Daily Log

Monday, September 16

Researched shadow detection methods. Looked into SVM and LSSVM classifiers, along with a few other options of identifying using neural networks. LSSVM was used to help calculate shadow probability but required being used on larger segmented images. A bit too complicated for me to fully understand and code at the moment, so I will focus more on it when I reach an image that combines the superpixels from the SLIC segmentation.

Tuesday, September 17

Researched shadow identification method using meanshift algorithm. It seems similar to k-means except no pre-specified number of segments.

Thursday, September 19

Coded meanshift algorithm using sklearn library. It is extremely slow, much slower than k-means. I may look into doing k-means twice instead of k-means then meanshift.

Timeline

Date	Goal	Met
September 8	Research SLIC algorithm and create program to segment images	Yes, but current implementation uses a pre-built library
September 15	Create program implementing SLIC without using SLIC method from pre-built library	Yes, but current implementation is much slower than the pre-built library
September 22	Identify 60% of superpixels that are shadows in the image	No. It's taking much longer than I predicted to be able to identify shadows.
September 29	Group superpixels into regions in under 2 minutes	
October 6	Train a functioning LSSVM or SVM classifier	

Reflection

Being able to identify the shadow regions in a picture is proving to be much harder than I originally predicted. It was hard to understand meanshift, especially for an RGB image, before I could start coding. I still have confusions about SVM and LSSVM and the math surrounding them, but I will cross that bridge when I get there.

At the moment, I have a functioning meanshift, but the implementation with the sklearn library is extremely slow. One run takes about an hour, so it's not ideal. It's probably due to the large size of the picture, but I will try to modify my input to the meanshift method to speed it up. I may also try to just run k-means over the super pixels from a first round of k-means. I may also look into other methods to group the superpixels into larger pixels.