



VizCam 6000 series



VizCam 8000 series

Smart Cameras.Smart Choice.

Contact us: Tel: 65 - 6297 0880 or E-mail: enquiry@vizcam.net

Contents

1	介 绍	18
1.1	系统要求	19
	编程系统	19
	目标系统	19
1.2	安装	20
	从 CD/DVD 上安装	20
	下载并安装	20
1.3	第一步	21
2	用户界面	22
2.1	启动	23
	标准模式	26
	远程模式	27
	GUI 设计工具	28
2.2	启动远程系统	31
2.3	检查过程	32
2.4	配置界面	32
2.5	主菜单	33
	文件 (File)	33
	编辑 (Edit)	34
	查看 (View)	35
	图像分析 (Image analysis)	36
	选项 (Options)	36
	窗口 (Window)	37
	帮助 (Help)	37
2.6	程序编辑器 (Program Editor)	38
	快捷键	38
	关联菜单	39
	序号列 (No. Column)	41
	停止栏 (Column Stop)	42
	图形栏 (Column Graphic)	43

文本栏 (Column Text)	44
反转 (Inversion)	44
命令栏 (Command column)	45
备注栏 (Comment Column)	45
2.7 相机取景器 (Camera Viewer)	46
相机取景器功能	47
相机取景器光标	48
相机取景器导航	48
2.8 命令集 (Instruction Set)	49
3 菜单	50
3.1 配置窗口和矩形探头	51
配置窗口 (Configure window)	52
3.2 配置圆和圆形探头	52
配置圆形探头 (Configure Circle Probe)	52
配置圆 (Configure Circle)	53
3.3 图像分析 (Image analysis)	53
图像检测 (Image inspector)	53
强度区显示 (Intensity zone display)	54
伪彩色显示 (Pseudo-color display)	55
显示轮廓 (Display contour)	56
线轮廓 (Line profile)	57
直方图 (Histogram)	57
保存像素值到文件 (Save pixle value in file)	58
3.4 应用参数 (Application Parameters)	59
自动程序控制 (Automatic program control)	59
检查过程 (Check process)	60
系统 (System)	61
配置 (Configuration)	61
选择硬件 (Hardware selection)	62
选择语言 (Language selection)	62
触发 (Trigger)	63
3.5 运行参数 (Operating parameters)	64
图像处理 (Image processing)	64
边缘检测参数 (默认) (Edge detection parameter (default))	64

探头方法（对比）（Probe method (cotrast)）	65
边缘选择（Edge selection）	65
校准（Calibration）	66
相机（Cameras）	66
属性（Properties）	67
智能相机属性	67
FireWire 相机参数	69
全局容差（Global Tolerances）	71
3.6 全局值（Global Value）	72
3.7 全局容差（Global Tolerances）	74
3.8 寄存器查看器（Register Viewer）	75
3.9 图形编辑器（Pattern Editor）	76
图形（Pattern）	76
点（Points）	77
创建相关性绘图（Pattern creation for the correlation）	78
3.10 文本终端（Text Terminal）	79
3.11 数字输入/输出（Digital I/O）	80
输入/输出表选项（I/O form options）	80
修改输入/输出名称（Changing I/O names）	80
反转输入/输出条件（Invert In-/Output conditions）	81
3.12 相机选项（Camera Options）	82
相机设置（Camera Settings）	82
高级模式（Expert Mode）（相机设置）	83
调试模式（Debug Mode）	83
编码器（Encoder）	83
输入/输出触发（I/O Trgger）	84
分级模式（Binning Mode）	84
3.13 远程设备控制（Remote Device Control）	85
下拉菜单（Pull-Down Menu）	85
远程设备（Rmote Device）	85
管理设备文件（Manage Device files）	86
激活检查程序（Active inspection program）	86
概览（Overview）	86
3.14 配置远程设备（Remote Device Configuration）	87

属性 (Properties)	88
更新相机/保存备份 (Update camera / restore backup)	90
3.15 保存点列表 (Store Point list)	91
3.16 重置变量 (Reset variables)	91
3.17 关闭打开的通讯端口 (Close Open Sockets)	91
3.18 相机命令集 (Instruction Set of the Camera)	91
3.19 密码管理 (可选) (Password Admin)	92
3.20 SmartBuilder 软件的数据结构	93
SmartBuilder 软件的寄存器	93
值 (Values)	95
点 (Points)	95
直线 (Straight lines)	96
圆 (Circles)	96
点列表 (Point list)	96
全局字符串 (Global String)	96
选取列表 (Pick-up Lists)	96
4 命令	97
4.1 容差和警戒限	98
容差 (Tolerance)	98
全局容差 (Global Tolerances)	98
值修正 (Value correction)	99
色彩选择表 (Color Selection Form)	99
确定探头参数	100
探头位置 (Probe position)	100
探头参数 (Probe parameter)	100
确定探头参数 (Probe parameter determination)	101
对象颜色 (Object color)	101
结果点 (Result point)	101
探测方法 (Probing method)	102
专业参数 (Expert parameter)	102
点的数量 Number of points	102
4.2 命令概述	103
访问变量	103
名称 (Name)	103

运行 (Operation)	103
变量类型 (Variable type)	104
数学 (Arithmetic)	104
元素1/元素2 (Element 1 / Element 2)	104
运算 (Operator)	104
角 (Angle)	105
定义角 (Angle definition)	105
第一条线/第二条线 (First straight line / Second straight line)	105
容差 (Tolerance)	105
检查区域 (Area check)	106
亮度 (Brightness)	106
异步图像采集 (Asynchronous image capture)	107
运行 (Operation)	107
条码 (Bar code)	108
条码类型 (Code type)	108
代码 (Code)	108
检查规范 (Check norm)	108
输出 (Output)	109
窗口 (Window)	109
亮度调整 (Brightness adjustment)	110
方法 (Method)	110
参考强度 (Reference intensities)	110
检查区域 (Correction area)	111
校准 (Calibration)	112
改变 (Change of)	112
调用外部功能 (Call for external function)	113
定义图像区 (Define image section)	113
增加结果到文本字符串 (Add result to text string)	113
用容差评估结果 (Evaluate result with tolerances)	113
游标卡尺 (水平/垂直) (Calliper (hor. /ver.))	114
设置 (Settings)	114
搜索区域 (Search area)	114
对象颜色 (Object color)	115
方向 (Direction)	115

自由游标卡尺 (Caliper free)	116
配置卡尺 (Configure caliper)	116
动态探头方向 (Dynamical probe direction)	117
卡尺两颚的探头参数 (Probe parameters for both measurement jaws)	118
相机控制 (Camera Control)	119
类型 (Type)	119
动作 (Action)	120
全局容差 (Global Tolerances)	120
IEEE1394, FireWire 寄存器访问 (Register access)	121
地址 (Address)	121
Quadlet	121
相机 (Camera)	121
相机设置 (Camera settings)	122
高级模式 (Expert Mode)	122
调试模式 (Debug mode)	122
编码器 (Encoder)	123
分级模式 (Binning Mode)	123
改变灰度值 (Change gray value)	124
窗口 (Window)	124
选项 (Options)	124
字符识别 (四叉树) (Character recognition (Quad Trees))	125
类列表 (Class list)	125
不关心区域 (Don't care-Areas)	125
识别输出 (Recognition output)	126
字体尺寸 (Font size)	126
容差 (Tolerances)	126
字符分隔 (Character separation)	127
窗口位置 (Window position)	127
分隔参数 (Separating parameter)	127
字符数量 (Number of characters)	128
聚类 (Clustering)	128
区域 (Area)	128
中期值 (Interim result)	128
字符串 (Character string)	129

命令选项 (Command options)	129
检查输入 (Check input)	130
输入 (Input)	130
模式 (Mode)	130
单片机控制 (半导体) ChipControl (Semiconductor)	131
圆形 (Circle)	132
确定圆 (Circle Determination)	132
圆计数器 (Circle counter)	133
兴趣圆 Circle of interest	133
颜色距离 (Color distance)	134
距离类型 (Distance type)	134
目标图像存储 (Target image memory)	135
容差 (Tolerances)	135
色彩评价 (Color evaluation)	136
实施 (Implementation)	137
彩色图像区域 (Color image area)	138
圆形 (Circle)	138
选项 (Options)	138
边缘检查 (Contour check)	139
边缘文件 (Contour file)	139
检查圆的亮度 (Brightness on inspection circle)	139
容差 (Tolerances)	139
边缘追踪 (Contour tracing)	140
窗口位置 (Window position)	140
相关性 (Correlation)	141
方法 (Method)	141
容差 (Tolerance)	141
参考 (Reference)	142
参数 (Parameter)	143
特殊相关性 (Correlation special)	145
列名称 (List name)	145
分隔符 (Separator)	145
匹配 (Match)	145
搜索范围 (Search area)	145

栅格尺寸 (Raster size)	145
点亮度列表 (Point brightness list)	145
计数 (Count objects (blob))	146
亮度 (Brightness)	146
区域 (Area)	146
显示高级参数 (Show expert parameters)	147
对象过滤 (Object filter)	147
备份对象 (Backup of objects)	147
保存 (Save in)	148
排序 (Sorting)	148
显示 (Display)	148
忽视边缘对象 (Ignore boundary objects)	148
扩展属性 (Ext. features)	149
容差 (Tolerances)	149
横截面 (Cross-section)	150
探头位置 (Probe position)	150
探头参数 (Probe parameters)	150
删除背景 (Delete background)	151
确定曲线的点 (Determine point of curve)	152
距离 (Distance)	153
测量距离 (Distance determination)	153
第一点/第二点 (First point/ second point)	153
距离类型 (Distance type)	154
评估结果 (Result evaluation)	154
到直线的距离 (Distance to a straight line)	155
距离类型 (Distance type)	155
参考直线 (Reference straight line)	156
卡尺 (Caliper)	156
容差 (Tolerance)	156
扭曲相关 (Distortion correction)	157
创建修正文件 (Create correction file)	157
执行修正 (Execute correction)	159
重置修正数据 (Reset correction data)	159
动作 (Action)	159

选项 (Options)	160
修正类型 (Correction type)	160
修正 (Correction of)	161
指定采样点的距离 (Specified distance of sampling point)	161
修正范围 (Correction range)	162
未修正像素 (Uncorrectable Pixel)	162
数据矩阵码阅读器 (DMC Reader)	163
执行 (Action)	163
初始化选项	163
读取 (Read)	164
区域 (Area)	164
结果 (Result)	165
值 (Values)	165
文本 (Text)	165
点 (Points)	166
初始化高级选项 (Initializing Expert)	166
高级配置 (Expert configuration)	167
比例 (Proportion)	167
模块 (Module)	167
变焦 (Zoom)	168
显示控制 (Display Control)	168
边计数器 (直线) (Edge counter (straight line))	169
边缘方向 (Edge direction)	169
搜索边缘 (Search for edges)	170
探头 (Probe)	170
容差 (Tolerances)	170
评估 (Evaluation)	171
变量类型 (Variable type)	171
来源 (Source)	171
容差 (Tolerances)	171
过滤工具集 (Filter Tool Set)	172
矩阵过滤 (Matrix Filter)	172
其他过滤	179
按字节过滤 (Bytewise filters)	181

MMX 过滤 (MMX filter)	184
对比过滤 (Comparison filter)	185
转换 (Transformations)	187
二进制过滤 (Binary filter)	188
算术过滤 (Arithmetic filter)	189
转换 (Conversion)	192
执行子程序 (Execute sub-program)	194
过滤 (Filter)	195
过滤类型 (Filtertype)	195
窗口 (Window)	196
图像捕获 (Image capture)	197
运行 (Operation)	197
运行 → 从相机捕获图像 (Operation → Capture image from camera)	198
运行 (Operation)	198
显示选项 (Display options)	198
触发输入 (Trigger input)	199
反跳时间 (Debounce time)	199
超时 (Timeout)	199
用相机捕获 (Capture with camera)	199
运行 → 保存图像 (Operation → Save image)	199
运行 (Operation)	200
保存/加载图像 (Save/ Load image)	200
运行 → 加载图像 (Operation → Load image)	201
显示选项 (Display options)	201
图像源 (Image source)	201
保存/加载图像 (Save/ Load image)	201
图像信息 (Image information)	202
信息 (Informations)	202
聚焦参数 (Focus parameters)	202
图像传输 (Image transfer)	203
接收 (Receiver)	203
执行的外部条件 (External condition of execution)	203
发送图像缓冲 (Image buffer to be sent)	203
UDPPaket Retries	203

图像分辨率 (Image resolution)	203
格式 (Format)	204
发送时间 (Time of sending)	204
执行的内部条件 (Internal condition of execution)	204
兴趣点 (Region of interest (ROI))	204
图形叠加 (Graphical overlay)	204
图像压缩 (Image compression)	204
INI 文件访问 (INI file access)	205
文件 (File)	205
章 (Section)	205
输入 (Entry)	205
访问类型 (Access type)	205
数据类型 (Data type)	206
浮点精度 (Floating point precision)	206
小数位 (Decimal places)	206
选项 (Options)	206
编译器控制 (Interpreter control)	207
根据 (信赖) (According (dependent on))	208
循环选项 (Loop options)	209
测量缝隙宽度 (Measure gap width)	210
探头1/探头2 (Probe 1 / Probe 2)	210
对齐 (Alignment)	211
容差 (Tolerances)	211
点 (Point)	213
确定点 (Point determination)	213
点列表 (Point list)	215
点列表操作 (Point list-Operation)	215
位置调整 (Position adjustment)	217
位置调整 (Position adjustment)	217
探测点 (圆探头) (Probe point (Circle probe))	218
探头 (Probe)	218
探头位置 (Probe position)	218
探头参数 (Probe parameters)	219
方向 (Direction)	219

容差 (Tolerances)	219
探测点 (Probe points)	220
探头 (Probe)	220
探头位置 (Probe position)	220
探头参数	221
容差 (Tolerances)	221
Riehmann 探头 (Riehmann-probe)	222
探头位置 (Probe position)	222
边的方向 (Edge direction)	222
显示高级参数	223
在图像上绘制投影 (Draw projection on image)	223
容差 (Tolerances)	223
环过滤 (Ring-filter)	224
环 (Annulus)	224
参数	224
参数	225
最小和最大角	225
其他颜色 (Object color)	225
RS232	226
接收 (Receive)	226
发送 (Send)	226
脚本编译器 (Script-Interpreter (BASIC))	227
选择颜色通道 (Select color channel)	229
色彩通道选择 (Color channel-selection)	229
设置输出 (Set output)	230
输出 (Output)	230
新状态 (New state)	230
模式 (Mode)	230
设置输出 (Set output)	231
智能匹配 (SmartMatch)	232
搜索范围 (Search area)	232
查找对象 (Find objects)	232
保存位置 (Save position in)	232
在选取列表保存结果 (Save results in pickup)	232

容差 (Tolerances)	233
显示 (Display)	233
扩展选项 (Extended options)	233
角 (Angle)	233
比例 (Scale)	233
排序 (Sorting)	234
扩展 (Extended)	234
开始外部程序 (Start external program)	235
程序 (Program)	235
调用的参数 (Call arguments)	235
选项 (Options)	235
统计 (Statistic)	236
运行 (Operation)	236
秒表 (Stop watch)	237
直线 (Straight line)	238
选项 (Options)	238
文本 (Text)	239
文本 (Text)	240
颜色 (Color)	240
位置 (Position)	240
文字尺寸 (Font size)	240
写文本 (Write text)	240
文本字符串 (Text string)	241
重置 (Reset)	241
输入 (Enter)	242
文本变量 (Text variable)	242
发送 (Send)	243
接收 (Receive)	244
5 实施概念	245
5.1 图像处理解决方案的预先考虑	246
安装位置	246
特殊环境	246
环境条件	246
检查速度 (处理速度)	247

检查组件	247
5.2 光学和照明 (Optics and illumination)	247
光学选择	247
远心镜头	247
照明的选择	247
5.3 生成一检查程序	248
5.4 例子	248
例 1: 查找对象和数洞	248
例 2: 位置调整, 校正和测量	249
例 5: 读码	255
例 6: 对象方向	257
6 配置支持	259
6.1 光标形状	259
7 接口协议	260
7.1 数字 I/O	261
通过数字 I/O 的程序转换	261
转换的运行模式 (没有硬件握手):	261
硬件握手	262
7.2 (RS232, RS485)	262
通过串行接口控制检查程序	262
控制系统协议	263
7.3 LAN 协议 (以太网)	263
通过 LAN 控制检查程序	263
控制系统协议	264
8 BASIC 编译器	265
8.1 SmartBuilder BASIC-编译器	266
8.2 SmartBuilder BASIC 检查命令	267
用户界面功能	267
8.3 SmartBuilder BASIC 语言(EVB)	268
数据类型	268
常量	268
变量	269
阵列	269
操作数	270

类型转换	271
分配	271
If 语句	272
Select 语句	272
For 循环	273
Do 循环	273
功能	274
Set 函数	275
模块	275
8.4 编译器	276
编译器的错误信息	276
处理器的错误信息	276
8.5 模块概览	278
SmartBuilder 图形编译器模块	278
EVB 模块	279
8.6 标准函数	280
标准模块	280
字符串模块	282
8.7 视频输入通道 (EVHDTVIC)	290
图形叠加 (EVHDSOI)	294
图像管理 (EVHDSIM)	298
Image Display Channel (EVHSDC)	303
数字输入输出 (EVHSDIO)	307
EVHDSERM	310
RS232模块 (EVHSD)	311
用户 (EVHDSUSER)	317
8.8 视觉功能	318
点列表	321
选取列表	322
计算	325
坐标系统函数	329
度量矩形探头	335
计量圆	337
绘图功能	340

9	附件	350
9.1	SmartBuilder 软件边缘探测方法	351
	二进制边缘检测法 Binary edge-detection（方法：Binary）	351
	对比边缘探测 Contrast edge-detection（方法：Minimum contrast）	351
	对比边缘探测 Contrast edge-detection（方法：Minimum contrast 2）	351
	（方法：minimum contrast 3）	352
9.2	Active X 接口	353
9.3	文件扩展名	353
9.4	SmartBuilder 命令使用的寄存器	354

1 介绍

SmartBuilder 软件协同相应的硬件为用户提供质量保证的完整光学系统。在这个图像处理软件的发展和实施过程中，始终以为专业人员和外行提供易于解决的光学测量和检测工具为指导思想。

我们为以上两种人员提供最大的使用灵活性，同时要尽可能简化操作。，你可能认为这两者是相互矛盾的，但是，幸运的是，现代的图形用户界面可以通过在显示器上控制图形图标来解决复杂的任务。所以，我们选择微软操作系统，正是因其能够满足这一需要。在 Windows XP 上使用 Smartbuilder，需要设置你的操作系统为经典模式，目前的计算机技术允许使用智能相机，以视频实时执行检验程序。

注:



本手册中记载了 SmartBuilder 框架的扩展功能。也许你的软件只有其中一部分功能，这取决于你买的产品和你的许可证类型。请查看你的许可证以确认你应有的功能。可以随时升级软件版本。

1.1 系统要求

首选要确定系统是单一系统还是双系统。在单一系统中，用户界面和软件是在同一系统中运行的，而在双系统中，用户界面和系统是分开的。

编程系统

目前，软件只能运行在微软系统之上，软件的系统要求如下：

处理器	Pentium 2 @ 500 MHz 或同等处理器以上
内存	最小512 MB RAM
	硬盘上约有500 MB 可用空间
显示器	最小分辨率 1024 x 768
光驱	CD/DVD
接口	USB 或加密狗平行接口（针对 PC 版）
操作系统	Microsoft Windows 2000 SP4以上；XP SP2以上或 VISTA 和 Windows 7

目标系统

运行的系统可在多种操作系统中运行。不只在 Windows 系统，也可以在不同的相机中运行（如 SONY 智能相机或 VizCam 智能相机系列）。

请查看以下支持的操作系统：

	Windows 和 Windows 2K; XP; VISTA; W7
---	-------------------------------------

1.2 安装

安装软件有以下几种方式。

从 CD/DVD 上安装

Smartbuilder 系统通常附带安装 CD / DVD。将 CD/DVD 放入光驱，自动安装程序将启动。如果程序没有自动启动，可通过双击 CD / DVD 中的 Setup.exe 手动安装。

之后软件的安装将由菜单驱动执行，请按程序指令操作。

下载并安装

如果你没有拿到 CD 或 DVD，你可以下载最新版软件。

如果你要从 VizCam 网站下载软件(www.vizcam.net)，你必须先注册。注册后，你会收到带有确认链接的邮件。注册确认后，你可以使用用户名和密码登录，并下载软件。

如果在注册时你已经填写了有效的序列号，你在两天内会收到首选客户访问。这意味着你能够下载更多的文件，如为硬件配置的新的软件运行版本。

双击下载的 Setup.exe 文件，开始安装软件。请按程序指令操作。

1.3 第一步

安装软件后请点击 *Windows 开始* → *程序*，你可以看到增加了一个名字 **SmartBuilder**（如果你在安装过程中更改了名字，请查找你填写的名字）。点击这个入口。**SmartBuilder** 程序有几个 PDF 文件，其中一个叫“*First Steps*”。

那里你可以看到完整的关于 **VizCam** 软件的解释（如怎样配置系统，如何定制）。建议在开始使用前阅读整个文件。

请注意：



请先完整阅读 PDF 文件 *First Steps* 或按一步一步地指令操作，完成硬件和软件的实施。成功地开始 **SmartBuilder** 软件和不同的 **VizCam** 相机的结合。

读过 *First Steps* 后，你已经了解到系统的不同组成部分，我们要开始使用软件以获取经验了。

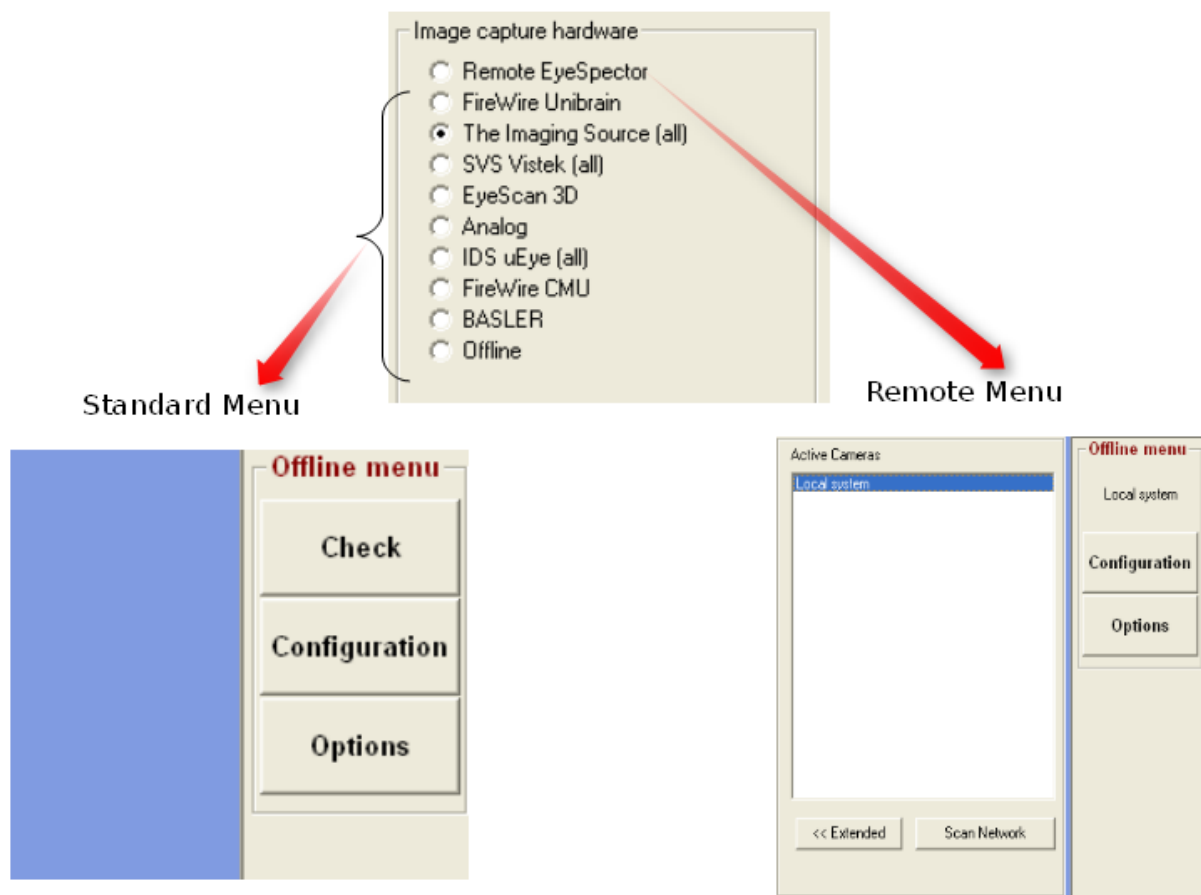
2 用户界面

运行 **SmartBuilder** 软件后，会显示几个用户界面。用户可以决定要显示哪些 **GUI** 运行参数。

如果系统第一次安装 **SmartBuilder** 软件，那么会始终由标准 **GUI** 开始。这取决于是单系统（用户界面和程序在同一电脑中运行），还是远程系统，即有两个开始屏幕。一个是标准的启动模式，一个是远程启动模式（后面详述）。

2.1 启动

根据不同的硬件和系统配置，软件启动有两种模式。选择 *view* → *application parameters* → *hardware* 可看到不同的硬件。如果选择的是远程模式，系统显示远程菜单 *Remote Menu*（右侧图）。如果系统启动时连接了相机，则显示标准菜单 *Standard Menu*（左侧图）。在菜单上点击“Check”按钮进入处理模式。

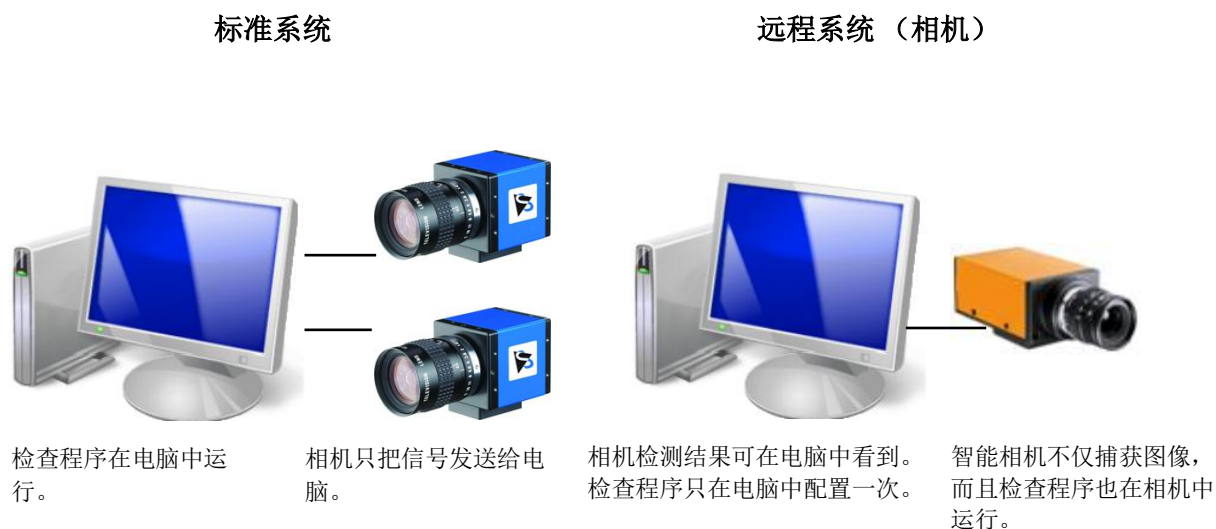


在 “Configuration”下配置程序。在“Check”下进行检查。检查模式要运行在电脑上。

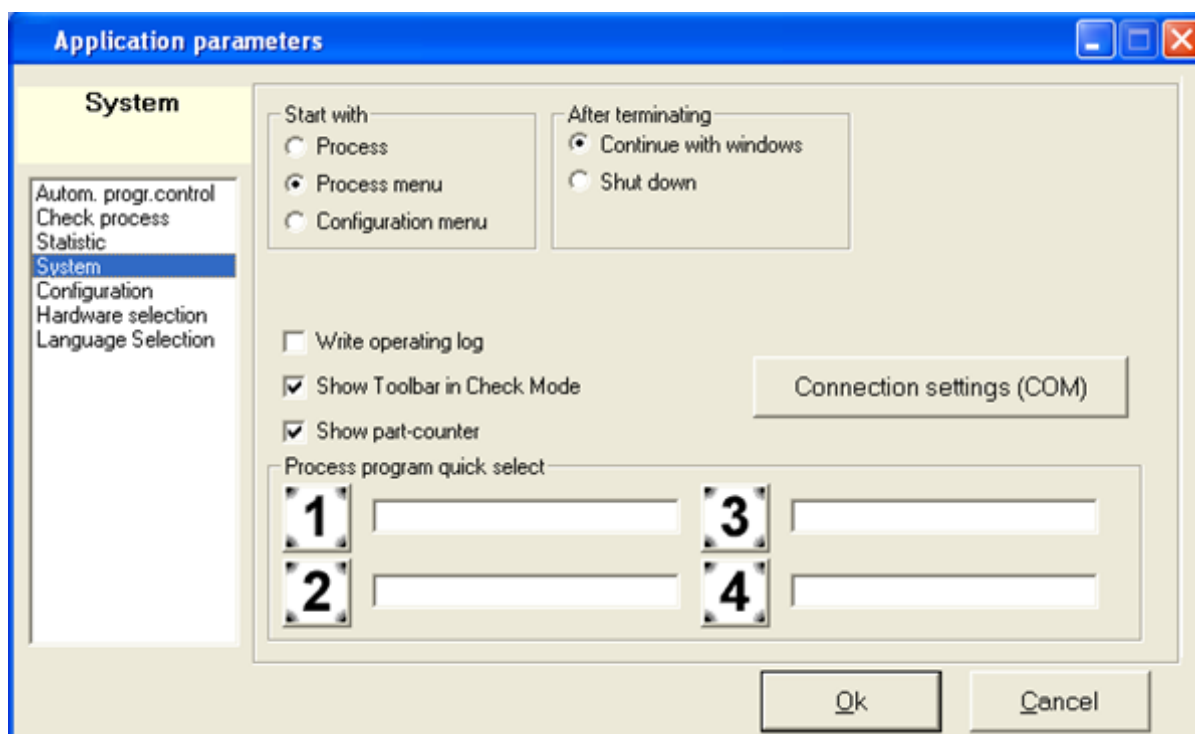
不同于电脑系统，你只能进入配置，因为检查程序在智能相机中运行。如果你进入配置模式，必须先连接相机（正处于检查状态）。现在可以配置一个检查程序并存储于相机。如果断开相机，相机又进入检查模式，并执行最新分配的任务。

除非用户通过控制中心启动系统（根据选择的硬件），系统启动时只有两种情况，一是有三个按钮的标准菜单，另一个是有两个按钮的远程菜单。网络扫描用于查找远程设备。

远程菜单 *Remote Menu* 和 标准菜单 *Standard Menu* 的区别:

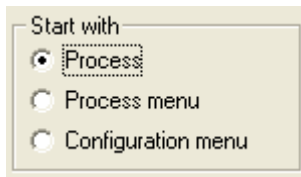


控制中心



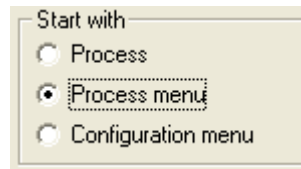
在控制中心 (CC) (*view* → *application parameters* → *system*) 可以定义软件启动后和停止后的动作。描述如

下：



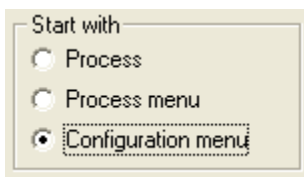
A screenshot of a dialog box titled "Start with". It contains three radio button options: "Process" (which is selected), "Process menu", and "Configuration menu".

如果选择 “Start with Process”，系统在标准模式下直接进入处理模式。这意味着检查程序在电脑上运行。不可以在远程模式下选择 “Process”，因为处理模式运行于智能相机而不是在电脑上。



A screenshot of a dialog box titled "Start with". It contains three radio button options: "Process", "Process menu" (which is selected), and "Configuration menu".

如果选择 “Start with Process menu”，系统将进入有三个按钮的标准模式或有两个按钮的远程模式。



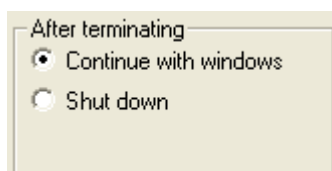
A screenshot of a dialog box titled "Start with". It contains three radio button options: "Process", "Process menu", and "Configuration menu" (which is selected).

如果选择 “Start with Configuration menu”，系统进入标准和用户配置界面的远程模式。

用户在远程模式下是不可以选择“Process”的，因为远程设备如 SmartBuilder 系统会在打开开关时自动进入 Process 模式。只有在标准模式下可选择 “Process”。因此软件要么在远程模式下以配置界面启动，要么立即启动 Process 模式。

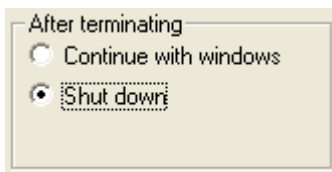
“Start with Configuration menu” 是一个用于示范的选项，或者你通常只在一台电脑上创建检查程序时选择。

对于主要在生产中使用的相机内的系统，当 SmartBuilder 软件终止时，可以选择退出整个系统或继续连接 Windows。通过以下参数设置 “After terminating”。默认设置是 “Continue with windows”。



A screenshot of a dialog box titled "After terminating". It contains two radio button options: "Continue with windows" (which is selected) and "Shut down".

软件停止后继续 Windows 系统。这是默认设置。



A screenshot of a dialog box titled "After terminating". It contains two radio button options: "Continue with windows" and "Shut down" (which is selected).

软件终止后关闭系统。

标准模式

软件启动时，PC 系统进入标准模式。此模式下，用户可以通过 “Check”和 “Configuration”转换至检查模式或配置模式。在检查模式下，可以开始或停止检查。

检查

	单击这个按钮开始之前配置的检查程序，同时 “STOP” 键被激活。
	单击这个按钮，回到有三个按键的标准模式。
	单击这个按钮停止检查程序。

	1 - 4 代表创建的检查程序。用户可以在不同检查程序间转换。
	打开 <i>Text terminal</i> 窗口
	打开 <i>Global tolerances</i> 窗口
	打开 <i>Operating parameters</i> 窗口
	打开 <i>Measured value statistic</i> 窗口：可显示、重置、打印或保存统计值。

配置

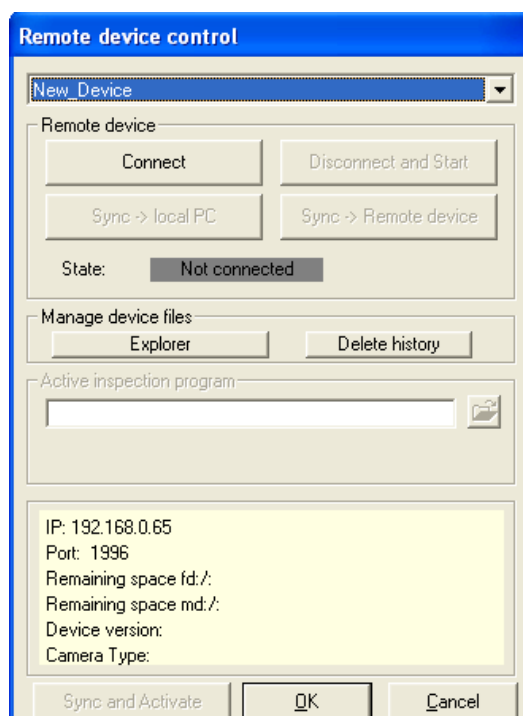
“Configuration”按钮可以启动配置界面。在配置界面中可以创建检查程序。在配置时检查模式被中止。

选项

“Options” 按钮可启动 *Operating parameters*。

远程模式

检查程序在智能相机中运行时，启动远程模式。如果相机与软件断开连接，智能相机自动进行检查模式。要进行配置模式，用户必须首先连接相机，命令 *Options* → *Remote device control*（如下图）。“Disconnect and Start”意思是断开相机与软件的连接，开始智能相机上的检验程序。

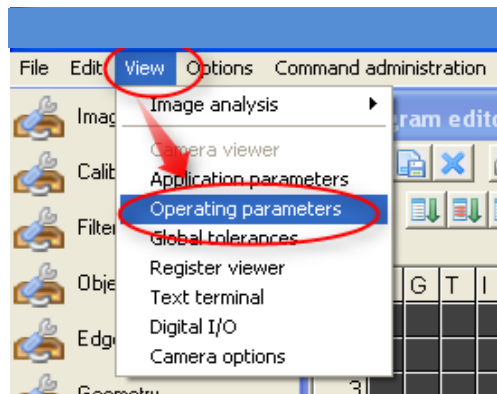


单一系统允许用户从检查模式转换到配置模式，或反向转换。如果系统有密码保护，进入配置模式前用户必须输入正确的密码。

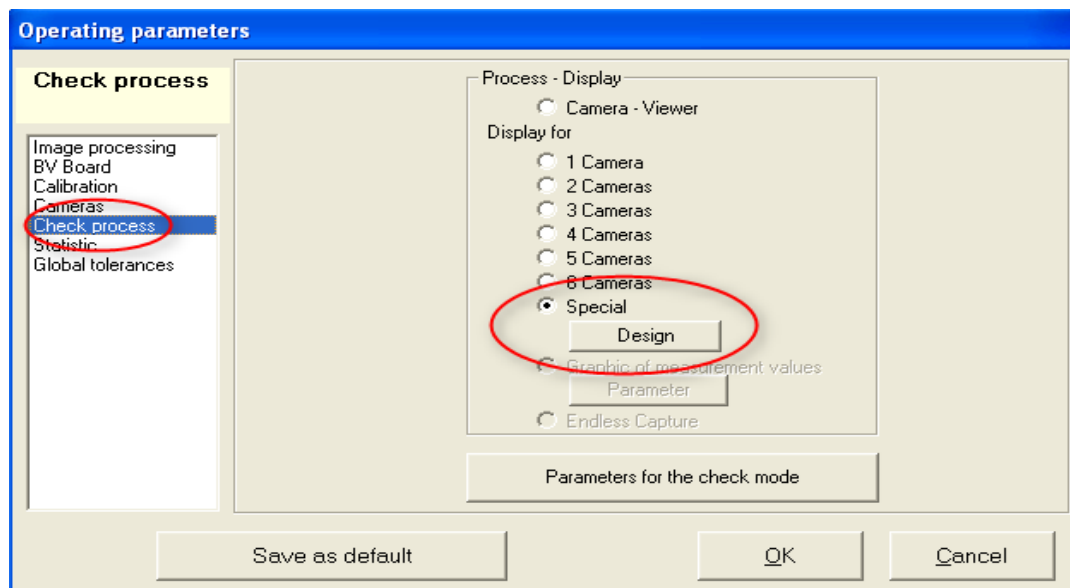
GUI 设计工具

通过 GUI 设计工具，用户可以设计自己的运行系统界面。不仅可以显示相机中的实时图像，还可以显示机器图像或零件，以及在背景中显示技术图纸。因此，用户可以全面地看到检查和检测参数。通过可选的统计工具，可以在用户界面上以图表形式清晰地显示数值。

要创建用户自定义界面，必须在配置界面中进行，因为根据检测程序不同设计也有不同。如果用户只看相机中的实时图像，只要提前设置相机取景器或选择1-6相机。但对于有图标或各种参数的用户自定义界面，就应该选择设计工具。



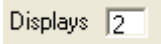
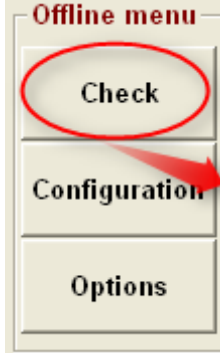
选择 view → operating parameters → check process 显示以下窗口：



根据相机数量，用户可为检测程序选择1-10个显示。选中 *Special*，再按“Design”按钮，就打开了显示的窗口，此窗口可在背景图像上配置和定位。

过程显示



	在这里可以设置显示的窗口数量。示例中显示了在背景的设备图像上有两个相机取景器。蓝色长方形窗口可以被修改。 可自定义修改窗口的位置和尺寸，因此用户可以创建最理想的显示效果。可在不同窗口上书写，因此用户可以将检测结果与相应的检测组件相匹配。	
	单击“Image”按钮，可以在设计器中显示背景图像。在“Resources”文件夹里有一些示例。JPG 图像都可加载，图像必须有正确的尺寸，因为图像不会被调整到背景的尺寸。	
		进入系统的自动模式 (<i>Check</i> → <i>Start</i>) 显示创建运行界面。 系统在检查模式下总是出现界面。按开始后，系统会马上进入检查模式。

下图中你可以看到，在背景机器图像上有两个窗口显示相机在检测中的实时图像。



2.2 启动远程系统

远程系统中按开始后有几种可能。用户可以进入配置模式（“Configure”按钮）。如果还没有选中相机，用户自动作为本地用户，即不进入远程系统。所有网络中的远程系统都可以显示出来。

如果用户选中其中一个，当用户转去配置模式时，系统会自动连接这个相机。如果远程设备有密码保护，在连接相机前会要求用户输入密码。

如果用户喜欢先进入本地系统再进入配置模式连接相机（通过 *Remote device control*），同样需要输入密码。所有有密码保护的远程系统都必须在身份验证后才能接入。

Additional Information	Standard Information	Active Cameras
	Camera Name: no camera selected Camera Type: no camera selected Colour Type: no camera selected Camera IP: no camera selected Connect-Port: no camera selected Control-Port: no camera selected Software Version: no camera selected OS Version: no camera selected Info Prot. Version: no camera selected License Ok: no camera selected License vs. Program: no camera selected Current Program: no camera selected Program Status: no camera selected Good Counter: no camera selected Bad Counter: no camera selected Cyclus Time: no camera selected	Local system
Update Selected Camera less >>	Control Port 5952	less >> Scan Network

Camera Direct Control	
Camera Name: No Camera Selected	Start Stop Stop Immediatly Restart Immediatly
IP address: 192 168 0 69	Switch to standard.ckp Reset Counters Camera Soft Reset
Control-Port: 5952	Simulate Trigger

2.3 检查过程

系统根据设置的运行参数直接进行检查过程。

你可以通过 **View → application parameters → check process** 改变这些参数。

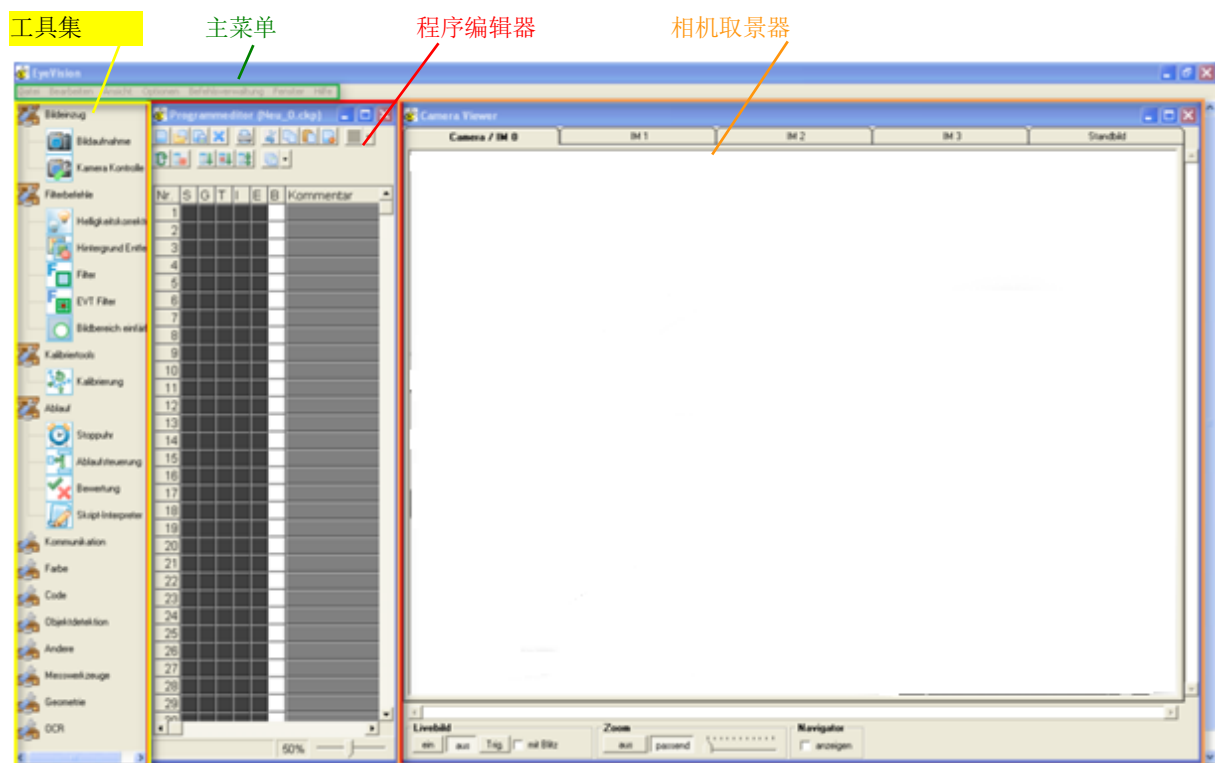
在这里你可以打开一个空白检测程序，上次检测的程度或已经保存的程序。

选择这些参数号，点击确定按钮并关闭 **SmartBuilder**（请完全关闭软件）此时如果重新启动软件，你选择的程序将自动启动。

2.4 配置界面

配置界面是主要的工作界面。界面上不同的组件可以创建检查程序，同时可以控制硬件。

下图是配置界面的说明：



后面章节将详细介绍配置界面的不同组件。

2.5 主菜单



通过主菜单的不同命令可以控制 SmartBuilder 软件

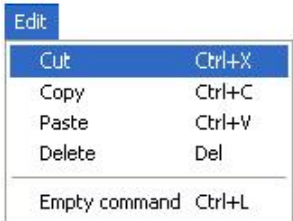
文件（File）

	在 File 中执行保存和加载程序。也可以创建一个支持文件，包括所有专业支持必需的软件信息。
--	--

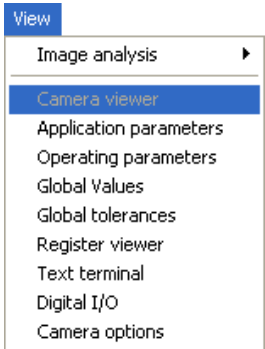
新程序（New Program）		在程序编辑器中创建一个空白程序。 在程序编辑器上方的任务栏上点击左侧的图标。 在程序编辑器上方的任务栏上点击左侧图标查看正在加载的程序，可以看到所有加载程序的概况。
打开程序（Open program）		点击这个菜单时，会打开一个新窗口（弹出）。打开的窗口显示当前打开的程序所在的目录。 你也可以通过任务栏上的图标（左侧）打开程序。程序编辑器中显示的就是当前激活的程序。
保存程序（Save program）		保存程序编辑器中当前的程序。你也可以点击任务栏上的图标（左侧）保存程序。
保存程序为（Save program as）		这个菜单项可以保存当前检查程序，用户需要输入一个程序名称。
打印程序（Print program）		检查程序被传送至打印机打印。 点击任务栏上的图标（左侧）打印当前的检查程序。
关闭程序（Close program）		关闭当前激活的检查程序，删除程序编辑器中的程序。 你也可以通过点击任务栏上的图标（左侧）进行这个操作。

创建支持文件（Create support file）		通过这个菜单项将所有的支持文件打包保存为存档文件。这样，可以很容易地重建应用程序。保存当前检查程序的同时也保存了校准文件和所有配置文件。存档文件可转化成支持，于是可以解决问题。
加载图像（Load image）		通过这个菜单可以选择一个图片文件，加载到相机取景器的图片内存并随后显示在屏幕上。
保存图像（Save image）		保存相机取景器中的图像，用户要输入一个文件名。
打印图像（Print image）		传送相机取景器中的图片内存图像至打印机。
图像到剪贴板（Image to clipboard）		存储相机取景器中的图像至剪贴板。
程序1-5（Program 1 – 5）		这里显示用户最近使用过的程序。单击任何一个程序将显示出对应的程序和图像。
退出配置模式（Exit configuration mode）		退出配置模式但不退出 SmartBuilder 软件。
退出（Quit）		关闭 SmartBuilder 软件。

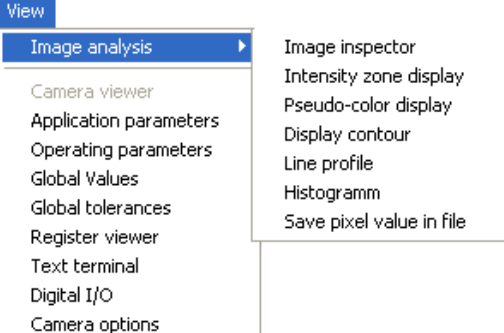
编辑（Edit）

		菜单项 Edit 允许用户对程序编辑器中不同的命令进行操作。
剪切（Cut）		这个命令可以剪切当前的命令或选中的区域。剪切的内容存储于剪贴板。后面的命令上移到被剪切的命令位置。
复制（Copy）		从剪贴板中复制当前命令或选中区域。
粘贴（Paste）		粘贴剪贴板中的命令。
删除（Delete）		在检查程序中删除激活的命令或标记的命令，其他命令移动到被删除命令的位置。
空白命令（Empty command）		增加一个空白命令行。

查看（View）

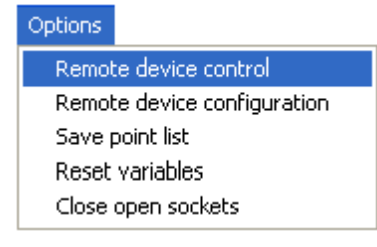
	这个菜单包含 SmartBuilder 所有的参数设置和分析功能。
图像分析（Image analysis）	子菜单包括各种评估图像的工具。这些工具帮助用户更好地了解图像。
相机取景器（Camera Viewer）	如果相机取景器被关闭，可以用这个命令激活。否则这个命令显示为灰色。
应用参数（Application parameters）	SmartBuilder 软件的具体应用参数。
运行参数（Operating parameters）	SmartBuilder 软件的具体操作参数。
全局容差（Global tolerances）	打开一个窗口设置全局容差。详细描述见对应的章节。
全局值（Global values）	打开一个窗口设置全局值。在“Global values”里设置位置值（Pick-up Value）变量。如果你进入 EyeVision.ini，可以从以下入口 <i>[EditGlobValues]</i> <i>Visible=1</i> 使用“Global values”。
寄存器视窗（Register viewer）	显示寄存器的值。显示的是缓冲值。在寄存器缓存中保存评价结果。
文本终端（Text terminal）	显示所有激活命令的文本输出。数字（如0001）表示命令的行号，数字后面是命令或功能（如 重置点列表），最后看到命令成功与否。
数字输入/输出（Digital I/O）	显示所有可用的输入输出接口和当前信号状态。用户可以通过点击显示 LED 改变输出状态。
图案编辑器（Pattern editor）	产生一个相关的图案。
相机选项（Camera Options）	修改当前相机的相机参数。

图像分析（Image analysis）

	子菜单 image analysis (View)提供评估图像数据的许多工具。
---	--

图像检测（Image Inspector）	用户可以在相机取景器选择一个区域。所有区域的像素值显示在一个矩阵中。
强度区显示（Intensity zone display）	菜单会打开一个滑动控制低阈值和一个控制高阈值的窗口。可以通过滑块控制或直接选择颜色（点击按钮“Color ½”）。你可以直接在相机取景器中看到效果。
伪彩色显示（Pseudo-color display）	伪彩色显示可以用8种不同的亮度值显示图像。
显示轮廓（Display contour）	用这个选项可以通过滑杆调整亮度值。显示相机取景器中，物体边缘的阈值，这个阈值可被调整。
线档案（Line profile）	在相机取景器的图像上放一条线。沿这条线的图像的亮度值将以图形方式显示出来。
直方图（Histogram）	可以选择相机取景器中图像的一部分区域。直方图将显示这一选定区域的亮度值分布。
保存像素值到文件（Save pixel value in file）	保存选定的像素值到文件。这个保存的像素值可以被其他命令调用。

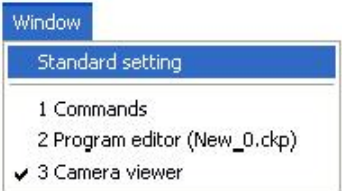
选项（Options）

	Options 菜单包括高级的硬件配置和一些高级设置（根据许可不同显示的选项不同）
远程设备控制（Remote device control）	通过这个菜单创建与远程设备的连接，目录要与编辑系统做比较。
远程设备配置（Remote device	在此安装新相机和新软件。

configuration)	
保存点列表 (Save point list)	保存程序编辑器中当前的点列表，以便其他命令调用。
重置变量 (Reset variables)	重置所有静态变量。
关闭打开的串行接口 (Close open sockets)	关闭检查程序打开的所有串行接口，释放端口。

Options 选项包括更多功能，后面会有详细的描述。

窗口 (Window)

	<i>Window</i> 菜单可以概览当前打开的窗口，亦可重建前面的设置。
标准设置 (Standard settings)	如果不同的窗口，如程序编辑器或相机取景器，被关闭或定位不好，将重建最初的配置。
命令 (Commands)	激活命令概览。
程序编辑器 (Program editor)	激活程序编辑器。
相机取景器 (Camera Viewer)	激活相机取景器。

帮助 (Help)

	在这里检索文档和软件版本的信息。
文档 (Documentation)	打开文档。
信息 (Info)	打开一个窗口显示重要的应用参数，如软件版本号。

2.6 程序编辑器（Program Editor）

程序编辑器是图像处理解决方案中最核心的部分。所以用户要熟悉编辑器中的不同功能。在创建新程序时或调试程序时都很重要。

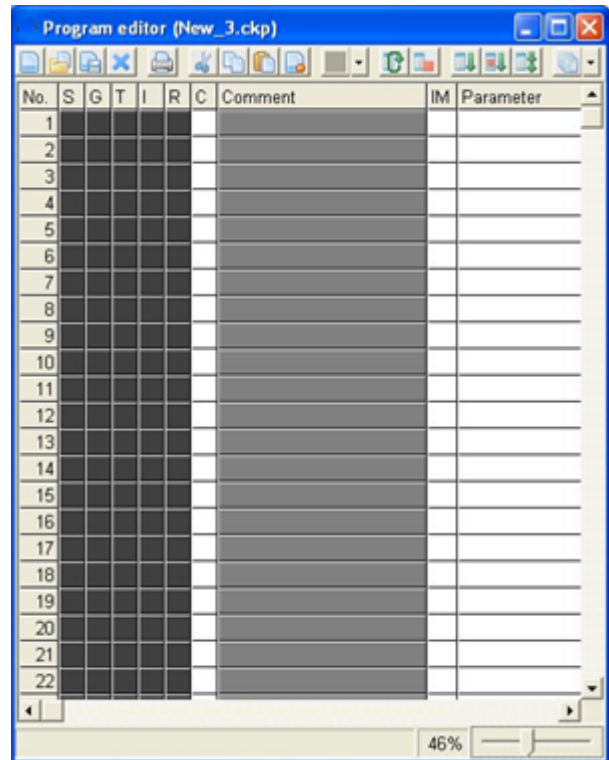
程序编辑器看起来像个列表：每个命令需要一行，可以粘贴或改变每一行的命令。

如果你要使用哪个命令，把这个命令从指令集中拖到 **B** 栏。

只有当你点击命令下面的白色空地时才会执行命令。

这一特性使你可以只执行一部分程序（对大程序很有用处）。

所有空白处下方的命令都不被执行。如果你在程序中设置了跳到下面的命令行（命令 *Interpreter control* → *Set mark*），只有当你在点击执行程序键的同时，按住 **CTRL** 键才会执行这个跳转。



快捷键

	创建一个新的空白程序。
	打开文件。用户可以打开一个程序。
	保存当前的检查程序。
	删除当前的检查程序。
	打印当前的检查程序。
	剪切程序编辑器中选定的命令。

	复制所有选定的命令。
	将缓存中的命令粘贴到当前的光标位置。
	删除选定的命令。
	循环运行检查程序。命令仍旧是在配置模式下运行的。
	停止运行的程序。
	完整执行检查程序（一次）。
	只执行选中的检查命令。
	一步一步执行检查程序——每检查一步，需要点击图标一次。
	概览程序编辑器中打开的所有程序。选中一个程序，该程序被激活。这个功能在有许多子程序时非常有用。可以在程序间自由切换。

关联菜单

程序编辑器在每一栏中单击右键都有更多附加功能。



单击右键打开每一栏的关联菜单。

No.	S	G	T	I	R	C	Comment	IM	Parameter
1									

No.

