Preliminary Investigation of Assembly Work Activity Recognition with Wearable Sensors via Unsupervised Learning

Qingxin Xia[1] Atsushi Wada[2] Joseph Korpela[1] Takuya Maekawa[1] Yasuo Namioka[2]

Abstract: This paper presents an unsupervised method for recognizing assembly work done by factory workers by using wearable sensor data. Such assembly work is a common part of line production systems and typically involves the factory workers performing a repetitive work process made up of a sequence of manual operations, such as setting a board on a workbench and screwing parts onto the board. This study aims to recognize the starting and ending times for individual operations in such work processes through analysis of sensor data collected from the workers along with analysis of the process instructions that detail and describe the flow of operations for each work process.
We propose a particle-filter-based factory activity recognition method that leverages (i) trend changes in the sensor data detected by a nonparametric Bayesian hidden Markov model, (ii) sensor-data similarities between consecutive repetitions of individual operations, and (iii) frequent sensor-data patterns (motifs) discovered in the overall assembly work processes. We evaluated the proposed method from six workers collected in actual factories, achieving a recognition accuracy of 83.3% (macro-averaged F-measure).

Keywords: Activity recognition, wearable sensor, factory work

1. Introduction

Many factories employ a line-production system in which each product passes through the same sequence of work processes in series. Often included amongst these processes is assembly work conducted manually by factory workers, where each worker repetitively performs a set of predefined processes with each process consisting of a sequence of operations, such as setting a board on a workbench and screwing parts onto the board. Such assembly work by factory workers still constitutes the core of line-production systems, making the improvement of assembly work one of the most important aspects of increasing productivity in these systems [1, 10].

The goal of this work is to recognize the individual operations conducted during an overall production process by using wearable sensor data. More specifically, we identify the starting and ending times of each individual operation through analysis of the sensor data. Having detected the timing of the operations, we can then support many useful applications, such as monitoring the status of workers and detecting outlying operations. Such automated monitoring systems for assembly-work activities are in high demand by manufacturers, since line-production systems often employ large numbers of workers and it can be difficult for line managers to monitor all their workers simultaneously.

Many of previous works have applied supervised learning techniques to recognize these types of actions (operations) using wearable sensor data. However, collecting labeled training data such as is required by these supervised approaches can be prohibitively costly when applied to assembly-work activities due to the following issues:

(1) Each worker may perform a different operation process, necessitating training data collection for every worker.
(2) The operations themselves may be modified frequently (e.g., weekly or monthly) due to frequent revisions of the production system, necessitating training data collection on a regular basis.

This study addresses this shortfall by attempting to recognize line-production operations in an unsupervised manner.

Figure 1 shows example sensor data collected from a worker. As shown in the figure, a typical work process is iterated several times, with each iteration of the process (i.e., work period) comprised of a sequence of operations. In this example, we have eight operations per work period, iterated three times. The goal of this study is to identify the starting and ending times for each operation, as well as to estimate class labels for the operations.

However, as can be seen in the figure, it is difficult to identify the sensor data segments corresponding to different operations solely using sensor data. The main feature of the proposed method is to leverage information included in process instructions in conjunction with the sensor data. Process instructions are documents that describe the detailed instruction and flow of the operations included in each work process. In many cases, these instructions include information about the standard duration of each operation in a process, which we can use when identifying the starting and ending times of the individual operations. However, in practice the actual duration of each operation will differ from one work period to the next as shown in Figure 1, meaning that

[1] Graduate School of Information Science and Technology, Osaka University
[2] Corporate Manufacturing Engineering Center, Toshiba Corporation

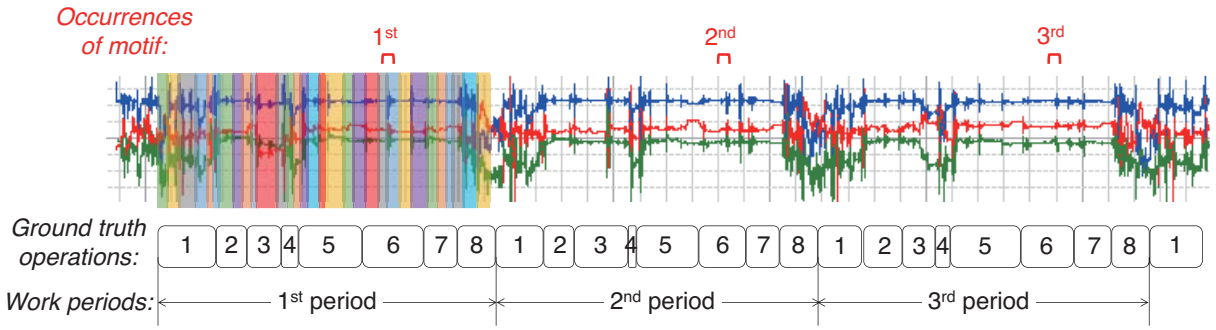

Fig. 1 Example acceleration data collected from a worker. Red, green, and blue lines show x-, y-, and z-axis data, respectively. The rectangles below show the ground truth labels of the operations. The duration of each operation period is about 120s. Each work period consists of eight operations, which are defined in the process instructions. Red brackets show the occurrences of a motif detected by the proposed method.

the process instructions alone cannot be used to accurately identify the boundaries between operations.

