J Intell Manuf (2018) 29:1203–1217
https://doi.org/10.1007/s10845-015-1168-8

Task recognition from joint tracking data in an operational manufacturing cell

Don J. Rude[1] · **Stephen Adams**[1] · **Peter A. Beling**[1]

Received: 16 March 2015 / Accepted: 3 November 2015 / Published online: 11 November 2015

Abstract This paper investigates the feasibility of using inexpensive, general-purpose automated methods for recognition of worker activity in manufacturing processes. A novel aspect of this study is that it is based on live data collected from an operational manufacturing cell without any guided or scripted work. Activity in a single-worker cell was recorded using the Microsoft Kinect, a commodity-priced sensor that records depth data and includes built-in functions for the detection of human skeletal positions, including the positions of all major joints. Joint position data for two workers on different shifts was used as input to a collection of learning algorithms with the goal of classifying the activities of each worker at each moment in time. Results show that unsupervised and semisupervised algorithms, such as unsupervised hidden Markov models, show little loss of accuracy compared to supervised methods trained with ground truth data. This conclusion is important because it implies that automated activity recognition can be accomplished without the use of ground truth labels, which can only be obtained by time-consuming manual review of videos. The results of this study suggest that intelligent manufacturing can now include detailed process-control measures of human workers with systems that are affordable enough to be installed permanently for continuous data collection.

Keywords Activity recognition · Machine learning · Depth camera · Kinect · Manual manufacturing processes · Hidden Markov models

✉ Don J. Rude
djr7m@virginia.edu

[1] Department of Systems and Information Engineering, University of Virginia, 151 Engineer's Way, Charlottesville, VA 22903, USA

Introduction

The idea of using technology to aid in the understanding of worker activities has been of interest since the earliest days of scientific research on manufacturing. Taylor and others used photography to help manufacturers measure and improve efficiency of manual production processes (Taylor 1913; Gilbreth and Gilbreth 1916). While it made use of the best technology of the time, Taylor's work scaled poorly because it relied on a cumbersome observational apparatus and painstaking manual review of photographs. The century following Taylor saw widespread adoption of automation in manufacturing. Even today, though, humans retain a significant role in manufacturing processes, handling 20 % of the work by some estimates (Knight 2012; IFR International Federation of Robotics 2013). Moreover, current trends in workforce demographics, such as the loss of experienced workers to retirement (Grice et al. 2011) and the leaning out of workforces (Deitz and Orr 2006), have given rise to increased interest in methods for improving manual processes. Yet even today, it is rare to have any detailed or direct measures of human activity in parts assembly and many other important processes. Current studies of manual work often involve only security cameras, checklists, and (often unreliable) work logs.

Recent developments in low cost sensing and in machine learning may open new methods for measuring and interpreting human activity on the factory floor. In the past decade, the gaming, entertainment, hobby, and mobile phone industries have spurred remarkable developments in small, cheap, lightweight sensors and computational devices to exploit them. Most notable among the new sensors is the Microsoft Kinect, a commodity-priced device that records depth data and includes built-in functions for the detection of human skeletal positions, including the positions of all major joints.

The field of machine learning has also seen major advances in the past decade that include the development of new supervised and unsupervised learning methods and novel applications of these methods to problems in human activity recognition.

If it could be achieved, automated assessment of worker activity would offer a number of important benefits. For the purposes of quality control investigations alone, it would be valuable to have detailed logs of exactly what tasks each worker executed throughout the day, and when they were completed. Further, in addition to output measures, an activity recognition system can easily provide process measures for manual processes. Process measurements are essential for rapid and precise diagnosis of quality failures; the alternative is forensic investigation of paper trails, unreliable memories, and video recordings. Even with a video system in place, it is nearly impossible to quickly find and access recordings based on a critical parameter, such as task name or part number as these are usually indexed by only a time stamp. A forensic process often requires enormous effort and can still fail to provide a clear path for permanently fixing the root cause.

This paper investigates the feasibility of using inexpensive, general-purpose automated methods for recognition of worker activity in manual manufacturing processes. A novel aspect of this study is that it is based on live data collected from an operational manufacturing cell without any guided or scripted work. The Kinect device was used to record activity in a single-worker manufacturing cell. Observations covered two shifts, each with a different worker. Joint position data was used to define input features to a collection of learning algorithms with the goal of classifying the activities of each worker at each moment in time. All the learning algorithms were general purpose in the sense that they did not use features specific to the manufacturing process, but rather relied only on worker joint positions and derivative angular and velocity features.

In the experiments performed, the accuracy of supervised learning algorithms in classifying worker activity was approximately 67 %. The quality of this result cannot be judged absolutely, as the accuracy that is necessary to support process re-engineering and management functions is specific to individual cases. Nonetheless, it appears that the accuracy is reasonable for many situations. More accurate sensors are emerging on a regular basis, which should further improve results. In the experiments, unsupervised and semisupervised algorithms, such as unsupervised hidden Markov models, show little loss of accuracy compared to supervised methods trained with ground truth data. This conclusion is important because it implies that automated activity recognition can be accomplished without the use of ground truth labels, which can only be obtained by time-consuming manual review of videos. The results of this study suggest that intelligent manufacturing can now include detailed process-control measures of human workers with systems that are affordable enough to be installed permanently for continuous data collection.

The intellectual merits of this work include the following:

1. a methodology for collecting and analyzing Microsoft Kinect data for the purposes of activity recognition ("Methodology" section),
2. a description of the requirements for an activity recognition system which is intended to be used in a live manufacturing cell ("Background research" section),
3. definitions for five key attributes in order to categorize activity recognition research and application ("Background research" section), and
4. a comparison of the accuracy for multiple algorithms and data-features, which are generalized, i.e. not tuned to this application ("Results" section).

This work is unique as compared to the majority of past activity recognition studies in that it includes these five attributes:

1. real world data,
2. continuous data which includes transitions between activities,
3. unscripted actions,
4. unsupervised or semisupervised data, and
5. a Kinect depth camera.

The remainder of the paper is organized as follows: "Background research" section gives background information on the research, "Methodology" section presents our methodology, "Models" section describes the models used in this study, "Model estimation" section outlines the training of the models, "Results" section provides the results of the different models, "Discussion" section gives a discussion of the results, and "Conclusion" section provides a summary and our conclusions.

Background research

To enable worker-centric process control, training, ergonomic feedback, etc., some critical requirements are:

1. low-cost and robust sensors,
2. generalized algorithms that require minimal supervision (for feature creation, model parameters, and ground truth generation),
3. algorithms that are robust to a variety of sensing conditions and human variability, and
4. algorithms which provide outputs that are accurate, valuable to an end-user, and interpretable.

These four requirements touch on several areas of research, and present conflicting goals that require careful balance. In particular, the mathematical models employed often trade supervision for accuracy or interpretability (Cohen et al. 2002; Nigam et al. 2000).

To begin, we define five attributes of activity recognition research. The first is how data are collected. Most publicly available data sets are collected in a laboratory environment or a staged setting, as opposed to the real world. Laboratory data generally consist of predefined actions that are repeated several times by several subjects. Data are often manually segmented, meaning that only one action is classified at a time, and there are no transitions between actions. In addition, laboratory data lack random or unknown actions that are often present in data collected from real-life activities and continuous recordings. This leads to our second and third attributes: continuous data versus segmented data and scripted actions versus unscripted actions. The fourth attribute is the type of algorithm—specifically, supervised algorithms versus unsupervised and semisupervised algorithms. Supervised algorithms use the labeled activity when training the model, while unsupervised algorithms do not use labels. Semisupervised algorithms use a combination of supervised and unsupervised data (with an emphasis on minimizing the proportion of supervised data). The final attribute is the type of sensor. We distinguish between a depth camera and all other types of sensors, which include, but are not limited to, body-worn, video, and microphones.

