

CVL 9.0 发行说明

2019 十一月 14
Revision: 9.0.0.1

法律声明

本文档所述软件按照许可提供，仅可根据该许可的条款使用或复制，并须包含本页所示版权声明。该软件、本文档或其任何副本均不得向该被许可方以外的任何人提供或供其使用。该软件的权益和所有权属于 Cognex Corporation 及其许可方。若在非 Cognex Corporation 提供的设备上使用此软件，Cognex Corporation 不对其使用情况或可靠性负责。Cognex Corporation 对于所述软件及其适销性、不侵权性或任何特定目的适用性不作出任何明示或暗示保证。

本文中的信息可能随时更改，恕不另行通知，且不应视作 Cognex Corporation 的承诺。Cognex Corporation 对于可能出现在本文或相关软件中的任何错误概不负责。

除非另有说明，本文示例所用的公司、名称和数据均为虚构。未经 Cognex Corporation 书面许可，不得出于任何目的以任何形式或任何方式（电子或机械）复制或传播本文任何部分，也不得转换为任何其他介质或语言。

版权所有 © 2019. Cognex Corporation。保留所有权利。

Cognex 提供的部分硬件和软件可能受到 Cognex 网站列出的一项或多项美国和外国专利、以及未决专利的保护：cognex.com/patents

以下为 Cognex Corporation 的注册商标：

Cognex、2DMAX、Advantage、AlignPlus、Assemblyplus、Check it with Checker、Checker、Cognex Vision for Industry、Cognex VSOC、CVL、DataMan、DisplayInspect、DVT、EasyBuilder、Hotbars、IDMax、In-Sight、Laser Killer、MVS-8000、OmniView、PatFind、PatFlex、PatInspect、PatMax、PatQuick、SensorView、SmartView、SmartAdvisor、SmartLearn、UltraLight、Vision Solutions、VisionPro、VisionView

以下为 Cognex Corporation 的商标：

Cognex 徽标、1DMax、3D-Locate、3DMax、BGAI、Checkpoint、Cognex VSoC、CVC-1000、FFD、iLearn、In-Sight（带十字准线的设计标志）、In-Sight 2000、InspectEdge、Inspection Designer、MVS、NotchMax、OCRMax、PatMax RedLine、ProofRead、SmartSync、ProfilePlus、SmartDisplay、SmartSystem、SMD4、VisiFlex、Xpand

部分版权所有 © Microsoft Corporation。保留所有权利。

部分版权所有 © MadCap Software, Inc.。保留所有权利。

本文涉及的其他产品和公司商标是其各自所有者的商标。

CVL 9.0 发行说明

CVL 版本 9.0 包含新功能或自先前 CVL 8.0 版本以来更改的功能。

系统要求

PC 必须符合安装和使用 CVL 的最低要求。

PC 要求

PC 应符合以下最低硬件要求：

- 1 Ghz 或更快的英特尔奔腾（或兼容）CPU，支持 MMX/SSE2 指令集。
- 开放式 PCI 插槽，如果正在使用 PCI 总线 Cognex 图像采集卡或 GigE 相机适配器
- 直接连接相机的适当端口（GigE 视觉的千兆以太网）
- PCI、PCI Express（推荐）或 AGP 视频适配器
 - CVL 需要至少 8 MB 的显卡内存，可提供有限功能。
 - Cognex 建议使用独立显卡，通过集成主板的基于芯片的视频系统插入主板插槽。
 - Cognex 建议使用至少 16 MB 内存的 AGP 或 PCI Express x16 独立显卡。
 - 显卡必须支持 DirectX 8.1 或更高版本。

Cognex 还提出了以下建议：

- PC 应该有足够的 RAM，这样 CVL 应用程序无需分页磁盘。
- 将 PC 的 BIOS 设置为支持非即插即用操作系统。
- 将 Windows 桌面设置为 8 位、16 位或 32 位的颜色模式。Windows 桌面必须在 16 位或 32 位模式下才能显示彩色图像。
- Cognex 建议 PC 使用总线控制的 DMA 磁盘驱动器，而不是 IDE (ATA) 硬盘的 PIO 驱动器。查主板供应商，以确定供应商是否提供总线控制的 IDE 驱动器。
- 如果 PC 有 64 MB 或更多内存，执行以下步骤，禁用内核和驱动器分页并将其保存在存储器中，以便更快访问：
 - 备份当前注册表设置。
 - 打开注册表编辑器并导航到：
HKLM/SYSTEM/CurrentControlSet/Control/Session Manager/Memory Management
 - 将秘钥 DisablePagingExecutive 设置为 1，以禁用内核和驱动器分页。
 - 重启 PC。

操作系统

CVL 9.0 支持使用本地语言（英文、日文、德文、韩文和简体中文）和以下 Windows 64 位操作系统在单处理器或多处理器上开发和部署：

- Windows 7 Premium
- Windows 7 Professional
- Windows 7 Ultimate (64 位)

- Windows 10
- Windows 10 IoT Enterprise

注意这可能是最后一个支持 Windows 7 的 CVL 版本。

支持的 Microsoft Visual Studio 开发环境

CVL 9.0 支持使用以下开发环境开发和部署视觉应用程序：

开发环境	编译器版本	服务包
Microsoft Visual Studio 2015	Visual C++ v14.0	Update 3
Microsoft Visual Studio 2017	Visual C++ v14.16	

CVL 与 Microsoft Visual Studio 2015 和 Microsoft Visual Studio 2017 CVL C++ 程序库一起发行。

CVL 9.0 是非托管 C++ SDK，仅支持在非托管应用程序中使用 Microsoft Visual Studio 构建的应用程序。CVL 9.0 不支持使用托管 .NET 构建的应用程序（使用 `/clr` 编译器开关）；也称为混合模式。

支持的取像平台

CVL 9.0 支持以下取像平台：

- Cognex MVS-8511
- Cognex MVS-8511e
- Cognex MVS-8514
- Cognex MVS-8514e
- GigE 视觉相机
- CFG-8724 系列图像采集卡
- 符合 CVL 用户指南所述的成像设备接口的第三方成像设备。

此版本中包含的新功能

CVL 9.0 是支持新功能的通用版本。

新 USB 安全密钥

此版本需要新的 Cognex USB 安全密钥附加到硬件平台，以确保软件正确授权。先前版本中支持的安全选项未与此 CVL 版本一起运行。现有的 Cognex 图片采集卡仍将执行各自的图像采集功能，与先前版本一样。

