Accelerating Robotic Arm Calibration on GPGPUs
Chong Chen
Advisor: Tarek M Taha, PhD.

Research Objective: To Reduce the time for industry robot calibration by implementing the algorithm in GPGPUs.

Introduction
- Ideally, a robot arm should move to an exact location.
- Bias errors cause the final position to be different.
- The object of calibration is to eliminate these errors.
- The output of the hidden layer of RBFNN is determined by Euler distance between the input and the center. The center is determined by real data from measurement.
- Parallelize Euler distance matrix operations. Obtain thousands of threads: each data point is on one thread.

Methodology
Radial Basis Function Neural Networks (RBFNN) are good at predicting this kind of distribution.

Result
- Red lines: original errors. Blue lines: errors after calibration.
- Average error before calibration: 0.71. Average error after calibration: 0.079. Improvement in precision: 88%

Conclusion
- GPGPU based parallelization provides significant speedup.
- This enables more detailed calibration that was not computationally possible earlier.

Methodology Diagram:
- Radial basis function neural networks (RBFNN) are good at predicting this kind of distribution.

Methodology Diagram:
- The learning curve is generated by 7-fold cross validation. We can see that the error decreases with increase of spread values.
- We set the spread value to 3500.

Methodology Diagram:
- Runtimes: Before (Matlab toolbox based code): 1 week. After (GPGPU implementation): 10 min. Final speedup of over 300 X.