1

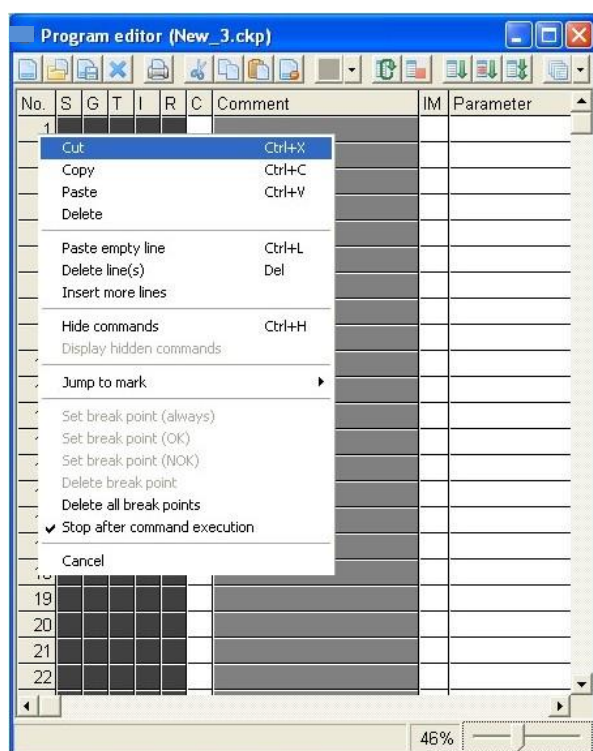
“No.” 意思是编号。表示命令行的编号。

S

“S” 表示停止执行。点击 S 栏表示停止命令。

G	“G”表示图形模式。G列标明何时命令显示为图形（如 OK，NOK 等）。
T	“T”表示文本模式。T列标明何时命令显示为文本（如 OK，NOK 等）。
I	“I”表示反转命令的结果。I列反转一个检查结果。
R	“R”表示命令的结果。R列显示指令执行的结果。如果指令执行成功显示绿灯，如果不成功显示红灯，如果是警告显示黄灯。
C	“C”表示命令。你从指令集中把命令拖出，如果你在 C 列放下命令，命令就粘贴到这里。
Comment 	在此列中可以对每个命令填写评论。双击评论列可以添加或修改评论。
IM 	“IM”表示图像内存。这里显示的数字表示使用哪个图像内存，命令作用于哪个图像内存。例如：如果显示为1，表示命令作用于图像内存1。你可以双击此列并输入新的数字（选择图像内存）。尽管你可以定义来源于图像内存0，目标图像内存是1。但实际上是“图像获取”命令从外部获取图像信息到图像内存，而不是真正地在图像内存上工作。
Parameter	此列显示配置时你对每个命令定义的参数。

序号列 (No. Column)



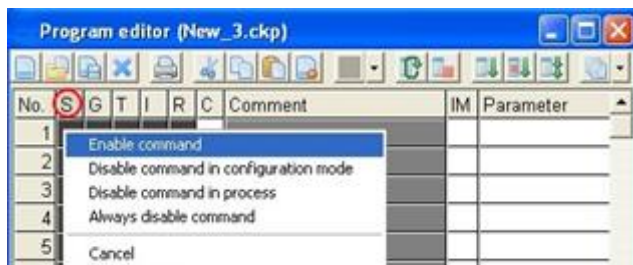
如果在序号列单击右键，将显示出左侧的菜单。



剪切 (Cut)	剪切当前命令或标记的命令。后面的命令上移。
复制 (Copy)	复制当前命令或标记的命令到缓存。
粘贴 (Paste)	在当前光标位置粘贴缓存中的命令。
删除 (Delete)	删除当前命令或标记的命令。
粘贴空白行 (Paste empty line)	在当前光标位置添加空白命令行。
删除行 (Delete line(s))	删除当前行或标记行。下面的所有行上移。
插入行 (Insert more lines)	用此命令增加更多的行。
隐藏命令 (Hide commands)	隐藏当前的命令或标记的命令。因此可以由用户清晰地安排检查程序。
显示隐藏的命令 (Display hidden commands)	显示所有隐藏的命令。
跳到标记 (Jump to)	此命令可以直接跳到一个标记。鼠标在这个命令上时，所有可用的标记显

mark)	示出来，通过鼠标左键选择并激活。
设置暂停点（永远）（Set break point (always)）	在指定的光标位置设置一个暂停点。程序执行到这一位置时停止。有暂停点的行标记有红色行号。
设置暂停点（OK）（Set break point (OK)）	设定有条件的暂停点。只有程序执行没有错误时到这一点才暂停。有暂停点的行标记有红色行号。
设置暂停点（NOK）（Set break point (NOK)）	设定有条件的暂停点。只有程序执行出现错误时到这一点才暂停。有暂停点的行标记有红色行号。必须选择“execute program”才能继续执行程序。
删除暂停点（Delete break point）	删除当前的暂停点。删除后此行号变为黑色。
删除所有暂停点（Delete all break points）	删除程序中的所有暂停点。所有的行号变成黑色。
执行命令后停止（Stop after command execution）	执行某个程序后停止检查程序。
取消（Cancel）	关闭当前的菜单，不保存新设置。

停止栏（Column Stop）



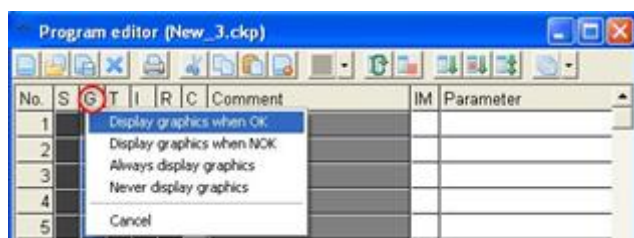
在一条命令行的 S 栏单击右键，可以控制此命令的状态。

单击左键可以禁用或启用命令。

启用命令（Enable command）	☐	此命令可被执行。
----------------------	--------------------------	----------

配置模式下禁用命令 (Disable command in configuration mode)		在配置模式下不执行这个命令。例如系统不会通过未连接的相机获取图像。
进程中禁用命令 (Disable command in process)		检查模式下不执行这个命令。例如，你可以配置，系统一旦进入检查模式，图像获取命令可以由相机的图像获取命令代替。
总是禁用命令 (Always disable command)		不执行这个命令。
取消 (Cancel)		关闭菜单。

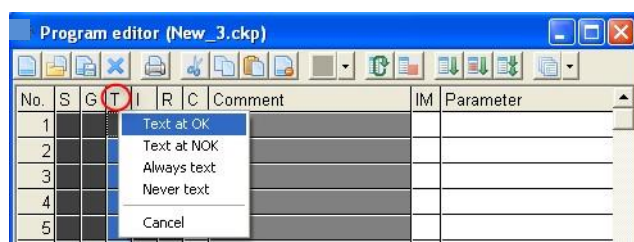
图形栏 (Column Graphic)



在 G 栏单击右键控制图形的显示。
单击左键可以依次激活不同的状态。

OK 时显示图像 (Display graphics when OK)		程序执行成功时显示图像。
NOK 时显示图像 (Display graphics when NOK)		程序执行不成功时显示图像。
始终显示图像 (Always display graphics)		始终显示图像。
从不显示图像 (Never display graphics)		始终不显示图像。
取消 (Cancel)		关闭菜单。

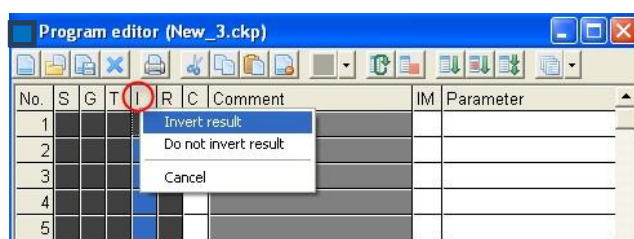
文本栏（Column Text）



在 T 栏单击右键控制输出文本的显示。
单击左键可以依次激活不同的状态。

OK 时文本（Text at OK）		程序执行成功时，在文本终端显示文本。
NOK 时文本（Text at NOK）		程序执行不成功时，在文本终端显示文本。
始终显示文本（Always text）		始终显示文本。
从不显示文本（Never text）		从不在文本终端显示文本。
取消（Cancel）		关闭菜单。

反转（Inversion）



在 I 栏单击右键反转命令的结果。
单击左键可以依次激活不同的状态。

反转结果（Invert result）		反转命令结果。
不反转结果（Do not invert result）		不反转命令结果。
取消（Cancel）		关闭菜单。

命令栏 (Command column)

在 B 栏单击右键可以粘贴命令或修改容差。

单击左键有两种可能：

1. 如果单击白色（空白）位置，会执行到此命令的所有命令。
2. 如果此行有命令，会打开一个配置窗口。

快速编辑 (Quick Edit)	打开命令的配置窗口（只有当此行有命令时）。
使用上次结果 (Use last result)	打开一个窗口，设定命令的容差。上次的测量值是默认的设定值。
参数列表 (Parameter list)	此行有命令时才会显示这一菜单。显示命令的参数列表。
显示实时菜单 (Show live menu)	如果系统连接了相机，这一菜单才会被激活，可在此显示或修改相机参数。
粘贴命令 (Paste command)	显示所有的命令。可选择命令粘贴到当前行。
取消 (Cancel)	关闭菜单。

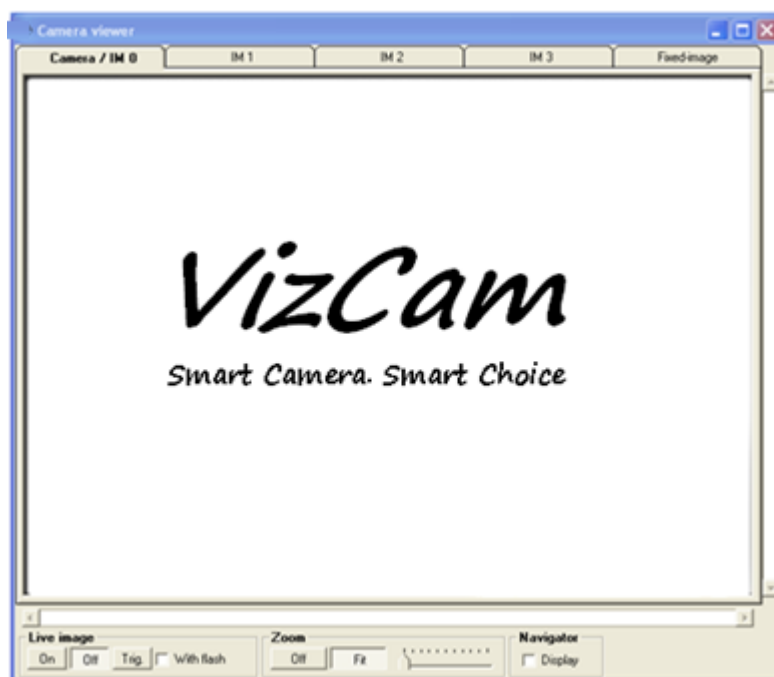
备注栏 (Comment Column)

在备注栏单击右键可以为命令增加备注。

双击左键可以输入新备注代替标准备注。

粘贴备注 (Paste memo)	如果在备注栏没有内容，你可以添加新内容。如果已经有备注，可以修改。已经有备注的，在备注栏右上角显示红色。
删除备注 (Delete memo)	删除现在的备注。备注栏右上角的红色消失。
取消 (Cancel)	取消备注。

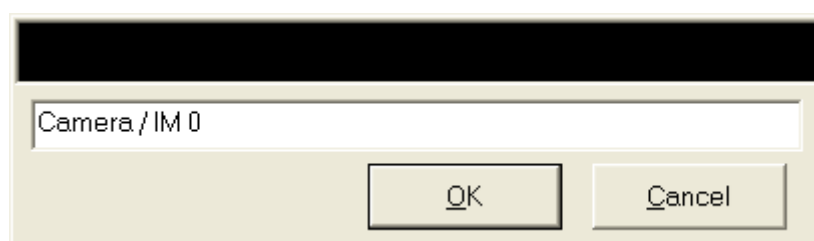
2.7 相机取景器（Camera Viewer）



相机取景器是 SmartBuilder 软件重要的一部分。相机取景器用来设置相机和定义检测任务。不同标签下显示相应图像内存的内容或实时图像。如果相机拍到某一图像，该图像显示在相机取景器中。如果图像内存没有连接相机，你看到的是图像内存的中间结果。如果你使用相关性模式，这个模式将加载到 IM1。

相机取景器功能

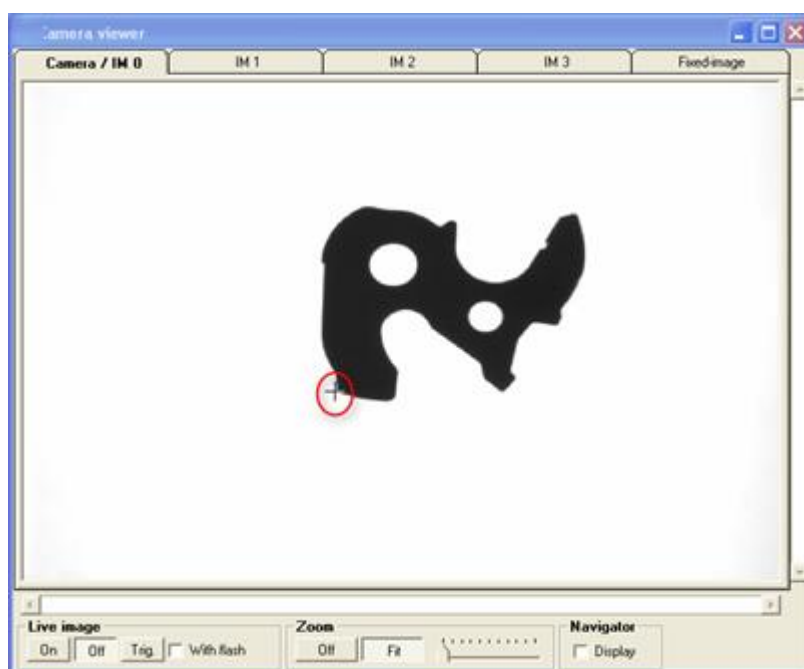
相机 / IM 0 ... IM 3 (Camera/IM 0... IM 3)	可以显示 n 个不同的图像或相机图像。相机取景器默认设置是显示四个图像内存。点击标签可以在不同的图像内存间切换。
修改相机名称	双击鼠标左键可以修改相机取景器标签的名称。
开启实时图像	显示当前选中的相机的实时图像。
关闭实时图像	关闭实时图像。
触发实时图像	通过触发激活实时图像。每个触发显示一个图片。
闪光实时图像	每次捕获图像有一个闪光输出。即图像采集激活闪光（闪光图像采集）。
缩放功能关闭	图像显示比例为1:1。
缩放以适应窗口	缩放图像尺寸以适应相机取景器。
显示导航	激活放大镜功能，缩放鼠标选中区域。



图像2.1: 改变相机名称（通过双击相机取景器标签）

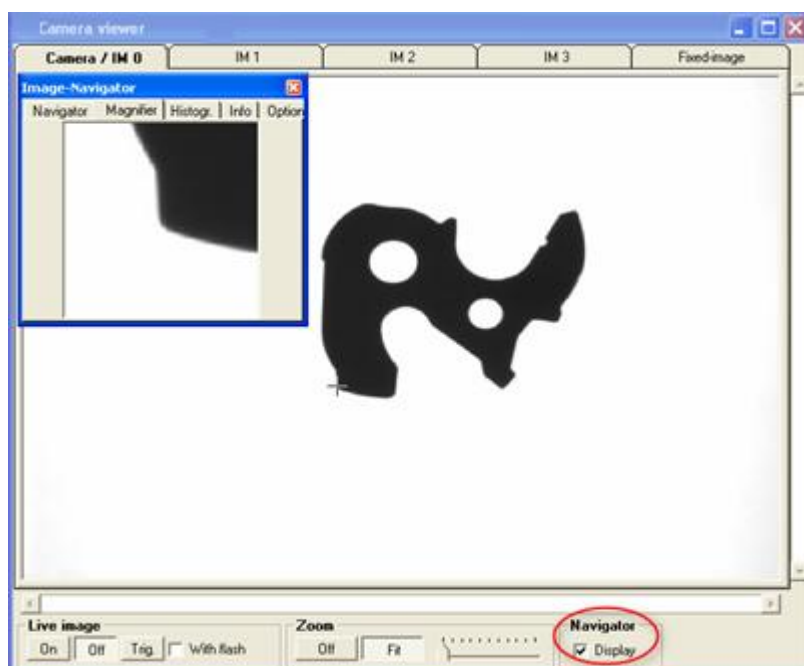
相机取景器光标

如果在相机取景器中移动光标，在 SmartBuilder 软件的任务栏上将显示出当前光标的坐标（红色部分）。括号中将显示光标位置的亮度值。



相机取景器导航

如果激活相机取景器导航，在相机取景器中移动光标，光标周围图像被放大显示。你可以决定光标周围的尺寸（通过图像导航“选项”标签 → x-section, y-section）。请修改 x-section 和 y-section 放大或减少周围尺寸。



2.8 命令集 (Instruction Set)

命令集包括用户激活的所有命令。

命令集的许可和目标系统或 PC 上运行的系统的许可有可能不同。所以建议通过 *command administration* → *activating command set* 调整命令。

通过数据交换功能可以延长系统。然后命令就可以用于运行的目标硬件系统。

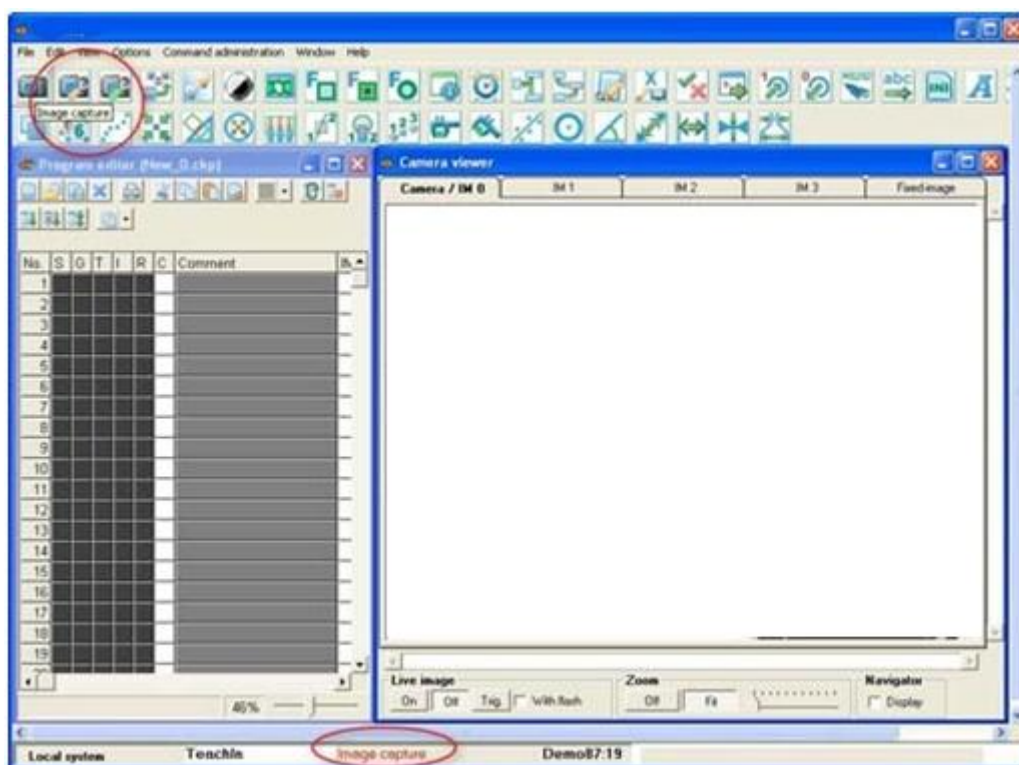
2.5UR 版本以后，用户可以变换命令集视图从经典视图到树状视图。此版本增加了许多新命令，根据检测任务命令，SmartBuilder 软件被分成许多类别。已经熟悉原始版本的用户可以继续使用原始版本命令集。

亦可以单独分类命令。

通过 *View* → *Application parameters* → *Configuration* → *Classical View* or *Tree View*，可以在两个工具箱中切换。

如果用户在经典视图下，用户在命令图标上移动时，命令名称将显示在任务栏上。如果光标在某个命令上停止，工具提示也会显示命令名称。

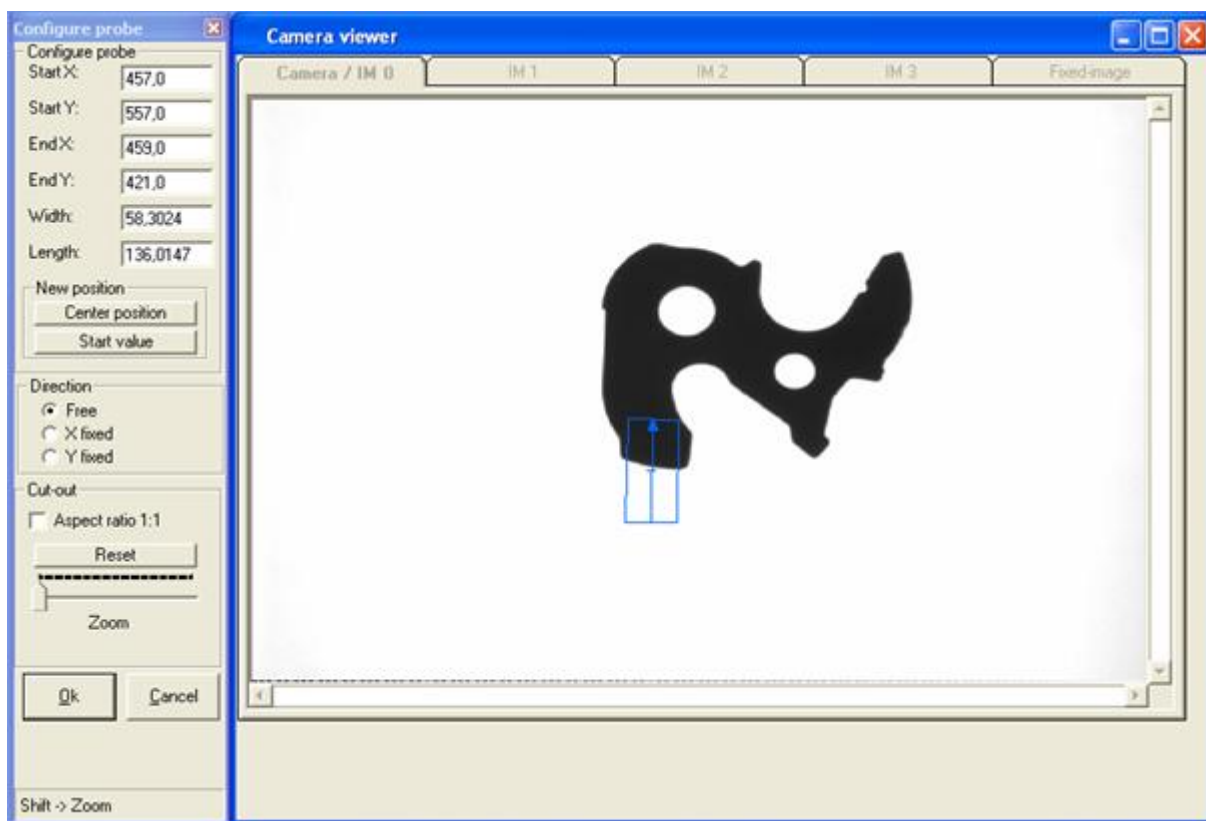
如果用户在经典视图下工作，提示工具会显示命令名称。



3 菜单

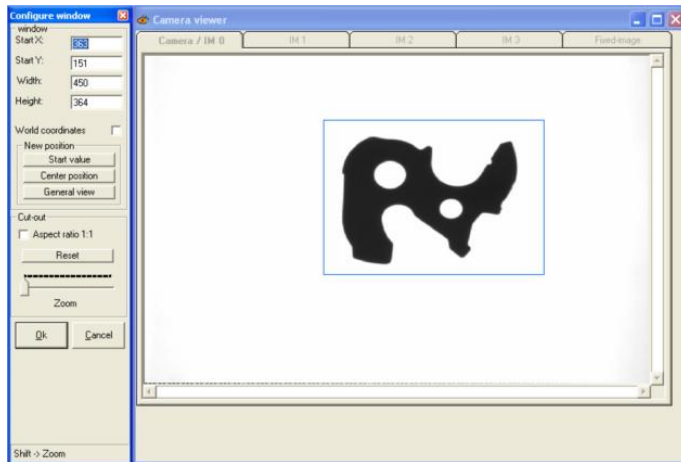
本章描述主菜单中的所有菜单。根据菜单的顺序进行介绍，所以从文件菜单开始。我们不介绍标准的操作系统的菜单，因为那些是由 Windows 操作系统去做的。

3.1 配置窗口和矩形探头



如果要配置一个探头，将有一个菜单显示在相机取景器旁边。在相机取景器中你可以看到蓝色探头。如果你没有看到探头，可以使用重置按钮（“Reset”按钮）将探头放回初始位置，之后再重新配置探头。如果用户已经知道要把探头放在哪个位置，可以直接输入坐标值。要放置探头，放大探头等，可以使用不同的光标形式。可以通过点击鼠标左键并按住左键修改探头。

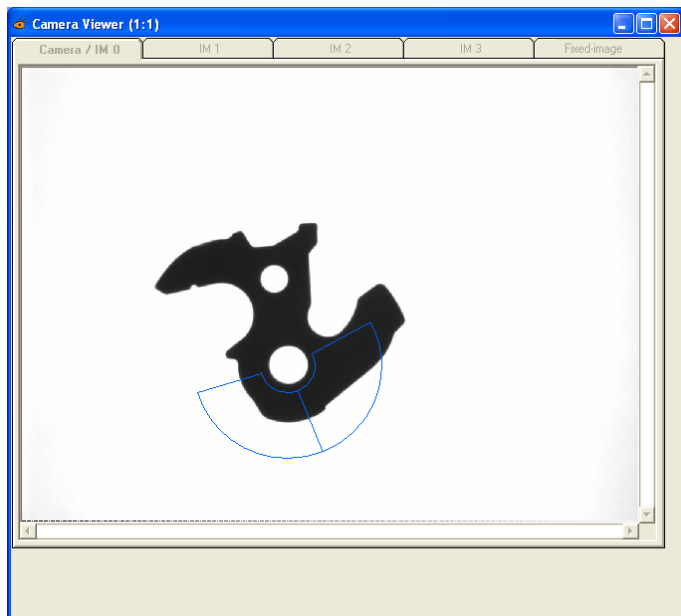
配置窗口（Configure window）



配置窗口与配置探头相似。

3.2 配置圆和圆形探头

配置圆形探头（Configure Circle Probe）



圆形探头会在圆或圆形的一部分中探测点。

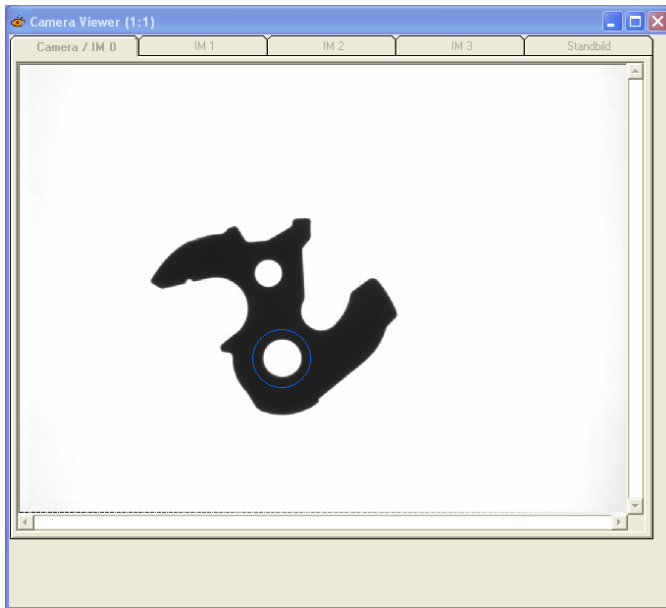
如果配置圆形探头，将有一个菜单显示在相机取景器旁边。

在相机取景器中可以看到蓝色圆形探头。如果你没有看到探头，可以使用重置按钮（“Reset”按钮）将探头放回初始位置，之后再重新配置探头。

如果用户已经知道要把探头放在哪个位置，可以直接输入坐标值。

要放置探头，放大探头等，可以使用不同的光标形式。可以通过点击鼠标左键并按住左键修改探头。

配置圆（Configure Circle）

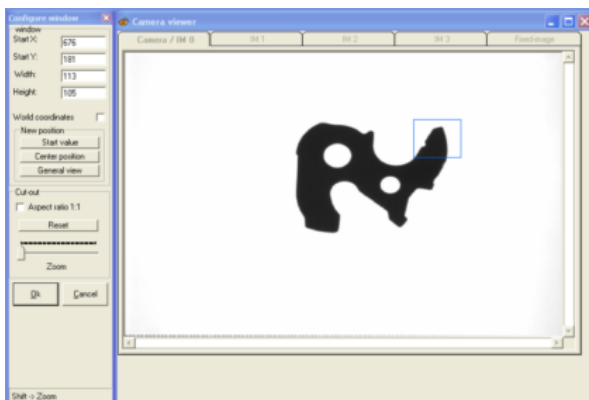


配置圆与配置圆形探头相似，只是你不可以配置扇形而只能是不同尺寸的圆形。它可以在圆形探头探测到的点上画一个圆。

3.3 图像分析（Image analysis）

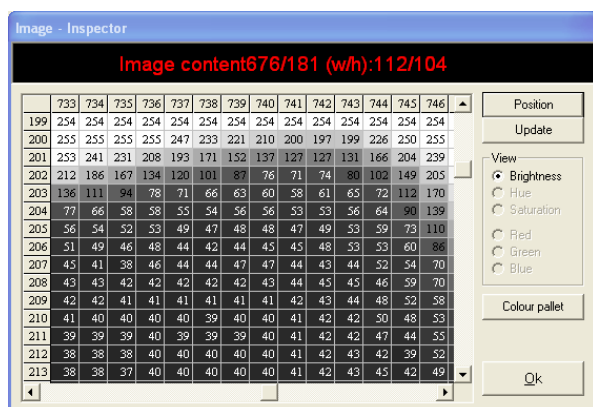
图像分析菜单可以进入相机光圈和灯光。也有特别的显示功能用于简化不同库的参数和基于 SmartBuilder 软件库的图形化编程。

图像检测（Image inspector）



定义搜索区域

选择检测图像后，在相机取景器旁显示一个配置窗口。现在要定义一个长方形区域，定义的区域显示在图像检测器中。

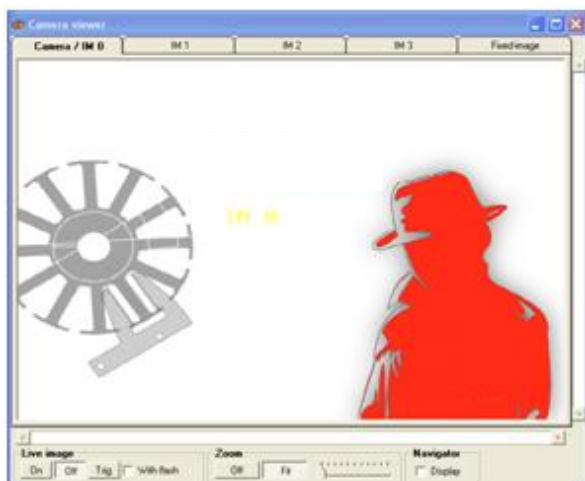


搜索范围的值

根据图像类型，显示颜色不同。像素值以数字形式显示。

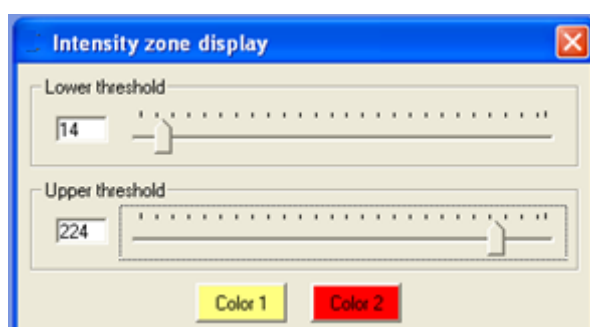
如果是灰色图像，每个位置都显示相应的灰色值。如果是彩色图像，你可以选择不同的颜色通道使用红，绿或蓝。相应地会显示亮度值，色调和饱和度。

强度区显示（Intensity zone display）



强度区显示可以用两种颜色显示灰度值。

这个工具非常有用。如果两种颜色被当作一种颜色调整其光线值，就相当于调整一个黑白图像。



确定阈值

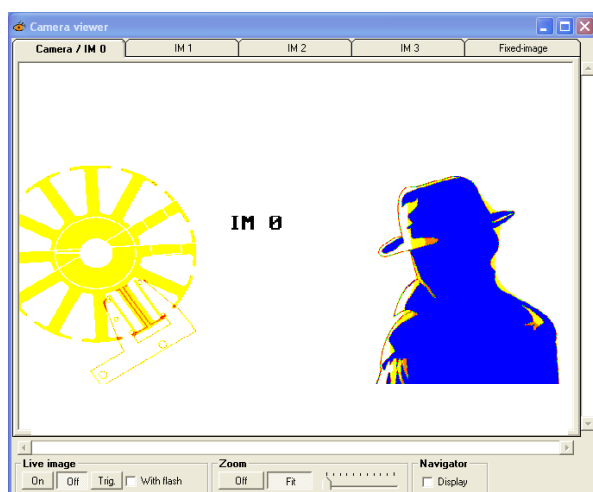
可以直接输入低阈值和高阈值，或者通过移动来确定（单点并按住鼠标左键，移动滑动块）。显示的值是现在相机取景器中的图像阈值。调整时可以直接在相机取景器中看到效果。



改变颜色

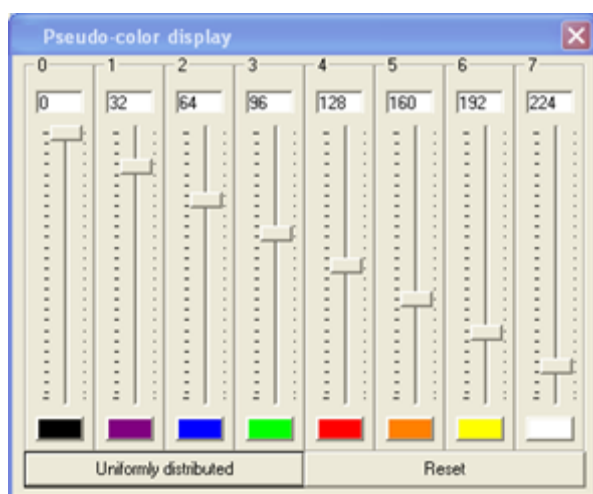
鼠标点击颜色1或颜色2可以打开颜色选择窗口。可以为高阈值和低阈值选择颜色。

伪彩色显示 (Pseudo-color display)



每个颜色代表了不同的灰度值间隔。可以选择七种不同的灰度值间隔。

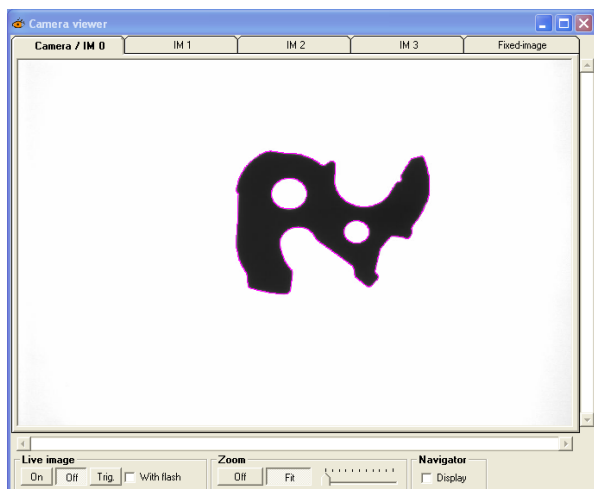
在一定的间隔内的点代表同一种颜色。例如：所有灰度值在0-32之间的点是黑色，所有灰度值在32-64之间的点是紫色等。所有设置都针对于相机取景器，所以可以直接看到结果。



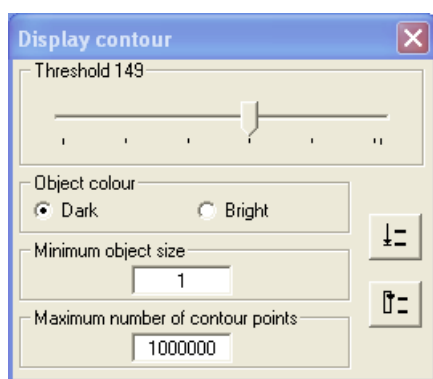
改变灰度值间隔

可以直接输入灰度值或通过滑块调整灰度值。
uniformly distributed 按钮确定灰度值，使8个间隔具备相同的尺寸。

显示轮廓（Display contour）



此工具用于确定应用的对象检测阈值，有两种可能去检查所需的更改。

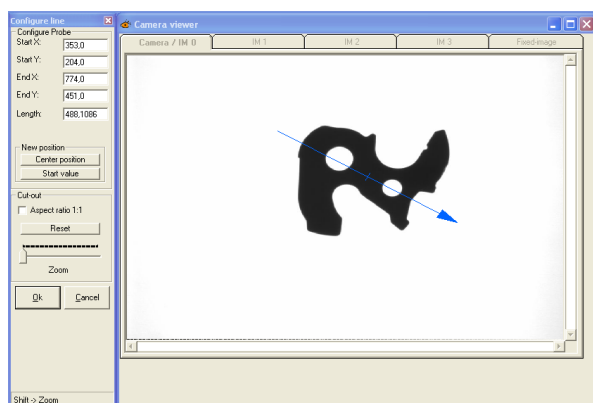


通过滑块可以很容易地改变边缘检测的阈值）（点击，按住鼠标左键并移动）。这个阈值用于边缘检测。移动滑块时，在相机取景器中，可以看到边缘的变化。

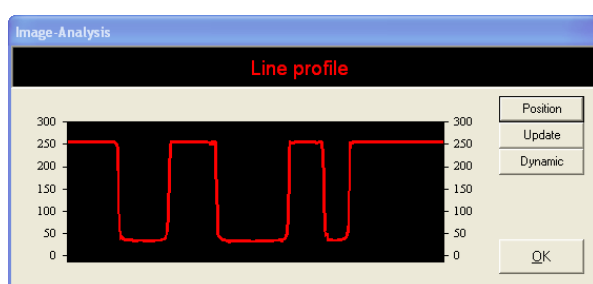
如果按“**One-time throughput**”按钮，所有调整的值传输至图片，并显示在相机取景器中。

如果按“**Permanent throughput**”按钮，所有调整的值直接显示在对象上。

线轮廓（Line profile）



在图中沿配置的线显示像素的灰度值。此工具非常适合关键的轮廓检查。此工具用于给边缘线确定一个合适的阈值。



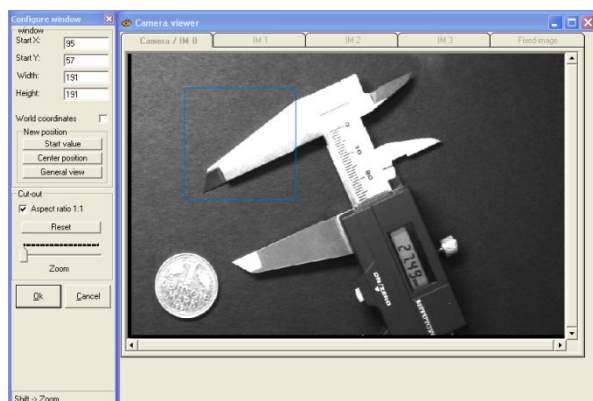
确定一条线

点击“Position”按钮打开一条直线的配置窗口。点击“Update”按钮重新计算图表。当相机取景器在现场模式下运行时是合理的。

动态显示

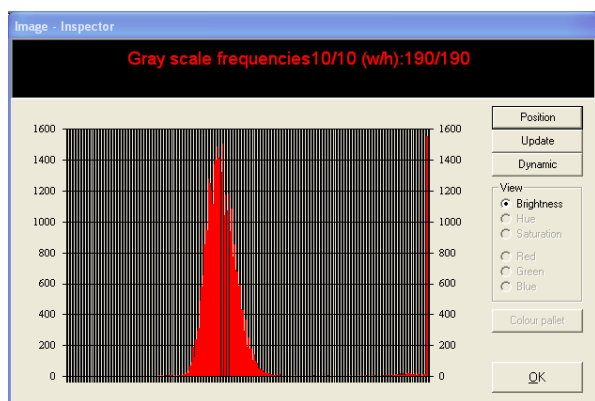
点击“Dynamic”按钮，线轮廓会不断更新。表示可以在相机取景器中直观地看到相机设置的改变效果。再次点击“Dynamic”按钮，又成为静态显示状态。

直方图（Histogram）



在直方图中显示像素的灰度值频率。此工具对调整相机（光圈）非常有帮助。

可在 X 轴读到灰度值（0-255），在 Y 轴读到频率。



配置窗口

点击 Position 按钮，可以配置灰度值的频率。点击 Update 按钮，重新计算图表。

动态显示

点击“Dynamic”按钮，会动态更新灰度值频率。表示可以在相机取景器中直观地看到相机设置的改变效果。再次点击“Dynamic”按钮，又成为静态显示状态。

保存像素值到文件（Save pixle value in file）

此工具可以保存图像的灰度值到文本文件。配置窗口打开后，可以输入文件名。

3.4 应用参数（Application Parameters）

应用参数对整个应用都有效，而运行参数只针对单一的检查程序。所以应用参数不保存于单一的程序，而是为多个程序设定。

需要激活硬件，否则没有可用的参数。

注：按下面的说明和图连接相机。如果没有连接相机，部分功能会不能用或未被激活。

自动程序控制（Automatic program control）

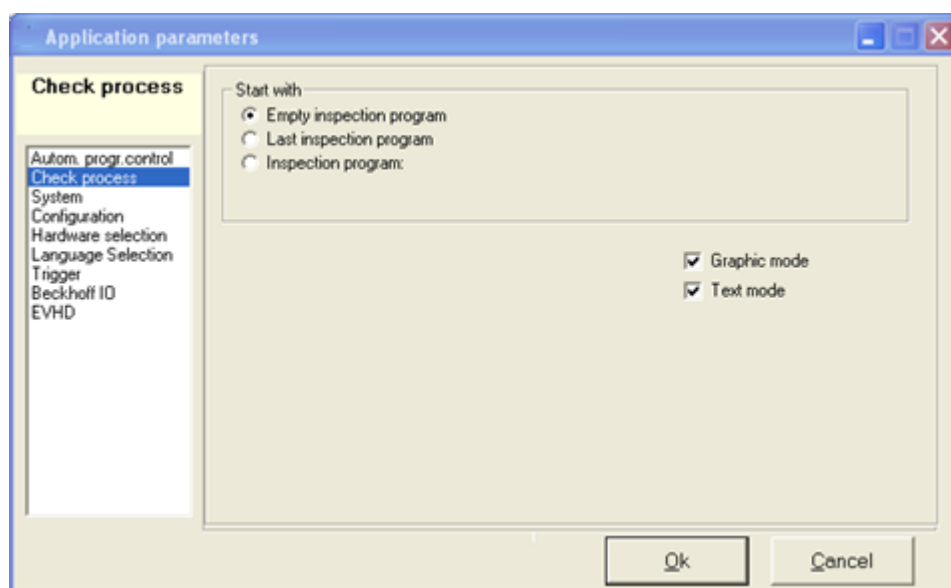


根据使用的硬件不同，有可能有或没有这个自动程序的功能。如果你用的是远程硬件，只有你与该硬件连接后，相应的选项才会出现。因为选项是与硬件相关联的。

如果系统未连接硬件，只能看到一个灰色的文字：自动程序改变（需要连接）（*automatic program change (connection required)*）。一旦硬件就绪，可通过数字输入/输出，串行接口或以太网激活“检查程序控制”。

数字 I/O	可通过数字输入/输出激活检查程序控制。
串行接口 (RS232)	可通过数字接口的电缆分配激活检查程序控制。在本章查看电缆的定义和检查程序的命名。
LAN (以太网)	根据 TCP 协议，相机可以获取检查程序改变的信息。
选项： Profibus, Interbus	如果连接了硬件，将显示此选项。

检查过程（Check process）



此标签下可设定检查过程的一般设置。这些设置对全部应用有效。针对程序特定的设置将在运行参数中设置。

空白检测程序（Empty inspection program）

程序开始于空白检测程序。

最后检测程序（Last inspection program）

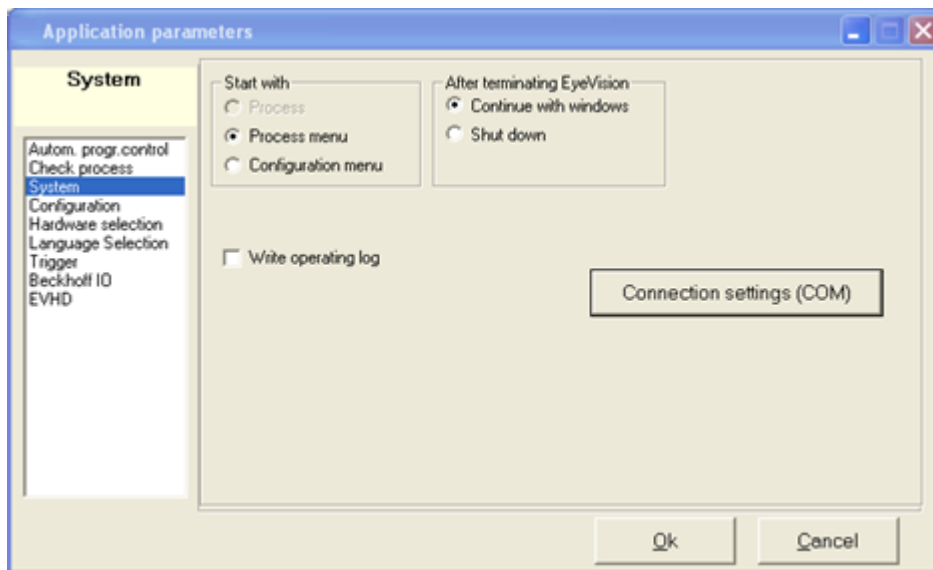
程序开始于最后一个检测程序。

检测程序（Inspection program）

选择一个要加载的检测程序。

注：确认请按“OK”。不能马上看到设置，因此，必须关闭 SmartBuilder 软件再开启，将会看到刚才你设定的空白程序，最后一个程序或者选择要加载的那个程序。

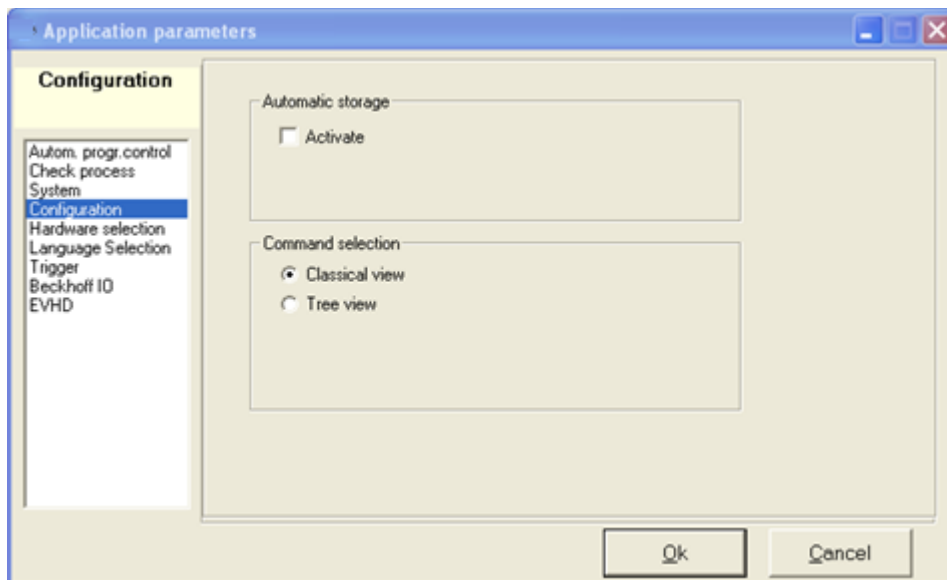
系统（System）



这里可以设定启动 SmartBuilder 软件时的模式。如果选择 *process*，SmartBuilder 会以检查模式开启检查程序。如果选择处理菜单（*process menu*），未开启程序但 SmartBuilder 启动后进入检查模式。如果选择配置菜单（*configuration menu*），启动 SmartBuilder 后会显示配置菜单。

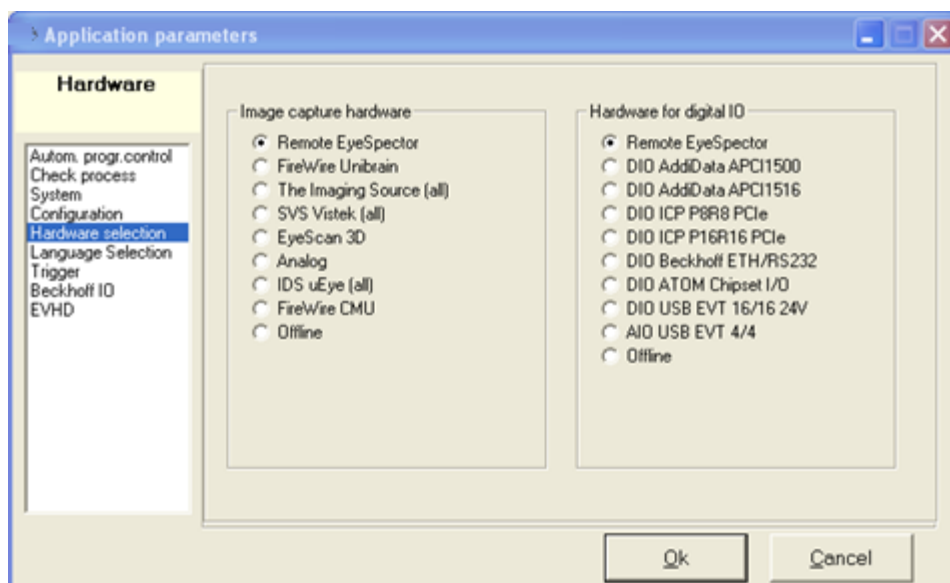
这里可以设置关闭 SmartBuilder 软件后的情况：要同时关闭 Windows 或者继续 Windows 操作。

配置（Configuration）



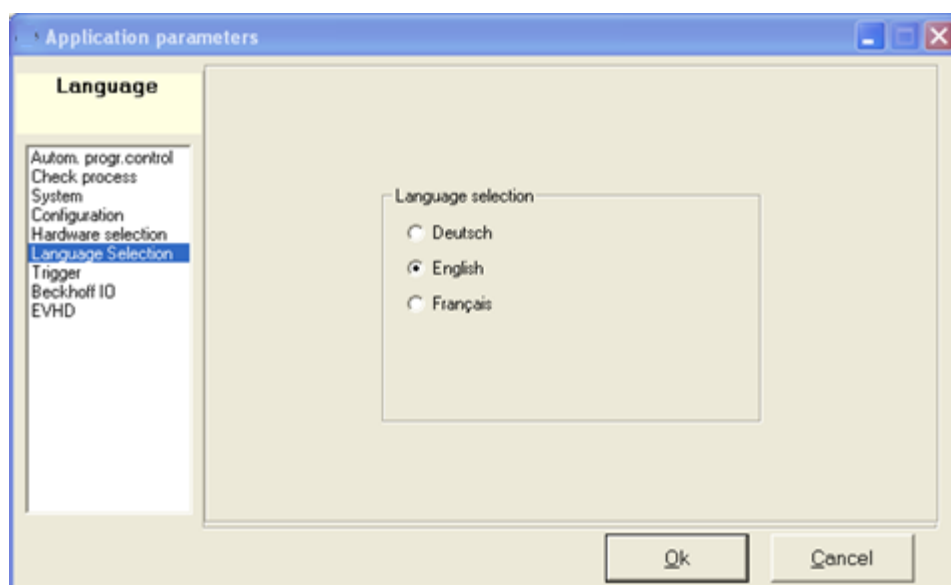
此选项用来确定自动保存当前检查程序的时间间隔，也可以改变命令的显示形式：经典模式或树状显示。（此设置在软件重启后生效）

选择硬件（Hardware selection）



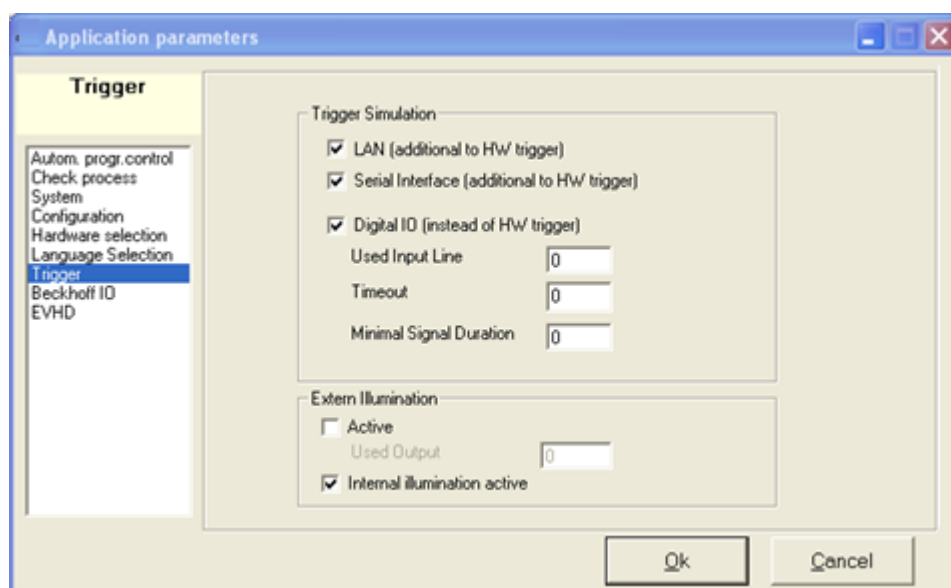
只要连接有硬件，可在此指定捕捉图像的硬件。

选择语言（Language selection）



SmartBuilder 软件有德语，英语和法语版，用户可以更换语言（重启后生效），更多语言版本需要另行安装。

触发 (Trigger)



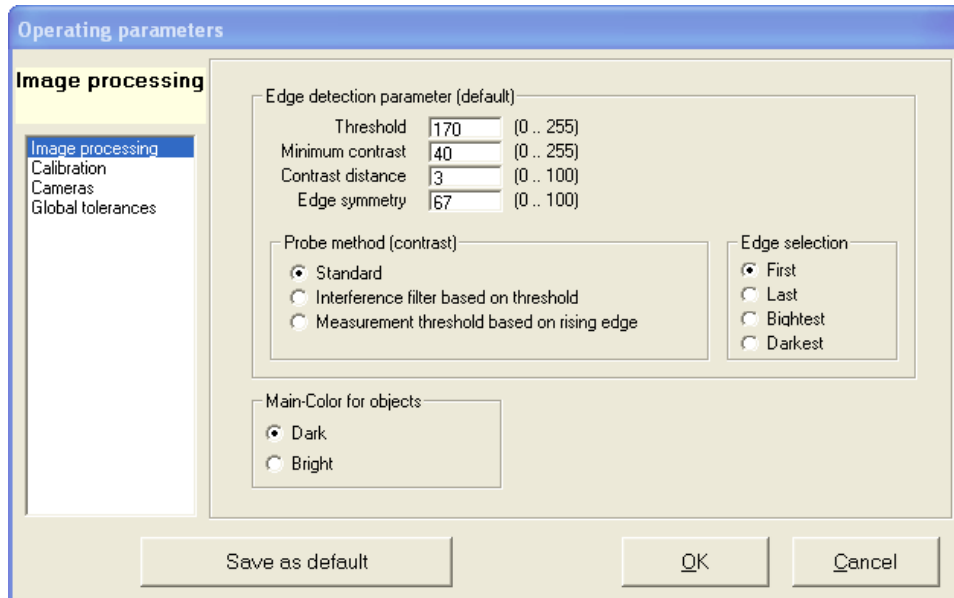
有不同的方式实现触发图像捕获。此菜单用来设置采用何种方式激活相机。如果某种方式不被硬件支持，将不可以选择。

3.5 运行参数（Operating parameters）

运行参数是为执行检查程序设定的参数。可以通过 *View → operating parameters* 或在离线模式下通过 *Options* 调用运行参数（因此必须退出配置模式）。

重要：运行参数只针对当前检查程序，也保存了当前检查程序中。

图像处理（Image processing）



边缘检测参数（默认）（Edge detection parameter (default)）

在此定义的参数对所有边缘检测命令有效（如点探头，矩形探头等）。因此，可以为同一种命令设置新的参数。

阈值参数决定了，检测边缘时，强度至少要超过的值。

最小对比度描述检测边缘时，强度至少要超过的值。

对比距离描述检测边缘时，最小对比度距离的值。

举例：

一个对象的强度是190（亮），背景强度30（黑）。请配置一个探头探测边缘。对比度是160。不过，由于阴影和模糊，过度不准确。对比距离是3像素，因为在之前设定的范围内，所以可以检测的边缘。边缘对称性的值描述边缘过渡的对称性，表示要检测边缘的百分比（这里是67%）。

探头方法（对比）（Probe method (cotrast)）

关于对比度，有三种不同的探测方法。标准方法（*Standard method*）指的是调整的边缘探测参数，不再进行任何验证。

基于阈值的干扰滤波器（*Interference filter based on threshold*）也是基于前面调整过的参数，并在强度交界处内过滤更多的干扰。

基于上升沿的阈值测量（*Measurement threshold based on rising edge*）方法能检测到其他方法不能检测到的边缘。因为虽然最小对比度在对比距离内，但对比又降下来，所以探测不到边缘。然而，这种方法能检测到对比的梯度从而探测到边缘。

对象主要颜色（*Main Color for objects*）选项可以定义对象是怎样映衬在背景上的，是更亮还是更黑。

要激活改变，点击 OK 按钮。新的运行参数被激活，并保存在当前程序中。

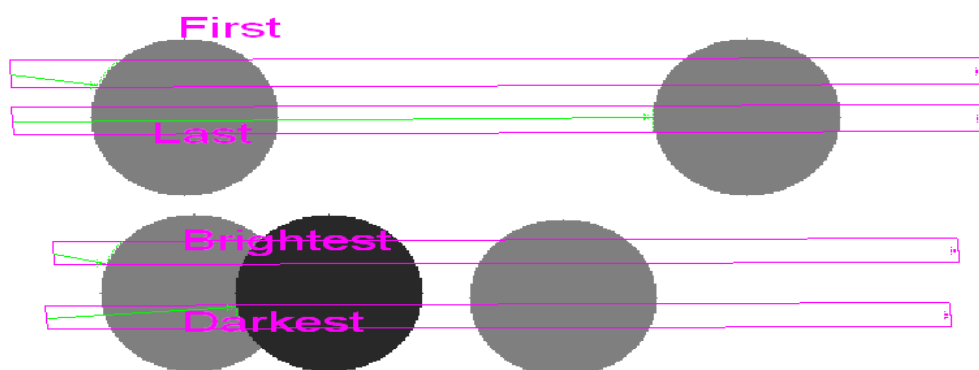
如果要当前设定适用于所有程序，点击保存为默认（"Save as default"）按钮，并在后面的保存当前设定为默认（"Save current settings as default"）的对话框中点击是（"Yes"）。

边缘选择（Edge selection）

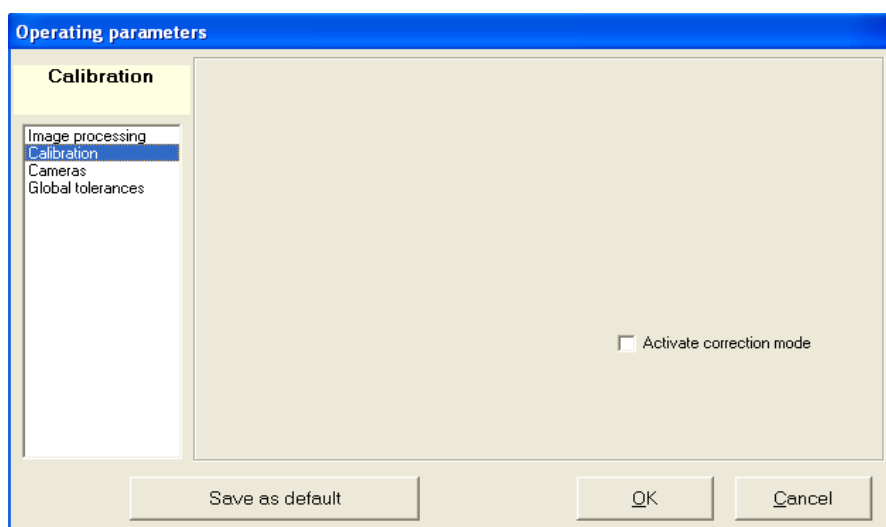
在边缘选择（*edge selection*）部分可以设置，如果有多个边缘，用户可以选择搜索方向上的第一个边，最后一个边，最亮的边或最黑的边。

下图显示了四个探头搜索边缘，示例从白到黑沿探头方向搜索边缘，于是从左到右：

第 一 个 (First)	表示在探头搜索方向上探测的第一个边。
最 后 一 个 (Last)	表示在探头搜索方向探测到的最后一个边。例中显示是从白到黑变化的边。
最 白 边 (Brightest)	在探头搜索方向上搜索到的最白的边缘过渡。
最 黑 边 (Darkest)	在探头搜索方向上搜索到的最黑的边缘过渡。

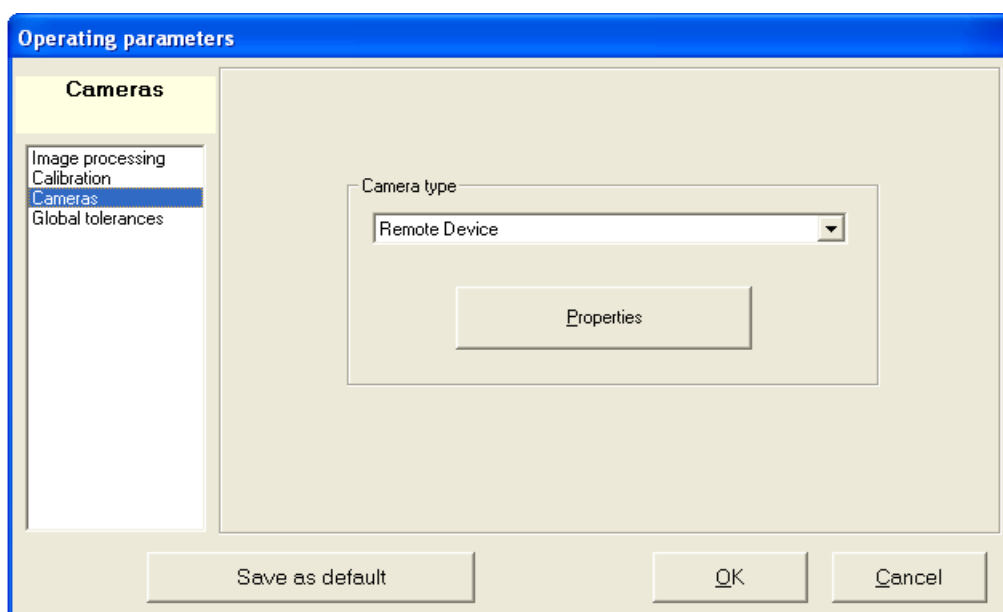


校准（Calibration）

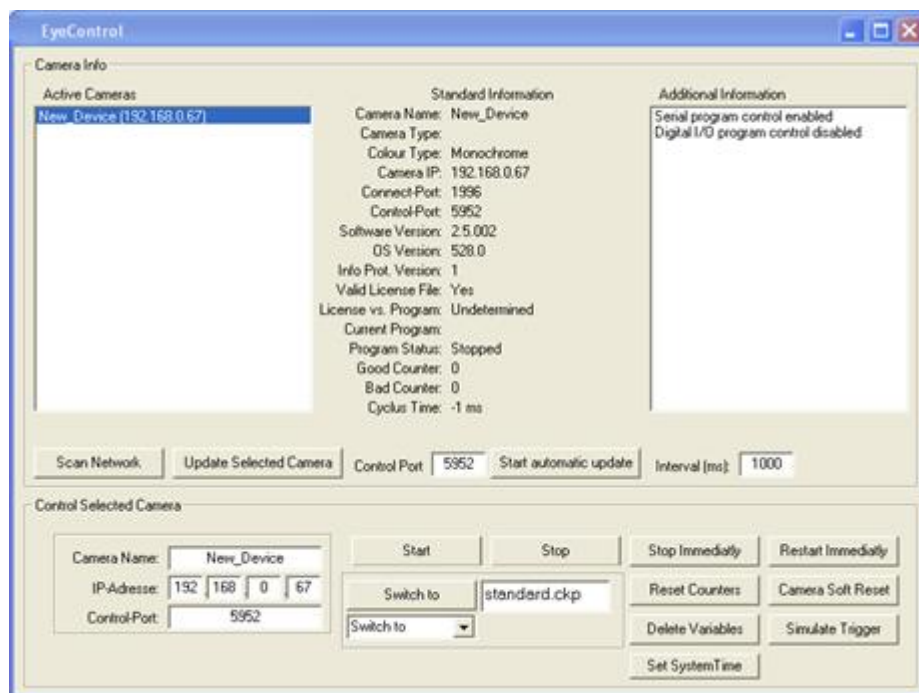


相机（Cameras）

选择一种相机，可以为一类相机设置统一的参数。对于智能相机系统，可以在属性（*properties*）里找到当前已连接至系统的可用的相机，并为相机设定参数，如当前的检查程序。



属性（Properties）



根据选择的相机，按属性（“Properties”）按钮进入相机属性。根据相机类型，要么没有附加的信息，要么打开 IEEE1394相机的寄存器概况，或在 SmartBuilder 窗口显示连接智能相机的信息及选项。

智能相机属性

扫描网络可以自动找到连接在 SmartBuilder 软件的所有智能相机。用鼠标在 *Active camera* 区域选择一个相机，可以激活相机并显示相机参数。用鼠标左键可以选择相机，使其成为当前激活的相机——用不同的按钮可以改变运行模式。

扫描网络（Scan network）

在可调整的 IP 地址范围内询问。在激活的相机（*Active Cameras*）中列出所有的相机。在列表中用鼠标左键选择相机，该相机即变成激活的相机。

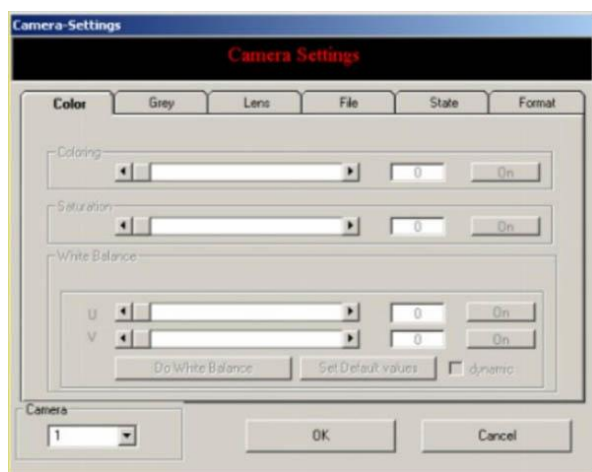
更新选中的相机（Update Selected Camera）

点击这个按钮，根据当前参数询问激活的相机。这些参数将被注册到信息区域中。

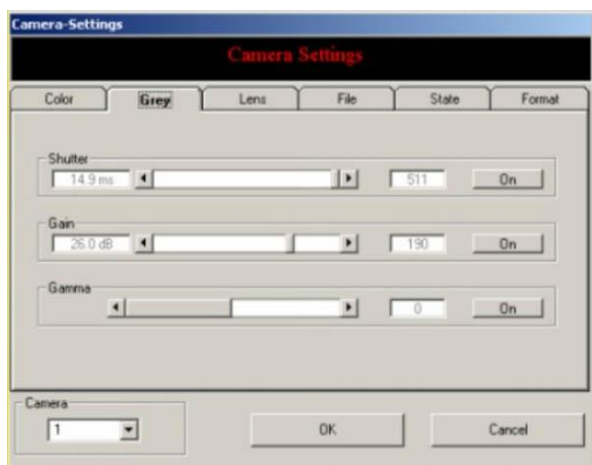
开始 (Start)	开始当前选中的检测程序。
停止 (Stop)	当前运行的检测程序在下次运行时停止。
立刻停止 (Stop Immediately)	当前运行的检测程序立刻停止，并且立即停止等待触发。
立刻重启动 (Restart Immediately)	中止当前运行的检测程序，并立刻重启动。
切换 (Switch to)	用这个按钮切换到指定的检测程序。
重置寄存器 (Reset Counter)	重置软件的好或坏的寄存器。
相机软启动 (Camera Soft Reset)	用这个按钮重新启动相机。相机的操作系统重新启动，之后 SmartBuilder 软件重新启动。
模拟触发 (Simulate Trigger)	用这个按钮释放一个软触发。
删除变量 (Delete Variables)	删除所有变量。
设置系统时间 (Set System Time)	在当前连接的相机中存入系统时间。

FireWire 相机参数

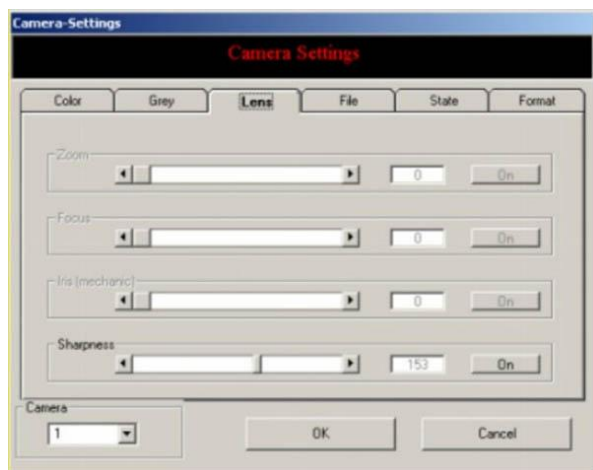
根据 FireWire 相机的特点，执行总线扫描将显示所有的 IEEE1394 相机。这些相机亦可用作图像的来源。此菜单提供多种选项用来调整相机，选项根据相机不同而不同。



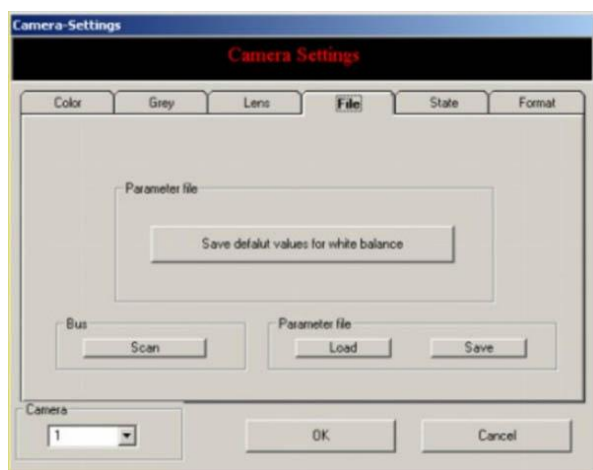
在此可改变相机的颜色和色彩饱和度。滑动滑块，操作键盘上的箭头键，或直接输入合适的值。按输入/输出，可以确定白平衡的调整值，可通过滑块或操纵按钮调整数值。白平衡亦可以动态完成。



