Noname manuscript No.
(will be inserted by the editor)

Top-down event-driven online workflow recognition

Athanasios S. Voulodimos · Dimitrios I. Kosmopoulos · Nikolaos D. Doulamis · Theodora A. Varvarigou

Received: March 25, 2011

Abstract In this paper a framework for automatic online workflow recognition in industrial environments where the issue of concurrent activities rises, is presented. The framework consists of three main parts: The first part is devoted to detecting activity in specific Regions of Interest (ROIs) of the video sequence. This is effected by separating each frame into ROIs and representing the resulting subimages through feature vectors. By observing these vectors we can determine when there is action in a particular ROI. The second part of the framework lies in examining whether the detected activity corresponds to a workflow related event. This is accomplished by HMM modeling. Finally, the third part employs a string matching based technique to confirm the validity of the observed sequence of events or correct any detection or classification errors. This last step also addresses a top down approach by informing lower system levels (such as image representation or object tracking) about the errors committed. The performance of the proposed approach is thoroughly evaluated under real-life complex visual workflow understanding scenarios, in an industrial plant. The obtained results are compared and discussed.

Keywords Workflow recognition · activity detection · HMM · top down

Athanasios S. Voulodimos
School of Electrical and Computer Engineering, National Technical University of Athens
Tel.: +30-210-7722559
E-mail: thanosv@mail.ntua.gr

Dimitrios I. Kosmopoulos
Department of Computer Science and Engineering - University of Texas at Arlington
E-mail: dkosmo@ieee.org

Nikolaos D. Doulamis
School of Electrical and Computer Engineering, National Technical University of Athens
E-mail: ndoulam@cs.ntua.gr

Theodora A. Varvarigou
School of Electrical and Computer Engineering, National Technical University of Athens
E-mail: dora@telecom.ntua.gr

1 Introduction

Event and activity recognition are domains with significant usefulness in a wide range of applications, thus attracting the interest of many researchers in the areas of computer vision, machine learning, multimedia and image processing. One of the most popular applications is smart monitoring of large-scale enterprises, such as industrial assembly lines, where the importance of activity recognition relates to the safety and security of the staff, the reduction of cost, production scheduling, as well as the quality of the production process. The latter is guaranteed by enforcing adherence to strictly predefined procedures and activities for production or service provision.

In most current approaches the goal is either to detect activities, which may deviate from the norm, or to classify some isolated activities. Attempts to address the problem under discussion are encumbered by a number of important hindering factors; the high diversity of the actions and types of activities that constitute a workflow and have to be recognized, is one of the most important among those. The complexity of static object detection and moving object tracking, with the occlusions and illumination changes, naturally affect adversely approaches that follow the bottom-up approach.

Despite the above impediments, focusing on monitoring the production line of an industrial plant (such as an automobile manufacturer), which is a fairly structured process, makes modeling of the activities more realistic than in the case of a more unsystematic area of interest, e.g. an airport or a service maintenance system. The former processes are often hierarchically structured as workflows, that comprise sequential tasks.

In [13, 5] workflow recognition is achieved with good success rates in an industrial use case like the one described, where however two important assumptions hold: i. the sequences are considered to be appropriately pre-segmented by an expert user, thus decreasing the industrial impact and ii. the workflows in the dataset are very well structured comprising mostly non-overlapping tasks, which is often not the case in real world industrial applications. Regarding the former, a framework that can provide online automatic activity and workflow recognition is bound to have a far more significant industrial impact. As for the latter, it is clear that an image-based feature vector representing a frame where two different tasks are executed in parallel is less successfully handled by the classification model (for the model of the first task, the visual content related to the second task is noise, and vice versa), in comparison to the case where there is no overlap. Moreover, multi-object detection and tracking based methods are also doomed to failure in such complex environments. Therefore the need to address the issue of activity and workflow recognition in complex datasets with overlapping or simultaneous activities and more relaxed structure is apparent.

Additionally, most systems tend to only follow the bottom-up paradigm, i.e. beginning from low level image analysis or object detection/tracking and moving upwards to motion analysis, action/activity recognition and behavior modeling. On the contrary the reverse *top-down* pathway, i.e., exploiting higher

level behavior recognition results to influence lower level image analysis, has been much less researched upon.

Taking these observations into consideration, the work presented in this paper contributes mainly in the following ways:

- We propose an online automatic workflow recognition framework for industrial monitoring, which is a novel contribution to our knowledge.
- We address the challenging issue of recognizing different simultaneous activities that partially or entirely overlap by breaking down the workflows into spatially confined events and by defining and observing appropriate Regions of Interest (ROI).
- The proposed framework also makes an attempt to implement the top-down pathway, by allowing the higher level event and workflow recognition results inform the lower level image analysis.

The remainder of this paper is structured as follows: Related work regarding event and activity recognition is discussed in Section 2, while the problem formulation is presented in Section 3. In Section 4 we describe the event modeling, while in Section 5 we present the online automatic recognition framework. The experimental setup and the outcoming results are described and analyzed in Section 6. Finally, Section 7 concludes the paper with a summary of the findings.

