

## Task 1

I have a few ongoing research related tasks which could be organized on a timeline. The term project for CISC 874, Neural Networks and Evolutionary Algorithms, has given me a chance to start on the scoliosis quantification part of my work. Over the summer, among other things, I placed landmark points on virtual models of 124 scoliotic patients' transverse processes. These are the anatomic structures [Ungi2014] used to estimate scoliotic curvature, and so they will be my data. Since my interests lie in neural networks, they are the methods for curvature estimation I will investigate.

- I am in the process of writing a Python module for 3DSlicer (<https://www.slicer.org/>) which introduces 3 kinds of errors into the landmark points. These errors are meant to simulate likely imperfections in real ultrasound data. It then writes them to a .csv file in a suitable format to be read by a C++ program.
  - This has begun because it is part of my CISC 874 project, and therefore, has a deadline in early December. Although this is reflected in my Gantt chart (Figure 1), I do not expect it to be definitive. I will probably return to this program after December and try to make the simulated errors more realistic.
- Since the structure of the data reflects the structure of the neural network (one input node for each datum in a scan), and some points will be deleted in the previous program, some means of completing the data set will probably be necessary. A number of missing value estimation techniques are available including regression analysis and, again, neural networks.
  - Although this has not begun yet, its relation to my degree's timeline is the same as the previous program.
- The project then requires a neural network with which to estimate the spinal curvature.
  - I have begun writing this in C++ as it has an immediate deadline for the course. I will revisit this task later.
  - Once I create a functional framework where I can implement and experiment with alternative network configurations and architectures, I will have reached a natural rollback point. Whatever particular neural network techniques I try, they can be adopted or abandoned as classes in the program. Non-approach-specific tools constitute much of the program, which provide building blocks for alternative approaches.
- This semester, I am writing a conference paper for SPIE's Medical Imaging conference. They accepted a paper based on a summary of spinal visualization work submitted over the summer.
  - I have begun the full paper, and its deadline is early December. This is reflected on the Gantt chart.
  - After submitting the summary manuscript in the summer, I noticed that it created something like a rollback point. The submitted PDF, and all of its supporting data, images, etc... form a snapshot of that aspect of my work at that time.
- My supervisor has suggested I submit a paper to MICCAI's Information Processing in Medical Imaging conference, and my CISC 874 project is just the thing.
  - I will probably begin writing this once I am generating rudimentary results from the neural network. The deadline for this is mid-December.
- Were it not for the holidays, I don't know when I could take a break long enough to schedule.

- Having presumably developed some functional neural network program during the fall semester, and identifying that as a good rollback point, the beginning of the winter semester would be a good time to discuss progress with my supervisor.
  - This could save me from collecting useless or irrelevant data as I refine various approaches.
  - He might suggest pursuing different aspects of the project which are needed to develop a desired clinical system, changing the course of my work.
- I am unsure of my research plan next semester. It may be that my supervisor suggests a new aspect of this project, and I pursue novel topics for some time.
  - For lack of a better plan, and because I will need to improve it, I list investigating alternative network structures and functionalities as a final, ongoing task on the Gantt chart. I do not expect the path to graduation to be so direct. There's at least the matter of a thesis and defense, but placing those on the Gantt chart would just be a guess.

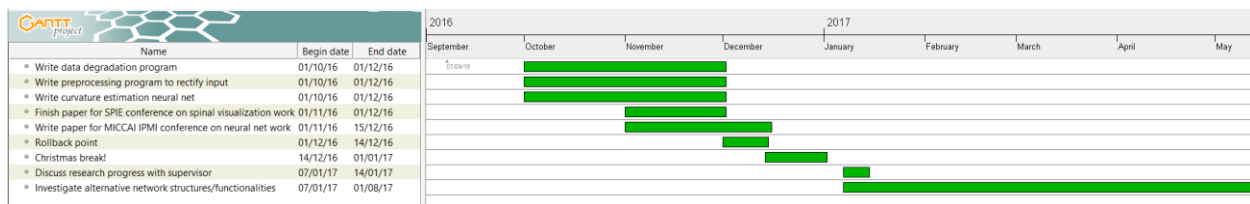


Figure 1: Gantt chart of current research plan timeline

## Task 2

The last 3 points of my Assignment #5's Methodology section list steps for which I have a number of tools to choose from.

- To apply the 3 types of error to my input data:
  - I have begun writing a Python module for 3DSlicer
    - The learning curve for this will be minimal since I used this tool throughout the summer.
  - 3DSlicer is a natural choice because it was the environment in which the input data was originally generated.
- To preprocess and repair the data:
  - The choice of software will depend on the exact approach taken to this task. The two approaches I have in mind for missing value estimation and outlier detection are statistical means, and neural networks.
  - Statistical means for missing value estimation include regression analysis, and function fitting then interpolation.
    - Statistical tools should have quick learning curves. The time taken to understand a tool sufficiently well to implement it in software will probably be less than the time taken for that implementation.
    - C++ and Python come to mind for automating these estimations, although it should even be possible to use Excel.
  - Statistical means for detecting outliers could build on those for missing value estimation by comparing the point location to its 'expected' location, were it missing.