In this work, we employ a particle filter to robustly identify the boundaries of the operations despite the variance in their durations. We propose the following techniques to enable the precise unsupervised recognition of assembly-work operations based on a particle filter:

(1) We note that the transition points between operations in a work period may contain noticeable trend changes in the sensor data, e.g., in Figure 1 we can see a dramatic change in the characteristics of the sensor data at each ending of the first operation. We can identify candidates for these transition points based on the trend changes detected by a nonparametric Bayesian hidden Markov model (HMM) and then determine the best sequence of transition points from these candidates by matching them to the standard operation durations using a particle filter.

(2) It is likely that the sensor data of an operation from one work period will be similar with data from the same operation in recent periods from the same worker. Thus, we can calculate the sensor data similarity for an operation across different work periods in order to identify the correct sensor data segments for the operation.

The research contributions of this study are summarized as follows.

- We propose an unsupervised factory-activity recognition method for wearable sensors.
- The proposed method leverages information extracted from process instructions as well as sensor data analysis to precisely recognize activities in an unsupervised manner.
- To our knowledge, this is the first study on recognizing factory activities without the use of labeled training data.
- We evaluate the proposed method using sensor data collected from six workers in actual factories. Using this data, the proposed method achieved 83.3% recognition accuracy.

In the rest of this paper, we first review activity recognition studies that use wearable sensors including factory-activity recognition studies. We then present the proposed activity recognition method and evaluate the method using sensor data collected in actual factories.

2. Related Work

Human activity recognition is a core topic in the ubicomp community, with many prior examples of it being conducted using supervised machine learning techniques that rely on labeled training data [4, 5, 9]. Along with these supervised-learning techniques, several studies have also been done on reducing the effort required to collect the labeled training data needed for supervised learning. Maekawa et al. [13] trained a target user's activity model on labeled sensor data from source users with similar physical traits to those of the target user. Huynh et al. [6] used topic models to cluster activity data in an unsupervised manner.

Due to the recent growing interest in smart manufacturing and Industry 4.0 [11, 15], studies on recognizing and supporting factory work using sensor technologies [2, 3] have been attracting attention. Of particular note are the previous studies done on monitoring and analyzing factory work using wearable sensors. One such study is Koskimäki et al. [8] in which they obtained acceleration and gyroscopic data from a wrist-worn inertial sensor device and analyzed operation processes in a line-production system to ensure that all necessary operations were performed. They recognized activities such as hammering and screwing using a kNN search. Another such study is Ward et al. [17] in which they obtained acceleration and audio data from a wrist-worn device to recognize woodworking activities by using HMMs and a linear discriminative classifier. Stiefmeier et al. [16] focused on assembly work on automobiles and used inertial sensors attached to several body parts, such as the upper and lower arms, to classify sensor data segments by computing the distance between the collected segments and sensor data templates prepared in advance. Examples of the activities used in their work include opening an engine hood and opening a trunk. All of these methods for analyzing factory work relied on supervised machine learning approaches and so required

the creation of labeled training data.

The most relevant recent work is Maekawa et al. [12] in which they measured the duration of each work period on a production line in an unsupervised manner. Their method tracked a motif (sensor data segment) that appears only once in each work period using a particle filter to estimate the duration of the periods. Our work uses this method to roughly capture the structure of work periods during our preprocessing stage.

3. Unsupervised Factory Activity Recognition Method

3.1 Assumed Environment

3.1.1 Sensor setting

In our experiment, the workers wore a smart watch on their right wrists, with a three-axis accelerometer embedded, the sampling rate is about 60 Hz.

3.1.2 Process instructions

Process instructions are prepared in advance by a line manager for each work process to be done by a worker. They specify the flow of operations included in a work process, e.g., (1) place a board on the workbench, (2) change the signal monitor mode, with some work processes containing optional operations, e.g., and (3) pack accumulated boards into a box once 10 have been completed. These instructions also describe the standard duration of each operation. For example, how long the above operations will cost in standard production scenario.

