

# Visualisation of synchronous firing in multi-dimensional spike trains

L. Stuart<sup>a,\*</sup>, M. Walter<sup>a</sup>, R. Borisyuk<sup>a,b</sup>

<sup>a</sup> Centre for Neural and Adaptive Systems, School of Computing, University of Plymouth, Plymouth, Devon PL4 8AA, UK

<sup>b</sup> Institute of Mathematical Problems in Biology, Russian Academy of Sciences, Pushchino, Moscow Region 142 290, Russia

Accepted 22 August 2002

---

## Abstract

The gravity transform algorithm is used to study the dependencies in firing of multi-dimensional spike trains. The pros and cons of this algorithm are discussed and the necessity for improved representation of output data is demonstrated. Parallel coordinates are introduced to visualise the results of the gravity transform and principal component analysis (PCA) is used to reduce the quantity of data represented whilst minimising loss of information.

© 2002 Elsevier Science Ireland Ltd. All rights reserved.

*Keywords:* Multi-dimensional spike trains; Synchronous activity; Information visualisation; Parallel coordinates; Gravity transform; Principal component analysis

---

## 1. Introduction

Solution of many problems in the field of Neuroscience is associated with the theoretical comprehension of a large body of experimental data. More specifically, investigation of information processing in the nervous system is associated with the analysis of vast quantities of simultaneously recorded multi-dimensional spike train data. Much of this analysis is based on the principle of synchronisation of neural activity (Borisyuk and Borisyuk, 1997; Fries et al., 2001).

The experimental evidence that is currently available requires further, in-depth analysis in

order to extract inherent information. Analysis of multi-dimensional spike trains using traditional tools such as cross-correlograms is increasingly complex due to the vast number of pairs involved. Hence, new methods of analysing this data are required. Among other methods, the ‘Gravity Transform’ algorithm, developed by Gerstein and Aertsen (1985), Gerstein et al. (1985), has a lot of potential.

In this paper, an implementation of the original Gravity Transform algorithm is presented alongside simulation results. Three trials, based on different neuronal assemblies, are presented. This includes a trial where  $n$  is large, relative to previously published results for the method. Subsequently, conclusions are drawn regarding the advantages and disadvantages of the method.

---

\* Corresponding author. Tel.: +44-1752-600600

E-mail address: [l.stuart@plymouth.ac.uk](mailto:l.stuart@plymouth.ac.uk) (L. Stuart).

As the size of  $n$  increased in these trials, the output of data became increasingly complex. In addition to traditional representations of gravity transform data, Parallel Coordinates, were also used to support interpretation of the data. These are an innovative means of representing  $n$ -dimensional data sets.

In addition to investigating new methods of visualising these large quantities of data, reduction of data sets was also investigated. Thus, this paper presents the results of using principal component analysis (PCA) to create more manageable data sets whilst maintaining the most significant characteristics of the data. An additional two trials, based on different neuronal assemblies, are discussed to highlight the effectiveness of using PCA. In these trials, PCA is used to reduce data sets, created by the gravity transform, which are subsequently plotted and analysed.

## 2. The gravity transform

The gravity transform is a method of analysis of spike train dependencies and synchronisation based on the principle of gravitational interaction of particles. Each neuron is represented by a ‘virtual particle’; the movement of those particles is described in  $n$ -dimensional space, where  $n$  is the number of spike trains under investigation. All particles start equidistant.

The gravitational force resulting from the charge of a particle is calculated on the basis of the spike train of its corresponding neuron. Each spike contributes into the charge and this contribution decays exponentially over time. If two or more neurons spike coincidentally, their corresponding particles will exert an attractive force that causes the particles to move closer together.

Let us suppose that several neurons have an above average synchrony. Over time this would result in a strong attractive force between their corresponding particles. In turn, this would cause the particles to aggregate into specific patterns in  $n$ -dimensional space. Gerstein specifies that over time all particles will eventually aggregate together into a single cluster due to these attractive forces. Since significant synchrony can indicate synaptic

coupling, (Baker and Gerstein, 2000), the aggregation of the particles can show the assemblies representing the neuronal interactions. In fact, the aggregation reflects neuronal activity.

Note that all spike trains used for experimentation were generated using an enhanced Integrate and Fire generator defined by Borisyuk and Borisyuk (1997).

### 2.1. Description of the gravity transform algorithm

Let us consider  $n$  simultaneously recorded spike trains with epoch  $[0, T]$ . Suppose that the  $i$ th spike train is represented by spikes at times  $T_1, T_2, \dots, T_k$ . The ‘charge’ of the  $i$ th particle corresponding to the spike train is described by the following procedure. Each spike contributes a quantity of charge  $a$ , which decays exponentially over time with constant  $\tau$ . Thus the charge on particle  $i$  at time  $t$  depends on the sum of all preceding spikes and is given by:

$$q_i(t) = \sum_{m=1}^k Q(t - T_m) - \lambda_i$$

$$Q(x) = \begin{cases} a & \text{if } x = 0 \\ ae^{-x/\tau} & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

where  $\lambda_i = a\tau k/T$  is the average firing rate of neuron  $i$ . The charge function  $q_i(t)$  of the particle is pre-calculated with a time step  $\Delta$  ( $\Delta = 1$  ms), thus  $q_i(t_j)$ ,  $t_j = j\Delta$ ,  $j = 0, 1, \dots$ , and intermediate charge values are linearly interpolated from the stored values. The dynamics of interactive particles in  $n$ -dimensional space is governed by the equations:

$$\frac{dx_i^k(t)}{dt} = bq_i(t) \sum_{j=1}^n q_j(t) \frac{x_j^k(t) - x_i^k(t)}{R_{ij}(t)}$$

where  $(x_i^1(t), \dots, x_i^n(t))$  is the position of particle  $i$  at time  $t$ ;  $k = 1, 2, \dots, n$ ;  $i = 1, 2, \dots, n$ ;  $R_{ij}(t)$  is the Euclidean distance between particles  $i$  and  $j$ ;  $a$ ,  $b$  and  $\tau$  are constants. All particles in the system are initialised at time  $t = 0$  to be equidistance:

$$\begin{aligned} x_i^m(0) &= 100 & \text{if } i = m & \text{ and} \\ x_i^m(0) &= 0 & \text{otherwise.} \end{aligned} \quad (1)$$

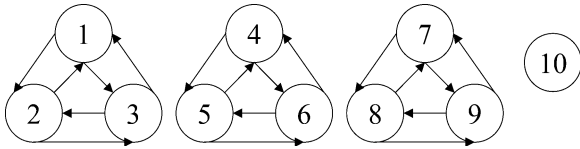


Fig. 1. Specification of the connections between the ten neurons used as input to the spike train generator.

Integration within this implementation of the algorithm is achieved by an adaptive step-size Runge–Kutta fourth order algorithm (Press et al., 1992). The use of this ODE solver permits the algorithm to progress through the integration with an optimal time step whilst maintaining a low cumulative error compared with an algorithm using a fixed time step.

2.2. Distance graphs

In the original implementation of the gravity transform method, output is represented by a ‘distance graph’. This graph depicts the Euclidean

distance between each pair of particles in the system over time.

