# VISIONGO智能视觉平台软件

使用手册







版	本记录	4
第·	一章 VISIONGO简介	5
	1.1 运行环境	5
	1.2 软件开发界面(IVP)	5
	1.3 支持的数据类型	6
第	二章 创建首个VISIONGO项目	11
	2.1 新建项目	11
	2.1.1 项目文件名称的命名格式	12
	2.1.2 修改相机IP地址	12
	2.2 相机设置	13
	2.2.1 触发设置	13
	2.2.2 曝光和增益设置	14
	2.2.3 TCP/IP包大小与巨型帧	14
	<i>2.2.3.1</i> 巨型帧的定义	15
	<i>2.2.3.2</i> 巨型帧设置(Windows)	16
	2.2.4 ROI设置	17
	2.2.5 白平衡	17
	2.3 添加产品定位	17
	2.3.1 模版定位	18
	<i>2.3.1.1</i> 训练模板	19
	<i>2.3.1.2</i> 查找模板	22
	2.3.2 线线定位	23
	2.3.3 斑点定位	26
	2.3.4 圆心定位	30
	2.3.5 线定位	32
	2.4 添加基准线	34
	2.4.1 通过查找边界	35
	2.4.2 通过两圆圆心的连线	35
	2.5 添加函数模块	36
	2.5.1 函数模块的排序	38
	2.5.2 函数模块的复制	38
	2.6 添加/删除IVU可显示参数	39
	2.7 添加/删除IVU可设置参数	40
	2.8 添加数字输入输出(DIO)	40
	2.9 添加串口输出	41
	2.9.1 标准的通讯协议	42
	2.9.2 Modbus通讯协议	43
	2.10 添加网口输出	43
	2.11 添加图像保存	44
	2.11.1 本地存储	44
	2.11.2 FTP存储	45
	2.12 项目保存/项目另存	45
	2.13 IVU中运行项目	46
	2.13.1 IVU主界面介绍	47
	2.13.2 多相机的支持	48
	2.13.3 自动切换任务	49



	50
3.1 相机标定	
3.1.1 圆点标定	50
3.1.2 简易标定	51
3.2 手眼标定	52
3.2.1 眼随手动/眼手分离(眼朝上)	52
3.2.2 手眼分离(眼朝下)	
第四章 检测模块详解	59
4.1 模版查找	59
4.2 斑点分析	61
4.3 亮度检测	64
4.4 灰度检测	65
4.5 对比度检测	66
4.6 颜色检测	66
4.7 条码识别	68
4.8 二维码识别	70
4.9 单边测量	71
4.10 对边测量	73
4.11 圆心测量	74
4.12 同心度测量	76
4.13 极点测量	78
4.14 极距测量	79
4.15 圆弧测量	81
4.16 阵列测量	82
4.17 图像比对	83
4.18 边缘检测	84
第五章 脚本(Script)编程	85
5.1 脚本编程界面介绍	85
5.2 基本的LUA函数	85
5.2.1 LUA的数学函数库(math)	85
5.2.2 LUA的字符串函数库(string)	
5.2.3 LUA的条件函数(If)	90
5.2.4 LUA的循环函数(for/while)	92
5.3 自定义的关键词及函数	93
5.3.1 项目及函数操作	93
5.3.2 变量操作	94
5.3.3 DI/O操作(Dio)	96
5.3.4 串口操作(Uart)	97
5.3.5 网口操作(Tcp)	98
5.3.6 脚本输入输出操作(Script)	





## 版本记录

版本	时间		描述
1.0	09/2017	初始版本	



## 第一章 VISIONGO简介

## 1.1 运行环境

- Windows XP / Vista / 7 / 8.x / 10
- Linux 64-bit

## 1.2 软件开发界面(IVP)

⟨> IVP免费版		11 ×
확퇴		
□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □	🗾 🧣 🏕 🗢 i 表示: 手服标准 网络 视觉医 幕阳 关于我们	
「 「 副 意思 の achine-vision.com.co		输出
		[1] 轮开
	a man and a second s	[2] 淡里 31
	T THE WAR AND THE TANK OF T	[4] ⊨ č
2017视谷图像深圳展会	and the second	[5] 문 방
完美落墓城川		[6] 蜀茂 [7] 半老
		[8] 半旁
视谷图像止在逐步融入我们的		[9] 水边比
生活		[10] 当度
		11 8161
	المتعادية ا	
图 0hm图片和文本性	- or the sta	
and the Line of dots	9-0.16 <sup>36</sup>	
中外同行		
正在热情地交流		
开始检测 在线	R示 Uest lit v 保存控象 ⊠ 憲线 元.120個年	
<	, , , , , , , , , , , , , , , , , , ,	
1.相机设置 2. 论测模块 3. 取坏填程 4. 前字输入输出 5. 用口通口	6. 网 通讯(109/09) // 国際存储	
☑ 建振系统 模板定立 ✓ 数据	編号   名称   英型   世 [13] 通过正知彩色选择成点	0; 0
□水平基准 資付管核決界(H) · 影后	000 FXN1 湖点分析 @ [14] 通过斜印米达选择版页	0; 0
□ <b>垂白長准</b> 通过直核边势(V) × 激音	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□	5, 0 5, 0
外帝可误置答数 外部可靠示参数	@ [17] 通过与度选择润点	0; 0
序号 函数名称 输入口名称	[18] 通过Y值进行排序	True
	[19]通过洋考运抨测点 [20] 店在2	False
	[21] 現合	False
+/-	新住 持刀 明除 国 [22] 原位社営堂	0; 0
连接到访真相机(IGGIP) CAM1_TSK001_1	51, 51, 51 Grab: 173.82 Inspect: 490.92	数字10 😝 馬口 😝 岡口 😝 디ヤ 😝



编号	功能划分	说明
AIX	工具栏	<ul> <li>a. 新建,新建一个项目;</li> <li>b. 打开,从文件夹中载入一个项目;</li> <li>c. 保存,保存当前项目;</li> <li>d. 另存为,保存当前项目为不同的文件;</li> <li>e. 模块显示,显示所有检测模块/显示当前检测模块;</li> <li>f. 工具框显示,是否显示检测模块的检测框;</li> <li>g. 结果显示,是否显示检测模块的检测结果到图像上;</li> <li>h. 相机标定,标定相机畸变,含简易标定、圆点标定,棋盘格标定;</li> <li>i. 手眼标定,标定相机与机械手臂配合,把相机的坐标转换到机械手的坐标;</li> <li>j. 网站,进入视谷网站www.machine-vision.com.cn;</li> <li>k. 视觉圈,进入视觉圈论坛;</li> <li>l. 帮助,打开帮助文档;</li> <li>m.关于我们,显示产品的主要信息;</li> </ul>
ΒX	帮助	显示公司最新资讯和软件使用帮助
C区	图像显示区	显示图像和相关数据
D区	控制与测试区	相关控制按钮和离线图像预览区,点击离线图像可以支持一 次基于当前选中图像的检测
ΕX	函数输出区	显示检测模块的输出结果
F区	参数设置区	参数设置

## 1.3 支持的数据类型

序号	类型	』   缩写	说明	定义	
1	Bool	В	布尔数据		
2	Long	I	整型数据		
3	Real	R	浮点型数据		
4	Enum	Е	枚举体		
5	String	STR	字符串		
6	RealPoint	Ρ	点	typedef struct _ivRealPoint { ivReal x; ivReal y; } ivRealPoint;	



序号	类型	缩写	说明	定义
7	Line	L	线	typedef struct _ivLine { ivReal x; ivReal y; ivReal length; ivReal angle; ivBool arrowFlag; char reserved[3]; }ivLine;
8	Rect1	RC1	正矩形,没有角度; 不可旋转,因此不能 跟随坐标系。	<pre>typedef struct _ivRect1 {     ivInt x;     ivInt y;     ivInt width;     ivInt height; } ivRect1;</pre>
9	Rect2	RC2	斜矩形,带角度,可 旋转	<pre>typedef struct _ivRect2 {     ivReal x;     ivReal y;     ivReal len1;     ivReal len2;     ivReal angle; } ivRect2;</pre>
10	Circle	CIR	圆	<pre>typedef struct _ivCircle {     ivReal x;     ivReal y;     ivReal radius; } ivCircle;</pre>
11	Ring	RIN	非封闭的圆环,方向为 从内到外或者从外到 内。	typedef struct _ivRing {



序号	类型	缩写	说明	定义
12	Arc	ARC	非封闭的圆环,方向为 逆时针或者顺时针。	typedef struct _ivArc { ivReal x; ivReal y; ivReal radius1; ivReal radius2; ivReal angleStart; ivReal angleExtent; ivInt radiusStep; ivBool clockwiseFlag; char reserved[3]; }ivArc;
13	Array	ARR	数组	<pre>typedef struct _ivArray {     ivVarType specType;     unsigned int nRows;     unsigned int nCols;     void *pVals;     char name[NAME_MAXLEN]; } ivArray;</pre>
14	Contour	CTR	轮廓	<pre>typedef struct _ivBlob {     ivRect2 rect2;     ivRealPoint baryPt;     ivRect1 rect;     unsigned int *pBits;     unsigned int area;     unsigned short label;     unsigned short realWidth;     unsigned short nChildBlobs;     char isValid_rect2;     char ivValid_baryPt; } ivBlob; typedef struct _ivContour {     ivBlob*restrict*restrict ppBlobs;     unsigned short imgWidth;     unsigned short imgHeight;     unsigned short nBlobs;     int *ext_pBinBuf;     char *ext_pbw_startCol;     short *ext_pbw_row;     short ext_idx;     char reserved[2]; }ivContour; </pre>



序号	类型	缩写	说明	定义
15	CoordSys	CRD	坐标系统	<pre>typedef enum _ivCoordType {     CT_X = 1,     CT_Y,     CT_XY,     CT_XYA }ivCoordType; typedef struct _ivCoordSys {     ivReal baseX, baseY;     ivReal baseAx;     ivReal baseAy;     ivCoordType type; }ivCoordSys;</pre>
16	Action	CMD	按钮	
17	Path	DIR	路径	
18	Tolerance	TOR	公差	typedef struct _ivTolerance { ivReal ILimit; ivReal uLimit; }ivTolerance;
19	Rect2Group	GRP	矩形阵列	<pre>typedef struct _ivRect2Group {     ivReal x;     ivReal y;     ivReal len1;     ivReal len2;     ivReal angle;     ivReal roi_pitch;     ivInt roi_count;     ivBool verticalFlag;     char reserved[3]; }ivRect2Group;</pre>



序号	类型	缩写	说明	定义
20	CompositeRoi	COP	复合检测框	<pre>typedef enum _ivCompositeType {     ECT_Rect2 = 0,     ECT_Circle,     ECT_Ring }ivCompositeType; typedef struct _ivCompositeRoi {     ivCompositeType type;     ivReal x;     ivReal y;     ivReal len1;     ivReal len2;     ivReal angle;     ivReal radius;     ivReal radius1;     ivReal angleStart;     ivReal angleExtent; }ivCompositeRoi;</pre>
21	CoupleRect2	COR	组合矩形框	<pre>typedef struct _ivCoupleRect2 {     ivReal x;     ivReal y;     ivReal len1;     ivReal len2;     ivReal angle;     ivReal len1_offset1;     ivReal len1_offset2;     ivReal x1;     ivReal y1;     ivReal y2;     ivReal y2;     ivBool in2OutFlag;     char reserved[3]; }ivCoupleRect2;</pre>
22	XYA	XYA	位置及角度	typedef struct _ivXYA {
23	COLOR	CLR	颜色	typedef struct _ivColor { ivReal r; ivReal g; ivReal b; }ivColor;



## 第二章 创建首个VISIONGO项目

### 2.1 新建项目

软件主界面点击[新建]按钮,打开如下对话框,需要说明的是,不同版本的 VISIONGO软件支持的相机类型不一样,因此下图中B区的列表名称也是不一样 的。

<⊘ 新建	조 지 뛰 쉬	s <b>+cz +n</b>	×
	日本 小城手抓取定 产品计数 / 高	5 <b>1日 17</b> 6 位及引导 / 读码 精度测量	3
相机	相机1		-
任务编号	1 Δ	X	· 🗸
专有名称	test		$\checkmark$
文件名称	CAM1_TSK001_tes	t.igo	
GigEVisio n	Balser B	X	
序号 型	묵	序列号	IP地址
L Da		X	192.108.1.21;
			7
刷新			
			确定

步骤:

1. 在A区的参数中确定新项目的名称,如果前面三个选项设置OK,第四个 选项(项目文件名称)会自动生成。项目文件名称的命名格式请参考<u>章节</u> <u>2.1.1</u>。值得注意的是,VISIONGO不会自动覆盖已存在的项目文件,因此如 果对应的项目文件已经存在的话,A区的名称验证标志将会显示错误,上图中 的√表示验证无误。

2. 在B区的范围内选择相机类型,大部分GigE相机可用GenlCam的方式打开,但也有部分相机不支持标准的GenlCam协议,因此要咨询工业相机厂家,或者查看相机说明书。如果不想连接相机,只需仿真测试,请跳过步骤3,直接进入步骤4。

3. 在C区选择要打开的相机,如果已经有相机连接到软件所在的PC,同时 C区又没有显示出来,请点击左下角的刷新按钮进行刷新。值得注意的是,在 C区双击相机会弹出IP修改界面,具体修改的方法请参考章节2.1.2。

4. 点击确认按钮进入主界面。

## 2.1.1 项目文件名称的命名格式

项目文件名称的命名格式CAM?\_TSK???\_\*.igo(其中?代表一个数字,\* 代表任意个字符或者数字)。

序号	符号	说明
1	?	1~6的数字,代表的是相机的序号,VISIONGO最多支持同时打开6个相 机
2	???	001~999的数字,代表任务编号。自动切换任务时需要通过任务编号找 到对应的项目文件并切换。
3	*	自定义的备注信息,只能是数字或者字符,不支持中文。

### 2.1.2 修改相机IP地址

a. 双击相机列表中的某一个相机, 弹出如下对话框;

IP设置		×
┌接口信息―		
MAC地址	50:7b:9d:c3:a	a2:c4
IP地址	192.168.1.54	AX
子网掩码	255.255.255.	0
网关	0.0.0.0	
-相机信息		
MAC地址	3c:ef:8c:94:ba	a:a4
IP地址	192.168.1.22	3  B区
子网掩码	255.255.255.	0
网关	0.0.0.0	
	确定	取消



b. 修改B区的IP地址, IP地址需保证跟A区的IP地址处于同一网段(即IP地址的前3段数据需相同, 如本例的192.168.1);

