VISUALIZATION OF MULTIVARIATE ATHLETE PERFORMANCE DATA

A. Telea
Eindhoven University of Technology
Eindhoven, the Netherlands
alext@win.tue.nl

P. de Hillerin
National Institute for Sports Research
Bucharest, Romania
director@sportscience.ro

V. Valeanu
Institute for Spatial Sciences
Bucharest, Romania
vlad@donnamaria.ro

ABSTRACT
We present a set of visualization methods for the analysis of multivariate data recorded from the measurement of the performance of athletes during training. We use a modified training device to measure the force, acceleration, displacement, and speed of the athlete's feet and arms while performing a certain training exercise. We are interested in visually measuring and comparing the performance over several training sessions of the same and/or different athletes. For this, we adapt and extend several visualization methods for multivariate data. First, we use an enhanced signal plot and statistics plot to visualize the regularity of repetitions within a given exercise. Second, we use a novel texture-based signal plot to eliminate signal noise and emphasize the average repetitive pattern of the exercise. Finally, we use a signal clustering technique, visualized with a matrix plot, to detect similar exercises over long periods of time. We demonstrate our approaches with actual data from training sessions of several athletes.

KEY WORDS
Multivariate visualization, information visualization, visual analytics

1 Introduction

Modern training of high-performance sports athletes starts making increasing use of computer equipment, data acquisition, and data analysis technologies. Using such techniques, both the athletes and their trainers can improve their understanding of the way athletes perform during the execution of exercises or in the field. This insight is invaluable in optimizing several aspects of the training process which are otherwise hard to detect and measure using conventional training techniques. Computer-assisted training techniques form an important element of the emerging domain of 'sports sciences' [4, 1]

In our research, we are interested in understanding and improving the performance of athletes involved in a number of sports, such as swimming, rowing, and boxing. In a typical training session, the athlete repeats a given motion, *e.g.* a row strike, or one sweep of a swim strike, several times. During the session, we measure several signals such as heart beat, and position, force, and acceleration of the athlete's limbs, as functions of time. A training session is repeated several times a day, but also during different days in a month or even over the period of an entire year. This scenario delivers hundreds of megabytes of data per athlete, which are stored for further analysis.

From such data, trainers are interested in answering a number of questions to improve the athlete's performance. Typical examples include (from simple to complex):

- What is the acquired data? A simple method for checking the raw acquired data is needed for control purposes.
- How regular is an exercise? Delivering regular, rhythmic, performance is a key quality factor for an athlete. However, measuring regularity can be quite complex.
- How do signals correlate in time? For the swimming exercise example, it is interesting to see how (and whether) the left and right arms work in rhythm.
- How do signals evolve in time? The quality (*e.g.* rhythm, speed, regularity) of an exercise can increase or decrease during a session. Ways to make this visible are of great importance.
- How do several training sessions resemble each other? This question is important for comparing either the performance of an athlete during a longer period (*e.g.* a whole year) or for comparing different athletes.

In this paper, we present a number of data analysis and visualization methods which support answering the above questions. Our methods adapt and extend existing visualization techniques for multivariate time series. First, we use an enhanced signal plot and statistics plot to visualize the regularity of repetitions within a given exercise. Second, we use a novel texture-based signal plot to eliminate signal noise and emphasize the average repetitive pattern of the exercise. Finally, we use a signal clustering technique, visualized with a matrix plot, to detect similar exercises over long periods of time. All in all, our visualizations empower trainers in detecting averages and trends and discovering outliers in the athletes' training in better ways as compared to their regular procedures.

This paper is structured as follows. Section 2 presents the work methodology and involved data types. Section 3 presents our enhanced visualization methods. Section 4 discusses our findings. Section 5 concludes the paper.

2 Methodology

The training application pipeline is sketched in Figure 1). First, we measure a number of exercise dynamic parameters, using a combination of body sensors on the athlete himself as well as several analog-to-digital converters mounted on the training device. The measurement technology is described in detail elsewhere [2]. As the athlete performs a number of repetitive exercises on the training device, several signals are monitored on his/her body and recorded. For a swimming exercise, for example, these signals include:

- the *strike force* the athlete strikes the water with
- the *displacement* of the athlete's arms
- the *speed* and *acceleration* of the athlete's arms
- the *heart beat* during the exercise

All above are measured for both the left and the right arms, yielding $N = 8$ signals. The acquired dataset is structured as follows. For each session S, all the signals $s_1, \ldots, s_N$ are recorded. A signal $s_i = \{s_i^j\}_j$ is a time series, or discrete set of numerical samples s_i^j measured by the acquisition devices, which represent the sampling of the time function $s(t)$. The sampling frequency can be different for each signal of the same or different sessions, depending on the measuring technology, but is typical quite high (tens of samples a second). The typical duration of a session is of 5-10 minutes. For each athlete in a team, several hundred sessions S_k are recorded during a whole year. All in all, this generates several hundreds of megabytes of data per athlete.

