PURE MID SEMESTER REPORT

ECE 297 – independent studies
Mentee: Jae Yong Lee
Mentor: Joseph Degol
Research Statement

• Our research group focuses on object recognition and implementation of other computer vision related works.

• To acquire independent computer vision software, creating new form of image data type was inevitable.

• New form of image data type needs new image rendering software that accepts the image data type as input.

• Goal of my research area includes creating compatible image / video viewer object that renders original image data type created by my mentor.
Background Information

• The original image data class includes mainly 4 forms of data:
  • Width value (integer)
  • Height value (integer)
  • Channel value (integer)
  • Data array (array of [unsigned char, short, int, float or double])

• For window UI, FLTK library is used, for renderer, OpenGL is used.
  • FLTK (Fast Light Tool Kit) provides diverse form of functions used for GUI application
  • OpenGL provides easy access to graphical acceleration and texturing of the image.
What I have accomplished

- Converting general data structure of images to new image format.
- OpenGL Texture rendering for multiple image inputs using original data type.
  - Image viewer will ‘smartly’ divide width and height of image viewing box for given window width/height, and number/size of images.
- OpenGL Texture rendering for stream of image inputs using callback function and original data type.
  - Callback function will receive new block of image in the stream and texture rendered by pointer of the stream will draw video stream on GUI.
Works to be done

• Image / Video viewer should be able to accept vector of key points which represents sub-pixel-wise location of given image. (i.e. it must be able to accept floating point inputs)

• For speed and latency issues the application might need threading loops for compatibility when nesting the image viewer object onto larger application is considered.
  • This image viewer class is not only able to function by itself, but also can be nested into other FLTK widget applications.