TwinCAT Vision 的模板匹配算法

##### 引言：

目标识别后的定位是视觉应用的一个重要分支，定位的精确度和准确性是后续工序的基础，比如运动控制的定点抓取，分拣，搬运等。本文通过介绍TwinCAT Vision 模板匹配算法，列举了几种使用模板匹配实现定位的方法。

##### 正文：

**方法一：** 基于像素分布的模板匹配。如果不是很精确的定位，可以直接用像素模板匹配从而直接获得模板图像的中心坐标Center（x，y）作为定位输出。

TwinCAT Vision提供了下列基于像素模板的模板匹配函数：

* **[F\_VN\_MatchTemplate](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931607947.html?id=5055382595838997025)**
* **[F\_VN\_MatchTemplateExp](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931632139.html?id=5793465274639940920)**
* [**F\_VN\_MatchTemplateAndEvaluate**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931616011.html?id=1419930009766537317)
* [**F\_VN\_MatchTemplateAndEvaluateExp**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931624075.html?id=4110125047843484094)

结果输出变量：

* **ipResultImage** ：是将模板图像从左到右，从上到下在需要检测的目标图像上按像素顺序移动，每移动一次比对模板图像与目标图像对应区域的像素相似度，接着再进行归一化处理，从而得到每个像素位置的对比结果。

所以新生成的 ipResultImage的尺寸为（ipSrcImage.width-ipTemplateImage.width+1）x（ipSrcImage.height-ipTemplateImage.height+1）；因为模板图像需要在目标图像的每个可能的位置进行匹配，所以结果图像的尺寸取决于目标图像和模板图像的相对大小，最佳的匹配位置是ipResultImage上像素值最大的点，而这个最大值在ipResultImage上的坐标表示的是ipTemplateImage的左上角坐标。由于这个ipResultImage结果是归一化处理过的图像变量，像素值范围[0..1]，是1通道图像，所以如果要用Image Watch窗口查看匹配结果还需要用到将归一化图像可视化的函数[F\_VN\_NormalizeImageForDisplay](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16932846603.html?id=527406636303538916)

图形用户界面

描述已自动生成

使用这个函数进行模板匹配需要有个前提：目标图像是几何对称的，且定位目标的放置姿态保持稳定即没有多少角度的变化。TwinCAT Vision的在线手册配套例程下载<https://github.com/Beckhoff/TF7xxx_Samples>里提供了一个使用F\_VN\_MatchTemplateAndEvaluate的样例，该函数模板匹配后的输出变量：

