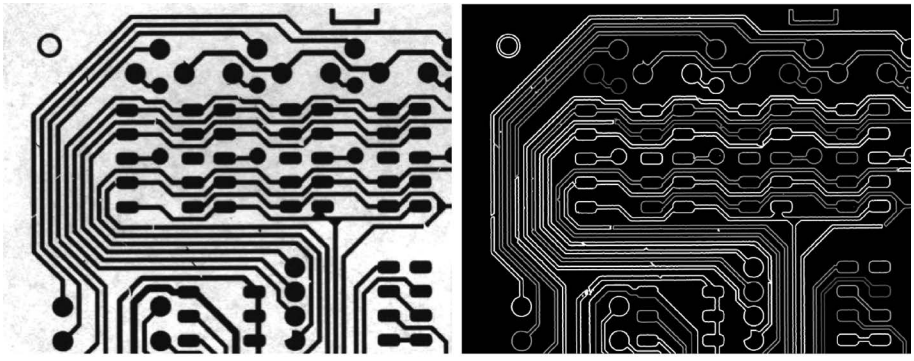
的替代方案，算子原型如下。

```
* 单通道图像的亚像素阈值边界提取
threshold_sub_pix(Image : Border : Threshold :)
```

- Image：输入的单通道灰度图像。
- Border：提取的阈值边界 XLD。
- Threshold：边界的阈值。

【例 5-3】 threshold_sub_pix()实现亚像素级阈值边界提取的示例，效果如图 5.3 所示。

```
* 关闭窗口
dev_close_window()
* 读取图像
read_image(Image, 'pcb')
* 打开适应图像大小的窗口
dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
* 对图像进行亚像素阈值边界提取
threshold_sub_pix(Image, Border, 50)
* 显示
dev_display(Image)
dev_display(Border)
```



(a) 原图　　(b) 边界提取XLD图

图 5.3　threshold_sub_pix()边界提取效果

5.2　XLD 的特征

视频讲解

XLD 的特征是 XLD 具有的基本内在属性，其包含基础特征、形状特征、点集特征和几何矩特征，可以计算这些特征值，作为测量结果输出，也可以根据这些特征值对 XLD 进行筛选，从而提取项目需要的 XLD 部分。

▶ 5.2.1　特征类型

HALCON 内置了检测 XLD 特征的工具，检测方法如图 5.4 中①～④步骤所示。第①步先选择一个 XLD 轮廓，单击选中该轮廓；第②步单击图标所示的“打开特征检测”工具；第③步在打开的特征检测工具中勾选 xld 选项，xld 选项下有 basic、shape、points 和 moments 四组特征，分别对应基础特征、形状特征、点集特征和几何矩特征，所有勾选的特征项都会在特征检测工具右侧的数值栏中显示；根据需要的特征，第④步可以右键单击数值栏中的特征项，选择“插入算子”项将代码插入程序中，便可在程序运行中获取需要的特征。但是这里需要注意，以

这种方法插入特征算子获取的是整个 XLD 中所有轮廓的相应特征，而并非第①步选中的那个轮廓的特征，因此插入特征算子的方法仅适用于快速找到对应的特征检测算子，在这之前必须先通过特征筛选或排序等方式定位到需要检测的 XLD 轮廓。

