

ROKAE 珞石

xVision

机器人视觉使用手册



人类生产生活的得力伙伴

文件名称: xVision 使用手册

单 位: 珞石（北京）科技有限公司

日 期: 2023-06

目录

目录	3
1 前言	7
1.1 适用版本	7
1.2 修订记录	7
2 产品简介	8
2.1 功能概述	8
2.2 功能特性	8
2.3 运行环境	8
2.4 硬件连接	8
2.5 软件部署	8
3 软件介绍	9
3.1 界面布局	9
3.2 快捷工具栏	9
3.3 流程编辑区	10
3.4 算子区	10
4 视觉算子介绍	11
4.1 采集	11
4.1.1 虚拟 2D 相机	11
4.1.2 图像显示	11

4.2 逻辑	12
4.2.1 协议拆分	12
4.2.2 分支	12
4.2.3 格式化	13
4.2.4 序列化	14
4.3 通信	16
4.3.1 网络配置	16
4.3.2 数据接收	16
4.3.3 数据发送	17
4.4 设备	18
4.4.1 虚拟 3D 相机	18
4.4.2 RS 相机	19
4.4.3 海康相机 (2d)	21
4.4.4 BX 相机	22
4.4.5 奥比相机	22
4.5 深度学习	23
4.5.1 深度学习检测	23
4.5.2 yolo 目标检测	24
4.5.3 yolact 实例分割	25
4.6 图像处理	27
4.6.1 颜色提取	27
4.6.2 颜色检测	28

4.6.3 颜色分割.....	29
4.6.4 形态学处理.....	30
4.6.5 彩色转灰度.....	32
4.6.6 深度补偿.....	32
4.7 定位 (相机)	34
4.7.1 无向物体定位.....	34
4.7.2 Marker 检测.....	35
4.7.3 有向物体定位.....	36
4.7.4 获取3D 位姿.....	37
4.8 定位 (图像)	39
4.8.1 卡尺工具.....	39
4.8.2 直线查找.....	41
4.8.3 圆查找.....	42
4.8.4 斑点检测.....	42
4.8.5 形状匹配.....	44
4.8.6 线线测量.....	46
4.8.7 点点测量.....	47
4.8.8 位置修正.....	48
4.9 定位 (机器人)	49
4.9.1 物体位姿.....	49
4.9.2 抓取位姿.....	50
4.9.3 抓取辅助.....	51

4.10 点云	53
4.10.1 点云加载.....	53
4.10.2 点云查看.....	54
4.10.3 深度图转点云.....	54
4.11 标定	56
4.11.1 手眼标定.....	56
4.11.2 N 点标定.....	56
4.11.3 标定转换.....	57
4.11.4 标定板标定.....	58
4.11.5 单点对位.....	60
4.11.6 抓取校正.....	60
5 案例介绍	62
5.1 相机标定	62
5.2 2D 视觉引导.....	66
5.3 颜色检测	74
5.4 3D 视觉引导流程.....	77
5.5 斑点检测、模板匹配、MARKER 定位.....	80
5.6 自启动功能	89
6 常见问题及处理.....	93

1 前言

1.1 适用版本

xVision V1.1.6;

xCore V1.7.3;

1.2 修订记录

版本号	修订内容	发布日期	作者
V0.1.0	● 初始版本	2021.01	ysn
V0.1.1~V0.1.11	● 基础版本	2021.03~2022.08	ysn,pch,lyn,wll
V1.0.0	● 根据当前算子状态更新、添加常用案例	2022.09.15	wll,hgq,zly
V1.1.1~V1.1.4	● 算子重构后更新	2023.02.15	cjl,lcr,hzx,wll
v1.1.6	● 算子 bug 修复、优化	2023.6.1	wll,cjl,lcr

2 产品简介

2.1 功能概述

xVision 是一款采用图形化编程的机器人视觉软件，主要用于机器人引导。

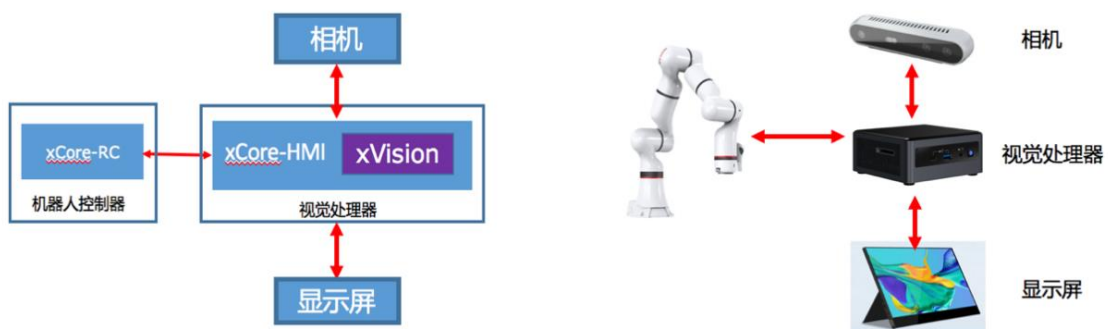
2.2 功能特性

- 与机器人运动编程统一的编辑运行环境，使用便捷；
- 利用算子连线自由搭建视觉解决方案，直观高效；
- 丰富的常用的 2D、3D 视觉算子，持续升级；
- 针对机器人引导场景的定制化算子，方便易用；

2.3 运行环境

参数	推荐配置
CPU	Intel i3 或以上
内存	4G 或更高
USB 接口	支持 USB3.0 或以上接口
操作系统	Win10 64bit

2.4 硬件连接



2.5 软件部署

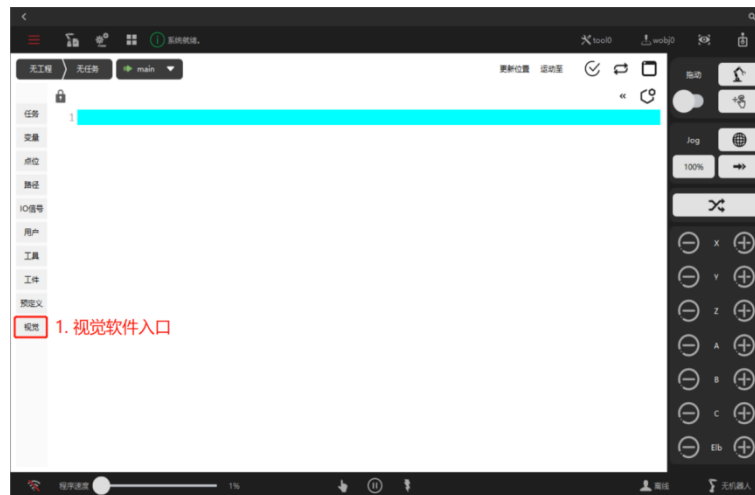
安装由珞石科技提供的 RobotAssist 软件，xVision 已深度整合其中。

3 软件介绍

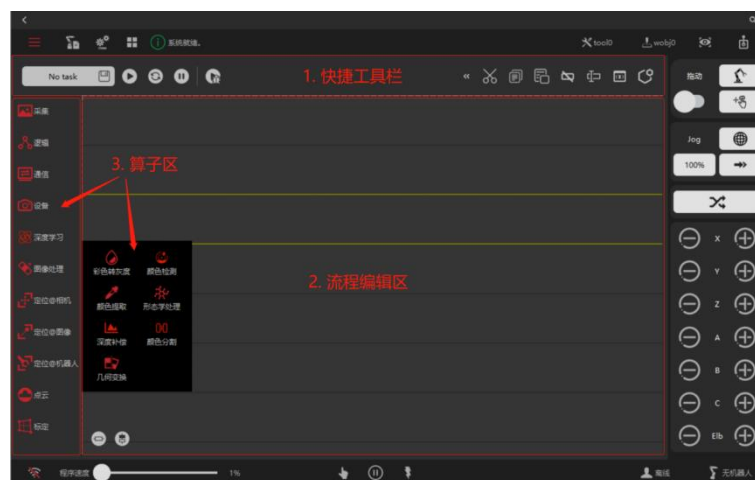
3.1 界面布局

点击左侧边栏的“视觉”进入 xVision 界面，由 3 个主要区域构成：

- 快捷工具栏：视觉任务执行快捷键及算子操作快捷键；
- 流程编辑区：视觉任务编辑区域；
- 算子区：视觉任务搭建所需的算子区域，算子按类型分类，可拖拽至流程编辑区，再连线建立算子拓扑关系，构建视觉任务；



xCore 主界面



xVision 界面

3.2 快捷工具栏



1~5 依次为：

- 保存视觉任务;
- 加载视觉任务, 并展示当前视觉任务名; (暂时只支持一个视觉任务)
- 单次运行;
- 连续运行;
- 中止运行;



6~9 依次为:

- 删除选中算子;
- 算子重命名;
- 打开 xVision 变量列表;
- 折叠算子区;

3.3 流程编辑区

从算子区选择合适的算子, 拖拽至流程编辑区, 通过连线确定算子间的拓扑关系, 构建视觉任务。

通过连线产生联系的算子们构成一张“图”, 算子是“图”的节点。一个算子可以把同一张“图”上另一个算子的输出作为自己的输入(前提是输入输出的类型相同, 软件会自动筛选), 也可以把自己的输出作为另一个算子的输入。

例如下图左, 各个算子构成一张“图”; 下图右, 各个算子构成两张“图”。下图右中, “图像显示”算子就无法获取“虚拟 2D 相机 1”算子的输出, 因为这俩不在一张图上; 而“图像显示 1”算子就可以获取“虚拟 2D 相机 1”算子的输出。



一张图



两张图

双击算子, 可以打开该算子的设置界面。

流程编辑区左下角有两个辅助算子排列整齐的按钮, 选中若干算子后单击按钮, 可以自动调整算子位置。

3.4 算子区

算子区包含数十个按照类型区分的算子, 持续完善中, 第 4 章详细介绍。

4 视觉算子介绍

4.1 采集

4.1.1 虚拟 2D 相机

➤ 功能

从指定路径读取 2D 图。



界面

➤ 参数

- 图像路径
- 打开 (打开并选择路径)

4.1.2 图像显示

➤ 功能

选择一个图像源，可以显示当前图像源对应的图像，可通过鼠标左键拖拽，滚轮缩放。



界面

4.2 逻辑

4.2.1 协议拆分

➤ 功能

将接收的数组信号拆分为独立元素。



The dialog box for the 'Protocol Splitting' operator. It has an 'Input' section with 'Input Source' set to '数据接收.meg1' and 'Delimiter' set to ','. Below is a table for output variables:

序号	变量名	输出类型	
0	out0	int	Delete
1	out1	int	Delete
2	out2	int	Delete

At the bottom of the table is an 'Add' button. At the bottom of the dialog are 'Confirm' and 'Cancel' buttons.

界面

➤ 参数

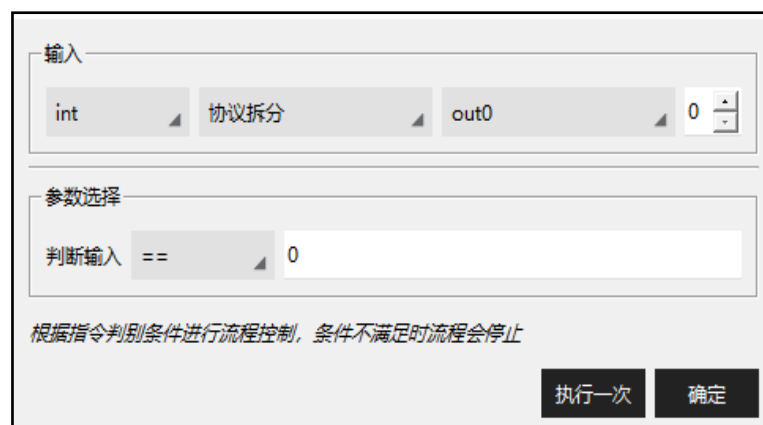
- 输入源：必须为 string 类型，一般以“数据接收”算子作为输入；
- 分隔符：用于分割输入源数组；
- 输出可根据需要任意增减，序号对应分割后元素的索引号；

注：配置好后需要点击“确定”，生成指定的输出变量。

4.2.2 分支

➤ 功能

可以根据指令判别条件进行流程控制，条件不满足时，流程会阻塞，常用于根据信号判断是否触发该分支所管理的流程。



The dialog box for the 'Branch' operator. It has an 'Input' section with 'int', '协议拆分', 'out0', and a value of '0'. Below is a 'Parameter Selection' section with '判断输入' set to '==' and a value of '0'. At the bottom is a note: '根据指令判别条件进行流程控制，条件不满足时流程会停止'. At the bottom right are 'Execute Once' and 'Confirm' buttons.

界面

➤ 参数

- 输入：分别指定输入类型、输入源算子、变量名、变量索引
- 判断输入：一般选择 “==”

➤ 样例

使用协议拆分、分支等搭建，按照接收信号的不同触发不同分支流程



样例

4.2.3 格式化

➤ 功能

可以组成任意格式，作为其余算子的输入，如输出形状匹配结果，以代替“形状匹配”算子作为“标定转化”算子的输入；也可以从各类输出中择取需要的格式，如制作 point 类型时，可以选择“形状匹配”结果、“圆查找”结果、“直线查找”结果等所有含 point 类型的输出。



界面

➤ 参数

- 输出类型：可选任意输出类型（int、double、point、pose、flag 等）

➤ 使用步骤

- 选择输出类型，算子将自动展示该类型需要的输入
- 依次选择各输入变量
- 对需要指定为常量的值，勾选对应的选择框，手动输入常量值（数字或数组，后者用“,” 隔开元素）

➤ 样例

使用该算子组建“形状匹配”结果，将某些量设为常量。



样例

4.2.4 序列化

➤ 功能

可以择取任意算子输出结果的变量，组成字符串。



界面

➤ 参数

- 添加变量/文本/结束符：选择添加类型
- 分隔符：各元素之间的分隔符

➤ 使用步骤

- 添加变量：点击按钮“+变量”，界面中增加一行组合框，第一个组合框选择添加的变量类型，可选 bool、int、string、double（point、pose 等均属于 double 类型），第二个组合框用于选择输入的算子，第三个组合框选择变量名称。自设值窗口用于指定变量的索引值——若该变量为非数组类型，则直接返回变量值；若该索引值不存在，返回 0
- 添加文本：点击按钮“+文本”，界面中增加一行文本输入栏，手动输入常量，也可输出符号
- 添加结束符：点击按钮“+结束符”，界面中增加一行组合框，选择结束符类型

⚠ 注：当变量为非数组类型时，无论索引值是多少，返回值均为该变量的值；
当没有索引值指定的元素时，返回值为-1

➤ 样例

使用该算子生成一组字符串



样例

4.3 通信

通信主要是网络通信相关的设置，以及数据接收、发送算子。

4.3.1 网络配置

➤ 功能

建立 TCP 连接，为其它通信算子提供网络套接字。在现阶段机器人应用中仅作为 TCP 服务端，机器人作为客户端进行连接，其它连接方式目前尚不支持，将根据后续应用进行适配。



注：

- 目前仅针对现阶段用法，支持 TCP 服务端协议以及短链接策略；
- 数据包分隔符使用回车符（ASCII 中的 0x0D，即 '\r'）



界面

➤ 参数

- 协议：包括 TCP 服务端、TCP 客户端、UDP，目前根据实际使用方法，仅支持 TCP 服务端监听端口进行单连接，也就是说，如果需要多个连接，需要多个算子配置不同端口；
- IP：TCP 服务端协议下表示本机 IP 地址，默认 0.0.0.0，多网卡需要指定网卡 IP 可以进行设置；
- 端口：服务器监听端口，算子拖拽后或加载后立即开始监听，默认 9797，所以多算子可能存在拖拽后监听失败，后续修改并保存即可；
- 策略：短连接表示每次运行根据实际连接状态通信，长连接表示连接断开即终止运行（尚不支持）；

4.3.2 数据接收

➤ 功能

将接收到的数据按照一定的格式和顺序转换成指定的变量。无连接、超时的情况下直接终结本次运行，数据未接受完全时会进行阻塞等待。



注：

- 对于目前应用格式仅支持逗号分隔符的字符串形式，有无数据包分隔符均可，数据接收是否完成的判断取决于逗号分隔内容总计是否达到收到指定类型变量总计需要的数据的个数；
- 接受数据变量名禁止为空、包含非法字符（.@\$%等）、重复，目前尚未报错，但是会影响使用；
- 运行时禁止修改接受数据，目前尚未报错处理；
- 并列多个数据接收可以同时分别处理相同的内容；

➤ 参数

- 格式：CSV，表示逗号分隔符，与目前机器人发送数据（SocketSendDouble）指令兼容；
- 增加接受数据：按照顺序增加一个指定类型的变量作为输出，后面连接的算子可以选择作为输入；
- 删除接收数据：减少输出变量，如果后面算子已经选择，再次运行时会报错；
- 输入：网络类型变量，来自于网络配置；
- 输出：接收数据，由用户指定变量类型与变量名，一般使用 string 类型，若使用其他类型则只能接收第一位；



样例

➤ 样例

接收：1,2,3,4\r5,6,7,8,9

输出：meg1={1,2,3,4}

4.3.3 数据发送

➤ 功能

将输入的变量依照相应网络配置进行发送。



界面

➤ 参数

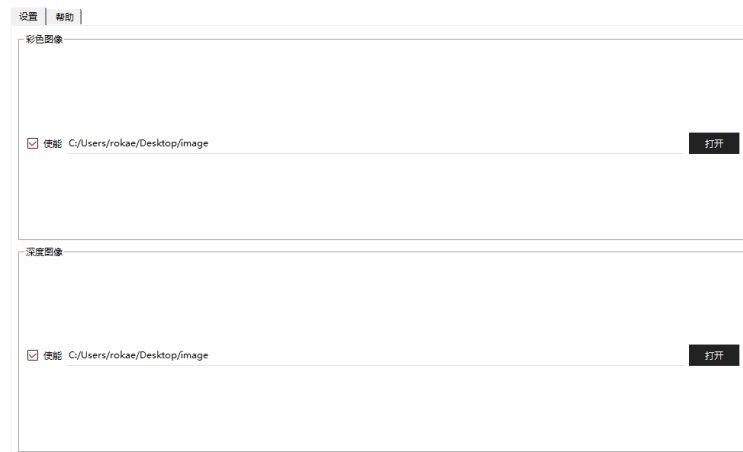
- 网络配置：网络的 IP 与端口号设置，通过网络配置算子获得
- 字符串：需要发送的数据，一般通过序列化算子将不同变量按一定规律组合在一起，注意需要在结尾加入结束符否则会导致通讯失败

4.4 设备

4.4.1 虚拟 3D 相机

➤ 功能

开发人员调试 debug，测试用，从指定文件夹读取 2D 图和深度图。



界面

➤ 参数

- 彩色图像路径
- 深度图像路径

➤ 使用步骤

- 使能设置（彩色和深度图是否选择）
- 打开（打开并选择路径）

4.4.2 RS 相机

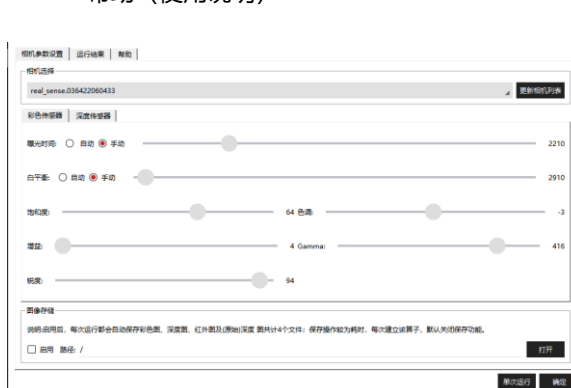
➤ 功能

作为 Realsense 415 相机的接口。

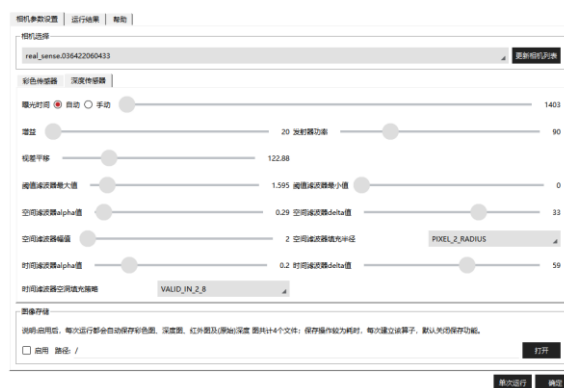
Realsense 415 是一款 3D 相机, 需要调节的参数主要针对 RGB 图像和深度图像。当相机与视觉处理器正常连接时, “相机选择”部分会自动检测到相机; 选择合适的相机, 点击“单次”或“连续”运行, 会采集并显示图像, 包括: 平面图(RGB 图像)、深度图、红外图。通过相机参数设置, 可以优化相机成像效果。

界面共 3 个 tab:

- 相机参数设置 (彩色和深度传感器, 相机列表选择, 图像存储: 自动保存彩色图或灰度图, 默认关闭)
- 运行结果 (3 个图片展示, 手动保存图片数据和相机内参)
- 帮助 (使用说明)



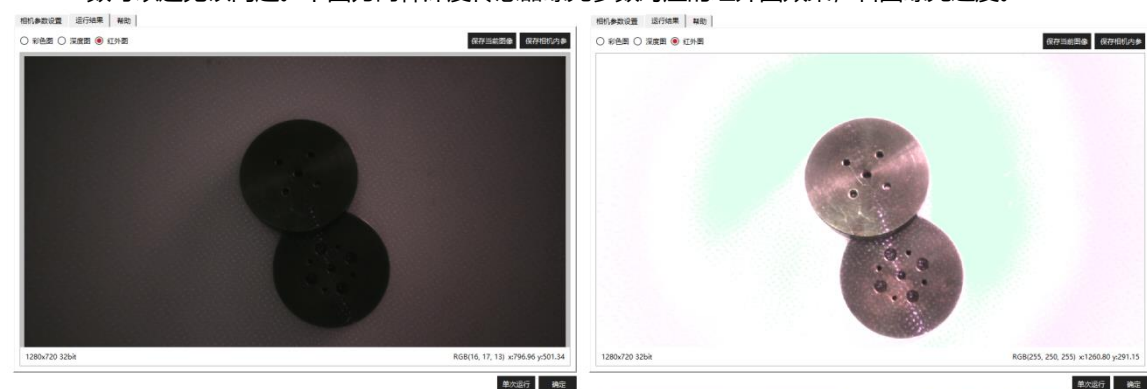
彩色传感器设置界面



深度传感器设置界面

➤ 参数

- Realsense 相机深度测量原理是通过传感器投射红外散斑, 然后根据两个红外相机成像中散斑的对应关系, 利用三角测距原理计算深度。深度测量相关常用的参数包括: 深度传感器曝光、增益, 发射器功率、视差平移等。
- 红外图过曝会影响深度测量, 形成没有有效数据的孔洞(深度图上的黑色部分), 通过调整深度传感器曝光参数可以避免该问题。下图为两种深度传感器曝光参数对应的红外图效果, 右图曝光过度。



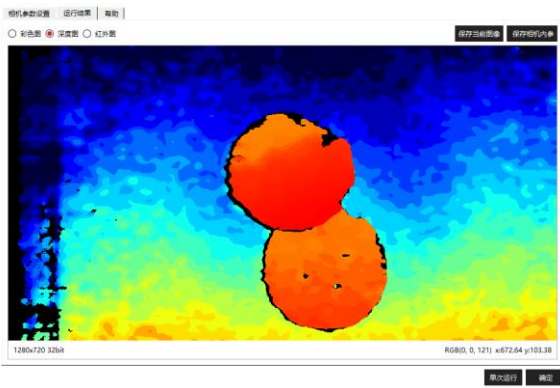
低曝光

高曝光

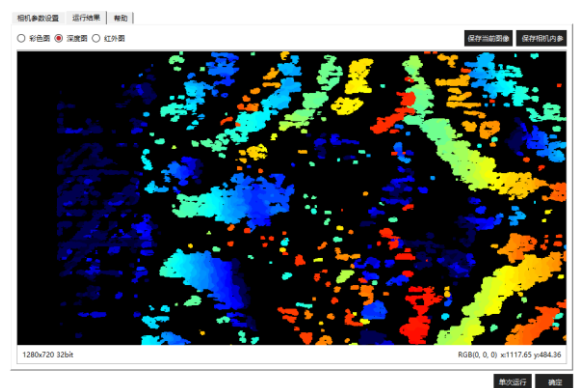
- 3D 相机的深度测量有景深(有效深度的测量范围), 景深可以通过视差平移参数来调整。例如当视差平移设置为 153 时, 景深为 18.5cm~30cm。越靠近物体, 视差平移参数需要调得越大, 相应的景深也越小。

⚠ 注: 当相机需要离被测物体比较近时, 视差平移需要设置得比较大, 此时深度图一侧的部分区域是不可用的(黑色), 要求用户选择合理的相机拍照位置, 让被测物体避开这部分区域。

- 根据深度图像判断视差平移参数是否合适。深度图像上，黑色部分的深度数据是无效的，彩色部分的不同颜色代表不同的深度值。粗调视差平移参数，直至深度图像能体现被测物的“形貌”，然后继续精细微调。

