

---

MVP 算法平台用户手册

---

# MVP 算法平台

## 用户手册

## 免责声明

- 在法律允许的最大范围内，本手册所描述的产品（含其硬件、软件、固件等）均“按照现状”提供，可能存在瑕疵、错误或故障，本公司不提供任何形式的明示或默示保证，包括但不限于适销性、质量满意度、适合特定目的、不侵犯第三方权利等保证；亦不对使用本手册或使用本公司产品导致的任何特殊、附带、偶然或间接的损害进行赔偿，包括但不限于商业利润损失、数据或文档丢失产生的损失。
- 使用本产品时，请您严格遵循适用的法律。若本产品被用于侵犯第三方权利或其他不当用途，本公司概不承担任何责任。



- 如本手册内容与适用的法律相冲突，则以法律规定为准。

## 前言

本节内容的目的是确保用户通过本手册能够正确使用产品，以避免操作中的危险或财产损失。在使用此产品之前，请认真阅读产品手册并妥善保存以备日后参考。

## 资料获取

访问本公司官网 ([www.xxxxx.com](http://www.xxxxx.com)) 获取说明书、应用工具和开发资料。

## 概述

本手册适用于 MVP3.0.0 版本算法平台。

本手册可能包含技术上不准确的地方、或与产品功能及操作不相符的地方、或印刷错误。我司将根据产品功能的增强或变化而更新本手册的内容，并将定期改进及更新本手册中描述的软硬件产品。更新的内容将会在本手册的新版本中加入，恕不另行通知。

## 符号约定

对于文档中出现的符号，说明如下所示。

符号	说明
 <b>说明</b>	说明类文字，表示对正文的补充和解释。
 <b>注意</b>	注意类文字，表示提醒用户一些重要的操作或者防范潜在的伤害和财产损失危险。
 <b>警告</b>	警告类文字，表示有潜在风险，如果不加避免，有可能造成伤害事故、设备损坏或业务中断。
 <b>危险</b>	危险类文字，表示有高度潜在风险，如果不加避免，有可能造成人员伤亡的重大危险。

## 目 录

<b>第 1 章 更新说明</b> .....	<b>1</b>
<b>第 2 章 产品简介</b> .....	<b>2</b>
2.1 功能概述.....	2
2.2 功能特性.....	2
2.3 运行环境.....	2
2.4 软件安装.....	3
<b>第 3 章 软件界面说明</b> .....	<b>6</b>
3.1 主界面.....	6
3.1.1 菜单栏.....	7
3.1.2 快捷工具条.....	11
3.2 工具.....	13
3.2.1 通用参数与功能介绍.....	14
3.3 切换用户功能.....	18
<b>第 4 章 视觉工具</b> .....	<b>20</b>
4.1 常用算子.....	20
4.1.1 概述.....	20
4.2 图像采集.....	20
4.2.1 概述.....	20
4.2.2 仿真器.....	20
4.2.3 相机.....	22
4.3 定位.....	24
4.3.1 斑点检测.....	24
4.3.2 找边.....	31
4.3.3 找圆.....	35
4.3.4 模板匹配.....	39

4.3.5 模板比对.....	47
4.4 图像处理.....	51
4.4.1 平滑滤波.....	51
4.4.2 图像二值化.....	55
4.4.3 裁剪.....	57
4.4.4 彩色转灰度.....	57
4.4.5 颜色提取.....	58
4.4.6 仿射变换.....	61
4.4.7 形态学处理.....	64
4.4.8 边缘梯度.....	66
4.4.9 亮度.....	68
4.4.11 图像相减.....	72
4.5 标定.....	74
4.5.1 棋盘格标定.....	74
4.5.2 N 点标定.....	80
4.5.3 读取标定文件.....	82
4.6 测量.....	83
4.6.1 两点生成直线.....	83
4.6.2 线段与圆交点.....	85
4.6.3 线段与直线交点.....	87
4.6.4 线段与线段交点.....	89
4.6.5 直线与圆交点.....	91
4.6.6 直线与直线交点.....	92
4.6.7 圆与圆交点.....	94
4.6.8 点与点夹角.....	96
4.6.10 点与圆距离.....	98
4.6.11 点与直线距离.....	100

4.6.12 点与线段距离.....	101
4.6.13 点与点距离.....	103
4.6.14 线段与圆距离.....	104
4.6.15 线段与直线距离.....	105
4.6.16 线段与线段距离.....	107
4.6.17 直线与圆距离.....	110
4.6.18 圆与圆距离.....	111
4.6.19 拟合线.....	113
4.6.20 拟合圆.....	114
4.6.21 卡尺工具.....	116
4.7 识别.....	119
4.7.1 条形码.....	119
4.7.2 二维码.....	121
4.7.3 字符识别.....	125
4.7.4 深度学习字符识别.....	129
4.8 逻辑控制.....	129
4.8.1 容器.....	130
4.8.2 数学表达式.....	133
4.8.3 顺序模块.....	134
4.8.4 循环模块.....	136
4.8.5 循环退出节点.....	137
4.8.6 分支节点.....	137
4.8.7 多任务同步模块.....	138
4.9 通讯.....	139
4.9.1 串口配置.....	139
4.9.2 网络配置.....	140
4.9.3 报文发送.....	142

4.9.4 报文接收.....	143
4.9.5 报文发送（参数可配）.....	145
4.9.6 报文接收（参数可配）.....	148
4.10 辅助工具.....	152
4.10.1 保存图像.....	152
4.10.2 循环次数.....	153
<b>第 5 章 案例展示.....</b>	<b>156</b>
5.1 PIN 针针头尺寸检测.....	156
<b>第 6 章 获得支持.....</b>	<b>错误！未定义书签。</b>

## 第 1 章 更新说明

更新版本	更新记录
详见菜单栏->帮助->关于	更新 MVP 界面显示风格。
	优化图像处理类算子性能。
	新增多画面功能。详见 3.1
	支持多线程异步功能。
	新增常用算子列表功能。详见 4.1
	模板匹配和 OCR 算子支持可直接导入图片训练模板。
	更新模板匹配和 OCR 算子配置界面。
	更新找边和找圆新算法和配置界面。
	测量算子可直接输出实际距离。详见 4.6
	模板匹配支持多形状。详见 4.3.4
	修复二维码识别算子异常崩溃问题。
	新增参数有效性检查。
	更新每个算子的默认输出参数。
	新增 OCR 深度学习算子。

## 第 2 章 产品简介

### 2.1 功能概述

MVP 算法平台集成 9 类机器视觉系统基础功能算子，分别为：1、图像采集；2、定位；3、图像处理；4、标定；5、测量；6、识别；7、辅助工具；8、逻辑控制；9、通讯。

- **丰富的算子集**可以完成图像采集，目标定位，图像预处理，检测系统标定，目标长度、夹角测量，条码、二维码和字符识别以及处理图像自动保存等图像处理相关工作。
- **逻辑控制模块**可以灵活的实现顺序、循环以及条件控制等流程编辑，无需编程，简单易学。
- **通讯模块**配合工业控制需求可以实现 TCP 和串口通讯，提供与其他控制器件的通讯接口。
- **平台强大的算法工具库**能够满足视觉定位、识别、测量、检测等视觉应用需求。

### 2.2 功能特性

- 组件形式集成算子，支持拖拉和双击添加算子操作，无需编程。
- 利用算子连线自由搭建视觉解决方案，提供视觉化流程编辑方式，减小用户操作流程，提高用户可视化体验和开发效率。
- 以用户体验为中心的界面设计，提供算子处理的图像与结果双显示区，参数配置区，自由切换有效节省屏幕空间。
- 采用多权限管理方式，保护视觉方案安全性，贴合不同用户群需求。
- 支持多平台运行，适应 Windows7/Windows10（32/64bit），兼容性好。

### 2.3 运行环境

	最低配置	推荐配置
操作系统	Win7/Win10 32/64bit	

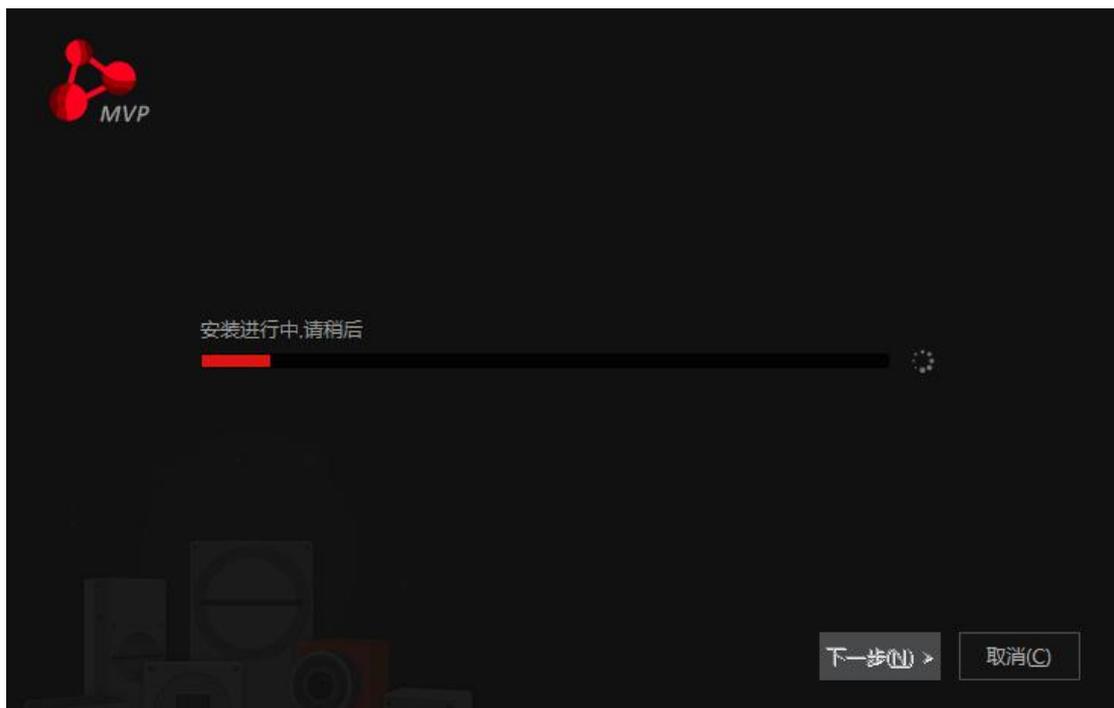
<b>CPU</b>	IntelAtom(TM)Processor E3940@1.6G	Intel Pentium IV3.0GHz 或以上
<b>内存</b>	2GB	4G 或更高
<b>显卡</b>	显存 1G 以上显卡	
<b>网卡</b>	推荐使用 Intel Pro1000 系列以上网卡	
<b>USB 接口</b>	如果需要接 U3 相机，电脑需要支持 USB3.0 接口，否则无要求。	

## 2.4 软件安装

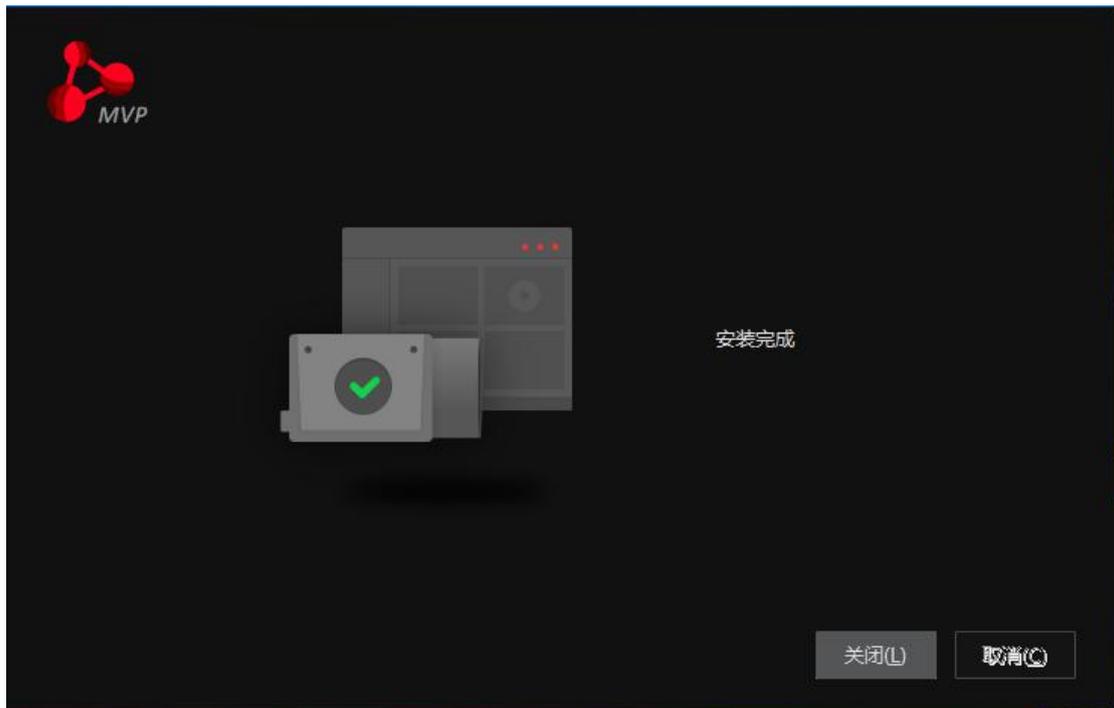
MVP 提供 32 位和 64 位安装包,用户可视系统情况选择对应的安装包进行安装,在安装 MVP 软件的过程中,会提示用户选择安装对应相机所需的驱动,以便支持相机高性能拉流功能。在安装 MVP 之前若客户已安装更早版本,本安装包会自动卸载老版本并安装新版本。

MVP 客户端安装步骤如下:

- 1.双击 MVP 安装包进行安装,如下图所示。软件安装前,需要设置软件安装的路径,选择需要安装的相机驱动。若需要更改设置,可直接进行修改。确认后,点击“安装”开始软件安装过程。



2.安装完成后，点击“关闭”完成安装过程，如下图所示。

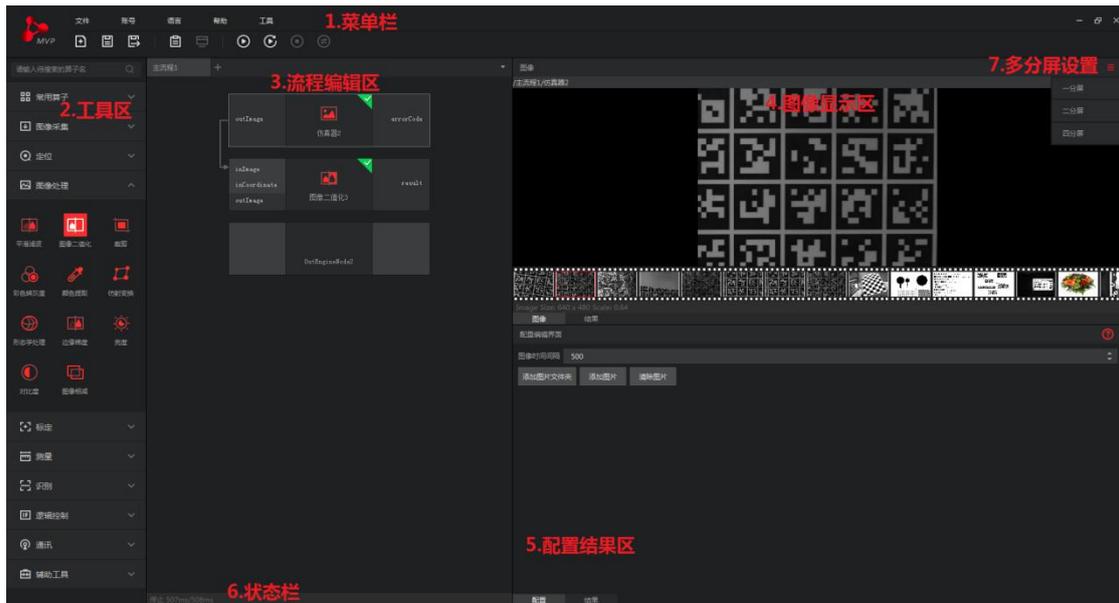


## 第 3 章 软件界面说明

### 3.1 主界面



双击  启动软件，随后即进入 MVP 主界面，如下图所示。



- **标题栏：**包含菜单栏及快捷工具条。
- **工具区：**视觉方案搭建所需要的算子区域，包含图像采集、定位、图像处理、标定、测量、识别、逻辑控制、通讯及辅助工具这几类功能。
- **流程编辑区：**视觉方案流程编辑区域。
- **图像显示区：**图像显示区域。
- **配置结果区：**通过下方标签切换，对选中的算子进行参数配置或查看算子运行后的结果信息。
- **状态栏：**状态显示区，显示所选算子的运行耗时及整个视觉解决方案的运行耗时。
- **多屏设置：**包括一分屏、二分屏、四分屏三种设置，可以将流程编辑区的任意一子流程的输出拉到任意一屏幕进行绑定，即可显示对应子流程的图像信息，并且支持在连续运行过程中进行、切换绑定。



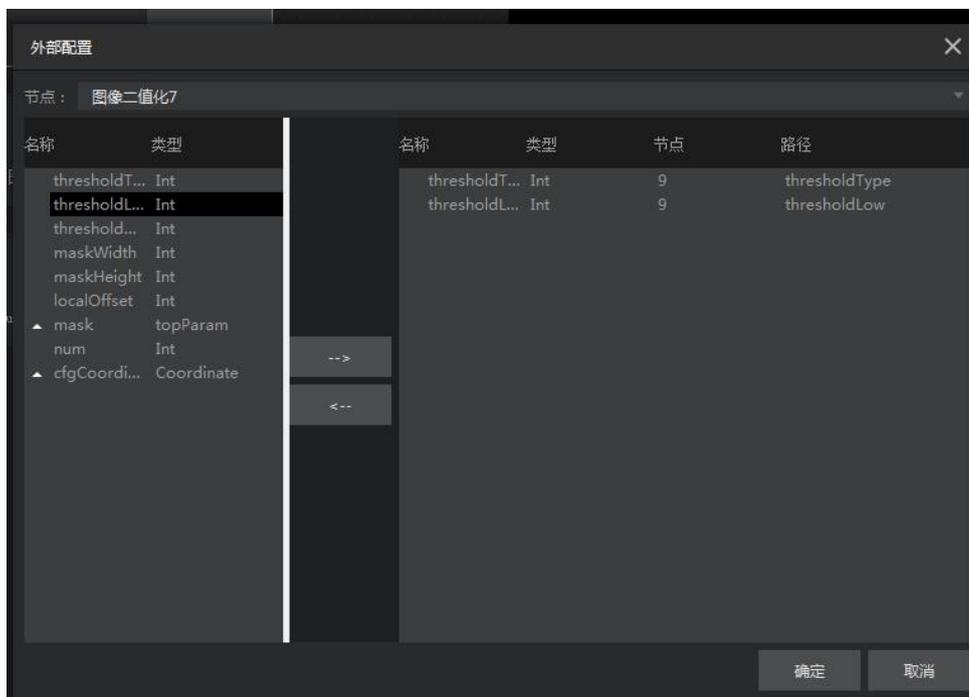
**文件：** 文件菜单有新建工程、打开、最近打开、打开示例、保存、另存为、导出工程和外部配置等子菜单。

- **新建工程：** 创建一个新的算法平台工程，当前工程未保存时会给出相应的保存提示，用户按需选择即可。
- **打开：** 打开之前保存的算法平台工程。
- **最近打开：** 打开最近打开过的算法平台工程。
- **打开示例：** 打开算法平台中自带的示例工程，主要包含常见的视觉解决方案的算法平台工程。
- **保存：** 保存当前配置好的视觉解决方案为算法平台工程文件，其后缀名为.job。
- **另存为：** 将当前视觉解决方案另存为到指定的路径。
- **导出工程：** 将当前配置好的视觉解决方案导出为 C++或 C#工程，用于对当前视觉解决方案进行二次开发，方便嵌入现有的视觉检测设备。导出工程软件包含指定解决方案算子代码和引擎层框架代码，导出解决方案代码可集成到客户系统软件之中，提供同一套视觉解决方案 C 接口，接口无差异化编程，提供 C++和 C#两种开发语言 Demo。MVP 包含软件框架 4 层所有代码，为客户解决通用应用问题，目前不提供软件界面定制功能，界面组件拖拉式开发，可快速出解决方案。

导出工程会生成 bin、MFCMVsSample 和 Job 文件，bin 文件夹下保存着算法平台的安装文件（依赖库、示例和配置等）。可用于 修改 Job 文件配置，制定视觉解决方案；MFCMvsSample 文件夹下为导出的基于 MFC 的 C++代码；Job 文件为在算法平台中配置的视觉解决方案。打开 MFCMvsSample 文件夹，找到 MFCMvsSample.sln，使用 VS2013 打开该文件。在 VS2013 编译器中，运行该程序。会弹出如图 3-5 所示的 MFC 界面。点击 **【openProject】** 按钮，选择导出的 Job 文件，加载成功后，openProject 按钮置灰，其他按钮使能。点击**【runOnce】**可以执行得到 Job 的单个运行结果。点击**【Run】**连续 运行 Job。在图像显示界面会显示 Job 中二值化图像处理后的结果。同时，输出参数 显示列表会实时更新显示每次运行后通过输出引擎节点导出的输出参数。该程序和算 法平台运行后得到的结果一致。用户可以很方便的在算法平台中配置视觉解决方案， 导出

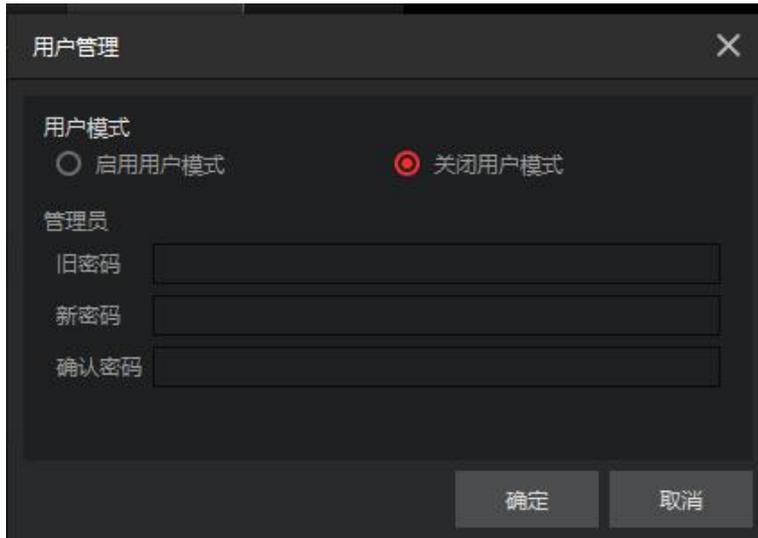
工程后再 MFC 程序中做相应的定制开发。当需修改视觉方案时，可打开 bin 文件夹的 MVP.exe，加载 Job 文件，修改视觉方案，待算法平台验证 OK 后，保存 Job，在导出的 C++工程中不需要修改代码，重新加载修改的 Job 文件，可获得同样的运行结果。通过这种方式可以很方便的将视觉方案嵌入视觉设备。导出的工程可以很方便的嵌入现有的视觉检测设备，算法平台提供了供用户配置参数的外部配置功能模块及一套通用的 C 接口。

**外部配置：**此功能配合导出工程使用，选取当前视觉解决方案中相关算子的相关参数为外部配置参数，即可在导出的 C++或 C#工程中直接对此参数进行相应的配置，如下图所示。



**账号：**账号菜单有用户管理、分配权限、切换用户三个子菜单项。

- **用户管理：**用于开启用户模式，可以在弹出的对话框中对管理员密码及技术、操作员的权限和密码进行相应的设置，如下图所示。



- **分配权限：** 启用用户模式后，并开启技术员权限的情况下，管理员用户可以自定义为技术员分配视觉解决方案中算子的权限，使技术员可以对相应的算子的参数进行配置，如下图所示。



- **切换用户：** 启用用户模式后，在开启技术员和操作员权限后，可以通过输入相应角色的密码来切换角色，如下图所示。



**语言：**语言菜单包含中文、英文两个子菜单项，用户可以点击相应的菜单项来对 MVP 界面进行中英文语言切换。

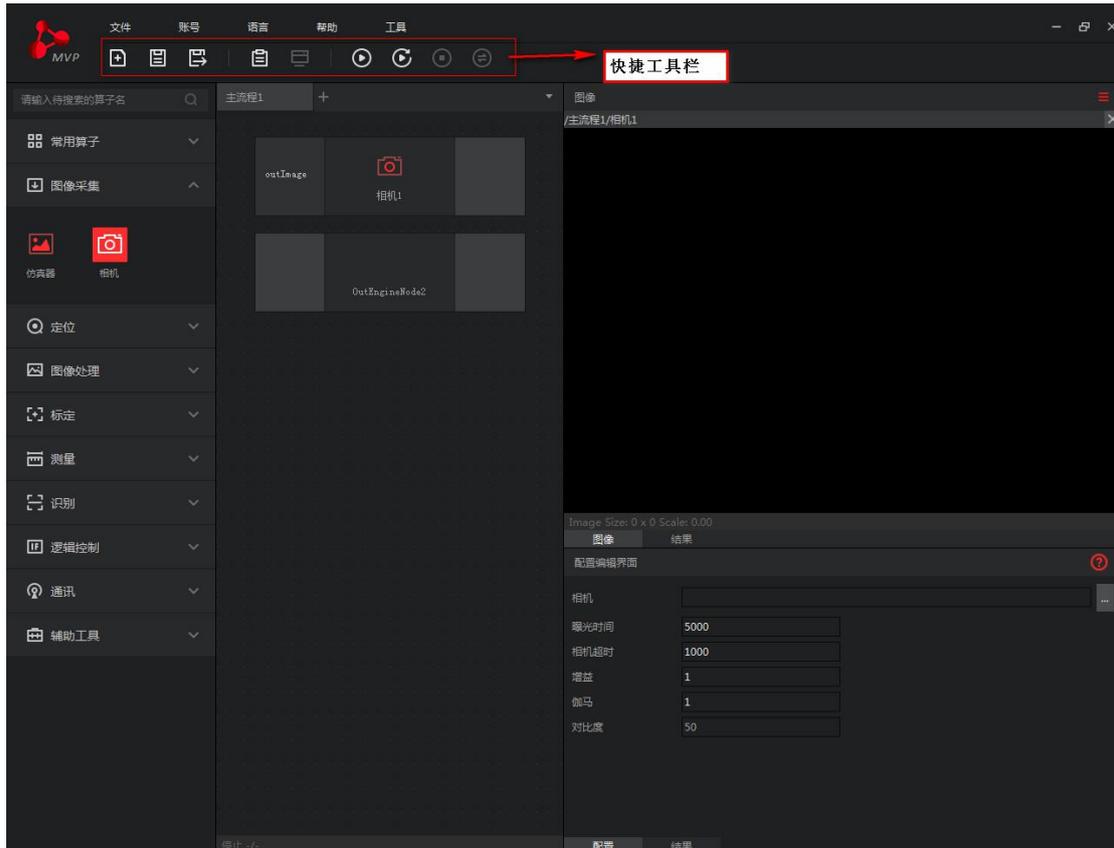
**帮助：**帮助菜单包含帮助文档和关于两个子菜单项。

- **帮助文档：**打开 MVP 的用户手册，以此来获取 MVP 的相关帮助信息。
- **关于：**可以查看当前 MVP 软件的版本信息及版权信息。

**工具：**工具菜单包含相机管理工具子菜单项，通过该子菜单项可以打开 MV Viewer 工具对 MVP 需要连接的相机进行参数配置。

## 3.1.2 快捷工具条

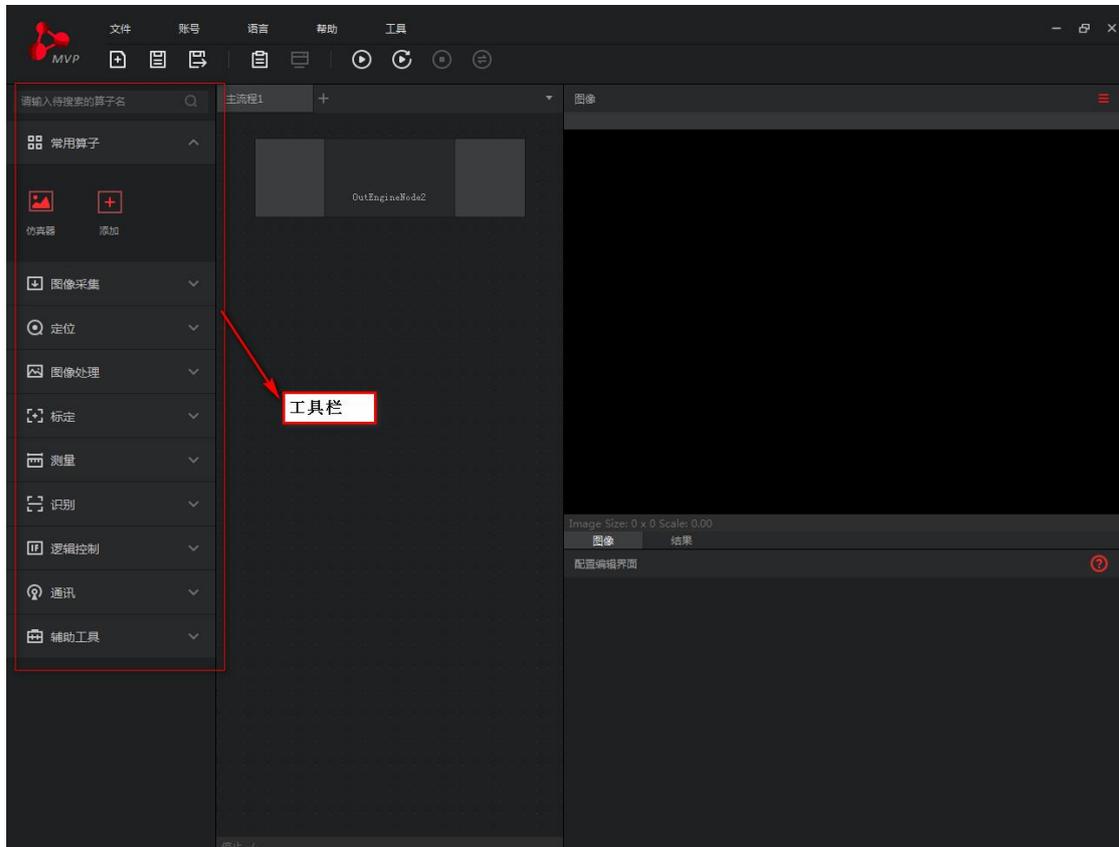
快捷工具条位于菜单栏下方，工具条中的按钮可以快速方便地对视觉方案进行相应的操作，如下图所示。下图中从左往右的按钮的含义依次为新建工程、保存、另存为、打开示例、运行界面、运行一次、连续运行、停止运行、同步多流程。



- **新建工程：**创建一个新的算法平台工程，当前工程未保存时会给出相应的保存提示，用户按需选择即可。
- **保存：**保存当前配置好的视觉解决方案为算法平台工程文件，其后缀名为.job。
- **另存为：**将当前视觉解决方案另存为到指定的路径。
- **打开示例：**打开算法平台中自带的示例工程，主要包含常见的视觉解决方案的算法平台工程。
- **运行界面：**弹出运行界面弹窗。
- **运行一次：**单击后运行一次当前视觉解决方案。
- **连续运行：**单击后连续运行当前视觉解决方案。
- **停止运行：**单击后提前终止目前正在运行的视觉解决方案。
- **同步多流程：**当前视觉解决方案中存在多个异步流程时，若需要同步流程之间的数据则单击此按钮，各个流程将会同步。

## 3.2 工具

工具栏如下图所示



工具区包含常用算子、图像采集、定位、图像处理、标定、测量、识别、逻辑控制、通信和辅助工具

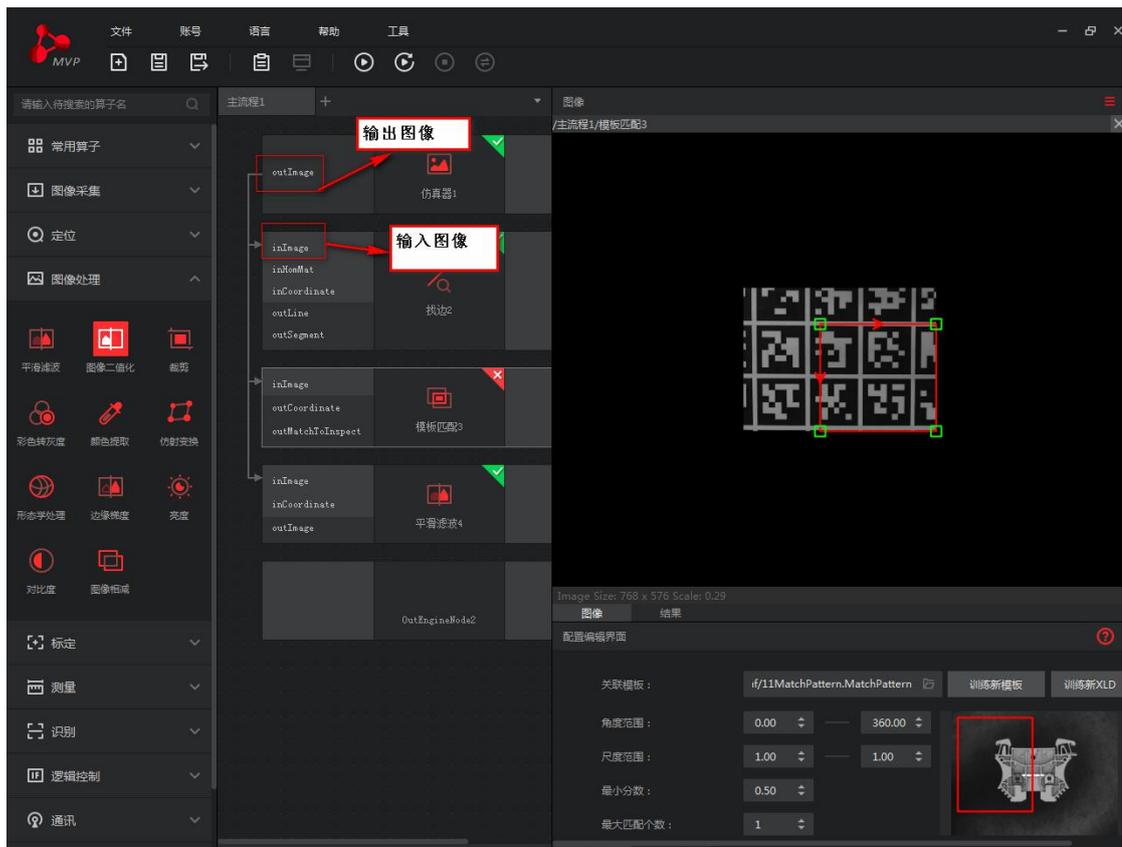
- **常用算子**：可以视自己使用习惯自定义常用算子到此类中。
- **图像采集**：分为相机和仿真器，相机是从工业相机获取图像，仿真器则是从本地获取图像。
- **定位**：类算子主要是根据不同的算法配置定位到图像中的特征并标识来。
- **图像处理**：含图像基本处理类算子，一般作用于图像预处理或形态学处理。
- **标定**：包含棋盘格标定、N点标定以及读取坐标文件算子，作用于不同坐标系之间的转换。
- **测量**：类算子主要完成测量距离、夹角等基本测量功能。
- **识别**：要包含一维码、二维码以及字符识别算子。
- **逻辑工具**：要用于视觉解决方案中数据的逻辑处理。

- **通信**：含业界常用的工业通信协议，支持 TCP/IP 和串口两种通信方式。
- **辅助工具**：包含保存图像、循环次数控制等常用算子。

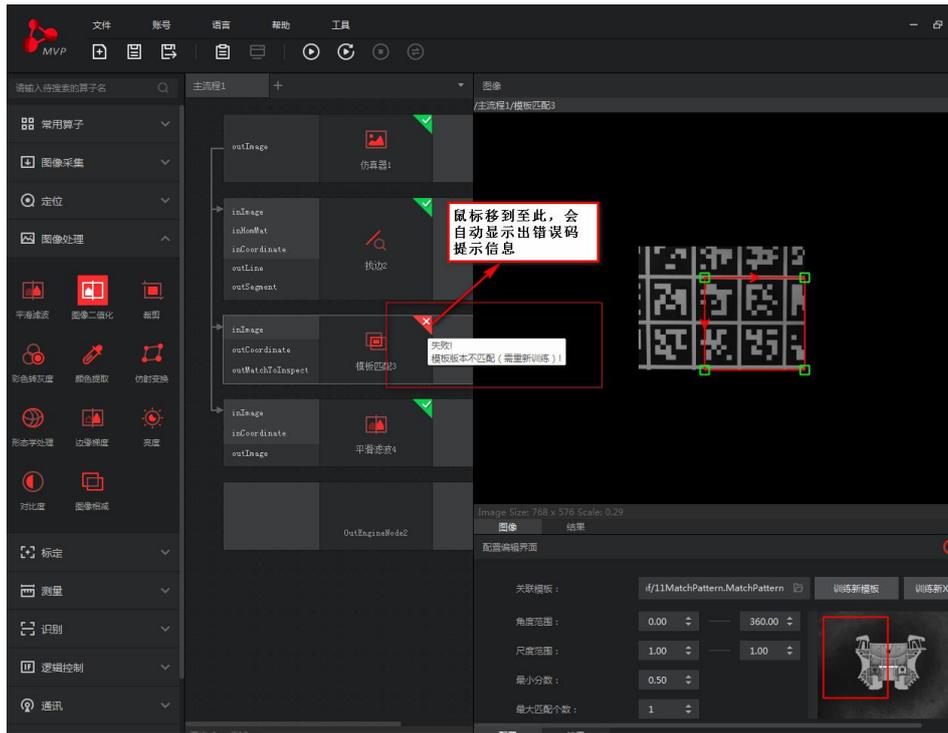
## 3.2.1 通用参数与功能介绍

### 参数前缀说明

默认情况下，每个算子的输入参数前缀为 **in**，输出参数前缀为 **out**。



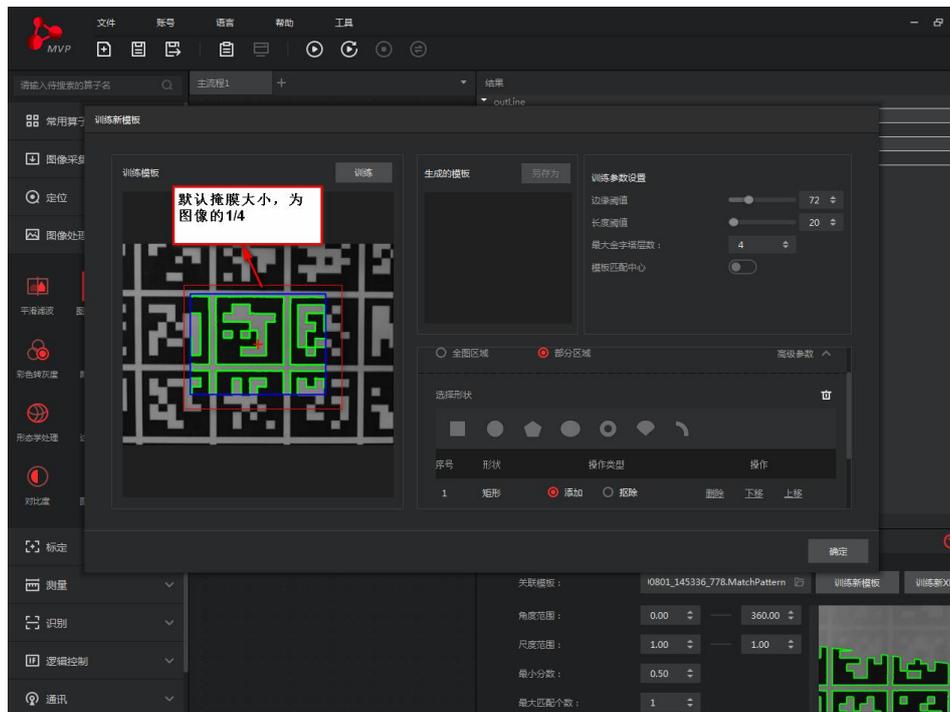
### 错误码



错误码提示

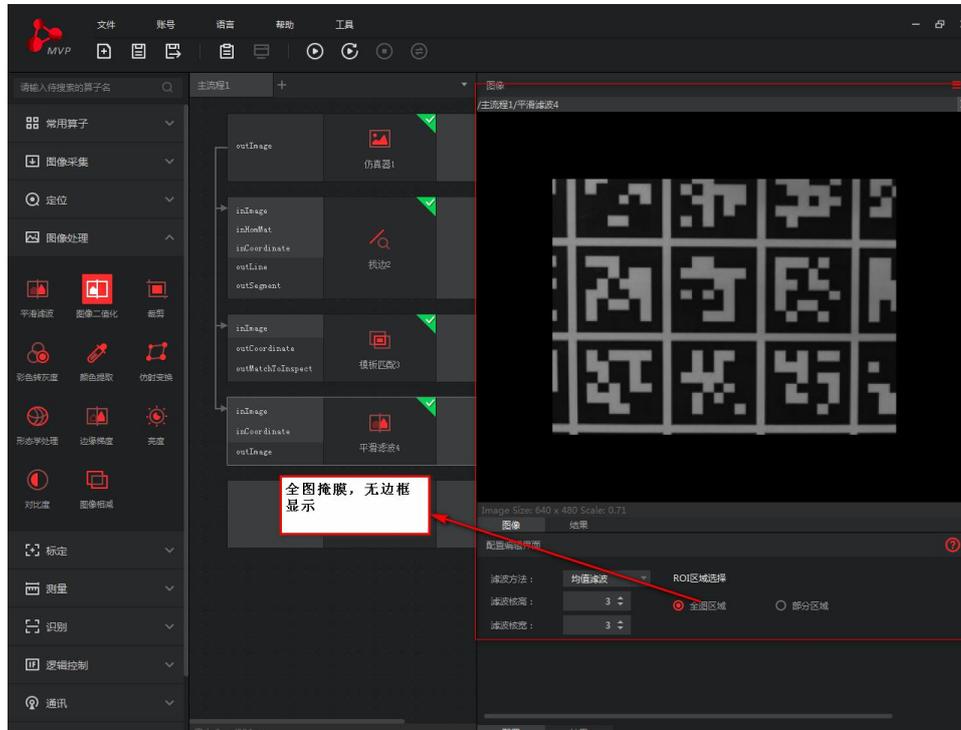
## 掩膜操作

目前掩膜支持的形状有矩形、圆、椭圆、多边形、圆环、扇形以及同心圆，当掩膜作用于算子的训练参数时，默认形状是矩形，大小为原图的 1/4；当作用



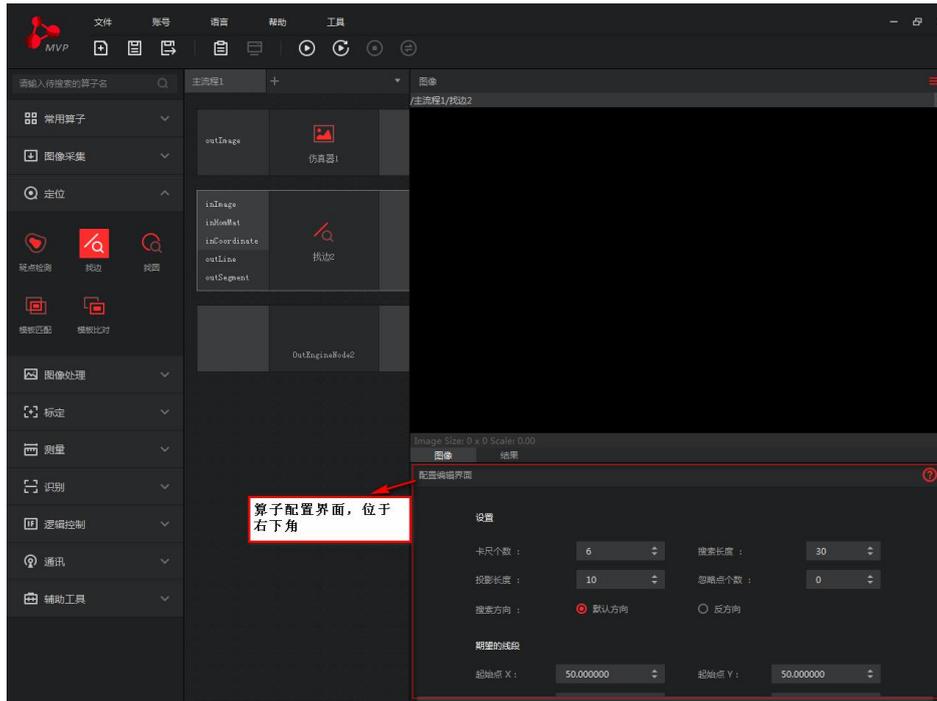
## 训练掩膜

于运行参数时，默认大小是全图，界面上无掩膜边框。具体如下：



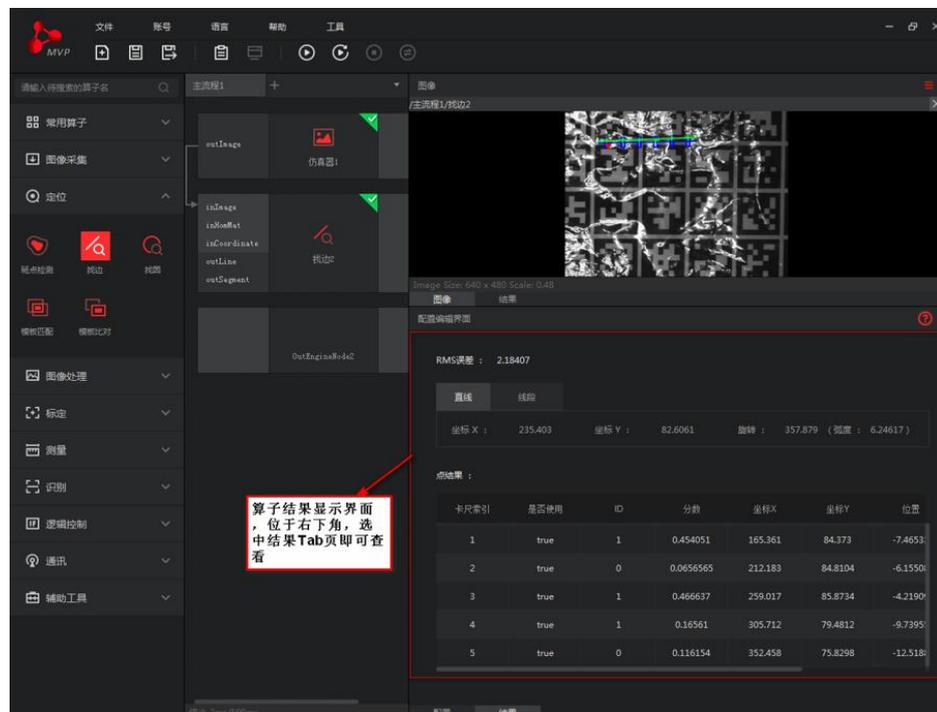
## 全图掩膜

- 添加算子  
选择待添加算子之后，双击或者按住左键把该算子拖到流程编辑界面。
- 算子配置参数

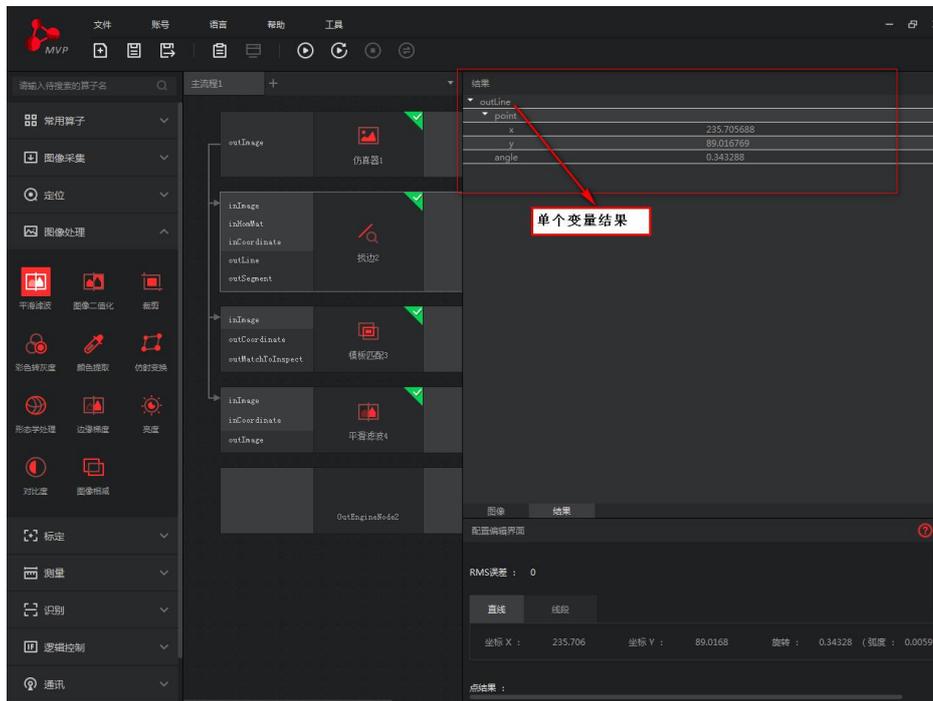


算子配置参数一般分为常用参数、高级参数，高级参数默认隐藏，需要设置时点击对应按钮打开可进行配置。

## ● 算子输出结果



算子执行之后的结果展示区域，一般以表格形式展示，因算子之间的差异，每个算子所展示的字段不同。



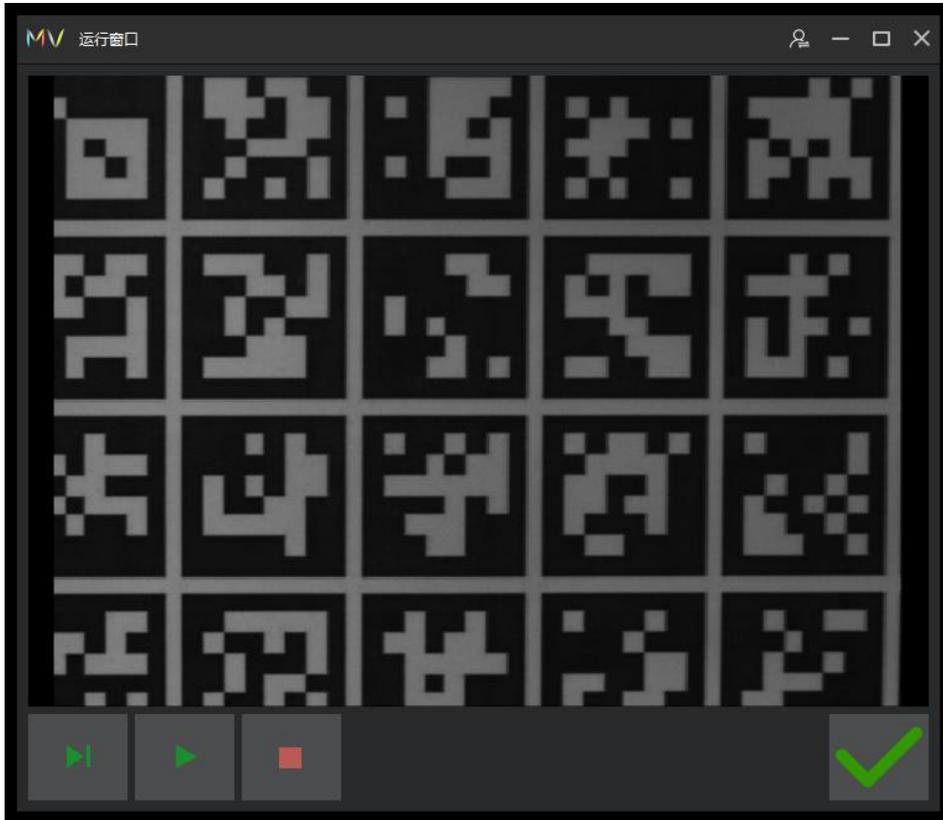
除了可以查看算子表格结果数据以外，还可查看算子单个变量的结果，首先把图像 Tab 切换到结果 Tab，然后选中待查看变量，按住左键，拖到右上角显示区域进行查看。