Cognex 支持在有时间限制的基础上进行应用程序开发的安全密钥，以及在生产环境中与应用程序一起使用的部署密钥。

更多信息，请联系您的 Cognex 销售代表。

Visual Studio 2017

此版本增加对使用 Microsoft Visual Studio 2017 (v. 15.10) 开发和编译应用程序的支持。

LineMax 工具

此版本增加了对 LineMax 工具的支持，该工具可以定位图像中所有的候选边缘点，然后根据规定的标准来匹配可能的最佳线段。LineMax 工具支持 8 位和 16 位图像。

LineMax 工具比拟合工具更耐用，拟合工具需要精确放置卡尺以便找到单独边缘点。Cognex 推荐新应用程序使用 LineMax 工具，并继续支持现有应用程序的拟合工具。

此版本中包含的 CVL 类引用不包括 LineMax 工具的 API 介绍。在 %VISION_ROOT%\defs\ch_cvl. 中找到标头文件及其源代码注释。

有关 LineMax 工具的详细信息，请参阅 CVL 视觉工具指南。

CFG-8724 图像采集卡

此版本支持具有集成安全许可的 CFG-8724 4 端口 GigE 视觉图像采集卡。CFG-8724 允许将最多 4 个 GigE 视觉相机连接到 PC 上。此外，可以在同一台 PC 上安装多个 CFG-8724 图像采集卡。CFG-8724 还可以存储您的 Cognex 软件许可证，以确保您的 Cognex 软件已获得适当的使用授权。请注意，CVL 只会从单个 CFG-8724 读取软件许可证。有关安全性和版本信息的更多详情，请参阅 CVL 用户指南。

Cognex 会提供 CFG-8724 直接替换 CFG-8174e，在未来无法使用。

如果您在安装 CFG-8724 之后安装了 Cognex 视觉软件，请注意必须在安装软件之后关闭计算机电源。仅执行一次热重启就可以防止 CFG-8724 正常运行。

支持 16 位图像显示

此版本增加了对显示使用相同 API（用于 8 位图像类型 `ccPelBuffer<c_UInt8>`）的 16 位图像类型 `ccPelBuffer<c_UInt16>` 的支持。CVL 自动映射 16 位灰度像素值范围到 8 位显示范围。

请注意，具有大范围灰度值的 16 位图像会在显示屏中保持该范围，而具有小范围灰度值的 16 位图像会在显示屏中看到扩展的该范围。

例如，如果 16 位图像包含深灰和浅灰度值的像素，CVL 会在 8 位深灰和浅灰度值范围内显示该图像。但是，如果 16 位图像仅包含深灰度值，则最高像素值会在 8 位显示屏中显示为浅灰度值。

有关 CVL 图像显示 API 详情，请参阅标头文件 `%VISION_ROOT%\defs\ch_cvl\display.h`。

CNLSearch 模型尺寸

此版本支持 CNLSearch 工具的新 4x4 模型尺寸。先前 CVL 版本支持最小的 CNLSearch 8x8 的模型尺寸。

更多视觉工具的 16 位图像支持

此版本增加对以下视觉工具的 16 位图像支持：

- 卡尺
- 图像扭转器
- 高斯采样滤波器掩模
- PelSet 工具掩模

兼容性说明兼容性说明

有关使用 Cognex 硬件和 CVL 9.0 软件的问题，请参阅以下部分。

大型图像兼容性问题

本部分介绍了与支持大型图像相关的兼容性问题。

仿射采样工具界面更改

在 CVL 6.7 CR1 和较早版本中，构件用于为仿射采样工具指定 X 和 Y 方向的样本数量

(`ccAffineSamplingParams::xNumSamples`和`ccAffineSamplingParams::yNumSamples`)，是类型 `c_Int16`。

为了支持对大型图像进行采样，在以 CVL 6.7 CR3 开始的发行版本中，使用 `c_Int32` 值，存储样本数量。

将 `ccAffineSamplingParams` 中的 setter 函数简单更改为 `c_Int32` 值；这并不会导致与现有代码不兼容，因为 C++ 编译器将 16 位值提升为 32 位值。但是，在无法与现有代码不兼容的情况下，不能更改 getter 函数。CVL 6.7 CR3 增加了两个新构件到 `ccAffineSamplingParams`：

- `ccAffineSamplingParams::xNumSamples32()` 返回 X 方向上的样本数量。
- `ccAffineSamplingParams::yNumSamples32()` 返回 Y 方向上的样本数量。

弃用返回 16 位值的现有 getter 函数，如果样本数量大于或等于 32768，会引发错误。

并不是所有仿射采样模式都支持大型图像

仿射采样工具仅支持以下采样模式，用于在 X 维或 Y 维大于 32768 像素的输入或输出图像：

- `ccAffineSamplingParams::eNone`
- `ccAffineSamplingParams::eBilinearAccurate`

其他模式 (`eBilinearApprox`, `eHighPrecision`, 与 `eBilinear`) 不支持；如果指定一种不支持的模式，并且输入或输出图像的 X 维或 Y 维大于 32768，则该工具会引发错误。

更改边缘工具接口支持大型图像

在 CVL 6.7 CR1 和较早版本中，边缘工具使用 16 位值存储亚像素边缘 (edgelets) 的全像素位置。这意味着在 -32768 到 +32767 范围之外的全像素位置的任何 edgelets 都无法表示。

从 CVL 6.7 CR3 开始，边缘工具允许为 edgelets 指定存储方法。

- 如要求 `ccEdgeletDefs::eRequestContiguous`，edgelets 将存储在单个连续块中。如果 edgelet 没有在 -32768 到 +32767 范围之外 X 或 Y 轴图像的全像素位置，则将 edgelets 存储在 `ccEdgelet` 向量中并通过 `ccEdgeletSet::edges()` 获取。如果任何 edgelet 具有在 -32768 到 +32767 范围之外 X 或 Y 轴图像的全像素位置，则将 edgelets 存储在 `ccEdgelet2` 向量中并通过 `ccEdgeletSet::edges2()` 获取或通过 `ccEdgeletIterator` 或通过调用 `ccEdgeletSet::begin()` 获得的 `ccEdgeletIterator_const` 类别获取。
- 如要求 `ccEdgeletDefs::eRequestEdges2`，Edgelets 存储在 `ccEdgelet2` 向量中并通过 `ccEdgeletSet::edges2()` 获取或通过 `ccEdgeletIterator` 或通过调用 `ccEdgeletSet::begin()` 获得的 `ccEdgeletIterator_const` 类别获取。
- 如要求 `ccEdgeletDefs::eRequestEdgesAndOffsets`，Edgelets 存储在一个或多个独立内存块中，其中每个内存块包含一系列 `ccEdgelet` 对象和一对 32 位偏移说明符。使用 `ccEdgeletIterator` 或 `ccEdgeletIterator_const` 类别获得 Edgelets。通过调用 `ccEdgeletSet::begin()` 获取其中一个对象。

