ViDX: Visual Diagnostics of Assembly Line Performance in Smart Factories

Panpan Xu, Honghui Mei, Liu Ren*, and Wei Chen

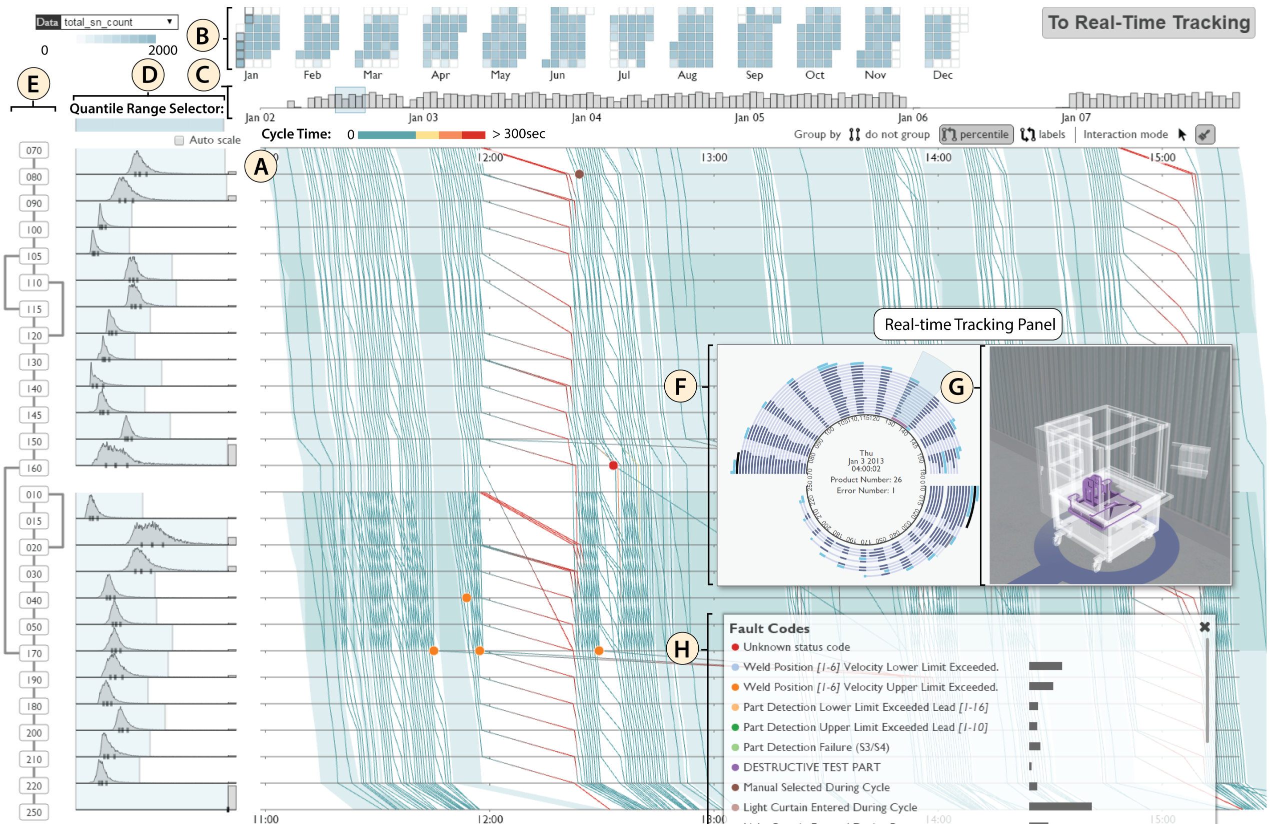


Fig. 1. A screenshot of the ViDX system for historical analysis and real-time tracking of assembly line performance. The historical data analysis panel consists of an extended Marey's graph (A) for troubleshooting inefficiencies and faults on the assembly lines. It is linked with a calendar based visualization (B) and a timeline (C) for multi-scale temporal exploration. Supplementary views include small multiples of histograms (D) showing the distribution of the cycle times on each station and a map (E) showing the assembly line schema. The real-time monitoring panel consists of a radial graph (F) and an explorable 3D station model visualization (G). (H) shows the fault codes.