## 3.3 切换用户功能

启用用户模式后，可以根据实际需要，切换不同的角色来运行 MVP，达到视觉解决方案权限管理的目的，各个角色的权限如下表所示。

角色	权限
管理员	MVP 中的所有操作均可操作。
技术员	只可对管理员分配的算子进行参数配置，不可添加算子和流程。
操作员	只能运行已经配置好的视觉解决方案。

切换角色时，当切换为操作员时，此时 MVP 将只显示一个可运行的界面，界面上只有图像显示区、运行停止按钮以及运行结果，如下图所示；切换为管理员与技术员时，将显示 MVP 主界面。

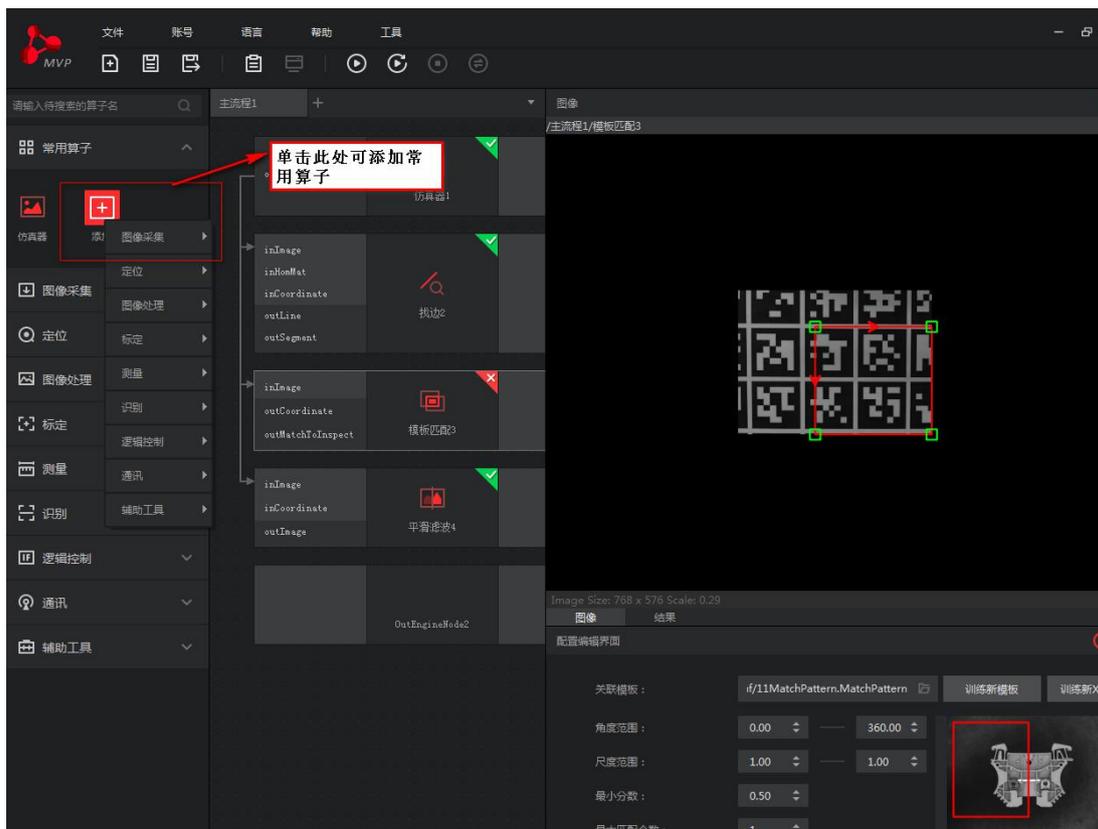


## 第 4 章 视觉工具

### 4.1 常用算子

#### 4.1.1 概述

目前常用算子中只包含仿真器算子，用户可根据自己使用习惯，自定义算子加入此类算子大类中，具体操作可如下图所示：



### 4.2 图像采集

#### 4.2.1 概述

图像采集算子主要功能是获取图像源，可以从本地或者相机获取。

#### 4.2.2 仿真器

图像采集算子主要功能是获取图像源，可以从本地或者相机获取。

## 配置参数

仿真器配置参数表

参数名称	数据类型	说明
图像时间间隔	-	可以调整连续运行时图像切换的时间间隔
仿真器路径	-	可以自由选择图像源文件夹

## 输出参数

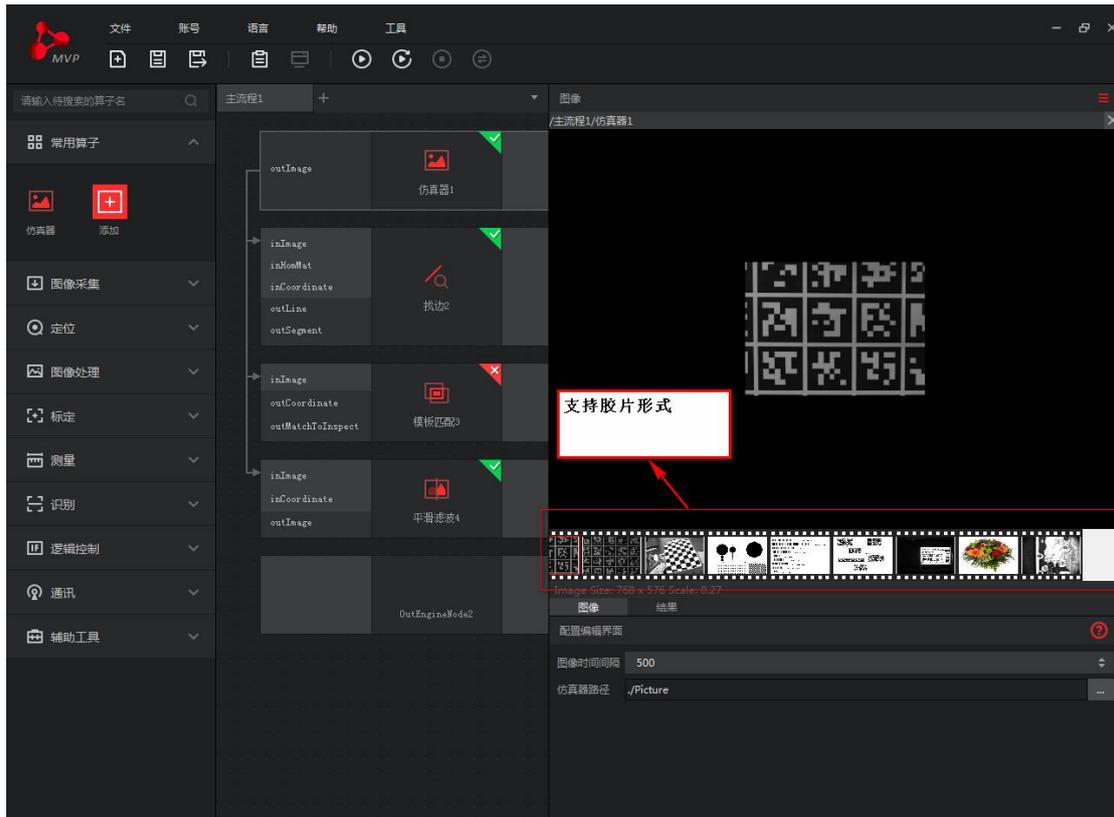
输出参数主要包含仿真器算子加载图片的路径以及加载的图片，该输出参数可以作为其他算子 `inImage` 的输入。输出参数通过拖拉到结果界面可以直接显示。

仿真器输出参数表

参数名称	数据类型	说明
<code>costTime</code>	<code>int</code>	运行时长
<code>errorCode</code>	<code>int</code>	错误识别码
<code>outImagePath</code>	<code>String</code>	输出加载图片的路径
<code>outImage</code>	-	输出加载图片大小，类型
<code>result</code>	<code>bool</code>	输出仿真器加载图片结果

## 操作流程

通过鼠标左键双击或拖拉仿真器算子添加仿真器，仿真器默认路径为安装运行目录下自带的图片路径，通过点击仿真器路径可以更改图片路径。仿真器加载图片显示方式分为两种，一种是以电影胶片的模式显示当前路径下所有的图片，在运行一次和停止状态下可以选中胶片图片，运行时就会在选中图片之间自动来回切换，连续运行下无法选中胶片图片；另一种方式是在图像显示窗口中放大显示图片，在连续运行状态下，图片会依次显示，当鼠标移到某张胶片图像上时，图像也会在显示窗口中放大显示。仿真器算子主要是通过 `outImage` 参数为其他算子提供本地处理图片。



## 注意事项

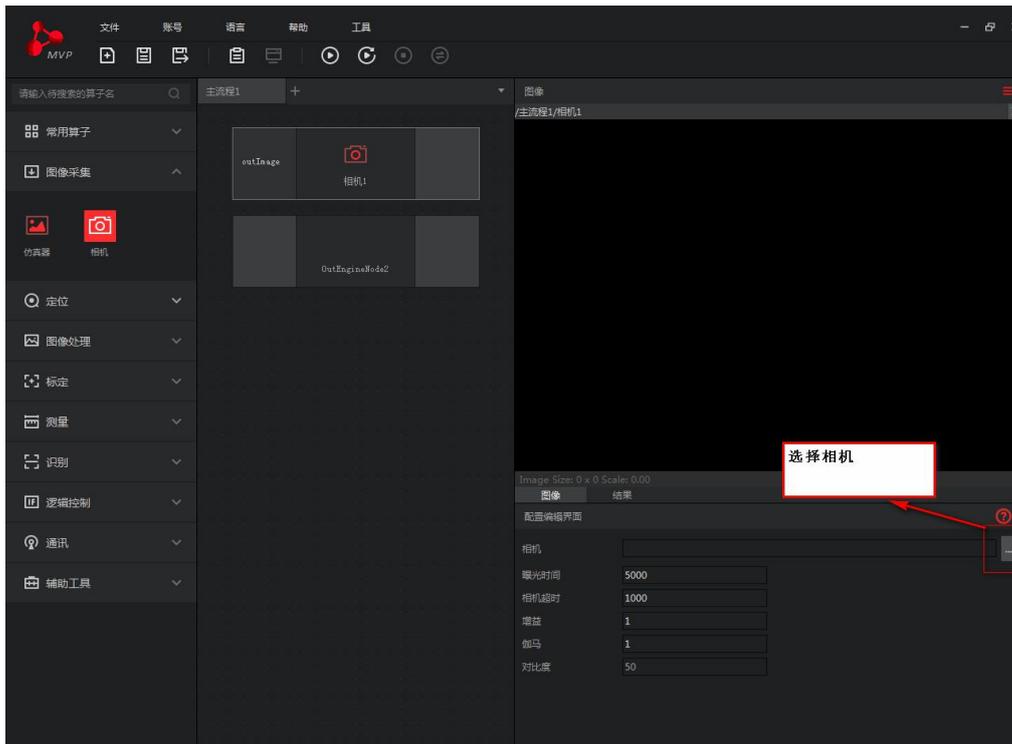
仿真器算子在连续运行时可以自动按照图像胶片序列切换显示图片，若在连续运行前选中了胶片图片，连续运行时就会在选中图片之间自动来回切换。当点击停止后，显示区域会清空，再次点击运行一次才会有图片显示。以下算子在连续运行状态下停止时都会清空显示区域的图像与算法处理结果。

## 4.2.3 相机

相机算子主要用于从相机采集图片，目前只支持 USB、GIGE 以及 X86 智能相机，相机配置硬触发模式和软触发模式。软触发模式下，每执行一次会向相机下发一次软触

发命令；硬触发模式，则等待相机外触发信号，从相机获取一帧图像。与仿真器不同的是将实时采集图片提供给其他算子作为输入图像使用。

## 操作流程



设置 IP 过程与 MVViewer 一致，双击选中相机，相机配置参数会显示该相机的 Key。单次运行，拉一帧相机的流；连续运行，连续拉相机的流。

## 配置参数

相机配置参数表

参数名称	数据类型	说明
Camera	-	相机配置设备显示
exposureTime	int	曝光时间
timeOutLen	int	硬触发超时时间

<b>gainRaw</b>	int	增益
<b>gamma</b>	int	----
<b>contrast</b>	int	对比度

相机配置参数主要包含相机设备选择，曝光时间、增益等。

## 输出参数

相机输出参数表

参数名称	数据类型	说明
<b>costTime</b>	int	运行时长
<b>errorCode</b>	int	错误识别码
<b>outImage</b>	-	输出加载图片大小，类型
<b>result</b>	bool	输出相机加载图片结果

输出参数主要包含相机算子从相机拉流读取的图片，该输出参数可以作为其他算子 `inImage` 的输入。输出参数可以直接拖拉到结果界面显示。

## 4.3 定位

### 4.3.1 斑点检测

斑点检测算子主要检测和定位图像中指定灰度值范围内、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，另外还可以通过斑点检测来发现图像中的特征在拓扑结构上的关联。通过对斑点进行面积过滤，圆度过滤和矩形度过滤可以对检测到的斑点进行筛选，从而得到预期斑点。利用得到的检测斑点，可以作为后续算子的粗定位也可对零件表面进行缺陷检测。如图 3.1.1 所示，利用圆度过滤和面积过滤可以得到零件轮廓边缘。

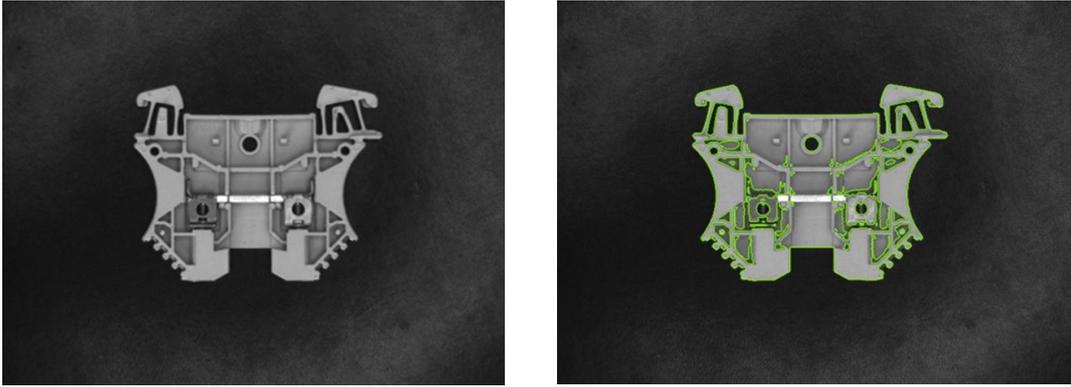


图 3.1.1 零件轮廓提取

如图 3.1.2 所示，利用斑点检测算子可以对 CCD 表面进行缺陷检测。

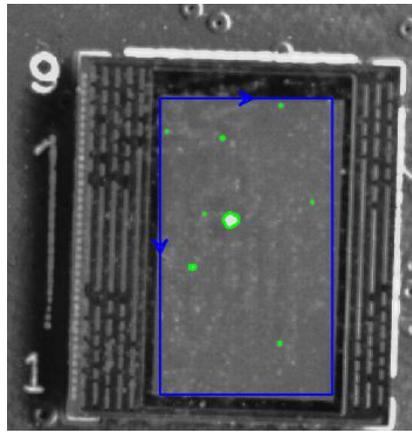


图 3.1.2 CCD 缺陷检测

## 配置参数

分组	参数名称	数据类型	取值范围	默认值	说明
二值化 Binarization	阈值 Threshold	单选按钮	自动阈值/硬 阈值	自动 阈值	设置二值化时采用的 阈值策略
	极性 Polarity	单选按钮	黑底白点/白 底黑点	黑底 白点	极性是指斑点像素值 是否高于背景像素
输出过滤	检测模式	单选按钮	全部检测/ 只检测斑点/	全部 检测	是否检测斑点中带有 的孔洞

# MVP 算法平台用户手册

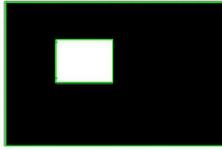
Export Mode	Detect Mode		只检测孔洞/ 填充孔洞后 检测		
	输出模式 Result Mode	单选按钮	正常斑点 / 仅连通域 / 凸包型斑点	正常 斑点	设置输出的斑点类型
	输出最大斑点个数 Max Detect Num	int	[0,4096]	100	设置可输出的斑点个 量
	排序模式 Sort Mode	单选按钮	面积/周长/质 心坐标 X/质 心坐标 Y/圆 度/矩形度	面积	设置排序的依据
	排序顺序 Sort Order	单选按钮	降序/升序	降序	设置排序的顺序
	面积过滤使能 Filter by Area	开关		关闭	按照面积大小对 blob 过滤输出
	最小面积 Min Area	int	[0, 2^31]	0	按面积过滤最小阈值
	最大面积 Max Area	int	[0, 2^31]	50000	按面积过滤最大阈值
	圆度过滤使能 Filter by Circularity	开关		关闭	按照圆度大小对 blob 过滤输出, 圆度表示与 标准圆的相似程度, 越 相似, 越接近 1
	最小圆度 Min Circularity	float	[1, 1000]	1	按圆度过滤最小值

	最大圆度 Max Circularity	float	[1, 1000]	2	按圆度过滤最大值
	矩形度过滤使能 Filter by Rectangularity	开关		关闭	按照矩形度大小对 blob 过滤输出，表示与标准矩形的相似程度，越相似，越接近 1
	最小矩形度 Min Rectangularity	float	[0, 1]	0	按矩形度过滤最小值
	最大矩形度 Max Rectangularity	float	[0, 1]	1	按矩形度过滤最大值
形态学 Morphology	Morph Mode 形态学操作模式	按钮	腐蚀/膨胀/开 操作/闭操作		设置形态学操作
	Kernel Size 操作核大小	int	[3,31]	3	设置操作核尺寸

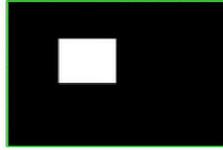
## 详细介绍

- **阈值：**表示图像中基于像素值的分界点。若设置自动阈值，表示可自适应寻找分割阈值，其原理是遍历不同的阈值，计算不同阈值下对应的前景和背景之间的类内方差，当类内方差取得极大值时，此时对应的阈值即为自适应阈值。当设置固定阈值时，以用户输入的阈值为基准，将高于阈值的像素点设置为白色，否则为黑色。
- **极性：**极性表示从图像背景到对象的整体灰度值变化，通常有由黑到白（0 黑底白点）、由白到黑（1 白底黑点）。根据所检测的特征与背景阈值的差异调整设置，若背景较暗，检测特征较亮，可以设置成黑底白点，反之亦然。
- **检测模式：**斑点中存在的孔洞也可视为一种特殊的斑点，可在检测模式中选择是否检测该类型的斑点。下例中将极性设置为白底黑点，检测模式设

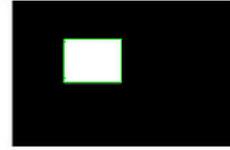
置全部检测时，将检测包括斑点和内部孔洞。当设置只检测斑点时，内部的孔洞将忽略。



(a) 全部检测

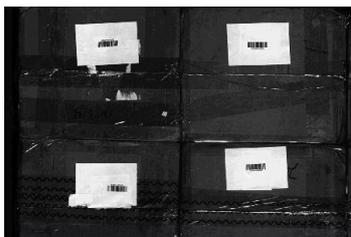


(b) 只检测斑点

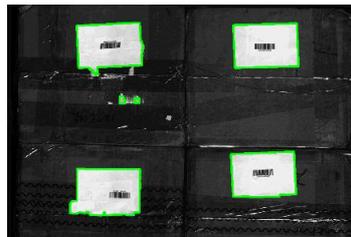


(c) 只检测孔洞

- **输出模式：**可选择输出包括轮廓、外接矩、圆度等完整的斑点信息，也可以只输出连通域信息，还支持输出凸包型的斑点。
- **输出最大斑点个数：**可设置输出的斑点数量。
- **排序模式：**对于输出结果，可以依据面积、周长、圆度等信息排序。
- **排序顺序：**可设置升序或者降序的排序方式。
- **面积过滤使能：**当需要输出特定面积范围内的斑点，可打开面积过滤使能。下例中检测箱体上的白色面单，其具有面积大致相同的特性，因此可利用面积过滤将其准确识别。



(a)输入图像



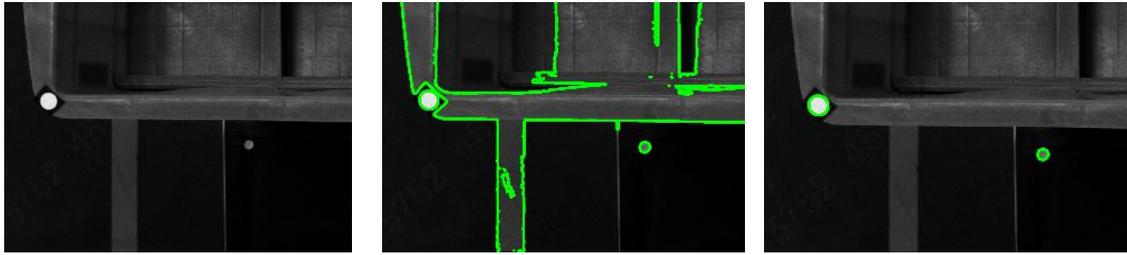
(b)面积过滤前



(c)面积过滤后

- **最小面积：**面积过滤时的面积下限。
- **最大面积：**面积过滤时的面积上限。
- **圆度过滤使能：**圆度表示与标准圆的相似程度，越相似，越接近 1。当需要输出特定圆度范围内的斑点，可打开圆度过滤使能。当待检测的斑点具

有特定的形状时，例如下例中要检测小车上的白色 Mark 点，使用圆度过滤能将其准确区分开。

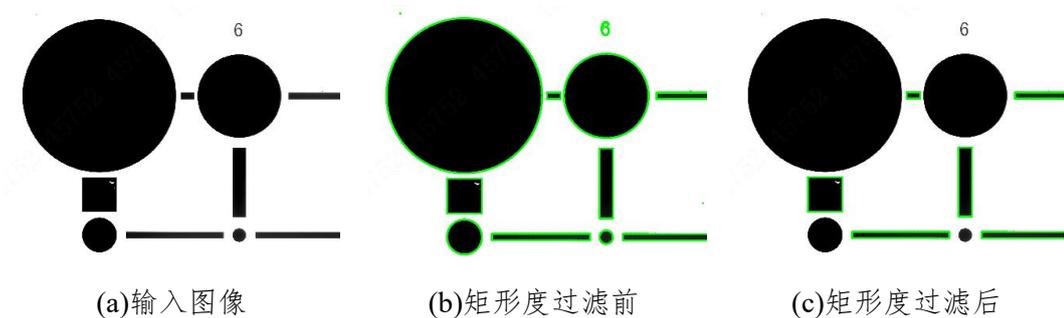


(a)输入图像

(b)圆度过滤前

(c)圆度过滤后

- **最小圆度：**圆度过滤时的圆度下限。
- **最大圆度：**圆度过滤时的圆度上限。
- **矩形度过滤使能：**矩形度表示与标准矩形的相似程度，越相似，越接近 1。当需要输出特定矩形度范围内的斑点，可打开矩形度过滤使能。下例中期望检测出特定矩形的斑点，利用矩形度过滤可以准确区分。



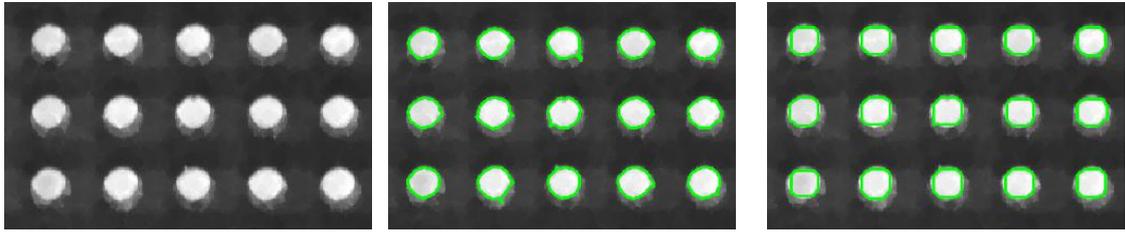
(a)输入图像

(b)矩形度过滤前

(c)矩形度过滤后

- **最小矩形度：**矩形度过滤时的矩形度下限。
- **最大矩形度：**矩形度过滤时的矩形度上限。
- **形态学操作模式：**有腐蚀、膨胀、开运算、闭运算四种。腐蚀是使检测的图像特征缩小，使用腐蚀可以消除图像的细节部分，产生滤波器的作用。膨胀是使检测的图像特征扩大。开操作是先腐蚀后膨胀，可以使图像的轮廓变得光滑，断开狭窄的间断和消除细的突出物。闭操作是先膨胀后腐蚀，同样可以使图像轮廓变得光滑，但与开操作相反，它能消除狭窄的间断和

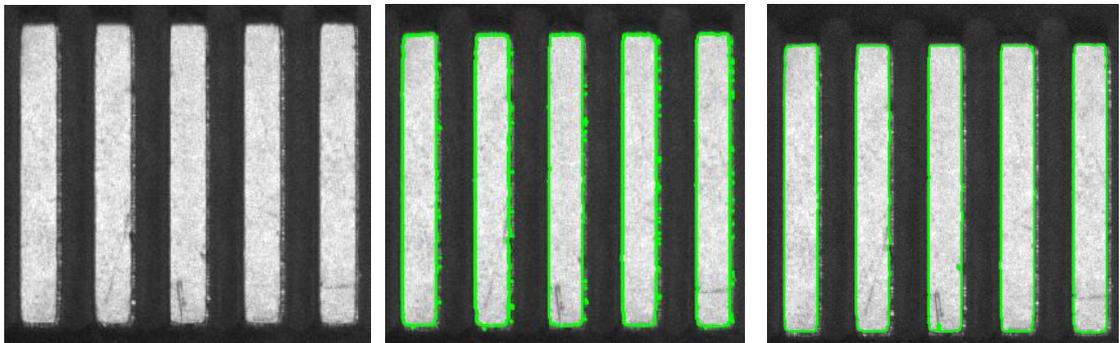
长细的鸿沟，消除小的空洞，并填补轮廓线中的裂痕。下例中，分别利用形态学操作使轮廓变光滑，以及消除杂乱点。



(a)输入图像

(b)形态学操作前

(c)形态学操作后



(a)输入图像

(b)形态学操作前

(c)形态学操作后

- **操作核大小：**滤波器滑窗尺寸的大小，滑窗尺寸为奇数，如 3、5、7...，滑窗尺寸越大则形态学处理程度越强。

## 输出参数

参数名称	数据类型	说明
ID	int	输出斑点 ID
面积 Area	int	斑点面积
质心 X Mass CenterX	float	斑点质心横坐标
质心 Y Mass CenterY	float	斑点质心纵坐标

是否为孔洞 IS Hole	String	0:Bole(斑点)/1:Hole(洞)
outCoordinate		输出斑点坐标信息
outBlobPoints		所有斑点详细信息
outCount		斑点个数
outFirstBlobArea		第一个斑点面积
outFirstBlobCenter		第一个斑点坐标

### 4.3.2 找边

找边工具主要用于查找图像中的直线，在图像中放置一系列卡尺工具，根据卡尺工具得到的边缘点集结果拟合出直线。

找边算子效果如下，其中蓝色卡尺为期望线段区域，红色箭头为搜索方向，绿色直线为卡尺找到的直线。

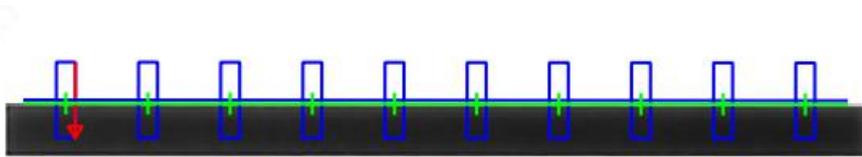
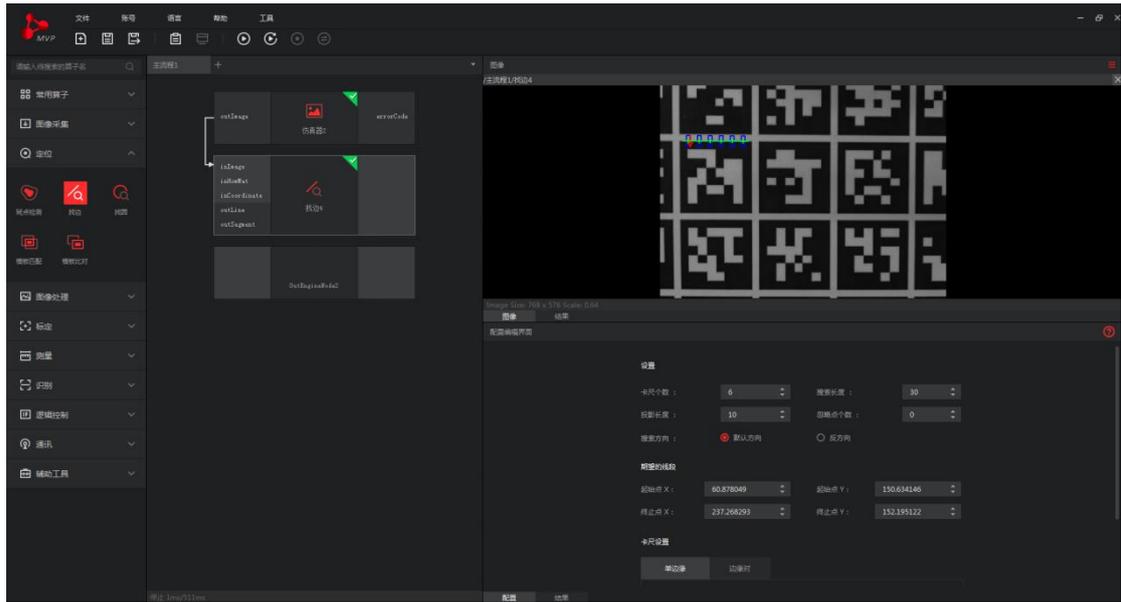


图 4.3.2 找边算子效果

操作流程：



进入配置界面配置卡尺个数、搜索长度、投影长度、忽略点数以及搜索方向；并设置卡尺设置，对比度阈值、高斯半径、排序模式等来调整找边卡尺工具，使得找边结果更加精准理想。

## 配置参数：

分组	参数名称	数据类型	取值范围	默认值	说明
设置 Settings	卡尺个数 Number of Calipers	int	[2, 100]	6	找边卡尺数量
	搜索长度 Search Length	int	[0, 1000]	30	找边搜索长度
	投影长度 Projection Length	int	[0, 1000]	10	与边平行方向投影长度
	忽略点个数 Num Points to Ignore	int	[0, 卡尺个数-2]	0	将原来拟合组内的参数移到这里，去掉搜索模式
	交换搜索方向 Swap Search	是否选中			搜索方向

## MVP 算法平台用户手册

	Direction				
期望的线段 Expected Line Segment	起始点 X Start X	float	[0,5000]	50	期望边的起点 X
	起始点 Y Start Y	float	[0,5000]	50	期望边的起点 Y
	终止点 X End X	float	[0,5000]	100	期望边的终点 X
	终止点 Y End Y	float	[0,5000]	100	期望边的终点 Y
卡尺设置(套用卡尺工具里面的参数) Caliper Settings	边缘模式 Edge Mode	枚举	单边缘、边缘对	单边缘	包括单边缘、边缘对
	边缘极性 Edge Polarity	枚举	由暗到明、由明到暗、任何极性	任何极性	找边极性
	对比度阈值 Contrast Threshold	int	[0, 255]	5	边缘对比度阈值
	过滤一半像素 Filter Half Size Pixels	int	[1, 100]	2	滤波宽度
	最大结果数 Max Detect Num				期望找到的边数量
排序模式 Sort By	枚举	按对比度排序, 按中心分数排序, 按搜索方向排序	按对比度排序	设置排序模式	

## 详细介绍

- **拟合方式：**有忽略点拟合和 Ransac 拟合。忽略点拟合就是忽略掉一部分点后再进行拟合。Ransac 是随机抽样一致性算法，它可以从一组包含“局外点”的数据集中，通过迭代方式估计直线参数。在拟合时使用最小二乘法进行最佳拟合，即利用已知数据得出一条直线，使之在坐标系上与已知数据之间的距离的平方和最小：



$$d = \sqrt{\frac{\sum_{i=1}^n (d_i)^2}{n}}$$

建立模型时利用直线的一般方程  $AX+BY+C=0$ ，随机选取两点构建直线模型，计算每个点到此直线的 TLS（Total Least Square），TLS 小于一定阈值时的点为符合模型的点，点数最多时的模型即为最佳直线模型，再根据此时的直线参数画出最终拟合直线。

- **边缘极性：**边缘极性表示从图像背景到对象的整体灰度值变化，通常有任何极性、由暗到明、由明到暗。例如当设置为由暗到明时，卡尺工具会根据卡尺搜索方向去搜索从暗到明变化的边缘。
- **边缘对比度：**设置边缘的对比度阈值，若该值设置的较大，则一些对比度不强的边缘将被过滤。
- **垂直滤波：**若设置采用垂直滤波，可增强水平方向的边缘特征。

### 注意：

- ◆ 找边算子配合模板匹配进行自动定位，当目标匹配不到时，找边卡尺会发生偏离，因此注意自动找边需保证图像中目标能被匹配到。
- ◆ 由于卡尺工具具有搜索方向，应根据实际情况来设置边缘极性参数，若不根据实际情况进行调整，容易引发错误。

## 输出参数列表

参数名称	数据类型	取值范围	默认值	说明
直线 outLine	CMVParamLinePA	--	--	输出匹配到的直线
线段 outSegment	CMVParamLineSeg	--	--	输出匹配到的线段

- **直线**：采用点角式表示直线，具体结构如图\*\*编号\*\*所示。其中 outLine 的类型为 CMVParamLinePA, point 类型为 CMVParamPoint2Df32, x, y 的类型为 float。

结果	
outLine	
point	
x	360.759399
y	216.975143
angle	24.681755

- **线段**：采用两点式表示线段，具体结构如图\*\*编号\*\*所示。其中 outSegment 的类型为 CMVParamLineSeg, startPoint 和 endPoint 的类型为 CMVParamPoint2Df32, x, y 的类型为 float。

结果	
outSegment	
startPoint	
x	83.711090
y	89.674934
endPoint	
x	637.807678
y	344.275360

## 4.3.3 找圆

找圆工具在图像中放置一系列卡尺工具，根据卡尺工具得到的边缘点集结果拟合出圆，用于圆的定位与测量。

找圆算子效果如下，其中蓝色卡尺为期望圆弧区域，绿色圆即为找到的圆。

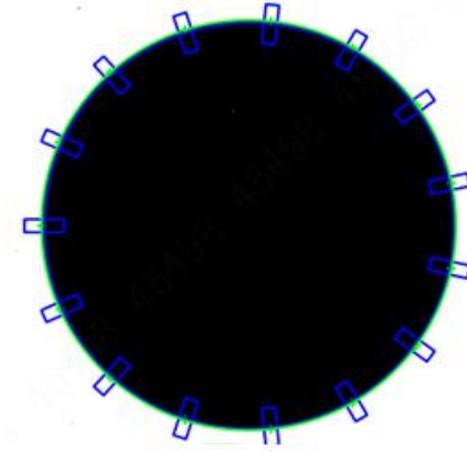
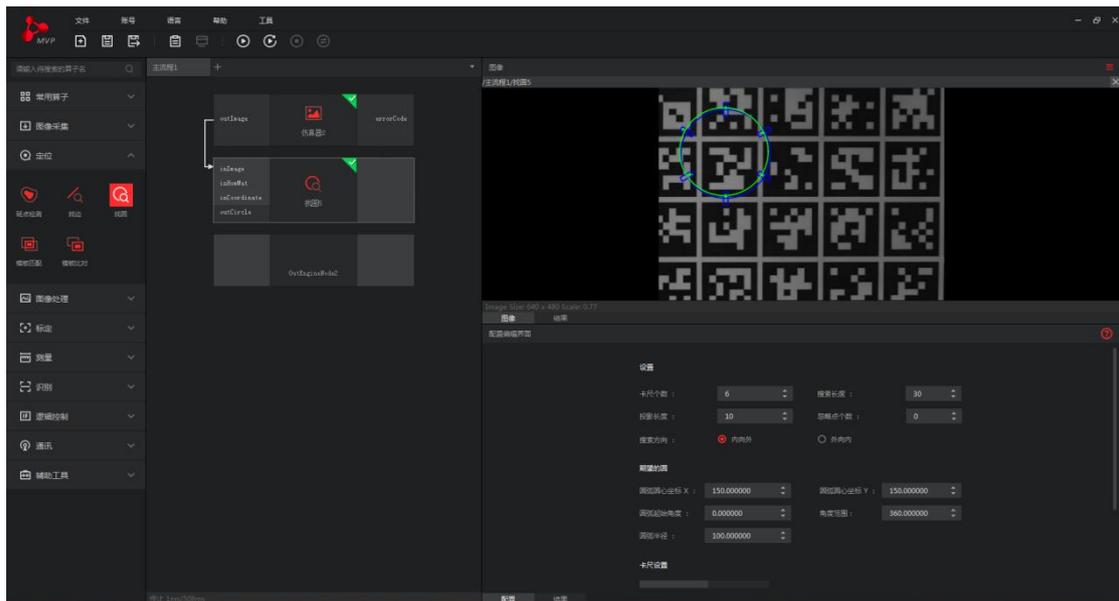


图 4.3.3 找圆算子效果

## 操作流程：



进入配置界面配置卡尺个数、搜索长度、投影长度、忽略点数以及搜索方向；并设置卡尺设置，对比度阈值、高斯半径、排序模式等来调整找边卡尺工具，使得找圆结果更加精准理想。

## 配置参数：

分组	参数名称	数据类型	取值范围	默认值	说明
----	------	------	------	-----	----

# MVP 算法平台用户手册

设置 Settings	卡尺个数 Number of Calipers	int	[3, 100]	6	找圆摆放卡尺数量
	搜索长度 Search Length	int	[1, 1000]	30	找圆搜索边界范围
	投影长度 Projection Length	int	[1, 1000]	10	找圆切线方向投影长度
	忽略点个数 Num Points to Ignore	int	[0, 卡尺个数-3]	0	拟合忽略点个数
	交换搜索方向 Swap Search Direction	Bool	默认方向, 反方向	默认方向	交换搜索方向
期望的圆 Expected Circular Arc	圆弧圆心坐标 X Arc Center X	float	[0,5000]	150	设置期望圆
	圆弧圆心坐标 Y Arc Center Y	float	[0,5000]	150	
	圆弧半径 Arc Radius	float	[1,5000]	100	
	圆弧起始角度 Arc Angle Start	float	[0, 360]	0	
	角度范围 Angle Range	flaot	[0, 360]	360	
卡尺设置 (套用卡尺工具里面的参数) Caliper	边缘模式 Edge Mode	枚举	单边缘、边缘对	单边缘	包括单边缘、边缘对、边缘极性等等
	边缘极性 Edge Polarity	枚举	由暗到明、由明到暗、任何极性	任何极性	边缘的极性方向

Settings	对比度阈值				
	Contrast Threshold	int	[0, 255]	5	边缘对比度阈值
	高斯半径 (过滤一半像素) Gauss Radius (Filter Half Size Pixels)	int	[1, 100]	2	滤波的宽度
	最大结果数 Max Detect Num				期望找到的最大结果数
	排序模式 Sort By	枚举	按对比度排序, 按中心分数排序, 按搜索方向排序	按对比度排序	设置排序模式

