双 翌
SHUANGYI

MicroDistPro 开发手册

深圳市双翌光电科技有限公司

修订记录:

Rev	Data	Author	Description
1.0	20140530	ShuangYi	创建初始文档
1.1	20140610	ShuangYi	增加标定信息函数 SY_MD_Camera_Info 增加 4 相机对位函数 SY_MD_4Camera_Align
1.2	20170627	ShuangYi	删除部分旧内容
2.0	20171204	ShuangYi	增加图示说明，丰富文档
2.1	20180710	ShuangYi	增加直线对位、特殊对位内容
2.2	20200709	ShuangYi	匹配库文件 20200609 版本
3.0	20230110	ShuangYi	增加不映射对位函数、长度检测函数
3.0	20231204	ShuangYi	增加 SY_MD_UVW_Table_Control 说明

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

目　　　录

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

简介

MicroDistPro 对位算法是深圳市双翌光电科技有限公司专门针对高精度对位需求而自主研发的软件产品算法包，MicroDistPro 是一款全功能的视觉对位算法包，提供了原型化的对位场景应用，用户可以灵活、快速简易地搭建自己的对位算法应用。

MicroDistPro 视觉对位算法助您快速构建机械手智能装配、机械手上下料、全贴合、偏光片贴合、丝网印刷、COG、FOG、玻璃切割等行业应用。算法应用广泛、具有强大的兼容性，使得工程人员有更多时间和精力来开发自己的产品。

免责声明[Disclaimer]

为了改进产品的可靠性、设计和功能，本文档中的信息如有更改，恕不另行通知，且本文档中的信息并不代表制造商所作的承诺。若因产品或文档使用不当而造成的直接、间接、特殊、意外或从属损坏（即使已告知可能造成这种损坏），制造商将不承担任何责任。

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

第 1 章 MicroDistPro 开发说明

软件支持函数库

MicroDistPro 对位算法支持 Windows7、Windows10 等 32/64 位操作系统，并提供完整的函数库与动态链接库(DLL)，用户可以轻松完成其应用程序。

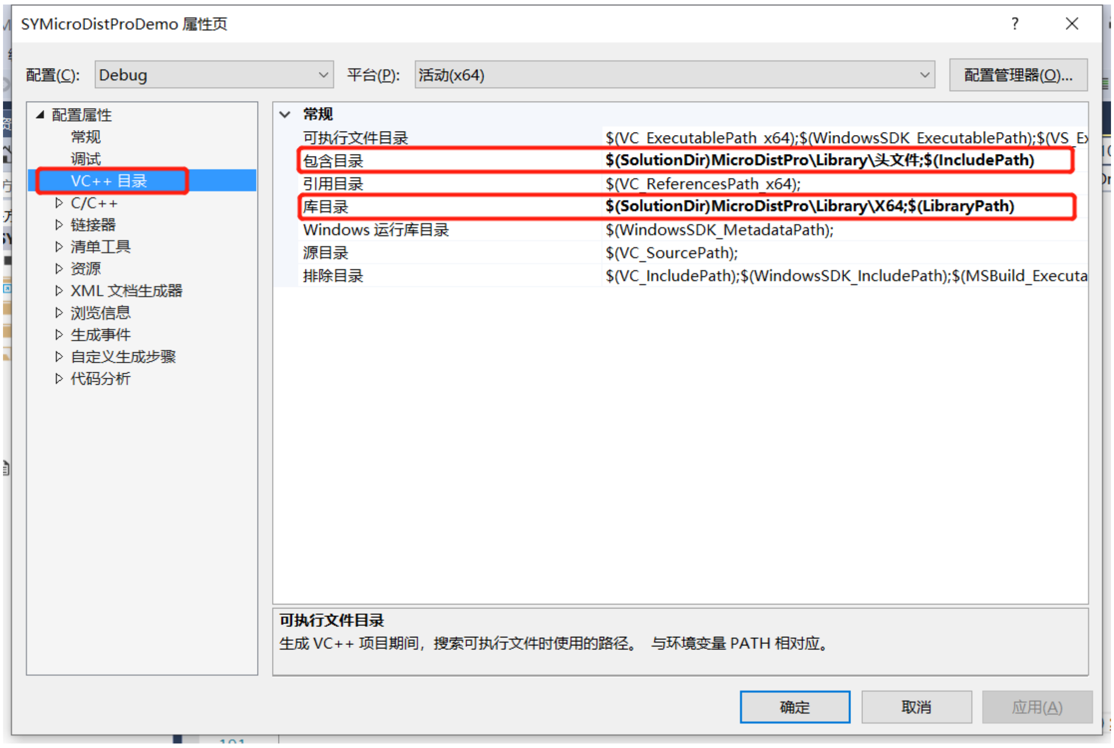
支持 IDE(Integrated Development Environment,集成开发环境)

开发语言	Using......	操作系统(OS)	开发环境	版本	MicroDistPro 版本
C++	C++ classes	Windows 7	MS Visual Studio C++	15.0	支持

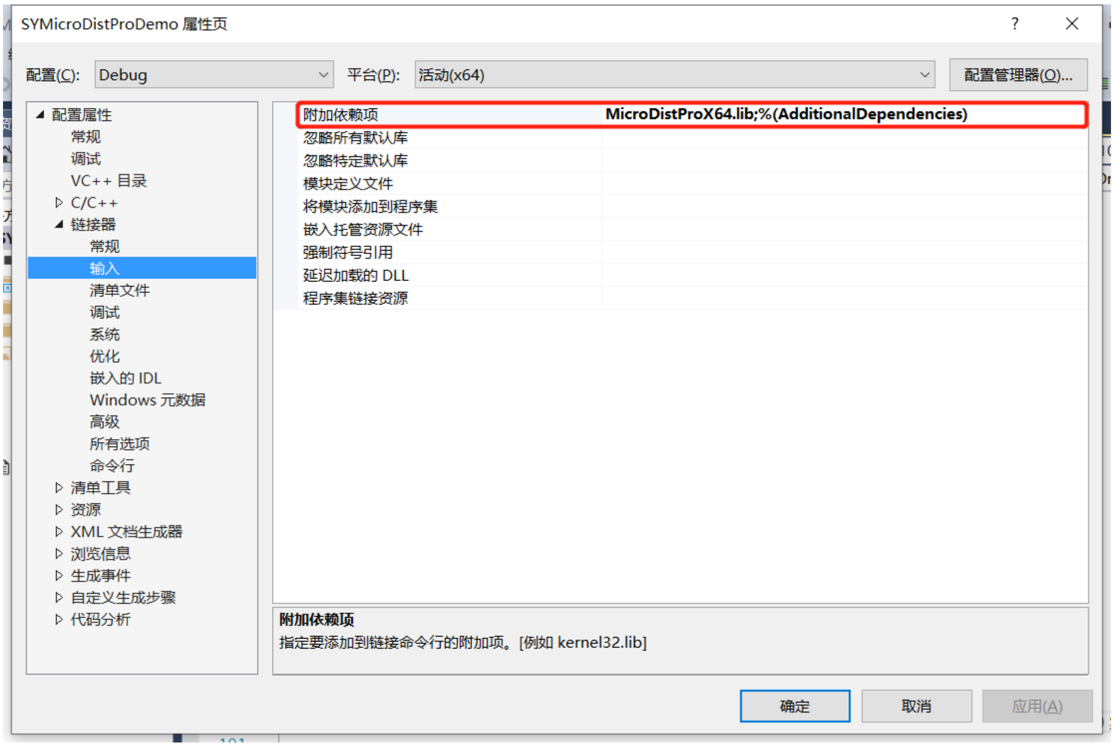
编码:Unicode 编码方式

1.1 函数库使用

在 Windows 系统下，用户可以使用支持动态链接库的开发工具来开发应用程序。
使用 Visual Studio 2010 来开发图像处理流程：
【1】 启动 Visual Studio 2010， 新建一个工程；
【2】 将安装目录的 Library 文件夹内 dll 文件、 h 文件和 lib 文件复制到工程文件夹中；
【3】 选择“Project” 菜单下的“Properties…” 菜单项；
【4】选择“VC++目录”，设置包含目录为.h 文件的目录位置，库目录为.lib 文件的目录位置。



【5】切换到“链接器/输入”标签页，在“附加依赖项”栏中输入 lib 文件名 MicroDistProX64.lib（32 位使用 MicroDistProX86.lib），如下图所示。



【6】在应用程序文件中加入函数库头文件的声明， #include ″MicroDistPro.h″；至此，用户就可以在Visual C++中调用库中的函数开发定位应用程序。如下图所示。

```
// SYMicroDistProDemoDlg.h : header file
//
#include "MicroDistPro.h"
#include "SYdef.h"
#include "SYIntegrationClass.h"
#pragma once
```

【注意】另外，请将hast_rt.exe放在应用程序.exe同一目录下，否则会报0XC000044错误。

【注意】应将“MicroDistProX64.dll”（32 位使用 MicroDistProX86.dll）文件和建立的应用程序放在同一目录内或者系统文件目录（一般如：c:\windows\system32）下，否则会出现找不到 dll 文件的提示。

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

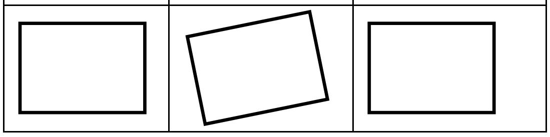
1.2 应用指南

1.2.1 对位方式

常见的对位模式有以下方式

(1)对位方式 1：对位目标固定

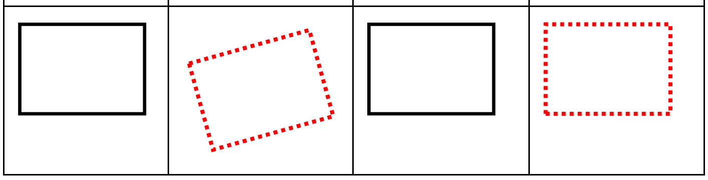
设定对位目标位置后，每次都与目标位置比较，计算出位置偏差，使对准目标位置。

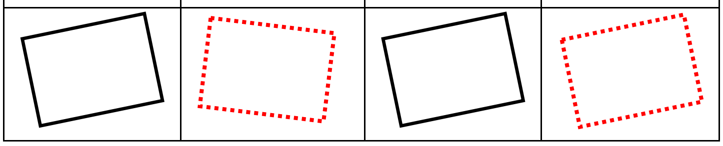
标准模板	当前位置	执行对位后
		

(2)对位方式 2：对位目标变化

对位目标位置由产品 A 提供，而产品 A 每次位置都不一样，所以产品 B 的对位目标位置是变化的。这种方式，需要把产品 A 的目标位置通过相机映射到产品 B 坐标系中。

