

# LeadVision 软件使用手册

- ◆ 非常感谢您本次购买雷赛产品
- ◆ 使用前请仔细阅读此说明书，正确使用产品
- ◆ 请妥善保管此说明书

©Copyright 2024 Leadshine Technology Co., Ltd.

All Rights Reserved.

# 前 言

## 版权说明

本手册版权归深圳市雷赛控制技术有限公司所有，未经本公司书面许可，任何人不得翻印、翻译和抄袭本手册中的任何内容。

本手册中的信息资料仅供参考。由于产品改进和功能升级等原因，雷赛公司保留对本资料的最终解释权，内容如有更改，恕不另行通知。

## 版本变更记录

修订日期	发布版本	变更内容
2024/3	V1.0.0.0	初版发行

# 目 录

前 言.....	1
<b>1. 产品简介.....</b>	<b>6</b>
1.1 软件概述.....	6
1.2 运行环境.....	6
1.3 软件安装.....	6
<b>2. 软件操作.....</b>	<b>10</b>
2.1 软件 UI 概述 .....	10
2.2 菜单栏.....	11
2.2.1 文件 .....	11
2.2.2 工程 .....	14
2.2.3 相机 .....	15
2.2.4 通信 .....	17
2.2.5 图像 .....	25
2.3 快捷工具条.....	26
2.4 全局变量.....	26
2.5 工具列表.....	27
2.5.1 工具介绍 .....	28
2.5.2 工具操作 .....	28
2.6 流程编辑栏.....	29
2.6.1 流程选项卡 .....	30
2.6.2 操作工具栏 .....	30
2.6.3 流程编辑区 .....	31
2.7 结果显示.....	32
2.7.1 图像显示区 .....	32

2.7.2	工具结果显示区 .....	34
<b>3.</b>	<b>算子使用说明.....</b>	<b>34</b>
3.1	图像采集.....	34
3.1.1	采集图像 .....	34
3.2	标定.....	35
3.2.1	测量标定 .....	35
3.2.2	标量转换 .....	37
3.2.3	N 点标定 .....	38
3.2.4	手动标定 .....	41
3.2.5	坐标转换 .....	44
3.3	定位.....	45
3.3.1	轮廓匹配 .....	45
3.3.2	圆查找 .....	49
3.3.3	圆弧查找 .....	53
3.3.4	直线查找 .....	55
3.3.5	特征点定位 .....	58
3.3.6	BLOB 分析.....	60
3.4	测量.....	62
3.4.1	线线距离 .....	63
3.4.2	线线交点 .....	64
3.4.3	间距测量 .....	66
3.4.4	线线计算 .....	68
3.4.5	线圆计算 .....	69
3.4.6	圆圆计算 .....	70
3.4.7	点圆计算 .....	71
3.4.8	点线计算 .....	72
3.4.9	点点计算 .....	73
3.4.10	直方图统计 .....	74

3.4.11	像素统计 .....	76
3.5	识别.....	78
3.5.1	一维码 .....	79
3.5.2	二维码 .....	80
3.6	图像处理.....	80
3.6.1	图像运算 .....	80
3.6.2	图像保存 .....	82
3.6.3	图像预处理 .....	83
3.7	颜色处理.....	87
3.7.1	彩色转灰 .....	87
3.7.2	颜色空间转换 .....	88
3.7.3	图像通道分离 .....	89
3.8	系统工具.....	90
3.8.1	读取全局变量 .....	90
3.8.2	写入全局变量 .....	91
3.8.3	局部变量 .....	92
3.8.4	写入局部变量 .....	93
3.8.5	数组索引 .....	93
3.8.6	阈值判断 .....	94
3.8.7	结果显示 .....	95
3.8.8	计数器 .....	102
3.8.9	定时器 .....	102
3.8.10	计时开始 .....	102
3.8.11	计时结束 .....	103
3.9	逻辑控制.....	103
3.9.1	IF 语句.....	103
3.9.2	Else 语句 .....	104
3.9.3	For 循环.....	105

3.9.4	循环中断 .....	105
3.9.5	选择分支 .....	106
3.9.6	CaseX.....	106
3.9.7	工具组 .....	107
3.9.8	并行子流程 .....	107
3.9.10	并行同步 .....	107
3.9.11	JavaScript 脚本.....	108
3.10	通信.....	132
3.10.1	TCP 通信发送.....	133
3.10.2	TCP 通信接收.....	134
3.10.3	Modbus 写入 .....	135
3.10.4	Modbus 读取 .....	137
<b>4.</b>	<b>案例搭建.....</b>	<b>138</b>
4.1	零件定位检测.....	138
4.1.1	案例目标 .....	138
4.1.2	方案搭建 .....	138

# 1. 产品简介

## 1.1 软件概述

LeadVision 是一款标准化的机器视觉开发平台软件，集成手眼标定、图像处理、定位识别、检测测量、通信工具、逻辑控制等丰富的视觉工具。通过简易图形化编程方式，开发者无需代码编程能力即可快速完成平台对位、检测识别、尺寸测量等视觉应用的开发工作。该视觉平台具备良好的可扩展性，支持用户嵌入自定义算子及工艺。同时，也提供算法 SDK 开发包，支持用户深度二次开发。

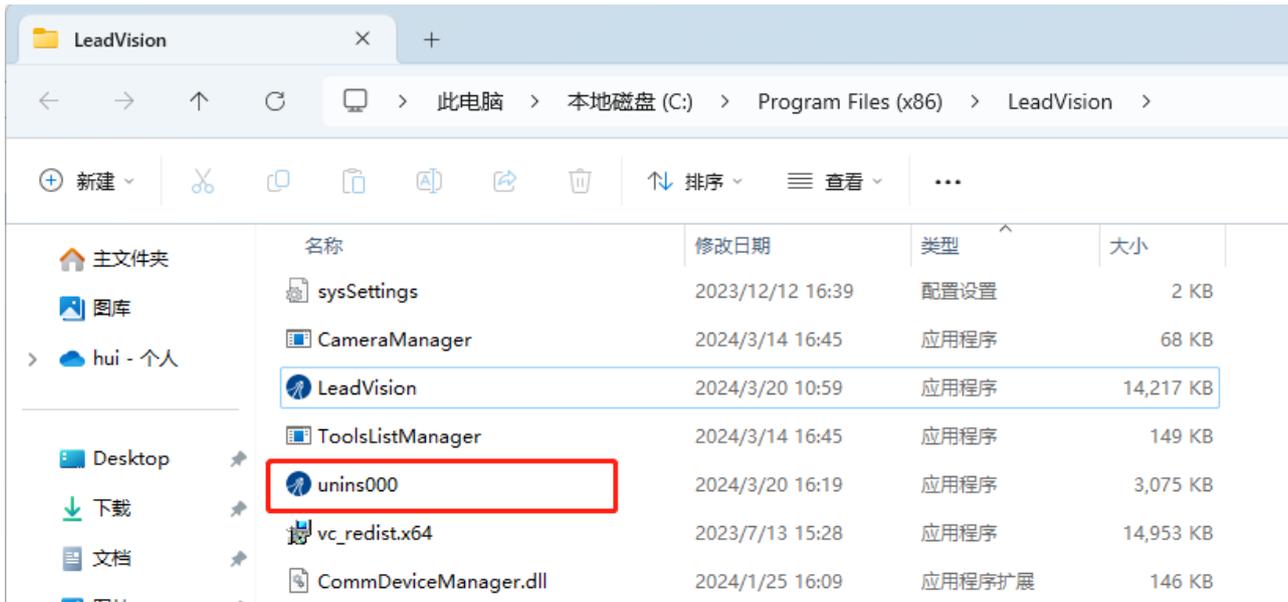
## 1.2 运行环境

为确保软件正常安装及运行，推荐使用以下 PC 配置。

项目	推荐配置	最低配置
CPU	多核 Intel Core i5 2GHz 或以上	Intel 赛扬 J1900
内存	8GB 或以上	4GB
硬盘	SSD 128GB 或以上	SSD 64GB
显卡	显存 1GB 或以上	
网卡	Intel i210 系列或以上	
USB 接口	USB3.0 接口（如需使用 USB3.0 相机）	
操作系统	Win10 64 位	Windows7

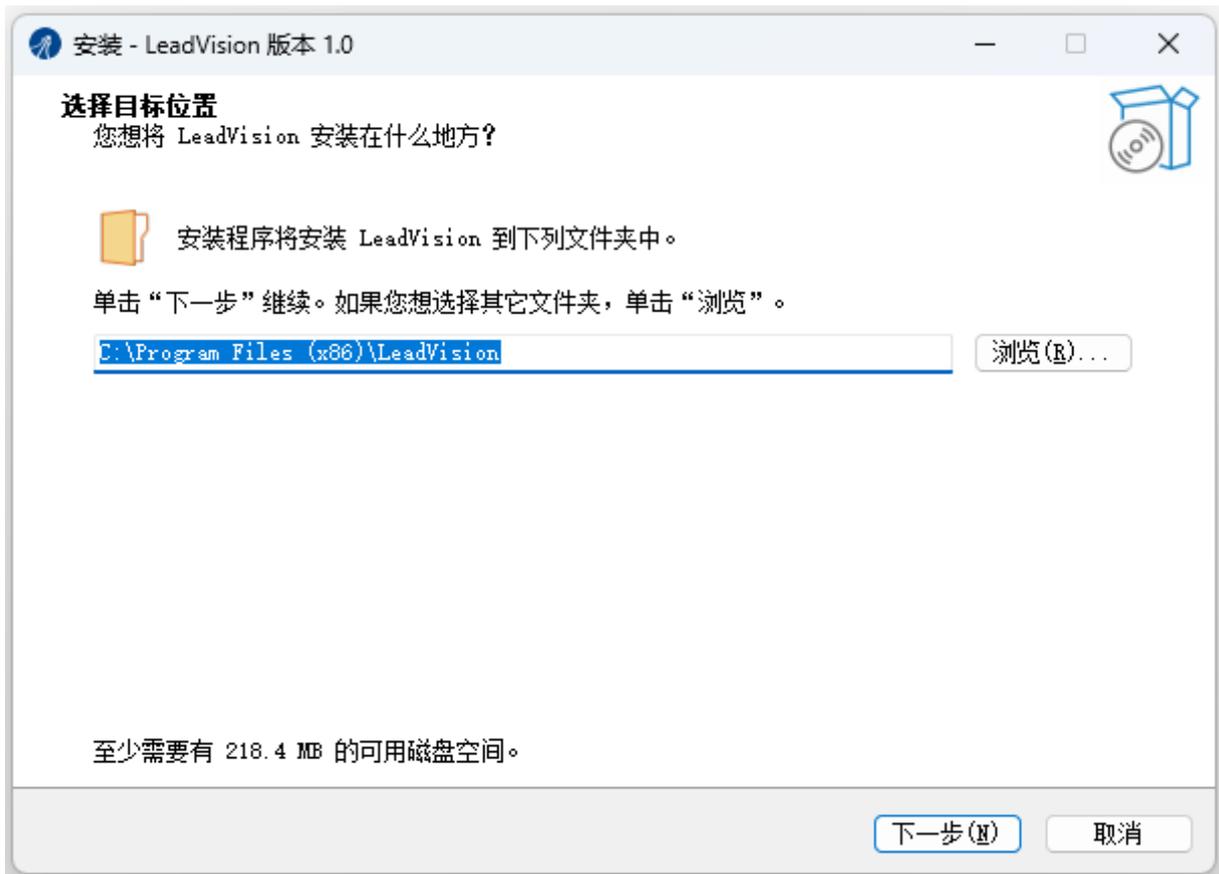
## 1.3 软件安装

在安装 LeadVision 之前，若已安装之前版本，需先手动卸载老版本，再运行安装包。卸载文件 unins000.exe，默认在软件运行目录下，如下图 C:\Program Files (x86)\LeadVision。

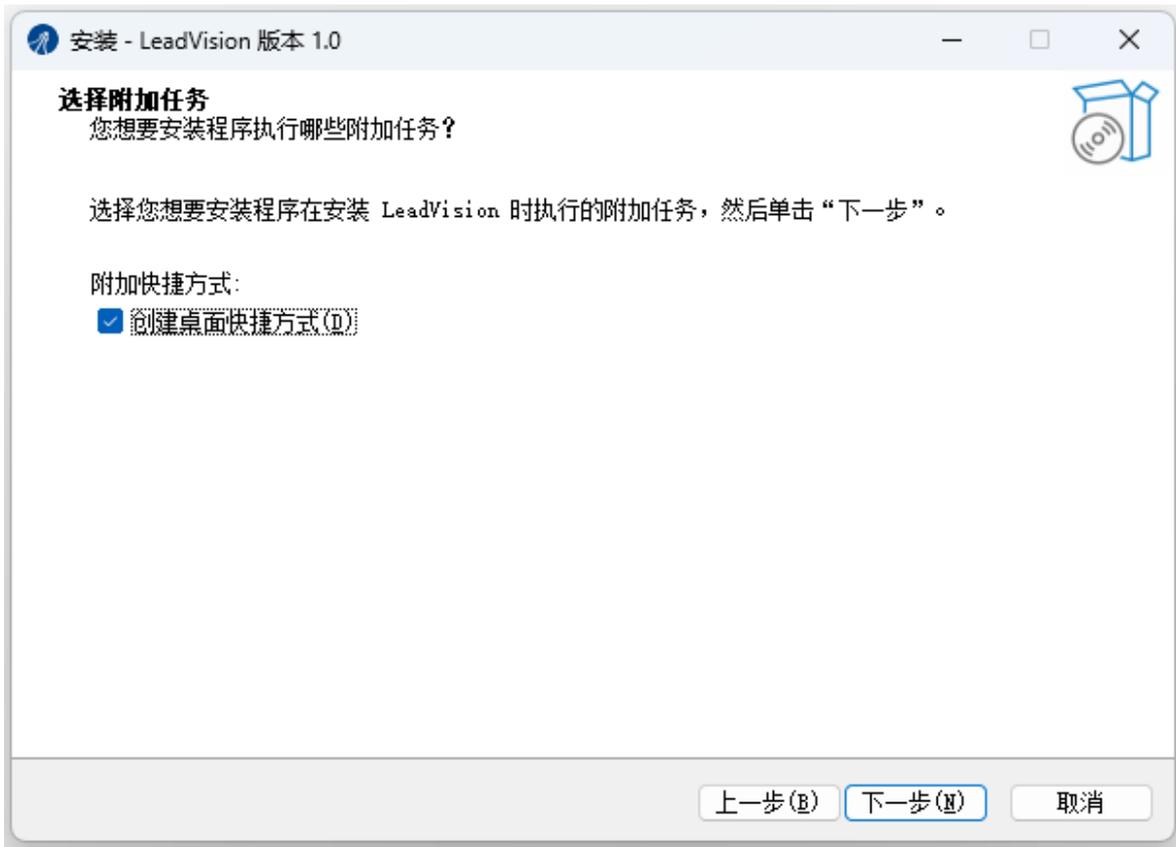


LeadVision 提供 64 位安装包，安装步骤如下：

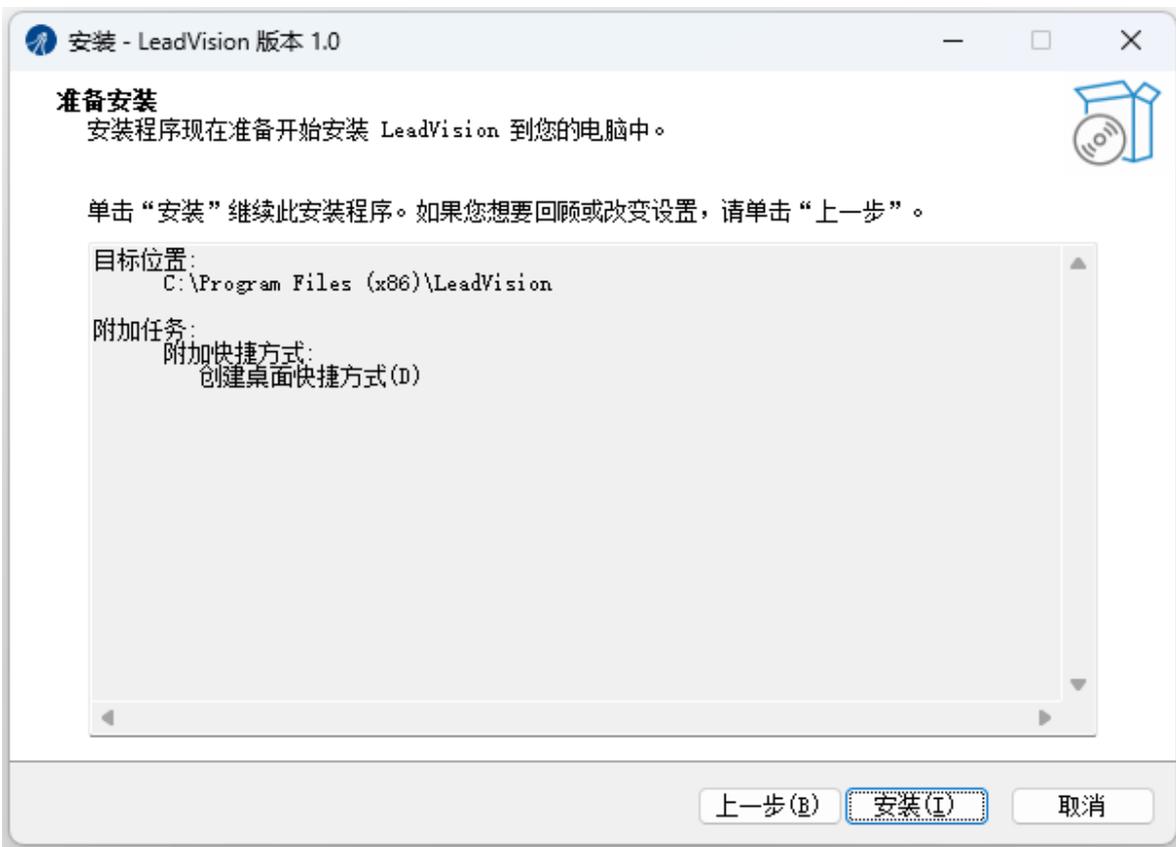
- (1) 双击 LeadVision.exe 安装包进行安装。
- (2) 设置软件安装路径。



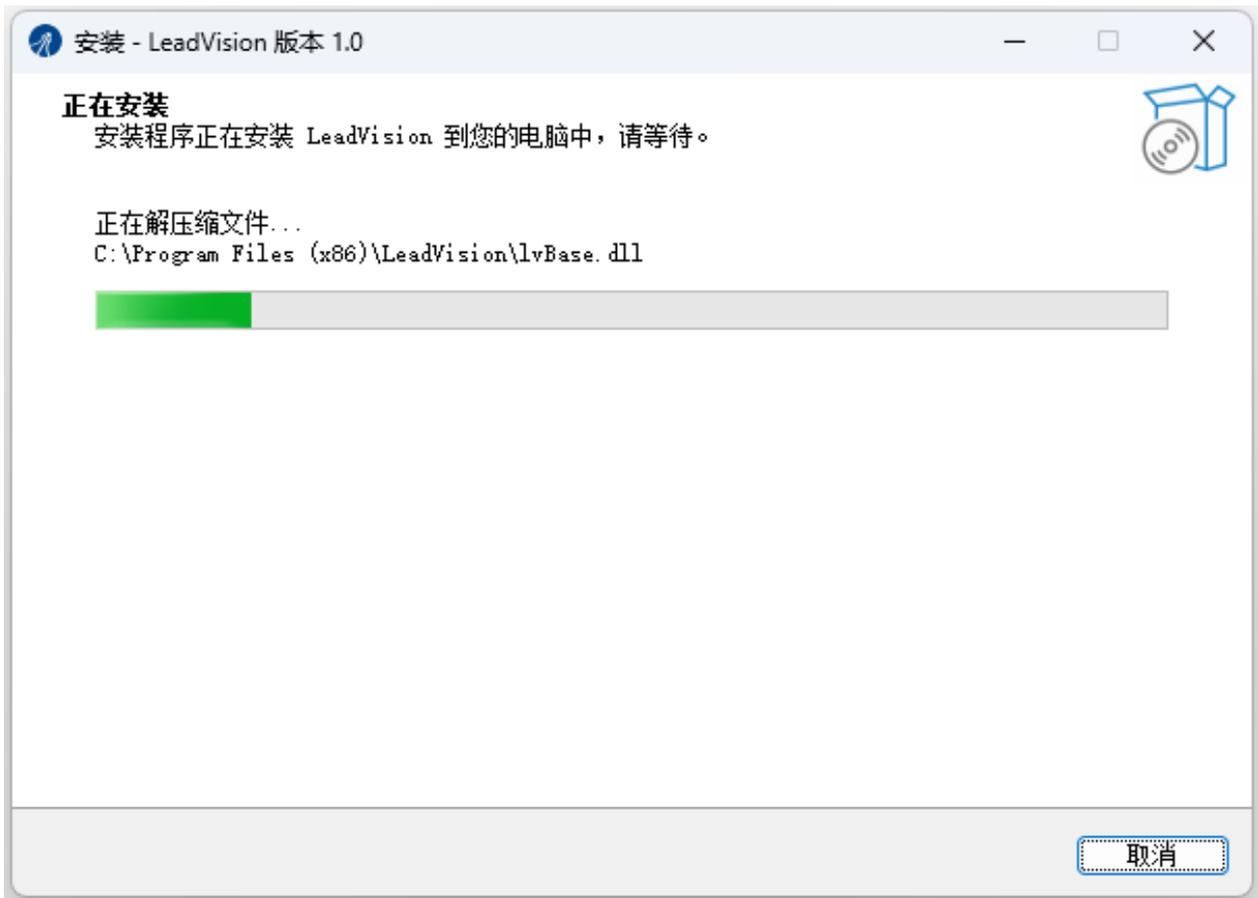
- (3) 创建快捷方式。



(4) 单击“安装”



(5) 等待安装完成。



(6) 完成安装。





## ⑥ 结果显示区

- a) 显示工具的输出变量及值。
- b) 显示软件运行状态信息。

## ⑦ 运行看板区——配置自动运行界面的电子看板信息，结合阈值判断工具可列表方式显示数据结果。

## 2.2 菜单栏

主界面中最上方显示软件的菜单栏。包含了文件、视图、工程、相机、通信、调试、图像、帮助等选项。



### 2.2.1 文件

#### 2.2.1.1 权限登入

验证登录权限的密码进行登录确定，权限等级分为三级权限，可逐级授权每个层级的使用权限。初始状态无需登录，直接为最高权限。

系统商(Developer)——最高权限，可以获得软件的所有操作权限，默认密码为 123。

技术员(Technician)——中等权限，由系统商授权使用权限，可以获得基本的编辑、调试、运行权限。

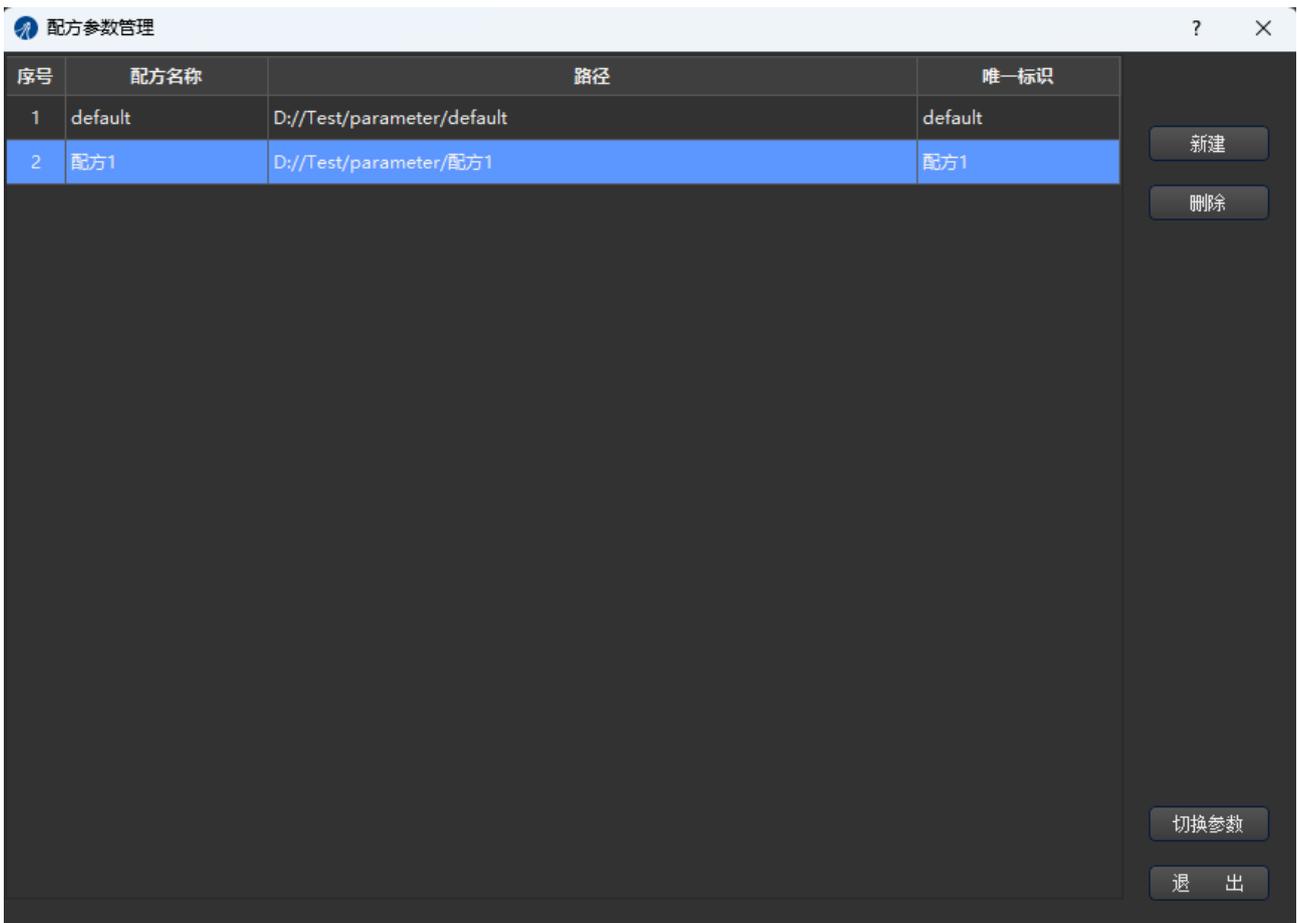
操作员(Operator)——最低权限，由技术员授权使用权限，可以获取有限的权限。



### 2.2.1.2 配方管理

同一工程中的视觉工具可能需要设置不同参数，来定位或检测同一款但颜色不同的产品，此时可通过配方实现，工程中的流程完全相同，只是运行时视觉工具的参数不同。

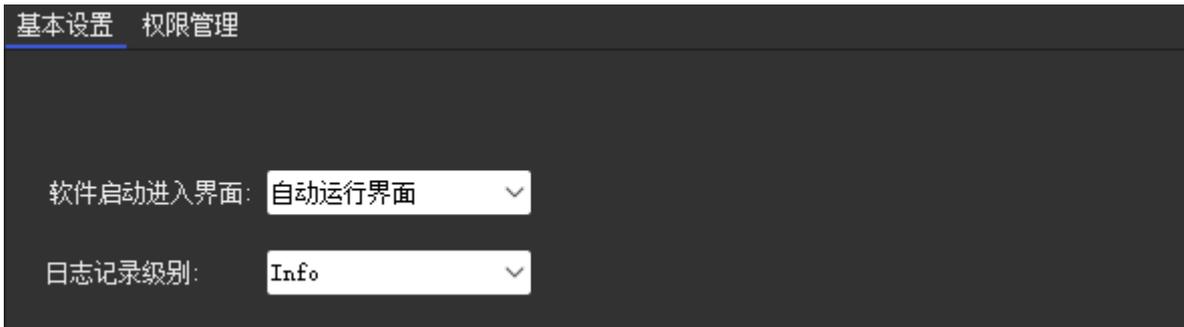
配方管理用于新建、删除、切换配方。工程中默认使用 **default** 配方，用户可以修改该配方名称，也可基于现有配方新建一组配方，或切换不同配方。在工程标题栏会显示当前使用的配方名称。



### 2.2.1.3 系统设置

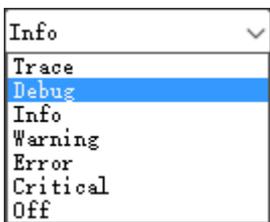
用于设置软件启动界面、日志记录级别、用户操作权限等。

- 基本设置



软件启动进入界面：设置软件启动时显示的界面，包括自动运行界面和编辑调试界面。

日志记录级别：记录相机、通信、软件操作等运行信息。日志由下往上，依次为包含关系，即 Trace 日志会记录所有信息，包括 Debug、Info、Warning、Error、Critical 记录的信息。设置为 off 时，关闭日志记录功能。



- 权限管理

序号	功能	是否向技术员授权
1	软件启动画面设置/Set Startup UI	No
2	工程设置/Project setting	No
3	编辑任务流程/Edit task flow	No
4	编辑任务名称/Edit task flow name	No
5	编辑标定流程/Edit calib flow	No

用于设置当前登入者的下一等级的用户访问权限，包括菜单、流程编辑、工具栏等操作权限。当登入为系统商时，可以配置技术员的访问权限。当登入为技术员时，可以配置操作员的访问权限。

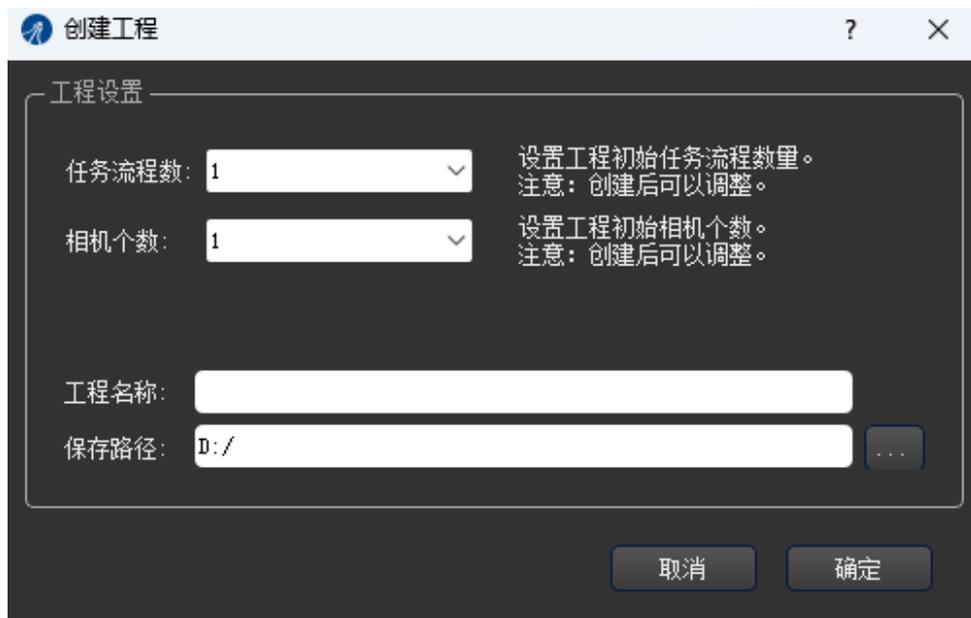
## 2.2.2 工程

该子菜单栏包括新建工程、加载工程、工程另存为、工程设置、最近使用的工程等操作选项。工程搭建一般流程，先创建工程，然后添加任务流程，最后在任务流程中根据应用需要拖拽添加对应的视觉插件工具。



- 新建工程

可创建新的工程。创建新工程时，需设置当前工程使用的相机个数、任务流程数、工程名称及工程保存路径。



- 加载工程

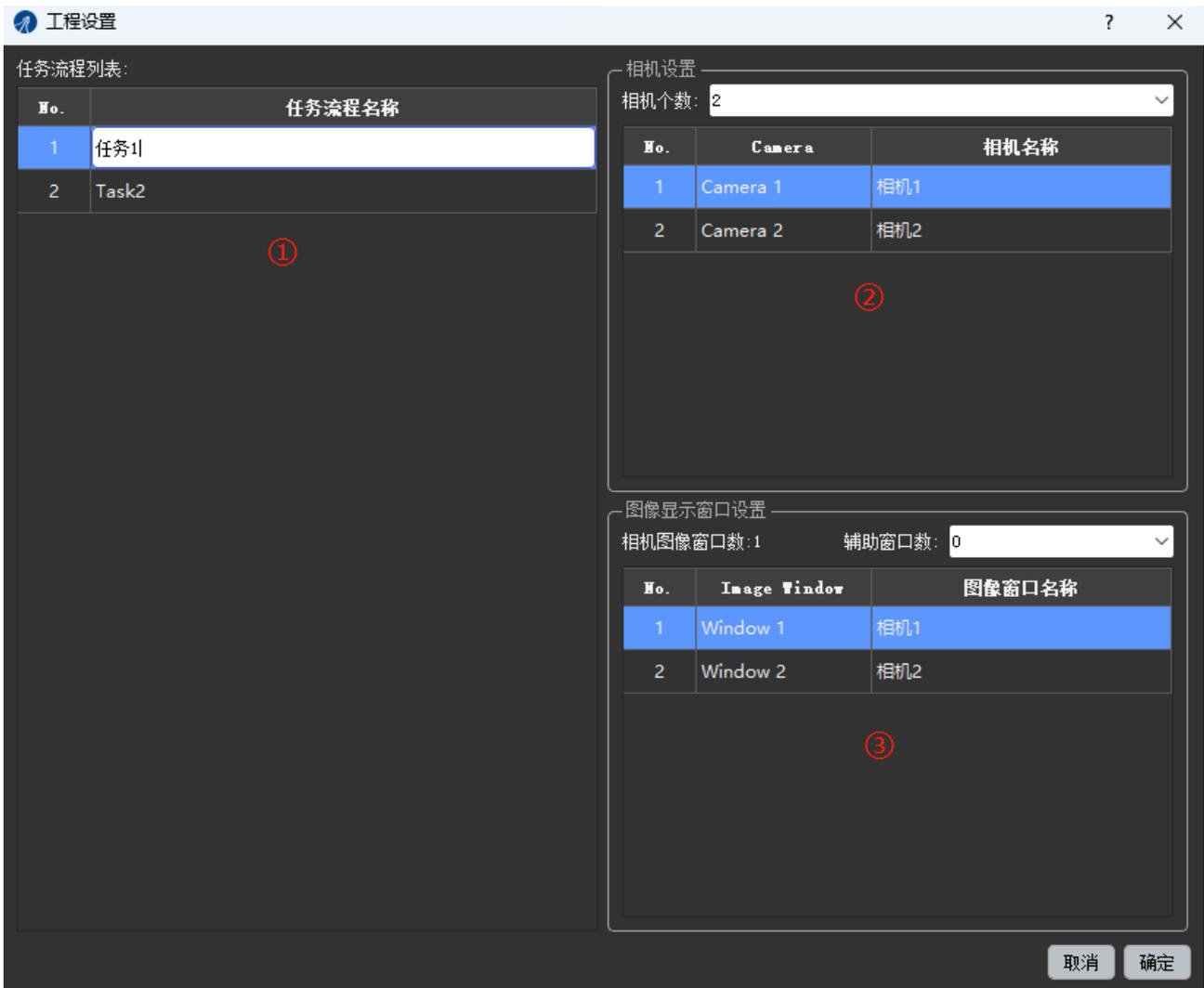
加载示例工程或之前保存的工程。

- 工程另存为

将当前视觉平台工程另存为到指定的路径。

- 工程设置

可修改任务名称、相机名称、相机个数、窗口名称、窗口个数等。



① 任务流程

工程中所有任务流程的名称修改。

② 相机数量

工程中相机数量的设置，相机名称的修改。

③ 窗口数量

工程中图像窗口数量的设置，默认与相机个数相同，如需增加窗口，可设置辅助窗口数量。以及图像窗口名称的修改。

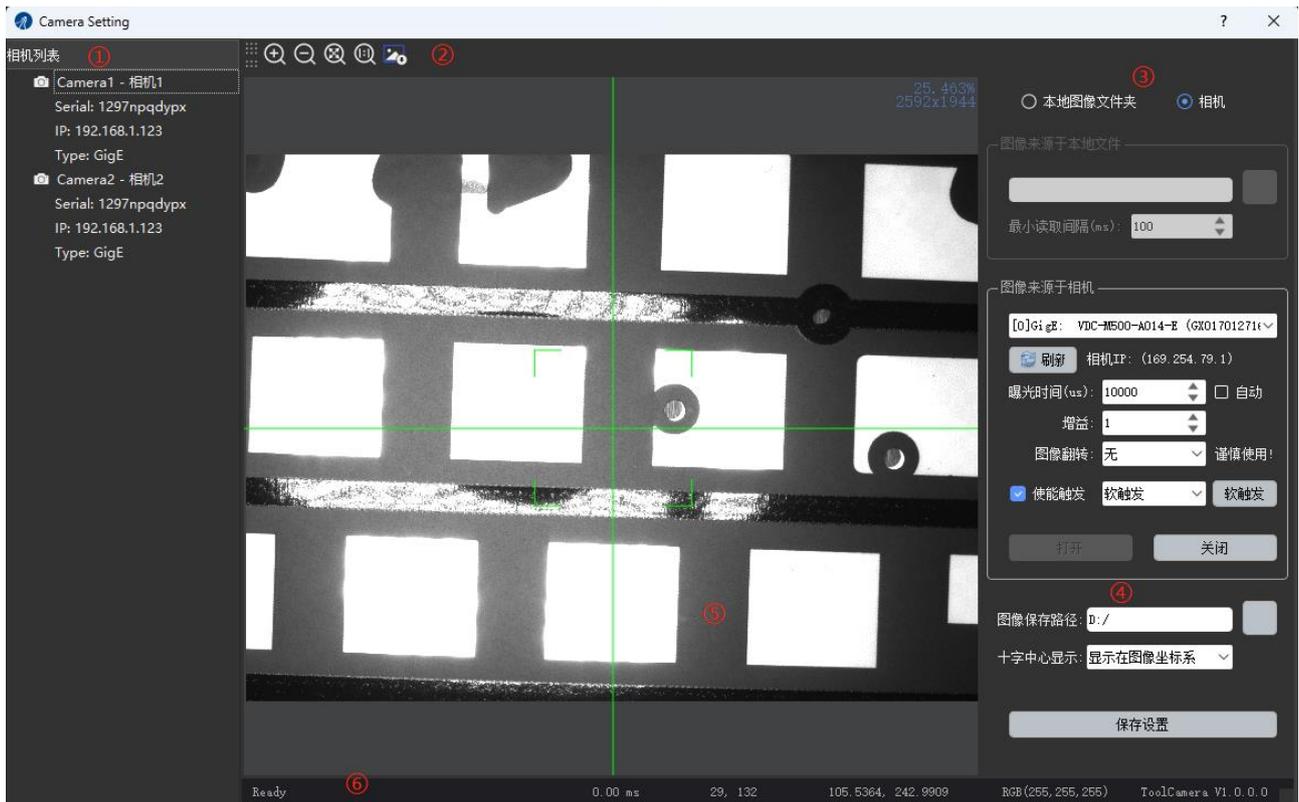
● 最近使用的工程

打开最近打开过的视觉平台工程

## 2.2.3 相机

● 相机配置

➤ 配置图像源及相机参数。



### ① 相机列表

- 显示工程使用的相机数，可切换相机并配置该相机的图像来源及参数。

### ② 工具栏



放大图像



缩小图像



使图像适合窗口



显示原始图像



保存图像

### ③ 图像源选择

- 可选择本地图像或相机图像
- 图像来源于本地图像文件
  - 本地图像文件夹：从本地文件夹中读取图片。
  - 最小读取间隔：两张图片读取的间隔时间，ms 单位。
- 图像来源于相机

- 刷新：刷新当前连接的相机，在下拉框中选中当前使用的相机。
- 相机 IP：当前使用的相机 IP 地址。
- 曝光时间：设置相机手动或自动曝光时间，单位 us。
- 增益：设置当前相机的增益
- 图像翻转：对采集的图像旋转处理，包括垂直翻转、水平翻转、旋转 180 度、旋转 90 度、旋转-90 度、垂直翻转+旋转 90 度、水平翻转+旋转 90 度。
- 使能触发：勾选时，相机为外触发模式，包括软触发、外部硬件触发；不勾选时，相机为内触发模式的连续采集模式，即相机自动地按帧率进行曝光输出图像。

#### ④ 保存

- 图像保存路径：设置保存图像时的存储路径，保存图像的名称为 image+当前时间，如 image2023-12-06\_11-51-03\_255
- 十字中心显示：可选择显示在图像坐标系、显示在窗口坐标系和不显示。
- 保存设置：保存当前配置的设置选项。

#### ⑤ 显示区

- 显示图像信息

#### ⑥ 状态栏

- 显示光标在窗口上的坐标、在图像上的像素坐标、颜色等信息。

#### ● 坐标标定

- 将任务流程切换到标定任务流程。在此处编写标定流程。

## 2.2.4 通信

### 2.2.4.1 通信设备管理器

集中管理工程中所使用的通信设备，支持 TCP、UDP、Modbus、串口等通信协议。用于视觉软件平台与外部数据的交互，当通信构建后，既可以把处理结果发送给外部，也可以通过外部发送字符触发软件运行或相机拍照。



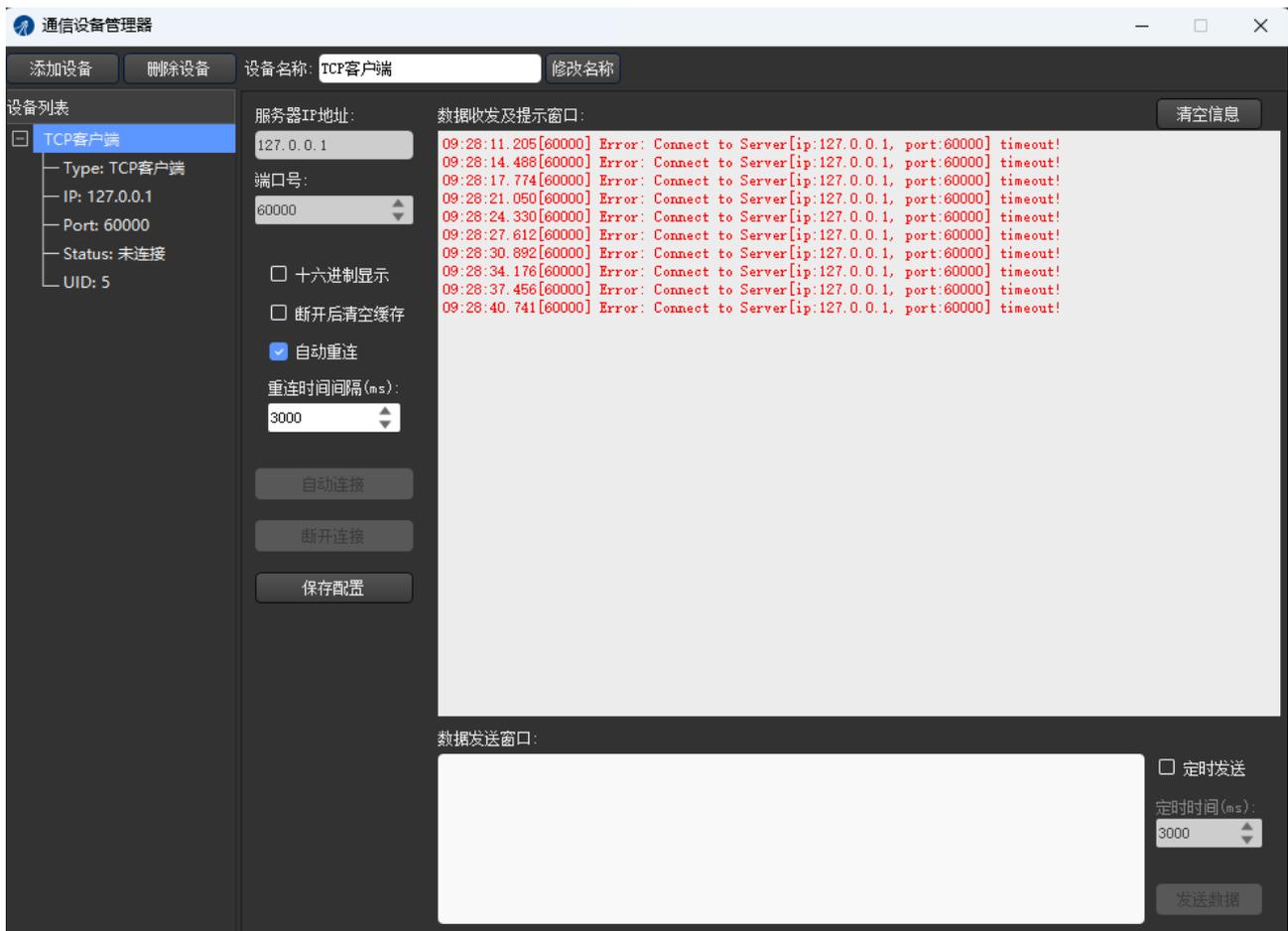
- 添加设备：点击添加配置相应的通信设备。
- 删除设备：选择设备列表中的设备进行删除。



