

Visual mining of time series using a tubular visualization

Fatma Bouali · Sébastien Devaux · Gilles Venturini

Published online: 5 December 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract In this paper, we study the visual mining of time series, and we contribute to the study and evaluation of 3D tubular visualizations. We describe the state of the art in the visual mining of time-dependent data, and we concentrate on visualizations that use a tubular shape to represent data. After analyzing the motivations for studying such a representation, we present an extended tubular visualization. We propose new visual encodings of the time and data, new interactions for knowledge discovery, and the use of rearrangement clustering. We show how this visualization can be used in several real-world domains and that it can address large datasets. We present a comparative user study. We conclude with the advantages and the drawbacks of our method (especially the tubular shape).

Keywords Visual data mining · Time series · 3D interactive visualizations · User evaluation

1 Introduction

Time is involved in many real-world datasets, and many methods are devoted to temporal data mining [8]. Among

the methods that can be used for knowledge discovery in time-dependent data, we focus on those that are user-centered and that use interactive visualizations. These methods belong to the more general fields of visual data mining (VDM) [13,26,39,47] and visual analytics (VA) [25]. Time is such a specific dimension that many visual and interactive approaches are devoted for exploring and analyzing temporal data [1,2,35].

Several authors have proposed using a tubular shape as a visual metaphor, initially as a focus+context technique [34] and then as a means to visually represent the evolution of time series [5,40]. The principles of such a representation consist of using the axis of a tube to represent the time and the sides of the tube to represent the evolution of the data attributes. As underlined in [34] (see the next section for further details), this visual metaphor has several interesting properties. It is argued that it is a “natural” metaphor that can be easily understood by the users, that is, it presents an overview of the data as well as details (a focus+context technique) and that it can represent a large quantity of data.

In these studies, the main point was to introduce the tubular visualization and its focus+context property. The visualizations were mostly 2D and had static views [34,40]. No navigation or interactions were proposed within the visualizations, and the report of user experience and evaluation was limited. Thus, there are still many unanswered questions about the use of a tube as a visual metaphor: Do the users easily accept such a metaphor? Can it really address large datasets? Can these visualizations be further extended with navigation and interactions with the aim of mining time series? Does user performance increase (or decrease) with such a representation and in comparison with a more standard one?

In this paper, our aim was to contribute to the study of such tubular visualizations and to better highlight their strengths

F. Bouali
University of Lille2, IUT, 25-27 Rue du Maréchal Foch,
59100 Roubaix, France
e-mail: fatma.bouali@univ-lille2.fr

S. Devaux
Airbus Defence and Space-Space Systems,
TSEOC12 Simulation, Les Mureaux, France
e-mail: sebastien.devaux@astrium.eads.net

G. Venturini (✉)
Computer Science Laboratory, University François-Rabelais of Tours,
64 Avenue Jean Portalis, 37200 Tours, France
e-mail: venturini@univ-tours.fr

and weaknesses. For this purpose, we used the initial work as a basis and starting point, and we tried to go beyond these proposals. We extended the tubular visualization in several ways (new visual elements, 3D navigation, interactions) with the aim of improving its ability to solve time series mining tasks. Then, with this extended visualization, we studied how it can be accepted and used by users from different domains and how it can represent large datasets. To further analyze this visualization, we performed a user study to compare this approach to a more standard 2D matrix representation.

The remainder of this paper is organized as follows. In Sect. 2, we describe the visual approaches that are devoted to temporal data, especially methods that process large datasets and those that use a tubular representation. Section 3 focuses on the extensions we made to the tubular visualization, which yield to the definition of DataTube2. In Sect. 4, we present experimental results on real or large data, and in Sect. 5, we detail a comparative user study. Section 6 concludes this paper and presents future perspectives.

2 State of the art and principles of tubular visualizations

2.1 Visual mining of time-dependent data

The visual mining of time-dependent data has a long history [33], and several surveys can be found such as [1, 2, 35]. In this paper, we focus our survey on multivariate time series. We consider the common case of n numerical attributes whose values have been recorded over k time steps. Such attributes can describe the evolution of a given phenomenon, such as market stocks, the consumptions of products, hits on Web pages or glucose concentrations in patients' blood. We do not address tree-structured data (such as in [42]), geo-localized data (see [4], for instance), or the evolution of texts over time (see, for instance, [14]).

Many visualizations have been proposed for the visual exploration of time series, such as diagrams [29, 44], spirals [12, 43, 45], and metaphors (calendars [15, 46], pencils [16]). These visualizations are interesting, but are generally limited to small datasets (i.e., $n \times k < 75,000$, according to the values presented in those references). Here, as will be shown in the following sections, using the tubular shape allows us to address larger datasets, so we focused our survey on methods that can display the largest amounts of temporal data. In [19], a clustering step is performed with a Kohonen map on 2,665 curves of 144 values each to visualize 100 clusters (each cluster is visualized as a "mean" curve with 144 values). Thus, the resulting visualization consists of $n = 100$ curves over $k = 144$ values. Another example is the "Recursive Pattern" [24], in which $n \times k = 530,000$ values are displayed with $n = 100$, or "Circle segments" [7] (with $n = 50$). These approaches belong to pixel-oriented methods. More

recently, a visualization called CloudLines was proposed in [28]. CloudLines is based on a multi-resolution technique in which more space is devoted to recent time steps than to older time steps for which data values are highly aggregated. This compact display can represent up to $n \times k = 117,000$ events. In [18], a visualization is proposed to analyze frequent patterns in large time series ($k = 85,035$), but with a limited number of series ($n < 5$). Another method, called Line Graph Explorer, can display large datasets [27]. This method was applied, for instance, to climate data ($n = 324$ cities and $k = 3,650$, which amounts to $n \times k = 1,182,600$). This approach can also reorder the attributes to place attributes with similar temporal behavior next to each other. LiveRAC [32] uses a condensed display of many time series. These are, as far as we know, the visual and interactive methods that can represent the largest amount of temporal data. Several of these methods do not visualize the complete dataset, but rather aggregated values, so the real number of displayed data might be smaller than what we mention in this section.

In general, the interactions are limited to the visualization only: the user can zoom in or out, possibly with a semantic zoom that displays several levels of detail. The user can navigate within the data. Interactions are rarely proposed to help the user in mining the data. In [18], the user is assisted in finding patterns, and in Line Graph Explorer, the time series can be reordered according to their similarities. Indeed, other systems, such as [38], propose several interactions (such as the search for patterns), but the display in this system is limited to a small number of time series (approximately ten). In addition, user evaluations are rare in the context of time series visual mining [3, 32].

In the visualization we propose, we address datasets of similar size (or larger), and we include several interactions to help the user during the mining process. We also present a user study to evaluate our approach.

2.2 Tubular approaches

The idea of a 3D axial representation of the time around which the data attributes are wrapped can be found in several visualizations. For instance, in the "Kiviat tube" [17], a star graph [23] is defined for each time step of the time axis. Each star graph represents the values of the n attributes for the considered time step. Many such graphs are stacked along the time axis. This method was applied to the visualization of the load of $n = 64$ processors. However, this stacked representation seems to create many occlusions.

The principles of a 3D tube were initially proposed in [34]. The 3D tube was used as a focus+context technique [30]. This property was achieved with the 3D perspective that results from the 2D projection of the 3D tube. It was illustrated in several domains, including the display of time series. The "attributes \times time steps" matrix was represented with gray

levels, and both sides of this matrix were folded over each other to form a tube. This tube had a square shape/section (not a circular one, as in other works). The axis of this tube represented the time. The point of view on the tube was fixed (tip of the tube) and no navigation or interactions were possible. No specific evaluations or user feedback were studied.

Another tubular representation, called DataTube, was proposed in [5]. This visualization can be seen as a possible evolution of Recursive Pattern, but with the use of the 3D space. DataTube was applied to the visualization of market stocks (with $n = 50$ and $k \leq 100$). It included a technique to reorganize the attributes around the tube. DataTube was not further developed or evaluated.

To conclude the state of the art, we mention a recent visualization [40] in which a tubular shape was used as a focus+context technique, again with a fixed user point of view. This visualization was intended to represent events over time rather than multivariate time series. Hence, only two attributes were represented in the tube, i.e., the time and another symbolic attribute whose values are encoded as a position on the tube. 3D widgets with varying colors were used to represent several types of events. The visualization had no interactions. Case studies with simulated data were presented, but with no real user feedback or evaluation. This work confirmed the interest of such a tubular representation for mining temporal data in real-world domains.

