

Applied Computer Vision

David Vernon
Carnegie Mellon University Africa

vernon@cmu.edu
www.vernon.eu

Lecture 2

OpenCV

Software Development Tools for Exercises and Assignments

Software Development Tools for Exercises and Assignments

- Installation of software development environment

`www.vernon.eu/wiki/ACV_Software_Development_Environment`

- Windows 7 OS
 - Microsoft Visual C++ Express compiler, version 10.0
(also known as Visual C++ 2010 or MSVC++ 2010)
 - Cmake
 - OpenCV (Version 2.10)
 - Other libraries as the course progresses
- Let's walk through the process to install these tools ...

Software Development Tools for Exercises and Assignments

Warning: the instructions here are very detailed

`www.vernon.eu/wiki/ACV_Software_Development_Environment`

- Every line matters
- Do not skip over anything
- If you follow the instructions faithfully, everything will work
- If you don't ... expect trouble and frustration

Software Development Tools for Exercises and Assignments

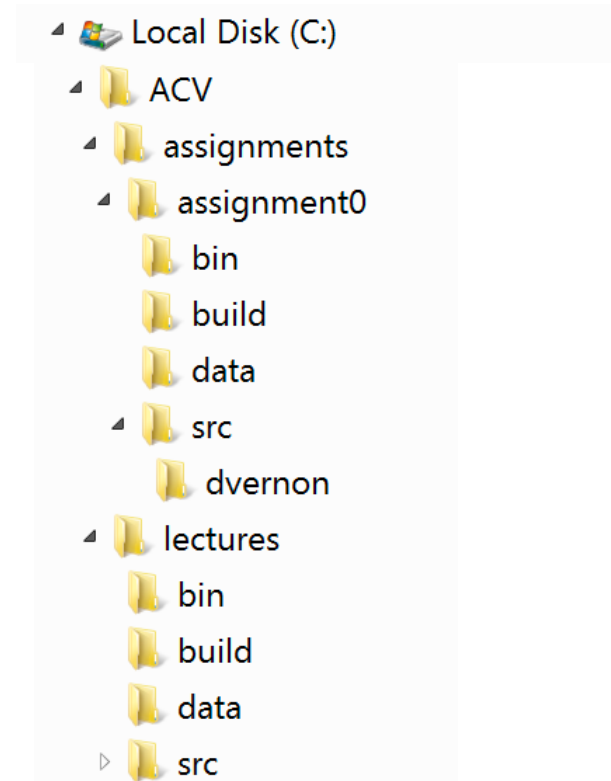
- Installation of software development environment

`www.vernon.eu/wiki/ACV_Software_Development_Environment`

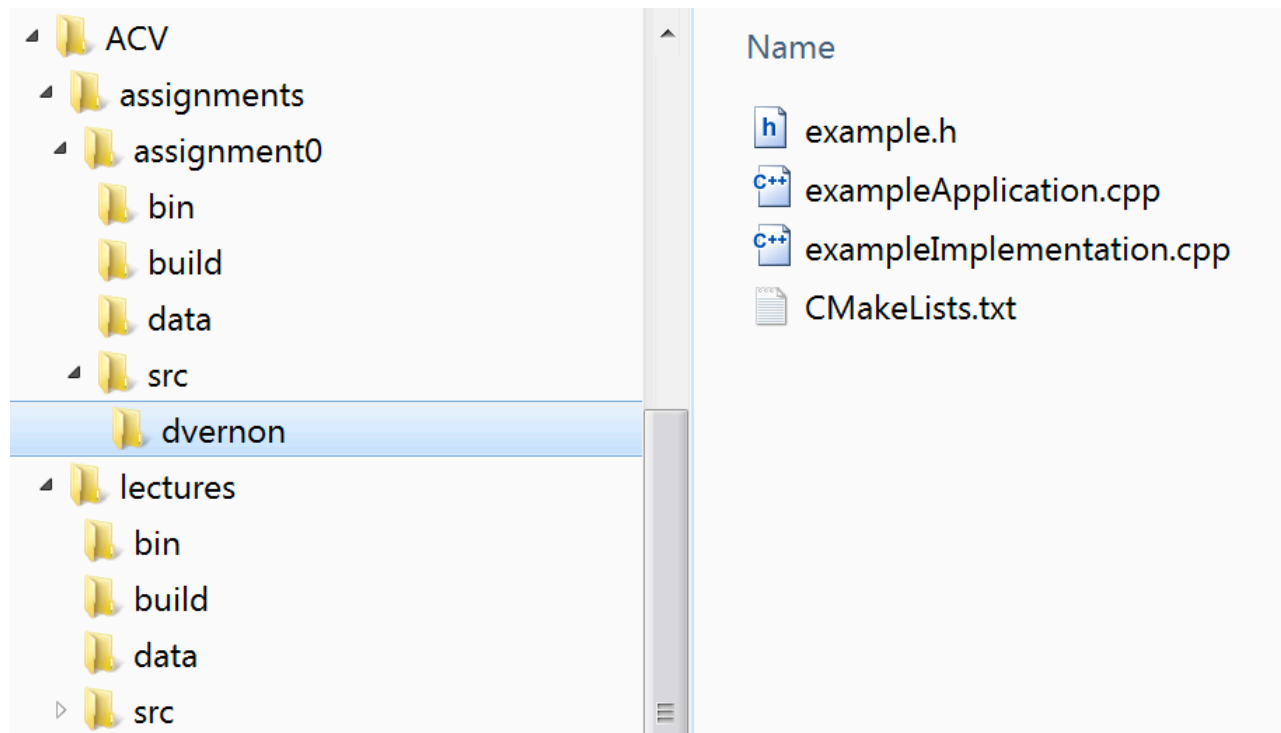
- “C:\ACV” Applied Computer Vision Repository
- Fixed file organization

