Exploiting Petri-net Structure for Activity Classification and user Instruction within an Industrial Setting

S. F. Worgan
School of Computing
University of Leeds
Leeds, LS2 9JT
S.Worgan@leeds.ac.uk

A. Behera
School of Computing
University of Leeds
Leeds, LS2 9JT
A.Behera@leeds.ac.uk

A. G. Cohn
School of Computing
University of Leeds
Leeds, LS2 9JT
A.G.Cohn@leeds.ac.uk

D. C. Hogg
School of Computing
University of Leeds
Leeds, LS2 9JT
D.C.Hogg@leeds.ac.uk

ABSTRACT

Live workflow monitoring and the resulting user interaction in industrial settings faces a number of challenges. A formal workflow may be unknown or implicit, data may be sparse and certain isolated actions may be undetectable given current visual feature extraction technology. This paper attempts to address these problems by inducing a structural workflow model from multiple expert demonstrations. When interacting with a naive user, this workflow is combined with spatial and temporal information, under a Bayesian framework, to give appropriate feedback and instruction. Structural information is captured by translating a Markov chain of actions into a simple place/transition petri-net. This novel petri-net structure maintains a continuous record of the current workbench configuration and allows multiple sub-sequences to be monitored without resorting to second order processes. This allows the user to switch between multiple sub-tasks, while still receiving informative feedback from the system. As this model captures the complete workflow, human inspection of safety critical processes and expert annotation of user instructions can be made. Activity classification and user instruction results show a significant on-line performance improvement when compared to the existing Hidden Markov Model or pLSA based state of the art. Further analysis reveals that the majority of our model's classification errors are caused by small de-synchronisation events rather than significant workflow deviations. We conclude with a discussion of the generalisability of the induced place/transition petri-net to other activity recognition tasks and summarise the developments of this model.

ICMI'11, November 14–18, 2011, Alicante, Spain.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Applications and expert systems, Knowledge representation formalisms and methods, Learning; I.5 [**Pattern Recognition**]: Models, Clustering, Applications

General Terms

Algorithms, Experimentation, Performance

Keywords

Petri-net, qualitative spatial relation, histogram of pairwise relation (HoPR), workflow modeling, Hidden Markov Model (HMM)

1. INTRODUCTION

Despite increasing automation in manufacturing, the adaptability and capabilities of human workers remains vital in industrial environments [26]. However further efficiencies can be identified as current experts train and supervise naive workers on an ad-hoc basis, tutoring by example and correction. We can improve upon this process through visual activity recognition and augmented reality instruction, enabling ambient multi-modal interaction between system and user. By deploying expert systems, training costs can be lowered and safety critical tasks can be continuously monitored. Before this potential can be realised however, we have to overcome a number of existing problems. Frequently, the maintenance or construction task under consideration will not be directed by a formal workflow model and the action sequence required to complete the task will be an implicit part of the experienced user's knowledge. To formalise this ad-hoc approach, we need to capture implicit expert knowledge as an explicit workflow process. Through accurate activity recognition we can then guide the user through the complete task sequence.

Historically, activity detection approaches have focused upon wearable sensor networks, operating over clearly distinguishable highly active gestures. The features of an established set of actions (walking, standing, jogging, hopping, climbing, etc..) coupled with wearable accelerometers have resulted in accurate classification results [17, 10, 12].

Current research employs a number of techniques ranging from Echo State Networks [21] to boosted-SVMs [12] to give a true positive rate between 60 and 90%. Building upon these techniques, a number of researchers have attempted to classify more challenging actions within an industrial setting [21, 13, 7, 14, 24] by supplementing accelerometer data with microphones [13], ultrasonic hand tracking [19] or marker based video tracking [7].

Within these industrial settings, existing Hidden Markov Model (HMM) [16] activity recognition systems [24, 14] perform well on offline sequential action recognition tasks but our own research suggests that when required to classify temporally incomplete or underspecified data, the resulting deterioration in performance can be marked. In order to exploit the constrained sequential nature of most workflows, these approaches [24, 14] train HMMs on large training sets (one for each atomic action) and then employ a Context Free Grammar (CFG) to determine their sequential likelihood. For example, the Gesture and Activity Recognition Toolkit builds directly upon the HTK recognition library [25] – hand-coding the workflow as a CFG and training each action on a separate HMM. In testing, the model constructs a Viterbi path [6] with fixed transition functions from one model to the next as determined by a stochastic or deterministic CFG. As stated by Ivanov, [11] "The grammar and parser provide longer range temporal constraints, disambiguate uncertain low-level detections, and allow the inclusion of a priori knowledge about the structure of temporal events in a given domain."

