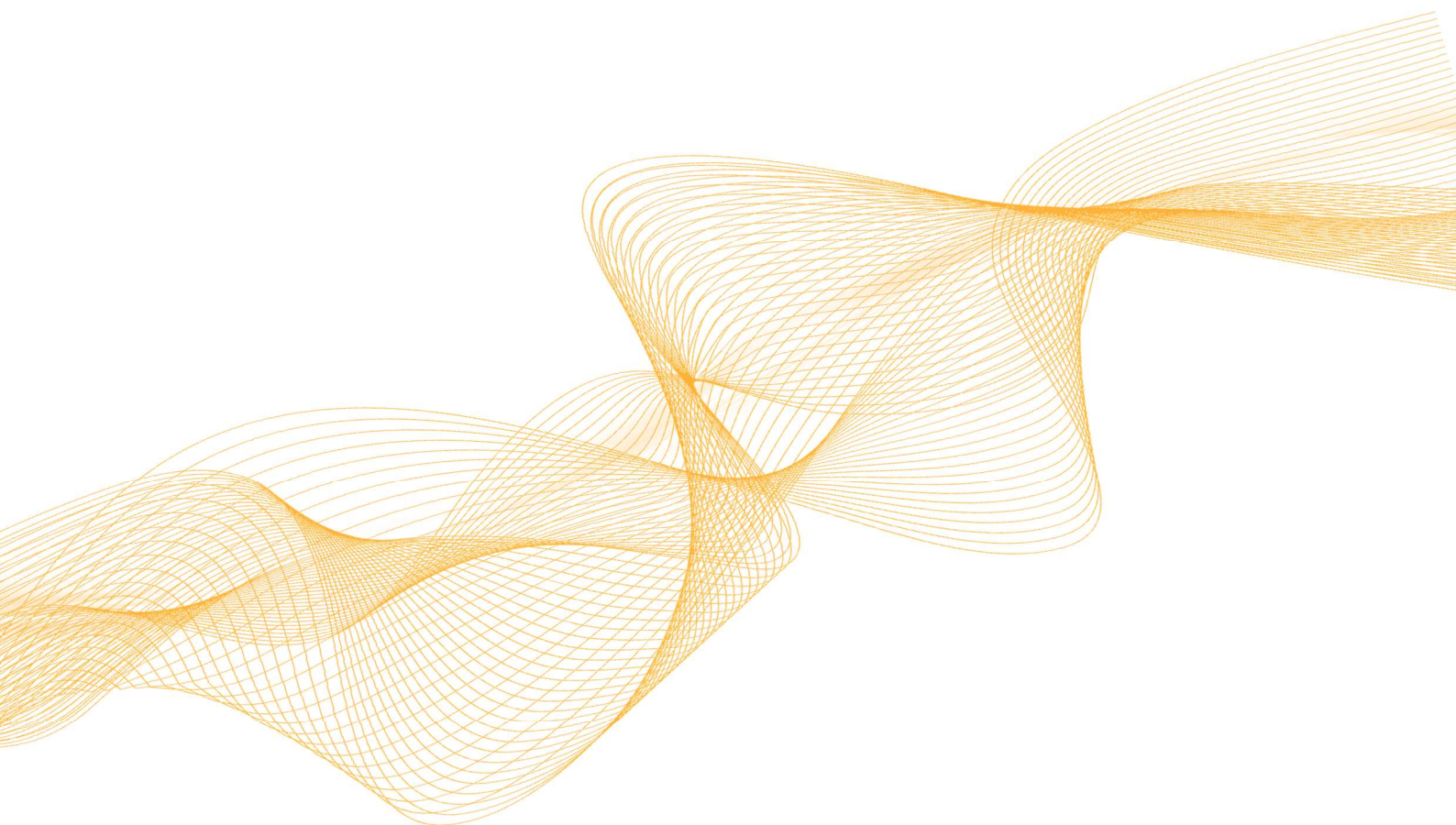




SYMV 图像处理库开发手册



深圳市双翌光电科技有限公司

修订记录:

Rev	Date	Author	Description
1.0	20151027	Shuangyi	SYMV 图像处理库开发手册
1.1	20190823	Shuangyi	完善 SyGauge 部分说明
1.2	20200306	Shuangyi	更新图像处理库新增函数说明
1.3	20200615	Shuangyi	更新图像处理库函数说明
1.4	20200621	Shuangyi	增加了图像模块调用方法
2.0	20240504	Shuangyi	补全对应版本的函数

版权声明

本文档所有权归深圳市双翌光电科技有限公司(后面简称“双翌”)所有；双翌具有本产品及其软件的专利权、版权和其它知识产权。未经授权，任何单位和个人不得直接或者间接地复制、制造、加工、使用本产品及其相关部分。

双翌保留在不事先通知的情况下，修改本手册中的产品和产品规格等文件的权力。

双翌全力维护本文档的正确性，但不承担由于本文档错误或使用本产品不当，所造成直接的、间接的、特殊的、附带的或相应产生的损失或责任。



运动中的机器有危险！使用中有责任在机器中设计有效的出错处理和安全保护机制，故双翌没有义务或责任对此造成的附带的或相应产生的损失责任

联系我们

深圳市双翌光电科技有限公司

地址：深圳市宝安区沙井街道后亭茅洲山工业园工业大厦全至科技创新园科创大厦 2 层 A

电话：86-0755-23712116

传真：86-0755-23020631

公司网址：[Http//www.shuangyi-tech.com](http://www.shuangyi-tech.com)

目 录

版权声明	3
第 1 章 开发说明	14
1.1 开发环境配置	14
1.2 如何调用图像处理相关函数	15
第 2 章 函数说明	16
2.1、SYBASE:General Library 基础库	16
2.1.1 SYROI Classes and Class Members	16
2.1.2 SYGrayImage Classes and Class Members	21
2.1.3 SYColorImage Classes and Class Members	27
2.1.4 SYBase Classes and Class Members	32
2.2、SYColor Classes and Class Members	33
SYColor::ClrGetComponent	33
SYColor::ClrSetComponent	34
SYColor::ImgConvert	34
SYColor::Mean	34
2.3、SYImgproc Classes and Class Members	35
SYImgproc::Threshold	37
SYImgproc::ImgAutoThreshold	37
SYImgproc::GetThreshold	37
SYImgproc::CalOptimizeTheshold	37
SYImgproc::ImgGainOffset	38
SYImgproc::ImgHorizontalMirror	38
SYImgproc::ImgVerticalMirror	38
SYImgproc::ImgXYMirror	38
SYImgproc::ImgHistogramEqualize	39
SYImgproc::ImgRotate	39
SYImgproc::Erode	39
SYImgproc::Dilate	40
SYImgproc::Open	40
SYImgproc::Close	40
SYImgproc::WhiteTopHat	41

SYImgproc::BlackTopHat	41
SYImgproc::Gradient	42
SYImgproc::ImgOper	42
SYImgproc::ImgPixelAverage.....	43
SYImgproc::ImgPixelStdDev.....	43
SYImgproc::ImgPixelStat.....	43
YImgproc::ImgPixelCount.....	44
SYImgproc::ImgTransparent.....	44
SYImgproc::Add	44
SYImgproc::And	45
SYImgproc::Canny	45
SYImgproc::DifferenceAbs.....	45
SYImgproc::Blur	45
SYImgproc::GaussianBlur.....	46
SYImgproc::GetFocusValue.....	46
SYImgproc::ImgNormalize.....	46
SYImgproc::ImgPixelCountNotZero.....	46
SYImgproc::ImgStdDev	47
SYImgproc::Invert	47
SYImgproc::LinearPolar	47
SYImgproc::MedianBlur	47
SYImgproc::Min	48
SYImgproc::PointInRegion.....	48
SYImgproc::Uniformize	48
2.4、SYMch Classes and Class Members.....	49
SYMch::SetMinScore	51
SYMch::SetMinAngle	51
SYMch::SetMaxAngle	51
SYMch::SetMinScale	51
SYMch::SetMaxScale	51
SYMch::SetPivot	52
SYMch::SetMaxPositions	52
SYMch::LearnPattern	52
SYMch::IsLearnt	52
SYMch::Match	52

SYMch::DrawPosition	53
SYMch::DrawPositions	53
SYMch::GetNumPositions	53
SYMch::GetPosition	54
SYMch::SetPivot	54
SYMch::SetContrastMode	54
SYMch::GetContrastMode	55
SYMch::GetNumberLevelOfPyramid.....	55
SYMch::SetCoarseLevel	55
SYMch::GetCoarseLevel	55
SYMch::SetFineLevel	55
SYMch::GetFineLevel	56
SYMch::SetOverLapRate	56
SYMch::GetOverLapRate	56
SYMch::SetTimeOut	56
SYMch::GetTimeOut	56
SYMch::SetGreediness	57
SYMch::GetGreediness	57
2.5、SYBlob Classes and Class Members.....	58
SYBlob::BuildBlobs	59
SYBlob::AnalysisBlobsFeature.....	59
SYBlob::SortBlobs	59
SYBlob::SelectBlobsInRange.....	60
SYBlob::GetNumOfBlob	60
SYBlob::GetFirstBlobPtr.....	60
SYBlob::GetNextBlobPtr	60
SYBlob::GetBlobData	61
SYBlob::GetBlobPtrByCoordinates.....	61
SYBlob::DrawBlob	61
SYBlob::DrawBlobs	62
SYBlob::DrawFeature	62
SYBlob:: RemoveBlob	63
SYBlob::GetAbsoluteThreshold.....	63
SYBlob::SetAbsoluteThreshold.....	63
SYBlob::GetBlobCopyInFullImage.....	63

SYBlob::GetBlobCopy	64
SYBlob::GetThresholdMode.....	64
SYBlob::SetThresholdMode.....	64
SYBlob::GetContour	64
2.6、SYGauge Classes and Class Members.....	65
2.6.1 SYCalibration	65
2.6.2 SYPointGauge.....	70
SYPointGauge::SetPlacement.....	72
SYPointGauge::SetModelCenter.....	72
SYPointGauge::SetModelAngle.....	72
SYPointGauge::SetModelTolerance.....	72
SYPointGauge::GetPlacement.....	73
SYMV::GetModelCenter	73
SYPointGauge::GetModelCenterX.....	73
SYPointGauge::GetModelCenterY.....	73
SYPointGauge::GetModelAngle.....	74
SYPointGauge::GetModelTolerance.....	74
SYPointGauge::GetHitShape.....	74
SYPointGauge::Drag	74
SYPointGauge::Draw	75
SYPointGauge::SetTransitionType.....	75
SYPointGauge::GetTransitionType.....	76
SYPointGauge::SetTransitionChoice.....	76
SYPointGauge::GetTransitionChoice.....	77
SYPointGauge::SetThreshold.....	77
SYPointGauge::GetThreshold.....	77
SYPointGauge::SetSmoothing.....	78
SYPointGauge::GetSmoothing.....	78
SYPointGauge::SetMinArea.....	78
SYPointGauge::GetMinArea.....	79
SYPointGauge::Attach	79
SYPointGauge::Detach	79
SYPointGauge::InImageRegion.....	79
SYPointGauge::Measure	79
SYPointGauge::GetMeasuredPoint.....	80

SYPointGauge::GetNumMeasuredPoints.....	80
SYPointGauge::SetThickness.....	80
SYPointGauge::GetThickness.....	80
SYPointGauge::SetTransitionIndex.....	81
SYPointGauge::GetTransitionIndex.....	81
SYPointGauge::SetDerSize.....	81
SYPointGauge::GetDerSize.....	81
2. 6. 3 SYLineGauge.....	82
SYLineGauge::SetPlacement.....	83
SYLineGauge::SetModelCenter.....	84
SYLineGauge:: SetModelLength.....	84
SYLineGauge:: SetModelTolerance.....	84
SYLineGauge:: SetModelAngle.....	84
SYLineGauge::GetPlacement.....	85
SYLineGauge::GetModelCenter.....	85
SYLineGauge::GetModelCenterX.....	85
SYLineGauge::GetModelCenterY.....	85
SYLineGauge::GetModelLength.....	86
SYLineGauge::GetModelTolerance.....	86
SYLineGauge::GetModelAngle.....	86
SYLineGauge::GetHitShape.....	86
SYLineGauge::Drag	86
SYLineGauge::Draw	87
SYLineGauge::SetTransitionType.....	87
SYLineGauge::GetTransitionType.....	88
SYLineGauge::SetThreshold.....	88
SYLineGauge::GetThreshold.....	88
SYLineGauge::SetSamplingStep.....	88
SYLineGauge::GetSamplingStep.....	88
SYLineGauge::SetSmoothing.....	89
SYLineGauge::GetSmoothing.....	89
SYLineGauge::SetFilterOutlierThreshold.....	90
SYLineGauge::GetFilterOutlierThreshold.....	90
SYLineGauge::SetFilterOutlierNumPass.....	90
SYLineGauge::GetFilterOutlierThreshold.....	90

SYLineGauge::GetNumSamples.....	90
SYLineGauge::GetNumValidSamples.....	91
SYLineGauge::Measure	91
SYLineGauge::GetMeasuredLine.....	91
SYLineGauge::Attach	91
SYLineGauge::Detach	92
SYLineGauge::SetTransitionChoice.....	92
SYLineGauge::GetTransitionChoice.....	92
SYLineGauge::SetMinArea.....	92
SYLineGauge::GetMinArea.....	92
SYLineGauge::SetThickness.....	93
SYLineGauge::GetThickness.....	93
SYLineGauge::SetTransitionIndex.....	93
SYLineGauge::GetTransitionIndex.....	93
SYLineGauge::SetDerSize.....	93
SYLineGauge::GetDerSize.....	94
SYLineGauge::EnableClusterAnalysis.....	94
SYLineGauge::GetClusterAnalysis.....	94
SYLineGauge::SetClusterError.....	94
SYLineGauge::GetClusterError.....	94
2. 6. 4 SYCircleGauge.....	95
SYCircleGauge::SetPlacement.....	97
SYCircleGauge::GetPlacement.....	97
SYCircleGauge::GetHitShape.....	97
SYCircleGauge::Drag	98
SYCircleGauge::Draw	98
SYCircleGauge::SetMeasureDirection.....	99
SYCircleGauge::GetMeasureDirection.....	99
SYCircleGauge::SetTransitionType.....	99
SYCircleGauge::GetTransitionType.....	100
SYCircleGauge::SetThreshold.....	100
SYCircleGauge::GetThreshold.....	100
SYCircleGauge::SetSamplingStep.....	100
SYCircleGauge::GetSamplingStep.....	100
SYCircleGauge::SetSmoothing.....	101

SYCircleGauge::GetSmoothing.....	101
SYCircleGauge::SetFilterOutlierThreshold.....	102
SYCircleGauge::GetFilterOutlierThreshold.....	102
SYCircleGauge::SetFilterOutlierNumPass.....	102
SYCircleGauge::GetFilterOutlierNumPass.....	102
SYCircleGauge::GetNumSamples.....	102
SYCircleGauge::GetNumValidSamples.....	103
SYCircleGauge::Measure	103
SYCircleGauge::GetMeasuredCircle.....	103
SYCircleGauge::Attach	103
SYCircleGauge::Detach	104
SYCircleGauge::SetModelCenter.....	104
SYCircleGauge::SetModelRadius.....	104
SYCircleGauge::SetModelTolerance.....	104
SYCircleGauge::GetModelCenter.....	105
SYCircleGauge::GetModelCenterX.....	105
SYCircleGauge::GetModelCenterY.....	105
SYCircleGauge::GetModelRadius.....	105
SYCircleGauge::GetModelTolerance.....	105
SYCircleGauge::SetMinArea.....	105
SYCircleGauge::GetMinArea.....	106
SYCircleGauge::SetThickness.....	106
SYCircleGauge::GetThickness.....	106
SYCircleGauge::SetTransitionIndex.....	106
SYCircleGauge::GetTransitionIndex.....	106
SYCircleGauge::SetTransitionChoice.....	107
SYCircleGauge::GetTransitionChoice.....	107
SYCircleGauge::SetStartAngle.....	107
SYCircleGauge::GetStartAngle.....	107
SYCircleGauge::SetSpanAngle.....	107
SYCircleGauge::GetSpanAngle.....	108
SYCircleGauge::SetDerSize.....	108
SYCircleGauge::GetDerSize.....	108
SYCircleGauge::EnableClusterAnalysis.....	108
SYCircleGauge::GetClusterAnalysis.....	108

SYCircleGauge::SetClusterError.....	108
SYCircleGauge::GetClusterError.....	109
2. 6. 5 SYRectangleGauge.....	110
SYRectangleGauge::SetPlacement.....	112
SYRectangleGauge::GetPlacement.....	112
SYRectangleGauge::GetHitShape.....	113
SYRectangleGauge::Drag	113
SYRectangleGauge::Draw	113
SYRectangleGauge::SetMeasuredDirection.....	114
SYRectangleGauge::GetMeasuredDirection.....	114
SYRectangleGauge::SetTransitionType.....	115
SYRectangleGauge::GetTransitionType.....	115
SYRectangleGauge::SetThreshold.....	115
SYRectangleGauge::GetThreshold.....	115
SYRectangleGauge::SetSamplingStep.....	116
SYRectangleGauge::GetSamplingStep.....	116
SYRectangleGauge::SetSmoothing.....	116
SYRectangleGauge::GetSmoothing.....	116
SYRectangleGauge::SetFilterOutlierThreshold.....	117
SYRectangleGauge::GetFilterOutlierThreshold.....	117
SYRectangleGauge::SetFilterOutlierNumPass.....	117
SYRectangleGauge::GetFilterOutlierNumPass.....	117
SYRectangleGauge::GetNumSamples.....	118
SYRectangleGauge::GetNumValidSamples.....	118
SYRectangleGauge::Measure.....	118
SYRectangleGauge::GetMeasuredRectangle.....	118
SYRectangleGauge::Attach.....	119
SYRectangleGauge::Detach.....	119
SYRectangleGauge::SetModelCenter.....	119
SYRectangleGauge::SetModelWidth.....	119
SYRectangleGauge::SetModelHeight.....	119
SYRectangleGauge::SetModelTolerance.....	120
SYRectangleGauge::SetModelAngle.....	120
SYRectangleGauge::GetModelCenter.....	120
SYRectangleGauge::GetModelCenterX.....	120

SYRectangleGauge::GetModelCenterY.....	120
SYRectangleGauge::GetModelWidth.....	120
SYRectangleGauge::GetModelHeight.....	121
SYRectangleGauge::GetModelTolerance.....	121
SYRectangleGauge::GetModelAngle.....	121
SYRectangleGauge::SetMinArea.....	121
SYRectangleGauge::GetMinArea.....	121
SYRectangleGauge::SetThickness.....	121
SYRectangleGauge::GetThickness.....	122
SYRectangleGauge::SetTransitionIndex.....	122
SYRectangleGauge::GetTransitionIndex.....	122
SYRectangleGauge::SetTransitionChoice.....	122
SYRectangleGauge::GetTransitionChoice.....	122
SYRectangleGauge::SetDerSize.....	123
SYRectangleGauge::GetDerSize.....	123
2.7、SYFind Classes and Class Members.....	124
SYFind::SetMinScore	126
SYFind::GetMinScore	126
SYFind::SetMinAngle	126
SYFind::SetMaxAngle	126
SYFind::SetAnisotropicScaleXY.....	126
SYFind::SetMinScale	127
SYFind::SetMaxScale	127
SYFind::SetMinScaleX	127
SYFind::SetMaxScaleX	127
SYFind::SetMinScaleY	127
SYFind::SetMaxScaleY	128
SYFind::SetCoarseLevel	128
SYFind::GetCoarseLevel	128
SYFind::SetFineLevel	129
SYFind::GetFineLevel	129
SYFind::SetPivot	129
SYFind::GetPivot	129
SYFind::SetMaxInstances.....	129
SYFind::LearnPatten	130