- c. 点击确定按钮;
- d. 点击刷新按钮刷新相机列表查看;

#### 2.2 相机设置

#### 2.2.1 触发设置

触发方式		
◉无 〇.	上升沿 ○下降沿 ○ 软件	
- 触发信息 	0	<b></b>
触发图像数里	1	
帧率(fps)	1	
触发延时 <mark>(us)</mark>	0	
滤波等级 <mark>(us)</mark>	10	

a. **触发方式**,如上图所示,VISIONGO支持4种工作方式,分别是无触 发、上升沿触发、下降沿触发、软件触发。其中无触发也即自由运行状态, 程序将使用指定的帧率连续从相机中采集图像。上升沿和下降沿都是硬触发 模式,硬触发模式下外部触发信号直接接入到工业相机的触发输入口,无需 经过VISIONGO软件。最后一个为软件触发模式,在此工作状态时,外部硬 件(如PLC,机械手臂等)可给VISIONGO发送固定格式的软件触发指令

(一般为字符串), VISIONGO接收到软件触发指令后立即从相机中采集一 幅图像。

b. **触发指令**,这里是指软件触发模式下的触发指令,这里可以填写完整的 触发指令,也可以填写真正指令的一部分,如果是填写真实触发指令的一部 分,则必须填写前面的一部分。假设有一个应用场合,相机被安装在机械手 臂上,机械手臂移动相机在多个工位拍照,第一个工位的触发指令为 CAMTRIGGER#1,第二个工位的触发指令为CAMTRIGGER#2,第三个工



位的触发指令为CAMTRIGGER#3,则只要在这里输入CAMTRIGGER#即可。

c. **触发图像数量**,是在触发状态下,每次触发需要采集的图像数量,默认 情况下为1,即触发一次采集一幅图像,也可以设置大于1的整数,即触发一 次采集多幅,该功能经常配合帧率一起使用,可实现在每次接收到一次触发 信号后,以恒定的帧率持续拍摄一定数量的图像。

d. 帧率,设置相机的帧率。

e. **触发延时**,单位微秒,即收到外部触发后不立即采集图像,而是等待一段时间再采集。

f. **滤波等级**,单位微秒。当机械设备有电磁干扰时需要设置该参数,该参数的意义是能自动过滤掉一些干扰信号,从而提取有效的触发输入信号。

#### 2.2.2 曝光和增益设置

曝光时间和增益可用来调节图像的亮度。曝光时间越长,增益越大,图像 也越亮,反之,图像则越暗。

在运动状态下拍照时,需把曝光时间设置小,这主要是当曝光时间较长时, 会在图像上形成一定的拖影,速度越快拖影越严重。

曝光时间 <mark>(us)</mark>	8000	•
増益	0	•

增益的调节能增加图像的对比度,但是会带来较大的信号干扰,大增益的 图像上一般都是雪花点性质的干扰信号。增益一般在现场光照强度比较弱的时 候使用。

#### 2.2.3 TCP/IP包大小与巨型帧

TCP/IP包大小 1500	•
TCP/IP包大小   1500	-

默认情况下,以太网的MTU是1500字节。然而,如果PC支持巨型帧,为 提高网络传输效率,需修改包大小,以实现更高的传输速度。具体的包大小请 根据网卡驱动可支持的最大包大小设置。



2.2.3.1 巨型帧的定义

巨型帧(Jumbo Frame)是帧长大于1522字节的以太网帧。这是一种厂商标 准的超长帧格式,专门为千兆以太网而设计。巨型帧的长度各厂商有所不同, 从9000字节~64000字节不等。采用巨型帧能够令千兆以太网性能充分发挥, 使数据传输效率提高50%~100%。

巨型帧需要在相互通讯的2个通讯端口端上同时支持,而且与以前的以太网 产品不兼容,因此主要会应用于千兆主干的端口之间以及服务器端口接入到网 络主干的链路。交换机把巨型帧格式的数据转发向不兼容巨型帧的端口时应进 行帧格式的转换,即把巨型帧帧格式的数据转换成标准以太网的帧格式,从而 保证其正常工作。相反,从不兼容巨型帧的端口向支持巨型帧的端口转发数据 时,交换机可以把多个标准以太网帧合并成超长巨型帧帧,从而提高传输效率。 应该说巨型帧在一些领域里是非常有用的,它是有意设计为加速大文件传输服 务的。以太网标准定义的最大帧长度为1518字节,这样一个大的文件就需要被 切碎成为若干块,放到多个以太网帧中。而每个数据块传输的时候都会引入帧 头和尾的开销。倘若能够用一个大的帧完成文件的传输,则会减少很多帧的开 销,提高网络的利用率和传输速率。



#### 2.2.3.2 巨型帧设置(Windows)

a. 打开设备管理器

b. 网络设备器属性框

在设备管理器中找到对应的网 卡,然后右击选择属性,弹出右边 对话框,在高级页面的属性列表中 找到Jumbo Frame类似字眼的属 性,然后把值设置到最大。





#### 2.2.4 ROI设置

当我们关心的图像区域远小于图像大小时,可设置图像ROI。设置图像ROI 可提高图像的采集速度和传输速度。

- <mark>ROI</mark> 设置					
偏移量X	0	•	偏移量Y	0	•
宽度	0	* *	高度	0	▲ ▼

### 2.2.5 白平衡

设置彩色相机的白平衡参数,在连接黑白相机时为不可用状态,白平衡参 数可自动获取。

白平衡			
红色通道	1	*	
绿色通道	1	*	自动
蓝色通道	1	*	

### 2.3 添加产品定位

<b>1.</b> 相机设置	2. 检测模块	3. 脚本编程	4. 数字输入输出	<mark>5.</mark> 串	口通讯
🗹 坐标系统	模板定位			~	激活
					$\top$
	选择定位方	ī式			
	选择是否启	用坐标系统			
	把定位模块	设置为活动模	模块(界面显示的需要	문)	

我们从参数设置区的[2. 检测模块]可以找到产品定位的相关设置选项;产品 定位的相关设置较少,如上图所示只有3个部分;要使用产品定位的功能前,必 须先启动坐标系统。



那么,什么时候需要用到产品定位(即坐标系统)呢?产品定位,顾名思 义,就是通过一定的方法,准确地找出待测产品在图像中的位置。因此,当待 检测产品在图像中的位置不固定时都是需要使用产品定位的。VISIONGO十分 重视产品定位的作用,目前在VISIONGO中,一共集成了5种定位方式,分别是 模板定位、线线定位、斑点定位、圆心定位、线定位。各种定位方式的应用场 合和优缺点比较如下:

序号	定位方式	原理	优缺点
1	模板定位	通过在训练时,预先保存产品的全 部或者部分轮廓;正常检测时,则 在图像范围内查找相似的轮廓实现 产品的XYA定位。	优势:可适用任意角度 [-180°~180°]的产品定位。 劣势:速度较慢。
2	线线定位	通过使用两条边线的交点,和其中 一条边线的角度来实现对产品的 XYA定位。	优势:速度很快。 劣势:定位的角度方位大概为 [-45°~45°],且需要产品本身有 可使用的边缘。
3	斑点定位	使用Blob的重心,以及其最小外接 矩形的角度来实现对产品的XYA定 位	优势:速度较快。 劣势:定位的角度方位大概为 (0°~180°);Blob定位不太适合对 角度敏感的产品,一般可用于XY 定位,即实现X方向、Y方向等不 包含方向的定位
4	圆心定位	通过查找圆心的中心坐标来实现XY 定位	优势:速度很快 劣势:只能实现XY定位,不能实 现角度定位
5	线定位	通过查找产品的一条边缘实现X、Y 定位	优势:速度很快。 劣势:只能实现X、Y定位,不能 实现角度定位,不能实现X、Y同 时定位。

### 2.3.1 模版定位

1.相机设置	<b>2.</b> 检测模块	3. 脚本编程	4. 数字输入输出	<b>5.</b> ह	印通讯
🗹 坐标系统	模板定位			$\sim$	激活

首先,如上图所示,先启用坐标系统,并选择模板定位,然后点击激活, 此时软件界面如下:



序号	界面类型	界面图
1	图像界面	
2	输入参数界面	I双击训练模板轮廓]         Default           [2][E] 图像通道选择         Default           [3][C] 查找框设置         100; 100; 50; 50           [4][F] 起始角度         -45           [5][F] 终止角度         45           [6][F] 最小匹配分数         0.5           [7][F] 贪心算法         0.9           [8][E] 极性类型         Normal           [9][P] 点位补偿量         0; 0
3	输出参数界面	<ul> <li>輸出</li> <li>【1][CRD] 坐标系</li> <li>【2][C] 中心点和角</li> <li>【3][F] 匹配分数 0.000000</li> <li>【4][A] 轮廓点</li> </ul>

2.3.1.1 训练模板

双击函数输入列表中的第一个参数, 弹出如下模板训练对话框1,

<sup>1</sup>本对话框中的图像跟主界面的图像保持一致,因此每个人弹出的对话框中的图像是不一样的。



模板编辑器	>
图像通道 红色通道 人 🔀	
最大对比度 111	
最小对比度 55 ▲	
金字塔级别 6	
	L'Dodt
画笔大小 5 レビン	ipau
添加所有像素	
重置所有	
	123KJHFAZS
	1
371, 6	
	确定取消

如上图所示,模板编辑器的界面可分成3个区域

a. A区,模板创建区,可选择图像通道及设置相关参数,不过一般情况下除 图像通道外,不建议修改剩余的3个参数,因为剩余的3个参数软件会自动计 算最优值。

b. B区,模板修改区,要启用B区的功能需先选中编辑遮罩这个选项。所谓模板修改,即通过擦除不必要的轮廓以减少外部干扰。

c. C区,图像及轮廓显示区。

第一步:创建模板,如上图所示,通过在图像界面拖动ROI工具,使其刚 好覆盖模板区域,软件将自动获取最优模板和参数。模板参数以点集的形式绘 制到图像界面,最优参数也将更新到A区。

第二步:修改模板,即去除不需要的轮廓,如本例中我们去掉整个i的轮 廓。修改完成后,点击确定完成模板训练。

G	
---	--

模板编辑器	×
图像通道 红色通道 ~	^
最大对比度 111 ▲	
最小对比度 55 ▲	
金字塔级别 6	
添加所有像素 移除所有像素	
重置所有	- 15
	VIHFA2345
123	KITT
-29, 205	
-30, 230	~
	确定取消

a. 画笔大小,及用来擦除轮廓的画笔的大小;

b. 反转,功能反转,正常情况下我们是擦除已有的轮廓,但是当我们不小 心擦除得太多时候,就需要选择反转功能,反转之后可以把已经擦除的轮廓 重新补回来。

c. 添加所有像素,清除所有擦除痕迹。

d. 移除所有像素,清除所有模板痕迹。

e. 重置到模板图像和原始模板。



2.3.1.2 查找模板

其次,在图像上拖动查找框,使其能够覆盖模板可能出现的所有区域。



#### 表格 2-3-1-1 输入参数

序号	名称	参数说明
1	双击训练模板轮廓	双击之后进入模板训练界面开始训练模板轮廓
2	图像通道选择	选择颜色通道,当输入图像为彩色图像时,通过选择具有更高对比度的颜色通道,有助于更准确稳定地找到目标模板。 取值范围: • Default:默认通道,为CT_R。 • CT_R:红色通道 • CT_G:绿色通道 • CT_B:蓝色通道
3	查找框设置	查找框的位置和大小
4	起始角度	查找模版轮廓的起始角度
5	终止角度	查找模版轮廓的终止角度
6	最小匹配分数	默认值0.5。取值范围(0~1),值越大,检测速度越快。



7	贪心指数	默认值0.9,在找不到模版轮廓的情况下可适当增大至0.95左 右
8	极性类型	设置查找模版轮廓时的极性。 • Normal:无极性查找 • Ignore global polarity:忽略全局明暗变化 • Ignore global polarity:忽略局部明暗变化
9	点位补偿量	补偿值,模版的真实中心点加上补偿值再输出
10	输出坐标类型	选择颜色通道,当输入图像为彩色图像时,通过选择具有更高对比度的颜色通道,有助于更准确稳定地找到目标模板。 取值范围: • Default:默认通道,为CT_XYA。 • CT_X:红色通道 • CT_Y:绿色通道 • CT_XY:蓝色通道 • CT_XYA:蓝色通道

表格 2-3-1-2: 输出参数

序号	名称	参数说明
1	坐标系统	
2	中心点和角度	查找到的轮廓的中心点坐标以及相对于模版轮廓的角度
3	匹配分数	匹配分数
4	轮廓点	组成轮廓的所有点的坐标

### 2.3.2 线线定位

<b>1.</b> 相机设置	<b>2.</b> 检测模块	3. 脚本编程	4. 数字输入输出	<b>5.</b> a	串口通讯
☑ 坐标系统	线线定位			$\sim$	激活

首先,如上图所示,先启用坐标系统,并选择线线定位,然后点击激活, 此时软件界面如下:



#### 序号 界面类型

1

界面图



			[1][E] 图像通道选择	Default
2 输入参数界面		[2][E] 坐标系类型	Default	
	Ŧ	[3][C] 检测框1	100; 100; 90; 80; 0	
		[4][E] 查找模式1	Light to dark	
		[5][I] 梯度1	30	
	÷	[6][C] 检测框2	120; 120; 90; 80; 270	
			[7][E] 查找模式2	Light to dark
			[8][I] 梯度2	30

输出

3 输出参数界面

	[1][CRD] 坐标系统	
界面		
		l



表格2-3-2-1: 输入参数

序号	名称	参数说明
1	图像通道选择	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
2	检测框1	调整检测框1的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框1的颜色为红色)
3	边缘模式1	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘
4	灰度梯度1	设置检测框1查找边缘时的灰度梯度
5	检测框2	调整检测框2的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框2的颜色为绿色)
6	边缘模式2	设置检测框2的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘
7	灰度梯度2	设置检测框2查找边缘时的灰度梯度
8	输出坐标类型	选择颜色通道,当输入图像为彩色图像时,通过选择具有更高对比度的颜色通道,有助于更准确稳定地找到目标模板。 取值范围: • Default:默认通道,为CT_XYA。 • CT_X:红色通道 • CT_Y:绿色通道 • CT_XY:蓝色通道 • CT_XYA:蓝色通道