对位方式示意图如下：

当前位置		执行对位后	
产品 A	产品 B	产品 A	产品 B
			

当前位置		执行对位后	
产品 A	产品 B	产品 A	产品 B
			

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

1.2.2 标定和对位流程

对位算法主要实现计算功能，根据外部提供的 5 个运动特征点完成标定；根据提供的目标位置及样本位置，计算返回偏移量。

标定和对位过程中，运动控制及图像处理是由外部完成的。

标定流程（5 步）：

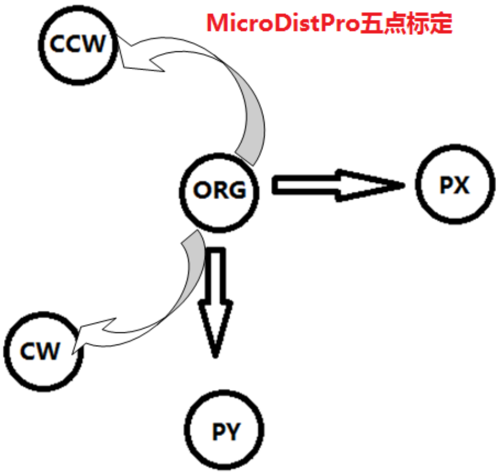
【1】平台在基准位置时候，获取 Mark 点的像素坐标

【2】使平台从基准位置向+X 方向移动，获取移动后的 Mark 点的像素坐标

【3】使平台从基准位置向+Y 方向移动，获取移动后的 Mark 点的像素坐标

【4】使平台从基准位置向+θ方向[顺时针旋转]移动，获取移动后的 Mark 点的像素坐标

【5】使平台从基准位置向-θ方向[逆时针旋转]移动，获取移动后的 Mark 点的像素坐标

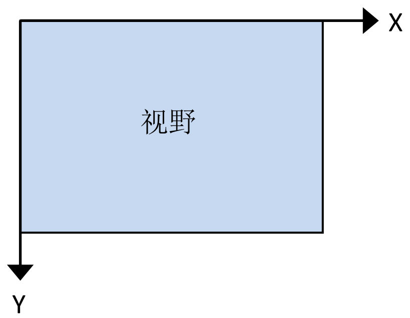


调用 SY_MD_Free_Calibration( )函数完成标定。

调用 SY_MD_Free_Align () 函数返回 X/Y/R 的位置偏移量，执行运动控制完成对位。

1.2.3 对装有棱镜装置的视觉系统

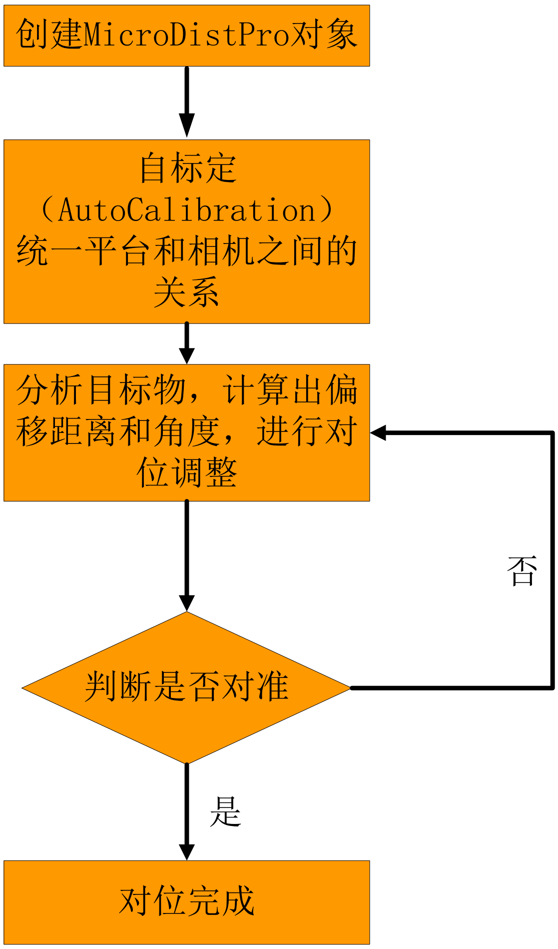
一般地，相机像素坐标系轴向关系如下图所示，即 Y 正方向在正方向右旋转 90 度方位。



MicroDistPro 算法中相机的坐标系采用此标准坐标系，所有的数据运算均在标准坐标系下进行。

由于安装需要而带有棱镜装置的视觉系统，请先对相机做 X 或 Y 方向的镜像。

1.2.4 程序流程图


创建MicroDistPro对象
自标定（AutoCalibration）统一平台和相机之间的关系
分析目标物，计算出偏移距离和角度，进行对位调整
判断是否对准
否
是
对位完成

第 2 章 函数说明

函数一览表		
SY_MD_GetVersion	获取版本信息	Get the MicroDist information
SY_MD_Diagonal_XY	对角补偿量计算	Diagonal Compensation
SY_MD_RTCP	计算虚拟点函数	Real position transfer to pixel
SY_MD_Free_Calibrate	相机标定	Free calibration
SY_MD_Free_Align	2 相机对位函数	Free Align
SY_MD_Free_Align2		Free Align 2 补偿各视野独立坐标系
SY_MD_Camera_Info	获取标定信息	Information of the camera
SY_MD_StudyAtoB	相机映射学习函数	Cameras inline study
SY_MD_TransferAtoB	相机映射函数 A	Cameras inline study
SY_MD_TransferBtoA	相机映射函数 B	Cameras inline study
SY_MD_Nominal_Dist	视野内测距函数	
SY_MD_Nominal_Dist2	2 相机测距函数	
SY_MD_Physic_To_Pulse	物理量转换脉冲	Transfer Physic to pulse
SY_MD_Free_Align3C	3 相机对位函数	Three Camera Align
SY_MD_Free_Align4C	4 相机对位函数	Four Camera Align
SY_MD_Expose_Align2	曝光机对位函数	Expose Machine 3 Camera
SY_MD_Expose_Check2	曝光机精度检查函数	Expose Machine 4 Camera
SY_MD_Expose_Check3	曝光机精度检查函数	
SY_MD_SignalPoint_Align	单点对位函数	单点+角度差对位
SY_MD_SignalPoint_Align2	单点对位函数 2	
SY_MD_LineCalibration5_1	5 相机边标定函数 1	【类型 1】
SY_MD_LineAlign5_1	5 相机边对位函数 1	
SY_MD_LineCalibration5_2	5 相机边标定函数 2	【类型 2】
SY_MD_LineAlign5_2	5 相机边对位函数 2	
SY_MD_LineCalibration6_1	6 相机边标定函数 1	【类型 1】
SY_MD_LineAlign6_1	6 相机边对位函数 1	
SY_MD_LineCalibration6_2	6 相机边标定函数 2	【类型 2】
SY_MD_LineAlign6_2	6 相机边对位函数 2	
SY_MD_Pixel_To_World	像素转换至机械平台坐标	将像素坐标转换到运动平台坐标
SY_MD_Point_MirrorX	像素点镜像函数 X	
SY_MD_Point_MirrorY	像素点镜像函数 Y	
SY_MD_NoMap_Align_2To2	不映射 2 对 2 函数	
SY_MD_NoMap_Align_3To3	不映射 3 对 3 函数	
SY_MD_NoMap_Align_4To2	不映射 4 对 2 函数	
SY_MD_NoMap_Align_2To4	不映射 2 对 4 函数	
SY_MD_NoMap_Align_4To4	不映射 4 对 4 函数	

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

SY_MD_Check_Size	长度检查函数	
SY_MD_Check_Size_AUTO	长度检查函数	

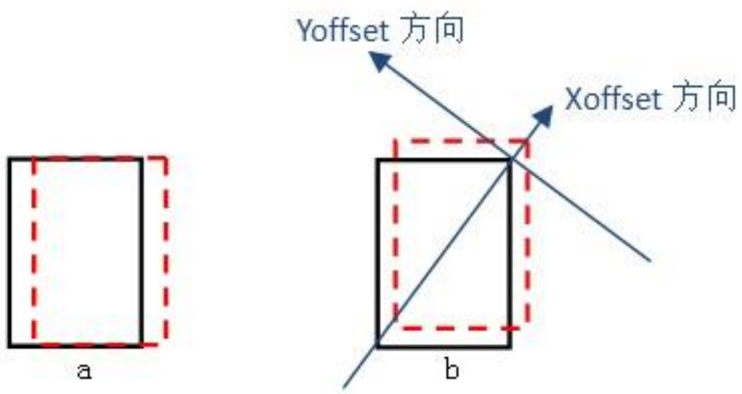
2.1 MicroDistPro 版本信息

```
char*            SY_MD_GetVersion()
```

2.2 对角补偿分量计算函数

```
Point2dF         SY_MD_Diagonal_XY(float a, float b, float Dx, float Dy)
```

由于 MicroDistPro 系统的标定特性，会认为两特征点的连线就是像素坐标系的 X 轴，垂直两特征点的方向为像素坐标系的 Y 轴。所以，对于以对角特征点的对位应用，输入的 Xoffset、Yoffset 补偿量就会作用于两对角特征点连线方向及其垂直方向上。



举例，如左图 a 是需要的产品 X 方向补偿量，由于上述的原因，补偿效果作用在对角特征点连线方向上如右图 b 所示。为了得到左图 a 的补偿效果，就需要 Xoffset 和 Yoffset 的相互作用来实现。

此函数针对对角特征点对位应用，根据产品的特征值（长度与宽度），计算出实际产品补偿量所对应

的 Xoffset 和 Yoffset 的物理标量。

用户请根据设备的具体结构，改变标量的正负号甚至调换 X 和 Y 轴的值，填入对位函数中的 Xoffset 和 Yoffset 参数进行补偿。

float a、b： 产品的长度和宽度 （mm）

float Dx、Dy： 产品 X、Y 方向的补偿量（mm）

注：对于特征点在产品同一边的对位应用，直接填入产品方向补偿量至对位函数中的 Xoffset 和 Yoffset 参数进行补偿，无需调用此函数。

2.3 物理量转换像素量函数

```
Point2dF        SY_MD_RTCP(Point5dVec CPM, Point2dF P, float Xmm, float Ymm)
```

通过此函数计算出当前像素坐标点 P 移动 Xmm 和 Ymm 之后的坐标推算值，即虚拟点计算。

CPM: 相机标定信息。

P： 当前 Mark 点的像素坐标值。

Xmm、Ymm：X、Y 方向的物理距离，单位为“mm”。

2.4 自标定函数

```
Point5dVec     SY_MD_Free_Calibrate(
                    float m_XYdist,
                    float m_Angle,
                    Point10dF MPoint,
                    bool b_CW,
                    bool b_Y
                    )
```

返回值为 Point5dVec 结构体，此结构体记录相机和运动平台之间的关系。

m_XYdist: X 和 Y 轴的移动量，以 Mark 不超出视野为准，单位为“mm”.