2 Related work

Event detection and especially human action recognition has been the focus of interest of computer vision and machine learning communities for years, mostly as isolated activities and not as part of a continuous process. A variety of methods has addressed these problems, including semilatent topic models [24], spatial-temporal context [9], optical flow and kinematic features [2], and random trees and Hough transform voting [27]. Wada et al. [22] employ Non-deterministic Finite Automaton as a sequence analyzer to present an approach for multiobject behavior recognition based on behavior driven selective attention. Other works focus on more specific domains, e.g., event detection in sports [20], [11], retrieving actions in movies [14], human gesture recognition (using Dynamic Time Warping [3] and Time Delay Neural Networks [26]), and automatic discovery of activities [8]. Comprehensive literature reviews regarding isolated human action recognition can be found in [1], [10].

One of the key functionalities of any machine learning model (classifier) suitable for application in visual behavior understanding is the ability to extract the *signature* of a behavior from the captured visual input. The key requirements when designing such a classifier is (a) to support task execution in various time scales, since a task or parts of it may have variable duration; and (b) to support stochastic processes, because of the task intra-class variability and noise.

A very flexible framework for stochastic classification of time series is the HMM (see e.g., [19]). It can be easily extended to handle outliers (see e.g., [4])

and to fuse multiple streams (e.g., [28]). It is very efficient for application in previously segmented sequences (see e.g.[12]), however, when the boundaries of the sequence that we aim to classify are not known in advance, the search space of all possible beginning and end points make the search very inefficient [6]. A typical way to treat this problem is given in [16], where a dynamic programming algorithm of cost which is proportional to the cube of the duration, is used to perform segmentation and classify then the segments; such a cost is restrictive in real applications.

In the past there have been some efforts to exploit the hierarchical structure of some time series, e.g., by using the hierarchical HMMs [7]. Each state is considered to be a self-contained probabilistic model (an HHMM). Examples of such approaches can be found in [18], where the worflow in a hospital operating room is described. Another approach is the layered hidden Markov model (LHMM) (see [17]), which consists of N levels of HMMs where the HMMs on level $N+1$ corresponds to observation symbols or probability generators at level N. Every level i of the LHMM consists of K_i HMMs running in parallel. In that work a LHMM is used for event identification in meetings. In [25] structure learning in HMMs is addressed in order to obtain temporal dependencies between high-level events for video segmentation. A HMM models the simultaneous output of event-classifiers to filter the wrong detections.

In many workflows, such as in industrial production where a sequence of different tasks has to be completed, the execution of a task means that it will not appear again in the same workflow. Therefore the whole history of tasks must be kept in memory to exclude false positives and the Markovian property is obviously not applicable. Thus, approaches such as the LHMM and the HHMM have an inherent problem to describe such workflows.

3 Problem statement

In this paper the focus is on detecting and recognizing visual tasks in complex industrial processes (workflows), being executed in an automobile production line. The visual tasks are recognized from visual cues being captured from a camera.

A workflow is a process that happens repetitively and consists of a sequence of discrete tasks. The order in which tasks appear matters, however permutations are allowed in some cases (which have to be learned). Tasks may have different durations, as a result of the natural differences in workers' productivity and other situational parameters. The definition of tasks stems from domain knowledge. An example of such a task is: "A worker picks part1 from rack1 and places it on the welding cell". The goal is to determine which tasks are executed and when, given instances of workflows, which are described by sequences of visual observations, so as to automatically "supervise" the successful completion of workflows.

Nevertheless, the significant overlap between task execution described in Section 1 dictates a necessary adjustment to the above described formulation.

To address the issue of simultaneity, we split the tasks into shorter sub-tasks or "events", whose appropriate execution according to a rule system indicates the execution of tasks, and consequently the completion of workflows. The sub-tasks/"events" definition is done in such a way to ensure the fulfillment of the following assumptions:

1. Workflow recognition is possible through the events recognition combined with an intelligent events combination and evaluation system .
2. Events are as spatially confined as possible, so that they can be observed in specific Regions Of Interest (ROIs) efficiently.
3. Events are as temporally short as possible so that the simultaneity phenomenon is less frequent.

A more formal statement of the problem under discussion would be the following: Given an image sequence $\mathcal{I} = \{I_0, \ldots, I_t\}$ and a set of $E+1$ possible events/"subtasks", $\mathcal{E}^{\#} = \{1, \ldots, E, \#\}$, where $\#$ corresponds to activity not related to the workflow, we want to associate an event $e_A^{\star} \in \mathcal{E}^{\#}$ with a time interval $A_k : [t_{sk}, t_{ek}]$, $k = 1, 2, ..., E$, where t_{sk} and t_{ek} are the starting and ending times of the event k in the workflow and, in general, $A_m \cap A_n \neq \emptyset$, subject to a $\mathcal{R}(e_k)$ set of constraints. The aforementioned constraints pertain to the particularities and interdependencies of the observed industrial process, where different permutations are allowed.

4 Event modeling

Here we describe the representation of the sub-tasks/events to be recognized. In subsection 4.1 we present means of representation of each frame, while in subsection 4.2 we present the HMM for modeling time series.

