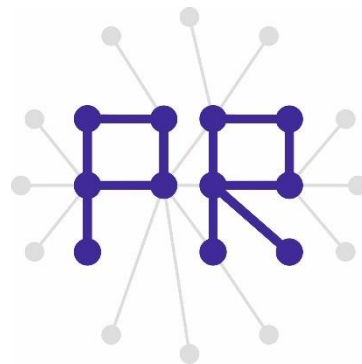


Multimedia Retrieval Exercise Course

1 Introduction and OpenCV installation

Kimiaki Shirahama, D.E.

Research Group for Pattern Recognition
Institute for Vision and Graphics
University of Siegen, Germany



About This Course

Place/Time

- H-F 107/108
- Every Thursday 12:15-13:45

Lecturer

- Dr. Eng. Kimiaki Shirahama
- kimiaki.shirahama@uni-siegen.de

Recommendation

- Students in the Bachelor course

Main Purpose

- Study/Exercise basics of multimedia data processing and retrieval

Schedule of the Course

1. **19.10** Introduction and OpenCV installation
2. **26.10** Running a simple program of OpenCV
3. **02.11** Basic image processing by OpenCV
4. **09.11** Query by example: Color histogram extraction (1/2)
5. **16.11** Query by example: Color histogram extraction (2/2)
6. **23.11** Query by example: Similarity computation (1/2)
7. **07.12** Query by example: Similarity computation (2/2)
8. **14.12** Query by example: Finishing the retrieval system
9. **21.12** Query by example: Evaluating the retrieval system
10. **04.01** Image classification using SVM (1/2)
11. **11.01** Image classification using SVM (2/2)
12. **18.01** (Buffer: How to improve the image classification performance)
13. **25.01** (Buffer: Local Features (SIFT: Scale-Invariant Feature Transform))
14. **01.02** (Buffer: Extracting and Matching Local Features)
15. **08.02** Disucssion

One day before each lesson, the slides will be uploaded the following Web site:
(the Web site will be prepared until the next course)

Requirements for This Course

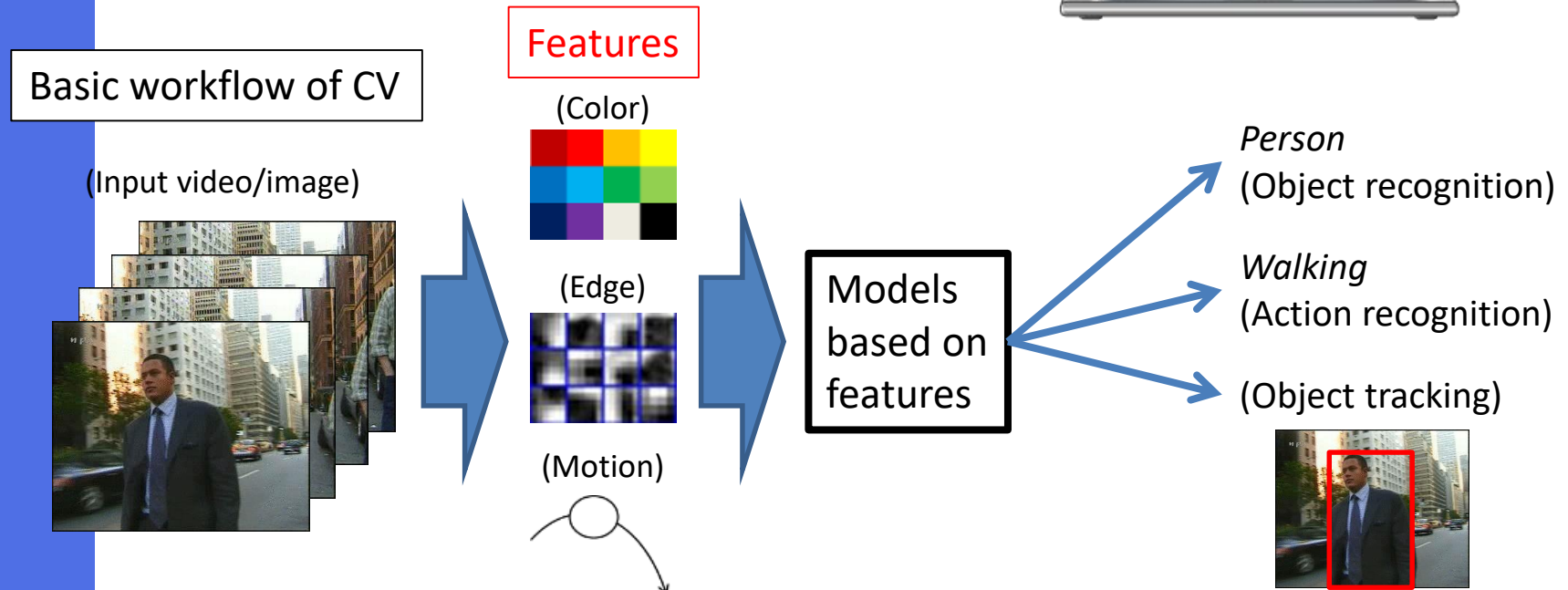
1. Take your own laptops
2. Programming skill for C is necessary for implementing the codes
(C++ is desirable, but is not necessary)
3. In the oral examination, you will be asked several questions about this course