3.2 Overview

Figure 2 shows an overview of the proposed method. We start with a time series of acceleration data that has been collected from a worker that corresponds to multiple iterations of that worker's work period. We then identify and track a motif that occurs once and only once per work period in this sensor data. Since this motif occurs only once per work period, the period of sensor data between the first and second occurrences of this motif must contain the starting time of the second work period, as shown in Figure 1.

In addition to the large-scale segmentation done using motifs, we also segment the sensor data on a smaller scale using a nonparametric Bayesian HMM. As can be seen in Figure 1, visible trend changes are evident at the transition points between many of the operations. Such trend changes can be detected using HMMs and can be considered as possible start times for individual operations in the work period, including the start time of the first operation in the work period.

Having decided the region of sensor data containing the start time for the second work period by motif and all possible start times for operations by HMM, we then run our particle filter. First we initialize our particle filter by generating an initial particle at a discovered trend change, representing a likely start point for the second work period (i.e., the start point of the work period's first operation). Then, as shown in Figure 3, particles are generated (sampled) along possible paths from the work period's start point through all the individual operation start times by iteratively sampling, weighting, and resampling at each estimated operation start time to generate estimates for the next.

The particle filter begins with the sampling phase, in which it uses a work model that encapsulates data derived from the process instructions to generate candidate start times (particles) for the following operation in the work period from an initial particle. In the weighting phase, the candidate start times (particles) are assigned weights based on (i) trend changes in the sensor data detected by HMM, (ii) data similarities between operations and corresponding operations in preceding work periods, and (iii) the consistency of the motif's location in the current work period with its location in the previous work period. In the resampling phase, the particle filter resamples the candidate particles based on their weights, leaving it with the most likely start times for that operation. Each of these remaining particles then serve as initial particles when starting the sampling phase of the next iteration. At the end of each work period, we choose our estimated sequence of operation start times as being the path generated by our particle filter with the highest weight.

3.3 Work Model

The first step in our process is to construct a work model that encapsulates the information obtained from the process instructions. The model is a tree-structured representation of the flow of operations, which is represented as a tree structure whose root node corresponds to the first operation with at least one additional node in the tree for every possible operation in the work period. Edges connect two neighboring operations by time sequence, the tree branch indicates optional operations. Each tree node contains information about the expected duration of the operation of interest, modeling this duration as a lognormal distribution. This operation flow model is then used to sample the durations of operations when generating particles to use as candidate start times for operations.

3.4 Motif Detection and Tracking

The first step in identifying the start time for an overall work period is to identify and track a motif that occurs only once in each work period based on the method proposed in [12]. This method takes a time series of acceleration data and the standard duration of a work period from our work model as input and outputs timestamps for the locations of the most likely motif found in the sensor data (see Figure 1 for an example of this output). Because the starting time of the n-th work period exists between the $n-1$-th and n-th occurrences of the motif, as shown in Figure 1, we can then narrow our search for this starting time to the area between these two occurrences of the motif. Based on [12], the best motif is selected by three steps: (1) Random generation of motif candidates: We build a Gaussian mixture model to generate motif candidates between time 0 and time t_{init} by

