

Multi Vision Camera

HTC-2030R Camera Development Manual

Ver 2.1

2013. 06. 25



<http://www.hontek.co.kr>

<http://cafe.naver.com/hontek>

Jun 25, 2013

목 차

1. HTC-2030R Camera Specification	4
1.1 General feature	4
1.2 PC Requirements	4
1.3 Contents	5
1.4 외 형	6
1.5 Lens 사양	6
1.6 Header Board Pin Description	7
1.7 Dimension	8
2. 개발 환경	9
2.1 개발 요구사항	9
2.2 Development Architecture	9
3. Software Architecture	10
3.1 Multi Camera connection and Access	10
3.2 Sequence of Initialization	11
3.3 Main Control Dialog	12
3.4 ISP Control Dialog	20
3.5 OpenCV Control Dialog	26
4. System File Description	39
4.1 MVC_Vision.ini	39
5. API Lists	40
5.1 I2C Communication	40
5.2 OpenCV 관련 함수	41

목 차

6. API 설명	42
6.1 MVC_Init	42
6.2 MVC_Release	43
6.3 MVC_EmpiaispAutoGainControlEnable	44
6.4 MVC_EmpiaispGetAWBGainControlRange.....	45
6.5 MVC_EmpiaispSetAWBGainControl	46
6.6 MVC_EmpiaispGetExposureRange	47
6.7 MVC_EmpiaispSetExposureTime	48
6.8 IIC_Read16i	49
6.9 IIC_Write16i	50
6.10 MVC_OpenCV_Create	51
6.11 MVC_OpenCV_Release	52
6.12 MVC_OpenCV_Display	53
6.13 MVC_OpenCV_StartAVI	54
6.14 MVC_OpenCV_StopAVI	55
6.15 MVC_OpenCV_TM_targetFile	56
6.16 MVC_OpenCV_LoadObjectOpenCV	57
6.17 MVC_OpenCV_MatchingObjectOpenCV	58
6.18 MVC_OpenCV_LaserTracking	59
6.19 MVC_OpenCV_SelEdge	60
6.20 MVC_OpenCV_Labeling	61
6.21 MVC_OpenCV_Sharpning	62
7. 기타 문의 사항	63

1. HTC-5030R Camera Specification

1.1 General feature

- ☞ 30만 화소(VGA) 비전카메라, 1/4.5 Inch CMOS Sensor
 - ☞ Effective pixel 640(H) x 480(V)
 - ☞ Frame rate Max. 30fps@VGA
 - ☞ PC Interface USB 2.0 (480Mbps)
 - ☞ Sensor video data format 8Bit Bayer
 - ☞ Sensor command interface 표준 I2C protocol
 - ☞ Compression 압축 안 함
 - ☞ Image Processing Host 프로그램 구현 또는 센서 ISP 컨트롤 가능
 - ☞ Multi-Camera connect 동일 PC 최대 4대 연결 가능
- ※ 비전카메라 사용 시 되도록 다른 **USB** 장비 (휴대폰 연결 등) 를 사용하지 않는 것이 좋습니다.

1.2 PC Requirements

- ☞ CPU Pentium4 2.0GHz이상 (2.8GHz 이상 추천)
- ☞ RAM 1GB 이상 (2GB이상 권장)
- ☞ USB 2.0 Host Controller Intel Controller chip 권장
- ☞ Operating System Windows 2000 SP4 이상 / WindowsXP SP2이상
Windows Vista / Windows 7

1. HTC-2030R Camera Specification

1.3 Contents

- ☞ HTC-2030R Camera
- ☞ USB(A+B) Cable (1EA)
- ☞ User's manual / Program CD



Fig 1. HTC-2030R 포장상태 및 포함 내용물

1. HTC-2030R Camera Specification

1.4 외형 모습

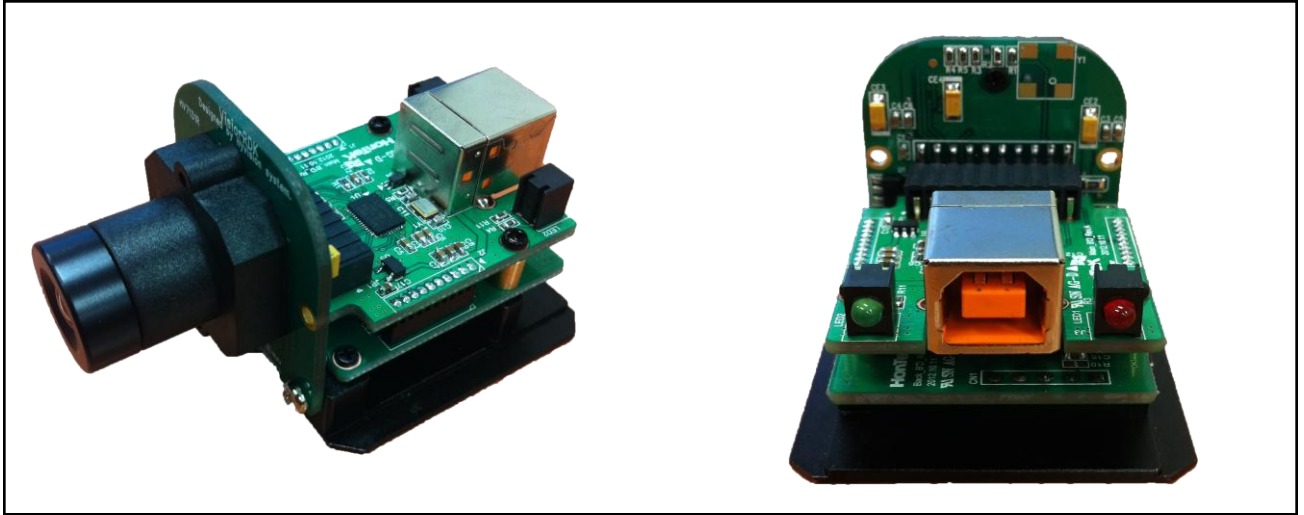


Fig 2. HTC-2030R 전면부 / 후면부 모습

1.5 렌즈 사양

1. Sensing Area : 1/3 inch CMOS
2. Focal Length : 6.0mm
3. Back focal length : 8.12mm
4. F no. : 1.8
5. Iris : Fixed
6. Optical Distortion : -10.62%
7. Field Angle (Diagonal) : 60°
8. Focus extent : 5cm - ∞

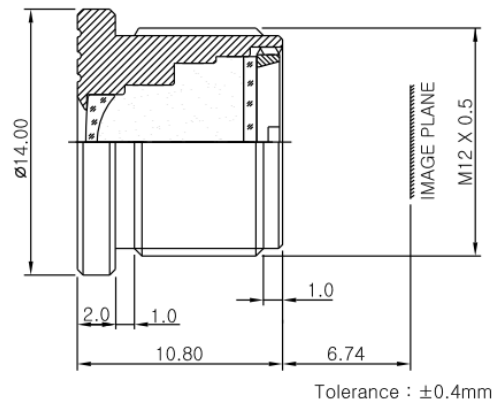
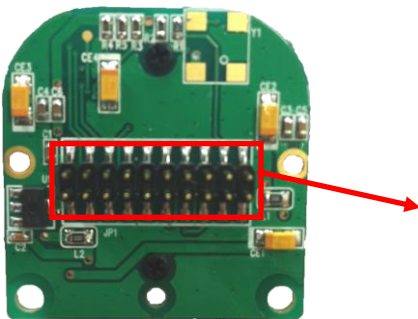


Fig 3. Lens Dimension

1. HTC-2030R Camera Specification

1.6 Header Board Pin

<Sensor board & Pin Number>



2	4	6	8	10	12	14	16	18	20
1	3	5	7	9	11	13	15	17	19

Header board Pin 배열

Fig 6. Header board 뒷면

<Pin Description>

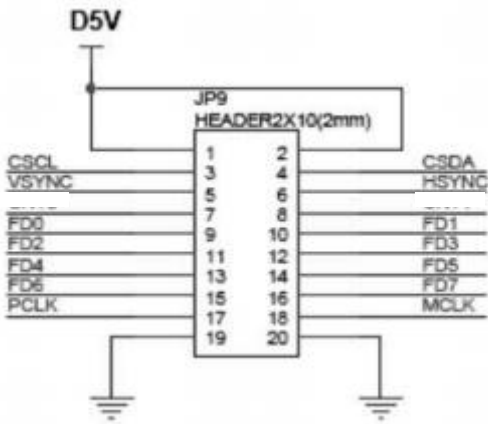


Fig 7. Pin Description

1,2 번 핀의 DC5V는 메인 파워를 공급하는 부분으로 DC 5V가 입력되어야 하며, 100mA 정도의 용량을 공급할 수 있어야 한다. 3,4 번 핀의 CSCL,CSDA는 12C 제어 Port로 별도의 프로토콜을 참조해야 한다. 5,6 번 핀의 VSYNC,HSYNC 는 위에서 설명한 Vertical Sync/Horizontal Sync로 1 frame의 신호를 구성하는 중요 핀이다.

9 ~16번 핀까지가 센서의 8Bit 데이터 출력 핀이다. 17번 핀은 PCLK으로 동기 Clock으로 데이터는 이 PCLK에 맞추어 읽어내야 한다. 18번 핀은 MCLK으로 Master Clock을 Sensor에 공급해 주는 핀이다. PLL 기능이 내장된 센서가 아닌 한 대체로 느린 MCLK를 공급하게 되면 Frame Rate 또한 느려지게 된다. 19,20번 핀은 Digital Ground, DGND 로 접지신호다.