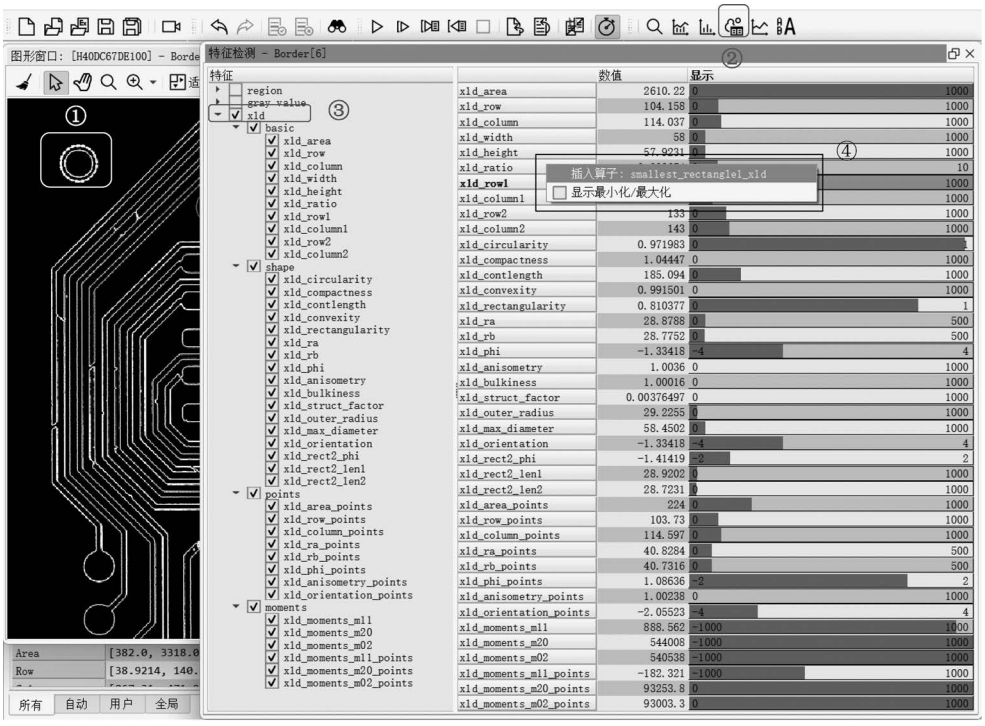


图 5.4　XLD 特征检测

1. 基础特征

XLD 的基础特征包含轮廓面积、质心坐标、宽高及宽高比和轴向外接矩形，相应的特征计算算子如下。

```
* XLD 面积和质心坐标的计算
area_center_xld(XLD : : : Area, Row, Column, PointOrder)
```

- XLD：输入的 XLD。
- Area：XLD 闭合区域的面积。
- Row、Column：XLD 的质心坐标。
- PointOrder：沿边界的点顺序（正/负）。

```
* XLD 宽高和宽高比的计算
height_width_ratio_xld(XLD : : : Height, Width, Ratio)
```

- XLD：输入的 XLD。
- Height、Width：XLD 的高和宽。
- Ratio：XLD 的宽高比。

```
* XLD 轴向外接矩形的计算，返回左上角和右下角的坐标
smallest_rectangle1_xld(XLD : : : Row1, Column1, Row2, Column2)
```

- XLD：输入的 XLD。
- Row1，Column1：XLD 轴向外接矩形左上角的行列坐标。

- Row2，Column2：XLD 轴向外接矩形右下角的行列坐标。

2. 形状特征

XLD 的形状特征包含轮廓圆度、紧密度、长度、凸度、矩形度、等效椭圆、等效椭圆特征、外接圆、最大点矩、方向、最小有向外接矩形，相应的特征计算算子如下：

```
* XLD 轮廓圆度的计算
circularity_xld(XLD : : : Circularity)
```

- XLD：输入的 XLD。
- Circularity：XLD 的轮廓圆度。

```
* XLD 紧密度的计算
compactness_xld(XLD : : : Compactness)
```

- XLD：输入的 XLD。
- Compactness：XLD 的紧密度。

```
* XLD 轮廓的长度计算
length_xld(XLD : : : Length)
```

- XLD：输入的 XLD。
- Length：XLD 轮廓的长度。

```
* XLD 凸度的计算
convexity_xld(XLD : : : Convexity)
```

- XLD：输入的 XLD。
- Convexity：XLD 的凸度。

```
* XLD 矩形度的计算
rectangularity_xld(XLD : : : Rectangularity)
```

- XLD：输入的 XLD。
- Rectangularity：XLD 的矩形度。

```
* XLD 等效椭圆的计算
elliptic_axis_xld(XLD : : : Ra, Rb, Phi)
```

- XLD：输入的 XLD。
- Ra、Rb：XLD 等效椭圆的长半轴和短半轴。
- Phi：长半轴与短半轴的夹角。

```
* XLD 等效椭圆特征的计算
eccentricity_xld(XLD : : : Anisometry, Bulkiness, StructureFactor)
```

- XLD：输入的 XLD。
- Anisometry：XLD 等效椭圆的长半轴和短半轴之比。
- Bulkiness：XLD 等效椭圆面积与轮廓面积之比。
- StructureFactor：XLD 的结构因子，StructureFactor = Anisometry * Bulkiness − 1。

```
* XLD 最小外接圆的计算
smallest_circle_xld(XLD : : : Row, Column, Radius)
```

- XLD：输入的 XLD。
- Row、Column：XLD 最小外接圆的圆心坐标。
- Radius：最小外接圆的半径。

```
* XLD 最大点距的计算
diameter_xld(XLD : : : Row1, Column1, Row2, Column2, Diameter)
```