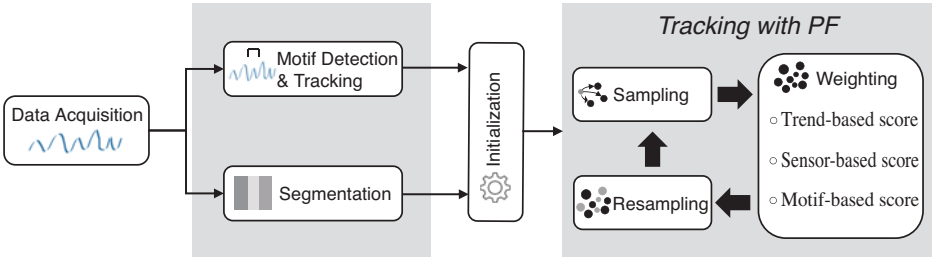

Fig. 2 Overview of the proposed method

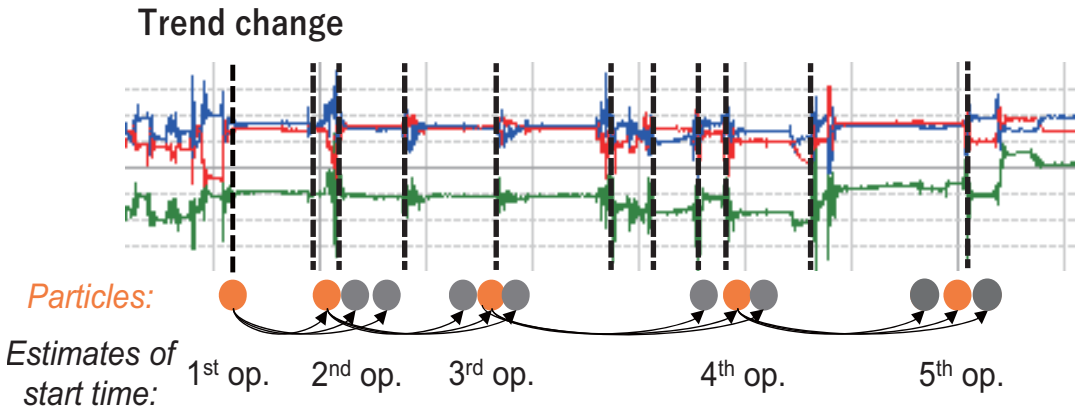

Fig. 3 An example of how paths of estimated start times for each operation are generated using a particle filter. We start with an initial particle at some trend changes between the first and second motifs, which is a candidate start time for the first operation. We then sample several particles from this initial particle (three in this example), which are the initial estimates for the start time of the second operation. These sampled particles are then weighted and resampled, leaving us with our most likely estimates for the start time (one in this example). The most likely estimates are then sampled to generate the estimates for the start time of the third operation. The particle filter continues in this way, iteratively generating the start times for each operation based on its estimates for the start time of the previous operation.

the duration of operation. Time t_{init} is a hyperparameter for the algorithm, set as half of the standard lead time in our work. (2) Tracking motif candidated using a particle filter: We then search for potential locations where each of these motif candidates will reoccur in the data using a particle filter.This particle filter initializes a particle at the exact location (timestamp) of each one of our randomly generated motif candidates, with each particle storing information about the motif candidate it represents, i.e., the sensor data segment for the motif. The particle filter then generates each motif's potential subsequent locations by iterating through phases of sampling, weighting, and resampling. (3) Computing final scores and selecting the best motif: Using the steps outlined above, we now have several candidate motifs for each of which we have several sequences of particles that represent possible occurrences of that motif across our sensor data. Finally, we compute a single score for each of these candidate motifs, which represents the sequence of particles for that motif that had the highest weights.

3.5 Segmentation

In order to identify likely starting locations for work periods in the sensor data as well as likely starting locations for operations in the work periods, we segment the sensor data based on trend changes in the data. We do this in an unsupervised manner by employing a Bayesian nonparametric version of HMM called the hierarchical Dirichlet process HMM (HDP-HMM) [7].

The 1^{st} period in Figure 1 shows an example result from this segmentation method when applied to acceleration data. The overlaid color rectangles represent the segments generated by HDP-HMM, while the numbered rectangles below the data show the ground truth operations for the data. As can be seen in this figure, while HDP-HMM segmentation resulted in many false positives, many of the ground truth segmentation boundaries were included in its results. Therefore, we can use HDP-HMM to generate a list of candidate start times for the individual operations, which in turn can also be treated as candidate start times for the work period.

3.6 Tracking with Particle Filtering

3.6.1 Overview

Using the methods described in Section 3.4 and Section 3.5, we are able to obtain timestamps for the occurrences of a motif $\mathbf{M}$ and timestamps for trend changes (candidate start times) $\mathbf{C}$. We now want to use a particle filter to produce our estimate for the most likely starting time for our work period along with the starting times for all operations in that work period. We start by initializing this particle filter based on $\mathbf{C}$ and $\mathbf{M}$ to generate particles that act as

Algorithm 1: `Particle filtering for operation recognition`

Input: $\mathcal{X}$: time-series of acceleration, $\mathcal{W}$: work model of work process, $\mathbf{M}$: timestamps of occurrence of motif, $\mathbf{C}$: timestamps of trend changes

```
1  /* Initialize particles */
2  P ← ∅ /* For initial particles */
3  P^F ← ∅ /* For final results */
4  for ∀c_i ∈ C do
5      if M[1] < c_i < M[2] then
6          P ← P ∪ {c_i}
7  /* Particle filtering */
8  for ∀p_j ∈ P do
9      p_j.score ← 0
10     p_j.period ← 1
11     P^- ← {p_j}
12     repeat
13         for ∀p_i ∈ P^- do
14             p_i.state ← W.root /* Set to initial
                state */
15             p_i.period ← p_i.period + 1
16         repeat
17             /* Sampling */
18             P^+ ← P^-
19             P^- ← ∅
20             for ∀p_i ∈ P^+ do
21                 P^- ← P^- ∪ Sampling(p_i)
22             /* Weighting */
23             for ∀p_j ∈ P^- do
24                 if Is-Leaf-Node(p_j.state) then
25                     if p_j.period > 2 then
26                         p_j.score ←
                            Trend-based-Score(p_j, C) +
                            Sensor-based-Score(p_j, X) +
                            Motif-based-Score(p_j, M) /*
                            Normalize before sum */
27                     else
28                         p_j.score ←
                            Trend-based-Score(p_j, C)
29                 else
30                     p_j.score ←
                        Trend-based-Score(p_j, C)
31             /* Resampling */
32             P^- ← Roulette-Wheel-Selection(P^-)
33             for ∀p_j ∈ P^- do
34                 p_j.state ←
                    Get-Next-State(W, p_j.state)
35         until repeat the process until the leaf nodes of W
36     until repeat the process until the end of X
37     P^F ← P^F ∪ P^-
```