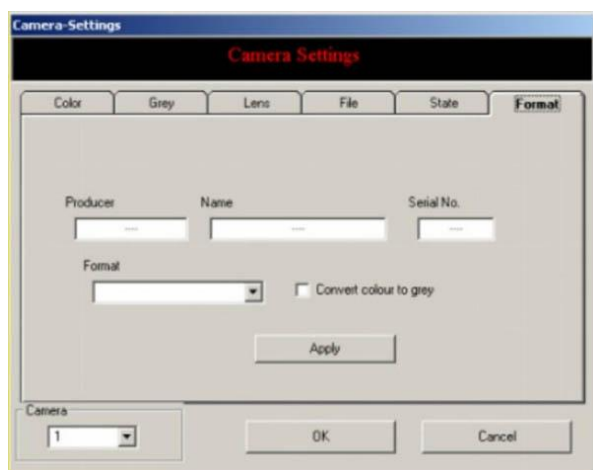
在此选项卡中可调整曝光时间。这个值适用于加强图像信号。



如果可以电子控制相机，则可以调整相机的变焦，聚焦和光圈。适用于图像锐度值。



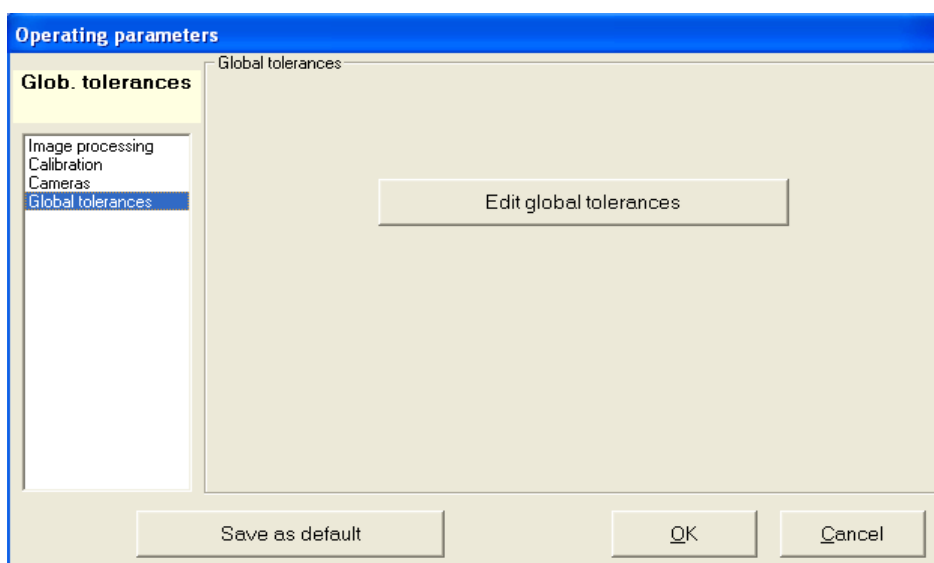
执行白平衡后可以通过“保存白平衡默认值”按钮保存值到文件，以后可以直接打开。另外，可以保存修改的参数或加载已经保存的调整参数。“扫描（Scan）”功能扫描总线并重置到标准值。



在此选项卡中显示使用的相机和卡的状态。

另外，也显示连接使用的是相机还是哪种图像格式。

全局容差（Global Tolerances）



点击“编辑全局容差（edit global tolerances）”，与直接点击“全局容差”选项得到的结果相同，因此我们直接讨论菜单选项。

3.6 全局值（Global Value）

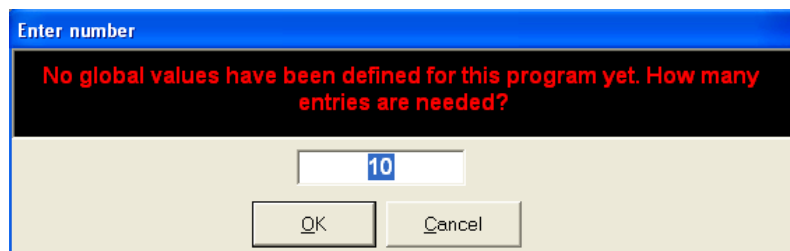
如果进入 EyeVision.ini，可输入“Global values”：

```
[EditGlobValues]
Visible=1
```

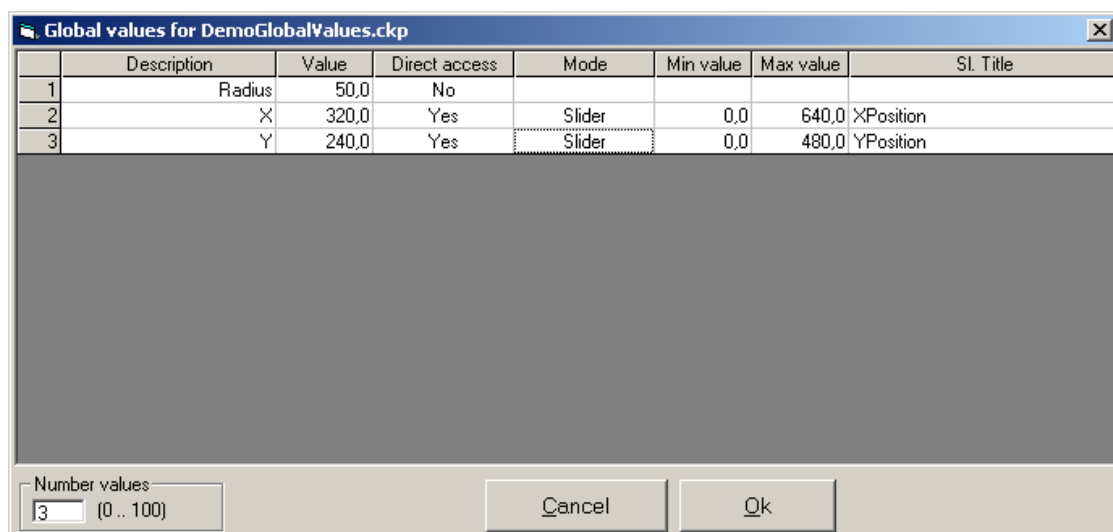
再次启动 SmartBuilder 软件，可以看到全局值（Global values）在以下菜单中，

查看→全局值（View → Global values）

点击查看→全局值（View → Global values），出现一个对话框定义全局值。使用全局值可以一次定义多个位置值（Pick-up value）。



可为每个检查程序定义全局值，并直接保存在程序文件中。变量只定义非静态的。这表示当程序开始时，变量将被设置为定义值或每次用户输入的值。



描述（Description） 输入变量的名称。

值（Value） 输入变量的初始值。

直接访问（Direct Access） “NO”：用户不能从外部修改值。
“YES”：用户可以在文本框或用滑块修改值。

模式 (Mode) 如果选择直接访问, 输入模式。

最小值 (Min value) 滑块最小值

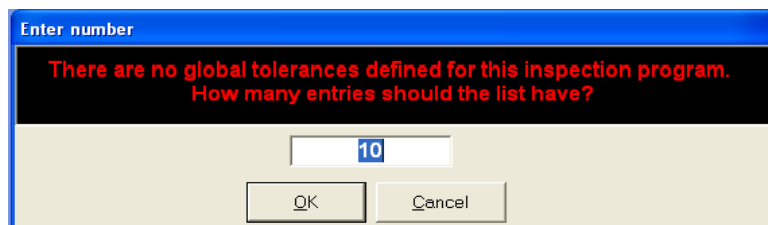
最大值 (Max value) 滑块最大值

S1标题 (SI title) 滑块标题 (可选)

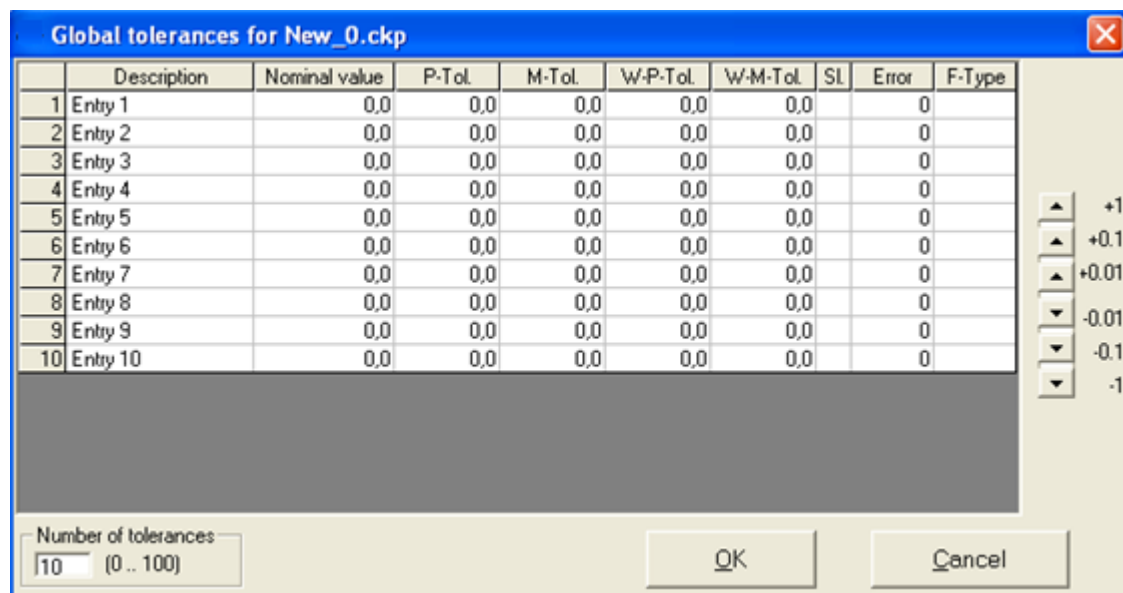
数字值 (Number values) 全局值数值, 重启后生效。

3.7 全局容差（Global Tolerances）

点击编辑全局容差（*edit global tolerances*）显示定义全局容差值的对话框，需要填写数值，窗口显示如下：



例如：上面显示的示例要定义十个不同的全局容差，点击 **OK** 后，如下图显示。有时，几个命令可以使用相同的容差值，于是可以用一个全局容差值连接多个命令。例如，要检查10个钻孔的尺寸精度，这些测量可绑定一个全局容差。

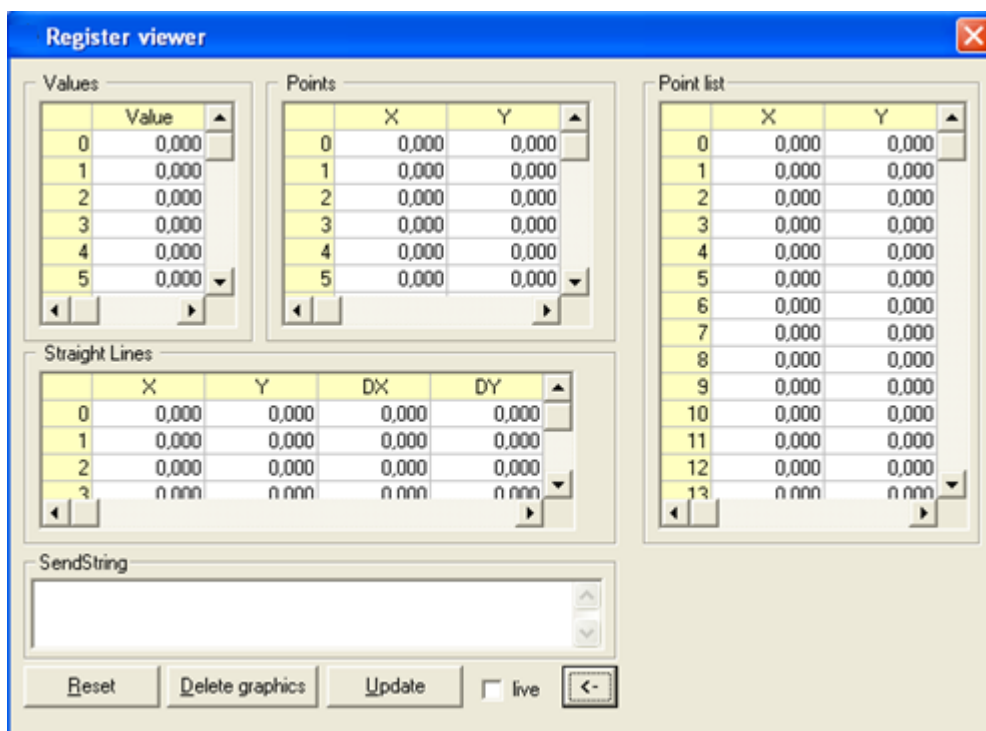


现在，可以定义每个输入及相应的值。

点击描述（**description**）可以重命名。可直接输入不同的容差值，或者通过窗口右边的键头按钮增加或减少对应位置的数值。

要增加或减少输入的值，需要填入新的容差值。所有改变将在下次打开此表时生效。

3.8 寄存器查看器（Register Viewer）



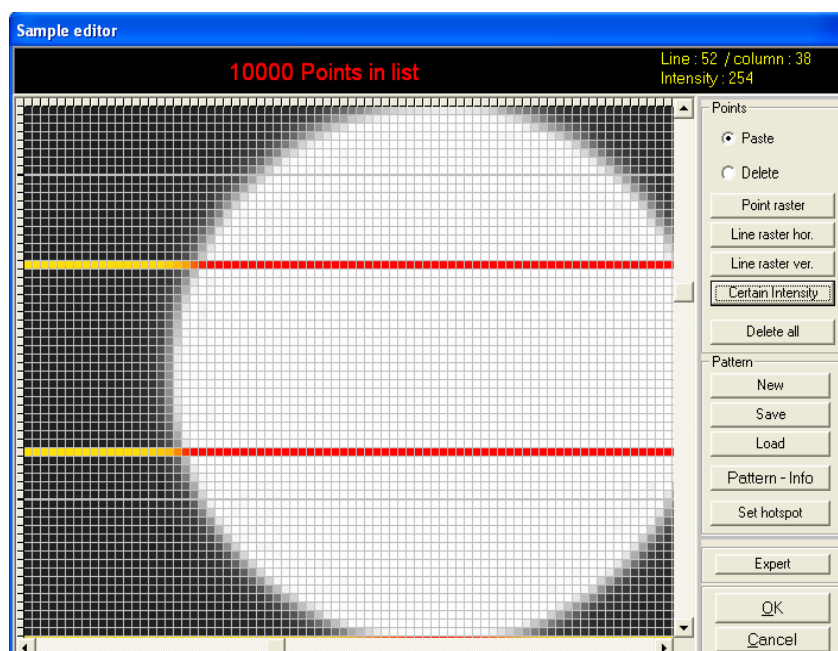
每个值，点和直线都存在对应的寄存器中。在寄存器视图中可以查看这些信息，更新信息并设置寄存器条目。按删除绘图（**delete graphics**）按钮，删除图像对象上的所有绘图。

重置（Reset）	寄存器视图中的所有值和对应的列表被重置。
删除绘图（Delete graphic）	删除显示的绘图。
更新（Update）	按更新按钮，在寄存器视图中显示内部结构值。
实时（Live）	按实时按钮，内部结构值将在程序处理过程中永久转移并显示。
	隐藏点列表。

3.9 图形编辑器（Pattern Editor）

在此视图中，可以创建一个相关图形。针对不同的可能性和程序，介绍如下。

通过相关命令进入图形编辑器。（点击 *point-brightness list*，在 *reference onto pattern Configuration* 下）。



图形（Pattern）

新图形（New）	用新图形（ <i>new pattern</i> ），选择当前图像的部分区域作为绘图的基础。用矩形探头配置一个区域，之后选中的部分将在图形编辑器中以像素值显示出来。要创建一个相关性的图形，可以选择输入对应的点，或者使用下面的方法选择点。
保存（Save）	保存当前图形及图形图像。
加载（Load）	加载图形及相应的图像。
信息（Info）	点图形信息（ <i>pattern-info</i> ）按钮，点的数量及平均强度将显示在上面的文本区域。
设置热点（Set hot	热点是成功进行轮廓检测后返回的点。所以，如果检测到轮廓，这个点就被放进点寄存器，并可以被其他命令调用。在图形中这些点标记为蓝色，默认位于选中图像

spot) 部分的中心。如果有特殊的边缘很重要，那里也会有热点。

点 (Points)

增加/删除 (add / delete) 用增加选项手动标记点，可通过鼠标左键增加一个像素点。如果按住鼠标不放，可以标记一条线或一个区域。要合理的标记探测的轮廓，要遵循一定的规则，详见示例。

点 栅 格 (Point raster) 在图像上创建点栅格，定义栅格的像素宽度。用此方法可快速找到轮廓并执行相关性。缺点是要评估许多点，所以栅格不够准确。

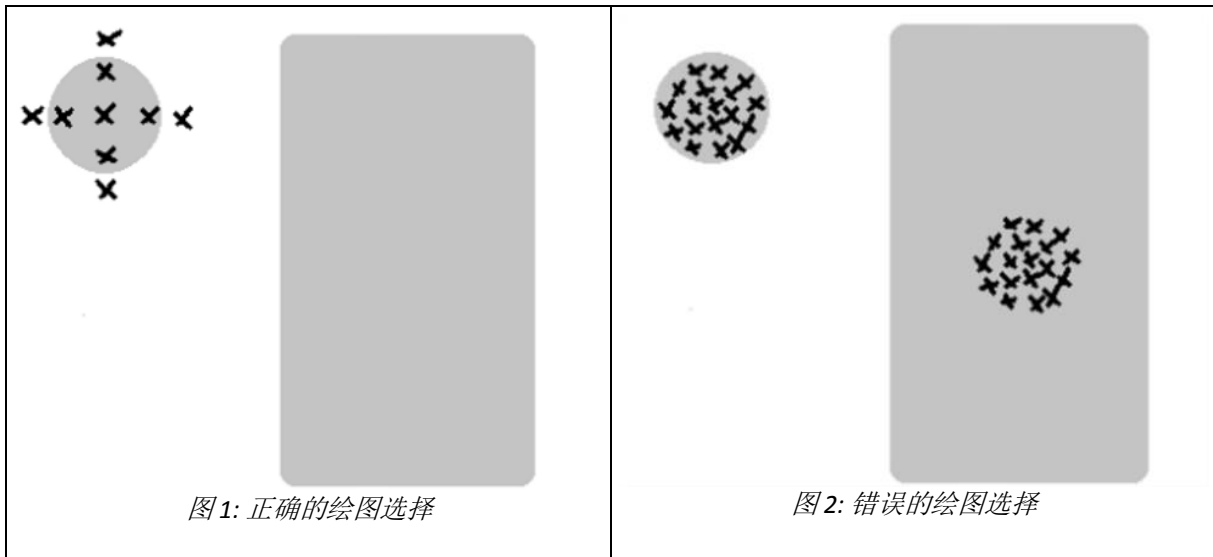
线 栅 格 水平/垂直 (Line raster hor./ ver.) 与点栅格类似，创建一个水平或垂直的线栅格，必须手动设置栅格宽度。

强度 (Certain intensity) 用此功能标记一定强度值的点。所以在定义理想强度范围的低阈值或高阈值之间标记所有符合要求的点。此功能可快速标记边缘（如字母），之后还必须标记几个轮廓周围的其他点。当然，你也可以减少轮廓的点以提高检查速度。

删除所有 (Delete all) 删除图形上的所有点。

创建相关性绘图（Pattern creation for the correlation）

这里有几个点是在绘图编辑器中创建绘图时，用以保证检测的平滑和可靠性的。



选择的点要沿边缘并始终保持每边成对出现。（图1）

通常用户习惯用同样的颜色标记区域而没有过渡（图2）。这样看起来很好，但是同样强度的区域在图像上会出现多次，与搜索的对象没有任何相关性，如图2。

对象像素的扩展与选择的光栅宽度之间的关系也同样重要。用5像素的光栅宽度搜索10*10像素的对象是没有意义的。理论上在绘图上移动选择的绘图点并认为其他位置不合格，可以想象像素与光栅宽度间的这种联系。



绘图越大，越空，光栅宽度越大。

但是，光栅宽度越大，在另一个位置找到绘图或完全找不到绘图的机率更大。

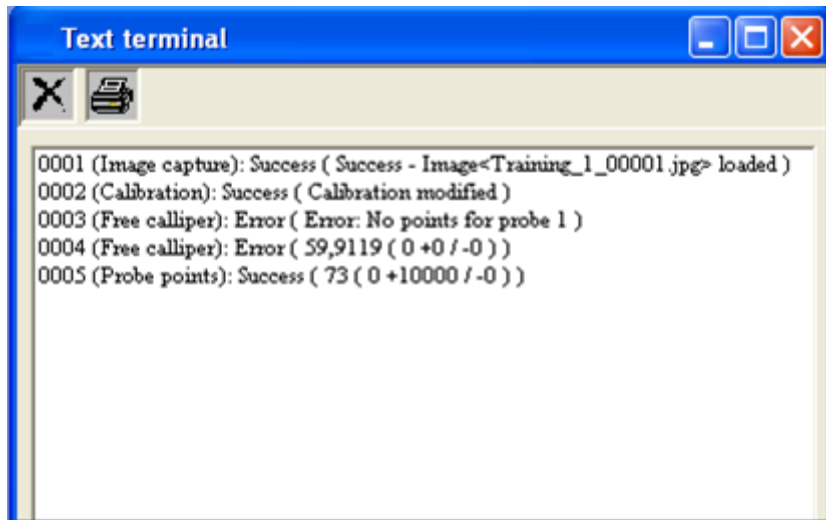
由于每个绘图不同，而且要对比的图像本身也非常重要，所以没有一定之规。只有通过经验的积累。在检查模式下可以选择性地增加亮度适应。但这种做法并不适用于所有图像检查。有时对于黑白图像，亮度适用的调整甚至带来不好的影响。但对于有许多不同灰度值的图像，这样做扩展了亮度范围，检查时仍然是安全的。不论是否使用亮度调整，对于一个检查安全性起决定性因素的仍旧是最大强度差参数和容差参数。

创建绘图技巧：

- 标记独立的点而不是一片区域
- 围绕对象也标记一些点以弥补背景干扰
- 用创建点栅格（*create point raster*）选项在视图上快速执行测试
- 最重要的是：用成对的点标记轮廓
- 标记有少量的点的绘图会比标记很多点的绘图运行速度快很多，因为栅格宽度较大。
- 因为速度的原因，对象越大，标记关键点就越重要。

3.10 文本终端（Text Terminal）

在此窗口中，检查程序的过程和结果被保存为文本形式，并可打印。包括检查程序输出的说明，当前评估的结果，和当前程序中每行的容差值。

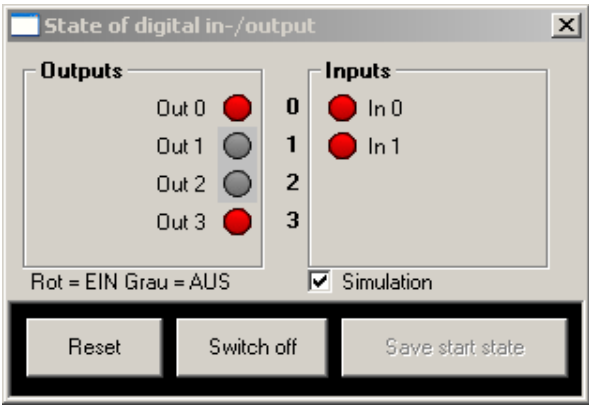


	删除文本终端中的输出。
	打印文本终端中的输出。

3.11 数字输入/输出（Digital I/O）

数字输入/输出显示可以快速查看系统嵌入的所有输入/输出接口。在 LED 中会显示实际的输入/输出和他们的状态。要设置一个特殊的输出，只要在相应地 LED 处点击鼠标左键。LED 现在在不同的状态下，输出线马上改变。

输入/输出表选项（I/O form options）

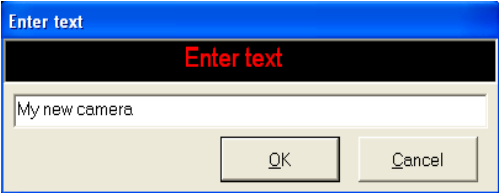


模 拟 (Simulation)	激活模拟选项后，模拟数字输入/输出。在模拟模式下，忽略连接的输入/输出硬件，通过点击 LED 设置输入。关闭数字输入/输出窗口结束模拟模式。
重置（Reset）	输入/输出被重新设置成默认状态。
关闭（Switch off）	关闭所有输出。
保 存 初 始 状 态 (Save start state)	选中此选项，保存初始状态。

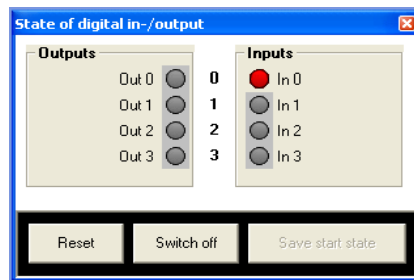
修改输入/输出名称（Changing I/O names）

	双击输入/输出名称可修改名称。新名称将用于输入/输出命令的检查程序。
--	------------------------------------

此例中，双击输出名称 Out 0，显示文本输入框，输入新的名称 BV RDY。



点击 OK 关闭文本输入框，新名称将显示在显示菜单中。确认后 BV RDY 就是输出0的新名称。



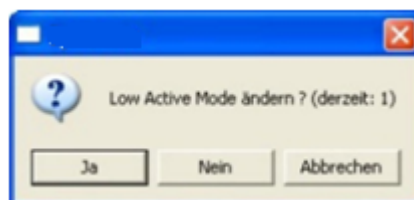
反转输入/输出条件（Invert In-/Output conditions）

在输入/输出表中，可以指定每个输入/输出的信号是否应反转的逻辑条件。

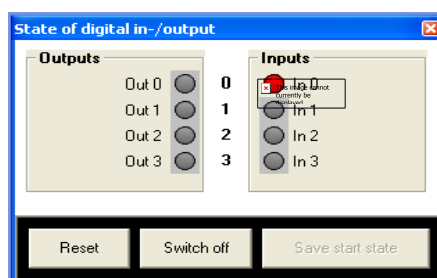
	通过鼠标右键点击 LED 符号，可修改数字代表的输入/输出的逻辑条件。
---	-------------------------------------

右键点击输出数据的 LED 区域后显示如下选项。

如果在此选项中改变模式，将有更多的数字输入/输出状态菜单显示在 LED 区域。



当逻辑层次0连接到反向输入时，可以看到程序执行的结果。反向确保了软件可识别逻辑层次1。

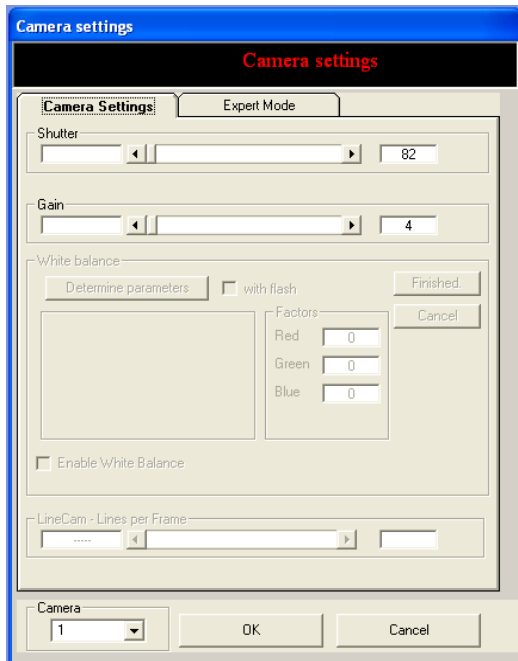


3.12 相机选项（Camera Options）

相机选项表的内容由当前使用的相机和激活的相机决定。SmartBuilder 系统菜单提供不同的设置如针对黑白和彩色相机。在系统列中，可以找到针对不同入口和相机的设置。（智能相机 IEEE1394, USB2.0, CameraLink, GigE, 相似物）

在以下描述中可看到针对智能相机的菜单。

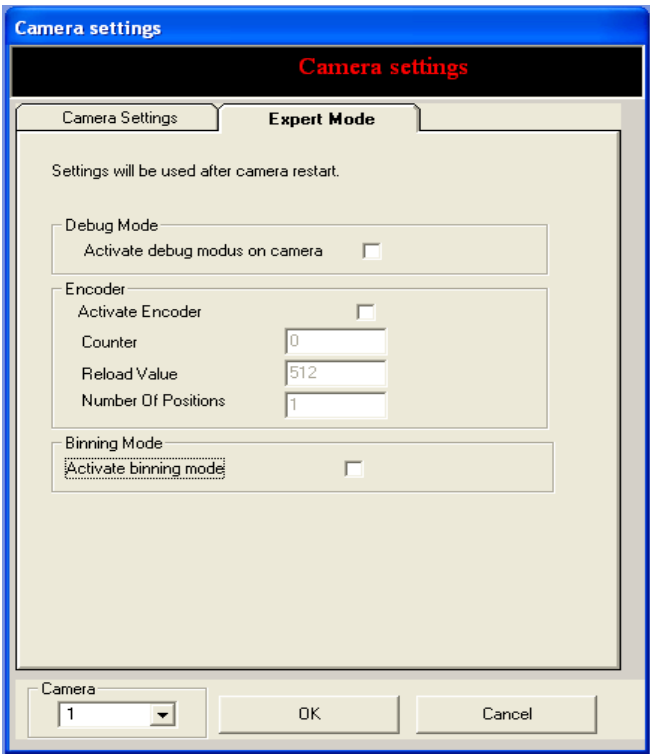
相机设置（Camera Settings）



设置此菜单，可直接用鼠标左键点击文本框并输入数值。或者，通过移动箭头键修改数值。另外，也可以按住鼠标滑动滑块控制数值。

快门（Shutter）	用滑块或直接输入数值控制快门速度。如果连接了相机，并且处于实时模式，可立即看到快门速度的改变。
增益（Gain）	由增益滑块控制数字化的放大倍数。也可以直接通过输入数值定义放大倍数。
白平衡（White balance）	白平衡仅适用于彩色相机。白色表面可以由三种基本颜色来确定。
LineCam	在此定义行相机运行的模式，循环模式或全幅模式。循环模式下，相机运转在无终止的图像上。在全幅模式下，图像被组成可调的 N 条线然后进行评估。
相机（Camera）	如果系统连接多个相机，选择设置一个相机。

高级模式（Expert Mode）（相机设置）



调试模式（Debug Mode）

激活 调试 模式 （Activate Debug Mode）	相机运行在调试模式下，在下次系统启动后，相机通过 Telnet 调试输出信息。
--------------------------------	---

编码器（Encoder）

激 活 编 码 器 （Activate Encoder）	有编码接口的相机，可以用此功能触发编码器。
寄存器（Counter）	启动寄存器，直到触发被释放。
重新载入（Reload）	触发寄存器期满后重新载入值。
位置数量（Number of Positions）	编码器的位置数量。

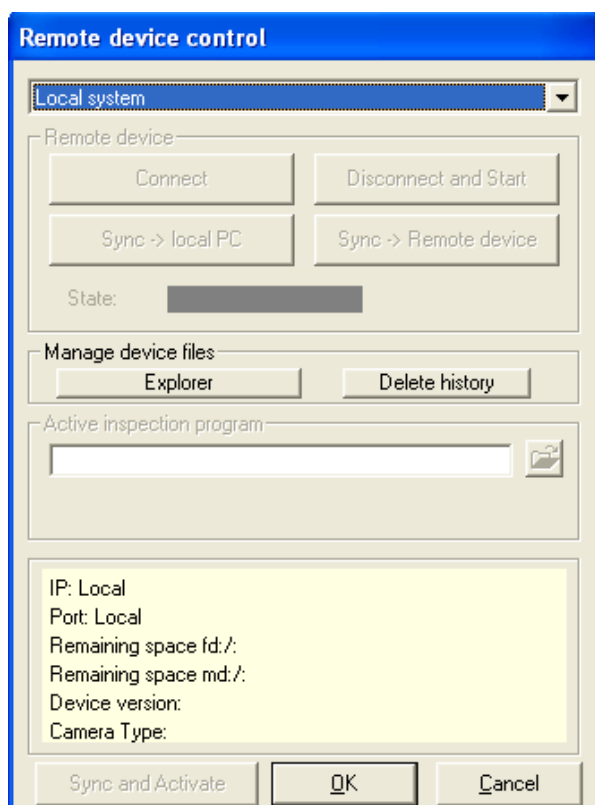
输入/输出触发 (I/O Trigger)

使用输入/输出触发 (Use I/O Trigger)	标记激活输入/输出图片触发。
输入 (Input)	触发输入。
超时 (Timeout)	等待触发的时间。
最小信号时间 (Min. signal time)	接受有效触发信号的最短时间。

分级模式 (Binning Mode)

激活 (activate)	激活相机分组模式，对图片做几何调整。
---------------	--------------------

3.13 远程设备控制（Remote Device Control）



通过远程设备控制，所有非单一系统的通讯介质可用，因此程序层面的系统和运行系统是分开的。

本地系统（*Local system*）选项是“脱机测试版本”，我们建议始终修改至系统配置。只要激活另一个系统，即使是脱机版，所有在这里的文件也会在下次输入并转移到相机中。

下拉菜单（Pull-Down Menu）

	用鼠标左键点击箭头可以显示出所有的连接的相机。再点击就可以选择一个模式。如果下拉菜单中没有什么内容，则需要到远程设备配置中添加。
---	---

远程设备（Rmote Device）

连接（Connect）	选择一个设备后可连接到系统。如果选择了相机中的检查程序（ <i>inspection program on the camera</i> ），当前检查程序将被中止。
断开连接并开始（Disconnect and Start）	断开当前的连接。如果选择了执行相机中的检查程序（ <i>inspection program execution on the camera</i> ），相机中的检查程序自动开始。

同步到本地 PC (Sync → local PC)	首次连接时，或相机使用多个编辑系统时，首先需要同步电脑和相机，以保证工作于相机的当前文件。这样做可以避免由于使用其他系统做修改而引起的无意删除行为。
同步到远程设备 (Sync → Remote Device)	用此按钮同步当前文件编程系统中的图像到相机，以确保所有执行需要的文件都存入相机（如 OCR 读码所需的字符集文件）。
状态 (State)	1. 未连接 – 系统未连接相机 2. 连接 – 系统连接了相机 3. 测试版，如果连接了相机，但相机中没有有效证书。

管理设备文件 (Manage Device files)

资源管理器 (Explorer)	打开浏览文件的资源管理器视窗，可在此删除或转移，并在同步到远程设备后修改。
------------------	---------------------------------------

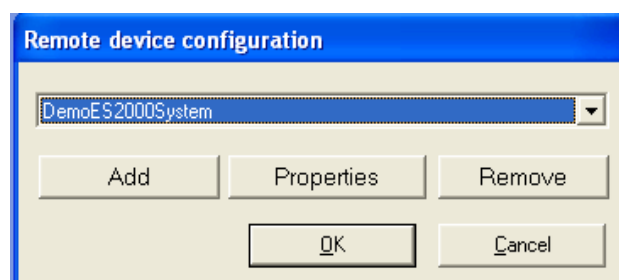
激活检查程序 (Active inspection program)

选择 (Selection)	在激活的检查程序中显示当前使用的检查程序。如果要激活相机中的另一个程序，可以用鼠标左键选择另一个文件。
执行检查程序 (Inspection program execution)	只要编程系统连接相机，选择这一选项将停止相机中的检查程序。由于编辑系统到相机的访问是透明的，因此可以改变顺序。这样做会引起部分程序没被执行。所以，我们建议在连接过程中中止程序。

概览 (Overview)

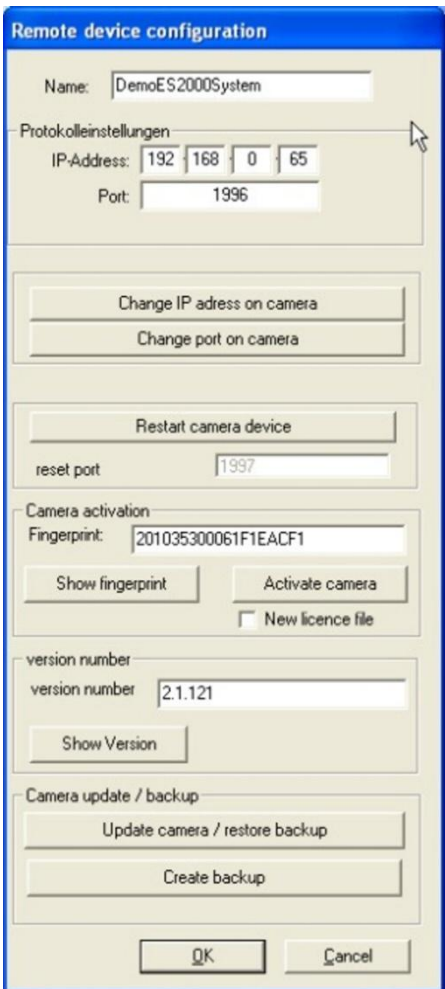
IP: 本地 (IP: Local)	连接编程系统的相机的 IP 地址。
Port: 本地 (Port: Local)	编程系统与相机沟通的端口。
闪存空间 (Remaining space fd)	相机闪存卡中的剩余空间。
SD 卡空间 (Remaining space md)	相机 SD 卡中的剩余空间。
同步并激活 (Sync and Activate)	调整相机中的当前图像和计算机系统文件，之后开始执行检查程序。

3.14 配置远程设备（Remote Device Configuration）



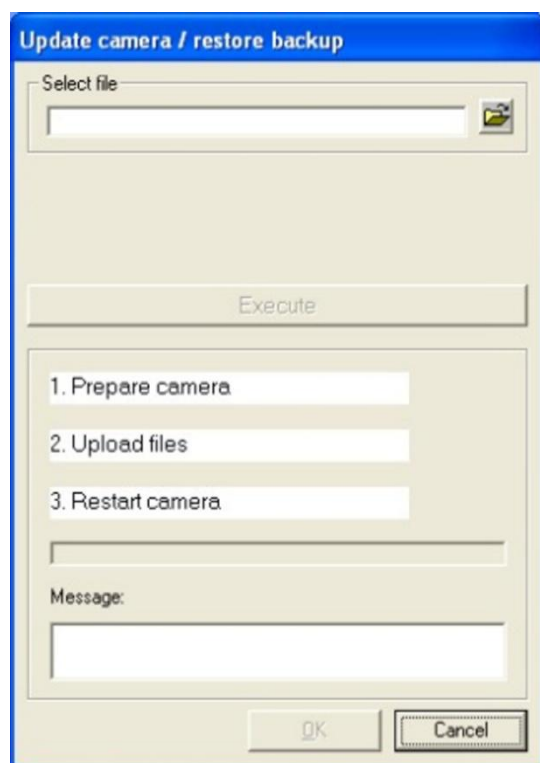
增加（Add）	新安装的系统还没有相机的设置。必须通过增加（Add）设置相机，用鼠标左键选择属性（property），在里面输入相机名称。如果这是个新出厂的相机，那么可以直接使用其中的预设置，否则就需要在这里分配给相机一个 IP 地址。
属性（Properties）	增加后，可以此设置所有其他属性。（如下）。
移除（Remove）	删除设置的连接。

属性（Properties）

	远程设备配置的属性中，可修改或更新相机的地址。此修改必须同时在协调设置中进行，否则将连接不到相机。
--	--

名称（Name）	为相机命名。
IP 地址（IP-Address）	显示当前 IP 地址。如果要修改 IP 地址，也必须在这里输入。
端口（Port）	与相机交换信息的端口。如果要改变信息端口，也必须在此输入。
修改相机 IP 地址（Change IP-Address on camera）	打开一个新窗口，在此输入相机的新 IP 地址。必须在 IP 地址栏输入，否则将连接不到相机。
修 改 相 机 端 口 （ Change Port on	打开一个新窗口，在此输入交换信息的新端口。如果要增加一

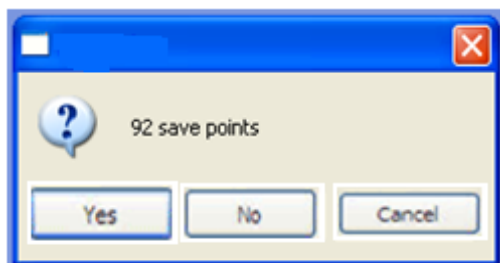
Camera)	个新的端口，也要在此输入端口号。
修改相机名称 (Change name of camera on camera)	为相机指定新名称。
重启相机 (Restart Camera device)	向相机发送信息，实现相机的热启动。重启端口是固定的不可以修改。
版本号 (Version number)	在此显示相机的版本号，这个号码应该相等或低于编程接口显示的版本号。最好的情况是两个版本号相同。
指纹 (Fingerprint)	指纹是与相机精确匹配的，基于此进行所有的软件行为。如果要对相机进行更多操作，需要把指纹发给 VizCam。
显示指纹 (Show fingerprint)	显示当前激活的相机的指纹。
激活相机 (Activate camera)	用此按钮激活相机。如果同时选中新证书文件 (New License File)，用户需要选择一个新证书文件。
新证书文件 (New license file)	用此选项为相机指定一个新的证书文件，否则相机会用原来的证书再次被激活。
更新相机/保存更新 (Update camera/Restore update)	用此按钮更新相机，或将创建的备份恢复。
创建备份 (Create Backup)	用此按钮备份相机中的所有文件。以后可以随时将相机状态返回到备份时刻。

更新相机/保存备份（Update camera / restore backup）

选择文件（Select file）	点击打开图标，选择一个要更新或备份的文件，此文件将被传送到相机。相机中所有的文件夹消失，之后相机恢复到出厂状态或备份时的状态。
执行（Execute）	按执行按钮后，更新或备份将传送到相机，所有步骤由软件记录输出。请按指示操作。
信息（Message）	执行过程中，保存在此窗口中的所有动作。

3.15 保存点列表（Store Point list）

用此菜单选项保存当前的点列表。例如当前的轮廓点可保存后用于以后的对比。在确定要保存当前的轮廓点后，会出现文件选择对话框，在此输入文件名称。如果这个列表将来要用于远程设备，那么不要改变目录。这些点将记录在寄存器中。



这个对话框提示是否保存点列表。如果选择是，将显示一个文件选择对话框，在那里输入文件名称。

3.16 重置变量（Reset variables）

重置所有静态变量。

3.17 关闭打开的通讯端口（Close Open Sockets）

关闭所有检查程序过程中打开的通讯端口。

3.18 相机命令集（Instruction Set of the Camera）

根据连接的目标系统显示系统中存在的命令集。因此需要有效的证书代码，即相机读取证书文件，只显示激活的命令集。

在连接 PC 的系统中，需要通过密码狗读取可用的命令集。

3.19 密码管理（可选）（Password Admin）

SmartBuilder 软件有一套综合的密码保护机制。能满足美国 FDA 在21CFR11描述的关于记录和访问控制的要求。

如果这个选项出现在系统中，用户会收到更多文件保证能够修改访问控制。

要访问远程系统，用户需要访问密码。编程界面立刻接受，并要求用户名和密码。

3.20 SmartBuilder 软件的数据结构

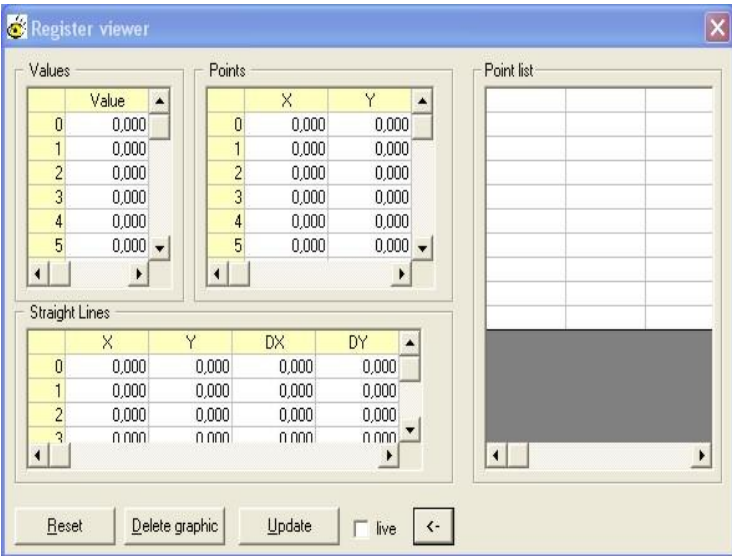
SmartBuilder 做测量的所有指令的基础是数据结构。他们代表了几何对象和值。主要有五种数据结构：

- 点
- 直线
- 圆
- 值
- 点列表
- 字符串

除点列表外，其他都在寄存器中管理。

SmartBuilder 软件的寄存器

寄存器是以 FIFO（先进先出）为原则的缓存。可存入或读取不同的对象，每个保存在缓存中的对象有一个序列号。



SmartBuilder 中有四种寄存器：

- 值寄存器
- 点寄存器
- 直线寄存器
- 圆寄存器

每种类型寄存器有20个位置。
值，点和直线内容可在寄存器视窗中看到。

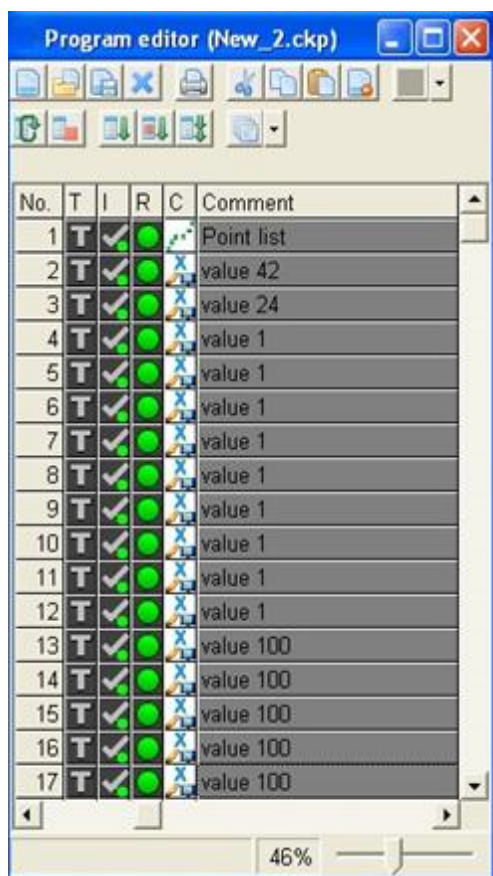
先进先出寄存器功能示例：

在桌子上摆书，直到桌子摆满。如果要再放一本书，向前推所有的书，第一本书掉到地上。这就是先进先出的原则。

寄存器分布示例:

寄存器中值的分配示例，如果值是逐渐增加的:

command 1: set back			command 2: Value 42 added			command 3: Value 24 added	
Index0	0		Index0	42		Index0	24
Index1	0		Index1	0		Index1	42
Index2	0		Index2	0		Index2	0
...			...			...	
Index17	0		Index17	0		Index17	0
Index18	0		Index18	0		Index18	0
Index19	0		Index19	0		Index19	0
command 4: Value 1 added 9 times			command 5: Value 100 added 10 times			Command6: Value 1000 added 10 times	
Index0	1		Index0	100		Index0	1000
Index1	1		Index1	100		Index1	1000
Index2	1		Index2	100		Index2	1000
Index3	1		Index3	100		Index3	1000
Index4	1		Index4	100		Index4	1000
Index5	1		Index5	100		Index5	100
Index6	1		Index6	100		Index6	100
Index7	1		Index7	100		Index7	100
Index8	1		Index8	100		Index8	100
Index9	24		Index9	100		Index9	100
Index10	42		Index10	1		Index10	100
Index11	0		Index11	1		Index11	100
Index12	0		Index12	1		Index12	100
Index13	0		Index13	1		Index13	100
Index14	0		Index14	1		Index14	100
Index15	0		Index15	1		Index15	1
Index16	0		Index16	1		Index16	1
Index17	0		Index17	1		Index17	1
Index18	0		Index18	1		Index18	1
Index 19	0		Index 19	24		Index 19	1



如果加载 `registerexample.CKP` 程序并一步一步执行程序，可以在寄存器视窗中看到上面的情况。

值 (Values)

<table border="1"> <thead> <tr> <th>Value</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.000</td> </tr> <tr> <td>1</td> <td>0.000</td> </tr> <tr> <td>2</td> <td>0.000</td> </tr> <tr> <td>3</td> <td>0.000</td> </tr> <tr> <td>4</td> <td>0.000</td> </tr> <tr> <td>5</td> <td>0.000</td> </tr> </tbody> </table>	Value	Value	0	0.000	1	0.000	2	0.000	3	0.000	4	0.000	5	0.000	可能是正的或负的浮点值。值存储于值寄存器。
Value	Value														
0	0.000														
1	0.000														
2	0.000														
3	0.000														
4	0.000														
5	0.000														

点 (Points)

Points

	X	Y	
0	0,000	0,000	
1	0,000	0,000	
2	0,000	0,000	
3	0,000	0,000	
4	0,000	0,000	
5	0,000	0,000	

点代表 X-/Y-坐标系统中的单独的点或像素。
点存储于点寄存器中，有两个值， X 和 Y 坐标值。

直线（Straight lines）

Straight Lines				
	X	Y	DX	DY
0	0,000	0,000	0,000	0,000
1	0,000	0,000	0,000	0,000
2	0,000	0,000	0,000	0,000
3	0,000	0,000	0,000	0,000

直线有点的尺寸和用户定义的扩散。
在寄存器中直线存有四个值—直线通过的点的值和直线梯度。

圆（Circles）

Points			Values	
	X	Y	Value	
0	0,000	0,000	0	0,000
1	0,000	0,000	1	0,000
2	0,000	0,000	2	0,000
3	0,000	0,000	3	0,000
4	0,000	0,000	4	0,000
5	0,000	0,000	5	0,000

圆包括圆心点和半径。圆本身没有寄存器，圆心的点保存在点寄存器，半径保存在值寄存器。

点列表（Point list）

	X	Y
0	15,143	-15,360
1	15,162	-15,217
2	15,168	-15,069
3	15,186	-14,925
4	15,199	-14,779
5	15,205	-14,706
6	15,214	-14,559
7	15,229	-14,414
8	15,257	-14,274
9	15,299	-14,139
10	15,412	-14,029
11	15,571	-13,936
12	15,696	-13,831
13	15,738	-13,771

点列表与点寄存器相似，点列表用于命令，而此命令会提供大量的数据，如卡尺。
点列表中的点存有两个值，X 和 Y 坐标值。
新的点将永远加在点列表的最后面。

全局字符串（Global String）

SendString

全局字符串变量用于数据的准备和发送。所有产生字符串的命令，增加字符串到缓存。在 SendString 文本框中显示缓存中的当前字符串。用重置下一行（reset text line）命令可重置 SendString 缓存中的内容。

选取列表（Pick-up Lists）

选取列表是连接不同命令的结果值与位置的完全数据结构，并在此保存这些数据。现在，用户有了访问结果的办法，而不需要受任何增加的检查程序的影响。例如如果要通过两条直线确定角度，可以直接从寄存器中选择直线确定。或者可以通过选取列表，选择直线并用于这个命令。

4 命令

以下章节包含了的标准 **SmartBuilder** 软件的所有命令的详细描述。当然，因为这是一个标准的手册，所以不包括所有可用的命令。也就是说，该手册不包含特殊指令集，因为那是针对特殊行业的指令。

4.1 容差和警戒限

所有命令会得到一个结果值，在这里可以为结果值定义一个范围，即在什么情况下是成功，不成功或者警戒。“警戒限”允许用户在测量超过容差限制之前及时做出反应，而不是超过限制以后。

以下描述的是相应的输入对话框，每个命令的对话框都是相同的。因此，对于每个命令中的容差都可参考以下描述。

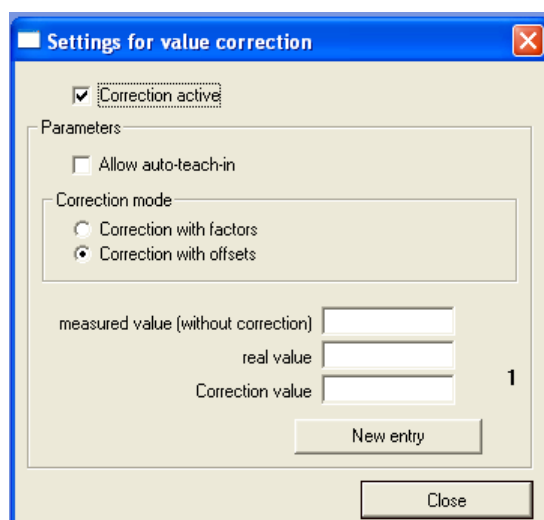
容差（Tolerance）

设定（Setpoint）	设定值决定了结果是 OK 或 NOK。
警戒限（Warning limit）	如果超过了警戒限，会有警告。表示结果虽未到达 NOK 的标准，但已经接近 NOK 了。

全局容差（Global Tolerances）

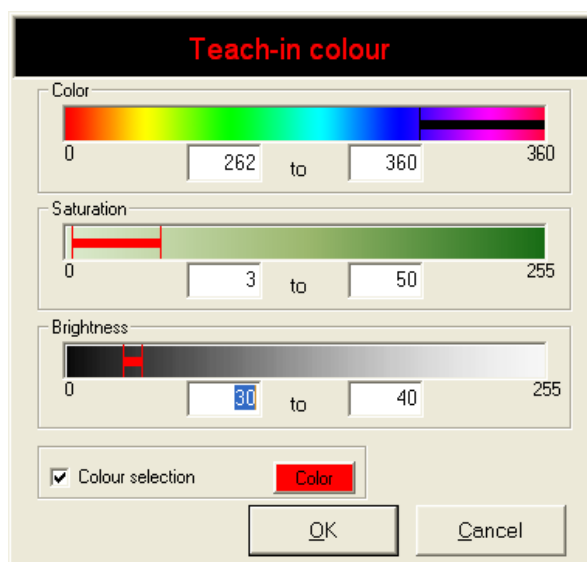
全局容差（Global Tolerances）	如果选中此项，可从全局容差中选择一个容差。从表中为此命令选择一个容差。这种方法的好处是用户不需要通过进入检查程序修改容差。可在全局容差的表格中设定，之后，就可以调用不同的全局容差了。
-------------------------	---

值修正 (Value correction)



修正激活 (Correction active) 如果选中修正激活，会对每个有此选项的命令结果值进行修正。

色彩选择表 (Color Selection Form)



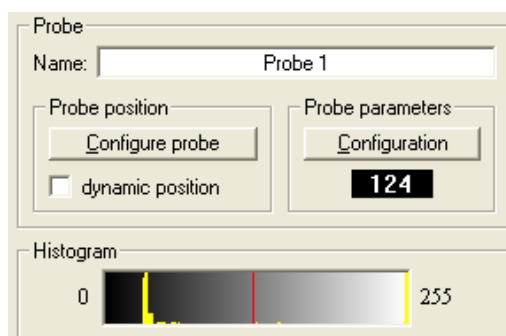
在这里可以通过输入定义颜色的值，饱和度和亮度。也可以通过滑块定义，但必须在相对应的显示区域滑动。单击鼠标左键定义左侧的值，单击鼠标右键定义右侧的值。拖动时需要相应地按住鼠标左键或右键。

确定探头参数

所有命令中的探头参数菜单都是相同的。用户可以为边缘探测定义不同的参数。标准的程序参数是对齐的，所以能够适合有照白条件的白亮的物体加黑背景的图像。

如果有不同的照白条件，需要根据照白修改值。如果边缘探测是在不同的照白条件下进行，最简单的办法是在全局边缘参数中修改值。这些修改会自动传送到命令。所以不必为每个命令修改边缘参数。

不同探头的配置区域看起来很类似。



探头位置（Probe position）

配置探头（Configure probe）

探头参数（Probe parameter）

配置（Configure）

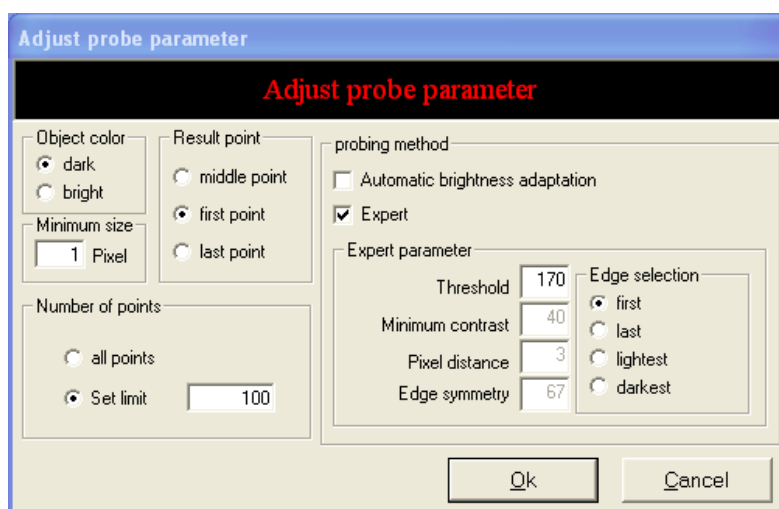
按此按钮，显示下面的调整探头参数（*adjust probe parameter*）菜单。

G170

用鼠标左键双击这个图标以改变搜索对象的颜色。（现在的例子显示的是对象是白色，黑影是黑色。

确定探头参数（Probe parameter determination）

在此修改边缘探测的探头参数。



对象颜色（Object color）

黑（dark）	设定对象颜色为黑。
---------	-----------

白（bright）	设定对象颜色为白。
-----------	-----------

结果点（Result point）

中间点（middle point）	搜索到的中间点作为结果点。
-------------------	---------------

第一点（first point）	搜索到的第一个点作为结果点。
------------------	----------------

最后一点（last point）	搜索到的最后一点是结果点。
------------------	---------------

探测方法 (Probing method)

自动亮度调整 (Automatic brightness adoption)	在探头内有足够的参数去搜索探测边缘。
--	--------------------

专业参数 (Expert parameter)

边缘选择 (Edge selection)	用设定的边缘参数检测边缘位置。
-----------------------	-----------------

阈值 (Threshold)	要超过的最小值。
----------------	----------

最小对比度 (Minimum contrast)	边缘位置的对比度。
--------------------------	-----------

像素距离 (Pixel distance)	关于对比度的像素距离。
-----------------------	-------------

边缘对称 (Edge symmetry)	与边缘位置有关的对称位置。
----------------------	---------------

点的数量 Number of points

在这里设置探头内的点的数量，默认情况下，这意味着所有点被减少成一个点，并可提高处理速度。

4.2 命令概述



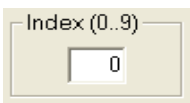
访问变量

通过命令访问变量就象用编程语言处理系统变量一样。可以在这里创建和保存变量，读取变量或计算。变量可保存于寄存器，用于其他系统功能调用。

名称（Name）

为变量命名。

运行（Operation）

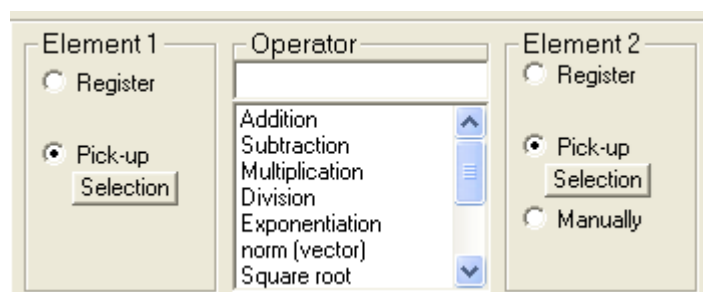
创建（Create）	创建一个新的变量类型。可以使用选项静态变量（ <i>static variable</i> ）创建静态变量。
将寄存器存入变量（Put register into variable） 	选定的序号和实际寄存器输入被保存于之前创建的变量。

将变量存入寄存器（Put variable onto register）	将选择的变量内容存入寄存器。
数学（Arithmetic）	数学运算用于两个选择的成分。
将值存入寄存器（Put value onto register）	输入的值存入寄存器。
将值存入变量（Put value into variable）	输入的值存入变量。
静态变量（Static variable）	新变量是静态的。如果不选择这个选项，当检查程序执行到结束时，变量就会被重置。 静态变量在所有程序循环中保持不变。你可以使用他们当作计数器。如果选中静态变量，将只在命令首次执行时创建变量。命令的结果会说明是否需要创建变量。

变量类型（Variable type）

在此创建变量类型：值，点等。

数学（Arithmetic）



元素1/元素2（Element 1 / Element 2）

你可以选择从寄存器或选取列表中选择元素。

运算（Operator）

在运算列表中选择一种运算。之后这个运算作用于两个元素。



角 (Angle)

Configure command Angle

Angle

Determines and evaluates the angle between two straight lines or three points (in degree) Measures the angle between two straight lines or three points.

