Can entropy-based image alignment metrics offer improved image aggregation of tissue density for mammographic risk assessment?

Report Name	De-Luca & Termini Fuzzy Entropy Implementation
Author (User Id)	Laura Collins (lac32)
Supervisor (User Id)	Dr. Neil MacParthalain (ncm)
Module	CS39440
Degree Scheme	GH7P (Artificial Intelligence & Robotics)
Date	March 14, 2016
Revision	0.1
Status	Draft

1 Fuzzy entropy description

De-Luca & Termini fuzzy entropy algorithm [2] is considered to be the first to build upon Shannon entropy. Their implementation takes into account a set of data, along with their various membership degrees.

$$H_A = -K \sum_{i=1}^{n} \{\mu_i \log(\mu_i) + (1 - \mu_i) \log(1 - \mu_i)\} \tag{1}$$

Al-sharhan et al's paper compiling several Fuzzy Entropy algorithms [1] contains a methodical,in-depth derivation of their algorithm, and has been instrumental in building my knowledge on the algorithm in question.

We will assume $-K$, the positive constant, is defined as $\frac{1}{n}$ as outlined in [2].

2 MATLAB implementation

After some research into current implementations of Fuzzy Entropy algorithms in MATLAB, it was concluded the best approach would be to implement De-Luca & Termini's algorithm from scratch. This entailed creating a membership class, which computes the grey-level membership of each pixel in the mean image (calculated from a set of input images).

This array of pixel memberships is fed into a 'De Luca' function where it is iteratively passed into latter part of equation 1 (after $\sum$). The output array is then summed and multiplied by $\frac{1}{n}$ as defined in Section 1. The final mean pixel entropy is calculated by taking the image entropy and dividing by the number of pixels in the image.

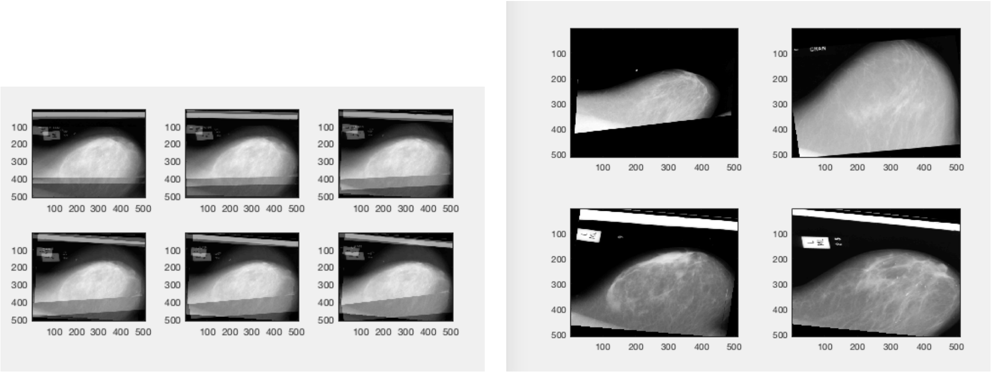
This is all relatively straight forward to implement in MATLAB, as it is designed to run mathematical equations.

3 Technical challenges

The main technical challenge for this implementation is ensuring maximum optimisation to keep running times to a minimum. Leveraging MATLAB's own functions for the membership saves a lot of time and lines of code, however it's been important to check what they call from within. One membership function was redrawing the trapeziums every time it was called, significantly slowing down the process - reducing the amount of times the initial function was called helped reduced the run-time by over 60seconds.

Another technical challenge faced whilst implementing the De Luca & Termini algorithm, isn't directly tied to the implementation of their specific equation, but more of my lack of experience in MATLAB, slowing down the programming rate. It has indeed been a steep learning curve, getting to grips with standard error messages, the debugger tool and knowing which 'Toolboxes' are needed to run specific MATLAB functions.

4 Results



(a) Mean image over each iteration

(b) Adjusted original input images

Figure 1: Output of 5 congealing iterations

Annotated Bibliography

[1] S. Al-Sharhan, F. Karray, W. Gueaieb, and O. Basir, *Fuzzy entropy: a brief survey*, 2001, vol. 3, p. 11351139.

Some other annotation

[2] A. De Luca and S. Termini, “A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory,” *Information and Control*, vol. 20, no. 4, p. 301312, May 1972.

Some annotation here