SYFind::ModifyPattern	130
SYFind::SetFilterLenght.....	131
SYFind::GetFilterLenght.....	131
SYFind::SetGradientThreshold.....	131
SYFind::GetGradientThreshold.....	131
SYFind::SetMinContourSize.....	131
SYFind::GetMinContourSize.....	132
SYFind::IsLearnt	132
SYFind::DrawPosition	132
SYFind::DrawPositions	132
SYFind::DrawModel	133
SYFind::Find	133
SYFind::GetNumPositions.....	133
SYFind::GetPosition	133
SYFind::GetNumberLevelOfPyramid.....	134
SYFind::SetPolarityMode.....	134
SYFind::GetPolarityMode.....	134
SYFind::SetSubpixelMode.....	134
SYFind::GetSubpixelMode.....	135
SYFind::SetOverLapRate	135
SYFind::GetOverLapRate	135
SYFind::SetTimeOut	135
SYFind::GetTimeOut	135
SYFind::SetMinContrast	136
SYFind::GetMinContrast	136
SYFind::AutoMinContrast.....	136
SYFind::SetGreediness	136
SYFind::GetGreediness	136

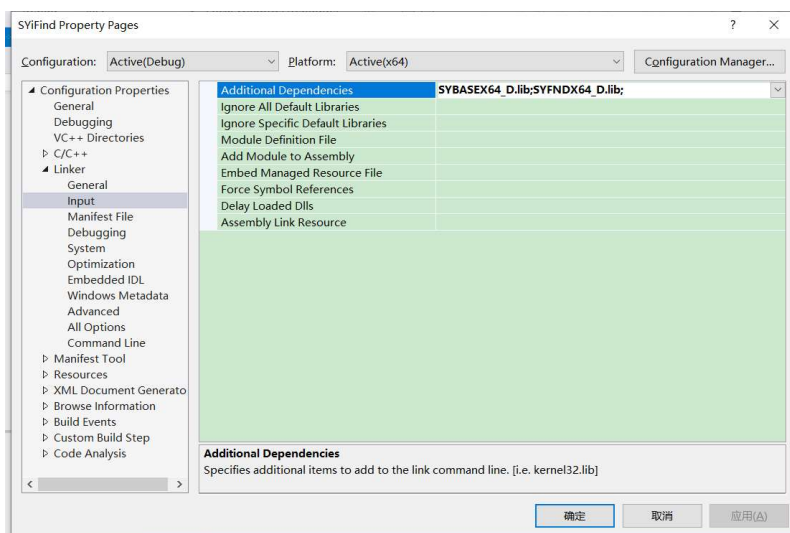
第 1 章 开发说明

1.1 开发环境配置

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。目前图像模块支持 Visual Studio 及 Qt 开发环境。

使用 Visual Studio 来开发图像处理流程：

- 【1】 启动 Visual Studio，新建一个工程；
- 【2】 将安装目录的 Library 文件夹内 dll 文件（包括“opencv_world341d.dll”）、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】 切换到“Link/Input”标签页，在“Additional Dependencies”栏中输入 lib 文件名 SYBASEX64_D.lib，SYBLOBX64_D.lib，SYColorX64_D.lib，SYGAUGEX64_D.lib，SYMCHX64_D.lib，SYFNDX64_D.lib；如下图所示。【注意】除了 base 库必须添加之外，其他库是独立模块分开使用



- 【5】 在应用程序文件中加入函数库头文件的声明，#include “SYBase.h”，“SYBlob.h”，“SYColor.h”，“SYFind.h”，“SYGauge.h”，“SYMch.h”；如下图所示。

【注意】除了 Base 库必须添加之外，其他库是独立模块分开使用



至此，用户就可以在 Visual C++ 中调用库中的函数，开发 SYMV 图像处理应用程序。

【注意】请将 hast_rt.exe 放在应用程序 exe 同一目录下，否则会报 0XC0000044 错误。

1.2 如何调用图像处理相关函数

配置工作准备好后，接下来可以在 cpp 文件中调用函数了，方法如下所示：
声明 SYBase 库的对象，同时声明使用到的图像模块的对象，如下图所示。使用声明的对象即可调用模块的所有函数

```
SYBase      m_SYBase;
CPen        m_BluePen;
CPen        m_GreenPen;
CBrush      brush;

//匹配
SYGrayImage m_ROIImage;
SYFind      m_Find;
```

第 2 章 函数说明

2.1、SYBASE:General Library 基础库

2.1.1 SYROI Classes and Class Members

List of Function 函数列表说明

Function Name(函数名称)	说明
SYRoi::Draw	在窗口中绘制 ROI
SYRoi::SetCursor	设置当前光标坐标，响应鼠标事件
SYRoi::GetHitShape	修改落在 ROI 控制点上的光标形状
SYRoi::Drag	鼠标操作拖动改变 ROI 大小及位置
SYRoi::SetPlacement	设定 ROI 的位置及大小
SYRoi::CopyROItoImg	复制 ROI 区域的局部图像
SYRoi::GetOrgX	获取 ROI 左上角起始点 X 坐标
SYRoi::GetOrgY	获取 ROI 左上角起始点 Y 坐标
SYRoi::GetWidth	获取 ROI 宽度
SYRoi::GetHeight	获取 ROI 高度
SYRoi:: SetOrgX	设置 ROI 左上角起始点 X 坐标
SYRoi:: SetOrgY	设置 ROI 左上角起始点 Y 坐标
SYRoi:: SetWidth	设置 ROI 宽度
SYRoi:: SetHeight	设置 ROI 高度

SYRoi::Draw

```
void Draw(HDC h_Dc, float ZoomX=1.0f, float ZoomY=1.0f, int PanX= 0, int PanY= 0)
```

在窗口中绘制 ROI。

参数：

h_Dc	绘画窗口的设备上下文
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYRoi::SetCursor

```
void SetCursor(int CursorX, int CursorY, float ZoomX, float ZoomY, int PanX, int PanY)
```

设置当前光标坐标，响应鼠标事件。

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYRoi::GetHitShape

```
int GetHitShape(int CursorX, int CursorY, float ZoomX, float ZoomY, int PanX, int PanY)
```

修改落在 ROI 控制点上的光标形状，配合 SetCursor 和 Drag 函数来使用。

此函数判断鼠标是否落在 ROI 控制点上，并修改光标形状，此时可进行拖曳操作

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYRoi::Drag

```
void Drag(int CursorX, int CursorY, float ZoomX, float ZoomY, int PanX, int PanY)
```

鼠标操作拖动改变 ROI 大小及位置

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYRoi::SetPlacement

```
void SetPlacement(int OrgX, int OrgY, int Width, int Height)
```

设定 ROI 的位置及大小

参数：

OrgX	ROI 起始点 X 坐标
OrgY	ROI 起始点 Y 坐标
Width	ROI 宽度
Height	ROI 高度

SYRoi::CopyROItoImg

```
void CopyROItoImg(SYGrayImage &Src, SYGrayImage &Dst)
```

复制 ROI 区域的局部图像

参数：

Src：源图像

Dst：目标图像，得到在源图像上的 ROI 区域内的局部图像

返回值： 无

SYRoi::GetOrgX

```
int GetOrgX()
```

获取 ROI 左上角起始点 X 坐标

返回值：返回 ROI 左上角起始点 X 坐标。

SYRoi::GetOrgY

```
int GetOrgY()
```

获取 ROI 左上角起始点 Y 坐标

返回值：返回 ROI 左上角起始点 Y 坐标。

SYRoi::GetWidth

```
int GetWidth()
```

获取 ROI 宽度

返回值：ROI 的宽度

SYRoi::GetHeight

```
int GetHeight()
```

获取 ROI 高度

返回值：ROI 的高度

SYRoi::SetOrgX

```
void SetOrgX(int OrgX)
```

设置 ROI 左上角起始点 X 坐标

参数：

OrgX ROI 起始点 X 坐标

返回值：无

SYRoi:: SetOrgY

```
void SetOrgY(int OrgY)
```

设置 ROI 左上角起始点 Y 坐标

参数：

OrgY ROI 起始点 Y 坐标

返回值：无

SYRoi:: SetWidth

```
void SetWidth(int Width)
```

设置 ROI 宽度

参数：

Width ROI 宽度

返回值：无

SYRoi:: SetHeight

```
void SetHeight(int Height)
```

设置 ROI 高度

参数：

Height ROI 高度

返回值：无

2.1.2 SYGrayImage Classes and Class Members

List of Function 函数列表说明

Function Name(函数名称)	说明
SYGrayImage::Draw	在窗口中绘画图像
SYGrayImage::Load	载入图像
SYGrayImage::Save	保存图像
SYGrayImage::SetImgPtr	设置目标图像内存地址
SYGrayImage::GetHeight	得到图像高度
SYGrayImage::GetWidth	得到图像宽度
SYGrayImage::SetSize	设置图像大小
SYGrayImage::CopyTo	复制图像
SYGrayImage::GetPixel	得到当前坐标的灰度值
SYGrayImage::SetPixel	设置当前坐标的灰度值
SYGrayImage::CopyFromBuffer	从新复制一个和原图一样的图像
SYGrayImage::GetRowPitch	得到行间距
SYGrayImage::IsValid	判断图像数据是否有效
SYGrayImage::OpenImageDC	获取图像设备上下文 HDC
SYGrayImage::CloseImageDC	关闭图像设备上下文 HDC
SYGrayImage::ReSize	重新设置图像大小
SYGrayImage::SetTo	设置整幅图像的灰度值

SYGrayImage::Draw

```
void Draw(HDC hDC, float ZoomX, float ZoomY, int PanX, int PanY)
```

在窗口中绘画图像。

参数：

hDC	设备上下文
ZoomX	水平方向比例
ZoomY	垂直方向比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYGrayImage::Load

```
void LoadImage(const char* filename)
```

载入路径指向的图像文件。

参数：

filename	图像路径
----------	------

返回值：无

SYGrayImage::Save

```
void SaveImage(const char* filename)
```

将图像保存到指定文件路径中。

参数：

filename	图像路径
----------	------

返回值：无

SYGrayImage::SetImgPtr

```
void SetImgPtr(unsigned int width, unsigned int height, void* ImgBuffer)
```

设置目标图像内存地址

参数：

width	图像宽度
height	图像高度
ImgBuffer	图像数据内存地址

返回值：无

SYGrayImage::GetHeight

```
int GetHeight()
```

得到图像的高度

返回值：返回图像的高度。

SYGrayImage::GetWidth

```
int GetWidth()
```

得到图像的宽度

返回值：返回图像的宽度。

SYGrayImage::SetSize

```
void SetSize(int n32Width, int n32Height)
```

设置图像的大小

参数：

n32Width	图像宽度
----------	------

n32Height	图像高度
-----------	------

SYGrayImage::SetSize

```
void SetSize(SYGrayImage *src)
```

设置图像的大小与源图像一致。

参数：

src	源图像
-----	-----

SYGrayImage::CopyTo

```
void CopyTo(SYGrayImage *Dst)
```

将 SYGrayImage 类的源图像复制到目标图像

参数：

Dst	目标图像
-----	------

返回值： 无

SYGrayImage::GetPixel

```
int GetPixel(int n32X= 0, int n32Y= 0)
```

得到当前坐标的灰度值

参数：

n32X X 像素坐标

n32Y Y 像素坐标

返回值：返回 X 和 Y 像素坐标位置的灰度值

SYGrayImage::SetPixel

```
void SetPixel(int Value, int n32X= 0, int n32Y= 0)
```

设置当前坐标的灰度值

参数：

Value 灰度值

n32X X 像素坐标

n32Y Y 像素坐标

SYGrayImage:: CopyFromBuffer

```
void CopyFromBuffer(unsigned int width, unsigned int height, void* ImgBuffer)
```

从新复制一个和原图一样的图像

参数：

width 图像宽度

height 图像高度

ImgBuffer 图像数据内存地址

返回值：无

SYGrayImage:: GetRowPitch

```
int GetRowPitch()
```

得到行间距

返回值：返回行间距。

SYGrayImage::IsValid

`bool IsValid()`

判断图像数据是否有效

返回值：返回 false 表示源图像无数据，true 表示源图像有数据

Example:

```
SYGrayImage m_SyGrayImg;
```

```
//.....
```

```
if(!m_SyGrayImg.IsValid ())          //图像无数据则返回
```

```
    return;
```

```
else
```

```
{
```

```
    //.....          // m_SyGrayImg有效，后续进行相应的
```

```
    图像处理
```

```
}
```

SYGrayImage::OpenImageDC

`HDC OpenImageDC()`

打开图像设备上下文 HDC，在图像上进行绘画或文字输出等。

返回值：HDC

SYGrayImage::CloseImageDC

`void CloseImageDC()`

关闭图像设备上下文 HDC。

返回值：HDC

SYGrayImage:: ReSize

```
void ReSize(unsigned int width, unsigned int height, SYGrayImage* dst)
```

把当前图像缩放为 width * height 大小并保存到 dst 变量中。

参数：

width	目标图像宽度
height	目标图像高度
dst	目标图像变量名

返回值：无

SYGrayImage:: SetTo

```
void SetTo(unsigned char pixelValue)
```

把整幅图像设置为灰度值 pixelValue

参数：

pixelValue	目标灰度值
------------	-------

返回值：无

2.1.3 SYColorImage Classes and Class Members

List of Function 函数列表说明

Function Name(函数名称)	说明
<code>SYColorImage::Draw</code>	在窗口中绘画图像
<code>SYColorImage::Load</code>	载入图像
<code>SYColorImage::Save</code>	保存图像
<code>SYColorImage::SetImgPtr</code>	设置目标图像内存地址
<code>SYColorImage::GetHeight</code>	得到图像高度
<code>SYColorImage::GetWidth</code>	得到图像宽度
<code>SYColorImage::SetSize</code>	设置图像大小
<code>SYColorImage::CopyTo</code>	复制图像
<code>SYColorImage::GetPixel</code>	得到当前坐标的灰度值
<code>SYColorImage::SetPixel</code>	设置当前坐标的灰度值
<code>SYColorImage::CopyFromBuffer</code>	从新复制一个和原图一样的图像
<code>SYColorImage::GetRowPitch</code>	得到行间距
<code>SYColorImage::IsValid</code>	判断图像数据是否有效
<code>SYColorImage::OpenImageDC</code>	获取图像设备上下文 HDC
<code>SYColorImage::CloseImageDC</code>	关闭图像设备上下文 HDC
<code>SYColorImage::ReSize</code>	重新设置图像大小
<code>SYColorImage::SetTo</code>	设置整幅图像的灰度值

SYColorImage::Draw

```
void Draw(HDC hDC, float ZoomX, float ZoomY, int PanX, int PanY)
```

在窗口中绘画图像。

参数：

hDC	设备上下文
ZoomX	水平方向比例
ZoomY	垂直方向比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYColorImage::Load

```
void LoadImage(const char* filename)
```

载入路径指向的图像文件。

参数：

filename 图像路径

返回值：无

SYColorImage::Save

```
void SaveImage(const char* filename)
```

将图像保存到指定文件路径中。

参数：

filename 图像路径

返回值：无

SYColorImage::SetImgPtr

```
void SetImgPtr(unsigned int width, unsigned int height, void* ImgBuffer)
```

设置目标图像内存地址

参数：

width	图像宽度
height	图像高度
ImgBuffer	图像数据内存地址

返回值：无

SYColorImage::GetHeight

```
int GetHeight()
```

得到图像的高度

返回值：返回图像的高度。

SYColorImage::GetWidth

```
int GetWidth()
```

得到图像的宽度

返回值：返回图像的宽度。

SYColorImage::SetSize

```
void SetSize(int n32Width, int n32Height)
```

设置图像的大小

参数：

n32Width 图像宽度

n32Height 图像高度

SYColorImage::SetSize

```
void SetSize(SYColorImage *src)
```

设置图像的大小与源图像一致。

参数：

src 源图像

SYColorImage::CopyTo

```
void CopyTo(SYColorImage *Dst)
```

将 SYColorImage 类的源图像复制到目标图像

参数：

Dst 目标图像

返回值： 无

SYColorImage::GetPixel

```
int GetPixel(int n32X= 0, int n32Y= 0)
```

得到当前坐标的灰度值

参数：

n32X X 像素坐标

n32Y Y 像素坐标

返回值：返回 X 和 Y 像素坐标位置的灰度值

SYColorImage::SetPixel

```
void SetPixel(int Value, int n32X= 0, int n32Y= 0)
```

设置当前坐标的灰度值

参数：

Value 灰度值

n32X X 像素坐标

n32Y Y 像素坐标

SYColorImage:: CopyFromBuffer

```
void CopyFromBuffer(unsigned int width, unsigned int height, void* ImgBuffer)
```

从新复制一个和原图一样的图像

参数：

width 图像宽度

height 图像高度

ImgBuffer 图像数据内存地址

返回值：无

SYColorImage:: GetRowPitch

```
int GetRowPitch()
```

得到行间距

返回值：返回行间距。

SYColorImage::IsValid

bool IsValid()

判断图像数据是否有效

返回值：返回 false 表示源图像无数据，true 表示源图像有数据

Example:

```
SYColorImage    m_SyColorImg;
```

```
//.....
```

```
if(!m_SyColorImg. IsValid ())          //图像无数据则返回
```

```
    return;
```

```
else
```

```
{
```

```
    //.....          // m_SyColorImg有效，后续进行相应的
```

```
    图像处理
```

```
}
```

SYColorImage::OpenImageDC

HDC OpenImageDC()

打开图像设备上下文 HDC，在图像上进行绘画或文字输出等。

返回值：HDC

SYColorImage::CloseImageDC

void CloseImageDC()

关闭图像设备上下文 HDC。

返回值：HDC

SYColorImage::ReSize

void ReSize(unsigned int width, unsigned int height, SYColorImage* dst)

把当前图像缩放为 width * height 大小并保存到 dst 变量中。

参数：

width 目标图像宽度

height 目标图像高度

dst 目标图像变量名

返回值：无

2.1.4 SYBase Classes and Class Members

List of Function 函数列表说明

Function Name(函数名称)	说明
<code>SYBase::StartTiming</code>	开始计时
<code>SYBase::StopTiming</code>	计时结束

`SYBase::StartTiming`

```
void StartTiming(void)
```

开始计时

`SYBase::StopTiming`

```
int StopTiming(unsigned int resolution)
```

计时结束，返回时间。

参数：

resolution 时间分辨率 1000：返回毫秒 1000000：返回微秒 其它默认毫秒

Example:

```
StartTiming();
```

```
// Process
```

```
...
```

```
// Elapsed time, in milliseconds
```

```
n32Elapsed= StopTiming(1000);
```

StopTiming 和 StartTiming 要配合使用，用于得到程序段执行所消耗的时间。

2.2、SYColor Classes and Class Members

List of Function 函数列表说明

Function Name(函数名称)	说明
<code>SYColor::ClrGetComponent</code>	从彩色图像中分离出来 RGB 分量
<code>SYColor::ClrSetComponent</code>	将 RGB 分量组合成彩色图像
<code>SYColor::ImgConvert</code>	图像转换
<code>SYColor::Mean</code>	求 RGB 各分量的平均值

SYColor::ClrGetComponent

`void ClrGetComponent(SYColorImage*ColorSrc, SYGrayImage*Dst_Component, _Compon Type)`

从彩色图像中分离出来 RGB 分量

参数：

ColorSrc 彩色源图像

Dst_Component 目标分量图像

Type RGB 分量

返回值：无

Type	描述
<code>BlueChanel</code>	蓝色分量
<code>GreenChanel</code>	绿色分量
<code>RedChanel</code>	红色分量

`enum _ComponentIndex`

```
{
    BlueChanel = 0,
    GreenChanel,
    RedChanel,
};
```

SYColor::ClrSetComponent

```
void ClrSetComponent(SYGrayImage *Src_Component_B, SYGrayImage *Src_Component_G,  
SYGrayImage*Src_Component_R, SYGrayImage *ColorDst)
```

将 RGB 分量组合成彩色图像

参数：

Src_Component_B	图像蓝色分量灰度图
Src_Component_G	图像绿色分量灰度图
Src_Component_R	图像红色分量灰度图
ColorDst	彩色目标图