4.1 Visual Observations

One of the key challenges real-time action recognition systems are confronted with concerns selection of appropriate features for representing the observed raw data. The ideal features should describe different actions accurately, with high discrimination capability, and should be efficiently calculated. Ideally, these features should also provide a hierarchical representation scheme (coarse to fine) so that a desirable, application-wise trade-off between representation capabilities and computational complexity can be reached.

The employment of features directly extracted from the video frames has the significant advantage of *obviating the need of detecting and tracking the salient scene objects*, a task which is notoriously difficult in cases of occlusions, target deformations, illumination changes, etc. Thus, by using such an approach, the intermediate levels of semantic complexity, as met in typical *bottom-up* systems, are completely bypassed. For this purpose, either local or holistic features (or both [21]) may be used. *Holistic features* remedy these

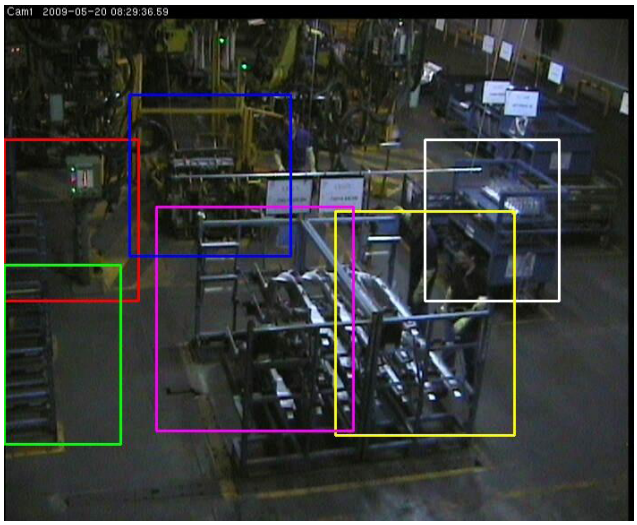


Fig. 1 Depiction of six ROIs defined in the working area of the automobile manufacturer dataset. Each ROI observes the execution of one or more events

drawbacks of local features, while also requiring a much less tedious computational procedure for their extraction. In [12], we have described how to represent pixel change history (which is able to capture the motion history of foreground objects) using the Zernike moments.

However, addressing the challenge of concurrent events taking place in different parts of the image requires us to take this approach further. The first step is to define appropriate Regions of Interest (ROI), so that every event takes place in a specific ROI (however a given ROI may be the area of execution of more than one events). The selection of ROIs naturally depends on the particularities of the application to be monitored, nonetheless as a general rule their number and size should be such as to avoid both having many events happening in the same ROI and observing an unreasonably large number of ROIs, which would have computational cost and reduce the framework's impact. Figure 1 depicts the ROIs defined for our automobile industry production line dataset.

In our approach, the feature extraction process described above is performed for each ROI of every frame separately. Each frame will therefore be represented by R feature vectors (descriptors) $\mathbf{o}_{rt}$, one for each observed ROI, where t is the time instance, or in discrete domain the frame number, and $r = 1..R$ the ROI.

4.2 Single event modeling and recognition through HMM

In this subsection we briefly describe the modeling process for the observed events. A common approach for stochastically modeling time series is to use

Hidden Markov Models (HMMs). A Hidden Markov Model consists of states, transitions, observations and probabilistic behavior, and is formally defined as a tuple $\lambda = \langle \mathbf{Q}, \mathbf{A}, \mathbf{B}, \pi \rangle$ satisfying the following conditions:

- $\mathbf{Q} = \{q_1, ..., q_S\}$ is a finite set of S states. In our case, the number of states is an indication of the order (complexity) of the stochastic representation.
- $\mathbf{A}$ is the transition matrix, which represents the transition probabilities between states.
- $\mathbf{B}$ is the observation matrix, which represents the observation probability given the state.
- π represents the probability of each state at the beginning of the sequence.

A supervised training algorithm is used to obtain the parameters λ of the HMM. The training set is formed using representative samples of industrial tasks or, in our case, subtasks/events which have been manually (supervisedly) classified to one of the L available classes. This implies that we need first to annotate the tasks/events, exploiting, for example, the experience of industrial engineers. We also need to identify the start and finish times for each industrial workflow even during the testing phase something which is a burden for a real-life exploitation of the algorithm in industrial environments. In real-world scenarios it is usually unknown when a task starts or finishes. Therefore, HMM modeling by itself can not be used for online recognition of the tasks. This is because online classification requires searching in the space of possible beginning and end points to perform Viterbi matches in order to find the optimally fitting sequence [19].

Assuming that tasks' appearance follow Markovian behavior (the conditional probability distribution of future tasks depends only upon the present task; that is, given the present, the future does not depend on the past) it is possible to perform online classification by applying techniques such as hierarchical (HHMM) and Layer hidden Markov models (LHMM) [7] [17]. However, such assumptions are not true in a real-world industrial environment, since the processes considered are structured. Usually, in a real-world production environment, the current execution of a task will affect the execution of future tasks, i.e., a task may be executed only once in a workflow.