1. HTC-2030R Camera Specification

1.7 Dimension

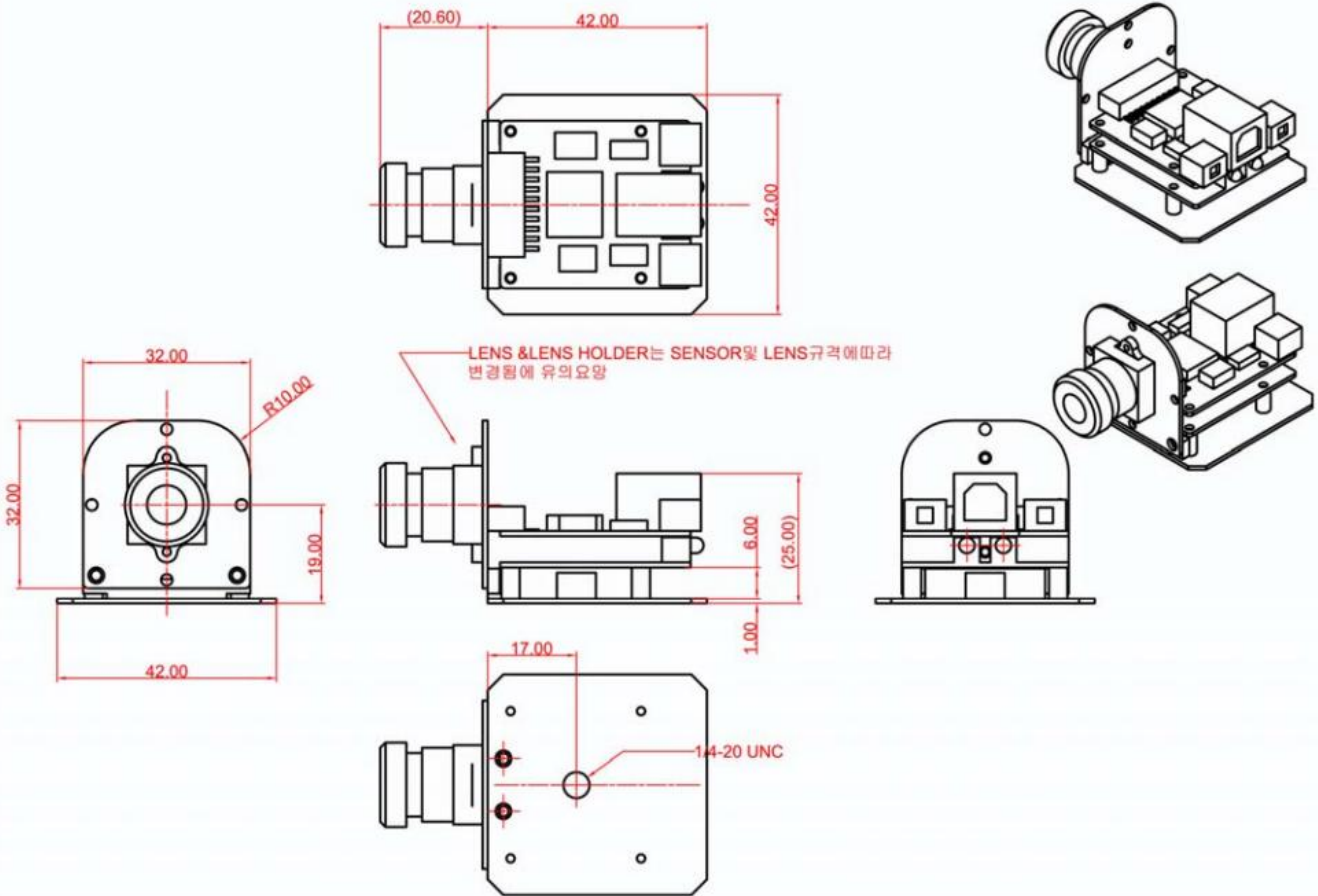


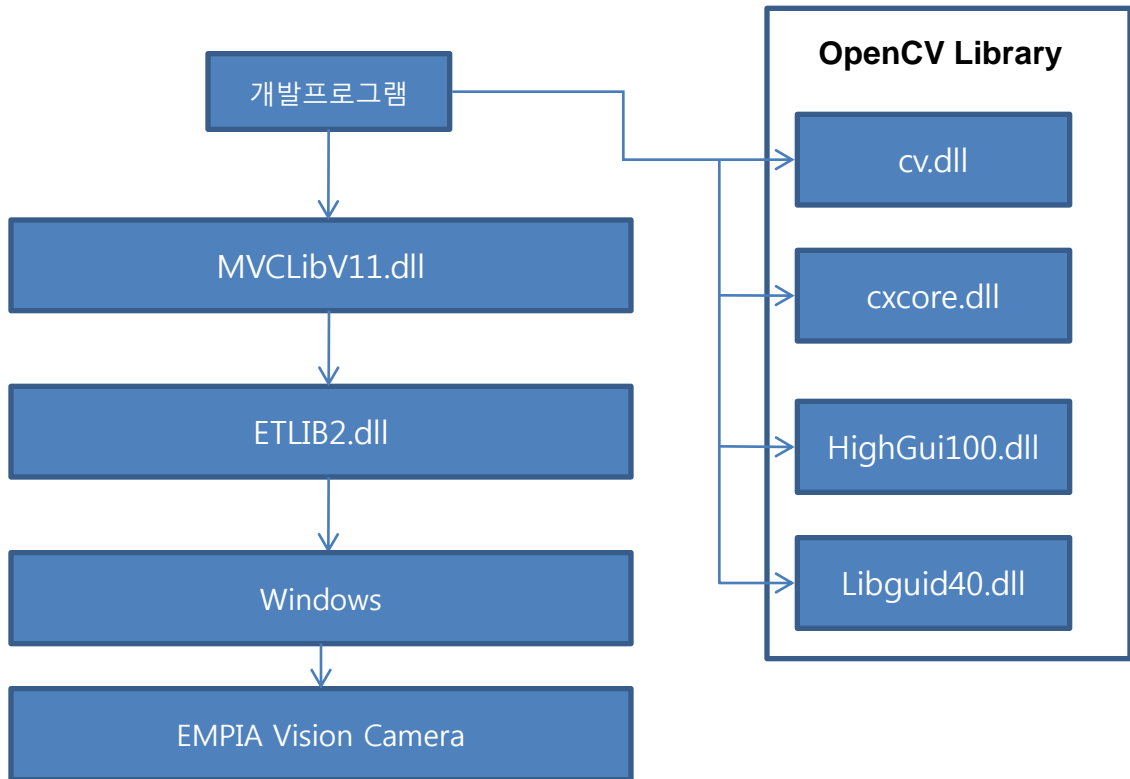
Fig 8. HTC-2030R Dimension

2. 개발 환경

2.1 개발 요구사항

- ☞ 운영체제 : Microsoft Windows 2000 SP4 이상, WindowsXP SP2 이상,
Windows Vista, Windows 7
- ☞ 컴파일러 : Microsoft Visual Basic / Visual C++ 6.0
Visual Basic .NET / Visual C++ .NET
- ☞ 기타 : OpenCV 라이브러리

2.2 Development Architecture



3. Software Architecture

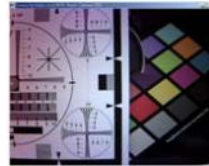
3.1 Multi Camera connection and Access

HTC-2030R 카메라는 동일 PC상에 최대 3대의 카메라 연결이 가능합니다. 카메라에 시리얼 번호를 각각 부여하여(1001~1003) 각 카메라를 식별합니다.

Single Connection



Serial_Number[5]=L"1001";

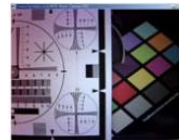


M fps

Multi Connection



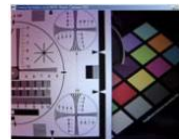
Serial_Number[5]=L"1001";



M/N fps



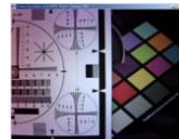
Serial_Number[5]=L"1002";



M/N fps



Serial_Number[5]=L"1003";



M/N fps

Fig 9. Multi Camera connection and Access

3. Software Architecture

3.2 Sequence of Initialization

실행파일 실행시, Device Check에 관한 초기화 과정을 설명합니다. 개발자분은 아래 과정을 참조하시기 바랍니다.

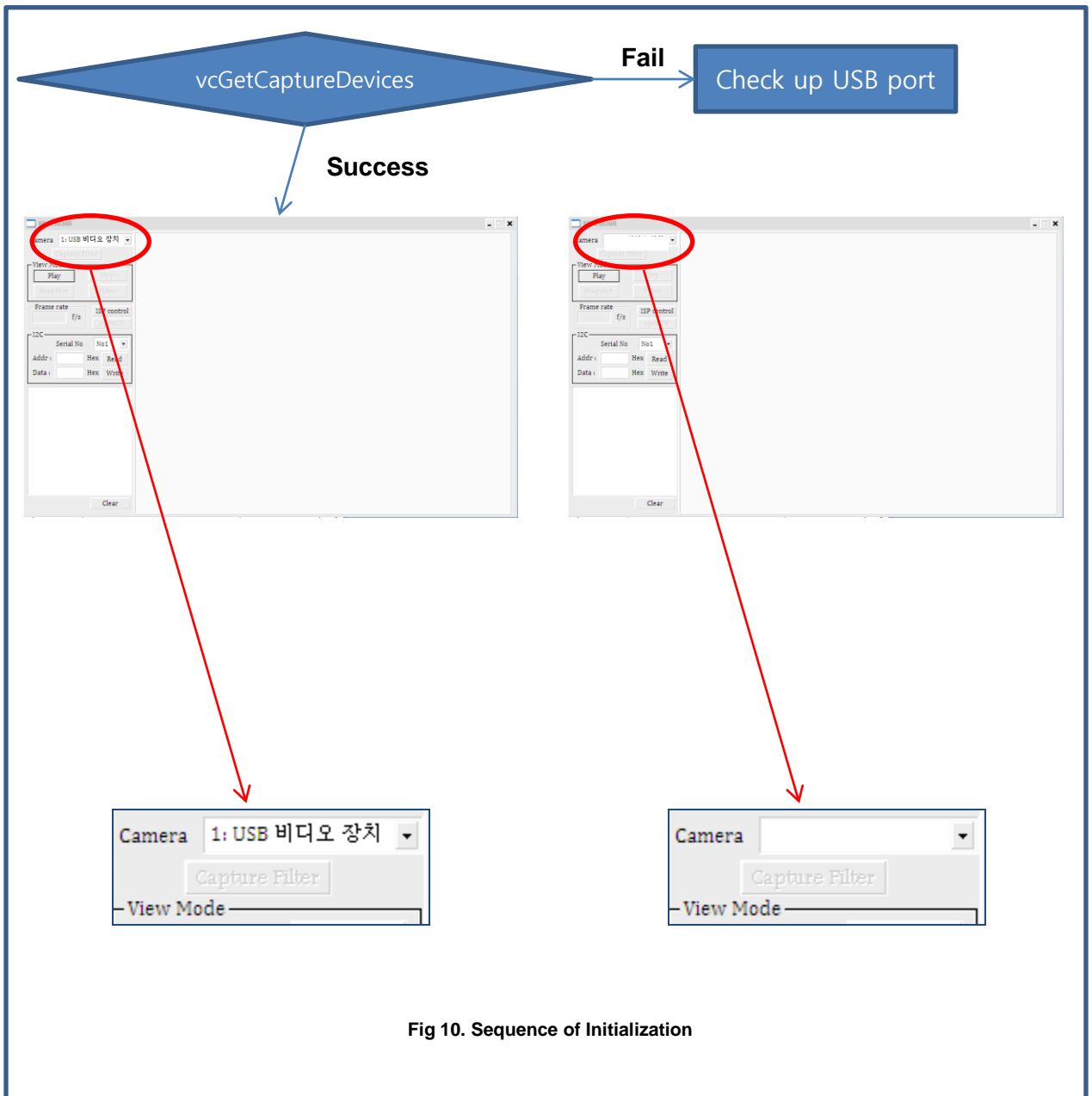
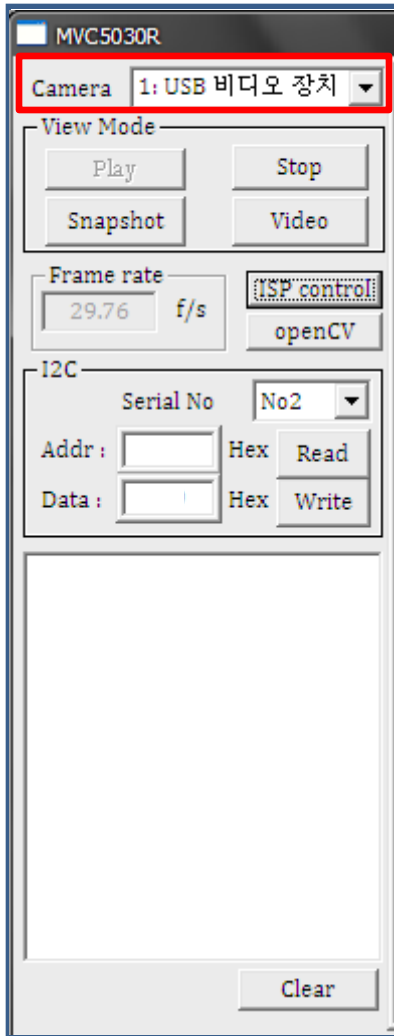


Fig 10. Sequence of Initialization

3. Software Architecture

3.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



3.3.1 카메라 검색

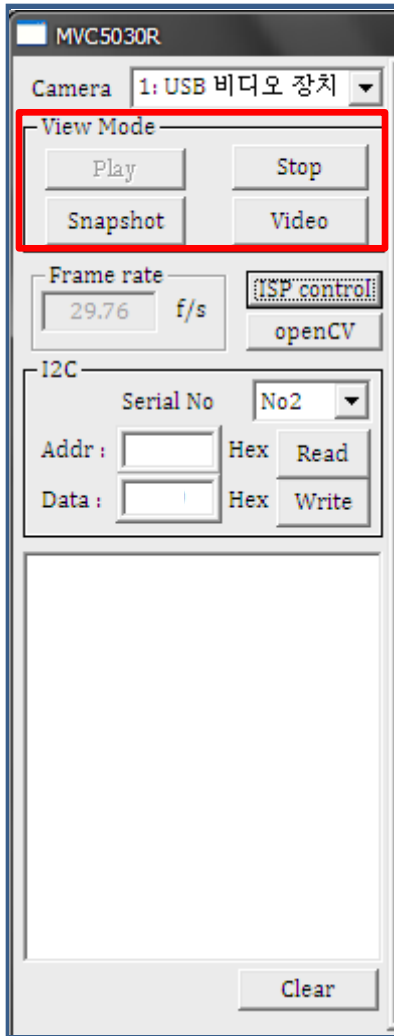
장비와 연결된 카메라를 검색하고 연결된 카메라를 Combo Box에 나열한다.

