

埃夫特智能装备股份有限公司

固定视觉说明书

版本号: v0.1

软件二组

2019-6-30

修订历史

[illegible]

目录

1 固定视觉功能介绍	1
2 固定视觉 App 界面介绍	1
2.1 固定视觉主界面	1
2.2 固定视觉设计界面	2
2.3 手眼标定界面	3
3 固定视觉的标定及测试用例	4
3.1 相机标定	4
3.1.1 设置界面参数设定及相机标定	4
3.1.2 相机标定测试用例	4
3.2 手眼标定	6
3.2.1 设置界面参数设定及手眼标定	6
3.2.2 测试标定结果和测试用例	9

1 固定视觉功能介绍

视觉功能是指机器人与视觉系统通过 TCP/IP 协议进行通讯，视觉系统作为服务器，机器人作为客户端，视觉系统将获取的基于视觉系统坐标下物体的位置信息转化成机器人坐标下的位置，从而实现机器人按指定轨迹运动。

固定视觉是指相机安装在固定台架上，拍摄的物体在固定的工作台面上。

2 固定视觉 App 界面介绍

2.1 固定视觉主界面

点击  进入桌面，点击桌面上的 ，进入固定视觉 APP 主界面，如下图 2-1 所示：



图 2-1 固定视觉 APP 主界面

图中红框内信息说明如下：

1．固定视觉开关：用来是否开启控制器中的固定视觉功能；

2．TCP/IP 连接状态：表示当前机器人与相机通讯的连接状态， 表示当前处于断开

状态， 表示当前处于连接状态；

3．固定视觉坐标系：用于显示手眼标定成功后的标定结果；

4．在机器人坐标系下工件的位置：在相机标定模式或手眼标定模式下用于显示拍照后

工件在机器人坐标系下的位置；

5. 在相机坐标系下工件的位置：在相机上进行手眼标定后用于显示拍照后工件在相机坐标系下的位置；

6. 工件属性和 ID：用于显示工件的信息；

7. 相机触发指令、拍照间隔和拍照按钮：相机触发指令，当相机为指令触发模式时，输入相机内设置的拍照指令（该指令只能为 int 型）；拍照间隔用于设定测试时拍照的时间间隔（即：当触发相机拍照后，机器人获取相机数据的最长时间，该值设置不能低于 600 毫秒）；拍照按钮用于触发相机拍照；

