

---

文件名称: xVision 使用手册  
单 位: 珞石（北京）科技有限公司  
日 期: 2022-09

# 目录

<b>目录</b> .....	<b>2</b>
<b>1 前言</b> .....	<b>6</b>
1.1 适用版本 .....	6
1.2 修订记录 .....	6
<b>2 产品简介</b> .....	<b>7</b>
2.1 功能概述 .....	7
2.2 功能特性 .....	7
2.3 运行环境 .....	7
2.4 硬件连接 .....	7
2.5 软件部署 .....	7
<b>3 软件介绍</b> .....	<b>8</b>
3.1 界面布局 .....	8
3.2 快捷工具栏 .....	9
3.3 流程编辑区 .....	9
3.4 算子区 .....	10
<b>4 视觉算子介绍</b> .....	<b>11</b>
4.1 采集 .....	11
4.1.1 虚拟2D相机 .....	11
4.1.2 图像显示 .....	11

4.2 设备 .....	11
4.2.1 虚拟3D 相机.....	12
4.2.2 RS 相机.....	12
4.2.3 海康相机 (2d) .....	15
4.2.4 BX 相机.....	16
4.3 逻辑 .....	17
4.3.1 协议拆分.....	17
4.3.2 分支.....	18
4.3.3 格式化.....	19
4.3.4 序列化.....	19
4.4 通信 .....	21
4.4.1 网络配置.....	21
4.4.2 数据接收.....	22
4.4.3 数据发送.....	22
4.5 图像处理 .....	23
4.5.1 颜色提取.....	23
4.5.2 颜色检测.....	25
4.5.3 颜色分割.....	26
4.5.4 形态学处理.....	28
4.5.5 彩色转灰度.....	29
4.5.6 深度补偿.....	30
4.6 定位 (图像) .....	32

4.6.1 卡尺工具.....	32
4.6.2 直线查找.....	34
4.6.3 圆查找.....	35
4.6.4 斑点检测.....	36
4.7 定位 (相机) .....	38
4.7.1 无向物体定位.....	38
4.7.2 Marker 检测.....	39
4.7.3 有向物体定位.....	40
4.7.4 形状匹配.....	41
4.7.5 获取3D 位姿.....	42
4.8 定位 (机器人) .....	43
4.8.1 物体位姿.....	43
4.8.2 抓取位姿.....	44
4.9 标定.....	45
4.9.1 手眼标定.....	45
4.9.2 N 点标定.....	45
4.9.3 标定转换.....	47
4.9.4 单点对位.....	47
4.10 点云.....	48
4.10.1 点云加载.....	48
4.10.2 点云查看.....	48
4.10.3 深度图转点云.....	49

4.11 深度学习 .....	50
4.11.1 深度学习检测.....	50
<b>5 案例介绍.....</b>	<b>51</b>
5.1 相机标定 .....	51
5.2 2D 视觉引导.....	56
5.3 颜色检测 .....	68
5.4 3D 视觉引导流程.....	73
5.5 斑点检测、模板匹配、MARKER 定位.....	76
<b>6 常见问题及处理.....</b>	<b>85</b>

# 1 前言

## 1.1 适用版本

xVision V1.0.0;

xCore V1.6.0;

## 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	● 更新相机、斑点检测等算子	2021.03	ysn
V0.1.2	● 添加手眼标定和使用案例	2021.04	pch
V0.1.3	● 添加手眼标定 pose	2021.04	pch
V0.1.4	● 添加深度补偿	2021.04	ysn
V0.1.5	● 同步至最新软件版本, 内容大幅更新	2021.05	ysn
V0.1.6	● 更新案例介绍, 增加颜色检测算子	2021.05	ysn
V0.1.7	● 添加颜色分割工具	2021.06	lyn
V0.1.8	● 添加深度学习目标检测工具	2021.07	lyn
V0.1.9	● 升级深度补偿算子	2021.08	ysn
V0.1.10	● 所有算子重构	2022.	
V0.1.11	● 添加格式化、序列化算子	2022.08	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

## 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。

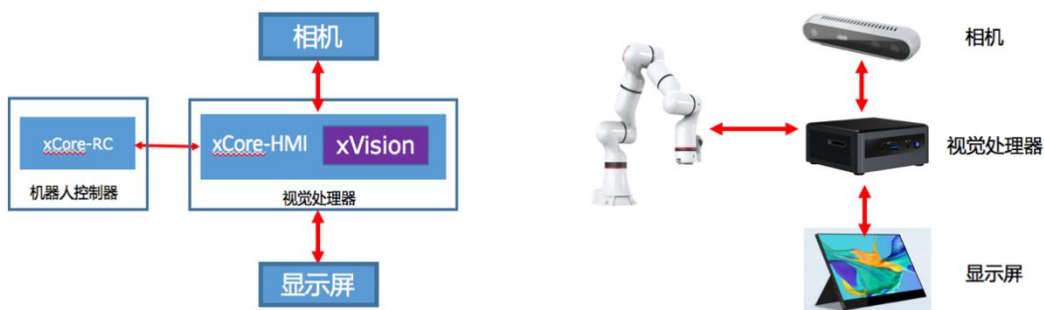
### 2.2 功能特性

- 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

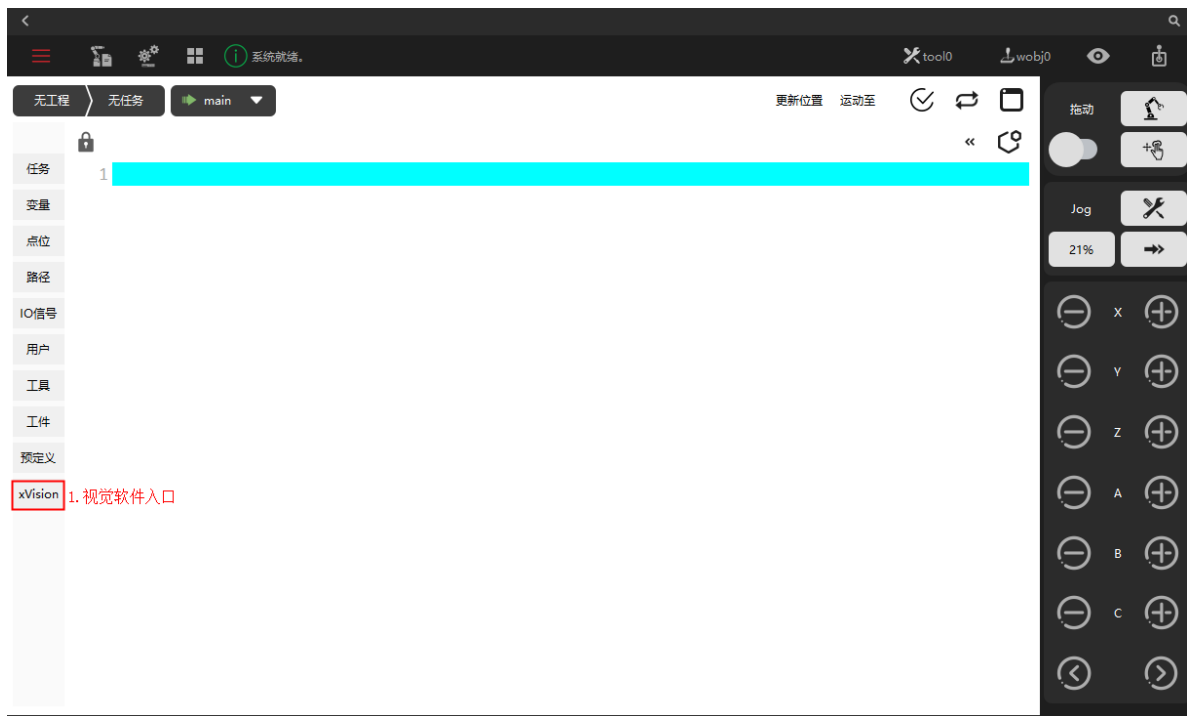
安装由珞石科技提供的 RobotAssist 软件，xVision 已深度整合其中。

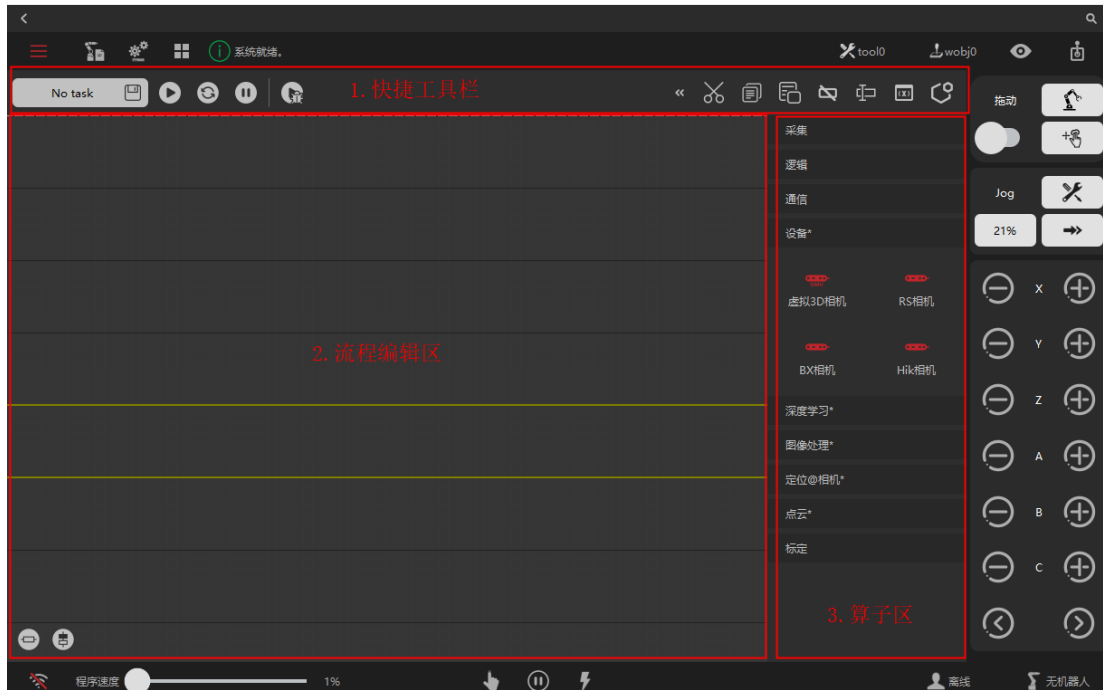
## 3 软件介绍

### 3.1 界面布局

点击左侧边栏的“视觉导图”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；





## 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。

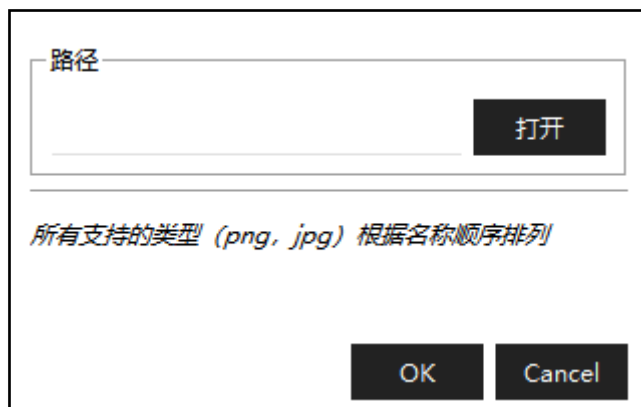
## 4 视觉算子介绍

### 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

#### 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，可通过鼠标缩放。



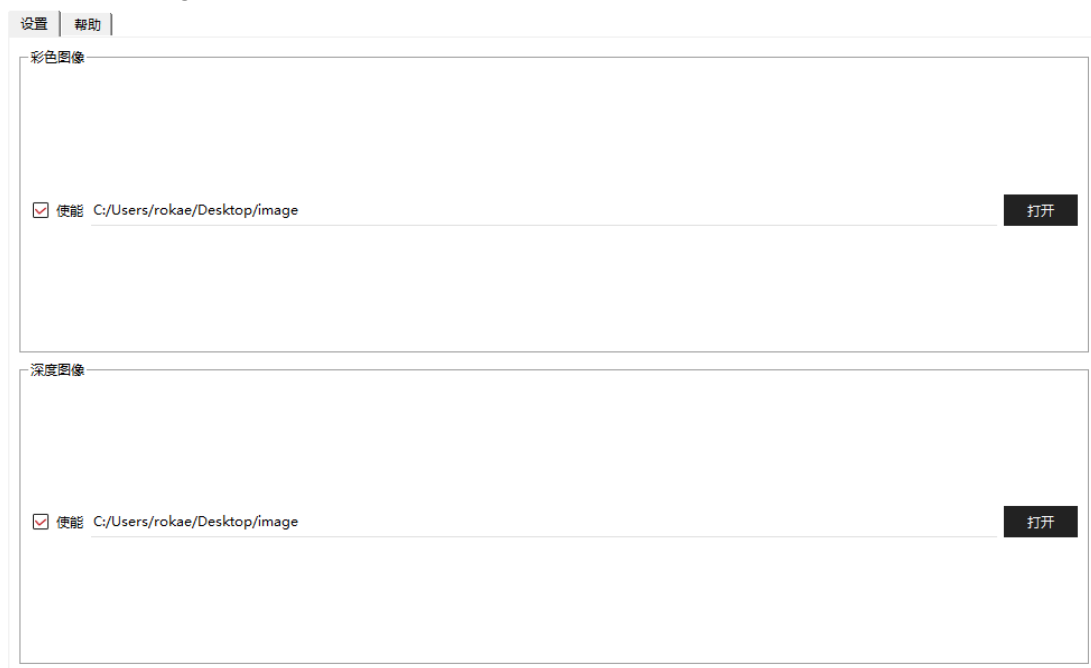
界面

## 4.2 设备

### 4.2.1 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)

### 4.2.2 RS 相机

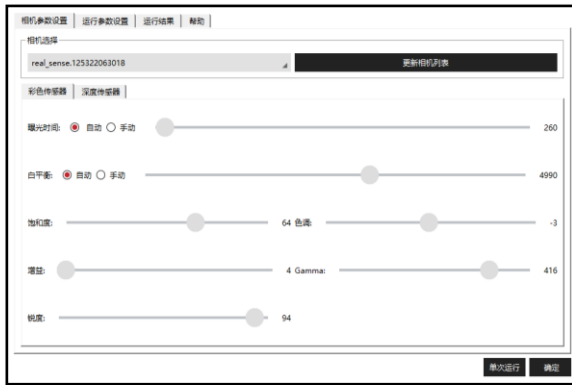
➤ 功能:

作为 Realsense 相机接口。

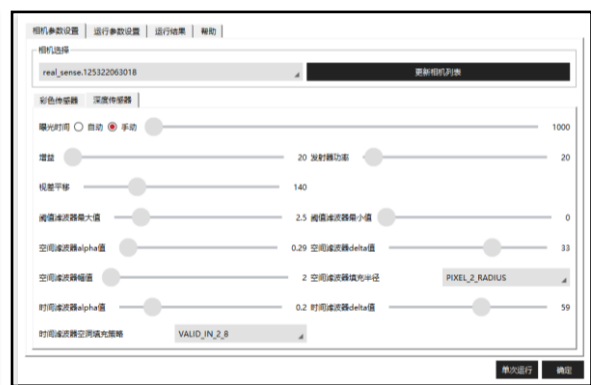
Realsense 是一款 3D 相机, 需要调节的参数主要针对 RGB 图像和深度图像。当相机与视觉处理器正常连接时, “相机选择”部分会自动检测到相机; 选择合适的相机, 点击“单次”或“连续”运行, 会采集并显示图像, 包括: 平面图(RGB 图像)、深度图、红外图。通过相机参数设置, 可以优化相机成像效果。

界面共 4 个 tab:

- 相机参数设置 (彩色和深度传感器, 相机列表选择)
- 运行参数设置 (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)
- 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 (使用步骤说明)



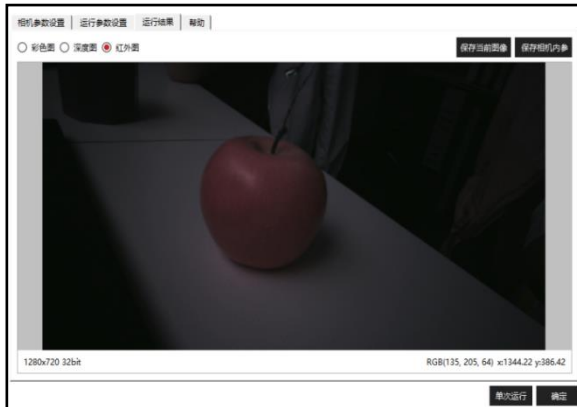
彩色传感器设置界面



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, 然后通过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。
- 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

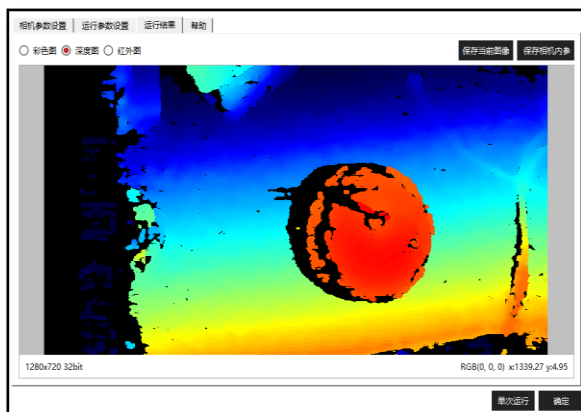


高曝光

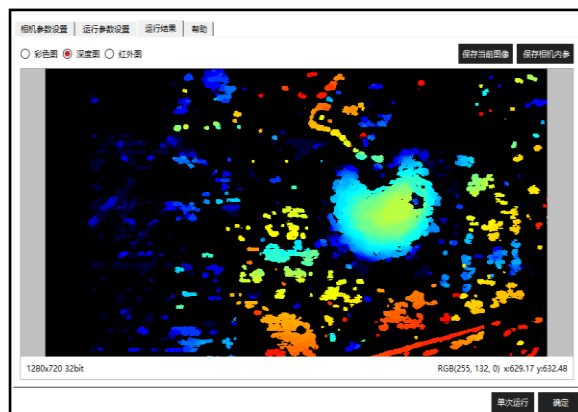
- 3D 相机的深度测量有景深(有效深度的测量范围), 景深可以通过视差平移参数来调整。例如当视差平移设置为 153 时, 景深为 18.5cm~30cm。越靠近物体, 视差平移参数需要调得越大, 相应的景深也越小。

**⚠ 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。

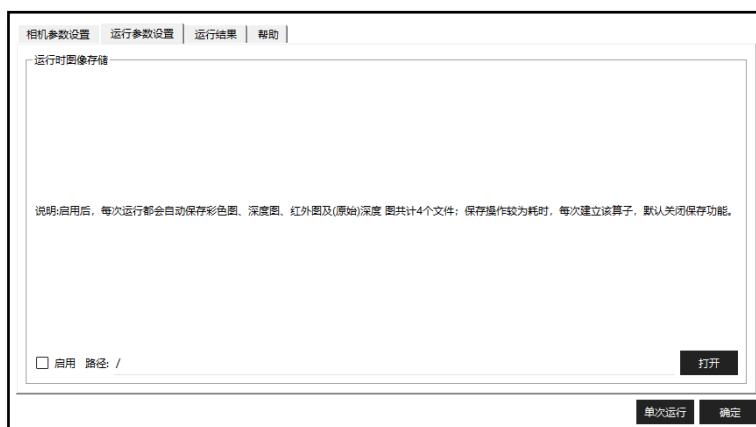


低白平衡

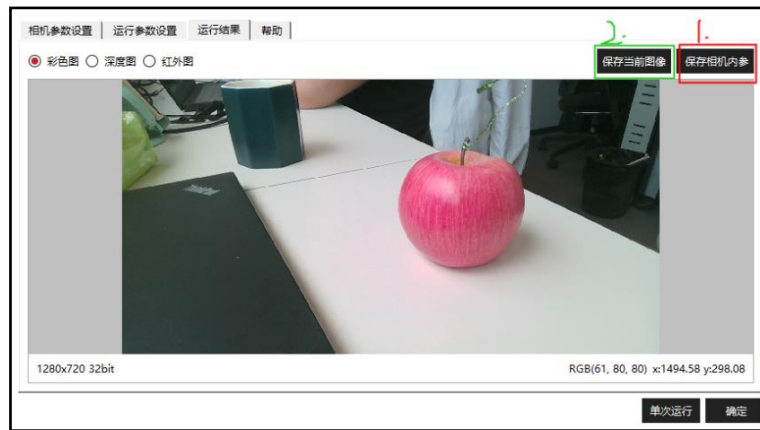


高白平衡

- 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。
- 相机内参可选择保存。
- 当前图像可手动保存。



自动保存图像



保存当前图像与保存内参

### 4.2.3 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

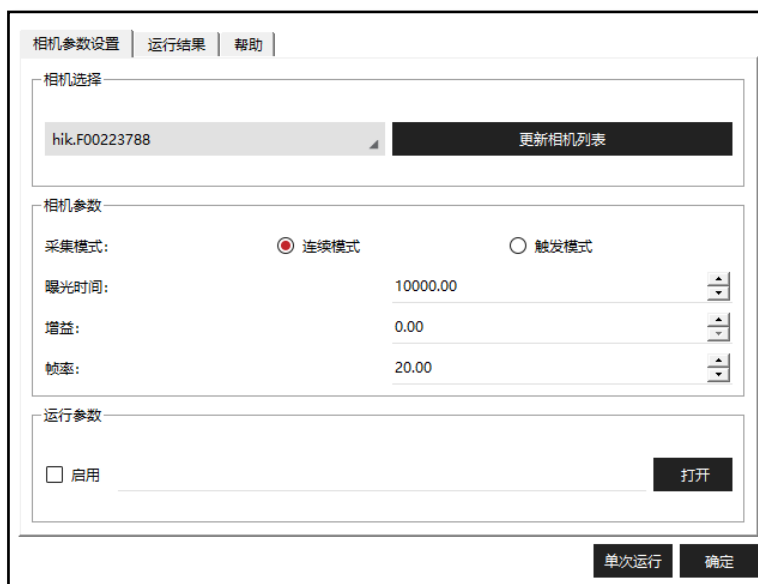
本着精益求精的设计理念，海康机器人推出 CS 系列二代工业相机产品。系列产品从外观设计、产品开发到生产管控，都力求实现技术突破，给原有用户带来使用上的升级体验。同时更加入了丰富的进阶 ISP 功能，减轻用户图像处理的负担。

- 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，运行参数：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（使用步骤说明）



参数设置界面



运行结果界面

#### 4.2.4 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 (彩色和深度传感器，相机列表选择)
- 运行参数设置 (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 (使用步骤说明)



界面

## 4.3 逻辑

### 4.3.1 协议拆分

#### ➤ 功能

将接收的数组信号拆分为独立元素。



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

### 4.3.2 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。



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



### 4.3.3 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：

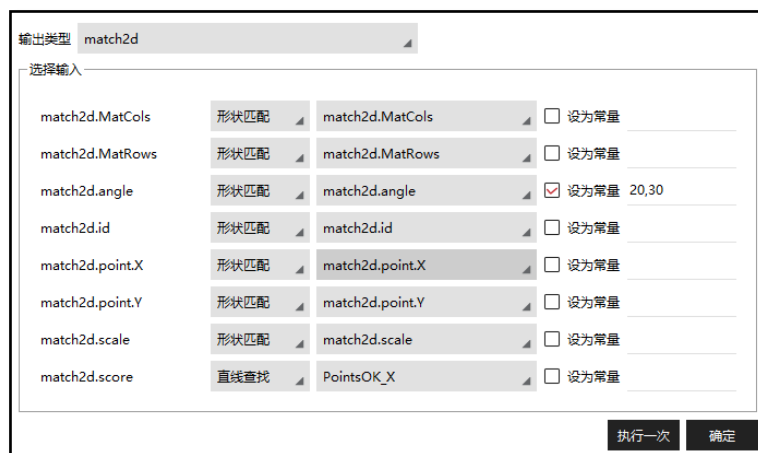
- 输出类型：可选任意输出类型（int、double、point、pose、flag 等）

