Machine Vision and Applications
DOI 10.1007/s00138-013-0528-7

ORIGINAL PAPER

On hierarchical modelling of motion for workflow analysis from overhead view

Banafshe Arbab-Zavar · John N. Carter · Mark S. Nixon

Received: 3 September 2012 / Revised: 1 June 2013 / Accepted: 10 June 2013

Abstract Understanding human behaviour is a high level perceptual problem, one which is often dominated by the contextual knowledge of the environment, and where concerns such as occlusion, scene clutter and high within-class variations are commonplace. Nonetheless, such understanding is highly desirable for automated visual surveillance. We consider this problem in a context of a workflow analysis within an industrial environment. The hierarchical nature of the workflow is exploited to split the problem into 'activity' and 'task' recognition. In this, sequences of low level activities are examined for instances of a task while the remainder are labelled as background. An initial prediction of activity is obtained using shape and motion based features of the moving blob of interest. A sequence of these activities is further adjusted by a probabilistic analysis of transitions between activities using hidden Markov models (HMMs). In task detection, HMMs are arranged to handle the activities within each task. Two separate HMMs for task and background compete for an incoming sequence of activities. Imagery derived from a camera mounted overhead the target scene has been chosen over the more conventional oblique views (from the side) as this view does not suffer from as much occlusion, and it poses a manageable detection and tracking problem while still retaining powerful cues as to the workflow patterns. We evaluate our approach both in activity and task detection on a challenging dataset of surveillance of human operators in a car manufacturing plant. The experimental results show that our hierarchical approach can automatically segment the timeline and spatially localize a series of predefined tasks that are performed to complete a workflow.

Keywords Workflow analysis · Human behaviour recognition · Hierarchical motion modelling · Video feature extraction · Probabilistic sequence analysis

B. Arbab-Zavar (✉) · J. N. Carter · M. S. Nixon
School of Electronics and Computer Science,
University of Southampton, Southampton, UK
e-mail: baz10v@ecs.soton.ac.uk

J. N. Carter
e-mail: jnc@ecs.soton.ac.uk

M. S. Nixon
e-mail: msn@ecs.soton.ac.uk

1 Introduction

Recently, there has been an increased interest in the area of human activity and behaviour recognition. While many different subsets of activities are investigated, the conditions and overall specifications of the problem vary largely. One of the applications of this is in visual surveillance. The goal here is to detect a set of pre-defined tasks which occur within a workflow of a complex industrial environment.

1.1 Human motion types

Bobick [4] defined different human motion types and an understanding of these motions. In this, Bobick proposed a three-level motion-understanding scheme based on the order of the implied knowledge required to understand the motion. In increasing order of knowledge, these motion levels are: 'movement', 'activity', and 'action'. The majority of work in the area of human motion analysis is focused on the recognition of human 'activities'. There we have 'activities' such as jumping, boxing, and waving. However, the study of what Bobick described as 'action', the most knowledge intensive form of motion understanding, in which contextual or causal relations play a critical role, appears in a smaller number of works and it is a less definitive problem. This high contex-

Published online: 07 August 2013

tual dependency also encourages diverse terminology, while the approaches and techniques to understand them do not lend themselves to direct comparison either. It is also worth noting that, while this may not always be the case, often a knowledge-based hierarchy of motion results in a hierarchical structure for the perceived motion itself; 'activities' are made up of 'movements' and 'actions' are made up of 'activities'.

1.2 Related work

Manufacturing [7,9,19,20] and office [3,10,17], environments, medical operating rooms [12], TV studios [13] and smart homes for the elderly [8] are amongst the settings wherein 'action' perception and understanding from video streams had been studied. Pinhanez and Bobick [13] were among the first to analyse the actions within a video stream. They used this information to build an 'intelligent studio' wherein the cameras were controlled automatically based on a predefined script and the visual data from the cameras themselves. Considering the sequential nature of the patterns, using a hidden Markov model is a justified and popular choice. Padoy et al. [12] studied the workflows in a simulated operating theater. They introduced the Workflow-HMM which is a form of two-layered hierarchical HMM. Nguyen et al. [10], who experimented in an office environment, also used hierarchical HMMs, however, movement trajectories are used there to learn and recognize actions. Behera et al. [3] have also exploited qualitative spatial information to inform action recognition. In this, all possible pair-wise spatial relations among objects are represented in a relational state space and a probabilistic Latent Semantic Analysis is used to model workflow from this relational state space. Voulodimos et al. [20] and Oliver et al. [11] used HMMs as a means for low level sensory data or high level decision fusion. Criticizing HMMs as requiring well-defined states and robust features, Veres et al. [19] opt for a neural network: they used an echo state network to learn the patterns within an annotated global scene descriptor. Shi et al. [17] proposed the use of propagation networks to handle parallel tracks within one action. Kosmopoulos et al. [7] incorporated user's feedback into the learning process using a neural networks based method to dynamically correct erroneous classifications. While on a slightly different note, Nater et al. [9] have proposed a novel approach for unsupervised discovery of tasks within a workflow. In this, assuming a temporal consistency and cyclic repeated patterns, they have used Slow Feature Analysis method to learn and extract invariant components in the temporal signal.

In these works, the modelling of workflows and the recognition of actions are addressed as well as temporal and spatial segmentation of the sequences. However, the recognition of actions is often addressed separately from the problem of sequence segmentation. Temporal segmentation, as in detecting the start and the end of a sequence which makes up an action, is also the focus of a related line of work which is devoted to unusual activity or unusual scene detection with the aim of detecting sequences that have not been previously observed [1,5,8]. In many approaches the sequence to be classified starts with a person entering the field of view and it ends when they exit this field. Thereby every movement is interpreted toward a meaningful behaviour. While such examples largely simplify the problem, this is not normally the case in real scenarios. In this paper, the imagery derived from a camera monitoring a working cell in a car manufacturing plant is analysed. In this, the human operators may be involved in multiple consecutive actions. On the other hand they might perform no predefined actions at all. The other point of concern is the spatial segmentation of actions. The complex and cluttered environment which is analysed in this paper, poses an overwhelming detection and tracking problem as was experiences by [7,19,20]. They have thus opted for a global scene descriptor and abandoned the problem of spatial segmentation.

1.3 Overview of the method

In this paper, Bobick et al.'s hierarchy of human motion approach is adopted, and the problem is split into activity and task recognition. The tasks are the specific actions of this environment and our goal is to detect these tasks. These tasks, however, are composed of lower level activities which are not specific to this environment and can be observed in other environments and in other workflows. Therefore by describing our environment-specific tasks in terms of generic activities we move toward a more general solution which can be used in other applications. This approach also helps to divide the problem into smaller, more manageable and better defined problems which can be tuned, tested and improved separately. In terms of functionality, it enables us to determine what stage the task is in at each point in time which can be very useful when analysing complex or long duration tasks. The activities are classified using a set of motion and pose features in a small time window. A probabilistic model examines the sequences of activities for instances of a task while the remainder will be labelled as background. In this, two separate HMMs for task and background compete for a sequence. The sequence of activities is also subjected to a probabilistic analysis by activity HMMs which remove the irregularities in the activity sequence prior to searching for task instances. Briefly, the various levels of our hierarchical approach are: (i) detection and tracking; (ii) activity classification; (iii) boosted activity sequence (via activity HMMs); (iv) task extraction; and (v) task label assignment. Using a surfeit of informative cues and handling the uncertainty in the input from preceding layers of the hierarchical process, the

error which would otherwise accumulate through the stages of the hierarchical approach is reduced. We address both problems of temporal and spatial segmentation in identifying a series of predefined tasks in imagery monitoring a car manufacturing plant. We will show that the overhead view poses a more manageable tracking problem while providing a full visual access to the environment which is viable for a human motion analysis. We are faced with problems such as noise, occlusion and overlapping tasks, which are inevitable in a loosely constrained environment such as this. Also, there is a series of background actions which are not within our predefined tasks, these include: idle or random actions of the human operators; maintenance; cleaning; and replacement of empty racks.

The main contribution of this work is to use computer vision to enable the analysis of the patterns of workflow in complex industrial environments. Our hierarchical approach serves to successfully divide this complex problem into manageable parts. Although applied to a specific problem for detecting a set of specific tasks, we will discuss that the distinction between the concepts of activity and task helps to improve the reusability and the generality of this approach. We will outline the conditions which would need to be satisfied for this approach to be applicable to a new problem and provide guidelines for how to do so. In addition to what has been offered in [19,20] our approach enables us to (i) localize the tasks in image frame and (ii) offer a stage by stage commentary-like report of the task as it happens.

