

CVL 9.0 릴리스 정보

2019 11월 14
Revision: 9.0.0.1

법적 고지

본 안내서에서 설명하는 소프트웨어는 라이선스 계약에 따라 제공되며, 그러한 라이선스의 조건과 이 페이지에 나오는 저작권 고지에 따라서만 사용 또는 복사할 수 있습니다. 라이선스 소지자 이외의 사람에게 소프트웨어 및 본 안내서 또는 그 본문 중 어느 것도 제공 또는 사용을 허락할 수 없습니다. 이 소프트웨어의 소유권은 Cognex Corporation 또는 그 라이선서에게 있습니다. Cognex Corporation은 Cognex Corporation이 제공하지 않은 장비에 있는 소프트웨어의 사용 또는 그 신뢰성에 대해 책임을 지지 않습니다. Cognex Corporation은 설명한 소프트웨어, 그 소프트웨어의 상품성, 비침해성 또는 특정 사용에 대한 적합성 등과 관련하여 어떠한 명시적 또는 암시적 보증도 하지 않습니다.

본 문서에 포함된 정보는 사전 통보 없이 변경될 수 있으며 Cognex Corporation에서 책임지는 것으로 해석되어서는 안 됩니다. Cognex Corporation은 본 안내서 또는 관련 소프트웨어에 있을 수 있는 모든 오류에 대해 책임지지 않습니다.

이 문서의 예에서 사용된 회사, 이름 및 자료는 별도의 언급이 없는 경우 허구적인 것입니다. 본 안내서의 어떠한 부분도 Cognex Corporation의 서면 동의 없이 어떤 목적으로든 전자 또는 기계적 방법 등 어떠한 형태나 수단으로 재생산 또는 전달될 수 없으며 다른 기타 매체 또는 언어로 양도될 수 없습니다.

Copyright © 2019. Cognex Corporation. All Rights Reserved.

Cognex가 제공하는 하드웨어 및 소프트웨어는 Cognex 웹 사이트(cognex.com/patents)에 나와 있는 미국 및 해외 특허법을 비롯하여 출원 중인 미국 및 해외 특허법 중 하나 이상의 적용을 받을 수 있습니다.

다음 항목은 Cognex Corporation의 등록 상표입니다.

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

다음 항목은 Cognex Corporation의 상표입니다.

Cognex 로고, 1DMax, 3D-Locate, 3DMax, BGAll, CheckPoint, Cognex VSoC, CVC-1000, FFD, iLearn, In-Sight(십자선이 있는 디자인 휘장), In-Sight 2000, InspectEdge, Inspection Designer, MVS, NotchMax, OCRMax, PatMax RedLine, ProofRead, SmartSync, ProfilePlus, SmartDisplay, SmartSystem, SMD4, VisiFlex, Xpand

Portions copyright © Microsoft Corporation. All rights reserved.

Portions copyright © MadCap Software, Inc. All rights reserved.

본 안내서에서 언급된 기타 제품 및 회사 상표는 해당 소유자의 상표입니다.

CVL 9.0 릴리스 정보

CVL 버전 9.0에는 CVL 8.0 이전 릴리스 이후로 변경된 기능과 새로운 기능이 포함되어 있습니다.

시스템 요구사항

CVL을 설치하고 사용하려면 PC가 최소 요구사항을 충족해야 합니다.

PC 요구사항

사용자의 PC는 다음과 같은 하드웨어 최소 요구사항을 충족해야 합니다.

- 1GHz 이상의 Intel Pentium(또는 호환) CPU와 MMX/SSE2 명령 세트 지원.
- PCI 버스 Cognex 프레임 그래버 또는 GigE 카메라 어댑터를 사용하는 경우 열린 PCI 슬롯 하나
- 직접 연결 카메라를 위한 포트 하나(GigE Vision의 경우 Gigabit-Ethernet)
- PCI, PCI Express(권장) 또는 AGP 비디오 어댑터
 - CVL에는 최소 8MB의 비디오 카드 메모리가 필요하며, 이를 사용할 경우 기능이 제한됩니다.
 - 마더보드와 통합된 칩셋 기반의 비디오 시스템을 통해 마더보드 슬롯에 꽂을 수 있는 독립형 비디오 카드를 사용하는 것이 좋습니다.
 - Cognex는 AGP 또는 PCI Express x16 독립형 비디오 카드와 최소 16MB의 메모리를 권장합니다.
 - 비디오 카드는 DirectX 8.1 이상을 지원해야 합니다.

그 밖의 Cognex 권장 사항은 다음과 같습니다.

- CVL 응용 프로그램이 디스크 페이지를 사용할 필요가 없도록 PC의 RAM이 충분해야 합니다.
- 플러그 앤 플레이 방식이 아닌 운영 체제를 지원하도록 PC의 BIOS를 설정하십시오.
- Windows 데스크탑의 색상 모드를 8비트, 16비트 또는 32비트로 설정하십시오. 컬러 영상을 표시하려면 데스크탑이 16비트 또는 32비트 모드여야 합니다.
- Cognex는 IDE(ATA) 하드 디스크용 PIO 드라이버가 아닌 버스 마스터링 방식의 DMA 디스크 드라이버가 설치된 PC를 권장합니다. 마더보드 업체에 문의하여 버스 마스터링 방식의 IDE 드라이버를 제공하는지 확인하십시오.
- PC의 RAM이 64MB 이상인 경우, 아래의 단계에 따라 커널 및 드라이버의 페이지징을 비활성화하여 메모리에서 더 빨리 액세스할 수 있도록 하십시오.
 - 현재의 레지스트리 설정을 백업합니다.
 - 레지스트리 편집기를 열고
*HKLM/SYSTEM/CurrentControlSet/Control/Session Manager/Memory Management*로 이동합니다.
 - *DisablePagingExecutive* 키를 1로 설정하여 커널 및 드라이버 페이지징을 비활성화합니다.
 - PC를 재부팅합니다.

운영 체제

CVL 9.0은(는) 다음과 같은 Windows 64비트 운영 체제의 싱글 프로세서 또는 멀티 프로세서 시스템에서 기본 언어(영어, 일본어, 독일어, 한국어, 중국어(간체))를 사용한 개발 및 배포를 지원합니다.

- Windows 7 Premium
- Windows 7 Professional

- Windows 7 Ultimate(64비트)
- Windows 10
- Windows 10 IoT Enterprise

Windows 7 지원은 이번 CVL 릴리스를 마지막으로 지원되지 않을 수 있으니 주의하십시오.

지원되는 Microsoft Visual Studio 개발 환경

CVL 9.0에서는 다음과 같은 개발 환경을 이용한 Vision 응용 프로그램의 개발 및 배포를 지원합니다.

개발 환경	컴파일러 버전	서비스 팩
Microsoft Visual Studio 2015	Visual C++ v14.0	업데이트 3
Microsoft Visual Studio 2017	Visual C++ v14.16	

CVL과 함께 Microsoft Visual Studio 2015 라이브러리 및 Microsoft Visual Studio 2017 CVL C++ 라이브러리 둘 다 제공됩니다.

CVL 9.0은(는) 관리되지 않는 C++ SDK이며, 관리되지 않는 응용 프로그램에서 Microsoft Visual Studio로 구축된 응용 프로그램만 지원합니다. CVL 9.0은(는) 관리되는 .NET(/clr 컴파일러 스위치 사용)으로 구축된 응용 프로그램(혼합 모드)을 지원하지 않습니다.

지원되는 촬상 플랫폼

CVL 9.0에서 지원되는 촬상 플랫폼은 다음과 같습니다.

- Cognex MVS-8511
- Cognex MVS-8511e
- Cognex MVS-8514
- Cognex MVS-8514e
- GigE Vision 카메라
- CFG-8724 시리즈 프레임 그래버
- CVL 사용자 안내서에 설명된 영상 장치 인터페이스에 맞는다사 촬상 장치.

이번 릴리스의 새로운 기능

CVL 9.0은(는) 새로운 기능을 지원하는 GA(일반 사용) 릴리스입니다.

새로운 USB 보안 키

이 릴리스에서는 하드웨어 플랫폼에 새로운 Cognex USB 보안 키가 부착되어 있어야 올바른 소프트웨어 라이선스로 간주합니다. 이전 릴리스에서 지원되던 어떤 보안 옵션도 이 CVL 릴리스에 사용할 수 없습니다. 단, 기존의 Cognex 프레임 그래버는 이전 릴리스와 마찬가지로 해당하는 영상 활상 기능을 수행합니다.

Cognex는 응용 프로그램 개발은 물론 배포를 위한 보안 키를 생산 환경의 응용 프로그램에 사용할 수 있도록 제한된 시간 동안 지원하고 있습니다.

