I. Troch, F. Breitenecker, eds.   ISBN 978-3-901608-35-3

NET-BASED PHASE-ANALYSIS IN MOTION PROCESSES

Jürgen Perl

University of Mainz

Abstract. Although data can be taken from motions automatically it is still a problem to get useful information from them. In contrast, the huge amount of automatically recorded data can hide that information. Standard analysis and search algorithms are not helpful as long as the 'striking' property is not known a priori – e.g. if the unknown reason for a non-optimal motion is looked for.
Approaches and experience from the last ten years have shown that artificial neural networks of type SOM (Self Organizing Map) can be extremely successful in the task of making data transparent and transferring them to useful information.
To meet the different requirements of data pre-processing, net training, and data analysis our working group has developed the DyCoN-approach (Dynamically Controlled Network), which offers a number of helpful concepts and components.
Three typical examples are introduced in case studies, dealing with motion processes from different disciplines and demonstrating the way how networks like DyCoN together with phase diagrams support the analysis of motion processes.

1 Introduction

During the last years automatic data recording of motion processes has become much more easy and comfortable. Taking 100 up to 1000 or even more data sets per second from a motion and automatically storing it to a file or a data base is no problem anymore.

However, the result of that automatic recording process, the huge amount of data, very often just hides the information instead of making it available (see Figure 1). It is up to 'intelligent' algorithms to discover major and interesting information from those unreadable data.

Simple searching algorithm, statistical methods and even data mining approaches are often not very effective for finding striking features in complex data – in particular if the aim of the search, the 'striking' property, is not known a priori.

5,03104639	6,46911097	0,10483552	5,15031385	6,59917021	0,04163717	6,47479677	7,69050312
5,03777027	6,68576622	0,10780407	5,11890030	6,62742519	0,02438611	6,73762274	7,62696791
5,05529451	6,86134338	0,10682026	5,10009193	6,60800695	0,01897740	6,89312649	7,59397602
5,05457735	6,98835230	0,05482650	5,11600971	6,61605597	0,02359813	7,01356792	7,53796482
5,07145643	7,18406582	0,05096275	5,08964396	6,60420275	0,02546339	7,12869978	7,48530436
5,12005520	7,36243296	0,05067700	5,10526752	6,62029648	0,03157387	7,21657324	7,54475403
5,09289598	7,39977694	0,05047477	5,09392023	6,64073515	0,03073863	7,23693895	7,49039268
5,11981010	7,47275782	0,03159675	5,09944296	6,70974827	0,08942872	7,28861475	7,49125338
5,11187792	7,48516655	0,03267447	5,13252068	6,68009186	0,18040624	7,24033213	7,46992397
5,13529968	7,49355793	0,03783185	5,14509487	6,79271984	0,25275797	7,21726227	7,51670742
5,12905788	7,46498203	0,02202561	5,16971493	6,89359522	0,32274479	7,26564837	7,49506187
5,11754608	7,49200201	0,03080530	5,22191095	7,05952215	0,32932368	7,27288485	7,48656034
5,10944510	7,49055338	0,02704750	5,26726437	7,11704540	0,30878049	7,39904737	7,39434052
5,15156174	7,49714088	0,04466580	5,38063288	7,35691357	0,24213991	7,55183411	7,24390221
5,15175438	7,52105045	0,02303171	5,43032360	7,60129547	0,15511097	7,70289564	6,99997091
5,14069557	7,53045273	0,02865566	5,46706867	7,84140921	0,11364356	7,93202829	6,83660173
5,12898445	7,51534128	0,03817748	5,49047852	8,05863285	0,13588946	8,10264778	6,56999922
7,24770546	7,31500292	-0,15191412	7,75247002	7,98050785	-0,07523750	8,30122757	6,38929176
5,18993521	7,66030455	0,06429993	5,53000021	8,43306923	0,13131590	8,43180656	6,20461416
5,24514389	7,66827488	0,05393844	5,50694275	8,48431873	0,08367404	8,52183056	5,99114799

Figure 1. Example of automatically recorded data (player positions in volleyball, part).

Approaches and experience from the last ten years have shown that artificial neural networks of type SOM (Self Organizing Map) can be extremely successful in the task of transferring data to information. The way they do it is characterized by a kind of fuzzy-mapping, which reduces amount and precision of data dramatically, making transparent the very information structure behind the data. In particular if the data are highly correlated - as are the attribute values at one point in time or the time-depending attribute vectors in case of motion processes - this approach is extremely helpful ([2], [4], [10], [13]). There are also examples of successful approaches in the field of game analysis ([1], [3], [7], [9] [12]) as well as in creativity analysis ([5], [6], [11]).

The next paragraph gives a very brief overview on how SOM-type networks work and why fuzzy and aggregated data help for a better recognition and analysis of process patterns.
The following paragraphs introduce three typical examples, dealing with motion processes from different disciplines, demonstrating the way how networks together with phase diagrams support the analysis of the corresponding process patterns.

2 Net-based data processing and presentation

2.1 Self organizing maps (SOM)

Briefly spoken, a network of SOM-type works as follows (see Figure 2; for more details see [8]):

During the training, data are input to the neurons of the net, where the neurons decide in a self-organizing way which data are right to them for learning. This means that the neurons change their stored information during the data input until, at the end of the training, the neurons contain (most of) the input information. However, there is a discrepancy between the huge amount of possible input data and the comparably small number of neurons of a net (normally about 400) – which means that one neuron has to learn not just one but a class of different but similar inputs. This way such a class of inputs is aggregated to a fuzzy representation which characterizes the neuron.