#### ➤ 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



### 4.3.4 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



## 4.4 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。

### 4.4.1 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）；

## 4.4.2 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；
- 超时：0 表示一直等待，直到所有数据接收完毕；>0 的值指超过该超时时间后还未收到全部数据时终结执行；
- 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}

## 4.4.3 数据发送

### ➤ 功能

将输入的变量按照一定的顺序和格式合并成数据包进行发送。无连接时跳过执行，格式目前进行特殊定制方便现场调试程序。

### ! 注:

- Habot 定制格式是指：标志位在末尾按照数据顺序排列，且用一个数据包全部发送；后续如果有其他需要可以继续定制或者取消定制；
- 运行时禁止修改发送数据，目前尚未报错处理；
- 并列多个发送同时执行先后顺序未知；

界面

### ➤ 参数

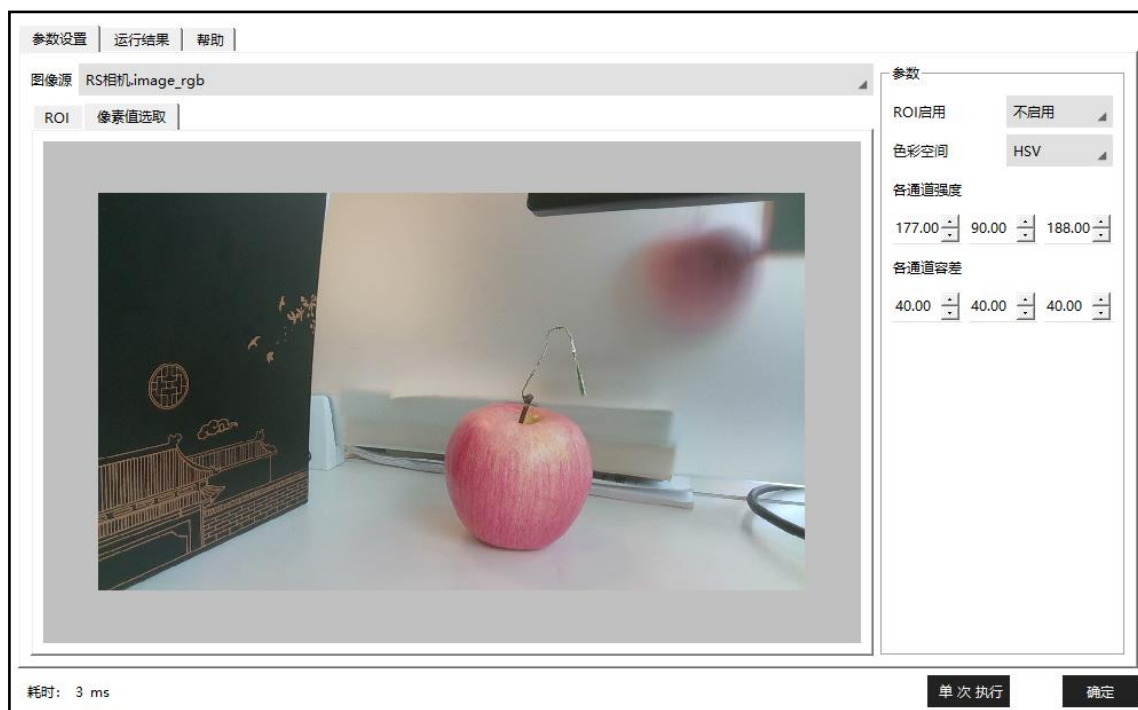
- 格式: CSV、habot; CSV 格式指逗号分割符组成的字符串，每个发送数据为一个数据包，以回车换行符 (ASCII 中的 0x0D，即 '\r')；habot 格式同样使用逗号分隔符但是只用一个数据包一次全发送；
- 标志位: 勾选表示每个数据包除了发送变量还在前面附加一个整数标志位，用以表示发送数据对应的算子执行状态，0 表示算子状态正常，1 表示算子跳过。通常情况下，有错误的算子会跳过执行，即发送 1。对于 habot 格式，标志位按照与发送数据相同的顺序在末尾依次排列，并且都用逗号分隔符合成一个数据包进行发送，可以节省机器人程序代码；
- 增加发送数据: 按照顺序增加指定类型输入变量，可以选择连接输入的输出变量；
- 删除发送数据: 减少输入变量；

## 4.5 图像处理

### 4.5.1 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:

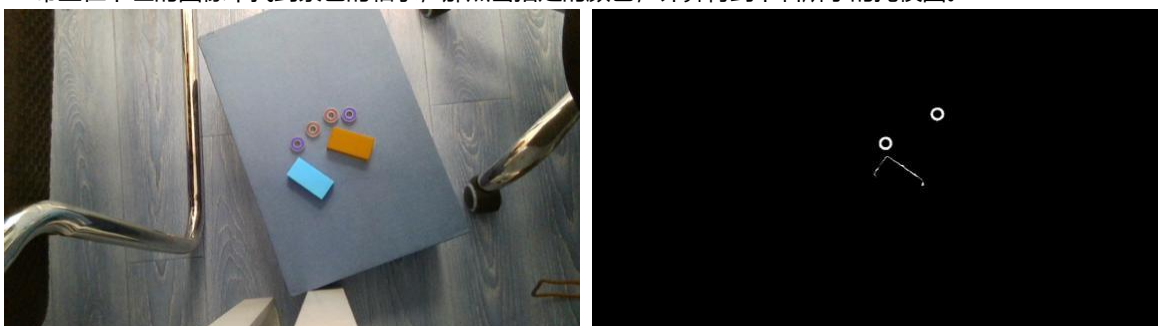
- ROI 启用
- 色彩空间
- 各通道强度 (HSV)
- 各通道容差 (HSV 阈值)

➤ 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

## 4.5.2 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 3 个 ROI: 第 1 个 ROI 检测到白色, 输出对应标志位 1; 第 2 个 ROI 检测到黄色, 输出对应标志位 2, 第 3 个 ROI 未检测到白色或黄色, 输出默认标志位-1。

相应的结果可以通过网络调试助手接收并查看。



网络配置

#### 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
- 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
- 根据任务在 1 中的图片中框选 ROI，选完之后点击 ROI 确认按钮，点击单次执行即可得到结果

### 4.5.3 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。

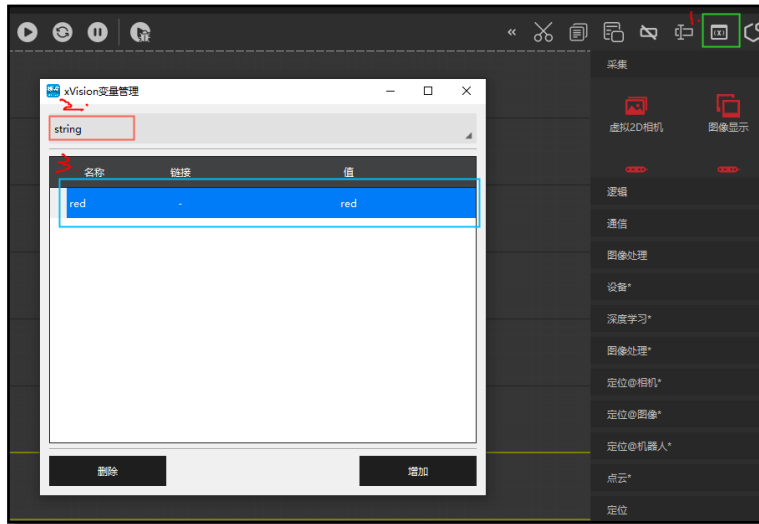
颜色分割模板参数	
HSV 阈值	H 为色相，S 为饱和度，V 为亮度，具体含义参见百科。通过设置阈值限制物体与标签模板相近的颜色。

#### 参数:

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

#### 使用步骤:

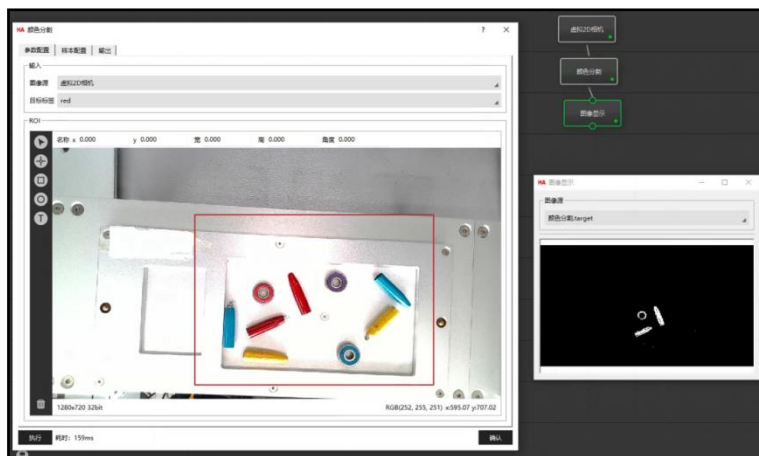
- 选择图像源；
- 目标标签可以在全局建立或者从数据接收算子建立。全局建立如下图所示：



建立变量



- 参数配置页面画 ROI



- 点击执行，结果页面查看

#### 4.5.4 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、黑帽运算、顶帽运算、梯度运算；

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场合，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；

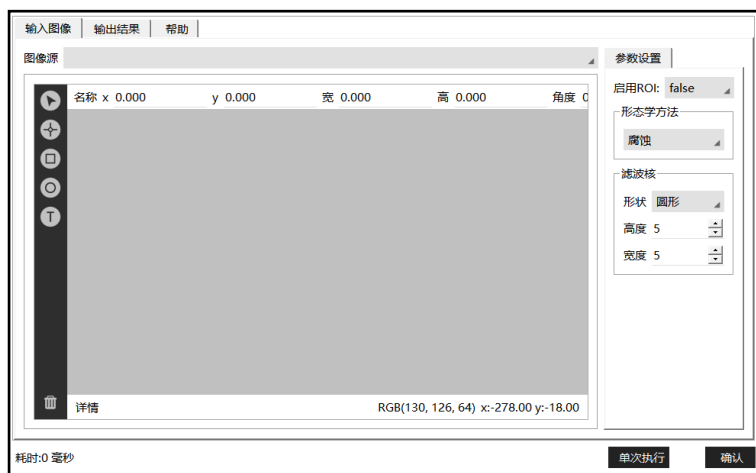
开运算: 开运算=先腐蚀后膨胀，开运算与闭运算相反；

黑帽运算: 是进行闭运算后的图像与原始图像做差而得到；顶帽操作往往用于分离比临近点亮一些的斑块。

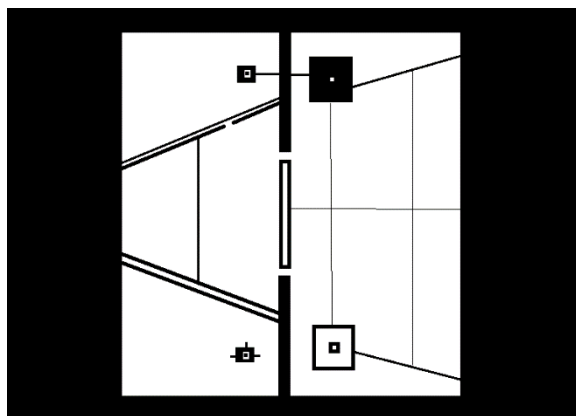
顶帽运算: 又称礼帽运算，是原始图像与进行开运算之后的图像做差而得到；黑帽操作往往用于分离比临近点暗一些的斑块。

梯度运算: 膨胀后的图像与腐蚀后的图像做差得到，可得到白色部分的轮廓像素点；

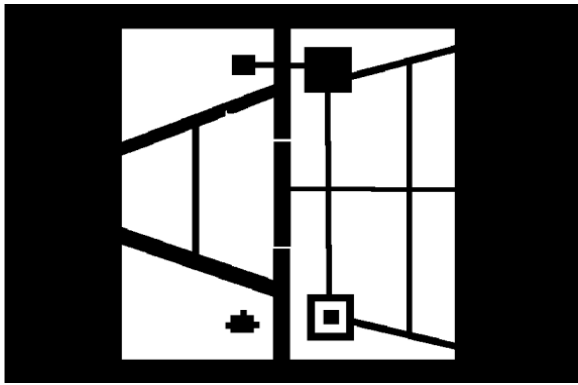
- 滤波核包括: 矩形、椭圆、菱形；可以通过宽度和高度设置滤波核大小。
- 是否启用 ROI: 仅处理感兴趣区域。



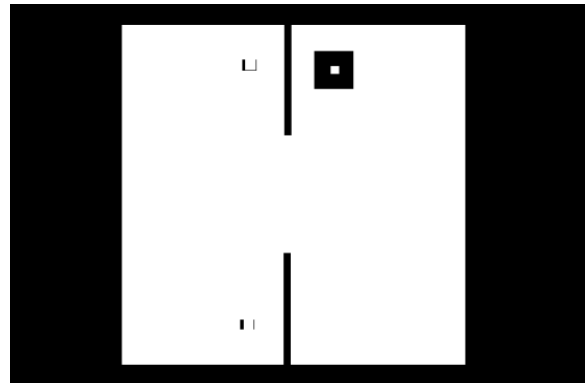
##### ➤ 样例



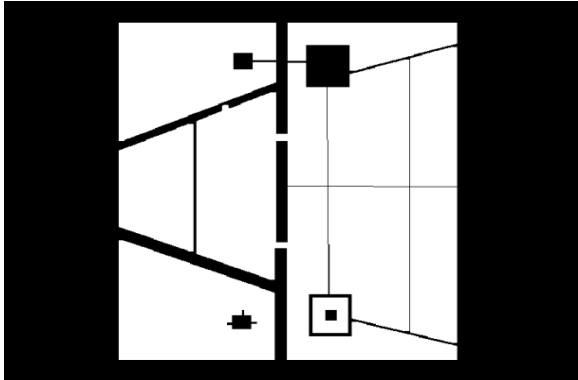
原图



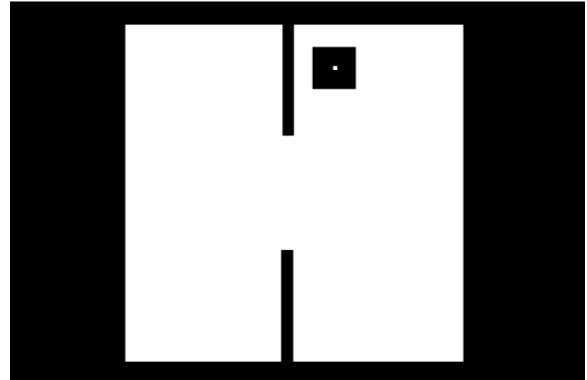
腐蚀



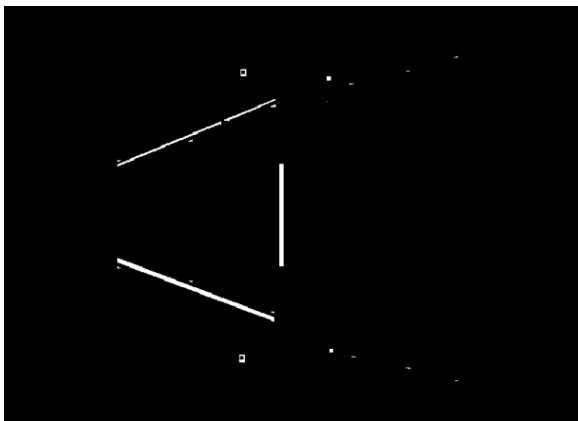
膨胀



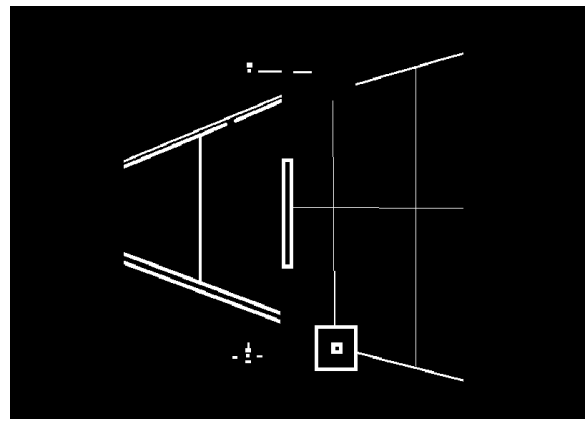
开运算



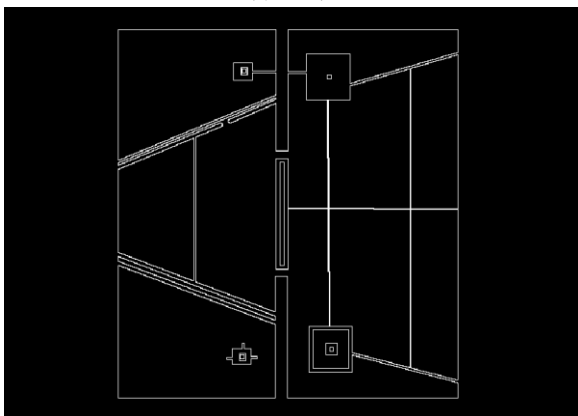
闭运算



顶帽运算



黑帽运算



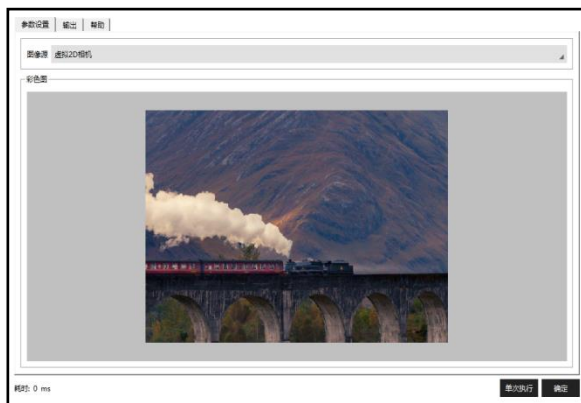
梯度运算

### 4.5.5 彩色转灰度

➤ 功能:

xVision 用户手册 v1.0.0

将输入的彩色图像转为灰度图像并输出



彩色图



灰度图

### 4.5.6 深度补偿

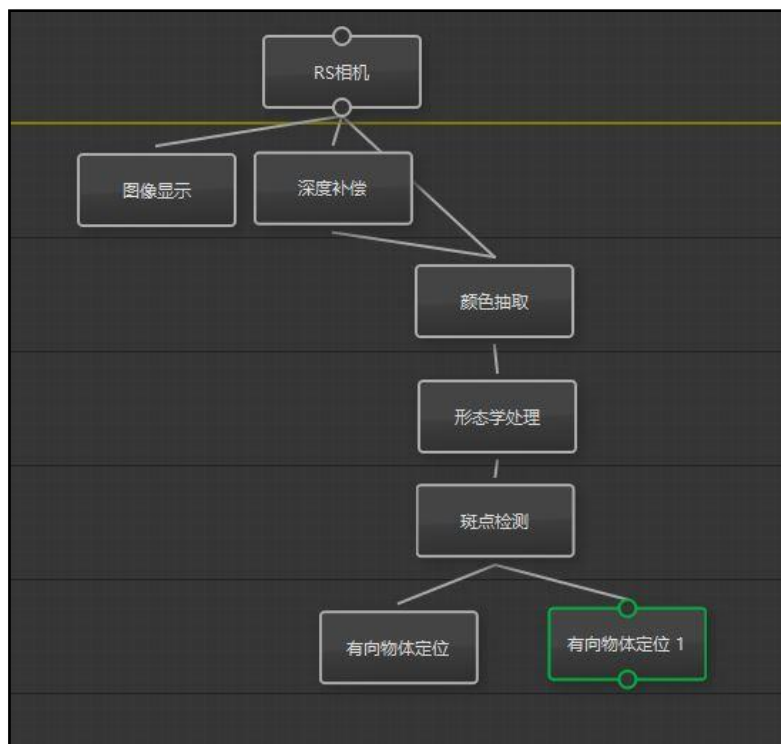
➤ 功能:

Realsense 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。

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

⚠ 注:

- 后续算子的深度图输入选择“深度补偿.depth\_refined”,不要选“深度补偿.refined\_color”或“深度补偿.before\_color”



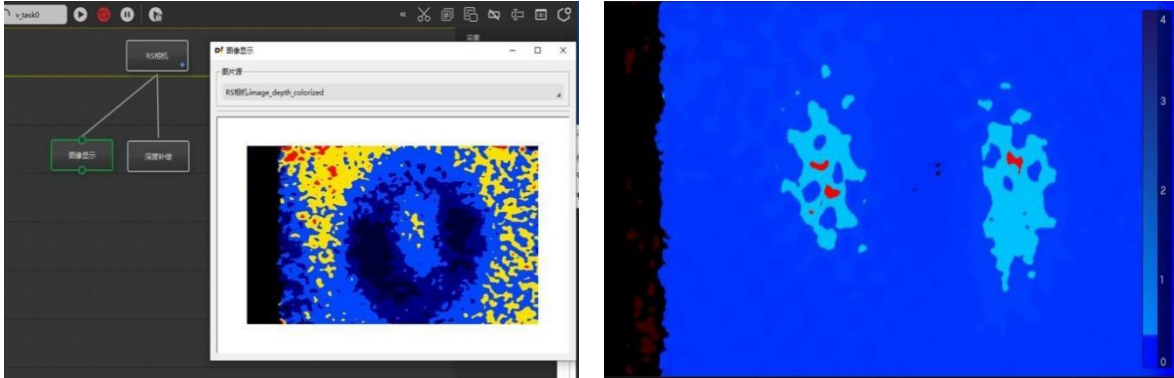
流程样例

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。

➤ 使用步骤:

**补偿计算阶段：**

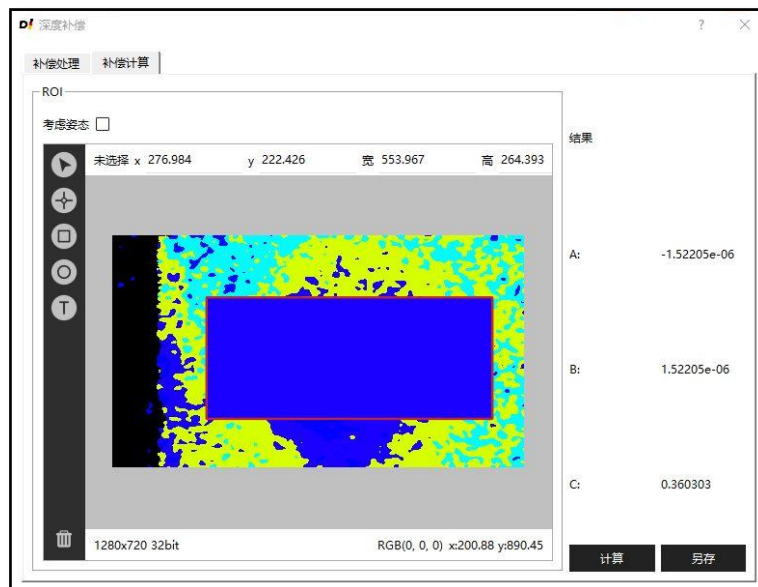
- 找一个平板(尽可能平、最好不是黑色、且不反光)，大小覆盖相机视野范围；
- 将该平板平放于摆放工件的平面上(平板与平面平行)；
- 调节相机与平板平行，判断是否平行可以通过 xVision 图像显示算子(下图左)或者 Realsense Viewer(下图右)观察深度图。下图就是一个典型的水平状态：平板各个位置深度值基本一样，中间布局会出现内凹或外凸，这些局部误差正是深度补偿算子需要处理的。**为了保证效果，尽量调整至水平状态！**