If a system is to continuously monitor and instruct a live user, offline classification accuracy must be maintained in an online setting. In the HMM/CFG approach, the detection accuracy is often dependent upon the final re-estimation of the Viterbi path, far too late for live activity recognition and user instruction. In addition to this, CFG structures are incapable of capturing certain real workflow features such as limited depth recursion and temporally independent subtasks, while hand crafted workflows can fail to capture expert behaviour [20].

Avoiding these problems, we will build upon the general approach of workflow mining [8] and induce a petri-net to capture the probable structure of the observed workflow. Unlike the workflow limitations of previous activity detection approaches [14], petri-net formalisms can capture advanced structural restrictions and can be easily translated into other – human readable – workflow formats. Formed from a graph of states, transitions, arcs and markers, petri-nets capture "an intuitive graphical representation of the processes being modelled" [4] while possessing strong mathematical foundations. Unlike other structural representations petri-nets can model both limited depth recursion and temporally independent subtasks. In general terms, a marker location in a current state represents the overall state of the system (*i.e.* the current state of the workbench). Markers can move from state to state across arcs and through transitions, with each marker recording the current state of an identified sub-sequence. Sub-sequence switching then becomes possible as each marker records the progress within each independent task.

A combination of probabilistic reasoning and petri-net structural constraints has previously been demonstrated in [2]. By exploiting a hand crafted petri-net to capture domain knowledge and combining it with uncertain observations, various activity classes can be differentiated. This paper takes a similar approach but will instead induce a petri-net workflow model to record the overall global state of the system. Due to the persistent marker values of a petri-net, users are free to switch between multiple activity streams in the course of completing a global task. This will provide a high degree of user task autonomy while maintaining a useful level of interaction with the system.

In the next section, this induced structure will be integrated with a naive Bayes model to address the identified HMM/CFG shortcomings. Specifically, this model:

1. Maintains 'on-line' classification accuracy from partial information – enabling the accurate instruction of a live user.
2. Exploits a marker based transition network to capture the structural properties of the workbench.
3. Provides structural classification constraints that can be translated into real workflows.
4. Successfully trains over relatively sparse annotated data sets.
5. Induces and exploit workflows from example activity sequences – capturing what is done not what 'should' be done.

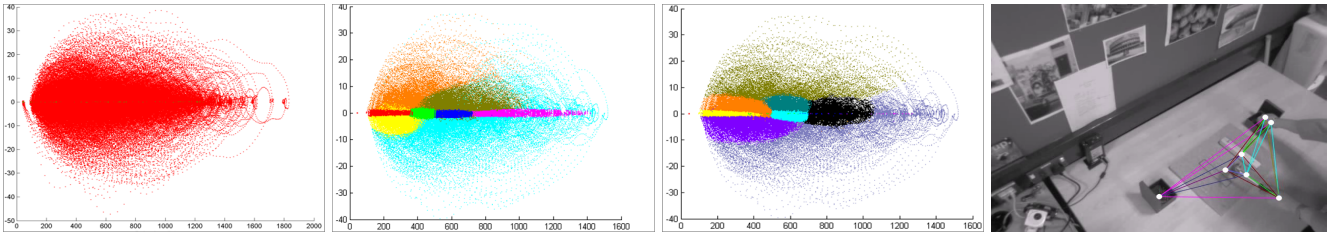
We will then, in Section 3, assess our model's classification and user instruction accuracy before concluding in Section 4.

2. WORKFLOW INTERACTION

To monitor the naive user, we record the object configuration of the workbench and wrist positions of the user at any given time. To counter any residual noise or occlusion we first quantise these continuous observations as the latent states of a Hidden Markov Model [18]. Separately, we induce the structure of our place/transition petri-net from multiple annotated expert examples. Once induced, our naive Bayes activity recognition and user instruction model operates over the quantised workspace configuration. We will now describe each part in detail.

2.1 Feature Quantisation

A general workbench consists of a number of key objects $[o_1 \dots o_i \dots o_n]$. At each time step, the relation between pairs of objects o_i and o_j, with positions $\boldsymbol{x}_t^i$ and $\boldsymbol{x}_t^j$ at time t, is represented in a view-invariant fashion by a real valued vector composed of the separation and the first derivative of separation with respect to time i.e. $\boldsymbol{r}_t^{i,j} = (d_t^{i,j}, \dot{d}_t^{i,j}) \in \Re^2$, for $\forall i < j$ where $d_t^{i,j} = \|\boldsymbol{x}_t^i - \boldsymbol{x}_t^j\|$. For notational convenience, we vectorise the set of pair-wise relations $\{\boldsymbol{r}_t^{i,j}, i < j\}$ into $\{\boldsymbol{r}_t^m, 1 \leq m \leq M\}$, where $M = \frac{n*(n-1)}{2}$. These real vectors $\boldsymbol{r}_t^m$ are then quantised, q_t^m, as the latent states of a Gaussian Hidden Markov Model, χ, to minimise noise and capture the temporal dependencies. As shown by [18] this quantisation approach outperforms k-means clustering for visual activity classification. A complete workflow sequence of T time steps will have M parallel series of relational features i.e. $R = [\boldsymbol{r}_t^m]_{M \times T \times 2}$ and after discretisation it will be represented by corresponding HMM states $S_\chi = [q_t^m]_{M \times T}$. Where T captures the entire complete activity



(a) 2-state HMM (b) 8-state HMM (c) 12-state HMM (d) Workbench configuration of pairwise relationships.