m_Angle： 旋转角度量，以 Mark 不超出视野为准,单位为 degree。

MPoint： 记录运动坐标点， 见标定结构变量 Point10dF 定义

b_CW: 发正脉冲时，运动平台顺时针旋转，则为 true，否则为 false。

b_Y： 运动平台 x、y 轴方向关系与相机一致，则为 true，否则为 false。

b_CW、b_Y的设置详细请参见3.2 关于b_CW、b_Y参数

2.5 基于坐标系进行补偿的对位函数

```
AlignData     SY_MD_Free_Align(
                        Point5dVec CPMA, Point5dVec CPMB,
```


地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

```
Point2dF ORGA, Point2dF ORGB,
Point2dF SAMPA, Point2dF SAMPB,
float Xoffset, float Yoffset, float Roffset,
bool b_CW, bool b_Y
)
```

标准的两相机对位函数，返回原点与样本点之间的偏移量，请参见 3.3.5 对位结果结构变量。

CPMA、CPMB：相机 A、B 的标定信息，通过 2.4 自标定函数 标定后得到。

ORGA、ORGB： 相机 A、B 的对位原点。

SAMPA、SAMPB：相机 A、B 的样本点。

Xoffset、Yoffset： X、Y 方向的补偿值 单位 “mm”。

Roffset：角度补偿值，单位为“度”。

b_CW： 发正脉冲时，运动平台顺时针旋转，则为 true，否则为 false。

b_Y： 运动平台中 x、y 轴方向关系与相机一致，则为 true，否则为 false。

b_CW 、b_Y 的设置详细请参见 3.2 关于 b_CW、b_Y 参数。

2.6 基于每个相机进行补偿的对位函数

```
AlignData SY_MD_Free_Align2(
    Point5dVec CPMA, Point5dVec CPMB,
    Point2dF ORGA,Point2dF ORGB,
    Point2dF SAMPA, Point2dF SAMPB,
    Point2dF CompensateA, Point2dF CompenSateB,
    bool b_CW, bool b_Y
)
```

标准的两相机对位函数，返回原点与样本点之间的偏移量，请参见 3.3.5 对位结果结构变量。

CPMA、CPMB：相机 A、B 的标定信息，通过 2.1 自标定函数 标定后得到。

ORGA、ORGB： 相机 A、B 的对位原点。

SAMPA、SAMPB：相机 A、B 的样本点。

CompensateA：相机 A 的补偿量（x，y）。

CompenSateB：相机 B 的补偿量（x，y）。

b_CW： 发正脉冲时，运动平台顺时针旋转，则为 true，否则为 false。

b_Y： 运动平台中 x、y 轴方向关系与相机一致，则为 true，否则为 false。

b_CW 、b_Y 的设置详细请参见 3.2 关于 b_CW、b_Y 参数。

2.7 标定信息函数

```
CamInfo    SY_MD_Camera_Info(int ImgWidth, int ImgHeight, Point5dVec CPM)
```


地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

返回值为CamInfo结构体，可以得到相机的视野大小和像素精度。

ImgWidth： 图像宽度。

ImgHeight：图像高度。

CPM： 相机标定信息，通过2.4 自标定函数标定后得到。

2.8 相机映射学习相关函数

2.8.1 相机映射学习函数

TransferParam SY_MD_StudyAtoB(Point2dF A[2], Point2dF B[2], bool CXdirRev, bool CYdirRev)

学习两相机间的数据映射关系，对于 1.2.1 对位方式 中提到方式 2 的对位形式，需要将对位原点映射至对位平台坐标上。

A[2]：相机A获取的两点坐标

B[2]：相机B获取的两点坐标

CXdirRev、CYdirRev：A、B相机坐标轴的关系，详细请参见 3.1相机映射说明。

2.8.2 相机映射函数 A

Point2dF SY_MD_TransferAtoB(Point2dF A, TransferParam TP, bool CXdirRev, bool CYdirRev)

把相机A中的点坐标映射到相机B坐标系中。

A：相机A获取的点坐标

TP：SY_MD_StudyAtoB相机学习时候返回的结构体，此结构体记录相机和相机之前的关系。

CXdirRev、CYdirRev：A、B相机坐标轴的关系，详细请参见 3.1相机映射说明。

2.8.3 相机映射函数 B

Point2dF SY_MD_TransferBtoA(Point2dF B, TransferParam TP, bool CXdirRev, bool CYdirRev)

把相机 B 中的点坐标映射到相机 A 坐标系中。

B：相机B获取的点坐标

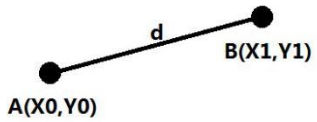
TP：SY_MD_StudyAtoB相机学习时候返回的结构体，此结构体记录相机和相机之前的关系。

CXdirRev、CYdirRev：A、B相机坐标轴的关系，详细请参见 3.1相机映射说明。

2.9 测距函数

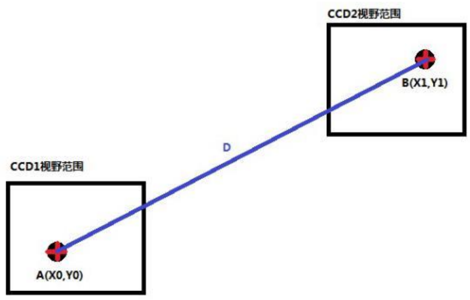
为了应用方便，我们设计出测量两点之间的距离函数。

（1）测量同一个视野范围内两点之间的距离



如上图所示，若是视野范围内有 A 和 B 两点，且知道两个点之间的像素坐标，那么我们可通过函数 SY_MD_Nominal_Dist 得到 A 和 B 两点之间的物理距离。

(2)测量不同视野范围内两点之间的距离



如上图所示，若是不同的相机的视野范围内各自有一个坐标点为 A 和 B，我们要求出两个坐标点 A 和 B 之间的物理距离，那么我们可通过函数 SY_MD_Nominal_Dist2 来实现。

注明：

需要特别注意的地方，我们在使用 SY_MD_Nominal_Dist2 函数来求得的物理距离，由于存在标定的误差，并不能反应真实的物理距离。但在一些特殊的场合，我们仅需要通过此函数求得两个位置之间的相对关系，则可通过此方法来实现。

2.9.1 图像通道测距函数

```
double        SY_MD_Nominal_Dist(Point5dVec CPM, Point2dF PointA, Point2dF PointB)
```

测量同一视野内两点间的物理距离，返回值单位为“mm”。

CPM:　　相机标定信息。

PointA：相机视野内 A 点的像素坐标。

PointB：相机视野内 B 点的像素坐标。

2.9.2 图像通道测距函数 2

```
double        SY_MD_Nominal_Dist2(
                                Point5dVec CPMA, Point5dVec CPMB,
                                Point2dF PointA, Point2dF PointB,
                                bool b_Y
                                )
```

通过两相机测量两 Mark 点之间的物理距离，返回值单位为“mm”。

CPMA、CPMB：相机 A、B 标定信息，2.4 自标定函数 标定后得到。

PointA：相机视野范围内 A 点的像素坐标。

PointB：相机视野范围内 B 点的像素坐标。
b_Y：　运动平台中 x、y 轴方向关系与相机一致，则为 true，否则为 false。
　详细参见 3.2　关于 b_CW、b_Y 参数

2.10　物理量转换为脉冲量

```
PULSE            SY_MD_Physic_To_Pulse(
                     Stage Table,
                     AxisInfo XDir_Axis, AxisInfo YDir_Axis, AxisInfo RDir_Axis,
                     float Xmm, float Ymm, float Degree
                     )
```

　　将物理量转换为对应平台各轴的脉冲量，请参见 3.3.3 脉冲结构变量。

Table:选择平台类型，包括 XYR，XXY，XYY 和 XXYY 等平台类型.

XDir_Axis 和 YDir_Axis：

表示 X 方向和 Y 方向的某个轴信息。

XYR 平台，则填入 X 和 Y 的轴信息。

XXY 平台，则填入 X1 和 Y 的轴信息 （或 X2 和 Y 的轴信息）。

XYY 平台，则填入 X 和 Y1 的轴信息（或 X 和 Y2 的轴信息）。

XXYY 平台，则走入 X1 和 Y1 的轴信息（或者任意 X 方向和 Y 方向的轴信息组合）。

RDir_Axis：旋转轴信息【XXY、XXYY 平台时建立一个空变量作为参数即可】。

Xmm、Ymm: X、Y 方向的物理量，单位为“mm”。

Degree:旋转方向的物理量，单位为“度”。

【*****UVW平台专用函数*****】

```
PULSE            SY_MD_UVW_Table_Control(
                     Stage Table,
                     AxisInfo UAxis, AxisInfo VAxis, AxisInfo WAxis,
                     Point2dF UPos, Point2dF VPos, Point2dF WPos,
                     float Xmm, float Ymm, float Degree, float Degreed
                     )
```