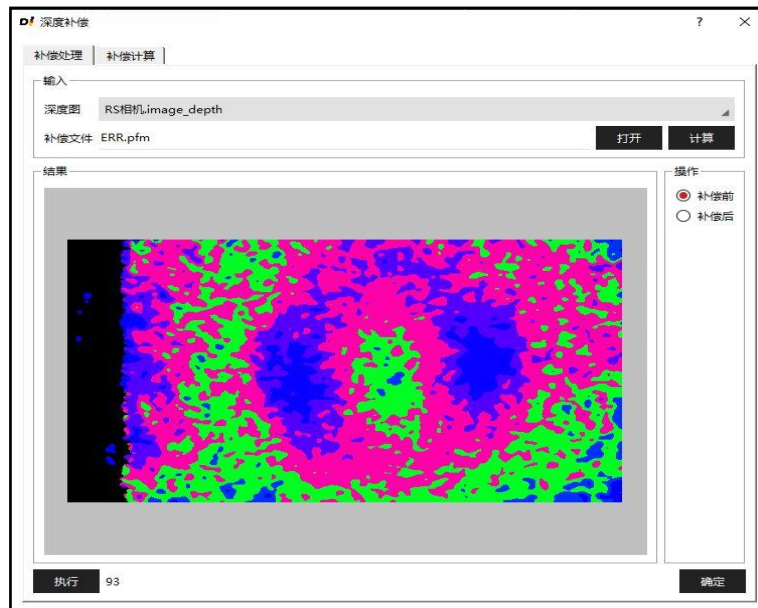
- 调节完成后，在深度补偿算子的“补偿处理”页面选择深度图来源；
- 在算子的“补偿计算”页面设置矩形 ROI 区域，后续只对 ROI 范围内的数据做计算和补偿处理；“考虑姿态”默认不勾选，勾选后，后续补偿会考虑后面将介绍的 A 和 B 参数。
- ROI 设置完成后，点“计算”，根据处理器性能不同，计算可能会耗时 1~6 秒，完成后会显示结果 A、B、C，这是拟合出来的平板所在平面参数，平面方程为  $Z=Ax+By+C$ 。A、B 越小(最好小于  $10E-6$  量级)，说明相机与平面越平行；C 代表相机与平面的距离，单位 m。如果觉得这三个参数比较合理，可以点“另存”，存储一个格式为 pfm 的误差补偿文件。

**! 注：**

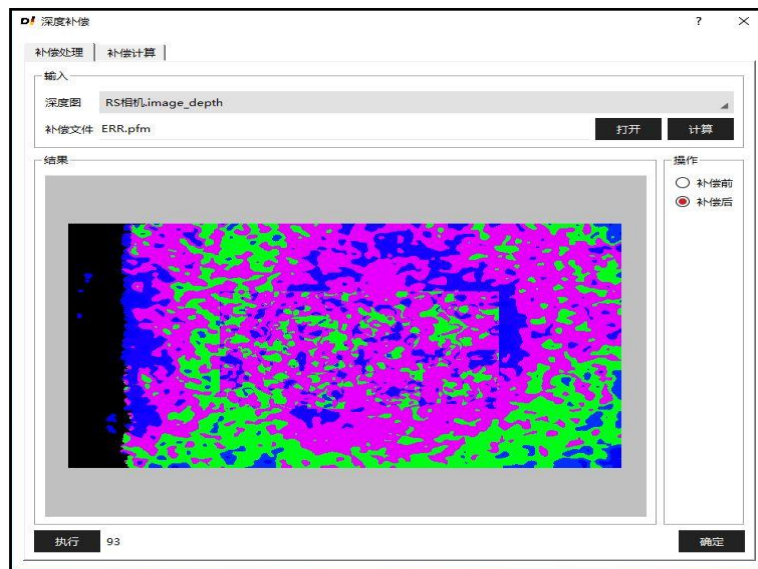
如果没有“另存”，误差补偿文件会默认覆盖 ERR.pfm 这个默认文件，当有多个深度补偿算子时，可能会对其他深度补偿算子计算造成影响；所以建议用户单独另存。

**补偿处理阶段：**

- 选择深度图来源，以及补偿文件。点“补偿计算”按钮会切换至补偿计算页面。
- 点击“执行”，可以单次执行补偿，并查看对比补偿前、后的效果。



补偿前



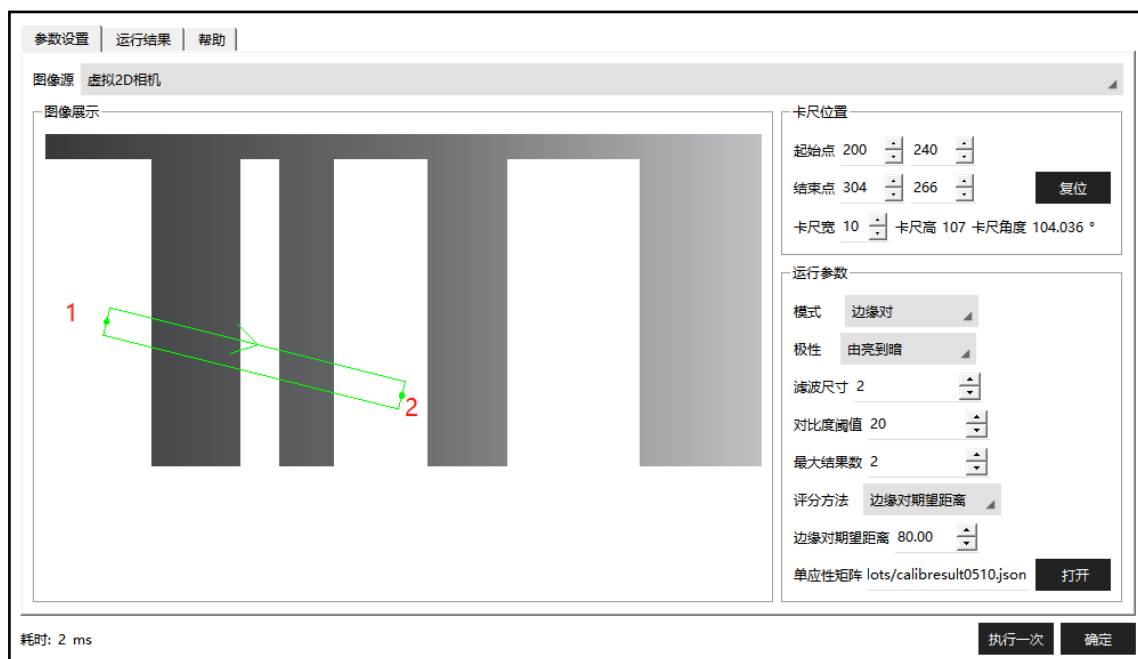
补偿后

## 4.6 定位（图像）

### 4.6.1 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。

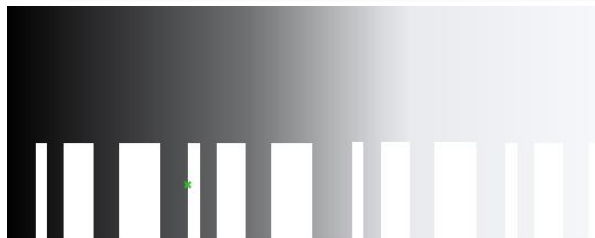
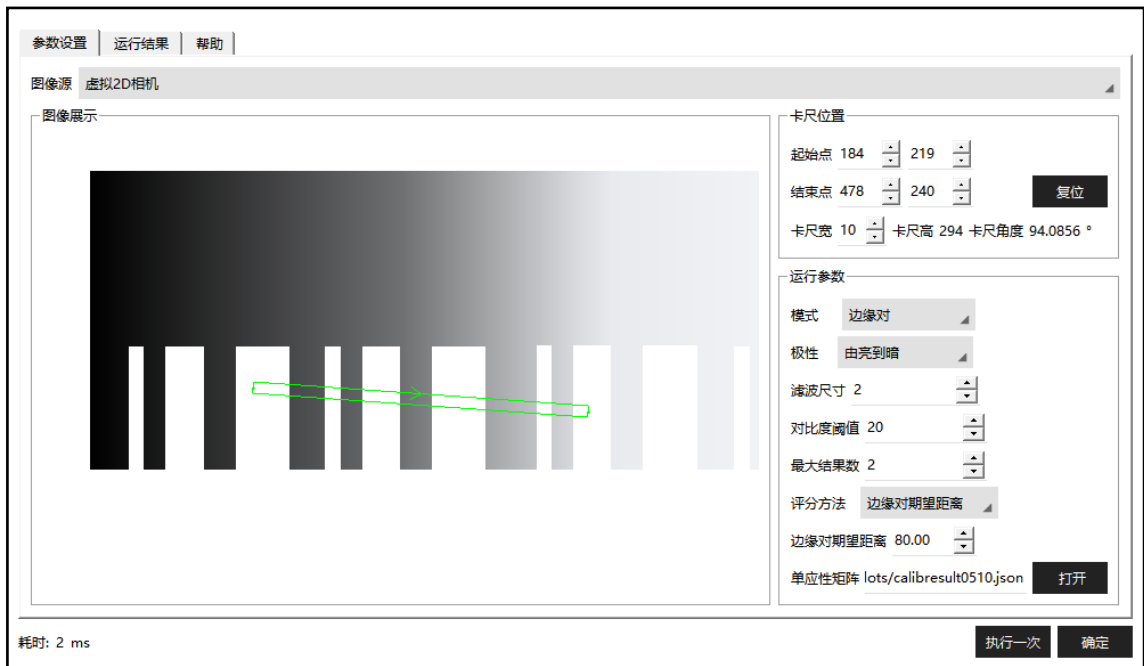


界面

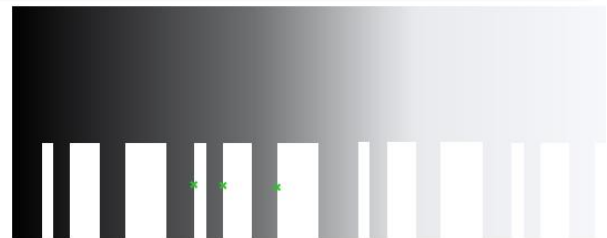
#### ➤ 参数：

- 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。

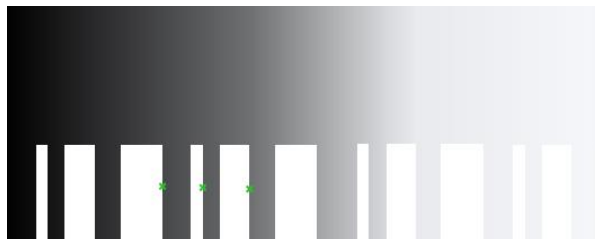
#### ➤ 样例



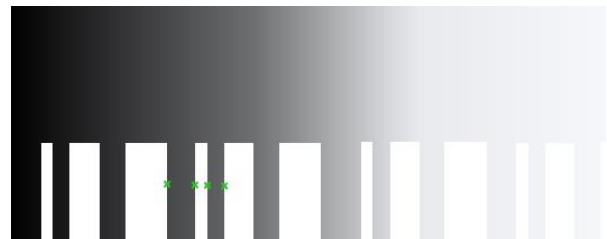
找 1 个最大对比度的边缘点，极性由暗到亮



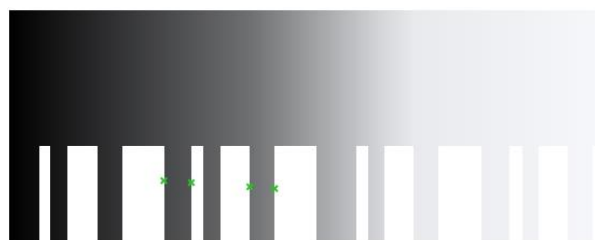
找 3 个最大对比度的边缘点，极性由暗到亮



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

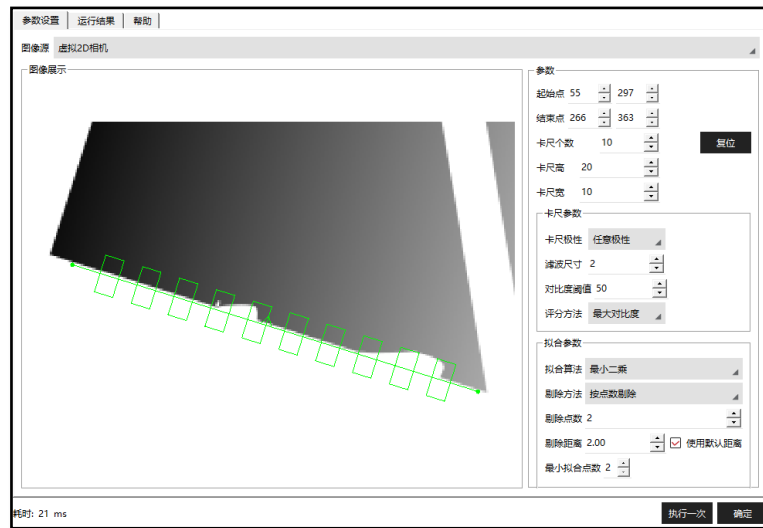


找 2 组期望距离最接近 20 的边缘对，极性由暗到亮

## 4.6.2 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。

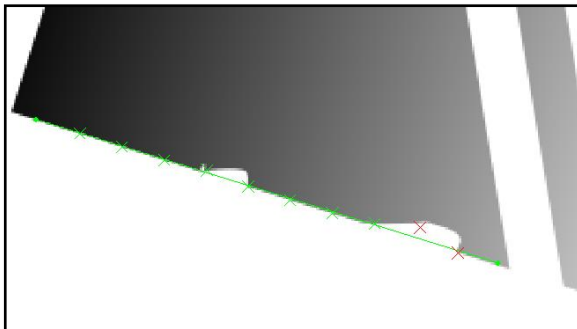


界面

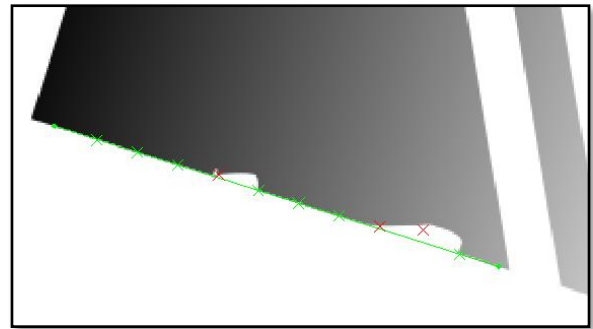
### ➤ 参数

- 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：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 EEROR
- 误差阈值：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

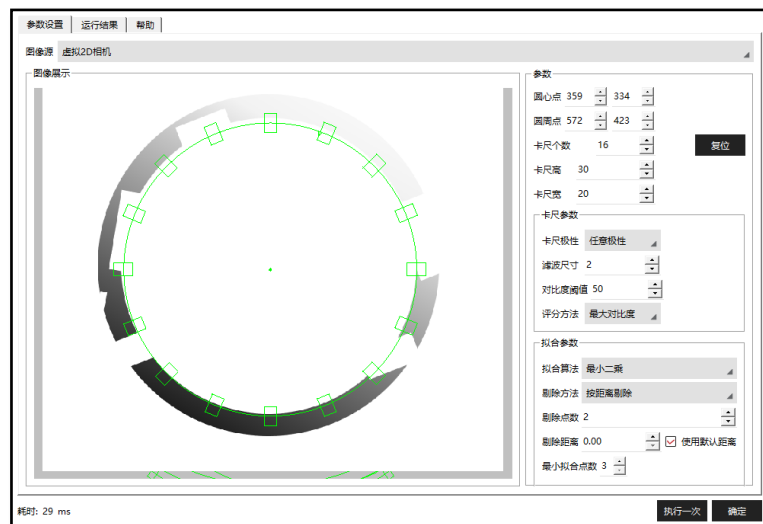


拟合方法为 RANSAC  
可以看到自动剔除了 3 个误差最大的点

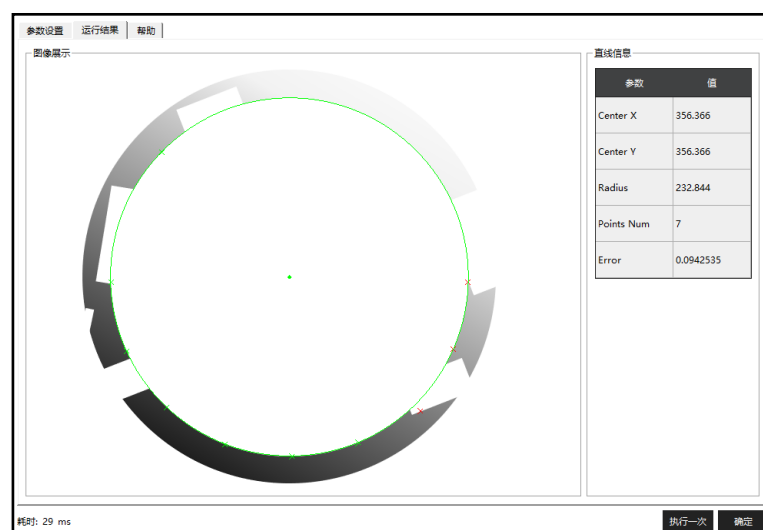
## 4.6.3 圆查找

### ➤ 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

#### 4.6.4 斑点检测

##### ➤ 功能:

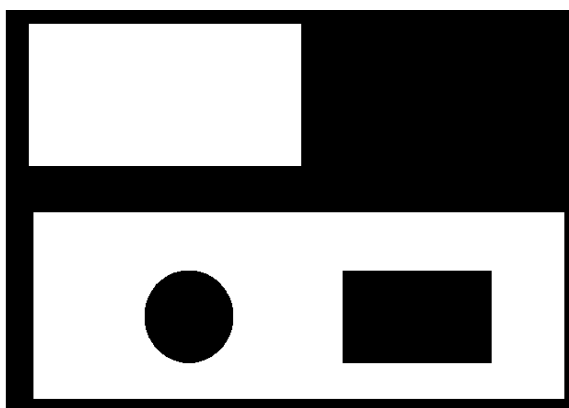
斑点检测主要用于检测和定位图像中指定灰度范围内、形状通常不可测的区域。使用斑点检测可以对图像中是否有斑点，以及斑点数量、位置、形状等特征进行定位。通过对斑点进行面积过滤、圆度过滤、矩形度过滤，可以对检测到的斑点进行筛选，从而得到预期的斑点。利用得到的检测斑点，可以用作后续其他算子粗定位或零件表面缺陷检测。



### 参数

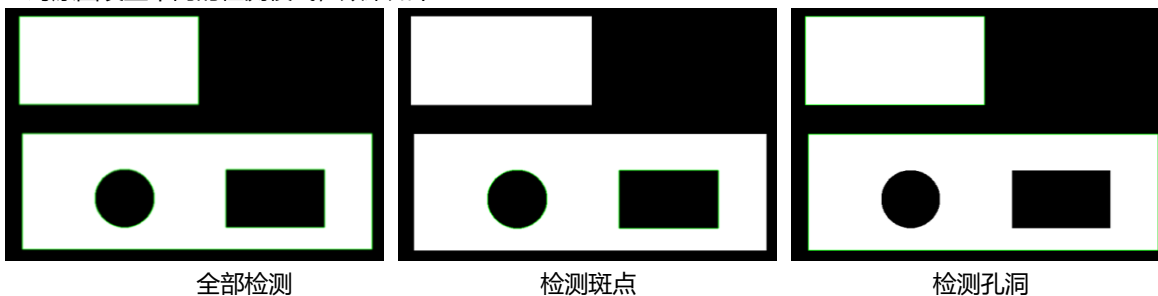
- 图像源：待检测图像，要求图像为灰度图；
- 二值化阈值：包含自动选取阈值，手动选取单阈值，手动选取双阈值等。
- 最大检测数：设置检测到的斑点数量上限；
- 检测模式：检测斑点、检测空洞或全部检测；（所谓空洞、斑点，其实是两种相对的概念，举个例子：白纸上放个黑块，就认为黑块是个斑点；黑纸上放个白块，就认为白块是个空洞）；
- 排序：当使能排序后，可以对检测到的斑点结果进行排序，可选择升序或降序；
- 排序要素：排序依据，支持根据面积、周长、质心 x、质心 y、圆度、矩形度进行排序；
- 面积过滤：通过面积对滤去除不想要的斑点结果，面积过滤指的是像素面积；圆度/矩形度越接近 1，相应的斑点越近似于标准的圆/矩形；
- 圆度过滤：通过圆度对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；
- 矩度过滤：通过矩形度对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；

### 样例

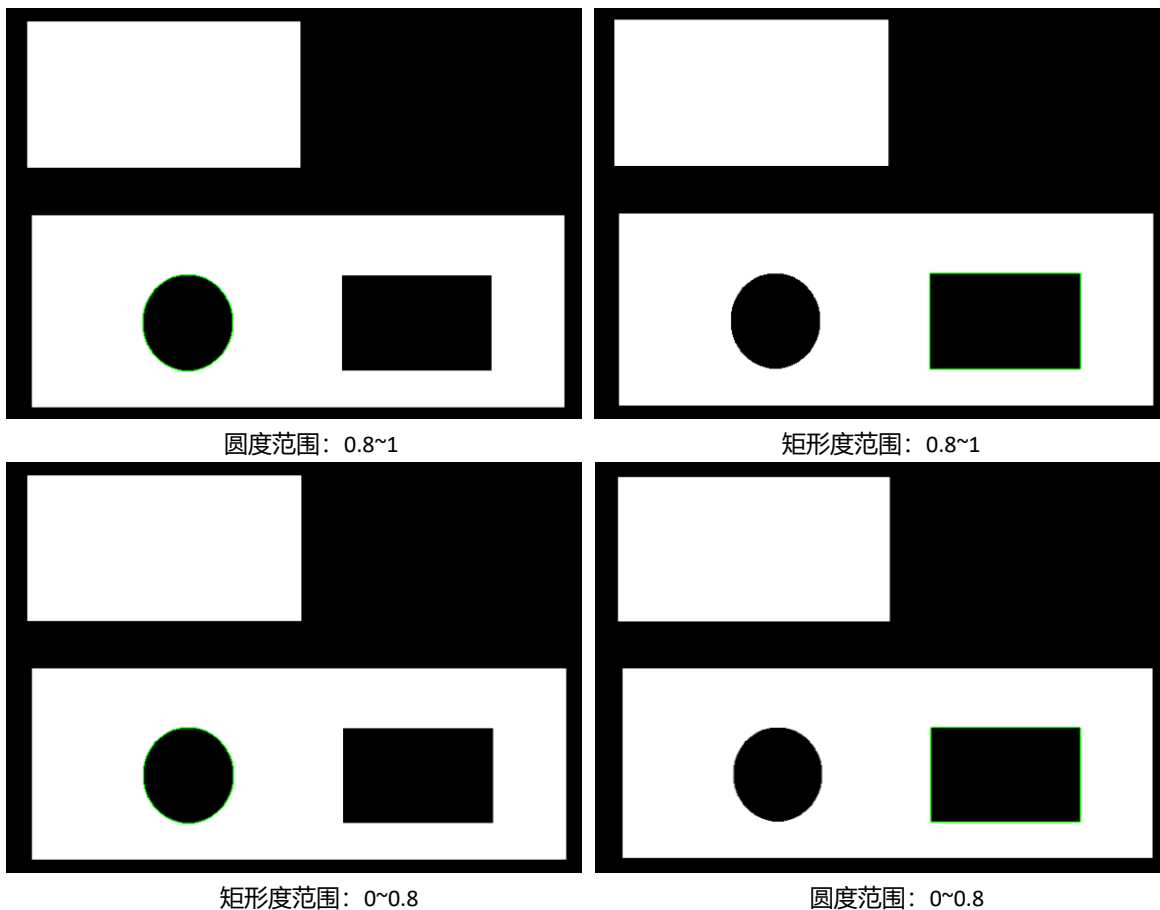


原图

对原图设置不同的检测模式，效果如下：



在斑点检测的基础上，结合圆度和矩度过滤，进行斑点筛选，效果如下：



## 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。

#### ⚠ 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。

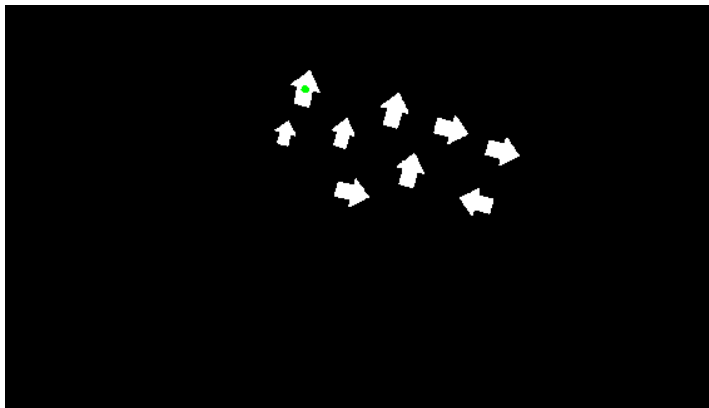


## 界面

- 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围矩形，通过长度阈值上限和下限过滤掉不想要的斑点。此处的长度阈值指物体的实际包围框长度（单位为 mm）。尖端监测范围为相对值（取值在 0~0.5 之间）。

