ViDX: Visual Diagnostics of Assembly Line Performance in Smart Factories

Category: Application

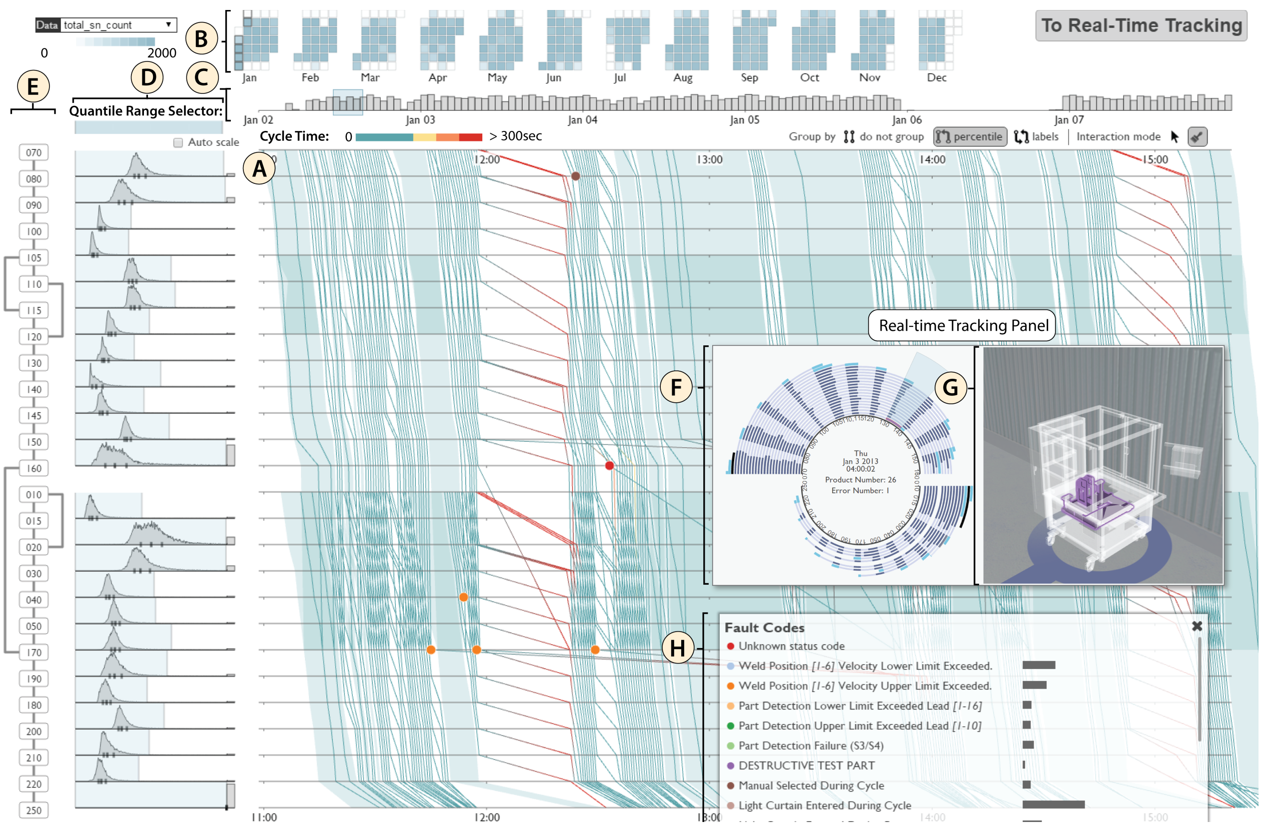

Fig. 1. A screenshot of the ViDX system for the historical analysis and real-time tracking of assembly line performance. The historical data analysis panel consists of an extended Marey's graph (A), for trouble-shooting inefficiencies and faults occurring on the assembly lines. It is combined with a set of interconnected views including a calendar based visualization (B) and a timeline (C) for multi-scale temporal exploration. Auxiliary views include small multiples of histograms (D) showing the distribution of the cycle time on each station with a quantile range selector, and a station map denoting the assembly line schema (E). The real-time monitoring panel includes a radial graph (F) and an explorable 3D station model visualization (G). (H) shows the color codes of the faults and their total occurrences.

Abstract— Visual analytics plays a key role in the era of connected industry (or industry 4.0, industrial internet) as modern machines and production (assembly) lines can generate large amount of data, and effective visual exploration techniques are needed for troubleshooting, process optimization and decision making. However, developing effective visual analytics solutions for this application domain is a challenging task due to the sheer volume and the complexity of the data generated in the sophisticated manufacturing processes. In this paper, we report the design and implementation of a comprehensive visual analytics system, ViDX. It supports both real-time tracking of the assembly line performance, and exploration of historical data to identify the inefficiencies, locate the abnormalities, and form hypotheses about their causes and effects in assembly lines. The system is designed based on a set of requirements gathered through discussions with the managers and operators from manufacturing sites. It features interlinked views displaying data at different level-of-details. In particular, we apply and extend Marey's graph by introducing a time-aware outlier-preserving visual aggregation technique to support effective troubleshooting in manufacturing processes. We also introduce two novel interaction techniques, namely quantiles brush and samples brush, for the users to interactively steer the outlier detection algorithms. We evaluate the system with example use cases and an in-depth user interview, both conducted together with the managers and operators from manufacturing plants. The result demonstrates its effectiveness and usability, and reports a successful pilot application of visual analytics for manufacturing in smart factories.

Index Terms—Temporal Data, Marey's Graph, Visual Analytics, Manufacturing, Smart Factory, Connected Industry, Industry 4.0

1 Introduction

Connected industry (or industry 4.0, industrial internet) is an increasingly important topic of worldwide significance [3, 10, 11, 17]. It facilitates the vision and execution of "Smart Factories". The smart factories, in comparison to traditional manufacturing environments, are equipped with machines that are highly digitalized and connected. Every status and condition change, or occurrence of abnormal events can be continuously recorded and stored. The investigation of such data has the potential to bring important insights to the managers and operators to perform troubleshooting and further optimize the processes to reduce operation cost and increase profit. Recently, a number of successful use cases have already been reported, ranging from pharmaceutical to mine industries [4], where, for example, statistical methods have been applied to track the production process and analyze factors related to the yield. However, to the best of our knowledge, few examples have been reported to apply visual analytics to the investigation of manufacturing data, despite that it has been identified as an important component in connected industry, where it can play an crucial role in making sense of the increasingly complex and large data collected [27]. We believe that it would be very valuable, for both the industry stakeholders, and the visualization research community, to explore the possibility of applying visual analytics in this domain.

We work closely with managers and operators in manufacturing sites which produce automotive parts, to develop a visual analytics system to support real-time tracking of assembly line performance, and historical data analysis. The data include both real-time and historical records of the status and operational information from the shop floor, where the assembly lines are located.

Assembly lines on the shop floor consist of sequences of work stations (machines). Each station corresponds to a stage of production where specific procedures are carried out on the products. The products (automotive parts in the study) are moved through the stations, tested, and shipped out in the end (to car manufacturers). During the operation of the assembly line, data are recorded about when the product is moved from one machine to the next, and also about any fault that occurred during the process. This kind of setting is becoming increasingly common in the modern assembly lines where almost every operation is trackable. The collected manufacturing process data is valuable for monitoring real-time assembly line performance to facilitate rapid response of operators and managers. Furthermore, by analyzing historical records, they can gain insight about when, where, and how the production efficiency decreases, and identify if there is any systematic problem with the assembly lines and the manufacturing environment.