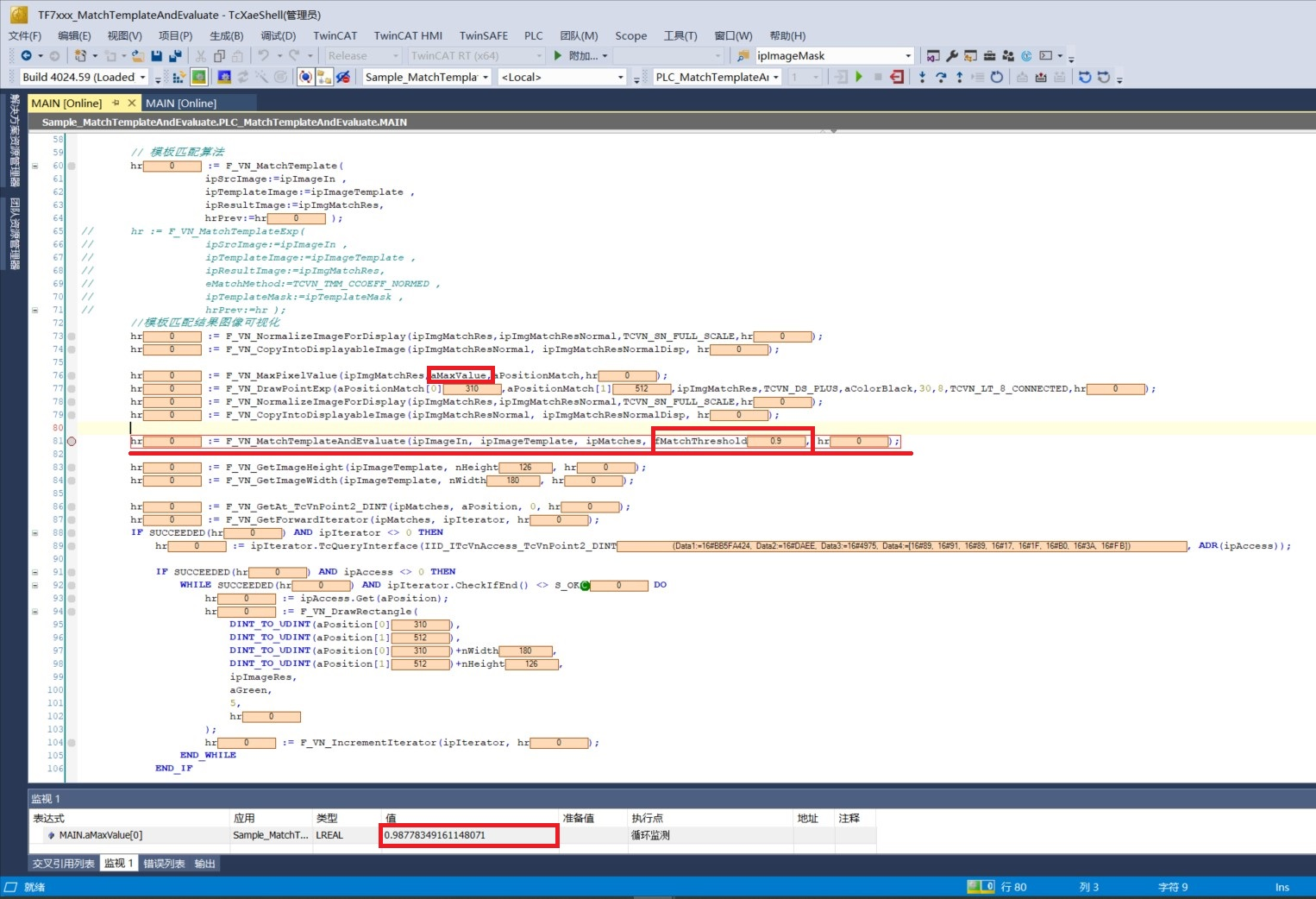
* **ipMatches** ：是Container变量（ContainerType\_Vector\_TcVnPoint2\_DINT），其中每个元素

表示ipTemplateImage的左上角在ipSrcImage图像内的坐标，也就是把可能的匹配坐标都罗列了出来，按照相关性大小，最佳匹配优先，也就是前面提及的ipResultImage里的所有像素值按照从大到小进行排序，排在第一位的就是最佳匹配。所以该样例被归类在图像分析（ImageAnalysis）下的目标识别（ObjectDetection）文件夹中，因为该函数的主要应用并非定位，而是通过使用这个评估函数设置fMatchThreshold的阈值大小来判断当前画面里是否有与模板匹配的图像。下图程序中分别使用：