Angle definition

☒ Straight line 1 - straight line2
☐ Straight line 2 - straight line1
☐ Three points
☐ Vector (betw. 2 points)

☐ Global Tolerances ☐ Dynamic evaluation

Tolerance

Nominal value: 0,0 + 0,0 - 0,0

☐ Warning limit:

First straight line

☒ from register Index: 0
☐ from pick-up list:

Second straight line

☒ from register Index: 1
☐ from pick-up list:

OK Cancel

用此功能定义一个角，可以是两条直线间的角，三点间的角或两点间的微量。

定义角 (Angle definition)

直线 1 – 直线 2	直线1到直线2间的角。
直线 2 – 直线1	直线2到直线1间的角。
三点	从点寄存器中选择三个点，计算角度。
两点间的向量	计算两点间的向量。

第一条线/第二条线 (First straight line / Second straight line)

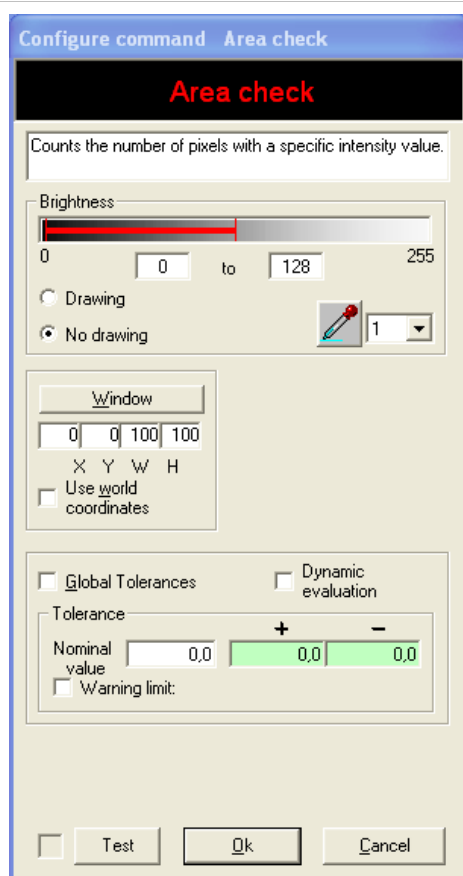
在此选择直线，可从寄存器中选择或从选取列表中选择。

容差 (Tolerance)

这里有许多不同的命令，允许 OEM 客户在这里创建有效的系统。尤其要指出的是 AIWIA（增量积累的自适应图像捕获），连接智能相机平台，允许以最大切割尺寸捕获242图像。

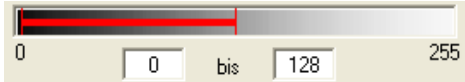


检查区域（Area check）



用此命令在配置窗口定义亮度，并与配置像素做对比。

亮度（Brightness）

	可直接输入数值，或点选左右区域确定亮度值。
图形 / 无图形（Graphic / no graphic）	如果激活图形，选择的亮度的像素点会标记为彩色。
	用滴管可直接在图像上选择亮度区域。
窗口（Window）	定义一个窗口，在此窗口中进行过滤运行。
位置追踪（Position tracing）	选定窗口，进行位置追踪。
容差（Tolerances）	



异步图像采集（Asynchronous image capture）

Configure command Asynchronous image capture

Asynchronous image capture

This command starts an asynchronous image capture or checks its status

Operation

☒ Start image capture

☒ triggered ☐ with flash

Camera: 1

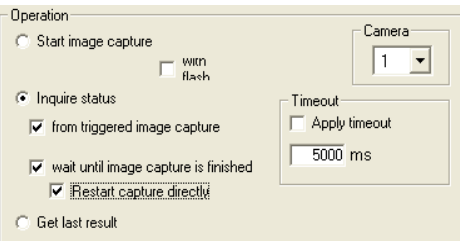
☐ Inquire status

☐ Get last result

OK Cancel

用此命令启动异步图像采集或检查其状态。即可通过触发输入或闪光触发相机捕获图片。异步图像采集用于快速检测和没有规律地检测。

运行（Operation）

开始捕获图像（Start image capture）	通过触发或闪光执行捕获图像。
 询问状态（Inquire Status）	在这里，人们可以查询触发图像采集的状态，或等到完成图像采集。该命令启动图像采集并马上返回。如果选中“询问状态”，它会返回告知是否准备好配置。另外可以指定，通过（闪光）重新启动图像采集或使用超时限制。
获得上次结果（Get last result）	在此可获取上次图像捕获的结果。



条码（Bar code）

Configure command Barcode

Barcode

Code type

☒ Two widths barcode

☐ Multiple widths barcode

☐ any

Code:

2/5 Interleaved

Window

☐ Configuration

Search window

0050

X Y Wi He

☒ Complete window

Output

☒ Delete text string before use

Check norm

☐ Code content

XXXX

☐ Check code length

Scan

☐

Ok

Cancel

用此功能可读取不同的条码。

条码类型（Code type）

两种宽度条码（Two widths barcode）	使用两种宽度的条码。
多种宽度条码（Multiple widths barcode）	使用多种宽度的条码。
任何（any）	读码器搜索所有条码，因此读取速度会比指定了条码类型时慢。

代码（Code）

下拉菜单可以定义要读取的代码。
注：根据客户需要可以激活更多代码类型。

检查规范（Check norm）

代码内容（Code content）	检查代码内容，在文本框中显示读取结果。
代码长度（Code length）	检查代码长度。

输出 (Output)

使用前删除文本字符串	设定此项，决定是否在输出前删除文本内存。
------------	----------------------

窗口 (Window)

配置 (Configuration)	配置搜索窗口。
整个窗口 (Complete window)	在整个窗口搜索条形码

亮度调整（Brightness adjustment）

Configure command Brightness adjustment

Brightness adjustment

Method

☐ Compensation of brightness offset

☒ Greyscale spreading

Allowed deviation: Grey values

Reference intensities

Dark area

Window position

X Y W H

Bright area

Window position

X Y W H

☐ Determine intensities via histogram-analysis

☐ Expert

Correction area

☒ Correct whole image

☐ Use world coordinates

Ok

Cancel

可对选定的区域或整个图像在两个强度参考区间修改灰度值。

方法（Method）

灰度蔓延（Grayscale spreading）	位于白黑间的灰度值会在修正范围内蔓延到整个区域。
---------------------------	--------------------------

参考强度（Reference intensities）

黑区（Dark area）	配置黑的区域。
白区（Bright area）	配置白的区域。
通过直方图分析确定强度（Determine intensities via histogram-analysis）	由区域直方图确定强度分布。

专业 (Expert)	用户可直接输入白黑区域的强度分布。
-------------	-------------------

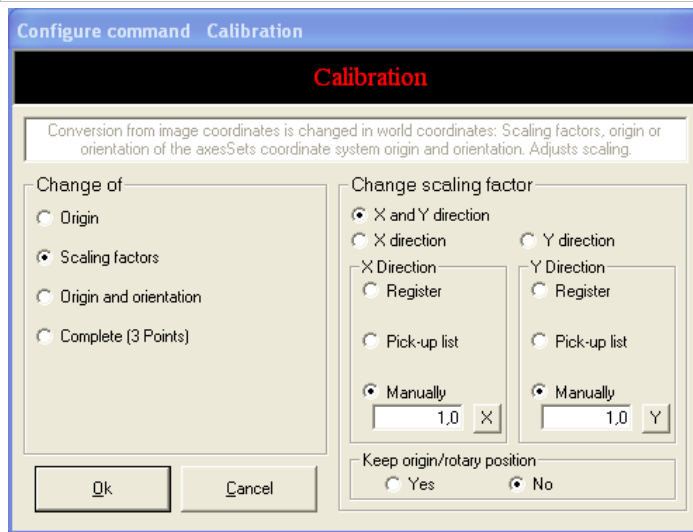
检查区域 (Correction area)

检查整个图像 (Correct whole image)	调整整个图像或配置的长方形区域。
------------------------------	------------------

使用世界坐标 (Use world coordinates)	考虑位置调整。
--------------------------------	---------



校准（Calibration）



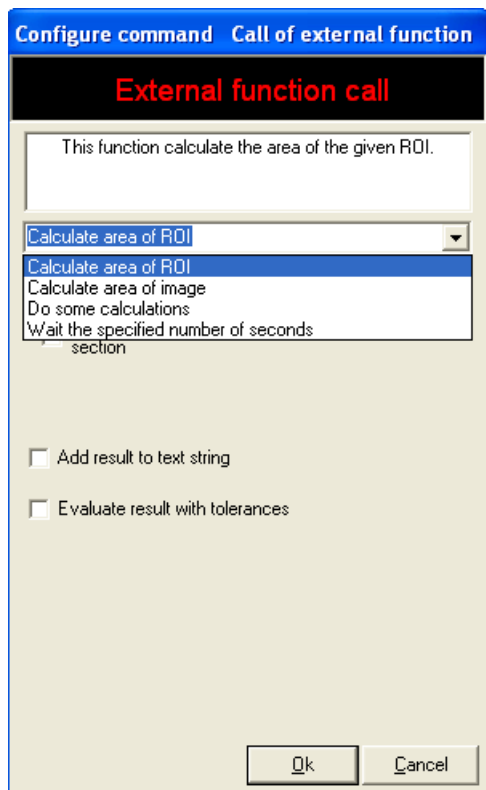
校准命令可用于缩放图像区域。亦可以用来追踪图像在 X 轴 Y 轴上的移动，和旋转的位置。

改变（Change of）

原点（Origin）	定义新的原点。平行于坐标的移动，如 X 值或 Y 值的变化。或者两个值同时变化，即全新的坐标系统原点。
缩放因子（Scaling factors）	用缩放因子可以把相机坐标改成测量中的全局坐标。可由确定的数值或根据计算值变换。
原点和定位（Origin and Orientation）	不仅可以定义一个新的坐标原点，也可以追踪到转动的位置。
完整三点（Complete (3 Points)）	根据三点进行校准不仅适用于定义新的原点和旋转方向，而且适用于缩放。



调用外部功能（Call for external function）



利用此功能，用户可以进行多种计算，计算结果将保存在寄存器中。用户必须用 C++ 编写程序。该工具提供了四种按需提供的源代码示例。

示例计算（*Do some calculations*）显示了可以定义的任何积分值和浮点值。在寄存器中显示数学运算。

定义图像区（Define image section）

用此功能定义图像区域，在此区域内使用功能。

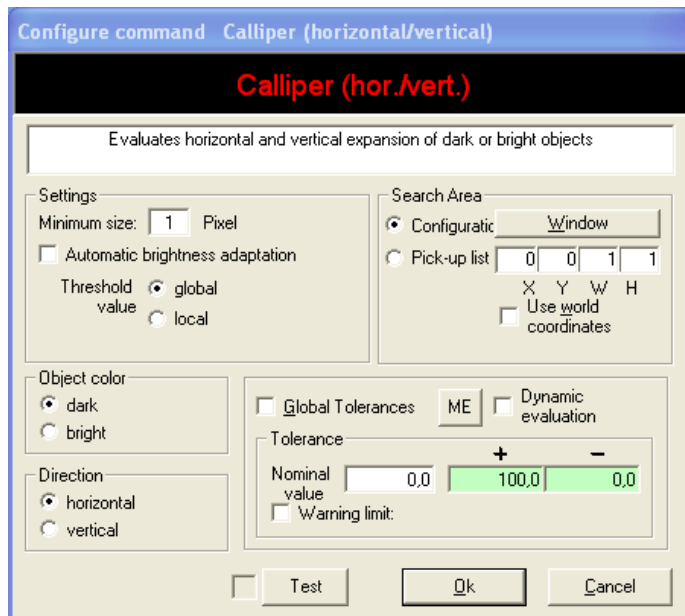
增加结果到文本字符串（Add result to text string）

将结果增加到文本字符串。

用容差评估结果（Evaluate result with tolerances）



游标卡尺（水平/垂直）（Calliper (hor. /ver.)）



配置一个区域，定义要搜索的指定颜色的对象。水平搜索时，从顶部和底部同时搜索指定颜色的对象。一旦两个方向上都找到了对象，即在两个位置间画一条虚拟的直线，并测量其距离。

设置（Settings）

最小尺寸（Minimum size）	定义对象有最小像素尺寸。
☐ Automatic brightness adaptation 自动亮度调整	自动定义阈值识别对象边缘。
全局阈值（Threshold value global）	边缘探测时使用全局阈值。
本地阈值（Threshold value local）	使用指定的边缘值。

搜索区域（Search area）

配置（Configure）	
选取列表（Pick-up list）	
位置追踪（Position tracing）	如果选择这一选项，会自动追踪游标卡尺。

对象颜色 (Object color)

暗 (dark)	搜索第一个暗的对象。
亮 (bright)	搜索第一个亮的对象。

方向 (Direction)

水平 (horizontal)	从顶部和底部搜索定义的对象颜色。
垂直 (vertical)	从左右两侧搜索定义的对象颜色。



自由游标卡尺（Caliper free）

Configure command Free caliper

Free caliper

Just like with a real measurement slide, two points will be probed and their distance will be evaluated.

Configure caliper

Dynamic probe direction

☐ Align measurement slide on straight line Selection (?)

☐ Dynamic position from register

Probe parameters of both measurement jaws

Parameters of both jaws

G170

Parameters lower jaw

Histogram

0 255

☐ Global Tolerances ME ☐ Dynamic evaluation

Tolerance

Nominal value 0,0 + 0,0 - 0,0

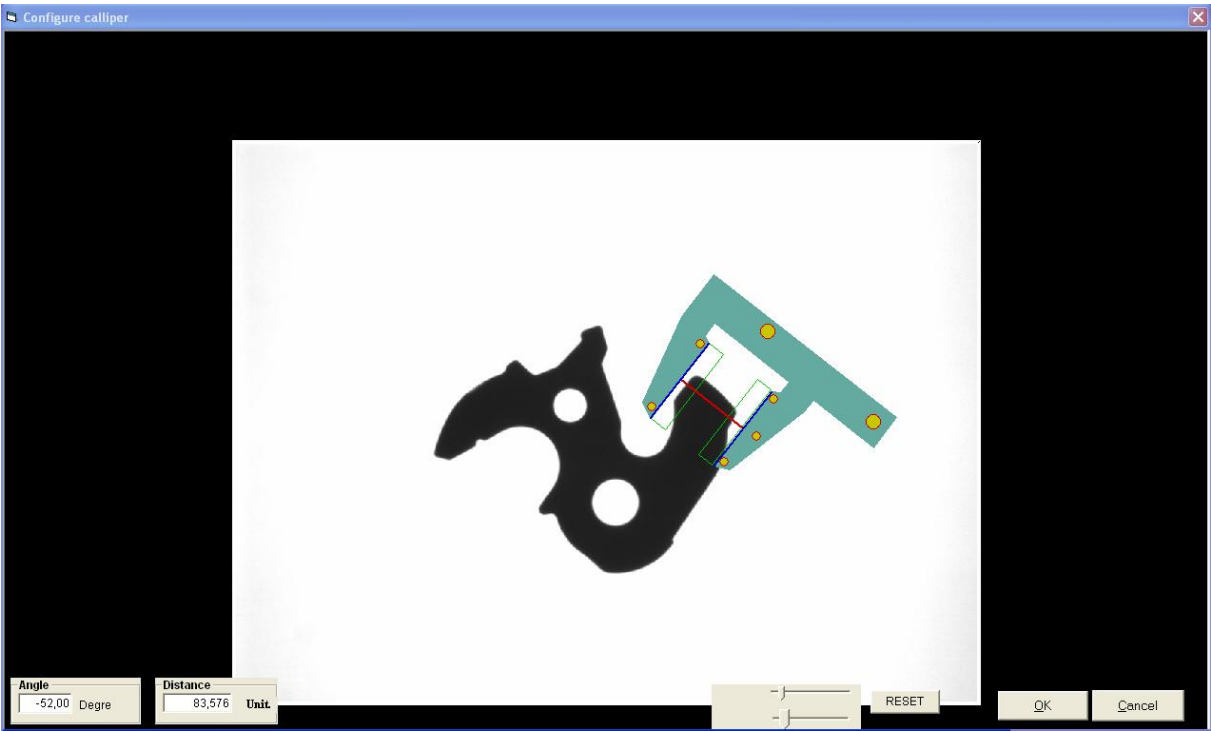
☐ Warning limit:

☐ Test

使用自由卡尺很象使用真正的卡尺。将自由卡尺放在要测量的位置。用 F1和 F2控制两颚，完成整个测量任务的配置。

配置卡尺（Configure caliper）

在此配置卡尺位置和两颚的距离。将对象放在卡尺两颚间，可用卡尺测量其距离，卡尺不需要垂直测量，如最小距离，而是测量用户指定方向上的两颚间的距离。点击配置卡尺按钮，打开卡尺配置窗口如下图。可用卡尺上的黄点定位卡尺，用光标拖动卡尺到理想的位置。同样地，可以拖动颚匹配要测量的距离。唯一需要配置的参数是搜索区域的颜色。例如，如果要测量黑色背景上的白色圆形，两颚参数“parameters of both jaws”中的搜索颜色必须修改成从亮到暗。



Angle -52,00 Degree	显示卡尺的角度，如上图，卡尺角度是-52,00度。
Distance 83,576 Unit	显示卡尺两颚的距离。
	用滑块控制探头尺寸。上面的滑块控制上颚探头，下面的滑块控制下颚探头。
RESET	点击重置按钮后，卡尺回到初始位置。

动态探头方向（Dynamical probe direction）

沿直线测量对齐（Align measurement slide on straight line）	卡尺沿直线列表中已经存在的一条直线对齐。
来自寄存器的动态位置（Dynamic position from register）	以点寄存器中0作为初始点，1作为终点。宽度来自值寄存器。

卡尺两颚的探头参数 (Probe parameters for both measurement jaws)

两颚参数 (Parameter of both jaws)	
	表示对象表面是黑色。双击对象颜色可变成白色。或者可以用两端参数改变对象颜色。



相机控制（Camera Control）

Configure command Camera Control

Camera Control

Set or Read value of gain- or shutter settings.

Type

☐ Gain

☒ Shutter

☐ White balance red

☐ White balance green

☐ White balance blue

Action

☒ Set value

☐ Change value

☐ Read value

Camera

1

from register

from pick-up list:

☒ manual 100

Selection

☐ Slider

☒ Free

☐ Global Tolerances

☐ Dynamic evaluation

Tolerance

Nominal value 0,0

+ 10000,0 - 0,0

☐ Warning limit:

OK Cancel

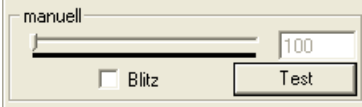
设置或读取增益或快门设置。可以输入一个绝对值，也可以为当前调整增加正值或负值。命令评估进行的值，并将其增加到寄存器。

类型（Type）

在此定义设置或读取的值的类型。

增益（Gain）	定义、修改或读取增益值。
快门（Shutter）	定义、修改或读取快门值，即指定快门速度，确定每个图像的曝光时间。
白平衡红蓝绿（White balance red, green, blue）	定义、修改或读取白平衡值，即哪个色调在特定的灯光条件下被设定为“白”色。

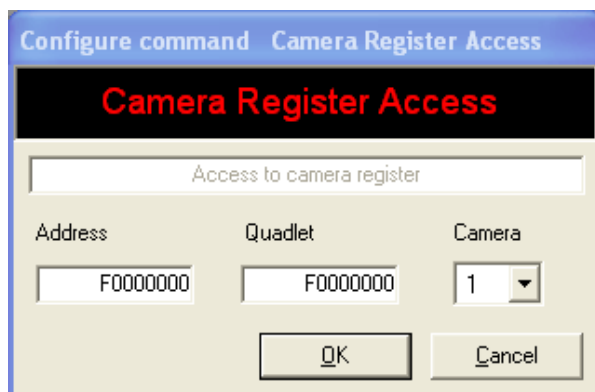
动作 (Action)

设定值 (Set value) 	在此为相应地调整设定一个数值。可以从寄存器或选取列表选择数值，或手动输入数值。输入数值时，可通过滑块选择，也可以直接在文本框中输入。（如左图）
改变值 (Change value)	在此修改各调整的数值。可以从寄存器或选取列表选择数值，或手动输入数值。
读取值 (Read value)	读取对应的调整值。

全局容差 (Global Tolerances)



IEEE1394, FireWire 寄存器访问（Register access）



用这个命令可以为相机写寄存器地址，并可直接使用 IEEE1394相机。用户不必了解专业的 IEEE1394知识，只需要操作标准相机的寄存器。此命令只针对 IEEE1394, FireWire 和 I.Link 相机。

地址（Address）

在 IEEE1394 地址栏输入地址。

Quadlet

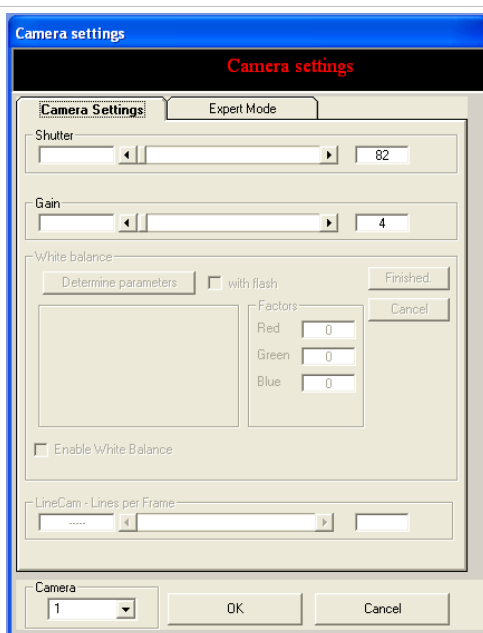
每个地址可以写四个字节。

相机（Camera）

确定相机数量。

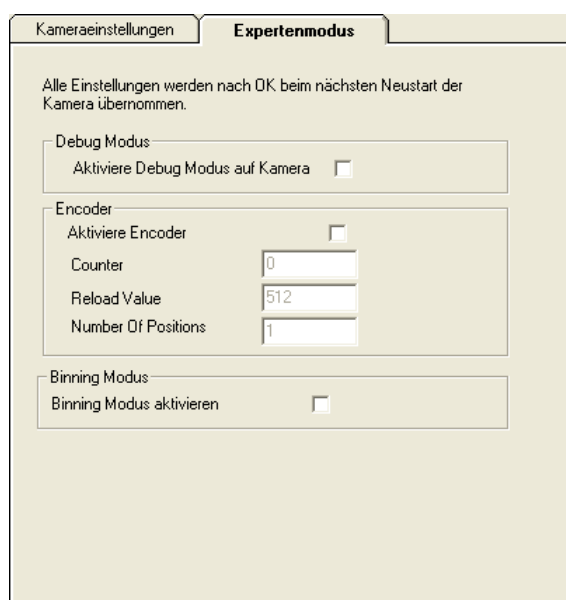


相机设置（Camera settings）



这里，与相机控制命令相似，可以改变相机的快门、增益或白平衡。接口具有控制智能相机的选项，但对于 Firewire 相机可进行更多的调整，如亮度、饱和度、变焦，触发模式等。

高级模式（Expert Mode）



所有的调整改变将在重启相机后生效。

调试模式（Debug mode）

如果激活调试模式，可以获得对每个可能发生的错误的确认。

编码器（Encoder）

激活编码器（Activate Encoder）	在编码器模式下，关闭串行接口 RS232。
计数器（Counter）	激活这个功能，计数器开始递增/递减。
重载值（Reload Value）	如果在图像捕获时没有这个值，为第二次或接下去的图像捕获设定值。

分级模式（Binning Mode）

激活分级模式，近似的像素集中于一个像素区，即获得光敏感度。



改变灰度值（Change gray value）

Configure command Change gray values

Change grey value

All grey values in selected area are set to a certain grey value

Circle

☐ from pick-up list

☐ Configuration

☒ (Radius from value register, center point from

Options

Grey values from

☒ Manually

☐ from register

☐ from pick-up list

to

☒ Manually

☐ from register

☐ from pick-up list

replace by grey value

☒ Manually

☐ from register

☐ from pick-up list

ok Cancel

设定选择的区域内的灰度值。

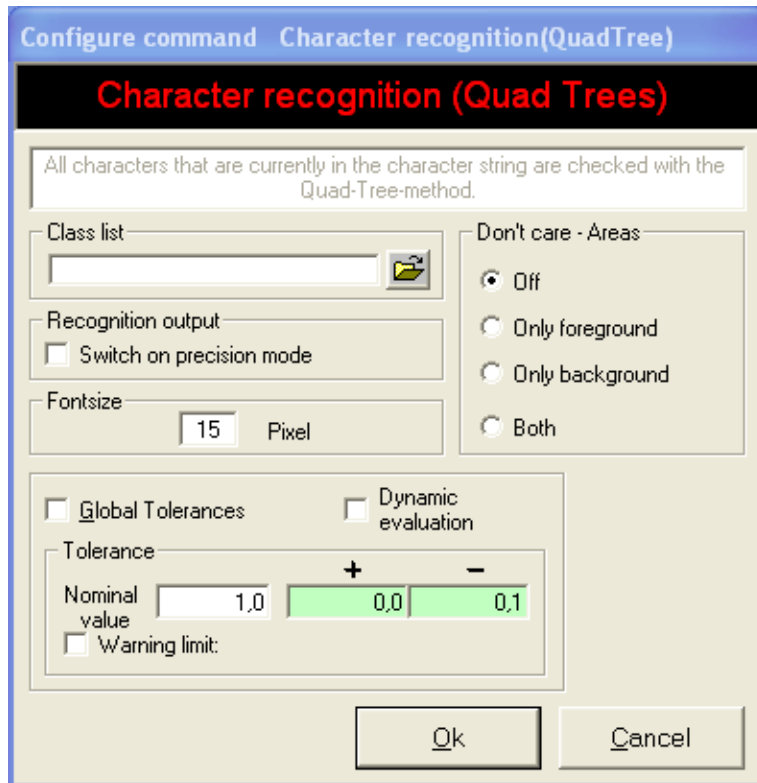
窗口（Window）

选项（Options）

灰度值来自（Gray values from）	定义最小灰度值。手动定义，选自寄存器或选取列表。
到（to）	定义最大灰度值。手动定义，选自寄存器或选取列表。
更换灰度值（Replace gray value）	为源图像设定哪种灰度值。



字符识别（四叉树）（Character recognition (Quad Trees)）



基于为读取字符串提供的字符数据库，此命令用于识别字符。

类列表（Class list）

为每个字符串创建的，分别代表不同字符的字符数据库。

不关心区域（Don't care-Areas）

关闭（Off）	只应使用字符颜色的范围
前景（Only foreground）	也应使用前景。
背景（Only background）	也应使用背景。
前景和背景（Both）	前景和背景都用做识别。

识别输出（Recognition output）

切换到精准模式（Switch on precision mode）	精确模式开始对图像进行精确搜索。此模式可获得更好的结果，但需要大量计算。
-----------------------------------	--------------------------------------

字体尺寸（Font size）

通过字体尺寸参数可以调整显示在叠加层的识别字符大小。

容差（Tolerances）

如果用的是3D 图像获取，将改变 SmartBuilder 软件中的图像获取，在相机取景器中将显示对象的高度，而不是图像灰度值。

根据使用的相机不同，可以做到每秒钟获取3个3D 图像。对3D 图像的评估在相应的命令集中实现。



字符分隔（Character separation）

Configure command Character separation with objects

Character separation with bounding

Objects of a certain brightness are searched in a given window. Set the parameter in a way that only the characters remain as objects.

Window position

Window

10 10 100 100

X Y W H

☐ Use world coordinates

Number of characters

☐ Set number

Area (in pixel)

Min 100

Max 100000

☒ Ignore margin objects

Separating parameter

Set threshold

Lower threshold 171

Upper threshold 255

☐ Autom. brightness adaptation

Clustering

Distance X 0

Distance Y 0

Interim result

Maximum obj. area:

Number of objects: 0

OK Cancel

在读取字符前字符必须彼此分隔，（标准是四叉树（QuadTree））。要参考阅读范围的灰度值。

窗口位置（Window position）

分隔参数（Separating parameter）

设定阈值（Set threshold）	可以输入数值或滑动定义阈值。
自动亮度调整（Automatic Brightness adaptation）	用此选项自动调整阈值分辨字符。

字符数量（Number of characters）

预设定要读取的字符。

聚类（Clustering）

如果字符没有与这些参数关联，定义要确定的字符范围。

区域（Area）

指定要探测的字符所在区域。

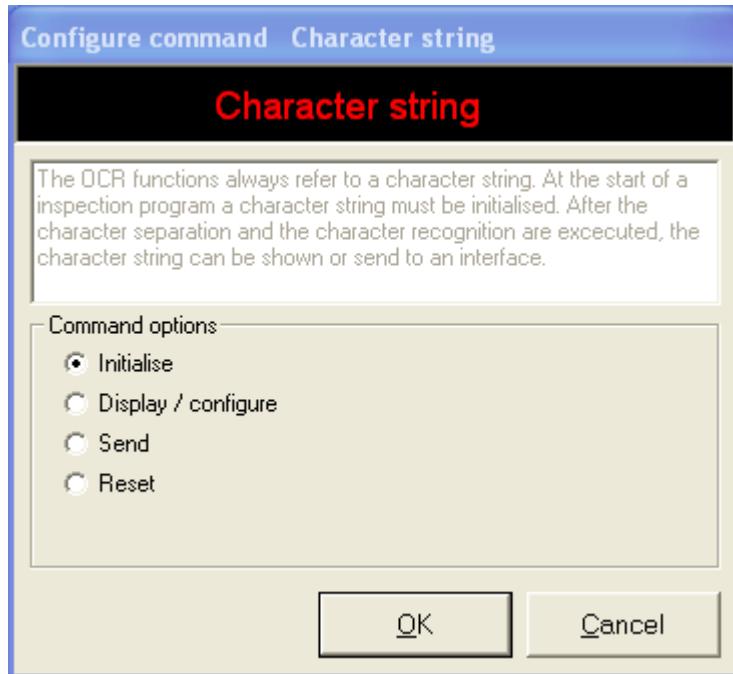
中期值（Interim result）

用户一旦指定一个分隔参数，最大的对象范围和对象数量被认定为中期值。

例如，加载图像 `ManOCRDemo.jpg`，指定低阈值为0，高阈值为247。命令自动识别对象数量和最大对象范围，并将结果显示为中期值。



字符串（Character string）



此命令适用于控制 OCR/OCV 功能。

命令选项（Command options）

初始化（Initialize）	初始化读取字符串。
显示（Display）	显示读取结果。如果没有现存的字符数据库，此命令允许用户创建和配置新的字符数据库。
发送（Send）	读取结果插入完全文本缓冲。
重置（Reset）	重置所有之前运行的读取结果。



检查输入（Check input）

检查数字输入。可通过序列号或输入/输出菜单中的输入名称找到对应的输入。

输入（Input）

序号（Index）	通过序号（首次输入为0）。
名称（Name）	通过输入/输出菜单中的输入名称。

正常状态（Nominal state）	与目标状况对比输入，如果相等命令显示 OK，否则 NOK。
反跳时间（Debounce time）	反跳时间是指输入保持在目标状态下排除故障的最小时间。

模式（Mode）

边沿触发（edge-triggered）	如果选中这个选项，IO 将显示信号从当前状态到目标状态的改变。
水平触发（level-triggered）	如果选中这个选项，IO 显示是否到达目标状态。

单片机控制（半导体） **ChipControl** (Semiconductor)

易于检测半导体领域的 OEM 命令集。客户可单独索取此部分的操作手册。

下面将用到以下一些命令：

方向（Orientation）

共面（Co-planarity）

Pin-One Marker

针隙（Pin-Gap）

表面检测（Surface Inspection）

印刷（Printing）

磁带检测（Tape Inspection）

引线框架检查（Lead frame Inspection）

LED 检测（LED Inspection）

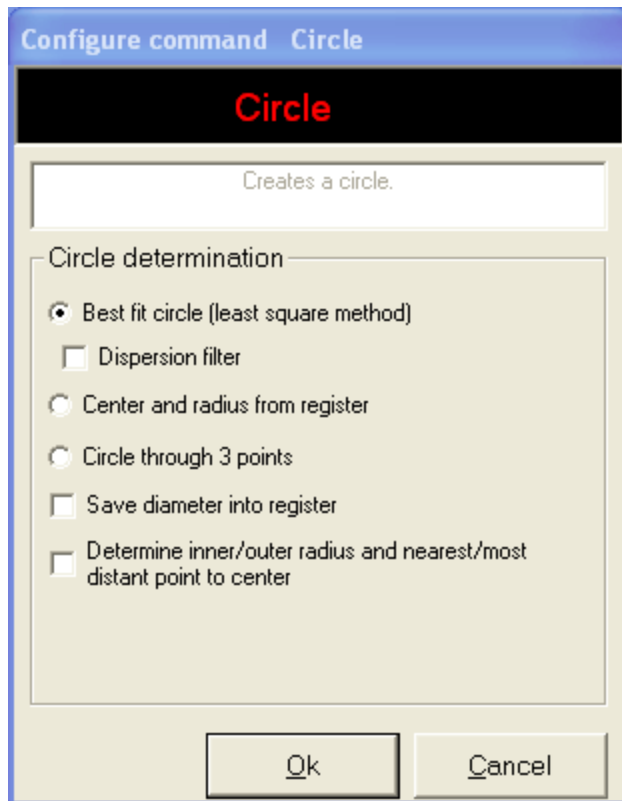
极性（Polarity）

房屋破裂（Housing cracks）

如果你对这一部分的命令集有兴趣，可联络 VizCam 销售部门。你可以索要详细介绍及试用版本。



圆形（Circle）



此命令确定圆。从点列表中确定或三点确定一个圆。作为结果，将圆心的点放入点寄存器，并可选择半径和直径放入值寄存器。

确定圆（Circle Determination）

最适合的圆（最小二乘法）（Best fit circle (least square method)）	通过点列表的点计算最理想的圆。
色散过滤（Dispersion filter）	确保删除跑出的点。
来自寄存器的圆心和半径（Center and radius from register）	从点寄存器中取圆心，从值寄存器中取半径。
通过三点的圆（Circle through 3 points）	用点寄存器的最后三个点确定圆。
保存直径到寄存器（Save diameter into register）	将直径放入寄存器。
确定内/外径（Determine inner/ outer radius）	确定内径/外径。



圆计数器（Circle counter）

Configure command Circle counter

Circle counter

Evaluate number of edges on circle of interest

Circle of interest

☐ from pick-up list

☒ Configuration

Circle

Configure position

100,0 100,0 50,0

MP X MP Y radius

Configure parameters

G170

Histogram

0 255

☐ Global Tolerances ☐ Dynamic evaluation

Tolerance

Nominal value 0,0 + 0,0 - 0,0

☐ Warning limit:

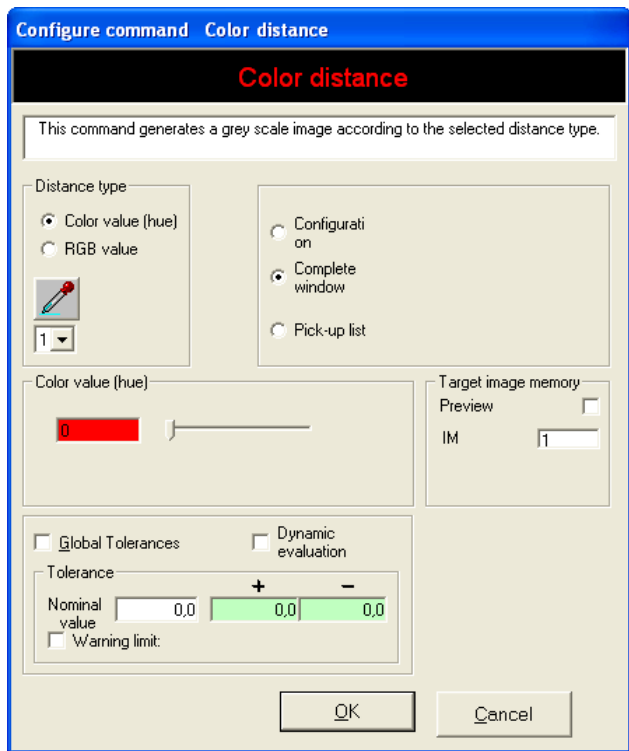
☐ Test

此命令用于在配置的圆上寻找边缘的转换并计算他们。边缘的位置保存到点寄存器。

兴趣圆 Circle of interest

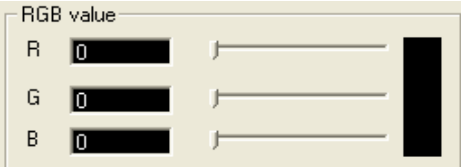
来自选取列表（From pick-up list）	从选取列表选择一个圆。
配置（Configure）	通过直接输入数字或交互选择配置兴趣圆。
配置参数（Configure parameters）	

颜色距离（Color distance）



此命令生成一个灰度图像，在图像上确定到参考色值的距离。

距离类型（Distance type）

距离值（色调）（Color value (hue)）	确定一个参考色值。可手动设置色值或用滴管选择色值，之后在相机取景器中点击满意的色值。自动选择了新的颜色。 
RGB value R G B RGB value 	通过到 RGB 值的距离生成灰度值图像。可以通过滑块或手动输入数值确定该值。左图中的黑色位置将显示出颜色对应的数值。 

	如果选择滴管，可在相机取景器中的图像上移动，并点击理想的颜色。通过下拉框选项中的数字定义颜色扩展特性，可扩展 N 个像素。
配置（Configuration）	在选定的窗口确定参考颜色。颜色与参考颜色相差越大，显示的窗口越暗。反之，越接近参考值颜色，显示的窗口越亮。
整个窗口（Complete window）	将整个窗口生成灰度图像。
选取列表（Pick-up list）	从选取列表中的值生成灰度图像。

目标图像存储（Target image memory）

在 IM0...IM N 中显示目标图像。

容差（Tolerances）

色彩评价（Color evaluation）

Configure command Color evaluation

Color evaluation

The number of pixel in the selected color, saturation and brightness range will be evaluated. Additionally the min., max., and average values will be put on the register.

 Hue range (0 ... 360) Saturation range (0 ... 255) Value range (0 ... 255)

1 0 to 180 0,2 to 1 0,2 to 1

Implementation

☐ Floating point ☒ Integer Information

Window position

100 100 100 100

X Y W H

☐ Global Tolerances ☐ Dynamic evaluation

Tolerance

Nominal value 0,0 + 0,0 - 0,0

☐ Warning limit:

☐ Use world coordinates

Colorize pixel

☒ In original ☐ In grey scale image

☒ good

☐ bad

Ok Cancel

此命令用于评估选定颜色的像素数量。可定义颜色，饱和度 and 亮度范围。图像的理想颜色可通过滴管选定。

	如果选择滴管，可在相机取景器中的图像上移动，并点击理想的颜色。通过下拉框选项中的数字定义颜色扩展特性，可扩展 N 个像素。
色彩范围（Color range）	颜色范围可由数值确定，或者由鼠标左键点击打开颜色选择表。
饱和度范围（Saturation range）	在此设置颜色饱和度。可直接输入数值或通过颜色选择表调整饱和度。
亮度范围（Brightness range）	在此设置亮度值。可直接输入数值或通过颜色选择表调整亮度值。

实施（Implementation）

选择实施色彩评估。

浮动点（Floating point）	浮动点使用标准化值范围。颜色范围值介于0-360之间，饱和度范围介于0-1，强度范围介于0-1。
整数（Integer）	整数使用标准化值范围。颜色范围值介于0-360之间，饱和度范围介于0-255，强度范围介于0-255。

窗口位置（Window position）	通过直接输入数值或点击窗口位置确定评估窗口。
-----------------------	------------------------

容差（Tolerance）	
---------------	--

位置追踪（Position tracing）	如果激活这个选项，将追踪评估的区域。
------------------------	--------------------

色彩像素（Color pixel）	如果激活这个选项，匹配的像素将显示为彩色。
-------------------	-----------------------



彩色图像区域（Color image area）

Configure command Color defined area

Color image area

Selected region will be colored with chosen color

Circle

☐ from Pickup-list
 from register
☐ (Radius from value register, Midpoint from point register)
☒ Configuration

Circle

Configure position

100,0 100,0 50,0

MP X MP Y Radius

Options

☒ Inside
☐ Outside

Fill color

127

OK Cancel

在定义的圆内或圆外标记成指定颜色。

圆形（Circle）

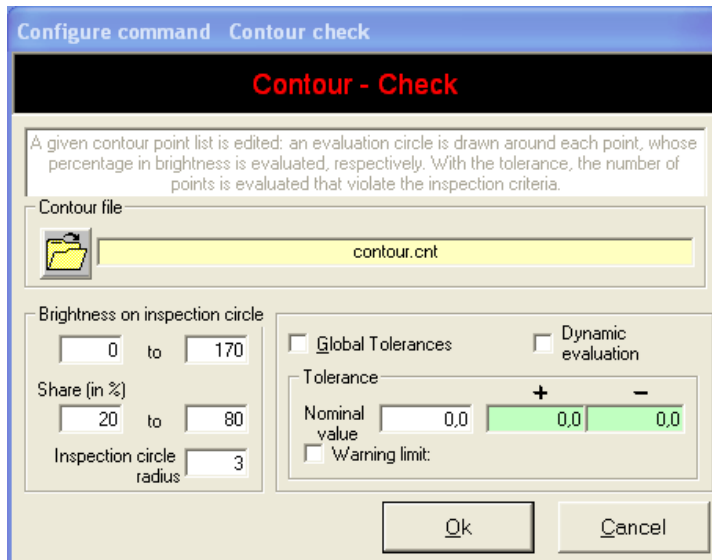
来自寄存器/选取列表（From register / from pick-up list）	根据不同的选项，从寄存器或选取列表中选择圆心和半径。
配置（Configuration）	如果手动配置，将激活配置选项表。

选项（Options）

内部（inside）	确定是否要填充颜色到圈的内部。
外部（outside）	确定是否要填充颜色到圈的外部。
涂色（Fill color）	选取1.. 255间的颜色涂色。



边缘检查（Contour check）

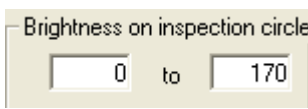
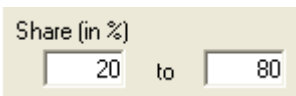
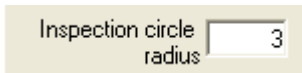


对要检测的对象加载已有的轮廓点，检查对象轮廓的每个点的亮度是否符合标准。评估那些不符合标准的点。

边缘文件（Contour file）

	可在此选择已有的对象轮廓文件。
---	-----------------

检查圆的亮度（Brightness on inspection circle）

	指定要检查的圆的亮度。
	指定要检查的圆的亮度百分比。
	指定要检查的圆的半径。

容差（Tolerances）

可通过轮廓追踪命令产生一个轮廓文件，并用保存点列表菜单保存创建的点列表。关于轮廓追踪命令的功能可参看相应的描述。



边缘追踪（Contour tracing）

Configure command Contour tracing

Contour tracing

Starting from the last determined point the contour-line of a given window is determined. Then the number of located

Window position

Window

X Y W H

0 0 100 100

☐ Use world coordinates

Parameter

☐ Automatic brightness adaptation

Threshold value 170

Max number of contour points 100000

☐ Global Tolerances ☐ Dynamic evaluation

Tolerance

Nominal value 0,0 + 0,0 - 0,0

☐ Warning limit:

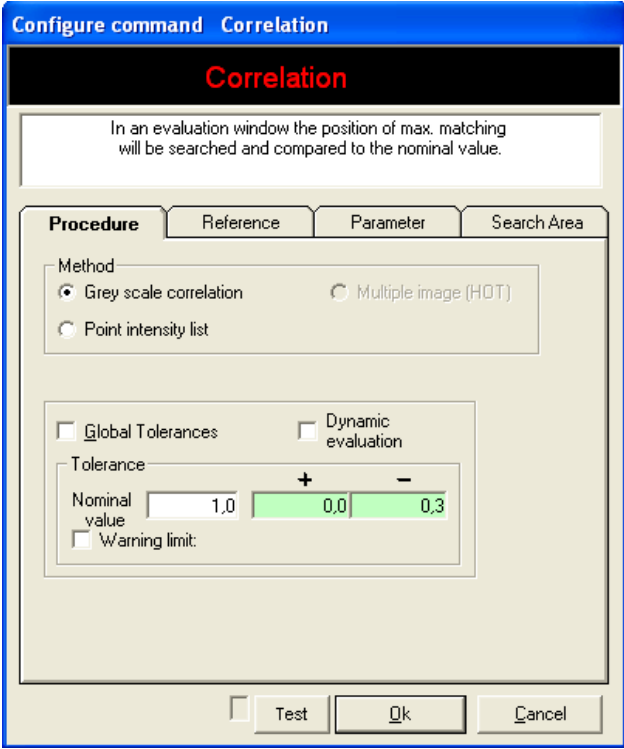
☐ Test

用此命令追踪对象的外部边缘，返回点数量的值。如果之前有重置点列表，那么属于对象边缘的点将位于点列表并被保存。此命令需要一个对象边缘追踪的起始点。这一点可由如探头定义。

窗口位置（Window position）



相关性（Correlation）



相关命令用于在搜索表中寻找已经配置的图案。如果探测到图案就报告其位置。评估与配置图案相关性。根据硬件不同，有不同的相关方法。Intel 或 AMD 处理器，如 Sony 智能相机，另外提供了实时相关性，可以在任何旋转的角度找到对象。

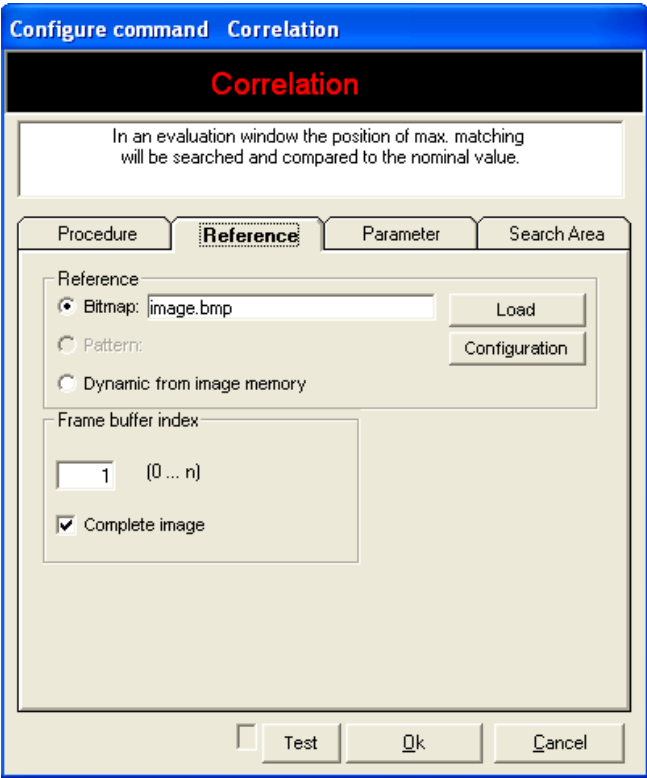
方法（Method）

灰度相关性（Gray scale correlation）	用相关方法可以比较检查图像与参考图像的灰度值相关性。如果与参考图像匹配在一定偏差之内，此命令设置成 OK。在对比之前要先设定一个参考图像，定义搜索范围、参考值和容差值。
点强度列表（Point intensity list）	用此方法在参考图像上编辑出图形，或加载已有的图形。将用此图形与检查区域对比，如果有相关性则设置为 OK。编辑图形时，需要设定更多的参数。

容差（Tolerance）

相关建立在不同的方法上，这些方法根据硬件不同而不同。不是所有方法都包括在标准操作手册里。

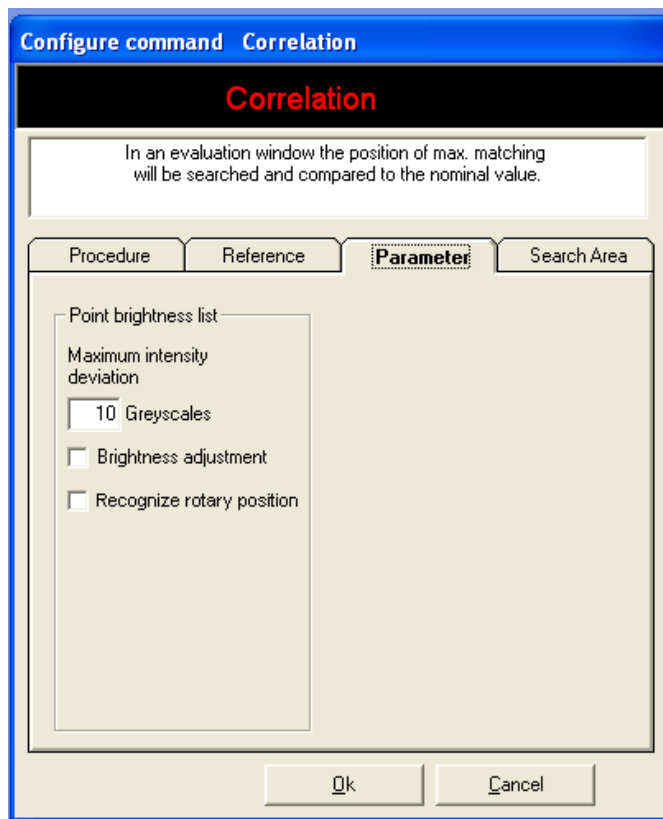
参考（Reference）



在寄存器卡中，用户可确定要加载的参考图像来源。可直接选择加载位图，或从图像内存中动态选择。之后，可在图形编辑器中编辑参考图形。

重要：要指出的是参考图像和要检查的图像不能保存在相同的图像内存中。因此，要在自由图像内存中选择图像浮动序号。

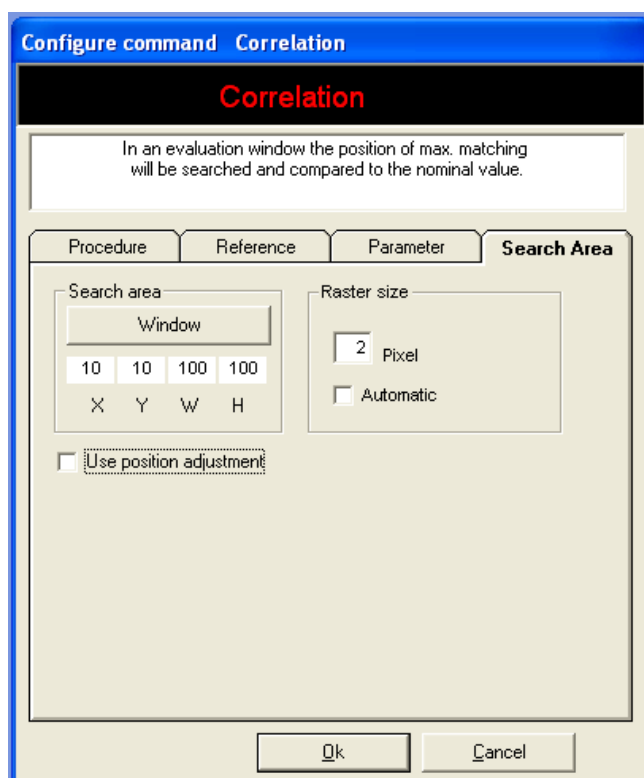
位图（Bitmap）	根据不同的方法加载位图或在图形编辑器中创建图形。
动态来自图像内存（Dynamic from image memory）	另一种方法是动态创建图形，如，从图像内存中读取图像。在此图像中，标记要检查的点，创建一个图形。
帧缓冲索引（Frame buffer index）	通过配置窗口，用户可确定用整个图像或是图像的一部分作为参考。用此功能用户可确定导入的图像的参考区域和特定图形的检查区域。

参数 (Parameter)

此选项卡不适用于灰度值相关。点亮度列表方法提供了两种创建参考图形的方法。点光栅功能与图形编辑器功能相对应。就此点栅格以一定距离布满整个图像，此时使用的光栅宽度仍可调整。

如果激活边缘选项，通过输入的阈值可在参考图像上用点搜索和标记边缘，创建图形。在此也会给出光栅尺寸。

最后，可选择亮度调整，定义允许的最大亮度偏差。请注意对每个图像做这个设置是没有意义的。如果图像是黑白的，这样做可能带来负面的效果。只有当图像有很多不同的灰度值时，获得提高亮度的图像会节省检查。

搜索区域（Search area）

定义搜索图形的搜索范围和搜索粒度。

搜索区域（Search area）	通过直接输入或配置窗口定义搜索范围。
栅格尺寸（Raster size）	栅格尺寸表示原始图形与参考图形对比的像素距离。

注意：用栅格尺寸5去搜索10*10的对象是没有意义的。移动选择的点可看到连贯性，稍微超过示例模型以观察相关性。

图形越大，栅格尺寸可能越大。但是栅格尺寸越大，越有可能在错误的地方找到图形或根本找不到图形。

对于栅格没有一定之规，因为每个检测图形都是不同的。要提高速度，用户必须用不同的值去实践。



特殊相关性（Correlation special）

在评价窗口中搜索最大匹配位置并与标称值比较。根据外轮廓生成新图像的矢量列表。

列名称（List name）

点击“**No file selected**”文本框选择相关列表。

分隔符（Separator）

用分隔符将每个区域分开。

匹配（Match）

在此为匹配指定标称值和容差。

搜索范围（Search area）

栅格尺寸（Raster size）

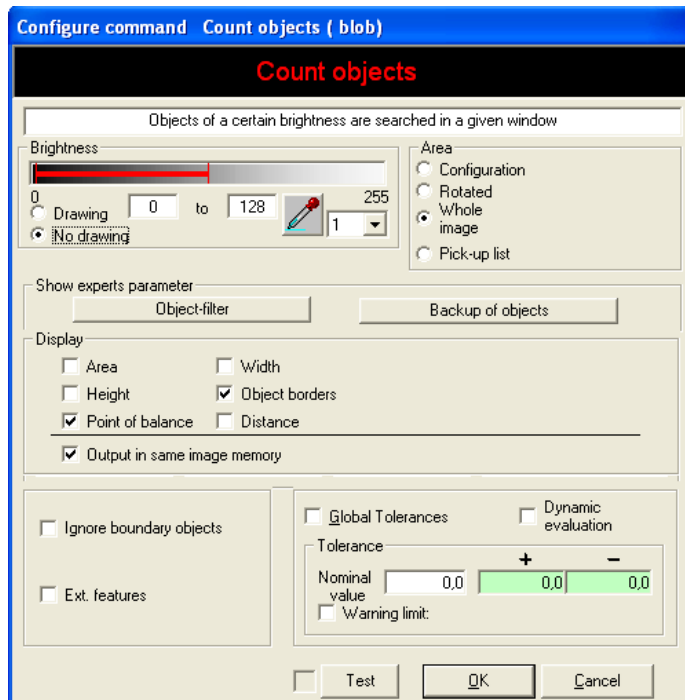
在此，有一定距离的点栅格移过整个图像，在此情况下可调整使用的栅格尺寸。

点亮度列表（Point brightness list）

可定义最大灰度值容差，也可以调整亮度。



计数（Count objects (blob)）



计数功能是非常通用的命令。基于图像不同区域有不同的亮度。通过输入的范围或用滴管选择，可确定要分离对象的颜色。可通过检查（Test）按钮确定是否找到了正确的对象。之后可以找到所有相同颜色的对象。如果找到的对象太多，可使用对象过滤（Object filter）减少结果。

例如，如果只想找到1400像素的对象，可以在对象过滤中使用最小值1300和最大值1500。因此，只会找到符合要求的尺寸的对象。重要的是如果要使用过滤功能，必须首先激活过滤。

亮度（Brightness）

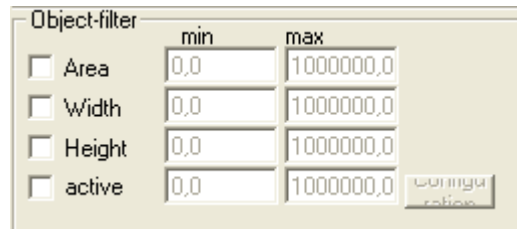
可以直接输入值或用滴管确定亮度范围。

用滴管可直接在图像上选取亮度。

用下拉菜单中的数字定义灰度值扩散的像素特性。

区域（Area）

配置（Configuration）	配置要搜索的对象区域。
旋转（Rotated）	在此区域内要搜索的图像是旋转的。在此情况下，可用位置追踪旋转窗口。
整幅图像（Whole image）	使用整个图像用于搜索对象。
选取列表（Pick-up list）	

显示高级参数 (Show expert parameters)对象过滤 (Object filter)


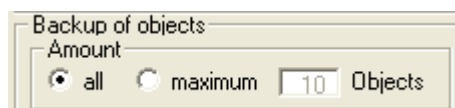
The 'Object-filter' dialog box contains a table with four rows: 'Area', 'Width', 'Height', and 'active'. Each row has a checkbox on the left and two input fields labeled 'min' and 'max' on the right. All input fields are currently set to '0,0' and '1000000,0' respectively. A 'Configure' button is located at the bottom right of the dialog.

	min	max
☐ Area	0,0	1000000,0
☐ Width	0,0	1000000,0
☐ Height	0,0	1000000,0
☐ active	0,0	1000000,0

面积 (Area)	如果选中此项，只输出面积介于最小值与最大值间的对象。.
宽度 (Width)	如果选中此项，只输出宽度介于最小值与最大值间的对象。
高度 (Height)	如果选中此项，只输出高度介于最小值与最大值间的对象。
距离中心的距离 (Distance from center)	如果选中此项，只输出距离中心距离介于最小值与最大值间的对象。

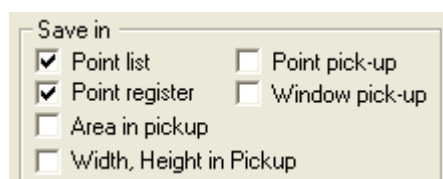
备份对象 (Backup of objects)

因为对象命令没有定义要输出的对象数量，所以可用此选项定义要转移到如点列表的对象数量。



The 'Backup of objects' dialog box has a section titled 'Amount' with two radio buttons: 'all' (selected) and 'maximum'. Next to the 'maximum' radio button is a text input field containing the number '10' and the label 'Objects'.

数量 (Amount)	在此确定要保存的对象结果数量。
-------------	-----------------

保存（Save in）


Save in

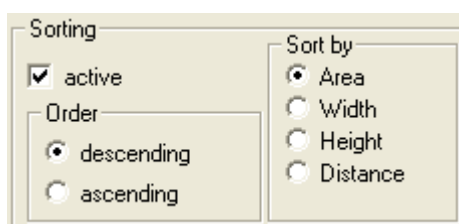
☒ Point list ☐ Point pick-up

☒ Point register ☐ Window pick-up

☐ Area in pickup

☐ Width, Height in Pickup

在此选择将结果保存到哪里，如保存到选取列表。

排序（Sorting）


Sorting

☒ active

Order

☒ descending

☐ ascending

Sort by

☒ Area

☐ Width

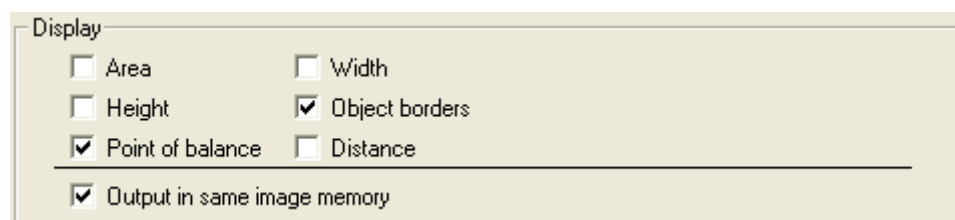
☐ Height

☐ Distance

在此确定对象如何排序及排序方式。

显示（Display）

这里显示的值可用于输入到对象过滤中用于作为选择界限。



Display

☐ Area ☐ Width

☐ Height ☒ Object borders

☒ Point of balance ☐ Distance

☒ Output in same image memory

面积（Area）	显示对象面积。
高度（Height）	显示对象高度。
宽度（Width）	显示对象宽度。
距离（Distance）	显示对象到中心的距离。
平衡点（Point of balance）	标记对象的平衡点。
对象界限（Object borders）	标记界限。

忽视边缘对象（Ignore boundary objects）

如果选中此项，与窗口边缘颜色相同的对象将被删除。

扩展属性（Ext. features）

如果选中此项，打开对象过滤（*Object-filter*），显示扩展选项，包括“角度”，“偏心率”，“长轴”（如下图）。

扩展属性—对象过滤（Ext. features – Object-filter）

角度（Angle）	如果选中此项，只输出角度介于最小值与最大值间的对象。.
偏心率（Excentricity）	如果选中此项，只输出偏心率介于最小值与最大值间的对象。.

扩展属性—显示（Ext. features – Display）

长轴（Mj. Axis）	显示对象的长轴。
角度（Angle）	显示对象的角度。
偏心率（Excentricity）	显示对象的偏心率。

容差（Tolerances）



横截面（Cross-section）

Configure command Cross-section

Cross-section

The cross-section of an object with the selected color is determined along a taught-in probe. The determination is made with the threshold-value-edge-method.

Probe

Probe position: ☐ dynamic position

Probe parameter: **G170**

Histogram

0  255

☐ Global Tolerances **ME** ☐ Dynamic evaluation

Tolerance

Nominal value:

☐ Warning limit:

在配置窗口确定对象的横截面。

探头位置（Probe position）

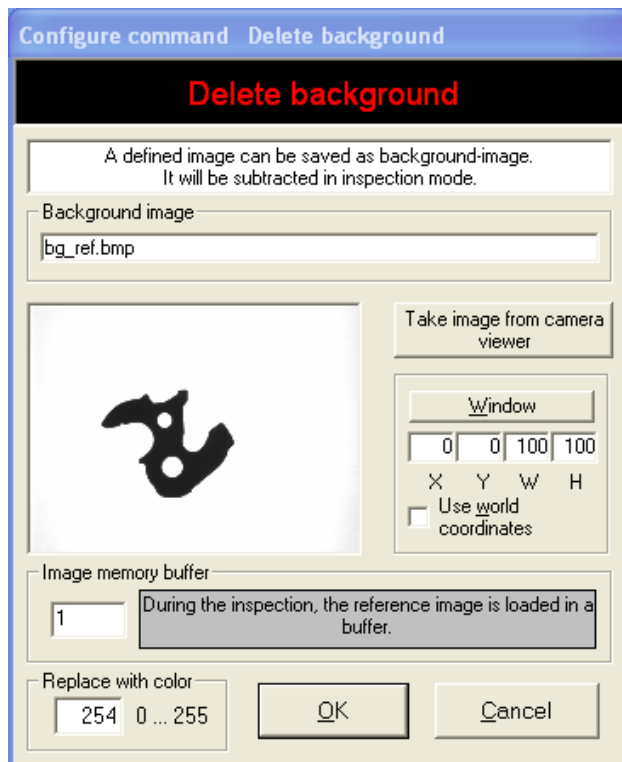
配置搜索线（Configure search line）	配置搜索线的位置。
------------------------------	-----------

探头参数（Probe parameters）

配置探头参数（Configure probe parameters）	定义测定边缘的参数。
------------------------------------	------------



删除背景（Delete background）



该命令从当前图像中减去一个预先配置的图像。

背景图像（Background image）	指定一个图像做背景。
从相机取景器中选取图像（Take image from Camera Viewer）	从相机取景器中选取将被用作背景的图像。
窗口（Window）	配置一个要做去除的窗口。
图像内存缓冲（Image memory puffer）	用于参考图像的图像内存缓存数。
用彩色替换（Replace with color）	在此区域内由颜色代替背景图像。



确定曲线的点（Determine point of curve）

Configure command Determine point of curve

Determine point of curve

The first probe defines the reference edge, the second one defines the curve point.

Rz-value (~) Repetition

Probe parameter

G170

Histogram
 

Histogram
 

☐

此命令多应用于要确定直线到弯的情况。此任务通常发生在预置工具时。

Rz-value	表示表面粗糙，不应该算做弯曲。
重复（Repetition）	表示应执行的重复次数。
配置探头参数（Configure probe parameters）	
配置参考边缘（Configure reference edge）	在此，用探头为要探测的临界点配置参考边缘。
配置搜索区域（Configure search area）	在此，用探头配置要探测临界点的区域。



距离（Distance）

Configure command Distance

Distance

Determines and evaluates the distance according to selected options

Distance determination

☒ Point Point ☐ Point Straight Line ☐ Point Circle

First point

☒ From register Index ☐ From pick-up list

Second point

☒ From register Index ☐ From pick-up list

☐ Global Tolerances ME ☐ Dynamic evaluation

Tolerance

Nominal value

☐ Warning limit:

Distance type

☒ direct ☐ axially parallel in X ☐ axially parallel in Y

Ok Cancel

根据不同的选项，测量两点间的距离。

测量距离（Distance determination）

点到点（Point Point）	测量两点间的距离。在距离类型（ <i>distance type</i> ）中定义怎样测量。
点到直线（Point Straight line）	测量点到直线的距离。在距离在距离类型（ <i>distance type</i> ）中定义怎样测量。
点到圆（Point Circle）	由于圆心的点已经在点寄存器中，因此已经自动显示在点到点的测量中，此处显示为灰色未激活。

第一点/第二点（First point/ second point）

来自寄存器（From register）	
来自选取列表（From pick-up list）	

距离类型 (Distance type)

直接 (Direct)	测量直接距离。
平行 X 轴 (Axially parallel X)	测量平行于 X 轴的距离。
平行 Y 轴 (Axially parallel Y)	测量平行于 Y 轴的距离。

评估结果 (Result evaluation)



到直线的距离（Distance to a straight line）

Configure command Distance to a straight line

Distance to a straight line

Measures the distance between a detected edge and a straight line.

Distance type

☒ Average value ☐ Minimum ☐ Maximum

Reference straight line

☒ from register ☐ from pick-up list

Index: 0

Probe

Configure position Configure parameters

☐ dynamic position

☐ Dispersion filter: **G170**

Histogram

0 255

☐ Global Tolerances ME ☐ Dynamic evaluation

Tolerance

Nominal value 0,0 + 0,0 - 0,0

☐ Warning limit:

OK Cancel

测量卡尺间或配置的直线间的距离。

距离类型（Distance type）

平均值（Average value）	测量卡尺到参考直线间的平均值。
最小值（Minimum）	测量卡尺到参考直线间的最小值。
最大值（Maximum）	测量卡尺到参考直线间的最大值。

参考直线 (Reference straight line)

参考直线来自寄存器或选取列表。

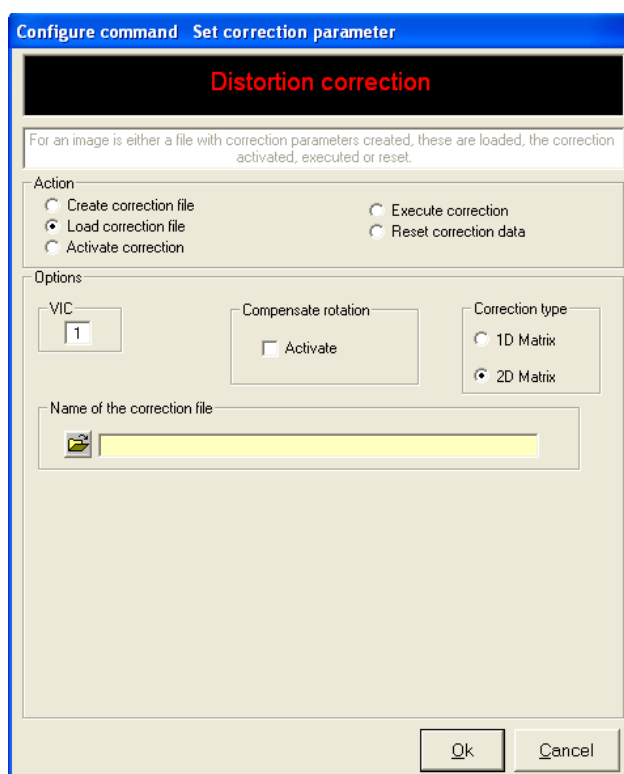
卡尺 (Caliper)

配置位置 (Configure position)	配置卡尺。
配置参数 (Configure parameters)	配置卡尺参数，如对象颜色，探测方法。

容差 (Tolerance)



扭曲相关（Distortion correction）



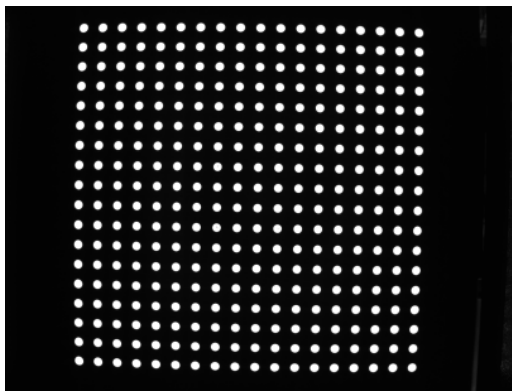
此命令用于纠正由角度错误或镜头畸变引起的图像失真。

要纠正失真必须先执行以下操作：

- **创建修正文件**（在单独的检查程序中创建一次性的修正文件）。
- **加载并激活修正文件**（此步骤可在开启 SmartBuilder 时自动执行）。
- **执行修改**（在检测程序中执行此步骤）。

创建修正文件（Create correction file）

在检查程序中通过校准标准创建修正文件。*.cor 文件用于图像修正。要创建修正文件，只能执行一次检查程序。



在此，通过图像获取载入图像校准标准。要重置点列表并以校准标准搜索对象。（如使用“计

数”命令)。找到的对象坐标被置于点列表。

插入失真修正命令后，要做以下调整：
选择**创建修正文件**选项。

VIC（视频输入频道） (Video Input Channel)
获取图像的相机数量，默认 VIC=1.