Our work also fills a gap in the literature concerning 3-D depth cameras, and specifically, the Microsoft Kinect. A significant proportion of the research to date has combined depth cameras and activity recognition to focus on supervised algorithms that use data collected in a laboratory or other staged setting (Chen et al. 2015; Wang et al. 2012; Packer et al. 2012; Oreifej and Liu 2013; Yang and Tian 2012; Xia et al. 2012). [Zhang and Parker (2011) use an unsupervised learning algorithm with Kinect data, but create their own segmented, scripted, laboratory data set.] On the other hand, some studies have used unsupervised or semisupervised algorithms and real-world data with other types of sensors (Krause et al. 2003; Wang et al. 2009; Stikic et al. 2008; Niebles et al. 2008; Mahdaviani and Choudhury 2008).

Table 1 lists the attributes of several studies, including those cited in the previous paragraph, with a column of indicators for each attribute. The methodology we propose is the only study that contains all five attributes.

Some papers or attributes in Table 1 are not perfectly segmented. For instance, credit is given to Stikic et al. (2008) for using real-world data; the authors collected data on a couple who lived in an instrumented home environment for 10 weeks. They were not given instructions as to what types of activities to perform, only to continue to live as normal a daily life as possible. While we classify this as real-world data, it was not collected in a real-world setting, due to the intrusiveness of on-body sensors. Niebles et al. (2008) used video segments of figure skaters. While the activities were not scripted, the authors did preselect video segments for

Table 1 Attribute table for existing work in activity recognition

References	Real data	Continuous data	Unscripted	Unsupervised/ semisupervised	Kinect
Chen et al. (2015)		x			x
Zhang and Parker (2011)				x	x
Sung et al. (2011)					x
Yang and Tian (2012)					x
Xia et al. (2012)					x
Wang et al. (2012)					x
Oreifej and Liu (2013)					x
Huikari et al. (2010)		x			
Koskimaki et al. (2009)		x			
Krause et al. (2003)	x	x	x	x	
Niebles et al. (2008)	x		x	x	
Stiefmeier et al. (2006)					
Ward et al. (2006)		x			
Stikic et al. (2008)	x	x	x	x	
Wang et al. (2009)	x	x	x	x	
Mahdaviani and Choudhury (2008)	x	x	x	x	
Our proposed methodology	x	x	x	x	x

"x" indicates that the research includes the attribute

analysis. We labeled this work as having unscripted data, but this distinction is not always so clear.

We believe that the data we present in this paper are unique, because the data set collection was unscripted, collected in a real-world setting, with a depth camera, from continuous motion, and includes several (continuous) transitions between tasks. Semisupervised training algorithms are then employed for task recognition.

Sensors

Sensors are required to gather useful observations of workers. These observations exist to provide information about worker activity, such as the number of times a task has been executed and the duration of individual tasks. These measures can be particularly meaningful when they are collected in the context of other knowledge already available—for instance, in a manufacturing setting, data on factory output volume and quality are already collected and give contextual meaning to the task measurements. Fortunately, a variety of new sensor types are now being sold as commodities. For example, Microsoft created the Kinect for the Xbox gaming system, which offers highly effective tracking of human joints (Shotton et al. 2011) and distance measurements for a fraction of the cost of earlier time-of-flight and laser-ranging systems. This sensor is robust for indoor settings, even those in which traditional computer vision systems become compromised due to something as trivial as a change in lighting or clothing.

Ground truth

Minimally supervised algorithms are essential. Previous studies, such as Chen et al. (2015), Wang et al. (2012), Packer et al. (2012), Oreifej and Liu (2013), Yang and Tian (2012), Xia et al. (2012), have focused on data collected in a controlled experimental setting to create person-agnostic, static models using supervised data. The strictly controlled "lab setting" allows precise control of variables, but limits the types of data that can be collected. In addition, the performance of study participants is often affected by the act of being observed and by being in an unfamiliar setting. In contrast, participants who are in their normal work environment often remain focused on their primary job if observers and sensors are minimally intrusive. While these models (trained from a controlled setting) have been shown to be effective, their use requires significant up-front training, and it is currently unknown if they may require further training to cope with varying sensing environments and worker tasks which occur in real-world settings. To avoid these limitations, this study focuses on models that provide useful information, but are trained in an unsupervised or semisupervised manner using real-world data. Ideally, our models will prove to be effective despite noisy sensor conditions, potential interference from uncontrolled light sources, and any spurious detections.

Ground truth is time consuming—i.e. expensive—to obtain for collections of significant size. Coding in this study required 2× to 5× real time. For ongoing, continuous data collection, coding could take days or even months to code just a few hours of real-time. Further, incorrectly labeling the ground truth can result in models that perform poorly when supervised training is implemented. Ground truth is often labeled using video or other added sensors to annotate a collected data set. In some settings, such as hospitals or private homes, video might not be desired or even allowed due to privacy laws or other restrictions. In Stikic et al. (2008), the authors demonstrate the feasibility of semisupervised learning on an activity recognition data set to resolve the conflict between supervised and unsupervised methods.

Data collection requirement

Another of our major requirements is that the data set be collected in its natural setting rather than in a lab environment. While the latter can allow for very precise control of experimental variables, it misses both the variance that occurs naturally in human tasks and the subtle changes in performance or behavior that occur when workers are being actively or passively critiqued by their observers. In the manufacturing literature on activity recognition (Chen et al. 2015; Huikari et al. 2010; Koskimaki et al. 2009; Stiefmeier et al. 2006; Ward et al. 2006), data were collected in a simulated environment and not at an in-production facility. Also, these studies used on-body sensors—and, in some cases, other sensors such as microphones—to collect data on the worker. (Microphones would not be feasible in a production facility with several operational cells, due to interference from machinery.)

State-of-the-art methods

Several state-of-the-art research papers on activity recognition using a Kinect tested their algorithm and customized features on data sets collected by Microsoft Research (MSR) (Wang et al. 2012; Oreifej and Liu 2013; Yang and Tian 2012; Xia et al. 2012). However, these data sets are composed of a specific set of subjects performing predefined and limited actions repeatedly. They are also segmented, with only one task being performed per sequence of data; in contrast, in a real-world setting, data are collected continuously, with several tasks taking place in succession. Therefore, these advanced algorithms have unknown performance characteristics in real-world settings, and will require large amounts of supervised data to be created for each new task encountered in the field.

The authors of these studies (Wang et al. 2012; Oreifej and Liu 2013; Yang and Tian 2012; Xia et al. 2012) each designed and implemented a different set of custom features that require significant preprocessing, and no feature selection is performed. Each justifies the use of a given custom feature by demonstrating better performance than that documented in an earlier study based on the MSR data set. Furthermore, there is no accepted best feature in the area of activity recognition. This illustrates the need for feature selection in addition to traditional model training.

Huikari et al. (2010) compare feature selection and instance selection on data collected on a simulated industrial assembly line. For feature selection, they used principal components analysis (PCA) and sequential forward selection (SFS). For instance selection, data were randomly reduced and then tested. PCA is an unsupervised method, yet performs worse than SFS, which requires supervised data. Instance selection performs better than PCA but worse than SFS. To preserve the correct proportions of activities in the training set, supervised data are required for instance selection. In either case, however, these algorithms must be used and, possibly, tuned in addition to the hidden Markov model (HMM).

Methodology

Our methodology includes the following steps:

1. Collect data using a Microsoft Kinect.
2. Remove some clearly erroneous measurements from the data set.
3. Label the entire data set for the purposes of evaluating the classification algorithms.
4. Calculate a set of derived features.
5. Apply classification algorithms which fit stated requirements.
   (a) Select the size of the initialization set N_{start} and remove it from the training data.
   (b) Calculate initial parameters from initialization set.
   (c) If applicable, select hyperparameters for prior distributions.
   (d) Perform unsupervised learning with selected classification algorithm using training set.
   (e) If applicable, reduce the feature set and construct reduced models.
6. Map tasks to classifier outputs.
7. Calculate performance metrics for each classifier.
   (a) Predict tasks for the test set.
   (b) Calculate all accuracy measures.