Abstract— Visual analytics plays a key role in the era of connected industry (or industry 4.0, industrial internet) as modern machines and assembly lines generate large amounts 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 collected in the manufacturing processes. We report the design and implementation of a comprehensive visual analytics system, ViDX. It supports both real-time tracking of assembly line performance and historical data exploration to identify inefficiencies, locate anomalies, and form hypotheses about their causes and effects. 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 levels of detail. In particular, we apply and extend the 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 the 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 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

*Liu Ren is the corresponding author. Panpan Xu and Liu Ren are with Bosch Research North America. E-mail: {panpan.xu, liu.ren}@us.bosch.com. Honghui Mei and Wei Chen are with Zhejiang University. E-mail: {meihonghui, chenwei}@cad.zju.edu.cn.

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

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 mining industries [4], where 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 that 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 producing automotive parts to develop a visual analytics system that can support real-time tracking of assembly line performance and historical data analysis.

Assembly lines on the shop floor consist of sequences of work stations. Each station corresponds to a stage of production where specific procedures are carried out on the products. The products (automotive parts) are moved through the stations, tested, and shipped out to car manufacturers. During the operation of the assembly line, data is recorded about when the product is moved from one machine to the next and about any fault that occurred during the process. This type of setting is becoming increasingly common in 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.
- 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 anomalies 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 extended Marey's graph is described in Section 4 and the system is presented 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 [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 manufacturing industry will continue to grow in the future. Visual analytics, an important technique for gaining insight from large and complex data, can therefore play a crucial role in this application domain [19, 27].

So far only a few visual analytics solutions target the data analysis tasks in manufacturing scenarios. Matković et al. [21] 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. Wörner and Ertl [33] 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. Therefore the analytic tasks are fundamentally different from those used for planning and simulation purposes as described above.

2.2 Temporal Data Visualization

Temporal data visualization has been extensively studied in the past years. Temporal dimension can be found in many applications [28]. There are several surveys reporting the state of the 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 there are 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. The operations include 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 work. In particular, the event sequence visualization techniques can be grouped 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. The first category of techniques includes LifeLines [26, 30], Sankey diagrams [12, 22, 25, 32] and Matrix based visualizations [35] for analyzing patient medical records and website visiting patterns. Recently, a few interactive visualization systems have also been proposed for selecting a subset of the event sequences for focused analysis [13, 18, 34]. The second category includes Marey's travel graph [29]. It was first introduced in the 1880s for visualizing train schedules. Since then it has been used extensively to analyze public transportation schedules [8, 15, 20]. Inspired by the design, Palomo 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 enhance Marey's graph with a time-aware outlier-preserving visual aggregation technique to support effective identification of anomalies and inefficiencies in the manufacturing processes. Novel interaction techniques are also introduced, with which the users can interactively identify the anomalies 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 the next to be processed and assembled to form the final **product**. In recent years, there has been a widespread move towards using general-purpose computing devices to control and monitor industrial processes. Programmable logic controllers (PLCs), for example, are widely deployed to control the machineries on the assembly lines for manufacturing automation [14]. The PLCs on the assembly lines will send the status information of the parts to central databases when the parts arrive at the stations.

Assembly lines can be considered as directed acyclic graphs (DAGs). The nodes in the DAGs are the work stations, which we denote as

$S = \{s_i | i \in [1,n]\}$. The edge $(s_i, s_j) \in S \times S$ in the graph indicates that the operation on s_j takes place immediately after s_i. Fig. 2 shows the schematic diagram of an assembly line as a DAG. In this assembly line the parts choose either station s_3 or s'_3 after finishing on s_2 and undergo the same procedure in parallel. On 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 it is moved onto the next. All of these processes can be described by modeling the assembly lines as DAGs.