Fig 11. Main Control Dialog – 카메라 검색

3. Software Architecture

3.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



3.3.2 View Mode

선택된 카메라를 실행시키거나 중지, 또는 정지영상, 동영상을 저장한다.

Play : 카메라를 실행

Stop : 카메라 실생을 중지

Snapshot : 카메라 실행 시에만 동작하며, 하나의 정지영상을 BMP파일로 저장한다.

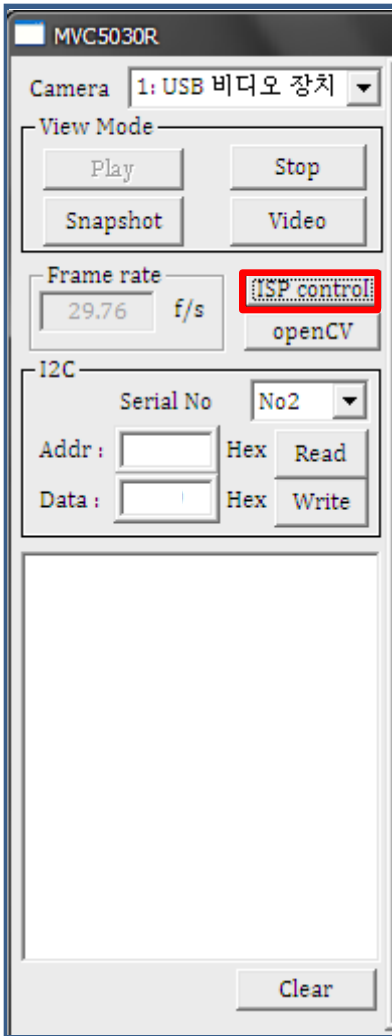
Video : 카메라 실행 시에만 동작하며, 동영상을 저장한다.

Fig 12. Main Control Dialog – View Mode

3. Software Architecture

3.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



3.3.3 ISP(Image Signal Processing) Control

카메라로부터 입력되는 영상을 가공하는 기능을 담당

ISP control : ISP 설정 다이얼로그 실행

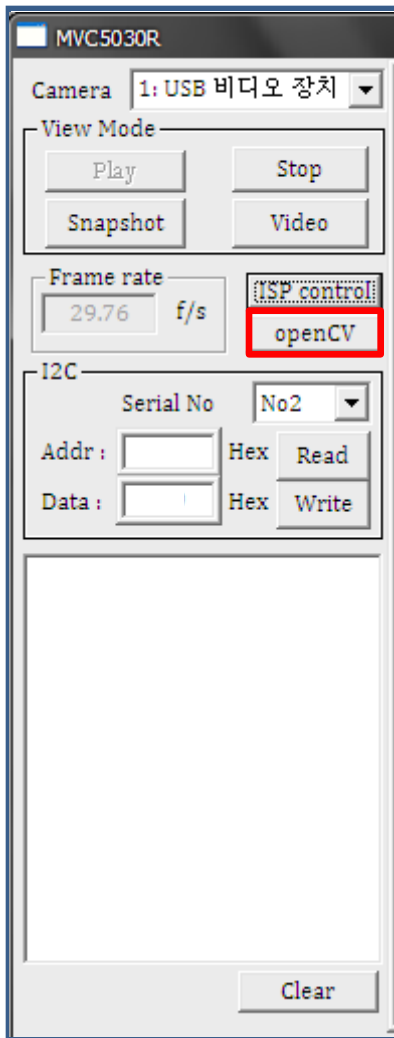
ISP control 메뉴 설명 : 20 ~ 25page

Fig 13. Main Control Dialog – ISP Control

3. Software Architecture

3.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



3.3.4 OpenCV

OpenCV 라이브러리의 이용하여 영상 데이터에 기본적인 효과를 적용

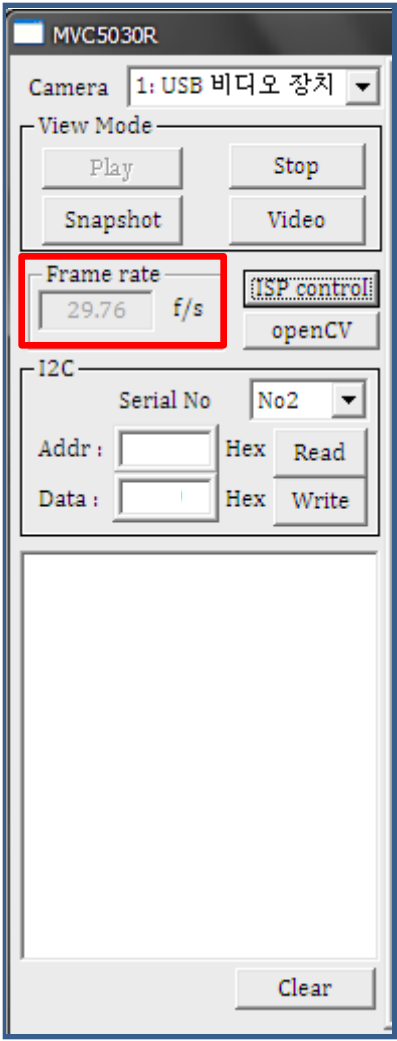
OpenCV 메뉴 설명 : 26 ~ 38 page

Fig 14. Main Control Dialog - OpenCV

3. Software Architecture

3.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



3.3.5 Frame Rate

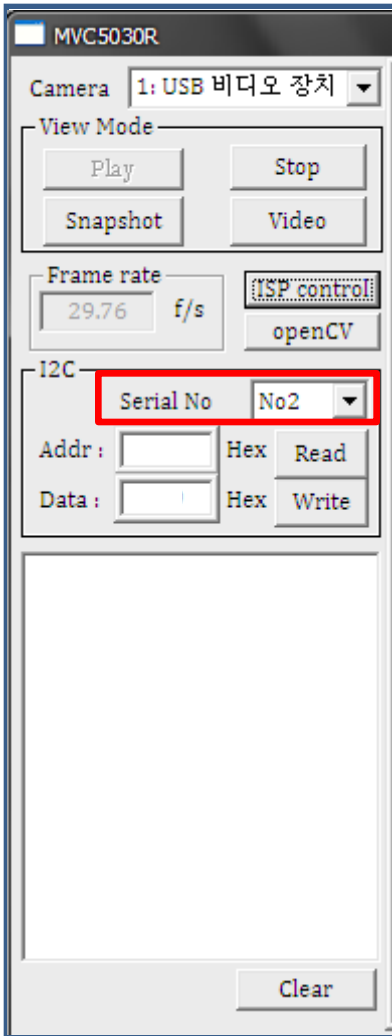
영상데이터를 카메라로부터 입력 받아 화면에 출력하는 속도. (Frame/Sec)

Fig 15. Main Control Dialog – Frame Rate

3. Software Architecture

3.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



3.3.6 Serial No

사용하고자 하는 카메라의 Serial No가 설정 된다.

1001 → No 1

1002 → No 2

1003 → No 3

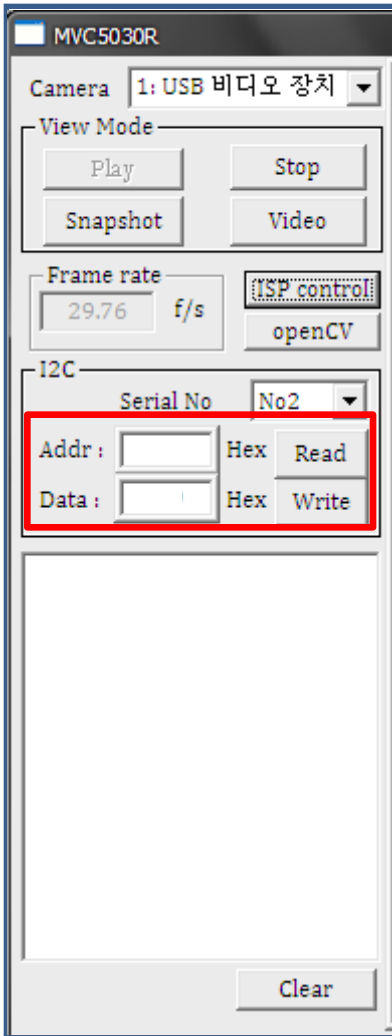
1004 → No 4

Fig 16. Main Control Dialog – Serial No

3. Software Architecture

3.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



3.3.7 I2C

I2C 인터페이스를 이용하여 카메라에 데이터를 쓰거나 읽는다.

```
IIC_Read16i(
Serial_Number, // 카메라 시리얼번호
I2cSubAddr); // 레지스트리 주소
```

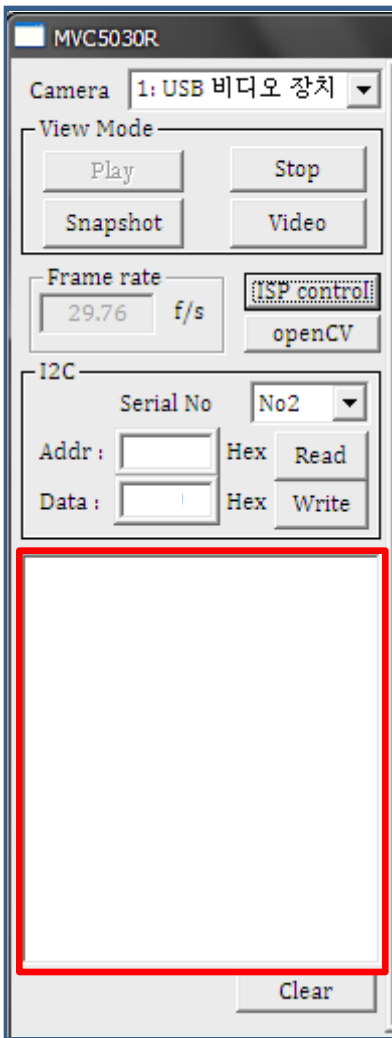
```
IIC_Write16i (
Serial_Number, // 카메라 시리얼 번호
I2cSubAddr, // 레지스트리 주소
I2cValue); // 데이터 값
```

Fig 17. Main Control Dialog – I2C

3. Software Architecture

3.3 Main Control Dialog

프로그램 전체를 운영하는 Dialog base의 controller로서, 그 역할을 설명한다.



3.3.8 상태정보

카메라의 상태나 실행 정보 등을 출력

Fig 18. Main Control Dialog – 상태 정보 창

3. Software Architecture

3.4 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

3.4.1 Exposure Time

카메라로부터 입력된 영상의 ExposureTime(노출 시간)을 설정한다.