## 2.2.4.2 TCP 通信

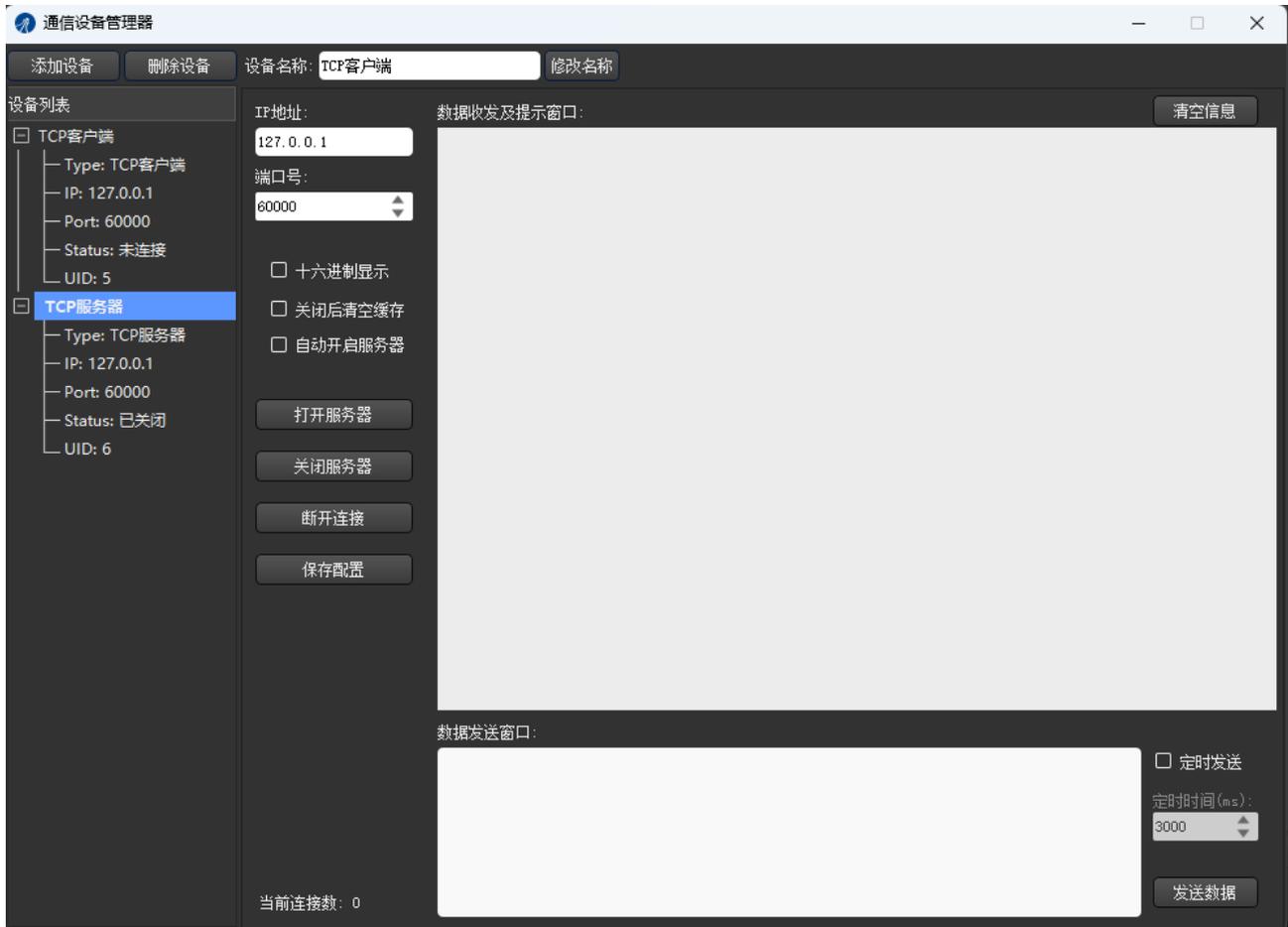
即可做客户端也可做服务器。

- TCP 客户端



- 设备名称：可自定义通信设备名称。
- 服务器 IP 地址：连接的目标设备的 IP 地址。
- 端口号：连接的目标设备的端口号。
- 十六进制显示：勾选后，接收数据以十六进制显示。
- 断开后清空缓存：勾选后，连接断开时，清空缓存区的数据。
- 自动重连：勾选后，与服务器断开时会自动重连。
- 重连时间间隔：设置重连时间间隔，单位 ms。
- 连接服务器：与服务器建立连接。
- 断开连接：与服务器断开连接。
- 保存配置：保存当前配置的参数。
- 定时发送：勾选后，按固定时间发送数据，仅供测试使用。
- 定时时间：两帧发送数据之间的间隔，单位 ms。
- 数据收发及提示窗口：显示接收数据及连接信息。
- 数据发送窗口：发送数据输入区。

- 发送数据：将数据发送窗口的数据发送给目标，仅供测试使用。
- 清空信息：清空接收窗口的数据。
- TCP 服务器



- IP 地址：本机服务器 IP 地址。
- 端口号：本机服务器端口号。
- 自动开启服务器：勾选后，重新打开软件时，自动开启服务器。
- 开启服务器：开启本机 TCP 服务器。
- 关闭服务器：断开所有已连接的客户端连接，不再接受任何客户端的连接。
- 断开连接：断开已经建立连接的客户端，客户端可再次操作建立连接。
- 当前连接数：显示当前已连接的客户端数量。
- 其他参数配置与 TCP 客户端基本一致，可参考 TCP 客户端章节的参数介绍。

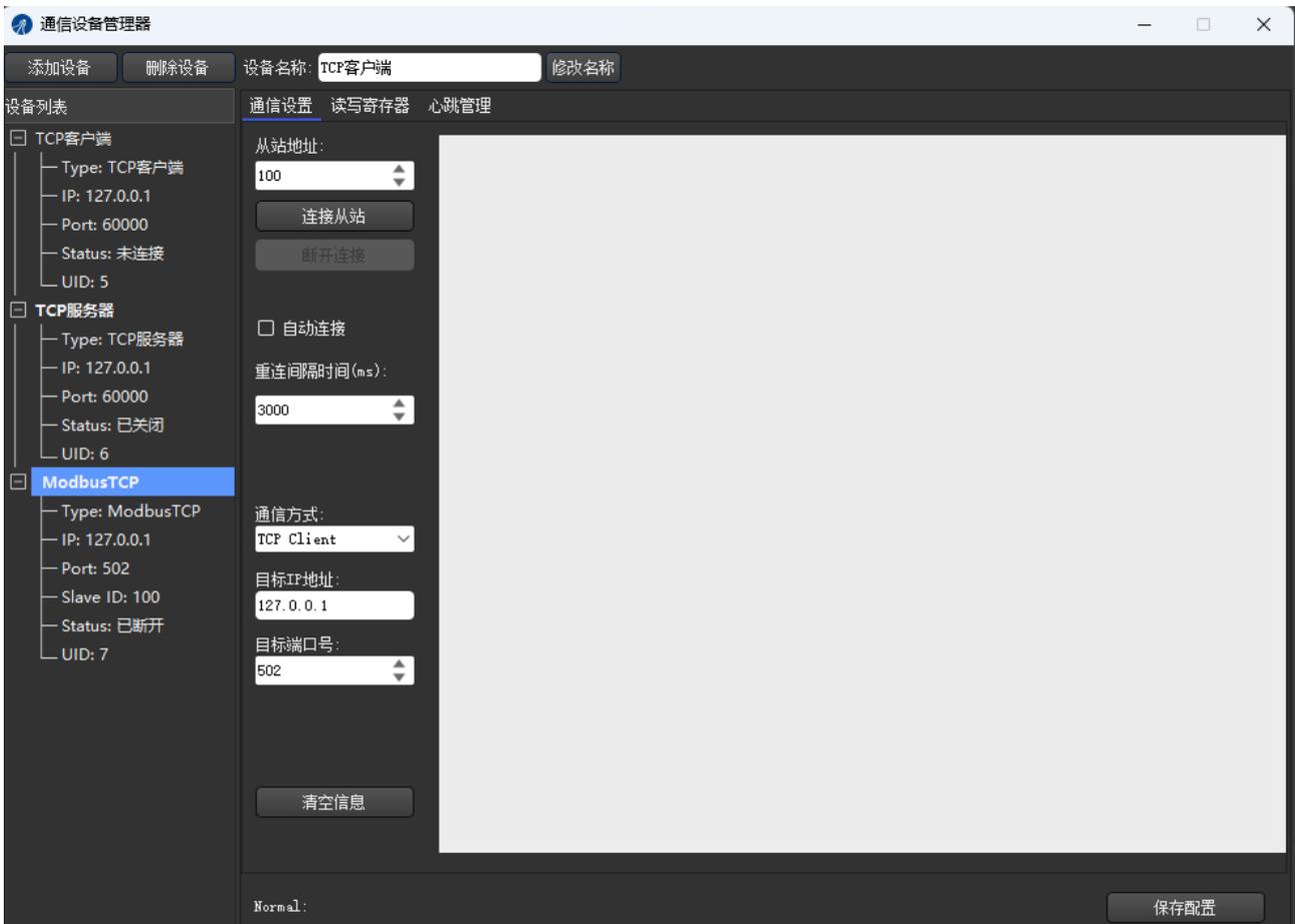
### 2.2.4.3 Modbus 通信

Modbus 协议是一个主从架构的协议。有一个节点是主站，其他使用 Modbus 协议参与通信的节点是从站，目前视觉平台仅支持做主站，从下位机（从站）中读取寄存器的值。支持的功能码如下：

功能	功能码
01 线圈状态(0x)	01: 读线圈状态
02 输入状态(1x)	02: 读输入状态
03 保持寄存器(4x)	03: 读保持寄存器
04 输入寄存器(3x)	04: 读输入寄存器

使用前需先在通信设备管理器中添加 Modbus 通信设备，可添加 ModbusRTU（串口）和 ModbusTCP（网口）。

#### ● ModbusTCP



➤ 设备名称：可自定义通信设备名称。

### ◆ 通信设置选项页

- 从站地址：ModbusTCP 从站的协议站地址。
- 连接从站：与从站建立 Modbus 通信。
- 断开连接：与从站断开 Modbus 通信。
- 自动连接：勾选后，检测到通信断开，会自动重连从站设备。
- 重连间隔时间：重连的时间间隔，单位 ms。
- 通信方式：仅支持客户端模式，即主站模式。
- 目标 IP 地址：从站的 IP 地址。
- 目标端口号：从站的端口号。
- 清空信息：清空消息区的的信息。
- 保存配置：保存设置的通信参数。

### ◆ 读写寄存器选项页

仅用于 Modbus 通信测试。

The screenshot displays a software interface for Modbus communication testing, divided into three tabs: 通信设置 (Communication Settings), 读写寄存器 (Read/Write Registers), and 心跳管理 (Heartbeat Management). The 读写寄存器 tab is active and contains two main sections: 读取设置 (Read Settings) and 写入设置 (Write Settings). Each section includes a button for configuration (读取设置/写入设置), a checkbox for periodic operation (定时读取/定时写入), and a dropdown menu for the periodic time interval (定时时间(ms)), currently set to 200 ms. Below these settings are two data tables, each with a header row containing '00000' and a table body with 10 rows (0-9) of data. The top table shows read data values, and the bottom table shows write data values.

	00000
0	65
1	3691
2	34
3	56
4	4
5	98
6	557
7	333
8	6
9	41

	00000
0	65
1	0
2	34
3	56
4	4
5	98
6	557
7	333
8	6
9	41

- 读取设置：配置读取的功能码、起始地址、寄存器数量。
- 定时读取：勾选后，按固定时间读取通信数据。
- 定时时间：设置定时读取的时间，单位 ms。
- 读取数据：点击时读取一次数据。
- 写入设置：配置写入的功能码、起始地址、寄存器数量。
- 定时写入：勾选后，按固定时间写入通信数据。
- 定时时间：设置定时写入的时间，单位 ms。
- 写入数据：点击时写入一次数据。

#### ◆ 心跳管理选项页

通信设置 读写寄存器 心跳管理

开启心跳功能

心跳设置

心跳间隔时间(ms):  
1000

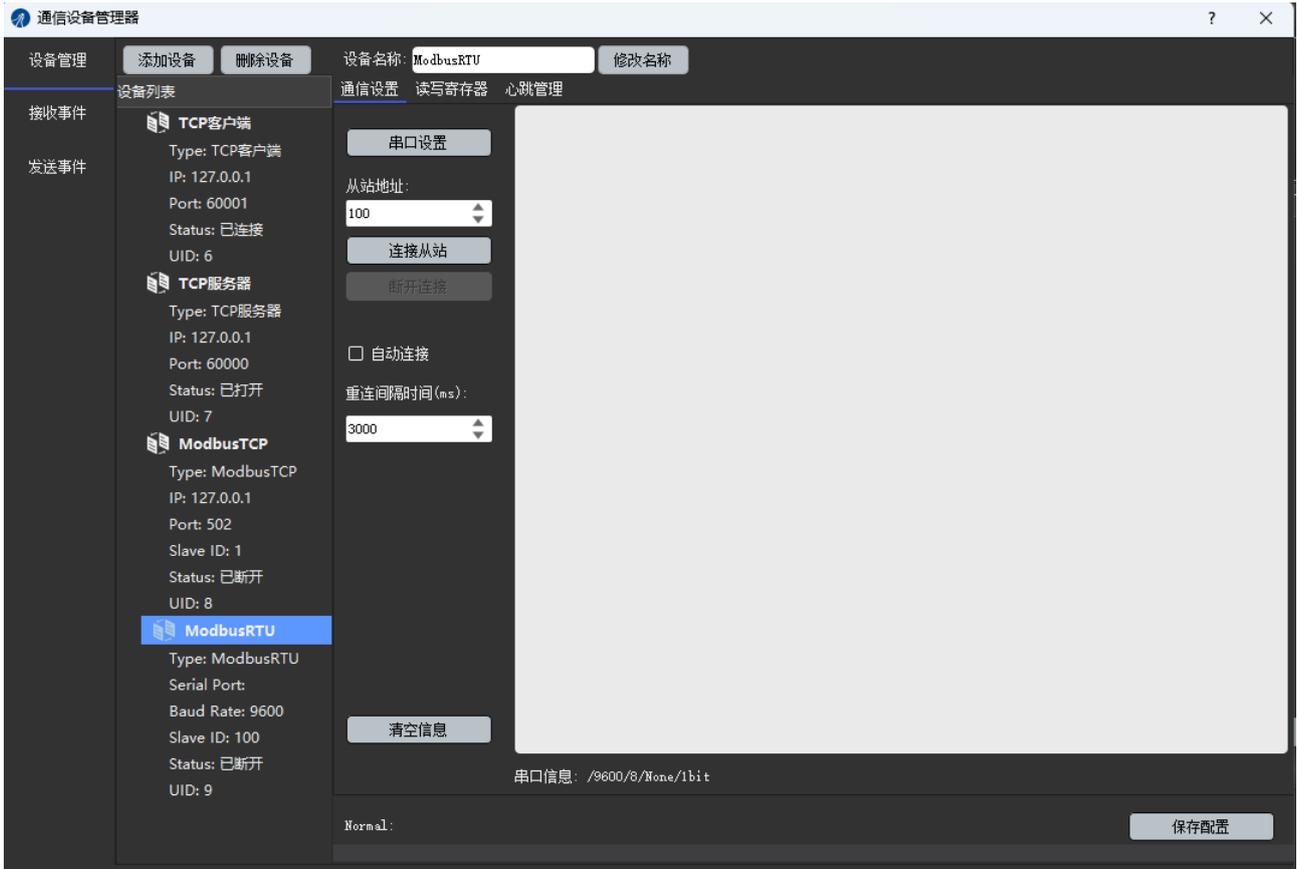
寄存器功能类型:  
03 保持寄存器(4x)

写入地址:  
0

写入值:  
1

- 开启心跳功能：勾选后，启用心跳包功能，定时写入数据。
- 心跳间隔时间：两帧数据的间隔时间，单位 ms。
- 寄存器功能类型：配置 Modbus 功能码。
- 写入地址：设置 Modbus 写入数据的地址。
- 写入值：设置 Modbus 写入的数据值。

#### ● ModbusRTU



设备名称：可自定义通信设备名称。

#### ◆ 通信设置选项页

串口设置：配置串口号、波特率、数据位、校验位、停止位等参数。

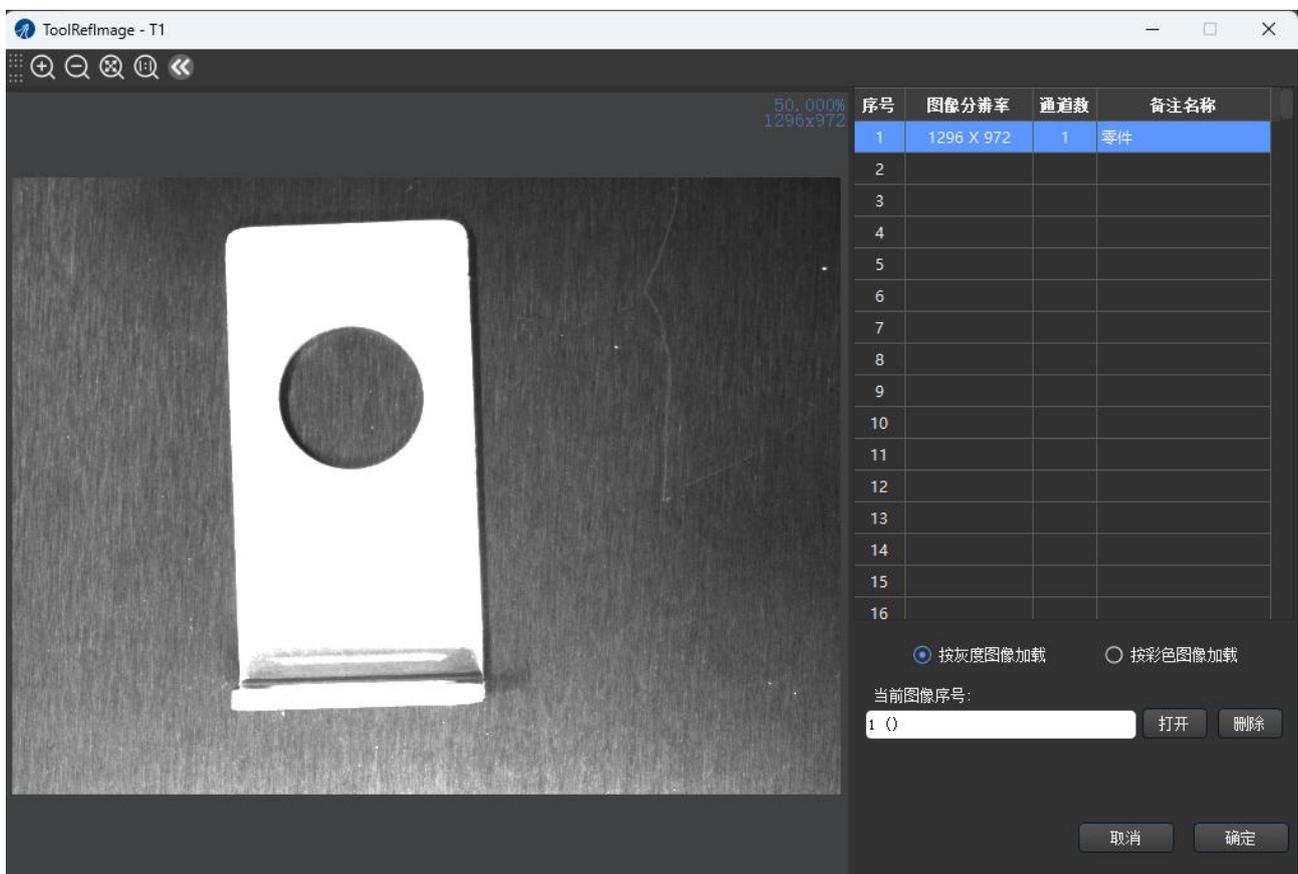


- 从站地址：ModbusRTU 从站的协议站地址。
- 其他参数配置以及读写寄存器选项页、心跳包选项页配置，与 ModbusTCP 基本一致，可参考 TCP 通信章节的参数介绍。

## 2.2.5 图像

### 基准图像

为图像处理工具设置 ROI 区域、MASK 区域提供图像源。集中式管理基准图像的加载、删除、命名等。视觉工程运行前，必须首先加载基准图像，流程中的图像处理工具需基于基准图像进行参数设置后才可正常运行。



具体功能：

- 图像的加载和删除，支持加载灰度和 RGB 彩色图像
- 基准图像信息显示
- 基准图像命名

## 2.3 快捷工具条

快捷工具条位于菜单栏下方，可以快速方便地对视觉方案进行相应的操作。



### (1) 权限登入

单击后进入登入界面，验证登录权限的密码进行登录确定。

### (2) 设置系统参数

单击后进入系统设置界面，用于设置软件启动界面、日志记录级别、用户操作权限等。

### (3) 配方管理

单击后进入配方管理界面，用于新建、删除、切换配方。

### (4) 系统变量

单击后查看当前工程路径、名称、相机数量、任务流程数等信息。

### (5) 全局变量

单击后可添加全局变量，全局变量可以在所有流程中访问和设置。

### (6) 基准图像

单击后进入基准图像界面，管理基准图像的加载、删除、命名等。

### (7) 进入坐标系标定界面

单击后进入标定流程界面，在此编写视觉标定流程。

### (8) 调试运行

在调试界面连续运行整个视觉方案。

### (9) 运行并进入运行界面

连续运行整个视觉方案，并将界面切到运行界面。

### (10) 复位所有流程

复位整个视觉方案的运行。

## 2.4 全局变量

全局变量是所有视觉工具外部定义的变量，它在整个工程内都有效，可以被工程内的所有流程调用或修改，可自定义变量名称、类型和初始值。在快捷工具条单击“”图标，可进行全局变量配置，支持添加 bool、int、float 和 string 类型的变量。



- 添加变量：单击后可新增全局变量。
- 上移/下移：可对变量的位置进行上下调整。
- 删除：可删除选中的变量。
- 取消：取消添加的全局变量。
- 确定：保存添加的全局变量。

## 2.5 工具列表

## 2.5.1 工具介绍

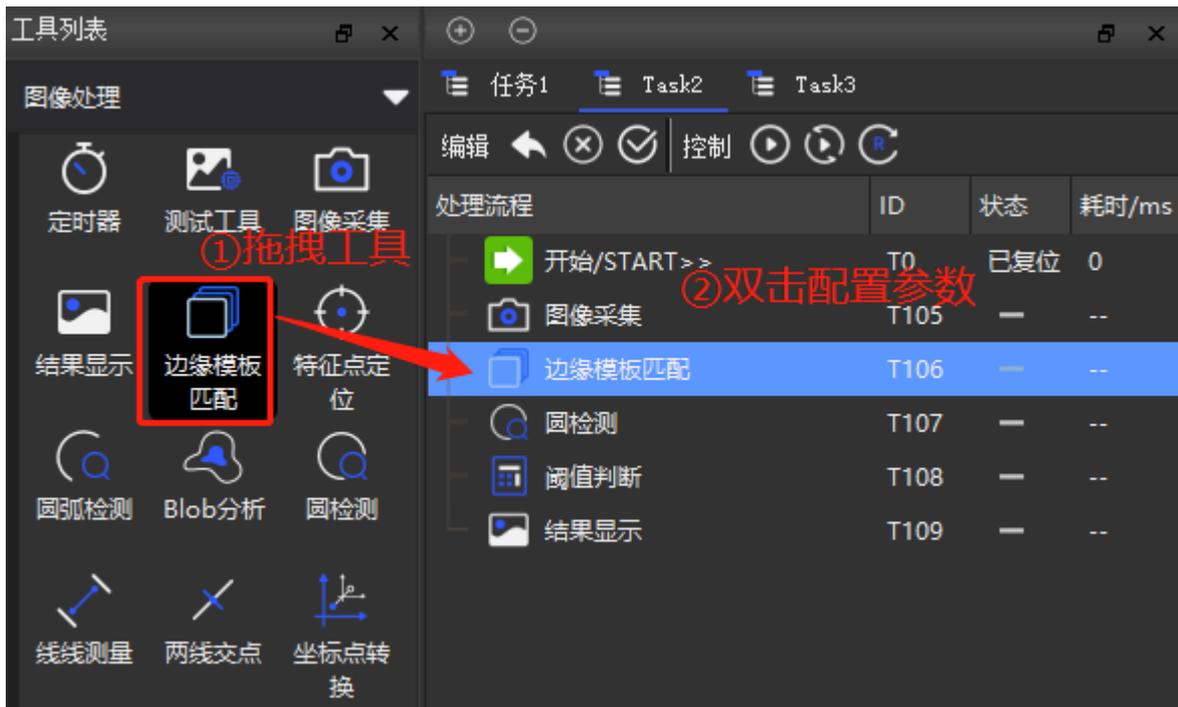


工具列表区的工具主要包括图像采集、图像处理、定位、测量、识别、生成工具、标定、颜色处理、逻辑处理和通信等模块。

- 图像采集：包括从相机中采集图像和保存图像。
- 图像处理：图像预处理类算子工具，如膨胀、腐蚀等形态学处理。
- 定位：模板匹配等特征提取工具，标识特征点用于定位。
- 测量：测量距离、夹角等基本测量功能。
- 识别：包含一维码、二维码识别等。
- 生成工具：直线拟合、圆拟合等。
- 标定：N点标定、测量标定等。
- 颜色处理：彩色图像的颜色提取、颜色识别、颜色测量等。
- 逻辑处理：视觉流程数据的逻辑处理。
- 通信：常用的工业通信协议，包括 TCP/IP、串口两种通信方式。

## 2.5.2 工具操作

鼠标停留在左侧对应工具列表栏，单击选中要使用的工具拖拉至流程编辑区域，按照项目逻辑需求合理放置工具先后顺序，双击配置参数即可，流程执行顺序为从上至下依次执行，如下图所示。



## 2.6 流程编辑栏

流程编辑栏用于编辑视觉方案的执行流程，可分成三部分，包括①流程选项卡、②操作工具栏和③流程编辑区。



## 2.6.1 流程选项卡

点击不同的选项卡可切换到不同的任务流程。单击“”图标可以添加任务流程，单击“”图标可以删除任务流程。任务流程名称默认为 Task1、Task2...依次往后排序，最大支持 8 个任务，如需修改任务名称，可双击任务名称修改，也可通过本软件菜单栏的“工程”->“工程设置”集中修改。



## 2.6.2 操作工具栏

操作工具栏分为编辑工具栏、调试工具栏和控制工具栏。

- 编辑工具栏



 撤销上一步操作

 删除选中的工具

 校验程序流程

- 调试工具栏



当在流程中设置断点后，运行流程时编辑工具栏将切换为调试工具栏。



删除断点



继续运行至下一个断点



单步调试运行

- 控制工具栏



运行一次本流程



连续运行本流程



复位本流程

### 2.6.3 流程编辑区

编辑区域主要用于编辑视觉流程，也可查看工具的运行状态、耗时、工具唯一 ID。点击可以选中工具，双击可以弹出工具的属性框，右击可以弹出选单（如下图）。工具状态列显示“”图标表示 工具执行 OK 状态，“”图标表示 工具执行结果 NG 状态，“”图标表示 工具未准备好 状态，“”图标表示 工具未执行 状态。

处理流程	ID	状态	耗时/ms
开始/START >>	T0	已暂停	3
图像采集	T105	✓	44.2
边缘模板匹配	T106	✗	25.5
圆检测	T107	✓	1.6
阈值判断			0.0
结果显示			0.0

功能	图标
删除	⊗
编辑名称	
添加备注信息	
设置断点	●
清除所有断点	⊗
全部折叠	⊞
全部展开	⊞
仅显示备注信息	

任务流程右键菜单功能：

删除：删除当前选中的工具。

编辑名称：修改当前选中工具的名称。

添加备注信息：给工具添加注释说明信息。

设置断点：给当前选中的工具设置断点。

清除所有断点：清除当前流程的全部断点。

全部折叠：折叠当前流程的所有节点，便于浏览程序。

全部展开：展开当前流程的所有节点。

仅显示备注信息：隐藏“状态”、“耗时”列，仅显示“处理流程”和“注释列”。

## 2.7 结果显示

通过主界面的图像显示区和工具结果显示区，可查看图像和工具运行结果信息。

### 2.7.1 图像显示区

显示流程中当前选中工具的图像信息。



- ① 工具栏：可对预览窗口的图像进行简单操作，包括缩小、放大、自适应、所有图像窗口自适应、保存图像、单次采集。
- ② 图像显示：显示流程中当前选中工具的图像。
- ③ 图像列表：当流程中的图像源为本地图像时，可通过该区域预览本地加载的图像并进行相关设置。
  - 复位：复位图像到第一张。
  - 循环读取：循环切换当前已加载的图像。
  - 遍历一次：仅运行一遍已加载的图像
  - 单张循环：仅运行勾选的图像。

## 2.7.2 工具结果显示区

工具的当前输出结果，便于用户分析查看。不同工具的输出结果有所差别，具体结果数据信息以各工具实际显示为准。

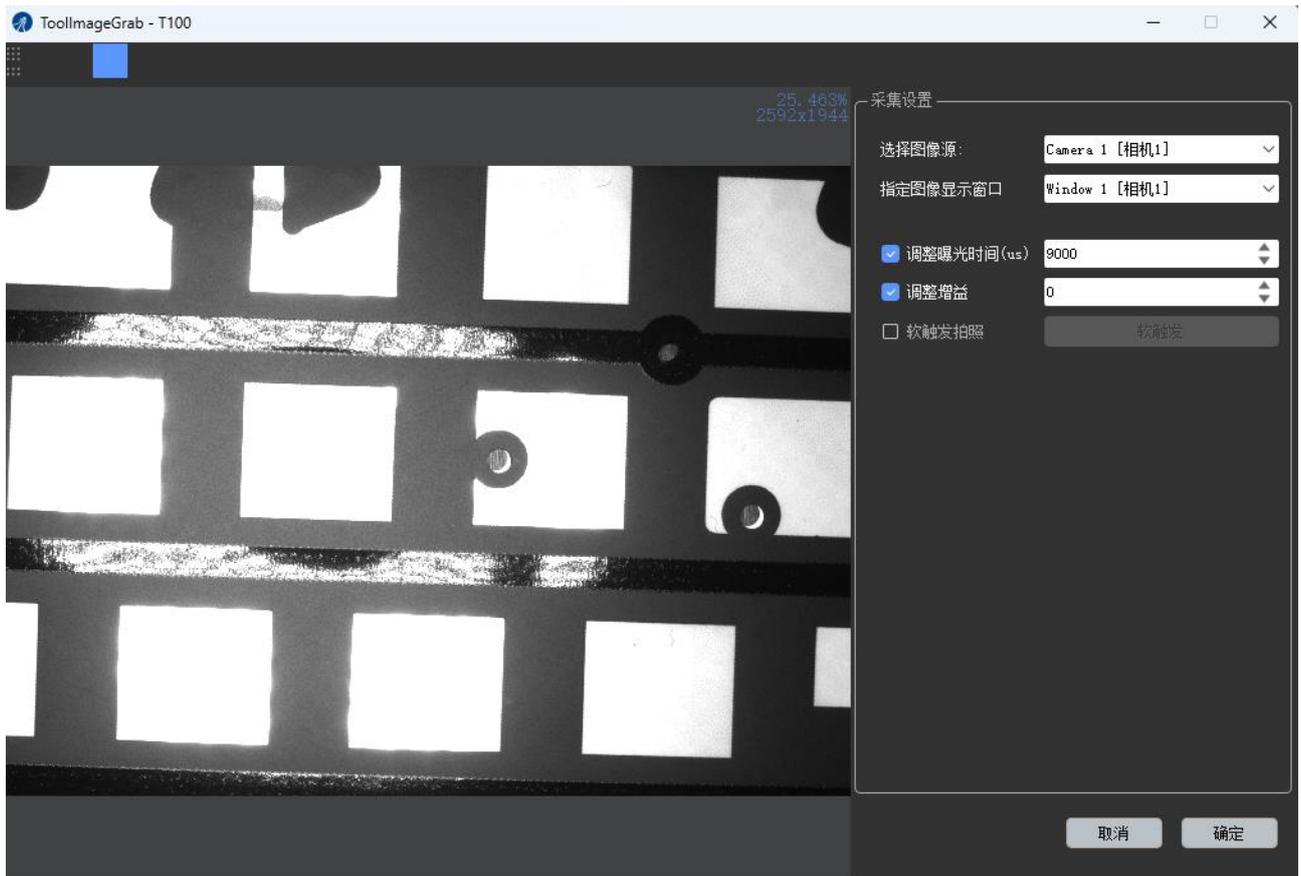
工具结果			
名称	描述	类型	值
Name	工具名称	string	圆检测
ID	工具ID	int	107
bitsOK	判断结果	bool	true
cx	圆心x坐标	float	2897.98
cy	圆心y坐标	float	3155.22
radius	圆半径	float	193.359
rltCircle	圆参数	DvCircle	78b7bab0
pointsNum	有效边缘点个数	int	64
validPointsRatio	有效边缘点占比	float	88.8889

# 3. 算子使用说明

## 3.1 图像采集

### 3.1.1 采集图像

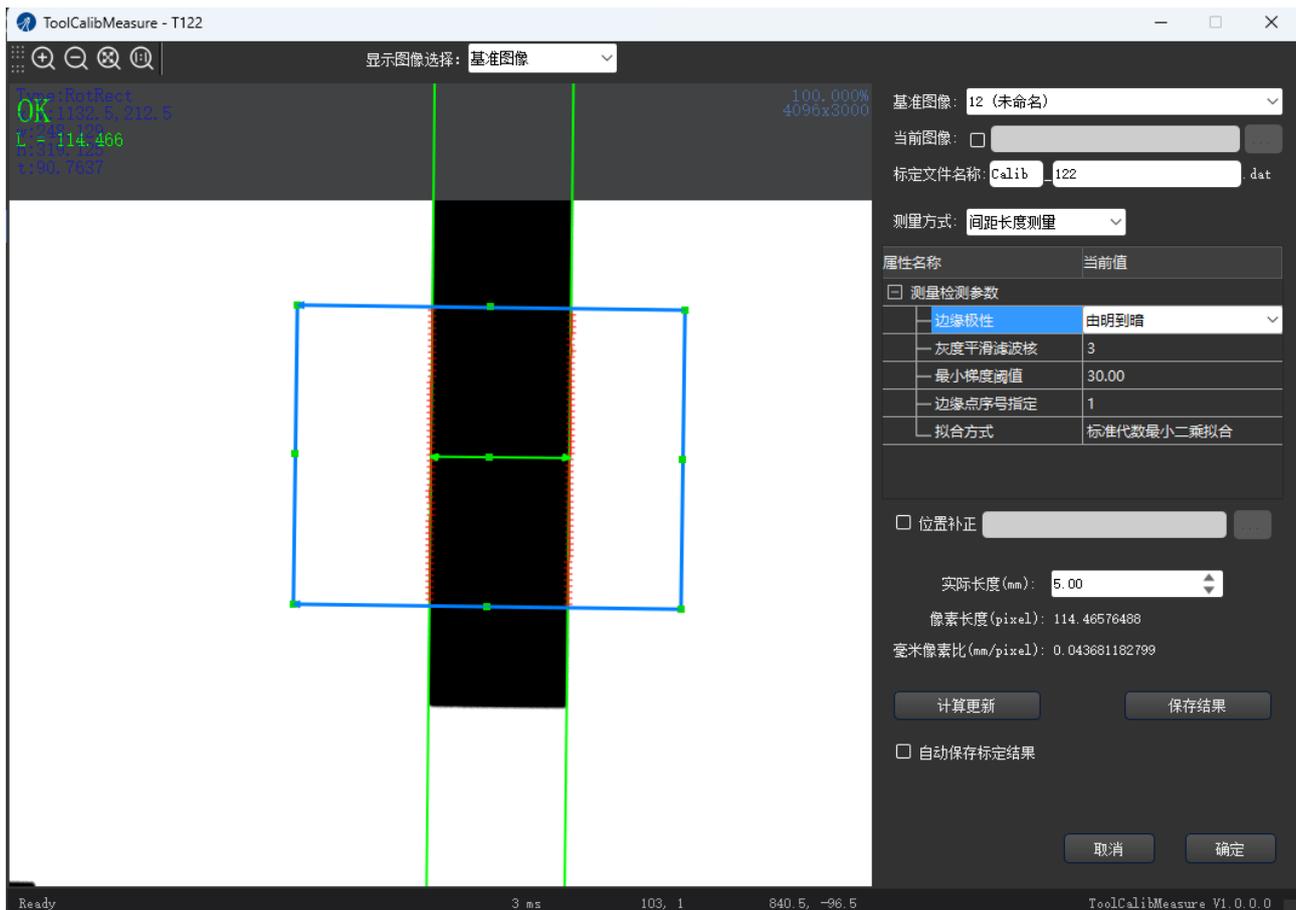
选择图像来源并指定该图像显示的窗口，图像源可以从本地文件夹或者相机获取，当图像来源为相机获取时，可调整相机的曝光时间和增益。



## 3.2 标定

### 3.2.1 测量标定

通过测量已知尺寸的标准块，计算出像素单位到物理单位的转换比例，通常作为测量项目的标定。



- 基准图像：选择执行测量标定的基准图像，基准图像在菜单栏“图像”->“基准图像”中设置。选择基准图像后，会自动添加一个矩形检测 ROI 或圆形 ROI。
- 当前图像：算子运行时的当前图像，一般绑定图像采集、图像处理等工具的输出图像。
- 标定文件名称：设置标定结果保存的文件名称。
- 测量方法：可以选择间距长度测量或圆半径测量。
- 测量方向：圆半径测量方法的参数，指定每个卡尺查找点的方向，有由圆心向圆外和由圆外向圆心两种模式。
- 测量宽度：圆半径测量方法的参数，卡尺的搜索长度，决定卡尺检测区域的纵向长度，即圆环宽度。
- 边缘极性：由明到暗、由暗到明和任意极性 3 种模式，由明到暗表示从亮度高过度到亮度低的边缘，由暗到明表示从亮度低过度到亮度高的边缘，任意极性则明到暗和暗到明的边缘都检测。
- 灰度平滑滤波核：垂直于卡尺测量线方向上的滤波核大小，像素单位。如滤波核设置为 3，则在该方向上使用了 3 个像素进行灰度均值计算。当边缘有噪音干扰时，可增大滤波核。

- 最小梯度阈值：设定检测边缘梯度阈值的下限，只有边缘梯度阈值大于该值的边缘点才被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除，取值范围 1~255。
- 边缘点序号指定：边缘点由多个卡尺卡出，该参数指定每个卡尺要找的边缘点序号，默认序号为 1，即第一条边，当 ROI 区域内存在多条边缘时，可修改该序号调整要检测的边缘。
- 拟合方式：拟合方式有标准代数最小二乘和带权重的最小二乘两种。
- 位置修正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置修正，一般绑定边缘模板匹配的仿射变换输出。
- 实际长度：设置测量目标的实际尺寸。当测圆形时，实际尺寸为半径。
- 像素长度：被测量目标的像素尺寸。
- 毫米像素比例：被测量目标的实际长度与像素长度的比值。
- 计算更新：执行标定计算出标定结果。
- 保存结果：保存标定结果，也可自动保存标定结果。

### 3.2.2 标量转换

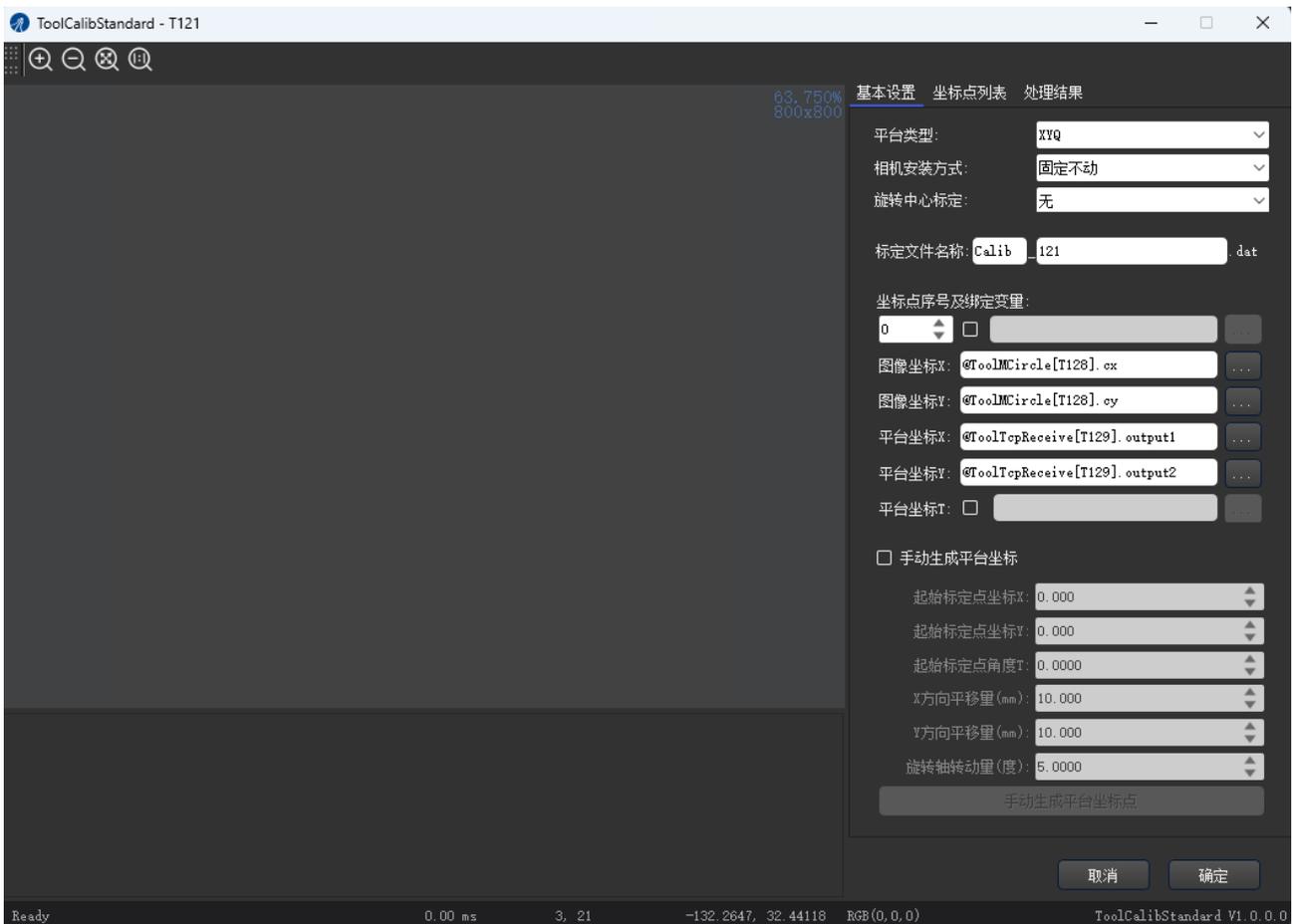
将输入的一组间距数据从图像坐标系转换到世界坐标系，可转换距离、宽度等像素单位到物理单位，可在“工具结果”查看转换后的标量值。



- 标定数据来源：加载“测量标定”工具生成的标定文件。
- 图像坐标系下标量值：设置需要转换的图像间距值，可关联。

### 3.2.3 N 点标定

根据配置的 N 组坐标点对（图像坐标—世界坐标），计算图像坐标映射到世界坐标的变换矩阵并输出标定文件。N 组数据可包括 9 组平移坐标和 7 组旋转坐标。一般用于一个 Mark 点 N 次拍照的标定。



该算子工具共 3 个设置页，分别为：基本设置、坐标点列表、处理结果。

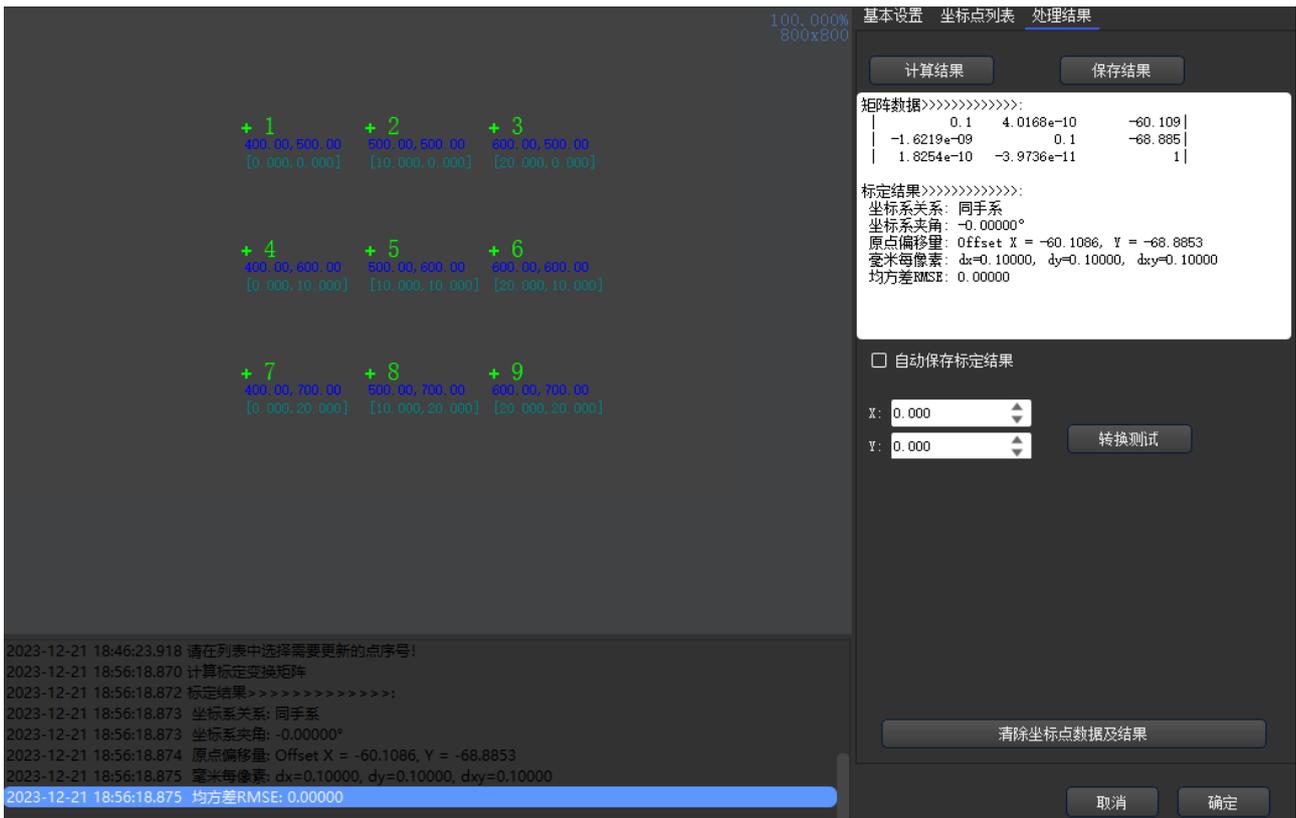
- 基本设置页

用于配置标定平台类型。

- 平台类型：支持 XY $\theta$ 、X $\theta$ +Y、Y $\theta$ +X 平台，其中 XY $\theta$  平台包含 XY 平台。
- 相机安装方式：可选固定不动、随 XY 移动、随 X 移动、随 Y 移动。
- 旋转中心标定：包括拟合圆法找旋转中心、三角函数法找旋转中心（精确）。当选择“无”时，则 XY $\theta$  平台为 XY 平台。
- 标定文件名称：设置标定结果保存的文件名称。