- XLD：输入的 XLD。
- Row1、Column1：XLD 第一个极值点坐标。
- Row2、Column2：XLD 第二个极值点坐标。
- Diameter：两个极值点的距离。

```
* XLD 方向的计算
orientation_xld(XLD : : : Phi)
```

- XLD：输入的 XLD。
- Phi：XLD 的弧度方向。

```
* XLD 最小有向外接矩形的计算
smallest_rectangle2_xld(XLD : : : Row, Column, Phi, Length1, Length2)
```

- XLD：输入的 XLD。
- Row、Column：XLD 最小有向外接矩形的中心坐标。
- Phi：XLD 最小有向外接矩形的方向。
- Length1、Length2：XLD 最小有向外接矩形的半长和半宽。

3. 点集特征

由于 XLD 是由直线连接起来的一系列二维控制点，因此这些点集也有相关的特征，XLD 的点集特征包含轮廓点集的面积及质心、等效椭圆及半轴比和方向，相应的特征计算算子如下。

```
* XLD 中点集的面积及质心坐标的计算
area_center_points_xld(XLD : : : Area, Row, Column)
```

- XLD：输入的 XLD。
- Area：XLD 中点集的面积，面积为点的数量。
- Row、Column：XLD 中点集的质心坐标。

```
* XLD 中点集的等效椭圆的计算
elliptic_axis_points_xld(XLD : : : Ra, Rb, Phi)
```

- XLD：输入的 XLD。
- Ra、Rb：XLD 中点集的等效椭圆的长半轴和短半轴。
- Phi：XLD 中的点集的等效椭圆的长半轴与短半轴的夹角。

```
* XLD 中点集的等效椭圆的长短半轴之比计算
eccentricity_points_xld(XLD : : : Anisometry)
```

- XLD：输入的 XLD。
- Anisometry：XLD 中点集的等效椭圆的长短半轴之比。

```
* XLD 中点集的方向计算
orientation_points_xld(XLD : : : Phi)
```

- XLD：输入的 XLD。
- Phi：XLD 中点集的方向。

4. 几何矩特征

XLD 的几何矩包含两种类型，一种是 XLD 包围区域的几何矩，另一种是 XLD 点集的几

何矩，相应的特征计算算子如下。

```
* XLD 包围区域的几何矩计算
moments_xld(XLD : : : M11, M20, M02)
```

- XLD：输入的 XLD。
- M11、M20、M02：XLD 包围区域的 M11、M20、M02 矩。
- 计算公式为 $M_{ij}=\sum_{(r,c)\in R}(r_0-r)^i(c_0-c)^j$，$(r_0,c_0)$ 为区域质心。

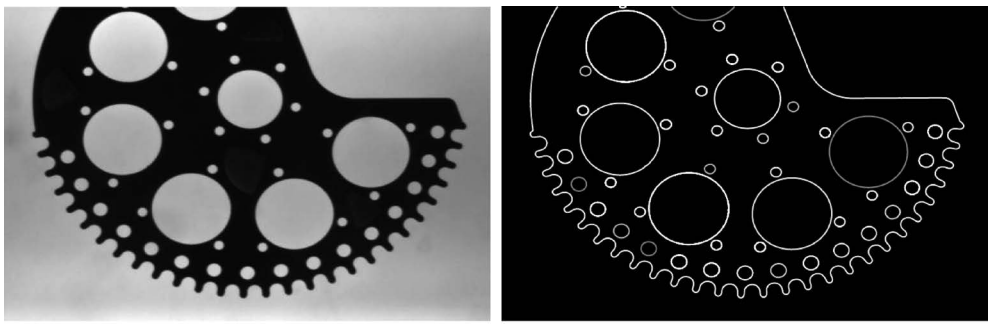
```
* XLD 点集的几何矩计算
moments_points_xld(XLD : : : M11, M20, M02)
```

- XLD：输入的 XLD。
- M11、M20、M02：XLD 点集的 M11、M20、M02 矩。
- 计算公式为 $M_{pq}=\sum_{i=0}^{n-1}(r_i-\bar{r})^p(c_i-\bar{c})^q$，$(\bar{r},\bar{c})$ 为点集质心。

▶ 5.2.2 特征筛选

XLD 的特征筛选是根据 XLD 的特征值，从 XLD 中提取出项目需要的那部分轮廓，HALCON 提供了快捷的特征筛选工具特征直方图，其使用方法与区域的特征筛选方法一致，核心是使用算子 select_shape_xld()进行筛选。下面以一个实例介绍 XLD 特征筛选的使用方法。