We summarize the main contributions of this work as follows:

- We formulate the design requirements for interactive visual diagnostics of assembly line performance, together with the target users, i.e., operators and managers from manufacturing sites [22,28].
- We design and implement a prototype system based on the requirements. We perform case studies and conduct user interviews to assess its effectiveness and usability.
- We apply and extend Marey's graph by introducing a novel time-aware outlier-preserving visual aggregation technique, to facilitate the identification of abnormalities and support troubleshooting in a large number of manufacturing process data.
- We propose two novel interaction techniques for user steerable outlier detection and aggregation of manufacturing processes data in the extended Marey's graph. One method is based on brushing quantiles and the other is built on a label propagation algorithm. We believe the methods are also generally applicable to the analysis of multivariate data in other domains.

The paper is organized as follows. First, related work is discussed in Section 2. The background and the design requirements are introduced in Section 3. The design of the extended Marey's graph is presented in Section 4 and the system is described in Section 5. In Section 6 we describe the implementation. In Section 7 we apply our approach to real-world data. We present discussion in Section 8 and conclude in Section 9.

2 Related Work

2.1 Manufacturing Data Visualization

Today's manufacturing industry has started using big data analytics to support its research and operational activities as discussed in a recent report [4]. With the launch of connected industry and industry 4.0 programs in the private and public domains [3,10,11,17], it could only be anticipated that the amount and the complexity of data collected in the industry will continue to grow in the future. Visual analytics, as an important component for gaining insight from large and complex data, can thus play a crucial role in this application domain [27].

So far only a few visual analytics solutions target at the data analysis tasks in manufacturing scenarios. Matković et al. [20] visualize sensor measurements for process monitoring. Jo et al. [16] extend the basic Gantt chart for the exploration of large schedules. They introduce novel interactions and algorithms to improve its scalability, explorability, and reschedulability. Worner and Ertl [34] propose a novel visual analytic system for simulated manufacturing processes.

These studies visualize the data related to the planning and simulation stages in manufacturing. In this work, we describe the design of a visual analytic system for manufacturing process data collected during the operation of the assembly lines in modern factories. The analytic tasks, therefore, are fundamentally different from those for planning and simulation purposes as described above.

2.2 Temporal Data Visualization

Time oriented data visualization has been extensively studied in the past years. Temporal dimension can be found in many applications [29] There are several surveys reporting the state of art of temporal data visualization techniques. Aigner et al. [1,2] categorize the visualization techniques based on the nature of the temporal dimension, i.e. whether it is cyclic, linear, or branching, and whether it is discrete time points or time intervals. Bach et al. [5] review a range of techniques and categorize them through a new perspective, by describing each technique as series of operations performed on a conceptual space-time cube, including extraction, flattening, filling, geometry transformation and content transformation.

Among the vast amount of temporal data visualization techniques, those visualizing event sequences are the most relevant to our study. In particular, the event sequence visualization techniques can be classified into two categories: the first category visualizes sequences with variant orderings and occurrences of events, and the second category visualizes sequences containing a set of prescheduled events. For the first category, examples include LifeLines [26,31] for visualizing patient medical records, Sankey diagram based visualizations for the analysis of electronic health records [12,21,25,33] and website visiting patterns [35], and most recently, matrix based visualizations [36], also for the analysis of website visiting patterns. Recently, a few interactive visualization systems have also been proposed for selecting a subset of the event sequences for focused study [13,18]. For the second category, examples include Marey's travel graph [30], which was first introduced in the 1880s for visualizing train schedules. Since then it has been used extensively to study public transportation schedules [8,15,19]. Inspired by the design, Palolm et al. [24] propose a visual analytic system for exploring transportation schedules. They apply kernel density estimation on the graph to improve the scalability of the visualization.

In this paper, we extend Marey's graph with a time-aware outlier-preserving visual aggregation technique, to support effective identification of abnormalities and inefficiencies in the manufacturing processes and facilitate troubleshooting. Novel interaction techniques are also introduced, with which the users can interactively identify the abnormalities/outliers by specifying sample normal records or brushing quantiles.

3 Data Abstraction and Requirement Analysis

3.1 Data Abstraction

A typical **assembly line** in a manufacturing environment consists of a set of **work stations**. The **parts** are moved from one station to another to be processed and assembled to form the final **product**. In recent years, there has been a widespread move to adopt general-purpose computing devices to control and monitor the industry processes. Programmable Logic Controllers (PLCs), for example, are widely deployed to control the machinery on the assembly lines for manufacturing automation [14]. The PLCs on the assembly line send the status information of the parts to a central database when they arrive at each station.

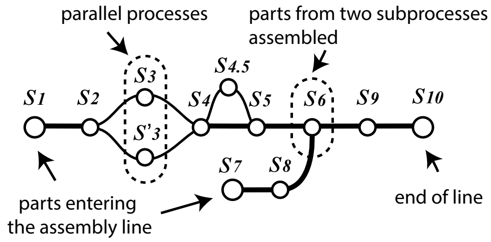

Fig. 2. The schematic view of an assembly line as a DAG (directed acyclic graph). The parts are moved among the stations following predefined paths. At stations s_1 and s_7 two different types of parts enter different sub-processes. At stations s_6 the two types of parts are assembled. Station s_3 and s'_3 perform the same procedures in parallel on the incoming parts. Sometimes the part undergoes additional procedures on station $s_{4.5}$.

Assembly lines can be considered as DAGs (directed acyclic graphs), with nodes being the work stations, which we denote as $S = \{s_i | i \in [1,n]\}$, and the edges $(s_i, s_j) \in S \times S$ in the graph indicate that the operation on s_j takes place immediately after s_i. Fig. 2 shows the schematic view of an assembly line as a DAG. In this assembly line, the parts can choose either station s_3 or s'_3 and undergo the same procedure in parallel after finishing at s_2. At station s_6, two parts from different sub-processes are brought together and assembled into a single product. On some occasions the part (or product) undergoes additional procedures on $s_{4.5}$ before being moved onto the next station. All of these structures can be modeled by describing the assembly lines as DAGs.

The PLCs record when each part p is moved onto a station s_i and starts being processed on it. We denote the time as $t(p, s_i)$. As a part is being moved along a path $P = (s_j, ..., s_k)$ on the assembly line, a sequence of timestamps is created, based on which we can calculate the time it takes for the part to finish its procedures on one station and be moved onto the next as $dt(p, s_i) = t(p, s_j) - t(p, s_i)$. This is referred as the **cycle time** of the part on station s_i. Besides the timestamps, the PLCs also record **fault codes** if any error has occurred when a part is being processed on a station. The timestamps and fault codes together are referred to as the **trace** or **process data** of the corresponding part. The process data of all the parts composing a product can be combined. Processes with comparatively longer cycle times on one or more stations, or with faults, are referred to as **outliers** or **abnormal processes**.

To summarize, the **invariants** in the data collected from the manufacturing processes are the predefined sequences of work stations and procedures described by the DAGs, and the **variants** are the timings when the parts (or product) reach a station (with the cycle times derived from it) and the occurrences of faults. The target users have informed us that these are the most important variables amongst many measurements they have recorded. One underlying reason is that the assembly lines employ pipelining to concurrently process multiple parts on different stations. Due to the inherent sequential dependency in a pipelined process, the delay on even a single station may stall and affect the throughput of an entire assembly line, thus having impact on the ability to meet targets of production, and eventually the profit. Therefore it is very desirable for the operators and the managers to be able to access real-time line performance and be notified of any potential problems. Moreover, the data provide an extremely accurate and complete description of the assembly line operations. By analyzing the data, the users can identify the abnormal processes, understand when, where, and why the efficiency decreases, and perform troubleshooting, with the ultimate goal of identifying opportunities to reduce losses and increase profit.

Therefore, our focus in this study is to design an informative and intuitive visualization interface for both real-time monitoring of assembly line performance and historical data analysis.

3.2 Design Process and Requirement Analysis

Based on discussions with the managers and operators, we formulate a set of requirements to guide the design of the system.