物体的方向默认根据最小包围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，最小包围矩形的方向(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围框长度方向，取多少比例的区域用于计算像素分布。



界面

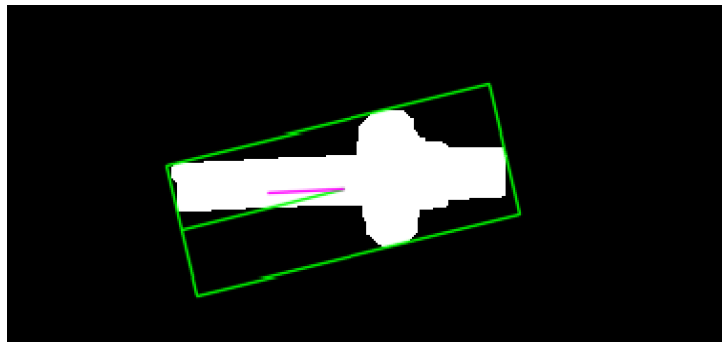
#### ➤ 参数

- 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 mm）；
- 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；

#### ➤ 样例：



流程图

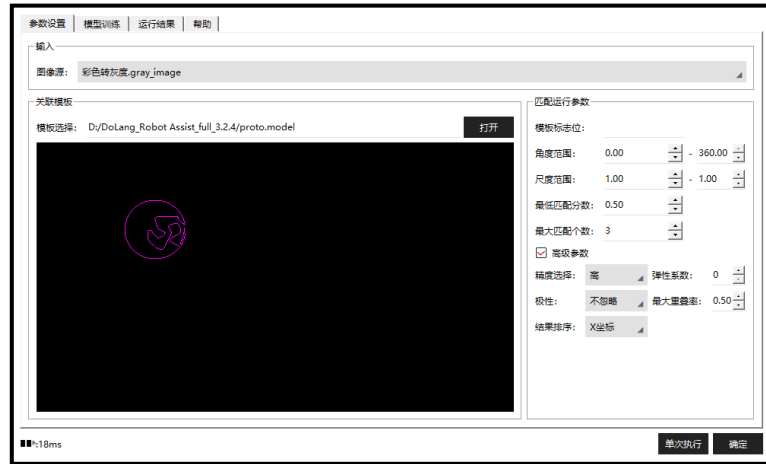


结果图

## 4.7.4 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

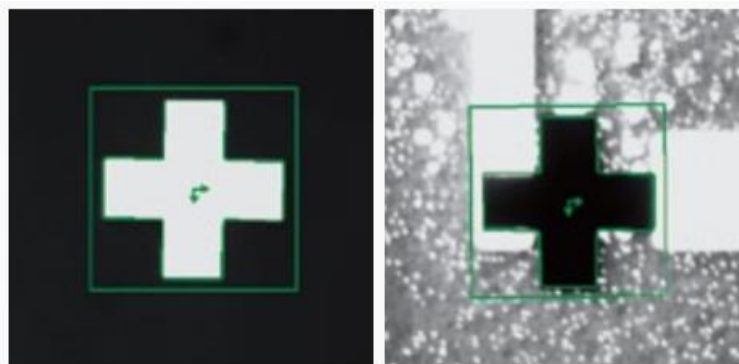


运行结果界面

## ➤ 参数:

- 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

- 模板选择：用于匹配的模板
- 模板标志位：用户自定义的数字或字符，可以作为结果的一部分输出给其他算子，当工程中有多个模板匹配算子时，可以用于区分结果来自哪个模板匹配算子；
- 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
- 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

- 排序：通过分数、x 坐标、y 坐标的一个对结果进行排序
  - 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
- 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
  - 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

#### 4.7.5 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。

获取 3D 位姿算子的输入是模板匹配的输出，可以选择若干个匹配结果。



参数配置界面

运行结果界面

### ➤ 参数

- 图像源：输入灰度图；
- 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；
- 匹配方式：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；
- 圆半径：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；
- 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；

## 4.8 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。

### 4.8.1 物体位姿

#### ➤ 功能：

确定目标物体相对机器人基坐标系的位姿。



界面

#### ➤ 参数

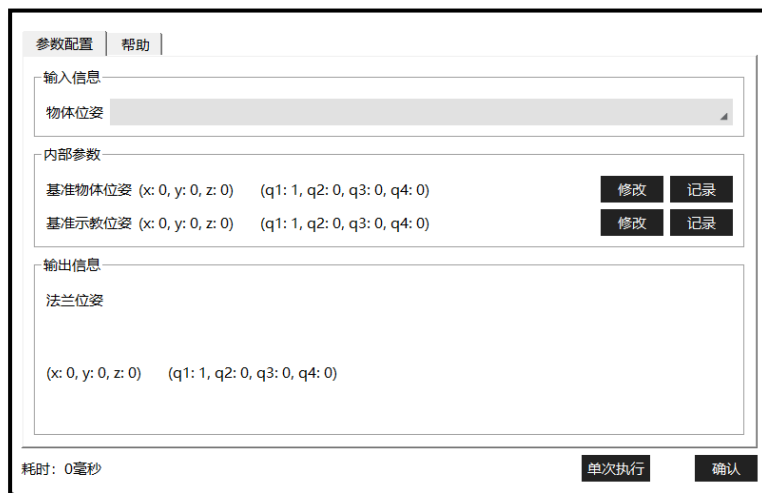
- 物体相对相机位姿：通常来自位姿转换或有向物体定位算子，通常由定位@相机中的相应算子提供；

- 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。

## 4.8.2 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。



界面

### ➤ 样例：



流程图



流程图

## 4.9 标定

### 4.9.1 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。

### 4.9.2 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

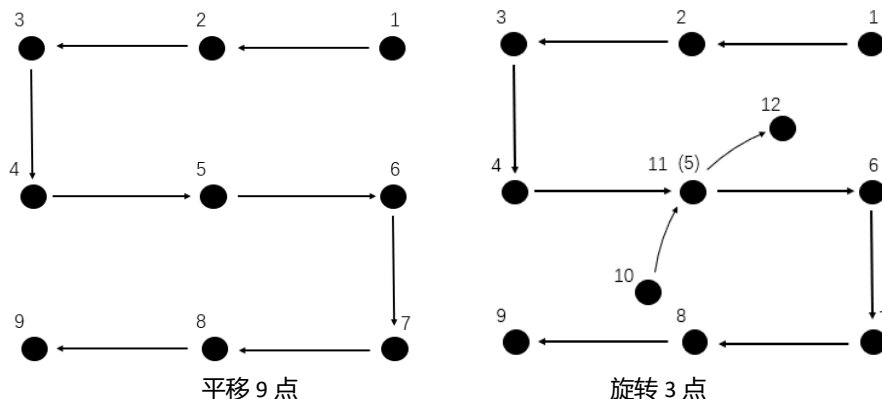
#### 参数:

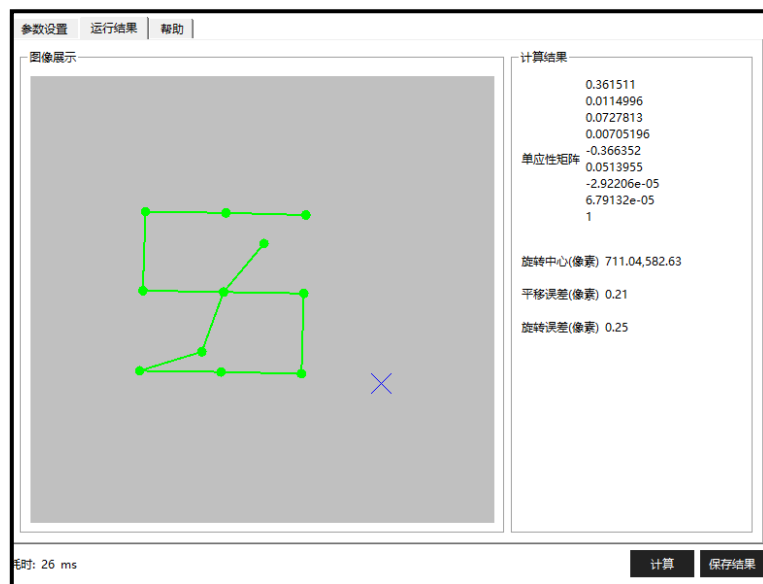
- 输入源 可选择从“模板匹配结果+机器人运行位姿”，算子将获取“形状匹配”算子的结果作为输入；若已有结果，也可选择从文件导入，导入的 json 文件必须含有机器人位姿、形状匹配位置等关键字。
- 平移次数 可选 3x3 或 5x5
- 旋转次数 可选 3 或 5
- 相机安装 相机安装方式：可选相机运动、相机运动-拍照位置非固定、相机静止-上相机、相机静止-下相机

#### 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

### 4.9.3 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:

- 形状匹配结果
- 标定文件: N 点标定的结果文件
- 转换结果: 当前形状匹配结果与其转为世界坐标系的位置 (相对基准点的位置)。

### 4.9.4 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

## 4.10 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:

用于显示点云加载算子加载的点云文件。界面应显示所加载点云的 XYZRGB 信息。目前点云只显示为白色，后续更改为可显示颜色信息。



界面

➤ 参数:

- 点云源: 指定点云来源

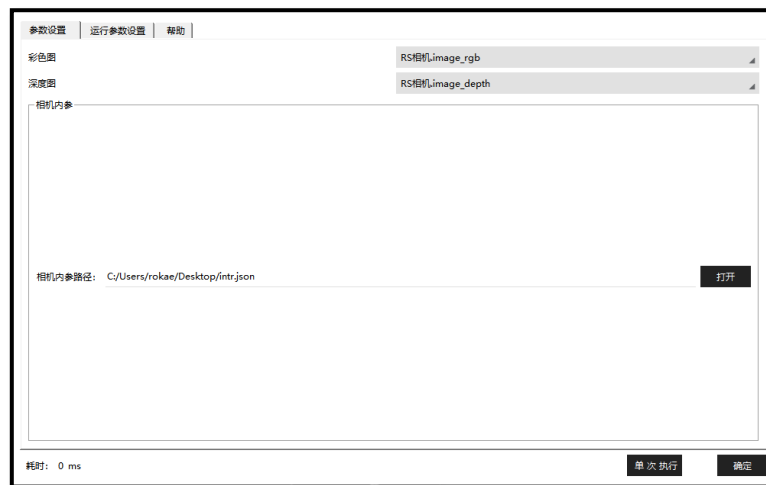
### 4.10.3 深度图转点云

➤ 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)
- 运行参数保存 (保存的地址和是否启用)
- 帮助 (使用步骤说明)



界面

➤ 参数:

- 彩色图
- 深度图
- 相机内参

## 4.11 深度学习

### 4.11.1 深度学习检测

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 detection: 深度学习目标检测的原始输出图片
- 输出 best\_object: 只包含最优抓取物体的二值图

## 5 案例介绍

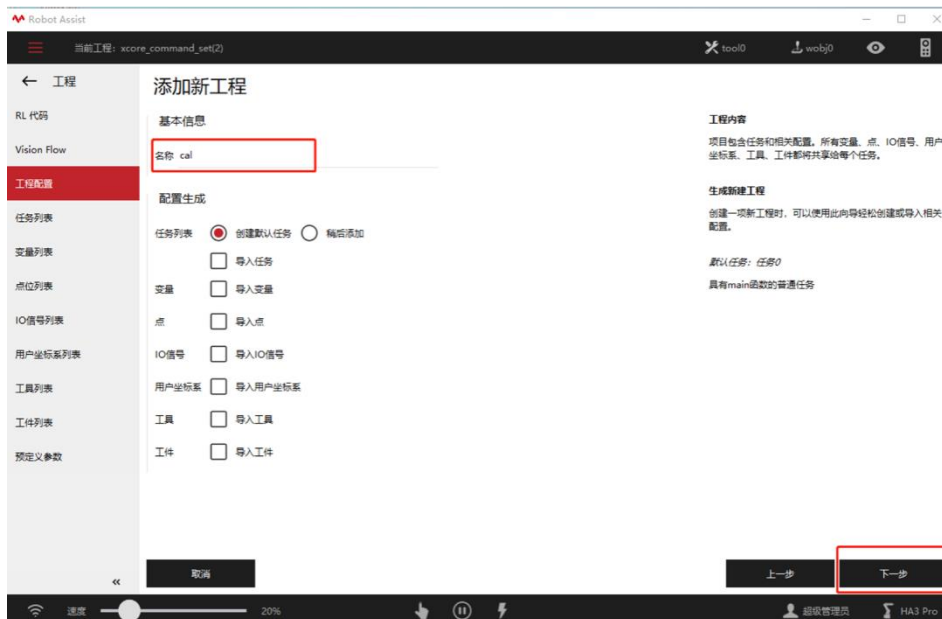
机器人视觉引导程序一般由两部分构成：(1)机器人程序，(2)视觉程序。二者通过网络进行通信。以下通过几个典型案例，介绍 xVision 如何配合机器人进行视觉引导作业。

**\*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。

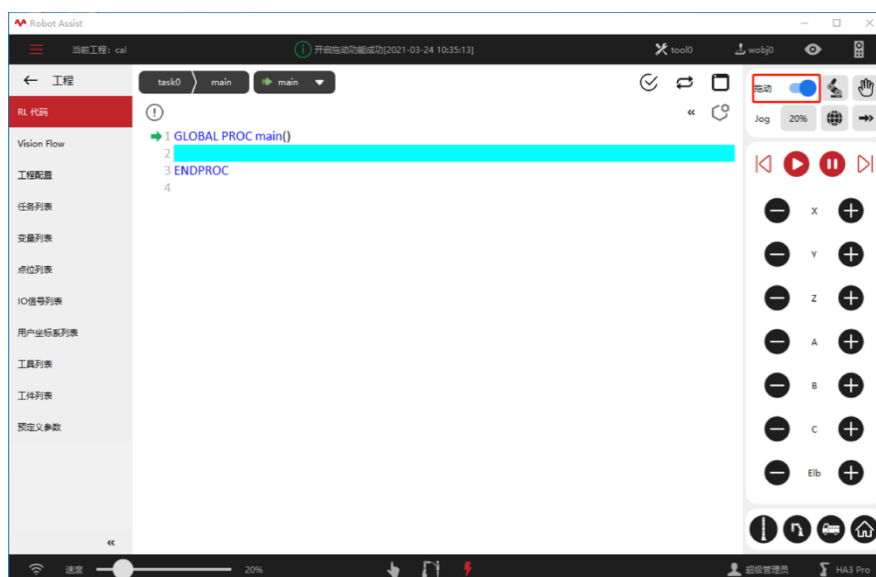


2. 创建完工程后，在 RL 代码区会自动生成两行机器人程序，如下图，点击“机器人操作面板”。

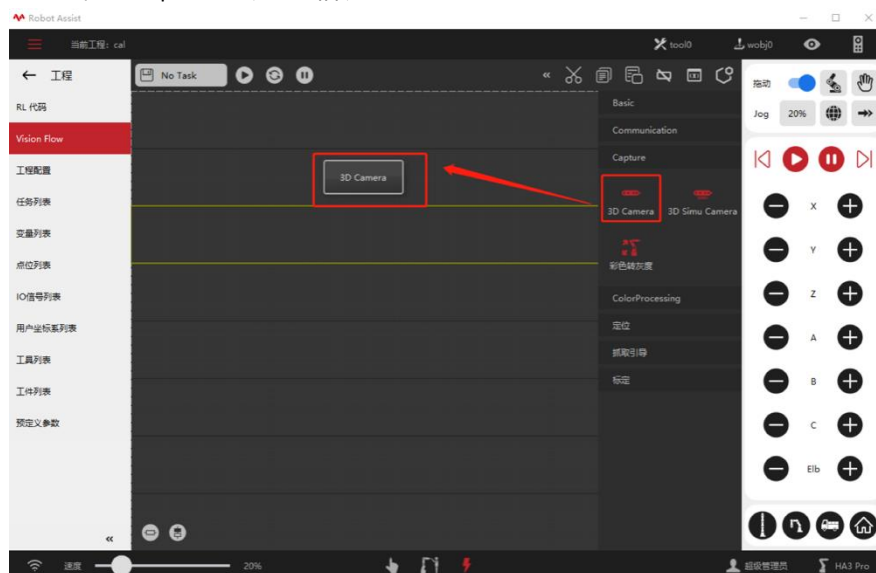


3. 进入‘机器人操作面板’后将“拖动”按钮打开，当正取连接机器后，点击该按钮后，机器人会发出“咔”的一个声响，表明机器上电，并进入了拖动模式，此时用手指同时按住法兰旁边的两片黑色按键，抓着机械臂是

可以轻易的拖动机械臂的。



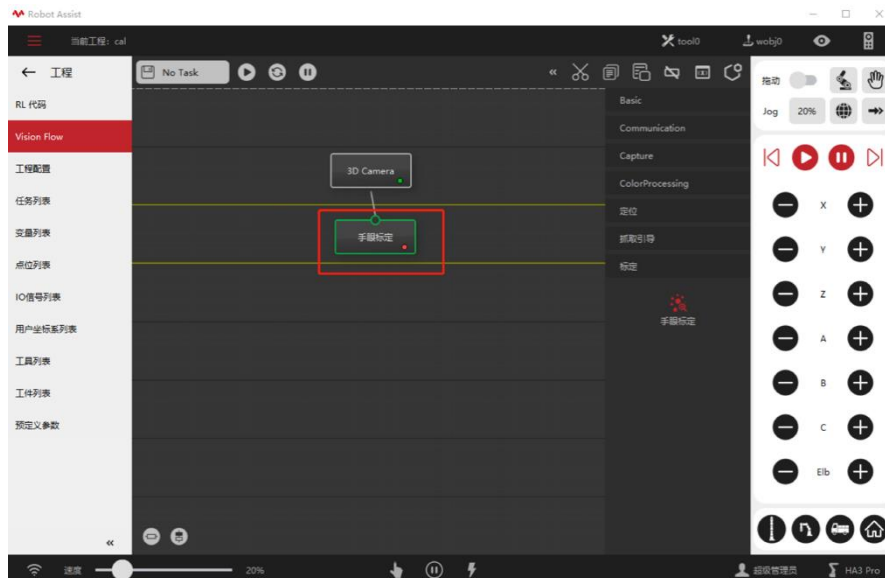
4. 打开“Vision flow”，从 Capture 拖动一个相机到工作区。



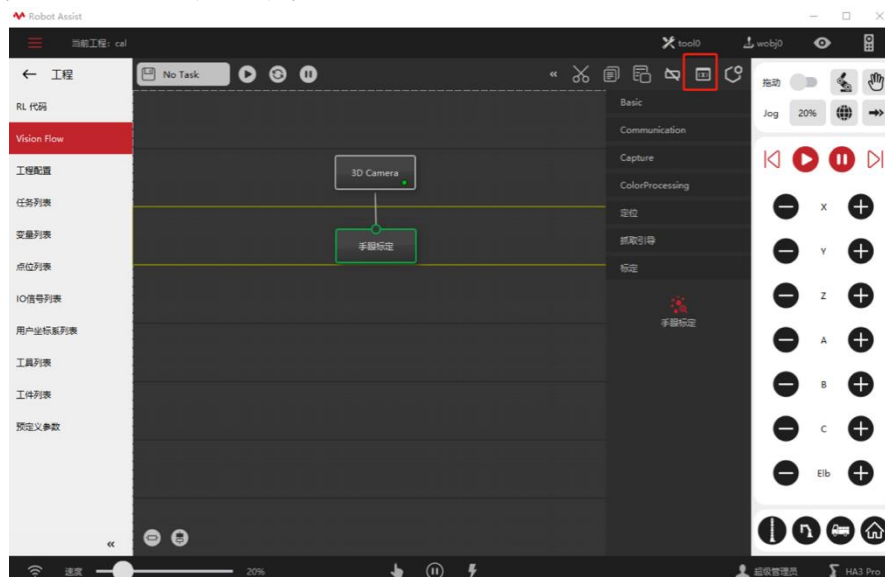
5. 双击工作区中的相机算子，就会弹出相机算子预览窗口，选择“平面图”并点击工作区左上角的视觉程序启动按钮，在相机算子预览界面就能看到，点击“保存内参”，将相机的内参保存到一个路径下。



