Anomaly Detection in Magnetic Motion Capture using a 2-Layer SOM network

Iain Miller
School of Computing
University of Paisley
United Kingdom
Email: iain.miller@paisley.ac.uk

Stephen McGlinchey
School of Computing
University of Paisley
United Kingdom
Email: stephen.mcglinchey@paisley.ac.uk

Benoit Chaperot
School of Computing
University of Paisley
United Kingdom
Email: benoit.chaperot@paisley.ac.uk

***Abstract*— Over recent years, the fall in cost, and increased availability of motion capture equipment has led to an increase in non-specialist companies being able to use motion capture data to guide animation sequences for computer games and other applications.[1] A bottleneck in the animation production process is in the clean-up of capture sessions to remove and/or correct anomalous (unusable) frames and noise. In this paper an investigation is carried out on the use of a system comprising of two layers of self-organising maps in identifying anomalous frames in a magnetic motion capture session.**

I. Introduction

Motion capture is the process of recording the motion of actors and/or objects, and this data is often used in computer games to animate characters and other game objects. The process normally involves tracking sensors or markers that have been placed in key positions on the actor's body, and detecting their locations in three-dimensional space. As the cost of equipment decreases, the realm of Motion Capture is no longer the preserve of specialist companies who take care of all aspects of data capture and post-processing. The task of supplying animation scenes from a motion capture system is now seen as a commodity, and so the focus has started to veer towards processing the output from a capture session as quickly and cheaply as possible. By improving post-processing, motion capture studios can get more useful (and commercial) application out of the capture equipment.

In previous work ([4]), a statistical method based on the variance of the distances between nodes, was used to detect anomalous points, whilst Kovar and Gleicher [3] use distance metrics to automatically detect similar motions in a session. Müller et al. [5] focus on using geometric relations to perform content-based retrieval and Gibson et al. [2] use principal component analysis and a multi-layered perceptron to extract motion information from a video or film. However, the latter two methods of feature extraction or recognition still require a considerable amount of input from an animator. Ideally, the animator interaction would be either non-exisentant or minimal and with this paper the aim is to investigate the usefulness and accuracy of a two-layered unsupervised neural network to the problem area of capture data clean-up. Section 2 gives an overview of the factors that produce noise and anomalies into the magnetic motion capture sessions, plus notes of the ideas behind the network design. Section 3 describes the form of the network and the parameters for learning, whilst section 4 provides a discussion and display of some of the results.

II. Background

The noise that can be produced during a capture are split into two types: sensor noise and positional anomalies. Sensor noise comes about by small variations in the magnetic fields used to induce a signal, the synchronization of the magnetic phases and interference from unwanted metal objects in or around the capture space. Positional anomalies come about when the sensors move too close to or too far from the field generators and where the sensors are unable to detect the field strength accurately and so produce anomalous results. The outcome being sensors reporting their positions that are inverted in the vertical axis or placed at a seemingly random position and breaking the skeleton of the captured article (which can be human, animal or an inanimate object).

The fact that the system should work autonomously of all external influences proscribes that, in a neural network method of anomaly detection, Self-Organising Maps, SOMs, provide one possible method for the system. The unsupervised nature of the SOMs allows the system to train each net to a session's particular structural make-up. The approach outlined here uses an initial layer of SOMs (one for each sensor in the capture session) to create inputs for a higher, second-layer SOM, thereby cutting the dimensionality of the final grid down by a third.