Figure 1: A partition of the recorded pairwise relationships with distance on the x-axis and the first derivative on the y-axis, formed from 2, 8 and 12 latent HMM states respectively. The complete configuration of relationships is shown on the right.

sequence, with each $t \in T$ recorded at 50Hz. Figure 1 shows the resulting partition of the pairwise relationships for a varying number of latent HMM states. The count of these quantised relationships then forms a Histogram of Pairwise Relationships (HoPR), $h_{q_m^t}$, at each timestep, where a set of relationships captures the current spatial configuration, this is then concatenated over a sliding window of size w, $\boldsymbol{h}_t = (h_{q_t^1 - w}, \cdots, h_{q_t^M})$. The naive Bayes model, detailed in Section 2.3, then operates over these features.

2.2 Workflow structure induction

To form an explicit workflow from implicit expert knowledge we first train a Markov chain on multiple annotated, expert demonstrations of the complete task. The Markov chain ψ is defined as follows:

$$\psi = (A_\psi, P_\psi, \pi_\psi) \tag{1}$$

$$P_\psi = A_\psi \times A_\psi \tag{2}$$

where A_ψ is the set of possible action states, P_ψ the associated transition probabilities, and π_ψ the initial state probabilities. The Markov chain transition probabilities are then obtained from the frequencies of observed transitions in the annotated expert set.

With each state capturing a unique action, the trained Markov chain identifies the highly probable state transitions from one action to the next and the initial states identify the initial actions within the workflow. Typically, this trained first-order structure, extended to a Hidden Markov Model to account for uncertain observations, would be sufficient to identify the temporal relationships between actions. However, workflows typically consist of temporally independent subtasks and certain actions only become possible once local tasks have aligned to form a specific global configuration state. A HMM is incapable of capturing the global consequences of local actions as no global record of local changes can be made without an intractably large state-space or an equally intractable second-order formalisation of the model. Accordingly, in this paper, the trained HMM will be converted into a simple place/transition petri-net [15] enabling marker configurations to give the global state of the workbench after an observed sequence of locally enacted actions. This petri-net θ is defined as follows:

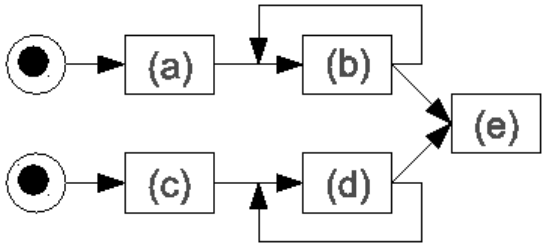


Figure 2: A simplified representation of an induced petri-net, showing parallel sequences and recursive processes.

$$\theta = f(\psi) = (S_\theta, T_\theta, W, M_0) \tag{3}$$

$$S_\theta = A_\psi \tag{4}$$

$$T_\theta = \{(a_i, a_j) : P_\psi(a_i, a_j) \geq r\} \tag{5}$$

$$W : (S_\theta \times T_\theta) \times (S_\theta \times T_\theta) \tag{6}$$

$$M_0 = \{a_\theta^j : \pi_\psi^j > 0\} \tag{7}$$

Forming a tuple $(S_\theta, T_\theta, W, M_0)$, the resulting petri-net consists of a set of places S_θ, transitions T_θ, initial marker values M_0, and a multiset of arcs, W joining places and transitions, Equation 3. Transitions, Equation 5, are simply formed by eliminating those that fall below a certain likelihood within the initial Markov chain. Each transition of the Markov chain, ψ, captures the fact that it is possible to transition from one action to the other. We then define each place $s_\theta \in S_\theta$ of the petri-net as an action potential for each $a_i \in A_\psi$ in the Markov chain. This action potential is tied to the unobserved state of the workbench as certain actions are only possible given certain configurations. Initially, each place is assigned to an action identified by the Markov chain, Equation 4, it is then the assignment of marker values and the inherent properties of the petri-net representation that converts transient actions into static configurations. The appropriate arcs, Equation 6, then results as a structural property of the petri-net, and all transition evaluation functions are set to true. The initial marker configuration is determined by the initial state probability of the Markov chain where all possible initial states are assigned a marker, as defined in Equation 7.