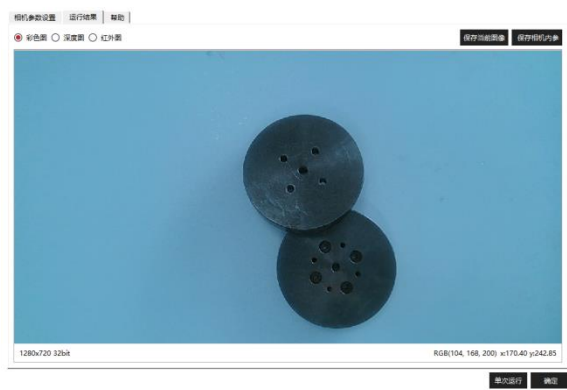


有效的深度图



无效的深度图

- RGB 图常用的参数包括：彩色传感器曝光、增益、白平衡。一般通过调节曝光参数，让被测物特征更明显；通过调节白平衡，让颜色更真实。

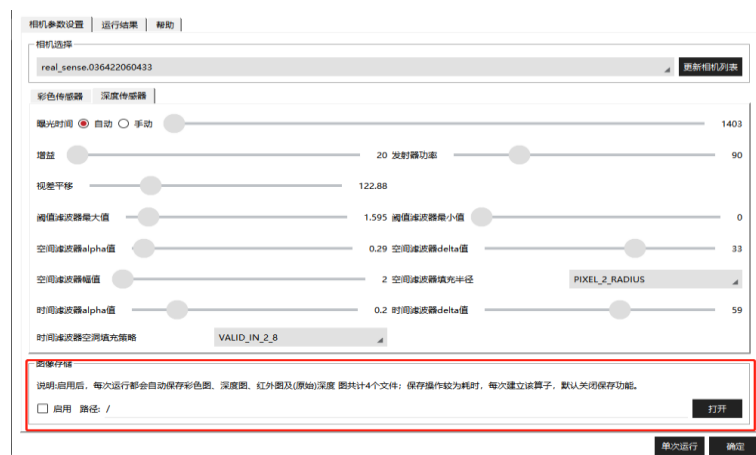


低白平衡



高白平衡

- 更多的高级参数，例如色彩、各类滤波器的参数需要对图像、点云处理有一定了解，如有需要可以联系厂家的客户支持。
- 相机采集的数据可以在运行时保存，便于分析。
- 点击“运行结果”界面上的“保存相机内参”按钮可保存相机初始内参。
- 点击“运行结果”界面上的“保存当前图像”按钮可同时保存相机当前抓取场景的彩色图、深度图及红外图。



自动保存图像

4.4.3 海康相机（2d）

➤ 功能

使用海康相机采集图片数据，可以调节关键相机参数（采集模式、曝光时间、增益、帧率）

海康相机机型介绍

CE 系列经济型

CE 系列是主打高性价比的经济型系列产品，像素覆盖面很广。以卷帘曝光为主，也有部分全局曝光。提供千兆网和 USB 3.0 数据接口，可以满足多种工业需求。

CA 系列进阶型

CA 系列是进阶型的系列产品，主打中低分辨率，高图像品质。相机大多为全局曝光相机，分辨率布局密集，可满足细分应用需求。同样具备网口和 U 口两种数据接口，能够满足高帧率，稳定输出高质量图像需求。主打 Sony 芯片程序质量优秀，Onsemi 芯片系列像元尺寸一致，无需调整镜头成像距离即可同像素精度下，实现视野升级。

CS 系列二代产品

本着精益求精的设计理念，海康机器人推出 CS 系列二代工业相机产品。系列产品从外观设计、产品开发到生产管控，都力求实现技术突破，给原有用户带来使用上的升级体验。同时更加入了丰富的进阶 ISP 功能，减轻用户图像处理的负担。

CH 系列高端型

CH 系列为高端型系列产品，针对 FPD 检测、3C、电子半导体、新能源等行业的高精尖应用开发，同时满足高分辨率、高帧率需求。产品覆盖 GigE、USB 3.0、10GigE、Camera Link、CoaXPress 多种数据接口。

界面共 3 个 tab：

- 相机参数设置（相机参数，相机列表选择，图像存储：自动保存彩色图或灰度图，默认关闭）
- 运行结果（图片展示，手动保存图片数据）
- 帮助（使用说明）

参数设置界面



运行结果界面

4.4.4 BX 相机

➤ 功能

作为 BX 相机接口。

BX 是一款 3D 相机，需要调节的参数主要针对 RGB 图像和深度图像。当相机与视觉处理器正常连接时，“相机选择”部分会自动检测到相机；选择合适的相机，点击“单次”或“连续”运行，会采集并显示图像，包括：平面图(RGB 图像)、深度图、红外图。通过相机参数设置，可以优化相机成像效果。

界面共 3 个 tab：

- 相机参数设置（相机列表选择，相机参数调节）
- 运行参数设置（图像存储：启用后自动保存彩色图、伪彩深度图、红外图以及原始深度图，默认关闭）
- 帮助（使用说明）



界面

4.4.5 奥比相机

➤ 功能

xVision 用户手册 v1.1.6

作为奥比中光 Gemini 2 相机接口。

Gemini 2 相机是一款 3D 相机,需要调节的参数主要针对 RGB 图像和深度图像。当相机与视觉处理器正常连接时,“相机选择”部分会自动检测到相机;选择合适的相机,点击“单次”或“连续”运行,会采集并显示图像,包括:平面图(RGB 图像)、深度图、红外图。通过相机参数设置,可以优化相机成像效果。

界面共 3 个 tab:

- 相机参数设置 (相机参数, 相机列表选择, 图像存储: 自动保存彩色图或灰度图, 默认关闭)
- 运行结果 (图片展示, 手动保存图片数据)
- 帮助 (使用说明)



界面

4.5 深度学习

4.5.1 深度学习检测

➤ 功能

深度学习是由传统神经网络发展的一类机器学习算法,让计算机像人一样在真实世界中吸收、学习和理解复杂的信息,完成高难度的识别任务。由于深度学习基于一定的数据基础,所以在使用前需要对大量的数据集进行训练,参与训练的数据集,需要进行标签的标注,要尽可能的保证数据的多样性(以苹果检测为例,需要至少 150 张图片样本)。

深度学习检测工具的输入为彩色图和深度图(用于对检测到的目标物进行排序,从而选取最优抓取目标),输出为包含最优抓取目标物体的二值图。其使用方式如下图所示。



界面

➤ 参数

- 彩色图：输入彩色图
- 深度图：输入深度图
- 目标标签：需要检测的目标物标签（字符串），如：目标标签为“apple:pear”或者“apple,pear”，则输出“apple”和“pear”类型的目标，如果想检测单一类型的目标，则输入“apple”或者“pear”即可，模型输出即为单一类型的目标
- 训练模型：使用深度学习训练平台训练好的模型
- Python 程序：用于模型预测及后处理的 python 程序
- 输出：图像中检测出来的目标 id 和类别名字
- 输出 best_object：只包含最优抓取物体的二值图

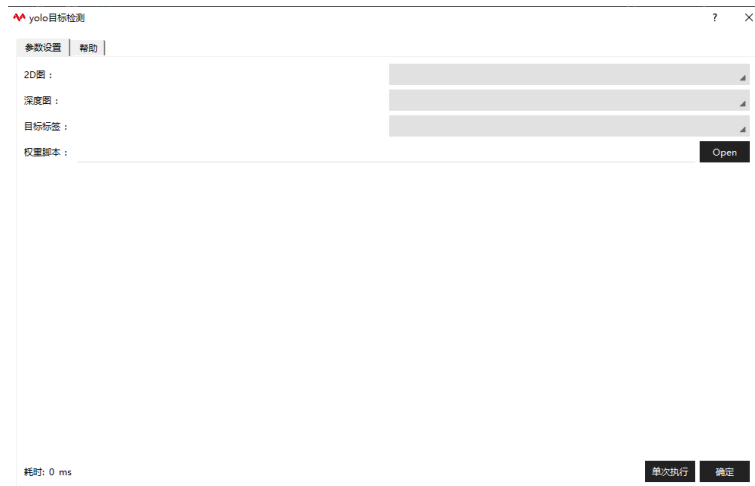
4.5.2 yolo 目标检测

➤ 功能

通过 yolo 目标检测算法（深度学习）检测图像中目标的位置并识别目标种类。其输入为彩色图和深度图，输出为带有检测结果的彩色图（将每个目标的检测框和类别的置信度画在输入的彩色图上）

➤ 参数

- 2D 图：输入彩色图
- 深度图：输入深度图
- 目标标签：需要检测的目标物标签（字符串），如：目标标签为“apple:pear”或者“apple,pear”，则输出“apple”和“pear”类型的目标，
- 权重脚本：yolo 神经网络训练得到的模型，后缀名为.onnx
- 输出：带有检测结果的 RGB 图像



界面

➤ 使用步骤

- 选择 2D 图源，通常为彩色图
- 选择深度图源
- 加载目标标签
- 指定权重文件路径

4.5.3 yolact 实例分割

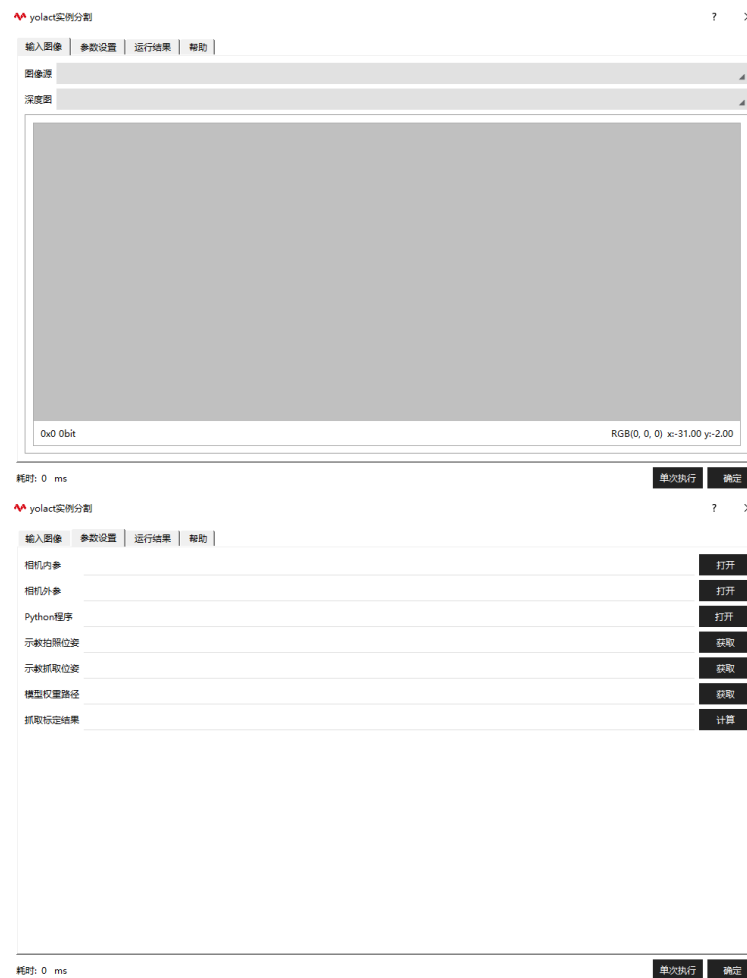
➤ 功能

基于 yolact 实例分割算法（深度学习）辅助抓取目标物体。

yolact 算法是单阶段的实例分割算法，其分割速度优于二阶段实例分割算法。本算子采用 yolact 神经网络分割出环境中的目标物体以得到掩膜图，结合上述无向物体定位、物体位姿、抓取位姿的算子功能，最后输出目标物体的抓取位姿。

➤ 参数

- 图像源
- 深度图
- 相机内参
- 相机外参
- Python 程序：深度学习推理脚本程序
- 示教拍照位姿：示教时拍照位机器人法兰位姿
- 示教抓取位姿：示教机器人抓取目标物体时法兰的位姿
- 模型权重文件：深度学习 onnx 格式模型文件
- 抓取标定结果：示教完成时得到的物体与法兰的相对位姿



➤ 使用步骤

- 选择图像源
- 选择深度图
- 选择相机内参路径
- 选择相机外参路径
- 选择 Python 深度学习推理脚本程序路径
- 模型权重文件：获取深度学习 onnx 模型权重文件

示教过程：

- 示教拍照位姿：获取示教拍照位姿法兰位姿
- 示教抓取位姿：获取示教抓取位姿法兰位姿
- 抓取标定结果：点击“全局运行”或“单次执行”获得通过抓取示教后得到的物体和法兰的相对位姿，点击“计算”按钮得到抓取标定结果
- 点击“确定”按钮，保存以上参数设置

运行过程：

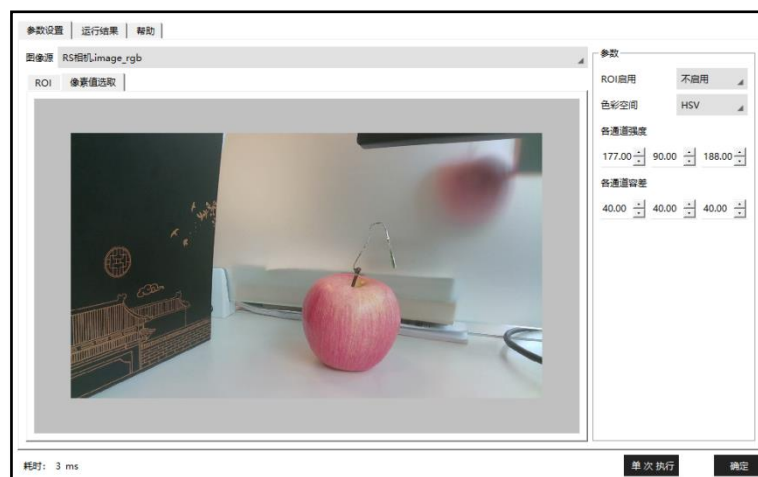
- 点击“全局运行”或“单次执行”，根据示教过程得到的抓取标定结果，获得运行时机器人法兰应达到的 6 自由度位姿。

4.6 图像处理

4.6.1 颜色提取

➤ 功能

颜色抽取用于从一张彩色图像上提取与指定颜色近似的部分，同时生成一张掩模(Mask)图。掩模图尺寸与原图一致，与指定颜色近似部分设置为白色，其余部分设置为黑色。抽取可以看成是一个二值化的过程，一般用于根据被测物的颜色特性进行物体分割。



界面

➤ 参数

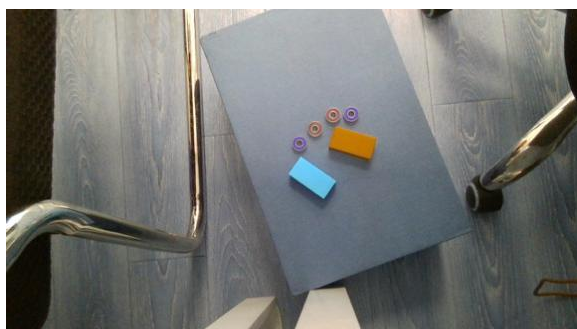
- ROI 启用
- 色彩空间 (可选 HSV 或 RGB 两种色彩空间)
- 各通道强度 (可通过手动输入或通过像素值选取界面读取像素点对应各通道强度)
- 允许的各通道容差阈值 (最终输出掩模图中白色区域在原图像中强度范围为参数设置中各通道强度 $\pm$ 阈值)

➤ 使用步骤

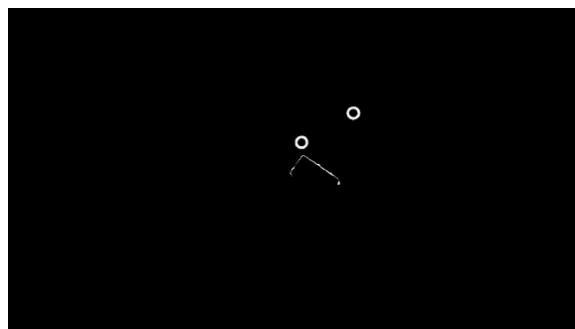
- 选择图像源，选择是否启用 ROI (默认不启用)，选择色彩空间 (HSV, RGB 一般使用 HSV 多)
- 点击“像素值选取”，然后在像素值选取下的图片上点击自己想要的颜色；会看到右边的各通道强度下的三个值发生变化，变化的值就是点击点的 HSV 三个值。三个值分别代表色调(H, Hue)、饱和度(S, Saturation)、亮度(V, Value)。RGB 同理
- 偏差阈值为三个通道允许偏差(相对指定颜色)阈值。(自己设定)
- 如果使用 ROI 启用后，可以通过  设置颜色抽取的矩形 ROI (Region of interest), ROI 内的像素参与计算，ROI 之外的像素在输出的掩模图中会被直接置为黑色
- 点击“确定”，参数设置生效；点击“单次执行”，按已生效参数允许一次

➤ 样例

希望在下左的图像中找到紫色的轴承，那点击指定的颜色，计算得到下右所示的掩模图。



RGB 图



Mask 图

4.6.2 颜色检测

➤ 功能:

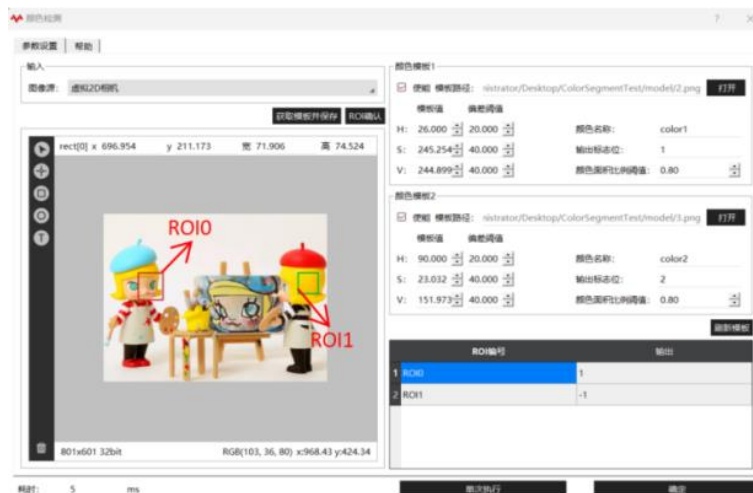
颜色检测可以设置若干种颜色模板(目前版本支持两种)、模板的标志位、以及若干个 ROI; 运行时检测各个 ROI 区域内的颜色与哪个模板最接近, 按照 ROI 定义次序输出对应的标志位。

➤ 参数:

颜色检测的参数跟颜色提取基本一致, 不同的参数包括:

- 启用: 确定是否启用该颜色模板参与检测
- 颜色占比: 某个颜色占据 ROI 面积百分比的阈值下限, 范围 0~1, 小于该阈值, 将被认为未检测到
- 输出标志: 用户可以自定义一个正数作为该模板的标志位, 当 ROI 内的颜色占比高于“颜色占比”阈值, 且比其他颜色模板的颜色占比更高时, 该 ROI 返回该标志位作为结果; 如果与所有模板都不符合, 会返回默认的标志位-1

支持选择若干的矩形 ROI, 按“确定”进行确认; 运行时, 各个 ROI 的颜色检测结果可以在结果输出栏查看。

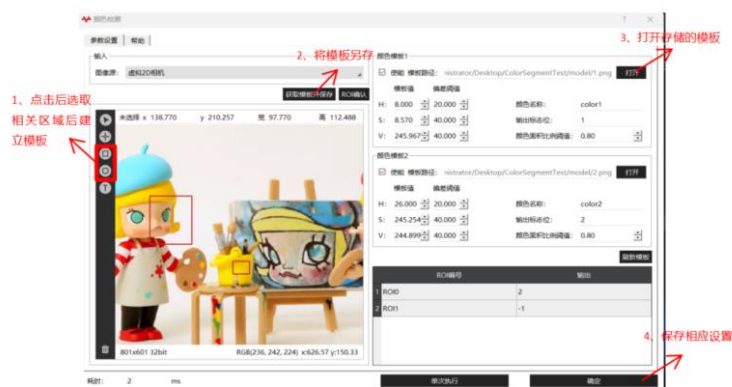


界面

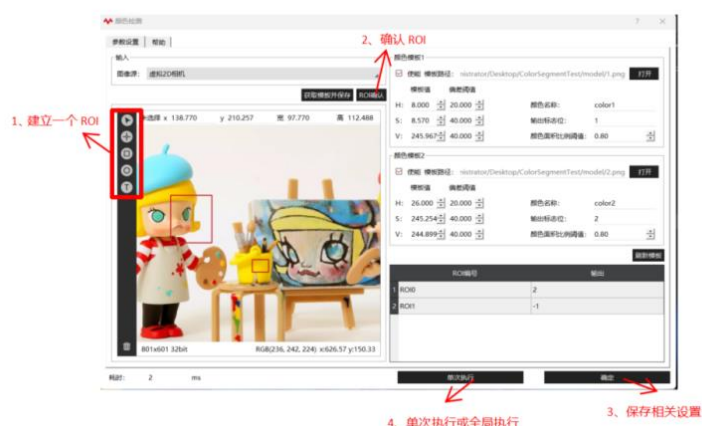
颜色检测的结果可以通过数据发送算子输出。上图示例有 2 个 ROI: 第 1 个 ROI 检测到黄色, 输出对应标志位 1; 第 2 个 ROI 未检测到白色或黄色, 输出默认标志位-1。(可通过双击选定的不同区域确定 ROI 序号, 双击选定框后对应的 ROI 将呈蓝色如上图所示)

➤ 使用步骤:

- 选择图像源, 然后全局运行一下可以见到图片显示
- 建立相关模板或手动输入 HSV 以及阈值等参数, 模板建立步骤如下:



- 颜色模板 1 和 2 选择（暂时支持两种模板，后续会修改），颜色名称、输出标志位、颜色面积比例，偏差阈值等设置
- 设置相关 ROI 区域进行颜色检测，步骤如下：



4.6.3 颜色分割

➤ 功能

颜色分割工具以物体颜色为模板进行图像分割，当不同类物体有着比较明显的颜色差异时可通过此工具实现准确的图像物体分割并输出只包含目标物体的二值图。

一类物体可以放入一个标签中，在完成建模以后可以调节模板参数，如下表所示。

颜色分割模板参数	
HSV 阈值	H 为色相，S 为饱和度，V 为亮度，具体含义参见百科。通过设置阈值限制物体与与标签模板相近的颜色。

➤ 参数

- 参数设置界面：图像源、目标标签
- 样本配置下生成的模板：样本标签、样本、HSV 阈值

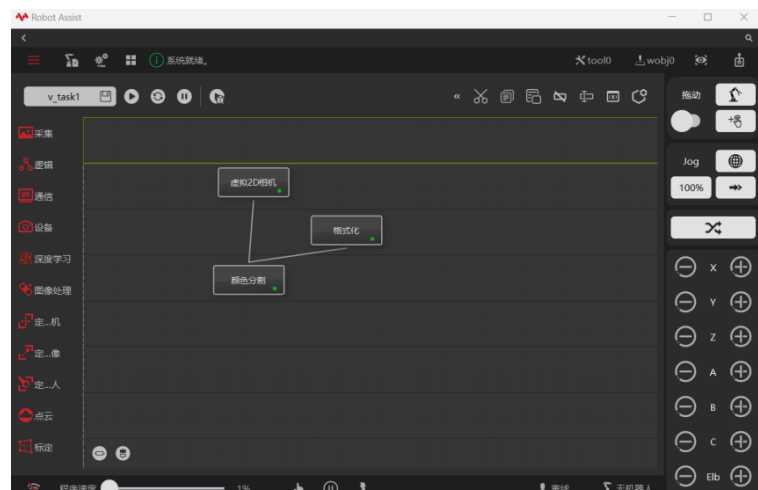
➤ 使用步骤

- 在参数设置界面选取图像源与目标标签，图像源可通过虚拟 2D 相机算子，或设备中相关相机算子获取，目标标签可通过数据接收算子或格式化算子获取，类型为 label；
- 在样本设置界面完成对于相关颜色模板的设置，步骤如下：



