159.731 Semester 1 - 2020

Computer Vision Mini-projects:

The mini-project: The mini-project consists on *your own implementation of a algorithm* or a *combination of algorithms* to solve a particular problem in computer vision. It can also be a modification of an existing implementation (open source). The (modified) source code also needs to be submitted to Stream. Preferably the mini-project should be developed with groups of two students.

The report:
You should prepare a small report, similar to a conference paper (**between 8 to 10 pages** using the given template). Use the report template (either Word or Latex) available on Stream.

The report **must** cover the following sections:
- *Abstract (short description of the project, summary of the results)*
- *Introduction or Background (a mini-literature review)*: a ***discussion*** about a few articles/books that presents the related topics (remember to *cite* your sources appropriately).
- *Methodology*: ***describe***, with your own words, the methods and/or algorithms presented.
- *Results:* ***show*** results you obtained and ***compare*** to the results from the literature.
- *Discussion:* ***advantages*** and ***limitations***, comparisons with other methods etc.
- *Conclusions:* your conclusions regarding the effectiveness of the method, ***what has been learnt***, future work etc.
- *Bibliography*: list the relevant articles and/or books you have used. Use a citation style (see the template instructions on Stream).

The presentation: The date for the presentations start at week 10 (18/May), during the normal time of classes. Prepare a short and informal presentation showing the main results. Use any resources you want, such as slide presentations, white board, etc. The presentation should last no more than ***20 minutes***. An important part of the presentation is to carry out a short ***demonstration*** of your code implementation.

The seminar carries ***40%*** of the total marks for the paper, and are distributed as follows:
- Presentation (*you get it automatically for presenting*). Work in progress is ok at this stage! 10%
- Code and results (originality of the results, well structured experiments, it works according to what is described in the report) 15%
- Report (scientific honesty for the results, appropriate citation, figures and tables relevant to the text, following the guidelines and descriptions) 15%

Due date for the reports: 19/06/2020

Topics

Methods and/or applications for computer vision that we did not discuss in details during our classes are topic candidates. The key is not to implement everything from scratch, but to experiment with different methods, compare two or more methods or explore a different way of using a certain method.

After choosing a topic, please approach the lecturer to formalise your choice and to confirm that there is no other group with the same topic.

Brief description of the topics available in 2020:

Depth maps with stereo vision: Measure and compare the real distances with that of the stereo vision algorithms. Change the distance between the two cameras to see how to improve the accuracy (and by how much).

High dynamic range approach: Implement a simple colour classifier based on multiple cameras based on HDR. Compare the results using one camera against more than one.

Traffic sign recognition: Use a well known database to train your own approach to recognise traffic signs. Compare the results.

Corner detection: Implement 3 different corner detection algorithms: Harris, Moravec and FAST corner detection. Use an image database to compare the corners detected and vote on the best one for those images.

Wavelet compression: Implement a simple wavelet approach to compress different greyscale images. Compare your implementation against JPEG and JPEG2000.

Texture analysis: Implement a and train a simple approach to classify texture within images. Using the simple classifier, segment grass, clouds and sea from images.

Video tracking (mean shift): Using the mean shift approach, use different videos to measure how well it can track objects. Try different objects (e.g., cars, athletes and balls in sport).

Optical flow: Implement two optical flow approaches, one using dense optical flow, the other using a feature based optical flow. Compare the two approaches in terms of accuracy and runtime.

Generalised Hough transform: Using Ballard's approach, specify R-tables for three different profiles (e.g., keys). Show the results of trying to detect these same profiles on clean and cluttered images. Show also the runtime data for all the detections.

Histogram of Oriented Gradients (HoG): Use this technique to detect pedestrians on a scene. Measure the accuracy on a public dataset as well as test it with the webcam to measure the frame rates.

Deep Learning and Computer Vision: Use an existing library to train a particular set of objects. Demonstrate that the application can recognise that object and analyse the performance using a camera.

Face recognition method comparison:
https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html

If you don't like any of these projects, you can also propose one based on information you read on articles, books or the web. As long as it is within the scopes of image processing and computer vision and it is not covered by the study guide, it may be a valid topic. Talk to the lecturer about your proposal.