High : 0 ~ 255 / Middle : 0 ~ 255 / Low : 0 ~ 255
 적용 값 (0xHHMMLL)

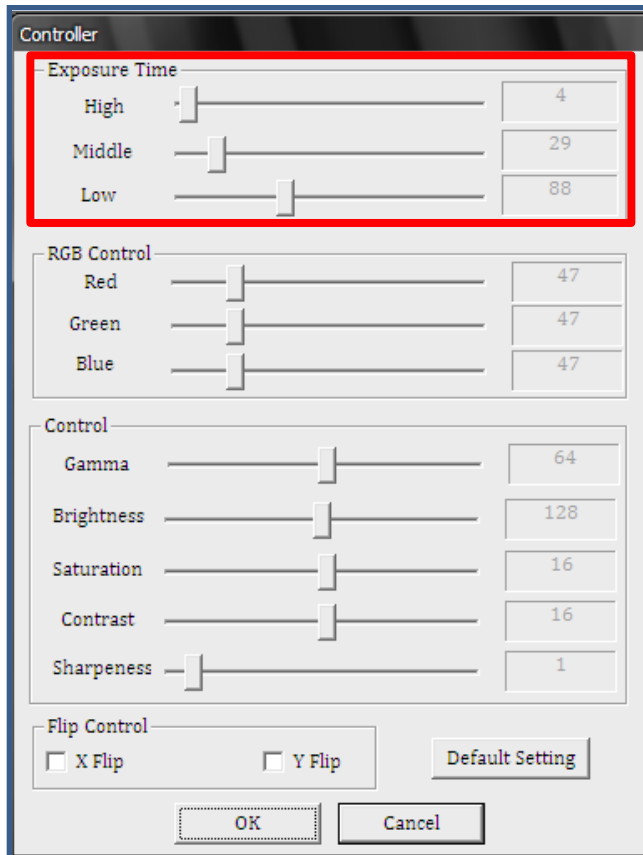


Fig 19. Soft ISP Control Dialog – Exposure Time

3. Software Architecture

3.4 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

3.4.2 RGB색상정보

카메라로부터 입력된 영상의 색상값(Red, Green, Blue)을 설정한다.

Min : 0
Max : 255

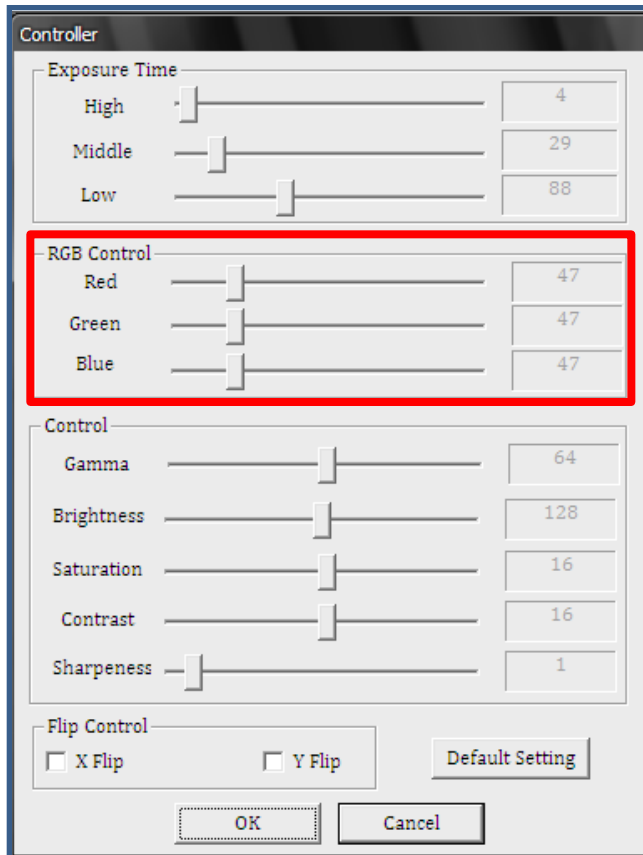


Fig 20. Soft ISP Control Dialog – RGB Control

3. Software Architecture

3.4 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

3.4.3 Control (카메라 설정 값 변경)

연결된 카메라의 설정정보(밝기, 대비, 색상, 채도, 선명도, 감마, 화이트 밸런스 등)를 관리할 수 있다.

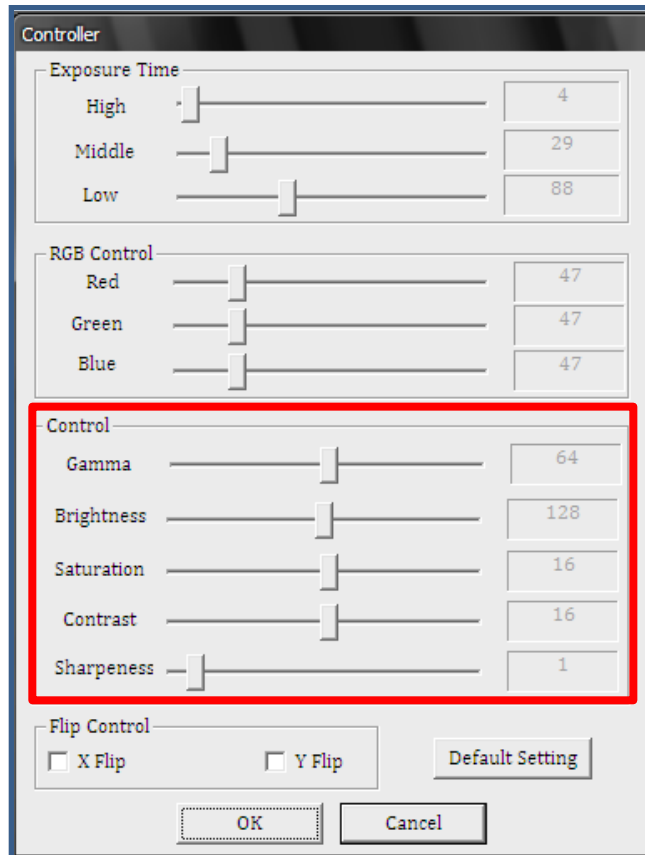


Fig 21. Soft ISP Control Dialog – Control (설정 값 변경)

3. Software Architecture

3.4 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

3.4.4 Flip

카메라로부터 입력된 영상을 X, Y축 회전시킨다.

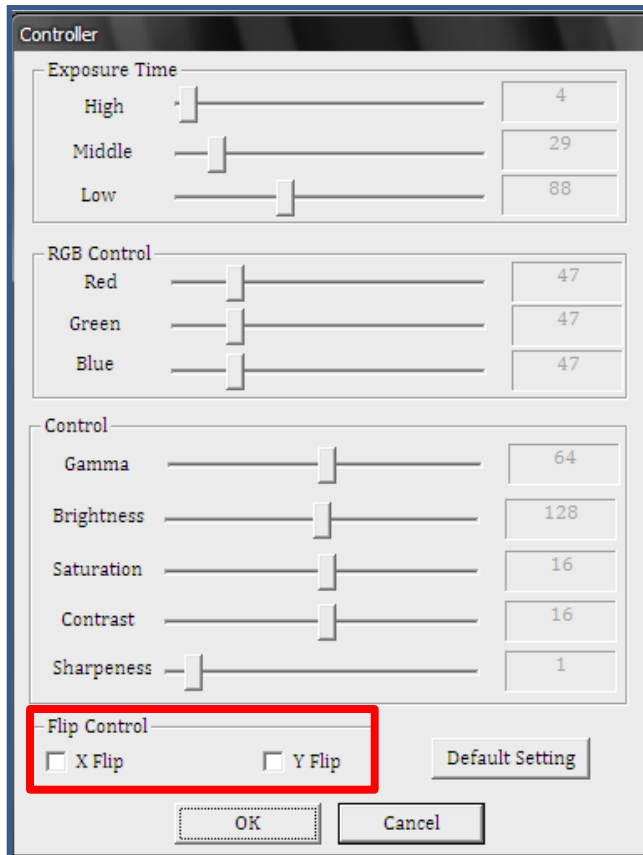


Fig 22. Soft ISP Control Dialog – Flip Control

3. Software Architecture

3.4 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

3.4.5 Default Setting

Exposure Time, RGB 값을 기본값으로 변경한다.

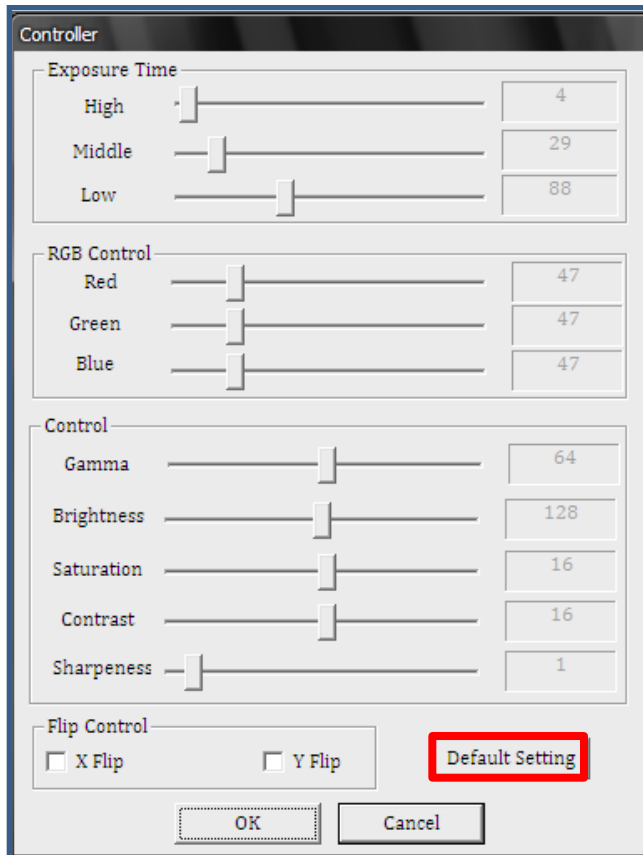


Fig 23. Soft ISP Control Dialog – Default Setting

3. Software Architecture

3.4 ISP(Image Signal Processing) Control Dialog

ISP를 설정할 수 있는 다이얼로그

3.4.6 설정 저장 및 취소

설정된 값들을 유지할지, 취소할지 결정

"OK" → 지금까지 설정된 값들을 유지한다.

"Cancel" → 지금까지 설정한 값을 취소하고, 이전 값으로 돌아간다.

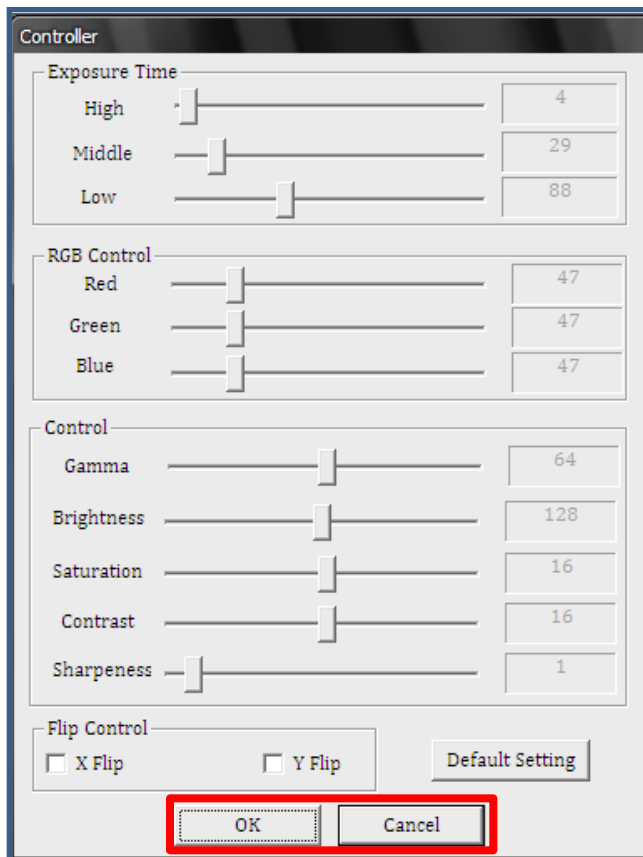


Fig 24. Soft ISP Control Dialog – 설정 저장 및 취소

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.

