Real time trajectory matching and outlier detection for assembly operator trajectories

Bauters Karel [1,2]
Johannes Cottyn[1,2]
Hendrik Van Landeghem[1]
Department of Industrial Systems Engineering and Products Design, Ghent University [1]
Flanders Make [2]
B-9052, Zwijnaarde (Gent),
Belgium
E-mail: karel.bauters@ugent.be

KEYWORDS
Trajectory matching, dynamic time warping, anomaly detection, manual assembly

ABSTRACT

Flexible, reactive and adaptive manufacturing systems are a prerequisite to cope with the demand for low volumes of highly customized products of today's market. For years, manufacturing companies have been using real-time data capturing systems, such as RFID, to gather the necessary data to obtain insights in their production processes, mainly in the domain of quality control and inventory management. However, very few work has been done on monitoring an assembly operator during his work cycle in real-time. This paper presents a method to match operator trajectories, obtained through a multi-camera vision system, in real-time to predefined models. This way, the performance of the operator can be assessed online and problematic or anomalous work cycles can be detected. This information can then be used to support the operator in his pursuit for continuous improvement by pointing out improvement potential.

INTRODUCTION

Over the recent years, the consumer market has made a shift towards more customized and highly variant products. To answer these demands, production systems need to be very reactive and flexible. Contemporary flexible production systems are able to react and adapt their behavior to the current circumstances, based on real-time data and information obtained through a variety of sensor systems. Industrial applications of data gathering using sensor systems can be found in inventory control (Visich, et al. 2009) and job floor control (Arkan and Van Landeghem 2013). RFID technology for example, is used to provide accurate real-time process data which can be used to keep the manufacturing execution system (MES) up-to-date. These examples are all part of the current paradigm shift in manufacturing companies towards Industry 4.0. This fourth industrial revolution is based on digital transformation and cyber-physical systems to overcome the challenges posed by the changing market demands (Sogeti Labs, 2016).
Also within assembly line work stations, sensor systems are being used to monitor the progress of the production process and use this information to update the central production database (Wang 2012). This information can also be used to provide the operator with contextualized work instructions and information about the required inspections and test procedures. These systems are mainly focused on the product rather than on the operator performing the assembly tasks. However, gaining insights in the performance of the operator could provide the assembly line worker with critical information to support him in the continuous improvement of his work methods. Up until today, this kind of information can only be obtained through manual analysis of video-images or at the work station. These analysis methods are prohibitively time-consuming and therefore not tailored to the flexibility and reactivity requirements of contemporary work stations.
Recently researchers presented an analysis system for manual assembly work stations in which multiple cameras are used to track the operators' position in the work station throughout the complete work cycle (Bauters, et al. 2018). These trajectories are then classified into clusters based on their similarity in order to detect outliers or anomalous work cycles. These outliers are work cycles in which irregularities or problems took place and are therefore interesting subjects for further investigation. By pointing towards these anomalous work cycles, this system can significantly decrease the time needed to perform the manual analysis. Furthermore the system calculates a number of performance indicators and visualizes the operators' performance indicators in an operational dashboard to unveil improvement potential.
One of the disadvantages of this system is the fact that the analysis of the video-images is still done offline. This is because the classification method relies on the dynamic time warping algorithm (DTW) to calculate the similarity between different trajectories. This method yields better classification results for this application than other existing similarity measures (Bauters, et al. 2018). However, calculation time for DTW is exponential, making it impossible to calculate the warping between a large number of rather long trajectories in real-time.
In this paper we present a method to match an operators' trajectory to a number of pre-recorded model trajectories in real time. This method makes it possible to detect anomalies in real-time and immediately ask the operator for feedback on what exactly happened during that particular work cycle. Furthermore it enables us to assess to operators' performance in real-time and suggest improvements to his work procedure.
The remainder of this paper is structured as follows. In the next section, a description of the data sets used in this research is given. Afterwards the real-time trajectory matching method is explained before presenting some results. Finally, the

conclusions of this research are presented and some ideas for further research are proposed.