多个内存块仅在至少一个 edgelet 具有在 -32768 到 +32767 范围之外 X 或 Y 轴图像的全像素位置时使用。如果只创建单内存块，则可以使用 `ccEdgeletSet::edges()` 和 `ccEdgeletIterator()` 方法，获取数集的 edgelets。

因为 `ccEdgeletIterator` 或通过调用 `ccEdgeletSet::begin()` 获得的 `ccEdgeletIterator_const` 类别获取可以用于获取 edgelets，无论存储方法如何，Cognex 建议您使用此接口。

某些 CVL 和 VisionPro 版本创建的 CDB 文件无效

使用随 CVL 6.8 CR1、CVL 6.8 CR2 或 VisionPro 6.0 一起提供的 IDBEdit 程序创建的任何 CDB 或 IDB 文件都无法被早期 CVL 和 VisionPro 版本提供的 IDBEdit 程序读取。

随 CVL 9.0（以及 VisionPro 6.1 或 6.2）一起提供的 IDBEdit 程序可用于修复无效的 CDB 或 IDB 文件。只需使用较新的 IDBEdit 程序打开受影响的文件，然后选择**文件->保存为...**然后用其他名称保存。新保存的文件均可通过**全部IDBEdit**版本读取。

带大型图像的 CDB 文件与早期的 CVL 版本不兼容

如果使用构建 CVL 6.8 CR3 的 CVL 应用程序保存大于在 X 维或 Y 维的 32768 像素的图像，则使用 CVL 版本 6.7 CR1 和早期版本，尝试加载图像到构建的 CVL 应用程序会产生 1x1 像素的图像。

设备驱动程序反向兼容性

CVL 9.0 提供的设备驱动程序与下表中列出的版本反向兼容。在这个意义上，反向兼容性意味着当使用此版本的新驱动程序时，指定的旧 CVL 版本会继续运行。

图像采集卡或相机	最早的 CVL 版本兼容 8.0 驱动程序 (Windows 7)	最早的 CVL 版本兼容 8.0 驱动程序 (Windows 8.1 and 10)
MVS-8511 MVS-8514 MVS-8511e MVS-8514e	CVL 7.0 (64 位)	CVL 8.0
GigE 视觉性能驱动程序	CVL 7.1 CR2 (64 位)	CVL 8.0

MVS-8510 系列图像采集卡兼容性问题

本部分介绍了影响 MVS-8510 和 MVS-8510e 图像采集卡使用的兼容性问题。

MVS-8510 系列图像采集卡的对比度和亮度

对比度和亮度设置位于具有 **ccContrastBrightnessProp** 类功能的 CVL 中。在通常使用的 0.05 到 0.95（对比度设置）和 0.20 到 0.80（亮度设置）的范围内，采集图像的灰度值会根据使用的电路板略有不同。

- 在相同的对比度和亮度设置中，MVS-8511、MVS-8511e、MVS-8514 和 MVS-8514e 图像采集卡产生的灰度级值可能多达 10 灰度级高于或 6 灰度级低于具有同一输入信号的（不支持）MVS-8501 和 MVS-8504 产生的像素值。在典型的输入范围内，灰度值的差值中位数大约为 0。
- 在相同的对比度和亮度设置中，MVS-8511、MVS-8511e、MVS-8514 和 MVS-8514e 图像采集卡产生的灰度级值可能多达 4 灰度级高于或 12 灰度级低于具有同一输入信号的（不支持）MVS-8500Le 和 MVS-8504e 产生的像素值。在典型的输入范围内，灰度值的差值中位数大约为 -2（较新的图像采集卡产生的值较低）。

可以通过调整亮度和对比度设置，最小化特定输入值范围内的灰度值差。

带电缆 300-0406 的软件地址

Cognex 电缆 300-0406 用于连接索尼 DXC-390 彩色相机和一个单色模拟相机到 MVS-8514 和 MVS-8514e。

使用该电缆时，连接到彩色相机的 R、G 和 B 分支电缆使用相机位置 0、1 和 2。这意味着连接到第四分支电缆的单色相机的软件地址为 3。使用电缆 300-0406 时，使用 CVL 代码中的该地址来控制单色相机。

仅使用 TTL 电缆选项时闪光灯极性反转

对于所有 MVS-8510 系列图像采集卡的三个并行 I/O 电缆选项中的两个，频闪线路可光学隔离。但是，当使用 TTL 直通电缆选项（电缆 300-0390 和 TTL 连接模块 800-5818-1）时，与相同频闪线路光学隔离时相比，这四条频闪线路的极性反转。

因此，当在全光学（300-0389）或半光学（300-0399）电缆选项上使用闪光灯时，有关 CVL 类引用中 `ccStrobeProp::strobeHigh()` 的信息书写准确：

```
void strobeHigh(bool polarity);
```

当极性准确，设置为高活动脉冲。如果错误，设置为低活动脉冲。

但是，在 TTL 直通电缆选项 300-0390 上使用闪光灯时，极性信息反转：

```
void strobeHigh(bool polarity);
```

当极性准确，设置为低活动脉冲。如果错误，设置为高活动脉冲。

已知漏洞

本部分列出了此版本中已知漏洞。

取像

本部分介绍 CVL 9.0 中图像采集的已知问题。

RGB 包函数错误复制转换

如果目标像素缓冲区预先分配，并且与源像素缓冲区的大小不同，则 `cfRGBPack()` 函数无法正确复制转换。

解决方法

执行下列任何一项内容：

- 调用 `cfRGBPack()` 后手动复制转换
- 确保传递给 `cfRGBPack()` 的目标像素缓冲区未约束。

`ccLightProp::LightsOff()` 不运行。

调用 `ccLightProp::lightsOff()` 函数无效果。可以使用以下代码来解决该问题：

```
fifo->properties().lightPower(0.0);
```

```
fifo->properties().auxLightPower(0.0);
```

```
fifo->prepare(0.0);
```