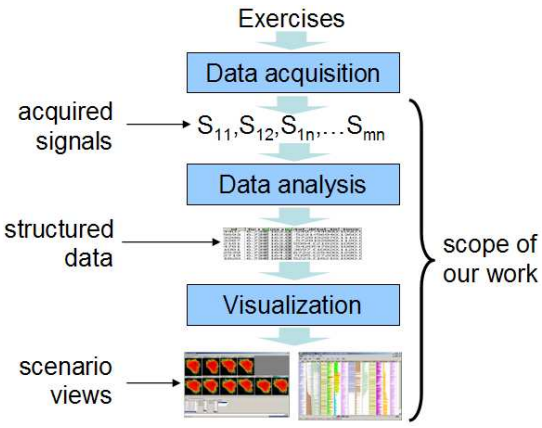

Figure 1. Data acquisition and visualization pipeline

3 Visualization

In the following, we describe several visualization methods that we have designed and implemented in a tool of ourselves in order to address typical trainer questions, such as those listed in Section 1.

Figure 2 shows a naive direct visualization of the force and position signals (for both arms) as functions of time for the swimming exercise. Given the high frequency of the signals, this visualization is of little use: We cannot detect any similarity or difference between the four displayed signals. This basic display is, however, useful to monitor the data acquisition on-line during the exercise, *e.g.* for calibration purposes.

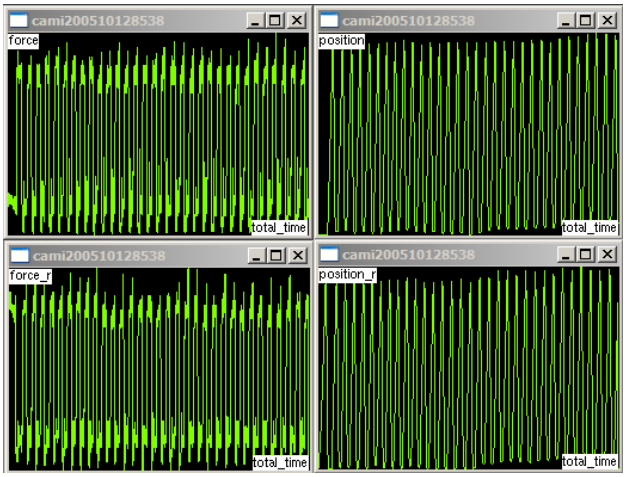

Figure 2. Force and position shown as functions of time

3.1 Two-signal visualization

Instead of visualizing a signal as a function of time $s = s(t)$, it is considerably more useful to visualize a signal s_i as a function of another signal s_j. We call this the two-signal plot. Figure 3 shows the left arm position (x) versus the right arm position (y) signals for nine different sessions (A..H). This simple but effective visualization already shows a number of useful facts. First, the *regularity* of the signals is now clearly captured in the picture. The graphs appear to 'wind' along themselves in cyclic patterns as the repetitive exercise proceeds. The *tightness* of the repetitions is now also easily visible in the tightness of the windings of the graphs. For example, exercises A-C and G-I are clearly the most regular (tight graphs), exercise D is somewhat looser but still has a clear pattern, whereas in exercises the two arms are clearly not well coordinated (loose graphs).