　　UVW 平台专用控制函数，请参见 3.3.3 脉冲结构变量。要求 UVW 每个直线轴回机械零位保证精度。

Table:选择平台类型，包括 XXY 和 XYY 平台类型；

UAxis、VAxis、WAxis：

表示 X 方向和 Y 方向的某个轴信息。

XXY 平台，按 X1、 X2 和 Y 的顺序填入轴信息。

XYY 平台，按 Y1、 Y2 和 X 的顺序填入轴信息。

UPos、VPos、WPos：在机械零位时，3 个直线轴轴承支点在被定义的运动坐标系中坐标。（见下面示例）

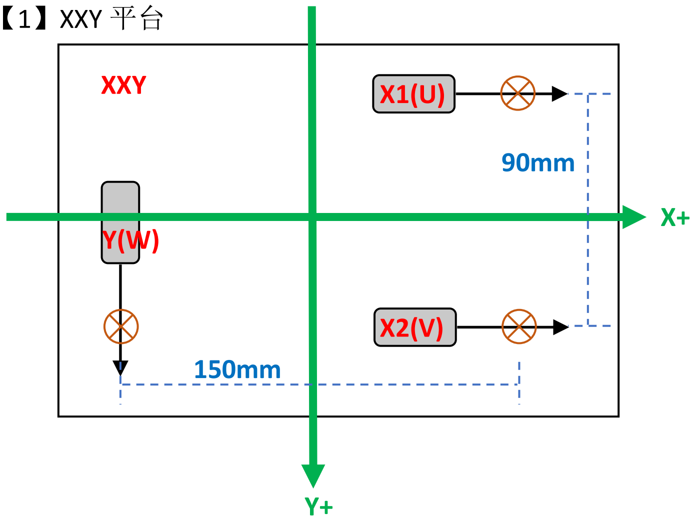
Xmm、Ymm: X、Y 方向的物理量，单位为“mm”。

Degree:旋转方向的物理量，单位为“度”。

Degreed: 当前平台已经旋转的角度，是一个累加量，单位为“度”。平台在机械零位时为 0 度。

UVW 平台定义坐标系及参数定义示例：

【1】XXY 平台

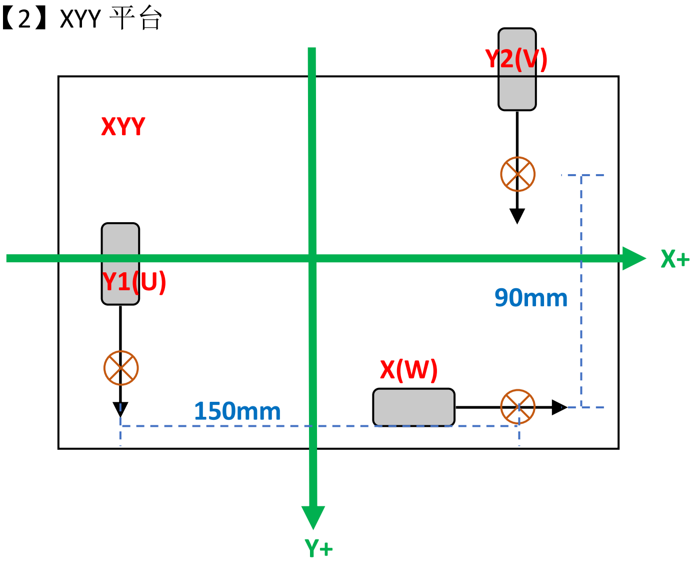


如上图为 XXY 平台类型，X1、X2 和 Y 轴分布如图所示。我们定义向右为运动坐标系的 X+方向，向下为运动坐标系的 Y+方向。

符号 ⊗ 表示轴承支点，由平台结构参数可知，水平支点距离为 150mm，垂直支点距离为 90mm。由此 X1 的坐标为（75，-45），X2 坐标为（75，45），Y 坐标为（-75，45）。

调用函数后，从 PULSE 取 X1mm、X2mm、Y1mm 值控制对应轴做直线运动。

【2】XYY 平台



同理，Y1 坐标为（-75，45），Y2 坐标为（75，-45），X 坐标为（75，45）。调用函数后，从 PULSE 取 X1mm、Y1mm、Y2mm 值控制对应轴做直线运动。

【要点 1】定义运动坐标系。
【要点 2】根据轴承支点在坐标系中位置，填写正确的坐标。
【要点 3】如有单轴运动方向与定义坐标方向不一致的，该轴的运动量取反即可。

2.11 3 相机对位函数

```
AlignData    SY_MD_Free_Align3C(
                          Point5dVec CPMA, Point5dVec CPMB, Point5dVec CPMC,
                          Point2dF ORGA, Point2dF ORGB, Point2dF ORGC,
                          Point2dF SAMPA, Point2dF SAMPB, Point2dF SAMPC,
                          float Xoffset, float Yoffset, float Roffset,
                          bool b_CW, bool b_Y, int AngleFlag
                          )
```

3 相机的对位函数，返回对位原点与样本点之间的偏移量，请参见 3.3.5 对位结果结构变量。

CPMA、CPMB、CPMC：相机 A、B、C 的标定信息，通过 2.4 自标定函数 标定后得到。

ORGA、ORGB、ORGC： 相机 A、B、C、D 的对位原点。

SAMPA、SAMPB、SAMPC：相机 A、B、C、D 的样本点。

Xoffset、Yoffset： X、Y 方向补偿值 单位 “mm”。

Roffset：角度补偿值，单位为“度”。

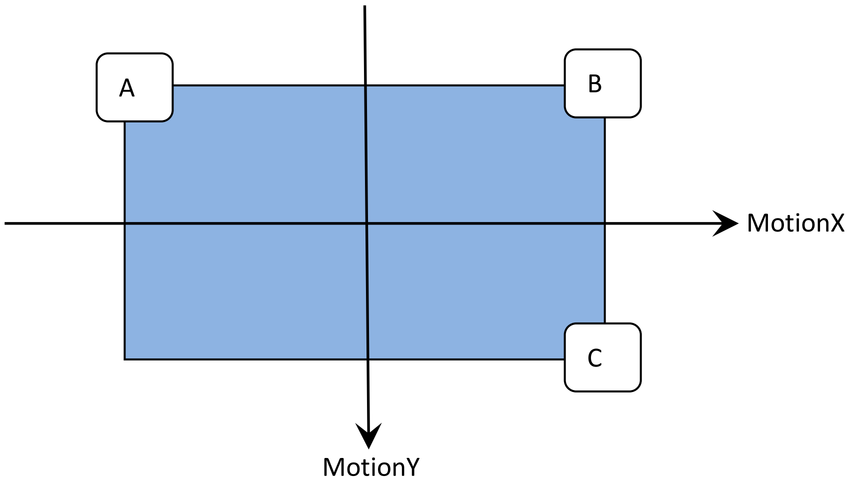
b_CW：发正脉冲时，运动平台顺时针旋转，则为 true，否则为 false。

b_Y：运动平台中 x、y 轴方向关系与相机一致，则为 true，否则为 false。

AngleFlag：角度计算方式 0：[0]AB、BC 同时约束 [1]AB 约束 [2]BC 约束

b_CW 、b_Y 的设置详细请参见 3.2 关于 b_CW、b_Y 参数。

A、B 和 C 三个通道的位置需要形成时针方向，算法默认由 A 指向 B 为 X 方向，B 指向 C 为 Y 方向。偏差量和补偿量的方向均由此方向关系决定。



2.12 4 相机对位函数

```
AlignData    SY_MD_Free_Align4C(
             Point5dVec CPMA, Point5dVec CPMB, Point5dVec CPMC, Point5dVec CPMD,
```

```
Point2dF ORGA, Point2dF ORGB, Point2dF ORGC, Point2dF ORGD,
Point2dF SAMPA, Point2dF SAMPB, Point2dF SAMPC, Point2dF SAMPD,
float Xoffset, float Yoffset, float Roffset,
bool b_CW, bool b_Y,
int AFlag, int XYFlag,
 float AngleX, float AngleY,
  bool TF
)
```

4相机的对位函数，返回对位原点与样本点之间的偏移量，请参见3.3.5 对位结果结构变量。

CPMA、CPMB、CPMC、CPMD:相机 A、B、C、D 的标定信息，通过 2.4 自标定函数 标定后得到。

ORGA、ORGB、ORGC、ORGD： 相机 A、B、C、D 的对位原点

SAMPA、SAMPB、SAMPC、SAMPD：相机 A、B、C、D 的样本点

Xoffset、Yoffset： X 方向补偿值，单位为“mm”。

Roffset：角度补偿值，单位为“度”。

b_CW：发正脉冲时，运动平台顺时针旋转，则为 true，否则为 false。

b_Y：运动平台中 x、y 轴方向关系与相机一致，则为 true，否则为 false。

AFlag: 0--对角线角度 1--水平线角度 2--垂直线角度 3--AB 边角度 4--DC 边角度 5--AD 边角度 6--BC 边角度

XYFlag: 0--4 通道中心 1--A 坐标 2--B 坐标 3--C 坐标 4--D 坐标 5--AB 连线 6--DC 连线 7--AD 连线 8--BC 连线

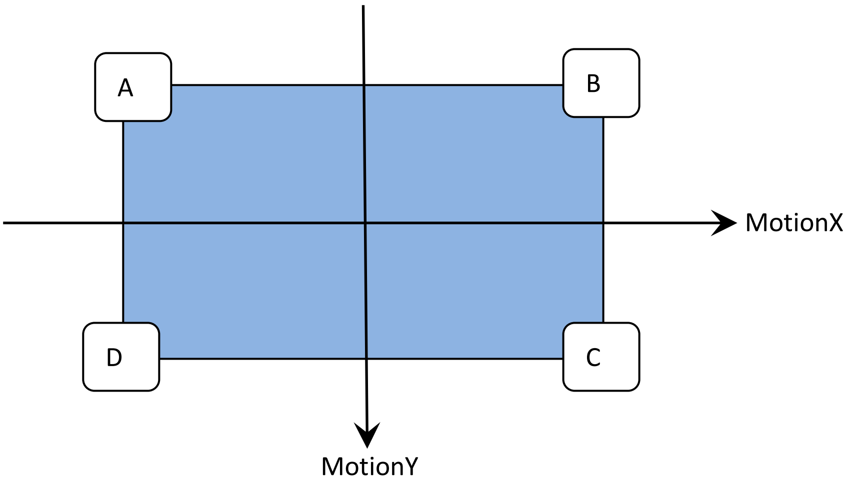
AngleX:保留参数，默认为 0。

AngleY:保留参数，默认为 0。

TF: 保留参数，默认为 false。

b_CW 、b_Y 的设置详细请参见 3.2 关于 b_CW、b_Y 参数。

A、B、C 和 D 四个通道的位置需要形成时针方向，算法默认由（A 指向 B）、(D 指向 C）为 X 方向；(A 指向 D)、(B 指向 C）为 Y 方向。偏差量和补偿量的方向均由此方向关系决定。



地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

2.13　曝光机对位函数

```
AlignData         SY_MD_Expose_Align2(
                        Point5dVec Param[4], Point2dF WPCB[4], Point2dF WFilm[4],
                        bool b_CW, bool b_Y,
                        float Xerror, float Yerror, float Avgerror,
                        float RadiusError, bool b_ChVaild[4], bool Diaflag)
```

曝光机对位应用，兼容2、3和4相机的对位。相机序号由左上角为0开始，顺时针排列。所有相机经由标定板进行统一标定。

Param[4]：4个通道的标定信息

WPCB[4]： PCB靶标的物理坐标

WFilm[4]：Film靶标的物理坐标

b_CW：发正脉冲时，运动平台顺时针旋转，则为 true，否则为 false。

b_Y：运动平台中 x、y 轴方向关系与相机一致，则为 true，否则为 false。

Xerror、Yerror、Avgerror、RadiusError：未使用参数

b_ChVaild[4]：标记某个相机是否使用

Diaflag：未使用参数

2.14　曝光机精度检查函数

```
bool              SY_MD_Expose_Check2(
                        Point5dVec Param[4], Point2dF PCB[4], Point2dF Film[4],
                        float Xerror, float Yerror, float Avgerror,
                        float RadiusError,
                        int mode, bool &b_AvgFlag02, bool &b_AvgFlag13,
                        bool b_ChVaild[4], float f)
```

考虑符号的曝光机对位精度检查，兼容 2、3、和 4 相机的精度检查。相机序号由左上角为 0 开始，顺时针排列。

```
bool              SY_MD_Expose_Check3(
                        Point5dVec Param[4], Point2dF PCB[4], Point2dF Film[4],
                        float Xerror, float Yerror, float Avgerror,
                        float RadiusError,
                        int mode, bool &b_AvgFlag02, bool &b_AvgFlag13,
                        bool b_ChVaild[4], float f)
```

