**University of Alberta**
**Department of Computing Science**
**MSc with Specialization in Multimedia**
**Winter Semester 2021**
**MM805 Computer Vision**
**Programming Assignment**
**Assigned on January 25**
**Deadline by Feb 26**

Implementation can be done in Python (OpenCV) or Matlab (Python preferred), along with a README to describe how to run your work.

**Image Stitching and Perspective Correction (30 max pts)**
The goal of this part for the assignment is to have full understanding of how homography is computed and how can it be used in different applications.
There are three components to be implemented for this part:

- DLT Algorithm: **10 points**
  - Provided Input:
    - Images from License Plates Dataset [1].
    - License bounding boxes groundtruth provided in numpy file.
  - Expected Deliverables:
    - dlt, dltnorm files and the eval script calling these that shows warped images.
  - Notes: It is not allowed to use third party library for this like OpenCV, if opencv method is called it will not be marked.
  - Parts:
    - Implement DLT algorithm as explained in Lecture 3. **6 points**
    - Implement Additionally the normalized DLT version as explained in Zisserman Book Page 109 Algorithm 4.2. **2 points**
    - Use your implementation to perform perspective correction to license plates. **2 points**
- Feature Extraction and Matching **10 points**
  - Provided Input: Images from Adobe Dataset [2].
  - Expected Deliverables: detect, and featurematch scripts.
  - Notes: Use OpenCV or other third party library for this part.
  - Parts:

[1] https://www.kaggle.com/andrewmvd/car-plate-detection

[2] https://sourceforge.net/projects/adobedatasets.adobe/files/adobe_panoramas.tgz/download

        - Pick and Implement one of the feature detectors we have explained or use other methods you learned about. **4 points**
        - Use one of the feature descriptors and implement a simple feature matching of your choice. **5 points**
        - Visualize the detected and matched features. **1 point**

- Compute Homography Overdetermined System **10 points**
    - Provided Input:
        - Images from Adobe Dataset.
    - Expected Deliverables: ransac, lmeds scripts and visualize the output from image stitching.
    - Notes:
        - Implement your own RANSAC and LMedS Do not use OpenCV for this.
        - Your implementation will not perfectly match OpenCV's as they use Levenberg Marq. Refinement which is out of our scope.
    - Parts:
        - Implement RANSAC as explained in Lecture 3. **6 points**
        - Modify your RANSAC implementation to the Least Median variant, which is similar to RANSAC but does not use a fixed threshold. Explained in [3]. **2 points**
        - Perform Image warping and stitching using OpenCV or third party library for the warping (not computing homography) and visualise. **2 points**

[3] http://www-sop.inria.fr/odyssee/software/old_robotvis/Tutorial-Estim/node25.html