III. Methodology

Data is read in and stored in a separate matrix for each sensor in the session (called nodes from here on). Equation 1 shows one node's data in one frame in the session, whilst equation 2 (i is the node number and F is the total number of frames) shows the overall storage matrix for a node.

$$n_i(t) = \begin{bmatrix} n_{i1}(t) & n_{i2}(t) & n_{i3}(t) \end{bmatrix} \tag{1}$$

$$N_i = \begin{bmatrix} n_i(1) \\ n_i(2) \\ \vdots \\ n_i(F) \end{bmatrix} \tag{2}$$

For each node, a one-dimensional SOM is created and initialised (equation 3 with 4 showing the weight vector of a neuron, M is the number of neurons in the net). Each SOM is trained for 100 epochs using only the data for its associated node. One epoch uses every frame in the session, fed into the network in a random order.

$$S_i = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_M \end{bmatrix} \tag{3}$$

$$s_m = \begin{bmatrix} s_{m1} & s_{m2} & s_{m3} \end{bmatrix} \tag{4}$$

The Euclidean distance between the input vector and each neuron in a SOM is calculated, and the neuron with the minimum distance being declared the winner (see equation 5, i is the node number and k is the vector element).

$$c_i = \arg \min_{1 \leq j \leq M} (\sqrt{\sum_{k=1}^{3} (n_{i_k}(t) - s_{j_k})^2}) \tag{5}$$

The weights for the winning neuron are then updated using equation 6 with α being the adaptive learning rate (equation 7, $T = 100F$, where F is the total number of frames and tc is the training cycle), and h_1 is the gaussian neighbourhood function (equation 8 and figure 1, j and c_i are the neuron numbers of the neuron being updated and winning neuron respectively) that modifies the neurons closest to the winner more than those further away.

$$s_l' = s_l + \alpha h(n_i(t) - s_l) \tag{6}$$

$$\alpha = \alpha_0(1 - \frac{tc}{T}) \tag{7}$$

$$h_1 = e^{\frac{-(j-c_i)^2}{2}} \tag{8}$$

The outputs from each of these SOMs form the input vector for the second-layer SOM (equation 9). The second-layer SOM is a two-dimensional array of neurons (equation 10) with each neuron having a weight vector of that shown in equation 11. The winner is decided by the minimum Euclidean distance, as in the first-layer SOMs, with the training updates calculated using the same adaptive learning rate and gaussian neighbourhood function h_2 (equation 12, with R and C being the row and column address of the neuron being updated and c_R and c_C the row and column address of the winning neuron).

$$In = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_M \end{bmatrix} \tag{9}$$

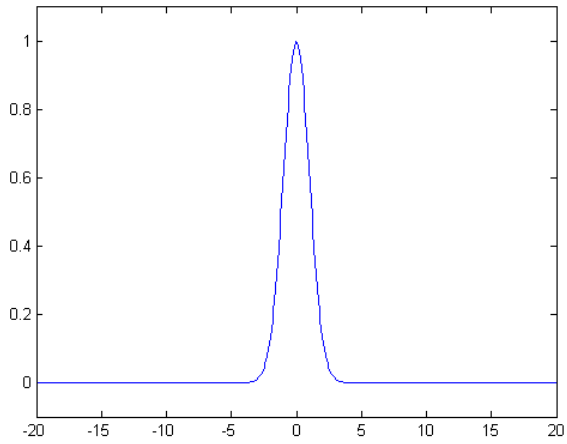


Fig. 1. Graph of the Neighbourhood function used to updated the SOM Weights

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1b} \\ v_{21} & v_{22} & \cdots & \\ \vdots & \vdots & \ddots & \vdots \\ v_{a1} & & \cdots & v_{ab} \end{bmatrix} \tag{10}$$

$$v = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_M \end{bmatrix} \tag{11}$$

$$h_2 = e^{\frac{-(R-c_R)^2-(C-c_C)^2}{2}} \tag{12}$$

In order to find the best combination of network sizes for the first and second layer SOMs, a series of empirical studies were carried out with the number of neurons in each of the first-layer SOMs varying between 11 and 51 in increments of 10 neurons. For the second-layer SOMs, the size of the neuron array was always kept square and used the following sizes: 11x11, 21x21, 31x31, 41x41, 51x51. The evaluation of what makes one network better than another is a subjective matter. Therefore, in order to make the decision more objective, three criteria were used to evaluate each resultant net:

1) Separation of the differing areas of ”clean” and ”anomalous” frames, the more defined a specific area is the better.
2) Minimisation of ”Overlapping Points”, where one neuron can win when a frame is either ”clean” or ”anomalous”.
3) Reduction in the proportion of ”Missing Neurons”,ones which do not win at any point for a session.