第一步：读取检测图片，并提取检测图片的 XLD 轮廓，如图 5.5 所示，目标是通过特征筛选提取检测 XLD 中的圆孔。



(a) 检测图片　　(b) 检测图片XLD轮廓

图 5.5　检测图片及其 XLD 轮廓

第二步：特征筛选。打开特征直方图，如图 5.6 中①图标位置的工具；然后在②位置的特征中选择需要的 XLD 特征，这里需要提取圆孔，所以可以选择 XLD 的圆度特征；接着在③的位置通过拖动限制范围的标线选择圆度的范围，可以根据图像中 XLD 的颜色变化观察是否选中需要的轮廓；最后单击④处的“插入代码”按钮即可将筛选结果的代码插入自己的程序中。

第三步：进一步筛选结果。通过圆度筛选的结果如图 5.7(a)所示，这时如果想进一步筛选其他轮廓，可以在第二步的图 5.6 的②位置添加一个特征属性，特征之间支持与、或两种运算；也可以继续打开特征直方图对 XLD 进行筛选。本例继续提取环形排列的圆孔，环形排列的圆孔的面积基本一致，因此可以通过 XLD 的面积特征，将环形排列的圆孔轮廓提取出来，其效果如图 5.7(b)所示。

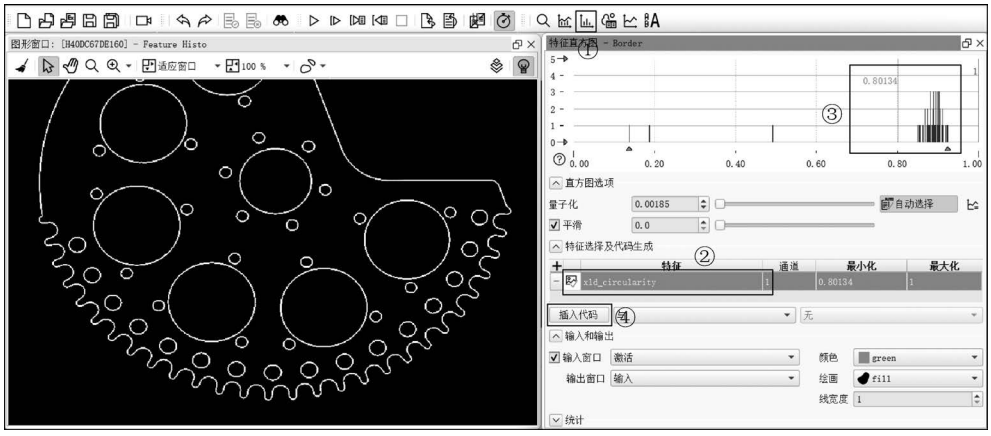
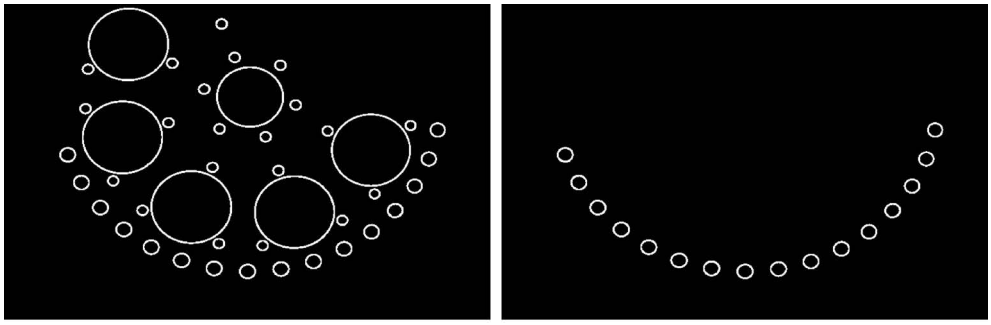


图 5.6　XLD 特征筛选



(a) 圆度筛选结果　　(b) 面积筛选结果

图 5.7　特征筛选结果

【例 5-4】 XLD 特征筛选示例。

```
* 关闭窗口
dev_close_window()
* 读取图像
read_image(Image, 'tooth_rim')
* 打开适应图像大小的窗口
dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
* 对图像进行亚像素阈值提取
threshold_sub_pix(Image,Border,80)
* 用圆度特征筛选圆孔
select_shape_xld(Border, SelectedXLD, 'circularity', 'and', 0.68629, 0.99031)
* 用面积特征筛选环形排列的圆轮廓
select_shape_xld(SelectedXLD, SelectedXLD1, 'area', 'and', 260.8, 1661.7)
* 显示
dev_display(Image)
dev_display(Border)
dev_display(SelectedXLD)
dev_display(SelectedXLD1)
```