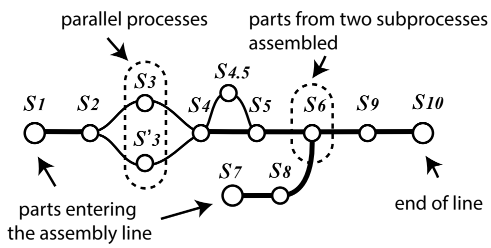


Fig. 2. The schematic view of an assembly line as a directed acyclic graph (DAG). The parts are moved among the stations following predefined paths. On stations s_1 and s_7 two different types of parts enter different sub-processes. On stations s_6 the two types of parts are assembled. Station s_3 and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

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 this, 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 some 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 timestamps when the parts (or products) reach a station 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. It would affect the ability of the manufacturing plant 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 provides 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, perform troubleshooting, and discover opportunities to reduce losses and increase profit.

Therefore, our focus 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 have formulated 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 rights to their production databases. They pointed us to the data that they were most interested in, i.e., the cycle times and the faults in the manufacturing processes. They also 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 later stages. 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 and maintenance of the databases. The design requirements were formulated iteratively throughout the six months.

The following design requirements are identified for historical data analysis:

R1 ***Facilitate the detection of abnormal processes.*** The visual encoding should highlight the abnormal processes and show when and on which stations the delay or faults has occurred. Detecting outliers is the first step to 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 experiences 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 understanding of technical details.

R4 ***Support predictive analysis by associating the anomalies with the surrounding context of assembly line operation.*** The occurrences of the delay and faults may have certain causes and effects. One may observe frequent sequential occurrences of a particular fault and pauses on the assembly line. This indicates potential causal relations between the faults and the pauses. Observations like this can help build a plausible predictive model. The operators and managers can take preventive measures to reduce losses based on the predictions.

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

R5 ***Highlight anomalies in real-time data.*** Similar to historical data analysis, anomalies such as delay and faults should be highlighted such that the operators and managers can respond immediately to 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 stations in physical environments. Furthermore, since the fault codes are related to specific components in the stations, we can highlight those components in 3D models to support troubleshooting.

Besides that, the following requirements are also equally important:

R7 ***Support smooth and interactive exploration of large amounts of process data.*** In the 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 data set, the system should be visually and algorithmically scalable.

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 extremely important to keep the visual designs intuitive and easy to understand. Therefore, we have to make careful design choices considering these aspects.

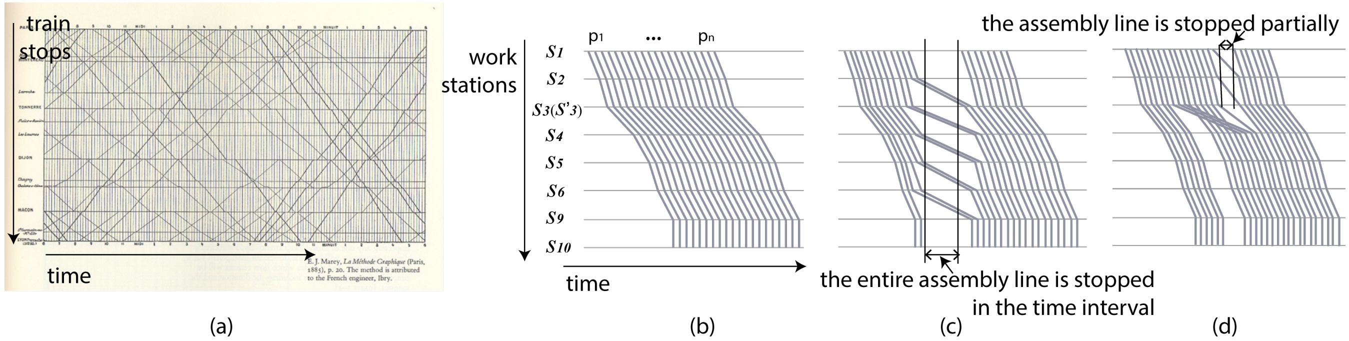


Fig. 3. Marey's graph [29] and the relevant visual patterns when applied to the manufacturing process data. It uses the path $(s_1, s_2, s_3(s_3'), s_5, s_6, s_9, s_{10})$ in the DAG in Fig. 2 for illustration. (a) the original Marey's graph shows the bus/train schedules; (b) Marey's graph showing when 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.

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 direct application of the Marey's graph would result in visual clutter and affect 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 [29] is a traditional method for depicting bus or train schedules. 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 or trains are expected to arrive at a stop (Fig. 3 (a)) based on the schedules.