- 坐标点序号及绑定变量：N 点坐标数据的编号，范围[1~16]，可关联，平台移动后可通过通信工具发送此编号。
- 图像坐标 X/Y：关联图像的 X/Y 坐标。平台每次移动后都需要相机进行一次拍照，并将特征点的图像坐标 X/Y 插入到“坐标点列表”页的图像坐标中。将 X 和 Y 分别关联到对应测量工具可快速插入数据，无需手动输入。
- 平台坐标 X/Y/T：平台的物理 X/Y 坐标和角度 T（单位度）。平台每次移动后都需要相机进行一次拍照，并将平台物理坐标 X/Y 和角度 T 插入到“坐标点列表”页的平台坐标中，可关联通信接收到的平台物理坐标。
- 手动生成平台坐标：勾选后使用下方手动设置的平台坐标。
- 起始标定点坐标 X/Y/T：N 点标定第 5 个点的平台物理 X/Y 坐标和角度 T（单位度）。
- X/Y 方向平移量：每次平台的移动量。
- 旋转轴转动量：旋转标定时，每次转动的角度（单位度）。
- 手动生成平台坐标点：点击后，应用平台设置的坐标数据，将在“坐标点列表”页的平台坐标中，自动生成平台坐标。
- 坐标点列表页  
图像坐标和平台物理坐标的对应点，用于生成坐标系的变换矩阵。

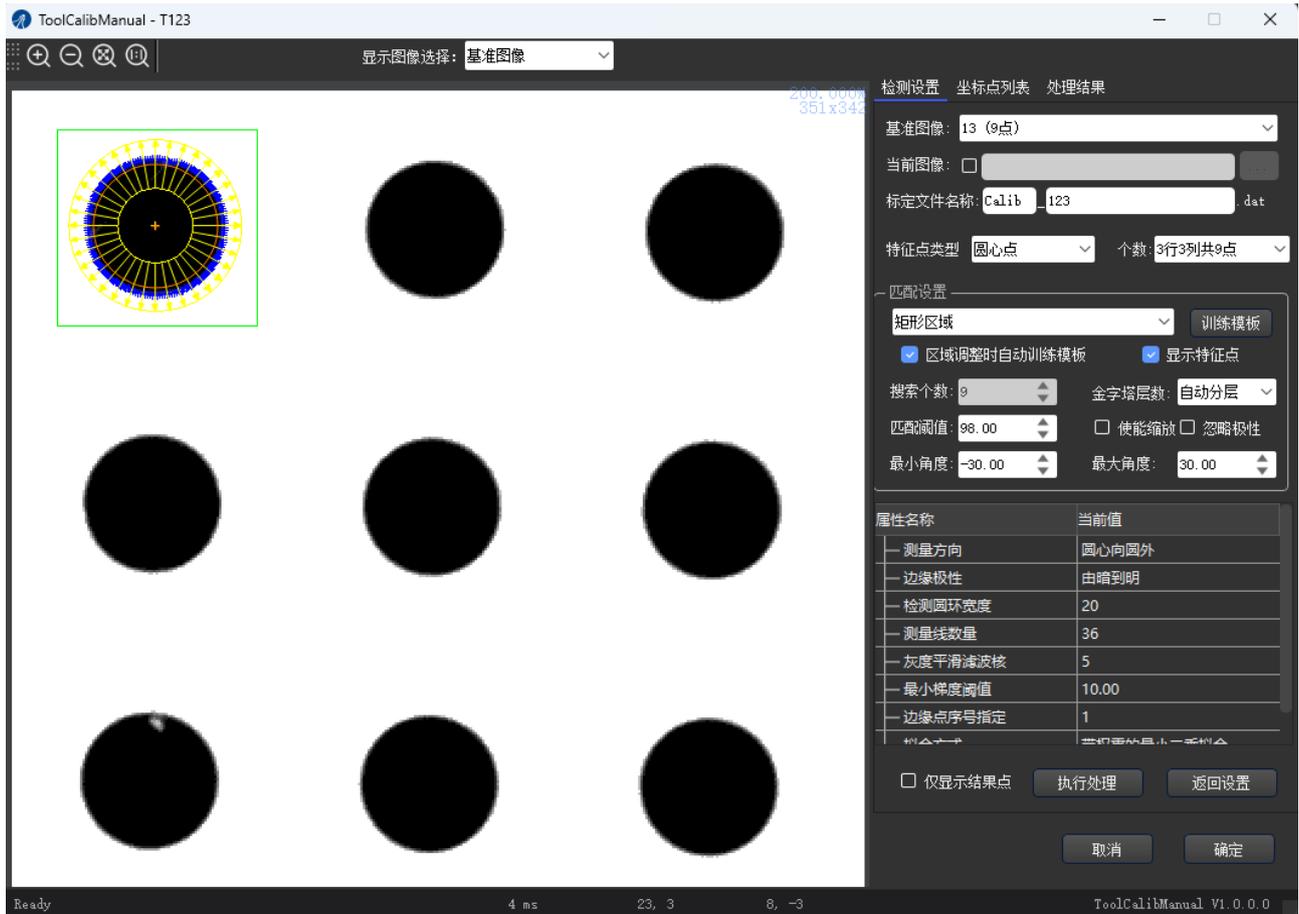




- 计算结果：生成标定矩阵数据。
- 保存结果：保存标定结果，也可勾选自动保存标定结果。
- 转换测试：输入图像坐标点，测试转换的平台物理坐标，验证标定是否正确。
- 清除坐标点数据及结果：清空坐标点及标定矩阵数据。

### 3.2.4 手动标定

通过构建一系列图像坐标和平台物理坐标的对应点，计算图像坐标系到平台物理坐标系之间的变换矩阵关系，并输出标定文件。一般用于 N 个 Mark 点一次拍照的标定。



该算子工具共 3 个设置页，分别为：检测设置、坐标点列表、处理结果。

### ● 检测设置页

用于配置检测目标的特征点参数，目标图形可以为圆形格、黑白格或圆形十字架等。

**基准图像：**选择建立检测特征点的基准图像，基准图像在菜单栏“图像”->“基准图像”中设置。

- **当前图像：**算子运行时的当前图像，一般绑定图像采集、图像处理等工具的输出图像。
- **标定文件名称：**设置标定结果保存的文件名称。
- **特征点类型：**可以选择模板匹配对象的中心点、圆心对象的圆心点、黑白格对象的直角点。当选择“圆心点”时，会自动添加圆环 ROI，用于检测圆形边缘点；当选择“直角点”时，会自动添加 2 个矩形 ROI，用于检测两条直线。
- **个数：**可以限定检测的目标个数为 9 个或不做限制。
- **匹配设置：**选择模板矩形 ROI 形状后，将在图像区自动添加对应的 ROI，将模板区域 ROI 拖动到需要作为模板的图像区域，一般与“特征点类型”选择同一个区域。
- **训练模板：**点击时生成一次模板。

- 区域调整时自动训练模板：勾选后，调整 ROI 区域自动生成新的模板。
- 显示特征点：勾选后，显示模板轮廓点。
- 搜索个数、金字塔层级、匹配阈值、最小角度、最大角度、使能缩放、忽略极性搜索模板参数设置，请参考[模板匹配](#)工具介绍章节。
- 检测属性设置：当选择“圆心点”时，请参考[圆查找](#)工具介绍章节。当选择“直角点”时，请参考[直线查找](#)章节。
- 执行处理：按上述设置的参数，执行一次本工具，得到特征点结果。
- 返回设置：返回到 ROI 设置状态，重新设置检测区域。
- 仅显示结果点：只显示找到的特征点。
- 坐标点列表页

构建图像坐标和平台物理坐标的对应点，用于生成坐标系的变换关系。

序号	x[图像]	y[图像]	X[平台]	Y[平台]
1	58.47	55.64	10.000	10.000
2	173.21	56.89	20.000	10.000
3	287.99	58.22	30.000	10.000
4	286.80	172.79	30.000	20.000
5	171.96	171.47	20.000	20.000
6	57.19	170.14	10.000	20.000
7	55.01	284.69	10.000	30.000
8	170.81	285.99	20.000	30.000
9	285.60	287.28	30.000	30.000

手动生成平台坐标

第1点坐标: X: 10.000 Y: 10.000

第2点坐标: X: 20.000 Y: 10.000

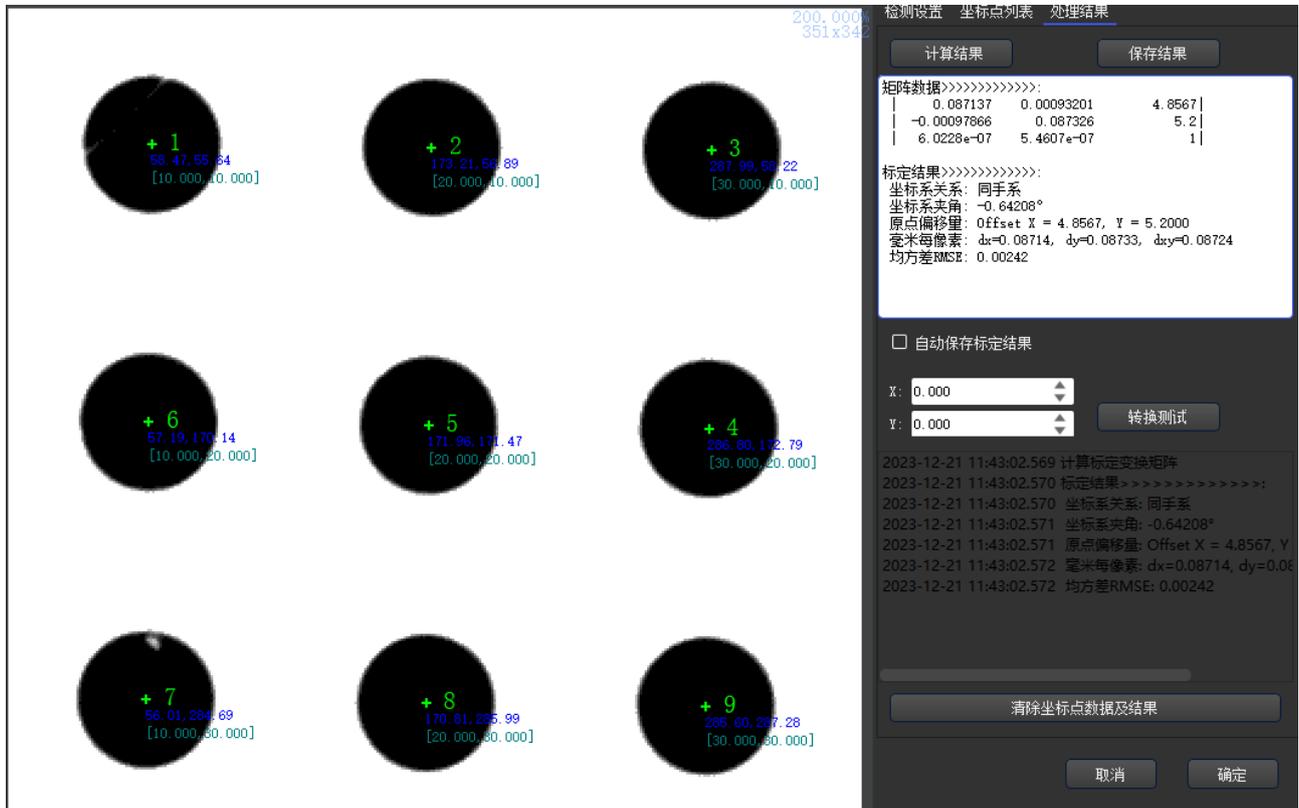
第3点坐标: X: 10.000 Y: 20.000

- 更新图像坐标点：将检测到的特征点坐标更新到表格。
- 显示坐标点：在左侧图像显示区显示所有点的坐标。
- 手动生成平台坐标：未勾选时，所有图像上的点均需移动平台对点，对点完成后，手动输入每一个图像坐标对应的平台实际物理坐标。勾选后，左侧图像显示区将标注 3 个特征点，平台移动到对应序号的位置，记录 3 点实际物理坐标，填入下方的第 1 点坐标~第 3

点坐标中，然后点击“手动生成平台坐标点”，剩余平台坐标将自动填充，用户只需对点3次即可。

- 处理结果页

计算标定结果并保存。



- 计算结果：生成标定矩阵数据。
- 保存结果：保存标定结果，也可勾选自动保存标定结果。
- 转换测试：输入图像坐标点，测试转换的平台物理坐标，验证是否正确。
- 清除坐标点数据及结果：清空坐标点及标定矩阵数据。

### 3.2.5 坐标转换

将输入的一组坐标数据从图像坐标系转换到世界坐标系，可在“工具结果”查看转换后的坐标值。

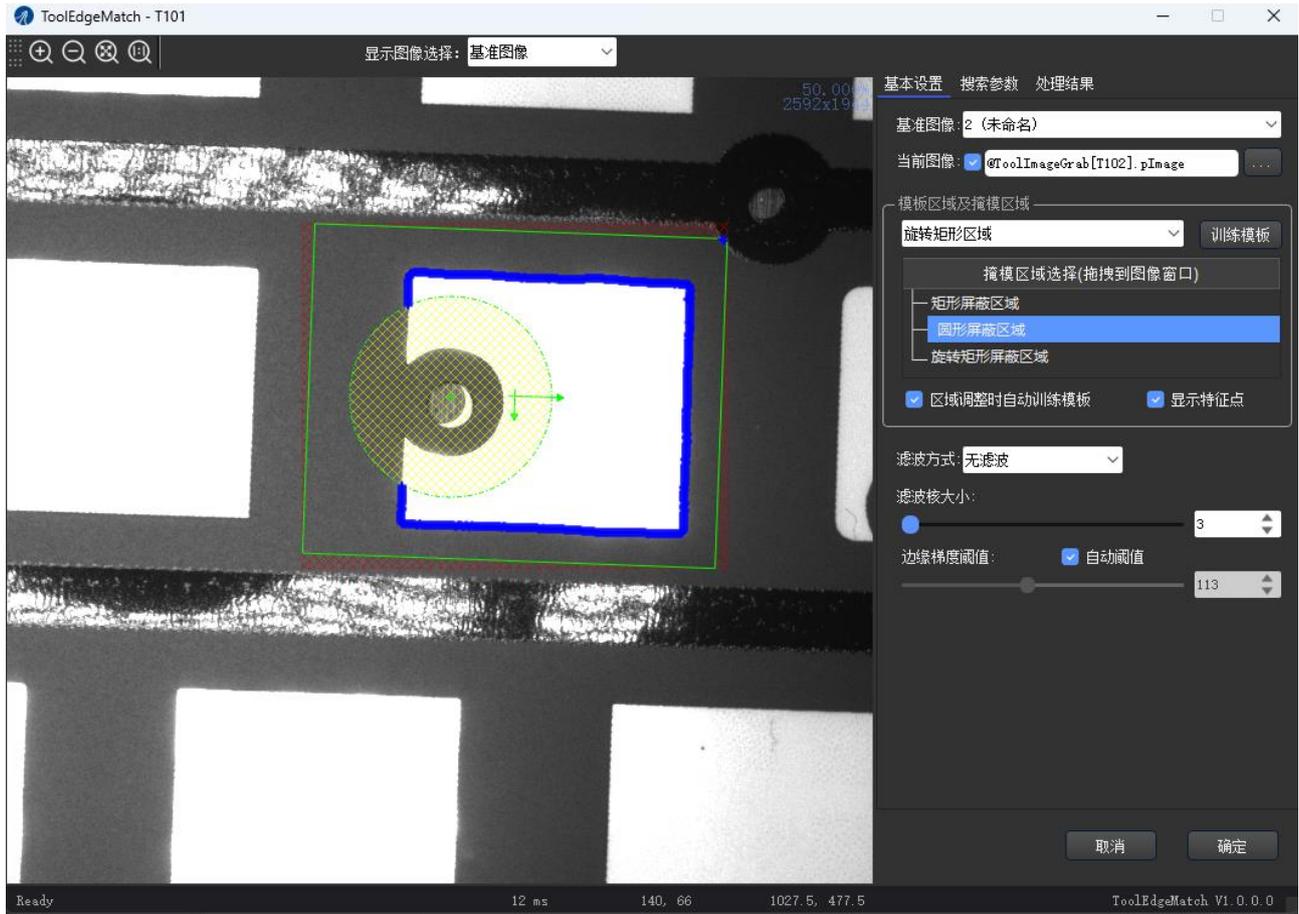


- 标定数据来源：加载标定工具生成的标定文件。
- 图像坐标 X：设置需要转换的坐标 X 轴数据，可关联。
- 图像坐标 Y：设置需要转换的坐标 Y 轴数据，可关联。
- 图像坐标系下角度：设置需要转换的角度数据，可关联。

## 3.3 定位

### 3.3.1 轮廓匹配

使用图像的边缘轮廓特征作为模板，按照预设的参数确定搜索空间，在图像中搜索与模板相似的目标，允许目标存在旋转、缩放、亮度变化、模糊、遮挡等情况，可用于定位、计数和判断有无等。



该算子工具共 3 个设置页，分别为：基本设置、搜索参数、处理结果。

- 基本设置页

主要用于配置创建模板时的参数。

- 基准图像：选择建立模板的基准图像，基准图像在菜单栏“图像”->“基准图像”中设置。
- 当前图像：算子运行时搜索模板的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 模板区域及掩膜区域：选择模板 ROI 形状（矩形、圆、旋转矩形）后，将在图像区自动添加对应的 ROI，将模板区域 ROI 拖动到需要作为模板的图像区域，如果需要屏蔽干扰区域，将掩膜区域 ROI 拖动到需要屏蔽的图像区域。
- 训练模板：点击时生成一次模板。
- 区域调整时自动训练模板：勾选后，调整 ROI 区域自动生成新的模板。
- 显示特征点：勾选后，显示模板轮廓点。
- 滤波方式：对图像进行滤波（高斯、中值、边缘保持）操作。
- 显示滤波效果：查看滤波后的图片效果。

- 边缘梯度阈值：可选自动和手动两种模式，建议勾选自动模式，当自动模式不满足要求时再切换到手动模式调整，范围 1~255，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，抗噪声能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除。
- 备注：模板搜索范围默认为全图像搜索，可通过调整图像显示区域最外边的矩形 ROI，将搜索区域限定在设置的 ROI 矩形区域内。

创建模板流程如下：

Step1：在“基准图像”下拉列表中，选择一张基准图像用于建立模板使用。并在“显示图像选择”下拉列表框中，选中基准图像显示。

Step2：在“模板区域及掩膜区域”下拉列表框中，选择 ROI 形状，并将 ROI 拖动到需要作为模板的图像区域。如果需要屏蔽模板区域的干扰区，可添加掩膜区域，将掩膜 ROI 拖动到需要屏蔽的图像区域。

Step3：根据实际效果，设置滤波方式、边缘梯度阈值。

Step4：点击训练模板，完成模板创建并观察模板效果，建议勾选区域调整时自动训练模板和显示特征点，便于实时观察 ROI 区域拖动中提取的模板特征。

Step4：点击确认按钮，保存模板。

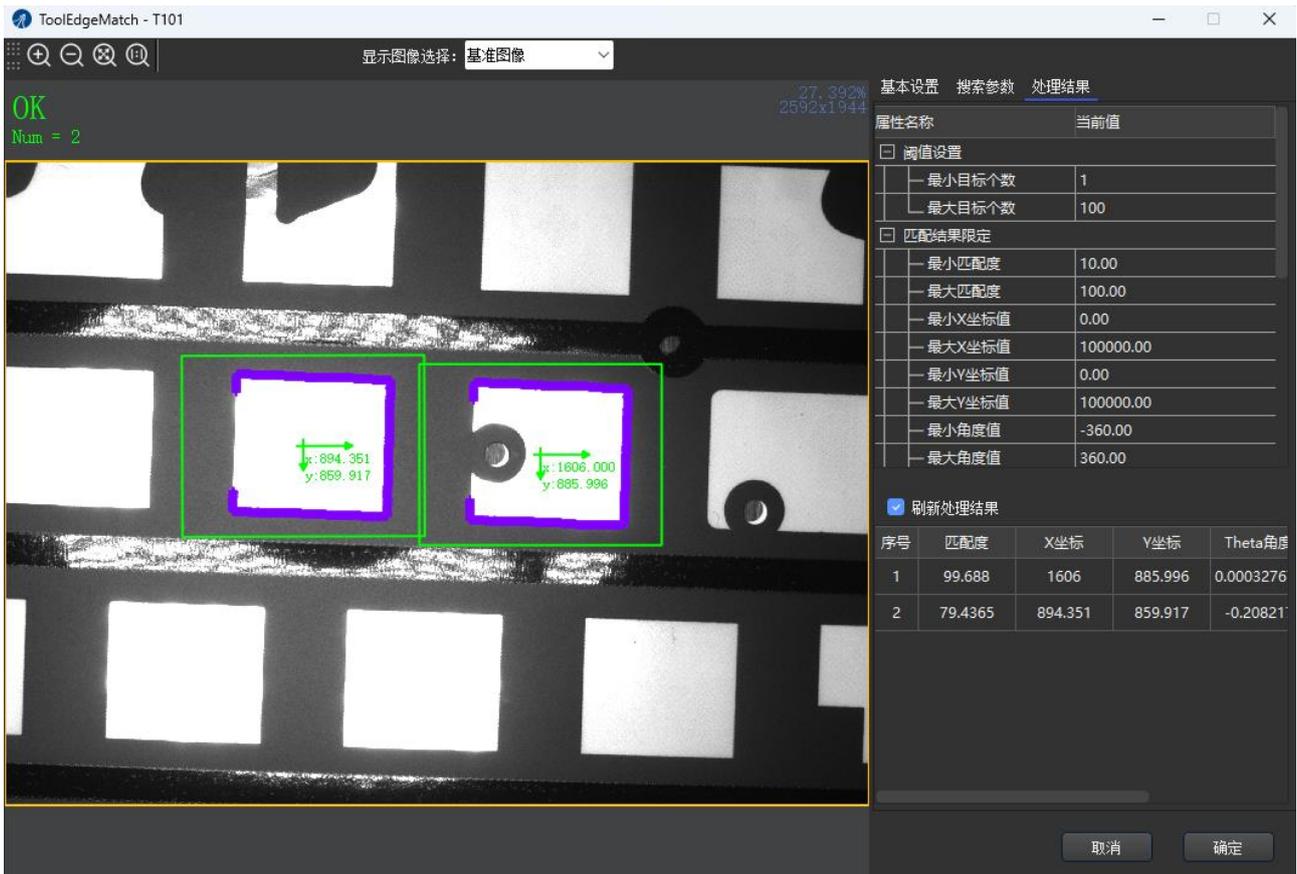
#### ● 搜索参数页

The screenshot shows a software interface with three tabs: '基本设置' (Basic Settings), '搜索参数' (Search Parameters), and '处理结果' (Processing Results). The '搜索参数' tab is active. The configuration options are as follows:

- 搜索个数: 1 (spin box)
- 金字塔层数: 自动分层 (dropdown)
- 匹配阈值: 50.00 (spin box)
- 搜索速度: 标准 (dropdown)
- 允许遮挡率: 0.20 (spin box)
- 使能加速
- 使能缩放
- 忽略极性
- 搜索角度范围:
  - 最小角度: -30.00 (spin box)
  - 最大角度: 30.00 (spin box)
- 允许缩放比:
  - 最小比例: 0.95 (spin box)
  - 最大比例: 1.05 (spin box)
- 角度输出范围:  [0, 360]  [-180, 180]
- 排序方式: 匹配度降序 (dropdown)
- 参数如有调整，自动执行匹配

可以修改运行时的匹配参数，以提高匹配成功率。比如，当搜索图像与模板相似度低时，可调低匹配阈值，提高匹配机会，但这也会增加错误匹配的机会。再比如，当匹配速度慢时，可将金字塔级别提高，以加快形状轮廓检测速度。

- 搜索个数：设置允许查找的最大目标个数，识别结果的数量将小于或等于该数量。
- 匹配阈值：模板与搜索图像中目标的相似程度，即相似度阈值，搜索时在相似度达到该阈值时才会被确认为目标，最大是 100，表示完全契合。
- 允许遮挡率：当搜索的目标存在遮挡时，匹配对象所允许的最大遮挡比例，遮挡率小于该值的对象将被正常匹配，范围 0~0.75。
- 金字塔层级：用于加速匹配过程，在目标面积较大时可以适当调高，级别越高，速度越快，但可能会降低识别能力，级别越小精度越高。
- 搜索速度：设置匹配过程中的匹配速度等级。搜索速度值越大速度越快，但可能会导致识别率降低，需要根据实际情况设置，在保证能够识别到目标的情况下再调大。
- 使能加速：通过减小匹配精细度，提升匹配速度。如模板的轮廓特征太多，勾选后将会自动减少特征点从而提高匹配速度。
- 使能缩放：使能搜索过程中最大最小缩放比例。
- 忽略极性：不勾选，待识别目标的边缘方向跟模板一致；勾选后，待识别目标的边缘方向跟模板一致或相反都可以，可用于目标和模板之间存在局部光照变化的情况，该模式会花费更多的时间。
- 最小角度：待识别目标相对于已创建模板的最小旋转角度，取值范围  $0\sim 360^\circ$  或  $-180^\circ\sim 180^\circ$ 。
- 最大角度：待识别目标相对于已创建模板的最大旋转角度，取值范围  $-0\sim 360^\circ$  或  $-180^\circ\sim 180^\circ$ 。
- 最小比例：待识别目标尺寸相对于模板尺寸的最小缩放比例，取值范围 0.8~1。
- 最大比例：待识别目标尺寸相对于模板尺寸的最大缩放比例，取值范围 1~1.2。
- 角度输出范围：限定模板输出的角度范围在  $0\sim 360^\circ$  或  $-180^\circ\sim 180^\circ$ 。
- 排序方式：当存在多个相同目标图形时，用于设置匹配结果的排序方式。
- 参数如果调整自动执行匹配：当修改参数时自动执行一次匹配。
- 处理结果页

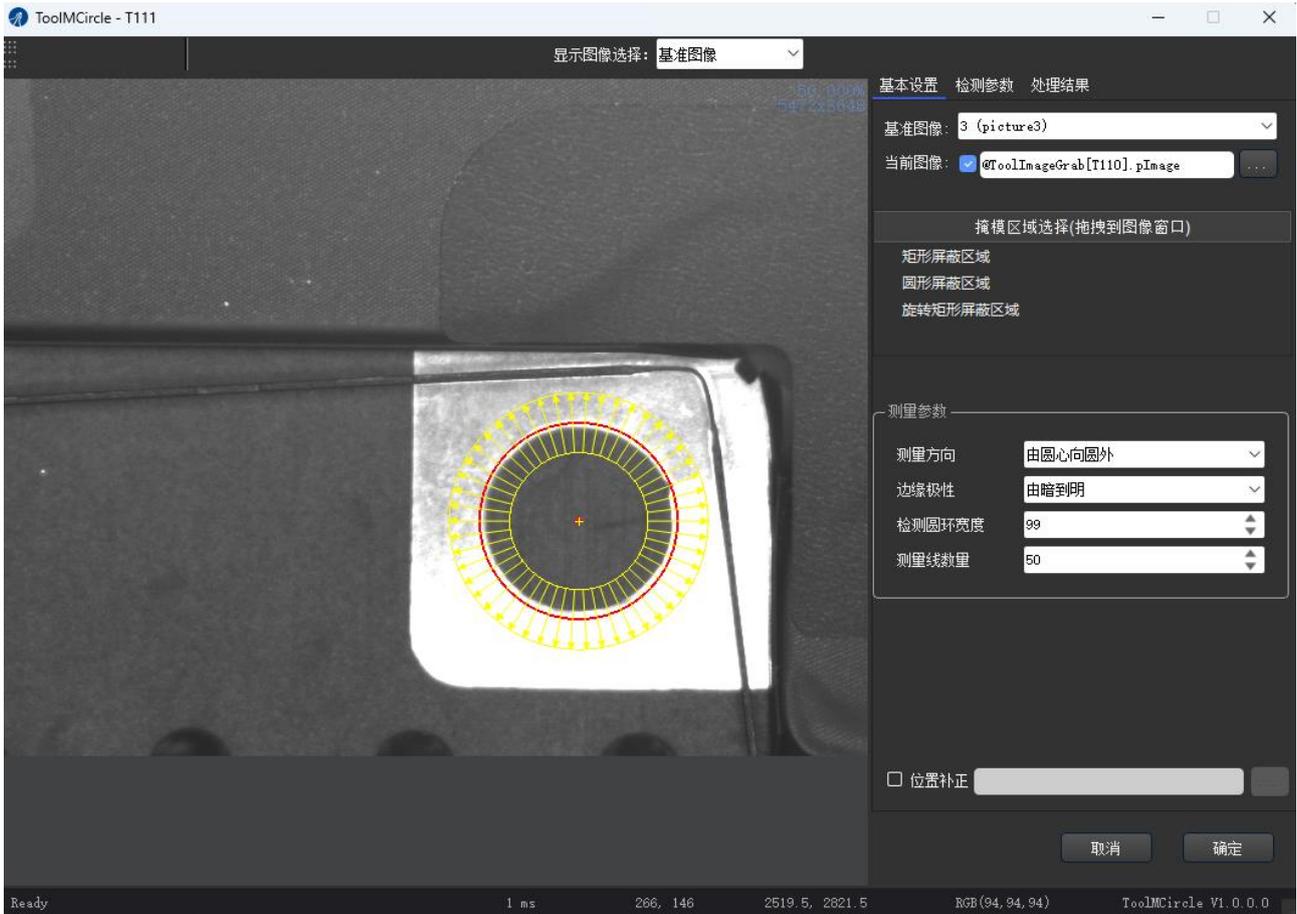


查看匹配结果及设置结果阈值判定范围。

- 阈值设置：当匹配目标个数、匹配度、X 坐标范围、Y 坐标范围、角度范围、缩放比范围均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：设置目标矩形框颜色、匹配目标特征点颜色、搜索区域显示颜色及线宽。
- 匹配度：匹配目标与模板的相似度。
- 坐标：匹配目标相对于当前图像的 XY 坐标位置。
- Theta 角度：匹配目标相对于模板的旋转角度。
- 缩放比：匹配目标相对于模板的缩放比例。

### 3.3.2 圆查找

从设定圆环形范围内检测符合条件的边界点，利用所有被检测的点拟合出一个圆，得到圆心坐标和半径。可用于圆的定位和测量。



该算子工具共 3 个设置页，分别为：基本设置、检测参数、处理结果。

### ● 基本设置页

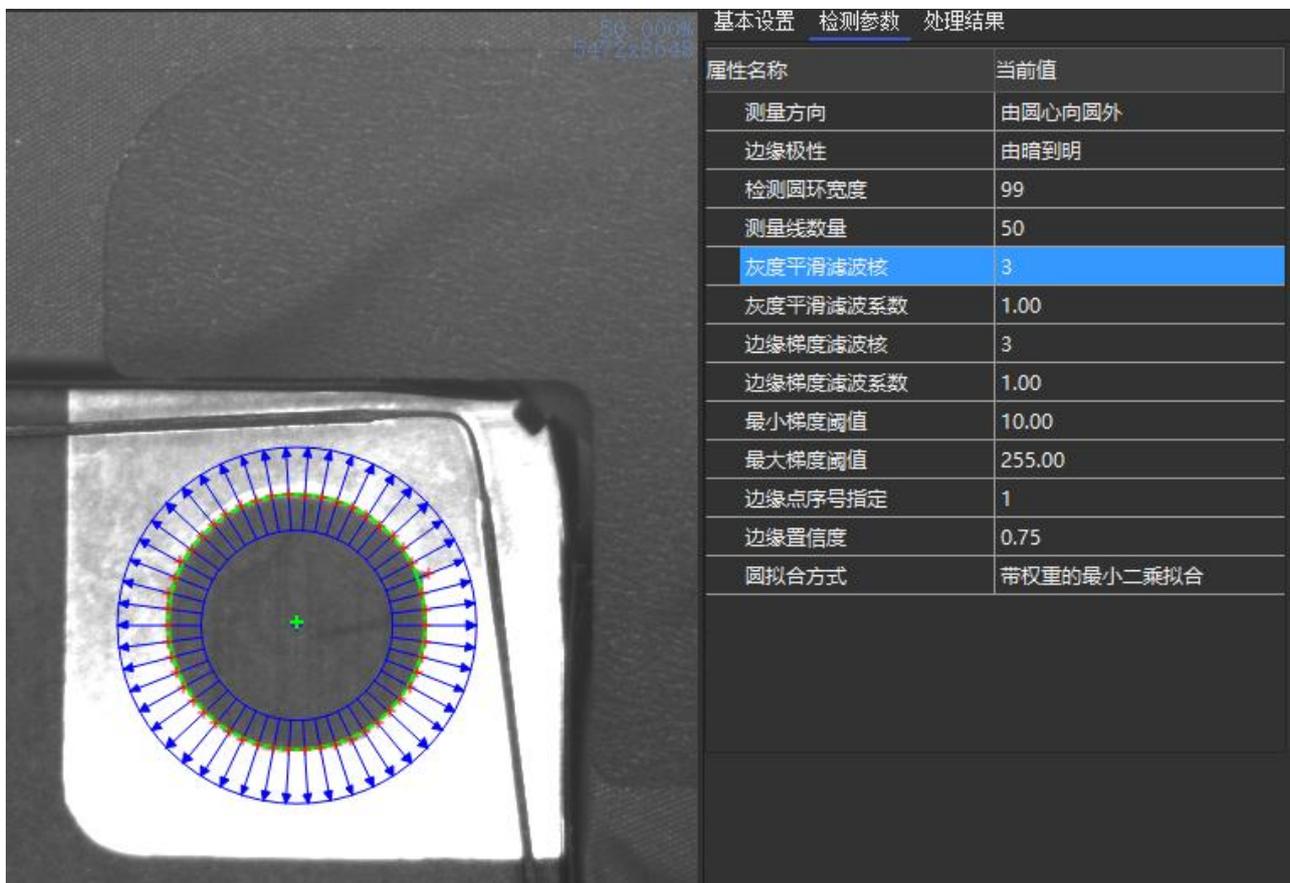
主要用于设置圆查找的测量参数。

- 基准图像：选择执行圆查找的基准图像，基准图像在菜单栏“图像”->“基准图像”中设置。选择基准图像后，会自动添加一个圆形检测 ROI。
- 当前图像：算子运行时检测圆形的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 测量方向：指定每个卡尺查找点的方向，有由圆心向圆外和由圆外向圆心两种模式。
- 边缘极性：由明到暗、由暗到明和任意极性 3 种模式，由明到暗表示从亮度高过度到亮度低的边缘，由暗到明表示从亮度低过度到亮度高的边缘，任意极性则明到暗和暗到明的边缘都检测。
- 检测圆环宽度：卡尺的搜索长度，决定卡尺检测区域的纵向长度，取值范围 5~99。
- 测量线数量：即卡尺的个数，决定找点的区域个数，每个卡尺里找到一个满足的点。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要

对目标进行定位，此时需要将检测该目标的圆形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定边缘模板匹配的仿射变换输出。

## ● 检测参数页

圆检测工具运行时的参数配置。



The screenshot displays a software interface for circle detection. On the left, a grayscale image of a ruler is shown with a circular region of interest (ROI) overlaid. The ROI is defined by a green circle with a red center, and blue arrows radiate from the center to the edge. On the right, a table titled '基本设置 检测参数 处理结果' (Basic Settings Detection Parameters Processing Results) lists various parameters and their current values.

属性名称	当前值
测量方向	由圆心向圆外
边缘极性	由暗到明
检测圆环宽度	99
测量线数量	50
灰度平滑滤波核	3
灰度平滑滤波系数	1.00
边缘梯度滤波核	3
边缘梯度滤波系数	1.00
最小梯度阈值	10.00
最大梯度阈值	255.00
边缘点序号指定	1
边缘置信度	0.75
圆拟合方式	带权重的最小二乘拟合

➤ 灰度平滑滤波核：垂直于卡尺测量线方向上的滤波核大小，像素单位。如滤波核设置为 3，则在该方向上使用了 3 个像素进行灰度均值计算。当边缘有噪音干扰时，可增大滤波核。

➤ 灰度平滑滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。

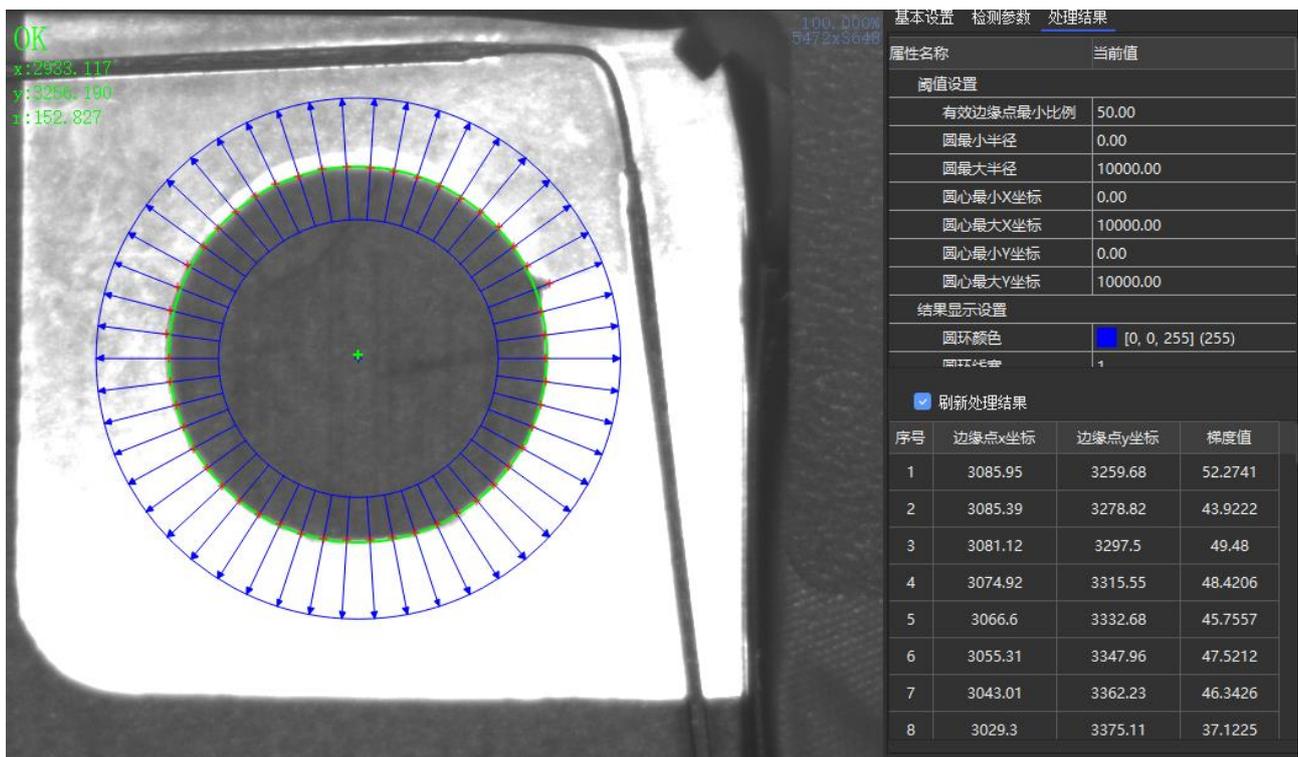
边缘梯度滤波核：卡尺测量线方向上的滤波核大小，用于增强边缘和抑制噪音干扰，最小值为 1，当边缘模糊不清晰或有噪音干扰时可以增大滤波核，这样可以使得检测结果更加稳定。

➤ 边缘梯度滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。

➤ 最小梯度阈值：设定检测边缘梯度阈值的下限，只有边缘梯度阈值大于该值的边缘点才被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除，取值范围 1~255。

- 最大梯度阈值：设定检测边缘梯度阈值的上限，只有边缘梯度阈值小于该值的边缘点才被检测到，取值范围 1~255。
- 边缘点序号指定：边缘点由多个卡尺卡出，该参数指定每个卡尺要找的边缘点序号，默认序号为 1，即第一条边，当 ROI 区域内存在多个圆时，可修改该序号调整要检测的圆。
- 边缘置信度：当边缘点序号设置为 0 时有效，边缘点梯度的权重占该卡尺区域梯度最大值的比重。
- 圆拟合方式：拟合方式有标准代数最小二乘、基于正交距离最小二乘和带权重的最小二乘三种。
- 处理结果页

查看圆查找结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的结果、拟合圆的圆心坐标和半径。



The screenshot displays a software interface for circle detection. On the left, a grayscale image shows a circular object with a fitted circle overlaid. The center of the circle is marked with a green crosshair. The coordinates of the center are displayed in green text: x: 2933.117, y: 3258.190, r: 152.827. On the right, a control panel is visible with tabs for '基本设置' (Basic Settings), '检测参数' (Detection Parameters), and '处理结果' (Processing Results). The '处理结果' tab is active, showing a table of edge points. The table has four columns: '序号' (Serial Number), '边缘点x坐标' (Edge Point X Coordinate), '边缘点y坐标' (Edge Point Y Coordinate), and '梯度值' (Gradient Value). The table contains 8 rows of data. Below the table, there are checkboxes for '刷新处理结果' (Refresh Processing Results) and '圆环颜色' (Circle Ring Color), which is set to blue [0, 0, 255] (255).