补偿旋转（Compensate rotation）
如果有旋转，可激活此选项。

修正类型（Correction type）
1D 矩阵的行扫描相机。
2D 矩阵的面扫描相机。

修正文件名称（Name of correction file）
保存修正文件的名称，如 name.cor.

在修正文件 “name.cor”中保存支持点，修正向量。文件保存在 Devices\PC_Local\Data\。

加载修正文件（Load correction file）

要加载修正文件，必须单独加载失真修正命令。这个步骤可以通过作为程序的一部分或作为子程序执行。也可以通过启动 SmartBuilder 实现。

选择加载修正文件选项

VIC 视频输入频道 (Video Input Channel)
适应输入以创建文件。

补偿旋转（Compensate rotation）
适应输入以创建文件。

修正类型（Correction type）
适应输入以创建文件。

修正文件名称（Name of the correction file）
适应输入以创建文件。

激活修正（Activate correction）

要加载修正文件，必须单独加载失真修正命令。这个步骤可以通过作为程序的一部分或作为子程序执行。也可以通过启动 SmartBuilder 实现。

选择选项，激活修正。

VIC 视频输入频道 (Video Input Channel)
适应输入以创建文件。

补偿修正 (Compensate correction)
适应输入以创建文件。

修正类型 (Correction type)
适应输入以创建文件。

修正

标准选项是“修正小图像或部分图像”。此选项可进行快速修正，但需要更多磁盘空间。“修正大图像”处理得较慢，但需要较少磁盘空间。

指定采样点的距离

自动测量此值，如果需要可手动调整。

执行修正 (Execute correction)

选择选项“**执行修正**”

VIC 视频输入频道 (Video Input Channel)

适应输入以创建文件。

目标图像内存 (Target image memory)

选择要保存修正图像的图像内存。如果同时选择了程序编辑器中的图像内存输入和目标图像内存输入，原始内存将被覆盖。

修正范围 (Correction range)

有三种选项可提高执行速度：“整幅图像”，“部分图像”，或“蒙面”。

未修正的像素 (Uncorrectable pixel)

在修正范围外的像素未被修正或设置成灰度值（默认为0）。

重置修正数据 (Reset correction data)

用此选项重置修正数据。

选择选项“**重置修正数据**”

VIC 视频输入频道 (Video Input Channel)

适应输入以创建文件。

动作 (Action)

创建修正文件 (Create correction file)	在此创建修正文件。
加载修正文件 (Load correction file)	在此加载已经存在的修正数据。

激活修正 (Activate correction)	在此确定要修正的内容，如大图像。
执行修正 (Execute correction)	在此确定要在哪里执行修正，如整幅图像。
重置修正数据 (Reset correction data)	在此重置所有相关数据。

选项 (Options)

加载/创建修正数据 (Load/ create correction data)

VIC	指定相机的视频输入频道。
补偿旋转 (Compensate rotation)	纠正一个旋转的扭曲图像。

修正类型 (Correction type)

1D 矩阵	获取一维点。
2D 矩阵	获取二维点。
修正文件名称 (Name of the correction file)	在此加载已经存在的修正点数据或创建新的修正数据。

激活修正 (Active correction)

Correction of

☒ small images or image parts

☐ large images

Correction type

☐ 1D Matrix

☒ 2D Matrix

Specified distance of sampling point

Specified horizontal distance: 0

Specified vertical distance: 0

Start X 0 Start Y 0

End X -1 End Y -1

Re-calculate

修正 (Correction of)

小图像或部分图像 (Small images or image parts)	修正小图像或部分图像。
大图像 (Large images)	修正大图像。

指定采样点的距离 (Specified distance of sampling point)

指定水平距离 (Specified horizontal distance)	在此指定采样点的水平距离。
指定垂直距离 (Specified vertical distance)	在此指定采样点的垂直距离。
X 起点 (Start X)	X 坐标的起点。
Y 起点 (Start Y)	Y 坐标的起点。
X 终点 (End X)	X 坐标的终点。
Y 终点 (End Y)	Y 坐标的终点。

执行修正 (Execute correction)

Options

VIC Target image me

Correction range

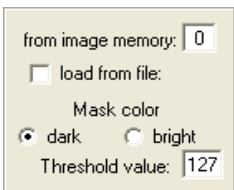
☒ Whole image
☐ Part of image
☐ as masked

Uncorrectable pixel

☐ Keep unchanged
☒ Set to following grey value:

VIC	指定相机的视频输入频道。
目标图像内存 (Target image memory)	指定目标图像内存, 在此显示结果 IMO....IMN。

修正范围 (Correction range)

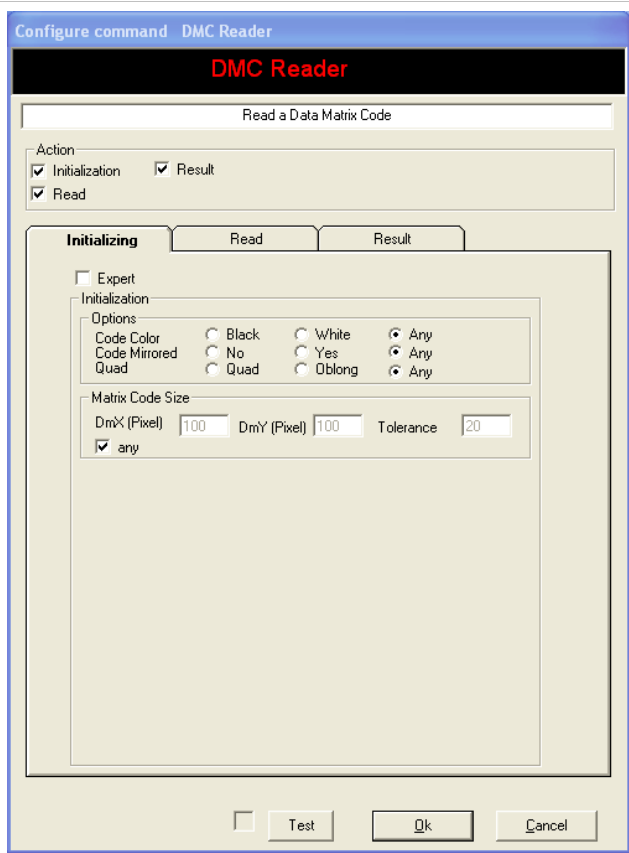
整幅图像 (Whole image)	修正整幅图像。
部分图像 (Part of image)	用户定义图像的一部分。
蒙面 (as masked) 	在此定义蒙面。在蒙面基础上执行修正。

未修正像素 (Uncorrectable Pixel)

保持未改变 (Keep unchanged)	不改变未修正的像素。
设置灰度值 (Set to following gray value)	将未修正的像素设置成指定的灰度值。



数据矩阵码阅读器（DMC Reader）



数据矩阵码阅读器特别用于二维码。它自动在图像上搜索二维码并显示结果。可通过设置搜索区域和设置选项提高其读取速度。

执行（Action）

如果选择所有三个选项，读取将被初始化，搜索并读取。

初始化（Initialization）	初始化系统。
读取（Read）	在图像搜索编码并读取。
结果（Result）	显示读取结果或错误的编码。

初始化选项

编码颜色（Code Color）	未定义编码颜色，系统自动识别颜色。
编码镜像（Code Mirrored）	不需要定义。系统自动识别编码是否有镜像。
四（Quad）	不需要定义。系统自动识别编码是四或正方。

读取（Read）

Initializing **Read** Result

Read

Data Matrix Code will be searched in the specified window.

Optionally the print quality information according to ISO 15415 and DPM-1-2006 will be read.

Area

☐ Configuration

☒ Complete Image

☐ Use Worldcoord.

☒ Print quality info

根据初始化读取编码

区域（Area）

配置（Configuration）	用户可配置要搜索的区域。
整幅图像（Complete Image）	在整幅图像中搜索编码。
使用完全坐标（Use World coordinates）	如果配置了搜索区域，可通过位置追踪移动搜索区域。

结果（Result）
值（Values）

字符数量（Number of characters）	测试，检查字符数量是否正确。
线的数量（Number of lines）	测试，检查线的数量是否正确。
列的数量（Number of columns）	测试，检查列的数量是否正确。
颜色（Color）	测试，检查字符颜色是否正确。
镜像（Mirrored）	测试，检查是否存在字符镜像。

文本（Text）

文本（Text）	读取的结果被写入全局文本内存。如果不是可打印的字符，可选择 HEX 选项。这样每个读取字节将以 HEX 显示写进文本内存。
错误（Error）	如果选择这个选项，一旦发生错误，错误将被写进文本内存。
编码类型（Code type）	如果选择这个选项，发现的编码将被写进文本内存。

点 (Points)

寄存器 (Register)	显示探测到的拐角点。这些拐角点以两个边界线的交汇点为起点，一个接一个地存储在寄存器中。
显示 (Display)	在叠加层里显示探测到的边角点。

初始化高级选项 (Initializing Expert)

Configure command DMC Reader

DMC Reader

Read a Data Matrix Code

Action

☒ Initialization ☒ Result

☒ Read

Initializing Read Result

☒ Expert

Initialization

Options

Code Color ☐ Black ☐ White ☒ Any

Code Mirrored ☐ No ☐ Yes ☒ Any

Quad ☐ Quad ☐ Oblong ☒ Any

Matrix Code Size

DmX (Pixel) DmY (Pixel) Tolerance

☒ any

Expert configuration

Delta Angle (Degree) Proportion

Closing Filter OppLine (%)

Background Noise NextLine (%)

Module

Number modules in one dir.

ModulNrMin ModulNrMax

Module size [pixel] in one dir.

ModulSizeMin ModulSizeMax

ModulSelect

ModulContrast

Zoom

FindZoomMin FindZoomMax

ReadZoomM ReadZoomMax

☐ Test

高级配置 (Expert configuration)

角度 (Delta Angle (Degree))	每四个数据矩阵角的角度应小于90度+/- Delta 角。
关闭过滤 (Closing Filter)	0 = 无预过滤 >0 关闭数据矩阵“L” 定义数据矩阵颜色 (DmSearchColor 应为 0 或 1, 不能是 -1)  关闭过滤 = 0  关闭过滤 > 0
背景噪音 (Background Noise)	背景噪音水平 (15 = 正常噪音 .. 35 = 强噪音)

比例 (Proportion)

对面的线 (OppLine)	定义对面的线的最大百分比比率。60%是指，对面的线 b 最短也要有长线 a 的60%。 
下一条线 (Next Line)	定义邻近线的最大百分比比例。20%是指邻近的线最小不应短于长线的20%。 

模块 (Module)

最小模块数量 (ModulNrMin)	最小模块数量是8 (最小数据矩阵码)
最小模块尺寸 (ModulSizeMin)	以像素表示的最小数据矩阵模块尺寸。
模块选择 (ModulSelect)	以百分比表示模块探头颜色 (0% = 最黑的颜色 / 100% = 最白的颜色)。
模块对比 (ModuleContrast)	模块与背景在灰度值上的最小对比。(75%表示模块必须有更高的对比)

最大模块数量（ModuleNrMax）	最大模块数量是 144（最大数据矩阵码）
最大模块尺寸（ModulSizeMax）	以像素表示的最大数据矩阵模块尺寸。

变焦（Zoom）

最小搜索变焦（FindZoomMin）	搜索数据矩阵的最小缩放因子。
最小读取变焦（ReadZoomMin）	读取数据矩阵的最小缩放因子。
最大搜索变焦（FindZoomMax）	搜索数据矩阵的最大缩放因子。
最大读取变焦（ReadZoomMax）	读取数据矩阵的最大缩放因子。

显示控制（Display Control）

显示控制的可选命令集是专门为检查 LED，LCD 和 OLED 显示屏设计的。这些命令专门针对七个或更多段的显示屏和矩阵显示检查。

用此指令集，检查显示范围内的背景灯光的亮度分布同样是可以的。对于恒定图像的显示，需要图案控制，它不仅决定是否能够看到图案，也决定了图案是否完整。同样属于这个指令集的有一个特殊的颜色评估指令，可以控制显示的颜色。只有使用的是彩色相机时才会有颜色评估这个指令。



边计数器（直线）（Edge counter (straight line)）

Configure command Edge counter(straight line)

Count edges on straight line

The number of edges is counted on a straight line. Optionally the search straight line is configured via a probe or the probe defines the position of an object edge, along which, in a certain distance, the straight line is positioned.

Edge direction

☒ all
☐ only bright / dark
☐ only dark / bright

Search for edges

☒ along the probe
☐ along the object contour

Probe

☐ dynamic position **G170**

☐ Global Tolerances ☐ Dynamic evaluation

Tolerance

Nominal value

☐ Warning limit:

沿配置的探头计算所有的边。

边缘方向（Edge direction）

所有（all）	探测并计算所有从暗到亮的边和从亮到暗的边。
只有亮/暗（only bright / dark）	只计算和显示从暗到亮的边。
只有暗/亮（only dark / bright）	只计算和显示从亮到暗的边。

搜索边缘 (Search for edges)

沿探头 (Along the probe)	沿配置的探头搜索边。
沿对象边缘 (Along the object contour)	沿对象边缘搜索边。

探头 (Probe)容差 (Tolerances)

评估（Evaluation）

Configure command Evaluation

Evaluation

A value or point is evaluated with tolerances

Variable type

☒ Value

☐ Point

Source

☒ Register

Index: 0

☐ Pick-up list:

☐ Enter nominal value to pickup list

☐ Global Tolerances

ME

☐ Dynamic evaluation

Tolerance

Nominal value

0,0

+

0,0

-

0,0

☐ Warning limit:

OK

Cancel

用容差评估值或点。容差在容差范围中管理。

变量类型（Variable type）

值（Value）	来自寄存器或选取列表的值。
点（Point）	来自寄存器或选取列表的点。

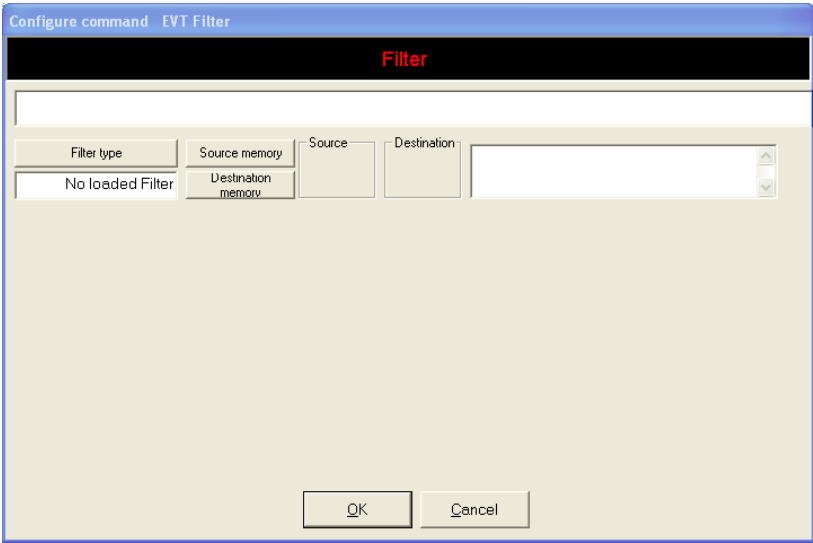
来源（Source）

寄存器（Register）	
选取列表（Pick-up list）	

容差（Tolerances）



过滤工具集 (Filter Tool Set)



通过过滤工具集可以在图像上执行不同的过滤操作。

过滤类型 (Filter type)	用此功能选择一种过滤。后面提供各种过滤的描述。
源内存 (Source Memory)	选择一个显示的初始屏幕 (通常是 IM 0)。
目标内存 (Destination memory)	过滤后的图像显示在不同的图像内存中 (IM1...IMN)。

注释： 所有参数都可通过手动或寄存器的值插入。为了这个目的，许多过滤参数旁有个增加的列。如果此框被选中，用户可以决定在寄存器的哪个位置选取值。

矩阵过滤 (Matrix Filter)

自由矩阵 (Free Matrix)

用户定义矩阵过滤 (浮动点)

源图像

结果图像



参数:

参数	值的范围	描述
常规除数（Norm-Divisor）	0以外的数	用此值除以计算机给出的值。
抵销（Offset）	可选	在除法的结果上增加此值。

对比扩展（Contrast spread）

对灰度值进行缩放变化。用户定义最小和最大灰度值。之间的所有灰度值自动缩放。



参数:

参数	值的范围	描述
最小灰度值（源）（Min. gray-value (Source)）	0...255	源图像的最小灰度值。
最大灰度值（源）（Max. gray-value (Source)）	0...255	源图像的最大灰度值。
最小灰度值（Min. gray-value (destination)）	0...255	目标图像的最小灰度值。
最大灰度值（Max. gray-value (destination)）	0...255	目标图像的最大灰度值。

中位数 (MedianNM)

在给定的邻里，计算每个像素的中位数。

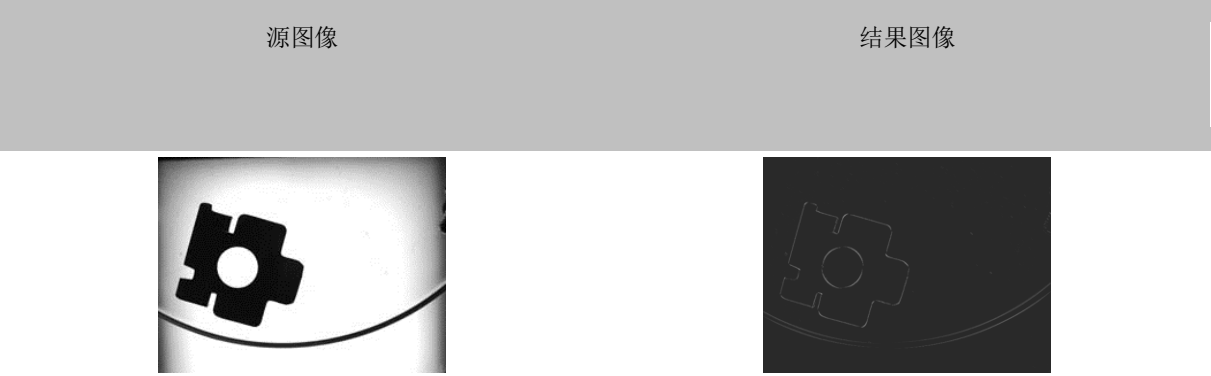


参数:

参数	值的范围	描述
矩阵高度 (Matrix height)	自选	在此确定矩阵尺寸，要包括邻近区域。
矩阵宽度 (Matrix width)	自选	在此确定矩阵尺寸，要包括邻近区域。

Sobel 过滤 (Sobel Filter)

边缘探测过滤。计算 X 和 Y 方向的偏差。



参数:

参数	值的范围	描述
Y 轴的第 N 个步骤	整数 1...	Y 轴的排序次序
X 轴的第 N 个步骤	整数 1...	X 轴的排序次序

注意！参数不可以为0！

变焦立方（Zoom cubic）

该过滤执行图像区域的尺寸变更。配置的窗口通过变焦，变成原图像的尺寸。使用插补法。



缩放 NN

该过滤执行图像区域的尺寸变更。配置的窗口通过变焦，变成原图像的尺寸。使用近邻插补法。



线性缩放（Zoom linear）

该过滤执行图像区域的尺寸变更。配置的窗口通过变焦，变成原图像的尺寸。使用双线性插值法。



扩张过滤（Dilation Filter）

该过滤执行 3 x 3 矩阵的放大，这意味着，黑点被放大了。



参数:

参数	值的范围	描述
重复（Repetition）	任何正数	在此定义扩张过滤重复使用的次数。

侵蚀过滤（Erosion Filter）

该过滤执行 3 x 3 矩阵的腐蚀操作，这意味着，扩大白色的点。

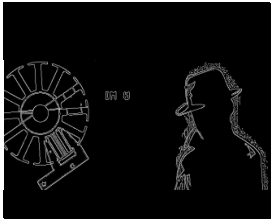


参数:

参数	值的范围	描述
重复（Repetition）	任何正数	在此定义腐蚀过滤重复使用的次数。

Canny 过滤（Canny Filter）

这是个边探测过滤。开始执行一个 Sobel 过滤，之后，在结果中像素值高于阈值上限的设定为255（OK），低于阈值下限的设定为0（NOK）。介于阈值上下限之间的像素，如果直接邻近一个好的像素，那么这个像素 OK，否则 NOK。建议将阈值上限设定为高于阈值下限三倍。

源图像	结果图像
	

参数:

参数	描述
----	----

阈值上限（Upper threshold）

在此可定义阈值，0-255之间。

阈值下限（Lower threshold）

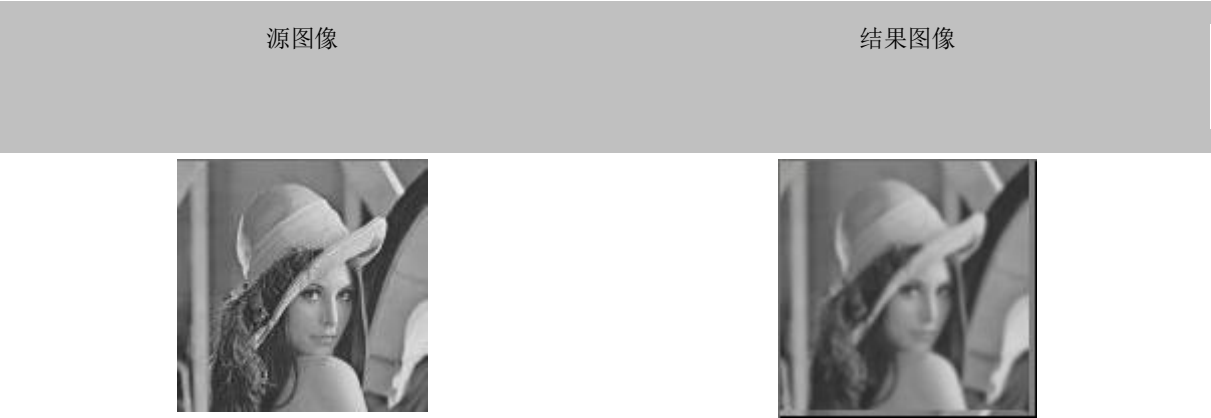
Robert Kreuz

这是个边探测过滤。该过滤执行2 x 2 Robert 矩阵的卷积。

源图像	结果图像
	

高斯5x5过滤（Gauss 5x5 Filter）

执行5 x 5 矩阵的内容平滑或视觉软化效果，使小的结构消失保留大致的结构。



参数:

参数	值的范围	描述
标准差（Standard deviation）	正数	标明图像分歧程度。

高斯3x3过滤（Gauss 3x3 filter）

执行3 x 3 矩阵的内容平滑或视觉软化效果，使小的结构消失保留大致的结构。



参数:

参数	值的范围	描述
标准差（Standard deviation）	正数	标明图像分歧程度。

高斯过滤（Gauss Filter）

执行内容平滑或视觉软化效果，使小的结构消失保留大致的结构。

源图像	结果图像
	

参数:

参数	值的范围	描述
标准差（Standard deviation）	正数	标明图像分歧程度。

其他过滤

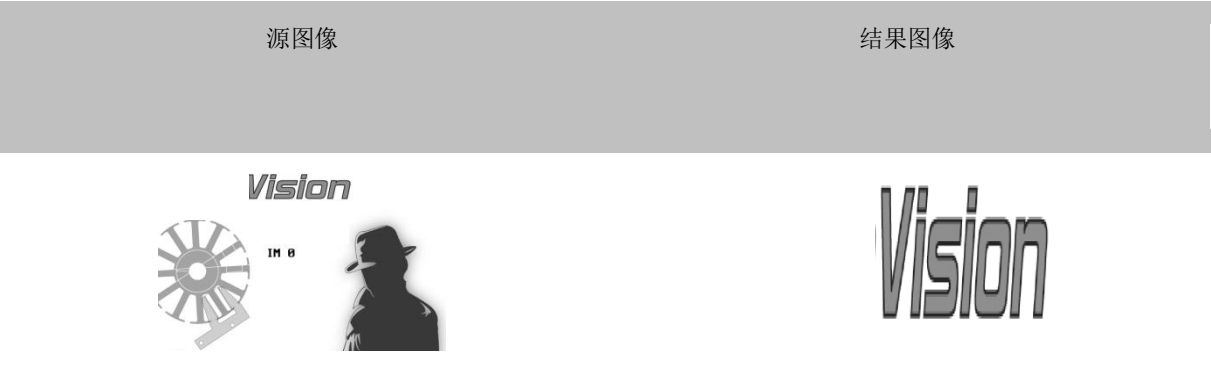
条纹过滤（Stripe filter）

逐行扫描图像，转换成图像亮度。

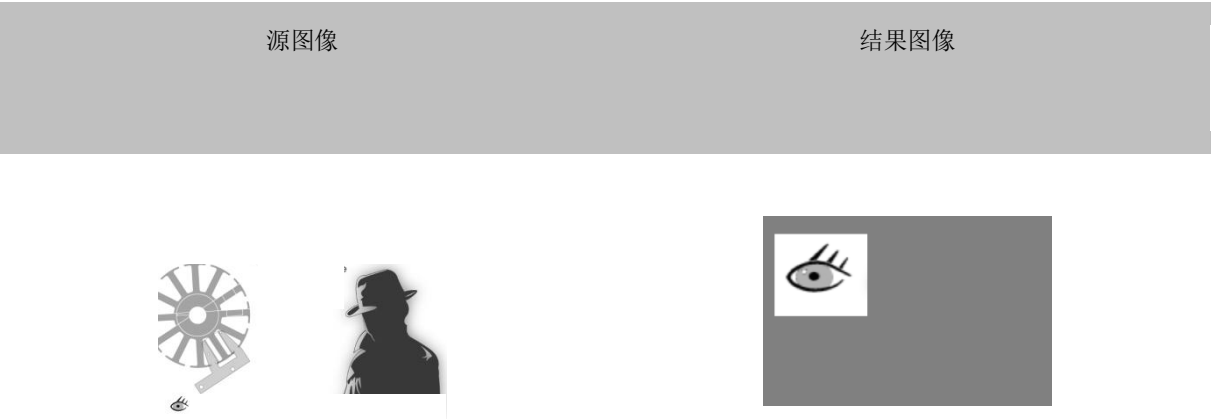
参数:

参数	描述	源图像	结果图像
水平条纹 （Horizontal stripes）	图像亮度分组水平显示。		
明亮条纹（Bright stripes）	图像亮度分组垂直，以亮条纹显示。		

复制部分图像（Copy image regions）
从源图像复制部分图像到目标内存。



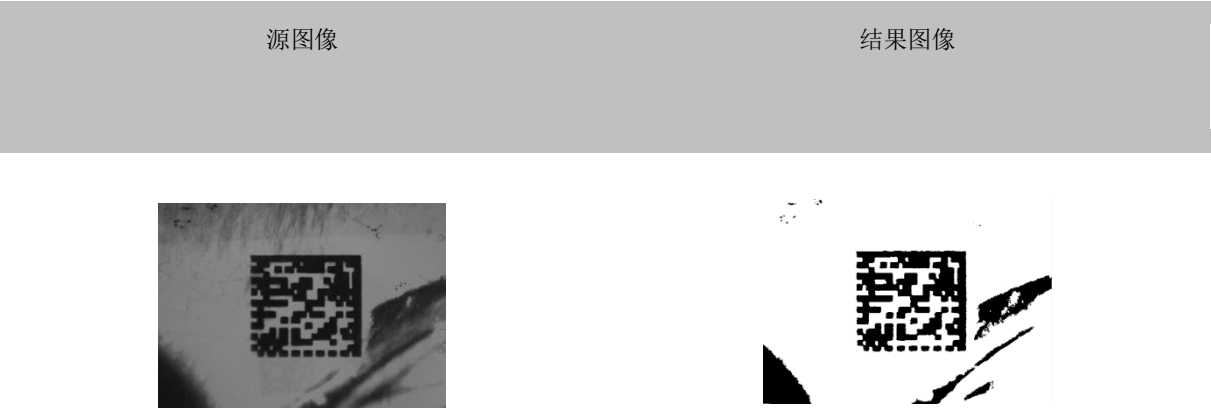
缩放（Zoom）
该过滤可以修改部分图像尺寸。配置的窗口会缩放到要求的尺寸，最适用于抽取无云纹的图像。



参数：

参数	描述
X, Y, 宽度, 高度 (X, Y, width, height)	在此定义缩放图像存储在目标内存的 X 轴和 Y 轴坐标。也可以定义要放大的图像区域的高度和宽度。

Binarisation 矩阵差异（Matrix difference binarisation）
指定高度的列向量在源图像内存中移动。计算其平均像素，然后将像素标记为黑色或白色。如果大于（小于）阈值，则标记像素为白色（黑色）。



参数:

参数	描述
阈值（Threshold）	在 0...255间定义阈值。
高度（Height）	在此定义列向量的高度。

多边形过滤（Polygon filter）

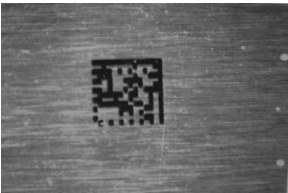
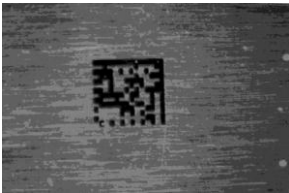
用此过滤可用矩形过滤检测不是矩形的对象。要使用矩形过滤，必须在对象周围使用多边形过滤，并移除背景。



按字节过滤（Bytewise filters）

切位（Cut bits）

此过滤设置最下面的像素字节为0。可事先地定义位数。

源图像	结果图像
	

参数	描述
----	----

位数（Number of bits）要切的位数。

OR 过滤（OR Filter）

在两个图像上执行逻辑位运算或用值结合图像位。

参数:

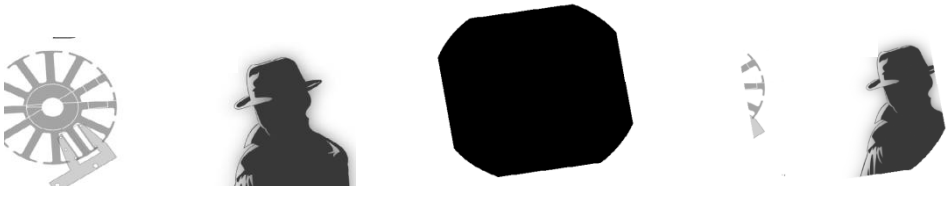
参数	描述	源图像	结果图像
----	----	-----	------

OR 过滤（OR Filter）在两个图像上执行逻辑位运算或用值结合图像位。



参数	源图像 1	源图像 2	结果图像
----	-------	-------	------

使用 2nd 源图像
（ Use 2nd source image）

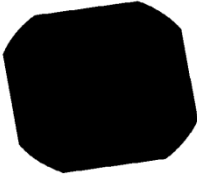


AND 过滤 (AND Filter)

在两个图像上执行逻辑位运算或用值结合图像位。

参数:

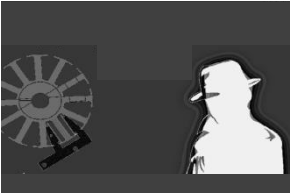
参数	描述	源图像	结果图像
AND 值 (AND Value)	在两个图像上执行逻辑位运算或用值结合图像位。		

参数	源图像 1	源图像 2	结果图像
使用 2 nd 源图像 (Use 2 nd source image)			

XOR 过滤 (XOR Filter)

在两个图像上执行逻辑位运算或用值结合图像位。

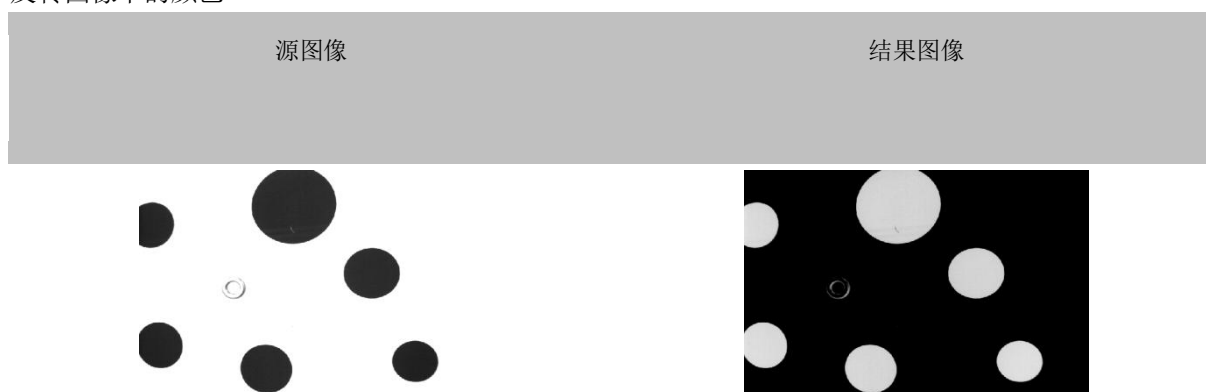
参数:

参数	描述	源图像	结果图像
XOR 值 (XOR Value)	在两个图像上执行逻辑位运算或用值结合图像位。		



反转 (Invert)

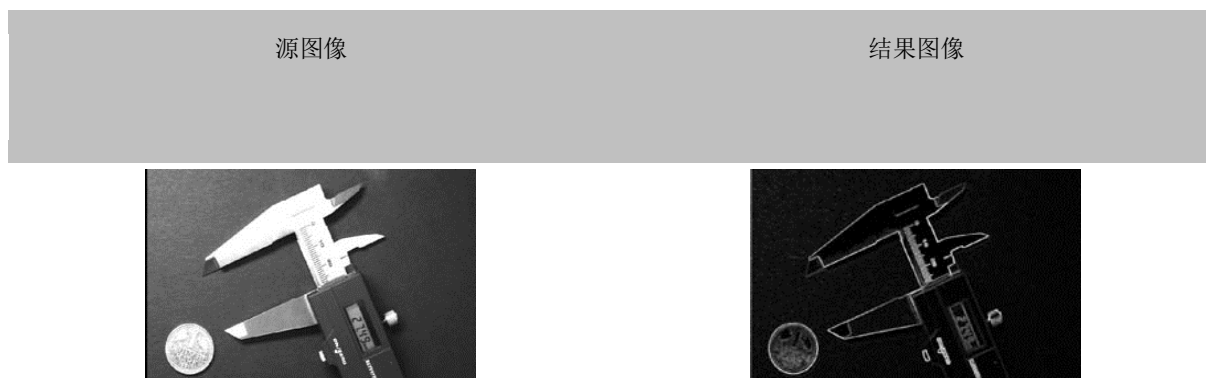
反转图像中的颜色



MMX 过滤 (MMX filter)

对比 (Contrast)

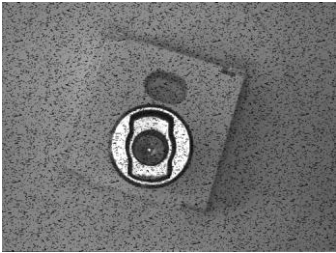
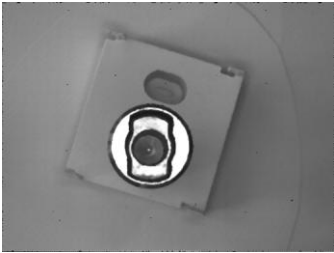
该过滤用每个点邻近的最大对比替换灰度值，因此可使黑白的变化更加明显。



参数	描述
过滤尺寸（Filter size）	显示对比度提高的程度。

中位数（Median）

测量指定区域周围每个像素的中位数。在这个邻里中心的像素根据尺寸排列，返回排列点的中位数，并由此中位数替换中心像素值。这样做可以去除噪音。

源图像	结果图像
	

对比过滤（Comparison filter）

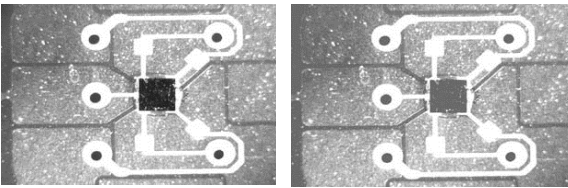
最大（Maximum）

确定两个源图像间的最大像素差，或一个源图像中像素和固定阈值间的最大差距。要做到这一点，必须对指定的区域的像素根据尺寸排列，用排序最大的灰度值代替实际像素。

参数:

参数	描述	源图像	结果图像

阈值（Threshold） 在 0...255 间定义阈值



参数	源图像 1	源图像 2	结果图像
使用 2 nd 源图像（Use 2 nd source image）			

最小（Minimum）

确定两个源图像间的最大像素差，或一个源图像中像素和固定阈值间的最大差距。要做到这一点，必须对指定的区域的像素根据尺寸排列，用排序最小的灰度值代替实际像素。

参数:

参数	描述	源图像	结果图像
阈值（Threshold）	在 0...255 间定义阈值		

参数	源图像 1	源图像 2	结果图像

使用 2nd 源图像 (Use 2nd source image)



转换 (Transformations)

垂直镜像过滤 (Vertical mirror filter)

用图像的垂直轴做图像镜像。



水平镜像过滤 (Horizontal mirror filter)

用图像的水平轴做图像镜像。



旋转 (Rotation)

根据指定的角度旋转图像。

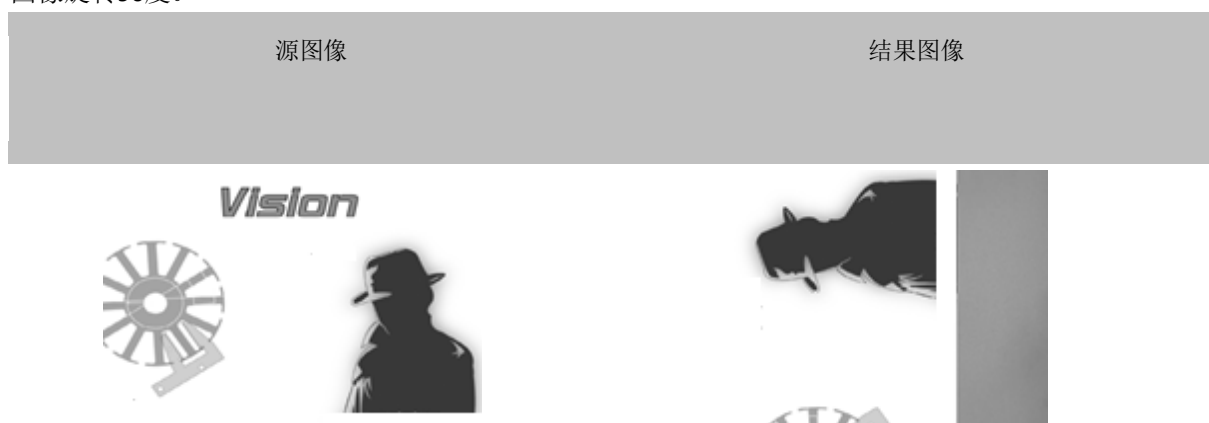


参数:

在此定义图像旋转的角度（如25度），此角度只可以使用整步（见以下说明）。

旋转90度（Rotation 90 degrees）

图像旋转90度。



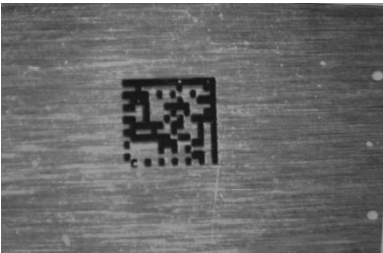
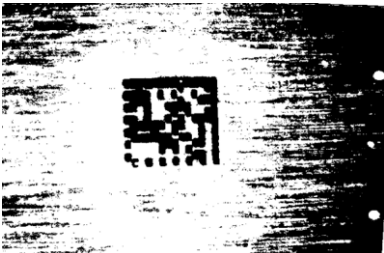
参数:

步 (*90°): 0 步表示图像将旋转 0*90 度，即旋转0度。1 步表示图像旋转 1*90 表示旋转90度等。

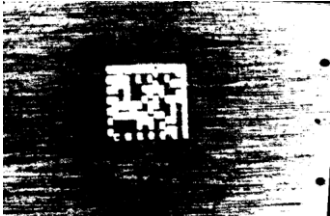
二进制过滤（Binary filter）

二进制过滤（Binary filter）

二进制图像表示像素只能是黑色或白色。高于阈值上限的像素将使用前景色，低于阈值下限的像素使用背景色。

源图像	结果图像
	

参数:

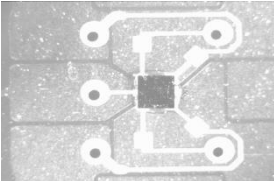
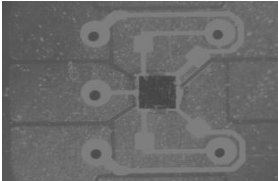
参数	描述	示例图像
前景 (Foreground)	高于 (低于) 阈值上限 (下限) 的值使用前景色 (背景色)。上图中的参数是 255 (前景), 0 (背景)。如果转换图像可得到右图结果。	
背景 (Background)		
阈值上限 (Upper Threshold)	在此定义阈值上限和下限。超过或低于阈值界限时, 图像的前景或背景将涂成灰度。	
阈值下限 (Lower Threshold)		

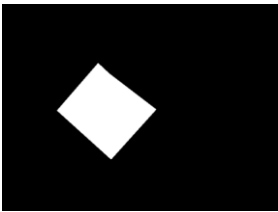
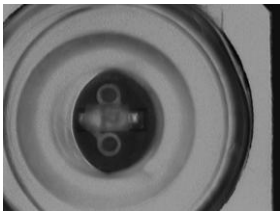
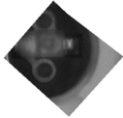
算术过滤 (Arithmetic filter)

除 (Division)

用两个图像互相除, 或者用对比值除以一个图像。

参数:

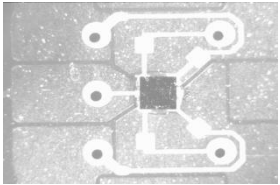
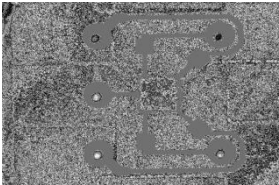
参数	描述	源图像	结果图像
除数 (Divisor)	定义一个要除图像的值。		

参数	源图像 1	源图像 2	结果图像
使用 2 nd 源图像 (Use 2 nd source image)			

乘 (Multiplication)

该过滤器用于两个图像相乘或一个值乘以图像。那么这个值将变成像素乘像素。阈值大于255的都显示为白色。

参数:

参数	描述	源图像	结果图像
乘数 (Multiplier)	定义一个去乘图像的值。		

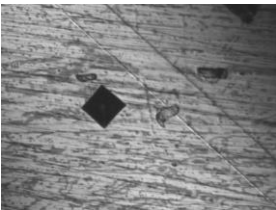
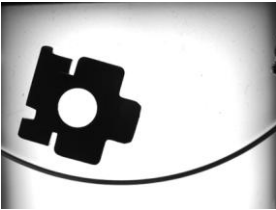
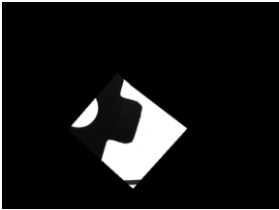
参数	源图像 1	源图像 2	结果图像
使用 2 nd 源图像 (Use 2 nd source image)			

减法 (Subtraction)

该过滤器可用做两个图像的减法或用图像减一个值。

参数:

参数	描述	源图像	结果图像
减数 (Subtractor)	定义要减掉的值。		

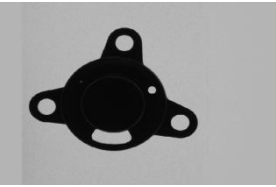
参数	源图像 1	源图像 2	结果图像
使用 2 nd 源图像 (Use 2 nd source image)			

加法 (Addition)

两个图像相加或在一个图像上增加值。

参数:

参数	描述	源图像	结果图像
加值 (Value to add)	定义要增加的值。		

参数	源图像 1	源图像 2	结果图像
使用 2 nd 源图像 (Use 2 nd source image)			

转换（Conversion）

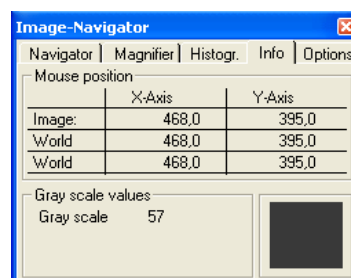
转换成 RGB 图像（Convert to RGB）

将灰度值图像转换成 RGB 图像。使用这个过滤后，理想的情况是，用户仍旧有灰度值图像。但图像实际上被分成了三个通道（RGB），尽管用户不能直观地看到。

我们在下面显示过滤的效果。

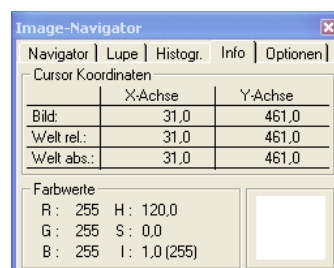


图像导航器（在相机取景器右下角勾选）显示如下图像：

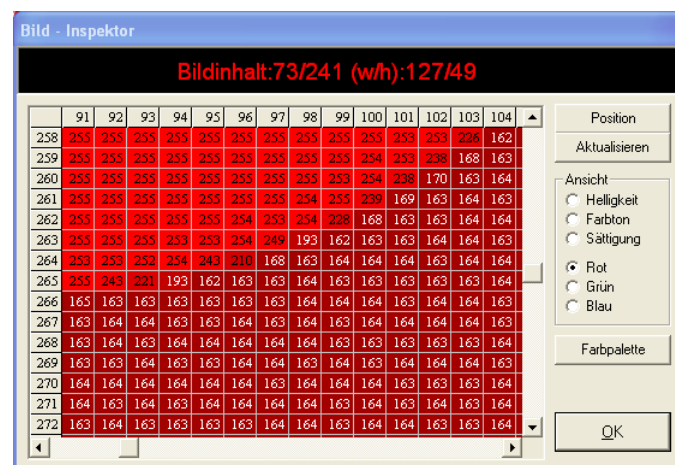


激活过滤后，图像导航显示 RGB

值。



例如，现在激活图像检查器，（View → Image analysis），图像现在可以显示为红，蓝和绿色。



转换成灰度（Conversion to Gray scale）

把 RGB 图像转换成灰度图像。

如果使用的是彩色相机，可在需要灰度图像时用此功能。例如，结合智能匹配功能。

源图像

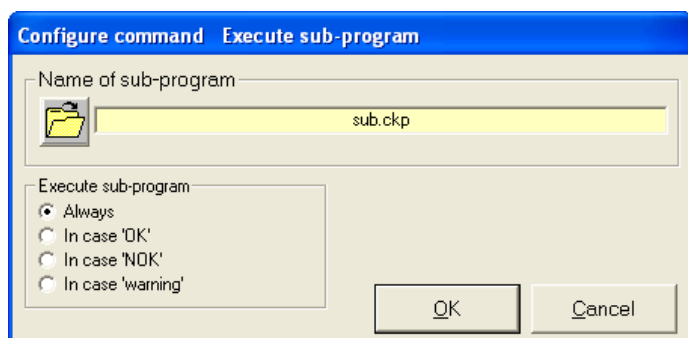


结果图像





执行子程序 (Execute sub-program)

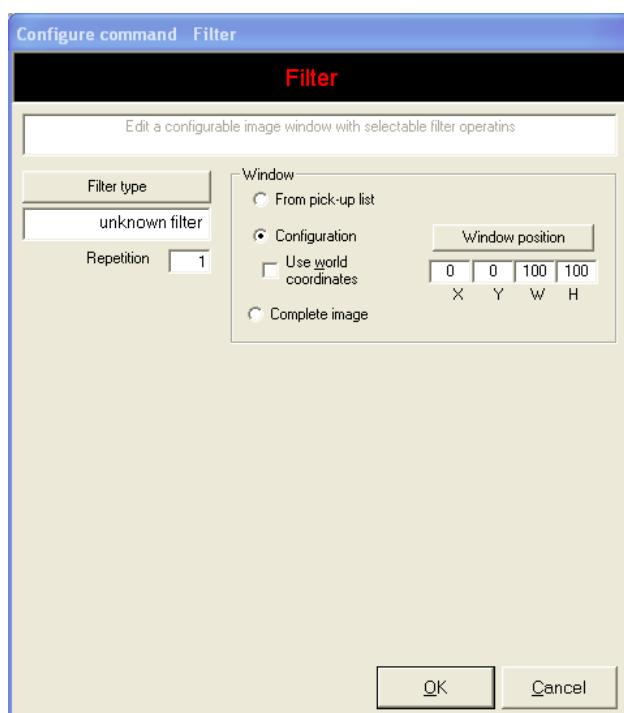


选择一个要执行的子程序。

在此可选择何时执行子程序，始终，OK，NOK，或警戒时。

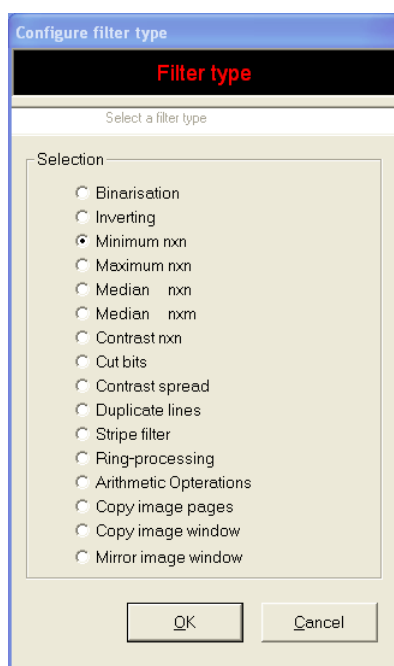


过滤 (Filter)



过滤命令可以对图像执行各种过滤操作。

过滤类型 (Filtertype)



点击过滤类型按钮，可看到供选择的过滤列表。根据参数不同显示的过滤数量不同，同样地，每种过滤的参数也不同。例如，N x N 过滤没有任何参数。但 Binarisation 过滤有两个阈值和前景背景色。

窗口 (Window)

来自选取列表 (From pick-up list)

配置 (Configuration)

位置追踪 (Position tracing)

整幅图像 (Complete image)

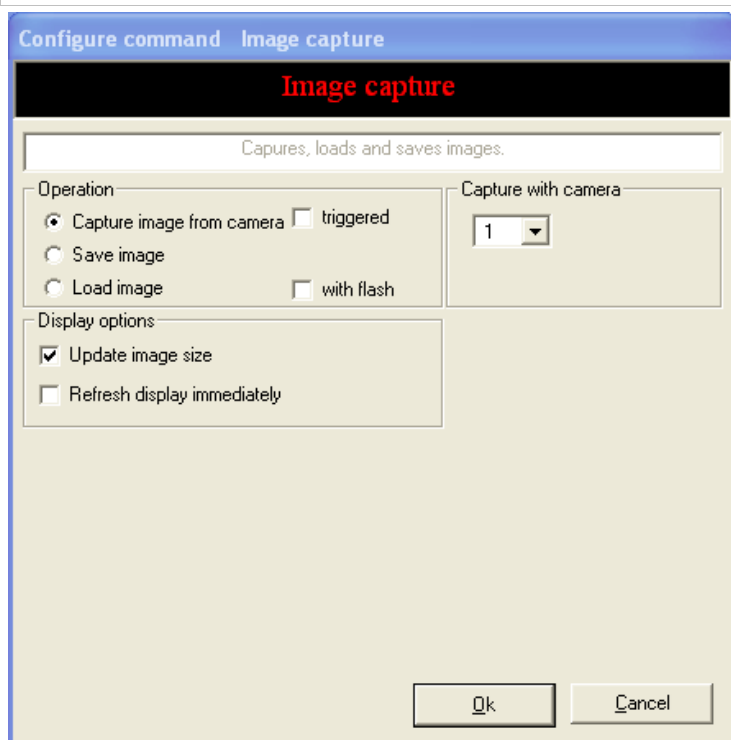
定义过滤运行的窗口。

激活位置追踪窗口。

对整幅图像进行过滤。



图像捕获（Image capture）



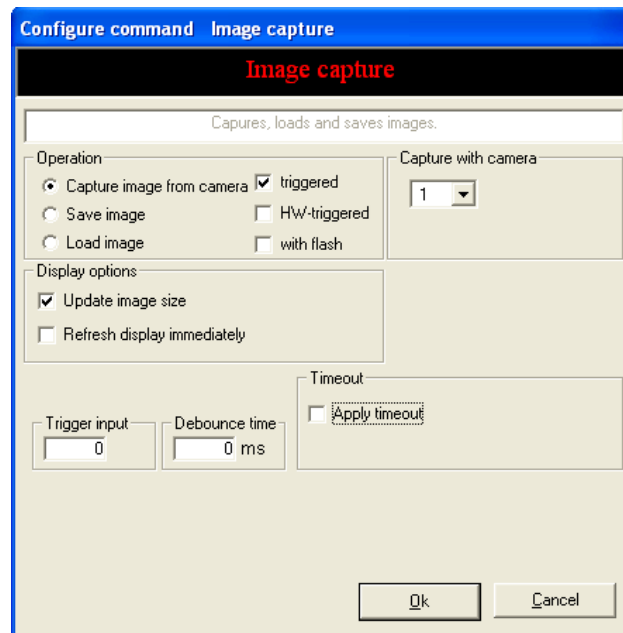
图像捕获功能的功能不只是名字显示的。不只可以用硬件捕获图像，也可以加载系统中的图像文件。另外，也可以将图像内存中的图像保存到系统文件中。图像可存为 bmp 或 jpg 文件。

运行（Operation）

从相机捕获图像（Capture image from camera）	如果激活从相机捕获图像功能，将试图用硬件捕获一个图像。
保存图像（Save image）	如果激活保存图像功能，将保存图像内存中的图像到系统文件。
加载图像（Load image）	如果激活加载图像功能，将加载一个系统文件图像到图像内存。

因为根据选择的操作不同，配置窗口不同，分别描述如下。

运行 → 从相机捕获图像（Operation → Capture image from camera）



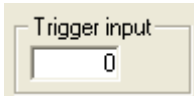
运行（Operation）

触发（triggered）	如果选中此项，系统会响应软触发。要确定一个数字输入为触发输入和反跳时间。
硬件触发（HW-triggered）	选中这个选项，如果图像捕获硬件支持，将自动激活硬件触发。硬件触发永远优于软件触发。
闪光触发（With flash）	如果激活这个选项，将根据图像捕获时间激活闪光。

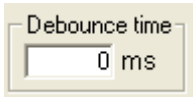
显示选项（Display options）

更新图像尺寸（Update image size）	针对图像捕获的选项，使软件调整图像尺寸到当前平台。如果在两个图像记录之间修改图像捕获模式为分级或 RIO，这样做是必要的。
立即更新显示（Refresh display immediately）	如果机器视觉系统连接有显示器，那么可以用此选项立即显示捕获的图像。否则只存放于系统中。

触发输入（Trigger input）

	在软触发模式下，确定触发输入。
---	-----------------

反跳时间（Debounce time）

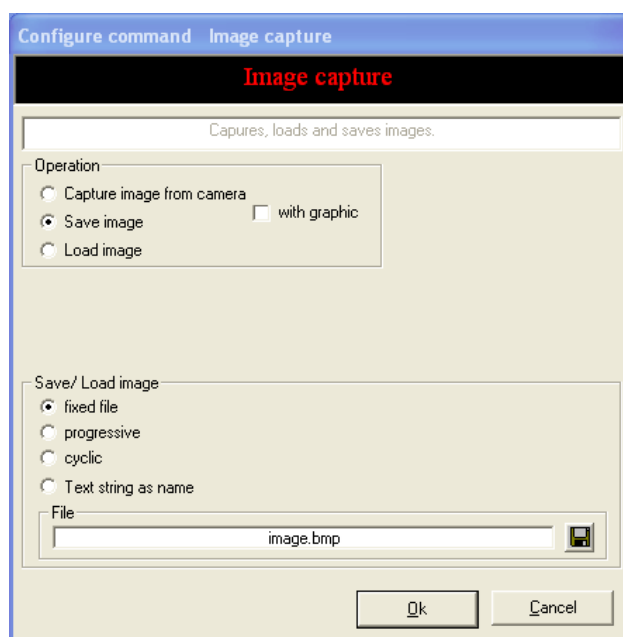
	在软触发模式下，确定触发输入的反跳时间。
---	----------------------

超时（Timeout）

申请超时（Apply timeout）	如果激活这个选项，可以调整超时时间。意思是，一定时间后触发信息将取消。如果发生超时，图像捕获有缺陷。
---------------------	--

用相机捕获（Capture with camera）

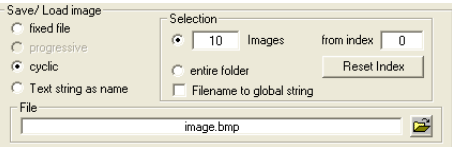
如果系统连接多个相机，用这个序号可确定用哪个通道（视频输入通道）的相机捕获图像。

运行→保存图像（Operation → Save image）

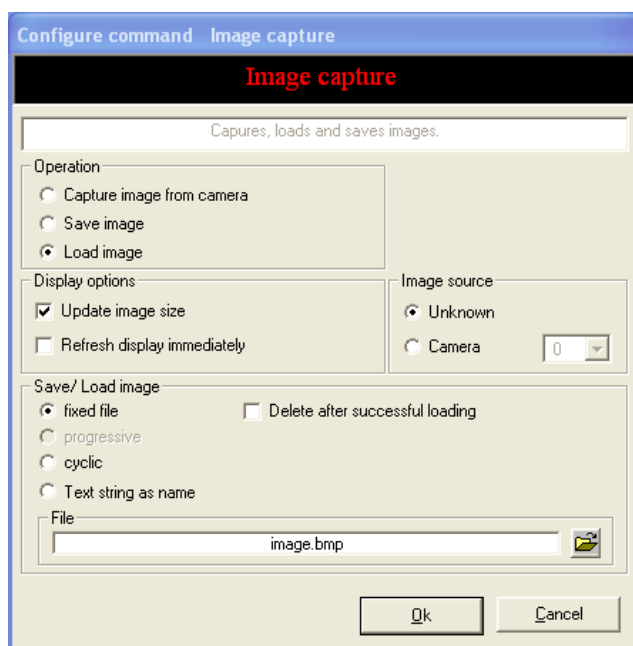
运行（Operation）

与图形（with graphic）	如果激活这个选项，叠加层的数据也保存在图像中。否则只保存图像灰度值。
-------------------	------------------------------------

保存/加载图像（Save/ Load image）

固定文件（Fixed file）	用此选项可定义要保存的文件的名称和保存的位置。 如果这个文件以后要被目标系统使用，最好不用修改保存位置，因为将无法同步。
渐进（Progressive）	如果选择 <i>progressive</i> 选项，将为文件定义一个前缀名称。将在文件名称前加上日期时间。
循环（Cyclic） 	用此选项在文件名称中加上循环的序号。序号从0开始到 N 并再次从0开始。 如果选中整个文件夹（ <i>entire folder</i> ）将按顺序加载文件夹中的图像。用重置序号（ <i>Reset Index</i> ）按钮，序号重置为默认设置。
文本字符串名（Text string as name）	此选项可用全局字符串命名文件。这样可以根据检查程序和其结果命名一个图像文件，如 NIOPart_4711。
文件（File）	显示当前文件名称。

运行 → 加载图像（Operation → Load image）



显示选项（Display options）

图像源（Image source）

未知（Unknown）	图像几何结构未知。
相机（Camera）	使用相机 N 中调整的图像几何结构。

保存/加载图像（Save/ Load image）

加载后删除（Delete after successful loading）	如果选中这个选项，加载图像后删除文件。
--	---------------------

针对 *fixed file*, *cyclic* 和 *text string as name* 在运行 → 保存图像中描述。



图像信息（Image information）

Configure command Image information

Image information

Wanted image information for given window is calculated and evaluated!!

Information

☐ Brightness (average)

☐ Value range

☒ Focus value

☐ Image format

Raster distance

Pixel

Focus parameter

Contrast threshold

Contrast pixel distance

Window

☐ from pick-up list

☒ Configuration

☐ Use world coordinates

☐ Whole image

Window position

X Y W H

☐ Global Tolerances

☐ Dynamic evaluation

Tolerance

Nominal value

☐ Warning limit:

OK Cancel

用此功能可确定或计算图像信息。除了可以得到直接结果，亦可用于评估。

信息（Informations）

亮度（平均）（Brightness (average)）	
焦点值（Focus value）	确定选定区域或整幅图像的焦点。
图像格式（Image format）	确定当前图像格式。

聚集参数（Focus parameters）

对比阈值（Contrast threshold）	此阈值用于确定焦点值。用此阈值可确定用于评估的像素。
对比像素距离（Contrast pixel distance）	确定两个像素间的对比度，像素距离。



图像传输（Image transfer）

Configure command Image transfer

Image transfer

Receiver

IP-Address: 192 168 0 1 Port: 4557 Address can be replaced temporarily by remote request ☒

Time of Sending

☒ Send immediately
☐ Send at end of program loop

External condition of execution

☐ Send only on remote request
☒ Send in every program loop

Internal condition of execution

☐ Send image on result OK
☐ Send image on result NOK
☒ Send image always

Image buffer to be sent

☒ IM according to editor
☐ Error IM No. 0

Region of interest (ROI)

☒ Complete image
☐ Static region of interest
☐ Dynamic region of interest
☐ World coordinates

Static ROI

X-Position: 0 Y-Position: 0 Width: 640 Height: 480 Teach in ROI

Dynamic ROI

☒ from register (index 0 to 3)
☐ from pickup list Selection

UDPPaket Retries

☒ Number: 3 Timeout [ms]: 5

Image resolution

☒ Original resolution
☐ Quarter resolution (every second pixel)
☐ 1/16 resolution (every 4. pixel)

Graphical overlay

☐ Send without overlay
☐ Embedd overlay in image
☒ Append symbolic overlay (only for EVT tools)

Format

☒ Grey scale image
☐ Color image

Image compression

☐ No compression
☒ JPEG compression

Test Ok Cancel

在此可将图像传送到任何 IP 地址。

接收（Receiver）

输入接收 IP 地址和端口。

执行的外部条件（External condition of execution）

在此确定何时发送图像，只有在有远程请求时发送或在每个程序循环中都发送。

发送图像缓冲（Image buffer to be sent）

在此确定发送哪个图像源（IM）。

UDPPaket Retries

如果超时，可确定是否重发和多久重发一次。

图像分辨率（Image resolution）

在此确定图像分辨率。可选择原始分辨率，1/4 分辨率或 1/16 分辨率。

格式 (Format)

可以灰度图像或彩色图像两种格式传送。

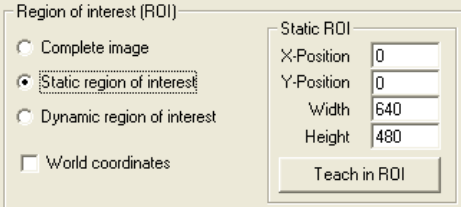
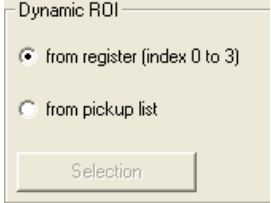
发送时间 (Time of sending)

在此定义图像发送时间：立即发送或检查程序结束后发送。

执行的内部条件 (Internal condition of execution)

选择发送图像的条件：永远，结果 OK 时或结果 NOK 时。

兴趣点 (Region of interest (ROI))

完整图像 (Complete Image)	发送整幅图像
静态兴趣点 (Static Region-of-Interest) 	可手动输入静态兴趣点的值或在实际图像上配置。
动态兴趣点 (Dynamic Region-of-Interest) 	可从寄存器或选取列表选择动态兴趣点的值。
完全坐标 (World coordinates)	可选择附加完全坐标。

图形叠加 (Graphical overlay)

发送的图像可有叠加或无叠加。

图像压缩 (Image compression)

发送的图像可被压缩成 JPEG 格式或未压缩。



INI 文件访问（INI file access）

用此命令可用 INI 格式编辑文件。保存值并易于用户再次根据名称找到文件。随时可更换名字。

文件（File）

通过文件选择框可以指定 INI 文件的存储位置和文件名，或选择要编辑的文件。

章（Section）

INI 文件名写入或读出的部分。

输入（Entry）

输入是要保存在章中的数据的名称。

访问类型（Access type）

写（Write）	在 INI 文件中写入输入。
读（Read）	读取 INI 文件。
删除文件（Delete file）	删除整个 INI 文件
删除章（Delete section）	删除选定的章。

删除输入（Delete entry）	删除选定的输入。
--------------------	----------

数据类型（Data type）

值（Value）	写值寄存器中的值。
点（Point）	写点寄存器上的点。
全局字符串（Global string）	写全局字符串。
常量（Constant）	写常量。
寄存器（Register）	写常量内容。
选取列表（Pick-up list）	写选定的选取值。

浮点精度（Floating point precision）

使用自己的格式（Use own format）	用此功能定义小数点后倍数。自己的格式适用于整个软件。
-------------------------	----------------------------

小数位（Decimal places）

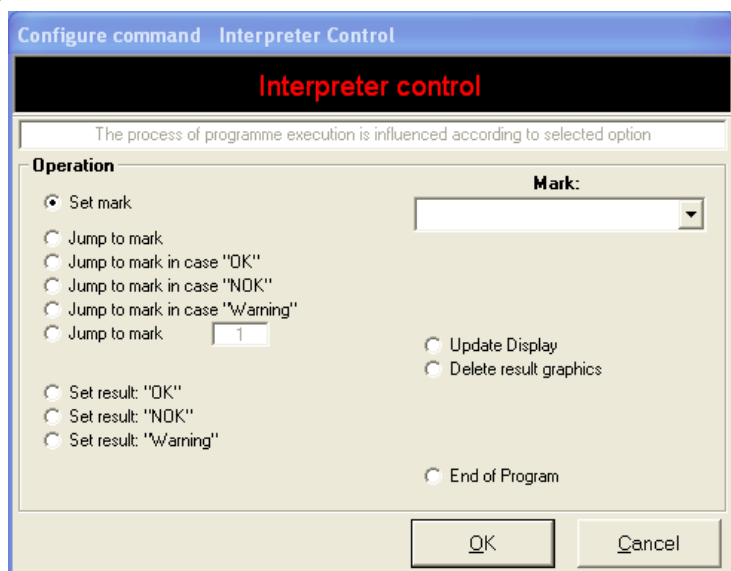
用此选项定义写浮点值的小数位数。

选项（Options）

此选项可在读取文件前再次加载文件，或在写文件后在文件系统中再次写入文件。



编译器控制（Interpreter control）



编译器控制程序的顺序。可创建标签，执行有条件的跳转或无条件的跳转。

设置标记（Set mark）	新的标记要使用一个新的名称。
跳到标记（Jump to mark）	执行无条件跳转。设置的标记已经定义好并在下拉菜单中。
OK 时跳到标记（Jump to mark in case “OK”）	结果 OK 时跳转到指定的标记。
NOK 时跳到标记（Jump to mark in case “NOK”）	结果 NOK 时跳转到指定的标记。
警告时跳到标记（Jump to mark in case “Warning”）	结果是警告时，跳转到指定的标记。
跳到标记（Jump to mark ...）	直到计数器到达默认时，有条件地跳到指定的标记。
设置结果 OK（Set result “OK”）	设置结果为 OK，覆盖所有之前 OK 的结果。
设置结果 NOK（Set result “NOK”）	设置结果为 NOK，在使用负逻辑时有用。
设置结果警告（Set result “Warning”）	设置结果为警告。

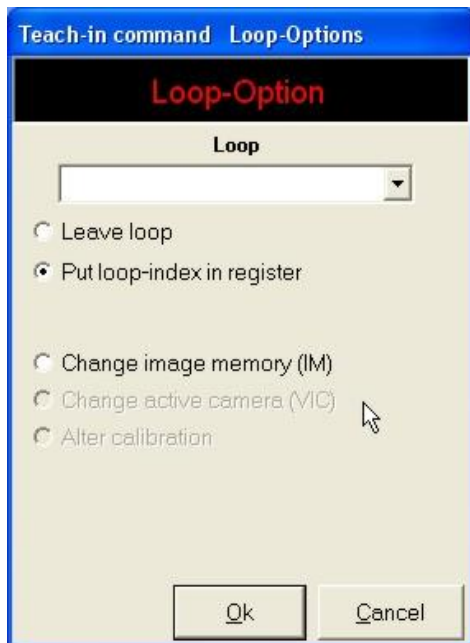
更新显示 (Update Display)	用此选项立即将图像内存中的图像传送到显示内存中。由于更新过程需要处理器花费一些时间，用此功能可不影响处理时间。
删除结果图像 (Delete result graphics)	用此选项删除结果图像内存。因为它发生在设定的时间点，所以处理过程不会受到影响。
程序结束 (End of Program)	编译器会显示主程序已过期，程序会再次从头开始。

根据（信赖）(According (dependent on))

总结果 (Total result)	如果所有命令都符合条件，则执行这个动作。.
命令 (Command)	如果所有标明的命令符合条件，则执行这个动作。



循环选项（Loop options）



通过标记和命令，可在一个循环内嵌入重复的操作顺序。然后根据选项不同，如循环指令修改如图像内存。

循环（Loop）	确定循环完成的区域。
离开循环（Leave loop）	取消循环执行。
循环编号放入寄存器（Put loop-index in register）	对应循环计数的当前循环编号放入寄存器。
改变图像内存（IM）（Change image memory (IM)）	根据图像内存列表改变当前图像内存。
改变激活的相机（VIC）（Change active camera (VIC)）	在可附带多个相机的系统中，系统会改变到下一个 VIC，即下一个相机。
改变校正（Change calibration）	在校正列表基础上切换校正。

可选命令螺丝控制可通过配置一个包括螺纹范围的测试窗口，提取螺丝的实质性特点。

根据不同螺丝对应的允许值修改特性，并得到好/坏的评价。



测量缝隙宽度 (Measure gap width)

Configure command Measure gap width

measure gap width

With probe 1 and 2 several points will be probed. The distance will be defined and evaluated.

