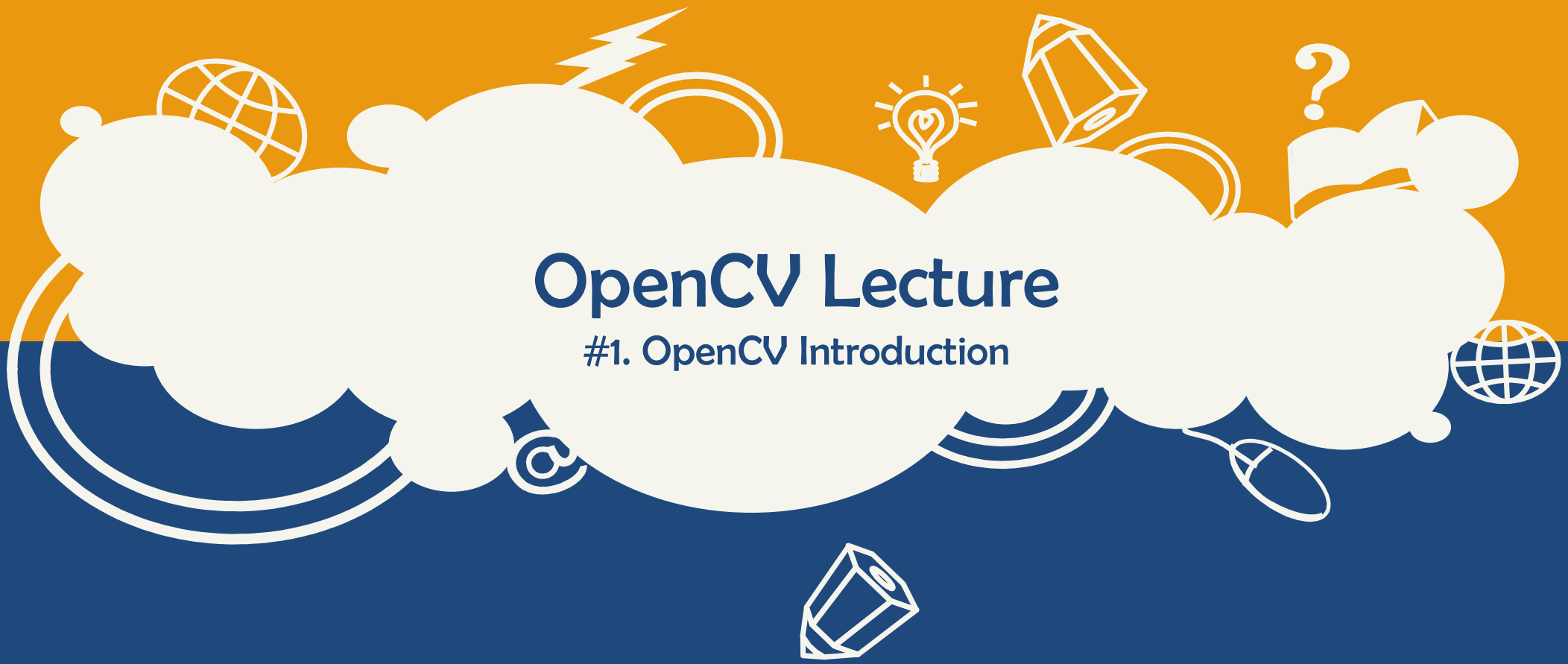


OpenCV Lecture

#1. OpenCV Introduction

MareArts
http://study.marearts.com

What is the OpenCV?

- OpenCV : Open Source Computer Vision Library
  - Site : http://www.opencv.org
  - 2006.10: 1.0(first) ... 2015.12: 3.1(present)
  - BSD license : Free academic and commercial
  - C++, Python, Java / Window, Linux, Mac OS, iOS, Android
  - Github : https://github.com/Itseez/opencv
    - Extra Contribute : https://github.com/Itseez/opencv_contrib

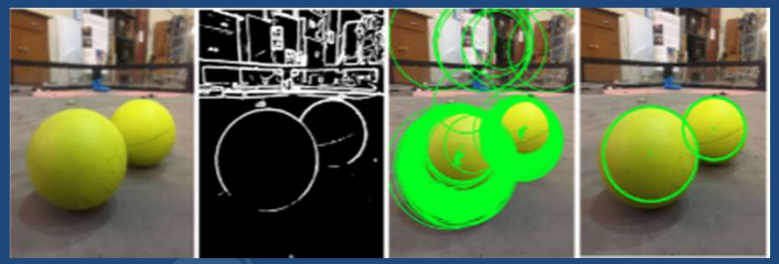
Good site to reference opencv : http://study.marearts.com

What is the OpenCV?