不考虑符号的曝光机对位精度检查，兼容 2、3、和 4 相机的精度检查。相机序号由左上角为 0 开始，顺时针排列。

Param[4]：相机的标定信息

PCB[4]： PCB 靶标的像素坐标

Film[4]：Film 靶标的像素坐标

Xerror：设置的 X 精度

Yerror：设置的 Y 精度

Avgerror：设置的均分精度

RadiusError：设置的 R 精度

mode：检查模式

0、3、6 - 任意 2 点对角检查

1、4、7 - 任意 3 点检查

2、5、8 - 4 点检查

b_AvgFlag02、b_AvgFlag13：02 对角点、13 对角点是否达到均分精度

b_ChVaild[4]：标记某个相机是否在使用

f：未使用参数

2.15 单点对位函数

```
AlignData           SY_MD_SignalPoint_Align(
                        Point5dVec CPM,
                        Point2dF org, Point2dF samp,
                        float OrgAngle,float DeltaAngle,
                        bool b_CW, bool b_Y,
                        float Xoffset, float Yoffset, float Roffset
                        )
```

该函数适用于 Org、Samp 产品都能提供各自 X、Y、Angle 的对位应用。

CPM: 标定参数

Org、Samp：基准点与样本点。

OrgAngle：基准角度。

DeltaAngle：角度偏差，即：(OrgAngle - SampAngle) Org 与 Samp 的角度差。

b_CW：发正脉冲时，运动平台顺时针旋转，则为 true，否则为 false。

b_Y：运动平台中 x、y 轴方向关系与相机一致，则为 true，否则为 false。

Xoffset、Yoffset：XY 方向补偿量，单位为“mm”。

Roffset：角度补偿量，单位为“度”。

2.16 5 相机直线对位

2.16.1 5 相机直线标定函数 1

```
LineCalibration   SY_MD_LineCalibration5_1(
                        Point2dF ImageSize[5],
                        CalibrationLines Line[5] ,
                        float XYDist, float RAngle
                        )
```

根据直线特征进行标定。

ImageSize[5]：相机的图像分辨率

Line[5]：直线标定的 5 条特征直线。

XYDist：XY 标定步长，单位为“mm”。

RAngle：标定角度，单位为“度”。

2.16.2 5 相机直线对位函数 1

```
AlignData             SY_MD_LineAlign5_1(
                        Point2dF ImageSize[5],
                        LineCalibration LC,
                        LineData OrgLine[5],
                        LineData SampLine[5],
                        float ComX, float ComY1, float ComY2
                        )
```

该函数适用于相机 0、1 检测水平直线边的直线对位应用，如下图所示。

ImageSize[5]:相机的图像分辨率

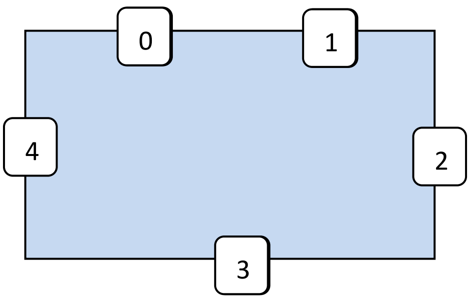
LC：直线标定参数，通过 SY_MD_LineCalibration5_1 函数来得到。

OrgLine[5]、SampLine[5]：基准直线与样本直线。

ComX：相机 2、4 方向补偿量，单位为“mm”。

ComY1：相机 0 方向补偿量，单位为“mm”。

ComY2：相机 1 方向补偿量，单位为“mm”。



地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

2.16.3 5 相机直线标定函数 2

```
LineCalibration   SY_MD_LineCalibration5_2(
                        Point2dF ImageSize[5],
                        CalibrationLines Line[5] ,
                        float XYDist, float RAngle
                        )
```

根据直线特征进行标定。

ImageSize[5]：相机的图像分辨率

Line[5]：直线标定的 5 条特征直线。

XYDist：XY 标定步长，单位为“mm”。

RAngle：标定角度，单位为“度”。

2.16.4 5 相机直线对位函数 2

```
AlignData         SY_MD_LineAlign5_2(
                        Point2dF ImageSize[5],
                        LineCalibration LC,
                        LineData OrgLine[5],
                        LineData SampLine[5],
                        float ComX1, float ComX2, float ComY
                        )
```

该函数适用于相机 0、1 检验垂直边的直线对位应用，如下图所示。

ImageSize[5]:相机的图像分辨率

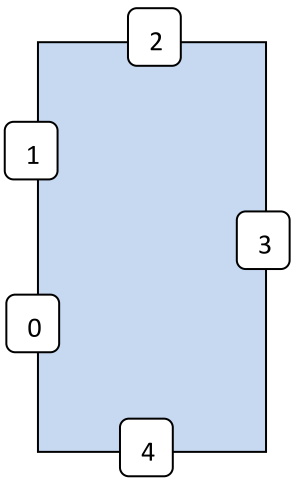
LC：直线标定参数，通过 SY_MD_LineCalibration5_2 函数来得到。

OrgLine[5]、SampLine[5]：基准直线与样本直线。

ComX1：相机 2、4 方向补偿量，单位为“mm”。

ComX2：相机 0 方向补偿量，单位为“mm”。

ComY：相机 1 方向补偿量，单位为“mm”。



2.17 6 相机直线对位

2.17.1 6 相机直线标定函数 1

```
LineCalibration   SY_MD_LineCalibration6_1(
                         Point2dF ImageSize[6],
                         CalibrationLines Line[6] ,
                         float XYDist, float RAngle
                         )
```

根据直线特征进行标定。

ImageSize[6]：相机的图像分辨率

Line[6]：直线标定的 5 条特征直线。

XYDist：XY 标定步长，单位为“mm”。

RAngle：标定角度，单位为“度”。

2.17.2 6 相机直线对位函数 1

```
AlignData         SY_MD_LineAlign6_1(
                         Point2dF ImageSize[6],
                         LineCalibration LC,
                         LineData OrgLine[6],
                         LineData SampLine[6],
                         float ComX, float ComY1, float ComY2
                           )
```

ImageSize[6]:相机的图像分辨率

LC：直线标定参数，通过 SY_MD_LineCalibration6 函数来得到。

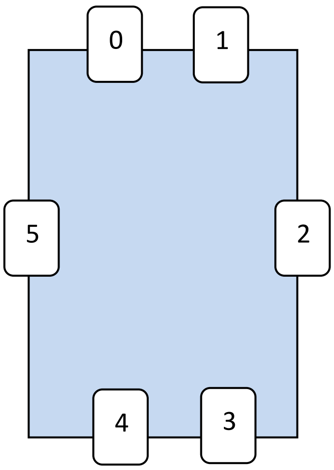
OrgLine[6]、SampLine[6]：基准直线与样本直线。

ComX：相机 2、5 方向补偿量，单位为“mm”。

ComY1：相机 0、4 方向补偿量，单位为“mm”。

ComY2：相机 1、3 方向补偿量，单位为“mm”。

该函数适用于以下对位应用，如下图所示。



2.17.3 6 相机直线标定函数 2

```
LineCalibration    SY_MD_LineCalibration6_2(
                        Point2dF ImageSize[6],
                        CalibrationLines Line[6] ,
                        float XYDist, float RAngle
                        )
```

根据直线特征进行标定。

ImageSize[6]：相机的图像分辨率

Line[6]：直线标定的 5 条特征直线。

XYDist：XY 标定步长，单位为“mm”。

RAngle：标定角度，单位为“度”。

2.17.4 6 相机直线对位函数 2

```
AlignData         SY_MD_LineAlign6_2(
                        Point2dF ImageSize[6],
                        LineCalibration LC,
                        LineData OrgLine[6],
                        LineData SampLine[6],
                        float ComX, float ComY1, float ComY2,
                        unsigned int flag
                        )
```

ImageSize[6]:相机的图像分辨率

LC：直线标定参数，通过 SY_MD_LineCalibration6_2 函数来得到。

OrgLine[6]、SampLine[6]：基准直线与样本直线。

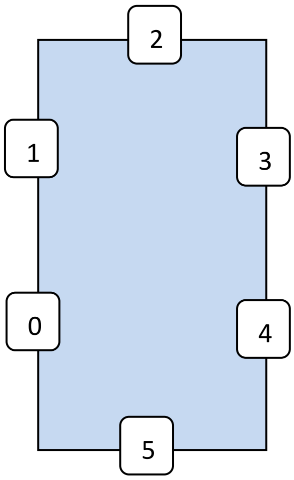
ComX：相机 2、4 方向补偿量，单位为“mm”。

ComY1：相机 0 方向补偿量，单位为“mm”。

ComY2：相机 1 方向补偿量，单位为“mm”。

flag:0：01　1:34　2：01&34

该函数适用于以下对位应用，如右图所示。



2.18 像素、平台坐标转换函数

```
Point2dD        SY_MD_Pixel_To_World(Point2dF pixel, Point5dVec CPM, bool b_Y)
```

该函数把像素坐标转换至标定的机构坐标系中。

pixel: 相机中像素坐标

CPM: 标定参数

b_Y: 运动平台中 x、y 轴方向关系与相机一致，则为 true，否则为 false

2.19 像素点镜像函数

```
Point2dF        SY_MD_Point_MirrorX(int ImgWidht, int ImgHeight, Point2dF p);
Point2dF        SY_MD_Point_MirrorY(int ImgWidht, int ImgHeight, Point2dF p);
```

该函数把像素坐标转换至 X 或者 Y 方向镜像后的坐标。

ImgWidht: 图片宽度

ImgHeight: 图片高度

p: 像素坐标

2.20 不映射对位函数

该系列函数支持不做相机映射学习或不需要做标定流程的对位应用，一般需要进行多次对位达到精度。共有参数如下表所示，私有参数见各函数注释。

CamStatus	对应的两相机坐标轴方向关系，两相机必须保证同轴 0：XY同方向　1：X同向Y异向　2：X异向Y同向　3：XY异方向
OrgStd、SampStd	Org产品和Samp产品的基准点
Org、Samp	Org产品和Samp产品的样本点
mpp	像素分辨率，mm/pixel
deltaX、deltaY	XY方向补偿量，单位为“mm”
deltaA	角度补偿量，单位为“度”。
param	相机标定参数
AngleFlag	[0]AB、BC同时约束　[1]AB约束　[2]BC约束 [3]视野内补偿
XYFlag	[0]AB中点　[1]A点　[2]B点