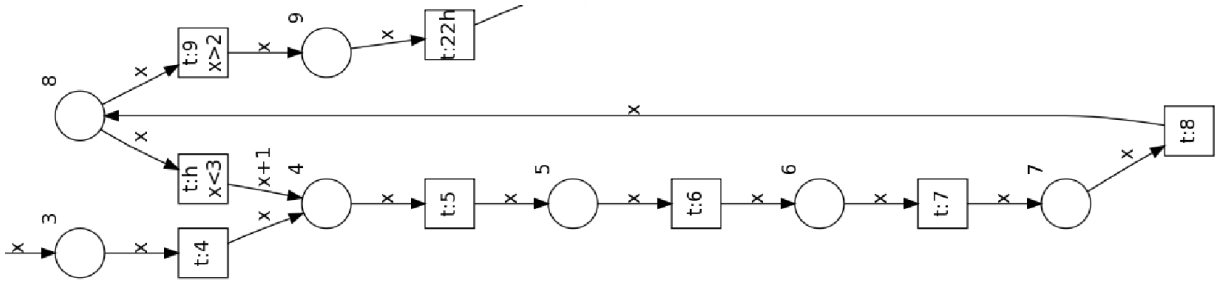


Figure 3: A detailed view of component (d) from Figure 2, by transitioning from a place/transition to a coloured petri-net we can limit the number of recursive transitions within the workflow.

A highly simplified representation of a typical induced petri-net is shown in Figure 2. Abstracting away from place or transition details, we can see that two independent tasks, on the same workbench, are each represented by an initial chain of events (components (a) and (c)) before entering a recursive action phase (components (b) and (d)) and then completing the overall task (component (e)). Due to the conversion of a petri-net representation and the introduction of marker states, the user is free to switch between these two tasks at will, as the markers record the local workbench configuration at the time of the switch. By comparison, two connected parallel HMMs would be unable to keep track of the previous local state whenever a switch occurred.

Figure 3 presents a more detailed view of component (b) in Figure 2 and demonstrates the transparency of this workflow representation. This transparency enables two kinds of expert annotation which can then be integrated into the live interaction system. Firstly, places can be annotated to provide a richer semantic representation; secondly, markers, arcs and transitions can be developed to capture more subtle structural constraints. For example, component (b) describes the hammering of a nail and the recursive relationship captures the hammering of multiple nails, by transitioning from a place/transition petri-net to a coloured petri-net we can enforce limited depth recursion as a structured constraint. By incrementing each marker as it transitions around the recursive loop and modifying the transition function to prevent markers over a certain value from further recursion we can ensure that the user hammers three *and only three* nails. Overall, this petri-net, whether coloured or place/transition, captures the structural constraints of the workflow and tracks workbench state changes in response to user actions.

As we do not directly observe this workbench the induced petri-net infers the current state by assuming that an observed transition from one action to another implies that the underlying state has changed. Under certain scenarios this assumption will not always hold true as switching from one asynchronous state to another may generate transitional actions. As these transitional actions represent apparent workflow violations they result in erroneous marker transitions within the petri-net model and are treated as insertion noise by the unified Naive Bayes model. This model accounts for such noise by combining the induced petri-net structural constraints with temporal and spatial observations to provide live user monitoring and instruction.

2.3 Naive Bayes Model Formation

We define a workflow instance e as a tuple (H_e, g), where $H_e = h_1 \cdots h_t$, representing a series of HoPR features, and g assigns an action to each HoPR window. Activity $a_i \in A$ occurs in e if there exists $h_j \in H_e$ with $g(h_j) = a_i$.

In total, three key components are trained on the observed data and combined into a naive Bayesian model, the first captures the temporally independent spatial HoPR features, h_t, by training a Support Vector Machine (SVM) equipped with a RBF kernel using a 1-vs-all methodology. This then returns a complete probability distribution over the candidate atomic actions at time t, $P(a_t^i|h_t, \Delta)$.

Accordingly, we define a set of Support Vector Machines (SVMs) Δ, where $|\Delta| = |A|$, and a function f_Δ assigning each $\delta_i \in \Delta$ to a corresponding action $a_i \in A$. Each SVM then solves a binary classification problem where for each δ_i, a set of input HoPR features is formed from all e drawn from the training set with a corresponding classification output set, Y_{δ_i}, such that $y_i \in 1, -1$, assigning 1 to all positive examples of the action and -1 otherwise. Each δ_i then attempts to solve the binary classification problem, employing the RBF kernel [5]. The resulting output for each action at time t, $P(a_t^i|h_t, \Delta)$ is then the probability of action a_t^i drawn from a complete probability distribution over the combined, normalised, beliefs of Δ.