## 详细介绍

- **边缘极性：**边缘极性表示从图像背景到对象的整体灰度值变化，通常有任何极性、由暗到明、由明到暗。例如当设置为由暗到明时，卡直线工具会根据卡尺搜索方向去搜索从暗到明变化的边缘。
- **边缘对比度：**设置边缘的对比度阈值，若该值设置的较大，则一些对比度不强的边缘将被过滤。
- **拟合方式：**有忽略点拟合和 Ransac 拟合。忽略点拟合就是忽略掉一部分点后再进行拟合。Ransac 是随机抽样一致性算法，它可以从一组包含“局外点”的数据集中，通过迭代方式估计直线参数。在拟合时使用最小二乘法进行最佳拟合，即利用已知数据得出一条直线，使之在坐标系上与已知数据之间的距离的平方和最小：



$$d = \sqrt{\frac{\sum_{i=1}^n (d_i)^2}{n}}$$

建立模型时利用直线的一般方程  $AX+BY+C=0$ ，随机选取两点构建直线模型，计算每个点到此直线的 TLS（Total Least Square），TLS 小于一定阈值时的点为符合模型的点，点数最多时的模型即为最佳直线模型，再根据此时的直线参数画出最终拟合直线。

- **垂直滤波**：若设置采用垂直滤波，可增强水平方向的边缘特征。



- ◆ **找圆算子配合模板匹配进行自动定位**，当目标匹配不到时，找边卡尺会发生偏离，因此注意自动找圆需保证图像中目标能被匹配到。

## 输出参数列表

参数名称	数据类型	取值范围	默认值	说明
圆 outCircle	CMVParamLinePA	--	--	输出匹配到的圆

- **直线**：直线采用点角式表示。具体结构如图\*\*编号\*\*所示。其中，outCircle 类型为 CMVParamCircle, center 类型为 CMVParamPoint2Df32, radius,x,y 类型为 float。

结果	
outCircle	
center	
x	146.082870
y	144.631607
radius	99.106262

## 4.3.4 模板匹配

在模板图像中选中特征区域建立模板，在检测图像中进行匹配，定位产品在图像中的位置。配合其他工具使用，可引导其他工具跟随产品实时调整位置和角度。

模板匹配效果如下，其中蓝色框表示搜索区域，绿色框表示匹配到的对象。

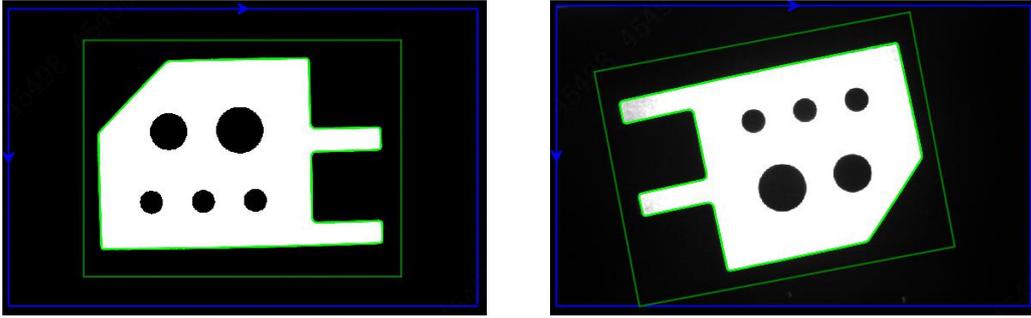
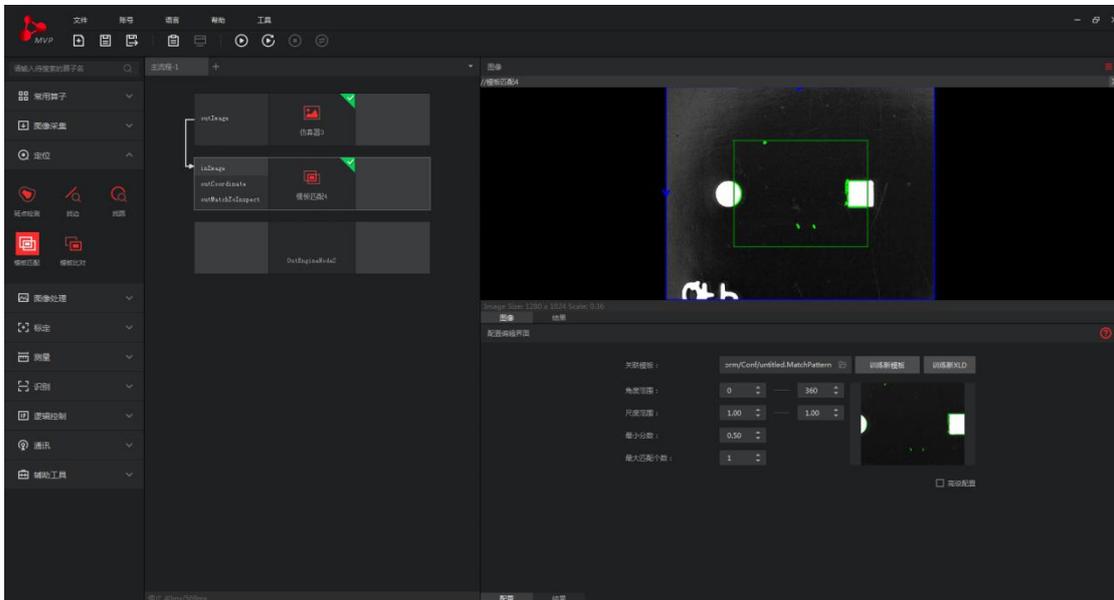


图 4.3.4 模板匹配效果

操作流程：

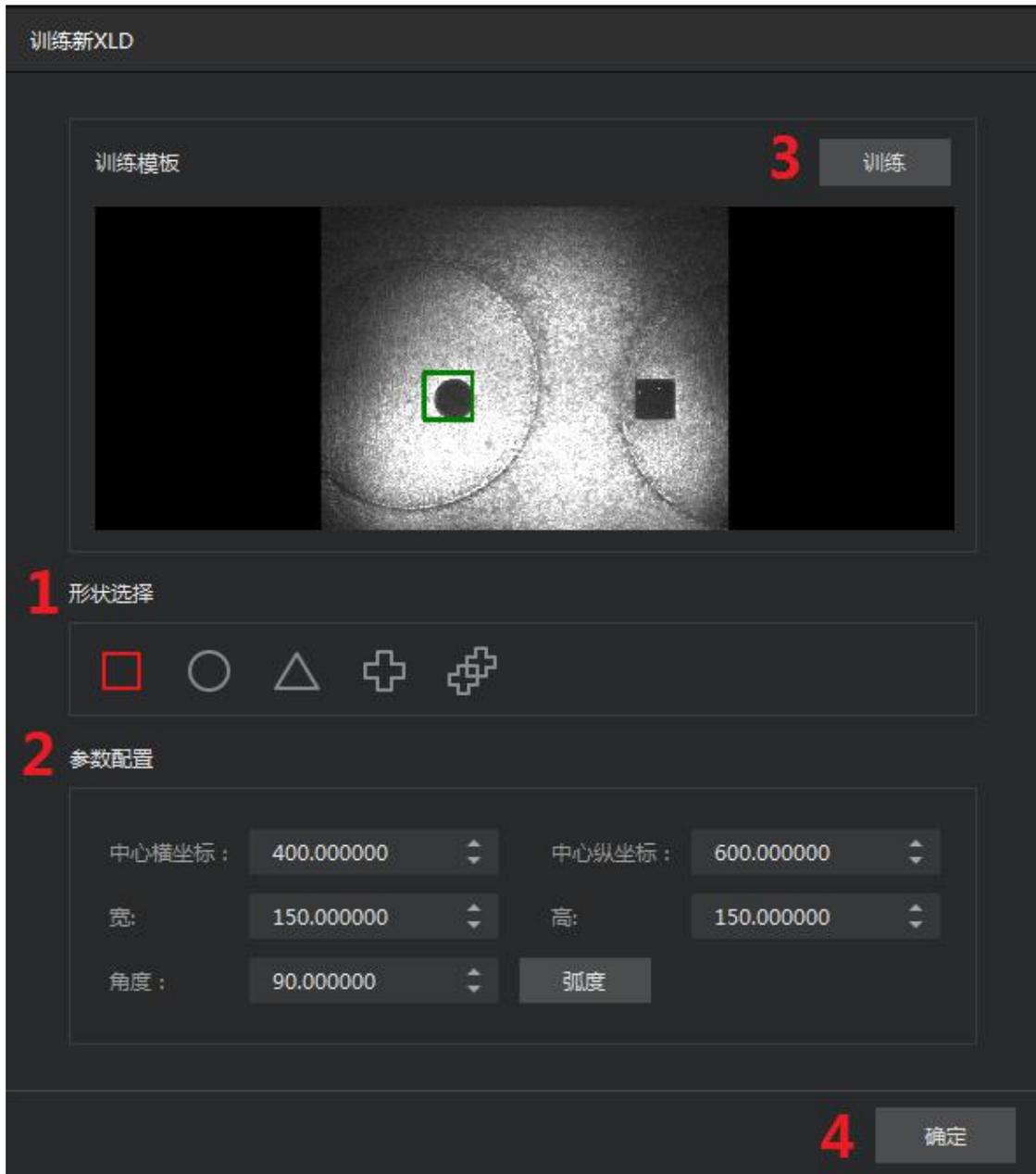


配置中设置关联模板：选择已训练好的模板文件、训练新模板、训练新 XLD。

训练新模板



训练新 XLD



## 配置参数

分栏	参数名称	数据类型	取值范围	默认值	说明
配置编辑 Config Editor	关联模板				
	Association Template				
	训练新模板				
	Training New				

	Template				
	训练新 XLD Training New XLD				
	最小匹配角度 minAngle	float	[-180,360]	0	匹配搜索最小角度
	最大匹配角度 maxAngle	float	[-180, 360]	360	匹配搜索最大角度
	最小匹配尺度 minScale	float	[0.1, 10.0]	1	匹配搜索最小尺度
	最大匹配尺度 maxScale	float	[0.1, 10.0]	1	匹配搜索最大尺度
	最小分数 Minimum Score	float	[0, 1]	0.5	内部匹配过滤阈值分数
	最大匹配目标个数 Maximum Number of Matches	int	[0, 50]	1	配置图像中匹配的最大数目

## 【训练新模板】

分组	参数名称	数据类型	取值范围	默认值	说明
训练新模板 Training New Template	训练模板 Training Template				
	ROI 区域选择 ROI Regional Selection	枚举			
	全图区域 Full graph				
	部分区域				

	Part graph				
	边缘阈值 Edge Threshold	int	[1,255]	72	训练过程, 边缘检测阈值
	长度阈值 Length Threshold	int	[0,500]	20	训练过程, 边缘长度过滤阈值
	生成的模板 Generated Template				

## 【训练新 XLD】

分组	参数名称	数据类型	取值范围	默认值	说明	
训练新 XLD Training New XLD	训练				保存训练模板特征数据	
	矩形 rectangle	中心横坐标	float	[0, 10000]	0	
		中心纵坐标	float	[0, 10000]	0	
		宽	float	[1, 1000]	100	
		高	float	[1, 1000]	100	
		角度	float	[0, 360]	0	
	三角形 triangle	中心横坐标	float	[0, 10000]	0	
		中心纵坐标	float	[0, 10000]	0	
		边长	float	[1, 1000]	50	
		角度	float	[0, 360]	0	
	圆 circle	圆心横坐标	float	[0, 10000]	0	
		圆心纵坐标	float	[0, 10000]	0	
		半径	float	[1, 1000]	50	
	十字星	中心横坐标	float	[0, 10000]	0	

# MVP 算法平台用户手册

	cross	中心纵坐标	float	[0, 10000]	0	
		边长	float	[1, 1000]	100	
		角度	float	[0, 360]	0	
	双十字星 double cross	中心横坐标	float	[0, 10000]	0	
		中心纵坐标	float	[0, 10000]	0	
		边长	float	[1, 1000]	100	
		角度	float	[0, 360]	0	

## 【高级参数内容】

分组	参数名称	数据类型	取值范围	默认值	说明	
高级参数 Advanced Configuration	最大金字塔层数 Maximum Pyramid Layer Number	int	[3, 7]	0	0 自适应计算金字塔层数，[3, 7] - 算法使用配置参数	
	边缘极性 Edge Polarity	不忽略极性 use_polarity	选择	0,1,2	0	0 表示不忽略，1 表示忽略全局，2 表示忽略局部
		忽略全局极性 ignore_global_polarity				
		忽略局部极性 ignore_local_polarity				
亚像素精度 subpixel	枚举	0,1,2	2	0 表示像素精度，1 表示亚像素一般精度，2 表示亚像素高精度，默认为 2		

	最大重叠率		float	[0, 1]	0.5	范围[0, 1], 默认为 0.5
	maxoverlap					
	模板匹配中心 originCenter	模板匹配中心 X originX	float	[0, width-1]	-1	模板匹配中心 (可选, 在训练前修改)
模板匹配中心 Y originY		float	[0, height-1]	-1		

## 详细介绍

- **最小分数**：匹配分数指特征模板与搜索图像中目标的相似程度，即相似度阈值，搜索到的目标在相似度达到该阈值时才会被搜索到，最大是 1，表示完全契合
- **边缘阈值**：训练过程中，边缘检测阈值，范围为 0~255
- **长度阈值**：训练过程中，边缘长度过滤阈值，范围为 0~500
- **最大匹配目标个数**：允许查找的最大目标个数
- **亚像素精度**：表示匹配的精度选项，可选像素精度、亚像素精度和亚像素高精度
- **最大重叠率**：当搜索多个目标时，两个被检测目标彼此重合时，两者匹配框所被允许的最大重叠比例，该值越大则允许两目标重叠的程度就越大



- 注意：**
- ◆ 模板训练区域和匹配区域的选择不要弄混。
  - ◆ 对于匹配目标可能有重叠的情况，应设置一定的最大重叠率。

## 输出参数

参数名称	数据类型	说明
第一个匹配目标中心 outFirstMatchCenter	MvSPoint2Df32	第一个匹配目标中心位置

匹配目标的坐标 <b>outCoordinate</b>	Coordinate	第一个匹配目标的坐标， 用于定位
匹配结果参数 <b>outMatchToInspect</b>	MatchToInspect	用于模板比对输入的模 板匹配输出参数
匹配到目标个数 <b>outCount</b>	Int	模板匹配到目标的个数
第一个匹配目标分数 <b>outFirstMatchScore</b>	float	第一个匹配目标分数
第一个匹配目标旋转 角度 <b>outRotation</b>	float	第一个匹配目标旋转角 度

### 4.3.5 模板比对

模板比对算子主要完成模板图像或者模板区域与检测图像或者检测区域之间的差异检测。可以用于缺陷检测、图像差异检测等应用。

模板比对效果如下，其中左图为训练模板图像，右图为模板比对的结果图像。

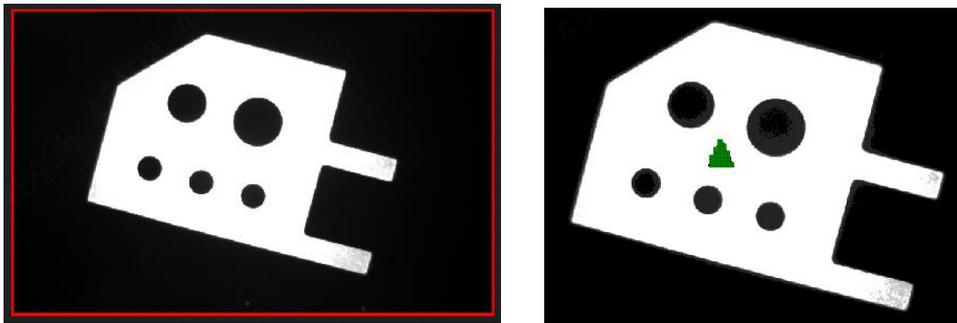
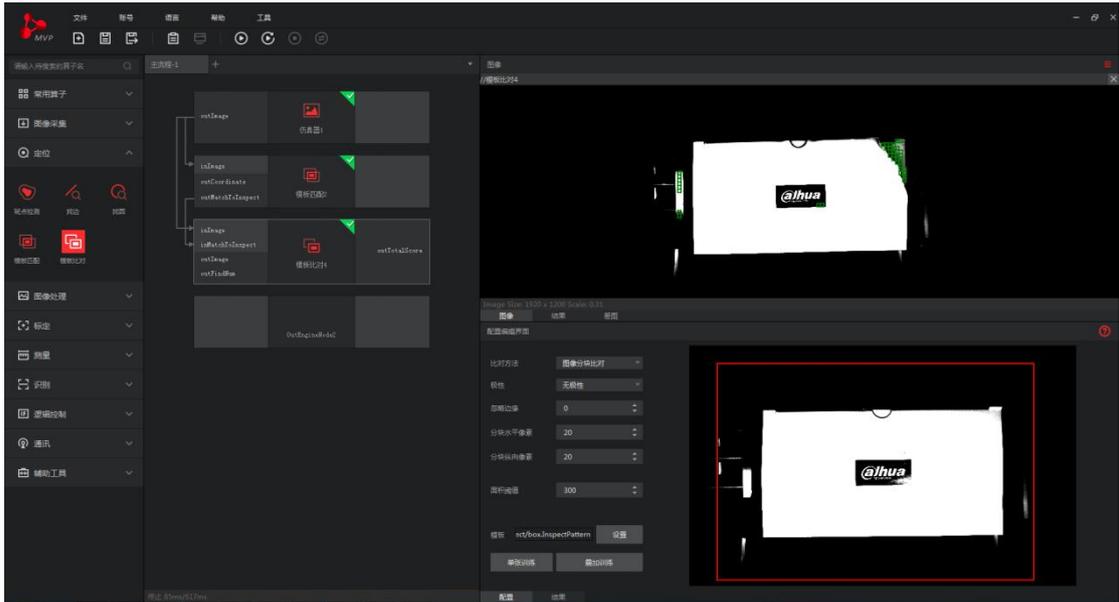


图 4.3.5 模板比对效果

操作流程：



在配置中配置模板文件，然后进行比对模板训练，此步骤前后都可以设置其他配置参数，比对模板训练如下图。



## 配置参数

分组	参数名称	数据类型	取值范围	默认值	说明
无	比对方法 funType	int	[1, 2]	1	选择比对方法, 1 为图像分块比对, 2 为梯度比对
	绝对阈值 absThresh	float	[0, 255]	50	模板比对绝对值差的阈值
	方差阈值 varThresh	float	[0, 100]	3	模板比对方差阈值
	极性 polarity	int	[0, 2]	0	模板比对结果保留较亮区域还是较暗区域, 默认无极性
	忽略边缘 ignoreEdge	int	[0, 1]	0	模板比对时忽略边缘附近的像素大小, 默认 0 时不忽略(目前只支持 0,1 两种状态)
	分块水平像素 xSize	int	(0, 图像宽)	20	分块比对方法时, 分块水平像素大小
	分块纵向像素 ySize	int	(0, 图像高)	20	分块比对方法时, 分块垂直像素大小
	灰度阈值 threshold	int	[0, 255]		全局 blob 时的图像二值化阈值
	面积阈值 areaThresh	int	[0, ∞)		全局 blob 时的区域面积过滤阈值
	分数阈值 scoreThresh	int	[0, 100]		模板比对结果分数阈值, 若小于该阈值则认为检测区域和比对区域相同
	分块分数阈值	int	[0, 100]		分块比对算法时, 每一个块的比对分数阈值

	patchScoreThresh				
	模板 model	.model 文件			存放模板比对中的比对模板信息
	单张训练 singleTrain	int	1		用一张图片作为模板比对的模板
	叠加训练 multiTrain	int	2		用多张图片作为模板比对的模板

## 详细介绍

- 极性：模板比对结果保留较亮区域还是较暗区域，一般默认无极性
- 忽略边缘：模板比对时忽略边缘附近的像素大小，默认 0 时不忽略，目前只支持 0,1 两种状态
- 分块水平像素及分块纵向像素：分块进行比对时，分块的水平或纵向像素大小
- 模板：存放模板比对中的比对模板信息，以后的比对操作都是基于此模板进行，所谓的差异也是与此模板比较的差异



- 注意：**
- ◆ 模板比对一般结合模板匹配一起使用，模板匹配给模板比对提供位置信息，模板比对进行比对检测差异。

## 输出参数

参数名称	数据类型	说明
不合格区域数量 NumPolygon	int	不合格区域数量
匹配分数 Score	float	比对分数
区域 ID	Int	对比差异区域 ID

Area's id		
区域点阵		差别分块区域点阵
Area's points		
outImage		输出图像
outCount		差异数量
OutScore		比对分数

## 4.4 图像处理

### 4.4.1 平滑滤波

平滑滤波即在尽量保留图像细节特征的前提下对目标图像的噪声进行抑制，是图像预处理中不可缺少的操作，其处理效果的好坏将直接影响到后续图像处理和有效性和可靠性。平滑滤波工具可以对灰度图像感兴趣区域进行噪声滤波，生成处理后的灰度图像；平滑滤波类型有：均值滤波，中值滤波及高斯滤波。

三种滤波器的滤波效果如下，其中蓝色矩形框为选中的处理感兴趣区域。

- 均值滤波

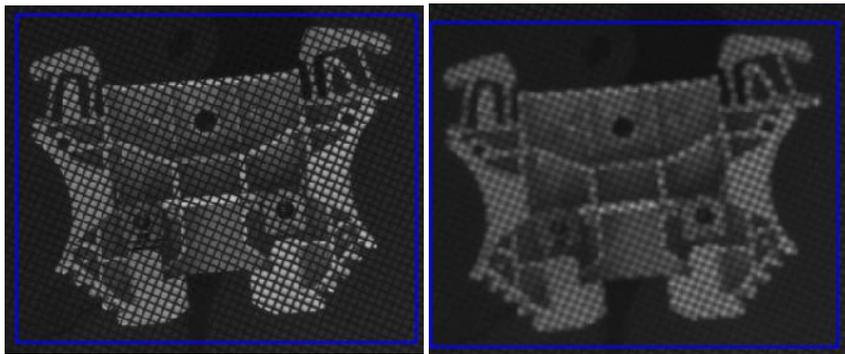


图 4.1.1 3x3 均值滤波效果，左图滤波前，右图滤波后

- 中值滤波

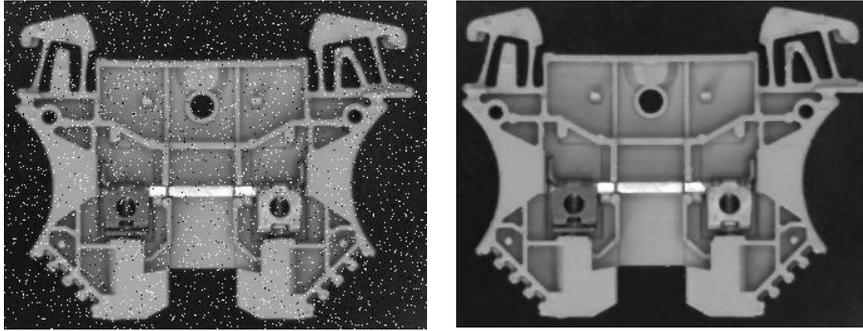


图 4.1.2 3x3 中值滤波效果，左图滤波前，右图滤波后

- 高斯滤波

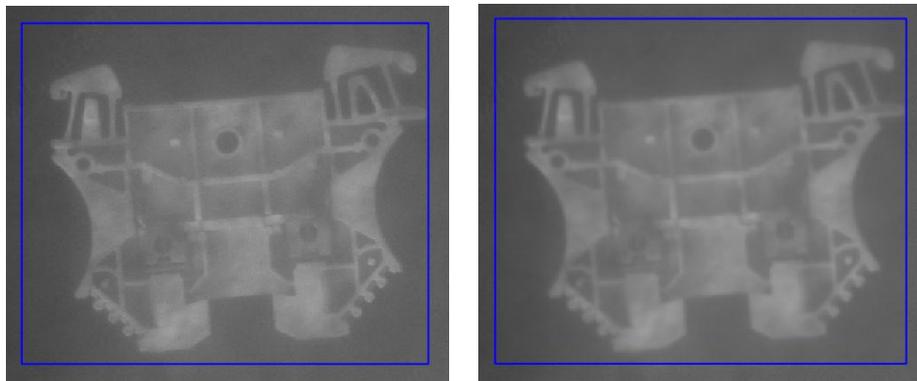
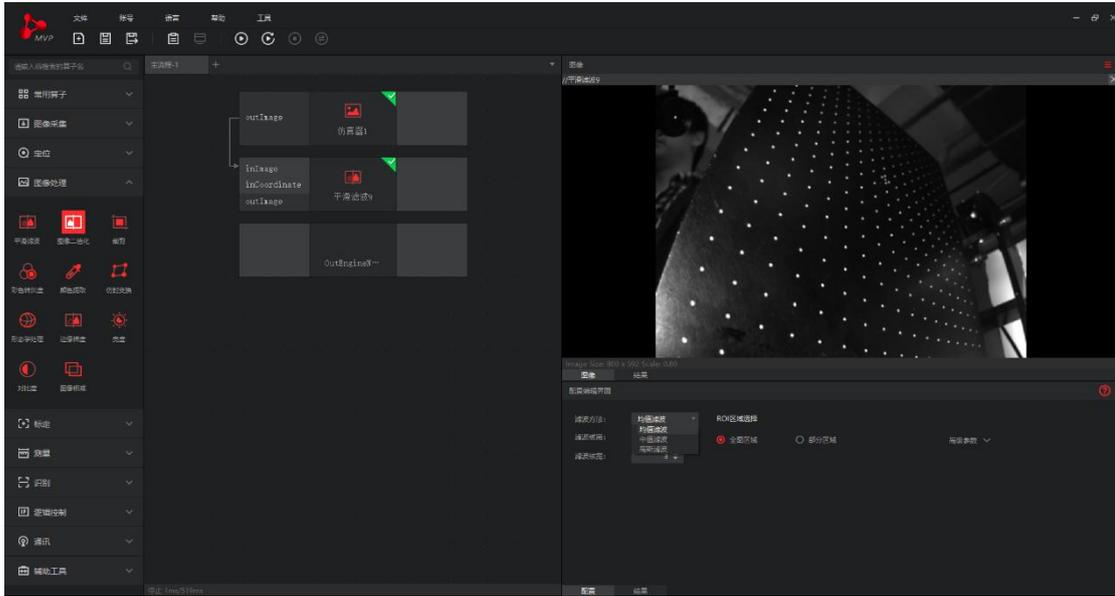


图 4.1.3 3x3 高斯滤波效果，左图滤波前，右图滤波后



## 操作流程

添加平滑滤波算子，输入图片或模板，单击运行按钮，可在右上方图像显示区域看到滤波效果。在右下方区域进行输入参数配置，达到需要的图像滤波效果。

## 配置参数

表 4.1.1 配置参数说明

参数名称	数据类型	取值范围	默认值	说明
滤波方法 Filter Method	int		均值滤波	滤波类型包括均值滤波、中值滤波、高斯滤波
滤波核高 Kernel Height	int	[1, 255]	3	滤波核高 当输入偶数时，算法内部自动+1 处理
滤波核宽 Kernel Width	int	[1, 255]	3	滤波核宽 当输入偶数时，算法内部自动+1 处理

## 详细介绍

- 滤波方法

均值滤波：在局部窗口内，中心点的像素值为邻域内像素的平均值。

中值滤波：在局部窗口内，中心点的像素值为邻域内像素的中值。适用于存在椒盐噪声的情况。

高斯滤波：在局部窗口内，中心点的像素值为邻域内像素的高斯加权值。适用于消除高斯噪声。

- 滤波核高：均值/中值/高斯滤波核的高度，增大滤波核高可使图像纵向平滑程度增加。
- 滤波核宽：均值/中值/高斯滤波核的宽度，增大滤波核宽可使图像横向平滑程度增加。



**注意：**

- ◆ 平滑滤波处理属于图像预处理方法，主要是为了消除图像噪声影响，但是会对图片像素进行改变。因此对于图像高精度测量应注意避免使用平滑处理算子。

## 4.4.2 图像二值化

图像二值化工具通过固定阈值或自适应阈值，将感兴趣区域内的灰度图像转换为二值图像；图像二值化类型有：硬阈值，自动阈值及局部阈值。

硬阈值，自动阈值及局部阈值的二值化效果如下：

- 硬阈值

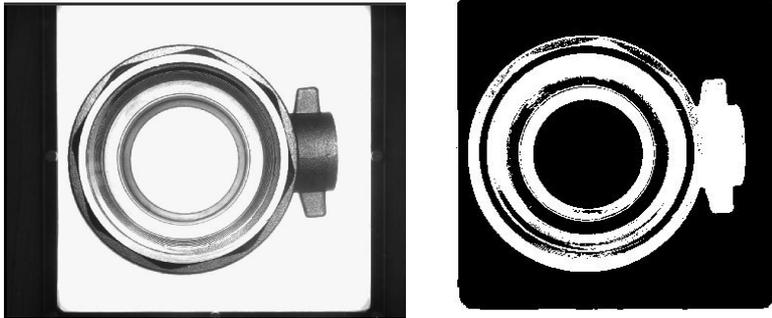


图 4.4.2.1 0（低阈值）—230（高阈值）硬阈值的二值化效果（左图为二值化前，右图为二值化后）

- 自动阈值

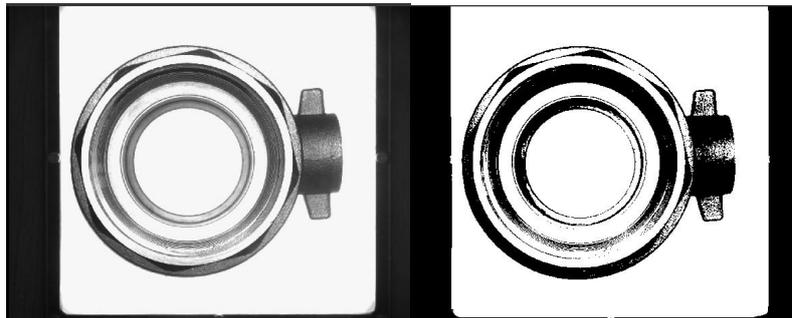


图 4.4.2.2 自动阈值的二值化效果（左图二值化前，右图二值化后）

- 局部阈值

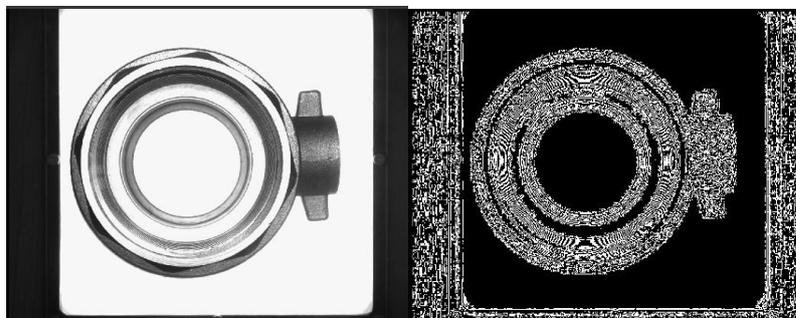
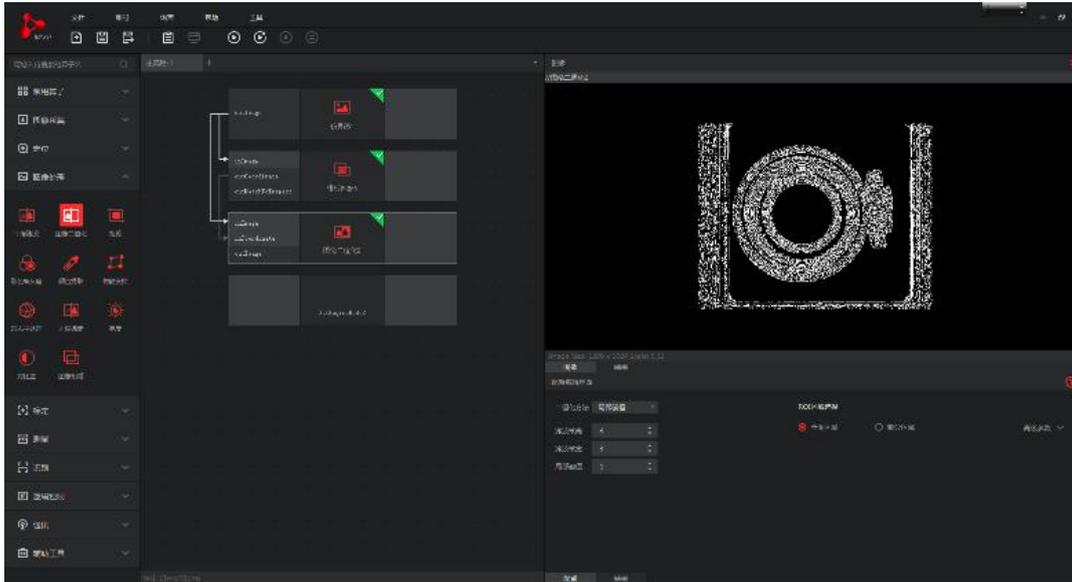


图 4.4.2.3 3x3 局部阈值的二值化效果（左图二值化前，右图二值化后）

## 操作流程



## 配置参数

参数名称	数据类型	取值范围	默认值	说明
二值化方法 Method	单选按钮	自动阈值/ 硬阈值 / 局部阈值	自动阈值	设置二值化时采用的阈值策略
低阈值 Threshold1	int	[0, 255]	128	介于两个阈值之间的像素置 255 ( $\text{thr1} \leq \text{gray} \leq \text{thr2}$ ) ;
高阈值 Threshold2	int	[0, 255]	255	其余置 0

## 详细介绍

### ● 二值化方法

**自动阈值：**设置自动阈值时，算法将自动确定分割的阈值，高于阈值的像素在二值图中设置为白色，低于阈值的像素在二值图中设置为黑色。

**硬阈值：**设置为硬阈值二值化时，输出灰度范围在低阈值和高阈值之间的值，对应输出的二值图像像素点为白色，否则为黑色。

局部阈值：根据像素点局部区域灰度值计算灰度均值，当像素值大于灰度均值与局部阈值参数之和，对应输出的二值图像像素点为白色，否则为黑色。

- **低阈值**：以此为分割的阈值设定的最小阈值。
- **高阈值**：以此为分割的阈值设定的最大阈值。

## 4.4.3 裁剪

裁剪工具可以截取指定的原灰度矩形 ROI 区域，并生成 ROI 大小的新的灰度图像；

裁剪效果图如下：

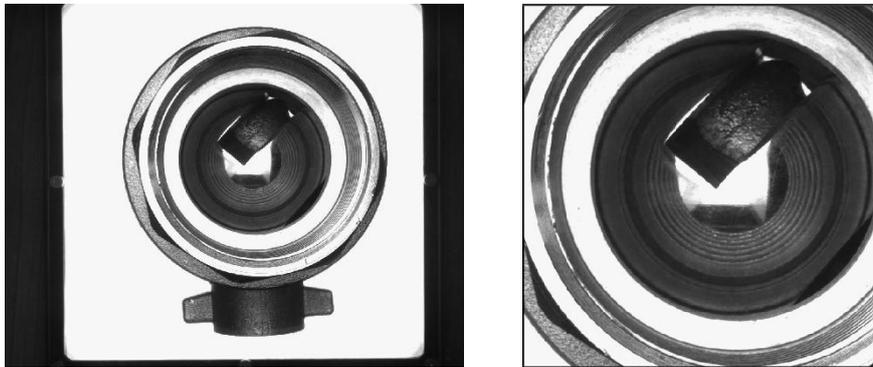


图 4.4.3.1 裁剪效果，左图裁剪前，右图裁剪后

## 4.4.4 彩色转灰度

彩色转灰度工具可将三通道的彩色图像转换为单通道的灰度图像；

效果图如下：



## 配置参数

参数名称	数据类型	取值范围	默认值	说明
通道选择 Channel Select	单选按钮	三通道 / 红色通道 / 绿色通道 / 蓝色通道	三通道	可选择 RGB 中的某一单独通道，也可选择三通道（即通过三通道权重混合后的灰度图像）

## 详细介绍

通道选择：通道模式下转换公式如下  $0.299r + 0.587g + 0.114b$ 。r 为 R 通道灰度值，g 为 G 通道灰度值，b 为 B 通道灰度值。

### 4.4.5 颜色提取

颜色提取算子首先提取框选区域的颜色，训练出对应颜色模板，可训练多种颜色，然后输入当前颜色，计算当前颜色与颜色模板的相似度，用于颜色匹配及颜色识别的项目。

例如识别人民币面额，因为各种面额人民币颜色不同，可用该算子区分，首先训练各个面额人民币的颜色模板，当当前图像是 100 面额时，算法会判断出当前图像与各个模板的匹配分数，选择最大匹配分数即为当前面额。

	匹配颜色	匹配分数
1	100	0.8999
2	20	0.702722
3	1	0.673672
4	50	0.635413
5	5	0.510613
6	10	0.412568

图 4.5.1 人民币识别

## 操作流程

添加仿真器，添加颜色提取，选择仿真器胶片，仿真器和颜色提取算子进行连线，设置颜色提取算子模型文件，如下：

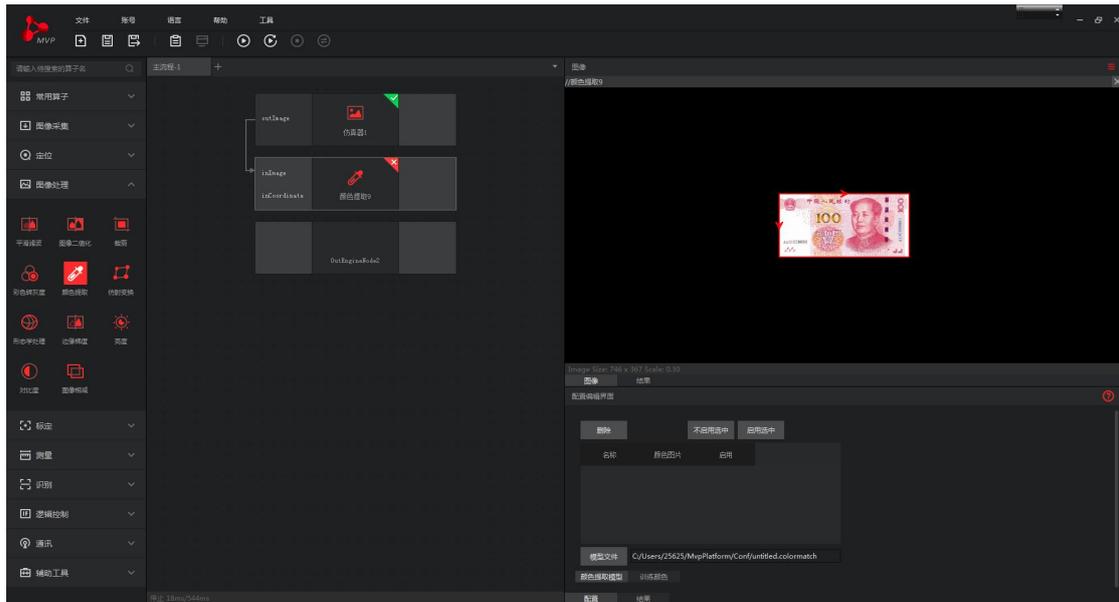


图 4.4.5.2 颜色提取操作图

切换到训练界面，选择部分区域，展开“选择形状”，选择形状，然后点击训练，可以进行若干次训练，最大支持 32 张训练图片，如下：

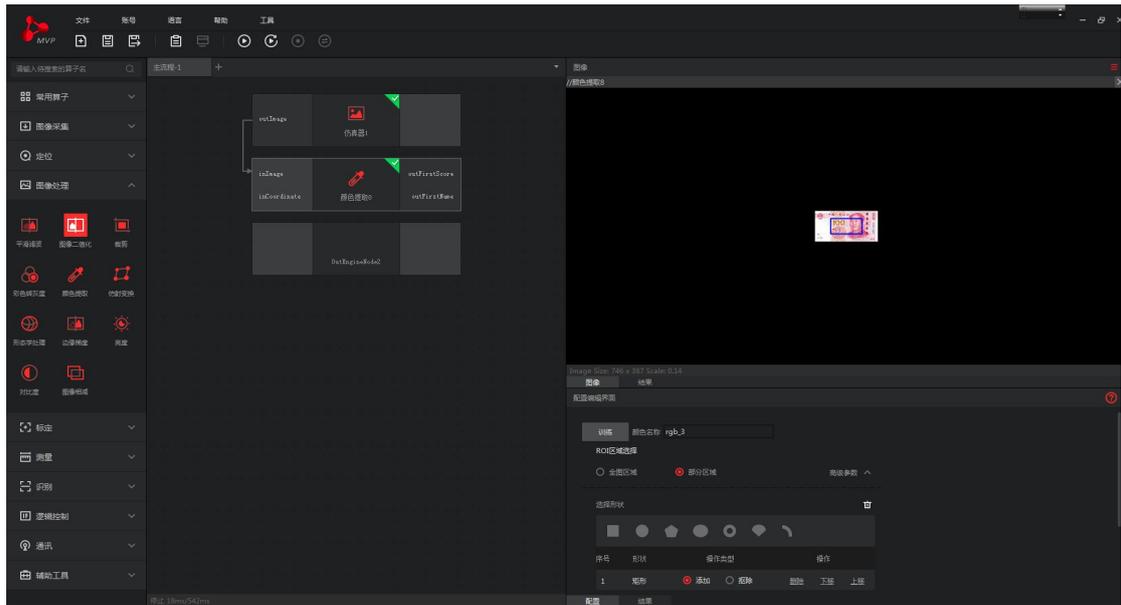


图 4.4.5.3 颜色提取训练界面

训练成功后切换到配置界面，可以观看已训练的图片，如下：

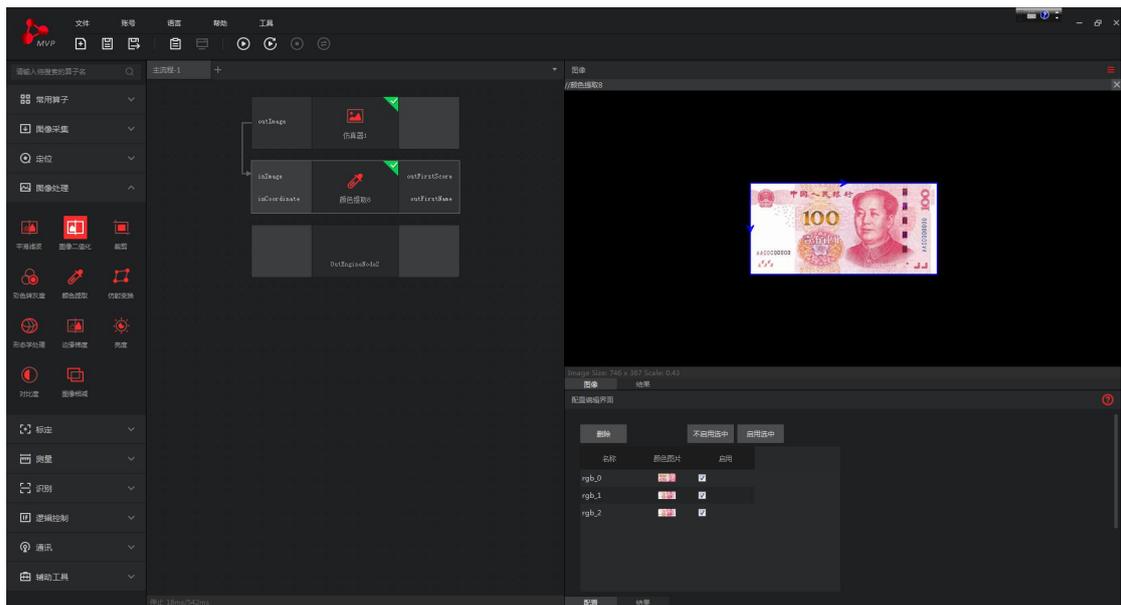


图 4.4.5.4 颜色提取配置界面

## 配置参数

参数名称	数据类型	取值范围	默认值	说明
训练颜色				训练颜色

Train Color				
模型文件 Model				选择相应的模型文件

## 详细介绍

- **训练颜色：**当用户框选待训练的区域后，计算其颜色直方图作为该区域的颜色模版，并保存在模型文件中。
- **模型文件：**当用户框选待匹配区域后，计算该区域的颜色直方图与模型文件中的模版进行对比得到匹配分数最高的区域。

## 输出参数

参数名称	数据类型	说明
匹配颜色 Match Color		罗列颜色模板中的各个颜色
匹配分数 Match Score	float	对应匹配颜色的匹配度

### 4.4.6 仿射变换

仿射变换工具可以将输入图像矩形框区域内的图像进行变换，并生成新的矩形输出图像。输入矩形支持平移，旋转，倾斜，XY比例缩放。仿射变换的典型应用场景包括但不限于以下：