This visual encoding can be directly applied to manufacturing processes data if we consider each work station on the assembly line as a bus or train stop, and the time when the parts are moved onto each work station as the time in bus or train schedules. The polylines would trace the complete history of a product on the assembly line. 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 in 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 with multiple parts. However it might introduce undesirable line overlap 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 and supports 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 the visual patterns the operators can form hypothesis about the causes of the inefficiency (**R2**). Here we summarize the visual patterns for the users to quickly read high-level semantic information from the visualization.

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

In Marey's graph, the users can identify ***out-of-order processes***, 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. There 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 delay or interruption of operations occur.
- ***Halt of the entire assembly line.*** In Fig. 3 (c), all the processes have experienced delay around a time point 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, waiting for s_3 to finish handling the parts whose processing have been delayed. This type of event is also a source 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 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 and 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, 30], only the cycle times on one station can be compared at a time. Moreover, the temporal context is lost. In Marey's graph, the lengths and angles of the line segments are strong visual cues for the comparison of cycle times even without aligning the starting time. The design invokes the Gestalt rule of similarity: line segments with similar slopes are perceived as a group by the reader [31] and the outliers will stand out (**R1**). Sankey diagram [32] and MatrixWave [35] are other possible ways to visualize event sequence data, although they emphasize the variation on 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, this method estimates the density of the lines and draws a heat map of it. However, it can blur out the anomalies (or outliers), as they usually reside in low density regions of the display. We introduce a method that can reduce visual clutter and in the mean while highlight the outliers, inspired by an approach originally proposed by Novotný and Hauser [23] for reducing the visual clutter in PCPs.

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 displayed as individual polylines and overlaid on top of the aggregated normal processes.

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 it should merge the process to the current group or create a new one. If the process scanned is temporally close to those in the current group (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 processes and still highlight the anomalies. The aggregated processes show the surrounding context for these abnormal processes for troubleshooting (**R4**). The visual patterns we described in the last section are still visible as the related abnormal processes are displayed individually and not hidden from the viewers.

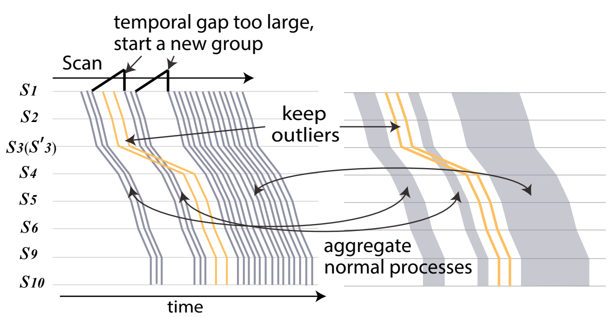


Fig. 4. Time-aware outlier-preserving visual aggregation: the outliers with faults or abnormal delays are identified; the normal processes in close temporal proximity to each other are aggregated and they are represented as thick bands instead of individual polylines.

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 experiences operating the assembly lines (**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., box plots) to give a 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. The graph interactively updates to show a new set of outliers detected as the user selects different quantiles. 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 the information the system can detect the outliers in the remaining data. We integrate the label propagation algorithm [36] 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, since 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 it iteratively propagates the labels along the graph edges, starting from the points with known labels. The iteration stops when the labels of the data points no longer change. The algorithm can be expressed formally as:

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

Where A is the adjacency matrix of the neighborhood graph and L_X codes the labels of the data points (please refer to [36] for more details). The matrix multiplication can be parallelized on modern GPUs for interactive performance [6] on large data sets.

We apply the method to identify abnormal processes based on the samples specified by the users (Fig. 9(Ⓐ and Ⓑ)). First, we construct a k-nn graph of all the processes based on their cycle times on the stations using an 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 and gradually covers 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 Ⓓ). The normal processes are aggregated (Fig. 9 Ⓒ).

The ***quantiles brush*** and the ***samples brush*** both engage users in the computational extensive process of outlier detection (**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" [28]. The display consists of a calendar based visualization, a timeline, and the extended Marey's graph, showing data at different temporal scales with different levels of detail to support the exploration of year long data (**R7**). Fig. 1 shows an overview of the system.