视频讲解

5.3　XLD 的处理

使用 XLD 进行测量等任务通常需要对 XLD 进行一系列处理，才能得到需要的结果，这些处理过程通常包含创建、分割、合并和拟合等处理方法，根据测量需求的不同，这些处理方法应

该灵活搭配使用才能获得理想的效果。

▶ 5.3.1 创建

XLD处理的创建与5.1节XLD的获取内容有所不同，XLD的获取是通过提取图像的亚像素边缘得到XLD，而XLD处理的创建是通过已有的轮廓或多边形信息生成XLD，如给出圆心和半径就可以创建圆形轮廓的XLD，这种方式创建的XLD通常用于显示对比效果或为了方便程序的处理，以下是一些常用的创建XLD的算子。

```
*创建圆或者圆弧的XLD
gen_circle_contour_xld(: ContCircle : Row, Column, Radius, StartPhi, EndPhi, PointOrder, Resolution :)
*根据数组给出的多边形创建具有圆角的XLD
gen_contour_polygon_rounded_xld(: Contour : Row, Col, Radius, SamplingInterval :)
*根据数组给出的多边形创建XLD
gen_contour_polygon_xld(: Contour : Row, Col :)
*根据区域创建XLD
gen_contour_region_xld(Regions : Contours : Mode :)
*创建椭圆弧的XLD
gen_ellipse_contour_xld(: ContEllipse : Row, Column, Phi, Radius1, Radius2, StartPhi, EndPhi,
PointOrder, Resolution :)
*根据多边形近似创建XLD
gen_polygons_xld(Contours : Polygons : Type, Alpha :)
*创建矩形的XLD
gen_rectangle2_contour_xld(: Rectangle : Row, Column, Phi, Length1, Length2 :)
*从骨架创建XLD
gen_contours_skeleton_xld(Skeleton : Contours : Length, Mode :)
```

▶ 5.3.2 分割

分割是实现提取既要轮廓的重要方法，HALCON提供了用于分割的算子segment_contours_xld()，该算子可以将已有的轮廓分割为线和线、线和圆、线和椭圆，算子原型如下。

```
*将XLD轮廓分割成线段和圆弧或椭圆弧
segment_contours_xld(Contours : ContoursSplit : Mode, SmoothCont, MaxLineDist1, MaxLineDist2 :)
```

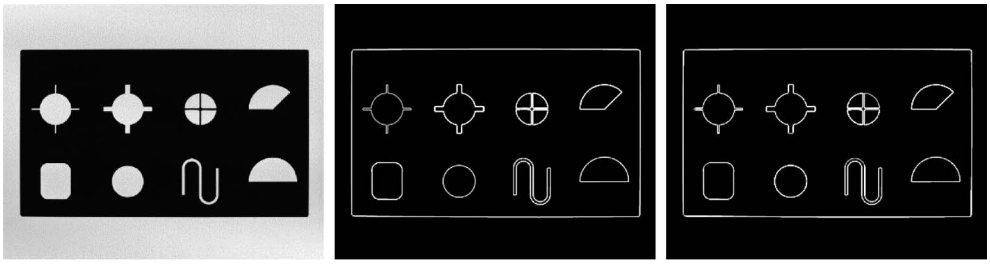
- Contours：输入的XLD轮廓。
- ContoursSplit：被分割后的XLD轮廓。
- Mode：分割XLD轮廓的模型。
- SmoothCont：平滑轮廓的点数。
- MaxLineDist1、MaxLineDist2：轮廓与近似线之间第一次迭代和第二次迭代的最大距离。

以下是使用线和圆的模型对XLD分割的用法实例，分割之前XLD各段轮廓都是相连的，分割完之后轮廓被分割为线段和圆弧(彩色原图见文前彩插)，效果如图5.8所示。

【例5-5】 分割XLD示例。

```
*关闭窗口
dev_close_window()
*读取图像
read_image(Image, 'circle_plate')
*打开适应图像大小的窗口
dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
*对图像进行亚像素阈值提取
threshold_sub_pix(Image,Border,80)
```

```
* 使用线和圆模型进行分割
segment_contours_xld(Border, ContoursSplit, 'lines_circles', 5, 4, 2)
* 显示
dev_display(Image)
dev_display(Border)
dev_display(ContoursSplit)
```