Using the neuron circuit shown in Fig. 1, depicting three groups of excitatory neurons and a solitary neuron, spike train data was generated. This trial lasted 5 s and neurons had a firing rate of approximately 0.2670 spike per s. A raster plot of a portion of this data is shown in Fig. 2.

The cross-correlograms of four neuron pairs are shown in Fig. 3.

The correlation of synchronous activity within the first group is shown in Fig. 3(a–c), which clearly shows pair-wise interdependencies (the high peak on each graph depicts synchronous firing). Additionally, the cross-correlogram (different scale) for neurons 1 and 4 is shown in Fig. 3(d). This shows that no dependency exists between these neurons.

The same data was input to the gravity transform and the distance graph shown in Fig. 4 resulted. This graph has been annotated, using set theory notation, in order to identify individual

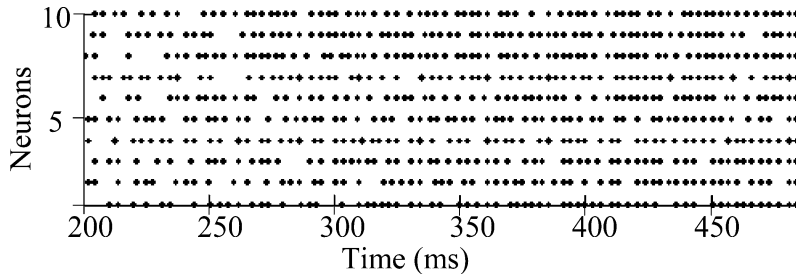


Fig. 2. A raster plot depicting a portion of the ten spike trains generated for the neuron assembly shown in Fig. 1.

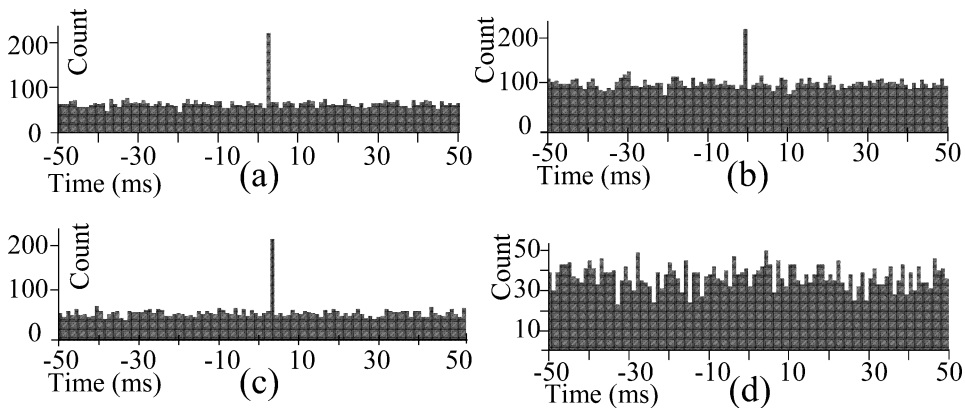


Fig. 3. This figure depicts cross-correlograms of (a) neurons 1 and 2; (b) neurons 2 and 3; (c) neurons 1 and 3; and (d) neurons 1 and 4.

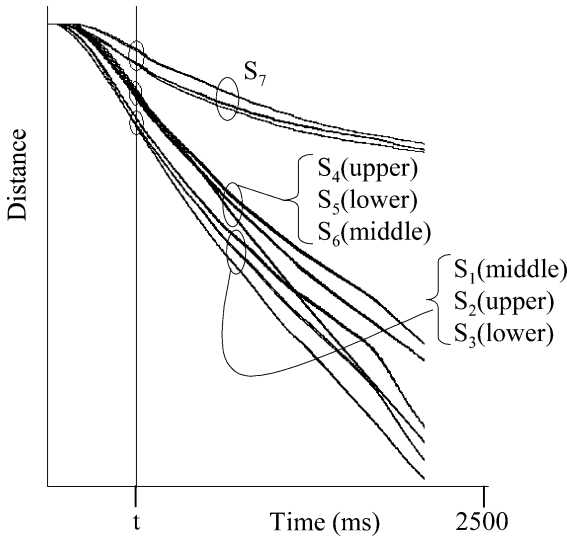


Fig. 4. Distance graph of the results of the gravity transform where  $a = 0.3$ ,  $\tau = 0.4$  and  $b = 1$ .

distance pair plots. Let  $d_{ij}$  represent the Euclidean distance between the particles  $i$  and  $j$ .

At time  $t$  three distinct groups are apparent. The top group represents the distances between neurons 1–9 and neuron 10. The lower, and middle, groups represent all intra-group, and inter-group, distances, respectively. Note that overlapping occurs.

The final intra-group distances,  $s_1 = \{d_{12}, d_{13}, d_{23}\}$ ,  $s_2 = \{d_{45}, d_{46}, d_{56}\}$  and  $s_3 = \{d_{78}, d_{79}, d_{89}\}$  show the three groups of neurons aggregating. The final inter-group distances,  $s_4 = \{d_{ij}; i = 1, \dots, 3, j = 4, \dots, 6\}$ ,  $s_5 = \{d_{ij}; i = 1, \dots, 3, j = 7, \dots, 9\}$  and  $s_6 = \{d_{ij}; i = 4, \dots, 6, j = 7, \dots, 9\}$ , show the distances between the three aggregating groups. The final distances between neurons 1–9 and neuron 10,  $s_7 = \{d_{ij}; i = 10, j = 1, \dots, 9\}$ , show that the solitary neuron has no tendency to group with any of the other neurons.

This interpretation of a distance graph highlights the usefulness of the technique in identifying groups of neurons within assemblies. However, it should be noted that it is generally difficult to produce a distance graph of similar quality to the one shown in Fig. 4. The quality of the distance graph is highly influenced by the number of neurons in the assembly and the choice of para-

meter values: (i) the increment of charge per spike,  $a$ ; (ii) the charge decay rate,  $\tau$ , (iii) the overall aggregation of the system,  $b$ . For Fig. 4,  $a = 0.3$ ,  $\tau = 0.4$  and  $b = 1$ .

The gravity transform is sensitive to the ‘appropriate’ specification of the constants that represent the decay, increment and aggregation. Inappropriate choice of these values may result in all particles becoming relatively coincidental within  $n$ -dimensional space, before any useful information about neuronal groups is discovered. Alternatively, it could result in the particles aggregating at such a low rate that they appear unrelated. Hence, it is sometimes necessary for the investigator to ‘fine-tune’ the specification of these constants in order to gain useful distance graphs.

To demonstrate this problem, first of all, consider the neuron assembly in Fig. 5. This assembly was used to produce spike train data for a trial lasting 20 s. This data set was input to the gravity transform with parameters  $a = 0.5$ ,  $\tau = 0.5$  and  $b = 5$ . The resultant distance graph is shown on the left in Fig. 6 where  $s_1 = \{d_{ij}; i = 1, \dots, 10, j = 2, \dots, 10, i < j\}$ .

Note that it is not possible to distinguish the two groups in the assembly, even though the scale is relatively small.