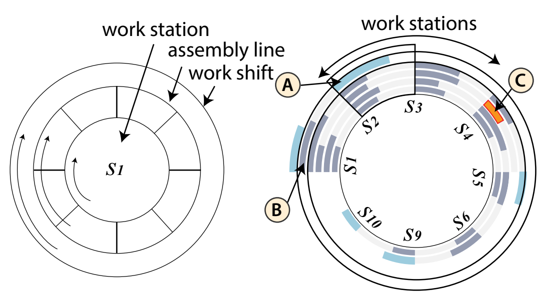
Calendar View The calendar view shows the summary statistics such as the number of products and the number of faults occurred on each day over a year (Fig. 1 (B)). We choose the calendar based display as it aligns the weekdays and weekends for better cross comparison. The user can select a continuous set of days on the calendar. The timeline (Fig. 1 (C)) will then update its range to the selected days and display the number of products in a finer resolution. By brushing the corresponding range on the timeline, the user can investigate the process data 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 ongoing 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 has three layers of concentric rotating circles. The inner circle completes one cycle when a product finishes its procedures on one station. The circle in the middle completes one cycle when the product finishes its procedures on the entire 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 using the movement speed 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.



(a) (b)

Fig. 5. (a) The original radial design proposed by the target users with three concentric rotating circles. (b) The redesign we proposed: Ⓐ 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; Ⓑ length of a bar represents how long it takes for a part to finish its process on a station; Ⓒ 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 assembly line. The circles are divided into sectors. Each sector corresponds 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 station, and its length reflect how long the product has been on the station. The length increases as time goes by. The faults are color coded and the color mapping is consistent with the extended Marey's graph. The visualization is updated through animated transition, showing real-time information.

The redesign applies a more principled usage of visual variables to display the key attributes, while still reflecting the mental model of the users (**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 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 which supports the 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 to the station at the focus, showing more details of it. The 3D view helps the users associate the process data with the physical context where the actual operations are carried out.

5.3 User Interaction

The prototype features a rich set of user interactions beside those already mentioned in the paper.

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 out on the time axes of the extended Marey's graph by scrolling the mouse wheel. When zoomed in, the graph shows higher temporal resolution and enables more precise reading of the timestamps. When zoomed out, the graph shows the process data within 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 will display the ids of the products and the fault codes; in the calendar view it will display 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. All the traces intersecting with it will be selected. The cycle times of the selected records can be 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 on any of the stations would indicate potential problems worth looking into. Users can also use this method to verify the results of the outlier detection algorithms.

6 System Architecture & Implementation

Fig. 7 illustrates the architecture of the system. We use a relational database to store the manufacturing process data and index the data by timestamps to support the efficient retrieval of data 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; 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 and guide the outlier detection algorithms.