자세한 내용은 Cognex 판매 담당자에게 문의하십시오.

Visual Studio 2017

이 릴리스에는 Microsoft Visual Studio 2017(v. 15.10)을 사용한 응용 프로그램 개발 및 컴파일 지원이 추가되었습니다.

LineMax 도구

이 릴리스에는 영상에서 가장자리점 후보를 모두 찾은 다음 사용자가 지정한 기준에 따라 최적의 선 세그먼트를 맞춰 주는 LineMax 도구 지원이 추가되었습니다. LineMax 도구는 8비트 및 16비트 영상을 지원합니다.

이 LineMax 도구는 개별 가장자리점을 찾으려면 캘리퍼를 정밀하게 배치해야 하는 맞춤 도구보다 강력합니다. Cognex는 신규 응용 프로그램에 LineMax 도구를 사용할 것을 권장하며, 기존 응용 프로그램에는 맞춤 도구를 계속해서 지원할 것입니다.

이 릴리스에 포함된 CVL 클래스 참조서에는 LineMax 도구용 API가 설명되어 있지 않습니다. 해당하는 헤더 파일과 소스 코드 설명은 `%VISION_ROOT%\Wdefs\Wch_cvl`에서 찾아볼 수 있습니다.

LineMax 도구에 대한 자세한 내용은 CVL 비전 도구 안내서를 참조하십시오.

CFG-8724 프레임 그래버

이 릴리스는 통합 보안 라이선스로 CFG-8724 4포트 GigE Vision 프레임 그래버를 지원합니다. CFG-8724에서는 PC 하나에 GigE Vision 카메라를 네 대까지 연결할 수 있습니다. 또한 동일한 PC에 CFG-8724 프레임 그래버를 여러 개 설치할 수 있습니다. 또한 CFG-8724는 Cognex 소프트웨어 라이선스를 저장하여 Cognex 소프트웨어의 적절한 사용 권한을 유지할 수도 있습니다. 단, CVL은 CFG-8724 하나에서만 소프트웨어 라이선스를 읽어 오니 주의하십시오. 보안 및 버전 정보에 대한 자세한 내용은 CVL 사용자 안내서를 참조하십시오.

Cognex는 앞으로 단종될 CFG-8174e의 직접 대체 품목으로 CFG-8724를 제공하고 있습니다.

CFG-8724를 설치한 이후에 Cognex Vision 소프트웨어를 설치하는 경우, 소프트웨어 설치를 마친 뒤 컴퓨터의 전원을 꺼야 합니다. 웬 재부팅만으로는 CFG-8724가 올바르게 작동하지 않을 수 있습니다.

16비트 영상 표시 지원

이 릴리스에는 `ccPelBuffer<c_UInt8>` 유형의 8비트 영상에 사용하는 것과 동일한 API를 사용하여 `ccPelBuffer<c_UInt16>` 유형의 16비트 영상을 표시할 수 있는 기능이 추가되었습니다. CVL은 16비트 범위의 그레이스케일 픽셀 값을 자동으로 8비트 범위로 매핑하여 표시해 줍니다.

회색도 값의 범위가 넓은 16비트 영상은 화면에서도 그 범위가 유지되지만 좁은 범위의 16비트 영상은 디스플레이에서 해당 범위가 확장 표시되니 주의하십시오.

예를 들어, 16비트 영상에 어두운 회색도 값과 밝은 회색도 값의 픽셀이 포함되어 있다면 CVL은 8비트 범위의 어두운 값과 밝은 값으로 그 영상을 표시합니다. 그러나 16비트 영상에 어두운 회색도 값만 포함되어 있다면 8비트 디스플레이에는 높은 쪽 범위의 픽셀 값이 밝은 회색도 값으로 표시됩니다.

CVL 영상 표시 API에 대한 내용은 헤더 파일 `VISION_ROOT\defs\ch_cv1\display.h`를 참조하십시오.

CNLSearch 모델 크기

이 릴리스에서는 CNLSearch 도구에 대해 새로운 4x4 모델 크기를 지원합니다. 이전 CVL 릴리스에서 지원하던 CNLSearch 모델의 최소 크기는 8x8입니다.

더 많은 비전 도구에 16비트 영상 지원

이 릴리스에는 다음과 같은 비전 도구에 대한 16비트 영상 지원이 추가되었습니다.

- 캘리퍼
- 영상 비틀림 도구
- 가우시안 샘플링의 필터링 마스크
- PeISet 도구의 마스크

호환성 정보

Cognex 하드웨어 및 CVL 9.0 소프트웨어의 사용과 관련된 문제는 다음 섹션을 참조하십시오.

큰 영상 호환성 문제

이 섹션에서는 큰 영상 지원과 관련된 호환성 문제를 설명합니다.

아핀 샘플링 도구 인터페이스 변경

CVL 6.7 CR1 및 이전 릴리스에서는 아핀 샘플링 도구를 위해 X 및 Y 방향의 샘플 수를 지정하는 데 사용하던 구성요소 (**ccAffineSamplingParams::xNumSamples** 및 **ccAffineSamplingParams::yNumSamples**)가 **c_Int16** 유형이었습니다.

그러나 큰 영상의 샘플링을 지원하기 위해 CVL 6.7 CR3 릴리스부터는 샘플 수를 **c_Int32** 값으로 저장하게 되었습니다.

그리고 **ccAffineSamplingParams** 의 설정기 함수도 **c_Int32** 값을 취하도록 변경되었습니다. C++ 컴파일러가 16비트 값을 32비트 값으로 올려 주기 때문에 이렇게 해도 기존 코드와의 호환성 문제는 발생하지 않습니다. 그러나 가져오기 함수는 기존 코드와의 호환성 문제 때문에 변경할 수 없었습니다. CVL 6.7 CR3에서는 **ccAffineSamplingParams** 에 다음 두 구성요소를 추가했습니다.

- **ccAffineSamplingParams::xNumSamples32()** 는 X 방향의 샘플 수를 반환합니다.
- **ccAffineSamplingParams::yNumSamples32()** 는 Y 방향의 샘플 수를 반환합니다.

16비트 값을 반환하는 기존의 가져오기 함수는 사용이 중단되었으며, 샘플 수가 32768개 이상인 경우 이 함수는 오류를 반환합니다.

큰 영상에서 일부 아핀 샘플링 모드가 지원되지 않음

X 차원 또는 Y 차원이 32768픽셀 이상인 입력 또는 출력 영상의 경우, 아핀 샘플링 도구는 다음과 같은 샘플링 모드만 지원합니다.

- **ccAffineSamplingParams::eNone**
- **ccAffineSamplingParams::eBilinearAccurate**

나머지 모드(**eBilinearApprox**, **eHighPrecision** 및 **eBilinear**)는 지원되지 않습니다. 지원되지 않는 모드를 지정하고 X 차원 또는 Y 차원이 32768픽셀 이상인 입력 또는 출력 영상을 사용하면 이 도구는 오류를 반환합니다.

큰 영상을 지원하기 위해 가장자리 도구 인터페이스 변경

CVL 6.7 CR1 및 이전 릴리스의 가장자리 도구는 16비트 값을 사용하여 하위 픽셀 가장자리(edgelet)의 전체 픽셀 위치를 저장했습니다. 다시 말해, 전체 픽셀 위치가 -32768~+32767 범위를 벗어나는 edgelet은 표현할 수가 없었습니다.

그러나 CVL 6.7 CR3부터는 가장자리 도구에서 edgelet의 저장 방법을 지정할 수 있게 되었습니다.

- 만일 **ccEdgeletDefs::eRequestContiguous** 를 요청한다면 edgelet은 인접한 블록 하나에 저장됩니다. 전체 픽셀 위치에서 영상의 X 또는 Y 좌표가 -32768~+32767 범위를 벗어나는 edgelet이 없다면 edgelet은 **ccEdgelet** 벡터에 저장되고 **ccEdgeletSet::edges()** 를 통해 액세스하게 됩니다. 전체 픽셀 위치에서 영상의 X 또는 Y 좌표가 -32768~+32767 범위를 벗어나는 edgelet이 있다면 edgelet은 **ccEdgelet2** 벡터에 저장되고, **ccEdgeletSet::edges2()** 또는 **ccEdgeletIterator** 또는 **ccEdgeletIterator_const** 클래스를 통해 액세스하며, 이 클래스는 **ccEdgeletSet::begin()** 을 호출하여 구합니다.
- 만일 **ccEdgeletDefs::eRequestEdges2** 를 요청한다면 edgelet은 **ccEdgelet2** 벡터에 저장되고, **ccEdgeletSet::edges2()** 또는 **ccEdgeletIterator** 또는 **ccEdgeletIterator_const** 클래스를 통해 액세스하며, 이 클래스는 **ccEdgeletSet::begin()** 을 호출하여 구합니다.