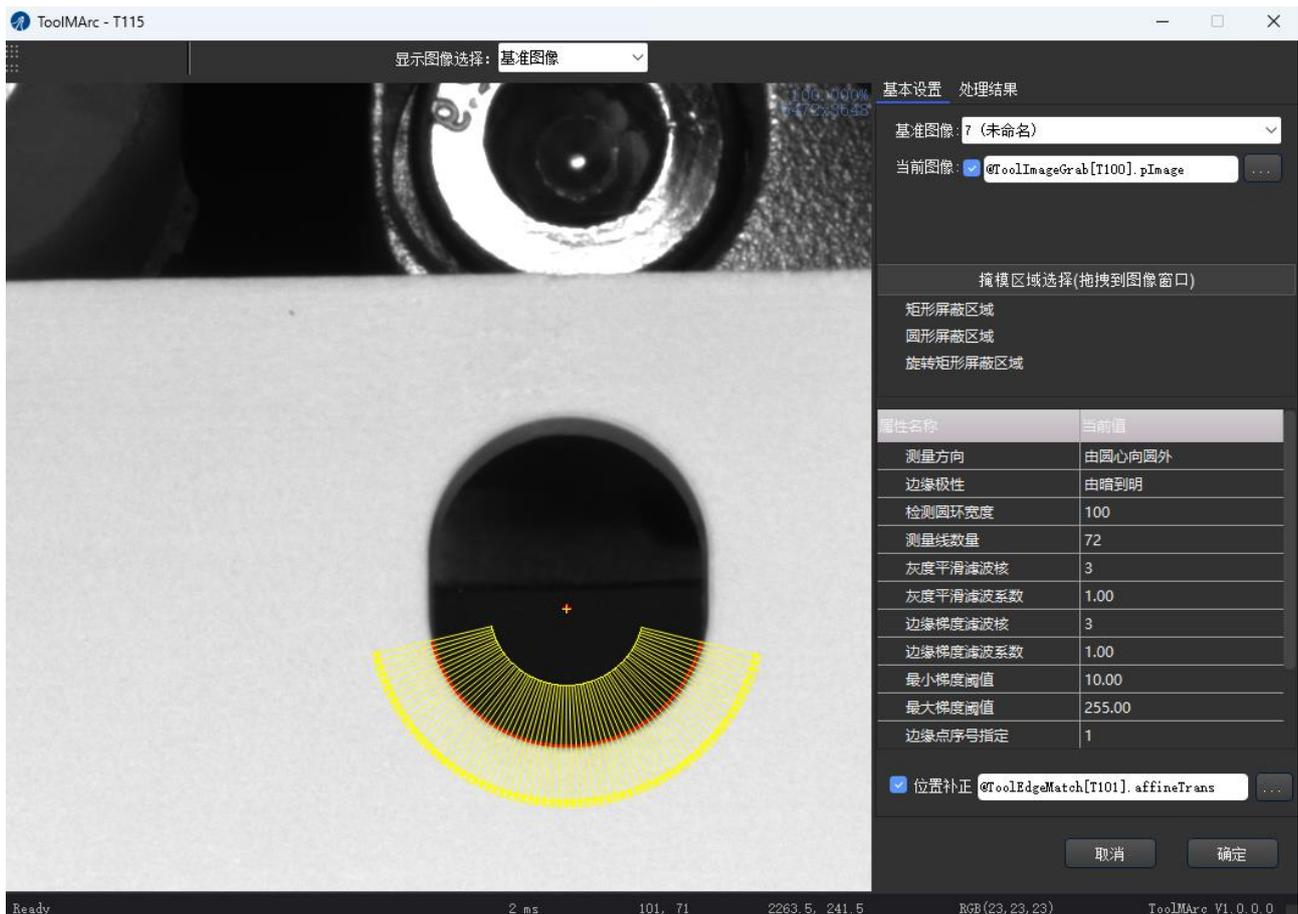
序号	边缘点x坐标	边缘点y坐标	梯度值
1	3085.95	3259.68	52.2741
2	3085.39	3278.82	43.9222
3	3081.12	3297.5	49.48
4	3074.92	3315.55	48.4206
5	3066.6	3332.68	45.7557
6	3055.31	3347.96	47.5212
7	3043.01	3362.23	46.3426
8	3029.3	3375.11	37.1225

- 阈值设置：当有效边缘点最小比例，圆最小最大半径、圆形最小最大 XY 坐标均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：圆环颜色、拟合的圆颜色、边缘点颜色及线宽设置。
- 序号：边缘点编号。
- 边缘点坐标：所有检测到的边缘点在当前图像的 XY 坐标位置。

- 梯度值：所有检测到的边缘点的边缘梯度阈值。

### 3.3.3 圆弧查找

从设定圆弧范围内检测符合条件的边界点，利用所有被检测的点拟合出一个圆，得到圆弧圆心坐标和半径，可用于圆弧定位和测量。



该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

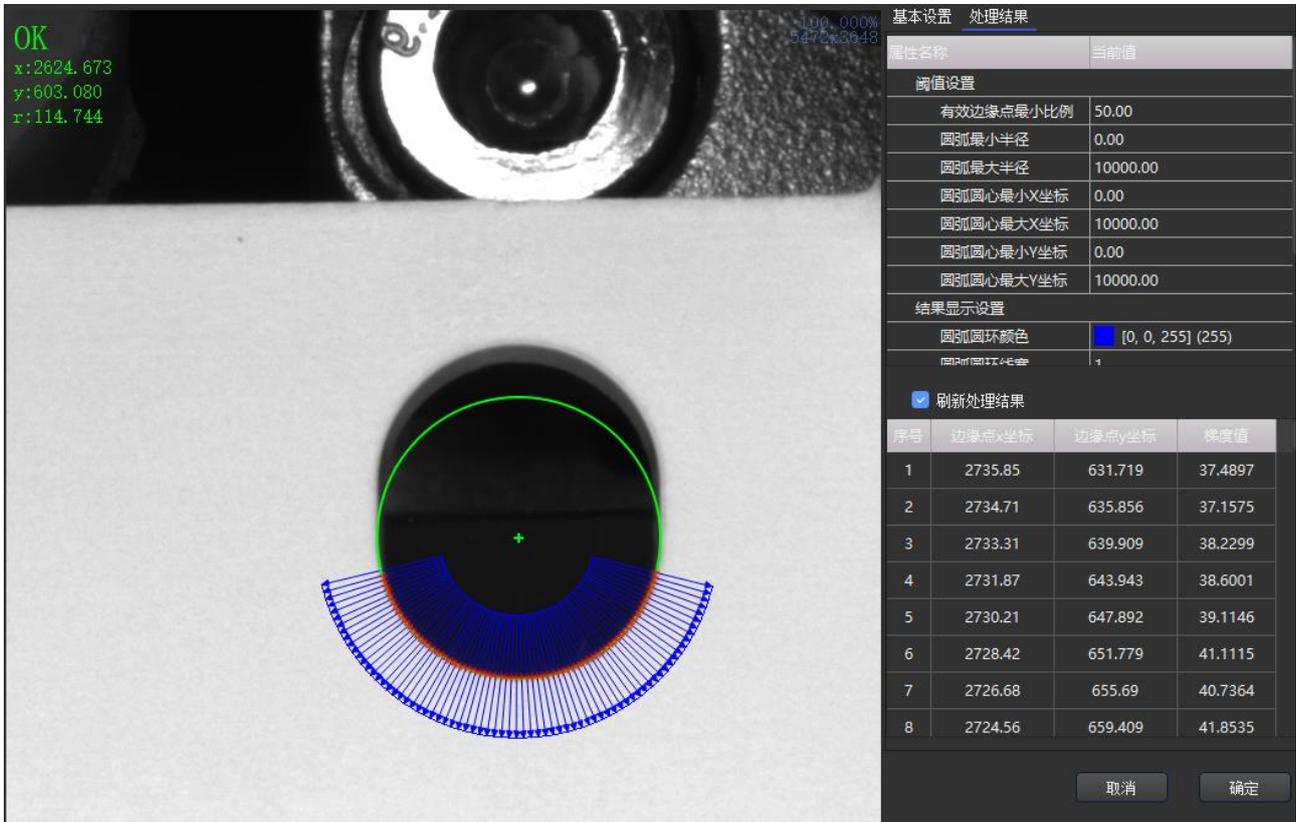
主要用于设置圆弧查找的测量参数。

- 基准图像：选择执行圆弧查找的基准图像，基准图像在菜单栏“图像”->“基准图像”中设置。选择基准图像后，会自动添加一个圆弧查找 ROI。
- 当前图像：算子运行时查找圆弧的图像，一般绑定图像采集、图像处理等工具的输出图像。

- 边缘极性：由明到暗、由暗到明和任意极性 3 种模式，由明到暗表示从亮度高过度到亮度低的边缘，由暗到明表示从亮度低过度到亮度高的边缘，任意极性则明到暗和暗到明的边缘都检测。
- 测量线数量：即卡尺的个数，决定找点的区域个数，每个卡尺里找到一个满足的点。
- 检测圆环宽度：卡尺的搜索长度，决定卡尺检测区域的纵向长度。
- 灰度平滑滤波核：垂直于卡尺测量线方向上的滤波核大小，像素单位。如滤波核设置为 3，则在该方向上使用了 3 个像素进行灰度均值计算。当边缘有噪音干扰时，可增大滤波核。
- 灰度平滑滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。
- 边缘梯度滤波核：卡尺测量线方向上的滤波核大小，用于增强边缘和抑制噪音干扰，最小值为 1，当边缘模糊不清晰或有噪音干扰时可以增大滤波核，这样可以使得检测结果更加稳定。
- 边缘梯度滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。
- 最小梯度阈值：设定检测边缘梯度阈值的下限，只有边缘梯度阈值大于该值的边缘点才被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除，取值范围 1~255。
- 最大梯度阈值：设定检测边缘梯度阈值的上限，只有边缘梯度阈值小于该值的边缘点才被检测到，取值范围 1~255。
- 边缘点序号指定：边缘点由多个卡尺卡出，该参数指定每个卡尺要找的边缘点序号，默认序号为 1，即第一条边，当 ROI 区域内存在多条直线时，可修改该序号调整要检测的直线。
- 边缘置信度：当边缘点序号设置为 0 时有效，边缘点梯度的权重占该卡尺区域梯度最大值的比重。
- 圆拟合方式：拟合方式有标准代数最小二乘、基于正交距离最小二乘和带权重的最小二乘三种。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定边缘模板匹配的仿射变换输出。

- 处理结果页

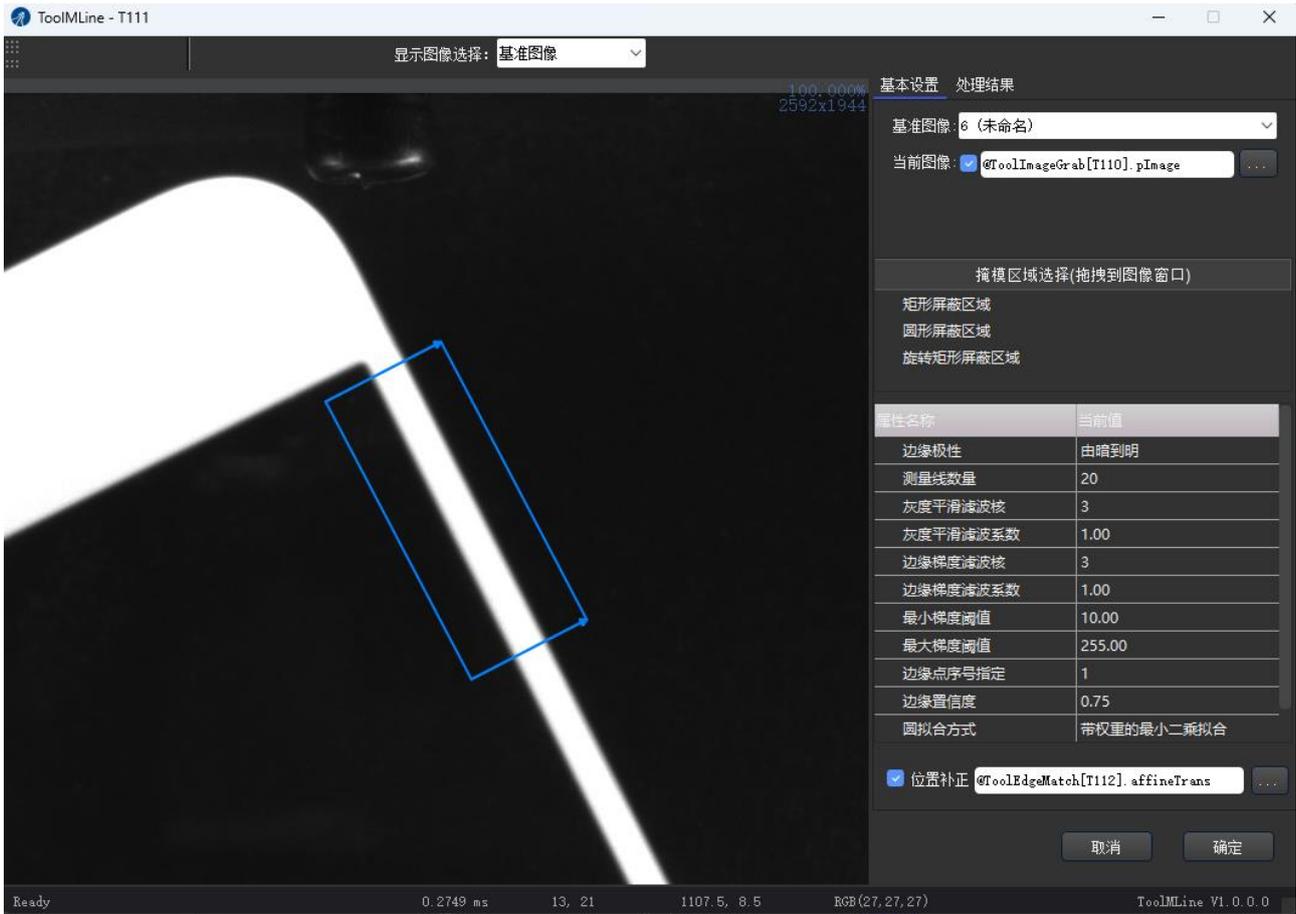
查看圆弧查找结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的结果、拟合圆弧的圆心坐标和半径。



- 阈值设置：当有效边缘点最小比例，圆弧最小最大半径、圆弧最小最大 XY 坐标均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：圆弧圆环颜色、拟合的圆弧颜色、边缘点颜色及线宽设置。
- 序号：边缘点编号。
- 边缘点坐标：所有检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：所有检测到的边缘点的边缘梯度阈值。

### 3.3.4 直线查找

从设定矩形范围内检测符合条件的边界点，利用所有被检测的点拟合出一条直线，得到矩形框中心点到该直线的垂足坐标和直线的方向向量，可用于直线定位和测量。



该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

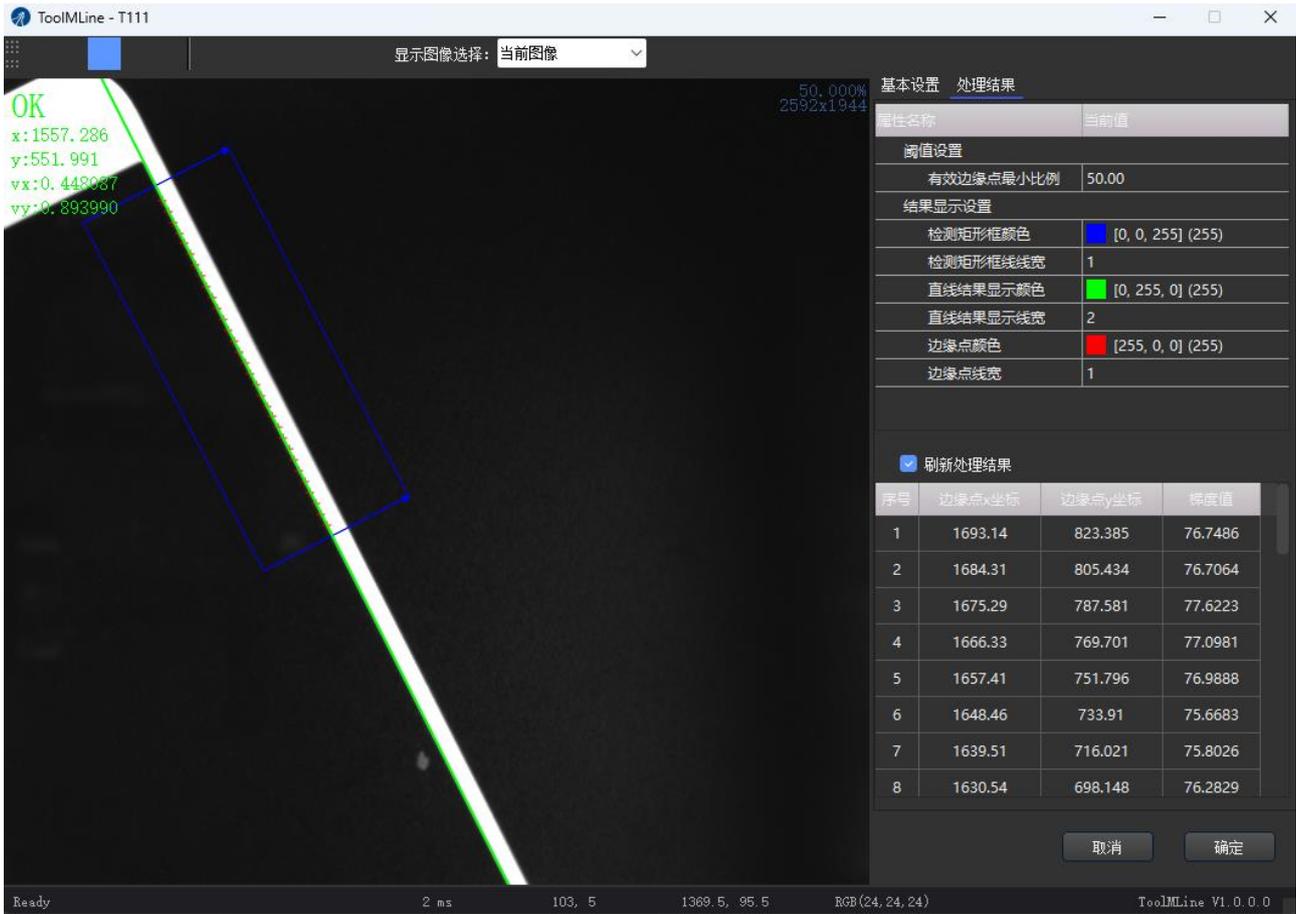
主要用于设置直线查找的测量参数。

- 基准图像：选择执行直线查找的基准图像，基准图像在菜单栏“图像”->“基准图像”中设置。选择基准图像后，会自动添加一个矩形检测 ROI。
- 当前图像：算子运行时查找直线的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 边缘极性：由明到暗、由暗到明和任意极性 3 种模式，由明到暗表示从亮度高过度到亮度低的边缘，由暗到明表示从亮度低过度到亮度高的边缘，任意极性则明到暗和暗到明的边缘都检测。
- 测量线数量：即卡尺的个数，决定找点的区域个数，每个卡尺里找到一个满足的点。
- 灰度平滑滤波核：垂直于卡尺测量线方向上的滤波核大小，像素单位。如滤波核设置为 3，则在该方向上使用了 3 个像素进行灰度均值计算。当边缘有噪音干扰时，可增大滤波核。
- 灰度平滑滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。

- 边缘梯度滤波核：卡尺测量线方向上的滤波核大小，用于增强边缘和抑制噪音干扰，最小值为 1，当边缘模糊不清晰或有噪音干扰时可以增大滤波核，这样可以使得检测结果更加稳定。
- 边缘梯度滤波系数：值越大，越接近均值滤波效果，对图像平滑效果越明显。
- 最小梯度阈值：设定检测边缘梯度阈值的下限，只有边缘梯度阈值大于该值的边缘点才被检测到。数值越大，抗噪能力越强，得到的边缘数量越少，甚至导致目标边缘点被筛除，取值范围 1~255。
- 最大梯度阈值：设定检测边缘梯度阈值的上限，只有边缘梯度阈值小于该值的边缘点才被检测到，取值范围 1~255。
- 边缘点序号指定：边缘点由多个卡尺卡出，该参数指定每个卡尺要找的边缘点序号，默认序号为 1，即第一条边，当 ROI 区域内存在多条直线时，可修改该序号调整要检测的直线。
- 边缘置信度：当边缘点序号设置为 0 时有效，边缘点梯度的权重占该卡尺区域梯度最大值的比重。
- 直线拟合方式：拟合方式有标准代数最小二乘和带权重的最小二乘两种。
- 位置补正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置补正，一般绑定边缘模板匹配的仿射变换输出。

- 处理结果页

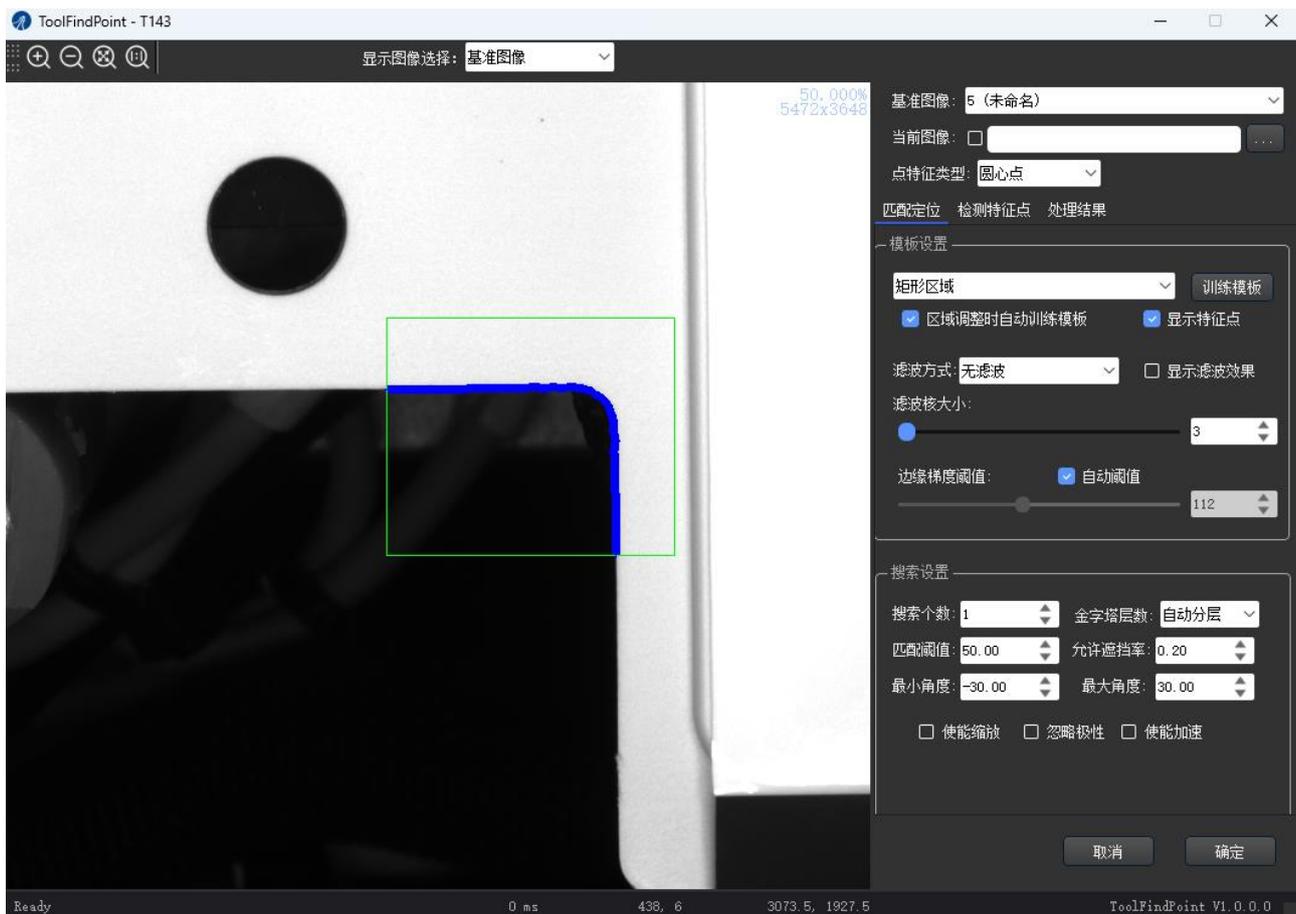
查看直线查找结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果、矩形框中心点到该直线的垂足坐标和该直线的方向向量。



- 阈值设置：当有效边缘点最小比例在设定的范围内时，工具状态结果为 True。
- 结果显示设置：检测矩形框颜色、拟合的直线颜色、边缘点颜色及线宽设置。
- 序号：边缘点编号。
- 边缘点坐标：所有检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：所有检测到的边缘点的边缘梯度阈值。

### 3.3.5 特征点定位

基于模板匹配查找图片中的圆心点或直角点并输出该特征点的坐标。



该算子工具共 3 个设置页，分别为：匹配定位、检测特征点、处理结果。

- 匹配定位页

主要用于创建模板及设置匹配参数，参数设置请参考[轮廓匹配](#)章节。

- 检测特征点页

设置检测的特征点类型、参数及 ROI 区域。

点特征类型：可选择圆心点和直角点。选择圆心点时，参数设置请参考[圆查找](#)章节；选择直角点时，参数设置请参考[直线查找](#)章节。

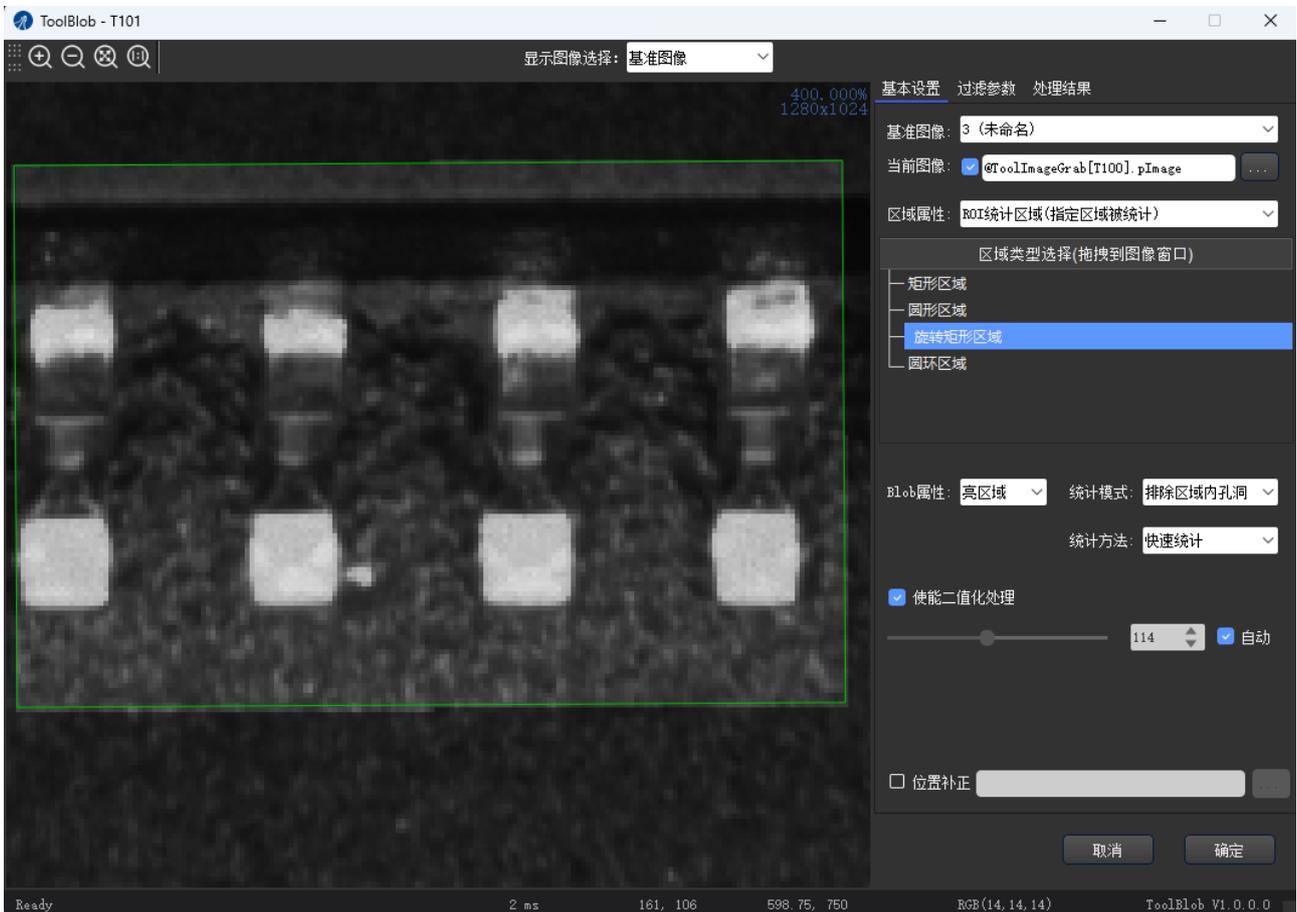
- 处理结果页

查看特征点结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的结果及特征点坐标。

特征点为圆心点时，参数说明请参考[圆查找](#)章节，特征点为直角点时，参数说明请参考[线交点](#)章节。

### 3.3.6 BLOB 分析

将图像指定区域二值化处理后统计分析亮区域或暗区域的信息，如面积、轮廓周长、重心坐标、外接矩形、最小外接圆等。一般用于对图像中的缺陷进行检测或定位。



该算子工具共 3 个设置页，分别为：基本设置、过滤参数和处理结果。

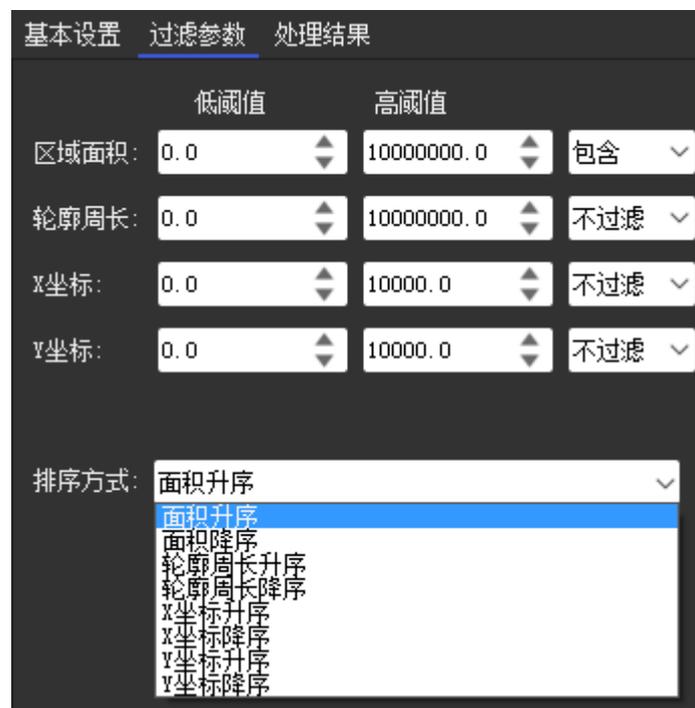
- 基本设置页

主要用于设置 BLOB 分析的 ROI 区域。

- 基准图像：选择执行 BLOB 分析的基准图像，基准图像在菜单栏“图像”->“基准图像”中设置。
- 当前图像：算子运行时进行 BLOB 分析的图像，一般绑定图像采集、图像处理等工具的输出图像。
- 区域属性：设定要检测的 Blob 区域为统计对象或掩膜对象。
- 区域类型选择：选择 ROI 形状（矩形、圆、旋转矩形、圆环）。
- Blob 属性：可以设置当前需要提取的目标为亮区域（白色）或暗区域（黑色）。

- 统计模式：计算区域面积时，是否计算区域内的孔洞，分为排除区域内孔洞和包含区域内孔洞。
  - 统计方法：快速统计和逐像素精确统计两种方式。快速统计方式速度快但有一定偏差，逐像素精确统计方式速度稍慢但精准。
  - 使能二值化处理：设置二值图像的分割阈值，当像素灰度值大于等于该值为白色（亮区域），否则为黑色（暗区域）。
  - 自动：根据直方图分布自动计算出最佳分割阈值。
  - 位置修正：设定检测 ROI 区域在当前图像状态下与基准图像状态时的矫正参数。该参数设定后检测 ROI 会随着图像的变化而变化。当被检测目标的位置发生变化的情况下需要对目标进行定位，此时需要将检测该目标的矩形 ROI 也跟随移动，这时就需要勾选位置修正，一般绑定边缘模板匹配的仿射变换输出。
- 过滤参数页

设置目标每种特征的最小值和最大值，当某个目标的特征不在该范围值内会被移除还是保留。



- 区域面积：设置目标面积过滤参数，不在低阈值~高阈值范围内的目标将被排除或保留，被排除的目标将不会再参与特征的计算。
- 轮廓周长：设置目标轮廓周长过滤范围。

- X 坐标：设置目标 X 坐标过滤范围。
- Y 坐标：设置目标 Y 坐标过滤范围。
- 排序方式：设置提取到的目标按面积、轮廓、坐标的升序或降序排列。
- 处理结果页

查看 BLOB 分析结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果。

序号	面积	X坐标	Y坐标	周长
1	2.5	665.733	800.533	7.41421
2	126.5	619.851	775.134	49.5563
3	156.5	571.973	774.915	57.2132
4	200.5	667.367	773.474	61.2132
5	203	715.465	772.035	57.3137
6	297	712.493	821.581	68.1421
7	306.5	570.593	822.66	70.3848
8	313.5	665.343	821.843	69.8995
9	313.5	618.174	822.117	84.8701

- 阈值设置：当 Blob 数量的上下限在设定的范围内时，工具状态结果为 True。
- 结果显示设置：轮廓颜色、重心十字标颜色、最小外接矩形框颜色及最小外接矩形圆颜色的设置。
- 序号：目标编号。
- XY 坐标：重心坐标，即目标对象所有像素坐标的平均值。
- 周长：目标对象的周长。
- 最小外接矩形：目标对象的最小外接矩形。
- 最小外接圆：目标对象的最小外接圆。

### 3.4 测量

### 3.4.1 线线距离

测量两条直线的距离。两条直线一般不会绝对平行，所以线线测量距离按照直线 1 的重心与直线 2 的重心生成的线段的中点，该中点到直线 1 的距离与到直线 2 的距离的总和。



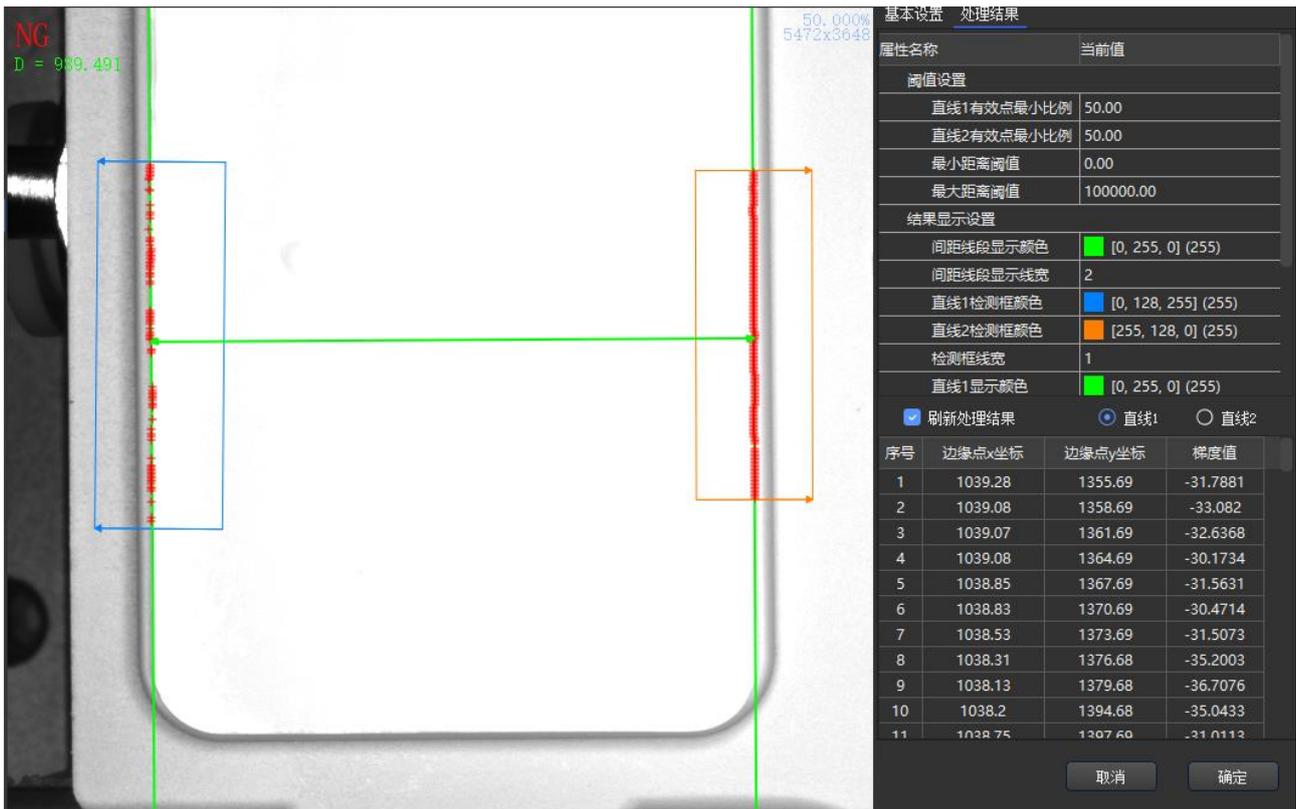
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置两条直线检测的测量参数。参数设置请参考[直线查找](#)章节。

- 处理结果页

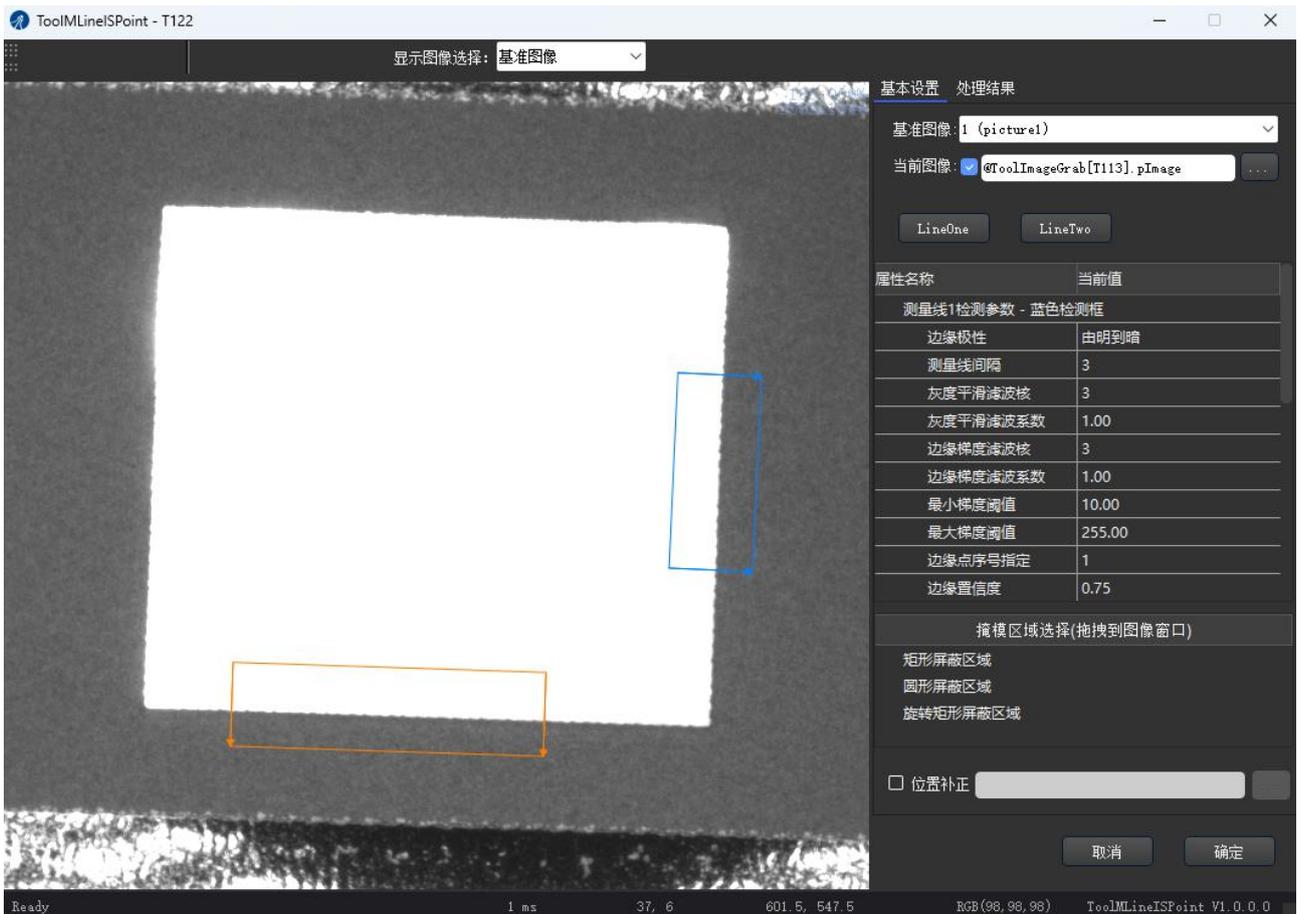
查看线线测量结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和两直线的距离。



- 阈值设置：当两直线的有效点最小比例、最大最小距离阈值均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：两直线的间距线段颜色、两直线的检测框颜色、拟合的两条直线颜色、两直线边缘点颜色及线宽设置。
- 序号：每条直线的边缘点编号。
- 边缘点坐标：每条直线检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：每条直线检测到的边缘点的边缘梯度阈值。

### 3.4.2 线线交点

计算两条直线的交点和夹角。



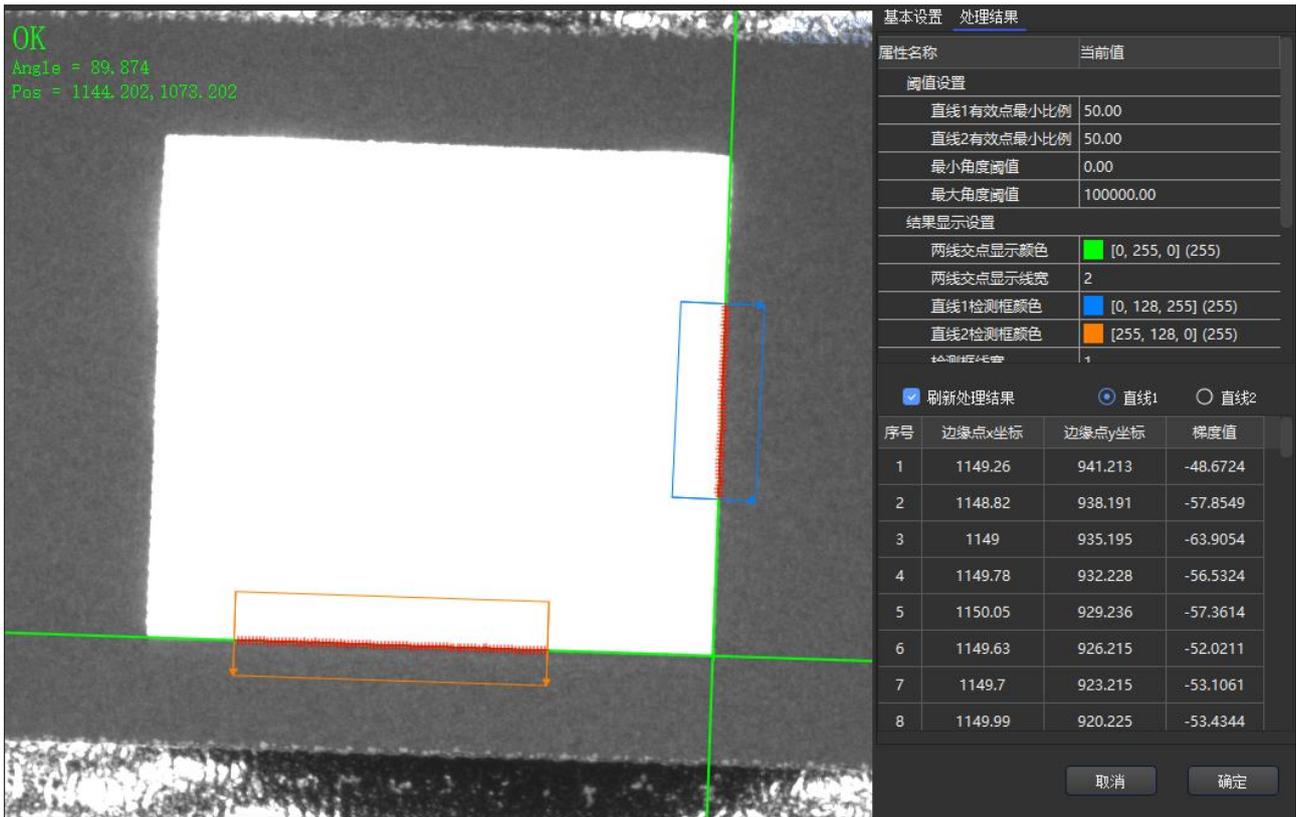
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置两条直线检测的测量参数。参数设置请参考[直线查找](#)章节。

- 处理结果页

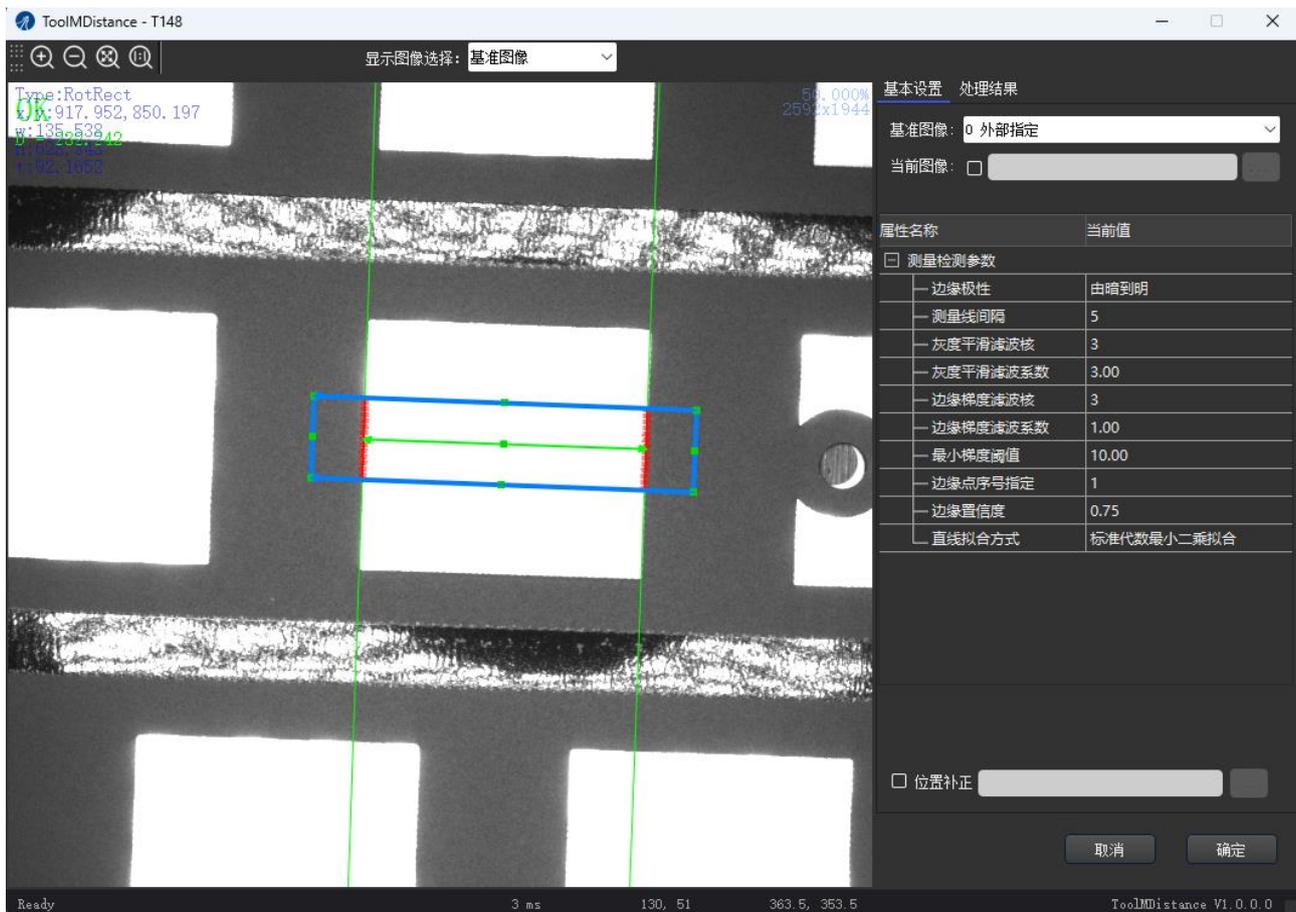
查看线线交点结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和两直线的角度、交点。



- 阈值设置：当两直线的有效点最小比例、最大最小距离阈值均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：两直线的交点颜色、两直线的检测框颜色、拟合的两条直线颜色、两直线边缘点颜色及线宽设置。
- 序号：每条直线的边缘点编号。
- 边缘点坐标：每条直线检测到的边缘点在当前图像的 XY 坐标位置。
- 梯度值：每条直线检测到的边缘点的边缘梯度阈值。

### 3.4.3 间距测量

检测指定区域内的边缘对间距，可用于测量长度。



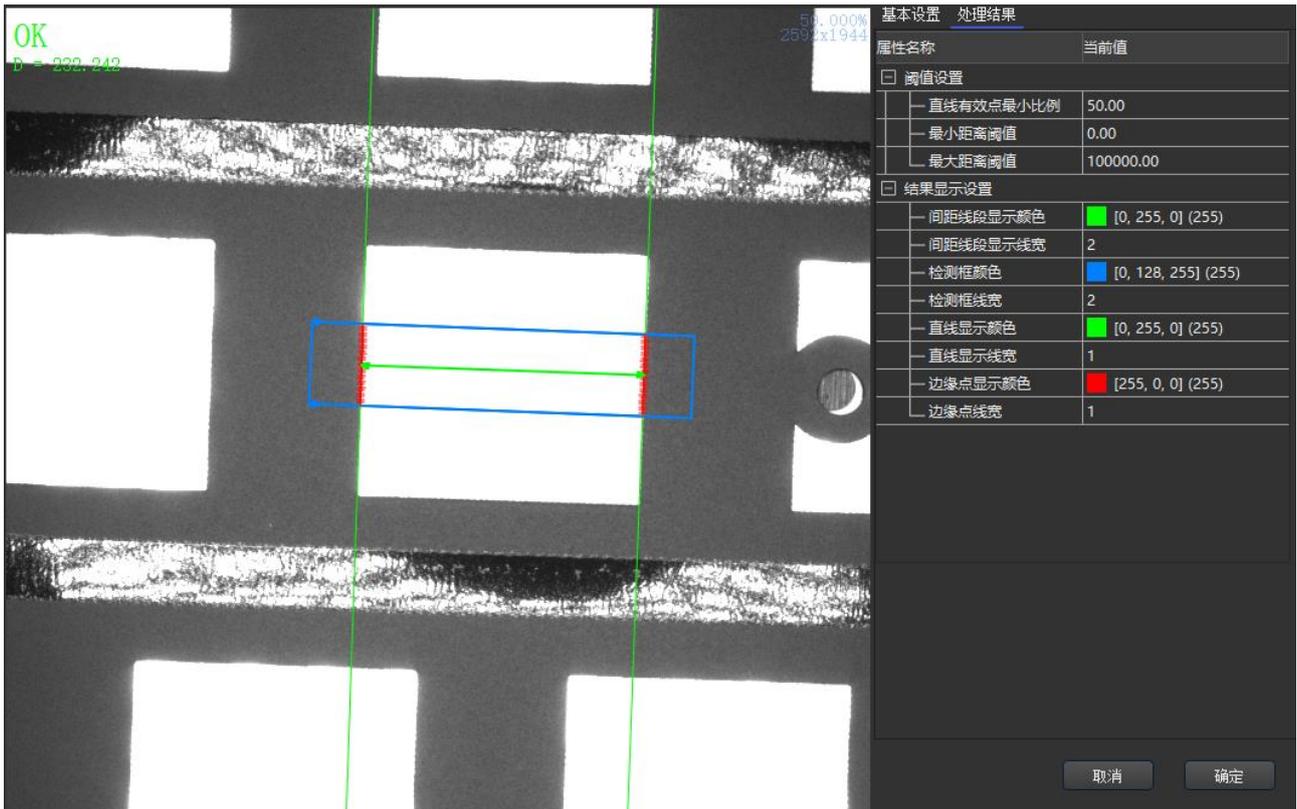
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置间距检测的测量参数。参数设置请参考[直线查找](#)章节。