- Let’s walk through the process to compile and run the program in

`C:\ACV\assignments\assignment0\dvernon`



Software Development Tools for Exercises and Assignments



Software Development Tools for Exercises and Assignments

- Preferred practice for software that supports encapsulation and data hiding (e.g. ADT & OO classes)
- 3 files: **Interface**, **Implementation**, and **Application** Files
 - Interface
 - between implementation and application
 - Header File that declares the class type
 - Functions, classes, are declared, not defined (except inline functions)
 - Implementation
 - `#includes` the interface file
 - contains the function definitions
 - Application
 - `#includes` the interface file
 - contains other (application) functions, including the `main` function

Software Development Tools for Exercises and Assignments

When writing an application, we are ADT/class users

- Should not know about the implementation of the ADT/class
- Thus, the **interface** must furnish all the necessary information to use the ADT/class
 - It also needs to be very well documented (internally)
- Also, the implementation should be quite general (cf. reusability)

Software Development Tools for Exercises and Assignments

- For your first assignment, you will simply copy the `assignment0` directory to `assignment1` and follow a similar compilation procedure, writing new assignment-specific code.
- There is just one thing you need to do: edit the

`C:ACV\assignments\assignment1\CMakeLists.txt`

and change the project name from `assignment0` to `assignment1`, viz:

```
#####  
PROJECT(assignment0)  
#####
```

Becomes

```
#####  
PROJECT(assignment1)  
#####
```

Software Development Tools for Exercises and Assignments

When submitting an assignment, all you have to do is submit a zipped version of your `myid` directory with

- Three source code files
- `CmakeList.txt` file
- Test data files
- External documentation (if required)

Do not include sub-directories (e.g. `src`, `data`, `build`, `bin`, ...)

Exercises

1. Install software development tools (should have been done already)

2. Download the current copy of the ACV repository

`http://www.vernon.eu/downloads/ACV.zip`

3. Unzip and merge with your existing copy (if you already have one)

4. Run Cmake against

```
C:\ACV\assignments\assignment0 &  
C:\ACV\assignments\assignment0\build
```

5. Open

```
C:\ACV\assignments\assignment0\build\assignment0.sln
```

Exercises

6. Compile and install the project
7. Run `C:\ACV\assignments\assignment0\bin\dvernon.exe`
8. Create, compile and run a new program in

`C:\ACV\assignments\assignment0\myid` **Replace with your Id**

Exercises

9. Run Cmake against

```
C:\ACV\lectures & C:\ACV\lectures\build
```

10. Open

```
C:\ACV\lectures\build\lectures.sln
```

11. Compile and install the project

12. Run some of the 52 examples in `C:\ACV\lectures\bin`

Exercises

backgroundModelGMM

backgroundModelMedian

backgroundModelRunningAverage

backgroundModelSelectiveUpdate

backgroundModelStatic

binaryThresholding

binaryThresholdingAdaptive

binaryThresholdingOtsu

cameraCalibration

cannyEdgeDetection

chamferMatching

colourHistogramBackprojection

colourHistogramMatching

colourSegmentation

colourToGreyscale

colourToHIS

connectedComponents

contourExtraction

faceDetection

featureExtraction

fourierTransform

gaussianFiltering

geometricTransformation

grabCut

harrisCornerDetection

histogram

histogramEqualization

houghTransformCircles

houghTransformLines

imageAcquisition

imageSamplingAndQuantization

kMeansClustering

laplacianOfGaussian

localAveraging

logPolarTransform

medianFiltering

morphologicalErosion

morphologicalDilation

morphologicalOpening

morphologicalClosing

morphologicalColourClosing

morphologicalColourOpening

opticalFlowFarneback

opticalFlowLucasKanade

personDetection

siftFeatureDetection

siftFeatureMatching

snakes

sobelEdgeDetection

trackingKalmanFilter

trackingMeanShift

templateMatching