Introduction to OpenCV

ISPL

목표

- 자신의 컴퓨터 환경에 맞게 올바르게 OpenCV 설치
- 간단한 예제들을 통하여 OpenCV 익히기

OpenCV 소개

- OpenCV
  - 의미: Open Source Computer Vision Library
  - 500여개 이상의 함수와 알고리즘들
    - Computer vision
    - Image processing
    - Math library
    - 이외 일반적인 목적의 함수들
  - 효율적이며 가볍다 (C/C++)
  - Absolutely free!
- 목표
  - 비전 응용프로그램 개발을 위한 **공통 기반 구조**를 제공하여 비전 지식을 보급
  - 이식성이 좋고, 성능이 최적화된 소스 코드를 공개하여 비전 기반 **상업용 응용프로그램**의 발전

OpenCV 소개

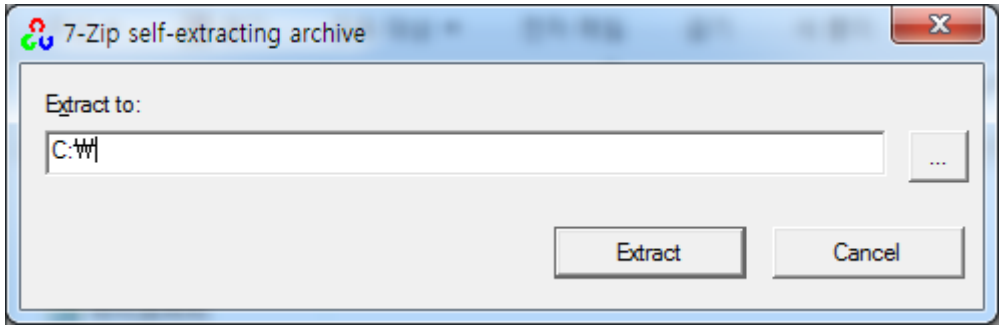
- 1999년부터 Intel 사가 개발
  - http://www.intel.com/technology/computing/opencv/index.htm
- Available on Windows, Linux and MacOSX.
- 많은 회사들과 리서치 센터 등에서 폭넓게 사용 중
- 현재 최신 버전 : 2.4.9

OpenCV 소개

- 주된 활용 분야
  - Computer Human Interaction (HCI)
  - Segmentation and Recognition
  - Face Recognition
  - Gesture Recognition
  - Motion Tracking
  - Motion Understanding
  - And so on

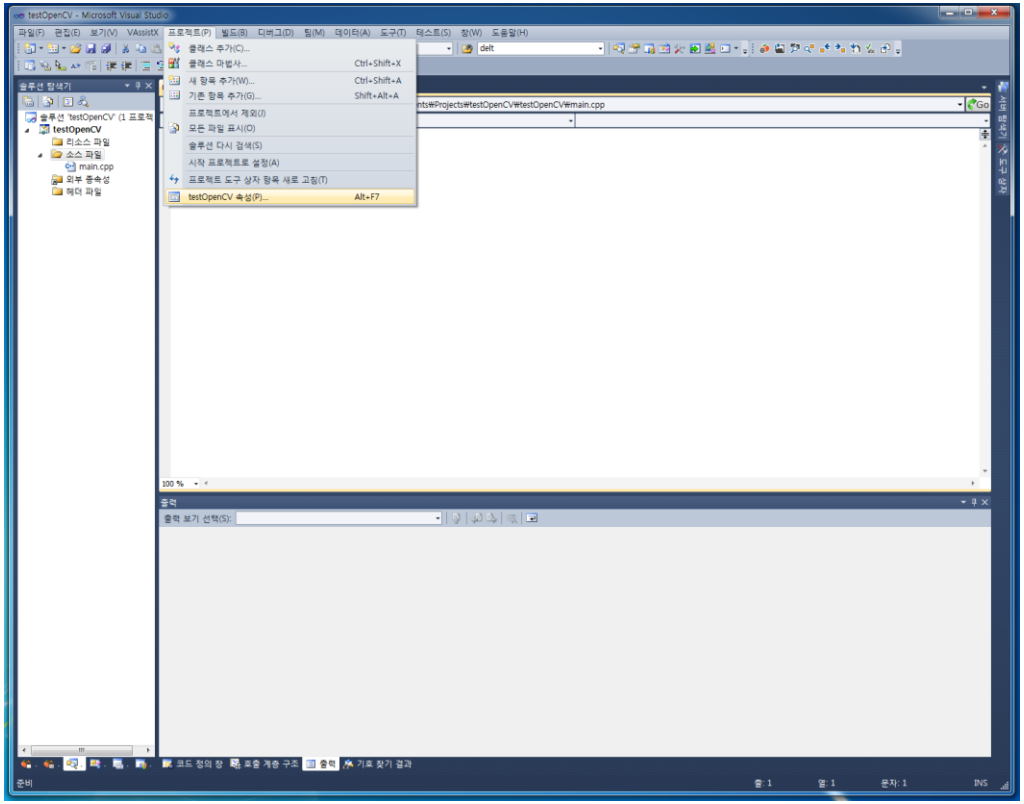
OpenCV 설치 (1) 다운로드

- OpenCV 다운 & 설치
  - http://sourceforge.net/projects/opencvlibrary/
  - 자신의 운영체제에 맞는 최신 version을 다운받음
  - Windows7 32bits, vc10(visual studio 2010), OpenCV 2.1 기준 (다른 사람들과의 호환성을 위해 visual studion 2010 에서 32bits로 설정하여 모두 통일)
- 다운 후 원하는 곳에 설치 경로를 지정