The dataset and the taxonomy of the tasks and activities are described in Sect. 2. In Sect. 3, the analysis of workflow patterns including the task and activity recognition are described in detail. The experimental results on activity classification and task modelling, segmentation and recognition are presented in Sect. 4. Finally, overall conclusions are reviewed and future works are discussed.

2 Workflow patterns and the dataset

The work here has been developed within a European Union project, SCOVIS (Self-Configurable Cognitive Video Supervision). The environment we aim to monitor is a working cell within a car manufacturing plant which contains a welding cell and various racks of parts. In this, parts are individually handled and carried to the welding cell by human operators. They are then welded together, assembling a frame which is then moved to the next stage of the manufacturing process. The same process is thereafter repeated in the work cell so a repeating pattern of activity can be observed. In this context, a series of tasks needs to be performed before the frame is ready to be transported to the next stage. Six tasks are identified within this environment:

Task 1: A part from Rack 1 (upper compartment) is placed on the welding cell by a worker(s).
Task 2: A part from Rack 2 is placed on the welding cell by worker(s).
Task 3: A part from Rack 3 is placed on the welding cell by worker(s).
Task 4: Two parts from Rack 4 are placed on the welding cell by worker(s).
Task 5: A part from Rack 1 (lower compartment) is placed on the welding cell by worker(s).
Task 6: A part from Rack 5 is placed on the welding cell by worker(s).

An additional task, 'welding', also exists in this environment. This task has significantly different definition and is not considered in this work. Figure 1a shows the position of the stationary components (the welding cell and the racks), and Fig. 1b shows tasks 3 and 4 being performed.

These six tasks can all be described generally as picking up a part from a rack and placing it on the welding cell. However, these tasks are different in terms of their visual appearance—the parts are of different shapes and sizes, they are handled in different manners by one or two human operators, while the position of the racks offers a vital cue. The sequential order of the required tasks is flexible, and they may also be performed concurrently. Four activity categories: walking, carrying, handling and standing are analysed. These are taken together from the basic building blocks of the higher level task analysis.

We use images derived from an overhead view to recognize the tasks in the workflow described above. The dataset includes synchronized video captured from four oblique views of the same cell. Veres et al. [19] and Voulodimos et al. [20] have analysed the workflow patterns from these oblique views. The overhead view analysis of the workspace might not be the optimal view angle to observe individual activities which include motion perpendicular to this viewing plane, also an estimate of height is not acquirable. However, it offers a much lesser occluded scene and a manageable detection and tracking problem. In the oblique views, Veres et al. [19] have reported a mere recall of 24 % with a precision of 9 % in detection using two different state-of-the-art person detection and tracking methods. The difficulty of the tracking problem in the oblique views compelled Veres et al. [19] and Voulodimos et al. [20] to opt for a global scene description and thus abandon the problem of spatially localizing a task. The other benefits of using the overhead view include: a potentially larger field of view through the use of a panoramic camera; and better estimation of position on the ground plane. Figure 2 shows sample images derived from an oblique and the top view, concurrently. Three workers are partially occluded from an oblique view angle, while two of them are fully visible in the image derived from the overhead

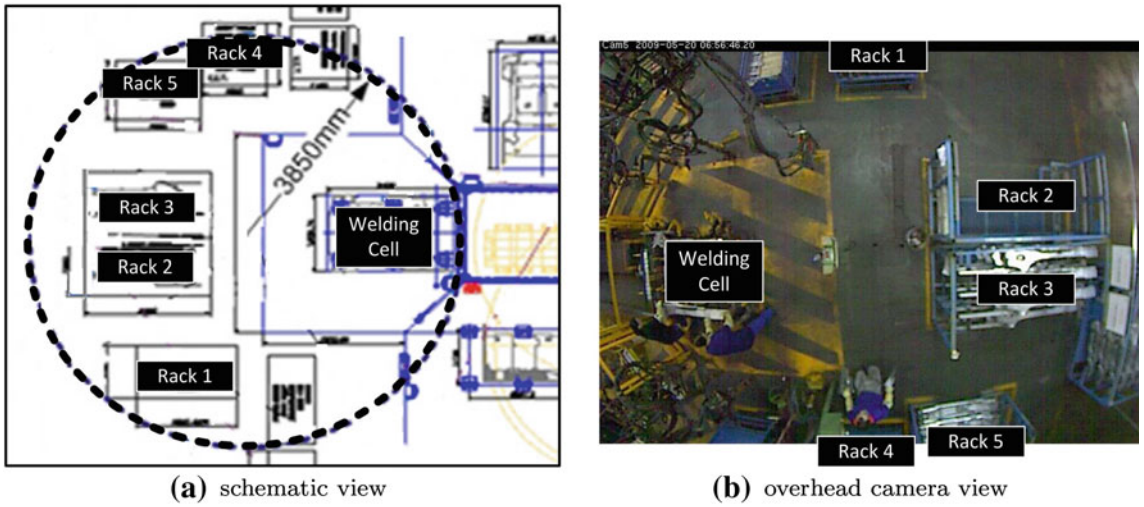


Fig. 1 **a** Schematic and overhead camera views of the racks and the welding cell. **b** Also shows tasks 3 and 4 being performed

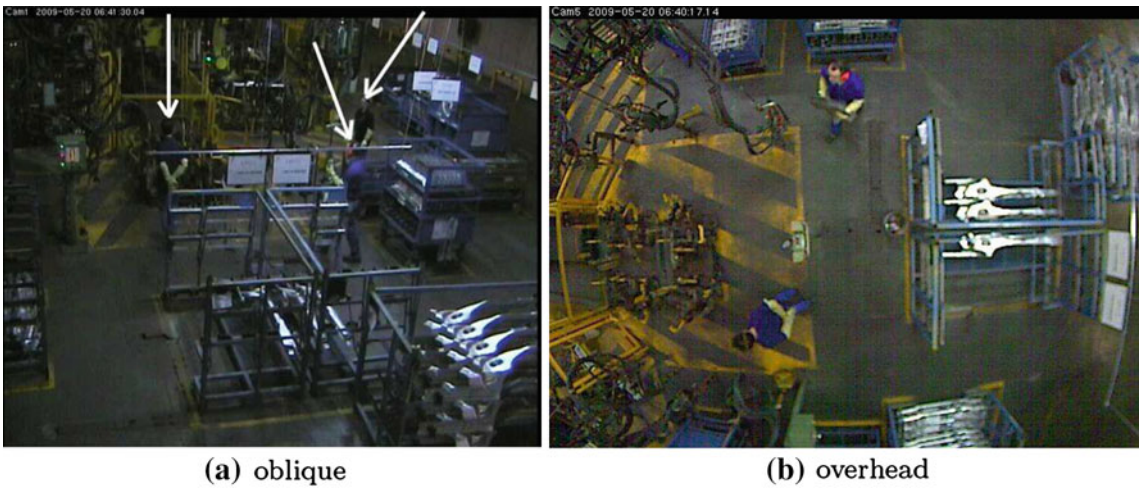


Fig. 2 Views of the manufacturing cell: the human workers partially occluded in the oblique view (**a**) are fully visible in the top view (**b**)

view, and the third worker is outside the active area of this working cell.

3 Workflow analysis

The aim of this work is to detect and identify a set of predefined tasks of a workflow. For this, we adopt the hierarchical structure of human motion by Bobick. The tasks are broken down into smaller entities: the activities and modelled using the workflow definitions inherent to our data. The set of activity categories are the distinguishable parts within a task sequence. Albeit, note that we have opted for a generic set of activities. Figure 3 demonstrates the overall approach. Tracking is performed using a simple blob tracker based on motion [2]. A set of shape and motion features are extracted from a 10-frame window for each moving blob, and their fluctuations are analysed using the Fourier transform. A simple KNN classifier then classifies the activity which is being performed based on a binary tree structure. Feature selection is performed at each node of the tree to select the most relevant features. This initial prediction of activity is further subjected to a probabilistic analysis of transitions between the activities in different activity zones using HMMs; the initial predictions of the activities are fed into a set of localized HMMs as observations while the hidden states represent the 'true' activities. This amended sequence of activities is analysed by probabilistic models for task and background, which compete for the incoming sequence.