8. 设置、标定和退出按钮：设置按钮用于从主页面切换到 TCP/IP 的设置界面；标定按钮用于从主页面切换到手眼标定页面；退出按钮用于退出固定视觉 APP。

2.2 固定视觉设计界面

点击固定视觉主界面的“设置”按钮，进入设置界面，设置界面如图 2-2 所示：

相机连接设置

相机品牌：通用相机

相机IP地址：192.168.1.103

相机登录帐号：admin

相机端口：1234

登录密码：*

拍照触发设置

相机触发方式：IO触发

标定方式：相机标定

拍照时间间隔：5000 ms

相机命令触发：sw8\r\n

相机触发IO口：7

相机数据获取指令：gvi030\r\n

保存 断开连接 返回

如图 2-2 固定视觉设置界面

图中红框内信息说明如下：

相机连接设置：

1. 相机品牌：目前相机品牌可选通用相机、康耐视相机和麦克玛视相机；当选择康耐视相机时，需要输入相机账号和密码；

2. 相机登录账号：当选择康耐视相机时，需要输入相机上设置好的登录账号；

3. 登录密码：当选择康耐视相机时，需要输入相机上设置的登录密码；

4. 相机 IP 地址：需要输入相机上设置的 IP 地址；

5. 相机端口：需要输入相机的端口号；

拍照触发设置：

6. 相机触发方式：有指令触发和 IO 触发两种方式；当选择指令触发时，相机触发 IO 呈灰色，不可设置；

7. 拍照时间间隔：用来设置拍照的时间间隔（与主界面相同）；
8. 相机触发 IO：当相机触发方式选择为 IO 触发时，可设置触发 IO 的地址；
9. 标定方式：有相机标定和手眼标定两种方式；相机标定是在相机上完成的；
10. 相机触发指令：在相机设置好命令触发方式后可在此处进行设置（与主界面相同）；
11. 相机数据获取指令：此设置只针对康耐视相机，当登录康耐视相机后，此设置才生效；
12. 保存、连接、返回按钮：当所有的设置完成后，须点击“保存”按钮，会将设置信息保存到控制器中；点击“连接”后，机器人会跟相机进行连接，连接成功后，界面右上角的状态会变成绿色；返回到主界面时须点击“返回”按钮。

2.3 手眼标定界面

在设置界面将标定方式选为“手眼标定”，点击“保存”按钮，然后返回到主界面，点击“标定”按钮，进入标定界面，如图 2-3 所示：

图中红框内信息说明如下：

1. 标定点点数：目前可选标定点的个数为：3~6；
2. 相机：当点击“示教”按钮时，显示工件在相机坐标系下的值；
3. 机器人：当点击“示教”按钮时，显示机器人当前的坐标值；
4. 示教、清除：工件在相机下的操作，当点击“示教”按钮时，记录工件在相机坐标系下的值；当点击“清除”按钮时，将当前显示的值清零；
5. 示教、清除：工件在机器人下的操作，其作用同上；
6. 当前点的切换：当按下 ，可以切换到上一点；当按下 ，可以切换到下一点；

示教点状态：

7. 相机：相机坐标系下，示教点的示教状态； 表示点未示教； 表示点已示教；
8. 机器人：机器人坐标系下，示教点的示教状态； 表示点未示教； 表示点已示教；
9. 计算、返回：当所有点示教完成后点击“计算”按钮，其计算结果会在主界面的固定视觉坐标系中显示；点击“返回”，返回主界面。

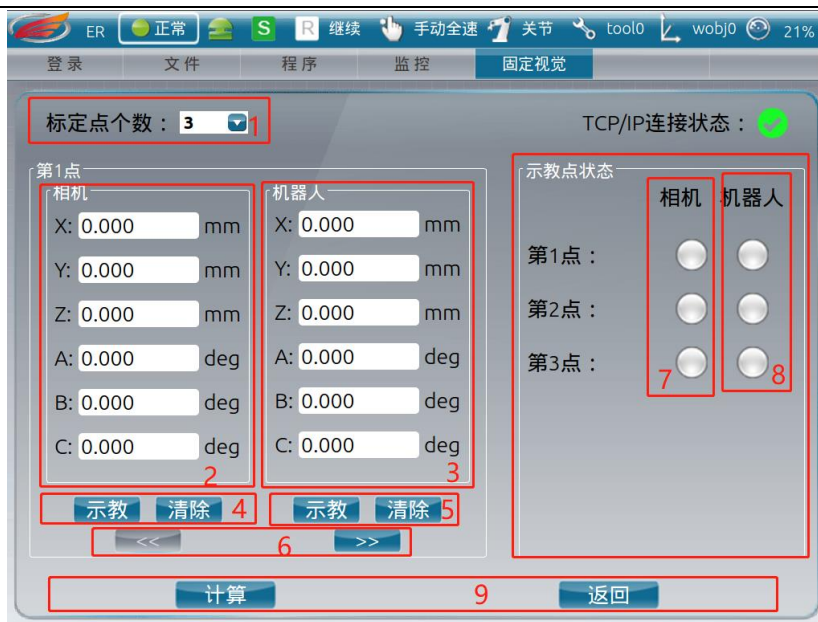


图 2-3 固定视觉标定界面

3 固定视觉的标定及测试用例

固定视觉的标定方式：相机标定和手眼标定。

3.1 相机标定

3.1.1 设置界面参数设定及相机标定

进入设置界面，选择使用的相机品牌，完成相关设置，将标定方式选为相机标定，点击“保存”，然后点击“连接”按钮与相机进行连接，连接上后返回到主界面，如图 1-4 所示。相机的标定在相机软件上完成，具体标定操作流程请根据相机提供的标定流程进行。在相机上标定完成后，进行拍照测试。