2.20.1 2 对 2 应用

```
AlignData       SY_MD_NoMap_Align_2To2(
                        int CamStatus,
                        Point5dVec paramA, Point5dVec paramB,
```

```
Point2dF OrgStdA, Point2dF OrgStdB,
Point2dF OrgA, Point2dF OrgB,
float LOrgStdAB,
float mmpA, float mmpB,
Point2dF SampStdA, Point2dF SampStdB,
Point2dF SampA, Point2dF SampB,
float deltaX, float deltaY, float deltaA,
unsigned int XYFlag)
```

不映射的2目标与2对象对位函数。

paramA、paramB：2相机通道的标定信息

OrgStdA、OrgStdB：2Org通道的基准坐标

OrgA、OrgB：2Org通道的产品坐标

LOrgStdAB：对应Org基准坐标的AB尺寸

mmpA、mmpB：2个Org通道的像素分辨率

SampStdA、SampStdB：2Samp通道的基准坐标

SampA、SampB：2Samp通道的产品坐标

deltaX、deltaY、deltaA：补偿量

XYFlag：XY对齐方式

0 – 以AB中点对齐

1 – 以A点对齐

2 – 以B点对齐

2.20.2 3 对 3 应用

```
AlignData    SY_MD_NoMap_Align_3To3(
        int CamStatus,
        Point5dVec paramA, Point5dVec paramB,Point5dVec paramC,
        Point2dF OrgStdA, Point2dF OrgStdB, Point2dF OrgStdC,
        float LOrgStdAB, float LOrgStdBC,
        Point2dF SampStdA, Point2dF SampStdB, Point2dF SampStdC,
        Point2dF OrgA, Point2dF OrgB, Point2dF OrgC,
        Point2dF SampA, Point2dF SampB, Point2dF SampC,
        float mpp_OrgA, float mpp_OrgB, float mpp_OrgC,
        float deltaX, float deltaY, float deltaA,
        int AngleFlag)
```

不映射的3目标与3对象对位函数。

paramA、paramB、paramC：4相机通道的标定信息

OrgStdA、OrgStdB、OrgStdC：4Org通道的基准坐标

LOrgStdAB、LOrgStdBC：对应Org基准坐标的AB、BC尺寸

SampStdA、SampStdB、SampStdC：4Samp通道的基准坐标

OrgA、OrgB、OrgC：3Org通道的产品坐标

SampA、SampB、SampC：3Samp通道的产品坐标

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

mmpA、mmpB、mmpC：3个Org通道的像素分辨率
deltaX、deltaY、deltaA：补偿量
AngleFlag：未使用参数

2.20.3 4 对 2 应用

```
AlignData        SY_MD_NoMap_Align_4To2(
                    int CamStatus,
                    Point5dVec paramA, Point5dVec paramB, Point5dVec paramC, Point5dVec paramD,
                    Point2dF OrgStdA, Point2dF OrgStdB,
                    Point2dF OrgA, Point2dF OrgB,
                    float LOrgStdAB, float mmpA, float mmpB,
                    Point2dF SampStdA, Point2dF SampStdB, Point2dF SampStdC, Point2dF SampStdD,
                    Point2dF SampA, Point2dF SampB, Point2dF SampC, Point2dF SampD,
                    float deltaX, float deltaY, float deltaA)
```

不映射的2目标与4对象对位函数。
paramA、paramB、paramC、paramD：4相机通道的标定信息
OrgStdA、OrgStdB：2Org通道的基准坐标
OrgA、OrgB：2Org通道的产品坐标
LOrgStdAB：对应Org基准坐标的AB尺寸
mmpA、mmpB：2个Org通道的像素分辨率
SampStdA、SampStdB、SampStdC、SampStdD：4Samp通道的基准坐标
SampA、SampB、SampC、SampD：4Samp通道的产品坐标
deltaX、deltaY、deltaA：补偿量

2.20.4 2 对 4 应用

```
AlignData        SY_MD_NoMap_Align_2To4(
                      int CamStatus,
                      Point5dVec paramA, Point5dVec paramB,
                      Point2dF OrgStdA, Point2dF OrgStdB, Point2dF OrgStdC, Point2dF OrgStdD,
                      Point2dF OrgA, Point2dF OrgB, Point2dF OrgC, Point2dF OrgD,
                      float LOrgStdAB, float LOrgStdBC,
                      float mmpA, float mmpB, float mmpC, float mmpD,
                      Point2dF SampStdA, Point2dF SampStdB,
                      Point2dF SampA, Point2dF SampB,
                      float deltaX, float deltaY, float deltaA)
```

不映射的4目标与2对象对位函数。

paramA、paramB：2相机通道的标定信息

OrgStdA、OrgStdB、OrgStdC、OrgStdD：4Org通道的基准坐标

OrgA、OrgB、OrgC、OrgD：4Org通道的产品坐标

LOrgStdAB、LOrgStdBC：对应Org基准坐标的AB、BC尺寸

mmpA、mmpB、mmpC、mmpD：4个Org通道的像素分辨率

SampStdA、SampStdB：2Samp通道的基准坐标

SampA、SampB：2Samp通道的产品坐标

deltaX、deltaY、deltaA：补偿量

2.20.5 4 对 4 应用

```
AlignData      SY_MD_NoMap_Align_4To4(
                   int CamStatus,
                   Point5dVec paramA, Point5dVec paramB, Point5dVec paramC, Point5dVec paramD,
                   Point2dF OrgStdA, Point2dF OrgStdB, Point2dF OrgStdC, Point2dF OrgStdD,
                   Point2dF OrgA, Point2dF OrgB, Point2dF OrgC, Point2dF OrgD,
                   float LOrgStdAB, float LOrgStdBC,
                   float mmpA, float mmpB, float mmpC, float mmpD,
                   Point2dF SampStdA, Point2dF SampStdB, Point2dF SampStdC, Point2dF SampStdD,
                   Point2dF SampA, Point2dF SampB, Point2dF SampC, Point2dF SampD,
                   float LSampStdAB, float LSampStdBC,
                   float deltaX, float deltaY, float deltaA)
```

不映射的4目标与4对象对位函数。

paramA、paramB、paramC、paramD：4相机通道的标定信息

OrgStdA、OrgStdB、OrgStdC、OrgStdD：4Org通道的基准坐标

OrgA、OrgB、OrgC、OrgD：4Org通道的产品坐标

LOrgStdAB、LOrgStdBC：对应Org基准坐标的AB、BC尺寸

mmpA、mmpB、mmpC、mmpD：4个Org通道的像素分辨率

SampStdA、SampStdB、SampStdC、SampStdD：4Samp通道的基准坐标

SampA、SampB、SampC、SampD：4Samp通道的产品坐标

LSampStdAB、LSampStdBC：未使用参数

deltaX、deltaY、deltaA：补偿量

2.21 距离检查函数

```
float          SY_MD_Check_Size(
                   Point2dF StdPA, Point2dF StdPB, float LStdPAB,
                   Point2dF SampA, Point2dF SampB,
                   float mmpA, float mmpB, unsigned int F)
```


地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

根据外部设定的距离检查函数。

StdPA、StdPB：A、B 通道的基准点

LStdPAB：对应基准点的物理长度

SampA、SampB：要测量的点像素坐标

mmpA、mmpB：A、B 通道的像素分辨率

F：0 – 两点水平； 1 – 两点垂直

```
float       SY_MD_Check_Size_AUTO(
            Point5dVec paramA, Point5dVec paramB,
            Point2dF StdPA, Point2dF StdPB, float LStdPAB,
            Point2dF SampA, Point2dF SampB,
            float mmpA, float mmpB)
```

自动识别 A、B 点姿态的长度检查函数。

paramA、paramB：A、B 通道标定信息

StdPA、StdPB：A、B 通道的基准点

LStdPAB：对应基准点的物理长度

SampA、SampB：要测量的点像素坐标

mmpA、mmpB：A、B 通道的像素分辨率

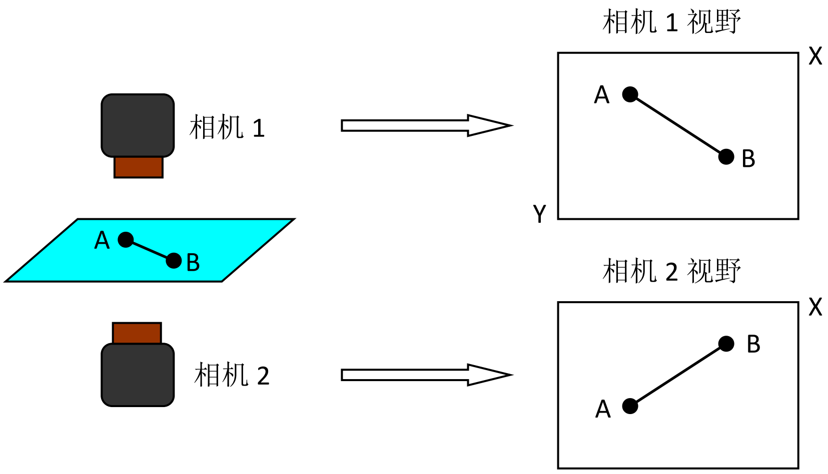
地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

第 3 章 重要的使用说明

以上我们介绍了在 VC++中如何配置 MicroDistPro 和函数说明，为了更好地理解 MicroDistPro 和正确地使用函数，在这一节，将论述 MicroDistPro 的重要概念与数据结构体。

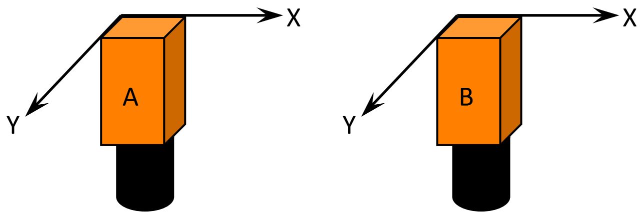
3.1 相机映射说明

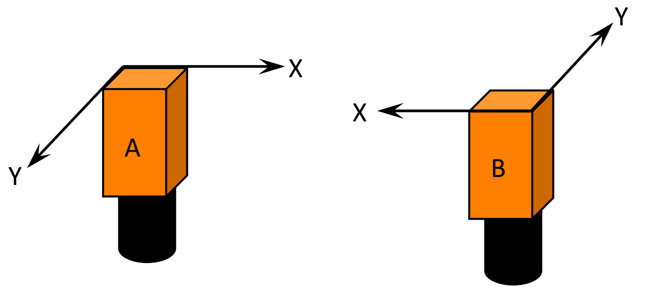
相关函数：SY_MD_StudyAtoB( )，SY_MD_TransferAtoB( )，SY_MD_TransferBtoA( )。
相机映射，是实现相机间坐标的互相转换，以满足目标位置变化的对位模式应用。