3.5.1 OpenCV On / Off

OpenCV 기능 On / Off 적용 유무 선택

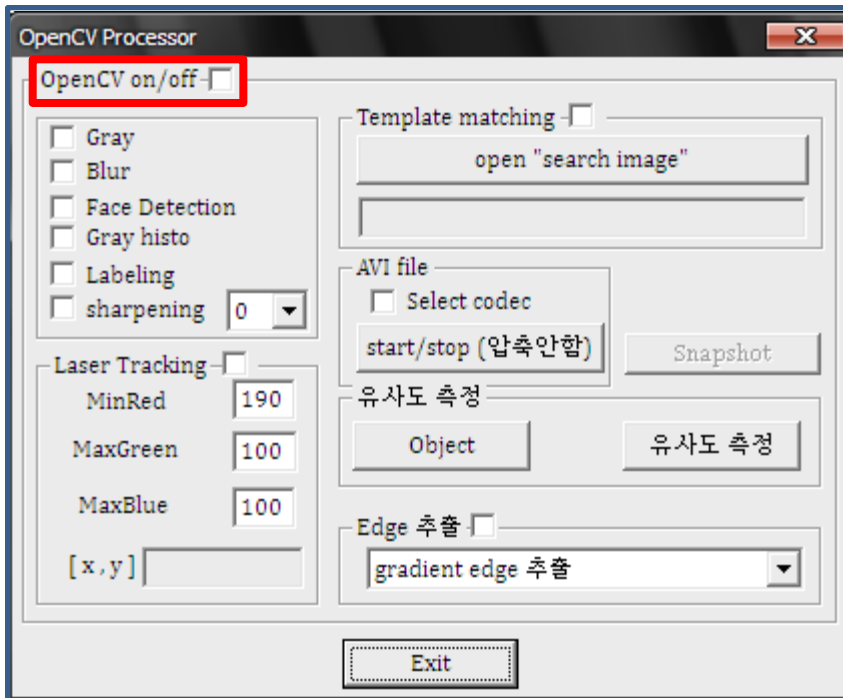
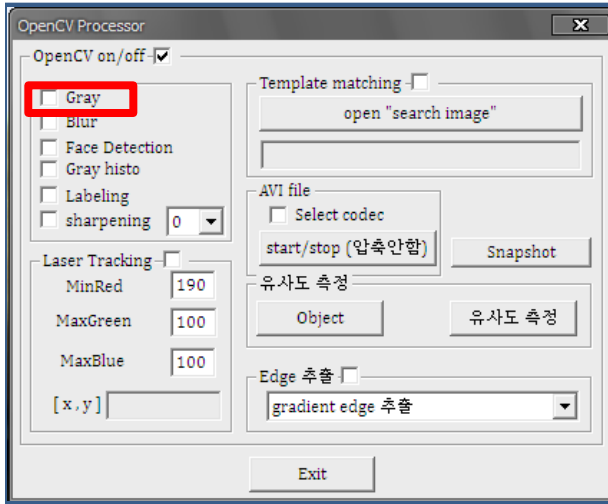


Fig 25. OpenCV Control Dialog – OpenCV on/off

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.2 Gray

영상을 회색영상으로 보여준다.

```
CVAPI(void) cvCvtColor(
const CvArr* src,
CvArr* dst,
int code );
```

Src : 원본영상

Dst : 출력영상

Code : 변형타입(BGR2GRAY)

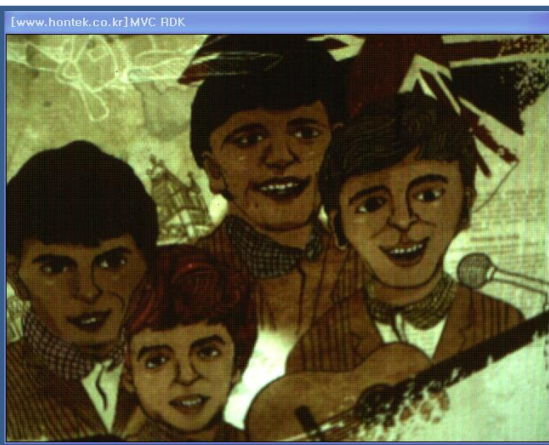
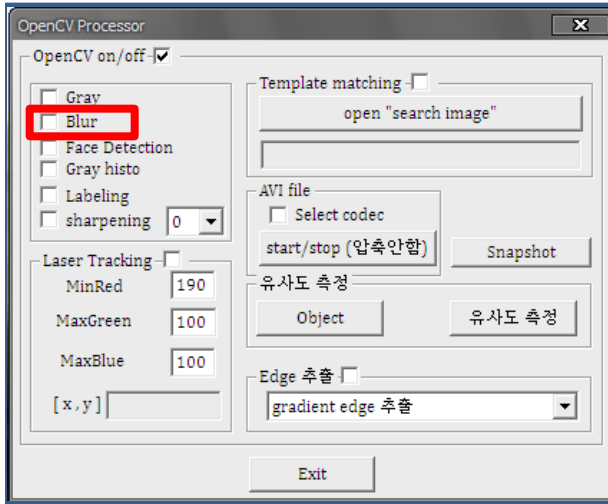


Fig 26. OpenCV Control Dialog - Gray

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.3 Blur

영상을 흐리게 나타낸다.

```
CVAPI(void) cvSmooth(
    const CvArr* src, // 입력 영상
    CvArr* dst, // 출력 영상
    int smoothtype, // 블러링 타입
    int param1, // 영상 너비
    int param2, // 영상 높이
);
```

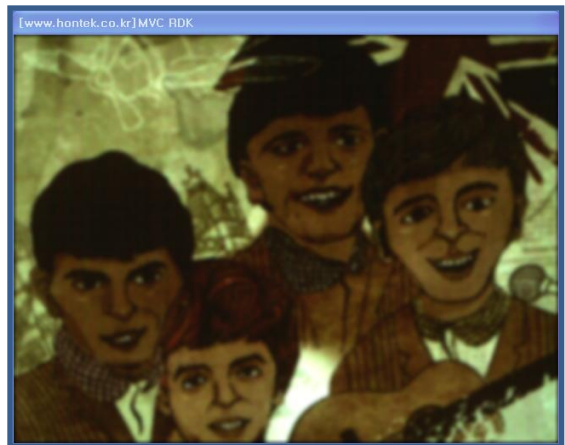
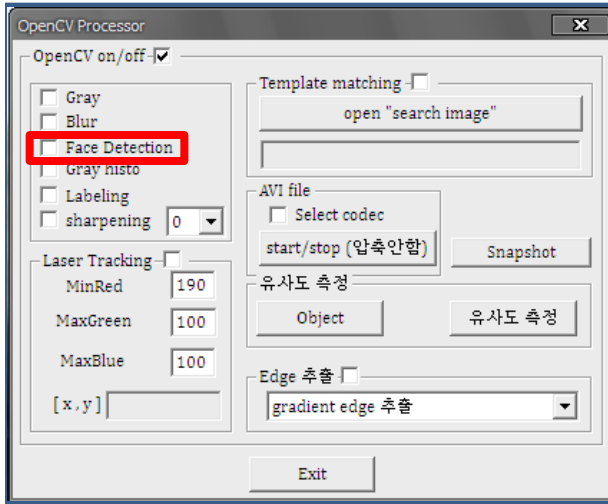


Fig 27. OpenCV Control Dialog - Blur

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.4 Face Detection

화면에서 얼굴을 찾아낸다.

IplImage *

MVC_OpenCV_FaceDetection(

IplImage* image);

Image : 원본 영상

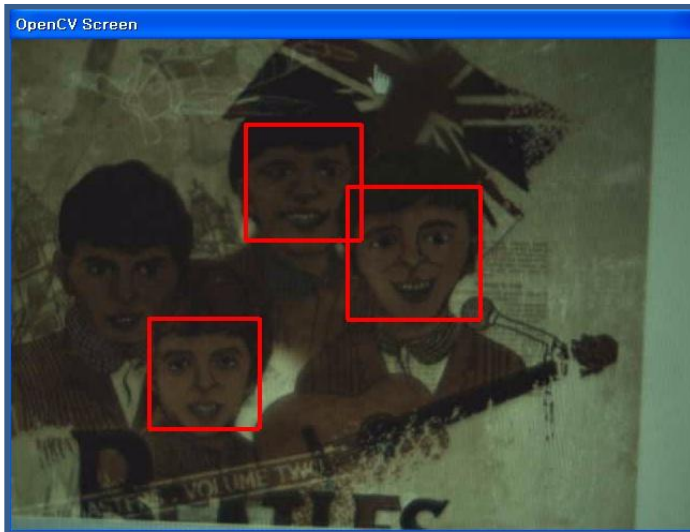
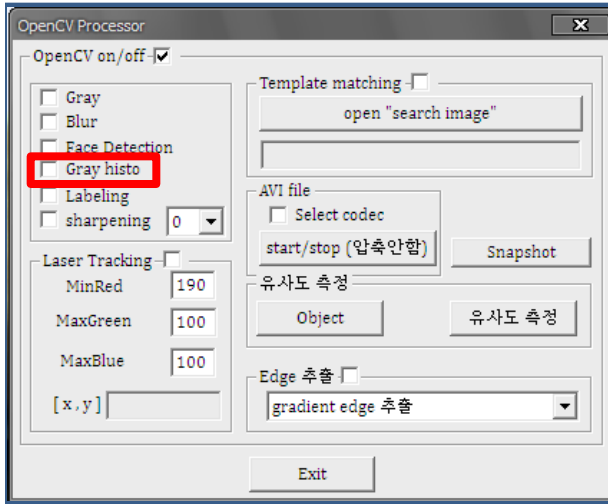


Fig 28. OpenCV Control Dialog – Face Detection

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.5 Gray Histo

영상의 밝기분포를 보여준다.

```
CV_INLINE void cvCalcHist(
    IplImage** image,
    CvHistogram* hist,
    int accumulate CV_DEFAULT(0),
    const CvArr* mask
    CV_DEFAULT(NULL) )
```