Using the same input data set, the gravity transform was executed with another set of parameters:  $a = 0.5$ ,  $\tau = 0.5$  and  $b = 0.1$ . The resultant distance graph is shown on the right in Fig. 6 where  $s_2 = \{d_{ij}; i = 6, \dots, 10, j = 7, \dots, 10, i < j\}$ ,  $s_3 = \{d_{ij}; i = 1, \dots, 5, j = 6, \dots, 10, i < j\}$  and  $s_4 = \{d_{ij}; i = 1, \dots, 5, j = 2, \dots, 5, i < j\}$ . From this graph, the structure of the assembly can be derived. Hence, the groups of neurons 1–5 and

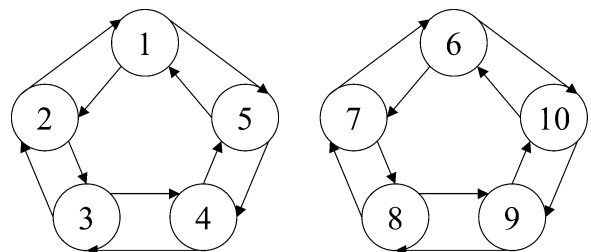


Fig. 5. Specification of the connections between the ten neurons used as input to the spike train generator.

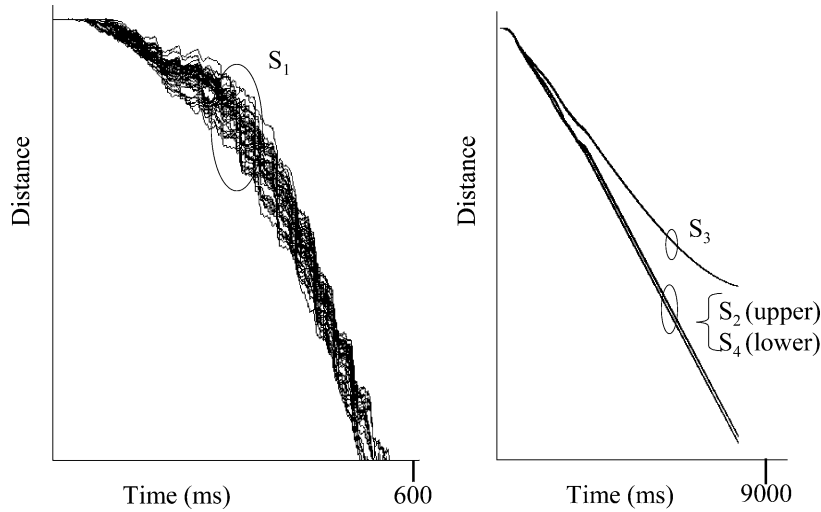


Fig. 6. Distance graphs of the results of the gravity transform where  $n = 10$ ,  $a = 0.5$ ,  $\tau = 0.5$ ,  $b = 5$  (left) and  $b = 0.1$  (right). Note the differing scales.

6–10 are notable as is the separation between the groups.

Limited simulation results have shown that appropriate ranges for the parameter values are  $0.1 < a < 0.5$ ;  $0.3 < \tau < 0.5$  and  $0.1 < b < 2$ . A method of automatic specification of these parameters, based on the distribution of inter spike intervals, is under investigation.

### 2.3. Increasing the number of particles

In the relevant publications, the maximum number of particles used within the gravity transform is relatively small,  $n \leq 10$ . This section reports on trials using the gravity transform based on relatively large values of  $n$ . In these trials, where  $n$  is relatively large, the use of distance graphs to display the output proved to be less useful, due to the vast number of individual pair plots that exist.

Consider an assembly of 50 neurons. The connected portion of this assembly, involving 20 neurons, is shown in Fig. 7. The remaining 30 neurons of this assembly are not shown as they are unconnected.

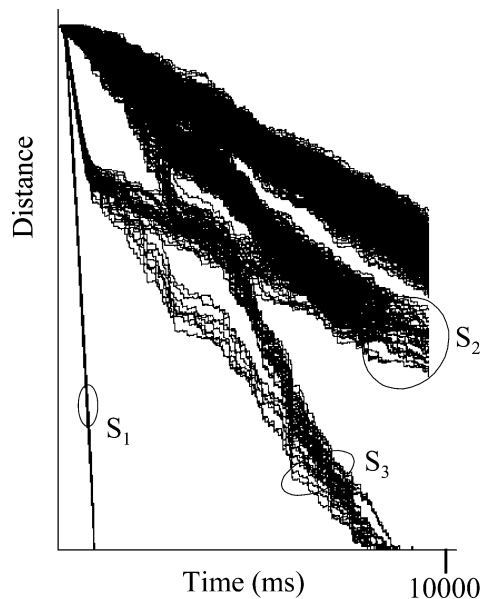


Fig. 7. Specification of the circuit used by the 20 connected neurons. The remaining 30 neurons are not shown as they are unconnected. The complete assembly was used as input to the spike train generator.

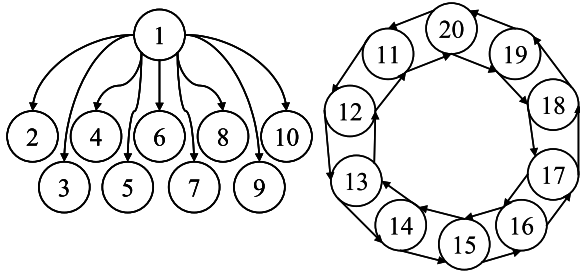


Fig. 8. Distance graph of the results of the gravity transform where  $n = 50$ ,  $a = 0.4$ ,  $\tau = 0.3$  and  $b = 1$ .

This assembly was used to produce spike train data for a trial lasting 20 s. This data was input to the gravity transform, with  $a = 0.4$ ,  $\tau = 0.3$  and  $b = 1$ .

The resultant distance graph is shown in Fig. 8 where  $s_1 = \{d_{ij}; i = 11, \dots, 20, j = 12, \dots, 20, i < j\}$ ,  $s_2 = \{d_{ij}; i = 1, j = 2, \dots, 10\}$  and  $s_3 = \{d_{ij}; i = 2, \dots, 10, j = 3, \dots, 10, i < j\}$ . The different clusters are notable, however, detail is obscured by the quantity of data displayed.

Since, this problem increases with the size of  $n$ , the interpretation and analysis of the large quantities of output from the gravity transform requires more sophisticated methods of representation. In the remainder of this paper, two solutions to the problem of representing this data are proposed: (a) utilisation of parallel coordinates to visualise the information, and; (b) reduction of the quantity of data dimensions displayed whilst minimising loss of information.

### 3. Data presentation

With the use of large numbers of particles, the gravity transform results in vast multi-dimensional data sets which represent the position of each particle in every dimension at every time point. The visual representation of the result should accurately convey the position of the particles particularly their direction. This poses a significant analysis problem which may be handled by introducing an alternative method of data representation.

The use of parallel coordinates, originally pioneered in the 1980s, is a technique used to

represent diverse sets of multi-dimensional data. In 1990, Inselberg and Dimsdale (1990), Wegman (1990) renewed the use of parallel coordinates for the analysis of large quantities of multi-dimensional data and introduced new representation features that led to a significant increase in their use.