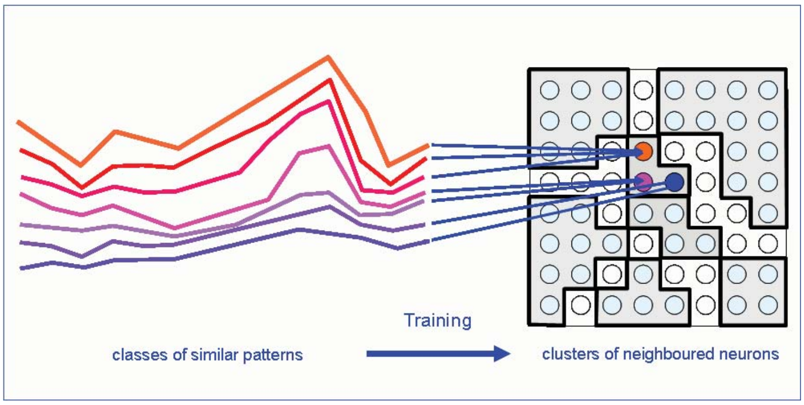

Figure 2. Network training with input-neuron-correspondences and neuron-clusters.

After having been trained, the net can be tested: A single input can be fed to the net in order to find out the neuron the entry of which is closest to and so characterizes the type of the input.

2.2 Trajectories and neuron colouring

In case of motion processes, the single input normally is an attribute vector containing the information of the motion at one point in time – e.g. coordinates or speed of the athlete or coordinates, speed, or angles of articulations. Mapping the input data of a motion process step by step to the corresponding neurons and connecting the successive neurons to a trajectory then leads to a 2-dimensional mapping of the high-dimensional process (see Figure 3 and [2]).

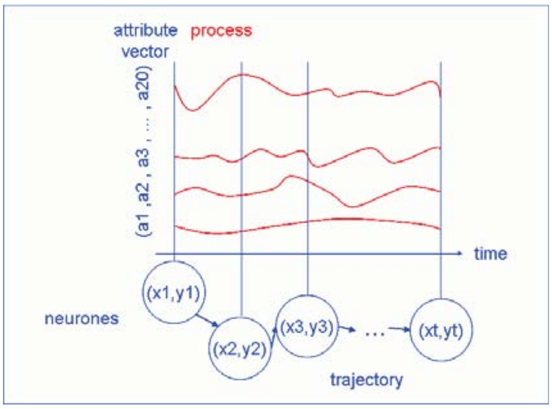

Figure 3. Trajectory as a sequence of connected neurons.

Obviously, the net-based data classification can reduce the precision of information dramatically. The reason is that – in particular in motion processes – the correlations of attribute values and process steps are rather high due to mechanical and anatomical restrictions. Therefore the reduction to major structures is not a lost but the only way of making process patterns visible.

Moreover, we can even improve the learning and analysing processes if reducing amount and precision of the data in a pre-processing procedure.

If process data have once been transformed to trajectories they can be compared – e.g. under the aspects of intra-individual stability or inter-individual similarity. Also special types or strange parts of motions can be recognized, as had been done with rowing in one of our first approaches (see [10]). As Figure 4 shows, there is a clear inter-individual similarity between the trajectories of all the presented strokes. It can, however, easily be seen that the strokes of rower A are more stable than those of rower B.

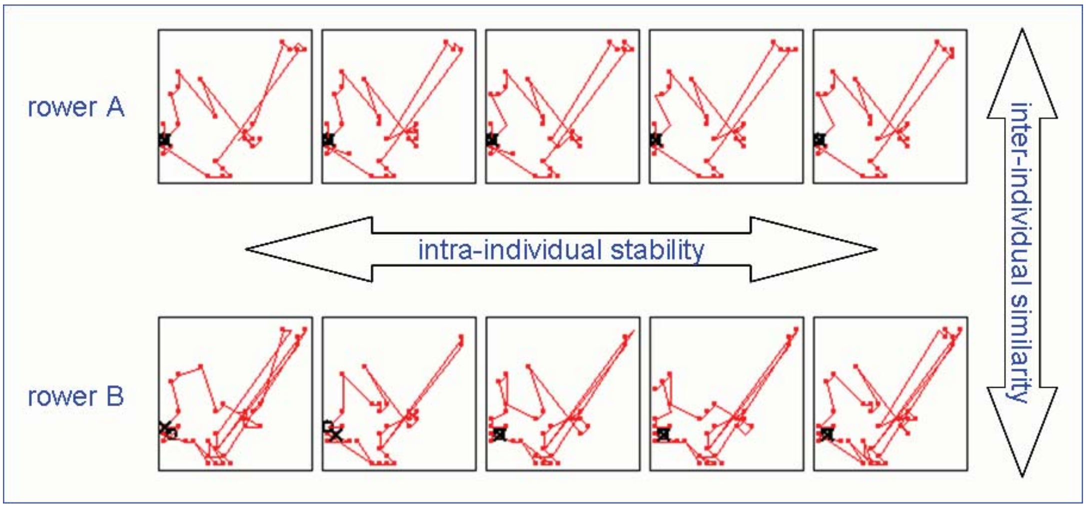

Figure 4. Trajectories from rowing, each trajectory meaning one complete stroke.

Although on the first glance the net-based trajectory approach works successfully there can arise some problems with the local similarity:

Normally, neighboured neurons of a net have similar entries, and distant neurons have dissimilar ones. Normally - but not always. Due to the fact that the network gives a 2-dimensional mapping of a high-dimensional input space, sometimes quite different areas of the input space can be mapped to neighboured areas of the net, and sometimes one area of neurons is completed by a distant one with similar entries.

Consequently, differences between trajectories not necessarily mean different processes. Instead, the differing neurons can belong to two parts of the same similarity area. In turn, trajectories can look quite similar because of closely neighboured neurons, which however might belong to quite different types of entries.