2.3 Motivations for further study of a tubular visualization

These authors (especially [34]) explained well the motivations for studying such a tubular representation. We list them below and add our own. These authors argue that the tube is a “natural” metaphor that can be easily understood by the users. This seems plausible to us because many objects in the real world have a tubular shape. This shape was used in several domains to convey information, such as in video games where the player is located in a tube.¹ An estate agency even used a 3D tube for its hiring advertising,² where the axis of the tube represented the time and the evolution of one’s career. A tubular shape was also used by a data designer (see <http://www.benwillers.com>). From these examples, and according to the initial papers, we can expect that the tubular shape will be easily understood by the users. This point was not especially evaluated in the previous work, so we intend to better quantify it.

In the initial work, the main reason for introducing a tubular metaphor was to use it as a focus+context technique. Usually, the 3D perspective is considered as a negative effect in visualizations because the size of objects depends on their

distance from the user. If the depth is not properly perceived, then a large object that is located far from the user will appear smaller than a small object located next to the user. In a tubular representation, this weakness is turned into a strength: the perspective is used to assign more or less display space to the data. Data located next to the user are given more space than data that are located far away. Consequently, a complete overview of the dataset is available as well as details: the sides of the tube represent the details, and the end of the tube provides an overview of the future (or the past). In the existing work, the user point of view was mainly fixed, i.e., no navigation was provided. Consequently, the user had no possibility to adjust the focus to a desired part of the dataset. However, adjusting the focus area is a typical interaction in focus+context techniques, such as the perspective wall [30]. In the presented work, we wish to improve this point by enabling the user to navigate within the tube and, thus, to adjust the point of focus as well as the context area.

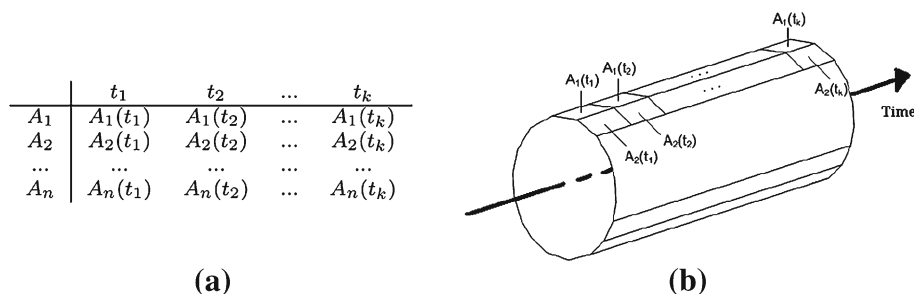
From the 3D point of view, the tubular shape avoids other drawbacks of 3D representations than the perspective effect. When the user moves inside a tube, no occlusions occur: no data are hiding another one. Furthermore, we will show in this paper that the 3D navigation can be simplified thanks to the tube shape and that it can be limited to two degrees of freedom. So a simple navigation mode can be implemented that is easy to understand for non-experienced users.

From the 2D point of view, the differences between a tubular shape and a 2D matrix representation (i.e., an unfolded tube) can be discussed. In a 2D matrix representation, the rows can represent the series, and the columns can represent the time steps. Such a representation is space efficient (following Ben Shneiderman’s statement, “a pixel is a terrible thing to waste” [10]) because it can use the whole screen thanks to its rectangular shape. As discussed in [34], the same property can be obtained with a tubular representation (i.e., only limited part of the screen are wasted, as shown in the figures presented in this paper). In addition, in a tubular representation, the circular layout of the attributes is more “homogeneous” than a linear ordering of a matrix (as in Line Graph Explorer). In a linear representation, the first line (i.e., the first attribute) on top of the visualization is located far away from the last line at the bottom of the visualization (i.e., the last attribute). If the attributes have been reorganized, then the comparison between the first and the last attributes might be more difficult because those attributes are located at the two farthest possible locations in the representation. In a tube, the layout of the attributes around the axis is such that there is no first or last line. Furthermore, distortion techniques such as fisheye views also implement a focus+context mechanism and could be applied to 2D matrix visualization. However, it seems that the evaluation of distortion techniques are mixed [20,36]. So we believe that it could be interesting to develop

¹ Among the Android game applications, more than 30 games can be found in which a 3D tube is used.

² <http://emploi.nicematin.com/fckeditor/image/image-logia>.

Fig. 1 a The data matrix that will be visualized in DataTube2. It contains n attributes A_1, \dots, A_n whose values were recorded over k time steps t_1, \dots, t_k . **b** The temporal tube as described in DataTube [5]. Each line of the matrix is mapped to a *line* in the tubular representation



alternative approaches that would be based on different techniques, here a 3D representation.

Finally, it is argued in the existing work that a tubular shape can represent large datasets because the length of the tube can be infinite, which is conceptually true. However, in practice the limited resolution of the screen will impose a limit, which was not tested in previous work. In this paper, and following our initial work in which we presented a first application of a tubular representation to web data [41], we would like to further study such a representation and its properties, to gain more insight into the tubular visualizations and to provide answers to the questions we raised.

3 DataTube2

3.1 Data representation and pre-processing

The data pre-processing step we describe in this section is not specific to the tubular representation, but it represents an important step in the preparation of the data before the visualization. This pre-processing step has been integrated in our tool in a 2D interface. With this interface, the user can easily set up the parameters that are used to prepare the data (as well as the other parameters of the visualization discussed in the following sections).

We denote by A_1, \dots, A_n the n numerical attributes that describe the data. We assume that these attributes values were recorded, possibly in an asynchronous way, over a $[0, T]$ time interval. A record is thus represented as a triple (*Attribute, time stamp, value*), and many such records represent the data to be visualized and explored. Thus, the first step is to convert these asynchronous data into a synchronous representation (i.e., an “attributes \times time steps” matrix). For this purpose, the user defines a time interval ΔT (in seconds, minutes, hours, days, etc.) and an aggregation operator (sum, mean, max, min) to aggregate the asynchronous attribute values (see also the discussion in [9]). The value of ΔT defines k time intervals (or time steps) that are such that $T = k \times \Delta T$. In the following, we denote these time steps as t_1, \dots, t_k . For a given time step t_j (i.e., $t_j = [(j - 1) \times \Delta T, j \times \Delta T]$), and for a given attribute A_i , all A_i records in the input data are

aggregated with the operator that was selected by the user. If no event was recorded for A_i during t_j , then the value in the matrix is labeled as “missing”. At the end of this pre-processing step, a data matrix is created as shown in Fig. 1a. This matrix has n lines (one for each of the n attributes) and k columns (one for each time step). Thus, the input data of our visualization are represented by an $n \times k$ matrix and we denote $A_i(t_j)$ as the value of attribute A_i at time step t_j . The values of an attribute A_i are then normalized on a $[0, 1]$ scale. This normalization can be achieved in several ways: we can use the minimum and maximum values of A_i , the global minimum and maximum values that are found in the matrix, or user given values.

3.2 Initial tubular visualization

The initial tubular visualization (as initially described in [5]) is a representation of this matrix with a time tube as shown in Fig. 1. In this visualization, each value $A_i(t_j)$ is represented by a rectangular facet. A time step t_j is thus encoded by a set of facets forming a ring on the tube. The evolution of an attribute $A_i(t_j)$ is represented by a line that is parallel to the tube axis. A gray level is assigned to each facet depending on the value of $A_i(t_j)$. We selected this circular tube representation [5, 40] rather than the square section tube [34]. In a square section (i.e., with 4 faces), series that are located in the corners are given less space than series located in the middle of the faces of the square tube. Thus, using a circular tube, the parts of the screen that are assigned to the attribute are more homogeneous than in the square tube.

3.3 The Datatube2 visualization

To the initial visualization, we added several visual features to make it more operational for time series analysis, namely a time axis with the possibility to add annotations, and the visualization of missing values.