DATA SETS

Two different data sets have been used to validate the trajectory matching methodology. In this section, these data sets are briefly discussed.

Experimental data set

The first data set is created by recording a human operator performing simulated assembly tasks in a laboratory setting. The parts produced in this experiment consist of a Duplo® base block on which different patterns of Lego® blocks are placed. The Duplo® base blocks are delivered to the work station using a conveyer belt that mimicks a moving assembly line. The Lego® blocks are stored in different locations in a picking rack equipped with a pick-to-light system. This way, each different product produced yields a different pattern or trajectory followed by the operator (Bauters, et al. 2018). An overview of the laboratory setting is given in Figure 1.

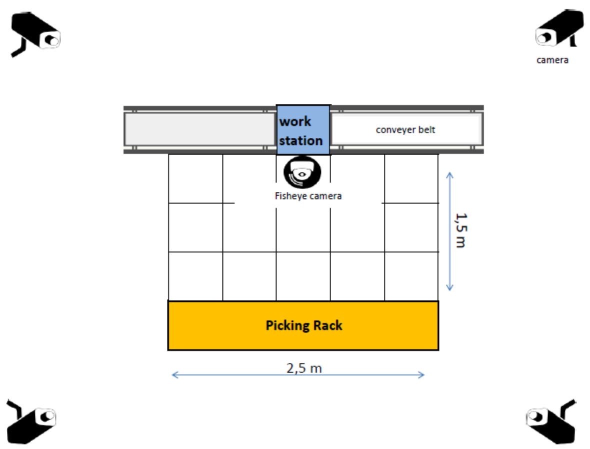

Figure 1: overview of the laboratory setting for data set 1

In this case, a system of multiple cameras (5 in total) was used to determine the operators' position throughout his work cycle. To do this, the principle of voxel carving is used to create a visual hull of the operators' body in every frame of the video sequence as described by several researchers (Laurentini 1994) (Slembrouck, et al. 2015). The position is then determined by projecting the center of mass of this visual hull on the ground plane (x, y). To filter the noise in the resulting trajectories, a Gauss kernel smoothing approach was implemented. Figure 2 shows the resulting visual hull, based on the images obtained through the five cameras.

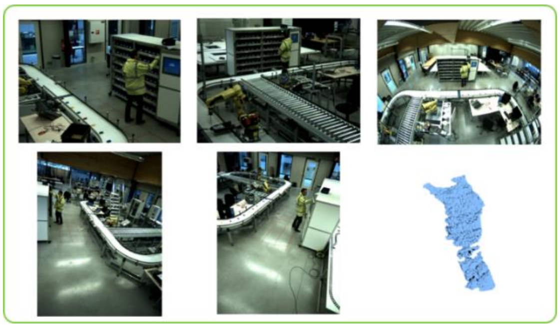

Figure 2: Output of the multi-camera system

This dataset contains 22 different trajectories. Two different patterns can be observed as well as 2 anomalous work cycles. In this paper, this dataset is mainly used to show the ability of the developed methodology to accurately distinguish normal trajectories from anomalous work cycles in real-time.

Omni1

The second data set used in this research is a data set containing over 200 trajectories of people walking through a lab as described in (Morris and Trivedi 2011). All trajectories were recorded within a 24 hour period without the knowledge of the people entering and leaving the laboratory. This dataset was constructed with one single omni-directional camera. Figure 3 provides an overview of the laboratory setting.