Note that this behaviour does not often appear or, at least, is not often a problem. Nevertheless, it helps a lot to use neuron presentations which reflect structural or semantic similarity. One way to do this is to colour similar neurons equally – either with regard to structural similarity clusters or with regard to semantic areas.

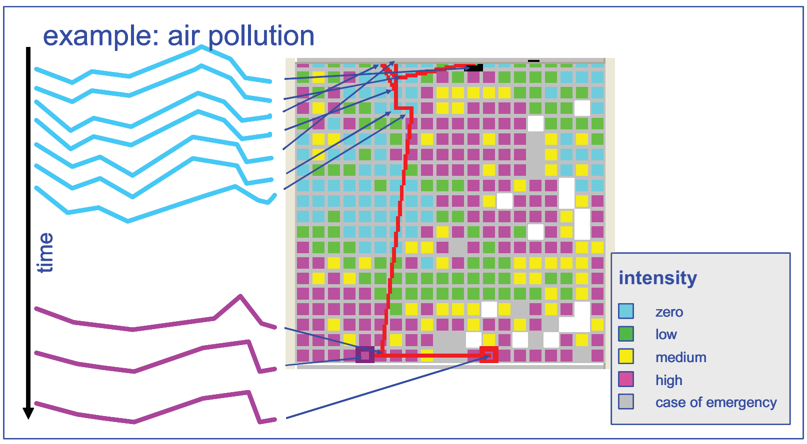

Figure 5. Colour landscape with trajectory (see explanation in the text).

The structural similarity clusters can be generated automatically using the entries of the neurons. The semantic areas like phases of a motion or a game can be found by calibrating the neurons with the real processes. In particular in case of games or motion processes the second way of neuron colouring is to prefer, because the types of the neurons then can tell what context or phase the trajectory respectively the corresponding process is moving through at the respective point in time.

Figure 5 shows a semantically coloured network representing the intensities of air pollution together with a trajectory representing a time-depending process of increasing pollution intensity. The obvious advantage of this approach is that those trajectories can be taken for synchronous semantic comparisons. (The example stems from a project dealing with rush hour pollution processes in big cities.)

A further advantage of neuron colouring (or multiple calibration) is that it has to be done just once: If the net has once been trained satisfyingly, the clusters can be taken for a first step colouring, which in a second step can be refined using real data to avoid just seeming similarities or dissimilarities.

2.2 Two-level-analysis

As will be figured out in more details in case studies 2 and 3, trajectories and neuron colouring not only help for a better understanding of processes but moreover can be taken for further net-based analyses on a second net-level: Colours can be represented by numbers. Doing it this way, the 2-dimensional trajectory of a process can be transformed to a 1-dimensional sequence of numbers – i.e. an attribute vector – which then can be used for training a new net. The neurons of this net represent types of the original processes, respectively their trajectories, and therefore the net is called process-net.

Using such a process-net, the different types of processes can automatically recognized by corresponding neurons. This approach is very helpful in case of similarity or stability analyses, which are dealt with in the case studies in Section 3.

However, although there are enough automatically recorded data for the first level analysis of processes, the number of those processes normally is comparably small and not sufficient for the training of a conventional SOM. Not least in this special situation the DyCoN-approach that we developed in our working group is helpful:

2.3 DyCoN-approach

Very briefly spoken, the DyCoN-approach replaces the external learning control of a conventional SOM by an internal state-based control of the individual neurons (see [7], [8], [9]). Besides a number of different advantages, this approach allows for continuous training, training in phases and therefore stopping, continuing, and completing training processes.

In particular in case of small numbers of original data, these data can be taken for Monte Carlo-generation of data. The generated data in a first training phase can be used to train the basic distribution to the net. In a second phase the precision of the net can be improved by a completing training with the original data. Both training phases can be controlled interactively, avoiding or adding more training data on demand. Therefore, the nets we use are always optimally trained to the maximum point of information representation.

Moreover, the DyCoN-approach offers a number of additional features which particular support the analysis of time-depending processes. Besides of integrated pre- and post-processing functionalities, some of those analysis features are automatic 2-dimensional 0-1-calibration, manual multidimensional calibration (neuron colouring), automatic generating of trajectories and transforming them to phase diagrams (see case studies).

All features can be handled fully interactively and therefore support the analysis processes in a quite comfortable way.

It finally should be added that the DyCoN-approach is not only used in the area of motor analysis but in a wide range of applications, spreading from fraud detection over air pollution recognition and logistic process optimization to health care and rehabilitation processes. Two major reasons for the success of the DyCoN approach in all of those applications are its abilities of integrating generated data in the training process and following development processes with continuously completing training.

3 Case Studies

In the following paragraphs three case studies are presented that show typical ways of colouring neurons and run phase diagram-based analyses on networks.

3.1 Case study 1: Feinting in handball

The data stem from simple left-right-feints with durations of about 2 seconds. The originally 100 data sets per second were reduced to 10 so that each process contained 20 data sets.

In a first step the net was trained with the reduced process data and the neurons were calibrated using colours that encoded particular types of motion. The result is presented in Figure 6.