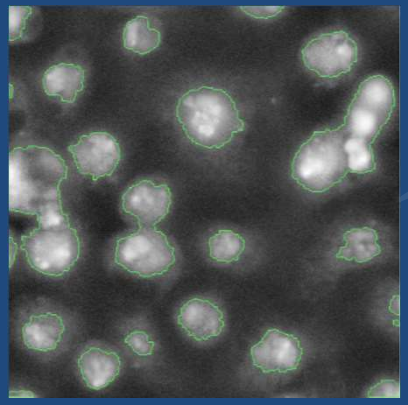
- Computer Vision Library
  - Image Processing :
    - Image Enhancement, Filter, Rotation, Hough Transform, Histogram...
  - Robot / Machine / Video / Vision :
    - Tracking, Feature description,
  - Artificial Intelligence
    - Pattern Recognition / Machine Learning
      - Neural Network, Deep learning, AdaBoost, SVM...
  - 3D geometry
    - Camera Calibration, 3D reconstruction, Stereo Camera
  - Etc
    - Parallel Processing : CUDA, OpenCL..
    - Optimization : nonlinear optimization, RANSAC...

What is the OpenCV?

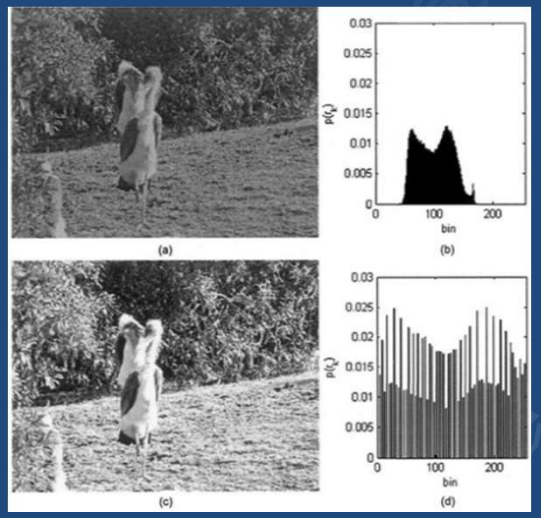
- Image Processing



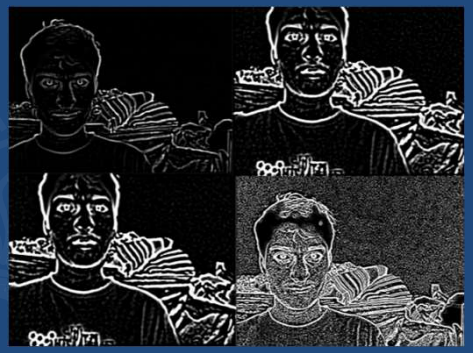
Circle Detection using Hough TF



Cell Segmentation



Histogram equalization

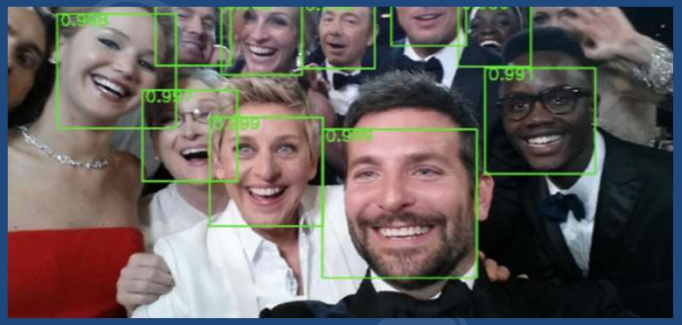


Edge Detection

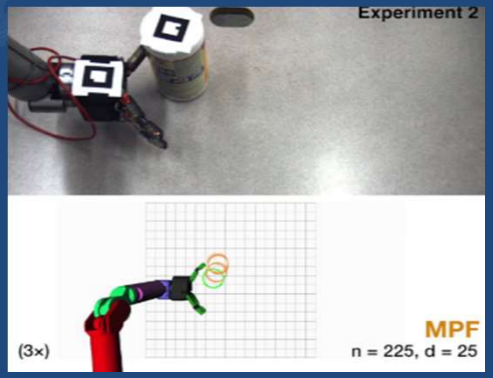
Deburring

What is the OpenCV?