Overall the project took about six months. In the beginning the collaborators gave us the access right to their production databases. They pointed us to the data that they interest most, i.e., the cycle times and the faults in the manufacturing processes, and presented us some initial visual design ideas (e.g., the radial display in Fig. 5(a)). During the following six months we had frequent (approx. biweekly) video conferences and in-person meetings as well as email discussions, mostly about the semantics of the data attributes when we started building the system, and more about the feedback on the prototypes at a later stage. The meetings usually involved a person at a managerial position responsible for the "big data in industry 4.0" program in the plant and technical staffs responsible for the design/maintenance of the databases. The design requirements are formulated iteratively throughout the six months.

For historical data analysis, we identify the following design requirements:

R1 ***Facilitate the detection of abnormal processes.*** The visual encoding should highlight the abnormal process and show when and on which stations the delay or faults has occurred. Detecting outliers is the essential first step to more in-depth analysis.

R2 ***Facilitate the detection of inefficiencies and support troubleshooting.*** The system should allow users to identify time periods with low production efficiency, and to form hypothesis about their causes.

R3 ***Engage users to detect outlier processes interactively.*** Many automatic outlier detection algorithms can be applied to support efficient identification of abnormal processes [9]. Although it is possible to directly apply those algorithms and encode the end results in the visualization, we believe that it would be extremely beneficial to engage the users with domain knowledge and experience in operating the assembly lines in this process. To this end, we should provide interactive outlier detection functionalities that are easy to use and do not require the users to understand the technical details.

R4 ***Support predictive analysis by associating the abnormal processes and inefficiencies with the surrounding context of assembly line operation.*** The occurrences of the delays and faults may have certain causes and effects. The causal relations identified can provide insights for building predictive models. Based on the predictions, the operators and managers can take preventive measures to reduce losses.

For tracking real-time assembly line performance, we identify the following design requirements:

R5 ***Highlight abnormalities in real-time data.*** Similar as in historical data analysis, abnormalities such as delays and faults should be highlighted such that the operators and managers can respond immediately and prevent losses.

R6 ***Associate data with the physical context; visually indicate problematic components in 3D models.*** Besides showing the abstract status information, it is also important for the users to be able to quickly locate the corresponding stations in physical environments. Since the fault codes are related to specific components in the stations, we can highlight those components in 3D model to support troubleshooting.

Besides the requirements described above, the following items are also equally important:

R7 ***Support smooth, interactive exploration of large amount of process data.*** In manufacturing industry, it is typical that thousands of products are made every day and millions of products are made every year on a single assembly line. To support interactive exploratory analysis of the large dataset, the system should be scalable, both visually and algorithmically.

R8 ***Use familiar visual metaphors and respect users' mental models about assembly line operation.*** Since few of our target users have experience with advanced visual analytics applications, it is particularly important to keep the visual designs intuitive and easy to understand. Therefore, we make careful design choices considering these aspects.

4 Extended Marey's Graph

In this section, we present the main visual component in the system, the extended Marey's graph, for historical data analysis. Because a

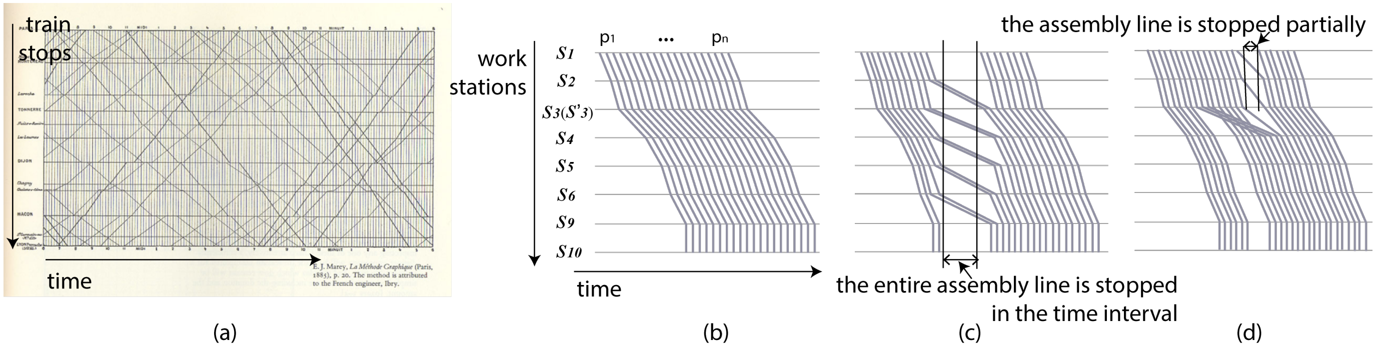

Fig. 3. Marey's graph [30] and the relevant visual patterns when applied to the manufacturing process data. It uses the path $(s_1, s_2, s_3(s'_3), s_5, s_6, s_9, s_{10})$ in the DAG in Fig. 2 for illustration. (a) the original Marey's graph shows the bus/train schedules; (b) Marey's graph showing when a product starts being processed on each work station on an assembly line, the visual pattern shows that no abnormal delay has occurred and the assembly line works smoothly; (c) the assembly line is completely stopped during a time interval, which can be caused by faults or prescheduled maintenance; (d) the assembly line is partially stopped to handle unprocessed products.

direct application of the Marey's graph would result in visual clutter, affecting the visibility of the outliers, we introduce a time-aware, outlier-preserving visual aggregation technique to enhance it. To support this technique, we include computational outlier detection methods in the system, and design interactions for the users to steer those algorithms.

4.1 Visual Encoding

Marey's graph is a traditional method for depicting bus or train schedules [30]. It employs a parallel layout of time axes. Each time axis corresponds to a train or bus stop. Polylines connecting the time points on the axes show when the buses/trains are expected to arrive at a stop (Fig. 3 (a)) based on the schedule.

This visual encoding can be directly applied to manufacturing processes data if we consider each work station on the assembly line as a bus/train stop, and the time when the parts are moved onto each work station as the time in bus/train schedules. The polylines would trace the complete history of a product on the production lines, and the angle of the line segments between the axes would indicate its cycle time on each station.

Similar as in parallel coordinate plots (PCPs), we have to decide on a linear ordering of the axes (stations) before drawing the polylines in Marey's graph. The ordering we use is a topological sort of the stations derived from the DAG. Manual adjustments are made to reduce the total lengths of the polylines. As illustrated in Fig. 1 (A), subprocesses ([070, 080, ... , 170] and [010, ... , 170]) and parallel processes ([105, 115, 120] and [105, 110, 120]) are overlaid on the same graph. This is helpful for tracing the complete history of a product which consists of multiple parts. However it might introduce undesirable line overlaps and intersections. To solve this problem, we include filtering interactions for the users to focus on particular paths on the DAG.

Marey's graph allows us to use the familiar metaphor of transportation schedules to explain the visual encoding (**R8**). It shows multivariate information that allows the detection of when and on which station the delay occurs (**R1**). More importantly, a set of recurring visual patterns emerge from the visualization, based on which the operators can form hypothesis of the causes of the inefficiency (**R2**). Here we summarize the visual patterns for the users to quickly read off some high-level semantic information from the visualization.

It should be noted that although both Marey's graph and PCPs use parallel layout of axes and polylines as the primary visual primitives to display data, they are fundamentally different on which visual patterns bear semantic meanings.

In the Marey's graph, the users can identify ***out-of-order processes***, as visually indicated by line segments crossing each other between the time axes, and abnormal ***delays***, indicated by line segments that stretch much longer than the others between two time axes.

Visual patterns can also be formed collectively by a number of visual elements. Their are listed as below. Fig. 3 illustrates the different types of visual patterns. It uses the path $(s_1, s_2, s_3(s'_3), s_5, s_6, s_9, s_{10})$ in the DAG in Fig. 2 for illustration.

