

CSC 2/412: Human-Computer Interaction
Homework 4 (to be worked with your final project group)

Assigned: 22nd of February, 2017

Due: 1st of March, 2017 (Before class)

Total number of points: 100

This homework has two components: 1) coding; 2) written

What to submit?

- Please submit your homework through this link: <https://goo.gl/forms/e83levwSZxbzBQi32> (you should zip your project folder and written answers as a pdf document, put it in the public folder of your dropbox or Google drive and provide us the link).

1) Written part:

Assigned reading:

<http://affect.media.mit.edu/pdfs/12.McDuff-etal-Crowdsourcing-TAC.pdf>

1. Try it:

- go to <https://labs-portal.affectiva.com/portal/web-demo>
- You can either opt-in and have your face recorded while watching an ad (click on “Affdex your Vote”), or opt-out and click on the “view results” option to see the results.
- If you clicked on the “Affdex your vote” option or “view results”, what video did you watch?
There are 5 videos available:
 - a) Finger licking
 - b) M & Ms
 - c) Puppy love
 - d) Office thief
 - e) Muppets
- For the video that you watched, write a brief paragraph summarizing the results for dimensions such as “surprise”, “smile”, “concentration”, “Dislike”, “valence”, “attention”, “expressiveness”, “age bands”, “gender”, “have you seen this ad before” and “location.” Is there any surprising finding?

2. Based on your reading of this paper,

<http://affect.media.mit.edu/pdfs/12.McDuff-etal-Crowdsourcing-TAC.pdf>, please explain how the technology works for it to capture and analyze facial expressions using a browser.

3. Did you turn the camera on or not in question 1? If you had a camera and chose not to turn it on, why not? If you turned it on, why was this fine?

5. Cameras are ubiquitous these days. You see them to ensure security (e.g., ATM's and transportation lobbies), safety (e.g., cameras in the back of the car as you back up, or cameras monitoring whether the drivers are drowsy) and even for entertainment (e.g. greeting cards that catch snowflakes when you stick out your tongue).

(a) Give an example where having a camera monitor facial expressions raises an ethical concern in your mind. (b) Why is it a problem? (c) How can this concern be mitigated or alleviated?

5. If your computer could recognize three unique facial expressions, what would you want them to be? Please justify your answers.

6. Look at the picture of Monalisa and write down the occurrence any possible action units (au):

Picture of Monalisa: <http://goo.gl/LNVxc>

details on action units: <http://www.cs.cmu.edu/~face/facs.htm>

2) Coding: Head Movements and Facial Expression Detection from Live Video Stream

In this homework, you'll learn to detect head movements (yes, no, and indian nod) and facial expressions. If you don't know what "Indian Nod" is, check it out from here:

https://www.youtube.com/watch?v=0RaBxH_MKQI

This tutorial assumes Windows Platform and a built-in webcam. (If no one in your group has a windows machine, please try the computers available at the library and at Rettner Hall. Please email the TA if you have further problems.)

Required Software

For this assignment you will need the following pieces of software:

- a) **Visual Studio Community 2015 (Free to download - recommended)**

<https://www.visualstudio.com/downloads/>

You will need to create a Microsoft Account if you don't have one.

(Optional) Often times, working in a group project require some form of version control. If you are unfamiliar with github (<https://github.com/>), it is advisable to learn to use it for this project. It will make working on your final project with your teammates a much smoother process.

You can also try using the Visual Studio Team Services
(<https://www.visualstudio.com/team-services/>) (Visual Studio's own team support) for collaboration.

b) **Jason Saragih's Face Tracking Code:** Please download it from here: <http://goo.gl/Y5kGe3>

Instructions

- Download and extract the face tracking code. We'll call this folder <Tracker>.
- You can use 7-zip(<http://www.7-zip.org/download.html>) or other similar programs to extract the code
- Open <Tracker>\msvc\FaceTracker.sln [please replace <Tracker> with the actual path to facetracker code]
- Right click on the “Tracker” project from the “Solution Explorer” pane on left as shown in the following picture and select Build.