We wanted to help the user better visually identify the time in the 3D tube because no time axis was proposed in the other tubular visualizations. For this purpose, we propose two possibilities. The first is to represent a ring on the tube to clearly identify a given time step (see Fig. 2a). This

Fig. 2 The extensions we made to the tubular visualization. In **a**, we added a ring and other types of selections to help the user evaluate the time and duration of events. In **b**, we added an optional time axis with labels that represent the time steps

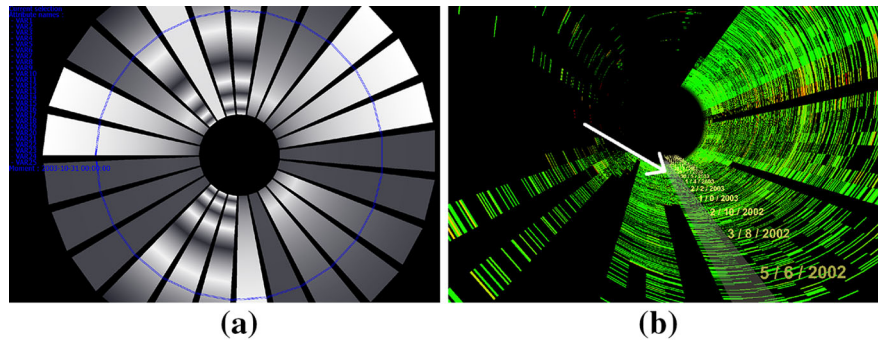
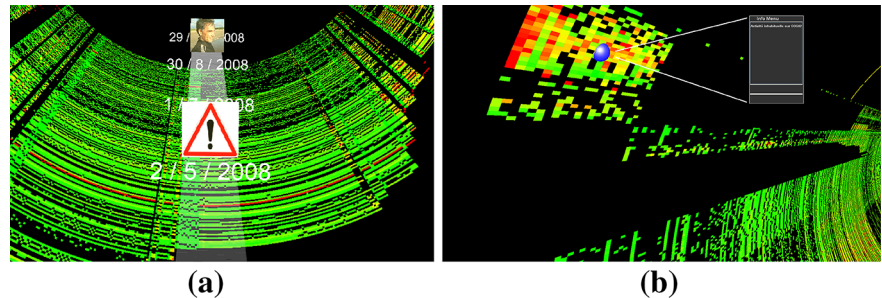


Fig. 3 Examples of annotations that were dynamically added. In **a**, the annotations highlight specific time steps, and in **b** an annotation underlines an event (one specific value of an attribute)



ring can be moved forward or backward, step by step. The corresponding time value is represented as a label on the left part of the screen. The second extension is a time axis that can be displayed or hidden (see Fig. 2b). This axis is composed of slabs with text labels that indicate the time. Such slabs are transparent, allowing the user to see the data located underneath.

The user can add annotations to the time axis or to a facet. Such an annotation is visually represented as an image for the time axis (see Fig. 3a) and as a small sphere for a facet (see Fig. 3b). The image can be used to represent a specific event. This can help the user analyze how the evolution of the attributes was affected by this event (for instance, if the data represented stocks, an influential event can be the speech of a politician, and an image of that politician can be represented in the tube at the time the speech was delivered). The user can associate to an annotation a text that can be displayed, a sound that can be played, or even a Web link (URL) that is opened in a Web browser. In this way, the user can directly store ideas or discover knowledge in the visualization. These annotations can be used to share knowledge with other experts or to perform an interactive presentation of the data to a panel of users (customers, managers).

Another practical issue we addressed in DataTube2 is the problem of missing values. Such values might occur because of the asynchronous to synchronous data conversion (see Sect. 3.1): for a given time step t_j and for a given attribute A_i , there might be no record in the input data. This case requires specific processing and an adapted visual ren-

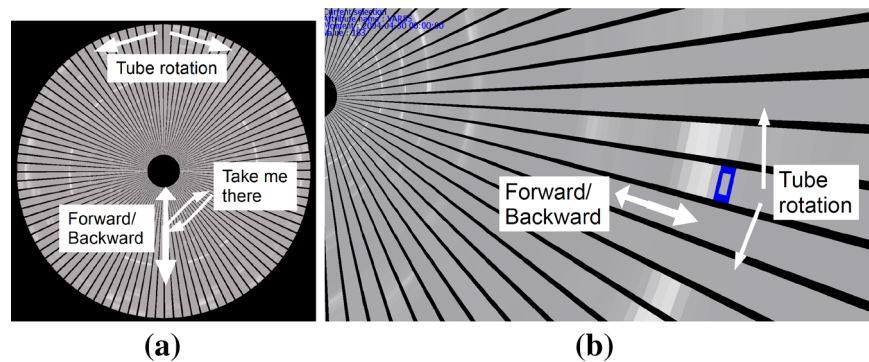
dering. Missing values might also be due to other reasons: for instance, if the attributes represent the values of sensors, then the absence of a signal might constitute important information for the user. To represent such missing values, we choose a user-determined color (black, in this paper, which is the same color as the background color). With such a representation, the missing values appear as “holes” in the tube. The user can then easily detect and analyze them.

Finally, the color encoding of the facets can be customized with a simple transfer function. This function turns each data value $A_i(t_j)$ into a color. It can normalize the values according to different schemes (e.g., local or global minimum and maximum, means and deviations, etc). One lower and one upper threshold can also be defined to restrict the scale of values. In the short term, we also wish to include preset color scales (such as in colorbrewer2.org).

3.4 From simple to advanced 3D navigation modes

Changing the area of focus is necessary for the data exploration. In the previous tubular visualizations, this was generally not possible; the area of focus was fixed. In a 3D tube, the focus can be changed by allowing the user navigate within the tube. With different 3D points of view, the user can adjust which part of the data will be the focus of the visualization, and which part will represent the context. However, the navigation in 3D can be difficult for inexperienced users. This is because 3D representations generally create occlusions and might require complex moves to obtain

Fig. 4 **a** Description of the simple navigation mode of DataTube2. The user can move forward/backward along the tube axis. When the user selects an area, the “take me there” move is triggered. The user can easily come back to the starting position. The user can rotate the tube. **b** The result of a “take me there” move followed by a small zoom



a desired point of view. However, in DataTube2, it was possible to define a simple navigation mode thanks to the tubular shape. If the user navigates inside the tube, then no occlusion occurs. Furthermore, the axis of the tube, which represents the time, is located in a central position from which all attributes can be observed. Therefore, we were able to define a navigation mode in which the user’s navigation is restricted and thus simplified (see Fig. 4a). The user can move forward or backward along the tube’s axis. The tube can be rotated on its axis to place any attribute on any side of the tube (e.g., top, left, right, bottom). To zoom in on a specific part of the tube, the user can double click on a facet, which triggers an automatic and progressive “take me there” move (see Fig. 4b). The user can use the mouse wheel to get closer to the tube’s side. With the forward/backward moves, or with the tube rotation, the user can analyze the context of the selected area. The user can observe what happened before and after a given temporal pattern. Once the analysis is finished, this “take me there” move can be progressively canceled, taking the user back to the starting position on the tube axis. With such a restricted mode, the moves are simple to learn because they are limited to two degrees of freedom.

For more experienced users, we added more complex 3D moves. With a right click, the user can change the orientation of the camera. The user is thus able to look and move in any directions. He can obtain a view of the future (as in the simple navigation mode), but if he turns around, he can obtain a view of the past. In the most complex navigation mode, we also integrated a 3D mouse with 6 degrees of freedom (Space Pilot) for users who might be familiar with such device.

In all navigation modes, the user is initially placed at the tip of the tube axis and faces the inside of the tube. The user thus obtains an overview of the data, highlighting, for example, major trends, groups of similar attributes or large areas of missing data (see Sect. 4). Then, the user can move along the temporal axis to observe the evolution of the attributes through time. With the automatic or manual moves, the user can zoom in on the data with a partial conservation of the

context. Once a zoom is performed on a given area, the user can explore the surroundings with the simple moves: the tube can be rotated to see the other nearby attributes, or the user can move forward/backward to see the temporal evolution of the attributes.³

3.5 Interactions in DataTube2

Interactions are important to facilitate solving of VDM tasks. In the context of time series, we concentrated on the detection of similarities in the data (similar attributes, similar patterns). To help the user to detect such similarities, we extended some functionalities of TimeSearcher2 [38], in which the user can select a visual pattern of interest and obtain similar patterns in the visual representation. Our proposal is different in that we can address a much larger number of series and the selected area can encompass several attributes (not just the values of a single attribute). More precisely, the user can select any area in the tube. Such an area can be a single facet, a time step (i.e., a ring on the tube), an attribute (i.e., a line in the tube), or any other “rectangular” area. Such selections are performed with a combination of a mouse click with several keys. To select a single facet, a left click is sufficient. With an additional key, the user selects in one click the complete time step or the complete attribute. The selection of any area of the tube is similar to the selection of a box in a matrix (as in a Spreadsheet). Such a selection is performed as follows: a click is used to select one corner (a first facet) of the area, followed by a shift-click to select the second corner (a second facet).

Each time an area is selected, the names of the selected attributes as well as the bounds of the corresponding time interval are displayed on the left side of the screen. The selection can be moved or extended with the arrow keys.

The selected area is a submatrix M_S of the data matrix. The user can trigger the search for areas in the tube that

³ For illustration, videos of DataTube2 can be found here <https://www.youtube.com/watch?v=Td2cT1a4OY0> and, with a better quality, here <http://www.vizassist.fr/DataTube2/>.