表格2-3-2-2:输出参数

序号	名称	参数说明
1	坐标系统	坐标系统,在其它函数中将自动应用



## 2.3.3 斑点定位

1.相机设置	2. 检测模块	3. 脚本编程	4. 数字输入输出	5.	串口通讯
2 坐标系统	Blob定位			~	激活

首先,如上图所示,先启用坐标系统,并选择斑点定位,然后点击激活, 此时软件界面如下:

序号 界面类型

界面图



1 图像界面



序号	界面类型		界面		
		+	[1][C] 工具框设置	Rect; 100; 100; 90; 80; 0	^
			[2][E] 图像通道选择	Default	
		Ŧ	[3][C] 二值化	128; 255	
			[4][I] 腐蚀系数	0	
			[5][I] 膨胀系数	0	
			[6][B] 填充1	False	
		÷	[7][C] 通过面积选择斑点	10; 99999	
		÷	[8][C] 通过中心点X选择斑点	0; 0	
	输入参数界面	÷	[9][C] 通过中心点Y选择斑点	0; 0	
		÷	[10][C]通过正矩形宽选择斑点	0; 0	~
		Ð	[11][C]通过正矩形高选择斑点	0; 0	
2		÷	[12][C] 通过斜矩形长边选择斑点	0; 0	
		Đ	[13][C] 通过斜矩形短边选择斑点	0; 0	
		Đ	[14][C] 通过斜矩形长宽比选择斑	0; 0	
		Đ	[15][C] 通过角度选择斑点	0; 0	
			[16][B] 通过Y值进行排序	True	
		÷	[17][C] 通过序号选择斑点	0; 0	
			[18][B] 填充2	False	
		T	[19][B] 联合	False	
	8	+	[20][P] 点位补偿里	0; 0	
			[21][E] 连通方式	CONN_4	
			[22][E] 坐标系类型	Default	~

输出

3 输出参数界面



通过改变检测框的大小和位置,以及相关的检测参数,使得最终的输出的 斑点个数为1。VISIONGO将使用该斑点的最小外接矩形的中心坐标(X、Y)以 及角度(A)组成一个XYA的坐标系统。

表格 2-3-3-1: 输入参数

序号	名称	参数说明
1	图像通道	选择所需颜色通道的图像进行操作 取值范围: ・Default: 默认通道,为CT_R。 ・CT_R: 红色通道 ・CT_G: 绿色通道 ・CT_B: 蓝色通道
2	复合检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色)
3	二值化	图像二值化参数
4	腐蚀	设置斑点的腐蚀系数,系数为0不腐蚀
5	膨胀	设置斑点的膨胀系数,系数为0不膨胀
6	填充	是否需要对斑点进行轮廓填充
7	通过面积选择斑点	设置面积阈值,只输出面积在阈值范围内的斑点,当上限值 为0时表示不启用此限制。
8	通过中心点x选择斑点	设置中心点X坐标的阈值,只输出中心点X坐标在阈值范围 内的斑点,当上限值为0时表示不启用此限制。
9	通过中心点y选择斑点	设置中心点Y坐标的阈值,只输出中心点Y坐标在阈值范围 内的斑点,当上限值为0时表示不启用此限制。
10	通过正矩形宽选择斑点	设置正矩形的宽度阈值,只输出正矩形宽度在阈值范围内的 斑点,当上限值为0时表示不启用此限制。
11	通过正矩形高选择斑点	设置正矩形的高度阈值,只输出正矩形高度在阈值范围内的 斑点,当上限值为0时表示不启用此限制。
12	通过斜矩形长边选择斑 点	设置斜矩形的长度阈值,只输出斜矩形长度在阈值范围内的 斑点,当上限值为0时表示不启用此限制。
13	通过斜矩形短边选择斑 点	设置斜矩形的宽度阈值,只输出斜矩形宽度在阈值范围内的 斑点,当上限值为0时表示不启用此限制。



14	通过斜矩形长宽比选择 斑点	设置斜矩形的长度比阈值,只输出斜矩形的长度比在阈值范 围内的斑点,当上限值为0时表示不启用此限制。
15	通过斜矩形角度选择斑 点	设置斜矩形的角度阈值,只输出斜矩形的角度在阈值范围内 的斑点,当上限值为0时表示不启用此限制。
16	通过Y值进行排序	是否将找到的斑点通过Y值进行排序。
17	通过序号选择斑点	输出指定序号的斑点。当序号不在斑点的个数范围之内时, 输出为空;
18	二次填充	是否对筛选后的斑点进行填充
19	联合	是否将筛选后的斑点联合成一个斑点
20	输出补偿	对斑点的中心点进行补偿再输出
21	连通域判定方式	连通域的判定方式 取值范围 ・CONN_4:4连通 ・CONN_8:8连通
22	输出坐标类型	选择颜色通道,当输入图像为彩色图像时,通过选择具有更高对比度的颜色通道,有助于更准确稳定地找到目标模板。 取值范围: • Default:默认通道,为CT_XYA。 • CT_X:红色通道 • CT_Y:绿色通道 • CT_XY:蓝色通道 • CT_XYA:蓝色通道

#### 表格 2-3-3-2: 输出参数

序号	名称	参数说明
1	坐标系统	
2	面积	检测到的每个斑点的面积
3	中心	检测到的每个斑点的中心坐标
4	正矩形宽度	检测到的每个斑点的正矩形的宽度
5	正矩形高度	检测到的每个斑点的正矩形的高度
6	斜矩形长边长	检测到的每个斑点的斜矩形的长边长
7	斜矩形短边长	检测到的每个斑点的斜矩形的短边长



8	斜矩形长宽比	检测到的每个斑点的斜矩形的长宽比
9	斜矩形角度	检测到的每个斑点的斜矩形的角度
10	轮廓	

## 2.3.4 圆心定位

<b>1.</b> 相机设置	2. 检测模块	3. 脚本编程	4. 数字输入输出	<b>5.</b> ឝ	串口通讯
🗹 坐标系统	圆心定位			$\sim$	激活

首先,如上图所示,先启用坐标系统,并选择圆心定位,然后点击激活, 此时软件界面如下:

序号 界面类型

1

界面图



	输入参数界面	<ul> <li>■ [1][C] 工具框显示</li> <li>[2][E] 模式</li> <li>[3][I] 梯度</li> </ul>	100; 100; 50; 80; 0; 360; 5; No Light to dark 30
2		[4][E] 图像通道选择 [5][E] 坐标系类型	Default Default
۷			



序号	界面类型	界面图		
	输出参数界面	输出		
		[1][CRD] 坐标系统		
0		[2][F] 直径 0.000000		
5		⊞ [3][P] 中心     □		
		æ		

#### 表格2-3-4-1: 输入参数

序号	名称	参数说明
1	图像通道选择	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
2	检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色)
3	边缘模式	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘
4	灰度梯度	设置检测框1查找边缘时的灰度梯度
5	输出坐标类型	选择颜色通道, 当输入图像为彩色图像时, 通过选择具有更高 对比度的颜色通道, 有助于更准确稳定地找到目标模板。 取值范围: • Default: 默认通道, 为CT_XYA。 • CT_X: 红色通道 • CT_Y: 绿色通道 • CT_XY: 蓝色通道 • CT_XYA: 蓝色通道



表格2-3-4-2: 输出参数

序号	名称	参数说明
1	坐标系统	
2	直径	目标圆的直径
3	中心	目标圆的圆心坐标
4	员	目标圆

## 2.3.5 线定位

1.相机设置	2. 检测模块	3. 脚本编程	4. 数字输入输出	5. 8	串口通讯
☑ 坐标系统	线定位			~	激活

首先,如上图所示,先启用坐标系统,并选择线定位,然后点击激活,此 时软件界面如下:

序号 界面类型

1

图像界面

界面图





序号	界面类型	界面图	
2	输入参数界面	■ [1][C] 工具框显示 100; 100; 90; 80; 0   [2][E] 图像通道选择 Default   [3][E] 模式 Light to dark   [4][I] 梯度 30   [5][E] 类型 CT_X	
3	输出参数界面	输出 [1][CRD] 坐标系统	

#### 表格2-3-5-1: 输入参数

序号	名称	参数说明
0	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
1	工具框设置	键入工具框的位置、大小和方向
2	查找模式	在工具框的大方向上,待测边缘的明暗变化情况
3	灰度梯度	待测边缘的灰度梯度



5

选择颜色通道,当输入图像为彩色图像时,通过选择具有更高对 比度的颜色通道,有助于更准确稳定地找到目标模板。 取值范围:

• Default: 默认通道,为CT\_X。

- ・CT\_X: 红色通道
- CT\_Y: 绿色通道

表格2-3-5-2: 输出参数

输出坐标类型

序号 名称 参数说明 \_\_\_\_\_\_

1 中心点和角度 水平基线标记线的起点和角度

#### 2.4 添加基准线

基准线,又分为水平基准线和垂直基准线,广泛用于接插件行业当中。基 准线的作用在于使得<u>阵列测量</u>中的输出[点到水平基准线的距离]和[点到垂直基 准线的距离]有效,

<b>1.</b> 相机设置	2	2.检测模块	3. 脚本编程	4. 数字输入输出	<b>5.</b> §	串口通讯
🗌 坐标系统		模板定位			$\sim$	激活
☑ 水平基准		通过查找边界(H) ~			激活	
□ 垂直基准		通过查找边界(V) ~			$\sim$	激活

水平基准线和垂直基准线的获取方法相同,目前支持的方法有通过查找边 界获取和通过两圆圆心的连线获取。

基准线的启用与坐标系的启用方法相同,首先选中需要启用的基准线(水 平基准或者垂直基准),然后在下拉框中选中使用的查找方法,最后点击激活, 此时可在界面上看到对应的检测工具框(如下图)。最终通过拖动检测工具框 获取需要的基准线。



## 2.4.1 通过查找边界

表格6-1-1: 输入参数

序号	名称	参数说明
0	图像通道	选择所需颜色通道的图像进行操作 取值范围: ・Default: 默认通道,为CT_R。 ・CT_R: 红色通道 ・CT_G: 绿色通道 ・CT_B: 蓝色通道
1	斜矩形检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色)
2	查找模式	设置检测框的边缘模式。 • Light to dark:查找由亮到暗的边缘 • Dark to light:查找由暗到亮的边缘
4	边缘梯度	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输 出出来

表格6-1-2: 输出参数

序号	名称	参数说明
1	中心点和角度	水平基线标记线的起点和角度
2	线段	基准线,主要用于显示

## 2.4.2 通过两圆圆心的连线

表格6-2-1: 输入参数

序号	名称	参数说明
1	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
2	圆环检测框1	调整检测框的位置、大小。可通过鼠标在软件界面操作,也 可以直接输入参数。(检测框的颜色为红色)



3	查找模式1	设置检测框1的边缘模式。 • Light to dark:查找由亮到暗的边缘 • Dark to light:查找由暗到亮的边缘
4	边缘梯度1	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输 出出来
5	圆环检测框2	调整检测框的位置、大小。可通过鼠标在软件界面操作,也 可以直接输入参数。(检测框的颜色为绿色)
6	查找模式2	设置检测框1的边缘模式。 • Light to dark:查找由亮到暗的边缘 • Dark to light:查找由暗到亮的边缘
7	边缘梯度2	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输 出出来

表格6-2-2: 输出参数

序号	名称	参数说明
1	线段	直线标记线的位置
2	圆1	圆1的位置和大小
3	圆2	圆2的位置和大小

## 2.5 添加函数模块

#### 第一步

在[检测模块]页面找到函数列表 框,并点击按钮[新建]

函数列表框显示所有添加的检测函 数,执行顺序是从上到下。特别注意 的是编号这一列,该编号是函数的唯 一ID号,编号可以不连续,也可以通 过排序修改,但是不能重复。

#### 6. 网口通讯(TCP/IP) 7. 图像存储

编号 000	名称 FXN1	类型 条码识别	
4	新建	拷贝	刪除


		() 算法	装模块		)	×
		序号	名称		描述	
		1	模板查找			
		2	斑点分析			
		3	亮度检测			
		4	灰度检测			
		5	对比度检测			
		6	颜色检测			
		7	条码识别			
第二步		8	二维码识别			
在弹出的对话中选择要添加的		9	单边测量			
函数模块,目前VISIONGO支持		10	对边测量			
17种功能的函数模块。		11	圆心测量			
		12	同心度测量			
		13	极点测量			
		14	极距测量			
		15	圆弧测量			
		16	阵列测量			
		17	图像比对			
		<				>
					确定	
				1005		
		C2] 上具机 1 米 刑	<b>玉</b> 示	1035	; 825; /1/; 80 EBAR	J; 0
	[2][E	」 吴空 ] 图像诵道	1	Defa	ult	
	[4][E	] 定位方式		ROI		
	[5][1]	块大小		8		
	[6][1]	曾强矩阵:	大小	7		
	[7][R	]增强因子		2		
		「粔邦附加 OP1 海ジナマ	密烈 S和社场政士	5	. 000000	
	[10]	CN 通过的 R1 通过的	叫我您好和难思。 剪辞选	2000	, ,,,,,,,,	
第三先	[10][	een heertender	41900.22	2000		



\_ . \_ \_

\_ · \_\_



VISIONGO使用手册



#### 2.5.1 函数模块的排序

函数模块的排序功能可以改变函数模板的ID号。从而实现改变函数模块在 界面的函数列表中的排序。通过上移按钮可以减小函数的ID号,通过下移按钮 可以增加函数的ID号。

#### 2.5.2 函数模块的复制

函数模块的复制能够在新建一个相同功能函数的同时复制被复制函数模块 的所有输入值。该功能对于个别场合非常有用,比如一幅图像上有多个产品, 每个产品的检测功能又都是一样时,如果没有复制功能,得一个一个地添加函 数,而且每个函数的参数都得重新设置,非常费时费力。 值得注意的是,在程序内部,每个函数模块都有一个ID号,该ID号显示在 界面的函数列表的第一列,请参考<u>章节2.5</u>。在使用函数复制功能时,新的函数 ID号是有可能比被复制的函数模块的ID小的,这也意味着新函数在界面的函数 列表中可能排在被复制的函数前。

# 2.6 添加/删除IVU可显示参数

第一步 在[检测模板]页面找到[IVU可 显示参数]的小页面,并点击 按钮[+/-]	IVU可设 序号 1	置参数 IVU可显示参数 函数名称 FXN1	輸出口名称 条码 <b>A 又</b>	
说明:右侧A区的列表框显示出所 有可在IVU中显示的参数			+	·/-
第二步 如右图所示,在弹出的对话 框中选择需要在界面显示的 参数。		请选择 FXN1 条码 条码中心点 面积 面积比 距离	× 条码识别 □ □ □ □ □ □ □ □ □ □ □ □ □	