- 单击确认保存相关设置
- 点击执行或全局运行

➤ 样例



4.6.4 形态学处理

➤ 功能

形态学处理根据设定的滤波核和形态学方法，对输入的二值图像进行膨胀、腐蚀等形态学操作。

➤ 参数

- 形态学方法：包括腐蚀、膨胀、开运算、闭运算

膨胀：膨胀操作可以将白色区域扩散(黑色区域缩小)，多用于希望将白色像素连接在一起的情况，但是白色部分会变大；

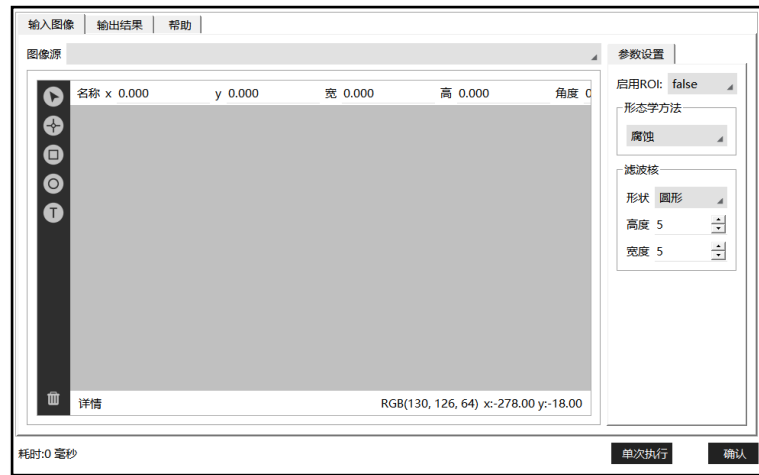
腐蚀：腐蚀与膨胀相反，可以将黑色区域扩散(白色区域缩小)；

闭运算：闭运算=先膨胀后腐蚀，可以将白色区域的断裂部分连通(黑色连通部分断开)，同时又不影响本身大小；

开运算：开运算=先腐蚀后膨胀，开运算与闭运算相反。

- 滤波核包括：矩形、椭圆、菱形；可以通过宽度和高度设置滤波核大小

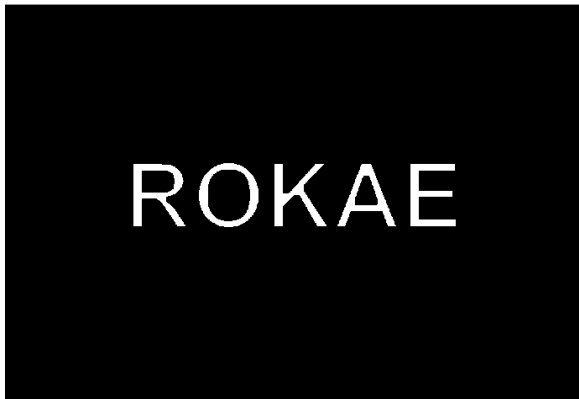
- 是否启用 ROI：仅处理感兴趣区域



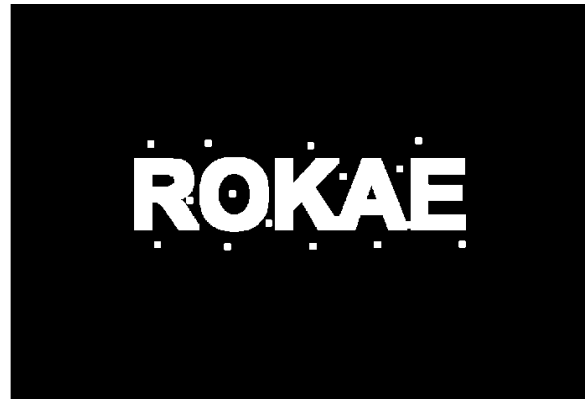
➤ 样例



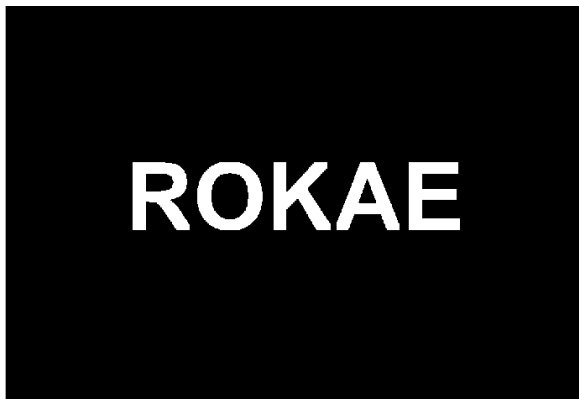
原图



腐蚀



膨胀



开运算

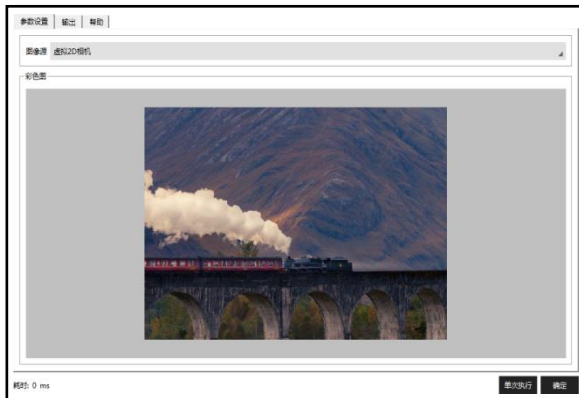


闭运算

4.6.5 彩色转灰度

➤ 功能

将输入的彩色图像转为灰度图像并输出。



彩色图



灰度图

4.6.6 深度补偿

➤ 功能

消费级 3D 相机测量的深度值局部误差较大,对于深度精度要求较高的场合,可以通过深度补偿的方式予以提升。深度补偿算子的原理是通过测量一个标准平面,确定相机深度的测量误差,在后续的处理中对该误差予以补偿。

➤ 参数

- 深度图：通常来源于 3D 相机 (RS 相机、BX 相机、奥比相机)
- 补偿文件：由补偿计算阶段训练的补偿文件



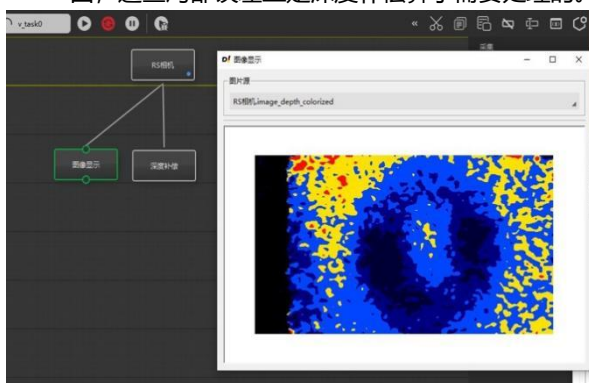
➤ 使用步骤

该算子的使用分为两个阶段：(1)补偿计算阶段；(2)补偿处理阶段。

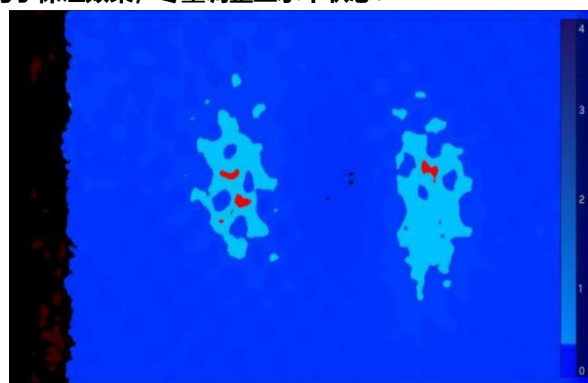
补偿计算阶段：

- 找一个平板(尽可能平、最好不是黑色、且不反光),大小覆盖相机视野范围；
- 将该平板平放于摆放工件的平面上(平板与平面平行)；

- 调节相机与平板平行，判断是否平行可以通过 xVision 图像显示算子(下图左)或者相机厂家提供的相机调试工具(下图右)观察深度图。下图就是一个典型的水平状态：平板各个位置深度值基本一样，中间布局会出现内凹或外凸，这些局部误差正是深度补偿算子需要处理的。**为了保证效果，尽量调整至水平状态！**



补偿前平面深度图效果



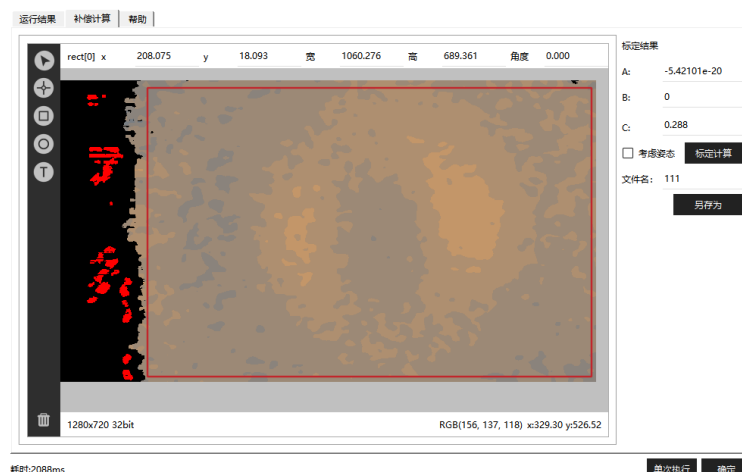
深度误差主要区域

调节完成后，在深度补偿算子的“运行结果”页面选择深度图来源；

在算子的“补偿计算”页面设置矩形 ROI 区域，后续只对 ROI 范围内的数据做计算和补偿处理；“考虑姿态”默认不勾选，勾选后，后续补偿会考虑后面将介绍的 A 和 B 参数。

ROI 设置完成后，点“标定计算”，根据处理器性能不同，计算可能会耗时 1~6 秒，完成后会显示结果 A、B、C，这是拟合出来的平板所在平面参数，平面方程为 $Z = Ax + By + C$ 。A、B 越小(最好小于 $10E-6$ 量级)，说明相机与平面越平行；C 代表相机与平面的距离，单位 m。如果觉得这三个参数比较合理，可以点“另存”，存储一个格式为 pfm 的误差补偿文件。

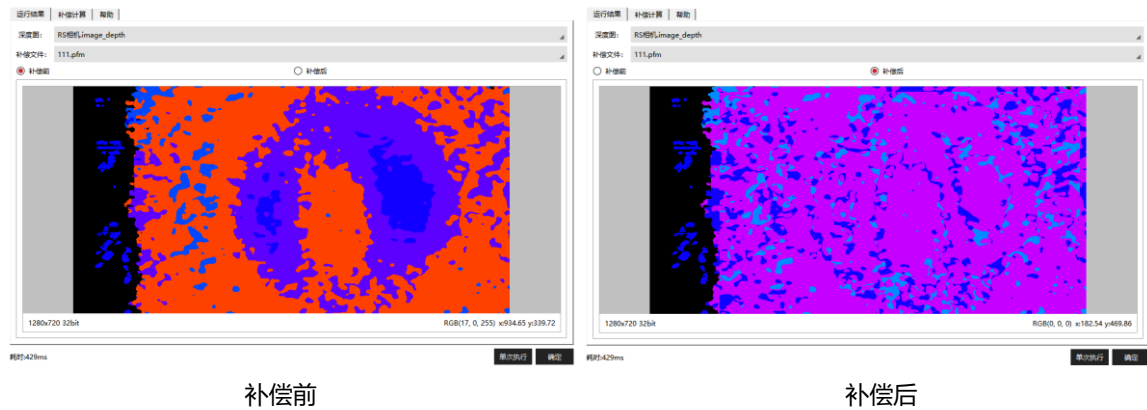
注意：“另存为”按钮会将误差补偿文件以用户给定文件名另存到相对路径./depth_refine_model 文件夹下，如果没有“另存为”，误差补偿文件会默认覆盖 refine_model.pfm 这个默认文件，当有多个深度补偿算子时，可能会对其他深度补偿算子计算造成影响；所以建议用户使用另存功能。



补偿处理阶段：

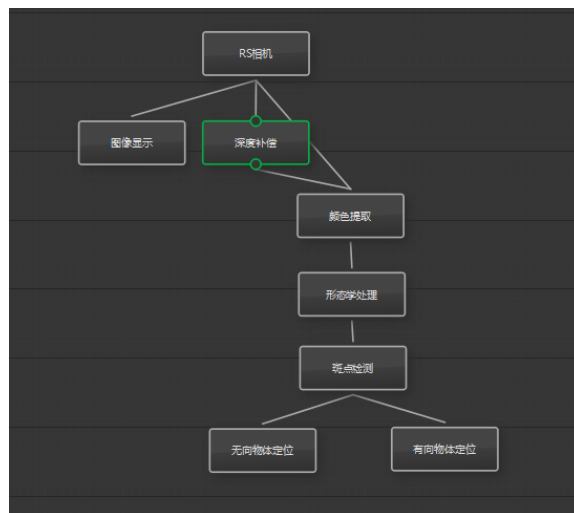
切换到“运行结果”页面，选择深度图来源及补偿文件。

点击“单次执行”按钮，可以单次执行补偿，并查看对比补偿前、后的效果。



➤ 样例

典型用法如下，由深度补偿算子对输入的相机的深度图进行补偿，输出一个补偿后的深度图，后续涉及深度的处理可以基于补偿后的深度图。



样例

4.7 定位（相机）

4.7.1 无向物体定位

➤ 功能

用于计算目标(各向同性的目标，例如圆形)相对相机坐标系的 6D 位姿。



界面



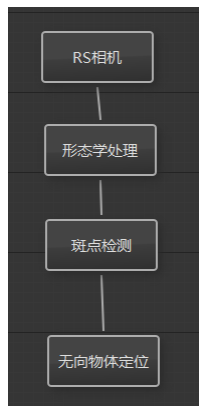
注:

- 不考虑物体方向时，可用于任意形状物体的定位。
- 当视野内存在多个目标物体时，只输出深度值最小的单个物体位置。

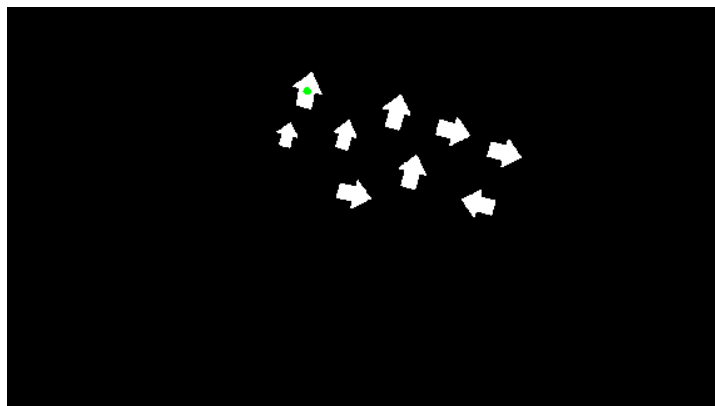
➤ 参数

- 图像源：输入二值图，通常来自于斑点检测、形态学运算等算子；
- 深度图：通常来自于相机；
- 相机内参：标定得到的相机内参文件。

➤ 样例



流程图



结果图

4.7.2 Marker 检测

➤ 功能

用于计算 April tag 类型的二维码相对相机坐标系的位姿。



参数设置界面



运行结果界面

➤ 参数

- 输入图像源：通常是来自相机的 RGB 图像
- 相机内参文件：RS 相机保存的内参，或手眼标定结果
- Marker 类型：April tag 的系列号
- Marker 尺寸：二维码实物的边长，单位 mm
- 允许 XY 旋转：不勾选时认为 XY 方向旋转为 0，二维码与相机平行

➤ 使用步骤

- 设置输入源、内参文件，以及 Marker 的参数，点击“单次执行”或“全局执行”
- 运行结果中，展示了 Marker 的位姿参数，窗口中显示了 Marker 的位姿朝向

4.7.3 有向物体定位

➤ 功能

用于计算有向物体(例如螺钉)相对相机坐标系的位姿。

本算子可计算掩模图像中各斑点的最小包络矩形，通过长度阈值上限和下限过滤掉不想要的斑点。此处的长度阈值指图像上物体的包络框长度（单位为像素）。尖端检测范围为相对值（取值在 0~0.5 之间）。

物体的方向默认根据最小包络矩形两端像素分布来确定，当“使用 PCA”置为启用时，物体的方向计算会进行主成分分析，部分情况下方向计算会更准确。例如下图，最小包络矩形的方向(中间绿线)与物体方向并不一致，使用 PCA 后，方向会更准确(中间紫色)。

尖端检测范围确定沿包络框长度方向，取多少比例的区域用于计算像素分布。



界面

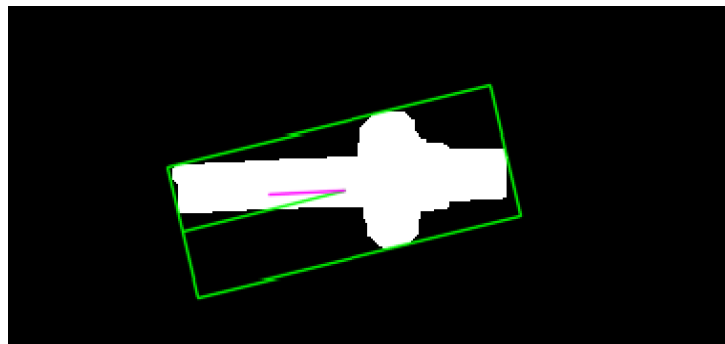
➤ 参数

- 灰度图：输入的掩模图，通常来自于斑点检测、形态学运算等算子；
- 深度图：通常来自于相机；
- 长度阈值：包含物体的长度上限与下限（单位为像素）；
- 尖端监测范围：取值范围在 0~0.5；
- 相机内参：标定得到的相机内参文件；
- 使能 PCA：部分情况下，若使能 PCA，方向计算会更准确（粉色线为使能 PCA 后的计算的方向）。

➤ 样例



流程图



结果图

4.7.4 获取 3D 位姿

➤ 功能

形状匹配只能获取物体的平面位姿(X、Y、Rz)，为了获取完整的六自由度位姿(X、Y、Z、Rx、Ry、Rz)，可以通过获取 3D 位姿算子进一步计算物体剩余的位姿数据。为消除欧拉角的歧义性，算子以输出四元数 Q1、Q2、Q3、Q4 代替欧拉角 Rx、Ry、Rz。

获取 3D 位姿算子的输入是“形状匹配”算子的输出，可前置多个“形状匹配”算子进行选择。



参数配置界面



运行结果界面

➤ 参数

- 图像源：输入 2D 图；
- 深度图：输入深度图，通常来源于相机；
- 相机内参：标定得到的相机内参文件；
- 排序方式：确定选择哪个匹配结果进一步计算 3D 位姿，选择依据可以是：第一个匹配结果或最高匹配分数；
- 测量半径：以选择的匹配结果的坐标(x、y)为圆心，以该参数为半径，采集该圆范围内的深度信息，计算目标点的深度；
- 抓取方式：“平面抓取”只考虑工件在 Rz 方向的旋转，“三维抓取”同时考虑了工件在 Rx, Ry, Rz 方向的旋转。
- 开启畸变校正：对精度要求高的场景，可以开启畸变校正，进一步提升目标定位精度。

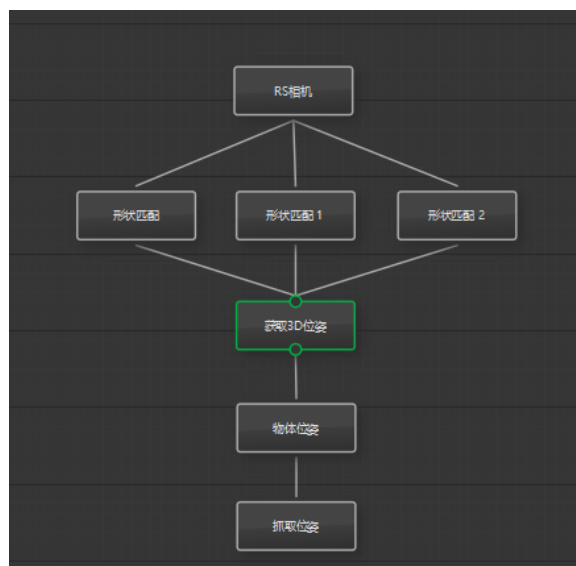
➤ 使用步骤

- 选取匹配结果：算子显示所有前置形状匹配算子，供用户选择使能
- 设置图像源、深度图源
- 设置内参文件、排序方法、测量半径、抓取方式及畸变校正使能；
- 点击“确定”按钮，保存以上设置
- 点击“单次执行”按钮，可单次执行算子，显示结果；



界面

➤ 样例



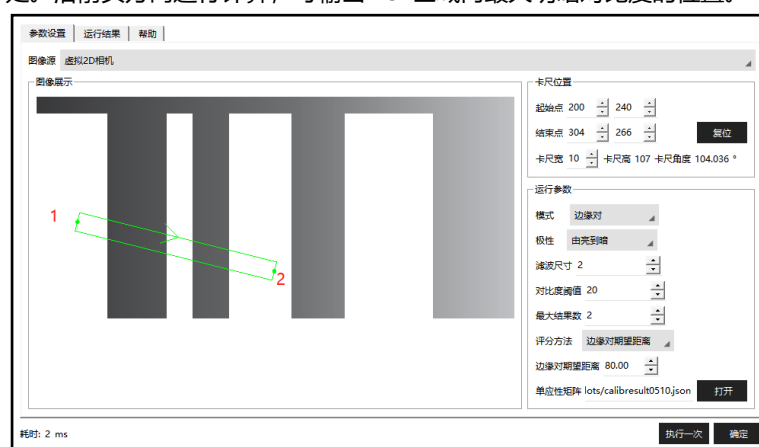
样例

4.8 定位（图像）

4.8.1 卡尺工具

➤ 功能

用于确定明暗交界处。沿箭头方向进行计算，可输出 ROI 区域内最大明暗对比度的位置。



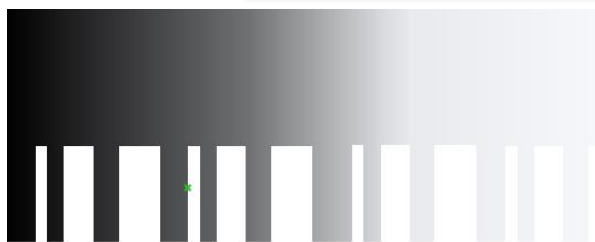
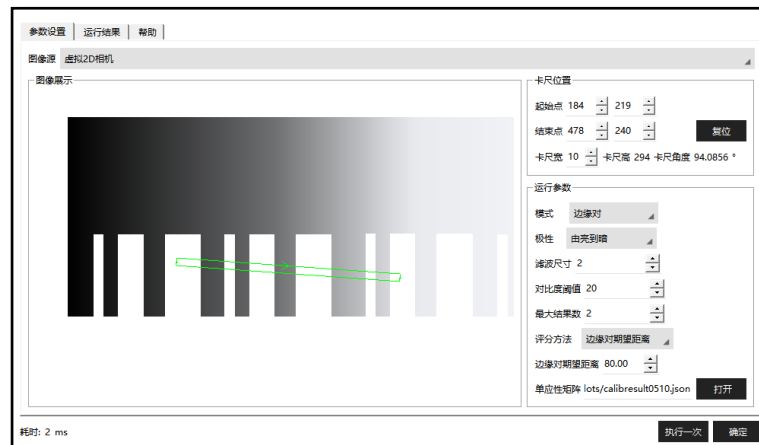
界面

➤ 参数

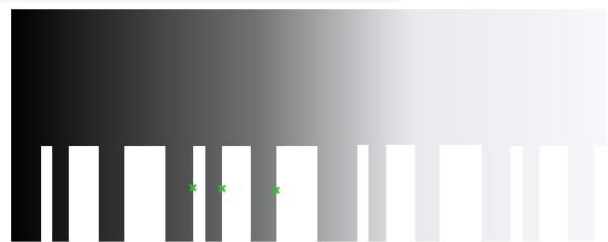
- 起始点、结束点：图中点 1、点 2。选取方法：使用鼠标在图片点击，第一次点击的位置自动记录为起始点，第二次为结束点。继续点击可修改两点位置。起始点到结束点的方向为搜索方向（箭头方向），用于计算对比度。
- 卡尺宽：卡尺宽度（箭头方向的垂直方向），用于计算平均灰度值。应避免宽度过大，对边界位置准确度的影响。
- “复位”按钮：恢复上一次“确定”或“执行”的卡尺位置和尺寸参数
- 模式：单边缘或边缘对。单边缘为单个边缘点；边缘对为一个“由亮到暗”和一个“由暗到亮”组成一对边缘对
- 极性：由暗到亮、由亮到暗，或忽略。
- 滤波尺寸：计算沿搜索方向的灰度值变化曲线，将曲线做滤波。滤波尺寸应为由黑到白经历的像素个数，若尺寸过大曲线将过于平滑，无法定位最大对比度的边缘点。

- 对比度阈值：最小对比度，找到的边缘的对比度值需高于此值
- 最大结果数：最多找到单边缘的个数或边缘对的个数。根据“评分方法”对结果进行排序，取前 n 名
- 评分方法：最大对比度、搜索方向，边缘对期望距离。当评分方法为“边缘对期望距离”时，结果中展示的分数为实际距离与期望距离的比值。当评分方法为其他时，分数为边缘对比度/255。
- 最大对比度：找到对比度最大的若干点
- 搜索方向：沿搜索方向找边缘点，距离起始点最近的边缘点最先被找到。
- 边缘对期望距离：当实际边缘对有固定距离时可以使用，需要输入单应性矩阵，本算子将根据边缘对图像上的宽度来计算实际宽度，宽度越接近期望距离得分越高，宽度大于 2 倍的期望距离将被淘汰。