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。

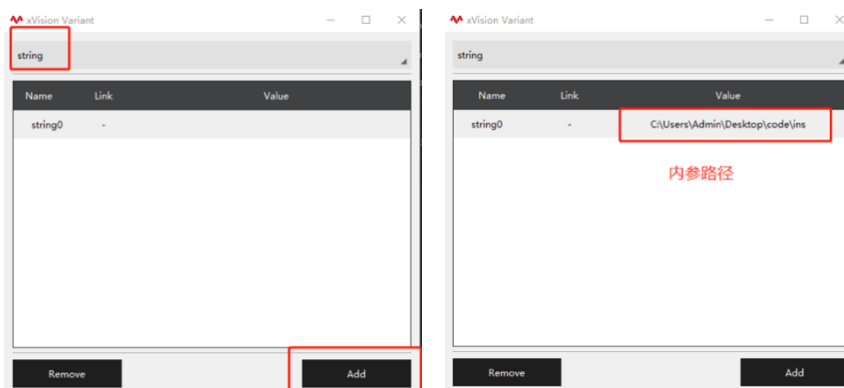


7. 如下图所示，点击工作区右上角的添加变量按钮。



8. 点击添加变量后，弹出添加变量对话框，如下图添加一个字符串，值为保存内参的路径，完成后，关闭添加变

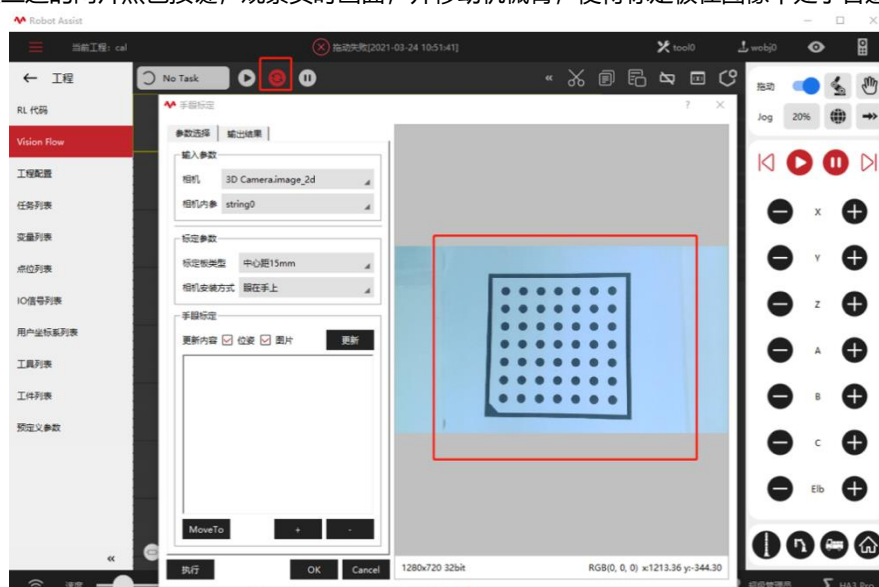
量对话框。



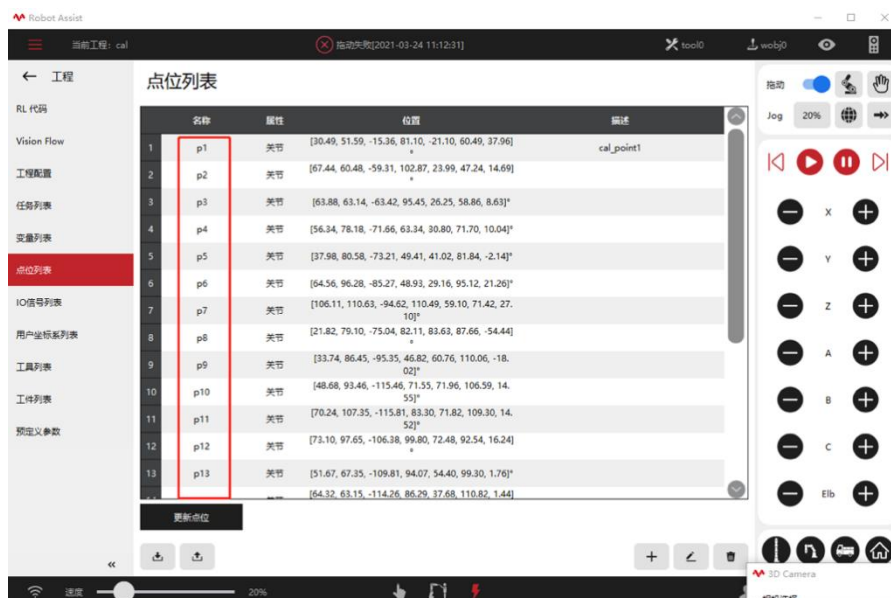
9. 双击“手眼标定”，弹出手眼标定配置窗口，按照下图选择相应的参数。



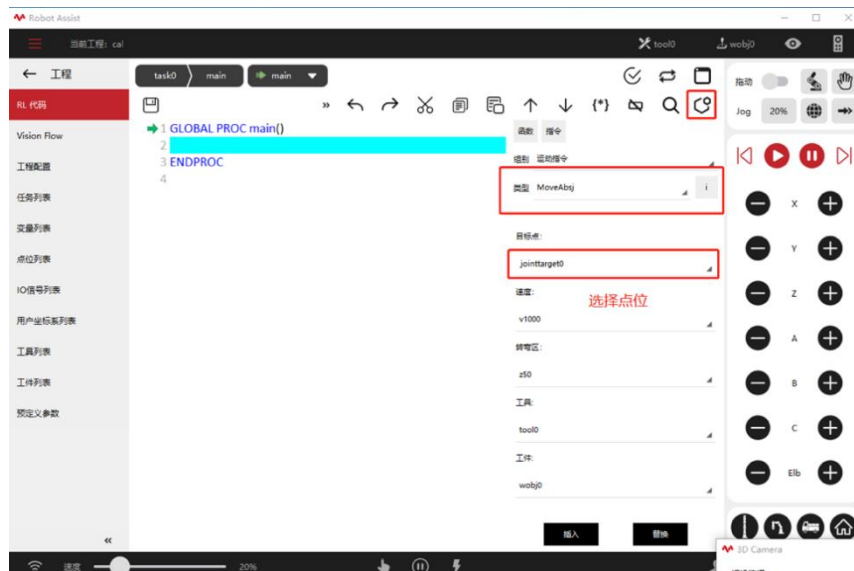
10. 点击视觉工作区左上角的视觉程序运行按钮，在手眼标定对话框中，可以看到实时的图像画面，用手同时按住机械臂法兰边的两片黑色按键，观察实时画面，并移动机械臂，使得标定板在图像中处于合适的位置。



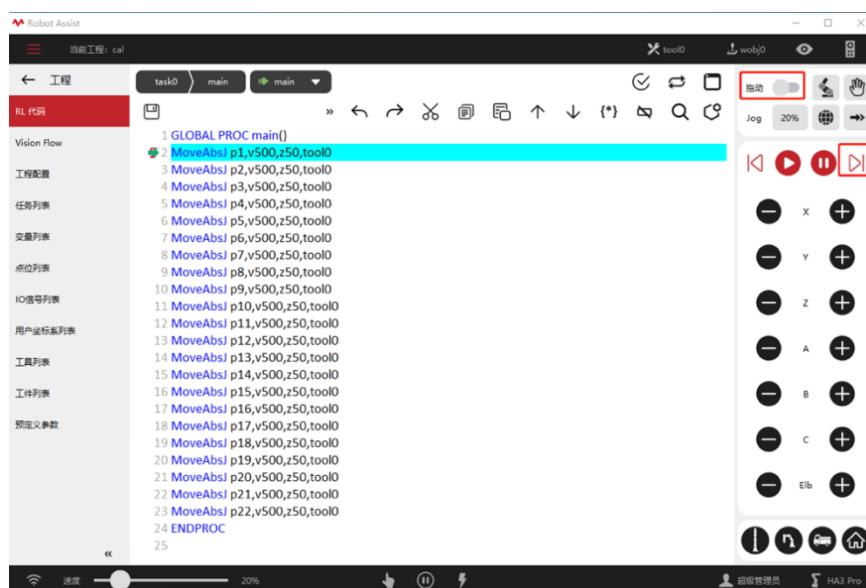
11. 点击“点位列表”，设置名称后，点击“更新位置”，在点击“下一步”，完成一个点位。重复第 10 步骤和第 11 步骤，获得更多的点位。如下图。



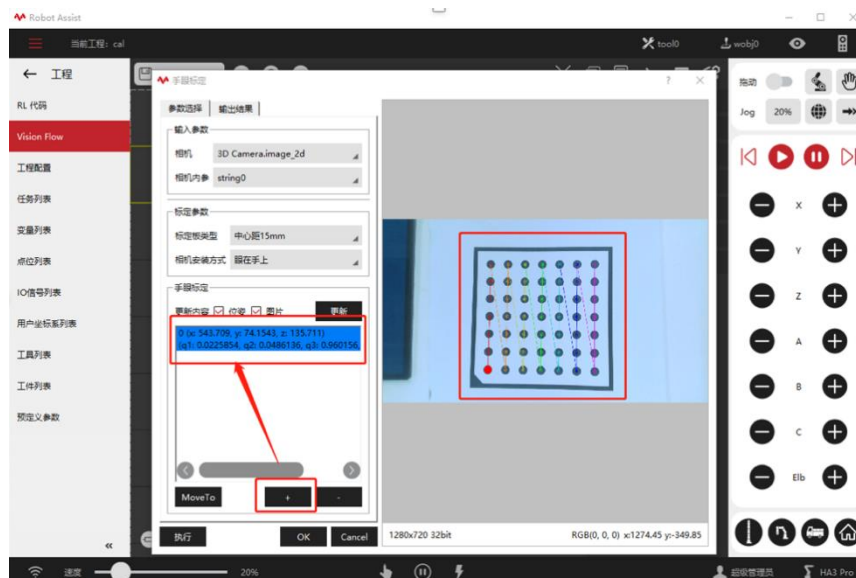
12. 点击“RL 代码”，插入点位。



13. 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。

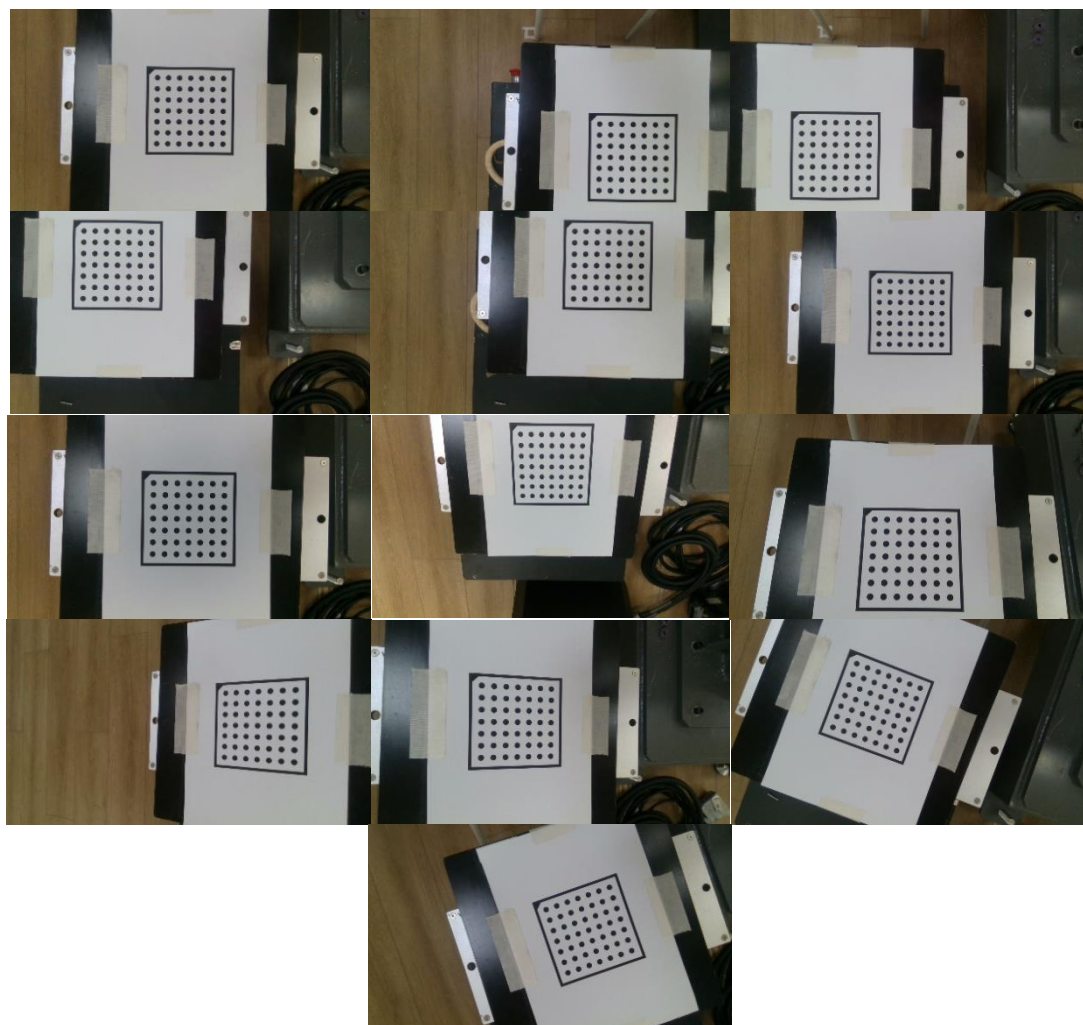


14. 单步执行完一条机器人移动指令后，如下图，点击添加一张标定图片。



重复第 13 步骤和 14 步骤，采集完成所有点位标定图片，选择“结果输出”选项卡，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。

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

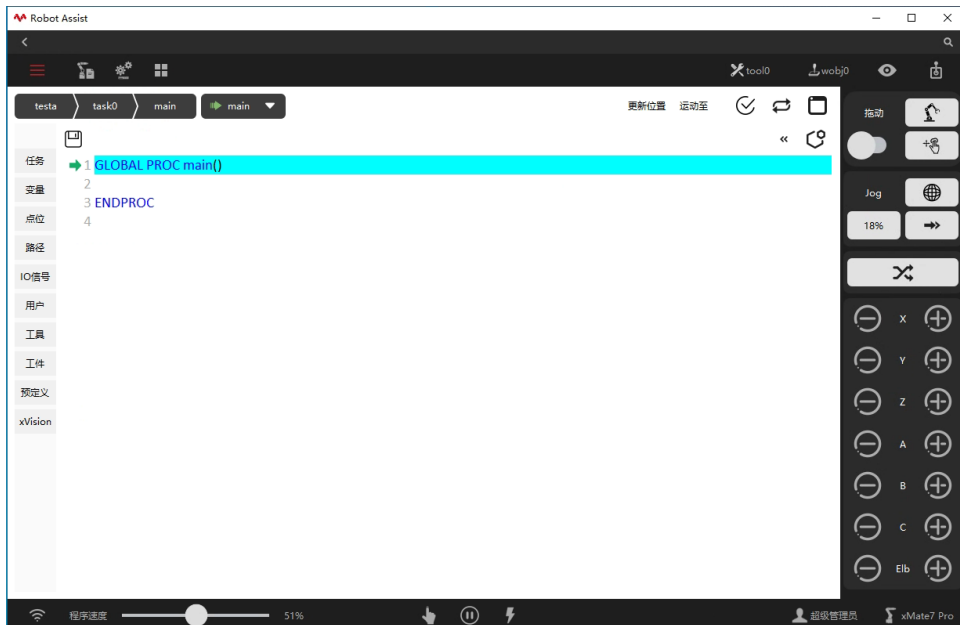
这个案例主要介绍如何进行视觉手眼标定，多用于 2D 视觉引导项目。

单点对位抓取三部曲：

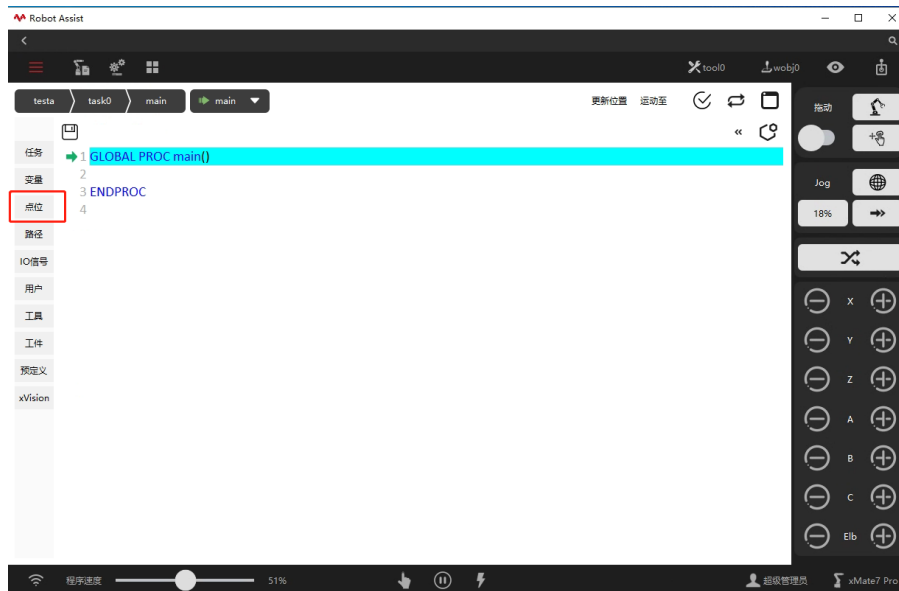
- 标定：相机运动：物料放置视野内，机构带动相机走 Z(9+3)字形拍照去点。
  - 示教：A. 引导机械手移动至正好能抓取到物料的位置，记录此时的物理点位坐标，即“示教物理点”  
B. 保持物料不动，引导机构移动回标定第五点拍照位（标定时 Z 字形轨迹中间的点位），记录此时的物理点位坐标，即为“示教拍照物理点”。拍照物料提取物料上稳定的图像特征点，即为“基准图像点”。
  - 做差：生产时，在“示教拍照物理点”拍照（第五点拍照位），提取物料上稳定的特征点，位置需要和示教时提取的基准点保持一致，计算偏差量（基准位-示教位）引导机构实现抓取。
1. 打开软件，点击“工程配置”，输入工程名称，点击“下一步”，完成工程创建。



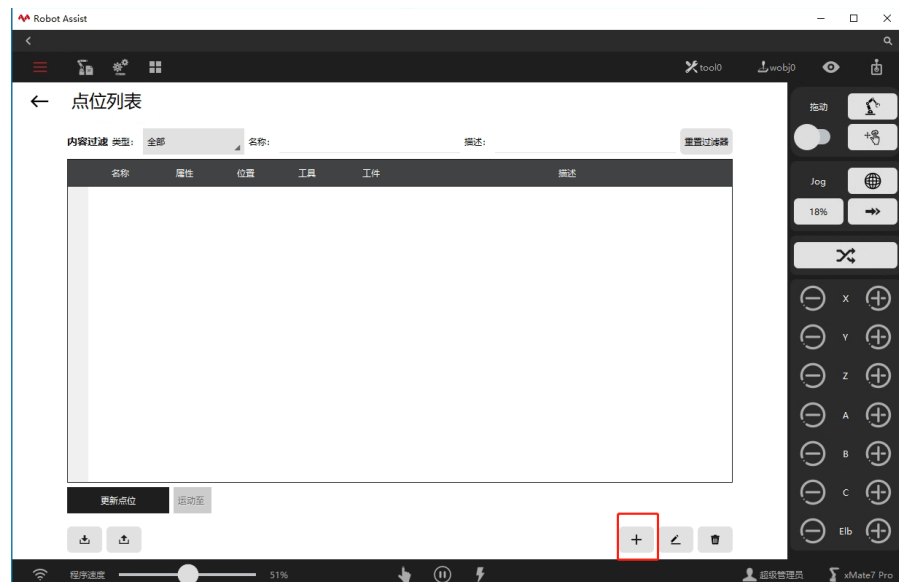
2. 创建完工程后，在 RL 代码区会自动生成两行机器人程序，如下图，点击“机器人操作面板”。



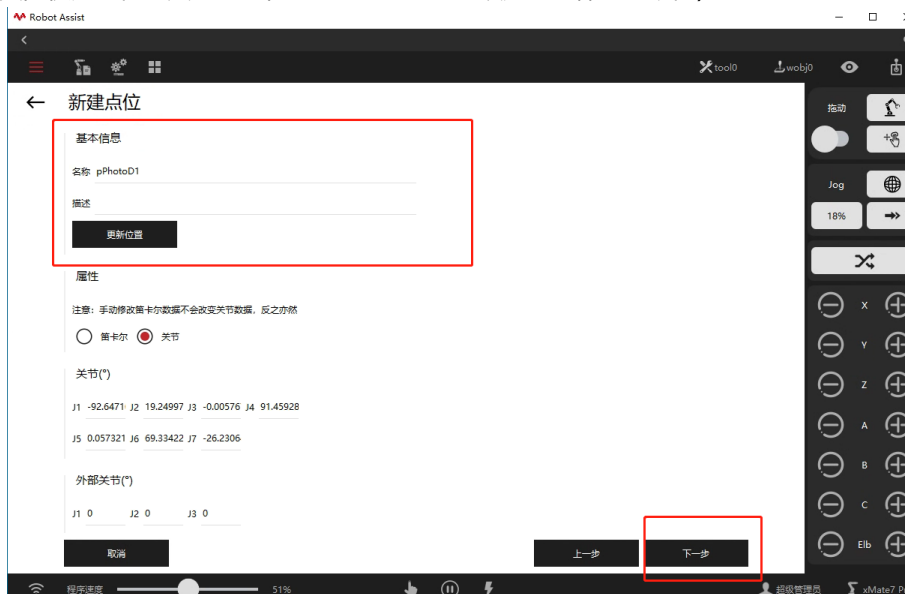
3. 新建机械手点位，设置“额定相机拍照位”，点击界面“点位”。



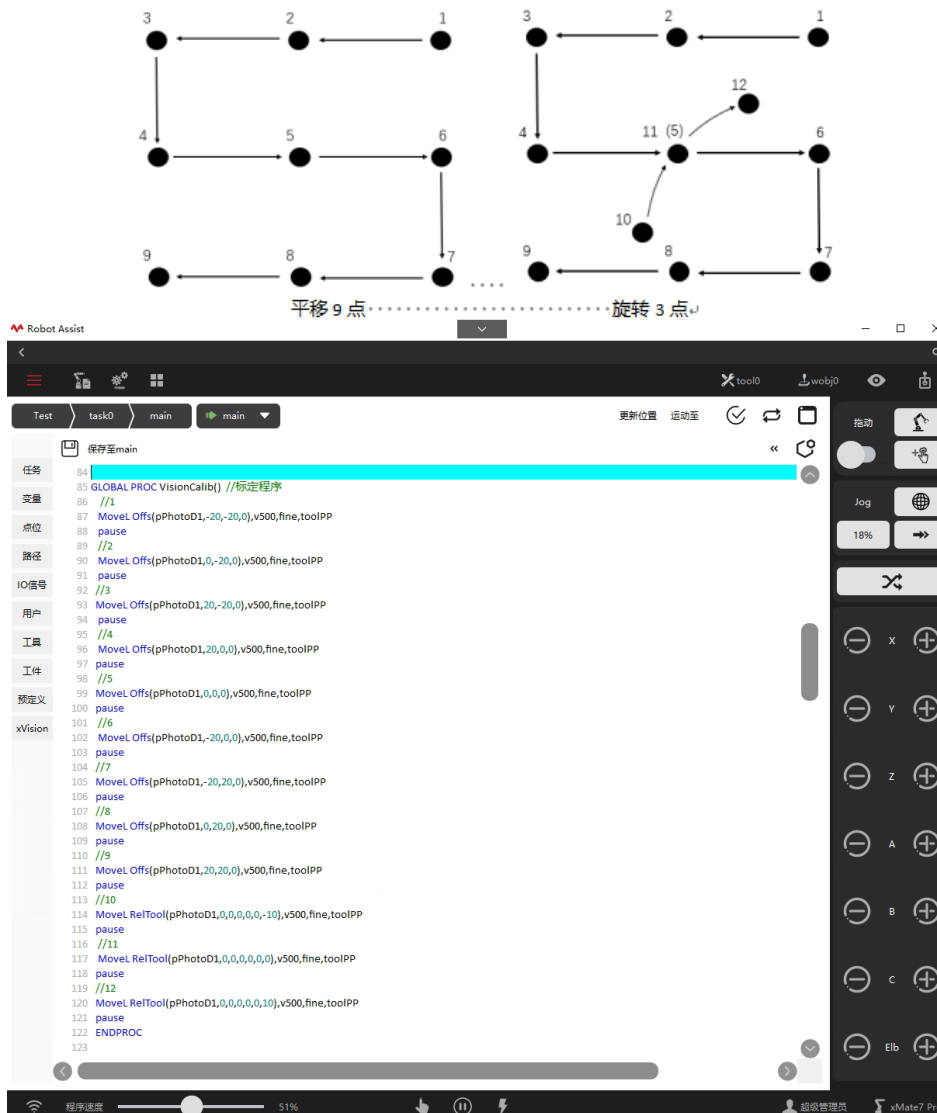
4. 进入“点位列表”界面，点击界面“+”。



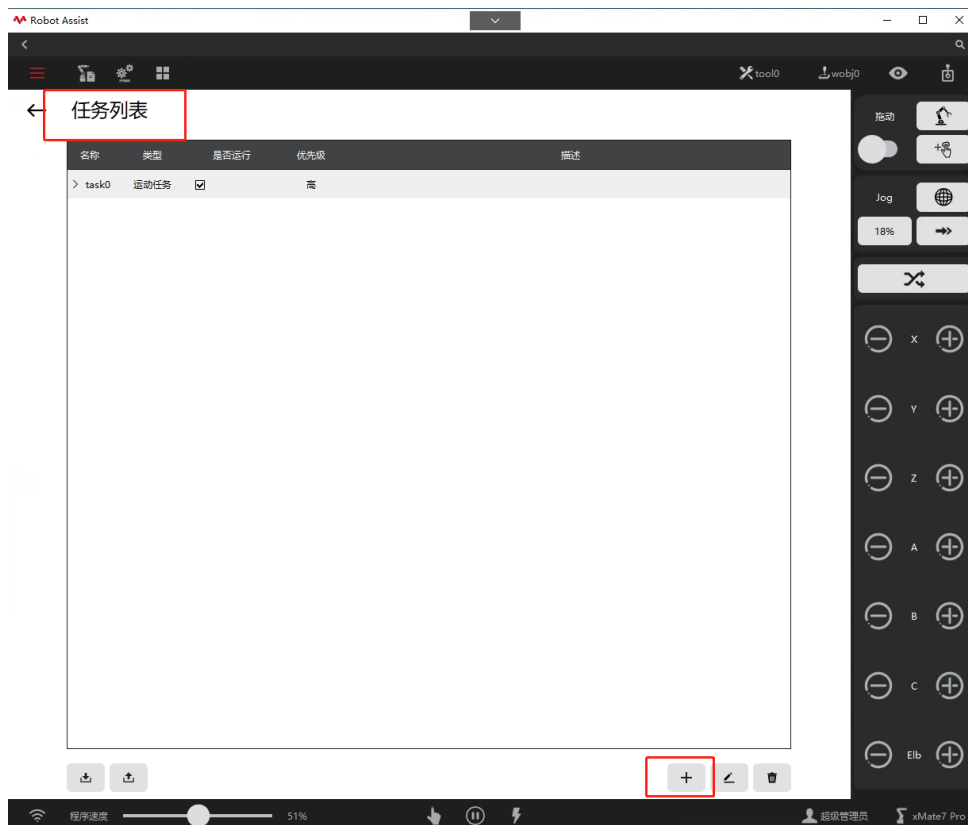
5. 新建点位，可自定义名称为 pPhotoD1,点击“更新位置”，点击下一步；  
(注意若后续实际使用时，工具坐标系和工件坐标系需与后续建立时保持一致；)