- Robot / Machine / Video / Vision



Face Detection

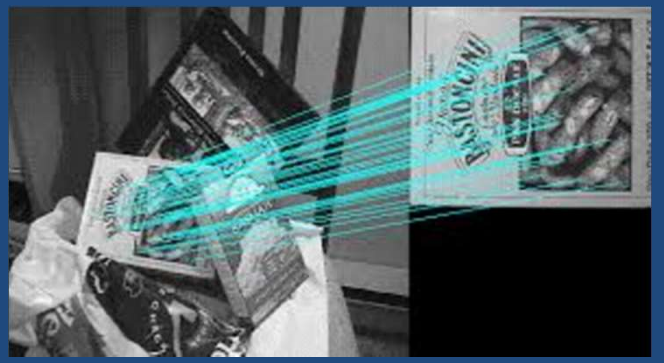


Cup pose estimation

Obstacle avoidance

Inspection

Tracking (Multi Pedestrian)



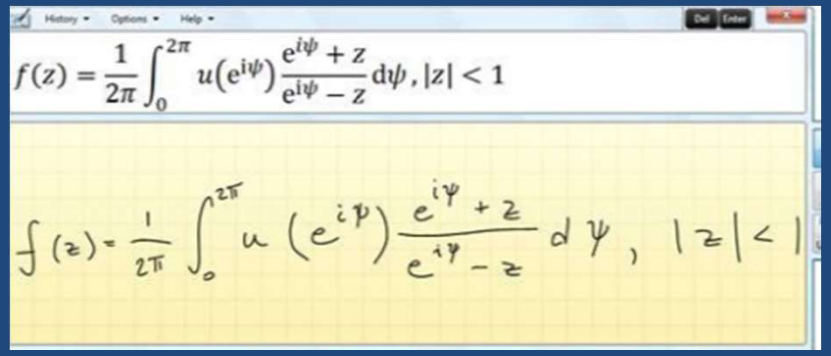
Feature Detection and Matching

What is the OpenCV?

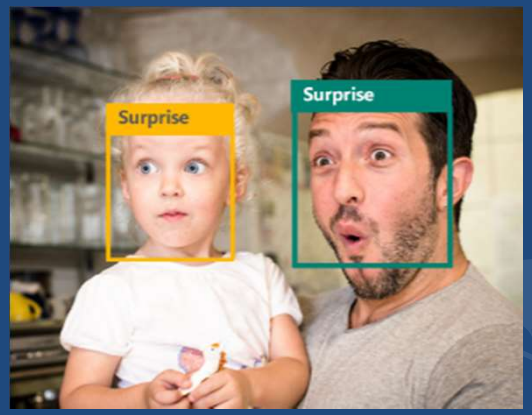
- Artificial Intelligence

Deep learning

Image understanding



Handwriting Recognition

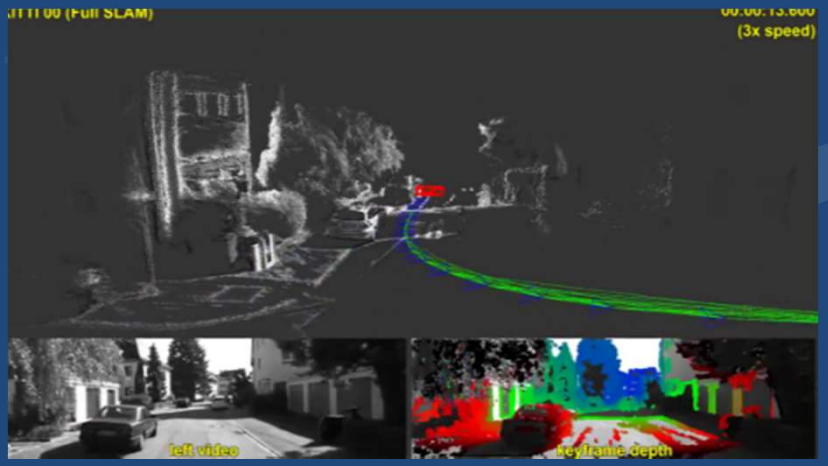


Emotion Detection

What is the OpenCV?