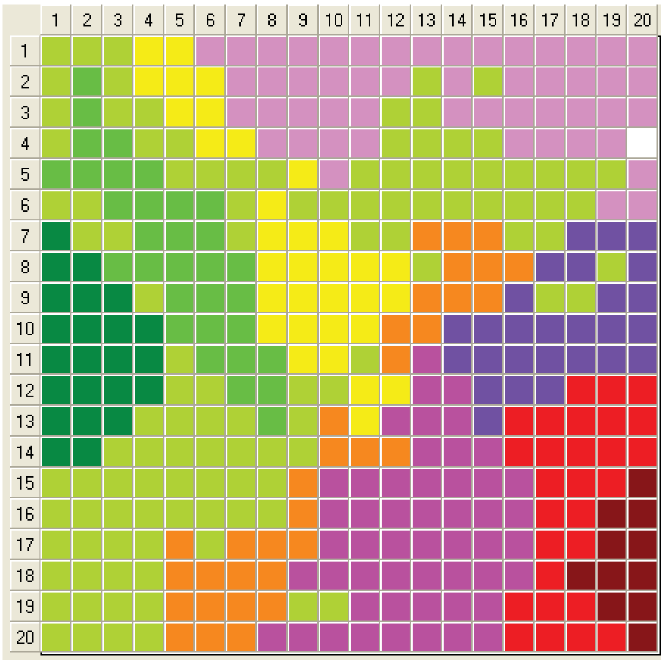

Figure 6. Feint net with motion type coloured neurons.

In the second step the data were fed to the net resulting in trajectories as 2-dimensional pictures of the process. In the third step those trajectories were projected to a diagram the colours of which represent the motion type colours of the neurons.

Such presentations are called phase diagrams because the colours can be understood as phases of the process, like move left or move right in case of the feint processes. Note that the transfer from trajectories to phase diagrams again reduces the dimensionality. As mentioned above, this is in so far important as the 1-dimensional phase diagrams very easily can be transformed to vectors of numbers, which then can be taken for net training on a second level. This method is dealt with in more detail in the next paragraphs.

Moreover, phase diagrams can give important information about the "regularity" of a motion, if the colour sequence reflects the "direction" of a motion from start to end – as is the case in Figure 7, where the feints normally should start with the green phases and end with the magenta ones.

In detail we have the following results for the 6 athletes (numbered from top to bottom by 1 to 6):

(1) Feint 1 and 2 are looking strange, feints 3 to 5 are similar to each other and seem to be regular with an extended finishing phase. The regarding trajectories look more similar than the phase diagrams do.

(2) The feints show a high similarity and regularity, regarding trajectories as well as regarding phase diagrams.

(3) The trajectories look similar but quite different from those of athlete 1. The phase diagrams, however, show that they very close to the regular type – with some "noise" in the middle of the process.

(4) The trajectories show an obvious similarity. They remind on those of athlete 1 and 2. The phase diagrams, however, show that they are more similar to those of athlete 3 – except the finishing phase and neglecting that particular "noise".

(5) The trajectories look rather strange an irregular, and so do the phase diagrams. The first two feints seem to be quite different from the second three ones.

(6) The trajectories look different from those of the other athletes. But the phase diagrams remind strongly on those of athlete 3.

Summed up, the 6 athlete show rather individual types of feinting with a clear orientation on the "regular" process. Generally, two sub-types of the phase diagrams can be distinct regarding the finishing phase, namely the magenta one (mainly athletes 1, 2, 4, and 5) and the brown one (mainly 3, 5, and 6).

Only the first two feints of athlete 1 and all feints of athlete 5 seem to be irregular. The differences between the feints of player 6 become more apparent in the phase diagrams. Both, trajectories and phase diagrams, enable for detecting conspicuous patterns like feint 1 and 2 of athlete 1 or feint 4 of athlete 5.

In general the phase diagrams seem to be better readable, which is due to the 1-dimensional relation between time and phase (colour).