What is OpenCV?

Open-source library consisting of programming functions for Computer Vision (CV)

Implement functionalities of human eyes on computers

- Object recognition
- Object tracking
- Scene understanding
- 3D model reconstruction etc.



Many functions in this workflow have been implemented in OpenCV!

OpenCV Installation (Windows)

- Requirement

- Microsoft Visual Studio Express 2015
<https://www.visualstudio.com/downloads/>
(Just download the exe file and double-click it)

- Download and install OpenCV

- <https://github.com/opencv/opencv/releases/tag/3.1.0>
(Just download the exe file and double-click it where you need to specify the installation directory)

- Add the following to “Path” in the environment system variables

- *C:\opencv\build\x64\vc14\bin* (If you install OpenCV directly under “C:\”)

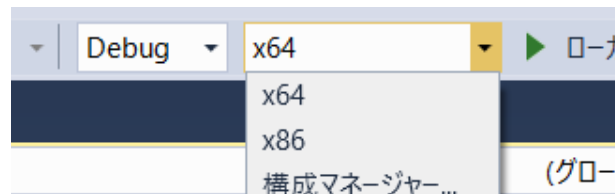
By referring to the above information and Web pages, please try to install OpenCV by yourself.

Tips to Run OpenCV Codes (Windows)

- Create a new project

Choose “Win32 console application”, and then just press “OK” or “yes”

- Change the mode from “x86” to “x64”



- Setting the include directory of OpenCV

Add C:\opencv\build\include to

Project property -> Configuration properties -> C/C++ -> general -> Additional include directories

- Setting libraries of OpenCV

Add C:\opencv\build\x64\vc14\lib to

Project property -> Configuration properties -> Linker -> general -> Additional library directories

Add opencv_world310d.lib (debug mode) or opencv_world310.lib (release mode)

Project property -> Configuration properties -> Linker -> input -> Additional dependency files

If you do the “Path” setting in the previous slide after creating a project, I think it is better to re-start Visual C++ in order to reflect the setting change. *Also, re-starting Visual C++ may helpful for other cases (I saw many students whose problems have been solved by re-starting).*

First program with OpenCV

- Read an image and show it in a dialog

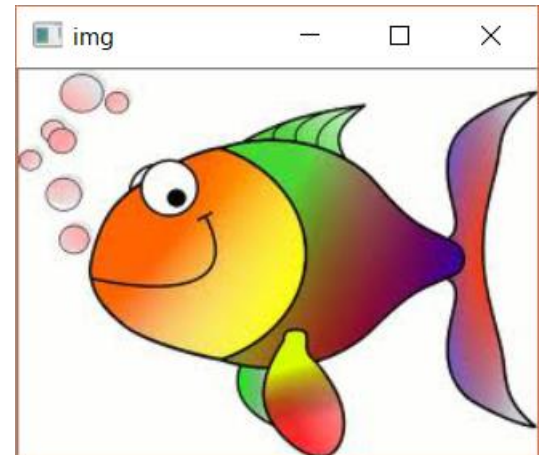
```
#include "stdafx.h" // This is a head file specific to my project
```

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

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

```
int main(int argc, const char* argv[]){  
    cv::Mat img = cv::imread("C:\\opencv\\sources\\samples\\data\\HappyFish.jpg", 1);  
    cv::namedWindow("img", cv::WINDOW_AUTOSIZE);  
    cv::imshow("img", img);  
    cv::waitKey(0);  
    cv::destroyAllWindows();  
    return 0;  
}
```

Please change the image filename depending on your environment.



For linux and mac users, please refer to web pages about OpenCV installation.
Roughly speaking:

1. Install required packages before compiling OpenCV
2. Use cmake to compile OpenCV source codes
3. Install and modify path settings

If you have questions or problems (even for linux and mac), please ask me.
I can help you install OpenCV.