The second model gives a rough measure of absolute temporal values (e.g. putting away tools typically occurs at the end of a workflow); accordingly a Gaussian, $\omega_i \in \Omega$, is trained for each recognised atomic action, $a_i \in A$. These Gaussians, Equation 8, are centered on the average absolute duration midpoints, $a_{\mu,i}$, for each action with their variance set equal to the average event duration variance, $a_{\sigma,i}$. At each time step, t, a normalised probability distribution is provided by the combination of Gaussian models, Ω, $P(a_t^i|t, \Omega)$.

$$\omega(a^i, t) = \frac{1}{\sqrt{2\pi a_{i,\sigma}^2}} e^{-\frac{(t-a_{i,\mu})^2}{2a_{i,\sigma}^2}} \tag{8}$$

$$P(a_t^i|t, \Omega) = \frac{\omega(a^i, t)}{\sum_{j=1}^{|A|} \omega(a^j, t)} \tag{9}$$

Finally, to capture the structural transitions of the induced workflow, a transition probability matrix is calculated according to the induced petri-net structure, as shown in Equation 10. This structure, which captures the believed global configuration of the workbench at a given time, rep-

resents a latent property of the system. We are faced with two uncertainties, the accuracy of the spatial/temporal observations and the validity of the marker transitions that we believe have taken place up to this point. Accordingly, we maintain a complete belief distribution over all states of the induced petri-net, this is drawn from the complete distribution formed by the previous timestep, S_{t-1} with $s^j_{t-1} \in S_{t-1}$.

$$P(a^i_t|\theta, S_{t-1}) = \sum_{j=1}^{|S^{t-1}|} P(s^i_t|s^j_{t-1}, d^{i,j}_\gamma)P(s^j_{t-1}) \quad (10)$$

When considering the likelihood of a current state, s^i_t, we generate a reachability graph, γ, [15] from each possible petri-net state in turn. These reachability graphs are bi-directional and capture the number of marker transitions required to reach the proposed state given the marker configuration and structural constraints of the network. The likelihood that all of these transitions have taken place since the previous set of observations is captured by a discrete Gaussian distance measure, $f_\gamma(d^{i,j}_\gamma)$, where $d^{i,j}_\gamma$ equals the minimum number of transitions between s^i_t and s^j_{t-1}, Equation 11. This discrete measure captures the high likelihood that the current workbench remains unchanged with a decreasing likelihood the greater the number of steps taken to reach the proposed petri-net state, s^i_t. By considering the probability of the previous state $P(s^j_{t-1})$ combined with the probability that the system has transitioned from that state to the current state $P(s^i_t|s^j_{t-1})$ we return (in Equation 10) the combined probability that the workbench has reached a state $P(s^i_t)$ that makes the proposed action possible $P(a^i_t) = P(s^i_t)$. In use, the complete set of transition probabilities, modified by the prior probability of the previous state, $P(a^j_{t-1})$, are considered as possible routes when calculating the transition likelihood.

$$f_\gamma(d^{i,j}_\gamma) = \frac{1}{\sqrt{2\pi^2}}\, e^{-\frac{(d^{i,j}_\gamma)^2}{2^2}} \quad (11)$$

$$P(s^i_t|s^j_t - 1, d^{i,j}_\gamma) = \frac{f_\gamma(d^{i,j}_\gamma)}{\sum_{i=1}^{|S|} f_\gamma(d^{i,j}_\gamma)} \quad (12)$$

All three of these models, spatial, temporal and structural are assumed to be independent, and are combined according to a naive Bayes formalisation (Equation 13). At each timestep the most likely action a_{MAP} is selected given this combined probability distribution.

$$a_{MAP} = \arg\max_{a_t} P(a_t|h_t, \Delta)P(a_t|t, \Omega)P(a_t|\theta, S_{t-1}) \quad (13)$$

In use, a_{MAP} and a^{t+1}_{MAP} are supplied to the user, in a similar manner to Equation 10, Equation 14 selects the next likely action given the current believed state of the petri-net.

$$a^{t+1}_{MAP} = \underset{a_{t+1}, a_{t+1} \neq a_t}{\operatorname{argmax}} P(a_{t+1}|\theta, S_t) \quad (14)$$

3. RESULTS

The complete set of workflow examples consists of two complex tasks performed eight times by two different people on the same workbench. The workbench itself consists of 7 objects, each of which is identified by a VICON marker [1], while the motion of the user is identified by two further markers – one on each wrist. This forms a complete set of 9 *key objects* $[o_1 \cdot o_i \cdot o_9]$. Formed from 22 atomic actions and totalling approximately 20 minutes of data, this small dataset represents a realistic amount of training data, given expected industry constraints. The task itself captures a basic construction activity, where one batten is affixed to an object by three screws and other is affixed by three nails. Each atomic action corresponds to semantically meaningful components of this task, e.g. place batten, hammer nail.