(a) 原图　　(b) 未分割XLD　　(c) 已分割XLD

图 5.8　XLD 分割示例（见彩插）

▶ 5.3.3　合并

XLD 处理中对于一些相近的或者同属一个形状，但是却断开的轮廓，HALCON 提供了将它们合并起来的方法。常用的合并方法是合并相邻的 XLD、合并共线的 XLD 和合并共圆的 XLD，相关算子原型如下。

```
* 合并相邻的 XLD
union_adjacent_contours_xld(Contours : UnionContours : MaxDistAbs, MaxDistRel, Mode :)
```

- Contours：输入的 XLD 轮廓。
- UnionContours：输出的 XLD 轮廓。
- MaxDistAbs：XLD 轮廓端点的最大距离。
- MaxDistRel：XLD 轮廓端点的最大距离与较长端轮廓线长度的比值。
- Mode：对轮廓附加属性的处理模型，是接收还是忽略。

```
* 合并共线的 XLD
union_collinear_contours_xld(Contours : UnionContours : MaxDistAbs, MaxDistRel, MaxShift, MaxAngle,
Mode :)
```

- Contours：输入的 XLD 轮廓。
- UnionContours：输出的 XLD 轮廓。
- MaxDistAbs：XLD 轮廓端点的最大距离。
- MaxDistRel：XLD 轮廓端点的最大距离与较长端轮廓线长度的比值。
- MaxShift：第二个轮廓到参考轮廓的回归线的最大距离。
- MaxAngle：两个轮廓的回归线之间的最大角度。
- Mode：对轮廓附加属性的处理模型，是接收还是忽略。

```
* 合并共圆的 XLD
union_cocircular_contours_xld(Contours : UnionContours : MaxArcAngleDiff, MaxArcOverlap,
MaxTangentAngle, MaxDist, MaxRadiusDiff, MaxCenterDist, MergeSmallContours, Iterations :)
```

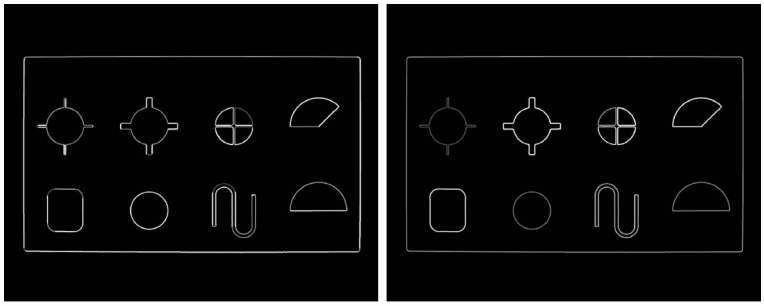
- Contours：输入的 XLD 轮廓。
- UnionContours：输出的 XLD 轮廓。
- MaxArcAngleDiff：两个圆弧的最大角距。

- MaxArcOverlap：两个圆弧的最大重叠。
- MaxTangentAngle：连接线与圆弧切线之间的最大夹角。
- MaxDist：两个圆弧间距的最大长度。
- MaxRadiusDiff：两个圆弧对应的圆的最大半径差。
- MaxCenterDist：两个圆弧对应的圆的最大中心距离。
- MergeSmallContours：确定没有拟合圆的小轮廓是否也应该合并。
- Iterations：迭代的次数。

下面以5.3.2节分割的实例为基础，在分割后的XLD轮廓中，使用合并相邻XLD的方法，将分割后的XLD再次合并为相连的XLD(彩色原图见文前彩插)，如图5.9(b)所示，详细合并过程如下。

【例5-6】 合并XLD示例。

```
*关闭窗口
dev_close_window()
*读取图像
read_image(Image, 'circle_plate')
*打开适应图像大小的窗口
dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
*对图像进行亚像素阈值提取
threshold_sub_pix(Image,Border,80)
*使用线和圆模型进行分割
segment_contours_xld(Border, ContoursSplit, 'lines_circles', 5, 4, 2)
*合并相邻XLD
union_adjacent_contours_xld(ContoursSplit, UnionContours, 10, 1, 'attr_keep')
*显示
dev_display(Image)
dev_display(Border)
dev_display(ContoursSplit)
dev_display(UnionContours)
```