返回值：无

SYColor::ImgConvert

```
void ImgConvert(SYColorImage*Src, SYGrayImage*Dst)
```

色彩空间转化，将彩色图像 Src 转换为灰度图像 Dst。

参数：

Src	源图像
Dst	转换的目标图像

返回值：无

```
void ImgConvert(SYGrayImage*Src, SYColorImage*Dst)
```

色彩空间转化，将灰度图像 Src 转换为彩色图像 Dst。

参数：

Src	源图像
Dst	转换的目标图像

返回值：无

SYColor::Mean

```
void Mean(SYColorImage*Src, float &R, float &G, float &B)
```

求彩色图像 Src 的各分量平均值

参数：

Src	源图像
R、G、B	各颜色分量的平均值

返回值：无

2.3、SYImgproc Classes and Class Members

List of Function 函数列表说明

Function Name(函数名称)	说明
SYImgproc::Threshold	二值化
SYImgproc::ImgAutoThreshold	自动二值化
SYImgproc::GetThreshold	得到二值化的参数值
SYImgproc::CalOptimizeTheshold	自动计算最优化的二值化值
SYImgproc::ImgGainOffset	图像增益和对比度参数设定
SYImgproc::ImgHorizontalMirror	图像水平镜像
SYImgproc::ImgVerticalMirror	图像垂直镜像
SYImgproc::ImgXYMirror	水平和垂直均进行镜像
SYImgproc::ImgHistogramEqualize	图像均衡化
SYImgproc::ImgRotate	图像旋转
SYImgproc::Open	形态学开运算处理
SYImgproc::Close	形态学开运算处理
SYImgproc::Erode	形态学腐蚀处理
SYImgproc::Dilate	形态学膨胀处理
SYImgproc::WhiteTopHat	形态学顶帽（礼帽）运算
SYImgproc::BlackTopHat	形态学黑帽运算
SYImgproc::Gradient	形态学梯度处理
SYImgproc::ImgOper	图像算数与逻辑运算
SYImgproc::ImgPixelAverage	像素均值统计
SYImgproc::ImgPixelStdDev	图像方差统计
SYImgproc::ImgPixelStat	图像像素最小最大值及所在坐标
SYImgproc::ImgPixelCount	像素计数，统计不同区域段像素个数
SYImgproc::ImgTransparent	图像重合运算处理
SYImgproc::Add	灰度图像相加
SYImgproc::And	灰度图像与掩码图像相与操作
SYImgproc::Canny	轮廓边缘 Canny 处理
SYImgproc::DifferenceAbs	求两幅灰度图像的误差绝对值
SYImgproc::Blur	平均值模糊滤波

SYImgproc::GaussianBlur	高斯模糊滤波
SYImgproc::GetFocusValue	求灰度图像的聚焦质量水平
SYImgproc::ImgNormalize	图像归一化
SYImgproc::ImgPixelCountNotZero	图像非零个数统计
SYImgproc::ImgStdDev	图像标准偏差
SYImgproc::Invert	图像反转
SYImgproc::LinearPolar	图像坐标系转换，极坐标转换笛卡尔坐标
SYImgproc::MedianBlur	中值模糊滤波
SYImgproc::Min	求两幅灰度图像的最小值
SYImgproc::PointInRegion	点与区域的关系检测
SYImgproc::Uniformize	图像均衡化，或一致化

SYImgproc::Threshold

```
void Threshold(SYGrayImage &src, SYGrayImage &dst, double theval)
```

对源图像进行二值化

参数：

src	源图像
dst	目标图像
theval	二值化阈值

SYImgproc::ImgAutoThreshold

```
void ImgAutoThreshold(SYGrayImage &src, SYGrayImage &dst)
```

自动二值化

参数：

src	源图像
dst	目标图像

SYImgproc::GetThreshold

```
int GetThreshold(SYGrayImage &Src)
```

得到二值化阈值

参数：

src 源图像

返回值：返回二值化的阈值。

SYImgproc::CalOptimizeTheshold

```
int CalOptimizeTheshold(SYGrayImage & _src)
```

通过大律法计算二值化阈值

参数：

src 源图像

返回值：返回二值化的阈值。

SYImgproc::ImgGainOffset

```
void ImgGainOffset(SYGrayImage &src, SYGrayImage &dst, double GainVaule, double OffsetVaule)
```

调整图像的增益和偏置值，进而调整图像的对比度(Contrast)和亮度(brightness)。

参数：

src	源图像
dst	目标图像
GainValue	增益值，GainVaule 值在 0 和 3 区间
OffsetVaule	偏置值，OffsetVaule 的值范围设置在-255~255 区间

SYImgproc::ImgHorizontalMirror

```
void ImgHorizontalMirror(SYGrayImage &src, SYGrayImage &dst)
```

图像水平镜像

参数：

src	源图像
dst	目标图像

SYImgproc::ImgVerticalMirror

```
void ImgVerticalMirror(SYGrayImage &src, SYGrayImage &dst)
```

图像垂直镜像

参数：

src	源图像
dst	目标图像

SYImgproc::ImgXYMirror

```
void ImgXYMirror(SYGrayImage &src, SYGrayImage &dst)
```

图像水平和垂直方向进行镜像处理

参数：

src	源图像
dst	目标图像

SYImgproc::ImgHistogramEqualize

```
void ImgHistogramEqualize(SYGrayImage &src, SYGrayImage &dst)
```

直方图均衡化是图像处理领域中，利用图像直方图对对比度进行调整的方法。这种方法通常用来增加许多图像的局部对比度，尤其是当图像的有用数据的对比度相当接近的时候。通过这种方法，亮度可以更好地在直方图上分布。这样就可以用于增强局部的对比度而不影响整体的对比度，直方图均衡化通过有效地扩展常用的亮度来实现这种功能。

直方图均衡化优缺点：

这种方法对于背景和前景都太亮或者太暗的图像非常有用，这种方法尤其是可以带来 X 光图像中更好的骨骼结构显示以及曝光过度或者不足照片中更好的细节。这种方法的一个主要优势是它的可逆操作，如果已知均衡化函数，那么就可以恢复原始的直方图，并且计算量小。

这种方法的一个缺点是它对处理的数据不加选择，它可能会增加背景噪声的对比度并且降低有用信号的对比度；变换后图像的灰度级减少，某些细节消失；某些图像，如直方图有高峰，经处理后对比度不自然的过分增强。

参数：

src	源图像
dst	目标图像

SYImgproc::ImgRotate

```
void ImgRotate(SYGrayImage &src, SYGrayImage &dst, float degree)
```

图像旋转处理

参数：

src	源图像
dst	目标图像
degree	旋转角度

SYImgproc::Erode

```
void Erode(SYGrayImage &src, SYGrayImage &dst, int val)
```

图像腐蚀运算，腐蚀可以用来消除小且无意义的像素。

参数：

src	源图像
dst	目标图像
val	腐蚀运算数值

SYImgproc::Dilate

```
void Dilate(SYGrayImage &src, SYGrayImage &dst, int val)
```

图像膨胀运算，膨胀可以用来填充物体边界，合并周边的像素。

参数：

src	源图像
dst	目标图像
val	膨胀运算数值

SYImgproc::Open

```
void Open(SYGrayImage &src, SYGrayImage &dst, int val)
```

先腐蚀后膨胀的过程称为开运算，图像开运算处理，此功能主要用来消除小物体、平滑较大物体的边界，且并不明显改变其面积

参数：

src	源图像
dst	目标图像
val	开运算数值

SYImgproc::Close

```
void Close(SYGrayImage &src, SYGrayImage &dst, int val)
```

先膨胀后腐蚀的过程称为闭运算，图像闭运算处理，主要用来填充物体内部细小空洞、连接邻近物体、平滑其边界的同时并不明显改变其面积。

参数：

src	源图像
dst	目标图像
val	开运算数值

SYImgproc::WhiteTopHat

```
void WhiteTopHat(SYGrayImage &src, SYGrayImage &dst, int val)
```

顶帽运算（Top Hat）又常常被译为“礼帽”运算，是源图像与“开运算”的结果图之差，数学表达式如下：

$$\text{dst} = \text{tophat}(\text{src}, \text{element}) = \text{src} - \text{open}(\text{src}, \text{element})$$

因为开运算带来的结果是放大了裂缝或者局部低亮度的区域，因此，从原图中减去开运算后的图，得到的效果图突出了比原图轮廓周围的区域更明亮的区域，且这一操作和选择的核的大小相关。

顶帽运算往往用来分离比邻近点亮一些的斑块。当一幅图像具有大幅的背景的时候，而微小物品比较有规律的情况下，可以使用顶帽运算进行背景提取。

参数：

src	源图像
dst	目标图像
val	顶帽运算数值

SYImgproc::BlackTopHat

```
void BlackTopHat(SYGrayImage &src, SYGrayImage &dst, int val)
```

黑帽（Black Hat）运算为“闭运算”的结果图与源图像之差。数学表达式为：

$$\text{dst} = \text{blackhat}(\text{src}, \text{element}) = \text{close}(\text{src}, \text{element}) - \text{src}$$

黑帽运算后的效果图突出了比原图轮廓周围的区域更暗的区域，且这一操作和选择的核的大小相关。所以，黑帽运算用来分离比邻近点暗一些的斑块。

参数：

src	源图像
dst	目标图像
val	黑帽运算数值

SYImgproc::Gradient

```
void Gradient(SYGrayImage &src, SYGrayImage &dst, int val)
```

形态学梯度（Morphological Gradient）为膨胀图与腐蚀图之差，数学表达式如下：

```
dst = morph_grad(src, element) = dilate(src, element) - erode(src, element)
```

对二值图像进行这一操作可以将像素块（blob）的边缘突出。我们可以用形态学梯度来保留物体的边缘轮廓，

参数：

src	源图像
dst	目标图像
val	形态学梯度运算数值

SYImgproc::ImgOper

```
void ImgOper(_ArithLogicOp eOperation, SYGrayImage &src, SYGrayImage &dst)
```

```
void ImgOper(_ArithLogicOp eOperation, SYGrayImage &src1, SYGrayImage &src2, SYGrayImage &dst)
```

算数与逻辑运算，

参数：

src	源图像
src1	源图像 1
dst	目标图像
eOperation	逻辑运算算子

eOperation 算子	描述
Copy_OP	图像复制
Invert_OP	图像翻转
Add_OP	两个图像矩阵相加
Subtract_OP	两个图像矩阵相减
Multiply_OP	两个图像矩阵相乘
Divide_OP	两个图像矩阵相除
Minimum_OP	两个图像矩阵比较，取矩阵最小值
Maximum_OP	两个图像矩阵比较，取矩阵最大值

```
enum _ArithLogicOp
{
    Copy_OP = 0,
    Invert_OP,
    Add_OP,
    Subtract_OP,
    Multiply_OP,
    Divide_OP,
    Minimum_OP,
    Maximum_OP
};
```

SYImgproc::ImgPixelAverage

```
double ImgPixelAverage(SYGrayImage &src)
```

统计像素平均值。

参数：

src 源图像

返回：返回源图像像素均值。

SYImgproc::ImgPixelStdDev

```
double    ImgPixelStdDev(SYGrayImage &src)
```

图像方差统计。

参数：

src 源图像

返回：返回源图像像素方差值。

SYImgproc::ImgPixelStat

```
void ImgPixelStat(SYGrayImage &src, double& MinPixVal, double& MaxPixVal, SYMV::Point2i  
& minLoc, SYMV::Point2i & maxLoc)
```

返回图像最小灰度和最大灰度值，以及它们所在的位置。

参数：

src 源图像

MinPixVal 最小灰度值

MaxPixVal 最大灰度值

minLoc 最小灰度值所在的像素位置

maxLoc 最大灰度值所在的像素位置

YImgproc::ImgPixelCount

```
void ImgPixelCount(SYGrayImage &src, unsigned char un8LowThreshold, unsigned char  
un8HighThreshold, signed int& n32PixelsBelow, signed int& n32PixelsBetween, signed int&  
n32PixelsAbove)
```

参数：

src	源图像
un8LowThreshold	下阈值
un8HighThreshold	上阈值
n32PixelsBelow	灰度值小于下阈值像素个数
n32PixelsBetween	灰度值在下阈值、上阈值之间的像素个数
n32PixelsAbove	灰度值大于上阈值的像素个数

SYImgproc::ImgTransparent

```
void ImgTransparent(SYGrayImage &Src1, double alpha, SYGrayImage &Src2, SYGrayImage &Dst)
```

图像透明重叠效果。

参数：

Src1	源图像 1
Src2	源图像 2
alpha	图像透明参数系数，参数在 0 和 1 范围之内。
Dst	目标图像

使用方法：

程序可以根据输入 alpha 透明系数值，从而影响两幅图片所显示的情况。alpha 值在 0 和 1 之间，alpha 值接近 0 的时候，显示的图片以 Src2 为主，当 alpha 值接近 1 的时候，显示的图片以 Src1 为主。

使用此函数，将图片进行透明化重叠处理。

SYImgproc::Add

```
void Add(SYGrayImage *Src1, SYGrayImage *Src2, SYGrayImage *Dst)
```

两幅图像相加，并保存到目标图像中；自溢出处理，即相加大于 255 时，值保持为 255。

参数：

Src1	源图像 1
Src2	源图像 2
Dst	目标图像

SYImgproc::And

```
void And(SYGrayImage *Src, SYGrayImage *mask, SYGrayImage *Dst)
```

图像与掩码进行相与操作, 即保留掩码有效区域的图像信息, 掩码无效区域全部置零。

参数:

Src	源图像
mask	掩码图
Dst	目标图像

SYImgproc::Canny

```
void Canny(SYGrayImage *src, SYGrayImage *dst, int Threshold1, int Threshold2)
```

采用著名的 Canny 算法, 对灰度图像进行边缘分析。

参数:

Src	源图像
Dst	边缘图像
Threshold1	低阈值
Threshold2	高阈值

SYImgproc::DifferenceAbs

```
void DifferenceAbs(SYGrayImage *Src1, SYGrayImage *Src2, SYGrayImage *Dst)
```

求两幅图像的误差绝对值。

参数:

Src1	源图像 1
Src2	源图像 2
Dst	目标图像

SYImgproc::Blur

```
void Blur(SYGrayImage *Src, SYGrayImage *Dst, unsigned int kernelSize)
```

对图像进行平均值模糊滤波, 求取平均值范围由 kernelSize 决定。

参数:

Src	源图像
Dst	目标图像
kernelSize	求平均值的范围

SYImgproc::GaussianBlur

```
void GaussianBlur(SYGrayImage *Src, SYGrayImage *Dst, unsigned int kernelSize)
```

对图像进行高斯模糊滤波, 高斯核大小由 kernelSize 决定。

参数:

Src	源图像
Dst	目标图像
kernelSize	求平均值的范围

SYImgproc::GetFocusValue

```
double GetFocusValue(SYGrayImage *Src)
```

灰度图像聚焦质量评估, 返回评分越大, 图像聚焦水平越好。

参数:

Src	源图像
-----	-----

SYImgproc::ImgNormalize

```
void ImgNormalize(SYGrayImage *Src, SYGrayImage *Dst, double minV, double maxV)
```

对图像进行归一化处理, 即将图像的灰度值搬移到[minV maxV]范围

参数:

Src	源图像
Dst	目标图像
minV、maxV	目标灰度值范围

SYImgproc::ImgPixelCountNotZero

```
int ImgPixelCountNotZero(SYGrayImage *Src)
```

统计灰度图像的非零像素个数。

参数:

Src	源图像
-----	-----

SYImgproc::ImgStdDev

```
double    ImgStdDev(SYGrayImage *Src)
```

求图像的标准偏差。

参数：

Src 源图像

SYImgproc::Invert

```
void Invert(SYGrayImage *Src, SYGrayImage *Dst)
```

对图像进行反转操作。

参数：

Src 源图像

Dst 目标图像

SYImgproc::LinearPolar

```
void LinearPolar(SYGrayImage *Src, SYGrayImage *Dst, SYMV::Point2f center, double minR, double maxR, double StartAngle, double SpanAngle)
```

对图像进行坐标系转换，从极坐标转换到笛卡尔坐标系。即将指定环形区域拉展成矩形。

参数：

Src 源图像

Dst 目标图像

Center 环形区域的圆心

MinR、MaxR 环形的内外半径

StartAngle 环形区域的起始角度

SpanAngle 环形区域的角度范围

SYImgproc::MedianBlur

```
void MedianBlur(SYGrayImage *Src, SYGrayImage *Dst, unsigned int kernelSize)
```

对图像进行中值模糊滤波，求中值的范围由 kernelSize 决定。

参数：

Src 源图像

Dst 目标图像

kernelSize 求中值的区域范围

SYImgproc::Min

```
void Min(SYGrayImage *Src1, SYGrayImage *Src2, SYGrayImage *Dst)
```

求两幅图像的最小值，保存到目标图像中。

参数：

Src1	源图像 1
Src2	源图像 2
Dst	目标图像

SYImgproc::PointInRegion

```
bool PointInRegion(vector<SYMV::Point2f> p, SYMV::Point2f pt)
```

检测点 pt 与区域 P 的位置关系，在区域 P 内即为 True，反之为 False。

参数：

p	围成多边形区域的点坐标
pt	被检测点

SYImgproc::Uniformize

```
void Uniformize(SYGrayImage *Src, SYGrayImage *lightImg, SYGrayImage *darkImg,  
SYGrayImage *dstImg)
```

根据 lightImg、darkImg 参考图，对 Src 图像进行一致性处理，使 Src 明暗变化连续性。

参数：

Src	源图像
lightImg	亮图像
darkImg	暗图像
Dst	目标图像

2.4、SYMch Classes and Class Members

List of Function 函数列表说明

Function Name(函数名称)	说明
<code>SYMch::SetMinScore</code>	设置模板匹配相似度分数
<code>SYMch::SetMinAngle</code>	设置模板匹配允许的最小角度
<code>SYMch::SetMaxAngle</code>	设置模板匹配允许的最大角度
<code>SYMch::SetMinScale</code>	设置模板匹配允许的最小缩放比例系数
<code>SYMch::SetMaxScale</code>	设置模板匹配允许的最大缩放比例系数
<code>SYMch::SetMaxPositions</code>	设置模板匹配个数
<code>SYMch::SetPivot</code>	设置匹配输出结果偏移量
<code>SYMch::LearnPattern</code>	学习模板
<code>SYMch::Match</code>	模板匹配
<code>SYMch::DrawPosition</code>	根据索引号画出匹配到的目标
<code>SYMch::DrawPositions</code>	画出匹配到的所有目标
<code>SYMch::GetNumPositions</code>	得到匹配到的目标个数
<code>SYMch::GetPosition</code>	得到目标信息（匹配分数、坐标、角度等）
<code>SYMch::IsLearnt</code>	判断是否已经进行模板学习
<code>SYMch::SetPivot</code>	设置模板匹配输出结果偏移量
<code>SYMch::SetContrastMode</code>	设置搜索目标的极性
<code>SYMch::GetContrastMode</code>	获取搜索目标的极性
<code>SYMch::GetNumberLevelOfPyramid</code>	获取内部金字塔层数
<code>SYMch::GetCoarseLevel</code>	获取粗糙度
<code>SYMch::SetCoarseLevel</code>	设置粗糙度
<code>SYMch::GetFineLevel</code>	获取精细度
<code>SYMch::SetFineLevel</code>	设置精细度
<code>SYMch::SetOverLapRate</code>	设置重叠率
<code>SYMch::GetOverLapRate</code>	获取重叠率
<code>SYMch::SetTimeOut</code>	设置超时范围
<code>SYMch::GetTimeOut</code>	获取超时范围

<code>SYMch::SetGreediness</code>	设置启发率
<code>SYMch::GetGreediness</code>	获取启发率

SYMch::SetPivot

void SetPivot(float X, float Y)

设置模板中心点。

参数：

X	中心点 X 坐标
Y	中心点 Y 坐标

SYMch::SetMaxPositions

void SetMaxPositions(int MaxPositions)

设置最大匹配个数。

参数：

MaxPositions	匹配模板数量
--------------	--------

SYMch::LearnPattern

bool LearnPattern(SYGrayImage &Pat)

模板学习。

参数：

Pat	模板
-----	----

SYMch::IsLearnt

bool IsLearnt()

判断是否已经进行模板学习，true 为已经进行模板学习，false 则为未进行模板学习。

SYMch::Match

void Match(SYGrayImage &SrcImg)

模板匹配。

参数：

SrcImg	源图像
--------	-----

SYMch::DrawPosition

```
void DrawPosition(HDC dc, int Idx, float ZoomX, float ZoomY, int PanX, int PanY)
```

