EuroVis Workshop on Visual Analytics (2013)
M. Pohl and H. Schumann (Editors)

Dataflow-based Visual Analysis for Fault Diagnosis and Predictive Maintenance in Manufacturing

M. Wörner[1,2], M. Metzger[1], and T. Ertl[1]

[1]Institute for Visualization and Interactive Systems, University of Stuttgart, Germany
[2]GSaME Graduate School for advanced Manufacturing Engineering, University of Stuttgart, Germany

Abstract

Predictive machine maintenance, which monitors the current condition of a machine, can be much more efficient than maintaining it on a strict schedule or only as a reaction to actual breakdowns. Although sophisticated theoretical models exist, these are not always employed in practice, presumably in part due to their abstract nature. Introducing interactive visualization into the analysis process may facilitate the adoption of predictive maintenance. We apply a dataflow-based visual analytics approach to the analysis of diagnostic machine data on a real-world dataset and collect feedback from domain experts.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications—

1. Introduction

Machine tools are important elements of manufacturing processes. A machine tool is a device that uses some kind of tool to cut or drill a metal work piece. It usually follows a computer controlled path to shape a raw metal block according to a CAD created design. They are high precision devices, made of many parts, equipped with numerous sensors, and controlled by complex electronics and software. They are also very expensive, so anyone investing into a machine tool has high expectations regarding its availability and reliability. Consequently, the precise diagnosis of actual and potential problems is an important issue for machine tool manufacturers, both for checking new machines before shipping them to the customer and for diagnosing machines during maintenance. Diagnosing a machine involves one or more tests, in which the machine performs certain tasks while recording various sensor readings. Viewing visualizations of these sensor readings, an expert will usually be able to judge whether certain parts and elements behave as expected or need to be repaired or replaced.

Visual analytics can make this process more effective by offering a more systematic and partly automatic approach. It can also enable non-experts to perform the diagnosis based on pre-configured automatic analysis steps. We present a visual analytics system we designed to demonstrate this approach. We were able to collaborate with a machine tool manufacturer, who supplied us with data sequences to be analysed, with exemplary expert assessments of some of these data sequences, and with subsequent feedback on the results and their assessment of the utility of the visual analytics method in this particular context.

2. Related work

Monitoring the condition of a machine has always been an important topic in the manufacturing domain. There are sophisticated approaches for fault diagnosis and predictive maintenance. These include wavelet transformations [WGY12] and machine learning techniques such as principal component analysis and support vector machines [TZY*10]. Some algorithms do not require the installation of dedicated diagnostic sensors but can operate on signals a machine generates during its normal operation [VHWM09]. Effort has been put into including diagnostic algorithms on a chip in the machine [OGC*10]. These methods strive for an automatic detection and diagnosis of problems. They do not require human intervention and have no need for an interactive visual interface. However, in an extensive review of condition-based maintenance [JLB06], Jardine et al. note that in spite of the availability of advanced maintenance techniques in the literature, it is still common in the industry to either simply maintain machines on a regular schedule or wait until a breakdown occurs. They list several possible reasons, among these a "lack of efficient communication between theory developers and practitioners". We believe that an explorative

DOI: 10.2312/PE.EuroVAST.EuroVA13.055-059

visual analytics approach may help to create a better understanding of available data collections and the opportunities modern analysis methods create. Using a combination of visual and automatic analysis may allow experts to judge visual data representations and determine patterns and correlations that can be used by automatic methods to generalize these findings to other measurements.

3. The data

These particular machines collect data during test runs. In a test run, an operator performs a defined task on the machine while the machine continuously records the values of several sensors. The data we received for analysis was from a ball screw test. This machine part converts the rotational movement of a rod into a precise linear movement, for example to position the tool at the work piece. The machine can measure the current position of the ball screw in two ways: Directly, using markings on a measuring rod, and indirectly, counting revolutions of the motor driving the ball screw. Over time, wear will reduce the precision of the positioning and thereby the quality of the parts manufactured by the machine. The analyst's task is to determine whether a given set of measurements indicate a defect and a necessity to replace the ball screw. During the test run, the ball screw head is moved forth and back over its entire range at a constant speed. Six sensors were recorded during this test. These include the nominal position, the difference between the nominal and the actual position, the velocity of the movement, and the difference between the direct (measuring rod) and indirect (motor revolutions) position measurements. There were 40 runs for this test. Every sensor was sampled at 167 Hz and recorded a total of 8192 numeric values (about 49 seconds). The machines sample all sensors simultaneously, but there is typically a gap between the start of the recording and the start of the test as both are triggered manually.