Inselberg's representation of parallel coordinates denotes data points as vertical axis coordinate values distributed along a horizontal axis. In this scheme, a specific point in  $n$ -dimensional Euclidean space is represented by  $n$  vertical axes values distributed along the horizontal axis. For example, suppose we have the points  $a$  and  $b$  in three-dimensional  $(p, q, r)$  space:  $a(2, 0, 0)$  and  $b(1, 2, 2)$ . Fig. 9 depicts these as parallel coordinates using three vertical axes  $p, q$  and  $r$ .

Parallel coordinates can be used to identify correlations between variables and to convey aggregation information. In this paper we, propose the idea of using parallel coordinates as a simple means of representing  $n$ -dimensional coordinates in a two-dimensional plane. This would represent a snapshot of the gravity transform. It is proposed that these coordinates are animated over time to represent the changing position of particle within the gravitational system.

#### 3.1. Parallel coordinates: example one

The assembly of ten neurons, depicted in Fig. 5, is used to generate spike train data for 20 s. This data is input to the gravity transform and its output is visualised using parallel coordinates. In total there are 20 000 intervals (time step = 1 ms). Hence, the animation is made up of 20 000 snapshots. Additionally, since  $n = 10$ , each snapshot of the parallel coordinates depicts the position of all ten particles in ten-dimensional space. The legend for all this data is shown in Fig. 10 and Fig. 11 depicts snapshot 2000 of the animation. Recall that all particles in the system are initialised at time  $t = 0$  to be equidistance (Eq. (1)).

Hence, the initial range of each vertical axis will be from 0 to 100. In Fig. 11 some change, from this initial position, is noted in the plot. However, change is more obvious in snapshot 6000, shown in Fig. 12, where a separation of the data in two

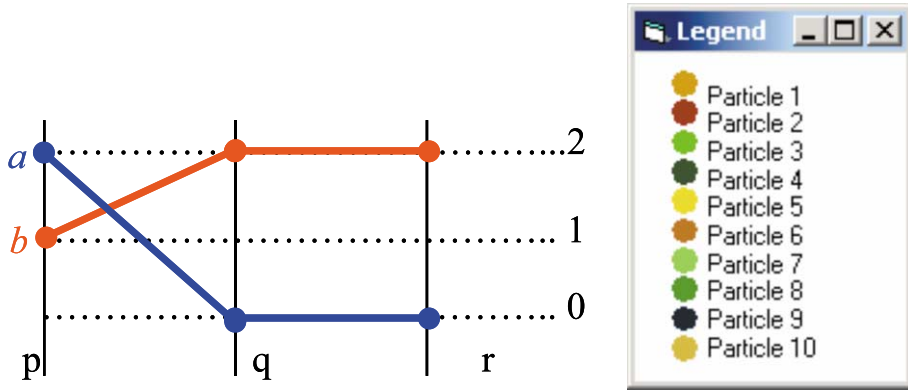


Fig. 9

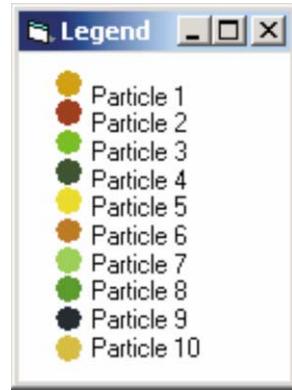


Fig. 10

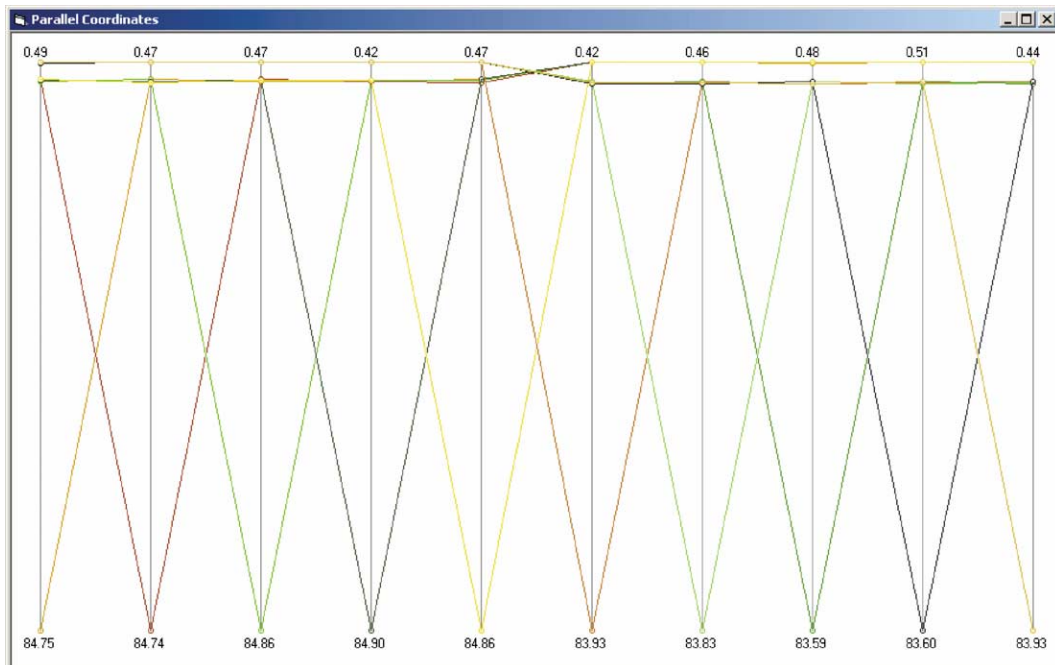


Fig. 11

Fig. 9. Illustration of a three-dimensional parallel coordinate plot. Representation of points  $a(2, 0, 0)$  and  $b(1, 2, 2)$ , using parallel coordinates.

Fig. 10. Legend of the animated parallel coordinates.

Fig. 11. Representation of particle positions using parallel coordinates. Snapshot 2000 showing all ten parallel coordinates in all ten dimensions.

groups is noted. Note that snapshot 10 000 of this animation, given in Fig. 13, shows two very distinct groups. Closer examination reveals that

all five particles, in the first group, are at the same position denoted in the diagram by the overlap of their parallel coordinates. All particles from the