IV. Results and Discussion

Initially the networks were tested on one file, F1, of 407 frames, that consists of a series of frames with the figure inverted (blue ◯), followed by a series of anomalous frames

(red +), then a series of clean frames (green □), finishing with a series of anomalous frames (magenta ⋄). Due to this there are three changeover points, from this it can be surmised that there are strong possibilities of overlapping points being generated at each of the changeovers. Hence the par score for overlapping points is three.

No. of Neurons			Total used	Overlap	Level of
1^{st} Lyr	2^{nd} Lyr	Used	neurons (%)	Neurons	Group
11	11	60	49.6	4	Fair
11	21	99	22.4	2	Poor
11	31	110	11.4	3	Poor
11	41	126	7.5	3	Poor
11	51	120	4.6	2	Fair
21	11	55	45.5	3	Poor
21	21	139	31.5	2	Fair
21	31	86	8.9	2	Poor
21	41	131	7.8	3	Poor
21	51	145	5.6	2	Good
31	11	49	40.5	3	Good
31	21	96	21.8	1	Good
31	31	115	12.0	3	Poor
31	41	116	6.9	3	Good
31	51	123	4.7	2	Fair
41	11	48	40.5	3	Poor
41	21	94	21.3	3	Good
41	31	97	10.1	2	Good
41	41	120	7.1	3	Fair
41	51	135	5.2	2	Good
51	11	51	42.1	3	Poor
51	21	122	27.7	2	Good
51	31	111	11.6	2	Fair
51	41	123	7.3	3	Fair
51	51	139	5.3	2	Good

TABLE I

TABLE OF RESULTS FOR THE EMPIRICAL STUDIES OF THE 2-LAYERED SOM NETWORK

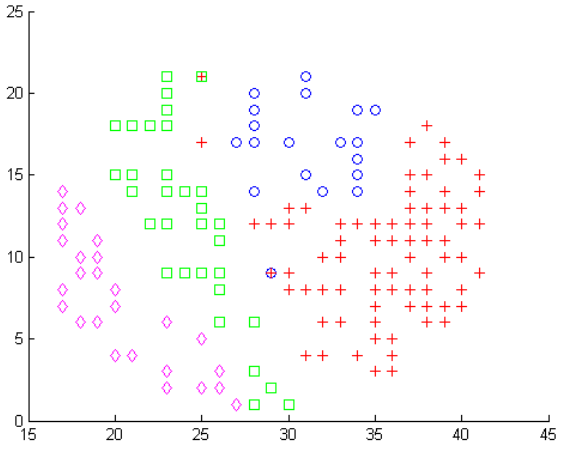


Fig. 2. Plot of the Winning Neurons for a 2-Layer SOM with 21 Neurons in each First-Layer SOM and 51x51 in the Second-Layer SOM

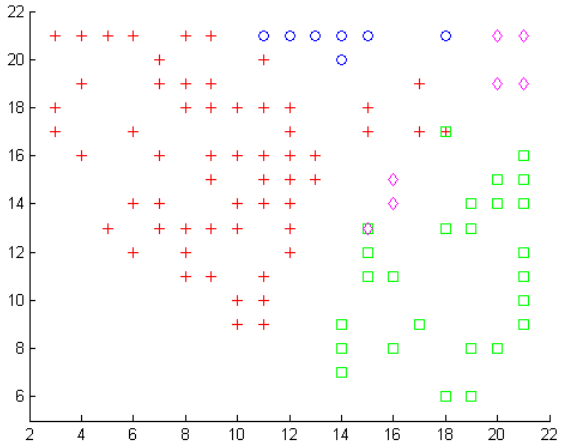


Fig. 3. Plot of the Winning Neurons for a 2-Layer SOM with 31 Neurons in each First-Layer SOM and 21x21 in the Second-Layer SOM