- 处理结果页

查看间距值结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和间距值。



- 阈值设置：当有效边缘点最小比例、最大最小距离阈值均在设定的范围内时，工具状态结果为 True。
- 结果显示设置：两直线的间距线段颜色、检测矩形框颜色、拟合的两条直线颜色、两直线边缘点颜色及线宽设置。

### 3.4.4 线线计算

接收两条直线参数，根据接收参数求得直线与直线的交点和夹角并输出，可切换到工具结果界面查看输出信息。与“结果显示”工具搭配，可显示输出的交点及角度信息。



- 整体指定：关联直线的来源。
- X 坐标：设置过直线的一个点的 X 坐标值，可关联。
- Y 坐标：设置过直线的一个点的 Y 坐标值，可关联。
- 向量  $v_x$ ：设置直线单位向量 x 方向分量，可关联。
- 向量  $v_y$ ：设置直线单位向量 y 方向分量，可关联。

### 3.4.5 线圆计算

接收一条直线和一个圆参数，根据接收参数求得直线与圆的位置关系并输出，可切换到工具结果界面查看输出信息。与“结果显示”工具搭配，可显示输出的交点坐标、圆心到直线的距离、圆心到直线的垂线及垂足点坐标等信息。



- 整体指定：关联直线或圆的来源。
- X 坐标：设置圆心或过直线的一个点的 X 坐标值，可关联。
- Y 坐标：设置圆心或过直线的一个点的 Y 坐标值，可关联。
- 向量 vx：设置直线单位向量 x 方向分量，可关联。
- 向量 vy：设置直线单位向量 y 方向分量，可关联。
- R 半径：设置圆的半径，可关联。

### 3.4.6 圆圆计算

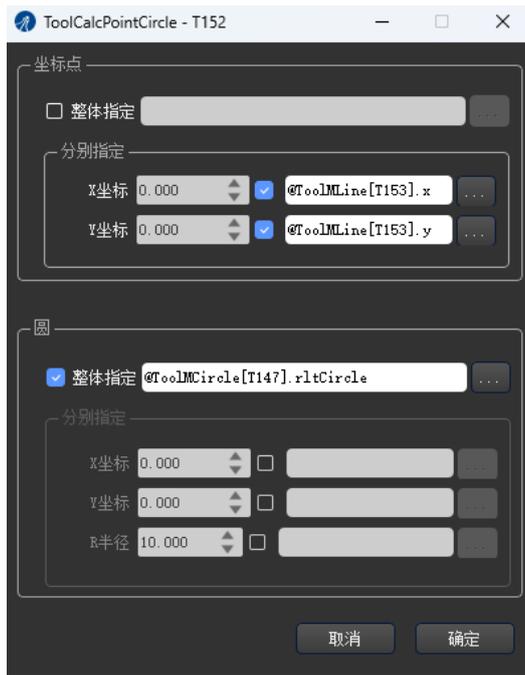
接收两个圆参数，根据接收参数求得圆与圆的位置关系并输出，可切换到工具结果界面查看输出信息。与“结果显示”工具搭配，可显示输出的两圆交点坐标、两圆圆心连线等信息。



- 整体指定：关联圆的来源。
- X 坐标：设置圆心 X 坐标值，可关联。
- Y 坐标：设置圆心 Y 坐标值，可关联。
- R 半径：设置圆的半径，可关联。

### 3.4.7 点圆计算

接收一个点和一个圆参数，根据接收参数求得点到圆的距离并输出，可切换到工具结果界面查看输出信息（点到圆心的距离，点到圆的最近距离，点到圆的最远距离）。与“结果显示”工具搭配，可显示输出的距离线段。



- 整体指定：关联点或圆的来源。
- X 坐标：设置点或圆心 X 坐标值，可关联。
- Y 坐标：设置点或圆心 Y 坐标值，可关联。
- R 半径：设置圆的半径，可关联。

### 3.4.8 点线计算

接收一个点和一条直线参数，根据接收参数求得点到直线的垂线并输出，可切换到工具结果界面查看输出信息（点到直线的垂足点、垂线、垂线角度、垂线距离）。与“结果显示”工具搭配，可显示输出的垂线信息。



- 整体指定：关联点或直线的来源。
- X 坐标：设置点或过直线的一个点的 X 坐标值，可关联。
- Y 坐标：设置点或过直线的一个点的 Y 坐标值，可关联。
- 向量 vx：设置直线单位向量 x 方向分量，可关联。
- 向量 vy：设置直线单位向量 y 方向分量，可关联。

### 3.4.9 点点计算

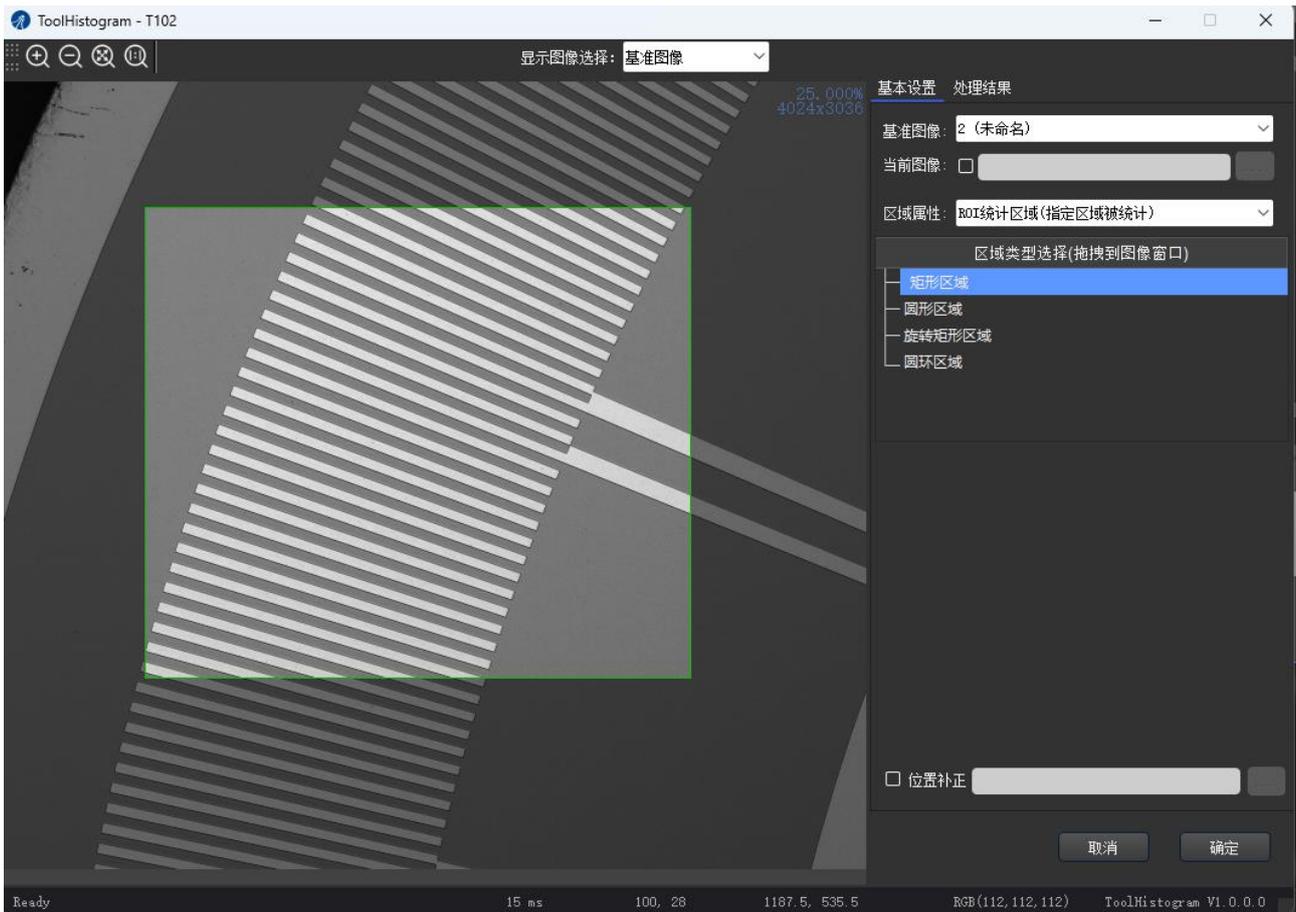
接收两个点参数，根据接收参数求得点到点的距离等信息并输出，可切换到工具结果界面查看输出信息（点到点的距离、两点生成的线段的角度、线段的中点坐标、两点生成的直线）。与“结果显示”工具搭配，可显示输出的线段信息。



- 整体指定：关联点的来源。
- X坐标：设置点的 X 坐标值，可关联。
- Y坐标：设置点的 Y 坐标值，可关联。

### 3.4.10 直方图统计

检测图像上指定一个目标区域，统计目标区域中的像素个数、灰度平均值、中值、标准差。还会生成灰度直方图，可以清晰看到各个灰度值下的像素点分布状态。



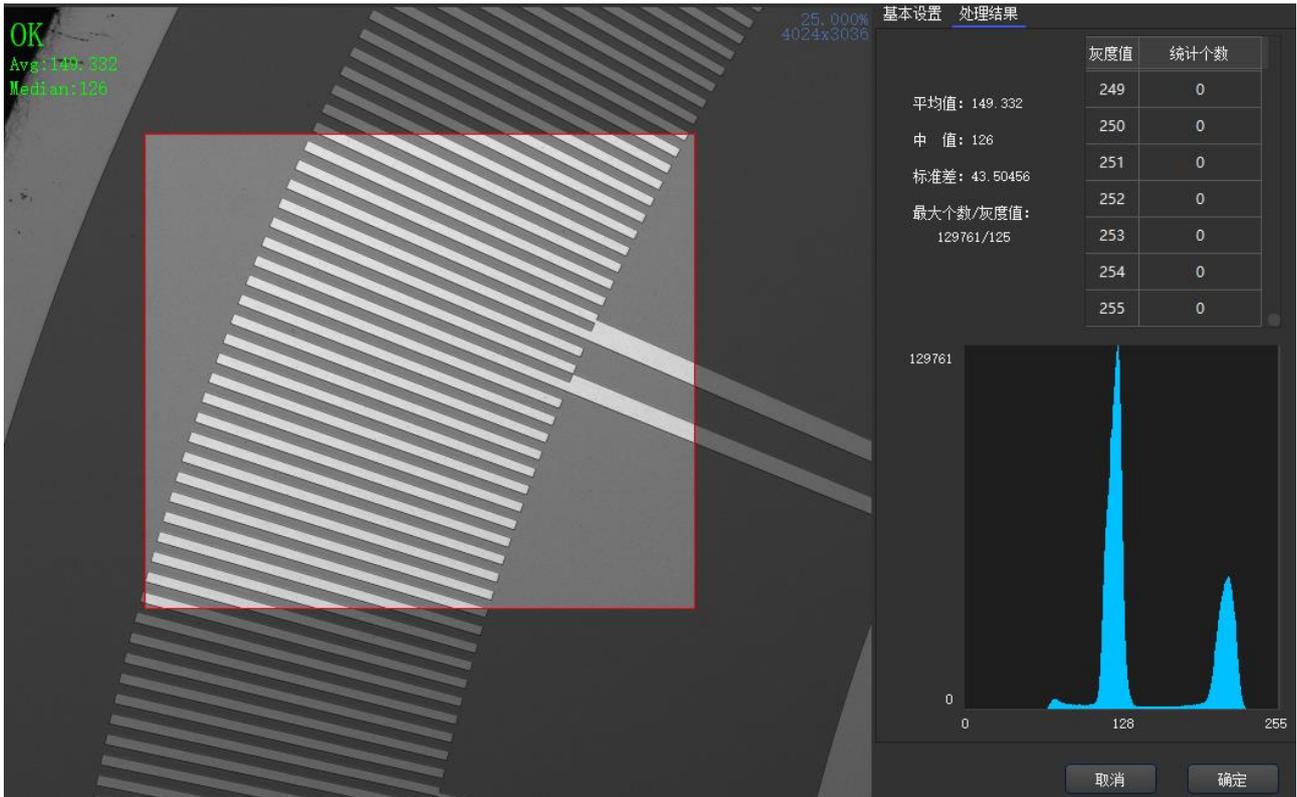
该算子工具共 2 个设置页，分别为：基本设置和处理结果。

- 基本设置页

主要用于设置统计直方图的 ROI 区域。参数设置请参考 [BLOB 分析](#) 章节。

- 处理结果页

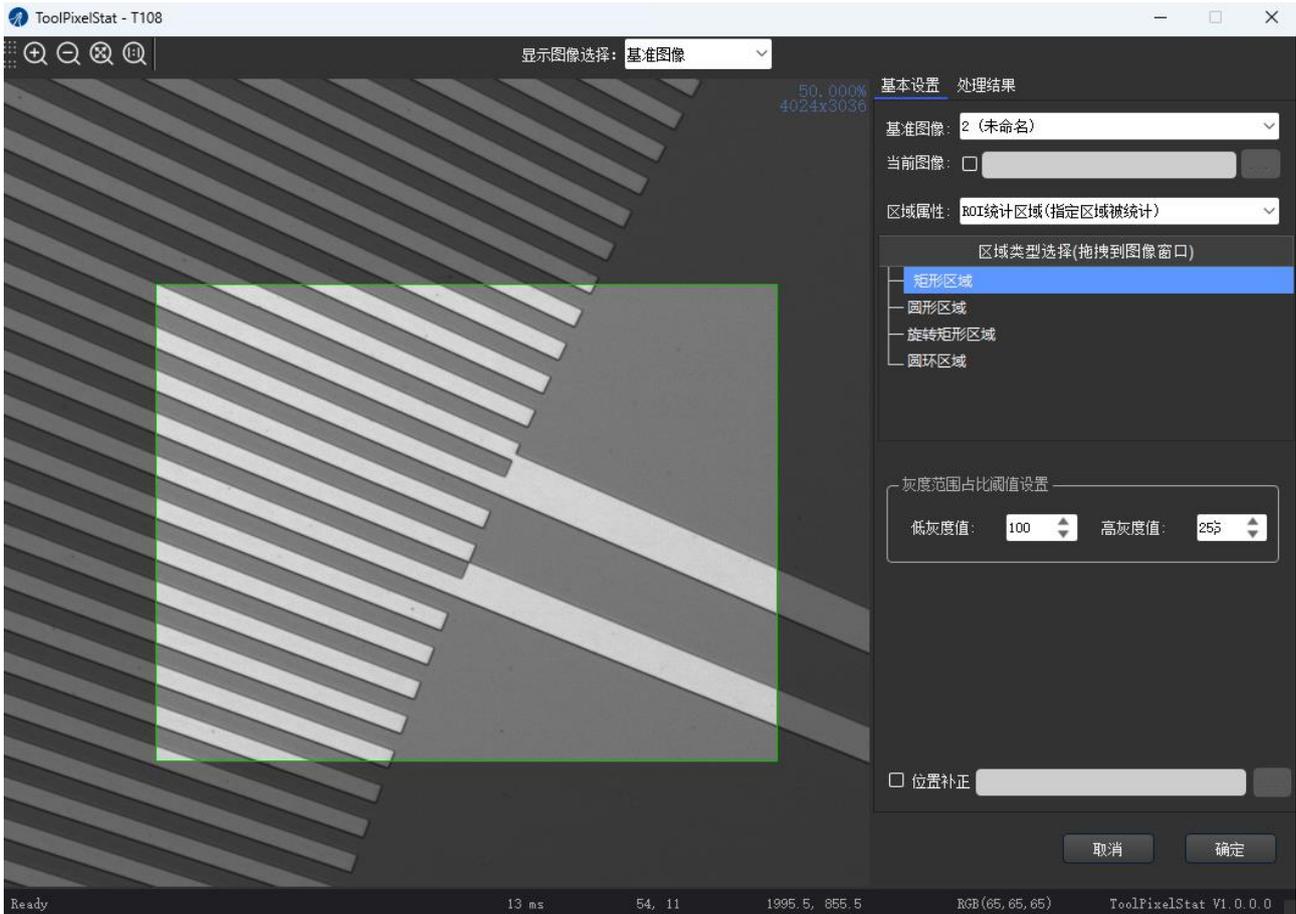
查看统计结果。其中图形显示窗左上角输出该工具的状态结果和灰度中值、平均值。



- 数据：显示灰度平均值、中值、标准差数据、峰值灰度。
- 直方图：显示图像 ROI 区域内的灰度直方图。

### 3.4.11 像素统计

检测图像上指定一个目标区域，统计目标区域内满足高低阈值的像素个数和比率。



该算子工具共 2 个设置页，分别为：基本设置和处理结果。

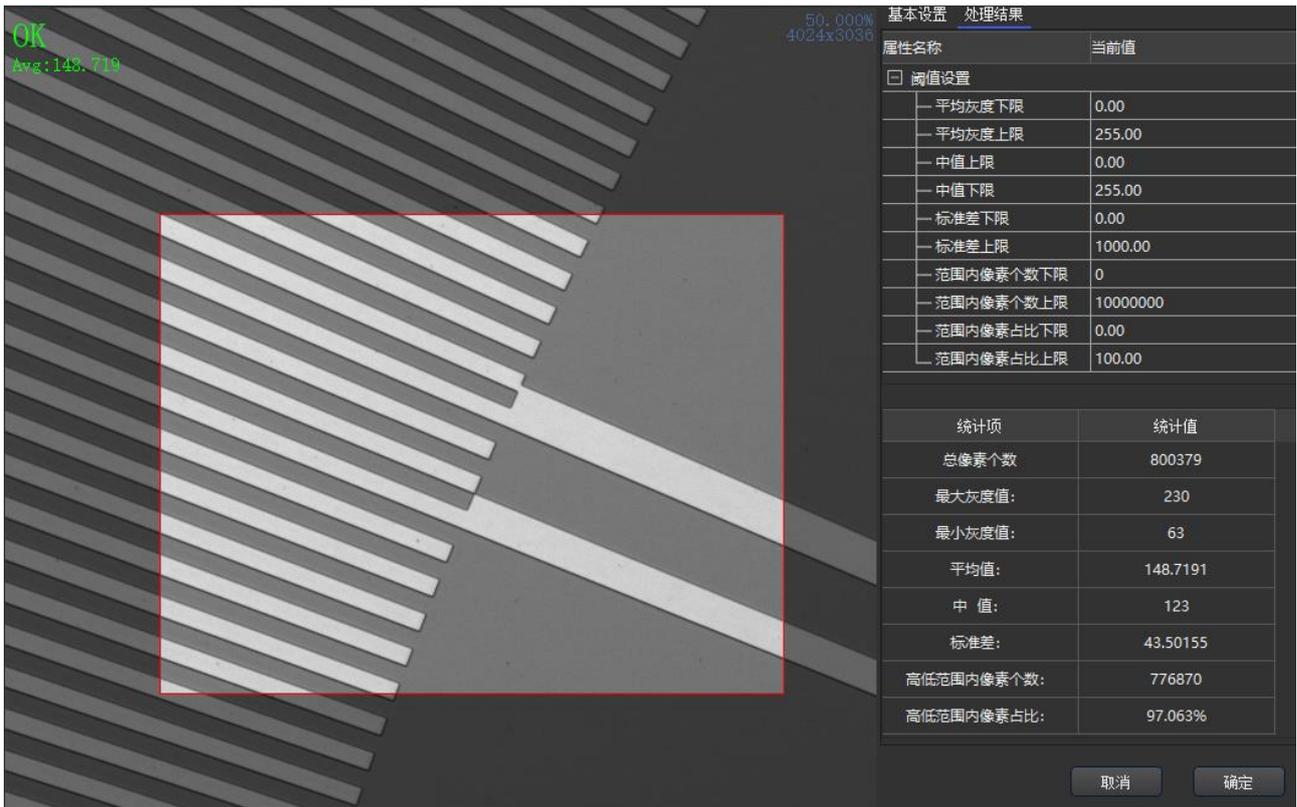
- 基本设置页

主要用于设置像素统计的 ROI 区域。

- 灰度范围占比阈值设置：设置目标区域内灰度阈值范围，用于统计该阈值范围内的像素个数和占比。
- 其他参数设置请参考 [BLOB 分析](#) 章节。

- 处理结果页

查看像素统计结果及设置结果阈值判定范围。其中图形显示窗左上角输出该工具的状态结果和灰度平均值。

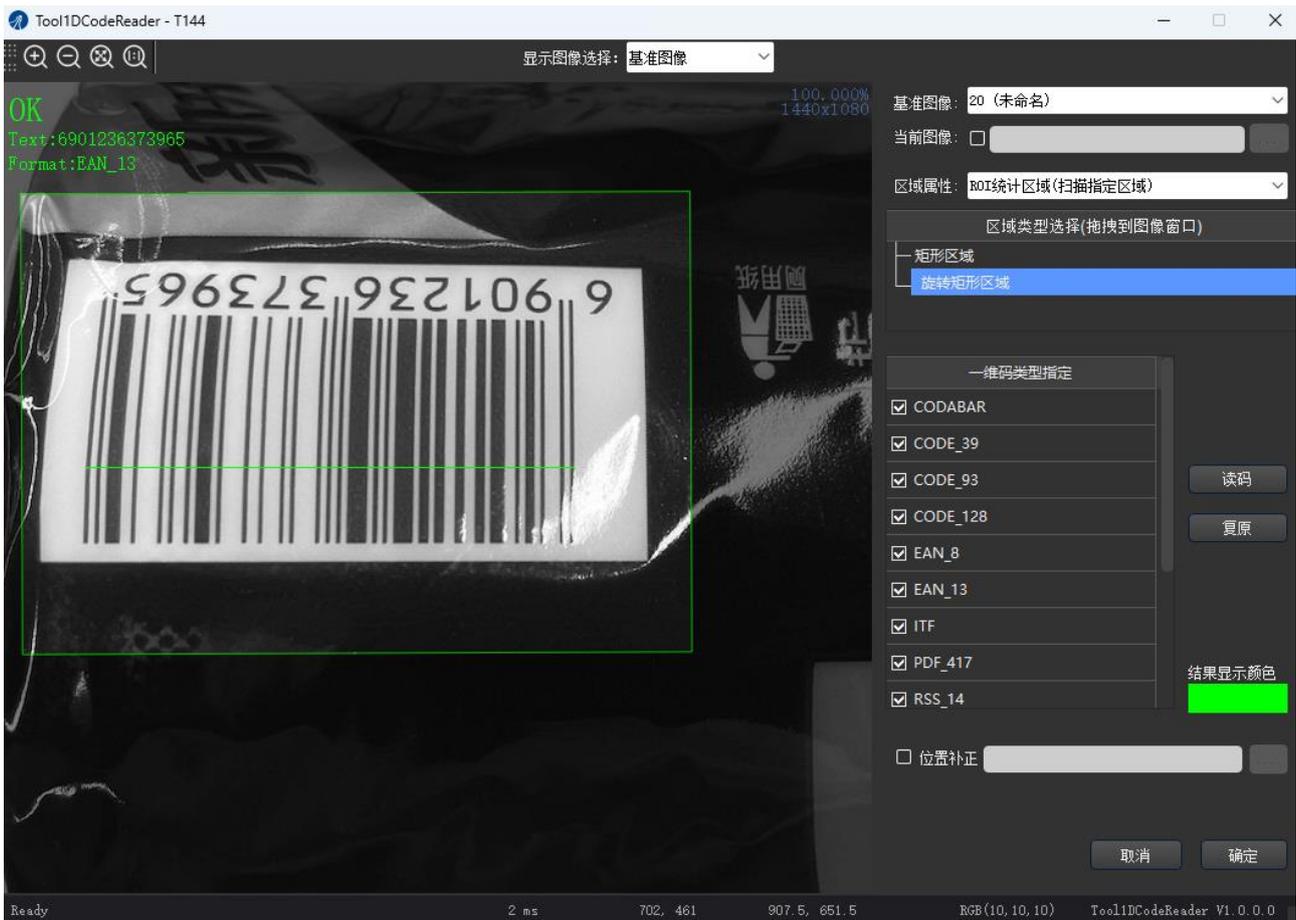


- 阈值设置：当灰度平均值、中值、标准差、灰度阈值范围内的像素个数及占比在设定的上下限范围内时，工具状态结果为 True。
- 总像素个数：目标区域内的像素总个数。
- 最大灰度值：目标区域内的灰度最大值。
- 最小灰度值：目标区域内的灰度最小值。
- 平均值：目标区域内的灰度平均值。
- 中值：目标区域内的灰度中值。
- 标准差：目标区域内的灰度标准差。标准差是方差的算术平方根，能反映一个数据集的离散程度。
- 高低范围内像素个数：设置的灰度阈值范围内的像素总个数。
- 高低范围内像素占比：设置的灰度阈值范围内的像素个数与总像素个数的比值。

### 3.5 识别

### 3.5.1 一维码

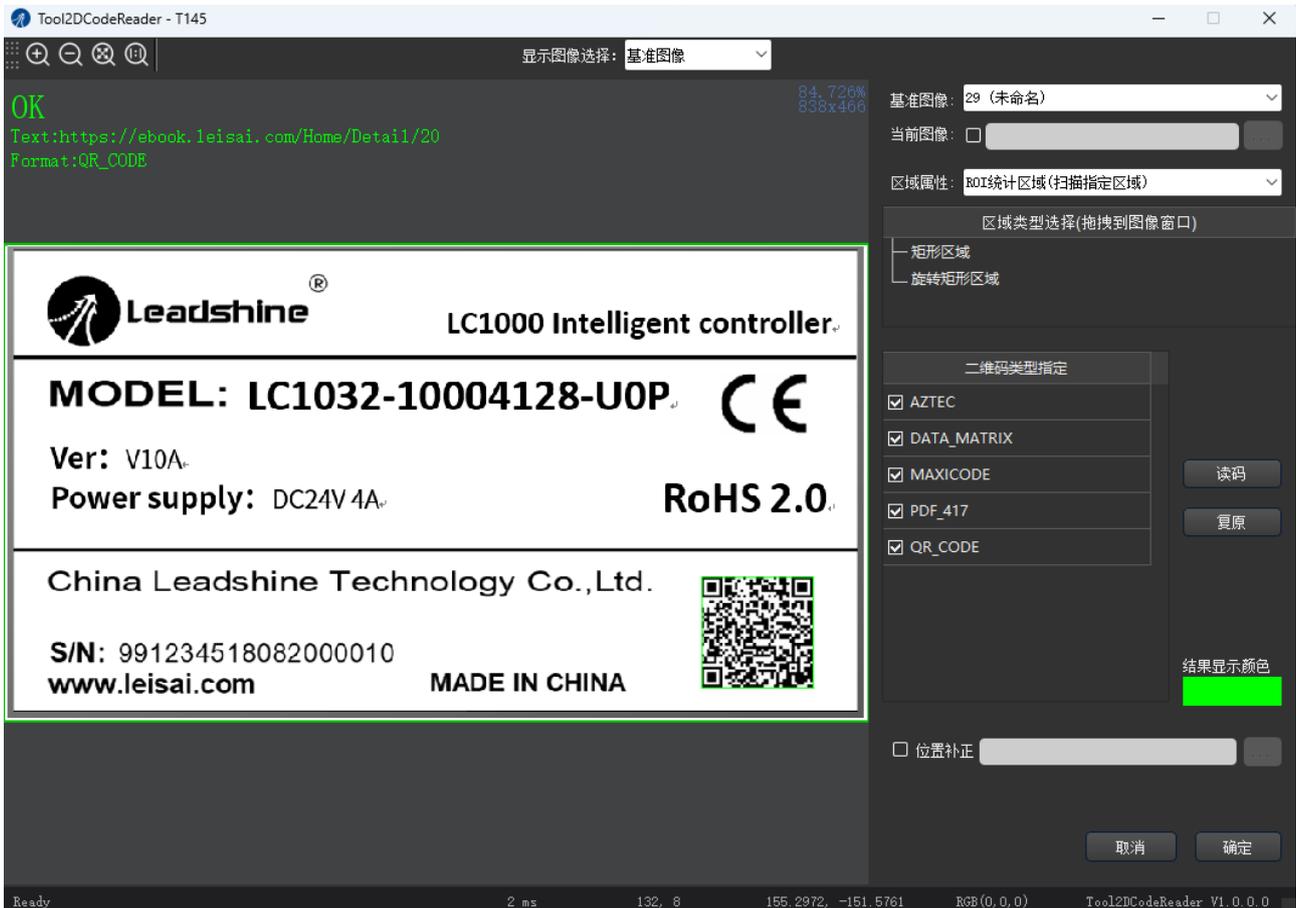
用于识别指定区域内的一维码并以字符的形式输出，容忍目标一维码具有一定量的角度旋转及倾斜，目前一次只能识别一个码，支持解码类型包括 CODABAR、CODE39、CODE93、CODE128、EAN-8、EAN-13、ITF、PDF\_417、RSS\_14、RSS\_EXPANDED、UPC\_A、UPC\_E、UPC\_EAN\_EXTENSION。



- 一维码类型指定：勾选需要识别的码类型。
- 读码：执行一次读码操作。
- 复原：恢复到读码前的状态。
- 结果显示颜色：设置读码结果显示的文本颜色。

## 3.5.2 二维码

用于识别指定区域内的二维码并以字符的形式输出，容忍目标二维码具有一定的角度旋转及倾斜，目前一次只能识别一个码，支持解码类型包括 QR\_CODE、DATA\_MATRIX、MAXICODE、AZTEC、PDF\_417。

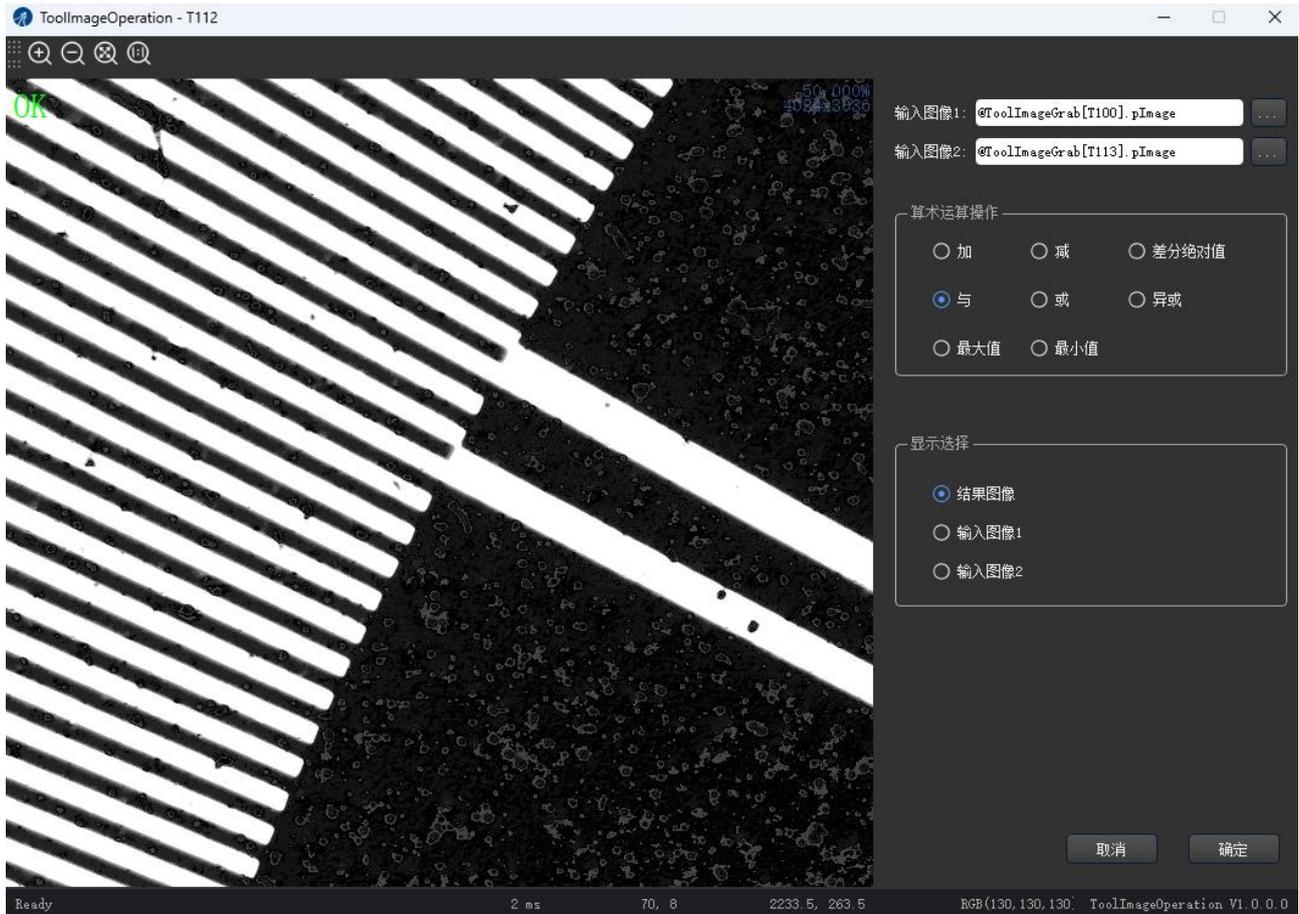


参数设置请参考一维码。

## 3.6 图像处理

### 3.6.1 图像运算

对两张图像做加、减、与、或等操作，并输出一张新的图像。



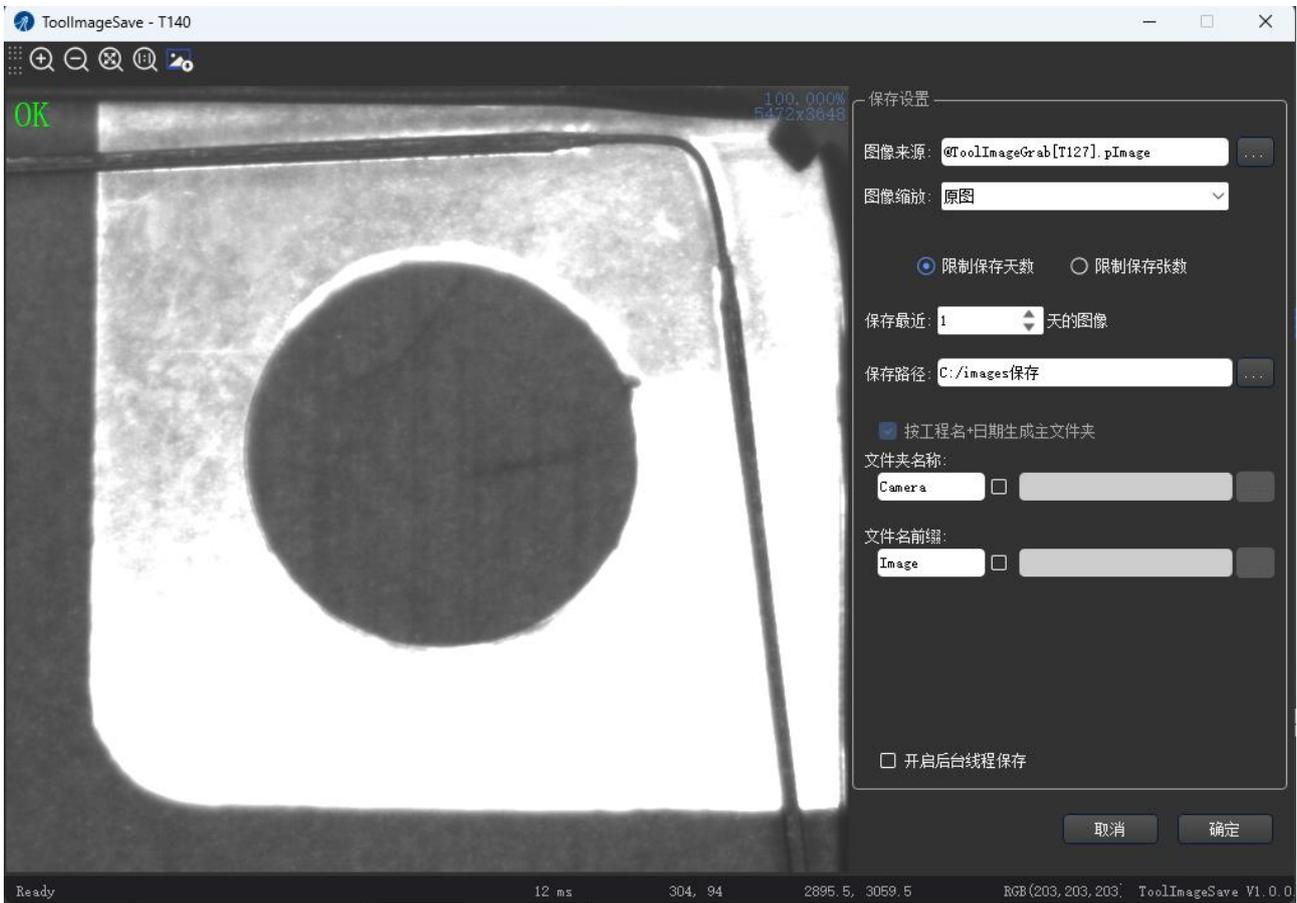
- 输入图像 1：关联第一张图像来源。
- 输入图像 2：关联第二张图像来源。
- 算术运算操作：可以选择加、减、差分绝对值、与、或、异或、最大值和最小值。

项目	概述
加	两张图像做加法运算，如果得出的值大于 255，则会被置为 255
减	两张图像做减法运算，如果得出的值小于 0，则会被置为 0
差分绝对值	两张图像做减法运算，如果得出的值小于 0，则会求绝对
与	两张图像做与运算，按二进制的每一位做与操作
或	两张图像做或运算，按二进制的每一位做或操作
异或	两张图像做异或运算，按二进制的每一位做异或操作
最大值	求两张图像的最大值，将每一个像素进行比较，输出值大的像素
最小值	求两张图像的最小值，将每一个像素进行比较，输出值小的像素

- 显示选择：设置当前需要显示的图像。

## 3.6.2 图像保存

流程运行一次保存一张图像到本地硬盘。

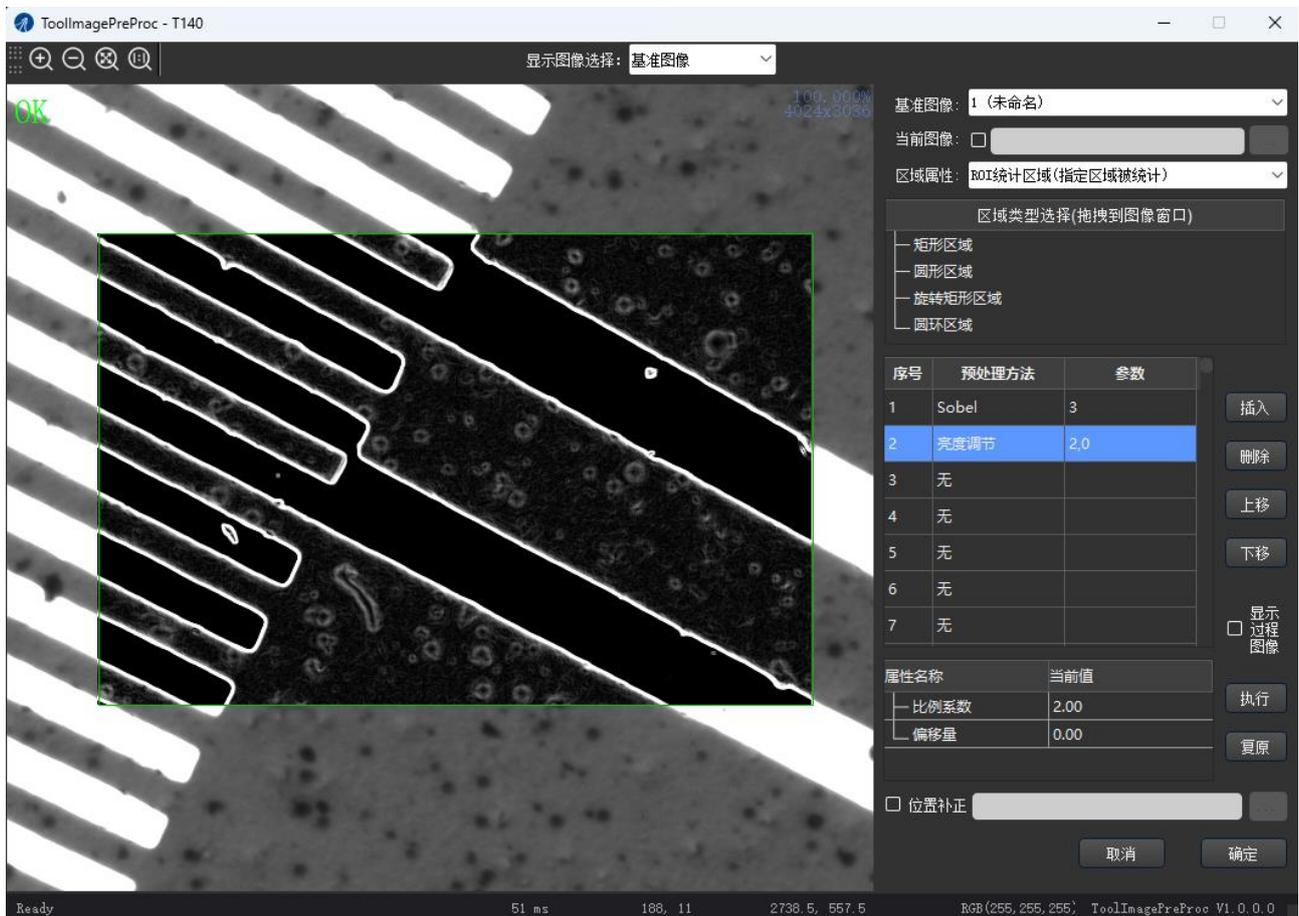


- 图像来源：关联当前需要保存的图像来源。
- 图像缩放：可将图像缩小后再保存。
- 限制保存天数：按运行天数进行保存，最大可保存最近 180 天的图像。
- 限制保存张数：按图像数量进行保存，最大可保存 5000 张图像。
- 保存路径：设置保存图像的目录。
- 按工程名+日期生成主文件夹：选择“限制保存天数”时勾选显示，根据当前工程名称和日期生成主文件夹名称（如 myProject\_2024\_12\_02）。
- 按工程名生成主文件夹：选择“限制保存天数”时勾选显示，根据当前工程名称生成主文件夹名称（如 myProject）。
- 文件夹名称：设置主文件夹下的子文件夹名称，可关联。
- 文件名前缀：设置保存图像的名称，可关联，会自动在名称后增加日期和时间后缀（如 Image\_12\_02\_09\_10\_32\_203.bmp）。

- 开启后台线程保存：设置单独开启后台线程保存图像。

### 3.6.3 图像预处理

对输入图像做预处理并输出处理后的图像。



- 插入：在两个预处理方法中间插入一个预处理方法。
- 删除：删除选择的预处理方法。
- 上移/下移：将选中的方法上下移动。
- 显示过程图像：勾选后点击“执行”，可以查看当前选中的预处理方法执行后的结果。
- 执行：运行所有设置的预处理方法，输出最后的图像。
- 复原：恢复到源图像状态。
- 其他参数设置请参考 [BLOB 分析](#) 章节。

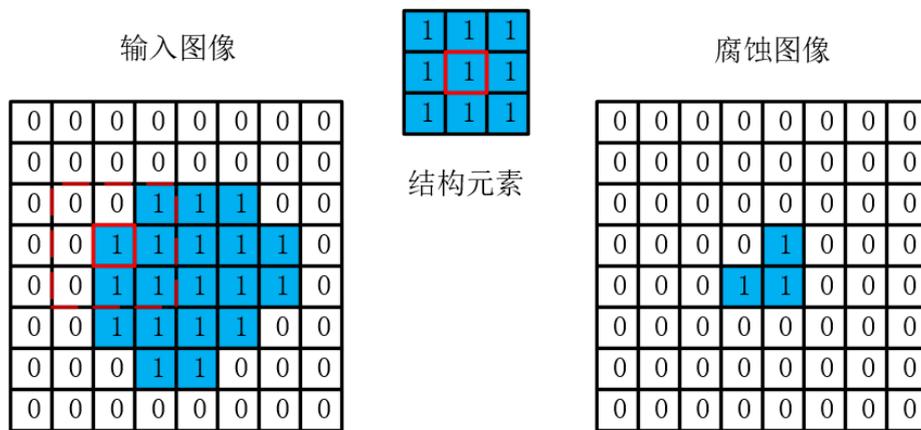
预处理方法包括以下几种：

- 二值化

二值化是将图像上的像素点的灰度值设置为 0 或 255，高于某个阈值全是 255，低于某个阈值全是 0，这样整个图像将呈现出明显的黑白效果。

- 形态学

形态学处理可以根据设定的结构元素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、梯度、顶帽、黑帽；支持的结构元素包括：矩形、十字、椭圆，它不是一个像素，而是一个几何形状的像素块。这个结构元素从左至右、从上到下遍历整个矩阵，每当在目标图像中找到一个与结构元素完全相同的子图像时，就把该子图像中与结构元素的中心位置对应的那个像素位置标记为 1，不一致则按 0 处理，这样图像中的高亮区域就会减少，这就是图像的腐蚀，如下图所示的示意图。