Probe 1

☐ dynamic position

☐ Dispersion filter:

G170

☒ Minimum

☐ Maximum

Histogram

0 255

Probe 2

☐ dynamic position

☐ Dispersion filter:

G170

Alignment

☒ No

☐ Probe 1

☐ Probe 2

Histogram

0 255

☐ Global Tolerances ☐ Dynamic evaluation

Tolerance

Nominal value

☐ Warning limit:

此命令通过搜索区域定义两个点列表，并测量其间的距离。根据容差检测两个探头间的列宽度。探头必须以这种方式配置，在两个点列表间有合理的距离。

探头1/探头2 (Probe 1 / Probe 2)

配置位置 (Configure position)	
配置参数 Configure parameters	
动态位置 (Dynamic position)	在寄存器值基础上动态记录搜索位置。
扩散过滤 (Dispersion filter)	确定允许的扩散；标准1。
最小 (Minimum)	测量最小列宽度。

最大 (Maximum)	测量最大列宽度。
--------------	----------

对齐 (Alignment)

不 (No)	两个探头相互独立。
探头1 (probe 1)	面向探头1。
探头2 (probe 2)	面向探头2。

容差 (Tolerances)



更多内容，查看字符串描述（ **description of the character string** ）。



更多内容，查看字符分隔描述（ **description of the character separation** ）。



更多内容，查看字符识别描述（ **description of character recognition** ）。

对于读取和比较字符串的基本命令在各自命令中描述。

除了标准工具集也有扩展工具，可以读取特殊应用中的字符，如有水，气或电时。

同时也可以读取特殊字符，比如6行可以从总体上视为一行字符。

根据需要提供特殊的解决方案。



点 (Point)

Configure command Point

Point

A point is determined according to

Point determination

☒ Create ☐ use position adjustment

☒ manual ☐ dynamic X: 0,0 Y: 0,0 Configuration

☐ Interception of two straight lines

☐ Middle point between 2 P

☐ Image center

☐ Nearest point on straight line

☐ Copy point on top of register

☐ From point list into point register

☐ Transformation from image into world coordinates

☐ Transformation from world into image coordinates

☐ Window center

☐ From point register into point list

☐ Transform coordinates into values

OK Cancel

此命令可在图像上，用不同方式定义点。

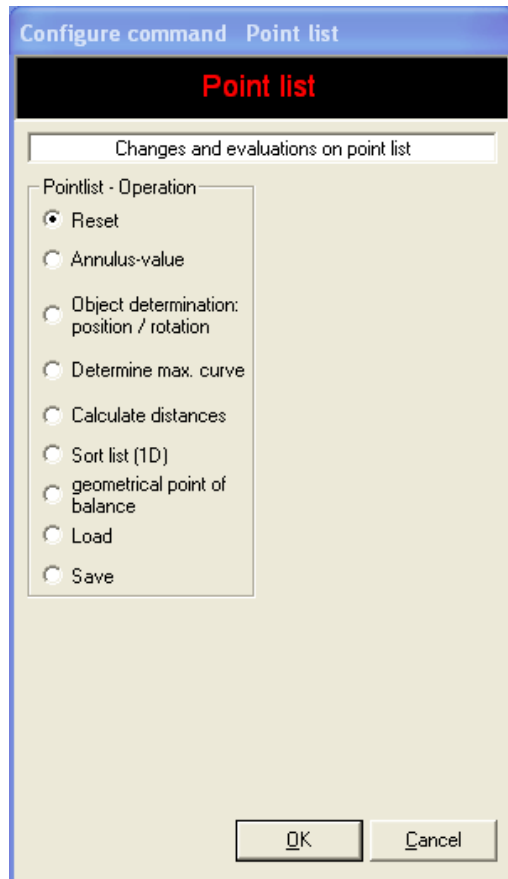
确定点 (Point determination)

手动创建 (Create manual)	手动填写数字，或在相机取景器中移动光标，创建一个点。
动态创建 (Create dynamic)	在寄存器或选取列表中选择，动态创建点。
两条直线的交点 (Interception of two straight lines)	确定两条直线，定义他们的交点为新的点。
两点间的中点 (Middle point between 2 P.)	定义两个点之间的中间点。
图像中心 (Image centre)	确定图像中心为新的点。
到直线最近的点 (Nearest point on straight line)	定义离直线最近的点为新的点。

寄存器最上面的点（Copy point on top of register）	将点放在寄存器最上面。
从点列表到点寄存器（From point list into point register）	从点列表中取出，存入寄存器。
从图像转换到完全坐标（Transformation from image into world coordinates）	点坐标从图像坐标转换成完全坐标。
从完全坐标转换到图像坐标 （Transformation from world coordinates into image coordinates）	点坐标从完全坐标转换成图像坐标。
窗口中心（Window center）	确定图像的中心点。
从点寄存器到点列表（From point register into point list）	寄存器点进入点列表。
转换坐标为值（Transform coordinates into values）	将点的 X 和 Y 值保存在值寄存器。

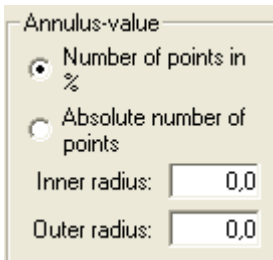


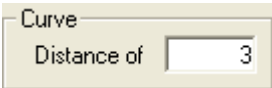
点列表 (Point list)



通过点列表命令来控制点列表。可以重置，也可以保存，排序和评价点列表。

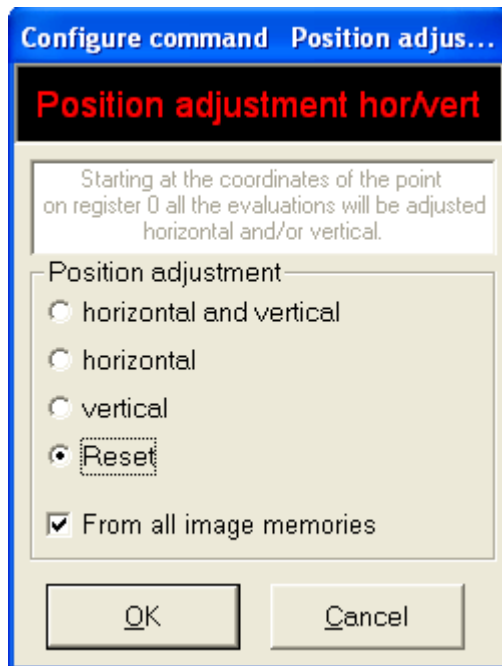
点列表操作 (Point list-Operation)

重置 (Reset)	清空点列表。当后面的命令需要评估点列表时需要先使用此命令。
环估值 (Annulus valuation) 	如果选择环估值，将在对象上放置一个环。确定环上的点的数量，点的数量可能以百分比或绝对值的方式表示。确定环的内半径和外半径。

对象探测：位置/旋转（Object determination: position/ rotation）	“ <i>object determination: position/ rotation</i> ” 选项用来确定点列表中点所在的对象的中心。在对象周围形成一个椭圆形， 椭圆形轴的交点和位于长轴上的一个点被保存在点寄存器中。这个功能用于圆心和 x 轴都有变化的动态校准。返回值取决于所选的选项。
最大曲线（Determine max. curve） 	选择 <i>determine max. curve</i> ， 可根据选择的像素确定点列表的最大曲率。用一个设定的曲率指示值去评估这个曲率。
计算距离（Calculate distances）	通过计算距离，确定点之间的距离。距离保存在值寄存器中。
排序列表（Sort list (1D)）	使用 <i>Sort list (1D)</i> 可以基于每个点的状态对点列表进行排序。这也同样适用于那些点，如用计数（ <i>count objects</i> ）写入点列表的点。.
平衡的几何中心（Geometrical point of balance）	确定点列表中所有点的几何中心。
加载（Load）	加载文件中的点列表的内存。
保存（Save）	保存点列表到文件。



位置调整（Position adjustment）



使用位置调整，它可以弥补一个工件的位置变化，通过跟踪测量随后执行命令窗口。通过参考点确定偏移叠加，参考点是位置调整命令从点寄存器中提取的。

位置调整（Position adjustment）

位置调整可以在水平或垂直方向上进行，也可以两个方向同时进行。

水平位置调整时，存入 X 偏移，垂直位置调整时，存入 Y 偏移。水平和垂直位置调整时，存入寄存器的值是在 X 偏移的上方和 Y 偏移的下方。

此命令需要点寄存器中的一个点。根据这个点，随后的测量和检查命令都在水平，垂直或两个方向上同时被改变。如果选择重置，则不再考虑位置参数。



探测点（圆探头）（Probe point (Circle probe)）

Configure command Circle probe

Probe points - circular

Annulus detection of points with the circular probe.
Number of detected points are evaluated.

Probe

Name:

Probe position

☐ dynamic position

Probe parameters

G170

Histogram

Direction

☐ outwards ☐ math. positive
☒ inwards ☐ math. negative

☐ Global Tolerances ☐ Dynamic evaluation

Tolerance

Nominal value

☐ Warning limit:

探头（Probe）

名称（Name）	在此给探头命名。
----------	----------

探头位置（Probe position）

配置探头（Configure probe）	配置探头的位置和尺寸。
-----------------------	-------------

探头参数 (Probe parameters)

配置 (Configuration)	定义边探测参数。
--------------------	----------

方向 (Direction)

向外 outwards	由圆心点向外搜索边。
向内 (inwards)	由圆心点向内搜索边。
正向 (math. Positive)	从0度向正向搜索。
负向 (math. Negative)	从0度向负向搜索。

容差 (Tolerances)



探测点（Probe points）

Configure command Probe points

Probe points

With the rectangular probe one or more points are parallelly probed. The number of detected points are evaluated.

Probe

Name

Probe 1

Probe position

Configure probe

☐ dynamic position

Probe parameters

Configuration

G170

Histogram

0



255

☐ Global Tolerances

☐ Dynamic evaluation

Tolerance

Nominal value

0,0

+

10000,0

-

0,0

☐ Warning limit:

☒

Test

OK

Cancel

探测点命令是最基础的抽取图像特征的命令。通过确定的阈值和自适应阈值，确定那些涉及灰色过渡的点。

探头（Probe）

名称（Name）	在此为探头命名。
----------	----------

探头位置（Probe position）

配置探头（Configure probe）	配置探头的位置和尺寸。
-----------------------	-------------

探头参数

配置（Configure）	定义确定边缘的参数。
---------------	------------

容差（Tolerances）



Riehmann 探头（Riehmann-probe）

Configure command Riehmann-probe

Find edges

An image region is defined by a rectangular probe in which the image data are projected transverse to the probe direction and then the edges are detected.

Probe position

Configure probe

☐ dynamic position

Edge direction

☒ all

☐ only bright / dark

☐ only dark / bright

☐ Show expert parameter

☐ Draw projection on image

☐ Global Tolerances

☐ Dynamic evaluation

Tolerance

Nominal value

0,0

+

10000,0

-

0,0

☐ Warning limit:

☐ Test

OK

Cancel

Riehmann 探头是投影探头，它以探头指令为基础，通过使灰度值总和横向探头方向，确保可以探测到很淡的边缘。

探头位置（Probe position）

配置探头（Configure probe）	配置探头的位置。
-----------------------	----------

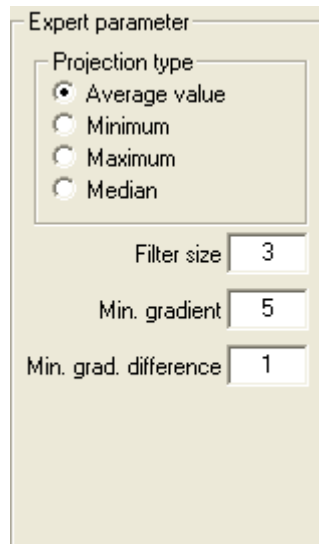
边的方向（Edge direction）

所有（all）	在探头区域显示所有的边。
---------	--------------

222

只有亮/暗 (only bright / dark)	只显示从亮到暗的边。
只有暗/亮 (only dark / bright)	只显示从暗到亮的边。

显示高级参数



在投影类型中可选择中间值，最小值，最大值和中位数。通过过滤尺寸，可确定平均值，最小值，最大值和中间值，中位数。

最小梯度 (Min. gradient)	确定投影图像边缘值。
----------------------	------------

在图像上绘制投影 (Draw projection on image)

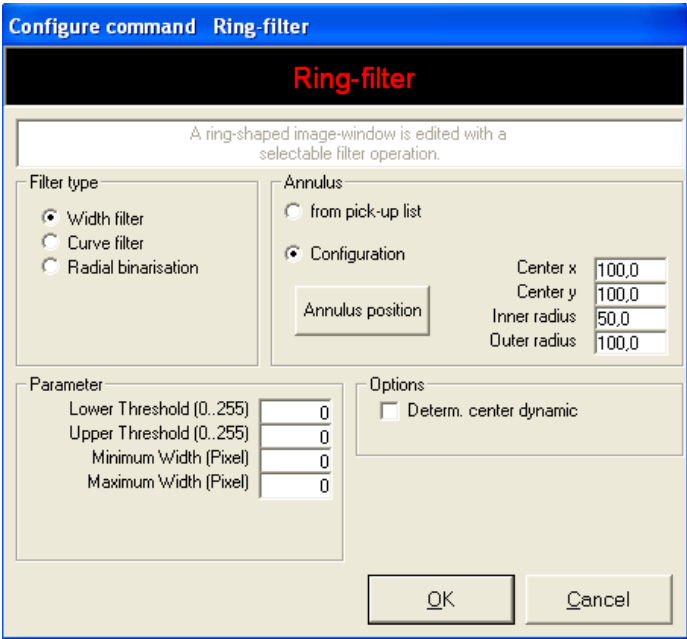
在图像上显示求和结果。

容差 (Tolerances)

通过可选的机器人命令，软件可以直接驱动多种机器人（如 KUKA, rice, ABB, Stäubli) 更多的信息参见手册。



环过滤（Ring-filter）



类似命令过滤器，用这个命令可以对图像进行过滤操作。这里图像是环形的，可用两种方法配置环形。在特定的位置定义环的特性（中心 x ，中心 y ，内半径，外半径），或者通过对话框定义。除了确定动态中心的其他输入选项根据过滤器类型不同而不同。

宽度过滤（Width filter）

宽度过滤放射状地检测配置的环。灰度值位于低阈值和高阈值间的环的各个部分和像素值位于最小宽度和最大宽度的部分，被宽度过滤标记上特定灰度值的颜色。不符合要求的环的部分显示成灰度值255（白色）。

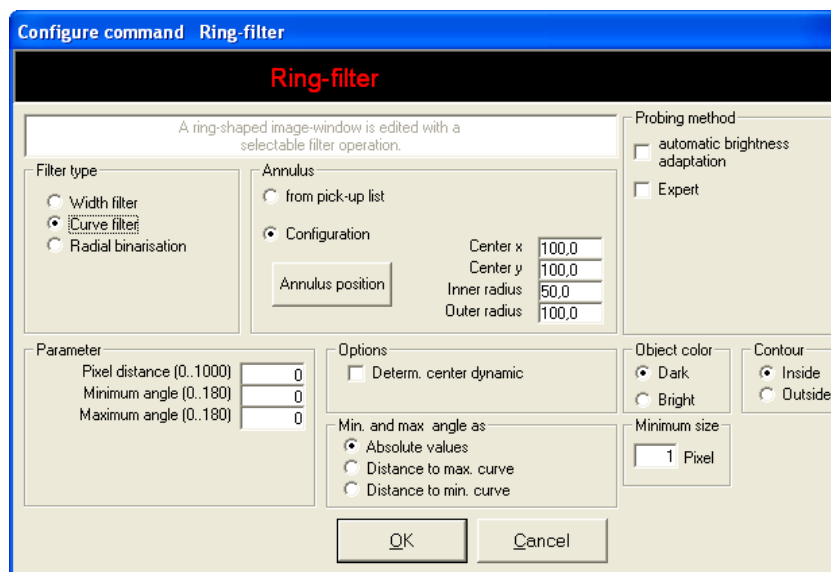
环（Annulus）

来自选取列表（From pick-up list）	通过选取列表定义环。
配置（Configuration）	另一个选择是手动配置环。

参数

此过滤适用于两个环形中间的区域。通过低/高阈值和最小/最大高度参数，定义两个环之间的区域。

曲线过滤 (Curve filter)



曲线过滤在弯曲的对象上检测配置的环。要确定对象的颜色（黑或白），也要确定轮廓在配置的环内或环外。过滤器之后会找到弯曲对象位于最小和最大角度（0°- 180°）间的环的所有部分。此外，还可以定义最小最大角度是绝对值，还是到最小或最大角度的距离。

参数

在此定义像素间的距离，最小和最大角度。

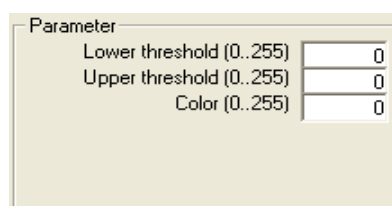
最小和最大角

在此还可以定义最小最大角度是绝对值，还是到最小或最大弯度的距离。

其他颜色 (Object color)

定义对象颜色，黑或白。

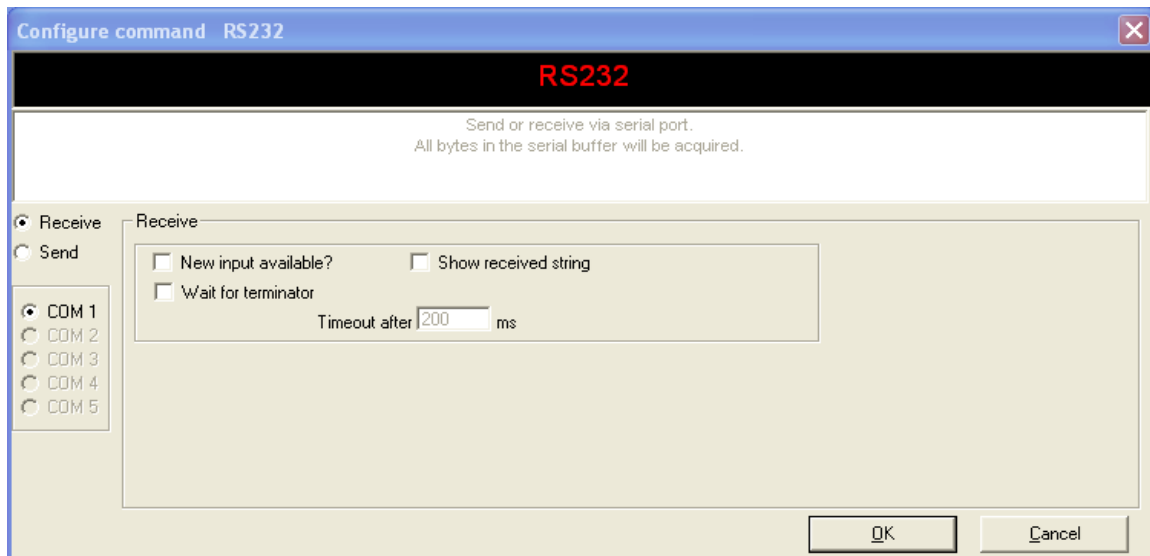
Radial binarisation



一个列向量在对象上移动。对象内的像素被平均化，如果平均值大于（小于）阈值，像素被定义成白色（黑色）。在 binarisation 中输入高/低阈值和颜色（180 灰度值）。



RS232

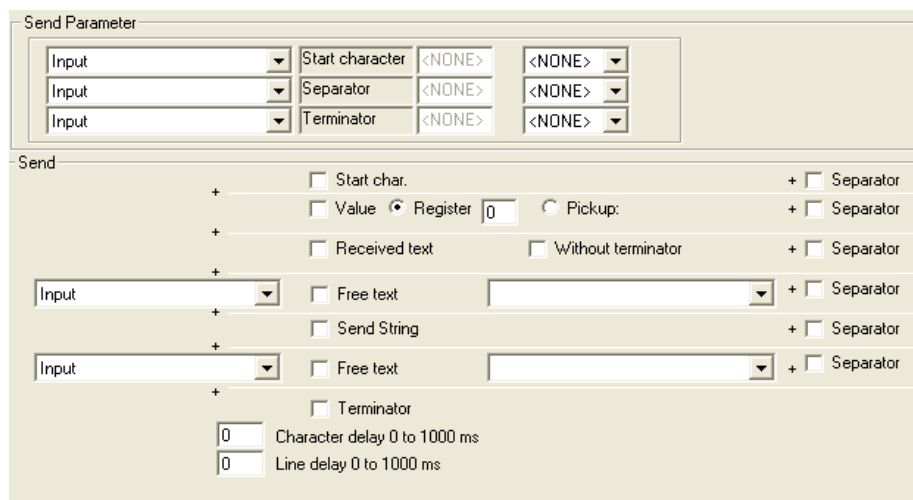


用此命令在选定的串口间发送或接收数据。

接收 (Receive)

试图接收。如果超时仍旧没收到任何内容，则取消接收。

发送 (Send)



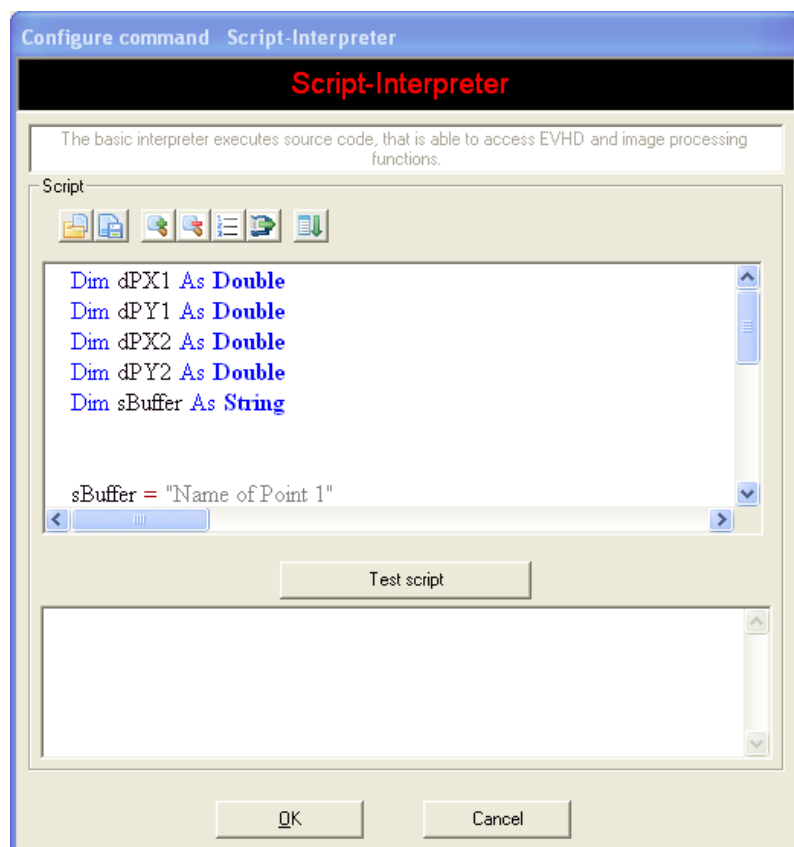
端口显示内容

COM 1....COM N (N=取决于系统，如
smartCam =1)

通过串口发送或接收数据。



脚本编译器（Script-Interpreter (BASIC)）



兼容的 VB6 BASIC 编译器隐藏在脚本编译器后。用户必须进入 SmartBuilder。你可以调用库中的多种功能来定义自己的图像评估宏。另外，可通过编译器进入图像内存，因此可实现对图像的评估。也可以进入操作系统，所以可以实现不同的交流协议。

导入脚本（Import Script）		如果点击导入脚本按钮，可打开一个文件，读取文件。因此可以在外问编辑器中编程，再从这里导入源代码。
导出脚本（Export Script）		用这个命令导出程序，在此可确定保存的位置和文件名。
扩大（Enlarge）		扩大编程窗口的字体。
缩小（Downsize）		缩小编辑窗口的字体。
行号（Line number）		在编程窗口为程序加上行号。

自动完成 (Automatic complete)		从数据库的外部功能选择适合的功能。用键盘 Ctrl+Space 可以直接激活自动完成命令。可以看到多种功能，用回车键选择某个功能。
测试脚本 (Test Script)		使用测试脚本命令执行编辑窗口中的脚本，并执行语法检查。在编辑窗口下面的窗口显示错误信息。
访问变量 (Variable Access)		打开高级窗口生成变量代码和选取访问。
布局编辑器 (Layout Editor)		在此定义语法突出显示 (如颜色, 字体等)



选择颜色通道 (Select color channel)



用此选项选择三个颜色，基于此可实现灰度值图像功能。如选择绿色，配置卡尺，只在绿色像素上寻找边缘。

色彩通道选择 (Color channel-selection)

红 (Red)	选择红色作为评估的基础。
绿 (Green)	选择绿色作为评估的基础。
蓝 (Blue)	选择蓝色作为评估的基础。

要更好地理解此命令，用户需要了解颜色像素在位图内存上的分布。在彩色图像中，每个像素保存为 RGB 值（红，绿，蓝）。灰度值图像与 8 像素颜色图像对比如下：

颜色线 [RGBRGBRGBRGBRGBRGBRGBRGB]

灰度线 [GGGGGGGGG]

用彩色命令为灰度值定义一个初始值如 **B**，之后定义颜色像素间的偏移。于是这个功能就象灰度值功能一样工作。唯一的区别是，这里到下一个 **B** 值的每个偏移是3而不是1。



设置输出（Set output）

用设置输出功能可以保存所有系统的输出。

输出（Output）

序号（Index）	从0 ... N（N = 输出号）中选择输出。
名称（Name）	命名输出，将显示在输出/输出菜单中。

新状态（New state）

输入一个必须采用的新状态（0 or 1）。

模式（Mode）

边缘触发（Edge-triggered）	如果选择边缘触发，会根据新状态产生一个侧面。
水平触发（Level-triggered）	如果选择水平触发，输出采用输入状态。

设置输出 (Set output)

好 (Good part)

结果 OK 时设置输出。

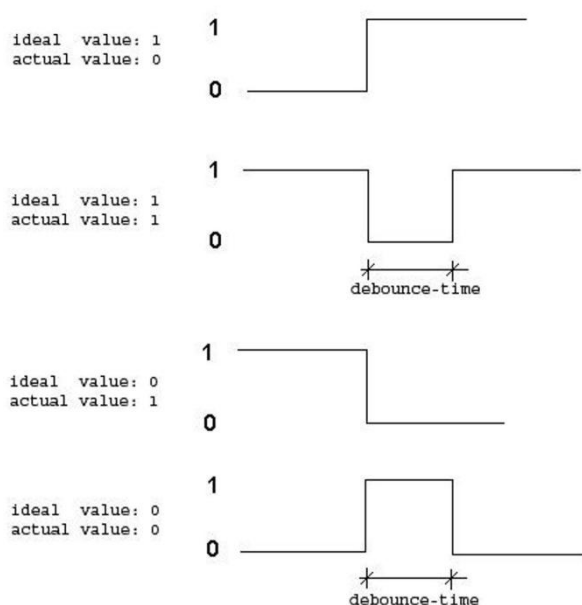
不好 (Bad part)

结果 NOK 时设置输出。

警告 (Warning)

结果是警告时设置输出。

如果使用数字输入/输出，有两种模式。如果在水平触发模式下，只有信号水平是重要的，例如：如果你设定1而输出水平已经有1了则没有变化。然而，如果你工作在边缘触发状态下（即侧面控制），设置状态信号和实际状态变化如下：



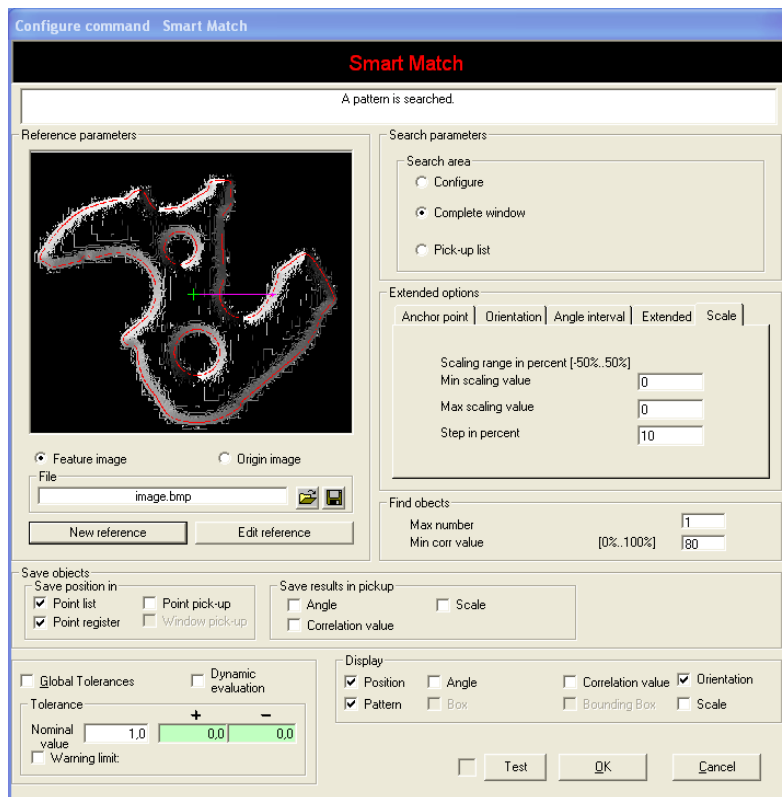
正如你看到的边缘和水平触发之间的差异，在从1信号变化到0时，状态短期内变成0。这是有用的，如果在接口端子不存储状态时这一点很有用，因此总是要设置一个状态。

另外两个选项之间的差异是在边沿触发模式下的信号立即发生变化。这在图像采集时用来激活闪光是必要的。水平触发的信号变化是更适合于图像采集前激活闪光。

反跳时间决定了设置信号的时间。例如，从边缘状态1到0的变化再变回0非常快，可以使终端未识别图像。因此，设置一个反跳时间，以确保信号的变化是由终端实现的。



智能匹配（SmartMatch）



这个工具可以在图像中找到之前配置的图形。不再受位置和旋转的影响。

我们将通过示例说明智能匹配的功能。

加载图像： Training_1_00000.jpg cyclic (10 个图像)。

点击 “new reference” 按钮，配置参考图像。保存配置的参考到文件，如 demo.smp。

点击 OK 确定。如果现在开始执行检查程序 – 开始检查循环 “Inspection program – start test loop”，可以看到图像中每个位置的模式。

搜索范围（Search area）

在此定义在哪里搜索图像。要在整个图像中寻找或只在用户定义的区域寻找。

查找对象（Find objects）

标明要查找的最大对象数量和最小相关值（0%....100%）。

保存位置（Save position in）

确定值保存的位置。

在选取列表保存结果（Save results in pickup）

确定要保存哪个值到选取列表。

容差 (Tolerances)

显示 (Display)

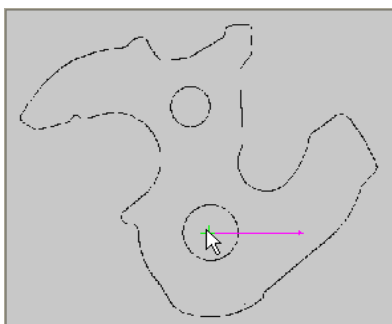
在相机取景器中显示实际的值，如对象的位置和角度。

扩展选项 (Extended options)

锚点 (Anchor point)

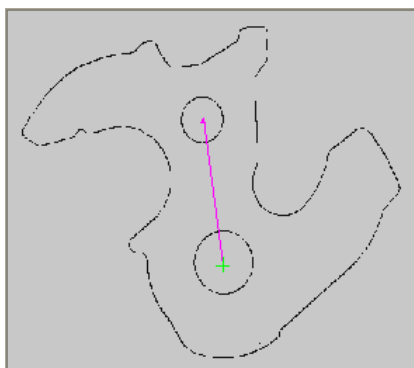
默认情况下，以图像中心作为定位点。上图中，锚的形象类似绿十字。

通过自定义锚点选项，可以按自己的意愿改变锚点。点击自定义锚点，然后点击配置。如果你现在把鼠标移到参考图像（图像智能匹配配置窗口左上方），那么锚点（绿十字）就在鼠标尖端。按一下就在此定义了锚点。



取向 (Orientation)

默认的方向是水平的（在图像中显示为粉红色箭头）。方向可以调整为垂直或自定义。对于后者，这个过程类似于自定义锚点。点击配置后，在参考图像上定义方向。按一下就确定了方向。



角 (Angle)

在这里，通过最大和最小的角可以对旋转对象限制搜索区域。在这种情况下，增量表示搜索的噪音。推荐增量1。

比例 (Scale)

智能匹配通过与参考图像对比寻找高度放大/缩小的对象。用户可以定义图像的最小/最大缩放比例。

排序 (Sorting)

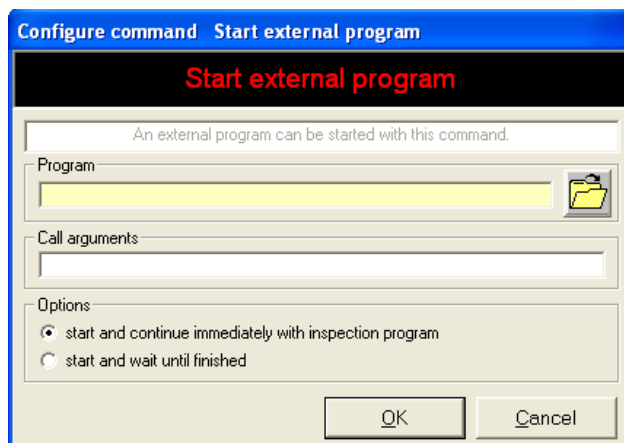
智能匹配命令的结果被存储在点列表中。用户可以决定，是否排序或怎样排序。

扩展 (Extended)

在此确定怎样保存命令及命令运行的速度。



开始外部程序（Start external program）



用此命令开始一个外部程序。

程序（Program）



在此调用一个外部程序。

调用的参数（Call arguments）

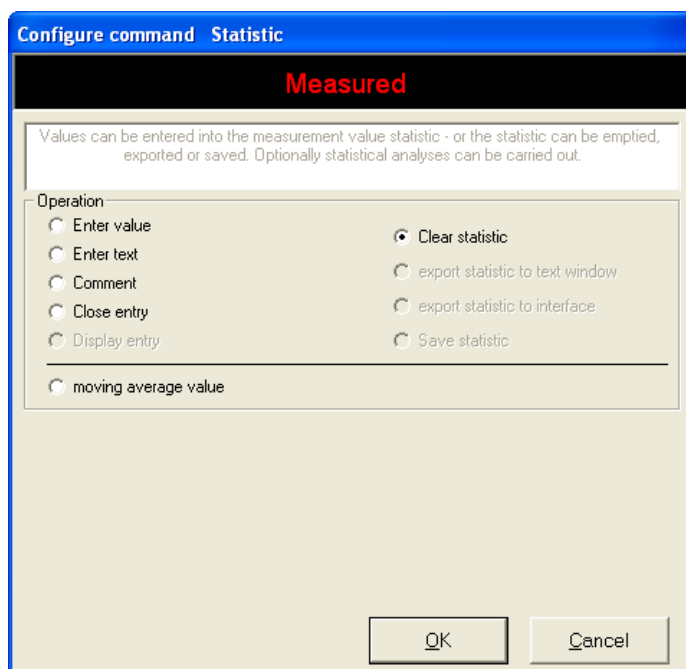
在此为程序设置更多的参数。

选项（Options）

在这里，你有两种方式启动程序。可以选择启动程序，并立即进行检验程序；或选择启动程序并等待。



统计 (Statistic)

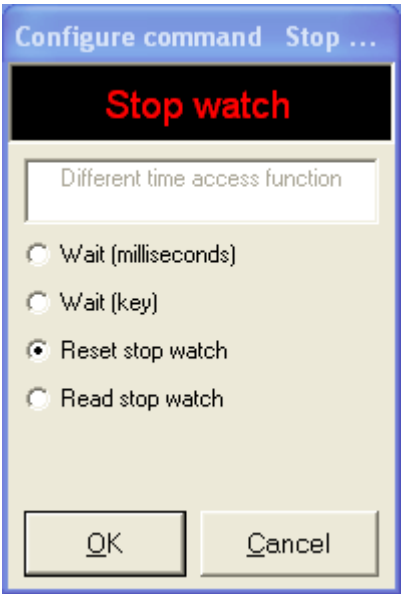


在测量统计命令中，可以输入值或删除，读取或保存统计值。也可以执行统计分析。测量统计由表格组成，统计值保存在表格中。

运行 (Operation)

输入值 (Enter value)	选择输入值和列号码，向表格对应的列输入值寄存器的上限值。
备注 (Comment)	保存带有备注的值。
关闭输入 (Close entry)	在一个测试周期结束时，为了停止向一行中输入，必须使用“关闭输入”命令去关闭“输入值”命令。之后的值被写入到下一行。
清除统计 (Clear statistic)	清除表中的内容。
移动平均值 (Moving average value)	计算最近输入的 N 列数值的移动平均值。计算的结果将与标称值和容差做对比。标称值在容差内时如果激活移动平均值，则结果显示 OK。否则显示 NOK。标称值保存在值寄存器中。

 秒表 (Stop watch)



秒表可以用来测量时间，定义等候时间，或等待运行命令。

等待（毫秒）（Wait (milliseconds)） Time (ms) 100	停止执行测试程序，以毫秒为单位调整值。
等待（按钮）（wait (key)）	停止检查程序，直到用户按任意键。
重置秒表（Reset stop watch）	秒表重置为0并立即开始运行。
读取秒表（Read stop watch）	读取当前秒表值并传送到值寄存器。



直线（Straight line）

Configure command Straight line

Straight line

A best-possible straight line is determined through a number of points.
(by minimising the distance squares)

Options

☒ Best fit straight line

☐ Dispersion filter

☐ 2 Points

☐ Perpendicular line

☐ Parallel straight line

OK

Cancel

根据点列表或寄存器值确定一条直线。

选项（Options）

最适合的直线（Best fit straight line）	基本点列表的最适合的直线。根据 LSR 过程，计算出通过点列表所有点的一条直线。
两点（2 Points）	由两点定义一条直线。两点可取自点寄存器或选取列表。
垂直线（Perpendicular line）	根据一条现有直线或点确定一条垂直线。
平行直线（Parallel straight line）	根据一条现有直线或点确定一条平等线。



文本 (Text)

Configure command Text

Text

Writes text and/or values on the Camera viewer inspection image.

☒ Text only ☐ Straight line
☐ Value ☐ Circle
☐ Point ☐ Text string

☐ Write into status line

Text

Text form and position

Color

☒ Standard
☐ Good
☐ Warning
☐ Bad
☐ Acc. to result
☐ Acc. to command

Position 512 384
 X Y

☐ Use position adjustment

Font size 20 Pixel

Write text

☒ Into overlay
☐ Into image

OK Cancel

在显示区域（相机取景器中）显示文本或值。根据硬件平台不同可用不同的颜色显示。

只有文本 (Text only)	只显示输入在下面文本框中的文本。
值 (Value) Source ☒ Register Index: 0 ☐ Pick-up list:	显示值寄存器中的值。
点 (Point)	显示值寄存器中的点。
直线 (Straight line)	显示值寄存器中的直线。

圆（Circle）	显示值寄存器中的圆形。
文本字符串（Text string）	显示全局字符串。
写入状态行（Write into status line）	如果激活这个功能，输出将显示在状态行。

文本（Text）

在此可输入要显示的文本。如当测量角度时，可选择值并在此输入“角度”。在相机取景器中将显示如：角度：65，00。

颜色（Color）

在此定义文本显示的颜色。标准颜色是粉色。如果选择好，文本颜色是绿色。如果选择警告，文本颜色是黄色。如果选择不好，文本颜色是红色。

位置（Position）

在此通过直接输入数字或用鼠标点击相机取景器的相应位置定义文本显示的位置。

文字尺寸（Font size）

直接输入数字定义显示文本的尺寸。

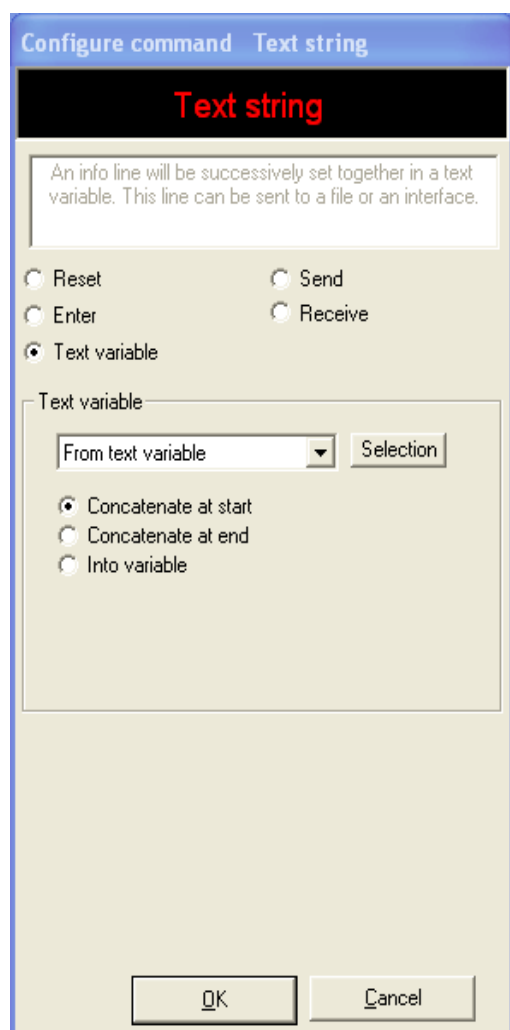
用户位置调整（User position adjustment）	显示全局坐标值而不是图像坐标值。
----------------------------------	------------------

写文本（Write text）

写入叠加层（into overlay）	如果硬件支持叠加，文本将写入叠加层。
写入图像（into image）	文本写入图像，于是文本成为图像的一部分。



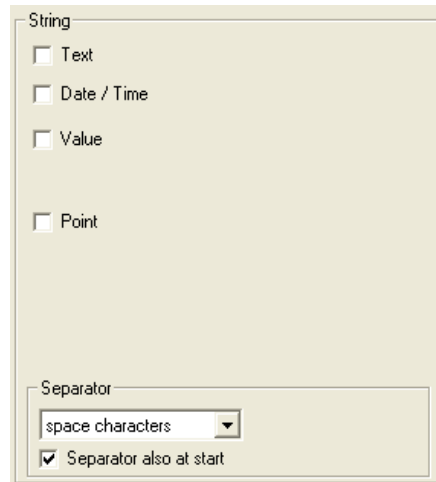
文本字符串（Text string）



用此命令用字符串填充全局字符串缓冲，之后传送到指定目标。

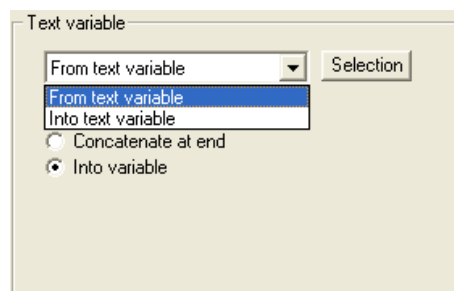
重置（Reset）

重置全局字符串缓存；之后缓存中再也没有符号。

输入 (Enter)

The 'String' dialog box contains a group box with four unchecked checkboxes: 'Text', 'Date / Time', 'Value', and 'Point'. Below this is a 'Separator' group box containing a dropdown menu set to 'space characters' and a checked checkbox labeled 'Separator also at start'.

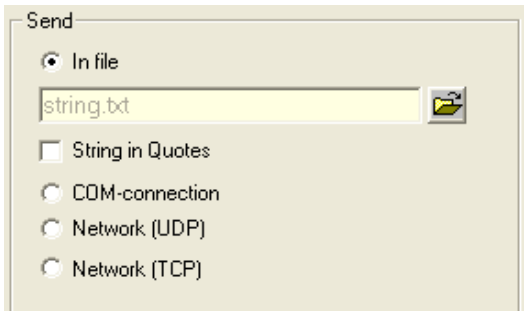
在全局字符串缓存中保存文本，值，日期/时间或点。如果有多个输入，就扩展字符串直到重置命令清空缓存。

文本变量 (Text variable)

The 'Text variable' dialog box features a dropdown menu with 'From text variable' selected, a 'Selection' button, and a list box containing 'From text variable' and 'Into text variable'. At the bottom, there are two radio buttons: 'Concatenate at end' (unselected) and 'Into variable' (selected).

选择一个文本变量。另外，可将实际文本字符串插入到文本变量，或者将文本变量内容插入到实际文本字符串。

发送（Send）



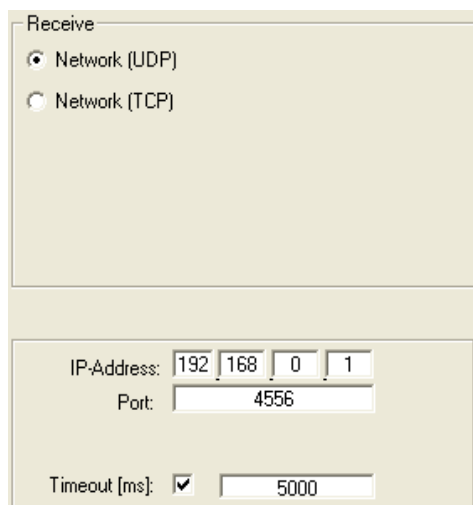
含有值的全局字符串缓存发送到选定的目标。

在文件	如果您单击“文件名”打开一个对话框，指定文件所在的位置并选择一个文件。如果提供的文件也需要同步调整，就不可以修改建议的文件保存位置。
引用字符串	在字符串开始和结尾处设置引用。
COM 连接	选择一个可用的串行接口用于传送字符串。
网络 (UDP / TCP)	你必须标明以太网传送文件的 IP 地址和网络端口。UDP 测试服务器保存了 SmartBuilder 软件的 BIN 文件夹中。



必须指定文件要接收和输入的目标系统的 IP 地址。数据发送和接收的位置要相同。目标系统中，要使用相同的端口读取和发送数据。

UDP 测试服务器基于 Windows，针对其他操作系统，源文件存于 UDPTTestServer.ZIP 里。如果没有数据到达目标系统，需要检查是否有防火墙阻止了数据传送。如果是这样，UDPTTestServer 要设定为不受防火墙限制。

接收 (Receive)

The image shows a 'Receive' configuration dialog box. It has a title bar 'Receive'. Inside, there are two radio buttons: 'Network (UDP)' which is selected, and 'Network (TCP)'. Below these, there are input fields for 'IP-Address' (192, 168, 0, 1) and 'Port' (4556). At the bottom, there is a 'Timeout [ms]' section with a checked checkbox and a value of 5000.

VizCam 提供可选的产品，可将数据自动采集到标准应用程序，如 Microsoft Office 和 OpenOffice 中，这样就可以将测量和检验的结果保存到如 MS Excel 或 OpenOffice Calc 中，并不断评估。从一个文件中的读取值很简单，因为利用“保存到文件”选项可创建一个 CSV 文件，因此可以自动读取这个文件。

5 实施概念

以下章节的目的是为了演示如何使用 **SmartBuilder** 图像处理软件的功能，方便、快捷地创建图像处理解决方案。在所有考虑因素中，确定基本参数是至关重要。对于一个高效的图像处理解决方案，它必须遵守这些参数。这听起来微不足道，但这些参数是图像处理解决方案得以成功实施的关键所在。

成功实施的一个图像处理的主题的一个非常重要的方面是，约30% - 40%的成功解决方案要依靠光学和照明的最佳选择和调整。然后遵循软件的其他部分也起着重要的作用。

出于这个原因，各种用于调整照明和光学质量的工具都被集成到 **SmartBuilder** 软件中。只有当确保这些工具能最佳地显示照明和光学特性，才应该去考虑分析算法。最后，才是用分析算法识别或检测这些特性，区别结果是好或坏。

5.1 图像处理解决方案的预先考虑

安装位置

检查现场在图像处理解决方案中也起着重要的作用。这里不仅要确定要分析的图像的区域和大小。而且还要确定检查位置的环境情况。

在这方面，重要的是要知道外部光线以何种方式影响图像区域。这种情况下，要知道是否有可能离开安装地点。由于图像处理主要取决于对比度和色彩，稳健的解决方案的重要之处在于如何改变照明条件。例如，如果在阳光可以到达的地方安装，那么它总是比安装在机器中的人工照明强烈。因此，该结果可以完全改变。

因此，安装时要做的首要工作是防止所有来源的外来光线。如果实现一个完全的屏蔽是不可能的，那么就要采取第一个例题的解决方案。

在这种情况下，可以想象如尘土飞扬的环境，镜头前被灰尘阻挡影响了对比度，同时也影响照明条件。一个例子是所谓的冷光源。光发射由一个玻璃纤维束组成，因此是冷的。在光产生的地方注入纤维，可发现高功率的灯（主要是卤素灯）很热。因此，光通过抛物面反射注入光纤。在非常脏的情况下，这种照明结构导致的灰尘被烧毁到抛物面反射镜，因此减弱了照明电源。在图像领域，导致对比度产生变化及错误的评价结果。

这是本章讨论的解决方案的例子之一。但同时，它显示了如果不考虑基本条件及其后果，就可能引起失败。

特殊环境

有关图像处理解决方案的可行性，测试的一部分永远是第一位的。在实验室中执行所有的分析，并随即选择相应的组件。

不幸的是，光学和照明的局限性常常被忽略。因此，第一步是要找出是否有办法获得要检查的特点。之后应立即检查，不论这样的安排是否可以在测试范围内。因此，我们建议，在特别情况下，要先检查，以找出哪种光学和照明的安排是可行的。

如果不可能实现透射光照明，如因为工作持有设备需要这种安排这种灯光测量某些值。那么就不需要进一步的分析了，因为机器永远不会满足这些要求。

另一个特别困难的例子是，当有建立了一个主题的远心目标。组件的长度为20毫米，但安装的空间不够大。在这种情况下不可能安装镜头。因此，所有的分析都没有必要，因为重要的是如何找到安装组件的办法。

环境条件

我们已经解决了关于环境条件的问题。任何条件下，这些困难都不应该被忽略。主要是相关的标准不能改变。这些标准往往会影响光学和照明。

不仅光学和照明受环境条件影响，组件也受其影响。如果是一个精确的光学测量系统，对于理解测量的值，基础条件是非常重要的。例如对于微米，测量工件和机器组件的热膨胀会产生很大的作用，因为图像处理组件是相连的。它们的线性扩展，可以被图像处理测量到。如果没有考虑到热膨胀则会发生错误，这实际上是由环境或由最理想的结构造成的错误。

图像处理过程中的灰尘粒子，也属于环境条件。但他们大多在样品期是不存在的，因为在运输或储存过程中多数灰尘消失了。油滴是一个典型的例子，因为机器的运转他们总能在一台机器上出现。但当发送的是样品时，完全没有油滴。但是，对比是图像处理评估的一部分，而油滴导致它不能与其他部分做对比。这导致了虚假的测量结果。当然，使用单位不接受油滴的失踪等，问题因此而产生。

建议测试过程中发现可能存在的薄弱点，并从开始明确要做什么，或在检查前通过技术安装排除所有故障。

检查速度（处理速度）

检查速度很少会引起问题。如果组件是移动的，问题是如何在运动中优化分析，这样才可以在此基础上做后期的评价。最大测量精度直接取决于此。在此背景下，有两个不同的过程：连续过程和离散过程。两者都必不可少地面对同样的问题：如何尽可能尖锐地显示个体特征。对于快速移动的过程图像处理解决方案有两种，这也是摄影领域了解的。要么由闪光灯或较快的快门冻结画面。现代传感器波塞的电子捕获，因此，实现快门速度与闪光速度相同，这取决于用户选择哪个程序。

检查组件

有关测试元件，你应该谨慎行事，不是说他们已经“老旧”，因为定向的图像处理总是检查对比点。通过生产后这些特点并未直接出现。仅在存储后材料发生积极或消极的变化，如果一些特点是在保存很长时间后才被发现，那么会很难探测到这些特点，在这种情况下，它是一种消极的变化。然后要花很多时间让这个特点出现，虽然它在生产过程中有高对比度。

对于那些非常简单就能探测到的特点，但在生产过程中却根本不能识别或非常难以识别，也是非常不利的。原因可能是如缺陷特征与大气中的氧气发生反应，因此出现了对比。另一个问题是，如果测试厂房是用来控制进货的。在这种情况下，检查部件可能被分配来定义可行性现在可以有一个完全不同的颜色，如另一个表面纹理。这就不可避免地导致测试厂产生了增加的伪错误。因此，要在之前定义和阐明测试部分的方差。

5.2 光学和照明（Optics and illumination）

光学和照明是必不可少的因素，这也决定了图像处理机器工作的方式。

光学选择

不幸的是，没有适当的光学规则指导。但道德要确定的参数是镜头。大多数情况下，值得向专家咨询，因为需要考虑有关镜片的各种参数

如果知道镜片的种类，建议考虑测试镜头。测试的方式也最好由专家进行。

对于用很短的快门速度或短的闪光速度进行的快速进程，要检查镜头的光强度。

远心镜头

关于极其精确的测量，选择这些镜头总是明智的。这些镜头的缺点，主要是价格和庞大的安装形式。

照明的选择

近年来随着照明的发展，由于有更高效率的 LED，使用螺旋灯丝灯泡下降。螺旋钨丝灯今天仍在使用，但它要求非常高的光功率，而这样一盏灯的光功率远高于 LED 的电源。

这种照明的缺点是较短的耐用性和快速的老龄化进程。因此，由于耐用性和可靠性，LED 照明是首选。

LED 照明的另一个优点是设计的灵活性。LED 照明几乎可以做成任何形式。只要有合适的盘去安装这些光源就可以。

5.3 生成一检查程序

在这一章中，我们想向您展示如何生成 SmartBuilder 的检测程序。即使你只有该软件的演示版本，要完成所有的程序步骤也是没有问题的。SmartBuilder 集成图像数据银行甚至可以在没有相机的情况下运行一个实际的程序。

不同的图像数据银行自动在安装程序的同时安装。他们允许编程来模拟现实的组成部分。

我们认为你已经了解软件的重要结构，因此，我们不详细描述所有的步骤，例如：在哪里可以找到哪些文件。

在 SmartBuilder 软件的设备列表的本地文件，以及其相应的图像数据银行，样本文件或字符集是最佳范例程序。下面你可以找到所有检测程序和相应的图片。

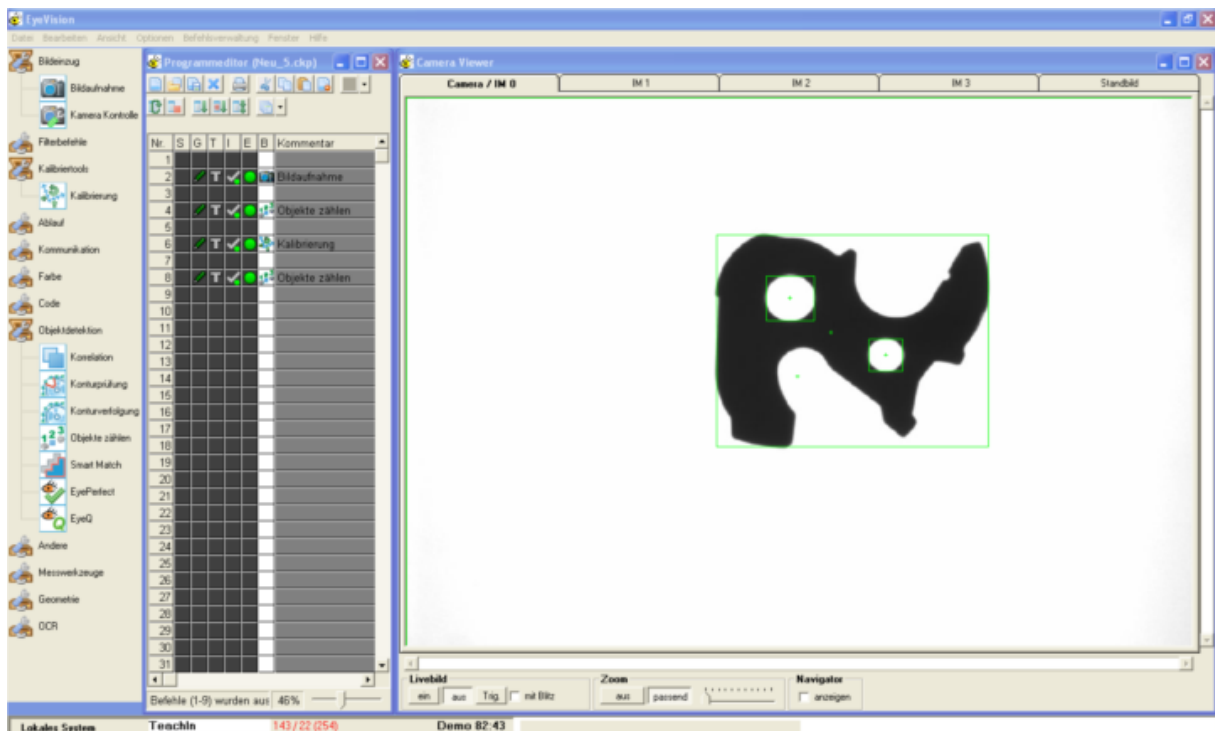
直接连接到本手册的检测程序总是以“Man”为前缀开始。如果包括一个更详细的例子，然后字母 BSP 添加以上命名，如 ManBspXXXX.ckp。这将是本手册的示例程序 XXXXX。

5.4 例子

在这一章中，你可以使用 VizCam 保存的检查程序。不要惊讶，如果你的相机取景器中的图像，与这里的屏幕截图图像不同。

而且，经常需要你再次加载图像。所以，如果你加载一个检查程序，但是没有图像显示在相机取景器中，请单击程序编辑器中的图像捕捉图标加。然后在数据（data）位置，你可以看到图像的路径。点击加载数据和选择要显示在取景器中的图像。点击 OK 确认。点击图像捕捉图标下面的空白处，显示图像，然后检验程序可以正常运行。