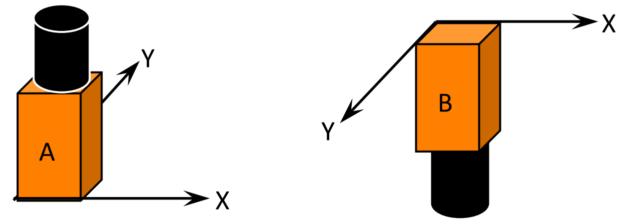


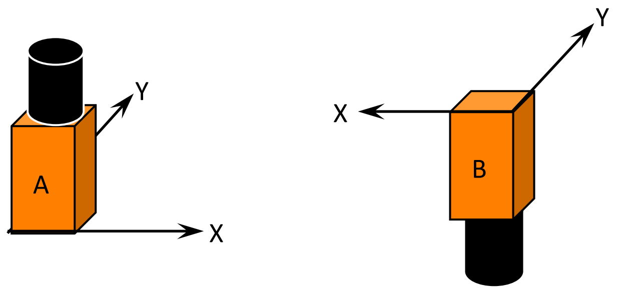
假设相机 1 和相机 2 观察同一片产品上 A、B 两点（一般地，产品需要从相机 1 移动至相机 2 处），相机 1 视野中 A、B 两点坐标被保存在 Data1[2]数组，相机 2 视野中 A、B 两点坐标被保存在 Data2[2]数组，那么我们就可以调用 SY_MD_StudyAtoB( )实现相机映射学习。

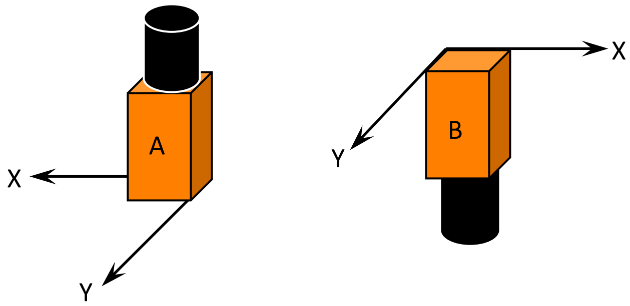
映射函数有两布尔量参数 bool CXdirRev、 bool CYdirRev,它们与映射相机的安装方式有关。为了正确使用函数，请将相机安装成以下方式中某一种。【其它情况请垂询】

方式 1	
	CXdirRev = false CYdirRev = false

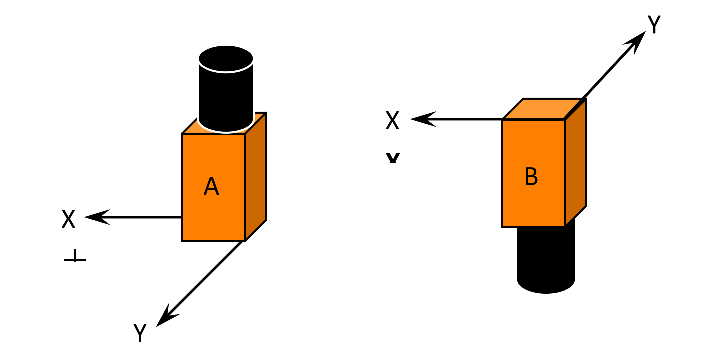
方式 2（A、B 相机位置可以左右互换）	
	CXdirRev = true CYdirRev = true

方式 3（A、B 相机位置可以左右互换）	
	CXdirRev = false CYdirRev = true

方式 4（A、B 相机位置可以左右互换）	
	CXdirRev = true CYdirRev = false

方式 5（A、B 相机位置可以左右互换）	
	CXdirRev = true CYdirRev = false

地址：深圳市宝安区沙井街道茅洲山工业园全至科创大厦2A-1
电话：0755-23712116
传真：0755-23020631

<table>
<tr><td colspan="2">方式 6（A、B 相机位置可以左右互换）</td></tr>
<tr><td>
</td><td>CXdirRev = false
CYdirRev = true</td></tr>
</table>

从以上各种方式可知，CXdirRev 和 CYdirRev 主要描述了两个相机之间像素坐标轴的方向异同性。

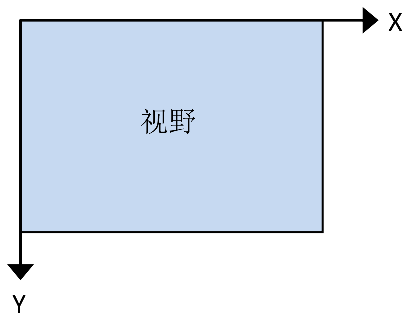
Example：

```
//相机1和相机 2参数学习转换
Point2dF    A[2], B[2];
TransferParam    TP12;
//图像分析，得到 A[2]、B[2]
//调用映射学习函数
TP12 = SY_MD_StudyAtoB(A,B,False,True);

//相机映射数据
Point2dF TA, TB;
//图像分析得到 TA
TB = SY_MD_TransferAtoB(TA,TP12,False,True);
```

3.2 关于 b_CW、b_Y 参数

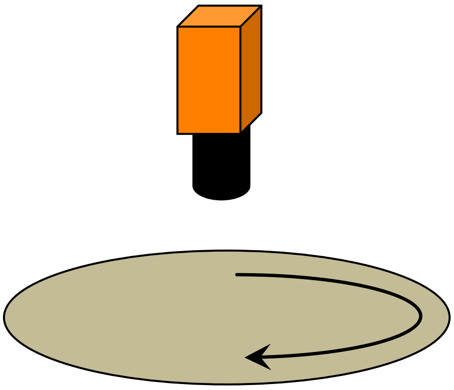
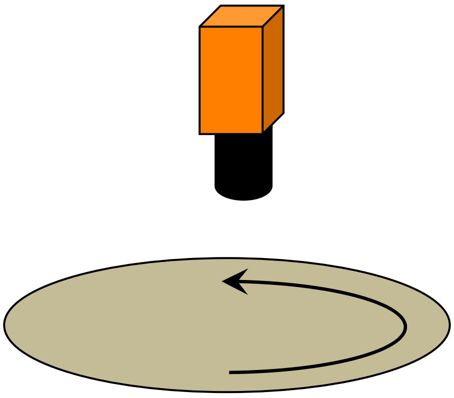
一般地，相机像素坐标系符合下图的关系：



即 Y 轴的正方向为 X 轴正方向的右旋转 90 度方位，我们的算法是基于这种标准的视觉坐标系的。

【*】某些应用的要求，相机需要透过棱镜取像，则这种标准坐标系被改变；此时，请做相机的 X 镜像或 Y 镜像，使改变成标准像素坐标系，方可正确地使用函数。

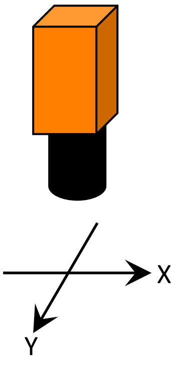
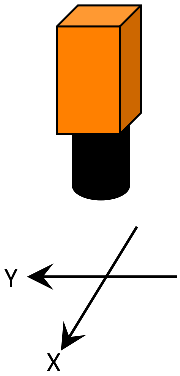
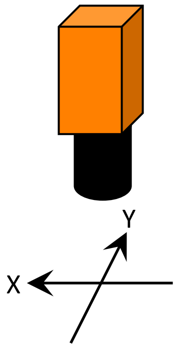
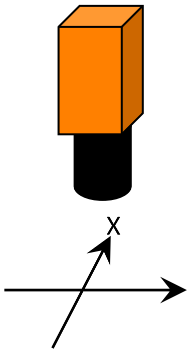
b_CW 描述了相机坐标系与机械旋转方向的关系，请参考图示进行设置。

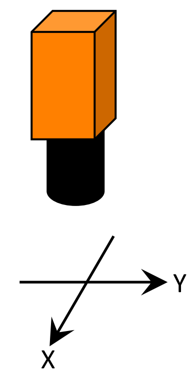
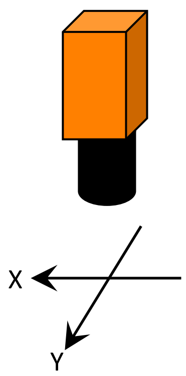
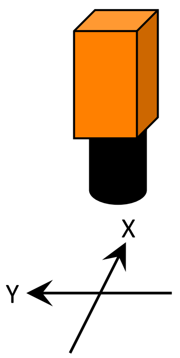
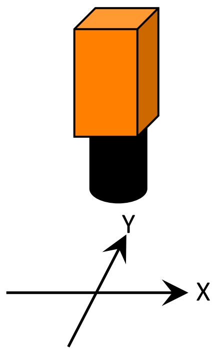
以相机的方向观察旋转运动（相机的自旋转角度不需要考虑）	
	
当发送正脉冲时，对位平台旋转方向为顺时针，则 b_CW = true	当发送正脉冲时，对位平台旋转方向为逆时针，则 b_CW = false
对于相机由下往上观察旋转运动的情况，亦遵循这个原则。	

b_Y 描述了相机坐标系与机械 X、Y 运动轴的关系，请参考图示进行设置。

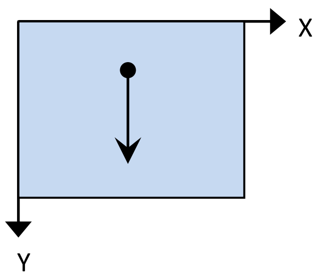
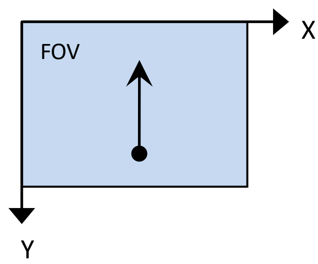
这里分两种情况，一种是相机能同时观察到 X、Y 的运动情况；另外一种情况是相机只能观察到 X 或 Y 的运动情况。

【1】以相机的方向观察平台 X、Y 运动（**相机的自旋转角度不需要考虑**），相机能同时观察 X、Y 方向的运动情况（标准模型）。

			
符合上面 4 种运动情况的，b_Y = true			

			
符合上面 4 种运动情况的，b_Y = false			
对于相机由下往上观察 X、Y 运动的情况，亦遵循这个原则。 这种情况下，对位函数返回的偏移量在数值与符号上都是正确的。			

【2】对于分离式的运动平台，相机只能观察到 X 或 Y 的运动情况(**不需要考虑相机的自旋转角度**)，则按下列原则设置：

相机只能观察到 X 方向运动	b_Y = true	
相机只能观察到 Y 方向运动		
	当 Y 轴正方向运动时，标记点在相机视野内从上往下运动 b_Y = true	当 Y 轴正方向运动时，标记点在相机视野内从下往上运动 b_Y = false
对于相机由下往上观察 X、Y 运动的情况，亦遵循这个原则。 这种情况下，对位函数返回的偏移量在数值上是正确的，但是符号需要进行测试并调整。		