Although each step of this process is a difficult problem and there is a tendency for errors to accumulate towards the higher levels of the hierarchical process, it is worth noting that there is often a surfeit of information available which could be used to reduce error. This includes: a surfeit of frames to classify activities—as noted by [16] we need approximately 5–7 frames to identify a simple activity; more than required

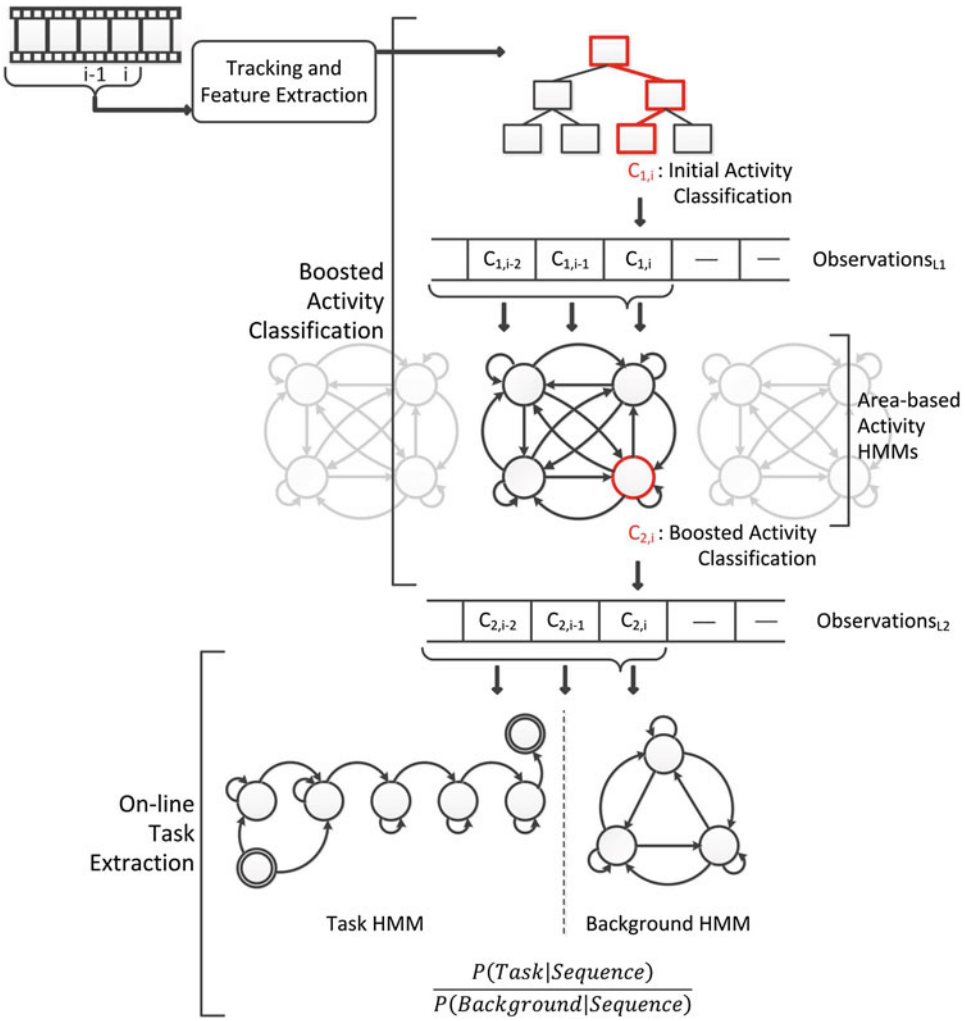


Fig. 3 Diagram of our approach

number of individual activities to recognize a task—often, only the key activities need to be observed to recognize a task; the use of location information or activity zones; and sequential constrains within activities, as exploited by the activity HMMs. The uncertainties are handled via different levels of HMM processing. However, clearly if a subject is not detected we cannot proceed with the classification of its activities and detect possible tasks. Thus we aim for a low false negative rate in detection and tracking. In the remainder of this section, starting with the activity classification we work through the layers of the hierarchical process depicted in Fig. 3.

3.1 Activity classification

A simple blob tracker is used to detect the main moving parts i.e. human beings. This is described in our earlier work [2]. Various shape-based and motion-based features are then extracted for the activity classification. Four activity categories: walking; carrying; handling and stand are considered. A binary decision tree which uses the features selected via the ASFFS (Adaptive Sequential Forward Floating Search) [18] provides an initial prediction for the activity. Exploiting our continuous video data, we can then analyse the plausibility of a predicted sequence of activities using HMMs.

3.1.1 Feature-based classification

Both the motion and the shape of the moving blob contain cues as to the activity which is being performed. We build a composite feature set by combining a set of shape-based and motion-based features. These features, which are extracted for each detected blob, are:

- motion-based features: average speed; instantaneous speed; and changes in the direction of motion. We shall collectively refer to these features as dynamic features, denoted by F_{D}.
- shape-based features: Hu Invariant Moments [6], which are seven moments providing a global description of the

shape. These are translation, scale and rotation invariant. Hu moments are computed using the normalised central moments, η_{pq}, which are in turn computed from centralised moments, μ_{pq}:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{(\frac{p+q}{2}+1)}} \quad \forall p+q \geq 2 \tag{1}$$

where

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q I(x,y) \tag{2}$$

and $I(x, y)$ is the image intensity at location (x, y). The first Hu moment, $M1$ is defined as:

$$M1 = \eta_{20} + \eta_{02}. \tag{3}$$

For all seven Hu invariant moments see [6]. Let F_{M} denote the collection of these moments:

$$F_{\mathrm{M}} = \{M1, \ldots, M7\}. \tag{4}$$

Some heuristic shape related properties are also considered. These are the area, diameter and the pixel density of the detected blob. In addition, considering that the parts which are being handled are metallic and they often have a strong projection in saturation and value axes of the HSV colour space, the sum of values in these two axes are also considered individually. F_{H} denotes the collection of these heuristic features.

A 10-frame window is considered for activity classification. For this, the 7-frame window suggested by [16] for classifying basic activities is used as a guideline. Clearly, for calculating the motion related features at least one previous frame is required. For estimating the features related to the change in the direction of motion two previous frames are needed, and thus these features can only be computed from the third frame onwards of the window. Thus a larger window of 10 frames is used here.

Apart from the average speed, which has a single value representing the average speed of the moving blob during these 10 frames, all the features listed above will have one value per frame. The mean value for each feature and a measure of its changes within these 10 frames are considered. Let ϕ be the set of all the shape and motion-based features;

$$\phi = \{F_{\mathrm{D}}, F_{\mathrm{M}}, F_{\mathrm{H}}\}. \tag{5}$$

Let f_i be a feature where $f_i \in \phi - \{\bar{v}\}$ and $\bar{v}$ is the average speed. Let the series of f_i values for blob b in a 10-frame window centered at time t be $W_i(b, t)$:

$$W_i(b,t) = [f_i(b,t-4), \ldots, f_i(b,t+5)]. \tag{6}$$

A frequency-based analysis of $W_i(b, t)$ provides insights as to the changes in the feature value in this window. Let $\mathcal{F}$ denote discrete Fourier transform.

$$X_i(n) = \mathcal{F}(W_i(b,t)), \; n = 0, \ldots, 9 \tag{7}$$

$$A_i(n) = |X_i(n)| \;\; \varphi_i(n) = arg(X_i(n)) \tag{8}$$

where A_i and φ_i are the magnitude and phase in different frequencies. Thereby the feature vector V is generated for each sample as:

$$V = \{A_i(n),\ \varphi_i(n),\ \mu_i,\ \sigma_i,\ \bar{v}\}$$
$$i = 1, \ldots, |\phi| - 1,\ n = 0, \ldots, 5 \tag{9}$$

where μ and σ denote the mean and the standard deviation of the feature values. The first six frequency components are considered since these are the most significant of the frequency responses. The mirror-symmetrical (negative frequency) components are not considered as they add no extra information. A binary decision tree approach has been adopted for the classification of activities. This splits the classes into still: (handling and standing); and moving: (walking and carrying). Due to the composite nature of our feature vector and that various feature types are susceptible to different levels of corruption in noise, a feature subset selection method, the ASFFS, is employed at each node of the tree to derive the discriminative cues whilst removing corrupt, irrelevant or redundant features. A KNN classifier is used to obtain a classification.

