# 一维测量

视觉测量技术（Vision Measuring Technique) 是机器视觉 (Machine Vision)在测量领域内的应用，即用机器视觉代替人眼来测 量和判断，解决生产生活中的检测问题。视觉检测中的“检”，是指发现和识别，“测”是指几何参数和物理量的测量。视觉测量技术来 源于机器视觉技术，又不完全等同机器视觉。 视觉测量技术是测量技术的重要手段，应遵从于测量的基本规律， 又有一定的特殊性。

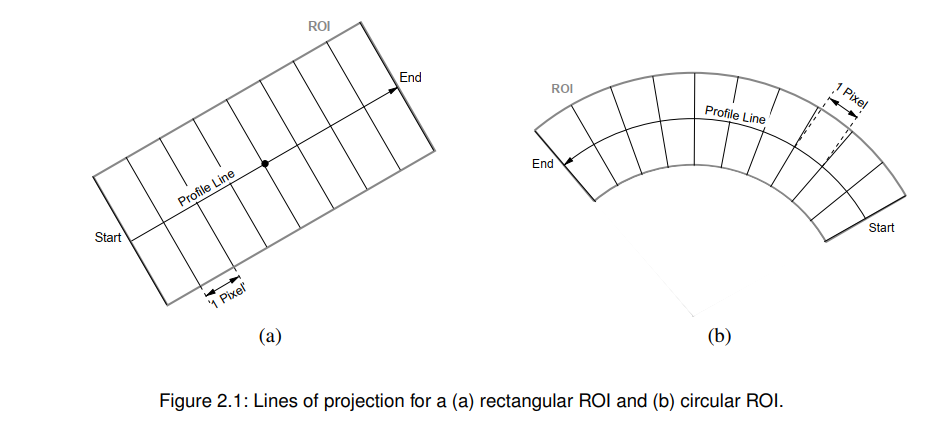
一般而言，有价值的测量方法应满足两个条件：首先是具备可靠 性和可用性，以及高度的环境适用性，对工作环境不能有过多限制和 苛刻的要求；其次，要有可靠的精度保障手段，要有可靠的误差系统 分析方法及精度传递手段，从理论和工程实践来保证测量的精度。

综上，视觉测量技术就是以机器视觉为理论基础，结合测量测试 理论，解决工程应用领域内的测量问题。其研究对象是三维空间内形 位（形态位置）尺寸，要求在满足一定精度的要求下，对被测对象实 现可靠测量。

测量对象背后的主要思想是提取与测量线或弧近似垂直的边缘。理解此过程非常重要，因为在创建度量对象和应用度量对象时，您都会使用参数影响该过程。请注意，与其他测量方法相比，测量roi不是使用reduce\_domain创建的，而是在创建测量对象时指定它们的维度。测量对象可用于确定与给定ROI近似垂直的直边的位置。

一维边缘提取过程

HALCON通过以下方法确定1D边缘的位置:首先，垂直于测量线或弧(也称为轮廓线)构建等距投影线，其长度等于ROI的宽度(图2.1)。



然后，计算沿每条投影线的平均灰度值。这些平均值的序列称为剖面。如果投影线不是水平或垂直方向的，则必须沿它们的像素值进行插值。在创建测量对象时，可以通过运算符的参数interpolation来选择插值方式。（如果interpolation = ' nearest\_neighbor '，则测量中的灰度值是从最近的像素的灰度值中获得的，即通过常数插值。这是最快的方法。然而，在这种模式下，几何精度略低。对于Interpolation = ' bilinear '，使用双线性插值，而对于Interpolation = ' bicubic '，使用双三次插值。双三次插值得到最精确的结果。然而，这是最慢的方法。）

投影线的长度，即ROI的宽度，决定了垂直于剖面线方向的平均程度。较宽的ROI相对应的轮廓噪声较小。因此，只要边缘近似垂直于轮廓线，ROI就应该选择尽可能宽的。如果边缘不垂直于轮廓线，则必须选择较小的ROI宽度以检测边缘。注意，在这种情况下，检测的对象将包含更多的噪声，即检测为边缘的准确率会受到小幅度的干扰。