- 만일 `ccEdgeletDefs::eRequestEdgesAndOffsets` 를 요청한다면 `edgelet`은 하나 이상의 분리된 메모리 블록에 저장되며, 이때 각 블록에는 `ccEdgelet` 개체 어레이와 32비트 오프셋 지정자 쌍이 포함됩니다. `edgelet`에 액세스할 때는 `ccEdgeletIterator` 또는 `ccEdgeletIterator_const` 클래스를 사용합니다. 이러한 개체는 `ccEdgeletSet::begin()` 을 호출하여 구할 수 있습니다.

전체 픽셀 위치에서 영상의 X 또는 Y 좌표가 -32768~+32767 범위를 벗어나는 `edgelet`이 하나 이상인 경우에만 블록을 여러 개 사용합니다. 블록이 하나만 생성되었다면 `ccEdgeletSet::edges()` 및 `ccEdgeletIterator()` 메서드 둘 다 해당 세트의 `edgelet`에 액세스하는 데 사용할 수 있습니다.

저장 방법과 관계없이 `ccEdgeletIterator` 또는 `ccEdgeletIterator_const` 클래스를 `ccEdgeletSet::begin()` 을 호출하여 구한 다음 이를 통해 `edgelet`에 액세스할 수 있기 때문에 이 인터페이스를 사용하는 것이 좋습니다.

일부 CVL 및 VisionPro 버전에서 잘못된 CDB 파일 생성

CVL 6.8 CR1, CVL 6.8 CR2 또는 VisionPro 6.0과 함께 제공된 IDBEdit 프로그램을 사용하여 생성한 CDB 또는 IDB 파일은 CVL 및 VisionPro 초기 버전에 제공된 IDBEdit 프로그램으로 읽을 수 없습니다.

CVL 9.0(및 VisionPro 6.1 또는 6.2)의 IDBEdit 프로그램을 사용하여 잘못된 CDB 또는 IDB 파일을 복구할 수 있습니다. 새 버전의 IDBEdit 프로그램을 사용하여 해당 파일을 열고 **파일->다른 이름으로 저장...**을 선택한 다음, 다른 이름으로 저장하기만 하면 됩니다. 새로 저장한 이 파일은 모든 버전의 IDBEdit에서 읽을 수 있습니다.

큰 영상이 있는 CDB 파일이 이전 CVL 릴리스와 호환되지 않음

CVL 6.8 CR3으로 구축한 CVL 응용 프로그램을 사용하여 X 차원 또는 Y 차원의 크기가 32768픽셀을 초과하는 영상을 저장하는 경우, CVL 릴리스 6.7 CR1 이상에서 구축한 CVL 응용 프로그램에 그 영상을 로드하려고 하면 1x1픽셀의 이미지가 생성됩니다.

장치 드라이버 역호환성

CVL 9.0에서 제공하는 장치 드라이버는 아래 표에 나열된 버전과 역호환이 가능합니다. 여기서 말하는 역호환성은 지정된 구형 CVL 버전이 이번 릴리스의 새 드라이버로도 계속 작동한다는 의미입니다.

프레임 그래버 또는 카메라	8.0 드라이버와 호환되는 CVL 최초 버전 (Windows 7)	8.0 드라이버와 호환되는 CVL 최초 버전 (Windows 8.1 및 10)
MVS-8511 MVS-8514 MVS-8511e MVS-8514e	CVL 7.0(64비트)	CVL 8.0
GigE Vision 고성능 드라이버	CVL 7.1 CR2(64비트)	CVL 8.0

MVS-8510 시리즈 프레임 그래버의 호환성 문제

이 섹션에서는 MVS-8510 및 MVS-8510e 프레임 그래버 사용에 영향을 미치는 호환성 문제에 대해 설명합니다.

MVS-8510 시리즈 프레임 그래버의 대비 및 밝기

대비 및 밝기는 CVL에서 `ccContrastBrightnessProp` 클래스 함수로 설정합니다. 자주 사용하는 대비 범위인 0.05~0.95 및 밝기 설정 0.20~0.80 범위에서는 활성화한 영상의 회색도 값이 사용 중인 보드에 따라 약간 달라집니다.

- 대비 및 밝기 설정이 동일할 때 MVS-8511, MVS-8511e, MVS-8514 및 MVS-8514e 프레임 그래버는 (지원되지 않는) MVS-8501 및 MVS-8504 프레임 그래버에서 동일한 입력 신호를 사용하여 얻은 픽셀 값보다 회색도가 최대 10단계 높거나 6단계 낮은 회색도 값을 출력합니다. 일반적인 입력 범위에서 회색도 값의 중앙값 차이는 약 0입니다.
- 대비 및 밝기 설정이 동일할 때 MVS-8511, MVS-8511e, MVS-8514 및 MVS-8514e 프레임 그래버는 (지원되지 않는) MVS-8500Le 및 MVS-8504e 프레임 그래버에서 동일한 입력 신호를 사용하여 얻은 픽셀 값보다 회색도가 최대 4단계 높거나 12단계 낮은 회색도 값을 출력합니다. 일반적인 입력 범위에서 회색도 값의 중앙값 차이는 약 -2입니다 (새 프레임 그래버에서 얻은 값이 낮음).

밝기 및 대비 설정을 조정하여 특정한 입력값 범위에서 회색도 값 차이를 최소화할 수 있습니다.

소프트웨어 주소 및 300-0406 케이블

Cognex 300-0406 케이블은 Sony DXC-390 컬러 카메라와 흑백 아날로그 카메라 한 대를 MVS-8514 및 MVS-8514e에 연결하기 위한 것입니다.

이 케이블을 사용하는 경우, 컬러 카메라에 연결되는 R, G, B 분기 케이블은 카메라 위치 0, 1, 2를 사용합니다. 다시 말해, 네 번째 분기 케이블에 연결되는 흑백 카메라의 소프트웨어 주소는 3이 됩니다. 300-0406 케이블을 사용하는 경우, 흑백 카메라 제어를 위해 CVL 코드에 이 주소를 사용하십시오.

TTL 전용 케이블 옵션을 사용하는 경우 스트로브 극성 반전

모든 MVS-8510 시리즈 프레임 그래버에서 병렬 I/O 케이블 옵션 셋 중 두 개는 스트로브 라인이 광절연되어 있습니다. 그러나 TTL 패스스루 케이블 옵션(케이블 300-0390 및 TTL 연결 모듈, 800-5818-1)을 사용하는 경우에는 스트로브 라인 네 개의 극성이 광절연된 동일 라인과 다르게 반전됩니다.

그러므로 전체 광절연된(300-0389) 케이블 옵션이나 절반씩 광절연된(300-0399) 케이블 옵션을 사용하는 경우에는 `ccStrobeProp::strobeHigh()`에 대한 설명(CVL 클래스 참조서 참조)이 맞으려면

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

로 작성해야 합니다. 극성이 참이면 활성 펄스는 높음으로 설정됩니다. 거짓이면 활성 펄스는 낮음으로 설정됩니다.

그러나 TTL 패스스루 케이블 옵션(300-0390)에 스트로브를 사용할 때는 극성이 반전되어

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

가 됩니다. 극성이 참이면 활성 펄스는 낮음으로 설정됩니다. 거짓이면 활성 펄스는 높음으로 설정됩니다.

미해결 버그

이 섹션에는 이 릴리스의 미해결 버그가 나열되어 있습니다.

활상

이 섹션에서는 CVL 9.0의 영상 활상과 관련된 미해결 문제를 설명합니다.

RGB 팩 함수가 변형을 잘못 복사함

대상 픽셀 버퍼가 미리 할당되어 있고 소스의 픽셀 버퍼와 크기가 다른 경우, `cfRGBPack()` 함수가 변형을 올바르게 복사하지 못합니다.

해결 방법

다음 중 하나를 수행하십시오.

- `cfRGBPack()` 호출 후 수동으로 변형을 복사합니다.
- `cfRGBPack()` 으로 전달하는 대상 픽셀 버퍼가 바인딩되지 않았는지 확인합니다.

`ccLightProp::LightsOff()`가 작동하지 않습니다.

`ccLightProp::lightsOff()` 함수를 호출해도 효과가 없습니다. 다음 코드를 사용하여 이 버그를 해결할 수 있습니다.

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

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

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

MVS-8500 활상

이 섹션에서는 CVL 9.0의 MVS-8500 활상과 관련된 미해결 문제를 설명합니다.