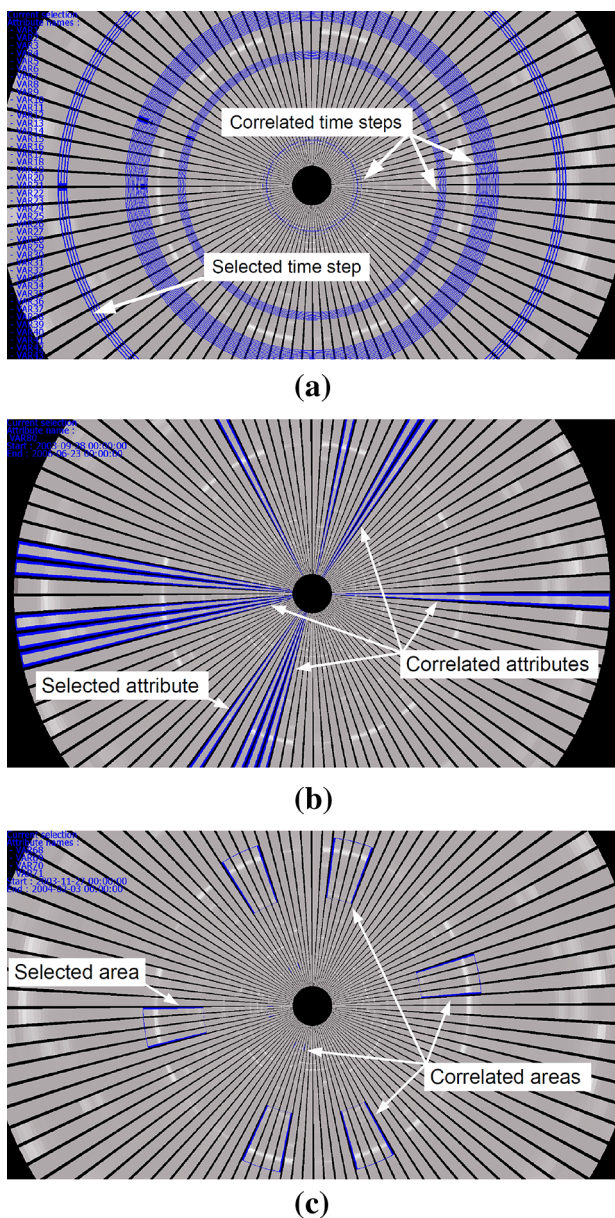


Fig. 5 Three different selections of an area with the search for correlated areas. In **a**, a time step (i.e., a ring in the tube) was selected, and the correlated time steps were highlighted (in blue). In **b**, an attribute was selected (i.e., a line in the tube), and the user triggered the search for similar attributes. In **c**, specific events were selected (i.e., a subset of the data matrix) and several correlated areas appeared. A slider is used to dynamically adjust the threshold that determines which areas are displayed

are correlated to M_S and with the same dimensions as M_S . DataTube2 can dynamically visualize the most correlated areas (see Fig. 5). For this purpose, we compute the correlation between M_S and all other submatrices in the data matrix with the same dimensions as M_S (i.e., the same number of rows and columns). Then, the user can dynamically adjust a threshold with a slider: all the submatrices whose correlation to M_S is higher than the threshold are displayed. The

correlated areas appear with a colored frame. This color is customizable. With such an interaction, the user can select a time step (i.e., a ring in the tube) and can obtain all correlated time steps (see Fig. 5a). The user can do the same with attributes (see Fig. 5b): with the selection of an attribute, the user can obtain all correlated attributes. Finally, if a specific event or behavior (several facets that form a submatrix) is observed, the user can select it and ask for similar events in the data (see Fig. 5c).

3.6 Reorganization of the attributes

To further improve this visualization for solving VDM tasks, we added another feature to help the user detect similarities in the data. The order in which the attributes are displayed around the tube is important because it might help the user to discover crucial information such as the clusters of attributes (i.e., attributes with similar/correlated behavior) or the outliers (i.e., attributes with a very specific behavior). Thus, it is important to display similar attributes next to each other in the tube (see Fig. 6). Furthermore, when the number of attributes increases, it is necessary to structure the visualization to help the user cope with large amounts of information. Initially, and without specific knowledge, DataTube2 considers the order that is given in the input file.

To find a more informative ordering of the attributes, we used a rearrangement clustering algorithm that places attributes with similar behavior next to each other. This problem is well known in information visualization [6], especially in the case of matrix visualization [11, 22]. Solving this problem requires the definition of a similarity measure (or distance) between the elements to be reordered, and the choice of an algorithm for the reorganization. For this purpose, we selected a classical and popular matrix reorganization method called bond energy algorithm (BEA) [31]. This algorithm uses the following similarity measure between two attributes:

$$ME(A_i, A_j) = \sum_{t=t_1}^{t_k} [A_i(t) * A_j(t)]$$

where ME stands for “measure of effectiveness”. The BEA heuristic works as follows: it uses a list of attributes that is denoted by L . L is initially empty. BEA starts by randomly selecting two attributes and then adds them to L . Among the remaining attributes, it then chooses the one that is the closest to the attributes already present in L . For this purpose, it considers all of the possible positions of insertion and chooses the one that maximizes ME . Once all of the attributes are inserted, the tube is reorganized according to the order given by L . The complexity of this heuristic is $O(n^2)$. A typical result of this heuristic is presented in Fig. 6.

Fig. 6 The effects of the reorganization of the attributes around the tube. **a** The initial representation with no specific ordering. **b** The BEA results, which reveal the attribute clusters (as well as one outlier)

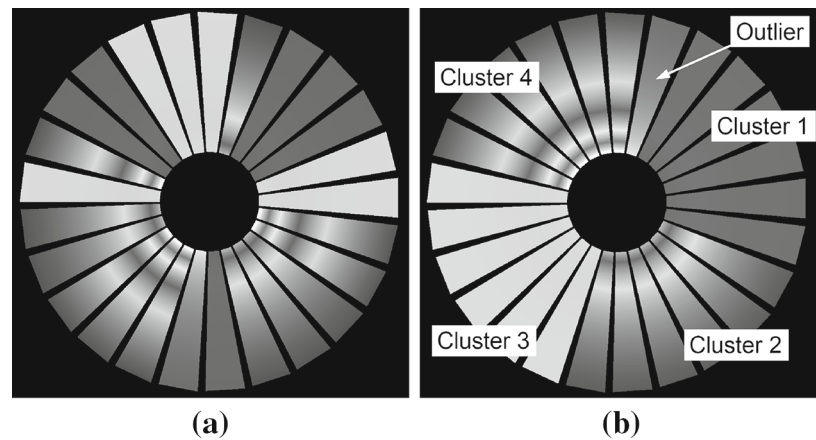


Table 1 The real-world datasets that were used to test DataTube2

Datasets	Description	No of values	n	k
POLYTECH- INIT	Web site log (documents and images)	1,447,839	9,463 elements	153 days
POLYTECH	Web site log (only documents)	563,668	1,148 pages	491 days
ANTSEARCH	Web site log (several Web sites)	66,875	107 pages	625 days
EEG	Electroencephalograms	67,136	64 electrodes	$1,049 \times 2$ ms
CONSO	Consumption of a product (anonymized)	365,000	1,000 individuals	365 days
BLOOD	Measurements from a medical device (glucose)	30,000	600 patients	Patient dependent

For each dataset, we indicate its number of attributes n and its number of time steps k

4 Properties: user acceptance and large datasets

4.1 Feedback from domain experts

The aim of these tests was to check whether the tubular shape and the general design of DataTube2 were sound and that they were potentially acceptable for users. To obtain an informal feedback, we presented DataTube2 to several experts from different domains using their own data. With users from different domains, we expected to broaden the feedback and suggestions. In an informal session, we let the experts navigate in the tube and observe their data, with our help. The users were able to use different interactions. The datasets we processed are represented in Table 1. The different domains we addressed were the following:

- Web usage mining: the three first datasets were typical examples of web site logs that were analyzed by a domain expert using DataTube2. This was the first use of DataTube2 [41]. In the tubular representation, one line corresponded to the daily activity of one web page,
- First medical domain: the EEG dataset represents an electroencephalogram that resulted from an experiment in Psychology. The activities of 64 electrodes were recorded over 1,000 time steps. The psychologists wanted to study the evolution of the electrodes' activities when