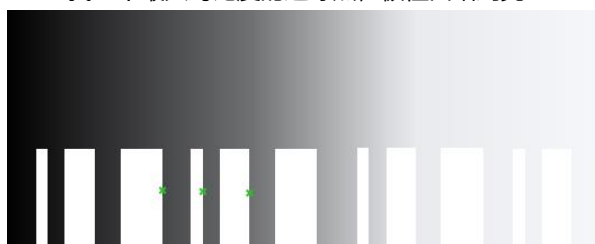
➤ 样例



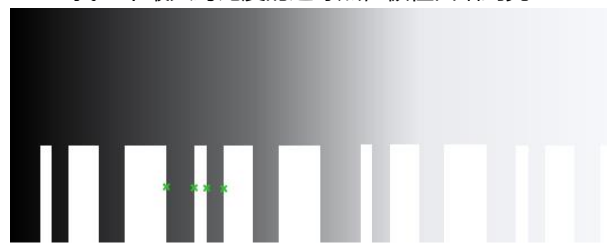
找 1 个最大对比度的边缘点，极性由暗到亮



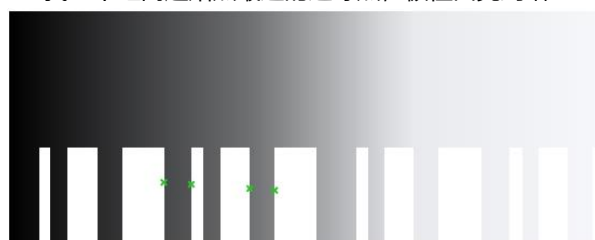
找 3 个最大对比度的边缘点，极性由暗到亮



找 3 个距离起始点最近的边缘点，极性由亮到暗



找 2 组对比度最大的边缘对，极性由亮到暗

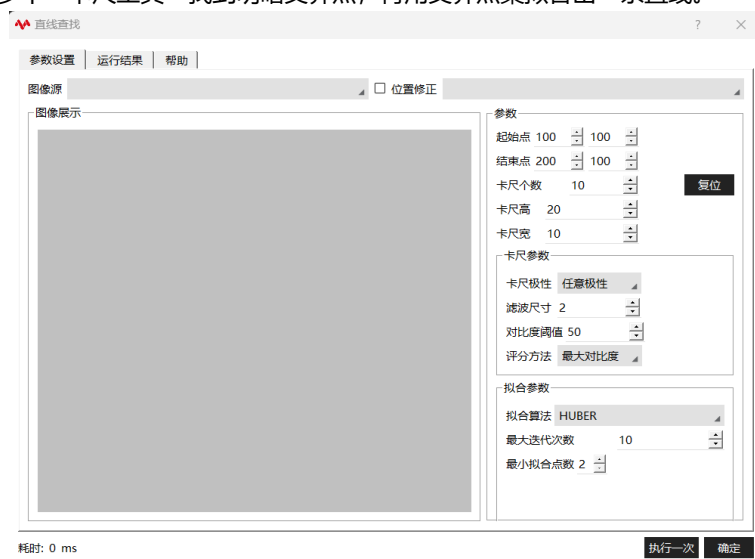


找 2 组期望距离最接近 20 的边缘对，极性由暗到亮

4.8.2 直线查找

➤ 功能

本算子相当于先用多个“卡尺工具”找到明暗交界点，再用交界点集拟合出一条直线。

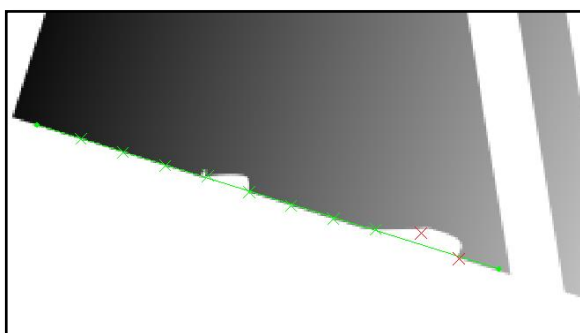


界面

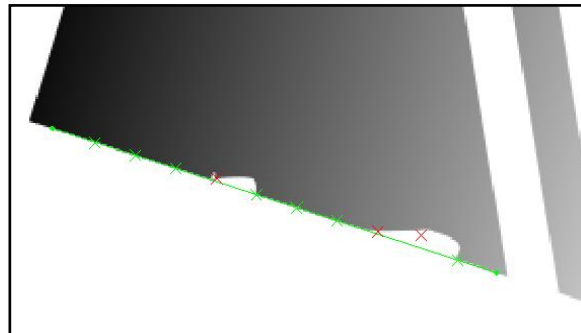
➤ 参数

- 位置修正：当卡尺需要随不同图像变换位置时，可通过与位置修正算子连接，并使能位置修正实现卡尺位置的变换
- 起始点、结束点、卡尺高、卡尺宽、卡尺极性、滤波尺寸、对比度阈值、评分方法与“卡尺工具”中相同
- 拟合算法：最小二乘：采用所有卡尺找到的边缘点进行拟合；RANSAC：随机采用部分点进行拟合，使拟合误差最小。误差过大的点被淘汰。HUBER 和 TUKEY：原理和最小二乘相同，只是对各点的权重进行重新分配，也就是说，误差越大的点对直线位置的影响越小。对误差过大的点，HUBER 削弱点的权重，TUKEY 直接将该点权重置零。也就是 TUKEY 算法受离群点的影响比 HUBER 小。若离群点较多，用 RANSAC 或 TUKEY。
- 剔除方法（使用最小二乘算法时可设置）：按点数剔除：按照误差由小到大对点进行排序，剔除最后 n 名；按距离剔除：误差高于该值的所有点进都被剔除
- 最小拟合点数：用于拟合直线的最小点数，若保留点（除了被剔除的点）低于该数则拟合失败，算子状态为 ERROR
- 误差阈值（使用 HUBER 算法时可设置）：拟合结果的误差超过该阈值，则拟合失败，算子 $flag=-1$

➤ 样例



拟合算法为最小二乘，剔除 2 个点
绿色为保留点，红色为剔除点



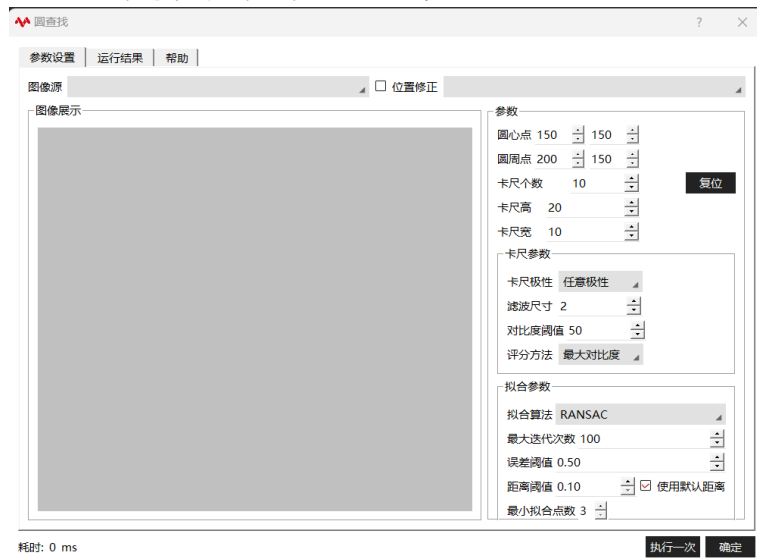
拟合方法为 RANSAC
可以看到自动剔除了 3 个误差最大的点

4.8.3 圆查找

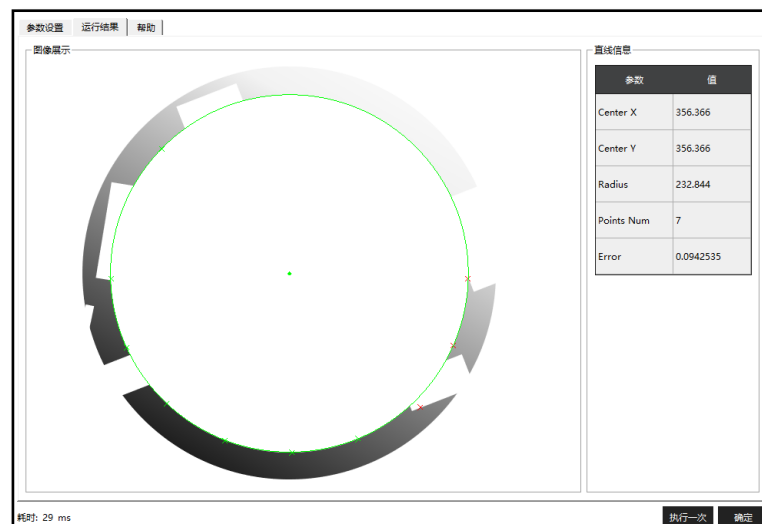
➤ 功能

本算子相当于先用多个“卡尺工具”找到明暗交界点，再用交界点集拟合出圆。

所有参数可参考“直线查找”算子，拟合算法少了 HUBER 和 TUKEY。



界面

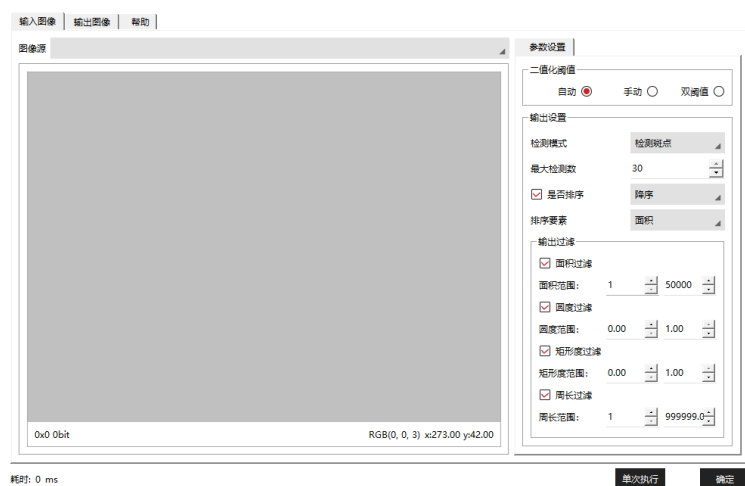


结果

4.8.4 斑点检测

➤ 功能

斑点检测主要用于检测和定位图像中指定灰度范围内、形状通常不可测的区域。使用斑点检测可以对图像中是否有斑点，以及斑点数量、位置、形状等特征进行定位。通过对斑点进行面积过滤、圆度过滤、矩形度过滤、周长过滤可以对检测到的斑点进行筛选，从而得到预期的斑点。利用得到的检测斑点，可以用作后续其他算子粗定位或零件表面缺陷检测。

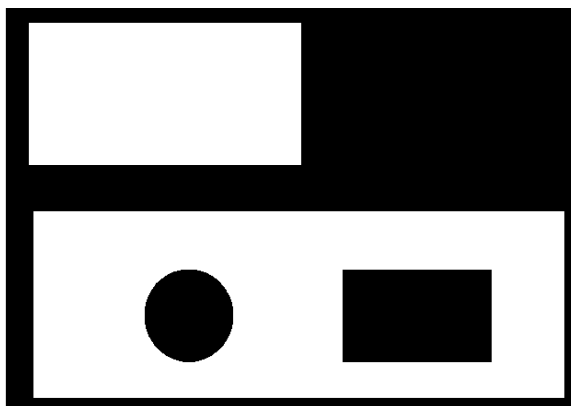


界面

参数

- 图像源：待检测图像，可以为灰度图或彩色图；
- 二值化阈值：包含自动选取阈值，手动选取单阈值，手动选取双阈值等；
- 最大检测数：设置检测到的斑点数量上限；
- 检测模式：检测斑点、检测孔洞或全部检测；（所谓孔洞、斑点，其实是两种相对的概念，举个例子：白纸上放个黑块，就认为黑块是个斑点；黑纸上放个白块，就认为白块是个孔洞）；
- 排序：当使能排序后，可以对检测到的斑点结果进行排序，可选择升序或降序；
- 排序要素：排序依据，支持根据面积、周长、质心 X、质心 Y、圆度、矩形度进行排序；
- 面积过滤：通过面积过滤去除不想要的斑点结果，面积指的是像素面积；
- 圆度过滤：通过圆度过滤去除不想要的斑点结果，圆度越接近 1，相应的斑点越近似于标准的圆；
- 矩形度过滤：通过矩形度过滤去除不想要的斑点结果，矩形度越接近 1，相应的斑点越近似于标准的矩形；
- 周长过滤：通过周长过滤去除不想要的斑点结果，周长指斑点或孔洞的边缘像素长度；

样例



原图

对原图设置不同的检测模式，效果如下：

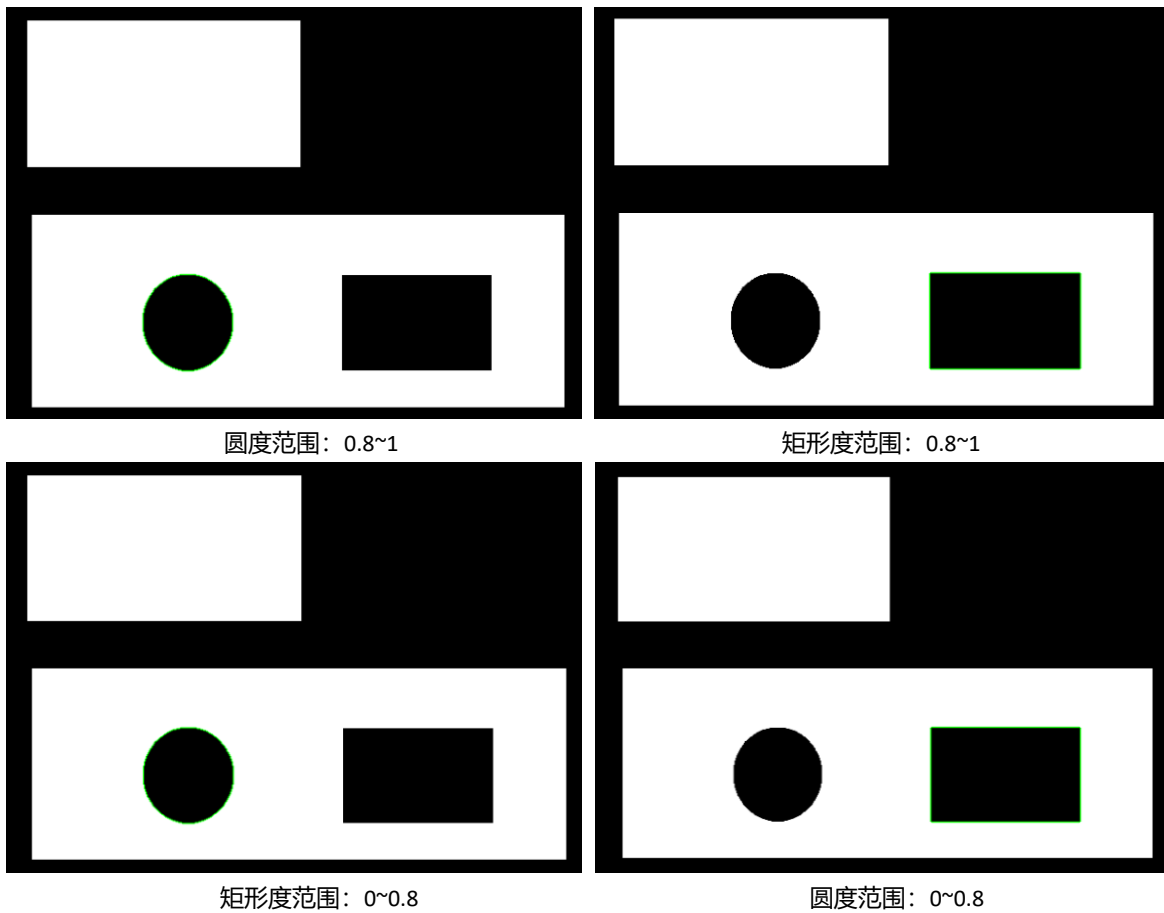


全部检测

检测斑点

检测孔洞

在斑点检测的基础上，结合圆度和矩形度过滤，进行斑点筛选，效果如下：



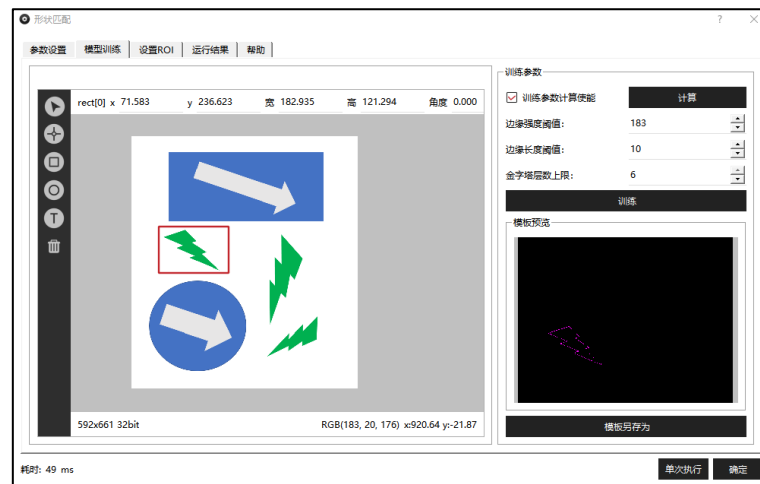
4.8.5 形状匹配

➤ 功能

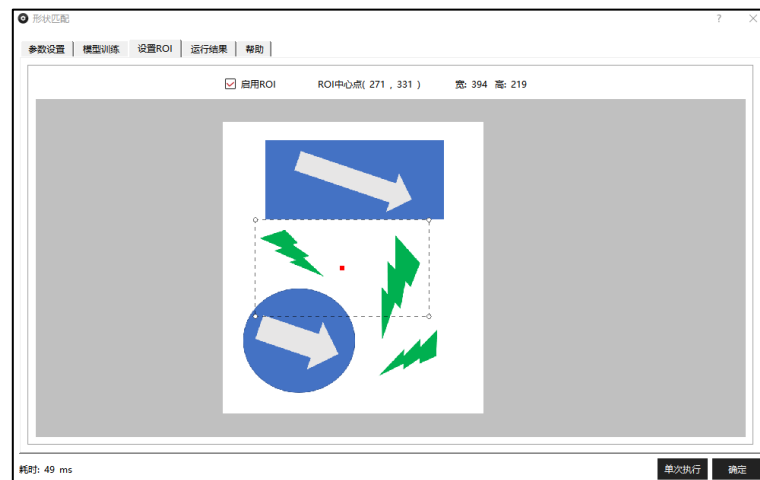
在目标检测图中寻找与模板相同图案的区域，以定位产品在图像中的 X、Y，和相对模板的旋转 R_z 。



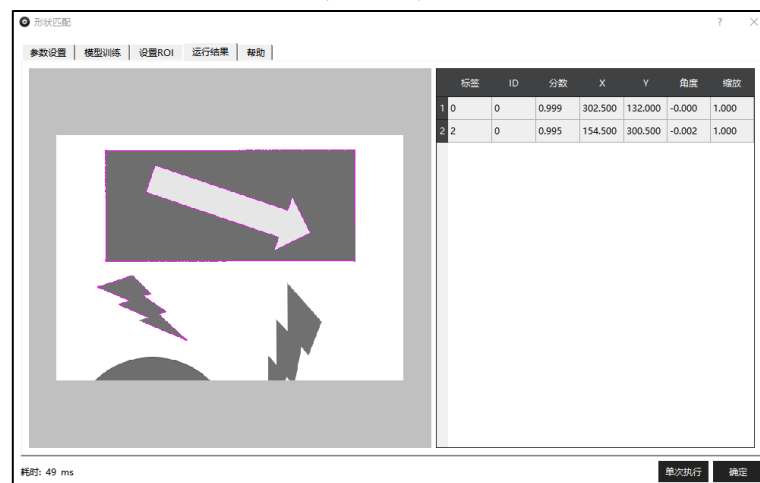
参数设置界面



模板训练界面



设置 ROI 界面



运行结果界面

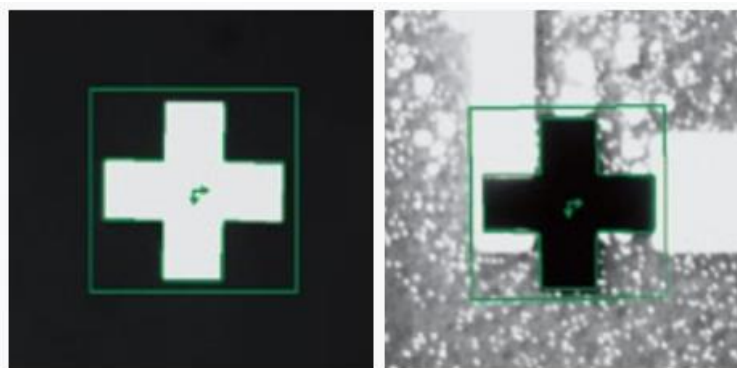
➤ 参数

参数设置：

- 输入图像源：必须是灰度图，可用“彩色转灰度”算子结果作为输入
- 关联模板：点击“+”按钮可以添加模板，“标签”为检测到的图像所符合的标签
- 搜索全部：复选框勾选时，对全部模板进行搜索；否则仅搜索第一个模板
- 角度范围：目标区域允许旋转的角度（相对于模板）
- 尺度范围：目标区域允许缩放的尺度（相对于模板）

- 最低匹配分数：目标区域最小得分，小于此分数则被淘汰
- 最大匹配个数：允许找到的最多目标
- 使能高级参数：
 - 精度选择：可以选择像素精度、亚像素一般精度、亚像素高精度，精度依次增加，相应的计算耗时也更长
 - 弹性参数：针对物体有轻微变形的场景，一般不做调整
 - 最大重叠率：有些场景下，部分目标会有互相重叠的现象，该参数用于判定两个目标重叠的阈值，高于该值的两个目标会被舍弃一个
 - 极性：不忽略、忽略全局、忽略局部；

（所谓极性，指的是边缘明暗对比，例如把黑色目标置于白色背景，与白色目标置于黑色背景，极性就是相反的，下图为两种边缘极性示意）。模板匹配的原理是通过形状确定目标与模板的相似度，一些场景下，当目标与模板的边缘极性发生改变时，可以通过修改该参数，依然能够实现匹配；



极性相反的两个模板

- 结果排序：通过分数、x 坐标、y 坐标的一个对结果进行排序

模型训练：

- 训练参数-边缘强度阈值：边缘对比度阈值，低于该阈值的不被识别为边缘
- 训练参数-边缘长度阈值：低于该长度的边缘被淘汰，用于排除杂点
- 训练参数-金字塔数上限：最大金字塔层数范围 3~7，一般不用设置，金字塔层数低有助于提高匹配的稳定性，但是消耗更多计算时间

设置 ROI：

- 框选感兴趣区域，使模板匹配仅在该区域内执行
- 可拖动 ROI 的四个角点调整四边形框大小、拖动中心点调整位置
- 勾选复选框“启用 ROI”以启用，取消勾选时会在整张图范围内进行匹配

➤ 使用步骤

- 第一次使用时首先构建新的模板并保存，此后使用可一直沿用该模板：
- 在“模型训练”界面，用 ROI 框选目标区域，勾选“训练参数计算”，点击“计算”按钮，自动生成了训练参数，点击“训练”按钮，在“模板预览”窗口中可观察获取的模板；若需要调整，可修改训练参数，再次训练，最后将模板另存为
- 在“参数设置”界面设置输入源，并选择制成的模板，根据需求更改模板匹配运行参数
- 根据需求决定是否启用 ROI，并调整 ROI 范围
- 点击“单次执行”，可获得当前图片模板匹配结果
- 运行结果中展示了所有匹配区域的像素坐标、旋转角度、匹配分数等

4.8.6 线线测量

➤ 功能

用于计算图像中两直线的交点并在图中显示计算直线以及计算结果。



注：

- 本算子通常连接直线查找算子；
- 当所计算交点不能在图中区域显示时，图像中仅显示计算所用直线和错误提示。



界面

➤ 参数

- 图像源：用于直线查找的图像；
- 直线 1 查找源：通常来自于直线查找算子；
- 直线 2 查找源：通常来自于直线查找算子（与直线 1 查找源并列的新算子）。

➤ 使用步骤

- 设置图像源，直线 1 查找源以及直线 2 查找源，点击“单次执行”或“全局执行”；
- 在“结果显示”中显示出了计算所用直线以及交点位置。在“结果输出”中，显示交点的具体坐标。

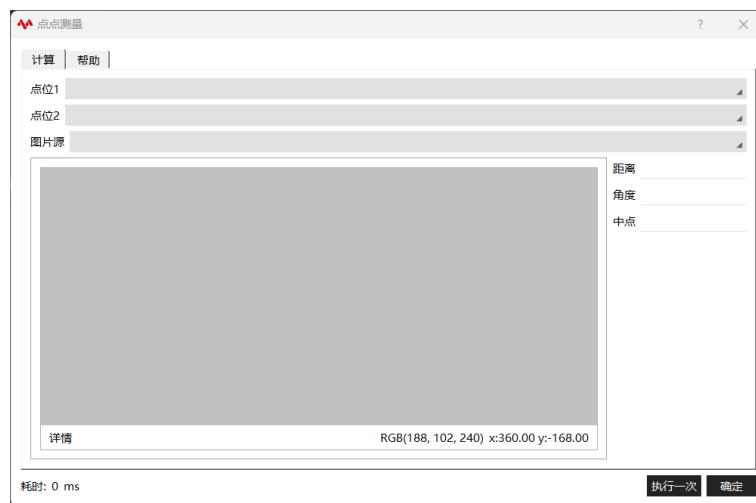
➤ 样例：



4.8.7 点点测量

➤ 功能

测量两点之间的距离，并返回两点中点以及与水平方向夹角。



界面

➤ 参数

- 点位 1：输入的第一个点位（可通过线线转换，格式化等算子获得，输入类型为 point）
- 点位 2：输入的第二个点位（可通过线线转换，格式化等算子获得，输入类型为 point）
- 图片源：绘制点位的图像