8500 프레임 그래버를 사용하여 색상 정보가 포함된 비디오 신호에서 활상할 때 ExtSync 활상의 잠금이 해제될 수 있음

NTSC 또는 PAL 등 색상 정보가 포함된 비디오 신호를 8500 프레임 그래버에 입력하는 경우, 카메라에 잡힌 장면의 유형에 따라 ExtSync 활상이 실패할 수 있습니다. 이 문제는 ~3.58MHz(NTSC) 및 4.43MHz(PAL) 주파수의 캐리어로 인코딩된 색

상 정보와 컬러 버스트가 RS170 또는 CCIR 신호에 포함되어 있기 때문에 발생합니다. 이러한 색상 정보는 8500과 충돌하여 HSync 펄스를 올바르게 감지하지 못하게 하고, 이로 인해 잘못된 잠금이 발생할 수 있습니다.

해결 방법

해결하려면 비디오 신호에서 색상 정보를 제거한 다음 8500에 입력해야 합니다. 이를 쉽게 해결하는 방법은 컴포지트 비디오를 S-비디오(Y/C)로 바꿔 주는 변환기를 사용하는 것입니다. 변환된 Y 구성요소에는 색상 정보가 들어 있지 않습니다.

Sony DXC-390 카메라에서 어설션이 발생할 수 있음

외부 동기화 소스를 사용하는 경우, 촬영이 진행되는 동안 카메라 케이블이 분리되면 어설션이 발생할 수 있습니다.

대기열의 소프트웨어가 트리거되면 쿼드코어 CPU에서 MVS-8500e 촬영 오류

특정한 시스템 구성에서 MVS-8500e 프레임 그래버에 간헐적으로 isAbormal() 촬영 오류가 발생할 수 있습니다. 250,000~1,000,000회 촬영 중 약 1회 일어나는 이러한 오류는 다음 조건이 모두 충족될 때 발생합니다.

- 응용 프로그램이 모든 코어가 활성화된 쿼드코어 CPU에서 실행 중입니다.
- 응용 프로그램이 수동 (소프트웨어) 트리거를 사용합니다.
- 응용 프로그램이 여러 차례의 촬영 시작을 대기열에 입력합니다(ccAcqFifo:start())를 여러 번 호출한 다음 ccAcqFifo::completeAcq() 호출).
- 응용 프로그램이 마스터-슬레이브 촬영을 사용하지 않고 3대 또는 4대의 카메라에서 동시에 영상을 비동기적으로 촬영합니다.
- 모든 카메라가 Sony XC-HR70 또는 XC-HR90 같은 고속 카메라입니다.

이 촬영 오류는 일시적으로 발생하여 촬영 하나에만 영향을 주고 자체 복구됩니다. 외부 트리거를 사용하는 촬영에는 이 오류가 발생하지 않으며, 여러 차례의 대기열 시작을 통해 소프트웨어에서 시작되는 촬영에만 적용됩니다.

해결 방법

다음과 같은 방법을 사용하여 이 오류를 방지할 수 있습니다.

- 응용 프로그램을 싱글코어 또는 듀얼코어 CPU에서 실행하십시오.
- 쿼드코어 CPU를 사용하는 경우에는 코어 하나를 비활성화하십시오.
- ccAcqFifo::completeAcq()을 호출하기 전에 ccAcqFifo:start()를 한 번 이상 호출하지 마십시오.
- 외부 트리거를 사용하여 영상을 촬영하십시오.

디스플레이

이 섹션에서는 영상 표시와 관련된 문제를 설명합니다.

타원의 호가 잘못 그려짐

타원의 중심이 표시 영역 밖에 멀리 떨어져 있을 때 타원 호가 잘못 그려집니다. **Sizeof(short)**보다 큰 반경 값을 사용하면 이런 효과가 나타납니다. 이 문제는 Microsoft GDI 그래픽 드로잉 코드의 제한 때문에 발생하며, 지원되는 모든 Windows 플랫폼(2000 및 Xp)에 존재합니다.

ccDisplayConsole 사용 시 일관성 없는 줌 표현

ccDisplay::mag()에 사용할 수 있는 최대 줌은 65,536입니다. 최소 줌은 영상의 크기에 따라 달라집니다. 예를 들어, 640x480 영상의 최솟값은 -60입니다. 이로 인해 ccDisplayConsole을 사용하여 줌할 때 원치 않는 동작이 나타납니다. 디스플레이 콘솔 화면 하단의 상태 표시줄에 표시되듯이, 툴바 버튼을 사용하여 줌하면 축소 배율이 1/2, 1/4, 1/8, 1/16, 1/32, 1/60이고, 여기서 확대하면 배율이 1/30, 1/15 등으로 진행됩니다.

ccUIManShape에서 파생된 하위 항목에 대해 세대화가 작동하지 않음

ccUIManShape에서 파생된 상위-하위 관계의 하위 모양이 올바르게 세대화되지 않습니다. 상위 모양을 드래그하면 ccUIManShape에서 파생된 하위 모양의 핸들만 상위와 함께 움직이고, 하위 모양 자체는 이동되지 않습니다.

ccUIGenAnnulus를 상위로 하는 세대화가 작동하지 않음

일반적으로, ccUIManShape에서 파생된 상위 모양과 ccUIPointShapeBase에서 파생된 하위 모양이 상위-하위 관계를 이룰 때는 세대화가 올바르게 작동합니다. 그러나 ccUIGenAnnulus(ccUIManShape의 파생 모양)는 예외이며 상위 역할을 제

대로 하지 못합니다.

startLiveDisplay()에 전달된 FIFO가 범위를 벗어나면 stopLiveDisplay() 중지

`ccDisplay::stopLiveDisplay()`를 호출하기 전에, `ccDisplay::startLiveDisplay()`에 전달된 FIFO가 손상되지 않았는지 확인해야 합니다. 예를 들어 함수에 FIFO를 만들고, `startLiveDisplay(fifo)`를 호출하여 이 FIFO를 전달한 다음, FIFO를 만들었던 함수가 값을 반환한 뒤에 `stopLiveDisplay()`를 호출하면 `stopLiveDisplay()` 메서드가 중지되고 결과를 반환하지 않습니다.

복수의 모니터 지원 안 함

CVL은 모니터가 두 대 이상인 시스템에서 디스플레이 콘솔 창의 출력을 특정 모니터로 보낼 수 있는 API를 제공하지 않습니다.

타원의 원환 섹션을 채울 수 없음

`ccGraphicProps::penfill()` 함수는 `ccEllipseAnnulusSection` 개체에 사용할 수 없습니다.

ccUIGenAnnulus에 채우기를 설정하면 그래픽 전체가 채워짐

`ccUIGenAnnulus`에 대해 채우기 속성을 참으로 설정하면 원환만이 아니라 모양 전체가 채워집니다.

Sony DXC-390 및 8비트 데스크탑의 디스플레이 공백 현상

MVS-8504, Sony DXC-390 카메라, "Sony DXC-390 640x480 IntDrv CCF" 비디오 형식, `ccAcqFifoPtrh`를 사용하여 실시간 디스플레이를 실행하는 테스트 케이스를 작성했습니다. 8비트 Windows 데스크탑으로 출력한 결과, 표시되어야 할 영상이 공백인 것이 관찰되었습니다.

해결 방법

이 구성을 사용할 때는 출력 영상을 16비트 또는 32비트의 데스크탑 색 심도로 보십시오.

드라이버

이 섹션에서는 CVL 9.0의 드라이버와 관련된 미해결 문제를 설명합니다.

상속된 PC 인터럽트 대기 시간으로 인해 고속에서 활상이 누락될 수 있음

Cognex의 테스트 중, 일부 PC의 인터럽트 응답 시간이 비교적 긴 것이 관찰되었습니다(5-6ms). 대부분의 CVL 응용 프로그램에서는 대기 시간이 이렇게 길어도 문제가 발생하지 않습니다. 그러나 활상 속도가 6ms당 활상 1회 이상으로 빠를 때는 이렇게 긴 대기 시간 때문에 인터럽트 응답이 제때에 이루어지지 않고, 이로 인해 활상이 누락됩니다. 일부 PC의 긴 대기 시간은 적시 인터럽트 응답이 필요한 다른 장치에도 문제를 일으킬 수 있습니다.

이 문제가 발생하는 것으로 알려진 PC는 다음과 같습니다.

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

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

이 문제는 ROI(관심 영역) 활상에 영향을 미칩니다. 영상은 작는데 프레임 속도가 정상보다 빠르면 인터럽트 오버런의 가능성이 있습니다. 그러므로 부분 스캔을 지원하는 MVS-8500으로 단순화하고(최종 결과는 ROI와 동일) ROI 컨트롤을 지원하는 카메라를 사용하면 알려진 문제가 발생할 수 있습니다.

개발 프레임워크

이 섹션에서는 CVL 9.0의 CVL 개발 환경과 관련된 미해결 문제를 설명합니다.

