# CS 763 Assignment 5

Due Date: 21 April, 23:00

April 9, 2019

---

The relevant skeleton code and directory structures for this assignment can be found in this link.

1. **Kanade Lucas Tomasi (KLT) Feature Tracker**
   As a part of this assignment you will implement KLT tracker. Please follow the following instructions:

   (a) Read the frames from `input/` folder. Display major features points overlay-ed on the first frame, for feature point detection you can use Harris corner detector or SURF; both are inbuilt in MATLAB. [**10 points**]

   (b) Select those feature points with good structure tensors (recall the class). There can be multiple points qualifying the criterion; so, define a parameter to choose how many feature points are to be tracked. [**5 points**]

   (c) Take a template patch centered around every feature point. The patch size is a tunable parameter, but, ideally, you should start with 40 pixels.

   (d) Use an affine motion model for creating the Jacobian matrix, resulting in a total of 6 parameters to be estimated.

   (e) Now, for every frame:-

      i. Begin with an initial estimate of the affine matrix.

      ii. Warp and crop part of frame based upon the initialized/updated parameters of affine matrix. [**5 points**]

      iii. Calculate the L2 error between the warped patch and the template patch. [**5 points**]

      iv. Compute the image gradients of the warped patch. [**5 points**] (tip: try smoothing the gradients)

      v. Compute the Jacobian and Hessian matrices and update the parameters of motion model. [**10 points**]

**10 points for quality of tracking**
**Note:** The solution is iterative, so use a threshold and max iteration parameters for convergence check. Actual trajectory of the motion will be obtained by storing the relevant parameters of motion model from each frame. Also, after every 10 to 20 frames you should extract a new template, since the patch may have changed a lot. Finally your script should dump the overlay-ed trajectory on image in **output** folder frame by frame. Also, you can use inbuilt functions for image warping and feature detector. **[50 points]**

2. In this assignment, you will implement a software routine to stabilize a video. Videos acquired by handheld cameras or smartphones often appear jerky or shaky due to inevitable motion of the hand during acquisition. This artifact is exacerbated in videos acquired from handheld devices while inside a moving vehicle. The process of removing the unwanted motion between consecutive frames (while maintaining or preserving the intended motion) is called as video stabilization.

   (a) Use `mmread()` to read the video files provided in the input folder. Take a look at `myMainScript.m`. Also, the folder contains a MATLAB function called `displayvideo.m` which takes a video in the form of a 3D array and displays it at a specified rate.

   (b) You now have to estimate the motion between frames $n$ and $n-1$ ($1 < n \leq T$) of the shaky video. For this, you should use the SURF algorithm (builtin MATLAB to (1) detect salient feature points in both the frames and (2) determine matches in between the points of those frames. Now, given this set of matching pairs of points produced by the SURF package, your first task is to estimate the motion in between them - which will (hopefully) be the same as the motion between the frames. You should perform this using RANSAC. Repeat this for all pairs of consecutive frames and generate a motion sequence. A motion sequence is a sequence containing the motion parameters at every frame. For instance, assuming a translation+rotation model, the motion sequence acquires the form $\{(t_x^{(n)}, t_y^{(n)}, \theta_x^{(n)})\}_{n=2}^{T}$. **[20 points]**

   (c) For a shaky video, this motion sequence will be very noisy. You should smooth the sequence using a simple averaging filter to generate a smoothed motion sequence. The width of the averaging filter is a user-choice. Make sure your averaging filter is wide enough or the amount of smoothing may be inadequate. Plot two examples of noisy and smoothed sequences in your report for every motion model you experiment with. **[10 points]**

   (d) Finally, generate a stable video by using the motion estimates between the frames. Since you have the jittery as well as the smoothed parameters, figure out a way to negate the parameters contributing to the shake/jitter. View the shaky and stabilized videos together by (1) putting them in a single array of size $2H \times W \times T$ where $(H, W)$ is the size of each frame and $T$ is the number of frames, and (2) using the routine 'displayvideo' mentioned earlier. Also your MATLAB program should write the combined video to a file and give it a sensible, self-explanatory name that you print on screen. **[20 points]**

   **Note:** While submitting please delete the `mmread` folder otherwise you wont be able to upload on moodle.