3.1.2 Activity models for sequential plausibility

The logical and structural patterns within a sequence of activities can be exploited to evaluate the plausibility of a sequence of predictions. Hidden Markov models (HMMs) are used to learn these patterns. HMMs are probabilistic finite-state machines which model distributions over sets of possible sequences. The activity HMM is shown in Fig. 4. In this, the hidden states are the activities: walking; carrying; handling and standing, and the observations are the predictions

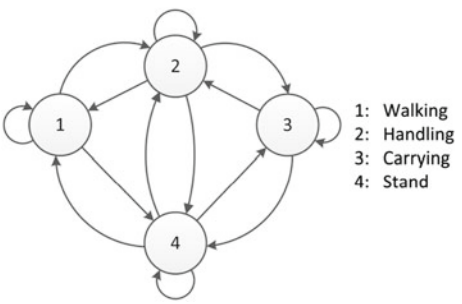


Fig. 4 Activity HMM

obtained by the binary tree classifier. Let a set of predictions, A, by the decision tree be: $A = \{a_t\}, t = 1..T$, and let H be the set of hidden states for the HMM. Given the set of predictions A, the probability of being in state α at time t is:

$$P(S_t = \alpha|A) = P(A_t = a_t|S_t = \alpha)\mathcal{P}_{t,\alpha,A} \tag{10}$$

where

$$\mathcal{P}_{t,\alpha,A} \equiv \max_{\beta \in H}[P(S_t = \alpha|S_{t-1} = \beta)P(S_{t-1} = \beta|A)] \tag{11}$$

The activity HMMs are trained using the Baum–Welch algorithm and Viterbi algorithm is used to predict the most likely sequence of activities.

Our data imposes that there is a spatial dependency regarding the expectation of various activities. Three distinct activity zones are identified and shown in Fig. 7. Thereby for each activity zone a separate activity HMM is learnt. Further details on the activity classification as well as the initial experimental results can be found in our earlier work [2]. There, it has been shown that the activity HMMs improve the performance significantly. Noting that in a sequence of activities a transition to a different activity is a rare occurrence, instead, the same activity tends to be observed in adjacent frames, and considering that the activity HMMs model this structure, one of the immediately visible effects of employing the activity HMMs is stabilizing or smoothing the sequence of predictions. However, it has been shown in [2] that the gained improvement is much more than a simple stabilizing effect. Further results are discussed in the next section.

3.2 Probabilistic task extraction

HMMs are used to identify a specific sequence of activities which is referred to as task, within a bigger, less severely constrained sequence, which we refer to as background. For this purpose, a task can be simply described as picking up a part from a rack and placing it on the welding cell. Two HMMs, one to model a task sequence; and one for background activity, are used to make a decision as to the occurrence of a task by comparing the posterior densities:

$$\begin{aligned} R(\text{Seq}_i) &= \frac{\Pr(\text{Task} \mid \text{Seq}_i)}{\Pr(\text{Bg} \mid \text{Seq}_i)} \\ &= \frac{\Pr(\text{Seq}_i \mid \text{Task})\Pr(\text{Task})}{\Pr(\text{Seq}_i \mid \text{Bg})\Pr(\text{Bg})} \\ &\approx \frac{\Pr(\text{Seq}_i \mid \text{HMM}_{\text{task}})\Pr(\text{Task})}{\Pr(\text{Seq}_i \mid \text{HMM}_{\text{bg}})\Pr(\text{Bg})}. \end{aligned} \tag{12}$$

Equation (12) presumes that the sequences, Seq_i, are segmented, where each sequence is either a task or background. However, the input sequence of activities is not temporally segmented. If the ratio, R, is calculated for the entire sequence up to the current point in time, the background HMM will almost always win against the task HMM, since between the two models the background HMM is the only one which can describe the whole data, even the task sections, albeit with a small probability, and in that sense it is inclusive of the task sequences. The background HMM is purposely general so that it can handle the other different actions, other than the tasks, which can happen in this environment. As a result and although it has not been trained for it, the background HMM can also accept the task sequences with some small probability. Also, unlike the task HMM it does not have an absorption state, and therefore it does not terminate. Due to this inclusive and continuous nature of the background model, a background sequence can, at any time, be interrupted if a task model obtains the starting conditions, as denoted by:

$$R(\text{Seq}_0) > T, \tag{13}$$

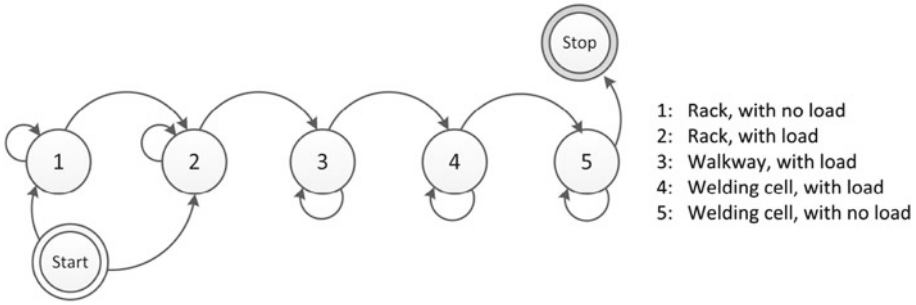
where $\text{Seq}_0 = \text{Act}_t$ is the sequence that only contains the current observed activity at time t, Act_t. Once a possible task sequence is started it is terminated only if the entire sequence fits the background model better, taking into account the sensitivity of the detection process;

$$R(\text{Seq}_i) < T. \tag{14}$$

A special case of Markov models sometimes referred to as left-to-right models are used for identifying a task sequence. In these, a sequence progresses through the states of the model from left to right and it will be absorbed in the final states. A task is detected if a sequence arrives at the absorbing or final state of the task HMM. The task HMM and background HMM are trained using the Baum–Welch reestimation method. Since the task HMM is a special case of HMM (left-to-right model) for which training cannot be performed using a single observation sequence, modified reestimation formulas for multiple observation sequences are given in [14]. In the test phase Viterbi algorithm is used to predict the probability of a sequence being generated by the task and background models.

As mentioned before, a task can be generally described as a sequence of activities which corresponds to picking up a part from one of the racks and placing it on the welding cell. Since the initiation and completion of a task depends directly on the location of the detected activities as well as the activity itself, the observation at each state of the HMM is determined based on both the activity and the location. Four activity categories have been considered in Sect. 3.1, while three activity zones are also distinguishable. These four activity categories are: walking; carrying; handling; and standing, and the three activity zones are: racks; welding cell; and walk ways. These three zones are highlighted in Fig. 7. Thereby, the alphabet of the observed symbols in the task and background HMMs includes: 12 area-based activities; a start symbol; and a null symbol. Prior to learning, task HMM is set up as shown in Fig. 5 by adjusting the observation and transition probabil-

Fig. 5 Task HMM



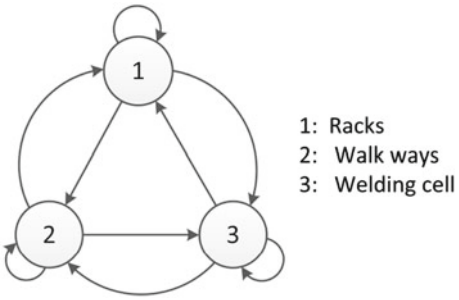


Fig. 6 Background HMM

ities. For example, intuitively the task HMM starts when a person is detected in a rack zone either handling or carrying a load. The observation probabilities are further adjusted to account for the rate of correct activity classification in each zone. The sequence then progresses from left to right with no backward steps or hopping. A small noise value has been introduced into the transition probabilities to prevent instabilities at the learning stage. Without which, the process will be too sensitive to erroneous activity classifications. Also, note that the probability of observing a sequence decreases exponentially with duration. Thus, downsampling is advisable. Downsampling is achieved by resampling with a lower frequency from a smoothed sequence of area-based activities which also serves to reduce the noise level. Each observation in the smoothed sequence is derived by taking the maximum vote in a neighbourhood around the corresponding point in the initial area-based activity sequence.

The background actions are represented by the background HMM shown in Fig. 6. The background HMM is a fully connected HMM with three states corresponding to the three activity zones. It provides a general model for the motion patterns of the human workers between these activity zones. The initialization of the background HMM, prior to training, allows for equal transition and observation probabilities. The sections of the training data which do not include a task sequence are used when training the background HMM.

Once an occurrence of a task sequence has been distinguished from the background, the trajectory of the segmented task sequence will be compared against the task map (see Fig. 7). The task map is the accumulated trajectories for each task over different occurrences in the training set. The distance between the trajectory of the segmented sequence and a specific task trajectory is the mean of the closest point distances of all the points in the detected sequence.