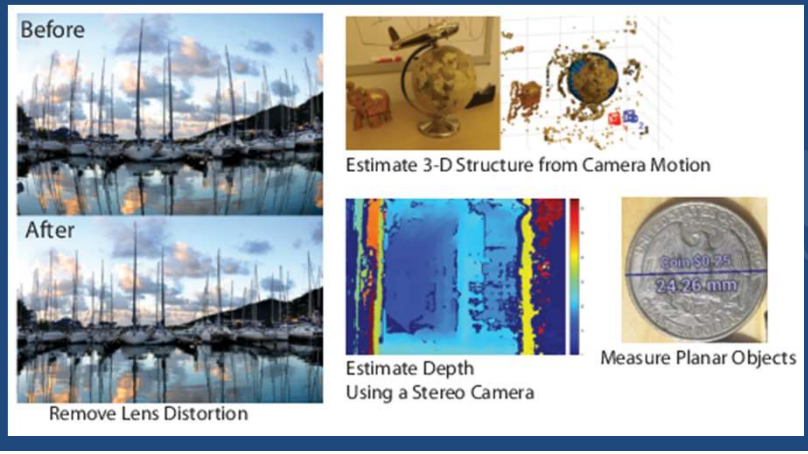
- 3D Geometry

Stereo Camera



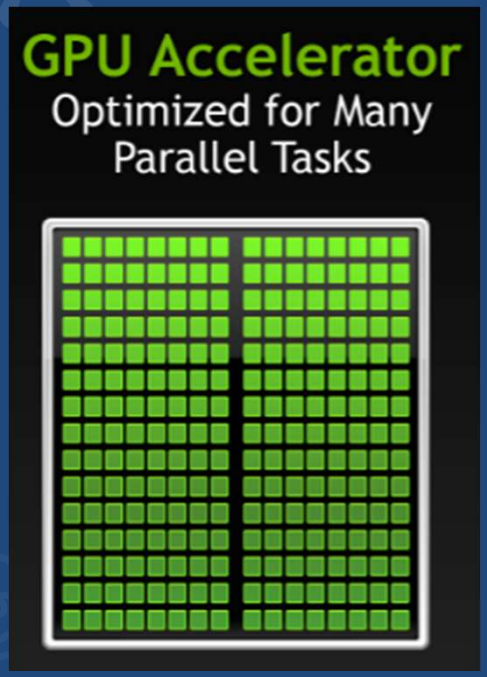
SLAM & 3D reconstruction

3D reconstruction

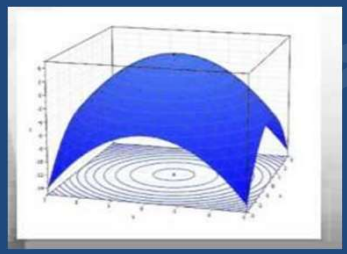


What is the OpenCV?

- Etc



Parallel Programming
- Nvidia CUDA
- OpenCL
- TBB(Thread Building Block)



Non-linear Optimization
- Bundle Adjustment
- RANSAC

Super Resolution

3D Visualizer

Explore the OpenCV functions

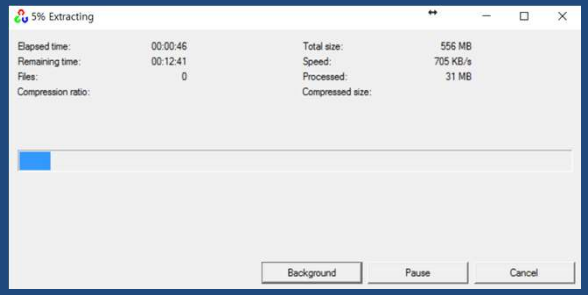
- OpenCV 3.1.0 Reference (http://docs.opencv.org/3.1.0/#gsc.tab=0)

- Main modules:
  - core. **Core functionality**
  - imgproc. **Image processing**
  - imgcodecs. **Image file reading and writing**
  - videoio. **Media I/O**
  - highgui. **High-level GUI**
  - video. **Video Analysis**
  - calib3d. **Camera Calibration and 3D Reconstruction**
  - features2d. **2D Features Framework**
  - objdetect. **Object Detection**
  - ml. **Machine Learning**
  - flann. **Clustering and Search in Multi-Dimensional Spaces**
  - photo. **Computational Photography**
  - stitching. **Images stitching**
  - cudaarithm. **Operations on Matrices**
  - cudabgsegm. **Background Segmentation**
  - cudacodec. **Video Encoding/Decoding**
  - cudafeatures2d. **Feature Detection and Description**
  - cudafilters. **Image Filtering**
  - cudaimgproc. **Image Processing**
  - cudalegacy. **Legacy support**
  - cudaobjdetect. **Object Detection**
  - cudaoptflow. **Optical Flow**
  - cudastereo. **Stereo Correspondence**
  - cudawarping. **Image Warping**
  - cudev. **Device layer**
  - shape. **Shape Distance and Matching**
  - superres. **Super Resolution**
  - videostab. **Video Stabilization**
  - viz. **3D Visualizer**