All the above imply that the use of a sole conventional HMM for stochastically classifying industrial tasks is overall inefficient, especially for real world sequences, which typically contain several thousands of frames. An exhaustive search for all possible combinations would be therefore practically prohibitive from a computational point of view. Hence, for an online automatic recognition framework, we need to identify the time boundaries, that is the start and finish times of an industrial task or subtask/event, which are part of a workflow. For this reason, we propose an alternative methodology that incorporates HMM into a framework that exploits the information given by the ROI-specific visual observation vectors described in subsection 4.1 and evaluates/combines recognition results via a top-down approach.

5 The online automatic workflow recognition framework

Algorithm 1 presents the steps of the proposed online event driven workflow recognition framework, which is described in detail in the subsections to follow.

5.1 Activity detection in ROIs

The first phase of the online recognition framework (lines 15-36 of Algorithm 1) consists of image representation for every frame and detection of change in specific ROIs of the observed sequence. For every frame captured by the surveillance camera, r different subimages are created corresponding to the r ROIs defined. For each of the r subimages a feature vector representing that particular subimage is calculated as described in [12]. Each frame F corresponding to time t is therefore represented by r different feature vectors o_{rt}, one for each ROI. As explained in subsection 4.1, these features are based on Pixel Change History; a zero feature vector thus indicates lack of visual change in the subimage, and consequently lack of activity in the respective ROI as well. As a result, the appearance of a non-zero feature vector after a series of many consecutive zero vectors denotes the beginning of some sort of activity in that ROI, whether it be a workflow related event or irrelevant action. As soon as some sort of activity is detected, the framework "observes" the continuation of this activity in the specific ROI by buffering the successive non-zero feature vectors to form a "candidate" sequence of vectors that might correspond to an event. This candidate sequence is considered to have ended after a certain number (determined by scene representation statistics) of consecutive zero vectors.

5.2 Event classification

The second phase of the framework (lines 37-47) focuses on recognizing whether the detected activity corresponds to a workflow related event and classifying it appropriately. This phase requires an offline training process based on the model described in subsection 4.2, through which we train one separate HMM for each of the events with appropriate ROI specific sequences to obtain the λ parameters of each HMM (lines 1-5). The candidate sequence created from the first phase can now be tested against all possible events pertaining to the particular ROI that produced this sequence. More specifically, we calculate the observation probability for the candidate sequence given each of the ROI related HMMs. If more than one events are linked with the particular ROI (which generally and most frequently is the case), then the event corresponding to the maximum observation likelihood is recognized as the executed subtask/event if it surpasses a certain threshold. In the case where only one event (and therefore HMM) is linked with the particular ROI, the observation likelihood only needs to be greater than the related threshold so that the

candidate sequence can be successfully recognized as the corresponding event. The aforementioned thresholds are calculated from training process statistics. This way, the detected activities can be classified as workflow related events or rejected as irrelevant activity.

5.3 Workflow recognition and error correction following a top-down pathway

The series of recognized events as generated from the first and second phase creates a sequence which has to be approved so that the execution of a workflow can be regarded as accomplished. This is the objective of the third phase of the framework (lines 49-58). As has been mentioned before and as is usually the case in many industrial processes, the hierarchy of the occuring events/"subtasks" is significant, however some permutations are allowed. In order to automatically check whether the sequence of recognized events matches one of the permitted permutations, and which one in particular among the latter, we employ an approach that is based on the Levenshtein distance [15].

Levenshtein distance (or edit distance) is the minimal quantity of character substitutions, deletions and insertions for transformation of a string s1 into string s2. In particular we have modified the Wagner Fischer algorithm [23] so as to accommodate the needs of our application domain. The Wagner Fischer algorithm [23] employs dynamic programming methods to calculate the Levenshtein Distance between any pair of strings. It uses an iterative process to find successive distances between increasingly longer pairs of prefixes of the two strings, computed with the aid of a matrix. In our case study, we regard the events as characters, while the sequences of recognized events as well as the allowed workflows as strings. The main modification introduced lies in the definition of the cost weights of the insertions / deletions / substitutions. In the standard version of the algorithm, the costs of all possible operations are equal to 1. In our method, each cost depends on the likelihood of the corresponding operation. For example, the substitution of an event e_1 by an event e_2 carries a lower cost when the two events bear significant visual resemblance and are thus difficult to distinguish, and a higher cost when they are totally different and much harder to confuse; moreover, the cost of the insertion of an event is high when the execution of another event is a prerequisite and the latter does not precede the former in the observed sequence.

As soon as a workcycle ends, the sequence of recognized events is compared against all possible legitimate workflow permutations. If there exists a workflow for which the calculated distance is zero, then we have a total match, and the workflow as well as its recognition are successful. If, on the other hand, there is no workflow permutation providing a total match, then the workflow yielding the minimum edit distance is selected as the true target workflow. Following the path that corresponds to the latter, the suggested event substitutions, insertions, or deletions can be determined, thus providing information about erroneous event classifications during the preceding steps of the recognition process. Conclusions can also be drawn as regards activity detection

in the respective ROIs therefore informing the interpretation of lower levels of the framework, such as motion detection. This way, not only is workflow recognition succesfully performed, but also a top down approach is provided, thus offering substantial added value to the usual bottom up approach based system.