如果后续的图像处理操作对位置和角度有较高的要求，可以通过仿射变换将输入图像调整到合适的位置和角度。如下图所示：

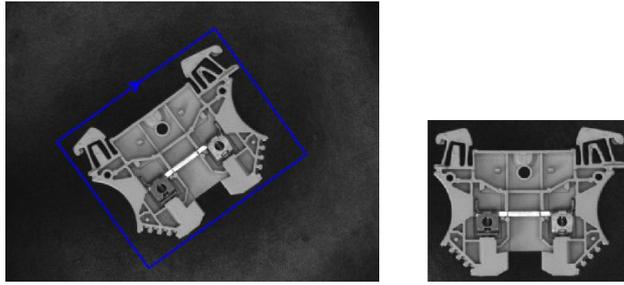


图 4.6.1. 利用仿射变换调整输入图像位置和角度

也可以通过图像仿射变换对输入图像局部感兴趣区域进行截取并放大：

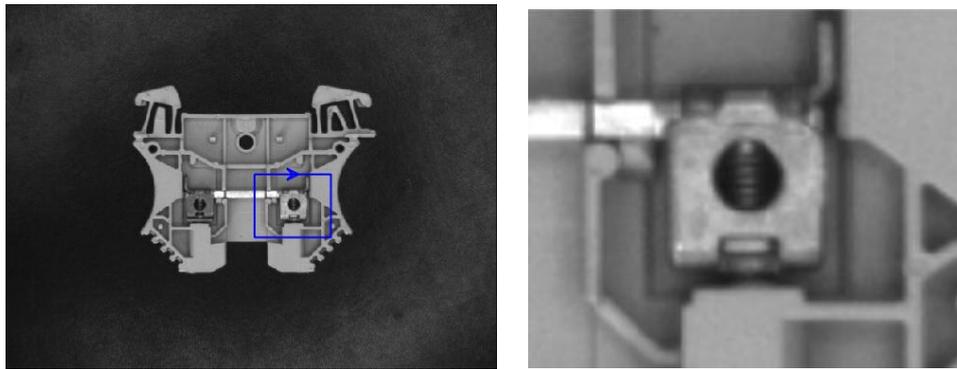


图 4.6.2. 利用仿射变换截取并放大局部

通过仿射变换将倾斜的图像校正回无倾斜的图像：

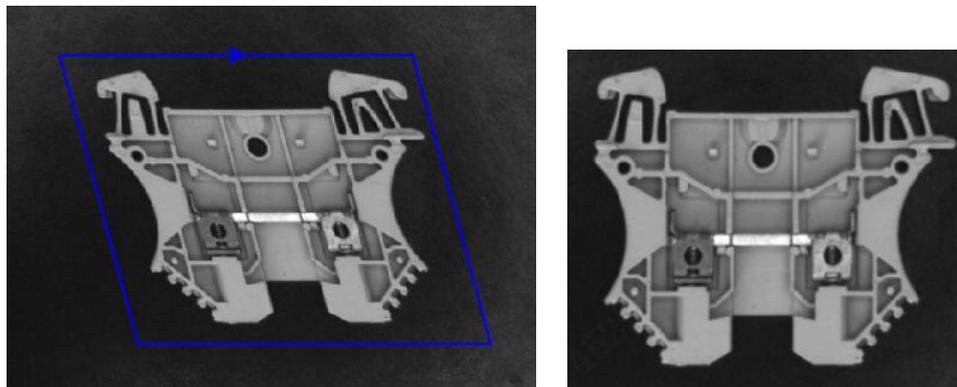


图 4.6.3. 利用仿射变换校正倾斜图像

## 配置参数

参数名称	数据类型	取值范围	默认值	说明
中心 X(Center X)	float			矩形区域中心 X 坐标

中心 Y(Center Y)	float			矩形区域中心 Y 坐标
边长 X(Length X)	Float			矩形区域边长 X
边长 Y(Length Y)	float			矩形区域边长 Y
旋转(rotation)	float	[-360,360]	0	矩形区域绕中心旋转角度
倾斜(slant)	float	[-360,360]	0	矩形区域绕 x 轴旋转
缩放 X	float		1	矩形区域 X 方向缩放比例
缩放 Y	float		1	矩形区域 Y 方向缩放比例
采样模式 (Interpolation)	float	最近邻 双线性 双三次	双线性	图像插值方式

## 详细介绍

- 采样模式会影响算法精度和运行速度。可选方法有：最近邻、双线性、双三次，对应的算法精度由低到高，算法运行速度由快到慢。
- 支持的变换类型效果如下图所示（灰色为变换前，蓝色为变换后），分别为：X 方向平移，Y 方向平移，旋转，XY 同比例缩放，X 缩放，Y 缩放，倾斜：

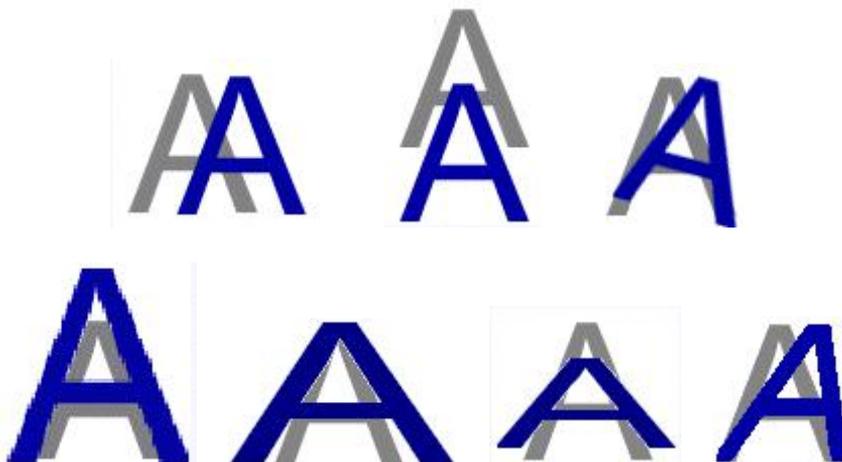


图 4.6.4. 仿射变换效果图

#### 4.4.7 形态学处理

形态学处理工具可以根据设定的掩模形状和形态学方法对输入灰度图像进行形态学操作；形态学方法包括：膨胀、腐蚀、开运算、闭运算；支持的掩模形状包括：矩形、圆形、菱形。

膨胀可以将白色区域扩散（黑色区域缩小），例如将白色点阵字符连通在一起，如下图所示：

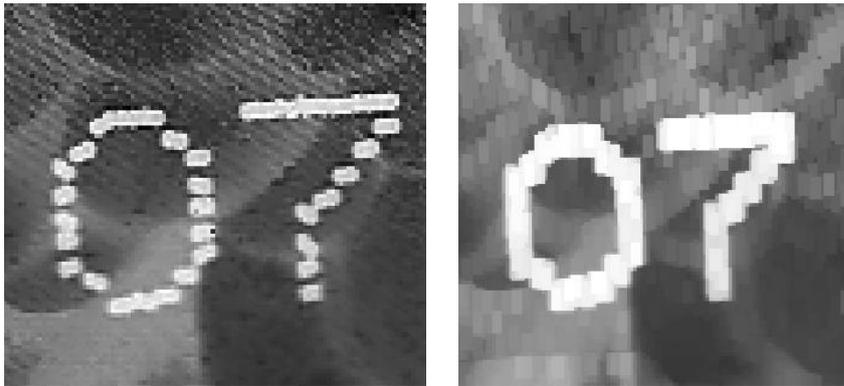


图 4.7.1 膨胀效果，左图为原图，右图为膨胀后的效果

腐蚀可以将黑色区域扩散（白色区域缩小），如下图所示：

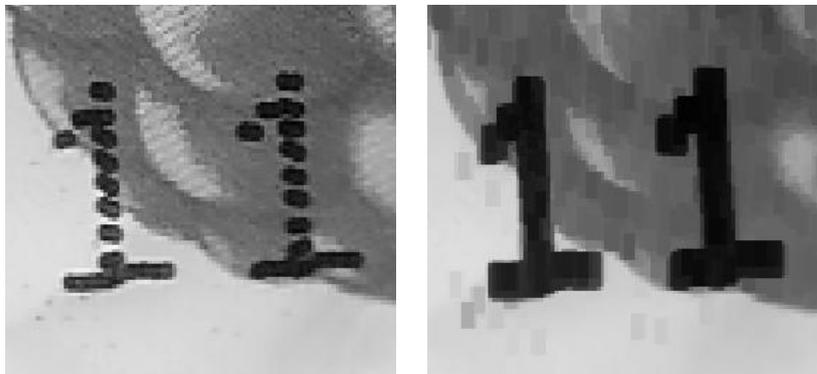


图 4.7.2 腐蚀效果，左图为原图，右图为膨胀后的效果

闭运算可以将白色区域的断裂部分连通（黑色连通部分断开），同时又不影响本身的大小：

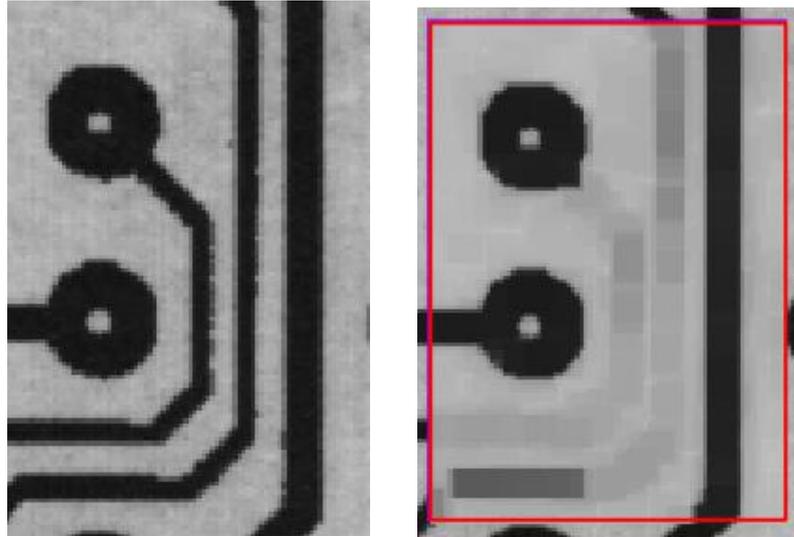


图 4.7.3 闭运算效果，左图为原图，右图为闭运算结果

## 配置参数

参数名称	数据类型	取值范围	默认值	说明
形态学方法 Filter Method	单选按钮	膨胀/腐蚀/ 开操作/闭 操作	膨胀	
滤波核形状 Kernel Shape	单选按钮	矩形/圆形/ 菱形	矩形	菱形形状下，滤波核宽高要求相等
滤波核高 Kernel Height	int	[1, 91]	3	滤波核高 当输入偶数时，算法内部自动+1 处理
滤波核宽 Kernel Width	int	[1, 91]	3	滤波核宽 当输入偶数时，算法内部自动+1 处理

## 详细介绍

### ● 形态学方法

膨胀：分割独立的图像元素，连接相邻的元素；

腐蚀：膨胀的对偶操作；

开操作：先腐蚀后膨胀；

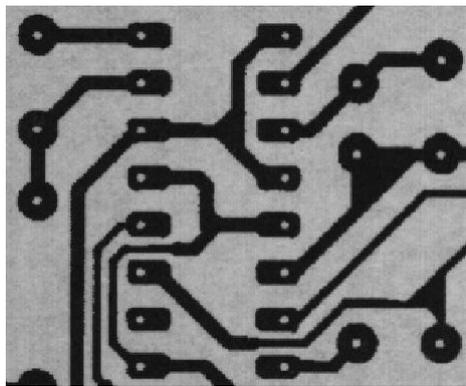
闭操作：先膨胀后腐蚀。

- **滤波核形状**：形态学操作掩膜的形状，目前支持矩形、圆形和菱形。
- **滤波核高**：邻域核高度。
- **滤波核宽**：邻域核宽度。

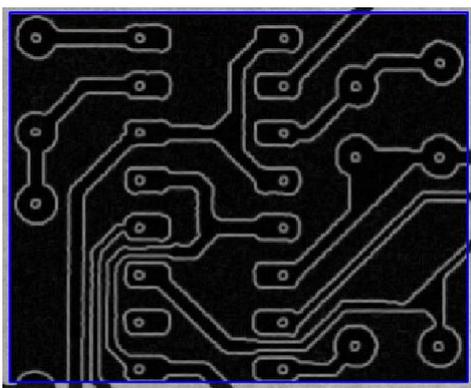
## 4.4.8 边缘梯度

边缘梯度工具可以计算灰度图像的边缘梯度，并根据梯度幅值生成新的灰度图像。

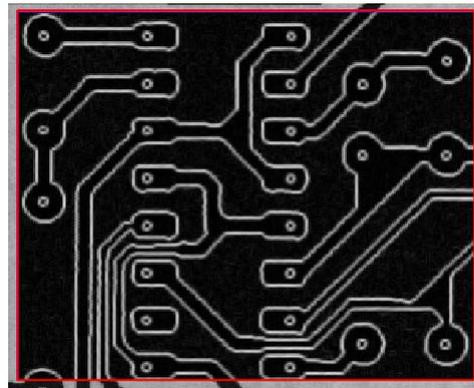
通过边缘梯度计算后得到的图像，将关注点集中在边缘区域，边缘后续的处理，以下分别展示了不同边缘梯度算法的效果：



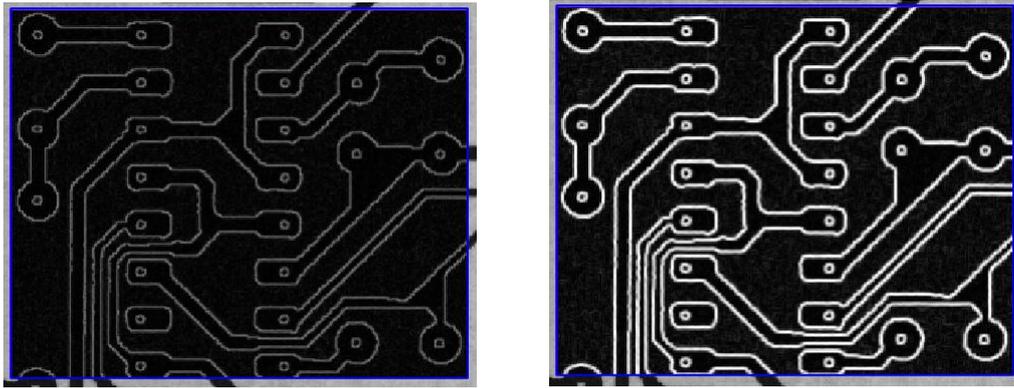
(a)



(b)



(c)



(d)

(e)

图 4.8.1. 边缘梯度效果图：(a)原图. (b)Sobel. (c)Prewitt.(d)Roberts. (e)Robinson.

### 配置参数

参数名称	数据类型	取值范围	默认值	说明
边缘梯度方法 Filter Method	单选按钮	Sobel/ Prewitt/ Roberts/ Robinson	Sobel	设置计算边缘梯度算法
Sobel 类型 Sobel Type	单选按钮	绝对值/ 开平方		设置不同方向梯度累积形式
Roberts 类型 Roberts Type	单选按钮	最大/梯 度最大/ 梯度和		设置不同方向梯度累积形式

### 详细介绍

- **边缘梯度方法**：分为 sobel、Prewitt、Roberts、Robinson 等计算方式。下图分别为 sobel 和 Prewitt 算子的计算模版。

-1	0	1
-2	0	2
-1	0	1

(a) Gx

1	2	1
0	0	0
-1	-2	-1

(b) Gy

## Sobel 算子

-1	0	1
-1	0	1
-1	0	1

(a) Gx

1	1	1
0	0	0
-1	-1	-1

(b) Gy

## Prewitt 算子

- **Sobel 类型：**不同方向梯度累积形式，分别为绝对值、开平方，公式如下

绝对值： $G = |Gx| + |Gy|$

开平方： $G = \sqrt{(Gx)^2 + (Gy)^2}$

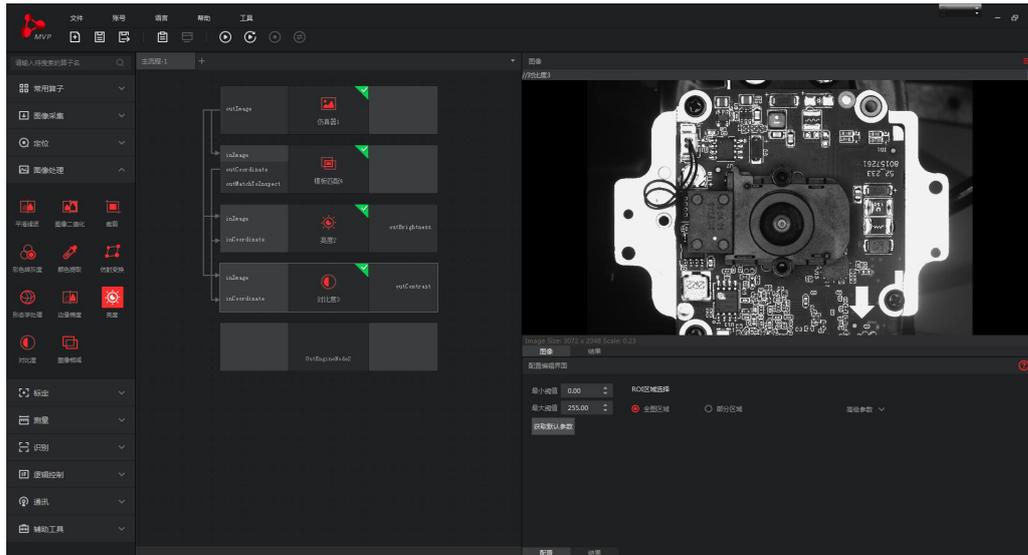
- **Roberts 类型：**最大、梯度最大、梯度和。

## 4.4.9 亮度

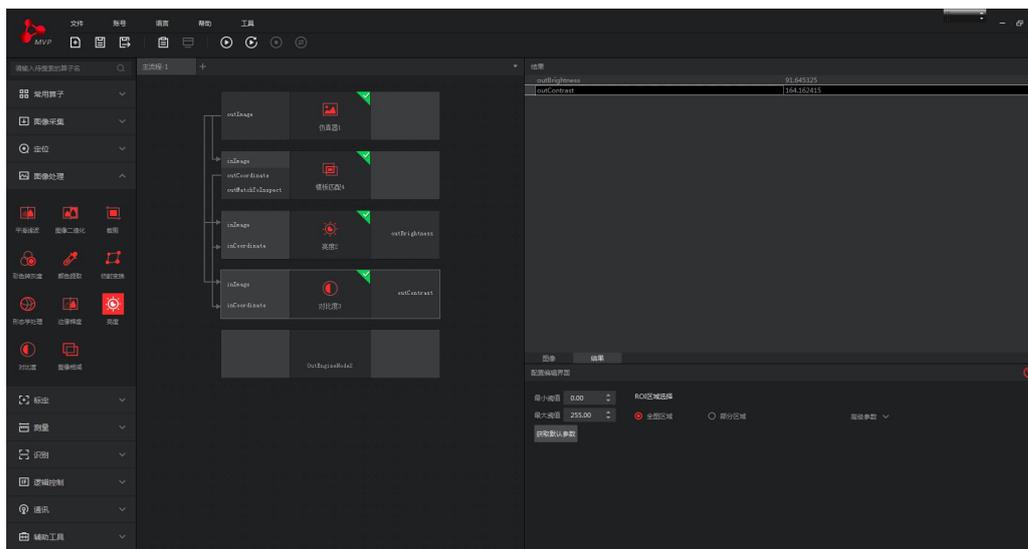
亮度算子计算掩膜区域中的平均亮度值，根据输出亮度值是否介于阈值区间内，判断被检测物体是否存在。

特征亮度的变化是对所检测物体良好与否的体现，该工具在产品亮度差异检测上非常有用。

## 操作流程



亮度输出结果如下：



## 配置参数

参数名称	数据类型	取值范围	默认值	说明
最小阈值 Min Threshold	float	[0, 255]	0	阈值下限
最大阈值 Max Threshold	float	[0, 255]	255	阈值上线
获取默认阈值	按钮			根据当前输出的平均

Get Default				亮度, 自动设定阈值
-------------	--	--	--	------------

## 详细介绍

- **最小阈值:** 阈值下限。
- **最大阈值:** 阈值上限。
- **获取默认阈值:** 根据输出的平均亮度, 自动获取阈值。

## 输出参数

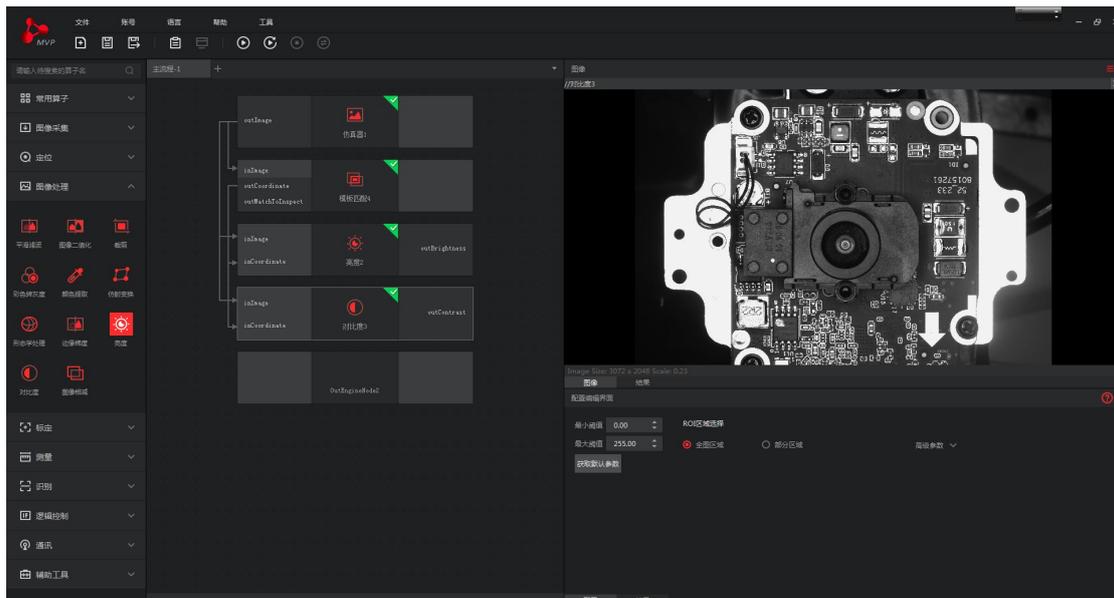
参数名称	数据类型	说明
亮度 Luminance	float	输出当前 ROI 的平均亮度

## 4.4.10 对比度

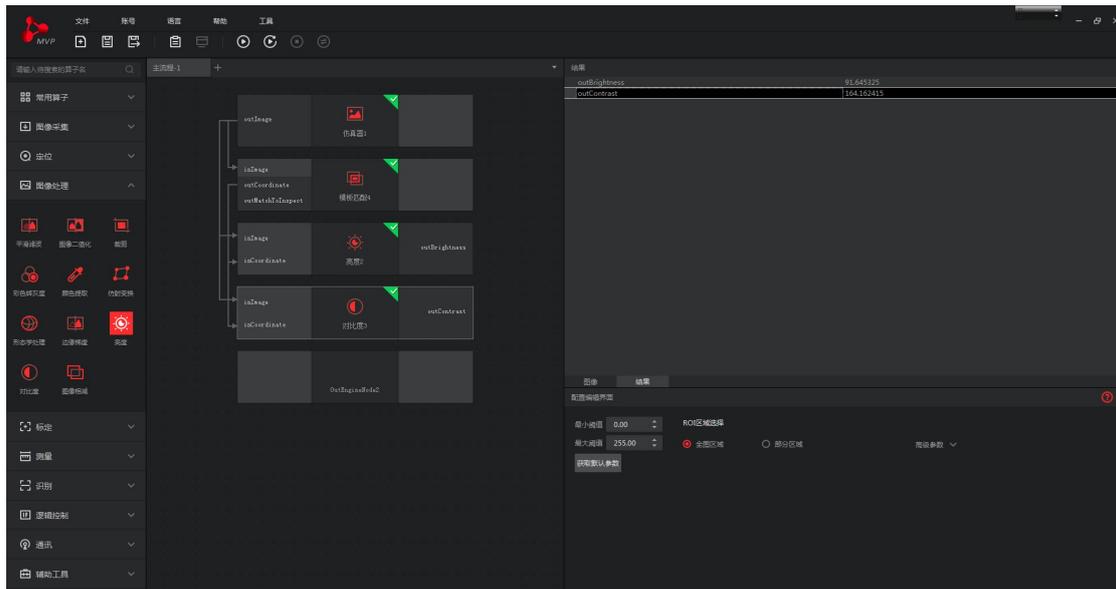
对比度算子对掩膜区域内图像对比度的计算, 甄别所需检测特征是否在设置的对比度范围以内, 以此配合进行相应工业生产的检测判断。

该工具在良品部件与次品部件的对比度相差明显的情况下非常有用。例如, 验证白色标签上是否存在暗点。

### 操作流程:



对比度的输出结果如下：



## 配置参数

参数名称	数据类型	取值范围	默认值	说明
最小阈值 Min Threshold	float	[0, 255]	0	阈值下限
最大阈值 Max Threshold	float	[0, 255]	255	阈值上线
获取默认阈值 Get Default	按钮			根据当前输出的对比度，自动设定阈值

## 详细介绍

- 最小阈值：阈值下限。
- 最大阈值：阈值上限。
- 获取默认阈值：根据输出的平均亮度，自动获取阈值。

## 输出参数

参数名称	数据类型	说明
对比度	float	输出当前 ROI 的对比度

Contrast

## 4.4.11 图像相减

图像相减的原理是对两张图像相同坐标像素的灰度值进行相减然后得到新的图像。

图像相减效果图如下：

图像相减输入图像：



图 4.4.11.1 图像相减输入图像，左图为第一张输入，右图为第二张输入

图像相减输出效果图：

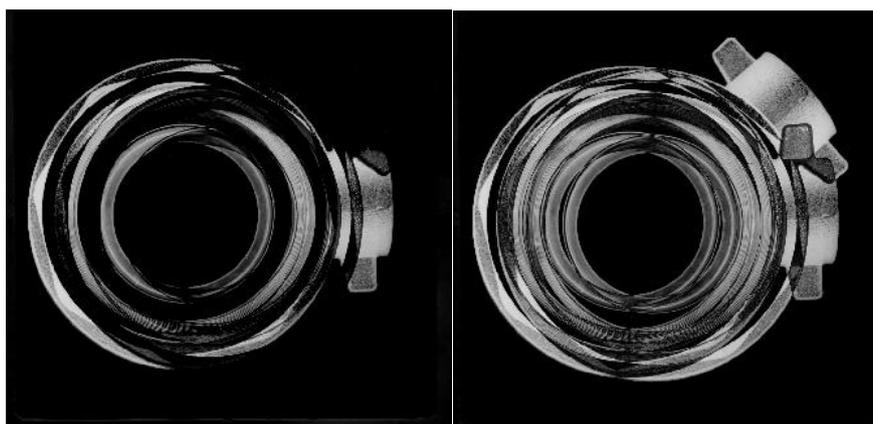


图 4.4.11.2 图像相减结果图：左图为截取负数，右图为绝对值

## 操作流程：

添加两个仿真器，添加图像相减算子，按如下图进行连线，选择算子，运行。

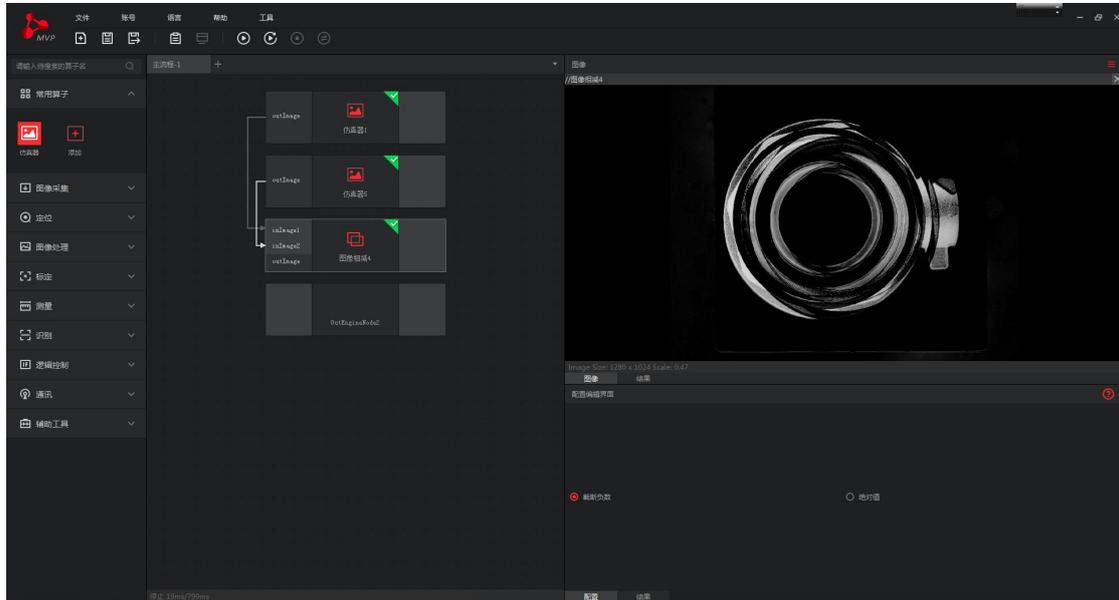


图 4.4.11.3 图像相减操作图

## 配置参数

参数名称	数据类型	取值范围	默认值	说明
计算模式 Computer Mode	单选按钮	截断负数/绝对值	截断负数	

## 详细介绍

### ● 计算模式

截断负数：当相减后的像素值出现负数，则将其置 0。

绝对值：对相减后的像素值取绝对值。

## 输出参数

参数名称	数据类型	取值范围	默认值	说明
计算模式	单选按钮	截断负数/绝对值	截断负数	

## 4.5 标定

### 4.5.1 棋盘格标定

棋盘格标定算子用于建立棋盘格所在平面和图像平面的关系，计算图像坐标映射到世界坐标的透视变换，输出标定文件。

如图 4.5.1 所示，利用棋盘格标定进行标定：

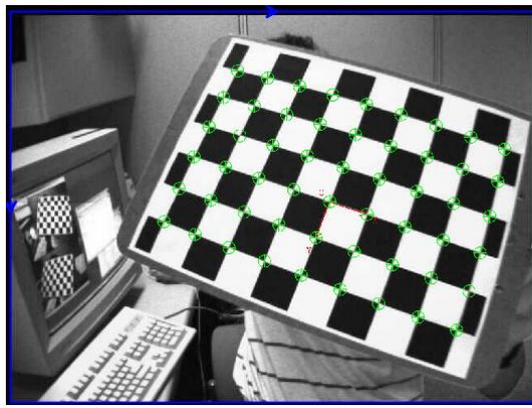
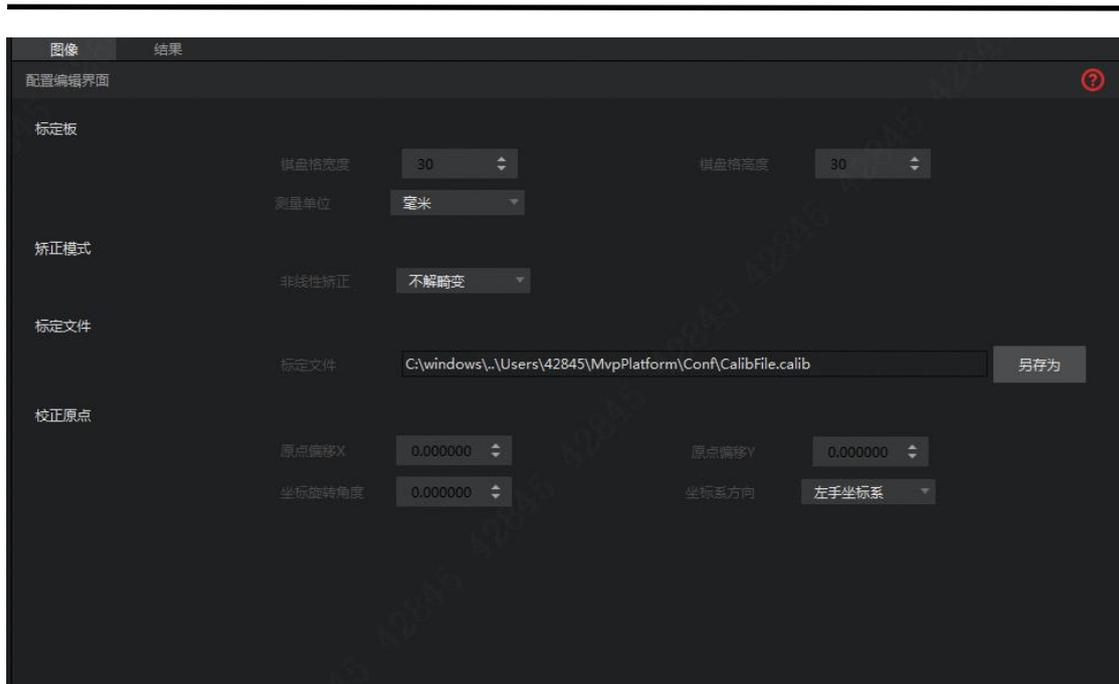


图 4.5.1 棋盘格标定

### 操作流程

输入棋盘格的宽度和高度，设置测量单位，选择是否解畸变，并设置好文件路径和矫正原点位置点击运行即可。



## 配置参数

表 4.5.1 配置参数说明

分组	参数名称	数据类型	取值范围	默认值	说明
标定板	测量长度单位 Length Unit		无单位/厘米/毫米/微米	无单位	物理单位选取
	棋盘格宽度 Tile Size X	int	[1, 999999]	30	棋盘格单个格子宽度
	棋盘格高度 Tile Size Y	int	[1, 999999]	30	棋盘格单个格子高度
矫正模式	非线性矫正 Distortion		解畸变/不解畸变	不解畸变	对于镜头畸变较严重的相机，开启非线性矫正功能会使得标定更准确，体现在标定结果中的 RMS（越小代表标定越准确）
标定文件	标定文件				标定结果保存于该文件
校正原点	原点偏移 X	float	[-999999, 999999]	0	原点 x 方向偏移的物理量

Origin X		999999]		
原点偏移 Y	float	[-999999, 999999]	0	原点 y 方向偏移的物理量
Origin Y				
坐标旋转角度	float	[-360,360)	0	坐标系旋转角度
Coordinate Rotation				
坐标系方向		左手坐标系/右手坐标系	左手坐标系	左右手坐标系
Handedness				

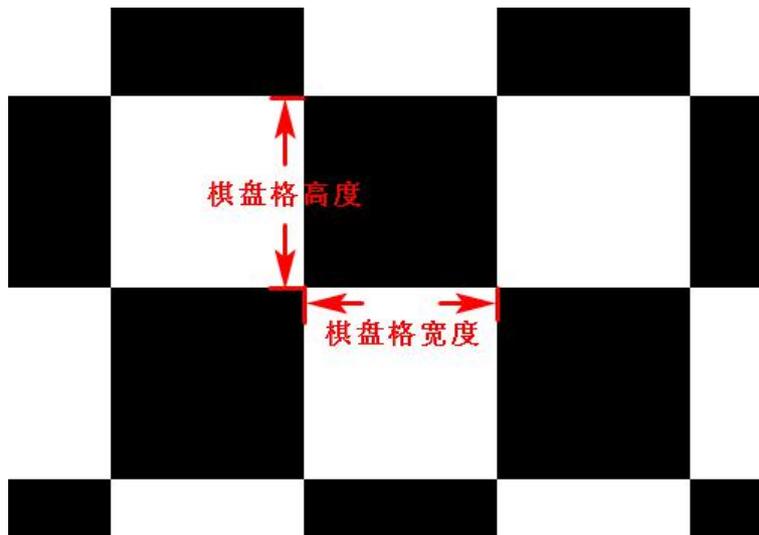
## 详细介绍

- 标定板

测量长度单位：标定板的实际物理尺寸单位

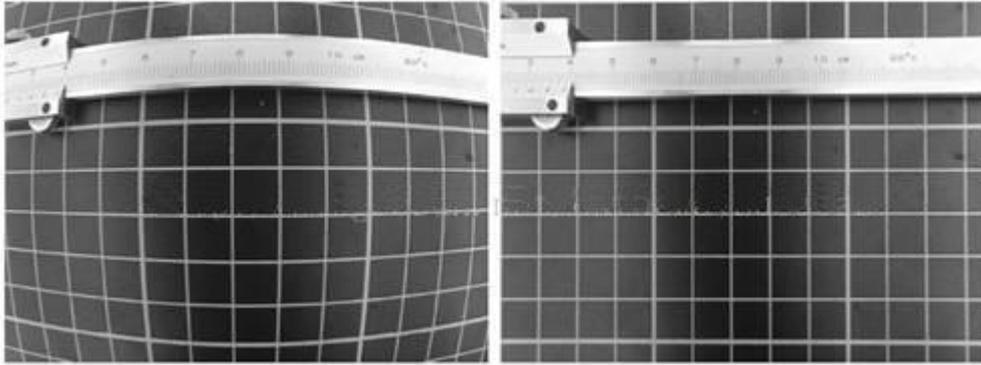
棋盘格宽度：棋盘格单个格子宽度

棋盘格高度：棋盘格单个格子高度



- 矫正模式

非线性矫正：由于透镜先天条件原因（透镜形状），会造成图像围绕光学中心产生径向畸变，随着向边缘移动，畸变越厉害，开启非线性矫正可抵消这部分畸变带来的精度损失。通常情况下若图像无畸变，可关闭该功能，设置不解畸变。



- 标定文件

标定文件：用于存放标定结果，通常存放于默认路径即可，若有需要用户可自行配置其他路径。

- 校正原点

运行棋盘格标定算法后，算法会给出一个内部计算的原点，若返回的原点跟预期不一致，可通过以下 3 个参数来校正。

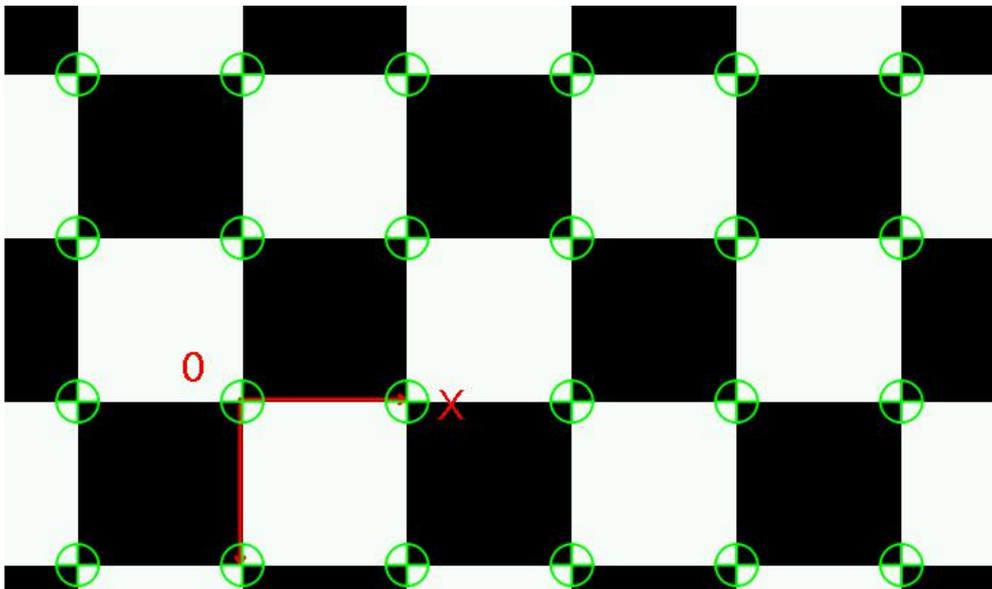
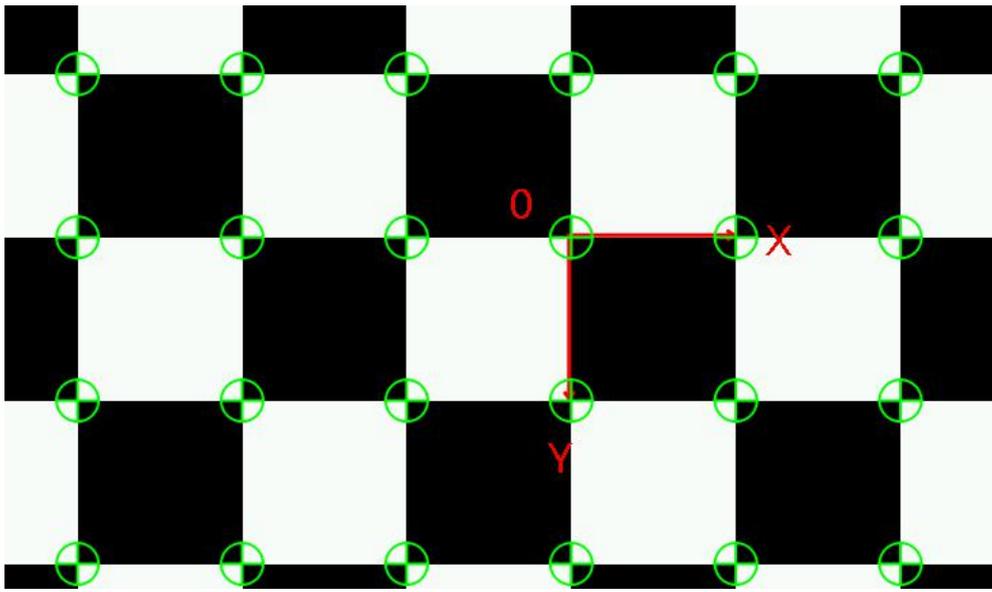
原点偏移 X：原点位置沿 x 轴偏移的实际尺寸

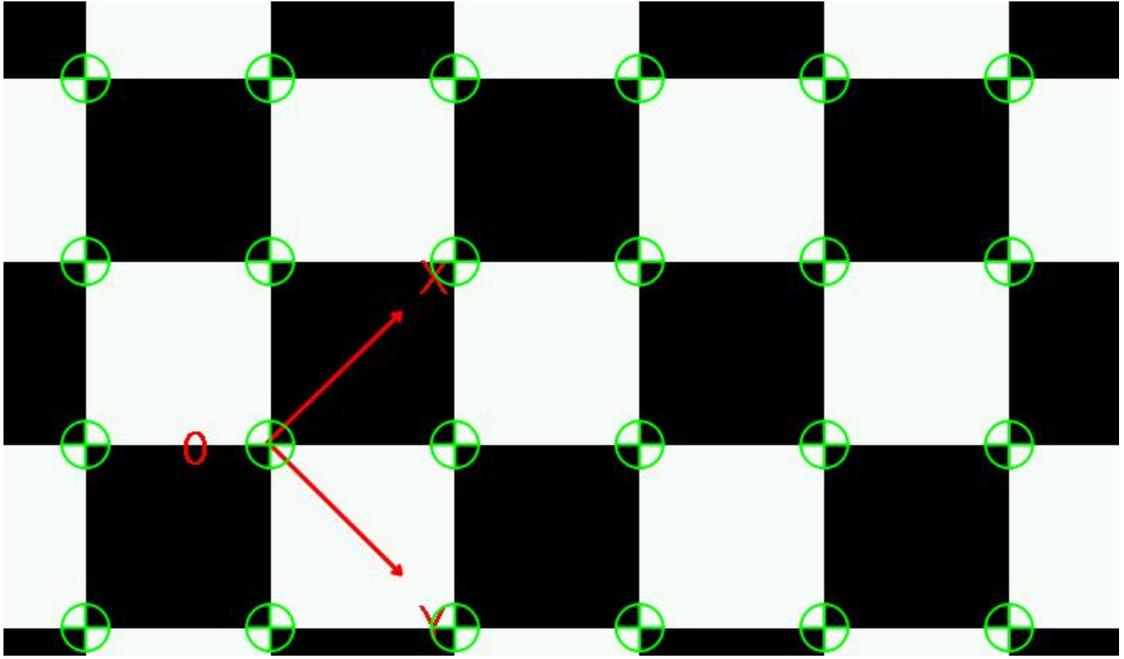
原点偏移 Y：原点位置沿 y 轴偏移的实际尺寸

坐标旋转角度：原点逆时针旋转的角度

以下 3 图分别为（红色坐标系）：

- ① 初始原点
- ② 原点偏移 X 配置 -60；原点偏移 Y 配置 30（棋盘格的宽高配置都是 30）
- ③ 在②的基础上坐标旋转角度配置  $45^\circ$



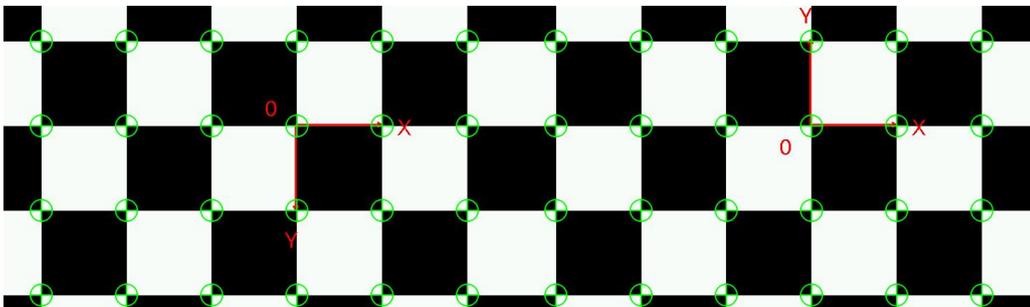


坐标系方向：

左手坐标系是依据图像坐标系，x 轴向右，y 轴向下；

右手坐标是常规的 x 轴向右，y 周向上。

以下示例左图为左手坐标系，右图为右手坐标系



 注意：

- ◆ 棋盘格标定算法只支持特制标准棋盘格标定板，不支持圆点标定板，拍摄标定板时尽量让整个标定板出现在相机视野，并覆盖视野 3/4 以上。

## 输出参数列表

参数名称	数据类型	取值范围	默认	说明
------	------	------	----	----

			值	
原点 X OriginPointX	Float	--	--	坐标系原点的 X 值
原点 Y OriginPointY	Float	--	--	坐标系原点的 Y 值
OutKC 畸变系数	Float	--	--	相机镜头的畸变
outRMS RMS 误差	Float	--	--	均方根误差