MVS-8500 取像

本部分介绍 CVL 9.0 中 MVS-8500 取像的已知问题。

使用 8500 图像采集卡，在采集包含颜色信息的视频信号时 ExtSync 取像可能会失锁

如果将包含颜色信息的视频信号（如 NTSC 或 PAL）提供给 8500 图像采集卡，根据相机正在查看的场景类型，ExtSync 取像可能会失败。该问题是由于色彩爆裂和作为 RS170 或 CCIR 信号一部分的频率为 ~ 3.58 MHz (NTSC) 和 4.43MHz (PAL) 的载波编码的颜色信息。该颜色信息会干扰 8500 正确检测 Hsync 脉冲并导致错误锁定。

解决方法

解决方法是在还没有输入进 8500 前，删除视频信号中的颜色信息。一个简单的方法是获取从复合视频到 S 视频 (Y/C) 的转换器。一旦转换，Y 组件没有任何颜色信息。

断言会与索尼 DXC-390 相机同时发生

使用外部同步源时，如果在进行取像时相机电缆断开，则可能发生断言。

队列软件触发器的四核 CPU 上的 MVS-8500e 取像错误

在某些特定系统配置下，MVS-8500e 图像采集卡可能偶尔会出现 `isAbnormal()` 取像失败。大约每 250,000 至 1,000,000 取像，及满足以下所有条件时，会发生上述故障。

- 应用程序运行在完全启动的四核 CPU 上
- 应用程序使用手动（软件）触发。
- 应用程序排序多个取像启动（在调用 `ccAcqFifo::completeAcq()` 之前多次调用 `ccAcqFifo:start()`）。

- 应用程序异步同时从 3 个或 4 个相机采集图像（不使用主从取像）。
- 所有相机都是高速相机，如索尼 XC-HR70 或 XC-HR90。

取像错误是暂时的，只影响一次取像，并且能自行恢复。此错误不影响任何使用外部触发器的取像；只影响使用多个队列启动的软件初始化取像。

解决方法

您可以使用以下任何一种解决方法防止该错误发生：

- 在单核或双核 CPU 上运行应用程序。
- 如果正使用的是四核 CPU，禁用一个核。
- 在调用 `ccAcqFifo::completeAcq()` 之前，请勿多次调用 `ccAcqFifo::start()`。
- 使用外部触发器采集图像。

显示

本部分介绍图像显示的问题。

错误绘制椭圆弧

当椭圆中心远超出显示区域时，错误绘制椭圆弧。使用比 `sizeof(short)` 更大的半径值会产生此效果。该问题由 Microsoft GDI 图形绘制代码中的局限引起，并且存在于所有支持的 Windows 平台（2000 和 XP）上。

使用 `ccDisplayConsole` 时显示缩放不一致

可以与 `ccDisplay::mag()` 一起使用的最大缩放倍数是 65,536。最小缩放取决于图像的大小；例如，对于 640x480 图像，最小值是 -60。在使用 `ccDisplayConsole` 进行缩放时，这可能导致不必要的行为。正如显示控制台窗口底部的状态栏所示，缩小（使用工具栏按钮进行缩放）将产生 1/2、1/4、1/8、1/16、1/32 和 1/60 的放大效果，但是放大将产生 1/30、1/15 的放大效果，依此类推。

母版不与 `ccUIManShape-Derived` 子版一起运行

在主从关系中，子版图形是由 `ccUIManShape` 衍生而来时，母版不如预期有起作用。如果拖动母版图形，则只有衍生自 `ccUIManShape` 的子版图形手柄会与其母版图形一起移动；子版图形本身不会移动。

母版不与 `ccUIManShape-Derived` 作为母版一起运行

一般来说，当主从关系中的母版图形由 `ccUIManShape` 衍生而来，子版图形由 `ccUIPointShapeBase` 衍生而来时，母版工作就能照预期运行。然而，`ccUIGenAnnulus`（`ccUIManShape` 衍生的图形）是个例外，不能作为母版正常工作。

如果传递给 `startLiveDisplay()` 的 FIFO 超出范围，则 `stopLiveDisplay()` 将挂起

在调用 `ccDisplay::stopLiveDisplay()` 之前，必须确保传递给 `ccDisplay::startLiveDisplay()` 的 FIFO 没有被销毁。例如，如果在一个函数中创建一个 FIFO，调用 `startLiveDisplay(fifo)` 在此 FIFO 中传递，然后在创建 FIFO 的函数返回后调用 `stopLiveDisplay()`，则 `stopLiveDisplay()` 方法将挂起而不返回。

不支持多个监视器

如果系统有多个监视器，CVL 不提供 API 允许您将显示控制台窗口定向到特定的监视器。

无法填充椭圆环空截面

`ccGraphicProps::penfill()` 函数不能处理 `ccEllipseAnnulusSection` 对象。

设置填充为 `ccUIGenAnnulus` 填充整个图形

将 `ccUIGenAnnulus` 的填充属性设置为 `true` 将会填充整个图形，而不仅仅是环形。

空白显示，索尼 DXC-390 和 8 位桌面

使用 MVS-8504、Sony DXC-390 相机、“Sony DXC-390 640x480 IntDrv CCF” 视频格式和 `ccAcqFifoPtrh` 设置测试用例来运行实时显示。当显示在 8 位 Windows 桌面时，观察到显示的图像为空白。

解决方法

使用此配置时，请使用 16 位或 32 位桌面颜色深度查看显示的图像。

驱动程序

本部分介绍 CVL 9.0 中驱动程序的已知问题。

固有的 PC 中断延迟可能导致在高速时错失取像

在测试期间，Cognex 观察到一些 PC 需要相对较长时间（5-6 毫秒）回应中断。此长时间延迟不会导致大多数 CVL 应用程序出现问题。但是，当取像率高于每 6 毫秒一次的取像率时，此长时间延迟可能会导致中断无法及时运行，从而导致错失取像。一些 PC 的长时间延迟也会导致其他需要及时中断响应的设备出现问题。