Data collection

Here the data collection process is summarized. For more details concerning data collection, see Rude et al. (2015).

The Microsoft Kinect uses a structured infrared light field to calculate depth data (Freedman et al. 2010) and provides human skeletal tracking (10 points for upper-body mode and 20 points for full body) (Shotton et al. 2011) up to 30 times per second over a living-room-sized area. This type of sensor allows for nonintrusive data collection, because it does not require any modification to the environment or people and the infrared light method is robust in most indoor environments. Because the Kinect has been designed as a human computer interface, its skeletal tracking is of sufficient quality to support tracking and gesture recognition. Examples of depth data and skeletal tracking from the actual data collection are shown in Figs. 1 and 2.

Data collection consisted of the Kinect joint tracking and a security camera video recording that was solely used for manually coding ground truth for worker tasks. The sensor suite was installed during a routine production stoppage and supervised by plant safety officers.

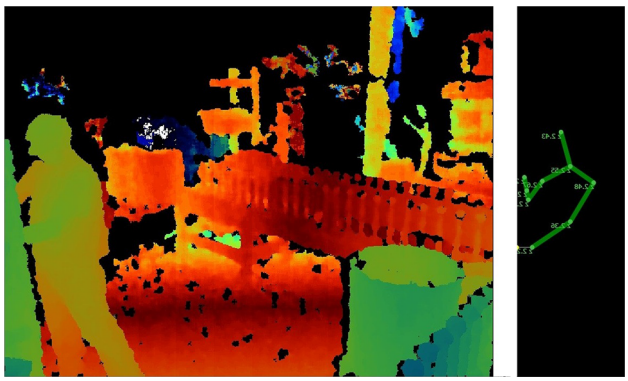

Fig. 1 Depth data mapped to colors and tracked joints (*inset*), as provided by the Kinect sensor during the *paint* task

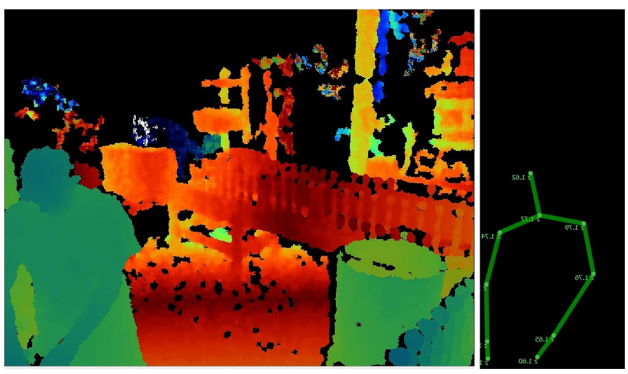

Fig. 2 Depth data mapped to colors and tracked joints (*inset*), as provided by the Kinect sensor during the *dry* task

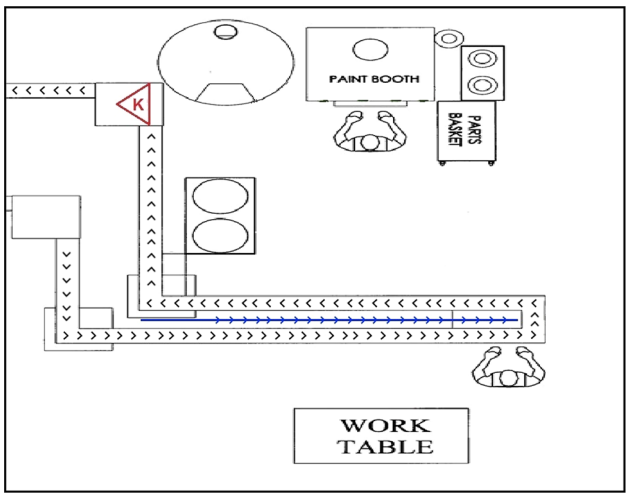

Fig. 3 Factory floor for the "in-cycle" area including a *red triangle marker* for the Kinect's position, very near the large circle which is the drying rack. The *blue line* is the location of the short conveyor for printing serial numbers immediately prior to boxing

Worker tasks

In brief, the factory floor job analyzed is referred to as the "in-cycle" work and is usually executed by a single worker in the area shown on the floorplan in Fig. 3. The primary job is to paint parts (interior and exterior), apply a serial number, and box the finished parts. A flow chart for this work sequence is displayed in Fig. 4. While this is treated as a one-person job, other workers often help with the serial numbering or box assembly to allow the in-cycle worker to maintain good work flow. Additional workers are also required to maintain other machinery and provide the prepped "raw" parts to the in-cycle worker. The subtasks of parts painting will now be described in detail so that sensor placement and data will be in the appropriate context. The four primary tasks of in-cycle work will be referred to by their ground truth coding labels of *fetch*, *paint*, *dry*, and *load*. Subtasks labeled as *load-serial* and *box* are grouped with the primary task of *load*, while walking between stations or any other tasks are labeled *unknown*. For additional details regarding the worker tasks, see Rude et al. (2015).

To capture normal variability in the workplace, no attempts were made to modify work flow, tasks, or machinery, or to interact with the workers. While many different activities were observed, some were not strictly related to the work being documented or "normal" operations. For example, brief interruptions for worker-to-worker coordination or certain one-time machine-maintenance tasks were considered to be outside the scope of the primary job and were not included in the task analysis. Instead, these activities were lumped into a single task named *unknown*. Because of the Kinects temporal fidelity, movement between stations was sampled repeatedly and also labeled *unknown*.

Under nominal conditions, a worker is seen executing the following sequence: fetch four parts, *paint* two, *dry* two, *paint* two, *dry* two, and *load* four. This cycle can be completed in as little as 30 s and, given the Kinect sampling rate, would result in approximately 900 samples of each of the 10 joints as the worker moves from station to station around the manufacturing cell.

Data description

Data from a single full workday were used, resulting in approximately 7 h of recording due to worker break times

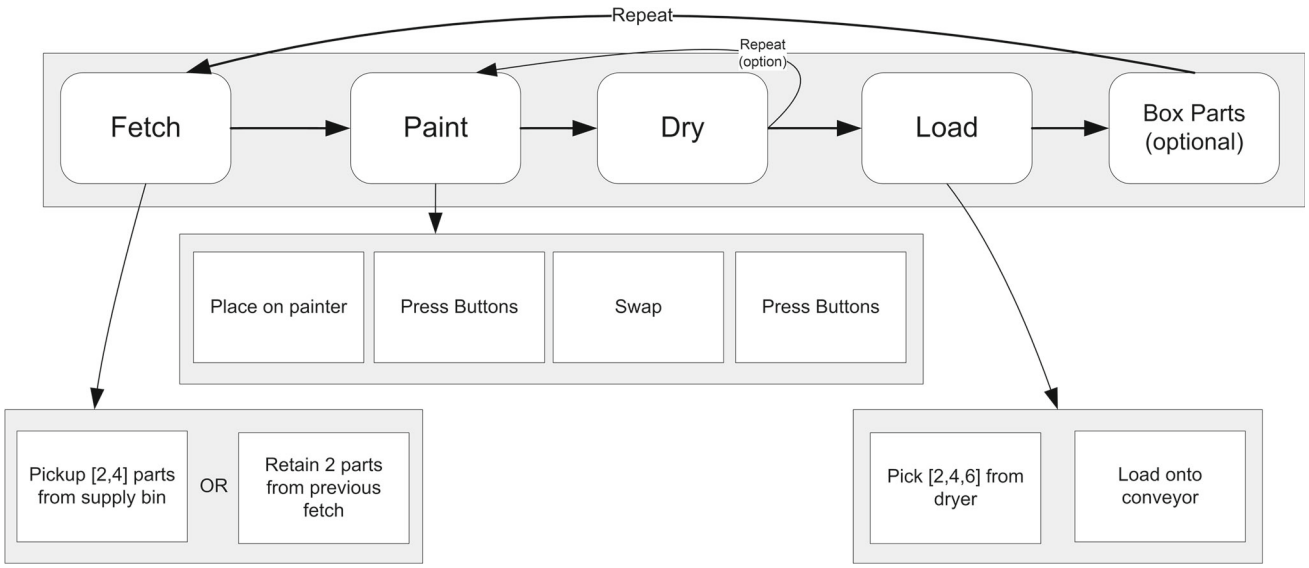

Fig. 4 A flowchart outlining the primary in-cycle tasks and some additional execution details