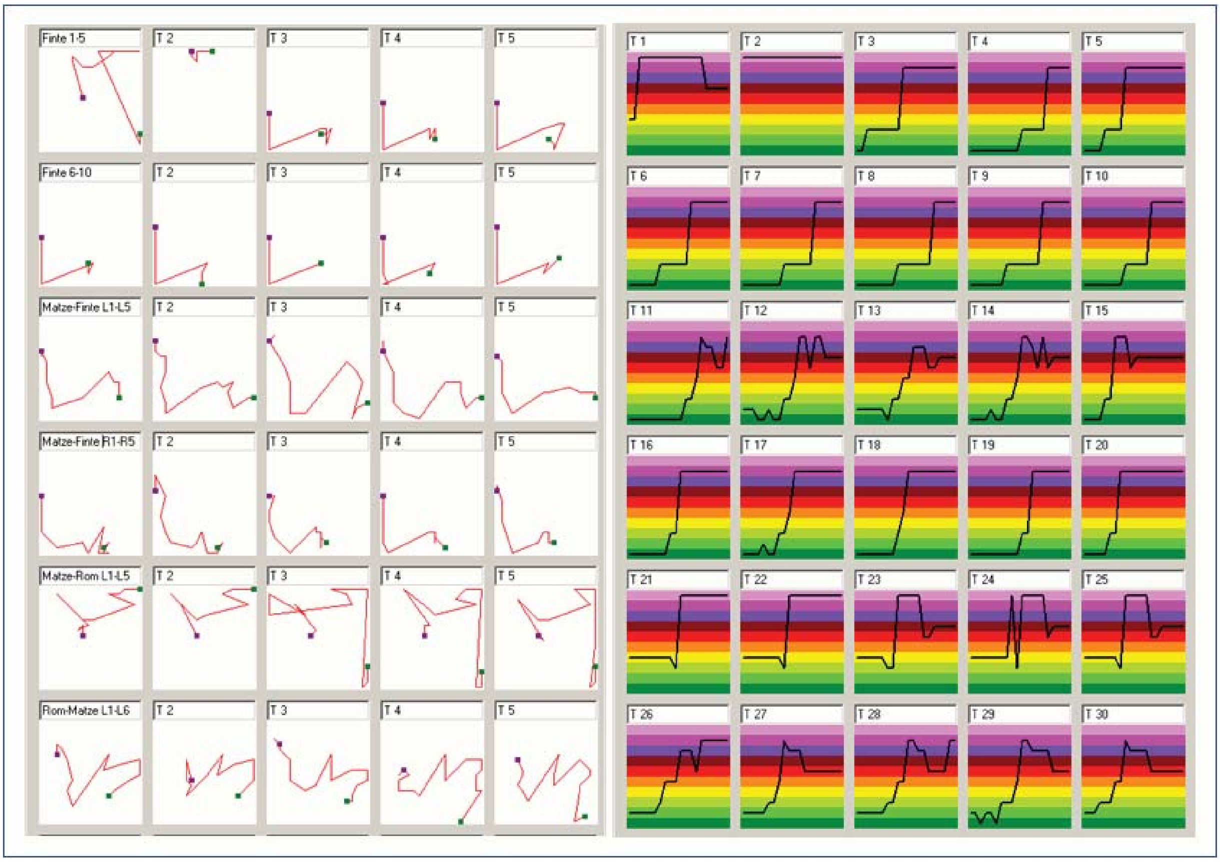

Figure 7. Trajectories and phase diagrams of feints in handball: 6 athletes (rows) with 5 feints each (columns).

The following case study shows that even phase diagrams sometimes fail in making processes transparent. In this case, however, phase diagrams can be taken for a net-based analysis on a second level as has been mentioned above.

3.2 Case study 2: Aiming process in biathlon shooting

The aim of the study was to find out the impact of the running load on the aiming process. The data stem from 4 athletes (***A***, ***C***, ***D***, ***I***) with 10 shots before (1) and 10 shots after (2) the load, 20 data sets per shot (my particular thanks is to Arnold Baca, University of Vienna, for those interesting data).

Different from case study 1, the data sets cannot easily be interpreted semantically.

Therefore the neuron colours were taken from automatic clustering for direct transforming the trajectories into phase diagrams (see Figure 8).

As can be seen the differences between the 10 shots before and the 10 shots after load are not always as obvious as in the case of athlete ***A***. Also intra-individual stability and inter-individual similarity of the aiming processes are not quite clear. This is due to the fact that the phase diagrams have no semantic interpretation and the 'ups' and 'downs' of the profiles have no 'natural direction' like in feinting. A way to handle this problem is to take the diagrams as patterns, which then in a second step can be trained and analysed by means of a second network.

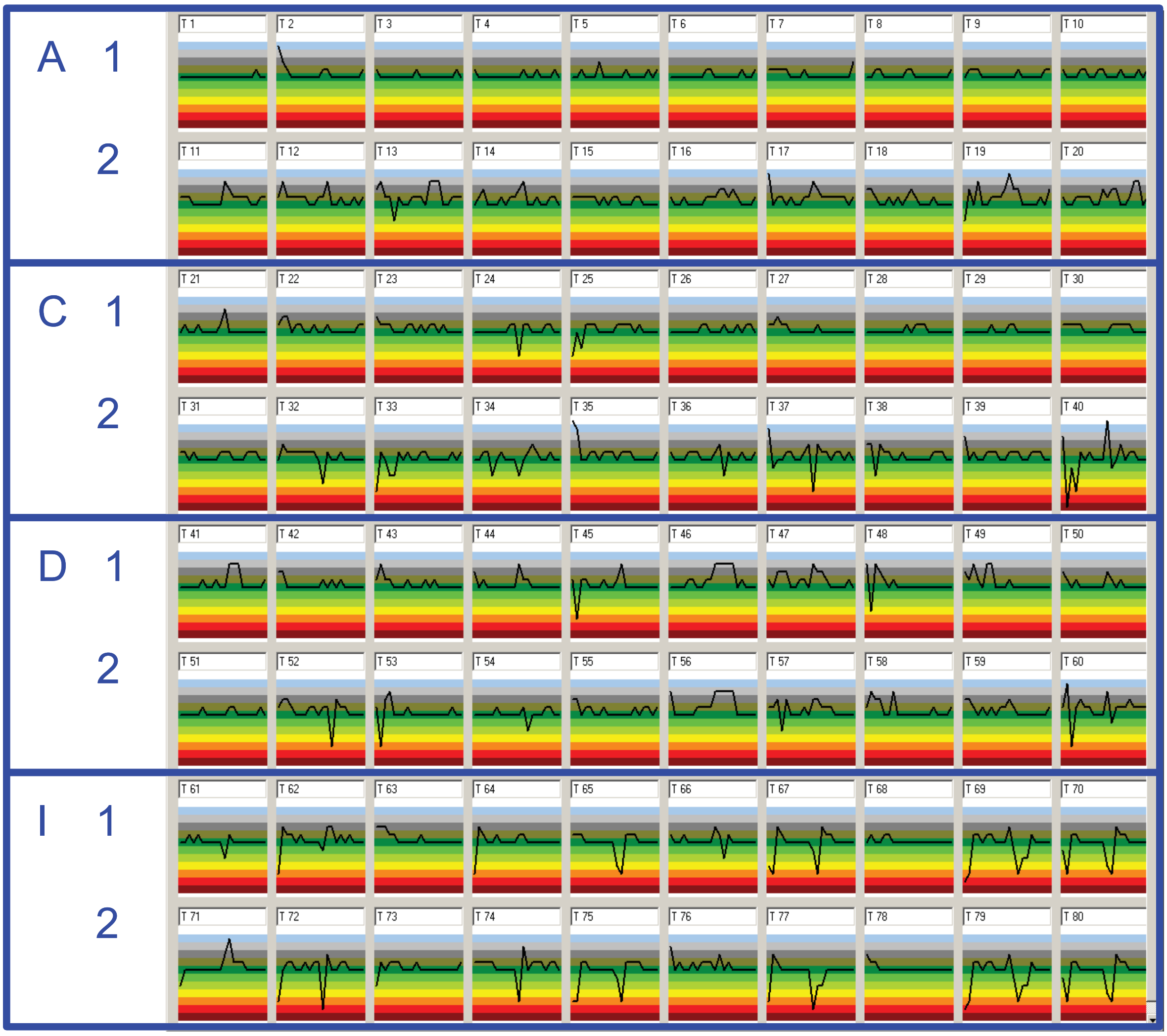

Figure 8. Phase diagrams of the aiming process in biathlon shooting.