3.1.2 相机标定测试用例

在相机标定的主界面中，点击“拍照”按钮，工件在机器人坐标系下的值会刷新。



图 3-1 相机标定的主界面

固定视觉 rpl 指令说明:

指令	名称	功能
Vision._init_()	视觉功能初始化命令	该命令视觉相关功能的初始化, 在正常使用中不需要调用, 该函数会在程序开始自动运行
Vision.getData()	相机拍照命令	调用该命令, 触发相机拍照动作并返回相应数据
Vision.setTrigCmd(int p)	设置相机触发指令	相机在指令触发的模式下, 该命令可设置相机触发的指令
变量	名称	
Vision.x real	工件位置: X 方向坐标	
Vision.y real	工件位置: Y 方向坐标	
Vision.z real	工件位置: Z 方向坐标	
Vision.a real	工件姿态: 绕 Z 轴角度	
Vision.b real	工件姿态: 绕 Y 轴角度	
Vision.c real	工件姿态: 绕 X 轴角度	
Vision.attr int	工件属性	
Vision.id int	工件 ID	

固定视觉 rpl 程序用例 (相机标定模式):

```
1 LABEL a :  
2 MJOINT (*, v500, fine, tool1) ;  
3 vision.setTrigCmd(1) ;  
4 hasObj := vision.getData() ;  
5 IF hasObj THEN  
6     hight := 400 ;  
7     point1pick := POINTC(vision.x, vision.y, hight, vision.a, 180, 0) ;  
8     point1 := POINTC(vision.x, vision.y, hight + 50, vision.a, 180, 0) ;  
9     MJOINT (point1, v500, fine, tool1) ;  
10    MJOINT (point1pick, v500, fine, tool1) ;  
11    DWELL (5) ;  
12    MJOINT (point1, v500, fine, tool1) ;  
13 END_IF ;  
14 GOTO a ;
```

Line1:循环开始;

Line2:机器人运动到工位 1;

Line3:设置相机触发指令为 1; (当相机设置为 IO 触发时, 不需要该步骤)

Line4:触发相机拍照, 当成功获取数据时变量 hasObj=true;

Line5:如果成功获取相机数据, 则执行 Line6-12;

Line6:定义工件高度位置补偿值 hight

Line7:定义工位 2 (抓取点) ;

Line8:定义工位 3 (抓取点上方位置) ;

Line9:机器人运动到工位 3;

Line10:机器人向下运动到工位 2;

Line11:等待抓取工件;

Line12:机器人向上运动到工位 3;

Line14:循环结束;

注意: 因为在相机端做手眼标定, 所以 vision.x,vision.y,vision.a 三个工件数据结果可以直接应用, vision.z,vision.b,vision.c, 需要根据现场实际情况进行数值上的补偿。

3.2 手眼标定

3.2.1 设置界面参数设定及手眼标定

相机安装是, 相机镜头平面需要尽量与平台的平面平行, 相机坐标系的 Z 轴方向需要竖直向上; 目前这个版本固定视觉只支持抓取点与相机识别点相同的情况。

进入设置界面, 选择使用的相机品牌, 完成相关设置, 将标定方式选为手眼标定, 点击“保存”, 然后点击“连接”按钮与相机进行连接, 连接上后返回到主界面, 如图 3-2 所示。



图 3-2 机器人手眼标定的主界面

点击主界面上的“标定”按钮，进入标定界面，如图 3-3 所示，以 3 点标定为例，进行机器人手眼标定。在标定前，选择所需的工具坐标系，默认的工具坐标系为 tool0。在标定时，将工件放在图 3-4 所示第一点的位置，点击相机下的“示教”按钮，此时会相机下的坐标值显示框中显示工件在相机坐标系下的坐标值，然后将机器人移动到工件的位置，点击机器人下的“示教”按钮，在机器人下的坐标值显示框中显示当前工件在机器人坐标系下的位置值，如果示教成功，则在右侧示教点的状态中相应的状态灯会变成橙色。根据图 3-4 所示的 3 点位置，依次示教完后，如图 3-5 所示，点击“计算”按钮。标定完后点击“返回”按钮，进入主界面。标定结果显示在固定视觉坐标系中，如图 3-6 所示，点击“拍照”按钮，会刷新工件在相机坐标系和机器人坐标系下的位置值。可以根据现场需求设置相机拍照时间间隔（注意：拍照的时间间隔不能小于 600 毫秒）。