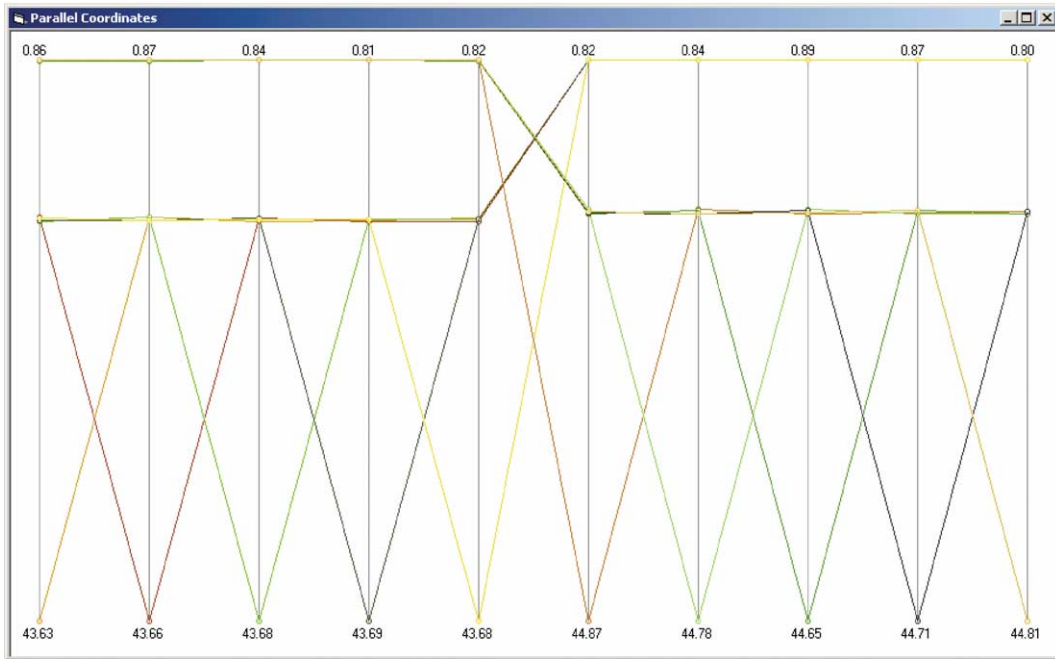


Fig. 12

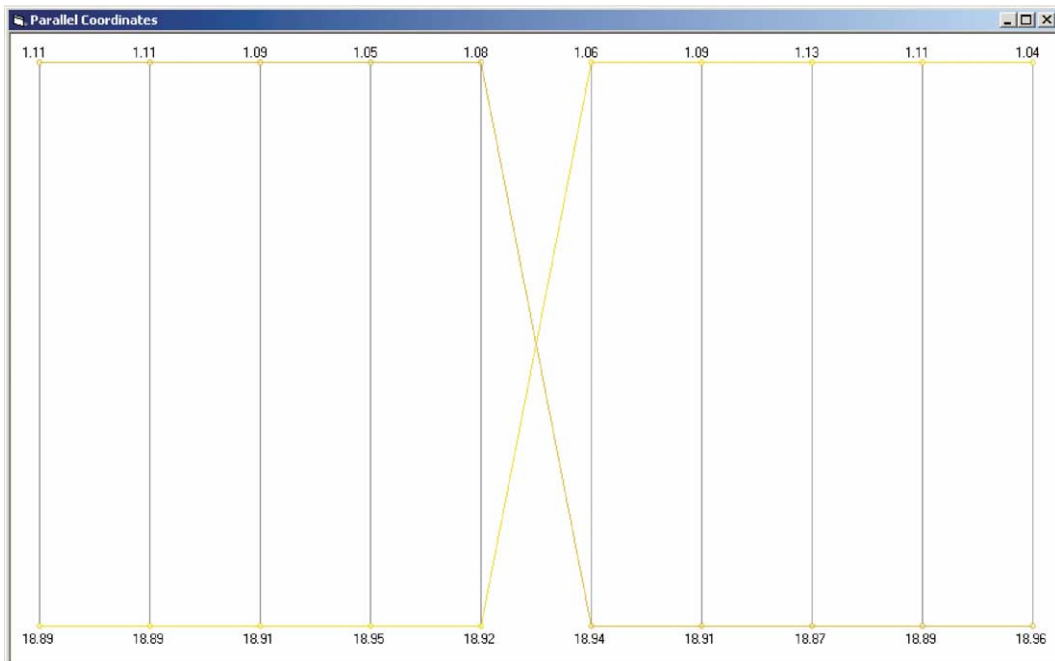


Fig. 13

Fig. 12. Snapshot 6000 showing all ten parallel coordinates in all ten dimensions.  
 Fig. 13. Snapshot 10000 showing all ten parallel coordinates in all ten dimensions.

second group are also coincident at a different location to the first group.

Figs. 11–13 reveal more information than the distance graph shown on the left in Fig. 6 even though they were generated from the same data set. Thus, parallel coordinates can be significantly less sensitive to change than distance graph displays.

### 3.2. Parallel coordinates: example two

The use of distance graphs to view the output from the gravity transform is somewhat limited to smaller values of  $n$ . In this section, the data used to create the distance graph in Fig. 8 is viewed using parallel coordinates. Recall that the assembly of connected neurons is given in Fig. 7 and that  $n=50$ . The trial lasted 20 s leading to 20 000 snapshots in the animation (time step = 1 ms).

Note that due to the large number of particles a legend is not included but grouping is described in detail. Additionally, due to the limitations of the equipment available for displaying output, only the most significant 29 axes of the overall 50 are captured in each snapshot. This is satisfactory to analyse this example as the major activity is confined within neurons 1–20.

Fig. 14 shows snapshot 5000 of the animation. Close inspection reveals that the group of particles denoted by the straight line along the bottom of the plot includes all the particles 11–20. Note that a zoom facility is used to isolate groups and identify these groups accurately. It is also possible to suggest that two more groups may exist in the top left of the plot.

Fig. 15, showing snapshot 7500, confirms the conjecture that two more groups exist. Closer inspection reveals that the upper group throughout this plot relates to particles 31–50. The lower group that zigzags at the left denotes particles 2–10. This directly reflects the neuron assembly. Note that a zoom facility was used, to include/exclude particles in order to accurately identify the members of each group.

### 3.3. Future use of parallel coordinates

One of the most significant benefits of using parallel coordinates in comparison to distance graphs, is the fact that, evidence to date suggests, they are significantly less sensitive to the specification of aggregation, decay and increment parameters. Numerous trials were performed in which the data used to generate ‘poor’ distance graphs was represented in parallel coordinates. In general, this data was analysed successfully using parallel coordinates.

One of the main disadvantages of parallel coordinates is the impact that standard display equipment limitations has on their use. This was demonstrated in Section 3.2, where the assembly of 50 neurons, highlighted the fact that the maximum number of vertical axes, easily viewed in parallel coordinates, is approximately 30 (based on a standard monitor size). Future research will incorporate alternative projection facilities to overcome these limitations.

Recently, within the domain of ‘Information Visualisation’ much research has focused on the development of parallel coordinates in order to analyse even greater quantities of data. An example of this is the concept of hierarchical parallel coordinates (Fua et al., 1999). Future work will investigate using hierarchical parallel coordinates for representation of larger data sets.

## 4. Dimension reduction

In addition, to alternative methods of data representation, it is also possible to reduce the quantity of data represented whilst maximising the amount of information portrayed. PCA is one method of achieving this type of reduction.