而膨胀与腐蚀正好相反，每当在目标图像中找到与结构元素有部分相同的子图像时，就把该子图像中与结构元素的中心位置对应的那个像素位置标记为 1，否则按 0 处理，这样图像中的高亮区域就会增加。

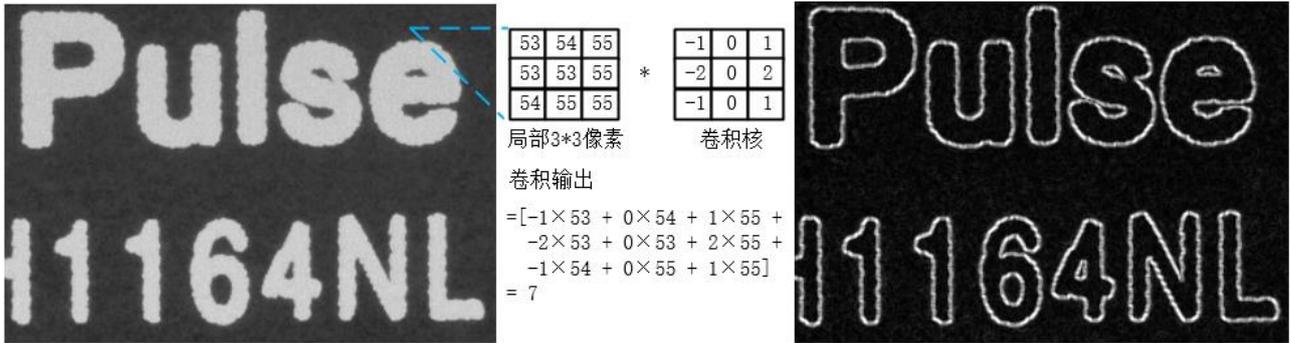
腐蚀和膨胀还可以组合起来对图像进行操作，比如开运算是先对输入图像执行先腐蚀后膨胀操作，而闭运算是先对图像执行先膨胀后腐蚀操作。腐蚀可对图像进行局部缩小，膨胀可对图像进行局部放大，因此开运算可有效清除二值图像中小的白色噪声像素块凸显高亮区域，闭运算则能填充二值图像中小的黑色像素凸显灰暗区域。

- 反色

对目标图像灰度取反。

- 滤波

图像滤波主要是为了图像平滑和去噪，是图像预处理很重要的一环。具体处理方法是使用一个小型滤波器（如 3x3），输出中心点的像素值由周围 8 个决定。



如果输出像素依赖的是输入像素的线性组合，则称为线性滤波器；如果输入输出是非线性的，则称为非线性滤波器。通过滤波器的方法对图像进行的操作称之为卷积，而滤波器则称为卷积核。

图像滤波可细分为均值滤波(卷积核所有像素对中心像素的贡献相等)，高斯滤波(邻域像素根据到中心像素距离不同有着不同的权重)，中值滤波(基于排序统计理论的一种能有效抑制噪声的非线性滤波)和边缘保持滤波(一种非线性、边缘保留、有效去噪声的滤波方法)。它们的区别如下：

- 均值滤波可对图像进行快速、简单的平滑处理，对于噪声的处理很差。
- 高斯滤波是一种比均值滤波稍微好一点的线性滤波，也不会改变原图像的边缘走向。
- 中值滤波处理速度快，对椒盐噪声处理效果很好，但对高斯噪声处理不好。
- 边缘保持滤波计算复杂，效率低，对高斯噪声处理效果很好，但对椒盐噪声处理不佳。

高斯滤波、中值滤波、均值滤波、边缘保持滤波的设置参数如下：

- **x 方向滤波核**：中值/高斯/均值滤波核的宽度，数值越大，图像越模糊。
- **y 方向滤波核**：中值/高斯/均值滤波核的高度，数值越大，图像越模糊。
- **sigma**：高斯滤波核的标准差，数值越大，图像越模糊。
- **滤波直径**：边缘保持滤波核直径，数值越大，图像越模糊。
- **颜色滤波 sigma**：边缘保持滤波颜色核的标准差，数值越大，越大范围内的颜色被聚合。
- **空间滤波 sigma**：边缘保持滤波空间核的标准差，数值越大，越大距离范围内的像素参与聚合。
- **Sobel**

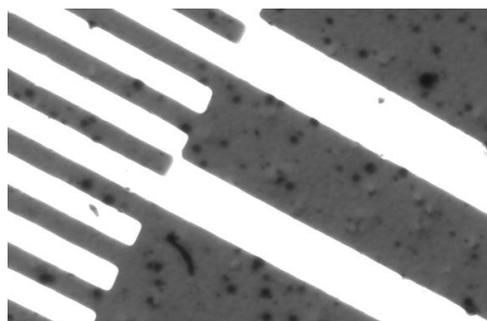
卷积核除了可以用作滤波，还能对图像进行梯度操作。比如图像从左向右看，像素值不断增大；或者从上到下看，像素值不断增大，这些都能反映出图像的梯度变化。Sobel 算子是一

种很经典的图像梯度提取算子，其本质是基于图像空间域卷积，背后的思想是图像一阶导数算子的理论支持。该算子主要用于获得数字图像的一阶梯度，这样可以把图像中的轮廓信息凸显出来。该算子包含两组 3x3 的矩阵，分别为水平方向及垂直方向，将之与图像作平面卷积，即可分别得出两个方向的亮度差分近似值。如下式卷积计算，其中 I 代表原始图像，Gx 及 Gy 分别代表经水平方向和垂直方向边缘检测的图像。

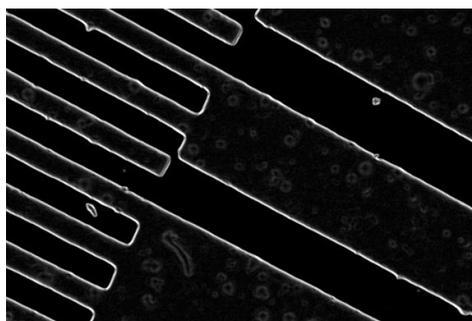
$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I$$

结合上述两个结果，在图像的每个像素点，按如下公式计算该点的梯度近似值，作为该像素点的输出值，其运算结果是一副边缘图像。

$$G = \sqrt{G_x^2 + G_y^2}$$



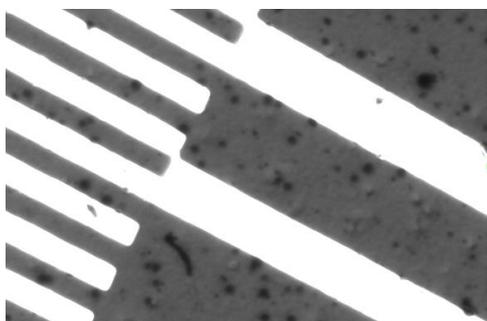
输入图像



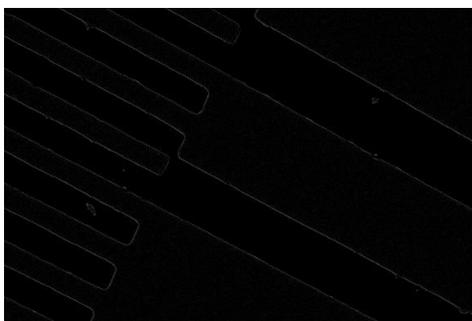
Sobel 图像

- 拉普拉斯

Sobel 算子属于一阶导数，获取到的边缘特征是极值，而拉普拉斯算子，是一种二阶导数滤波，获取到的边缘特征是过零点，能更好地提取边缘信息。



输入图像



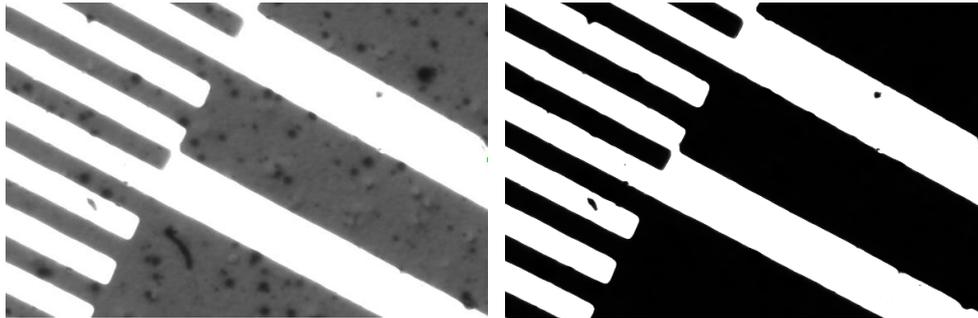
拉普拉斯图像

- 伽马变换

伽马变换是对输入图像灰度值进行非线性操作，使输出图像灰度值与输入图像灰度值呈指数关系，如下公式，这个指数就是 Gamma。

$$O = \left( \frac{I}{255} \right)^\gamma \times 255$$

伽马变换可以实现图像增强，当 gamma 值小于 1 时，图像整体亮度得到提升，同时低灰度处的对比度得到增加，更利于分辨低灰度值时的图像细节；当 gamma 值大于 1 时，则相反。



输入图像

gamma=5 图像

- 亮度调节

亮度调节可整体调节图像亮度和对比度，亮度和对比度调整公式如下，参数  $\alpha > 0$  和  $\beta$  通常称为比例系数和偏移量， $I$  为输入图像像素值， $O$  为输出图像像素值。

$$O = \alpha I + \beta$$

## 3.7 颜色处理

### 3.7.1 彩色转灰

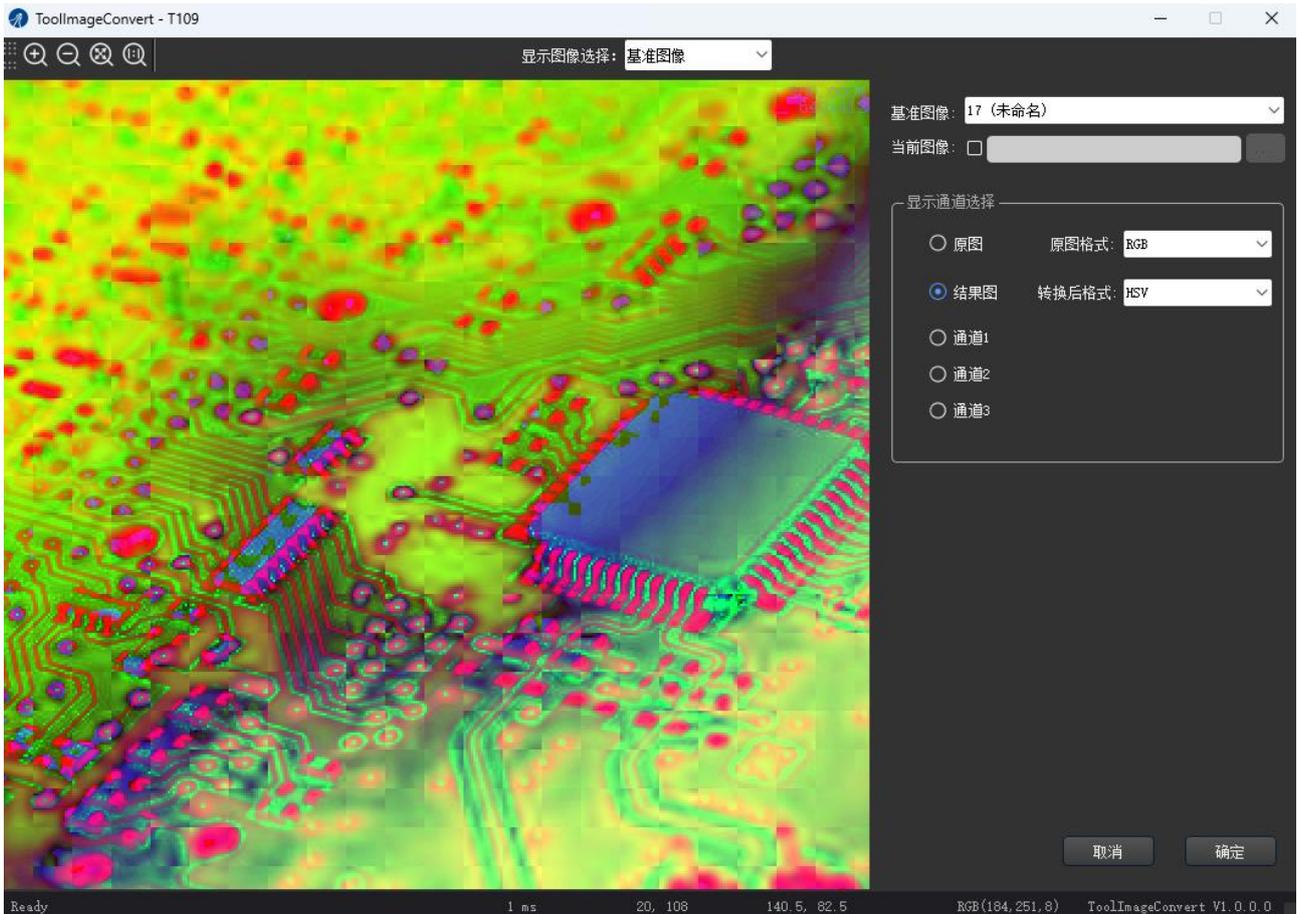
输入一幅 24 位彩色图像，转换为 8 位灰度图像输出。



- 基准图像：选择彩色转灰的基准图像。
- 当前图像：算子运行时执行处理的当前图像，一般绑定图像采集、图像处理等工具的输出图像。
- 原图格式：可选择 RGB、BGR 三通道彩色图像。
- 显示选择：设置当前需要显示的图像，可在当前画面中预览原图和转换后的灰度图像。

### 3.7.2 颜色空间转换

输入一幅 24 位彩色图像，对其进行颜色空间转换，用户可选择转 RGB、BGR、HSV、YUV、LAB 空间，并输出转换后图像指定颜色通道的灰度图像。



- 基准图像：选择彩色空间转换的基准图像。
- 当前图像：算子运行时执行处理的当前图像，一般绑定图像采集、图像处理等工具的输出图像。
- 原图格式：可选择 RGB、BGR 三通道彩色图像。
- 转换后格式：可选择转 RGB、BGR、HSV、YUV、LAB 空间。
- 显示通道选择：设置当前需要显示的图像，可在当前画面中预览原图和转换后的彩色图像或单通道灰度图像。

### 3.7.3 图像通道分离

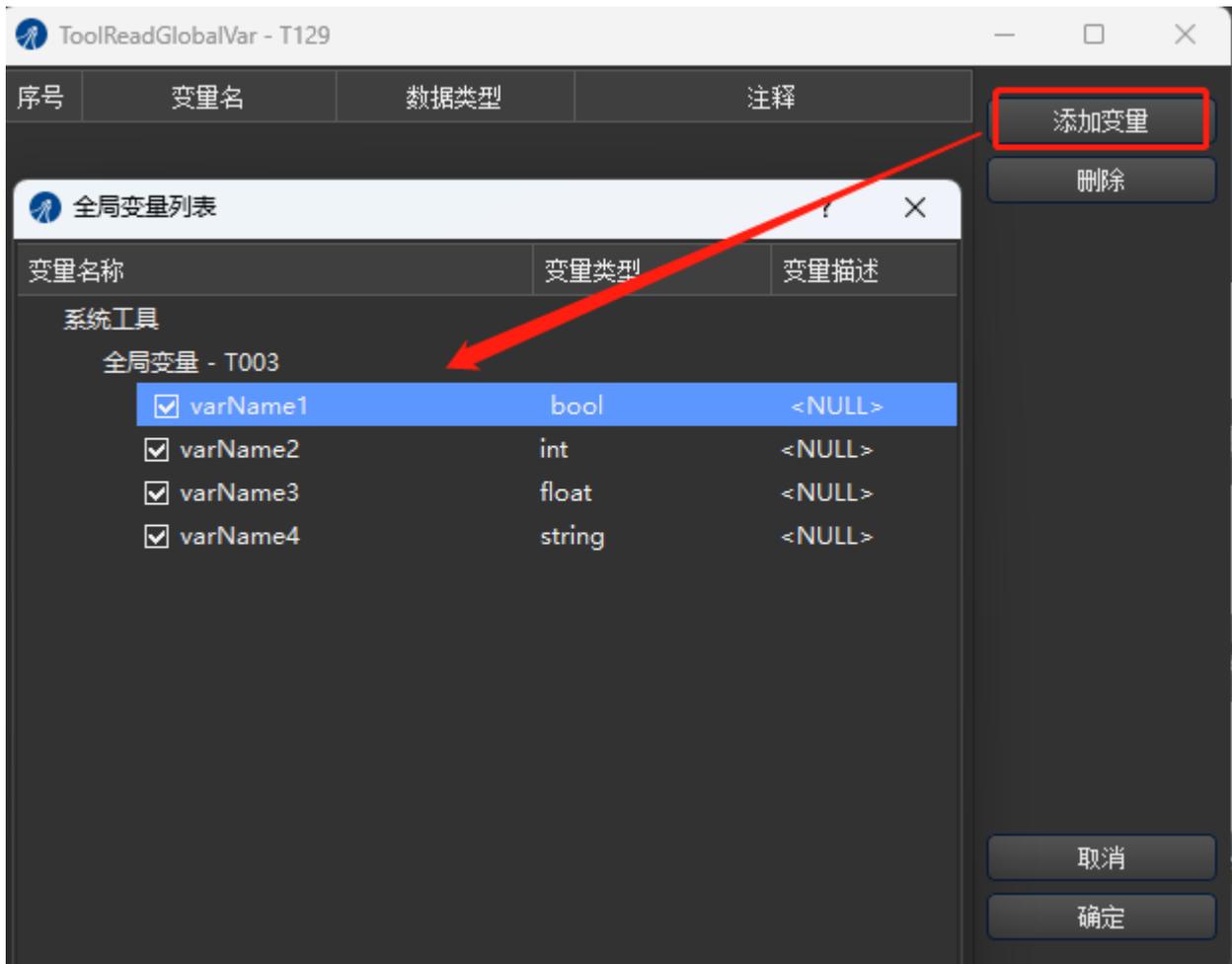
对一幅 24 位彩色图像，按 RGB 或 BGR 通道分离，输出每个通道的 8 位灰度图像。参数设置请参考[颜色空间转换](#)章节。



## 3.8 系统工具

### 3.8.1 读取全局变量

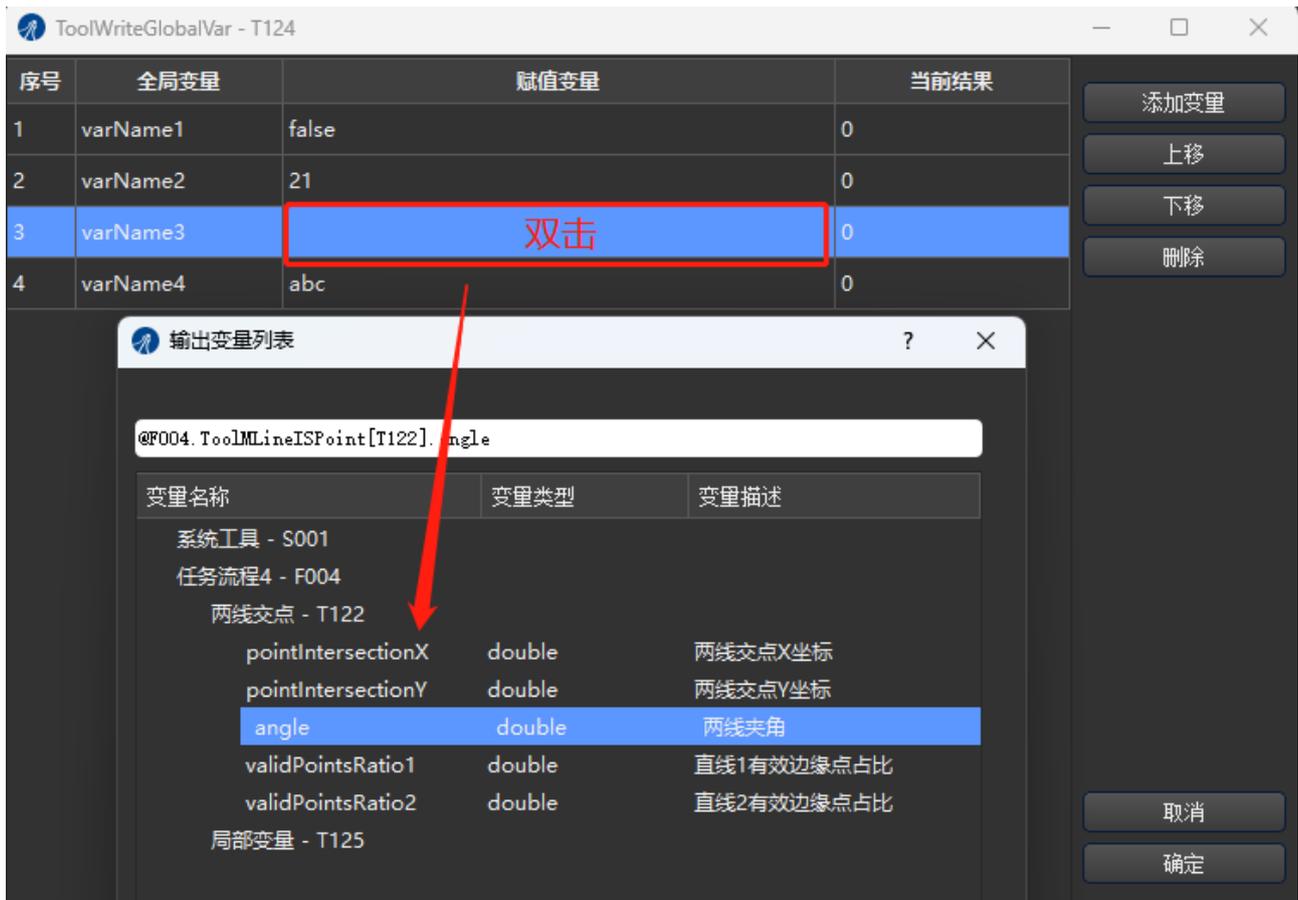
获取全局变量的值。需先在快捷工具条[配置全局变量](#)，再使用全局变量。点击“添加变量”在全局变量列表中勾选要读取的变量。



### 3.8.2 写入全局变量

设置全局变量的值。需先在快捷工具条[配置全局变量](#)，再使用全局变量。

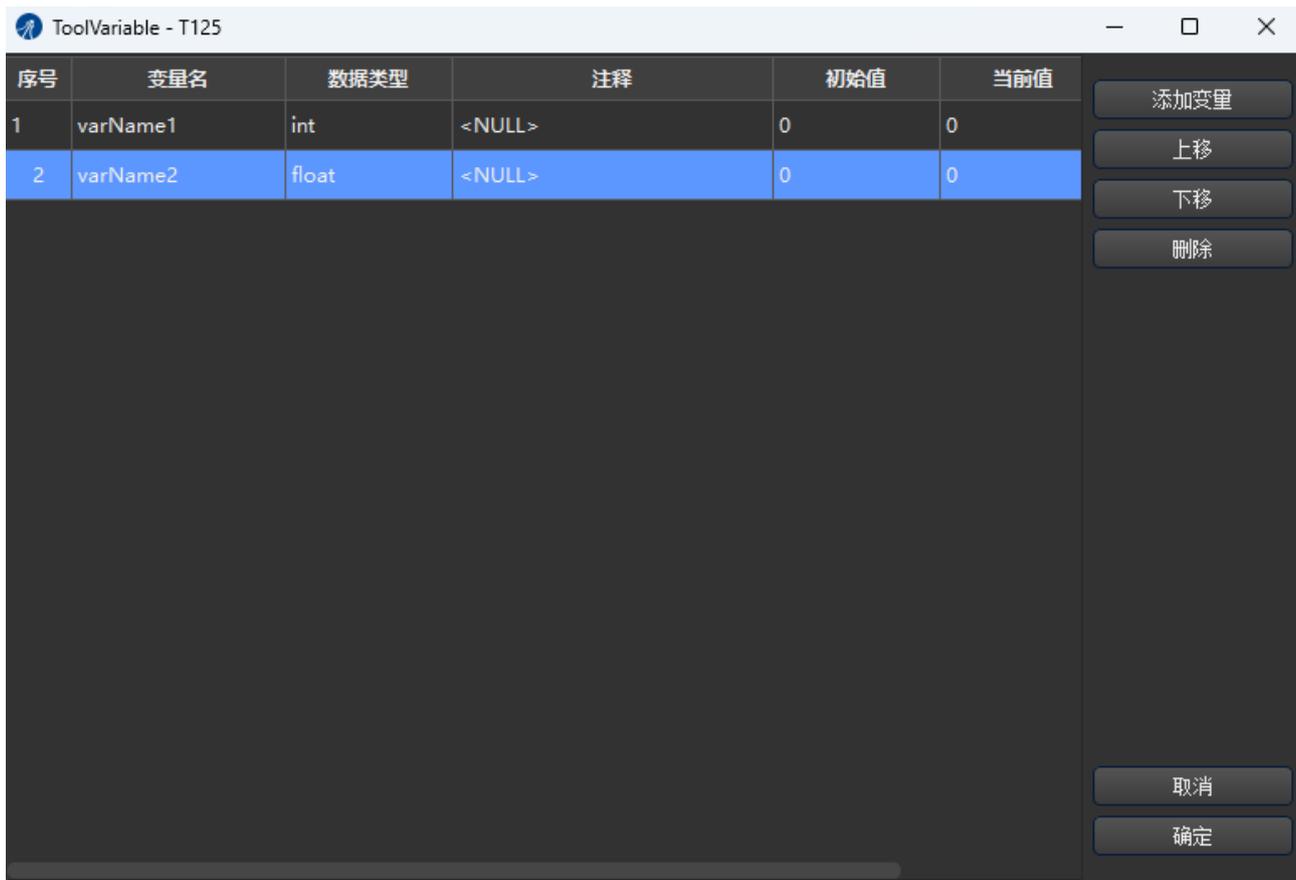
点击“添加变量”在全局变量列表中选择要写入的变量。添加变量后，双击“赋值变量”列，给添加的变量赋值。



- 添加变量：单击后弹出全局变量列表，选择写入的变量。
- 上移/下移：可对变量的位置进行上下调整。
- 删除：删除列表中当前选择的条目。
- 取消：取消添加的变量。
- 确定：保存添加的变量。

### 3.8.3 局部变量

局部变量是任务流程中自定义的变量，只能由创建它的任务流程调用或修改，可自定义变量名称、类型和初始值。



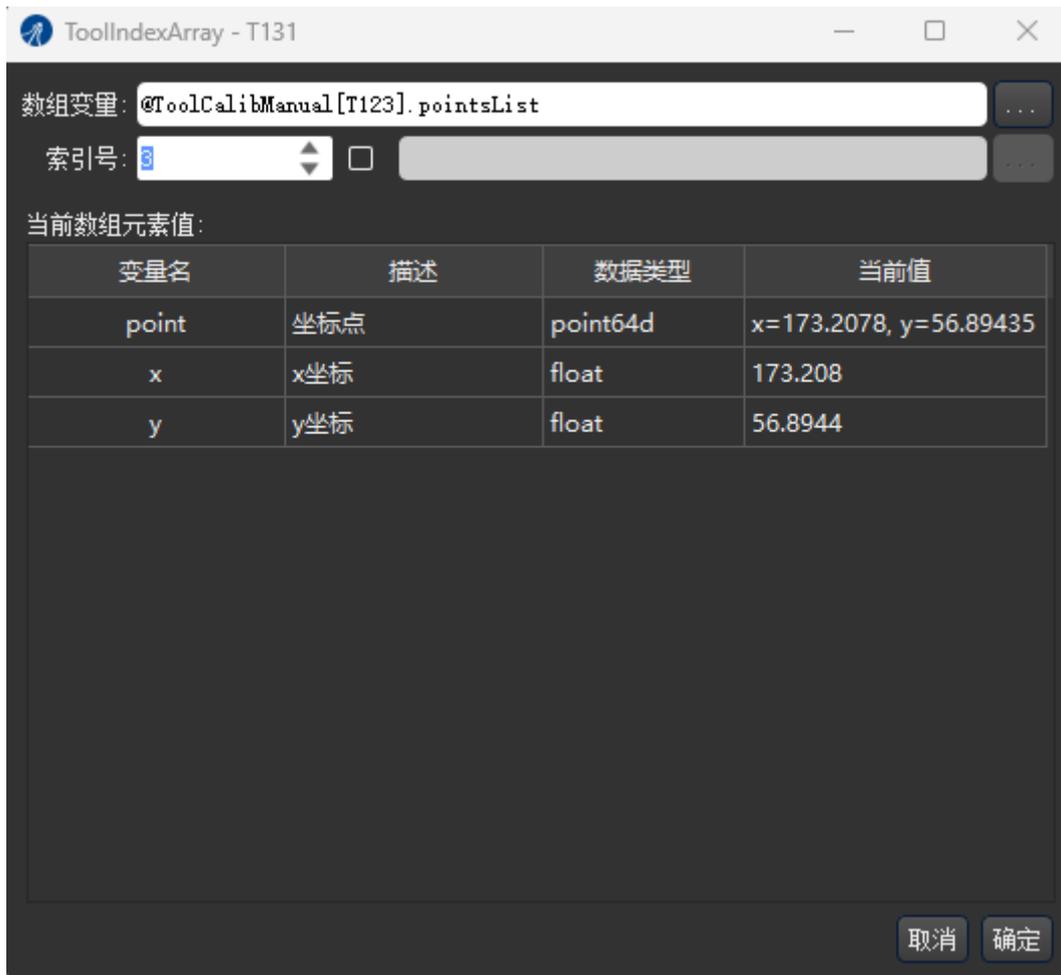
- 添加变量：单击后可新增全局变量。
- 上移/下移：可对变量的位置进行上下调整。
- 删除：删除列表中当前选择的条目。
- 取消：取消添加的变量。
- 确定：保存添加的变量。

### 3.8.4 写入局部变量

设置局部变量的值，需先使用“局部变量”工具创建变量。然后在局部变量列表中选择要写入的变量。操作方法参考[写入全局变量](#)。

### 3.8.5 数组索引

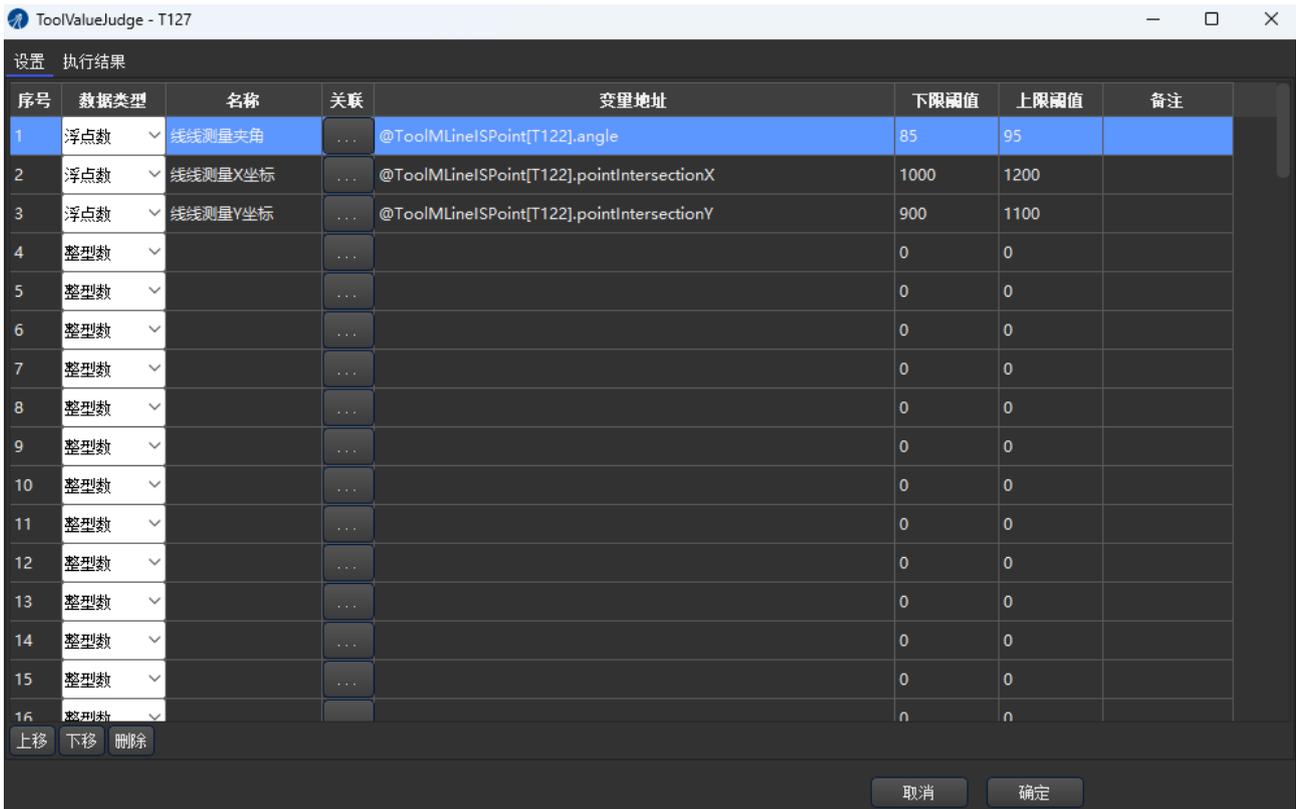
获取数组中指定索引号的元素。



- 数组变量：关联数组输出变量。
- 索引号：指定要获取的数组元素编号，编号范围[0, 数组长度-1]。

### 3.8.6 阈值判断

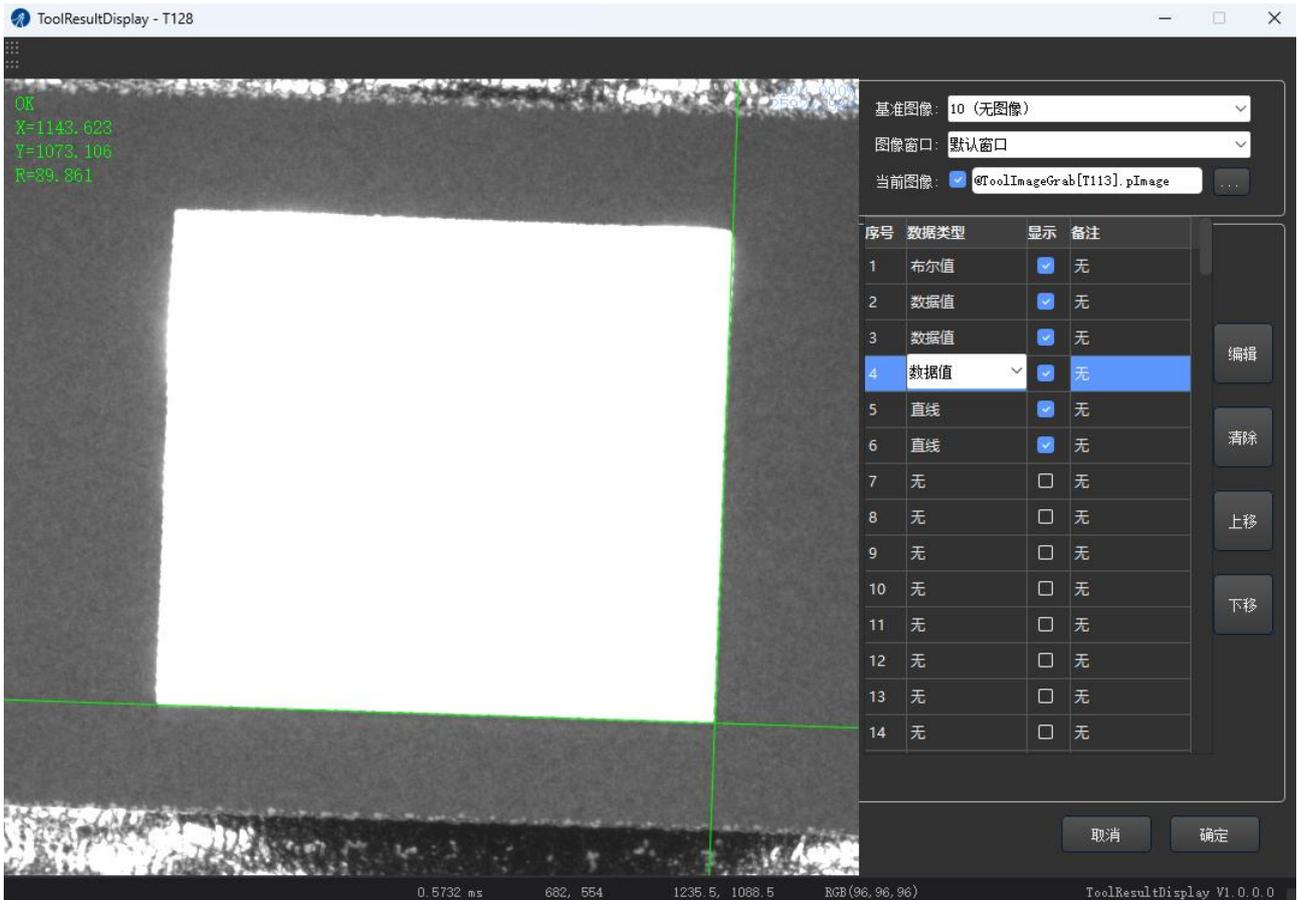
判断一组数据是否在设定的上下限阈值范围内，如果其中的一个数据超出范围则返回NG，一般用于输出结果的综合判定。



- 添加：添加需要判断上下限的数据。
- 删除：删除列表中当前选择的条目。
- 上移/下移：可对列表中当前选择的条目进行上下移动。

### 3.8.7 结果显示

用于在图像上显示检测结果数据，支持显示布尔值、数据值、字符串、点、线、圆、矩形。

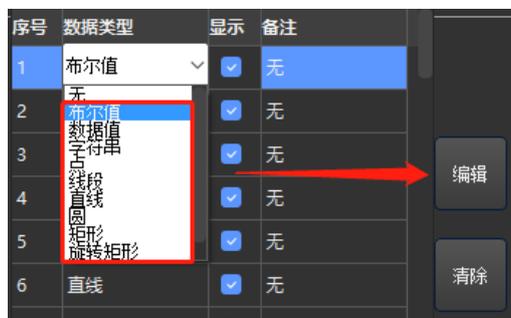


- 图像设置

- 基准图像：用户观察结果显示效果的图像。
- 图像窗口：选择结果数据显示的窗口。
- 当前图像：工具运行时显示结果数据的目标图像，一般绑定图像采集、图像处理等工具的输出图像。

- 数据设置

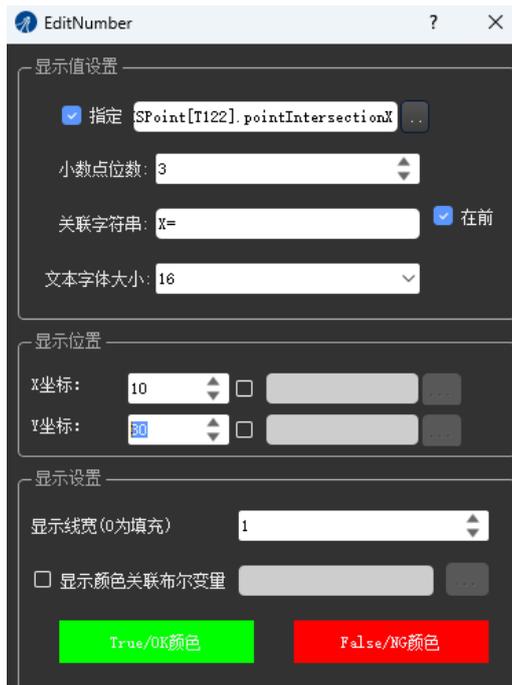
在数据类型下拉列表中，选择显示的条目，单击“编辑”会弹出该条目的设置页面。



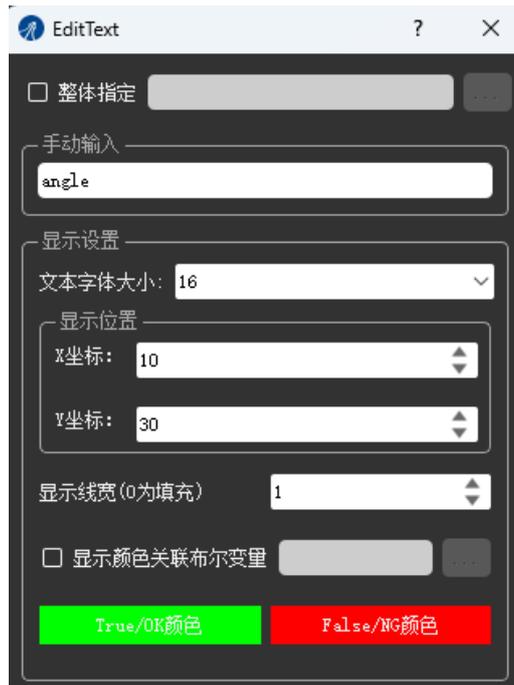
- 布尔值设置



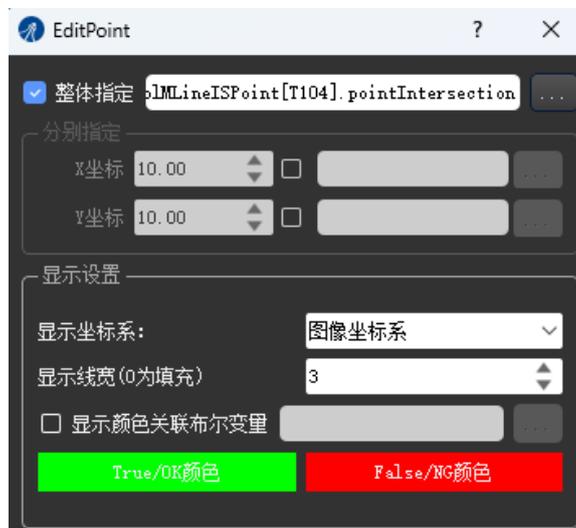
- 显示值需指定布尔变量来源。
- True 显示文本：设置布尔值为真时显示的文本。
- False 显示文本：设置布尔值为假时显示的文本。
- 文本字体大小：设置文本大小。
- X 坐标：设置文本显示的位置 X 轴坐标（窗口坐标系）。
- Y 坐标：设置文本显示的位置 Y 轴坐标（窗口坐标系）。
- 显示线宽：参数无效。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。
- 数据值设置



- 指定：设置需要显示的数据值来源，支持整型和浮点型。
- 小数点位数：指定需要显示的数据值小数位数。
- 关联字符串：在数据值前或后增加字符串一起显示。
- 文本字体大小：设置文本大小。
- X 坐标：设置文本显示的位置 X 轴坐标（窗口坐标系），可以关联来源。
- Y 坐标：设置文本显示的位置 Y 轴坐标（窗口坐标系），可以关联来源。
- 显示线宽：参数无效。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。
- 字符串设置



- 整体指定：设置需要显示的字符串来源。
- 文本字体大小：设置文本大小。
- X 坐标：设置文本显示的位置 X 轴坐标（窗口坐标系）。
- Y 坐标：设置文本显示的位置 Y 轴坐标（窗口坐标系）。
- 显示线宽：参数无效。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。
- 点设置



- 整体指定：设置需要显示的点来源。
- 分别指定：点按 X 坐标 + Y 坐标的方式指定，可以关联来源。
- 显示坐标系：可以显示在图像坐标系，也可以显示在窗口坐标系。
- 显示线宽：指定点的十字标识线的宽度。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

● 线设定



- 整体指定：设置需要显示的线段或直线来源。
- 分别指定：直线按中心点 + 方向向量的方式指定，线段按起点 + 终点的方式指定，可以关联来源。
- 显示线宽：指定线段或直线的轮廓宽度。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

● 圆设置



- 整体指定：设置需要显示的圆来源。
- 分别指定：按圆心坐标 + 半径的方式指定圆，可以关联来源。
- 显示线宽：指定圆轮廓线的宽度。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。
- 矩形设置



- 整体指定：设置需要显示的矩形来源。

- 分别指定：按矩形中心坐标 + 矩形宽高 + 角度的方式指定矩形，可以关联来源。
- 显示线宽：指定矩形轮廓线的宽度。
- 显示颜色关联布尔变量：勾选后设置文本显示颜色关联的变量。
- True/OK 颜色：当显示颜色关联的布尔值为真时文本的显示颜色。
- Flase/NG 颜色：当显示颜色关联的布尔值为假时文本的显示颜色。

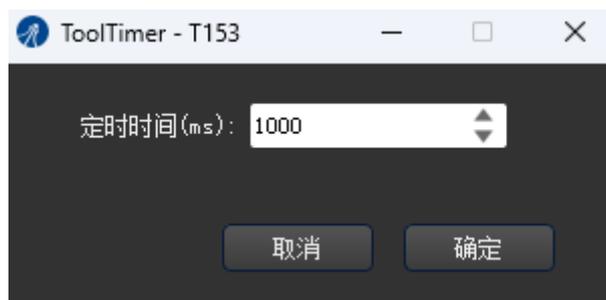
### 3.8.8 计数器

统计流程执行次数。



### 3.8.9 定时器

延时设定的时间，再执行之后的程序。



### 3.8.10 计时开始

记录开始计算时间的位置，该工具无参数配置。与“计时结束”工具搭配使用，用于计算流程中某一段落执行完成所需要的时间。

### 3.8.11 计时结束

计算“计时开始”到“计时结束”所产生的时间差，单位为毫秒。



- 计时开始工具：关联统计耗时的“计时开始”工具。

## 3.9 逻辑控制

### 3.9.1 IF 语句

满足设置的条件，则该工具下的所有子节点正常执行，否则不执行。可嵌套使用，可通过单操作数（非零为真）或双操作数进行比较判断。



- 操作数：单/双操作数选择。单操作数时，布尔类型 `true` 为真，整数/浮点类型非零值为真，字符串类型非空为真。双操作数时，按选择的运算符进行比较，满足比较条件为真。
- 数据类型：比较数据类型，支持布尔、整型、浮点型和字符串。
- 数据值：比较数据源，可关联。
- 运算符：双操作数时使用，操作数 1 与操作数 2 的比较方式，支持的比较方式为大于、小于、等于、大于等于、小于等于、不等于。