(0800 and 1400). For the day of this recording, one worker covers the first half of the day and a second worker begins around noon. The task sequences are generally the same for both workers, but there are changes throughout the day due to the state of the automated paint line. For example, the morning worker tends to load the line at a location closer to the drying rack and the afternoon worker loads the line closer to the boxing station. Other random events and variations, such as maintenance of clogged paint nozzles after 1300, occur throughout the collection. The 1200–1300 time frame was selected as a representative sample of the work, as it captured a transitional time before the second worker had established a rigid work flow. Overall, the manual tracking and coding of the workers observed approximately 3200 work tasks being executed (this ignores *walk* and other *unknown* events) and around 350 fully nominal work cycles, as described in the tasks section.

Remove erroneous measurements

A false skeleton detection occurred repeatedly in the sensor's field of view at an impossible location, possibly due to reflections off a wire mesh basket. For example, the shoulder-center point corresponding to these detections had an average location of $(-1.41, -0.12, 3.36)$. Given the Kinect sensor location, this would place the "person" somewhere beneath the main parts conveyor or possibly under the floor despite the fact that the surface of the basket was only about 1.25 m away, see Fig. 5. In early model runs, this spurious detection was always placed in its own hidden state and was therefore removed before processing, rather than being listed as a valid detection of an invalid data point.

This data set was used after removing the well-documented spurious skeleton detections and rotating the points (around the x-axis) to remove the Kinect's rotation relative to the floor. In total, approximately 500,000 samples were recorded across the seven work hours.

Label data to establish ground truth

A set of security cameras also recorded the in-cycle work area, which provided a method for creating ground truth with time resolution similar to that of the Kinect. Ground truth was coded by researchers watching the video at 1/2 to 1/5 normal speed and selecting task labels as the in-cycle worker moved from task to task. These labels were selected based on documentation of observed tasks such as the flowchart shown in Fig. 4.

Derived features

In addition to the raw joint positions, several features were derived for use in the model. These each attempt to more directly encode information about joint positions relative to the body, rather than the dimensions of the recorded scene. For example, joint angles or the distance between the hands. Features like this could provide the HMM with complicated or nonlinear information due to the kinematics of the human body. For a full definition of each of the 22 derived variables, see Rude et al. (2015). This results in 52 features to be used in the full models.

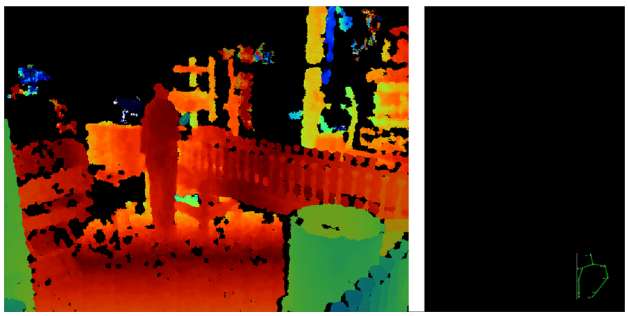

Fig. 5 Depth data mapped to colors and tracked joints (*inset*), as provided by the Kinect during erroneous detection

When derived features are added to collected joint positions, the size of the data set can grow rapidly due to their sampling rate. The large data and prediction processing times are dealt with in two ways. First, the skeletal tracking feature offered by the Kinect is used instead of the raw depth data. This decreases the amount of data collected significantly. Second, preference is given to classification algorithms that perform feature selection. Feature selection reduces calculation time, improves model accuracy, and can improve the ability to interpret the model due to simplified models. Further, feature selection is an important part of mining manufacturing data (Rokach and Maimon 2006).

Classification algorithms

An HMM (Rabiner 1989) is expected to be effective in our continuous-motion setting, due to the sequential yet repetitive nature of factory work. In general, an HMM is a widely used probabilistic model which consists of an unobserved sequence of states X and a sequence of observable emissions Y. The distribution of the emissions is conditional on the state, and the emissions are considered independent given the state. A graphical model of an HMM is shown in Fig. 6. In our implementation, hidden states correspond to individual tasks and emissions correspond to the data collected. This differs from most work in activity recognition (Stiefmeier et al. 2006; Ward et al. 2006; Xia et al. 2012) and manufacturing (Xu and Ge 2004) using HMMs, in which a separate HMM is trained for each class, which then requires supervised data. HMMs can be easily trained using unsupervised learning methods, a requirement for this study.

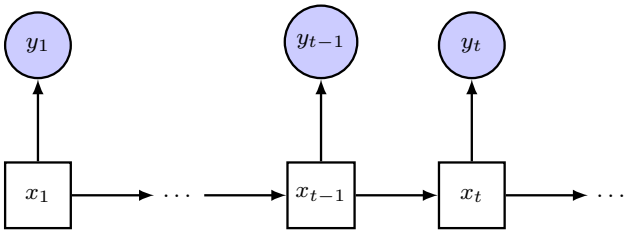

Fig. 6 Graphical model for HMM. *Squares* are hidden variables and *circles* are observed variables

Rokach et al. (2008) point out two drawbacks to using HMMs in their problem. First, the structure of the model is very important to the quality of the classifier, and there are many possible model structures. As previously stated, a task label is assigned to each of the hidden states and the collected Kinect data to the emissions of these hidden states. This allows for easy interpretation of the model and a single HMM to model the entire activity recognition problem. This relatively simple interpretation of the HMM will remove the second drawback stated in Rokach et al. (2008): "the model's meaning is unclear". In future work, the values of the estimated model parameters and the selected features can be used to help plant managers improve quality and diagnose problems.

We propose testing four types of HMMs. A supervised HMM is used as a baseline and a standard unsupervised HMM using all features is trained and tested. Two HMMs that perform feature selection are also tested. More information on the HMMs is given in "Models" section. In addition to HMMs, we propose two widely implemented unsupervised classification algorithms: unsupervised naive Bayes classifier and K-means classifier. Both of these algorithms treat each data-point independently whereas HMMs take into account their ordering.

Mapping model outputs

One of two major challenges when using HMMs to model task recognition data is mapping the states of the HMM to specific task. If supervised training data is available, one option is to map states based on majority rule. The states of the training set are predicted using the trained HMM, and the state with the greatest number of correctly classified tasks is assigned. This method however defeats the purpose of implementing unsupervised learning because the supervised training data is available. A small portion of the training data could be reserved and labeled for this purpose, but there are further issues with this method. Namely, it relies upon the accuracy of the trained model for mapping. Another option given supervised training data is to test the accuracy of every possible state combination. This has similar advantages and disadvantages as the previous mapping method but the number of possible state assignments to test grows rapidly as the number of states increases.

The second major challenge when implementing unsupervised learning with HMMs is choosing the initial parameters for the model. One option is to randomly select all initial values. However, the unsupervised learning algorithm for HMMs is sensitive to the initial values, and poor starting values can lead to poor parameter estimates. Generally, if random starting values are used, multiple runs of the algorithm with different random initial parameters are required in order to properly characterize model accuracy.