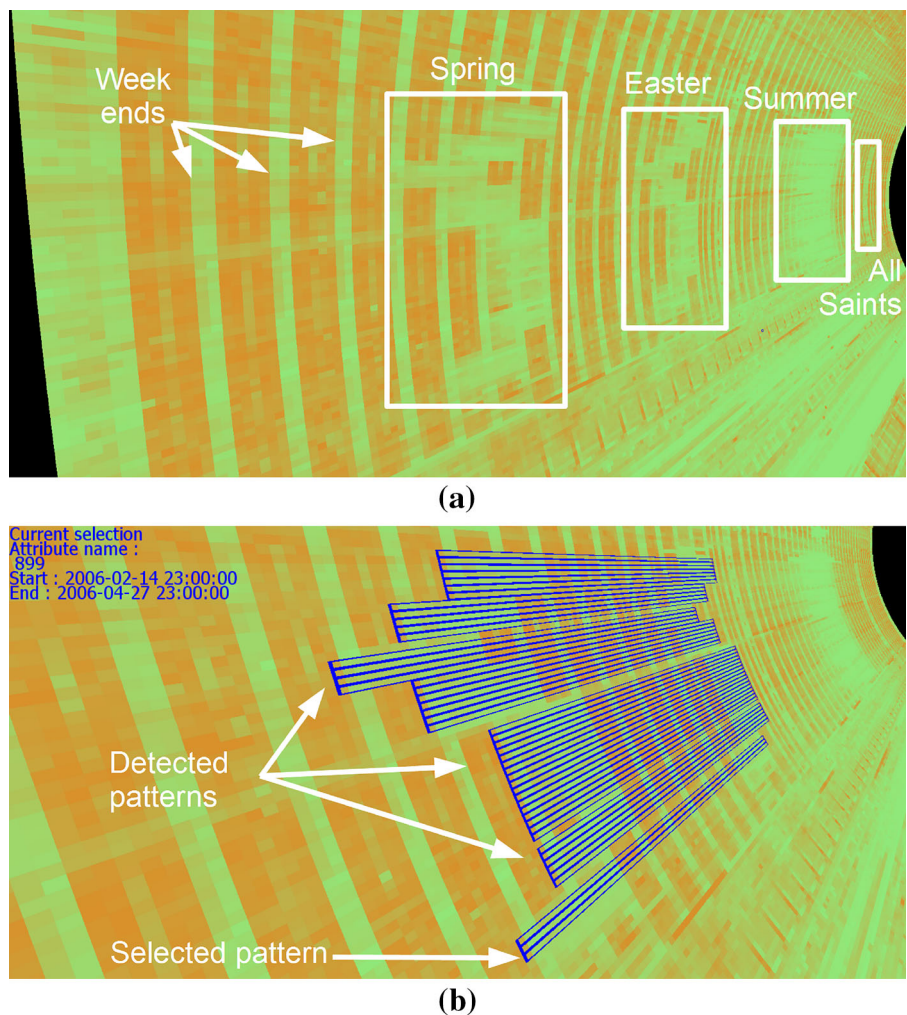
a visual stimulus was presented to a patient. For this dataset, a line in the tube represented the activity of an electrode,

- Second medical domain: the BLOOD dataset concerned the stay of 600 patients in a hospital's intensive care service (Chartres Hospital, France). The concentration of blood glucose was regularly measured for each patient. Each line in the tube represented one patient and the daily evolution of the patient's glucose concentration,
- Product consumption analysis: the CONSO dataset represented the daily consumption of a product over 1 year and for 1,000 customers. With such data, each line in DataTube2 corresponded to one consumer and his product consumption.

To gain a better idea of the effectiveness of our approach, we provide more details about the last domain (product consumption), and we present a typical scenario that corresponded to the analysis performed by the tested expert. A sample of the obtained visualizations is given in Fig. 7, and we illustrate how this user discovered several clusters of customers.

The user initially obtained an overview of the data. Thanks to the representation of missing values, he quickly made a first discovery: there were many missing values for several

Fig. 7 Typical information that was visually detected in a daily consumption dataset (1,000 customers over 365 days). In **a**, the user identified weekly patterns and a group of customers with a specific “holiday” behavior (lower consumption during all of the holidays). In **b**, with the selection of this “holiday” pattern, followed by the search for similar patterns, the user checked that other customers did not share those patterns and that this cluster was relevant



customers (that BEA clustered together). The user thought he had given us a dataset with no such values. Then, he identified several clusters of customers. One large cluster was characterized by a specific behavior that was related to the holidays. As shown in Fig. 7a, these customers used less products during the spring, Easter, summer and All Saints holidays than during the other work periods. One can observe that the spring and Easter holidays started and ended at different times in the year. With the “take me there” move, the user was able to zoom in to better observe some patterns of interest. He checked that these clients have a similar consumption for the WE (with respect to week). Using the selection/search function, the user was able to check that this “holiday” behavior occurred only among those customers (see Fig. 7b), which required an overview of the dataset. Thus, he found that this cluster was relevant.

The user also found another interesting group of customers with a regular and repeated behavior throughout the year (consumption during the week and with a pause for the weekend). The user was able to identify those customers using the

search function. For some customers, the user also noticed that the pause was occurring on Sundays and Mondays (rather than Saturdays and Sundays). He made this discovery thanks to the details given by the focus area. With the navigation, he was able to check that this pattern was repeated over time. Then, he searched for customers with similar behavior by selecting a line and searching for similar attributes. The user found that those customers were outliers (these were the only customers with such a behavior in the dataset). Other such customers were also found (e.g., with a pause on Sundays only).

The user repeated these steps several times: visually finding a pattern of interest, zooming in and observing this pattern in more detail, and selecting the pattern to search for correlated patterns. We also noticed that the user performed an interactive adjustment of the threshold that is used in the search function. After setting a first value, the user visually checked that the highlighted data were similar (according to his own criteria and knowledge) to the selected one. If too many data items were displayed (i.e., the similarity was visually too low), then the expert tightened the threshold.

4.2 Dealing with large datasets

We studied the limits of DataTube2's actual implementation with respect to the dataset size. We were able to process three large artificial datasets with $n = 800$ attributes and with $k = 1,000$, $k = 1,500$ and $k = 2,500$ time steps. Such volumes of data are similar or even larger than those handled by recent time series visualizations (see the references in Sect. 2.1). With such large datasets, the navigation was fluid because the tubular shape is simple to process by current GPUs. Thus, no decreasing of the display performances was observed. However, the interactions were longer. We measured the time for typical interactions, from the end of the user's graphical query to the redisplay of the new visualization. Such interactions were the selection of an attribute, the search for correlated attributes, the selection of a pattern (with a size of $n = 15 \times k = 30$), and the search for correlated patterns. For the $800 \times 1,000$ dataset, the interactions took < 2 s, except for the search for correlated patterns which took 20 s. For the $800 \times 1,500$ dataset, these times were, respectively, 3 and 35 s. For the $800 \times 2,500$, they were 4 and 55 s.

We were satisfied by the visualization of these large datasets. However, this statement requires more precise evaluation, including a specific user study once the multi-threaded implementation is effective. Here, we performed a first experiment to evaluate how the large datasets can be accurately visualized on a screen with a resolution of $1,920 \times 1,080$. For the three large datasets, we added a small pattern at the end of the time axis. This pattern was a $n = 4 \times k = 6$ submatrix with high values (i.e., a white spot). This pattern thus represented 24 values, which is $< 0.003\%$ of the smallest of the three datasets. Then, starting from the tip of the tube (i.e., from the initial time steps of the time axis), we translated our position along the tube axis, until this pattern became visible. Then, we measured the distance, in time steps, from our actual position to the pattern position. We repeated this experiment five times for each dataset. The temporal distance values we measured varied from 750 to 900 time steps, with an average of 800. This means a 24-value pattern became visible when the visualization was focused on an $n = 800 \times k = 800$ submatrix.

4.3 Discussion

From the tests with data experts, we obtained the following informal feedback. The tubular shape and the visual representation of the datasets were easily understood by the experts: the tubular shape seems to be indeed natural. The users had no difficulties with the navigation using the simple navigation mode. We did not test the other more complex modes because, in our previous work, we noticed that users had difficulties with advanced 3D navigation. The reordering of the

attributes was greatly appreciated. It helped several experts to identify clusters of similar attributes. For the EEG dataset, the user would have liked to know the electrode locations (i.e., on the head of the patient), which is not possible in DataTube2. This could be a part of our future work. In those sessions, we did not propose to the experts any other tool to visualize their data and to make some comparison. Thus, to compare DataTube2 with another approach, we decided to perform a user study (as will be described in the following section).

The tests with large datasets lead to the following conclusions. If the user is interested in the visualization only, then the times we obtained are acceptable: the visualization was equally fluid for small and large datasets. However, if interactions are necessary, then 55 s is somehow long. A way to improve these interaction times would be to parallelize the search of similar patterns. With a multi-threaded implementation, searching the matrix for correlated areas in parallel is straightforward and should greatly decrease the response times. This is one of our perspectives.