4.8.8 位置修正

➤ 功能

位置修正是一个辅助定位、修正目标运动偏移、辅助精准定位的工具。

根据“形状匹配”的结果，计算图像中匹配到的物体相对于模板的平移、旋转及缩放关系，由相对关系计算出匹配后的目标物相对于模板的变换矩阵。



界面

➤ 参数

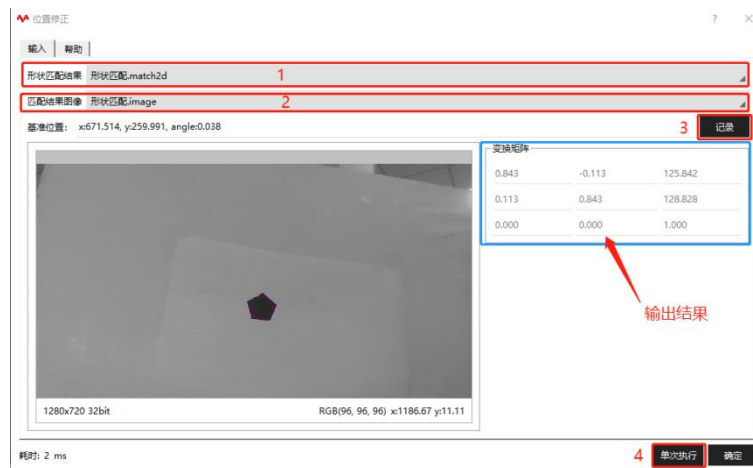
- 形状匹配结果：输入的的形状匹配结果（形状匹配算子提供）；
- 匹配结果图像：仅做显示使用（通常由形状匹配算子提供）；

- 基准位置：模板的基准位置及角度（基准角度较小，通常为 0 度左右小范围波动）。
- 使用步骤
- 选择形状匹配结果；
 - 选择匹配结果图像；
 - 在模板匹配算子建立模板后，及时点击本算子的“记录”按钮，记录模板的基准位置；
 - 点击“全局运行”或“单次执行”，计算位置修正的变换矩阵，即为输出结果。



注：

- 若“基准位置”记录有误，必然导致位置修正中变换矩阵的错误，影响视觉流程的正确执行；



使用步骤

4.9 定位（机器人）

在前述的定位章节中，位姿转换、有向物体定位、获取 3D 位姿算子输出的结果都是目标相对相机坐标系的位姿。这些位姿需要通过本部分的算子进一步处理，变为目标相对机器人基坐标系，或者法兰相对机器人基坐标系的位姿，从而可以发送给机器人，作为机器人程序可用的位姿。

4.9.1 物体位姿

➤ 功能

计算目标物体相对机器人基坐标系的位姿。

参数设置 帮助

输入

物体位姿(相对相机):

相机信息

相机外参(手眼标定): 打开

相机安装方式: 眼在手上

拍照时机器人法兰位姿

位姿状态: 位置固定 记录当前

法兰位姿: (x: 0, y: 0, z: 0) (q1: 1, q2: 0, q3: 0, q4: 0) 手动修改

物体位姿(相对机器人基坐标系)

物体位姿: (x: 0, y: 0, z: 0) (q1: 0, q2: 0, q3: 0, q4: 0)

耗时: 0 ms 单次执行 确定

界面

参数

- 物体相对相机位姿：通常来自位姿转换或有向物体定位算子，通常由定位@相机中的相应算子提供；
- 相机外参：即手眼标定结果文件；
- 相机安装方式：眼在手上或眼在手外两种方式；
- 法兰位姿：拍照时，机器人法兰坐标系相对基坐标系的位姿；可分为固定位姿或非固定位姿两类。
- 输出为物体相对于机器人 base 坐标系的位姿。

4.9.2 抓取位姿

功能

在前述的定位章节中，无向物体定位、有向物体定位、获取 3D 位姿算子输出的结果都是目标相对相机坐标系的位姿。这些位姿可以通过物体位姿算子，转换为机器人 base 坐标系下的位姿，若系统未进行工具（TCP）坐标系的标定时，需要通过抓取标定进一步处理，变为法兰相对机器人基坐标系的位姿，从而可以发送给机器人，作为机器人程序可用的位姿。

参数配置 帮助

输入信息

物体位姿

内部参数

基准物体位姿 (x: 0, y: 0, z: 0) (q1: 1, q2: 0, q3: 0, q4: 0) 修改 记录

基准示教位姿 (x: 0, y: 0, z: 0) (q1: 1, q2: 0, q3: 0, q4: 0) 修改 记录

输出信息

法兰位姿

(x: 0, y: 0, z: 0) (q1: 1, q2: 0, q3: 0, q4: 0)

耗时: 0毫秒 单次执行 确认

界面

参数

- 物体位姿：由“物体位姿”算子输入；
- 基准物体位姿：基准示教时，“物体位姿”算子计算的基准物体位姿；

- 基准示教位姿：示教机器人以特定姿态抓取物体时，获取到的机器人法兰位姿；
- 法兰位姿：输出运行过程中，为抓取物体，机器人法兰应该到达的位姿。

➤ 样例



流程图

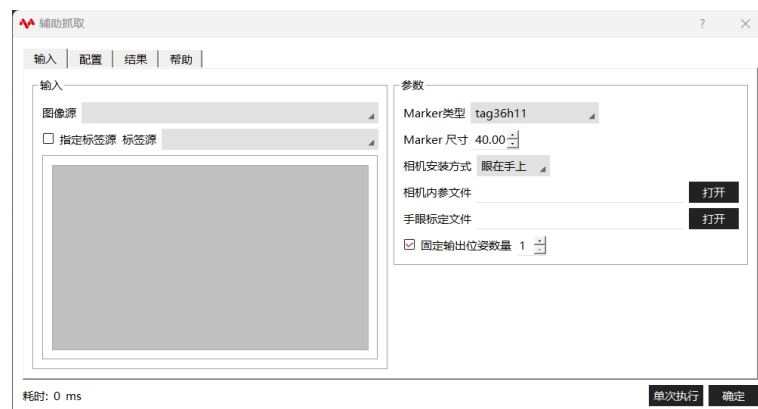


流程图

4.9.3 抓取辅助

➤ 功能

在多个物料台抓取场景中，根据不同的二维码 ID，输出对应的抓取位姿，也就是末端姿态。

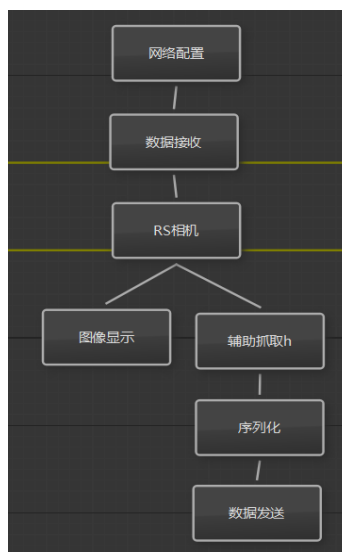


界面

➤ 参数

- 图像源：相机或虚拟相机
- 标签源：用户指定的标签 ID
- Marker 类型：所用二维码的种类
- Marker 尺寸：二维码实际尺寸（mm）
- 相机内参文件：包含相机内参的 json 文件
- 手眼标定文件：手眼标定结果文件
- 固定输出位姿数量：复选框为选中状态时生效，本算子输出的位姿数量为固定；否则将根据示教位姿的数量输出相应的全部抓取位姿
- “添加”按钮：添加一行 Marker 配置，可更改标签 ID、更新当前拍照位
- “编辑示教位”按钮：在打开的对话框中更新示教位，可添加任意数量
- 导入、导出按钮：将现有拍照位、示教位数据导出为.json 文件

➤ 样例



流程图



辅助抓取配置

- 在物料台 1 处拍照，点击“添加”按钮，将识别图中的 Marker 码，在参数配置完整的情况下点击“更新拍照位”，获取当前 Marker 相对机器人的位姿。



- 点击“编辑示教位”按钮：弹出下面的窗口。将机器人移动到抓取位置 1，点击“+示教位”可获取当前的法兰位姿；若当前物料台有多个抓取位，则依次移动和添加。完成后点击“确定”按钮；
- 有多个物料台时，重复执行上述两步，记录所有物料台的拍照位和示教位。记录完成后需要点击“确定”按钮；
- 每次运行将输出要求数量的或所有的法兰位姿，若当前检测到的 Marker 与用户要求的 ID（标签源）不符，则不输出；
- 当配置中存在多个 MarkerID 相同时，仅输出第一个配置中的结果。

```

GLOBAL PROC main()
wait 1
while(true)
    SocketClose("rs")
    wait 0.1
    if(SocketCreate("192.168.0.20",8000,"rs",1))
        wait 0.5
        break
    endif
endwhile
wait 2
//工作循环
WHILE(true)
    //移动到拍照位
    MoveAbsJ PhotoPose,v300,z50,tool0
    wait 1
    SocketClose("AGV")
    wait 0.1
    SocketCreate("192.168.0.20",9000,"AGV",10)
    string expectedID
    expectedID = SocketReadString(3,"AGV") //从AGV获取MarkerID
    print(expectedID)
    //发送expectedID给视觉
    SocketSendString(expectedID,"rs")
    wait 0.5
    //读取位姿并移动
    XV3D=SocketReadDouble(14,3,"rs")
    print(XV3D)

    print(XV3D)
    if(XV3D[1]!=0)
        //移动到过渡位
        MoveAbsJ middlePose,v300,z50,tool0
        grab.trans.x=XV3D[1]
        grab.trans.y=XV3D[2]
        grab.trans.z=XV3D[3]
        grab.rot.q1=XV3D[4]
        grab.rot.q2=XV3D[5]
        grab.rot.q3=XV3D[6]
        grab.rot.q4=XV3D[7]
        MoveJ grab,v300,z50,tool0
        wait 2
        grab.trans.x=XV3D[8]
        grab.trans.y=XV3D[9]
        grab.trans.z=XV3D[10]
        grab.rot.q1=XV3D[11]
        grab.rot.q2=XV3D[12]
        grab.rot.q3=XV3D[13]
        grab.rot.q4=XV3D[14]
        MoveJ grab,v300,z50,tool0
        wait 2
    endif
    //移动到过渡位，结束
    MoveAbsJ middlePose,v300,z50,tool0
    wait 2
ENDWHILE
ENDPROC
  
```

RL 代码

4.10 点云

4.10.1 点云加载

➤ 功能

用于加载不同格式的点云文件（目前仅支持 pcd、ply 两种格式）。加载完成后界面上不显示点云图像，只会显示加载使用时间及加载的点数。



界面

➤ 参数

- 点云路径: 点云的文件路径或文件夹路径。

当参数为点云文件路径时, 只加载该文件点云;

当参数为点云文件夹路径时, 全局运行时将依次加载文件夹下所有符合要求格式的点云文件。

4.10.2 点云查看

➤ 功能

用于显示点云对象 (若点云对象包含 RGB 信息, 显示其自身颜色, 否则, 显示为白色)。



界面

➤ 参数:

- 点云源: 指定点云来源

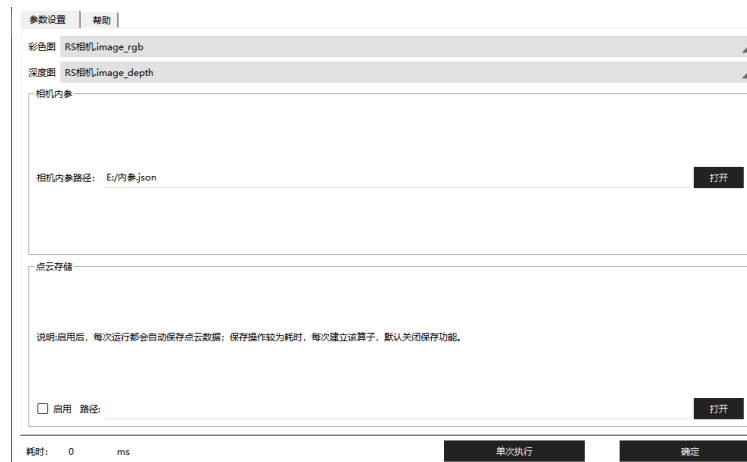
4.10.3 深度图转点云

➤ 功能:

相机采集的深度图和彩色图数据转成点云数据, 算子可以输出点云 pcd 格式的文件, 可以使用点云查看来显示转化的点云。

界面共 3 个 tab：

- 参数选择（彩色和深度图，相机内参）
- 点云存储（保存的地址和是否启用）
- 帮助（使用说明）



界面

➤ 参数：

- 彩色图
- 深度图
- 相机内参

4.11 标定

4.11.1 手眼标定

➤ 功能

手眼标定主要用于确定相机坐标系与机器人之间的关系。当相机安装在机器人末端时，为眼在手上，此时标定出来的是相机坐标系相对机器人法兰坐标系之间的关系；当相机安装在机器人之外固定安装时，为眼在手外，此时标定出来的是相机坐标系相对机器人基坐标系之间的关系。

详细标定过程可以参考第 5 章案例介绍中的相机标定部分。

4.11.2 N 点标定

➤ 功能

用于确定相机坐标系和机械臂世界坐标系之间的转换关系，配合标定转换、单点对位算子，可计算机械臂移动到目标位置的偏移量。



参数设置界面

➤ 参数

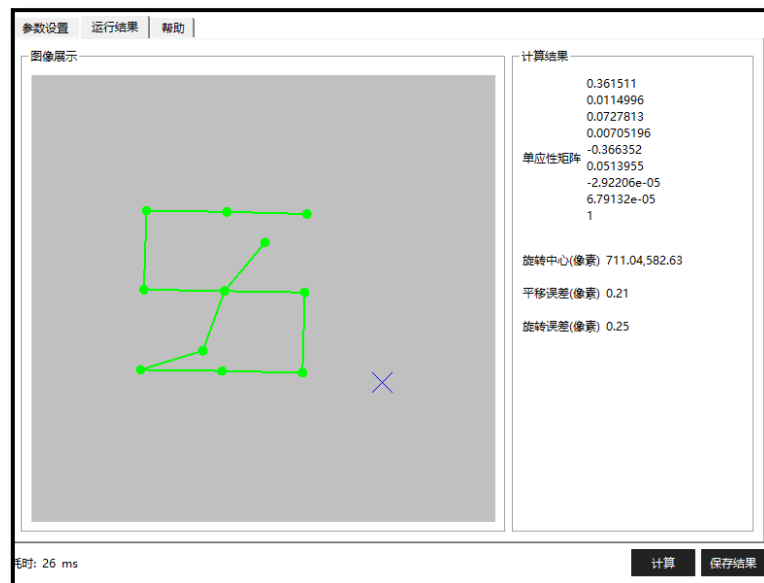
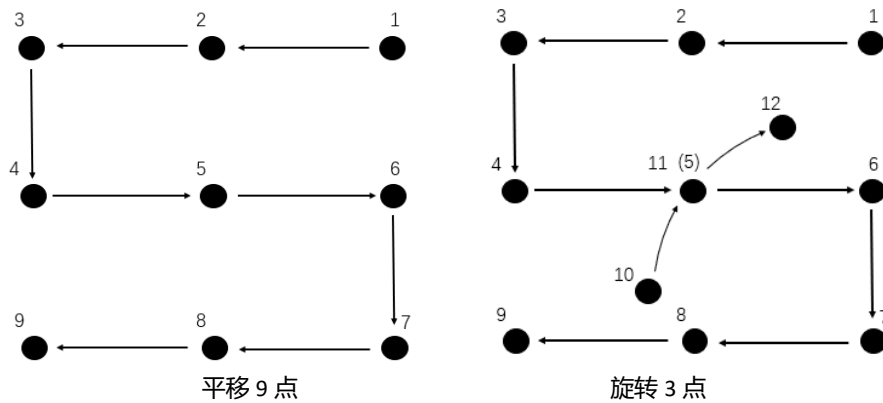
- 输入源：可选择从“模板匹配结果+机器人运行位姿”，算子将获取“形状匹配”算子的结果作为输入；若已有结果，也可选择从文件导入，导入的 json 文件必须含有机器人位姿、形状匹配位置等关键字。
- 平移次数：可选 3x3 或 5x5
- 旋转次数：可选 3 或 5
- 相机安装：相机安装方式：可选择相机静止-上相机
- 旋转角度步长：机器人旋转点位旋转的最小角度

➤ 使用步骤

以眼在手上、平移次数 3x3、旋转次数 3 为例

- 机械臂按照 12 点法进行走位（先平移 9 点，再旋转 3 点，其中第 11 点和第 5 点重合），使目标模板在视野中按照如下图的位置顺序出现。每到一个位置，使用“+”添加新的一行。算子将自动采集此时机械臂位置和角度。“-”按钮可删除选中行，更新按钮可按照更新选中行。采集完成后可保存数据为 json。

- 也可导入现有的数据进行标定，选择“从文件导入”，选择 json 或 txt 文件，点击“更新”按钮进行数据读取。
- 点击“计算”按钮，获取结果并保存为 json 文件。



计算结果界面

4.11.3 标定转换

➤ 功能

将像素坐标系转为相对的世界坐标系。

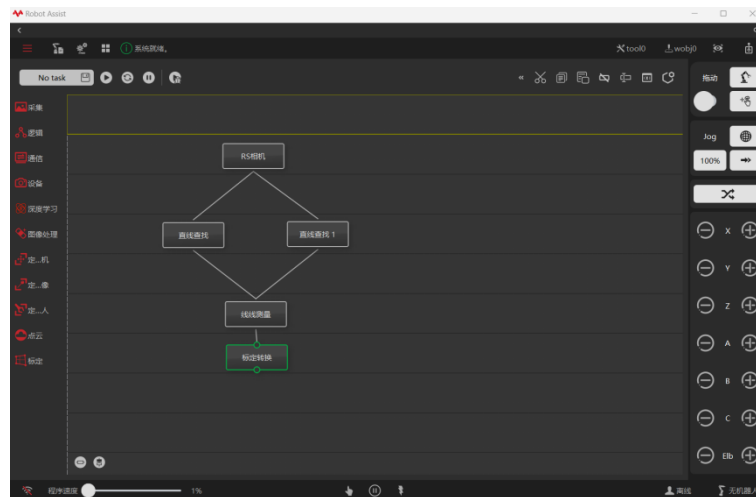


界面

➤ 参数

- 像素点
- 标定文件：N 点标定的结果文件
- 转换结果：当前像素点输入与其经过单应性矩阵转换的结果。

➤ 使用样例



求取两直线交点在基座坐标系下坐标

4.11.4 标定板标定

➤ 功能

利用标定板实现对于双相机的联合标定。



参数设置界面

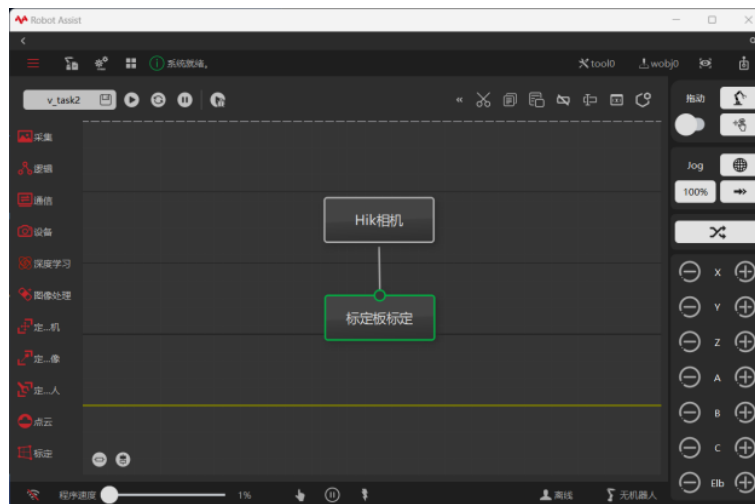


运行结果界面

➤ 参数

- 输入图像：相机采集的标定板图像
- 棋盘格长度：在长度方向上黑白色块总数
- 棋盘格宽度：在宽度方向上黑白色块总数
- 棋盘格边长：棋盘格黑色或白色色块实际尺寸，单位 mm
- 在参数设置界面可通过单击“保存标定板图像”按钮保存生成的标定板图像
- 在运行结果界面可通过单击“保存结果”按钮保存标定结果

➤ 使用样例



4.11.5 单点对位

➤ 功能

计算两个算子输出结果之间的差值（示教点-运行点），这里用于计算机械手偏移量。



界面

➤ 参数

- 示教点：示教点（基准点）的标定转换结果
- 运行点：运行点的标定转换结果

4.11.6 抓取校正

➤ 功能

计算现有图像点在基座标系下位置与基准点的偏移量。

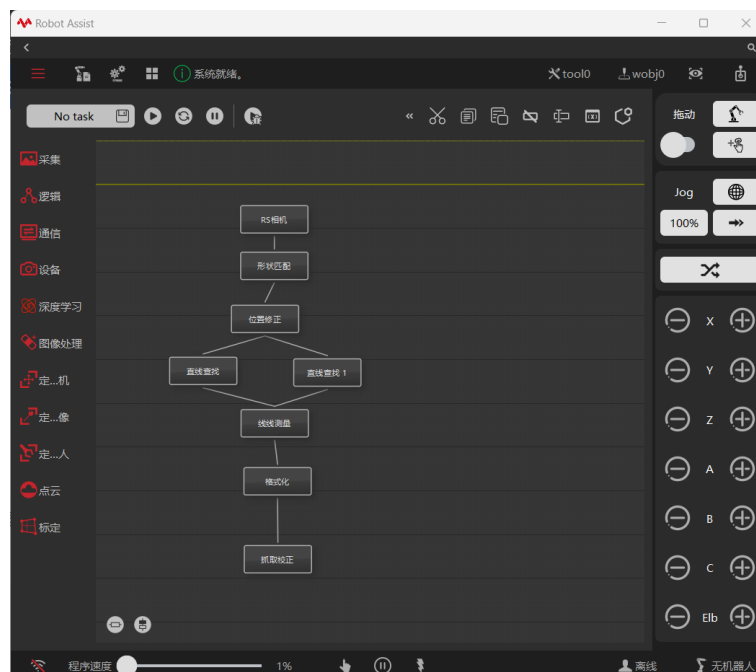


界面

➤ 参数

- 像素点：定位到的像素点
- 相机安装方式：可选眼在手外或眼在手上
- N点标定结果：打开N点标定结果
- 图像基准点：记录基准点位置

➤ 使用样例



返回识别到的位置与目标位置差距

5 案例介绍

机器人视觉引导程序一般由两部分构成：(1)机器人程序，(2)视觉程序。二者通过网络进行通信。以下通过几个典型案例，介绍 xVision 如何配合机器人进行视觉引导作业。

***注意：**以下案例因为软件版本的差异可能会略有不同，但是原理和主要步骤一致。

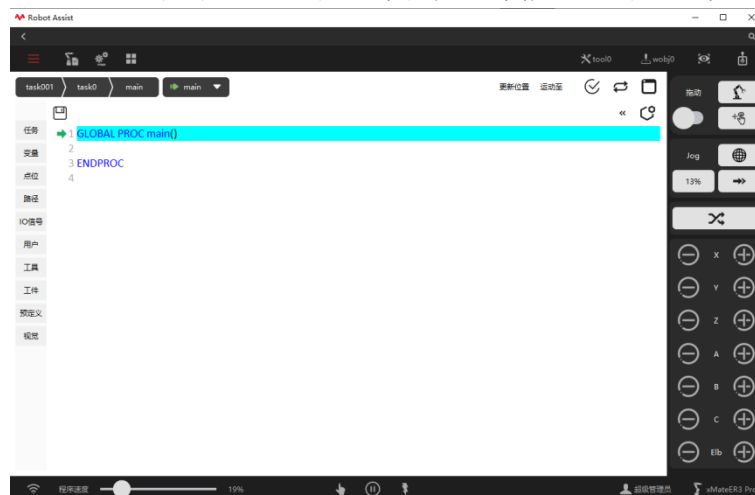
5.1 相机标定

这个案例主要介绍如何进行视觉手眼标定。

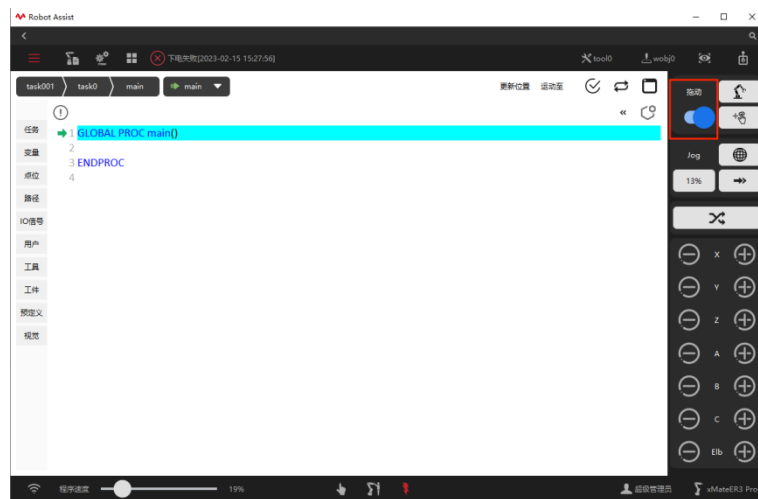
1. 打开随机的软件，点击“工程配置”，输入工程名称，点击“下一步”，完成工程创建。