6 Experiments and Results

We experimentally validated the proposed methods with video sequences obtained from a real assembly line of a major automobile manufacturer. The acquired datasets contain information pertaining to the production process of a real vehicle manufacturing facility. The workflow on this assembly line included tasks of picking several parts from racks and placing them on a designated cell some meters away where welding was performed. The information acquired from the recognition process could be used for the extraction of production statistics, anomaly detection and guarantee of safety and security.

6.1 Experimental setup

Adhering to the guidelines stated in Section 3 and taking into consideration the nature of the industrial process observed, we have defined 12 events, which are spatially confined in the six ROIs depicted in Figure 1. Each of these events was regarded as a class of behavioral patterns that had to be recognized. The events are briefly described as follows:

1. Worker picks part #1 from rack #1.
2. Worker places part #1 on the welding cell.
3. Two workers pick part #2a from rack #2.
4. Two workers place part #2a on the welding cell.
5. Two workers pick part #2b from rack #3.
6. Two workers place part #2b on the welding cell.
7. Worker picks up parts #3a and #3b from rack #4.
8. Worker places parts #3a and #3b on the welding cell.
9. Worker picks up part #4 from rack #1.
10. Worker places part #4 on the welding cell.
11. Worker(s) pick up part #5 from rack #5.
12. Worker(s) place part #5 on the welding cell.

The workspace configuration and the cameras' positioning is given in Figure 2. For our experiments, we have used 20 sequences representing full assembly cycles, each one containing each of the defined events. The length of

Algorithm 1 Our method

{OFFLINE TRAINING}
{Supervised event learning through HMM}
for $e = 1$ to $NumberOfEvents$ **do**
  $\prec \mathbf{Q}_e, \mathbf{A}_e, \mathbf{B}_e, \pi_e \succ$ = TrainHMM(EventTimeSeries)
end for
{INITIALIZATIONS}
Set value for $\mathbf{Threshold}_e$ from training statistics
Set value for **EndDecision** from scene representation statistics
for all r **do**
  seq_r=null;
  $candidateSeq_r$=null;
end for
$ActivityHappening$=FALSE;
{ONLINE RECOGNITION}
while (F=AcquireFrame()) $\neq$ NULL **do**
  for $r = 1$ to $NumberOfROIs$ **do**
    $\mathbf{o}_{rt}$ = ProcessFrame(F); {extraction of visual observations (features) by processing the image that corresponds to ROI r of the current captured frame}
  end for
  if $ActivityHappening$ = TRUE **then**
    $seq_r = append(seq_r, F)$
    if $\mathbf{o}_{rt} \neq 0$ **then**
      {non zero feature vector means that there is change in the image from the previous frame; therefore can be a sign of activity in the ROI}
      $lastNonZero[seq_r] = length(seq_r)$
    else
      {if zero feature vector}
      if $(length(seq_r) - lastNonZero[seq_r]) \geq$ EndDecision **then**
        {a certain number of zero vectors denotes ending of activity}
        $candidateSeq_r = seq_r(1..lastNonZero)$;
        $ActivityHappening$ =FALSE;
        seq_r=null;
      end if
    end if
  else
    {if ActivityHappening=FALSE}
    if $\mathbf{o}_{rt} \neq 0$ **then**
      $ActivityHappening$ =TRUE; $seq_r = append(seq_r, F)$;
    end if
  end if
  if $candidateSeq_r \neq$NULL **then**
    for every event e_k pertaining to ROI r **do**
      Calculate observation probability $p(candidateSeq_r|e_k)$ of $candidateSeq_r$ given the corresponding HMM
      $\hat{e} = argmax_k(p(candidateSeq_r|e_k))$
      if $p(candidateSeq_r)|\hat{e} >$ Threshold_e **then**
        Event Recognized;
        $w = append(w, \hat{e})$;
      end if
    end for
  end if
end while
{No more frames - workcycle completed}
for every possible workflow w_j **do**
  Calculate the Workflow Distance $WD(w, w_j)$;
end for
$\hat{w} = argmin_j(WD(w, w_j))$
if $WD(w, \hat{w}) = 0$ **then**
  {Total match; workflow w successfully recognized.}
else
  {No total match was found; The closest valid workflow is $\hat{w}$;}
  {Following the pathway of the calculation of $WD(w, \hat{w})$ the erroneous classifications are determined. The lower levels are appropriately informed.}
end if

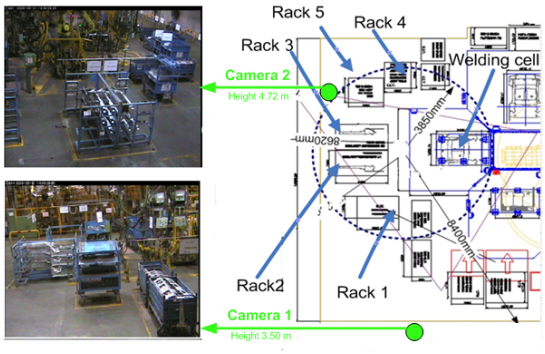


Fig. 2 Depiction of a workcell along with the position of our camera (camera 2) and the racks #1-5.

each sequence ranges from 2,000 frames to 3,500[1]. The annotation has been done manually.