3.3 A note on generality

It is easy to imagine that the description, appearance, components and peculiarities of tasks would be different from one application to another. However, the distinction between the concepts of activity and task can help to improve the reusability of this approach. For example, while task 1 in our application is a very specific task which will only be observed in this scenario, an activity such as walking is very generic and will appear in many other applications and can be detected and identified with the same method which was described here. To arrive at detecting tasks from activities, the overall methodology as was shown in Fig. 3 can be used, while only the task HMMs need to be redesigned from scratch. Another advantage of this approach is in providing better situation awareness, by breaking down the tasks into activities as opposed to using the raw video input for task detection. Thereby we can provide a real-time commentary-like report of the stages of tasks as they happen. We can also provide a probabilistic identification for different tasks on partial data which can be very useful if we are dealing with long duration tasks. Also in terms of algorithm design and parameter tuning, this approach provides a more intuitive and self-contained parts which can be tuned, tested and improved separately. In order to be able to use this approach the problem should satisfy these conditions:

(a) The overall objective should be to detect a set of well-defined repetitive tasks.
(b) The tasks should be performed by physical activities of humans.
(c) The image resolution should allow for the perception of the activities which make up the tasks.
(d) A stable detection and tracking should be obtainable.

We have encountered an example of a task for which our approach did not provide a suitable solution. This task is

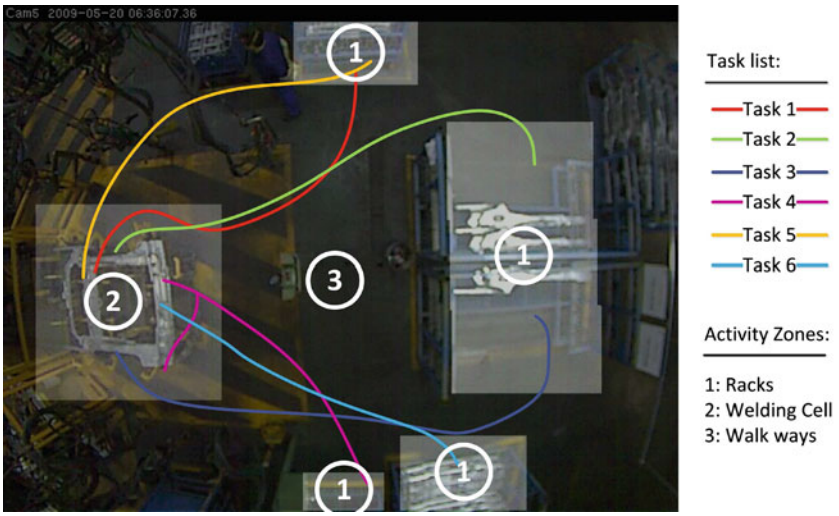


Fig. 7 Task map. Note that the task trajectories shown in this figure are the averaged trajectories and they are also smoothed for better clarity

welding. Once all the parts are carried and placed on the welding cell one to three workers will weld the pieces together. While welding, for the most part the human workers appear as standing motionless while they are holding the welding device. Welding also poses a challenging problem for our motion-based detection and tracking, where flares which are caused by welding fly about randomly. We shall demonstrate later in Sect. 4.4 that other approaches can be used to handle these cases.

Our approach can be suitable for other factory and manufacturing environments where certain well defined tasks are expected. To apply this approach, the system designer would first need to understand the tasks. The definitions of the tasks require knowledge of the environment and are to be outlined by the expert. The system designer will then need to: (i) identify the lower level activities which make up these tasks and identify the key areas of the visual field of view (if applicable); (ii) design the task HMMs; (iii) label ground truth samples for tasks and activities. Thereafter the feature extraction and feature selection for activity recognition parts can be re-used and activity HMMs and background HMM can be re-used with small adjustments. The overall hierarchical procedure would be the same as shown in Fig. 3.

4 Experimental results

4.1 Dataset

The dataset contains 410,000 frames captured at 24 fps. The overall duration of the dataset is approximately 4 h and 45 min. This data includes 20 workflows each containing the 6 tasks. The first seven workflows which appear in the first 120,000 frames are used in training of activity classification and modelling as well as in task and background modelling. The remainder of the data is used in testing. This data has a series of dropped frames at irregular frequencies and with arbitrary durations. These frame drops can be considered as sparse, high level, localized noise. Though the volume of data ensures a majority of uncorrupted samples dropped frames can be problematic in task modelling. By definition, the task HMM requires observation of a full task sequence for detection. A partial task sequence would either fail to obtain the starting conditions [as denoted by Eq. (13)] or be incomplete and fail to arrive at the absorption state of the task HMM, both leading the task being undetected. Dropped frames are not a common feature of these systems and only appeared in this dataset due to a technical bug. Therefore, in our experiments we remove the partial sequences rather than attempt to handle them.

There are three main situations which are caused by dropped frames and cannot be handled by the task detection, these are: (i) interruption in the task sequence due to tracking failure; (ii) missing start of; or (iii) missing end of a task sequence. We therefore define a new test set by removing any occurrence of these three cases from the initial test set. We will refer to this set as the loosely pruned (LP) test set. The other cases of dropped frames can also cause undesirable effects such as reduced accuracy in tracking and erroneous activity predictions which in turn may cause errors in task detection. Further to simulate clean data, any task sequence with more than three consecutive frame drops has been removed from the test set. As such a new test set was generated which we will refer to as the aggressively pruned (AP) test set. The LP and AP test sets are the results of two different noise reduction procedures. Table 1 shows the number of task sequences before and after the noise reduction steps that generate LP and AP test sets. The noise reduction procedure for the training set is the same as the one in AP test set. This table also shows the total number of task samples

Table 1 Training and test sets

	Frame span	Task samples	Background samples	Task sequences
Training set	120,000	3,406	75,748	16/42
AP Test set	290,000	5,410	159,793	25/78
LP Test set	290,000	7,156	159,793	35/78

and background samples which are the overall frames-wise detections for task and background. Please note that the training and the test sets were annotated manually per detected moving blob per frame with activity and task labels.

4.2 Activity classification

Figure 8 shows the confusion matrices for activity classification with and without feature selection, as well as the confusion matrix obtained by analysing the activity sequence by the activity HMMs. Correct activity classification rates of 58.38, 63.82 and 68.70 % were achieved using these three levels of activity classification on a test set consisting of 11,203 samples. Single frame examples of correctly classified activities can be seen in Fig. 9. Note that while the entire test set is manually annotated with task labels, the lower level annotations, i.e. the activities, are available only on this subset of the test set. Table 2 shows the number of samples and the true positive and false positive rates for each class obtained by using both feature selection and activity HMMs (corresponding to Fig. 8c). The true positive and false positive rates are also noted for the training set.

Intuitively, the single feature used to select between the still or moving activities is the average speed. To distinguish between the two moving activities, walking and carrying, the most significant feature, selected via ASFFS, is the mean area of the detected bounding box. This is reasonable since most of the carried parts are large causing an increase in the size of the bounding box. Other features within the first 10 most significant features selected with ASFFS are all shape-based, reflecting that walking and carrying involve similar motion dynamics. These shape features include some of the shape moments: (low frequency magnitude changes in M1 and mean value of M3) and some of the heuristic shape properties: (mean and high frequency changes in sum of pixel intensity changes; the density of pixel intensity changes; as well as mid-frequency magnitude changes in area). On the other hand, to distinguish between handling and standing the most important feature, selected with ASFFS, is the standard deviation of the instantaneous speed. As a person stretches and bends to lift or place a part (both classified as handling), it can appear as moving backwards and forwards, while its overall position is not changed. Thus the significance of the instantaneous speed is understandable. The mean instantaneous speed is also selected within the first 10 most significant features. The other dominating feature is a shape feature, and it is the density of pixel intensity changes which appears three times as mean value of density; low frequency magnitude changes in density and standard deviation of density.

Note that the least distinguishable activity is handling. Multiple causes have been identified for this observation: (i) there is an inherent uncertainty between handling and carrying, in that it is unclear when one ends and the other starts; (ii) handling occurs at the border between stationary and locomotive activities; (iii) the overhead view is incapable of perceiving a handling motion that occurs in a perpendicular direction to this viewing plane. This is mainly observed at the racks near the horizontal centre of the frame. Also note that although in task recognition we have removed samples from the training and test sets due to dropped frames, all the data is considered at this stage.