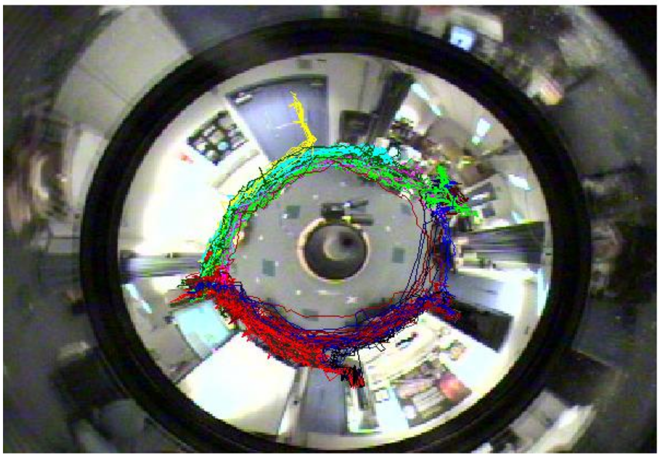

Figure 3: overview of the laboratory setting for data set 2

This dataset contains 7 different activities or patterns in total. All 206 trajectories in this data set are labeled, meaning that for each trajectory the class of the activity performed by the subject is known. In this research, this data set was mainly used to validate whether the trajectory matching methodology is capable of handling a higher number of trajectory models.

Real-time trajectory

For both data sets discussed above, determining the location of the operator/human subject is done off-line. Indeed, obtaining robust and accurate location data from video-images in real-time at high sample rates with existing image processing algorithms remains a challenging task. Allthough a lot of research in the field of image processing is being performed and real-time localization algorithms based on video images are expected to become available in the near

future. Also, there exist a number of different sensors which could deliver exactly the same information.
To overcome this problem, in this research we choose to use trajectory data which is calculated off-line and feed a new location to the system at fixed time intervals, which simulate the frame rate of the cameras. This way, we are able to prove that the developed methodology is capable of handling real-time data once it becomes available.

METHODOLOGY

In this section, the existing method for off-line classification of work cycle trajectories is briefly discussed. Afterwards the challenges encountered when using this method in real-time are clarified and the newly developed method for real-time trajectory matching is presented.

Off-line classification

The inherent variation in the assembly process leads to trajectories that vary in speed and length, even if the tasks performed by the operator are the same. Therefore, one can not simply use the Euclidean distance between concurrent points in two trajectories as a distance measure. To overcome this challenge, trajectories are compared using dynamic time warping (DTW). DTW is a technique that originally was implemented in speech recognition applications, but by now it has successfully been used to cope with deformations in all kinds of (multi-dimensional) time-dependent data (Müller 2007). The idea behind DTW is to find an optimal warping path that minimizes the distance between two trajectories, taking into account a number of warping constraints. The DTW distance can recursively be calculated: given two time series $X := (x_1, x_2, \ldots, x_N)$ and $Y := (y_1, y_2, \ldots, y_M)$ with respective lengths $N, M \in \mathbb{N}$, the cost of the alignment between these two time series can be calculated as follows:

$$C(X_i, Y_j) = \delta(x_i, y_j) + \min\{C(X_{i-1}, Y_{j-1}), C(X_{i-1}, Y_j), C(X_i, Y_{j-1})\}$$

where X_i an Y_j are the respective subsequences $(x_1, x_2, \ldots, x_i)$ and $(y_1, y_2, \ldots, y_j)$ and $\delta(x_i, y_j)$ is the Euclidean distance between two two-dimensional points x_i and y_j. $C(X_n, Y_m)$ determines the DTW distance between the two trajectories. Figure 4 visualizes how this alignment works for two one-dimensional time series. More detailed information on the implementation can be found in (Bauters, et al. 2018).

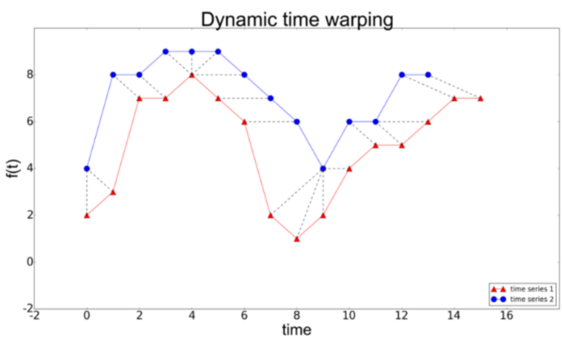

Figure 4: example of DTW alignment between two time-series

The distinction between regular or normal work cycles and anomalies for a set of trajectory sequences, is made based on a hierarchical clustering procedure. Hierarchical clustering methods are used to find a similarity structure in a dataset by initially dividing the data set in n clusters, with n being the number of objects in the data set. The most similar clusters are then merged into a new cluster and this is repeated until all objects are grouped into the same cluster. The similarity structure of the data set is typically visualized in a dendrogram Figure 5, showing the sequence in which clusters are merged. The distance between two merged clusters is indicated by the height of the links in the dendrogram.