# 2.7 添加/删除IVU可设置参数

#### 第一步

在[检测模板]页面找到[IVU可 设置参数]的小页面,并点击 按钮[+/-]

#### IVU可设置参数 IVU可显示参数



#### 说明:右侧A区的列表框显示出所 有可在IVU中设置的参数

	4	〉请选择	×
	Ξ	FXN1	条码识别
		块大小	
		增强矩阵大小	$\checkmark$
		增强因子	$\checkmark$
		矩形附加参数	
		通过面积选择斑点	
第二步 四左图所示 左硝出的对话框		通过距离筛选	
中选择需要在界面显示的参		是否使用坐标系?	
汝。			
			确定

# 2.8 添加数字输入输出(DIO)

1.相机设置 2. 检测模块 3. 脚	和本编程 4.	数字输入输	出 5. 串	口通讯	6.网	]通讯 <mark>(</mark> T	CP/IP)	7. 囹僂	存储									
通用		一输入一						输出								DO1	D1&&D131&&D132	~
		> 滤波等	级(us)	10			A V		延时(n	ns) [0	0				÷	plc simulator	True	
	177.77							脉冲	宽度(n	ns) 🛙	10				-	FXN1	D1	
端口	刊开															FXN131	D131	
		1	2 3	4	5	6 7	8	1	2	3	4	5	6	7	8	FXN132	D132	
																DO2		
																plc simulator	False	
		R	10/11	12	13	14 1	5 16	9	10	11	12	13	14	15	16	FXN1		2
		B	4											- 4		FXN131	レビ	
																FXN132		×



序号	功能分类	说明
AIZ	IO卡选择区	<ul> <li>a. 相机的类型为SGSMART的IB系列和RKT系列时,本区域的所有内容无须设置;</li> <li>b. 当相机的类型为其它时,需在下拉框中选择IO卡和端口。</li> <li>IO卡:目前IO卡只支持ADLINK的PCI-7230和南京雍异电子的COMRELAY-6。</li> <li>端口:对于PCI-7230来说,端口号是从0开始的整数,表示的是第几张IO卡,0表示第一张,1表示第二章;对于COMRELAY-6来说,端口号是指其与PC相连的串口号,如COM1,COM2。</li> </ul>
B⊠	输入设置及状态 监测区	<b>滤波等级:</b> 单位微秒,用来消除噪声和按键抖动带来的干扰。 例如当滤波等级设置为100us,可以有效去除100us以下的干扰 信号。
C⊠	输出设置及状态 测试区	<b>延时(ms):</b> 检测功能执行完到实际输出信号之间的时间间隔, 默认情况下该值为0,即功能检测完成后立即输出IO信号。如果 该值为10,表示检测功能执行完成后等待10ms再输出信号。 <b>脉冲宽度(ms):</b> 输出信号的持续时间。
DX	信号输出逻辑设 置区	可设置每个输出口的逻辑输出情况。 input check:控制输出信号的输出模式,当取值为false时,表 示采用延时输出的方式,及检测完成后经过一定的延时时间后 输出信号;当取值为true时,表示采用输入验证的方式,即判断 对应的输入口上是否有信号,如果有信号再执行信号输出操 作,否则一直等待输入口的信号。

# 2.9 添加串口输出

相机设置 2. 检测模块 3. 脚本编程 4. 數字输入输出 5. 串口通讯 6. 网口通讯(TCP/IP) 7. 图像存储								
com1 ~ 9600 ~ 8N1 ~ 打开	标准协议 シー							
+/- 标准协议 ~ 发送	字符格式 ASCII 🗸 3 🗘 2 🛟 🗆 STX(起始符02) 🗌 ETX(结束符03)							
	消息前缀							
	前缀 🔜 🔜 🔜 🔤 👘 👘 👘 👘							
	分隔符 🛛 🗸 🗸							
	分隔符(结构体) / 分隔符(数组) /							
上								
	消息后缀							

串口通讯设置界面主要由上图的左右两部分组成,其中右边部分主要为通 讯协议的设置,通讯协议的部分将放在<u>章节2.9.1</u>中详细说明,在本章节中将不 做说明。



左边为串口设置区,可以设置串口的基本参数,包括端口号、波特率,以 及发送时执行的协议等,通过点击按钮[+/-]可以添加或者删除需要从串口发送的 数据,具体如下图:

# 2.9.1 标准的通讯协议

标准协议					~
字符格式	ASCII ~	3	2	□ STX(起始符02)	🗌 ETX(结束符03)
消息前缀		~		🗌 标题	
前缀			~	🗌 函数序号	
分隔符	~		~		
分隔符 <mark>(</mark> 结构体 <mark>)</mark>	~		~		
后缀	~		~		
消息后缀		~			

序号	名称	说明
1	字符格式	数据传输的格式,默认为ASCII码; 数据的整数位的位数; 数据的小数位的位数;
2	消息前缀	整个消息字符串的前缀;
3	前缀	单个函数模块的前缀,可根据模块的结果(合格或者不合格) 分别进行设置;
4	分隔符	输出参数之间的间隔符;
5	分隔符(结构体)	结构体内部的间隔符,用来分割结构体内的元素;
6	分隔符(数组)	数组内部的间隔符,用来分开数组内部的元素;
7	后缀	单个函数模块的结束符;
8	消息后缀	整个消息字符串的结束符;
9	STX	起始符
10	ETX	结束符
11	标题	
12	函数序号	



## 2.9.2 Modbus 通讯协议

Modbus是由Modicon(现为施耐德电气公司的一个品牌)在1979年发明 的,是全球第一个真正用于工业现场的总线协议。ModBus网络是一个工业通信 系统,由带智能终端的可编程序控制器和计算机通过公用线路或局部专用线路 连接而成。其系统结构既包括硬件、亦包括软件。它可应用于各种数据采集和 过程监控。ModBus网络只有一个主机,所有通信都由他发出。网络可支持247 个之多的远程从属控制器,但实际所支持的从机数要由所用通信设备决定。采 用这个系统,各PC可以和中心主机交换信息而不影响各PC执行本身的控制任 务。

Modbus的特性

a. 标准、开放,用户可以免费、放心地使用Modbus协议,不需要交纳许可证费,也不会侵犯知识产权。目前,支持Modbus的厂家超过400家,支持Modbus的产品超过600种。

b. Modbus可以支持多种电气接口,如RS-232、RS-485等,还可以在各种 介质上传送,如双绞线、光纤、无线等。

c. Modbus的帧格式简单、紧凑,通俗易懂。用户使用容易,厂商开发简 单。

## 2.10 添加网口输出

网口通讯设置界面主要由上图的左右两部分组成,其中右边部分主要为通 讯协议的设置,通讯协议的部分将放在<u>章节2.9.1</u>中详细说明,在本章节中将不 做说明。

左边为网口设置区,可以设置网口的基本参数,包括IP地址、端口号,以 及发送时执行的协议等,通过点击按钮[+/-]可以添加或者删除需要从网口发送的 数据,具体如下图:

1.相机设置 2. #	检测模块 3. 脚本编	程 <b>4</b> . 数字输	入輸出	5. 串口通讯	、 6. 网口通讯(TC	P/IP) 7. 图	像存储				
□ 作为服务器	127.0.0.1	8000		断开	标准协议					~	
+/-		标准协议 ~		发送	字符格式	ASCII	3	2	🗌 STX(起始符02)	🗌 ETX(结束符03)	
	<u> </u>				消息前缀		~		□ 标题		
					前缀			~	□ 函数序号		
					分隔符	· · · · ·		~			
					分隔符(结构体)	×	/ 分隔符(数组)	~			
					一 后缀	×		~			
					消息后缀		~	]			

值得注意的是,这里还有一个作为服务器的选择项,当选中时, VISIONGO将在本机上创建一个TCP服务器,以供外部TCP客户端进行连接; 否则将创建一个TCP客户端。

## 2.11 添加图像保存

#### 2.11.1 本地存储

如下图所示,选择类型为"关闭"时,VISIONGO软件关闭图像保存功能;

选择类型为"本地"时,VISIONGO软件会自动将目标图片保存到本地PC中;

选者类型为"FTP"时,VISIONGO软件通过网口FTP协议将图像保存到另一台PC中。

─类型 ───		
〇 关闭	⊙ 本地	C FTP

当图像保存的类型选择类型为"本地"时,可在"本地"控件中指定图像保存的路径。

本地	
C:\Users\Alex\Pictures\Saved Pictures	Browse

在"保存方式"控件中设置保存图片的类型是任意图片、合格图片或不合格图 片。当选择为合格图片时,软件只保存检测结果为OK时的图片,同理,选择为 不合格时,软件只保存检测结果为NG时的图片。

同时可在"存储策略"控件中选择无限、限制数量或循环保存。选择为无限表示在指定路径中保存无限张图;选择为限制数量表示只保存在"图像数量"空间中键入的图片数量,数量满了之后就不保存;选择为循环保存表示循环保存在"图像数量"空间中键入的图片数量;

-保存方式-		
④ 任意	〇 确定	〇 不合格
存储策略	无限	•
图像数里	1	•

在"本地/FTP"控件中分别设置保存OK、NG图片的文件夹名称,软件运行 时会自动在"本地"控件设置的路径下新建子目录"项目特征名/相机号/年月日/时/ 文件夹名称",并将图片保存在该目录下。

┌本地/FTP			
合格文件夹	OK1	不合格文件夹	1NG

#### 2.11.2 FTP存储

FTP存储即软件通过网口FTP协议实现将目标图片保存到目标PC中,设置 步骤如下:

一、创建FTP服务器用户

在网上下载FTP服务器软件后在目标PC上安装并创建用户,设置用户名和 密码。

二.VISIONGO软件设置

在"类型"控件中选择图像保存的类型为FTP;

在"保存方式"控件中设置设置要保存的图片类型是任意图片、合格图片 或不合格图片;

在"FTP"控件中键入目标PC的IP地址、FTP服务器上已创建的用户名、 用户密码;

点击"连接"按钮连接FTP服务器,如下图所示。

FTP —	FTP				
用户名	xiaoxiong	密码	***		
IP地址	192.168.1.23		连接		

## 2.12 项目保存/项目另存

因项目新建或者打开时已经指定了相机的编号,因此项目保存是一件非常 简单的事情,简单到只要点击主界面的保存按钮,整个过程甚至不会出现一个 最简单的确认对话框。

本节我们主要讨论项目另存的方法。如<u>章节2.1</u>所述,VISIONGO支持6个 相机。因此在项目另存时(改变相机序号)需要选择新的相机序号。



保存	× <b>B</b> 系列智能相机 机械手抓取定位及引导 / 读码 产品计数 / 高精度测量
相机	相机1 ~
任务编号	1 🖌 🗸
专有名称	
文件名称	
	确定 取消

项目文件的命名方式请参考章节2.1.1。

# 2.13 IVU中运行项目

IVU为VISIONGO的用户界面,IVU具有界面整洁,跟IVP相比,IVU去掉了 所有的编程功能,因此更适合最终用户的使用。

IVU可自动读取配置文件夹中的配置文件并加载到软件中。根据配置文件的 的设置,IVU可实现对函数模块的输入参数的修改。修改后的值立即更新到软件 并会自动保存。



# 2.13.1 IVU主界面介绍



序号	区域	说明
1	AIZ	工具栏 a. 开机启动,选中后可实现开机自启动; b. 在线显示,实时显示相机采集到的图像; c. 开始检测,进入正常检测状态; d. 停止,停止显示或者检测; e. 参数,显示/隐藏参数页面; f. 统计,显示/隐藏统计页面; g. 检测框,是否显示检测框; h. 过程,是否显示检测过程,即可在图像上显示的结果;
2	Β⊠	显示统计结果,含合格数,总数,合格率,采集时间,检测时间等
3	C区	任务切换,选择下拉框可实现手动切换
4	D区	检测模块结果输出区,需要在IVP中设置,请参考 <u>章节 2.6</u>
5	EX	检测模块参数输入区,需要在IVP中设置,请参考 <u>章节 2.7</u>



序号	区域	说明
6	F区	图像显示 右上角显示检测结果(OK/NG) 左下角显示触发序号 右下角显示单次触发下的图像序号

# 2.13.2 多相机的支持

IVU最多支持6个相机同时工作,如下图所示。



序号	图例	说明	
CAM1			
CAM2			
CAM3			
CAM4			
CAM5			
CAM6			



## 2.13.3 自动切换任务

序号 切换方式

说明

1	DIO切换	切换指令: IO板上最后一个输入口为高电平 切换序号: 由IO板上其它输入口(除最后一个输入口)的输入状态组 成的二进制代表的数字。 以视谷的ROCKET系列相机为例, ROCKET相机拥有8个输入口 输入状态: 1-000-0001, 代表切换到任务1; 输入状态: 1-000-0010, 代表切换到任务2; 输入状态: 1-000-0011, 代表切换到任务3; 输入状态: 1-111-1111, 代表切换到任务127; 由上可知, 通过DIO切换任务, 可支持的切换任务数量根据IO板的输入 口的数量不同而不同。
		切换指令:SGCT 切换序号:??? (取值范围为001~999)

2 串口切换 例如发送SGCT001,可切换到任务1,发送SGCT002,可切换到任务 2,发送SGCT003可切换到任务3等。串口方式切换任务最多支持999 个任务的切换。

切换指令:SGCT

切换序号:???(取值范围为001~999)

3 网口切换 例如发送SGCT001,可切换到任务1,发送SGCT002,可切换到任务
 2,发送SGCT003可切换到任务3等。串口方式切换任务最多支持999
 个任务的切换。



# 第三章 标定

## 3.1 相机标定

相机标定用于实现单位的转换。一般情况下,软件处理后的尺寸结果是以像素(pixel)为单位的,经过相机标定后,可将其转换为实际指定的单位。 VISIONGO软件的相机标定有两种方法,分别是圆点标定和简易标定。

序号	标定方式	说明
1	圆点标定	精确标定,可适应于各种场合,但是需要用到特制的 标定板,因此在某些场合下不太好实施,比如接插件 检测,PIN脚的位置一般不太好放置标定板。
2	简易标定	粗略标定,通过添加比例转换值和偏移量来实现。简 单易实施,而且对于物体位置变化不大的情况下也能 实现较高的精度。

## 3.1.1 圆点标定

圆点标定一般需要用到特质的标定板,这种标定板在市场上是可以购买的,价格根据精度的不同有较大的差别。以下我们使用从CAD中打印出来的标定纸 进行标定过程说明,实际使用时建议从市场上采购精度更高的专业标定板。