4. The analysis system

Our task is to find ways to estimate the state of a ball screw given a set of test run. Domain experts can give us evaluations for some of the test results, which can serve both as a starting point for discovering what differences in the data may prompt these evaluations and as ground truth for calibrating the automatic analysis. In general, the domain experts will come to their decision by looking at data plots and judging them from their experience. This evaluation is mostly a black box for us. However, experts may also be able to point out specific characteristics in the data, which may give us some insight into their decision process and suggest ways to convert a given data series into key figures that can be used for classification. This approach is highly explorative. In order to clearly communicate what steps exactly a proposed analysis involves, how the raw data is processed by

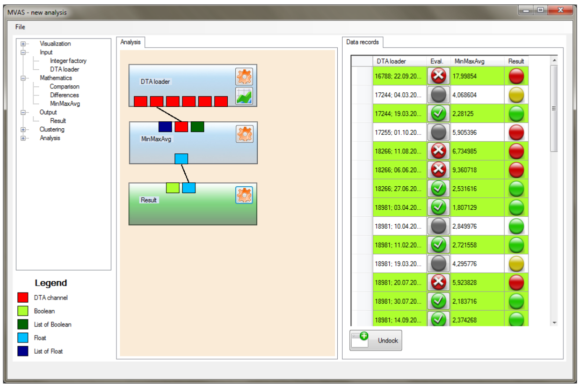

Figure 1: *The main window of the prototype implementation: Available components on the left, analysis graph in the centre, test runs and component results on the right. Green rows indicate cases in which the manual classification (second column) and the automatic classification (fourth column) match.*

these steps, and how this eventually results in the final classification, we use a dataflow graph to represent a particular analysis. This is an interactive visualization of the analysis from loading the data to calculating the classifications. In the long term, once domain experts have designed a graph for a particular analysis problem, non-expert users may be able to reuse this graph to perform analysis tasks on their own. Experts would only have to update the graph when the analysis task changes (such as when there is a modification to the machine or sensor design) or users report incorrect classifications. There are tools available that use a graph-based representation of an analysis process, such as Knime [BCD*09] or RapidMiner [MWK*06], but for this particular project, we chose to create our own prototype implementation for maximum flexibility.

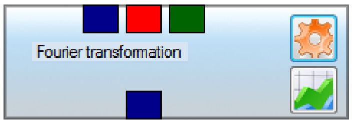

Figure 2: *A node representing the Fourier transformation component. It has both a settings button (top right) and a visualization button (bottom right). Three input ports at the top accept the data to be analysed as either a list of floating point values (blue) or a data channel (red) and a list of Boolean values (green), which can be used to select only a part of the input for analysis. One blue output port at the bottom provides a list of floating point values (the spectrum).*

The main window of this prototype (Figure 1) is divided into three parts. The left part of the window shows the available analysis components. The central part contains an interactive visualization of the current graph, which the user can edit by moving nodes around or connecting them to other nodes. Each node (Figure 2) represents an analysis component and has a certain number of input and output ports. The colour of a port indicates its data type. Ports of the same colour can be connected to form a dataflow through the graph. All nodes have a settings button, which can be used to specify parameters for this particular analysis step. Some nodes also show a visualization button, which will open a visual representation of relevant data. To the right of the graph view, there is a list of all recorded test runs showing the machine identification and the date of the run in the first column as well as the pre-classification as "good" (green tick), "bad" (red cross), or "indetermined/borderline" (grey circle) in the second column. Nodes can add columns to this list to display their results. As an example, the third column in Figure 1 displays a numeric result, the fourth the automatic classification in the form of a coloured circle.

Usually, the first component to be added is the Data Loader component. Its parameters define which test runs are to be read from these files. The corresponding node has one output port per sensor, each providing a data channel. A data channel contains the set of all time series of values recorded by a single sensor in the selected test runs. The node has a visualization button, which will open a plot of the loaded data. Here, the user can examine the various channels and measurements or compare plots of different machines. We draw these diagrams using the open source graph drawing library ZedGraph [zed]. Another central component is the Results component. Its input port accepts a Boolean value for each test run, giving the classification of this test as "good" or "bad". A second input port accepts a float value instead to allow for a fuzzy classification of test cases.

5. Analysing the data

We started out with a small set of test runs, some of which had been pre-classified by the experts. Our first step was to add a Data Loader node to the analysis graph that reads these test runs and lists them in the right part of the main window. We then marked them as "good", "bad", or "indetermined" according to the experts' assessment. When we compared plots of good and bad cases in the visualization of the Data Loader node, we noticed that an obvious difference appeared to be that the deviation between the nominal and the actual ball screw head position is generally larger in the bad cases (Figure 3). We added a MinMaxAvg component, which computes the minimum, maximum, or average of a series of values, to the graph and connected it to the Data Loader output port that corresponds to the position difference channel. We set the node to compute the average of the absolute input values. In the list of test runs, adding the node inserted a new column displaying these average values.