根据索引号画出匹配到的目标

参数：

dc	设备上下文
Idx	匹配到的目标索引号
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYMch::DrawPositions

```
void DrawPositions(HDC dc, float ZoomX, float ZoomY, int PanX, int PanY)
```

画出所有匹配到的目标

参数：

dc	设备上下文
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYMch::GetNumPositions

```
int GetNumPositions()
```

返回值：返回匹配到的目标个数

SYMch::GetPosition

`MatchPosition*` `GetPosition(int Index)`

返回目标的参数（如匹配分数、坐标、角度等）

参数：

Index 匹配到的模板索引号

返回值：MatchPosition，返回值里边包含匹配到的中心点，角度和相似度等相关信息。

`typedef struct MatchPosition`

```
{  
    float      CenterX;  
    float      CenterY;  
    float      Angle;  
    float      Score;  
    float      Scale;  
    bool       bSel;  
}
```

`}MatchPosition;`

SYMch::SetPivot

`void SetPivot(float X, float Y)`

设置模板匹配输出结果 offset

参数：

x 模板匹配输出结果 x 方向的 offset

y 模板匹配输出结果 y 方向的 offset

返回值：无

SYMch::SetContrastMode

`void SetContrastMode(SYMCH_CONTRAST_MODE contrastmode)`

设置搜索目标的极性。

参数：

contrastmode 极性枚举类型变量

极性枚举类型如下：

`enum SYMCH_CONTRAST_MODE`

```
{  
    SYMchContrastMode_Normal,  
    SYMchContrastMode_Inverse,  
    SYMchContrastMode_Any  
};
```

`};`

SYMch::GetContrastMode

`SYMCH_CONTRAST_MODE` `GetContrastMode()`

获取搜索目标的极性。

返回值：极性枚举类型变量

SYMch::GetNumberOfPyramid

`int` `GetNumberOfPyramid()`

获取内部建立金字塔的层数。

SYMch::SetCoarseLevel

`void` `SetCoarseLevel(int Level)`

设置粗糙度，范围[0 9]；粗糙度，表示从金字塔图像的第几层开始进行搜索；数值越大，速度越快，但丢失目标的机率也增大。反之则反。

参数：

Level 粗糙度

SYMch::GetCoarseLevel

`int` `GetCoarseLevel()`

获取粗糙度数值。

SYMch::SetFineLevel

`void` `SetFineLevel(int Level)`

设置精细度，范围[0 9]，且小于等于粗糙度；精细度，表示从金字塔图像第几层结束搜索并输出结果。数值越大，搜索速度越快，但精度会降低。反之则反。

参数：

Level 精细度

SYMch::GetFineLevel

```
int GetFineLevel( )
```

获取精细度数值。

SYMch::SetOverLapRate

```
void SetOverLapRate(float overlapratio)
```

设置重叠率。重叠率描述两个目标互相重叠面积与整个目标面积的比例。当两个目标重叠比例大于重叠率时，分数小者将被从目标结果序列剔除。

参数：

overlapratio 重叠率

SYMch::GetOverLapRate

```
float GetOverLapRatio( )
```

获取重叠率。

SYMch::SetTimeOut

```
void SetTimeOut(unsigned int MStime)
```

设置超时时间，单位毫秒，数值范围[10 10000]。

参数：

MStime 超时时间

SYMch::GetTimeOut

```
unsigned int GetTimeOut( )
```

获取超时时间。

SYMch::SetGreediness

```
void SetGreediness(float greediness)
```

设置启发率数值，范围[0 1]。数值越小，搜索时间越长，但不容易丢失目标。数值越大，搜索时间越快，但有机率丢失目标。

参数：

greediness 启发率

SYMch::GetGreediness

```
float GetGreediness()
```

获取启发率。

2.5、SYBlob Classes and Class Members

List of Function 函数列表说明

联通区域的几何特征分析，广泛应用于缺陷检验，物体定位、辨识等。

Function Name(函数名称)	说明
SYBlob::BuildBlobs	建立 Blob
SYBlob::AnalysisBlobsFeature	分析 Blob 特征
SYBlob::SortBlobs	对 Blob 特征进行排序
SYBlob::SelectBlobsInRange	对 Blob 进行特征筛选
SYBlob::GetNumOfBlob	返回 Blob 数量
SYBlob::GetFirstBlobPtr	得到第一个 Blob 的指针
SYBlob::GetNextBlobPtr	得到下一个 Blob 的指针
SYBlob::GetBlobData	得到指定 Blob 的指针
SYBlob::GetBlobPtrByCoordinates	通过坐标获得 Blob 的指针
SYBlob::DrawBlob	画出 pIt 指向的 Blob
SYBlob::DrawBlobs	画出所有的 Blob
SYBlob::DrawFeature	画出 pIt 指向 Blob 特征
SYBlob::RemoveBlob	删除指定 Blob
SYBlob::GetAbsoluteThreshold	获取绝对值模式下的分割阈值
SYBlob::SetAbsoluteThreshold	设置绝对值模式下的分割阈值
SYBlob::GetBlobCopyInFullImage	获取指定斑块的拷贝拓片，背景图为源图像大小
SYBlob::GetBlobCopy	获取指定斑块的拷贝拓片，背景图大小可根据参数设置
SYBlob::GetThresholdMode	获取当前的图像分割模式
SYBlob::SetThresholdMode	设置图像分割模式
SYBlob::GetContour	获取指定斑块的外包络点序列



SYBlob::BuildBlobs

```
void BuildBlobs(SYGrayImage &Src, int Class)
```

将相连的像素组成相连的区域称之为孔洞 (blob) 或者斑点， 创建相应的 blob 并给每一块相连区域 (blob) 分配唯一的标记。

参数:

Src	源图像
-----	-----

Class 黑色(Class =0)或者白色(Class = 1)的 blob

SYBlob::AnalysisBlobsFeature

```
void AnalysisBlobsFeature(unsigned short Feature)
```

分析斑点特征.

参数:

Feature 斑点特征

Feature 可为以下中的特征值。

```
enum OBJ_FEATURE
```

```
{
    OBJ_AREA = 1,
    OBJ_GRAVITY_X = 2,
    OBJ_GRAVITY_Y = 4,
    OBJ_LIMIT_BOX = 8,
    OBJ_FERET_BOX = 16,
    OBJ_ELLIPSE_BOX = 32
};
```

SYBlob::SortBlobs

```
void SortBlobs(SORT OPTIONS operation)
```

对 Blob 特征进行排序

参数:

operation 通过相应的特征对 blobs 进行相应的排序

```
enum SORT_OPTIONS
```

```
{
    OBJ_AREA_DESCENDING,
    OBJ_AREA_ASCENDING,
    OBJ_GRAVITY_X_DESCENDING,
    OBJ_GRAVITY_X_ASCENDING,
    OBJ_GRAVITY_Y_DESCENDING,
    OBJ_GRAVITY_Y_ASCENDING
};
```

SYBlob::SelectBlobsInRange

```
void SelectBlobsInRange(OBJ_FEATURE m_Feature, int minVal, int maxVal)
```

通过特征进行 Blob 筛选。

参数：

minVal	最小参数
maxVal	最大参数
m_Feature	特征值

```
enum OBJ_FEATURE  
{  
    OBJ_AREA = 1,  
    OBJ_GRAVITY_X = 2,  
    OBJ_GRAVITY_Y = 4,  
    OBJ_LIMIT_BOX = 8,  
    OBJ_FERET_BOX = 16,  
    OBJ_ELLIPSE_BOX = 32  
};
```

SYBlob::GetNumOfBlob

```
int GetNumOfBlob()
```

返回值：得到 Blob 的数量。

SYBlob::GetFirstBlobPtr

```
ListItem* GetFirstBlobPtr(void)
```

返回：得到第一个 Blob 的指针。

SYBlob::GetNextBlobPtr

```
ListItem* GetNextBlobPtr(ListItem *pItem)
```

返回：得到下一个 Blob 的指针，和 GetFirstBlobPtr 配合使用。

SYBlob::GetBlobData

`ListItem* GetBlobData(unsigned int Idx)`

返回：得到指定 Blob 的指针。

参数：

Idx Blob 的索引号

SYBlob::GetBlobPtrByCoordinates

`ListItem* GetBlobPtrByCoordinates(int x, int y, float ZoomX, float ZoomY, int PanX, int PanY)`

通过坐标获得 Blob 的指针。

参数：

x	水平坐标值
y	垂直坐标值
ZoomX	水平方向比例
ZoomY	垂直方向比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

返回：得到坐标所在 blob 的地址，可通过此地址获得相应 blob 的特征。

SYBlob::DrawBlob

`void DrawBlob(HDC dc, ListItem *pIt, COLORREF clrf, float ZoomX, float ZoomY, int PanX, int PanY)`

画出 pIt 指向的 Blob

参数：

dc	处理目标窗口的设备上下文
pIt	blob 所在的地址
clrf	斑点的颜色
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYBlob::DrawBlobs

```
void DrawBlobs(HDC dc, COLORREF clr, float ZoomX, float ZoomY, int PanX, int PanY)
```

画出所有的 Blob

参数:

dc	处理目标窗口的设备上下文
clr	斑点的颜色
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYBlob::DrawFeature

```
void DrawFeature(HDC dc, ListItem* pIt, int Feature, float ZoomX = 1.0f, float ZoomY = 1.0f,  
int PanX = 0, int PanY = 0)
```

画出 pIt 指向 Blob 的特征, **Feature** 是一个组合值组, 合方式 F1|F2|F3。

参数:

dc	处理目标窗口的设备上下文
pIt	blob 所在的地址
Feature	特征值

```
enum OBJ_FEATURE
```

```
{
```

```
    OBJ_AREA = 1,  
    OBJ_GRAVITY_X = 2,  
    OBJ_GRAVITY_Y = 4,  
    OBJ_LIMIT_BOX = 8,  
    OBJ_FERET_BOX = 16,  
    OBJ_ELLIPSE_BOX = 32
```

```
};
```

ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向偏移
PanY	垂直方向偏移

SYBlob:: RemoveBlob

```
void RemoveBlob(int Idx);
```

返回：删除指定 Blob

参数：

Idx Blob 的索引号

SYBlob::GetAbsoluteThreshold

```
unsigned int GetAbsoluteThreshold( );
```

获取绝对值分割模式下，正在使用的分割阈值。

SYBlob::SetAbsoluteThreshold

```
void SetAbsoluteThreshold(unsigned int threshold);
```

设置绝对值分割模式下使用的分割阈值。

参数：

threshold 分割阈值

SYBlob::GetBlobCopyInFullImage

```
void GetBlobCopyInFullImage(ListItem *pItem, SYGrayImage *blobCopy);
```

将指定的斑块拷贝绘画到拓片图像中。拓片图像的大小与源图像大小一致。

参数：

pItem Blob 的索引指针

blobCopy 拓片图像

SYBlob::GetBlobCopy

```
void GetBlobCopy(ListItem *pItem, float s, SYGrayImage *blobCopy);
```

将指定的斑块拷贝绘画到拓片图像中。拓片图像的大小可通过比例系数 s 进行控制。

参数：

pItem	Blob 的索引指针
s	比例系数，以斑块的最小外接正矩形为比例 1。生成该正矩形 s 倍大小的拓片图像。
blobCopy	拓片图像

SYBlob::GetThresholdMode

```
THRESHOLD_MODE GetThresholdMode();
```

获取当前使用的分割模式。

分割模式枚举如下：

```
enum THRESHOLD_MODE  
{  
    MIN_RESIDUE,  
    MAX_ENTROPY,  
    ABSOLUTETHRESHOLD  
};
```

SYBlob::SetThresholdMode

```
void SetThresholdMode(THRESHOLD_MODE mode);
```

设置图像分割模式

参数：

mode	分割模式枚举类型
------	----------

SYBlob::GetContour

```
void GetContour(ListItem *pItem, vector<SYMV::Point2i> &ctr);
```

获取指定斑块的外包络点序列。

参数：

pItem	Blob 的索引指针
ctr	外包络点序列

2.6、SYGauge Classes and Class Members

2.6.1 SYCalibration

List of Function 函数列表说明

Function Name(函数名称)	说明
<code>SYCalibration:: CalibrationDotGrid</code>	圆点标定板标定
<code>SYCalibration:: CalibrationChessBoard</code>	使棋盘格标定板标定
<code>SYCalibration:: CalibrationLandMark</code>	通过视觉与物理点对进行标定
<code>SYCalibration:: Load</code>	载入标定参数
<code>SYCalibration:: Save</code>	保存标定系数
<code>SYCalibration:: SensorToWorld</code>	像素坐标转换成物理坐标
<code>SYCalibration:: WorldToSensor</code>	物理坐标转换成像素坐标
<code>SYCalibration:: PerspectiveTrain</code>	透视变换训练
<code>SYCalibration:: PerspectiveTransform</code>	透视变换转换，点转换或图像转换
<code>SYCalibration:: UndistortImage</code>	图像纠正畸变

SYCalibration::CalibrationDotGrid

```
double CalibrationDotGrid(SYGrayImage &MarkGray, SYMV::Point2f LeftUp, SYMV::Point2f  
RightUp, SYMV::Point2f LeftDown, SYMV::Point2f RightDown, UINT cols, UINT rows,  
SYMV::Point2f &MarkPoint)
```

圆点标定板标定

参数:

MarkGray	圆点标定板图像
LeftUp	左上角 Mark 点物理坐标
RightUp	右上角 Mark 点物理坐标
LeftDown	左下角 Mark 点物理坐标
RightDown	右下角 Mark 点物理坐标
cols	圆点列数
rows	圆点行数
MarkPoint	圆点阵列的像素坐标数组

SYCalibration::CalibrationChessBoard

```
double CalibrationChessBoard(SYGrayImage &MarkGray, SYMV::Point2f LeftUp, SYMV::Point2f  
RightUp, SYMV::Point2f LeftDown, SYMV::Point2f RightDown, UINT cols, UINT rows,  
SYMV::Point2f &MarkPoint)
```

棋盘格标定板标定

参数:

MarkGray	棋盘格图像
LeftUp	左上角 Mark 点物理坐标
RightUp	右上角 Mark 点物理坐标
LeftDown	左下角 Mark 点物理坐标
RightDown	右下角 Mark 点物理坐标
cols	棋盘格点列数
rows	棋盘格点行数
MarkPoint	棋盘格点阵列的像素坐标数组

SYCalibration::CalibrationLandMark

```
double CalibrationLandMark(SYMV::Point2f *VP, SYMV::Point2f *MP, int ImgWidth, int  
ImgHeight, int TotalPointNum)
```

通过视觉与物理点对进行标定

参数：

*VP	像素坐标数组
*MP	物理坐标数组
ImgWidth	图像宽度
ImgHeight	图像高度
TotalPointNum	点对数量

SYCalibration::Save

```
void Save(const char* FilePath)
```

将标定数据保存到指定文件。

参数：

FilePath	标定数据文件
----------	--------

SYCalibration::Load

```
int Load(const char* FilePath)
```

从指定文件中导入标定数据。

参数：

FilePath	标定数据文件
----------	--------

SYCalibration::SensorToWorld

```
cv::Point2f SensorToWorld(SYMV::Point2f *mSensor)
```

像素坐标转换成物理坐标

参数

mSensor	像素坐标
---------	------

返回值为对应像素坐标的物理坐标

SYCalibration::WorldToSensor

cv::Point2f WorldToSensor(*SYMV::Point2f *mWorld*)

物理坐标转换成像素坐标

参数

mWorld 物理坐标

返回值为对应物理坐标的像素坐标

SYCalibration::PerspectiveTrain

void PerspectiveTrain(*SYMV::Point2f src[4], SYMV::Point2f dst[4]*)

透视变换训练，提供 4 个点对完成训练。

参数

src[4] 源坐标

dst[4] 转换坐标

SYCalibration::PerspectiveTransform

SYMV::Point2f PerspectiveTransform(*SYMV::Point2f srcPoint*)

将源坐标进行透视转换。

参数

srcPoint 源坐标

void PerspectiveTransform(*SYGrayImage *src, SYGrayImage *dst*)

将源图像进行透视转换。

参数

src 源图像

dst 目标图像

SYCalibration::UndistortImage

```
void UndistortImage(int ImgWidth, int ImgHeight, SYGrayImage *src, SYGrayImage *dst)
```

根据棋盘格、圆点标定板或自由点对标定信息，对图像进行畸变纠正。

参数

ImgWidth	图像宽度
ImgHeight	图像高度
src	源图像
dst	目标图像

2.6.2 SYPointGauge

List of Function 函数列表说明

Function Name (函数名称)	说明
<code>SYPointGauge::SetPlacement</code>	设定 PointGauge 的位置及大小
<code>SYPointGauge::SetModelCenter</code>	设定 PointGauge 的中心点位置
<code>SYPointGauge::SetModelAngle</code>	设定 PointGauge 的角度
<code>SYPointGauge::SetModelTolerance</code>	设定 PointGauge 的长度
<code>SYPointGauge::GetPlacement</code>	返回 PointGauge 的位置及大小
<code>SYPointGauge::GetModelCenter</code>	返回 PointGauge 的中心点位置
<code>SYPointGauge::GetModelCenterX</code>	返回 PointGauge 的中心点 X 位置
<code>SYPointGauge::GetModelCenterY</code>	返回 PointGauge 的中心点 Y 位置
<code>SYPointGauge::GetModelAngle</code>	返回 PointGauge 的角度
<code>SYPointGauge::GetModelTolerance</code>	返回 PointGauge 的长度
<code>SYPointGauge::GetHitShape</code>	修改落在 PointGague 控制点上的光标形状
<code>SYPointGauge::Drag</code>	鼠标操作拖动改变 PointGauge
<code>SYPointGauge::Draw</code>	绘画 PointGauge
<code>SYPointGauge::SetTransitionType</code>	设置 PointGauge 的测量极性
<code>SYPointGauge::GetTransitionType</code>	返回测量的极性
<code>SYPointGauge::SetTransitionChoice</code>	设置选取边缘点的模式
<code>SYPointGauge::SetThreshold</code>	设定 PointGauge 阈值
<code>SYPointGauge::GetThreshold</code>	获取 PointGauge 阈值
<code>SYPointGauge::GetTransitionChoice</code>	获取边缘点的模式
<code>SYPointGauge::SetMinArea</code>	设置 MinArea 值
<code>SYPointGauge::GetMinArea</code>	获取 MiniArea 值
<code>SYPointGauge::SetThreshold</code>	设置阈值
<code>SYPointGauge::GetThreshold</code>	获取阈值
<code>SYPointGauge::SetSmoothing</code>	径向平滑系数，平滑路径上噪点
<code>SYPointGauge::GetSmoothing</code>	返回径向平滑系数
<code>SYPointGauge::Measure</code>	测量边缘点
<code>SYPointGauge::GetMeasuredPoint</code>	返回边缘点特征（坐标、梯度等）
<code>SYPointGauge::GetNumOfMeasuredPoints</code>	返回测量到的边缘点个数
<code>SYPointGauge::Attach</code>	将 PointGauge 与 SYCalibration 标定坐标系建立关联

<code>SYPointGauge::Detach</code>	取消 PointGaug 与 SYCalibration 标定坐标系关联
<code>SYPointGauge::SetThickness</code>	设置切向平滑系数
<code>SYPointGauge::GetThickness</code>	获取切向平滑系数
<code>SYPointGauge::SetTransitionIdx</code>	设置返回指定序号的测量点
<code>SYPointGauge::GetTransitionIdx</code>	获取指定返回测量点的序号
<code>SYPointGauge::SetDerSize</code>	设置边缘强度系数
<code>SYPointGauge::GetDerSize</code>	获取边缘强度系数

SYPointGauge::SetPlacement

```
void SetPlacement(int CentreX, int CenterY, float Angle, int Tolerance)
```

设定 PointGauge 的位置及大小

参数：

CenterX	X 中心坐标
CenterY	Y 中心坐标
Angle	角度
Tolerance	长度

SYPointGauge::SetModelCenter

```
void SetModelCenter(int CenterX, int CenterY)
```

设定 PointGauge 的中心点位置

参数：

CenterX	X 中心坐标
CenterY	Y 中心坐标

SYPointGauge::SetModelAngle

```
void SetModelAngle (float Angle)
```

设定 PointGauge 的角度

参数：

Angle	角度
-------	----

SYPointGauge::SetModelTolerance

```
void SetModelTolerance (int Tolerance)
```

设定 PointGauge 的长度

参数：

Tolerance	长度
-----------	----

SYPointGauge::GetPlacement

PGData GetPlacement()