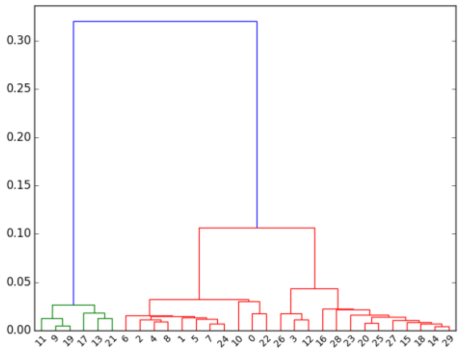

Figure 5: example of a dendrogram

To determine where to cut the dendrogram and thus decide whether the objects in two clusters actually represent the same or a different activity, an adapted version of the permutation testing method proposed by Bruzesse (Bruzesse and Vistocco 2015) is applied. This procedure is based on the assumption that, if two clusters contain similar objects, the distance between two-randomly sampled sets of objects from these clusters will not be significantly different from the distance between the original clusters.
Applying this method on a set of work cycle trajectories, this set is divided into groups or clusters of similar trajectories and single-item clusters which we call outliers or anomalies. The similar trajectories all represent the same assembly process under normal circumstances and can be used to build a model that serves as a template for real-time trajectory matching later on. To build this model, an average trajectory of all sequences in the cluster is calculated. This average trajectory is iteratively calculated using the DBA algorithm as proposed by Petitjean (Petitjean, et al. 2014). The resulting model for one of the 7 patterns in the Omni1 data set is shown in Figure 6.

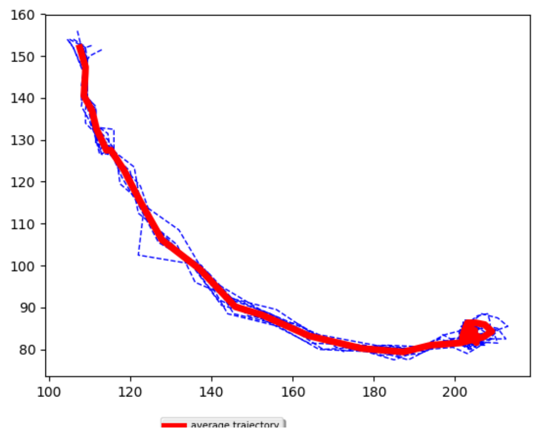

Figure 6: average trajectory model

Real-time trajectory matching

A naïve approach to try to match sequences to the previously calculated models, would be to calculate the DTW distance between the incoming sequence and all of the models, each time a new point is added to the new sequence. The sequence can then be matched to the model resulting in the minimal DTW distance. However, there are a number of downsides to this approach: (1) The calculation of the DTW has a O(n*m) computation time complexity, where n is the length of the incoming sequence and m represents the length of the model. Performing the DTW calculation for every new datapoint, leads to computation times far exceeding the framerate of the camera system, especially if the sequences are becoming longer and the number of trajectory models is high. (2) The incoming trajectory sequence only represents a fraction of the full work cycle. One can rightfully question wether matching such a partial sequence to the model of a complete work cycle actually provides meaningful results.
To overcome these challenges another approach was taken, based on Keogh's lower bound calculation for DTW (Keogh and Ratanamahatana 2005). The idea behind the approach is to calculate a lower bound for the DTW distance between the incoming sequence and a subsequence of the model that has the same length as the incoming trajectory. Based on this low-complexity lower bound calculation, it is possible to eliminate trajectory models from the set of possible candidate matches.
To calculate the Keogh lower bound, a bounding envelope is constructed for each of the trajectory models (Capitani and Ciaccia 2006). Let $M(a_1, \ldots, a_m)$ be a trajectory model of length m and *Env(M)* is the envelope around M defined by two time series *Up(M)* and *Low(M).* Then *Up(M)* and *Low(M)* as follows:

$$Up(M) = \max\left(M_j \middle| j \in [\max(1, i-b), \min(m, i+b)]\right)$$
$$Low(M) = \min\left(M_j \middle| j \in [\max(1, i-b), \min(m, i+b)]\right)$$