(a) 分割后的XLD　　(b) 合并后的XLD

图5.9　合并相邻XLD(见彩插)

▶ 5.3.4　拟合

拟合是将不规则XLD转换为规则XLD的重要方法，HALCON对XLD拟合的方法主要提供了线拟合、圆拟合、椭圆拟合和矩形拟合，以下为主要拟合方法的算子原型。

```
*拟合直线XLD(直线方程为Y×Nr+C×Nc-Dist=0)
fit_line_contour_xld(Contours : : Algorithm, MaxNumPoints, ClippingEndPoints, Iterations,
ClippingFactor : RowBegin, ColBegin, RowEnd, ColEnd, Nr, Nc, Dist)
```

- Contours：输入的 XLD 轮廓。
- Algorithm：直线拟合的算法。
- MaxNumPoints：用于计算的最大轮廓点数。
- ClippingEndPoints：在轮廓的开始和结尾处被忽略的点数。
- Iterations：最大迭代次数。
- ClippingFactor：用于消除异常值的剪切因子。
- RowBegin、ColBegin：输出的线段起始点的行列坐标。
- RowEnd、ColEnd：输出的线段结束点的行列坐标。
- Nr、Nc：输出的法向量的行列坐标。
- Dist：输出的直线到原点的距离。

```
* 拟合圆 XLD
fit_circle_contour_xld(Contours : : Algorithm, MaxNumPoints, MaxClosureDist, ClippingEndPoints,
Iterations, ClippingFactor : Row, Column, Radius, StartPhi, EndPhi, PointOrder)
```

- Contours：输入的 XLD 轮廓。
- Algorithm：拟合圆的算法。
- MaxNumPoints：用于计算的最大轮廓点数。
- MaxClosureDist：被认为是闭合轮廓的端点之间的最大距离。
- ClippingEndPoints：在轮廓的开始和结尾处被忽略的点数。
- Iterations：最大迭代次数。
- ClippingFactor：用于消除异常值的剪切因子。
- Row、Column：输出的圆心的行列坐标。
- Radius：输出的圆的半径。
- StartPhi、EndPhi：输出的起始点和终止点的角度。
- PointOrder：输出的沿边界的点顺序。

```
* 拟合椭圆 XLD
fit_ellipse_contour_xld(Contours : : Algorithm, MaxNumPoints, MaxClosureDist, ClippingEndPoints,
VossTabSize, Iterations, ClippingFactor : Row, Column, Phi, Radius1, Radius2, StartPhi, EndPhi,
PointOrder)
```

- Contours：输入的 XLD 轮廓。
- Algorithm：拟合椭圆的算法。
- MaxNumPoints：用于计算的最大轮廓点数。
- MaxClosureDist：被认为是闭合轮廓的端点之间的最大距离。
- ClippingEndPoints：在轮廓的开始和结尾处被忽略的点数。
- VossTabSize：用于 Voss 方法的循环段的数量。
- Iterations：最大迭代次数。
- ClippingFactor：用于消除异常值的剪切因子。
- Row、Column：输出的椭圆圆心的行列坐标。
- Phi：输出的椭圆的主轴方向。
- Radius1、Radius2：输出的椭圆的长半轴和短半轴。
- StartPhi、EndPhi：输出的起始点和终止点的角度。
- PointOrder：输出的沿边界的点顺序。

```
* 拟合矩形 XLD
fit_rectangle2_contour_xld(Contours : : Algorithm, MaxNumPoints, MaxClosureDist, ClippingEndPoints,
Iterations, ClippingFactor : Row, Column, Phi, Length1, Length2, PointOrder)
```

- Contours：输入的 XLD 轮廓。
- Algorithm：拟合矩形的算法。
- MaxNumPoints：用于计算的最大轮廓点数。
- MaxClosureDist：被认为是闭合轮廓的端点之间的最大距离。
- ClippingEndPoints：在轮廓的开始和结尾处被忽略的点数。
- Iterations：最大迭代次数。
- ClippingFactor：用于消除异常值的剪切因子。
- Row、Column：输出的矩形中心的行列坐标。
- Phi：输出的矩形的主轴方向。
- Length1、Length2：输出的矩形的半长和半宽。
- PointOrder：输出的沿边界的点顺序。