- Same as above.
  - The neural network approach to missing value estimation would involve creating a neural network with an input node for each available landmark point, and an output node for each missing point. The network would be trained to estimate the missing points' locations from the known ones. I have begun implementing this in C++ with Microsoft Visual Studio, rather than with an existing software package, for the flexibility and control it will offer over network architecture and functionality.
    - I should expect an ongoing learning curve for any neural network implementation. As with my tentative neural network to estimate curvature angles, my choice of design is limited by knowledge and experience. My neural networks class has given me the basic understanding needed to investigate some common network types, but ongoing learning will give me access to more throughout my Master's.
  - Estimating the correct values in the case of erroneous outliers will likely be similar to that of estimating missing values. The network might also take the probability that the point is an outlier into consideration. A separate network will be needed if this probability is to be estimated by neural means.
    - Having completed a few neural network assignments in C++ with Visual Studio, I am unlikely to move away from them in the near future. Python is an attractive alternative, for my familiarity and its ease of use.
- Despite the other uses neural networks may have in this project, the original idea was to use them to estimate the angle of maximum curvature of the patient's spine from the landmark data.
  - With one input node for each landmark point, and a single output, this network will be trained to estimate the patient's spinal curvature.
    - Again, C++ and Python are my first choices to implement neural networks.

Several results I plan to collect from the curvature estimation experiment (for given preprocessing procedures) are: estimation error (averaged over the test set) with respect to the three error types in the input data; estimation error with respect to training set size for fixed input degradation; and estimation error with respect to curvature magnitude.

- The estimation error of the network is the difference between its output and the ground-truth obtained from CT, averaged over all input data sets in the test set. By retraining and testing the network with many data sets, containing various amounts of the 3 error types, I can investigate the relationship between network error and input error. Collecting this data corresponds to filling a 3D volume with various values. For how difficult that would be to depict, I will also perform trials varying 1 or 2 of the errors' amounts.
- The relationship between estimation error and training set size can be investigated by performing trials with various amounts of the data partitioned into the training and test sets, with constant (perhaps 0) input error.
- The procedure is similar for the relationship between estimation and curvature severity.
- All of this data is generated by running the neural network program (C++ for now) on the appropriate data set. The data can simply be written to a .csv file and visualized in Excel.
- There should be no learning curve for visualizing this kind of data in Excel. The learning curve of the other 3 points might be thought of as the time it takes to produce a working program. I expect that to be 2 or 3 weeks, although the program will hopefully continue improving after that.

Visualizing the results from my experiment should be easy with Excel. Interpreting those results and evaluating my method on their basis will be new to me. As I mention in the Benchmarking section of my Assignment #5, I intend to compare my network's estimates to those of humans with corresponding, 'clean' data, and to the clinically acceptable limits of error.

- Basic, familiar statistics will be calculated from the data. For example, expected values and variances of estimated error for the various trials can be calculated automatically with Excel.
- I know (or remember) very little of the statistical tests used to compare my results to those of others.
  - [Chen2011] does not use involved statistics to assess the quality of their curvature quantification method. They report means and standard deviations in angle estimation for their method vs. the gold-standard over mild, moderate, and severe cases.
    - I will not need to learn any new statistics to compare the mean curvature estimates of a neural network to CT-derived ground-truth. However this may require a different statistical procedure altogether; ground-truth lacks the notion of a mean or variance, it is just a value. As such, the same comparison might not be appropriate.
  - [Ungi2014] reports the correlation coefficient between their ultrasound quantification method, and standard X-ray measurements. They provide the difference between the results of the two methods, with uncertainty, in degrees. Box plots depict the relative distributions of the two methods curvature estimations.
    - As with [Chen2011], these are simple statistics I can compute in Excel, but I don't know if I can form compelling evidence by comparing my results to a 'ground-truth'.
- The statistics used in these papers notwithstanding, I will need to review statistics in general to acquire the vocabulary needed to search for relevant statistics and tests. With much of the programming underway, learning the statistics could have a longer learning curve than developing the first working experiment.

## References

- [Chen2011] W. Chen, E. H. M. Low, and L. H. Le, "Using Ultrasound Imaging to Identify Landmarks in Vertebra Models to Assess Spinal Deformity", 33<sup>rd</sup> Annual International Conference of the IEEE EMBS 2011.
- [Ungi2014] T. Ungi, F. King, M. Kempston, Z. Keri, A. Lasso, P. Mousavi, J. Rudan, D. P. Borschneck, and G. Fichtinger, "Spinal Curvature Measurement by Tracked Ultrasound Snapshots", *Ultrasound in Medicine and Biology* 2014; 40(2):447-454.