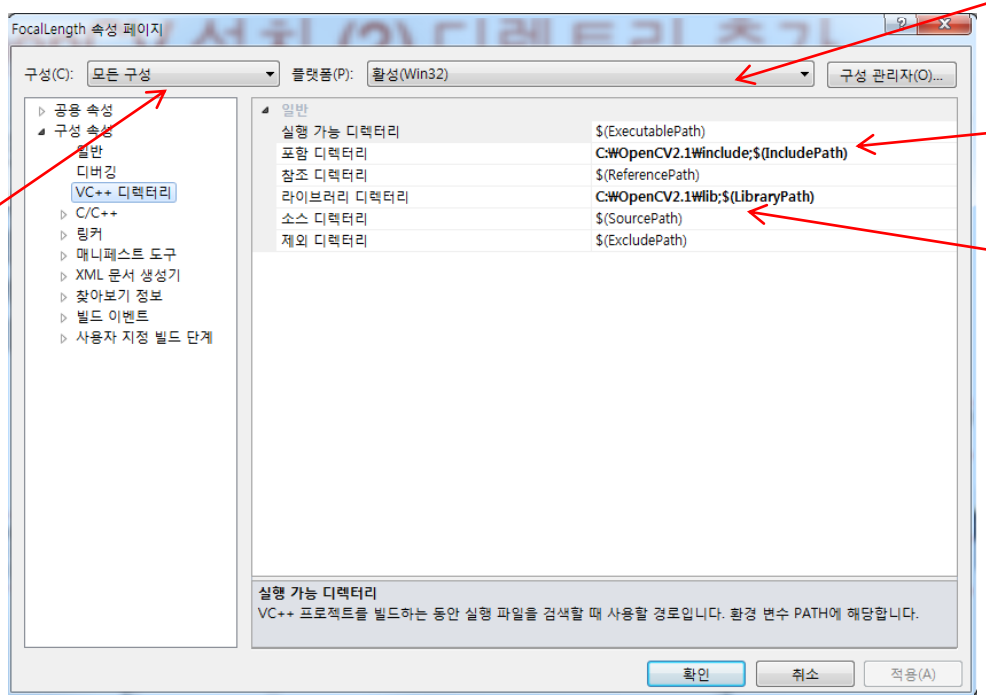
OpenCV 설치 (2) 디렉토리 추가

- 프로젝트 생성 및 설정
  - 1. 새 프로젝트 생성
  - 2. Win32콘솔 프로젝트 선택
  - 3. 빈 프로젝트 체크
  - 4. 프로젝트 – 속성 선택



OpenCV 설치 (2) 디렉토리 추가

- 매 프로젝트 마다 프로젝트 설정 필요
  - VC++ 디렉터리 – 포함 디렉터리
    : [OpenCV 설치 폴더]\include 추가
  - VC++ 디렉터리 – 라이브러리 디렉터리
    : [OpenCV 설치 폴더]\lib 추가



Window 32 bits 설정(64의 경우 활성(x64))

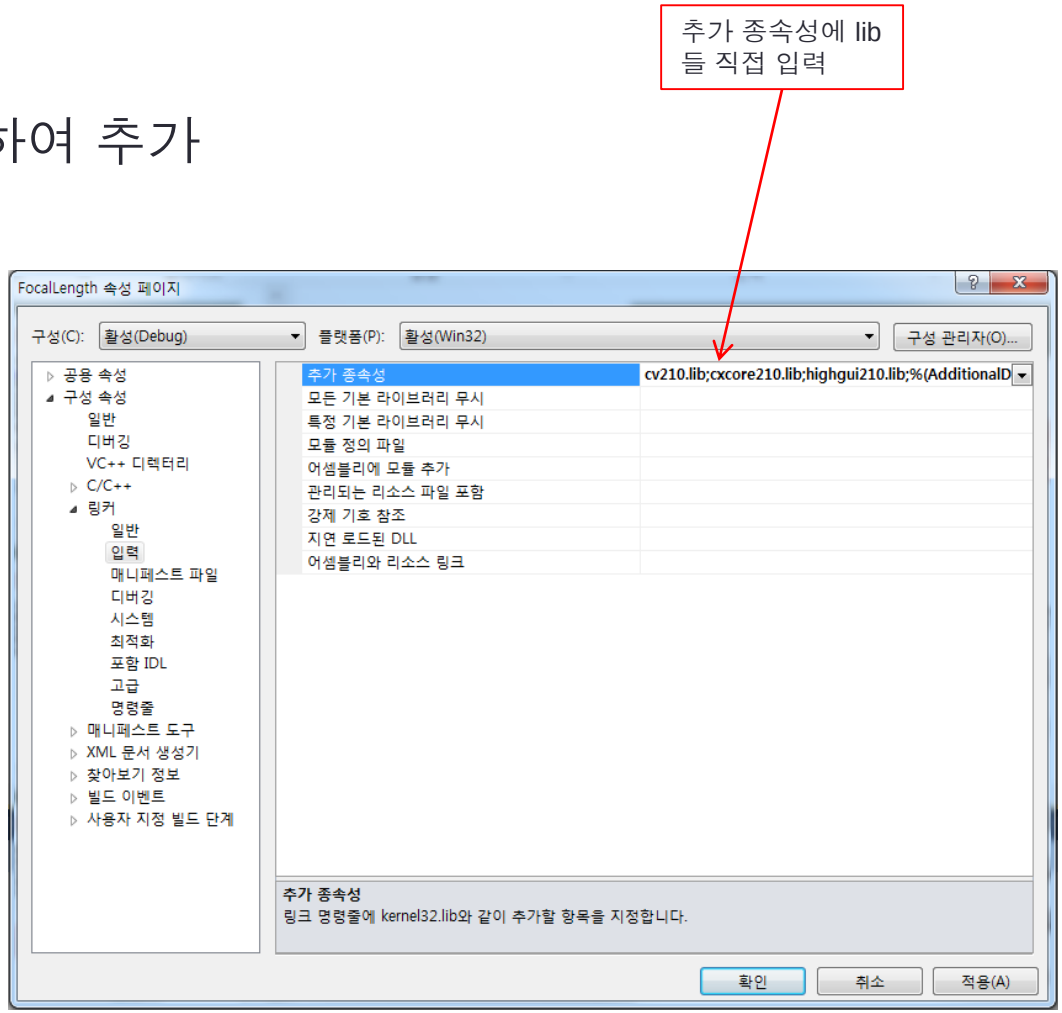
포함 디렉터리 설정

Debug, Release 모드 모두 포함하는 모든 구성으로

라이브러리 디렉터리 설정

OpenCV 설치 (3) 링크

- Library 추가
  - 링커 – 입력 – 추가 종속성
  - 다음과 같은 lib 직접 입력하여 추가
  - Debug 모드 기준
    - cv210.lib
      cxcore210.lib
      highgui210.lib