To combat both of these challenges an initialization set is used. A small portion (approximately 1 %) of the training data is reserved and assumed to be supervised. Initial parameter values are calculated from the initialization set. The mapping in the initialization set is used to map states to tasks in the learned model. For example, if state 1 in the initialization set represents task 1 (e.g. *fetch*), then state 1 will also represent task 1 for the trained HMM. This is a reasonable assumption as the initial values should be relatively close to the estimated parameters.

Models

Model parameters for a standard HMM are the initial state distribution, the state transition probabilities, and the distribution for the emissions. Assuming I tasks and T time steps, the initial state distribution is denoted as $\pi_i = \mathbb{P}(x_1 = i)$ and the transition probabilities as $a_{ij} = \mathbb{P}(x_t = j | x_{t-1} = i)$. It is also assumed that the emission distribution is Gaussian with mean μ and standard deviation σ. When the state sequence is known, these parameters can be calculated directly from the data. When X is unknown, model parameters must be estimated using an algorithm. The most common estimation algorithm is the expectation maximization (EM) algorithm, often referred to as Baum–Welch when applied to HMMs. The number of hidden states must be known a priori when performing EM. The joint probability of X and Y is:

$$p(X, Y) = \pi_{x_1} f_{x_1}(y_1) \prod_{t=2}^{T} a_{x_{t-1}, x_t} f_{x_t}(y_t), \tag{1}$$

where f_{x_t} is the state-conditional Gaussian distribution.

When numerous features are collected, it is likely that some will not be useful for activity recognition. Noisy features, for instance, can confuse activity recognition models and degrade accuracy. A softer requirement for our case study is some form of feature selection. One possible approach is to learn an HMM for every possible subset of features, and select the model with the highest predictive accuracy on a withheld subset of data. This would require supervised

data to train the model, and quickly becomes impractical as the number of feature subsets grows as a factorial of the number of features. The feature saliency HMM (FSHMM) (Adams 2015) simultaneously estimates model parameters and selects features. The maximum a posteriori (MAP) formulation of the FSHMM is used for this study. Zhu et al. (2012) use a similar formulation as in Adams (2015), but employ a variational Bayesian (VB) estimation algorithm. The VB algorithm can be used when the number of states is unknown, but significantly increases the complexity of the model and the estimation process and can decrease the model's accuracy. We will use both the FSHMM and the VB HMM here for comparison.

The FSHMM recasts the feature selection process as a parameter estimation problem by adding *feature saliencies* to the model. The feature saliencies ρ_l can be interpreted as the probability that the lth feature is relevant and helps distinguish between states. Let y_{lt} denote the observation of the lth component at time t. The likelihood of the emissions is expanded to a mixture of state-dependent Gaussian distributions and state-independent Gaussian distributions. A third binary random variable Z is added to the model. If $z_l = 1$ the lth feature is relevant; otherwise, the feature is irrelevant. The joint probability of X, Y, and Z is:

$$p(X, Y, Z) = \pi_{x_1} p(y_1, Z|x_1 = i) \prod_{t=2}^{T} a_{x_{t-1},x_t} p(y_t, Z|x_t = i), \quad (2)$$

$$p(y_t, Z|x_t = i) = \prod_{l=1}^{L} [\rho_l r(y_{lt}|\mu_{il}, \sigma_{il})]^{z_l} [(1 - \rho_l) q(y_{lt}|\epsilon_l, \tau_l)]^{1-z_l}, \quad (3)$$

where ϵ is the state-independent mean and τ is the state independent standard deviation.

Four HMMs are compared, starting with the training of a HMM on supervised data to establish a baseline for performance. A standard HMM (using EM), an FSHMM, and a VB HMM are all trained on unsupervised data. In addition, two nontemporal models are trained and tested on unsupervised data—naive Bayes and K-means. The ground truth has been established by examining video evidence after the fact, and therefore can provide precise measurement of model accuracy. However, in a final system design the ground truth would rarely, if ever, be available.

The goal of this case study is to compare algorithms that can be easily implemented on minimally supervised data collected in an in-production manufacturing cell. A second goal is to show that unsupervised algorithms can be used in place of a supervised algorithm with reasonable losses in accuracy. Perhaps more importantly, the results will show the general applicability and usefulness of this system design in a real-world setting, as it provides tracking at a level of abstraction useful to plant managers while meeting our four stated goals.

Model estimation

Six task-recognition models are compared: supervised HMM, unsupervised HMM (Rabiner 1989), FSHMM (Adams 2015), VB HMM (Zhu et al. 2012), unsupervised naive Bayes (Murphy 2012), and unsupervised K-means (Murphy 2012). The supervised HMM is trained by counting transitions and calculating state dependent means and variances for a Gaussian distribution. The unsupervised HMM, FSHMM, naive Bayes model, and K-means models are trained using EM. The VB HMM is trained using a variational Bayesian algorithm. The class-conditional distribution for the naive Bayes model is also assumed to be Gaussian.

An hour's worth of data—85,765 observations—are used to train each of the models. Data were collected in the middle of the workday, from 1200 to 1300 hours. This hour was chosen for the training set because it contains all relevant tasks and is an accurate representation of a typical work cycle. The models are tested on 12 half-hour data sets collected throughout the rest of the workday. The number of observations for each test set is shown in Table 3, column 2.

While the methods investigated here are technically semi-supervised, due to the use of a supervised subset of data for initialization of the training algorithms, the algorithms can be implemented in an unsupervised fashion by randomly choosing initial model parameters. We used the semisupervised approach in the experiments for ease of labeling model outputs and to ensure good and comparable starting points across models.

Initial values for the model parameters are needed when performing unsupervised learning. For a fair comparison of the models and consistency in matching the hidden states across models, initial estimates are calculated using the first 1000 observations of the training set rather than a random initialization. The first 1000 observations are chosen for the initialization set so that each of the five tasks are represented at least once. These observations in the initialization set are assumed to be supervised. The initial values for the features saliencies in FSHMM and VB HMM are 0.5, which sets the probability of a feature being relevant equal the probability that a feature is irrelevant.

The FSHMM hyperparameters are $\alpha = 2, \beta = 1, m = \mu_{init}, s = 0.25, \zeta = 2, \eta = 1, b = \epsilon_{init}, c = 0.25, \nu = 2, \psi = 0.5, k = 20{,}000$. α is chosen so that every transition is possible in the learned parameters. k is roughly $1/4$ the number of observations in the training set and as shown in Adams (2015) a good choice for larger data sets. The means of the priors on μ and ϵ are the initial values of μ and ϵ because this is a logical estimate given the supervised initial-

ization set. Other choices for hyperparameters could include random values between the minimum and maximum of each feature. The rest of the hyperparameters for FSHMM are chosen to minimize their effect on the estimated parameters. The hyperparameters for VB HMM are the same as in Zhu et al. (2012), where the goal is to minimize their effect on the learned parameters. The choice of hyperparameters in Bayesian analysis is not an exact science and can greatly effect the estimation. Nearly any set of hyperparameters could be justified and different applications could call for different choices.

For the FSHMM and the VB HMM, features with an estimated saliency less than 0.9 are removed from the model. The comparison models, excluding the VB HMM, cannot perform feature selection without the aid of another algorithm. Reduced models are built for the comparison by using the feature subset selected by the FSHMM and VB HMM. This will show that the FSHMM selects relevant features and that the FSHMM formulation is suitable for feature selection.

Results

Descriptive results

One simple and descriptive measurement that can be made from the raw data is to estimate the distance traveled by the worker(s) throughout the day. Because of the accuracy and fast update rate of the Kinect, distance traveled can be calculated every 1/30th of a second. Whether using the head or shoulder-center point, the result will be an overestimate of total distance walked due to upper-body movements insignificant to actual walking and some rounding errors. For this data set, the distance traveled was calculated to be around 8000 m—not unreasonable, since the work space was roughly 3 m by 3 m and at least 3200 tasks were recorded over 7 h.