返回 PointGauge 的位置及大小

```
typedef struct PGData
{
    int    CenterX;
    int    CenterY;
    int    Tolerance; //half length
    float  Rad;       //radian

    int    GetX() { return CenterX; }
    int    GetY() { return CenterY; }
    int    GetTolerance() { return Tolerance; }
    float  GetAngle() { return Rad / CV_PI*180.0f; }
}PGData;
```

SYMV::GetModelCenter

SYMV::Point2i GetModelCenter()

返回 PointGauge 的中心点位置

SYPointGauge::GetModelCenterX

int GetModelCenterX()

返回 PointGauge 的中心点 X 位置

SYPointGauge::GetModelCenterY

int GetModelCenterY()

返回 PointGauge 的中心点 Y 位置

SYPointGauge::GetModelAngle

```
void GetModelAngle ()
```

返回 PointGauge 的角度

SYPointGauge::GetModelTolerance

```
void GetModelTolerance ()
```

返回 PointGauge 的长度

SYPointGauge::GetHitShape

```
Point_Handle GetHitShape(int CursorX, int CursorY, float ZoomX, float ZoomY, int PanX,  
int PanY)
```

修改落在 PointGauge 控制点上的光标形状，配合 SetCursor 和 Drag 函数来使用。

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYPointGauge::Drag

```
void Drag(int CursorX, int CursorY, float ZoomX, float ZoomY, int PanX, int PanY)
```

鼠标操作拖动改变 PointGauge

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYPointGauge::Draw

`void Draw(HDC h_Dc, Drawing_Modes DrawMode, float ZoomX, float ZoomY, int PanX, int PanY)`

绘画 PointGauge。

参数：

h_Dc 处理目标窗口的设备上下文
 DrawMode 绘制的模式，见 Drawing_Modes 表格所示
 ZoomX 水平方向的缩放比例
 ZoomY 垂直方向的缩放比例
 PanX 水平方向的偏移量
 PanY 垂直方向的偏移量

Drawing_Modes	
DRAW_NOMINAL	常规测量工具
DRAW_ACTUAL	实际测量工具
DRAW_SAMPLE_POINTS	实际采样点

`enum Drawing_Modes`

```
{
    DRAW_NOMINAL,      //draw the gauge
    DRAW_ACTUAL,        //draw the result measured by gauge
    DRAW_SAMPLE_POINTS //draw all sample points
};
```

SYPointGauge::SetTransitionType

`void SetTransitionType(TransitionType mode)`

设置 PointGauge 的测量极性，从黑到白、从白到黑、白到黑或者黑到白等

参数：

mode 测量方向

Mode	描述
SYG_BW	从黑到黑
SYG_WB	从白到黑
SYG_BW_OR_WB	白到黑或者黑到白

`enum TransitionType`

```
{
    SYG_BW_OR_WB,
    SYG_BW,
    SYG_WB
};
```

SYPointGauge::GetTransitionType

`TransitionType` `GetTransitionType(void)`

获取 PointGauge 的测量极性，从黑到白、从白到黑、白到黑或者黑到白等

SYPointGauge::SetTransitionChoice

`void` `SetTransitionChoice(TransitionChoice mode)`

设置 PointGauge 的选取边缘点模式

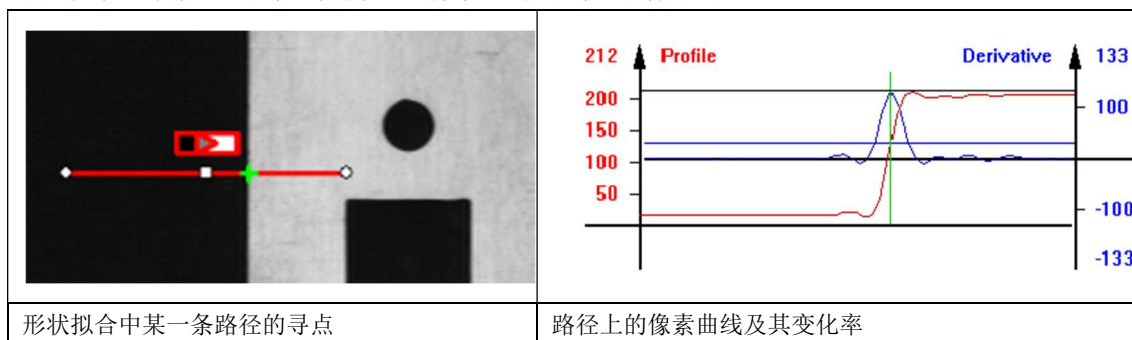
参数：

mode 边缘点模式

Mode	描述
SYG_FROM_BEGIN	指从设定寻点方向开始的第一个过渡点作为边缘点
SYG_FROM_END	指从设定寻点方向开始的最后一个过渡点作为边缘点
SYG_LARGEST_AMPLITUDE	以 Amplitude 最大值的过渡点作为边缘点
SYG_LARGEST_AREA	以 Area 最大值的过渡点作为边缘点

形状的拟合，首先是通过在设定的方向上寻找出边缘点，再进行相关形状的拟合的。

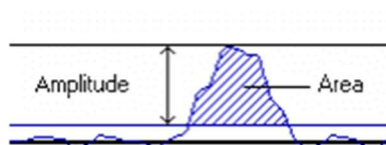
测量参数全部都是针对每一条路径上的像素曲线变化率进行操作的。



上右图中，红色曲线代表了左图中红色路径上像素值的排列；蓝色线代表了像素值曲线的变化率，数学上是像素值曲线的一阶导数。可以简单地理解，变化率曲线上的一个峰或一个谷对应的就是一个边缘上的过渡点，即是我们寻找的边缘点。

形状拟合的基础是每一条路径上的寻找边缘点操作，而寻找边缘点操作主要是对像素曲线变化率的控制。

像素变化率曲线上会有若干个概念，如下图所示：



上图中，蓝色为像素变化率曲线。底下的蓝色直线代表的是 Threshold；Threshold 与峰值（或谷值）之间的差称为 Amplitude；由 Threshold 及高于 Threshold 以上的变化率曲线围成的称为 Area。

<1> TransitionChoice：包括 From Begin、From End、Largest Amplitude、Largest Area，可能在一 条路径上会找到若干个过渡点，通过该参数控制选取哪一个作为边缘点。

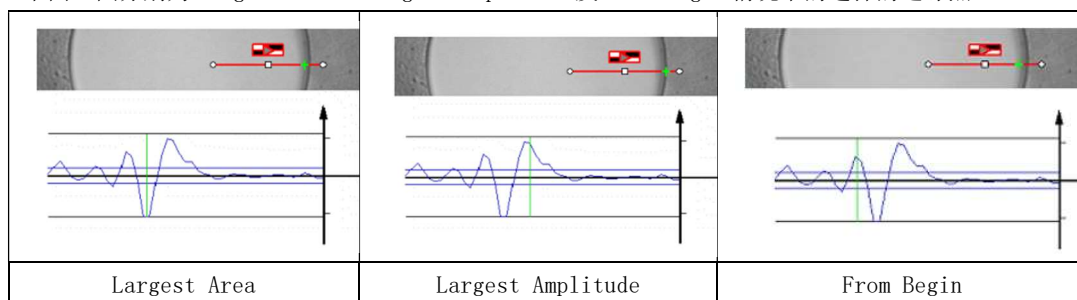
From Begin：指从设定寻点方向开始的第一个过渡点作为边缘点

From End：指从设定寻点方向开始的最后一个过渡点作为边缘点

Largest Amplitude：以 Amplitude 最大值的过渡点作为边缘点

Largest Area：以 Area 最大值的过渡点作为边缘点

下面 3 图分别为 Largest Area、Largest Amplitude 及 From Begin 情况下的选择的边缘点。



SYPointGauge::GetTransitionChoice

TransitionChoice GetTransitionChoice()

获取 PointGauge 的边缘点选择模式

SYPointGauge::SetThreshold

void SetThreshold(int threshold)

设定 PointGauge 阈值。

参数：

threshold 阈值范围在 0~255 之间

SYPointGauge::GetThreshold

int GetThreshold()

获取 PointGauge 阈值。

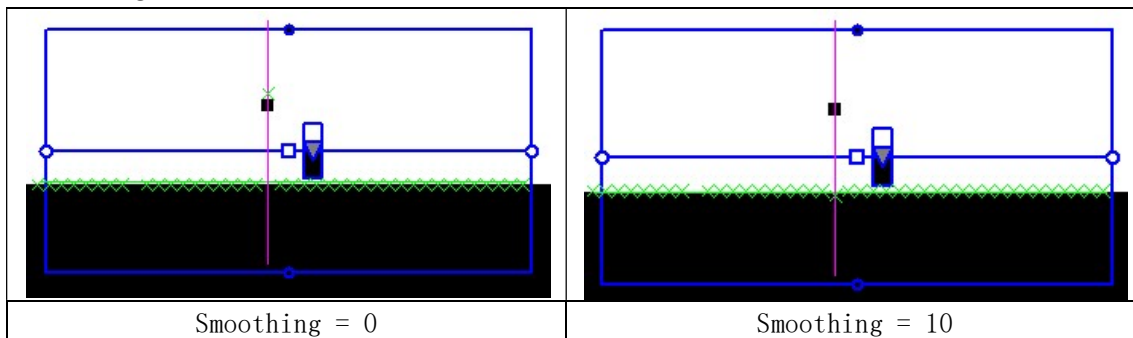
SYPointGauge::SetSmoothing

`void SetSmoothing(int smoothing)`

径向平滑系数，平滑路径上的噪点。

参数：

smoothing 径向平滑系数，可过滤路径上像素噪点，滤波参数范围在 0~25 之间



拟合路径上有一黑色噪点，调整 Smoothing 做中值滤波，噪点被过滤，最终选择了下方的拟合点。

SYPointGauge::GetSmoothing

`int GetSmoothing()`

获取径向平滑系数。

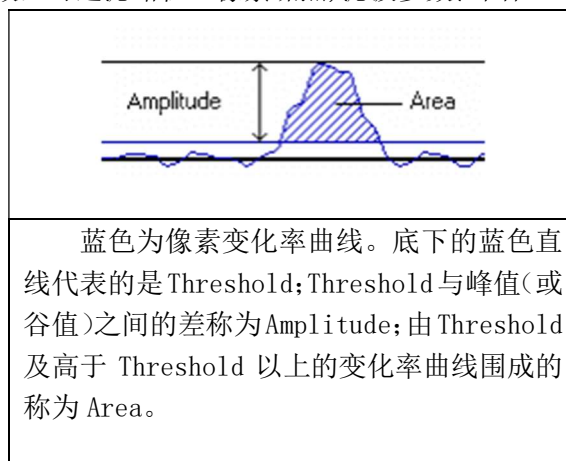
SYPointGauge::SetMinArea

`void SetMinArea(int minArea)`

设置 MinArea 值

参数：

minArea 径向平滑系数，可过滤路径上像素噪点，滤波参数范围在 0~25 之间



SYPointGauge::GetMinArea

```
int GetMinArea()
```

获取 MinArea 值。

SYPointGauge::Attach

```
void Attach(SYCalibration *syWorld)
```

使用 Attach 将 PointGauge 直线量测工具与 SYCalibration 标定坐标系建立关联

参数：

SYCalibration

标定数据

SYPointGauge::Detach

```
void Detach()
```

使用 Detach 将 PointGauge 直线量测工具与 SYCalibration 标定坐标系脱离关联

SYPointGauge::InImageRegion

```
Bool InImageRegion(SYMV::Point2i *pStr, SYMV::Point2i *pEnd, SYMV::Point2i *NewSPoint,  
SYMV::Point2i *NewEPoint, int Width, int Height)
```

使用 Detach 将 PointGauge 直线量测工具与 SYCalibration 标定坐标系脱离关联

参数：

Img

源图像

SYPointGauge::Measure

```
void Measure(SYGrayImage &Img)
```

进行 PointGauge 测量。

参数：

Img

源图像

SYPointGauge::GetMeasuredPoint

`PointInfo GetMeasuredPoint(int idx)`

根据索引号，得到测量点信息

参数：

idx 测量到的点的索引号

返回：得到测量到点的坐标值。

```
typedef struct PointInfo
{
    float      x;
    float      y;
    int        TransType;   //0: BW  1:WB
    int        peak;
    int        area;
    bool       bSel;
}PointInfo;
```

SYPointGauge::GetNumMeasuredPoints

`int GetNumOfMeasuredPoints(void)`

得到所有测量点的个数

返回：得到测量点的数量。

SYPointGauge::SetThickness

`void SetThickness(int Thickness)`

设置切向平滑系数，即求平均对比度的像素点个数。当遇到强噪声点时，通过切向平均拉低噪声点的对比度，达到剔除该强噪声点的目的。

参数：

Thickness 切向平滑系数

SYPointGauge::GetThickness

`int GetThickness()`

获取切向平滑系数。

SYPointGauge::SetTransitionIndex

```
void SetTransitionIndex(int idx)
```

设置返回指定序号的测量点信息。当 TransitionType 选择为 SYG_FROM_BEGIN 或 SYG_FROM_END 时，可通过该函数指定要返回第 idx 个测量点信息。

SYPointGauge::GetTransitionIndex

```
int GetTransitionIndex( )
```

获取被指定返回的测量点序号。

SYPointGauge::SetDerSize

```
void SetDerSize(int DSize)
```

设置边缘强度系数。弱边缘的对比度较低，不利于进行稳定的边缘点分析；通过设置边缘强度，可强化边缘点的响应强度，使其对比度拉高。

参数：

DSize 边缘强度系数

SYPointGauge::GetDerSize

```
int GetDerSize( )
```

获取边缘强度系数

2.6.3 SYLineGauge

List of Function 函数列表说明

Function Name(函数名称)	说明
SYLineGauge::SetPlacement	设定 LineGauge 的位置及大小
SYLineGauge::SetModelCenter	设定 LineGauge 的中心点位置
SYLineGauge::SetModelLength	设定 LineGauge 的长度
SYLineGauge::SetModelTolerance	设定 LineGauge 的高度
SYLineGauge::SetModelAngle	设定 LineGauge 的角度
SYLineGauge::GetPlacement	返回 LineGauge 的位置及大小
SYLineGauge::GetModelCenter	返回 LineGauge 的中心点位置
SYLineGauge::GetModelCenterX	返回 LineGauge 的中心点 X 位置
SYLineGauge::GetModelCenterY	返回 LineGauge 的中心点 Y 位置
SYLineGauge::GetModelLength	返回 LineGauge 的长度
SYLineGauge::GetModelTolerance	返回 LineGauge 的高度
SYLineGauge::GetModelAngle	返回 LineGauge 的角度
SYLineGauge::GetHitShape	修改 LineGauge 控制点上的光标形状
SYLineGauge::Drag	鼠标操作拖动改变 LineGauge
SYLineGauge::SetTransitionType	设置 LineGauge 的测量极性
SYLineGauge::GetTransitionType	得到测量的极性
SYLineGauge::SetThreshold	设置阈值
SYLineGauge::GetThreshold	获取阈值
SYLineGauge::SetSmoothing	径向平滑系数，平滑路径上像素噪点
SYLineGauge::GetSmoothing	得到径向平滑系数
SYLineGauge::SetSamplingStep	设置 LineGauge 的采样间隔
SYLineGauge::GetSamplingStep	获取 LineGauge 的采样间隔
SYLineGauge:: SetFilterOutlierNumPass	设置差异点剔除滤波次数
SYLineGauge:: GetFilterOutlierNumPass	获取差异点剔除滤波次数
SYLineGauge:: SetFilterOutlierThreshold	设置差异点剔除滤波阈值
SYLineGauge:: GetFilterOutlierThreshold	获取差异点剔除滤波阈值
SYLineGauge::Attach	将 LineGauge 与 SYCalibration 标定坐标系建立关联
SYLineGauge::Detach	取消 LineGauge 与 SYCalibration 标定坐标系关联

<code>SYLineGauge::Measure</code>	进行直线拟合
<code>SYLineGauge::GetMeasuredLine</code>	得到直线测量的结果，包含中心和角度等
<code>SYLineGauge::GetNumSamples</code>	得到 LineGauge 拟合点总数
<code>SYLineGauge::GetNumValidSamples</code>	得到 LineGauge 拟合的有效点个数
<code>SYLineGauge::Draw</code>	绘画 LineGauge
<code>SYLineGauge::SetTransitionChoice</code>	设置选择边缘过渡点的模式
<code>SYLineGauge::GetTransitionChoice</code>	获取当前边缘过渡点的模式
<code>SYLineGauge::SetMinArea</code>	设置边缘过渡区域的最小面积
<code>SYLineGauge::GetMinArea</code>	获取边缘过渡区域的最小面积
<code>SYLineGauge::SetThickness</code>	设置切向平滑系数
<code>SYLineGauge::GetThickness</code>	获取切向平滑系数
<code>SYLineGauge::SetTransitionIndex</code>	指定返回边缘点的序号
<code>SYLineGauge::GetTransitionIndex</code>	获取当前返回的边缘点序号
<code>SYLineGauge::SetDerSize</code>	设置边缘强度
<code>SYLineGauge::GetDerSize</code>	获取边缘强度
<code>SYLineGauge::EnableClusterAnalysis</code>	使能聚类分析功能
<code>SYLineGauge::GetClusterAnalysis</code>	获取聚类分析功能是否打开
<code>SYLineGauge::SetClusterError</code>	设置聚类分析的误差
<code>SYLineGauge::GetClusterError</code>	获取聚类分析误差

SYLineGauge::SetPlacement

`void SetPlacement(int CenterX, int CenterY, int Length, int Tolerance, float Angle)`

设定 LineGauge 的位置与大小

参数：

CenterX	X 中心坐标
CenterY	Y 中心坐标
Length	长度
Tolerance	高度的一半
Angle	角度



Y 中心坐标

长度

高度的一半

角度

SYLineGauge::GetPlacement

LGData GetPlacement()

返回 LineGauge 的位置与大小

```
typedef struct LGData
{
    int      CenterX;
    int      CenterY;
    int      Length;
    int      Tolerance; //Half Width
    float    Angle;

    int      GetX() { return CenterX; }
    int      GetY() { return CenterY; }
    int      GetLength() { return Length; }
    int      GetTolerance() { return Tolerance; }
    float    GetAngle() { return Angle; }
}LGData;
```

SYLineGauge::GetModelCenter

SYMV::Point2i GetModelCenter()

返回 LineGauge 的中心点位置

SYLineGauge::GetModelCenterX

int GetModelCenterX()

返回 LineGauge 的中心点 X 位置

SYLineGauge::GetModelCenterY

int GetModelCenterY()

返回 LineGauge 的中心点 Y 位置

SYLineGauge::GetModelLength

```
void GetModelLength()
```

返回 LineGauge 的长度

SYLineGauge::GetModelTolerance

```
void GetModelTolerance ()
```

返回 LineGauge 的高度

SYLineGauge::GetModelAngle

```
void GetModelAngle ()
```

返回 LineGauge 的角度

SYLineGauge::GetHitShape

```
Line_Handle GetHitShape(int CursorX, int CursorY)
```

修改 LineGauge 当前光标下的形状，配合 SetCursor 和 Drag 函数来使用。

参数：

CursorX 光标水平坐标值

CursorY 光标垂直坐标值

SYLineGauge::Drag

```
void Drag(int CursorX, int CursorY, float ZoomX, float ZoomY)
```

鼠标操作 LineGauge 拖动改变位置及大小

参数：

CursorX 光标水平坐标值

CursorY 光标垂直坐标值

ZoomX 水平方向的缩放比例

ZoomY 垂直方向的缩放比例

SYLineGauge::Draw

`void Draw(HDC h_Dc, Drawing_Modes DrawMode, float ZoomX, float ZoomY, int PanX, int PanY)`

绘画 LineGauge。

参数：

h_Dc	处理目标窗口的设备上下文
DrawMode	绘制的模式，见 Drawing_Modes 表格所示
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