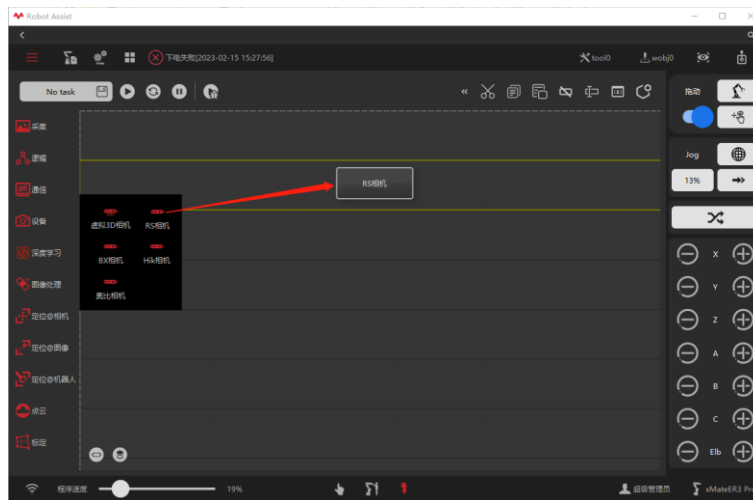
2. 创建完工程后，在 RL 代码区会自动生成两行机器人程序，如下图，点击“机器人操作面板”。



3. 进入‘机器人操作面板’后将“拖动”按钮打开，当正取连接机器后，点击该按钮后，机器人会发出“咔”的一个声响，表明机器上电，并进入了拖动模式，此时用手指同时按住法兰旁边的两片黑色按键，抓着机械臂是可以轻易的拖动机械臂的。



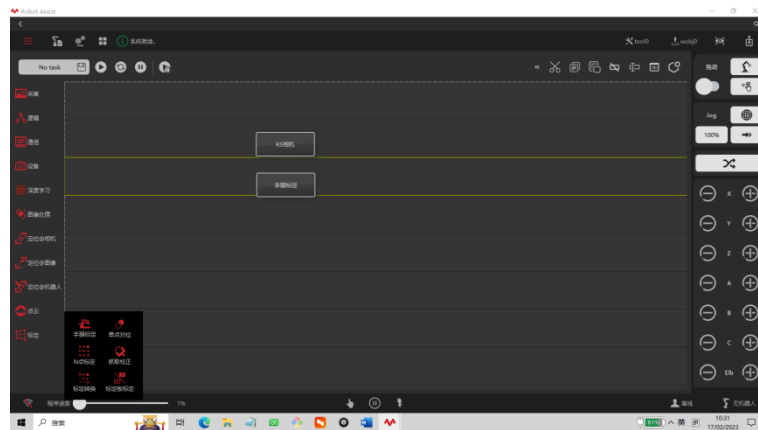
4. 打开“视觉”界面，从设备区拖入相机品牌对应的算子到工作区。



5. 双击工作区中的 RS 相机算子，就会弹出相机界面，选择“运行结果”tab 页中的“彩色图”选项，并点击工作区左上角的视觉程序启动按钮，在相机算子界面就能看到相机视野场景。



6. 暂停视觉程序，从右侧“标定”菜单中，将“手眼标定”拖到视觉工作区并将其与“相机算子”相连。



7. 双击“手眼标定”，选择图像源和标定板参数、相机安装位置。



8. 打开相机界面，或将“图像显示”算子连接于相机，持续运行，在图像显示中观察标定板位置。



注意：标定板在视野中应占据 1/4 到 2/3；

标定板和视野平面的倾斜角度过大时不易识别到；

每组数据和上一组数据相比，应该使机器人各个轴均有约 10°以上的转动；

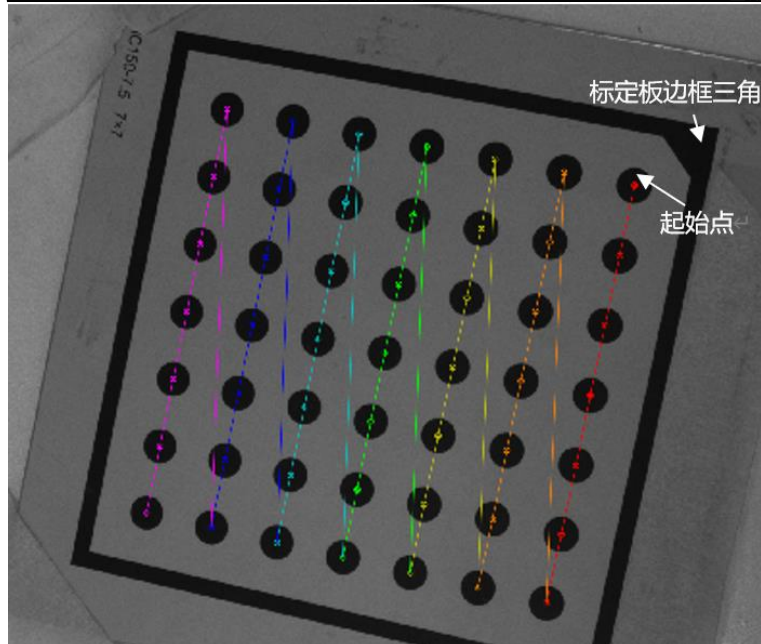
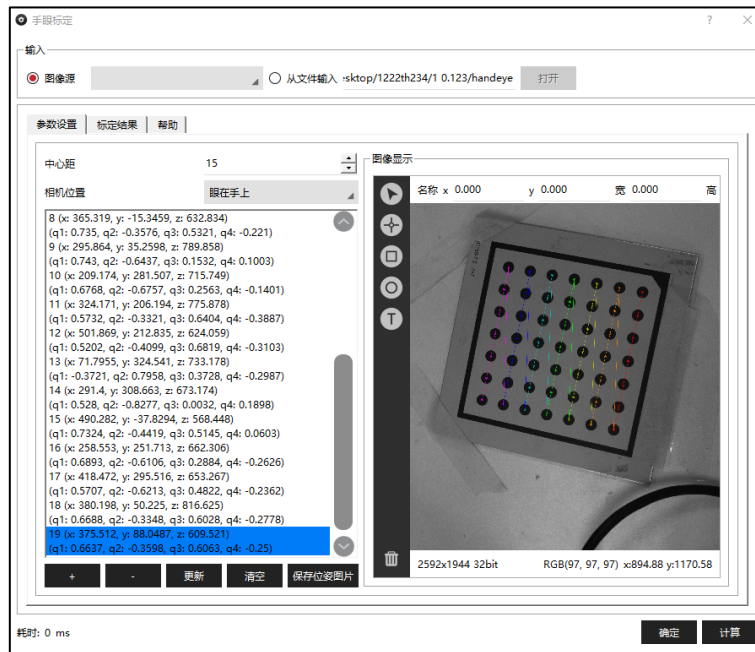
有效数据须在 13 组以上；

9. Jog 或拖动机器人，当标定板在视野中位置合适时，可在点位列表中插入点位，或直接在 RL 代码中添加运动指令。



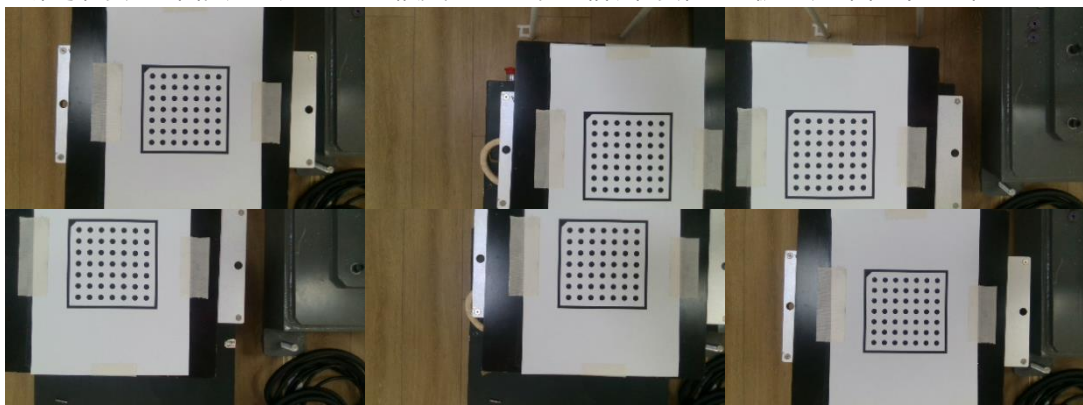
10. 插入完成点位后，关闭拖动模式，用手动使能对机器人上电，单步执行机器人运动指令。
11. 单步执行完一条机器人移动指令后，如下图，视觉运行一次采图，点击手眼标定算子中的“+”按钮添加一张标定图片。图片中可看到彩色或灰色的虚线依次连接各个标定点，且起始点在标定板黑色边框的三角处，此时说

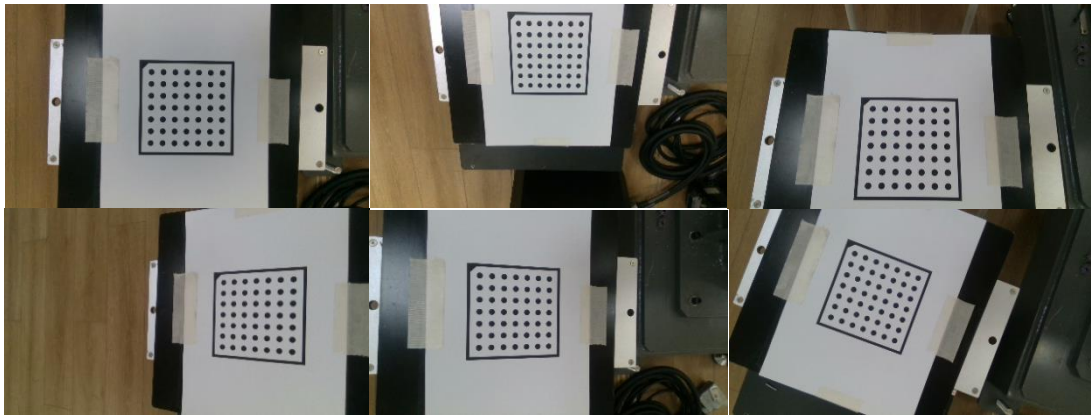
明该图像是有效的。若图像无效，则删除该图片，调整机器人位置重新拍图。



重复第 10 步骤和 11 步骤，采集完成所有点位标定图片，点击“计算”，等待程序计算出结果后，点击“保存”，保存最后的标定结果。也可点击“保存数据”按钮，保存所有图片和机器人位姿信息，保存的信息可在输入源为“从文件输入”时使用。

标定推荐位姿如下图，为了提高标定的精度，尽可能多的相机位姿和标定板能够在图像中各个位置都有出现。





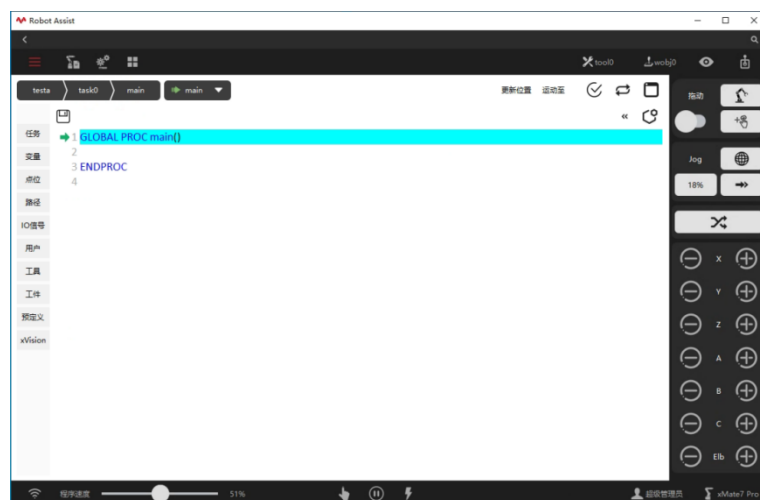
5.2 2D 视觉引导

这个案例主要介绍如何进行 2D 标定与应用，目前仅支持眼在手外（即相机静止）场景，标定时需要使使教点为标定基准点。本案例以上相机抓取场景为例，主要使用到的算子有 N 点标定、抓取校正、形状匹配、位置修正等。

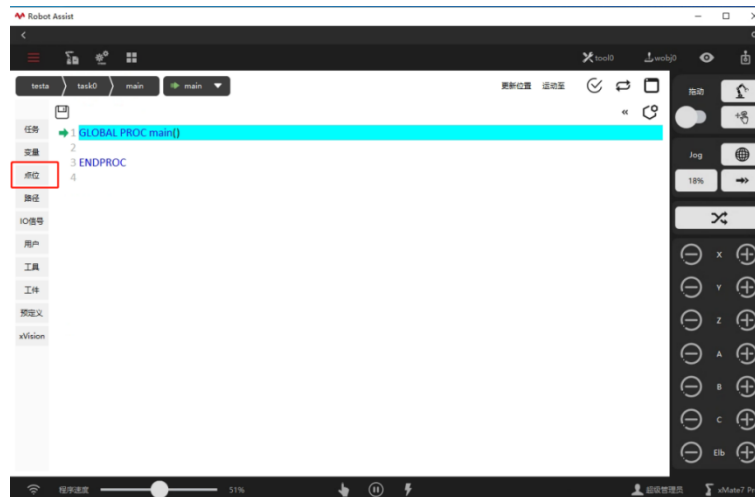
1. 打开软件，点击“工程配置”，输入工程名称,点击“下一步”，完成工程创建。



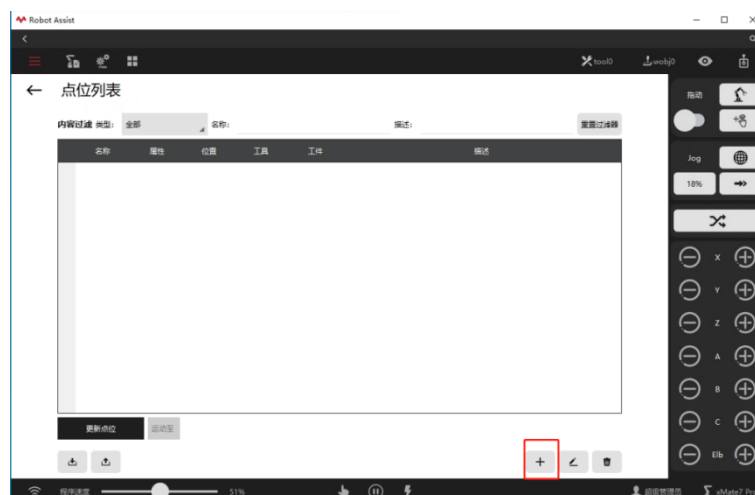
2. 创建完工程后，在 RL 代码区会自动生成两行机器人程序，如下图，点击“机器人操作面板”。



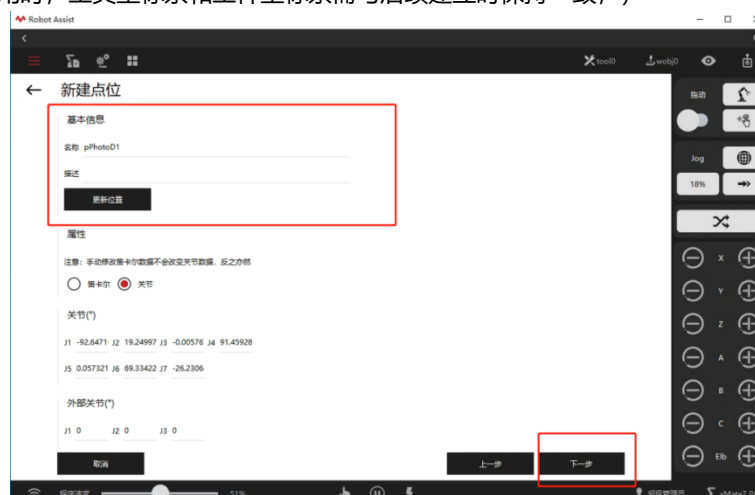
3. 新建机械手点位，设置“额定相机拍照位”，点击界面“点位”。



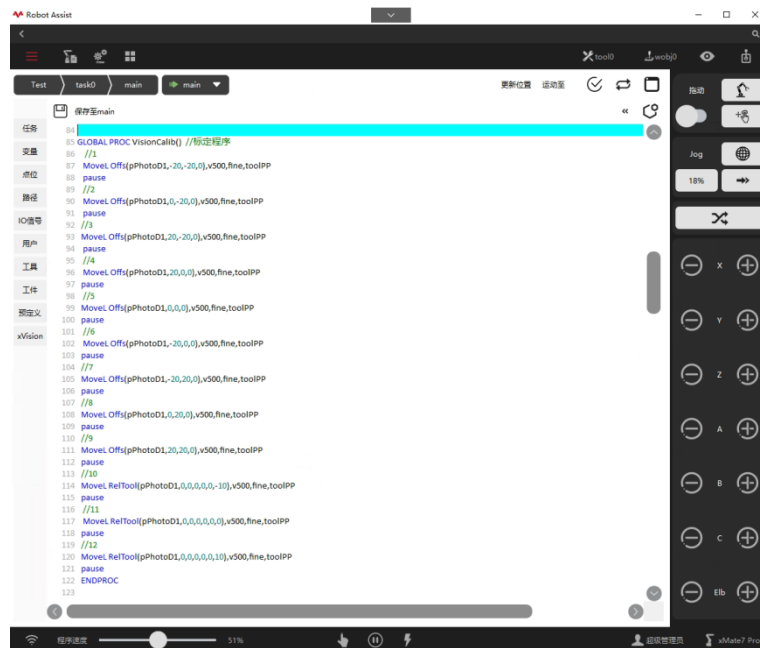
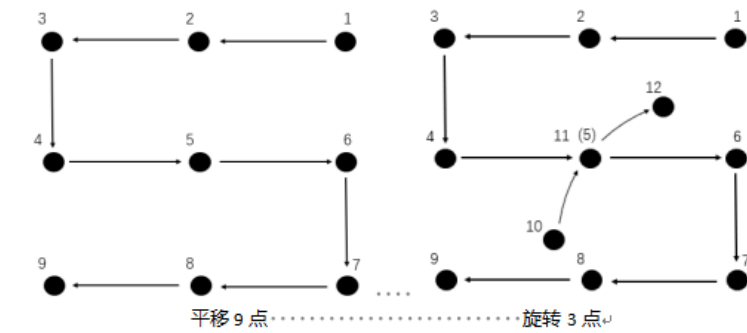
4. 进入“点位列表”界面，点击界面“+”。



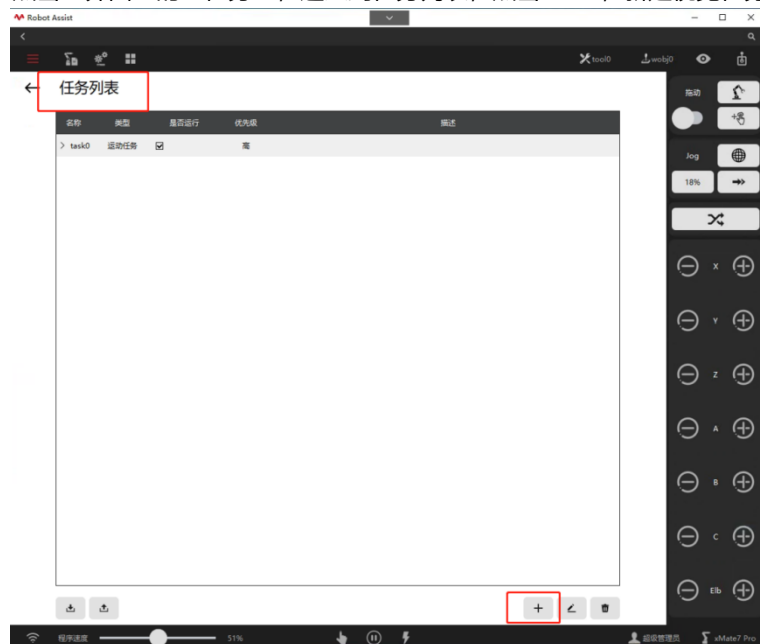
5. 新建点位，可自定义名称为 pPhotoD1, 点击“更新位置”，点击下一步；
(注意若后续实际使用时，工具坐标系和工件坐标系需与后续建立时保持一致；)



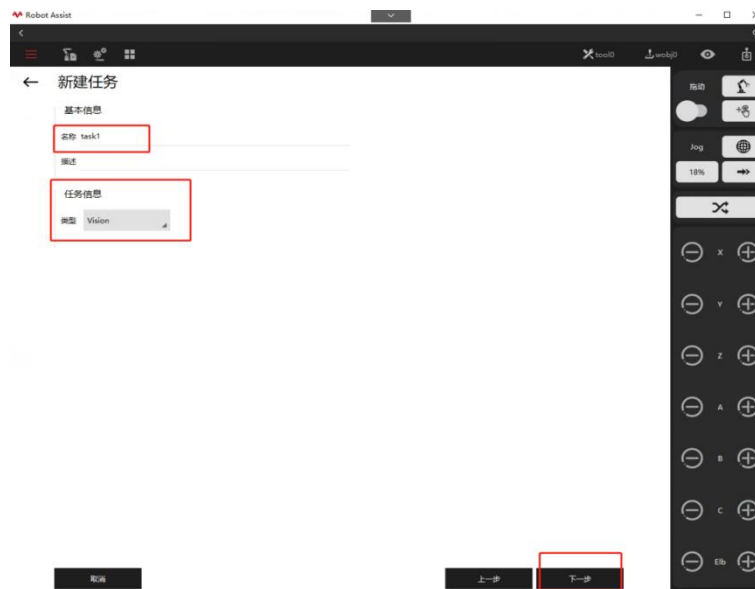
6. 编写机器人标定程序，机械人按照 12 点法进行走位（先平移 9 点，再旋转 3 点，其中第 11 点和第 5 点重合），第 5 点为标定的基准点，也是抓取示教点。在相机中观察目标，使其在视野中按照如下图的位置顺序出现。每到一个位置，使用“+”添加新的一行。算子将自动采集此时机械臂位置和角度。“-”按钮可删除选中行，更新按钮可按照更新选中行。采集完成后可保存数据为 json。



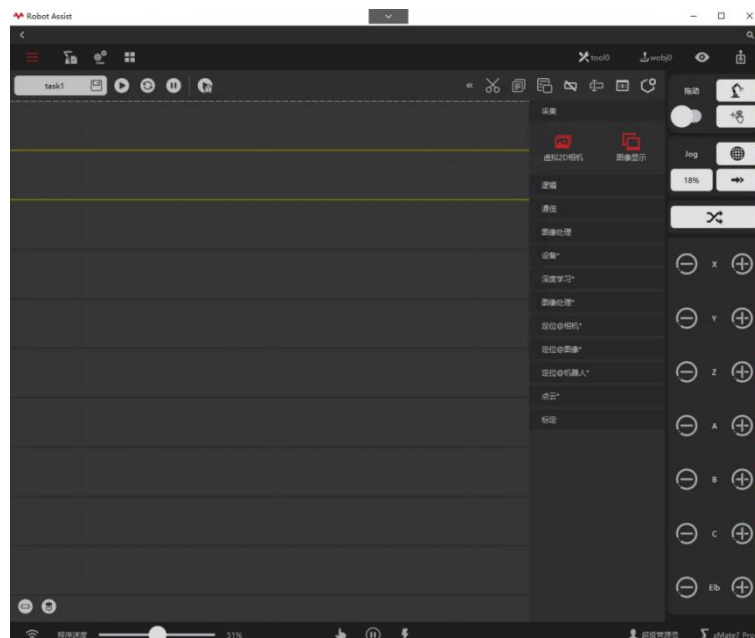
7. 新建视觉任务，点击主界面上的“任务”，进入到任务列表，点击“+”，新建视觉任务；



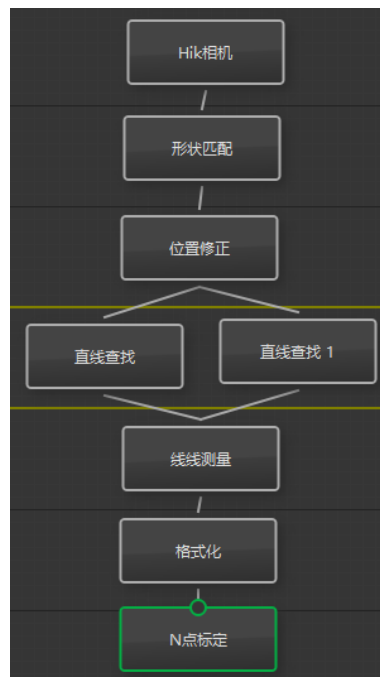
8. 任务信息处类型，选择“Vision”，点击“下一步”即可创建；



9. 主界面中点击“xVision”，进入到视觉任务界面，选择刚新建的 task1；



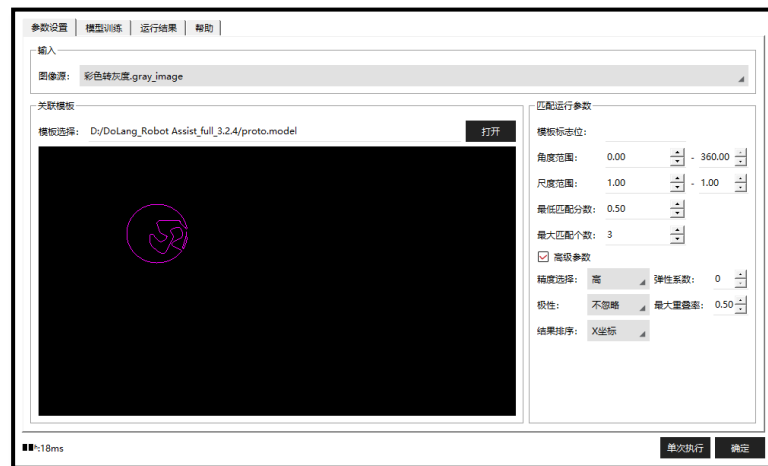
10. 搭建视觉标定流程，从“算子区”拖拽相应算子（Hik 相机、形状匹配、位置修正、直线查找、线线测量、N 点标定）到流程编辑界面，并将其完成连线；