구형 iostream 라이브러리와 신형 iostream 라이브러리를 둘 다 사용할 수 없음

CVL은 표준 네임스페이스의 신형 `iostream` 라이브러리를 사용합니다. 따라서 CVL 헤더를 포함시키면 신형 `iostream` 헤더도 포함됩니다. 구형 `iostreams` 라이브러리 또는 구형 `iostreams` 라이브러리를 사용하는 다른 제품(예: MatLab)을 사용하는 경우, 소스 파일 하나에 양쪽 라이브러리의 헤더를 포함시키고자 할 때 컴파일 오류가 발생할 수 있습니다. 이 문제는 Windows ios 클래스에서 기인하는 것으로 생각됩니다.

두 클래스에서 사용 가능한 enum의 값이 서로 다르기 때문에 일부 클래스에서는 컴파일 시간 문제인 동시에 다른 클래스에서는 잠재적인 런타임 문제가 되기도 합니다. 현재 소프트웨어가 구형 `iostream` 라이브러리를 사용한다면 신형 `iostream` 라이브러리로 업데이트하십시오. 구형과 신형 `iostream` 라이브러리를 둘 다 사용해야 하는 소프트웨어라면 항상 신형 `iostream` 라이브러리를 먼저 포함시키는 것이 좋습니다.

`stdafx.h`를 사용하는 경우에는(모든 소스 파일에서 먼저 등장) `stdafx.h` 맨 위에 다음과 같은 행을 추가하기만 하면 됩니다.

```
#include <ch_cv1/defs.h>
#include <fstream>
#include <iostream>
```

첫 번째로 포함시킨 defs.h는 defs.h의 pragma를 이용하여 "디버그 기호 이름이 너무 김" 컴파일러 경고(C4786)를 비활성화하기 위한 것입니다. 경고를 유발하는 파일을 포함시키기 전에, 또는 경고가 발생하기 전에 이 pragma가 실행되어야 합니다. defs.h를 넣는 대신 이 pragma를 직접 사용할 수도 있습니다.

샘플 코드

이 섹션에서는 CVL 9.0의 샘플 코드와 관련된 미해결 문제를 설명합니다.

런타임 중 Pmalign4 샘플 중단

현재 디렉터리가 실행 파일 위치로 설정된 상태에서 Windows 탐색기나 명령 프롬프트 명령줄을 사용하여 CVL과 함께 제공된 pmalign4 샘플 응용 프로그램(위치: \$VISION_ROOTWsampleWcvl)을 실행하면 "plate.idb를 열 수 없음" 오류와 함께 중단됩니다. 이 문제는 설치 중 plate.idb 파일의 상대 경로가 올바르게 설정되지 않았기 때문입니다.

해결 방법

샘플을 Visual Studio에서 구축하고 실행하거나, plate.idb 및 plate.dxf 파일을 WcvlprojW 디렉터리로 이동하십시오.

직렬 I/O

이 섹션에서는 CVL 9.0의 직렬 I/O와 관련된 미해결 문제를 설명합니다.

직렬 I/O: 첫 번째 바이트가 수신될 때까지 읽기 작업이 타임아웃을 무시함

유한한 타임아웃을 지정했는데도 시리얼 포트를 통해 최소한 1바이트가 수신될 때까지 ccSerialIO::read()가 무한 차단됩니다. 시리얼 포트에 첫 번째 바이트가 수신되는 시점부터 read()가 제대로 작동합니다.

모양 모델

이 섹션에서는 CVL 9.0의 모양 모델과 관련된 미해결 문제를 설명합니다.

바인딩되지 않은 픽셀 버퍼를 기록하려고 하면 충돌 발생

ccShapeModel::record() 메서드는 제공된 픽셀 버퍼가 바인딩되어 있어야 합니다. 그러나 바인딩되지 않은 픽셀 버퍼를 제공하면 이 메서드는 예외를 보내지 않고 0으로 나누려고 시도하기 때문에 결과적으로 응용 프로그램이 충돌합니다. 예를 들어, 다음 호출은 결과적으로 충돌을 일으킵니다. 이것은 두 번째 인수로 제공된 기본 구성의 8비트 ccPelBuffer가 기본적으로 바인딩되지 않기 때문입니다.

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

해결 방법

항상 너비와 높이를 0이 아닌 숫자로 지정하고 바인딩된 픽셀 버퍼를 ccShapeModel::record() 메서드의 두 번째 인수로 제공하십시오.

캘리퍼 도구

이 섹션에서는 CVL 9.0의 캘리퍼 도구와 관련된 미해결 문제를 설명합니다.

캘리퍼: 클리핑을 활성화한 고정밀 모드에서 추가 가장자리 식별

아핀 사각형으로 영상 클리핑을 설정하고 eHighPrecision 모드에서 캘리퍼를 실행하면 가장자리를 추가로 찾아냅니다. 예를 들어 결과의 투영된 영상(원본 영상의 반전 읽기에 해당)이 200 214 198 197 194 192 190 185 183 111 등으로 읽힐 때, 대비 임계값을 5로 설정하면 픽셀 값 214는 영상 오른쪽에 추가 가장자리가 나타나게 합니다. 모드를 eBilinear로 변경하면 투영된 영상이 198 198 197 196 193 191 189 184 176 107 등으로 읽히면서 원본 영상이 올바르게 표현됩니다.

해결 방법

고정밀 모드에서는 원래 가장자리에 링이 만들어집니다. 해결 방법은 bilinear 모드를 사용하는 것입니다.

캘리퍼: 클리핑하면 결과 그래픽이 그려지지 않음

drawMode 인수의 ccCaliperDefs::eDrawProjFilter를 ccCaliperOneResult::draw()로 지정하고 투영 영역을 입력 영상으로 클리핑하면 그래픽(투영 영상과 필터링된 투영 영상의 그래픽 표현)이 표시되지 않습니다. 이 문제가 발생하지 않게 하려면 투영 영역을 완전히 감싸는 입력 영상을 사용하십시오.

캘리퍼: 클리핑을 활성화하면 투영된 영상과 필터링된 영상이 그려지지 않음

투영 모드와 관계없이, 클리핑을 활성화하면 ccCaliperResultSet::draw() 메서드가 투영된 영상과 필터링된 영상을 지정한 대로 그리지 않습니다. 예를 들어 다음 코드는 라벨과 영역만 그립니다.

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

그러므로 투영 영상을 추적하기가 쉽지 않습니다.

가장자리 감지기 도구

이 섹션에서는 CVL 9.0의 가장자리 도구와 관련된 미해결 문제를 설명합니다.

가장자리: ccDiagObject 기록이 기본적으로 깊은 복사를 수행하지 않음

깊은 복사가 기본값이든 diagFlags로 알은 복사를 강제 적용하든 간에, cfEdgeDetect()가 진단 개체를 기록한 뒤 원본 영상, 확대 영상, 각도 영상에 그릴 수 있으며 그러한 수정 사항이 진단 뷰어에 표시됩니다.

히스토그램 도구

이 섹션에서는 CVL 9.0의 히스토그램 도구와 관련된 미해결 문제를 설명합니다.

cfPelHistogram() 동작이 헤더 파일 설명과 일치하지 않음

<ch_cvl/histo.h> 헤더 파일의 cfPelHistogram() 함수에 대한 설명에 따르면, 전달된 벡터의 크기가 0이면 함수가 그 벡터의 크기를 조정하고 무시한다고 잘못 설명되어 있습니다. 그러나 이 함수는 아무 요소도 없는 벡터의 크기를 조정하거나 무시하지 않습니다.

ID 도구

이 섹션에서는 CVL 9.0의 ID 도구와 관련된 미해결 문제를 설명합니다.

혼합 코드의 경우, 결과를 조합할 때 2D 구성요소의 하위 유형을 파악할 수 없음

ccSymbologyParamsComposite::combineResults()를 참으로 지정하는 경우, 2D 구성요소의 하위 유형(CC-A 또는 CC-B)을 파악할 수 없습니다. 선형 부품의 하위 유형만 반환됩니다.

OCV 도구

이 섹션에서는 CVL 9.0의 OCV 도구와 관련된 미해결 문제를 설명합니다.

회전 불확실성을 활성화하면 혼동되는 문자 오탐지 후 위치 확인 실패

문자 회전 불확실성이 0보다 큰 값으로 설정되어 있고 현재 키가 '4'로 설정되어 있는데 영상의 문자 'A'를 확인하려고 하면 (또는 그 반대) OCV 도구가 이상 동작을 보일 수 있습니다. 이러한 문자는 회전을 허용하면 혼동되기 쉽습니다.

