# TwinCAT Vision怎样实现目标识别定位

### 引言：

本文介绍使用TwinCAT Vision算法实现目标识别确定ROI的一种方法。特别适合图像像素尺寸比较大的场合。视觉检测不管是定位，测量，还是缺陷检测第一步要做的其实都是目标识别，确定目标所在的区域ROI（**Region of Interest**）。用过其他商用视觉算法库的工程师往往习惯于用有角度输出的模板匹配的算法，在视野内找到目标的位置，将目标识别和定位一次实现。但TwinCAT Vision 里的模板识别没有角度输出，所以需要位置和角度输出时，需要先做目标识别确定ROI，再对ROI内的目标特征分析获得准确的定位坐标和角度。

### 正文：

TwinCAT Vision的模板识别算法都在TF7200这个授权产品内，用于匹配 2D 的扩展功能：可根据学习到的参考特征、轮廓、特征点或其他属性（模板匹配/关键点检测和描述符匹配）查找和比较对象。常见的应用是对象分类。这里的模板匹配算法是基于像素分布的模板匹配，只能找到模板的中心位置的坐标。如果对目标在画面内的位置做粗略位置定位也是可以使用模板的中心点坐标加上模板的长宽尺寸的，但因为基于目标的像素分布用它来做定位会带来两个问题：

* 1. 要将模板上所有图像像素和待测图像像素循环遍历计算量大，影响任务节拍，消耗算力资源；
  2. 模板图像的像素复杂度和对称性会严重影响模板匹配结果的置信值，复杂度低，对称性强会导致漏判或误判。

所以，如果只是第一步确定目标的ROI，不建议用模板识别的方法。那么用TwinCAT Vison做目标识别还是回到图像处理的常用方法步骤：

1. 预处理：减少干扰，增强目标特征，然后做
2. 图像分割，提取特征
3. 分析特征，确定目标的位置。

这里做图像预处理的方法要根据检测需求和定制的取像方案获得的实际图像的情况来做处理，可以从像素值域，和频域两个方向着手。如果相机镜头搭配合适的打光方案能够获得突出要检测的特征（信号），消弱不需要的特征（噪声干扰）的效果，可以不做预处理直接进行下面一步图像分割，提取特征。

图像分割常用的是基于像素值大小的阈值分割，TwinCAT Vision提供的阈值分割函数有以下几个：

* [F\_VN\_AdaptiveThreshold(Exp)](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16926135051.html#5562872459)
* [F\_VN\_CheckColorRange](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16927252107.html#5777337739)
* [F\_VN\_DoubleThreshold](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16928443403.html#16715963275)
* [F\_VN\_Threshold](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16933962507.html#5562820107)
* [F\_VN\_WatershedSegmentationExp](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16934270219.html#7860553611)

**Samples**

[Image segmentation](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/5247041035.html)

使用手册里还附上了例程。

这里最常用的是黑白图像的单阈值分割函数F\_VN\_Threshold, 和彩色图像的上下阈值分割函数F\_VN\_CheckColorRange;

文本

描述已自动生成 图像分割后提取特征一般会用到轮廓提取函数。TwinCAT Vision 使用手册里把这类函数分类为图像分析（Image Analysis）的子类目标识别函数（Object detection）有如下几个：