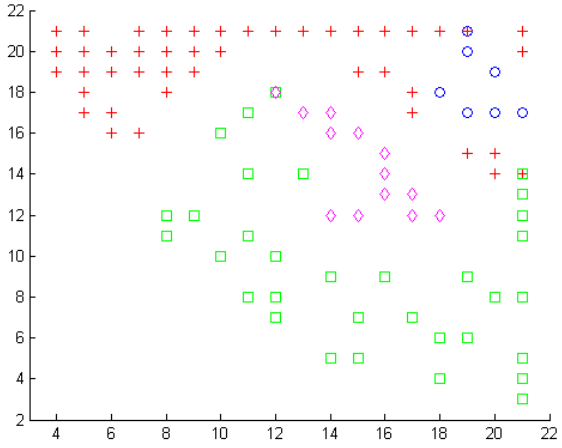


Fig. 4. Plot of the Winning Neurons for a 2-Layer SOM with 41 Neurons in each First-Layer SOM and 21x21 in the Second-Layer SOM

Some of the results from the empirical tests are shown above. Figures 2, 3 and 4 are examples of outcomes considered good and figures 5, 6 and 7 are examples of bad outcomes. In terms of timing issues a look at figure 8 shows that an increase in the size of the first layer SOMs does not have a significant effect on the training time of a network. However the increase in the time needed to train larger second layer SOMs increases in an exponential way.

From these results it was concluded that a second-layer SOM of size 21-by-21 neurons provided results with appropriate spread of the separate file groups. There is little difference between the outcomes whether you had 31 or 41 neurons in each first-layer SOM, but they produced the best grouping. Therefore, these networks were re-run three times each to see whether they produce consistency in their outcomes. Figure 9 show the results for the 31/21 network and figure 10 41/21

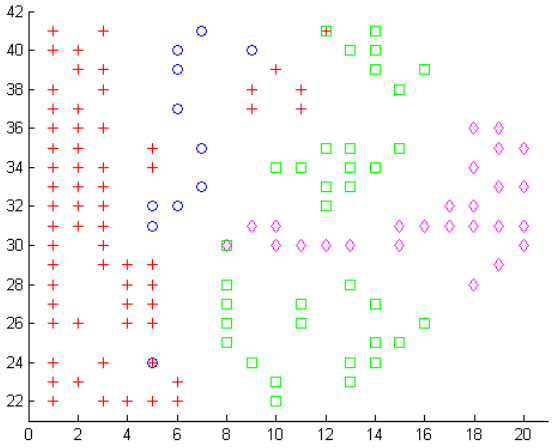


Fig. 5. Plot of the Winning Neurons for a 2-Layer SOM with 21 Neurons in each First-Layer SOM and 41x41 in the Second-Layer SOM

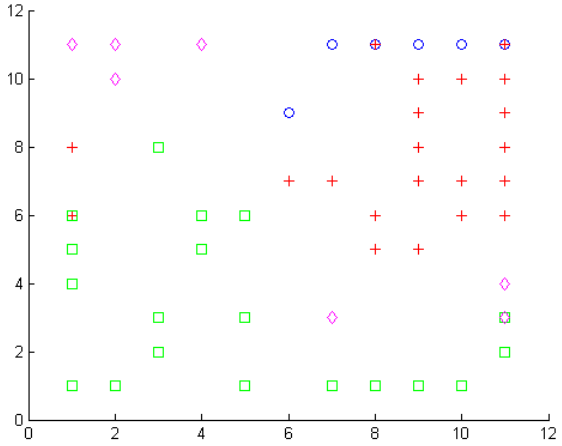


Fig. 6. Plot of the Winning Neurons for a 2-Layer SOM with 51 Neurons in each First-Layer SOM and 11x11 in the Second-Layer SOM