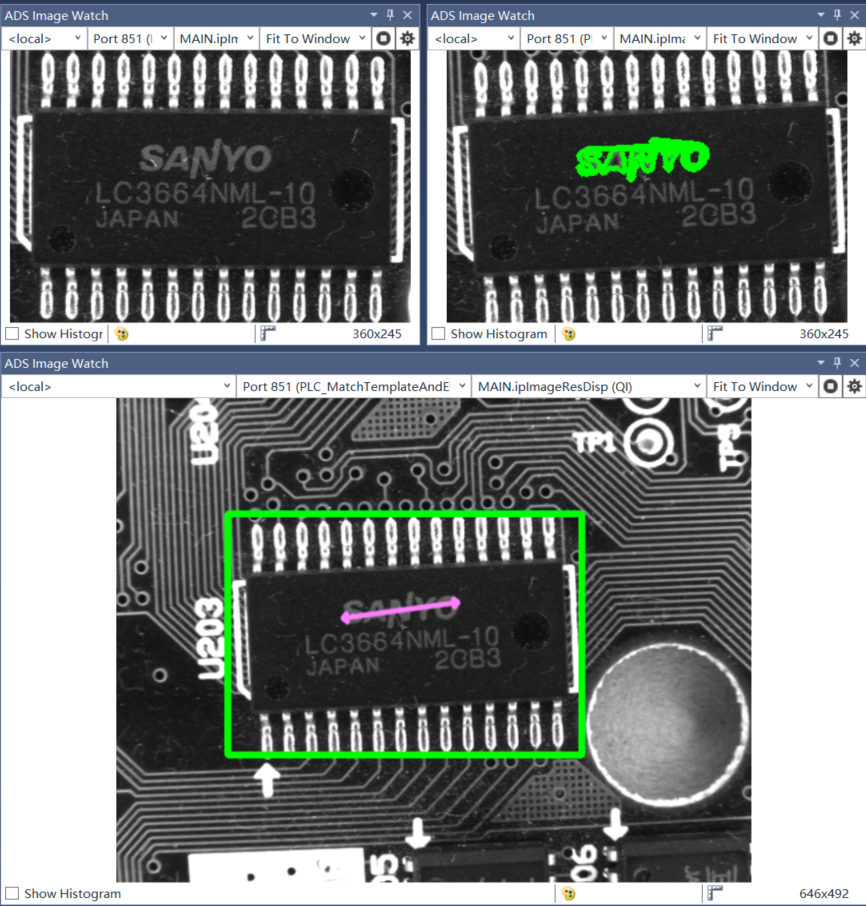
• F\_VN\_MatchTemplate

• F\_VN\_MatchTemplateAndEvaluate

两个函数。对F\_VN\_MatchTemplate的输出图像用函数F\_VN\_MaxPixelValue求最大像素值点。可以得出实际最佳匹配位置的相关性值为0.98778349161148071，这是归一化处理后的数值。所以把fMatchThreshold设置成0.9就可以得到最佳匹配模板的目标图像位置，可在目标图像上绘制出模板图像的所在位置，如上图绿色框所示。

****

但是这个定位只能确定模板图像的中心位置，或是计算模板图像某个固定特征的坐标加上模板左上角在待测图像里找到的坐标。定位不精确是因为如果目标图像里目标有角度的，必然会导致模板匹配后的相关性系数变低，可能达不到接近1的结果，这时放宽fMatchThreshold的阈值设置大小可以帮助判断待测图像里是否有模板目标，以及模板目标的大概位置区域，但是做定位就缺乏准确性了。如果需要实现精确定位，可以利用上述函数找到的最大相关系数值所对应的模板图像的左上角的坐标点，以其为局部坐标参考点，然后从X，Y轴方向适量选择长宽后设置ROI，在目标图像的ROI区域中去找固定的几何图形特征，比如边缘的交点，圆形或矩形的中心点等作为定位点的坐标输出。如果没有前面提到几何图形特征，只有纹理特征，可以使用函数提取到稳定的纹理特征轮廓，然后计算轮廓的最大外接旋转矩形的中心点。如下图所示



而这种有明显的轮廓特征的目标定位，可以用轮廓分析算法里的基于轮廓的模板匹配的算法来实现，即方法二。

**方法二：**基于轮廓模板的模板匹配。Twin CAT Vision提供了下列算法：

* [F\_VN\_MatchContours](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931500043.html?id=2786360644954059063)
* [F\_VN\_MatchContoursExp](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931525259.html?id=3474075367547610287)
* [F\_VN\_MatchContours1vsN](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931509131.html?id=6296583951486035848)
* [F\_VN\_MatchContours1vsNExp](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931517195.html?id=4895934046903978645)

在线例程[Samples](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/4333986955.html?id=6166133478361286762)的轮廓分析（ContourAnalysis）分类里给出了例程的代码说明。轮廓模板的获取渠道分为两类：