The result is shown in Figure 9: The phase diagrams of the aiming processes are tested on a special diagram network, the neurons of which were coloured following the automatic clustering. The white colour means rare processes without representing a specific type. This is because the diagrams of length 20 define a huge space of different patterns which could not all be representatively mapped to the net. However, only one process (athlete ***I*** "after") corresponds to that group of unknown type.

Nevertheless, as can be seen from Figure 9 a lot of aiming processes are corresponding to just 5 types and so help for comparing 'before' and 'after' as well as the different behaviour of the 4 involved athletes much more easily.

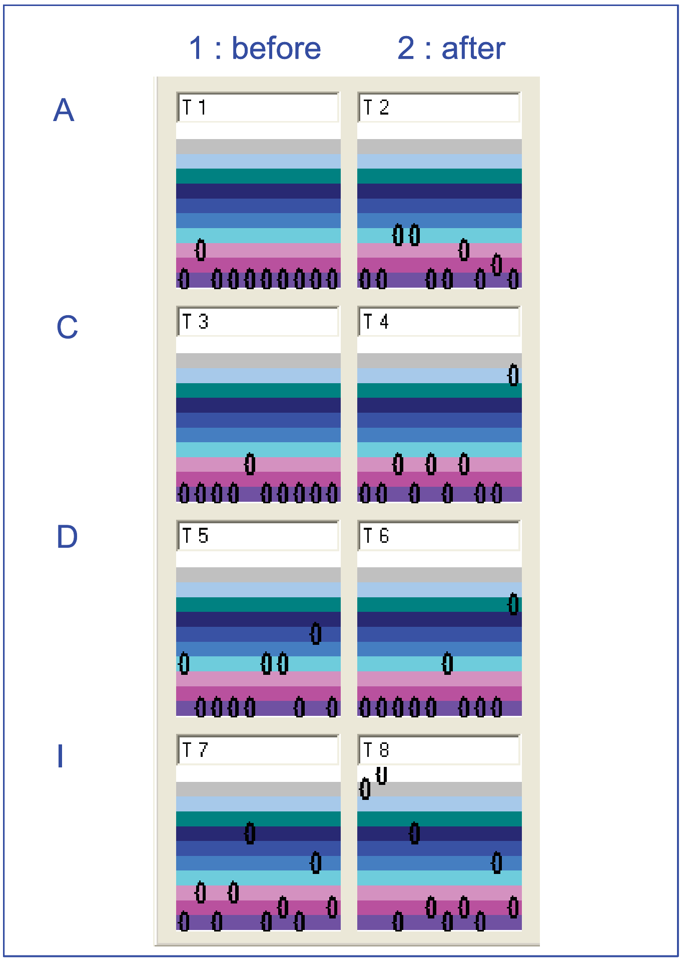

Figure 9. Diagram types of aiming processes in biathlon shooting.

Some more detailed results are:

Athletes ***A*** and ***C*** show rather similar profiles. Both are comparably stable, in particular in the phase before load. They do not have problems with aiming but could improve their relaxing or concentrating process.

Athlete ***D*** seems to be more stable after the load than before, which is really surprising. The coach should check what the reason for the somewhat unstable behaviour in the first phase is.

Athlete ***I*** shows no difference between the stabilities before and after load. The striking point obviously is not a lack in relaxing after load but the precision in aiming in general.

Finally, a deeper understanding and a closer look give even more information: Normally, the athletes shoot only five times in a sequence, while Figures 8 and 9 always show 10 shots. The reason is that the presented series contain 2 sub-series of 5 shots, from a lying and from a standing position, each. Under this particular aspect, the results from Figure 9 can be interpreted anew. It turns out that the first and the second series can be distinct rather well in the case of ***D***, while in case of ***A*** and ***C*** lying or standing obviously makes no big difference. Generally, as expected, the lying position seems to go with better stability – except case ***I***, where no difference or even a better result from the standing position can be seen.

3.3 Case study 3: Free-shots in basketball

The idea of using second level networks for the analysis of process diagrams is presented in the last case study in order to analyse and compare semantics of the processes.

The data in the example (see Figure 10) stem from 5 basketball free-shots of 7 athletes, each containing 11 data sets (my particular thanks is to Andrea Schmidt from the University of Bremen, Germany, who allowed me to take some of the results of her PhD research project).

Similar to the first case study and different from the second one the phases of the process are easily determined from start to shot/end. As in the case of feinting, stabilities and similarities on the first glance are recognizable in trajectories as well as in phase diagrams, except for a better readability of the diagrams because of their semantic 'order'.