例 1: 查找对象和数洞





通过图像获取命令可以从本地文件中选择图像（此处为 Training_1_00000.jpg）。



对象计数器被配置为在整个区域中搜索黑色区域。（用吸管点击对象的黑色区域）。

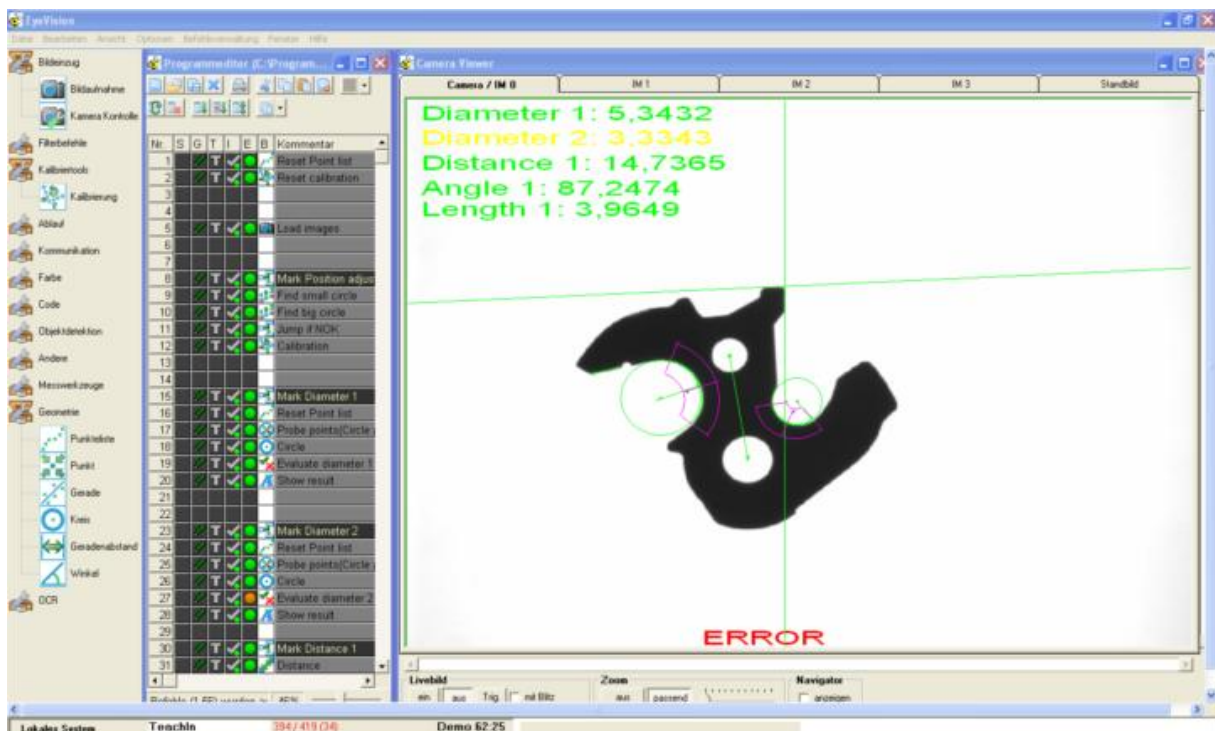


对象中心被当作后面命令的新的原点。



根据新的圆点，调整评价对象命令的窗口。物体颜色变为白色。

例 2: 位置调整，校正和测量



重置点列表。



选择缩放因子。图片从像素缩放到坐标图像。



选择图像 Training_1_00000.jpg



用设置标记（Set mark）命令设置一个标记。在此例中我们将标记命名为“Position ad-

justment”。



在组件中寻找小洞，定义搜索颜色为白色（用吸管点击相机取景器中的白色区域）。为了寻找小的孔，你要定义区域。激活对象过滤器（Object filter），设置最小值为16，最大值为19。现在只会找到小的孔。



如果要寻找大的孔，你需要重复上面相同的操作。只是把定义的最小值设定为32，最大值为35。



如果少了一个孔就跳到错误。要做到这一点，点击如果 NOK 时跳到标记（Jump to mark in case “NOK”）。在此例中我们将标记命名为 “Wrong number of objects”。



定义新的圆点和方向。要做到这一点只要点击圆点和方向（Origin and Orientation），旋转的方向选择 CCW。



设定一个新标记。这次我们命名为 “Diameter 1”。



重置点列表。



在小的开放的半圆形上配置一个圆探头。选择对象颜色为白色。（双击 G170）。



可以通过点列表中的值指定一个最匹配的圆。



现在可以用圆的直径与计算到的圆直接做对比（标称值是5,33; 容差是 +0,03 +0,00, 警戒限是 +0,02 -0,00）。



现在显示圆的直径。选择值并在文本框中输入 “Diameter 1”。



设定标记 (Diameter 2)。



重置点列表。



在大的开放的半圆形处配置环形探头。



可以通过点列表中的值指定一个最匹配的圆。



现在可以用圆的直径与计算到的圆直接做对比（标称值是3,31; 容差是 +0,03 +0,00, 警戒限 +0,02 -0,00）。



现在显示圆的直径。选择值并在文本框中输入“Diameter 2”。



设定标记 (Distance 1)。



定义两个点之间的距离。从选取列表中选择第一个点和第二个点。



现在显示距离。选择值并在文本框中输入“Distance”。



设定标记 (Angle 1)。



重置点列表（删除所有已经存在的点）。



在直线边缘配置一个探头（对象颜色为白色）。



确定一条通过几个点的最佳直线。



重置点列表（删除所有已经存在的点）。



在第二条直线边缘配置一个探头（对象颜色为白色）。



确定一条通过几个点的最佳直线。

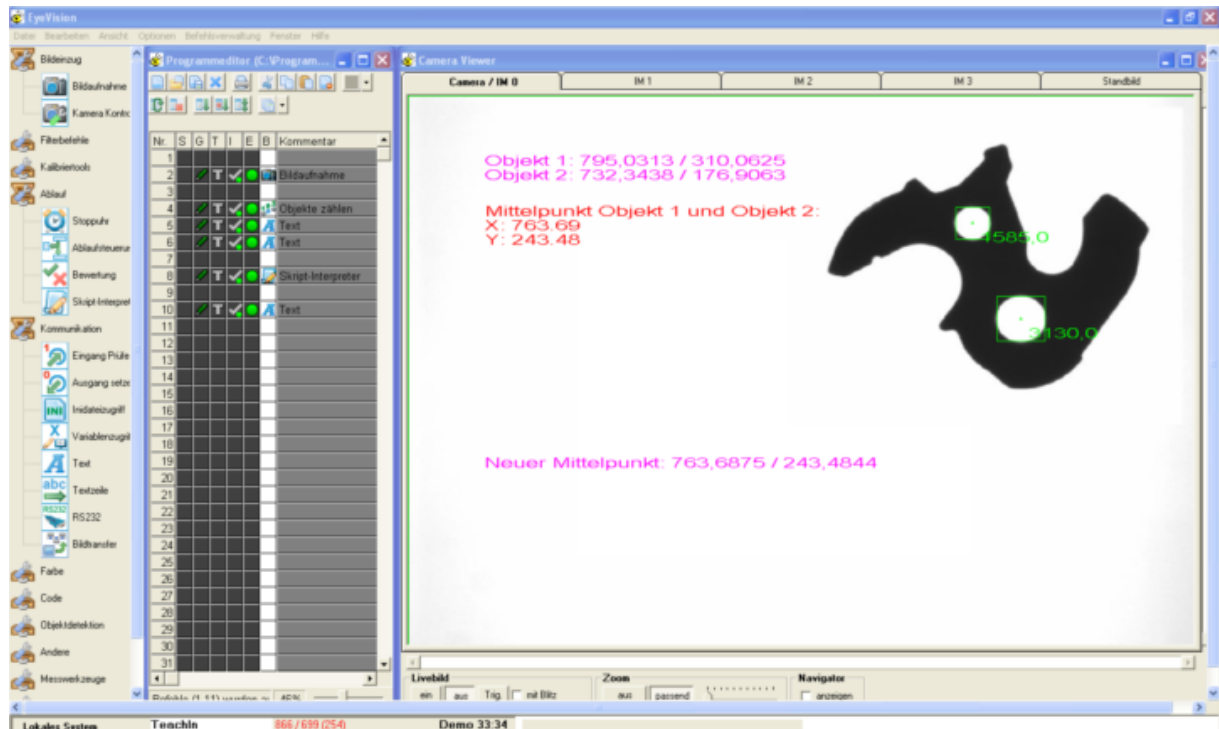


可以测量两条直线间的角度。



显示角度。(Angle 1)

例 3: 查找对象，确定中心点



图像捕获。这里，我们从本地选择一个图片。此便中选择 Training_1_00000.jpg，并在加载图像中选择循环（cyclic）。



对象计数器配置成搜索白色，对象过滤器设置成1400 – 3200，就剩下两个对象。



显示对象的主要中心。（源：寄存器 0）。选择点并在文本框中输入“Object 1”。



显示对象的主要中心。（源：寄存器 1）。选择点并在文本框中输入“Object 2”。



计算寄存器0和1中心点的中间点，保存值到寄存器0（对应源文本如下）。



显示寄存器 0 的值。（“new midpoint”）。

基本指令可以与其他计算机图形指示合并。通过功能界面，你可以访问程序库的所有功能，你也可以传递结果给其他功能。下面的脚本就是一个例子，显示如何直接访问的寄存器值，结果值可以用 BASIC 再编译。关于 BASIC 编译器和功能详述将在本手册结尾看到。

Program: ManCalcCenterPointBASIC.ckp
Source code of the BASIC Program:

```
Dim dPX1 As Double
Dim dPY1 As Double
Dim dPX2 As Double
Dim dPY2 As Double
Dim sBuffer As String

sBuffer = "Name of Point 1"

CKGetPointRegister(0, dPX1, dPY1, sBuffer)

sBuffer = "Name of Point 2"

CKGetPointRegister(1, dPX2, dPY2, sBuffer)

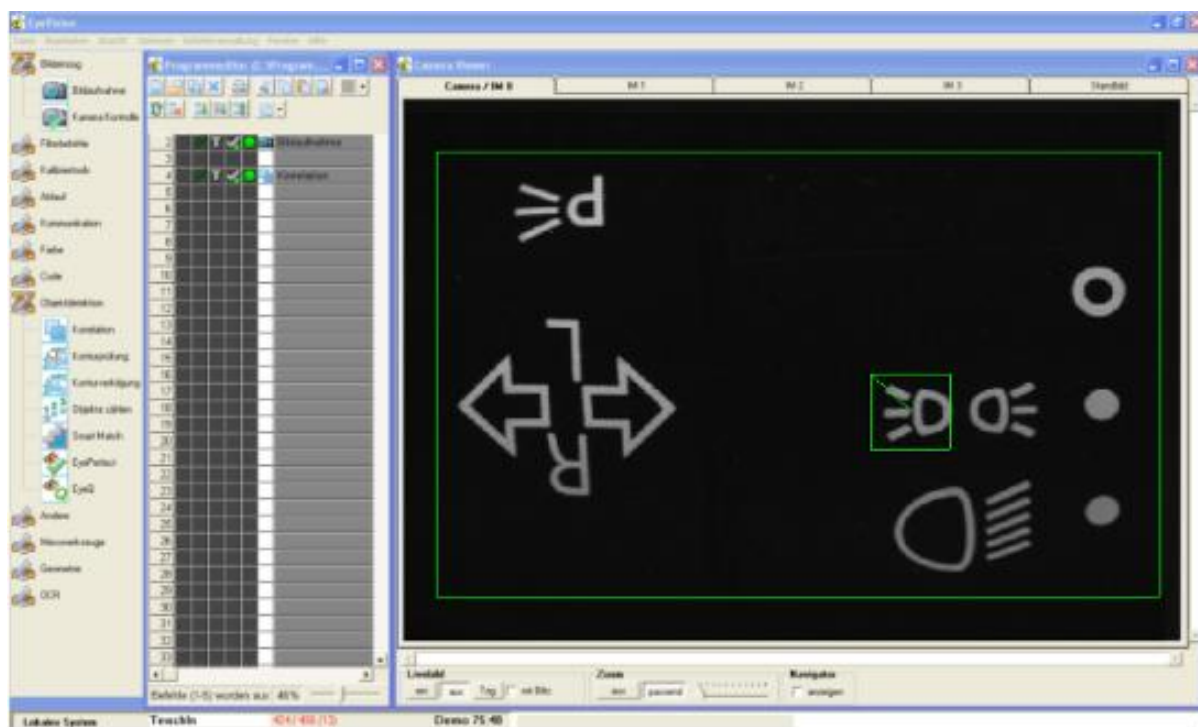
CKSetPointRegister(0.5*(dPX1 + dPX2), 0.5 * (dPY1 + dPY2), "Mittelpunkt_O1_O2")

EVHDGOIWriteString(0,0,0, 100,150,"Mittelpunkt Objekt 1 und Objekt 2:")

EVHDGOIWriteString(0,0,0, 100,170, "X: " & 0.5*(dPX1 + dPX2))

EVHDGOIWriteString(0,0,0, 100,190,"Y: " & 0.5*(dPY1 + dPY2))
```

例 4: 图形识别

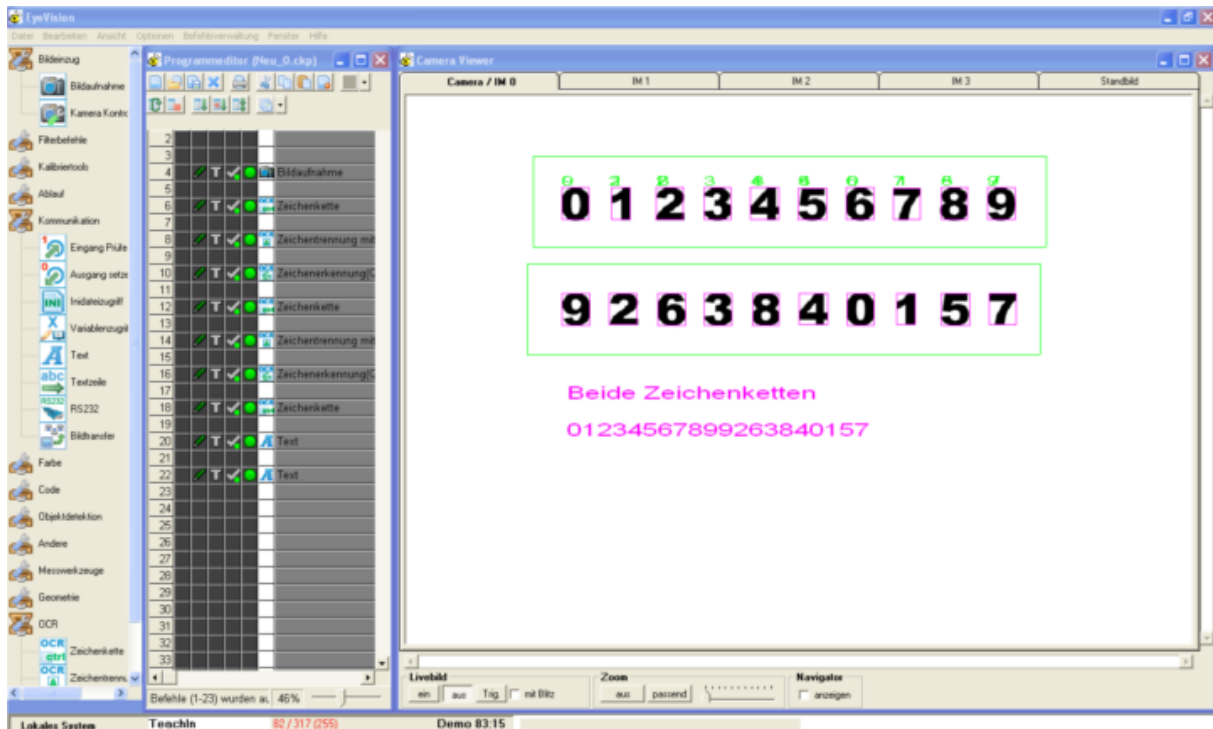


图像捕获。从文件夹“Correlation”中选择图像“CarCntrl_080426_001529_173.jpg”。



用相关性命令在图片中查找配置的图形。可以是由图形编辑器创建的灰度图像。

例 5: 读码



图像捕获。从本地图像文件夹中选择图像。此例中选择的是 OCR 文件夹中的图像 ManOCRDemo.jpg。



选择：OCR 系统。



在此我们为要读取的图像选择高字符串（通过窗口），然后定义阈值（0...216）。



通过加载程序 Man OCRTech.cgh，或点击选择的数据特征，激活图形探测和符号集文件和保存新程序。命名程序，如 New.cgh。



在全局字符串缓冲区输入字符串。选择显示/配置。确认后配置（执行配置模式）。0以下手动配置和1以下手动调整，然后执行配置模式以确定。

建议确认后关闭命令（点击 S 列），否则始终显示配置窗口。但不要忘记最后要重新激活命令。



在此选择读取范围的低字符串（不要忘记调整阈值）。



激活字符识别和符号集（再次选择 ManOCRTeach.cgh 或自命名的程序）。



单击在文本行输入（*Entry in Text line*）在全局字符串缓冲中增加文本字符串。

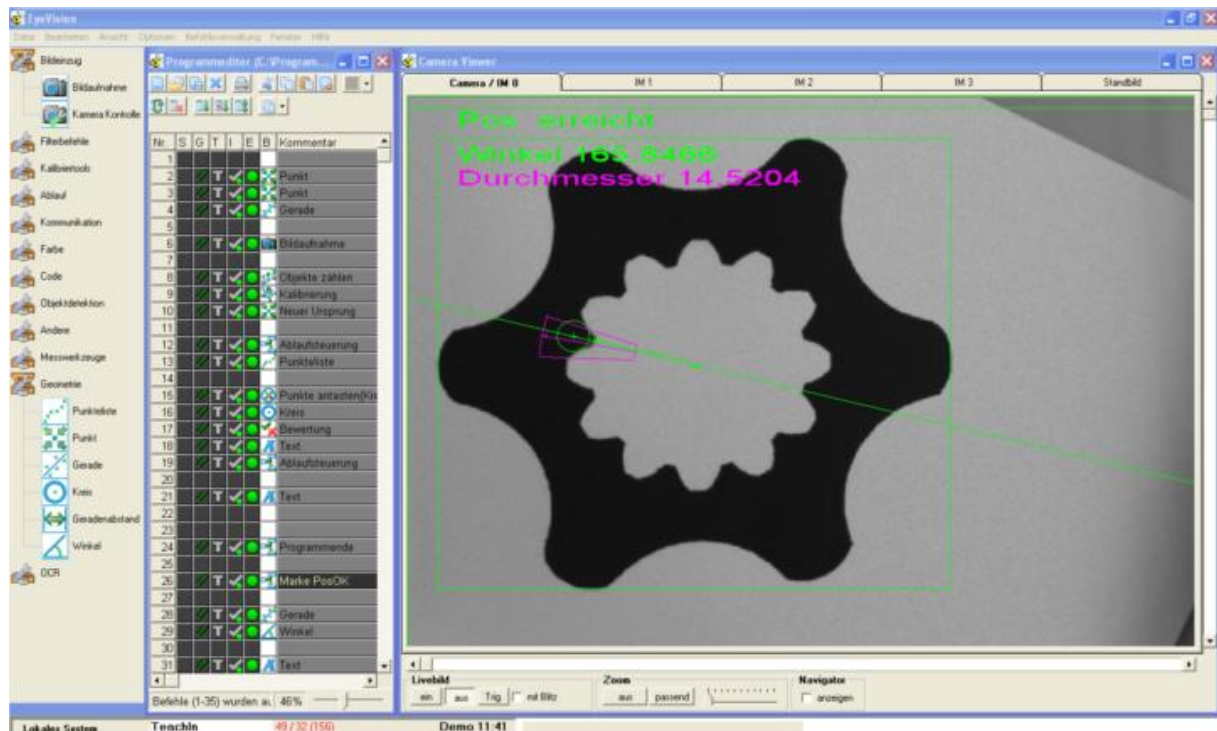


选择只有文本（*Only text*）在文本行输入 “Both character strings”。用户也可以调整文本位置。在上例中，文本位置是 X: 163 Y:313.



在此选择文本行和位置（如 X:163 Y:353）。在指定位置显示数字。

例 6: 对象方向



在此定义两个点和一条平行于 x 轴通过两条直线的直线。这条直线用于后面计算角度的参考直线。



文件夹 *Shaft Ring* 中的图像 *ShaftRing_080416_124732_296.jpg*。循环加载整个文件夹的图像。即使要加载整个文件夹也要点击第一个文件，否则不会加载文件夹。



寻找环并确定重心。根据亮度，设置范围在0-71.点击对象过滤 (*Object-filter*)，选择面积 (*area*) 指定最小面积1000，最大面积100000。现在在图像中检测到环。(点击检查按钮查看结果)



用重心0, 0定义一个新的坐标系统。



定义圆点为直线上的点 (圆点坐标为 $X:0$ and $Y:0$)。



定义整体结果为 IO。



重置点列表。



在环上配置圆形探头。（在图像中圆形探头是粉色的）（容差+ 10000）。



现在计算圆的直径（最佳的圆）。



检查圆的直径是否与标准值相同（标称值： 14,4， 容差： +0,06 -0,06）。



显示半径。



如果直径是 IO，则跳转到 OK 标记。



否则显示 NOK。



跳到程序结尾，并再次从程序开始处执行。



从选取列表选择两个点，定义一条穿过两点的直线。一个点是对象重心，一个点是圆形上的点。



计算基础直线（注意：在最开始，做图像捕获前定义的直线）与刚刚定义的直线间的角度。



显示定义的角度。



显示文本位置 OK。

6 配置支持

6.1 光标形状

如果你在不同的元素或子域的配置元素间移动光标，光标形状会改变。根据显示的形状，可以转移整个元素或改变其形状。

我们通过圆形探头的案例演示。



如果光标显示为此图形，可在图像中随意移动圆形探头。



如果光标显示为此图形，可通过按压鼠标左键改变圆形探头的半径。



如果光标显示为此图形，你可以改变或旋转扇形的角度。



可以用吸管直接从图像中把彩色或灰度值传送到不同的配置形式中。为此，首先要点击滴管图标，光标变成吸管形状，然后你可以单击图像中的位置与所需要的信息（亮度，饱和度等）。这些值自动被保存。

7 接口协议

软件安装了不同的标准接口协议。如果您遵守这些协议的要求，系统的连接过程是非常容易的。用户除了使用标准的协议，也可以使用自己的协议。对于这些用户自定义协议的执行情况，有各种输入和输出的软件指令。对于非常复杂的协议，最好使用集成在软件中的 VB6兼容的 BASIC 编译器，通过它，也可以访问系统接口。

7.1 数字 I/O

数字接口仍然是最常见的通信接口。它的普及可能是因为电缆是针对每个信号的，而传输的信号要么是1（24V 信号），要么是0（0V 信号）。如果需要多个信号，需要增加对应的信号线。为了简化整合，在 SmartBuilder 软件中集成了一些分配信号的标准程序。

通过数字 I/O 的程序转换

自版本2.0，您可以使用相机的数字输入和输出进行自动检查程序转换。在当前版本，检测程序转换是由用户界面配置的。在查看→应用参数→自动程序控制中，程序控制可以自动配置。

下面，我们再附加地描述如何通过改变中央配置文件 EVHD.CFG 手动控制检测程序。配置文件在要配置的相机中，例如在

C:\Programme\VizCam\SmartBuilder\Devices\YOUR_DEVICENAME\Current\parameters.

在配置文件中，[PROGRAMSWITCH]包括一些参数，如自动化，串行检测程序转换和通过输入输出的检测程序转换等。

以下输入对于数字输入/输出转换非常重要：

1) PS_DIO_ProgramControlActive

如果输入值是1，激活转换。任何其他值都关闭转换。

2) PS_DIO_UseHandshake

如果输入值是1，执行检测程序转换硬件握手（详情请参阅下文）。其他值不执行硬件握手。相机的数字输出数字0以及数字输入数字0用于硬件握手，因此他们并不适用于其他功能或者说其可用性是有限的。

3) PS_DIO_HandshakeTimeout

此输入值决定了硬件握手的最大信号保持时间。此值应以毫秒表示。（即100毫秒= 0.1秒）

4) PS_DIO_NumberOfSignalLines

此项的值表示的是用来选择检测程序数字的信号输入数量。用 n 条线输入，可以编辑 2^n 个不同的检测程序。这意味着，2个测试程序用1条电缆，4个程序用的2条电缆，8个程序用3条电缆等。以这样的方式，使用的数字输入不能被用于其他任务。请大家注意，相机输入的最大数量。还请注意，硬件握手已使用了一个数字输入，这意味着它不再用于其他的检测程序。

5) PS_DIO_CheckProgramPrefix

在这里，可以为检查程序的名称加一个前缀。默认值是自动。文本字符串不应该超过四个字符，否则根据相机版本不同，有可能引起相机文件系统出错。

转换的运行模式（没有硬件握手）：

完成每个当前调整的检测程序后，要根据转换的要求检查数字输入。如果当前的检查程序正在等待触发信号的循环，那么不可能转换。

检查过程中，数字输入数0到数（PS_DIO_NumberOfSignalLines-1）的状态，以二进制数字记录和编译（LSB 从输入数0），而且要把结果值与当前调整做对比。

如果启动改变检测程序转换的值。要被再次激活的程序的名称由 PS_DIO_CheckProgrammPrefix，如“**AUTO**”，生成的数值，如“**2**”，和检查程序结束“**.CKP**”，因此就成为“**AUTO2.ckp**”。然后加载这个名称的检查程序。这是用户的责任，所有检查程序可以自动调用，并以正确的名称保存在相机中。

硬件握手

如果相机检测到程序转换的要求，硬件握手的操作会反馈给呼叫者。

产生以下步骤：

1. 呼叫方调整与检查程序相结合的数字输入值1到数字 PS_DIO_NumberOfSignalLines。
2. 之后呼叫方调整相机的数字输入数字0为“高”，注册程序转换的要求。
3. 相机记忆输入的状态（模拟上述“转换的运行模式”的文章（无硬件握手））。
4. 相机通过调整数字输入数字0到高，注册程序转换要求的接收，并等待反馈。
5. 为了确认检查程序的修改，呼叫方要在 PS_DIO_HandshakeTimeout 毫秒时间内把相机输入值从0调回低。如果相机没收到确认（或太晚），检查程序的改变将被拒绝。
6. 相机通过调整其输出数从0到低，发信号确认接收。
7. 类似上面的“切换运行模式（无硬件握手）”，生成检验程序的名称和加载新的检查程序。

重要：

- 用于检查程序转换的输入端口和输出端口，不应该被用于其他任务！
- 请注意，相机有最大的输入的数量。请考虑，已有一个输入用于硬件握手，这样检查程序不可使用这个数字输入。
- 如果相机处于待机运行状态（如触发，其他数字输入/输出，串行数据，秒表等），在此期间，不可以修改检查程序。
- 如果检查程序生成的名称指向不存在的检查程序，加载失败。然后相机等待用户的新指令，并不会继续旧的检查计划。
- 一个成功的硬件握手不确认新的检查程序被加载并成功执行，但接收到改变的要求，并进行评估。

7.2 (RS232, RS485)

通过串行接口控制检查程序

从2.0版本开始，可以使用相机的数字输入和输出进行自动检测程序的转换。在当前版本，转换可以在图形用户界面配置。用查看（View）→ Application Parameters（应用参数）→ 自动程序的控制（autom. Program control），可自动配置检查程序的控制。

以下，我们会描述如何通过修改中央配置文件 EVHD.CFG，手动处理检查程序控制的参数。配置文件是在相机的配置之一，例如，C:\Programme\VizCam\SmartBuilder\Devices\YOUR_DEVICENAME\Current\Parameters

在配置文件中的一节[PROGRAMSWITCH]，包含自动，串行检查程序转换和对应地通过数字输入/输出做检查程序转换的参数。

以下条目对于通过串行数字接口控制是很重要的：

1. PS_SerialProgramControlActive

如果此项的值为1，专有协议的检查被激活。

2. B PS_StopBlockingProgramOperations

如果该条目值是1，正在进行的检查程序离开或忽略阻止待机运行（如触发，其他的数字输入/输出，串行数据，秒表等），直到下一次从接收缓冲区读到串行数据。在任何情况下，这是为了针对串行控制应用给予反应。如果该值设置为0，当前检测程序始终保持不变，直到执行一个串行控制应用程序。

控制系统协议

串行控制电缆通常包括起始符号“#”，三位命令数，如“001”和之后最多200位的数据位（可选）。命令以控制字符“#”结尾。

当前存在以下命令：

模式“Stop”：

电缆：#003#

功能：完成最后的任务后停止目前的检查程序。新的检查程序不会自动启动，直到调整“开始”模式。

模式“Start”：

电缆：#002#

功能：启动当前检验程序。“开始”的模式被保留，即使不存在或没有可读的测试程序进行调整。检查程序改变后立即启动新的检验程序。

模式“Inspection program change”：

电缆：#001Std.ckp#

功能：指示电缆改变成检查程序 Std.ckp。在“开始”模式，它立即开始连续工作。在“停止”模式，它仅加载。文件扩展名“.CKP”是强制性的。

不通过串行接口发送确认或确认状态，以便用户在检测程序中使用 RS-232发送命令。

7.3 LAN 协议（以太网）

通过 LAN 控制检查程序

有两种不同的可能性，可通过 LAN 控制检测程序：一方面可以在 SmartBuilder 的软件和相机间实现通信协议。这是比较复杂的，它的使用要遵循限制和访问特性（如在拨号时立即停止检测程序）。

另一方面，通过 UDP 控制检测程序有非常简单的专有协议。如下述。

通信端口的数量比为 SmartBuilder 连接的端口号高1。IP 地址与在相机连接时的 IP 地址相同。

控制系统协议

串行控制电缆通常包括起始符号“#”，三位命令数，如“001”和之后最多200位的数据位（可选）。命令以控制字符“#”结尾。

目前存在以下命令：

“停止”模式：

电缆：#003#

功能：完成最后的任务后停止目前的检查程序。新的检查程序不会自动启动，直到调整“开始”模式。

“开始”模式：

电缆：#002#

功能：启动当前检验程序。“开始”的模式被保留，即使不存在或没有可读的测试程序进行调整。检查程序改变后立即启动新的检验程序。

“检查程序改变”模式：

电缆：#001Std.ckp#

功能：启动当前检验程序。“开始”的模式被保留，即使不存在或没有可读的测试程序进行调整。检查程序改变后立即启动新的检验程序。

用户会收到一个确认响应：

无论是“OK”，如果可以完全解析电缆或“NOK”如果电缆确实没有适用的版本，但不知道命令代码（例如#004#）。如果电缆不能被解析，相机向发送人返回收到的电缆（调试用）。

重要附加：

为了避免不一致，SmartBuilder 软件总是优先，以避免通过 LAN 同时使用 EyeVision 软件和自动检查程序转换。如果在 SmartBuilder 的软件和相机间有一个访问，相机会忽略所有通过 LAN 接口收到的检查程序控制电缆。在这种情况下，用户会通过 LAN 接口收到确认“IGNORED”（不是“OK”或“NOK”）。

8 BASIC 编译器

SmartBuilder 的 BASIC 编译器，允许用户扩展图形用户界面，这样，可实现一些从未想到的解决方案。对我们来说，BASIC 编程语言的句法最理想，最容易学习。

我们面向 Microsoft Visual Basic 获得基本的语法，因为这个语法广为流传。因此，这对于有此语法经验的开发者寻找解决方案是非常简单。同时，我们为您提供一个全面的教科书学习语法。

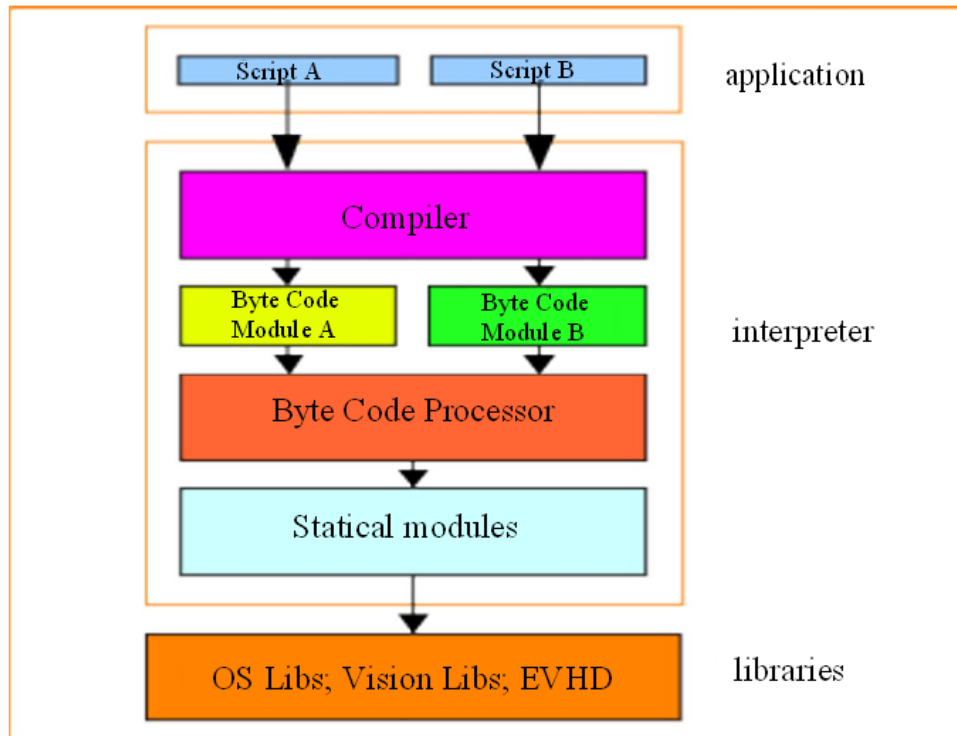
SmartBuilder 的功能程序库可以直接在 BASIC 中打开，这样可以在这个编程语言中实现更多复杂的图像处理情况，或编写复杂的宏。

在各种情况下，使用这些工具编写的程序可以作为一个独立的结构，集成到图形程序序列。因此，不需要放弃舒适的图形配置，就可以实现复杂的应用程序。

被嵌入的图形说明和评估程序，可以通过 SmartBuilder 软件的各种数据结构交换数据。

8.1 SmartBuilder BASIC-编译器

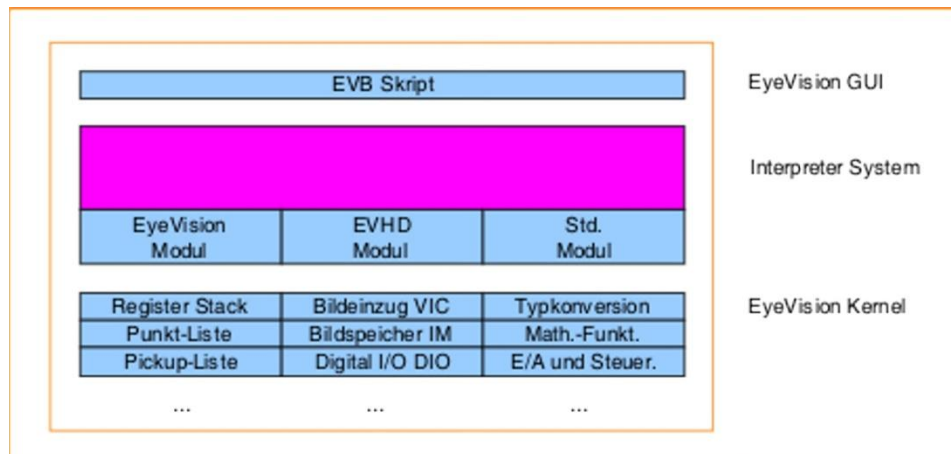
SmartBuilder BASIC 编译器是软件的组成部分，下图显示了编译器的整体结构和功能。



应用程序可以使用由脚本编译器来控制某些任务。这些脚本写在 **Eye Vision Basic** 中并加载在编译器上。编译器将源代码转换成快速可执行文件（字节码），并归档到一个模块。每个模块包含的函数和变量，可以用于其他模块。如果执行一个模块的功能是由应用程序引起的，处理器开始执行适当的代码，并在终止脚本后再次返回应用程序控制。静态模块有一个例外。他们没有一个字节代码，但嵌入了翻译的处理器代码。可以在编译器内访问用 **C** 语言编写的程序库。静态模块构成与外部软件的连接作用。

8.2 SmartBuilder BASIC 检查命令

检查命令是编译器的一个应用。可以为每个检查命令加载脚本，然后根据位置将这些命令粘贴到检测程序中。编译后在检查命令中填入字节码，意味着执行时这些代码立即可用。一个没有编译器的目标平台，节省内存和代码。这尤其简化了嵌入式系统的发展。



用这个模块可以访问 SmartBuilder 许多内部功能，如寄存批次，选取列表中的对象。与模块 SmartBuilder 访问。也可以访问 Eye Vision Virtual Video Device（模块 EVHD），使用户直接开发更多检查应用。

用户界面功能

脚本解释器象其他检测命令一样插入检查程序中。命令表中包含一个编辑器，写状态信息的文本框和用于导入导出的按钮，及确认和拒绝调整的按钮。

8.3 SmartBuilder BASIC 语言(EVB)

SmartBuilder BASIC 是一个基本的语言，它基于 Visual Basic 6，但不具有面向对象的语言元素。因此，该程序构成最大的语言单位。一个方案包括声明和说明。

数据类型

有以下几种基础的数据类型：

数据类型	描述	内存需要
Void	没有任何信息的抽象数据类型	0 Byte
Long	32 位带符号整数	4 Byte
Double	64 位 IEEE 浮点值	8 Byte
String	8位字符串	mind. 1 Byte

常量

支持以下常量表示。

模范常数	数据类型	值
123	Long	123
123.0	Double	123.0
123.4	Double	123.4
false	Long	0
true	Long	-1
„Basic!“	String	„Basic!“ 0

变量

标量变量的声明有以下形式：

```
Dim Variable identifier As Type identifier
```

变量标识符必须包含所有字母数字字符（包括 A-Z，a-z，0-9），包括下划线（“_”），但绝不能以数字开头。

例子：

```
Dim index As Long  
Dim xValue As Double
```

阵列

提出一个新的数组，并相应地做标注：

```
Dim Variable identifier(dimension) As Type identifier
```

尺寸超过，数组可用下面的命令批准：

```
Erase Variable identifier
```

例：

```
Dim pointsX(100) As Double  
Erase pointsX
```

操作数

支持以下操作数:

Prec.	Operator	描述	操作数	结果
14	Imp	Implication bit by bit	Long	Long
13	Eqv	Equivalence bit by bit	Long	Long
12	Xor	Exclusive-or operation bit by bit	Long	Long
11	Or	Disjunction bit by bit	Long	Long
10	And	Conjunction bit by bit	Long	Long
9	Not	Negation bit by bit	Long	Long
8	>=	Comparison: greater-or-equal	Long/Dbl./Str.	Long
8	<=	Comparison: less-or-equal	Long/Dbl./Str.	Long
8	>	Comparison: greater	Long/Dbl./Str.	Long
8	<	Comparison: less	Long/Dbl./Str.	Long
8	<>	Comparison: not equal	Long/Dbl./Str.	Long
8	=	Comparison: equal	Long/Dbl./Str.	Long
7	&	String-concatenation	String	String
6	+	Addition	Long/Double	Long/Double
6	-	Subtraction	Long/Double	Long/Double
5	%	Modulo	Long	Long
4	\	Division of integers	Long	Long
3	*	Multiplication	Long/Double	Long/Double

3	/	Division	Long/Double	Long/Double
2	+	Positive sign	Long/Double	Long/Double
2	-	Negative sign	Long/Double	Long/Double
1	^	Exponentiation	Double	Double

例:

```
... 3 + 4 ...
... index - 4 ...
... "x has the value" & pointsX(0) ...
```

类型转换

在所需的位置隐蔽地自动插入类型转换。因此在类型 `Long`, `Double` 和 `String` 间转换是可能的。库函数的 `Clong`, `Cdouble` 和 `CString` 完成明确的类型转换。

例:

```
... 3 + 4.0 ...           (result of the type Double)
... Clong(„123“) * 2 ... (result of the type Long)
```

分配

变量分配具有以下形式:

```
Variable identifier = expression
```

如果变量在数组中:

```
Variable identifier (index) = expression
```

例:

```
Index = 45
pointsX(0) = 123.4
```

If 语句

有两种不同的如果结构。单行选项：

```
If expression Then ... [ Else ... ] [ End If ]
```

复杂的字符串选项：

```
If expression Then  
...  
[ ElseIf expression Then ]  
...  
[ Else ]  
...  
End If
```

例：

```
If axis = 1 Then  
    c = x  
Elseif axis = 2 Then  
    c = y  
Elseif axis = 3 Then  
    c = z  
Else  
    ' Unknown  
End If
```

Select 语句

Select case 语句有以下布局：

```
Select Case expression  
    Case expression, ...  
    ...  
    Case Else  
    ...  
End Select
```

例：

```
Select Case color
  Case 1, 9
    ' shades of blue
  Case 2, 10
    ' shades of green
  Case 4, 12
    ' shades of red
  Case Else
    ' any other
End Select
```

For 循环

For 循环采用以下形式：

```
For Variable identifier = lower limit To upper limit [ Step step size ]
...
Next
```

如果步长等于1，可以省略关键字的步骤和值的指示。对循环体内的以下调用，立即放弃内部的 For 循环：

```
Exit For
```

例：

```
For index = 0 To 9
  sumX = sumX + pointsX(index)
Next
```

Do 循环

Do 循环采用以下形式：

```
Do While condition
...
Loop
```

进入循环时，要测试条件。循环中的命令被执行，直到不再符合条件为止，即该值等于 **false**。为了在第一个循环后测试条件，**DO-while** 循环要转化成 **while** 循环：

```

Do
...
Loop While condition

```

至少有一个循环在运行。根据上述定义，创建 **Do-Until** 和 **Loop-Until** 循环来代替 **While**。从而否定条件，即，如果条件为真，循环被放弃。循环内如果有下述命令，立即放弃 **Do** 循环：

```

Exit Do

```

While 循环具有以下格式，与 **Do-While** 循环相同：

```

While condition
...
Wend

```

例：

```

Do While index <= 9
    index = index + 1
Loop

```

功能

函数调用有一个类似数组的访问形式：

```

... function name (arguments, ... ) ...

```

如果不需要函数值，可使用一个调用指令：

```

[ Call ] ... function name (arguments, ... ) ...

```

例：

```

... getDistance(x1, y1, x2, y2) ...
Call setVersion(1, 0)

```

```
setVersion(1, 0)
```

Set 函数

SmartBuilder 的基本语言就是 SET 函数。如果函数名以字符串 “_set” 结束，而且其返回类型被定义为 “无效”，此函数可以被称为分配语法。你应该注意，如果一个变量声明产生歧义，在一个模块内不称为 SET 函数。

例：

```
Function myProperty_set(lIndex As Long, dValue As Long)
...
End Function

myProperty_set(123, 456.7) ' regular call
myProperty(123) = 456.7 ' set-call
```

模块

每个程序都相当于一个模块，包含源代码中定义的变量和函数。可使用其他模块的函数。

例：

```
Dim i As Long
For i = 1 To 10
    Print(i) ' defined in Module Std
Next
```

8.4 编译器

编译器的错误信息

函数必须返回一个值

如果试图评估在表达式中的函数调用，但函数不返回值（返回值无效）时发生此错误。在这种情况下，要使用调用命令。

未定义标识符 **x**

试图分配一个变量或函数给标识符 **x**。但没有发现匹配的定义。

无法加载左边表达式 **x**

如果没有指定标识符 **x** 是只读访问，如只存在一个匹配的 **set** 函数。

不能保存左边表达式 **x**

试图访问标识符，但无法找到匹配的 **set** 函数。

数组 **x** 采用1维（不是 **y**）

因为数组只有一维，如果数组有多个下标表达式时发生此错误。

变量 **x** 没用任何参数（不是 **y**）

如果因为变量没有参数，一个标量变量使用参数时发生此错误。

函数 **x** 使用 **y** 的变量（不是 **z**）

函数调用了错误的参数。请注意 **set** 函数！分配的值被认为是一个额外的参数。

不能保存常量 **x**

试图给常量分配值时发生此错误。

没有循环语句退出

在 Do-While/Until，Do-Loop-While/Until 或 while 循环以外，不能用 Exit Do 中止循环。

没有 for 语句退出 for

在 For 循环以外不可以使用 Exit For 退出循环。

语法错误

语法错误发生在程序分解过程。究其原因，可能是例如无效的标识符名称，无效的常量，无效的关键词或不正确的语句嵌套。

处理器的错误信息

无例外

程序在运行。

程序终止

程序被正确地终止。

无效命令类型

处理器发现一个无效的指令。这可能是传输字节代码时，由于内存损坏或字节码版本与处理器不兼容产生的错误。

无效类型修改器

处理器发现一个指令未按当前数据类型定义。这可能是传输字节代码时，由于内存损坏或字节码版本与处理器不兼容产生的错误。

无效模块

试图调用一个当前不可用的函数。

无效函数

试图调用一个当前不可用的函数。

无效函数类型

试图调用一个当前不可用的函数，因为执行的平台，例如不支持动态库。

堆栈下溢

有一个有效的堆栈范围以外的访问。这可能是传输字节代码时，由于内存损坏或字节码版本与处理器不兼容产生的错误。

数组出界

试图访问定义限制范围外的数组元素。该指数最小为0并小于维度。

8.5 模块概览

在以下方案的基础上创建以下参考：

函数的声明		
函数描述		
参数类型	参数	参数描述
...		
返回类型		返回描述

SmartBuilder 图形编译器模块

图形检查程序通过这个模块与 EVB 互动。

函数 JumpToMark (sJumpMark As String)		
终止执行脚本，模块跳转到检查计划规定的标志。		
String	sJumpMark	跳转标记名称
Void		

函数 JUMP_MARK As String		
产生终止脚本后跳转到的标志。		
String		跳转标记名称

函数 JUMP_MARK_set (sJumpMark As String)		
设置脚本终止后跳转到的标志。		
String	sJumpMark	跳转标记名称
Void		

函数 SCRIPT_RESULT As Long		
提供脚本的返回值。		
Long		Return Value

函数 SCRIPT_RESULT_set (lScriptResult As Long)		
设置脚本的返回值。		
Long	lScriptResult	Return Value
Void		

函数 EVHD_SUCCESS As Long		
成功时函数的返回值		
Long		0

函数 EVHD_ERROR As Long		
失败时函数的返回值		
Long		-1

函数 EVHD_WARNING As Long		
警告时函数的返回值		
Long		1

EVb 模块

以下将说明这些模块。它是关于可从 **EVb** 调用的库函数。函数模块列表不完整，除了以下列出的函数，还有更多的函数。你可以在软件的高级手册和应用程序界面得到更详细的介绍。

所有函数和它们的描述是用英文写的，因为 **EVb** 开发的程序，不依赖于各自国家的语言。

函数名称
函数参数
函数描述
参数 返回值 附加信息
例子

8.6 标准函数

标准模块

函数 Print (sText As String)		
在标准控制台上释放文本输出。		
String	sText	Text output
Void		

函数 CLong (IValue As Long) As Long		
执行一个显示类型转换到 Long。		
Long	IValue	Value
Long		Return value

函数 CDouble (IValue As Double) As Double		
执行一个显示类型转换到 Double.		
Double	IValue	Value
Double		Return value

函数 CString (IValue As String) As String		
执行一个显示类型转换到 String.		
String	IValue	Value
String		Return value

函数 Pi As Double		
提供数字常量 Pi.		
Double		3.1415926 ...

函数 Sin (dAngle As Double) As Double		
提供所指角度的正弦值。		
Double	dAngle	Angle in radiant
Double		Function value

函数 **Cos**(dAngle As Double) As Double

提供所指角度的余弦值。

Double	dAngle	Angle in radiant
Double		Function value

函数 **Sqr**(dArgument As Double) As Double

计算参数的平方根。

Double	dArgument	Argument
Double		Function value

函数 **Int**(dValue As Double) As Long

参数去小数向下取整数。

Double	dValue	Argument
Long		Function value

函数 **Fix**(dValue As Double) As Long

参数去小数向下取整数。

Double	dValue	Argument
Long		Function value

函数 **Round**(dValue As Double) As Long

参数去小数向上取整数。

Double	dValue	Argument
Long		Function value

字符串模块

函数 **Val** (sValue As String) As Double

函数返回作为数值包含在字符串中的数字。

String	sValue	Argument
Double		Function value

函数 **Str** (dValue As Double, IPrecision As Long) As String

包含指定小数精度的 dValue 字符串。

Double	dValue	Argument
Long	IPrecision	

函数 **Mid** (sValue As String, IStart As Long, ILength As Long) As String

函数返回包含指定字符的字符串。

String	sValue	Argument
Long	IStart	If IStart is bigger than the length of sValue, the function returns an empty String.
Long	ILength	Number of characters in the substring.
String		Substring beginning at IStart with ILength characters.

函数 **Left** (sValue As String, ILength As Long) As String

此函数返回一个包含 sValue 中指定的字符数的子串。它从第一个字符开始。

String	sValue	Argument
Long	ILength	Number of characters in the substring.
String		Substring begins with the first character in sValue with ILength characters.

函数 Right (sValue As String, ILength As Long) As String

此函数返回 sValue 中含有 ILength 字符的子串。

String	sValue	Argument
Long	ILength	Number of characters in the sub-string.
String		Substring contains the last ILength character of sValue.

函数 Right (sValue As String, ILength As Long) As String

此函数返回 sValue 中含有 ILength 字符的子串。

String	sValue	Argument
Long	ILength	Number of characters in the sub-string.

函数 GetSendString () As String

此函数返回 SmartBuilder 当前发送的全局字符串。您可以使用此功能管理字符串或条码，OCR 或数据矩阵码命令或任何其他字符串命令。

		Argument
String		The String contains the current global String that was sent after having called the function.

函数 SetSendString (ByVal sValue As String) As Long

此函数设置当前 SmartBuilder 发送的全局字符串。用户可以使用这个功能来管理字符串。操纵这个字符串之后，才有可能通过以太网或 RS232 把字符串发送给任何通讯伙伴。

注：此功能将立刻改变 SendString。调用 SetSendString 后，GetSendString 会返回新的 SendString。

String	sValue	sValue contains the current global
Long		EVHD_SUCCESS

Print		
Sub Print (sString Val As String)		
这个 Sub 将 sString 定入控制台。		
String	sValue	sValue contains the current global String that was sent after having called the function.
Long		EVHD_SUCCESS

例：

```
Dim sString As String
Print sString
leads to an run time error. Please use:
Dim sString As String
sString = ""
Print sString
```

CLong	
函数 CLong (ILongArg As Long) As Long	
此函数返回 ILongArg 的 Long 值。	
Arguments	
ILongArg	
Return value	long

CDouble	
函数 CDouble (dLongArg As Double) As Double	
此函数返回 dLongArg 的 Double 值。	
Arguments	
dDoubleArg	
Return value	double

CString

函数 CString (sStringArg As Double) As Double	
计算 sStringArg 的 Double 值。	
Arguments	
dDoubleArg	
Return value	double

Pi	
函数 Pi () As Double	
此函数返回数值 PI。	
Arguments	no
Return value	double

Sin	
函数 Sin (dArgDouble As Double) As Double	
此函数计算以弧度为 dArgDouble 的 dArgDouble 的正弦值。	
Arguments	
dArgDouble	
Return value	sin (dArgDouble)

Cos	
函数 Cos (dArgDouble As Double) As Double	
此函数计算以弧度为 dArgDouble 的 dArgDouble 的余弦值。	
Arguments	
dArgDouble	
Return value	sin (dArgDouble)

Sqr	
函数 Sqr (dArgDouble As Double) As Double	
此函数计算 dArgDouble 的平方根。	
Arguments	
dArgDouble	
Return values	Sqr (dArgDouble)

Int	
函数 Int (dArgDouble As Double) As Long	
此函数删除 dArgDouble 的小数部分并返回结果的 long 值。如果数字是负数，则 Int 返回小于或等于 dArgDouble 的第一个负整数。例如，Int 将-8.4转换为-9。	
Arguments	
dArgDouble	
Return values	Int (dArgDouble)

Fix	
函数 Fix (dArgDouble As Double) As Long	
此函数删除 dArgDouble 的小数部分并返回结果的 long 值。如果 dArgDouble 是负的，Fix 返回大于或等于 dArgDouble 的第一个负整数。例如 Fix 将-8.4转换为-8。	
Arguments	
dArgDouble	
Return values	Fix (dArgDouble)

Round	
函数 Round (dArgDouble As Double) As Long	
返回值是最接近的 dArgDouble 的有适当小数位的值。如果 dArgDouble 恰好是两个可以四舍五入值的中间值，此函数返回四舍五入后其右边的数字是偶数的数值。	
Arguments	
dArgDouble	
Return values	Round (dArgDouble)

Abs	
函数 Abs (dArgDouble As Double) As Double	
Abs 函数返回 dArgDouble 的绝对值。	
Arguments	
dArgDouble	
Return values	Abs (dArgDouble)
Additional Information	
Examples	

Atn	
函数 Atn (dArgDouble As Double) As Double	
此函数返回 dArgDouble 的余切值。	
Arguments	
dArgDouble	
Return values	Atn (dArgDouble)

Exp	
函数 Exp (dArgDouble As Double) As Double	
此函数计算 dArgDouble 的指数。	
Arguments	dArgDouble
Return values	Exp (dArgDouble)

Log	
函数 Log (dArgDouble As Double) As Double	
此函数计算 dArgDouble 的对数。	
Arguments	dArgDouble
Return values	Log (dArgDouble)
Additional Information	
Example	

Tan	
函数 Tan (dArgDouble As Double) As Double	
此函数计算 dArgDouble 的正切值。	
Arguments	
dArgDouble	
Return values	Tan (dArgDouble)

Rad	
函数 Rad (dArgDouble As Double) As Double	
此函数转换 dArgDouble（度）到辐射。	
Arguments	
dArgDouble	
Return values	Rad (dArgDouble)

Deg	
函数 Deg (dArgDouble As Double) As Double	
此函数转换 dArgDouble（辐射）到度。	
Arguments	
dArgDouble	
Return values	Deg(dArgDouble)

Pow	
函数 Pow (dArgDoubleA As Double, dArgDoubleP As Double) As Double	
数值 dArgDoubleA 升到 dArgDoubleP 次方。	
Arguments	
dArgDoubleA	The base number A
dArgDoubleP	The exponent number P
Return values	Pow (dArgDoubleA, dArgDoubleP)

debugMode_set	
Sub debugMode_set(IValue As Long)	
此函数设置调试模式。	
Arguments	
IValue	0: debug mode off
Return values	

SbiVersion	
函数 SbiVersion () As Long	
此函数返回 Script BASIC 编译器的版本号。	
Arguments	
Return values	

allocatedMemory	
函数 allocatedMemory () As Long	
此函数以返回系统分配的内存字节数。	
Arguments	
Return values	allocated memory in Byte

freeMemory	
函数 freeMemory () As Long	
此函数返回剩余内存字节数。	
Arguments	
Return values	remaining memory in Byte

8.7 视频输入通道 (EVHDTVIC)

EVHDTVICSetImageMemory	
Declare Function EVHDTVICSetImageMemory (ByVal IVICIndex As Long, ByVal IMemoryIndex As Long) As Long	
设置 VIC 到图像内存。	
Arguments	
IVICIndex	index of VIC
IMemoryIndex	index of image memory for the VIC
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

EVHDTVICGetImageMemory	
Declare Function EVHDTVICGetImageMemory (ByVal IVICIndex As Long, ByVal IMemoryIndex As Long) As Long	
从指定 VIC 获取图像内存。	
Arguments	
IVICIndex	index of VIC
IMemoryIndex	index of image memory for the VIC
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

EVHDTVICaptureImage		
Declare Function EVHDTVICaptureImage (ByVal IVICIndex As Long) As Long		
从指定 VIC 获取图像。		
Arguments		
IVICIndex	index of VIC	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	illegal configuration
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_IM_EXCEEDED	image memory overflow (GSI, RLE or OSI)

EVHDTVICStartCaptureImage		
Declare Function EVHDTVICStartCaptureImage (ByVal IVICIndex As Long) As Long		
开始异步图像采集进程。		
Arguments		
IVICIndex	index of VIC	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_SYSTEM	error during calling the hardware
	EVHD_ERR_IM_EXCEEDED	image memory overflow (GSI, RLE or OSI)

EVHDTVICGetCaptureImageState		
Declare Function EVHDTVICGetCaptureImageState (ByVal IVICIndex As Long, ByRef lptrState As Long) As Long		
检测异步图像获取进程是否结束。		
Arguments		
IVICIndex	index of VIC	
lptrState	state of the image capturing process:	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_IM_EXCEEDED	image memory overflow (GSI, RLE or OSI)

EVHDVICStartCaptureImageWithASFlash	
Declare Function EVHDVICStartCaptureImageWithASFlash (ByVal IVICIndex As Long) As Long	
用自动闪光触发异步图像获取。	
Arguments	
IVICIndex	index of VIC
lptrState	state of the image capturing process:
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM illegal index
	EVHD_ERR_SYSTEM error during calling the hardware
	EVHD_ERR_IM_EXCEEDED image memory overflow (GSI, RLE or OSI)

EVHDVICStopCaptureImageWithASFlash		
Declare Function EVHDVICStopCaptureImageWithASFlash (ByVal IVICIndex As Long) As Long		
立即停止已经开始的异步图像获取。		
Arguments		
IVICIndex	index of VIC	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_IM_EXCEEDED	image memory overflow (GSI, RLE or OSI)

EVHDVICGetCaptureImageWithASFlashState		
Declare Function EVHDVICGetCaptureImageWithASFlashState (ByVal IVICIndex As Long, ByRef lptrState As Long, ByRef lptrIMIndex As Long, ByRef lptrTriggers As Long) As Long		
函数的描述。		
Arguments		
IVICIndex	index of VIC	
lptrState	state of the image capturing process	
lptrIMIndex	index of image memory that holds last valid image	
lptrTriggers	number of triggers since start of snap trigger	

Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_IM_EXCEEDED	image memory overflow (GSI, RLE or OSI)

EVHDVICStopCaptureImage

Declare Function EVHDVICStopCaptureImage (ByVal IVICIndex As Long) As Long

停止异步图像获取进程。

Arguments

IVICIndex	index of VIC
-----------	--------------

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_IM_EXCEEDED	image memory overflow (GSI, RLE or OSI)

Additional Information

xamples

图形叠加 (EVHDKOI)

EVHDKOISetColour

Declare Function EVHDKOISetColour (ByVal IIMIndex As Long, ByVal IColourIndex As Long, ByVal IRedValue As Long, ByVal IGreenValue As Long, ByVal IBlueValue As Long) As Long

此函数设置图像内存的 GOI 值。

Arguments

IVICIndex	index of VIC
IColourIndex	index of color
IRedValue	red value of the entry (0 .. 255)
IGreenValue	green value of the entry (0 .. 255)
IBlueValue	blue value of the entry (0 .. 255)

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHDKOIClearGOI

Declare Function EVHDKOIClearGOI (ByVal IIMIndex As Long) As Long

此函数清除 GOI 的所有内容。

Arguments

IVICIndex	index of VIC
-----------	--------------

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

Additional Information

EVHDKOIDrawLine

Declare Function EVHDKOIDrawLine (ByVal IIMIndex As Long, ByVal ICoordMode As Long, ByVal IColourIndex As Long, ByVal IX1 As Long, ByVal IY1 As Long, ByVal IX2 As Long, ByVal IY2 As Long) As Long

此函数在 GOI 中画一条线。

Arguments

IVICIndex	index of VIC
-----------	--------------

ICoordMode	EVHD_COORD_IMAGE EVHD_COORD_DISPLAY_TERMINAL EVHD_COORD_DISPLAY_TERMINAL	or
IColourIndex	color	
IX1	X co-ordinate of the first point of the line	
IY1	Y co-ordinate of the first point of the line	
IX2	X co-ordinate of the second point of the line	
IY2	Y co-ordinate of the second point of the line	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_COORD	co-ordinates are out of range
	EVHD_ERR_IM	no access to the image memory, no image

EVHDGOISetPixel

Declare Function EVHDGOISetPixel (ByVal IIMIndex As Long, ByVal ICoordMode As Long, ByVal IColourIndex As Long, ByVal IX As Long, ByVal IY As Long) As Long

此函数为指定图像的图形叠加层的指定坐标设定像素值。

Arguments

IVICIndex	index of VIC	
ICoordMode	EVHD_COORD_IMAGE	
IColourIndex	color	
IX	X co-ordinate	
IY	Y co-ordinate	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_COORD	co-ordinates out of range
	EVHD_ERR_IM	no access to image memory, no image in memory

EVHDGOISetPixel		
Declare Function EVHDGOIDrawEllipse (ByVal IIMIndex As Long, ByVal ICoordMode As Long, ByVal IColourIndex As Long, ByVal IX As Long, ByVal IY As Long, ByVal IRadiusX As Long, ByVal IRadiusY As Long) As Long		
此函数在 GOI 中画一个椭圆形。		
Arguments		
ICoordMode	EVHD_COORD_IMAGE,	
IColourIndex	color	
IX	X co-ordinate of the center point	
IY	Y co-ordinate of the center point	
IRadiusX	radius of the ellipse horizontally	
IRadiusY	radius of the ellipse vertically	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_COORD	co-ordinates out of range
	EVHD_ERR_IM	no access to image memory, no image in memory

EVHDGOIDrawArc	
Declare Function EVHDGOIDrawArc (ByVal IIMIndex As Long, ByVal ICoordMode As Long, ByVal IColourIndex As Long, ByVal IX As Long, ByVal IY As Long, ByVal IRadiusX As Long, ByVal IRadiusY As Long, ByVal dFromAngle As Double, ByVal dLengthAngle As Double) As Long	
此函数在 GOI 中画一个弧线。	
Arguments	
ICoordMode	EVHD_COORD_IMAGE,
IColourIndex	color
IX	X co-ordinate of the center point
IY	Y co-ordinate of the center point
IRadiusX	radius of the ellipse horizontally
IRadiusY	radius of the ellipse vertically
dFromAngle	starting angle (0..360)
dLengthAngle	length of arc in degrees (0..360)
Return value	

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index
	EVHD_ERR_COORD	co-ordinates out of range
	EVHD_ERR_IM	no access to image memory, no image in memory

EVHDGOISetFontSize

Declare Function EVHDGOISetFontSize (ByVal IIMIndex As Long, ByVal IFontSize As Long) As Long

此函数在图像的图形叠加层，设置字符尺寸。

Arguments

IIMIndex	index of IM	
IFontSize	size of characters in pixel (0 .. EVHD_MAX_FONT_SIZE)	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or font size

EVHDGOIWriteString

Declare Function xxx (ByVal IIMIndex As Long, ByVal ICoordMode As Long, ByVal IColourIndex As Long, ByVal IX As Long, ByVal IY As Long, ByVal sString As String) As Long

此函数在 GOI 中写字符串。

Arguments

IIMIndex	index of IM	
ICoordMode	EVHD_COORD_IMAGE	
IColourIndex	color	
IX	X co-ordinate of the starting point: lower, ultimate left point of the first character	
IY	Y co-ordinate of the starting point: lower, ultimate left point of the first character	
sString	pointer to the string to be written to the GOI	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

	EVHD_ERR_COORD	co-ordinates are out of range
	EVHD_ERR_IM	no access to image memory, no image in

图像管理 (EVHDIM)

EVHDIMGetNumberOfIM

Declare Function EVHDIMGetNumberOfIM (ByRef IptrNumber As Long) As Long

获取图像内存的数值。

Arguments

IpPtrNumber	number of available video input channels
Return value	
	EVHD_SUCCESS

EVHDIMSetGSIPixel

Declare Function EVHDIMSetGSIPixel (ByVal IIMIndex As Long, ByVal IX As Long, ByVal IY As Long, ByVal ulValue As Long) As Long

此函数为指定图像内存的指定坐标设置像素值。

Arguments

IIMIndex	index of image memory	
IX	X co-ordinate	
IY	Y co-ordinate	
ulValue	value to set	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or value
	EVHD_ERR_COORD	co-ordinates are out of range
	EVHD_ERR_IM	no access to the image memory, no image

EVHDIMGetGSIPixel

Declare Function EVHDIMGetGSIPixel (ByVal IIMIndex As Long, ByVal IX As Long, ByVal IY As Long, ByRef ulPtrValue As Long) As Long

此函数获取指定图像内存中指定坐标的像素值。

Arguments

IIMIndex	index of image memory	
IX	X co-ordinate	
IY	Y co-ordinate	
ulValue	value to set	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or value
	EVHD_ERR_COORD	co-ordinates are out of range
	EVHD_ERR_IM	no access to the image memory, no image in the memory

EVHDIMSetLastVIC

Declare Function EVHDIMSetLastVIC (ByVal IIMIndex As Long, ByVal ILastVIC As Long) As Long

此函数为最后一个 VIC 设置序号。

Arguments

IIMIndex	index of image memory	
ILastVIC	index of last VIC that entered an image	
ulValue	value to set	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_PARM	illegal index or value

EVHDIMGetLastVIC

Declare Function EVHDIMGetLastVIC (ByVal IIMIndex As Long, ByRef lptrVIC As Long) As Long

此函数获取最后一个 VIC 的序号。

Arguments

IIMIndex	index of image memory	
lptrVIC	index of last VIC that entered an image	
ulValue	value to set	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_PARM	illegal index or value

	EVHD_ERROR	no image is in the image memory
--	------------	---------------------------------

EVHDIMGetGSIVPitch

Declare Function EVHDIMGetGSIVPitch (ByVal IIMIndex As Long, ByRef IptrOffset As Long) As Long

获取灰度垂直图像间距。

Arguments

IIMIndex	index of image memory
IptraOffset	The address of a pixel is offset to its upper neighbor

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or value
	EVHD_ERROR_IM	no access to the image memory, no image

EVHDIMGetPixelsPerLine

Declare Function EVHDIMGetPixelsPerLine (ByVal IIMIndex As Long, ByRef IptraPixels As Long) As Long

此函数获取图像内存中图像每行像素值。

Arguments

IIMIndex	index of image memory
IptraPixels	number of pixels per line

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or value
	EVHD_ERROR_IM	no access to the image memory, no image is in the image memory