Image : 원본데이터

Hist : 히스토그램 데이터 구조체

Accumulate : 각 빈도 누적 여부

Mask : 마스크 영상

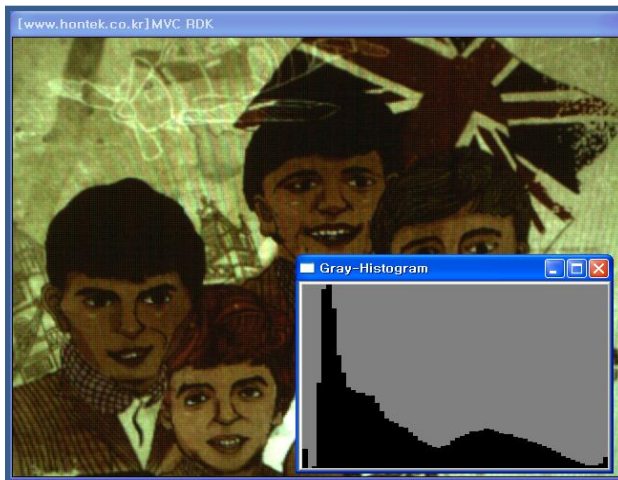
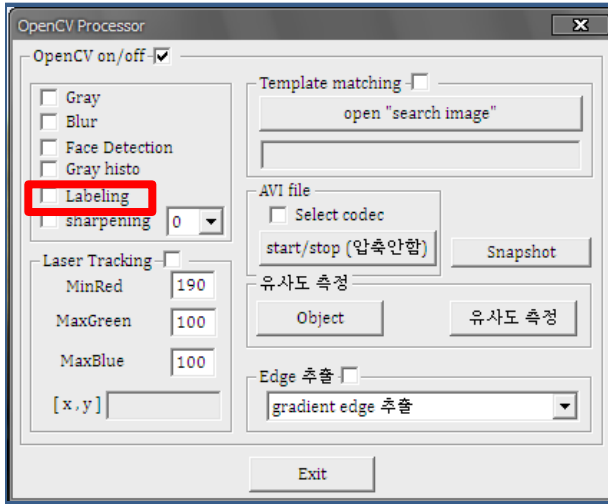


Fig 29. OpenCV Control Dialog – Gray Histo

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.6 Labeling

레이블링 기능

레이블링 : 인접한 화소에 모두 같은 번호(Label)를 붙이고 연결되지 않은 다른 성분에는 다른 번호를 붙이는 것

```
MVC_OpenCV_Labeling(
IplImage *imgIntP,
IplImage *imgGray);
```

imgIntP : 원본 영상 데이터
imgGray : 효과를 저장할 버퍼

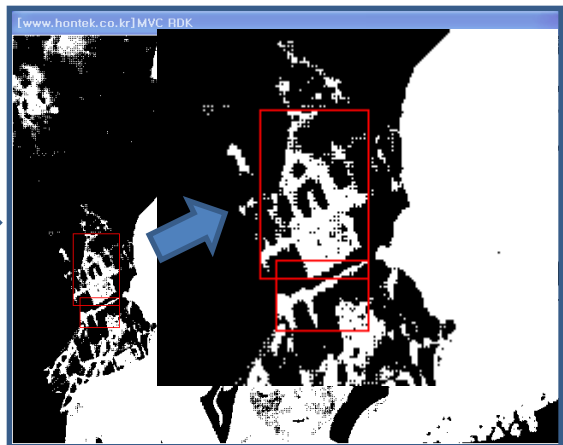
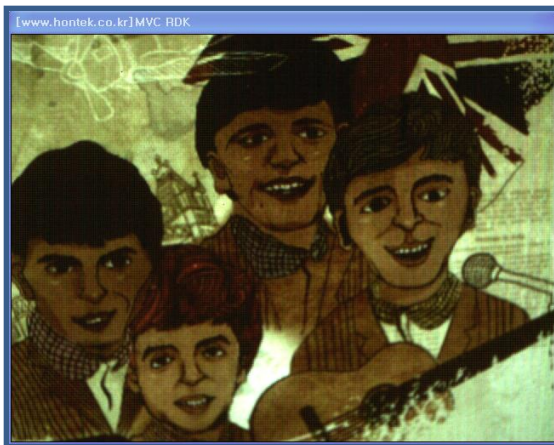
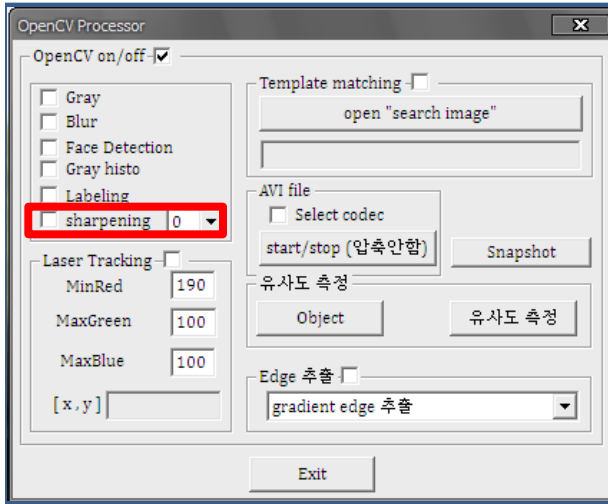


Fig 30. OpenCV Control Dialog - Labeling

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.7 Sharpening

영상을 선명하게 만든다.

```

IplImage*
MVC_OpenCV_Sharpening(
IplImage* imgIntP,
int m_sharpeninglevel);
    
```

imgIntP : 원본데이터

m_sharpeninglevel : level 0~5

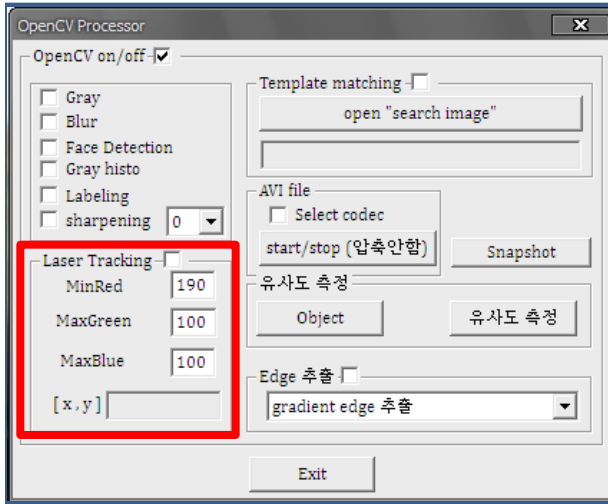


Fig 31. OpenCV Control Dialog - Sharpening

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.8 Laser Tracking

영상에서 붉은색을 찾는다.

```
char * __stdcall
MVC_OpenCV_convertRGB(
IplImage* img, // 원본 이미지
LPBYTE pBuf, // 결과 저장 버퍼
int iWidth, // 영상 너비
int iHeight, // 영상 높이
int minred, // Min Red Color
int maxgreen, // Max Green Color
int maxblue // Max blue Color
);
```

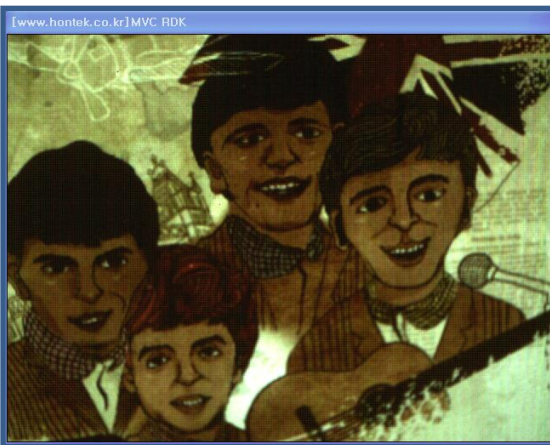
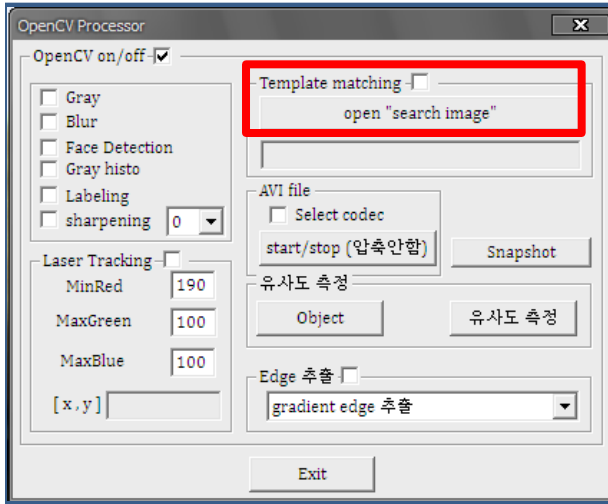


Fig 32. OpenCV Control Dialog – Laser Tracking

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.9 Template Matching

영상에서 특정영상을 찾는다.

```
CVAPI(void) cvMatchTemplate(
const CvArr* image,
const CvArr* templ,
CvArr* result,
int method );
```

- Image : 영상 데이터
- Templ : 검색할 영상
- Result : 결과 영상
- Method : 검출방법
- CV_TM_SQDIFF 0
- CV_TM_SQDIFF_NORMED 1
- CV_TM_CCORR 2
- CV_TM_CCORR_NORMED 3
- CV_TM_CCOEFF 4
- CV_TM_CCOEFF_NORMED 5

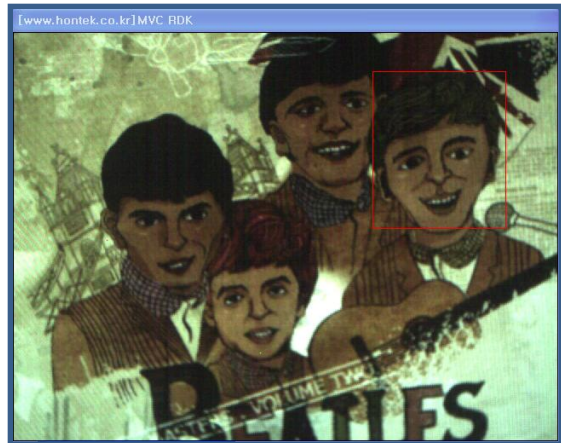
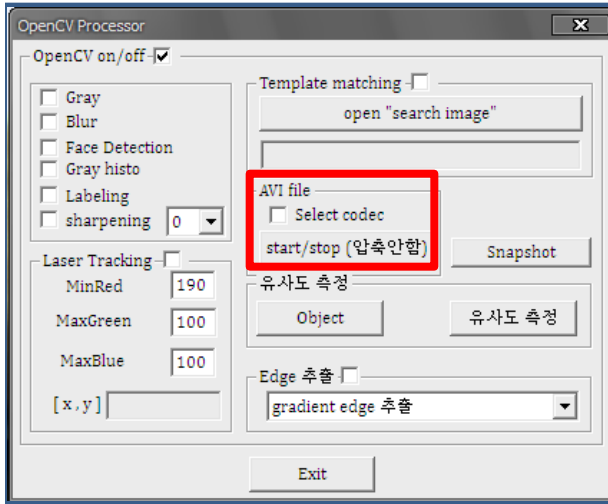


Fig 33. OpenCV Control Dialog – Template Matching

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.10 동영상 저장

동영상으로 저장한다.

```
// 저장시작
VideoOut =
cvCreateVideoWriter(
tmp,
-1,
value-5,
cvGetSize(imgGray),
1);
// 저장
cvWriteFrame(VideoOut, imgIntP);
// 저장 종료
cvReleaseVideoWriter(&VideoOut);
```

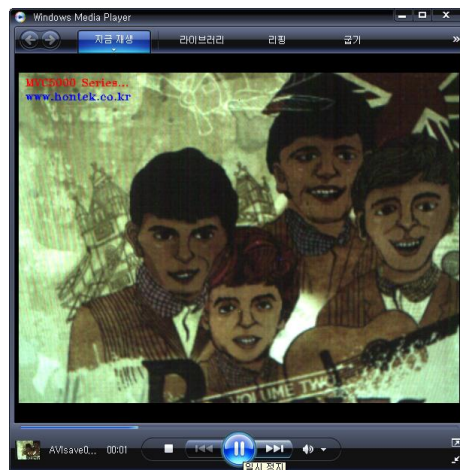
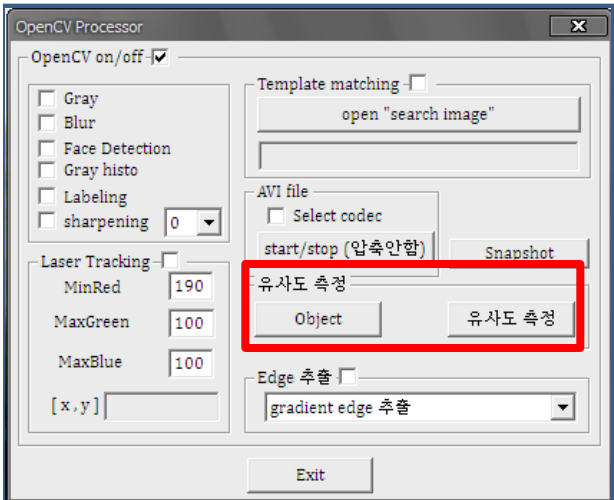


Fig 34. OpenCV Control Dialog – 동영상 저장

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.11 유사도 측정

두 개의 영상간에 유사도를 측정한다.