We noticed that "good" test runs indeed seem to have a lower average position difference and added a Results component to create a classification of the test runs based on these values. After connecting this new node to the output of the MinMaxAvg node, we opened its settings window and enabled the automatic value mapping. This searches for a linear separation of the input values based on the pre-classified test runs. In this case, the Results component determined that the greatest average value of a "good" test run was 2.72, whereas the least average of a "bad" test run was 5.92. All pre-classifed test runs in the list of test runs turned green, confirming that all classification results matched the expert assessments. The automatic value mapping maps values between 2.72 and 5.92 to a gradual progression from "good" to "bad", and among those test runs without a known expert classification, we can now identify some that are deemed borderline cases according to this metric. Figure 1 shows the state of the prototype implementation after these steps.

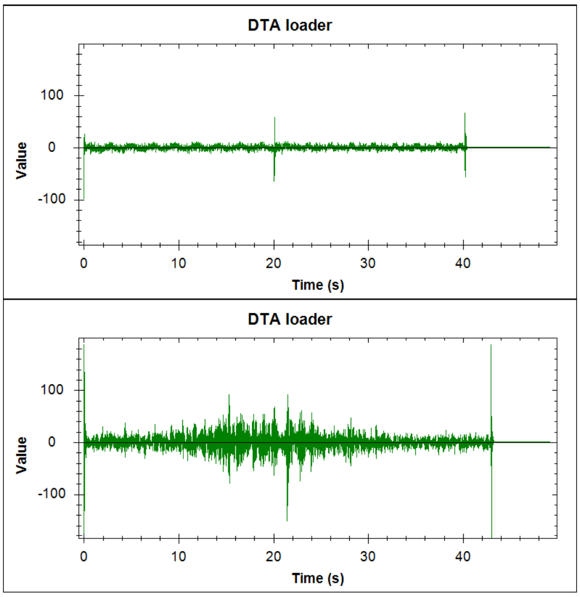

Figure 3: *The data channel recording the difference between the nominal and the actual ball screw head position in cases pre-classified as good (top) and bad (bottom) as seen in the data visualization view of the Data Loader node.*

While the average position deviation appears to be a good indicator, the plots show that even in the "good" cases, there are noticeable peaks (as in Figure 3 top). These usually occur at the beginning, in the middle, and at the end of the measurement. When we pointed this out to the domain experts, we were told these are caused by the starting, stopping, and turnaround motion of the ball screw head, are perfectly normal, and do not indicate a defect. With this additional knowledge, it seemed plausible to ignore these points when calculating the average deviation. The data channel containing the absolute nominal head position can be used to determine the time periods in which the head is meant to accelerate or decelerate and exclude these from the further analysis. To model this in the analysis graph, we add a Differences component and a Comparison component. The Differences component calculates the difference between two consecutive measurements in a data channel and the Comparison component can be configured to produce a list of Boolean values that state whether the corresponding absolute position difference (and thus head velocity) indicates a steady movement rather than acceleration or deceleration. This list of Booleans can then be connected to the MinMaxAvg node, acting as a mask that limits its calculation to these time periods. Once the connection is made, the MinMaxAvg component recalculates the averages and the Results node now classifies test runs as "good" if their average value is not greater than 2.99 and as "bad" if it is not less than 6.87. Compared to the first analysis without the mask, we were able to increase the separation between good and bad from 3.20 to 3.88.

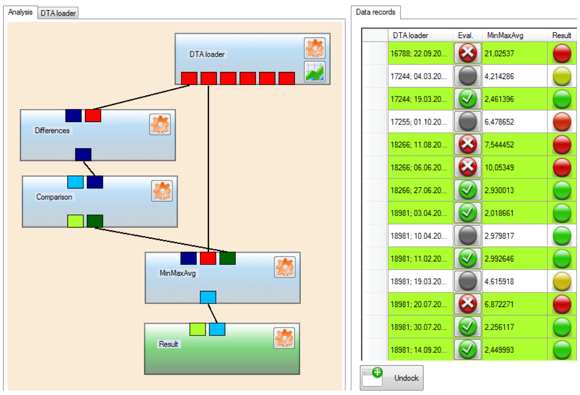

Figure 4: *Using the first data channel to create a mask that limits the analysis to periods of steady head movement.*