## 4.5.2 N 点标定

N 点标定算子根据输入或配置的 N 组坐标点对（图像—世界），计算将图像坐标映射到世界坐标的透视变换，并将图像的像素坐标转换到世界坐标，输出标定文件。

如图 4.5.2 所示，利用 N 点标定进行标定。

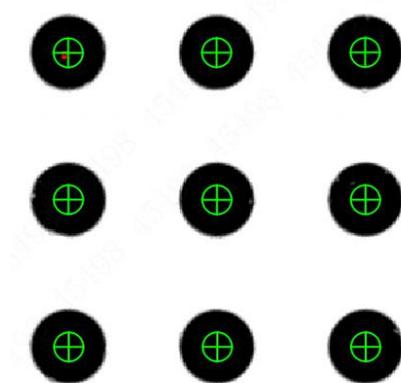


图 5.4.1 N 点标定

## 操作流程

添加图像坐标和对应的世界坐标设置标定文件的路径。



## 配置参数

表 4.5.2 配置参数说明

参数名称	数据类型	取值范围	默认值	说明
测量长度单位 Length Unit		无单位/厘米/ 毫米/微米	无单位	物理单位选取
图像坐标 X Image X	float[]	[-999999, 999999]	有输入时直接显示 输入的坐标值, 无输 入时默认 0	点序列的图像 X 坐标, 可 通过输入参数获取, 也可 手动配置
图像坐标 Y	float[]	[-999999,	有输入时直接显示	点序列的图像 Y 坐标, 可

Image X		999999]	输入的坐标值，无输入时默认 0	通过输入参数获取，也可手动配置
世界坐标 X World X	float[]	[-999999, 999999]	0	点序列的世界 X 坐标，由用户手动配置
世界坐标 Y World Y	float[]	[-999999, 999999]	0	点序列的世界 Y 坐标，由用户手动配置

## 输出参数列表

参数名称	数据类型	取值范围	默认值	说明
RMS RMS 误差	HomMat	--	--	均方根误差

## 4.5.3 读取标定文件

读取标定文件算子根据配置选择的标定文件（棋盘标定算子等算子生产的），将其解析出标定信息并输出。

## 输出参数列表

参数名称	数据类型	取值范围	默认值	说明
输出 H 矩阵 outHomMat	HomMat	--	--	输出读取到的 H 矩阵

- H 矩阵：该数据为 3\*3 的 H 矩阵。其中 outHomMat 由 9 个 float 数字排列成 3\*3 的矩阵。

结果	
outHomMat	
mat[0]	0.784169
mat[1]	0.196563
mat[2]	-206.747940
mat[3]	-0.159819
mat[4]	0.801531
mat[5]	-137.586884
mat[6]	-0.000098
mat[7]	-0.000142
mat[8]	1.050895

其组织方式如图\*\*编号\*\*所示

```
mat[0] mat[1] mat[2]
mat[3] mat[4] mat[5]
mat[6] mat[7] mat[8]
```

## 4.6 测量

测量类算子设计到标定坐标时，图像坐标只用于图像显示，其他值均已为世界坐标上相关联。所有元素（点、线段、距离等）都为世界坐标关联的值。转换关系详见 4.5 标定类算子。

### 4.6.1 两点生成直线

两点生成直线算子从输入或者配置接收两点参数，根据两点生成一条直线并输出，如果输出两点坐标相同则算子失败。可切换到结果界面查看输出生成的直线描述信息（输入两点的中点和生成的直线与图像坐标系的夹角）如图 4.6.1 所示。

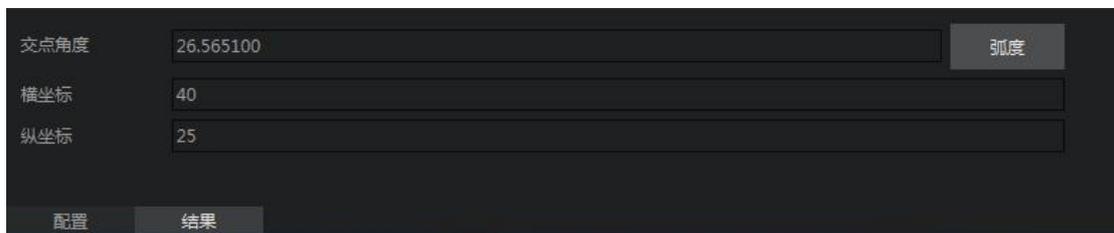


图 4.6.1 两点生成直线结果图

操作步骤：

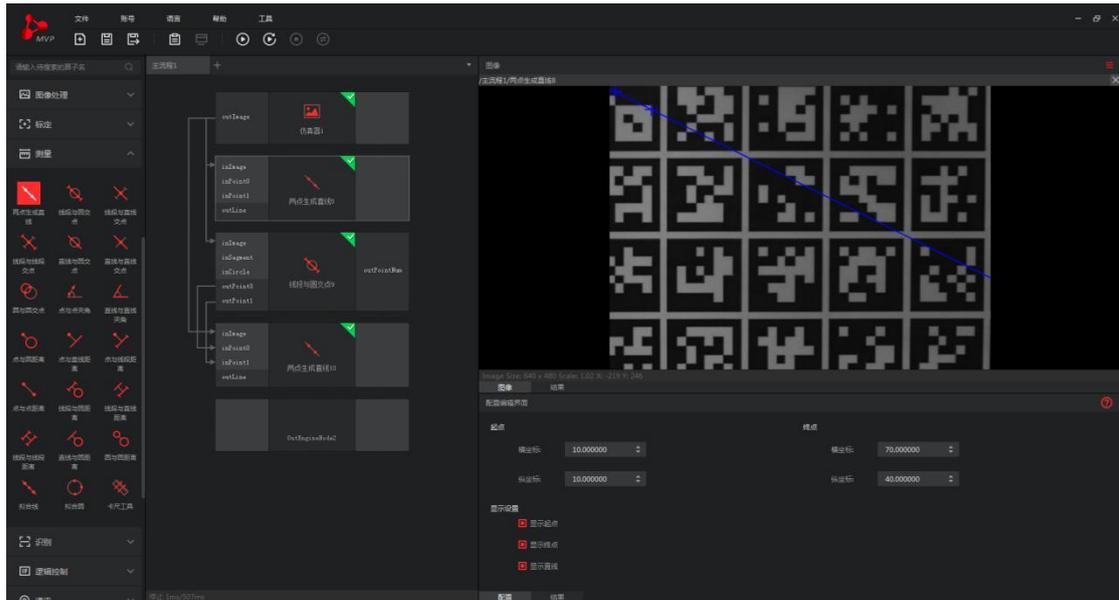


图 4.6.2 两点生成直线配置流程图

添加两点生成直线算子，添加仿真器或者相机作为图片输入，按如图 4.6.2 主流程连线后单机运行，在右上方显示区域看到生成的直线与输入的点，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入，如图 4.6.2 两点生成直线 10 所示），达到需要的效果。

### ⚠ 注意：

- ◆ 每次运行算子先查看是否有其他算子的输出作为此算子的输入，如果有则接收输入，不再接收配置参数，否则接收配置参数。其他算子也是同样策略。

### 参数描述

	参数名称	数据类型	取值范围	默认值	说明
起点	X (Start X)	float	[-10000,10000]		起点 X 坐标
	Y (Start Y)	float	[-10000,10000]		起点 Y 坐标
终点	X (End X)	float	[-10000,10000]		终点 X 坐标
	Y (End Y)	float	[-10000,10000]		终点 Y 坐标

## 4.6.2 线段与圆交点

线段与圆交点算子从输入或者配置接收一条线段和一个圆参数，根据接收参数求得线段和圆的位置关系并输出，可切换到结果界面查看输出信息（是否相交，交点个数和交点），如图 4.6.3 所示。



线段与圆交点:	是否相交	是	交点个数	2
	X0	105.355	Y0	105.355
	X1	34.6447	Y1	34.6447

图 4.6.3 线段与圆交点结果图

分为相交（交点个数为 2 或 1）、相切（交点个数为 1）、不相交（交点个数为 0）如图 4.6.4 所示四种情况。

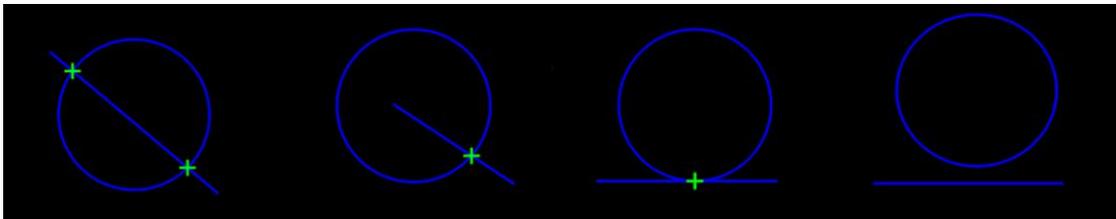


图 4.6.4 线段与圆位置分类图

操作步骤：

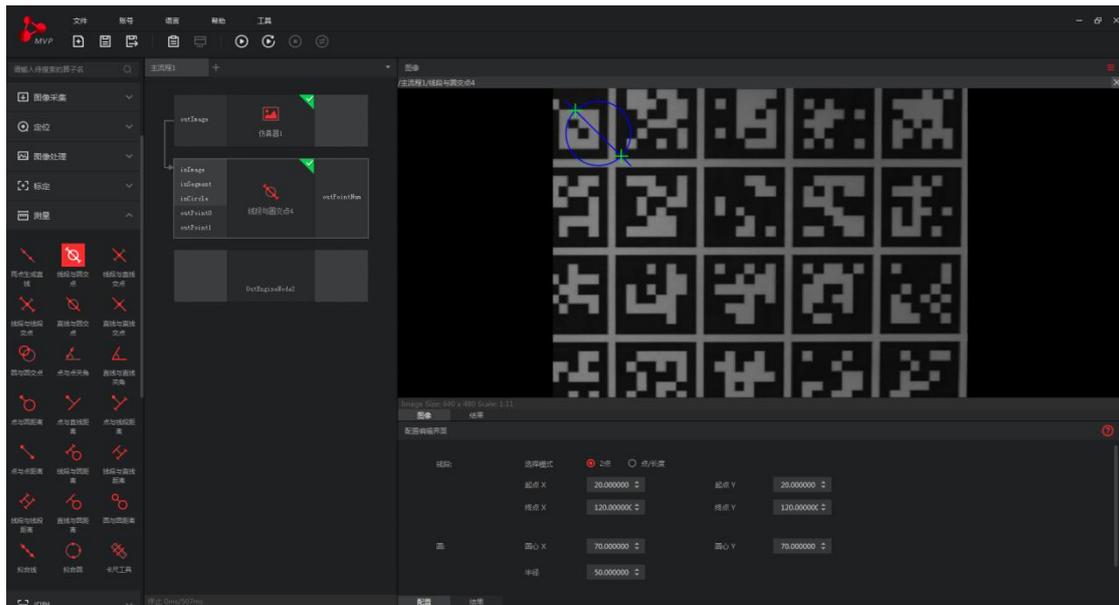


图 4.6.5 线段与圆交点配置流程图

添加线段与圆交点算子，添加仿真器或者相机作为图片输入，按如图 4.6.5 主流程连线后单机运行，在右上方显示区域看到生成的交点、输入的线段和圆，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称			数据类型	取值范围	默认值	说明
线段	2 点模式	起点 X	float	[-10000,10000]		线段起点 X 坐标
		起点 Y	float	[-10000,10000]		线段起点 Y 坐标
		终点 X	float	[-10000,10000]		线段终点 X 坐标
		终点 Y	float	[-10000,10000]		线段终点 Y 坐标
	点/长度模式	起点 X	float	[-10000,10000]		线段起点 X 坐标
		起点 Y	float	[-10000,10000]		线段起点 Y 坐标
		边长	float	[-10000,10000]		线段长度
		旋转	float	[-360,360]		顺时针旋转角度
圆	中心 X(Center X)	float	[-10000,10000]		圆中心 X 坐标	
	中心 Y(Center Y)	float	[-10000,10000]		圆中心 Y 坐标	

	半径	float	[-10000,10000]		圆半径
--	----	-------	----------------	--	-----

### 4.6.3 线段与直线交点

线段与直线交点算子从输入或者配置接收一条线段和一条直线参数，根据接收参数求得线段和直线的位置关系并输出，可切换到结果界面查看输出信息（是否相交，交点个数和交点），如图 4.6.6 所示。



图 4.6.6 线段与直线交点结果图

分为相交（交点个数为 1）、重合（交点个数为-1）、相离（交点个数为 0）如图 4.6.7 所示三种情况。



图 4.6.7 线段与直线位置分类图

操作步骤：

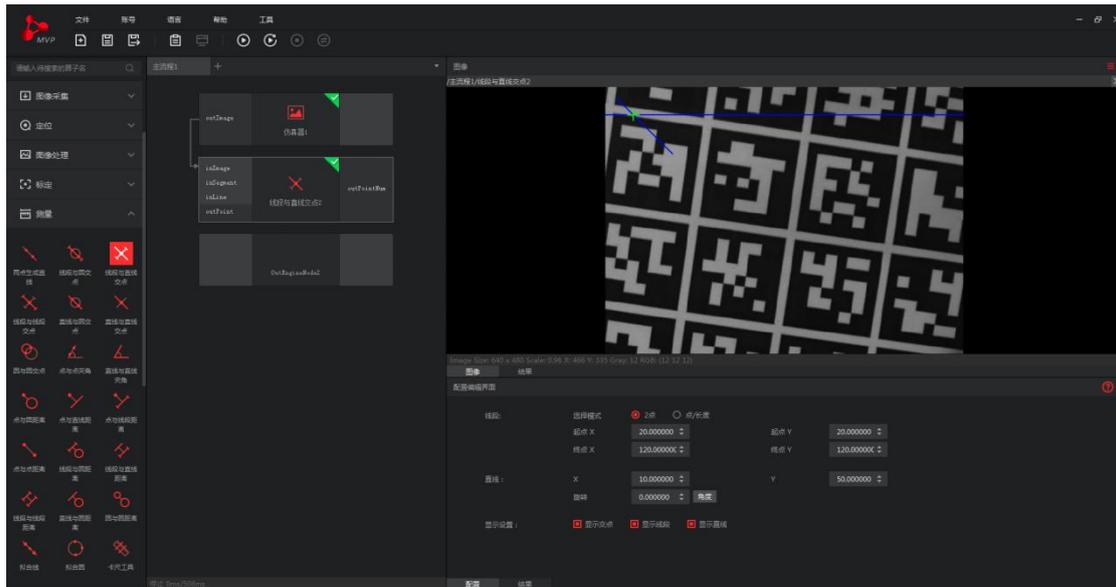


图 4.6.8 线段与直线交点配置流程图

添加线段与直线交点算子，添加仿真器或者相机作为图片输入，按如图 4.6.8 主流程连线后单机运行，在右上方显示区域看到生成的交点、输入的线段和直线，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称			数据类型	取值范围	默认值	说明
线段	2 点模式	起点 X	float	[-10000,10000]		线段起点 X 坐标
		起点 Y	float	[-10000,10000]		线段起点 Y 坐标
		终点 X	float	[-10000,10000]		线段终点 X 坐标
		终点 Y	float	[-10000,10000]		线段终点 Y 坐标
	点/长度模式	起点 X	float	[-10000,10000]		线段起点 X 坐标
		起点 Y	float	[-10000,10000]		线段起点 Y 坐标
		边长	float	[-10000,10000]		线段长度
		旋转	float	[-360,360]		顺时针旋转角度
直线	X	float	[-10000,10000]		直线上一点 X 坐标	
	Y	float	[-10000,10000]		直线上一点 Y 坐标	

	旋转	float	[-360,360]		顺时针旋转角度
--	----	-------	------------	--	---------

#### 4.6.4 线段与线段交点

线段与线段交点算子从输入或者配置接收两条线段参数，根据接收参数求得两条线段之间的位置关系并输出，可切换到结果界面查看输出信息（是否相交，交点个数和交点），如图 4.6.9 所示。



图 4.6.9 线段与线段交点结果图

分为相交（交点个数为 1）、重合（交点个数为-1）、相离（交点个数为 0）如图 4.6.10 所示三种情况。



图 4.6.10 线段与线段位置分类图

#### 操作步骤

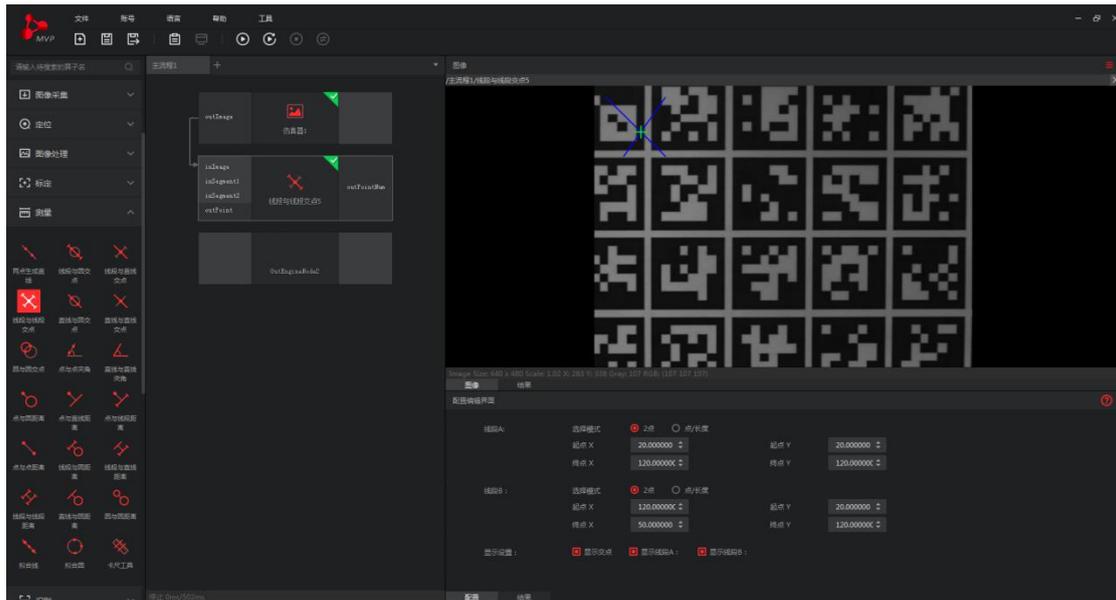


图 4.6.11 线段与线段交点配置流程图

添加线段与线段交点算子，添加仿真器或者相机作为图片输入，按如图 4.6.11 主流程连线后单机运行，在右上方显示区域看到生成的交点与输入的线段，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称			数据类型	取值范围	默认值	说明
线段 A	2 点模式	起点 X	float	[-10000,10000]		线段 A 起点 X 坐标
		起点 Y	float	[-10000,10000]		线段 A 起点 Y 坐标
		终点 X	float	[-10000,10000]		线段 A 终点 X 坐标
		终点 Y	float	[-10000,10000]		线段 A 终点 Y 坐标
	点/长度模式	起点 X	float	[-10000,10000]		线段 A 起点 X 坐标
		起点 Y	float	[-10000,10000]		线段 A 起点 Y 坐标
		边长	float	[-10000,10000]		线段 A 长度
		旋转	float	[-360,360]		顺时针旋转角度
线段 B	2 点模式	起点 X	float	[-10000,10000]		线段 B 起点 X 坐标
		起点 Y	float	[-10000,10000]		线段 B 起点 Y 坐标

		终点 X	float	[-10000,10000]		线段 B 终点 X 坐标
		终点 Y	float	[-10000,10000]		线段 B 终点 Y 坐标
	点/长度 模式	起点 X	float	[-10000,10000]		线段 B 起点 X 坐标
		起点 Y	float	[-10000,10000]		线段 B 起点 Y 坐标
		边长	float	[-10000,10000]		线段 B 长度
		旋转	float	[-360,360]		顺时针旋转角度

## 4.6.5 直线与圆交点

直线与圆交点算子从输入或者配置接收一条直线和一个圆参数，根据接收参数求得直线和圆的位置关系并输出，可切换到结果界面查看输出信息（是否相交，交点个数和交点），如图 4.6.12 所示。



图 4.6.12 直线与圆交点算子结果图

分为相交（交点个数为 2）、相切（交点个数为 1）、相离（交点个数为 0）如图 4.6.13 所示三种情况：

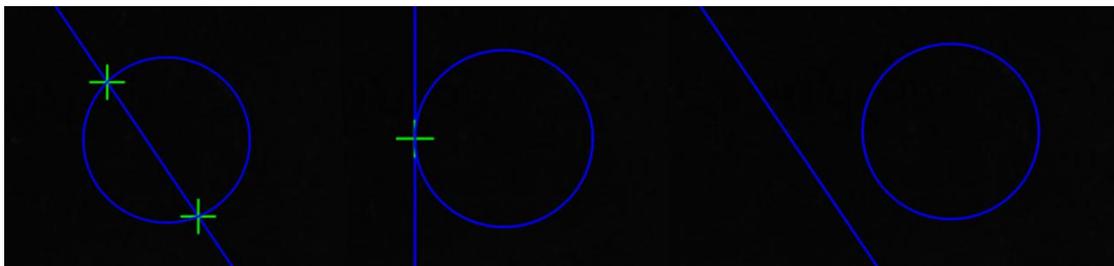


图 4.6.13 直线与圆位置关系图

### 操作步骤

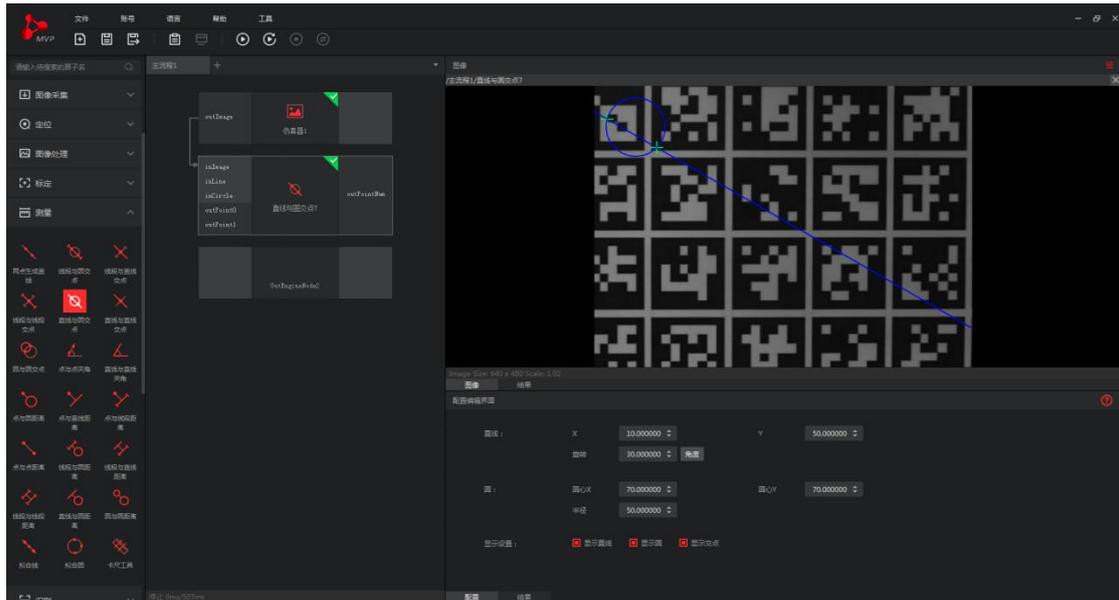


图 4.6.14 直线与圆交点配置流程图

添加直线与圆交点算子，添加仿真器或者相机作为图片输入，按如图 4.6.14 主流程连线后单机运行，在右上方显示区域看到生成的交点、输入的直线和圆，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

### 参数描述

参数名称	数据类型	取值范围	默认值	说明
直线	X	float	[-10000,10000]	直线上一点 X 坐标
	Y	float	[-10000,10000]	直线上一点 Y 坐标
	旋转	float	[-360,360]	顺时针旋转角度
圆	中心 X(Center X)	float	[-10000,10000]	圆中心 X 坐标
	中心 Y(Center Y)	float	[-10000,10000]	圆中心 Y 坐标
	半径	float	[-10000,10000]	圆半径

### 4.6.6 直线与直线交点

直线与直线交点算子从输入或者配置接收两条直线参数，根据接收参数求得线段和直线的位置关系并输出，可切换到结果界面查看输出信息（是否相交，交点个数、交点、直线间的夹角，提示：直线平行、重合夹角为 0），如图 4.6.15 所示。



图 4.6.15 直线与直线交点算子结果图

分为相交（交点个数为 1）、重合（交点个数为-1）、平行（交点个数为 0）如图 4.6.16 所示三种情况。

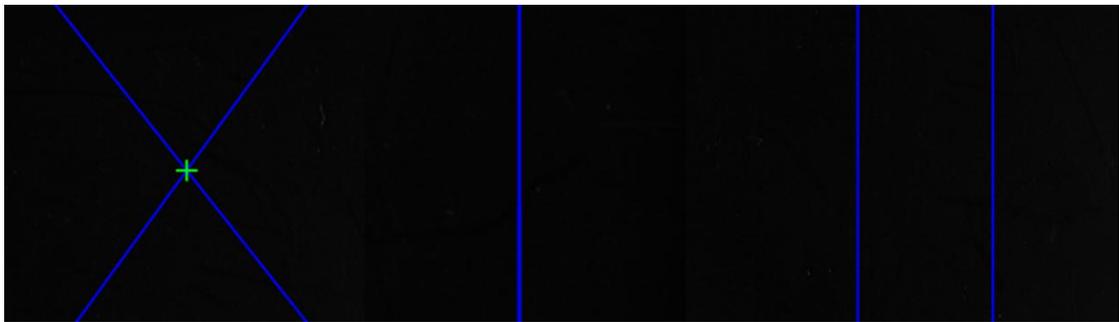


图 4.6.16 直线与直线位置分类图

操作步骤：

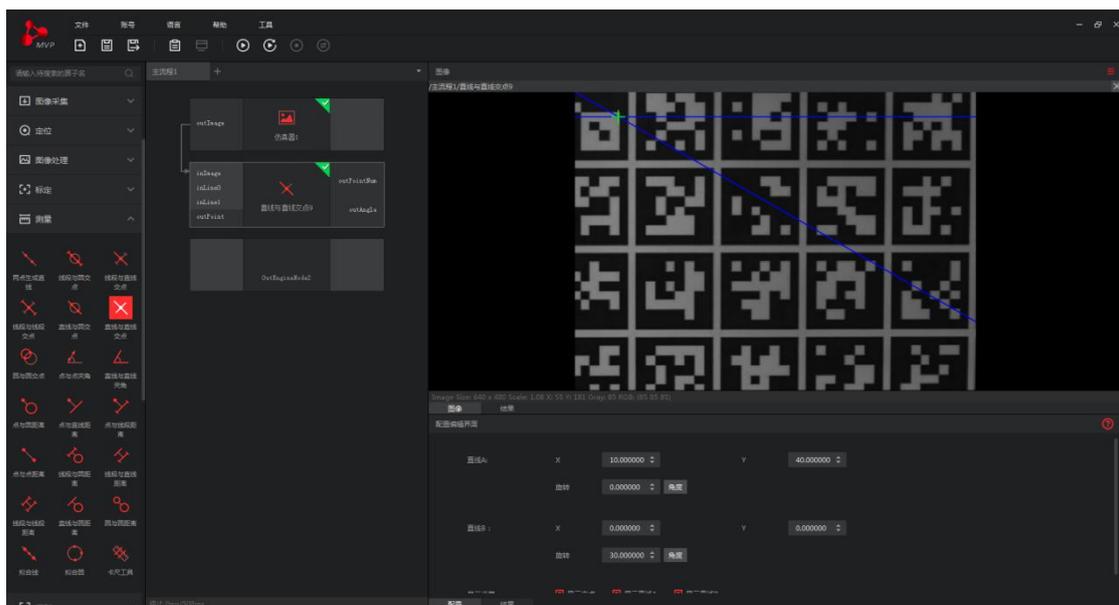


图 4.6.17 直线与直线交点配置流程图

添加线段与直线交点算子，添加仿真器或者相机作为图片输入，按如图 4.6.17 主流程连线后单机运行，在右上方显示区域看到生成的交点、输入的直线，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称	数据类型	取值范围	默认值	说明
直线 A	X	float	[-10000,10000]	直线 A 上一点 X 坐标
	Y	float	[-10000,10000]	直线 A 上一点 Y 坐标
	旋转	float	[-360,360]	顺时针旋转角度
直线 B	X	float	[-10000,10000]	直线 B 上一点 X 坐标
	Y	float	[-10000,10000]	直线 B 上一点 Y 坐标
	旋转	float	[-360,360]	顺时针旋转角度

## 4.6.7 圆与圆交点

圆与圆交点算子从输入或者配置接收两个圆参数，根据接收参数求得两圆之间的位置关系并输出，可切换到结果界面查看输出信息（是否相交，交点个数和交点），如图 4.6.18 所示。



图 4.6.18 圆与圆交点算子结果图

分为相交（交点个数为 2）、重合（交点个数为-1）相切（交点个数为 1）、相离（交点个数为 0）如图 4.6.19 所示四种情况。

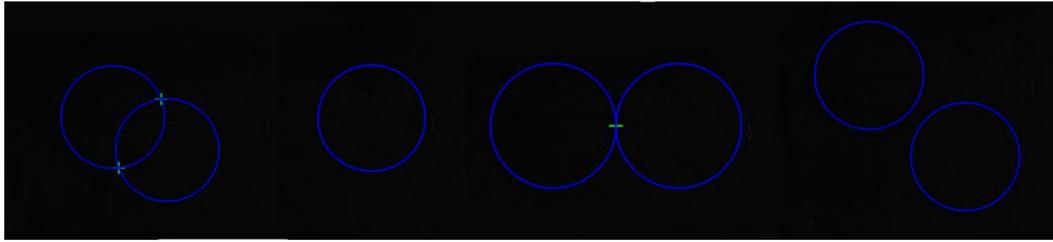


图 4.6.19 圆与圆位置分类图

## 操作步骤：

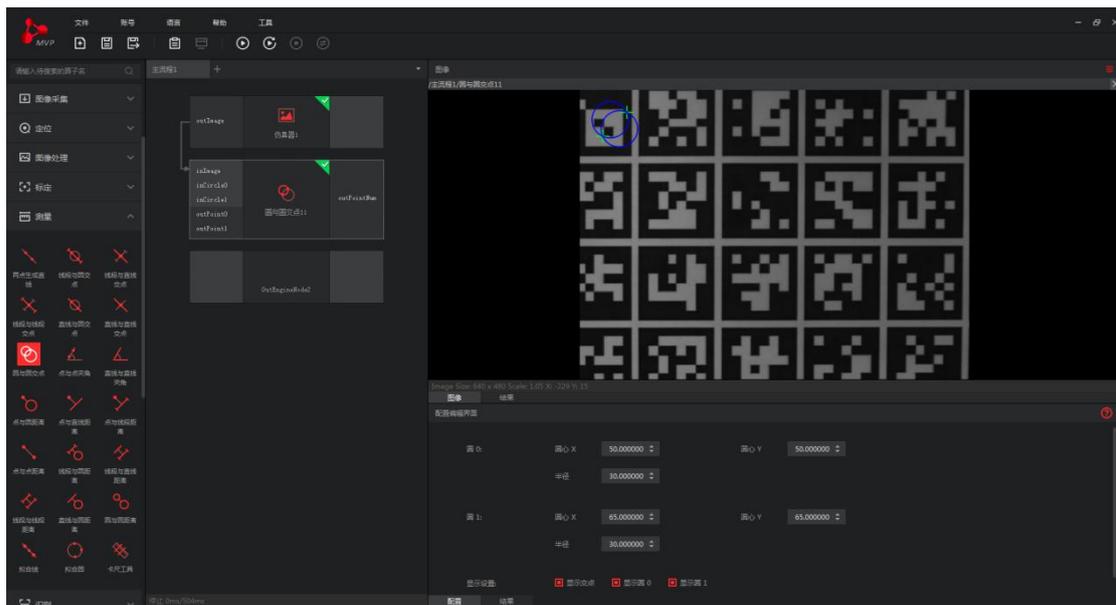


图 4.6.20 圆与圆交点配置流程图

添加圆与圆交点算子，添加仿真器或者相机作为图片输入，按如图 4.6.20 主流程连线后单机运行，在右上方显示区域看到生成的交点、输入的圆，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

	参数名称	数据类型	取值范围	默认值	说明
圆 A	中心 X(Center X)	float	[-10000,10000]		圆 A 中心 X 坐标
	中心 Y(Center Y)	float	[-10000,10000]		圆 A 中心 Y 坐标
	半径	float	[-10000,10000]		圆 A 半径
圆 B	中心 X(Center X)	float	[-10000,10000]		圆 B 中心 X 坐标

中心 Y(Center Y)	float	[-10000,10000]	圆 B 中心 Y 坐标
半径	float	[-10000,10000]	圆 B 半径

## 4.6.8 点与点夹角

点与点夹角算子从输入或者配置接收两个点参数，根据接收参数求得两点坐标连线和图像坐标系的夹角并输出，如果输出两点坐标相同则算子失败，可切换到结果界面查看输出信息（夹角大小），如图 4.6.21 所示。



图 4.6.21 点与点夹角算子结果图

### 操作步骤

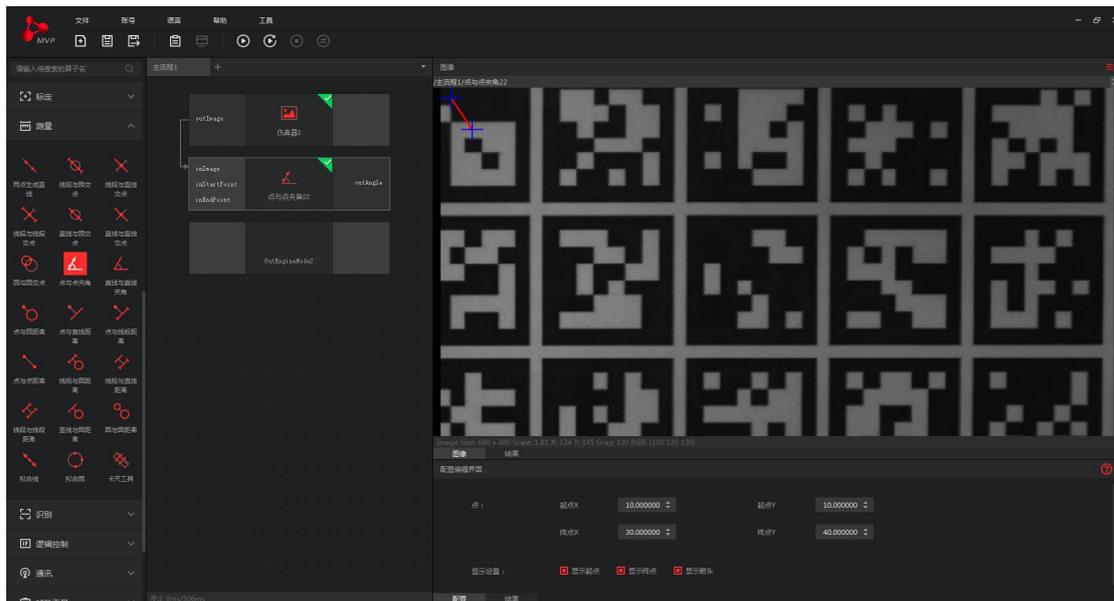


图 4.6.22 点与点夹角配置流程图

添加点与点夹角算子，添加仿真器或者相机作为图片输入，按如图 4.6.22 主流程连线后单机运行，在右上方显示区域看到生成的箭头、输入的点，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

### 参数描述

	参数名称	数据类型	取值范围	默认值	说明
起点	X (Start X)	float	[-10000,10000]		起点 X 坐标
	Y (Start Y)	float	[-10000,10000]		起点 Y 坐标
终点	X (End X)	float	[-10000,10000]		终点 X 坐标
	Y (End Y)	float	[-10000,10000]		终点 Y 坐标

## 4.6.9 直线与直线夹角

直线与直线夹角算子从输入或者配置接收两条直线参数，根据接收参数求得两直线之间的夹角并输出，可切换到结果界面查看输出信息（夹角大小，提示：直线平行、重合夹角为 0），如图 4.6.23 所示。



图 4.6.23 直线与直线夹角算子结果图

### 操作步骤

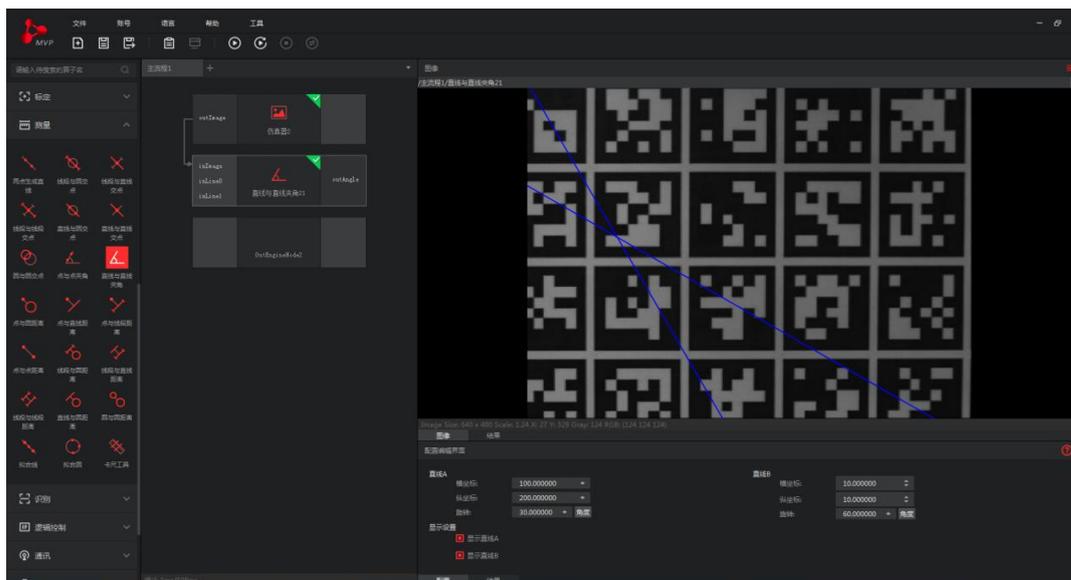


图 4.6.24 直线与直线夹角配置流程图

添加直线与直线夹角算子，添加仿真器或者相机作为图片输入，按如图 4.6.24 主流程连线后单机运行，在右上方显示区域看到输入的直线，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

### 参数描述

参数名称		数据类型	取值范围	默认值	说明
直线 A	X	float	[-10000,10000]		直线 A 上一点 X 坐标
	Y	float	[-10000,10000]		直线 A 上一点 Y 坐标
	旋转	float	[-360,360]		顺时针旋转角度
直线 B	X	float	[-10000,10000]		直线 B 上一点 X 坐标
	Y	float	[-10000,10000]		直线 B 上一点 Y 坐标
	旋转	float	[-360,360]		顺时针旋转角度

### 4.6.10 点与圆距离

点与圆距离算子从输入或者配置接收一个点和一个圆参数，根据接收参数求得点与圆之间的位置关系并输出，可切换到结果界面查看输出信息（点与圆之间的距离、圆上到已知点距离最短的点），如图 4.6.22 所示。

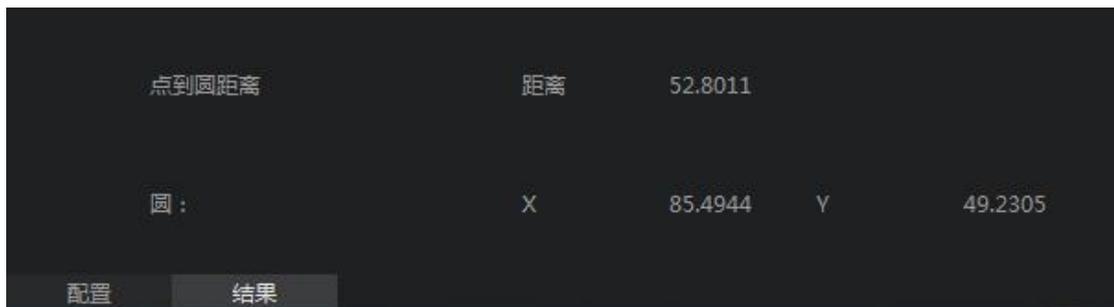


图 4.6.25 点与圆距离算子结果图

分为点在圆外（距离为正数）、点在圆上（距离为 0）、点在圆内（距离为负数）如图 4.6.26 所示三种情况。

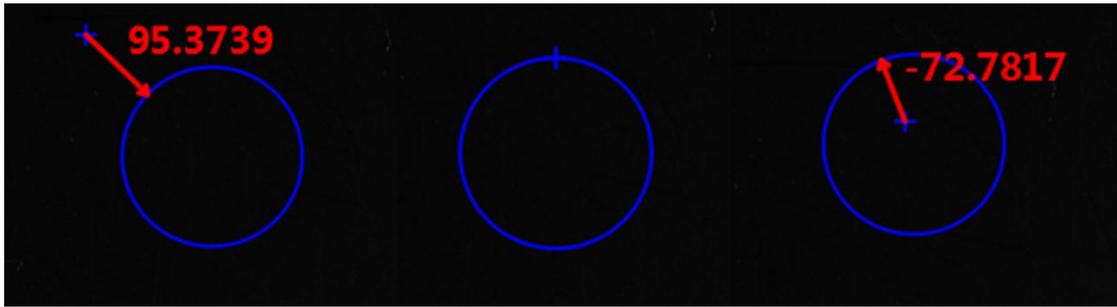


图 4.6.26 点与圆距离位置分类图

## 操作步骤

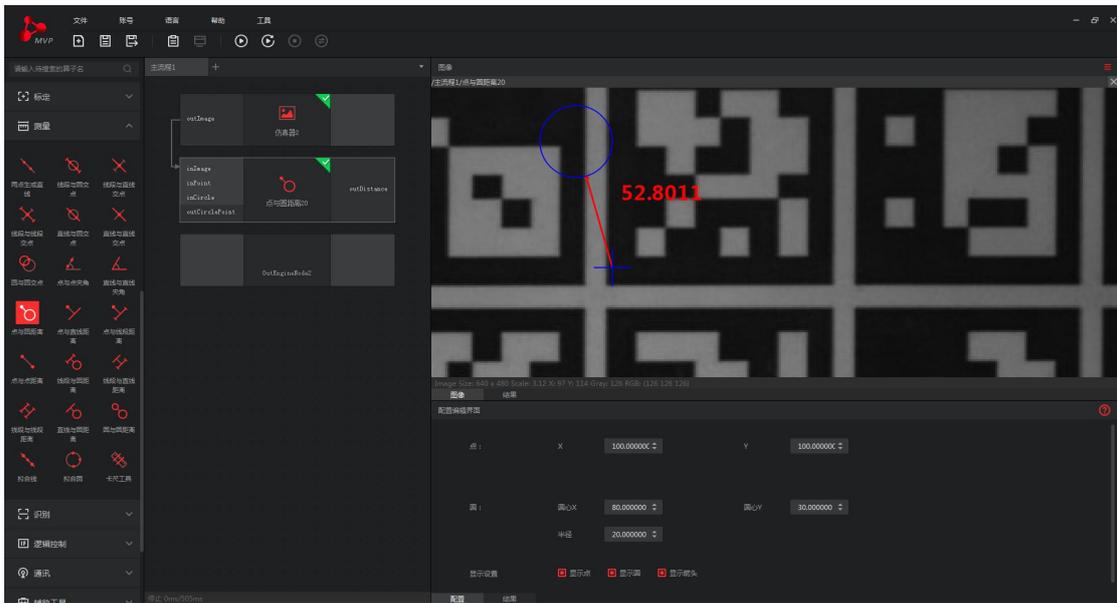


图 4.6.27 点与圆距离配置流程图

添加点与圆距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.27 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的圆和点，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

	参数名称	数据类型	取值范围	默认值	说明
点	X	float	[-10000,10000]		点的 X 坐标
	Y	float	[-10000,10000]		点的 Y 坐标
圆	中心 X(Center X)	float	[-10000,10000]		圆中心 X 坐标

中心 Y(Center Y)	float	[-10000,10000]	圆中心 Y 坐标
半径	float	[-10000,10000]	圆半径

## 4.6.11 点与直线距离

点与直线距离算子从输入或者配置接收一个点和一条直线参数，根据接收参数求得点与直线之间的位置关系并输出，可切换到结果界面查看输出信息（点与圆之间的距离、直线上到已知点距离最短的点），如图 4.6.28 所示。



图 4.6.28 点与直线距离算子结果图

分为点不在直线上（距离为正数）、点在直线上（距离为 0）两种情况。

### 操作步骤

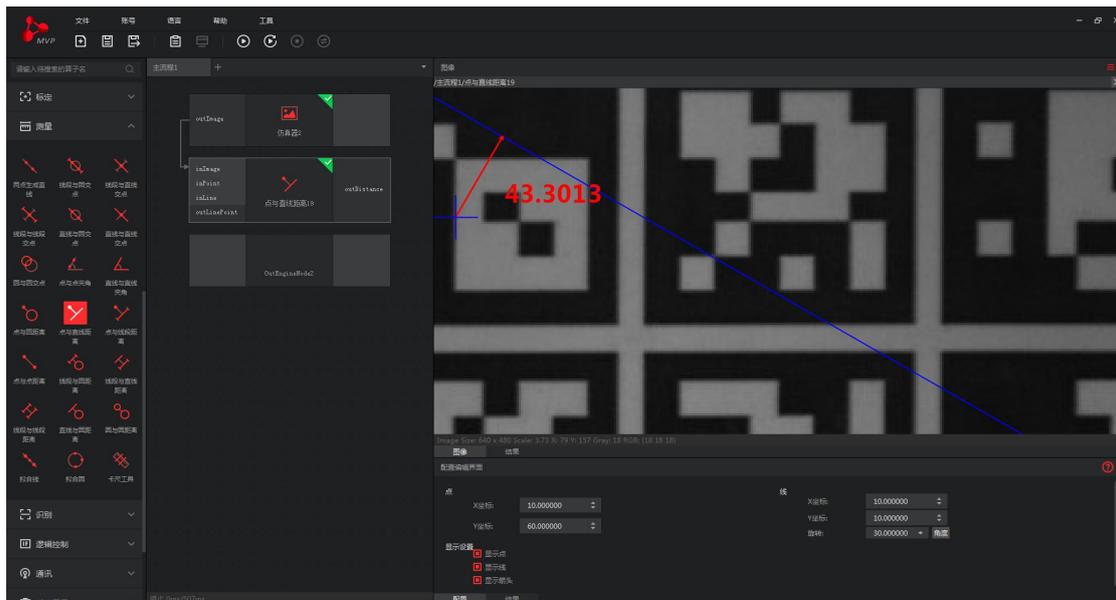


图 4.6.29 点与直线距离配置流程图

添加点与直线距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.29 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的直线和点，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