For all of the following results we have used a HoPR window length of 6 with 32 quantised states, petri-net transitions were induced with a threshold of $r = 0.005$, these parameter values where found through experimentation. Given this dataset the induced petri-net structure captures the entire workbench configuration and the assumed independence of the two tasks, within the data this assumption remains valid as no transitional actions, arbitrarily switching between the two tasks, occur.

We will assess this model according to its classification accuracy, prediction accuracy and robustness to occlusion for leave-one-out training and testing. Given 16 training sequences, with 2 people performing 8 complete workflows each, we will train on 15 and test on the remaining one. The following results are then averaged over all permutations of this division. When considering classification accuracy we will compare it to the current HMM based state of the art, a pLSA topic model and an unconstrained multi-class SVM.

The HMM based state of the art was developed according to the work of Lyons [25] resulting in a HMM-CFG based activity recognition model. To ensure a fair comparison, the Context Free Grammar (CFG) structure was translated directly from the induced petri-net model. Obviously, the task switching abilities could not be captured by the CFG and each sub-sequence, once started, had to be completed. A separate left-right 5-state HMM was then trained on each identified atomic action with online classification accuracy recorded as the current Viterbi path re-estimation at the given time. This approach has demonstrated accurate classification results in activities ranging from Tai-Chi (69.2% classification, [3]) to workshop construction tasks (66% recall, [23]).

The pLSA-topic model [9, 22], operated over the displacement of objects in the workspace with each k-means clustered value represented as a word, w, and each action identified as a topic, z. For each topic, z_i, we compute $P(w|z_i)$ and when testing an unknown sequence (document, d) identify $z^* = \operatorname{argmax}_z P(z|d)$ according to the procedure described in [9]. For testing, the documents represent a sliding window of duration 0.25 second and is scanned over the workflow sequences with 50% overlap. We will also train a standard Support Vector Machine (SVM) model on the HoPR features to identify the benefits of the complete naive Bayes model (HoPR-NB).

3.1 Classification Accuracy

In testing the current state of the art, the HMM approach performed well for offline classification (77.2%) – given the final Viterbi path re-estimation. However, the task of live user instruction and interaction requires accurate real time online classification so that useful instructions can be given at the appropriate time and not simply after

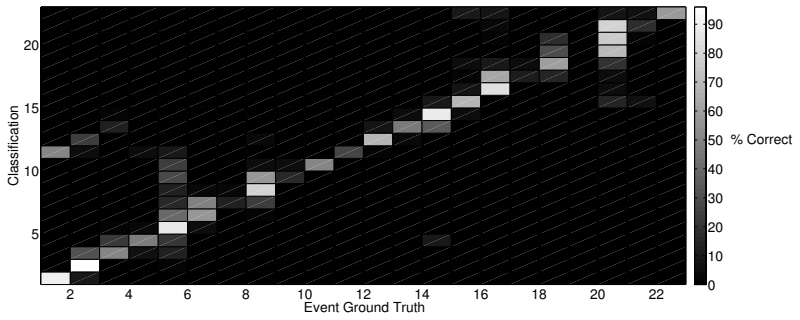


Figure 4: A confusion matrix for the HoPR-NB model.

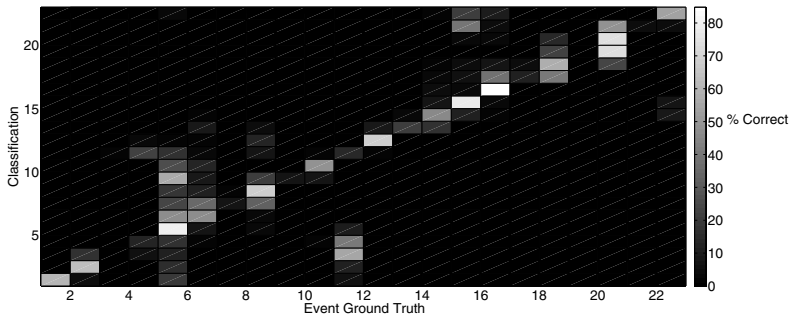


Figure 5: A confusion matrix for the HoPR-SVM model.

Model	Frame-wise accuracy %
Online-HMM:	12.2
pLSA:	36.8
HoPR-SVM:	62.5
HoPR-NB:	70.1

Table 1: Results showing a clear improvement in online performance for our model.

the event. As can be seen in Table 1, the current HMM state of the art deteriorates significantly when the complete workflow remains unknown. Further analysis reveals that the successful exploitation of the CFG requires knowledge of the end point of the sequence. Accordingly, the final re-estimation of the Viterbi path is significantly more accurate than all re-estimations up to that point. By comparison, HoPR-NB outperforms the current state of the art, the unconstrained SVM model (HoPR-SVM) and the alternative pLSA approach for leave-one-out classification accuracy. A significant amount of this improvement is clearly due to the multi-class SVMs operating over the HoPR features. However, results show that further gains can be made through the exploitation of the structural constraints found in the complete HoPR-NB model. We will now consider the nature of these improvements in greater detail.