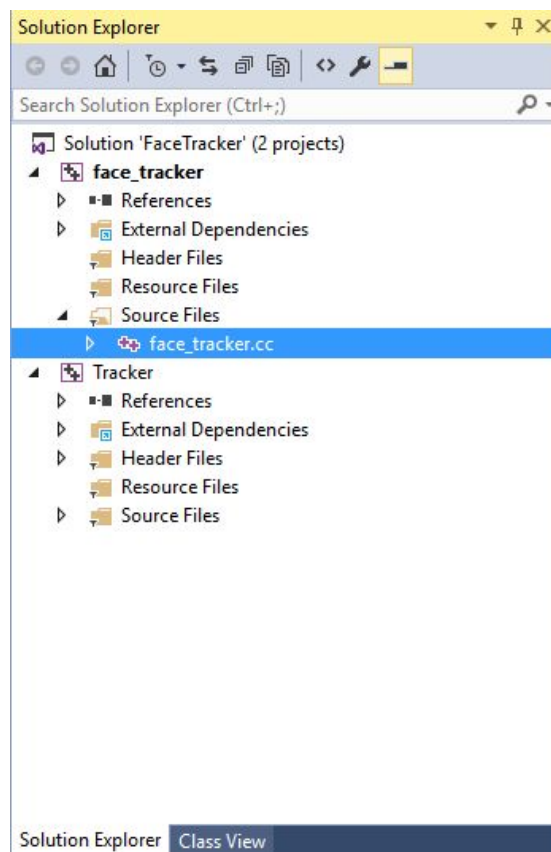


Figure 1: Solution Explorer

- You may be prompted with missing features, please install them when prompted and restart the program.
- Similarly, build the face_tracker project by right clicking on the face_tracker project. Be careful that you need to build the “Tracker” project BEFORE compiling “face_tracker” project.
- Make sure your webcam is properly attached in your computer
- Right click face_tracker and click "Setup as Startup Project" if face_tracker is not bolded



Figure 2: Run Configuration

- Make sure you are in release mode.
- Press F5 to run the program
- If you see a face tracker window with a mesh on top of your face then the face tracker is running

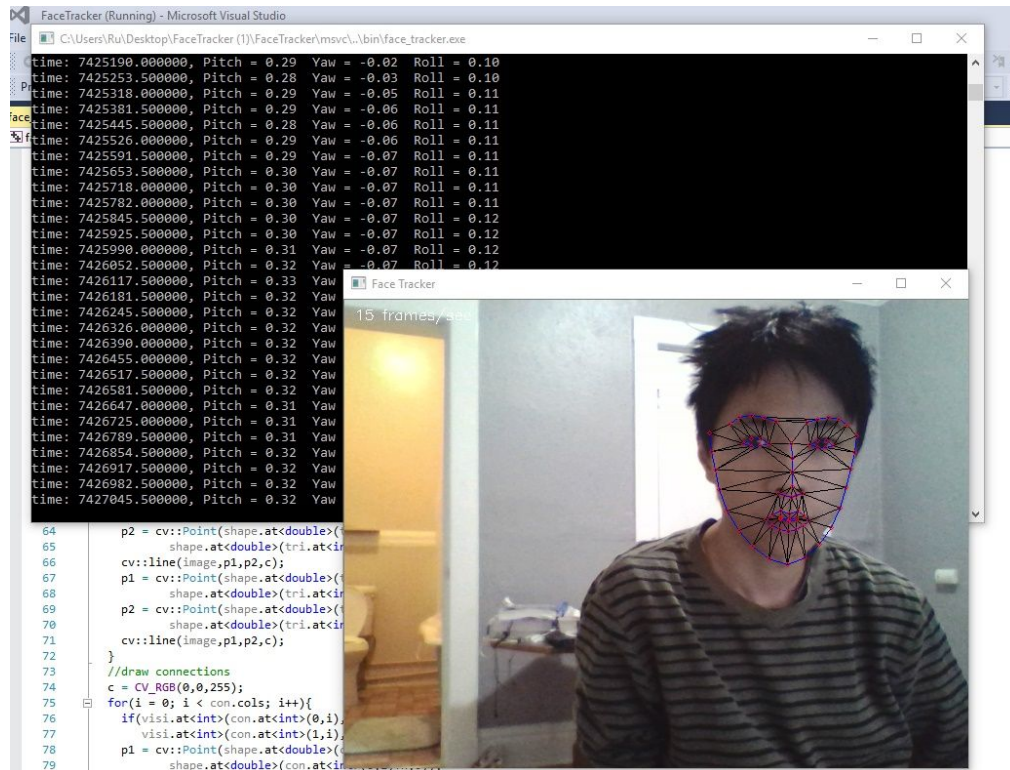
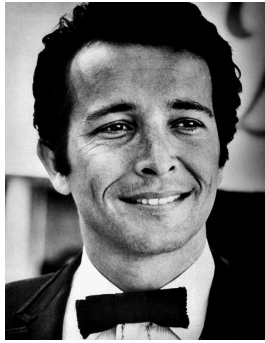