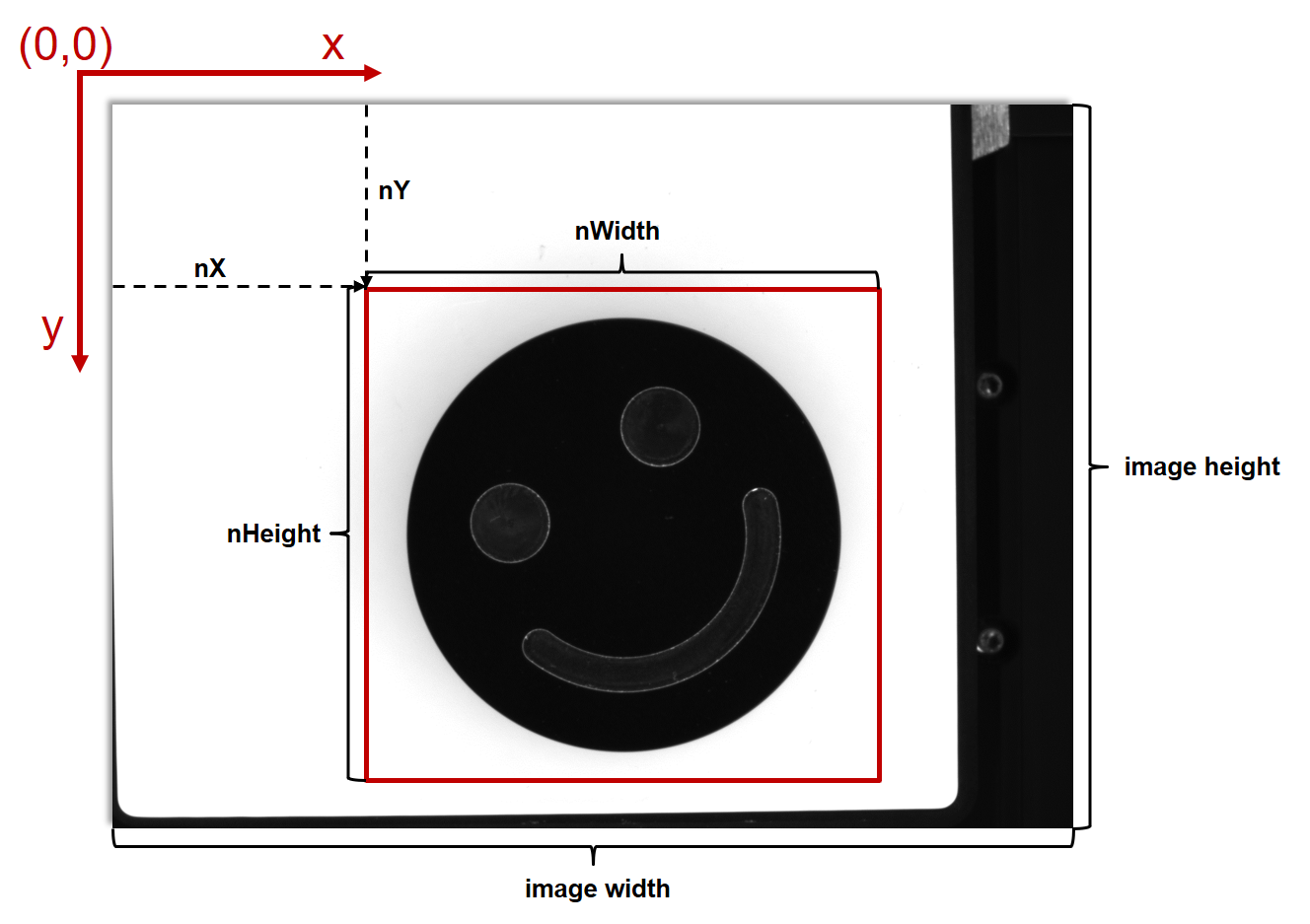
* [F\_VN\_DetectBlobs(Exp)](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16928324363.html#5562923147)
* [F\_VN\_FindContours(Exp)](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16929736715.html#5562930315)
* [F\_VN\_FindContourHierarchyExp](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16929719563.html#5562928523)
* [F\_VN\_HoughCircles(Exp)](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16930643211.html#5562926731)
* [F\_VN\_HoughLines(Exp)](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16930659339.html#7860487307)
* [F\_VN\_HoughLinesP(Exp)](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16930675467.html#7860494603)

这里常用的blob分析函数F\_VN\_DeltectBlobs(Exp), 其实集合了前面F\_VN\_Threshold阈值二值化图像，和F\_VN\_FindContours(Exp)或F\_VN\_FindContourHierarchyExp轮廓提取, 加上一些按不同轮廓特征值参数(例如：轮廓包围的面积，圆度，长宽比等等)筛选轮廓的功能，是一个复合的应用函数。所以常常可以直接对预处理过的图像用一个bolb分析实现目标识别。

最后对筛选识别后的轮廓，分析其特征，一般会找到其最大外接矩形，如果是圆形或椭圆形目标就做圆或椭圆的拟合，利用其中心点坐标做定位，如果需要得到目标的角度，还需要进一步再通过最大外接矩形获得目标的ROI后进一步对ROI内的目标特征提取，分析得到角度值。比如特殊关键点的连线的角度，或边缘的角度。这里TwinCAT Vision 提供了下列函数：