- ***Streak of efficient processes.*** In Fig. 3 (b), the line segments between the axes run parallel to each other and have equal-sized displacement. This visual pattern indicates a rhythmic and smooth processing of the products on the assembly line where no delays or interruption of operations occur.
- ***Halt of the entire assembly line.*** In Fig. 3 (c), each process has experienced some delay around a certain time as indicated by the lengths and the slopes of the line segments. What actually happens is that the entire assembly line halts, and no part is being moved from one station to another. This can be caused by scheduled maintenance, breaks, or other unexpected factors.
- ***Partial halt of the assembly line to wait for continuing tasks.*** In Fig. 3 (d), station s_1 and s_2 stopped processing to wait for s_3 finishing handling the parts whose processing have been delayed. These type of events are also sources of inefficiency.

Occurrences of faults are displayed as color coded circles on the time axes of the corresponding stations. The overlay of information allows the operators and managers to quickly locate faults (**R1**) and identify the effect of the fault occurrences on the operation of the assembly line (**R4**). Besides that, we redundantly code the cycle times in Marey's graph with a green-yellow-red color scale.

4.1.1 Alternative Visual Designs

We have considered several alternative visual encodings before finally deciding on using the Marey's graph. Gantt chart, which is often used for visualizing schedules, including bus/train schedules, is one possible way to display the manufacturing process data. However, it is difficult to compare the cycle times of different processes, as they start at different times on the Gantt chart. Although interactively aligning the processes at their starting times on each station may help [16, 31], only the cycle times on one station can be compared at a time. Moreover, the temporal context is lost. In Marey's graph, the length and angle of the line segments are strong visual cues for the comparison of cycle times even without alignment of the starting time. The design invokes the Gestalt rule of similarity: line segments with similar slopes are perceived as a group by the reader [32] and the outliers will stand out (**R1**). Sankey diagram [33] and MatrixWave [36] are other possible ways to visualize event sequence data, although they emphasize the variation of the relative ordering of the events (which is fixed in manufacturing schedules) rather than the timings and the cycle times [16].

4.1.2 Time-Aware Outlier-Preserving Visual Aggregation

While a direct application of Marey's graph could reveal many interesting visual patterns, it suffers from severe visual clutter with the overplotting of lines even with a moderate amount of data. The outliers can be obscured in the display. Kernel density estimation (KDE) [24] is one possible approach to address the overplotting issue. Instead of drawing individual lines, the method estimates the density of the lines and draws a heat map of it. However, it can blur out the abnormalities (or outliers), as they usually reside in low density regions of the display. In this study, we introduce a method to reduce visual clutter while highlighting the outliers, inspired by an approach originally proposed by Novotný and Hauser to reduce the visual clutter in PCPs [23].

Fig. 4 illustrates the method. First, the processes are classified as normal ones and outliers. Then the normal processes are aggregated based on their temporal proximity, and each aggregated group is displayed as a thick band instead of individual polylines. The outliers are overlaid on top of the aggregated normal processes, displayed as individual polylines.

The aggregation of the normal processes is implemented with a greedy algorithm. It scans the processes sorted by their starting time on the assembly line. For each process scanned, it will decide whether to merge it into the current group or create a new one. If the process scanned is temporally close to the last process in the current group (i.e., the difference of their starting times at the first station is smaller than a threshold), it will be merged into the group, otherwise a new one is created. The threshold for merging the processes is determined based on the average time it takes for a new product to enter the assembly line. The aggregated processes are rendered as thick bands composed of trapezoids connecting adjacent time axes. The vertices of the trapezoids are placed at the minimum and maximum timestamps of the aggregated processes.

In this way, we are able to visualize a larger number of process and still highlight the anomalies. The aggregated processes show the surrounding context for these abnormal processes for troubleshooting (**R4**). Note that the visual patterns we identified in the last section are still visible as the related abnormal processes are displayed individually and not hidden from the viewers.

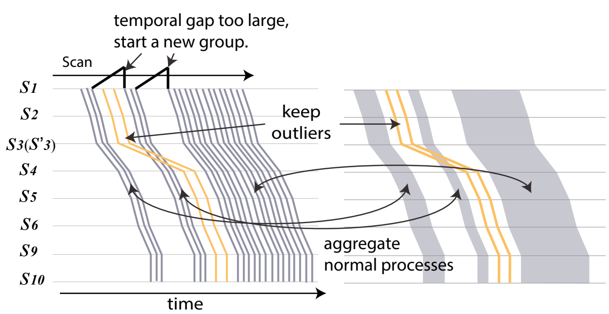

Fig. 4. Time-aware outlier-preserving visual aggregation: ① the outliers with faults or abnormal delays are identified; ② the normal processes in close temporal proximity to each other are aggregated and represented as thick bands instead of individual polylines.

However, one problem remains: which processes should be regarded as outliers and which should be considered as normal?

4.1.3 Interactive Identification of Outliers

We introduce two interactive techniques for identifying abnormal processes. We engage user input in ways that allow them to flexibly incorporate their experience with assembly line operation (**R3**). Both methods detect outlier processes based on their cycle times on the work stations.

Quantiles Brush Quantiles are descriptive statistics of a variable which splits a set of observations into equally sized bins. The p-quantile of a variable given a set of n samples is a value $q(p)$, for which there are at least np samples smaller than or equal to it and at least $n(1-p)$ samples larger than or equal to it. It is a generalization of the quartiles $(q(1/4), q(1/2), q(3/4))$ that appear in a box plot. Frequently, quantiles (mostly quartiles) are integrated in visualizations (e.g., as box plots) to give an succinct summary of the distribution of a single variable.

We introduce a brushing technique for the users to specify outliers among the processes based on quantiles. The user can select a pair of values (p_0, p_1) $(p_0 < p_1)$ from the range $[0, 1]$. The corresponding quantiles $(q(p_0), q(p_1))$ for the cycle times on each station will then be calculated. Processes with cycle times lying outside the range $[q(p_0), q(p_1)]$ on any stations are identified as outliers. The users can also fine tune the range for individual stations.

Fig. 1 (D) shows the quantile range selector implemented in the prototype, together with small multiples of histograms showing the distribution of the cycle times on each station. Outlier processes are displayed as individual polylines in the aggregated graph, and the graph interactively updates to show a new set of outliers detected. The quantile-based brushing widget provides a simple interface for specifying *statistically meaningful* parameters as the lower and upper bounds of normal cycle times.

Samples Brush We also introduce a sample-based approach to engage user input for the identification of abnormal processes. In this approach, the users label a set of normal processes, based on which the system can detect the outliers in the remaining data. We integrate the label propagation algorithm [37] for this purpose. This method can infer the class of a large number of data points even with a few labeled ones, with the prior assumption that data belonging to the same class (normal processes in this case) form densely populated regions in the high dimensional space. We find it suitable for this usage scenario, as it requires a minimum amount of user input.

Label propagation is a graph-based semi-supervised learning algorithm. It works by first constructing a neighborhood graph (e.g., k-nearest neighbor graph) containing both the labeled and unlabeled data points, then iteratively propagating the labels along the graph edges, starting from the labeled points. The iteration stops when the labels of the data points no longer change. The algorithm can be expressed formally as:

$$\text{Propagate labels:} L_X^t = A L_X^{t-1} \quad (1)$$
$$\text{Normalize rows in } L_X \quad (2)$$
$$\text{Reset originally labeled data in } L_X \quad (3)$$

Where A is the adjacency matrix of the neighborhood graph and L_X codes the labels of the data points (please refer to [37] for more details). The matrix multiplication can be parallelized on modern GPUs to support interactive performance [6].