以下 PC 已知会有了该问题：

IBM Intellistation Z Pro: 型号 # MT-M 6221-22U (2.66 GHz Xeon 单 CPU)

IBM Intellistation M Pro: 型号 # MT-M 6219-42U (3.06 GHz P4 单 CPU)

该问题对感兴趣区域 (ROI) 的取像有一定影响。图像很小时，帧速率可能会比平时高，这可能会导致中断溢出。因此，当使用支持 ROI 控制和 MVS-8500 采样的相机时，可能会出现上述问题，MVS-8500 支持部分扫描（与 ROI 具有相同的最终结果）。

开发框架

本部分介绍 CVL 9.0 中 CVL 开发环境的已知问题。

不能同时使用新与旧的 iostream 库

CVL 使用 std 命名空间中更新的 iostream 库；因此，包括任何 CVL 标头也包括新的 iostream 标头。使用旧的 iostreams 库，或者使用旧的 iostreams 库的其他产品（如 MatLab），在当单个源文件试图包含两个库的标头文件时，可能会导致编译错误。该问题似乎源于 Windows ios 类。

这对于某些类来说是编译时间问题，对于其他类来说是潜在的运行时间问题，因为两个类中可用的枚举具有不同值。如果您的软件目前使用旧的 iostream 库，Cognex 建议更新到新的 iostream 库。如果您的软件需要同时使用旧的和新的 iostream 库，Cognex 建议始终首先包含新的 iostream 库。

当使用 stdafx.h 时（因此每个源文件都首先包含），可以在 stdafx.h 的顶部添加以下行：

```
#include <ch_cvl/defs.h>
#include <fstream>
#include <iostream>
```

第一个包含 defs.h 的是利用 defs.h 中的编译注解，以禁用“调试符号名太长”编译器警告 (C4786)。必须在包含生成警告的文件前执行此编译注解，否则将发生警告。您也可以直接使用此编译注解来代替 defs.h。

样本代码

本部分介绍 CVL 9.0 中样本代码的已知问题。

运行时 Pmalloc4 样本崩溃

当从 Windows 资源管理器，或命令提示符命令行与当前目录设置为可执行文件的位置一起运行时，与 CVL 一起提供的 Pmalloc4 样本应用程序（在 \$VISION_ROOT\sample\cvl 目录下）崩溃并出现“plate.idb 无法打开”错误。问题在于 plate.idb 文件的相对路径在安装过程中未正确设置。

解决方法

在 Visual Studio 中构建并运行样例，或移动 plate.idb 和 plate.dxf 文件到 \cvlproj\ 目录。

串行 I/O

本部分介绍 CVL 9.0 中串行 I/O 的已知问题。

串行 I/O: 读取操作忽略超时，直到收到第一个字节

ccSerialIO::read() 无限期阻塞，直到在串行端口上接收至少一个字节，即使规定了非无限超时。在串行端口接收到第一个字节后，read() 从该点开始正常运行。

图形模型

本部分介绍 CVL 9.0 中图形模型的已知问题。

尝试记录未约束的像素缓冲区可能导致崩溃

`ccShapeModel::record()` 方法需要约束提供像素缓冲区。但是，如果提供未约束像素缓冲区试图除以 0，它不会抛出并最终会导致应用程序崩溃。例如，以下调用将以崩溃结束。这是因为默认情况下，作为第二个参数提供的默认建构的 8 位 `ccPelBuffer` 未约束。

```
ccShapeModel::record(diagObj, ccPelBuffer<c_UInt8>(), shape,
gprops, gprops, gprops, cmT("Synthetic model"));
```

解决方法

始终提供具有非零宽度和高度的约束像素缓冲区作为 `ccShapeModel::record()` 方法的第二个参数。

卡尺工具

本部分介绍 CVL 9.0 中卡尺工具的已知问题。

卡尺：剪切启用时高精度模式识别额外边缘

当卡尺在 `eHighPrecision` 模式下运行时，会在仿射矩形设置为剪辑图像时发现额外边缘。例如，当投影图像（对应于原图像的反向读取）在结果中读取 200 214 198 197 194 192 190 185 183 111...，当对比度阈值设置为 5 时，像素值 214 将导致图像右侧出现额外边缘。如果把模式改为 `eBilinear`，投影图像读取 198 198 197 193 191 189 184 176 107...，较好地呈现了原始图像。

解决方法

高精度模式的本质是边缘环。解决方法是使用双线性模式。

卡尺：如果剪切，结果图形无法绘制

若您在 `drawMode` 参数中指定 `ccCaliperDefs::eDrawProjFilter` 至 `ccCaliperOneResult::draw()`，如果投影区域被输入图像剪切，图像（会显示出投影图案的图示，及过滤后的投影图像）则不会显示出来。为防止这种情况发生，需提供完全覆盖投影区域的输入图像。

卡尺：启用剪切时未绘制投影和滤波图像

不管投影模式如何，当启用剪切时，`ccCaliperResultSet::draw()` 方法无法绘制指定的投影图像和滤波图像。例如，以下代码将仅绘制标签和区域。

```
clpResultSet.draw(graphList, ccCaliperDefs::eDrawLabel
| ccCaliperDefs::eDrawProjFilter
| ccCaliperDefs::eDrawProjectionRegion
| ccCaliperDefs::eDrawStandard);
```

因此，追踪投影图像不便。

边缘检测工具

本部分介绍 CVL 9.0 中边缘检测工具的已知问题。

边缘：默认情况下 `ccDiagObject` 记录不执行深度复制

无论默认为深度复制还是使用 `diagFlags` 强制浅层复制，都可以在 `cfEdgeDetect()` 记录诊断对象后将其绘制到源图像、强度图像和角度图像中，并在诊断查看器中显示这些修改。

条形图工具

本部分介绍 CVL 9.0 中条形图工具的已知问题。

`cfPelHistogram()` 反应与标头文件注释不一致

<Ch_cv1/histo.h> 标头文件中的 `cfPelHistogram()` 函数的注释未准确表明，如果传入向量大小为 0，则函数调整其大小并将其归零。此函数未调整无元素的向量的大小或将其归零。

ID 工具

本部分介绍了 CVL 9.0 中 ID 工具的已知问题。

对于复合代码，在组合结果时无法确定 2D 组件的子类型

如果指定 `ccSymbologyParamsComposite::combineResults()` 为 `true`，则无法确定 2D 组件的子类型（CC-A 或 CC-B）。仅线性部分的子类型返回。