说明

a. 添加斑点分析函数模块,拖动检测工 具框并设置好相关参数,使工具能准 确找到圆点;





- b. 点击菜单栏中的"相机标定"按钮, 弹出 右图所示的标定窗口;
- c. 输入标定板对应的参数, 如行列和间 距等。
- d. 选择数据源, 所谓的源即上一步中找 到的Blob的中心点的位置坐标(也即 Blob工具的输出),可直接从下拉框 中选择。
- e. 点击按钮[相机标定],此时会自动计算 出公差
- f. 如需测试, 可输入对应的图像坐标, 可立即看到输出的世界坐标。

나는 귀평

#### 3.1.2 简易标定

简易标定通过键入尺寸比例值和偏移量来进行标定、以实现尺寸的单位转 换。简易标定的一般步骤如下

		况明	
a.	找一个标准产品并用游标卡尺(相似测量 工具也可以)多次测量出其相关尺寸,并 记录平均值V1。	相机标定 ×	
b.	通过IVP软件多次测量与第一步中同样的 尺寸,并记录多次测量后的平均值V2。	比列值(mm/pixel) 0.0327 偏移量(mm) 0	
c.	R=V1/V2;把R值填入到右图中的比例值输 入框中。		
d.	点击确定完成标定。		
e.	继续多次用软件测量相关尺寸并与游标卡 尺的测量值进行比较,适当的时候添加相 应的偏移量到标定界面的偏移量输入框 中		
	.1.0	确定	

# +B+D+C=

相机标定			×
<ul> <li></li></ul>	Ē		
, 行数 3	▲ ● 间距	15	•
列数 4	▲ ● 间距	15	▲ ▼
源 000	•	ħ	即标定
公差统计			
和 0.760	平均值 0.063		重要
最小值 0.000	最大值 0.181		±#
测试			
图像坐标(pixel) 10	00.008	99.99	8
世界坐标(mm)4	1.778	-1.06	9
			确定

~ ~ ~ ~

说明



## 3.2 手眼标定

手(机械手)眼(相机)标定用于建立相机与机械手或其它自带坐标系统 的平台间的坐标系关系矩阵,以实现从相机坐标系到平台坐标系的转换。经过 手眼标定后,VISIONGO软件可以直接输出特征原点在平台的坐标系下的坐标 或基于平台坐标系下某点的偏移量,坐标原点为平台的坐标原点。

VISIONGO软件的手眼标定针对不同的相机安装方法有不同的标定方式, 相机固定朝下安装时选择相机朝下的标定方式,相机固定朝上安装时选择相机 朝上的标定方式,相机安装在机械手上时选择眼随手动的标定方式。

## 3.2.1 眼随手动/眼手分离(眼朝上)

眼随手动是相机安装在机械手上的一种标定方法。与相机朝上标定方法类 似,该方法也是通过输入多个点对来实现标定的。不同的是眼随手动标定的多 个点的获取是通过机械手带着相机一起移动不同的位置得到的(相机动、机械 手动、样品不动)。标定完成后,软件输出的x、y是基于相机移动的第一个位 置x1、y1的相对值,即在在x1、y1上的偏移量。

以九点标定为例,标定方式如下:

步骤

说明

a. 机械手抓带着相机移动到样品上方,调整相机相关参数式样品能正常成像;



- b. 点击菜单栏中的"手眼标定"按钮, 在弹出的标定窗口中选择"眼随手 动"选项,如下图所示,其中图像坐 标表示特征点在图像中的坐标,世 界坐标表示特征点在机器人坐标系 统下的坐标,X、Y分别对应坐标x 的值和y的值,图像坐标的x、y可 以键入也可以通过数据源获取,世 界坐标则只能键入;
- c. 移相机(机械手),使待测原件成 像在图像的左上角,添加函数模 块"斑点分析"(因为待测原件为规 则的黑色矩形块,斑点分析函数可 以根据灰度阈值的设定找出待测原 件的中心点)用于获取样品的图像 坐标,调整斑点分析函数模块的检 测框并设置相关参数,使其能正确 查找到黑色原件所有区域,如下图 所示,软件在查找到原件后会用颜 色标记出来;



说明







说明

d. 如下图所示,图像坐标的数据源选择"000,Center",然后点击"获取"按钮就可以直接获得特征点的图像坐标,输入对应点的世界坐标后点击"+"按钮将该点对添加到点列表中;

图像坐标(pixel) 192.000 210.000 000,Center 🗸 获取					
世界	₹坐标(mm) 0	1	0		▼ 获取
	Α	В	с	D	+
1	192.000	210.000	0.000	0.000	•
相机标定 <b>结果与测试</b>					
					确定

e. 按照九宫图的顺序依次将同一个样品移动到相机视野中的上左->上中->上右->中右->中中->中左->下左->下中->下右等位置,如下图所示, 共移动9次;







说明

- f. 每次移动后重复步骤c、d,依次添加9个点的图像坐标和世界坐标,添加完成后如下图所示;
- g. 步骤四完成后、点击"相机标定"按 钮进行标定计算并显示标定结果, 如下图所示,在标定结果中可以查 看标定的总误差、平均误差、最小 误差和最大误差,误差的单位即世 界坐标系的单位,标定结果满足需 求后点击"确认"按钮完成标定;

机器人	标定					×	
标定类型 C 相机朝下 C 相机朝上 ⊙ 眼随手动							
图像坐标(pixel) 860.000 777.000 000,Center 🗸 获取							
世界	坐标(mm)	60	60		•	获取	
	Α	В	с	D		^ +	
1	192.00	0 210.000	0.000	0.000			
2	511.00	0 211.000	30.000	0.000		-	
3	821.00	0 210.000	60.000	0.000		_	
4	820.00	0 489.000	60.000	30.000			
5	477.00	489.000	30.000	30.000			
6	133.00	488.000	0.000	30.000			
7	142.00	0 778.000	0.000	60.000			
8	498.00	0 776.000	30.000	60.000		¥	
					相机	は标定	
▼ 结	果与测试						
和	0.459	平均值	0.051		(	<b>—</b> —	
最小	值 0.000	最大值	0.280		L	里宜	
测试							
图像4	삼标(pixel)	99.995	99.999				
世界公	坐标(mm)	-10.121	-13.827				
					i	确定	

# 3.2.2 手眼分离(眼朝下)

相机朝下是相机固定朝下安装时采用的标定方法。该方法通过获取视野中 多个(最少六个)固定特征点的图像坐标和与其一一对应的机器人的坐标来确 定两个坐标系间的关系矩阵。标定完成后,软件输出的坐标为机械手的绝对坐 标,即在机械手坐标系下的坐标。

标定前准备:

a. 标定板一块,用于选取特征点;

b. 在机械手上安装校准工具,用于指示特征点。

以八点标定为例,标定基本步骤如下:



L	E	ΠEQ
/	ァ	镢

说明

- a. 调整相关参数使特征点清晰成像,采集特 征点图像,选取下图所示八个点作为标定 用的特征点;
- b. 添加能获取特征块中心点的函数模块(如本例的特征块是规则的原点对称图形,可以选择斑点分析或圆心测量函数模块,本例选择斑点分析来获取图像的中心点),调节函数模块工具框和相关参数,使其能正确查找到指定特征块的中心点
- c. 点击菜单栏中的"手眼标定"按钮,在弹出的标定窗口中选择"相机朝下"选项,如下图所示,其中图像坐标表示特征点在图像中的坐标,世界坐标表示特征点在机器人坐标系统下的坐标,X、Y分别对应坐标x的值和y的值,图像坐标的x、y可以键入也可以通过数据源获取,世界坐标则只能键入;







#### VISIONGO使用手册

#### 步骤

说明

d. 如下图所示,图像坐标的数据源选 择"000,Center",然后点击"获取"按钮就 可以直接获得特征点的图像坐标,输入对 应点的世界坐标后点击"+"按钮将该点对 添加到点列表中;

e. 重复步骤c和步骤d,依次添加八个点的图 像坐标和世界坐标,添加完成后如下图所 示;

机器	人标定					×
标	定类型					
•	相机朝下 〇相	1机朝上 () 服	随手动			
图像	坐标(pixel) 2	18.448	154.049	000,Center	•	获取
世界	₹坐标(mm) 0	)	0		•	获取
	A	В	с	D		+
1	218.448	154.049	0.000	0.000		
						-
					相机	「标定
····································						
					i	确定
					_	

机器	人标定				×
标	定类型				
•	相机朝下 〇 相	朝朝上の眼	随手动		
图像	象坐标(pixel) 2	206.701	1068.658	000,Center	▼ 获取
世界	界坐标(mm) C	)	30		▼ 获取
	Α	В	с	D	^ +
1	218.448	154.049	0.000	0.000	
2	675.435	156.760	15.000	0.000	-
3	1136.840	162.427	30.000	0.000	
4	1590.700	172.539	45.000	0.000	
5	1576.850	1083.840	45.000	30.000	
6	1122.780	1080.580	30.000	30.000	
7	663.380	1075.770	15.000	30.000	
8	206.701	1068.660	0.000	30.000	~
相机标定					
▶ 结果与测试					
					确定



#### VISIONGO使用手册

确定

步骤

说明

	机器/ - 标选 ⓒ 木	↓标定 送型 目机朝下 ○	相机朝上〇間	1000手动		×
	图像	坐标(pixel)	206.701	1068.658	000,Center	获取
	世界	坐标(mm)	0	30		▼ 获取
		Α	В	с	D	^ +
	1	218.44	48 154.049	0.000	0.000	
f. 点击"相机标定"按钮进行标定计算并显示	2	675.43	35 156.760	15.000	0.000	-
标定结果。如下图所示。在标定结果中可	3	1136.84	40 162.427	30.000	0.000	
	4	1576.85	0 1/2.539 50 1083 840	45.000	30,000	
以查看标定的总误差、半均误差、最小误	6	1122.78	1080.580	30.000	30.000	
差和最大误差,误差的单位即世界坐标系	7	663.38	1075.770	15.000	30.000	
的单位,标定结果满足需求后点击"确	8	206.70	1068.660	0.000	30.000	▼ 相机标定
认"按钮完成标定;	▼ 结	课与测试				
	和	0.432	平均值	0.054		
	最小	值 0.000	最大值	0.153		里豆
	测试 图像≙	坐标(pixel)	99.992	99.994		
	世界	坐标(mm)	-3.891	-1.717		



# 第四章 检测模块详解

## 4.1 模版查找



与模板定位一样,在使用模板查找的功能前,需先进行模板训练,具体方法请请参考章节2.3.1.1。

模板查找模块可实现在检测框范围查到多个模板,并以数组的形式输出查 找到的模板的相关信息,同时也可以输出多个坐标系,极大地简化了多模板和 多坐标系的视觉应用。

表格 4-1-1: 输入参数

序号	名称	参数说明
1	双击训练模板轮廓	双击之后进入模板训练界面开始训练模板轮廓
2	输入坐标系	选择输入坐标系
3	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道



序号	名称	参数说明
4	斜矩形检测框 [RECT2]	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色)
5	起始角度	查找模版轮廓的起始角度
6	终止角度	查找模版轮廓的终止角度
7	最小匹配分数	待查找的形状与模板的相似程度,取值范围(0,1],值越大,检 测速度越快,但是查找到目标的可能性越低,默认值为0.5。
8	贪心指数	取值范围(0,1],默认值0.9
9	极性类型	设置查找模版轮廓时的极性。 • Normal:无极性查找 • Ignore global polarity:忽略全局明暗变化 • Ignore local polarity:忽略局部明暗变化
10	目标模板数量	设置需要查找的目标模板数量,当找出的模板达到该值时,不 再查找其它模板。即最多只输出跟该值一样数量的模板。
11	位置偏移量	设置模版中心点的偏移量; 当偏移量的值不为0时,程序会自动在找出的模板中心点的 X、Y坐标上分别加上偏移量,然后再输出。直接影响的输出 口的编号为2。
12	是否输出坐标系	设置是否需要输出一个新的坐标系,如果选择输出新的坐标 系,新的坐标系与之前的坐标系将同时存在,并不会覆盖之前 的坐标系。

#### 表格 4-1-2: 输出参数

序号	名称	参数说明
1	数量	查找到的模版数量
2	中心点和角度	模板中心点坐标及角度;该角度不是真实的角度,是相对于模 板的角度。
3	匹配分数	匹配分数
4	轮廓	模板的轮廓
5	输出坐标系	新的输出坐标系,因模板查找可能输出多个新模板,因此也将 可能会输出多个新的坐标系



# 4.2 斑点分析



斑点,在数学上也成为连通域。所谓斑点分析,一般是先通过连通域分析 的相关方法把图像上不相连的区域分割开来(如下图所示,每种颜色代表一个 连通域),然后再通过连通域的相关属性进行分类和刷选,并从中挑选出需要 的连通域的过程。

连通域分析的方法主要有4连通和8连通两种方式,如下图。所谓4连通,即 把与目标点在上下左右等4个方向具有相连关系点才认为是一个连通域;而8连 通把与目标点在8个方向具有相连关系的点都认为属于同一个连通域。



#### 4连通

8连通

从上图中很容易判断出来,4连通的算法要比8连通的算法的运算量小,因此4连通算法的运算速度也更快。显然8连通也有自己的优势,因为它更符合我们的习惯,比如以下场合,一般我们都会认为是一个连通域,但是如果采用4连通的算法,则会判定为3个连通域。





至于到底是用4连通还是8连通的办法,完全可以根据需要自由选择,在能 满足现场要求的情况下,尽量使用4连通的方法,以提高运算速度。

表格 4-2-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	复合检测框	调整检测框的位置、大小和方向。可通过鼠标在软件 界面操作,也可以直接输入参数。(检测框的颜色为 红色) 可修改检测框的类型,如圆形、矩形、圆弧形。
4	二值化参数	用此参数的上下限对图像进行二值化操作,把图像灰 度在上下限范围内的设置为前景,在二值化参数之外 的设置为背景。通过二值化操作后,把具有256个灰 度级别的灰度图像转化为只有2个灰度级别(0和 255)的二值化图像。
4	腐蚀系数	设置斑点的腐蚀系数,系数为0不腐蚀
5	膨胀系数	设置斑点的膨胀系数,系数为0不膨胀
6	是否填充	选择是否对斑点进行填充操作
7	通过数量判定结果	根据检测的数量是否在阈值内判定产品OK/NG
8	通过距离判定结果	根据检测的距离是否在阈值内判定产品OK/NG
9	通过面积选择斑点	选出在面积阈值内的斑点
10	通过中心点x选择斑点	选出在中心点x阈值内的斑点