IplImage *

```
MVC_OpenCV_Match(
IplImage *src_image1,
IplImage *src_image2,
LPBYTE pBuffer);
```

src_image1 : 원본 영상 데이터
src_image2 : 비교영상 데이터
pBuf : 비교결과 스트링

Fig 35. OpenCV Control Dialog – 유사도 측정

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.

3.5.12 이미지 저장

OpenCV 적용 이미지를 저장한다.

이미지 이름은 입력된 값을 이용하며, 사용 가능한 확장자가 없을 때는 Jpg 라는 확장자를 추가하여 저장한다.

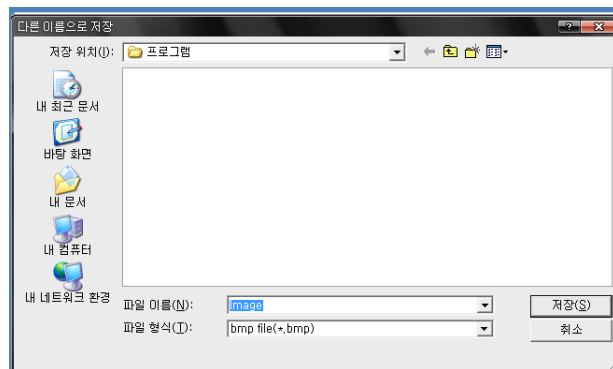
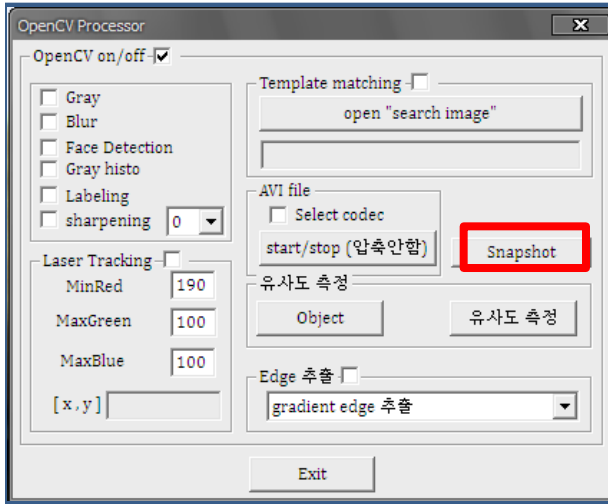
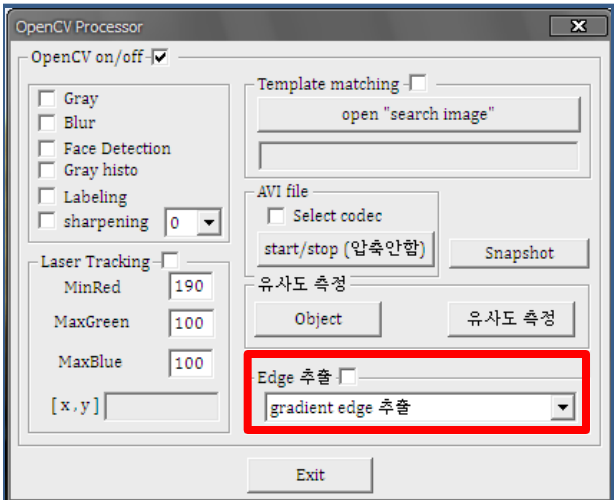


Fig 36. OpenCV Control Dialog – 이미지 저장

3. Software Architecture

3.5 OpenCV Control Dialog

OpenCV 라이브러리를 이용한 예.



3.5.12 Edge 추출

윤곽선을 추출한다.

```

IplImage * MVC_OpenCV_Edge(
IplImage *imgCanny,
IplImage *imgGray,
int m_seledge);
        
```

imgCanny : 원본 영상 버퍼
imgGray : 수정된 영상을 담을 버퍼
m_seledge : Edge검출 방법




➔


Fig 37. OpenCV Control Dialog – Edge 추출

4. System File Description

4.1 MVC_VISION.ini

실행파일과 같은 디렉토리에 위치하며, 초기 설정들의 기본값을 File Read/Write 방식으로 관리

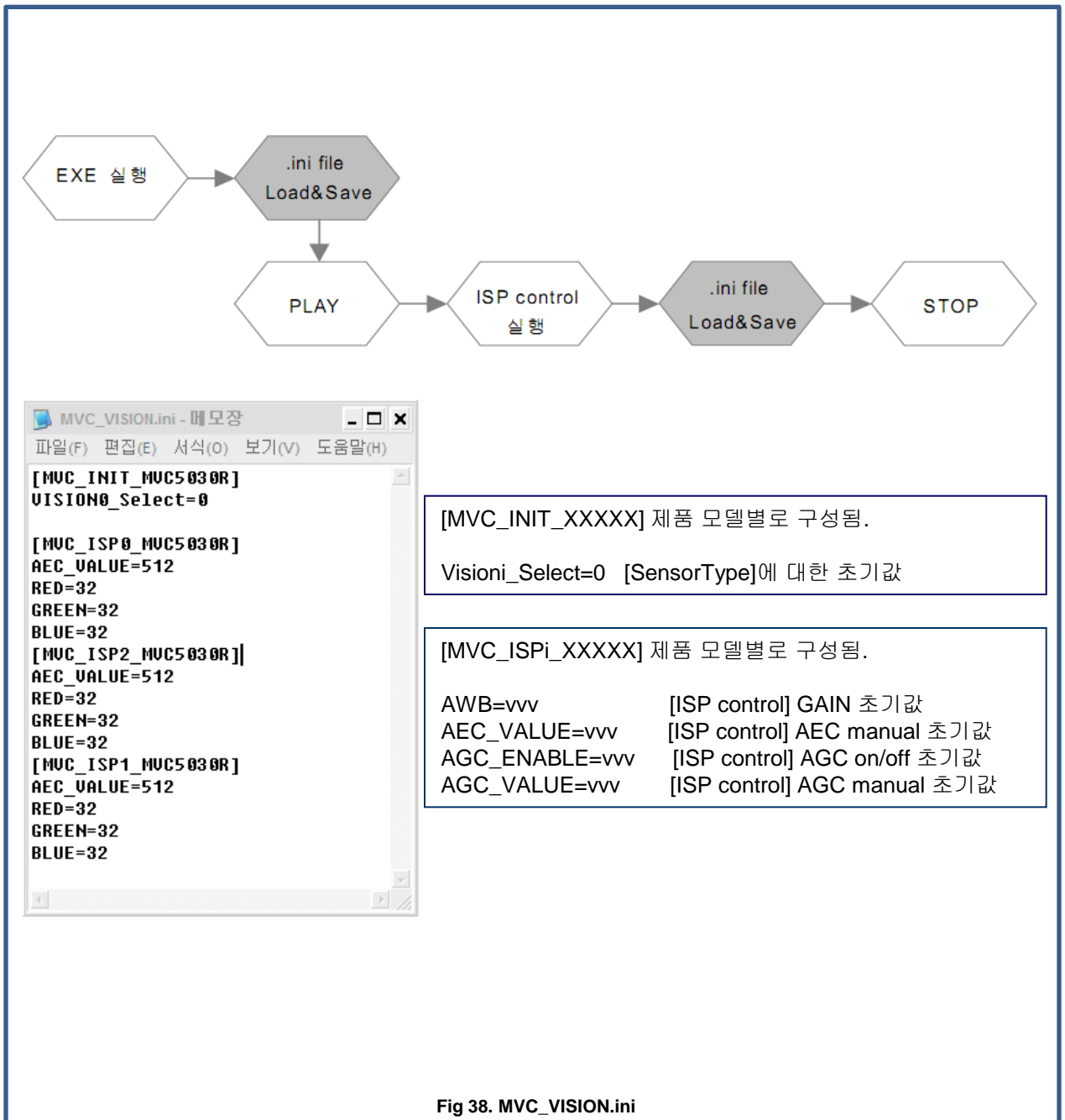


Fig 38. MVC_VISION.ini

5. API Lists

5.1 I2C Communication

API	내용
MVC_Init	카메라와 I2C통신을 시작한다
MVC_Release	카메라와 I2C통신을 종료한다.
MVC_EmpiaispAutoGainControlEnable	Auto GAIN Control 기능 사용 유무 설정
MVC_EmpiaispGetAWBGainControlRange	Gain 값과 최대 최소값을 얻는다..
MVC_EmpiaispSetAWBGainControl	Gain 값을 설정한다.
MVC_EmpiaispSetExposureTime	Exposure Time(빛을 받아들이는 시간)을 설정한다.
MVC_EmpiaispGetExposureRange	Exposure Time(빛을 받아들이는 시간)의 최대, 최소값, 설정값을 얻는다.
IIC_Read16i	I2C를 이용해서 데이터를 읽는다
IIC_Write16i	I2C를 이용해서 데이터를 쓴다.

5. API Lists

5.2 OpenCV 관련 함수

API	내용
MVC_OpenCV_Create	OpenCV 프로세스 시작.
MVC_OpenCV_Release	OpenCV 프로세스 종료.
MVC_OpenCV_Display	OpenCV 출력
MVC_OpenCV_StartAVI	동영상 저장 시작.
MVC_OpenCV_StopAVI	동영상 저장 종료
MVC_OpenCV_TM_targetFile	Template Matching 할 대상파일 설정
MVC_OpenCV_LoadObjectOpenCV	유사도 측정을 위한 대상 화면 설정
MVC_OpenCV_MatchingObjectOpenCV	유사도 측정
MVC_OpenCV_LaserTracking	Laser Tracking 동작
MVC_OpenCV_SelEdge	Edge 검출 수행
MVC_OpenCV_Labeling	레이블링 기능 수행
MVC_OpenCV_Sharpenering	Sharpening 기능 수행

6. API 설명

6.1 MVC_Init

카메라와 I2C통신을 시작한다

구분	내용
Syntax	Int HVR_Init ()
Parameter	없음
Return Value	성공 : 1, 실패 : 0
Sample	Int m_etLib2_Init = HVR_Init();

6. API 설명

6.2 MVC_Release

카메라와 I2C통신을 종료한다.

구분	내용
Syntax	Int HVR_Release()
Parameter	없음
Return Value	성공 : 1, 실패 : 0
Sample	<pre> Int m_etLib2_Init = HVR_Init(); ... if (m_etLib2_Init) HVR_Release(); </pre>

6. API 설명

6.3 MVC_EmpiaispAutoGainControlEnable

카메라와 I2C통신을 시작한다.

구분	내용	
Syntax	int HVR_EmpiaispAutoGainControlEnable(WCHAR *Serial_Number, int Value);	
Parameter	Serial_Number	카메라 Serial_number
	Value	AGC_ENABLE:사용, AGC_DISABLE:미사용
Return Value	성공 : 1, 실패 : 0	
Sample	<pre> WCHAR Serial_Number[5]=L"1001"; Int m_etLib2_Init = HVR_Init(); HVR_EmpiaispAutoGainControlEnable(Serial_Number, AGC_ENABLE); HVR_Release(); </pre>	

6. API 설명

6.4 MVC_EmpiaispGetAWBGainControlRange

Gain값과 최대 최소, 현재값을 얻는다...

