Term Project Guideline (ver-6).

Goal:

The goal of Term-project is to offer you the opportunity to design and implement a real working system of computer vision, and to gain first-hand experiences with primary research methods.

The Term-Project will involve some implementation and experimentation as well as a written project-report (8-12 pages long A4 PDF). In solving your term-project, you are asked to present solutions by combining existing methods or proposing new solutions.

There are 4 candidate project topics to choose from, which will be described below.

It is important that you decide on the project/topic that you want to work on (before the due date--- which is only one week away from now -- check the Study-Plan for the exact time), and then focus on the project throughout the semester.

Later change of mind is strongly discouraged, and each of the later changes will result in a loss of 20% marks of the full mark for your term-project.

You may program in Matlab (preferred), or C/C++, or java. In either case you may call OpenCV functions if you want, though this is not necessary. Other programming languages are also acceptable as well, such as Pascal, Java, Fortran, Basic, Lisp, Python, or Prolog.

Operating system: Windows (XP/7) is preferred. Tutor may test your source code under Windows.

Group Policy:

Term project is a group project, and each group must have 3~4 students. Teamwork is necessary and encouraged, however, each individual's contribution ratio will be reported by your team-members confidentially to tutor.

Assessment:

(1) Final Project Demo: 15% (During your demo session, Tutor will ask technical questions, and your answers will affect your final project score).

(2) A 8-12 pages of Project Report (in PDF format) + commented source code (in Zip file, with a "readme" file, demo entry point: Term-project-group-ID.m): 15%.

The final mark of your Term project will be based on the following components.

(a) the technical results, and the demonstration of your project;

(b) the efforts and thoughtfulness that you have put into the project;

(c) any innovation/creativity that you have proposed, and/or implemented.

(d) teamwork and collaboration (based on the confidential individual contribution report).

Your 8-12 page Project-Report must contain the following contents:

1. Introduction (problem statement –what, why, with real-world applications)
2. Related Work (What has been done, according to literature survey, with a few references)
3. Your Approach (How you are going to do it, with concept design, method, algorithms, equations, figures)
4. Your Implementation and Experiment results (What you do, what are your results, with images, tables, and numbers)
5. Your Conclusions (Itemized conclusions, observations and discussions, future work)
6. Learning outcomes: give your at least 3 points about what you have learned from doing this term project.

Don't worry if you fail the project task, because you may get credits for the other components (such as the report, teamwork, efforts,...).

Descriptions of the 4 optional Term-Projects are given in the next Page.

Project-1:   3D Modelling/Reconstruction.

Background:

In this project, you are given 24 images of a 3D rigid object (Here is an action figure: Morpheus in "the Matrix"),   along with 24   camera matrices each of which gives the camera projection matrix (of size 3x4).   Also provided are 24 silhouette apparent contours, one for each of the images.   (Project idea and dataset credit: courtesy of Yasutaka Furukawa and Jean Ponce of UIUC).

...

File formats

Images are provided in the JPEG format.

The input calibration file is a simple ASCII file containing the 3x4 projection matrices as blank separated floats, in sequence for all of the N views.

An **apparent contour** is represented as a chain of points in our data sets (a piece-wise linear structures) and provided in a simple format. Each file starts with a single line header and an integer representing the number of apparent contours in the corresponding image, which are followed by the data sections of apparent contours. Note that a single image may have multiple apparent contours. A data section of an apparent contour starts with an integer representing the number of points in the component, followed by their actually 2D image coordinates. Points are listed in a counter clock-wise order for apparent contours containing foreground image region inside. Similarly, points are listed in a clock-wise order for apparent contours containing background image region inside (holes).

Your Task:

If you choose to work on this project, then the task is to develop a program that can take the images as the input, and output a 3D mesh model of the object.
You must represent your final reconstructed 3D model as a triangular mesh model and display it on screen using Matlab, or any other suitable free VRML-viewing program (such as FreeWRL).

Hint: Depending on the 3D-Reconstruction method you plan to use, not all the files are necessary. For example, if you are to use 3D point cloud structure from motion, then the apparent contours are of no use.   If you are using visual-hull method, then these contours are important.   **Bonus points:** By default, your reconstructed 3D model will be everywhere gray in colour. However, if you manage to colourize or texture mapping your 3D model correctly, then you will get up to 5 extra bonus points (as judged by your tutor).

Project-2: Fruit counting: how many pears/bananas are there ?

You are given 6 video clips, and you task to program a single program that,

(i) read in the video clip,

(ii) display the video frames,

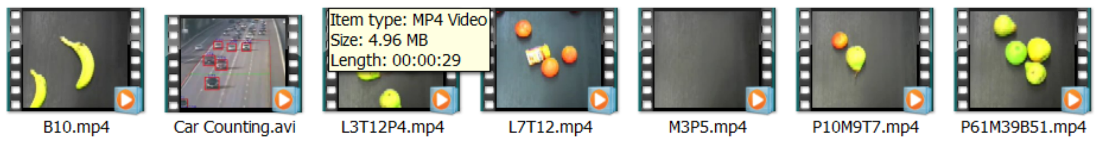
(iii) detect fruits, draw a bounding box on each fruit

(iv) tracking the detected fruits until they disappear.