EVHDIMGetBytesPerPixel

Declare Function EVHDIMGetBytesPerPixel (ByVal IIMIndex As Long , ByRef IptraBytes As Long) As Long

此函数获取指定图像内存中实际图像的每个像素的字节数。

Arguments

IIMIndex	index of image memory
IptraBytes	number of bytes per pixel

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or value
	EVHD_ERROR_IM	no access to the image memory, no image is in the image memory

EVHDIMGetLinesPerFrame

Declare Function EVHDIMGetLinesPerFrame (ByVal IIMIndex As Long, ByRef lptrLines As Long) As Long

此函数获取每帧的线数。

Arguments

IIMIndex	index of image memory
lptrLines	number of lines per frame

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or value
	EVHD_ERROR_IM	no access to the image memory, no image is in the image memory

EVHDIMGetGSIPixelAddress

Declare Function EVHDIMGetGSIPixelAddress (ByVal IIMIndex As Long, ByVal IX As Long, ByVal IY As Long,

此函数获取 GSI 图像内存的像素地址。

Arguments

IIMIndex	index of image memory
IX	X co-ordinate
IY	Y co-ordinate
vpPtrPixelPtr	pointer to the given pixel

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or value
	EVHD_ERROR_IM	illegal memory index or file name

EVHDIMSaveToFile		
Declare Function EVHDIMSaveToFile (ByVal IIndex As Long, ByVal sFileName As String) As Long		
此函数允许保存图像内容到指定文件。		
Arguments		
IIndex	index of image memory	
sFileName	Pointer to path and file name	
Return value	EVHD_SUCCESS	illegal memory index or file name
	EVHD_ERR_PARM	illegal memory index or file name

EVHDIMFillImage		
Declare Function EVHDIMFillImage (ByVal IIndex As Long, ByVal IBPP As Long, ByVal IPPL As Long, ByVal ILPF As Long) As Long		
此函数用指定大小的新位图填充图像。		
Arguments		
IIndex	index	
IBPP	bytes per pixel	
IPPL	pixels per line	
ILPF	lines per frame	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_PARM	illegal memory index or bytes per pixel

EVHDIMGetRawPixel	
Declare Function EVHDIMGetRawPixel (ByVal IIMIndex As Long, ByVal IX As Long, ByVal IY As Long, ByRef dptrValue As Double, ByVal IChannel As Long) As Long	
此函数获取原始像素。	
Arguments	
IIMIndex	index of image memory
IX	X co-ordinate
IY	Y co-ordinate
dptrValue	value
IChannel	channel
Return value	
	EVHD_SUCCESS

	EVHD_ERROR
	EVHD_FALSE
	EVHD_ERR_COORD
	EVHD_ERR_PARM
	EVHD_ERR_IM

EVHDIMSetRawPixel

Declare Function EVHDIMSetRawPixel (ByVal IIMIndex As Long, ByVal IX As Long, ByVal IY As Long, ByVal dValue As Double, ByVal IChannel As Long) As Long

此函数设置原始像素。

Arguments

IIMIndex	index of image memory
IX	X co-ordinate
IY	Y co-ordinate
dValue	Entered value
IChannel	channel
Return value	EVHD_SUCCESS
	EVHD_ERROR
	EVHD_ERR_COORD
	EVHD_ERR_IM
	EVHD_ERR_PARM

Image Display Channel (EVHIDC)**EVHIDCGetNumberOfIDC**

Declare Function EVHIDCGetNumberOfIDC (ByRef IptrNumber As Long) As Long

获取 IDC 数值。

Arguments

IpnrNumber	number of available image display channels
Return value	
	EVHD_SUCCESS

EVHIDCSetDisplayMode

Declare Function EVHIDCSetDisplayMode (ByVal IIDCIndex As Long, ByVal IMode As Long) As Long

设置/重置 IDC 显示模式。

Arguments

IIDCIndex	index of IDC	
IMode	1: set mode (display active)	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	
	EVHD_ERROR	illegal index or mode
	EVHD_ERR_CONFIG	error during activating/deactivating mode

EVHDIDCSetFitMode

Declare Function EVHDIDCSetFitMode (ByVal IIDCIndex As Long, ByVal IMode As Long) As Long

此函数重置 IDC 的适应模式。

Arguments

IIDCIndex	index of IDC
IMode	1: set mode (display active) 0: reset mode
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

EVHDIDCSetDisplayWindow

Declare Function EVHDIDCSetDisplayWindow (ByVal IIDCIndex As Long, ByVal IX As Long, ByVal IY As Long, ByVal IWidth As Long, ByVal IHeight As Long) As Long

此函数设置显示终端的窗口。

Arguments

IIDCIndex	index of IDC
IX	X position of upper, left corner
IY	Y position of upper, left corner
IWidth	width of the window on the display terminal
IHeight	height of the window on the display terminal
Return value	

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or image memory
	EVHD_ERR_COORD	

EVHDIDCSetImageWindow

Declare Function EVHDIDCSetImageWindow (ByVal IIDCIndex As Long, ByVal IX As Long, ByVal IY As Long, ByVal IWidth As Long, ByVal IHeight As Long) As Long

此函数设置图像内存窗口。

Arguments

IIDCIndex	index of IDC
IX	X position of upper, left corner
IY	Y position of upper, left corner
IWidth	width of the image window
IHeight	height of the image window

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or image memory
	EVHD_ERR_COORD	

EVHDIDCRefreshDisplayWindow

Declare Function EVHDIDCRefreshDisplayWindow (ByVal IIDCIndex As Long) As Long

在显示窗口内将 IDC 显示为实际的内容。

Arguments

IIDCIndex	index of IDC
-----------	--------------

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHDIDCSetImageMemory

Declare Function EVHDIDCSetImageMemory (ByVal IIDCIndex As Long, ByVal IMemoryIndex As Long, ByVal IMemoryType As Long) As Long

设置图像内存的序号，设置图像类型 (GSI, RLE or OSI)。

Arguments

IIDCIndex	index of IDC	
IMemoryIndex	index of image memory for the IDC	
IMemoryType	EVHD_IMAGE_TYPE_GSI	or

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or image memory

EVHDIDCGetImageMemory

Declare Function EVHDIDCGetImageMemory (ByVal IIDCIndex As Long, ByRef IptrMemoryIndex As Long, ByRef IptrMemoryType As Long) As Long

获取图像内存的序号和 IDC 的图像类型。

Arguments

IIDCIndex	index of IDC	
IptrMemoryIndex	index of image memory for the IDC	
IptrMemoryType	EVHD_IMAGE_TYPE_GSI EVHD_IMAGE_TYPE_RLE EVHD_IMAGE_TYPE_OSI	or or

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or image memory

Additional Information

EVHDIDCSaveToFile

Declare Function EVHDIDCSaveToFile (ByVal IIDCIndex As Long, ByVal sFile As String) As Long

以 RGB 形式保存 IDC 的内容，包括图像和指定图像内存的图形。

Arguments

IIDCIndex	index of IDC
sFile	file to save content of IDC

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or image memory
	EVHD_ERROR	error during activating/deactivating mode
Additional Information		
Examples		

数字输入输出 (EVHDDIO)

EVHDDIOGetNumberOfInputLines	
Declare Function EVHDDIOGetNumberOfInputLines (ByRef lpPtrNumber As Long) As Long	
获取输入线的数量。	
Arguments	
lpPtrNumber	number of available input lines
Return value	
	EVHD_SUCCESS
Additional Information	
Examples	

EVHDDIOGetNumberOfOutputLines	
Declare Function EVHDDIOGetNumberOfOutputLines (ByRef lptrNumber As Long) As Long	
获取输出线的数量。	
Arguments	
lptrNumber	number of available output lines
Return value	
	EVHD_SUCCESS

EVHDDIOSetOutputLine		
Declare Function EVHDDIOSetOutputLine (ByVal lOutputIndex As Long, ByVal lState As Long) As Long		
设置/重置输出线。		
Arguments		
lOutputIndex	index of the output line	

IState	1: on 0: off	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or state

EVHDDIOGetOutputLine

Declare Function EVHDDIOGetOutputLine (ByVal IOutputIndex As Long, ByRef IptrState As Long) As Long

获取输出线的状态。

Arguments

IOutputIndex	index of the output line
IState	1: on 0: off

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHDDIOGetInputLine

Declare Function EVHDDIOGetInputLine (ByVal IInputIndex As Long, ByRef IptrState As Long) As Long

获取输入线的状态。

Arguments

IInputIndex	index of the input line
IptrState	1: on 0: off

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

Additional Information**Examples****EVHDDIOSetOutputLineLowActiveMode**

Declare Function EVHDDIOSetOutputLineLowActiveMode (ByVal IOutputIndex As Long, ByVal IState As Long) As Long		
设置/重置输出线到低有效模式。		
Arguments		
IOutputIndex	index of the output line	
IState	1: on 0: off	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or state

EVHDDIOGetOutputLineLowActiveMode		
Declare Function EVHDDIOGetOutputLineLowActiveMode (ByVal IOutputIndex As Long, ByRef IptrState As Long) As Long		
获取输出状态到低有效模式。		
Arguments		
IOutputIndex	index of the output line	
IState	1: on 0: off	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index or state

EVHDDIOSetInputLineLowActiveMode	
Declare Function EVHDDIOSetInputLineLowActiveMode (ByVal IInputIndex As Long, ByVal IState As Long) As Long	
设置/重置输入线。	
Arguments	
IInputIndex	index of the input line
IState	1: on 0: off
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG

	EVHD_ERR_PARM	illegal index or state
--	---------------	------------------------

EVHDDIOGetInputLineLowActiveMode

Declare Function EVHDDIOGetInputLineLowActiveMode (ByVal lInputIndex As Long, ByRef lptrState As Long)

获取输入线的低有效状态。

Arguments

lInputIndex	index of the input line
lState	1: on 0: off

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHDTERM**EVHDTERMResetModule**

Declare Function EVHDTERMResetModule () As Long

重置模式。

Arguments**Return value**

	EVHD_SUCCESS
--	--------------

Additional Information**Examples****EVHDTERMWriteLine**

Declare Function EVHDTERMWriteLine (ByVal sLine As String) As Long

在调试终端写一行。

Arguments

sLine	line is written to the debug terminal
-------	---------------------------------------

Return value

	EVHD_SUCCESS
--	--------------

RS232模块 (EVHD)

EVHRS232InitModule

Declare Function EVHRS232InitModule (ByVal INumberOfIndicees As Long) As Long

此函数初始化 EVHD RS232并指明使用的连接数。每个连接都必须设置 EVHRS232Set 参数，而且在使用前要打开。

Arguments

INumberOfIndicees	Number of possible connections
-------------------	--------------------------------

EVHRS232IsOpen

Declare Function EVHRS232IsOpen (ByVal IRS232Index As Long, ByRef lIsOpen As Long) As Long

如果 RS232连接打开，此函数返回信息。

Arguments

IRS232Index	index of RS232Connection
lIsOpen	1 if open 0 if not open

Return value

EVHD_SUCCESS
EVHD_ERR_CONFIG

EVHRS232Open

Declare Function EVHRS232Open (ByVal IRS232Index As Long) As Long

此函数打开 RS232 连接。

Arguments

IRS232Index	index of RS232
-------------	----------------

Return value

EVHD_SUCCESS	
EVHD_ERR_CONFIG	
EVHD_ERR_PARM	illegal index

EVHRS232Close		
Declare Function EVHRS232Close (ByVal IRS232Index As Long) As Long		
此函数关闭 RS232 连接。		
Arguments		
IRS232Index	index of RS232	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHDRS232Receive		
Declare Function EVHDRS232Receive (ByVal IRS232Index As Long, ByRef ucpBuffer As String, ByVal INumberOfBytesToRead As Long) As Long		
此函数通过 RS232 连接接收字符串。		
Arguments		
IRS232Index	index of RS232	
ucpBuffer	String variable that contains the received data. Please initialize the variable before the first use.	
INumberOfBytesToRead	Number of Bytes that will be read.	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHRS232Send	
Declare Function EVHRS232Send (ByVal IRS232Index As Long, ByVal ucpBuffer As String, ByVal INumberOfBytesToRead As Long) As Long	
此函数通过 RS232 连接发送字符串。	
Arguments	
IRS232Index	index of RS232
ucpBuffer	String variable that contains the data to send.
INumberOfBytesToRead	Number of Bytes that will be read.
Return value	
	EVHD_SUCCESS

	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHRS232GetNumberOfPorts

Declare Function EVHRS232GetNumberOfPorts (ByRef IpNumberOfPorts As Long) As Long

此函数返回可用端口数量。

Arguments

IRS232Index	index of RS232
INumberOfBytes	Number of available Bytes.
INumberOfBytesToRead	Number of Bytes that will be read.

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHRS232GetNumberOfBytes

Declare Function EVHRS232GetNumberOfBytes (ByVal IRS232Index, ByRef IpNumberOfBytesToRead As Long) As Long

此函数返回选择的 RS232连接的可读字节数。

Arguments

IRS232Index	index of RS232
INumberOfBytes	Number of available Bytes.
INumberOfBytesToRead	Number of Bytes that will be read.

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHRS232SetIOTimeout		
Declare Function EVHRS232SetIOTimeout (ByVal IRS232Index, ByRef ITimeoutAsMilliseconds As Long) As Long		
此函数设置超时毫秒数。		
Arguments		
IRS232Index	index of RS232	
ITimeoutAsMilliseconds	Timeout in [ms]	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHRS232GetNumberOfBytes		
Declare Function EVHRS232GetNumberOfBytes (ByRef IpRS232Index as Long) As Long		
此函数返回上次使用的 RS232序号。		
Arguments		
IRS232Index	Index of the last used RS232	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHRS232Setparameter		
Declare Function EVHRS232Setparameter (ByVal IRS232Index As Long, ByVal IPortIndex As Long, ByVal IBaudRate As Long, ByVal IDataBits As Long, ByVal IParity As Long, ByVal IStopBits As Long, ByVal IFlowControl As Long) As Long		
此函数设置在 IRS232Index 中指明的 RS232连接的参数。参数的结果要根据硬件，例如 EyeVision RS232 命令使用的参数：EVHRS232Setparameter (IRS232Index, IPortnumber , 9600, 8, 3, 1, 3);		
Arguments		
IRS232Index	index of RS232	
IPortIndex	Index of the port	

IBaudRate	possible values are: CK_SERIAL_BAUDRATE_INVALID, CK_SERIAL_BAUDRATE_110 = 110, CK_SERIAL_BAUDRATE_300 = 300, CK_SERIAL_BAUDRATE_600 = 600, CK_SERIAL_BAUDRATE_1200 = 1200, CK_SERIAL_BAUDRATE_2400 = 2400, CK_SERIAL_BAUDRATE_4800 = 4800, CK_SERIAL_BAUDRATE_9600 = 9600, CK_SERIAL_BAUDRATE_14400 = 14400, CK_SERIAL_BAUDRATE_19200 = 19200, CK_SERIAL_BAUDRATE_38400 = 38400, CK_SERIAL_BAUDRATE_56000 = 56000, CK_SERIAL_BAUDRATE_57600 = 57600, CK_SERIAL_BAUDRATE_115200 = 115200, CK_SERIAL_BAUDRATE_128000 = 128000, CK_SERIAL_BAUDRATE_256000 = 256000	
IDataBits	CK_SERIAL_DATABITS_INVALID = 0, CK_SERIAL_DATABITS_4 = 4, CK_SERIAL_DATABITS_5 = 5, CK_SERIAL_DATABITS_6 = 6, CK_SERIAL_DATABITS_7 = 7, CK_SERIAL_DATABITS_8 = 8	
IParity	CK_SERIAL_PARITY_TYPE_INVALID = 0, CK_SERIAL_PARITY_TYPE_EVEN = 1, CK_SERIAL_PARITY_TYPE_MARK = 2, CK_SERIAL_PARITY_TYPE_NO_PARITY = 3, CK_SERIAL_PARITY_TYPE_ODD = 4, CK_SERIAL_PARITY_TYPE_SPACE = 5	
IStopBits	CK_SERIAL_STOPBITS_INVALID = 0, CK_SERIAL_STOPBITS_ONE = 1, CK_SERIAL_STOPBITS_ONE_DOT_FIVE = 2, CK_SERIAL_STOPBITS_TWO = 3	
IFlowControl	CK_SERIAL_FLOWCONTROL_INVALID = 0, CK_SERIAL_FLOWCONTROL_XON_XOFF = 1, CK_SERIAL_FLOWCONTROL_HARDWARE = 2, CK_SERIAL_FLOWCONTROL_NONE = 3	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

EVHDRS232Getparameter

Declare Function EVHDRS232Getparameter (ByVal IRS232Index As Long, ByRef IpPortIndex As Long, ByRef IpBaudRate As Long, ByRef IpDataBits As Long, ByRef IpParity As Long, ByRef IpStopBits As Long, ByRef IpFlowControl As Long) As Long

此函数设置由 IRS232Index 指定的 RS232 连接的参数。

Arguments

IRS232Index	index of RS232
IPortIndex	Index of the port
IpBaudRate	CK_SERIAL_BAUDRATE_INVALID, CK_SERIAL_BAUDRATE_110 = 110, CK_SERIAL_BAUDRATE_300 = 300, CK_SERIAL_BAUDRATE_600 = 600, CK_SERIAL_BAUDRATE_1200 = 1200, CK_SERIAL_BAUDRATE_2400 = 2400, CK_SERIAL_BAUDRATE_4800 = 4800, CK_SERIAL_BAUDRATE_9600 = 9600, CK_SERIAL_BAUDRATE_14400 = 14400, CK_SERIAL_BAUDRATE_19200 = 19200, CK_SERIAL_BAUDRATE_38400 = 38400, CK_SERIAL_BAUDRATE_56000 = 56000, CK_SERIAL_BAUDRATE_57600 = 57600, CK_SERIAL_BAUDRATE_115200 = 115200, CK_SERIAL_BAUDRATE_128000 = 128000, CK_SERIAL_BAUDRATE_256000 = 256000
IpDataBits	CK_SERIAL_DATABITS_INVALID = 0, CK_SERIAL_DATABITS_4 = 4, CK_SERIAL_DATABITS_5 = 5, CK_SERIAL_DATABITS_6 = 6, CK_SERIAL_DATABITS_7 = 7, CK_SERIAL_DATABITS_8 = 8
IpParity	CK_SERIAL_PARITY_TYPE_INVALID = 0, CK_SERIAL_PARITY_TYPE_EVEN = 1, CK_SERIAL_PARITY_TYPE_MARK = 2, CK_SERIAL_PARITY_TYPE_NO_PARITY = 3, CK_SERIAL_PARITY_TYPE_ODD = 4, CK_SERIAL_PARITY_TYPE_SPACE = 5
IpStopBits	CK_SERIAL_STOPBITS_INVALID = 0, CK_SERIAL_STOPBITS_ONE = 1, CK_SERIAL_STOPBITS_ONE_DOT_FIVE = 2, CK_SERIAL_STOPBITS_TWO = 3

IpFlowControl	CK_SERIAL_FLOWCONTROL_INVALID = 0, CK_SERIAL_FLOWCONTROL_XON_XOFF = 1, CK_SERIAL_FLOWCONTROL_HARDWARE = 2, CK_SERIAL_FLOWCONTROL_NONE = 3	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal index

用户 (EVHDUSER)

EVHDUSERDoEvents		
Declare Function EVHDUSERDoEvents () As Long		
DoEvents		
Arguments		
Return value		
	EVHD_SUCCESS	

8.8 视觉功能

CKSetValueRegister	
Declare Function CKSetValueRegister (ByVal dValX As Double, ByVal sName As String) As Long	
此函数将值放入值寄存器的最上面。	
Arguments	
dValX	Value
sName	Name/comment of the object
Return value	
	EVHD_SUCCESS

CKGetValueRegister		
Declare Function CKGetValueRegister (ByVal lIndex As Long, ByRef dValue As Double, ByVal sName As String) As Long		
此函数返回值寄存器最上面的值。		
Arguments		
lIndex	index relative to the top of the register	
dValue	Value	
sName	name / comment of the object (string with up to 16 characters)	
Return value		
	EVHD_SUCCESS	
	EVHD_ERROR	illegal index

CKSetPointRegister	
Declare Function CKSetPointRegister (ByVal dValX As Double, ByVal dValY As Double, ByVal sName As String) As Long	
此函数将指定的点放到点寄存器的最上面和点列表中。	
Arguments	
dValX	x co-ordinate of the point
dValY	y co-ordinate of the point
sName	name / comment of the object
Return value	
	EVHD_SUCCESS

CKGetPointRegister	
Declare Function CKGetPointRegister (ByVal lIndex As Long, ByRef dValX As Double, ByRef dValY As Double, ByVal sName As String) As Long	
此函数返回点寄存器最上面的点。	
Arguments	
lIndex	index relative to the top of the register
dValX	x co-ordinate of the indexed point
dValY	y co-ordinate of the indexed point
sName	name / comment of the object (string with up to 16 characters)
Return value	
	EVHD_SUCCESS

CKSetLineRegister	
Declare Function CKSetLineRegister (ByVal dValX As Double, ByVal dValY As Double, ByVal dx1 As Double, ByVal dy1 As Double, ByVal sName As String) As Long	
此函数将指定的线放到线寄存器最上面。	
Arguments	
dValX	x co-ordinate of the point on the line
dValY	y co-ordinate of the point on the line
dx1	Direction of the vector of the line
dy1	Direction of the vector of the line
sName	name / comment of the object
Return value	
	EVHD_SUCCESS

CKGetLineRegister	
Declare Function CKGetLineRegister (ByVal lIndex As Long, ByRef dValX As Double, ByRef dValY As Double, ByRef dx1 As Double, ByRef dy1 As Double, ByVal sName As String) As Long	
此函数返回线寄存器最上面的线。	
Arguments	
lIndex	Index relative to the top of register
dValX	x co-ordinate of point on indexed line
dValY	y co-ordinate of point on indexed line
dx1	x co-ordinate of the vector of the indexed line
dy1	y co-ordinate of the vector of the indexed line
sName	name / comment of the object (string with up to 16 characters)
Return value	
	EVHD_SUCCESS

CKSetCircleRegister	
Declare Function CKSetCircleRegister (ByVal dValX As Double, ByVal dValY As Double, ByVal dR As Double, ByVal sName As String) As Long	
此函数将指定的圆放入圆寄存器最上面。	
Arguments	
lIndex	Index
dValX	x co-ordinate of the center point
dValY	y co-ordinate of the center point
dR	Radius of the circle
sName	name / comment of the object
Return value	
	EVHD_SUCCESS

CKGetCircleRegister	
Declare Function CKGetCircleRegister (ByVal lIndex As Long, ByRef dValX As Double, ByRef dValY As Double, ByRef dX1 As Double, ByRef dY1 As Double, ByVal sName As String) As Long	
此函数返回圆寄存器最上面的圆。	
Arguments	
lIndex	Index
dValX	x co-ordinate of the center point
dValY	y co-ordinate of the center point
dR	Radius of the circle
sName	name / comment of the object
Return value	
	EVHD_SUCCESS

点列表

CKResetPointList	
Declare Function CKResetPointList () As Long	
此函数清除点列表的内容。	
Arguments	
none	
Return value	
	EVHD_SUCCESS

CKGetLengthOfPointList		
Declare Function CKGetLengthOfPointList (ByVal lIndex As Long, ByRef dValX As Double, ByRef dValY As Double) As Long		
此函数检索点列表的点。		
Arguments		
lIndex	Index	
lLength	Number of entries	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal parameter value or type

CKEnterPointToPointList		
Declare Function CKEnterPointToPointList (ByVal dValX As Double, ByVal dValY As Double) As Long		
此函数在点列表中增加一个点。		
Arguments		
dValX	X co-ordinate of the point in the point list	
dValY	Y co-ordinate of the point in the point list	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_PARM	illegal parameter value or type

CKGetPointOfPointList		
Declare Function CKGetPointOfPointList (ByVal lIndex As Long, ByRef dValX As Double, ByRef dValY As Double) As Long		
此函数检索点列表上的一个点。		
Arguments		
lIndex	index of the point (0: oldest one)	
dValX	x co-ordinate of the point	
dValY	y co-ordinate of the point	
Return value		
	EVHD_SUCCESS	
	EVHD_ERROR	invalid index (no more points)

选取列表

CKTRANSGetPointFromPickupList		
Declare Function CKTRANSGetPointFromPickupList (ByVal xxx As Long) As Long		
此函数返回选取列表上的一个点。		
Arguments		
lIndex	Unique index of the element of the pickup list	
dX	x co-ordinate of the point	
dY	y co-ordinate of the point	
sName	String that identifies the element of the pickup list	
lUpdateMode	Please see explanation of update mode	

Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal parameter value or type

CKTRANSGetLineFromPickupList		
Declare Function CKTRANSGetLineFromPickupList (ByVal xxx As Long) As Long		
此函数返回选择列表上的一条直线。		
Arguments		
lIndex	Unique index of the element of the pickup list	
dX1	x co-ordinate of the point	
dY1	y co-ordinate of the point	
dX2	x co-ordinate of the point	
dY2	y co-ordinate of the point	
sName	String that identifies the element of the pickup list	
lUpdateMode	Please see explanation of update mode	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal parameter value or type

CKTRANSGetCircleFromPickupList	
Declare Function CKTRANSGetCircleFromPickupList (ByVal xxx As Long) As Long	
此函数返回选取列表上的一个圆。	
Arguments	
lIndex	Unique index of the element of the pickup list
dCenterX	x co-ordinate of the center point
dCenterY	y co-ordinate of the center point
dRadius	Radius
sName	String that identifies the element of the pickup list
lUpdateMode	Please see explanation of update mode
Return value	
	EVHD_SUCCESS

	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal parameter value or type

CKTRANSGetWindowFromPickupList

Declare Function CKTRANSGetWindowFromPickupList (ByVal xxx As Long) As Long

此函数返回选取列表上的一个窗口。

Arguments

lIndex	Unique index of the element of the pickup list
dTop	top of the window
dLeft	left corner of the window
dWidth	Width of the window
dHeight	Height of the window
sName	String that identifies the element of the pickup list
lUpdateMode	Please see explanation of update mode

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	illegal parameter value or type

KTRANSGetResultFromPickupList

Declare Function CKTRANSGetResultFromPickupList (ByVal xxx As Long) As Long

此函数返回选择取列中的一个值或命令结果。

Arguments

lIndex	Unique index of the element of the pickup list
lResult	Long value that can represent a result value
dValue	Value
sName	String that identifies the element of the pickup list
lUpdateMode	Please see explanation of update mode

Return value

	EVHD_SUCCESS	
	EVHD_ERR_CONFIG	
	EVHD_ERR_PARM	Illegal parameter value or type

计算

CKGetEdgeCountOnCircleBinary

Declare Function CKGetEdgeCountOnCircleBinary (ByVal IIMIndex As Long, ByVal lxcenter As Long, ByVal lycenter As Long, ByVal IRadius As Long, ByVal IThreshold As Long, ByVal IFilterLength As Long) As Long

此函数使用边缘探测方法“binary”，在指定圆内探测所有边缘并返回边缘数量。

Arguments

IIMIndex	Index of the the image memory
lxcenter	center point of radius
lycenter	center point of radius
IRadius	Radius of the circle
IThreshold	threshold for edge detection
IFilterLength	minimum object size
	The function will affect the following registers:

Return value

EVHD_SUCCESS

CKGetIntensityAreaOnCircle

Declare Function CKGetIntensityAreaOnCircle (ByVal IIMIndex As Long, ByVal lx As Long, ByVal ly As Long, ByVal lr As Long, ByVal IMin As Long, ByVal IMax As Long) As Long

此函数检查一个圆上给定强度的像素数量。

Arguments

IIMIndex	memory index
lx	Midpoint of a given circle
ly	Midpoint of a given circle
lr	radius of a given circle
IMin	minimum intensity of pixels
IMax	maximum intensity of pixels
	The function will affect the following registers:

Return value

EVHD_SUCCESS

CKGetExtremePoints	
Declare Function CKGetExtremePoints (ByVal IIMIndex As Long, ByVal lx As Long, ByVal ly As Long, ByVal lw As Long, ByVal lh As Long, ByVal IThreshold As Long, ByVal IColour As Long, ByVal IAutoParm As Long, ByVal IMinExtension As Long) As Long	
此函数在指定矩形窗口内的四个方向上获取黑或白的极值点。这四个点将被放入点寄存器最上面。	
Arguments	
IIMIndex	memory index
lx	X position of the window
ly	Y position of the window
lw	Width of the window
lh	Height of the window
IThreshold	threshold parameter
IColour	PEG_DARK or PEG_BRIGHT
IAutoParm	automatical parameter optimization
IMinExtension	minimum extension of an object
	The function will affect the following registers: Point: [0]: top point [1]: right point [2]: bottom point [3]: left point
Return value	
	EVHD_SUCCESS

CKGetExtremePointsFast	
Declare Function CKGetExtremePointsFast (ByVal IIMIndex As Long, ByVal lx As Long, ByVal ly As Long, ByVal lw As Long, ByVal lh As Long, ByVal IThreshold As Long, ByVal IColour As Long, ByVal IAutoParm As Long, ByVal IMinExtension As Long) As Long	
此函数在指定矩形窗口内的四个方向上获取黑或白的极值点。这四个点将被放入点寄存器最上面。（快速版本）	
Arguments	
IIMIndex	memory index
lx	X position of the window
ly	Y position of the window
lw	Width of the window
lh	Height of the window

IThreshold	Threshold parameter
IColour	PEG_DARK or PEG_BRIGHT
IAutoParm	Automatical parameter optimization
IMinExtension	minimum extension of an object
	The function will affect the following registers: Point: [0]: top point [1]: right point [2]: bottom point [3]: left point
Return value	
	EVHD_SUCCESS

CKGetBoundingEdgesOnLine

Declare Function CKGetBoundingEdgesOnLine (ByVal IIMIndex As Long, ByVal lxstart As Long, ByVal lystart As Long, ByVal lxend As Long, ByVal lyend As Long, ByVal IThreshold As Long, ByVal IColour As Long, ByVal IAutoParm As Long, ByVal IMinExtension As Long) As Long

此函数通过在一条线的两端开始查找对象的两个边缘。

Arguments

IIMIndex	memory index
lxstart	X position of the starting point
lystart	Y position of the starting point
lxend	X position of the end point
lyend	Y position of the end point
IThreshold	Threshold parameter
IColour	PEG_DARK or PEG_BRIGHT
IAutoParm	Automatically parameter optimization
IMinExtension	minimum extension of an object
	The function will affect the following registers: Point: [0]: point detected, starting from 'start' [1]: point detected, starting from 'end'
Return value	
	EVHD_SUCCESS

CKGetContourPoints	
Declare Function CKGetContourPoints (ByVal IIMIndex As Long, ByVal lx As Long, ByVal ly As Long, ByVal lw As Long, ByVal lh As Long, ByVal IThreshold As Long, ByVal IAutoParm As Long, ByVal IMinExtension As Long, ByVal INumberOfPoints As Long) As Long	
此函数在指定矩形窗口内获取对象边缘的所有点。对象轮廓线必须与窗口相交。	
Arguments	
IIMIndex	memory index
lx	X position of the window
ly	Y position of the window
lw	Width of the window
lh	Height of the window
IThreshold	threshold for detection
IAutoParm	Automatical parameter optimization
IMinExtension	minimum extension of an object
INumberOfPoints	number of detected points
	The function will affect the following registers: Value: [0]:number of detected points
Return value	
	EVHD_SUCCESS

CKGetContourPointsWithStartingPoint	
Declare Function CKGetContourPointsWithStartingPoint (ByVal xxx As Long) As Long	
此函数在指定矩形窗口内获取对象边缘的所有点。边缘探测从指定点开始。	
IIMIndex	memory index
lx	X position of the window
ly	Y position of the window
lw	Width of the window
lh	Height of the window
IThreshold	threshold for detection
IAutoParm	Automatically parameter optimization
IMinExtension	minimum extension of an object to be detected
INumberOfPoints	number of detected points

Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

坐标系统函数

CKSetScaleFactors	
Declare Function CKSetScaleFactors (ByVal IVICIndex As Long, ByVal dptrX As Double, ByVal dptrY As Double) As Long	
此函数为一个指定图像内存设置比例因子。	
Arguments	
IVICIndex	index of video input channel
dptrX	horizontal scale factor
dptrY	vertical scale factor
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKGetScaleFactors	
Declare Function CKGetScaleFactors (ByVal IVICIndex As Long, ByRef dX As Double, ByRef dY As Double) As Long	
此函数获取一个指定图像内存的比例因子。	
Arguments	
IVICIndex	index of video input channel
dX	horizontal scale factor
dY	vertical scale factor
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKTransformToImage	
Declare Function CKTransformToImage (ByVal IVICIndex As Long, ByRef dX As Double, ByRef dY As Double) As Long	
此函数将一个点的坐标转换成图像坐标系统。	
Arguments	
IVICIndex	index of video input channel
dX	x co-ordinate to be transformed
dY	y co-ordinate to be transformed
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKTransformToWorld	
Declare Function CKTransformToWorld (ByVal IVICIndex As Long, ByRef dX As Double, ByRef dY As Double) As Long	
此函数将一个点的坐标转换成全局坐标系统。	
Arguments	
IVICIndex	index of video input channel
dX	x co-ordinate to be transformed
dY	y co-ordinate to be transformed
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGetAngleTwoLines	
Declare Function CKGetAngleTwoLines (ByVal sL1 As String, ByVal sL2 As String, ByVal sAngleName As String) As Long	
此函数计算线寄存器中最上面的两条直线间的角度。	
Arguments	
sL1	name of line 1
sL2	name of line 2
sAngleName	name of angle
	The function will affect the following registers: Value: [0]: angle

Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGetDistancePointToLine

Declare Function CKGetDistancePointToLine (ByVal sPointFrom As String, ByVal sLineTo As String, ByVal sDistName As String) As Long

此函数计算线寄存器最上面的直线到点寄存器最上面的点的距离。

Arguments

sPointFrom	
sLineTo	
sDistName	name of distance
	The function will affect the following registers: Value: [0]: distance

Return value

	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGetDistancePointToPoint

Declare Function CKGetDistancePointToPoint (ByVal sPointFrom As String, ByVal sPointTo As String, ByVal lDistType As Long, ByVal sDistName As String) As Long

此函数计算欧氏空间，x 或 y 方向上最上面的两点间的距离。.

Arguments

sPointFrom	
sPointTo	
lDistType	distance in x- or in y-direction
sDistName	name of distance
	The function will affect the following registers:

Return value

	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGetMiddlePointOfTwoPoints

Declare Function CKGetMiddlePointOfTwoPoints (ByVal sResName As String) As Long	
此函数计算点寄存器中最上面的两点间的直线的中点。	
Arguments	
sResName	name of the result
	The function will affect the following registers: point: [0]: middle point
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKGetNearestPointToLine

Declare Function CKGetNearestPointToLine (ByVal sResName As String) As Long	
此函数选择点列表中到线寄存器中最上面的直线最近的点。	
Arguments	
sResName	name of the result
	The function will affect the following registers: point: [0]: nearest point to line
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGetBestFitLineLeastSquareRoot

Declare Function CKGetBestFitLineLeastSquareRoot (ByVal sResName As String) As Long	
此函数用最小二乘法计算通过点列表的最合适的直线。	
Arguments	
sResName	name of the result
	The function will affect the following registers: line: [0]: best fit line
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKGetBestFitCircleLeastSquareRoot	
Declare Function CKGetBestFitCircleLeastSquareRoot (ByVal sResName As String) As Long	
此函数用最小二乘法计算通过点列表所有点的最适合的圆。	
Arguments	
sResName	name of the result
	The function will affect the following registers: circle: [0]: best fit circle point: [0]: center of circle value: [0]: radius of circle
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKGetCircleThroughThreePoints	
Declare Function CKGetCircleThroughThreePoints (ByVal sResName As String) As Long	
此函数选择点寄存器最上面的三个点，计算通过这三点的一个圆。	
Arguments	
sResName	name of the result
	The function will affect the following registers: circle: [0]: circle point: [0]: center of circle value: [0]: radius of circle
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKGetBestFitEllipseAndRectangle	
Declare Function CKGetBestFitEllipseAndRectangle (ByVal INOfPointsToUse As Long) As Long	
此函数计算通过点列表所有点的最适合的椭圆。	
Arguments	
INOfPointsToUse	number of points to use
	The function will affect the following registers: point: [0]: center of ellipse point: [1]: orientation point of ellipse point: [2]: corner 1 point of oriented rectangle point: [3]: corner 2 point of oriented rectangle point: [4]: corner 3 point of oriented rectangle point: [5]: corner 4 point of oriented rectangle
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGetBestFitEllipseAndRectangle	
Declare Function CKGetBestFitEllipseAndRectangle (ByVal INOfPointsToUse As Long) As Long	
此函数计算通过点列表所有点的最适合的椭圆。	
Arguments	
INOfPointsToUse	number of points to use
	The function will affect the following registers: point: [0]: center of ellipse point: [1]: orientation point of ellipse point: [2]: corner 1 point of oriented rectangle point: [3]: corner 2 point of oriented rectangle point: [4]: corner 3 point of oriented rectangle point: [5]: corner 4 point of oriented rectangle
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

度量矩形探头

CKGatRectSetPosition

Declare Function CKGatRectSetPosition (ByVal IIMIndex As Long, ByVal dYStart As Double, ByVal dXStart As Double, ByVal dXEnd As Double, ByVal dYEnd As Double, ByVal dWidth As Double, ByVal IAutoParm As Long) As Long

此函数设置度量工具的位置。

Arguments

IIMIndex	memory index
dXStart	x co-ordinate of the starting point of the search ray
dYStart	y co-ordinate of the starting point of the search ray
dXEnd	x co-ordinate of the end point of the search ray
dYEnd	y co-ordinate of the end point of the search ray
dWidth	width of rectangle (given in world units)
IAutoParm	Automatical parametrization Using parameter IAutoParm = 1, the width is selected from the value register, starting point and end point are fetched from the point register [0] [1].

Return value

	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKGatRectSetDetectionparameters

Declare Function CKGatRectSetDetection parameters (ByVal IColour As Long, ByVal IAutoParm As Long, ByVal IMinObjSize As Long, ByVal INumberOfPoints As Long, ByVal IFocusPointMode As Long) As Long

此函数设置度量工具的探测参数。

Arguments

IColour	Object color: PEG_BRIGHT,PEG_DARK or PEG_ANY
IAutoParm	automatical parametrisation
IMinObjectSize	minimum object size
INumberOfPoints	number of points to be detected by the gauging tool
IFocusPointMode	point that should be put into register

Return value

	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGatRectSelectEdgeDetectionMethodBinary

Declare Function CKGatRectSelectEdgeDetectionMethodBinary (ByVal IThreshold As Long) As Long

此函数选择边缘探测方法“binary”并设置参数。

Arguments

IThreshold	threshold for binary edge detection
------------	-------------------------------------

Return value

EVHD_SUCCESS

EVHD_ERR_PARM

CKGatRectSelectEdgeDetectionMethodMinimumContrast

Declare Function CKGatRectSelectEdgeDetectionMethodMinimumContrast (ByVal IMinContrast As Long, ByVal IEdgeWidth As Long) As Long

此函数选择边缘探测方法“minimum contrast”并设置参数。

Arguments

IMinimumContrast	minimum contrast (intensity difference) to define an edge
------------------	---

IEdgeWidth	distance (in pixels) between the pixels to define contrast
------------	---

Return value

EVHD_SUCCESS

EVHD_ERR_CONFIG

EVHD_ERR_PARM

CKGatRectDoApply

Declare Function CKGatRectDoApply () As Long

此函数使用具有实际参数的度量工具矩形。

Arguments

The function will affect the following registers: Value: [0]: number of points detected Point: [0]: point detected, started from „start“
--

Return value

EVHD_SUCCESS

EVHD_ERR_CONFIG

EVHD_ERR_PARM

CKGatRectDraw	
Declare Function CKGatRectDraw (ByVal IColour As Long) As Long	
此函数在指定的叠加颜色的图形叠加层绘制度量工具。	
Arguments	
IColour	graphic overlay color
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

计量圆

CKGatCircleSetPosition	
Declare Function CKGatCircleSetPosition (ByVal IIMIndex As Long, ByVal dXCenter As Double, ByVal dYCenter As Double, ByVal dRadius As Double, ByVal IAutoParm As Long) As Long	
此函数设置计量工具“圆”的位置。	
Arguments	
IIMIndex	memory index
dXCenter	x co-ordinate of the center
dYCenter	y co-ordinate of the center
dRadius	radius of the gauging tool
IAutoParm	automatical parametrization
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGatCircleSetDetection parameters	
Declare Function CKGatCircleSetDetection parameters (ByVal IColour As Long, ByVal IAutoParm As Long, ByVal IMinObjSize As Long, ByVal INumberOfPoints As Long, ByVal IFocusPointMode As Long) As Long	
此函数设置计量工具的边缘探测参数。	
Arguments	
IColour	object color: PEG_DARK or PEG_BRIGHT
IAutoParm	automatical parametrization
IMinObjectSize	minimum object size
INumberOfPoints	number of points to be detected by the gauging tool

IFocusPointMode	determines, which of the detected points will be put on top of the point register
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGatCircleSetDetection parameters

Declare Function CKGatCircleSetDetection parameters (ByVal IColour As Long, ByVal IAutoParm As Long, ByVal IMinObjSize As Long, ByVal INumberOfPoints As Long, ByVal IFocusPointMode As Long) As Long

此函数设置计量工具的边缘探测参数。

Arguments

IColour	object color: PEG_DARK or PEG_BRIGHT
IAutoParm	automatical parametrization
IMinObjectSize	minimum object size
INumberOfPoints	number of points to be detected by the gauging tool
IFocusPointMode	determines, which of the detected points will be put on top of the point register
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGatCircleSelectEdgeDetectionMethodBinary

Declare Function CKGatCircleSelectEdgeDetectionMethodBinary (ByVal IThreshold As Long) As Long

此函数选择边缘探测方法“binary”并设置参数。

Arguments

IThreshold	threshold for binary edge detection
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGatCircleSelectEdgeDetectionMethodMinimumContrast	
Declare Function CKGatCircleSelectEdgeDetectionMethodMinimumContrast (ByVal IMinContrast As Long,	
此函数选择边缘探测方法“minimum contrast”并设置参数。	
Arguments	
IMinimumContrast	minimum contrast (intensity difference) to define an edge
IEdgeWidth	distance (in pixels) between the pixels to define contrast
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGatCircleSetSearchDirection	
Declare Function CKGatCircleSetSearchDirection (ByVal ISearchDirection As Long) As Long	
此函数让度量工具探测中心到边缘的半径或反之。	
Arguments	
ISearchDirection	search direction: NEMGAT_OPTION_FROM_CENTER, NEMGAT_OPTION_TO_CENTER
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKGatCircleDoApply	
Declare Function CKGatCircleDoApply () As Long	
此函数将实际的参数应用于度量工具。	
Arguments	
	The function will affect the following registers: Value: [0]: number of points detected Point: [0]: point detected, started from 'start'
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKGatCircleDraw	
Declare Function CKGatCircleDraw (ByVal IColour As Long) As Long	
此函数在指定的叠加颜色的图形叠加层绘制度量工具。	
Arguments	
IColour	graphic overlay color
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

绘图功能

CKDrawPoint	
Declare Function CKDrawPoint (ByVal IIMIndex As Long, ByVal IColour As Long, ByVal IStackIndex As Long) As Long	
此函数以指定颜色绘制点寄存器的索引点。	
Arguments	
IIMIndex	Memory index
IColour	graphic overlay color
IStackIndex	index of point in point register (0 == topmost)
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKDrawLine	
Declare Function CKDrawLine (ByVal IIMIndex As Long, ByVal IColour As Long, ByVal IStackIndex As Long) As Long	
此函数以指定颜色绘制线寄存器的索引直线。	
Arguments	
IIMIndex	Memory index
IColour	graphic overlay color
IStackIndex	index of line in line register (0 == topmost)
Return value	
	EVHD_SUCCESS
	EVHD_ERR_PARM

CKDrawCircle	
Declare Function CKDrawCircle (ByVal IIMIndex As Long, ByVal IColour As Long, ByVal IStackIndex As Long) As Long	
此函数以特定颜色绘制圆寄存器的索引圆形。	
Arguments	
IIMIndex	Memory index
IColour	graphic overlay color
IStackIndex	index of circle in circle register (0 == topmost)
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

CKDrawArrow	
Declare Function CKDrawArrow (ByVal IIMIndex As Long, ByVal IColour As Long, ByVal dX1 As Double, ByVal dY1 As Double, ByVal dX2 As Double, ByVal dY2 As Double, ByVal IBothEnds As Long) As Long	
此函数在指定的起点向指定的终点绘制一个箭头。	
Arguments	
IIMIndex	Memory index
IColour	graphic overlay color
dX1	starting point, x co-ordinate
dY1	starting point, y co-ordinate
dX2	end point, x co-ordinate
dY2	end point, y co-ordinate
IBothEnds	0: arrow with one end
Return value	
	EVHD_SUCCESS

CKWriteString	
Declare Function CKWriteString (ByVal IIMIndex As Long, ByVal IX As Long, ByVal IY As Long, ByVal IForeColour As Long, ByVal IBackColour As Long, ByVal IFontSizeWidth As Long, ByVal IFontSizeHeight As Long, ByVal sCKWriteString As CKWriteString) As Long	
此函数直接在指定图像内存写一个指定字符串。	
Arguments	
IIMIndex	Memory index
IX	x position of the written text
IY	y position of the written text
IForeColour	intensity of foreground
IBackColour	intensity of background
IFontWidth	width of font
IFontHeight	height of font
sString	string to be written
Return value	
	EVHD_SUCCESS

PGDrawHoricontalLine	
Declare Function PGDrawHoricontalLine (ByVal IIndex As Long, ByVal IColour As Long, ByVal Ixstart As Long, ByVal Iystart As Long, ByVal ILength As Long) As Long	
此函数在图像内存中画一个水平直线。	
Arguments	
IIMIndex	Memory index
IColour	color (0..255) , if < 0: use XOR operation
IXStart	starting position x co-ordinate
IYStart	starting position y co-ordinate
ILength	length of the line
Return value	
	EVHD_SUCCESS

PGDrawVerticalLine	
Declare Function PGDrawVerticalLine (ByVal IIndex As Long, ByVal IColour As Long, ByVal Ixstart As Long, ByVal Iystart As Long, ByVal ILength As Long) As Long	
此函数在图像内存中画一条垂直直线。	
Arguments	
IIMIndex	Memory index
IColour	color (0..255) , if < 0: use XOR operation
IXStart	starting position x co-ordinate
IYStart	starting position y co-ordinate
ILength	length of the line
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

PGDrawLine	
Declare Function PGDrawLine (ByVal IIndex As Long, ByVal IColour As Long, ByVal Ixstart As Long, ByVal Iystart As Long, ByVal Ixend As Long, ByVal Iyend As Long) As Long	
此函数在图像内存中画一条直线。	
Arguments	
IIMIndex	Memory index
IColour	color (0..255) , if < 0: use XOR operation
IXStart	starting position x co-ordinate
IYStart	starting position y co-ordinate
IXEnd	end position x co-ordinate
IYEnd	end position y co-ordinate
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

PGDrawCircle	
Declare Function PGDrawCircle (ByVal IIndex As Long, ByVal IColour As Long, ByVal Ixstart As Long, ByVal Iystart As Long, ByVal IRadius As Long) As Long	
此函数在图像内存中画一个圆。	
Arguments	
IIMIndex	Memory index
IColour	color (0..255) , if < 0: use XOR operation
IXStart	center point x co-ordinate
IYStart	center point y co-ordinate
IRadius	Radius of the circle
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

PGDrawCircleExact	
Declare Function PGDrawCircleExact (ByVal IIndex As Long, ByVal IColour As Long, ByVal Ixstart As Long, ByVal Iystart As Long, ByVal IRadius As Long) As Long	
此函数在图像内存中画一个圆（图像轮廓的裁剪圆）。	
Arguments	
IIMIndex	Memory index
IColour	color (0..255) , if < 0: use XOR operation
IXStart	center point x co-ordinate
IYStart	center point y co-ordinate
IRadius	Radius of the circle
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

PGDrawRectangle	
Declare Function PGDrawRectangle (ByVal IIndex As Long, ByVal IColour As Long, ByVal lx As Long, ByVal ly As Long, ByVal lw As Long, ByVal lh As Long) As Long	
此函数在指定位置画一个矩形。	
Arguments	
IIndex	Memory index
IColour	color (0..255) , if < 0: use XOR operation
lx	x co-ordinate of upper left corner
ly	y co-ordinate of upper left corner
lw	width of window rectangle
lh	height of window rectangle
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

PGFillRectangle	
Declare Function PGFillRectangle (ByVal IIndex As Long, ByVal IColour As Long, ByVal lx As Long, ByVal ly As Long, ByVal lw As Long, ByVal lh As Long) As Long	
此函数在指定图像内存画一个填充的窗口矩形。	
Arguments	
IIndex	Memory index
IColour	color (0..255) , if < 0: use XOR operation
lx	x co-ordinate of upper left corner
ly	y co-ordinate of upper left corner
lw	width of window rectangle
lh	height of window rectangle
Return value	
	EVHD_SUCCESS
	EVHD_ERR_CONFIG
	EVHD_ERR_PARM

PGCopyImage		
Declare Function PGCopyImage (ByVal ISourceIMPar As Long, ByVal ISourceXPar As Long, ByVal ISourceYPar As Long, ByVal IWidthPar As Long, ByVal IHightPar As Long, ByVal IDestIMPar As Long, ByVal IDestXPar As Long, ByVal IDestYPar As Long) As Long		
此函数从一个图像内存中复制一个指定窗口到另一个图像内存。		
Arguments		
ISourceIMPar	index of source image memory	
ISourceXPar	x position of source window	
ISourceYPar	y position of source window	
IWidthPar	width of window to copy	
IHightPar	height of window to copy	
IDestIMPar	index of destination image memory	
IDestXPar	x position of destination window	
IDestYPar	y position of destination window	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_PARM	window too small (width or height < 1)
	EVHD_ERR_PARM	window wraps image
	EVHD_ERR_PARM	illegal EVHD image memory index
	EVHD_ERR_CONFIG	error in EVHD configuration
	EVHD_ERR_COORD	EVHD coordinate out of range
	EVHD_ERR_IM	no valid image in memory

PGTranslateAndRotateImage		
Declare Function PGTranslateAndRotateImage (ByVal ISourceIMPar As Long, ByVal ISourceXPar As Long, ByVal ISourceYPar As Long, ByVal IWidthPar As Long, ByVal IHightPar As Long, ByVal dSrcTorseXPar As Double, ByVal dSrcTorseYPar As Double, ByVal IDestIMPar As Long, ByVal dDestTorseXPar As Double, ByVal dDestTorseyPar As Double, ByVal dTorseAnglePar As Double, ByVal lIntpol4NFuncPar As Long) As Long		
此函数从一个图像内存翻译和旋转一个指定窗口到另一个图像内存。		
Arguments		
ISourceIMPar	index of source image memory	
ISourceXPar	x position of source window	
ISourceYPar	y position of source window	
IWidthPar	width of window to copy	
IHightPar	height of window to copy	
dSrcTorseXPar	Point x of source image torsion center	

dSrcTorseYPar	Point y of source image torsion center	
IDestIMPar	index of destination image memory	
dDestTorseXPar	translation point x of destination image torsion center	
dDestTorseyPar	translation point y of destination image torsion center	
dTorseAnglePar	torsion angle of the window	
lIntpol4NFuncPar	4-neighborhood interpolation method of rotation	
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_PARM	window too small (width or height < 1)
	EVHD_ERR_PARM	window wraps image
	EVHD_ERR_PARM	illegal EVHD image memory index
	EVHD_ERR_CONFIG	error in EVHD configuration
	EVHD_ERR_COORD	EVHD coordinate out of range
	EVHD_ERR_IM	no valid image in memory

PGSubtractImage

Declare Function PGSubtractImage (ByVal lImg1IMPar As Long, ByVal lImg1XPar As Long, ByVal lImg1YPar As Long, ByVal lWidthPar As Long, ByVal lHightPar As Long, ByVal lImg2IMPar As Long, ByVal lImg2XPar As Long, ByVal lImg2YPar As Long, ByVal lDestIMPar As Long, ByVal lDestXPar As Long, ByVal lDestYPar As Long) As Long

此函数从源图像内存2的窗口像素强度减掉源图像内存1的窗口像素强度，并在目标图像内存窗口写下差异的绝对值。

Arguments

lImg1IMPar	index of source image memory 1
lImg1XPar	x position of source window 1
lImg1YPar	y position of source window 1
lWidthPar	width of windows to subtract
lHightPar	height of windows to subtract
lImg2IMPar	index of source image memory 2
lImg2XPar	x position of source window 2
lImg2YPar	y position of source window 2
lDestIMPar	index of destination image memory
lDestXPar	x position of destination window
lDestYPar	y position of destination window

Return value		
	EVHD_SUCCESS	
	EVHD_ERR_PARM	window too small (width or height < 1)
	EVHD_ERR_PARM	window wraps image
	EVHD_ERR_PARM	illegal EVHD image memory index
	EVHD_ERR_CONFIG	error in EVHD configuration
	EVHD_ERR_COORD	EVHD coordinate out of range
	EVHD_ERR_IM	no valid image in memory

PGSubtractAndMaximumImage

Declare Function PGSubtractAndMaximumImage (ByVal xxx As Long) As Long

此函数从源图像内存2的窗口像素强度减掉源图像内存1的窗口像素强度，如果差异的绝对值大于目标内存中的值，在目标图像内存窗口写下差异的绝对值。

Arguments

IImg1IMPar		index of source image memory 1
IImg1XPar		x position of source window 1
IImg1YPar		y position of source window 1
IWidthPar	width of win-	width of windows to subtract
IHeightPar	height of win-	height of windows to subtract
IImg2IMPar		index of source image memory 2
IImg2XPar		x position of source window 2
IImg2YPar		y position of source window 2
IDestIMPar		index of destination image memory
IDestXPar		x position of destination window
IDestYPar		y position of destination window
Return value		
	EVHD_SUCCESS	
	EVHD_ERR_PARM	window too small (width or height < 1)
	EVHD_ERR_PARM	window wraps image
	EVHD_ERR_PARM	illegal EVHD image memory index

	EVHD_ERR_CONFIG	error in EVHD configuration
	EVHD_ERR_COORD	EVHD coordinate out of range
	EVHD_ERR_IM	no valid image in memory

PGApplyFilterRingToRectangle

Declare Function PGApplyFilterRingToRectangle (ByVal lImg1IMPar As Long, ByVal lImg1XPar As Long, ByVal lImg1YPar As Long, ByVal lWidthPar As Long, ByVal lHightPar As Long, ByVal lImg2IMPar As Long, ByVal lImg2XPar As Long, ByVal lImg2YPar As Long, ByVal lDestIMPar As Long, ByVal lDestXPar As Long, ByVal lDestYPar As Long) As Long

此函数适用于特殊的过滤器在一个长方形的窗口包装环的内容。

Arguments

lImg1IMPar	index of source image memory 1
lImg1XPar	x position of source window 1
lImg1YPar	y position of source window 1
lWidthPar	width of windows to subtract
lHightPar	height of windows to subtract
lImg2IMPar	index of source image memory 2
lImg2XPar	x position of source window 2
lImg2YPar	y position of source window 2
lDestIMPar	index of destination image memory
lDestXPar	x position of destination window
lDestYPar	y position of destination window

Return value

	EVHD_SUCCESS	
	EVHD_ERR_PARM	window too small (width or height < 1)
	EVHD_ERR_PARM	window wraps image
	EVHD_ERR_PARM	illegal EVHD image memory index
	EVHD_ERR_CONFIG	error in EVHD configuration
	EVHD_ERR_COORD	EVHD coordinate out of range
	EVHD_ERR_IM	no valid image in memory

9 附件

9.1 SmartBuilder 软件边缘探测方法

SmartBuilder 文档关于边缘检测方法。

问题：根据图形资料，应选择哪种探头方法？

二进制边缘检测法 Binary edge-detection（方法：Binary）

这种方法通过线性插值计算，阈值高于或低于目标像素之间的边缘位置。边缘检测的二进制是最快以及最精确的方法。然而，图像对比度必须非常稳定，因为二进制以一个固定的阈值进行边缘检测工作。这意味着，亮度偏移可能导致测量误差。使用此方法的一个例子是透射光的金属部分。

对比边缘探测 Contrast edge-detection（方法：Minimum contrast）

（在 Smartbuilder 中，如果激活了自动亮度调整，这个过程是标准的。）

这种方法没有分析绝对灰度值，但比较灰度值的差异。因此，如果边缘的灰度值从0到100或灰度值从50到150是没关系的。灰度值序列的搜索是“平台增加对应平台减少”。

计算标称的灰度值：

如果设置值66%作为边对称，设置的灰度值=第一平台的灰度值+/-第一平台和第二平台间的差距值的66%（根据边缘方向）。

计算位置：

在灰度值过渡发生的像素间使用线性插值（见图纸）。

使用二进制方法失败时使用此方法，也就是说，可能亮度有变化。（例如：木材的结孔，其背景亮度可以有很大差异）。

注：上述两个方法是边缘检测任务中最常用的方法。

对比边缘探测 Contrast edge-detection（方法：Minimum contrast 2）

（在 SmartBuilder 中，如果在探测命令中激活了自动亮度调整，基于阈值的界面过滤）

该命令的工作方式类似于前面的方法。虽然干扰不根据尺寸来确定，但根据灰度值来确定。在探测方向上从亮到暗的边缘被拒绝，其中在平台2发生的干扰比平台1减最小对比度还亮。在检测方向上由暗到亮边缘被拒绝，因为平台2发生的干扰，比平台1加最小对比度还暗。以下的合成图像作为例子：



你能看到一个由亮到暗的过渡，显示了一个较短的明亮干扰。上面描述的方法不会将此干扰探测为边缘。

（方法： minimum contrast 3）

（在 SmartBuilder 中，如果在探测命令中激活了自动亮度调整，测量阈值）

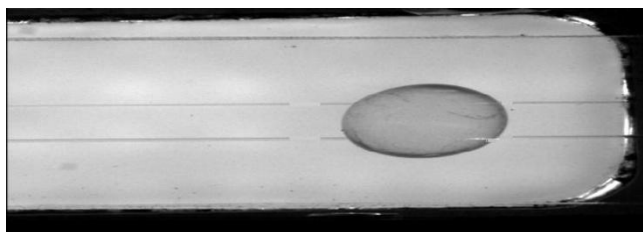
在一些边缘梯度用上述两种方法不能找到第二平台，因为，对比度条件在探测到边缘后又再次被满足，（即 x 和 x 的灰度值差异大于最小对比度）



两个例子：

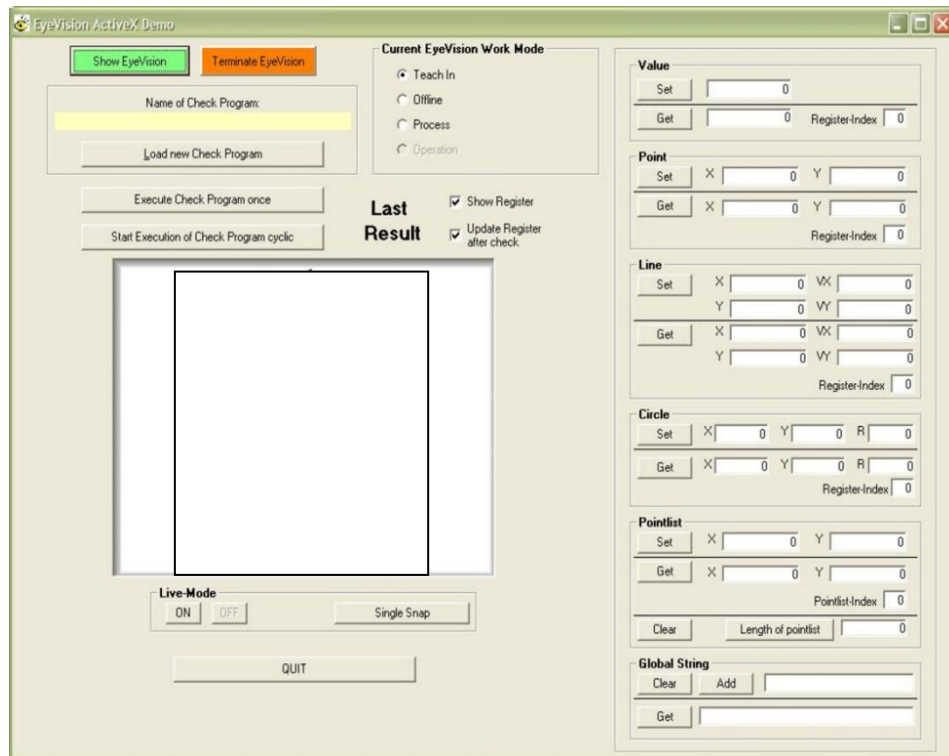
图像显示了一个由暗到亮的过渡，不是由平台，而是一个长期的上升坡道。上述两种方法无法检测到边缘。

另一个例子：在胶囊的边缘，是一个很好的由亮到暗的过渡，却又跟着一个较长的由暗到亮的斜坡。



9.2 Active X 接口

SamrtBuilder 软件也可以集成到自己的软件产品。这就是为什么软件提供微软的 ActiveX 接口。详细信息请查阅 SmartBuilder ActiveX 手册。



9.3 文件扩展名

*.ckp	检查程序	SmartBuilder 检查程序自动保存为此扩展名。
*.cgh	OCR 示例文件	保存示例的 OCR 数据。
*.sbs	BASIC	输出的 BASIC 源文件。
*.pat	相关性	相关性示例。
*.bmp	图像	微软位图图像。
*.jpg	图像	JPG 格式图像。
*.txt	数据	不同的命令保存数据到 ASCII 文本。
*.ini	结构文件	基于微软的 INI 文件数据。

9.4 SmartBuilder 命令使用的寄存器

在这一章中，我们描述 SmartBuilder 软件不同命令完成后，不同寄存器的检取。尤其是对于多功能命令，有结果的寄存器检取并不明显。

命令	值寄存器		点寄存器		直线寄存器		点列表
动作	索引	描述	索引	描述	索引	描述	描述
访问变量							
回到寄存器> 值	0	来自变量的值					
回到寄存器> 点			0	来自变量的点			
回到寄存器> 直线					0	来自变量的直线	
回到寄存器> 结果	0	来自变量的 OK / NOK / 警告					
数学> 值	0	结果					
数学> 点			0	结果			
角							
	0	角					
卡尺 – 水平/垂直							
	0	距离	0	最上面的点			
			1	最右边的点			
			2	最下面的点			
			3	最左边的点			
圆							
	0	边的数量	0	第一条边坐标			所有边
计数							
	0	对象数量					

保存在点寄存器			0-N	平衡点			
保存在点列表							平衡点
横截面							
	0	直线	0,1	测量点			
距离							
	0	直线					
边缘计数（直线）							
	0	边的数量	0	第一条边坐标		所有边	0
自由卡尺							
	0	距离					
图像获取							
循环加载/循环保存	0	图像索引号					
图像信息							
亮度	0	平均灰度值					
	1	最小灰度值					
	2	最大灰度值					
焦点值	0	图像锐度					
图像格式	0	图像水平像素					

	1	图像垂直像素					
INI-文件访问							
读 INI-文件 > 值	0	来自文件的值					
读 INI-文件 > 点			0	来自文件的值			
测量间隙							
	0	最短距离	0	探头 1 最短			所有点

	1	最长距离	1	探头 2 最短			
			2	探头 1 最长			
			3	探头 2 最长			
点							
	0	点坐标					点坐标
点列表							
重置							重置所有为0
环值	0	发现的点的数量					
对象探测			0	平衡点			
			1	矩形的边			
			2	1. 矩形的角			
			3	2. 矩形的角			
			4	3. 矩形的角			
			5	4. 矩形的角			
计算距离	0	平均距离					
	1	最短距离					
	2	最大距离					
排序列表							所有点排序
几何中心			0	点坐标			
加载							来自文件的点
探头点							
	0	检测到的点的数量	0	第一/最后/中间点			
探头点（环形探头）							
	0	检测到的点的数量	0	第一/最后/中间点			

直线							
					0	直线坐标	
直线距离							
	0	距离					
面积检查							
	0	检测到的像素数量					
颜色控制							
	0	检测到的像素数量					
	1	平均色度					
	2	最小色度					
	3	最大色度					
	4	平均饱和度					
	5	最小饱和度					
	6	最大饱和度					
	7	平均亮度					
	8	最小亮度					
	9	最大亮度					
颜色过滤							
	0	检测到的像素数量					
边缘检查							
	0	未检测到的点的数量					
边缘追踪							
	0	探测到的点的数量					点坐标

秒表							
	0	毫秒时间					