We apply the method to identify abnormal processes based on the samples specified by the users (Fig. 9(① and ②)). First, we construct a k-nn graph of the processes, based on their cycle times on the stations, using a Euclidean distance metric. Additionally, we set a threshold on the maximum neighborhood distances in the k-nn graph to stop labels from propagating to very dissimilar processes. Second, the system propagates the normal label through the k-nn graph, gradually covering the dense regions in the data set containing the sample normal processes. The remaining unlabeled processes are outliers, which will be displayed in the extended Marey's graph as individual polylines (Fig. 9 ④), and the normal processes are aggregated (Fig. 9 ③).

The two approaches, including ***quantiles brush*** and ***samples brush***, both engage users in the computational extensive process of outlier identification (**R3**).The system will give immediate visual feedback about the results after the users change their inputs.

5 The ViDX system

5.1 Historical Data Analysis

To support the exploration and analysis of historical data, we have designed a multi-scale hierarchical display, following the visual data analysis mantra, "overview first, zoom and filter, then details-on-demand" [29]. The display consists of a calendar based visualization, a timeline, and the extended Marey's graph, showing data at different temporal scales with different level of details to support the exploration of year long data (**R7**). Fig. 1 shows an overview of the system.

Calendar View The calendar based overview shows the summary statistics including the number of productions and the faults occurred on each day over a year (Fig. 1 (B)). We choose the calendar chart as it aligns the weekdays and weekends for better cross comparison.

The user can select a continuous set of days on the calendar view. The timeline (Fig. 1 (C)) will then update its range to the selected days, and display the number of productions in a finer resolution. By brushing the corresponding range on the timeline, the user can investigate the manufacturing process information in more detail with the extended Marey's graph.

Other Contextual Views A schematic diagram (Fig. 1(E)) shows the assembly line structure. The user can select stations on the diagram to focus on a particular route related to a subprocess or one of the parallel processes. A legend (Fig. 1(H)) shows the color codes of the faults along with their total number of occurrences.

5.2 Real-Time Monitoring

For real-time monitoring, we combine a 2D radial display and a 3D visualization of the station models (Fig. 1 (F)(G)).

Radial Graph The radial graph shows the statuses of all the currently on-going processes on the assembly line. It is the redesign of a visualization proposed by our target users for monitoring real-time assembly line status. Any delay or faults currently occurring on the assembly line can be observed from the graph (**R5**). Fig. 5 (a) is the original design. It consists of three layers of concentric rotating circles. The inner circle completes one cycle when a product finishes its procedures on one machine. The circle in the middle completes one cycle when the product finishes its procedures on the assembly line. The outer circle completes one cycle for an eight hour work shift. A slower rotation speed of the inner circle means longer cycle time on a station. However, in general it is not considered effective to use the speed of movement to encode data. Furthermore, multiple circles would be needed to display all the products currently on the assembly line, which will be hard to keep track of simultaneously. Hence, we propose a redesign of the visualization.

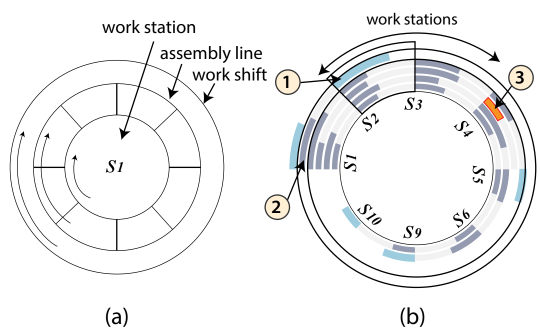

Fig. 5. (a) The original radial design proposed by the target users with three concentric rotating wheels. (b) The redesign we proposed: ① each concentric circle represents a product, the highlighted product is currently being processed on station s_2. light blue color represents ongoing processes on a station; ② lengths of the bars represent how long it took for a part to finish its process on a station; ③ fault occurs.

Fig. 5 (b) shows the new radial visualization we propose. Each of the concentric circles corresponds to a part that is currently being processed on the line. The circles are divided into sectors, each corresponding to a station. The stations are linearly arranged around the circle based on their order within the assembly line. The lengths of the dark blue arcs on the circles reflect the time it takes for the part to finish on the corresponding stations. The light blue arc indicates that the product is currently on the corresponding station, and its length reflect how long the product has been on the station. The lengths increases as time goes by. The faults are color coded, and the color mapping is consistent with that of the extended Marey's graph. The visualization is updated through animated transition, showing real-time information.

Overall, the redesign applies a more principled usage of visual variables to display the key attributes, while still reflecting the mental model that the manufacturing process is cyclic (**R8**). Therefore it is easier for the users to understand.

3D Station Visualization We further show the physical models of the stations in the assembly line in an explorable 3D visualization (**R6**). The exteriors of the stations are displayed transparently and components within the stations associated with the occurrences of the faults are highlighted. The operators can quickly locate the problematic areas by viewing the 3D visualization (**R1**). Fig. 1 (G) shows the 3D view of a station. The 2D radial display serves as a mini-map to support flexible exploration of the 3D scene. The user can select any station for a close-up view by clicking on the corresponding sector on the radial display. The camera will move smoothly and the station at focus will fill the view port, showing more details of it, and highlighting the component causing the fault. This will help the users associate the abstract process data with the physical context where actual operations are carried out.

5.3 User Interaction

The prototype features a rich set of user interactions for exploratory data analysis.

Detail-on-demand The calendar view, the timeline, and the extended Marey's graph form a hierarchical structure for the exploration of temporal data at different levels of detail (**R7**). Users can also zoom in and zoom out on the time scale of the extended Marey's graph by scrolling up and down the mouse wheel. Zooming in shows higher temporal resolution and enables more precise reading of the time when the parts are being processed on each station. Zooming out shows the process data over a longer time span for overview. When the mouse hovers over the visualizations, detailed information will be displayed in tooltips: in the extended Marey's graph it is the serial numbers of the products and the fault codes; in the calendar view it is the statistics of the day in focus.

Brushing and comparative analysis of cycle times Users can select a set of records from the extended Marey's graph by drawing a line on the visualization and all the traces intersecting with it will be selected. The cycle times of the selected set of records are compared to the baseline distributions by overlaying histograms in the small multiples. The baseline distributions are computed from the entire dataset. A significant deviation from the baseline distribution on any of the stations would indicate potential problems worth looking into. Besides that, users can also use this method to verify the results of the outlier detection algorithms.

Brushing and labeling A set of records selected by brushing can be labeled by users as normal records, as input to the outlier detection algorithm based on label propagation. Users can add or remove the labels in a pop up menu opened by right click.

6 System Architecture & Implementation

Fig. 7 illustrates the architecture of the system. We use a relational database to store the manufacturing process data, and indexed the data by timestamps to support the efficient retrieval of data that falls within a specified time interval. The data analysis module performs three tasks: 1) compute summary statistics used in the visualizations in advance and cache the results for faster response time, 2) detect outlier processes, and 3) aggregate the normal processes based on temporal proximity. The user can interact with the historical data visualizations to specify quantile ranges or label normal processes to guide the outlier detection algorithm.

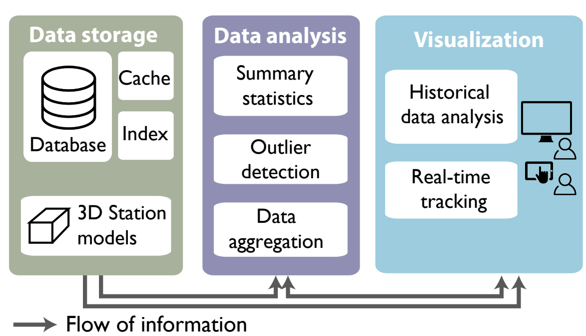

Fig. 7. System architecture.