Output: A particle in $\boldsymbol{P}^{\boldsymbol{F}}$ with the maximum score

candidates for the starting time of the work period. We then track each of these particles to identify the starting times of the other operations in the work period by iterating through the particle filter's three phases of sampling, weighting, and resampling as shown in Figure 3. Figure 4 illustrates the sampling, weighting, and resampling procedures, while Algorithm 1 shows the algorithm of this procedure.

3.6.2 Initialization

As can be seen in Figure 1, the starting time of the second work period exists somewhere between the first and second occurrences of our motif. Therefore, we can generate initial particles at each timestamp from $\mathbf{C}$ that fall between the first and second occurrences of the motif (found in $\mathbf{M}$), which become our candidates for the starting time of the first operation in the second work period (line 1 in Algorithm 1). We also set the current state of each initial particle as the root node (first operation) of the operation flow from our work model.

3.6.3 Sampling

As shown in Figure 4 (a), with each iteration we sample new particles from each existing particle that represent estimates of the starting time for the following operation (line 17 in Algorithm 1). Assuming that $t(p_i, n, k)$ is the timestamp for an existing particle p_i that represents the starting time of the n-th operation in the k-th work period, then the timestamp $t(p_j, n+1, k)$ for a new particle p_j for the $n+1$-th operation is sampled from p_i as follows:

$$t(p_j, n+1, k) = t(p_i, n, k) + \Delta t,$$

where Δt is randomly sampled based on the duration-time distribution found in the currently selected node in the work model's operation flow (i.e., the n-th operation's node). Therefore, $t(p_j, n+1, k)$ is an estimate of the starting time of the $n+1$-th operation. Note that each new particle p_j stores time $t(p_i, n, k)$ along with the times of all previous particles in its chain as a history of the starting times for previous operations.

3.6.4 Weighting

Each particle generated in the sampling phase is then assigned a weight using a combination of three scores: a trend-based score, a sensor-based score, and a motif-based score (line 22 in Algorithm 1). We determine which of these scores to use depending on the current state of the particle, with the trend-based score always used, as shown in Figure 4 (b), and the sensor-based, and motif-based scores used when the state corresponds to the final operation of a work period, as shown in Figure 4 (c). These scores are calculated as follows:

Trend-based score: This score enables us to detect operation starting times based on the trend changes discovered by HDP-HMM. As shown in Figure 4 (b), this score is computed based on the temporal distance between the timestamp of a particle and the timestamp of the closest trend change in $\mathbf{C}$ as follows:

$$\textsc{CumulativeDist}(p_j, n, k, \mathbf{C}) = \sum_{m=2}^{n+1} \textsc{Dist}(t(p_j, m, k), \mathbf{C}))$$

where $t(p_j, m, k)$ is the timestamp for the ancestor particle of p_j corresponding to the m-th operation in the k-th work period (when m is equal to $n+1$, it is the timestamp for p_j itself) and

$$\textsc{Dist}(t, \mathbf{C}) = \min_{c \in \mathbf{C}} |t - c|.$$

The inverse of this distance is then used as the score, with this score used each time we compute weights for particles.