A second, more important, finding discovered when discussing these images with the actual trainers is that actual exercise quality can be inferred from the *shapes* of the graphs. For example, in exercise H the athlete is keeping one arm at the starting (zero) position whereas doing a full swing with the other - one arm is always at the zero position. From this perspective, we can compare the efficiency of an exercise by measuring the sum $d_x + d_y$ of the distances d_x and d_y along which a single arm position signal

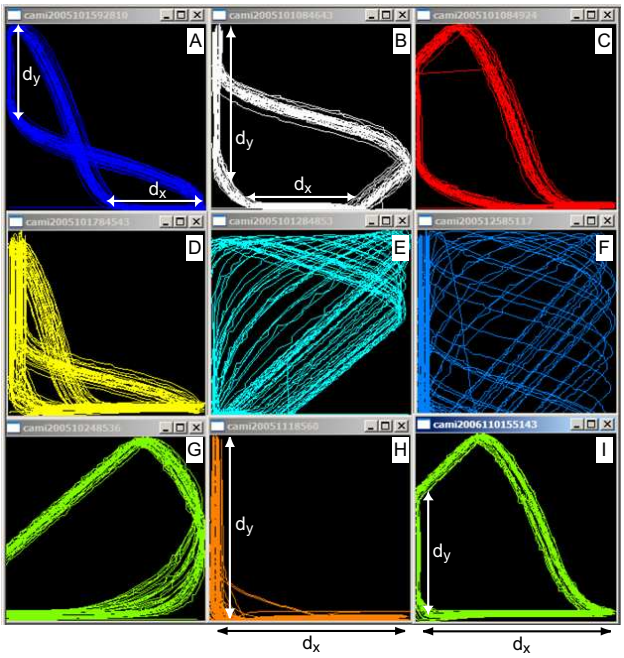

Figure 3. Left arm (x) versus right arm (y) position

of the two visualized is non-zero. Looking at exercise H, we see this sum is quite large. Although his is a very regular exercise, it is a less efficient one as compared to the other patterns, where both arms work together. For example, in exercises B and I the sum $d_x + d_y$ is, while not optimal, still smaller than in exercise H. Exercise A is a clear outlier too. Here, both arms work in almost perfect symmetry, and the sum of the 'passive distances' $d_x + d_y$ is minimal. This was actually recognized by the trainer as a high-quality exercise.

A limitation of the visualization in Fig. 3 is that the exercise evolution in time is not visible. We solved this by coloring the graph using a blue-to-red (rainbow) colormap where blue denotes $t = 0$ (exercise start) and red $t = t_{max}$ (exercise end). Figure 4 shows this method when visualizing the right arm force (x) versus left arm force (y) for four different exercises. The first thing we see are the exercise graph patterns. Exercises B and C show a L-shaped pattern, which indicates that there is force applied only to one arm at a time, similar to the position pattern in Fig. 3 H. Exercise D shows a square-shaped pattern. This indicates a 'four-phase' motion cycle, where both arms work together, albeit in clearly delimited phases. Exercise A is a mix between the L-shaped and square-shaped pattern. The color also conveys important information. In exercises A and D, which are the less regular ones, we see that the graph lines which tend to deviate from the regular pattern core, are blue. This means the athlete started the exercise in a wrong (suboptimal) state, but eventually entered in the rhythm. The less blue lines in a graph, the quicker has the athlete reached the regular, high-performance pattern.

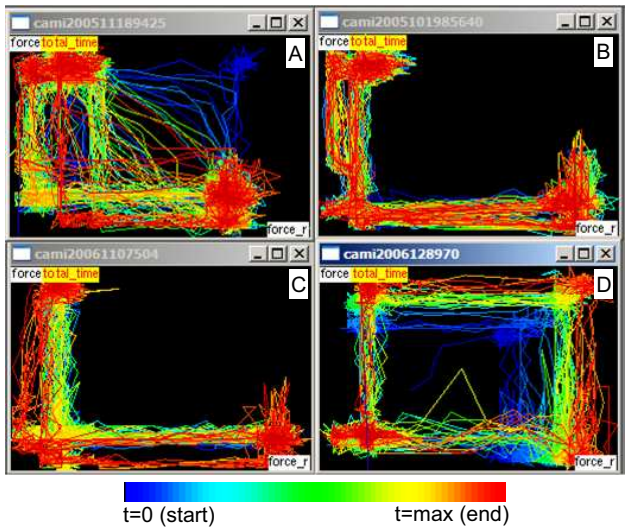

Figure 4. Right arm force (x) versus left arm force (y). Color indicates time.