OpenCV 설치 (4) dll 파일복사

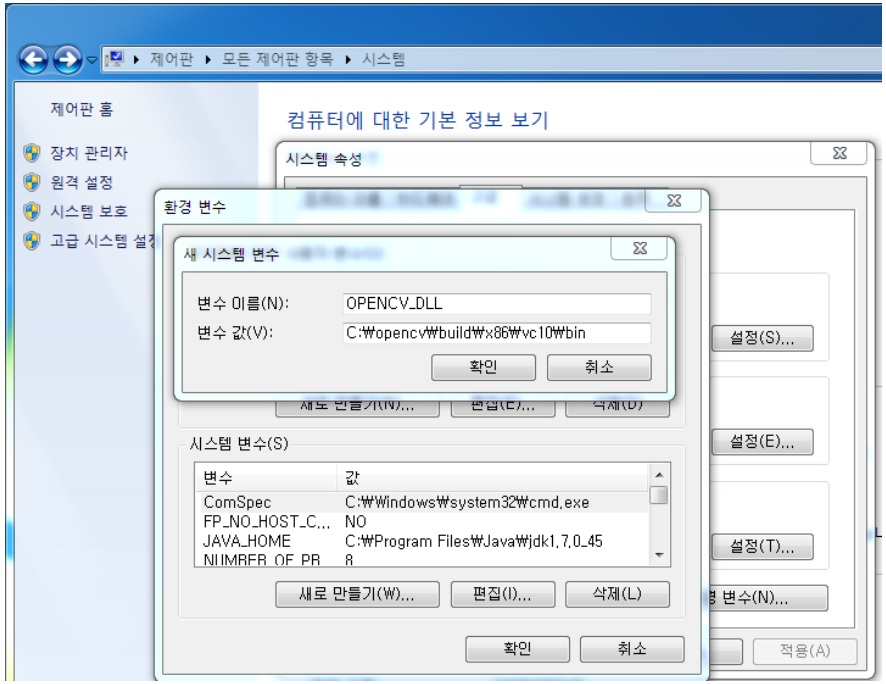
- 프로젝트 내부 main.cpp 있는 폴더에 dll 직접 붙임
  - [OpenCV 설치 폴더]bin
  - 폴더 안의 dll 파일들 모두를 “windows” 폴더에 복사

OpenCV 설치 (5) include

- 주로 쓰이는 헤더파일
  - #include <opencv\cv.h>
  - #include <opencv\cxcore.h>
  - #include <opencv\highgui.h>

OpenCV 설치 (6) Else 방법

- 매번 dll 파일을 복사하기 귀찮다면 시스템 변수에 PATH를 추가하는 방법을 사용
  - http://www.youtube.com/watch?v=cgo0UitHfp8
  - 제어판 – 모든 제어판 항목 – 시스템 – 고급 시스템 설정 – 환경변수 – 시스템 변수 – 새로 만들기



OpenCV 설치

- Summary

  1. OpenCV을 다운

  2. Visual studio 새 프로젝트를 만들 때 마다 다음과 같은 과정을 반복해줌

     1) 프로젝트 – 속성 – VC++ 디렉터리 – 포함 디렉터리

     2) 프로젝트 – 속성 – VC++ 디렉터리 – 라이브러리 디렉터리

     3) 프로젝트 – 속성 – 링커 – 입력 – 추가종속성

기본 데이터 타입

- CvPoint
  - 이미지에서의 x, y 좌표를 나타낼 때
- CvSize
  - width, height를 이용하여 이미지의 크기를 표현할 때
- CvRect
  - 이미지 내의 특정 사각형 공간을 정의할 때
- CvScalar
  - RGBA 색상을 표현하기 위해 사용

Table 3-1. Structures for points, size, rectangles, and scalar tuples

Structure	Contains	Represents
CvPoint	int x, y	Point in image
CvPoint2D32f	float x, y	Points in $\Re^2$
CvPoint3D32f	float x, y, z	Points in $\Re^3$
CvSize	int width, height	Size of image
CvRect	int x, y, width, height	Portion of image
CvScalar	double val[4]	RGBA value