Although the average value of the position difference appears to be well suited to classify these test runs, it completely ignores any temporal component of the signal. The plots of bad machines show that defects often do not simply increase the average value, but show time-dependent anomalies. An example is the data series in Figure 3 bottom, which reaches greater values near the turning point in the centre. There are other cases in which values are greater in the second half of the measurement or values are generally within the limits except for a short period with extreme oscillations. To include the temporal component of the time series in the analysis, we added a component that performs a Fourier transformation on a data channel. This computes its frequency spectrum and can thus detect temporal patterns in the data. The visualization of the Fourier node shows a graph of the spectrum. When we compared several spectra, we noticed that "bad" test runs typically exhibited significant oscillations at around 20 Hz, which the "good" test runs did not, and concluded that limiting the average calculation to a frequency band around 20 Hz might further increase the robustness of the analysis.

6. Results

With these components in place, we took the system to a field test at the manufacturer. We started by analysing a set of 40 test runs that represented the same ball screw scenario as our test set. Together with a domain expert, we set up an analysis graph to calculate the average intensity in the 16–32 Hz band. We asked the expert for an opinion on 20 of the test runs and selected the classification limits in the Results node accordingly. We then added the remaining 20 test runs and compared the analysis results with the expert's assessment. He agreed with 19 of these automatic classifications. In one case, however, the analysis marked a test run as "bad" while the expert considered the plot to show only a minor problem. Looking at this test run's spectrum revealed a high intensity in the 28–40 Hz range, which was not found in the other test runs. This might indicate a difference in the configuration of this particular machine or an issue with the ball screw that is different from the defects seen in the other machines. We then moved on to a second set of test runs from measurements of another ball screw component moving perpendicular to the first. After adjusting the Results node to account for the slightly different value range, the graph classified all of the 100 test runs correctly. One machine, which the expert expected to fail very soon, was scored at 0.1, another, which the expert said was having a minor problem, was scored at 0.9. After completing this analysis, we asked for an assessment of the utility of the approach, especially when compared to the current method of mostly manual evaluation. The expert stated his impression was "rather positive" and that the prototype "illustrates an interesting way of how this kind of data can be analysed in the future". He also said that "more experiments are necessary to visualize and analyse a much larger number of measurements" and that he can imagine analysing about 1,000 measurements per year on 10 machine components using this approach.

7. Conclusion

We presented our prototype implementation of a system for the visual analysis of diagnostic machine data from the manufacturing domain. We used an analysis graph to visually represent the analysis steps and allow for an easy and flexible modification of the process during an exploratory data analysis. The visual representation facilitated the communication with domain experts, who provided ground truth evaluations, explanations on certain data features, and a general assessment of the utility of the approach. For now, its usefulness is limited by the lack of available data, which is currently only collected when a machine is prepared for shipping or a defect is known to have occurred. The encouraging results achieved even with these comparatively simple means may have demonstrated the possibilities of modern visual data analysis and prompt a regular collection of data for an eventual introduction of visual analytics as a valuable element of the machine diagnosis process.

References

[BCD*09] BERTHOLD M. R., CEBRON N., DILL F., GABRIEL T. R., KÖTTER T., MEINL T., OHL P., THIEL K., WISWEDEL B.: KNIME-the konstanz information miner: version 2.0 and beyond. *ACM SIGKDD Explorations Newsletter 11*, 1 (2009), 26–31. 2

[JLB06] JARDINE A. K., LIN D., BANJEVIC D.: A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing 20*, 7 (2006), 1483–1510. 1

[MWK*06] MIERSWA I., WURST M., KLINKENBERG R., SCHOLZ M., EULER T.: Yale: Rapid prototyping for complex data mining tasks. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (2006), pp. 935–940. 2

[OGC*10] ONG S. H., GOH K. M., CHAN H. L., LIM T. Y., LING K. V.: Towards machine diagnostics on chip. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on* (2010), pp. 1353–1358. 1

[TZY*10] TANG J., ZHAO L., YU W., YUE H., CHAI T.: Soft sensor modeling of ball mill load via principal component analysis and support vector machines. *Advances in Neural Network Research and Applications* (2010), 803–810. 1

[VHWM09] VERL A., HEISEL U., WALTHER M., MAIER D.: Sensorless automated condition monitoring for the control of the predictive maintenance of machine tools. *CIRP Annals-Manufacturing Technology 58*, 1 (2009), 375–378. 1

[WGY12] WANG J., GAO R. X., YAN R.: A hybrid approach to bearing defect diagnosis in rotary machines. *CIRP Journal of Manufacturing Science and Technology 5*, 4 (2012), 357–365. 1

[zed] ZedGraph. http://sourceforge.net/projects/zedgraph/. 3