In other words, $Up_i(M)$ and $Low_i(M)$ are respectively the maximum and minimum values of M in the interval [i-b, i+b], where b is a user-defined parameter and taking into account the border effects. The squared Keogh LB distance between a subsequence M_n of the model M and an incoming sequence S_n of length n, is defined as follows:

$$LB_{Keogh}(Env(M), S)^2 = \sum_{i=1}^{n} \begin{cases} \left(S_i - Up_i(M)\right)^2 & if\ S_i > Up_i(M) \\ 0 & if\ Low_i(M) \le S_i \le Up_i(M) \\ \left(S_i - Low_i(M)\right)^2 & if\ Low_i(M) > S_i \end{cases}$$

It can be proven that the Keogh LB distance is a lower bound for the DTW distance for 1-dimensional time series. However, the trajectory sequences under investigation in this case, are 2-dimensional. This issue can be overcome by constructing separate envelopes for the x and y component of the model and performing the Keogh LB distance calculation on both the x and y component of the incoming sequence. Rath and Manmatha (Rath and Manmatha 2002) proved that in this case:

$$\begin{aligned} LB_{Keogh}(Env(M_x), S_x)^2 + LB_{Keogh}\left(Env(M_y), S_y\right)^2 &\le DTW(M_x, S_x)^2 + DTW\left(M_y, S_y\right)^2 \\ &= DTW(M, S)^2 \end{aligned}$$

The Keogh LB calculations for the respective x- and y-component of a sequence and model in the Omni1 data set are visualized in Figure 7 and Figure 8.

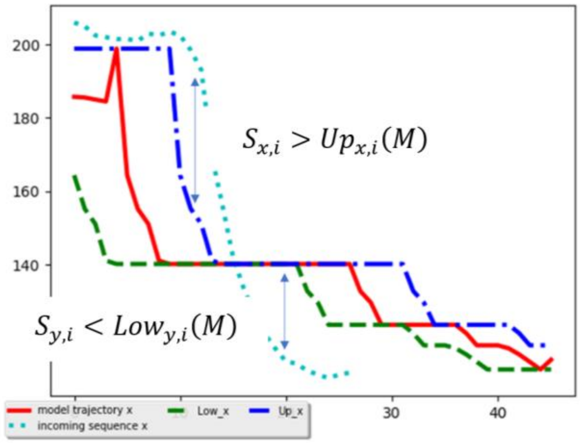

Figure 7: Keogh LB calculation x component

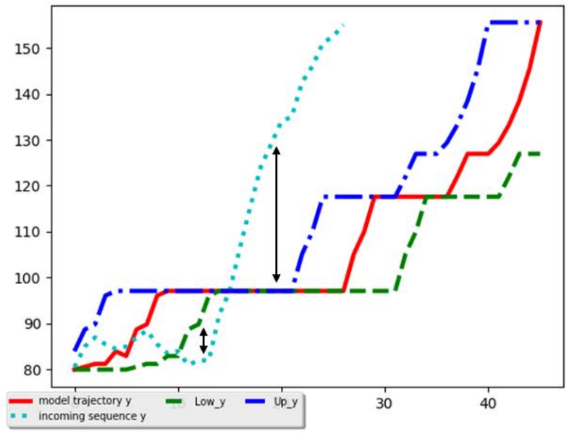

Figure 8: Keogh LB calculation y component

This lower bound calculation requires less computation time than the full DTW calculation. Based on this knowledge, the real-time trajectory matching methodology was developed. In this methodology, the lower bound distance between the incoming sequence and an even long subsequence of all the candidate models is calculated. For the model yielding the best lower bound distance, the DTW distance is calculated and saved as the *best_so_far* distance. Subsequently the LB distances of all candidate models are compared to this *best_do_far* DTW distance and candidate models for which the LB distance is higher than the *best_so_far* are eliminated from the set of candidate models, under the assumption that those models are unlikely to provide a good match for the incoming sequence. The outline of the method is provided in Figure 9.