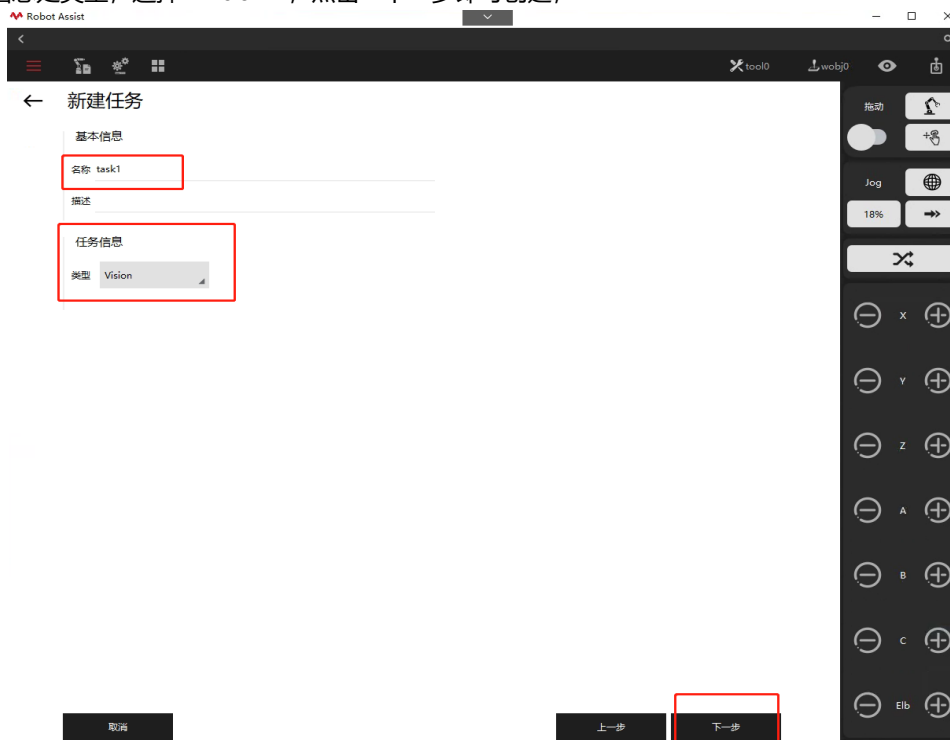
6. 编写机器人标定程序，机械人按照 12 点法进行走位 (先平移 9 点, 再旋转 3 点, 其中第 11 点和第 5 点重合), 使目标模板在视野中按照如下图的位置顺序出现。每到一个位置, 使用 “+” 添加新的一行。算子将自动采集此时机械臂位置和角度。“-” 按钮可删除选中行, 更新按钮可按照更新选中行。采集完成后可保存数据为 json。



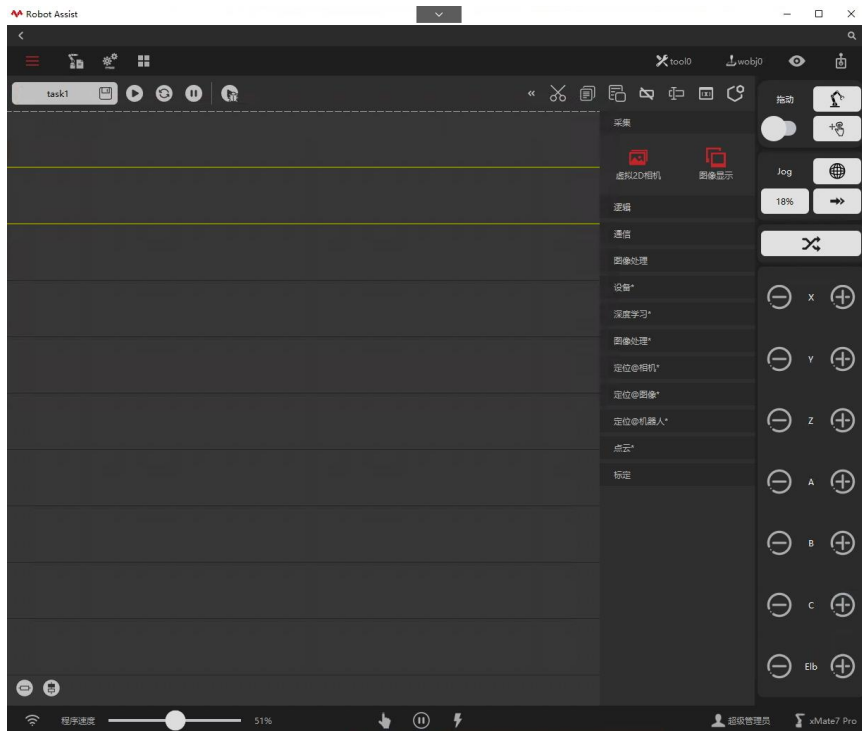
7. 新建视觉任务，点击主界面上的“任务”，进入到任务列表，点击“+”，新建视觉任务；



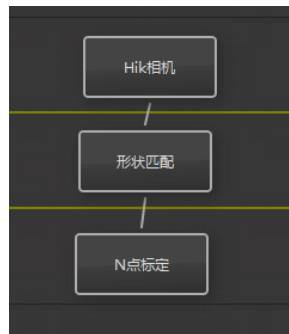
8. 任务信息处类型，选择“Vision”，点击“下一步”即可创建；



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



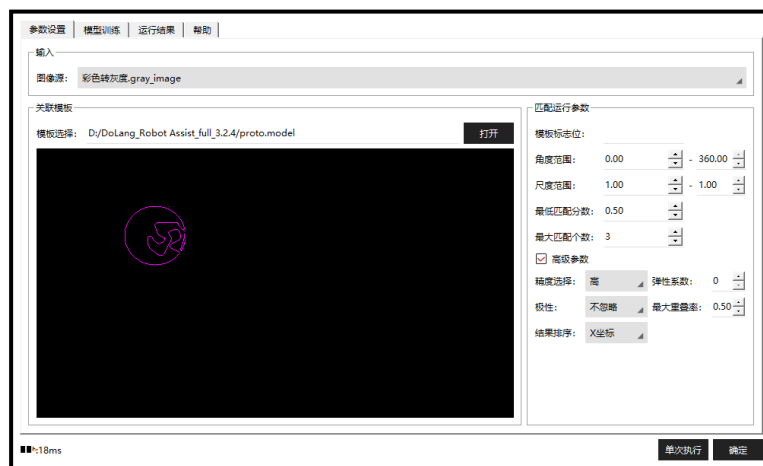
10. 搭建视觉标定流程，从“算子区”拖拽相应算子（Hik 相机，形状匹配，N 点标定）到流程编辑界面，并将其完成连线；



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. 形状匹配算子设置，



参数设置界面



模板训练界面

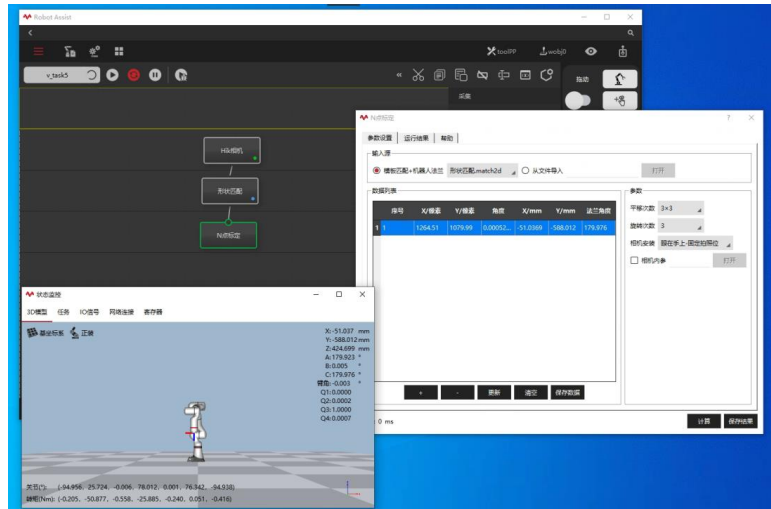


运行结果界面

- A. 输入源：选择其他算子输出图像；
- B. 模板选择：默认本地路径，训练时也会保存其中；  
当建立多个模板时，可选择模板另存为，并加载对应模板；

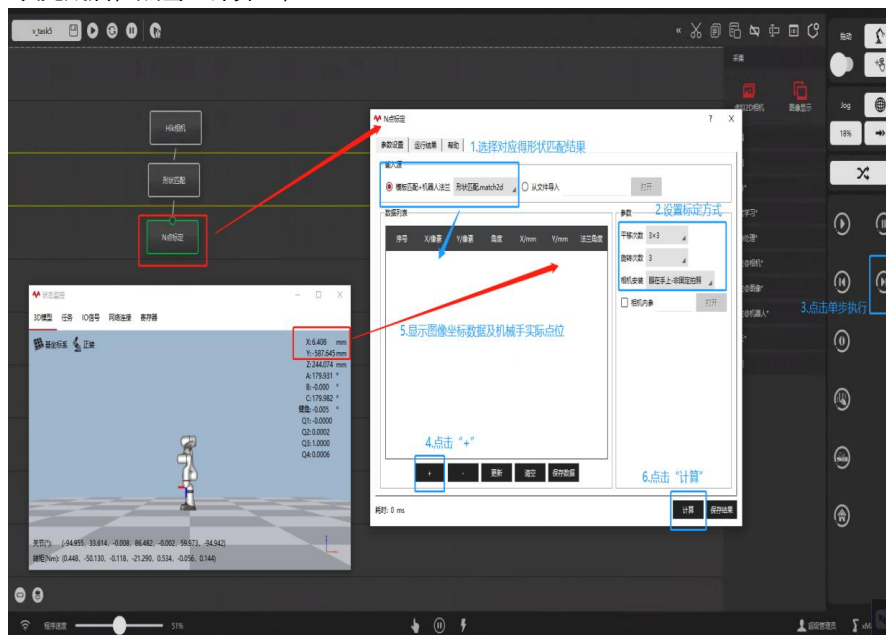
- C. 匹配运行参数：角度范围-90 到 90
- D. 最低匹配分数：目标区域最小得分，小于此分数则被淘汰；

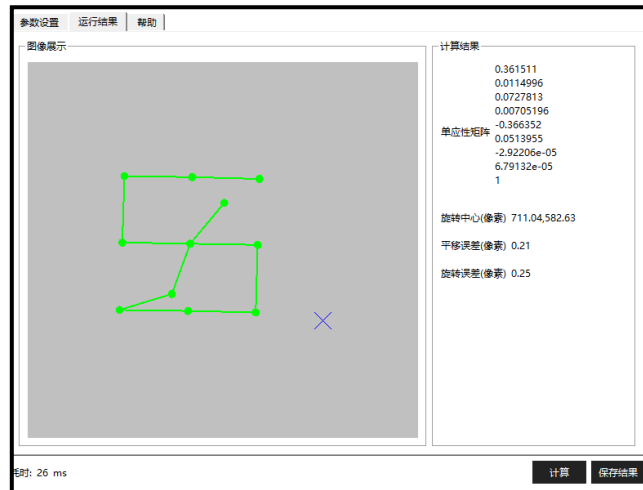
13. N 点标定设置：



此场景为：相机在机械手上；

- A. 设定好拍照基准位；
- B. 当前图像通过”形状匹配“建立基准模板，
- C. 机械手带动相机根据要求移动 12 次；
- D. 每次到位后，点击 N 点标定算子中的”+“；
- E. ”数据列表“将会记录”形状匹配“找到模板的图像坐标 XY 及机械手实际坐标；
- F. 移动 12 次完成后，点击”计算“，





计算结果界面

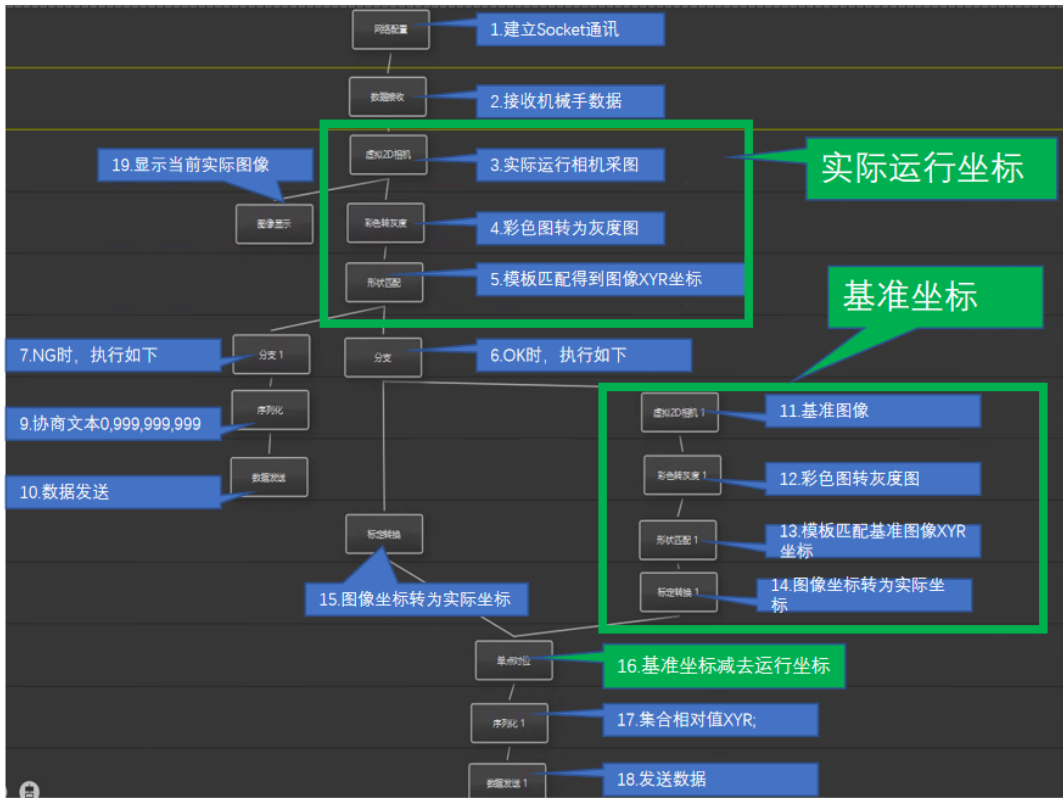
14. 搭建“机械手运行程序”，实现场景为单相机物料的抓取场景，基于图像基准点与图像运行点坐标实现抓取，输出相对偏差值；若视觉异常时，则对应输出 0,999,999,999；

```

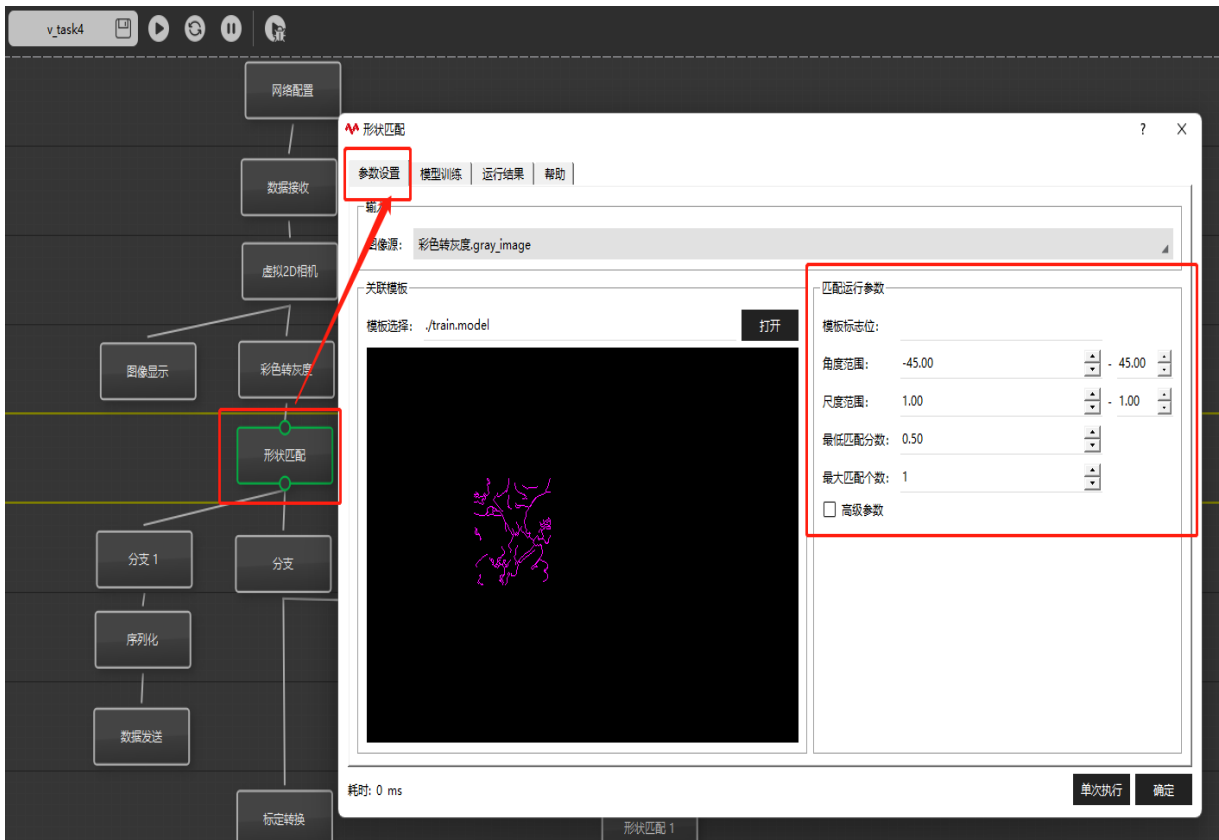
26 GLOBAL PROC VisionGetPosXV() //转点位
27 SocketClose("sc") //关闭通讯 (防止残留数据)
28 wait 0.1
29 SocketCreate("192.168.0.34",9797,"sc") //建立通讯
30 MoveJ pPhotoD1,v1000,fine,toolPP //移动到拍照位pPhotoD1
31 wait 0.5 //避免抖动
32 SocketSendString("D1","sc") //发送消息"D1"
33 wait 1
34 visionHK=SocketReadDouble(4,60,"sc") //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R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 //移动到更新值后的抓取位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

```

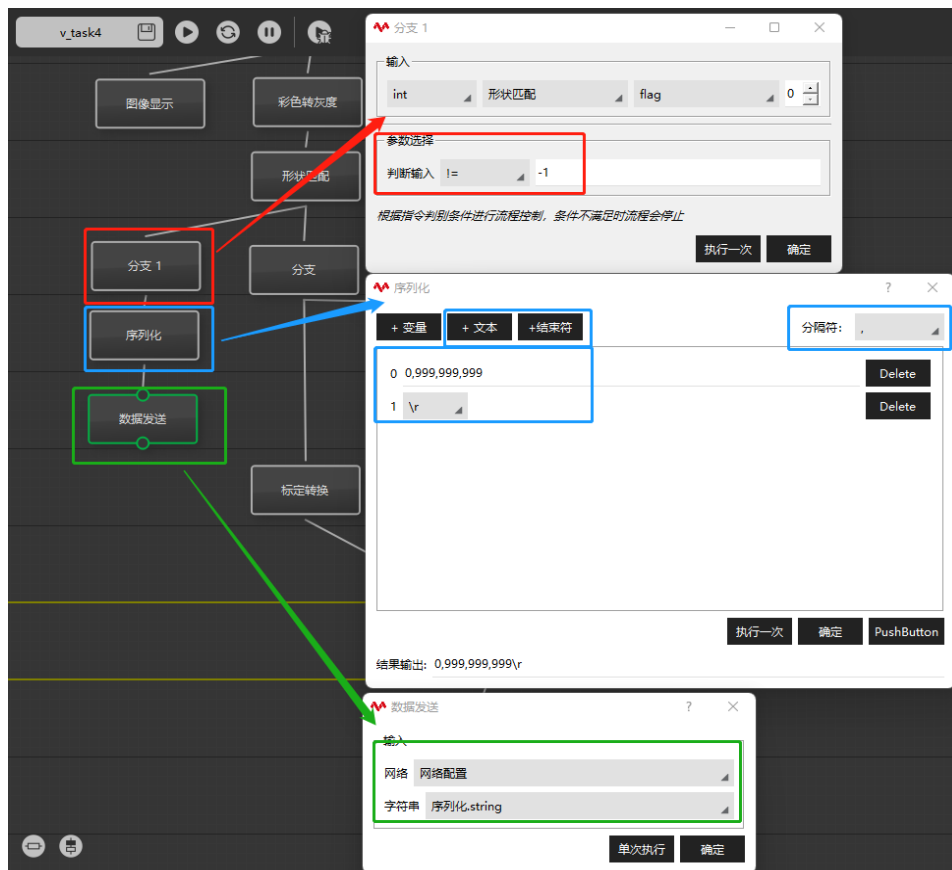
15. 搭建“视觉引导流程”，实现场景为单相机物料的抓取场景，基于图像基准点与图像运行点坐标实现抓取，输出相对偏差值；若视觉异常时，则对应输出 0,999,999,999；