3.3 实用的结构体变量参数

3.3.1 运动轴信息结构变量

MicroDistPro 提供此结构来管理对位的运动轴，是对位系统正确进行运动控制的关键。形式如下：

```
typedef struct AxisInfo
{
    float DriveRatio;
    float DriveRadius;
    float Pitch;
    float PPR;
}AxisInfo;
```

Pitch：　运动轴的螺纹距，单位为“mm”

PPR：　运动轴电机的分辨率,单位为 pulse

DriveRatio：　带比例的旋转轴比率,直驱时此值为 1

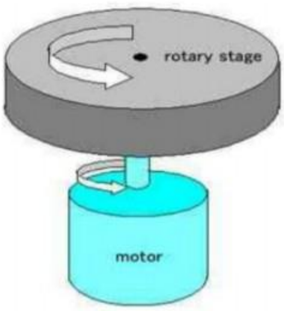
DriveRadius：带半径的旋转轴半径值，单位为“mm”

注：DriveRatio 与 DriveRadius，其中一种形式有效时，另外形式的值为 0；

假设现有一带比例的旋转装置，驱动电机分辨率 6400，比例系数 90.0，结构体的赋值举例如下：

```
#define Raxis 0;
AxisInfo MotionAxisInfo[3];
MotionAxisInfo[RAxis].Pitch = 1.0;
MotionAxisInfo[RAxis].PPR = 6400;
MotionAxisInfo[RAxis].DriveRadius = 0;
MotionAxisInfo[RAxis].DriveRatio = 90.0;
```

带比例的旋转运动　电机通过如蜗杆、齿轮等装置，带动做旋转运动。

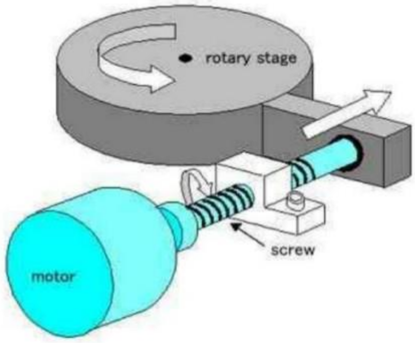


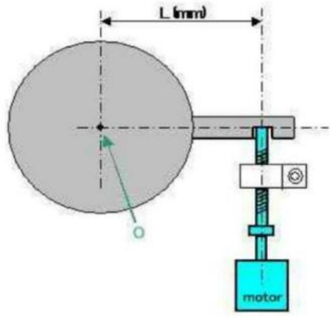
假设有一比例装置，其传动比为 100，电机的分辨率为 3600；那么此旋转装置的角度分辨率：

360÷3600÷100 = 0.001（度/脉冲）

此时 DriveRatio = 100 而 DriveRadius = 0;

带半径的旋转运动　电机通过丝杆装置带动做旋转运动





此时 DriveRatio = 0 而 DriveRadius = L；

3.3.2 运动平台类型

```
enum Stage {XYR, XXY, XYY, XXYY, XXYY_LLR };
```

在对位的应用中，大部分的运动平台为 XYR（R 代表旋转平台）或者采用 XXY 和 XXYY 平台，为了方便大家在使用相关函数的时候，我们在对位函数 SY_MD_Free_Align 和物理量转换为脉冲量的函数 SY_MD_Physic_To_Pulse 中引入了平台的类型，通过选定使用的平台类型后，函数会计算对应各轴的脉冲量或物理量(详见结构体 PULSE 和 AlignData 的说明)。

3.3.3 脉冲结构变量

```
typedef struct PULSE
{
  //XYR
  float X;
  float Y;
  float R;

  //XXY & XYY & XXYY
  float X1;
  float X2;
  float Y1;
  float Y2;

  float X1mm;
  float X2mm;
  float Y1mm;
  float Y2mm;
}PULSE;
```

【1】XYR 平台

X:X 轴的位移量，单位为脉冲；Y：Y 轴的位移量，单位为脉冲；R：旋转轴的位移量，单位为脉冲。

【2】XXY 或 XYY 平台

X1：X1 轴的位移量，单位为脉冲；X2 为 X2 轴的位移量，单位为脉冲；Y1：Y 轴的位移量，单位为脉冲。

X1：X1 轴的位移量，单位为脉冲；Y1 为 Y1 轴的位移量，单位为脉冲；Y2：Y2 轴的位移量，单位为脉冲。

【3】XXYY 平台

X1：X1 轴的位移量，单位为脉冲；X2 为 X2 轴的位移量，单位为脉冲；Y1：Y1 轴的位移量，单位为脉冲;Y2:Y2 轴的位移量，单位为脉冲。

由于对位应用中所用到的对位平台有 XYR 和 XXY 以及 XXYY 甚至一些较为复杂的对位平台，为了方便使用，我们设计 SY_MD_Physic_To_Pulse 函数用于将物理量转换为相应的脉冲量，方便用户直接给运动控制卡或者其他运动控制器直接下达脉冲位置信息即可完成相应的移动量。

3.3.4 五点标定坐标结构变量

MicroDistPro 提供此结构来配合外部抓取 Mark 点，完成自标定，确定平台和相机之间的关系。形式如下：

```
typedef struct Point10dF
{
    float ORGX;
    float ORGY;
    float PX_X;
    float PX_Y;
    float PY_X;
    float PY_Y;
    float CWX;
    float CWY;
    float CCWX;
    float CCWY;
}Point10dF;
```

参数说明：

ORGX 和 ORGY：相机抓取原点坐标的像素坐标。

PX_X 和 PX_Y：控制运动轴从原点向 X 方向移动后，抓取的像素坐标。

PY_X 和 PY_Y：控制运动轴从原点向 Y 方向移动后，抓取的像素坐标。

CWX 和 CWY：设备正方向旋转后，抓取的像素坐标。

CCWX 和 CCWY：设备负方向旋转后，抓取的像素坐标。

此结构体一般配合 SY_MD_Free_Calibrate(float m_XYdist, float m_Angle, Point10dF MPoint)来使用。

3.3.5 对位结果结构变量

MicroDistPro 提供此结构来获得运动偏移量的信息，即 X、Y 和旋转轴的偏移量，单位以脉冲或物理量单位给出。形式如下：

```
typedef struct AlignData
{
    float xmm;          //mm
    float ymm;        //mm
    float angle;      //deg
    double org_width;
    double org_heigth;
    double samp_width;
    double samp_height;
}AlignData;
```

参数说明：

xmm：X 方向偏移量，单位为“mm”。

ymm：Y 方向偏移量，单位为“mm”。

angle：角度偏移量，单位度。

org_width:保留参数，不需要使用。

org_heigth:保留参数，不需要使用。

samp_width:保留参数，不需要使用。

samp_height:保留参数，不需要使用。

3.3.6 标定信息结构变量

MicroDistPro 通过执行标定函数后，统一相机和平台的坐标系，使用 CaliInfos 此结构来获得当前相机的分辨率。形式如下：

```
typedef struct CamInfo
{
    int      c_width;     //Image width
    int      c_height;    //Image height
    float    f_width;     //FOV width   (mm)
    float    f_height;    //FOV height   (mm)
    float    xp;          //um/pixel in Camera X axis  (mm/pixel)
    float    yp;          //un/pixel in Camera Y axis  (mm/pixel)
}CamInfo;
```

参数说明：

c_width 和 c_height：图像的长度和宽度【单位为像素】。

f_width 和 f_height：视野大小【单位为“mm”】。

xp 和 yp：相机水平方向和垂直方向的像素精度【单位 um/Pixel】。

3.3.7 标定直线结构变量与直线标定结果结构变量

MicroDistPro 执行直线标定函数时，使用 CalibrationLines 此结构来记录 5 条用于标定的直线，并最终用 LineCalibration 结构体记录标定结果。形式如下：

```
struct CalibrationLines
{
    LineData OrgLine;//ORG
    LineData XPLine;//X+
    LineData YPLine;//Y+
    LineData RPLine;//R+
    LineData RMLine;//R-
};

struct LineCalibration
{
    double V[18];
};
```

参数说明：

OrgLine：标定起点处的直线信息。

XPLine：设备沿着 X 正方向移动后的直线信息。

YPLine：设备沿着 Y 正方向移动后的直线信息。

RPLine：设备正方向旋转后的直线信息。

RMLine：设备反方向旋转后的直线信息。

3.3.8 自标定结果结构变量

MicroDistPro 执行自标定函数时，使用 Point5dVec 此结构来记录标定的结果。形式如下：

```
typedef struct Point5dVec
{
    double V1;
    double V2;
    double V3;
    double V4;
    double V5;
}Point5dVec;
```

参数说明：

V1：标定的旋转中心 X 坐标。

V2：标定的旋转中心 Y 坐标。

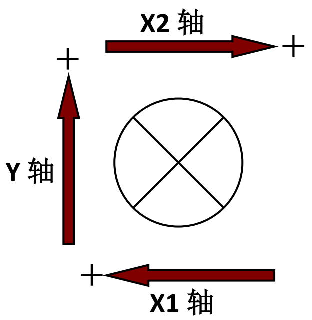
V3：相机 X 轴与平台 X 轴之间的夹角（弧度）。

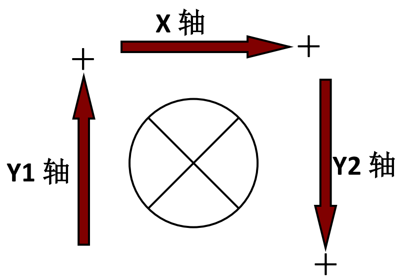
V4：X 方向的像素分辨率。

V5：Y 方向的像素分辨率。

3.4 关于 XXY、XYY 与 XXYY 平台

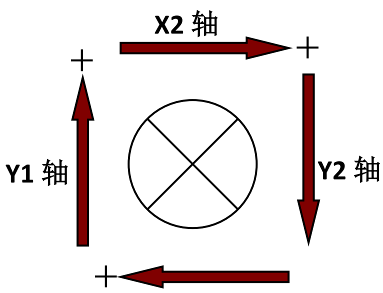
UVW 平台的三轴结构与方向搭配多样，为了方便客户使用 MicroDistPro 的函数及管理 UVW 平台，MicroDistPro 将 UVW 平台定义为 XXY 型平台，且规定三轴的正方向为使得 XXY 平台顺时针方向旋转的方向，如下图所示：





图中，列举了 XXY、XYY 平台的结构，三轴的同方向运动使平台能作顺时针或逆时针旋转运动。

同样地，对于四轴的 XXYY 平台，我们规定四轴的正方向为使得平台顺时针方向旋转的方向国，如下图所示：



四轴的同方向运动使平台能作顺时针或逆时针旋转运动。

双翌
SHUANGYI

0755-23712116

网址：www.shuangyi-tech.com

邮箱：contact@shuangyi-tech.com

地址：深圳市宝安区沙井街道后亭茅洲山工业园全至科创大厦2A-1

微信公众号