

# Image Registration Techniques

## Homework 6

**Due: Friday March 19 by 10am (the start of class) via email**

This homework explores Lectures 13, 14 and 15. There are only two problems, however, because we want you to begin to focus on the longer programming project. Please note that the due date is the same day as the class set aside to discuss individual progress on your longer programming project. If you will have trouble completing this homework and preparing for that class, please let me know.

Submit your code and any output images for question 2 by forming a tar or zip file and attaching this file in an email to Professor Stewart (stewart@cs.rpi.edu). Submit printed answers to question 1.

1. **(20 points)** Recall the following expression, derived in class (Lecture 13) for the negative log-likelihood of a normal distribution

$$\frac{n}{2} \log \sigma^2 + \sum_{i=1}^n \frac{(x_i - \hat{\mu})^2}{2\sigma^2}.$$

Substituting a robust  $\rho$  function for the quadratic error term we have:

$$\frac{n}{2} \log \sigma^2 + \sum_{i=1}^n \rho((x_i - \hat{\mu})/\sigma^2).$$

Starting from this expression and following the derivation given in the lecture notes, derive a reweighted least-squares estimate for  $\sigma$ .

2. **(30 points)** Write a `vx1/rgrl` program to align the range data set posted on the web page with itself (`scaled_dragonStandRight.0.txt`). The range data set is taken from the Dragon data of the Stanford repository. See

<http://graphics.stanford.edu/data/3Dscanrep/>

Assume the initial alignment is

$$\mathbf{A} = \begin{pmatrix} 0.98106 & -0.172987 & 0.0871557 \\ 0.173648 & 0.984808 & 0 \\ -0.0858317 & 0.0151344 & 0.996195 \end{pmatrix} \quad \text{and} \quad \mathbf{t} = \begin{pmatrix} 10 \\ 0 \\ 15 \end{pmatrix}$$

Estimate an affine transformation (even though the actual transformation is only rigid). Use face point features and robust estimation.

Using face points requires that you compute surface normals in the “fixed” data set (though not in the moving data set). You can do this by finding a

small set of nearest points to each point in the data set and then computing the surface normal using orthogonal least-squares. Storing the points in a k-d tree will make this more efficient. Finally, range data registration generally does not use all data points from the moving image. Sampling every 10th or even every 50th point should work here.