3.2 Regularity visualization

After seeing the previous visualizations, one requirement of the trainers was to visualize the *average pattern*, and deviation from it, of a given exercise. We do this as follows. Consider any of the measured signals as a function of time, *i.e.* $s_i = s_i(t)$. For each such signal, the measuring device also records the time length, or period, of each repetition. This signal, called $\tau_i(t)$, has a typical sawtooth aspect, as shown in Fig. 5 a. Let us call T_k the k-th period, or pulse, of $\tau_i(t)$. Detecting these periods is trivial once we have τ_i. We can now split the actual measured signal s_i into a set of signals $\sigma_i^k : T_k$, one for each detected period T_k, *i.e.* one per exercise repetition. From now on, we shall use the split signals σ_i^k instead of the complete signal s_i, as these will allow us to capture the repetition regularity.

Given the set of K split-signals σ_i^k, we now compute the normalized *average* signal $s_i^{avg} : [0, 1]$ as

$$s_i^{avg}(t) = \frac{1}{K} \sum_{k=1}^{K} \sigma_i^k(t \cdot T_k) \qquad (1)$$

That is, we average the signals σ_i^k normalized over the time domain $[0, 1]$. Similarly, we compute a normalized *standard deviation* signal $s_i^{stdev} : [0, 1]$. Given a two-signal plot of some s_i versus s_j, we compute s_i^{avg}, s_i^{stdev}, s_j^{avg}, s_j^{stdev}, and visualize them using a new method, called the *distribution plot*, as follows. We plot the average signal s_i^{avg} versus s_j^{avg}, just like the two-signal plot. Figure 5 b shows this average signal drawn with white line for the position-versus-force plot in Fig. 5 a. Next, we like to show also the deviation from this average at every moment in time. For this, we draw a band on both sides of the average plot curve whose width is given by the projection of the vector $(s_i^{stdev}, s_j^{stdev})$ along the normal of the average

plot curve. This gives a band which is thick where the standard deviations are high *and* normal to the curve direction, and thin otherwise. Intuitively, this band conveys the 'local tightness' of the repetitive signals. Finally, we color the band just as for the two-signal plot, *i.e.* by showing the value of a third signal. We fade the intensity from maximal at the band's center to zero (black) at the periphery. This creates a nice smooth effect which suggests the decreasing density of signals as the standard deviation increases.

Figure 5 c illustrates the distribution plot for the position-versus-force plot in Fig. 5 b. The color indicates the value of the normalized time t in Equation 1, *i.e.* the position along a repetition cycle, using the same blue-to-red colormap as in Fig. 4. This image is clearly easier to interpret than the original two-signal plot (Fig. 5 a), and achieves exactly the desired goal of the trainers to see the average exercise pattern (and its deviation) in time.

Yet, the distribution plot can be quite noisy in cases of poorly coordinated repetitions of a given exercise. We provided also a smoothed option to this plot, by applying a Laplacian filter on both the average and deviation signals (s^{avg} and s^{stdev}) on both axes, prior to the visualization. Figure 5 e shows the smoothed result for the noisy force (left)-versus force (right) distribution plot in Figure 5 d. The smoothed plot is useful when we are interested only in comparing global exercise patterns and want to filter out small-scale deviations.

3.3 Clustering visualization

The previous visualizations answer well questions that target the repetitions of a single signal or comparing two signals. In this section, we present a way to compare a whole set containing hundreds of signals. This is useful to find groups of exercises which are similar from the perspective of a given signal s_i, *e.g.* force, position, or velocity. Since the comparison signal will be fixed, in the following we shall use subscript indices to denote the exercise in a given exercise set. If we compare exercises from the perspective of the force signal, then s_i will denote the force signal of the i^{th} exercise from the total set of E exercises.

For two exercises i and j, we can define the distance $d(i,j) : [1,E] \times [1,E] \rightarrow \mathbb{R}_+$ as

$$d(i,j) = \int_{t=0}^{1} |s_i^{avg}(t) - s_j^{avg}(t)|dt \qquad (2)$$

This is the distance between the exercises' average signals. The closer two exercises are, from the perspective of their average signals, the smaller this distance is. We have experimented also with different other distance metrics, such as a combination of the distance between averages and distance between standard deviations, the simple distance $\int_0^1 |s_i(t) - s_j(t)|dt$ and the minimal distance $\min_{t\in[0,1]} |s_i(t) - s_j(t)|$. The average distance has given the most robust (and predictable) results.