* 人工创建一个几何图形的模板

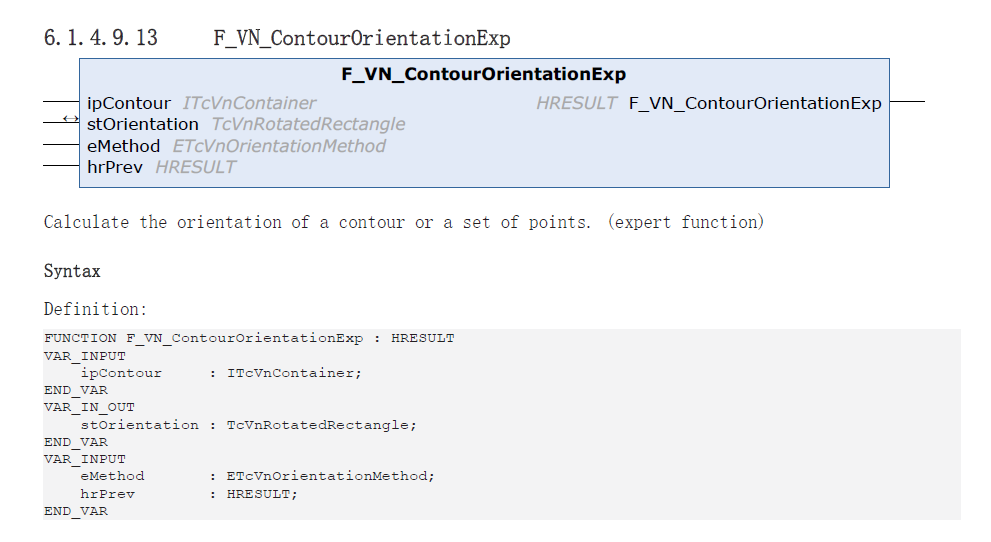
[Match Contours 1vsN (manual shapes)](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/5483334539.html?id=18306933283316650)

* 对模板图形提取轮廓特征

[Match Contours (extracted shapes)](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/5483464843.html?id=3077670356412699392)

这两种模板匹配的基础函数都是[F\_VN\_MatchImageHuMoments](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931598859.html?id=2178412109479000284)，该函数比较的是模板轮廓和待检测图形轮廓的Hu距，得到轮廓的相异性值fDissimilarity，如果相似度越大这个值越接近于零。Hu矩在图像旋转、缩放、平移等操作后，仍能保持矩的不变性，所以用Hu矩可以实现有角度旋转的模板匹配，从而实现位置定位，和角度定位输出。那么获得最佳匹配结果后，怎么得到目标轮廓的角度呢？

可以使用函数：[F\_VN\_ContourOrientationExp](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16927609483.html?id=5568617624824339409)