OCV 工具

本部分介绍了 CVL 9.0 中 OCV 工具的已知问题。

在混淆字符的错误验证之后，启用旋转不确定度会导致位置验证失败

OCV 工具在试图验证图像中字符“A”时表现出异常行为，当前密钥设置为“4”（反之亦然），字符旋转不确定度设置为大于零。允许旋转时，这些字符很容易混淆。

当设置旋转不确定度时，该工具通常验证 A-4 不匹配，虽然得分较低，但不匹配后的所有位置都验证失败。在三种不同的字体（Courier、Lucida Sans、Terminal）和不确定度范围（5 到 45 度）下观察到此反应。对于某些不确定的情况，除了不匹配外，Courier 字体可能会验证一个或两个位置，然后失败。该反应是一致的，无论何处发生不匹配（无论是字符串的开头还是结尾），也无论不匹配的数量有多少。

空格（密钥 32）在混淆重写时不可用

OCV 工具假定空白字符永远不会与实际字符混淆。因此，例如，该工具从不检查预期为“A”的字符是否为空白。然而，事实并非如此。该工具会检查预期的空白字符是否看起来像“A”（如果认为“A”易混淆）。

行布置边界框不包括行尾的通配符

OCV 行布置边界框不包括行尾的通配符。

未找到行时诊断显示最终位姿的红十字

完全没有找到行（也就是说，没有找到所有字符，得分为 0.0）时，“最终位姿的行布置”诊断将绘制行布置位姿的红十字。在客户端坐标中绘制红十字。

PatInspect

本部分介绍了 CVL 9.0 中 PatInspect 的已知问题。

PatInspect：错误报告超边的或缺失的边缘

PatInspect 边界差异模式可能会错误报告位于检查区域边缘附近的特征为缺失或超边。例如，当检测区域与训练区域重合时，所有的边界特征都应该是匹配类型。在 CVL 5.4 反向和非反向兼容模式中都观察到了此反应。

解决方法

确保矩形检查区域在您想要训练的模式周围提供至少 5 个像素的填充。

PatMax

本部分介绍了 CVL 9.0 中 PatMax 的已知问题。

PatFlex 忽略运行时参数中的标称 DOF 值

PatFlex 算法使用 `nominal()` setter 忽略 `cc_PMRunParams` 中设置的任何标称 DOF 值。

解决方法

若要在 PatFlex 中使用标称 DOF 值，请启用给定的 DOF 并将其高区域值和低区域值设置为所需的标称值。例如，代替：

```
runParams.zoneEnable(0);  
  
runParams.nominal(ccPMAlignDefs::kUniformScale, 2.0);  
  
使用  
  
runParams.zoneEnable(ccPMAlignDefs::kUniformScale);  
  
runParams.zone(ccPMAlignDefs::kUniformScale, 2.0, 2.0);
```

晶片预对准工具

本部分介绍了 CVL 9.0 中晶片预对准工具的已知问题。

在 NotchMax 模式下，平面长度计算不准确

在 NotchMax 模式下，平面长度计算使用晶片面积、内径、外径 来计算平面长度，可以使用 `ccWaferPreAlignResult::flatLength()` 查询。然而，每个晶片可能有 1 个、2 个甚至 3 个平面。这会影响面积的计算，进而影响内径和外径的计算。这可能导致返回的平面长度不准确。因此，平面晶片上不支持 NotchMax 模式。

局限性

局限性是硬件、操作系统软件、产品架构或其他外部约束强加到 CVL 上的表现方式。解决方法可能存在一些局限性。

安全性

本部分介绍了此版本中有关软件安全的局限性。

此版本不支持 ccSecurityInfo

此版本不支持 CVL 类引用中所述的 `ccSecurityInfo` API。

取像

本部分介绍了图像采集的一般局限性。

默认曝光用于频闪取像

所有相机的 20 μ s 默认曝光旨在频闪取像。如果使用环境照明与快速复位相机，如索尼 Xc-55，可能需要增加曝光时间超过 1 毫秒。

使用错误同步模型创建 Pulnix TM-6CN 静态视频格式

Pulnix TM-6CN 静态视频格式通过启用内部同步来实例化，这对该款相机来说不正确，而且会产生奇怪的图像。TM-6CN 视频格式的 CCF 版本正确设置同步模型。

解决方法

使用静态视频格式时，请在代码中将同步属性手动设置为 `cfSyncExternal1PLLComposite()`。

MVS-8500 取像

本部分介绍了 CVL 9.0 中 MVS-8500 取像的局限性。

取像亮度属性默认值

对于 8504 和 8501 图像采集卡，取像亮度属性默认值更改为 0.5。因此，使用此版本采集的图像可能与先前版本使用不同默认值（在创建取像 `fifo` 之后）采集的图像不同。

CFG-8704e 取像

本部分介绍了 CVL 9.0 中 CFG-8704e 图像采集卡的局限性。

不支持两个以上的 CFG-8704e 图像采集卡

请注意，不能在同一台计算机中使用两个以上的 CFG-8704e 图像采集卡。

显示

显示

本部分介绍了显示图像的局限性。

添加和删除许多 UI 图形需要很长时间

向显示屏中添加大量可操作图形，或者从显示屏中删除，一个循环操作一次可能需要很长时间。

解决方法

在添加或删除可操作图形之前调用 `ccDisplay::disableDrawing()`，然后在添加或删除所有图形之后调用 `ccDisplay::enableDrawing(true)`。

`ccUIGenAnnulus::move()` 正确应用

`ccUIGenAnnulus::move()` 方法移动 `ccUIGenAnnulus` 到 `cc2Vect` 提供的既定位置。该操作行为错误。应该根据 `cc2Vect` 所提供的 `x` 和 `y` 值来调整图形位置。

解决方法

使用 `ccUIGenAnnulus::frontKid()`，检索 UI 通用环的前子对象的位置，将偏移量添加到结果中，然后调用 `ccUIGenAnnulus::pos(p)` 传递调整后的位置作为参数。

8 位桌面的高精度模式中的插值缩放

在 8 位桌面上，通过调用 `ccUITablet::interpolation(eHighPrecision)` 将插值缩放设置为高精度模式可能会导致显示灰度范围之外的颜色。