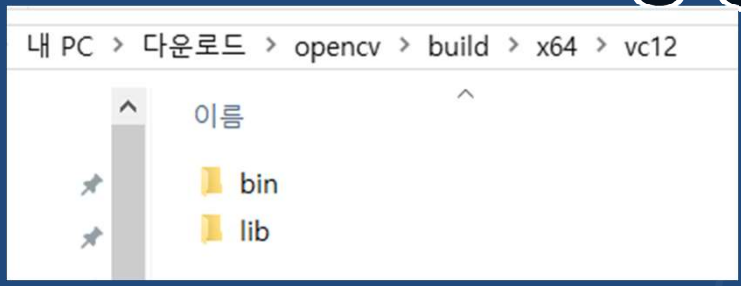
- Extra modules:
  - aruco. **ArUco Marker Detection**
  - bgsegm. **Improved Background-Foreground Segmentation Methods**
  - bioinspired. **Biologically inspired vision models and derivated tools**
  - ccalib. **Custom Calibration Pattern for 3D reconstruction**
  - cvv. **GUI for Interactive Visual Debugging of Computer Vision Programs**
  - datasets. **Framework for working with different datasets**
  - dnn. **Deep Neural Network module**
  - dpm. **Deformable Part-based Models**
  - face. **Face Recognition**
  - fuzzy. **Image processing based on fuzzy mathematics**
  - hdf. **Hierarchical Data Format I/O routines**
  - line_descriptor. **Binary descriptors for lines extracted from an image**
  - matlab. **MATLAB Bridge**
  - optflow. **Optical Flow Algorithms**
  - plot. **Plot function for Mat data**
  - reg. **Image Registration**
  - rgbd. **RGB-Depth Processing**
  - saliency. **Saliency API**
  - sfm. **Structure From Motion**
  - stereo. **Stereo Correspondance Algorithms**
  - structured_light. **Structured Light API**
  - surface_matching. **Surface Matching**
  - text. **Scene Text Detection and Recognition**
  - tracking. **Tracking API**
  - xfeatures2d. **Extra 2D Features Framework**
  - ximgproc. **Extended Image Processing**
  - xobjdetect. **Extended object detection**
  - xphoto. **Additional photo processing algorithms**

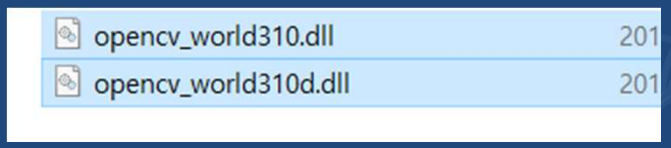
How to use OpenCV

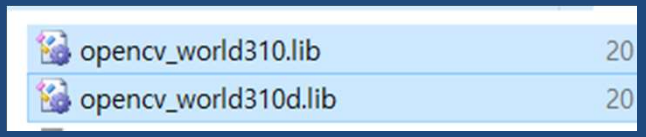
- 1. download from official site
  - http://opencv.org/downloads.html
  - Select version and your OS



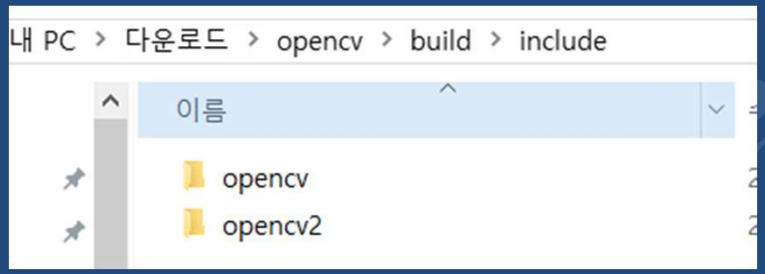
Download and Extraction







Lib, Dll, pre-compiled

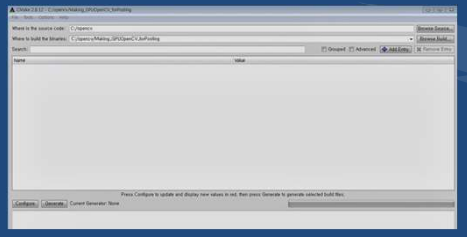
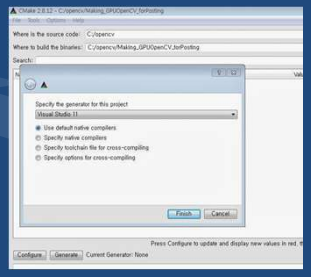
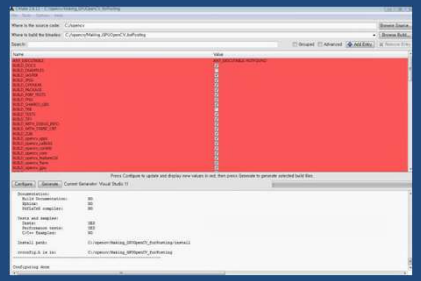
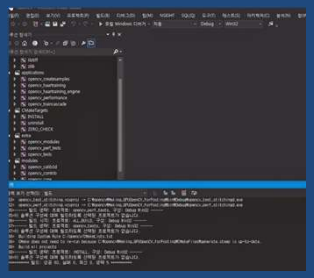