其中输入变量eMethod枚举了计算点集（轮廓）的几种方法：

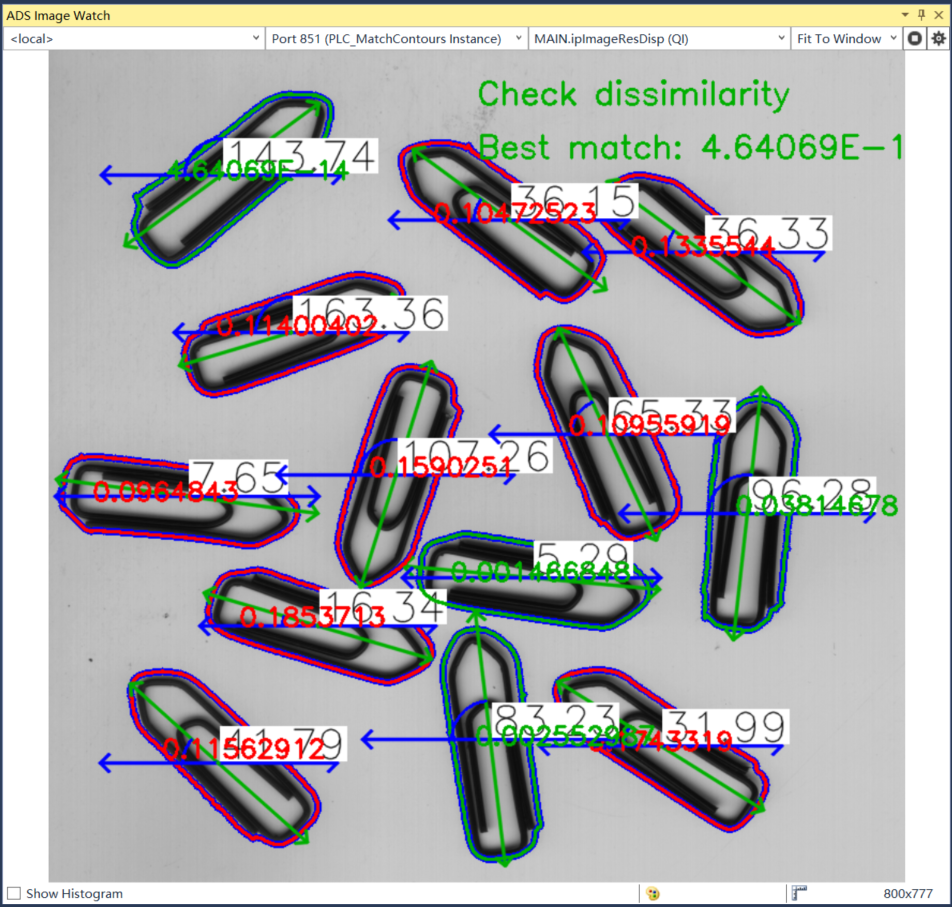


输出变量stOrientation 的数据类型是TcVnRotatedRectangle，其数据成员有旋转矩形的中心点aCenter和角度fAngle，也就是目标轮廓定位需要的坐标和角度。