Real_Time_Trajectory_Matching(M: [(x_1, y_1), …, (x_m, y_m)], S: [(x_1, y_1), …, (x_n, y_n), …])

```
Initialization
Best_so_far ← inf.
Incoming_traj ← []
Start
For new data point:
        LBs := [Keogh_LB for model in
    traj_models]
        Best_model ← traj_models[min(LBs)]
        Best_so_far = DTW(Best_model)
        For model in traj_models:
            If LB>Best_so_far:
                Remove from traj_models
            End if
        End for
        Return best_model
End for
```

Figure 9: outline of the real time trajectory matching algorithm

As shown in Figure 9, the algorithm uses the Keogh LB to estimate what the best matching model is. This way, only one DTW calculation needs to be performed per new incoming data point. The algorithm was then further sped up by implementing a warping window for the DTW calculation.
To detect outliers, the average distance between the average trajectory of the best matching model and all the trajectories used to build up that model (*avg_dist*), is calculated together with the standard deviation σ on those distances. Once the incoming trajectory is fully completed, the DTW distance between the new trajectory and the average trajectory of the best matching model is compared to *avg_dist*. If DTW(incoming, avg_traj) > avg_dist + z.σ , the incoming trajectory is considered to be an outlier.

RESULTS

Experimental data set

The first data set contains two regular trajectories, one in which parts are only picked on the left side of the rack and one for which the necessary parts are stored on the left and right side of the picking rack. In the anomaluous work cycles, the operator travels back-and-forth alongside the rack to set right a picking mistake. The models and outliers are visualized in following Figure 10.
For this data set, the proposed method was able to classify all incoming segments correctly. For every segment, the average calculation time per frame was logged. The average calculation time is 0.07710 seconds, with a maximum time of 0.089 seconds. Twenty frames per second are obtained through the camera system. However, trajectories can safely be downsampled up to a factor 10, without compromising the classification results (Bauters, et al. 2018). Therefore, it can be concluded that the proposed method is capable of performing real-time trajectory matching on this particular data set.

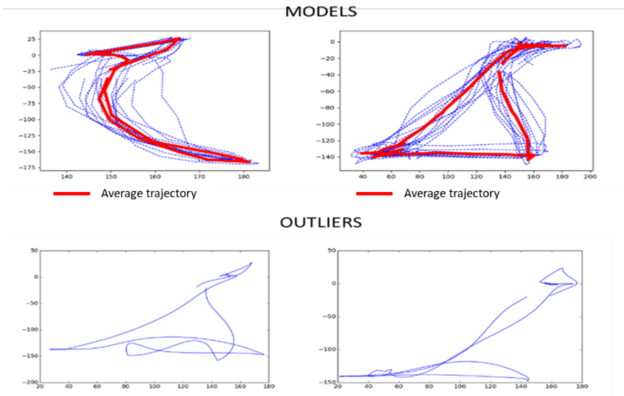

Figure 10: models and outliers experimental data set

Omni1 data set

This data set contains seven models. Therefore one would expect the average calculation time per frame to be higher. The opposite however is true. Due to the fact that the trajectories in this data set are generally shorter than the ones in the first data set, the average calculation time per frame only amounts up to 0.0678 seconds, with a maximum of 0.077 seconds.
On the downside, applying the proposed method on this data set only yields an accuracy 94.3% percent. In the experiments described in this paper, no false negatives (no matching pattern was found when it does exist) were detected. The 5.7% mistakes detected are trajectories that are matched to the wrong model (false positive). This can be explained by the fact that some of the models in this data set share common subtrajectories. Sometimes this results in a slightly higher similarity of the incoming trajectory to a subtrajectory of the wrong model. This occasionally leads to the preliminary elimination of the actual best matching model.