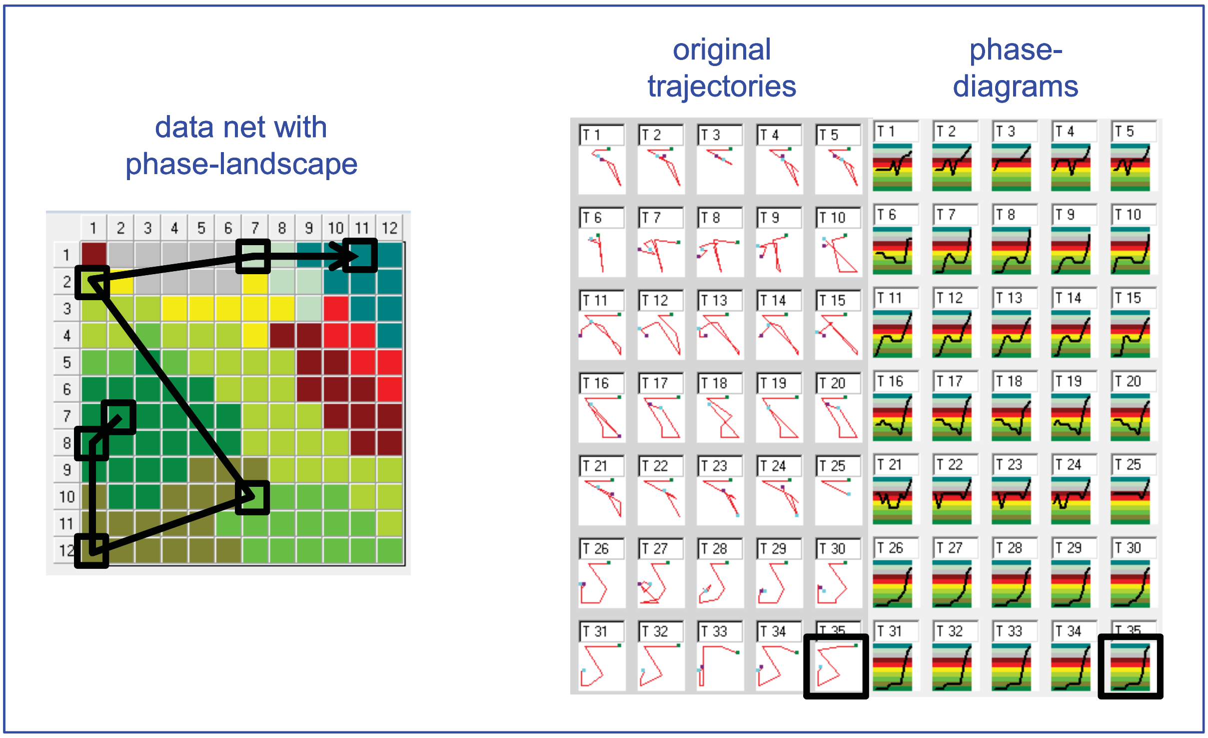

Figure 10. Trajectories and phase diagrams of free-shots in basketball.

One further advantage of phase diagrams can be seen in Figure 10: The original data net shows a typical trajectory of a free-shot. Counting the neurons covered by the trajectory results in '7'. This is different from the number '11' of the corresponding data sets. The reason is that some of the data sets correspond to identical neurons, which makes it difficult to decide if trajectories are similar or not. Phase diagrams support those decisions by representing each data set by one corresponding step. Moreover it specifies characteristic types of behaviour, like short or long phases of concentration at the beginning and slow or fast motions at the shot. Under these aspects it can be seen that the 6th and the 7th player seem to be experienced ones with a long concentration and preparing phase followed by a fast shot, while player 1 and 5 seem to have no concentration and preparing phase at all.

For a more systematic process analysis the phase diagrams of free-shots can be trained to a second net, as has been done in case of aiming process in biathlon in case study 2. The result is shown in Figure 11: The 'shot net' represents about 120 shots automatically clustered to about 20 types (coloured neurons). So far, the neuron colours of the shot net just mean the frequencies of corresponding processes. With a semantic multiple calibrating the neurons get additional colours encoding additional information: The particular meaning or semantics of corresponding processes of a neuron is encoded by a respective colour, as has been done with the types of the processes ('shot type'-calibration) and with their qualities or success ('shot quality'-calibration).