Header files

How to use OpenCV

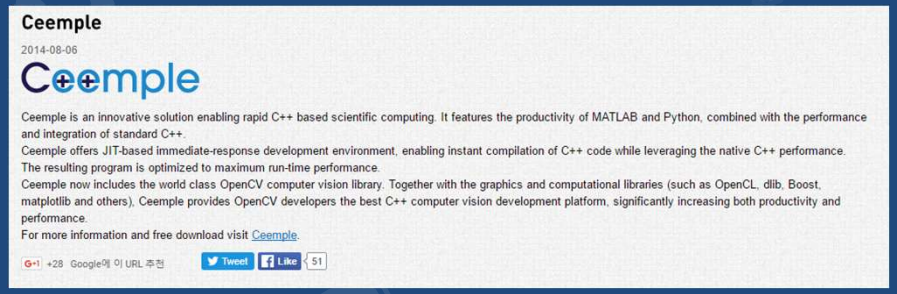
- 2. build lib/dll (refer to : http://study.marearts.com/search/label/Opencv%20Build)
  - Ready to make the source code
  - Generate code for your environment and your option
  - Code compile, create dll, lib files
  - Including options. ex) cuda, TBB ..
  - And that can include extra modules

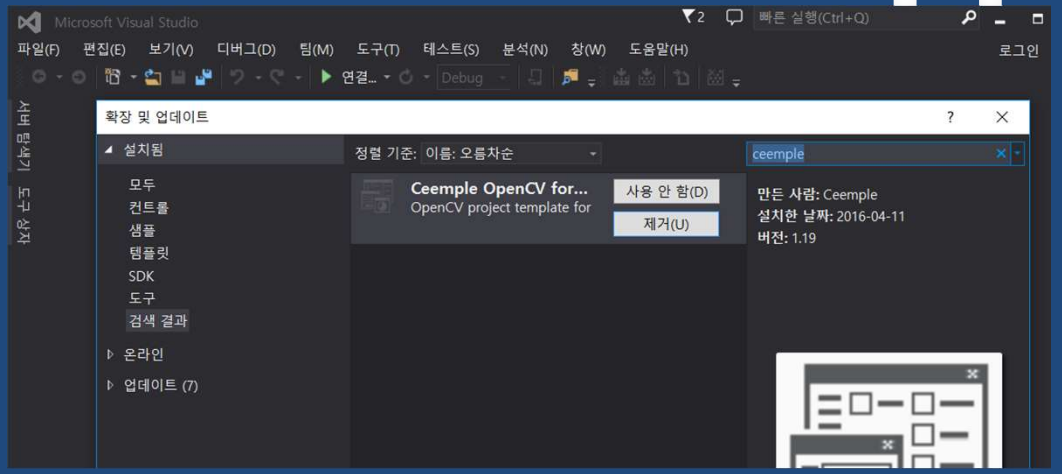
    (refer to : http://study.marearts.com/2015/01/mil-boosting-tracker-test-in-opencv-30.html