In each workflow all 12 events/subtasks are performed, in different orders and often simultaneously. Challenges of the dataset include occlusions, visually complex background, similar colours, variable subtask and task durations, high intra-class and low inter-class variance. Moreover, the silhouettes get overlayed in a random fashion, thus making the motion signatures much more difficult to model. The core challenge, however, lies in the concurrent execution of different subtasks in various parts of the scene.

6.2 Scene representation

To represent each video frame with r feature vectors, where r is the number of ROIs in the image, we followed the method described in the subsection 4.1. For capturing the spatiotemporal variations we have set the parameters at $\varsigma = 10$ and $\tau = 70$. We have chosen to use the Zernike moments up to sixth order along with the center of gravity and the area, as feature vector for each one of the r sub-images stemming from the r ROIs of each frame. The higher the order of moments that we employ, the more detailed the region reconstruction will be, but also the more processing power will be required.

Specifically we employed the complex moments A_{00}, A_{11}, A_{20}, A_{22}, A_{31}, A_{33}, A_{40}, A_{42}, A_{44}, A_{51}, A_{53}, A_{55}, A_{60}, A_{62}, A_{64}, A_{66} for each of which we used the norm and the angle, except for A_{00},A_{20},A_{40},A_{60} for which the angle was always constant. Additionally the center of gravity and the area were used, making a total of 31 parameters, thus providing an acceptable scene reconstruction without a computationally prohibitive dimension.

[1] We are going to make the dataset publicly available. It is currently available for review purposes on http://www.4shared.com/dir/sYeCqK5d/SignalProcessingVideoAnalytics.html (folder:dataset2 - password:xyz543)

For event recognition we used three-state HMMs with one mixture component per state to model each of the events/subtasks described above, which is a good trade-off between performance and efficiency. In all cases, we employed full covariance matrices for the adopted observation (mixture) models. We trained all our models using the EM algorithm and we used the first ten scenarios for training and the rest ten for testing.

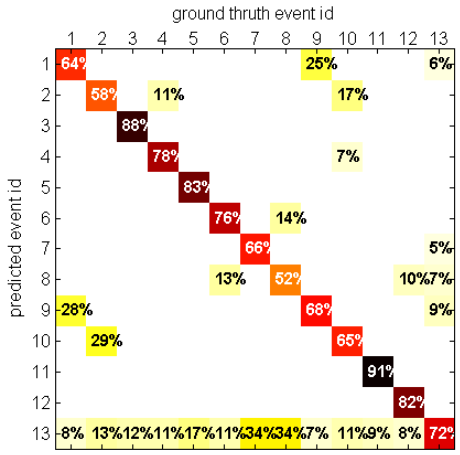
6.3 Results

To begin with, it should be noted that the same dataset was used in [12] and the results for task identification were not better than 60%, using multicamera fusion. This highlights the difficulty of the task by using a holistic approach without using ROIs.

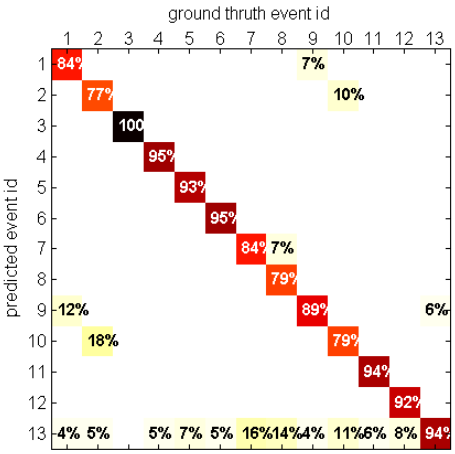
On the contrary, the methodology described in this paper provided rather promising results. The event detection and classification method led to satisfactory recognition rates. The related confusion matrix showing the results of the event detection and classification method before employing the error correction / workflow recognition is depicted in Figure 3(a). The success rates are lower in the cases where two events bear significant visual resemblance, such as event 1 with event 9, and event 2 with event 10. The 13th "event" corresponds to void, that is a "non-event". We further extracted recall and precision. Recall indicates the number of true positives divided by the total number of positives in the ground truth ($REC = TP/(TP + FN)$). Precision is the number of true positives divided by the number or true and false positives ($PRC = TP/(TP + FP)$). Average Recall in this case is $72.6 \pm 12.2\%$ while average Precision is $83.4 \pm 15.7\%$.

Following, we examine the effect of the error correction and workflow recognition part of the framework, as described in subsection 5.3. Figure 3(b) shows the confusion matrix after the modified Levenshtein distance based mechanism has been employed in order to scrutinize the matching to a valid workflow. The improvement in the success rates is significant, since Recall is increased to $88.4 \pm 7.6\%$ and Precision up to $92.1 \pm 7.1\%$.

