What I’ve done:

* Ported all the mex TSP code from C++ to MATLAB.
* Improved efficiency by reducing function calls, using structs less, and using efficient for loops.
* Solved the problem of the long thin lines by designing and implementing an algorithm that
  + Checks to see if trying to move a pixel that is part of a line
  + Sees how big the line is
  + If it’s over a threshold, splits the line at what is most likely the base of the larger portion of the TSP
  + Moves all the pixels from the line to neighboring TSPs
    - If it’s moved more than a threshold, it decides that the bigger part of the TSP was at the other side of the line, so it goes back and splits it at the other end
    - If it finds that the TSP is a loop, it just gets rid of the line, because the TSP is connected on the other side

1. Introduction
   1. Tracking objects from motion can’t be done just looking frame-by-frame—not enough information. (show an example of frame-by-frame)
2. Use many frames
   1. Check several frames into the future and into the past to find large-scale movement. [1]
3. Look forward and backwards
   1. Use information from both backward flow (frame 2 to 1) and forwards flow (frame 2 to 3). [2]
   2. Use shape from frames in front and behind. [3]
   3. Smooth direction of movement by looking at several frames instead of just the current one.
4. Interpolate
   1. Interpolate missing frames of movement in an object. [4]
   2. Combine multiple objects that are really the same.
5. Future work
   1. Right now this only works with a stationary frame of reference—what do we want the results to be when we have a moving frame of reference?
   2. Combine these results with image segmentation.
   3. Combine the large-scale results (looking at every 5th frame) with short-term results to get a more accurate outline.