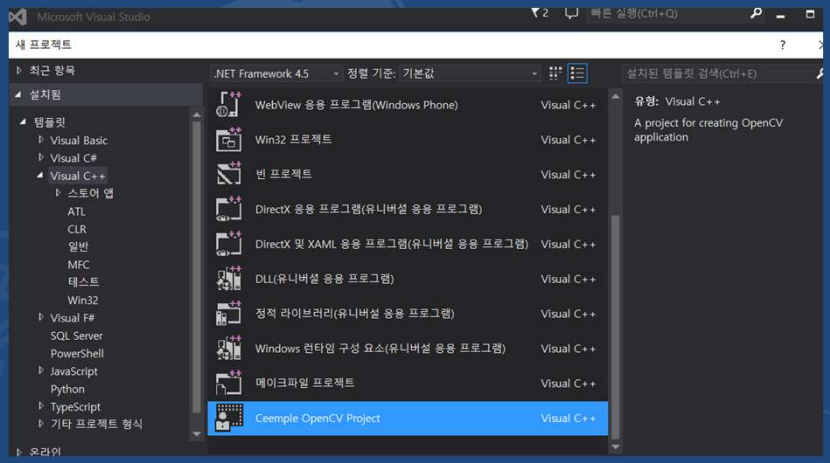





Try OpenCV firstly

3. use Ceemple OpenCV







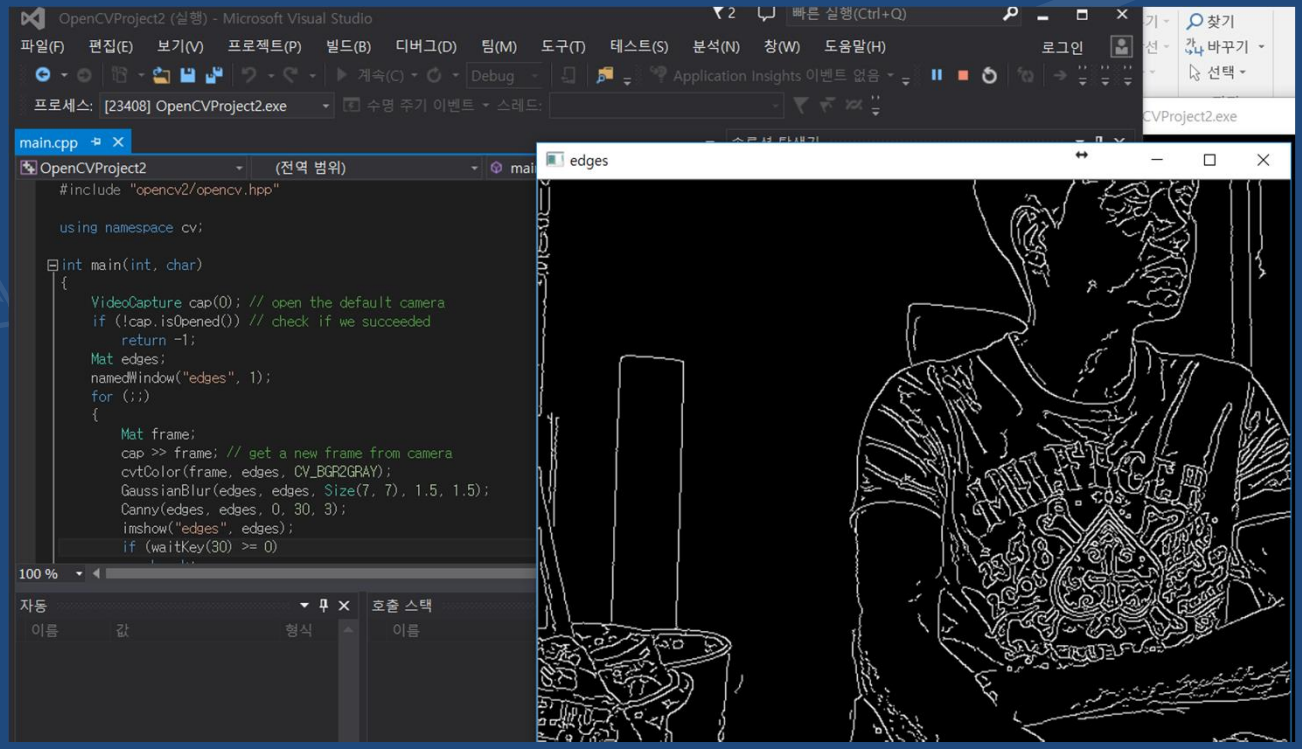
Visual Studio

- Tools - Extension and Update
- Search ceemple -> install
- After, you can make openCV project very easy.

(Refer to http://study.marearts.com/2016/01/ceemple-opencv-pre-build-and-quick.html)

Try OpenCV firstly

- Use OpenCV firstly with me



How to build own OpenCV

- Assignment #1
  - Build your own OpenCV
  - Option #1 : Include CUDA, TBB
  - Option #2 : include extra opencv module
    - https://github.com/ltseez/opencv_contrib : github repository for extra module
- Assignment #2
  - Use ceemple OpenCV, build and run!

Reference..
Official Site : http://docs.opencv.org/3.1.0/df/d65/tutorial_table_of_content_introduction.html#gsc.tab=0
Build : http://study.marearts.com/search/label/Opencv%20Build
extra module : http://study.marearts.com/2015/01/mil-boosting-tracker-test-in-opencv-30.html

About OpenCV Mat (study deeply..)

- What about Mat?
  - Matrix class
  - Image, values, ... Think of all the data in a Matrix!
  - Let's use a Mat simply

About Mat (study deeply..)

- First Mat use
  - Creation
  - Set value
  - cout

```
int main(int, char)
{

    //Declaration and at the same time created
    Mat mtx(3, 3, CV_32F); // make a 3x3 floating-point matrix
    Mat cmtx(10, 1, CV_64FC2); // make a 10x1 2-channel floating-point
    // matrix (10-element complex vector)
    Mat img(Size(5, 3), CV_8UC3); // make a 3-channel (color) image
    // of 1920 columns and 1080 rows.

    //Created after the declaration
    Mat mtx2;
    mtx2 = Mat(3, 3, CV_32F);
    Mat cmtx2;
    cmtx2 = Mat(10, 1, CV_64FC1);

    //Create a point
    Mat* mtx3 = new Mat(3, 3, CV_32F);
    delete mtx3;

    //value set and print
    mtx.setTo(10);
    cout << mtx << endl;

    cmtx2.setTo(11);
    cout << cmtx2 << endl;


    return 0;
}
```

```
[10, 10, 10;
 10, 10, 10;
 10, 10, 10]
[11;
 11;
 11;
 11;
 11;
 11;
 11;
 11;
 11;
 11;
 11]
```

About Mat (study deeply..)