회전 불확실성을 설정하면 이 도구는 대개 A-4 불일치를 확인하고(점수는 낮음), 이 불일치 이후에는 모든 위치에서 확인에 실패합니다. 5도에서 45도까지 다양한 불확실성 범위와 세 가지 글꼴(Courier, Lucida Sans, Terminal)에서 이러한 동작이 관찰되었습니다. 일부 불확실성 설정에서 Courier 글꼴은 불일치 이후에도 한두 곳의 위치를 확인하지만, 그 뒤로 실패하기 시작합니다. 이러한 동작은 불일치 발생 지점(문자열 시작 또는 끝부분)과 불일치 개수 모두 관계없이 일정하게 발생합니다.

혼동 문자 대신 공백(키 32) 사용 불가

OCV 도구는 공백 문자와 실제 문자를 혼동할 리 없다고 가정합니다. 그러므로 이 도구는 예를 들어 문자 'a'가 있어야 하는 자리에 공백이 보이는지 여부를 결코 확인하지 않습니다. 그러나 그 반대는 성립하지 않습니다. 이 도구는 공백이어야 하는 문자가 'a'처럼 보이는지 여부를 반드시 확인합니다('a'가 혼동되는 문자인 경우).

라인 배열 경계 상자의 라인 끝에 와일드카드 배제

OCV 라인 배열 경계 상자에서는 라인 끝에 와일드카드를 사용하지 않습니다.

진단 후 라인이 발견되지 않으면 최종 위치에 빨간색 십자가 표시

라인을 전혀 찾을 수 없는 경우(모든 문자가 발견되지 않고 점수는 0.0), "최종 위치의 라인 배열" 진단이 라인 배열 위치에 빨간색 십자가를 그립니다. 이 십자가는 클라이언트 좌표에 작성됩니다.

PatInspect

이 섹션에서는 CVL 9.0의 PatInspect와 관련된 미해결 문제를 설명합니다.

PatInspect: 가장자리 추가 또는 누락이 잘못 보고됨

PatInspect 경계 차이 모드에서 검사 영역 가장자리 근처에 자리잡은 특징이 없거나 추가된 것이라고 잘못 보고될 수 있습니다. 예를 들어 검사 영역이 트레이닝 영역과 일치하고 모든 경계 특징이 동일한 유형일 때 이러한 현상이 발생합니다. CVL 5.4 역호환 모드 및 호환 모드 양쪽에서 이러한 동작이 관찰되었습니다.

해결 방법

사각형 검사 영역에서 트레이닝할 패턴 주변에 최소 5픽셀의 패딩을 두어야 합니다.

PatMax

이 섹션에서는 CVL 9.0의 PatMax와 관련된 미해결 문제를 설명합니다.

PatFlex가 런타임 매개변수의 명목 DOF 값 무시

PatFlex 알고리즘은 nominal() 설정기로 cc_PMRunParams에 설정된 명목 DOF 값을 모두 무시합니다.

해결 방법

PatFlex와 함께 명목 DOF 값을 사용하려면, 주어진 DOF를 활성화하고 구역 상한값과 하한값 둘 다 원하는 명목 값으로 설정합니다. 예를 들어, 다음 값을 쓰지 마십시오.

```
runParams.zoneEnable(0);  
runParams.nominal(ccPMAAlignDefs::kUniformScale, 2.0);  
그 대신 다음 값을 쓰십시오.  
runParams.zoneEnable(ccPMAAlignDefs::kUniformScale);  
runParams.zone(ccPMAAlignDefs::kUniformScale, 2.0, 2.0);
```

웨이퍼 미리 맞춤 도구

이 섹션에서는 CVL 9.0의 웨이퍼 미리 맞춤 도구와 관련된 미해결 문제를 설명합니다.

NotchMax 모드에서 평면 길이 계산 부정확

NotchMax 모드에서 평면 길이를 계산할 때는 웨이퍼 면적, 내부 직경 및 외부 직경을 사용하며 사용자는 ccWaferPreAlignResult::flatLength()로 이를 쿼리할 수 있습니다. 그러나 웨이퍼 하나에 평면이 1개, 2개, 심지어 3개까지 있을 수 있습니다. 이로 인해 면적 계산이 달라지고, 이는 다시 내부 및 외부 직경 계산에 영향을 미칩니다. 따라서 부정확한 평면 길이가 반환될 수 있습니다. 이러한 이유로 평면 웨이퍼에서는 NotchMax 모드가 지원되지 않습니다.

제약

제약이란 하드웨어, 운영 체제 소프트웨어, 제품 아키텍처 또는 기타 외부의 구속 조건으로 인해 CVL에 적용되는 동작을 말합니다. 일부 제약에는 해결 방법이 있을 수 있습니다.

보안

이 섹션에서는 이번 릴리스의 소프트웨어 보안과 관련된 제약을 설명합니다.

이 릴리스는 ccSecurityInfo를 지원하지 않음

이 릴리스에서는 CVL 클래스 참조서에 설명된 ccSecurityInfo API를 지원하지 않습니다.

촬상

이 섹션에서는 영상 촬상과 관련된 일반적인 제약을 설명합니다.

스트로브 촬상의 기본 노출

모든 카메라의 기본 노출인 $20\mu\text{s}$ 는 스트로브 촬상용입니다. Sony XC-55와 같은 고속 리셋 카메라에 주변 조명을 사용하는 경우, 노출 시간을 1ms 이상으로 늘려야 할 수 있습니다.

Pulnix TM-6CN 정적 비디오 형식이 잘못된 동기화 모델로 생성됨

Pulnix TM-6CN의 정적 비디오 형식이 내부 동기화를 사용하는 것으로 표시되지만 이 카메라에는 맞지 않으며 이상한 영상이 출력될 수 있습니다. TM-6CN CCF 버전의 비디오 형식에서는 동기화 모델이 올바르게 설정됩니다.

해결 방법

정적 비디오 형식을 사용할 때는 코드의 동기화 속성을 수동으로 `cfSyncExternalPLLComposite()`로 설정하십시오.

MVS-8500 촬상

이 섹션에서는 CVL 9.0의 MVS-8500 촬상의 제약에 대해 설명합니다.

촬상 밝기 속성 기본값

8504 및 8501 프레임 그래버의 촬상 밝기 속성 기본값이 0.5로 변경되었습니다. 이로 인해 이번 릴리스에서 촬상한 영상은 이전 릴리스에서 촬상 FIFO를 만든 직후에 다른 기본값을 사용하여 촬상한 것과 다르게 보일 수 있습니다.

CFG-8704e 촬상

이 섹션에서는 CVL 9.0의 CFG-8704e 프레임 그래버와 관련된 제약을 설명합니다.

CFG-8704e 프레임 그래버를 두 개까지만 지원

동일한 컴퓨터에서 CFG-8704e 프레임 그래버를 두 개까지만 사용할 수 있습니다.

디스플레이

디스플레이

이 섹션에서는 영상 표시의 제약에 대해 설명합니다.

UI 모양을 여러 개 추가하거나 제거하는 데 시간이 오래 걸림

조작 가능한 모양을 디스플레이에 여러 개 추가하거나 디스플레이에서 이를 제거할 때 한 번에 하나씩 차례대로 하면 시간이 오래 걸릴 수 있습니다.

해결 방법

조작 가능한 모양을 추가하거나 제거하기 전에 `ccDisplay::disableDrawing()`을 호출하고, 모든 모양이 추가 또는 제거되면 이어서 `ccDisplay::enableDrawing(true)`를 호출하십시오.

ccUIGenAnnulus::move()가 잘못 구현됨

`ccUIGenAnnulus::move()` 메서드가 `ccUIGenAnnulus`를 제공된 `cc2Vect`가 지정하는 위치로 옮겨 놓습니다. 이것은 잘못된 동작입니다. 이 메서드는 제공된 `cc2Vect`의 x 값과 y 값으로 모양 위치를 조정해야 합니다.

해결 방법

`ccUIGenAnnulus::frontKid()`를 사용하여 UI에서 생성된 원환의 전면 하위 개체의 위치를 검색하고, 검색된 결과에 오프셋을 추가하고, 그런 다음 `ccUIGenAnnulus::pos(p)`를 호출하여 조정된 위치를 인수로 전달합니다.

8비트 데스크탑에서 고정밀 모드의 줌 보간

8비트 데스크탑에서 `ccUITablet::interpolation(eHighPrecision)`을 호출하여 줌 보간을 고정밀 모드로 설정하면 그레이스케일 범위를 벗어나는 색상이 표시될 수 있습니다.

해결 방법

고정밀 보간법 대신 8비트가 아닌 디스플레이 및/또는 양선형 보간법을 사용하십시오.