较小的ROI宽度可以用高斯平滑滤波器对轮廓进行平滑，其标准差用测量算子的参数Sigma指定。一阶导数的所有局部极值的亚像素精确位置都是候选边缘。这些候选边缘由从导数的各自位置指向平滑的灰度值轮廓的向量表示。只有那些一阶导数绝对值大于给定阈值(测量算子的另一个参数)的候选边缘才被认为是图像中的边缘。对于每条边，返回它与轮廓线交点的位置。

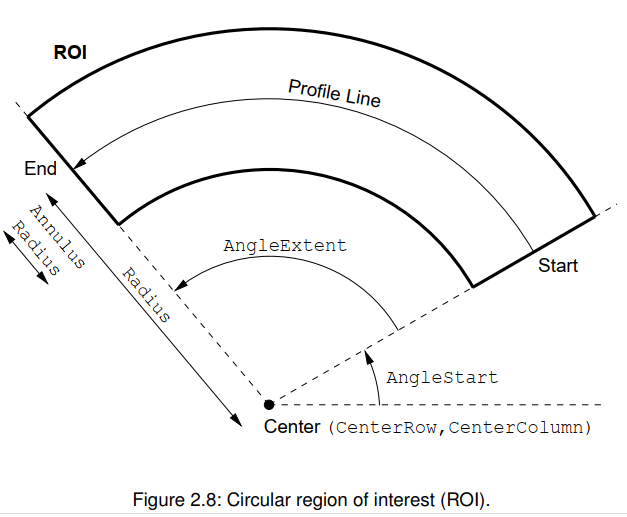
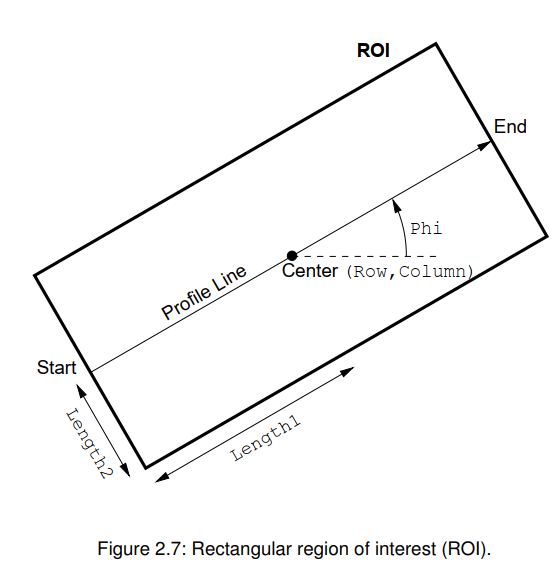
图2.4为检测到的边缘，用直线显示，其长度与ROI的宽度相似。在1D边缘提取的情况下，仅确定边缘与轮廓线相交的位置。测量对象被设计用来测量与轮廓线近似垂直的直边的位置。这意味着，测量对象不适合测量含有曲线轮廓的位置，例如，弯曲的边缘。

测量对象可以创建两种不同的形状:旋转矩形和圆弧：

\*创建矩形测量对象  
gen\_measure\_rectangle2( : : Row, Column, Phi, Length1, Length2, \  
Width, Height, Interpolation : MeasureHandle)  
\*创建圆形测量对象  
gen\_measure\_arc( : : CenterRow, CenterCol, Radius, AngleStart, \  
AngleExtent, AnnulusRadius, Width, Height, Interpolation : MeasureHandle)

一个旋转的矩形由它的中心(Row, Column)，它的方向Phi，以及它的半边长度Length1和Length2来定义(参见图2.7)。

圆弧由以下参数定义(见图2.8):圆弧的位置和大小由圆心(CenterRow, CenterColumn)和半径(radius)定义。圆弧由其起始角度(AngleStart)和角度范围(AngleExtent)定义。圆形ROI的宽度由环半径(AnnulusRadius)定义。