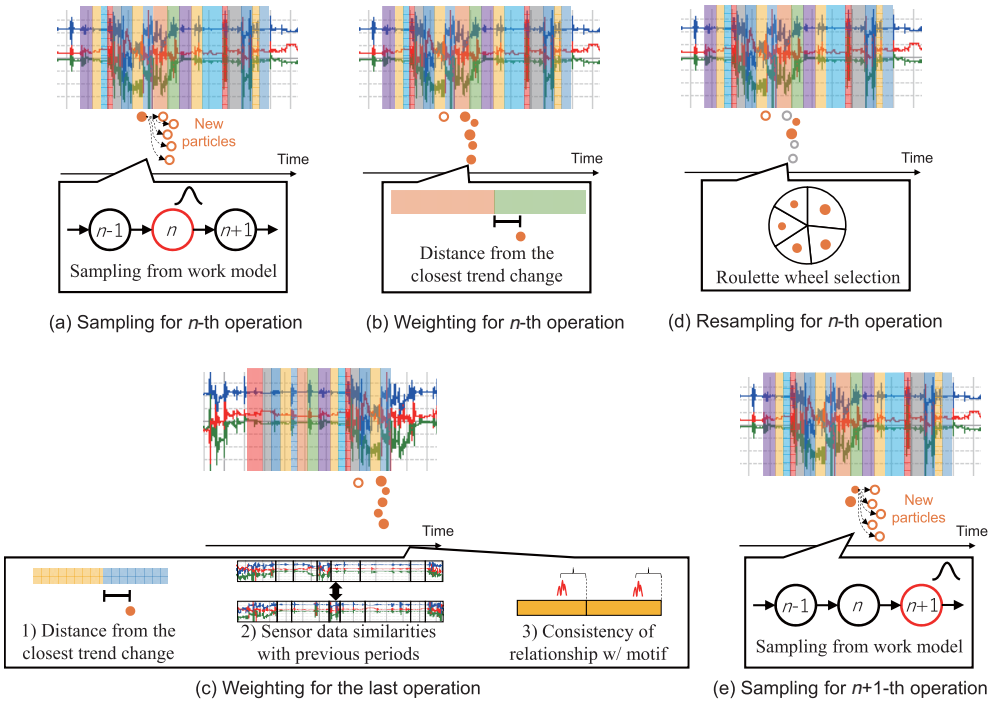

Fig. 4 Sampling, weighting, and resampling procedures in the proposed method

Sensor-based score: The sensor-based score is only computed for particles representing the end of a work period. It enables us to compare two work periods by leveraging the expected similarity of sensor data for the same operation even when conducted in different work periods. We calculate the sensor data similarity for each operation using dynamic time warping (DTW) as follows:

$$DTW(\mathcal{X}(p_j, n, k), \mathcal{X}(p_j, n, k-1)),$$

where $DTW(\cdot, \cdot)$ calculates the DTW distance between two sensor data segments. Here, $\mathcal{X}(p_j, n, k)$ gives the sensor data segment of the n-th operation in the k-th work period identified by particle p_j.

$\mathcal{X}(p_j, n, k-1)$ gives the sensor data segment of the n-th operation in the $k-1$-th work period identified by an ancestor particle of p_j corresponding to the n-th operation in the $k-1$-th work period. We compute the sum of the dynamic time warping distances and the regard the inverted value as the sensor-based score.

Motif-based score: Like the sensor-based score, the motif-based score is only computed for particles representing the end of a work period. Because the motif tracking method [12] finds motifs that occur only once per work period, motifs will occur with an interval corresponding to the lead time for the work period, with the location of a motif within any work period consistent across adjacent work periods (see Figure 1), even when some work periods contain optional operations (which is proven in [12]). Therefore, we can evaluate the quality of an estimated series of operation start times based on the consistency of motif locations in the estimated work periods. We start by computing two values for each occurrence of a motif that measure the temporal distances from the motif to the start time (d_s) and end time (d_e) of the work period. We then use these temporal distances to compare the k-th and $k-1$-th occurrences of a motif by computing the absolute difference between each value for k-th and $k-1$-th occurrences (e.g., $|d_s^k - d_s^{k-1}|$), using the smaller of these two absolute differences in the motif-based score.

Additionally, since the motif should be observed in sensor data segments corresponding to the same operation in each work period (as shown in Figure 1), we also compute the temporal distance between its location and the starting/ending time of the operation in which it occurs. We then incorporate the absolute difference between these distances for the current work period and the previous work period as part of the motif-based score. Furthermore, when the motif does not appear in the same operation for the current and previous work periods, we apply a penalty to the score. This score enables us to reduce the overall score for estimated paths that do not match the standard lead time of the work period. The three scores described above are computed at the end of each work period. The final score used for any given particle is then calculated as the cumulative sum of these three scores for the current particle together with these scores from all of its ancestor particles (i.e., the particles in its path from each previous work period).

3.6.5 Resampling

The sampled particles are probabilistically resampled using roulette wheel selection as shown in Figure 4 (d) (line 31 in Algorithm 1). We then set the current state for each remaining particle as the following operation in the operation flow (line 34 in Algorithm 1). In the case of a branch in the operation flow, we randomly select the branch to follow. Each of these remaining particles is then used to generate

new particles during the sampling phase of the next iteration, as shown in Figure 4 (e). Note that, because the durations of some operations may have high variance, such as optional operations, we resample two times as many particles as are sampled for later periods. Further, because an estimate of the ending time of the last operation of the k-th work period should exist between the k-th and $k+1$-th occurrences of the motif as shown in Figure 1, we discard estimates (particles) for the ending time of the last operation that do not satisfy this condition.

We iterate through sampling, weighting, and resampling until reaching the end of the sensor data, at which point we select the particle with the highest weight as representing the most likely sequence of start times for operations in the sensor data.