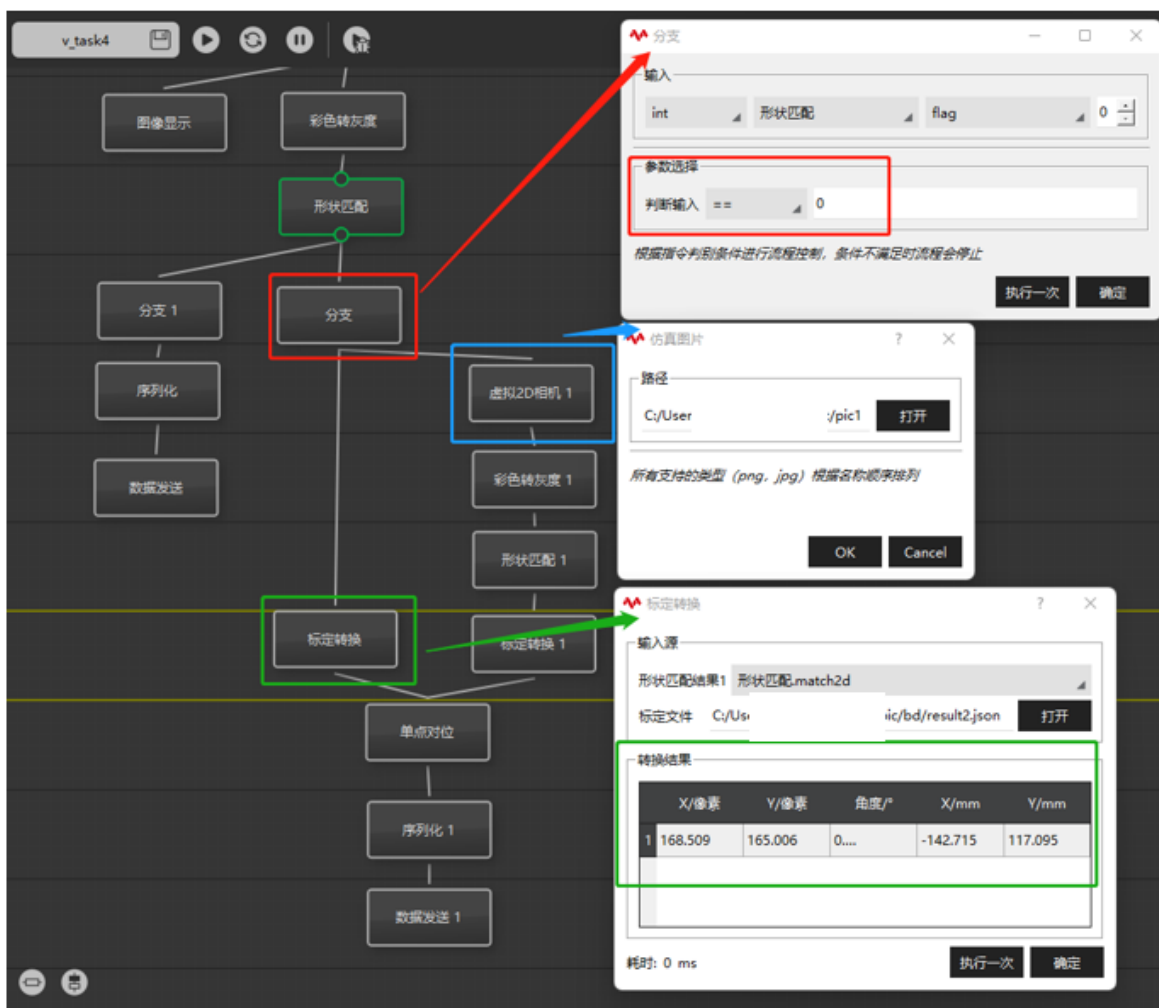
16. xVision 识别模板，使用“形状匹配”算子，在目标检测图中寻找与模板相同图案的区域，以定位产品在图像中 X,Y 和相对模板的旋转角度；



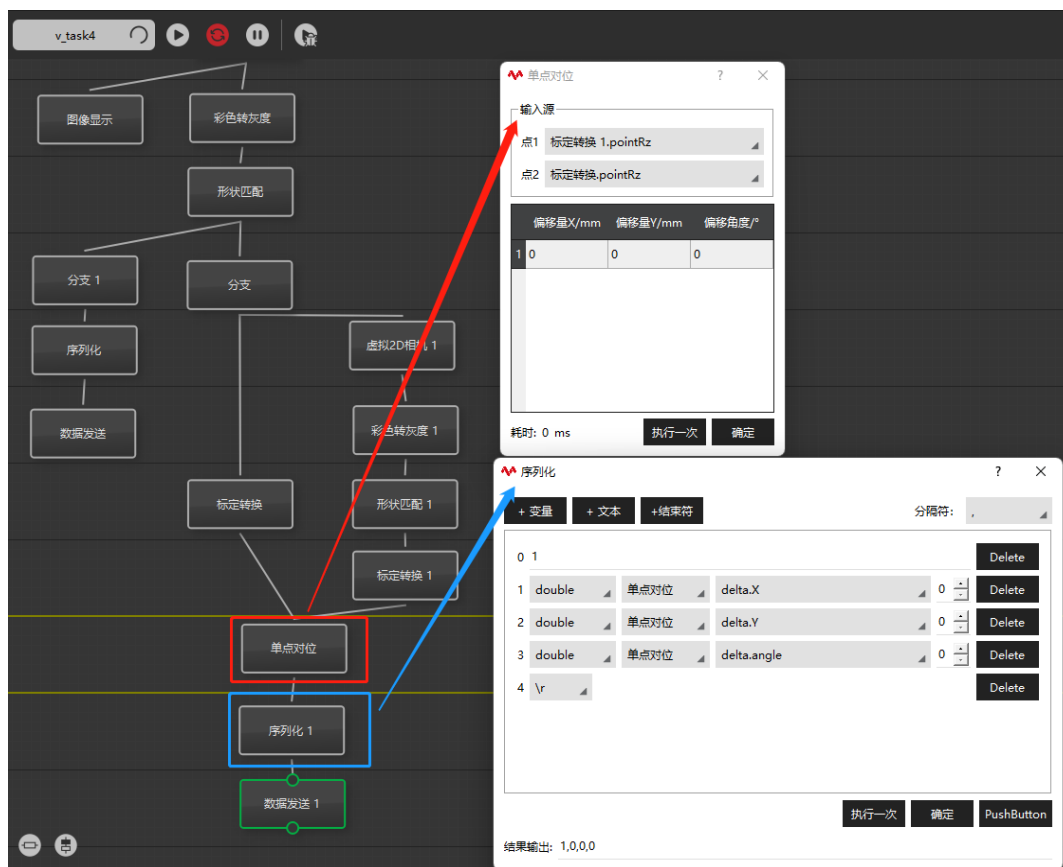
17. xVision-NG 情况处理，使用“分支”算子对“形状匹配”算子的识别结果 OK/NG 进行区分，使用“序列化”算子可以实现添加自定义协商文本“0,999,999,999”，并设置分割符及添加结束符；通过“数据发送”算子将“序列化”算子设置内容通过 Socket 通讯发送给机械手



18. 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“虚拟 2D 相机”算子加载本地基准图片, 使用“标定转换”算子将像素坐标转换为相对的世界坐标;



19. xVision-做差, 使用“单点对位”算子, 基准点减去运行点, 得到实际位置与基准位置的相对偏差值, 使用“序列化”算子, 将“单点对位”算子输出记过进行集合, 四组数据集合为一串字符串, 分割符设置为“,”; 结束符设置为“\r”, 分别代表着状态位 1, 相对偏差值 X, 相对偏差值 Y, 相对偏差值 R, 通过“数据发送 1”算子将“序列化”算子内容发送给机械手;