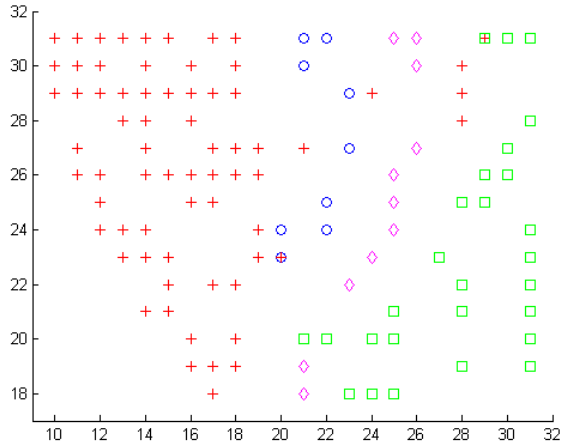


Fig. 7. Plot of the Winning Neurons for a 2-Layer SOM with 51 Neurons in each First-Layer SOM and 31x31 in the Second-Layer SOM

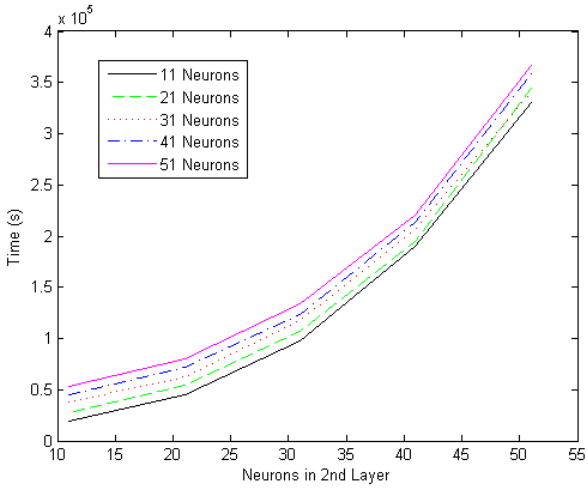
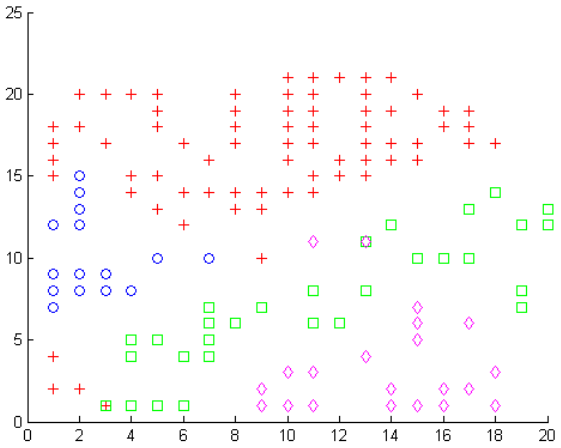


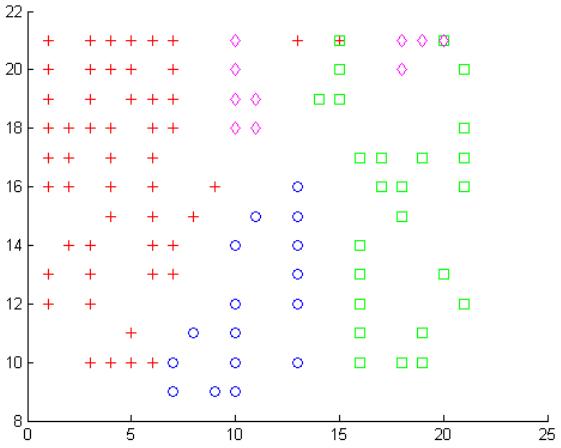
Fig. 8. Plot of the Change in Time Taken by Increasing the Size of the First Layer SOMs (Legend indicates the number of Neurons in each First Layer SOM)

network.