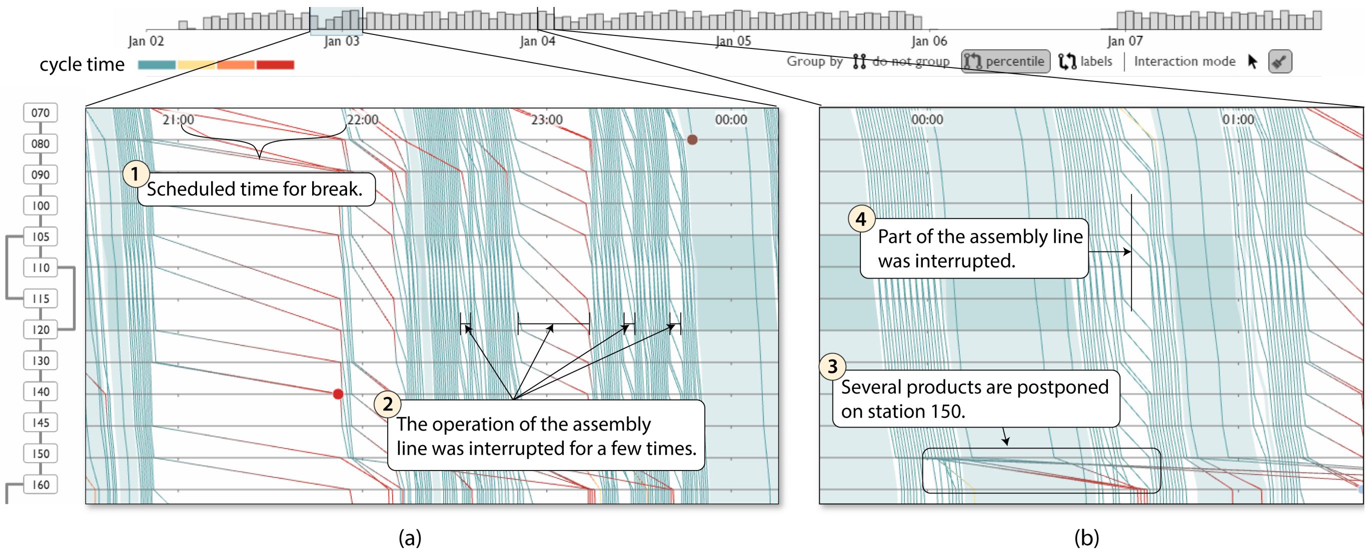

Fig. 6. Detect inefficiencies and perform troubleshooting with the extended Marey's graph: (a) After a scheduled break ①, the assembly line stopped and restarted for a few times before operating smoothly ②; (b) The processing of several products were postponed on station 150 ③ and when those products continued, the other products had to wait on the assembly line ④. In both figures, the outliers are detected and the records are aggregated with quantiles brush set to the range [0, 0.97].

We implement a web application so the target users can access the visualizations more easily on different types of devices and platforms without any native software package installation. The front-end visualization is implemented with a combination of HTML5, CSS, JavaScript, the JavaScript Data-Driven Documents (D3) library [7], the Three.js [1] WebGL library (for 3D model rendering and faster 2D rendering), and several JavaScript framework & utility libraries including Underscore.js [2], Backbone.js[3] and JQuery[4].

The back-end of the prototypes runs on a Python web server built with Flask [5] and Sqlite. We use the label propagation algorithm implemented in scikit-learn [6], a Python machine learning library, for interactive outlier detection. Statistics such as the daily number of productions and faults, and the quantiles of the cycle time at each station are precomputed and cached in advance. Our prototype works at an interactive rate for real world manufacturing data with millions of products per year when running locally on a main stream desktop machine.

7 System Evaluation

We performed two assessments on the system. First, we conducted case studies that illustrate the effectiveness of the system for visual diagnostics of assembly line performance based on historical and real-time data. Then, we conducted a pilot study and had in-depth interviews with managers and operators from manufacturing sites. The data used in the case studies and the user interviews are provided by our target users.

7.1 Case Studies

7.1.1 Detect Inefficiencies and Perform Troubleshooting with Extended Marey's Graph

Several patterns were identified by the users when they use the extended Marey's graph to explore the manufacturing process data.

Fig. 6 (a) shows that between 21:00 and 22:00, the entire assembly line stopped for approximately one hour. This one hour was the scheduled time for break as noted by the users. After the scheduled time for break, the production line didn't come up to speed immediately, and experienced several glitches. It stopped completely and restarted for a few times before operating smoothly from 00:00. This kind of pattern occurred frequently in the assembly line as observed by the users.

Fig. 6 (b) shows that around 00:00, the processing of many products were postponed on station 150. When they continued to be processed on station 150, the other products already on the line had to wait, and could no longer proceed down the assembly line. It thus appeared that part of the assembly line was stopped for five to ten minutes between 00:00 and 01:00. From both (a) and (b), and the data from other time intervals, the users observed that station 150 had triggered many delays and inefficiencies in the manufacturing process. It would be beneficial for the operators and managers to investigate further about the root causes, and come up with solutions to reduce the delays and improve the overall throughput of the assembly line.

To highlight the abnormal records for troubleshooting the inefficiencies, in both of the two figures (a) and (b), a quantile range $[0, 0.97]$ was selected. The quantile range defined the normal cycle times on each station. Processes with longer than normal cycle time on any of the stations were classified as outliers and displayed as individual polylines. The others were aggregated and displayed as thick bands.

Overall, we find that the visualization has great potential to uncover the inefficiencies in the manufacturing process and can point to important opportunities about when, where and how the efficiency can be improved.

7.1.2 Access the Effect of Faults

Since the occurrences of faults are plotted on the time axes in the extended Marey's graph, it is relatively easy for users to associate them with the manufacturing records in close temporal proximity, and assess the causes and effects of those faults. As illustrated in Fig. 8, the users observed that when faults like "weld position 6 velocity upper limit exceeded" occurred on station 050, the affected products were no longer processed on the assembly line. After frequent occurrences of this fault, the entire assembly line would stop for approximately ten minutes before continuing operation.

The frequent sequential co-occurrences of the two events, i.e., the fault and the pause of the entire assembly line, pointed to potential causal relations. Predictive analysis thus became possible based on such observations, as users could anticipate what would follow after the occurrence of a particular fault. The managers and operators could then take preventive measures to prevent losses based on the prediction.

[1]http://threejs.org/
[2]http://underscorejs.org/
[3]http://backbonejs.org/
[4]http://jquery.com/
[5]http://flask.pocoo.org
[6]http://scikit-learn.org/stable

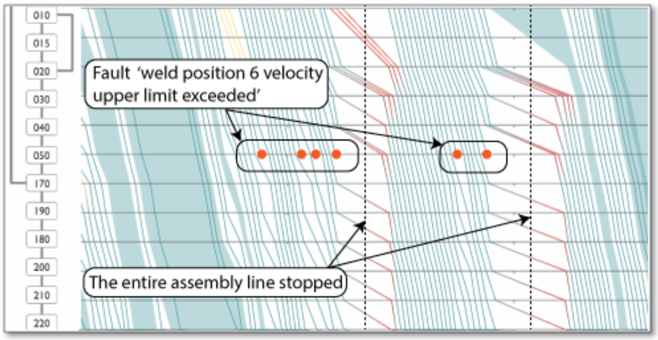

Fig. 8. Occurrences of faults and their effects on the operation of the assembly line. The affected products are no longer processed and the entire assembly line stopped for around ten minutes after frequent faults.

7.1.3 Interactively Identify Outliers with Samples Brush

Fig. 9 shows how users interactively identified the outliers by specifying a set of sample normal processes. The user brushed a set of records on the unaggregated graph, and labeled those as normal processes (Fig. 9 (a)). The system inferred and aggregated the normal processes, and drew the outliers as individual polylines (Fig. 9 (b)). It could be more clearly observed that the occurrence of a fault (colored red, code unknown) had stopped a product from further proceeding down on the assembly line.