解决方法

使用非 8 位显示和/或双线性而非高精度插值。

W2K 电源管理干扰直播视频

如果正在使用 Windows 2000，请禁用电源管理功能，如 CVL 用户指南中关于优化系统配置以获得最佳性能的文章所述。特别是，确保系统配置为永久关闭视频监视器。如果电源管理功能配置为关闭视频监视器，CVL 应用程序可能出现问题，试图显示直播视频图形。

CVL 5.3 中存档的椭圆形无法在较新的 CVL 版本中读取

在 CVL 5.4 中更改了椭圆几何类 (`ccEllipseGeom`) 的内部结构。针对 CVL 5.4 或更高版本构建的应用程序，如果试图显示包含在 CVL 5.3 或更高版本中存档的 `ccEllipse`、`ccEllipseArc`、`ccEllipseAnnulus` 或 `ccEllipseAnnulusSection` 对象 `ccUISketch` 列表，则应用程序可能挂起。

静态图形在发行和调试模式中呈现不同

静态图形在发行和调试模式中呈现不同根据图形的几何形状，在两种模式下呈现的像素之间可能会有细微差异。请勿依赖于在调试和发行构建之间呈现的完全相同像素。

开发框架

开发框架

本部分介绍了 CVL 开发环境的局限性。

“`treat wchar_t as builtin type`” 的 Visual Studio 项目设置

Visual Studio 在 `treat wchar_t as builtin type` 默认使用 `true`。Cognex 使用 Visual Studio 默认设置，因此您的项目也必须使用 `treat wchar_t as builtin type == true`。

安装

本部分介绍了在安装 CVL 时应注意的局限性。

单击命令提示符窗口暂停卸载

卸载 CVL 时，如果为命令提示符窗口启用 QuickEdit 模式，并且在卸载过程中点击打开的命令提示符窗口，则光标进入选择模式，卸载似乎已停止。在命令提示符窗口右键单击，以退出选择模式并继续卸载。

离散 I/O

本部分介绍了 CVL 中离散 I/O 的局限性。

并行 I/O: 外部 I/O 模块输出极性不一致

通用并行 I/O 卡 (P/N 800-5726-3) 的外部配置与外部 I/O 模块 (P/N 800-5712-2) 一起使用，为并行 I/O 线路提供光学隔离连接点。当与 MVS-8120 一起使用时，外部 I/O 模块上的输出行 4 到 7 的极性反转，并与输出行 0 到 3 相比，

与其他 Cognex 硬件上的光学隔离输出相比。

CVL 类引用正确表明 `ccOutputLine::set(true)` 成员函数设置光学隔离输出的无电流流动状态。对于外部 I/O 模块上的输出行 0 到 3, 以及其他 Cognex 电路板, 均正确无误。但是, 对于与 MVS-8120 一起使用的外部 I/O 模块的输出行 4 到 7, `set(true)` 和 `set(false)` 的含义相反。

实际脉冲宽度可小于规定值

在某些情况下, 实际脉冲信号可能比提供给 `ccOutputLine::pulse()` 的宽度参数短。例如, 如果创建如下 500 毫秒脉冲:

```
oline.pulse(true, 0.5, false); // 低到高转换,
                                // 0.5 秒宽度, 无阻塞
```

产生的脉冲实际上可能只有 490 毫秒。

这种反应是操作系统计时器的已知限制, 并且影响所有支持 `pulse()` 功能的 Cognex 硬件平台。此问题在同时调用多个并行输出行上的 `pulse()` 时更常见。

解决方法

如果应用程序需要绝对最小预期脉冲持续时间, 则为 `ccOutputLine::pulse()` 函数提供一个宽度值, 该值要比实际所需的脉冲宽度长 1 毫秒。这将把实际脉冲宽度提升到与多媒体计时器的粒度相匹配的下一个间隔。

此外, 使用 Microsoft Win32 API 调用 `timeBeginPeriod(1)` 将多媒体计时器周期减少到 1 ms 可提高脉冲精度。但是, 如果使用此方法, 请注意修改计时器将影响整个系统的性能, 因为这会导致计时器中断, 从而需要更频繁处理。

要使用 `timeBeginPeriod()`, 必须包含 Win32 `mmsystem.h` 标头文件和代码中 `winmm.lib` 库。在 Visual C++ 6.0 中, 标头文件位于 `Vc98\include` 目录下; 在 Visual C++ .NET (7.0) 中, 标头文件位于 `VC7\PlatformSDK\include` 目录下。标头文件也可以在平台 SDK 和 DDK 包含文件中找到。

例如, 可以添加以下代码到应用程序, 调整系统计时器的准确性:

```
#include <windows.h> // 添加此代码以避免编译错误
#include <mmsystem.h>
#pragma comment(lib, "winmm.lib")
timeBeginPeriod(1);
//设置脉冲宽度, 在 CVL 中执行 I/O 操作
timeEndPeriod(1); // 在应用程序结束前调用以返回
                //将多媒体定时器恢复到原始状态
```

有关修改系统计时器的相关说明, 请参阅 CVL 用户指南中显示图像章节的显示实时图像部分。

实用程序

本部分介绍了适用于 CVL 9.0 配备的实用程序的局限性和限制。

CVL 命令行实用程序必须保存在 CVL bin 目录中

诸如 Cogtool 之类的 CVL 命令行实用程序通常只在 Cognex 技术支持的指导下运行。自此版本开始, 这些实用程序的可执行文件必须保存在包含 CVL 提供的 DLLs 的目录中。这意味着实用程序的可执行文件 (如 `cogtool.exe`) 必须保存在 `%VISION_ROOT%\bin\win32\cvl` 中才能产生正确结果。如果将实用程序的可执行文件复制到另一个目录并运行, 该工具会运行, 但会错误报告系统中无 Cognex 电路板, 即使 Cognex bin 目录位于该路径中。

这与 CVL 6.2 反应不同, 您可以将实用程序的可执行文件复制到另一个目录并运行。

条码

本部分介绍了 CVL 9.0 中条码工具的局限性。

decode() 函数中添加的参数破坏了反向兼容性

在 CVL 5.5.0 和 5.5.1 之间，默认值为 true 的布尔参数 autoRetry 被添加到 `ccAcuBarcodeTool` 类的几个虚拟函数的参数列表中，尤其是 `decode()` 函数。由于这改变了函数签名，如果使用 CVL 5.5.0 中的旧签名重写这些函数，您不仅无法在 CVL 5.5.1 及更高版本中重写这些函数，而且重写也不再被调用。编译器错误不会警告函数签名已更改。将 CVL 应用程序从 CVL 5.5.0 升级到 5.5.1 或更高版本时，请确保重写的 `ccAcuBarcodeTool::decode()` 函数与最新函数签名匹配。