From these it can be seen that although both networks can reproduce their results they do not do so with complete consistency. However, those produced by the 41/21x21 network have a greater degree of consistency. To these ends this network was tested with 3 other, much larger files (all 2823 frames). Two were fairly simple files, T2 and T3, with a series of clean frames (blue ◯) followed by a series of anomalous frames (red +). T2 contained many more clean frames than anomalous, whilst T2 contained more anomalous than clean. The third file, T1, contains 60% clean frames, split between two series (blue ◯ and green □) , interspersed with four series of anomalous frames (red + and ◁, magenta ◇ and ▷) and four series of unknown frames (black ◁, ▷, ◇ and +). The unknown frames are those where it is very difficult to tell whether or not the structure in the frame has been kept or whether it is slightly anomalous. The results for these are shown in figures 11.

As can be seen in all three files there is a good degree of grouping for the different elements. In file T1 there are three overlapping nodes, which can be considered as a par score (with there being two periods of clean series), but there is a void of unused neurons in the middle of the network. Another issue with the grouping in this file is that two series of clean neurons are each spread over two areas. However, seeing as these areas are distinctly separate to the anomalous or unknown frames, it lends evidence to support the supposition that this network is capable of separately grouping clean and anomalous data.





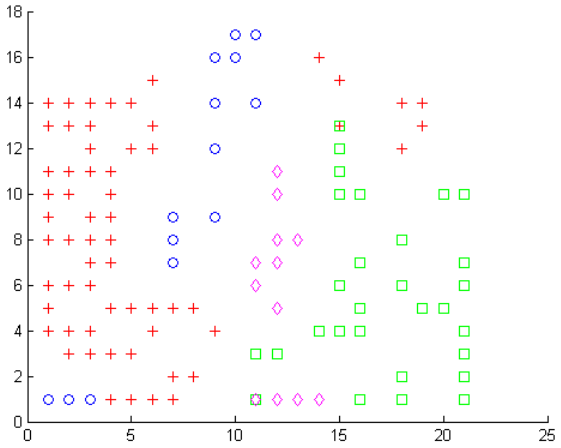
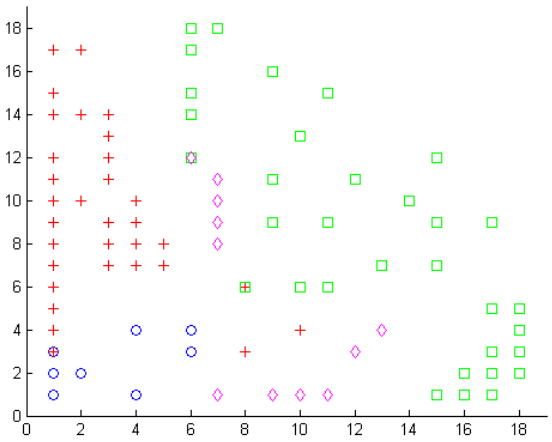
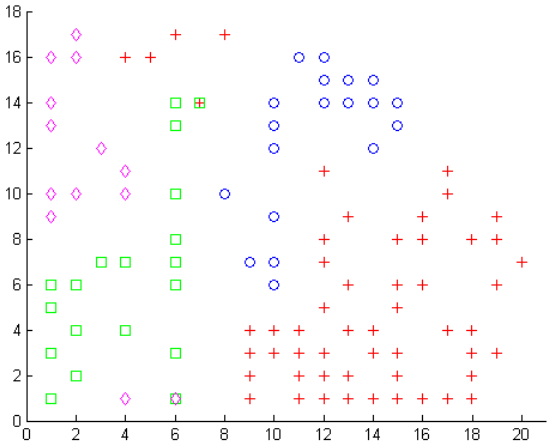


Fig. 9. Plots of the Winning Neurons for the Re-Runs of a 2-Layer SOM with 31 Neurons in each First-Layer SOM and 21x21 in the Second-Layer SOM