Drawing_Modes	
DRAW_NOMINAL	常规测量工具
DRAW_ACTUAL	实际测量工具
DRAW_SAMPLE_POINTS	实际采样点

`enum Drawing_Modes`

```
{
    DRAW_NOMINAL,      //draw the gauge
    DRAW_ACTUAL,       //draw the result measured by gauge
    DRAW_SAMPLE_POINTS //draw all sample points
};
```

SYLineGauge::SetTransitionType

`void SetTransitionType(TransitionType mode)`

设置 LineGauge 的测量极性，从黑到白、从白到黑、白到黑或者黑到白等

参数：

mode	测量方向
------	------

Mode	描述
SYG_BW	从黑到黑
SYG_WB	从白到黑
SYG_BW_OR_WB	白到黑或者黑到白

`enum TransitionType`

```
{
    SYG_BW_OR_WB,
    SYG_BW,
    SYG_WB
};
```



返回值：得到测量的极性。

threshold	閾值范围在 0~255 之间
-----------	----------------

获取 LineGauge 阈值。

SamplingStep	最小采样间隔为 1 个像素
--------------	---------------

获取 LineGauge 的采样间隔。

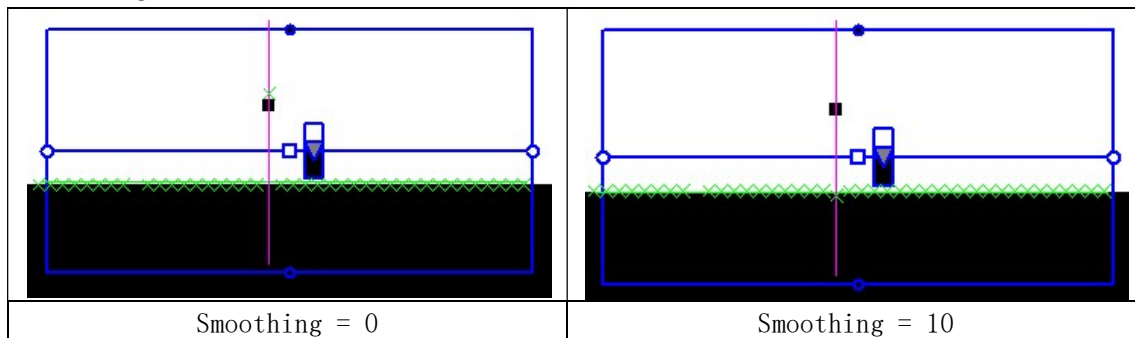
SYLineGauge::SetSmoothing

`void SetSmoothing(int smoothing)`

设置径向平滑系数，平滑路径上的噪点。

参数：

`smoothing` 径向平滑系数，可过滤路径上像素噪点，滤波参数范围在 0~25 之间



拟合路径上有一黑色噪点，调整 Smoothing 做中值滤波，噪点被过滤，最终选择了下方的拟合点。

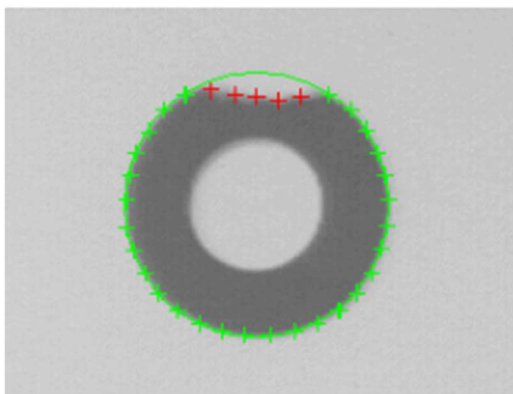
SYLineGauge::GetSmoothing

`int GetSmoothing()`

获取径向平滑系数。

【差异点剔除】

差异点是远离有效点集合的点，它的出现会影响形状（直线、圆、矩形）的逼近程度，需要将其剔除。下图所示为剔除圆上的差异点：



SYLineGauge::SetFilterOutlierThreshold

`void SetFilterOutlierThreshold(float mThreshold)`

设定剔除差异点滤波阈值，该值的倒数即为所有边缘点到拟合出的直线的平均距离。
到拟合直线距离大于该平均距离的点即为差异点。

参数：

mThreshold 滤波阈值

SYLineGauge::GetFilterOutlierThreshold

`float GetFilterOutlierThreshold()`

获取剔除差异点滤波阈值。

SYLineGauge::SetFilterOutlierNumPass

`void SetFilterOutlierNumPass(int mNumPass)`

设定剔除差异点滤波次数，在拟合过程中检测差异点的次数。

参数：

mNumPass 滤波次数

SYLineGauge::GetFilterOutlierThreshold

`int GetFilterOutlierNumPass()`

获取检测差异点滤波次数。

SYLineGauge::GetNumSamples

`int GetNumSamples()`

获取拟合直线测量上所有采样点总数

SYLineGauge::GetNumValidSamples

```
int GetNumValidSamples()
```

获取拟合直线测量上有效采样点的个数

SYLineGauge::Measure

```
void Measure(SYGrayImage &Img)
```

进行直线拟合测量。

参数：

Img 源图像

SYLineGauge::GetMeasuredLine

```
LineResult GetMeasuredLine()
```

获得 LineGauge 测量结果

```
typedef struct LineResult
```

```
{  
    float                  Angle; //the Angle in [-90 90]  
    float                  x;     //one point X in the fitting-line  
    float                  y;     //one point Y in the fitting-line  
}LineResult;
```

SYLineGauge::Attach

```
void Attach(SYCalibration *syWorld)
```

使用 Attach 将 LineGauge 直线量测工具与 SYCalibration 标定坐标系建立关联

参数：

SYCalibration 标定数据

SYLineGauge::Detach

```
void Detach()
```

使用 Detach 将 LineGauge 直线量测工具与 SYCalibration 标定坐标系取消关联

SYLineGauge::SetTransitionChoice

```
void SetTransitionChoice(TransitionChoice mode)
```

设置选择边缘点的模式。详细可参见 SYPointGauge::SetTransitionChoice 函数。

参数：

mode 边缘点模式枚举类型

```
enum TransitionChoice
```

```
{  
    SYG_FROM_BEGIN,  
    SYG_FROM_END,  
    SYG_LARGEST_AMPLITUDE,  
    SYG_LARGEST_AREA,  
    SYG_ALL  
};
```

SYLineGauge::GetTransitionChoice

```
TransitionChoice GetTransitionChoice()
```

获取选择边缘点的模式。详细可参见 SYPointGauge::SetTransitionChoice 函数。

SYLineGauge::SetMinArea

```
void SetMinArea(int minArea)
```

设置边缘点过渡区域的最小面积。详细可参见 SYPointGauge::SetMinArea 函数。

SYLineGauge::GetMinArea

```
int GetMinArea()
```

获取边缘点过渡区域最小面积。

SYLineGauge::SetThickness

`void SetThickness(int Thickness)`

设置切向平滑系数，即求平均对比度的像素点个数。当遇到强噪声点时，通过切向平均拉低噪声点的对比度，达到剔除该强噪声点的目的。

参数：

Thickness 切向平滑系数

SYLineGauge::GetThickness

`int GetThickness()`

获取切向平滑系数。

SYLineGauge::SetTransitionIndex

`void SetTransitionIndex(int idx)`

设置返回指定序号的测量点信息。当 TransitionType 选择为 SYG_FROM_BEGIN 或 SYG_FROM_END 时，可通过该函数指定要返回第 idx 个测量点信息。

SYLineGauge::GetTransitionIndex

`int GetTransitionIndex()`

获取被指定返回的测量点序号。

SYLineGauge::SetDerSize

`void SetDerSize(int DSize)`

设置边缘强度系数。弱边缘的对比度较低，不利于进行稳定的边缘点分析；通过设置边缘强度，可强化边缘点的响应强度，使其对比度拉高。

参数：

DSize 边缘强度系数

SYLineGauge::GetDerSize

```
int GetDerSize( )
```

获取边缘强度系数。

SYLineGauge::EnableClusterAnalysis

```
void EnableClusterAnalysis(bool bEnable)
```

使能聚类分析功能。聚类分析，是一种强效的边缘干扰点剔除方法。以多者为优的原则，即共线点最多的点集为最优直线拟合点，其它被认为是干扰点。

SYLineGauge::GetClusterAnalysis

```
bool GetClusterAnalysis( )
```

获取聚类分析功能是否打开。

SYLineGauge::SetClusterError

```
void SetClusterError(float err)
```

设置聚类分析误差。

SYLineGauge::GetClusterError

```
float GetClusterError( )
```

获取聚类分析误差。

2.6.4 SYCircleGauge

List of Function 函数列表说明

Function Name(函数名称)	说明
<code>SYCircleGauge::SetPlacement</code>	设定 CircleGauge 的位置及形状大小
<code>SYCircleGauge::GetPlacement</code>	返回 CircleGauge 的位置及形状大小
<code>SYCircleGauge::GetHitShape</code>	修改 CircleGauge 当前光标下的形状
<code>SYCircleGauge::Drag</code>	鼠标操作拖动改变 CircleGauge 位置及形状大小
<code>SYCircleGauge:: SetMeasureDirection</code>	设置 CircleGauge 的测量方向，从里往外、从外往里
<code>SYCircleGauge:: GetMeasureDirection</code>	得到 CircleGauge 的测量方向
<code>SYCircleGauge::SetTransitionType</code>	设置 CircleGauge 的测量极性，从黑到白、从白到黑等来去测量
<code>SYCircleGauge::GetTransitionType</code>	得到测量的极性
<code>SYCircleGauge::SetThreshold</code>	设置阈值
<code>SYCircleGauge::GetThreshold</code>	得到阈值
<code>SYCircleGauge::SetSmoothing</code>	径向平滑系数，平滑路径上像素噪点
<code>SYCircleGauge::GetSmoothing</code>	得到径向平滑系数
<code>SYCircleGauge::SetSamplingStep</code>	设置 CircleGauge 的采样间隔
<code>SYCircleGauge::GetSamplingStep</code>	获取 CircleGauge 的采样间隔
<code>SYCircleGauge::SetFilterOutlierNumPass</code>	设置差异点剔除滤波次数
<code>SYCircleGauge::GetFilterOutlierNumPass</code>	获取差异点剔除滤波次数
<code>SYCircleGauge::SetFilterOutlierThreshold</code>	设置差异点剔除滤波阈值
<code>SYCircleGauge::GetFilterOutlierThreshold</code>	获取差异点剔除滤波阈值
<code>SYCircleGauge::Attach</code>	将 CircleGauge 工具与 SYCalibration 标定坐标系建立关联
<code>SYCircleGauge::Detach</code>	将 CircleGauge 工具与 SYCalibration 标定坐标系取消关联
<code>SYCircleGauge::Measure</code>	进行圆拟合测量
<code>SYCircleGauge::GetMeasuredCircle</code>	得到圆测量的结果，包含中心和半径等

SYCircleGauge::GetNumSamples	得到 CircleGauge 拟合点总数
SYCircleGauge::GetNumValidSamples	得到 CircleGauge 拟合出来的有效点个数
SYCircleGauge::Draw	绘画 CircleGauge
SYCircleGauge::SetModelCenter	设置圆工具的中心
SYCircleGauge::SetModelRadius	设置圆工具的半径
SYCircleGauge::SetModelTolerance	设置圆工具的测量范围，即环宽度
SYCircleGauge::GetModelCenter	获取圆工具中心
SYCircleGauge::GetModelCenterX	获取圆工具中心 X 坐标
SYCircleGauge::GetModelCenterY	获取圆工具中心 Y 坐标
SYCircleGauge::GetModelRadius	获取圆工具半径
SYCircleGauge::GetModelTolerance	获取圆工具测量范围，即环宽度
SYCircleGauge::SetMinArea	设置边缘过渡区域最小面积
SYCircleGauge::GetMinArea	获取边缘过渡区域最小面积
SYCircleGauge::SetThickness	设置切向平滑系数
SYCircleGauge::GetThickness	获取切向平滑系数
SYCircleGauge::SetTransitionIndex	指定返回边缘点的序号
SYCircleGauge::GetTransitionIndex	获取返回边缘点的序号
SYCircleGauge::SetTransitionChoice	设置边缘点选择模式
SYCircleGauge::GetTransitionChoice	获取边缘点选择模式
SYCircleGauge::SetStartAngle	设置圆工具的测量起始角度
SYCircleGauge::GetStartAngle	获取圆工具的测量起始角度
SYCircleGauge::SetSpanAngle	设置圆工具的测量角度范围
SYCircleGauge::GetSpanAngle	获取圆工具的测量角度范围
SYCircleGauge::SetDerSize	设置边缘强度
SYCircleGauge::GetDerSize	获取边缘强度
SYCircleGauge::EnableClusterAnalysis	使能聚类分析功能
SYCircleGauge::GetClusterAnalysis	获取聚类分析功能是否打开状态
SYCircleGauge::SetClusterError	设置聚类分析误差
SYCircleGauge::GetClusterError	获取聚类分析误差

SYCircleGauge::SetPlacement

```
void SetPlacement(int CenterX, int CenterY, int Radius, int Tolerance)
```

设定 CircleGauge 的位置及形状大小

参数：

CenterX	X 中心坐标
CenterY	Y 中心坐标
Radius	半径
Tolerance	CircleGauge 的环形带厚度的一半

SYCircleGauge::GetPlacement

```
CGData GetPlacement()
```

返回 CircleGauge 的位置及形状大小

```
typedef struct CGData
{
    int    CenterX;
    int    CenterY;
    int    Radius;        //the radius of the mid-circle
    int    Tolerance;     //the distance between outer & inner circle and mid-circle

    int    GetX() { return CenterX; }
    int    GetY() { return CenterY; }
    int    GetRadius() { return Radius; }
    int    GetTolerance() { return Tolerance; }
}CGData;
```

SYCircleGauge::GetHitShape

```
Circle_Handle GetHitShape(int CursorX, int CursorY)
```

修改 CircleGauge 当前光标下的形状，配合 SetCursor 和 Drag 函数来使用。

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值

SYCircleGauge::Drag

```
void Drag(int CursorX, int CursorY, float ZoomX, float ZoomY)
```

鼠标操作拖动改变 CircleGauge 的位置及形状大小

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例

SYCircleGauge::Draw

```
void Draw(HDC h_Dc, Drawing_Modes DrawMode, float ZoomX, float ZoomY, int PanX, int PanY)
```

绘画 CircleGauge

参数：

h_Dc	处理目标窗口的设备上下文
DrawMode	绘制的模式，见 Drawing_Modes 表格所示
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

Drawing_Modes	
DRAW_NOMINAL	常规测量工具
DRAW_ACTUAL	实际测量工具
DRAW_SAMPLE_POINTS	实际采样点

```
enum Drawing_Modes
```

```
{  
    DRAW_NOMINAL,        //draw the gauge  
    DRAW_ACTUAL,         //draw the result measured by gauge  
    DRAW_SAMPLE_POINTS  //draw all sample points  
};
```

SYCircleGauge::SetMeasureDirection

`void SetMeasureDirection(MeasureDirection direction)`

设置 CircleGauge 的测量方向，从里往外、从外往里

参数：

direction 测量方向

Mode	描述
From_In_To_Out	从里往外
From_Out_To_In	从外往里

`enum MeasureDirection`

```
{  
    From_In_To_Out,  
    From_Out_To_In  
};
```

SYCircleGauge::GetMeasureDirection

`MeasureDirection GetTransitionType()`

返回值：得到 CircleGauge 的测量方向。

SYCircleGauge::SetTransitionType

`void SetTransitionType(TransitionType mode)`

设置 CircleGauge 的测量极性，从黑到白、从白到黑、白到黑或者黑到白等

参数：

mode 测量极性

Mode	描述
SYG_BW	从黑到黑
SYG_WB	从白到黑
SYG_BW_OR_WB	白到黑或者黑到白

`enum TransitionType`

```
{  
    SYG_BW_OR_WB,  
    SYG_BW,  
    SYG_WB  
};
```



返回值：得到 CircleGauge 的测量极性。

threshold	閾值范围在 0~255 之间
-----------	----------------

获取阈值。

SamplingStep	最小采样间隔为 1 个像素
--------------	---------------

获取 CircleGauge 工具的采样间隔。

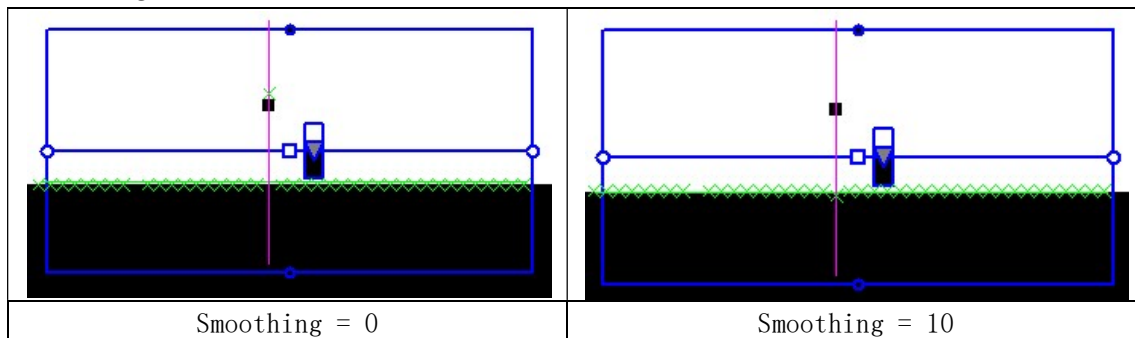
SYCircleGauge::SetSmoothing

`void SetSmoothing(int smoothing)`

设置径向平滑系数，平滑路径上的像素噪点。

参数：

`smoothing` 径向平滑系数，可过滤路径上像素噪点，滤波参数范围在 0~25 之间



拟合路径上有一黑色噪点，调整 Smoothing 做中值滤波，噪点被过滤，最终选择了下方的拟合点。

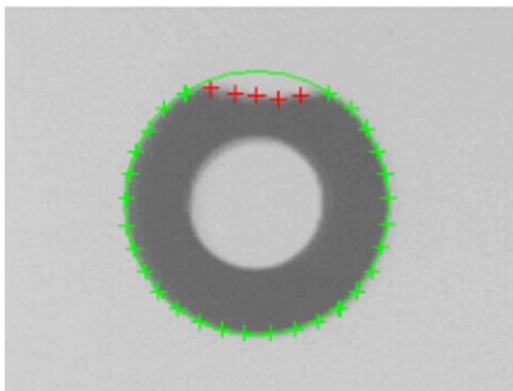
SYCircleGauge::GetSmoothing

`unsigned int GetSmoothing()`

获取径向平滑系数。

【差异点剔除】

差异点是远离有效点集合的点，它的出现会影响形状（直线、圆、矩形）的逼近程度，需要将其剔除。下图所示为剔除圆上的差异点：



SYCircleGauge::SetFilterOutlierThreshold

`void SetFilterOutlierThreshold(float mThreshold)`

设定剔除差异点滤波阈值，该值的倒数即为所有边缘点到拟合出的直线的平均距离。
到拟合直线距离大于该平均距离的点即为差异点。

参数：

mThreshold 滤波阈值

SYCircleGauge::GetFilterOutlierThreshold

`float GetFilterOutlierThreshold()`

获取剔除差异点滤波阈值。

SYCircleGauge::SetFilterOutlierNumPass

`void SetFilterOutlierNumPass(int mNumPass)`

设定剔除差异点滤波次数，在拟合过程中检测差异点的次数。

参数：

mNumPass 滤波次数

SYCircleGauge::GetFilterOutlierNumPass

`unsigned int GetFilterOutlierNumPass()`

获取检测差异点滤波次数。

SYCircleGauge::GetNumSamples

`int GetNumSamples()`

获取 CircleGauge 工具的采样点总数

SYCircleGauge::GetNumValidSamples

```
int GetNumValidSamples()
```

获取 CircleGauge 工具的有效采样点的个数

SYCircleGauge::Measure

```
void Measure(SYGrayImage &Img)
```

进行 CircleGauge 测量。

参数：

Img 源图像

SYCircleGauge::GetMeasuredCircle

```
CircleResult GetMeasuredCircle()
```

获得 CircleGauge 测量结果

```
typedef struct CircleResult
```

```
{
```

```
    float                  CenterX;
```

```
    float                  CenterY;
```

```
    float                  Radius;
```

```
}CircleResult;
```