PCA (Biswas et al., 1981; Gnanadesikan and Wilk, 1969), is used to find the ‘best’ subspace for the projection of multi-dimensional data. This method achieves a higher degree of representation accuracy by maintaining as much of the overall data structure as possible. The PCA method used in these trials was achieved by analysing the covariance matrix of a selected time slice of the output from the gravity transform. A co-variance

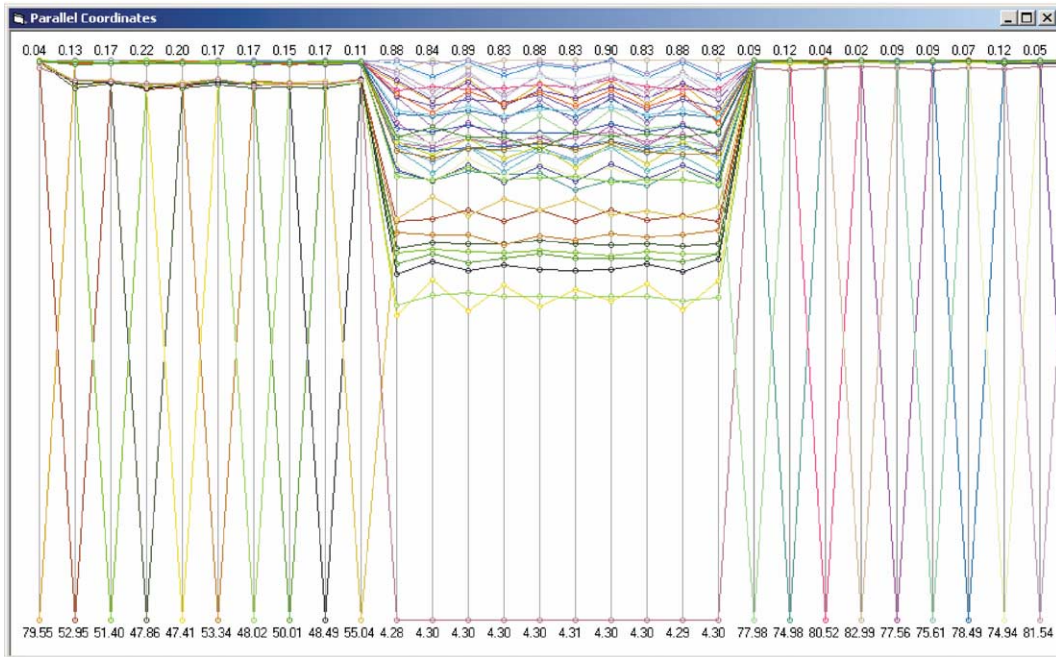


Fig. 14

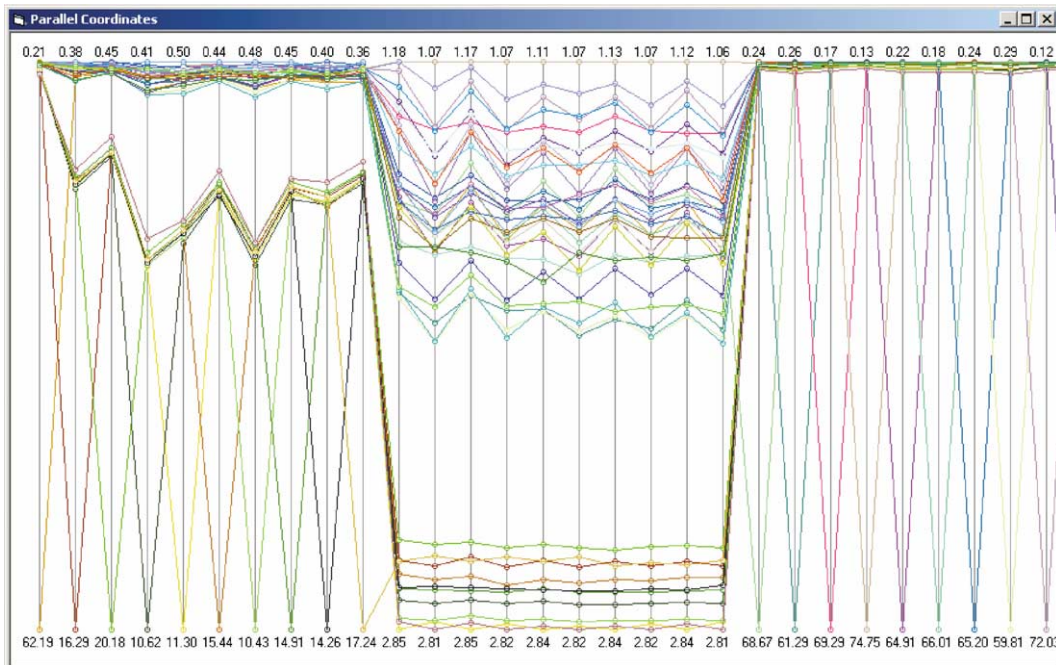


Fig. 15

Fig. 14. Snapshot 5000 of the animation showing all 50 parallel coordinates in 29 dimensions.  
 Fig. 15. Snapshot 7500 of the animation showing all 50 parallel coordinates in 29 dimensions.

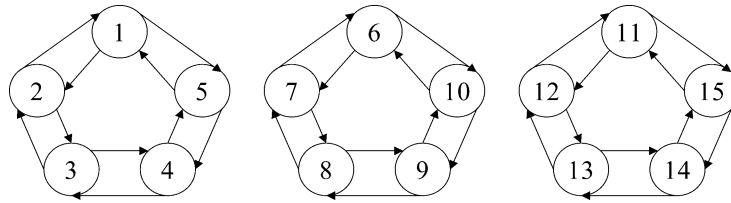


Fig. 16. Specification of the connections between the 15 neurons used as input to the spike train generator.

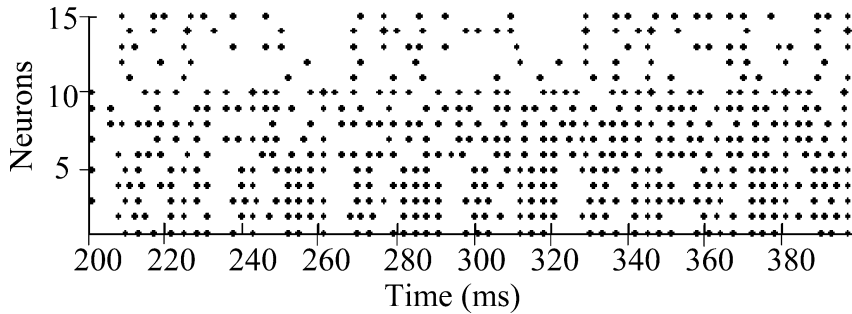


Fig. 17. A raster plot depicting a portion of the 15 spike trains generated for the neuron assembly shown in Fig. 16.

matrix depicts the position of each particle in each dimension at a chosen time  $t$ . Note that  $t$  is chosen to be a point after which useful aggregation has occurred. The eigenvalues and eigenvectors are derived using Householder’s reduction and an Implicit QL algorithm, (Press et al., 1992; StatLib, 2002). Subsequently, the same eigenvectors are used to project each time slice of the output from the gravity transform.

4.1. PCA example one

Using the assembly of 15 neurons given in Fig. 16, spike train data was generated. This trial lasted 20 s and neurons had a firing rate of approximately 0.3231 spike per s. Fig. 17 shows this data using a standard raster plot representation.