### 3.9.2 Else 语句

与之配对的 IF 条件（最近的 IF 判断-结束）为 `False` 时，该工具下的所有子节点正常执行，否则不执行。该工具无参数，与 IF 条件配套使用。

### 3.9.3 For 循环

设置的循环条件满足时，该工具下的所有子节点重复执行，否则不执行。每执行一次，工具内部的变量自动加 1，直至条件不满足或执行了循环中断语句。



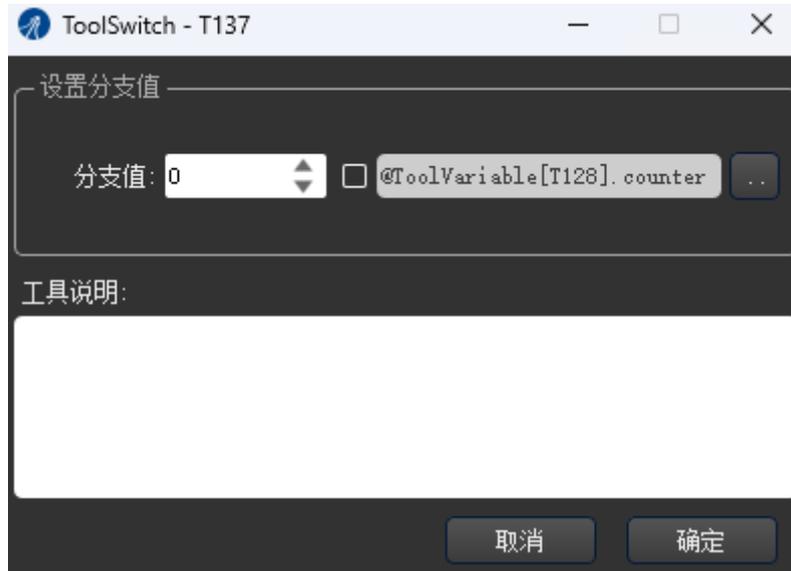
- 循环起始值：设置循环索引的初始值，从该值开始循环，可关联。
- 循环终止值：设置循环索引的结束值，小于该值时继续执行，可关联。
- 当前值：当前循环值。
- 复位：将当前循环值归零。

### 3.9.4 循环中断

配合循环工具使用，放在循环工具中间，即使循环还没完成也立即退出循环，执行此工具时，中断包含该工具最近的一个循环工具，该工具无参数配置。

### 3.9.5 选择分支

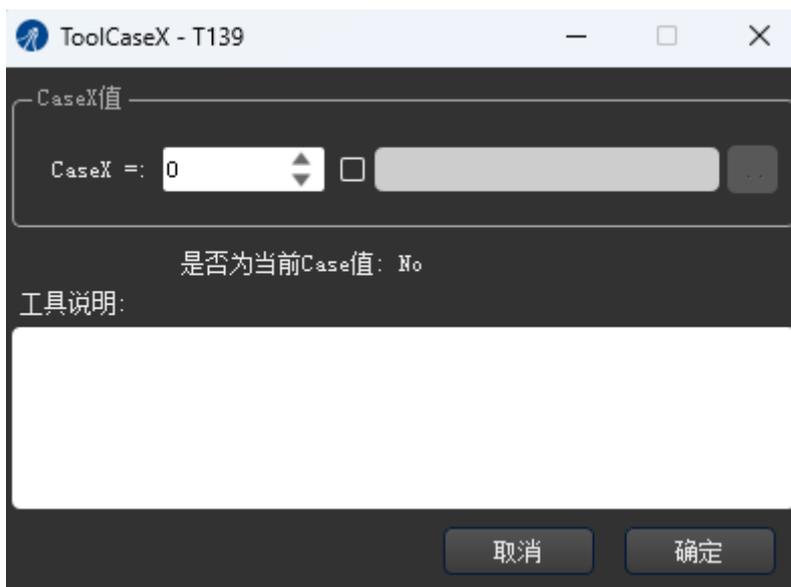
选择分支工具通过一个分支值来执行不同的分支，每个 CaseX 分支对应一个分支值，满足分支值时，对应的 CaseX 分支被执行。



- 分支值：关联一个用于控制分支的整型数据。

### 3.9.6 CaseX

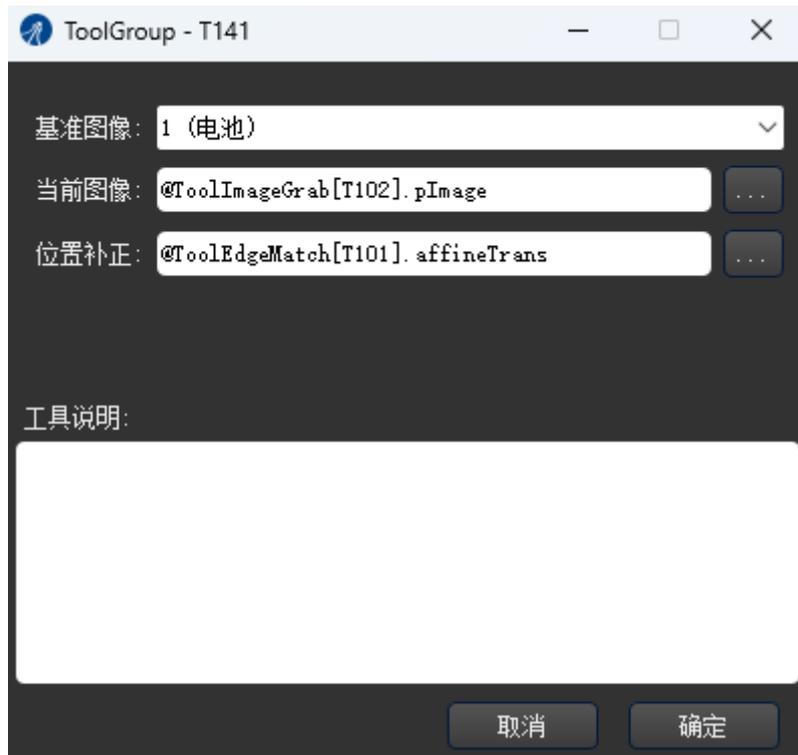
与选择分支的分支值匹配的，则该工具的所有子节点正常执行，否则不执行。



- CaseX=: 设置当前 CaseX 工具的分支值，可关联。

### 3.9.7 工具组

在复杂视觉方案中工具过多会导致查看不够直观，此时可利用工具组进行折叠整合，同时工具组也可配置基准图像、当前图像、位置补正等公共参数，该工具组下的所有子节点可继承使用，而无需对单个工具多次重复操作。



### 3.9.8 并行子流程

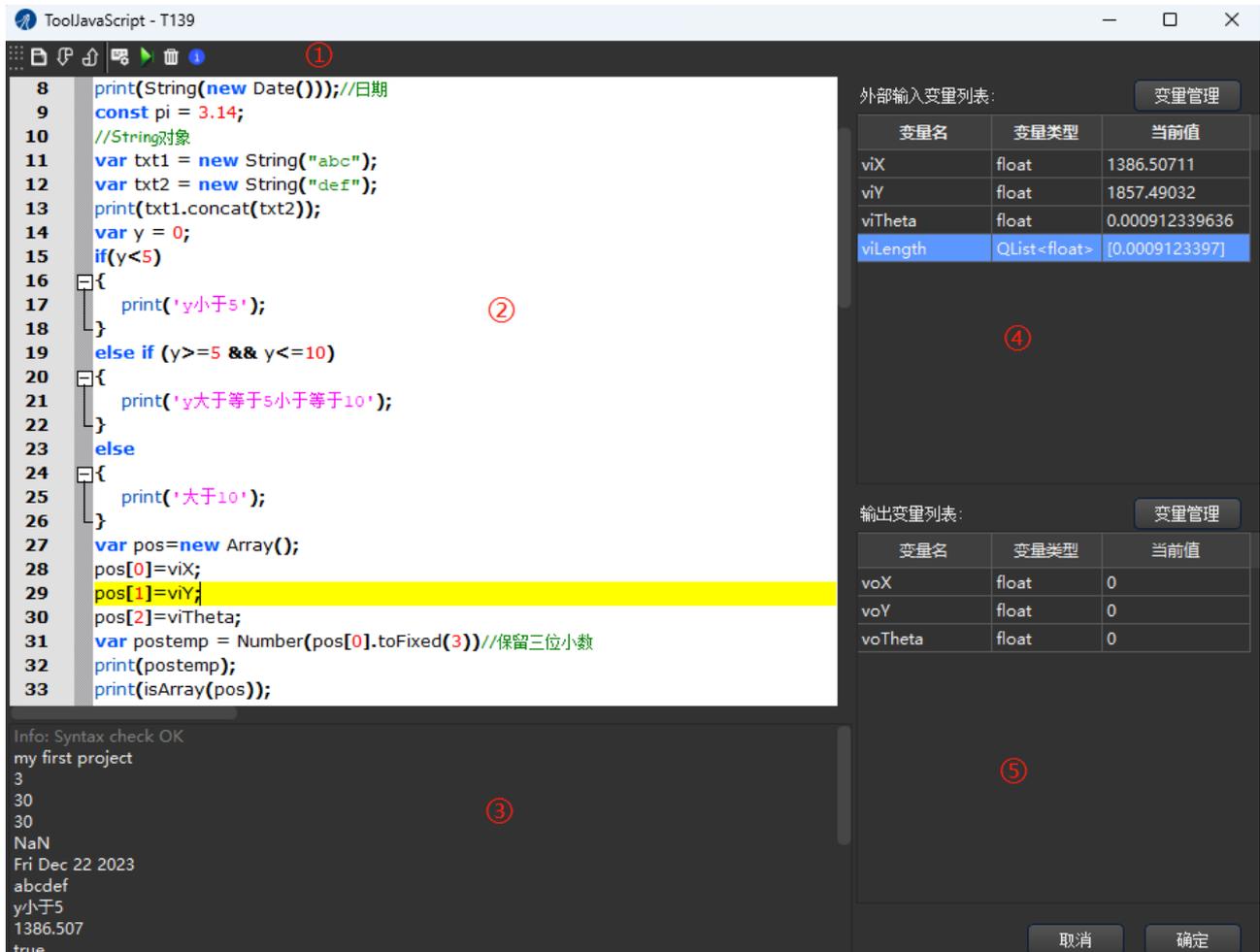
同时执行并行子流程到并行子流程结束之间的流程。若需要等待所有并行子流程执行完成再执行之后的程序，则需要结合同步工具，该工具无参数配置。

### 3.9.10 并行同步

与并行子流程搭配使用，等到前述并行子流程全部执行结束，再同步执行之后的流程，该工具无参数配置。

### 3.9.11 JavaScript 脚本

JavaScript 是一个轻量级，但功能强大的编程语言，使用 JavaScript 脚本可以进行复杂的数据处理。脚本工具可保存和加载已编写的脚本内容，脚本代码长度无限制，支持导入导出。



该脚本工具分为如下五个区域:

#### ① 工具栏

主要对脚本进行保存、导入、导出、运行测试及删除调试信息。

#### ② 代码编辑区

JavaScript 脚本编写区域，用户直接敲入代码即可。

#### ③ 调试窗口

程序编译信息输出，编译失败后将打印异常信息，用户也可通过 print 函数输出中间变量信息。

#### ④ 变量输入

外部传入脚本工具的变量，单击“变量管理”可添加变量并关联输入来源。

#### ⑤ 变量输出

脚本工具对外输出的变量，单击“变量管理”可添加输出变量。

### 3.9.11.1 输出

在编写脚本程序时，往往需要监视过程值，来确认代码执行是否正确，此时可以使用 `print` 语句打印输出到调试窗口，如下所示。

```
print("my first project");  
var a=30;  
print(a);
```

### 3.9.11.2 注释

注释可提高代码的可读性，脚本不会执行注释代码。单行注释以 `//` 开头，多行注释以 `/*` 开始，以 `*/` 结尾。

### 3.9.11.3 变量

变量用于存放值（如 `a=5`）和表达式（如 `z=x+y`）。变量命名应按以下原则：

- ◆ 变量必须以字母开头
- ◆ 变量也能以 `$` 和 `_` 符号开头（不推荐）
- ◆ 变量名称对大小写敏感（`a` 和 `A` 是不同的变量）

创建变量通常称为“声明”变量，使用 `var` 关键词来声明变量，如下声明变量 `theta`。

```
var theta;
```

变量声明之后，该变量没有值，未使用值来声明的变量，其值实际上是

**undefined, Undefined** 这个值表示变量不含有值，如需向变量赋值，请使用等号。

```
theta= 90;
```

可以通过将变量的值设置为 `null` 来清空变量。

```
theta= null;
```

也可以在声明变量时对其赋值 `var theta= 90;`

支持一条语句中声明多个变量。该语句以 `var` 开头，并使用逗号分隔变量。

```
var x=3.14, i=10, name="Zoe";
```

当然，为了明确数据类型，可以在声明新变量时，使用关键词 `"new"` 来声明其类型。

```
var name=new String;
```

```
var x=    new Number;
```

```
var y=    new Boolean;
```

```
var cars= new Array;
```

```
var person= new Object;
```

### 3.9.11.4 数据类型

JavaScript 脚本有 6 种不同的数据类型：字符串（string）、数字(number)、布尔(boolean)、对象(object)、函数(function)、符号（symbol）。3 种对象类型：对象(Object)、数组(Array)和日期（Date）。2 个不包含任何值的数据类型：空（null）和未定义（undefined）。

变量的数据类型可以使用 `typeof` 操作符来查看，实例如下：

```
typeof "Zoe"           // 返回 string
typeof 6.35            // 返回 number
typeof false          // 返回 boolean
typeof [1,2,3]        // 返回 object
typeof new Date()     // 返回 object
typeof {name:'Lucy', age:20} // 返回 object
typeof NaN            // 返回 number
typeof function () {} // 返回 function
typeof myHome         // 返回 undefined (如果 myHome 没有声明)
typeof null           // 返回 object
```

通过上例可以看出,如果对象是 `Array` 或 `Date` ,则无法通过 `typeof` 来判断他们的类型,因为都是返回 `object`, 此时可通过 `constructor` 属性, 在脚本语言中 `constructor` 属性返回所有变量的构造函数。实例如下:

```
"John".constructor      // 返回函数 String() { [native code] }
(3.14).constructor     // 返回函数 Number() { [native code] }
false.constructor      // 返回函数 Boolean() { [native code] }
[1,2,3].constructor    // 返回函数 Array() { [native code] }
{name:'Lucy', age:20}.constructor // 返回函数 Object() { [native code] }
new Date().constructor // 返回函数 Date() { [native code] }
function () {}.constructor // 返回函数 Function(){ [native code] }
```

使用 `constructor` 属性来查看对象是否为数组 (包含字符串 `"Array"`):

```
function isArray(myArray)
{
    return myArray.constructor.toString().indexOf("Array") > -1;
}
```

使用 `constructor` 属性来查看对象是否为日期 (包含字符串 `"Date"`):

```
function isDate(myDate)
{
    return myDate.constructor.toString().indexOf("Date") > -1;
}
```

### 3.9.11.4.1 字符串

字符串是存储字符 (如 `"position"`) 的变量, 引号中的任意文本都是字符串, 可以使用单引号或双引号。

```
var answer = "right";
var answer = 'right';
```

反斜杠是一个转义字符, 转义字符将特殊字符转换为字符串字符, 转义字符 `(\)` 可以用于转义撇号、换行、引号等其他特殊字符。例如, 下面的 `str` 将输出 `This is big "A"`。

```
var str = "This is big \A\"";
```

下表中列举了在字符串中可以使用转义字符转义的特殊字符。

属性	描述
\'	单引号
\"	双引号
\\	反斜杠
\n	换行
\r	回车
\t	tab(制表符)
\b	退格符
\f	换页符

字符串属性如下表：

属性	描述
constructor	返回创建字符串属性的函数
length	返回字符串的长度
prototype	允许您向对象添加属性和方法

字符串方法如下表：

方法	描述
charAt()	返回指定索引位置的字符
charCodeAt()	返回指定索引位置字符的 Unicode 值
concat()	连接两个或多个字符串，返回连接后的字符串
fromCharCode()	将 Unicode 转换为字符串
indexOf()	返回字符串中检索指定字符第一次出现的位置
lastIndexOf()	返回字符串中检索指定字符最后一次出现的位置
match()	找到一个或多个正则表达式的匹配
replace()	替换与正则表达式匹配的子串
search()	检索与正则表达式相匹配的值
slice()	提取字符串的片断，并在新的字符串中返回被提取的部分

split()	把字符串分割为子字符串数组
substr()	从起始索引号提取字符串中指定数目的字符
substring()	提取字符串中两个指定的索引号之间的字符
toLowerCase()	把字符串转换为小写
toUpperCase()	把字符串转换为大写
toString()	返回字符串对象值
trim()	移除字符串首尾空白
valueOf()	返回某个字符串对象的原始值

实例:

```
var str="Hello World!";
print(str.length);// 输出 12
print(str.indexOf("or"));//输出 7
txt="a,b,c" ;
txt.split(","); // 使用逗号分隔提取 a b c, 可用数组 new Array()存储提取后的值
```

### 3.9.11.4.2 数字

JavaScript 脚本不是类型语言。与许多其他编程语言不同，它不定义不同类型的数字，比如整数、短、长、浮点等，在脚本语言中数字统一为 `Number`。所有的数字都是浮点型类型。JavaScript 采用 IEEE754 标准定义的 64 位浮点格式表示数字，它能表示最大值 (`Number.MAX_VALUE`) 为  $\pm 1.7976931348623157e+308$ ，最小值 (`Number.MIN_VALUE`) 为  $\pm 5e-324$ 。默认情况下，数字为十进制；如果前缀为 0，则为八进制数；如果前缀为 0x，则为十六进制数。

```
var x=34.56; //使用小数点
var y=12; //不使用小数点
var z=123e5; // 12300000
var u = 0377; //前缀 0 为 8 进制
var v = 0xFF; //前缀 0x 为 16 进制
```

数字可以是数字也可以是对象。如下实例:

var x = 123;

var y = new Number(123);

typeof(x) // 返回 Number

typeof(y) // 返回 Object

Number 属性

方法	描述
Number.MAX_VALUE	最大值, 1.7976931348623157e+308
Number.MIN_VALUE	最小值, 5e-324
Number.NaN	非数字, NaN
Number.NEGATIVE_INFINITY	负无穷, 在溢出时返回
Number.POSITIVE_INFINITY	正无穷, 在溢出时返回

数字原型的方法:

方法	描述
toExponential()	把对象的值转换为指数计数法。如: 1.23e+2
toFixed()	把数字转换为字符串, 结果的小数点后有指定位数的数字。 var a=123; b=a.toFixed(2); // b="123.00"
toPrecision()	把数字格式化为指定的长度。如下例中, a=123 中, 3 会由于精度限制消失。 var a=123; b=a.toPrecision(2); // b="1.2e+2"

数字对象的方法:

方法	描述
Number.parseFloat()	将字符串转换成浮点数。
Number.parseInt()	将字符串转换成整型数字。
Number.isFinite()	判断传递的参数是否为有限数字。
Number.isInteger()	判断传递的参数是否为整数。
Number.isNaN()	判断传递的参数是否为 isNaN()。

Number.isSafeInteger()	判断传递的参数是否为安全整数。
------------------------	-----------------

### 3.9.11.4.3 布尔

布尔（逻辑）只有两个值：`true` 或 `false`，常用在条件语句中。

```
var a=true;
```

```
var b=false;
```

`Boolean`（布尔）对象用于将非布尔值转换为布尔值（`true` 或者 `false`）。布尔对象定义如下：

```
var myBoolean=new Boolean();
```

`Boolean` 对象属性

方法	描述
constructor	返回对创建此对象的 <code>Boolean</code> 函数的引用

`Boolean` 对象方法

方法	描述
toString()	把布尔值转换为字符串，并返回结果。
valueOf()	返回 <code>Boolean</code> 对象的原始值。

### 3.9.11.4.4 数组

数组（`Array`）对象是一组数据的集合，数组下标是基于零的，所以第一个元素是 `[0]`，第二个是 `[1]`，以此类推，下面的代码创建名为 `cars` 的数组：

```
var cars=new Array();
```

```
cars[0]="A";
```

```
cars[1]="B";
```

```
cars[2]="M";
```

`Array` 对象属性

属性	描述
constructor	返回创建数组对象的原型函数。
length	设置或返回数组元素的个数。
prototype	<p>允许你向数组对象添加属性或方法。如下实例给数组增加一个新方法 myFun，并在 fruits 数组实例中调用，将其数组值全部转为大写。</p> <pre> Array.prototype.myFun = function() {     for (i=0;i&lt;this.length;i++)     {         this[i]=this[i].toUpperCase();     } }  var fruits=["Banana","Orange","Apple","Mango"]; fruits.myUcase(); </pre>

### Array 对象方法

方法	描述
concat()	<p>连接两个或更多的数组，并返回结果。实例如下：</p> <pre> var a = ["year", "month"]; var b = ["date", "time"]; var c = a.concat(b); </pre>
copyWithin()	<p>从数组的指定位置拷贝元素到数组的另一个指定位置中。语法如下：</p> <pre> array.copyWithin(target, start, end) </pre>
every()	<p>检测数值元素的每个元素是否都符合条件。如下实例为检测数组 <i>ages</i> 的所有元素是否都大于等于 18。</p> <pre> var ages = [51, 22, 16, 35]; function checkAdult(age) {     return age &gt;= 18; } </pre>

	<code>ages.every(checkAdult);</code>
<code>filter()</code>	<p>检测数值元素，并返回符合条件所有元素的数组。在上例中返回数组 <code>ages</code> 中所有元素都大于 18 的元素。</p> <p><code>ages.filter(checkAdult);</code></p>
<code>indexOf()</code>	<p>搜索数组中的元素，并返回它所在的位置。语法如下：</p> <p><code>array.indexOf(item,start);</code></p>
<code>isArray()</code>	<p>判断对象是否为数组。语法如下：</p> <p><code>Array.isArray(obj);</code></p>
<code>join()</code>	<p>把数组的所有元素放入一个字符串。语法如下：</p> <p><code>array.join(separator);</code></p> <p>省略 <code>separator</code> 参数，则使用逗号作为分隔符。</p>
<code>lastIndexOf()</code>	<p>搜索数组中的元素，并返回它最后出现的位置。语法如下：</p> <p><code>array.lastIndexOf(item,start);</code></p>
<code>map()</code>	<p>返回一个新数组，数组中的元素为原始数组元素调用函数处理后的值。语法如下：</p> <p><code>array.map(function(currentValue,index,arr), thisValue);</code></p> <p>如下实例返回一个数组，数组中元素为原始数组的平方根。</p> <p><code>var numbers = [4, 9, 16, 25];</code></p> <p><code>numbers.map(Math.sqrt);</code></p>
<code>pop()</code>	<p>删除数组的最后一个元素并返回删除的元素。语法如下：</p> <p><code>array.pop()</code></p>
<code>push()</code>	<p>向数组的末尾添加一个或更多元素，并返回新的长度。语法如下：</p> <p><code>array.push(item1, item2, ..., itemX);</code></p>
<code>reduce()</code>	<p>将数组元素计算为一个值（从左到右）。语法如下：</p> <p><code>array.reduce(function(total, currentValue, currentIndex, arr), initialValue);</code></p> <p>如下实例求 <code>numbers</code> 数组的和为 54。</p> <p><code>var numbers = [4, 9, 16, 25];</code></p> <p><code>function getSum(total, num)</code></p>

	<pre> {     return total + num; } numbers.reduce(getSum); </pre>
reduceRight()	将数组元素计算为一个值（从右到左）。
reverse()	<p>反转数组的元素顺序。语法如下：</p> <pre>array.reverse();</pre>
shift()	<p>删除并返回数组的第一个元素。语法如下：</p> <pre>array.shift();</pre>
slice()	<p>选取数组的一部分，并返回一个新数组。语法如下：</p> <pre>array.slice(start, end);</pre>
some()	<p>检测数组元素中是否有元素符合指定条件。语法如下：</p> <pre>array.some(function(currentValue,index,arr),thisValue);</pre> <p>如下实例将输出 true。</p> <pre>var ages = [51, 22, 16, 35]; function checkAdult(age) {     return age &gt;= 18; } ages. some (checkAdult);</pre>
sort()	<p>对数组的元素进行排序。语法如下：</p> <pre>array.sort();</pre>
splice()	<p>从数组中添加或删除元素，会改变原始数组。语法如下：</p> <pre>array.splice(indexadd,howmanydel,item1,.....,itemX);</pre>
toString()	<p>把数组转换为字符串，并返回结果。语法如下：</p> <pre>array.toString();</pre>
unshift()	<p>向数组的开头添加一个或更多元素，并返回新的长度。语法如下：</p> <pre>array.unshift(item1,item2, ..., itemX);</pre>
valueOf()	返回数组对象的原始值。语法如下：

array.valueOf();
------------------

### 3.9.11.4.5 对象

JavaScript 对象是拥有属性和方法的数据。对象也是一个变量，但对象可以包含多个值（多个变量），每个值以 name:value 对呈现。在以下实例中，3 个值 ("Axis", 500, 100) 赋予变量 point。

```
var point = { name:"Axis", x:500, y:100};
```

- 对象定义

可以使用字符来定义和创建 JavaScript 对象，定义对象可以跨越多行。

```
var robot = {  
  name:"Nio",  
  x:10,  
  y:10,  
  z:30,  
  methodSum:function(a,b){  
    return a+b;  
  }  
};
```

- 对象属性

对象是变量的容器”。但是，通常认为 "JavaScript 对象是键值对的容器”。键值对通常写法为 name : value (键与值以冒号分割)，键值对即为对象属性。

- 访问对象属性

可以通过两种方式访问对象属性：

```
person.lastName;
```

```
person["lastName"];
```

- 对象方法

对象的方法定义了一个函数，并作为对象的属性存储。对象方法通过添加 () 调用，可以使用以下语法创建对象方法：

```
methodName : function() {  
    // 代码  
}
```

使用以下语法访问对象方法：

```
objectName.methodName()
```

如访问上文定义的 robot 对象的 methodSum 方法，该方法返回值为 7。

```
robot.methodSum (3,4);
```

### 3.9.11.4.6 Math 对象

Math 对象的作用是执行普通的算数任务。Math 对象提供多种算数值类型和函数。使用前无需对它进行定义。

使用 Math 的属性/方法的语法如下：

```
var x = Math.PI;
```

```
var y = Math.sqrt(16);
```

Math 对象属性

属性	描述
E	返回算术常量 e，即自然对数的底数（约等于 2.718）。
LN2	返回 2 的自然对数（约等于 0.693）。
LN10	返回 10 的自然对数（约等于 2.302）。
LOG2E	返回以 2 为底的 e 的对数（约等于 1.4426950408889634）。
LOG10E	返回以 10 为底的 e 的对数（约等于 0.434）。
PI	返回圆周率（约等于 3.14159）。
SQRT1_2	返回 2 的平方根的倒数（约等于 0.707）。
SQRT2	返回 2 的平方根（约等于 1.414）。

Math 对象方法

方法	描述
abs(x)	返回 x 的绝对值。
acos(x)	返回 x 的反余弦值。
asin(x)	返回 x 的正弦值。
atan(x)	以介于 $-\pi/2$ 与 $\pi/2$ 弧度之间的数值来返回 x 的正切值。
atan2(y,x)	返回从 x 轴到点 (x,y) 的角度（介于 $-\pi/2$ 与 $\pi/2$ 弧度之间）。
ceil(x)	对数进行上舍入。
cos(x)	返回数的余弦。
exp(x)	返回 $E^x$ 的指数。
floor(x)	对 x 进行下舍入。
log(x)	返回数的自然对数（底为 e）。
max(x,y,z,...,n)	返回 x,y,z,...,n 中的最高值。
min(x,y,z,...,n)	返回 x,y,z,...,n 中的最低值。
pow(x,y)	返回 x 的 y 次幂。
random()	返回 0~1 之间的随机数。
round(x)	四舍五入。
sin(x)	返回数的正弦。
sqrt(x)	返回数的平方根。
tan(x)	返回角的正切。
tanh(x)	返回一个数的双曲正切函数值。
trunc(x)	将数字的小数部分去掉，只保留整数部分。

### 3.9.11.4.7 Date 对象

Date 对象用于处理日期与时间。

创建 Date 对象如下：

```
var d = new Date();
```

## Date 对象属性

属性	描述
constructor	返回对创建此对象的 Date 函数的引用。
prototype	允许向对象添加属性和方法

## Date 对象方法

方法	描述
getDate()	从 Date 对象返回一个月中的某一天 (1 ~ 31)。
getDay()	从 Date 对象返回一周中的某一天 (0 ~ 6)。
getFullYear()	从 Date 对象以四位数字返回年份。
getHours()	返回 Date 对象的小时 (0 ~ 23)。
getMilliseconds()	返回 Date 对象的毫秒(0 ~ 999)。
getMinutes()	返回 Date 对象的分钟 (0 ~ 59)。
getMonth()	从 Date 对象返回月份 (0 ~ 11)。
getSeconds()	返回 Date 对象的秒数 (0 ~ 59)。
getTime()	返回 1970 年 1 月 1 日至今的毫秒数。
getTimezoneOffset()	返回本地时间与格林威治标准时间 (GMT) 的分钟差。
getUTCDate()	根据世界时从 Date 对象返回月中的一天 (1 ~ 31)。
getUTCDay()	根据世界时从 Date 对象返回周中的一天 (0 ~ 6)。
getUTCFullYear()	根据世界时从 Date 对象返回四位数的年份。
getUTCHours()	根据世界时返回 Date 对象的小时 (0 ~ 23)。
getUTCMilliseconds()	根据世界时返回 Date 对象的毫秒(0 ~ 999)。
getUTCMinutes()	根据世界时返回 Date 对象的分钟 (0 ~ 59)。
getUTCMonth()	根据世界时从 Date 对象返回月份 (0 ~ 11)。
getUTCSeconds()	根据世界时返回 Date 对象的秒钟 (0 ~ 59)。
setDate()	设置 Date 对象中月的某一天 (1 ~ 31)。
setFullYear()	设置 Date 对象中的年份 (四位数字)。
setHours()	设置 Date 对象中的小时 (0 ~ 23)。
setMilliseconds()	设置 Date 对象中的毫秒 (0 ~ 999)。

setMinutes()	设置 Date 对象中的分钟 (0 ~ 59)。
setMonth()	设置 Date 对象中月份 (0 ~ 11)。
setSeconds()	设置 Date 对象中的秒钟 (0 ~ 59)。
setTime()	setTime() 方法以毫秒设置 Date 对象。
setUTCDate()	根据世界时设置 Date 对象中月份的一天 (1 ~ 31)。
setUTCFullYear()	根据世界时设置 Date 对象中的年份 (四位数字)。
setUTCHours()	根据世界时设置 Date 对象中的小时 (0 ~ 23)。
setUTCMilliseconds()	根据世界时设置 Date 对象中的毫秒 (0 ~ 999)。
setUTCMinutes()	根据世界时设置 Date 对象中的分钟 (0 ~ 59)。
setUTCMonth()	根据世界时设置 Date 对象中的月份 (0 ~ 11)。
setUTCSeconds()	setUTCSeconds() 方法用于根据世界时 (UTC) 设置指定时间的秒字段。
toDateString()	把 Date 对象的日期部分转换为字符串。
toJSON()	以 JSON 数据格式返回日期字符串。
toLocaleDateString()	根据本地时间格式, 把 Date 对象的日期部分转换为字符串。
toLocaleTimeString()	根据本地时间格式, 把 Date 对象的时间部分转换为字符串。
toLocaleString()	根据本地时间格式, 把 Date 对象转换为字符串。
toString()	把 Date 对象转换为字符串。
toTimeString()	把 Date 对象的时间部分转换为字符串。
toUTCString()	根据世界时, 把 Date 对象转换为字符串。
valueOf()	返回 Date 对象的原始值。

### 3.9.11.4.8 类型转换

- String()方法

String() 方法可以将数字、布尔、日期等类型转换为字符串。实例如下:

```
String(123);    // 将数字 123 转换为字符串并返回
```

```
String(false);    // 返回 "false"
String(new Date()); // 返回 Mon Dec 25 2023
Number 方法 toString() 也能实现同样的效果。
(123).toString();
false.toString(); // 返回 "false"
obj = new Date();
obj.toString(); // 返回 Mon Dec 25 2023
```

- Number()方法

Number() 方法可以将字符串、布尔、日期转换为数字。其中空字符串转换为 0，非数字字符串会转换为 NaN (不是数字)。实例如下：

```
Number("3.14"); // 返回 3.14
Number(" ");    // 返回 0
Number("99 ab"); // 返回 NaN
Number(true);   // 返回 1
d = new Date();
Number(d);      // 返回 1703483796513
```

### 3.9.11.5 函数

函数是包裹在大括号中的代码块，前面使用了关键词 `function`，当调用该函数时，会执行函数内的代码。函数内部声明的变量是局部变量，只能在函数内部访问它。

```
function myFunction ()
{
    // 执行代码
}
```

函数也可以带参数，参数由逗号 (,) 分隔，在调用带参数的函数时，可以向其传递值，这些值可以在函数中使用。

```
function myFunction(var1,var2)
{
```

```
// 执行代码  
}
```

函数也可以带返回值，使用 `return` 语句。

```
function myFunction(a,b)  
{  
    return a*b;  
}
```

### 3.9.11.6 运算符

#### 3.9.11.6.1 算术运算

算术运算符用于给变量进行四则运算。

运算符	描述	举例	运算结果
+	加法	$x = y + 3$ , 式中 $y$ 值为 5	8
-	减法	$x = y - 3$ , 式中 $y$ 值为 5	2
*	乘法	$x = y * 2$ , 式中 $y$ 值为 5	10
/	除法	$x = y / 2$ , 式中 $y$ 值为 5	2.5
%	取模（余数）	$x = y \% 2$ , 式中 $y$ 值为 5	1
++	自增	$x = y++$ , 式中 $y$ 值为 5 $x = ++y$ , 式中 $y$ 值为 5	5 6
--	自减	$x = y--$ , 式中 $y$ 值为 5 $x = --y$ , 式中 $y$ 值为 5	5 4

#### 3.9.11.6.2 字符串加法

`+` 运算符可用于把文本值或字符串变量加起来（连接起来）。

如需把两个或多个字符串变量连接起来，请使用 `+` 运算符，如下实例：

```
txt1 = "How are you ";
```

```
txt2 = "Today";
```

```
txt3 = txt1 + txt2;
```

txt3 运算结果为"How are you Today"。

如果数字和字符串相加，将返回字符串，如下实例：

```
x = "5"+5;
```

```
y = "Hio"+5;
```

x 输出 55， y 输出 Hio5。

### 3.9.11.6.3 赋值运算

赋值运算符用于给变量赋值。

运算符	描述	举例	运算结果
=	等于	x = y,式中 y 值为 5	5
+=	加等	x += y,式中 y 值为 5,x 初始值为 5	10
-=	减等	x -= y,式中 y 值为 5,x 初始值为 5	0
*=	乘等	x *= y,式中 y 值为 5,x 初始值为 5	25
/=	除等	x /= y,式中 y 值为 5,x 初始值为 5	1
%=	模等	x %= y,式中 y 值为 5,x 初始值为 5	0

### 3.9.11.6.4 比较运算

比较运算符在逻辑语句中使用，以测定变量或值是否相等。

运算符	描述	举例	运算结果
==	等于	x == 1,式中 x 初始值为 5	false
!=	不等于	x != 1,式中 x 初始值为 5	true
>	大于	x > 1,式中 x 初始值为 5	true
<	小于	x < 1,式中 x 初始值为 5	false
>=	大于或等于	x >= 5,式中 x 初始值为 5	true
<=	小于或等于	x <= 5,式中 x 初始值为 5	true

### 3.9.11.6.5 逻辑运算符

逻辑运算符用于测定变量或值之间的逻辑。

给定  $x = 5$ 、 $y = 2$ ，下表解释了逻辑运算符：

运算符	描述	举例	运算结果
&&	and	$(x > 1 \ \&\& \ y < 10)$	true
	or	$(x == 5 \    \ y == 7)$	true
!	not	$!(x == y)$	true

### 3.9.11.6.6 条件运算符

脚本包含基于某些条件对变量进行赋值的条件运算符。语法如下：

`Variablename = (condition)?value1:value2 ;`

如果 `condition` 条件为真，输出 `value1`，否则，输出 `value2`。

`ang=(theta<30)?"角度正常":"角度过大";`

### 3.9.11.6.7 条件语句

条件语句用于基于不同的条件来执行不同的动作。

- if 语句

只有当指定条件为 `true` 时，该语句才会执行代码。语法如下：

```
if (condition)
{
    //当条件为 true 时执行的代码;
}
```

- if...else 语句

当条件为 `true` 时执行代码，当条件为 `false` 时执行其他代码。语法如下：

```
if (condition)
{
```

```
    //当条件为 true 时执行的代码;
}
else
{
    //当条件不为 true 时执行的代码;
}
```

- **if...else if...else 语句**

使用该语句来选择多个代码块之一来执行。语法如下：

```
if (condition1)
{
    //当条件 1 为 true 时执行的代码;
}
else if (condition2)
{
    // 当条件 2 为 true 时执行的代码;
}
else
{
    //当条件 1 和 条件 2 都不为 true 时执行的代码;
}
```

### **3.9.11.6.8 switch 语句**

switch 语句用于基于不同的条件来执行不同的动作。语法如下：

```
switch(n)
{
    case 1:
        //执行代码块 1;
        break;
```

```
case 2:
    //执行代码块 2;
    break;
default:
    //与 case 1 和 case 2 不同时执行的代码;
}
```

### 3.9.11.6.9 循环

- for 循环

for 循环可以将代码块执行指定的次数。语法如下：

```
for (语句 1; 语句 2; 语句 3)
{
    //被执行的代码块;
}
```

语句 1：（代码块）开始前执行。

语句 2：定义运行循环（代码块）的条件。

语句 3：在循环（代码块）已被执行之后执行。

- while 循环

while 只要指定条件为 true，循环就可以一直执行代码块。语法如下：

```
while (条件)
{
    //需要执行的代码;
}
```

- do/while 循环

do/while 循环会在检查条件是否为真之前执行一次代码块，然后如果条件为真的话，就会重复这个循环。语法如下：

```
do
{
```

```
//需要执行的代码;  
}  
while (条件);
```

### 3.9.11.6.10 中断语句

- break 语句

break 语句用于跳出循环。break 语句跳出循环后，会继续执行该循环之后的代码。如下实例，当 i 等于 6 时，将退出 for 循环。

```
for (i=0;i<10;i++)  
{  
    if (i==6)  
    {  
        break;  
    }  
    x= "The number is " + i;  
}
```

- continue 语句

continue 语句中断当前的循环中的迭代，然后继续循环下一个迭代

### 3.9.11.7 正则表达式

正则表达式（regex）使用单个字符串来描述、匹配一系列符合某个句法规则的字符串搜索模式。搜索模式可用于文本搜索和文本替换。语法如下：

/正则表达式主体/修饰符(可选)

实例：

```
var a= /Lead/i
```

实例解析：

Lead 是一个正则表达式主体 (用于检索)。

i 是一个修饰符 (搜索不区分大小写)。

- 正则表达式修饰符

修饰符	描述
i	执行对大小写不敏感的匹配。
g	执行全局匹配 (查找所有匹配而非在找到第一个匹配后停止)。
m	执行多行匹配。

- 正则表达式模式

方括号用于查找某个范围内的字符。

表达式	描述
[abc]	查找方括号之间的任何字符。
[0-9]	查找任何从 0 至 9 的数字。
(x y)	查找任何以   分隔的选项。

- 元字符

元字符是拥有特殊含义的字符。

元字符	描述
\d	查找数字。
\s	查找空白字符。
\b	匹配单词边界。
\uxxxx	查找以十六进制数 xxxx 规定的 Unicode 字符。

- 量词

元字符	描述
n+	匹配任何包含至少一个 n 的字符串。
n*	匹配任何包含零个或多个 n 的字符串。
n?	匹配任何包含零个或一个 n 的字符串。

在脚本中，正则表达式通常用于两个字符串方法：`search()` 和 `replace()`。

`search()` 方法用于检索字符串中指定的子字符串，或检索与正则表达式相匹配的子字符串，并返回子串的起始位置。

replace() 方法用于在字符串中用一些字符串替换另一些字符串，或替换一个与正则表达式匹配的子串。

实例：

```
var str = "Welcom to LeadShine!";  
var n = str.search(/lead/i);    // 返回 10  
var txt = str.replace(/leadshine /i,"LS");//返回 Welcom to LS!
```

### ● RegExp 对象

RegExp 对象是一个预定义了属性和方法的正则表达式对象。语法如下：

```
var pat=new RegExp(pattern,modifiers);
```

等价于 var pat = /pattern/modifiers;

其中 pattern（模式）描述表达式的字符串。

modifiers(修饰符) 用于指定全局匹配、区分大小写的匹配和多行匹配的字符。

RegExp 对象的方法

方法	描述
exec	检索字符串中指定的值。返回找到的值，并确定其位置。
test	检索字符串中指定的值。返回 true 或 false。
toString	返回正则表达式的字符串。

实例：

```
var str = 'cctv';  
var patt1 = new RegExp('v', 'i'); //  
patt1.test(str) //输出 true  
var patt2=new RegExp("e");  
print(patt2.exec("The best things in life are free")); //输出 e
```

## 3.10 通信

### 3.10.1 TCP 通信发送

可将数据发送到 TCP 通信设备。通信设备在菜单栏“通信”->“通信设置”中添加，详见[通信设备管理器](#)章节。



在“选择通信设备”中选取发送数据的通信设备。在“数据发送组装表”中添加要发送的数据类型并绑定要发送的变量，通过分割符组装要发送的数据，并可设置组装后的数据是

否添加结束符。工具执行时，“组装后发送的数据”栏将显示当前发送的数据，数据仅在当前窗口打开时刷新显示。

### 3.10.2 TCP 通信接收

TCP 通信接收可从通信设备中获取数据。。通信设备在菜单栏“通信”->“通信设置”中添加。



在“选择通信设备”中选取接收数据的通信设备。需在“数据解析表”中添加接收数据的变量及类型，通过分隔符解析接收到的数据。在“接收模式选择”中配置是否等待数据或匹配等待数据。当选择不等待时，工具立即执行；选择无数据等待时，工具将阻塞在此处，直到收到数据才继续执行流程中的后续工具；选择不匹配等待时，可在数据解析表中勾选需要匹配的数据值，收到的数据与匹配的数据值相等时才会继续执行流程中的后续工具。

### **3.10.3 Modbus 写入**

可将数据发送到 Modbus 通信设备。通信设备在菜单栏“通信”->“通信设置”中添加。本平台软件作为主站，使用 Modbus 协议，包括 ModbusTCP 和 ModbusRTU，将数据写入从站的寄存器。



- 起始地址：功能码对应的寄存器区的首地址。
- 功能类型：modbus 功能码，支持读线圈、读输入、读保持寄存器、读输入寄存器。
- 数据类型：指定写入的单个数据类型长度
- 大小端格式：数据高低字节对齐方式。

在“选择通信设备”中选取发送数据的 Modbus 通信设备。在“数据发送组装表”中添加要发送的数据类型并绑定要发送的变量（也可在数据值列输入固定值发送）。工具执行时，当前值列将显示本次发送的数据。

### 3.10.4 Modbus 读取

可从 Modbus 通信设备中获取数据。通信设备在菜单栏“通信”->“通信设置”中添加。本平台软件作为主站，使用 Modbus 协议，包括 ModbusTCP 和 ModbusRTU，接收从站寄存器的数据。



选择通信设备: ModbusTCP - ModbusTCP [UID=0]

起始地址: 0

功能类型: 01 线圈状态(0x)

数据类型: 16位整形

大小端格式: CDAB

读取模式选择

无条件执行，不等待

接收数据，无数据则等待

接收指定命令，不匹配则等待

数据解析表:

序号	字段名称	数据类型	解析值	匹配值
1	output1	bool	true	<input type="checkbox"/> false
2	output2	bool	false	<input type="checkbox"/> false

添加 删除 取消 确定

- 起始地址：功能码对应的寄存器区的首地址。
- 功能类型：modbus 功能码，支持读线圈、读输入、读保持寄存器、读输入寄存器。

- 数据类型：指定读取的单个数据类型长度
- 大小端格式：数据高低字节对齐方式。

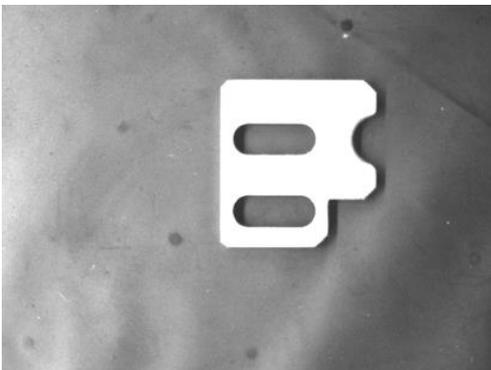
在“选择通信设备”中选取接收数据的 Modbus 通信设备。需在“数据解析表中”添加接收数据的变量名称及类型。在“读取模式选择”中配置是否等待数据或匹配等待数据。当选择不等待时，工具立即执行；选择无数据等待时，工具将阻塞在此处，直到收到数据才继续执行流程中的后续工具；选择不匹配等待时，可在数据解析表中的匹配值列勾选需要匹配的数据值，收到的数据与匹配的数据值相等时才会继续执行流程中的后续工具。

## 4. 案例搭建

### 4.1 零件定位检测

#### 4.1.1 案例目标

测量如下零件的中心点位置。



#### 4.1.2 方案搭建

整体思路：

- 通过轮廓匹配查找零件在图像中的位置。
- 查找零件最外围四条直线边：左直线，上直线，右直线，下直线。
- 测量相邻两直线边的交点：左上交点，右上交点，右下交点，左下交点。
- 根据交点生成两条对角线：左上-右下交点生成一条直线，右上-左下交点生成一条直线。