SYCircleGauge::Attach

```
void Attach(SYCalibration *syWorld)
```

使用 Attach 将 CircleGauge 工具与 SYCalibration 标定坐标系建立关联

参数：

SYCalibration 标定数据

SYCircleGauge::Detach

`void Detach()`

使用 Detach 将 CircleGauge 工具与 SYCalibration 标定坐标系取消关联

SYCircleGauge::SetModelCenter

`void SetModelCenter(int CenterX, int CenterY)`

设置圆工具中心坐标。

参数：

CenterX 中心 X 坐标

CenterY 中心 Y 坐标

SYCircleGauge::SetModelRadius

`void SetModelRadius(int Radius)`

设置圆工具半径。

参数：

Radius 半径

SYCircleGauge::SetModelTolerance

`void SetModelTolerance(int Tolerance)`

设置圆工具的测量范围， 即环宽度。

参数：

Tolerance 测量范围

SYCircleGauge::GetModelCenter

`SYMV::Point2i GetModelCenter( )`

获取圆工具的中心坐标。

SYCircleGauge::GetModelCenterX

`int GetModelCenterX( )`

获取圆工具的中心 X 坐标。

SYCircleGauge::GetModelCenterY

`int GetModelCenterY( )`

获取圆工具的中心 Y 坐标。

SYCircleGauge::GetModelRadius

`int GetModelRadius( )`

获取圆工具的半径。

SYCircleGauge::GetModelTolerance

`int GetModelTolerance( )`

获取圆工具的测量范围。

SYCircleGauge::SetMinArea

`void SetMinArea(int minArea)`

设置边缘过渡区域的最小面积，详细参见 SYPointGauge::SetMinArea 函数。

SYCircleGauge::GetMinArea

```
int GetMinArea( )
```

获取边缘过渡区域的最小面积。

SYCircleGauge::SetThickness

```
void SetThickness(int Thickness)
```

设置切向平滑系数，即求平均对比度的像素点个数。当遇到强噪声点时，通过切向平均拉低噪声点的对比度，达到剔除该强噪声点的目的。

参数：

Thickness 切向平滑系数

SYCircleGauge::GetThickness

```
int GetThickness( )
```

获取切向平滑系数。

SYCircleGauge::SetTransitionIndex

```
void SetTransitionIndex(int idx)
```

设置返回指定序号的测量点信息。当 TransitionType 选择为 SYG_FROM_BEGIN 或 SYG_FROM_END 时，可通过该函数指定要返回第 idx 个测量点信息。

SYCircleGauge::GetTransitionIndex

```
int GetTransitionIndex( )
```

获取被指定返回的测量点序号。

SYCircleGauge::SetTransitionChoice

`void SetTransitionChoice(TransitionChoice mode)`

设置选择边缘点的模式。详细可参见 SYPointGauge::SetTransitionChoice 函数。

参数：

mode 边缘点模式枚举类型

`enum TransitionChoice`

```
{  
    SYG_FROM_BEGIN,  
    SYG_FROM_END,  
    SYG_LARGEST_AMPLITUDE,  
    SYG_LARGEST_AREA,  
    SYG_ALL  
};
```

SYCircleGauge::GetTransitionChoice

`TransitionChoice GetTransitionChoice()`

获取选择边缘点的模式。详细可参见 SYPointGauge::SetTransitionChoice 函数。

SYCircleGauge::SetStartAngle

`void SetStartAngle(float StartAngle)`

设置圆工具测量的起始角度。

SYCircleGauge::GetStartAngle

`float GetStartAngle()`

获取圆工具测量的起始角度。

SYCircleGauge::SetSpanAngle

`void SetSpanAngle(float SpanAngle)`

设置圆工具测量的角度范围。

SYCircleGauge::GetSpanAngle

`float GetSpanAngle( )`

获取圆工具的测量角度范围。

SYCircleGauge::SetDerSize

`void SetDerSize(int DerSize)`

设置边缘强度系数。弱边缘的对比度较低，不利于进行稳定的边缘点分析；通过设置边缘强度，可强化边缘点的响应强度，使其对比度拉高。

参数：

DerSize 边缘强度系数

SYCircleGauge::GetDerSize

`int GetDerSize( )`

获取边缘强度系数。

SYCircleGauge::EnableClusterAnalysis

`void EnableClusterAnalysis(bool bEnable)`

使能聚类分析功能。聚类分析，是一种强效的边缘干扰点剔除方法。以多者为优的原则，即共圆点最多的点集为最优直线拟合点，其它被认为是干扰点。

SYCircleGauge::GetClusterAnalysis

`bool GetClusterAnalysis( )`

获取聚类分析功能是否打开。

SYCircleGauge::SetClusterError

`void SetClusterError(float err)`

设置聚类分析误差。

SYCircleGauge::GetClusterError

`float GetClusterError( )`

获取聚类分析误差。

2.6.5 SYRectangleGauge

List of Function 函数列表说明

Function Name(函数名称)	说明
<code>SYRectangleGauge::SetPlacement</code>	设定矩形工具的位置及形状大小
<code>SYRectangleGauge::GetPlacement</code>	返回矩形工具的位置及形状大小
<code>SYRectangleGauge::GetHitShape</code>	修改矩形工具当前光标下的形状
<code>SYRectangleGauge::Drag</code>	鼠标操作拖动改变矩形工具位置及形状大小
<code>SYRectangleGauge:: SetMeasureDirection</code>	设置矩形工具的测量方向
<code>SYRectangleGauge:: GetMeasureDirection</code>	获取矩形工具的测量方向
<code>SYRectangleGauge::SetTransitionType</code>	设置矩形工具的测量极性
<code>SYRectangleGauge::GetTransitionType</code>	获取矩形工具的测量极性
<code>SYRectangleGauge::SetThreshold</code>	设置阈值
<code>SYRectangleGauge::GetThreshold</code>	得到阈值
<code>SYRectangleGauge::SetSmoothing</code>	径向平滑系数，平滑路径上像素噪点
<code>SYRectangleGauge::GetSmoothing</code>	得到径向平滑系数
<code>SYRectangleGauge::SetSamplingStep</code>	设置矩形工具的采样间隔
<code>SYRectangleGauge::GetSamplingStep</code>	获取矩形工具的采样间隔
<code>SYRectangleGauge::SetFilterOutlierNumPass</code>	设置差异点剔除滤波次数
<code>SYRectangleGauge::GetFilterOutlierNumPass</code>	获取差异点剔除滤波次数
<code>SYRectangleGauge::SetFilterOutlierThreshold</code>	设置差异点剔除滤波阈值
<code>SYRectangleGauge::GetFilterOutlierThreshold</code>	获取差异点剔除滤波阈值
<code>SYRectangleGauge::Attach</code>	将矩形工具工具与 SYCalibration 标定坐标系建立关联
<code>SYRectangleGauge::Detach</code>	将矩形工具工具与 SYCalibration 标定坐标系取消关联
<code>SYRectangleGauge::Measure</code>	进行矩形拟合测量
<code>SYRectangleGauge::GetMeasuredRectangle</code>	获取矩形测量结果，包含中心和宽高等
<code>SYRectangleGauge::GetNumSamples</code>	得到矩形工具拟合点总数

SYRectangleGauge::GetNumValidSamples	得到矩形工具拟合出来的有效点个数
SYRectangleGauge::Draw	绘画矩形工具
SYRectangleGauge::SetModelCenter	设置矩形工具的中心
SYRectangleGauge::SetModelWidth	设置矩形工具的宽度
SYRectangleGauge::SetModelHeight	设置矩形工具的高度
SYRectangleGauge::SetModelTolerance	设置矩形工具的测量范围
SYRectangleGauge::SetModelAngle	设置矩形工具的角度
SYRectangleGauge::GetModelCenter	获取矩形工具中心
SYRectangleGauge::GetModelCenterX	获取矩形工具中心 X 坐标
SYRectangleGauge::GetModelCenterY	获取矩形工具中心 Y 坐标
SYRectangleGauge::GetModelWidth	获取矩形工具宽度
SYRectangleGauge::GetModelHeight	获取矩形工具高度
SYRectangleGauge::GetModelTolerance	获取矩形工具测量范围
SYRectangleGauge::GetModelAngle	获取矩形工具的角度
SYRectangleGauge::SetMinArea	设置边缘过渡区域最小面积
SYRectangleGauge::GetMinArea	获取边缘过渡区域最小面积
SYRectangleGauge::SetThickness	设置切向平滑系数
SYRectangleGauge::GetThickness	获取切向平滑系数
SYRectangleGauge::SetTransitionIndex	指定返回边缘点的序号
SYRectangleGauge::GetTransitionIndex	获取返回的边缘点序号
SYRectangleGauge::SetTransitionChoice	设置边缘点选择模式
SYRectangleGauge::GetTransitionChoice	获取边缘点选择模式
SYRectangleGauge::SetDerSize	设置边缘强度
SYRectangleGauge::GetDerSize	获取边缘强度

SYRectangleGauge::SetPlacement

```
void SetPlacement(int CenterX, int CenterY, int width, int height, int tolerance,  
float angle)
```

设定 RectangleGauge 的位置及形状大小

参数：

CenterX	X 中心坐标
CenterY	Y 中心坐标
width	宽度
height	高度
tolerance	测量范围
angle	角度

SYRectangleGauge::GetPlacement

```
RGData GetPlacement()
```

返回 RectangleGauge 的位置及形状大小

```
typedef struct RGData  
{  
    int CenterX;  
    int CenterY;  
    int Width;  
    int Height;  
    int Tolerance;  
    float Angle;  
  
    int GetX() { return CenterX; }  
    int GetY() { return CenterY; }  
    int GetWidth() { return Width; }  
    int GetHeight() { return Height; }  
    int GetTolerance() { return Tolerance; }  
    float GetAngle() { return Angle; }  
  
    RGData()  
    {  
        CenterX = 0;  
        CenterY = 0;  
        Width = 0;  
        Height = 0;  
        Tolerance = 0;  
        Angle = 0.f;  
    }  
}
```

```
}RGData;
```

SYRectangleGauge::GetHitShape

```
Rectangle_Handle GetHitShape(int CursorX, int CursorY, float ZoomX, float ZoomY, int PanX, int PanY)
```

修改 RectangleGauge 当前光标下的形状，配合 SetCursor 和 Drag 函数来使用。

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值
ZoomX	绘画 X 比例
ZoomY	绘画 Y 比例
PanX	绘画 X 偏移
PanY	绘画 Y 偏移

SYRectangleGauge::Drag

```
void Drag(int CursorX, int CursorY, float ZoomX, float ZoomY, int PanX, int PanY)
```

鼠标操作拖动改变 RectangleGauge 的位置及形状大小

参数：

CursorX	光标水平坐标值
CursorY	光标垂直坐标值
ZoomX	绘画 X 比例
ZoomY	绘画 Y 比例
PanX	绘画 X 偏移
PanY	绘画 Y 偏移

SYRectangleGauge::Draw

```
void Draw(HDC h_Dc, Drawing_Modes DrawMode, float ZoomX, float ZoomY, int PanX, int PanY)
```

绘画 RectangleGauge

参数：

h_Dc	处理目标窗口的设备上下文
DrawMode	绘制的模式，见 Drawing_Modes 表格所示
ZoomX	绘画 X 比例
ZoomY	绘画 Y 比例
PanX	绘画 X 偏移
PanY	绘画 Y 偏移

Drawing_Modes	
DRAW_NOMINAL	常规测量工具
DRAW_ACTUAL	实际测量工具
DRAW_SAMPLE_POINTS	实际采样点

```
enum Drawing_Modes
{
    DRAW_NOMINAL,        //draw the gauge
    DRAW_ACTUAL,         //draw the result measured by gauge
    DRAW_SAMPLE_POINTS   //draw all sample points
};
```

SYRectangleGauge::SetMeasureDirection

void SetMeasureDirection(*MeasureDirection direction*)

设置 RectangleGauge 的测量方向，从里往外、从外往里

参数：

direction 测量方向

Mode	描述
From_In_To_Out	从里往外
From_Out_To_In	从外往里

```
enum MeasureDirection
{
    From_In_To_Out,
    From_Out_To_In
};
```

SYRectangleGauge::GetMeasureDirection

MeasureDirection GetTransitionType()

返回值：得到 RectangleGauge 的测量方向。

SYRectangleGauge::SetTransitionType

`void SetTransitionType(TransitionType mode)`

设置 RectangleGauge 的测量极性，从黑到白、从白到黑、白到黑或者黑到白等

参数：

mode 测量极性

Mode	描述
SYG_BW	从黑到黑
SYG_WB	从白到黑
SYG_BW_OR_WB	白到黑或者黑到白

```
enum TransitionType
{
    SYG_BW_OR_WB,
    SYG_BW,
    SYG_WB
};
```

SYRectangleGauge::GetTransitionType

`TransitionType GetTransitionType(void)`

返回值：得到 RectangleGauge 的测量极性。

SYRectangleGauge::SetThreshold

`void SetThreshold(int threshold)`

设定阈值。

参数：

threshold 阈值范围在 0~255 之间

SYRectangleGauge::GetThreshold

`unsigned int GetThreshold()`

获取阈值。

YRectangleGauge::SetSamplingStep

`void SetSamplingStep(int SamplingStep)`

设定 RectangleGauge 工具的采样间隔。

参数：

SamplingStep 最小采样间隔为 1 个像素

SYRectangleGauge::GetSamplingStep

`unsigned int GetSamplingStep()`

获取 RectangleGauge 工具的采样间隔。

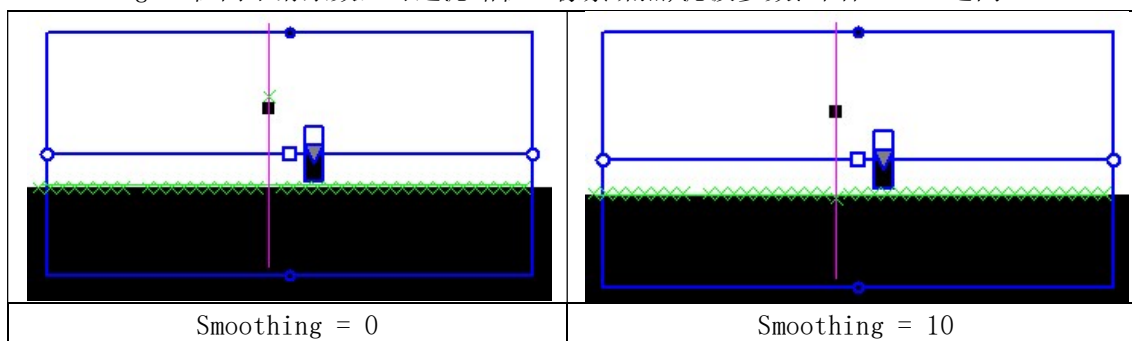
SYRectangleGauge::SetSmoothing

`void SetSmoothing(int smoothing)`

设置径向平滑系数，平滑路径上的像素噪点。

参数：

smoothing 径向平滑系数，可过滤路径上像素噪点，滤波参数范围在 0~25 之间



拟合路径上有一黑色噪点，调整 Smoothing 做中值滤波，噪点被过滤，最终选择了下方的拟合点。

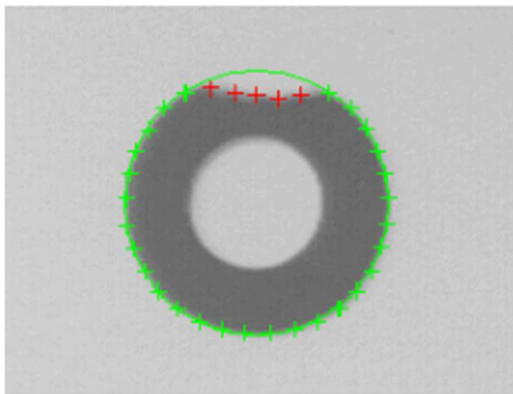
SYRectangleGauge::GetSmoothing

`unsigned int GetSmoothing()`

获取径向平滑系数。

【差异点剔除】

差异点是远离有效点集合的点，它的出现会影响形状（直线、圆、矩形）的逼近程度，需要将其剔除。下图所示为剔除圆上的差异点：



SYRectangleGauge::SetFilterOutlierThreshold

`void SetFilterOutlierThreshold(float mThreshold)`

设定剔除差异点滤波阈值，该值的倒数即为所有边缘点到拟合出的直线的平均距离。
到拟合矩形距离大于该平均距离的点即为差异点。

参数：

mThreshold 滤波阈值

SYRectangleGauge::GetFilterOutlierThreshold

`float GetFilterOutlierThreshold()`

获取剔除差异点滤波阈值。

SYRectangleGauge::SetFilterOutlierNumPass

`void SetFilterOutlierNumPass(int mNumPass)`

设定剔除差异点滤波次数，在拟合过程中检测差异点的次数。

参数：

mNumPass 滤波次数

SYRectangleGauge::GetFilterOutlierNumPass

`unsigned int GetFilterOutlierNumPass()`

获取检测差异点滤波次数。

SYRectangleGauge::GetNumSamples

```
int GetNumSamples()
```

获取矩形工具的采样点总数

SYRectangleGauge::GetNumValidSamples

```
int GetNumValidSamples()
```

获取矩形工具的有效采样点的个数

SYRectangleGauge::Measure

```
void Measure(SYGrayImage *Img)
```

进行矩形拟合测量。

参数：

Img 源图像

SYRectangleGauge::GetMeasuredRectangle

```
RectangleResult GetMeasuredRectangle()
```

获得矩形工具测量结果

```
typedef struct RectangleResult
{
    float      x;
    float      y;
    float      width;
    float      height;
    float      angle;

    float      GetX() { return x; }
    float      GetY() { return y; }
    float      GetWidth() { return width; }
    float      GetHeight() { return height; }
    float      GetAngle() { return angle; }
}RectangleResult;
```

SYRectangleGauge::Attach

```
void Attach(SYCalibration *syWorld)
```

使用 Attach 将 RectangleGauge 工具与 SYCalibration 标定坐标系建立关联

参数：

SYCalibration	标定数据
---------------	------

SYRectangleGauge::Detach

```
void Detach()
```

使用 Detach 将 RectangleGauge 工具与 SYCalibration 标定坐标系取消关联

SYRectangleGauge::SetModelCenter

```
void SetModelCenter(int CenterX, int CenterY)
```

设置矩形工具中心坐标。

参数：

CenterX	中心 X 坐标
---------	---------

CenterY	中心 Y 坐标
---------	---------

SYRectangleGauge::SetModelWidth

```
void SetModelWidth(int width)
```

设置矩形工具的宽度。

参数：

width	宽度
-------	----

SYRectangleGauge::SetModelHeight

```
void SetModelHeight(int height)
```

设置矩形工具的高度。

参数：

height	高度
--------	----

SYRectangleGauge::SetModelTolerance

```
void SetModelTolerance(int Tolerance)
```

设置矩形工具的测量范围。

参数：

Tolerance	测量范围
-----------	------

SYRectangleGauge::SetModelAngle

```
void SetModelAngle(float angle)
```

设置矩形工具的角度。

参数：

angle	角度
-------	----

SYRectangleGauge::GetModelCenter

```
SYMV::Point2i GetModelCenter( )
```

获取矩形工具的中心坐标。

SYRectangleGauge::GetModelCenterX

```
int GetModelCenterX( )
```

获取矩形工具的中心 X 坐标。

SYRectangleGauge::GetModelCenterY

```
int GetModelCenterY( )
```

获取矩形工具的中心 Y 坐标。

SYRectangleGauge::GetModelWidth

```
int GetModelWidth( )
```

获取矩形工具的宽度。

SYRectangleGauge::GetModelHeight

`int GetModelHeight( )`
获取矩形工具的高度。

SYRectangleGauge::GetModelTolerance

`int GetModelTolerance( )`
获取矩形工具的测量范围。

SYRectangleGauge::GetModelAngle

`float GetModelAngle( )`
获取矩形工具的角度。

SYRectangleGauge::SetMinArea

`void SetMinArea(int minArea)`
设置边缘过渡区域的最小面积，详细参见 SYPointGauge::SetMinArea 函数。

SYRectangleGauge::GetMinArea

`int GetMinArea( )`
获取边缘过渡区域的最小面积。

SYRectangleGauge::SetThickness

`void SetThickness(int Thickness)`
设置切向平滑系数，即求平均对比度的像素点个数。当遇到强噪声点时，通过切向平均拉低噪声点的对比度，达到剔除该强噪声点的目的。