序号	名称	参数说明
11	通过中心点y选择斑点	选出在中心点y阈值内的斑点
12	通过正矩形宽选择斑点	选出在正矩形宽阈值内的斑点
13	通过正矩形高选择斑点	选出在正矩形高阈值内的斑点
14	通过斜矩形长边选择斑点	选出在斜矩形长边阈值内的斑点
15	通过斜矩形短边选择斑点	选出在斜矩形短边阈值内的斑点
16	通过斜矩形长宽比选择斑点	选出在斜矩形长宽比阈值内的斑点
17	通过斜矩形角度选择斑点	选出在斜矩形角度阈值内的斑点
18	通过Y值进行排序	选择是否将找到的斑点通过Y值进行排序
19	通过序号选择斑点	选择序号在序号阈值内的斑点
20	是否填充	选择是否对筛选后的斑点进行填充
21	联合	选择是否将筛选后的斑点联合成一个斑点
22	位置偏移量	对斑点的中心点进行补偿再输出
23	连通算法	选择连通域算法

#### 表格 4-2-2: 输出参数

序号	名称	参数说明
1	轮廓	斑点的轮廓,显示在图像界面
2	斑点数量	检测到的斑点的个数
3	面积	检测到的每个斑点的面积
4	中心	检测到的每个斑点的中心坐标
5	宽度	检测到的每个斑点的正矩形的宽度
6	高度	检测到的每个斑点的正矩形的高度
7	斜矩形长边长	检测到的每个斑点的斜矩形的长边长
8	斜矩形短边长	检测到的每个斑点的斜矩形的短边长
9	斜矩形长宽比	检测到的每个斑点的斜矩形的长宽比
10	斜矩形角度	检测到的每个斑点的斜矩形的角度



序号	名称	参数说明
11	距离	序号相邻的斑点的中心点间距

#### 4.3 亮度检测

通过判断图像某个区域内的平均灰度值和标准差是否在一定范围内,从而 判定产品是否合格的方法。该方法广泛用于产品的有无检测等。对于 RGB图 像,该函数可对图像的任意一个颜色通道进行判断。默认情况下不对标准差进 行判断,除非在输入中设置了标准差的上下限范围。

表格 4-3-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	复合检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界 面操作,也可以直接输入参数。(检测框的颜色为红 色) 可修改检测框的类型,如圆形、矩形、圆弧形。
4	灰度平均值范围	灰度平均值的合格范围,当检测区域内的的平均灰度值 不在此范围内,则认为产品不合格。
5	灰度标准差范围	灰度标准差的合格范围,当检测区域内的的灰度标准差 不在此范围内,则认为产品不合格。 如果该值的上限为0,则表示不使用本参数对产品进行合 格/不合格的判断



表格 4-3-2: 输出参数

序号	名称	参数说明
1	灰度平均值	检测区域内的灰度平均值
2	灰度标准差	检测区域内的灰度标准差

## 4.4 灰度检测

通过判断图像某个区域内指定像素值区间的像素点个数,以及像素点个数 占整个区域内的像素点的百分比是否在一个范围内来判断产品是否合格的方法。

表格 4-4-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	复合检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操作,也 可以直接输入参数。(检测框的颜色为红色) 可修改检测框的类型,如圆形、矩形、圆弧形。
4	二值化参数	用此参数的上下限对图像进行二值化操作,把图像灰度在上下限范 围内的设置为前景,在二值化参数之外的设置为背景。通过二值化 操作后,把具有256个灰度级别的灰度图像转化为只有2个灰度级别 (0和255)的二值化图像。
5	像素数量 [TOR]	公差值,二值化图像中前景像素点的个数;如果该参数的上限值为 0,则不启用该条件判断;否则将启用该条件判断,且如果二值化 图像中的前景像素点的个数不在此范围内,则判定产品不合格。
6	百分比 [TOR]	公差值,即二值化图像中前景像素点占整个区域像素点的比值范 围,当比值范围不在此区间时,产品将会判定为不合格。



表格 4-4-2: 输出参数

序号	名称	参数说明
1	百分比	二值化图像中前景像素点占整个区域像素点的比值
2	像素数量	二值化图像中前景像素点的个数
3	轮廓	

## 4.5 对比度检测

表格 4-5-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	复合检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色) 可修改检测框的类型,如圆形、矩形、圆弧形。
4	对比度	公差值,当计算出来的对比度不在此区间时,产品将会判定为 不合格。

表格 4-5-2: 输出参数

序号	名称	参数说明
1	对比度	工具框内某通道图像的明暗对比度

#### 4.6 颜色检测

通过颜色的判断来区分待检测产品属于哪一类产品。

本模块支持六个颜色模板,即可实现在六个颜色模板范围内找出与当前产 品颜色最接近的模板,模块可直接输出最接近的颜色模板编号,以及颜色值。



用户需手动设置颜色模板,颜色模板的设置方法:把模板产品图片载入软 件并设置为当前活动图像,使用颜色工具测量出指定区域的颜色值,并把值抄 入到对应的颜色模板中。

表格 4-6-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	复合检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色) 可修改检测框的类型,如圆形、矩形、圆弧形。
3	颜色模版1	
4	颜色模版2	
5	颜色模版3	
6	颜色模版4	
7	颜色模版5	
8	颜色模版6	
9	最小色差	当输出色差小于该最小色差值时,产品将判定为合格,否则 判定不合格。

表格 4-6-2: 输出参数

序号	名称	参数说明
1	颜色值	当前产品的颜色值
2	颜色模板序号	与检测框内的图像最接近的颜色模版的ID
3	色差	检测框内的颜色值与最接近的颜色模板的颜色值的差异(欧 拉距离)



# *4.7* 条码识别



表格 4-7-1: 输入参数

序 号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	斜矩形检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色)
4	类型	条码类型
5	定位方式	选择条码定位方式: ROI:手动调整矩形大小并覆盖条码区域进行单条码读码; AUTO:手动调整斜矩形检测框大小,斜矩形检测框的大小 可远远大于真实的条码区域,软件可在斜矩形检测框的范围 内自动查找出真实的条码区域并进行读码,此种方式可一次 性读取多个通类型的条码。



序 号	名称	参数说明
6	块大小	块的大小不应小于条码的条和空的宽度,块设置过小容易找 不到边缘(基于梯度检测),生成不了条码区域,设置过大 容易忽略细密的边缘变化,生成的部分条码区域
7	增强矩阵大小	用于增强图像的对比度
8	增强因子	图像增强倍率
9	外接矩形附加参数	用于条码外接矩形辅助,设置说明:条码定位完成后,外接 矩形大小刚好大致为条码大小,通过延伸外接矩形的长度使 外接矩形完整覆盖条码
10	条码区域面积筛选	用于筛除定位干扰,滤除小区域的干扰块,设置说明:下限 值针对条码的大小可以相应的调整(随条码面积变化),上 限值无需修稿
11	距离筛选	距离的平方值,设置说明:当单个条码被多个检测框同时检 测时,通过比较检测框中心距来筛除额外检测框

#### 表格 4-7-2: 输出参数

序号	名称	参数说明
1	条码内容	解析出的条码的内容
2	最小外接矩形框	
3	条码中心点	
4	面积	
5	面积比	
6	距离	
7	轮廓	



# 4.8 二维码识别



表格 4-8-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	斜矩形检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色)
4	二维码类型	
5	增强矩阵大小	用于增强图像的对比度
6	增强因子	图像增强倍率
7	腐蚀矩阵大小	腐蚀通常用来消除噪点,分割出独立的图像元素
8	膨胀矩阵大小	腐蚀的反操作



序号	名称	参数说明
9	模式	用于选择膨胀腐蚀的操作顺序,设置说明:Image Open:先腐 蚀后膨胀,具有消除细小物体,在纤细处分离物体和平滑较大 物体边界的作用;Image Close:先膨胀后腐蚀,具有填充物 体内细小空洞,连接邻近物体和平滑边界的作用
10	均值滤波矩阵大小	用于对原图进行卷积操作,值越大图像越模糊同时去噪效果则 越好

表格 4-8-2: 输出参数

序号	名称	参数说明
1	Code	
2	Point List	

#### 4.9 单边测量

单边测量模块具有设置模板线段的功能,正常运行时可实时判断实际线段 与模板线段的角度和距离关系。因此可用于对产品进行位置校准。



表格 4-9-1: 输入参数

序号 名称	参数说明

1 输入坐标系 选择输入坐标系



序号	名称	参数说明
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: ・Default: 默认通道,为CT_R。 ・CT_R: 红色通道 ・CT_G: 绿色通道 ・CT_B: 蓝色通道
3	斜矩形检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色)
4	查找模式	设置检测框1的边缘模式。 • Light to dark:查找由亮到暗的边缘 • Dark to light:查找由暗到亮的边缘
5	边缘梯度	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输出 出来
6	边缘类型	设置输出直线方向 Follow:标记线的方向与已查找到的线的方向保持一致 Vertical:标记线的方向垂直于画布 Horizital:标记线的方向平行于画布
7	训练模板线段	选择训练一条模板线段
8	模板线段	模板线段的参数值
9	角度范围	通过当前检测到的线段与模板线段之间的夹角来判断产品是否 合格,如果夹角在设定的范围内,则合格;
10	距离范围	通过当前检测到的线段与模板线段之间的距离来判断产品是否 合格,如果距离在设定的范围内,则合格;

#### 表格 4-9-2: 输出参数

序号	名称	参数说明
1	线段	查找到的线段
2	中心点和角度	线段的中心点和角度,可用于直接输出给机械手
3	距离	查找到的线段到模版线段的距离
4	角度	查找到的线段与模板线段的夹角


## 4.10 对边测量

## 对边测量,可用来测量两条直线的距离和角度等参数。



表格 4-10-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	斜矩形检测1	调整检测框的位置、大小和方向。可通过鼠标在软件界面操作,也 可以直接输入参数。(检测框的颜色为红色)
4	查找模式1	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘
5	边缘梯度1	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输出出来
6	斜矩形检测2	调整检测框的位置、大小和方向。可通过鼠标在软件界面操作,也 可以直接输入参数。(检测框的颜色为绿色)
7	查找模式2	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘
8	边缘梯度2	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输出出来



序号	名称	参数说明
9	边缘类型	设置输出直线方向 Follow:标记线的方向与已查找到的线的方向保持一致 Vertical:标记线的方向垂直于画布 Horizital:标记线的方向平行于画布
10	距离范围	当两条输出线段的距离在此区间内,产品才会被判定为合格,否则 认为产品不合格
11	角度范围	当两条输出线段的角度在此区间内,产品才会被判定为合格,否则 认为产品不合格

#### 表格 4-10-2: 输出参数

序号	名称	参数说明
1	距离	两条输出线段之间的距离,需要特别注意的是,当输出线段的夹角 大于2°时,将不输出距离值。
2	夹角	两条输出线段之间的夹角
3	中心点和角度1	线段1的中心点和角度,可用于直接输出给机械手
4	中心点和角度2	线段2的中心点和角度,可用于直接输出给机械手
5	线段1	用于显示
6	线段2	用于显示
7	距离线段	用于显示

## 4.11 圆心测量

通过圆环测量工具测量出圆周上的点并拟合出一个圆,同时计算一条圆的 切线。圆心测量工具可用来对圆形物体进行高精度半径测量,精度可达0.02像 素。

#### VISIONGO使用手册





表格 4-11-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	圆环检测框	调整检测框的位置、大小。可通过鼠标在软件界面操作,也可 以直接输入参数。(检测框的颜色为红色)
4	查找模式	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘
5	边缘梯度	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输出 出来
6	直径范围	根据直径测量值是否在直径阈值内判定产品OK/NG

#### 表格 4-11-2: 输出参数

序号	名称	参数说明
1	直径	目标圆的直径



序号	名称	参数说明
2	圆心	目标圆的圆心坐标
3	员	目标圆的位置和大小
4	切线	目标圆的切线标记线的位置

## 4.12 同心度测量

测量两个圆的圆心距离,如下图所示:



#### 表格 4-12-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	圆环检测框1	调整检测框的位置、大小。可通过鼠标在软件界面操作,也可 以直接输入参数。(检测框的颜色为红色)



序号	名称	参数说明
4	查找模式1	设置检测框1的边缘模式。 • Light to dark:查找由亮到暗的边缘 • Dark to light:查找由暗到亮的边缘
5	边缘梯度1	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输出 出来
6	圆环检测框2	调整检测框的位置、大小。可通过鼠标在软件界面操作,也可 以直接输入参数。(检测框的颜色为绿色)
7	查找模式2	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘
8	边缘梯度2	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输出 出来
9	距离范围	两个输出圆的圆心距离范围,当测量出的圆心距离在此范围 内,则认为产品合格,否则认为产品不合格

#### 表格 4-12-2: 输出参数

序号	名称	参数说明
1	圆心距离	两个输出圆的圆心距离
2	圆心连线	两个输出圆的圆心连线
3	圆心1	输出圆1的圆心坐标
4	直径1	输出圆1的直径
5	圆1	输出圆1,主要用于显示
6	圆心2	输出圆2的圆心坐标
7	直径2	输出圆2的直径
8	圆2	输出圆2,主要用于显示



### 4.13 极点测量

极点测量,如下图所示,可用来测量一个极高点到一条线段的垂直距离, 其中线段和点的测量参数可以分别设置,该功能还可用于测量产品边缘是否有 毛刺等,只要测量出极点的存在且与线段的距离超出一定的范围,即认为产品 边缘存在毛刺。



表格 4-13-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	斜矩形检测框	调整检测框的位置、大小和方向。可通过鼠标在软件界面操 作,也可以直接输入参数。(检测框的颜色为红色)
4	线的查找模式	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘
5	线的边缘梯度	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输 出出来
6	点的查找模式	设置检测框1的边缘模式。 • Light to dark:查找由亮到暗的边缘 • Dark to light:查找由暗到亮的边缘



序号	名称	参数说明
7	点的边缘梯度	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会输 出出来
8	距离范围	极点到线段的距离范围,当测量出的实际距离在此范围内, 则认为产品合格,否则认为产品不合格
9	位置偏移量	对查找到的极点进行补偿