TCVN\_OM\_PCA 里PCA是主成分分析，PCA通过计算轮廓点的平均中心作为中心点，进而计算出主轴的旋转角度。这种方法特别适用轴左右对称，又有明显上下区分的轮廓结构，如下图所示目标：

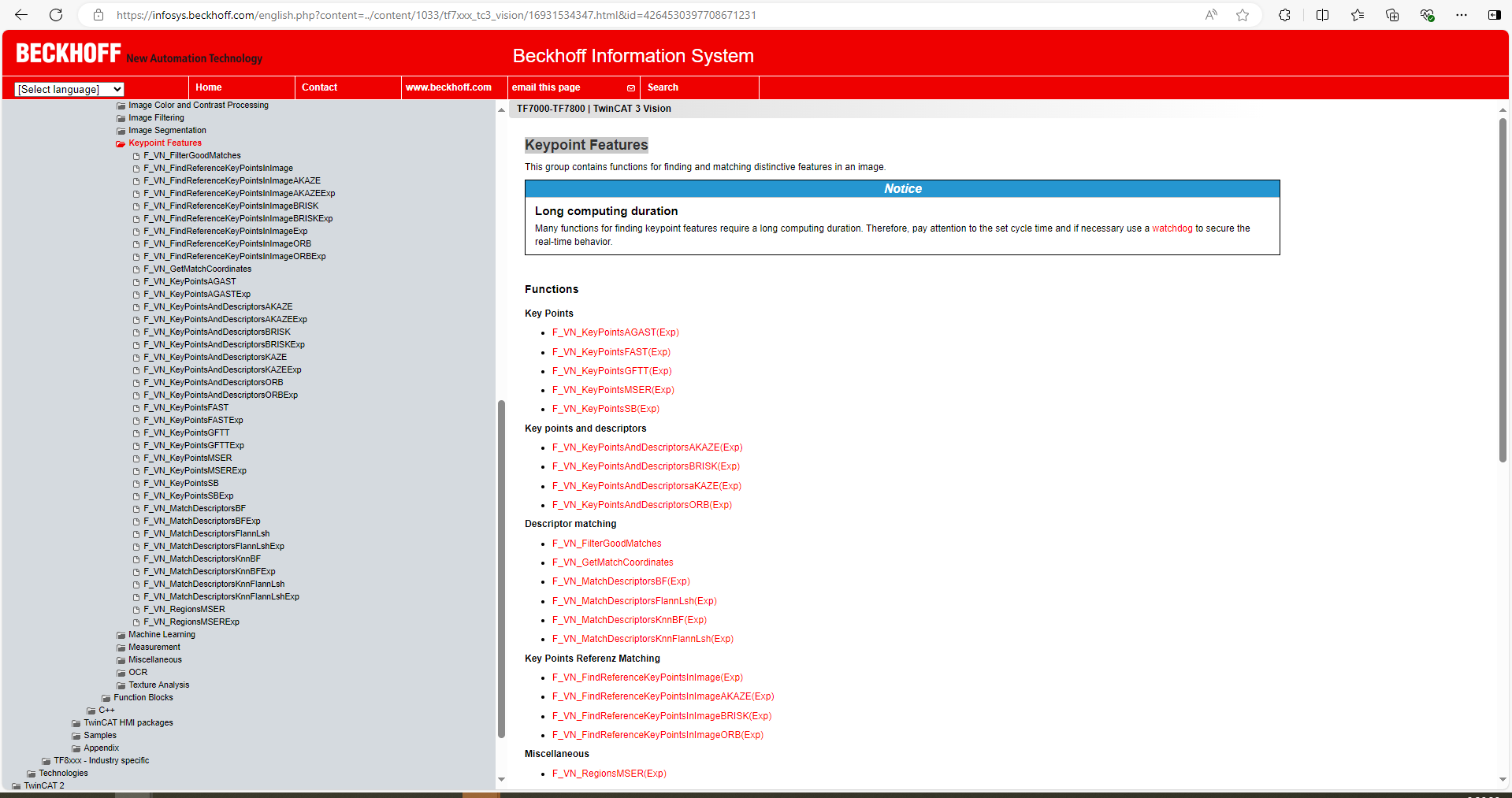
而TCVN\_OM\_ENCLOSINGRECT，就是前文提到的找轮廓的最大外接旋转矩形。