Distance traveled per task, shown in Table 2 for each task, is expected to be useful to a plant manager, safety officer, or process engineer. While this is a relatively simple measure to calculate using tracking data, it provides the richest information when task labels are available. Table 2 also shows the number of times each task was observed being executed by the workers. While this data set is limited to a single factory, our algorithms could be required to identify hundreds or thousands of individual tasks in the midst of continuous motion.

Table 2 Occurrence counts and distance traveled, in meters, per task obtained from the ground truth

Task	Travel (m)	Count
Fetch	740.18	1195
Paint	1937.37	515
Dry	1299.93	513
Load	792.00	1130
Unknown	3222.98	2701
Total	7992.45	6054

Model accuracy results

Each model has its own fitting and prediction steps, as described or referenced above. All models were given the same input data, the 1200–1300 hours, for unsupervised training (supervised only for Sup. HMM). For the sake of completeness, this hour of data is also scored for point-prediction accuracy, indicated by bold rows, but not included in the average accuracy score for each algorithm.

For the HMM models, the estimated task sequence is calculated using the Viterbi algorithm. The expectations for approximate distributions are used for point estimates for VB HMM model parameters. The task with the highest probability is assigned for the naive Bayes model, and the task with the minimum distance from the task mean is predicted for the K-means algorithm.

The fraction of correctly classified tasks, compared to the ground truth, will be referred to as point-prediction accuracy or simply accuracy. The accuracy for each half-hour test set, as well as total accuracy over all test sets, is shown in Table 3 for each model.

During feature selection, 36 of the possible 52 features are removed by the FSHMM, while only 25 of the features are removed by the VB HMM. The features included in the reduced FSHMM are Head X, Head Y, Head Z, Shoulder Center X, Shoulder Center Z, Shoulder Left X, Shoulder Left Y, Shoulder Left Z, Elbow Left X, Elbow Left Z, Wrist Left X, Hand Left X, Shoulder Right X, and Elbow Right X—all from the raw features—as well as Dist and Left Hand Size from the derived features. The VB reduced model includes the features previously listed, excluding Left Hand Size, plus Shoulder Center Y, Elbow Left Y, Wrist Left Z, Hand Left Z, Shoulder Right Z, Elbow Right Y, Elbow Right Z, Wrist Right X, Wrist Right Z, Hand Right X, Hand Right Z, and Right Elbow Over Shoulder. Only the predictions from the reduced model are given for FSHMM and VB HMM.

Multiple measures of worker activity can be calculated from the model output, which could be valuable for a plant manager. In this data set, tasks are performed for time periods much longer than the sampling rate of the Kinect, which results in many observations for each task. To support these measures, a model that does not oscillate between different predictions unnecessarily is preferred. For instance, if *fetch* is being performed for 50 time steps, a model that predicts the first 25 steps as *fetch* and the second 25 as *paint* would be preferable to a model that alternates between *fetch* and *paint* multiple times in the same period. One way to assess this is by calculating the number of individual tasks—that is, how many times any task is performed continuously.

Table 3 Point prediction accuracy for test sets for each half hour and average accuracy for all test data

Test set (time)	Observations	Sup. HMM	Unsup. HMM	FSHMM	VB HMM	Naive Bayes	K-means
0700	34131	0.5381	0.5444	0.5461	0.5429	0.5516	0.5384
0730	42441	0.5073	0.5325	0.5262	0.5199	0.5042	0.5105
0830	40349	0.6676	0.6962	0.6920	0.6892	0.6971	0.6752
0900	41156	0.6897	0.6821	0.6875	0.6781	0.6770	0.6726
0930	37465	0.6376	0.6090	0.6218	0.6124	0.6807	0.6287
1000	40032	0.6588	0.6792	0.6879	0.6761	0.6594	0.6650
1030	42128	0.6034	0.6244	0.6494	0.6461	0.6044	0.6471
1200	**42603**	**0.7739**	**0.8011**	**0.7947**	**0.7930**	**0.7641**	**0.7682**
1230	**44162**	**0.6795**	**0.6476**	**0.6430**	**0.6425**	**0.6750**	**0.6542**
1300	35421	0.6083	0.3942	0.4603	0.4402	0.3575	0.4434
1330	42397	0.8016	0.7681	0.7483	0.7411	0.7160	0.7238
1430	44006	0.7993	0.7888	0.7560	0.7644	0.7337	0.7437
1500	40928	0.7873	0.7913	0.7968	0.7884	0.7935	0.7829
1530	14930	0.7635	0.7022	0.7036	0.7003	0.7842	0.7243
Average	455384	0.6703	0.6539	0.6583	0.6520	0.6436	0.6465

Bold rows indicate training accuracy and are not included in the average

Table 4 Total number of individual tasks per model and in the ground truth

Test set (time)	Truth	Sup. HMM	Unsup. HMM	FSHMM	VB	Naive Bayes	K-means
Total	4930	6873	6438	5440	5557	6463	6349

Table 5 Average time in seconds per task for the ground truth and each model

Task	Truth	Sup. HMM	Unsup. HMM	FSHMM	VB	Naive Bayes	K-means
Fetch	2.59	2.20	2.70	2.36	2.22	1.44	1.58
Paint	6.54	4.69	5.09	6.25	5.85	4.41	5.69
Dry	1.70	2.40	1.84	2.22	2.30	2.85	2.84
Load	2.61	2.78	4.58	4.45	4.35	3.46	2.36
Unknown	2.42	0.97	0.99	1.15	1.10	1.15	0.79
RMSE	NA	1.11	1.27	1.04	1.07	1.38	1.07

Root mean square error (RMSE) is calculated between each model and the ground truth

In terms of the HMM, this means treating all hidden state "self-transitions" as a single execution of that particular state. Other models may not contain a concept of state transition, but each datum produced by the Kinect must be classified by the model, which produces a similar stream of task labels that must be simplified by looking at contiguous blocks of task labels. In the above example, this means that the contiguous block of 25 estimated *fetch* tasks is counted as a single *fetch*, and its elapsed time can be measured by looking at the time stamps of the first and last datum.

Table 4 contains the sum of individual tasks over all test half hours for the ground truth and each model. Another example measure is the average time workers spend in each task, which is shown in Table 5. The table shows the statistics for the ground truth and the estimated task sequences for each model on the test set. The root mean squared error (RMSE) is calculated between each model and the ground truth. FSHMM produces the lowest RMSE for this measure.

Table 6 shows the number of specific state-to-state transitions occurring in the ground truth and each model's output, while ignoring the *unknown* state. This enhances representation of the transition probabilities that were estimated by the Markov models and may offer a method for identifying erroneous task ordering. *Unknown* is excluded from this calculation, because it is essentially a null category that includes walking between work stations and any unknown activity. When workers are able to follow standard procedures, the transitions with the highest occurrence rates will correspond to the steps of the nominal sequence.

Discussion

The system constructed in this study—in particular, the FSHMM–has been shown to meet the requirements outlined in "Background research" section. A low-cost sensor, the

Table 6 Number of transitions between tasks excluding *unknown* for ground truth and predicted by each model

Transition	Truth	Sup. HMM	Unsup. HMM	FSHMM	VB	Naive Bayes	K-means
Fetch–paint	421	677	653	516	595	959	583
Fetch–dry	0	0	0	6	15	32	53
Fetch–load	2	0	0	20	23	3	363
Paint–dry	906	936	926	924	913	910	935
Paint–load	5	9	17	2	0	0	0
Dry–load	364	383	450	454	468	115	204
Paint–fetch	4	239	187	121	195	558	227
Dry–fetch	98	140	97	57	53	349	271
Dry–paint	462	459	427	473	457	496	531
Load–fetch	319	287	369	365	386	84	500
Load–paint	29	49	48	55	53	11	46
Load–dry	21	51	52	58	54	22	21
RMSE	NA	102.53	91.43	55.88	84.91	254.81	158.98

Root mean square error (RMSE) is calculated between each model and the ground truth

Kinect, provides data to a generalized model that can be run with little or no supervised data, minimal parameter tuning, and minimal (or no) custom data features. The sensing and outputs are shown to be robust to a challenging real-world data collection with continuous human activities that include many uncategorized or unexplained actions. Finally, it was shown that the system outputs are very similar in interpretation and accuracy to that of hand-labeled ground truth.