4. Evaluation

4.1 Dataset

We evaluated the proposed method using data collected from factory workers in actual factories. Table 1 gives an overview of the work processes observed for this study. The parentheses in the table give optional operations case. The sensor data was collected using a Sony SmartWatch3 SWR50 attached to each worker's right wrist, with the acceleration data sampled at approximately 60 Hz. All data processing was conducted offline in our laboratory after collection.

4.2 Evaluation Methodology

Our method provides the starting (and ending) times for all operations in each work period, allowing us to classify each sensor data point as belonging to an operation class (excluding data points in the first work period). During evaluation, we can then treat our results as classification results estimated per data point, allowing us to calculate the macro-averaged F-measure of our results.

To evaluate the effectiveness of the proposed method, we prepared the following methods.

- **Proposed**: This is the proposed method.
- **w/o Sens**: The proposed method without the use of the sensor-based score during the weighting phase of the particle filter.
- **w/o Motif**: The proposed method without the use of the motif-based score during the weighting phase of the particle filter.
- **only Trend**: The proposed method using only the trend-based score during the weighting phase of the particle filter. That is, this method uses neither the semantics-based score, the sensor-based score, nor the motif-based score.

4.3 Results

4.3.1 Recognition accuracy

Figure 5 shows the classification accuracy for each method. Note that because the methods used in this study are based on particle filters, each method can produce several estimated segmentation paths per work period. The results shown in Figure 5 select a single path when calculating the F-measure by selecting the path corresponding to the particle with the highest score at the end of the *final* work period. Additionally, the accuracies for the w/o Motif and only Trend methods are zero for some work processes because all particles were discarded in the resampling phase at the end of a work period, i.e., no particles satisfied the condition that estimates for the ending time of the last operation of the k-th work period should be between the k-th and $k+1$-th occurrences of the motif.

Overall, the Proposed method achieved the highest average accuracy across all six work processes (83.3%). Also of note are the poor results from the only Trend method, which indicate that it is not possible to recognize the operations using only trend changes in the sensor data.

Fig. 5 Accuracies (F-measures) of the four methods calculated using the path of the particle with the highest score at the end of the final work period

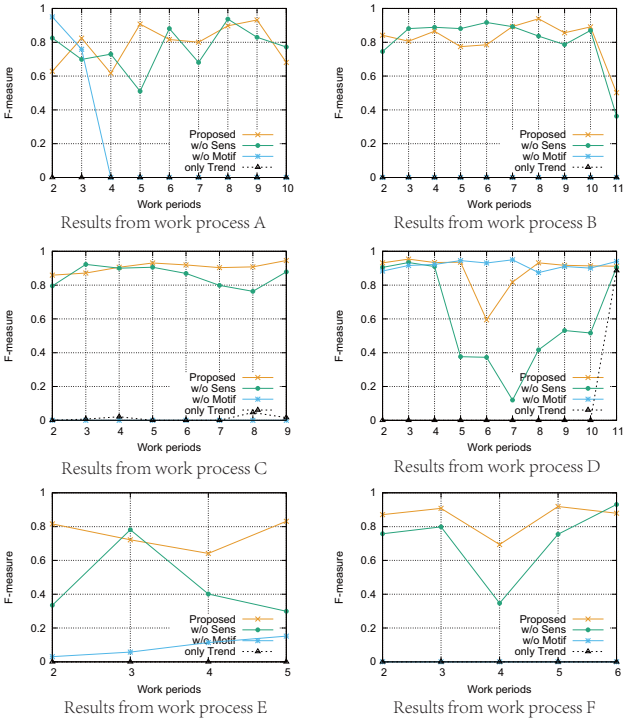

Fig. 6 Transitions in the accuracies (F-measure) of the methods calculated from the path of the particle with the highest score at the end of each work period

Table 1 Overview of our dataset

	work A	B	C	D	E	F
number of operations	11 (12)	8	6 (8)	11	7	7 (8)
standard duration [s]	50 (59)	124	55 (115)	52	55	46 (60)
periods	11	12	10	12	6	7
data duration [s]	655	1440	614	639	360	420
work overview	install screws on circuit board	test and record circuit board information	bag and box circuit boards	install screws on circuit board	check final product and record results	check final product and record results

4.3.2 Transitions in recognition accuracy

Figure 6 displays the average F-measures of every period of works with paths selected from Figure 5. As can be seen from Figure 6, the accuracies for the Proposed and w/o Sens methods can stay a considerable high level in many work periods, while the w/o Motif and only Trend approaches cannot find a good path of particles in most of the work processes. Since the motif consistently locates at each work period, the motif-based score discarded many bad paths with large shift to the motif, remaining better path candidates for sensor- and trend-based scores' calculation.