W2K 전원 관리가 라이브 비디오와 충돌

Windows 2000 사용자는 CVL 사용자 안내서에 수록된 최적의 성능을 위한 시스템 구성 조정 섹션의 설명에 따라 전원 관리 기능을 비활성화하십시오. 특히 시스템이 절대로 비디오 모니터를 종료하지 않도록 해야 합니다. 비디오 모니터를 종료하도록 전원 관리 기능을 구성하는 경우, CVL 응용 프로그램이 라이브 비디오가 있는 그래픽을 표시하는 데 문제가 생길 수 있습니다.

CVL 5.3에서 보관한 타원을 새 버전의 CVL에서 읽지 못함

CVL 5.4부터는 타원 기하형상 클래스(ccEllipseGeom)의 내부 구조가 변경되었습니다. CVL 5.4 이후 버전을 기준으로 구축된 응용 프로그램이 CVL 5.3 또는 이전 버전에서 보관한 ccEllipse, ccEllipseArc, ccEllipseAnnulus 또는 ccEllipseAnnulusSection 개체가 포함된 ccUISketch 목록을 표시하려고 하면 응용 프로그램이 중지될 수 있습니다.

릴리스 모드와 디버그 모드에서 정적 모양이 다르게 렌더링됨

릴리스 모드와 디버그 모드에서는 정적 모양이 다르게 렌더링됩니다. 모양의 기하형상에 따라, 두 모드 간에 렌더링되는 픽셀이 약간 달라질 수 있습니다. 디버그 모드와 릴리스 모드의 빌드로 작업할 때는 정확히 같은 픽셀이 렌더링될 것으로 확인하지 마십시오.

개발 프레임워크

개발 프레임워크

이 섹션에서는 CVL 개발 환경의 제약을 설명합니다.

"`__declspec(wchar_t)` as builtin type"를 위한 Visual Studio 프로젝트 설정

Visual Studio에서 사용하는 `__declspec(wchar_t)` as builtin type의 기본값은 true입니다. Cognex는 Visual Studio 기본값을 사용하므로, 사용자의 프로젝트에서도 `__declspec(wchar_t)` as builtin type == true이어야 합니다.

설치

여기에서는 CVL을 설치할 때 알아 두어야 할 제한사항을 설명합니다.

명령줄 창을 클릭하면 설치 제거가 일시 중지됨

CVL을 설치 제거할 때 명령 프롬프트 창의 QuickEdit 모드가 활성 상태이고 설치 제거 중 열리는 명령 프롬프트 창을 클릭하면 커서가 선택 모드로 바뀌면서 설치 제거가 중지된 것처럼 보입니다. 이 경우 명령 프롬프트 창을 마우스 오른쪽 버튼으로 클릭하여 선택 모드를 종료하고 설치 제거를 계속하면 됩니다.

분리형 I/O

이 섹션에서는 CVL 9.0 분리형 I/O의 제약에 대해 설명합니다.

병렬 I/O: 외부 I/O 모듈의 출력 극성에 일관성 없음

범용 병렬 I/O 카드(P/N 800-5726-3)의 외부 구성을 외부 I/O 모듈(P/N 800-5712-2)과 함께 사용하여 병렬 I/O 라인의 광절연 연결 지점을 제공합니다. 여기에 MVS-8120을 함께 사용하면 외부 I/O 모듈의 출력 라인 4에서 7까지의 극성이 출력 라인 0에서 3까지와 다르게 반전되며, 이는 다른 Cognex 하드웨어의 광절연 출력 라인과도 반대입니다.

CVL 클래스 참조에는 `ccOutputLine::set(true)` 구성원 함수가 광절연 출력 라인의 경우 전류가 흐르지 않는 상태로 설정한다고 올바르게 설명되어 있습니다. 외부 I/O 모듈의 출력 라인 0에서 3까지 및 다른 Cognex 기판의 경우 이 설명이 맞습니다. 그러나 MVS-8120과 함께 사용하는 경우 외부 I/O 모듈의 출력 라인 4에서 7까지는 `set(true)` 및 `set(false)`의 의미가 반대입니다.

실제 펄스 폭이 지정한 값보다 짧을 수 있음

경우에 따라, 실제 펄스 신호가 `ccOutputLine::pulse()`에 입력된 너비 매개변수보다 짧을 수 있습니다. 예를 들어, 다음과 같이 500ms 펄스를 만든다고 가정하겠습니다.

```
oline.pulse(true, 0.5, false); // 저-고 전환
```


// 0.5초 폭, 비차단

그 결과 생성되는 펄스는 실제 길이가 최소 490ms에 불과할 수 있습니다.

이것은 운영 체제 타이머의 알려진 제약이며, **pulse()** 기능을 지원하는 모든 Cognex 하드웨어 플랫폼에서 발생합니다. 또한 여러 개의 병렬 출력 라인에서 동시에 **pulse()**를 호출할 경우 이 문제가 더 자주 발생하는 것으로 알려져 있습니다.

해결 방법

응용 프로그램에 예상 펄스 기간의 절대 최솟값이 필요한 경우, 실제로 원하는 펄스 폭보다 1ms 긴 폭 값을 **ccOutputLine::pulse()** 함수에 입력하십시오. 이렇게 하면 실제 펄스 폭이 멀티미디어 타이머의 척도에 따른 다음번 간격까지 늘어납니다.

이와 함께 Microsoft Win32 API 호출 **timeBeginPeriod(1)**을 사용하여 멀티미디어 타이머의 기간을 1ms로 줄이는 것도 펄스 정확도를 높이는 데 도움이 됩니다. 그러나 이 방법을 사용할 때는 타이머를 수정하면 전체 시스템 성능이 저하될 수 있다는 점에 주의해야 합니다. 이렇게 하면 타이머 인터럽트에 더 자주 응답해야 하기 때문입니다.

timeBeginPeriod()를 사용하려면 Win32 mmsystem.h 헤더 파일과 winmm.lib 라이브러리를 코드에 포함시켜야 합니다. Visual C++ 6.0에서는 이 헤더 파일이 VC98WinInclude 디렉터리에 있고, Visual C++ .NET(7.0)에서는 VC7WPlatformSDKWinInclude 디렉터리에 있습니다. 플랫폼 SDK 및 DDK 포함 파일에서도 이 파일을 찾을 수 있습니다.

예를 들어, 응용 프로그램에 다음 코드를 추가하여 시스템 타이머의 정확도를 조정할 수 있습니다.

```
#include <windows.h> // 컴파일러 오류 방지를 위해 추가
#include <mmsystem.h>
#pragma comment(lib, "winmm.lib")
timeBeginPeriod(1);
// CVL에서 펄스 폭 설정 및 I/O 작업 수행
timeEndPeriod(1); // 응용 프로그램 종료 전 호출하여
// 멀티미디어 타이머를 원래 상태로 복구
```

CVL 사용자 안내서의 영상 표시 장에 있는 실시간 영상 표시 섹션에서 시스템 타이머 수정에 대한 정보를 참조하십시오.

유틸리티

이 섹션에서는 CVL 9.0에서 제공하는 유틸리티 프로그램의 제약과 제한사항을 설명합니다.

CVL 명령줄 유틸리티를 CVL bin 디렉터리에 유지해야 합니다.

Cogtool과 같은 CVL 명령줄 유틸리티는 대개 Cognex 기술 지원의 지시에 따라서만 실행하게 됩니다. 이번 릴리스에서는 이러한 유틸리티의 실행 파일이 CVL 제공 DLL이 들어 있는 디렉터리에 상주해야 합니다. 다시 말해, 유틸리티의 실행 파일(예: cogtool.exe)이 %VISION_ROOT%\bin\Wwin32\Wcvl에 들어 있어야 올바른 결과를 얻을 수 있습니다. 유틸리티의 실행 파일을 다른 디렉터리로 복사하고 그곳에서 실행하면 도구는 실행되지만, PATH에 Cognex bin 디렉터리가 있는데도 시스템에 Cognex 보드가 없다는 잘못된 보고 메시지가 표시됩니다.

이것은 유틸리티의 실행 파일을 다른 디렉터리로 옮겨 그곳에서 실행할 수 있었던 CVL 6.2의 동작과 달라진 점입니다.

바코드

이 섹션에서는 CVL 9.0의 바코드 도구와 관련된 제약을 설명합니다.

decode() 함수에 추가된 매개변수가 역호환성 위반

CVL 5.5.0이 5.5.1로 바뀌면서 특히 **decode()** 함수 등 **ccAcuBarCodeTool** 클래스의 일부 가상 함수에 대한 인수 목록에 부울 매개변수인 **autoRetry**가 기본값 참으로 추가되었습니다. 이에 따라 함수의 서명이 변경되었으므로 CVL 5.5.0의 기존 서명을 사용하여 이러한 함수를 덮어쓰더라도 CVL 5.5.1 이상에서는 더 이상 덮어쓸 수 없을 뿐 아니라 덮어쓰기 함수도 호출되지 않습니다. 함수 서명이 바뀌었다는 사실을 알리는 컴파일러 오류는 표시되지 않습니다. CVL 응용 프로그램을 CVL

5.5.0에서 5.5.1 이상으로 업그레이드하는 경우, `ccAcuBarCodeTool::decode()`의 덮어쓰기 함수가 최신 함수 서명과 일치하도록 해야 합니다.