As mentioned before the detection is based on motion. For a detected person who stops motionless in the scene, a

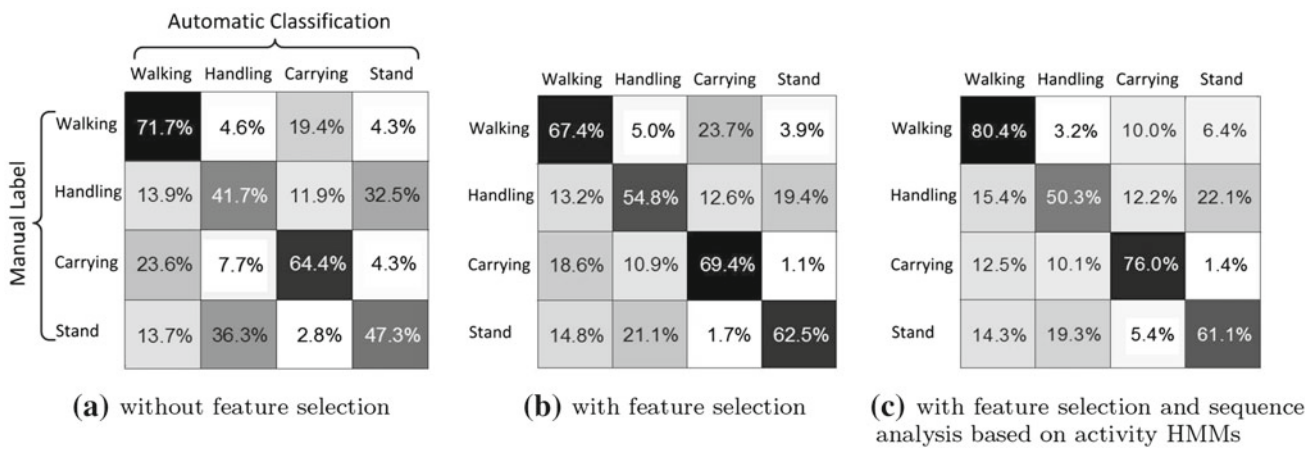


Fig. 8 Confusion matrix of activities in an 11,203 sampled test set for the three levels of activity classification

record of the last detection is kept alive for a certain time. During this time the activity is classified as stand. If the person starts moving within this time, tracking will be resumed and the activity sequence will be uninterrupted. However, if the person starts moving after a long period of stillness, a new detection with a new activity sequence is initiated. A threshold of 80 frames has been chosen as the maximum period for stillness. This threshold has been chosen empirically for this dataset. However, note that the choice of value for this threshold is not critical in our dataset since it is observed that once the workers start a task they do not pause for long periods of time until the task is finished. They may be motionless while waiting for their task to start. However, since these periods of stillness are not within a task, the potential interruptions of the activity sequences do not affect the task detection performance. Although the value for stillness threshold is not critical, it is beneficial to have this threshold since unlike the human workers who enter and exit the field of view, erroneous detections may seem to disappear from the middle of the frame and clearly it is not desirable to keep a record of these erroneous detections indefinitely.

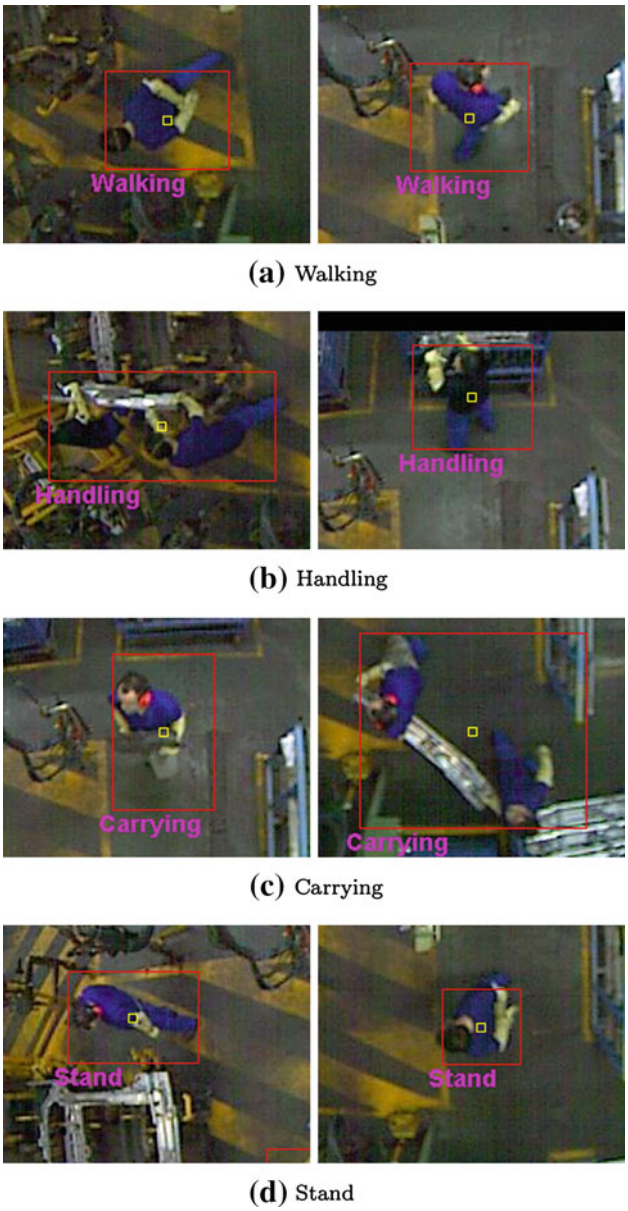


Fig. 9 Single frame examples of the four activity categories

4.3 Task detection

The training set described in Table 1 is used to train the task HMM. For this, the automatically tracked human operators are manually labelled when engaging in a specific task. The remaining detections in the first 120,000 frames of the data, which are not part of a task sequence, are used for training the background HMM. The initial state transition probabilities and the observation probabilities for the task and background HMMs loosely impose the structure shown in Figs. 5 and 6. In these, a high probability is assigned to remaining at the same state ($P_{\text{reoccur}} = 0.9$). The initial observation probabilities for the task HMM are adjusted using the activity confusion matrix in Fig. 8c. For example, at the state 1 of the task HMM we assume equal probabilities for observing activities walking and standing in a vicinity of a rack, while assuming a zero probability for any other observation symbol. However, walking and standing may be miss-classified with probabilities determined by the confusion matrix. Thereby an updated version of the initial observation probabilities is obtained. A small noise value has also been introduced to these probabilities to prevent instability and zero likelihoods at the learning stage. The task model stabilizes within about 40 iterations.

In evaluation, the task detection algorithm segments the continuous video input into localized sequences of task and

Table 2 Activity classifications performance using both feature selection and activity HMMs on the test and training sets

		Walking	Handling	Carrying	Stand
Test	True positive	3,610/4,491 ≈ 80.38 %	1,207/2,402 ≈ 50.25 %	1,258/1,655 ≈ 76.01 %	1,621/2,655 ≈ 61.05 %
	False positive	957/6,712 ≈ 14.26 %	823/8,801 ≈ 9.35 %	886/9,548 ≈ 9.28 %	841/8,548 ≈ 9.84 %
Training	True positive	10,454/11,924 ≈ 87.67 %	2,010/4,177 ≈ 48.12 %	2,058/2,463 ≈ 83.56 %	4,880/6,245 ≈ 78.14 %
	False positive	1,687/12,885 ≈ 13.09 %	884/20,632 ≈ 4.28 %	1,318/22,346 ≈ 5.90 %	1,518/18,564 ≈ 8.18 %

background. We then compare the manual labels to the automatic detection (task/background) at each frame for each detected blob. The sensitivity of detection can be adjusted using threshold T in Eqs. 13 and 14. However, since the value of $\frac{\Pr(\text{Task})}{\Pr(\text{Bg})}$ is also unknown, we tune for:

$$T' = T\frac{\Pr(\text{Bg})}{\Pr(\text{Task})} \tag{15}$$

Figure 10 demonstrates the changes in the sensitivity of detection as T' goes to zero. In this, precision and recall values are shown as well as the F_2 measure, where F_β measure is defined as:

$$F_\beta = (1+\beta^2)\frac{\text{precision}\cdot\text{recall}}{(\beta^2\cdot\text{precision})+\text{recall}} \tag{16}$$

Derived by van Rijsbergen [15], F_β measures the effectiveness with respect to a user who considers recall β times more important than precision. It can be seen that as the value of T' approaches zero the value of recall increases at the expense of precision. For $T' \approx 0$ recall is 87.7 %, while precision is merely 18.3 %. Using a very small value for T' is equivalent to having no background model where any sequence with a probability bigger than zero of being observed from the task model is accepted as a task. The F_2 measure reaches its peak value at around $T' = 4$, where recall equals 79.0 % and precision is 60.2 %.