Subsequently, the data was input to the gravity transform where  $a = 0.1$ ,  $\tau = 0.2$  and  $b = 0.1$ . The resultant distance graph is shown Fig. 18 where  $s_1 = \{d_{ij}; i = 1, \dots, 10, j = 2, \dots, 10, i < j\}$ ,  $s_2 = \{d_{ij}; i = 1, \dots, 10, j = 11, \dots, 15\}$  and  $s_3 = \{d_{ij}; i = 11, \dots, 15, j = 12, \dots, 15, i < j\}$ . The data set output

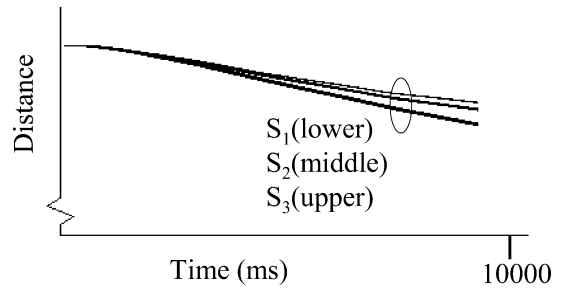


Fig. 18. Distance graph of the results of the gravity transform where  $n = 15$ ,  $a = 0.1$ ,  $\tau = 0.2$  and  $b = 0.1$ .

from the gravity transform, shown in Fig. 18 as a distance graph, was reduced to two dimensions using PCA. The output of this method is displayed in Fig. 19.

Note that the trajectory of each particle is represented as a number of discrete points in space over time. In this representation, neurons 1–5 and neurons 6–10 can be seen as two distinct groups. In addition, note that neurons 11–15 are beginning to form a third group. Recall that only two of the original 15 projected dimensions are portrayed.

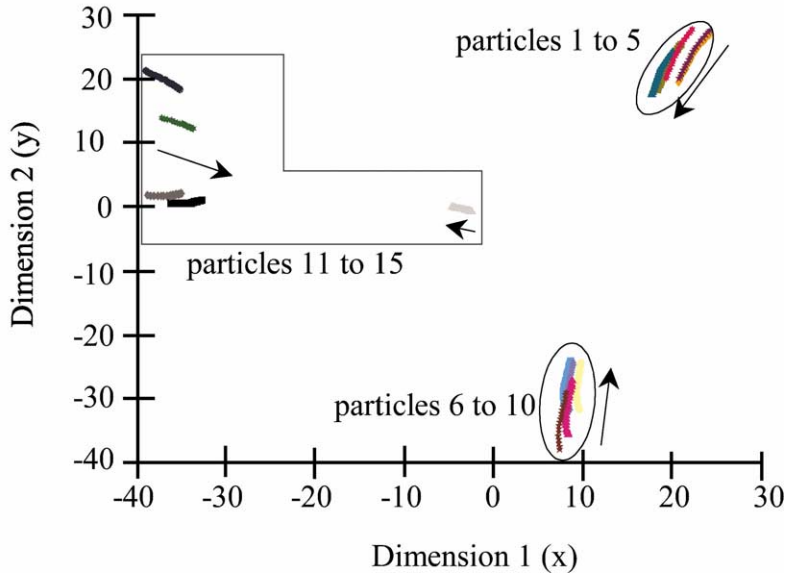


Fig. 19. This plot depicts the trajectories of all 15 particles over time, after the output of the gravity transform was reduced from 15 to two dimensions using PCA.

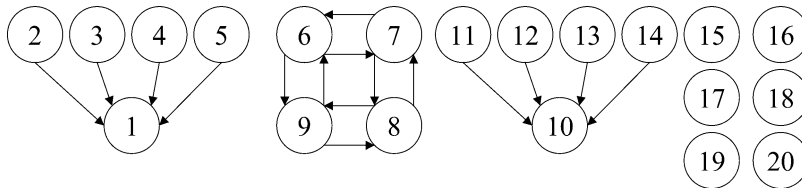


Fig. 20. Specification of the connections between the 20 neurons used as input to the spike train generator.

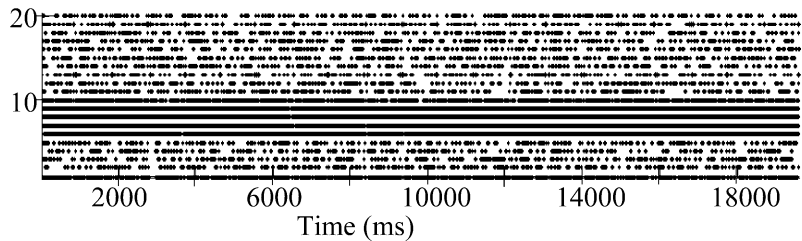


Fig. 21. A raster plot depicting a portion of the 20 spike trains generated for the neuron assembly shown in Fig. 20.

4.2. PCA example two

In Fig. 20, an assembly of 20 neurons is shown. This assembly was used to generate spike train data for a trial lasting 20 s where neurons had a firing rate of approximately 0.0438 spike per s.

Fig. 21 shows the data set generated using a raster plot.

This data set was subsequently used as input to the gravity transform with  $a = 0.5$ ,  $\tau = 0.7$  and  $b = 1$ . The distance graph shown in Fig. 22 was produced. Note that  $s_1 = \{d_{ij}; i = 6, \dots, 9, j = 7,$

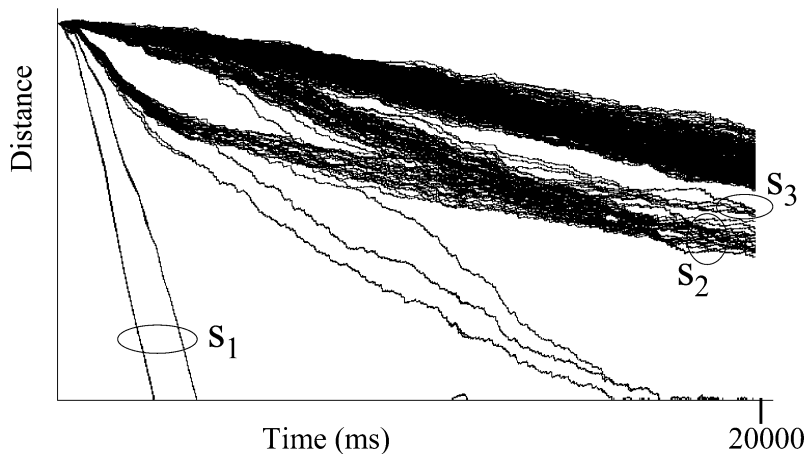


Fig. 22. Distance graph of the results of the gravity transform where  $n = 20$ ,  $a = 0.5$ ,  $\tau = 0.7$  and  $b = 1$ .

$\dots, 9, i < j\}$ ,  $s_2 = \{d_{ij}; i = 1, j = 2, \dots, 5\}$  and  $s_3 = \{d_{ij}; i = 1, j = 10, \dots, 13\}$ . In Fig. 22, neurons 6–9 ( $s_1$ ) are easily identified whilst neurons 1–5 ( $s_2$ ) and 10–14 ( $s_3$ ) are masked by the other distances pairs plots.

The data set output from the gravity transform, shown as a distance graph in Fig. 22, was reduced to two dimensions using PCA. The output of this method is displayed in Fig. 23.

The distance pairs relating to the solitary neurons cluster at the centre of the display, indicating that they have no tendency to cluster with other neurons. A zoom of this cluster is shown in Fig. 24.

Fig. 23 also shows another group, neurons 6–9 clearly distinct from the rest. However, it is difficult to draw any further conclusions from this plot. Note that the arrows indicate the general direction of the trajectory.