The experiment on the context area with a small pattern to be detected gives a rough idea of the level of details DataTube2 can efficiently reach for large datasets. It counterbalances the argument that was presented in [34] about the infinite length of the tube: in practice, the precision of the screen limits the amount of data that can be efficiently perceived. In the future, we would like to perform a specific user study about this point.

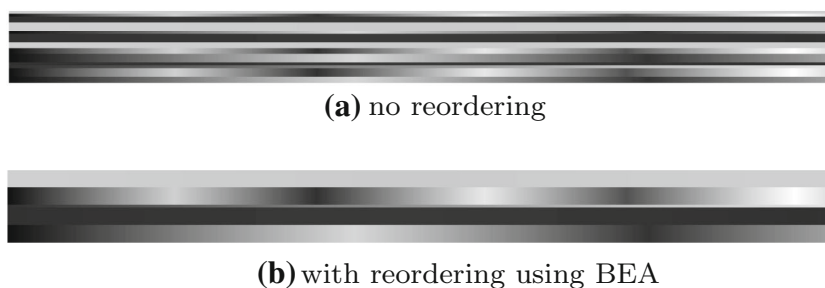
5 Experimental comparison

5.1 Overview

We performed a user study to better evaluate the effectiveness of DataTube2 and to compare it to a 2D representation. As far as we know, no such study has yet been conducted for tubular visualizations. Other evaluations of temporal visualizations exist, but they are still rare [3, 32]. In addition, radial representations were compared to Cartesian ones, but with 2D representations only [21, 37].

After analyzing the case studies, our intuition was that the tubular shape, with the perspective effect, helps to obtain an overview and details about the data. Thus, we wondered whether the tubular representation could be more efficient than a 2D representation that would be based on a 2D matrix (i.e., an unfolded tube). Such an evaluation would also contribute to the 2D versus 3D debate. In addition, we also wanted to know whether the reorganization influences the comparison between the tubular shape and the 2D shape. A final objective of the user study was to obtain informal feedback from the users.

Fig. 8 A representation of a dataset with a 2D matrix (Spreadsheet). **a** The initial matrix and **b** the reorganized matrix. This representation can be compared to the DataTube2 representation of the same dataset (see Fig. 6)



To perform a comparative study between DataTube2 and another method that would use a 2D matrix, we looked for a software program that could visualize similar data and that users would know well. This comparison was also suggested by several colleagues who attended a demo of DataTube2. Therefore, we selected a spreadsheet (Microsoft Excel) where a 2D matrix representation can be defined with a color or gray level encoding of the data (see Fig. 8). The user can move in this visualization and can zoom in on specific areas using the standard spreadsheet commands. For our approach, the use of advanced interactions (such as the discovery of correlated areas) was prohibited because the comparison with the spreadsheet would not have been fair. No such functionality is implemented in Excel. So DataTube2 was limited to the simple navigation mode (i.e., the equivalent of the zoom the user can obtain in the spreadsheet).

Three tasks were defined to evaluate the tested methods. The first task consisted of finding a specific event in the data. This event was defined as a high or maximum value that could represent an abnormal peak in the time series. This task was intended to be “context-free”: finding the peak should only require a local and very limited context. This would test the ability of each method to detect a local event. The second task was related to similarity detection and was intended to involve a broader context for decision making. The user had to find an outlier attribute, with a behavior that was different from the other attributes (see the next section for more details). Hence, a complete overview would be necessary to solve this task. This allowed us to test the ability of the tube to represent an overview and details. The third task was the same as the previous one, but with the use of the reorganization for both methods (2D and 3D).

For each of these tasks, we designed the datasets to be used and we defined standard criteria to measure the users’ performances (response time, quality of the response). We formulated several hypotheses to be tested with statistical tests (see the next section). We recruited 14 users in our laboratory and university. The users had a minimum level of education in Computer Science, Statistics and Data Mining ranging from College to PhD or PostDoc. They were aged from 22 to 41 years (with 5 women and 9 men). To obtain informal feedback, we prepared a post-task questionnaire that allowed users to rate DataTube2 and the spreadsheet.

5.2 Tasks and hypotheses

The tasks to be solved were selected among common Data Mining tasks:

- T_1 was an event detection task. An outlier value was defined in the data. This event was a peak in a given attribute. This peak was a global maximum value, for all attributes and time steps. The user was asked to scan the data and to find this event. This task ended when the reader read aloud the value. Reorganization was not used for this task.
- T_2 was an attribute detection task. We defined an attribute with a behavior that was different from the others: this attribute belonged to none of the existing groups of attributes (considering that the attributes within a group had a similar behavior). Again, the reorganization was not used for this task.
- T_3 was similar to T_2 , but with a different testing condition: the reorganization was used. All datasets were previously reorganized.

We measured the time needed to answer for all tasks. No time limit was imposed on the users. The quality of the answer was a boolean value (0 for a wrong answer, and 1 for a correct answer).

The hypotheses we wanted to test concerned the comparison between the two visualizations. The general form of the tested hypotheses was:

$$H_0(\text{time or quality, Task}) : \text{Mean}(\text{Spreadsheet}) \\ = \text{Mean}(\text{DataTube2})$$

This hypothesis can be formulated for the two possible measures (time or quality) and for each task. If it is rejected, then it means a significant difference exists between the two compared methods, for the considered measure and task. To compare the response times (continuous variables), we used the R software and the ANOVA procedure. To compare the quality (boolean variable), we used the Fisher exact test for proportions.

In the final questionnaire, we used a 5-point Likert scale to encode the answers. The tested hypotheses were:

$$H_0(\text{question}) : \text{Mean}(\text{Spreadsheet}) = \text{Mean}(\text{DataTube2})$$

For these ordinal data, we used the Wilcoxon signed-rank test.

5.3 Design and apparatus

For each task and each method (Spreadsheet or DataTube2), the 14 users were given a dataset for training, before the task began. Then, two repetitions of each task were performed for each method, with a different dataset each time. This yielded $2 \times 2 \times 14 = 56$ tests for each task (and a total of 168 tests). The number of generated datasets,⁴ including the one for training, was $1 + (2 \times 2) = 5$ for each task (and 15 for all tasks). All tests were randomized in such a way that the user never used, in the same order, the same method or the same dataset.

For T_1 , we defined five datasets with identical properties: all data matrices had $n = 100$ attributes that were described over $k = 1,000$ time steps. Data values ranged from $-1,000$ to $2,000$, with an average of approximately 100. The event to be detected was the 2,000 value (unique in the dataset). This data matrix looked like a “starry sky” (see Figs. 4, 5): we used a gray level encoding of the data, and many multimodal peaks were present around the 1,000 value. The 2,000 value was thus an outlier that appeared as a white spot in the data. Finding this peak required visually scanning all of the data values and detecting a spot that was whiter than the other darker values in the background. Only a local context was necessary to find this value because of the “starry sky” (sparse) aspect of the matrix.

For T_2 and T_3 , we defined 10 datasets with the following common properties: they had $n = 25$ attributes described over $k = 350$ time steps. Five clusters of attributes were defined: four of them had six attributes, and the last cluster had one attribute and was the outlier to be found. Within a cluster, all attributes had an equal behavior. For T_2 , the ordering of the attributes was randomly generated (see an example in Fig. 6a), and for T_3 , BEA was used to reorganize the attributes (see an example in Fig. 6b). We initialized BEA in a deterministic way to remove any dependency in the random initialization of this method (and to guaranty that the testing conditions would not change from one user to the next). This means that for a given dataset, the reorganization proposed by BEA was always the same. For this task, the perception of a larger context was important because an overview of the complete evolution of the attributes was necessary to determine the similarity between the attributes.

⁴ The datasets can be send to the reader upon request to the authors.

Table 2 The results obtained with DataTube2 and with Spreadsheet for each task

Task	Method	Time (s)	(P value)	Quality	(P value)
T_1	DataTube2	110.6	(0.09)	1	
	Spreadsheet	73.9		1	
T_2	DataTube2	33.4	(<0.001)	0.89	(0.05)
	Spreadsheet	91.2		0.67	
T_3	DataTube2	5.2	(<0.001)	1	
	Spreadsheet	11.9		1	

The times were compared with an ANOVA test, while the qualities were compared with a Fisher exact test

The users visualized the datasets using a 24” screen with a resolution of $1,920 \times 1,080$, and a standard PC was used (i7 CPU), equipped with a basic GPU.