기본 데이터 타입

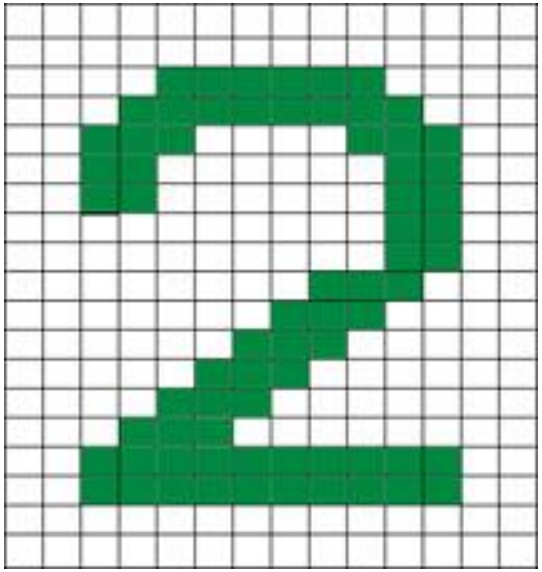
- 예제

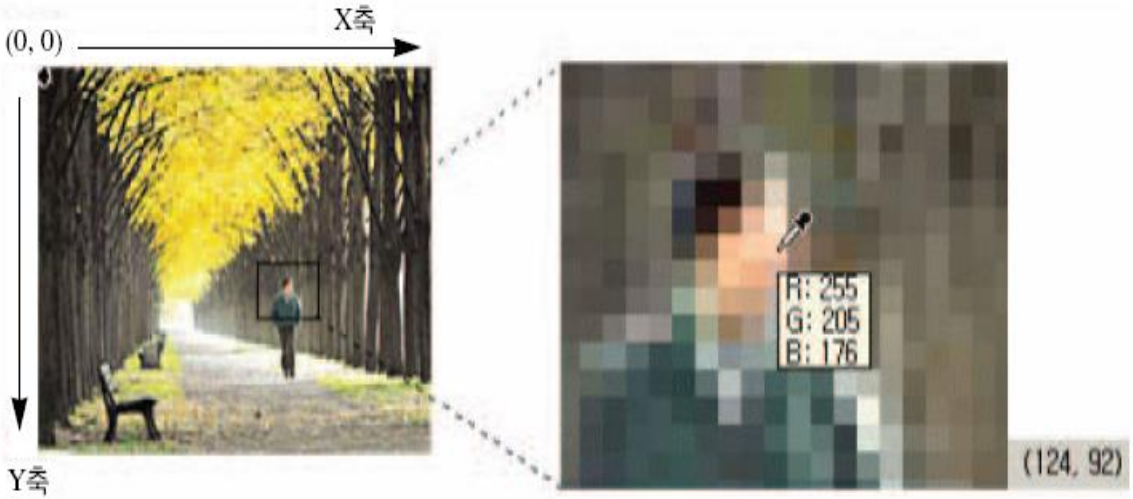
```
#include <opencv₩cv.h>
#include <opencv₩cxcore.h>
#include <opencv₩highgui.h>

int main()
{
    CvPoint point = cvPoint(5,5);
    CvSize size = cvSize(640, 480);
    CvRect rect = cvRect(10, 30, 5, 5);
    CvScalar scalar = cvScalar(255, 255, 255);

    printf("==CvPoint==₩n");
    printf("x = %d, y = %d₩n", point.x, point.y);
    printf("==CvSize==₩n");
    printf("width = %d, hight = %d₩n", size.width, size.height);
    printf("==CvRect==₩n");
    printf("x = %d, y = %d, width = %d, hight = %d₩n", rect.x, rect.y, rect.width, rect.height);
    printf("==CvScalar==₩n");
    printf("val[0] = %lf, val[1] = %lf, val[2] = %lf, val[3] = %lf₩n",
        scalar.val[0],
        scalar.val[1],
        scalar.val[2],
        scalar.val[3]);

    return 0;


}
```

```
C:₩Windows₩system32₩cmd.exe
==CvPoint==
x = 5, y = 5
==CvSize==
width = 640, hight = 480
==CvRect==
x = 10, y = 30, width = 5, hight = 5
==CvScalar==
val[0] = 255.000000, val[1] = 255.000000, val[2] = 255.000000, val[3] = 0.000000

계속하려면 아무 키나 누르십시오 . . .
```

행렬과 영상 타입

- 이미지 : 픽셀의 집합
- 보통 행렬로 표현





행렬과 영상 타입

- **IplImage**
  - 영상을 표현하는 데이터 타입
  - 그레이영상, 컬러영상, 4채널 영상(RGB+알파)를 표현 가능
  - 각 채널은 정수, 실수형 데이터를 저장 할 수 있음
- **IplImage 구조**
  - 객체 지향적 구조 구성
  - CvArr는 CvMat의 추상 기반 클래스, IplImage는 CvMat를 상속
  - CvArr* 타입이 사용된 자리에는 CvMat* 또는 IplImage* 타입을 대체해서 사용 할 수 있음

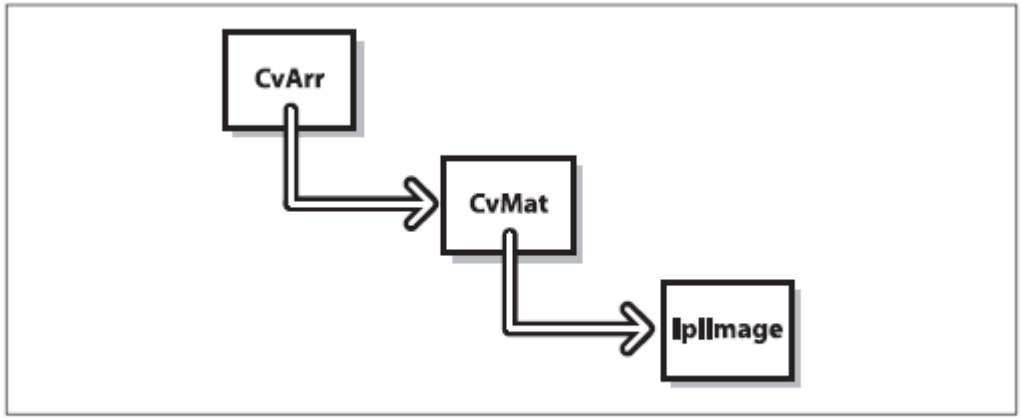


Figure 3-1. Even though OpenCV is implemented in C, the structures used in OpenCV have an object-oriented design; in effect, IplImage is derived from CvMat, which is derived from CvArr

CvMat 구조체

- 구조체 정의

Example 3-1. CvMat structure: the matrix "header"

```
typedef struct CvMat {
    int type;
    int step;
    int* refcount;    // for internal use only
    union {
        uchar*  ptr;
        short*  s;
        int*    i;
        float*  fl;
        double* db;
    } data;
    union {
        int rows;
        int height;
    };
    union {
        int cols;
        int width;
    };
} CvMat;
```

CvMat 구조체

- MxN float 타입 행렬 만들기
  - CvMat *matrix = cvCreateMat(M, N, CV_32FC1);
  - cvReleaseMat(&matrix);
- 행렬 원소 접근
  - cvmSet(matrix, i, j, 2.0);
  - float value = cvmGet(matrix, i, j);
  - matrix->data.fl[i*matrix->cols+j] = 2.0;
  - float value = matrix->data.fl[i*matrix->cols+j];

IplImage 구조체

- IplImage (Image Processing Library)
- Image structure
  - cvMat을 상속하며 영상을 표현하기에 적합한 추가 항목들을 가지고 있음

