

接口声明	int GetDetectResult(const char *szTXTDir, const char *szMarkPath, const char *szModelPath, const char *szCaliPath, const char *szOutDir, int iParams[], float fParams[], char *szResult);			
参数列表	szTXTDir	[IN]	depth.txt 和 center.txt 所在文件夹，如：C:\\filesDir	
	szMarkDir	[IN]	mark.txt 所在文件夹，如：C:\\markDir	
	szModelPath	[IN]	标记点模板文件路径，如：C:\\mark.txt	
	szCaliPath	[IN]	校准文件 cali.txt 文件路径，如：C:\\cali.txt	
	szOutDir	[IN]	输出点云的文件夹，如：C:\\outfileDir	
	iParams[]	[IN]	iParams[0]	图像宽度
			iParams[1]	图像高度
			iParams[2]	光栅片数(1,2,3)
			iParams[3]	szResult 数组长度
	fParams[]	[IN]	fParams[0]	缺陷面积阈值，以像素为单位
szResult	[OUT]	返回缺陷信息（长度由参数传入）		
接口返回值	0	成功		
	-2	txt 读取失败		
	-3	无效的标记点图像		
	-4	无效的光栅片数		
	-6	参数列表有误		

备注:

1. szTXTDir 文件夹存放缺陷的深度计算结果 depth.txt 以及中心视角图 center.txt，命名格式遵循以下准则：

光栅片数为 1 时，需要进行 2 次深度计算；光栅片数为 2 时，进行 3 次深度计算；光栅片数为 3 时，进行 4 次深度计算；

第 N 次深度计算生成的文件命名格式为 depthN.txt;centerN.txt(N=1,2...光栅片数+1)

举例：

若光栅片数为 1，第一次深度计算生成 depth1.txt,center1.txt，第二次深度计算生成 depth2.txt,center2.txt；

2. szMarkPath 文件夹存放标记点的深度计算结果 depth.txt 以及中心视角图 center.txt，命名格式遵循以下准则：

光栅片数为 1 时，不需要拍摄 mark 点，因此，可输入空值；

光栅片数为 2 时，拍摄一个 mark 点，生成文件命名为 depth_mark1.txt,center_mark1.txt；

光栅片数为 3 时，拍摄两个 mark 点：

第一个 mark 点生成文件命名为 depth_mark1.txt,center_mark1.txt;

第二个 mark 点生成文件命名为 depth_mark2.txt,center_mark2.txt;

其中上方的 mark 点标记为 1。

3. 缺陷信息

输出字符串遵循以下原则:

<缺陷所在层: 缺陷个数<缺陷信息>;缺陷所在层: 缺陷个数<缺陷信息>...>

详述:

缺陷所在层: 如果光栅片数为 1, 则缺陷所在层从上往下分别为 0, 1; 如果光栅片数为 2, 则缺陷所在层从上往下分别为 0,1,2,3; 如果光栅片数为 3, 则缺陷所在层从上往下分别为 0,1,2,3,4,5;

缺陷个数: 返回该层缺陷的个数;

缺陷信息: <缺陷 1 面积大小, 缺陷 1 长, 缺陷 1 宽, 缺陷 1X 坐标, 缺陷 1Y 坐标; 缺陷 2 面积大小, 缺陷 2 长, 缺陷 2 宽, 缺陷 2X 坐标, 缺陷 2Y 坐标; ...; 缺陷 N 面积大小, 缺陷 N 长, 缺陷 N 宽, 缺陷 NX 坐标, 缺陷 NY 坐标>

举例:

假设光栅片数为 1, 则缺陷信息分两部分, 第 0 层缺陷信息和第 1 层缺陷信息

<0:2<23.2,2.0,3.0,5.0,7.0;23.2,2.0,3.0,5.0,7.0>;1:3<23.2,2.0,3.0,5.0,7.0;23.2,2.0,3.0,5.0,7.0;23.2,2.0,3.0,5.0,7.0>>