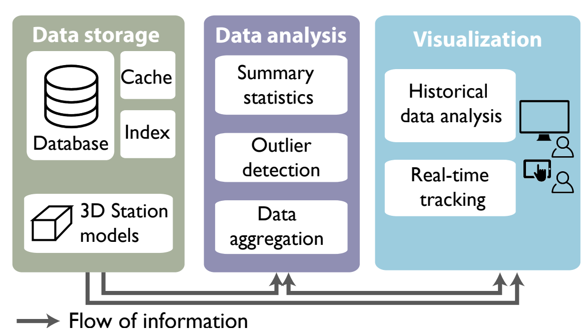
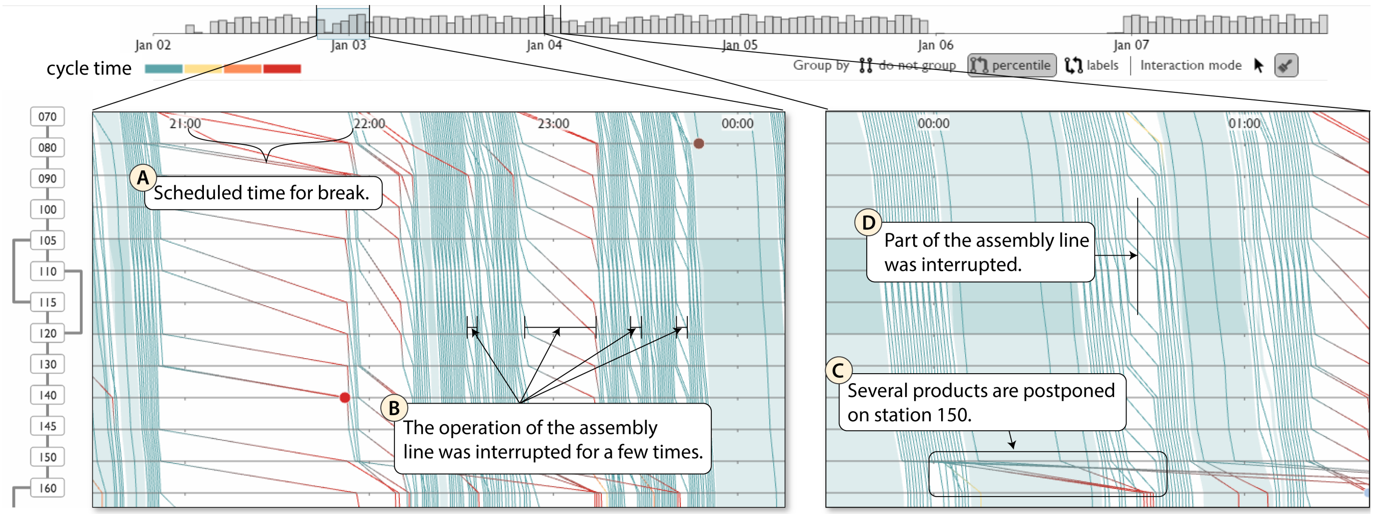


Fig. 7. System architecture.

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

[1]http://threejs.org/



(a) (b)

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

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] 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 for interactive performance. Our prototype works at an interactive rate for real world manufacturing data with millions of products per year when running locally on a mainstream desktop machine.

7 System Evaluation

We performed two assessments on the system. First, we conducted case studies that illustrated the effectiveness of the system for visual diagnostics of assembly line performance. 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 were provided by our target users.

7.1 Case Studies

7.1.1 Detect Inefficiencies with Extended Marey's Graph

Several patterns were identified by the users when they used 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 commented 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 at 00:00. This 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 inefficiencies in the manufacturing process. It would be beneficial for the operators and managers to investigate further about the root causes, come up with solutions to reduce the delays on station 150, and improve the overall throughput of the assembly line.

To highlight the abnormal records for troubleshooting, in both Fig. 6 (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 insights about when and where the efficiency can be improved.

7.1.2 Access the Effect of Faults

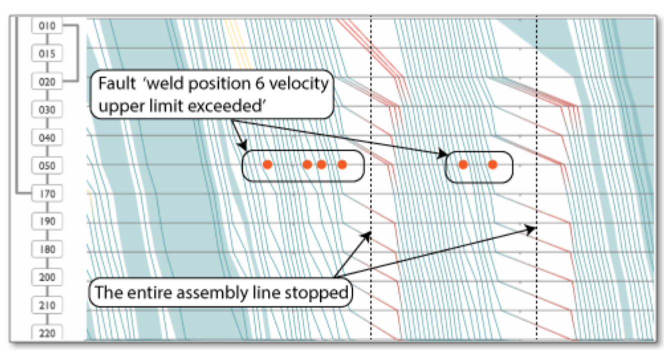


Fig. 8. Occurrences of faults and their effects on the operation of the assembly line. The affected products are no longer processed and the entire assembly line stopped for around ten minutes after frequent faults.

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

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

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 the operations.

The frequent sequential occurrences of the two events, i.e., the fault and the pause of the entire assembly line, pointed to potential causal relations. Such observation provides insights for building predictive models. Users could now anticipate what would frequently follow after the occurrence of a particular fault.

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 processes on the unaggregated graph and labeled those as normal processes (Fig. 9 (a)). The system inferred the normal processes, aggregated them, and displayed 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.

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 the others. Any anomaly like this could be further investigated in the extended Marey's graph (Fig. 6).