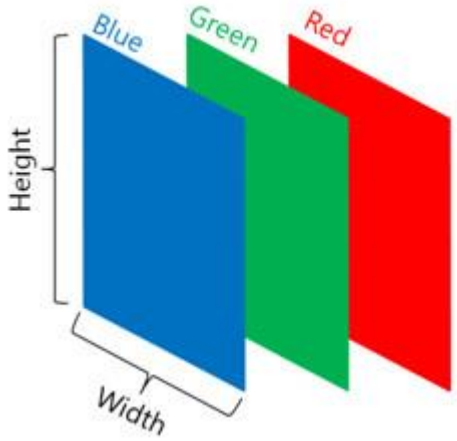
```
IplImage
  |-- int  nChannels;     // Number of color channels (1,2,3,4)
  |-- int  depth;         // Pixel depth in bits:
  |                       //   IPL_DEPTH_8U, IPL_DEPTH_8S,
  |                       //   IPL_DEPTH_16U,IPL_DEPTH_16S,
  |                       //   IPL_DEPTH_32S,IPL_DEPTH_32F,
  |                       //   IPL_DEPTH_64F
  |-- int  width;         // image width in pixels
  |-- int  height;        // image height in pixels
  |-- char* imageData;    // pointer to aligned image data
  |                       // Note that color images are stored in BGR order
  |-- int  dataOrder;     // 0 - interleaved color channels,
  |                       // 1 - separate color channels
  |                       // cvCreateImage can only create interleaved images
  |-- int  origin;        // 0 - top-left origin,
  |                       // 1 - bottom-left origin (Windows bitmaps style)
  |-- int  widthStep;     // size of aligned image row in bytes
  |-- int  imageSize;     // image data size in bytes = height*widthStep
  |-- struct _IplROI *roi;// image ROI. when not NULL specifies image
  |                       // region  to be processed.
  |-- char *imageDataOrigin; // pointer to the unaligned origin of image data
  |                          // (needed for correct image deallocation)
  |
  |-- int  align;         // Alignment of image rows: 4 or 8 byte alignment
  |                       // OpenCV ignores this and uses widthStep instead
  |-- char colorModel[4]; // Color model - ignored by OpenCV
```

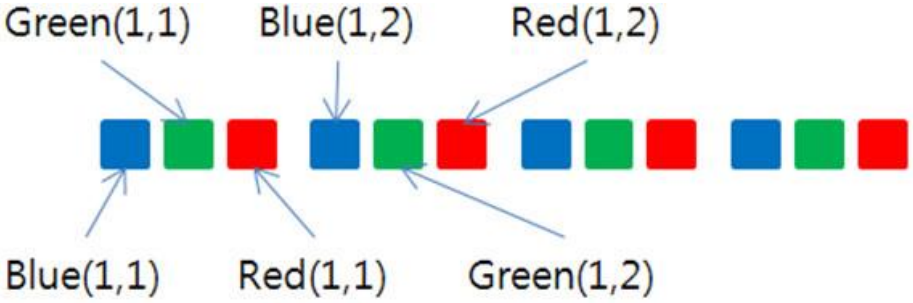
IplImage 구조체

- int nChannels
  - Number of color channel(ex: gray->1, rgb color->3)
- int depth
  - Pixel depth in bits(ex: IPL_DEPTH_8U, 8bits unsigned char, 0~255)
- int width
  - Image width in pixels
- int height
  - Image height in pixels
- char* ImageData
  - Pointer to aligned image data
- int widthStep
  - Size of aligned image row in bytes

IplImage 구조체 – Pixel data 접근

- imageData
  - OpenCV는 벡터를 표현하지 못해 일차원 행렬로 저장
  - 가로로 B, G, R 순서





IplImage 구조체 – Pixel data 접근

- imageData
  - (unsigned char)img->imageData[y*img->widthStep+x]
    cf. 3 channel : imageData[y*img->widthStep+channel*x+0,1,2]
  - pData = (unsigned char*) img->imageData
    pData[y*img->widthStep+channel*x+0,1,2]

- 8bit 1 channel
  - (unsigned char) img->imageData [image->widthStep*y+x]
- 8bit 3 channel
  - B: (unsigned char) img->imageData [image->widthStep*y+3*x+0]
  - G: (unsigned char) img->imageData [image->widthStep*y+3*x+1]
  - R: (unsigned char) img->imageData [image->widthStep*y+3*x+2]

IplImage 구조체 – Pixel data 접근

- CvScalar
  - 4개 이하의 숫자를 저장할 수 있도록 만든 구조체

```
struct CvScalar
{
        double val[4];
};
```

  - 0번은 blue, 1번은 green, 2번은 red값을 저장
  - Example)

```
CvScalar s;
s.val[0]=200;        (Blue)
s.val[1]=11;         (Green)
s.val[2]=123;        (Red)
```

IplImage 구조체 – Pixel data 접근

- CvScalar cvGet2D(IplImage* ,y ,x)
  - (x,y)에서의 intensity 값 얻기
  - Example)

```
CvScalar s;
s=cvGet2D(img, 30, 40);
```

- Void cvSet2D(IplImage* ,y ,x ,CvScalar)
  - (x,y)에서의 intensity 값 변경
  - Example)

```
Cvscalar s=cvScalar(100,0,0);
cvSet2D(img, 30, 40, s);
```

IplImage 구조체 – 예시1

- 이미지 parameter 출력

600

525

```
#include <opencv₩cv.h>
#include <opencv₩cxcore.h>
#include <opencv₩highgui.h>

in
{
    j=150;

    unsigned char b=img->imageData[j*img->widthStep+3*i+0];
    unsigned char g=img->imageData[j*img->widthStep+3*i+1];
    unsigned char r=img->imageData[j*img->widthStep+3*i+2];

    printf("width:%d ₩nheight:%d ₩nwidthStep:%d ₩nnChannels:%d ₩ndepth:%d ₩nb:%d ₩ng:%d ₩nr:%d ₩n",
        width, height, widthStep, nChannels, depth, b, g, r);

    cvReleaseImage(&img);

    return 1;
}
```

C:₩Windows₩system32₩cmd.exe

```
width:600
height:525
widthStep:1800
nChannels:3
depth:8
b:159
g:155
r:154
계속하려면 아무 키나 누르십시오 . . .
```

IplImage 구조체

- IplImage* cvLoadImage(image_path);
  - 파일로부터 이미지를 읽는다.
  - 성공할 경우 메모리가 생성되어서 IplImage pointer를 반환
  - 실패할 경우 NULL 값을 반환
- IplImage* cvCreateImage(size, depth, channel);
  - IplImage 구조체의 메모리를 생성하여 그 포인터를 반환
- cvSaveImage(image_path, IplImage *);
  - 이미지를 파일로 저장
- cvReleaseImage(IplImage **);
  - 이미지를 저장하고 있던 메모리를 해제