可以用[F\_VN\_DrawOrientationExp](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16928711947.html?id=5128074275843863589)函数直观地绘制并显示定位结果的中心点和角度。



**方法三：特征点匹配**

严格来说特征点匹配不能算模板匹配，但是有些特别的应用，可以用特征点匹配的算法来实现模板匹配的功能，和轮廓特征不同，特征点的特征具有更大的离散性。TwinCAT Vison也提供了大量的提取特征点的算法函数



常见特征点匹配算法：

[**F\_VN\_MatchDescriptorsBF**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931534347.html?id=4264530397708671231)

[**F\_VN\_MatchDescriptorsBFExp**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931542411.html?id=4132802412280217451)

[**F\_VN\_MatchDescriptorsFlannLsh**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931550475.html?id=3519770276674740112)

[**F\_VN\_MatchDescriptorsFlannLshExp**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931558539.html?id=5551413825922356613)

[**F\_VN\_MatchDescriptorsKnnBF**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931566603.html?id=787917588191989827)

[**F\_VN\_MatchDescriptorsKnnBFExp**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931574667.html?id=1318777692091259793)

[**F\_VN\_MatchDescriptorsKnnFlannLsh**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931582731.html?id=1774544868096088684)

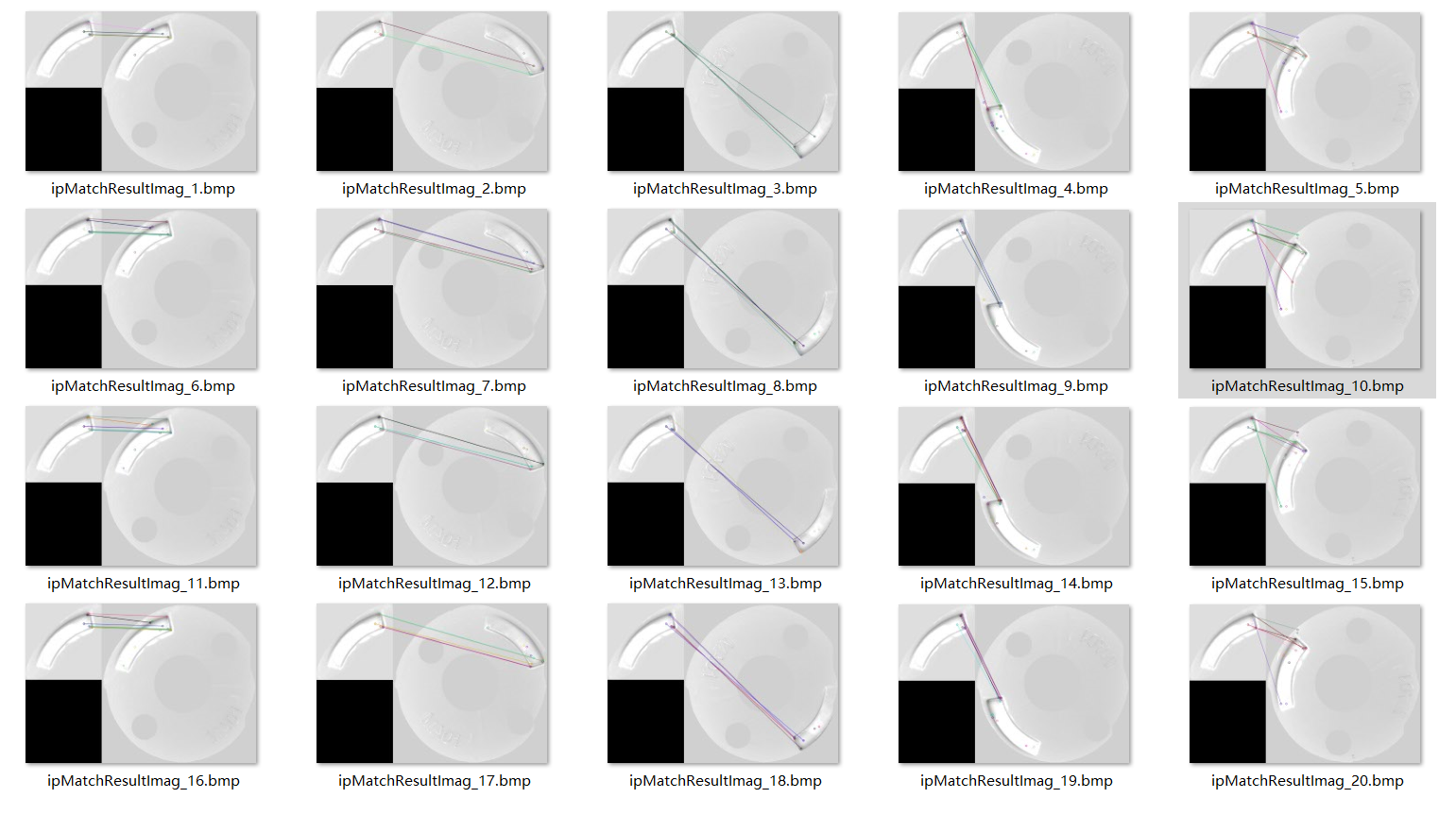
[**F\_VN\_MatchDescriptorsKnnFlannLshExp**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16931590795.html?id=2649312425173355489)

特征点匹配得到的结果：

* **ipMatches：是**Container变量（ContainerType\_Vector\_TcVnDMatch），

给出匹配的两幅图像的特征点之的匹配结果。继而可以用函数[**F\_VN\_GetMatchCoordinates**](https://infosys.beckhoff.com/content/1033/tf7xxx_tc3_vision/16930408971.html?id=7016256170119229977)获得相互匹配的点的坐标变换矩阵。

通过分析这些坐标点的特征可以得到定位和角度。但使用该函数的前提是特征点的特征非常稳定，且具有一定的结构复杂性。这种特征提取匹配一般用在大场景图像的拼接算法中，或者医学影像学的图像配准领域等复杂环境。因为计算量大，会消耗大量运算资源，所以在工业在线检测场合下，一般不推荐使用。



结束语：

所以如果需要用模板匹配的方式做定位，输出目标的位置和角度结果，有几种方法：

* 基于像素匹配函数初步定位目标后，进一步利用目标的固定的几何特征（如中心点或边缘的相交点）进行精确定位，最终获得目标的位置坐标和角度。
* 基于轮廓匹配函数，然后分析出目标轮廓的特征的中心点和角度
* 少数情况下可以直接使用特征点匹配的方式进行目标识别定位