Figure 6.3 displays the flow of an example scenario over a frame timeline. Figure 4(b) displays the detections and classifications of events resulting from the first and second part of the framework. As can be seen, the detector/classifier gives good results in general; it recognizes most events correctly and more importantly, events 8 and 9, which happen simultaneously. However, it produces some erroneous detections and classifications: first, an inexistent occurrence of event 1 is made just before event 3, when in the ground truth there is no event happening at that time (false positive); second, event 10 is fallaciously classified as event 2 (which is not surprising given the visual resemblance of these two events). These two serious errors are rectified when the third part of the framework is employed. By trying to match the observed workflow to a valid sequence, the framework leads to the deletion of the false positive event 1, and the substitution of the second occurrence of event 2 by



(a) Event detection and classification



(b) Workflow recognition / error correction

Fig. 3 Confusion matrices

an occurrence of event 1. This way, the observed final result (Figure 4(c)) is very close to the ground truth (Figure 4(a)). What's more, these two major rectifications (deletion of inexistent event 1 and substitution of erroneous event 2 by event 10) can inform the lower levels of the system (image based or object detection and tracking modules, if those are employed in parallel) to help them enhance their performance, thus contributing to the implementation of a top-down pathway.

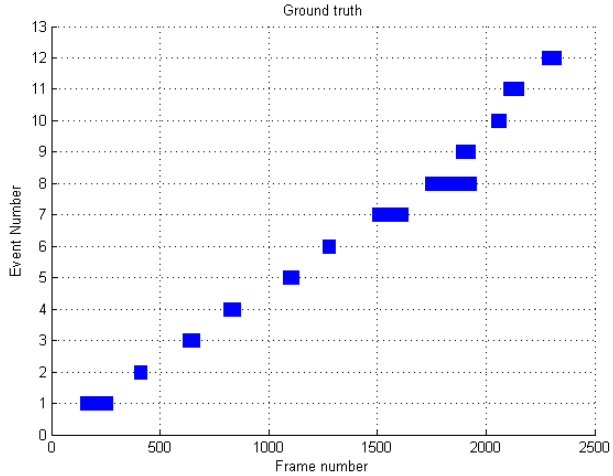
7 Conclusion

In this work we have proposed a novel online framework for event driven workflow recognition in industrial environments in real-time. In the context of the framework we have handled the important problem of recognizing concurrent or overlapping events that may happen simultaneously on the same video sequence. This is effected by observing different Regions of Interest and extracting separate feature vectors for each one of them; we then employ HMM to model the events and following, a modified string matching technique based on the Levenshtein distance is used to evaluate the validity of the extracted recognized results and to correct erroneous event detections and classifications. The latter approach gives the opportunity to improve the performance of lower levels (motion detection or object tracking) by informing them about the errors committed during the event detection and classification processes, thus contributing to the realization of a top down pathway, along with the usual bottom up paradigm. The proposed methods have been applied with promising results in some very challenging real sequences from an automobile manufacturing process. The event detection and classification processes followed by error correction and workflow matching/recognition result in a very successful overall performance of the framework.

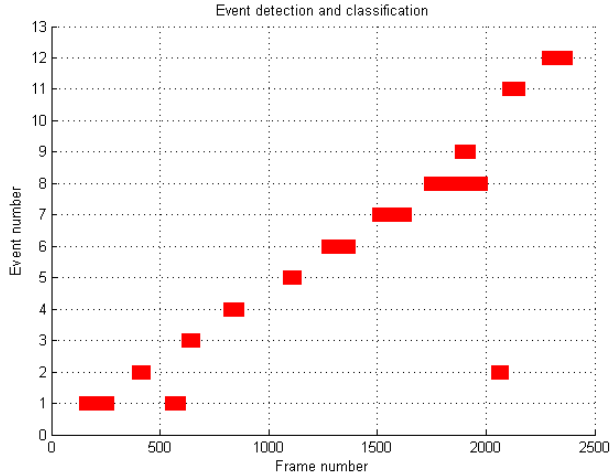
References

1. Aggarwal, J.K., Cai, Q.: Human motion analysis: A review. Computer Vision and Image Understanding **73**(3), 428 – 440 (1999)
2. Ali, S., Shah, M.: Human action recognition in videos using kinematic features and multiple instance learning. Pattern Analysis and Machine Intelligence, IEEE Transactions on **32**(2), 288 –303 (2010). DOI 10.1109/TPAMI.2008.284
3. Bobick, A., Wilson, A.: A state-based approach to the representation and recognition of gesture. Pattern Analysis and Machine Intelligence, IEEE Transactions on **19**(12), 1325 –1337 (1997). DOI 10.1109/34.643892
4. Chatzis, S.P., Kosmopoulos, D.I., Varvarigou, T.A.: Robust sequential data modeling using an outlier tolerant hidden Markov model. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(9), 1657–1669 (2009)
5. Doulamis, N., Voulodimos, A., Kosmopoulos, D., Varvarigou, T.: Enhanced human behavior recognition using hmm and evaluative rectification. In: ACM Multimedia, ARTEMIS Workshop (2010)
6. Eickeler, S., Kosmala, A., Rigoll, G.: Hidden markov model based continuous online gesture recognition. In: In Int. Conference on Pattern Recognition (ICPR, pp. 1206–1208 (1998)
7. Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden markov model: Analysis and applications. Machine Learning **32**(1), 41–62 (1998)
8. Hamid, R., Maddi, S., Bobick, A., Essa, M.: Structure from statistics - unsupervised activity analysis using suffix trees. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1 –8 (2007). DOI 10.1109/ICCV.2007.4408894
9. Hu, Q., Qin, L., Huang, Q., Jiang, S., Tian, Q.: Action recognition using spatial-temporal context. In: Pattern Recognition (ICPR), 2010 20th International Conference on, pp. 1521 –1524 (2010). DOI 10.1109/ICPR.2010.376
10. Hu, W., Tan, T., Wang, L., Maybank, S.: A survey on visual surveillance of object motion and behaviors. Systems, Man and Cybernetics, Part C, IEEE Transactions on **34**(3), 334–352 (2004). DOI 10.1109/TSMCC.2004.829274