IplImage 구조체

- cvNamedWindow(window_name);
  - 윈도우를 생성, 윈도우에 이름을 부여하여 이 이름을 기준으로 윈도우들을 구분
- cvShowImage(window_name, IplImage*);
  - 주어진 이름에 해당하는 윈도우에 이미지를 출력
- cvDestroyWindow(window_name);
  - 주어진 이름에 해당하는 윈도우를 제거

IplImage 구조체 – 예시2

- 이미지 읽고

```
#include <opencv
#include <opencv
#include <opencv

int main()
{
    IplImage * i

    if(img==NULL
        return 1

    cvNamedWindo
    cvShowImage(
    cvWaitKey(0)
    cvDestroyWin

    cvReleaseIma

    return 0;
}
```

IplImage 구조체 – 예시3

- 3-chann

```
#include <ope
#include <ope
#include <ope

int main(){
    IplImage

    if(img==N
        retur

    IplImage
    img_1chan
    int x, y;

    unsigned
    double in

    for(y=0;y
        for(x
            b
            g
            r

            i
            i
        }
    }

    cvShowIma
    cvWaitKey
    cvReleaseImage(&img);
    cvReleaseImage(&img_1channel);

    return 0;
}
```

IplImageWrapper

- pixel data 접근을 더 쉽게 하기 위해 Wrapper class (첨부)

```
#pragma once

template<class T>
class ClplimageWrapper
{
private:
    IplImage* imgp;
public:
    ClplimageWrapper(IplImage* img=0) {imgp=img;}
    ~ClplimageWrapper(){imgp=0;}
    void    Bind(IplImage* img) {imgp=img;}
    void operator=(IplImage* img) {imgp=img;}
    inline T* operator[](const int rowIndx) {
        return ((T *)(imgp->imageData + rowIndx*imgp->widthStep));
    }
};

typedef struct{
    unsigned char b,g,r;
} RgbPixel;

typedef struct{
    float b,g,r;
} RgbPixelFloat;

typedef unsigned char   BYTE;

typedef ClplimageWrapper<RgbPixel>          RgbImage;
typedef ClplimageWrapper<RgbPixelFloat> RgbImageFloat;
typedef ClplimageWrapper<BYTE>              BwImage;
typedef ClplimageWrapper<float>         BwImageFloat;
typedef ClplimageWrapper<int>           BwImageInt;
```

pixel 접근을 위한 pointer 설정

uchar type의 r, g, b 픽셀 값

IplImageWrapper

- Color image
  - RgbImage wimg(img);
    - RgbImage wimg는 IplImage *img의 data를 공유
  - wimg[y][x].r : red channel pixel intensity
  - wimg[y][x].g : green channel pixel intensity
  - wimg[y][x].b : blue channel pixel intensity
- Gray image
  - BwImage wgimg(gimg);
    - BwImage wgimg는 grah 이미지인 IplImage *gimg의 data를 공유
  - wgimg[y][x] : gray pixel intensity

IplImageWrapper – 예시

```
#include <opencv2/opencv.hpp>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>

#include "IplImageWrapper.h"

int main()
{
        IplImage*img = cvLoadImage( "lena.jpg", CV_LOAD_IMAGE_COLOR );
        if( img == NULL )
        {
                fprintf(stderr, "cannot load image\n");
                exit(0);
        }

        const int width = img->width;
        const int height = img->height;
        const int min_width_and_height = std::min( width, height );

        RgbImage  wimg(img);
        for(int i = 0; i < min_width_and_height; i++ )
        {
                wimg[i][i].r = 255;
                wimg[i][i].g = 0;
                wimg[i][i].b = 0;
        }
```

IplImageWrapper 헤더 파일

픽셀 접근

IplImageWrapper – 예시

```
IplImage*gimg = cvCreateImage( cvGetSize(img), IPL_DEPTH_8U, 1 );
cvCvtColor( img, gimg, CV_RGB2GRAY );

BwImage  wgimg(gimg);
for(int i = 0; i < min_width_and_height; i++ )
{
                wgimg[min_width_and_height-i-1][i] = 255;
}

cvNamedWindow("img");
cvShowImage( "img", img);

cvNamedWindow("gimg");
cvShowImage( "gimg", gimg);

cvWaitKey(0);

cvReleaseImage(&gimg);
cvReleaseImage(&img);

return 0;
}
```

영상처리

- cvSmooth

  (const CvArr* src, CvArr* dst, int smoothtype=CV_GAUSSIAN, int param1=3, int param2=0, double param3=0, double param4=0)