In addition, we can find sudden decreases in the recognition accuracy. This is mainly caused by the first occurrence of an optional operation. Because we cannot employ the sensor-based score for such operations, the recognition accuracy significantly decreases in such cases. Further, as can be seen in the results, the transitions of Proposed seem to be more stable than those of the other methods. This robust estimation is achieved by increasing the diversity of input by incorporating all three score functions.

As above, we could confirm the effectiveness of our particle filter-based architecture and our proposed scoring methods using sensor data collected in actual factories.

5. Conclusion

This paper presented an unsupervised method for recognizing assembly work done by factory workers using wearable sensor data. The proposed method detects the starting and ending times of each operation included in a work process by leveraging three scores, which achieved a high average accuracy of about 83.3%.

The complete version with an extended score taking into account of semantic information of operations is proposed in [14].

Acknowledgment

This work is partially supported by JST CREST JPMJCR15E2, JSPS KAKENHI Grant Number JP16H06539 and JP17H04679. The first author is supported by China Scholarship Council.

References

[1] Mario Aehnelt, Enrico Gutzeit, and Bodo Urban. 2014. Using activity recognition for the tracking of assembly processes: Challenges and requirements. In *WOAR 2014*. 12–21.

[2] Mario Aehnelt and Karoline Wegner. 2015. Learn but work!: towards self-directed learning at mobile assembly workplaces. In *the 15th International Conference on Knowledge Technologies and Data-driven Business*. 17.

[3] Sebastian Bader, Frank Krüger, and Thomas Kirste. 2015. Computational causal behaviour models for assisted manufacturing. In *the 2nd international Workshop on Sensor-based Activity Recognition and Interaction*. 14.

[4] Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *Pervasive 2004*. 1–17.

[5] Mark Blum, Alex Sandy Pentland, and Gehrard Tröster. 2006. Insense: Interest-based life logging. *IEEE Multimedia* 13, 4 (2006), 40–48.

[6] Tâm Huynh, Mario Fritz, and Bernt Schiele. 2008. Discovery of activity patterns using topic models. In *UbiComp 2008*. 10–19.

[7] Matthew J. Johnson and Alan S. Willsky. 2013. Bayesian Nonparametric Hidden Semi-Markov Models. *Journal of Machine Learning Research* 14, 1 (2013), 673–701.

[8] Heli Koskimäki, Ville Huikari, Pekka Siirtola, Perttu Laurinen, and Juha Röning. 2009. Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In *17th Mediterranean Conference on Control and Automation (MED 2009)*. 401–405.

[9] Jonathan Lester, Tanzeem Choudhury, and Gaetano Borriello. 2006. A practical approach to recognizing physical activities. In *Pervasive 2006*. 1–16.

[10] Bruno Lotter and Hans-Peter Wiendahl. 2013. *Montage in der industriellen Produktion: Ein Handbuch für die Praxis*. Springer-Verlag.

[11] Dominik Lucke, Carmen Constantinescu, and Engelbert Westkämper. 2008. Smart factory-a step towards the next generation of manufacturing. In *Manufacturing systems and technologies for the new frontier*. Springer, 115–118.

[12] Takuya Maekawa, Daisuke Nakai, Kazuya Ohara, and Yasuo Namioka. 2016. Toward practical factory activity recognition: unsupervised understanding of repetitive assembly work in a factory. In *UbiComp 2016*. 1088–1099.

[13] Takuya Maekawa and Shinji Watanabe. 2011. Unsupervised activity recognition with user's physical

characteristics data. In *International Symposium on Wearable Computers (ISWC 2011)*. 89–96.

[14] Xia Qingxin, Atsushi Wada, Joseph Korpela, Takuya Maekawa, and Yasuo Namioka. 2019. Unsupervised Factory Activity Recognition With Wearable Sensors Using Process Instruction Information. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 2 (2019), 60.

[15] Agnieszka Radziwon, Arne Bilberg, Marcel Bogers, and Erik Skov Madsen. 2014. The Smart Factory: Exploring adaptive and flexible manufacturing solutions. *Procedia Engineering* 69 (2014), 1184–1190.

[16] Thomas Stiefmeier, Daniel Roggen, and Gerhard Tröster. 2007. Fusion of string-matched templates for continuous activity recognition. In *11th IEEE International Symposium on Wearable Computers (ISWC 2007)*. 41–44.

[17] Jamie A Ward, Paul Lukowicz, and Gerhard Tröster. 2005. Gesture spotting using wrist worn microphone and 3-axis accelerometer. In *The 2005 Joint Conference on Smart Objects and Ambient Intelligence: Innovative context-aware services: usages and technologies*. 99–104.