While the literature addresses many forms of activity tracking (and associated models), it is extremely rare, as discussed earlier, to find any that collect data from an operational manufacturing environment. In addition, the Kinect is a relatively new device that originated in the entertainment industry, and therefore has not yet undergone significant crossover to intelligent manufacturing research. For limitations of the Kinect system and the configuration used in this study please see, Rude et al. (2015), and for accuracy details of the Kinect depth sensing, see Choo et al. (2014), Landau et al. (2015). This case study, while limited to a single factory, shows the feasibility of low-cost worker tracking systems. Furthermore, while we focused on generating accurate task labeling, many other applications can be extended from this level of tracking, including, among others, robot interactions and ergonomic safety tracking.

Some of the primary productivity measures seen in the manufacturing setting endeavor to match worker efforts to cell- or line-level outputs, quantity, and/or quality. In some cases, worker tracking can be as simple as time sheets to relate man-hours to production, and in others it might include computer terminals where workers enter notes on quality or reasons for line stoppages. Our results demonstrate the possibility of correlating highly detailed records of both worker movements and more abstract task labels. This is what enables us to obtain measures such as number of tasks, number of nominal cycles, tasks per hour, distance per task, and distance per hour. Any of these could be used in a one-time study, but this system configuration should allow for continuous daily—or even near-real-time—studies of changes to the production process. While this type of information is clearly useful for process optimization, many other potential measurements with more exotic applications are outside the scope of this study; for example, real-time ergonomic feedback, time-integrated repetitive stress measures, expertise modeling, and worker training.

The FSHMM gives the highest accuracy of the unsupervised methods and outperforms all other models, including the supervised HMM, in terms of the number of individual tasks, average time per task, and number of transitions. While the difference in accuracy between the FSHMM and the other unsupervised models is not very high due to test size, the difference in the number of correctly classified observations is significant.

The supervised HMM produces the highest accuracy of all the models, but requires a significant amount of supervised data. The supervised HMM produces estimated sequences with the highest number of individual tasks (furthest from the ground truth), which indicates a significant amount of spurious switching between states.

The VB HMM is the other model that incorporates feature selection. It is the worst HMM in terms of accuracy, but outperforms naive Bayes and K-means. For number of individual tasks, the VB HMM is the second closest to ground truth behind FSHMM.

The models that do not take into account transitions between states, Naive Bayes and K-means, do not perform as well as the HMMs in terms of accuracy or the other three metrics. We conclude that transitions between tasks are a critical component for predicting tasks in a manufacturing cell.

In this study accuracy has been the primary point of comparison between supervised and unsupervised algorithms. As discussed, our numerical results suggest that when using unsupervised methods only a small accuracy penalty is incurred. However, computation time is also an important consideration. The training of supervised methods requires only the calculation of frequency statistics; therefore, they are generally more time efficient than other methods. Training of the unsupervised methods rely on iterative procedures such as EM, and thus their computation times are dependent upon the stopping criteria, number of observations, and number of features provided, as well as the per-iteration computational complexity. Our empirical results suggest that the unsupervised methods require computation time to train which is practically feasible (hours or minutes of computation for an hour's worth of data). Importantly, once trained, both supervised and unsupervised models can be used in a computationally efficient way to support activity prediction in real-time operations.

To facilitate reproduction of results or testing of other algorithms, data files are available online[1] including the raw joint tracking information, derived features, and ground truth as first published in Rude et al. (2015). Data from a second Kinect, which was located at the worktable and rotated approximately 130° in the xz-plane, are also available for the same time periods. This second sensor has a coverage that often overlaps with the Kinect used in these models, but has different false detections. In total, the two sensors performed approximately 17 h of skeletal tracking. However, ground truth task coding is only available for the second half of the data.

This case study has shown that unsupervised methods can be employed instead of a supervised method with a drop in accuracy, but with significantly less effort required to label the data. As previously stated, the unsupervised methods we test in this case study are technically semisupervised, due to the use of a supervised initialization set, but can be easily implemented without any type of supervision. All of the unsupervised methods outperform the supervised HMM in estimating the number of individual tasks. In terms of average time per task, the unsupervised methods outperform the supervised method three out of five times. The supervised HMM is the worst HMM for predicting number of transitions, but does outperform the non-HMM models. When the accuracy of each half-hour test set is examined, the unsupervised methods tend to outperform the supervised HMM in the morning. We believe that this is due to training the model on data collected from Worker 2, but testing on data from Worker 1. The accuracy of the test sets in the afternoon favor the supervised HMM over the unsupervised methods. The supervised HMM is biased toward the worker the model was trained on, while the unsupervised methods are more transferable to different workers. In application, transferable models would be preferable so that a different training data set would not need to be collected for each worker in a facility. Another explanation is the change in work flow as the day progresses. In the early morning, *load* can be skipped due to no products on the line. Unsupervised methods are more adaptable to changes in the work process, and adaptability is also a desirable trait in application.

The FSHMM jointly estimates model parameters and selects features, which means that this set of relevant features will not necessarily be transferable to other models or applications. An algorithm that performs feature selection offers the advantage of selecting a relevant set customized to the application. While this set of features performs well in this manufacturing cell, a different set of features could be selected in another cell.

By reducing the number of features, noise is removed from the model; this renders the predictions more stable, which is illustrated by the number of individual tasks. Models using the full feature set (all but FSHMM and VB HMM) have a higher number of individual tasks. This indicates that the model is switching between task predictions more frequently. In application, the reduced number of features reduces calculation time, which is necessary when performing online predictions. The models performing feature selection also produce the two lowest RMSEs for average time per task and number of transitions (the K-means algorithm produces the same RMSE as VB HMM for average time per task).

The features removed by FSHMM and VB HMM tend to be the derived features. The FSHMM only includes two derived features in the reduced model, Dist and Left Hand Size, while the VB HMM only includes Right Elbow Over Shoulder. In application, these features would not need to be generated once the system has been trained, and would reduce calculation time for online predictions. Excluding the supervised HMM, the models performing feature selection tend to outperform the models using the full feature set in both accuracy and average time per task. This is due to the feature selection process removal the of noisy features that confuse task prediction.

Another intriguing trend can be seen in the features selected by FSHMM. Eight of the selected features are from the left side of the body, while only two are specifically from the right side of the body (leaving six non-sided features). This is interesting because the Kinect has a tendency to assume that the person is facing the sensor. Therefore, when there are partial occlusions or self-occlusions, or the user is facing away from the sensor, the most stable joint data will still be labeled as "left". This ability to consistently select the more stable and informative joints is strong, albeit qualitative, evidence that FSHMM is performing as designed.

[1] http://people.virginia.edu/~djr7m/incom2015/.

While this case study shows great promise for applying tracking technology to the manufacturing environment, and includes many hundreds of independent repetitions of the work tasks, it would be best to collect data from additional workers and work environments. Having data from additional conditions would help improve the generalizability of the models and, more importantly, identify new types of work that can be detected by this combination of sensor and model.

Even for the existing data, it may prove fruitful to push the models further and attempt to detect more subtle and short (elapsed time) tasks, which would yield richer data and improve overall model robustness. Also, the models could be refactored to provide continuous, or online, calculations so that model outputs could be output continuously rather than in batches.