使用一D测量

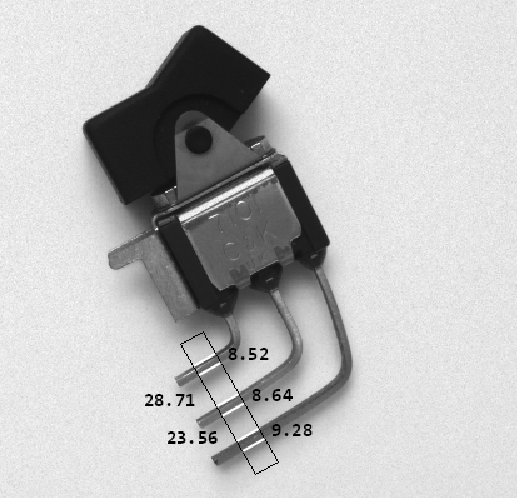
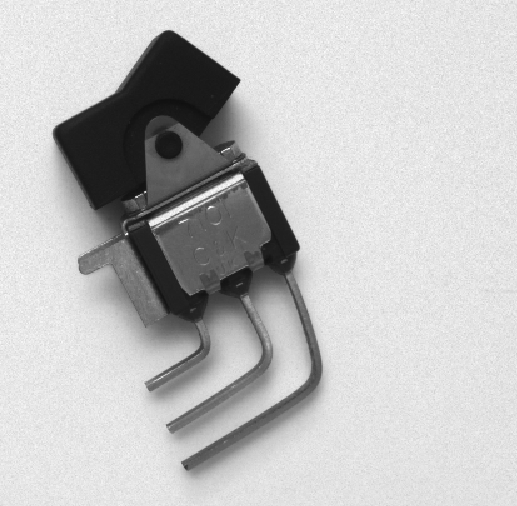
对于不同的任务，根据应用需求，可以测量各种边缘的位置和它们之间的距。

\* 读取图像  
read\_image (Image, 'bin\_switch/bin\_switch\_1')  
get\_image\_size (Image, Width, Height)  
\*定义ROI区域并创建测量对象  
Row := 385  
Column := 385  
Phi := rad(-60)  
Length1 := 60  
Length2 := 10  
Interpolation := 'nearest\_neighbor'  
gen\_measure\_rectangle2 (Row, Column, Phi, Length1, Length2, Width,\  
Height, Interpolation, MeasureHandle)  
\*确定所有具有负过渡的边对，即边对包围黑暗区域  
Sigma := 1.1  
Threshold := 20  
Transition := 'negative'  
Select := 'all'  
measure\_pairs (Image, MeasureHandle, Sigma, Threshold, Transition, \  
Select, RowEdgeFirst, ColumnEdgeFirst, AmplitudeFirst, RowEdgeSecond, \  
ColumnEdgeSecond, AmplitudeSecond, IntraDistance, InterDistance)  
\*删除测量对象  
close\_measure (MeasureHandle)

参数Transition可用于选择具有特定过渡的边对。如果Transition设置为“negative”，则为负过渡。如果将过渡设置为“position”,将返回包围明亮图像区域的边缘对。如果设置为“all”，则过渡由第一条边决定。即根据测量对象的位置，返回具有光-暗-光过渡的边缘对或具有暗-光-暗过渡的边缘对。这适用于测量相对于背景具有不同亮度的物体。