- First Mat use
  - Simple operation
  - +, -, /, *
  - Inv
  - Transpose

```
Mat m = Mat::ones(3, 3, CV_64F);
m = m * 3;
cout << m << endl;

double dm[3][3] = { { 1, 2, 1 }, { 0, 1, 1 }, { 1, 0, 0 } };
Mat m2 = Mat(3, 3, CV_64F, dm);
cout << m2 << endl;
cout << m+m2 << endl;
cout << m-m2 << endl;
cout << m*m2 << endl;
cout << m/m2 << endl;
cout << m2.inv() << endl;
cout << m2.t() << endl;
```

```
[3, 3, 3;
 3, 3, 3;
 3, 3, 3]
[1, 2, 1;
 0, 1, 1;
 1, 0, 0]
[4, 5, 4;
 3, 4, 4;
 4, 3, 3]
[2, 1, 2;
 3, 2, 2;
 2, 3, 3]
[6, 9, 6;
 6, 9, 6;
 6, 9, 6]
[3, 1.5, 3;
 0, 3, 3;
 3, 0, 0]
[0, 0, 1;
 1, -1, -1;
 -1, 2, 1]
[1, 0, 1;
 2, 1, 0;
 1, 1, 0]
```

About Mat (study deeply..)

- First Mat use
  - Image load
  - Show image
  - Simple processing

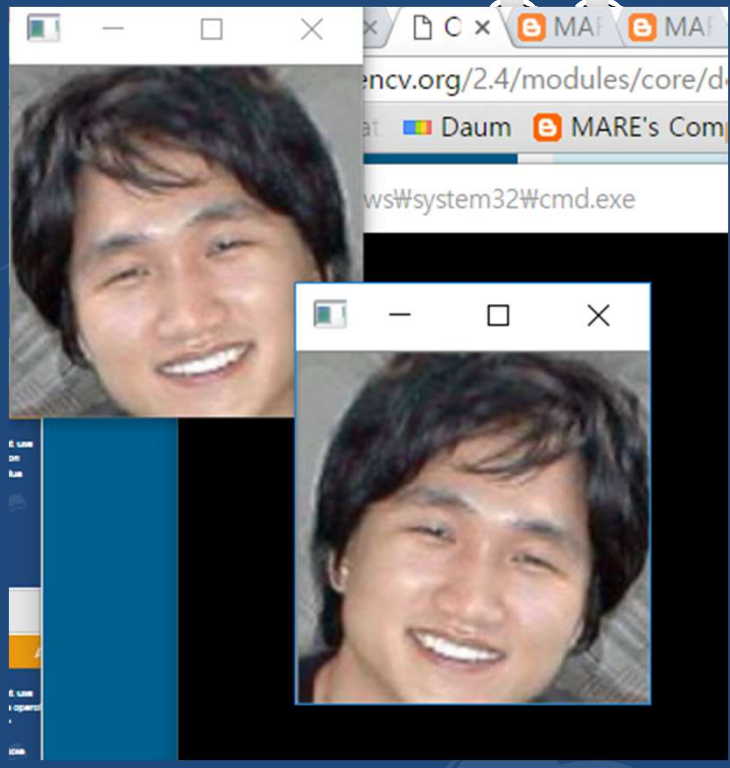
```
Mat img = imread("me.jpg");
imshow("AAA",img);
flip(img, img, 1);
imshow("BBB", img);
waitKey(0);
```

201 221 100 223 40 21

201 221 100 223 40 21

201 221 100 223 40 21
30 11 231 91 32 189
200 40 23 43 88 92
231 129 231 122 18
.....

RGB



About Mat (study deeply..)

- First Mat use
  - Set value
    - http://study.marearts.com/2014/04/opencv-study-mat-point-access-method.html
  - Vector to mat, Mat to vector
    - http://study.marearts.com/2014/01/opencv-vector-to-mat-mat-to-vector.html

About Mat (study deeply..)

- Assignment #3
  - To use over 20 functions related to the Mat
  - Create example code

Thank you.

- See you later
  - Do not forget your assignment!!
  - I will miss you very much!!

Busan BEXCO