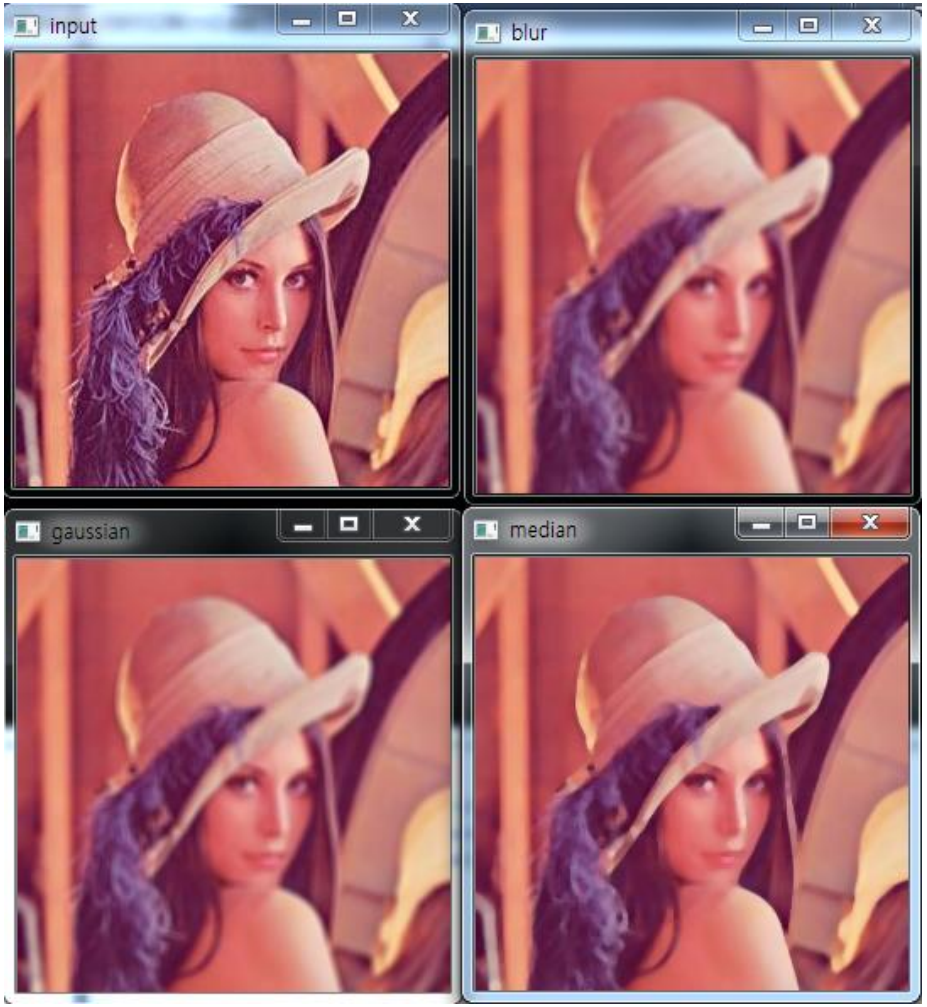
- cvMorphologyEx

  (const CvArr* src, CvArr* dst, CvArr* temp, IplConvKernel* element, int operation, int iterations=1)

  - Dilate: 최대값 연산, 밝은 영역 확장
  - Erode: 최소값 연산, 어두운 영역 확장
  - Open: 침식 후 팽창(밝은 잡음제거)
  - Close: 팽창 후 침식(어두운 잡음제거)
  - Gradient: dilate – eroad
  - TopHat = src –open(밝은 부분 강조)
  - BlackHat = close – src(어두운 부분 강조)

영상처리 – 예시1) cvSmooth

```
#include <opencv\cv.h>
#include <opencv\cxcore.h>
#include <opencv\highgui.h>

int main(){
IplImage *img = cvLoadImage("lena.jpg", CV_LOAD_IMAGE_COLOR);

if(img==NULL)
return 1;

IplImage *img_blur, *img_gaussian, *img_median;

img_blur=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 3);
img_gaussian=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 3);
img_median=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 3);

cvSmooth(img, img_blur, CV_BLUR, 5, 5, 0, 0);
cvSmooth(img, img_gaussian, CV_GAUSSIAN, 5, 5, 15, 10);
cvSmooth(img, img_median, CV_MEDIAN, 5, 5, 0, 0);

cvShowImage("input", img);
cvShowImage("blur", img_blur);
cvShowImage("gaussian", img_gaussian);
cvShowImage("median", img_median);

cvWaitKey(0);

cvReleaseImage(&img);
cvReleaseImage(&img_blur);
cvReleaseImage(&img_gaussian);
cvReleaseImage(&img_median);

return 0;
}
```



영상처리 – 예시2) cvMorpologyEx

```
#include <opencv\cv.h>
#include <opencv\cxcore.h>
#include <opencv\highgui.h>

int main(){
IplImage *img = cvLoadImage("test_img.jpg", CV_LOAD_IMAGE_GRAYSCALE);

if(img==NULL)
return 1;

IplConvKernel *element;
element = cvCreateStructuringElementEx (5, 5, 3, 3, CV_SHAPE_RECT, NULL); // 필터의 크기를 5x5로 설정, 중심은 3x3

IplImage *img_erode, *img_dilate, *img_open, *img_close, *img_gradient, *img_tophat, *img_blackhat;

img_erode=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);
img_dilate=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);
img_open=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);
img_close=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);
img_gradient=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);
img_tophat=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);
img_blackhat=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);

cvErode(img, img_erode, element, 1);
cvDilate(img, img_dilate, element, 1);
cvMorphologyEx(img, img_open, NULL, element, CV_MOP_OPEN, 1);
cvMorphologyEx(img, img_close, NULL, element, CV_MOP_CLOSE, 1);
cvMorphologyEx(img, img_gradient, NULL, element, CV_MOP_GRADIENT, 1);
cvMorphologyEx(img, img_tophat, NULL, element, CV_MOP_TOPHAT, 1);
cvMorphologyEx(img, img_blackhat, NULL, element, CV_MOP_BLACKHAT, 1);
```

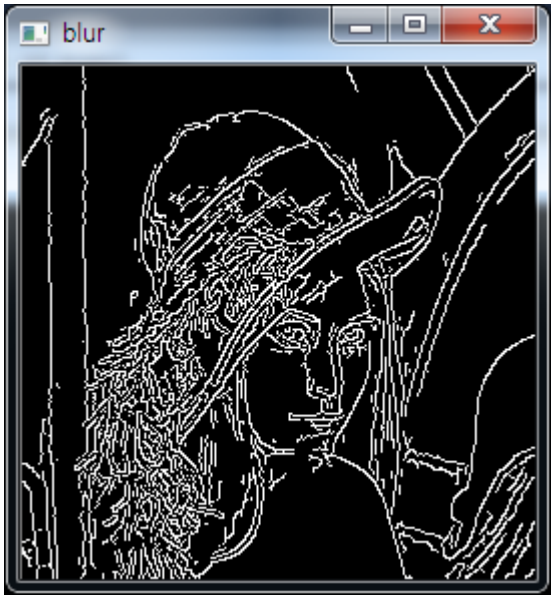
Kernel 설정

영상처리 – 예시2) cvMorpologyEx

```
cvShowImage("input", img);
cvShowImage("erode", img_erode);
cvShowImage("dilate", img_dilate);
cvShowImage("open", img_open);
cvShowImage("close", img_close);
cvShowImage("gradient", img_gradient);
cvShowImage("tophat", img_tophat);
cvShowImage("blackhat", img_blackhat);

cvWaitKey(0);

cvReleaseImage(&img);
cvReleaseImage(&img_erode);
cvReleaseImage(&img_dilate);
cvReleaseImage(&img_open);
cvReleaseImage(&img_close);
cvReleaseImage(&img_gradient);
cvReleaseImage(&img_tophat);
cvReleaseImage(&img_blackhat);

return 0;
}
```