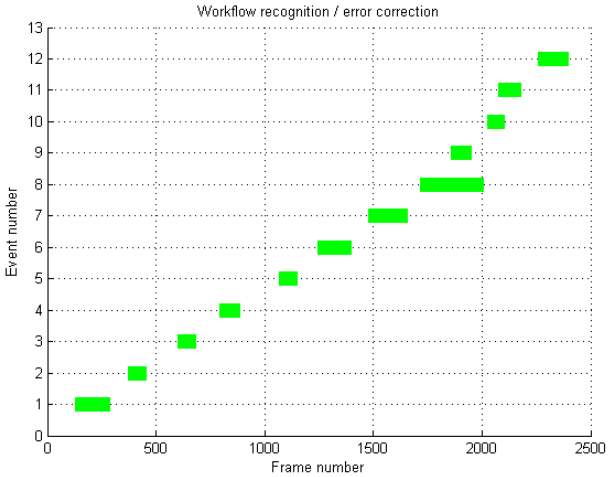
11. Hung, M.H., Hsieh, C.H.: Event detection of broadcast baseball videos. Circuits and Systems for Video Technology, IEEE Transactions on **18**(12), 1713 –1726 (2008). DOI 10.1109/TCSVT.2008.2004934
12. Kosmopoulos, D., Chatzis, S.: Robust visual behavior recognition. Signal Processing Magazine, IEEE **27**(5), 34 –45 (2010). DOI 10.1109/MSP.2010.937392
13. Kosmopoulos, D., Voulodimos, A., Varvarigou, T.: Robust human behavior modeling from multiple cameras. In: Proceedings - International Conference on Pattern Recognition, pp. 3575–3578 (2010)
14. Laptev, I., Perez, P.: Retrieving actions in movies. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1 –8 (2007). DOI 10.1109/ICCV.2007.4409105
15. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Tech. Rep. 8 (1966)
16. Lv, F., Nevatia, R.: Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In: ECCV06, pp. IV: 359–372 (2006)
17. 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). DOI http://dx.doi.org/10.1016/j.cviu.2004.02.004
18. Padoy, N., Mateus, D., Weinland, D., Berger, M.O., Navab, N.: Workflow Monitoring based on 3D Motion Features. In: Workshop on Video-Oriented Object and Event Classification in Conjunction with ICCV 2009, pp. 585–592. IEEE, Kyoto Japan (2009)
19. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
20. Sadlier, D., O'Connor, N.: Event detection in field sports video using audio-visual features and a support vector machine. Circuits and Systems for Video Technology, IEEE Transactions on **15**(10), 1225 – 1233 (2005). DOI 10.1109/TCSVT.2005.854237
21. Sun, X., Chen, M., Hauptmann, A.: Action recognition via local descriptors and holistic features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 58–65 (2009)
22. Wada, T., Matsuyama, T.: Multiobject behavior recognition by event driven selective attention method. Pattern Analysis and Machine Intelligence, IEEE Transactions on **22**(8), 873 –887 (2000). DOI 10.1109/34.868687
23. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. J. ACM **21**, 168–173 (1974). DOI http://doi.acm.org/10.1145/321796.321811. URL http://doi.acm.org/10.1145/321796.321811
24. Wang, Y., Mori, G.: Human action recognition by semilatent topic models. Pattern Analysis and Machine Intelligence, IEEE Transactions on **31**(10), 1762 –1774 (2009). DOI 10.1109/TPAMI.2009.43
25. Xiang, T., Gong, S.: Optimising dynamic graphical models for video content analysis. Comput. Vis. Image Underst. **112**, 310–323 (2008)
26. Yang, M.H., Ahuja, N.: Extraction and classification of visual motion patterns for hand gesture recognition. In: Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, pp. 892 –897 (1998). DOI 10.1109/CVPR.1998.698710
27. Yao, A., Gall, J., Van Gool, L.: A hough transform-based voting framework for action recognition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2061 –2068 (2010). DOI 10.1109/CVPR.2010.5539883
28. Zeng, Z., Tu, J., Pianfetti, B., Huang, T.: Audio-visual affective expression recognition through multistream fused HMM. IEEE Trans. Multimedia **10**(4), 570–577 (2008)



(a) Ground truth



(b) Event detection and classification



(c) Workflow recognition / error correction

Fig. 4 Workflow related events of an example scenario on a timescale as they appear (a) in ground truth, (b) after event detection and classification, (c) after workflow recognition and error correction