We can visualize the distance using a matrix plot. For this, we compute the similarity matrix $M = \{m_{ij}\}_{i,j\in[1,E]}$, where $m_{ij} = d(i,j)$. Next, we draw the matrix where every cell m_{ij} is colored using a blue-to-red colormap as function of its distance value. Figure 6

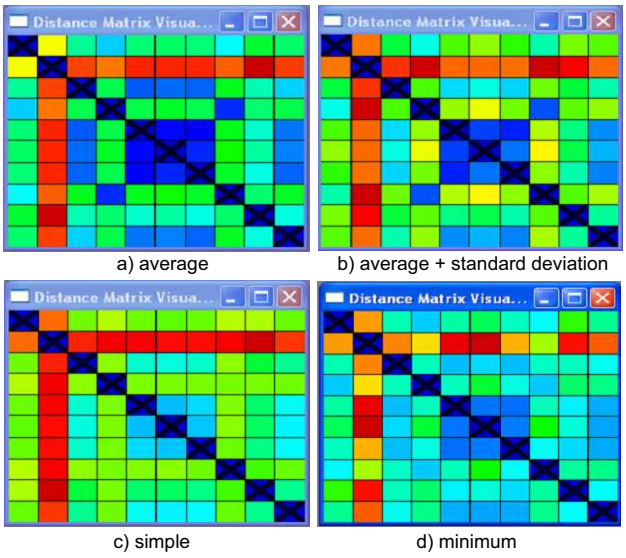

Figure 6. Distance matrix visualizations for different distance metrics. Blue indicates similar exercises, red indicated different ones.

shows four matrix visualizations for the above-mentioned distance metrics. Each row and column corresponds to an exercise, and each cell shows the similarity of an exercise pair. This visualization is the starting point for our main goal: Showing the similarity of groups of exercises.

Given the distance matrix, we now construct a hierarchical clustering of the exercises. The clustering algorithm is bottom-up agglomerative with full linkage [3], and works as follows. First, we create a list of clusters C_i, one cluster for each exercise $i \in [1,E]$. We next pick the two clusters C_i and C_j from the list such that $d(C_i,C_j)$ is minimal, remove them from the cluster list, and replace them with a cluster containing both exercises i and j. We repeat the process until a single cluster is left in the list which contains all exercises. The distance $d(C_i,C_j)$ between two clusters is computed as the minimal distance between any pairs of exercises in the two cluster (this is the definition of full linkage), *i.e.*

$$d(C_i,C_j) = \min_{e_i\in C_i, e_j\in C_j} d(e_i,e_j) \qquad (3)$$

The clustering produces a tree having the individual exercises as leaves and groups of similar exercises as nodes. Given this tree, one task we want to support is finding the n most similar exercise groups, where n is given by the user. To do this, we simply select n largest nodes in the cluster tree which have disjoint subtrees. This gives us the n largest, *i.e.* most representative, clusters. Finally, we render all cells in the distance matrix corresponding to the exercises in a given cluster with a distinct color. The matrix cells which are not corresponding to pairs of exercises