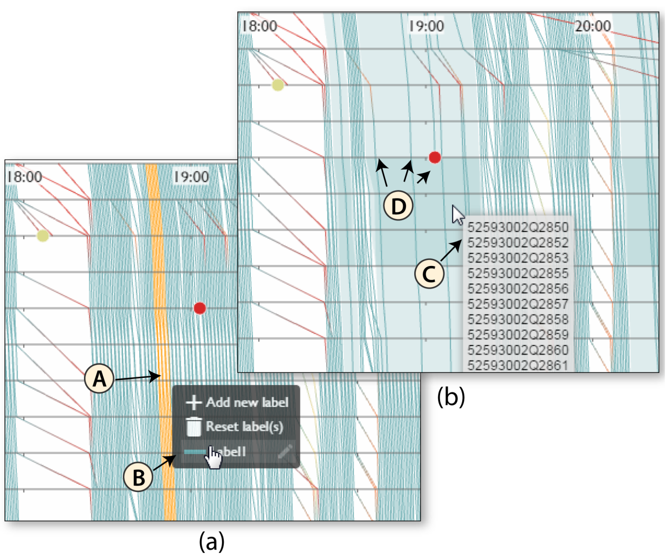


Fig. 9. Identify outliers by specifying sample normal processes: Ⓐ brush a set of records; Ⓑ label them as normal; Ⓒ a group of normal processes detected by the label propagation algorithm; Ⓓ outlier processes.

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.

7.2 User Interview

We interviewed the target users to validate the design decisions and assess the effectiveness of the system.

Before presenting the prototype to a large group of users, we conducted a pilot study with our target users from manufacturing plants. Two users from two different plants were invited to the pilot study. We closely collaborated with one plant and used the real manufacturing process data collected there to develop the prototype. The other plant had similar data and we were planning to adapt the current prototype for it. The purpose of the study was to identify potential usability issues such that we could refine the system accordingly. Besides that, we also planned to collect some initial feedback on the system features. During the pilot study, the two participants explored the system on their own after we introduced the visual designs and the interactions to them. We encouraged them to think aloud during the process.

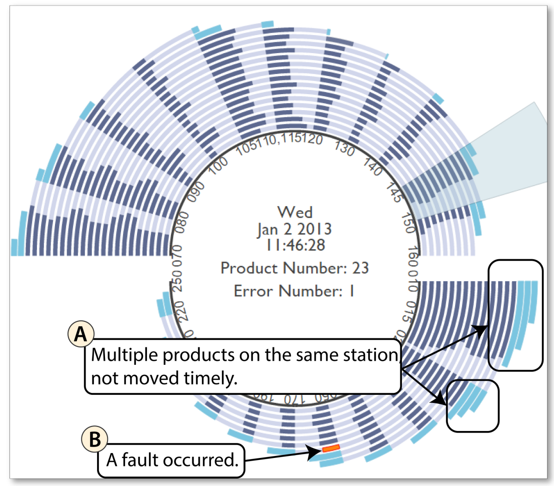


Fig. 10. Radial graph shows Ⓐ multiple products stayed at the same station and were not moved in a timely manner to the next station on the assembly line, and Ⓑ a fault occurred on station 050.

The two participants were impressed by the designs. They commented that the radial graph and the extended Marey's graph were intuitive representations of the manufacturing process data. They remarked that it was very beneficial to be able to explore year long data by simply brushing on the calendar and the timeline. One person also commented that the schematic diagram (Fig. 1 (E)) showed clearly the structure of the assembly line. They regarded the visual analytics functionalities in the system as similar to data mining to some extent (they were also collaborating with data mining experts). Besides that, they also expressed interest in deploying the system in real manufacturing environments during the pilot study.

We also identified several usability issues through the pilot study. For example, initially all the views in the real-time tracking panel and the historical analysis panel were displayed on a single screen. The screen space assigned to the radial graph was not sufficient to emphasize real-time information. We therefore added options such that the user can choose to maximize the radial graph or keep its normal size. Eventually we made the decision to make two separate full screen panels. The users can switch from one to another.

After the pilot study, we interviewed a larger group of users with 11 operators and managers from manufacturing sites to have detailed assessment of the individual components in the system. The users were familiar with basic visualization techniques such as bar charts. We installed the application locally on one of the machines in a plant and introduced the prototype. Then the users tried out the system themselves. After that we conducted a semi-structured interview guided by a set of questions (Table 1). During the interview, we took notes of the users' comments. Some of the users also sent us their feedback via email later. Overall the feedback is encouraging, although we also noticed some limitations of the current system. For example, it was not easy for the users to understand the quantile brush, and the 3D station visualization needed further improvement. The users' comments were summarized as follows.