#### 表格 4-13-2: 输出参数

序号	名称	参数说明
1	距离	极点到直线的距离
2	垂线	极点到直线的垂线标记线的位置
3	点	极点的坐标
4	线段	目标直线标记线的位置

### 4.14 极距测量



极距测量可用来测量两个极点之间的距离。在很多情况下我们要测量的不 是两条边的距离,而是两个最高点之间的距离,这个时候用距离测量工具无法



实现,因为距离测量工具是通过拟合线然后求取线与线之间的距离。这个时候 我们就可以用到极距测量工具。

表格 4-14-1: 输入参数

序号	名称	参数说明	
1	输入坐标系	选择输入坐标系	
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: ・Default: 默认通道,为CT_R。 ・CT_R: 红色通道 ・CT_G: 绿色通道 ・CT_B: 蓝色通道	
3	组合矩形框	键入工具框的位置、大小和方向	
4	查找模式	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘	
5	边缘梯度1	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会 输出出来	
6	查找模式2	设置检测框1的边缘模式。 ・Light to dark:查找由亮到暗的边缘 ・Dark to light:查找由暗到亮的边缘	
7	边缘梯度2	待测边缘的梯度最小值,只有满足该梯度要求的边缘才会 输出出来	
8	距离范围	两个极点之间的距离范围,当测量出的实际距离在此范围 内,则认为产品合格,否则认为产品不合格	

#### 表格 4-14-2: 输出参数

序号	名称	参数说明
1	点1	极点1的坐标
2	点2	极点2的坐标
3	距离	两个极点间的距离



## 4.15 圆弧测量

圆弧测量,可在圆弧范围内测量半径方向的边缘,如测量时钟的指针位置 等。圆弧测量可同时测量两条边缘,并可同时输出两条边缘的中心线。

表格 4-15-1: 输入参数

序号	名称	参数说明	
1	输入坐标系	选择输入坐标系	
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道	
3	圆弧检测框	键入工具框的位姿	
4	查找模式	设置检测框1的边缘模式。 • Light to dark:查找由亮到暗的边缘 • Dark to light:查找由暗到亮的边缘	
5	边缘梯度	在工具框的大方向上,待测边的灰度梯度	
6	角度范围	根据两条边的夹角测量值是否在角度阈值内判定产品OK/ NG	

表格 4-15-2: 输出参数

序号	名称	参数说明
1	夹角	两条直线的夹角
2	直线1	直线1标记线的位置
3	直线2	直线2标记线的位置
4	角平分线	角平分线标记线的位置
5	角平分线角度	角平分线的角度



## 4.16 阵列测量

阵列测量广泛应用于接插件测量当中,集合基准线,阵列测量可以很容易 地输出诸如共面度的指标。



#### 表格 4-16-1: 输入参数

序号	名称	参数说明
1	输入坐标系	选择输入坐标系
2	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
3	矩形阵列检测框	键入工具框的位置、大小和方向
4	查找模式	设置检测框1的边缘模式。 • Light to dark:查找由亮到暗的边缘 • Dark to light:查找由暗到亮的边缘
5	边缘梯度	在工具框的大方向上,待测边的灰度梯度
6	输出的边缘模式	选择在工具框内查找的直线类型
7	间距范围	根据相邻针脚间距的测量值是否在针脚间距阈值内判定产 品OK/NG
8	点到水平基线的距离范 围	根据点到水平基线的距离测量值是否在点到水平基线距离 阈值内判定产品OK/NG



序号	<del>,</del> 名称	
9	点到垂直基线的距离范 围	根据点到垂直基线的距离测量值是否在点到水平基线距离 阈值内判定产品OK/NG

表格 4-16-2: 输出参数

序号	名称	参数说明	
1	线段	每个ROI查找到的线的集合	
2	间距	相邻针脚间距测量值集合	
3	点到水平基准线的距离	针尖上中心点到水平基线的距离集合	
4	点到垂直基准线的距离	针尖上中心点到垂直基线的距离集合	
5	首条边的位置	第一个ROI找到的线的位置	
6	首条边的角度	第一个ROI找到的线的角度	

### 4.17 图像比对

采用图像相减的方法判断当前图像与模板图像之间的差异。对于彩色图像, 可选择对某个通道单独处理,默认为红色通道。因本工具的特殊性,暂不支持 输入坐标系。

表格 4-17-1: 输入参数

序号	名称	参数说明
1	图像通道	选择所需颜色通道的图像进行操作 取值范围: • Default: 默认通道,为CT_R。 • CT_R: 红色通道 • CT_G: 绿色通道 • CT_B: 蓝色通道
2	正矩形检测框	键入工具框的位置和大小
3	对比方式	选择不同的比较方式 Light:只计算当前图像比模板图像更亮的部分; Dart:只计算当前图像比模板图像更暗的部分; Both:以上两个情况同时计算



序号	名称	参数说明
4	最小灰度差值	图像相减后灰度值小于二值化下限的定义为暗区域,大于二值 化下线的定义为亮区域
5	腐蚀系数	腐蚀系数
6	膨胀系数	
7	最小面积差	用于选出面积大于最小面积差的不同区域

#### 表格 4-17-2: 输出参数

序号	名称	参数说明
1	数量	不同区域的数量
2	面积	
3	轮廓	不同区域的面积集合

## 4.18 边缘检测



## 第五章 脚本(Script)编程

VISIONGO采用Lua作为脚本语言,Lua 是一个小巧的脚本语言。是巴西里 约热内卢天主教大学(Pontifical Catholic University of Rio de Janeiro)里的一 个研究小组,由Roberto lerusalimschy、Waldemar Celes 和 Luiz Henrique de Figueiredo所组成并于1993年开发。其设计目的是为了嵌入应用程序中,从而 为应用程序提供灵活的扩展和定制功能。Lua由标准C编写而成,代码简洁优 美,几乎在所有操作系统和平台上都可以编译,运行。Lua脚本可以很容易的被 C/C++代码调用,也可以反过来调用C/C++的函数,这使得Lua在应用程序中可 以被广泛应用。在目前所有脚本引擎中,Lua的速度是最快的。

### 5.1 脚本编程界面介绍

1.相机设置 2. 检测模块 3. 脚本编程 4. 数字输入输出 5. 串口通讯	6. 网口通讯(TCP/IP) 7. 图像存储	
全局变量 输入列表 输出列表	1	示例代码 7. 网口读写(TCP) ~
[1] Trigger Cmd		1Open a topinParam1: inAddr
[2] Grab Images for One Trigg 1		2Param2: Port
[3] Recive Trigger time 0		3Param3: asServer
[4] Image Index In One Trigg 0		4 michip - ich.open( 192.100.1.1,00000,18
		6Wirte message to tcp
		7Param1: Handle
		9Param3: Buffersize
		10 Param4: Timeout
		11 Buffer = abc 12 Ten Wirte(hTenin Buffer #Buffer 100)
	编译	< >

以上为脚本编程界面,主体分为3个部分,左边红色区主要是变量设置;中间绿色部分是编程区;右边蓝色区是demo区;

左边的红色区域有全局变量、输入列表,输出列表等3个表格。该三个表格 的数据都可以在中间编程区直接使用,使用的方法在右边蓝色区可以找到帮助, 也可以在本文档的章节5.3.6得到帮助。

### 5.2 基本的LUA函数

### 5.2.1 LUA的数学函数库(math)

#### 序号 函数名称

1	abs	取绝对值	math.abs(-15)	15
2	acos	反余弦函数	math.acos(0.5)	1.04719755
3	asin	反正弦函数	math.asin(0.5)	0.52359877
4	atan2	x / y的反正切值	math.atan2(90.0, 45.0)	1.10714871



序号	函数名称			
5	atan	反正切函数	math.atan(0.5)	0.463647609
6	ceil	不小于x的最大整数	math.ceil(5.8)	6
7	cosh	双曲线余弦函数	math.cosh(0.5)	1.276259652
8	COS	余弦函数	math.cos(0.5)	0.87758256
9	deg	弧度转角度	math.deg(math.pi)	180
10	exp	计算以e为底x次方值	math.exp(2)	2.718281828
11	floor	不大于x的最大整数	math.floor(5.6)	5
12	fmod (mod )	取模运算	math.mod(14, 5)	4
13	frexp	把双精度数val分解为数字部 分(尾数)和以2为底的指 数n,即val=x*2n	math.frexp(10.0)	0.625 4
14	ldexp	计算value * 2的n次方	math.ldexp(10.0, 3)	80 = 10 * (2 ^ 3)
15	log10	计算以10为基数的对数	math.log10(100)	2
16	log	计算一个数字的自然对数	math.log(2.71)	0.9969
17	max	取得参数中最大值	math.max(2.71, 100, -9 8, 23)	100
18	min	取得参数中最小值	math.min(2.71, 100, -9 8, 23)	-98
19	modf	把数分为整数和小数	math.modf(15.98)	15 98
20	pow	得到x的y次方	math.pow(2, 5)	32
21	rad	角度转弧度	math.rad(180)	3.14159265358
22	random	获取随机数	math.random(1, 100) math.random(100)	获取1-100的 随机数
23	randomseed	设置随机数种子	math.randomseed(os.ti me())	在使用 math.random 函数之前必须 使用此函数设 置随机数种子
24	sinh	双曲线正弦函数	math.sinh(0.5)	0.5210953
25	sin	正弦函数	math.sin(math.rad(30))	0.5



#### 序号 函数名称

-					
	26	sqrt	开平方函数	math.sqrt(16)	4
	27	tanh	双曲线正切函数	math.tanh(0.5)	0.46211715
	28	tan	正切函数	math.tan(0.5)	0.5463024

### 5.2.2 LUA的字符串函数库(string)

Lua解释器对字符串的支持很有限。一个程序可以创建字符串并连接字符 串,但不能截取子串,检查字符串的大小,检测字符串的内容。在Lua中操纵字 符串的功能基本来自于string库。

字符串库中的一些函数是非常简单的:

1. string.len(s): 返回字符串s的长度;

2. string.rep(s, n): 返回重复n次字符串s的串;

你使用*string.rep("a", 2^20*)可以创建一个*1M bytes*的字符串(比如,为了 测试需要)

3. string.lower(s)函数和string.upper(s)函数用来将字符在大小写之间转换

如果你想不关心大小写对一个数组进行排序的话,你可以这样: table.sort(a, function (a, b) return string.lower(a) < string.lower(b) end)

*string.upper和string.lower*都依赖于本地环境变量。所以,如果你在 *European Latin-1*环境下,表达式: *string.upper("a??o") --> "A??O*"

#### 4. string.sub(s,i,j): 截取字符串s的从第i个字符到第j个字符之间的串。

*Lua*中,字符串的第一个字符索引从1开始。你也可以使用负索引,负索引 从字符串的结尾向前计数:-1指向最后一个字符,-2指向倒数第二个,以此类 推。所以,*string.sub(s, 1, j)*返回字符串*s*的长度为*j*的前缀;*string.sub(s, j, -1)* 返回从第*j*个字符开始的后缀。如果不提供第*3*个参数,默认为-1,因此我们将最 后一个调用写为*string.sub(s, j)*;*string.sub(s, 2, -2)*返回去除第一个和最后一个 字符后的子串。



s = "[in brackets]"

print(string.sub(s, 2, -2)) --> in brackets

记住: Lua中的字符串是恒定不变的。string.sub函数以及Lua中其他的字符 串操作函数都不会改变字符串的值,而是返回一个新的字符串。一个常见的错 误是认为string.sub(s, 2, -2)这个函数会改变字符串s的值。如果你想修改一个字 符串变量的值,你必须将变量赋给一个新的字符串: s = string.sub(s, 2, -2)。

### 5. string.char函数和string.byte函数用来将字符在字符和数字之间转换。

*string.char*获取0个或多个整数,将每一个数字转换成字符,然后返回一个 所有这些字符连接起来的字符串。*string.byte(s, i)*将字符串*s*的第*i*个字符的转换 成整数;第二个参数是可选的,缺省情况下*i=1*。下面的例子中,我们假定字符 用*ASCII*表示:

print(string.char(97)) --> a

*i* = 99; *print(string.char(i, i+1, i+2)) --> cde* 

print(string.byte("abc")) --> 97

print(string.byte("abc", 2)) --> 98

print(string.byte("abc", -1)) --> 99

上面最后一行,我们使用负数索引访问字符串的最后一个字符。

6. string.format()函数来生成具有特定格式的字符串。

函数的第一个参数是格式(formatstring), 之后是对应格式中每个代号的各种数据. 由于格式字符串的存在, 使得产生的长字符串可读性大大提高了. 这个函数的格式很像C语言中的printf().函数string.format在用来对字符串进行格式化的时候, 特别是字符串输出, 是功能强大的工具。这个函数有两个参数, 你完全可以照C语言的printf来使用这个函数。第一个参数为格式化串: 由指示符和控制格式的字符组成。指示符后的控制格式的字符可以为: 十进制'd'; 十六进制'x'; 八进制'o'; 浮点数'f'; 字符串's'。在指示符 % 和控制格式字符之间还可以有其他的选项: 用来控制更详细的格式, 比如一个浮点数的小数的位数:

格式字符串可能包含以下的转义码:

%c-接受一个数字,并将其转化为ASCII码表中对应的字符

%d, %i-接受一个数字并将其转化为有符号的整数格式

%o-接受一个数字并将其转化为八进制数格式 %u-接受一个数字并将其转化为无符号整数格式 %x-接受一个数字并将其转化为十六进制数格式,使用小写字母 %X-接受一个数字并将其转化为十六进制数格式,使用大写字母 %e-接受一个数字并将其转化为科学记数法格式,使用小写字母e %E-接受一个数字并将其转化为科学记数法格式,使用大写字母E %f-接受一个数字并将其转化为浮点数格式

*%g(%G)* - 接受一个数字并将其转化为*%e(%E,*对应*%G)*及*%f*中较短的一种格式

%q-接受一个字符串并将其转化为可安全被Lua编译器读入的格式

%s-接受一个字符串并按照给定的参数格式化该字符串

为进一步细化格式,可以在%号后添加参数.参数将以如下的顺序读入:

(1) 符号: 一个+号表示其后的数字转义符将让正数显示正号. 默认情况下只 有负数显示符号.

(2) 占位符:一个0, 在后面指定了字串宽度时占位用. 不填时的默认占位符是 空格.

(3) 对齐标识:在指定了字串宽度时,默认为右对齐,增加-号可以改为左对齐.

(4) 宽度数值

(5) 小数位数/字串裁切: 在宽度数值后增加的小数部分n, 若后接f(浮点数转 义符, 如%6.3f)则设定该浮点数的小数只保留n位, 若后接s(字符串转义符, 如 %5.3s)则设定该字符串只显示前n位.