3.2 Synchronisation Error

Further analysis of these classification results reveals, as shown in Figure 4, that due to the structural constraints of the petri-net the majority of classification errors occur at the transition points from one action to the next. This

Model	Safe%	Predictive%	Erroneous%
HoPR-NB:	11.6	56.3	32.1

Table 2: Results showing safe, predictive and erroneous workflow instructions.

compares favourably to the SVM errors, shown in Figure 5, where clearly nonsensical misclassification errors can be identified. The errors, identified in Figure 4, could be due to uncertainty over the annotation of the ground truth as the exact transition point from one action to the next is a matter of subjective judgement by the human annotators. Figure 4 also illustrates how errors are distributed for different actions, for example there is a wide spread of misclassification for action 5. The remaining errors, confusion between action 1 and action 11 for example, are due to an initial misidentification of the two sub-tasks with sub-task 1 capturing actions 1-10 and subtask 2 actions 11-21. In use, these synchronisation errors should not significantly degrade user experience as they typically result in a slight transition delay from one action to the next. This delay is significantly less intrusive than a clear misclassification.

3.3 Action Prediction Accuracy

The role of synchronisation errors can be further explored when we consider the action prediction accuracy results, Table 2. The majority of user action predictions remain accurate ('Predictive') while a number of predictions are incorrect but simply reinforce the users current state ('Safe'). These 'Safe' predictions are significantly less annoying than the remaining 'Erroneous' predictions, while the 'Predictive'

Occluded objects	HoPR-NB %
Screw driver	66.7
Wood piece:	66.9
Nail baton:	66.9
Hammer:	66.8
Nail box:	66.7
Screw baton:	66.8
Screw box:	66.6
Left wrist:	66.7
Right wrist:	67.1

Table 3: Performance for the HoPR-NB given the complete occlusion of one object.

output results in a successful interaction with the user. In safety critical systems a slight delay in instruction is preferable to a, potentially dangerous, erroneous instruction.

3.4 Occluded Objects

In our unmodified dataset some markers were naturally occluded for a total of 2.37% time steps. To further investigate the robustness of HoPR features to occlusion, we completely occluded each marker in turn for the duration of the workflow. The average performance of complete removal of an individual object in testing sequences for leave-one-out experiment is shown in Table 3.4. As shown by these results, our approach considers the spatio-temporal configurations (pair-wise relations) of objects positioned in 3D space and therefore, a workflow can be recovered even though one of the objects in the workspace is occluded.

4. DISCUSSION

Returning to our specific claims in Section 1, the combined temporal, spatial and structural information allows on-line classification accuracy to be maintained. The induced petri-net structure is human readable and can be edited or adjusted to suit other workflow representations. Live user instruction then consists of presenting the next available action given the current marker configuration. Our second claim is also fulfilled, as compared to the classification accuracy of the unconstrained SVMs, the decisions taken by the complete model result in a higher overall performance. Given only 20mins of training data our model is also robust to sparse training data (claim 4) as the combination of multiple sources of information in the naive Bayes model allows activity detection to take place in scenarios where the production of large training sets remains impractical. Finally, the induced transparent petri-net representation, claims 3 and 5, enable expert domain knowledge to be integrated into the classification and instructional accuracy of the system.

4.1 Future Work

In future work, the role of the petri-net workflow representation will be explored in greater detail. This workflow, and resulting stochastic model, can either be constructed by individuals with expert knowledge of the workflow under consideration or it can be induced from noisy unconstrained action classifications. In initial testing, induction from the results of the unconstrained SVM model proved accurate enough to enable subsequent accurate user instruction. This approach can then be adapted to any activity recognition task that can be formed from multiple independent subtasks. Further work will also have to consider how erroneous subtask transition actions can be accounted for by the petri-net workflow model.

4.2 Conclusion

This paper has considered the current HMM-CFG approach to continuous activity recognition: in previous work a high degree of off-line activity recognition accuracy could be maintained by combining the structural constraints of a CFG with the stochastic classification results of a set of HMMs. However, this approach faces difficulties when confronted with sparse training data, increasingly complex underlying workflow models and the requirement to provide on-line classification results. Accordingly, we have presented an alternative HoPR-NB approach that is capable of accurate on-line classification and can form complex workflow models from a limited set of observations. These workflow models exploit the state based petri-net formalism to maintain a global record of the current workbench configuration. This allows the system to keep track of the user as they switch between multiple independent independent subtasks. This represents an improvement over HMM or CFG approaches to structural knowledge as each is incapable of sustaining multiple simultaneous independent subtasks. Furthermore, as each component of the model is trained offline the combined naive Bayes approach can provide on-line activity classification and user instructions in real-time. Accurate continuous activity recognition is maintained over a wide range of expressive (hammering etc...) and discreet (button pressing, switch flicking) workshop activities, enabling informative live user direction and accurate safety critical monitoring of current workflow activity.