Figure 3: Face tracker running

Congratulations! You've setup your system for the homework

Homework

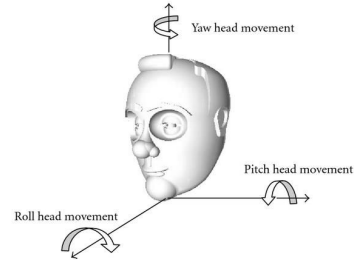
- Yes, no, and indian nod are just changes in the pitch, yaw and roll angles of the head. The code already extracts those values. Modify the code so that when a person shows those gestures, your code prints "Yes", "No", or "Indian Nod". Be careful of cases when a person just slowly moves the head without actually showing a gesture.
- The code also extracts 2d coordinates of some landmark points in face. Use those points to detect prototypical smile and surprise cases as described below
 - **Prototypical Smile:** When the two corners of the lips get far apart as shown in the following picture



Smile



Surprise



Pitch, Yaw and Roll

- **Prototypical Surprise:** When the eyebrows are raised and mouth is wide open as shown above.
- Make sure that your smile and surprise detection algorithm is not affected by just movements in the head

Note: Prototypical smile or surprise may not actually look like spontaneous smiles or surprises. But that is okay.

Grading

- We will copy and paste the code that you have written in our own project and see if it performs the recognition.
- You will get full credit if your program runs and recognizes head gestures.
- If you were unable to get your program to work, please write a page explaining the process you followed so that we can give you partial credit.
- Create a 1 minute YouTube video showing the recognition results (it does not mean that you just take a bunch of pictures and make a slide show).
 - Example videos from last year. <https://goo.gl/8e1mux>

Troubleshooting (If necessary)

However, the previous steps might not work as intended. If you see any error message like the following, you need to do further troubleshooting

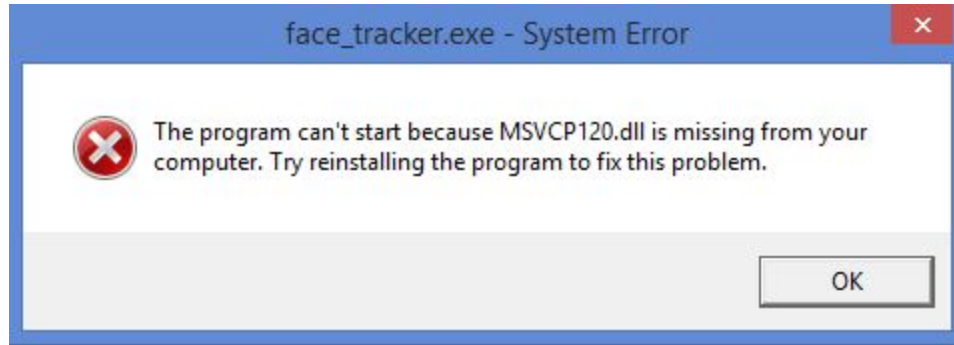


Figure 3: Error message

Step 1: Redownload the facetracker code and recompile following the previous steps. Be sure to completely delete the previous one you downloaded and not only replace the files inside. This has resolved the issue in the past.