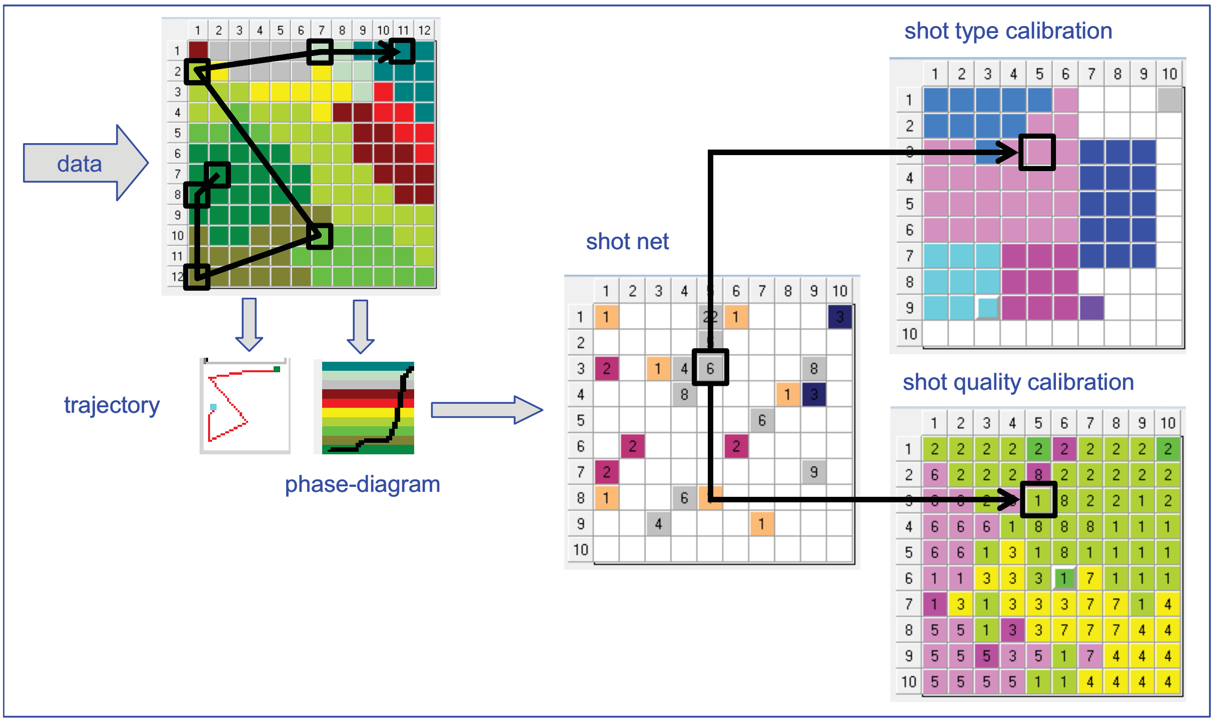

Figure 11. Net of free-shot types together with semantic colourings.

If this work of calibration has been done once, detection of type and quality can be done automatically for arbitrary free-shots (as has been got to work in the PhD-project of Andrea Schmidt):

The data to be analysed are fed to the original phase-calibrated data net, resulting in the phase diagram, which corresponds to a sequence of numbers encoding the colours respectively the phases.

This sequence of phase numbers is fed to the shot net, identifying the corresponding neuron and therefore the regarding type and quality.

4 Conclusion and outlook

The major result from the case studies from above is that artificial neural networks of type SOM can be helpful for the analysis of motion processes. The results in the area of process type recognition and comparison can be improved if adjusting data structures and pre-processing to net-based analysis features. In particular, trajectories and phase diagrams improve the analysability and transparency of process dynamics; calibration enables for combining structural analyses with semantic ones. Not least, the presented approaches allow for automatic analysis and therefore simplify the handling of mass data by far.

To meet the different requirements of data pre-processing, net training, and data analysis our working group has developed the DyCoN-approach (Dynamically Controlled Network), which offers a number of helpful concepts and components.

Not least, up to date-developments of DyCoN enable the handling of rare but relevant data, which normally is missing in SOM-approaches. This opens the door for net-based creativity analysis, where the 'creativity' of an action is understood to be 'temporary rareness' combined with 'adequateness' regarding the situation.

First results show that creativity analysis can be helpful for recognition and evaluation of compensatory components of motion processes, which might open the way for a more complex simulation and a more differentiated training of motions (see [5] and [6]).

5 References

[1] Grunz, A., Hillebrand, T., Memmert, D., Perl, J. & Schmidt, M. (2008). Typisierung komplexer Sportspielmuster mittels selbstorganisierender neuronaler Netze. In A. Woll, W. Klöckner, M. Reichmann & M. Schlag (Hrsg.), Sportspielkulturen erfolgreich gestalten (S. 191–194). Hamburg: Czwalina.

[2] Lippold, T., Schöllhorn, W. I., Perl, J., Bohn, C., Schaper, H. & Hillebrand, Th. (2004). Differenzielles Training im leichtathletischen Sprint mit Simulation und Optimierung eines Trainingsprozesses. In BISp Jahrbuch 2003, (S. 267-274). Bonn.

[3] Jäger, J., Perl, J. & Schöllhorn, W. (2007). Analysis of players' configurations by means of artificial neural networks. In International Journal of Performance Analysis of Sport, 3, 7, (pp. 90-103).

[4] Memmert, D. & Perl, J. (2006). Identifikation und Generierung kreativer Lernverläufe mit Hilfe dynamischer Neuronaler Netze. In M. Raab, A. Arnold, K. Gärtner, J. Köppen, C. Lempertz, N. Tielemann, & H. Zastrow (Eds.), Zukunft der Sportspiele: fördern, fordern, forschen (S. 78). Flensburg: University Press.

[5] Memmert, D., & Perl, J. (in press-a). Game Creativity Analysis by Means of Neural Networks. Journal of Sport Science.

[6] Memmert, D., & Perl, J. (in press-b). Analysis and Simulation of Creativity Learning by Means of Artificial Neural Networks. Human Movement Science.

[7] Perl, J. (2001). Artificial Neural Networks in Sport: Concepts and Approaches. International Journal of Performance Analysis of Sport, 1, 10–15.

[8] Perl, J. (2002). Adaptation, Antagonism, and System Dynamics. In G. Ghent, D. Kluka, & D. Jones (Eds.), Perspectives – The Multidisciplinary Series of Physical Education and Sport Science, 4 (pp. 105–125). Oxford: Meyer & Meyer Sport.

[9] Perl, J. (2008). Modelling. In P. Dabnichki & A. Baca (Eds.), Computers in Sport (pp. 121–160). Wit Press.

[10] Perl, J. & Baca, A. (2003). Application of neural networks to analyze performance in sports. In E. Müller, H. Schwameder, G. Zallinger & V. Fastenbauer (Eds.), Proceedings of the 8th annual congress of the European College of Sport Science, S. 342. Universität Salzburg, Institut für Sportwissenschaft.

[11] Perl, J., Memmert, D., Bischof, J., & Gerharz, Ch. (2006). On a First Attempt to Modelling Creativity Learning by Means of Artificial Neural Networks. International Journal of Computer Science in Sport, 5, 33–37.

[12] Pfeiffer, M. & Perl, J. (2006). Analysis of Tactical Structures in Team Handball by Means of Artificial Neural Networks. In International Journal of Computer Science in Sport, 5, 1, (pp. 4 - 14).

[13] Raab, M., Perl, J. & Zechnall, D. (2003). The Mapping of Intrinsic and Extrinsic Information in Continuous Visuo-Motor Control. In E-Journal Bewegung und Training.