5. ACKNOWLEDGMENTS

The research in this paper is supported by EU under grant ICT-248290 via COGNITO (`www.ict-cognito.org`) project. We thank Dima Damen and Andrew Gee (Department of Computer Science, University of Bristol) for providing the Vicon dataset used in our experiments. We thank Elizabeth Carvalho (Center for Computer Graphics, Portugal) for supplying ground-truth for this dataset.

6. REFERENCES

[1] Vicon systems. `http://www.vicon.com`.

[2] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V. Subrahmanian, P. Turaga, and O. Udrea. A constrained probabilistic petri net framework for human activity detection in video. *IEEE Transaactions on Multimedia*, 10(8):1429–1443, 2008.

[3] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *CVPR*, pages 994–999, 1997.

[4] V. Bulitko. Machine learning for time interval Petri nets. *AI 2005: Advances in Artificial Intelligence*, pages 959–965, 2005.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[6] G. David Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[7] B. Hartmann, C. Schauer, and N. Link. Worker behavior interpretation for flexible production. In

World Academy of Science, Engineering and Technology, pages 494–502. CESSE, 2009.

[8] J. Herbst. Workflow mining with InWoLvE. *Computers in Industry*, 53(3):245–264, 2004.

[9] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artigicial Intelligence*, pages 22–29, 1999.

[10] T. Huynh and B. Schiele. Analyzing features for activity recognition. In *sOc-EUSAI '05*, New York, New York, USA, 2005. ACM Press.

[11] Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.

[12] N. C. Krishnan and S. Panchanathan. Analysis of low resolution accelerometer data for continuous human activity recognition. In *ICASSP '08.*, pages 3337–3340. IEEE, 2008.

[13] P. Lukowicz, J. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, and T. Starner. Recognizing workshop activity using body worn microphones and accelerometers. *Pervasive Computing*, pages 18–32, 2004.

[14] K. Lyons, H. Brashear, T. Westeyn, J. Kim, and T. Starner. GART: The gesture and activity recognition toolkit. In *Proc. of the 12th int. conf. on Human-computer interaction*, pages 718–727. Springer-Verlag, 2007.

[15] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.

[16] L. Rabiner and B. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

[17] N. Ravi, N. Dandekar, P. Mysore, and M. Littman. Activity recognition from accelerometer data. In *Proc. of the Nat. Conf. on AI*, volume 20. London; AAAI Press, 2005.

[18] M. Sridhar, A. G. Cohn, and D. C. Hogg. From video to RCC8: Exploiting a distance based semantics to stabilise the interpretation of mereotopological relations. In *COSIT*, 2011.

[19] T. Stiefmeier, G. Ogris, H. Junker, P. Lukowicz, and G. Troster. Combining Motion Sensors and Ultrasonic Hands Tracking for Continuous Activity Recognition in a Maintenance Scenario. *2006 10th IEEE International Symposium on Wearable Computers*, 1:97–104, Oct. 2006.

[20] W. M. P. van der Aalst, B. Vandongen, J. Herbst, L. Maruster, G. Schimm, and a. Weijters. Workflow mining: A survey of issues and approaches. *Data & Knowledge Engineering*, 47(2):237–267, Nov. 2003.

[21] G. Veres, H. Grabner, L. Middleton, and L. Van Gool. Automatic Workflow Monitoring in Industrial Environments. In *Asian Conference on Computer Vision*, pages 1–14, 2010.

[22] X. Wang, X. Ma, and W. E. L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Trans. PAMI*, 31:539–555, 2009.

[23] J. a. Ward, P. Lukowicz, G. Tröster, and T. E. Starner. Activity recognition of assembly tasks using body-worn microphones and accelerometers. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1553–67, Oct. 2006.

[24] T. Westeyn, H. Brashear, A. Atrash, T. Starner, and A. Drive. Georgia Tech Gesture Toolkit : Supporting Experiments in Gesture Recognition. In *ICMI '03*, pages 85–92, 2003.

[25] S. Young. HTK: Hidden Markov Model Toolkit: Design and philosophy.

[26] D. Zuehlke. Smartfactory – from vision to reality in factory technologies. In *Proceedings of the 17th International Federation of Automatic Control*, pages 82–89, 1997.