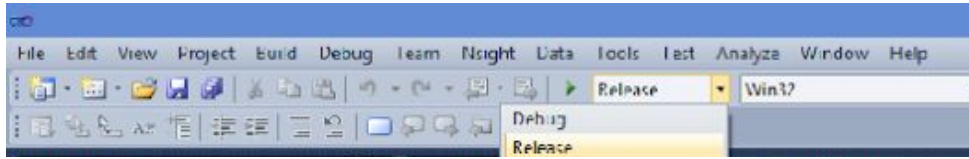
Step 2: If you are not using Visual Studio Community 2015 and are using the professional version, make sure it is the VS 2013, 32 bit version. If you are using the 90-day trial version offered on the website, you may have to recompile opencv and update the dependencies. If it is the correct version of visual studio and you still encounter the problem, you'll need to re-compile opencv for your computer. The following section mentions how to do it.

Compiling OpenCV

OpenCV comes with prebuilt binary libraries for windows. So you should be able to skip this step. However, to avoid frustrations arising from weird problems, it is always recommended to re-compile opencv for your own version of visual studio. The following instruction describes how to compile opencv.

- **Download OpenCV:** Download opencv-2.4.9.exe, right click on it and select "Run as administrator". Show a suitable location for your OpenCV Installation. I usually put "C:\Program Files (x86)" (without quotes) in the "Extract to" field. This will create an "opencv" folder in the path you've shown. From now, we'll call this folder as <opencv> folder.
- **Install CMake:** Install CMake Windows binary installer from <http://www.cmake.org/>
- Open cmake-gui. Put your <opencv>/sources folder path in "Where is the source code" field and <opencv>/sources/msvc path in "Where to build the binaries" field. Do not forget to replace <opencv> with your opencv path.
- Click Configure and select your version of Visual Studio (10 means 2010, 11 means 2012 and 12 means 2013. Do NOT select win64 version)
- Click "Generate"
- This will create the visual studio solutions inside the <opencv>/sources/msvc folder. Go to that folder and open OpenCV.sln using your visual studio
- Go to Build menu and click "Build Solution". Let the compilation finish. Do not worry if some packages are not compiled correctly (For my case compilation failed for 15 projects). You'll not need those projects for this assignment.

- Change the solution configuration to “Debug” or “Release” as shown in the picture below and build the solution again



- This will compile opencv with your own version of visual studio. This compilation will create a set of .dll files and .lib files. The dll files are located inside the <opencv>\sources\msvc\bin folder. Lib files are located inside <opencv>\sources\msvc\lib folder.
- Copy the following files from <opencv>\build\x86\vc??\lib folder to <Tracker>\opencv\lib folder. Make sure you replace “??” by the correct version of your visual studio (10 means 2010, 11 means 2012 and 12 means 2013)

- opencv_core249.lib
- opencv_core249d.lib
- opencv_highgui249.lib
- opencv_highgui249d.lib
- opencv_imgproc249.lib
- opencv_imgproc249d.lib
- opencv_objdetect249.lib
- opencv_objdetect249d.lib

Also, copy the following files from <opencv>\build\x86\vc??\bin to <Tracker>\opencv\

- opencv_core249.dll
- opencv_core249d.dll
- opencv_highgui249.dll
- opencv_highgui249d.dll
- opencv_imgproc249.dll
- opencv_imgproc249d.dll
- opencv_objdetect249.dll
- opencv_objdetect249d.dll