参数：

Thickness 切向平滑系数

SYRectangleGauge::GetThickness

```
int GetThickness( )
```

获取切向平滑系数。

SYRectangleGauge::SetTransitionIndex

```
void SetTransitionIndex(int idx)
```

设置返回指定序号的测量点信息。当 TransitionType 选择为 SYG_FROM_BEGIN 或 SYG_FROM_END 时，可通过该函数指定要返回第 idx 个测量点信息。

SYRectangleGauge::GetTransitionIndex

```
int GetTransitionIndex( )
```

获取被指定返回的测量点序号。

SYRectangleGauge::SetTransitionChoice

```
void SetTransitionChoice(TransitionChoice mode)
```

设置选择边缘点的模式。详细可参见 SYPointGauge::SetTransitionChoice 函数。

参数：

mode 边缘点模式枚举类型

```
enum TransitionChoice
```

```
{
```

```
    SYG_FROM_BEGIN,
```

```
    SYG_FROM_END,
```

```
    SYG_LARGEST_AMPLITUDE,
```

```
    SYG_LARGEST_AREA,
```

```
    SYG_ALL
```

```
};
```

SYRectangleGauge::GetTransitionChoice

```
TransitionChoice GetTransitionChoice( )
```

获取选择边缘点的模式。详细可参见 SYPointGauge::SetTransitionChoice 函数。

SYRectangleGauge::SetDerSize

```
void SetDerSize(int DerSize)
```

设置边缘强度系数。弱边缘的对比度较低，不利于进行稳定的边缘点分析；通过设置边缘强度，可强化边缘点的响应强度，使其对比度拉高。

参数：

DSize 边缘强度系数

SYRectangleGauge::GetDerSize

```
int GetDerSize( )
```

获取边缘强度系数。

2.7、SYFind Classes and Class Members

List of Function 函数列表说明

Function Name(函数名称)	说明
SYFind::SetMinScore	设置最小相似分数
SYFind::GetMinScore	获取最小相似分数
SYFind::SetMinAngle	设置匹配允许的最小角度
SYFind::SetMaxAngle	设置匹配允许的最大角度
SYFind::SetAnisotropicScaleXY	设置 XY 不等比例变化状态
SYFind::SetMinScale	设置匹配允许的最小缩放比例系数
SYFind::SetMaxScale	设置匹配允许的最大缩放比例系数
SYFind::SetMinScaleX	设置 X 方向最小比例系数
SYFind::SetMaxScaleX	设置 X 方向最大比例系数
SYFind::SetMinScaleY	设置 Y 方向最小比例系数
SYFind::SetMaxScaleY	设置 Y 方向最大比例系数
SYFind::SetCoarseLevel	设置粗糙度系数
SYFind::GetCoarseLevel	获取粗糙度系数
SYFind::SetFineLevel	设置精细度系数
SYFind::GetFineLevel	获取粗糙度系数
SYFind::SetPivot	设置匹配输出结果偏移量
SYFind::GetPivot	获取匹配输出结果偏移量
SYFind::SetMaxInstances	设置匹配个数
SYFind::LearnPattern	学习模板轮廓
SYFind::ModifyPattern	模板检查或修改
SYFind::SetFilterLenght	设置剔除杂点强度
SYFind::GetFilterLenght	获取剔除杂点强度
SYFind::SetGradientThreshold	设置模板轮廓中的最小对比度
SYFind::GetGradientThreshold	获取模板轮廓中的最小对比度
SYFind::SetMinContourSize	设置模板轮廓中的最小长度
SYFind::GetMinContourSize	获取模板轮廓中的最小长度

<code>SYFind::Find</code>	目标匹配
<code>SYFind::DrawModel</code>	绘画被学习的模板轮廓
<code>SYFind::DrawPosition</code>	根据索引号画出匹配到的目标
<code>SYFind::DrawPositions</code>	画出匹配到的所有目标
<code>SYFind::GetNumPositions</code>	得到匹配到的目标个数
<code>SYFind::GetPosition</code>	获取匹配结果（如匹配分数、坐标、角度等）
<code>SYFind::IsLearnt()</code>	判断是否已经进行模板学习
<code>SYFind::GetNumberLevelOfPyramid</code>	获取内部金字塔图像的层数
<code>SYFind::SetPolarityMode</code>	设置极性模式
<code>SYFind::GetPolarityMode</code>	获取极性模式
<code>SYFind::SetSubpixelMode</code>	设置亚像素模式
<code>SYFind::GetSubpixelMode</code>	获取亚像素模式
<code>SYFind::SetOverLapRate</code>	设置重叠率
<code>SYFind::GetOverLapRate</code>	获取重叠率
<code>SYFind::SetTimeOut</code>	设置超时时间
<code>SYFind::GetTimeOut</code>	获取超时时间
<code>SYFind::SetMinContrast</code>	设置图像的最小对比度
<code>SYFind::GetMinContrast</code>	获取当前设置的图像最小对比度
<code>SYFind::AutoMinContrast</code>	自动计算图像的最小对比度
<code>SYFind::SetGreediness</code>	设置启发率系数
<code>SYFind::GetGreediness</code>	获取启发率系数



SYFind::SetMinScale

```
void SetMinScale(float minScale)
```

设置匹配允许的最小缩放比例。

参数:

MinScale

最小缩放比例

SYFind::SetMaxScale

```
void SetMaxScale(float maxScale)
```

设置匹配允许的最大缩放比例。

参数:

MaxScale

最大缩放比例

SYFind::SetMinScaleX

```
void SetMinScaleX(float minScaleX)
```

设置 X 方向最小比例系数。

参数:

MinScaleX

X 最小缩放比例

SYFind::SetMaxScaleX

```
void SetMaxScaleX(float maxScaleX)
```

设置 X 方向最大比例系数。

参数:

MaxScaleX

X 最大缩放比例

SYFind::SetMinScaleY

```
void SetMinScaleY(float minScaleX)
```

设置 Y 方向最小比例系数。

参数:

MinScaleY

Y 最小缩放比例

SYFind::SetMaxScaleY

`void SetMaxScaleY(float maxScaleY)`

设置 X 方向最大比例系数。

参数：

MaxScaleY

Y 最大缩放比例

SYFind::SetCoarseLevel

`void SetCoarseLevel (int Level)`

设置粗糙度系数，即匹配从第几层开始搜索目标。

参数：

Level

层数

返回值：无

匹配方法是采用金字塔方法，以较低的分辨率开始搜索。图像和图案的大小逐渐减小，在检测到最低分辨率下的最佳相关性分数后，SYFind 将逐步“返回”到正常图像分辨率。所以层数越大时间越久。

原始图片	模板	
3 		16 x 11 176 pixels Resolution 1:8 64x less pixels
2 		31 x 21 651 pixels Resolution 1:4 16x less pixels
1 		62 x 41 2,542 pixels Resolution 1:2 4x less pixels
0 		124 x 82 10,168 pixels Normal resolution 1:1

SYFind::GetCoarseLevel

`int GetCoarseLevel ( )`

获取粗糙度系数。

SYFind::SetFineLevel

```
void SetFineLevel(int Level)
```

设置粗细度系数， 即从第几层金字塔图像退出搜索。

SYFind::GetFineLevel

```
int GetFineLevel()
```

获取粗细度系数。

SYFind::SetPivot

```
void SetPivot(float X, float Y)
```

设置匹配输出结果偏移量。默认地，XY 的输出位置为模板子图的中心；通过该函数可修改输出 XY 的位置（相对于模板子图中心）。

参数：

x 匹配输出结果 x 方向的 offset

y 匹配输出结果 y 方向的 offset

返回值： 无

SYFind::GetPivot

```
SYMV::Point2f GetPivot()
```

获取匹配输出结果偏移量。

SYFind::SetMaxInstances

```
void SetMaxInstances(int maxInstances)
```

设置最大匹配个数。

参数：

maxInstances 匹配模板数量

SYFind::LearnPatten

```
bool LearnPattern(SYGrayImage *pat, SYGrayImage *mask)
```

学习模板轮廓。

参数：

Pat	模板
Mask	掩码

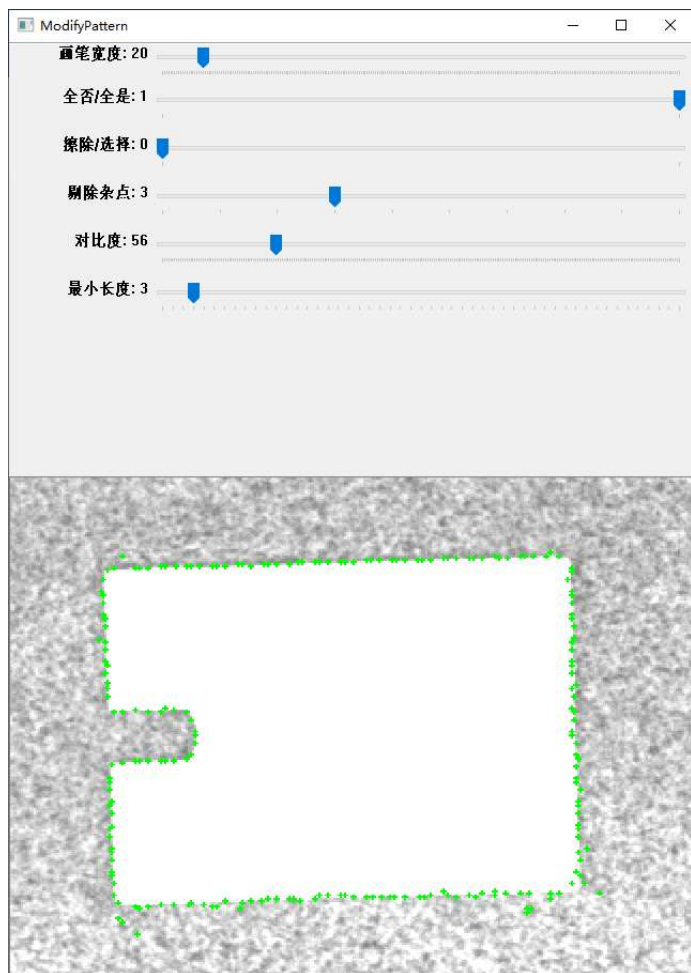
SYFind::ModifyPattern

```
bool ModifyPattern(SYGrayImage *pat, SYGrayImage *mask, int Language)
```

对学习的模板轮廓进行检查或修改。执行该函数弹出窗口，对模板轮廓进行观察，并提供修改功能。如下图所示。

参数：

Pat	模板
Mask	掩码
Language	语言 0—中文 1—英文



SYFind::SetFilterLenght

`void SetFilterLenght(int filterLenght)`

设置剔除杂点强度。该参数对应模板修改界面中的【剔除杂点】参数。

参数：

<code>filterLenght</code>	剔除杂点强度
---------------------------	--------

SYFind::GetFilterLenght

`void GetFilterLenght ( )`

获取剔除杂点强度。该参数对应模板修改界面中的【剔除杂点】参数。

SYFind::SetGradientThreshold

`void SetGradientThreshold(int gradientthreshold)`

设置模板轮廓中的最小对比度。该参数对应模板修改界面中的【对比度】参数。

参数：

<code>gradientthreshold</code>	剔除杂点强度
--------------------------------	--------

SYFind::GetGradientThreshold

`int GetGradientThreshold ( )`

获取模板轮廓中的最小对比度。该参数对应模板修改界面中的【对比度】参数。

SYFind::SetMinContourSize

`void SetMinContourSize(int minsize)`

设置模板轮廓中的最小长度。该参数对应模板修改界面中的【最小长度】参数。

参数：

<code>minsize</code>	最小长度
----------------------	------

SYFind::GetMinContourSize

`int GetMinContourSize ( )`

获取模板轮廓中的最小长度。该参数对应模板修改界面中的【最小长度】参数。

SYFind::IsLearnt

`bool IsLearnt()`

判断是否已经进行模板学习，true 为已经进行模板学习，false 则为未进行模板学习。

SYFind::DrawPosition

`void DrawPosition(HDC dc, int Idx, float ZoomX, float ZoomY, int PanX, int PanY)`

根据索引号画出匹配到的目标

参数：

dc	处理目标窗口的设备上下文
Idx	匹配到的模板索引号
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYFind::DrawPositions

`void DrawPositions(HDC dc, float ZoomX, float ZoomY, int PanX, int PanY)`

画出所有匹配到的目标

参数：

dc	处理目标窗口的设备上下文
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例
PanX	水平方向的偏移量
PanY	垂直方向的偏移量

SYFind::DrawModel

```
void DrawModel(HDC dc, float zoomX, float zoomY, int PanX, int PanY)
```

绘画被学习的模板轮廓

参数：

dc	处理目标窗口的设备上下文
ZoomX	水平方向的缩放比例
ZoomY	垂直方向的缩放比例

SYFind::Find

```
bool Find(SYGrayImage *imp)
```

目标匹配。

参数：

Img	源图像
-----	-----

SYFind::GetNumPositions

```
int GetNumPositions()
```

返回值：返回匹配到的目标个数

SYFind::GetPosition

```
FoundPosition* GetPosition(int Index)
```

获取匹配结果（如匹配分数、坐标、角度等。

参数：

Index	匹配到的模板索引号
-------	-----------

返回值：FoundPosition，返回值里边包含匹配到的中心点，角度和相似度等相关信息。

```
typedef struct FoundPosition
```

```
{
```

float	x;
float	y;
float	Angle;
float	Score;
float	Scale;
bool	bSel;

```
}FoundPosition;
```

SYFind::GetNumberLevelOfPyramid

```
int GetNumberLevelOfPyramid( )
```

获取内部金字塔图像的层数。

SYFind::SetPolarityMode

```
void SetPolarityMode(SYFND_POLARITY_MODE polarityMode)
```

设置极性模式。

参数：

polarityMode 极性模式枚举类型

```
enum SYFND_POLARITY_MODE
```

```
{
```

```
    SYFndPolarityMode_Normal,                      正常极性，与模板一致
```

```
    SYFndPolarityMode_Inverse,                    相反极性
```

```
    SYFndPolarityMode_Any                         任意极性
```

```
};
```

SYFind::GetPolarityMode

```
SYFND_POLARITY_MODE GetPolarityMode( )
```

获取极性模式。

SYFind::SetSubpixelMode

```
void SetSubpixelMode(SYFND_SUBPIXEL_MODE subpixelMode)
```

设置亚像素模式。

参数：

subpixelMode 亚像素模式枚举类型

```
enum SYFND_SUBPIXEL_MODE
```

```
{
```

```
    SYFndSubpixelMode_Interpolation,              插值法
```

```
    SYFndSubpixelMode_Least_Squares              最小二乘法
```

```
};
```

SYFind::GetSubpixelMode

SYFND_SUBPIXEL_MODE GetSubpixelMode()

获取亚像素模式。

SYFind::SetOverLapRate

void SetOverLapRate(float overlpratio)

设置重叠率。重叠率描述两个目标互相重叠面积与整个目标面积的比例。当两个目标重叠比例大于重叠率时，分数小者将被从目标结果序列剔除。

参数：

overlpratio 重叠率

SYFind::GetOverLapRate

float GetOverLapRatio()

获取重叠率。

SYFind::SetTimeOut

void SetTimeOut(unsigned int MStime)

设置超时时间，单位毫秒，数值范围[10 10000]。

参数：

MStime 超时时间

SYFind::GetTimeOut

unsigned int GetTimeOut()

获取超时时间。

SYFind::SetMinContrast

```
void SetMinContrast(int minContrast)
```

设置图像边缘的最小对比度。图像中边缘对比度小于该值的，该点的计分贡献为 0。

参数：

minContrast 最小对比度

SYFind::GetMinContrast

```
int GetMinContrast( )
```

获取当前设置的图像边缘最小对比度。

SYFind::AutoMinContrast

```
int AutoMinContrast(SYGrayImage &img)
```

根据图像中的噪声比率，自动计算出图像中结构的最小对比度。

SYFind::SetGreediness

```
void SetGreediness(float greediness)
```

设置启发率数值，范围[0 1]。数值越小，搜索时间越长，但不容易丢失目标。数值越大，搜索时间越快，但有机率丢失目标。

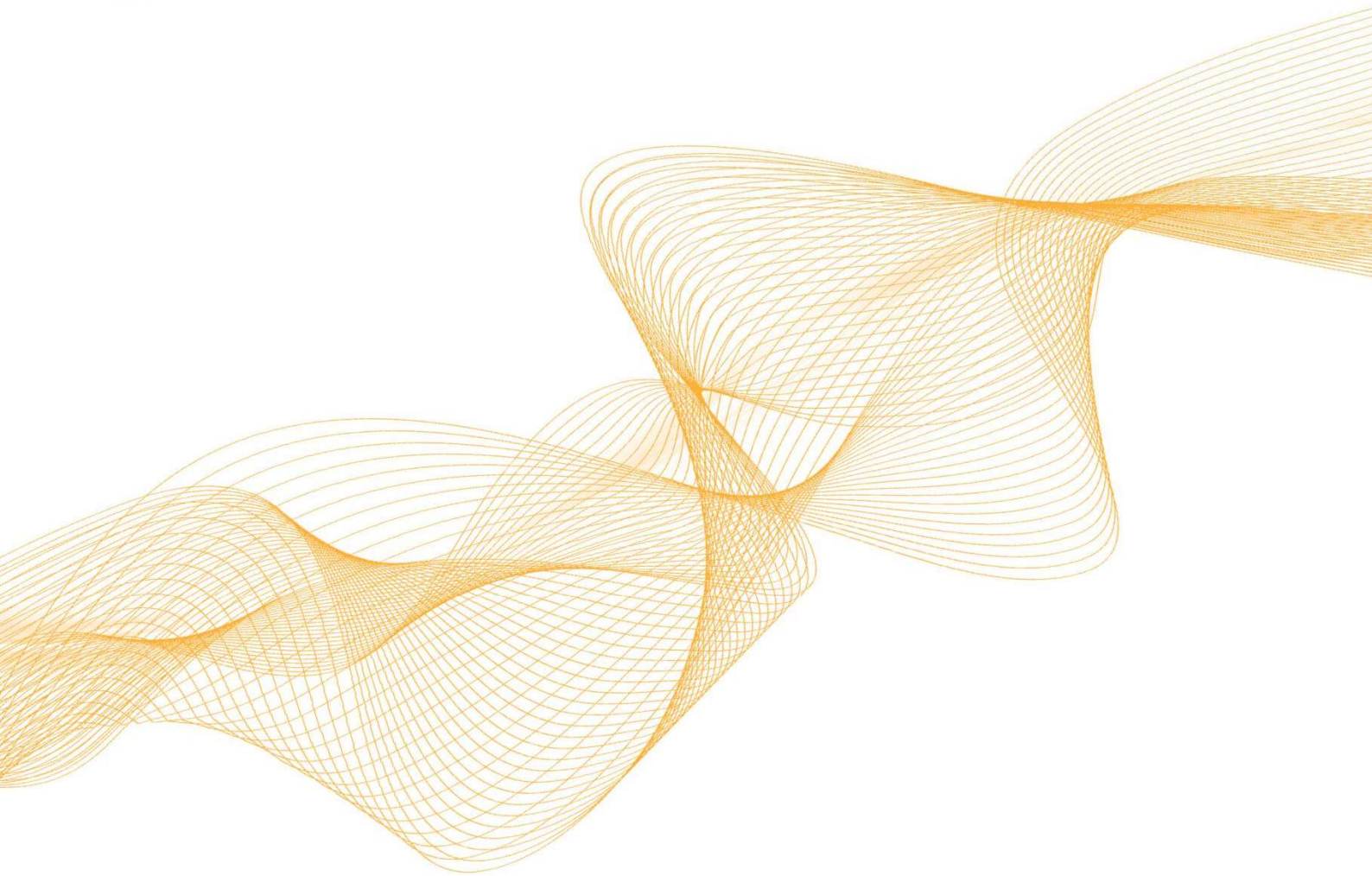
参数：

greediness 启发率

SYFind::GetGreediness

```
float GetGreediness( )
```

获取启发率。



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