#### 4.3. Limitations of PCA

As demonstrated, dimension reduction can be used to convey the majority of the information contained in the original  $n$ -dimensional data set. Moreover, it can result in an improved representation of the data. However, a balance needs to be maintained between dimension reduction and loss

of information. A comparison of Figs. 22 and 23 highlights the loss of information that can result from dimension reduction. Note that the use of a non-linear reduction may ease this problem (Sammon, 1969).

In addition to the reduction of dimensions, investigating innovative methods of representing large quantities of data, such as parallel coordinates, can alleviate the problem of representing vast data sets.

## 5. Conclusions

This paper reports on the use of visualisation in the analysis of synchronous activity in multi-dimensional spike train data derived from the gravity transform algorithm. On the basis of our simulations we derived approximate values for the algorithm parameters: aggregation, charge and delay. Numerous trials were run using relatively large numbers of spike trains. These were successful whilst highlighting the limitations of using distance graphs to output the data.

Parallel coordinates and animation techniques were used to support the analysis of these vast data sets. In addition to these innovative methods of

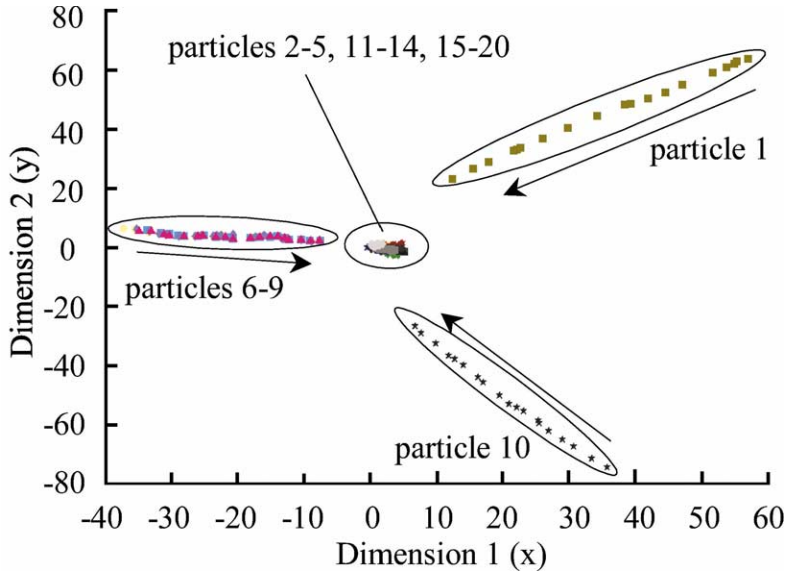


Fig. 23

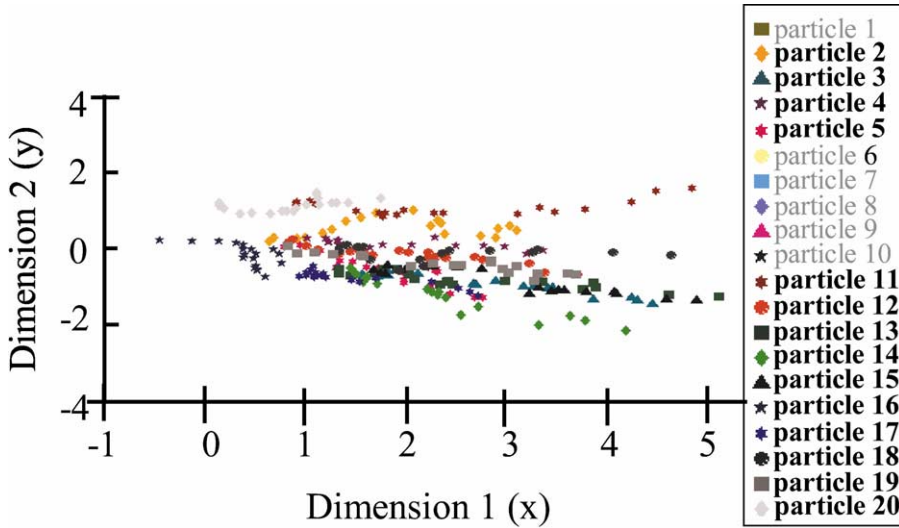


Fig. 24

Fig. 23. This plot depicts the trajectories of all 20 particles over time, after the output of the gravity transform was reduced from 20 to two dimensions using PCA.

Fig. 24. This plot depicts the trajectories of particles 2–5 and 11–20, after the output of the gravity transform was reduced from 30 to two dimensions using PCA.

display, concentration of the data can also alleviate the problem of representing vast data sets.

PCA was used to reduce the quantity of data whilst maximising the quality of the data retained.

This method is very useful in creating manageable data sets, yet limited due to display. Hence, future work will incorporate the use of parallel coordinates for the display of PCA output data in

addition to output directly from the gravity transform.

In conclusion, no single method will overcome the problems of analysing synchronous activity in multi-dimensional data sets. However, the combination of many diverse methods from domains such as mathematics and statistics, information visualisation and graphics will provide a very practical platform on which to analyse this quantity of data.

### Acknowledgements

This research is supported by the Engineering and Physical Sciences Research Council grant number GR/N04904. Additionally, the research of RB is supported in part by the Russian Foundation of Basic Research (grant 99-04-49112) and by EPSRC (grant GR/N63888/01).

### References

- Baker, S.N., Gerstein, G.L., 2000. Improvements to the sensitivity of the gravitational clustering for multiple neuron recordings. *Neural Comput.* 12, 2597–2620.
- Biswas, G., Jain, A.K., Dubes, R.C., 1981. Evaluation of projection algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* 12 (6), 701–708.
- Borisyuk, R.M., Borisyuk, G.N., 1997. Information coding on the basis of synchronisation of neuronal activity. *BioSystems* 40, 3–10.
- Fries, P., Neuenschwander, S., Engel, A.K., Goebel, R., Singer, W., 2001. Rapid feature selective neuronal synchronization through correlated latency shifting. *Nat. Neurosci.* 4 (2), 194–200.
- Fua, Y., Ward, M., Rundensteiner, E., 1999. Hierarchical parallel coordinates for exploration of large datasets. In: *Proceedings of Visualization'99*, pp. 58–64.
- Gerstein, G.L., Aertsen, A.M., 1985. Representation of cooperative firing activity among simultaneously recorded neurons. *J. Neurophysiol.* 54 (6), 1513–1528.
- Gerstein, G.L., Perkel, D.H., Dayhoff, J.E., 1985. Cooperative firing activity in simultaneously recorded populations of neurons: detection and measurement. *J. Neurosci.* 5 (4), 881–889.
- Gnanadesikan, R., Wilk, M.B., 1969. Data analytic methods in multivariate statistical analysis. *Multivariate Anal.* 2, 593–638.
- Inselberg, A., Dimsdale, B., 1990. Parallel coordinates: a tool for visualising multi-dimensional geometry. In: *Proceedings of Visualization'90*, pp. 361–378.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannerty, B.P., 1992. *Numerical Recipes in C: the Art of Scientific Computing*, second ed.. Cambridge University Press, New York.
- Sammon, J.W., Jr, 1969. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* C-18 (5), 401–409. STATLIB. <http://lib.stat.cmu.edu/multi/>.
- Wegman, E.J., 1990. Hyper-dimensional data analysis using parallel coordinates. *J. Am. Stat. Assoc.* 411 (85), 664–675.