Artificial intelligence approaches to interpret data from piezoresistive smart fabrics

James Moffet, Lara Tomholt

November 02, 2017

Piezoresistive Smart Fabric

Domain:

Piezoresistive, pressure-sensing “smart” fabric

Problem in the domain:

Infer the true shape of the surface from a grid of noisy embedded sensors

Test case (piezoresistive smart fabric):

What does it do?

The fabric has a top and bottom layer, each of which has an 8x8 grid of embedded sensors. Senses its own shape and maps this shape in grasshopper.

What are the hardware and software components?

Two layers of piezoresistive fabric

8 conductive threads on the outside of the top layer, 8 threads on the outside of the bottom layer and 8 thread sandwiched in between.

In the two outside layers threads lay horizontal, on the inside layer threads lay vertical

A microcontroller (arduino) reads resistance values through the fabric at 128 points (64 through the top layer, 64 through the bottom layer) and streams them to a serial buffer

The serial data is read by a C# script and displayed in 3D graphical environment

(Rhino/Grasshopper)

How does it work?

64 sensors, each sensor takes two measurements at some point on the fabric, one through the top layer and one through the bottom layer (measuring current from thread y_n to thread x_a_n, and from y_n to x_b_n).
It deduces its shape by measuring the difference in resistance between the two measurements in each sensor. When the fabric is folded or creased at some point, the top and bottom layers respond differently, but not necessarily as we would hope (i.e. in an equal and opposite, linear manner).

Currently, the software averages the value for a given node's neighbors to get a topside average value and a bottom side average value. The top is subtracted from the bottom and the difference (including its sign) determines a "z" value for the displayed surface. The displayed surface is not an accurate representation of the actual pose of the fabric (i.e. it doesn't maintain length and width across the surface, etc...)

<u>Issues with interpreting the data that AI might be useful for:</u>

The current software does no real inference about the shape, instead representing a simple heightfield based on the difference at each point between the top layer sensor and the bottom. A simple heightfield is fast and easy, but does not accurately represent the shape (i.e. it does not maintain a constant area, etc...)

We would like to infer the most plausible shape that does not violate certain physical constraints (namely, the surface must remain developable).

1) What does the data mean? When a point on the fabric is part of a crease/fold, the compression of both the top and bottom layers increases but at slightly different rates.
2) Because of meaningful non-uniformity in the fabric (fairly large fibers coated with piezoresistive solution), the relationship between top and bottom readings is not uniform from node to node
3) The relationship between the resistance measurements and the angle of folding is also non-linear.
4) Additionally, elements in the system outside of the fabric (multiplexors, arduino, static on hands that manipulate the fabric) add noise to the readings.

For these reasons, we don't believe it would be effective to attempt to formulate a solver for this system. Instead, we intend to try several inference models, based on strategies we've learned in class.

AI approaches

To deal with the challenges of noisy, non-linear data, we will survey a number of approaches, evaluate and compare their success at inferring the true model shape

Data:

1. Embedded sensor data: Two continuous values (top and bottom) for each of 64 points on a square surface
2. Ground truth data from kinect or similar: xyz values observed for fabric as it is manipulated into different shapes, we can translate these into UV representation

Possible state representations:
1. For each node/sensor/point in the grid:
   - Fold angle(continuous)
   - Concave/convex, or “up”/”down” (binary)
   - Rotation (continuous), orientation of fold with respect to surface U and V
2. Two angles per node (continuous), one in u direction, one in v direction.
3. Node rep could be some smoothed average of neighboring values in above regimes

Potential AI approaches to test:

1. HMM and Bayes’ net
   These AI approaches might work based on the relationship between the nodes. For example, the probability of next neighbouring node having some angle, given the angle of the current node. It is likely that we will have complete connection between degree-1 neighbors.

2. Reinforcement Learning
   Given that we are manipulating the fabric and want accurate shapes in real-time, it may be useful to estimate future shape based on past shapes. We could use a q-learning algorithm to estimate the next node state (state rep includes sensor data and “truth” data), based on the previous node state, with the reward being the difference between the sensor data and the truth data.

3. CSP
   E.g. if one node is convex beyond a certain threshold, the neighbour node cannot be concave.
   We think this approach might be a bad choice, because testing the constraints might be expensive given that we’re dealing with continuous variables.

4. SVM
   We could potentially use SVM as a component of our final solution. (See references)

How to evaluate success? Compare between approaches:
- Computing time (ideal is ~100 hertz, which would mean we could produce an estimated shape as quickly as the raw sensor data is coming in (10,000 samples per second))
- Learning rate / Number of iterations
- Shape likeness (compare output with input of actual shape)

References

Charles, J., & Everingham, M. (2011, November). Learning shape models for monocular human pose estimation from the Microsoft Xbox Kinect. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* (pp. 1202-1208). IEEE.

Shape estimation based on grid of depth sensor data. Uses model iteration to infer 3D pose of a simplistic tree of human “bones.”

Gastaldo, P., Pinna, L., Seminara, L., Valle, M., & Zunino, R. (2014). Computational Intelligence Techniques for Tactile Sensing Systems. *Sensors (Basel, Switzerland)*, *14*(6), 10952–10976. http://doi.org.ezp-prod1.hul.harvard.edu/10.3390/s140610952

Harmon L.D. (1984) Tactile Sensing For Robots. In: Brady M., Gerhardt L.A., Davidson H.F. (eds) Robotics and Artificial Intelligence. NATO ASI Series (Series F: Computer and Systems Sciences), vol 11. Springer, Berlin, Heidelberg

Kim, W., Kak, A.(1991) 3-D object recognition using bipartite matching embedded in discrete relaxation. IEEE transactions on pattern analysis and machine intelligence, vol 13, no. 3, March 1991
https://engineering.purdue.edu/RVL/Publications/Kim913-D.pdf

Schmeder et. al. (2010) Support Vector Machine Learning for Gesture Signal Estimation with a Piezo-Resistive Fabric Touch Surface, Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME 2010), Sydney, Australia

Uses support vector machines to classify samples in a similar, but lower-dimensional problem (fabric touch pad, one pressure reading per point).

Sohn, et al. (2017) An extremely simple macroscale electronic skin realized by deep machine learning. Scientific Reports 7, Article number: 11061(2017)
doi:10.1038/s41598-017-11663-6
http://www.nature.com.ezp-prod1.hul.harvard.edu/articles/s41598-017-11663-6

I. Bandyopadhyaya, D. Babu, A. Kumar and J. Roychowdhury, "Tactile sensing based softness classification using machine learning," *2014 IEEE International Advance Computing Conference (IACC)*, Gurgaon, 2014, pp. 1231-1236.
http://ieeexplore.ieee.org.ezp-prod1.hul.harvard.edu/document/6779503/media