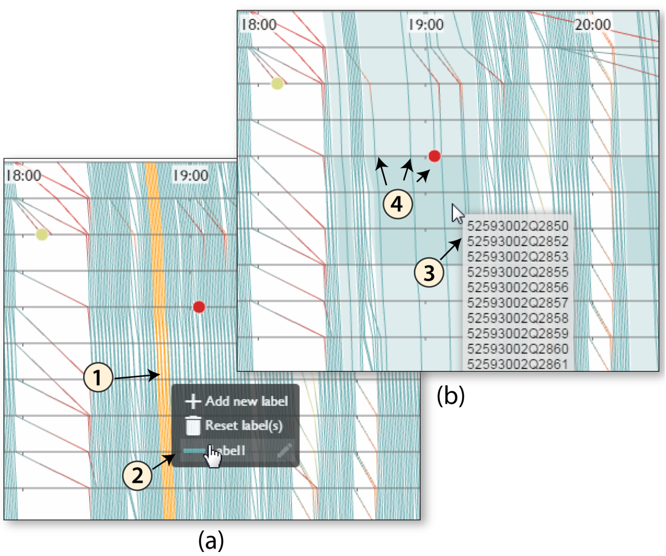

Fig. 9. Identify outliers by specifying sample normal processes: ① brush a set of records; ② label them as normal; ③ a group of normal processes detected by the label propagation algorithm; ④ outlier processes.

7.1.4 Explore Historical Data in Different Temporal Scales

The calendar based visualization shows that in the second half of the year (Fig. 1 (B)) there were more work shifts scheduled on weekends. The user selected a few days and more information about the rate of production was displayed at a finer temporal resolution (Fig. 1 (C)). During certain hours the throughput of the assembly line was lower compared to others, and any abnormalities like this could be further investigated in the extended Marey's graph (Fig. 6).

7.1.5 Track Real-Time Performance with the Radial Graph

When the radial graph was demonstrated to the users, they immediately identified that sometimes two or more products stayed at the same station (Fig. 10) on the assembly line. They commented that the extra products were not moved to the next station in a timely manner, which would affect the performance of the assembly line.

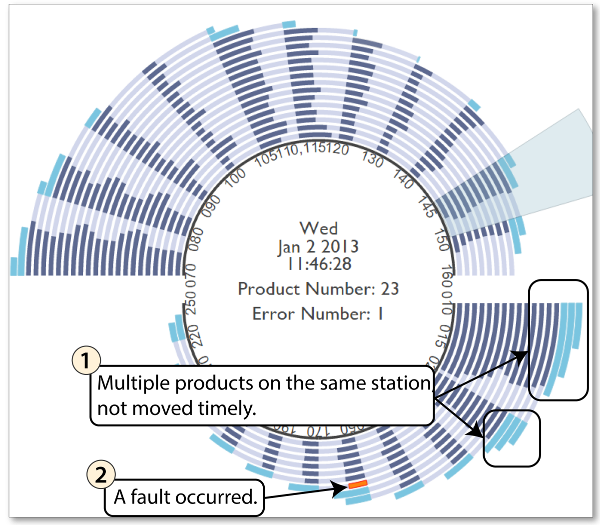

Fig. 10. Radial graph shows ① multiple products stayed at the same station and were not moved in a timely manner to the next station on the assembly line, and ② a fault occurred on station 050.

7.2 User Interview

We interviewed the target users of the system to validate the design decisions, and assessed the effectiveness and learnability of the system.

Before presenting the prototype to a larger group of users, we conducted a pilot study with our target users from manufacturing sites. Two users from two different manufacturing plants were involved in the pilot study. The purpose was to identify potential usability issues to refine the system accordingly, and to gather initial feedback on the system features. We introduced the visual designs and the interactions in the system. After that, they further explored the system on their own. We encouraged them to think aloud during the process.

They were impressed by the system, commenting that the radial design and the extended Marey's graph were very intuitive representations of the manufacturing process data, and were very easy to understand. They liked the functionalities to interactively explore year long data by selecting on the calendar visualizations and brushing the timeline. One of the users also noted that the schematic diagram (Fig. 1 (E)) shows very clearly the structure of the assembly line. After we demonstrated the interactive functionalities for detecting outlier processes and visualized them in the extended Marey's graph, he commented that these features are very helpful to identify the abnormalities. They regarded the visual analytics functionalities as a form of data mining to some extent (they were also collaborating with a data mining team on this project). For the next step, they would like to see it deployed in real manufacturing environments.

We also identified several usability issues throughout the pilot study. For example, initially, the real-time tracking panel and historical analysis panel were displayed simultaneously on a single screen, thus the screen space assigned to the radial graph was very small. We later added the options to maximize the radial graph, and eventually made both panels full screen size with switches for changing between them.

After the pilot study, we further interviewed a larger group of users consisting of 11 operators and managers from manufacturing sites, to have detailed assessments of individual components in the system. The users are familiar with basic visualization techniques such as bar charts and line charts. We installed the server locally on one of the machines in the plant and introduced the system. Then we conducted a semi-structured interview guided by a set of questions. The questions covered various aspects including the learnability of the visual designs and the interactive features, the informativeness and intuitiveness of the system, and the improvements to be made.

For the overall system, the users commented that *"it's very effective in the system's ability to show real-time data and highlight abnormalities"*, and *"it will be useful to see it in action in the active environment"*.

They liked the extended Marey's graph a lot, and one person commented that more process related data can be encoded in the graph: *"Marey's graph would be good to be able to further manipulate other process data for the specific parts, or to link to additional process information. "*.

Between the two interactive outlier detection methods, the samples brush is slightly better received by the users, probably because it can be interpreted more intuitively compared to the quantile brush.

For the multi-scale temporal exploration features, they commented that *"it's very intuitive to navigate between items in different time frames"*.

Many users commented that the 3D station visualization can be further improved. One user suggested that we can add a top down overview of the entire assembly line in the 3D visualization.

They also saw a lot of potential in the current prototype :

"It is interesting, I can see where more uses could continue to be generated from this platform." "This is a good interface for gaining an intuitive picture of how the line is running. These same methods could be applied to process parameters during the manufacture of parts giving engineers the intuitive picture of process stability"

Overall, the results are encouraging. Although we are unable to conduct a controlled user study due to the lack of comparable systems, we plan to conduct long term studies, and record the users' experience with the system as the deployment of the system is under discussion.

8 DISCUSSION

Lessons Learned When reflecting on the design choices, we think that familiarity of the visual metaphor and intuitiveness of the visual encodings play crucial roles for the users to quickly familiarize themselves with the visualizations. Moreover, advanced analytic methods incorporated in the system should be explained in an intuitive manner to the users. For example, the label propagation algorithm can be explained as polylines with similar shapes to the specified examples are considered as normal records, and the user can immediately understand it in this way. Besides that, in the system, we decide to include both the extended Marey's graph and the radial graph to encode similar information (i.e., cycle times and faults) for different purposes: one for analyzing a large amount of historical data, and one for monitoring real-time conditions. Such scenario arises in many application domains with streaming data. In these scenarios, the visualizations need to be tailored for different uses even for data with same attributes.

As we later reflect upon the design process, we consider that a crucial step is identifying the variants and the invariants in the data as described in Section 3. Usually the domain experts are quite familiar with the invariants (i.e. the production process as described by the DAG), and it is not necessarily helpful developing visualizations for such information. To distinguish between the variants and invariants, it is helpful to have a quick analysis of the data attributes or consult with the domain experts first.

General Applicability Although many visualization and interactive techniques presented in the system are tailored to the specific application domain, we believe that some components can be easily adapted and be applied to other use cases. For example, it is not difficult to image that the two interactive outlier detection techniques, based on brushing quantiles and the label propagation algorithm, can be easily adapted for boarder application domains that use high dimensional data. More importantly, we found that the manufacturing process data as described in Section 3 is being collected in many assembly lines. The prototype system can thus be applied to visualize and analyze data from many manufacturing plants, not limited to the ones we are currently working with.