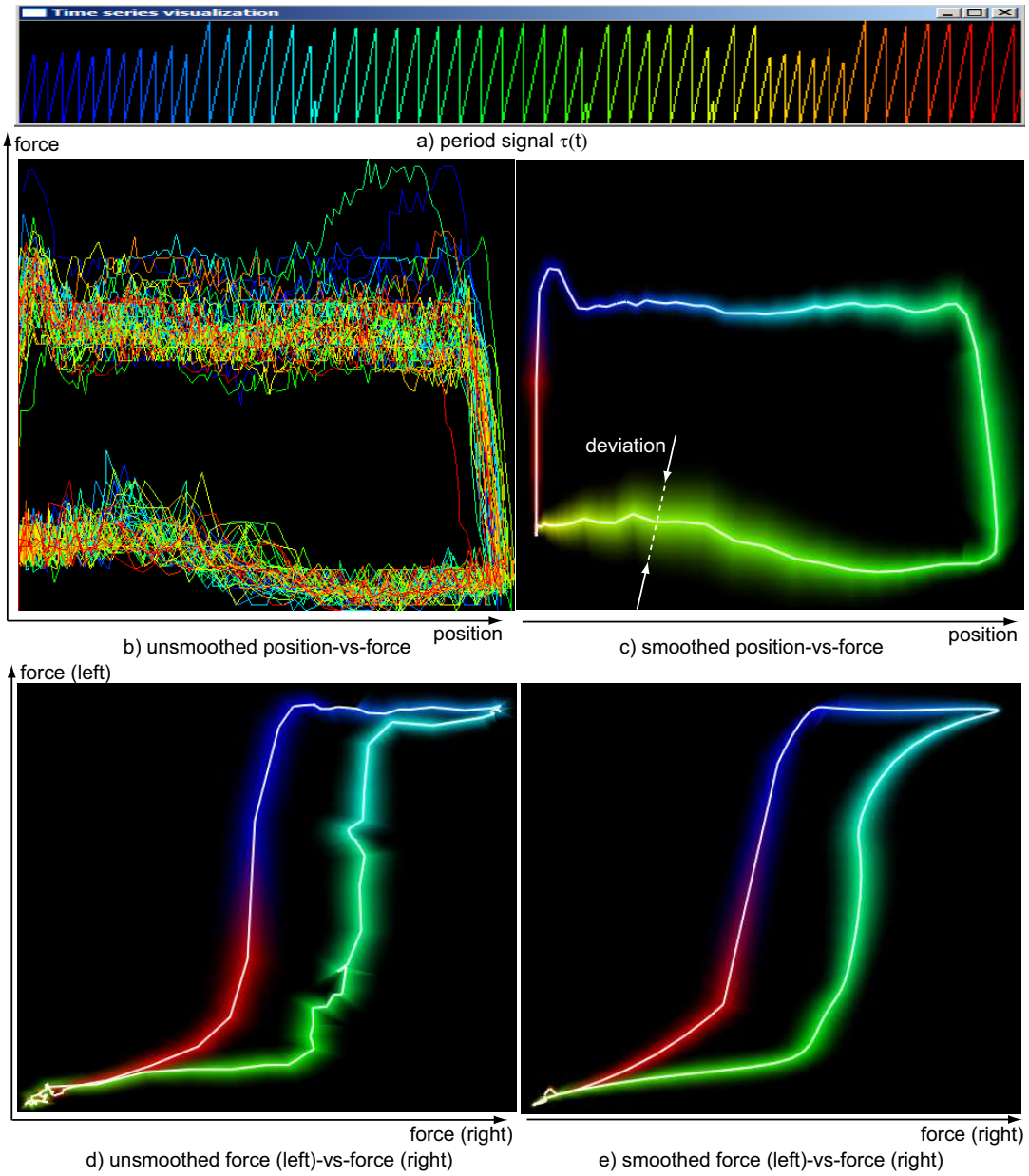

Figure 5. Visualizing signal regularity

in the same cluster are rendered in gray, with a luminance value indicating their distance.

Figure 7 illustrates this technique. Here, we clustered around 100 exercises. The similarity function uses the force signal, and the similarity metric is the average distance (Equation 2). We picked four large clusters (A,B,C) containing 43, 9, and 16 exercises respectively. The remaining 32 exercises are located in smaller clusters which are not visualized. The large clusters are indicated in the figure and drawn in three distinct colors. We also implemented a 'details window' which shows information for the cluster under the mouse position, such as size (number of exercises), diameter, and maximal error.

Although the clustering method is not forced to group *adjacent* rows and columns in the same cluster, we easily see that cluster A (largest one) is quite compact, occupying the upper-left matrix quarter. Clusters B and C (the smaller ones) are also relatively compact. This intrigued us and made us have a deeper look at the data. We then discovered that the *order* of the rows and columns in the matrix is temporal, *i.e.* the exercises are listed in increasing order of their execution date over roughly two years. Using the details window, we then discovered that the exercises in the large cluster (A) have all taken place in the year 2005. The pink cluster (C) corresponds to the first and last part of 2006, while the smallest, yellow, cluster (C) corresponds to exercises done in mid-2006. This suggests that the training was, at least regarding the athletes' force, more regular in

2005 than in 2006. Moreover, we see a fragmentation of the clusters around the moment corresponding to the center of cluster C, *i.e.* August 2006. This indicates a high variability of exercises in that period. Interestingly, this finding was confirmed by the trainers who remarked that the athletes did undergo some major training schedule changes around that moment.