11. Hik 相机算子设置，调试相机，使图像清晰及识别特征明显；



12. 机器人在示教点抓取物料，在标定基准位置拍照，进入形状匹配算子，进行模板训练，并将训练好的模板导入关联模板。设置形状匹配算子参数，包括角度、分数、精度等；若模板为简单的非闭合形状，如直角等，形状匹配误差可能较大，此时需要用其他方法进一步定位特征点，如使用两条直线的交点等方法；若模板复杂度高，则使用模板匹配输出点位即可。



参数设置界面

13. 上一步执行完成后打开位置修正算子，选择形状匹配为输入源，依次点击“记录”、“确定”、“单次执行”按钮：



- 15、打开“直线查找”算子，设定输入源和参数等，使两个直线查找算子分别查找物料特征边缘。

- 16、使用线线测量，输出两个直线交点；

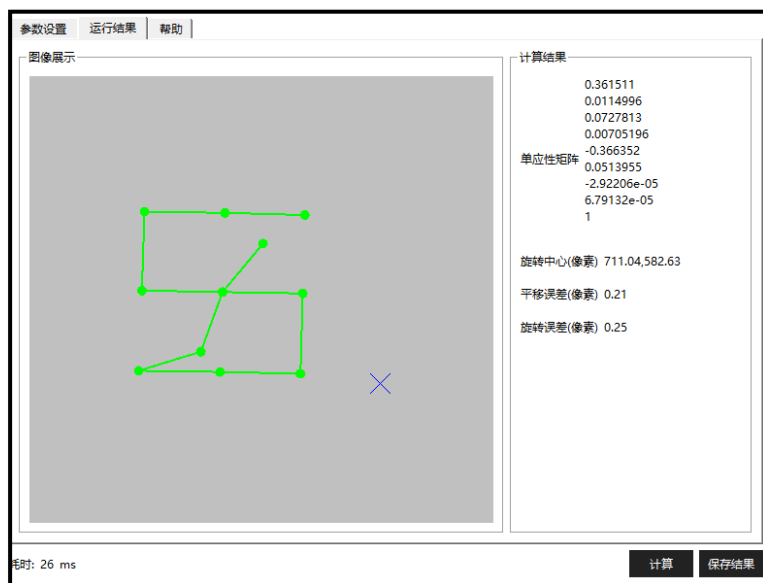
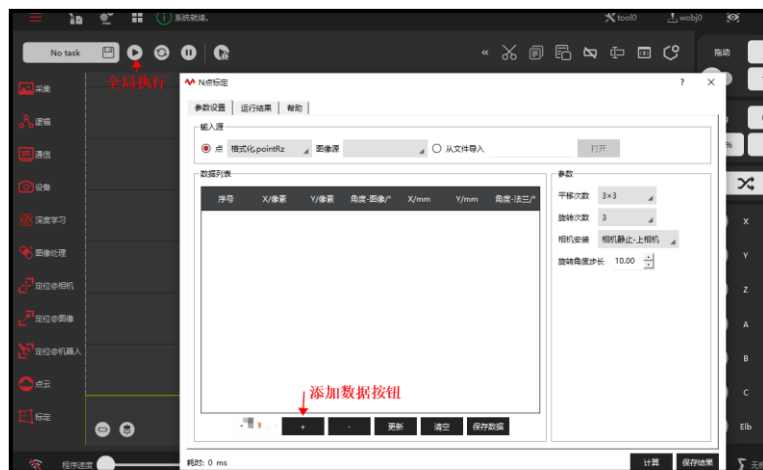
- 17、线线测量的输出为 point 类型，需要使用格式化构造 pointRz 类型的输出，配置如下：



14. N 点标定设置，按照实际点数与旋转角度步长等，设置参数：



15、机械臂每运行到一个新的点位，点击全局执行，使视觉执行依次，检查形状匹配、直线查找结果没有错误后，再点击 N 点标定中“+”按钮，添加一组数据，直到所有点位添加完成。最后点击计算，保存结果。



计算结果界面

15. 搭建“机械手运行程序”，实现场景为单相机物料的抓取场景，基于图像基准点与图像运行点坐标实现抓取，

输出相对偏差值；若视觉异常时，则对应输出 0,999,999,999；

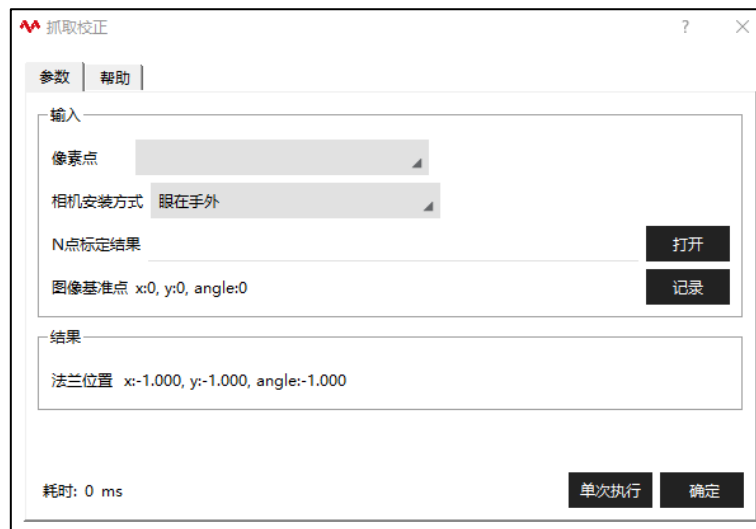
```

26 GLOBAL PROC VisionGetPosXV() //转点位
27   SocketClose("sc") //关闭通讯 (防止残留数据)
28   wait 0.1
29   SocketCreate("192.168.0.34",9797,"sc") //建立通讯
30   MoveJ pPhotoD1,v1000,fine,toolPP //移动到拍照位pPhotoD1
31   wait 0.5 //避免抖动
32   SocketSendString("D1","sc") //发送消息"D1"
33   wait 1
34   visionHK=SocketReadDouble(4,60,"sc") //接收视觉反馈数据
35   int0 = int0 +1
36   print(visionHK,int0) //显示当前接收数据
37   wait 0.1
38 //
39   If[visionHK[1] == 1&&-50<visionHK[2]&&visionHK[2]<50&&-50<visionHK[3]&&visionHK[3]<50&&-40<visionHK[4]&&visionHK[4]<40] //防止数据异常
40     QuaternionToEuler(EULER_XYZ,pPosD1.rot.q1,pPosD1.rot.q2,pPosD1.rot.q3,pPosD1.rot.q4,vision_data[1],vision_data[2],vision_data[3])
41     vision_data[3]=vision_data[3]+visionHK[4] //将偏差加入机械手基准位pPosD1-C值中
42     EulerToQuaternion(EULER_XYZ,vision_data[1],vision_data[2],vision_data[3],pPosD2.rot.q1,pPosD2.rot.q2,pPosD2.rot.q3,pPosD2.rot.q4)
43 //
44     pPosD2.trans.x=pPosD1.trans.x+visionHK[2] //将偏差x加入机械手基准位pPosD1-X值中
45     pPosD2.trans.y=pPosD1.trans.y+visionHK[3] //将偏差y加入机械手基准位pPosD1-Y值中
46 //
47     MoveL Offs(pPosD2,0,0,0),v50,fine,toolPP //移动至更新值后的抓取位pPosD2
48   Else
49     print("YOU+++++++"+IntToStr(int1))
50     pause //若接收数据异常，则停止
51 //
52   ENDIF
53 ENDPROC

```

16. 搭建“视觉引导流程”，使用的算子和流程与 N 点标定主体一致，将 N 点标定替换为抓取校正，并添加了网络收发等算子；打开抓取校正，导入 N 点标定结果





抓取校正算子界面中导入 N 点标定结果

17. xVision-NG 情况处理, 使用“分支”算子对“形状匹配”算子的识别结果 OK/NG 进行区分, 使用“序列化”算子可以实现添加自定义协商文本“0,999,999,999”, 并设置分割符及添加结束符; 通过“数据发送”算子将“序列化”算子设置内容通过 Socket 通讯发送给机械手

5.3 颜色检测

实现场景位: 相机在机械手上, 机械手带动相机完成两个工位的颜色检测;

工位 1: 拍照位 pPhotoD1,

此时检测物料为红色, 机械手发送 red; 视觉 OK 时, 反馈 1,1;NG 时, 反馈 1,0;

工位 2: 拍照为 pPhotoD2,

此时检测物料为黄色, 机械手发送 yellow, 视觉 OK 时, 反馈 2,1;NG 时, 反馈 2,0;

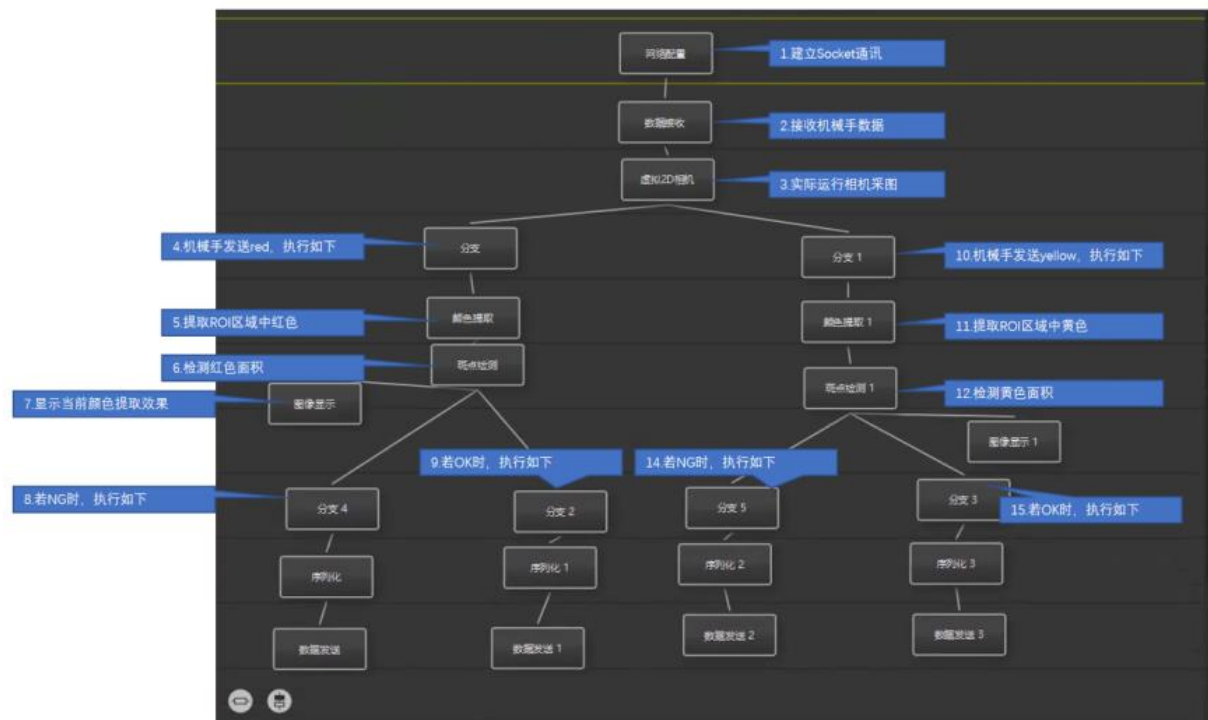
1. 搭建机械手运行程序;

```

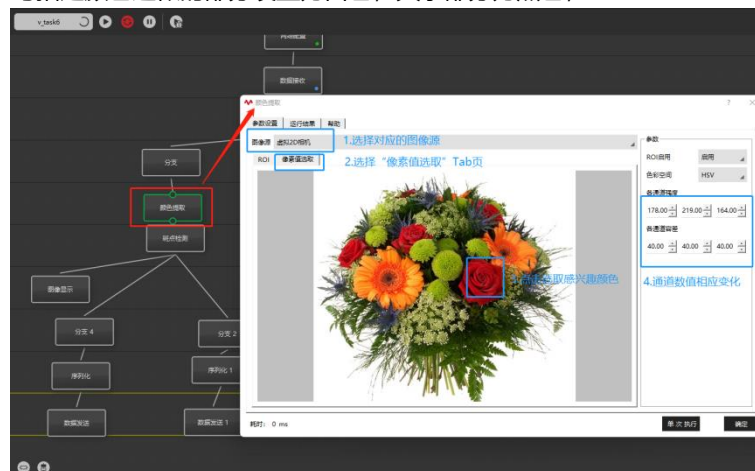
305 GLOBAL PROC VisionColorXV() //颜色检测
306 SocketClose("sc") //关闭通讯 (防止残留数据)
307 wait 0.1
308 SocketCreate("192.168.0.34","9797","sc") //建立通讯
309 MoveJ pPhotoD1,v1000,fine,toolPP //移动到拍照位pPhotoD1, 检测红色物料
310 wait 0.5 //避免抖动
311 SocketSendString("red","sc") //发送消息"red"
312 wait 1
313 visionHK=SocketReadDouble(2,60,"sc") //接收视觉反馈数据
314 int0 = int0 + 1
315 print(visionHK,int0) //显示当前接收数据
316 wait 0.1
317 //
318 IF(visionHK[1] == 1&&visionHK[2]==1) //防止数据异常
319 print("RED,OK")
320 Else
321 print("RED+++++++"+IntToStr(int1))
322 pause //若接收数据异常, 则停止
323 ENDIF
324 wait 2
325 MoveJ pPhotoD2,v1000,fine,toolPP //移动到拍照位pPhotoD2, 检测黄色物料
326 wait 0.5 //避免抖动
327 SocketSendString("yellow","sc") //发送消息"yellow"
328 wait 1
329 visionHK=SocketReadDouble(2,60,"sc") //接收视觉反馈数据
330 int0 = int0 + 1
331 print(visionHK,int0) //显示当前接收数据
332 wait 0.1
333 //
334 IF(visionHK[1] == 2&&visionHK[2]==1) //防止数据异常
335 print("YELLOW,OK")
336 Else
337 print("YELLOW+++++++"+IntToStr(int1))
338 pause //若接收数据异常, 则停止
339 ENDIF
340 ENDPROC

```

2. 搭建视觉颜色检测程序;



3. 设置“颜色提取”算子，从一张彩色图像上提取与指定颜色近似部分，同时生成一张掩膜(Mask)图，掩膜尺寸与原图一致，与指定颜色近似的部分设置为白色，其余部分为黑色；



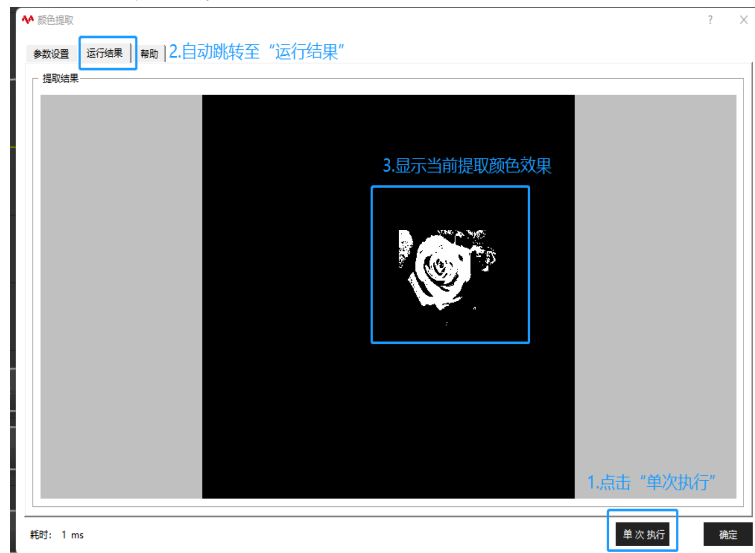
参数设置:

- 输入源: 选择需提取颜色的算子输出图像
- 选择“像素值选取”Tab页;
- 在图像中, 点击感兴趣颜色, 作为提取颜色;
- 色彩空间一般选择 HSV, 提取颜色后, 通道强度数值显示当前提取数值;



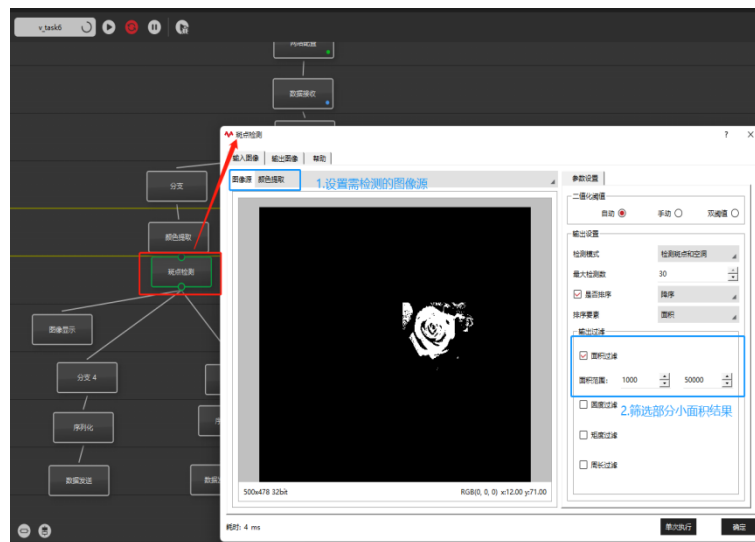
参数设置：

- 选择"ROI" Tab 页
- 选择 ROI 启用，---图像中会出现 ROI 区域
- 设置 ROI 区域包含需检测区域；



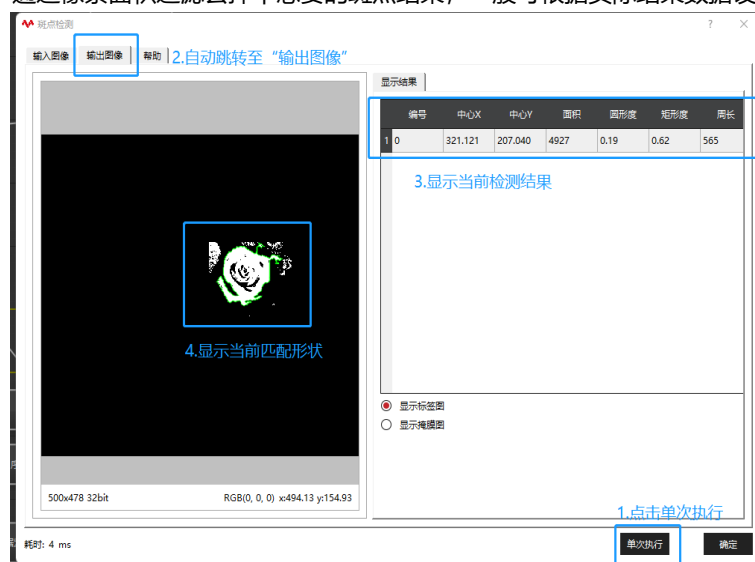
参数设置：

- 点击"单次执行"或"全局执行"
 - 会自动跳转至"运行结果"界面；
 - 图像显示当前提取颜色效果；（可被斑点检测算子使用，来计算该颜色占比面积）
4. 设置"斑点检测"算子，主要用于检测和定位图像中指定灰度范围内、形状通常不可测的区域。



参数设置：

- A. 图像源：待检测图像，要求图像为灰度图；
- B. 面积筛选：通过像素面积过滤去掉不想要的斑点结果；一般可根据实际结果数据设置



参数设置：

- A. 点击“单次执行”
- B. 自动跳转至“输出图像”
- C. 显示当前结果；
- D. 显示当前匹配形状；
- E. 若存在结果时，为 OK，状态为 0；否则为 NG，状态为-1.

5.4 3D 视觉引导流程

实现场景：相机在机械手上，机械手带动相机完成不同高度位置的物体定位，视觉定位出物体所在的 3D 位姿，引导机械手完成抓取；

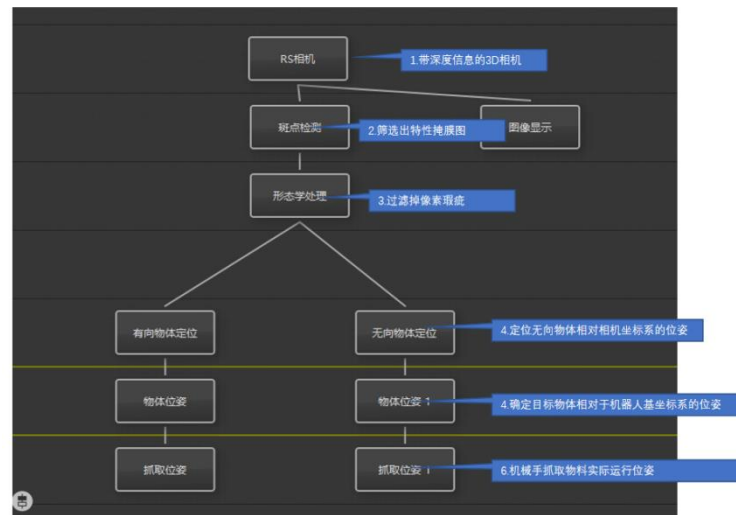
工位 1：有向物体定位+抓取位姿

用于计算有向物体（例如螺钉）的 3D 位姿

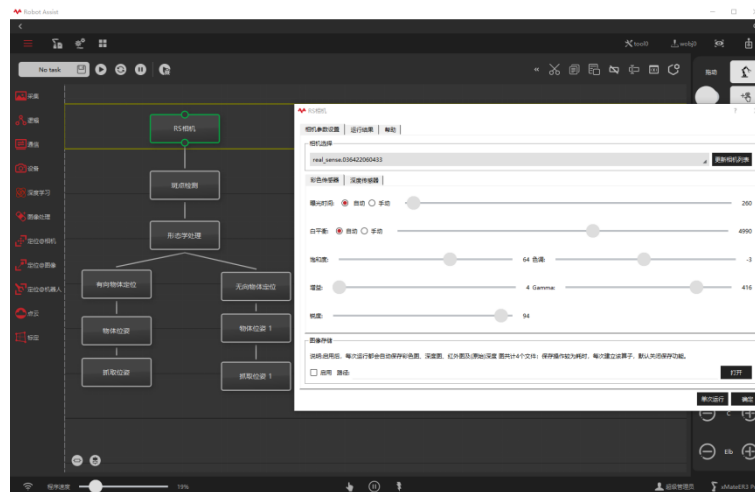
工位 2：无向物体定位+抓取位姿

用于计算目标（各向同性的目标，例如圆形）的 3D 位姿

1. 搭建视觉运行程序：



2. 设置 RS 相机，带深度信息的 RGB 相机

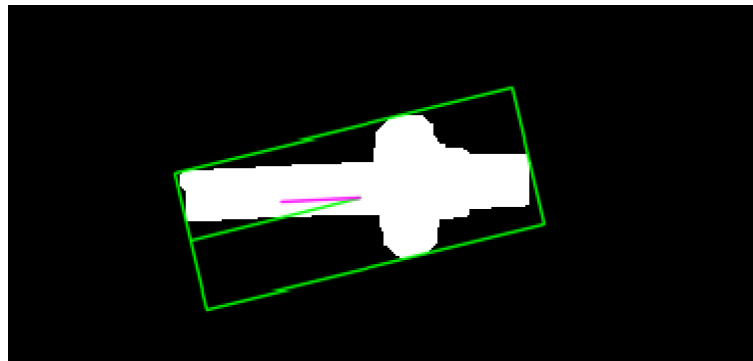


参数设置：

- 相机连接时可自动进行识别，也可以手动选择相机；
- 相机参数设置，运行参数设置，运行结果，帮助
- 相机参数设置：曝光时间（调节图像亮度）；
- 图像存储：设置保存图像路径，
- 运行结果：展示彩色图，深度图，红外图；
- 保存当前图像设置，保存相机内参设置

3. 设置“有向物体定位”算子，“用于计算有向物体（螺钉）相对于坐标系的位姿本算子可计算掩模图像中各斑点的最小包络矩形，通过长度阈值上限和下限可以过滤掉不想要的斑点。此处的长度阈值指物体的实际包络框

长度（单位为像素）。尖端监测范围为相对值（取值在 0~0.5 之间）。



参数设置:

- A. 灰度图像: 输入的掩膜图像, 通常来自斑点检测;
- B. 深度图: 通常来源于相机深度图
- C. 相机内参: 标定得到的相机内参

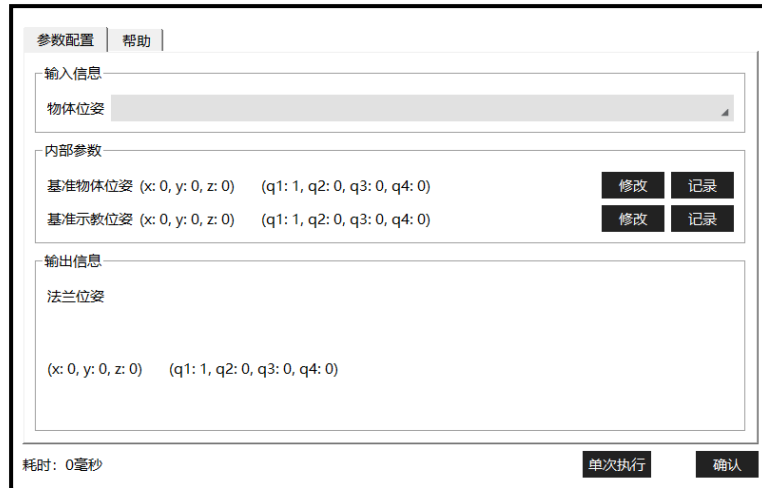
4. 设置“物体位姿”算子, 确定目标物体相对机械手基座坐标的位姿;



参数设置:

- A. 物体位姿: 通常来自位姿转换或有向物体的位姿;
- B. 相机外参: 即手眼标定文件;

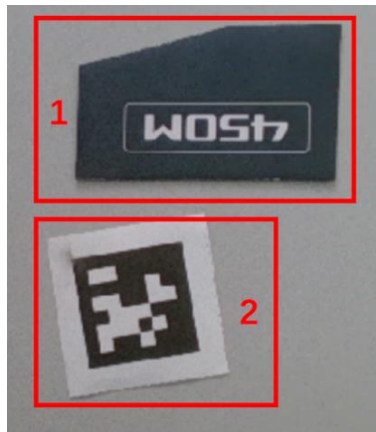
- C. 相机安装方式：眼在手上或眼在手外；
- D. 记录拍照时的相机位姿，到达拍照位后点击记录即可。
5. 设置“抓取位姿”算子，位姿可以通过“物体位姿”算子，转换为机器人 base 坐标系下的位姿，若系统未进行工具坐标系的标定，或工具坐标系与工件坐标系标定不相符时，需要通过抓取标定进一步处理，变为法兰相对机器人基坐标系的位姿，从而可以发送给机器人，作为机器人程序可用的位姿。



5.5 斑点检测、模板匹配、Marker 定位

功能简介

- 基于“模板匹配”、“颜色抽取+斑点检测”两种方式进行下图 1 所示目标物体的定位引导；
- 通过 Marker 检测进行下图 2 所示 Marker 的定位引导；
- 利用网络算子进行数据收发；
- 利用分支算子确定哪个分支被执行；



任务方案整体介绍

拖拽需要的视觉算子(下图部分算子已经人为修改名字)至流程编辑区，连接各个算子，搭建下图所示的视觉任务。



最上方为网络数据接收算子，根据接收到的数据，由三个分支算子决定 1、2、3 哪个分支被选中执行。

网络配置

参数 结果

网络配置

协议 TCP 服务端

IP 0.0.0.0

端口 9797

连接策略

策略 短连接

应用 OK Cancel

网络配置

数据接收

输入

网络 网络配置

参数

格式 CSV

超时(ms) 0

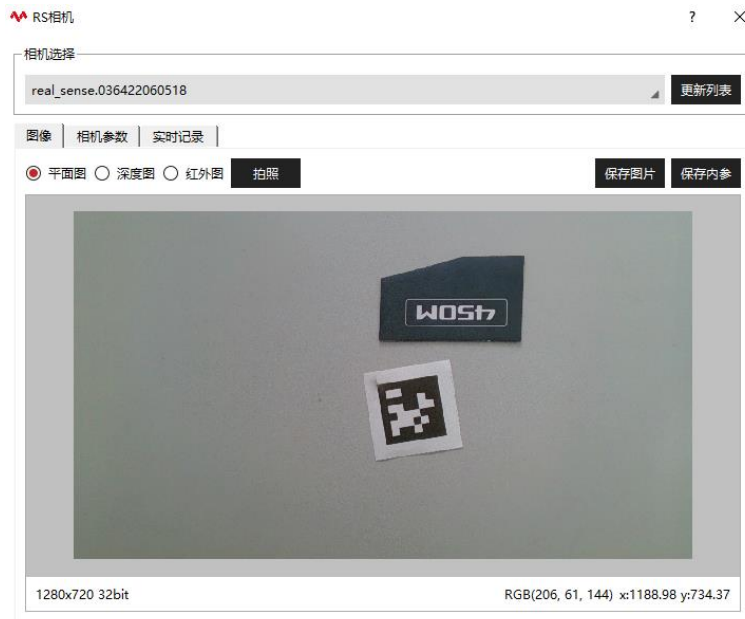
接收数据

string str 删除

增加

OK Cancel

数据接收

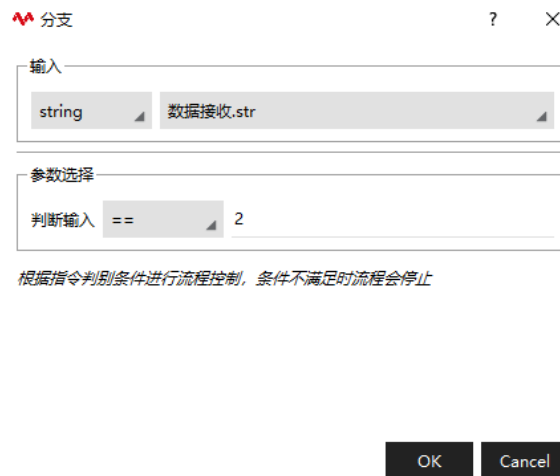


RS 相机

分支 1

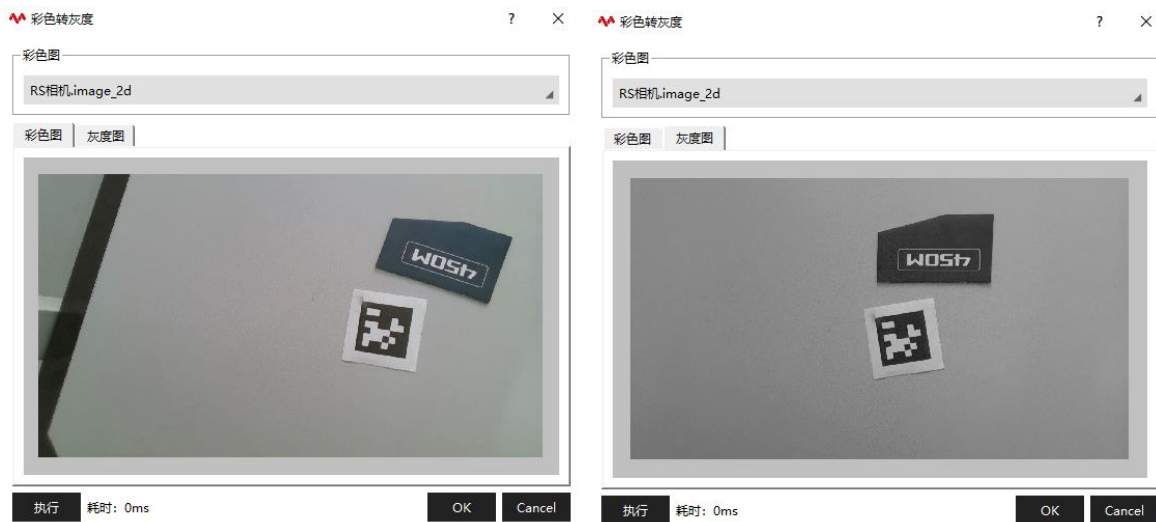
分支 1 的思路为通过“彩色转灰度”，将彩色图像转化为“模板匹配”所需的灰度图像；利用“模板匹配”确定目标的平面位姿，进而通过“获取 3D 位姿”算子计算目标相对相机坐标系的立体位姿；再通过“物体位姿”算子计算目标相对机器人基坐标系的位姿；最后由“数据发送”算子把位姿数据发送出去。

当网络接收的数据为 2 时，进入该分支。



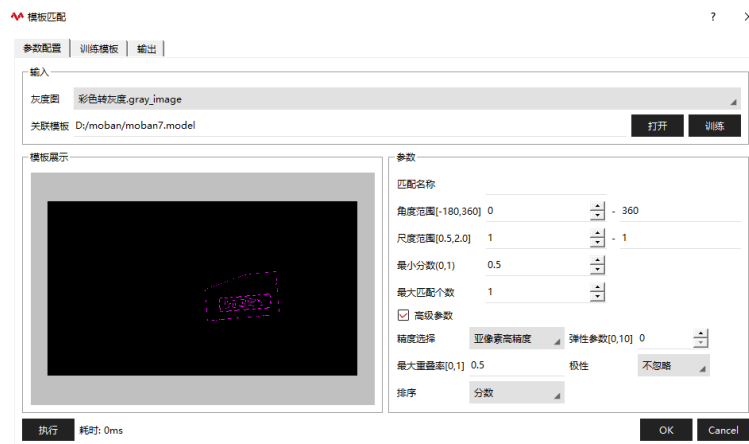
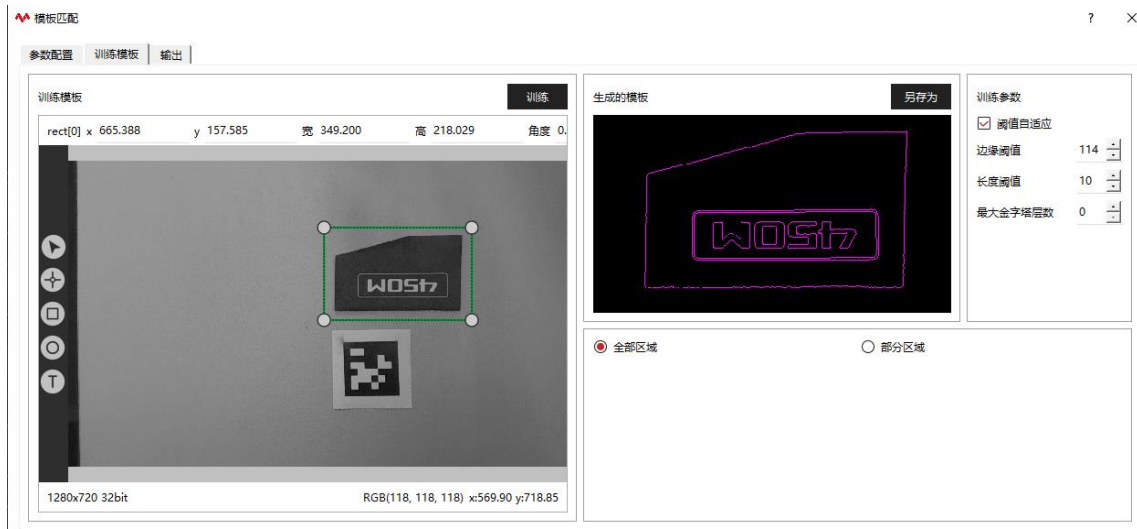
分支

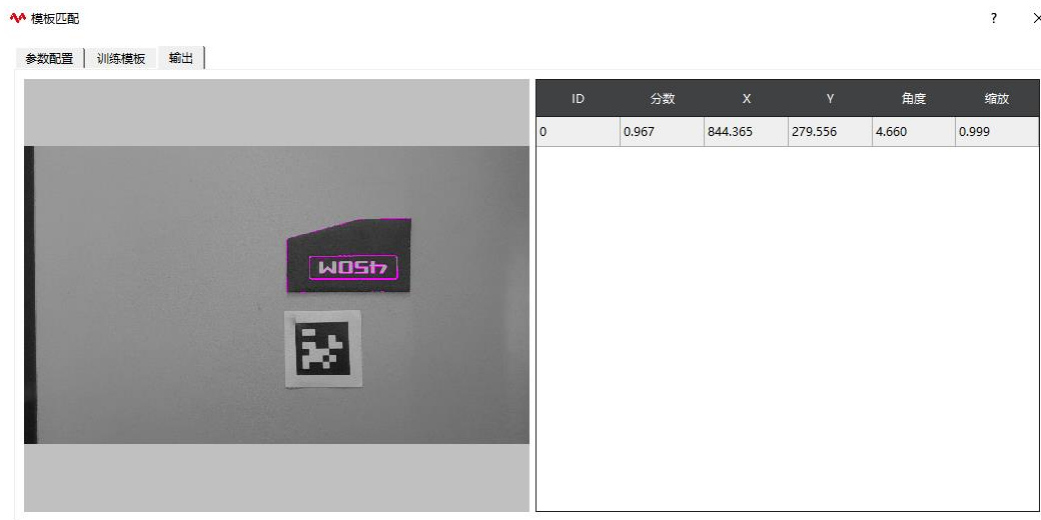
彩色转灰度算子完成颜色转换。



彩色转灰度

设置 ROI，训练模板，并调整运行参数。





设置并查看获取 3D 位姿结果。



假设机器人法兰处于(0,0,0,1,0,0,0)，计算此时物体相对机器人基座的位姿。



位姿数据可以通过网络发送出去。



分支 2

分支 2 的思路是通过“颜色抽取”算子分割得到目标；这时候的结果往往有很多噪点，可以通过“形态学处理”算子优化；再通过“斑点检测”进一步过滤得到期望的目标位置，由“有向物体定位”算子处理得到目标相对相机的位姿；最后通过“物体位姿”得到物体相对机器人基坐标系的位姿，并由“数据发送”发送出去。

当网络接收的数据为 1 时，进入该分支。



分支

颜色抽取获得初步的目标。



颜色抽取

利用形态学处理的闭运算，获取更完整的目标。



形态学处理

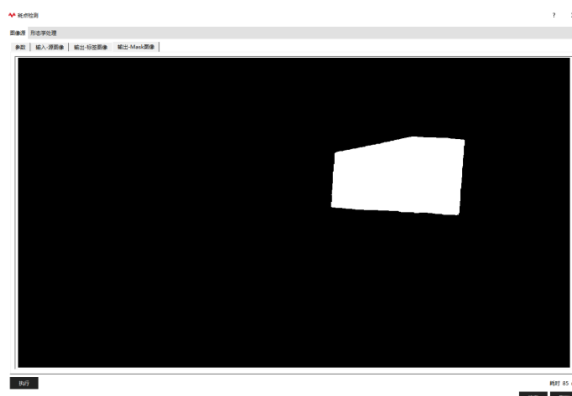
进一步通过斑点检测的面积过滤，去除噪点，获取目标 mask(掩模)图像。



斑点检测

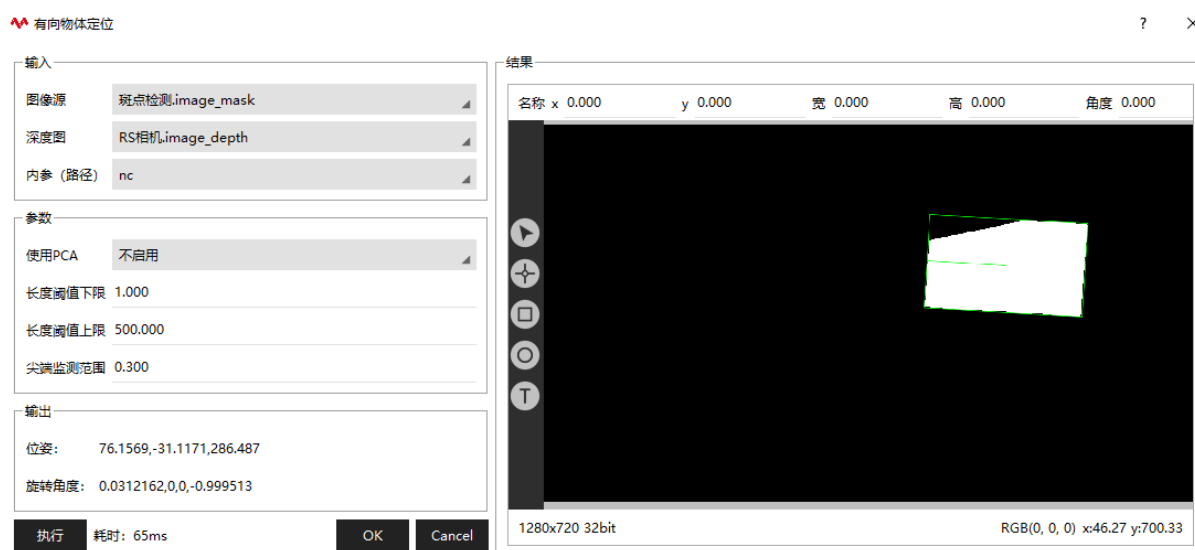


斑点检测标号图



斑点检测掩模图

利用有向物体定位确定目标相对相机位姿。



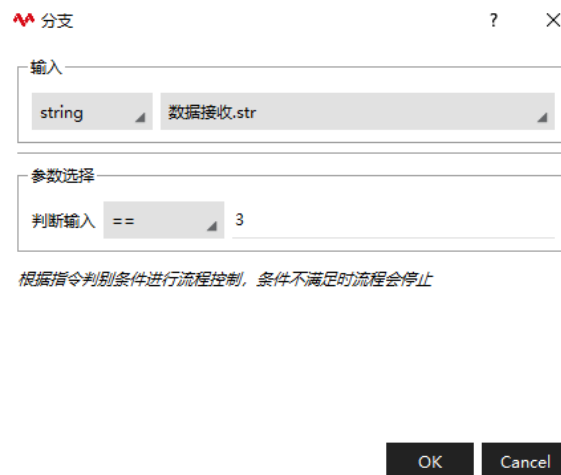
其他跟分支 1 类似，不再赘述。



分支 3

分支 3 的主要思路是通过“Marker 检测”算子进行 Marker 定位，获取 Marker 相对相机的位姿，进而由“物体位姿”得到 Marker 相对机器人基坐标系的位姿，并发送出去。

当网络接收的数据为 1 时，进入该分支。



分支

利用 Marker 检测确定 Marker 相对相机的位姿。



Marker 检测

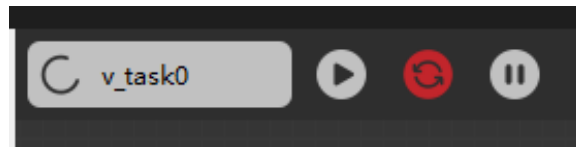
其他跟分支 1 类似，不再赘述。



运行

可以利用网络调试助手发送数据给视觉任务，并接受视觉任务的反馈值。

点击“连续运行”，启动视觉任务。



利用网络调试助手分别发送 1、2、3，视觉任务可以反馈对应分支的执行结果。

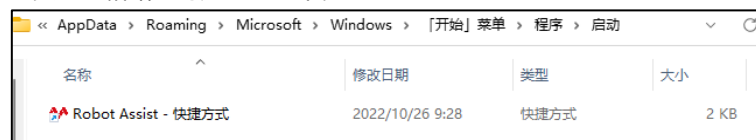


5.6 自启动功能

1、设置 RobotAssist 软件开机自启动。

“WIN+R” 打开 “cmd” 命令行，输入 “shell:startup” ；

将 RobotAssist 快捷方式复制后粘贴到打开的文档中；



2、设置 xVision 开机自启动。

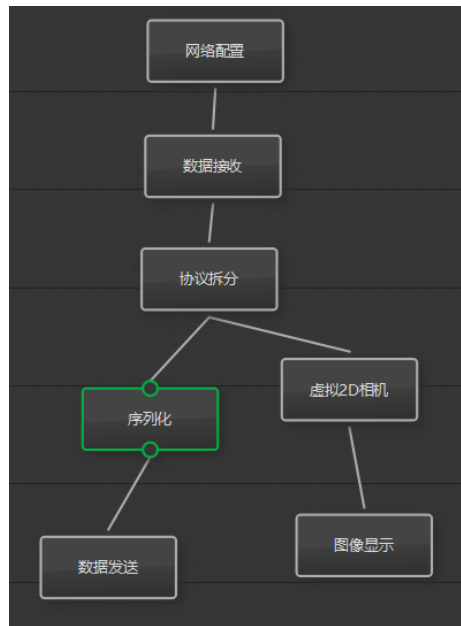
进入工程后新建视觉任务，在任务列表中的视觉任务“是否运行”按钮勾选；



3、xVision 自启动测试。

新建的视觉任务添加相关的测试算子并运行程序；

将图中所示算子拖入并连接；



点击“网络配置”算子，并进行如图设置；



点击“数据接收”算子，并进行如下配置；



点击“协议拆分”算子，并进行如下配置；



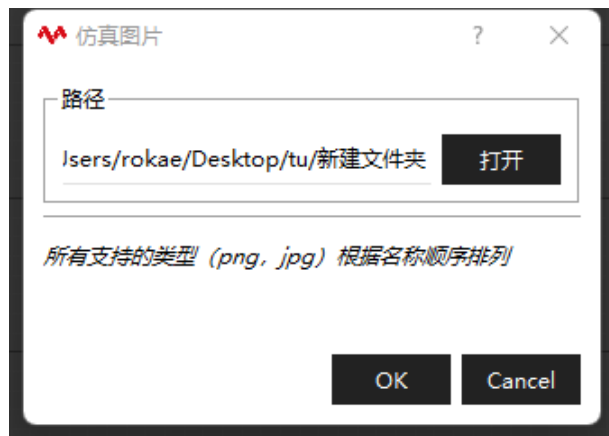
点击“序列化”算子，并进行如下配置；



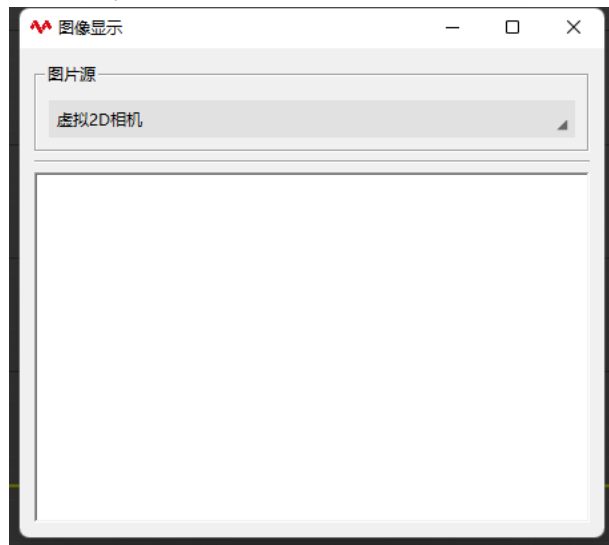
点击“数据发送”算子，并进行如下配置；



点击“虚拟 2D 相机”算子，（路径选择一个存放图像的文件夹）并进行如下配置；



点击“图像显示”算子，并进行如下配置；



在 HMI 新建的工程中，输入相关的测试代码测试，使机器人运行至程序设定点位；

```

1 GLOBAL PROC main()
2
3 while(true)
4   SocketClose("sc")
5   wait 0.5
6   if (SocketCreate("192.168.0.34",9797,"sc",1))
7     wait 0.5
8     SocketSendString("1,33","sc")
9     wait 0.5
10    print("suc")
11    xv = SocketReadDouble(1,1,"sc")
12    print(xv)
13    if (xv==33)
14      break
15    ENDIF
16    wait 0.5
17  else
18    print("fail")
19  endif
20  wait 0.5
21 ENDWHILE
22
23 MoveJ PointHome,v1000,z50,tool_throat_6
24 wait 10
25
26 pausea
27

```

将机器人切换至“手动”模式，移动机器人使其偏离程序设定的目标点位；

重启电脑，在机器人上电后，机器人自动运行到程序设定点位则说明 xVision 程序自启动设置完成。



注意：因为通信延迟，在 SocketCreat、send、read 之间需要加入一定等待时间；

断开时，建议先断开与机器人连接，再关闭 HMI 界面；

6 常见问题及处理

ID	问题描述	处理
1	网络通信算子连接问题	网络连接算子需要注意，每次打开后，机器人程序只连接一次后，就不要再连接该算子，建议先单步连接一次 TCPConetction 算子后，再将连接算子的机器人代码注释掉，或者将除连接 TCPConetction 代码都用一个 while/endwhile 语句包含起来，将连接 TCPConetction 算子的代码部分放到 while 循环前。
2	算子标志位问题	视觉算子的算法执行成功或者失败会返回一个标志位（如第 8 位），如果算子执行成功，检测到物体位姿时，标志位为 0，如果算子执行失败或者未检测到物体的位姿时，标志位为 1.在机器人代码中可视实际情况处理。
3	颜色抽取稳定性问题	颜色抽取算子是基于检测物体颜色来分割出物体的，由于光照因素会导致颜色抽取失效或者错误，遇见这种情况，首先建议要保证机器人工作环境的光照稳定性，其次是可以调节“颜色抽取”算子 HSV 空间的范围。
4	机器人精度问题	不论是在标定还是在抓取，尽量要保证机器人的“臂角”接近于零，这样能够保证机器人可以工作在高的精度内。