5.4 Results of the user study

The results for each task are presented in Table 2. Concerning T_1 , we found that all users were able to fulfill the task and to find the high peak in the data: the mean quality is 1, which means all answers were correct. Regarding the time to answer, we observed that this task required some attention (the dataset had $n \times k = 100,000$ values), and that 1–2 min were sometimes necessary to solve it. In general, the users scanned the data several times, both with DataTube2 or Spreadsheet. No significant difference can be detected between the two methods, but the means are in favor for Spreadsheet. More precisely, the users generally exhibited the following behavior with the 2D representation. They zoomed in to visualize in as much detail as possible, the 100 series, without regard to the shortened time interval that resulted from such a zoom. They adjusted the zoom level to keep the 100 horizontal lines (attributes) inside a single screen. Thus, it was a zoom on the vertical axis (attributes) rather than a the time axis. Then, they navigated along the time axis using the scroll bar. Indeed, this was a good strategy because this task required only a limited context, a point the users naturally understood. With DataTube2, the users only navigated along the time axis. They were not interested in trying to zoom in on the data because they wanted in general to see all of the attributes yielding less details than with the spreadsheet. Perhaps we could have more emphasized the zoom and the “take me there” moves during the training, but we did not want to interfere too much with the users when they were discovering the tasks.

For the second task T_2 , some users were not able to successfully identify the outlier attribute. The proportion of success was more important with DataTube2 compared to Spreadsheet, but the statistical significance is borderline. Perhaps the tests would have been more significant with a few more users. For the response times, DataTube2 clearly out-

Table 3 A summary of the answers given by the users in the final questionnaire (comparison between Spreadsheet and DataTube2)

Question	Answers for Spreadsheet	Answers for DataTube2	Prob.
Simple navigation	3.35 (1.15)	4.14 (0.86)	0.027*
Easy to locate data	2.78 (0.97)	4.41 (0.66)	0.007*
Aestheticism	2.71 (0.99)	4.78 (0.42)	0.001*
Speed/frame rate	3.21 (1.05)	4.28 (0.72)	0.008*
Would use it for my data	2.92 (1.32)	4.64 (0.63)	0.0005*

The values are given as an average of the 14 users (with standard deviations in parenthesis). “Prob.” is the probability resulting from the Wilcoxon tests (when Prob < 0.05, the compared averages are significantly different, as denoted by “*”).

performs Spreadsheet. To solve this task, the users needed to look at the complete data to compare their evolution through time (from beginning to end). Keeping the context either in the users’ memory or on the screen was thus important for this task, and the tubular shape was clearly more helpful than the 2D shape.

For the final task, T_3 , we first observed that the quality was now 1 and that there was no difference in quality for the two tested methods. In addition, the response times were approximately eight times shorter compared to T_2 . This confirms that the reorganization greatly eases the search of the outlier. Furthermore, we observed the same difference between DataTube2 and Spreadsheet: the response times were significantly shorter with the tubular representation. Thus, even if the reorganization makes the search for an outlier much easier, we still observe an important difference between DataTube2 and Spreadsheet.

The results of the post-task questionnaire are presented in Table 3. Each question concerned specific and subjective aspects of our method. All answers are in favor of DataTube2. For a subset of answers, the difference is more important than for others. For instance, the answers to the “Easy to locate data” question show that the users were able to easily understand the tubular shape and the organization of the data on the tube. The answers to the “Aestheticism” question underline that the tube is a more elegant way to represent the data than a 2D matrix. In general, the users are more enthusiastic about 3D representations than 2D. The answers to “Would use it for my data” show that the users were likely to reuse DataTube2 for their own data. Then, we found that the users were able to easily navigate within the tube (“simple navigation” question), which shows that the tubular shape was easily understood and that the simplified moves we introduced were easily learned.

6 Discussion and conclusion

In this paper, we contributed to the study of the tubular metaphor in the context of the interactive visualization of large temporal datasets.

We showed that this visualization can be extended to make it more operational for VDM tasks. In the visual representation, we added two elements that help the user to better locate himself along the time axis. We proposed representing missing values, which often occur in temporal data. We added the possibility to create annotations in the visualization, which can be useful for identifying and storing specific events and discovered knowledge. We integrated a rearrangement clustering algorithm to reorganize the visualization and to better highlight the similarities among the attributes. For the interactions, we added the possibility to change the focus and context areas. This was achieved through using simple and complex navigation modes adapted for beginners or more advanced users, respectively. We gave the user the ability to select data and to automatically detect correlations, with a threshold that can be interactively adjusted. Such interactions can be useful for data analysis and for detecting periodic behaviors or similarities in the attributes and/or time steps.

We argue that these extensions go beyond the previous tubular visualizations, which used a fixed point of view or had no interactions. The proposed extensions increase the possibilities that are offered by such approaches. Other evolutions could also be studied in the future. For instance, alternative and possibly more efficient visual encodings could be studied, such as [37]. We also wish to study other alternatives for the visual representation of the time axis. One of them would consist of adding, at regular time intervals, a ring with a specific color. This would be an equivalent of a grid as used in 2D graphs. About the detection of correlated areas, we want to improve further their visual exploration. For this purpose, to be sure that the user does not miss any detected area, we intend to add a small annotation to each of them. We would like to implement a specific move that would automatically take the user from one area to the other. About such “take me there” moves, we observed that they greatly helped the users to navigate in 3D, and we would like to further evaluate this point in a more quantitative way.

We tested this visualization in several real-world contexts by allowing data experts to explore their own data. The infor-

mal feedback we obtained was positive. We checked that the tubular shape and the simple navigation mode were easily understood. For one dataset, we gave more details about how a user could interact with the visualization and make some discoveries. We studied the limits of DataTube2 for datasets whose sizes are competitive with other recent visualizations. We showed that the frame rate was acceptable, but that some interactions can take a long time. We found that a multi-threaded implementation would certainly solve this problem. We also showed that, despite the fact that the tube length can be conceptually infinite, the level of detail is restricted to a shorter length. We would like to further study the visual scalability of our approach, and especially its limits in terms of the number of attributes. This will require a specific user study. Above this limit, we would like to propose the use of a clustering method to adjust the amount of displayed information to a level that would be efficiently perceived.

We performed a comparative user study to better evaluate the effectiveness of the tubular shape and to quantify the users' feedback. We compared the tubular shape to a 2D matrix. To make a fair comparison, we did not use the advanced interactions in DataTube2. The results showed that when the context is necessary to solve a task, the tubular shape is significantly more efficient than a 2D matrix. This suggests, for the next evaluation we intend to perform, that DataTube2 could be compared to another 2D focus+context technique and with different zooming conditions. The informal feedback from the tested users was very positive and in favor of DataTube2. It seems that our tubular visualization is simple to understand and that it avoids possible drawbacks of 3D representations (such as occlusions, negative perspective effects, and difficulty of navigation). This ease of use and the aestheticism of the 3D tube contribute to the users' satisfaction we observed.

Other issues remain to be addressed, which will require additional evaluations and developments. One possible drawback of a tubular shape is that the user, when navigating forward, for instance, only sees the future and not the past. During our tests, no user complained about this, but we would like to design a specific user study to better evaluate this point. In such a study, the user would be asked to find two similar patterns, one being located in the past and another in the future. Two conditions would be tested: with the interactive search and without. Another issue we would like to address is the integration of DataTube2 with a 2D representation (based on standard graphs). We believe that once the user has selected some data in the tube (with the interactive selection or with the search for similar patterns), he should be able to filter such selection and to observe it in greater detail on a classical graph-based representation.