블롭

이 섹션에서는 CVL 9.0의 블롭 도구와 관련된 제약을 설명합니다.

블롭: 장면 설명을 참조로 유지하면 메모리 누출

`ccBlobSceneDescription` 개체를 참조로 계속 사용하면서 보관된 파일에서 이 개체를 다시 로드하면 응용 프로그램의 메모리 누출이 발생합니다. `ccBlobSceneDescription` 개체를 참조가 아닌 포인터로 유지하면 이러한 문제를 막을 수 있습니다.

보정

이 섹션에서는 CVL 9.0의 보정 도구와 관련된 제약을 설명합니다.

체커보드 최소 타일 크기

특정 일치 보정을 수행할 때, 크기가 15x15픽셀 미만인 타일에서는 특징 감지기가 안정적으로 작동하지 않습니다. 범위 한계에서 작동하는 것을 막으려면 대부분의 응용 프로그램에서 체커 크기가 15x15 이상이어야 합니다.

캘리퍼

이 섹션에서는 CVL 9.0의 캘리퍼 도구와 관련된 제약을 설명합니다.

필터링된 클리핑 투영의 잘못된 픽셀 범위

필터링된 클리핑 투영 영상의 픽셀 값 범위가 올바르지 않습니다. 필터링된 영상의 범위는 $-255*n \sim +255*n$ 이어야 하는데 $-255 \sim +255$ 입니다(여기서 n 은 해당 투영의 샘플 수). 올바른 범위를 얻으려면 필터링된 이미지의 스케일에 n 을 곱하여 조정하십시오.

가우시안 샘플링

이 섹션에서는 CVL 9.0에 포함된 가우시안 샘플링 도구의 제약을 설명합니다.

cfGaussSample()이 잘못된 결과 생성

16비트 대상 픽셀 버퍼를 사용하는 `cfGaussSample()` 오버로드의 결과값이 32,767로 고정됩니다. 따라서 보통 32,767 이상의 결과를 반환해야 할 모든 값(예: 스케일 설정이 255이고 입력 픽셀 값이 128 초과)이 잘못된 결과를 반환하게 됩니다.

해결 방법

스케일을 충분히 낮게 설정하여 모든 결과가 32,767 미만이 되도록 하십시오.

그레이스케일 모폴로지

이 섹션에서는 CVL 9.0의 그레이스케일 모폴로지와 관련된 제약을 설명합니다.

3x3 커널 구성 이외의 NxM 모폴로지는 멀티코어에 완전히 최적화되지 않음

3x3 커널로 구성되지 않은 NxM 모폴로지의 가변 크기 커널 영상 도구는 멀티코어에 완전히 최적화되지 않았습니다.

ID

이 섹션에서는 CVL 9.0의 ID 도구와 관련된 제약을 설명합니다.

EAN.UCC 혼합 기호 방식에서 RSS 혼합 기호만 지원됨

이 릴리스에서는 RSS 혼합 기호 방식을 사용하는 EAN.UCC 혼합 기호만 지원됩니다.

acuRead

이 섹션에서는 CVL 9.0의 acuRead 도구와 관련된 제약을 설명합니다.

사용자 정의 글꼴에서 끝 문자 누락

사용자 정의 글꼴에서 끝 문자가 누락될 수 있습니다. 이 문제는 고정 길이 모드와 가변 길이 모드 둘 다에서 발생합니다. 고정 길이 모드에서는 마지막 문자가 공백이 됩니다.

해결 방법

ROI의 범위를 넓혀 처음과 끝부분의 공백 픽셀을 늘리십시오.

OCV

이 섹션에서는 CVL 9.0의 OCV 도구와 관련된 미해결 문제를 설명합니다.

공백이 있는데도 문자가 잘못 확인됨

라인 배열의 위치 하나를 현재 키에 공백 문자가 있게 정의했는데 런타임 영상에서는 그 위치가 공백이 아니고 다른 어떤 현재 키와도 일치하지 않는 경우, OCV 도구가 그 위치를 잘못 확인하거나 혼동할 수 있습니다. 현재 키가 공백이면 런타임 시 검색 위치에 실제로 무엇이 있던 관계없이 항상 0이 아닌 점수를 반환합니다. 공백 키에 대해 반환되는 점수는 대개 낮지만, '-' 또는 '.' 처럼 너비가 좁은 문자에서는 점수가 수락 임계값이나 혼동 임계값을 넘어설 수 있으며 따라서 확인에 실패하게 됩니다.

아래에 공백 문자 점수의 몇 가지 예가 나열되어 있습니다.

문자	점수
-	0.423
.	0.422
'	0.382
:	0.312
,	0.276

공백 키 하나만 있을 때보다 공백 키와 다른 현재 키가 하나 더 있을 때 점수가 더 낮은 경우가 관찰되기는 했지만, 이 문제는 해당 위치에 정의된 현재 키가 한 개든 아니면 여러 개든 간에 발생합니다. 이 문제는 OCV 도구가 공백 문자와 "일반" 문자를 다르게 처리하도록 설계된 것과 관련이 있습니다. OCV는 공백 문자가 다른 어떤 것과 혼동되는지 확인하지 않으며, 공백의 점수만 보고 공백 문자의 상태를 파악합니다. 그러므로 작은 문자는 간과하기 쉽습니다.

PatMax

이 섹션에서는 CVL 9.0의 PatMax 도구와 관련된 제약을 설명합니다.

합성 정렬 모델의 트레이닝 문제

영상 좌표에 투영했을 때 경계 상자가 2Kx2K보다 훨씬 큰 모양으로 트레이닝 시간 변형을 사용하여 PatMax를 트레이닝하면 액세스 위반이 발생하거나 응용 프로그램이 train()에서 중지됩니다.

해결된 버그

이 릴리스에는 이전 릴리스에서 고객이 보고한 문제에 대한 수정사항이 포함되어 있습니다.

밝기 및 대비 값으로 인한 오류 발생

영상을 촬상할 때, 이제 간격이 1이 아닌 카메라에 대해 밝기 및 대비 값을 잘못된 해당 설정으로 매핑해도 오류가 발생하지 않습니다.

이 릴리스에 제공되는 설명서

설치된 CVL에는 다음 설명서가 포함되어 있으며, 시작->Cognex->CVL->설명서에서 찾아보실 수 있습니다.

- CVL 사용자 안내서
샘플 코드 작업, 영상 및 그래픽의 활상과 표시, 비전 도구와 개념 소개 등 CVL 프로그래밍의 개괄합니다.
- CVL 비전 도구 안내서
각 CVL 비전 도구의 기능과 각각 사용할 수 있는 매개변수 및 그에 따른 결과를 설명합니다.
- CVL 클래스 참조서
CVL C++ 클래스 라이브러리에 대한 클래스별 참조서입니다.
- CFG-8724 하드웨어 참조서
GigE Vision 영상 활상을 위한 PCI Express x4 호환 장치와 CFG-8724 프레임 그래버를 자세히 설명합니다.

향후 릴리스에 대한 정보

향후의 CVL 릴리스에서는 다음 API가 지원되지 않음을 알려 드립니다.

- 특정 2D 기호와 관련된 `ccAcuSymbol*` 클래스:
 - `ccAcuSymbolDataMatrixDefs`
 - `ccAcuSymbolDataMatrixLearnParams`
 - `ccAcuSymbolDataMatrixTool`
 - `ccAcuSymbolDefs`
 - `ccAcuSymbolFinderParams`
 - `ccAcuSymbolLearnParams`
 - `ccAcuSymbolQRCodeDefs`
 - `ccAcuSymbolQRCodeLearnParams`
 - `ccAcuSymbolQRCodeTool`
- 기본적인 OCR 읽기 및 조정 기능과 관련된 `ccAcuRead*` 클래스:
 - `ccAcuRead`
 - `ccAcuReadDefs`
 - `ccAcuReadFont`
 - `ccAcuReadResult`
 - `ccAcuReadResultSet`
 - `ccAcuReadRunParams`
 - `ccAcuReadTuneParams`

-
- 1D 기호 도구와 관련된 ccAcuBarCode* 클래스:
 - ccAcuBarCodeCalibrationResult
 - ccAcuBarCodeDefs
 - ccAcuBarCodeResult
 - ccAcuBarCodeRunParams
 - ccAcuBarCodeTool
 - ccAcuBarCodeTuneParams