输出参数IntraDistance和InterDistance包含边对两端的距离和边对之间的距离。

参数Select可用于将结果限制为“第一个”或“最后一个”边对。要确定所有边对，Select必须设置为all



\* 引脚测量:测量引脚相关尺寸的应用示例  
\*   
\*清空屏幕，显式控制图像显示  
dev\_close\_window ()  
read\_image (Image, 'ic\_pin')  
get\_image\_size (Image, Width, Height)  
dev\_open\_window (0, 0, Width / 2, Height / 2, 'black', WindowHandle)  
set\_display\_font (WindowHandle, 14, 'mono', 'true', 'false')  
dev\_display (Image)  
disp\_continue\_message (WindowHandle, 'black', 'true')  
stop ()  
\* draw\_rectangle2 (WindowHandle, Row, Column, Phi, Length1, Length2)  
\*绘制测量矩形  
Row := 47  
Column := 485  
Phi := 0  
Length1 := 420  
Length2 := 10  
dev\_set\_color ('green')  
dev\_set\_draw ('margin')  
dev\_set\_line\_width (3)  
gen\_rectangle2 (Rectangle, Row, Column, Phi, Length1, Length2)  
gen\_measure\_rectangle2 (Row, Column, Phi, Length1, Length2, Width, Height,\  
'nearest\_neighbor', MeasureHandle)  
disp\_continue\_message (WindowHandle, 'black', 'true')  
stop ()  
\*测量所有引脚的宽度和引脚之间的距离  
measure\_pairs (Image, MeasureHandle, 1.5, 30, 'negative', 'all', \  
RowEdgeFirst, ColumnEdgeFirst, AmplitudeFirst, RowEdgeSecond, \  
ColumnEdgeSecond, AmplitudeSecond, PinWidth, PinDistance)  
disp\_continue\_message (WindowHandle, 'black', 'true')  
stop ()  
\*显示引脚的位置  
dev\_set\_color ('red')  
disp\_line (WindowHandle, RowEdgeFirst, ColumnEdgeFirst, RowEdgeSecond, ColumnEdgeSecond)  
\*对测量结果进行处理并显示  
avgPinWidth := sum(PinWidth) / |PinWidth|  
avgPinDistance := sum(PinDistance) / |PinDistance|  
numPins := |PinWidth|  
dev\_set\_color ('yellow')  
disp\_message (WindowHandle, 'Number of pins: ' + numPins, 'image',\  
200, 100, 'yellow', 'false')  
disp\_message (WindowHandle, 'Average Pin Width: ' + avgPinWidth, \  
'image', 260, 100, 'yellow', 'false')  
disp\_message (WindowHandle, 'Average Pin Distance: ' + avgPinDistance,\  
'image', 320, 100, 'yellow', 'false')  
\* dump\_window (WindowHandle, 'tiff\_rgb', 'C:\\Temp\\pins\_result')  
disp\_continue\_message (WindowHandle, 'black', 'true')  
stop ()  
\* draw\_rectangle1 (WindowHandle, Row1, Column1, Row2, Column2)  
\*局部放大测量芯片引脚宽的测量矩形  
Row1 := 0  
Column1 := 600  
Row2 := 100  
Column2 := 700  
dev\_set\_color ('blue')  
disp\_rectangle1 (WindowHandle, Row1, Column1, Row2, Column2)  
stop ()  
dev\_set\_part (Row1, Column1, Row2, Column2)  
dev\_display (Image)  
dev\_set\_color ('green')  
dev\_display (Rectangle)  
dev\_set\_color ('red')  
disp\_line (WindowHandle, RowEdgeFirst, ColumnEdgeFirst, RowEdgeSecond, ColumnEdgeSecond)  
disp\_continue\_message (WindowHandle, 'black', 'true')  
stop ()  
dev\_set\_part (0, 0, Height - 1, Width - 1)  
dev\_display (Image)  
disp\_continue\_message (WindowHandle, 'black', 'true')  
stop ()  
dev\_set\_color ('green')  
\* draw\_rectangle2 (WindowHandle, Row, Column, Phi, Length1, Length2)  
\*绘制测量芯片引脚高度的测量矩形  
Row := 508  
Column := 200  
Phi := -1.5708  
Length1 := 482  
Length2 := 35  
gen\_rectangle2 (Rectangle, Row, Column, Phi, Length1, Length2)  
gen\_measure\_rectangle2 (Row, Column, Phi, Length1, Length2, Width, \  
Height, 'nearest\_neighbor', MeasureHandle)  
stop ()  
\*测量俩端引脚的高度  
measure\_pos (Image, MeasureHandle, 1.5, 30, 'all', 'all', RowEdge, \  
ColumnEdge, Amplitude, Distance)  
PinHeight1 := RowEdge[1] - RowEdge[0]  
PinHeight2 := RowEdge[3] - RowEdge[2]  
\*显示测量结果  
dev\_set\_color ('red')  
disp\_line (WindowHandle, RowEdge, ColumnEdge - Length2, RowEdge, ColumnEdge + Length2)  
disp\_message (WindowHandle, 'Pin Height: ' + PinHeight1, 'image',\  
RowEdge[1] + 40, ColumnEdge[1] + 100, 'yellow', 'false')  
disp\_message (WindowHandle, 'Pin Height: ' + PinHeight2, 'image',\  
RowEdge[3] - 120, ColumnEdge[3] + 100, 'yellow', 'false')  
dev\_set\_draw ('fill')  
dev\_set\_line\_width (1)