### 参数描述

参数名称	数据类型	取值范围	默认值	说明
点	X	float	[-10000,10000]	点的 X 坐标
	Y	float	[-10000,10000]	点的 Y 坐标
直线	X	float	[-10000,10000]	直线上一点 X 坐标
	Y	float	[-10000,10000]	直线上一点 Y 坐标
	旋转	float	[-360,360]	顺时针旋转角度

### 4.6.12 点与线段距离

点与线段距离算子从输入或者配置接收一个点和一条线段参数，根据接收参数求得点与线段之间的位置关系并输出，可切换到结果界面查看输出信息（点与圆之间的距离、以及线段上到已知点距离最短的点），如图 4.6.30 所示。分为点不在线段上（距离为正数）、点在线段上（距离为 0）两种情况。



图 4.6.30 点与线段距离算子结果图

操作步骤：

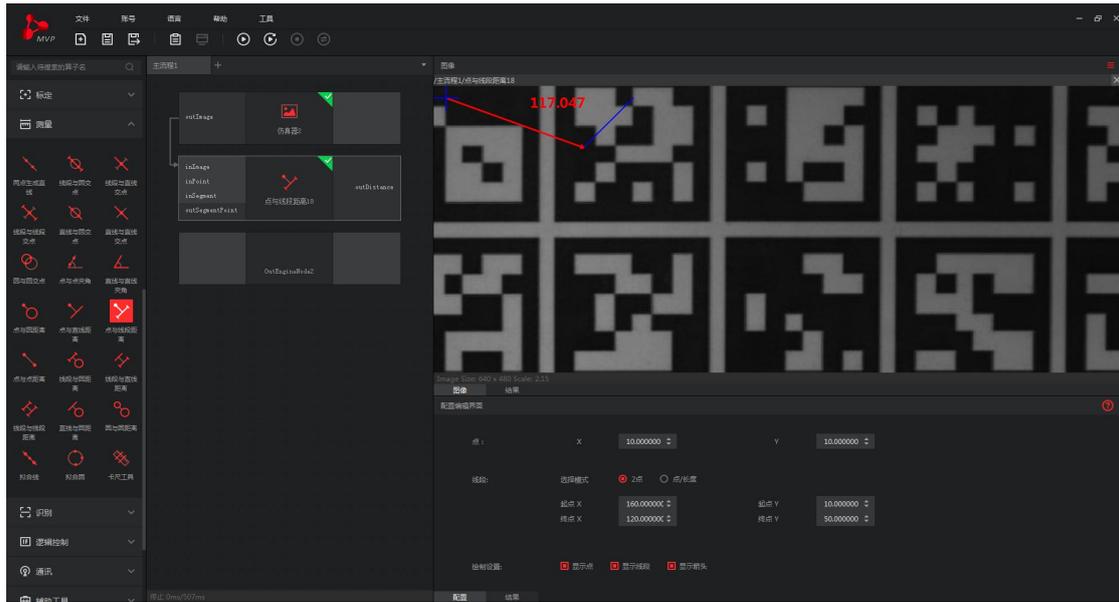


图 4.6.31 点与线段距离配置流程图

添加点与线段距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.31 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的线段和点，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称		数据类型	取值范围	默认值	说明	
点	X	float	[-10000,10000]		点的 X 坐标	
	Y	float	[-10000,10000]		点的 Y 坐标	
线段	2 点模式	起点 X	float	[-10000,10000]		线段起点 X 坐标
		起点 Y	float	[-10000,10000]		线段起点 Y 坐标
		终点 X	float	[-10000,10000]		线段终点 X 坐标
		终点 Y	float	[-10000,10000]		线段终点 Y 坐标
	点/长度模式	起点 X	float	[-10000,10000]		线段起点 X 坐标
		起点 Y	float	[-10000,10000]		线段起点 Y 坐标
		边长	float	[-10000,10000]		线段长度

旋转

float

[-360,360]

顺时针旋转角度

### 4.6.13 点与点距离

点与点距离算子从输入或者配置接收两个点参数，根据接收参数求得点与点之间的位置关系并输出，如果输入两点坐标相同，不显示交点角度，可切换到结果界面查看输出信息（点与点之间的距离、两点坐标连线和图像坐标系的夹角），如图 4.6.32 所示。



图 4.6.32 点与点距离算子结果图

### 操作步骤

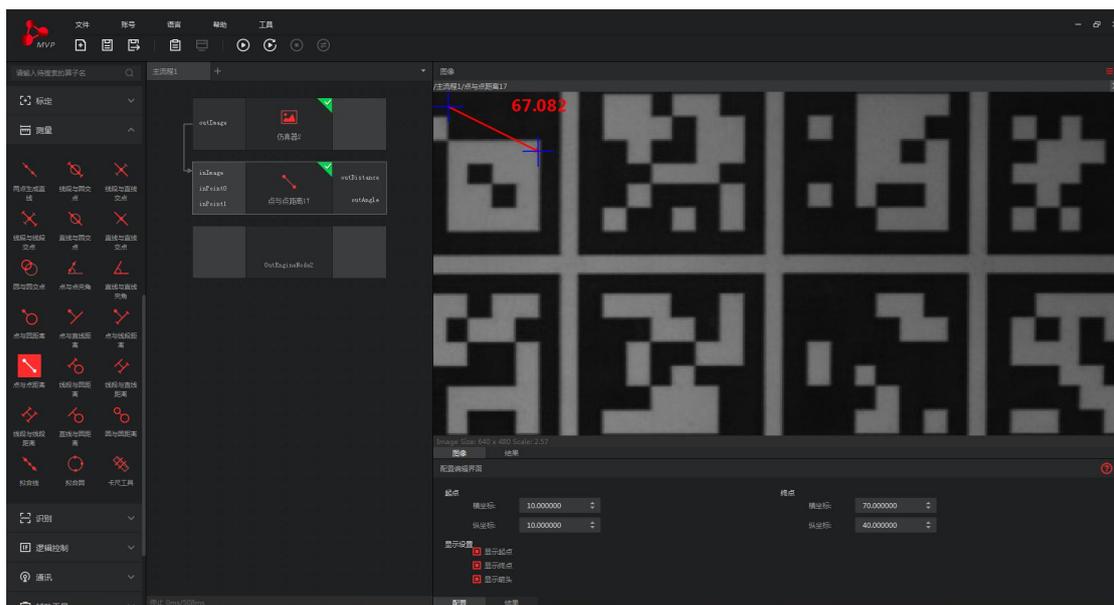


图 4.6.33 点与点距离配置流程图

添加点与点距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.33 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的点，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

### 参数描述

参数名称	数据类型	取值范围	默认值	说明
起点	X (Start X)	float	[-10000,10000]	起点 X 坐标
	Y (Start Y)	float	[-10000,10000]	起点 Y 坐标
终点	X (End X)	float	[-10000,10000]	终点 X 坐标
	Y (End Y)	float	[-10000,10000]	终点 Y 坐标

#### 4.6.14 线段与圆距离

线段与圆距离算子从输入或者配置接收一条线段和一个圆参数，根据接收参数求得线段与圆之间的位置关系并输出，可切换到结果界面查看输出信息（线段与圆之间的距离与圆和线段上都任意取一点，距离最短时线段上和圆上所取的点），如图 4.6.34 所示。



图 4.6.34 线段与圆距离算子结果图

分为相交（距离为 0）、相切（距离为 0）、相离（距离为正数）、线段在圆内（距离为负数）如图 4.6.35 所示五种情况。

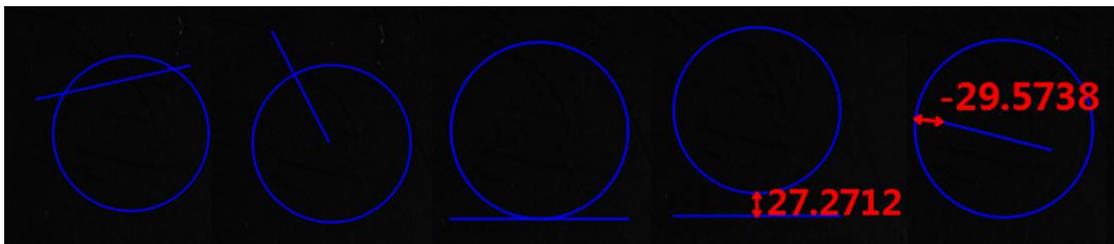


图 4.6.35 线段与圆位置分类图

#### 操作步骤

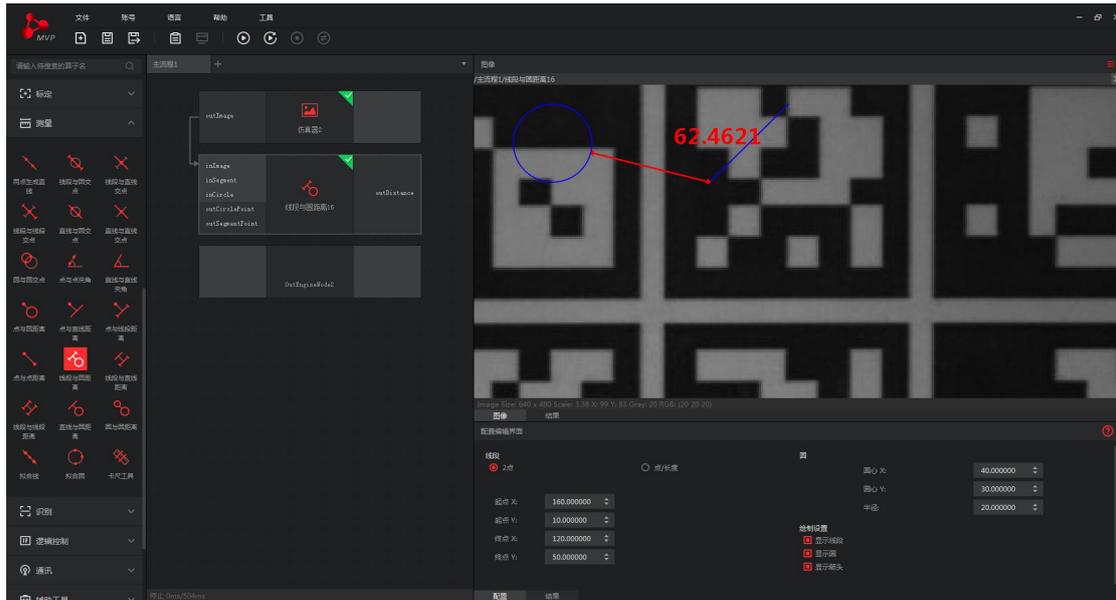


图 4.6.36 线段与圆距离配置流程图

添加线段与圆距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.33 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的圆和线段，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称			数据类型	取值范围	默认值	说明
线段	2 点模式	起点 X	float	[-10000,10000]		线段起点 X 坐标
		起点 Y	float	[-10000,10000]		线段起点 Y 坐标
		终点 X	float	[-10000,10000]		线段终点 X 坐标
		终点 Y	float	[-10000,10000]		线段终点 Y 坐标
圆	中心 X(Center X)	float	[-10000,10000]		圆中心 X 坐标	
	中心 Y(Center Y)	float	[-10000,10000]		圆中心 Y 坐标	
	半径	float	[-10000,10000]		圆半径	

## 4.6.15 线段与直线距离

线段与直线距离算子从输入或者配置接收一条线段和一个直线参数，根据接收参数求得线段与直线之间的位置关系并输出，可切换到结果界面查看输出信息（线段与直线之间的距离与直线和线段上都任意取一点，距离最短时线段上和直线上所取的点），如图 4.6.37 所示。



图 4.6.37 线段与直线距离算子结果图

分为相交（距离为 0）、重合（距离为 0）、相离（距离为正数）如图 4.6.38 所示三种情况。

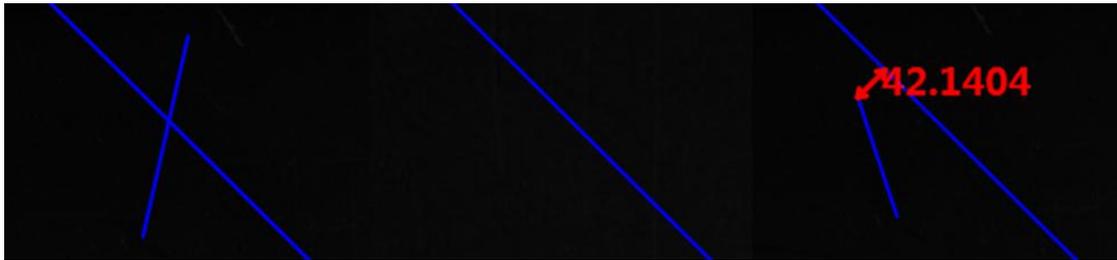


图 4.6.38 线段与直线位置分类图

## 操作步骤

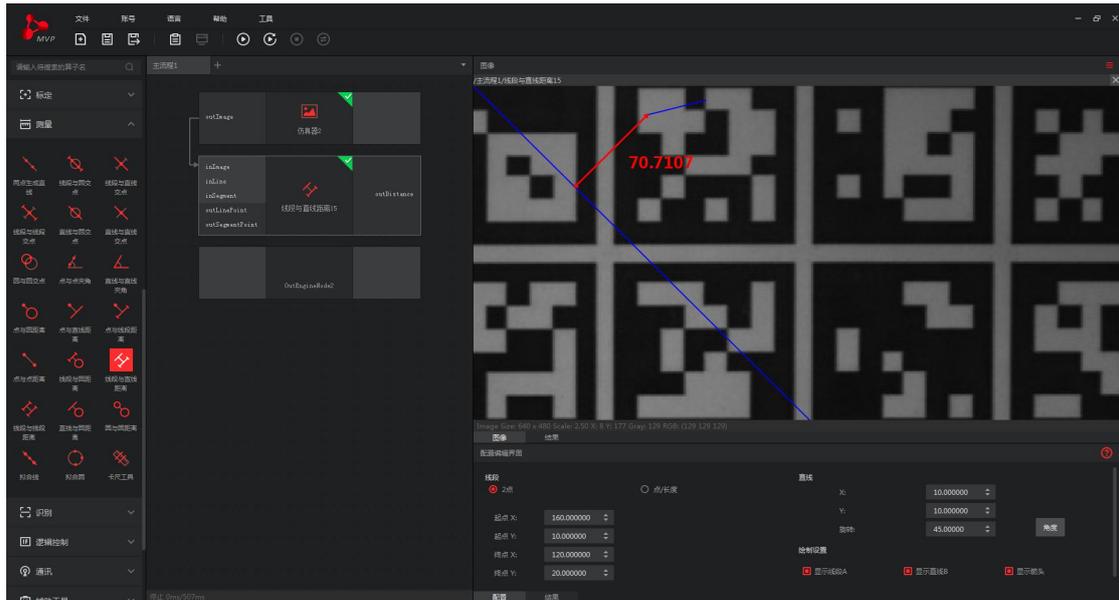


图 4.6.39 线段与直线距离配置流程图

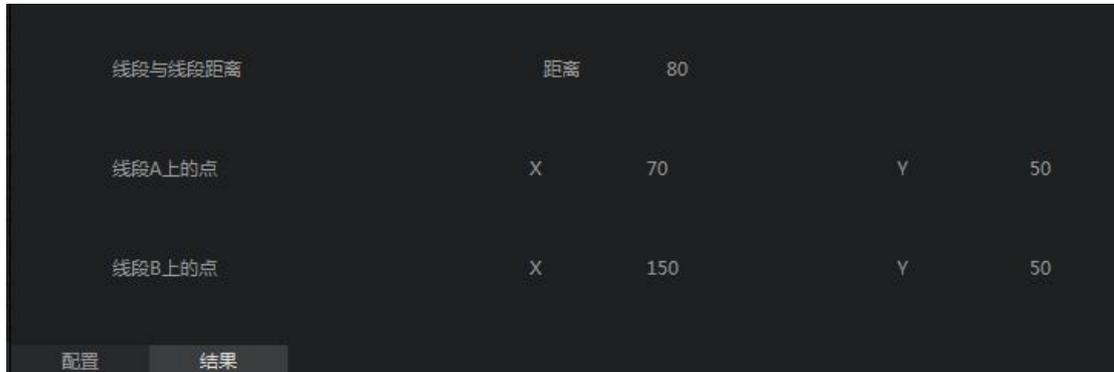
添加线段与直线距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.39 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的直线和线段，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

### 参数描述

参数名称			数据类型	取值范围	默认值	说明
线段	2 点模式	起点 X	float	[-10000,10000]		线段起点 X 坐标
		起点 Y	float	[-10000,10000]		线段起点 Y 坐标
		终点 X	float	[-10000,10000]		线段终点 X 坐标
		终点 Y	float	[-10000,10000]		线段终点 Y 坐标
直线	X	float	[-10000,10000]		直线上一点 X 坐标	
	Y	float	[-10000,10000]		直线上一点 Y 坐标	
	旋转	float	[-360,360]		顺时针旋转角度	

## 4.6.16 线段与线段距离

线段与线段距离算子从输入或者配置接收两条线段参数，根据接收参数求得线段与线段之间的位置关系并输出，可切换到结果界面查看输出信息（线段与线段之间的距离与两线段上各都任意取一点，距离最短时两线段上所取的点），如图 4.6.40 所示。



线段与线段距离	距离	80		
线段A上的点	X	70	Y	50
线段B上的点	X	150	Y	50

配置    结果

图 4.6.40 线段与线段距离算子结果图

分为相交（距离为 0）、重合（距离为 0）、相离（距离为正数）如图 4.6.41 所示三种情况。

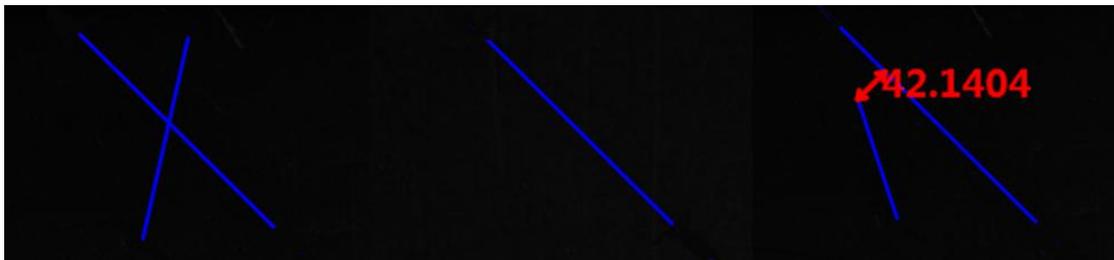


图 4.6.41 线段与线段位置分类图

操作步骤：

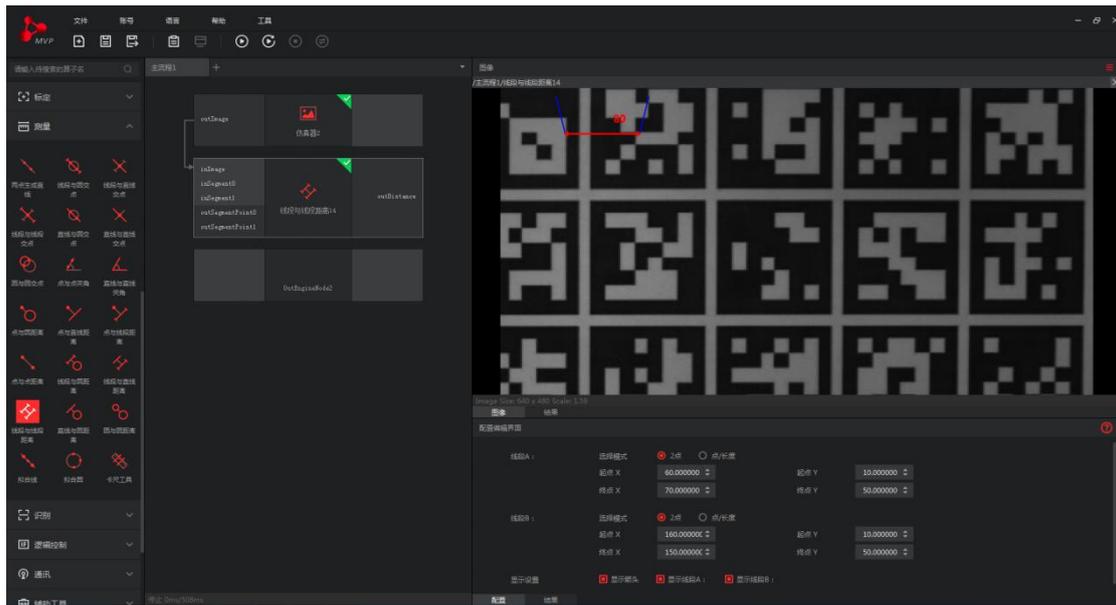


图 4.6.42

添加线段与线段距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.42 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的线段，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称			数据类型	取值范围	默认值	说明	
线段 A	2 点模式	起点 X	float	[-10000,10000]		线段 A 起点 X 坐标	
		起点 Y	float	[-10000,10000]		线段 A 起点 Y 坐标	
		终点 X	float	[-10000,10000]		线段 A 终点 X 坐标	
		终点 Y	float	[-10000,10000]		线段 A 终点 Y 坐标	
线段 B	2 点模式	起点 X	float	[-10000,10000]		线段 B 起点 X 坐标	
		起点 Y	float	[-10000,10000]		线段 B 起点 Y 坐标	
		终点 X	float	[-10000,10000]		线段 B 终点 X 坐标	
		终点 Y	float	[-10000,10000]		线段 B 终点 Y 坐标	
	点/长度模式	起点 X	float	[-10000,10000]		线段 B 起点 X 坐标	
		起点 Y	float	[-10000,10000]		线段 B 起点 Y 坐标	

	边长	float	[-10000,10000]		线段 B 长度
	旋转	float	[-360,360]		顺时针旋转角度

### 4.6.17 直线与圆距离

直线与圆距离算子从输入或者配置接收一条直线和一个圆参数，根据接收参数求得直线与圆之间的位置关系并输出，可切换到结果界面查看输出信息（直线与圆之间的距离与圆和直线上都任意取一点，距离最短时直线和圆上所取的点），如图 4.6.43 所示：

直线与圆距离：	距离	37.9423		
直线：	X	78.9711	Y	49.8205
圆：	X	60	Y	82.6795
配置		结果		

图 4.6.43 直线与圆距离算子结果图

分为相交（距离为 0）、相切（距离为 0）、相离（距离为正数）如图 4.6.44 所示三种情况。

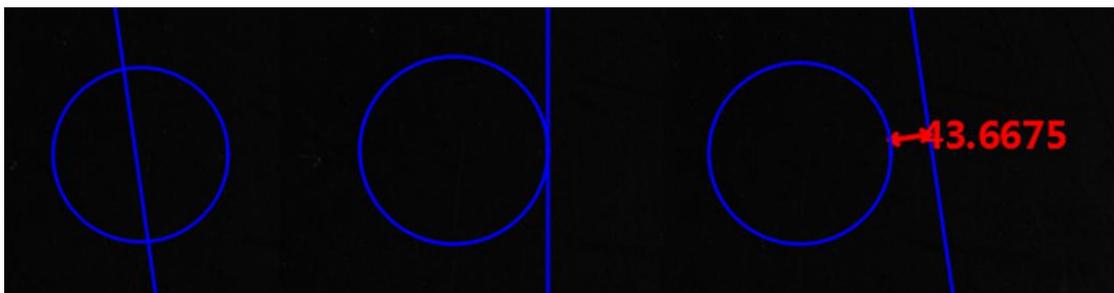


图 4.6.44 直线与圆位置分类图

操作步骤：

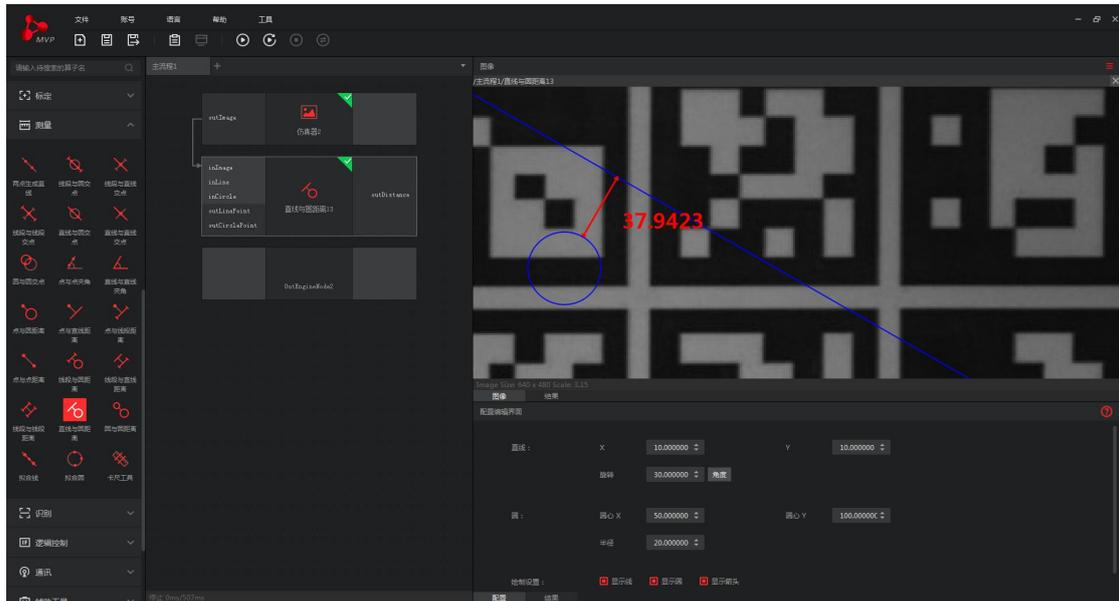


图 4.6.45 直线与圆距离配置流程图

添加直线与圆距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.45 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的圆和直线，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

### 参数描述

	参数名称	数据类型	取值范围	默认值	说明
直线	X	float	[-10000,10000]		直线上一点 X 坐标
	Y	float	[-10000,10000]		直线上一点 Y 坐标
	旋转	float	[-360,360]		顺时针旋转角度
圆	中心 X(Center X)	float	[-10000,10000]		圆中心 X 坐标
	中心 Y(Center Y)	float	[-10000,10000]		圆中心 Y 坐标
	半径	float	[-10000,10000]		圆半径

## 4.6.18 圆与圆距离

圆与圆距离算子从输入或者配置接收两个圆参数，根据接收参数求得圆与圆之间的位置关系并输出，可切换到结果界面查看输出信息（圆与圆之间的距离与两圆上都各任意取一点，距离最短时两圆上所取的点），如图 4.6.46 所示。



图 4.6.45 圆与圆距离算子结果图

分为相交（距离为 0）、相切（距离为 0）、相离（距离为正数）、重合（距离为 0）、小圆在大圆内（距离为负数）如图 4.6.47 所示五种情况。

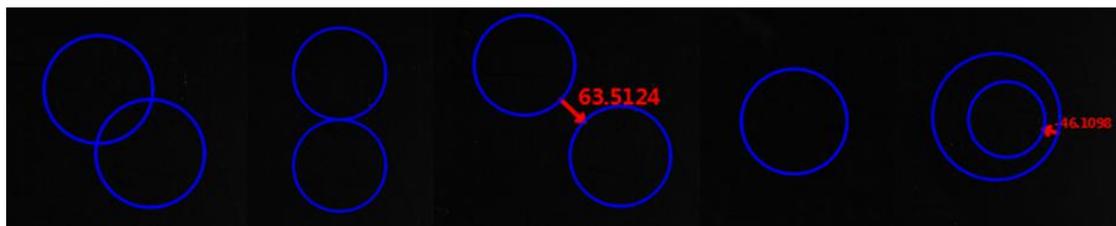


图 4.6.47 圆与圆位置分类图

操作步骤:

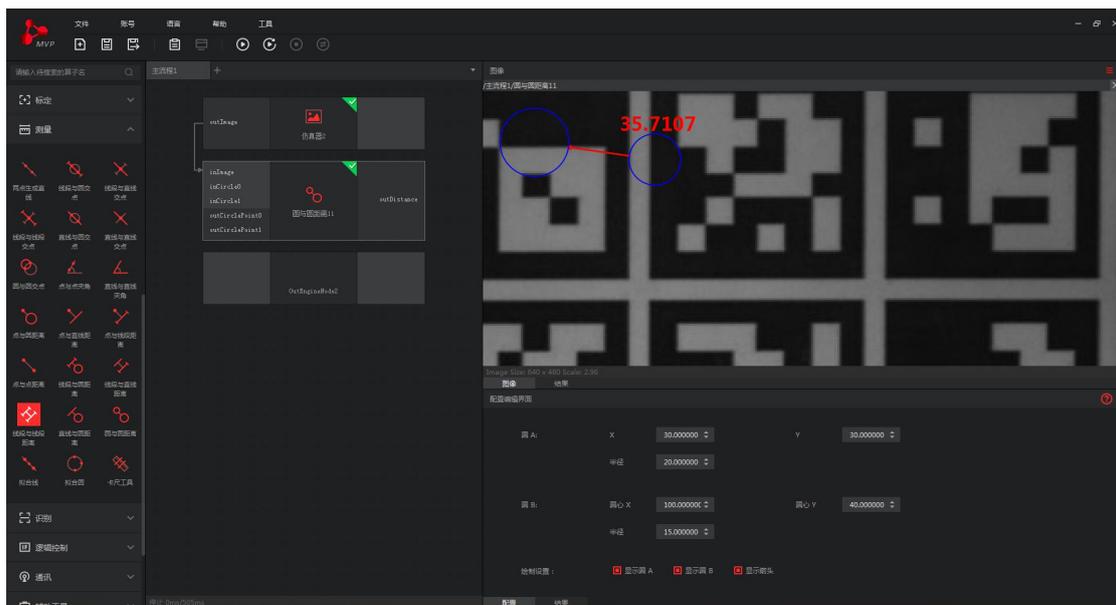


图 4.6.48 圆与圆距离配置流程图

添加圆与圆距离算子，添加仿真器或者相机作为图片输入，按如图 4.6.48 主流程连线后单机运行，在右上方显示区域看到距离箭头、输入的圆，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

### 参数描述

	参数名称	数据类型	取值范围	默认值	说明
圆 A	中心 X(Center X)	float	[-10000,10000]		圆 A 中心 X 坐标
	中心 Y(Center Y)	float	[-10000,10000]		圆 A 中心 Y 坐标
	半径	float	[-10000,10000]		圆 A 半径
圆 B	中心 X(Center X)	float	[-10000,10000]		圆 B 中心 X 坐标
	中心 Y(Center Y)	float	[-10000,10000]		圆 B 中心 Y 坐标
	半径	float	[-10000,10000]		圆 B 半径

### 4.6.19 拟合线

拟合线算子从输入或者配置接收十个以内点参数，根据接收参数（输入点与添加点）和配置中设置的忽略点数来拟合线并输出，可切换到结果界面查看输出信息（拟合直线、RMS 误差），如图 4.6.49 所示。



图 4.6.49 拟合线算子结果图

### 操作步骤

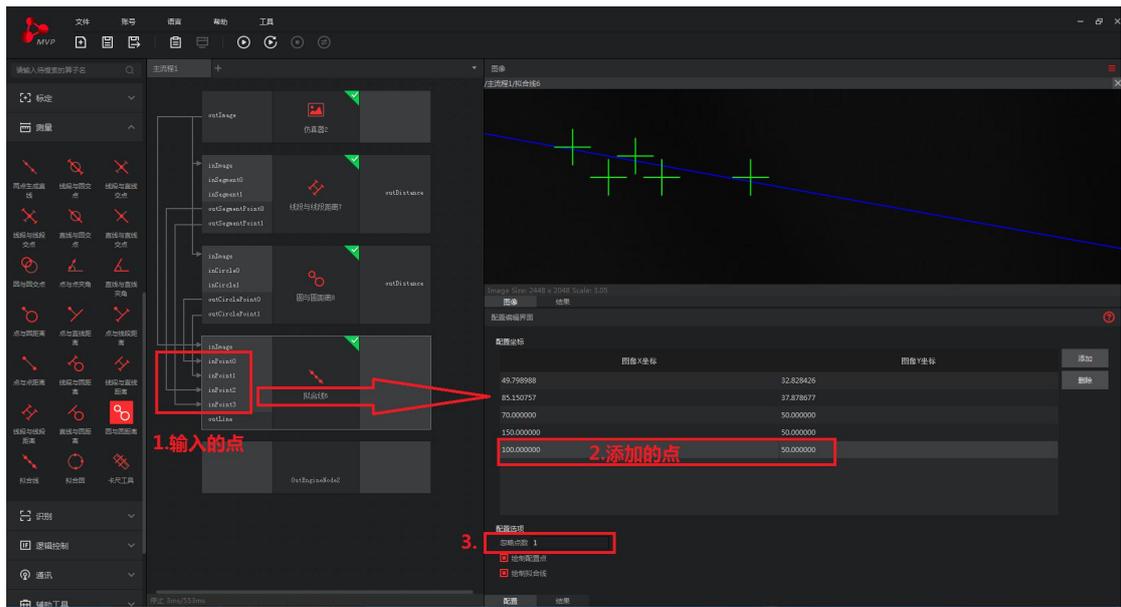


图 4.6.50 拟合线算子配置流程图

拟合线算子，添加仿真器或者相机作为图片输入，添加其他算子输出点作为拟合线算子的输入，按如图 4.6.50 主流程连线后单机运行，在右上方显示区域看到拟合的直线、输入的点，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称	数据类型	取值范围	默认值	说明
图像 X 坐标	float	[-999999, 999999]		输入用于拟合直线的点的 X 坐标
图像 Y 坐标	float	[-999999, 999999]		输入用于拟合直线的点的 Y 坐标
忽略点数	int32	[0,100]	0	忽略点的个数

## 4.6.20 拟合圆

拟合圆算子从输入或者配置接收十个以内点参数，根据接收参数和配置中设置的忽略点数来拟合圆并输出，可切换到结果界面查看输出信息（拟合圆、RMS 误差），如图 4.6.51 所示。



图 4.6.51 拟合圆算子结果图

## 操作步骤:

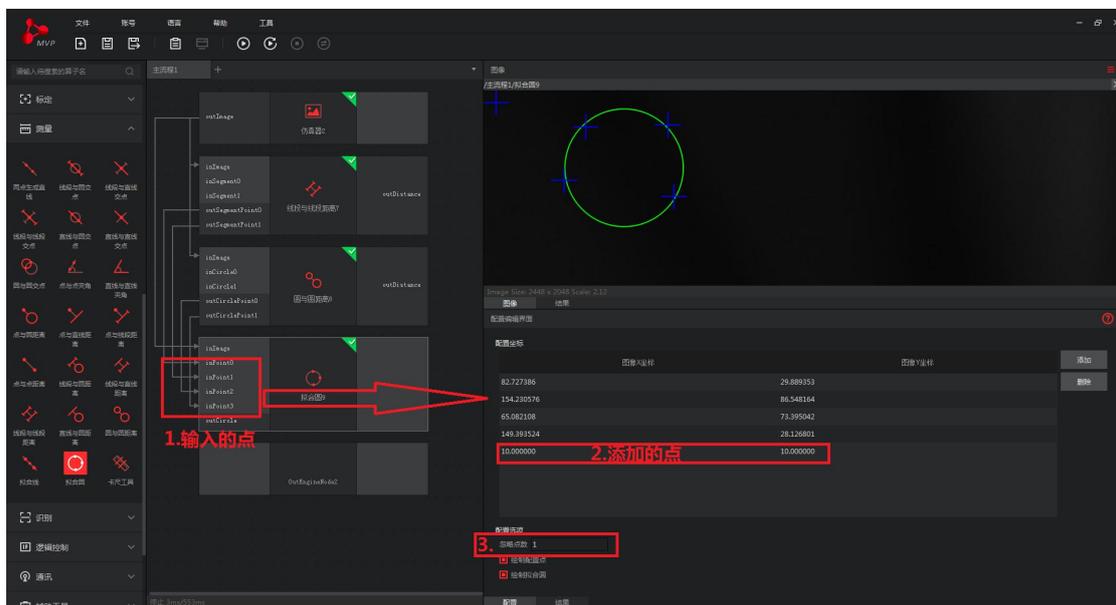


图 4.6.52 拟合圆算子配置流程图

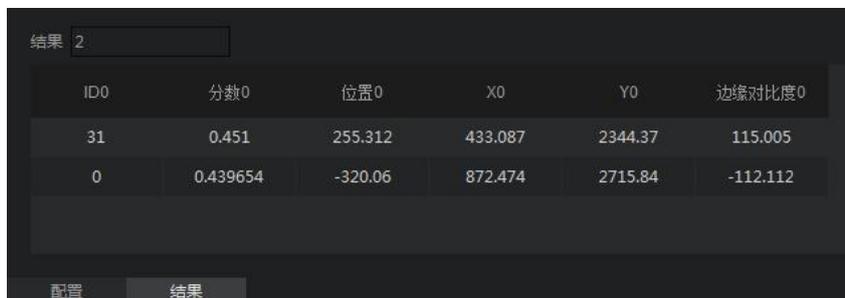
拟合圆算子，添加仿真器或者相机作为图片输入，添加其他算子输出点作为拟合圆算子的输入，按如图 4.6.52 主流程连线后单机运行，在右上方显示区域看到拟合的直线、输入的点，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

参数名称	数据类型	取值范围	默认值	说明
图像 X 坐标	float	[-999999, 999999]		输入用于拟合圆的点的 X 坐标
图像 Y 坐标	float	[-999999, 999999]		输入用于拟合圆的点的 Y 坐标
忽略点数	int32	[0,100]	0	忽略点的个数

### 4.6.21 卡尺工具

卡尺工具是一种测量目标对象的宽度、边缘的位置、特征或边缘对的位置和边缘对之间距离的视觉工具。卡尺工具在使用的时候，需要使用者提供期望测量的或者定位的大致区域、目标对象或是边缘的特性等。可切换到结果界面查看输出如图 4.6.53 所示：



ID0	分数0	位置0	X0	Y0	边缘对比度0
31	0.451	255.312	433.087	2344.37	115.005
0	0.439654	-320.06	872.474	2715.84	-112.112

图 4.6.53 卡尺工具结果图

操作步骤：

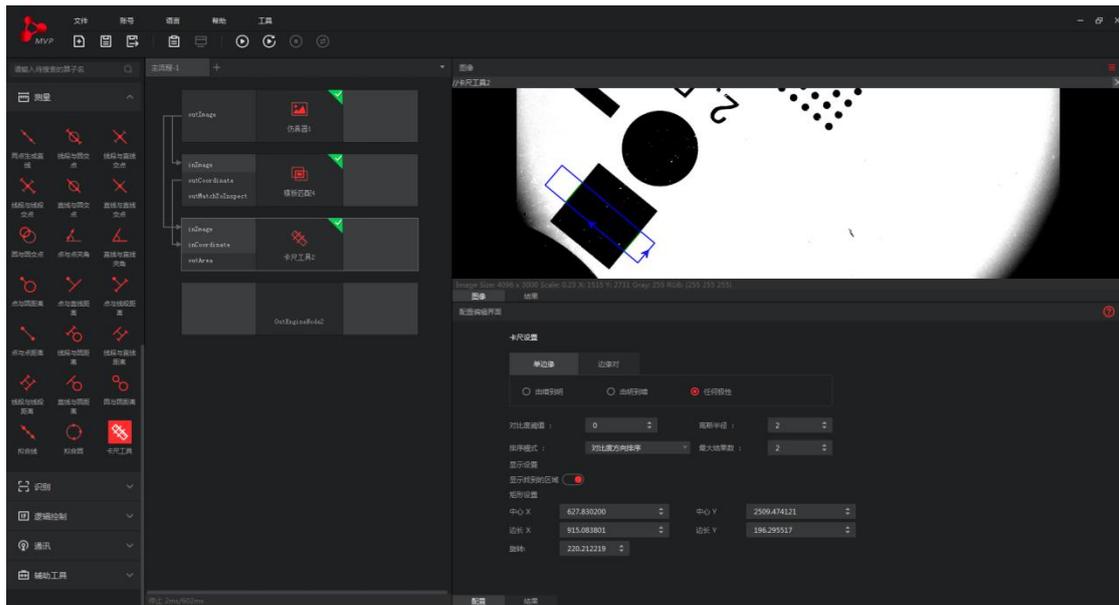


图 4.6.53 卡尺工具配置流程图

添加卡尺工具算子，添加仿真器或者相机作为图片输入，可添加模板匹配算子输出作为此算子输入，按如图 4.6.54 主流程连线后单机运行，在右上方显示区域看到矩形卡尺、找到的区域，在右下方区域进行配置参数设定（也可添加其他算子用起输出作为此算子的输入），达到需要的效果。

## 参数描述

区域设置参数如下：

参数名称	数据类型	取值范围	默认值	说明
矩形设置	中心 X	float	[-10000,10000]	矩形中心的 X 坐标
	中心 Y	float	[-10000,10000]	矩形中心的 Y 坐标
	边长 X	float	(0,10000]	短（长）边长
	边长 Y	float	(0,10000]	长（短）边长
	旋转	float	[-360,360]	顺时针旋转角度

## 设置基本参数

参数名称	数据类型	取值范围	默认值	说明
------	------	------	-----	----

边缘模式		-	单个边缘		查找单个边缘模式
		-	边缘对		查找边缘对模式
	边缘0极性		由暗到明 由明到暗 任何极性	由暗到明	
	边缘1极性		由暗到明 由明到暗 任何极性	由暗到明	
	边缘对宽度	Int	[0,99]	10	
对比度阈值		float	[0,255]		根据对比度筛选
过滤一半像素		float	[1,10000]		除噪、增强峰值
最大结果数		float	[0,10000]		选择最强的边缘数

## 详细介绍

- **边缘极性**：有由暗到明、由明到暗和任何极性三种选择
- **边缘对宽度**：期望输出的边缘对的像素间距尺寸
- **对比度阈值**：即梯度阈值，范围 0~255，只有边缘梯度阈值大于该值的边缘点才能被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，设置导致目标边缘点被筛除
- **过滤一半像素**：用于增强边缘和抑制噪声，最小值为 1。当边缘模糊或有噪声干扰时，增大该值有利于使检测结果更加稳定，但如果边缘与边缘之间挨得太近时反而会影响边缘位置的精度甚至丢失边缘，该值需根据实际情况设置

## 4.7 识别

### 4.7.1 条形码

条形码识别算法用于精确定位图片中一维码的位置并解析其包含的字符信息。算法平台支持解码类型包括 CODE128、EAN-13、CODE39、EAN-8、UPCA、UPCE、CODE93、ITF25、CODABAR。使用者可以根据需求对解码类型进行配置。

条形码的典型应用场景包括但不限于：

- 物流作业或其他场景中，对条形码的识别和解码：

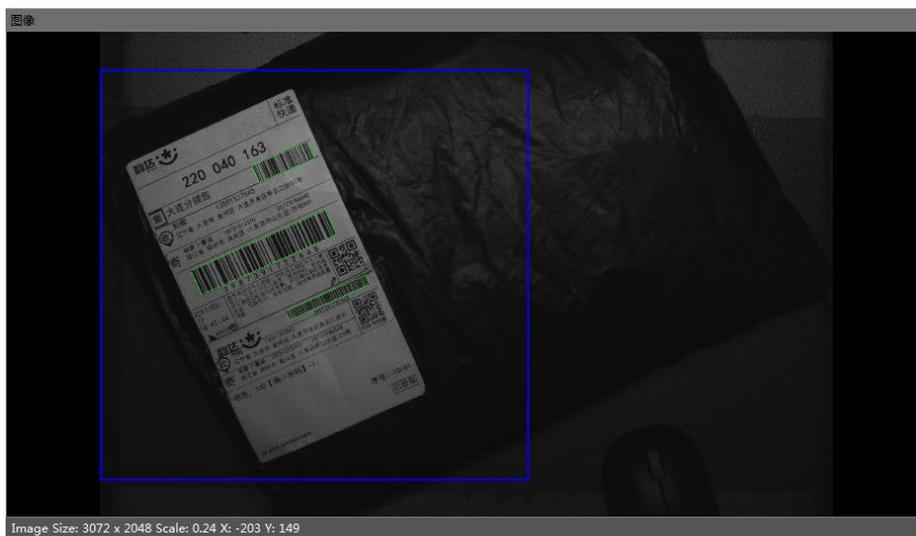


图 4.7.1. 物流单中的条形码定位

结果  成功数量

	类型	条码	长度	单模块像素	条码完整性	边界对比度	数字质量等级	字母质量等级
1	CODE128B_C...	J116200	7	2.72182	0.95	42.45	4	A
2	CODE128A_C...	3987391722665	14	1.88824	0.8	19.5294	4	A
3	CODE128C_C...	3987391722665	14	4.52182	1	54.55	4	A

图 4.7.2. 物流单中的条形码解码（与上图对应）

### 操作流程：

### 配置参数

表 4.7.1 配置参数说明

参数名称	数据	取值范	默认值	说明
------	----	-----	-----	----

	类型	围		
条码数量 CodeNum	int32_t	[0, 100]	5	待识别的条码数量
条码类型 Code Type			所有类型都开启	条码类型数组,在数组中存入需要识别的条码类型。默认识别所有支持的条码类型。
checksumCode39			关	Code39 是否进行校验开关。
checksumITF25			关	ITF25 是否进行校验开关。
质量评价 Quality Evaluation			关	条码打印质量评价开关