图 3-3 手眼标定界面

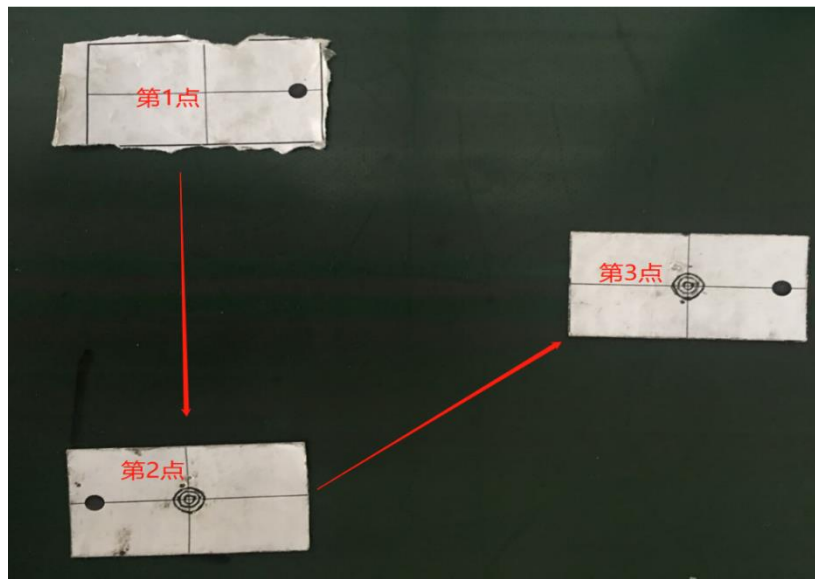


图 3-4 三点标定工件放置的位置



图 3-5 三点示教完成界面



图 3-6 手眼标定后的主界面

3.2.2 测试标定结果和测试用例

编写一个测试程序，如下：

```

1 LABEL a :
2 MJOINT (*, v500, fine, tool1) ;
3 vision.setTrigCmd(1) ;
4 hasObj := vision.getData() ;
5 IF hasObj THEN
6     point1pick := POINTC(vision.x, vision.y, vision.z + 5, -180 - vision.a, 180, 0) ;
7     point1 := POINTC(vision.x, vision.y, vision.z + 50, -180 - vision.a, 180, 0) ;
8     MJOINT (point1, v500, fine, tool1) ;
9     MJOINT (point1pick, v500, fine, tool1) ;
10    DWELL (5) ;
11    MJOINT (point1, v500, fine, tool1) ;
12 END_IF ;
13 GOTO a ;
    
```

Line1:循环开始；

Line2:机器人运动到工位 1；

Line3:设置相机触发指令为 1；（当相机设置为 IO 触发时，不需要该步骤）

Line4:触发相机拍照，当成功获取数据时变量 hasObj=true；

Line5:如果成功获取相机数据，则执行 Line6-11；

Line6:定义工位 2（抓取点）；

Line7:定义工位 3（抓取点上方位置）；

Line8:机器人运动到工位 3；

Line9:机器人向下运动到工位 2；

Line10:等待抓取工件；

Line11:机器人向上运动到工位 3;

Line13:循环结束;

注意：因为在机器人端做手眼标定，所以 vision.x,vision.y,vision.z 三个工件位置数据结果可以直接应用，vision.a,vision.b,vision.c，需要根据现场实际情况进行数值上的补偿。

根据测试时机器人的抓取点来判断标定的准确性。