References

1. Aigner, W., Miksch, S., Müller, W., Schumann, H., Tominski, C.: Visualizing time-oriented data—a systematic view. *Comput. Graph.* **31**(3), 401–409 (2007)
2. Aigner, W., Miksch, S., Schumann, H.: *Visualization of Time-Oriented Data*. Human-Computer Interaction Series. Springer, Berlin (2011)
3. Aigner, W., Rind, A., Hoffmann, S.: Comparative evaluation of an interactive time-series visualization that combines quantitative data with qualitative abstractions. In: *Computer Graphics Forum*, vol. 31, pp. 995–1004. Wiley Online Library (2012)
4. Andrienko, G., Andrienko, N.: Dynamic time transformations for visualizing multiple trajectories in interactive space-time cube. In: *International Cartographic Conference, ICC* (2011)
5. Ankerst, M.: Visual data mining with pixel-oriented visualization techniques. In: *Proceedings of the ACM SIGKDD Workshop on Visual Data Mining* (2001)
6. Ankerst, M., Berchtold, S., Keim, D.A.: Similarity clustering of dimensions for an enhanced visualization of multidimensional data. In: *Proceedings of the 1998 IEEE symposium on information visualization, INFOVIS '98*, pp. 52–60. IEEE Computer Society, Washington, DC, USA (1998)
7. Ankerst, M., Keim, D.A., Kriegel, H.P.: Circle segments: a technique for visually exploring large multidimensional data sets. In: *Proceedings of IEEE Visualization'96, Hot Topics 96* (1996)
8. Antunes, C.M., Oliveira, A.L.: Temporal data mining: An overview. *KDD Workshop on Temporal Data Mining* (2001)
9. Aris, A., Shneiderman, B., Plaisant, C., Shmueli, G., Jank, W.: Representing unevenly-spaced time series data for visualization and interactive exploration. In: *Human-Computer Interaction-INTERACT 2005*, pp. 835–846. Springer (2005)
10. Beardsley, T.: PROFILE: humans unite! *scientific american* (1999). <http://www.cs.ucsd.edu/users/goguen/courses/171sp02/shneiderman.html>
11. Bertin, J.: *La graphique et le traitement graphique de l'information*. Nouvelle Bibliothèque Scientifique. (1977)
12. Carlis, J.V., Konstan, J.A.: Interactive visualization of serial periodic data. In: *Proceedings of the 11th annual ACM symposium on User interface software and technology* pp. 29–38 (1998)
13. Cleveland, W.S.: *Visualizing Data*. Hobart Press, Summit (1993)
14. Craig, P., Roa-Seiler, N.: A vertical timeline visualization for the exploratory analysis of dialogue data. In: *Information Visualisation (IV), 2012 16th International Conference on IEEE*, pp. 68–73. (2012)
15. Daassi, C., Dumas, M., Fauvet, M.C., Nigay, L., Scholl, P.C.: Visual exploration of temporal object databases. In: *proceedinga of BDA00 conference* pp. 24–27 (2000)
16. Francis, B., Pritchard, J.: Visualisation of historical events using Lexis pencils. *Case Stud. Vis. Soc. Sci.* **30** (2003)
17. Hackstadt, S.T., Malony, A.D.: Visualizing parallel program and performance data with IBM Visualisation data explorer. Master's thesis (1994)
18. Hao, M.C., Marwah, M., Janetzko, H., Dayal, U., Keim, D.A., Patnaik, D., Ramakrishnan, N., Sharma, R.K.: Visual exploration of frequent patterns in multivariate time series. *Inf. Vis.* **11**(1), 71–83 (2012)
19. Hébrail, G., Debregeas, A.: Interactive interpretation of Kohonen maps applied to curves. In: *Proceedings of the 4th international conference on knowledge discovery and data mining*. AAAI press, Menlo Park pp. 179–183 (1998)
20. Heer, J., Shneiderman, B.: Interactive dynamics for visual analysis. *Queue* **30**(30), 30–55 (2012)

21. Hofmann, H., Follett, L., Majumder, M., Cook, D.: Graphical tests for power comparison of competing designs. *IEEE Trans. Vis. Comput. Graph* **18**(12), 2441–2448 (2012)
22. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv. (CSUR)* **31**(3), 264–323 (1999)
23. Kandogan, E.: Star coordinates: a multi-dimensional visualization technique with uniform treatment of dimensions. *IEEE Symp. Inf. Vis.* **2000**, 4–8 (2000)
24. Keim, D.A., Ankerst, M., Kriegel, H.P.: Recursive pattern: a technique for visualizing very large amounts of data. In: *Proceedings of the 6th conference on visualization '95* pp. 279–286 (1995)
25. Keim, D.A., Kohlhammer, J., Ellis, G., Mansmann, F. (eds.): *Mastering the information age—solving problems with visual analytics*. Eurographics (2010). <http://www.vismaster.eu/book/>
26. Keim, D.A., Kriegel, H.P.: Visualization techniques for mining large databases: a comparison. *IEEE Trans. Knowl. Data Eng.* **8**(6), 923–938 (1996)
27. Kincaid, R., Lam, H.: Line graph explorer: scalable display of line graphs using focus+context. In: *Proceedings of the working conference on advanced visual interfaces, AVI '06*, pp. 404–411. ACM (2006)
28. Krstajic, M., Bertini, E., Keim, D.: Cloudlines: compact display of event episodes in multiple time-series. *IEEE Trans. Vis. Comput. Graph.* **17**(12), 2432–2439 (2011)
29. Lin, J., Keogh, E., Lonardi, S., Lankford, J.P., Nystrom, D.M.: Viztree: a tool for visually mining and monitoring massive time series databases. In: *Proceedings of international conference on very large data bases*, pp. 1269–1272 (2004)
30. Mackinlay, J.D., Robertson, G.G., Card, S.K.: The perspective wall: Detail and context smoothly integrated. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 173–176. ACM (1991)
31. McCormick, W.T., Schweitzer, P.J., White, T.W.: Problem decomposition and data reorganization by a clustering technique. *Operat. Res.* **20** **5**(5), 993–1009 (1972)
32. McLachlan, P., Munzner, T., Koutsofios, E., North, S.: Liverac: interactive visual exploration of system management time-series data. In: *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 1483–1492. ACM (2008)
33. Minard, C.J.: *Carte figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie*. pp. 1812–1813 (1861)
34. Mitchell, K., Kennedy, J.: The perspective tunnel: an inside view on smoothly integrating detail and context. In: *Visualization in scientific computing '97: proceedings of the Eurographics Workshop*, Springer Computing Science. Springer (1997)
35. Muller, W., Schumann, H.: Visualization methods for time-dependent data—an overview. *Simulation conference, 2003*. In *Proceedings of the 2003 Winter 1*, pp. 737–745 (2003)
36. Nekrasovski, D., Bodnar, A., McGrenere, J., Guimbretière, F., Munzner, T.: An evaluation of pan and zoom and rubber sheet navigation with and without an overview. In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 11–20. ACM (2006)
37. Oelke, D., Janetzko, H., Simon, S., Neuhaus, K., Keim, D.A.: Visual boosting in pixel-based visualizations. In: *Computer Graphics Forum*, vol. 30, pp. 871–880. Wiley Online Library (2011)
38. Shmueli, G., Jank, W., Aris, A., Plaisant, C., Shneiderman, B.: Exploring auction databases through interactive visualization. *Decis. Support Syst.* **42**(3), 1521–1538 (2006)
39. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: *IEEE visual languages, UMCP-CSD CS-TR-3665*, pp. 336–343. College Park, Maryland 20742, U.S.A. (1996). <http://www.citeseer.ist.psu.edu/shneiderman96eyes.html>
40. Suntinger, M., Obweger, H., Schuh, J., Gröller, M.E.: Event tunnel: exploring event-driven business processes. *IEEE Comput. Graph. Appl.* **28**(5), 46–55 (2008)
41. Sureau, F., Plantard, F., Bouali, F., Venturini, G.: Visual mining of web logs with DataTube2. In: *Tenth international conference on web information system engineering (WISE), LNCS*, Springer, pp. 555–562 (2008)
42. Theron, R.: Hierarchical-temporal data visualization using a tree-ring metaphor. *Lect. Notes Comput. Sci.* **4073**(2006), 70–81 (2006)
43. Tominski, C., Schumann, H.: Enhanced interactive spiral display. In: *Proceedings of the annual SIGRAD conference, special theme: interactivity*, pp. 53–56 (2008)
44. Wattenberg, M.: Arc diagrams: visualizing structure in strings. *information visualization, 2002. INFOVIS 2002. IEEE symposium on* pp. 110–116 (2002)
45. Weber, M., Alexa, M., Muller, W.: Visualizing time-series on spirals. *Information visualization, 2001. INFOVIS 2001. IEEE symposium on* pp. 7–13 (2001)
46. van Wijk, J.J., van Selow, E.R.: Cluster and calendar based visualization of time series data. In: *Proceedings of IEEE symposium on information visualization* pp. 4–9 (1999)
47. Wong, P.C., Bergeron, R.D.: 30 years of multidimensional multivariate visualization. *Scientific visualization—Overviews. Methodologies and Techniques*, pp. 3–33. IEEE Computer Society Press, Los Alamitos, CA (1997)



Fatma Bouali is currently an associate Professor of Computer Science at the University of Lille2 in France. She received a PhD from the University of Paris-Sud. Her research interests include visualizations for Business Intelligence, interactive exploration of databases and visual data mining.



Sébastien Devaux obtained a Master degree in Computer Science from the University Francois-Rabelais of Tours. His research interest concern time dependent data visualization. He is working in the industry.



Gilles Venturini is a Professor of Computer Science at the University Francois-Rabelais of Tours in France. His main research interests concern visual data mining, the optimization of visualizations (using biomimetic algorithms such as Genetic Algorithms), Human-Computer interaction (in the context of visual data mining), 3D data acquisition and immersive visualization. He is the co-editor in chief of the French New IT Journal (Revue des Nouvelles Technologies de

l'Information) and was recently elected President of the French Data Mining Society (Extraction et Gestion des Connaissances).