## 详细介绍

- 条码数量  
算法输出的条码个数<=所设置的条码数量。  
假设相机视野中有 10 个码，条码数量配置 5，结果输出 5 个码；  
假设相机视野中有 1 个码，条码数量配置 5，结果输出 1 个码；  
条码数量过大会增加算法耗时，建议根据实际使用场景配置。
- 条码类型  
支持解码类型包括 CODE128、EAN-13、CODE39、EAN-8、UPCA、UPCE、CODE93、ITF25、CODABAR，默认全部开启；  
使用时建议根据实际情况勾选对应的条码类型，过多的条码类型会增加算法耗时。
- 质量评价  
打开时结果会增加输出条码质量一栏。

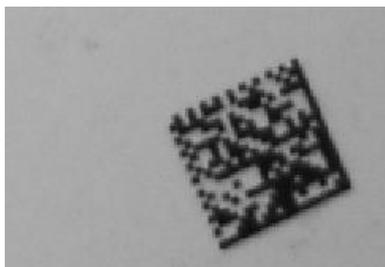
## 输出参数

参数名称	数据类型	说明
结果 Result	Bool	检测结果
成功数量 SuccessNum	int	检测成功数量
类型	String	条码类型

Type		
条码 Code	String	条码内容
长度 Len	Int	条码长度
单模块像素 Ppm	Float	条码单模块像素
条码完整性 symbolIntegrity	Float	条码完整性
边界对比度 edgeContrast	Float	条码边界对比度
数字质量等级 nPrintQualityGrade	Int	条码中数字质量等级
字母质量等级 aPrintQualityGrade	String	条码中字母质量等级
outBarCodeInfo		所有条码的详细信息
outCount		检测到条码数量

## 4.7.2 二维码

二维码识别算法会对图像中的二维码进行定位并解码。算法平台软件支持解码类型包括：DM 码、QR 码，分别对应如下：



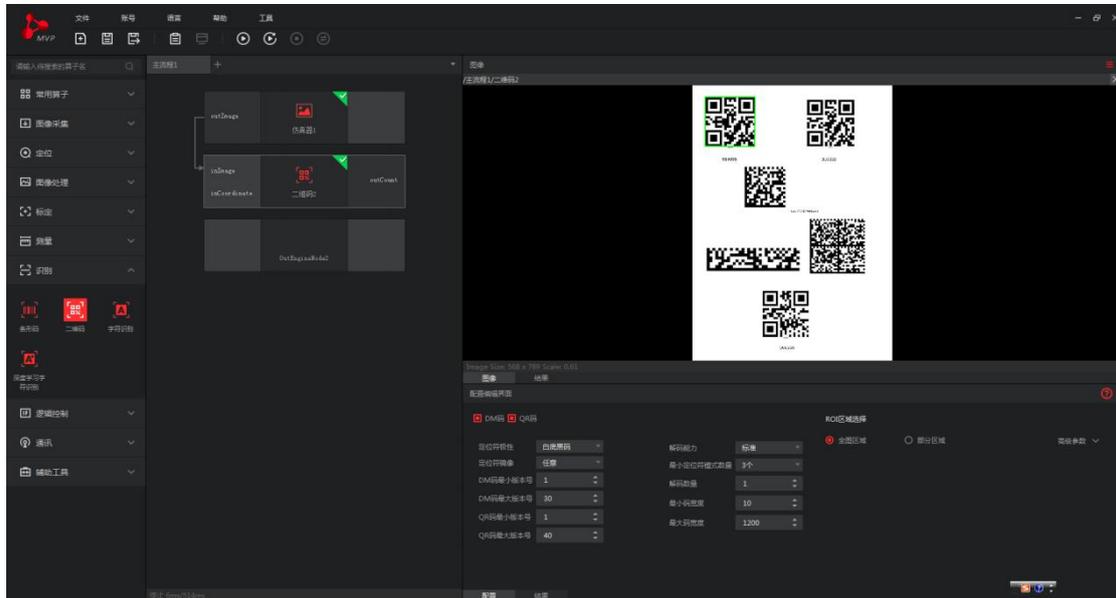
(DM)



(QR)

图 7.2.1 二维码识别图

操作流程：



配置参数

表 4.7.1 配置参数说明

参数名称	数据类型	取值范围	默认值	说明
二维码类型 codeType		DM/QR	全选	二维码类型,支持 DM、QR 码
定位符极性 signPolarity		白底黑码/黑底白码/任意	白底黑码	极性标志
定位符镜像 signMirror		任意/无镜像/带镜像	任意	镜像标志,根据实际图像是否存在镜像码进行配置
解码能力 decoderCapacity		标准/增强/最强	标准	解码能力
解码个数 decodeNum	int32_t	[1,50]	1	解出设定个数后停止算法,范围:1~50,默认值为1
DM 最小版本 versionMinDM	int32_t	[1,30]	1	DM 解码最小版本号,范围:1~30,默认值为30
DM 最大版本 versionMaxDM	int32_t	[1,30]	30	DM 解码最大版本号,范围:1~30,默认值为30
QR 最小版本 versionMinQR	int32_t	[1,40]	1	QR 解码最小版本号,范围:1~40,默认值为1

QR 最大版本 versionMaxQR	int32_t	[1,40]	40	QR 解码最大版本号，范围：1~40，默认值为 40
最小码宽度 codeMinWidth	int32_t	[10,1200]	10	像素尺寸，范围：10~1200，默认值为 10
最大码宽度 codeMaxWidth	int32_t	[10,1200]	1200	像素尺寸，范围：10~1200，默认值为 1200

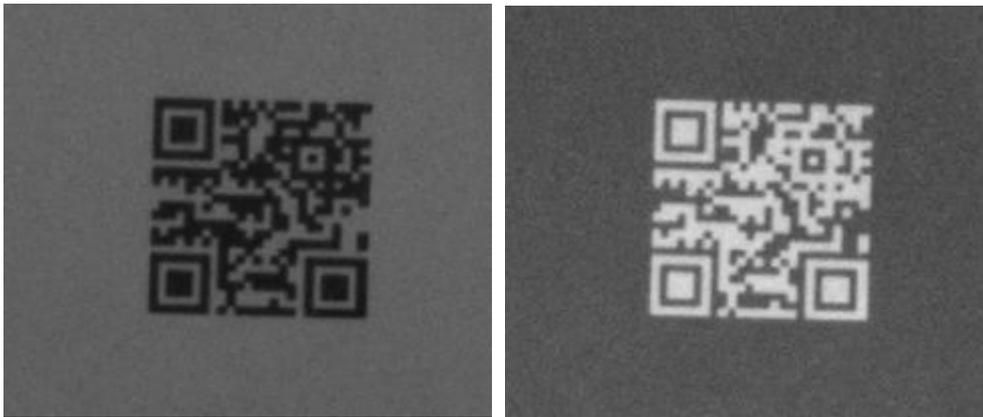
## 详细介绍

- 二维码类型

支持 DM 码和 QR 码，默认全部开启；

使用时建议根据实际情况勾选对应的二维码类型，全选时会增加算法耗时。

- 定位符极性

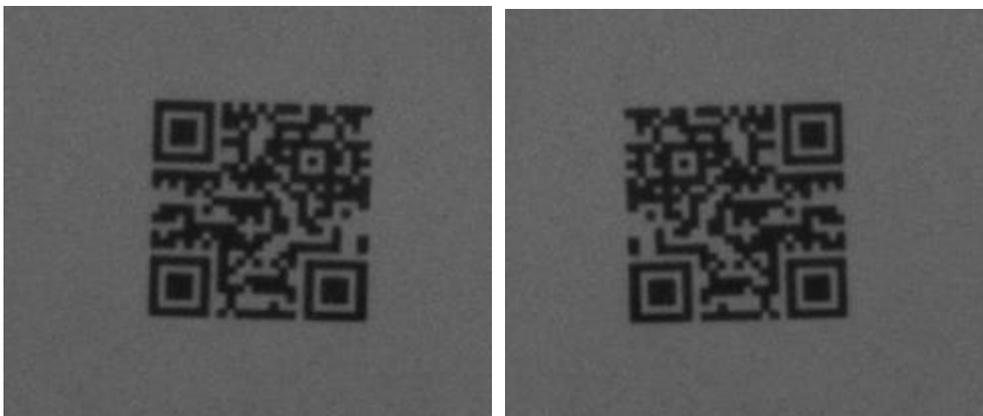


白底黑码

黑底白码

使用时建议根据实际情况选择，选择任意时会增加算法耗时。

- 定位符镜像



无镜像

带镜像

使用时建议根据实际情况选择，选择任意时会增加算法耗时。

- **解码能力**

标准/增强/最强，能力越强越耗时，一般情况下标准就够用。

- **解码个数**

算法输出的二维码个数<=所设置的二维码数量。

假设相机视野中有 10 个码，条码数量配置 5，结果输出 5 个码；

假设相机视野中有 1 个码，条码数量配置 5，结果输出 1 个码；

二维码数量过大会增加算法耗时，建议根据实际使用场景配置。

- **DM 最小版本/DM 最大版本**

默认情况下，算法开启最大范围，1-30；

在确定输入码版本的情况下，缩小该范围可减少算法耗时。

- **QR 最小版本/QR 最大版本**

默认情况下，算法开启最大范围，1-40；

在确定输入码版本的情况下，缩小该范围可减少算法耗时。

- **最小码宽度/最大码宽度**

码宽度的单位是像素，及二维码的单边长在图像中所占的像素；

根据实际情况适当缩小这个范围，可减少算法耗时。

## 输出参数

参数名称	数据类型	说明
结果 Result	Bool	检测结果
成功数量 SuccessNum	int	检测成功数量
码的类型 codeType	String	二维码类型
角度 Angle	Float	二维码旋转角度

质量 Quality	Int	二维码的质量
二维码文本 CodeText	String	二维码内容
二维码长度 CodeLen	Int	二维码长度
outDataCodeInfo		所有二维码详细信息
outCount		检测到二维码数量

### 4.7.3 字符识别

主要包括 OCR 及 OCV 功能：

- **OCR 算法包括**

**文本行定位：**文本基于整张图像的自动文本行定位算法，可以自动识别文本行个数，同时也可以配合模板定位只选择单行文本进行识别；

**字库训练：**支持在线训练字符模板，同时支持读取系统字体训练模板。

**字符识别：**基于模板匹配和神经网络的字符识别方法，支持英文、数字、符号、及中文字符识别，

- **OCV 算法包括**

根据用户设定正则表达式得到用户自定义输出内容；字符质量的评价。

示例：

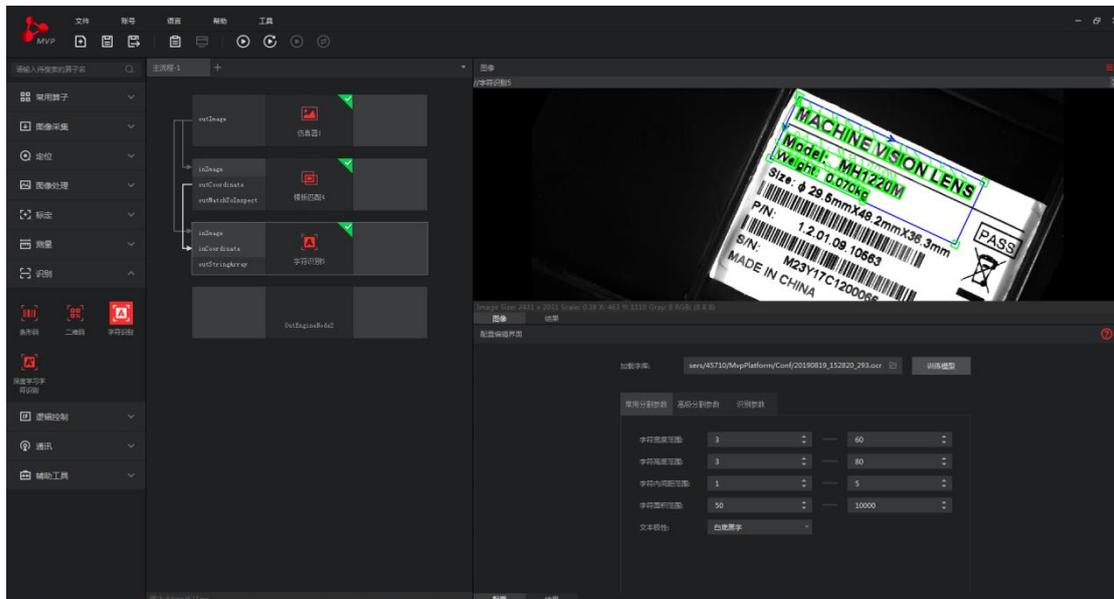
文本行自动定位：



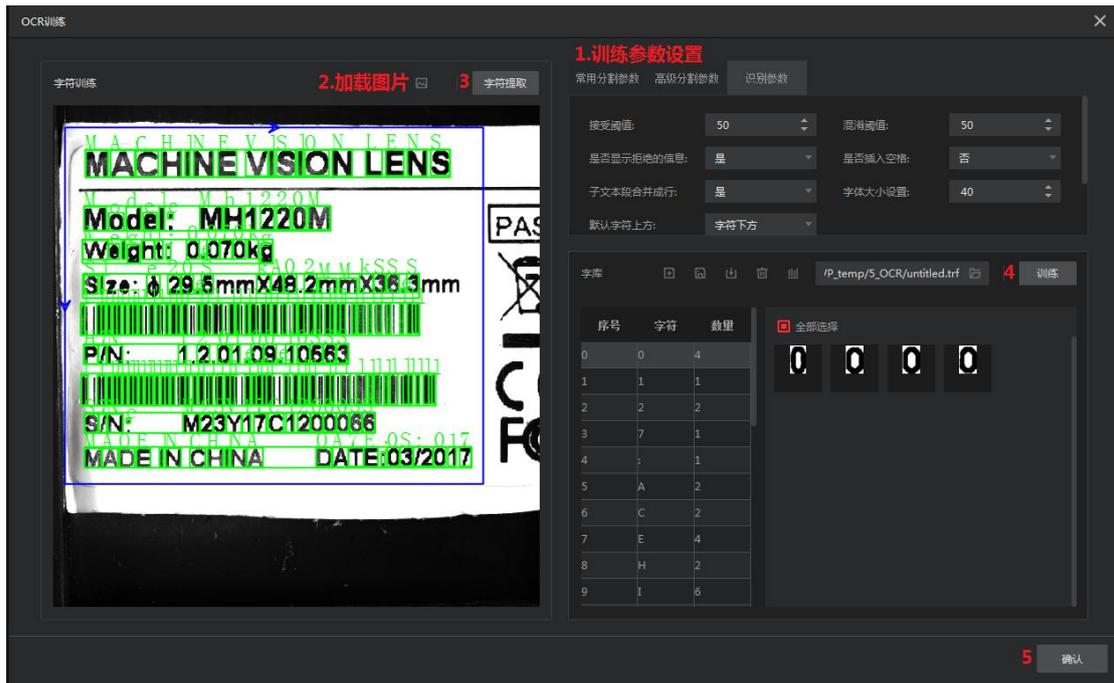
字符识别:



操作流程:



模板训练:



## 配置参数

常用分割参数	数据类型	取值范围	默认值	说明
segmentPolarity 字符区域分割极性	int32_t	1,-1	1	字符极性配置： 1：白底黑字 -1：黑底白字
maxWidth 字符最大宽度	int32_t	999	100	最大字符宽度
minWidth 字符最小宽度	int32_t	999	3	最小字符宽度
maxHeight 字符最大高度	int32_t	999	100	最大字符高度
minHeight 字符最小高度	int32_t	999	20	最小字符高度
maxIntraGap 最大字符内间距	int32_t	(1, 999)	5	最大内间距像素个数

## MVP 算法平台用户手册

<b>minIntraGap</b> 最小字符内间距	int32_t	(1, 999)	1	最小内间距像素个数
<b>maxCharSize</b> 最大字符尺寸	int32_t	(1, 99999)	8000	最大字符尺寸
<b>minCharSize</b> 最小字符尺寸	int32_t	(1, 99999)	20	最小字符尺寸

高级分割参数	数据类型	取值范围	默认值	说明
文本滤波	int32_t	0,1,2	0	待分割文本噪声程度： 0: 无噪声使用原图 1: 噪声一般 2: 噪声严重
fragMergeMode 字符融合	int32_t	0,1	0	片段融合模式： 0: 不合并； 1: 合并
Overlap 字符是否有重叠	int32_t	0,1	0	像素是否有粘连重合： 0: 不粘连； 1: 粘连
distanceMainLine 字符偏离主线距离	int32_t	(0, 999)	5	与主线的最大偏离距离
minAspectRatio 字符最小高宽百分比	int32_t	(1, 100)	10	最小字符高宽占比百分比
angleRange 文本倾斜角度	int32_t	(0, 45)	0	文本倾斜角度范围
skewRange 字符倾斜角度	int32_t	(0, 45)	0	字符倾斜角度范围

maxStrokeWid 最大字符笔画宽度	int32_t	(1, 999)	25	最大笔画宽度
minStrokeWid 最小字符笔画宽度	int32_t	(1, 999)	2	最小笔画宽度
SegmentMode 文本分割模式	int32_t	(0, 1)	0	0: 局部阈值 1: 自适应阈值
字符印刷模式	int32_t	(0, 1)	0	0: 常规印刷体 1: 其他
字符碎片尺寸	int32_t	(0, 999)	5	
忽略边缘字符	int32_t	(0, 1)	1	去除边缘 blob
最小对比度	int32_t	(0, 255)	3	对比度参数
字符分割模式	int32_t	(0, 1)	1	0: 固定宽度 1: 自适应

识别参数	数据类型	取值范围	默认值	说明
接受阈值	int32_t	(0, 100)	50	得分阈值
混淆阈值	int32_t	(0, 100)	50	相似字符混淆阈值
是否插入空格	int32_t	0, 1	0	自适应插入空格
之文本段合并	int32_t	0, 1	0	之文本合并成行
字符个数范围	int32_t	(0, 100)	100	字符个数范围
字符外间距范围	int32_t	(1, 999)	5	外间距范围
字符显示设置	int32_t		上方	显示配置

#### 4.7.4 深度学习字符识别

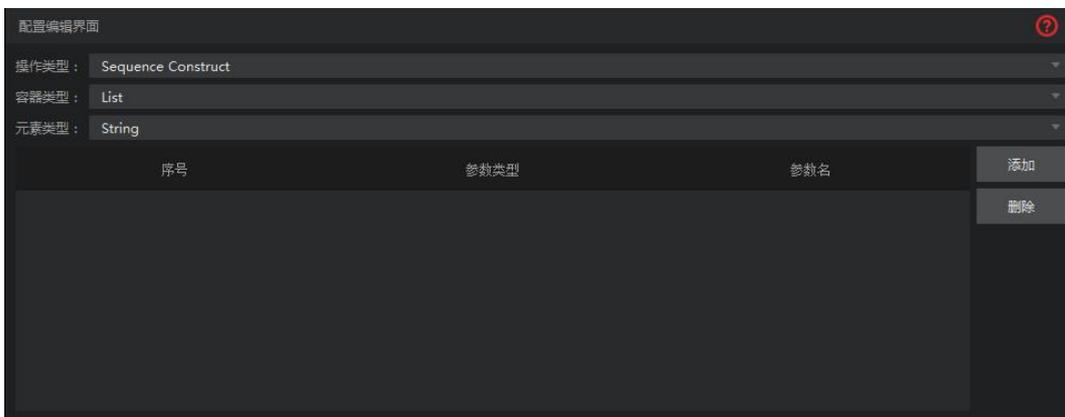
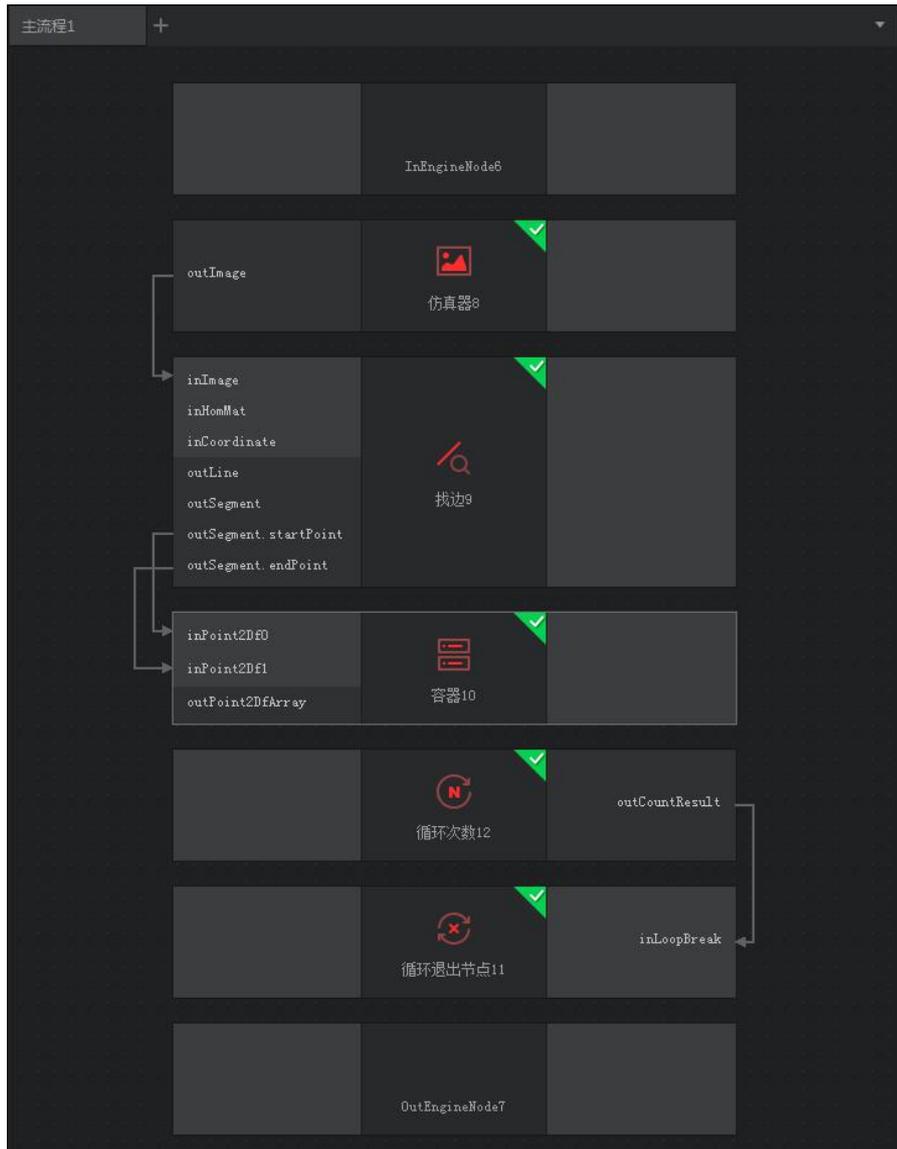
### 4.8 逻辑控制

## 4.8.1 容器

容器顾名思义，和编程中的容器有同样的含义，算法平台的容器算子有两大功能，一个是将数据加入一个容器中，一个是将数据从一个容器中取出，容器目前支持所有算法平台包含的数据类型。

在选择“操作类型”，“容器类型”，“元素类型”后，添加一定数量的输入，然后按照下图去配置，点击运行。（截图所在的流程编辑界面是在循环模块内）





则输出数组中会添加相应的输入参数，如下图所示：

结果

outPoint2DfArray	
outPoint2Df[0]	
x	1058.550659
y	1295.655273
outPoint2Df[1]	
x	1712.205200
y	1296.391113
outPoint2Df[2]	
x	1058.550659
y	1295.655273
outPoint2Df[3]	
x	1712.205200
y	1296.391113

容器算子的配置界面如下图所示：

**操作类型：**操作类型选项有 Sequence Construct, Loop Construct, Access 三个类型。

**Sequence Construct：**无论容器是否在循环模块内，每次容器内包含数据的大小均是配置界面中添加的数据类型数量。

**Loop Construct：**如果容器在循环模块内，则容器内的数据会做累计叠加（如上图例子，输入参数的数量是两个点，循环次数是 2 次，则在每一次点击运行后，容器中始终包含 4 个数据）

**Access：**列表的数据读取，在 Access 模式下，算子有两个输入，一个是列表，一个是索引，输出一个类型数据。一次运行后，会根据传入的索引，从列表中取出对应索引的数据，在输出参数中做输出。

容器类型仅支持 List 类型。

元素类型：在下拉列表中包含算法平台目前支持所有的数据类型。

添加：添加按钮，在点击添加按钮后，在左侧的算子上，会新增相应的输入参数，如下图所示：

配置编辑界面

操作类型： Sequence Construct

容器类型： List

元素类型： Point2Df

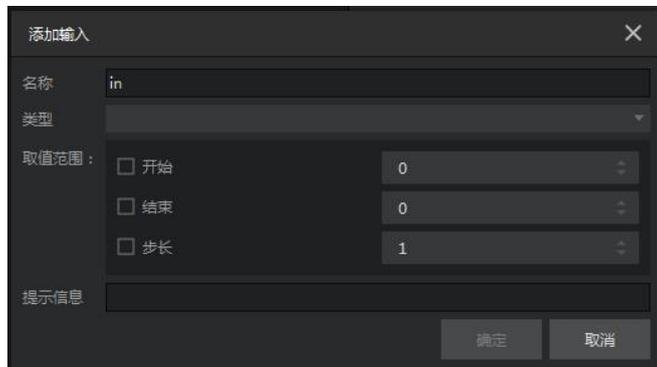
序号	参数类型	参数名	添加
0	Point2Df	inPoint2Df0	删除
1	Point2Df	inPoint2Df1	



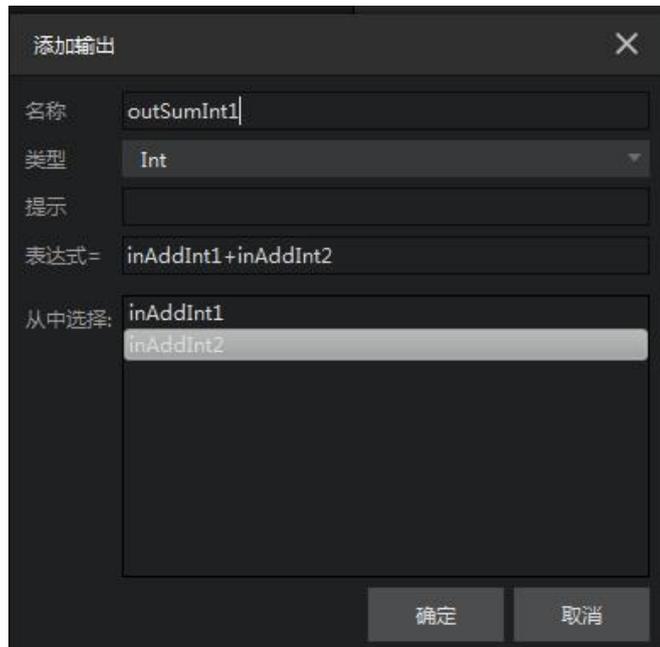
## 4.8.2 数学表达式

数学表达式算子可以自己创建输入和输出参数，具体的参数类型为算法平台目前支持的数据类型，然后其可以通过输入的值，进行运算，将运算结果赋值给输出结果。

添加输入参数：

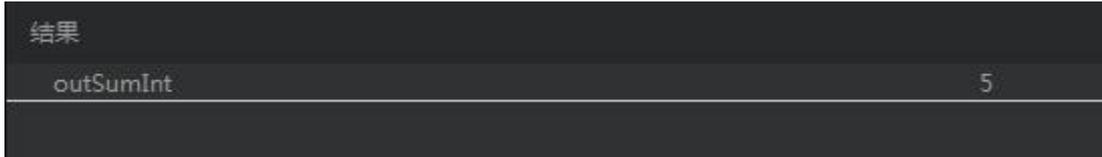


添加输出参数：





上述配置会将两个输入参数 `inAddInt1` 和 `inAddInt2` 做加法，然后将值输出到输出参数 `outSumInt` 中。



## 4.8.3 顺序模块

顺序模块，在该模块中添加的算子，会顺序执行。

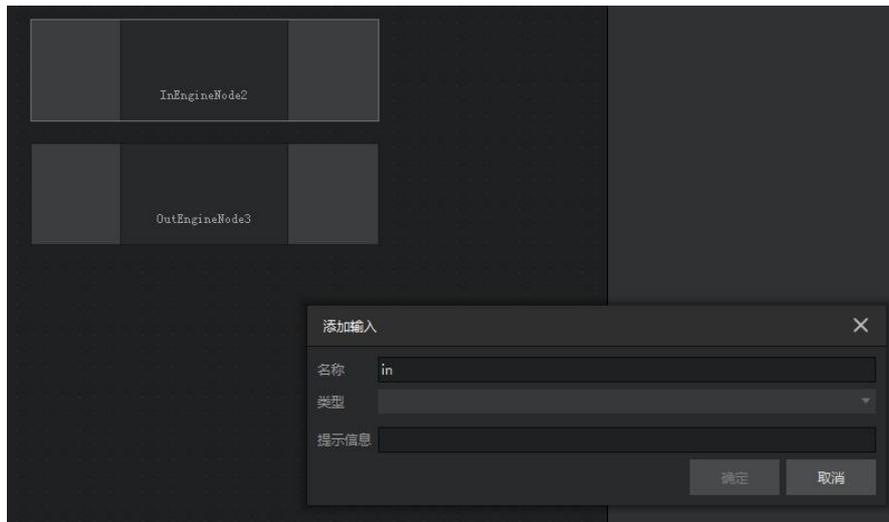


若按照上图进行配置工程，则执行顺序是：先执行“仿真器”，再执行“顺序模块”内的算子，等“顺序模块”内的算子都执行完了以后，会跳出顺序模块，再执行“两点生成直线”算子。

顺序模块算子支持将模块内算子的数据输出到模块外，或者是将数据从模块外传入到模块内，具体方式如下：

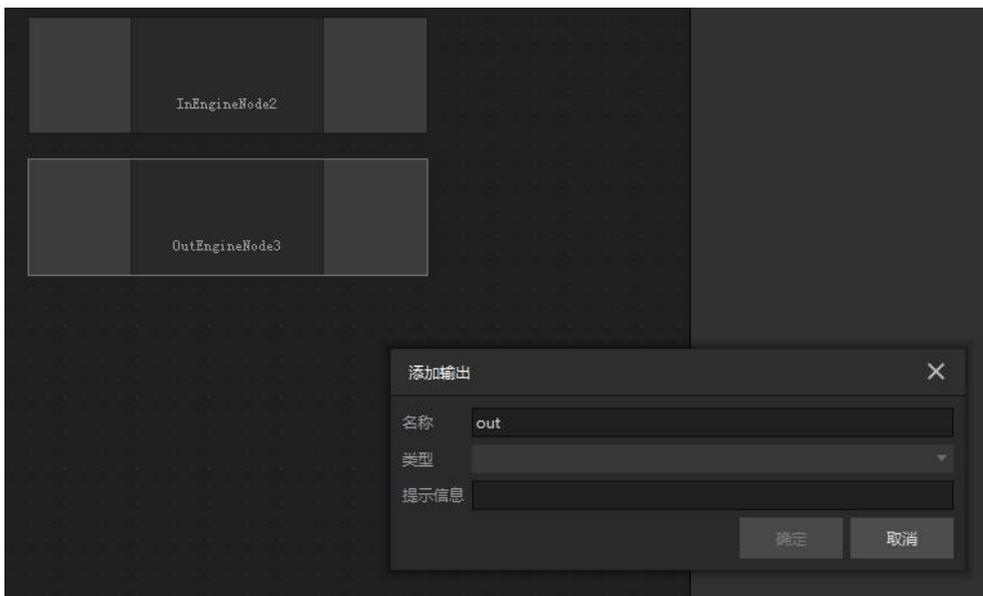
### 模块外数据传输模块内：

双击进入模块内，右键点击输入引擎节点，选择“添加输入”，填入“名称”“类型”后点击确定，在模块外就会显示对应类型的输入。



### 模块内数据传输到模块外：

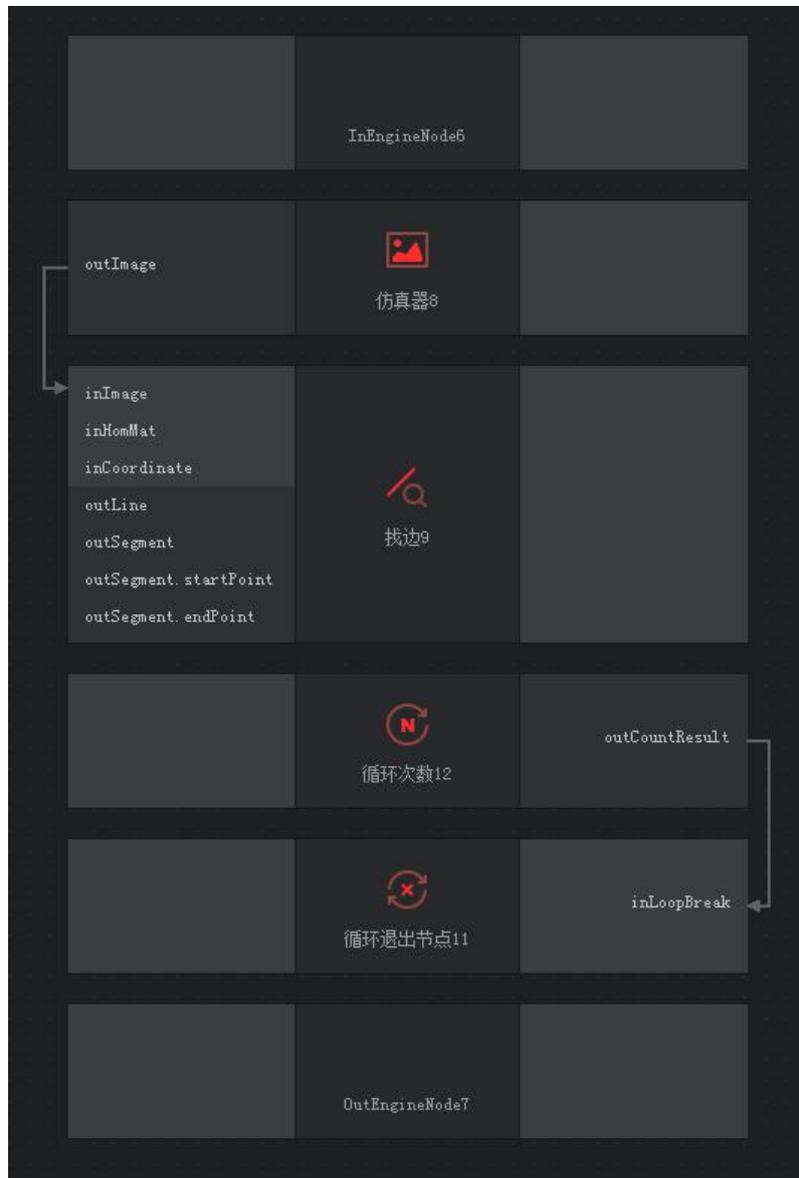
双击进入模块内，右键点击输入引擎节点，选择“添加输出”，填入“名称”“类型”后点击确定，在模块外就会显示对应类型的输出。



## 4.8.4 循环模块

循环模块也是属于模块算子的一种。当工程进入循环模块以后，如果没有“循环退出节点”，则工程会一直在循环模块内部一直运行。双击算子后，可以进入算子内部，右键输入和输出引擎节点可以增加输入参数和输出参数，将动态参数从模块外部传入模块内部，或者将动态参数从模块内部传出模块。

下图中有一个例子工程，“循环次数”配置成2次，输出连接到“循环退出节点”算子，当运行至循环模块内部时，同时运行到第二次，“循环次数”算子会输出 true，当“循环退出节点”的输入得到 true 时，会使得运行跳出循环模块，则此时循环模块的一次运行得以终止。



## 4.8.5 循环退出节点

“循环退出节点”是一个特殊的算子，其内部无法添加其他算子，其实一个独立的算子，而且其只能添加到“循环模块”内部，用作使运行退出“循环模块”去使用。因为循环模块无法自动退出，靠着该算子在满足条件时退出循环模块，工程配置的流程如上一个例子所示。其算子配置界面如下图：



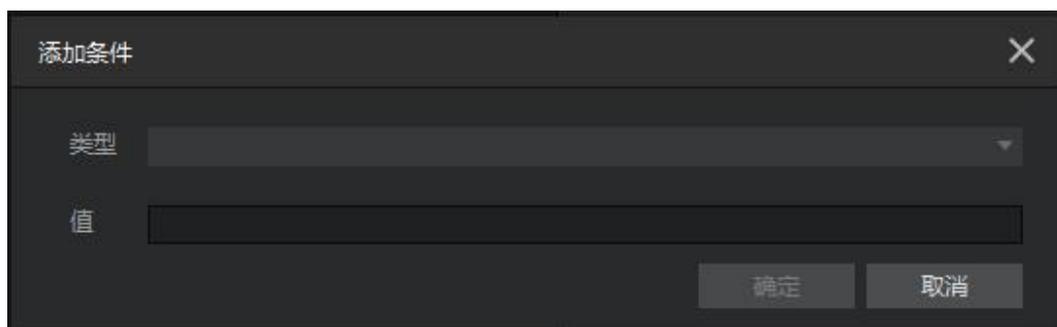
默认为 `false`，为 `false` 时，当“循环退出节点”的输入是 `false` 时，会跳出当前的“循环模块”。

当设置为 `true` 时，当“循环退出节点”的输入是 `true` 时，会跳出当前的“循环模块”。

## 4.8.6 分支节点

节点类似编程语言中的 `switch` 语句，可以创建多个分支条件，当输入参数满足某一个分支条件时，会进入某一个特定模块，执行相应的操作。

双击分支节点，会弹出下面对话框，选择条件分支参数类型，同时填写参数等于什么值时，进入该分支。填写完以后点击确定。即可生成一个条件分支。



当需要添加条件分支时，点击主流程下方的下拉列表，点击其中的“添加条件分支”，在弹出框内继续填写和之前参数类型相同的参数，即可添加对应的条件分支。



当条件分支创建以后，在“分支节点”上会产生 **Condition** 的输入参数，根据输入参数的不同，会进入不同的条件分支。对分支内的算子进行处理。

## 4.8.7 多任务同步模块

步模块其作用是为其模块内的每一个算子新开一个线程去运行，保证模块下的每一个算子同步运行，和顺序模块和循环模块的算子结构是一样的，内部可以添加算子，同时可以添加外部输入的参数和内部输出的参数。

在多任务同步模块中添加的每一个算子，在点击运行之后，都会启动一个线程去处理算子，所以通常会配合顺序模块去使用，如下图所示。



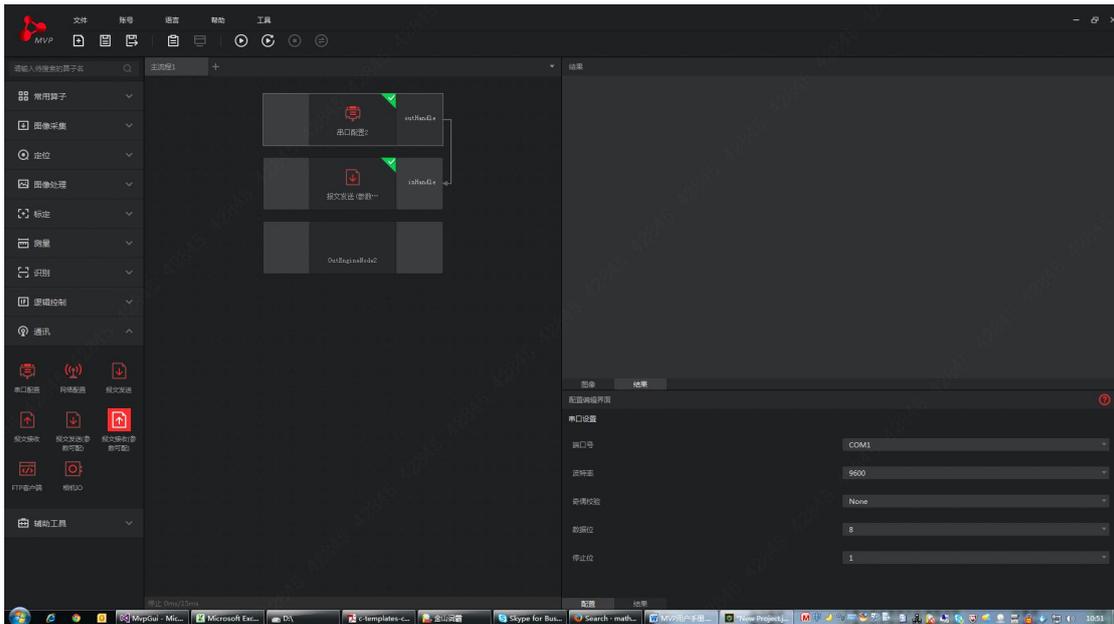
在多任务同步模块添加两个顺序模块，在顺序模块中添加需要添加的算子，点击运行后，两个顺序模块因为也是属于算子，所以将会在不同的线程中一起运行，保证了两个顺序模块同步多线程工作。

## 4.9 通讯

### 4.9.1 串口配置

串口配置用于打开并设置计算机串口，并将得到的句柄即编号传给报文发送和接收类算子。上述两泪算子利用串口配置算子打开的串口实现报文的发送和解析。串口配置必须与发送或接收类算子联合使用，否则没有意义。

操作流程：



添加串口配置和报文发送(参数可配)算子，outHandle 与报文发送的 inHandle 连接。调整端口号，波特率，奇偶校验，数据位，停止位到设定值，点击运行一次即可发送内容，对于接收操作类似。

配置参数

参数名称	数据类型	取值范围	默认值	说明
端口号 PortName	enum	COM1~Com10	COM1	计算机端口编号
BaudRate 波特率	enum	1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200	9600	每秒钟传输速度

奇偶校验 Parity	enum	None,Odd,Even	None	选择串口通讯的校验方式
数据位 DataBits	Enum	5~8	8	报文中发送数据所占用的位数
停止位 StopBits	Enum	1,1.5,2	1	报文中停止位所占用的位数

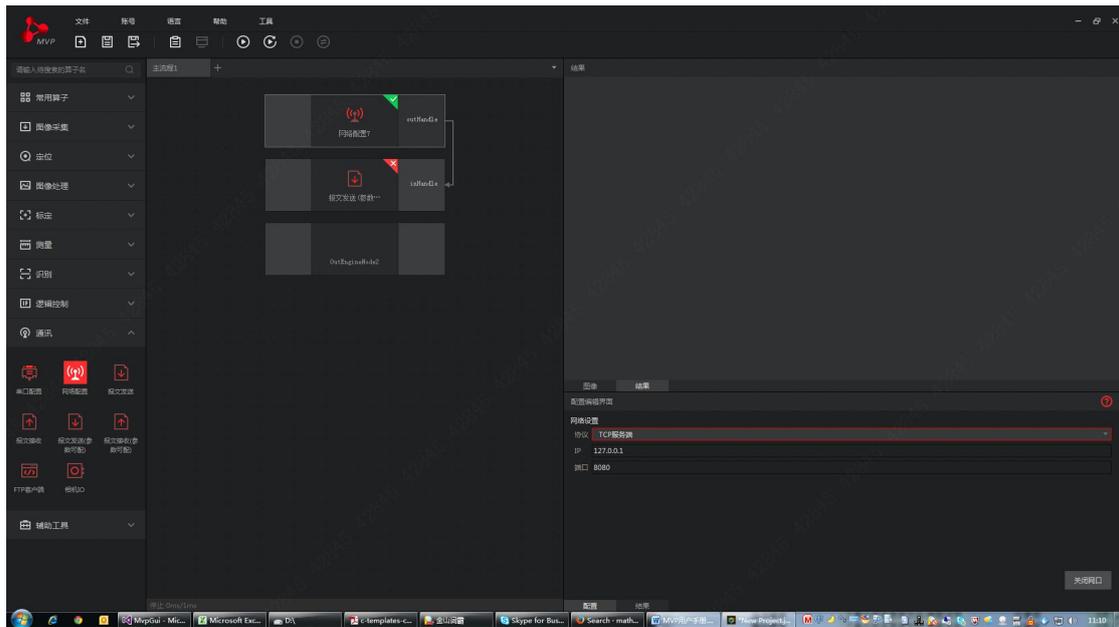
## 输出参数

参数名称	数据类型	取值范围	默认值	说明
输出句柄 outHandle	int	--	--	表示串口的句柄，本质是一个编号

## 4.9.2 网络配置

网络配置用于打开并设置计算机网络，并将得到的句柄即编号传给报文发送和接收类算子。上述两泪算子利用网络配置算子打开的网口实现报文的发送和解析。网络配置必须与发送或接收类算子联合使用，否则没有意义。

## 操作流程



添加网络配置和报文发送(参数可配)算子, outHandle 与报文发送的 inHandle 连接。调整协议, IP 地址和端口号到设定的值, 运行一次。报文接受与之类似。

## 配置参数

参数名称	数据类型	取值范围	默认值	说明
协议 Protocol	enum	Server,Client	Server	使用客户端还是服务器
IP 地址 IP Address	String	0.0.0.0~255.255.255.255	127.0.0.1	IP 地址
端口号 Port	Int	0~99999	8080	网络的端口号