구분	내용	
Syntax	Int HVR_EmpiaispGetAWBGainControlRange(WCHAR *Serial_Number, int *pMin, int *pMax, int *pValue);	
Parameter	Serial_Number	카메라 Serial_number
	pMin	GAIN 최대값
	pMax	GAIN 최소값
	pValue	GAIN 현재값
Return Value	성공 : 1, 실패 : 0	
Sample	<pre> WCHAR Serial_Number[5]=L"1001"; Int iMin, iMax, iWB_R_E; Int m_etLib2_Init = HVR_Init(); HVR_EmpiaispGetAWBGainControlRange(Serial_Number, &iMin, &iMax, &iWB_R_E); HVR_Release(); </pre>	

6. API 설명

6.5 MVC_EmpiaispSetAWBGainControl

GAIN 값을 설정한다..

구분	내용	
Syntax	Int HVR_EmpiaispSetAWBGainControl(WCHAR *Serial_Number, int iGain);	
Parameter	Serial_Number	카메라 Serial_number
	Value	Exposure Time 설정값
Return Value	성공 : 1, 실패 : 0	
Sample	<pre> WCHAR Serial_Number[5]=L"1001"; Int m_etLib2_Init = HVR_Init(); VR_EmpiaispSetAWBGainControl(Serial_Number, 300); HVR_Release(); </pre>	

6. API 설명

6.6 MVC_EmpiaispGetExposureRange

Exposure Time(빛을 받아들이는 시간)의 최대, 최소값, 설정값을 얻는다

구분	내용	
Syntax	int HVR_EmpiaispGetExposureRange(WCHAR *Serial_Number, int *pMin, int *pMax, int *pValue);	
Parameter	Serial_Number	카메라 Serial_number
	pMin	Exposure Time 최대값
	pMax	Exposure Time 최소값
	pValue	Exposure Time 현재값
Return Value	성공 : 1, 실패 : 0	
Sample	<pre> WCHAR Serial_Number[5]=L"1001"; Int iMin, iMax, iEC_E; Int m_etLib2_Init = HVR_Init(); HVR_EmpiaispGetExposureRange(Serial_Number, &iMin, &iMax, &iEC_E); HVR_Release(); </pre>	

6. API 설명

6.7 MVC_EmpiaispSetExposureTime

Exposure Time(빛을 받아들이는 시간)을 설정한다...

구분	내용	
Syntax	Int HVR_EmpiaispSetExposureTime(WCHAR *Serial_Number, int Value);	
Parameter	Serial_Number	카메라 Serial_number
	Value	Exposure Time 설정값
Return Value	성공 : 1, 실패 : 0	
Sample	<pre> WCHAR Serial_Number[5]=L"1001"; Int m_etLib2_Init = HVR_Init(); HVR_EmpiaispSetExposureTime(Serial_Number, 20); HVR_Release(); </pre>	

6. API 설명

6.8 IIC_Read16i

I2C를 이용해서 데이터를 읽는다

구분	내용	
Syntax	ULONG IIC_Read16i(WCHAR *Serial_Number, WORD wRegAddr);	
Parameter	Serial_Number	카메라 Serial_number
	wRegAddr	어드레스
Return Value	레지스트리로 부터 읽어온 데이터	
Sample	<pre> WCHAR Serial_Number[5]=L"1001"; ULONG I2cAddr = 0; Int m_etLib2_Init = HVR_Init(); I2cValue = IIC_Read16i(Serial_Number,0); HVR_Release(); </pre>	

6. API 설명

6.9 IIC_Write16i

I2C를 이용해서 데이터를 쓴다

구분	내용	
Syntax	BOOL IIC_Write16i(WCHAR *Serial_Number, WORD wRegAddr, WORD wRegData);	
Parameter	Serial_Number	카메라 Serial_number
	wRegAddr	어드레스
	wRegData	데이터
Return Value	성공 : 1, 실패 : 0	
Sample	<pre> WCHAR Serial_Number[5]=L"1001"; ULONG I2cAddr = 0; Int m_etLib2_Init = HVR_Init(); IIC_Write16i (Serial_Number, 0xb, 10); HVR_Release(); </pre>	

6. API 설명

6.10 MVC_OpenCV_Create

OpenCV 프로세스 시작.

구분	내용
Syntax	Int HVR_OpenCV_Create();
Parameter	
Return Value	성공 : 1
Sample	<pre> BOOL gAVI = 0, gGRAYhisto = 0; HVR_OpenCV_Create(); HVR_OpenCV_Release(gAVI, gGRAYhisto); </pre>

6. API 설명

6.11 MVC_OpenCV_Release

OpenCV 프로세스 종료.

구분	내용	
Syntax	Int HVR_OpenCV_Release(BOOL AVI, BOOL GRAYhisto);	
Parameter	AVI	현재 동영상 저장 유무
	GRAYhisto	현재 Grayhist 설정 유무
Return Value	성공 : 1	
Sample	<pre> BOOL gAVI = 0, gGRAYhisto = 0; HVR_OpenCV_Create(); HVR_OpenCV_Release(gAVI, gGRAYhisto); </pre>	

6. API 설명

6.12 MVC_OpenCV_Display

OpenCV 출력.

구분	내용	
Syntax	Int HVR_OpenCV_Display(BYTE *pBuffer,long lBufferLen, LPBYTE pBuf,BOOL gGRAY,BOOL gBLUR,BOOL gGRAYhisto,BOOL gCANNY,BOOL gFACEDETECTION,BOOL gTMATCHING,BOOL gAVI,BOOL bTargetFile,BOOL bLaserSearch,HWND ghwndApp);	
Parameter	pBuffer,	출력할 영상 데이터
	lBufferLen	영상 데이터 길이
	pBuf	효과를 적용한 영상데이터를 임시로 저장하기 위한 버퍼
	gGRAY	흑백효과 사용 유무
	gBLUR	Blur 효과 사용 유무
	gGRAYhisto	Gray histogram 사용 유무
	gCANNY	Edge 기능 사용 유무
	gFACEDETECTION	얼굴 검출 기능 사용 유무
	gTMATCHING	Template Matching 사용 유무
	gAVI	동영상 저장 상태
	bTargetFile	Template Matching 을 위한 비교파일 설정 유무
	bLaserSearch	Laser Tracking 기능 사용 유무
	ghwndApp	효과를 적용한 영상데이터를 출력할 화면의 핸들
Return Value	성공 : 1	
Sample	HVR_OpenCV_Create(); HVR_OpenCV_Display(pBuffer, lBufferLen, pBuf, gGRAY, gBLUR, gGRAYhisto, gCANNY, gFACEDETECTION, gTMATCHING, gAVI, bTargetFile, gLaser, ghwndApp); HVR_OpenCV_Release(gAVI, gGRAYhisto);	

6. API 설명

6.13 MVC_OpenCV_StartAVI

동영상 저장 시작.

구분	내용	
Syntax	Void HVR_OpenCV_StartAVI(bool mode, int value);	
Parameter	mode	Codec 사용 유무(1: 사용, 2:미사용)
	value	FrameRate
Return Value		
Sample	<pre>HVR_OpenCV_StartAVI(0, 20); ... HVR_OpenCV_StopAVI();</pre>	

6. API 설명

6.14 MVC_OpenCV_StopAVI

동영상 저장 종료.

구분	내용
Syntax	void HVR_OpenCV_StopAVI();
Parameter	
Return Value	
Sample	HVR_OpenCV_StartAVI(0, 20); ... HVR_OpenCV_StopAVI();

6. API 설명

6.15 MVC_OpenCV_TM_targetFile

Template Matching 할 대상파일 설정

구분	내용	
Syntax	BOOL HVR_OpenCV_TM_targetFile(LPCTSTR filepath);	
Parameter	filepath	비교할 대상 파일이름
Return Value	성공 : 1	
Sample	if(HVR_OpenCV_TM_targetFile(targetfileName)) bTargetFile=TRUE;	

6. API 설명

6.16 MVC_OpenCV_LoadObjectOpenCV

유사도 측정을 위한 대상 화면 설정

구분	내용
Syntax	Void HVR_OpenCV_LoadObjectOpenCV()
Parameter	
Return Value	
Sample	HVR_OpenCV_LoadObjectOpenCV(); ... HVR_OpenCV_MatchingObjectOpenCV();

6. API 설명

6.17 MVC_OpenCV_MatchingObjectOpenCV

유사도 측정

구분	내용
Syntax	Void HVR_OpenCV_MatchingObjectOpenCV();
Parameter	
Return Value	
Sample	HVR_OpenCV_LoadObjectOpenCV(); ... HVR_OpenCV_MatchingObjectOpenCV();

6. API 설명

6.18 MVC_OpenCV_LaserTracking

Laser Tracking 동작

구분	내용	
Syntax	Void HVR_OpenCV_LaserTracking(int red, int green, int blue);	
Parameter	Red	탐지할 위치의 red 최소값(0~255)
	Green	탐지할 위치의 green 최대값(0~255)
	Blue	탐지할 위치의 blue 최대값(0~255)
Return Value		
Sample	HVR_OpenCV_LaserTracking(m_red, m_green, m_blue);	

6. API 설명

6.19 MVC_OpenCV_SelEdge

Edge 검출

구분	내용	
Syntax	Void HVR_OpenCV_SelEdge(int no);	
Parameter	no	알고리즘 0 : 그래디언트 에지 추출 1 : 영역 필터 에지 추출 2~5 : 소벨 에지 추출 6~8 : 프리윗 에지 추출 7 : 프레이첸 에지 추출 8: 서터캐스틱 에지 추출 9 : 로버츠 에지 추출 10~11 : 라플라시안 에지 추출 12 : 라플라시안 샤프닝 13~14 : LoG 에지 추출 15~17 : 캐니 에지 추출
Return Value		
Sample	HVR_OpenCV_SelEdge(15);	

6. API 설명

6.20 MVC_OpenCV_Labeling

OpenCV 라이브러리 레이블링 기능 수행

구분	내용	
Syntax	void HVR_OpenCV_Labeling(int state);	
Parameter	state	1: 적용, 0 : 미적용
Return Value		
Sample	HVR_OpenCV_Labeling(gLabeling);	

6. API 설명

6.21 MVC_OpenCV_Sharpener

OpenCV 라이브러리 Sharpening 기능 수행

구분	내용	
Syntax	Void HVR_OpenCV_Sharpener(bool state, int level);	
Parameter	state	기능 사용 유무(1:사용, 0: 미사용)
	level	Level(0 ~ 5)
Return Value	Sharpening 영상 구조체	
Sample	HVR_OpenCV_Sharpener(gSharpening, m_Sharpening.GetCurSel());	

7. 기타 문의 사항

본 문서는 MVC-5030R 비전카메라를 이용하기 위한 사용자의 이해를 돕기 위해 만든 문서로 하드웨어 구성, 소프트웨어 구성, 프로그램 개발에 대한 설명을 나열 하였습니다. 기타 추가적인 질문사항이 있으시면 저희 웹 카페 [혼텍 제품 자료실] → [제품 문의] 에 글을 남겨 주시거나, 아래의 연락처로 문의하시기 바랍니다.

혼 텍

주소 : 경기도 성남시 중원구 상대원동 190-1 SK@테크노파크 테크동 810호

전화 : 031-776-3897, 3899

팩스 : 031-776-3898

홈페이지 : <http://www.hontek.co.kr>

E-Mail : hontek@hontek.co.kr