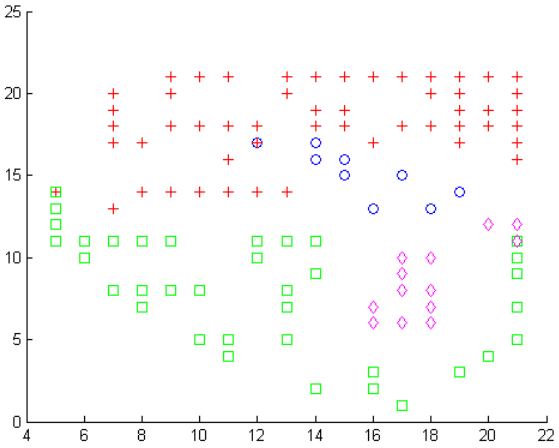


Fig. 10. Plots of the Winning Neurons for the Re-Runs of a 2-Layer SOM with 41 Neurons in each First-Layer SOM and 21x21 in the Second-Layer SOM

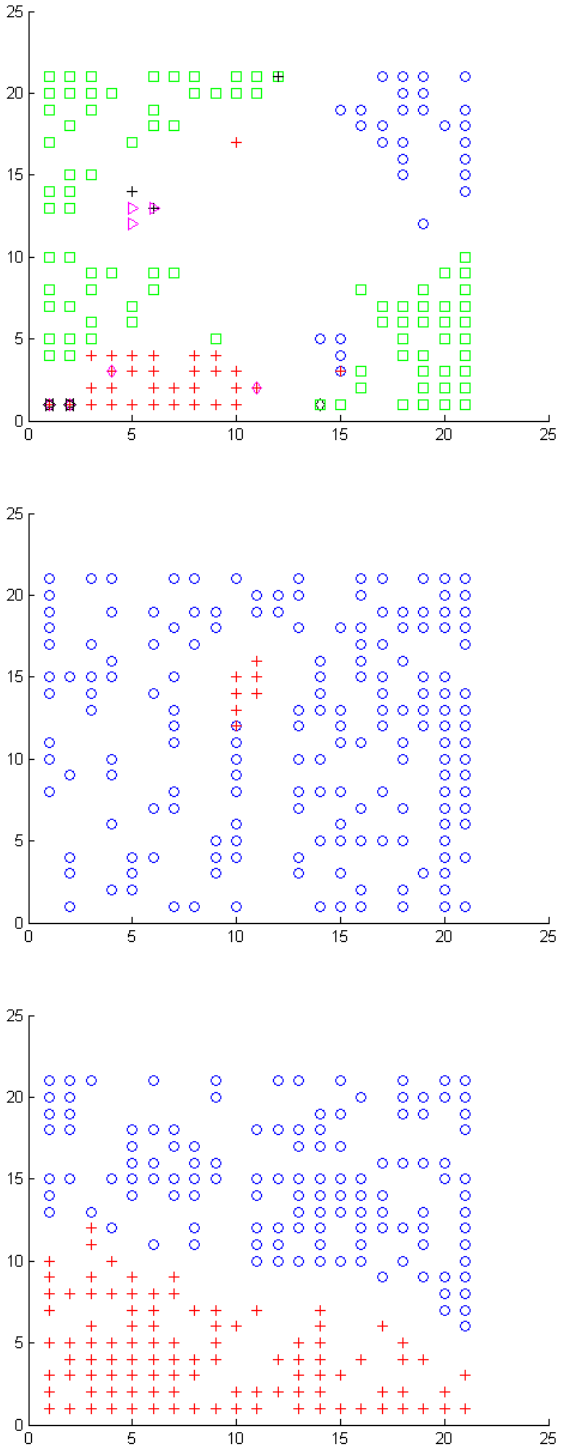


Fig. 11. Plots of the Winning Neurons of a 2-Layer SOM with 31 Neurons in each First-Layer SOM and 21x21 in the Second-Layer SOM for Files T1, T2 and T3 respectively top to bottom

A. Automating Classification

As has been shown, a network with 41 neurons in each first layer SOM and a 21 by 21 SOM in the second layer can produce a network capable of grouping and thereby classifying magnetic motion capture data into clean, inverted and anomalous. The key to making a system like this of viable commercial use, is to then limit the amount of animator interaction required for the system to identify which group corresponds to which classification. A look at a graph of the Euclidean distances between the winning neuron in the second layer and the input vector (see figure 12), suggests that there could be a link between an increase in the Euclidean distance and change in classification of data in a series of frames. The changeover points in F1 come in frames 51, 218 and 349, and as can be seen, there are spikes in the Euclidean distance around those points. There is however another spike/group of spikes around frame 290 that would need to be explained or compensated for in an automated scene. However, from looking at the larger files there is a doubt to this being a universal solution. One reason for this could be that the larger epoch size introduces a degree of over-training into the network.