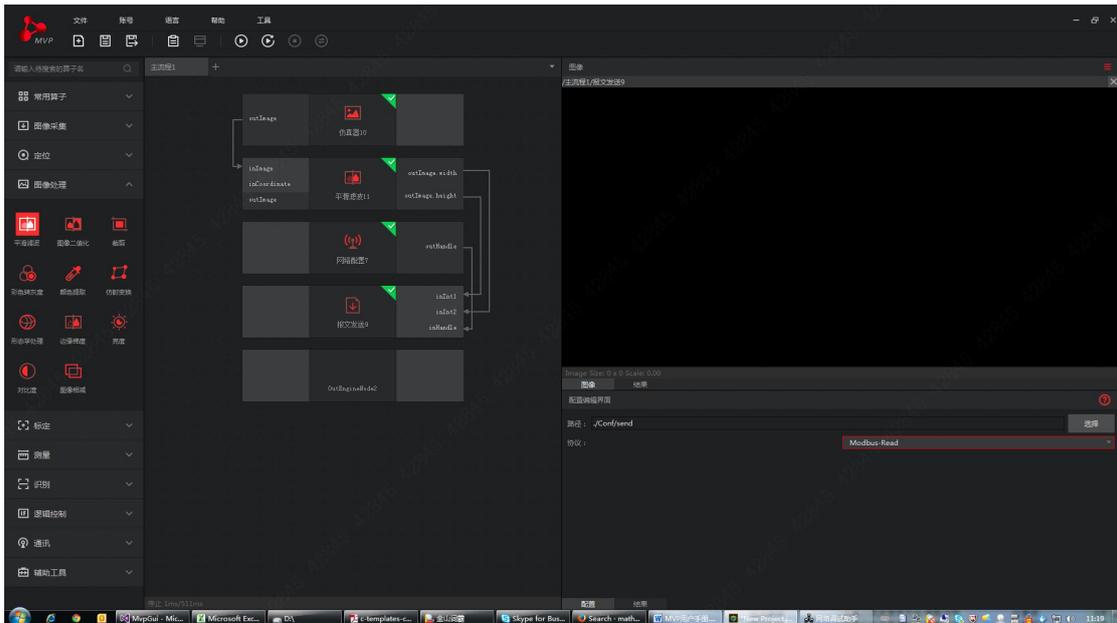
## 输出参数

参数名称	数据类型	取值范围	默认值	说明
输出句柄 outHandle	int	--	--	表示网口的句柄, 本质是一个编号

### 4.9.3 报文发送

报文发送利用 Python 脚本将发送内容组合成报文并将其通过串口或网络发送给其他设备或软件。

#### 操作流程



首先应该编辑脚本，将发送的值组合成报文的形式；其次选择该脚本；再次将需要输出的参数和网络配置或串口配置的句柄连接到该脚本，最后点击运行即可发送数据。

#### 配置参数

参数名称	数据类型	取值范围	默认值	说明
路径 Path	String	--	./Conf/send	Python 脚本的路径
协议 Protocol	Enum	--	--1	选择脚本

#### 脚本格式

```
import struct
import random
def getInputParamList(str):
    ret = {}
    ret['inInt1'] = 'Int'
    ret['inInt2'] = 'Int'
    return repr(ret).replace("'", '')

#50 00 00 00 FF FF 03 00 00 14 00 10 00 01 14 00 00 64 00 00 A8 02 00 BC 19 83 0
def getBody(str):
    obj = eval(str)
    xxx = obj['inInt1']
    yyy = obj['inInt2']

    TSid =random.randint(4097,65535)
    ProID = 0x0000
    Len = 0x0006
    UnitID = 0x01
    FuncCode = 0x03
    RefNum = 0x0001
    WordCount = 0x0002

    sendText = struct.Struct('>HHBHH')
    ccc = sendText.pack(TSid,ProID,Len,UnitID,FuncCode,RefNum,WordCount)
    return len(ccc), ccc
# getBody('{\"inInt1\":20,\"inInt2\":30}')
```

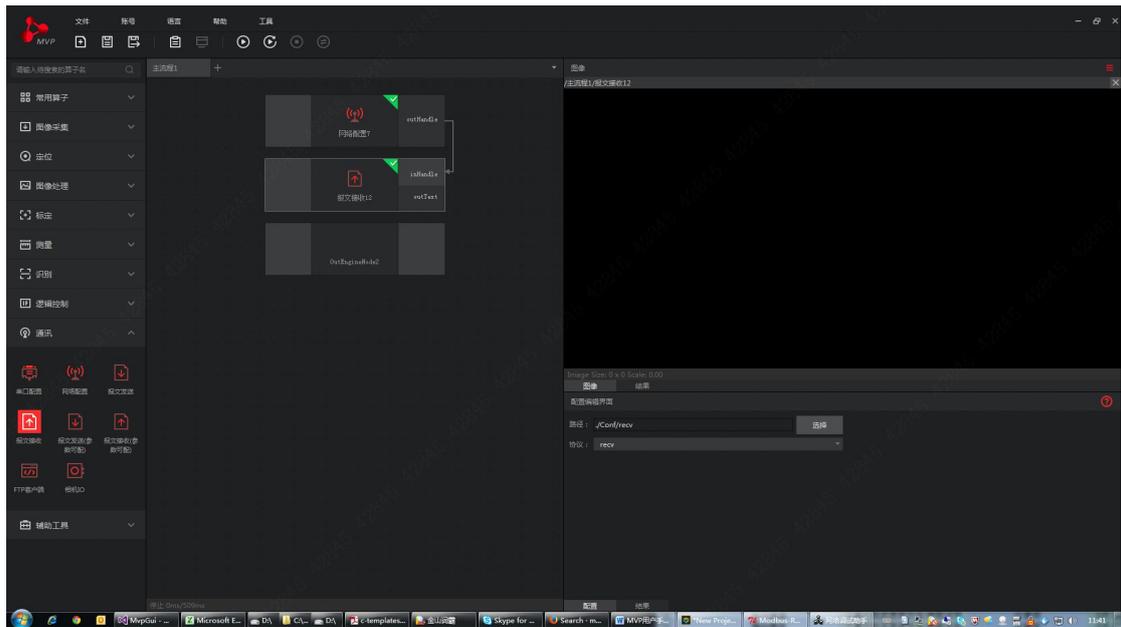
其中 `getInputRaramList(str)` 为配置输入内容，其中 `inInt1` 和 `inInt2` 位输入名称，类型为 `Int`。

`getBody(str)` 为报文组合接口。

#### 4.9.4 报文接收

报文发送利用 Python 脚本将发送内容组合成报文并将其通过串口或网络发送给其他设备或软件。

#### 操作流程



首先应该编辑脚本，将接受的报文解析成值；其次选择该脚本；再次将需要输出的参数和网络配置或串口配置的句柄连接到该脚本，最后点击运行即可发送数据。

## 配置参数

参数名称	数据类型	取值范围	默认值	说明
路径 Path	String	--	./Conf/send	Python 脚本的路径
协议 Protocol	Enum	--	--1	选择脚本

## 脚本格式

```

def getoutputParamList(str):
    ret = {}
    ret['outText'] = 'Real'
    return repr(ret).replace("'", '"')

import struct

def getRsp(str):
    pack = struct.Struct('>f')
    (ccc,) = pack.unpack(str)
    obj = {}
    obj['outText'] = ccc
    return repr(obj).replace("'", '"')

```

其中 `getoutputParamList(str)` 为配置输入内容，其中 `outText` 位输入名称，类型为 `Real`（对应 C++ `float`）。

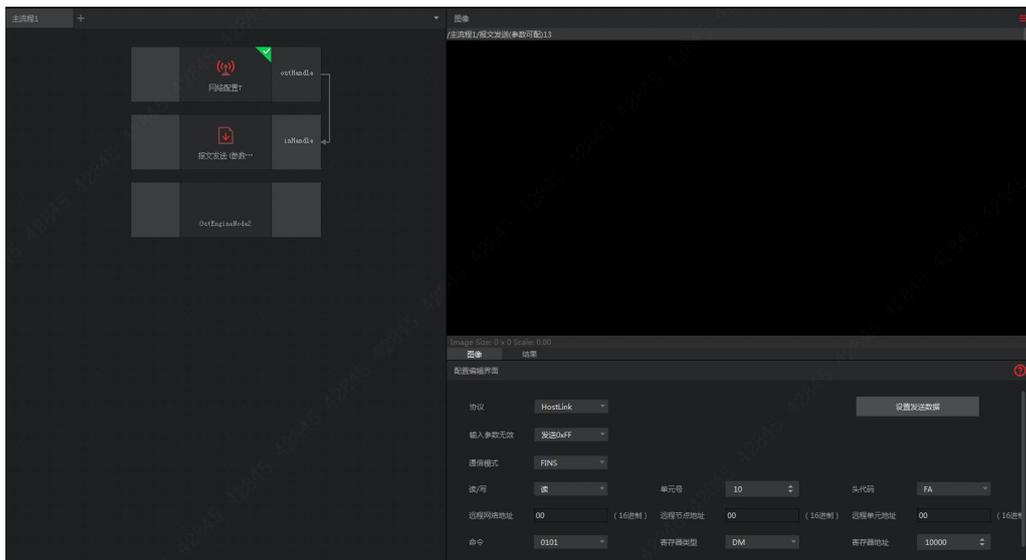
`getRsp(str)` 为报文组合接口。

## 4.9.5 报文发送（参数可配）

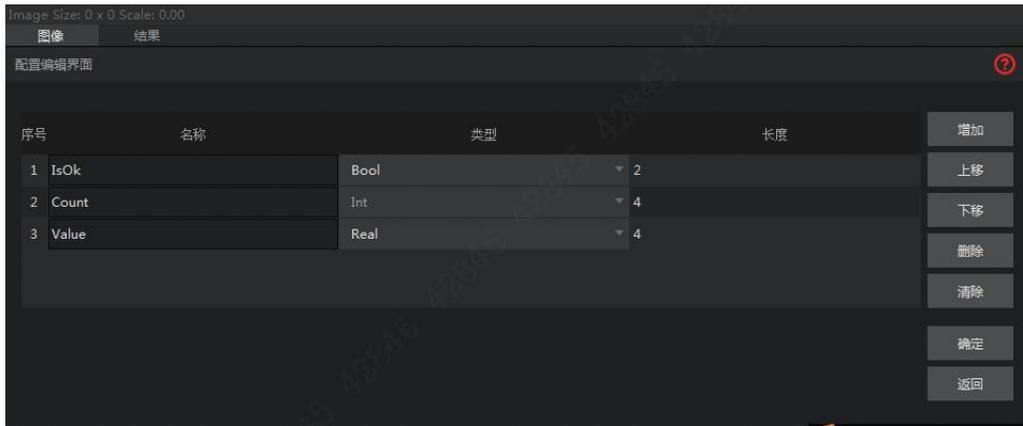
报文发送（参数可配）通过预先定义的协议格式发送数据到外部设备或外部软件。

### 操作流程

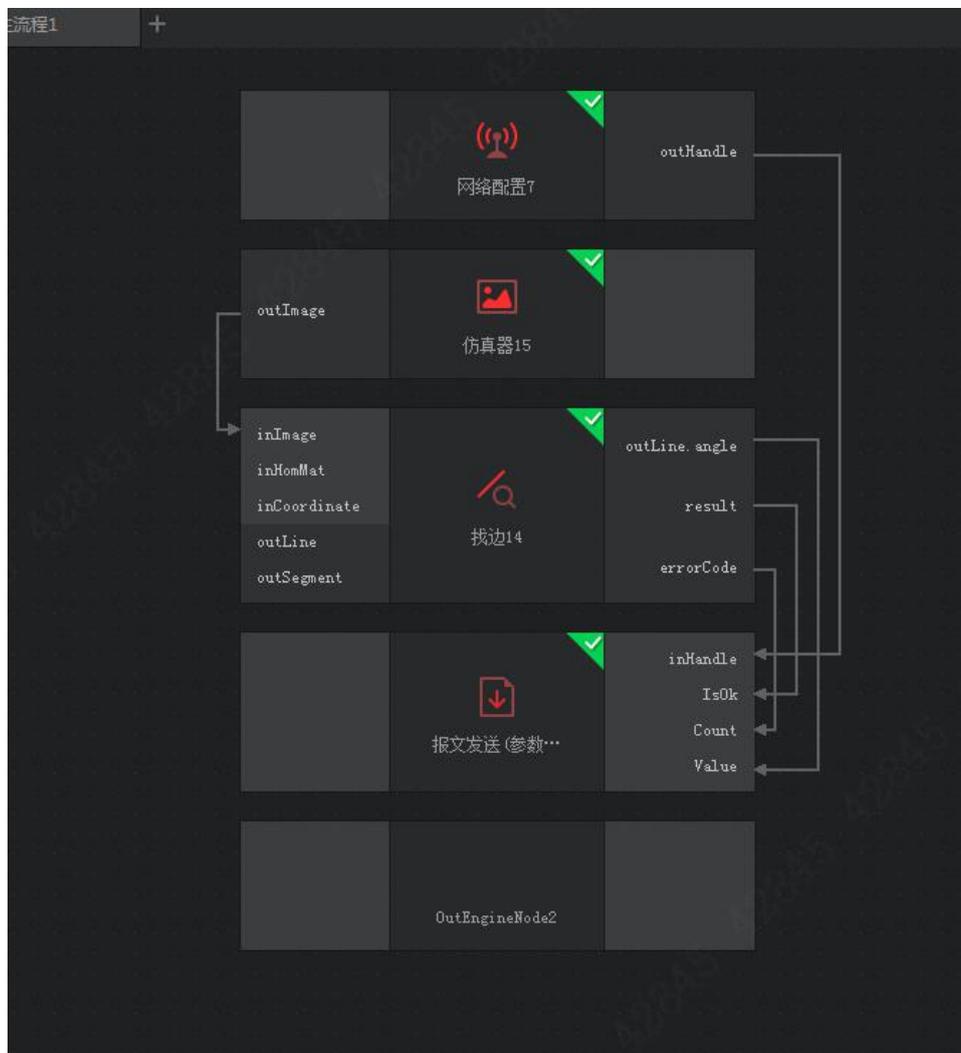
1. 正常添加网络配置或串口配置和报文发送（参数可配）算子并将 `outHandle` 连接到 `inHandle`。如图\*\*编号\*\*所示。



2. 点击配置发送数据，在新对话框中添加输出参数，并配置其类型名称后点击确定完成。



3. 添需要输出参数的算子，并将输出的参数对应到发送算子的输入。



完成设置后，算子会将数据利用发送算子配置的格式按照用户在第二步中设定的顺序组成一条报文发送给其他设备或软件。

## 配置参数

参数名称	数据类型	取值范围	默认值	说明
协议 Protocol	Enum	HostLink, MCProtocol, Modbus, Custom	HostLink	选择使用的协议
输入参数无效 Invalid input	Enum	发送 0xff 发送 0x00 不发送	发送 0xff	当输入参数无效或无输入参数时，发送该值作为占位

## 详细介绍

- HostLink ，该协议用于欧姆龙公司 PLC 的通讯，分为 CMode 和 FINS 两种工作模式，分别对应串口和网口。具体协议请参考欧姆龙官方文档“SYSMAC CS/CJ/CP Series SYSMAC One NSJ Seriea Communication Commands Reference Manual”。目前本软件实现了其中 0101 和 0102 两种指令，并且只支持 W 和 DW 两种寄存器。
- MCProtocol 即 MC 协议，该协议用于与三菱 PLC 的通讯，分别支持了 3E, 4E, 3C, 4C, SLMP 五种工作方式的 ASCII 和 Bin 两种形式的 0401 和 1401 两个指令对 D 和 M 两种寄存器的操作。具体设置数值请参考三菱官方文档“MELSEC 通讯协议参考手册”和“MELSEC iQ-F FX5 用户手册（SLMP 篇）”。
- Modbus 该协议被大部分 PLC 支持，目前支持 03 和 16 两个功能码的 TCP 和 RTU 两种模式。
- Custom 属于用户自定义协议。

参数名称	数据类型	取值范围	默认值	说明
字节次序 Endian	Enum	高位优先, 低位优先	高位 优先	高位优先即大端低位优先 即小端
发送数据 Send Packet	Enum	自定义输入, 外部输入	自定 义输	当使用自定义输入时，输出的是文本框的内容；否则输

			入	出算子输入内容
按十六进制发送 Send as Hex	Bool	True, false	False	是否使用 16 进制
报文头 Head Code	Hex Value	0000~FFFF	0001	自定义协议的报文头
报文尾 End Code	Hex Value	0000~FFFF	0001	自定义协议的报文尾

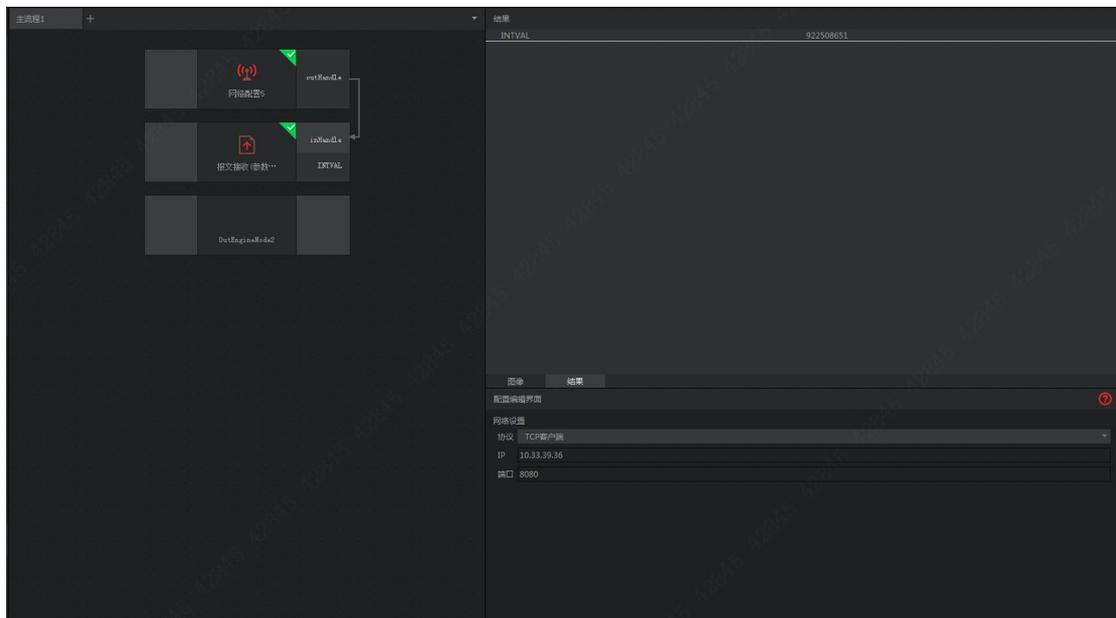
当发送数据采用外部输入时，可以设定报文头和报文尾。

## 4.9.6 报文接收（参数可配）

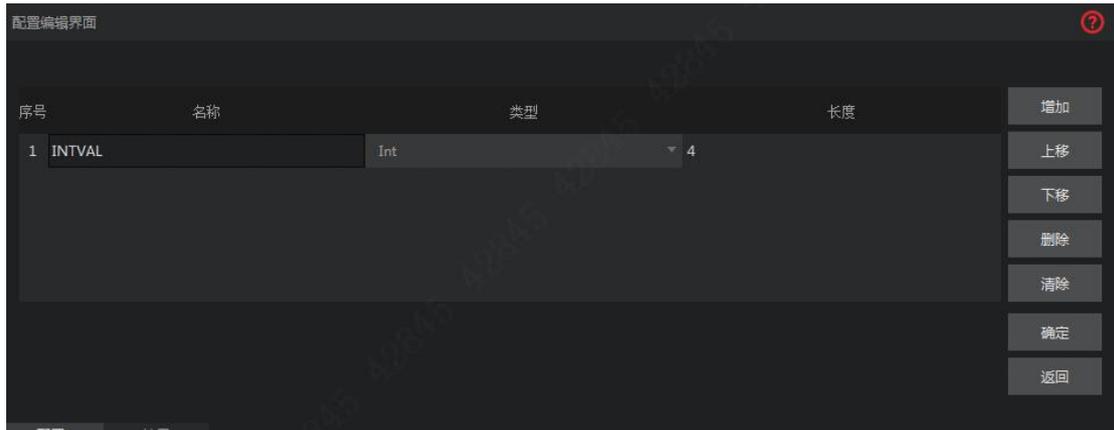
报文接收（参数可配）通过预先定义的协议格式将接收到的报文解析为特定的值并输出给其他算子使用。

### 操作流程

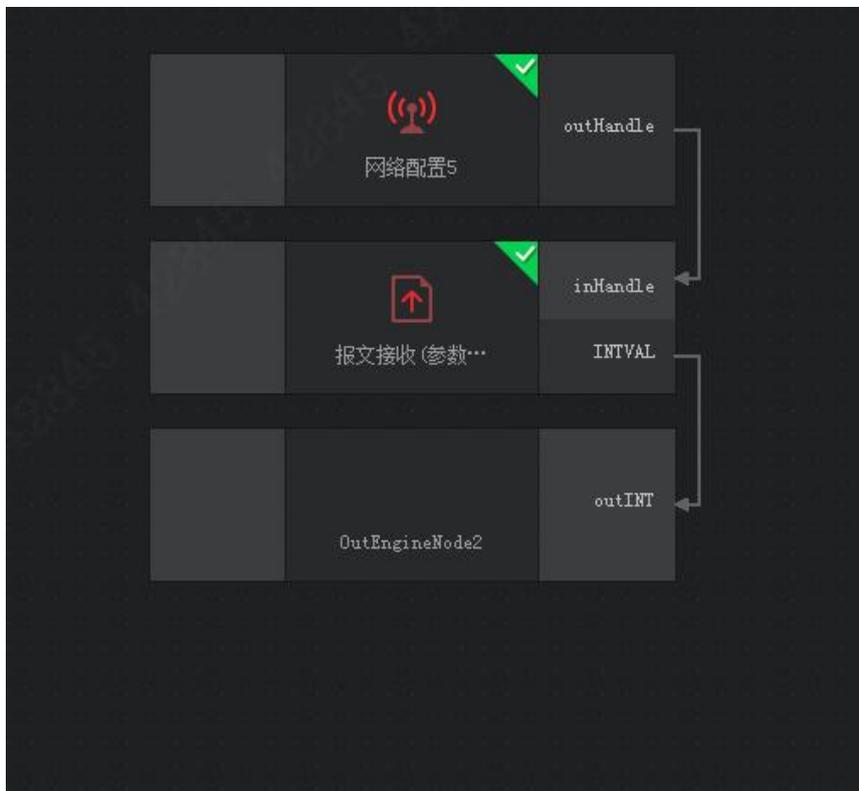
1. 正常添加网络配置或串口配置和报文接收（参数可配）算子并将 outHandle 连接到 inHandle。如图\*\*编号\*\*所示。



2. 点击配置接受数据，在新对话框中添加输出参数，并配置其类型名称后点击确定完成



3. 添需要输出参数的算子，并将输出的参数对应到接收算子的输出



完成设置后，算子会将数据利用接收算子配置的格式按照用户在第二步中设定的顺序解析报文。

## 配置参数

参数名称	数据类型	取值范围	默认值	说明
协议	Enum	HostLink,	HostLink	选择使用的协议

Protocol		MCProtocol, Modbus, Custom		
报文头 Head Code	Hex Value	0000~FFFF	0001	自定义协议的报文头
报文尾 End Code	Hex Value	0000~FFFF	0001	自定义协议的报文尾

## 详细介绍

- HostLink ，该协议用于欧姆龙公司 PLC 的通讯，分为 CMode 和 FINS 两种工作模式，分别对应串口和网口。具体协议请参考欧姆龙官方文档“SYSMAC CS/CJ/CP Series SYSMAC One NSJ Seriea Communication Commands Reference Manual”。目前本软件实现了其中 0101 和 0102 两种指令的返回指令，并且只支持 W 和 DW 两种寄存器。
- MCProtocol 即 MC 协议，该协议用于与三菱 PLC 的通讯，分别支持了 3E, 4E, 3C, 4C, SLMP 五种工作方式的 ASCII 和 Bin 两种形式的 0401 和 1401 两个指令对 D 和 M 两种寄存器的操作的返回指令。具体设置数值请参考三菱官方文档“MELSEC 通讯协议参考手册”和“MELSEC iQ-F FX5 用户手册（SLMP 篇）”
- Modbus 该协议被大部分 PLC 支持，目前支持 03 和 16 两个功能码的 TCP 和 RTU 两种模式的返回指令
- Custom 属于用户自定义协议

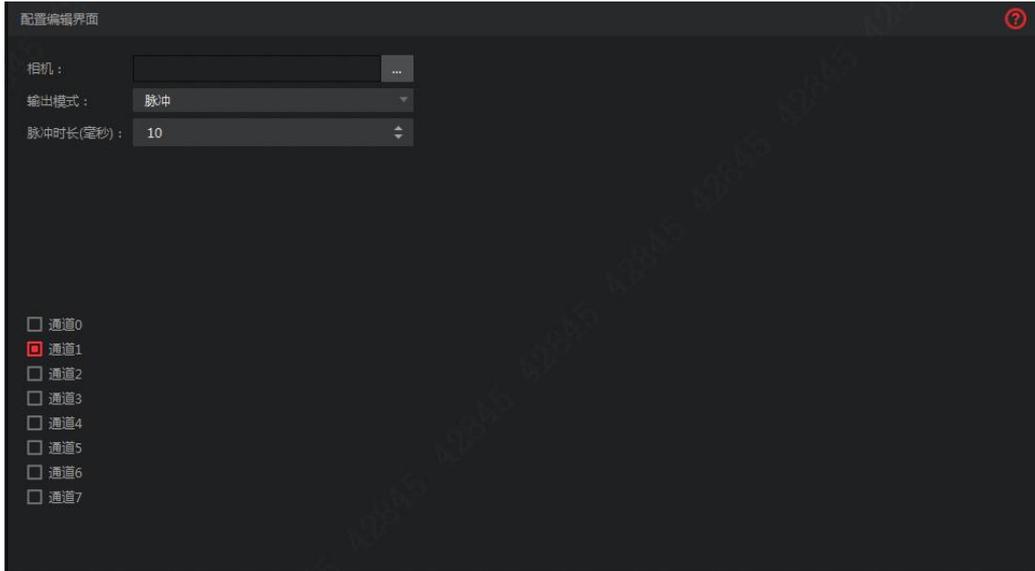
参数名称	数据类型	取值范围	默认值	说明
字节次序 Endian	Enum	高位优先, 低位优先	高位 优先	高位优先即大端低位优先 即小端
十六进制	Bool	True, false	False	是否使用 16 进制

## 4.9.7 相机 IO

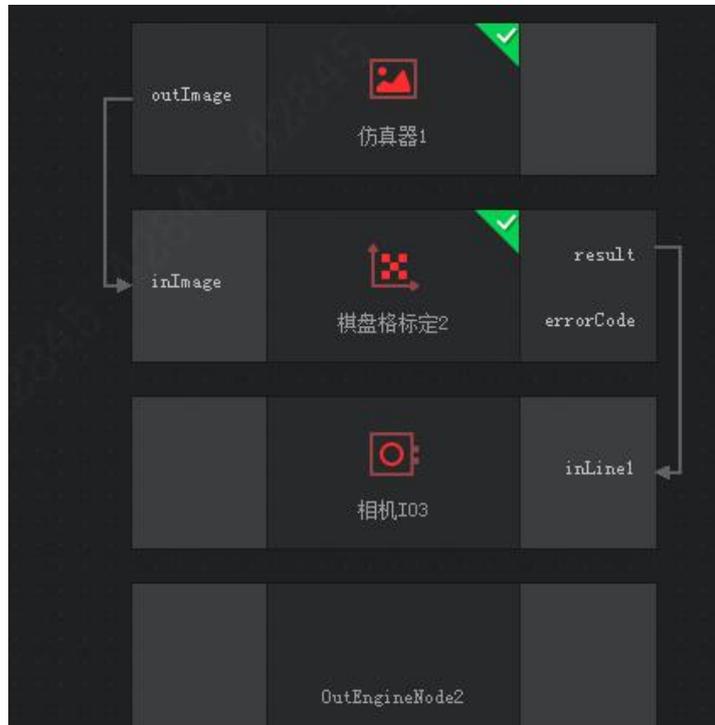
相机 IO 用于通过相机直接输出 bool 变量。

## 操作步骤

1. 点击相机“...”按钮，在弹出对话框中选择相机并设置相机其中 1 个或多个相机 IO 口为输出后退出。
2. 在相机 IO 的配置界面选择对应的 IO 通道，确定输出模式和脉冲时长，当模式为持续时不需要设置时长



3. 完成设置后将其他算子的 Bool 型输出连接到本算子即可完成输出



## 配置参数

参数名称	数据类型	取值范围	默认值	说明
相机	--	--	--	选择输出的相机
输出模式	Enum	脉冲, 持续	脉冲	输出电平是否自动拉回
脉冲时长	Int	0-200	10	电平持续时间
通道	--	通道 0~通道 10	无	选择哪个相机通道输出

## 4.10 辅助工具

## 4.10.1 保存图片

自由选择路径保存算法处理后的图片，主要将 `inImage` 参数与带有 `outImage` 参数的算子连线以将算子处理结果图片保存到目标路径下。

算子参数说明

## a. 配置参数

保存图片配置参数主要包含对保存图片的格式设置以及对保存图片路径的选择。

表 4.10.1 保存图片配置参数表

参数名称	数据类型	说明
<code>type</code>	-	保存图片格式，支持 BMP, JPG, PNG 格式
<code>directory</code>	-	选择保存图片路径

## b. 输出参数

输出参数主要包含算子读取输入图片的保存路径。输出参数可以直接拖拉到结果界面显示。

表 4.10.2 保存图片输出参数表

参数名称	数据类型	说明
costTime	int	运行时长
errorCode	int	错误识别码
outPath	-	输出保存图片路径
result	bool	保存图片结果

## 操作流程

### a. 流程编辑

如图 4.10.1 所示，保存图片算子输出参数 `inImage` 连接其他算子 `outImage` 参数，读取图片。输出参数可以显示保存的图像路径以及保存结果。



图 4.10.1 保存图片流程编辑图

### b. 配置参数编辑

配置编辑界面	
type	BMP
directory	/imageStorage/

图 4.10.2 保存图片配置参数编辑界面

## 4.10.2 循环次数

循环次数算子在配置中设置循环次数，当运行次数达到设置的次数时，循环次数输出“true”。一般可以配合循环模块使用，设置循环的次数。

算子参数说明

## 操作流程

### a. 流程编辑

如图 4.10.2 所示，循环次数算子主要放在循环模块中，设置循环模块循环次数，当循环模块运行次数达到设置次数时，循环次数输出参数 `outCountResult` 为 `true`，将 `outCountResult` 与循环退出节点相连，退出循环模块。

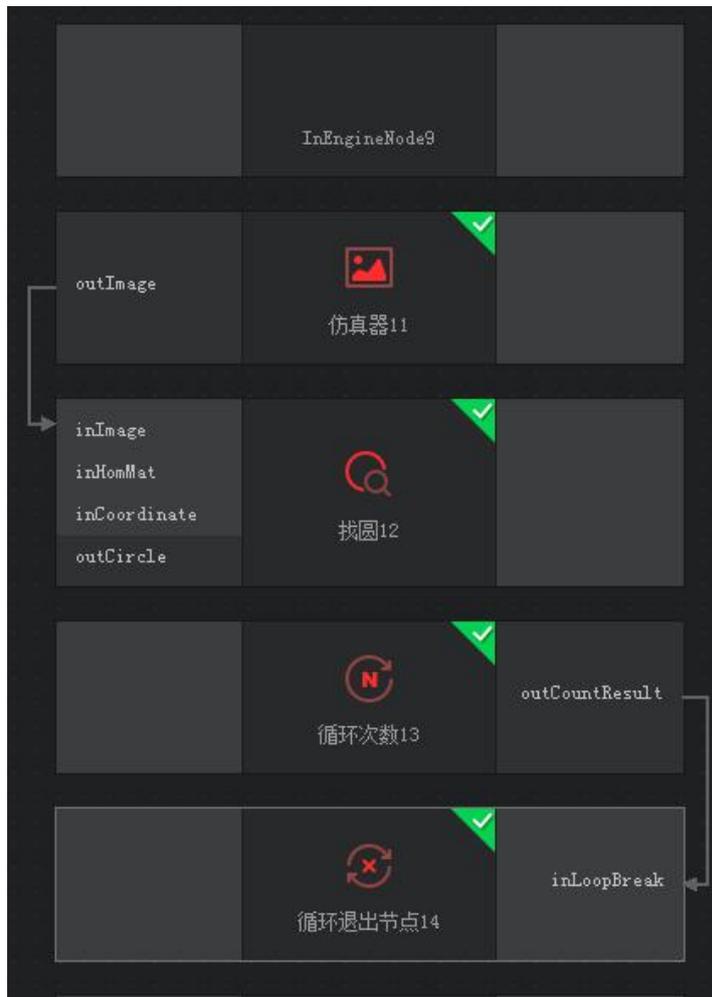


图 4.10.2 循环次数流程编辑

### b. 配置参数编辑

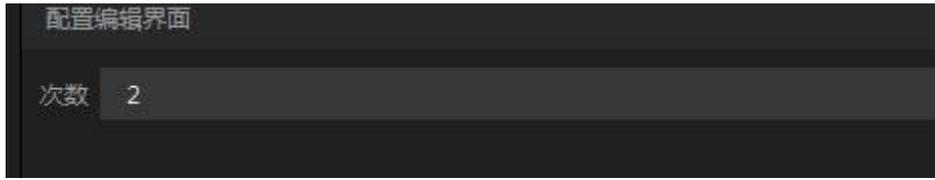


图 4.10.3 循环次数配置参数编辑

图 4.10.3 中，可以设置设置算子的次数。

## 配置参数

循环次数的配置参数主要设置循环模块循环运行的次数，当循环模块运行次数达到设置次数时配合循环退出节点就会退出循环模块。

表 4.10.3 保存图片配置参数表

参数名称	数据类型	说明
counterVal	int	循环次数

## 输出参数

输出参数主要包含算子读取输入图片的保存路径。输出参数可以直接拖拉到结果界面显示。

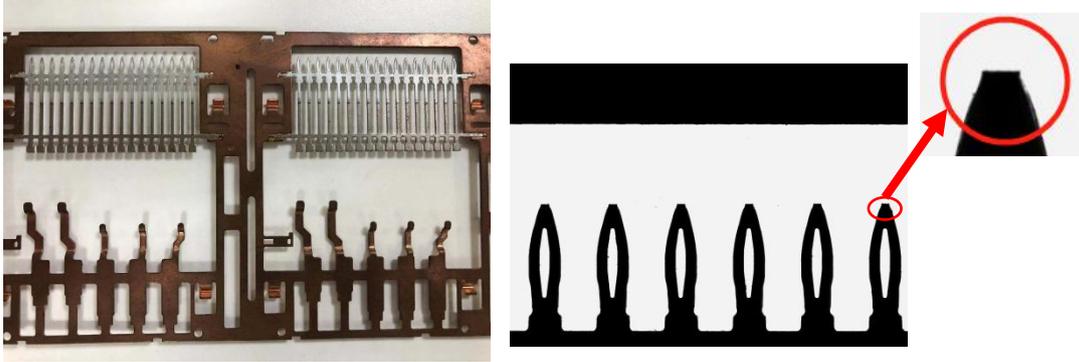
表 4.10.4 相机输出参数表

参数名称	数据类型	说明
costTime	int	运行时长
outCountResult	bool	循环次数达到设置次数时显示为 <b>true</b> ，否则为 <b>false</b>
result	bool	循环次数算子运行结果

## 第 5 章 案例展示

### 5.1 Pin 针针头尺寸检测

案例效果：测量汽车连接器 pin 针针头尺寸，如图



步骤 1：方案搭建思路。

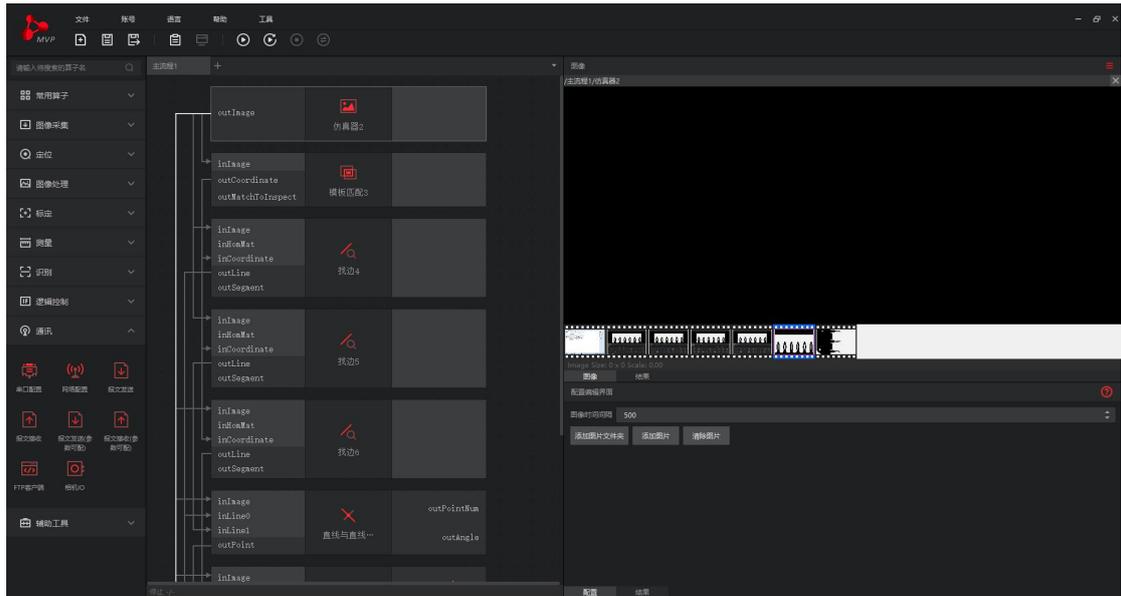
- 通过模板匹配 pin 针在图像中的位置。
- 分别查找每个 pin 针针头的三条边（左、中、右）。
- 分别找出每个 pin 针针头边缘的交点。
- 测量每个 pin 针左顶点与右顶点的距离。

以其中一个 pin 针的针头尺寸测量为例，整体的解决方案如下图所示：

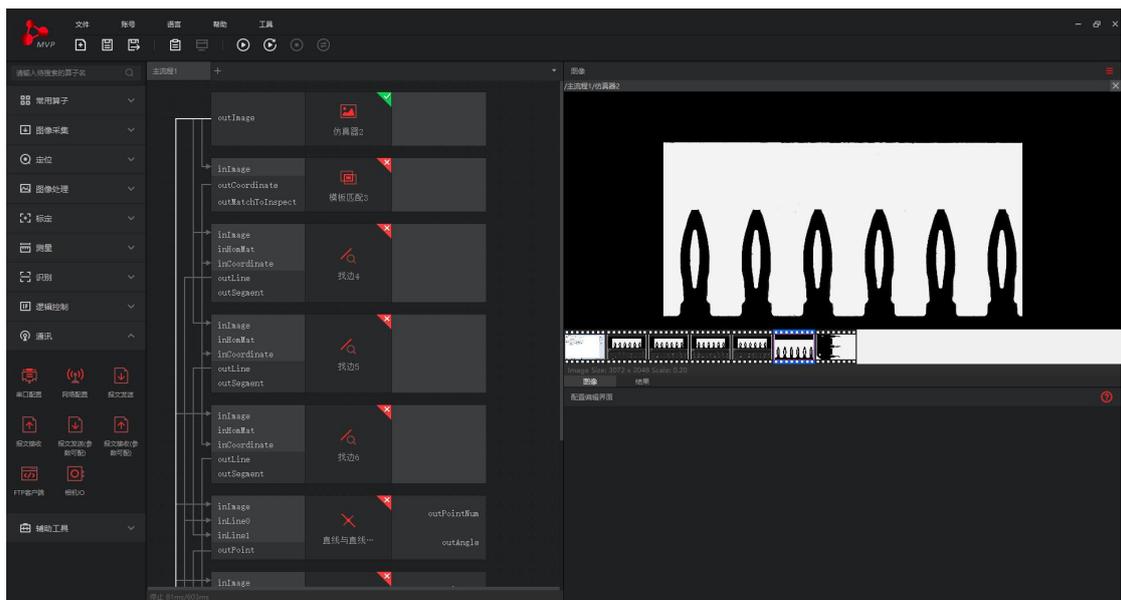


步骤 2：方案保存。

步骤 3：在仿真器中加载拍摄的 pin 针图片（或连接相机实时采图），如下图所示：

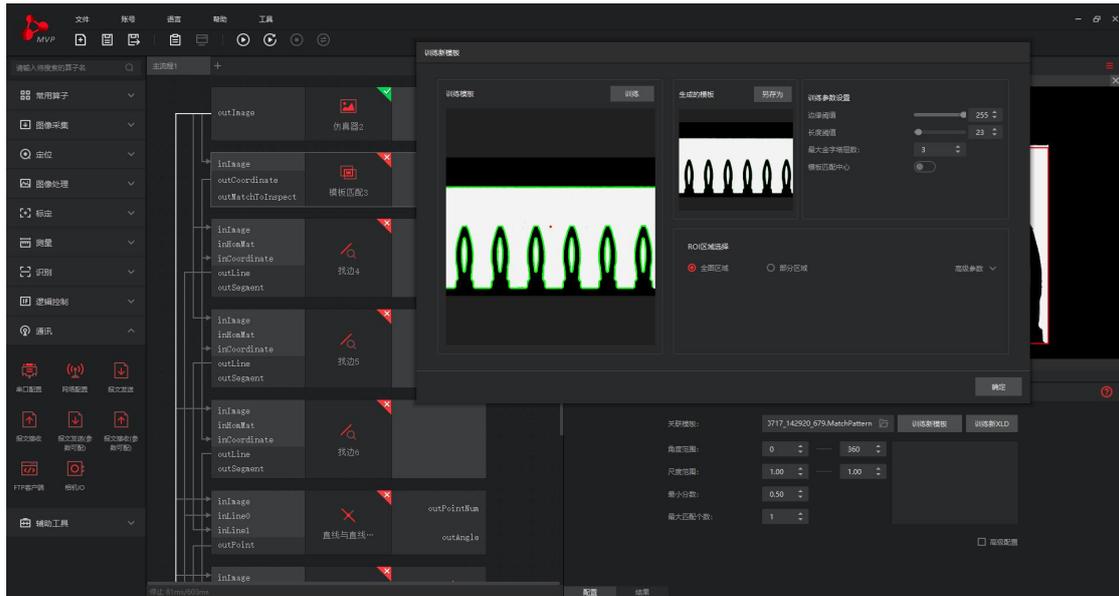


步骤 4：单击单次运行，如下图所示：

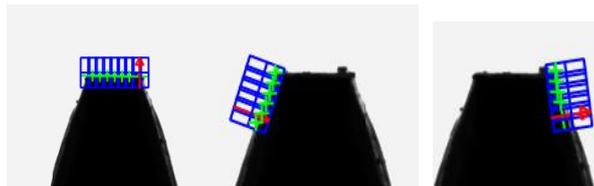


步骤 5：训练模板（如下图所示）

- 单击模板匹配算子；
- 在右侧配置编辑界面单击训练新模板；
- 在训练新模板界面 ROI 区域选择，选择全图区域；
- 调整配置参数至框选出想要的模板轮廓，单击训练，点击确定。

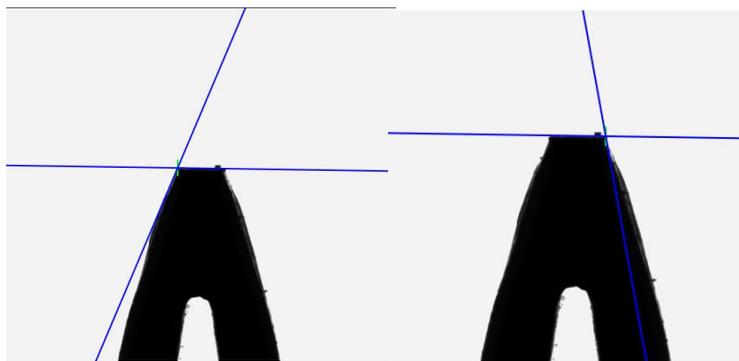


步骤 6：单击运行一次，使用三个找边工具，分别找到 pin 针针头的三条边。

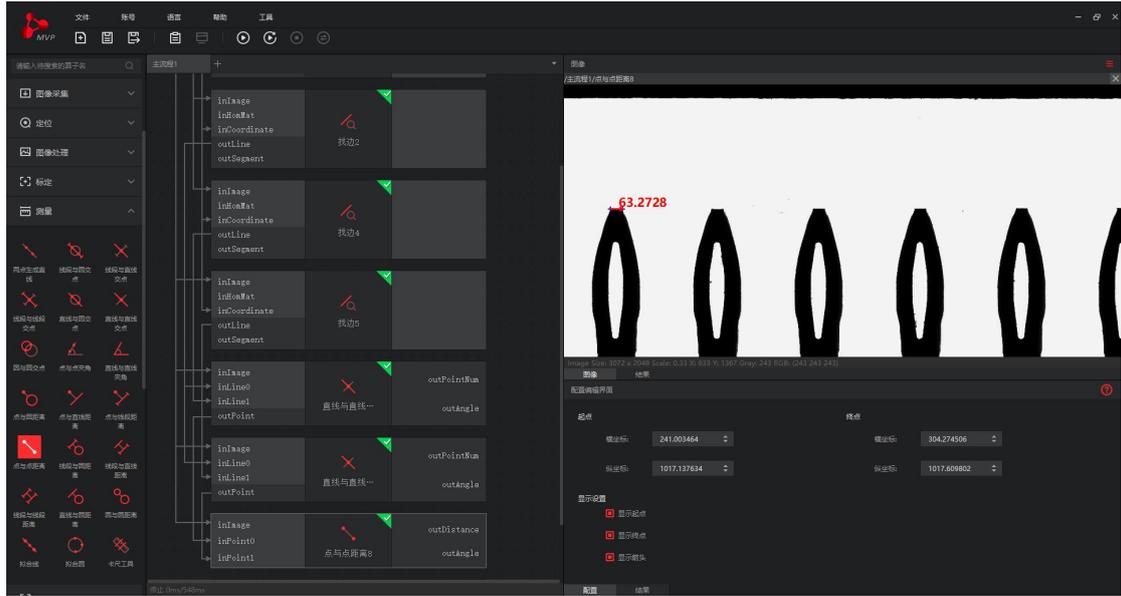


如下图所示：

步骤 7：单击运行一次，找边 4 与边 5、边 4 与边 6 的交点。如下图所示：



步骤 8：单击运行一次，找到 pin 针针头的像素尺寸。如下图所示：



注：若配置好标定文件，可获得工件的实际坐标。