**HW#6 (Lucas-Kanade)**

* Implement Lucas-Kanade feature point detection and tracking.
  + *Detection.* For each pixel in a grayscale image, construct the 2x2 covariance matrix of the gradients in the 5x5 window surrounding the pixel.  Then compute the minimum eigenvalue of the gradient covariance matrix for each pixel.  Perform non-maximal suppression to detect the n most salient features, separated from each other by a distance of at least k pixels, where n=100 and k=8.
  + *Tracking.*  For each feature, track its location from one image frame to the next by iteratively solving the Lucas-Kanade equation Zd=e, where Z is the 2x2 gradient covariance matrix and e is the 2x1 vector of gradients multiplied by the temporal derivative.  Display a movie of the original images with features overlaid.  You will want to smooth the images first by convolving with a Gaussian to increase the basin of attraction, particularly to handle swift camera motion, and you should use a large window size, e.g., 11x11 or 17x17, for the same reason.  For more details, you may want to refer to Jean-Yves Bouguet's [technical report](http://robots.stanford.edu/cs223b04/algo_tracking.pdf) (but ignore the pyramidal part) or the [KLT references](http://www.ces.clemson.edu/~stb/klt/).  Keep your feature coordinates as floating point values throughout the tracking process, only rounding for display purposes. Use bilinear interpolation to compute the image pixel values at non-integer locations.
* Your code should accept as input parameters the filename, start frame, and end frame. An easy way to do this is to specify the format string, then to use sprintf or CString::Format. Example:CString str; str.Format("img%04d.bmp", i);
* In a figure window, display the current grayscale image with the feature points overlaid as red dots. Call this display function within a for loop, so that a movie is displayed within the window.
* Run your code on a synthetic sequence generated by translating an image (such as from one of the sequences below) a known amount, e.g., 1 pixel to the right each frame.
* Run your code on the following image sequences:  [flowergarden.zip](https://cecas.clemson.edu/~stb/ece847/fall2012/flowergarden.zip) and \*[statue\_sequence.zip](https://cecas.clemson.edu/~stb/ece847/fall2012/statue_sequence.zip), overlaying the features on the original images.  Your code will be tested on these images. The feature points should appear as thought they are latched onto the image, not floating on top of it.
* For this assignment you may not use any of the Lucas-Kanade or KLT implementations in Blepo, or any other existing implementations of Lucas-Kanade.  You also may not use any of the Interp functions.
* No report is due for this assignment.