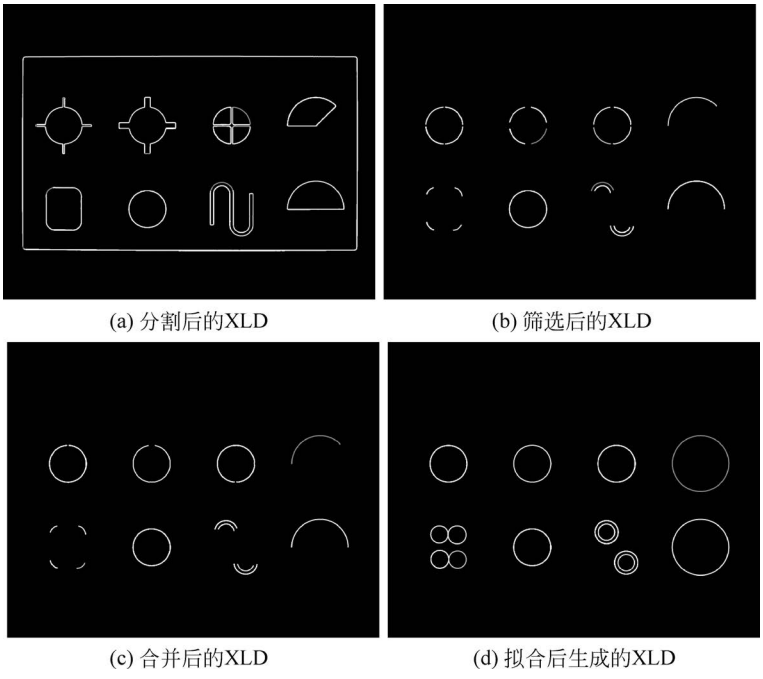
继续以分割小节的实例为基础，在分割之后，通过特征筛选，将 XLD 中的所有圆弧提取出来，并将共圆的圆弧合并，之后再通过拟合圆的方法，实现所有圆的拟合并创建相应的圆，以完成本节的拟合示例(彩色图片见文前彩插)，效果如图 5.10 所示，具体程序如下。

【例 5-7】 拟合 XLD 示例。

```
* 关闭窗口
dev_close_window()
* 读取图像
read_image(Image, 'circle_plate')
* 打开适应图像大小的窗口
dev_open_window_fit_image(Image, 0, 0, -1, -1, WindowHandle)
* 对图像进行亚像素阈值提取
threshold_sub_pix(Image,Border,80)
* 用圆度特征筛选圆孔
segment_contours_xld(Border, ContoursSplit, 'lines_circles', 5, 4, 2)
* 特征筛选
select_shape_xld(ContoursSplit, SelectedXLD, ['circularity','contlength'], 'and', [0.0697,31.6],
[1,2000])
* 合并共圆
union_cocircular_contours_xld(SelectedXLD, UnionContours1, 0.5, 0.1, 0.2, 30, 10, 10, 'true', 1)
* 拟合圆
fit_circle_contour_xld(UnionContours1, 'algebraic', -1, 0, 0, 3, 2, Row, Column, Radius,
StartPhi, EndPhi, PointOrder)
* 生成圆
gen_circle_contour_xld(ContCircle, Row, Column, Radius, 0, rad(360), 'positive', 1)
* 显示
dev_display(Border)
dev_display(SelectedXLD)
dev_display(UnionContours1)
dev_display(ContCircle)
```

▶ 5.3.5 其他

XLD 处理除了创建、分割、合并和拟合操作外，HALCON 还提供了一些其他方面方便 XLD 使用的算子，有用于 XLD 轮廓集合运算的差集算子 difference_closed_contours_xld()、交集算子 intersection_closed_contours_xld()和并集算子 union2_closed_contours_xld()；有



(a) 分割后的XLD　(b) 筛选后的XLD

(c) 合并后的XLD　(d) 拟合后生成的XLD

图 5.10　拟合示例 XLD(见彩插)

变换 XLD 形状的算子 shape_trans_xld()，该算子可以将原有 XLD 形状变换为相应的凸包、椭圆、外接圆或外接矩形等形状；有对 XLD 进行平滑的算子 smooth_contours_xld()，使用该算子可以得到更光滑的曲线；有对 XLD 进行排序的算子 sort_contours_xld()，XLD 排序之后使用循环配合 select_obj()算子，有利于对感兴趣的轮廓进行逐个处理。

XLD 处理算子的使用没有固定的模式，灵活搭配使用才能获得理想的效果。

在线测试

习题

5.1　XLD 获取的常见方法有哪些？

5.2　XLD 的特征筛选包含哪些特征？

5.3　XLD 的分割、合并和拟合分别实现了什么功能？