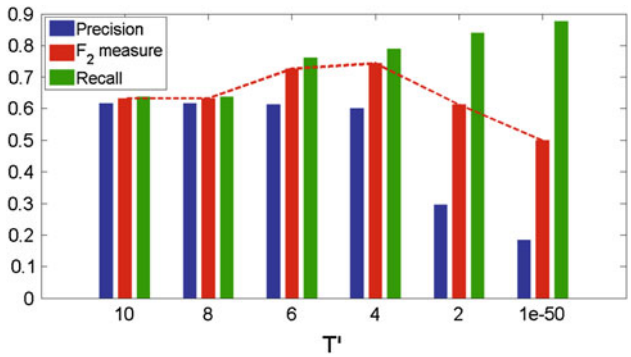


Fig. 10 Precision, recall and the F_2 measure at different T' values

Unlike the oblique views, the overhead view offers a straightforward ground plane estimation of position. Thus, as was described in Sect. 3.2, determining the task ID is easily performed using a closest point trajectory matching of the segmented task to the task map. Table 3 summarizes the overall detection of individual tasks on the aggressively pruned (AP) and the loosely pruned (LP) testsets. These results have been obtained using $T_1' = 4$, for which the best F_2 measure is obtained. Correct detection rates are also reported with $T_2' \approx 0$ to support a discussion on the detection of task 4. Table 4 shows the number of samples per task in the AP and LP testsets. Note that the detection rates are generally lower in LP test set compared to the AP test set.

The recognition performance of the tasks is mainly influenced by how well they can be detected from the background. There is little confusion between the tasks, due to their distinctive trajectories. This is true for all the tasks except for tasks 1 and 5, where they start from and end in nearby positions and their trajectories are similar. This is one of the drawbacks of using the overhead view. In this, the pickup rack for task 5 is situated directly above that for task 1. A significant part of the error in the detection of task 1 and task 5 is due to the confusion between these task labels. An 84.22 % correct recognition rate can be achieved for task 1 and 5 (with $T_1' = 4$, AP test set) if the two tasks are presumed to be the same. While in comparison smaller recognition rates of 70.82 and 62.35 % are obtained for task 1 and task 5 individually. The other point is regarding the detection of task 4. When experimenting with $T_1' = 4$, task 4 is left undetected in the AP testset and it is poorly detected (detection rate ~13 %) in the LP testset. This is mainly due to the fact that the pickup rack for this task is partially outside the field of view. Thus, the handling activity, here the pickup, is barely perceived, or it is not perceived at all. Decreasing the value of T' and thereby allowing for more potential task sequences to be detected, we still obtain low detection rates of 52 and

Table 3 Task detection by task ID

	Task 1 (%)	Task 2 (%)	Task 3 (%)	Task 4 (%)	Task 5 (%)	Task 6 (%)
$T_1' = 4$, AP test set	70.82	97.79	100	0	62.35	86.03
$T_1' = 4$, LP test set	58.64	87.73	70.65	12.58	60.49	86.03
$T_2' \approx 0$, AP test set	76.46	98.09	100	52.19	81.02	100
$T_2' \approx 0$, LP test set	69.58	88.00	92.39	64.51	75.51	100

Table 4 Number of samples per task

	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
AP test set samples	939	1674	597	799	943	458
LP test set samples	1,134	1,866	920	1,606	1,172	458

65 % for this task. Furthermore, task 4 is a more complex task than that which has been modelled by our task model. This is the only task wherein two parts are picked up and placed separately on the welding cell. As can be seen in Fig. 7, the put down point for these parts are situated at the opposite sides of the welding cell. However, the task model terminates when the first part has been placed. Due to this premature termination, the latter part of the task is left undetected. Apart from task 4, we obtain high detection rates for the remaining tasks.

4.4 Comparisons and discussions

The same workflows in the same industrial environment have also been examined by Veres et al. [19] and Voulodimos et al. [20]. The data from these workflows have been recorded via five cameras, four of which are captured from an oblique view and one is the overhead camera. The work presented here analyses the images from the overhead camera while [19,20] both work with oblique views. Therefore a direct comparison of the results is not possible. Furthermore the ground truth task labels differ in those works. Albeit, a comparison of the approaches, the solutions they offer, and the expected outcomes will be beneficial for any potential future application.

The overhead view is often criticized for having less visual cues for human activity recognition as compared to frontal or side views. On the other hand it was found to be impractical to track subjects by detection in the oblique views due to the severity of occlusion and clutter, and thus both [19,20] resorted to using a global scene descriptor which captures the properties of the entire image. In these, coarse scale features are detected using local motion monitors which reduce every 100 by 100 grid of pixels (with an overlap of 50 %) to a single logical value and therefore losing most of the fine scale visual cues for activity recognition. It appears that the motion-grid which was used in [19,20] mainly captures the trajectory of moving elements within the frames. Unlike [19,20], the focus of this paper is to detect the activities which can describe a worker picking up a part from a rack carrying it and placing it in on the welding cell. A method which would solely rely on the trajectories to detect a task would detect a task whenever a person so much as walks to the welding cell from the vicinity of a rack which is obviously not desirable.

As mentioned before the goal in [19,20], as in this work is to detect a set of predefined tasks of a workflow. However, all these approaches are different in what they offer as result. In [19], each frame is labelled with a task label using measurements obtained up to the current frame. In [20], sequences of images are probed with a constant-sized window to segment a task. Once an image sequence has been segmented as a task a task label is assigned to all the frames in that sequence. The sequence of detected tasks in a workflow is then continuously re-analysed for the most probable sequence of tasks given the prior knowledge. In this, the initial label of a task might change as more tasks are observed and the best sequence of tasks and the final task labels are determined when the entire workflow has been observed.

One of the main differences in this work to the two works mentioned in the previous paragraph is that, here the task labels are not assigned to frames but they are assigned to the detected workers. Thus the task is also spatially segmented. More importantly the tasks in this environment are not temporally separable. In other words, more than one task might be happening at each one time. We have noted that about 17 % of the frames which contain tasks, contain more than one task. Veres et al. [19] and Voulodimos et al. [20] have not discussed this and it is not clear how they handle these instances. We automatically label detected workers with potential task labels while the task is being performed therefore we are able to provide a live commentary like output. For example, assume two people have been detected in the scene, an output such as following may be generated: (person 1: task 5—picking up part from the rack), (person 2: not performing a task). Once a sequence arrives at the final state of the task HMM the task is verified and the detected sequence is labelled with the corresponding task label. Otherwise the sequence is labelled as background.

Table 5 compares the task detection performances. Keeping in mind that the results are reported from images captured with different cameras and the fact that the task classification is performed per frame in [19,20] and per person detection

Table 5 Comparison of task detection performances of this work and those by Veres et al. [19] and Voulodimos et al. [20]

	Camera	Unit	Task 1 (%)	Task 2 (%)	Task 3 (%)	Task 4 (%)	Task 5 (%)	Task 6 (%)	Background (%)	Welding (%)
Veres et al. [19]	Cam 1	PF	19.6	65.5	57.2	61.3	66.6	77.5	80.5	81.4
Voulodimos et al. [20]	Cam 1	PF	88.1	92.7	90.7	74.8	67.1	90.7	82.7	–
Voulodimos et al. [20]	Cam 2	PF	98.6	96.8	98.2	78.0	61.1	85.0	82.7	–
This paper ($T_1' = 4$, AP)	Cam 5	PDPF	70.8	97.8	100	0	62.4	86.0	98.2	–
This paper ($T_2' \approx 0$, AP)	Cam 5	PDPF	76.5	98.1	100	52.2	81.0	100	86.9	–

PF Per frame, *PDPF* per person detection per frame

per frame in this work, we refrain from discussing small differences. From Table 5 it is immediately obvious that the work by Veres et al. [19] is the only work which can detect the welding phase. It should also be noted that although we had to remove samples from our test set due to dropped frames these frame drops do not occur in the data from the other cameras thus the results reported by [19,20] are obtained on more data instances. For tasks 2, 3, 5 and 6 our results are similar to the best performance. Task 4 is not well handled in this work, as was discussed before, this is due to the starting point being partially outside of the field of view and having two end points instead of one would require the definition of a more complex task HMM. For task 1 our results are lower than that by [20] but outperform the results by [19]. Equally as important as detecting the tasks is distinguishing what is not a task; this is the background, as we call it here, also called task 8 by [19] and task 7 or the void by [20]. As previously discussed, instead of assigning task labels to frames, here the labels are assigned to detections within frames. Although there might be multiple detections on one frame, we find that there are about 40 % fewer instances of detections to classify than there are frames. This is due to periods when no activity occurs and thus there are no detections. It would be easy to classify these frames as background. Even without these easy test samples we obtain a high classification rate of 98.2 % for the background. In comparison, the 81 and 83 % classification rates which were obtained by [19,20] respectively appear low especially considering the large number of instances of the background. Even when $T_2^{'} \approx 0$ is used for which the competing background HMM is effectively disabled and every sequence with the smallest chance of being detected as a task is accepted as a task, still an 86.9 % recognition rate is obtained for the background. This demonstrates that our approach is much more specific in what it recognizes as a task as opposed to the other actions with can occur in this environment.