영상처리 – 예시3) Canny edge detector

```
#include <opencv\cv.h>
#include <opencv\cxcore.h>
#include <opencv\highgui.h>

int main()
{
	IplImage *img = cvLoadImage("lena_gray.jpg", CV_LOAD_IMAGE_GRAYSCALE);

	if(img==NULL)
		return 1;

	IplImage *img_dst;

	img_dst=cvCreateImage(cvGetSize(img), IPL_DEPTH_8U, 1);

	cvCanny(img, img_dst, 50, 100);

	cvShowImage("blur", img_dst);

	cvWaitKey(0);

	cvReleaseImage(&img);
	cvReleaseImage(&img_dst);

	return 0;
}
```

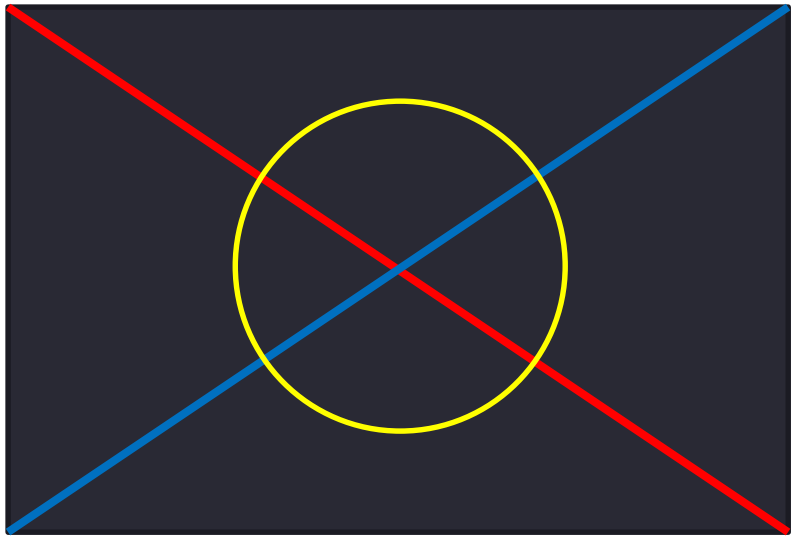


영상처리 – 예시4) ROI 설정

- ROI(Region of Interest)
  - 원하는 특정 영역만 작업을 수행하고자 할 경우

```
#include <opencv\cv.h>
#include <opencv\cxcore.h>
#include <opencv\highgui.h>

int main()
{
        IplImage *img = cvLoadImage("lena_gray.jpg", CV_LOAD_IMAGE_GRAYSCALE);

        if(img==NULL)
        return 1;

        int x, y, width, height;
        x=100;
        y=100;
        width=100;
        height=100;

        cvSetImageROI(img, cvRect(x, y, width, height));
        cvAddS(img, cvScalar(100), img);
        cvResetImageROI(img);

        cvShowImage("window", img);

        cvWaitKey(0);

        cvReleaseImage(&img);

        return 0;
}
```

Exercise - 1



- Width = 640
- Height = 480

Exercise - 2

크기가 다름. cvResize 함수 사용

Exercise - 3

R, G, B 분리

컬러 영상을 3개의 gray 영상으로 분리

컬러 영상의 Red 값으로 이루어진 gray 영상, Green 값으로 이루어진 gray 영상, Blue 값으로 이루어진 gray 영상이 출력

cvSplit 함수를 사용해도 되고, 모든 픽셀에 대하여 직접적인 접근을 통하여 분리해도 됨

color red green blue

Mat class

- OpenCV C++ matrix class
- “cv.h”의 cv namespace를 사용
  - using namespace cv; 정의 후 function 사용
  - or 매 function마다 cv::function 사용
- 생성과 소멸이 자동으로 이루어 진다는 특징
  - 기존의 Create나 Release 없이 자동으로 이미지를 생성

Mat class

- Mat 생성
  - Mat A(cv::Size(x,y), CV_8UC3, cv::Scalar(0, 0, 255));
  - 혹은 Mat A(y, x, CV_8UC3, cv::Scalar(0, 0, 255));
  - x×y Mat 생성, 8 bit 3 channel 이고, (0,0,255)로 초기화
- 이미지 읽어오기
  - Mat A=cv::imread(image_path, CV_LOAD_IMAGE_COLOR);
  - color 이미지 image_path를 읽어와서 Mat A에 대입
- 이미지 보여주기
  - cv::imshow(“window name”, Mat A);
  - “window name”창에 Mat A를 출력

Mat class

- 이미지 복사하기
  - Mat B(A); or B=A;
    - B는 A와 data를 공유하기 때문에 A가 바뀌면 B도 바뀜
  - B=A.clone()
    - 단순 data값만 복사하는 것으로 A가 바뀌어도 B가 바뀌지 않음
- 픽셀 접근
  - gray-scale(8UC1 type) (x,y)
    - cv::Scalar intensity = img.at<uchar>(y,x);
    - cv::Scalar intensity = img.at<uchar>(cv::Point(x, y));
  - 3 channel image width BGR color ordering
    - cv::Vec3b intensity = img.at<Vec3b>(y, x);
    - uchar blue = intensity.val[0];
    - uchar green = intensity.val[1];
    - uchar red = intensity.val[2];

Mat class - 예시

```
#include <opencv\cv.h>
#include <opencv\cxcore.h>
#include <opencv\highgui.h>

int main()
{
cv::Mat img, gimg;
img = cv::imread(  "lena.jpg", cv::IMREAD_COLOR );

const int width = img.size().width;
const int height = img.size().height;

const int min_width_and_height = std::min( width, height );

for(int i = 0; i < min_width_and_height; i++ ){
img.at<cv::Vec3b>(i,i)[0] = 255;// blue
img.at<cv::Vec3b>(i,i)[1] = 0;// green
img.at<cv::Vec3b>(i,i)[2] = 0;// red
}

cv::cvtColor( img, gimg, CV_RGB2GRAY );

for(int i = 0; i < min_width_and_height; i++ )
                        gimg.at<uchar>(min_width_and_height-i-1,i) = 255;

cv::namedWindow( "img" );
cv::imshow( "img", img );

cv::namedWindow( "gimg" );
cv::imshow( "gimg", gimg );

cv::waitKey(0);

return 0;
}
```