在这些参数的后面则是上述所列的转义码类型(*c*, *d*, *i*, *f*, …). print(string.format("pi = %.4f", PI)) —> pi = 3.1416 d = 5; m = 11; y = 1990 print(string.format("%02d/%02d/%04d", d, m, y)) —> 05/11/1990 tag, title = "h1", "a title"

print(string.format("<%s>%s</%s>", tag, title, tag)) —> <h1>a title</

第一个例子,%.4f代表小数点后面有4位小数的浮点数。第二个例子%02d 代表以固定的两位显示十进制数,不足的前面补0。而%2d前面没有指定0,不 足两位时会以空白补足。对于格式串部分指示符得详细描述清参考lua手册,或 者参考C手册,因为Lua调用标准C的printf函数来实现最终的功能。

以下是一些例子:

输出	C

string.format("%%c: %c", 83)	输出 <i>S</i>
string.format("%+d", 17.0)	输出 <i>+17</i>
string.format("%05d", 17)	输出 <i>00017</i>
string.format("%o", 17)	输出 <i>21</i>
string.format("%u", 3.14)	输出 <i>3</i>
string.format("%x", 13)	输出 <i>d</i>
string.format("%X", 13)	输出 <i>D</i>
string.format("%e", 1000)	输出 <i>1.000000e+03</i>
string.format("%E", 1000)	输出1.000000E+03
string.format("%6.3f", 13)	输出 <i>13.000</i>
string.format("%q", "One\nTwo")	输出 <i>"One\Two"</i>
string.format("%s", "monkey")	输出 <i>monkey</i>
string.format("%10s", "monkey")	输出 <i>monkey</i>
string.format("%5.3s", "monkey")	输出 <i>mon</i>

## 5.2.3 LUA的条件函数(If)

```
1. if ... else语句
```

```
if(boolean_expression)
```

then

--[ statement(s) will execute if the boolean expression is true --] else

--[ statement(s) will execute if the boolean expression is false --] end



如果布尔表达式的值为*true*,那么*if*代码块将被执行,否则*else*代码块将被执行。

Lua程序设计语言假定布尔true和非零值的任意组合作为true,以及它是否 是布尔假或零,则假定为false值。但应当注意的是,在Lua零值被视为true。

```
--[ local variable definition --]
a = 100;
--[ check the boolean condition --]
if( a < 20 )
then
    --[ if condition is true then print the following --]
    print("a is less than 20" )
else
    --[ if condition is false then print the following --]
    print("a is not less than 20" )
end
print("value of a is :", a)</pre>
```

当建立和运行上面的代码,它会产生以下结果。 a is not less than 20 value of a is: 100

### 2. if...else if...else 语句

*if*语句后面可以跟一个可选的*else if … else*语句,这是非常有用的使用,以 测试各种条件单个*if...else if* 语句。

当使用if, else if, else语句有几点要记住使用:

·if 可以有零或一个 else,但必须在elseif之前。

·if 之后可以有零到很多else if在else之前。

•一旦一个else if成功,其它的elseif将不会被测试。

if(boolean\_expression 1)

then

--[ Executes when the boolean expression 1 is true --]



else if( boolean\_expression 2)

--[ Executes when the boolean expression 2 is true --]

else if( boolean\_expression 3)

--[ Executes when the boolean expression 3 is true --]

else

--[ executes when the none of the above condition is true --] end

### 5.2.4 LUA的循环函数(for/while)

#### 1. for循环

for循环是一个循环控制结构,可以有效地编写需要执行的特定次数的循环。

Lua for语句语法如下:

for init, max/min, increment

do

statement(s)

end

下面是控制在一个循环的流程:

·初始化步骤首先被执行,并且仅一次。这个步骤可让您声明和初始化任何循环控制变量。

• 接着是*max/min*,这是最大或最小值,直到该循环继续执行。它在内部创建了一个条件检查的初值和最大值/最小值之间进行比较。

• for循环体执行后,控制流跳回至递增/递减声明。这个语句可以更新任何 循环控制变量。

·条件现在重新计算评估。如果这为真则循环执行,并重复这个过程(循环体,然后增加一步,然后再条件)。如果条件为假,则循环终止。

### 2. while循环语句

在Lua编程语言中的while循环语句,只要给定的条件为真时将重复执行的目标语句。

Lua while语句语法如下:



while(condition)

do

statement(s)

end

在这里,声明(S)可以是单一语句或语句块。该条件可以是任何表达式,并 且真正是任意非零值。循环迭代当条件是*true*。

当条件为假,则程序控制进到紧接在循环之后的一行。

### 3. break语句

当循环中遇到break语句,循环立即终止,程序控制继续下一个循环语句后面。

如果您正在使用嵌套循环(即一个循环里面另一个循环), break 语句将停止 最内层循环的执行并开始执行的下一行代码的程序后段。

Lua break语句语法如下:

break

## 5.3 自定义的关键词及函数

### 5.3.1 项目及函数操作

序 号	天键 词	支持的函数	说明	示例
1	prj	GetResult()	获取当前项目的检测结果(Script的 结果除外)	flag = prj.GetResult()
		SetResult(flag)	设置当前项目的检测结果	prj.SetResult(true)
		GetResult(fxnld)	获取指定函数的检测结果。 参数说明: 1. fxnld:函数的lD	flag = fxn.GetResult(0)



序 号	关键 词	支持的函数	说明	示例
2	fxn	SetResult(fxnld,fla g)	设置指定函数的检测结果。 参数说明: 1. fxnld:函数的ID 2. flag:要设置的结果, true或者 false	fxn.SetResult(0,true)
		SetUsingFlag(fxnld ,flag)	设置指定函数是否启用。默认所有 的函数都是启用状态;禁用状态的 函数不会执行。 参数说明: 1. fxnld:函数的ID 2. flag:要设置的结果, true或者 false	fxn.SetUsingFlag(0,fa Ise)
3	ilk	GetPtr(fxnld, iLinkld)	获取指定函数输入端口的指针。 参数说明: 1. fxnld:函数的ID 2. iLinkld:输入端口的序号	iptr = ilk.GetPtr(0,0)
4	olk	GetPtr(fxnld, oLinkld)	获取指定函数输出端口的指针。 参数说明: 1. fxnld:函数的ID 2. oLinkld:输出端口的序号	optr = olk.GetPtr(0,0)

## 5.3.2 变量操作

1. Get函数

序号	关键词	示例
1	Bool(布尔数据)	val = Bool.Get(optr)
2	Long(整型数据)	val = Long.Get(optr)
3	Real(浮点数据)	val = Real.Get(optr)
4	Str(字符串)	val = Str.Get(optr)
5	RealPoint(点)	x, y = RealPoint.Get(optr)
6	Line(线段)	x, y, length, angle = Line.Get(optr)



序号	关键词	示例
7	Circle(圆)	x, y, radius = Circle.Get(optr)
8	XYA(位置与方向)	x, y, a = XYA.Get(optr)
9	Rect2(斜矩形)	x, y, len1, len2, angle = Rect2.Get(optr)
10	Array(数组)	for i = 1,#optr do b = Bool.get(optr[i]) end

## 2. New函数

序号	关键词	支持的函数	示例
1	Bool(布尔数据)	New(val)	val = Bool.New(true)
2	Long(整型数据)	New(val)	val = Long.new(100)
3	Real(浮点数据)	New(val)	val = Real.new(123.45)
4	Str(字符串)	New(val)	val = Str.new("abc")
5	RealPoint(点)	New(x, y)	val = RealPoint.new(123.45,147.85)
6	Line(线段)	New(x, y, length, angle)	val = Line.new(10.0,20.0,30.0,0.0)
7	Circle(圆)	New(x, y, radius)	Circle.new(10.0,20.0,30.0)
8	XYA(位置与方 向)	New(x, y, a)	XYA.new(10.0,20.0,30.0)
9	Rect2(斜矩形)	New(x, y, len1, len2, angle)	Rect2.new(10.0,20.0,30.0,40.0,0.0)
10	Array(数组)	New(name, size, initVal)	Array.new("array1", 10,Bool.new(false))



# *5.3.3 DI/O*操作(Dio)

序号	支持的函数	说明	
1	Open(type, cardIdx)	打开指定IO卡并返回该IO卡的句柄,如 果该IO卡已经打开,则直接返回句柄。 参数说明: 1. type: IO卡的,取值范围 为"PCI7230","Comrelay"; 2. cardIdx: IO卡的编号,为从0开始的 整数值,取值为0时表示第1张IO卡, 取值为1时表示第2张IO卡。	hDio = Dio.Open("Comrelay", 3)
2	Close(hDio)	关闭指定的IO卡。因为VISIONGO会自 动管理所有的IO卡,所以大部分情况下 不需要调用该函数。 参数说明: 1. hDio: IO卡的句柄,可从Open函数 的返回值中获得。	Dio.Close(hDio)
3	GetDINum(hDio)	获取指定IO卡的输入端口数量 参数说明: 1. hDio:IO卡的句柄,可从Open函数 的返回值中获得。	Dio.GetDINum(hDio)
4	GetDONum(hDio)	获取指定IO卡的输出端口数量 参数说明: 1. hDio:IO卡的句柄,可从Open函数 的返回值中获得。	Dio.GetDONum(hDio)
5	GetDIStatus(hDio , port)	获取IO卡某个输入端的状态 参数说明: 1. hDio: IO卡的句柄,可从Open函数 的返回值中获得。 2. port: 输入端口号,为从0开始的整 数	status = Dio.GetDIStatus(hDio, 0)
6	GetDOStatus(hDi o, port)	获取IO卡某个输入端的状态 参数说明: 1. hDio: IO卡的句柄,可从Open函数 的返回值中获得。 2. port: 输入端口号,为从0开始的整 数	status = Dio.GetDOStatus(hDio ,0)



序号	支持的函数	说明	
7	SetDOStatus(hDi o, port, status)	获取IO卡某个输入端的状态 参数说明: 1. hDio:IO卡的句柄,可从Open函数 的返回值中获得。 2. port:输入端口号,为从0开始的整 数	Dio.SetDOStatus(hDio ,0)

### DI/O示例

hDio = DIO.Open("Comrelay",3)	打开连接到COM3的Comrelay卡
iNum = DIO.GetDINum(hDio)	获取该IO卡的输入总数
for i=0, iNum,1 do	循环开始
status = DIO.GetDIStatus(hDio, 0)	获取0号输入口的状态
if(status) then	判断是否0号输入口为高电平
DIO.SetDOStatus(hDio, 0, true)	设置0号输出为高电平
end	条件判断结束
end	循环结束

# 5.3.4 串口操作(Uart)

序 号	支持的函数	说明	示例
1	Open(port, baudrate, parity, wordlen, stopbits)	打开串口并返回该串口的句柄,如果该串口 已经打开,则直接返回句柄。 参数说明: 1.port:串口号,"com1","com2",, 等 2.baudrate:波特率。 3.parity:串口极性 4.wordlen: 5.stopbits:停止位	hUart = Uart.Open("com1", 9600, 0, 8, 1)



序 号	支持的函数	说明	示例
2	Close(hUart)	关闭串口。因为VISIONGO会自动管理所有 的串口,所以大部分情况下不需要调用该函 数。 参数说明: 1.hUart:串口句柄,可从Open函数的返回 值中获得。	Uart.Close(hUart)
3	Write(hUart, buffer, buffersize, timeout)	<ul> <li>发送串口消息。</li> <li>参数说明:</li> <li>1. hUart:串口句柄,可从Open函数的返回 值中获得;</li> <li>2. buffer:需要发送的字符串;</li> <li>3. buffersize:待发送字符串的长度;</li> <li>4. timeout:超时时间,单位为毫秒;</li> </ul>	Uart.Write(hUart, "Visiongo", 8, 100)
4	Read(hUart, readsize, timeout)	读取串口消息。返回读取到的字符串 参数说明: 1.hUart:串口句柄,可从Open函数的返回 值中获得; 2.readsize:需要接收的字符串长度; 3.timeout:超时时间,单位为毫秒;	buffer = Uart.Read(hUart, 5,100)
5	Clear(hUart)	清除串口缓存 参数说明: 1.hUart:串口句柄,可从Open函数的返回 值中获得。	Uart.Clear(hUart)

## 5.3.5 网口操作(Tcp)

序号	支持的函数	说明	示例
1	Open(ip, port, serverFlag)	建立网口连接并返回该连接的句柄,如野 该网口已经建立,则直接返回句柄。 参数说明: 1. ip: ip地址 2. port: 端口号。 3. serverFlag: 是否创建服务器	₹ hTcp = Tcp.Open("192.168.1.1", 6001,false)



VISIONGO使用手册

序号	支持的函数	说明	示例
2	Close(hTcp)	关闭网络连接。因为VISIONGO会自动管 理所有的网络连接,所以大部分情况下不 需要调用该函数。 参数说明: 1.hTcp:网口句柄,可从Open函数的返 回值中获得。	Tcp.Close(hTcp)
3	Write(hTcp, buffer, buffersize, timeout)	发送网口消息。 参数说明: 1.hTcp:网口句柄,可从Open函数的返 回值中获得。 2.buffer:需要发送的字符串; 3.buffersize:待发送字符串的长度; 4.timeout:超时时间,单位为毫秒;	Buffer = "abc" Tcp.Wirte(hTcp,Buffer,#B uffer,100)
4	Read(hTcp, readsize, timeout)	读取网口消息。返回读取到的字符串。 参数说明: 1. hTcp:网口句柄,可从Open函数的返 回值中获得。 2. readsize:需要接收的字符串长度; 3. timeout:超时时间,单位为毫秒;	Buffer = Tcp.Read(hTcp, 5,100)
5	Clear(hTcp)	清除串口缓存。 参数说明: 1. hTcp:网口句柄,可从Open函数的返 回值中获得。	Tcp.Clear(hTcp)

# 5.3.6 脚本输入输出操作(Script)

序 号	支持的函 数	说明	示例
1	GetInput( port)	获取脚本函数的输入,该输入一般在IVU中由用 户设置。 参数说明: 1.port:输入端口号。	Script.GetInput(0)



序 号	支持的函 数	说明	示例
2	SetOutput (port,val)	设置脚本函数的输出,该输出可在IVU中查看; 需要注意的是,因为输出时通过其它界面设置 的,该函数只是赋值,因此需要注意值的类型 与输出的类型是否一致。 参数说明: 1.port:输入端口号。 2.val:新的输出值	Script.SetOutput(0,1 00)
3	GetGlobal Var(port)	获取全局变量的值。 参数说明: 1. port:全局变量的编号。	Script.GetGlobalVar( port)
4	SetResult( flag)	获取脚本函数的执行结果,默认脚本函数的执 行结果为true,如果需要根据条件设置执行结 果,需调用该函数。 参数说明: 1.flag:true或者false。	Script.SetResult(true )