CONCLUSIONS AND FUTURE RESEARCH

In this paper, a method for real-time trajectory matching and outlier detection was presented. The aim of the method is to develop a system that is able to monitor assembly line work station operators and detect problems and mistakes in real-time. The monitoring of the operator is done using a multi-camera video analysis system. By detecting difficulties and problems on-line and linking this to real-time operator feedback and video images, a vast amount of valuable information for improving the process and/or redesigning the work station is created. Until today, this kind of information can only be obtained through manual analyses of video recordings and interviews with operators, which are heavily time-consuming.
The developed outlier detection method is based on dynamic time warping. The Keogh lower bound concept was used to speed up the similarity measurement to enable real-time outlier detection. The method was validated on two different experimental data sets. Results show that the proposed system is capable to accurately detect outliers in real-time.
Further research will focus on accelerating the video analysis in order to evolve to a (near) real time analysis tool. The vision technology and the 3D-model of the operator created

by the visual hull method can also be used to perform an ergonomics analysis of the work cycle. This would be a valuable extra to the system.

REFERENCES

Arkan, Ihsan, and Hendrik Van Landeghem. 2013. "Evaluating the performance of a discrete manufacturing process useing RFID: A case study." *International journal of Robotics and Computer-integrated manufacturing* 29 (6): 502-512.

Bauters, Karel, Johannes Cottyn, Dieter Claeys, Maarten Slembrouck, Peter Veelaert, and Hendrik Van Landeghem. 2018. "Automated work cycle classification and performance measurement for manual work stations." *International journal of robotics and compuetr-integrated manufacturing.*

Bruzesse, D., and D. Vistocco. 2015. "DESPOTA: Dendrogram slicing through a permutation test approach." *Journal of classification* 32 (2): 285-304.

Capitani, Paolo, and Paolo Ciaccia. 2006. "Warping the time on data streams." *Data & knowledge engineering* 62: 438-458.

Keogh, Eamonn, and Chotirat Ann Ratanamahatana. 2005. "Exact indexing of dynamic time warping." *Knowledge and information systems* 7 (3): 358-386.

Laurentini, A. 1994. "The visual hull concept for silhouette-based image understanding." *IEEE transactions on Pattern Analysis and Machine Intelligence* 16 (2): 150-162.

Morris, Brendan Tran, and Mohan Manubhai Trivedi. 2011. "Trajectory learning for activity understanding: Unsupervised, Multilevel, and Long-Term adaptive approach." *IEEE Transactions on pattern analysis and machine intelligence* 33 (11): 2287-2302.

Müller, M. 2007. "Dynamic Time Warping." *Information Retrieval for music and motion* (Springer) 69-84.

Petitjean, F., G. Forestier, G. Webb, A. Nicholson, Y. Chen, and E. Keogh. 2014. "Dynamic time warping averaging of time series for faster and more accurate classification." *IEE international conference on data mining.* Shenzen: IEEE.

Rath, T., and R. Manmatha. 2002. "Lower-bounding of dynamic time warping distances for multivariate time series." Tech Report MM40, University of Massachusetts.

Slembrouck, M., J. N. Castaneda, G. Allebosch, D. Van Cauwelaert, P. Veelaert, and W. Philips. 2015. "Gigh performance multi-camera tracking using shapes-from-silhouettes and occlusion removal." *Proceedings of the 9th international conference on distributed smart cameras.* New York.

Visich, J. K., B. M. Khumawala, P. M. Reyes, and S. Li. 2009. "Empirical evidence of RFID impacts on supply chain performance." *International journal of operations & production management* 29 (12): 1290-1315.

Wang, L. 2012. "RFID-based information technology abd labagement in agriculture and food supply chains." *International journal of information technology and management* 11 (3): 225-239.

WEB REFERENCES

Industry 4.0: Factory to smactory, disruption meets the manufacturing industry.
http://labs.sogeti.com/industry-4-0-factory-to-smactory-disruption-meets-the-manufacturing-industry
Accessed on: 2018-05-03