(v) count the number of fruit of each type, and update the counting result in real-time.

(vi) At the end of each clip, report the success rate (100%) of your program.

(You can manually/visually obtain the ground-truth result.)



You program should be able to detect, recognize and count at least 3 type of fruits types. You decide which three are the targets to detect, recognize and count. (There are probably five types shown in the videos, i.e. Apple, Orange, Banana, pear, Tomato.)

Requirement: Your program must perform in near real-time (e.g., process 5+ frames per second.)

(Hint, this system is in fact a toy-example of car classification and car counting module in a computer-vision-based traffic surveillance system. In the zip file you will find an AVI video, which gives such an example. Your program's GUI (user interface) should look similar to this one. **Bonus point:** if you also report each fruit's size (using its region areas), you get up to 3 bonus points. Btw: I consider tomato as a kind of fruit rather than vegetable.)

Project-3: Angry birds: Spot the pigs.

(Project idea credit: Dr Jochen Renz, Dr Steve Gould: Angry Bird Aus-AI Contest;**)**

In this project, you are given a video clip of the "angry birds" (in game trailer).

You are asked to code a computer vision program that is able to

(i) read in the video clip,

(ii) display the video frames,

(iii) detect the pigs, draw a flashing bounding box on each pig.

(iv) Display the x-y coordinates of each pig, relative to each image frame's origin (i.e. the top left point).

(v) Count the total number of pigs that your program has detected. Report the counting accuracy rate (100%) against the ground-truth. (You can visually count and get the ground-truth).

Requirements:

1. Your pig-detector must adapt to rotation and scale change.

2. Your program must perform in near real-time (e.g., process 5+ frames per second.)

Bonus point: In step (iv)**,** if you also report the coordinates of the detected pig relative to the slingshot correctly, then you will get up to 5 bonus points.

Programming language: Matlab, or Python, or C/C++, or Java.

Project-4: Butterfly Recognition.

(Dataset credit: courtesy of Jean Ponce of UIUC)

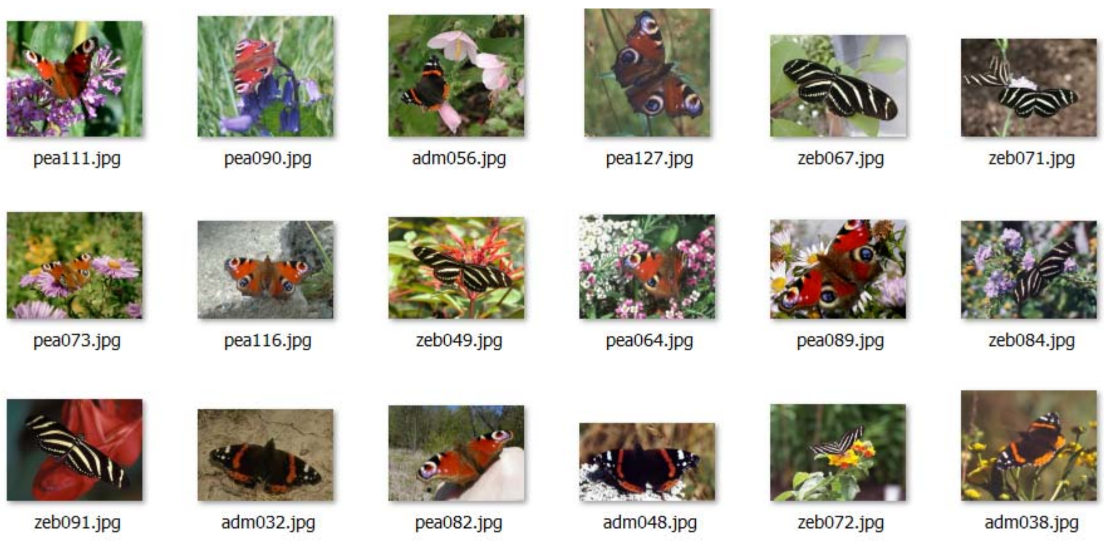
In this project, you are given 150 training images of 5 species of butterfly (30 images per specie), and 200 random test butterfly images of unknown species.

You are asked to implement a butterfly recognizer that can

(i) Read an image.

(ii) Report the name of the butterfly.

(iii) Report the average recognition rate (in 100%) out of all the 200 test images.



Bonus points: You will be awarded up to 5 bonus point if your program can automatically detect where the butterfly is, and draw a very-tight rectangle bounding box around it.

================= Have fun ! ====================

(Project designed by HL/YM for 2013 S1 students in ENGN4528/6528, with the helps of other colleagues at CECS ANU and NICTA Australia).