Conclusion

The Microsoft Kinect is a low-cost, commonly available sensor that was used without modifications. Furthermore, unlike other studies using this sensor, we collected live data from an operational manufacturing cell without any guided or scripted work. Data collection and ground truth show the sensor to be robust for indoor conditions, even with industrial machinery, highly variable workers, and their non-work-related movements. HMMs in general, and the FSHMM in particular, are shown to be imperfect but effective unsupervised learning methods. This case study focused on a real-world data collection with all of the inherent variability and challenges, yet our methods still produced analytics that will likely be useful to process engineers. The estimation models point-by-point accuracy approaches 70 %, with FSHMM reaching 65.8 %. Ultimately, we have demonstrated the feasibility of applying commodity sensors and generalized models to a real-world setting and obtaining analytical results which can be used directly by plant managers.

Acknowledgements We would like to thank Aerojet Rocketdyne for allowing us to collect data at one of their facilities. We also thank reviewers of an earlier manuscript for their valuable feedback. This research was supported in part by both SAIC and the Commonwealth Center for Advanced Manufacturing.

References

Adams, S. (2015). *Simultaneous feature selection and parameter estimation for hidden Markov models*. Dissertation, University of Virginia.

Chen, T., Wang, Y.-C., & Lin, Z. (2015). Predictive distant operation and virtual control of computer numerical control machines. *Journal of Intelligent Manufacturing*, 1–17. doi:10.1007/s10845-014-1029-x.

Choo, B., Landau, M., DeVore, M., & Beling, P. A. (2014). Statistical analysis-based error models for the Microsoft Kinect depth sensor. *Sensors*, *14*(9), 17430–17450. doi:10.3390/s140917430. http://www.mdpi.com/1424-8220/14/9/17430.

Cohen, I., Cozman, F. G., & Bronstein, A. (2002). The effect of unlabeled data on generative classifiers, with application to model selection. In Proceedings of AAAI (submitted). http://www.hpl.hp.com/techreports/2002/HPL-2002-140.pdf.

Deitz, R., & Orr, J. (2006). A leaner, more skilled U.S. Manufacturing Workforce. *Current Issues in Economics and Finance*, *12*(2), 1–7. http://www.newyorkfed.org/research/current_issues/ci12-2.html.

Freedman, B., Shpunt, A., Machline, M., & Arieli, Y. (2010). Depth mapping using projected patterns. Publication number: US 2010/0118123 A1 U.S. Classification: 348/46.

Gilbreth, F. B., & Gilbreth, L. M. (1916). *Fatigue study: The elimination of humanity's greatest unnecessary waste*. Whitefish: Kessinger Publishing.

Grice, A., Peer, J., & Morris, G. (2011). Today's aging workforce—Who will fill their shoes?. In *Protective relay engineers, 2011 64th annual conference for* (pp. 483–491). doi:10.1109/CPRE.2011.6035641.

Huikari, V., Koskimaki, H., Siirtola, P., & Roning, J. (2010). User-independent activity recognition for industrial assembly lines-feature vs. instance selection. In *Pervasive computing and applications (ICPCA), 2010 5th international conference on* (pp. 307–312). IEEE.

IFR International Federation of Robotics: Robots Create Jobs- IFR International Federation of Robotics (2013). http://www.ifr.org/robots-create-jobs/.

Knight, W. (2012). This robot could transform manufacturing. http://www.technologyreview.com/news/429248/this-robot-could-transform-manufacturing/.

Koskimaki, H., Huikari, V., Siirtola, P., Laurinen, P., & Roning, J. (2009). Activity recognition using a wrist-worn inertial measurement unit: A case study for industrial assembly lines. In *Control and automation, 2009. MED'09. 17th Mediterranean conference on* (pp. 401–405). IEEE.

Krause, A., Siewiorek, D. P., Smailagic, A., & Farringdon, J. (2003). Unsupervised, dynamic identification of physiological and activity context in wearable computing. In *2012 16th international symposium on wearable computers* (pp. 88–88). IEEE Computer Society.

Landau, M., Choo, B., & Beling, P. A. (2015). Simulating kinect infrared and depth images. *IEEE Transactions on Cybernetics*, 14. doi:10.1109/TCYB.2015.2494877.

Mahdaviani, M., & Choudhury, T. (2008). Fast and scalable training of semi-supervised crfs with application to activity recognition. In *Advances in Neural Information Processing Systems* (pp. 977–984).

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. Cambridge: The MIT Press.

Niebles, J. C., Wang, H., & Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial–temporal words. *International Journal of Computer Vision*, *79*(3), 299–318.

Nigam, K., Mccallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*(2–3), 103–134. doi:10.1023/A:1007692713085. http://link.springer.com/article/10.1023/A%3A1007692713085.

Oreifej, O., & Liu, Z. (2013). Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Computer vision and pattern recognition (CVPR), 2013 IEEE conference on* (pp. 716–723). IEEE.

Packer, B., Saenko, K., & Koller, D. (2012). A combined pose, object, and feature model for action understanding. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1378 –1385). doi:10.1109/CVPR.2012.6247824.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286. doi:10.1109/5.18626.

Rokach, L., & Maimon, O. (2006). Data mining for improving the quality of manufacturing: A feature set decomposition approach. *Journal of Intelligent Manufacturing*, *17*(3), 285–299.

Rokach, L., Romano, R., & Maimon, O. (2008). Mining manufacturing databases to discover the effect of operation sequence on the product quality. *Journal of Intelligent manufacturing*, *19*(3), 313–325.

Rude, D. J., Adams, S., & Beling, P. A. (2015). A benchmark dataset for depth sensor based activity recognition in a manufacturing process. *IFAC-PapersOnLine*, *48*(3), 668–674. doi:10.1016/j.ifacol.2015.06.159. http://www.sciencedirect.com/science/article/pii/S2405896315003985.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., et al. (2011). Real-time human pose recognition in parts from single depth images. In *2011 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1297 –1304). doi:10.1109/CVPR.2011.5995316.

Stiefmeier, T., Ogris, G., Junker, H., Lukowicz, P., & Troster, G. (2006). Combining motion sensors and ultrasonic hands tracking for continuous activity recognition in a maintenance scenario. In *Wearable computers, 2006 10th IEEE international symposium on* (pp. 97–104). IEEE.

Stikic, M., Van Laerhoven, K., & Schiele, B. (2008). Exploring semi-supervised and active learning for activity recognition. In *Wearable computers, 2008. ISWC 2008. 12th IEEE international symposium on* (pp. 81–88). IEEE.

Sung, J., Ponce, C., Selman, B., & Saxena, A. (2011). Human activity detection from RGBD Images. In *Workshops at the twenty-fifth AAAI conference on artificial intelligence* (pp. 47–55).

Taylor, F. W. (1913). *The principles of scientific management*. New York: Harper.

Wang, J., Liu, Z., Wu, Y., & Yuan, J. (2012). Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1290–1297). doi:10.1109/CVPR.2012.6247813.

Wang, X., Ma, X., & Grimson, W. E. L. (2009). Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(3), 539–555.

Ward, J. A., Lukowicz, P., Troster, G., & Starner, T. E. (2006). Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(10), 1553–1567.

Xia, L., Chen, C. C., & Aggarwal, J. (2012). View invariant human action recognition using histograms of 3d joints. In *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on* (pp. 20–27). IEEE.

Xu, Y., & Ge, M. (2004). Hidden Markov model-based process monitoring system. *Journal of Intelligent Manufacturing*, *15*(3), 337–350.

Yang, X., & Tian, Y. (2012). Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on* (pp. 14–19). IEEE.

Zhang, H., & Parker, L. E. (2011). 4-Dimensional local spatio-temporal features for human activity recognition. In *Intelligent robots and systems (IROS), 2011 IEEE/RSJ international conference on* (pp. 2044–2049). IEEE.

Zhu, H., He, Z., & Leung, H. (2012). Simultaneous feature and model selection for continuous hidden Markov models. *IEEE Signal Processing Letters*, *19*(5), 279–282.