### 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;

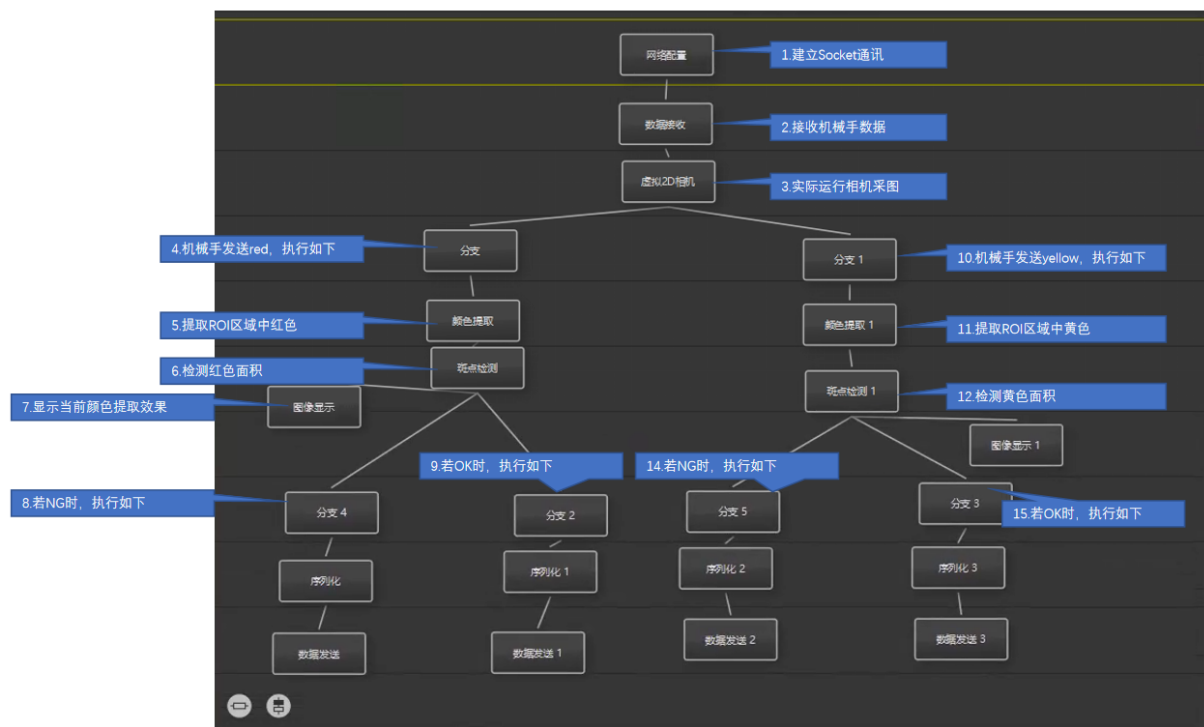
1. 搭建机械手运行程序；

```

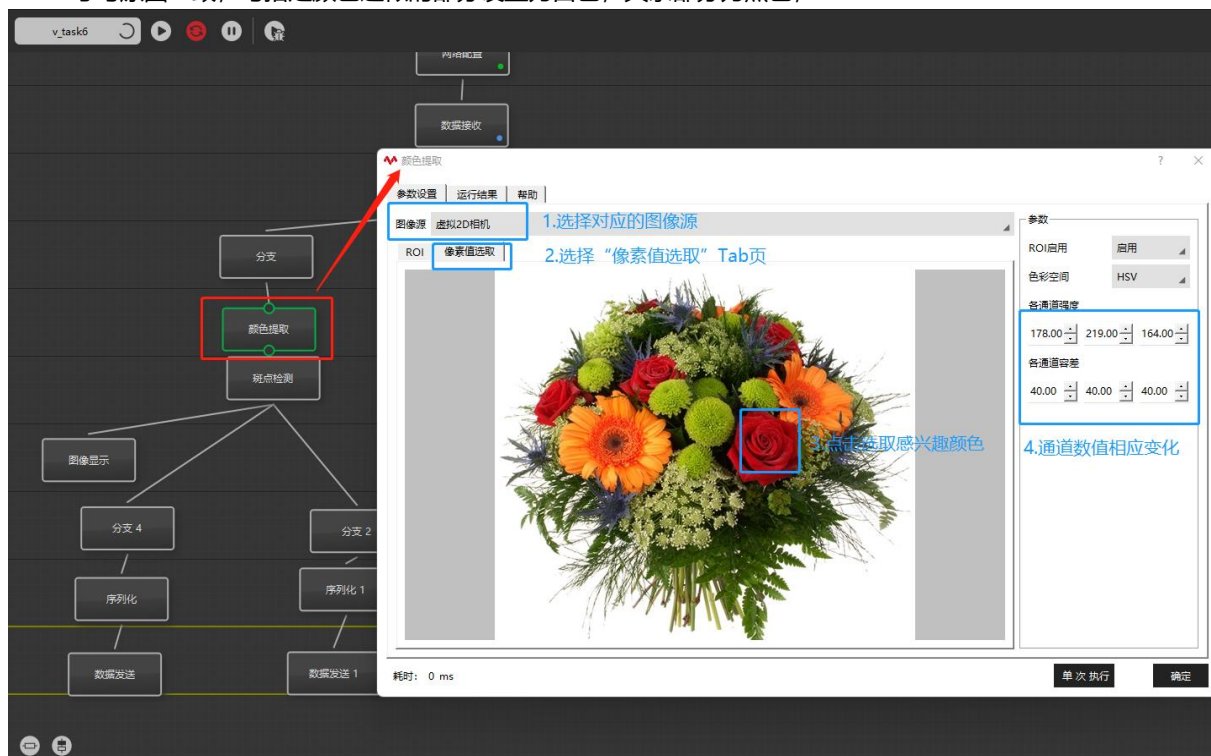
305 GLOBAL PROC VisionColorXV() //颜色检测
306   SocketClose("sc")           //关闭通讯 (防止残留数据)
307   wait 0.1
308   SocketCreate("192.168.0.34",9797,"sc") //建立通讯
309   MoveJ pPhotoD1,v1000,fine,toolPP //移动到拍照位pPhotoD1, 检测红色物料
310   wait 0.5 //避免抖动
311   SocketSendString("red","sc") //发送消息"red"
312   wait 1
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;

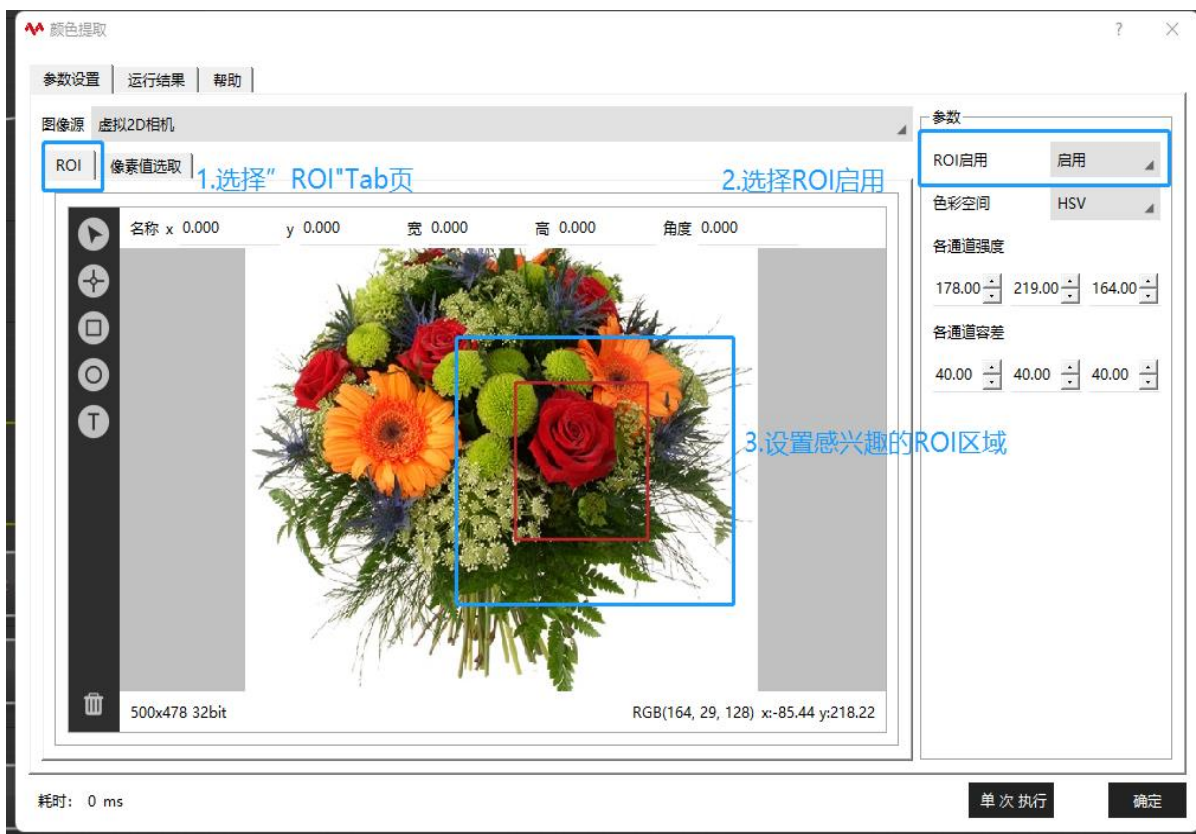


3. 设置“颜色提取”算子，从一张彩色图像上提取与指定颜色近似部分，同时生成一张掩膜(Mask)图，掩膜尺寸与原图一致，与指定颜色近似部分设置为白色，其余部分为黑色；



参数设置：

- A. 输入源：选择需提取颜色的算子输出图像
- B. 选择“像素值选取”Tab页；
- C. 在图像中，点击感兴趣颜色，作为提取颜色；
- D. 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:

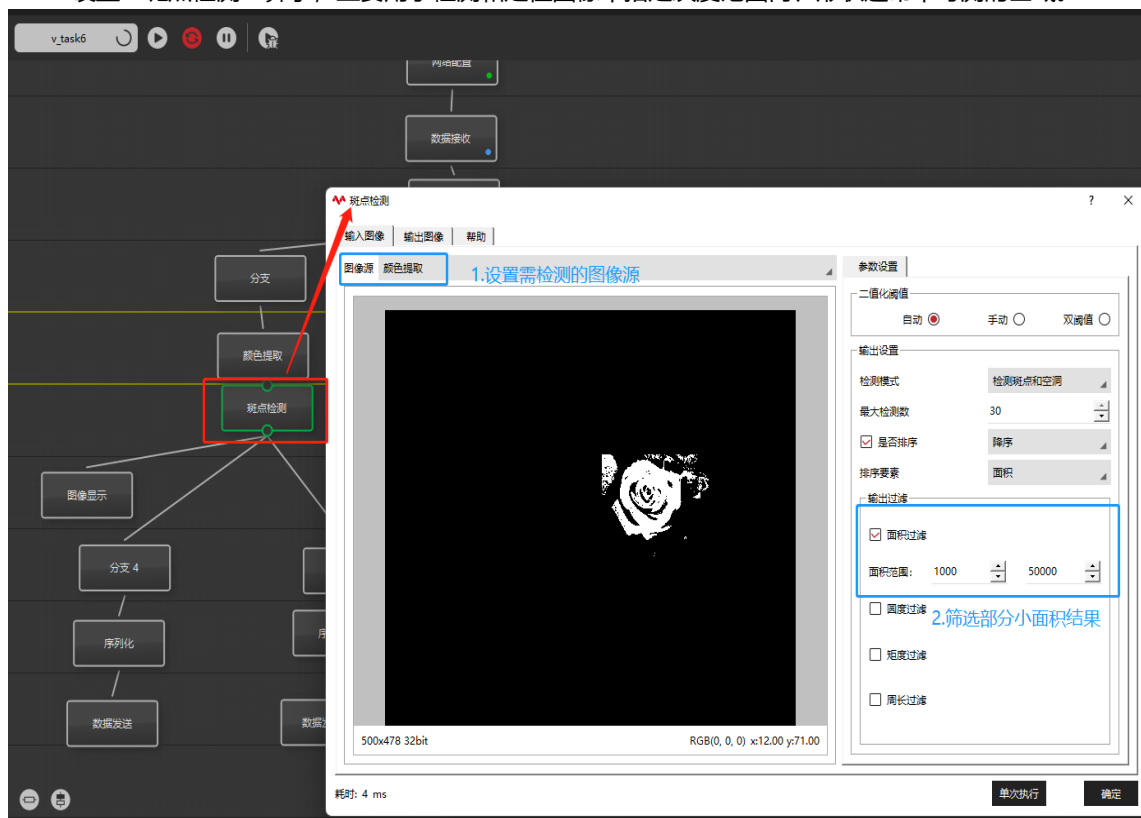
- 选择"ROI" Tab 页
- 选择 ROI 启用, ---图像中会出现 ROI 区域
- 设置 ROI 区域包含需检测区域;



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;

- C. 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;

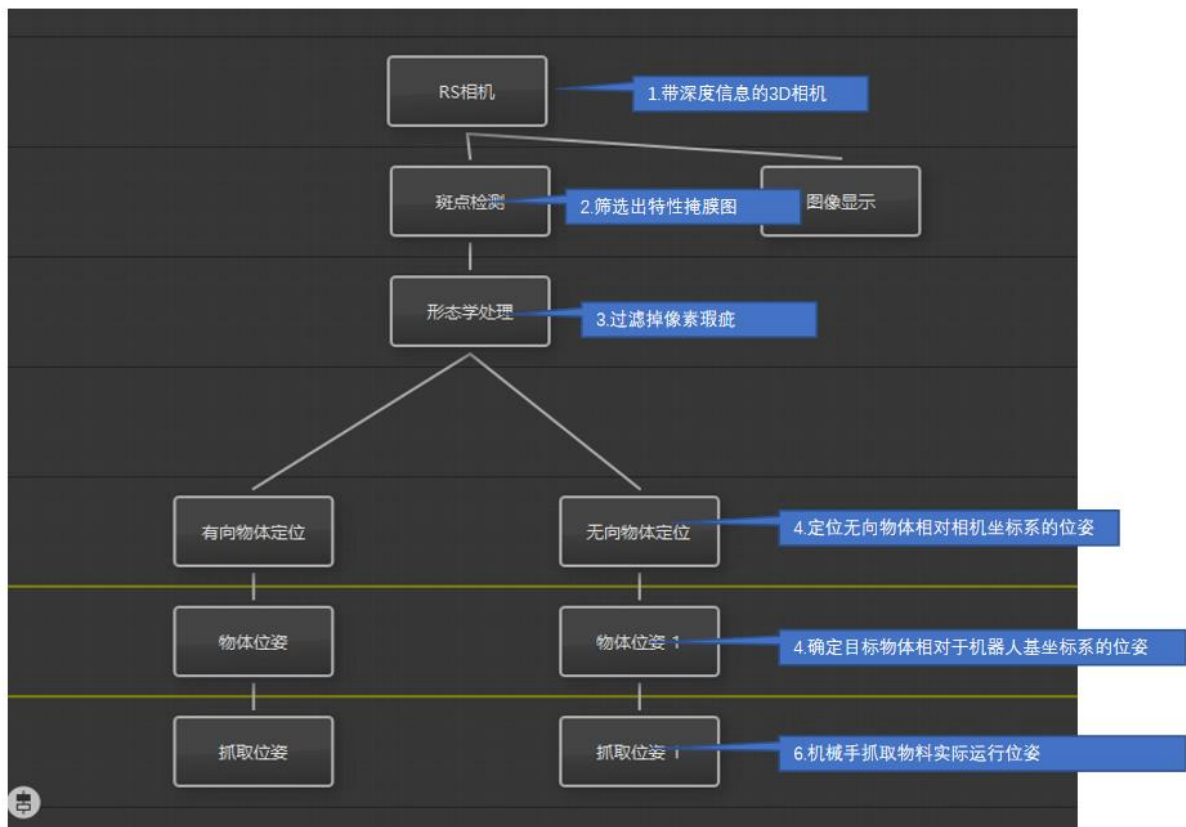
工位 1: 有向物体定位+抓取位姿

用于计算有向物体 (例如螺钉) 的 3D 位姿

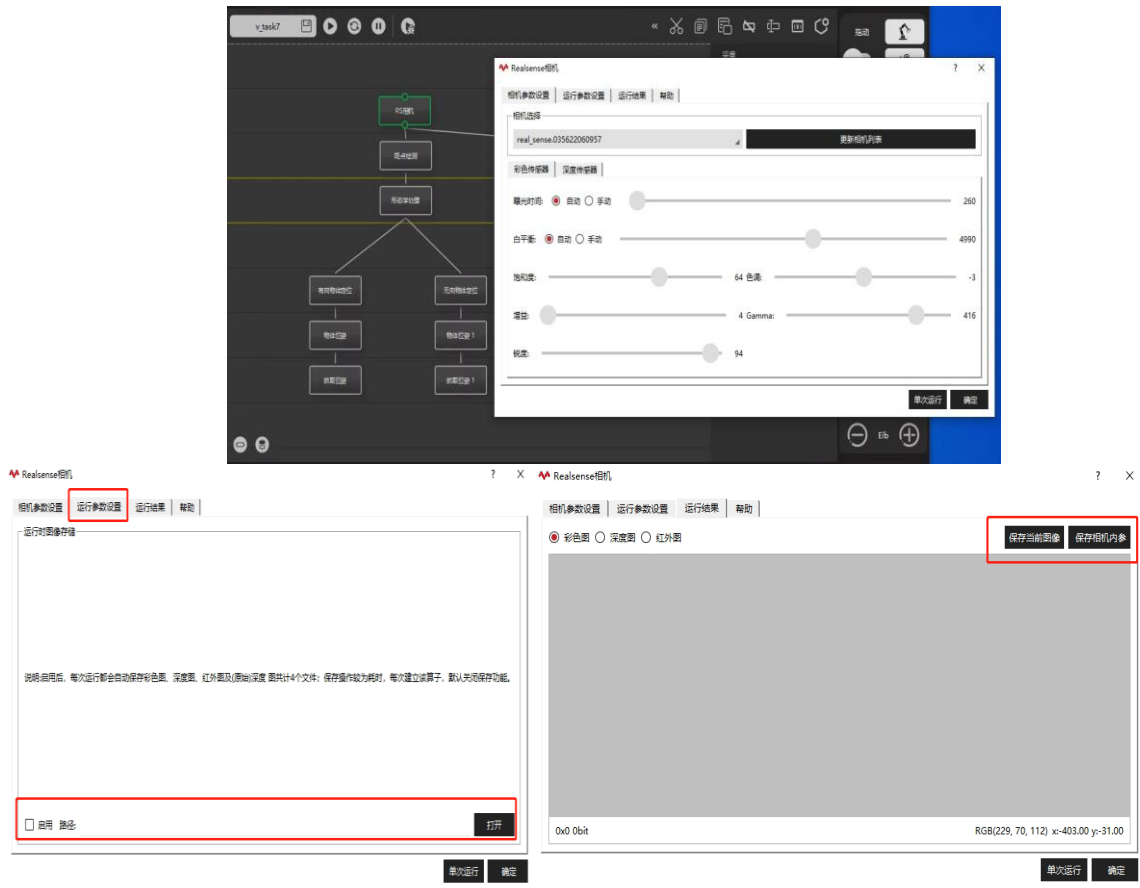
工位 2: 无向物体定位+抓取位姿

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:



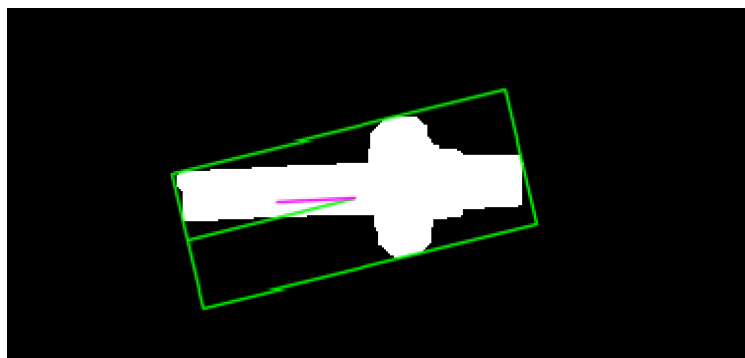
### 2. 设置 RS 相机, 带深度信息的 RGB 相机



参数设置:

- A. 相机连接时可自动进行识别，也可以手动选择相机；
  - B. 相机参数设置，运行参数设置，运行结果，帮助
  - C. 相机参数设置： 曝光时间：调节图像亮度；
  - D. 运行参数设置： 设置保存图像路径，
  - E. 运行结果：展示彩色图，深度图，红外图；
  - F. 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 mm）。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系的位姿;



参数设置:

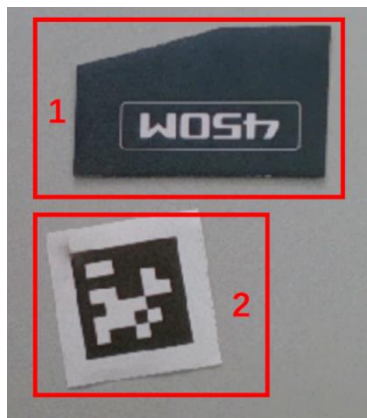
- A. 物体位姿: 通常来自位姿转换或有向物体的位姿;
  - B. 相机外参: 即手眼标定文件;
  - C. 相机安装方式: 眼在手上或眼在手外;
  - D. 记录拍照时的相机位姿, 到达拍照位后点击记录即可。
5. 设置“抓取位姿”算子, 位姿可以通过“物体位姿”算子, 转换为机器人 base 坐标系下的位姿, 若系统未进行工具坐标系的标定, 或工具坐标系与工件坐标系标定不相符时, 需要通过抓取标定进一步处理, 变为法兰相对机器人基座坐标系的位姿, 从而可以发送给机器人, 作为机器人程序可用的位姿。



## 5.5 斑点检测、模板匹配、Marker 定位

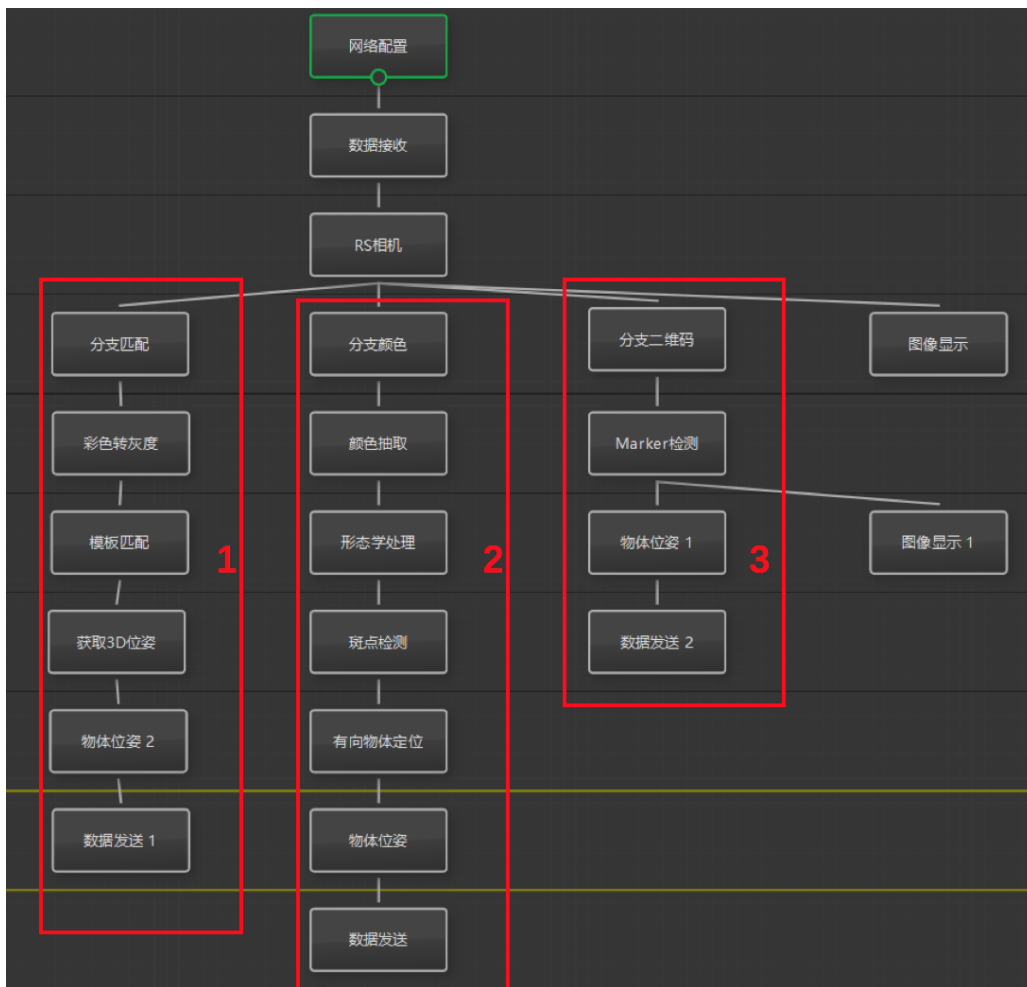
### 功能简介

- 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；



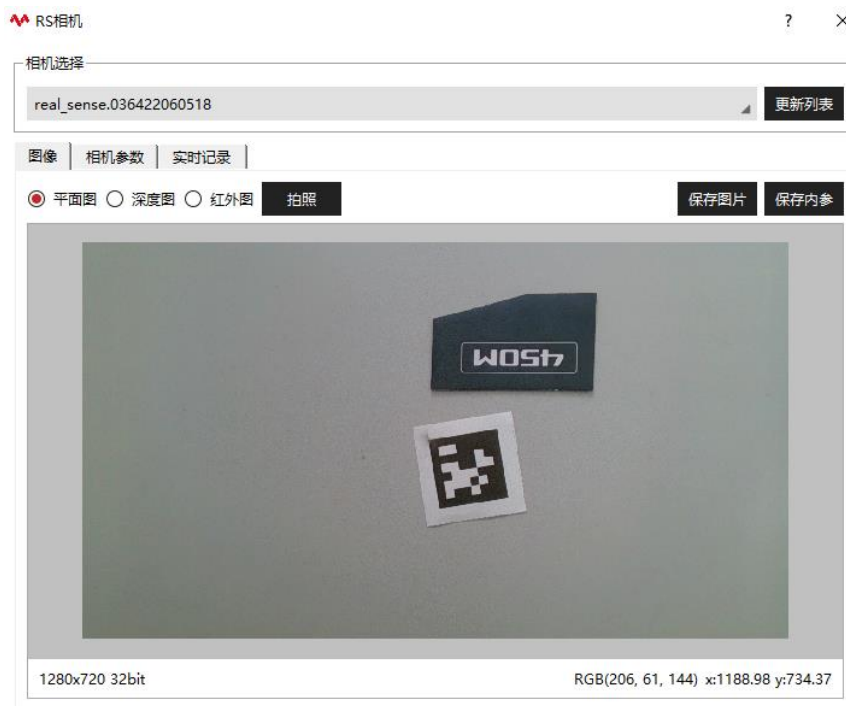
### 任务方案整体介绍

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。



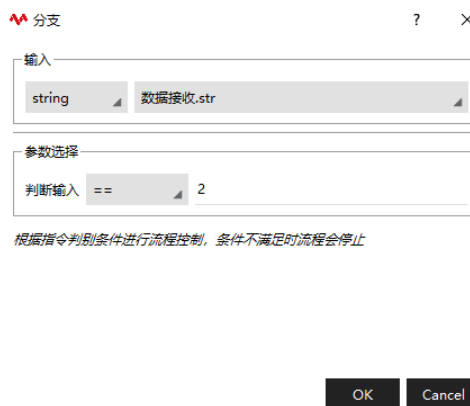


RS 相机

**分支 1**

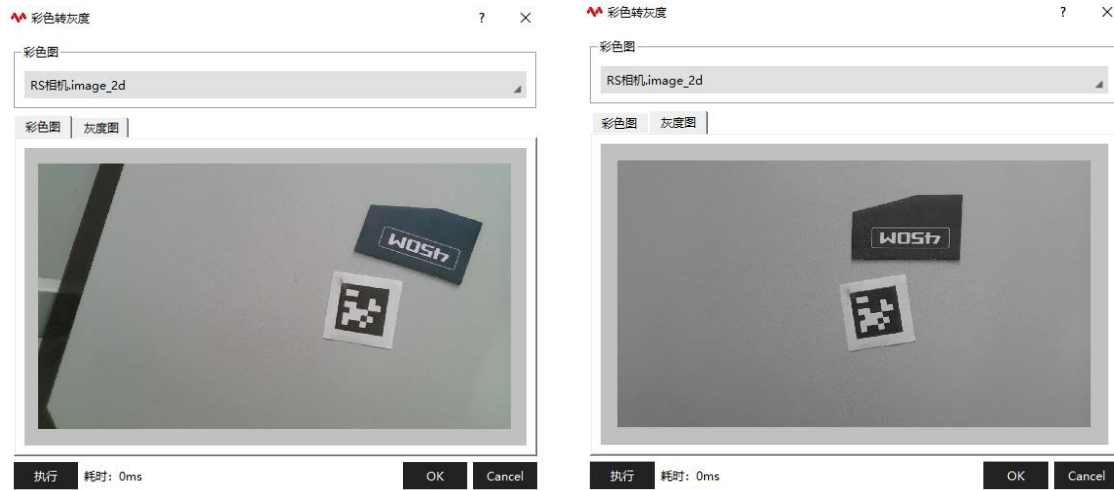
分支 1 的思路为通过“彩色转灰度”，将彩色图像转化为“模板匹配”所需的灰度图像；利用“模板匹配”确定目标的平面位姿，进而通过“获取 3D 位姿”算子计算目标相对相机坐标系的立体位姿；再通过“物体位姿”算子计算目标相对机器人基坐标系的位姿；最后由“数据发送”算子把位姿数据发送出去。

当网络接收的数据为 2 时，进入该分支。



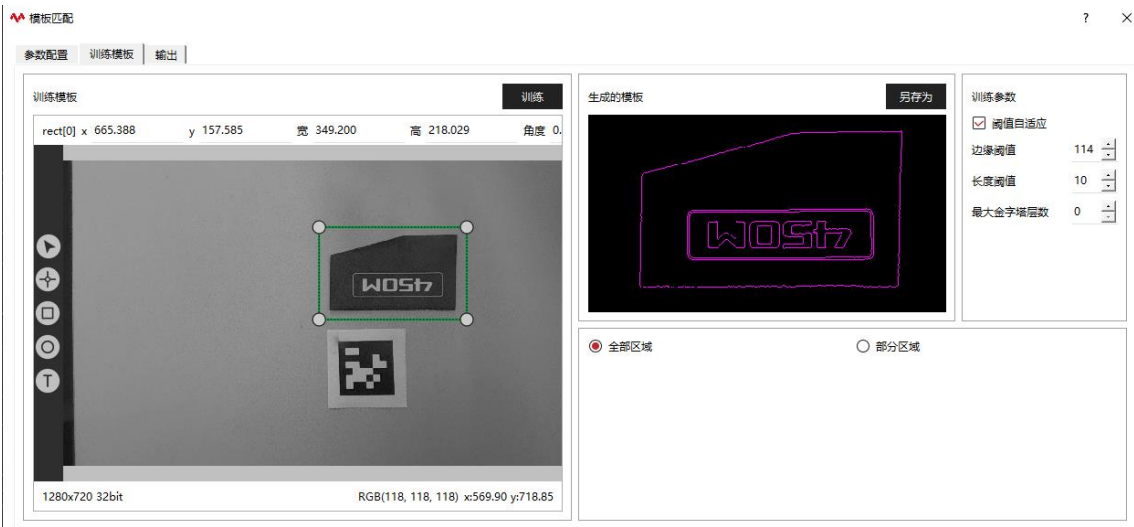
分支

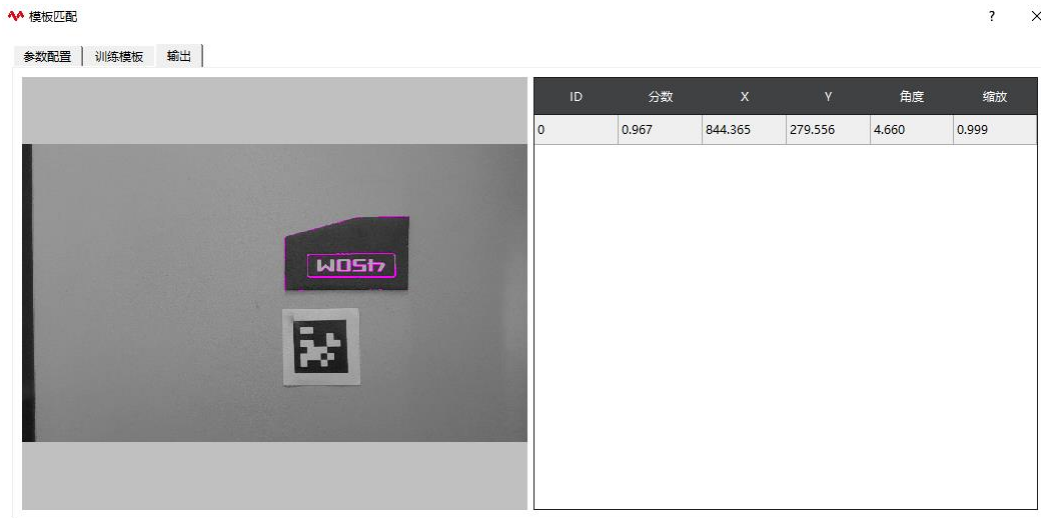
彩色转灰度算子完成颜色转换。



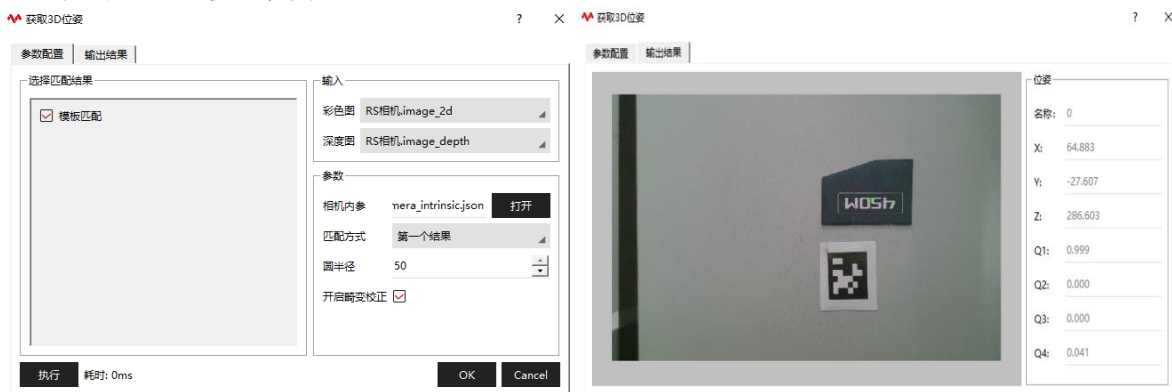
彩色转灰度

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。



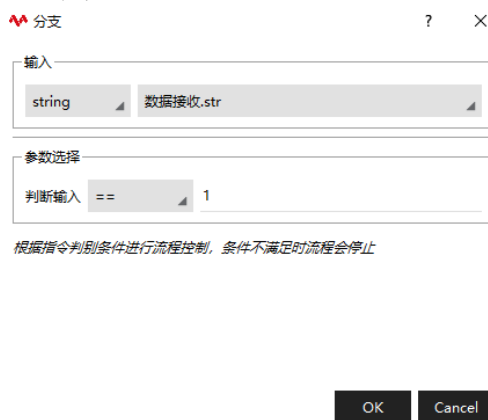
位姿数据可以通过网络发送出去。



## 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。



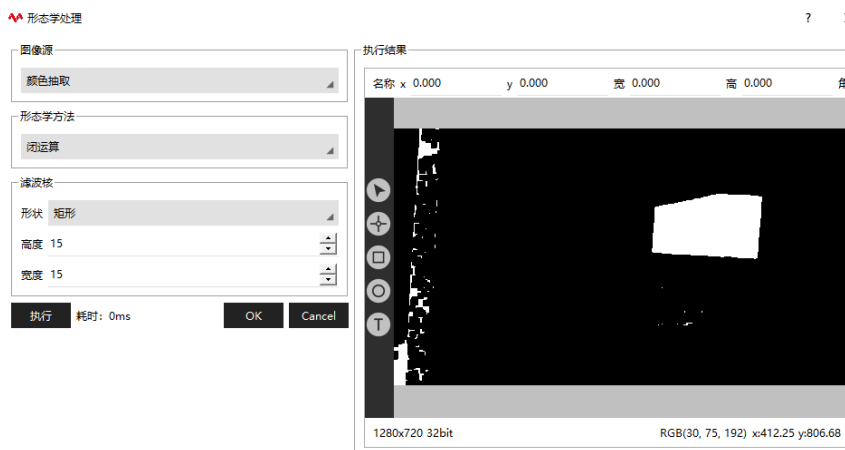
## 分支

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。

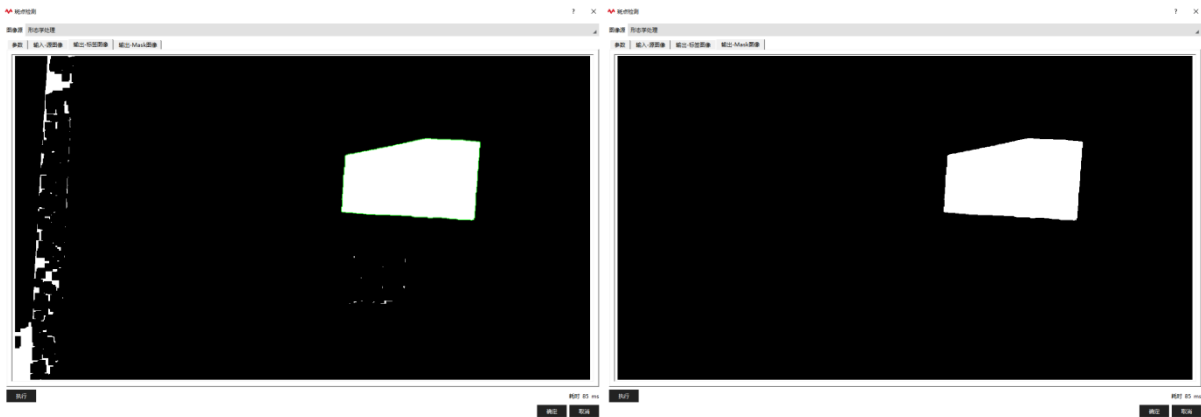


形态学处理

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。

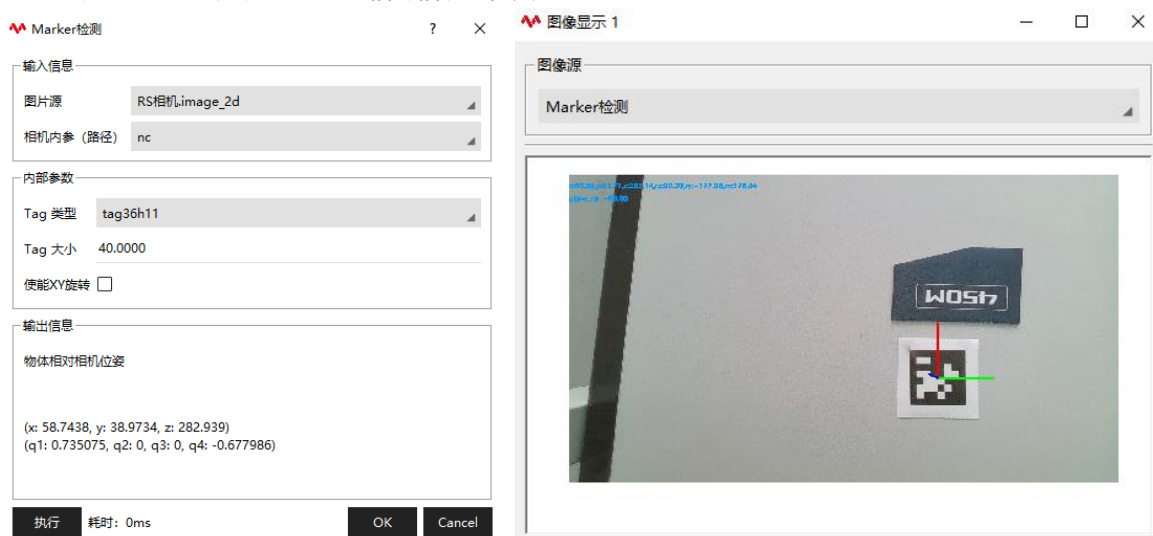
当网络接收的数据为 1 时，进入该分支。



OK Cancel

### 分支

利用 Marker 检测确定 Marker 相对相机的位姿。



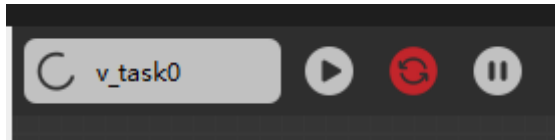
Marker 检测

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。



## 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背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。