5 Conclusions and further work

In this paper, a hierarchical approach has been adopted for modelling human motion and understanding behaviour through visual sensors in a complex environment. It has been shown that although the output from each stage of the analysis is not error free, the final task detection obtains promising results by exploiting a potential surfeit of information and handling the uncertainties between the layers of the hierarchical process. In this, HMMs are used to model the structure, with regards to the context, and to handle probabilistic inputs.

The overhead view is shown to be superior to the oblique views in potentially posing: lesser occlusion, simplified detection and tracking, and straightforward estimation of position. On account of dealing with a manageable detection and tracking problem a spatially localized solution for task detection was achieved, which is in contrast with the global scene descriptor approach used for the oblique views [19,20].

The definition of tasks is taken directly from the workflow description. The activity categories are derived by splitting the task sequences. Although a basic set of activities was considered, this set is shown to be adequate for recognizing the tasks in this environment. Note that there are many similar, albeit not identical, working cells within this manufacturing environment which can be modelled and analyzed in the same manner. Finally, as a future work avenue, it might be beneficial to automatically explore the space of the shape and motion features for an alternative set of activity categories.

References

1. Adam, A., Rivlin, E., Shimshoni, I., Reinitz, D.: Robust real-time unusual event detection using multiple fixed-location monitors. Trans. Pattern Anal. Mach. Intell. **30**(3), 555–560 (2008)
2. Arbab-Zavar, B., Bouchrika, I., Carter, J., Nixon, M.: On supervised human activity analysis for structured environments. In: International Symposium on Visual Computing, Las Vegas, USA (2010)
3. Behera, A., Cohn, A.G., Hogg, D.C.: Workflow activity monitoring using dynamics of pair-wise qualitative spatial relations. In: International Conference on MultiMedia Modeling, Klagenfurt, Austria (2012)
4. Bobick, A.F.: Movement, activity and action: the role of knowledge in the perception of motion. Philos. Trans. R. Soc. Lond. Ser. B Biol. Sci. **352**(1358), 1257–1266 (1997)
5. Breitenstein, M.D., Grabner, H., Gool, L.V.: Hunting nessie—real-time abnormality detection from webcams. In: International Conference on Computer Vision (ICCV) Workshop on Visual Surveillance, pp. 1243–1250, Kyoto, Japan (2009)
6. Hu, M.: Visual pattern recognition by moment invariants. IEEE Trans. Inf. Theory **8**(2), 179–187 (1962)
7. Kosmopoulos, D.I., Doulamis, N.D., Voulodimos, A.: Bayesian filter based behavior recognition in workflows allowing for user feedback. Comput. Vis. Image Underst. **116**(3), 422–434 (2012)
8. Nater, F., Grabner, H., Gool, L.V.: Exploiting simple hierarchies for unsupervised human behavior analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco (2010)
9. Nater, F., Grabner, H., Gool, L.V.: Unsupervised workflow discovery in industrial environments. In: International Conference on Computer Vision (ICCV) Workshop on Visual Surveillance, pp. 1912–1919, Barcelona, Spain (2011)
10. Nguyen, N.T., Phung, D.Q., Venkatesh, S., Bui, H.: Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 955–960 (2005)
11. Oliver, N., Garg, A., Horvitz, E.: Layered representations for learning and inferring office activity from multiple sensory channels. Comput. Vis. Image Underst. **96**(2), 163–180 (2004)
12. Padoy, N., Mateus, D., Weinland, D., Berger, M.O., Navab, N.: Workflow monitoring based on 3d motion features. In: ICCV Workshop on Video-oriented Object and Event Classification, Kyoto, Japan (2009)

13. Pinhanez, C.S., Bobick, A.F.: Intelligent studios: modeling space and action to control tv cameras. Appl. Artif. Intell. **11**(4), 285–305 (1997)
14. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 57–286 (1989)
15. Rijsbergen, C.J.V.: Information Retrieval, 2nd edn. Butterworth-Heinemann, Newton (1979)
16. Schindler, K., van Gool, L.:Action snippets: how many frames does human action recognition require? In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK (2008)
17. Shi, Y., Huang, Y., Minnen, D., Bobick, A., Essa, I.: Propagation networks for recognition of partially ordered sequential action. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 862–869, Atlanta, USA (2004)
18. Somol, P., Pudil, P., Novovičová, J., Paclík, P.: Adaptive floating search methods in feature selection. Pattern Recognit. Lett. **20**(11–13), 1157–1163 (1999)
19. Veres, G., Grabner, H., Middleton, L., Gool, L.V.: Automatic workflow monitoring in industrial environments. In: Asian Conference on computer Vision, Queenstown, New Zealand (2010)
20. Voulodimos, A., Kosmopoulos, D., Veres, G., Grabner, H., Gool, L.V., Varvarigou, T.: Online classification of visual tasks for industrial workflow monitoring. Neural Netw. **24**(8), 852–860 (2011)

Author Biographies

Banafshe Arbab-Zavar received her B.Sc. degree in computer engineering from the Ferdowsi University of Mashhad, Iran, in 2004, and her Ph.D. degree in image processing from the School of Electronics and Computer Science, University of Southampton, Southampton, UK, in 2009. Since then she has been working as a research fellow and a research engineer at the University of Southampton. She is currently with the IT Innovation Centre, University of Southampton. Her research interests include image processing, feature extraction, biometrics, and human motion analysis.

John N. Carter (M'90) received his B.A. degree in experimental physics from Trinity College, Dublin, Ireland, and Ph.D. degree in astrophysics from the University of Southampton, Southampton, UK. In 1985, he changed discipline and joined the School of Electronics and Computer Science, University of Southampton, as a lecturer, researching in signal and image processing, where he is currently a senior lecturer with the Communications, Signal Processing and Control Research Group. In the past, he has worked on programs as diverse as diesel engine diagnostics and vocal tract imaging. A recent success in this field is the development of a new dynamic form of magnetic resonance imaging, which makes it possible to reconstruct high time resolution multiplanar views of the human vocal tract while a subject is repeating a short phrase. His current research interest is in the general area of 4-D image processing, i.e., analysing sequences of images to extract both 2-D and 3-D features, exploiting coherence over the whole sequence, i.e., imposing simple smoothness and continuity constraints. This has applications in object tracking and feature detection, where it is tolerant to high levels of noise and missing data. This has found application in biometrics, particularly in automatic gait analysis where he has had grants from DARPA, MOD and European Union.

Mark S. Nixon is a Professor in computer vision at the University of Southampton, Southampton, UK. His research interests are in image processing and computer vision. His team develops new techniques for static and moving shape extraction, which have found applications in automatic face and automatic gait recognition and in medical image analysis. His team contains early workers in face recognition, and they later came to pioneer gait recognition and more recently joined the pioneers of ear biometrics. His vision textbook, with A. Aguado, *Feature Extraction and Image Processing* (Academic), reached second edition in 2008. With T. Tan and R. Chellappa, the book *Human ID Based on Gait* is part of the Springer Series on Biometrics and was published in 2005. He has chaired or had major involvement in many conferences (BMVC, AVBPA, IEEE Face and Gesture, ICPR, ICB, IEEE BTAS) and has given many invited talks. Dr. Nixon is a Fellow at IET and FIAPR.