* F\_VN\_[EnclosingRectangle](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16928846219.html?id=2789217319482039495)
* F\_VN\_[EnclosingCircle](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16928837131.html?id=1158981792106570595)
* F\_VN\_[FitEllipse](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16929821579.html?id=4023855923166475545)
* F\_VN\_[FitLine](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16929830667.html?id=3757975906612981916)

这里如果需要设置局部区域ROI，那就需要得到一个包含轮廓平行X轴和Y轴的图像区域。



TwinCAT Vision提供了函数 [**F\_VN\_UprightBoundingRectangle**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16934154123.html?id=5018305615897039485)用来获取ROI的矩形变量TcVnRectangle\_UDINT

**表格

描述已自动生成**

获得了这个ROI的矩形，就可以用TwinCAT Vision提供下列函数，灵活的设置局部ROI的图像变量，可以降低待处理图像尺寸，提高检测速度，实现在原图中检测多个目标物。

* F\_VN\_[SetRoi](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16933793035.html?id=2006667510009455934)
* F\_VN\_[SetRoi\_TcVnRectangle\_DINT](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16933801099.html?id=6866556482047281716)
* F\_VN[\_SetRoi\_TcVnRectangle\_UDINT](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16933810187.html?id=3940628104670562105)
* F\_VN\_[ResetRoi](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16933306635.html?id=4216479573235130633)

但用这种图像处理分析的方法，特别是用blob分析的方法找到ROI的过程，其实也是基于像素的处理，避免不了遍历原图所有的像素，同样存在计算量大，影响任务节拍，消耗算力资源的问题，这里如果原图像素比较大，又有检测节拍的要求时可以利用金字塔图层F\_VN\_[PyramidDown](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16933029259.html?id=4175779436999820695)降采样算法，先对原图做降采样处理，然后再做上面这一套确定目标ROI的处理。因为这里只需要确定目标的ROI区域实现初步定位目标区域，用这个区域来设置局部ROI裁剪原图尺寸，从而实现降低图像尺寸，降低图像处理数据量，缩减运行时间，提高处理节拍。所以使用降采样导致的找到ROI区域的误差可以忽略不计。

FOR i:=0 TO nlevel DO

hr := F\_VN\_PyramidDown(ipImageTemp , ipImageTemp, hr);

END\_FOR

fPow :=EXPT(nBase,nlevel+1);

在找到ROI后按照降采样的比例放大ROI的尺寸：

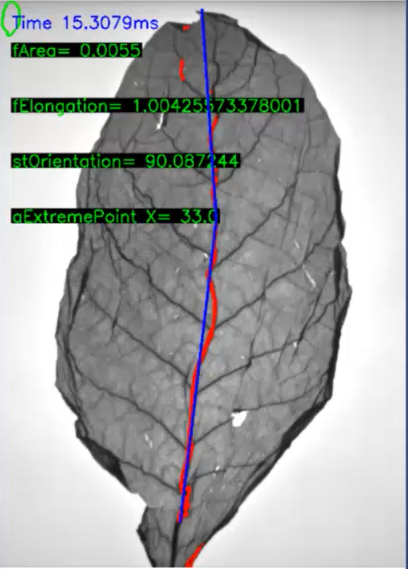
ROIRect.nX:=stRect.nX\*LREAL\_TO\_UDINT(fPow);

ROIRect.nY:=stRect.nY\*LREAL\_TO\_UDINT(fPow);

ROIRect.nWidth:=stRect.nWidth\*LREAL\_TO\_UDINT(fPow);

ROIRect.nHeight:=stRect.nHeight\*LREAL\_TO\_UDINT(fPow);

这样处理后可以大量压缩整个视觉处理的处理时间，再加PyramidDown和blob分析函数都支持job task 分核加速，一般都可以获得一个非常短得处理时间。如下图所示左上角绿色轮廓线是经过3次降采样处理找到的叶子的外轮廓：



### 结束语

TwinCAT Vision 提供大量非常灵活的图像处理的算法，可以根据实际应用需求发挥工程师的创造力进行搭配实现相应的功能，而且还可以保证代码执行的效率，这可能需要工程师投入一定的时间和精力。采用其他图形化编程或针对某些行业应用开发的商用库可能在初期评估开发时上手快，但最终应用到实际的项目中因为灵活度差或者代码冗余太多，可能会不能满足最终程序节拍要求。所以有灵活性定制要求和高速实时性要求的应用，TwinCAT Vison的视觉解决方案有可能获得其他视觉产品达不到的效果。