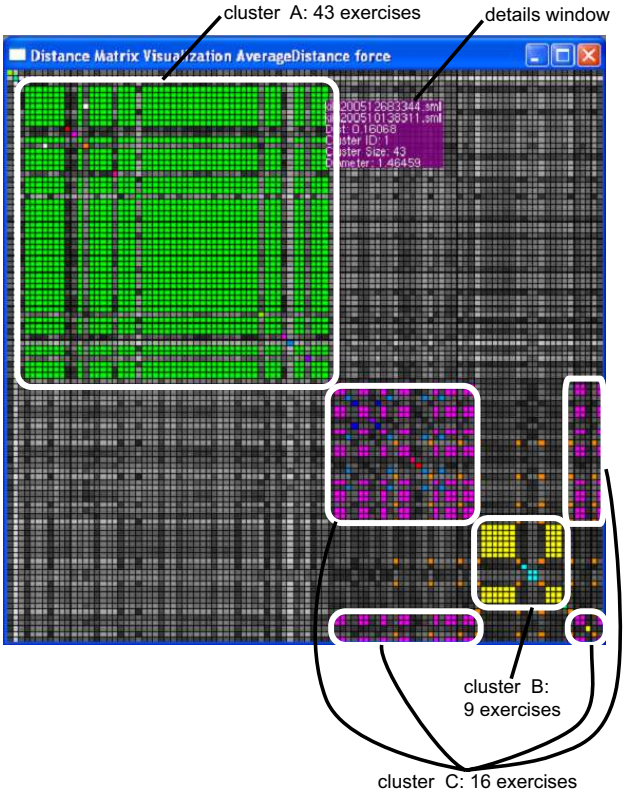

Figure 7. Hierarchical cluster visualization showing four large clusters containing similar exercises.

4 Discussion

We tested the three visualizations discussed here, *i.e.* the two-signal plot, the distribution plot, and the hierarchical matrix plot, on several hundreds of exercise datasets, each containing up to 16 different measured signals, for a couple of athletes. We quickly discovered, after some discussions with the trainers, that the two-signal plot is only useful when augmented with color showing time (Fig. 4). Also, the distribution plot design (Fig. 5) was fully derived from discussions with the trainers, to satisfy the need for depicting the regularity of an exercise.

Our entire visualization software is implemented in C++ under Windows, using OpenGL for graphics [5] and wxWidgets for the user interface [6]. The most computation-expensive part is the distance matrix calculation and the full-linkage clustering, due to the costly distance integrals (Equation 2). A single integral takes tens of thousands of sample points, and there are E^2 integrals to be computed in the worst case for a dataset of E exercises, where E can be several hundred. To optimize this computation, which takes several hours on a modern PC, we added a file cache mechanism that saves and reuses already computed distances for a given exercise set.

As expected, a careful design of the user interface was very important to make our tool accepted, as the user group (athlete trainers) are not typical savvy computer users and required simple and intuitive interfaces. In this respect, some improvements on the intuitiveness of the hierarchical matrix plot were suggested.

5 Conclusions

We have presented three correlated visualizations for analyzing large multivariate time series originating from measuring athlete performance. The two-signal plot is a simple but effective tool to measure the effectiveness, and detect the type, of exercise patterns. The distribution plot simplifies the two-signal plot by visualizing the average and deviation with optional smoothing. The clustered matrix view emphasizes groups of similar exercises by using a hierarchical clustering technique based on inter-signal distances. Overall, the strongest confirmation we received for our work was in the interest of the trainers and sports science experts to use our tool in practice, and the fact that they validated our findings by their direct experience.

We next plan to improve the above techniques by including several signals in the exercise distance computation instead of a single one; visualize additional data, such as time of exercise and type of training; and finally compare the results of different athletes to each other.

References

[1] A. Baca, L. Katz, J. Perl, and O. Spaniol. Computer science in sport. In *Proc. of the Dagtuhl Seminar 06381*. available at http://drops.dagstuhl.de/opus/volltexte/2006/833, 2006.

[2] DonnaMaria, Inc. The *MOVEnD* and *DIGInS* data acquisition product line, 2007. available at `www.donnamaria.ro/move.php`.

[3] A. K. Jain and M. N. Murty. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[4] S. P. R. Jerkins. *Sport Science Handbook*. Multi-Science Publishing Ltd., 2005.

[5] M. Woo, J. Neider, T. Davis, and D. Shreiner. *OpenGL Programming Guide, 3rd edition*. Addison-Wesley, 2001.

[6] wxWidgets. The *wxWidgets* user interface library, 2007. `www.wxwidgets.org`.