Assignment: Implementation of Congealing

February 23, 2010

In this assignment, you will implement the congealing algorithm for joint alignment of binary images.

Load a set of 3's (I recommend starting with about 20 of them) into memory in a 3-D array, where each image represents one layer. Add a final image with intermediate gray values on the end of your array. I'll discuss why this should be done in class. (It will "smooth" your entropy estimates.

Write a function called sumOfStackEntropies.m that calculates the entropy of each pixel stack in the 3-D array and adds them together.

You will be keeping track of an affine transformation for each image in your stack. As discussed in class, it will be important at each step in the algorithm to *transform the original image* rather than continuously transforming an image over and over, which will accumulate errors in the image due to pixel mixing.

You should write a function called generateTransform.m which takes a vector of 7 arguments corresponding to x-translation, y-translation, rotation, x-scale, y-scale, x-shear, and y-shear, returns the appropriate transformation that is a composition of these. To do this, I recommend first writing functions to generate transformations for translation, for scaling, for each shear, and for rotation, and then composing them like this:

$$TotalTransform = T * R * S * Sx * Sy,$$

where T is the translation matrix, R is the rotation matrix, S is the scale matrix, Sx is the x-shear matrix, and Sy is the y-shear matrix.

0.1 Outline of algorithm

The general outline of the algorithm goes like this:

- Let E be the pixel stack entropy sum of the original set of images. Each iteration of the algorithm will attempt to reduce this entropy.
- until you cannot reduce E anymore:
  - For each image:
    - For each transformation:
      - Change the current image by adding or subtracting a small amount of the current transformation parameter. Try re-inserting the newly transformed images into the image stack in place of the original image to see if it raised or lowered the entropy. Keep any change which lowers the entropy.
  - After adjusting each transformation parameter for each image, you may want to "rebalance" the parameters. This will avoid what I refer to as "parameter drift". To re-balance the parameters:
    - For translations, rotations, and shears, subtract the mean value of each parameter from the value of the parameter for each image. This will give the parameter a mean of 0.
    - For the scale parameters, divide the parameter by the geometric mean of the parameters. This will result in the scale parameters having a geometric mean of 1. Equivalently, you can make the log-scale parameters have a mean of 0 using the abovep procedure.