Limitations There are several limitations of the current system. First, although both outlier detection algorithms including brushing quantiles and label propagation can return the results in real-time for the data displayed in the extended Marey's Graph, they cannot be easily scaled to year long data which could contain millions of product records. Improving the scalability of the two algorithms is very much desirable, as the site managers would like to immediately know how many abnormal records there are on each day in the calendar visualization when they update the quantile ranges and specify sample normal records. Second, the extended Marey' graph can not effectively depict the data over relatively longer time span in a display with limited width, as the traces will all become vertical lines. This is the reason that we introduced the calendar and the timeline for multi-scale temporal data browsing. In the future, we would like to improve the visual encoding such that it can show rich information about the delays and faults in long term data. Third, the current system is fine-tuned to fit a screen with 1920 $\times$ 1080 resolution. However more adaptive layout mechanisms of the visual components should be incorporated in the system such that the users can access it from devices with different screens. Last but not least, in the current prototype the subprocesses and parallel processes are overlaid on the same graph, and this can introduce undesirable visual clutter. This problem is alleviated to a certain extent by introducing user interactions for selecting and filtering the routes the products take on the assembly line. If the complexity of the manufacturing processes increase further, the current prototype needs to incorporate more advanced filtering and aggregation functions for scalability.

Future Work There are several directions for future work. First, as the deployment of the system in real production lines is in plan, it becomes possible to study the long-term usage of the system. Methods such as automated logging of user activities and observational study can be applied, to gather usage data about how visualization is received in real working environments. Second, we plan to improve the scalability of the system as discussed in the limitations. Third, the occurrence of individual outlier records are atomic events, based on which we can define composite events. For example, the occurrence of a fault and the delays following it together can be considered as a composite event. We plan to further our investigation to develop techniques facilitating the identification of such events, in order to support predictive analysis on the data.

9 CONCLUSION

In this paper, we present a novel visual analytics solution targeted at the application domain of big data analytics in manufacturing industry. We propose a comprehensive system for the real-time tracking and historical analysis of assembly line performance. It consists of multiple linked views showing data at different levels of detail. In particular, we present the application of the Marey's graph in this domain and extend it to improve its visual scalability. Moreover, we propose two novel interactive techniques for user-steerable outlier detection, which can be potentially applied to more general usage scenarios. The initial feedback from the target users is very encouraging and the deployment of the system in manufacturing sites is being planned. Last but not least, the system is designed and developed for a pilot use case to demonstrate the importance of visual analytics in the application domain of connected industry (industry 4.0). To the best of our knowledge, there is no prior visual analytics research addressing this application domain. We believe that the successful showcase and deployment of the system is a very promising starting point, and will open the door to many challenging research problems.

REFERENCES

[1] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski. Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):47–60, Jan 2008.

[2] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer Publishing Company, Incorporated, 1st ed., 2011.

[3] M. Annunziata and P. C. Evans. The industrial internet @ work. *General Electric Reports*.

[4] E. Auschitzky, M. Hammer, and A. Rajagopaul. How big data can improve manufacturing. *McKinsey & Company*, 2014.

[5] B. Bach, P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. A Review of Temporal Data Visualizations Based on Space-Time Cube Operations. In R. Borgo, R. Maciejewski, and I. Viola, eds., *EuroVis - STARs*, 2014.

[6] N. Bell and M. Garland. Efficient sparse matrix-vector multiplication on CUDA. NVIDIA Technical Report NVR-2008-004, NVIDIA Corporation, Dec. 2008.

[7] M. Bostock, V. Ogievetsky, and J. Heer. D3: Data-driven documents. *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis)*, 2011.

[8] Y.-J. Byon, C. E. Cortés, C. Martinez, F. Javier, M. Munizaga, and M. Zuniga. Transit performance monitoring and analysis with massive gps bus probes of transantiago in santiago, chile: Emphasis on development of indices for bunching and schedule adherence. In *Transportation Research Board 90th Annual Meeting*, number 11-2233, 2011.

[9] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[10] R. Drath and A. Horch. Industrie 4.0: Hit or hype?[industry forum]. *Industrial Electronics Magazine, IEEE*, 8(2):56–58, 2014.

[11] P. C. Evans and M. Annunziata. Industrial internet: Pushing the boundaries. *General Electric Reports*, 2012.

[12] D. Gotz and H. Stavropoulos. DecisionFlow: Visual Analytics for High-Dimensional Temporal Event Sequence Data. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1783, 2014.

[13] D. Gotz, S. Sun, and N. Cao. Adaptive contextualization: Combating bias during high-dimensional visualization and data selection. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, IUI '16, pp. 85–95. ACM, New York, NY, USA, 2016.

[14] M. P. Groover. *Automation, Production Systems, and Computer-Integrated Manufacturing*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd ed., 2007.

[15] R. Hranac, J. Kwon, M. Bachmann, and K. Petty. Using marey graphs to visualize transit loading and schedule adherence. In *Transportation Research Board 90th Annual Meeting*, number 11-0350, 2011.

[16] J. Jo, J. Huh, J. Park, B. Kim, and J. Seo. Livegantt: Interactively visualizing a large manufacturing schedule. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2329–2338, 2014.

[17] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group*. Forschungsunion, 2013.

[18] J. Krause, A. Perer, and H. Stavropoulos. Supporting iterative cohort construction with visual temporal queries. *IEEE transactions on visualization and computer graphics*, 22(1):91–100, 2016.

[19] E. Mai, M. Backman, and R. Hranac. Visualizing bus schedule adherence and passenger load through marey graphs. In *18th ITS World Congress*, 2011.

[20] K. Matković, H. Hauser, R. Sainitzer, and M. E. Gröller. Process visualization with levels of detail. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pp. 67–70, 2002. doi: 10.1109/INFVIS.2002.1173149

[21] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal Event Sequence Simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227, 2013.

[22] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928, Nov. 2009.

[23] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.

[24] C. Palomo, Z. Guo, C. T. Silva, and J. Freire. Visually exploring transportation schedules. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):170–179, 2016.

[25] A. Perer and F. Wang. Frequence: interactive mining and visualization of temporal frequent event sequences. In *Proceedings of the 19th international conference on Intelligent User Interfaces*, pp. 153–162. ACM, 2014.

[26] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: visualizing personal histories. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 221–227. ACM, 1996.

[27] J. Posada, C. Toro, I. Barandiaran, D. Oyarzun, D. Stricker, R. De Amicis, E. Pinto, P. Eisert, J. Döllner, and I. Vallarino. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. *Computer Graphics and Applications, IEEE*, 35(2):26–40, Mar 2015.

[28] M. Sedlmair, M. Meyer, and T. Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[29] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pp. 336–343, Sep 1996.

[30] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, Cheshire, CT, USA, 1986.

[31] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 457–466. ACM, 2008.

[32] C. Ware. *Information visualization: perception for design*. Elsevier, 2012.

[33] K. Wongsuphasawat and D. Gotz. Exploring Flow, Factors, and Outcomes of Temporal Event Sequences with the Outflow Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2659, 2012.

[34] M. Wörner and T. Ertl. Visual Analysis of Advanced Manufacturing Simulations. In S. Miksch and G. Santucci, eds., *EuroVA 2011: International Workshop on Visual Analytics*. The Eurographics Association, 2011.

[35] E. Zgraggen, S. M. Drucker, D. Fisher, and R. DeLine. (s—qu)eries. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, 2015.

[36] J. Zhao, Z. Liu, M. Dontcheva, and A. Hertzmann. MatrixWave: Visual Comparison of Event Sequence Data. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, 2015.

[37] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, June 2002.