- 计算两条对角线的交点，即为零件的中心。

搭建步骤：

### Step1: 新建工程

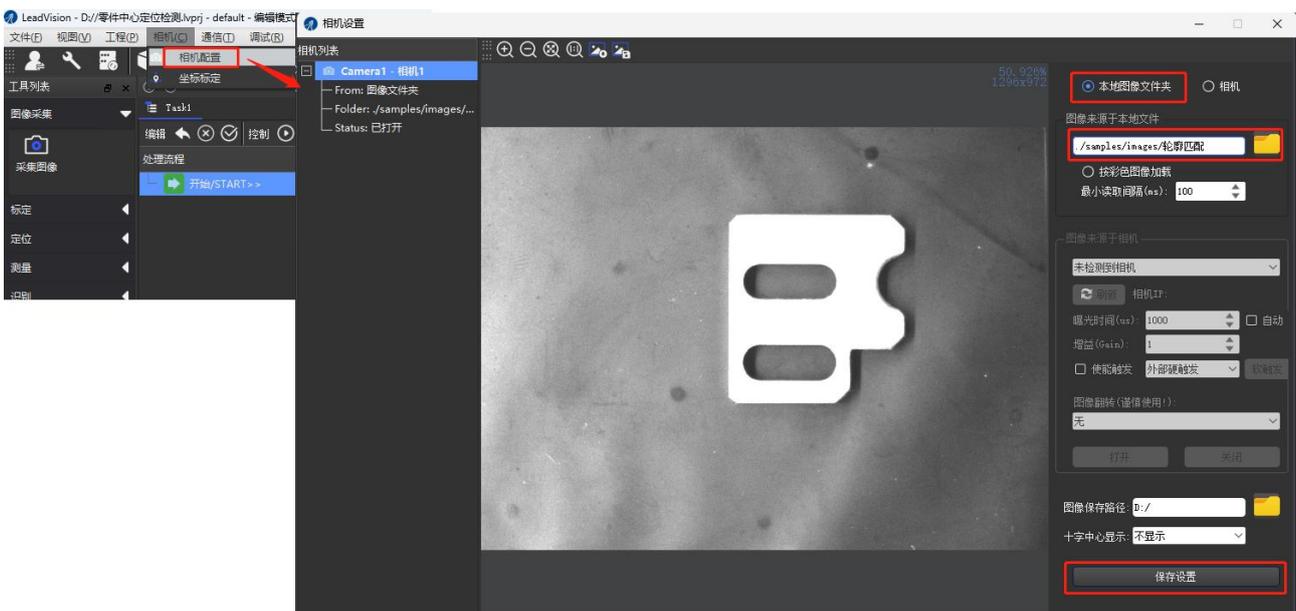
在菜单栏选择“工程”—》“新建工程”，弹出新建工程对话框，输入相机数、任务流程数、工程名称和保存路径，如下图所示。



### Step2: 相机配置

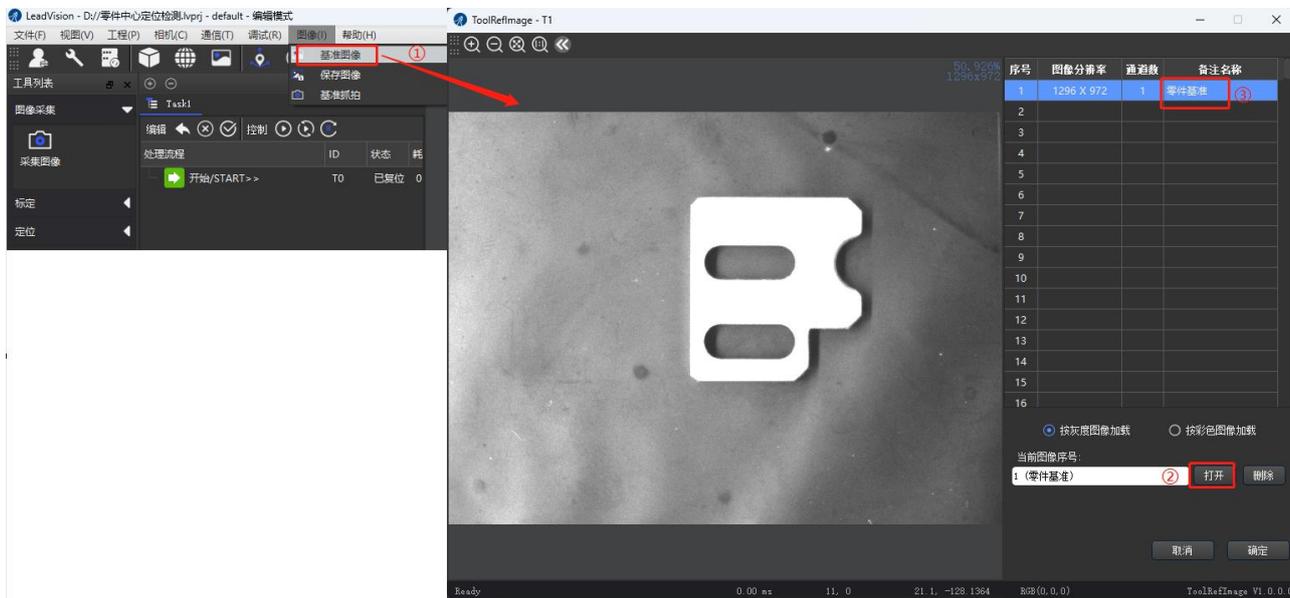
在菜单栏选择“相机”—》“相机配置”，弹出相机设置对话框，本例图像来源为本地图像文件夹(如果实际连接了相机，此处勾选相机)；图像来源路径中输入相对路径(./samples/images/轮廓匹配)，当然也可以通过打开文件对话框，选择图片存放的绝对路径；其他参数按默认配置即可，最后单击“保存设置”，如下图所示。

注意：图像来源于路径为相对路径时，输入完路径后，需按回车生效。



### Step3: 设置基准图像

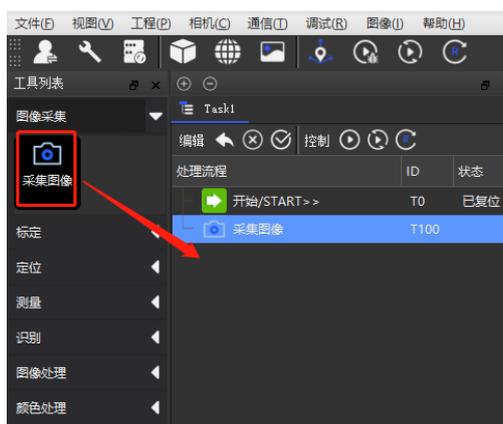
基准图像用于配置后续各工具参数使用，每新建一个工程都需要设置基准图像。在菜单栏选择“图像”——“基准图像”，弹出基准图像设置对话框。点击“打开”选择一张基准图像，然后修改基准图像的备注名称，便于个人识别。



### Step4: 拖放工具搭建流程

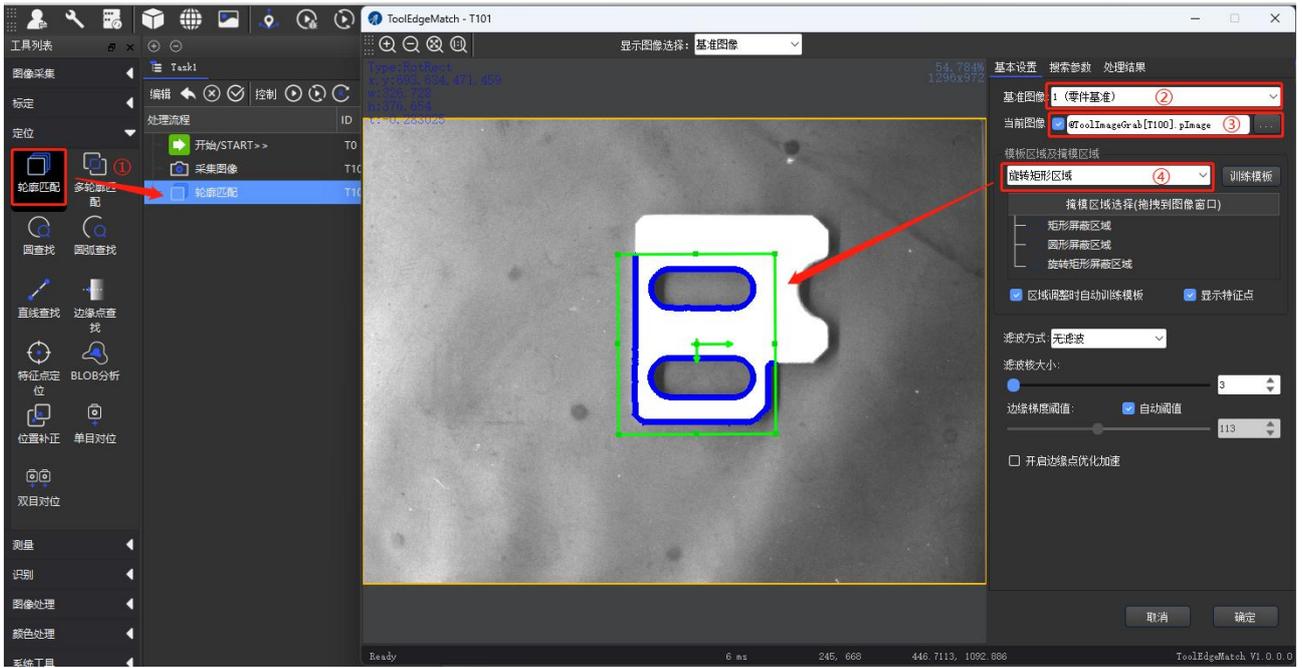
(1) 添加“图像采集”工具。

在“工具列表”中，鼠标左键点击“采集图像”，拖动到 Task1 的处理流程后松开鼠标左键，即实现该工具的加载。双击“图像采集”，弹出“图像采集”对话框，可配置该工具参数，该工具使用默认参数即可。

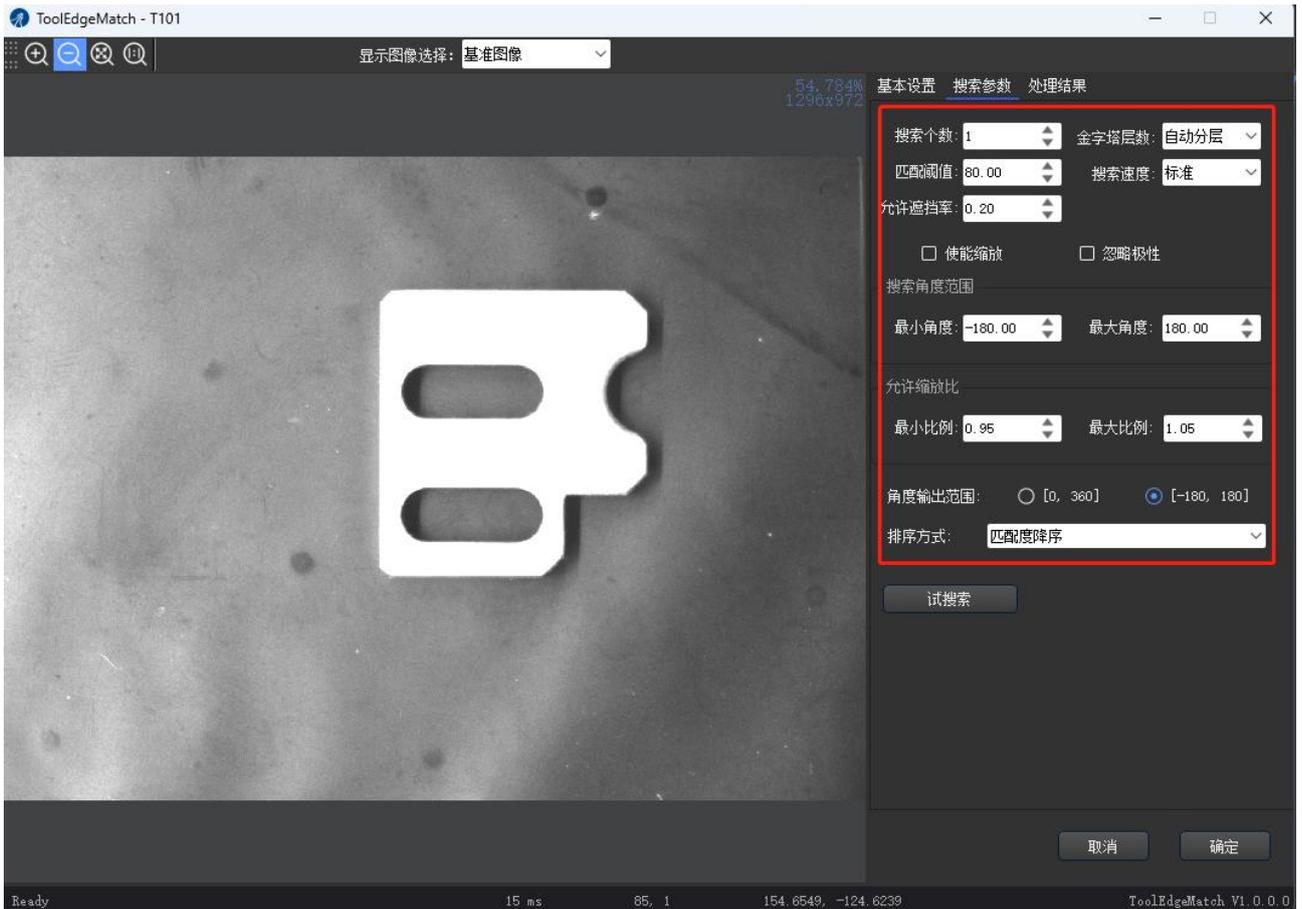


(2) 添加“轮廓匹配”工具。

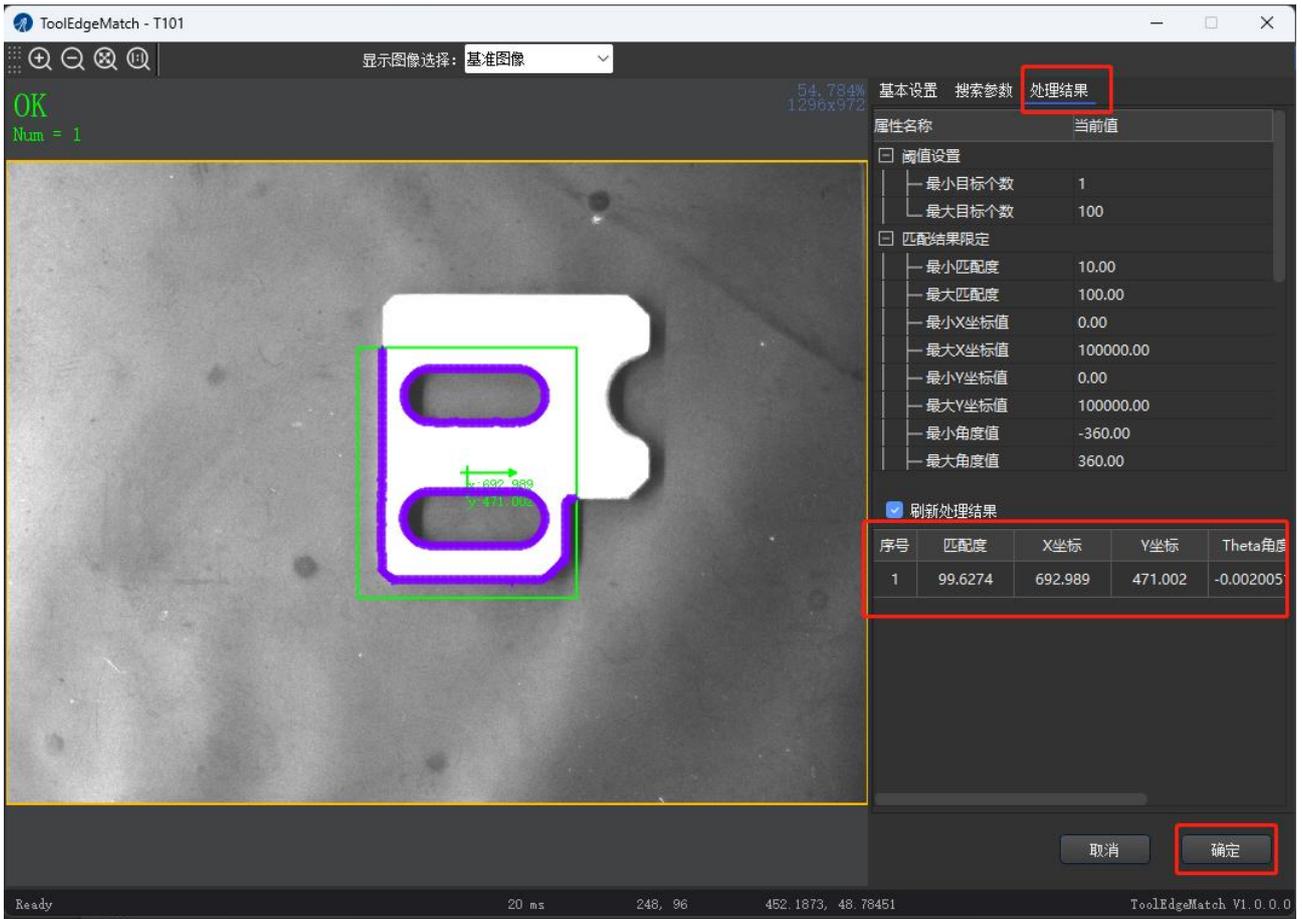
同上述操作，拖放“轮廓匹配”工具到处理流程中，双击配置该工具的基本设置。在基本设置中选择基准图像来训练模板区域，如下图所示操作步骤。



切换到搜索参数页面，设置查找模板参数，如下所示。

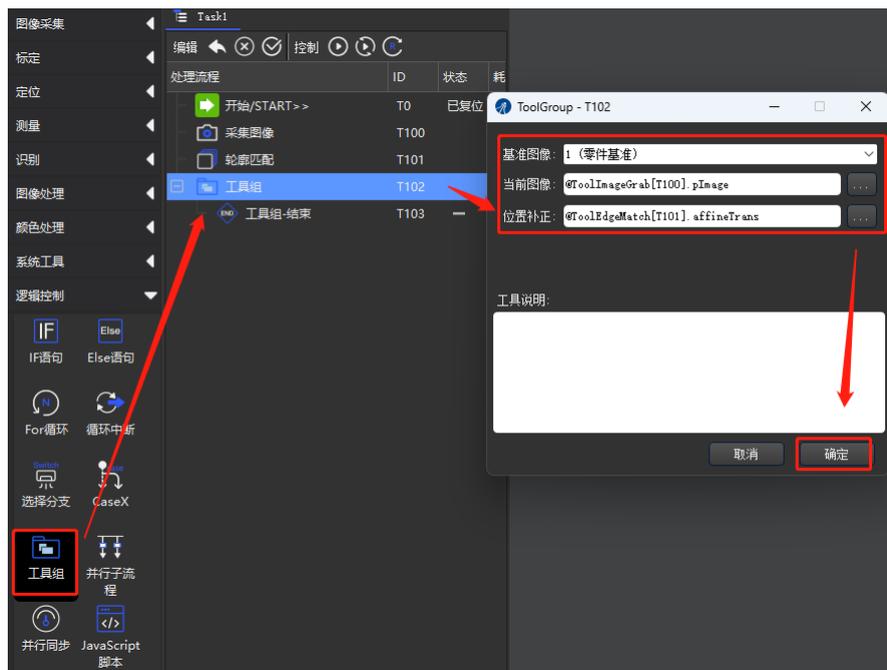


最后切换到处理结果页面，确认能正常找到模板，然后点击“确认”，保存轮廓匹配参数。至此，该工具配置完成。



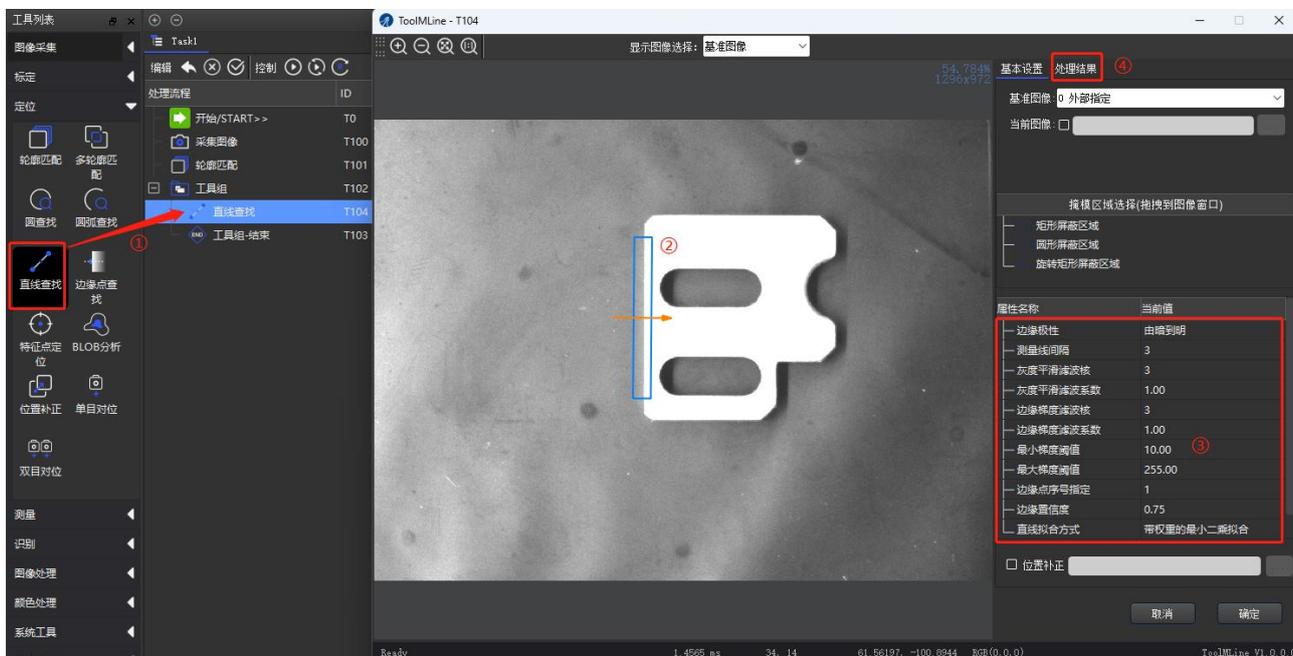
### (3) 添加“工具组”

同上述操作，拖放“工具组”工具到处理流程中，双击配置该工具参数。此处“工具组”用于统一配置后续工具的基准图像、当前图像、位置补正等公共参数，避免多次重复操作。

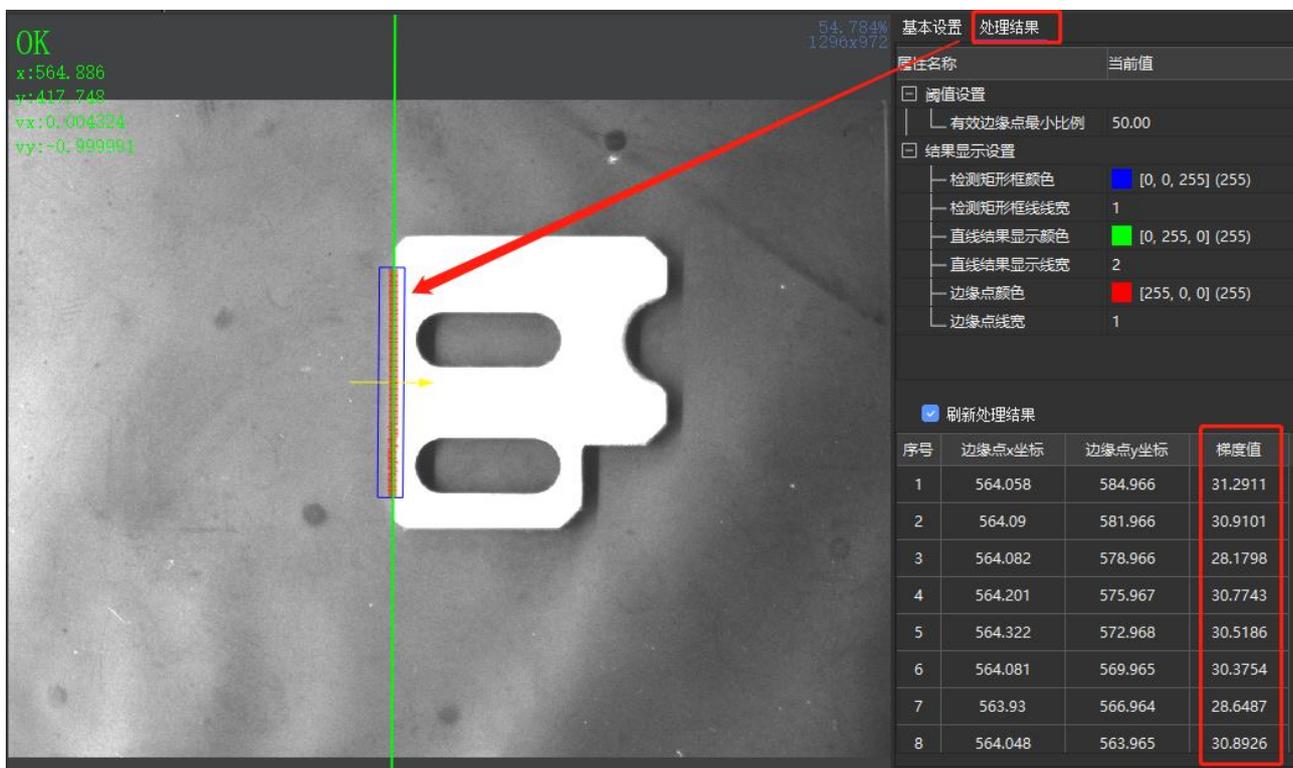


#### (4) 添加“直线查找”工具

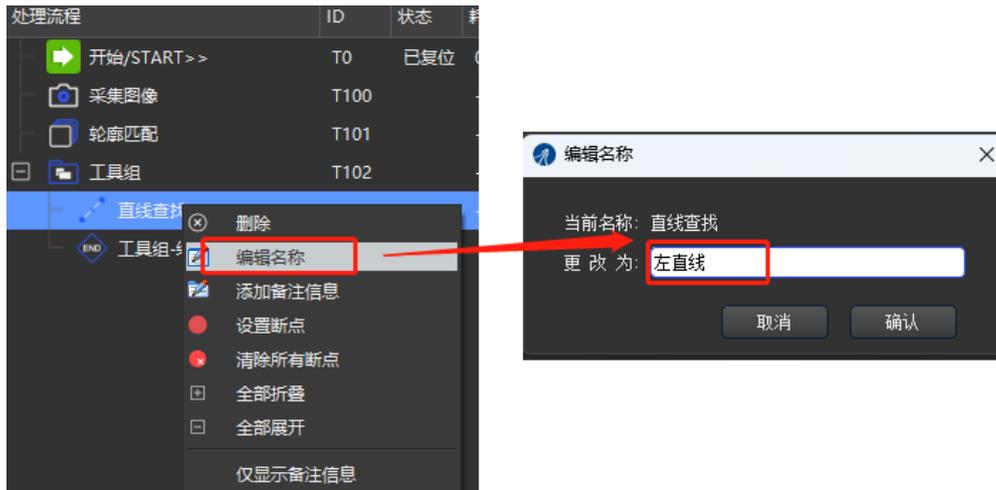
拖动“直线查找”工具到工具组中，双击配置该工具参数。此处查找左直线，在“基本设置”页中，拖动 ROI 到左侧边缘，根据箭头方向设置边缘极性为由暗到明，其他参数使用默认值。



最后点击“处理结果”页面，确认直线能否正确找到。如果未能找到，则返回“基本设置”页面修改梯度值、滤波核等参数，再切回“处理结果”页面确认。



修改“直线查找”工具名称为“左直线”。

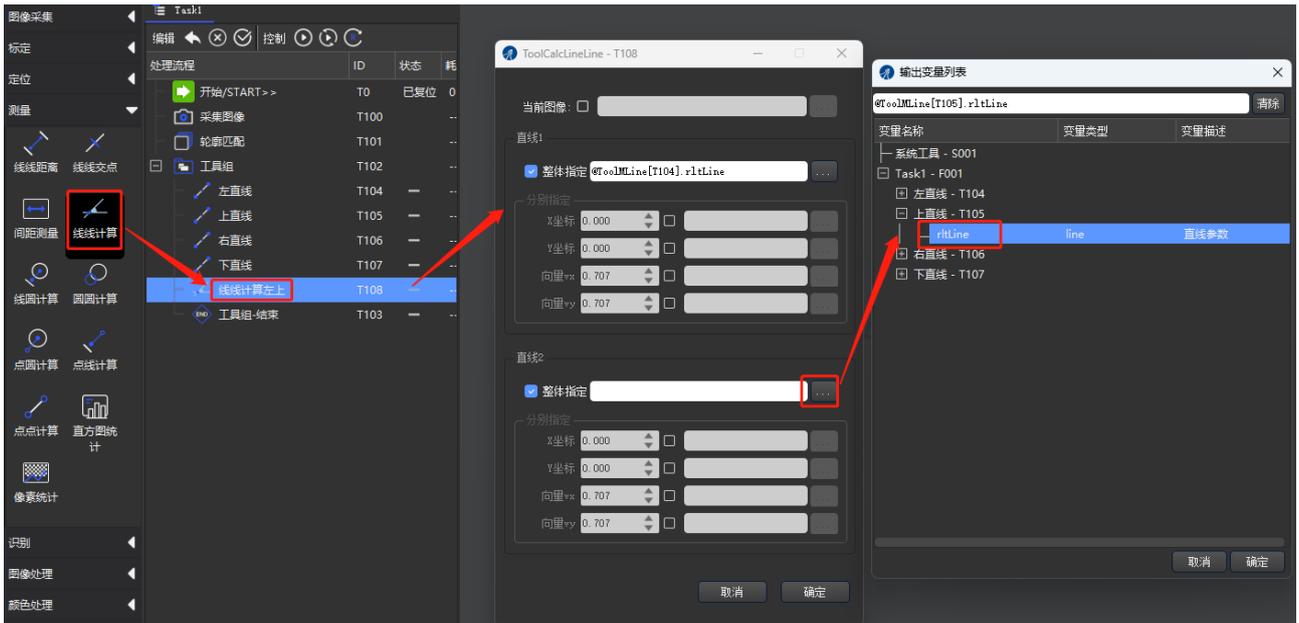


依上述相同配置方式，再添加 3 个“直线查找”工具并修改名称为“上直线”、“右直线”、“下直线”，最终效果如下：

处理流程	ID	状态	耗
开始/START>>	T0	已复位	0
采集图像	T100		--
轮廓匹配	T101		--
工具组	T102		--
左直线	T104	—	--
上直线	T105	—	--
右直线	T106	—	--
下直线	T107	—	--
工具组-结束	T103	—	--

#### (5) 添加“线线计算”工具

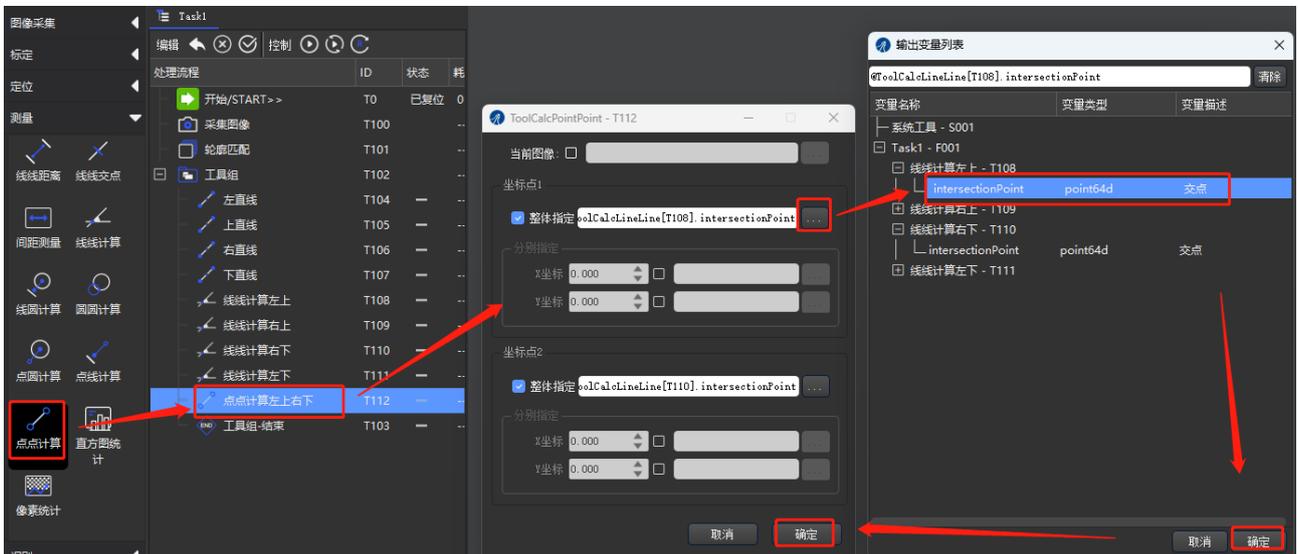
拖放“线线计算”工具到工具组中，并修改名称为“线线计算左上”。按下图选择“线线计算”的两条直线，分别为“左直线”和“上直线”。



同理，再添加 3 个“线线计算”工具，名称修改为“线线计算右上”、“线线计算右下”、“线线计算左下”。

#### (6) 添加“点点计算”工具

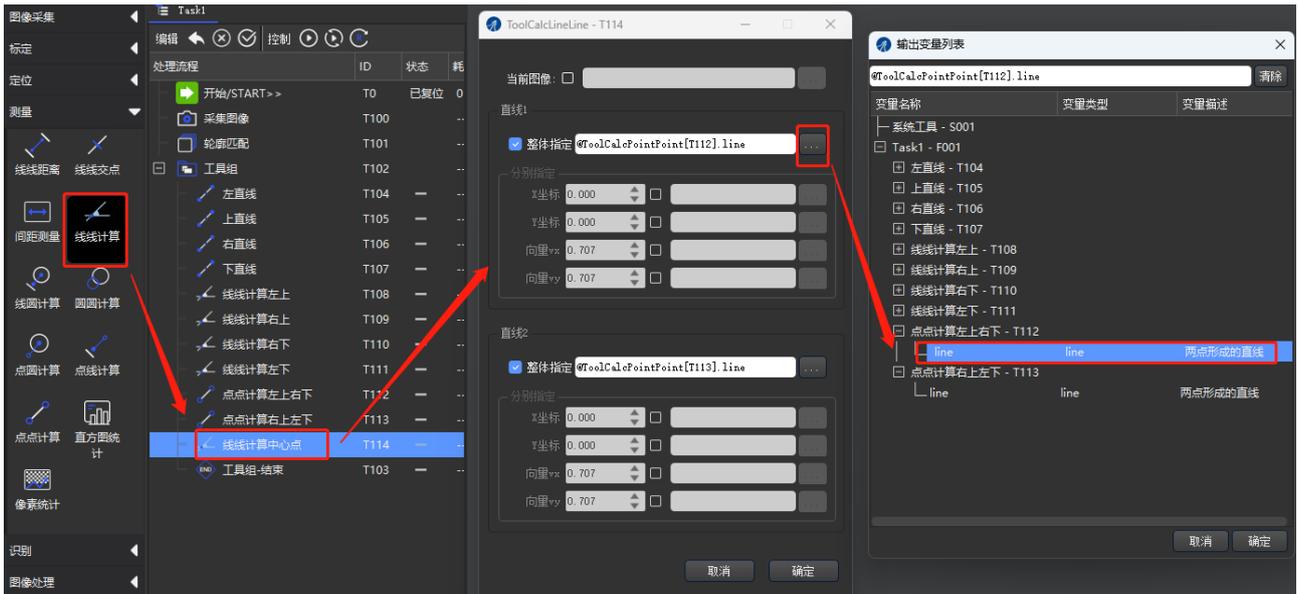
拖动“点点计算”工具到工具组中，并修改名称为“点点计算左上右下”。设置坐标点 1 为“点点计算左上”，坐标点 2 为“点点计算右下”，如下图所示。



同理，再添加一个“点点计算”工具，并修改名称为“点点计算右上左下”。

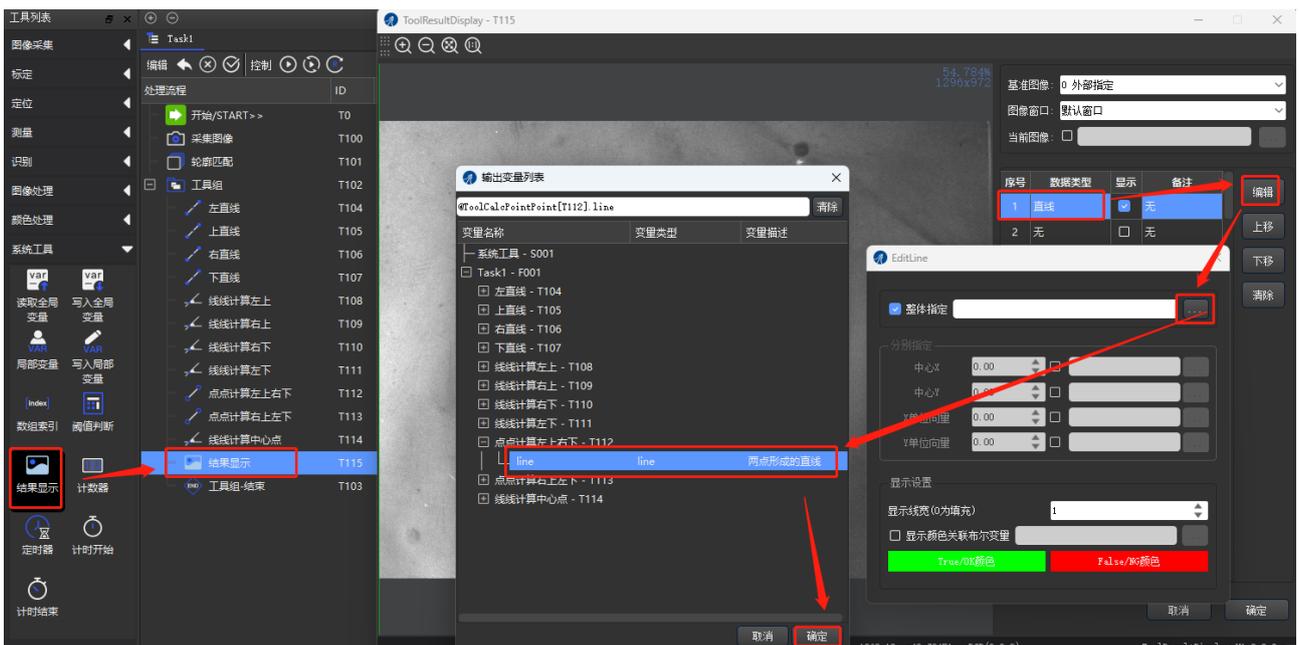
#### (7) 添加“线线计算”工具

拖动“线线计算”工具到工具组中，并修改名称为“线线计算中心点”。直线 1 选择“点点计算左上右下”，直线 2 选择“点点计算右上左下”，如下图所示。

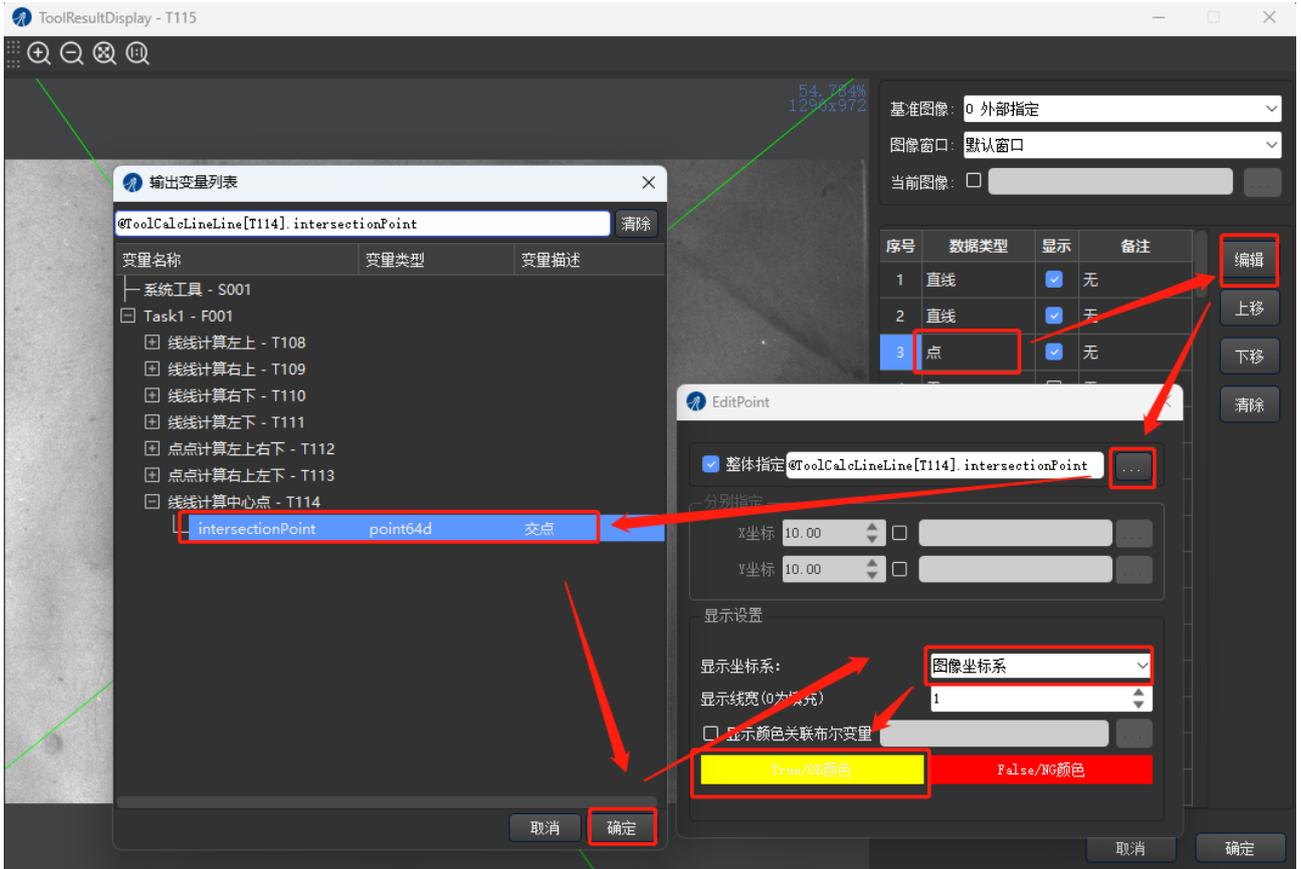


### (8) 添加“结果显示”工具

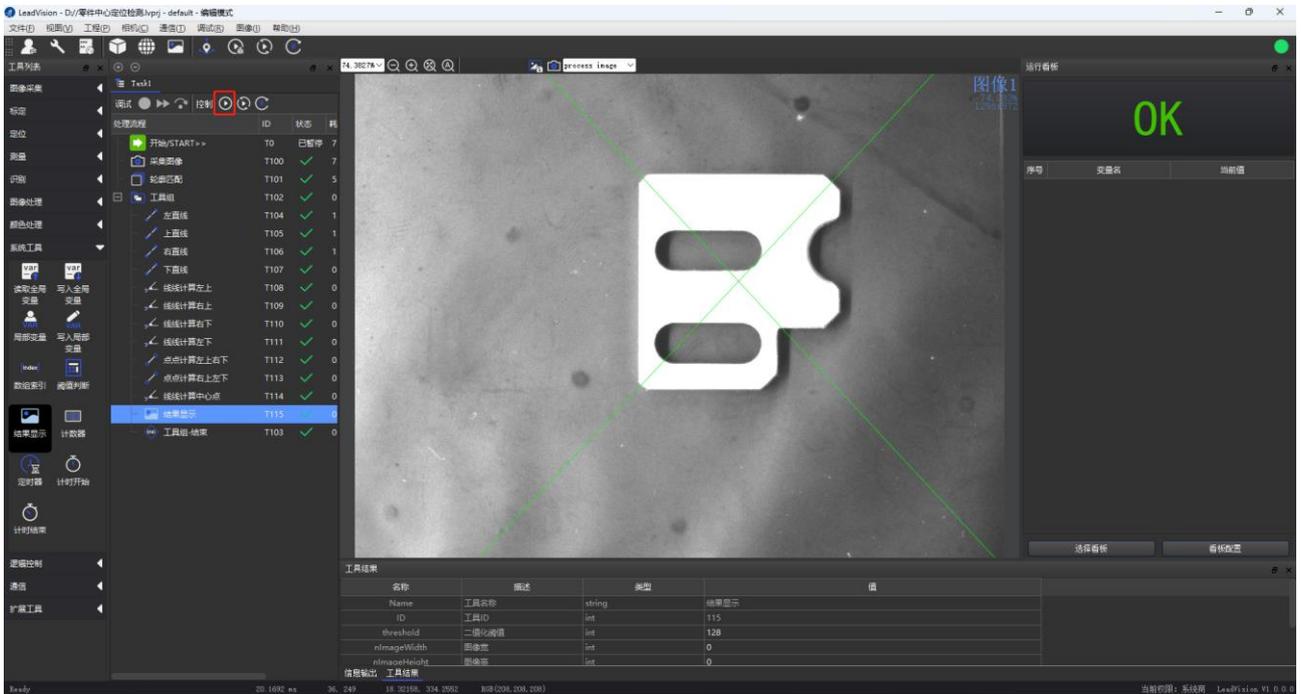
拖放“线线计算”工具到工具组中，用于显示最终找到的对角线及中心点。左上右下对角线配置如下所示，同理配置右上左下对角线。



中心点配置如下，“点”显示需选择“图像坐标系”，显示颜色按需设置，此处配置为黄色。至此方案流程搭建完成。



单次运行“run once”按钮，试运行流程，最终效果如下。





客户咨询中心  
目录索取·技术咨询·产品解惑

400-885-5521 销售热线  
400-885-5501 技术热线

更多最新的雷赛资讯, 请扫码关注!



公众号



视频号

# 成就客户 共创共赢

## 深圳市雷赛智能控制股份有限公司 China Leadshine Technology Co., Ltd.

深圳市南山区沙河西路3157号南山智谷产业园B栋15-20层  
邮编: 518052  
电话: 400-885-5521 传真: 0755-26402718  
网址: www.leisai.com E-Mail: marketing@leisai.com

上海分公司  
上海市嘉定区金园五路601号

山东办事处  
济南市天桥区滨河商务中心D座2003室

合肥办事处  
合肥市蜀山区潜山路与高河东路交口绿地蓝海大厦A座1209室

温州办事处  
浙江省温州市瓯海区潘桥街道宁波路阳光城愉景嘉园8幢2604

杭州办事处  
浙江省杭州市余杭区瓶窑镇桂花溪园(南区)2幢1单元402

北京办事处  
北京市大兴区绿地启航国际3号楼1109室

苏州办事处  
江苏省苏州市苏州工业园区金尚路1号仙峰大厦南楼7层

武汉办事处  
湖北省武汉市东湖新技术开发区长城园路2号海贝孵化器209

青岛办事处  
山东省青岛市城阳区金日紫都小区12号楼1单元301室

广州办事处  
广州市番禺区汉溪大道280号时代E-park A3栋1707单元