斑点

本部分介绍了 CVL 9.0 中斑点工具的局限性。

斑点：通过引用泄漏内存保存场景描述

如果通过引用并从存档中重新加载 `ccBlobSceneDescription` 对象进行保存，应用程序会泄漏内存。可以通过指示而非引用保存 `ccBlobSceneDescription` 对象来防止这种情况发生。

标定

本部分介绍了 CVL 9.0 中标定工具的局限性。

棋盘格最小平铺尺寸

在进行特征匹配标定时，如果平铺尺寸小于 15x15 像素，特征检测器就无法可靠工作。对于大多数应用程序，检查器尺寸应大于 15x15，以避免在范围限制下操作。

卡尺

本部分介绍了 CVL 9.0 中卡尺工具的局限性。

滤波剪切投影的像素范围不正确

滤波剪切投影中的像素值范围不正确。滤波图像范围为 -255 到 +255，同时应从 $-255*n$ 到 $+255*n$ ，其中 n 为投影中样本的个数。为获得正确范围，重新调整（乘以 n ）滤波图像。

高斯采样

本部分介绍了 CVL 9.0 中高斯采样工具的局限性。

cfGaussSample() 会产生错误结果

使用 16 位目标像素缓冲夹的 `cfGaussSample()` 超载导致数值变为 32,767。因此，任何通常会产生大于 32,767 的结果数值（例如，输入的像素值大于 128 的刻度设置为 255）都会产生错误结果。

解决方法

设置刻度足够低，使所有结果小于 32,767。

灰度形态学

本部分介绍了 CVL 9.0 中灰度形态学工具的局限性。

除 3x3 内核外配置的 NxM 形态学并没有完全进行多核优化

除 3x3 内核外配置的 $N \times M$ 形态学可变尺寸内核图像工具并没有完全进行多核优化。

ID

本部分介绍了 CVL 9.0 中 ID 工具的局限性。

仅 RSS 组合符号支持 EAN.UCC 复合码

仅此版本中支持的 EAN.UCC 组合符号使用 RSS 复合码。

acuRead

本部分介绍了 CVL 9.0中 acuRead 工具的局限性。

用户定义的字体中缺少结束字符

在用户定义的字体中，结束字符可能会缺失。这均适用于固定和可变长度模式。在固定长度模式下，最后一个字符变成一个空格。

解决方法

扩大 ROI，允许开始和结束时有更多空白像素。

OCV

本部分介绍了 CVL 9.0中 OCV 工具的局限性。

尽管有空格还是错误验证了字符

如果行布置中的位置是用当前密钥之间的空白字符定义的，并且如果该位置在运行时图像中并非空白，并且与任何其他当前密钥不匹配，则 OCV 工具可能会错误验证或混淆该位置。当前空白密钥总是返回一个非零分数，不管在运行时搜索位置实际的内容。返回空白密钥的分数通常较低，但对于“-”或“”之类的窄字符，分数可能超过接受或混淆阈值，因此不会验证失败。

以下列出了一些空白字符分数的例子：

字符	得分
-	0.423
.	0.422
,	0.382
:	0.312
'	0.276

不管是否为这些位置定义了多个或单个当前密钥，都会发生这种情况，尽管已观察到一个实例，其中使用空白密钥加上另一个当前密钥比仅使用空白密钥时得分更低。该问题与 OCV 工具的设计有关，其中空白字符和“正常”字符有所不同。OCV 无法检查空白字符是否与任何内容混淆，仅查看空白分数来确定空白字符的状态。因此，很容易忽略小字符。

PatMax

本部分介绍了 CVL 9.0中 PatMax 工具的局限性。

合成对齐模型训练问题

使用训练时间转换将边界框投射到图像坐标时，使用边界框明显大于 2Kx2K 的模型训练 PatMax 会导致访问冲突或应用程序在 train() 中挂起。

已关闭漏洞

此版本包含先前版本中客户报告的问题修复。

亮度和对比值产生误差

采集图像时，亮度和对比值在映射到非等间隔相机的无效等价设置时不再产生误差

此版本提供的文档

CVL 安装包括以下可用标题**开始**→**Cognex**→**CVL**→**文档**：

- CVL 用户指南

CVL 编程概述，包括使用样本代码，取像和显示图像和图形，以及介绍视觉工具和概念。

- CVL 视觉工具指南

介绍每个 CVL 视觉工具的功能，每次确认的参数及其产生的结果。

- CVL 类引用

CVL C++ 类库的按类引用

- CFG-8724 硬件参考

详细介绍了 CFG-8724 图像采集卡，这是一种兼容 PCI Express x4 的 GigE 视觉图像采集设备。

未来版本说明

请注意，未来 CVL 版本会取消支持以下 API：

- 与特定二维条码相关的 `ccAcuSymbol*` 类别：
 - `ccAcuSymbolDataMatrixDefs`
 - `ccAcuSymbolDataMatrixLearnParams`
 - `ccAcuSymbolDataMatrixTool`
 - `ccAcuSymbolDefs`
 - `ccAcuSymbolFinderParams`
 - `ccAcuSymbolLearnParams`
 - `ccAcuSymbolQRCodeDefs`
 - `ccAcuSymbolQRCodeLearnParams`
 - `ccAcuSymbolQRCodeTool`
- 与基本 OCR 读取和调节功能相关的 `ccAcuRead*` 类别：
 - `ccAcuRead`
 - `ccAcuReadDefs`
 - `ccAcuReadFont`
 - `ccAcuReadResult`
 - `ccAcuReadResultSet`
 - `ccAcuReadRunParams`
 - `ccAcuReadTuneParams`
- 与一维条码工具相关的 `ccAcuBarCode*` 类别：
 - `ccAcuBarCodeCalibrationResult`
 - `ccAcuBarCodeDefs`
 - `ccAcuBarCodeResult`
 - `ccAcuBarCodeRunParams`
 - `ccAcuBarCodeTool`
 - `ccAcuBarCodeTuneParams`