The users agreed that the extended Marey's graph was easy to read

Table 1. Questions for user interviews.

#	Aim	Question
Q1	Visual Design	Is it easy/hard to learn to read the Marey's Graph? Why?
Q2	Visual Design	Is it easy/hard to learn to read the Radial Graph? Why?
Q3	Interaction Design	Is it easy/hard to learn to detect abnormal processes by selecting percentiles? Why?
Q4	Interaction Design	Is it easy/hard to learn to detect abnormal processes by labeling sample normal processes? Why?
Q5	General	Which part of the visual interface do you think can be further improved? How?
Q6	General	Do you think the system is informative in presenting the data?

and was informative in presenting the process data. One person also commented: *"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 [measurement] information."*. Between the two interactive outlier detection methods, the samples brush was slightly better accepted by the users, probably because it had a more intuitive interpretation compared to the quantiles brush. Many users commented that the 3D station visualization could be further improved. One user suggested that a top down overview of the entire assembly line could be a valuable additional feature.

For the overall system, they commented that *"it's very effective in the system's ability to show real-time data [in the radial graph] and highlight abnormalities"*, *"it will be useful to see it in action in the active environment"* and *"it's very intuitive to navigate between items in different time frames [using the multi-scale temporal exploration feature]"*. They also saw a lot of potential in the current prototype. One person commented: *"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"*. Although we were unable to conduct a controlled user study due to the lack of comparable systems, we planned to perform long term studies and record the users' experience using the system. The deployment of the system was being planned.

8 Discussion

Lessons Learned When reflecting on the design choices, we think that the familiarity of the visual metaphor and intuitiveness of the visual encoding play crucial roles for the users to quickly familiarize themselves with the visualizations. Moreover, advanced analytic methods in a visualization 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. 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 (Section 3). Usually the domain experts are 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 for other use cases. For example, it is not difficult to image that the two interactive outlier detection techniques can be applied to boarder application domains with high dimensional data. More importantly, 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 small amount of data, they can not be easily scaled to year long data with millions of records. It is necessary to improve their efficiency, 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 or 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. The traces will become vertical lines. In the future, we would improve the visual encoding to show the delays and faults in long term data. Third, the current system is fine-tuned to fit a screen with 1920×1080 resolution. More adaptive layout should be incorporated in the system such that the users can access it from different devices. Besides that, in the current prototype the subprocesses and parallel processes are overlaid on the same graph. This can introduce undesirable visual clutter. The problem is alleviated to a certain extent by introducing user interactions for selecting the routes 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 being planned, 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 used 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 to support predictive analysis.

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 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 encouraging and the deployment of the system in manufacturing sites is in plan. 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 research addressing this application domain. We believe that the successful showcase and deployment of the system will be a promising starting point and will open the door to many challenging research problems.

Acknowledgments

We would like to thank Kenneth Podd and his colleagues from Bosch Plant in Anderson in the U.S. for the helpful discussions and user study support. In addition, we would like to thank Race Oswald and Alex Makarov for preparing the 3D models, editing the video and proofreading the paper. Finally, we thank the VAST reviewers for their valuable feedback. The work was performed while Honghui Mei was an intern at Bosch Research North America. Wei Chen is supported by 973 Program of China (2015CB352503), NSFC (61232012, 61422211).

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. D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 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):1783–1792, 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] S. Liu, W. Cui, Y. Wu, and M. Liu. A survey on information visualization: recent advances and challenges. *The Visual Computer*, 30(12):1373–1393, 2014.

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

[21] 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.

[22] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal Event Sequence Simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227–2236, 2013.

[23] M. Novotný 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] 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.

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

[30] 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.

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

[32] 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.

[33] 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.

[34] E. Zgraggen, S. M. Drucker, D. Fisher, and R. DeLine. (s—qu)eries: Visual regular expressions for querying and exploring event sequences. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*, 2015.

[35] 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.

[36] 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.