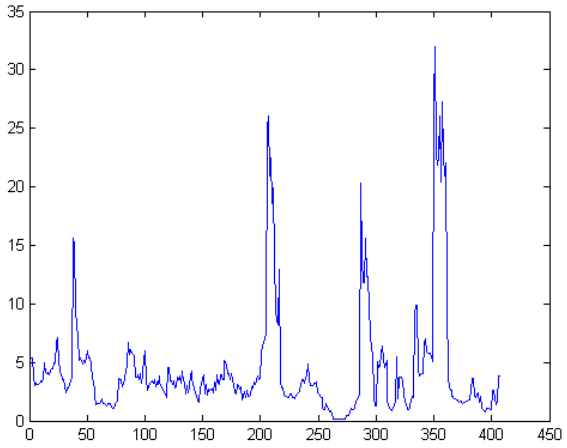


Fig. 12. Plot of the Euclidean Distance between the Winning Neuron and the Input Vector for the 41/21 Network with F1

V. Conclusion

Any system that seeks to automate the clean-up process of magnetic motion capture data is required to both provide a means of classifying into groups (A,B,C and D, etc.) and then identify the meaning of a group (i.e. that group A is clean data, B is anomalous data, C is inverted, etc.). In this paper we have shown a mechanism that has the ability to complete the first part of these requirements, and that there may be a means to developing the second. Though several network combinations produce good results for the separate of frames into groups the one that gave good results for both the small and large files was one with a 41-neuron 1D network for each sensor used in the session, with the results feeding into the inputs of a 21-by-21 2D network in the second-layer. There are problems with the process and the time taken for training a network are

still an issue. However, some of these could be alleviated by the generation of a generic test file for a specific sensor set-up, which contained series of clean, anomalous and inverted frames for a given capture space. A network could then be trained for that file and the different groups identified by a human operator, this could then be used to identify frames in other capture sessions using the same capture sensor set-up and space.

So far no pre-processing has been applied to the data before it is fed into the network, so that the effects of the capture space can be taken into account. However, it may be that the use of pre-processing techniques (such as centring or sphering), improve the grouping of the outputs and/or make the identification of what groups are easier. Other experiments could focus on the size of an epoch and whether using a random sample of all the frames rather than all of the frames in a session can produce quicker training, without compromising the usefulness of the technique. Alternatively, the use of some form of stopping criteria could be employed to save unnecessary training cycles and thereby improve the overall timing of the system.

ACKNOWLEDGMENT

The authors would like to thank Artem Digital for the provision of the test data, www.artem-digital.co.uk.

REFERENCES

[1] Margaret S. Geroch, *Motion Capture for the Rest of us*, Journal of Computing Sciences in Colleges,Vol. 19 No. 3, 2004. pp157-164.

[2] David P. Gibson, Neill W. Campbell, Colin J. Dalton and Barry T. Thomas, *Extraction of Motion Data from Image Sequences to Assist Animators*, Proceedings of the British Machine Vision Conference 2000.

[3] Lucas Kovar, Michael Gleicher, *Automated Extraction and Parameterization of Motions in Large Data Sets*, ACM Transactions on Graphics, Vol. 23, Issue 3, p.559-568. August 2004.

[4] Iain Miller, Stephen McGlinchey *Automating the Clean-up Process of Magnetic Motion Capture Systems* Proceedings of the Game Design and Technology Workshop, November 2005.

[5] Meinard Müller, Tido Röder, Michael Clausen *Efficient Content-Based Retrieval of Motion Capture Data* ACM Transactions on Graphics, Vol. 24, Issue 3, p677-685. July 2005.